* * * ZUM NEUEN JAHR * * *

Hallo Clubfreunde !

Wann auch immer dieses INFO erschienen ist - ich wünsche uns allen ein erfolgreiches Einläuten der letzten Dekade dieses Jahrtausends!

(Es gab Ende 1989 eine TV-Sendung, in der tatsächlich behauptet wurde, ein Jahrzehnt fange an, wenn hinten eine "1" steht, also erst 1991. Wir wissen das besser. Jede vernünftige Zählung beginnt mit 0 (Null) ! Das Jahr 2000 ist demnach bereits der Beginn des 21.Jahrhunderts! Ich erwarte deshalb, daß der Beginn des 3.Jahrtausend offiziell in genau 10 Jahren, nämlich 2000 gefeiert wird. Dann werden nicht nur Computer-Clubs, sondern alle Zeitgenossen die 10 Ziffern nicht mehr

"1234567890",

sondern "0123456789"

aufsagen.)

Wie altmodisch, ja unkonsequent ist doch mein alter TRS-80; denn die obere Zahlen-Tastenreihe beginnt noch mit 1, der Ziffernblock jedoch mit 0...

Doch noch ist es nicht soweit. Noch haben wir andere Sorgen. Werden wir neben den "Großen", den "Kompatiblen", den Ibe-Ems, den Aibi-Ems und den fortgeschrittenen DOSes nebst Windows etc.pp. noch bestehen können?

Inzwischen habe ich aus so manchen Beiträgen und Zuschriften *(danke, Oskar!)* entnehmen können, daß es eine grabentiefe Kluft gibt zwischen den 'Professionellen' einerseits und den "Freaks" andererseits: Dort die "USER" - das sind die Leute, die

"Universale Sachen Endlos Reproduzieren" -

hier der "FREAK", der

"Freie Routinen Erschafft Am Keyboard"!

Was wäre aber unser "Häuflein der 82 Aufrechten" (14.9.89), das nicht Gottfried Keller, sondern wasweißich für ein Gott noch im Bastelkeller zusammenhält, ohne ein verbindendes Organ, ohne sein lesbares INFO?

Nun bringt jedes Periodical*), das etwas auf sich hält, zum Jahresanfang ein Verzeichnis der Publikationen des abgelaufenen Jahres heraus, also ein sogenanntes Jahres-Inhaltsverzeichnis. So auch wir.

Da seit dem letzten Verzeichnis zwei Jahre vergangen sind, habe ich nicht nur die Beiträge der Hefte von 1989 (es waren ihrer ganze 3!) aufgelistet, sondern nochmals alle Beiträge seit Bestehen des CLUB-80 zusammengestellt. Am alten Verzeichnis wurden bei dieser Gelegenheit einige Korrekturen angebracht.

Als Datenbanksystem verwendete ich nicht DBASE, sondern S-DBAZ. Dies ist nämlich die Schnellste DatenBank Aller Zeiten, etwa 8-mal schneller als "dBASE". Sie läuft unter NEWDOS und befindet sich nach dem Laden komplett im Arbeitsspeicher. Man kann die Taste, die eine Anfrage auslöst, gar nicht so schnell wieder loslassen, wie die Antwort - das Gesuchte - auf dem Bildschirm erscheint! DBASE kann das in der Tat nicht, sondern muß sich immer wieder erst die nächste Portion Daten (einen "Block"?) von der Diskette (Platte) holen. (S-DBAZ hieß früher AIDSIIL..)

Ich wünsche euch im Neuen Jahr ebenso schnelle Erfolge beim Heben der Schätze aus den Tiefen der bisherigen INFOS!

Good Luck and Huck in XMM MXM!

*) ...

PAGE 1 - BEITRAEGE ZUM TH[illegible] SOFTWARE; STAND: 31.12.89

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
AMATEURFUNK	Die Betriebsart PACKET-Radio (PR)	HARTMANN, H.-G.	28/111
ASCII	Wogegen selbst ZEUS vergeblich kaempft	MUEHLENBEIN	25/05
	ASCII-TABELLE IMMER ZUR HAND	SOPP	19/19
ASSEMBLER	KLEINSCHRIFT fuer den EDTASM+	HEIDENREICH	20/37
	SCHUETZEN VON ASSEMBLERPROGRAMMEN	KONRAD J.	5 / ?
	HALBAUTOMATISCHE FESTLEGUNG VON ORG	KONRAD, J.	9/3
	MEIN DRITTES ASSEMBLERPROGRAMM	MUEHLENBEIN	16/33
	MEIN ERSTES ASSEMBLERPROGRAMM	MUEHLENBEIN	14/9
	"INPUT"	SCHROEDER	11/5
	ASSEMBLER CONTRA LISP	SCHROEDER	17/19
	DIE WERKZEUGE DES ASSEMBL.PROGRAMMIERERS	SCHROEDER	21/9
	EINIGE ASSEMBLER-ROUTINEN	SCHROEDER	14/17
	LMOFFSET with DISABLE DOS	SCHROEDER	9/9
	WERKZEUGE DES ASSEMBLER-PROGRAMMIERERS	SCHROEDER	21/9
	Assembler unter CP/M	SCHROEDER, G.	25/49
	Bedingt assemblieren mit ZEUS	SOPP	25/24
	Bedingte Assembly - notfalle ohne IF	SOPP	25/28
	LPRINT ALLES einfacher! (Teil 3)	SOPP	9/11
	SCHOENER LISTEN MIT ZEUS	SOPP	18/41
	WER HAT ANGST VOR ASSEMBLER?	SOPP	5 / ?
	Was tun bei zerschossener Source ?	SOPP	24/25
	ZEUS MIT 64 UND 80 ZEICHEN	SOPP	18/36
	BASIC RUFT ASSEMBLER	WOLLSCHLAEGER	21/11
	Z80-Programmtricks in Assembler	WOLLSCHLAEGER	27/39
	TIP FUER SCHNELLERES ARBEITEN MIT EDTASM	ZWICKEL	6/67
BANKING	256K-RAM fuer Z80-SYSTEME	BERNHARDT	14/91
	Bangen um Banken	MUEHLENBEIN	25/07
	SYSCOPY1 und RAMDISK f.HELMUT's BANKER	SCHROEDER	16/32
	Nochmaliges Banken	SOERENSEN	28/47
BASIC	EXTENDED BASIC(RESTOREn u. LPRINT alles)	OBERMANN	11/33
	EIN MYSTERIOESER BASIC-BEFEHL: IsA	SOPP	5 / ?
BERICHT	Testbericht KX 1083	MAND	23/89
	GENIE IIs - Erfahrungsbericht	SCHROEDER	9/67
BILDFORMAT	FUTTER fuer GENIE IIIs CRTC	HEIDENREICH	20/22
BILDSCHIRM	BILDSCHIRMAUSGABE mit INVERSEN ZEICHEN	?	12/79
	BIN ICH STRAHLENGEFAEHRDET?	BOIKAT, Ute	20/109
	DAS PROJEKT (FERNSEHER --> MONITOR)	MUELLER, K.	18/61
	NOCH EIN BILDSCHIRMSCHONER	SCHMID, A.	22/26
	SCHWIMMENDE BILDER	SCHMID, A.	22/66
	80-ZEICHENKARTE X80, SEDIT und TSCRIPS	SCHROEDER	13/17
	DIE 80-ZEICHEN-KARTE	SCHROEDER	12/63
	EIN NEUER 80ZEICHEN-TREIBER	SCHROEDER	19/77
	EIN SCHONER fuer die BILDROEHRE	SOPP	20/20
	EIN WEITERER BILDSCHIRMSCHONER	SOPP	22/28
	FUTTER fuer den CRTC	SOPP	19/40
	MEHR FERNSEHEN fuer's GELD (f.GENIE IIIs	SOPP	14/21
	NOCH MEHR FERNSEHN fuers GELD	SOPP	19/39
	VIDHEX - HEXANZEIGE d.BILDSCHIRMS m.HRG	SOPP	9/13
CHIP	BEI SIEMENS HEISST ES: CHIP CHIP HURRA!	ZEITSCHRIFT	18/76
	ERSTER MEGABIT-CHIP IN SERIE	aus Zeitschrift	18/18
	DER LEIDENSWEG DER CMOS-CHIPS	aus c't	19/67
	NANOHENRY BLOCKIEREN DEN FORTSCHRITT	aus c't	19/70
CODE	BASICODE-INTERFACE	?(aus ELEKTOR)	3 / ?
	BASICODE-2	?(aus Elektor)	3 / ?
COMPILER	BASCOM-KURZANLEITUNG	MUEHLENBEIN	13/38
	RPNL - DIE ANDERE ART ZU PROGRAMMIEREN	MUELLER, KURT	15/57
	RPNL-BESCHREIBUNG	MUELLER, KURT	16/37
	RPNL	Mueller, Kurt	26/31
	CALL mit BASCOM	OBERMANN	12/33
COMPUTER	MODEL 4p ROM-IMAGE-LOADER	OBERMANN	15/43
	TANDY<--> SCHNEIDER	OBERMANN	17/49
	UNTERSCHIEDE (zw. MODEL 1 und 3 bzw. 4)	OBERMANN	14/55

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
COMPUTER	CP/MAC-eine BRUECKE zw.G-/NEWDOS u.CP/M	RENSCH	18/44
	WIE PHOENIX AUS DER ASCHE (4 COMP.in 1)	SCHROEDER	17/9
	DEIN GIIIs... (NACHTRAG)	SOPP/BERNHARDT	17/54
	TRS-80 MODELL 4p, DER SCHOENSTE PORTABLE	WOLLSCHLAEGER	12/69
	TANDY - BETTER AGAIN	ZEITSCHRIFT	16/40
	32 BIT zum NIEDRIGPREIS (TANDY 4000)	aus - ? -	22/79
CONTROLLER	FLOPPYCONTROLLER-KOMPATIBILITAET	RETZLAFF	19/47
CP/M	WAS HABT IHR GEGEN CP/M ?	BERNHARDT	21/53
	GENIE, CP/M UND 80 ZEICHEN UND HRG???	HELD	21/43
	PROMMER 80 unter G-DOS	MAGNUS	28/58
	BDOS-Replacement - was'n das?	SCHMID, A.	26/27
	Grafik unter CP/M	SCHMID, A.	25/35
	CP/M-Warmstart ohne Nachladen	SCHOBER	26/25
	CMD ==> COM ohne CPMac	SCHROEDER	21/23
	CP/M intern - kurz und kompakt	SCHROEDER	23/49
	Wie baue ich mir ein CP/M ?	SCHROEDER	21/25
	Fensterln mit CP/M	SCHROEDER, G.	28/64
	Backen und Banken	SOERENSEN	28/44
	ZCPR33 - die Vierte	SOERENSEN	28/53
	Z-CP/M - eine deutliche Verbesserung	SOERENSEN, R.	25/37
	ZCPR33 - Installation auf M4...	SOERENSEN, R.	25/45
	CP/MAC - NA UND?	SOPP	19/78
CPU	Z80-Hochintegrations-CPUs	BETZ	23/88
DATEI	INDEX-DATEIEN	HERMANN	19/15
DATEN	VIDEODAT 300	?	13/51
	DATENVERLUSTE	aus c't	19/80
DATENBANK	Dateiprogramm	DRECHSLER	28/22
	BDBS - BASIC DATENBANK SYSTEM (MODEL 4)	HERMANN	22/37
	dBASE II	MAND	23/65
	dBASE II - eine Einfuehrung	MAND	21/33
	Teamwork	SCHMID, A.	28/37
	Datenbanken richtig aufgebaut	aus WINDOW 9'88	26/73
DBASE	dBASE II, III und III+ Ueberblick	aus DOS 5'88	26/67
DDE	DDE fuer das GENIE IIIs	SOPP	12/17
DEF.FUNCTION	BASIC-PROGRAMMIERBARE FUNKTIONEN	HERMANN	19/26
DENKAUFGABEN	MATHEMATIK fuer FEINSCHMECKER	WEBER	20/101
DFUe	DATENUEBERTRAGUNG p.FERNSEHBILD/WDR-CLUB	EDDE u.HERIBERT	12/62
	Datenuebertragung / M4p als Terminal	HERMANN	23/30
DISASSEMBLER	Kleine Hexen bevorzugt	SOPP	23/27
	Kleine Hexen ohne Aufwand	SOPP	25/17
	Kurzanleitung fuer SCREENER/CMD	SOPP	25/15
	Screening mit DSMBLR/CMD	SOPP	25/14
DOS	Moeglichkeiten und Grenzen des MS-DOS	HEIDENREICH	27/19
	R-BEFEHL MSDOS-LIKE (ein Editor f. DOS)	HEIDENREICH	20/44
	LS-DOS 6.3 - KOPIERSCHUTZ	HERMANN	21/55
	Der Teufel steckt oft im Detail	OBERMANN	25/30
	EXTENDED NEWDOS 80	OBERMANN	10/5
	NEWDOS/80+10 VERSION 2.4 (26.10.85)	OBERMANN	11/25
	SUPERDOS	OBERMANN	2 /?
	DIR/SYS ODER INHALT/SYS?	RENSCH	15/21
	Ein kleiner Fehler im Z80-DOS	SCHMID, A.	27/14
	Microcomputer-Betriebssysteme	SCHROEDER, G.	28/103
	FILTER UNTER TRSDOS 6.x	SOERENSEN, R.	18/29
	MDOS im INFO???????	SOERENSEN, R.	20/11
	DOS - (FAST) OHNE FLOPPY	SOPP	14/69
DOS-FEHLER	FEHLER IM NEWDOS-BEFEHL 'BOOT'	RETZLAFF	11/21
DOS-Fehler	Ein Fehler in SYS14/SYS	SOPP	24/19
DOUBLER	SELBSTBAU-DOUBLER fuer EXP1		17/45
DRUCKEN	Druckausgabe f. COBOL-Programme	FOERSTER	23/32
	Druckausgabe mit PRBUF	FOERSTER	24/12
	DREISPALTIG DRUCKEN	[illegible]PP	21/15
	EIN FEHLER IM SYS28/SYS	[illegible]P	22/33

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
DRUCKEN	LPRINT MIT MODERNEN GENIES	SOPP	18/19
DRUCKER	Noch ein Druckerpatch	HELD	23/71
	SPEED WRITING (fuer EPSON MX-DRUCKER)	LINDSEY	12/65
	...WEIL KUERZE SO WUERZIG IST...	MUEHLENBEIN	20/35
	DIE DRUCKERABFRAGE	SCHMITZ	18/12
	VON 8 AUF 24 PINS (= DRUCKNADELN)	SOPP	22/20
DRUCKERGRAF.	GRAPHIK-JKL, TSCRIPS UND DER NEC P6	SOPP	16/27
DRUCKZEICHEN	AM BILDSCHIRM DEFINIERT/EDITOR f.DRUCKER	KLUGE	9/19
	Zeichengenerator fuer Nadeldrucker	aus mc	23/18
ECB-BUS	ECB-ADAPTER-PLATINE	?	15/65
	VERBINDEND	ASSENBAUM	11/39
	Anm. zum CLUB80-ECB-Bus-Projekt	BERNHARDT	23/73
	Gepufferter DMA- u. IM2-faehiger ECB	BERNHARDT	23/74
	Prof 180x und mc-68000-ECB	BERNHARDT	25/57
	DIE EINGANGSPLATINE	DROWAELDER	15/73
	ECB-BUS-GRUNDEINHEITSAUFBAU	NEUEDER	15/71
	ECB-BUS-PROJEKT	NEUEDER	14/101
	ECB-BUS-TRS80-PROJEKT	NEUEDER	13/39
	ECB-BUS fuer MODEL 3/4!???	OBERMANN	15/84
ECHTZEITUHR	ACH DU LIEBE ZEIT!	SOPP	15/32
	REAL TIME BLACK BOX?	SOPP	15/53
ERFAHRUNG	PC STATT GENIE? (Ueber weitere PC's)	SCHMITZ	20/99
ERROR	"PRUEFZAHLFEHLER BEIM LESEN"	BERNHARDT	22/71
FILEVERGLCH	VERGLEICH ZWEIER DISK-DATEIEN	SOPP	22/53
	VERGLEICH ZWEIER PROGRAMME: noch'n TOOL	SOPP	22/57
	Vergleich zweier Disk-Dateien	SOPP	25/20
FLOPPY	FLOPPY-TRICKS	OBERMANN	13/41
FORMATIERUNG	40 TRACKS? WARUM NICHT 42?	OBSCHERNINGKAT	18/11
GENAUIGKEIT	MATHEMATISCHER UEBERFLUSS	MUEHLENBEIN	20/17
GRAFIK	ERGAENZUNG ZU GRAFIK-SHORTY	HERMANN	19/32
	TRS80 M1/3 SCHNELLE GRAFIK IN BASIC	HORNUNG	2 /?
	HRG-BILDUEBERTRAGUNG OHNE HRG (WDR-FILES	OBERMANN	15/11
	SCHNELL BEWEGTE GRAFIK AUF DEM TRS80	ROECKRATH	2 /?
	GRAFIK-STANDARD: ALLEIN GEHT'S AUCH	SCHROEDER	19/11
	EINF.GRAFIKPROGRAMM: PLOT/BAS	SCHROEDER /ARPS	14/29
	LINIENMODELLE IN 3 DIMENSIONEN	SCHROEDER/ ARPS	14/35
	GRAFIK-CLS fuer das GENIE IIIs	SOPP	13/35
	STRINGANZEIGE MIT VOLLDAMPF	SOPP	6/41
GRAFIK-JKL	GRAFIK-JKL fuer das GENIE IIIs	SOPP	13/32
HAENDLER	TANDY-VERTAGSHAENDLER		19/79
	Reingefallen mit Rheintec	SCHMITZ	23/90
	TANDY SCHLIESST COMPUTER-CENTER	ZEITSCHRIFT	18/75
HARDCOPY	HARDCOPY AUCH OHNE INTERRUPT	RYCHLIK	20/15
	FLOPCOPY AUCH OHNE INTERRUPT	SOPP	21/14
	TRS80 erstellt HARDCOPY	aus mc	23/17
HRG	HRG vom BASIC aus (DEMO)	NEUEDER	10/31
	HRG-PIXELS aus dem BASIC!	NEUEDER	10/23
	HRG: 384x192 oder 192x192	NEUEDER	13/14
	HRG-BILDUEBERTRAGUNG per DFUe	OBERMANN	14/24
	HRG-Bilduebertragung	OBERMANN	25/31
	UMKEHR!	OBERMANN	20/58
	GENIE IIIs, DIE HRG DES -	SOPP	12/21
	HRG ib PROGRAMMIEREN, BESSER ERKLAERT	SOPP	10/25
	HRG UND BASIC NETTO	SOPP	10/27
	HRG, dasz die Schwarte kracht !	SOPP	24/40
	HRG-HARDCOPY fuer das G IIIs	SOPP	13/25
	HRG-SPEICHER LOESCHEN, DEN -	SOPP	10/29
	Traumlandschaften aus dem Computer	aus Comp.Pers.	27/41
	Wie funktioniert ein Vektorgenerator	aus Comp.Pers.	27/31
INTERRUPT	Unterbrechung gestattet ?	SOERENSEN	28/52
JKL	JKL - AB[illegible] DRUCKERSCHONEND!	SOPP	12/13
KASSETTE	CASSETTE[illegible]ZEICHNIS ERSTELLEN	WAGNER, G.	10/33

PAGE 4 - BEITRAEGE ZUM THEMA SOFTWARE; STAND: 31.12.89

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
KOPIERSCHUTZ	LS-DOS 6.3 - KOPIERSCHUTZ	HERMANN	21/55
LAUFWERK	LWT fuer GENIE I und TRS-80 MODELL I	RETZLAFF	13/40
	Ordnung musz sein !	SOPP	28/33
LIBRARY	WAS TUN BEI VOLLER LIBRARY?	SOPP	12/15
LITERATUR	BUECHERLISTE: DIE ERSTEN KURZKRITIKEN	SCHROEDER	21/67
	WEITERE KURZKRITIKEN	VERSCH.	22/DIV
LOGIK	BOOLE XOR ABEL	MUEHLENBEIN	17/16
	STENO-LOGIK	MUEHLENBEIN	16/24
	LOGISCHE OPERATIONEN 2	OBERMANN	16/9
	LOGISCHE OPERATIONEN IN BASIC	OBERMANN	15/9
	LOGISCHES ZU LOGIK	SOPP	16/14
	LOGIKSIMULATION mit BASIC	ZEITSCHRIFT	16/11
M4/4p	AUFRUF DER SUPERVISOR-CALLS in BASIC	HERMANN	14/62
	Atari-Bilder auf dem 4p	OBERMANN	26/21
	DAS BOOT-ROM des 4p	OBERMANN	14/63
	HRG fuer MODEL 3/4/4p	OBERMANN	14/67
	Das neueste DOS fuer M4/4p	OBSCHERNINGKAT	24/29
	AUFGESTIEGEN?	WAGNER, G.	14/59
MATH.STATIST	BIVARANA (2fache Varianzanalyse)	MUEHLENBEIN	19/21
MATHE-SPIEL	WAS IST MIT DER PRIMZAHL '37' LOS?	MUEHLENBEIN	11/23
MATHE/GRAFIK	RECHNEN (ZEICHNEN) MIT DEM COMPUTER...	SCHROEDER	10/53
MATHEMATIK	WIE der NAT.LOG.die RECHENZEIT VERKUERZT	GUAGLIANO	14/77
	PRIMZAHLBERECHNUNGEN (3 PROGRAMME)	KASPER	5 / ?
	SPLINE-INTERPOLATION (KUBISCH)	KASPER	16/19
	Ableitg.d.Negationsregel f.Binaerzahlen	MUEHLENBEIN	25/09
	CRAMER UND DIE DETERMINANTEN	MUEHLENBEIN	19/13
	DAS CARTESISCHE BLATT	MUEHLENBEIN	18/5
	DER SCHNELLSTE ZAHNARZT (WURZELZIEHEN)	MUEHLENBEIN	22/11
	Ein letzter 10Zeiler (Determinantenber.)	MUEHLENBEIN	28/10
	GOLDENER FIBONACCI	MUEHLENBEIN	16/21
	PASCAL & GALTON	MUEHLENBEIN	28/17
	Polynomberechnung	MUEHLENBEIN	24/13
	Sitzverteilung nach d'Hondt	MUEHLENBEIN	28/11
	UMWANDLUNG: ROEMISCHE IN ARABISCHE ZAHL	MUEHLENBEIN	16/7
	Fraktale Geometrie	SCHROEER	27/15
	Bin.-Umwandlg.HEX/DEZ;kuerzere E-Routine	SOPP	24/23
	Binaerumwandlung, letzter (?) Teil	SOPP	25/12
	Was man weisz - schon immer wissen wollt	WOLF, Thomas	26/15
MAUS	Die Pratze laeszt das Mausen nicht	SOPP	25/18
MEMDISK	RESET-FESTE MEMDISK M4-CP/M	OBERMANN	19/43
MODEL 4	BOOT-ROM-VERSION und DATUM M4	HERMANN	19/42
	TURBO TIP	OBERMANN	21/31
	KLEINE ZIPPERLEIN RASCH KURIERT	WOLLSCHLAEGER	12/57
MONITOR	MONITOR IM GENIE	KASPER	6/23
MUSIK	MEIN ZWEITES ASSEMBLERPROGRAMM	MUEHLENBEIN	15/15
	SCHWERE KAVALLERIE ("DAS CLUBLIED")	MUEHLENBEIN	22/17
PASCAL	APFELMAENNCHEN auf der GDP64	BERNHARDT	21/37
	GENIE, CP/M und 80 ZEICHEN und HRG ???	HELD	21/43
	PASCAL-PROBLEME	KONRAD, J.	9/73
	Vergleich BASIC mit TURBO-PASCAL	MUEHLENBEIN	28/39
	TURBO PATCH	OBERMANN	21/32
	TURBO PATCH (FEHLERKORREKTUR)	OBERMANN	21/32
	TURBO TIP	OBERMANN	21/31
	TURBO-TIP	OBERMANN	21/31
	Vergleichstest (versch.PASCAL-Versionen)	OBERMANN	23/33
	KURZKRITIKEN und BUECHERLISTE	SCHROEDER	21/71
	PASCAL-VORSTELLUNG TEIL 2	SOERENSEN	22/40
	PASCAL fuer INTERESSIERTE	SOERENSEN, R.	20/12
	Echte Programme meiden PASCAL	aus DATAMON7/83	23/91
PASSWORT	PASSWORT-BYPASS BEI TRSDOS 6.2	OBERMANN	19/42
PATCH	GROSZBUCHSTABENMODUS AB SYSTEMSTART (M4)	HERMANN	21/30
PDRIVE	Wer sucht, der soll finden	MUEHLENBEIN	23/19

PAGE 5 - BEITRAEGE ZUM THEMA SOFTWARE; STAND: 31.12.89

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
PDRIVE	PDRIVE-IDENTIFIKATION fuer NEWDOS 80	OBERMANN	10/17
PLOTTER	KREIS OHNE TRIGONOMETRIE	SCHROEDER	15/8
POKEPEEK	POKE+PEEK-ECKE	HANS -?-	4 / ?
PORT	PORTBELEGUNG von COMPUTERKONFIGURATIONEN	KUHN	21/44
	OUT,PORT#,xx,yy,...	SOPP	11/29
	VERBESSERT: IO,PORT#(,xx,yy,..)	SOPP	12/11
PROGRAMMIERG	STRUKTURIERTE PROGRAMMIERUNG	?	7/17
	KEINE MEINUNG ZUR STRUKTUR?	GRELL	7/28
	EDIT-BEFEHLE	HERMANN	14/6
	BASIC- LAUFENDE PROGRAMME ERWEITERN	KRENZKE	16/18
	LOKALE VARIABLE BEIM TRS 80	PAULO	7/7
	PLANVOLL PROGRAMMIEREN	SACHSE	7/23
	DIRECTORY LESEN	WAGNER, G.	9/4
	EDITOR	WAGNER, G.	9/5
	PROGRAMMIEREN HEISST m. DATEIEN ARBEITEN	WOLLSCHLAEGER	7/29
	TRICKS MIT STRINGS	WOLLSCHLAEGER	6/65
	DEM INTERPRETER aufs BIT GESCHAUT	ZEITSCHRIFT	12/37
	DIE RICHTIGE DIMENSION	Zeitschrift	18/26
PROZESSOR	Z80, HD64180 UND ILLEGALS	SCHROEDER	17/27
	TAV's Z80-TUNING	ZEITSCHRIFT	17/33
RAM	512K-RAM-ADRESSEN m. FLEXIBLER MMU	BERNHARDT	18/67
RAMDISK	Die RAMDISK und das 11.Gebot	SCHMID, A.	25/56
	MINI-RAM-FLOPPY im "SICHEREN PLAETZCHEN"	SOPP	10/37
RECHT	COMPUTERKRIMINALITAET		19/81
	URHEBERRECHTSSCHUTZ fuer PROGRAMME	KAHLEN	10/55
RECORDORGAN.	ZU GERALDs ARTIKEL ueber LMOFFSET	SOPP	10/3
RESTORE	BASIC-PATCH FUER RESTORE N	OBERMANN	5 / ?
ROM-ROUTINEN	CALLERLEI	MUEHLENBEIN	20/27
SCHALTALGEBR	KLEINES LOGIK-LEXIKON	aus c't	19/61
SEMANTIK	TRAU - SCHAU, WEM! Ein Bitchen Englisch	MUEHLENBEIN	18/47
SICHERHEIT	RAM-DISK KOSTENLOS	WOLLSCHLAEGER	2 / ?
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES	SOPP	13/23
	SONDERTASTEN DIVERSER GENIES (NACHTRAG)	SOPP	14/53
SORTIEREN	IN VISICALC SORTIEREN	MUEHLENBEIN	20/93
	Ordnung musz sein !	MUEHLENBEIN	28/13
	TRS80: Sortieren	aus mc	23/12
SPEICHER	CALL UM DIE ECKE (MEM-BEREICHE ERREICHEN	SOPP	15/49
SPIEL	Lottozahlen	MUELLER, Kurt	26/19
	JOYSTICK AN MODEL 4/4p	OBERMANN	19/50
	TAPE-GAMES AUF MODEL III/4/4p	OBERMANN	20/57
	ADVEN-80	SCARGILL	11/9
	DAS PIRATE ADVENTURE	SCHROEDER	11/15
	ADVENTURE GEHEIMAGENT XP-05	WAGNER, ALEX.	3 / ?
SPRACHE	COBOL-EINFUEHRUNG		8/5
	DER-DIE-DAS MACHT MIR SPASZ	MUEHLENBEIN	19/75
	Welche Sprache fuer wen ?	SCHMID, A.	27/59
	COBOL - EINE EINFUEHRUNG	TRAPP	5 / ?
SPURANZAHL	80/40-TRACKUMSCHALTUNG	KUHN	19/49
SYSTEM	Quatsch in DVL4/SYS	SOPP	24/27
	Schon wieder SYS14/SYS	SOPP	24/20
TAKT	TAKTUMSCHALTUNG - DIE ALLERLETZTE LOESG.	MUELLER, KURT	16/45
TAPE	SUPERTAPE FUER TRS 80	BURGWITZ/STILLR	7/44
TASMON	ERWEITERUNG FUER TASMON (NEWDOS-BEFEHLE)	KONRAD J.	6/25
TASTATUR	UMLAUTE IM NEWDOS? ABER NATUERLICH!	NEUEDEER	10/41
	KNOEPFE	OBERMANN	14/57
	ES GEHT NOCH SCHNELLER (TAST.ABSCHALTEN)	SCHMID, A.	22/32
	UNIVERSALGENIE	aus c't	19/57
TELEFON	TELEFONBUCH MIT PFIFF	FRISCH	2 / ?
TEXT-GRAFIK	(TELEFON) - EIN UD-ZEICHEN auf d.DRUCKER	SOPP	16/29
TEXTSYSTEM	UMDEFINIERUNG v.SCRIPSIT-KOMMANDOTASTEN	KONRAD	8/14
	KEINE ANGST VOR... (TRENNEN IN TSCRIPS)	MUEHLENBEIN	20/61
	Save Our Script !	MUEHLENBEIN	24/15

INHALTSÜBERSICHT CLUB-INFOS S3

INHALTSÜBERSICHT CLUB-INFOS 54

PAGE 6 - BEITRAEGE ZUM THEMA SOFTWARE; STAND: 31.12.89

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
TEXTSYSTEM	TSCRIPS retten	MUEHLENBEIN	28/38
	Was TSCRIPS nicht kann	MUEHLENBEIN	23/20
	DUMP (TSCRIPS retten)	MUELLER, Kurt	26/18
	SCRIPSIT AUF DEM ALTEN GENIE	OBERMANN	14/43
	Wie sag' ich's meinem WORDSTAR ?	SCHMID, A.	24/11
	WORDSTAR auf allen Laufwerken	SCHOBER	28/35
	AENDERUNGEN IN TSCRIPS und NAME/CMD	SCHROEDER	13/16
	FREIRAEUME IN TSCRIPS	SCHROEDER	14/47
	SCRIPSIT-TABELLEN	SCHROEDER	14/45
	TEXTUMLEITUNG fuer TSCRIPS	SCHROEDER	14/49
	TSCRIPS-VERSCHLUESSELUNG	SCHROEDER	15/29
	§ CONTRA	SOPP	19/27
	KEINE ANGST VOR PROPORTIONALSCHRIFT	SOPP	18/13
	NEUES ZUR PROPORTIONALSCHRIFT	SOPP	18/15
	TIPS fuer SCRIPS	SOPP	13/19
	TSCRIPS & DIE &-CODES	SOPP	19/16
	TSCRIPS UND DIE 24 NADELN	SOPP	22/23
	VON DER WIEGE BIS ZUR BAHRE (ADR.AUTOMAT	SOPP	19/33
	DAS TSCRIPS-DRAMA - EIN LEHRSTUECK	SOPP/MUHLENBEIN	21/56
	KLEINES TEXTPROGRAMM	aus mc	23/10
TONERZEUGUNG	DA CAPO... (WIE MAN TOENE ERZEUGT)	MUEHLENBEIN	22/13
	TONGENERATOR FUER TRS 80	PLUMHOFF	8/19
TRS 80	DER BEKNACKTE TRS 80	ROECKRATH	6/33
UHR	UHRUNTEN fuer MODEL 4p	OBERMANN	15/47
	Es war einmal ein Datum	SCHMID, A.	26/29
	THE TIMES, THEY ARE A-CHANGING	SOPP	22/35
	WENN DIE UHR MAL STOERT	SOPP	14/39
UTILITY	EINGABEROUTINE MIT KOMFORT	BULLING	7/13
	Startprogramm f. BASIC-Programme	DRECHSLER	28/21
	SHELL/CMD	HERERMANN	24/34
	TRACE-ON	HOEH (COMP.PERS	4 /?
	AUSGABEUMSCHALTUNG IN BASIC	KONRAD J.	6/24
	ALLERLEI NUETZLICHES FUER DEN TRS 80	LICHTERMANN /MC	4 /?
	DIRECTORY LESEN UND KOMMENTARE SPEICHERN	MUEHLENBEIN	13/15
	FILES WIE SAND AM MEER...	MUEHLENBEIN	11/7
	FILEVERGLEICH (s.auch Progr.-Bibliothek)	NEUEDER	10/43
	DDE & LWT	OBERMANN	10/19
	LPRINT ALLES einfacher	OBERMANN	8/3
	ZAP IM SUPERZAP/Zappen im Zeichenbereich	OBERMANN	8/13
	LPRINT ALLES	SOPP	6/31
	BILDSCHIRMAUSDRUCK BEIM TRS80+VIDEOGENIE	UEBERSCHAAR	5 /?
	SMARTSCREEN, SMARTKEY und SMARTPRINT	ZEITSCHRIFT	12/45
	PAGELIST/CMD	aus Zeitschrift	18/21
VERTRETUNGEN	TRACE od.DIE GESCHICHTE EINER ZEITSCHR.	OBSCHERNINGKAT	18/75
VIDEO	VIDEODAT-SYSTEM		18/57
VISICALC	Korrigieren und Steuern innerhalb VC	MUEHLENBEIN	23/24
	SORTICALC	MUEHLENBEIN	26/11
Z80	ARBEITSTIER MIT GESUNDEM HERZEN	BETZ	21/52
	422 NEUE Z80-BEFEHLE	HASSELBERG (MC)	4 /?
ZAPPEN	EIN BYTE UND SEINE FOLGEN	OBERMANN	10/21
ZEICHENSATZ	Geheimnisse eines Zeichengenerators	BIESE	28/27
	NEUE ZEICHENSAETZE fuer das GENIE IIIs	SOPP	12/28
ZEITMESSUNG	ETWAS fuer LEUTE, die NIE ZEIT haben	SCHMID, A.	22/31

PAGE 1 - BEITRAEGE ZUM THEMA HARDWARE; STAND: 31.12.89

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
ADAPTER	Testadapter	REGGE	9/45
AUFRUESTEN	Modernisierung statt Neukauf	MATTHAEI/H.COMP	5/ ?
BILDSCHIRM	Bessere Darst.v.Text u.HRG auf d.Bildsch	BERNHARDT	26/37
BUS	PROF 180x und mc68000-ECB	BERNHARDT	25/57
	ECB-BUS-Projekt	DROWAELD.&OBERM	15/65
	" "	OBERMANN	9/42
	Freiheit fuer den BUS	SCHROEDER, G.	25/55
	Indikatorkarte fuer ECB-BUS	SPIESS	19/45
	Analog / Digital - Umsetzer	STOBER	24/37
COMPUTER	Der fluesternde Portable	BETZ	28/67
	Modell 4	WOLLSCHLAEGER	12/69
	SHARP 3541	aus Comp.Pers.	26/47
CP/M	Billiges CP/M	BREWER	9/53
DECODER	Eleganter Adressdecoder fuer GENIE I	BERNHARDT	28/73
DFUe	Datenuebertragung per Fernsehbild	Edde & Heribert	12/62
	Bewaehrtes DFUe-Protokoll	aus Comp.Pers.	27/51
DISK	Sicherer Platz fuer Daten? (Disketten)	aus test 9/87	22/73
DRUCKER	STAR NL 10 abwechselnd am CP/M-R. + PC	BERNHARDT	28/72
	Speed Writing (erhoeht Druckgeschwind.)	LINDSEY	12/65
	Funktionsweise des Drucker-Puffers	TRAPP	6/18
FREQUENZ	Taktfrequenz-Verdopplung GENIE I/II	DREYER	5/ ?
	Taktumschaltung	MUELLER, Kurt	16/45
	Taktumschaltung fuer VIDEO-GENIE	OBERMANN	8/25
	Synchronisierte Taktumschaltung	RETZLAFF	10/49
GRAFIK	Auslesen des Grafikspreichers	BERNH./SEELMANN	20/65
	Es lebe der Standard!	BERNHARDT	28/78
GRUNDLAGEN	Digitaltechnik, Grundlagen der -	gesammelt: KUHN	SONDER
HRG	HRG-1b-Karte	MUELLER, KURT	14/89
IC	Hardware-Tips: Neue IC's	ZEITSCHRIFT	17/38
INTERFACE	Relais-Interface	?	9/47
	Interface-Adapter 8255, Programmierbarer	KUHN	13/47
JOYSTICK	Joysticks fuer den TRS-80 M1	NEUEDER	5/ ?
	Joystick-Anschluss	SOPP	6/29
LAUFWERK	EXP1-Floppy-Controller	BERNHARDT	26/38
	Floppydisk-Controllerkarte	DROWAELDER	18/49
	80/40-Trackumschaltung Mitsubishi	OBERMANN	26/39
	Floppy-Tricks	OBERMANN	14/87
	Ein klebriges Problem	SCHMID, A.	28/71
MAUS	Die Maus am TRS-80 !	OBERMANN	22/60
	Maus mit sieben Beinen	SOPP	23/69
MODEM	Akustikkoppler	TRAPP	4/ ?
PORT	Portbelegung	KUHN	21/44
PROMMER	EPROM-Programmiergeraet	WOLF (aus mc)	5/ ?
RAM	EXTRA RAM ("Ein sicheres Plaetzchen...")	OBERMANN	9/63
RAMDISK	Darf's ein bisserl mehr sein?	SCHMID, A.	26/46
	Die RAMDISK und das elfte Gebot	SCHMID, A.	25/56
SCHALTSTATUS	WACHHUND, Der-, b.unkontr.Schaltzustand	BETZ	20/63
SCHNITTSTELL	Entfernung der Kassettenschnittstelle	SCHROEDER, G.	26/40
	Memory mapped I/O <=> Port I/O bei GIIs	SCHROEDER, G.	26/42
	Die Parallelschnittstelle	aus Comp.Pers.	27/49
SENSOR	Digitales Temperatur-Zeit-Uhrenmodul	Obermann	16/41
SPEICHER	Sicheres, Ein -, Plaetzchen	BERNHARDT	7/43
TASTATUR	Tastatur- und Drucker-Platine	DROWAELDER	20/71
TRS-80	TRS-80 Japan-Modell aufbohren	BERNHARDT	25/60
	TRS-80, Der - und die Aussenwelt	KLEIN, WILFRIED	4/ ?
	Umbau eines TRS-80 Mod.1 16K auf 48K	RETZLAFF	10/45
UMLAUTE	Umlaute auf der Tastatur des GENIE I	DRECHSLER	28/69
VIDEO	Das 80-Zeichen-Karte-Problem u.d.Loesung	BERNHARDT	22/63
	80-Zeichen-Karte	SCHROEDER, G.	12/63
Z80	INTERFACE mit Kassetten-Schnittstelle	SCHROEDER, G.	26/40

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
BILDSCHIRM	BILDSCHIRMAUSGABE mit INVERSEN ZEICHEN	?	12	79
ECB-BUS	ECB-ADAPTER-PLATINE	?	15	65
INTERFACE	Relais-Interface	?	9	47
PROGRAMMIERG	STRUKTURIERTE PROGRAMMIERUNG	?	7	17
CODE	BASICODE-INTERFACE	?(aus ELEKTOR)	3	?
CODE	BASICODE–2	?(aus Elektor)	3	?
ECB-BUS	VERBINDEND	ASSENBAUM	11	39
CHIP	NANOHENRY BLOCKIEREN DEN FORTSCHRITT	aus c' t	19	70
TASTATUR	UNIVERSALGENIE	aus c’t	19	57
COMPUTER	SHARP 3541	aus Comp.Pers.	26	47
DFUe	Bewaehrtes DFUe-Protokoll	aus Comp.Pers.	27	51
HRG	Traumlandschaften aus dem Computer	aus Comp.Pers.	27	41
HRG	Wie funktioniert ein Vektorgenerator	aus Comp.Pers.	27	31
SCHNITTSTELL	Die Parallelschnittstelle	aus Comp.Pers.	27	49
CHIP	DER LEIDENSWEG DER CMOS-CHIPS	aus c't	19	67
SCHALTALGEBR	KLEINES LOGIK-LEXIKON	aus c't	19	41
PASCAL	Echte Progranne neiden PASCAL	aus DATAMON7/83	23	91
DBASE	dBASE II, III und III+ Ueberblick	aus DOS 5'88	26	67
SORTIEREN	TRS80: Sortieren	aus mc	23	12
TEXTSYSTEM	KLEINES TEXTPROGRAMM	aus mc	23	10
DISK	Sicherer Platz fuer Daten? (Disketten)	aus test 9/87	22	73
CHIP	ERSTER MEGABIT-CHIP IN SERIE	aus Zeitschrift	18	18
UTILITY	PAGELIST/CMD	aus Zeitschrift	1B	21
DRUCKZEICHEN	Zeichengenerator fuer Nadeldrücker	aut nc	23	18
HARDCOPY	TRS80 erstellt HARDCOPY	aut nc	23	17
GRAFIK	Auslesen des Grafikspreichers	BERNH./SEELMANN	20	65
BANKING	256K-RAM fuer Z80-SYSTEME	BERNHARDT	14	91
BILDSCHIRM	Bessere Darst.v.Text u.HRG auf d.Bildsch	BERNHARDT	26	37
BUS	PROF 180x und mc68000-ECB	BERNHARDT	25	57
DECODER	Eleganter Adressdecoder fuer GENIE I	BERNHARDT	28	73
DRUCKER	STAR NL 10 abwechselnd am CP/M-R. + PC	BERNHARDT	28	72
ECB-BUS	Anm. zum CLUB80-ECB-Bus-Projekt	BERNHARDT	23	73
ECB-BUS	Gepufferter DMA- u. IM2-faehiger ECB	BERNHARDT	23	74
ECB-BUS	Prof 180x und mc-68OOO-ECB	BERNHARDT	25	57
ERROR	"PRUEFZAHLFEHLER BEIM LESEN"	BERNHARDT	22	71
GRAFIK	Es lebe der Standard!	BERNHARDT	28	78
LAUFWERK	EXP1Floppy-Controller	BERNHARDT	26	38
PASCAL	APFELMAENNCHEN auf der GDP64	BERNHARDT	21	37
RAM	512K-RAM-ADRESSEN m. FLEXIBLER MMU	BERNHARDT	18	67
SPEICHER	Sicheres, Ein -, Plaetzchen	BERNHARDT	7	43
TRS-80	TRS-80 Japan-Modell aufbohren	BERNHARDT	25	60
VIDEO	Das 80-Zeichen-Karte-Problem u.d.Loesung	BERNHARDT	22	63
COMPUTER	Der fluesternde Portable	BETZ	28	67
SCHALTSTATUS	WACHHUND, Der-, b.unkontr.Schaltzustand	BETZ	20	63
Z80	ARBEITSTIER MIT GESUNDEM HERZEN	BETZ	21	52
BILDSCHIRM	BIN ICH STRAHLENGEFAEHRDET?	BOIKAT, Ute	20	109
CP/M	Billiges CP/M	BREWER	9	53
UTILITY	EINGABEROUTINE MIT KOMFORT	BULLING	7	13

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
TAPE	SUPERTAPE FUER TRS 80	BURGWITZ/STILLR	7	44
UMLAUTE	Umlaute auf der Tastatur des GENIE I	DRECHSLER	28	69
UTILITY	Startprogramm f. BASIC-Programme	DRECHSLER	28	21
FREQUENZ	Taktfrequenz-Verdopplung GENIE I/II	DREYER	5	?
BUS	ECB-BUS-Projekt	DROWAELD. &OBERM	15	65
ECB-BUS	DIE EINGANGSPLATINE	DROWAELDER	15	73
LAUFWERK	Floppydisk-Controllerkarte	DROWAELDER	18	49
TASTATUR	Tastatur- und Drucker-Platine	DROWAELDER	20	71
DFUe	Datenuebertragung per Fernsehbild	Edde & Heribert	12	62
DFUe	DATENUEBERTRAGUNG p.FERNSEHBILD/WDR-CLUB	EDDE u.HERIBERT	12	62
TELEFON	TELEFONBUCH MIT PFIFF	FRISCH	2	?
GRUNDLAGEN	Digitaltechnik, Grundlagen der -	gesammelt: KUHN		
ZEICHENSATZ	Geheimnisse eines Zeichengenerators	GIESE	28	27
PROGRAMMIERG	KEINE MEINUNG ZUR STRUKTUR?	GRELL	7	28
MATHEMATIK	WIE der NAT.LOG.die RECHENZEIT VERKUERZT	GUAGLIANO	14	77
POKEPEEK	POKE+PEEK-ECKE	HANS -?-	4	?
AMATEURFUNK	Die Betriebsart PACKET-Radio (PR)	HARTMANN, H.-G.	28	111
Z81	422 NEUE Z80-BEFEHLE	HASSELBERG (MC)	4	?
ASSEMBLER	KLEINSCHRIFT fuer den EDTASM+	HEIDENREICH	20	37
BILDFORMAT	FUTTER fuer GENIE IIIs CRTC	HEIDENREICH	20	22
DOS	R-BEFEHL MSDOS-LIKE (ein Editor f. DOS)	HEIDENREICH	20	44
DOS	Moeglichkeiten und Grenzen des MS-DOS	HEIDENREICH	27	19
DRUCKER	Noch ein Druckerpatch	HELD	23	71
PASCAL	GENIE, CP/M und 80 ZEICHEN und HRG ???	HELD	21	43
UTILITY	SHELL/CMD	HERERMANN	24	34
GRAFIK	ERGAENZUNG ZU GRAFIK-SHORTY	HERMANN	19	32
KOPIERSCHUTZ	6.3 - KOPIERSCHUTZ	HERMANN	21	55
M4/4p	AUFRUF DER SUPERVISOR-CALLS in BASIC	HERMANN	14	62
MODEL 4	BOOT-ROM-VERSION und DATUM M4	HERMANN	19	42
PATCH	GROSZBUCHSTABENMODUS AB SYSTEMSTART (M4)	HERMANN	21	30
PROGRAMMIERG	EDIT-BEFEHLE	HERMANN	14	4
UTILITY	TRACE-ON	HOEH (COMP.PERS	4	?
GRAFIK	TRS80 Ml/3 SCHNELLE GRAFIK IN BASIC	HORNUNG	2	?
RECHT	URHEBERRECHTSSCHUTZ fuer PROGRAMME	KAHLEN	10	55
MATHEMATIK	SPLINE-INTERPOLATION (KUBISCH)	KASPER	16	19
MATHEMATIK	PRIMZAHLBERECHNUNGEN (3 PROGRAMME)	KASPER	5	?
MONITOR	MONITOR IM GENIE	KASPER	6	23
TRS-81	TRS-80, Der - und die Aussenwelt	KLEIN, WILFRIED	4	?
DRUCKZEICHEN	AM BILDSCHIRM DEFINIERT/EDITOR f.DRUCKER	KLUGE	9	19
TEXTSYSTEM	UMDEFINIERUNG v.SCRIPSIT-KOMMANDOTASTEN	KONRAD	8	14
ASSEMBLER	SCHUETZEN VON ASSEMBLERPROGRAMMEN	KONRAD J.	5	?
TASMON	ERWEITERUNG FUER TASMON (NEWDOS-BEFEHLE)	KONRAD J.	4	25
UTILITY	AUSGABEUMSCHALTUNG IN BASIC	KONRAD J.	6	24
ASSEMBLER	HALBAUTOMATISCHE FESTLEGUNG VON ORG	KONRAD, J.	9	3
PASCAL	PASCAL-PROBLEME	KONRAD, J.	9	73
PROGRAMMIERG	BASIC- LAUFENDE PROGRAMME ERHEITERN	KRENZKE	16	18
INTERFACE	Interface-Adapter 8255, Programmierbarer	KUHN	13	47
PORT	PORTBELEGUNG von COMPUTERKONFIGURATIONEN	KUHN	21	44

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
PORT	Portbelegung	KUHN	21	44
SPURANZAHL	80/4O-TRACKUMSCHALTUNG	KUHN	19	49
UTILITY	ALLERLEI NUETZLICHES FUER DEN TRS 80	LICHTERMANN /MC	4	?
DRUCKER	SPEED WRITING (fuer EPSON MX-DRUCKER)	LINDSEY	12	65
DRUCKER	Speed Writing (erhoeht Druckgeschwind.)	LINDSEY	12	65
BERICHT	Testbericht KX 1083	MAND	23	89
AUFRUESTEN	Modernisierung statt Neukauf	MATTHAEI/H.COMP	5	?
ASCII	Wogegen selbst ZEUS vergeblich kaempft	MUEHLENBEIN	25	5
ASSEMBLER	MEIN ERSTES ASSEMBLERPROGRAMM	MUEHLENBEIN	14	9
ASSEMBLER	MEIN DRITTES ASSEMBLERPROGRAMM	MUEHLENBEIN	16	33
BANKING	Bangen um Banken	MUEHLENBEIN	25	7
COMPILER	BASCOM-KURZANLEITUNG	MUEHLENBEIN	13	38
DRUCKER	...WEIL KUERZE SO WUERZIG IST...	MUEHLENBEIN	20	35
GENAUIGKEIT	MATHEMATISCHER UEBERFLUSS	MUEHLENBEIN	20	17
LOGIK	STENO-LOGIK	MUEHLENBEIN	16	24
LOGIK	BOOLE XOR ABEL	MUEHLENBEIN	17	16
MATH.STATIST	BIVARANA (2fache Varianzanalyse)	MUEHLENBEIN	19	21
MATHEMATIK	GOLDENER FIBONACCI	MUEHLENBEIN	16	21
MATHEMATIK	UMWANDLUNG! ROEMISCHE IN ARABISCHE ZAHL	MUEHLENBEIN	16	7
MATHEMATIK	DAS CARTESISCHE BLATT	MUEHLENBEIN	18	5
MATHEMATIK	CRAMER UND DIE DETERMINANTEN	MUEHLENBEIN	19	13
MATHEMATIK	DER SCHNELLSTE ZAHNARZT (WURZELZIEHEN)	MUEHLENBEIN	22	11
MATHEMATIK	Polynonberechnung	MUEHLENBEIN	24	13
MATHEMATIK	Ableitg.d.Negationsregel f.Binaerzahlen	MUEHLENBEIN	25	9
MATHEMATIK	Ein letzter 10Zeiler (Determinantenber.)	MUEHLENBEIN	28	10
MATHEMATIK	PASCAL St GALTON	MUEHLENBEIN	28	17
MATHEMATIK	Sitzverteilung nach d'Hondt	MUEHLENBEIN	2B	11
MATHE-SPIEL	WAS IST MIT DER PRIMZAHL '37' LOS?	MUEHLENBEIN	11	23
MUSIK	MEIN ZWEITES ASSEMBLERPROGRAMM	MUEHLENBEIN	15	15
MUSIK	SCHWERE KAVALLERIE ("DAS CLUBLIED")	MUEHLENBEIN	22	17
PASCAL	Vergleich BASIC nit TURBO-PASCAL	MUEHLENBEIN	28	39
ROM-ROUTINEN	CALLERLEI	MUEHLENBEIN	20	27
SEMANTIK	TRAU - SCHAU, WEM! Ein Bitchen Englisch	MUEHLENBEIN	18	47
SORTIEREN	IN VISICALC SORTIEREN	MUEHLENBEIN	20	93
SORTIEREN	Ordnung musz sein !	MUEHLENBEIN	28	13
SPRACHE	DER-DIE-DAS MACHT MIR SPASZ	MUEHLENBEIN	19	75
TEXTSYSTEM	KEINE ANGST VOR... (TRENNEN IN TSCRIPS)	MUEHLENBEIN	20	61
TEXTSYSTEM	Was TSCRIPS nicht kann	MUEHLENBEIN	23	20
TEXTSYSTEM	TSCRIPS retten	MUEHLENBEIN	2B	3B
TONERZEUGUNG	DA CAPO... (WIE MAN TOENE ERZEUGT)	MUEHLENBEIN	22	13
UTILITY	FILES WIE SAND AM MEER...	MUEHLENBEIN	11	7
UTILITY	DIRECTORY LESEN UND KOMMENTARE SPEICHERN	MUEHLENBEIN	13	15
VISICALC	Korrigieren und Steuern innerhalb VC	MUEHLENBEIN	23	24
VISICALC	SORTICALC	MUEHLENBEIN	26	11
BILDSCHIRM	DAS PROJEKT (FERNSEHER –> MONITOR)	MUELLER, K.	18	61
COMPILER	RPNL - DIE ANDERE ART ZU PROGRAMMIEREN	MUELLER, KURT	15	57
COMPILER	RPNL-BESCHREIBUNG	MUELLER, KURT	16	37
COMPILER	RPNL	Mueller, Kurt	26	31

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
FREQUENZ	Taktumschaltung	MUELLER, Kurt	16	45
HRG	HRG-1b-Karte	MUELLER, KURT	14	89
SPIEL	Lottozahlen	MUELLER, Kurt	24	19
TAKT	TAKTUMSCHALTUNG - DIE ALLERLETZTE LOESG.	MUELLER, KURT	14	45
TEXTSYSTEM	DUMP (TSCRIPS retten)	MUELLER, Kurt	26	18
TASTATUR	UMLAUTE IM NEWDOS? ABER NATUERLICH!	NEUEDEER	10	41
ECB-BUS	ECB-BUS-TRS80-PROJEKT	NEUEDER	13	39
ECB-BUS	ECB-BUS-PROJEKT	NEUEDER	14	101
ECB-BUS	ECB-BUS-GRUNDEINHEITSAUFBAU	NEUEDER	15	71
HRG	HRG vom BASIC aus (DEMO)	NEUEDER	10	31
HRG	HRG-PIXELS aus dem BASIC!	NEUEDER	10	23
HRG	HRG: 384x192 oder 192x192	NEUEDER	13	14
JOYSTICK	Joysticks fuer den TRS-80 M1	NEUEDER	5	?
UTILITY	FILEVERGLEICH (s.auch Progr.-Bibliothek)	NEUEDER	10	43
BASIC	EXTENDED BASIC(RESTOREn u. LPRINT alles)	OBERMANN	11	33
BUS	ECB-BUS-Projekt	OBERMANN	9	42
COMPILER	CALL mit BASCOM	OBERMANN	12	33
COMPUTER	UNTERSCHIEDE (zw. MODEL 1 und 3 bzw. 4)	OBERMANN	14	55
COMPUTER	MODEL 4p ROM-IMAGE-LOADER	OBERMANN	15	43
COMPUTER	TANDY<—> SCHNEIDER	OBERMANN	17	49
ECB-BUS	ECB-BUS fuer MODEL 3/4!???	OBERMANN	15	84
FLOPPY	FLOPPY-TRICKS	OBERMANN	13	41
FREQUENZ	Taktumschaltung fuer VIDEO-GENIE	OBERMANN	8	25
GRAFIK	HRG-BILDUEBERTRAGUNG OHNE HRG (WDR-FILES	OBERMANN	15	11
HRG	HRG-BILDUEBERTRAGUNG per DFUe	OBERMANN	14	24
HRG	UMKEHR!	OBERMANN	20	58
HRG	HRG-Bilduebertragung	OBERMANN	25	31
LAUFWERK	Floppy-Tricks	OBERMANN	14	87
LAUFWERK	80/40-Trackumschaltung Mitsubishi	OBERMANN	26	39
LOGIK	LOGISCHE OPERATIONEN IN BASIC	OBERMANN	15	9
LOGIK	LOGISCHE OPERATIONEN 2	OBERMANN	16	9
M4/4p	DAS BOOT-ROM des 4p	OBERMANN	14	63
M4/4p	HRG fuer MODEL 3/4/4p	OBERMANN	14	67
M4/4p	Atari-Bilder auf den 4p	OBERMANN	26	21
MAUS	Die Maus am TRS-80 !	OBERMANN	22	60
MEMDISK	RESET-FESTE MEMDISK M4-CP/M	OBERMANN	19	43
MODEL 5	TURBO TIP	OBERMANN	21	31
PASCAL	TURBO PATCH	OBERMANN	21	32
PASCAL	TURBO PATCH (FEHLERKORREKTUR)	OBERMANN	21	32
PASCAL	TURBO TIP	OBERMANN	21	31
PASCAL	TURBO-TIP	OBERMANN	21	31
PASCAL	Vergleichstest (versch.PASCAL-Versionen)	OBERMANN	23	33
PASSWORT	PASSWORT-BYPASS BEI TRSDOS 6.2	OBERMANN	19	42
PDRIVE	PDRIVE-IDENTIFIKATION fuer NEWDOS 80	OBERMANN	10	17
RAM	EXTRA RAM ("Ein sicheres Plaetzchen...")	OBERMANN	9	63
RESTORE	BASIC-PATCH FUER RESTORE N	OBERMANN	5	?
SENSOR	Digitales Temperatur-Zeit-Uhrenmodul	Obermann	16	41
SPIEL	JOYSTICK AN MODEL 4/4p	OBERMANN	19	50

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
SPIEL	TAPE-GAMES AUF MODEL III/4/4p	OBERMANN	20	57
TASTATUR	KNOEPFE	OBERMANN	14	57
TEXTSYSTEM	SCRIPSIT AUF DEM ALTEN GENIE	OBERMANN	14	43
UHR	UHRUNTEN fuer MODEL 4p	OBERMANN	15	47
UTILITY	DDE St LWT	OBERMANN	10	19
UTILITY	LPRINT ALLES einfacher	OBERMANN	8	3
UTILITY	ZAP IM SUPERZAP/Zappen im Zeichenbereich	OBERMANN	8	13
ZAPPEN	EIN BYTE UND SEINE FOLGEN	OBERMANN	10	21
FORMATIERUNG	40 TRACKS? WARUM NICHT 42?	OBSCHERNINGKAT	18	11
M4/4p	Das neueste DOS fuer M4/4p	OBSCHERNINGKAT	24	29
VERTRETUNGEN	TRACE od.DIE GESCHICHTE EINER ZEITSCHR.	OBSCHERNINGKAT	18	75
PROGRAMMIERG	LOKALE VARIABLE BEIM TRS 80	PAULO	7	7
TONERZEUGUNG	TONGENERATOR FUER TRS 80	PLUMHOFF	B	I9
ADAPTER	Testadapter	REGGE	9	45
FREQUENZ	Synchronisierte Taktumschaltung	RETZLAFF	10	49
LAUFWERK	LWT fuer GENIE I und TRS-80 MODELL I	RETZLAFF	13	40
TRS-82	Umbau eines TRS-80 Mod.I 16K auf 48K	RETZLAFF	10	45
GRAFIK	SCHNELL BEWEGTE GRAFIK AUF DEM TRS80	ROECKRATH	2	?
TRS 80	DER BEKNACKTE TRS 80	ROECKRATH	6	33
HARDCOPY	HARDCOPY AUCH OHNE INTERRUPT	RYCHLIK	20	15
PROGRAMMIERG	PLANVOLL PROGRAMMIEREN	SACHSE	7	23
SPIEL	ADVEN-80	SCARGILL	11	9
BILDSCHIRM	NOCH EIN BILDSCHIRMSCHONER	SCHMID, A.	22	26
BILDSCHIRM	SCHWIMMENDE BILDER	SCHMID, A.	22	66
LAUFWERK	Ein klebriges Problem	SCHMID, A.	28	71
RAMDISK	Die RAMDISK und du 11.Gebot	SCHMID, A.	25	54
RAMDISK	Die RAMDISK und das elfte Gebot	SCHMID, A.	25	56
RAMDISK	Darf's ein bisserl mehr sein?	SCHMID, A.	26	46
SPRACHE	Welche Sprache fuer wen ?	SCHMID, A.	27	59
TASTATUR	ES GEHT NOCH SCHNELLER (TAST.ABSCHALTEN)	SCHMID, A.	22	32
TEXTSYSTEM	Wie sag' ich's meinem WORDSTAR ?	SCHMID, A.	24	11
UHR	Es war einmal ein Datum	SCHMID, A.	26	29
ZEITMESSUNG	ETWAS fuer LEUTE, die NIE ZEIT haben	SCHMID, A.	22	31
DRUCKER	DIE DRUCKERABFRAGE	SCHMITZ	18	12
ERFAHRUNG	PC STATT GENIE? (Ueber weitere PC's)	SCHMITZ	20	99
HAENDLER	Reingefallen alt Rheintec	SCHMITZ	23	90
TEXTSYSTEM	WORDSTAR auf allen Laufwerken	SCHOBER	28	35
ASSEMBLER	"INPUT"	SCHROEDER	11	5
ASSEMBLER	EINIGE ASSEMBLER-ROUTINEN	SCHROEDER	14	17
ASSEMBLER	ASSEMBLER CONTRA LISP	SCHROEDER	17	19
ASSEMBLER	DIE WERKZEUGE DES ASSEMBL.PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	WERKZEUGE DES ASSEMBLER-PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	LMOFFSET with DISABLE DOS	SCHROEDER	9	9
BANKING	SYSCOPY1 und RAMDISK f.HELMUT's BANKER	SCHROEDER	16	32
BERICHT	GENIE IIs - Erfahrungsbericht	SCHROEDER	9	67
BILDSCHIRM	DIE 80-ZEICHEN-KARTE	SCHROEDER	12	63
BILDSCHIRM	80-ZEICHENKARTE X80, SEDIT und TSCRIPS	SCHROEDER	13	17
BILDSCHIRM	EIN NEUER 80ZEICHEN-TREIBER	SCHROEDER	19	77

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
GRAFIK	GRAFIK-STANDARD: ALLEIN GEHT'S AUCH	SCHROEDER	19	11
LITERATUR	BUECHERLISTE: DIE ERSTEN KURZKRITIKEN	SCHROEDER	21	67
MATHE/GRAFIK	RECHNEN (ZEICHNEN) MIT DEM COMPUTER...	SCHROEDER	10	53
PASCAL	KURZKRITIKEN und BUECHERLISTE	SCHROEDER	21	71
PLOTTER	KREIS OHNE TRIGONOMETRIE	SCHROEDER	15	8
PROZESSOR	Z80, HD64180 UND ILLEGALS	SCHROEDER	17	27
SPIEL	DAS PIRATE ADVENTURE	SCHROEDER	11	15
TEXTSYSTEM	AENDERUNGEN IN TSCRIPS und NAME/CMD	SCHROEDER	13	16
TEXTSYSTEM	FREIRAEUME IN TSCRIPS	SCHROEDER	14	47
TEXTSYSTEM	SCRIPSIT-TABELLEN	SCHROEDER	14	45
TEXTSYSTEM	TEXTUMLEITUNG fuer TSCRIPS	SCHROEDER	14	49
TEXTSYSTEM	TSCRIPS-VERSCHLUESSELUNG	SCHROEDER	15	29
GRAFIK	EINF.GRAFIKPROGRAMM: PLOT/BAS	SCHROEDER /ARPS	14	29
ASSEMBLER	Assembler unter CP/M	SCHROEDER, G.	25	49
BUS	Freiheit fuer den BUS	SCHROEDER, G.	25	55
SCHNITTSTELL	Entfernung der Kassettenschnittstelle	SCHROEDER, G.	26	40
SCHNITTSTELL	Memory mapped I/O <==> Port I/O bei GIIs	SCHROEDER, G.	26	42
VIDEO	BO-Zeichen-Karte	SCHROEDER, G.	12	63
Z80	INTERFACE mit Kassetten-Schnittstelle	SCHROEDER, G.	26	40
GRAFIK	LINIENMODELLE IN 3 DIMENSIONEN	SCHROEDER/ ARPS	14	35
MATHEMATIK	Fraktale Geometrie	SCHROEER	27	15
BANKING	Nochmaliges Banken	SOERENSEN	28	47
INTERRUPT	Unterbrechung gestattet ?	SOERENSEN	28	52
PASCAL	PASCAL-VORSTELLUNG TEIL 2	SOERENSEN	22	40
PASCAL	PASCAL fuer INTERESSIERTE	SOERENSEN, R.	20	12
ASCII	ASCII-TABELLE IMMER ZUR HAND	SOPP	19	19
ASSEMBLER	ZEUS MIT 64 UND 80 ZEICHEN	SOPP	18	36
ASSEMBLER	Was tun bei zerschossener Source ?	SOPP	24	25
ASSEMBLER	Bedingt assemblieren mit ZEUS	SOPP	25	24
ASSEMBLER	Bedingte Assembly - notfalle ohne IF	SOPP	25	28
ASSEMBLER	WER HAT ANGST VOR ASSEMBLER?	SOPP	5	?
ASSEMBLER	LPRINT ALLES einfacher! (Teil 3)	SOPP	9	11
BASIC	EIN MYSTERIOESER BASIC-BEFEHLs IsA	SOPP	5	?
BILDSCHIRM	MEHR FERNSEHEN fuer's GELD (f.GENIE IIIs	SOPP	14	21
BILDSCHIRM	FUTTER fuer den CRTC	SOPP	19	40
BILDSCHIRM	NOCH MEHR FERNSEHN fuers GELD	SOPP	19	39
BILDSCHIRM	EIN SCHONER fuer die BILDROEHRE	SOPP	20	20
BILDSCHIRM	EIN WEITERER BILDSCHIRMSCHONER	SOPP	22	28
BILDSCHIRM	VIDHEX - HEXANZEIGE d.BILDSCHIRMS m.HRB	SOPP	9	13
DRUCKEN	LPRINT MIT MODERNEN GENIES	SOPP	18	19
DRUCKER	VON 8 AUF 24 PINS (= DRUCKNADELN)	SOPP	22	20
DRUCKERGRAF.	GRAPHIK-JKL, TSCRIPS UND DER NEC Pi	SOPP	16	27
ECHTZEITUHR	ACH DU LIEBE ZEIT!	SOPP	15	32
ECHTZEITUHR	REAL TIME BLACK BOX?	SOPP	15	53
FILEVERGLCH	VERGLEICH ZWEIER DISK-DATEIEN	SOPP	22	53
FILEVERGLCH	VERGLEICH ZWEIER PROGRAMME! noch'n TOOL	SOPP	22	57
FILEVERGLCH	Vergleich zweier Disk-Dateien	SOPP	25	20
GRAFIK	GRAFIK-CLS fuer das GENIE IIIs	SOPP	13	35

STICHWORT [2]	**TITEL aus den Club-80 Infos 31/12/1989**	***VERFASSER*** [1]	**Info**	**Seite**
GRAFIK	STRINGANZEIGE MIT VOLLDAMPF	SOPP	6	41
GRAFIK-JKL	GRAFIK-JKL fuer das GENIE IIIs	SOPP	13	32
HARDCOPY	FLOPCOPY AUCH OHNE INTERRUPT	SOPP	21	14
HRG	HRG 1b PROGRAMMIEREN, BESSER ERKLAERT	SOPP	10	25
HRG	HRG UND BASIC NETTO	SOPP	10	27
HRG	HRG-SPEICHER LOESCHEN, DEN -	SOPP	10	29
HRG	GENIE IIIs, DIE HRG DES -	SOPP	12	21
HRG	HRG-HARDCOPY fuer das G IIIs	SOPP	13	25
HRG	HRG, dasz die Schwarte kracht !	SOPP	24	40
JKL	JKL - ABER DRUCKERSCHONEND!	SOPP	12	13
JOYSTICK	Joystick-Anschluss	SOPP	6	29
LAUFWERK	Ordnung muss sein !	SOPP	28	33
LIBRARY	WAS TUN BEI VOLLER LIBRARY?	SOPP	12	15
LOGIK	LOGISCHES ZU LOGIK	SOPP	16	14
MATHEMATIK	Bin.-Umwandlg.HEX/DEZ;kuerzere E-Routine	SOPP	24	23
MATHEMATIK	Binaeruwaandlung, letzter (?) Teil	SOPP	25	12
MAUS	Maus mit sieben Beinen	SOPP	23	69
MAUS	Die Pratze laeszt das Mausen nicht	SOPP	25	18
PORT	OUT,PORT#,xx,yy,...	SOPP	11	29
PORT	VERBESSERT: IO.PORT#<,xx,yy,.. >	SOPP	12	11
RAMDISK	MINI-RAM-FLOPPY im "SICHEREN PLAETZCHEN"	SOPP	10	37
RECORDORGAN.	ZU GERALDs ARTIKEL ueber LMOFFSET	SOPP	10	3
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES	SOPP	13	23
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES (NACHTRAG)	SOPP	14	53
SPEICHER	CALL UM DIE ECKE (MEM-BEREICHE ERREICHEN	SOPP	15	49
SYSTEM	Quatsch in OVL4/SYS	SOPP	24	27
SYSTEM	Schon Mieder SYS14/SYS	SOPP	24	20
TEXT-GRAFIK	(TELEFON) - EIN UD-ZEICHEN auf d.DRUCKER	SOPP	16	29
TEXTSYSTEM	TIPS fuer SCRIPS	SOPP	13	19
TEXTSYSTEM	NEUES ZUR PROPORTIONALSCHRIFT	SOPP	18	15
TEXTSYSTEM	§ CONTRA .	SOPP	19	27
TEXTSYSTEM	TSCRIPS & DIE &-CODES	SOPP	19	16
TEXTSYSTEM	VON DER WIEGE BIS ZUR BAHRE (ADR.AUTOMAT	SOPP	19	33
TEXTSYSTEM	KEINE ANGST VOR PROPORTIONALSCHRIFT	SOPP	18	13
TEXTSYSTEM	TSCRIPS UND DIE 24 NADELN	SOPP	22	23
UHR	WENN DIE UHR MAL STOERT	SOPP	14	39
UHR	THE TIMES, THEY ARE A-CHANGING	SOPP	22	35
UTILITY	LPRINT ALLES	SOPP	6	31
ZEICHENSATZ	NEUE ZEICHENSAETZE fuer das GENIE IIIs	SOPP	12	28
ASSEMBLER	SCHOENER LISTEN MIT ZEUS	SOPP	18	41
COMPUTER	DEIN GIIIs... (NACHTRAG)	SOPP/BERNHARDT	17	54
TEXTSYSTEM	DAS TSCRIPS-DRAMA - EIN LEHRSTUECK	SOPP/MUHLENBEIN	21	56
BUS	Indikatorkarte fuer ECB-BUS	SPIESS	19	45
BUS	Analog / Digital - Umsetzer	STOBER	24	37
DRUCKER	Funktionsweise des Drucker-Puffers	TRAPP	6	18
MODEM	Akustikkoppler	TRAPP	4	?
SPRACHE	COBOL - EINE EINFUEHRUNG	TRAPP	5	?
UTILITY	BILDSCHIRMAUSDRUCK BEIM TRS80+VIDEOGENIE	UEBERSCHAAR		

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
LITERATUR	WEITERE KURZKRITIKEN	VERSCH.	22	DIV
SPIEL	ADVENTURE GEHEIMAGENT XP-05	WAGNER, ALEX.	3	?
KASSETTE	CASSETTEVERZEICHNIS ERSTELLEN	WAGNER, G.	10	33
M4/4p	AUFGESTIEGEN?	WAGNER, G.	14	59
PROGRAMMIERG	DIRECTORY LESEN	WAGNER, G.	9	4
PROGRAMMIERG	EDITOR	WAGNER, G.	9	5
PROMMER	EPROM-Programmiergeraet	WOLF (aus mc)	5	?
MATHEMATIK	Was man weiss - schon inner wissen wollt	WOLF, Thomas	26	15
ASSEMBLER	BASIC RUFT ASSEMBLER	WOLLSCHLAEGER	21	11
ASSEMBLER	ZBO-Prograamtricks in Assembler	WOLLSCHLAEGER	27	39
COMPUTER	Modell 4	WOLLSCHLAEGER	12	69
MODEL 6	KLEINE ZIPPERLEIN RASCH KURIERT	WOLLSCHLAEGER	12	57
PROGRAMMIERG	TRICKS MIT STRINGS	WOLLSCHLAEGER	4	45
PROGRAMMIERG	PROGRAMMIEREN HEISST m. DATEIEN ARBEITEN	WOLLSCHLAEGER	7	29
SICHERHEIT	RAM-DISK KOSTENLOS	WOLLSCHLAEGER	2	?
CHIP	BEI SIEMENS HEISST ES: CHIP CHIP HURRA!	ZEITSCHRIFT	18	76
HAENDLER	TANDY SCHLIESST COMPUTER-CENTER	ZEITSCHRIFT	18	75
IC	Hardware-Tips: Neue IC's	ZEITSCHRIFT	17	38
LOGIK	LOGIKSIMULATION mit BASIC	ZEITSCHRIFT	16	11
PROGRAMMIERG	DEM INTERPRETER auf» BIT GESCHAUT	ZEITSCHRIFT	12	37
PROGRAMMIERG	DIE RICHTIGE DIMENSION	Zeitschrift	18	24
PROZESSOR	TAV's Z80-TUNING	ZEITSCHRIFT	17	33
UTILITY	SMARTSCREEN, SMARTKEY und SMARTPRINT	ZEITSCHRIFT	12	45
ASSEMBLER	TIP FUER SCHNELLERES ARBEITEN MIT EDTASM	ZWICKEL	6	67
COMPUTER	WIE PHOENIX AUS DER ASCHE (4 COMP.in 1)		17	9
COMPUTER	CP/MAC-eine BRUECKE zw.G-/NEWDOS u.CP/M		18	44
COMPUTER	32 BIT zum NIEDRIGPREIS (TANDY 4000)		22	79
CONTROLLER	FLOPPYCONTROLLER-KOMPATIBILITAET		19	47
CP/M	CP/MAC - NA UND?		19	78
CP/M	WAS HABT IHR GEGEN CP/M ?		21	53
CP/M	GENIE, CP/M UND 80 ZEICHEN UND HRG???		21	43
CP/M	CMD ==> COM ohne CPMac		21	23
CP/M	Wie baue ich mir ein CP/M ?		21	25
CP/M	CP/M intern - kurz und kompakt		23	49
CP/M	Grafik unter CP/M		25	35
CP/M	Z-CP/M - eine deutliche Verbesserung		25	37
CP/M	ZCPR33 - Installation auf M4...		25	45
CP/M	BDOS-Replacement - was'n du?		26	27
CP/M	CP/M-Warmstart ohne Nachladen		26	25
CP/M	PROMMER 80 unter G-DOS		28	58
CP/M	Fensterin mit CP/M		28	64
CP/M	Backen und Banken		28	44
CP/M	ZCPR33 - die Vierte		28	53
CPU	Z80-Hochintegrations-CPUs		23	88
DATEI	INDEX-DATEIEN		19	15
DATEN	VIDEODAT 300		13	51
DATEN	DATENVERLUSTE		19	80
DATENBANK	dBASE II - eine Einfuehrung		21	33

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
DATENBANK	BDBS - BASIC DATENBANK SYSTEM (MODEL 4)		22	37
DATENBANK	dBASE II		23	65
DATENBANK	Datenbanken richtig aufgebaut		26	73
DATENBANK	Dateiprogramm		28	22
DATENBANK	Teamwork		28	37
DDE	DDE fuer das GENIE IIIs		12	17
DEF.FUNCTION	BASIC-PROGRAMMIERBARE FUNKTIONEN		19	26
DENKAUFGABEN	MATHEMATIK fuer FEINSCHMECKER		20	101
DFUe	Datenuebertragung / M4p als Terminal		23	30
DISASSEMBLER	Kleine Hexen bevorzugt		23	27
DISASSEMBLER	Kleine Hexen ohne Aufwand		25	17
DISASSEMBLER	Kurzanleitung fuer SCREENER/CMD		25	15
DISASSEMBLER	Screening mit DSMBLR/CMD		25	14
DOS	EXTENDED NEWDOS 80		10	5
DOS	NEWDOS/80+10 VERSION 2.4 (26.10.85)		11	25
DOS	DOS - (FAST) OHNE FLOPPY		14	69
DOS	DIR/SYS ODER INHALT/SYS?		15	21
DOS	FILTER UNTER TRSDOS 6.x		18	29
DOS	SUPERDOS		2	?
DOS	MDOS im INFO???????		20	11
DOS	LS-DOS 6.3 - KOPIERSCHUTZ		21	55
DOS	Der Teufe) steckt oft im Detail		25	30
DOS	Ein kleiner Fehler im Z80-DOS		27	14
DOS	Microcomputer-Betriebssysteme		28	103
DOS-FEHLER	FEHLER IM NEWDOS-BEFEHL 'BOOT1		11	21
DOS-Fehler	Ein Fehler in SYS14/SYS		24	19
DOUBLER	SELBSTBAU-DOUBLER fuer EXP1		17	45
DRUCKEN	DREISPALTIG DRUCKEN		21	15
DRUCKEN	EIN FEHLER IM SYS28/SYS		22	33
DRUCKEN	Druckausgabe f. COBOL-Programme		23	32
DRUCKEN	Druckausgabe mit PRBUF		24	12
HAENDLER	TANDY-VERTAGSHAENDLER		19	79
RECHT	COMPUTERKRIMINALITAET		19	81
SPRACHE	COBOL-EINFUEHRUNG		8	5
VIDEO	VIDEODAT-SYSTEM		1B	57
VERFASSER [1]	**Der hochgestellte Index gibt die Reihenfolge der Sortierung an**			
STICHWORT [2]				

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
ADAPTER	Testadapter	REGGE	9	45
AMATEURFUNK	Die Betriebsart PACKET-Radio (PR)	HARTMANN, H.-G.	28	111
ASCII	Wogegen selbst ZEUS vergeblich kaempft	MUEHLENBEIN	25	5
ASCII	ASCII-TABELLE IMMER ZUR HAND	SOPP	19	19
ASSEMBLER	KLEINSCHRIFT fuer den EDTASM+	HEIDENREICH	20	37
ASSEMBLER	SCHUETZEN VON ASSEMBLERPROGRAMMEN	KONRAD J.	5	?
ASSEMBLER	HALBAUTOMATISCHE FESTLEGUNG VON ORG	KONRAD, J.	9	3
ASSEMBLER	MEIN ERSTES ASSEMBLERPROGRAMM	MUEHLENBEIN	14	9
ASSEMBLER	MEIN DRITTES ASSEMBLERPROGRAMM	MUEHLENBEIN	16	33
ASSEMBLER	"INPUT"	SCHROEDER	11	5
ASSEMBLER	EINIGE ASSEMBLER-ROUTINEN	SCHROEDER	14	17
ASSEMBLER	ASSEMBLER CONTRA LISP	SCHROEDER	17	19
ASSEMBLER	DIE WERKZEUGE DES ASSEMBL.PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	WERKZEUGE DES ASSEMBLER-PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	LMOFFSET with DISABLE DOS	SCHROEDER	9	9
ASSEMBLER	Assembler unter CP/M	SCHROEDER, G.	25	49
ASSEMBLER	ZEUS MIT 64 UND 80 ZEICHEN	SOPP	18	36
ASSEMBLER	Was tun bei zerschossener Source ?	SOPP	24	25
ASSEMBLER	Bedingt assemblieren mit ZEUS	SOPP	25	24
ASSEMBLER	Bedingte Assembly - notfalle ohne IF	SOPP	25	28
ASSEMBLER	WER HAT ANGST VOR ASSEMBLER?	SOPP	5	?
ASSEMBLER	LPRINT ALLES einfacher! (Teil 3)	SOPP	9	11
ASSEMBLER	SCHOENER LISTEN MIT ZEUS	SOPP	18	41
ASSEMBLER	BASIC RUFT ASSEMBLER	WOLLSCHLAEGER	21	11
ASSEMBLER	ZBO-Prograamtricks in Assembler	WOLLSCHLAEGER	27	39
ASSEMBLER	TIP FUER SCHNELLERES ARBEITEN MIT EDTASM	ZWICKEL	6	67
AUFRUESTEN	Modernisierung statt Neukauf	MATTHAEI/H.COMP	5	?
BANKING	256K-RAM fuer Z80-SYSTEME	BERNHARDT	14	91
BANKING	Bangen um Banken	MUEHLENBEIN	25	7
BANKING	SYSCOPY1 und RAMDISK f.HELMUT's BANKER	SCHROEDER	16	32
BANKING	Nochmaliges Banken	SOERENSEN	28	47
BASIC	EXTENDED BASIC(RESTOREn u. LPRINT alles)	OBERMANN	11	33
BASIC	EIN MYSTERIOESER BASIC-BEFEHLs IsA	SOPP	5	?
BERICHT	Testbericht KX 1083	MAND	23	89
BERICHT	GENIE IIs - Erfahrungsbericht	SCHROEDER	9	67
BILDFORMAT	FUTTER fuer GENIE IIIs CRTC	HEIDENREICH	20	22
BILDSCHIRM	BILDSCHIRMAUSGABE mit INVERSEN ZEICHEN	?	12	79
BILDSCHIRM	Bessere Darst.v.Text u.HRG auf d.Bildsch	BERNHARDT	26	37
BILDSCHIRM	BIN ICH STRAHLENGEFAEHRDET?	BOIKAT, Ute	20	109
BILDSCHIRM	DAS PROJEKT (FERNSEHER –> MONITOR)	MUELLER, K.	18	61
BILDSCHIRM	NOCH EIN BILDSCHIRMSCHONER	SCHMID, A.	22	26
BILDSCHIRM	SCHWIMMENDE BILDER	SCHMID, A.	22	66
BILDSCHIRM	DIE 80-ZEICHEN-KARTE	SCHROEDER	12	63
BILDSCHIRM	80-ZEICHENKARTE X80, SEDIT und TSCRIPS	SCHROEDER	13	17
BILDSCHIRM	EIN NEUER 80ZEICHEN-TREIBER	SCHROEDER	19	77
BILDSCHIRM	MEHR FERNSEHEN fuer's GELD (f.GENIE IIIs	SOPP	14	21
BILDSCHIRM	FUTTER fuer den CRTC	SOPP	19	40
BILDSCHIRM	NOCH MEHR FERNSEHN fuers GELD	SOPP	19	39

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
BILDSCHIRM	EIN SCHONER fuer die BILDROEHRE	SOPP	20	20
BILDSCHIRM	EIN WEITERER BILDSCHIRMSCHONER	SOPP	22	28
BILDSCHIRM	VIDHEX - HEXANZEIGE d.BILDSCHIRMS m.HRB	SOPP	9	13
BUS	PROF 180x und mc68000-ECB	BERNHARDT	25	57
BUS	ECB-BUS-Projekt	DROWAELD. &OBERM	15	65
BUS	ECB-BUS-Projekt	OBERMANN	9	42
BUS	Freiheit fuer den BUS	SCHROEDER, G.	25	55
BUS	Indikatorkarte fuer ECB-BUS	SPIESS	19	45
BUS	Analog / Digital - Umsetzer	STOBER	24	37
CHIP	NANOHENRY BLOCKIEREN DEN FORTSCHRITT	aus c' t	19	70
CHIP	DER LEIDENSWEG DER CMOS-CHIPS	aus c't	19	67
CHIP	ERSTER MEGABIT-CHIP IN SERIE	aus Zeitschrift	18	18
CHIP	BEI SIEMENS HEISST ES: CHIP CHIP HURRA!	ZEITSCHRIFT	18	76
CODE	BASICODE-INTERFACE	?(aus ELEKTOR)	3	?
CODE	BASICODE—2	?(aus Elektor)	3	?
COMPILER	BASCOM-KURZANLEITUNG	MUEHLENBEIN	13	38
COMPILER	RPNL - DIE ANDERE ART ZU PROGRAMMIEREN	MUELLER, KURT	15	57
COMPILER	RPNL-BESCHREIBUNG	MUELLER, KURT	16	37
COMPILER	RPNL	Mueller, Kurt	26	31
COMPILER	CALL mit BASCOM	OBERMANN	12	33
COMPUTER	SHARP 3541	aus Comp.Pers.	26	47
COMPUTER	Der fluesternde Portable	BETZ	28	67
COMPUTER	UNTERSCHIEDE (zw. MODEL 1 und 3 bzw. 4)	OBERMANN	14	55
COMPUTER	MODEL 4p ROM-IMAGE-LOADER	OBERMANN	15	43
COMPUTER	TANDY<—> SCHNEIDER	OBERMANN	17	49
COMPUTER	DEIN GIIIs... (NACHTRAG)	SOPP/BERNHARDT	17	54
COMPUTER	Modell 4	WOLLSCHLAEGER	12	69
COMPUTER	WIE PHOENIX AUS DER ASCHE (4 COMP.in 1)		17	9
COMPUTER	CP/MAC-eine BRUECKE zw.G-/NEWDOS u.CP/M		18	44
COMPUTER	32 BIT zum NIEDRIGPREIS (TANDY 4000)		22	79
CONTROLLER	FLOPPYCONTROLLER-KOMPATIBILITAET		19	47
CP/M	Billiges CP/M	BREWER	9	53
CP/M	CP/MAC - NA UND?		19	78
CP/M	WAS HABT IHR GEGEN CP/M ?		21	53
CP/M	GENIE, CP/M UND 80 ZEICHEN UND HRG???		21	43
CP/M	CMD ==> COM ohne CPMac		21	23
CP/M	Wie baue ich mir ein CP/M ?		21	25
CP/M	CP/M intern - kurz und kompakt		23	49
CP/M	Grafik unter CP/M		25	35
CP/M	Z-CP/M - eine deutliche Verbesserung		25	37
CP/M	ZCPR33 - Installation auf M4...		25	45
CP/M	BDOS-Replacement - was'n du?		26	27
CP/M	CP/M-Warmstart ohne Nachladen		26	25
CP/M	PROMMER 80 unter G-DOS		28	58
CP/M	Fensterin mit CP/M		28	64
CP/M	Backen und Banken		28	44
CP/M	ZCPR33 - die Vierte		28	53
CPU	Z80-Hochintegrations-CPUs		23	88

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
DATEI	INDEX-DATEIEN		19	15
DATEN	VIDEODAT 300		13	51
DATEN	DATENVERLUSTE		19	80
DATENBANK	dBASE II - eine Einfuehrung		21	33
DATENBANK	BDBS - BASIC DATENBANK SYSTEM (MODEL 4)		22	37
DATENBANK	dBASE II		23	65
DATENBANK	Datenbanken richtig aufgebaut		26	73
DATENBANK	Dateiprogramm		28	22
DATENBANK	Teamwork		28	37
DBASE	dBASE II, III und III+ Ueberblick	aus DOS 5'88	26	67
DDE	DDE fuer das GENIE IIIs		12	17
DECODER	Eleganter Adressdecoder fuer GENIE I	BERNHARDT	28	73
DEF.FUNCTION	BASIC-PROGRAMMIERBARE FUNKTIONEN		19	26
DENKAUFGABEN	MATHEMATIK fuer FEINSCHMECKER		20	101
DFUe	Bewaehrtes DFUe-Protokoll	aus Comp.Pers.	27	51
DFUe	Datenuebertragung per Fernsehbild	Edde & Heribert	12	62
DFUe	DATENUEBERTRAGUNG p.FERNSEHBILD/WDR-CLUB	EDDE u.HERIBERT	12	62
DFUe	Datenuebertragung / M4p als Terminal		23	30
DISASSEMBLER	Kleine Hexen bevorzugt		23	27
DISASSEMBLER	Kleine Hexen ohne Aufwand		25	17
DISASSEMBLER	Kurzanleitung fuer SCREENER/CMD		25	15
DISASSEMBLER	Screening mit DSMBLR/CMD		25	14
DISK	Sicherer Platz fuer Daten? (Disketten)	aus test 9/87	22	73
DOS	R-BEFEHL MSDOS-LIKE (ein Editor f. DOS)	HEIDENREICH	20	44
DOS	Moeglichkeiten und Grenzen des MS-DOS	HEIDENREICH	27	19
DOS	EXTENDED NEWDOS 80		10	5
DOS	NEWDOS/80+10 VERSION 2.4 (26.10.85)		11	25
DOS	DOS - (FAST) OHNE FLOPPY		14	69
DOS	DIR/SYS ODER INHALT/SYS?		15	21
DOS	FILTER UNTER TRSDOS 6.x		18	29
DOS	SUPERDOS		2	?
DOS	MDOS im INFO???????		20	11
DOS	LS-DOS 6.3 - KOPIERSCHUTZ		21	55
DOS	Der Teufe) steckt oft im Detail		25	30
DOS	Ein kleiner Fehler im Z80-DOS		27	14
DOS	Microcomputer-Betriebssysteme		28	103
DOS-FEHLER	FEHLER IM NEWDOS-BEFEHL 'BOOT1		11	21
DOS-Fehler	Ein Fehler in SYS14/SYS		24	19
DOUBLER	SELBSTBAU-DOUBLER fuer EXP1		17	45
DRUCKEN	LPRINT MIT MODERNEN GENIES	SOPP	18	19
DRUCKEN	DREISPALTIG DRUCKEN		21	15
DRUCKEN	EIN FEHLER IM SYS28/SYS		22	33
DRUCKEN	Druckausgabe f. COBOL-Programme		23	32
DRUCKEN	Druckausgabe mit PRBUF		24	12
DRUCKER	STAR NL 10 abwechselnd am CP/M-R. + PC	BERNHARDT	28	72
DRUCKER	Noch ein Druckerpatch	HELD	23	71
DRUCKER	SPEED WRITING (fuer EPSON MX-DRUCKER)	LINDSEY	12	65
DRUCKER	Speed Writing (erhoeht Druckgeschwind.)	LINDSEY	12	65

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
DRUCKER	...WEIL KUERZE SO WUERZIG IST...	MUEHLENBEIN	20	35
DRUCKER	DIE DRUCKERABFRAGE	SCHMITZ	18	12
DRUCKER	VON 8 AUF 24 PINS (= DRUCKNADELN)	SOPP	22	20
DRUCKER	Funktionsweise des Drucker-Puffers	TRAPP	6	18
DRUCKERGRAF.	GRAPHIK-JKL, TSCRIPS UND DER NEC Pi	SOPP	16	27
DRUCKZEICHEN	Zeichengenerator fuer Nadeldrücker	aut nc	23	18
DRUCKZEICHEN	AM BILDSCHIRM DEFINIERT/EDITOR f.DRUCKER	KLUGE	9	19
ECB-BUS	ECB-ADAPTER-PLATINE	?	15	65
ECB-BUS	VERBINDEND	ASSENBAUM	11	39
ECB-BUS	Anm. zum CLUB80-ECB-Bus-Projekt	BERNHARDT	23	73
ECB-BUS	Gepufferter DMA- u. IM2-faehiger ECB	BERNHARDT	23	74
ECB-BUS	Prof 180x und mc-68OOO-ECB	BERNHARDT	25	57
ECB-BUS	DIE EINGANGSPLATINE	DROWAELDER	15	73
ECB-BUS	ECB-BUS-TRS80-PROJEKT	NEUEDER	13	39
ECB-BUS	ECB-BUS-PROJEKT	NEUEDER	14	101
ECB-BUS	ECB-BUS-GRUNDEINHEITSAUFBAU	NEUEDER	15	71
ECB-BUS	ECB-BUS fuer MODEL 3/4!???	OBERMANN	15	84
ECHTZEITUHR	ACH DU LIEBE ZEIT!	SOPP	15	32
ECHTZEITUHR	REAL TIME BLACK BOX?	SOPP	15	53
ERFAHRUNG	PC STATT GENIE? (Ueber weitere PC's)	SCHMITZ	20	99
ERROR	"PRUEFZAHLFEHLER BEIM LESEN"	BERNHARDT	22	71
FILEVERGLCH	VERGLEICH ZWEIER DISK-DATEIEN	SOPP	22	53
FILEVERGLCH	VERGLEICH ZWEIER PROGRAMME! noch'n TOOL	SOPP	22	57
FILEVERGLCH	Vergleich zweier Disk-Dateien	SOPP	25	20
FLOPPY	FLOPPY-TRICKS	OBERMANN	13	41
FORMATIERUNG	40 TRACKS? WARUM NICHT 42?	OBSCHERNINGKAT	18	11
FREQUENZ	Taktfrequenz-Verdopplung GENIE I/II	DREYER	5	?
FREQUENZ	Taktumschaltung	MUELLER, Kurt	16	45
FREQUENZ	Taktumschaltung fuer VIDEO-GENIE	OBERMANN	8	25
FREQUENZ	Synchronisierte Taktumschaltung	RETZLAFF	10	49
GENAUIGKEIT	MATHEMATISCHER UEBERFLUSS	MUEHLENBEIN	20	17
GRAFIK	Auslesen des Grafikspreichers	BERNH./SEELMANN	20	65
GRAFIK	Es lebe der Standard!	BERNHARDT	28	78
GRAFIK	ERGAENZUNG ZU GRAFIK-SHORTY	HERMANN	19	32
GRAFIK	TRS80 Ml/3 SCHNELLE GRAFIK IN BASIC	HORNUNG	2	?
GRAFIK	HRG-BILDUEBERTRAGUNG OHNE HRG (WDR-FILES	OBERMANN	15	11
GRAFIK	SCHNELL BEWEGTE GRAFIK AUF DEM TRS80	ROECKRATH	2	?
GRAFIK	GRAFIK-STANDARD: ALLEIN GEHT'S AUCH	SCHROEDER	19	11
GRAFIK	EINF.GRAFIKPROGRAMM: PLOT/BAS	SCHROEDER /ARPS	14	29
GRAFIK	LINIENMODELLE IN 3 DIMENSIONEN	SCHROEDER/ ARPS	14	35
GRAFIK	GRAFIK-CLS fuer das GENIE IIIs	SOPP	13	35
GRAFIK	STRINGANZEIGE MIT VOLLDAMPF	SOPP	6	41
GRAFIK-JKL	GRAFIK-JKL fuer das GENIE IIIs	SOPP	13	32
GRUNDLAGEN	Digitaltechnik, Grundlagen der -	gesammelt: KUHN		
HAENDLER	Reingefallen alt Rheintec	SCHMITZ	23	90
HAENDLER	TANDY SCHLIESST COMPUTER-CENTER	ZEITSCHRIFT	18	75
HAENDLER	TANDY-VERTAGSHAENDLER		19	79
HARDCOPY	TRS80 erstellt HARDCOPY	aut nc	23	17

STICHWORT [1]	**TITEL aus den Club-80 Infos 31/12/1989**	*VERFASSER* [2]	Info	Seite
HARDCOPY	HARDCOPY AUCH OHNE INTERRUPT	RYCHLIK	20	15
HARDCOPY	FLOPCOPY AUCH OHNE INTERRUPT	SOPP	21	14
HRG	Traumlandschaften aus dem Computer	aus Comp.Pers.	27	41
HRG	Wie funktioniert ein Vektorgenerator	aus Comp.Pers.	27	31
HRG	HRG-1b-Karte	MUELLER, KURT	14	89
HRG	HRG vom BASIC aus (DEMO)	NEUEDER	10	31
HRG	HRG-PIXELS aus dem BASIC!	NEUEDER	10	23
HRG	HRG: 384x192 oder 192x192	NEUEDER	13	14
HRG	HRG-BILDUEBERTRAGUNG per DFUe	OBERMANN	14	24
HRG	UMKEHR!	OBERMANN	20	58
HRG	HRG-Bilduebertragung	OBERMANN	25	31
HRG	HRG 1b PROGRAMMIEREN, BESSER ERKLAERT	SOPP	10	25
HRG	HRG UND BASIC NETTO	SOPP	10	27
HRG	HRG-SPEICHER LOESCHEN, DEN -	SOPP	10	29
HRG	GENIE IIIs, DIE HRG DES -	SOPP	12	21
HRG	HRG-HARDCOPY fuer das G IIIs	SOPP	13	25
HRG	HRG, dasz die Schwarte kracht !	SOPP	24	40
IC	Hardware-Tips: Neue IC's	ZEITSCHRIFT	17	38
INTERFACE	Relais-Interface	?	9	47
INTERFACE	Interface-Adapter 8255, Programmierbarer	KUHN	13	47
INTERRUPT	Unterbrechung gestattet ?	SOERENSEN	28	52
JKL	JKL - ABER DRUCKERSCHONEND!	SOPP	12	13
JOYSTICK	Joysticks fuer den TRS-80 M1	NEUEDER	5	?
JOYSTICK	Joystick-Anschluss	SOPP	6	29
KASSETTE	CASSETTEVERZEICHNIS ERSTELLEN	WAGNER, G.	10	33
KOPIERSCHUTZ	6.3 - KOPIERSCHUTZ	HERMANN	21	55
LAUFWERK	EXP1Floppy-Controller	BERNHARDT	26	38
LAUFWERK	Floppydisk-Controllerkarte	DROWAELDER	18	49
LAUFWERK	Floppy-Tricks	OBERMANN	14	87
LAUFWERK	80/40-Trackumschaltung Mitsubishi	OBERMANN	26	39
LAUFWERK	LWT fuer GENIE I und TRS-80 MODELL I	RETZLAFF	13	40
LAUFWERK	Ein klebriges Problem	SCHMID, A.	28	71
LAUFWERK	Ordnung muss sein !	SOPP	28	33
LIBRARY	WAS TUN BEI VOLLER LIBRARY?	SOPP	12	15
LITERATUR	BUECHERLISTE: DIE ERSTEN KURZKRITIKEN	SCHROEDER	21	67
LITERATUR	WEITERE KURZKRITIKEN	VERSCH.	22	DIV
LOGIK	STENO-LOGIK	MUEHLENBEIN	16	24
LOGIK	BOOLE XOR ABEL	MUEHLENBEIN	17	16
LOGIK	LOGISCHE OPERATIONEN IN BASIC	OBERMANN	15	9
LOGIK	LOGISCHE OPERATIONEN 2	OBERMANN	16	9
LOGIK	LOGISCHES ZU LOGIK	SOPP	16	14
LOGIK	LOGIKSIMULATION mit BASIC	ZEITSCHRIFT	16	11
M4/4p	AUFRUF DER SUPERVISOR-CALLS in BASIC	HERMANN	14	62
M4/4p	DAS BOOT-ROM des 4p	OBERMANN	14	63
M4/4p	HRG fuer MODEL 3/4/4p	OBERMANN	14	67
M4/4p	Atari-Bilder auf den 4p	OBERMANN	26	21
M4/4p	Das neueste DOS fuer M4/4p	OBSCHERNINGKAT	24	29
M4/4p	AUFGESTIEGEN?	WAGNER, G.	14	59

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
MATH.STATIST	BIVARANA (2fache Varianzanalyse)	MUEHLENBEIN	19	21
MATHE/GRAFIK	RECHNEN (ZEICHNEN) MIT DEM COMPUTER...	SCHROEDER	10	53
MATHEMATIK	WIE der NAT.LOG.die RECHENZEIT VERKUERZT	GUAGLIANO	14	77
MATHEMATIK	SPLINE-INTERPOLATION (KUBISCH)	KASPER	16	19
MATHEMATIK	PRIMZAHLBERECHNUNGEN (3 PROGRAMME)	KASPER	5	?
MATHEMATIK	GOLDENER FIBONACCI	MUEHLENBEIN	16	21
MATHEMATIK	UMWANDLUNG! ROEMISCHE IN ARABISCHE ZAHL	MUEHLENBEIN	16	7
MATHEMATIK	DAS CARTESISCHE BLATT	MUEHLENBEIN	18	5
MATHEMATIK	CRAMER UND DIE DETERMINANTEN	MUEHLENBEIN	19	13
MATHEMATIK	DER SCHNELLSTE ZAHNARZT (WURZELZIEHEN)	MUEHLENBEIN	22	11
MATHEMATIK	Polynonberechnung	MUEHLENBEIN	24	13
MATHEMATIK	Ableitg.d.Negationsregel f.Binaerzahlen	MUEHLENBEIN	25	9
MATHEMATIK	Ein letzter 10Zeiler (Determinantenber.)	MUEHLENBEIN	28	10
MATHEMATIK	PASCAL St GALTON	MUEHLENBEIN	28	17
MATHEMATIK	Sitzverteilung nach d'Hondt	MUEHLENBEIN	2B	11
MATHEMATIK	Fraktale Geometrie	SCHROEER	27	15
MATHEMATIK	Bin.-Umwandlg.HEX/DEZ;kuerzere E-Routine	SOPP	24	23
MATHEMATIK	Binaeruwaandlung, letzter (?) Teil	SOPP	25	12
MATHEMATIK	Was man weiss - schon inner wissen wollt	WOLF, Thomas	26	15
MATHE-SPIEL	WAS IST MIT DER PRIMZAHL '37' LOS?	MUEHLENBEIN	11	23
MAUS	Die Maus am TRS-80 !	OBERMANN	22	60
MAUS	Maus mit sieben Beinen	SOPP	23	69
MAUS	Die Pratze laeszt das Mausen nicht	SOPP	25	18
MEMDISK	RESET-FESTE MEMDISK M4-CP/M	OBERMANN	19	43
MODEL 4	BOOT-ROM-VERSION und DATUM M4	HERMANN	19	42
MODEL 5	TURBO TIP	OBERMANN	21	31
MODEL 6	KLEINE ZIPPERLEIN RASCH KURIERT	WOLLSCHLAEGER	12	57
MODEM	Akustikkoppler	TRAPP	4	?
MONITOR	MONITOR IM GENIE	KASPER	6	23
MUSIK	MEIN ZWEITES ASSEMBLERPROGRAMM	MUEHLENBEIN	15	15
MUSIK	SCHWERE KAVALLERIE ("DAS CLUBLIED")	MUEHLENBEIN	22	17
PASCAL	Echte Progranne neiden PASCAL	aus DATAMON7/83	23	91
PASCAL	APFELMAENNCHEN auf der GDP64	BERNHARDT	21	37
PASCAL	GENIE, CP/M und 80 ZEICHEN und HRG ???	HELD	21	43
PASCAL	PASCAL-PROBLEME	KONRAD, J.	9	73
PASCAL	Vergleich BASIC nit TURBO-PASCAL	MUEHLENBEIN	28	39
PASCAL	TURBO PATCH	OBERMANN	21	32
PASCAL	TURBO PATCH (FEHLERKORREKTUR)	OBERMANN	21	32
PASCAL	TURBO TIP	OBERMANN	21	31
PASCAL	TURBO-TIP	OBERMANN	21	31
PASCAL	Vergleichstest (versch.PASCAL-Versionen)	OBERMANN	23	33
PASCAL	KURZKRITIKEN und BUECHERLISTE	SCHROEDER	21	71
PASCAL	PASCAL-VORSTELLUNG TEIL 2	SOERENSEN	22	40
PASCAL	PASCAL fuer INTERESSIERTE	SOERENSEN, R.	20	12
PASSWORT	PASSWORT-BYPASS BEI TRSDOS 6.2	OBERMANN	19	42
PATCH	GROSZBUCHSTABENMODUS AB SYSTEMSTART (M4)	HERMANN	21	30
PDRIVE	PDRIVE-IDENTIFIKATION fuer NEWDOS 80	OBERMANN	10	17
PLOTTER	KREIS OHNE TRIGONOMETRIE	SCHROEDER	15	8

STICHWORT [1]	**TITEL aus den Club-80 Infos 31/12/1989**	***VERFASSER*** [2]	**Info**	**Seite**
POKEPEEK	POKE+PEEK-ECKE	HANS -?-	4	?
PORT	PORTBELEGUNG von COMPUTERKONFIGURATIONEN	KUHN	21	44
PORT	Portbelegung	KUHN	21	44
PORT	OUT,PORT#,xx,yy,...	SOPP	11	29
PORT	VERBESSERT: IO.PORT#<,xx,yy,.. >	SOPP	12	11
PROGRAMMIERG	STRUKTURIERTE PROGRAMMIERUNG	?	7	17
PROGRAMMIERG	KEINE MEINUNG ZUR STRUKTUR?	GRELL	7	28
PROGRAMMIERG	EDIT-BEFEHLE	HERMANN	14	4
PROGRAMMIERG	BASIC- LAUFENDE PROGRAMME ERHEITERN	KRENZKE	16	18
PROGRAMMIERG	LOKALE VARIABLE BEIM TRS 80	PAULO	7	7
PROGRAMMIERG	PLANVOLL PROGRAMMIEREN	SACHSE	7	23
PROGRAMMIERG	DIRECTORY LESEN	WAGNER, G.	9	4
PROGRAMMIERG	EDITOR	WAGNER, G.	9	5
PROGRAMMIERG	TRICKS MIT STRINGS	WOLLSCHLAEGER	4	45
PROGRAMMIERG	PROGRAMMIEREN HEISST m. DATEIEN ARBEITEN	WOLLSCHLAEGER	7	29
PROGRAMMIERG	DEM INTERPRETER auf» BIT GESCHAUT	ZEITSCHRIFT	12	37
PROGRAMMIERG	DIE RICHTIGE DIMENSION	Zeitschrift	18	24
PROMMER	EPROM-Programmiergeraet	WOLF (aus mc)	5	?
PROZESSOR	Z80, HD64180 UND ILLEGALS	SCHROEDER	17	27
PROZESSOR	TAV's Z80-TUNING	ZEITSCHRIFT	17	33
RAM	512K-RAM-ADRESSEN m. FLEXIBLER MMU	BERNHARDT	18	67
RAM	EXTRA RAM ("Ein sicheres Plaetzchen...")	OBERMANN	9	63
RAMDISK	Die RAMDISK und du 11.Gebot	SCHMID, A.	25	54
RAMDISK	Die RAMDISK und das elfte Gebot	SCHMID, A.	25	56
RAMDISK	Darf's ein bisserl mehr sein?	SCHMID, A.	26	46
RAMDISK	MINI-RAM-FLOPPY im "SICHEREN PLAETZCHEN"	SOPP	10	37
RECHT	URHEBERRECHTSSCHUTZ fuer PROGRAMME	KAHLEN	10	55
RECHT	COMPUTERKRIMINALITAET		19	81
RECORDORGAN.	ZU GERALDs ARTIKEL ueber LMOFFSET	SOPP	10	3
RESTORE	BASIC-PATCH FUER RESTORE N	OBERMANN	5	?
ROM-ROUTINEN	CALLERLEI	MUEHLENBEIN	20	27
SCHALTALGEBR	KLEINES LOGIK-LEXIKON	aus c't	19	41
SCHALTSTATUS	WACHHUND, Der-, b.unkontr.Schaltzustand	BETZ	20	63
SCHNITTSTELL	Die Parallelschnittstelle	aus Comp.Pers.	27	49
SCHNITTSTELL	Entfernung der Kassettenschnittstelle	SCHROEDER, G.	26	40
SCHNITTSTELL	Memory mapped I/O <==> Port I/O bei GIIs	SCHROEDER, G.	26	42
SEMANTIK	TRAU - SCHAU, WEM! Ein Bitchen Englisch	MUEHLENBEIN	18	47
SENSOR	Digitales Temperatur-Zeit-Uhrenmodul	Obermann	16	41
SICHERHEIT	RAM-DISK KOSTENLOS	WOLLSCHLAEGER	2	?
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES	SOPP	13	23
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES (NACHTRAG)	SOPP	14	53
SORTIEREN	TRS80: Sortieren	aus mc	23	12
SORTIEREN	IN VISICALC SORTIEREN	MUEHLENBEIN	20	93
SORTIEREN	Ordnung musz sein !	MUEHLENBEIN	28	13
SPEICHER	Sicheres, Ein -, Plaetzchen	BERNHARDT	7	43
SPEICHER	CALL UM DIE ECKE (MEM-BEREICHE ERREICHEN	SOPP	15	49
SPIEL	Lottozahlen	MUELLER, Kurt	24	19
SPIEL	JOYSTICK AN MODEL 4/4p	OBERMANN	19	50

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
SPIEL	TAPE-GAMES AUF MODEL III/4/4p	OBERMANN	20	57
SPIEL	ADVEN-80	SCARGILL	11	9
SPIEL	DAS PIRATE ADVENTURE	SCHROEDER	11	15
SPIEL	ADVENTURE GEHEIMAGENT XP-05	WAGNER, ALEX.	3	?
SPRACHE	DER-DIE-DAS MACHT MIR SPASZ	MUEHLENBEIN	19	75
SPRACHE	Welche Sprache fuer wen ?	SCHMID, A.	27	59
SPRACHE	COBOL - EINE EINFUEHRUNG	TRAPP	5	?
SPRACHE	COBOL-EINFUEHRUNG		8	5
SPURANZAHL	80/4O-TRACKUMSCHALTUNG	KUHN	19	49
SYSTEM	Quatsch in OVL4/SYS	SOPP	24	27
SYSTEM	Schon Mieder SYS14/SYS	SOPP	24	20
TAKT	TAKTUMSCHALTUNG - DIE ALLERLETZTE LOESG.	MUELLER, KURT	14	45
TAPE	SUPERTAPE FUER TRS 80	BURGWITZ/STILLR	7	44
TASMON	ERWEITERUNG FUER TASMON (NEWDOS-BEFEHLE)	KONRAD J.	4	25
TASTATUR	UNIVERSALGENIE	aus c't	19	57
TASTATUR	Tastatur- und Drucker-Platine	DROWAELDER	20	71
TASTATUR	UMLAUTE IM NEWDOS? ABER NATUERLICH!	NEUEDEER	10	41
TASTATUR	KNOEPFE	OBERMANN	14	57
TASTATUR	ES GEHT NOCH SCHNELLER (TAST.ABSCHALTEN)	SCHMID, A.	22	32
TELEFON	TELEFONBUCH MIT PFIFF	FRISCH	2	?
TEXT-GRAFIK	(TELEFON) - EIN UD-ZEICHEN auf d.DRUCKER	SOPP	16	29
TEXTSYSTEM	KLEINES TEXTPROGRAMM	aus mc	23	10
TEXTSYSTEM	UMDEFINIERUNG v.SCRIPSIT-KOMMANDOTASTEN	KONRAD	8	14
TEXTSYSTEM	KEINE ANGST VOR... (TRENNEN IN TSCRIPS)	MUEHLENBEIN	20	61
TEXTSYSTEM	Was TSCRIPS nicht kann	MUEHLENBEIN	23	20
TEXTSYSTEM	TSCRIPS retten	MUEHLENBEIN	2B	3B
TEXTSYSTEM	DUMP (TSCRIPS retten)	MUELLER, Kurt	26	18
TEXTSYSTEM	SCRIPSIT AUF DEM ALTEN GENIE	OBERMANN	14	43
TEXTSYSTEM	Wie sag' ich's meinem WORDSTAR ?	SCHMID, A.	24	11
TEXTSYSTEM	WORDSTAR auf allen Laufwerken	SCHOBER	28	35
TEXTSYSTEM	AENDERUNGEN IN TSCRIPS und NAME/CMD	SCHROEDER	13	16
TEXTSYSTEM	FREIRAEUME IN TSCRIPS	SCHROEDER	14	47
TEXTSYSTEM	SCRIPSIT-TABELLEN	SCHROEDER	14	45
TEXTSYSTEM	TEXTUMLEITUNG fuer TSCRIPS	SCHROEDER	14	49
TEXTSYSTEM	TSCRIPS-VERSCHLUESSELUNG	SCHROEDER	15	29
TEXTSYSTEM	TIPS fuer SCRIPS	SOPP	13	19
TEXTSYSTEM	NEUES ZUR PROPORTIONALSCHRIFT	SOPP	18	15
TEXTSYSTEM	§ CONTRA .	SOPP	19	27
TEXTSYSTEM	TSCRIPS & DIE &-CODES	SOPP	19	16
TEXTSYSTEM	VON DER WIEGE BIS ZUR BAHRE (ADR.AUTOMAT	SOPP	19	33
TEXTSYSTEM	KEINE ANGST VOR PROPORTIONALSCHRIFT	SOPP	18	13
TEXTSYSTEM	TSCRIPS UND DIE 24 NADELN	SOPP	22	23
TEXTSYSTEM	DAS TSCRIPS-DRAMA - EIN LEHRSTUECK	SOPP/MUHLENBEIN	21	56
TONERZEUGUNG	DA CAPO... (WIE MAN TOENE ERZEUGT)	MUEHLENBEIN	22	13
TONERZEUGUNG	TONGENERATOR FUER TRS 80	PLUMHOFF	B	I9
TRS 80	DER BEKNACKTE TRS 80	ROECKRATH	6	33
TRS-80	TRS-80 Japan-Modell aufbohren	BERNHARDT	25	60
TRS-81	TRS-80, Der - und die Aussenwelt	KLEIN, WILFRIED	4	?

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
TRS-82	Umbau eines TRS-80 Mod.I 16K auf 48K	RETZLAFF	10	45
UHR	UHRUNTEN fuer MODEL 4p	OBERMANN	15	47
UHR	Es war einmal ein Datum	SCHMID, A.	26	29
UHR	WENN DIE UHR MAL STOERT	SOPP	14	39
UHR	THE TIMES, THEY ARE A-CHANGING	SOPP	22	35
UMLAUTE	Umlaute auf der Tastatur des GENIE I	DRECHSLER	28	69
UTILITY	PAGELIST/CMD	aus Zeitschrift	1B	21
UTILITY	EINGABEROUTINE MIT KOMFORT	BULLING	7	13
UTILITY	Startprogramm f. BASIC-Programme	DRECHSLER	28	21
UTILITY	SHELL/CMD	HERERMANN	24	34
UTILITY	TRACE-ON	HOEH (COMP.PERS	4	?
UTILITY	AUSGABEUMSCHALTUNG IN BASIC	KONRAD J.	6	24
UTILITY	ALLERLEI NUETZLICHES FUER DEN TRS 80	LICHTERMANN /MC	4	?
UTILITY	FILES WIE SAND AM MEER...	MUEHLENBEIN	11	7
UTILITY	DIRECTORY LESEN UND KOMMENTARE SPEICHERN	MUEHLENBEIN	13	15
UTILITY	FILEVERGLEICH (s.auch Progr.-Bibliothek)	NEUEDER	10	43
UTILITY	DDE St LWT	OBERMANN	10	19
UTILITY	LPRINT ALLES einfacher	OBERMANN	8	3
UTILITY	ZAP IM SUPERZAP/Zappen im Zeichenbereich	OBERMANN	8	13
UTILITY	LPRINT ALLES	SOPP	6	31
UTILITY	BILDSCHIRMAUSDRUCK BEIM TRS80+VIDEOGENIE	UEBERSCHAAR		
UTILITY	SMARTSCREEN, SMARTKEY und SMARTPRINT	ZEITSCHRIFT	12	45
VERTRETUNGEN	TRACE od.DIE GESCHICHTE EINER ZEITSCHR.	OBSCHERNINGKAT	18	75
VIDEO	Das 80-Zeichen-Karte-Problem u.d.Loesung	BERNHARDT	22	63
VIDEO	BO-Zeichen-Karte	SCHROEDER, G.	12	63
VIDEO	VIDEODAT-SYSTEM		1B	57
VISICALC	Korrigieren und Steuern innerhalb VC	MUEHLENBEIN	23	24
VISICALC	SORTICALC	MUEHLENBEIN	26	11
Z80	ARBEITSTIER MIT GESUNDEM HERZEN	BETZ	21	52
Z80	INTERFACE mit Kassetten-Schnittstelle	SCHROEDER, G.	26	40
Z81	422 NEUE Z80-BEFEHLE	HASSELBERG (MC)	4	?
ZAPPEN	EIN BYTE UND SEINE FOLGEN	OBERMANN	10	21
ZEICHENSATZ	Geheimnisse eines Zeichengenerators	GIESE	28	27
ZEICHENSATZ	NEUE ZEICHENSAETZE fuer das GENIE IIIs	SOPP	12	28
ZEITMESSUNG	ETWAS fuer LEUTE, die NIE ZEIT haben	SCHMID, A.	22	31
STICHWORT [1]	**Der hochgestellte Index gibt die Reihenfolge der Sortierung an**			
VERFASSER [2]				

CLUB 80 - Clubinfo Nr. 1

```
Kontaktadresse:    CLUB 80                          Juni 1984
---------------    Günther Wagner                   ---------
--------------     Gartenstraße 4
-------------      8201 Neubeuern
------------       Tel.08035/3361
-----------        (18 Uhr - 20 Uhr)
----------
---------
--------
-------
------
-----
---
--  Hallo Leute !
```

Jetzt habt Ihr das 1. Clubinfo in der Hand. Es erhalten alle Clubmitglieder und alle Interessenten, die sich noch nicht für den CLUB 80 entscheiden konnten (das Clubinfo soll die Entscheidung erleichtern).

Gleich vorneweg - das Clubinfo soll nicht durch das äußere Erscheinungsbild sondern durch den Inhalt bestimmt werden. Es ist also mehr oder weniger eine Loseblattsammlung und kein Magazin mit ausgefeilten Überschriften versehen mit Illustrationen usw. Der zeitliche Aufwand steht in keinem Verhältnis zum Nutzen. Ich hoffe, Ihr seid damit einverstanden.

Natürlich will ich das Clubinfo nicht alleine bestreiten - Ihr seid also zur Mitarbeit aufgerufen. Da das Clubinfo auch nicht in 1-2 Tagen sondern in ca. 2 Wochen nach und nach entsteht, fehlt teilweise der sog. rote Faden.

Natürlich freue ich mich über jede Kritik, jeden Verbesserungsvorschlag. Schreibt doch, was Ihr gut und was Ihr schlecht findet. Die Form des Clubinfo's muß im Laufe der Zeit gefunden werden.

In diesem 1. Clubinfo wird nun einmal auf 4 Seiten über die vorhandenen Programme informiert. Sie stammen fast ausschließlich von mir. Ich hoffe, daß auch alle anderen Clubmitglieder alle Ihre Programme beisteuern. Weiters werden die Programm-Namen erläutert. Zahlreiche Anfragen, viele Vorschläge und Denkanstöße usw. runden das Clubinfo ab.

Im nächsten Clubinfo, das Ende Juli erscheinen wird, werden u.a. alle Clubmitglieder vorgestellt (bisher sind es 5). Mich selbst darf ich als Gründer des Clubs auf der nächsten Seite vorstellen.

Nun aber genug für die 1. Seite!

Günther Wagner

Nun, ich heiße Günther Wagner und bin 1962 geboren. Nach einer Bauzeichner-Lehre studiere ich nun Bauingenieur.

Die ersten Begegnungen mit dem Computer hatte ich durch einen Kurs an der Volkshochschule. Meine Kenntnisse vertiefen konnte ich an der Fachhochschule (der gute alte PET 3001). Über ZX-81 kam ich dann endlich im September 1983 zu meinem jetzigen Computer - ein TRS-80 Modell III (deutsche Tastatur) mit 1 Laufwerk intern und 2 Laufwerken extern (80 Tracks, doppelseitig, doppelte Dichte). Ferner habe ich noch den LP VI von Tandy. An Betriebssystemen habe ich außer TRS-DOS nun seit einigen Monaten auch NEW-DOS. Basic beherrsche ich gut, Pascal und Maschinensprache leider bisher nur wenig. Aus Zeitgründen programmiere ich zur Zeit nur sehr wenig selbst.

Kurz nach dem Computer-Kauf im September machte ich mich auf die Suche nach einem Anwender-Club. Ich fand leider keinen guten bzw. seriösen Club mit annehmbarer Clubsatzung (durch den ZX-81 kannte ich schon einige sog. 'USER-CLUBS'). Deshalb beschloß ich, selbst einen Club zu gründen. Ich war dabei so gut wie auf mich selbst gestellt. Nun, man kann sagen der Club existiert seit Mai 1984 und ich rechne, daß es bis in einem Jahr mit dem Club sehr gut aussieht.

Ich hoffe, daß ich durch den CLUB 80 meine eigene Programmbibliothek ausbauen kann. Ferner kann ich mir aus Computer-Büchern das für mich wichtige raussuchen. Wenn ich Fragen habe, kann ich diese im Clubinfo stellen oder jemanden anrufen. Wenn ich ein Programm geschrieben habe, kann ich es im CLUB z.B. auch testen lassen. So ähnliche Erwartungen werden wohl auch die meisten anderen Mitglieder haben. Mir ist dabei klar, daß der Club mit der Mitarbeit der Mitglieder steht und fällt.

--- club 80 --- club 80 --- club 80 --- club 80 --- club 80 ---

---

So, nach dieser kurzen Vorstellung möchte ich einen ganz kurzen Bericht über den derzeitigen Kassenstand geben.

An Einnahmen haben wir aus Beitragszahlungen bisher 120.00 DM eingenommen.
Dem stehen Ausgaben in Höhe von 120.00 DM gegenüber (Photokopien 20 DM, Briefmarken 70 DM und Anzeige in der Chip 30 DM).
Bei den Ausgaben kommen nun noch die Kosten für 2 Disketten und dieses Clubinfo hinzu.

Wir haben also z.Z. ein kleines Minus in der Kasse - durch weitere Clubbeitritte ist dies aber bald behoben.

P R O G R A M M E

Für viele Mitglieder am interessantesten dürfte die Programmbibliothek sein. Damit wir hier was aufbauen können ist unbedingt die Bereitschaft eines jeden nötig, Programme beizusteuern.

Damit alle Mitglieder Interesse haben, Ihre Programme beizusteuern, habe ich mir das Punkte-System überlegt. So erhält man größere Mengen von Programmen nur kostenlos, wenn man Programme eingesandt hat.

Die Programme sollten auf Diskette oder Kassette mit Programm-Listing eingesandt werden. Bitte immer ganz genau angeben, unter welchem Namen die Programme gespeichert sind. Bei Disketten nach Möglichkeit auf System-Diskette einsenden mit genauer Angabe einseitig oder doppelseitig, einfache oder doppelte Dichte, verwendetes Betriebssystem und Anzahl der Spuren. Nach Möglichkeit immer ausführliche Programm-Dokumentation beilegen. Die Diskette bzw. Kassette erhalten Sie wieder zurück (bei gleichzeitiger Programmbestellung bitte angeben, ob Diskette überspielt werden darf!).

Bei Programmbestellung müssen Sie die Diskette oder Kassette der Bestellung beilegen. Bei Diskette müssen Sie wieder die genauen Angaben über verwendetes Laufwerk und Betriebssystem machen. Die gewünschten Programme bestellen Sie durch Angabe des Programm-Namens (z.B. S0012G03/BAS) - wenn Sie wollen, können Sie den Programm-Titel mitangeben.

Dringend benötigt wird ein gutes Adressenverwaltungsprogramm. Ein wirklich gutes habe ich noch nicht, und ich würde die Mitgliederspeicherung am liebsten natürlich gleich auf einem guten vornehmen. Vielleicht hat jemand ein gutes.

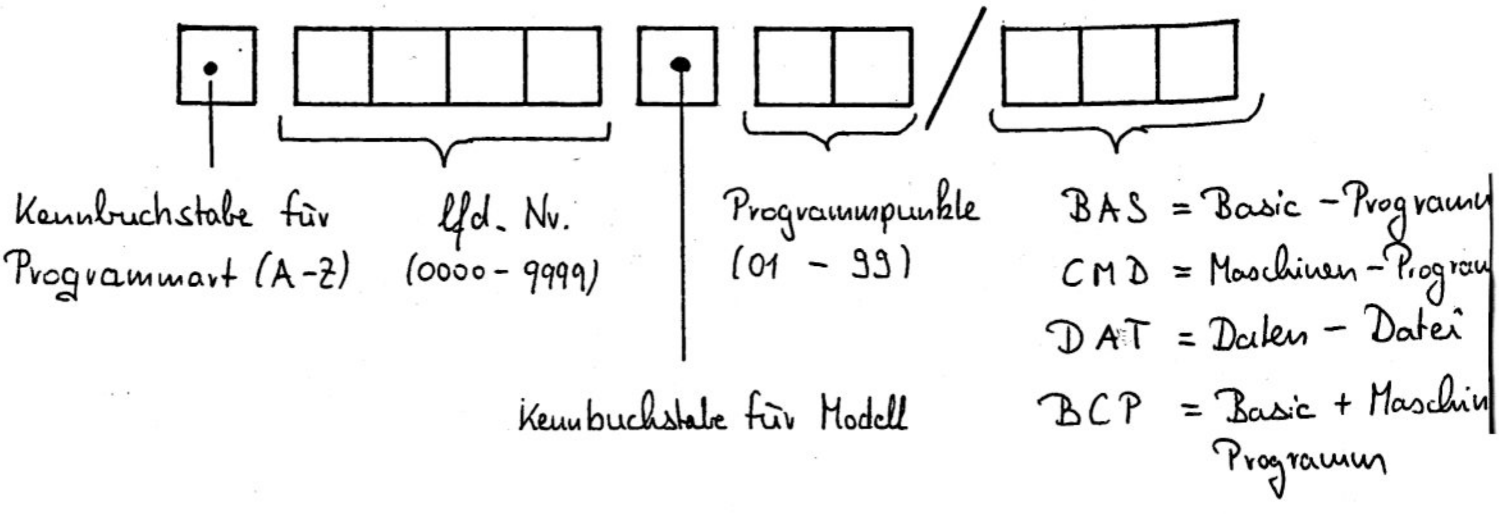
Wenn Ihr jetzt die Programm-Liste durchseht, werdet Ihr feststellen, daß Ihr das eine oder andere Programm bereits habt. Wenn es aber eine andere Version des gleichen Programms sein könnte, dann sendet diese auch ein - vielleich ist diese besser. Grundsätzlich ist jedes noch so kleine Programm von Interesse. Programmpunkte erhaltet Ihr allemal.

Nun, ich hoffe, im nächsten Clubinfo viele neue Programme vorstellen zu können.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Wer Fragen hat, oder wer für das Clubinfo einen Beitrag irgendwelcher Form hat, der sollte mir das bald genug zusenden. Hoffentlich quillt mein Postkasten bald über.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



Vorläufige Belegung der Kennbuchstaben für:

Programmart:

A = Adressen- und allgemeine Dateiverwaltung
D = Denkspiel
G = Geschicklichkeitsspiel
K = Karten- oder Würfelspiel
M = Mathematik und ähnliches
S = Sonstiges
T = Textverarbeitung
Z = Zeichentrick und ähnliches

Modell:

A = Model I, III, IV
B = nur Model I
C = nur Model III
D = nur Model IV

Diese Kenbuchstaben werden bei Bedarf erweitert.

Bei der Programmliste steht neben dem Programm-Titel immer der Programm-Name nach diesem Muster. Unter diesem Programm-Namen sind alle Programme in der Datei gespeichert. Die lfd. Nummer wird dabei für ein Programm nie verändert; die anderen Kennbuchstaben können bei Bedarf geändert werden. Diese Programm-Namen haben den Vorteil, daß man daran bereits die wichtigsten Informationen erhält. Der Nachteil ist, daß man erst in der Liste nachschauen muß, um welches Programm es sich handelt. Aber Sie können sich ja zu Hause das jeweilige Programm unter dem Namen den Sie wollen abspeichern.

Wenn natürlich jemand hierzu einen besseren Vorschlag hat, soll er mir schreiben.

Im Club sollten Entscheidungen über irgendwelche Sachen soweit möglich durch die Mitglieder getroffen werden (einfache Mehrheit der eingegangenen Stellungnahmen entscheidet).

1. Weitere Mitglieder-Werbung ist besonders am Anfang notwendig. Macht Vorschläge, wie das bewerkstelligt werden soll (beiliegender farbiger Aushang ist gedacht zum aufhängen an Schulen, Uni's oder sonstigen guten Plätzen). Mein Vorschlag - auf alle Fälle nochmal eine Anzeige in der CHIP (Ja oder nein).

2. Was soll vom ersten größeren Plus in der Club-Kasse angeschafft werden? Bücher (bisher noch keine vorhanden), Programme oder Sonstiges. Bitte mit Vorschlag (z.B. welche Bücher angeschafft werden sollen).

Schreibt doch zu diesen beiden Punkten sobald wie möglich Euere Meinung.

*** Vorschläge *** Vorschläge *** Vorschläge *** Vorschläge ***

Die nachfolgenden Vorschläge sollten Denkanstöße und Diskussionsgrundlage sein, bitte geht darauf ein.

Jedes Mitglied könnte aus Computerzeitungen und Bücher Photokopien (den Club betreffend; z.B. Programm-Listings, Berichte, etc.) machen und an den Club senden. Diese werden im Clubinfo veröffentlicht, so daß jedes Mitglied auf diesem Weg viele Informationen erhält. Was haltet Ihr davon, wie sieht es mit der rechtlichen Seite aus (Nachdruck)?

Für Clubmitglieder wäre es meiner Meinung nach sehr interessant, miteinander über Akustik-Koppler (über Telefon) z.B. Programme tauschen zu können. Man braucht dazu einen Akustik-Koppler (ca. 300 DM bei Tandy). Dieser wird an der seriellen Schnittstelle angeschlossen (ca. 300 DM). Wer hat auf diesem Gebiet Erfahrung? Geht das ohne Probleme? Wie geht's billiger? Welche Vorteile hat man noch? Wer wäre daran interessiert?

FRAGENFRAGENFRAGENFRAGENFRAGENFRAGENFRAGENFRAGENFRAGENFRAGEN

Meine Computer-Anlage habe ich bereits erläutert.

Nun einige Fragen:

Wer weiß, wie ich unter NEWDOS Model I - Disketten lesen oder kopieren kann?

Wer weiß, wie ich Programme von NEWDOS-Disketten auf TRSDOS-Disketten kopieren kann?

Wer hat Tandy's Superscripsit unter NEWDOS lauffähig?

Unter NEWDOS kann ich zwar deutsche Umlaute ganz normal auf dem Bildschirn darstellen. Leider lassen sich diese nicht ausdrucken? Wer weis Abhilfe?

Wer kann mir, Günther Wagner, ein günstiges Laufwerk (35 Spuren, einfache Dichte, einseitig) anbieten?

****

Durch die Veröffentlichung der Club-Gründung bekomme ich sehr viele Angebote und Prospekte. Teilweise sind diese recht günstig. So wurden mir ein Matrixdrucker (DRH 80 von TA) für 700 DM und ein Typenraddrucker von sehr guter Qualität und großer Vielseitigkeit (TRD 7020 von TA) für 1500 DM angeboten.

Leider kann ich unmöglich alle Angebote im Clubinfo veröffentlichen - es werden ja auch die Prospekte nur einzeln zugesandt. Ich werde allerdings auf besonders interessante Angebote hinweisen.

Wer irgendetwas kaufen möchte und nähere Informationen etc. sucht, der kann ja bei mir nachfragen, ob ich in der Richtung etwas in meinen Akten habe. Falls ja, bekommt er die entsprechenden Unterlagen zugesandt - diese müssen aber dann wieder zurückgesandt werden. Ich glaube, das ist die beste Lösung.

H I N W E I S :

Sollte das nächste Clubinfo nicht bis Ende Juli bei Ihnen auftauchen, dann habe ich die Fertigstellung vor meinem Urlaub nicht geschafft. Es erscheint dann erst Ende August.

Das 2. Clubinfo erhalten dann auch nur noch CLUB 80 - Mitglieder (also alle Nichtmitglieder - schnell das Aufnahmeformular zusenden!!!).

Bitte werdet nicht ungeduldig, wenn mal die Beantwortung oder Bearbeitung von irgendetwas länger dauert; oft habe ich halt erst nach einer Woche dazu Zeit.

Bitte macht mit bei der Gestaltung des CLUB 80: Sendet Euere Programme ein, bantwortet die Fragen, stellt Fragen wenn Ihr welche habt und schickt Beiträge zum Clubinfo zu, falls Ihr da was liefern könnt.

Viele Grüße bis zum nächsten Mal

Euer Günther Wagner

DASWARDIELETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZTESEITEDAS
WARDIELETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZTESEITDASWARD
IELETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZTESEITEDASWARDIEL
ETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZ
TESEITEDASWARDIELETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZTES
EITEDASWARDIELETZTESEITEDASWARDIELETZTESEITEDASWARDIELETZTESEIT
EDASWARDIELETZTESEITDASWARDIELETZTESEITEDASWARDIELETZTESEITE!!!

CLUB 80 - Clubinfo Nr.2

Juli 1984

Kontaktadresse: CLUB 80
*************** Günther Wagner
************** Gartenstraße 4
************* 8201 Neubeuern
************ Tel.08035/3361
*********** (18 Uhr - 20 Uhr)
****
****
****
****
******
*****
****
***
** Hallo Freunde !

Vor der Zusammenstellung dieses 2. Clubinfos hatte ich zunächst tiefgreifende überlegungen angestellt - sollte ich alles, was ich an Material für's Info habe auf einmal verarbeiten oder sollte ich einen Teil für's nächste Info aufheben.

Am Umfang seht Ihr, daß ich das ganze Material in dieses Clubinfo gepackt habe - ich werde für's nächste Info schon wieder Material von Euch bekommen.

Euch erwartet nun in diesem Info die Mitgliederliste (mit 2 Vorstellungen), eine Erweiterung der Softwareliste um 70 Programme, eine Bücherliste, Rückblicke auf unser 1. Clubinfo, ein Bericht über SUPERDOS und noch vieles mehr.

Die Mitarbeit von Euerer Seite funktioniert bis jetzt recht gut (von den meisten habe ich was bekommen). Am meisten überraschte mich eine Lieferung von Hartmut Obermann, die ich bald nach dem Info von Ihm erhielt. Neben vielen schönen Programmen befand sich auf der Diskette auch ein Programm für die Adressenverwaltung für unseren CLUB 80. Dieses Programm hatte er eigens für den Club geschrieben. Nochmals vielen Dank an die Adresse von Hartmut Obermann, der auch den Artikel SUPERDOS für dieses Info zur Verfügung stellte. Vielen Dank an alle Mitglieder für Ihre Tips, Anregungen und Kritik und für die vielen Programme (die Archivierung der Programme war ein ganz schönes Stück Arbeit).

Die Mitgliederzahl ist seit dem letzten Info um sagenhafte

*** + 140 % ***

gestiegen - etwas bescheidener könnte man auch sagen, der Club hat 7 Neuzugänge erhalten, so daß wir jetzt das Dutzend voll haben. Hoffen wir, daß es so weiter geht.

Die Form des Clubinfo's wurde allgemein befürwortet - niemand beanstandete meine Rechtschreibfehler. Ich werde deshalb in gleicher Weise mit dem Clubinfo fortfahren und hoffe, daß Euch diese Ausgabe gefällt.

Bis zum nächsten Mal viele Grüße,

Euer Günther

Zum Thema B Ü C H E R

Also - vom ersten größeren Plus in unserer Club-Kasse werden Bücher angeschafft. Dies hat die Reaktion der einzelnen Mitglieder auf meine Frage ergeben. Im nächsten Clubinfo werden Buchvorschläge veröffentlicht - bis dahin könnt Ihr also noch Vorschläge zusenden.

Zum Thema Bücher gab es aber noch einige interessante Ideen und Anregungen, die ich gerne aufgreifen möchte.

Herr Zwickel meint, die Literatur muß nicht über den CLUB gekauft werden. Jeder hat einiges an Literatur herumliegen. Wenn jedes Mitglied alle seine Bücher im Clubinfo bekanntgibt, dann kann sich ein Interessent das Buch 4 Wochen direkt ausborgen. Rücksendung ist in unserem (noch) kleinem Club ja wohl Ehrensache. Bei der Rücksendung kann der Ausleiher einen Scheck für die Portokosten des Besitzers beilegen.

Wir machen das jetzt ganz einfach - wer mit diesem Vorschlag von Herrn Zwickel einverstanden ist, soll mir seine Bücherliste zusenden.

Der nächste Vorschlag stammt von Herrn König, der der Meinung ist, daß der Grundstock zur Bibliothek von den Mitgliedern selbst gelegt werden sollte. Dazu müßten die Mitglieder bereit sein, Bücher die sie selten nutzen dem Club zu spenden oder zu leihen. Herr König hat gleich den Anfang gemacht, indem er das Buch 'BASIC-Brevier' gespendet hat.

Nun, ich finde den Vorschlag sehr gut und habe deshalb auch einige Bücher dem Club gespendet - unsere kleine Bibliothek umfaßt jetzt 5 Bücher. Bestimmt ist der eine oder andere auch zu einer Spende bereit.

Ein weiterer Vorschlag stammt von Herrn Obermann. Er wäre dafür, Bücher (von Mitgliedern) zu photokopieren. Dies rentiert sich bei schwer erhältlichen Büchern und bei Büchern, wo der Photokopierpreis unter dem Kaufpreis liegt. Die Bücher könnten die Mitglieder selbst kopieren und mir die Photokopien zusenden (gegen Erstattung sämtlicher Unkosten - gute Qualität der Photokopien nötig!), oder die Mitglieder könnten mir die Bücher zum photokopieren zusenden (Portokosten werden auf Verlangen erstattet - mich kostet eine Photokopie 8 bis 10 Pfennig).

Was meint Ihr dazu! Sollten wir die Bücher-Bibliothek zum Teil mit photokopierten Büchern bestreiten. Schreibt Euere Meinung zu diesem und auch zu den anderen Vorschlägen. Danke!

Änderung der Clubsatzung

Ich, Günther Wagner, beantrage eine Änderung der Clubsatzung (s. auch Nr. 11 unserer Clubsatzung). Für die angestrebte Änderung ist eine 3/4 Mehrheit der abgegebenen Stimmen erforderlich (also auch alle die dagegen sind müssen das schriftlich mitteilen). Sendet beiliegenden Stimmzettel bitte bis spätestens 1. September an mich zurück.

Die Änderung betrifft Nr. 4 unserer Satzung:

Es heißt: "Das Buch wird auf Club-Kosten versandt".

Angestrebte Änderung: "Das Buch wird auf Kosten des Entleihers versandt".

Begründung: Die Porto-Kosten für den Versand der Bücher würden die Clubkasse sehr belasten, da ja die meisten Bücher ziemlich schwer sind und auch nicht als Büchersendung versandt werden können. Durch die Änderung könnten ev. mehr Bücher angeschafft werden.

Bücherliste

Nr. 0001: BASIC-Brevier: Eine Einf. in d. Prog. v. Heimcomp.

Siegmar Wittig --- Heinz Heise GmbH

Ein BASIC-Kurs fuer Nicht-Mathematiker und echte Amateure. Neben Grund- und Aufbaukurs gibt es zahlreiche Beispiele (eine gute Sammlung!).

Nr. 0002: Computerwissen

Michael Scharfenberger --- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von Hard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knobeleien programm. in BASIC

Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

\- --- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel --- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Mitglieder CLUB 80 Mitglieder

Die Liste der Mitglieder wird nachstehend abgedruckt. Dadurch ist es Euch möglich, auch untereinander Kontakt aufzunehmen. Bitte denkt daran, daß interessante Sachen ins Clubinfo gehören.

A C H T U N G : Bitte prüft die Daten Eueres Eintrages (bes. die Punkte) und teilt mir ev. falsche Daten mit. D a n k e !

Nach der Mitgliederliste werden zwei persönliche Vorstellungen abgedruckt - wer sich von den anderen Mitgliedern noch persönlich vorstellen will, soll mir entsprechendes zusenden.

Obermann Hartmut
Schwalbacher Str. 6
6209 Heidenrod/Kemel
06124/3913
Clubmitglied seit 26.04.84
Beitrag bezahlt?   ja
Punktestand : 407
System- und Drivekonfiguration :
TRS-80 Model I mit OKIDATA Microline 80 Matrixdrucker 3 Laufwerke (
je 40 Tracks/ ss/ sd)
bevorzugtes Betriebssystem : NEWDOS

Schroeder Gerald
Am Schuetzenplatz 14
2105 Seevetal 1
04105/2602
Clubmitglied seit 12.06.84
Beitrag bezahlt?   ja
Punktestand : 134
System- und Drivekonfiguration :
Genie I/ Expander/ EG64/ Drucker Gemini 10x 2 Laufwerke (Je 39 Spur
en/ ss/ dd)
bevorzugtes Betriebssystem : NEWDOS und SUPERDOS

Stevens Peter
Postfach 6327
7800 Freiburg
0761/35384
Clubmitglied seit 15.06.84
Beitrag bezahlt?   ja
Punktestand :   60
System- und Drivekonfiguration :
TRS-80 Model III (48K)/ Epson RX80 2 Laufwerke (je 40 Spuren/ ss/ d
d)
bevorzugtes Betriebssystem : ?

Troesch Eberhard
Altenessener Str. 414
4300 Essen 12
0201/342324
Clubmitglied seit 28.05.84
Beitrag bezahlt?   ja
Punktestand : 25
System- und Drivekonfiguration :
TRS-80 Model III (48K)/ Epson RX 2 Laufwerke (je 40 Spuren/ ss/ dd)
bevorzugtes Betriebssystem : TRS-DOS

Kasper Dieter
Zeppelinstr. 9
8952 Marktoberdorf
08342/1630
Clubmitglied seit 25.06.84
Beitrag bezahlt?   ja
Punktestand : 25
System- und Drivekonfiguration :
Video Genie I (83'er Modell - 64K) -
bevorzugtes Betriebssystem : -

Koenig Hans J.
Hebbelstr. 25
2080 Pinneberg
04101/209444
Clubmitglied seit 07.06.84
Beitrag bezahlt?   ja
Punktestand : 95
System- und Drivekonfiguration :
TRS-80 Model I Level II (48K)/ Epson RX-80/ 2 Laufwerke
bevorzugtes Betriebssystem : NEWDOS und G-DOS

Konrad Josef
Anzengruber 35
8038 Groebenzell
08142/8494
Clubmitglied seit 27.06.84
Beitrag bezahlt?   ja
Punktestand : 25
System- und Drivekonfiguration :
TRS-80 Model I (48K)/ Nadeldrucker DRH 80 2 Laufwerke (ss/ sd)
bevorzugtes Betriebssystem : ?

Neueder Jens
Panoramastr. 21
7178 Michelbach/Bilz
0791/44-450 (geschaeftlich)
Clubmitglied seit 11.05.84
Beitrag bezahlt?   ja
Punktestand : 437
System- und Drivekonfiguration :
TRS-80 Model I Level II/ 48K (Expansion)/ NEC 8023-B-C (Drucker) 2
Laufwerke (je 80 Spuren/ ds/ sd)
bevorzugtes Betriebssystem : ?

Wagner Guenther
Gartenstr. 4
8201 Neubeuern
08035/3361
Clubmitglied seit 01.03.84
Beitrag bezahlt? ja
Punktestand : -
System- und Drivekonfiguration :
TRS-80 Model III mit LP VI (Punktmatrixdrucker) System-Laufwerk (40 Spuren/ ss/ dd) und 2 externe Laufwerke (je 80 Spuren/ ds/ dd)
bevorzugtes Betriebssystem : NEWDOS (und TRSDOS)

Weiser Wolfgang
Behaimstr. 12
1000 Berlin 10
030/3421448
Clubmitglied seit 30.04.84
Beitrag bezahlt? ja
Punktestand : 25
System- und Drivekonfiguration :
TRS-80 Color-Computer 2/ Joysticks Kassettenrekorder
bevorzugtes Betriebssystem : -

Wright Donald
Apothekenweg 5a
8056 Neufahrn
-
Clubmitglied seit 17.05.84
Beitrag bezahlt? ja
Punktestand : 25
System- und Drivekonfiguration :
TRS-80 Model I Level II/ Quick Printer/ Bus Monitor (selbst gebaut) 2 Laufwerke und Kassettenrekorder
bevorzugtes Betriebssystem : ?

Zwickel Walter
Lengfelden 123
A - 5101 Bergheim
0043662/51130
Clubmitglied seit 06.06.84
Beitrag bezahlt? ja
Punktestand : 317
System- und Drivekonfiguration :
Komtek 1 mit 64 K (wie TRS-80 Model I)/ Star Gemini X10/ Eprommer/ Parallel Ports PIO/ Serielle Schnittstelle SIO 1 Laufwerk
bevorzugtes Betriebssystem : NEWDOS

Mein Name ist Jens Neueder, und ich bin als Fernmeldehandwerker beim Fernmeldeamt Schwäbisch Hall tätig. Geboren wurde ich im November 1959 in Schwäbisch Hall.
Meinen TRS-80 kaufte ich mir 1981 als Grundgerät mit 16K und 10-er Tastatur zum Anschluß an ein Fernsehgerät. Später folgte dann eine Stringy Floppy und ein Monitor. Die Speichererweiterung durch eine Expansionsplatine baute ich dann ein Jahr darauf selbst auf. Danach kam als Drucker ein NEC PC-8023B-C, der mit den Itoh-Druckern vergleichbar ist. Die vorletzte Anschaffung waren dann die zwei 80-Track Floppylaufwerke mit doppelten Köpfen. Zur Zeit bastle ich an einer hochauflösenden Grafikplatine von der Firma RB-Elektronik. Meine Programmiertätigkeit ist hauptsächlich in Basic. Mit der Maschinensprache möchte ich mich in der nächsten Zeit etwas anfreunden.
Durch den CLUB 80 hoffe ich nun, daß ich im Tausch an verschiedene interessante Programme für Grafik sowie für den Drucker komme. Auch bei eventuellen Fragen Hilfe und Tips aus der Clubgemeinschaft zu bekommen, würde mich sehr freuen.

Mein Name ist Hartmut Obermann, geboren am 24.08.1958. Nach Abschluß einer Berufsfachschule Elektrotechnik und einem Berufsabschluß als Informationselektroniker bin ich seit nunmehr sechs Jahren bei der Bundeswehr (Luftwaffe) als Radarmechanikermeister tätig.
Meine ersten Schritte auf dem Computer habe ich während eines Elektronik-Weiterbildungskurses gmacht (technischer Aufbau CPU 8080, Maschinensprache und Assemblerprogrammierung). Dies war 1980.
Meinen TRS-80 Model I Level 2 habe ich mir dann 1982 aus Amerika mitgebracht. Im Laufe der Zeit habe ich das System auf 48k Speicherplatz und drei 5 1/4" Diskettenlaufwerke ausgebaut. Bis vor kurzem arbeitete ich ausschließlich mit TRSDOS 2.3. Jetzt besitze ich auch NEWDOS und SUPERDOS.

??? Fragen ???

Wer kennt eine akzeptable Möglichkeit, zwei zusätzliche Indexlöcher zu machen, um die Disketten doppelseitig nutzen zu können? (Gerald Schröder)

Wer hat Erfahrung mit der RS232-Schnittstelle für das Modell I? Wer weiß wo ic diese Schnittstelle günstig beziehen kann oder hat einen Schaltplan für eine solche Schnittstelle? (Hartmut Obermann)

Wer hat seinen TRS-80 Model I selbst mit einem DD-Controller ausgerüstet und kann mir Bezugsquellen nennen und Tips geben? (Hartmut Obermann)

Wer hat einen Schaltplan für den TRS-80 Model I und das Expention-Interface? (Hartmut Obermann)

------ S U P E R D O S ------

Ein Erfahrungsbericht von Hartmut Obermann:

Vor kurzem habe ich unserem Freund Günther Wagner mitgeteilt, daß ich in meiner Programmbibliothek ein SUPERDOS genanntes Diskettenbetriebssystem habe. Daraufhin bat er mich, einen kurzen Erfahrungsbericht für unser Clubinfo zu verfassen. Dieser Bitte komme ich hiermit nach.

SUPERDOS ist ein Diskettenbetriebssystem von J. Wulff. Bei der ersten Inbetriebnahme fällt sofort die Ähnlichkeit mit NEWDOS80 auf. Im Endeffekt ist das eigentliche DOS auch nichts anderes als das bekannte NEWDOS, jedoch mit deutschen Meldungen. Steht bei NEWDOS z.B. "Device not available", wenn man einen nicht vorhandenen Diskdrive anspricht, so bringt SUPERDOS die Fehlermeldung "Bauteil nicht erreichbar". Die deutschen Fehlermeldungen erstrecken sich auch auf das Diskbasic von SUPERDOS. Diese Art der Fehlermeldung erscheinem einem geübten NEWDOS-Benutzer im ersten Augenblick sehr ungewohnt,(z.B.:Prüfzahlfehler beim Lesen) erweisen sich aber für einen Computerlaien (z.B. meine Frau) als durchaus nützlich.

Dies vor allem, da auch die sog. START-File, welche durch das Kommando AUTO LIST START bei jedem Neuinitialisieren auf dem Bildschirm erscheinen, in Deutsch geschrieben sind. Diese Files sind nichts anderes als eine Auflistung der im SUPERDOS untergebrachten Utilitys. Und hilft selbst dem unerfahrensten Operator beim Aufruf der benötigten Hilfsprogramme.

Als solche sind im SUPERDOS eingebaut:

1. Ein "Adapter Tape/Disk Disk/Tape" der sehr dem bekannten LMOFFSET ähnelt, jedoch zusätzlich die Möglichkeit enthält, die Adressen der zu übertragenden Programme auf dem Drucker auszugeben. Ein weiterer Vorteil besteht darin, daß nicht nur Start-, End- und Einsprungadresse sondern zusätzlich die Blockladeadressen angezeigt werden.

2. Ein Bildschirm-Editor, der es erlaubt, ähnlich wie z.B. auf Commodore-Computern Basicprogramme bildschirmorientiert zu editieren. Der Umgang mit diesem Modul will jedoch gelernt sein. Allzu schnell kann man, an den Zeileneditor gewohnt, sein komplettes Programm verhunzen. Der normale Zeileneditor bleibt übrigens erhalten und kann weiterhin benutzt werden.

3. Eine Chain-File-Erstellungshilfe, wie es sie ähnlich schon mehrfach gibt (z.B.:JOB/CMD oder CHAINBLD/CMD). Eine Besonderheit enthält dieses Programm allerdings. Aber leider keine positive. Wenn der Chainfile gespeichert werden soll, ist die Angabe eines Ziel-Drives nicht möglich und das Programm wird irgendwo (ich arbeite mit 3 Drives) auf den nächsten freien Speicherplatz geschrieben.

4. Ein File-Retter, der, bis auf die deutschen Meldungen natürlich, dem Programm RESCUE/CMD entspricht das vor kurzem in der CHIP abgedruckt war. Mir ist übrigens die Funktion dieses Utility-Programms immer noch nicht geheuer. Bei Versuchen damit ist mir aufgefallen, daß sich bei einem Aufruf des Programms direkt nach dem Löschen der Files diese nicht mehr wiederherstellen liesen, obwohl noch kein Zugriff auf die entsprechende Diskette erfolgt war.

5. Ein Dircheck-Programm, welches auch unter NEWDOS80 Verwendung findet.

6. Ein Disketten-Katalog, bekannt unter dem Namen INDEX/CMD. Dieses Programm ist sehr nützlich, da man sich in kürzester Zeit einen guten Überblick über seine Disketten verschaffen kann. Auch dieses Programm ist natürlich mit deutscher Anleitung und entsprechenden Fehlermeldungen ausgerüstet.

7. Ein MINI-SUPERZAP genanntes Programm, welches in Anlehnung an die großen ZAP-Programme die Anzeige des Inhalts eines Files und deren Änderung ermoeglicht.

8. Schon erheblich nützlicher ist der ebenfalls eingebaute Printer-Buffer. Dieses Programm reserviert einen festlegbaren Teil des RAM als Buffer für die Printerausgabe. Programme mit häufiger Druckerausgabe können so erhebich schneller ablaufen.

9. Ein sog. Supercopierer, mit dem man (fast) alle Disketten kopieren kann. Ich selbst habe damit schon TI 99/4A- und Commdore-Disketten erfolgreich (d.h. lauffaehig) kopiert. Die Sache hat jedoch den Haken, daß man zwei Drives benötigt. Im übrigen ist auch dieses Programm bekannt (COPYCAT/CMD).

10. Ein weiteres, sehr empfehlenswertes Hilfsmittel ist das eingebaute Tasten-Shorthand-Programm. Es belegt alle Tasten mit BASIC-Befehlen, welche mit der Tastenfolge "CLEAR" + Buchstabe aufgerufen werden. Auch dieses Programm ist nicht neu. Das Programm KEY48/CMD hat die gleiche Funktion. Einen bedauerlichen Umstand möchte ich im Bezug auf dieses sonst so nützliche Utility noch anführen. Es läuft leider nicht mit dem schon angesprochenen Bildschirmeditor, obwohl dies wünschenswert wäre.

11. Ein Utility-Pack mit folgenden Unterprogrammen:
   - -Auflisten (auch auf den Drucker) der Systemparameter.
   - -Aendern der Systemparameter (komfortabel).
   - -Auflisten der PDRIVE-Parameter (Drucker).
   - -Aenderungs-Routine für die PDRIVE-Parameter (weniger komfortabel).
   - -Eine kurze Inforoutine, die über den druckbaren Zeichensatz, die angeschlossenen Drives und anderes informiert.
   - -Eine komfortable Routine zum Einstellen der Druckparameter von Itoh-Druckern (wohl dem der einen solchen sein eigen nennt).

12. Last not least ein Wahlprogramm mit dem auf einen Tastendruck entweder Programme aufgerufen oder DOS-Kommandos ausgeführt werden können. Man erspart sich damit viel Tipperei, vor allem bei den teilsweise ja sehr langen COPY- Befehlen.

Alle angesprochenen Utilitys sind natürlich nicht neu. Jedes dieser Programme gibt es einzeln für die verschiedenen Betriebssysteme und SUPERDOS ist im Endeffekt nur (?) eine Zusammenziehung nützlicher Unterroutinen zu einem eine Einheit bildenden Betriebssystem. Trotzdem, oder vielleicht gerade deswegen lohnt sich die Anschaffung dieses DOS. Alle hier enthaltenen Utilitys einzeln zu erwerben kostet entweder viel Zeit (Tausch) oder eine Menge Geld (Kauf). So bietet z.B. der bekannte (berüchtigte) Softwareverkäufer Hanke (dies soll keine Schleichwerbung sein) das SUPERDOS für 50.00 DM an. Gleichzeitig kann man über ihn auch die Utilitys einzeln beziehen. Man kommt dann jedoch auf einen Preis von über 100.00 DM (in Worten: einhundert). Ob sich die Anschaffung lohnt, hängt also davon ab, wieviele der angesprochenen Utilitys man schon besitzt.Ganz kurz möchte ich noch das im SUPERDOS enthaltene BASIC ansprechen. Es hat zusätzlich zum normalen Basic, wie es z.B. NEWDOS80 enthält, zwei weitere Befehle. Und zwar den Line- und den Name-Befehl. Mit dem Line-Befehl können sehr schnell Linien, Rahmen, ausgefüllte Rechtecke usw. erzeugt werden. Der Befehl Name bewirkt ein formatiertes Listing von Basicprogrammen, welches die übersichtlichkeit erheblich erhöht. Mit einem ":" als Zusatz kann auch auf dem Drucker formatiert gelistet werden.

Ich hoffe, dieser kleine Bericht hat Euch interessiert und so informiert, daß Ihr in der Lage seid, zu entscheiden, ob sich die Anschaffung von SUPERDOS lohnt.
(PS. Die Adresse von Hartmut Obermann könnt Ihr dem Clubinfo entnehmen.)

Zum 1. Vorschlag der letzten Ausgabe ...

Zum ersten Vorschlag aus dem Clubinfo Nr. 1 gibt es folgendes Hauptproblem: Wie kann man es vermeiden, daß der Club Listings und Berichte doppelt erhält? Ansonsten waren eigentlich alle für den Vorschlag.

Nun, nachstehend folgt eine Liste der Computer-Zeitungen, welche ich bereits besitze. Ich werde also im Laufe der Zeit sämtliche Berichte und Listings, welche für den Club von Interesse sind, zweimal kopieren. Zweimal deswegen, weil (wie auch bei Disketten üblich und wichtig) eine Original-Sammlung und eine Verleih-Sammlung angelegt wird. Die Verleih-Sammlung kann von Mitgliedern ausgeliehen werden. Diese wird unterteilt in verschiedene Bereiche (z.B. Adventures, Dateiverwaltung, Utilitis, etc). Kurzbeschreibungen der vorhandenen Berichte und Listings werden im Clubinfo veröffentlicht. Die Portokosten für die Zusendung der einzelnen Verleih-Sammlungen muß aber der Ausleiher bestreiten.

Wer also nun Computer-Zeitungen hat, welche ich nicht habe, sollte mir die für den CLUB interessanten Berichte und Listings photokopiert in zweifacher Ausfertigung zusenden. Dabei ist auf gute Qualität der Photokopien zu achten. Kleinere PhotokopierKosten könnten ja vom jeweiligen Clubmitglied getragen werden - wer etwas mehr photokopiert, kann mir ja eine Rechnung über die Photokopier-Kosten zusenden. Diese Kosten werden dann aus der Clubkasse erstattet. Bitte gebt bei den Photokopien immer an, aus welcher Zeitschrift mit Jahrgang und Nummer. Ich glaube, wir erhalten auf diese Weise ein tolles Archiv.

Folgende Computer-Zeitschriften sind vorhanden:

CHIP:	alle Ausgaben (bis auf März 1982) ab Dezember 1982 bis jetzt.	
Computer Persönlich:	1982: Nr. 25, 23 1983: Nr. 1, 3, 9 bis 16, 19, 21 bis 26 1984: Nr. 2, 3, 4, 6, 8, 9	
MC:	1982: Nr. 5 1983: Nr. 4, 6, 12 1984: Nr. 2, 4, 5	
Homecomputer:	1983: Nr. 3, 8, 10, 12 1984: Nr. 3, 5	
Happy Computer:	alle bisher erschienenen Ausgaben	
HC:	1983: Nr. 12	1984: Nr. 4
Elrad:	1982: Nr. 12	1983: Nr. 1
Genie Data:	1984: Nr. 1, 3	
80micro:	Februar 1984	
CPU:	1984: Nr. 6	
Elcomp:	1984: Nr. 4/5	

Kurzbericht vom Modell 4 P (Tandy)

(Günther Wagner)

Das Modell 4 von Tandy dürfte mittlerweile allen bekannt sein. Das Modell 4 P ist die 'portable' Version davon. Es hat 64 K-RAM, 2 Slimline-Laufwerke, eine frei bewegliche Tastatur und einen Preis von ca. 5700 DM.

Vor einigen Wochen war ich in der glücklichen Lage, kurze Zeit an diesem Gerät zu arbeiten - ich möchte deshalb in einem Bericht Euch kurz mitteilen, was mir aufgefallen ist.

Vom Design gefällt mir das Modell 4 P recht gut, ebenso der Bildschirm, weil er für ein tragbares Gerät zum einen recht groß ist (22 cm - 80*24), zum anderen mit der grünen Schrift gut abzulesen ist. Die beiden Laufwerke (je 184 K) sehen optisch recht gut aus - die Bedienung ist aber etwas unbequem. Man kann nämlich die Diskette nicht (wie bei anderen Laufwerken üblich) einlegen und dann den Schließhebel drücken, sondern man muß die Diskette ins Laufwerk ganz reindrücken und dann mit der zweiten Hand den Hebel betätigen. Das ganze ist ganz schön unbequem. Mir ist dann noch die Tastaur aufgefallen - sie klappert furchtbar (für Textverarbeitung möchte ich dieses nicht verwenden). Das Gewicht des Modell 4 P ist mir unbekannt - abschließend sei nur noch gesagt, daß das Modell 4 P zwar transportabel ist, aber bestimmt nicht überallhin mitgenommen werden kann!

Mehr kann ich nicht zum 4 P sagen, da ich ja nur kurz daran gearbeitet habe. Wer sich dafür näher interessiert, kann von mir einen Prospekt haben, oder auch in ein Tandy-Geschäft gehen.

//

P r o g r a m m i e r t i p :

Datensätze kann man auf verschiedene Art und Weise löschen. Eine Möglichkeit, die zumindest mir bis vor kurzem unbekannt war, benutzte Hartmut Obermann beim Clubadressen-Programm.

Prinzipiell funktioniert das Löschen von Datensätzen folgendermaßen: Nachdem der zu ändernde Datensatz gefunden wurde, wird gefragt, ob er gelöscht werden soll. Wird mit 'J' geantwortet, wird das Feld mit Leerzeichen (eines reicht) aufgefüllt. Danach wird durch eine Sortierroutine (CMD"O"...usw) diese Adresse ans Ende der Datenliste gebracht. Als letzte Aktion wird dann das Feld in dem die Anzahl der Datensätze gespeichert ist um 1 vermindert. Und schon ist der Datensatz verschwunden.

Wer kann Tips für's Programmieren weitergeben? Bitte teilt Euere Erfahrungen und Tips im Clubinfo mit (wir haben ja auch Einsteiger im Club, die sich dafür besonders interessieren dürften).

Luidger Röckrath

Schnell bewegte Grafik auf dem TRS-80

Will man größere Objekte schnell auf dem Bildschirm bewegen, dann stößt man sehr bald an die Geschwindigkeitsgrenzen des Basic-Interpreters. Es sei denn, man wendet das Verfahren an, das sich der Autor dieses Beitrags ausgedacht hat. Zur Demonstration dient ihm das bekannte Mondlandespiel.

Der TRS-80 besitzt eine Punktgrafik mit 128 × 48 Bildpunkten, die sich zwar nicht durch hohe Auflösung auszeichnet, aber dennoch für viele Anwendungen ausreichend ist.
Zur Steuerung der 6144 Bildpunkte werden vom Level-1- und Level-2-Basic zwei Befehle (SET(X,Y) und RESET(X,Y)), die einen durch die X- und Y-Koordinate spezifizierten Punkt setzen bzw. löschen, und eine Funktion (POINT(X,Y)), mit der der Zustand jedes Punktes abgefragt werden kann, zur Verfügung gestellt. Unbewegte und kleine bewegte Objekte können damit problemlos und einfach erstellt werden. Möchte man größere Objekte schnell bewegen, stößt man schnell an eine Grenze, die unüberwindbar erscheint, wenn man den TRS-80 als Black-Box betrachtet und sich nicht mit Bildschirmspeicher und Grafikcodes beschäftigt. Wer diesem Denken verhaftet bleiben will, sollte diesen Beitrag besser überblättern. Allen anderen sei im folgenden ein kurzer Exkurs in die „Innereien" des TRS-80 vergönnt.

Jede Adresse im Bildschirmspeicher entspricht einem Zeichen

Auf dem Bildschirm des TRS-80 werden normalerweise in 16 Zeilen zu je 64 Zeichen 1024 Buchstaben, Ziffern oder Sonderzeichen dargestellt. Wie gelangt nun das, was vom Programm ausgegeben wird, auf den Bildschirm? Dazu sind 1024 Speicherplätze, Bildschirmspeicher oder Video-RAM genannt, vorhanden. Jeder dieser Speicherplätze entspricht einem Zeichen auf dem Bildschirm. Die ersten 64 Speicherplätze entsprechen der ersten Bildschirmzeile und so weiter. Um einen ganz bestimmten Speicherplatz anwählen zu können, erhält jeder eine Nummer (Adresse). Der Zusammenhang zwischen Bildschirmspeicher-Adresse und Bildschirmplatz verdeutlicht die Grafik auf Seite C/7 in [1]. Die Bildschirmspeicher-Adressen beginnen nicht bei 0, sondern der Bildschirmspeicher belegt den Bereich von 15 360 bis 16 383, da die Adressen nicht nur dazu dienen, einen der Bildschirm-Speicherplätze auszuwählen, sondern gleichzeitig zur Ansteuerung aller anderen Einheiten des TRS-80.
Eine weitere Schaltung, das sogenannte Bildschirminterface, fragt nun den Bildschirmspeicher zyklisch ab und erzeugt aus dem Speicherinhalt ein Signal für den Monitor, so daß dieser die entsprechenden Zeichen anzeigt. Die bekannten schwarzen Striche auf dem Bildschirm kommen zustande, wenn Zentraleinheit und Bildschirminterface gleichzeitig auf den Bildschirmspeicher zugreifen wollen. Tritt dieser Fall ein, wird das Bildschirminterface gesperrt, während die Zentraleinheit den Bildschirmspeicher ansteuert.
Was auf dem Bildschirm erscheint, hängt davon ab, was im Bildschirmspeicher steht. Steht in einem Bildschirm-Speicherplatz ein Wert zwischen 0 und 63, so erscheint auf der entsprechenden Stelle auf dem Bildschirm ein Buchstabe, eine Ziffer oder ein Sonderzeichen (die Codes von 64 bis 127 bewirken das gleiche, da der Bildschirmspeicher kein Bit 6 beinhaltet; aus diesem Grunde zeigt der TRS-80 auf dem Bildschirm keine Kleinbuchstaben an, obwohl sie im Zeichengenerator vorhanden sind). Wie gelangt nun die Grafik auf den Bildschirm? Um es kurz zu sagen: genauso! Ein Platz für ein Zeichen auf dem Bildschirm kann auch sechs Punkte in einer 2 × 3-Matrix anzeigen. Will man jeden Punkt unabhängig von den fünf anderen setzen oder löschen, so müssen alle 64 (= 2^6) möglichen Kombinationen vorhanden sein. Die Codes von 128 bis 191 bilden die 64 möglichen Grafikcodes. Den Zusammenhang zwischen den einzelnen Bits und den Punkten gibt *Bild 1* an.

POKE-Befehle: Ein erster Schritt

Mit diesem Wissen gewappnet, lassen sich nun die Geschwindigkeitsprobleme und deren Behebung leicht verstehen. Dazu muß man noch wissen, daß der TRS-80 einen Basic-Interpreter beinhaltet, das ist ein Programm, das den Programmtext während der Ausführung analysiert und über Tabellen für die Basic-Anweisungen Unterprogramme anspringt. Dieses Verfahren ist vom Zeitaufwand immer dann günstig, wenn die Zeit, die zur Ausführung eines Befehls benötigt wird, wesentlich die Interpretationszeit überschreitet (z. B. Arithmetik). Bei den Grafikbefehlen ist die interne Verarbeitung minimentechnisch äußerst einfach, d. h. die Interpretationszeit fällt stark ins Gewicht. Dieses Verhältnis läßt sich durch die im Level-2-Handbuch [1] auf Seite 11/2 vorgeschlagene POKE-Grafik etwa bis Faktor sechs verbessern.

Aus dem bereits Gesagten ist leicht zu ersehen, wie die POKE-Grafik funktioniert. Die einzelnen Punkte eines Objektes werden nicht mehr einzeln gesetzt, sondern die Punkte (maximal sechs), die auf einen Bildschirmplatz fallen, werden zu einem Grafikcode zusammengefaßt und durch einen POKE-Befehl auf den Bildschirm transportiert. Das kleine Beispielprogramm in *Bild 2* verdeutlicht das Verfahren.

Bild 1. Die unteren sechs Bits jedes Bildschirmspeicherplatzes bestimmen die Zusammensetzung eines Grafiksymbols; eine Eins „setzt" den entsprechenden Block

Der Zeitgewinn durch dieses Verfahren ist noch immer nicht überwältigend. Aber es weist schon in die richtige Richtung. Es geht darum, mit möglichst wenigen Basic-Befehlen möglichst „viel Grafik" auf den Bildschirm zu bringen.

Grafikstrings lösen das Geschwindigkeitsproblem

Wie schon bei der POKE-Grafik erläutert, werden die maximal sechs auf einen Bildschirmplatz zusammenfallenden Punkte zu einem Code zusammengefaßt und so zusammen auf den Bildschirm befördert. Erstreckt sich nun ein Objekt über mehrere Bildschirmplätze, warum sollten dann nicht einfach die Grafikcodes aneinandergereiht und in einem String gespeichert werden? Den kann man dann durch einen einfachen PRINT-Befehl auf den Bildschirm bringen. Dieses Verfahren scheitert auch nicht, wenn das Objekt sich über mehr als einige unmittelbar aufeinanderfolgende Bildschirmplätze erstreckt. Dann müssen in die Strings außer den Grafikcodes auch noch die Cursorbewegungen eingebaut werden. Wie auf diese Weise ein Gebilde wie die Mondlandefähre in einen String gebracht werden kann, zeigt *Bild 3*.

Was aber, wenn das Objekt dann noch bewegt werden soll? Wie gesagt, wird der String durch einen PRINT-Befehl auf den Bildschirm transportiert. Ergänzt man den Befehl durch @,X, so kann das Objekt an jeder Position X erscheinen. Damit ist jedoch nur eine Bewegung möglich, die um den Faktor zwei (in horizontaler Richtung) bzw. drei (in vertikaler Richtung) gröber ist als bei normaler Grafik. Um eine fließendere Bewegung zu erhalten, bei der das Objekt jeweils nur um einen Grafikpunkt verschoben wird, ist eine weitere Verfeinerung der Stringgrafik notwendig.
Statt eines Strings für das Objekt werden drei gebildet (bei horizontaler Bewegung), jeder um einen Punkt verschoben (siehe Bild 3). Im vorliegenden Programm heißen diese drei Strings (1...3). Aus der Höhe der Fähre wird nun zuerst die Y-Koordinate berechnet. Aus dieser werden durch Division modulo 3 die Position für den PRINT-Befehl und der Index zur Auswahl der Fähre errechnet. Bei horizontaler Bewegung verfährt man analog.

```
10 REM DEMONSTRATIONSPROGRAMM
20 REM SET-GRAHIC
30 REM DURCH 'RUN'
40 CLS
50 FORX=60TO69:SET(X,23):NEXT
60 SET(64,22)
70 GOTO70
80 REM POKE-GRAPHIC
90 REM DURCH 'RUN100'
100 CLS:A=15838
110 POKEA,176:POKEA+1,176:POKEA+2,180:POKEA+3,176:POKEA+4,176
120 GOTO120
130 REM STRING-GRAPHIC
140 REM DURCH 'RUN150'
150 CLS
160 A$=CHR$(176)+CHR$(176)+CHR$(180)+CHR$(176)+CHR$(176)
170 PRINT@478,A$;
180 GOTO180
```

Bild 2. Demonstrationsprogramm zur POKE-Grafik: Das gleiche Objekt wird hier einmal durch die normalen Grafikbefehle erzeugt und einmal durch POKE-Befehle; ab Zeile 130 wird die Stringgrafik demonstriert

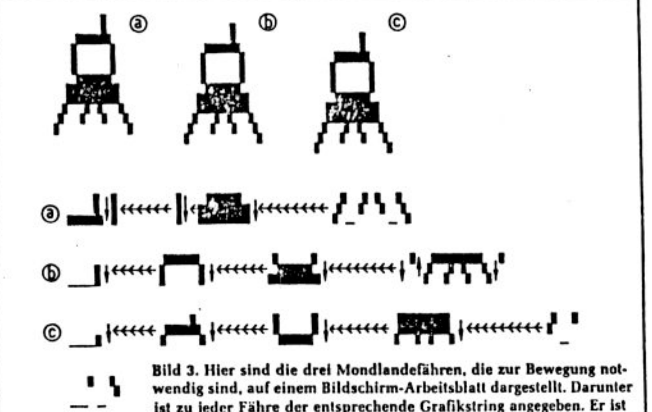


Bild 3. Hier sind die drei Mondlandefähren, die zur Bewegung notwendig sind, auf einem Bildschirm-Arbeitsblatt dargestellt. Darunter ist zu jeder Fähre der entsprechende Grafikstring angegeben. Er ist zusammengesetzt aus Grafikzeichen und Cursorbewegung und ist, umgesetzt in die entsprechenden Codes, in den DATA-Zeilen des Programmes wiederzufinden. Die Cursorbewegungen werden durch die Codes 24...27 erzeugt

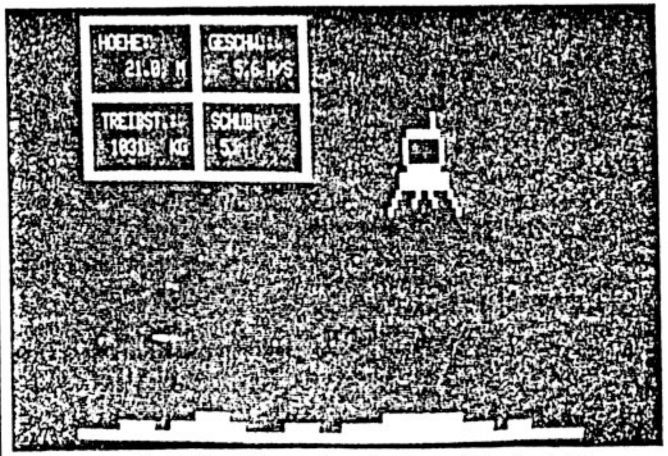


Bild 5. Hier setzt die Mondlandefähre gerade zur Landung an. Links oben sind die wichtigen Flugdaten angezeigt, rechts die Fähre

```
10 REM COPYRIGHT BY LUIDGER ROECKRATH, 15.3.81 (V 1.2)
20 CLS:CLEAR500:DIMA$(4):DEFINTI,K,J,N,X,Y,S,F
30 PRINTTAB(19)"* M O N D L A N D U N G *"
40 PRINT:PRINT"BITTE WARTEN SIE EINEN AUGENBLICK."
50 PRINT:PRINT"ES GEHT GLEICH LOS!"
60 REM -----------------------------------
70 REM GRAPHIC-STRINGS AUS DATAFELDERN LESEN
80 REM -----------------------------------
90 RESTORE:FORI=1TO14:READK:NEXT:FORN=0TO4:READI
100 FORJ=1TOI:READK:A$(N)=A$(N)+CHR$(K):NEXT
110 NEXT:I$=A$(0):A$(0)=A$(2):A$(2)=I$
120 REM ----------------------
130 REM EINFUEHRUNG DES BENUTZERS
140 REM ----------------------
150 CLS:PRINTTAB(19)"* M O N D L A N D U N G *"
160 PRINT:PRINT"ANLEITUNG ERFORDERLICH ? (J/N)"
170 GOSUB640:IFI$="N"THEN280
180 PRINT:PRINT"SIE BEFINDEN SICH AN BORD DER MONDLANDEFAEHRE 15
000 M"
190 PRINT"UEBER DEM MOND. SIE BEWEGEN SICH MIT 55 M/S AUF DEN MO
ND"
200 PRINT"ZU. IHRE AUFGABE IST ES MIT DEN ZUR VERFUEGUNG STEHEND
EN"
210 PRINT"2500 KG TREIBSTOFF SICHER AUF DEM MOND ZU LANDEN."
220 PRINT"IHRE LANDEGESCHWINDIGKEIT SOLL KLEINER ALS 4 M/S SEIN!
"
230 PRINT"SIE KOENNEN DEN SCHUB ZWISCHEN 0 UND 9 WAEHLEN, INDEM
SIE"
240 PRINT"EINFACH KURZ DIE ENTSPRECHENDE TASTE ANTIPPEN."
250 PRINT:PRINTTAB(40)"VIEL GLUECK!"
260 PRINT:PRINT"VERSTANDEN ? (J)":POKE16537,0
270 I$=INKEY$:IFI$()"J"THEN270
280 H=15000:V=55:T=2500:F=1000:S=0:M=4500:G=1.62:ZL=.3:ZH=1
290 CLS:PRINT@67,;
300 REM -----------------------------------
310 REM KONTROLLFELD AUF BILDSCHIRM ZEICHNEN
320 REM -----------------------------------
330 FORX=2TO55:SET(X,1):SET(X,10):SET(X,19):NEXT
340 FORY=2TO18:SET(2,Y):SET(3,Y):SET(28,Y):SET(29,Y):SET(54,Y):S
ET(55,Y):NEXT
350 PRINT"HOEHE:";:PRINT@80,"GESCHW.:";:PRINT@259,"TREIBST.:";:P
RINT@272,"SCHUB:";
360 PRINT@140,"M";:PRINT@151,"M/S";:PRINT@331,"KG";
370 P=41
380 RESTORE:FORN=1TO14:READA:PRINT@A,".";:NEXT
390 REM ------------
400 REM HAUPTSCHLEIFE
410 REM ------------
420 I$=INKEY$:IFI$()""THENI=ASC(I$):IFI)47ANDI(58THENS=I-48
430 IFT(=0THENT=0:S=0
440 PRINT@131,USING"#####.#";H;
450 PRINT@144,USING"####.#";V;
460 PRINT@324,USING"####";T;:PRINT@336,S;
470 NA=N:PA=P:I=INT((H+2)/3):P=744-64*I
480 IFP(40THENP=40:PRINT@960,STRING$(63," ");:N=2ELSEPRINT@960,A
$(4);:N=H-3*I+2
490 IFPA()PORNA()NTHENPRINT@PA,A$(3);:PRINT@P,A$(N);
500 IFH(100THENZ=ZLELSEZ=ZH
510 V=V-(F*S/M-G)*Z:H=H-V*Z:T=T-S*Z:M=M-S*Z
520 IFH)0THEN420
530 REM -----------------
540 REM ERFOLGSMITTEILUNGEN
550 REM -----------------
560 PRINT@131,USING"#####,#";0;:PRINT@144,USING"####.#";V;:PRINT
@P,A$(3);:PRINT@960,A$(4);
570 PRINT@512,"IHRE LANDEGESCHWINDIGKEIT BETRUG:";V;"M/S";:IFV)=
10THEN610
580 PRINT@744,A$(2);:PRINT@640,;:IFV(1THENPRINT"HEVORRAGENDE LAN
DUNG!";:GOTO630
590 IFV(4THENPRINT"GUTE LANDUNG!";:GOTO630
600 PRINT"HARTE LANDUNG";:GOTO630
610 IFV(100THENPRINT@808,LEFT$(A$(2),22);:PRINT@640,"HARTE LANDU
NG!";:PRINT@704,"DIE FAEHRE IST STARK BESCHAEDIGT.";:GOTO630
620 PRINT@997,CHR$(202);:PRINT@1003,CHR$(168);:PRINT@640,"SIE HA
BEN EINEN NEUEN KRATER ERZEUGT!";
630 PRINT@768,"NOCH EINMAL? (J/N)";:GOSUB640:IFI$="J"THEN280:ELS
EEND
640 POKE16537,0
650 I$=INKEY$:IFI$="J"ORI$="N"THENRETURNELSE650
660 REM ---------------------------
670 REM DATAFELDER FUER GRAPHIC-STRINGS
680 REM ---------------------------
690 DATA791,30,590,440,609,821,528,511,798,480,60,190,770,650
700 DATA40,176,176,176,181,26, 149,24,24,24,24,24,24,170,26,24
710 DATA188,191,191,191,191,188,26,24,24,24,24,24,24,24,24,160
720 DATA134,32,136,129,130,132 ,32,137,144
730 DATA51,32,32,32,148,26,24,24,24,24,24,168,131,131,131,131
740 DATA148,26,24,24,24,24,24,24,178,188,188,188,188,177,26,24
750 DATA24,24,24,24,24,24,24,26,130,27,152,131,163,135,139,147
760 DATA131,164,26,129
770 DATA65,32,32,32,144,26,24,24,24,24,24,160,140,140,140,141
780 DATA144,26,24,24,24,24,24,24,138,176,176,176,176,133,26,24
790 DATA24,24,24,24,24,24,160,143,143,159,175,143,143,144,26,24
800 DATA24,24,24,24,24,24,24,24,136,129,32,130,32,32,129,32,130,
132
810 DATA30,8,8,8,202,26,8,8,8, 8,8,8,8,8,8,8,26,202,26,8,8,8,8
820 DATA8,8,8,8,8,8,26,202
830 DATA63,176,176,176,176,176,188,188,188,188,176,176,188,188
840 DATA188,191,191,191,191,188,188,176,176,176,176,176,188,188
850 DATA188,188,176,176,176,188,188,188,188,188,191,191,191,191
860 DATA191,191,191,191,191,191,188,188,188,188,176,176,188,188
870 DATA188,176,176,176,176,176,176,176
```

Bild 4. Das Programm für den TRS-80 ist eine Variante der beliebten Mondlandesimulation mit bewegter Grafik

„Mondlandung" erklärt sich selbst

Zum Programm (*Bild 4*) läßt sich weiter nicht viel sagen. Den Spielablauf erklärt das Programm selbst. Der Ablauf der Landung wurde möglichst realistisch gestaltet. Um unnötige Frustrationen zu vermeiden, wurde der Treibstoff großzügig bemessen. Die Anfangsphase der Landung wird um Faktor drei gerafft, um die Wirkung der Endphase nicht zu beeinträchtigen.

Interessant ist vielleicht noch, daß die gesamten Berechnungen der Höhe, Geschwindigkeit usw. in einer einzigen Zeile erfolgen. Der Rest des Programmes dient nur zur komfortablen Ein- und Ausgabe.

Das Programm läuft auf einem 16-K-Modell mit Level-2-Basic. Durch Entfernen der REM-Anweisungen und Kürzen der Texte kann es auch auf einem 4-K-System lauffähig gemacht werden. Und nun viel Spaß beim Spielen. Einen Vorgeschmack gibt *Bild 5*.

Literatur

[1] Level II Reference Manual. Radio Shack.

H I N W E I S *** H I N W E I S *** H I N W E I S ***

Anfangs hatte ich den Fehler gemacht, zu vergessen, daß es neben Tandy auch Genie und Komtek gibt. Bitte verzeiht dieses unbeabsichtigte Vergessen der beiden letztgenannten Firmen; wie Ihr sehen könnt, habe ich die Etiketten und auch den Briefkopf mittlerweile entsprechend abgeändert.

W E R B U N G *** W E R B U N G *** W E R B U N G ***

Wenn ich die Reaktion der Mitglieder zusammenfasse, komme ich zu folgendem Ergebnis:

- Mitgliederwerbung ist gut mit einem Plakat auf Computertagen etc. möglich; Plakate an der UNI in München haben auch Wirkung.
- Es sollte in der GENIEDATA geworben werden - Schritte in dieser Richtung habe ich mittlerweile unternommen.
- Eine Anzeige in der Chip sollte wieder erscheinen, jedoch zu einem späteren Zeitpunkt. Es wird im November oder Dezember eine Anzeige erscheinen mit dem Zusatz, daß gewerbliche Inserenten für unser Clubinfo willkommen sind. Dem nächsten Info liegt eine Anzeige (Bücher) bei; wir erhalten dafür 20 DM.
- Bringt den CLUB 80 in eueren Tandy-Filialen und Genie-Geschäften in Erinnerung; bittet das Personal um Unterstützung (Werbung). Von einigen Tandy-Filialen ist mir bekannt, daß Sie Interessenten an unseren Club weiter verweisen. Werbt Bekannte an, die für den Club in Frage kommen.

Vielleicht hat jemand noch weitere Ideen; vielleicht entwirft jemand ein gutes Plakat bzw. Aushang mit gutem Text für unsere Mitgliederwerbung (der letzte Aushang gefällt mir nicht so recht).

Ü B R I G E N S *** Ü B R I G E N S *** Ü B R I G E N S

Bitte beachtet das Disketten-Angebot, welches unser Mitglied Hartmut Obermann an Land gezogen hat.

Jedem Info liegen je 2 kostenlose Kleinanzeigencoupons für das Blitz-Magazin bei. Dieses Magazin hat eine verkaufte Auflage von rund 25.000 Exemplaren und erscheint im gesamten Rhein/Ruhrgebiet und am Niederrhein. Ich habe insgesamt ca. 50 Coupons erhalten mit der Bitte um Weitergabe an die Mitglieder. Wer noch weitere Coupons braucht, kann diese bei mir anfordern.

In letzter Minute für Euch entdeckt, vielleicht ist jemand so nett, und schickt das fertige Programm für die Bibliothek ein!
die fertigen Programme

RAM-Disk kostenlos

Die Praxis zeigt, daß Programmierer eine RAM-Disk nur für eine Aufgabe benutzen, nämlich um einen Bearbeitungsstand schnell zu sichern. Das geht allerdings auch ohne DOS und RAM-Disk, wenn sie einen TRS-80 Modell I oder II oder einen Video Genie besitzen.

Schon ein Basic-Programm von 8 KByte Länge ist ein ganz schöner Brocken, auf einem 16-KByte-System bleiben da noch 8 KByte frei, auf einem 48-KByte-Computer entsprechend mehr. Diese freie Speicherkapazität läßt sich sinnvoll nutzen. Nehmen wir an, Sie haben ein Programm gerade eben zum Laufen gebracht, es sind nur noch ein paar Verbesserungen anzubringen. Sie ändern dafür einige Zeilen, und das geht (natürlich) schief. Also alle Zeilen zurückändern? Normalerweise hat man ja für diesen Zweck eine Sicherungskopie. Da diese aber auf einem Kassettensystem anzulegen und wieder zu laden lange dauert und bei Disketten zumindest noch eine Menge Tipperei bedeutet, fehlt eine solche Kopie meistens. Aber genaugenommen hat nicht der Programmierer Schuld, sondern seine Tools taugen lediglich nichts. So ein Software-Werkzeug ist nämlich nur dann gut, wenn es auch gerne benutzt wird. In diesem Falle bedeutet dies: Das Sichern und Laden muß mit einem einzigen Tastendruck zu erledigen sein.

Das Prinzip ist recht einfach. Das aktuelle Basic-Programm wird in einen RAM-Puffer kopiert, und von da bei Abruf zurückgeladen. Das Sichern geschieht mit Control-S, das Laden mit Control-L. Auf dem TRS-80 Modell I ist die Control-Taste mit Shift-Pfeil-abwärts zu simulieren, das Modell III und alle Video Genies haben eine eigene CTRL-Taste.

Der Programmumfang von 185 Bytes erklärt sich aus folgenden Eigenschaften:

1. Der Anwender soll die Größe des RAM-Puffers selbst wählen können.
2. Sowohl der Puffer als auch das Maschinenprogramm müssen vor Basic geschützt werden.
3. Das Programm mitsamt seinem Puffer soll sich immer unter dem aktuellen HIMEM befinden. Es muß also selbst feststellen, wieviel KByte der Computer hat und sich entsprechend selbst verschieben.

Zu 3.: Das Programm ist nicht relokatierbar geschrieben. Es enthält an 12 Stellen absolute Adressen. Dafür gibt es eine Tabelle, die der Lader zur Korrektur benutzt. Der Relokator wurde an den Basic-Lader angehängt, weil ich der Ansicht bin, daß sich das besser abtippen läßt als DATA-Items. Da diese Technik sehr nützlich, aber ziemlich unbekannt ist, wird sie ausführlich beschrieben.

Der Lader (Listing 1) fragt zuerst nach der Puffergröße BZ (Buffer Size). Dann stellt er in Zeile 30 fest, wo HIMEM (HM) liegt. »HM — (BZ+Pgm — Länge)« ergibt das neue HM, was in den HIMEM-Pointer (HP) eingetragen würde (Zeile 40). Das Maschinenprogramm wird mit »ORG F000H (—4096)« assembliert. Daraus folgt ein Offset (OS) von »(neuem) HM — ORG« (Zeile 70). In Zeile 80 wird das Programm durch POKE unverändert in den RAM geschrieben. Es bleiben aber noch 12 DATA-Items übrig. Diese halten die niederwertigen Bytes aller Adressen, die verschoben werden sollen. Das höherwertige Byte der Adresse ist in diesem Falle konstant, nämlich das der Anfangsadresse. Somit ergibt sich eine Adresse eines zu ändernden Bytes als »AD=HM+B« (Tabelle), siehe Zeile 100. Das Byte und das ihm folgende werden als Adresse (des Maschinenprogramms) gelesen. Darauf wird OS addiert. Die Summe wird zurückgeschrieben.

Diese Technik ist gut für Programme von nicht mehr als 255 Bytes. Werden es mehr, dann ist so zu verfahren: Schauen Sie zuerst auf Listing 2. Da steht in Zeile 13: »RELA1 EQU $-2«. Im Code-Feld finden Sie F004. Das ist die Adresse des Adreßteils des vorhergehenden Befehls mit absoluter Adressierung. Auf diese Art sind alle zwölf entsprechenden Adressen mit »RELA1-RELA12« bezeichnet. Ab Zeile 100 ist mit diesen Labels eine Tabelle definiert. Wollen Sie die vollständigen Adressen, müssen Sie nur anstatt »DEFB« »DEFW« schreiben und das "-OR" jeweils weglassen. In Listing 1 hätten Sie dann 24 Bytes zu bearbeiten und zwar so: »FOR I=1 TO 12:READ BL,BH:AD=BL+256*BH...«.

Der Lader darf nach dem Kaltstart des Computers nur einmal laufen. Deshalb endet er auch mit NEW. Dennoch: Einfach das Programm wieder zu laden und zu starten, reicht nicht. Sie sollten vorher einen RESET erzwingen. Die Kontrolle auf ein »Re-init« ist ziemlich aufwendig, wenn man unterstellt, daß das Programm auf den verschiedenen TRS-80's und Video Genies laufen soll. Wenn Sie die Kontrolle einbauen wollen, dann so: Stellen Sie fest, welche Inhalte 4016H und 4017H ursprünglich haben. Zu Programmbeginn prüfen Sie dann, ob diese beiden Bytes noch »original« sind. Wenn nicht, drucken Sie »Lader schon gelaufen, Sytem neu starten«.

Noch etwas: Im Lader wird nur einmalig festgestellt, ob der Puffer über oder unter 32767 startet. Das erspart Ihnen viel Tipperei, setzt aber voraus, daß Sie sinnvolle Puffergrößen wählen. Auf einem 16-KByte-System hat der Anwender 16 KByte unter 32767 zur Verfügung, auf einem 48-KByte-Rechner 32 KByte mehr (unter 65535), wegen Disk-Basic und DOS aber nur 38 KByte insgesamt. Da der Puffer höchstens die Hälfte davon belegen sollte, ergeben sich 8 KByte für den kleinen Ausbau, 19 KByte für den großen. In beiden Fällen überlagert der Puffer nicht die magische Grenze von 32767. Die vielen Tests darauf können also entfallen. Listing 2 zeigt das Assembler-Listing von »Pgm-Disk«.

Einige Meldungen gibt das Programm aus:

»PGM SAVED«: Nach CTRL-S. Programm ist gesichert, altes Programm im Puffer ist überschrieben.
»NO PGM TO SAVE«: Nach CTRL-S. Basic-Speicher ist leer. Verhindert, daß auch Puffer gelöscht wird.
»PGM TO BIG«: Nach CTRL-S. Programm größer als Puffer, alter Puffer bleibt erhalten.
»NO PGM SAVED«: Nach CTRL-L. Kein Programm im Puffer.
»LIST«: Nach CTRL-L. Programm wurde aus Puffer geladen und wird gelistet. Gegebenenfalls mit »BREAK« abbrechen.

(Peter Wollschlaeger/lg)

Listing 2. Assemblerlisting der »Pgm-Disk«, schlägt im Tempo jede echte RAM-Disk

```
F050          00061 RELA8   EQU    $-2
F052 1820     00062         JR     PRINT           ;NO PGM TO SAVE
F054 D1       00063 PGM     POP    DE
F055 A7       00064         AND    A
F056 ED52     00065         SBC    HL,DE           ;ELSE GET LEN
F058 ED5BB5F0 00066         LD     DE,(BUFSIZ)
F05A          00067 RELA9   EQU    $-2
F05C DF       00068         RST    18H             ;COMPARE TO BUFF SIZE
F05D 3805     00069         JR     C,OK
F05F 21A3F0   00070         LD     HL,MSG4         ;IF BAD
F062 1810     00071         JR     PRINT           ;TELL "PGM TO BIG"
F064 22B3F0   00072 OK      LD     (LEN),HL        ;IT FITS, SAVE LEN
F065          00073 RELA10  EQU    $-2
F067 44       00074         LD     B,H             ;BC=LEN
F068 4D       00075         LD     C,L
F069 2AA440   00076         LD     HL,(40A4H)      ;SOURCE
F06C 11C3F0   00077         LD     DE,BUFFER       ;DEST.
F06D          00078 RELA11  EQU    $-2
F06F EDB0     00079         LDIR
F071 217AF0   00080         LD     HL,MSG1         ;TELL
F072          00081 RELA12  EQU    $-2
F074 CD752B   00082 PRINT   CALL   PRTMSG          ;PGM SAVED
F077 C3CC06   00083         JP     READY
F07A 50       00084 MSG1    DEFM   'PGM SAVED'
F083 0D       00085         DEFB   0DH
F084 00       00086         DEFB   0
F085 4E       00087 MSG2    DEFM   'NO PGM TO SAVE'
F093 0D       00088         DEFB   0DH
F094 00       00089         DEFB   0
F095 4E       00090 MSG3    DEFM   'NO PGM SAVED'
F0A1 0D       00091         DEFB   0DH
F0A2 00       00092         DEFB   0
F0A3 50       00093 MSG4    DEFM   'PGM TO BIG'
F0AD 0D       00094         DEFB   0DH
F0AE 00       00095         DEFB   0
0002          00096 HEDSAV  DEFS   2
0002          00097 TALSAV  DEFS   2
F0B3 0000     00098 LEN     DEFW   0000H
F0B5 0A00     00099 BUFSIZ  DEFW   10              ;CHANGED BY LOADER
F0B7 04       00100 RELTAB  DEFB   RELA1-OR
F0B8 07       00101         DEFB   RELA2-OR
F0B9 1B       00102         DEFB   RELA3-OR
F0BA 22       00103         DEFB   RELA4-OR
F0BB 27       00104         DEFB   RELA5-OR
F0BC 30       00105         DEFB   RELA6-OR
F0BD 3D       00106         DEFB   RELA7-OR
F0BE 50       00107         DEFB   RELA8-OR
F0BF 5A       00108         DEFB   RELA9-OR
F0C0 65       00109         DEFB   RELA10-OR
F0C1 6D       00110         DEFB   RELA11-OR
F0C2 72       00111         DEFB   RELA12-OR
F0C3          00112 BUFFER  EQU    $
0000          00113         END
```

Listing 1.
Basic-Lader mit Relokator der RAM-Disk für Programme

```
1 'RAM-DISK für das aktuelle Programm
2 '
3 'Ein Programm kann mit CTRL-S im RAM gesichert und
4 'mit CTRL-L wieder geladen werden. Der Anwender kann die
5 'Puffergröße wählen.
6 '
7 '(C) 1984 Peter Wollschlaeger, Hildesheim
8 '
10 CLS:INPUT"Größe des Puffers in Bytes ";BZ
20 PRINT:PRINT"Moment bitte.....":PRINT
30 HP=16561:HM=PEEK(HP)+256*PEEK(HP+1):'get HIMEM
40 HM=HM-BZ-185:H=INT(HM/256):L=HM-256*H:POKE HP,L:POKE HP+1,H
50 IF HM>32767 THEN HM=HM-65536
60 OG=-4096:'Auf F000H wurde assembliert
70 OS=HM-OG:'Offset
80 FOR I=0 TO 182:READ B:POKE HM+I,B:NEXT
90 FOR I=1 TO 12:READ B
100 AD=HM+B:NA=PEEK(AD)+256*PEEK(AD+1)+OS:H1=INT(NA/256):L1=NA-256*H1
110 POKE HM+B,L1:POKE HM+B+1,H1
120 NEXT
130 H1=INT(BZ/256):L1=BZ-256*H1:POKE HM+181,L1:POKE HM+182,H1
140 DEFUSR=HM:'********** Nur Disk-Basic *************
150 POKE 16526,L:POKE 16527,H:'******* Nur Level 2 ***********
160 X=USR(0)
170 PRINT"Pgm-Buffer jetzt bereit"
180 CLEAR 50
190 PRINT"Basic stehen noch ";MEM;" Bytes zur Verfügung"
200 PRINT:PRINT:INPUT"<ENTER> drücken";A$
210 NEW:'******* Hier END einsetzen bis Pgm gesichert und OK *******
220 DATA 42,22,64,34,14,240,33,13,240,34,22,64,201,205,0,0
230 DATA 183,200,254,19,40,35,254,12,192,237,75,179,240,120,177,32
240 DATA 5,33,149,240,24,78,33,195,240,237,91,164,64,237,176,42
250 DATA 177,240,34,249,64,175,195,46,43,42,249,64,34,177,240,237
260 DATA 91,164,64,213,19,19,123,189,32,10,122,188,32,6,209,33
270 DATA 133,240,24,32,209,167,237,82,237,91,181,240,223,56,5,33
280 DATA 163,240,24,16,34,179,240,68,77,42,164,64,17,195,240,237
290 DATA 176,33,122,240,205,117,43,195,204,6,80,71,77,32,83,65
300 DATA 86,69,68,13,0,78,79,32,80,71,77,32,84,79,32,83
310 DATA 65,86,69,13,0,78,79,32,80,71,77,32,83,65,86,69
320 DATA 68,13,0,80,71,77,32,84,79,32,66,73,71,13,0,255
330 DATA 0,255,0,0,0,10,0,4,7,27,34,39,48,61,80,90
340 DATA 101,109,114
```

In letzter Minute – Programm **Software** bitte für die Bibliothek einsenden!

Telefonbuch mit Pfiff

Das Programm »Telefonregister« ist für Genie- und TRS-80-Computer ab 16 KByte Speichergröße geeignet. Es läßt sich aber auch als universeller Datenspeicher einsetzen. Ein Diskettenlaufwerk ist nicht unbedingt notwendig.

Das Programm ermöglicht das Durchsuchen einer Liste von Daten (in diesem Fall Namen, Telefonnummern und weitere Angaben von Telefonpartnern), wobei nicht nur nach Namen, sondern auch nach jeder anderen in den DATA-Zeilen abgelegten Information gesucht werden kann. So findet man beispielsweise anhand einer Telefonnummer den dazugehörigen Namen oder sucht alle Teilnehmer mit gleicher Vorwahlnummer heraus. Das Programm ist also vielseitig und läßt sich als universelle Datenbank einsetzen.

In der vorliegenden Fassung wird zuerst mit Hilfe eines schnellen String-Befehls die Grafik eines Telefons aufgebaut. Dann fragt der Computer nach dem Suchbegriff, mit dem er die DATA-Zeilen auf Übereinstimmung durchsucht. Werden mehrere Daten gefunden, so kann man mit »W« weiterblättern.

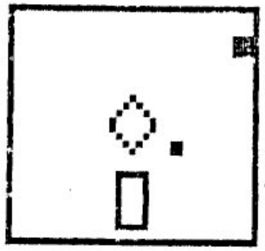
Das Programm ist durch REM-Zeilen erläutert, die bei der Eingabe natürlich gefahrlos weggelassen werden können. Nur zwei Unterprogramme sind vorhanden, die in den Zeilen 470 und 530 beginnen. Das Einlesen der Grafik erfolgt über eine GOTO-Schleife, das Durchsuchen der DATA-Zeilen mit einer FOR-NEXT-Schleife.

(Stefan Frisch)

```
10 '******************************************
    * TELEFONREGISTER MIT FREIEM SUCHBEGRIFF *
    * V 1.2 1983 IN GENIE/TRS-80 BASIC       *
    * PROGRAMMIERT VON STEFAN FRISCH         *
    ******************************************
20 CLS
30 CLEAR 100
40 DEFINT A-Z: REM *** GESCHWINDIGKEITSVERBESSERUNG ***
50 Z=6 :REM DRUCKPOS. BILD
60 PRINT @ Z,"";
70 REM *** EINLESEN DER GRAFIKZEICHEN UND IHRER ANZAHL ***
80 READ X, Y
90 IF X=99 THEN GOSUB 130 : REM *** NEUE ZEILE ? ***
100 IF X=9999 THEN 170 : REM *** ENDE DER GRAFIK ? ***
110 PRINT STRING$(X, Y);
120 GOTO 80
130 Z=Z+64 : REM *** NÄCHSTE ZEILE ***
140 PRINT @ Z,"";
150 READ X, Y
160 RETURN
170 PRINT @ 285," 110 ";
180 PRINT @ 413," 111 ";
190 PRINT @ 348,"NOTRUFE";
200 GOTO 240 : REM *** ZUM HAUPTPROGRAMM ***
210 DATA 1, 32, 1, 176, 1, 140, 45, 131, 1, 140, 1, 176, 99, 99,
 1, 191, 11, 32, 3, 176, 21, 188, 3, 176, 11, 32, 1, 191, 99, 99
, 1, 191, 7, 32, 1, 176, 1, 188, 31, 191, 1, 188, 1, 176, 7, 32,
 1, 191, 99, 99, 1, 191, 7, 32, 10, 191, 1, 188, 3, 32, 7, 176,
3
220 DATA 1, 191, 7, 32, 1, 131, 1, 143, 3, 191, 2, 143, 1, 131,
2, 179, 1, 176, 1, 32, 1, 188, 9, 191, 1, 188, 1, 32, 1, 176, 2,
 179, 1, 131, 2, 143, 3, 191, 1, 143, 1, 131, 7, 32, 1, 191, 99,
 99, 1, 191, 10, 32, 1, 176, 2, 188, 5, 191, 1, 32, 11, 191, 1,
32
230 DATA 1, 191, 10, 32, 8, 191, 1, 176, 1, 143, 9, 191, 1, 143,
 1, 176, 8, 191, 10, 32, 1, 191, 99, 99, 1, 191, 10, 32, 10, 191
, 1, 188, 7, 179, 1, 188, 10, 191, 10, 32, 1, 191, 99, 99, 1, 19
1, 10, 32, 29, 143, 10, 32, 1, 191, 99, 99, 1, 32, 1, 131, 1, 14
0
240 PRINT @ 657,"T E L E F O N R E G I S T E R"
250 RESTORE
260 READ X, Y : REM *** GRAFIKDATEN UEBERLESEN ***
270 IF X<9999 THEN 260
280 Z=768 : REM *** DRUCKPOSITION ***
290 PRINT @ 640,""
300 PRINT : REM *** ALTE EINGABE UEBERSCHREIBEN ***
310 PRINT @ 704,"SUCHBEGRIFF";
320 INPUT A$ : REM *** EINGABE DES SUCHSTRINGS ***
330 K=LEN(A$)
340 PRINT @ 740,"*SUCHEN*"
350 READ D$ : REM *** TELEFON-DATAS LESEN ***
360 IF D$="ENDE" THEN PRINT @ 960,"*** ENDE DER LISTE ***";: GO
SUB 470: GOSUB 530: GOTO 250
370 REM *** DURCHSUCHEN DER DATAS ***
380 S=LEN(D$)-K :REM *** S = LÄNGE DES DATAS MINUS LÄNGE DES
              *** SUCHSTRINGS
390 IF S<0 THEN 350
400 FOR I=1 TO S+1
410 M$=MID$(D$, I, K)
420 REM *** BEI ERFOLG PLATZ MACHEN, DATA SCHREIBEN, AUF
        *** "W" WARTEN UND DANN WEITERSUCHEN
430 IF M$=A$ THEN GOSUB 530: PRINT @ Z, D$;: GOSUB 470: GOTO 350
440 NEXT
450 GOTO 350
460 REM *** ABFRAGE WEITERBLÄTTERN ? ***
470 PRINT @ 740,"W=WEITER";
480 Z$=INKEY$
490 IF Z$<>"W" THEN 480
500 PRINT @ 740,"*SUCHEN*";
510 RETURN
520 REM *** UNTERE BILDSCHIRMHÄLFTE LOESCHEN ***
530 PRINT @ 704,""
540 PRINT: PRINT: PRINT: RETURN
550 REM *** AB HIER IHRE DATEN ***
600 DATA COMPUTER PERSOENLICH 089/4613-0
610 DATA HOBBY COMPUTER 089/4613-0
HANS-PINSEL-STR.2
8013 HAAR BEI MUENCHEN
620 DATA ...TRAGEN SIE IHRE DATEN AM BESTEN ALPHABETISCH EIN.
630 DATA ...VERWENDEN SIE NIE KOMMA DOPPELPUNKT UND APOSTROPH.
640 DATA ...DAS LETZTE DATA MUSS "ENDE" SEIN.
650 DATA ...BEI MEHRZEILIGEN DATAS ABWÄRTSPFEIL BENUTZEN.
660 DATA ENDE
```

Listing zu »Telefonregister«

SUPERPREIS



5 1/4" DISKETTEN

10 STUECK inkl. Verpackung und Porto

42,--

5 1/4 Zoll Disketten Single Sided / Double Density (Markenlos)

Liebe Clubmitglieder

durch freundschaftliche Beziehungen zu einem Computershop kann ich die oben genannten Disketten zu diesem wirklich sagenhaft guenstigen Preis beschaffen. Die Disketten sind ungestempelte Markenware. Sie laufen bei mir mit einfacher Dichte / einseitig und bei einem Kollegen mit doppelter Dichte / doppelseitig mit jeweils 40 Tracks. Selbstverstaendlich kann ich dafuer keine Garantie uebernehmen. Aber selbst fuer Single Sided/Single Density - Disketten ist der Preis unuebertroffen.
Eure Bestellung schickt Ihr bitte unter Beilage eines Schecks oder Bargeld an folgende Adresse: Hartmut Obermann
Schwalbacher Strasse 6
(06124/3913)
6209 Heidenrod/Kemel
oder ruft kurz an und uebersendet das Geld auf folgendes Postscheckkonto: 418 009 - 604 Postscheckamt Frankfurt a.M..
Die ganze Aktion ist keine Sammelbestellung. Ich kann also Bestellungen aufnehmen, solange der Computerchop die Disketten liefern kann.
Auch an andere Artikel komme ich zu einem besonder guenstigen Preis heran. So z.B. an Mannesmann Tally - Drucker, voll Epson-kompatibel (RX 80 F/T) fuer ca. 850,- DM (ohne Versand).
Solltet Ihr spezielle Fragen haben, ruft mich einfach mal an oder schreibt mir. Ich wuerde mich freuen.

Mit freundlichen Gruessen euer Clubkamerad

Hartmut Obermann
Hartmut Obermann

Was ich bis jetzt vergessen habe:

Herr König schrieb: Bei der Zusammenstellung (Herstellung) des Clubs sollten Clubmitglieder die schon eine gewisse Erfahrung haben, eine Seite übernehmen.

So denke ich mir das auch, und hoffe auf weitere Beiträge jeglicher Art seitens der Mitglieder.

Und dann zwei Anregungen von Gerald Schröder:

1. Es ist immer schwer, deutsche Anleitungen für große Programme (Pencil, Scripsit, etc) zu bekommen. Bei vielen ist es sogar unmöglich. Vielleicht sollte man solche übersetzungen in die Programmliste aufnehmen und zu eigenen übersetzungen anregen (ev. mit Punkten). Interessant wäre in diesem Zusammenhang die Eindeutschung von Programmen (Superdos - Newdos). Solche Arbeiten könnten von Günther Wagner koordiniert werden.

2. Es gibt viele interessante Adventures, bei denen man aber oft mittendrin hängenbleibt. Eine Art Adventure-Ecke könnte hier Abhilfe schaffen.

Bitte macht Euch Gedanken über diese beiden Anregungen und schreibt mir Euere Meinung, damit wir im nächsten Info weiter darauf eingehen können.

! Einen sehr guten Tip habe ich von Hartmut Obermann erhalten. Ihr könnt Euere Beiträge, Fragen, persönliche Vorstellungen etc. als Textfile zusenden, da ich ja im Prinzip mit meinem Scripsit alle Textfiles laden kann. Die Änderungen sind nur gering. Ihr erspart mir dadurch eine Menge Schreiberei. !

Zu meinen Fragen im letzten Clubinfo möchte ich kurz berichten:
Zu 1. und 2.: Danke für die ganzen Tips, aber die PDRIVE-Angaben und die COPY-Befehle waren klar. Mein Fehler war, daß ich WRDIRP unter NEWDOS aufgerufen habe, dies muß aber unter MINI-DOS erfolgen. Ich kann jetzt sämtliche Disketten schreiben und lesen.
Zu 3.: Hier erhielt ich einiges - aber leider läuft bis jetzt noch nichts.
Zu 4.: Also, mein Drucker hat keinerlei Möglichkeiten der Umschaltung über irgendwelche Schalter. Das Problem liegt ganz einfach am Drucker-Treiber des NEWDOS der deutsche Umlaute nicht richtig an den Drucker weitergibt.

Danke für die Antworten auf meine Fragen!

INTERFACE MIT 8251 UND 8255 FÜR Z80-RECHNER

Das Interface wurde entwickelt, um beliebige Geräte mit dem Computer verbinden zu können.
Es gibt bereits folgende Anwendungen:
Eprom - Programmierer
Joystick Anschluß
Morsezeichen - Decoder
Morsetrainer
Funkfernschreiben
Steuerung von el. Verbrauchern über Relais
Bei der Entwicklung wurde darauf Bedacht genommen, daß es nicht nur in Maschinensprache, sondern auch in Basic leicht programmiert werden kann.
Bei Verbindung der ser. Schnittstelle mit RS-232 od. V-24 ist zu beachten, daß der 8251 TTL - Signale verarbeitet. Sollte die Schnittstelle mit EIA-Pegeln (+- 3.....15 Volt) arbeiten, sind noch entsprechende Line-Driver nachzuschalten. (z.B.: 75189)
Literatur zur Programmierung dieser Bausteine gibt es bei Siemens: Die Beschreibungen der SAB 8080-Serie

SCHALTUNGSBESCHREIBUNG

Der Adreß - und Datenbus wird aus dem Computer über ein Kabel entnommen. Die Steckerbezeichnungen gelten für den KOMTEK. Bei anderen Geräten bitte im User-Manual nachsehen.
Direkt am Eingang ins Interface sind alle Bus-Leitungen mit je 3,9 k nach Masse terminiert. Diese Widerstände habe ich im Schaltbild nicht eingezeichnet.
Alle Signale sind mit 74245 voll gepuffert, sodaß Rückwirkungen auf den Computer ausgeschlossen sind.
Das Signal I/O request ist mit einem 74121 um ca. 0.4 usec verzögert. Das ist notwendig bei Floppy-Betrieb, um die Timer-Interrupts vom Interface fernzuhalten.
Das Signal 'OSZ 2 MHZ' an Pin 45 ist in der Frequenz völlig unkritisch. Notfalls lässt sich auch die Leitung 'MI' verwenden. Diese Clock ist nur zur internen Steuerung des 8251 notwendig.
Die Adreß-Decodierung besorgt ein 74138. Die Ausgänge liegen an Lötstiften; sodaß eine leichte Änderung möglich ist.
Originalbelegung:

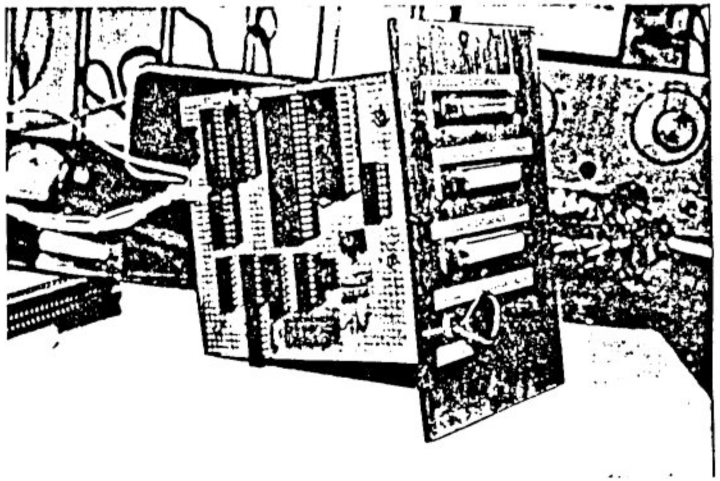
PIO 8255:	7C = Port A	
	7D = Port B	
	7E = Port C	
	7F = Control-Register (für Steuerung der Ports)	
SIO 8251:	64 = Datenregister	
	66 = Steuer- und Statusregister	
Timer :	60 = Steuerwort für Baudrate (1,2,4,8)	

AUFBAU UND ABGLEICH

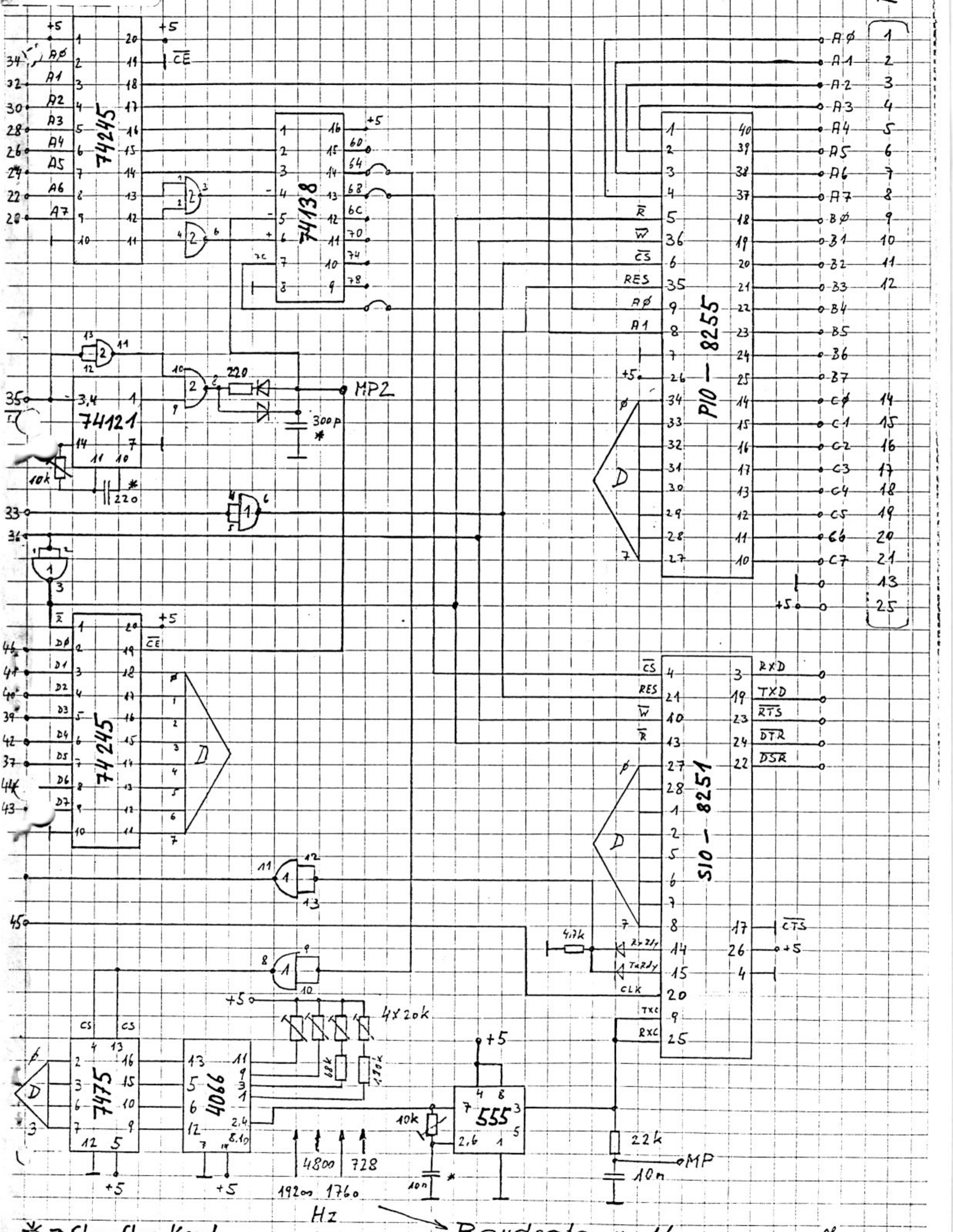
Die Verdrahtung habe ich auf einer Veroboard-Platte in der Größe einer Europa-Karte (160 X 100 mm) vorgenommen.
Es wurde die sogenannte Fädeltechnik angewandt, die für ein Einzelstück die schnellste Aufbauart darstellt.
Bei einer größeren Stückahl würde sich der Entwurf einer Platine durchaus lohnen. Es wurde allerdings noch kein Layout dafür erstellt.

Einstellung:
Der Timer kann mit einem OUT-Befehl auf 4 verschiedene Takt-Raten eingestellt werden. Diese ist am Punkt MP mit einem Freqenzzähler meßbar. Im Schaltbild sind die Werte für 45,110,300,1200 Baud angegeben. Durch Wahl der Potis und ggf der Vorwiderstände läßt sich jede beliebige Frequenz einstellen. Formel: Frequenz=Baudrate X 16 oder X64 oder X1 je nach Steuerwort im 8251.
Das Poti am NE555 bestimmt das Tastverhältnis. Es wird so eingestellt, daß bei der niedrigsten und auch bei der höchsten Taktrate ein sicheres Schalten erfolgt. (nicht sehr kritisch, aber etwas Einfluß auf die Frequenz)
Einstellung des Trimmers am 74121: Oszilloskop mit I/O req triggern. Signal am MP2 auf 0.4 usec Verzögerung mit dem Trimmpoti einstellen. (Fallende Flanke)
Ohne Oszi: Trimmer auf Mittelstellung

Walter Zwickel
Lengfelden 123
A5101-Bergheim



Parallel + Seriell Interface f. Komtek, TRS 80, VG



* = Styroflex Kond.

Alle Gater = 7400

Baudrate × 16

od. × 64

Werbung !!!

*** EDV-ZUBEHÖR-VERSAND ***

POSTFACH 1322, 3502 VELLMAR 1

RECYCLING-ENDLOS-PAPIER
12 ZOLL X 240 MM, 2000 BL. DM 32,00

BASF FLEXI DISKETTEN
5.25 1X SS,SD 10 ST. DM 46,00
5.25 1D SS,DD 10 ST. DM 47,00
5.25 2D DD,DD 10 ST. DM 69,00
5.25 1/96 SS, 96 TPI 10 ST. DM 65,00
5.25 2/96 DS, 96 TPI 10 ST. DM 77,00
ALLE DISKETTEN MIT VERSTÄRKUNGSRING

FARBBÄNDER FÜR TANDY
LP 2 NACHFÜLLK.ZIP-PACK 4 ST.DM 49,00
LP 2 KASSETTE DM 22,50
LP III, IV, V, VI UND VIII ST. DM 16,80

WEITER DISKETTEN, FARBBÄNDER UND PAPIERFORMATE AUF ANFRAGE.
ALLE PREISE ZZGL. MWST. SOWIE POSTÜBLICHE VERSANDKOSTEN PER NACHN.
LIEFERZEIT CA. 1-2 WOCHEN NACH AUFTRAGSEINGANG.

TRS-80 Modell I/III – Schnelle Grafik in Basic

Wenn Sie jemals versucht haben, in Basic ein schnel[les] Grafikprogramm zu schreiben, werden Sie festgestellt [ha]ben, daß das unmöglich ist. Wenn Sie eine etwas größere E[r]fahrung mit Computern haben, kennen Sie bestimmt d[ie] String-Pack-Technik, die erstmals von Leo B. Christophers angewendet wurde. Mit dieser Technik sind vie[le] TRS-80-Besitzer in die Welt der schnellen Grafik vorgest[o]ßen. Diese Methode erzeugt eine der schnellsten Grafik[en], die von Basic aus möglich ist.

Es gibt jedoch noch eine Alternative, zum Beispiel für ein[en] Programmvorspann oder für Displays, um Aufmerksamk[eit] zu erregen. Ich wurde auf diese Möglichkeit aufmerksam, [als] ich mit dem 32-Zeichen-Mode arbeitete.

Um von der schnellen Grafik Gebrauch zu machen, m[uß] das Programm die Bildschirmanzeige in den 32-Zeich[en]-Mode bringen und die gewünschten Zeichen in die unger[a]den Speicherstellen POKEn oder PRINTen. Nun sind die Z[ei]chen noch unsichtbar im Bildschirmspeicher, und wenn m[an] in den 64-Zeichen-Mode umschaltet, werden sie mit (für B[a]sic) unglaublicher Geschwindigkeit am Bildschirm ang[e]zeigt.

Das Beste an dieser Technik ist, daß man üblicherwei[se] Speicherplatz spart und trotzdem durch ein blinkendes M[e]nü, eine Grafik und so weiter die Aufmerksamkeit erwec[kt]. Hier ist ein kurzes Beispielprogramm:

```
10 CLS:PRINTCHR$(23)
20 FOR X=1 TO 1023 STEP 2:POKE 15360+X,191:NEXT X
30 FOR X=0 TO 1023 STEP 6:POKE 15360+X,191:NEXT X
40 FOR T=1 TO 9:FOR Z=1 TO 50
50 OUT 255,8:FOR R=1 TO Z:NEXT R
60 OUT 255,0:FOR R=1 TO Z:NEXT R
70 NEXT Z,T
```

Wenn Sie mit einem TRS-80 MIII arbeiten, müssen Sie a[lle] 255 in 236 und die Zahlen 8 und 0 in 4 und 0 ändern.

(G. Hornu[ng])

DIE LETZTE SEITE — ETIES ETZTEL EID

Jetzt kann ich beruhigt in den Urlaub gehen – endlich bin ich auf der letzten *Seite* angelangt – jetzt brauche ich das Info nur noch photokopieren und versenden. Ich hoffe, es macht Euch nichts aus, daß ich fast alles verkleinert habe, aber das Clubinfo wird dadurch um einiges billiger (Photokopierkosten und Porto).

Bitte laßt mich nicht im Stich. Sendet Euere Programme zu, schreibt Artikel über irgendwelche Sachen die von Interesse sein könnten, schreibt Euere Kritik und Anregungen, ... usw. Ich hoffe, das nächste Clubinfo wird wieder annähernd so umfangreich wie dieses.

Für's nächste Clubinfo ist bis jetzt geplant:

- Veröffentlichungen seitens der Mitglieder
- Bekanntgabe des Ergebnisses der Clubsatzungsänderung
- Ein Bericht über G-DOS
- Neue Bücher, Programme und Mitglieder
- 'feste' Bestandteile des Info's (z.B. Fragen)
- ... ??? (ich laße mich überraschen)

Bestimmt habe ich in diesem Info etwas vergessen; erinnert mich daran, damit ich es im nächsten Info nachholen kann. Das nächste Clubinfo ist geplant für Ende September, es kann aber auch sein, daß es erst im Oktober erscheint. Bis dahin viele Grüße,

Euer
Günther Wagner

Diese Wolke nehm' ich noch mit" sagte der Fallschirmspringer und tauchte in den Bodennebel ein.

Ein Bayer und ein Engländer sitzen an der Bar. Fragt der Engländer: „Are you smoking?" Darauf der Bayer: „No! Dös is a Trachtenanzug!"

In einer preußischen Schulklasse fragte der Lehrer: „Wer von euch kann mir sagen, was das Deutsche Reichsgericht ist?" Egon meldete sich: „Ich glaube Eisbein mit Sauerkraut, Herr Lehrer!"

„Herr Ober, bringen Sie mir eine Forelle!" Ein zweiter Gast: „Mir bitte auch, aber eine frische!" Geht der Ober zur Küche und ruft: „Zweimal Forelle, eine davon frisch!"

Eine Dame sitzt in einem Lokal und wartet schon eine Stunde auf ihr Essen. Sie ruft ärgerlich den Kellner zu sich: „Herr Ober, haben Sie mich vergessen?" „Aber nein", erwidert der, „Sie sind doch die Dame mit dem Schweinebauch!"

Psychiater: „Wann merkten Sie, daß die Flitterwochen zu Ende waren?" „Als mein Mann wieder anfing, seine Lieblingsspeisen zu essen!" „Und was ißt er am liebsten?" „Rohe Zwiebeln, Limburger Käse und Knoblauch!"

Karl und seine fünf Kumpane sitzen in der Kneipe. Karl liest die Zeitung und ruft: „Ich glaub', ich habe einen Job für uns entdeckt!" „So, und das wäre?" „Hier steht: Sechs Cognacschwenker gesucht!"

Das neue Dienstmädchen kommt atemlos vom Einkaufen zurück. Es zieht einen Mann hinter sich her, der sich heftig wehrt. „Ich habe alles bekommen, gnädige Frau. Zwei Apfeltaschen, einen Streuselkuchen, nur der Berliner macht immer noch große Schwierigkeiten, wie Sie sehen können!"

Die vierjährige Enkelin hat beim Gemüsemann Bananen eingekauft. „Hast du denn auch schön ‚Guten Tag' gesagt?" fragt die Tante, als die Enkelin stolz von ihrer Besorgung zurückkommt. Lässig meint sie: „Brauchte ich überhaupt nicht, Omi. Das hat der Mann ganz von allein gesagt."

CLUB 80 - Clubinfo Nr. 3

September 1984

Hallo liebe Clubfreunde !

Wieder ist es soweit - ein neues Clubinfo kommt ins Haus. Ich hoffe, daß Euch auch dieses gefällt. Es wurde unter Zeitmangel erstellt und es geht thematisch etwas drunter und drüber - aber Ihr werdet schon finden was Ihr sucht.

Viel neues gibt es zu berichten: Die Adventure-Ecke macht von sich reden, viele Bücher und Anleitungen werden angeboten, viele neue Programme und 2 neue Bücher sind dabei, neue Anregungen und Fragen, eine Ausarbeitung über BASICODE, eine tolle Seite vom Hartmut, zweimal Werbung, ein kleines Problem mit einem Mitglied wird beschrieben, 4 neue Mitglieder werden vorgestellt, etc.

Extra erwähnen möchte ich die Änderung der Clubsatzung. Alle Mitglieder können Änderungen der Clubsatzung beantragen - als Verfahrensweise kann diese 1. Änderung als Vorbild gelten. Weitere Änderungen sind durchaus denkbar, da ja die Clubsatzung bei der Clubgründung nur von mir alleine aufgesetzt worden ist.

Übrigens - es laufen einige (ganz entfernte) Überlegungen in Richtung Clubtreffen und Eintragung des CLUB 80 ins Vereinsregister. Wer auch Überlegungen anstellt kann mir ja mal schreiben.

Nun muß ich aber endlich zu meiner ganz großen ENTSCHULDIGUNG kommen. Viele Mitglieder mußten sehr lange auf die Bearbeitung Ihrer Sendungen warten. Bitte habt Verständnis, nach dem Urlaub hatte ich einen großen Berg Post vor mir liegen und bis Mitte September kaum Zeit (2. Kurzurlaub und zwischendurch 16 Tage wo ich von 7 Uhr bis 23 Uhr gearbeitet habe).

Ein großes DANKESCHÖN an alle, die wieder so eifrig mitgearbeitet haben - ein toller Schwung ist im Club und in ca. 5-6 Monate wurde wirklich beachtliches erreicht. Hoffen wir, daß es so weiter geht. Das 4. Clubinfo erscheint in der Woche nach Allerheiligen, das 5. Clubinfo wird dann unser Weihnachts-Clubinfo.

Viele Clubgrüße bis zum nächsten Mal

Servus

Euer Günther

Die Neuen !!!

Wagner Alexander
Theresienstr. 21c
8224 Chieming
08664/1500

Clubmitglied seit : 13.07.84
Punktestand = 25+
System- und Drivekonfiguration :
TRS-80 Model I Level II (48K) Interface 2 Laufwerke (je 40 Spure n/ ss/ sd) Kassetten-Recorder CTR-80
bevorzugtes Betriebssystem : ?

May Holger
Marienstr. 9
5768 Sundern 2
02935/1668

Clubmitglied seit : 12.08.84
Punktestand = 30
System- und Drivekonfiguration :
Genie I (64K)/ Star Gemini 10x/ Joystick/ Siemens Fernschreiber (mal als Drucker benutzt) 1 Laufwerk (40 Spuren/ ss/ sd)
bevorzugtes Betriebssystem : NEWDOS

Wies Jean-Claude
Harthweg 9
6600 Saarbruecken
0681/582513

Clubmitglied seit : 11.09.84
Punktestand = 30
System- und Drivekonfiguration :
Video Genie II (84'er Modell)/ Kassettenrekorder/ Monitor/ Star Gemini 10x 2 Laufwerke (je 40 Spuren/ ss/ sd)
bevorzugtes Betriebssystem : NEWDOS 80 Vers. 2.025

Voigtlaender Holm
Haselnussweg 30
6940 Weinheim
06201/65241

Clubmitglied seit : 03.09.84
Punktestand = 30
System- und Drivekonfiguration :
Genie II / Drucker Epson RX-80 1 Laufwerk (40 Tracks/ ss/ sd)
bevorzugtes Betriebssystem : NEWDOS

Ä N D E R U N G   D E R   C L U B S A T Z U N G + + + + + + +

Bereits im zweiten Clubinfo wurde eine Änderung unserer noch jungen Clubsatzung beantragt. Heute darf ich Euch über das Ergebnis informieren.

Bis zum 16. September gingen 9 Stimmzettel ein. Das Ergebnis lautet:

8 Stimmzettel für Änderung der Clubsatzung
1 Stimmzettel gegen Änderung der Clubsatzung

Die erforderliche 3/4 Mehrheit für eine Clubsatzungsänderung wurde hiermit erreicht. Somit heißt es jetzt:

"Das Buch wird auf Kosten des Entleihers versandt".

Um das Ausleihen von Büchern aus der Bücherbibliothek des Club einfach zu halten, wird folgende Regelung festgesetzt:

> Wer ein Buch vom Club ausleiht, teilt mir mit, welches Buch er will. Mehr ist nicht nötig. Das Buch wird im zugesandt (falls ausgeliehen kommt er auf die Warteliste). Erhält er das Buch, sieht er anhand des Umschlags die Portokosten. Diese müssen mit dem Buch zurückgesandt werden (Briefmarken oder Geld).

A D V E N T U R E   E C K E

Die Anregung vom Gerald Schröder fand großes Interesse. Es wird also eine sogenannte Adventure-Ecke eingerichtet - was dort geschieht, bestimmen alle 'Abenteuerer' zusammen mit

Alexander Wagner
Theresienstraße 21c
8224 Chieming
Tel. 08664/1500

der sich bereit erklärt hat, die Adventure-Ecke zu übernehmen. Ihr schreibt entweder direkt an den Alexander oder an mich (ich leite es dann weiter). Schön wäre es, wenn für jedes Info ein Beitrag aus der Adventure-Ecke dabei wäre - lassen wir uns mal überraschen!

Adventure-Ecke

Der Alexander Wagner (unser Mann von der Adventure-E :) sandte mir folgenden Beitrag:

Vielen Dank für das Clubinfo. Es war sehr interressant und enthielt einige gute Vorschläge. Einer der Vorschläge war die Einführung einer Adventure-Ecke. Ich kann diesen Vorschlag nur begrüßen,da ich selber gerne Adventures spiele und auch an gewissen Stellen meine Probleme habe. Ich würde Vorschlagen, daß jedes (halbwegs) gelößte Adventure-oder Fragen (wie komme ich wo weiter ?) in der Ecke abgedruckt werden. Auch sind oftmals eine komplette Befehlsliste und ein genauer Lageplan sehr hilfreich. Ich möchte gleich mit gutem Beispiel vorangehen und die Lösung, den Wortschatz und den Lageplan von dem Adventure GEHEIMAGENT XP-05, das sich ja in der CLUB-Programmsammlung befindet, geben.

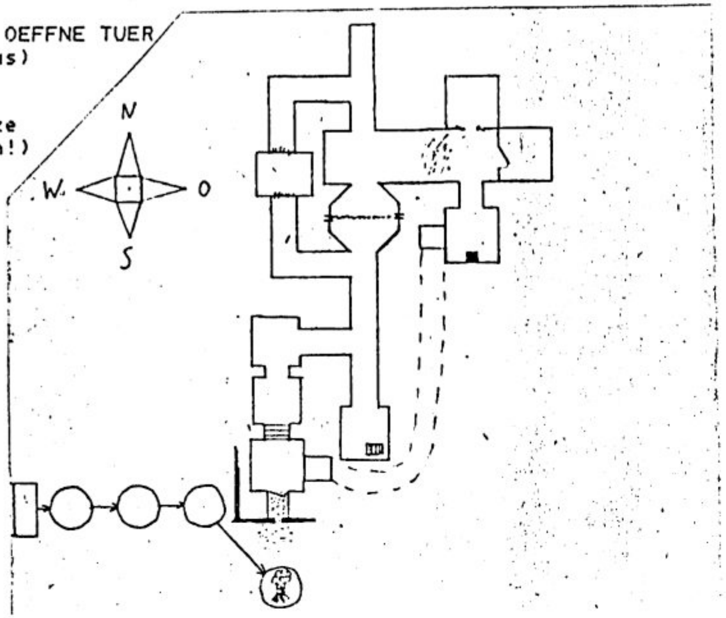
Das Ziel des Adventures GEHEIMAGENT XP-05 ist es:
----

1. Eindringen in das Haus und
2. Finden des Microfilmes
3. Verlassen des Hauses

Der Wortschatz:
LADE (Pistole) / GREIF / SCHLIESS (z.b. Tür) / NIMM / BEOBACHTE / WARTE / KLETTER / SPRING / GEH / RENN / SCHLEICH / LAUF / LIES / DRUECK / ISS / LEG / WIRF / OEFFNE / SCHAU / SCHIESS / AUSRUESTUNG (was habe ich dabei)

Und nun zur Lösung ! :

1. Im Wald : GEH OST
2. Im Wald : GEH OST
3. Im Wald, Im Osten eine Hecke : GEH OST
4. An einer Hecke : GEH SUED (Zum Eingang)
5. Hinter einem Baum, vor dem Eingang, der von einem Agenten bewacht wird : WARTE
6. Solange 'WARTE' eingeben bis der Agent den Rücken zudreht; dann : SCHLEICH NORD
7. Der Agent dreht sich um : GREIF AN
8. Wenn der Agent überwältigt ist, dann NIMM PISTOLE ansonsten von Vorne anfangen
9. Am Eingang zum Grundstück : GEH NORD (zur Jagdhütte; nicht auf den Rasen, da dort Tretminen versteckt sind)
10. Vor der geschloßenen Tür zur Hütte : OEFFNE TUER
11. GEH NORD (in die Hütte)
12. In der Hütte : GEH NORD (die Treppe hinab; nicht nach Westen, da der Fahrstuhlschacht leer ist)
13. In einer Kammer : GEH NORD (Dose Kaviar nicht aufmachen, da diese Schlafgas enthält)
14. In einem Gewölbe : GEH OST (der einzige Ausgang)
15. In einem Nord-Süd Gang : GEH SUED
16. In einem quadratischen Raum mit Munitionskiste: NIMM MUNITION
17. LADE PISTOLE
18. GEH NORD (zurück)
19. GEH NORD (Jetzt der Nord-Teil des Ganges)
20. Weggabelung (N-W) : der West-Weg führt zum Kopf einer Rakete, ist aber ohne Bedeutung. Der Kaugummi kann ohne Bedenken gegessen werden (bedeutungslos). Also: GEH NORD
21. In einem achteckigen Raum : Achtung, die seltsamen öffnungen sind Lichtschranken ! Also : SPRING NORD
22. Im Nordeingang dieses Raumes: GEH NORD
23. An der Westseite einer Halle : GEH NORD (Ostteil kommt später)
24. Weggabelung : Im Norden kommt eine Sackgasse. Also : GEH WEST
25. Ecke : GEH SUED
26. Am Triebwerk der Rakete; Schlüssel : NIMM SCHLUESSEL
27. GEH NORD (zurück)
28. GEH OST (zurück)
29. GEH SUED (zur Halle)
30. In der Halle : GEH OST (die Ostseite)
31. Zuerst die Tür : OEFFNE TUER
32. GEH OST (hinein)
33. In der Schaltzentrale, roter und grüner Knopf: DRUECK ROTEN KNOPF (nicht den grünen, da die Rakete sonst startet)
34. Alles normal, also wieder hinaus : Die Tür ist verschlossen! Jetzt der Schlüssel : SCHLIESS TUER AUF
35. OEFFNE TUER
36. GEH WEST (hinaus)
37. Oha ! Wand ist eingestürtzt und Rückweg verschüttet: GEH NORD
38. Im Kerker; ein Skelett und ein Zettel : LIES ZETTEL (die Nummer ist wichtig ! Merken !)
39. GEH SUED (hinaus)
40. Jetzt zum Ausgang : GEH SUED
41. Büroraum mit Tresor an der Südwand : GEH SUED
42. Alarm! 3 Agenten greifen an! Also die Pistole : SCHIESS
43. Solange SCHIESSen bis alle Agenten erledigt sind
44. Nach gewonnenem Kampf zum Tresor : GEH SUED
45. Vor dem Tresor : die Nummer des Zettels eingeben!
46. Vor dem offenen Tresor, in dem DER MICROFILM liegt: NIMM FILM
47. Was nun ? : SCHLIESS TUER
48. Geheimtür öffnet sich und Aufzug wird frei : GEH WEST
49. Aufzug fährt nach oben, Tür öffnet sich : GEH WEST (aus dem Aufzug)
50. In der Hütte : OEFFNE TUER
51. GEH SUED (hinaus)
52. GEH SUED
53. GEH SUED
(Das war die letztze Anweisung! Gewonnen!)



14. Sep. 1984

Holger May

Marienstr. 9
5768 Sundern 2

Tel.02935/1668

Holger May, Marienstr. 9, 5768 Sundern 2

13.9.1984

An den
CLUB 80 / Guenther Wagner
Gartenstasse 4

8201 Neubeuern

Ihr Schreiben:	Ihr Zeichen:	Mein Schreiben:	Mein Zeichen:
20.8.1984			HM/pr

Betr.: Programmtausch mit dem Club

Lieber Guenther,

Wenn ich jetzt unverschaemt waere, wuerde ich fragen, ob der Club noch existiert, oder ob er sich schon aufgeloest hat.
Ich schickte Dir vor ca. drei Wochen eine Diskette mit Programmen fuer die Bibliothek, und wuenschte mir gleichzeitig Prgme. aus dieser. Bis jetzt warte ich vergeblich auf meine Disk.
Ich wuerde mich freuen, wenn Du mir die Verzoegerung erklaeren wuerdest, und mir natuerlich dann auch die Disk schicken koenntest. Falls Du in der Zwischenzeit die Disk abgesandt hast, will ich alles vergessen, und Du kannst diesen Brief als bedeutungslos ansehen.
In Erwartung Deiner Antwort verbleibe ich

mit vielen Gruessen

Holger

Hallo Leute, dieser Brief hat mich geärgert. Im Antwortschreiben wies ich auf den Urlaub und auf meine Arbeit hin. Ferner verwies ich auf Clubinfo Nr. 1, wo ich um Verständnis bat, wenn mal etwas länger dauert. Holger schickte mir eine Diskette, welche ich am 31. August erhielt. Bereits am 13. September schrieb er diesen Brief. Ich habe ihm nicht mitgeteilt, daß Antwortschreiben, Diskette und Programme zusammen mit dem 3. Info kommen. Ich wollte Porto sparen (seine Programme waren sogar schon archiv.) Im Antwortschreiben wies ich sogar darauf hin! Nun lest bitte das 2. Schreiben vom 18.9.84 und teilt mir dann bitte mit, was ihr so denkt. Sollte ich auf Portokosten achten, oder sollte ich alles so bald wie möglich zusenden? Übrigens: Die Rechnung stimmt wie und nimmer! Mittlerweile sind es 7 Disketten + 2 Disketten (diese beschädigte die Bundespost), als der Hartmut mir diese zurücksandte. Der Hartmut testet alle Programme, die (auf meinem Model III nicht laufen).

20. Sep. 1984

Holger May

Marienstr. 9
5768 Sundern 2

Tel.02935/1668

Holger May, Marienstr. 9, 5768 Sundern 2

18.9.1984

An den
CLUB 80 / Guenther Wagner
Gartenstrasse 4

8201 Neubeuern

Ihr Schreiben:	Ihr Zeichen:	Mein Schreiben:	Mein Zeichen:
14.9.1984			HM/pr

Betr.: Deine Postkarte

Lieber Guenther,

ich moechte mich mit Dir nicht streiten, aber, dass Du in der Clubkasse keine 1.90 DM hast, um die Disk zurueckzuschicken, kann ich Dir nicht abnehmen.
Pro Mitglied hast Du folgende ungefaehre Ausgaben gehabt:

Porto	ca. 2.60 DM
Kopien	ca. 5.20 DM
Chip-Anzeige	1.50 DM
2 Disks	ca. 0.60 DM
naechste Info	ca. 3.90 DM
Gesamt	ca.13.80 DM

Da nun pro Mitglied entweder 20 DM oder 30 DM eingegangen sind, hast Du pro Mitglied ein ungefaehres Plus von ca. 6.20 DM oder ca. 16.20 DM.
Ich glaube, dass davon ruhig ein paar Portokosten ausserhalb der fuer die Infos bezahlt werden koennen. Es bleibt auch immer noch genug Geld in der Kasse, um Buecher, Programme oder aehnliches zu kaufen.
Falls Du immer noch dabei bleibst, die Disk mit dem Info zu schicken, kann ich nur hoffen, dass das Info bald kommt. (Bitte sorge dafuer, dass die Disk auch heil hier ankommt, wenn Du sie zum Info packst!)

mit besten Gruessen

Holger

Springertour D 0151 A 04 /BAS 16

Das bekannte Schachproblem koennen Sie auf dem Computer probieren, wobei der Computer die Zuege kontrolliert.

Timer NE 555 S 0152 A 04 /BAS 16

Berechnung des NE 555 als astabile Schaltung.

Visisort S 0153 A 03 /BAS 16

Demonstriert den Ablauf von 3 verschiedenen Sortierroutinen.

Formlist U 0154 A 05 /BAS 16

Dieses Programm ermoeglicht ein formatiertes Ausdrucken von BASIC-Programmen. Diese muessen als ASCII-Files abgespeichert sein.

Ganzrationale Funktion M 0155 A 05 /BAS 16

Dieses sehr gute Programm dient zur Bestimmung ganzrationaler Funktionen durch gegebene Punkte der Funktion und ihrer 1. und 2. Ableitungen. Die Funktion kann am Bildschirm dargestellt werden.

Mondlandung G 0156 A 03 /BAS 16

Ein Mondlande-Spiel mit bewegter Grafik.

Neatlist und Neatlist 1 U 0157 A 05 /BAS 16

Beide Programme ermoeglichen das formatierte Ausdrucken von BASIC-Programmen. Neatlist muss aber vorher mit dem auszudruckenden Programm erst gemergt werden. Neatlist ist die gepackte Version von Neatlist 1

Olympic Decathlon G 0158 A 05 /CMD 16

Olympischer Zehnkampf fuer 1-8 Spieler auf dem Computer. Ein sehr gutes und interessantes Spiel mit guter Grafik.

PACK 48 U 0159 A 05 /CMD 16

Das sehr gute Programm erlaubt es u.a., BASIC-Programme sowohl enorm zu kuerzen (Zusammenfassung von Programmzeilen) als auch zu strecken.

Umrechnungsprogramm Physik M 0160 A 04 /BAS 16

Ein gutes Umrechnungsprogramm fuer physikalische Einheiten.

Zweiseitiges Beschreiben von Disketten

Gespeicherte Daten auf Floppy-Disk unterscheiden sich im Aufzeichnungsformat und in der Anzahl der beschriebenen Seiten. Das verwendete Aufzeichnungsformat single- oder double density – einfache oder doppelte Schreibdichte – ist abhängig von der Art des verwendeten Laufwerks. Ob eine oder beide Seiten der Diskette mit Daten beschrieben wird, hängt davon ab, ob das Laufwerk mit einem oder zwei Schreib-/Leseköpfen ausgestattet ist.

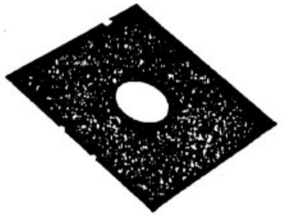
Ein Laufwerk mit nur einem Schreib-/Lesekopf benutzt also nur eine Seite der Diskette. Besteht demnach nur die Möglichkeit, durch Umdrehen der Floppy-Disk auch die zweite Seite mit Daten zu beschreiben? Die Antwort auf diese Frage müßte eigentlich von Radio Eriwan stammen: „Im Prinzip ja". Das heißt, man kann es auf einen Versuch ankommen lassen. Und wenn es klappt, kann man durchaus einige Mark sparen und jede Diskette doppelt verwenden. Einige Probleme dürften allerdings nicht außer acht gelassen werden.

Wenn man sich eine Diskette genau betrachtet, findet man auf der rechten Seite neben der Mitte eine kleine Öffnung, das Indexloch. Dreht man die Diskette, findet man an derselben Stelle auch ein kleines Loch im magnetischen Datenträger. Die Elektronik des Laufwerks kontrolliert an dieser Stelle über eine Lichtschranke die exakte Umdrehungsgeschwindigkeit der Diskette. Steckt man jetzt die Floppy verkehrt herum in das Laufwerk, funktioniert diese Kontrolle nicht mehr.

Erster Schritt bei der beidseitigen Nutzung ist also, das symmetrische Anbringen eines zweiten Indexlochs auf der linken Seite. Am einfachsten geht man dabei vor, indem man die Abmessungen auf die andere Seite überträgt und mit einem weichen Stift markiert. Anschließend locht man die Plastikummantelung an dieser Stelle und bringt zwei gleich große Löcher auf der Vorder- und Rückseite an. Durch Drehen des magnetischen Trägermaterials muß jetzt durch das kleine Loch durch die Diskette hindurch gesehen werden können.

Die zweite Unsymmetrie an der Floppy-Disk ist die Schreibschutzkerbe. Auch diese Öffnung muß symmetrisch auf die andere Seite übertragen werden, um die Diskette beidseitig beschreiben zu können.

Um sicher zu gehen, daß der magnetische Datenträger auf der zweiten Seite keine physikalischen Schäden aufweist, muß man Disketten mit der Bezeichnung „double-side" verwenden. Nur dann ist gewährleistet, daß Daten ohne Verlust gespeichert werden können.



Schwierigkeiten lassen sich nur dadurch vermeiden, indem man eine leere Diskette verwendet und erst einmal testet, ob das Verfahren funktioniert. Durch die magnetische Speicherung kann es nämlich vorkommen, daß das Beschreiben der einen Seite die Daten auf der anderen Seite verfälscht. Am besten speichert man irgendwelche Testdaten zuerst auf die eine Seite, dreht dann die Diskette und schreibt die Rückseite voll. Dann liest man wieder von beiden Seiten und überprüft die Daten auf ihre Korrektheit.

Erst wenn dieser Versuch erfolgreich klappt, sollte man wichtige Daten und Programme auf dieser Diskette abspeichern.

Ein gewisses Risiko, einmal Daten oder Programme zu verlieren, bleibt. Jeder muß selbst entscheiden, ob er es aufnimmt, um einige Mark bei der Anschaffung von Disketten zu sparen.

Zur 1. Frage des letzten Info's

Doppelseitiges Speichern auf einseitigen Disketten

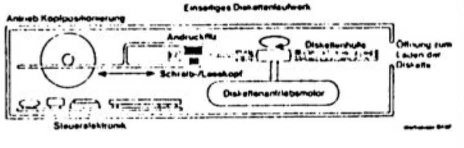
So mancher Anwender ist schon auf die Idee gekommen, zwecks Geldersparnis einseitige Disketten so „umzufunktionieren", daß auch die zweite Seite genutzt werden kann. Denn Disketten werden, je nach Anwendung, in einseitigen und kaum teureren doppelseitigen Versionen angeboten. Aber zahlt sich dies wirklich aus? Der Hersteller BASF hat hierzu Stellung genommen:

„Während und nach der Herstellung prüft die BASF ihre Disketten auf Funktionssicherheit. Bei allen Disketten sind beide Seiten der Trägerfolie mit Magnetpigment beschichtet. Zur Herstellung einseitiger Disketten wird jedoch nur eine Seite geprüft – und fehlerhafte Magnetscheiben werden automatisch aussortiert. Ändert ein Anwender eine einseitige in eine zweiseitige Diskette, so ist die zweite Seite ungeprüft und möglicherweise fehlerhaft.

Versucht man, diese „Kapazitätserweiterung" bei schon mehrfach genutzten einseitigen Disketten anzuwenden, wird das Risiko noch größer. In einem Einkopf-Laufwerk schleift die zweite Diskettenseite ständig über einen Andruckfilz. Er drückt die Magnetscheibe an den Schreib-/Lesekopf (*Bild*). Ein verschmutzter Andruckfilz beschädigt die vom Konstrukteur des Laufwerks nicht zur Datenspeicherung vorgesehene zweite Diskettenseite bis zum Datenverlust.

Die hohe mit BASF-Disketten erzielte Datensicherheit, die vom Anwender als selbstverständlich vorausgesetzt wird, läßt sich durch solche Handgreiflichkeiten nicht mehr gewährleisten."

Schreib-/Lesekopf und Andruckfilz bei einseitigen Diskettenlaufwerken



neue BÜCHER

Nr. 0006: Games For Your TRS-80

Chris Palmer --- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment.

Disketten doppelt genutzt: Bitte wenden

Mit einem einfachen Trick lassen sich einseitige 5¼-Zoll-Disketten so herrichten, daß sie auf beiden Seiten genutzt werden können.

Jede Diskette trägt auf beiden Seiten eine Magnetschicht, die zur Speicherung von Daten dienen kann. Einseitig arbeitende Laufwerke, die mit nur einem Magnetkopf arbeiten, können aber immer nur eine Seite beschreiben und lesen. Die Hälfte der möglichen Speicherkapazität der Diskette bleibt also ungenutzt.

Daß man eine 5¼-Zoll-Diskette nicht einfach wenden kann, liegt zum einen an der fehlenden Schreibschutzkerbe, zum anderen an dem fehlenden Indexloch auf der gegenüberliegenden Seite. Eine zweite Schreibschutzkerbe läßt sich leicht mit einer Schere schneiden. Nicht ganz so einfach ist aber das Anbringen eines zweiten Indexlochs, die Magnetfolie darf dabei nämlich nicht beschädigt werden.

Es gibt zwar spezielle Zangen, die hier Hilfestellung geben. Sie sind aber recht teuer. Es besteht auch die Möglichkeit, die Diskettenhülle zu öffnen und die Diskette vor dem Lochen vorsichtig herauszuziehen. Das ist jedoch erstens umständlich und zweitens nicht ganz problemlos, da ein Knick oder bereits ein Fingerabdruck die Diskette unbrauchbar machen kann.

Sehr viel einfacher ist das Herausschneiden einer passenden Lasche an der Position für das zweite Indexloch, und zwar auf beiden Seiten der Diskettenhülle. Dazu macht man von der Mitte her, also von der Antriebsöffnung der Diskette aus, mit einer Schere zwei passende, leicht schräg verlaufende Schnitte. Die so entstandene Laschen klappt man hoch und schneidet sie ab. Die Stabilität der Diskettenhülle wird durch diesen Eingriff auf keinen Fall beeinträchtigt.

Die zweite Schreibschutzkerbe braucht man nur entsprechend herauszuschneiden. Vorsicht: Manche Disketten haben eine leicht fasernde Beschichtung im Inneren der Hülle. Eine scharfe Schere ist daher empfehlenswert.

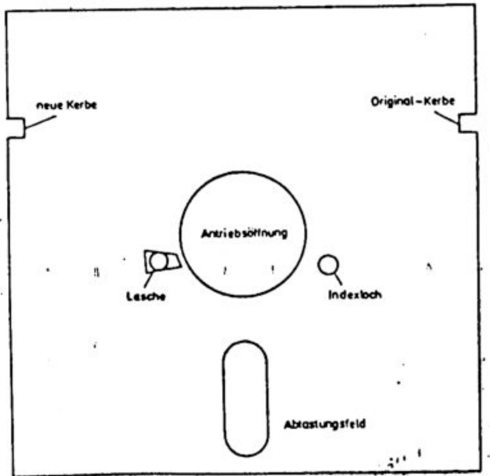
Nicht entnehmen

Die Diskette braucht man für diese Operation nicht aus ihrer Hülle herauszunehmen. Die Schere kommt nämlich nicht mit dem Teil der Magnetfläche in Berührung, der vom Magnetkopf des Laufwerks beschrieben wird. Allerdings sollte man keine Diskette für doppelseitigen Gebrauch präparieren, die wichtige Daten enthält. Eine etwaige Magnetisierung der Schere könnte diese Information zerstören. Allerdings ist diese Gefahr in der Praxis nicht allzu groß.

Von rund 200 Disketten der verschiedensten Hersteller, die auf diese Weise behandelt wurden, erwies sich nur zweimal die Rückseite als nicht brauchbar. Alle anderen Disketten arbeiteten anschließend au beiden Seiten einwandfrei, und zwar mit doppelter Dichte, obwohl keine von ihnen für doppelseitigen Gebrauch bestimmt war.

Übrigens verwenden manche Computer das Indexloch nicht. Für sie braucht man auch kein zweites Indexloch in eine Diskette zu schneiden, wenn man sie auf beiden Seiten verwenden will. Das gilt zum Beispiel für den Apple II oder auch für das Laufwerk 1541 für den C64 von Commodore.

Dr. Bernd Enders



So wird eine Diskette doppelseitig

Ein Mitglied stellt sich vor:

Mein Name ist Peter Stevens, geboren am 8.11.48 in London, England. Nach vielen Jahren der Wanderung und Selbstfindung bin ich 1973 nach Deutschland gekommen und arbeite als Heilpädagoge m.ü.L. Meinen TRS-80 habe ich mir erst im letzten Jahr besorgt. Mit Basic bin ich allmählich vertraut und versuche mich (leider nicht oft genug) mit Assembler anzufreunden. Ich arbeite sowohl mit TRSDOS als auch mit NEWDOS. Na das wärs!

*** B Ü C H E R *** B Ü C H E R *** B Ü C H E R ***

Der Peter hat dem Club 2 Bücher gespendet - Dankeschön! Zu verleihen hat der Peter folgende Bücher:

- How to do it on TRS80 & other Mys.
- The custom TRS & other Mys.
- TRS80 Disk & other Mys.
- Basic faster and better & other Mys.
- The first book of 80 US
- Programmieren mit TRS80 (Stübs)
- Anwenderprogramme für TRS80 (Stübs)
- TRS Programming (Radio Shack)
- TRS80 Model III ROM Listing (Röckrath)

Einige deutsche Anleitungen hat der Jens Neueder:

- Electric Pencil (Kurzzusammenfassung)
- Scripsit (" ")
- ESF-80 Monitor (" " für Stringy Floppy)
- NEWDOS 80 Vers. 2.0 (Anleitung)
- Packer (Anleitung)
- Basic + (Anleitung)

Folgende Bücher verleiht der Konrad Josef:

- Mikroprozessor Interface Techniken (Zaks R.)
- TRS-80 Assembly Language Programming (Barden W.)
- Programmieren mit TRS-80 (Stübs M.)
- Programmieren mit Pascal (Baumann R.)
- Technical Reference Book (Radio Shack)
- The first Book of 80 US
- Zeitschriften: 80 US (TRS-80 User Journal) - 1/82 bis 8/82

Folgende Liste hat mir der Walter Zwickel zugeschickt:

- Programmieren in Maschinensprache mit Z-80 (C. Lorenz)
- IC-Handbuch digital und analog (C. Lorenz)
- TRS-80 Programms (32 Programme; Radio Shack)
- Anwenderprogramme für TRS-80 und VG. (M. Stübs)
- 1978 Interface Databook (NS)
- Voltage-regulator Handbook (NS)
- Transistor-Berechnungs u. Bauanl.-Handbuch (Hofacker)
- Forth-Handbook (Einführung und Beispiele; E. Flögel)
- The fourth book of ohio scientific (S. Roberts)

Dieter Kasper verleiht folgendes Buch:

- Programm. i schinensprache mit Z80 - Band 2 (Dr. Schmitter)

Bei den meisten heutigen Hobby-Computern wird die Programmiersprache BASIC verwendet. Das heißt aber (leider) nicht, daß ein BASIC-Programm zwischen zwei verschiedenen Computertypen direkt oder über Kassette ausgewechselt werden kann. Die BASIC-Befehle mögen dann wohl das gleiche bedeuten, aber die interne Verarbeitung dieser Befehle und die Speicherung auf der Kassette ist häufig ganz anders. Nichts geht mehr . . . ? Doch: Mit Hilfe von BASICODE, einer Art universeller, aus BASIC abgeleiteter Computersprache, ist es möglich, BASIC-Programme zwischen verschiedenen Computertypen auszutauschen.

BASICODE-2

Endlich ein "allgemeinverständliches" BASIC

Es ist ungefähr zwei Jahre her, daß die Hobbyskoop-Redaktion von Radio Hilversum in den Niederlanden auf die Idee kam, einen universellen Kode auszuarbeiten, mit dem man BASIC-Programme zwischen verschiedenen Computern austauschen kann. Denn daß BASIC nicht gleich BASIC ist, merkt man schnell, wenn man zum Beispiel ein Junior-Programm auf dem ZX-81 laufen lassen will. Obwohl es ein Standard-BASIC gibt, verwendet jeder Computer-Hersteller seinen eigenen "Dialekt" mit seinen eigenen (meist durchaus sinnvollen) Erweiterungen. Der Unterschied zwischen ostfriesischem Platt und Ur-Bajuwarisch ist nichts dagegen . . . Dazu kommen Probleme mit Speicheradressen und der internen Programmabarbeitung und als eines der wesentlichsten Probleme das der Übertragung. Bei den meisten Hobbycomputern wird der Kassettenrekorder als externer Massenspeicher verwendet. Die Speichermethodik (das Datenformat), die Übertragungsgeschwindigkeit und die gebrauchten Frequenzen sind auch wieder bei jedem Computertyp anders. Da half bisher nur eins: sich hinzusetzen, das Programm umzuschreiben und dann mühsam Zeichen für Zeichen in den eigenen Computer einzutippen.
Der BASICODE-Standard ist ein festgelegter Audio-Kode, mit dem BASIC-Programme auf der Kassette gespeichert werden können. Dabei ist es im Prinzip unwichtig, ob diese Programme vom Computer oder vom Radio oder vom Fernseher angeliefert werden. Denn daß diese drei "Lieferanten" in Frage kommen, wissen aufmerksame Elektor-Leser aus dem September-Heft. Durch die Standardisierung sind Abspeichern und Wiedereinlesen unabhängig vom Computertyp. Die einzelnen Festlegungen, auf die wir unten noch eingehen, betreffen:

- Übertragungsformat, -geschwindigkeit, -frequenzen
- zulässiger BASIC-Befehlssatz
- Speicheraufteilung
- Variablenbenennung
- Schirmformat

Das neue, universellere BASICODE-2 enthält neben anderen Verbesserungen vor allem eine Anzahl Standard-Unterprogramme. Dieses BASICODE-2 werden wir hier besprechen.

BASICODE auf dem Band

Beim BASICODE werden die beiden Frequenzen 1200 Hz und 2400 Hz verwendet. Eine volle Periode der 1200 Hz bedeutet eine logische 0, zwei volle Perioden der 2400 Hz eine logische 1.
Die einzelnen Bytes werden mit einer Geschwindigkeit von 1200 Baud gesendet beziehungsweise empfangen, wobei die Bytes folgendermaßen aufgebaut sind (siehe auch Bild 1):

- 1 Startbit (logisch 0)
- 8 Datenbits, beginnend mit dem LSB
- 2 Stoppbits (logisch 1).

Das BASIC-Programm wird Zeichen für Zeichen kodiert, also so, wie es beim Kommando LIST zu sehen ist. Von den internen Op-Codes des Computers wird kein Gebrauch gemacht: Alle Zeichen werden im ASCII-Kode dargestellt. Jedem BASIC-Befehl muß eine Leertaste folgen, jede BASIC-Zeile muß mit CR (Wagenrücklauf, Hex 8D) abgeschlossen werden. Das MSB jedes ASCII-Zeichens muß logisch 1 gemacht werden.
Ein vollständiges Programm auf der Kassette besteht aus folgenden Teilen:

- Ein "leader", bestehend aus einem 5 Sekunden langen Ton von 2400 Hz
- dem ASCII-Zeichen "start text" (Hex 82)
- dem BASIC-Programm in ASCII-Kode
- dem ASCII-Zeichen "end of text" (HEX 83)
- einer "checksum" (Prüfsumme)
- einem "trailer", bestehend aus einem 5 Sekunden dauernden Ton von 2400 Hz.

Die Prüfsumme ermöglicht es dem Anwender zu kontrollieren, ob das Programm fehlerlos eingelesen wurde. Sie wird durch die bitweise Exklusiv-ODER-Verknüpfung aller ihr vorangehenden Bytes einschließlich des "start text"-Zeichens gewonnen. Daher hat sie selber auch die Länge von 1 Byte (8 bit).

Das BASICODE-2-Protokoll

Allgemeine Absprachen

Es dürfen nur die BASIC-Befehle eingesetzt werden, die alle Computer verwenden. Diese Befehle sind in Tabelle 1 aufgelistet, wir kommen noch darauf zurück. Eine bestimmte Anzahl Programmzeilen ist für vorgegebene Unterprogramme reserviert. Diese Routinen ermöglichen Programmabläufe, die im Standard-BASIC nicht ohne weiteres möglich sind. Die Arbeitsweise dieser Unterprogramme wird noch beschrieben. Sie werden nicht mit dem Programm mitgesendet, sondern müssen im BASICODE-Übersetzungsprogramm enthalten sein oder vor dem Programmablauf gesondert eingegeben werden.
Das Schirmformat ist festgelegt auf 24 Zeilen zu je 40 Zeichen. Da manche Computer noch weniger Zeilen beziehungsweise Zeichen abbilden, wird empfohlen, nicht mehr als 16 Zeilen für ein Schirmbild zu verwenden und die Zeilen nicht länger als nötig zu machen. Ferner darf eine Programmzeile einschließlich Zeilennummer, Leerzeichen und CR maximal 60 Zeichen lang sein.

Programm-Aufbau

Bei den Zeilennummern wird bei BASICODE-2 folgende Einteilung eingehalten:
0-999: Standard-Unterprogramme. Diese Routinen sind speziell für den betreffenden Computer entworfen und werden mit dem Übersetzungsprogramm oder gesondert eingelesen.
1000: Erste Zeile des Hauptprogramms. Es muß folgende Form eingehalten werden:
1000 A = (Wert): GOTO 20: REM Programmname. (Wert) ist die maximale Anzahl Zeichen, die von allen Strings zusammen benötigt werden. Durch den Sprung nach Zeile 20 reservieren die Computer, bei denen dies notwendig ist, entsprechenden Speicherraum für die Strings.
1010: Erste frei zur Verfügung stehende Programmzeile.
1010-32767: Platz für die eigenen Programme. Eine bestimmte Systematik in diesem Programm ist zwar nicht Pflicht, wird aber von den BASICODE-Entwicklern empfohlen:
1000-19999: Hauptprogramm.
20000-24999: Unterprogramme mit Befehlen, die im BASICODE-2 nicht zulässig sind.
25000-29999: DATA-Statements.
30000-32767: REM-Statements. Diese können eine Beschreibung des Programms, Literaturhinweise und als letztes Namen und Anschrift des Programmverfassers enthalten.
Weiter wird empfohlen, die Zeilennummern in 10er-Schritten zu setzen. Über die Unterprogramme in den Reihen 20000-24999 muß man anmerken, daß sie natürlich nach Möglichkeit zu vermeiden sind. Wenn dies nicht geht, muß man wenigstens eine gute Beschreibung mitliefern, was in diesen Unterprogrammen geschieht.

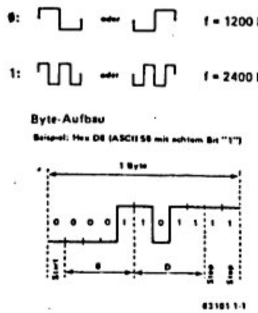


Bild 1. Der Aufbau des Übertragungsformates bei BASICODE. Achtung: Die Übertragung beginnt mit dem LSB (least significant bit = niederwertigstes Bit)!

Standard-Routinen in BASICODE-2

Wie diese Unterprogramme jeweils aussehen, ist stark vom jeweiligen Computertyp abhängig, so daß wir hier nur eine allgemeine Beschreibung geben.
GOSUB 100: Dieses Unterprogramm löscht den Schirm und setzt den Cursor auf Position 0,0 (links oben auf dem Schirm).
GOSUB 110: Setzt den Cursor auf eine bestimmte Stelle des Schirms. Die gewünschte Position wird durch die Variablen HO und VE festgelegt. VE gibt die Nummer der Bildschirmzeile an (die oberste Reihe hat die Nummer 0) und HO den Platz in der Reihe (außen links ist 0). Da das Schirmformat bei BASICODE-2 24 Reihen zu je 40 Zeichen ist, darf VE nicht größer als 23 und HO nicht größer als 39 werden. Die Variablen VE und HO dürfen durch den Aufruf dieses Unterprogramms ihren Wert nicht ändern.
GOSUB 120: Die Position des Cursors auf dem Schirm wird durch die Variablen VE und HO gespeichert. Hierbei ist VE = 0 die oberste Reihe und HO = 0 die erste Position in der Reihe (ganz links). Dieses Unterprogramm kann zusammen mit dem vorigen dazu eingesetzt werden, die Position des Cursors per Programm beliebig zu verändern (zum Beispiel bei selbstgestrickten Bildschirmspielen).
GOSUB 200: Prüft, ob eine Taste gedrückt ist, und setzt den zu dieser Taste gehörigen Inhalt in die Variable IN$. Ist in diesem Moment keine Taste gedrückt, ist IN$ leer. Im Prinzip kann man hier auch Steuerzeichen einlesen, aber dabei muß man vorsichtig sein, da diese bei manchen Computern eine unterschiedliche Bedeutung und Wirkung haben! Eine Ausnahme dabei ist RETURN. Das ist bei allen Heimcomputern der ASCII-Code 13.
GOSUB 210: Dieses Unterprogramm wartet, bis eine Taste gedrückt wird, und setzt dann den zugehörigen Inhalt in IN$. Vorsicht beim Programmieren, sonst wird das die berühmte Endlos-Warteschleife.
GOSUB 250: Erzeugt bei den Computern, bei denen dies möglich ist, einen Piepser. Tonhöhe und Tondauer sind hierbei nicht festgelegt.
GOSUB 260: Eine Zufallszahl zwischen 0 und 1 wird erzeugt und in die Variable RV eingeschrieben.
GOSUB 270: Der Variablenspeicher wird überprüft, und es wird festgestellt, wieviel Speicherplatz noch zur Verfügung steht. Die Variableninhalte werden dabei nicht verändert. Die Anzahl der freien Bytes wird in die Variable FR eingelesen.
GOSUB 300: Der Wert der Variablen SR wird als String in SR$ eingelesen. Der String darf kein Leerzeichen am Anfang oder Ende der Zahl enthalten (im Gegensatz zum Statement STR$, dieses Statement darf bei BASICODE-2 nicht verwendet werden).
GOSUB 310: Dieses Unterprogramm erzeugt einen String SR$, der folgendermaßen aufgebaut ist:
Der Wert von SR$ ist gleich dem Inhalt der Variablen SR und immer in Festkomma-

Notation. Die Gesamtlänge von SR$ beträgt CT Zeichen, die Anzahl der Zeichen nach dem Dezimalkomma wird durch CN festgelegt. Wenn die Zahl nicht in das vorgegebene Format paßt, besteht SR$ aus CT Sternen. Wenn nötig, wird SR entsprechend gerundet. CT, CN und SR werden durch den Zugriff auf dieses Unterprogramm nicht verändert. Ein Beispiel: CT = 7, CN = 3 und SR = 0,6666 ergibt in SR$ " 0,667".
GOSUB 350: Printed SR$, schließt aber die Zeile noch nicht ab. Dadurch ist es möglich, verschiedene Strings hintereinander in eine Zeile zu schreiben.
GOSUB 360: Abschluß einer Schreibzeile mit einem Wagenrücklauf- und Neue-Zeile-Kommando.

Variable

Um die Auswechselbarkeit der Programme zu garantieren, sind auch die Variablen bestimmten Festlegungen unterworfen.

- Numerische Variablen sind immer reell und mit "single precision".
- Der Name einer Variablen darf höchstens aus zwei Zeichen bestehen, wobei das erste Zeichen ein Buchstabe sein muß. Das zweite Zeichen kann, wenn nötig, aus einem Buchstaben oder einer Ziffer bestehen. Bei String-Variablen steht noch ein $ nach dem Namen. Kleinbuchstaben dürfen in Variablennamen nicht verwendet werden.
- Bei logischen Variablen wird lediglich auf wahr oder unwahr geprüft. Eventuell vom Computer erzeugte Werte, zum Beispiel +1 für wahr und 0 für unwahr, dürfen nicht verwendet werden.
- Man darf nicht davon ausgehen, daß eine Variable durch den Programmstart automatisch auf 0 gesetzt wird.
- Stringvariablen dürfen maximal 255 Zeichen lang sein.
- Namen von Variablen dürfen nicht mit dem Buchstaben O beginnen, dieser ist reserviert durch die Standard-Unterprogramme.
- Weiter sind folgende Variablennamen nicht zulässig: AS, AT, FN, GR, IF, PI, ST, TI, TI$ und TO.

Für den Datenaustausch zwischen dem BASIC-Programm und den Standard-Unterprogrammen werden die Variablen HO, VE, FR, SR, CN, CT, RV, IN$ und SR$ verwendet.

BASIC-Absprachen

Tabelle 1 gibt eine Übersicht über alle BASIC-Kommandos und -Operatoren, die verwendet werden dürfen. Auch in diesem Punkt sind strenge Absprachen nötig, da es verschiedene BASIC-Varianten gibt. D Bedeutung der Befehle stimmt mit der offiziellen BASIC-Version überein, so da wir hierauf an dieser Stelle nicht eingehen Einige spezielle Punkte müssen wir bei d BASIC-Befehlen noch besprechen. So da: man nach einem GOSUB oder GOTO keir Variablennamen verwenden. Die Form: A = 1000 : GOTO A darf also nicht verwe det werden. Nach IF muß immer ein THEN folgen. Also: IF ... THEN A = 5 IF ... THEN 1000 oder IF ... THEN GOSUB 20000. Die Form IF ... THEN .. ELSE ist verboten. Kommentare oder mehrere Variablen nach einer INPUT-Anweisung sind nicht zugelassen. Die Forr INPUT "Der Wert = "; A$ ist verboten. Eine Zeilennummer nach RUN ist nicht zulässig. Beim TAB-Befehl muß man berücksichtigen, daß der eine Computer bei Null und der andere bei Eins zu zählen beginnt.

Tabelle 1. Zulässige BASIC-Befehle und -Operatoren.

ABS	DIM	INPUT	NOT	RETURN	STOP
AND	END	LEFT$	ON	RIGHT$	TAB
ASC	FOR	LEN	OR	RUN	TAN
ATN	GOSUB	LET	PRINT	SGN	THEN
CHR$	GOTO	LOG	READ	SIN	TO
COS	INT	MID$	REM	SQR	VAL
DATA	IF	NEXT	RESTORE	STEP	
+	↑	<>			
−	=	<=			
*	<	>=			
/	>				

Die Praxis

So, das waren die wichtigsten BASICODE Absprachen. Um damit zu arbeiten, brauc man jetzt noch das Übersetzungs-Program und die Standard-Unterprogramme. Dazı müssen wir für die "Allgemeinheit" an die Stelle passen: Denn hier gilt (hoffentlich z letzten Mal), daß halt jeder Computer etw anders ist. Glückliche Junior-Computer-Besitzer: Sie finden das Übersetzungsprc gramm und die Standard-Unterprogramn sowohl für den erweiterten Junior mit Mikrosoft-Basic als auch für den DOS-Junior an anderer Stelle in diesem Heft. Und wenn sie die Sendung "Know-how-Special" empfangen können, bekommen sie das auch noch vom Fernsehen auf Kassette geliefert (siehe Elektor im September-Heft). Und die Besitzer der vielen anderen Computer? Wenn alles gut gegangen ist, ist inzwischen das BASICODE Buch Nummer 2 erschienen – allerdings i. niederländischer Sprache. Neben einer au führlichen Beschreibung von BASICODE- enthält es auch die Übersetzungsprogramn für diverse Computertypen. Sobald das Buch der Redaktion vorliegt, werden wi unsere Leser noch genauer informieren.

Müssen nun diejenigen, denen weder Rad Hilversum noch ein Fernsehprogramm noc dieses Elektorheft ein Übersetzungsprogramm liefert, etwa aufgeben? Nun, es gib schließlich viele, auf einen bestimmten Computertyp spezialisierte Computerclut bei denen man einmal anfragen kann.
Natürlich stellt sich jetzt mancher Leser di Frage: Kann ich denn, wo ich als Hamburç schon den WDR nicht empfangen kann, wenigstens Hilversum II empfangen? Leid können wir da auch keine eindeutige Antwort geben, weil dies von verschiedenen "funktechnischen" Bedingungen abhängt Es kann vorkommen, daß ein Elektor-Lese in Bayern einen guten Empfang hat, während ein Leser im viel näher gelegenen Hessen leer ausgeht. Aber was spricht dagegen, durch Leserbriefe an Fernseh-Intendanten dazu beizutragen, daß Know how-Special in alle 3ten Fernseh-Programr kommt?

BASICODE-Interface

Für TRS-80 und Video-Genie

Um Basicode-Programme von der Kassette in den TRS-80 oder den Video-Genie laden zu können, benötigt man ein spezielles Interface, mit dem die FSK-Signale in Rechteckimpulse umgewandelt werden. Die Schaltung (Bild 1) besteht aus einem Impulsaufbereiter-Teil (A1, A2 und "Umgebung"), einem Adreßdekoder (N1 ... N9 und N17) und einem Indikator (A3, D3). Sie wird an der Erweiterungsanschlußleiste des Computers angeschlossen, deren Belegung aus dem jeweiligen Handbuch hervorgeht.

Die Schaltung

Wenn der Ausgang von N17 auf "0" ist, wird das Signal auf die Datenbuslinie D0 durchgegeben. Das Übersetzungsprogramm — es muß vorher eingegeben beziehungsweise über den normalen Kassettenanschluß eingelesen werden — sorgt hier auch für die Umwandlung der seriellen Eingangssignale in "normale Bytes" und deren Abspeicherung. Der Adreßdekoder wird durch das Übersetzungsprogramm so angesteuert, daß beim Einlesen N10 und N11 freigegeben sind.
Die Dioden D1 und D2 begrenzen das Signal vom Kassettenrekorder. Mit A1 ist eine Verstärkerstufe aufgebaut, mit A2 ein Schmitt-Trigger. Der OP A3 dient als Treiber für die Indikator-LED. Ist kein Eingangssignal vorhanden, sorgt R9 dafür, daß der Ausgang von A2 auf "1" liegt. Die LED D3 leuchtet dann nicht. Im Signalweg liegen die beiden invertierenden Treiber N10 und N11, sie werden vom Ausgang des Adreßdekoders gesteuert.

Aufbau

Auf der Platine (Bild 2) finden — mit Ausnahme des Trafos — auch noch die Bauteile für ein kleines Netzteil Platz. Allerdings braucht man diese nur, wenn das

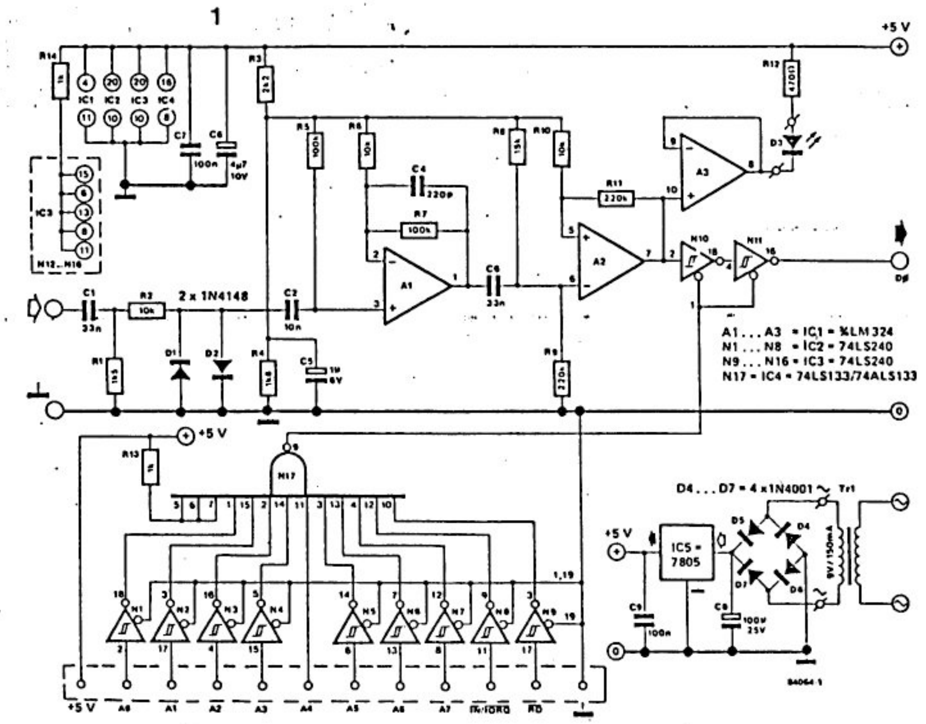


Bild 1. Die Schaltung des BASICODE-Interface für TRS-80 und Video-Genie.

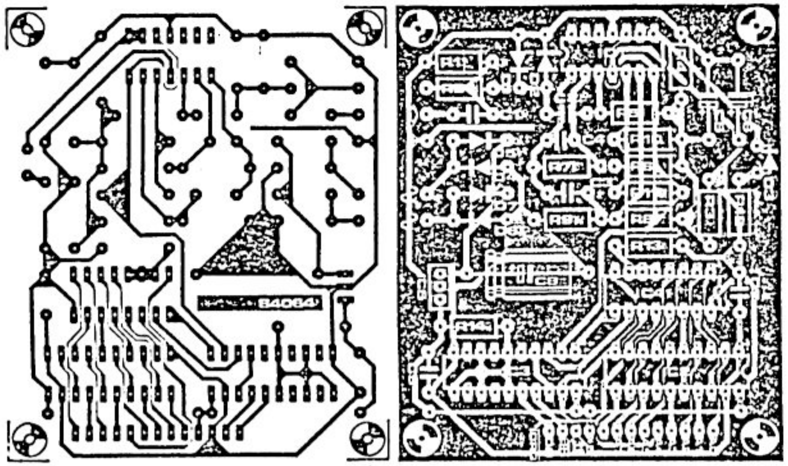


Bild 2. Mit der Platine fällt der Aufbau der Interface-Schaltung leicht.

Netzteil des Computers bereits "anderwei-ig" zusätzlich belastet ist: Die Schaltung les Interface benötigt lediglich 50 mA. In len meisten Fällen kann man also den 5-V-Anschluß auf der Erweiterungs-Anschlußleiste als "Kraftwerk" verwen-len.
Auf der Platine gibt es eine Leiterbahn, die "im Nichts endet". Das ist *kein* Fehler, diese Leiterbahn dient als einfache Abschirmung des Eingangs gegen die direkt daneben anliegende Netzteil-Wechselspannung.
Abschließend noch ein Hinweis: Über das BASICODE-Verfahren ist ausführlich im Oktober-Heft 1983 berichtet worden.

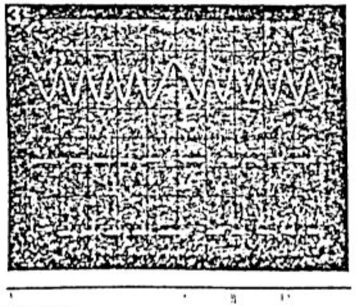


Bild 3. Auf dem Foto erkennt man oben das Eingangssignal und unten das Ausgangssignal der Interface-Schaltung.

Neues von BASICODE

Über das BASICODE-Verfahren haben wir ausführlich in Elektor vom Oktober 1983 berichtet. Das BASICODE-2-Buch (in Niederländisch und Englisch) von Radio Hilversum liegt der Redaktion inzwischen vor. Es enthält die Anpassungen und Übersetzungsprogramme für folgende Computer beziehungsweise Betriebssysteme:
Apple 2, BBC-Computer Modell A und B, Commodore (Pet und VC), CP/M-Systeme, DAI-Computer, Exidy Sorcerer, OSI-1P-Computer, Philips P-2000, Sharp MZ80 A und MZ80 K, SWTPC-6800, TRS-80 (I und III) und Video-Genie. Weitere Anpassungen sind bei Radio Hilversum — genauer: bei der Hobbyskop-Redaktion — in Vorbereitung. Außerdem soll in Kürze bei einem deutschen Verlag ein Sonderheft über BASICODE — ebenfalls mit Anpassungen — erscheinen, hier wissen wir aber noch nicht, für welche Computer (Rubrik Büchermarkt beachten).

Stückliste

Widerstände:
R1 = 1k5
R2,R6,R10 = 10 k
R3 = 2k2
R4 = 1k8
R5,R7 = 100 k
R8 = 15 k
R9,R11 = 220 k
R12 = 470 Ω
R13,R14 = 1 k

Kondensatoren:
C1,C3 = 33 n
C2 = 10 n
C4 = 220 p
C5 = 1 μ/6 V
C6 = 4μ7/10 V
C7,C9 = 100 n
C8 = 100 μ/25 V

Halbleiter:
D1,D2 = 1N4148
D3 = LED
D4,D5,D6,D7 = 1N4001
IC1 = LM 324
IC2,IC3 = 74LS240
IC4 = 74LS133 (oder 74ALS133)
IC5 = 7805

außerdem:
Tr1 = Netztrafo 9 V/ 150 mA sek.
Platine 84064

Abschließend nochmals die Adresse von Radio Hilversum:
NOS-Hobbyscoop
postbus 1200
1200 BE Hilversum
Nederland

So, war's b... jetzt schon unübersichtlich, so wird es jetzt bestimmt fü... ...uch chaotisch - ich gehe jetzt einfach Brief für Brief durch. Das ist die einzige Möglichkeit fertig zu werden und nicht zu viel zu vergessen!

ZUM THEMA BÜCHER :

Nun steht es endgültig fest; sollten wir am Jahresende ein Plus in der Kasse haben, so werden dafür ein oder zwei Bücher angeschafft. Das Photokopieren von Büchern sollte nur geschehen, wenn der Kopienpreis deutlich unter dem Kaufpreis liegt. Viele sind aber bereit, Ihre Bücher an Mitglieder 3-4 Wochen auszuleihen (siehe Liste in diesem Info!).

ZUM THEMA VERLEIHSAMMLUNG :

Auch hier waren alle dafür. Und ich war seit dem 2. Clubinfo auch nicht ganz untätig. Meine Computer-Zeitschriften habe ich fast alle 2-fach kopiert. Dabei kopierte ich alles, was direkt die Hard- und Software für Tandy und Genie angeht (auch z.B. Computer-Tests und Hardware-Erweiterungen). Ferner kopierte ich aber auch viele Programme, die zwar für andere Computer geschrieben sind, sich aber für unsere Computer umschreiben lassen müßten (z.B. Buchhaltungsprogramm, Dateiverwaltungsprogramm, etc). Das ist nun ein Riesenstoß von Blättern, der noch nicht sortiert und eingeordnet ist. Ab 1. November könnt Ihr bei mir die Verleihsammlungen anfordern; ich glaube, es stehen dann 3 Leitz-Ordner (Rückenbreite 5 cm) zur Verfügung.

Es wäre nun gut, wenn auch alle anderen Mitglieder nach und nach sich Ihre Computerzeitschriften vornehmen und alles kopieren, was von Interesse wäre. Ihr seht ja anhand des letzten Info's, welche Zeitschriften nun archiviert sind.

ZUM THEMA PROGRAMME :

Teilweise wurde die Bezeichnung der Programme kritisiert. Sicher - die Bezeichnung hat auch Nachteile. Man muß z.B. erst in die Liste gehen und nachsehen, welche Programmnummer welches Programm ist. Aber es bleibt ja jedem offen, sich mit dem RENAME-Befehl die Programme umzubenennen. Auf jeden Fall hat sich für mich die Bezeichnung bewährt - die Programmnummer (fortlaufend von 1 aufwärts) ist für mich sehr wichtig.

Eine Erleichterung kann ich aber anbieten. Zur Abspeicherung der Programme habe ich ja ein Programm - dieses erlaubt einigermaßen komfortabel das Suchen, Ändern und Löschen von Programmen. Man kann sich z.B. eine Liste aller Geschicklichkeits-Programme ausdrucken lassen, oder nach einem ganz bestimmten Programm suchen lassen. Bis zum übernächsten Clubinfo (Nr. 5) sendet mir jeder eine Diskette zu. Ihr erhaltet dann die Diskette zusammen mit dem 5. Clubinfo. Auf der Diskette befinden sich die Programme für die Programmbibliothek, die Bücherbibliothek und die Clubadressen mit den entsprechenden Dateien. Alle Mitglieder ohne Disketten-Laufwerk müssen auf diesen Service verzichten - Sie erhalten dafür die entsprechenden Ausdrucke.

Nun zum Brief vom Peter Stevens. Er hat die Betriebsbücher (Users manual) für DOSPLUS 3.5 (2 Bände). Falls jemand an diesem Betriebssystem interessiert ist, so soll er es dem Peter wissen lassen. Die Bücher kosten eine Menge Geld. Falls jemand Probleme mit englischen Anleitungen hat, so kann er sich mit dem Peter in Verbindung setzen. Er hat logischerweise beim Englisch keine Schwierigkeiten und ist gerne bereit zu helfen.

Eine gute Anregung stammt noch vom Peter. Er schlägt vor, eine Sammlung für POKE und PEEK zu machen. Er meint, daß es sich sehr lohnen würde und liefert auch gleich ein Beispiel:

```
100 IF PEEK(14312) größer 127 THEN PRINT "DRUCKER ANSCHALTEN!":
    GOTO 100
```

Ich hoffe, Ihr greift diese Idee auf und schreibt mir, daß es nur so POKE'd und PEEK'd.

Der Jens Neueder weist auf Probleme mit der Umlautschreibung hin. Manche Basic-Programme müssen vor dem 1. Programmlauf gelistet werden um Sie an den entsprechenden Stellen zu ändern. Der Jens hat Schaltpläne für das Model I und das Expansion (dieses ist allerdings von der Firma CE-Elektronik).

Nun komme ich zum Schreiben vom Konrad Josef. Er meint, das Info wäre leichter zu lesen, wenn die Rückseite in der gleichen Richtung wie die Vorderseite kopiert wäre. Nun - ich versuche, die Blätter besser zu kopieren, Ihr seht ja bereits schon das Ergebnis. Der Konrad will wissen, ob die Fragen im Info beantwortet werden sollen oder direkt beim Fragensteller. Ich bin dafür, die Antworten im Clubinfo zu veröffentlichen (dann haben alle etwas davon); wer will, kann dem Fragensteller noch extra schreiben, dann braucht der nicht solange auf die Antwort warten. übrigens - wenn man im Mondscheintarif telefoniert, kostet die Minute nur ca 0.20 DM.
Zur Frage von übersetzungen von Anleitungen meint der Konrad, daß dies eigentlich nur sinnvoll ist, wenn mehrere Leute Interesse an einer solchen übersetzung haben (abhängig vom Umfang des Manuals). Man könnte aber vielleicht deutsche Kurzanleitungen für größere Programme erstellen. Falls Interesse besteht, ist er bereit, eine solche Arbeit zu übernehmen. Er verfügt auch über deutsche Anleitungen zu SCRIPSIT und NEWDOS80, aber nur als (nicht sehr gute) Kopien und würde diese auch ausleihen.

Auch der Dieter Kasper hat mir geschrieben. Er trägt sich zur Zeit mit dem Gedanken, seinen Genie mit einer Diskettenstation aufzurüsten (ev. Station mit 2 Laufwerken, 80 Spuren, ds, dd). Vielleicht kann noch jemand dem Dieter Tips geben! Recht informativ fand der Dieter den Bericht über SUPER DOS, wenngleich ein Nicht-Diskettenbesitzer die Bedeutung einiger Abkürzungen nur ahnen kann. Laut Dieter läuft das Beispiel-Programm für schnelle Graphik auf dem Genie 1 nicht. Vielleicht kann ihm wer weiterhelfen (oder läuft's tatsächlich nicht?).
Auch dem Dieter gefällt die Idee vom Gerald Schröder sehr gut und er wäre bereit, im Rahmen seiner Englisch-Kenntnisse und einer begrenzten Zeit an der übersetzung von englischen Anleitungen mitzuarbeiten (ich glaube, da könnte doch was zusammengehen!).

Auch vom Holger May liegt mir schon ein Schreiben vor. Er hat einen Verbesserungsvorschlag, den ich sehr gerne aufgreife. Wo es nicht absolut notwendig ist, sollte auf Nachnamen und das doch relativ unpersönliche 'Sie' unter den Mitgliedern verzichtet werden. Nun ist das so ein Problem - nicht jeder mag das 'Du', auch gibt es einen größeren Altersunterschied im Club. Andererseits weiß ich aber auch, daß die meisten gegen das 'Du' nichts einzuwenden habe (ich bin z.B. mit vielen per 'Du'). Ich schlage also vor, daß jeder, der gerne mit 'Sie' und 'Herr' angeredet wird, mir das einfach mitteilt.

Der Gerald Schröder schreibt einiges zur Adventure-Ecke. Er bleibt bei vielen Adventures hängen. Er hat viele Möglichkeiten, weiter zu machen, doch findet er wegen 'Phantasiearmut' und mangelnder Englischkenntnisse keine weiteren Ansatzpunkte. Er möchte dann seine Situation in zwei, drei Sätzen schildern. Wer dieses Problem schon gelöst hat, könnte ihm dann weiterhelfen. Es gibt da noch eine zusätzliche Möglicheit, um das Rätseln (was ja den eigentlichen Sinn eines Adventures ausmacht) wieder in den Vordergrund zu rücken: das HINT-SHEET. Dieses HINT-SHEET gibt Hinweise, mit denen man die Probleme lösen kann.
Beispiel: "Vor mir ist ein Hebel, den ich nicht bewegen kann." Gestaffelt könnten die Hinweise so aussehen: "1. Vielleicht ist der Hebel eingerostet? 2. Sie finden kein öl? Erinnern Sie sich an den Roboter! 3. Der Roboter hilft Ihnen nicht? Sie sollten Ihn außer Gefecht setzen!"
Das Erstellen solcher Hinweisblätter kann auch sehr interessant sein und der Versand könnte über den Club oder direkt über den 'Löser' des Adventures laufen. Beim Verschicken eines Adventures könnten diese Blätter auch beigelegt werden. Wer es ohne Hilfe schaffen will, braucht sie ja nicht anzusehen.

Kurz vor 'Redaktionsschluß' erreichte mich noch ein Schreiben vom Hans König, daß ich gerade noch für's Info verarbeiten konnte. Auch er findet die Idee vom Gerald Schröder gut und bietet folgende Anleitungen an:

NEWDOS 80	ca.	200 Seiten	deutsch
GDOS		85 Seiten	deutsch
VISICALC		150 Seiten	deutsch
BUGOUT		40 Seiten	deutsch
NEWBASIC		5 Seiten	deutsch
TASMON	ca.	20 Seiten	englisch
NEWSCRIPT		180 Seiten	englisch
SUPERUTILITY 3.1		45 Seiten	englisch
LASY WRITER		2 Seiten	englisch
LEVEL III		15 Seiten	englisch
ENHBAS		40 Seiten	englisch
SCRIPSIT		65 Seiten	englisch
PENCIL		110 Seiten	englisch

Die Kopien müßten allerdings selbst gemacht werden.
Noch eine kurze Meldung....
Eine RS232 Schnittstelle mit 3 Ein- Ausgängen für Mod.I ist in Arbeit (Schaltungsentwurf). In Kürze mehr.

Programmiertip:

Im Clubinfo 2 hat unser Freund Guenter Wagner einen von mir im Programm "Clubadressen" angewandten Trick beschrieben, wie man Datensaetze aus einer Datei loeschen kann. Leider muss mir in dem Schreiben, indem ich ihm diesen Trick erklaerte, ein Fehler unterlaufen sein. Deshalb hier noch einmal die Funktion der Loeschroutine fuer Datensaetze aus dem Programm CLUBADR/BAS.

1. Suchen des zu loeschenden Datensatzes anhand des Namens
2. Ersetzen des Namens durch einen moeglichst hohen ASCII-Charakter (z.B. CHR$(255))
3. Sortieren der Datensaetze. Dadurch kommt der zu loeschende Datensatz an das Ende der Datensaetze.
4. Der Inhalt des Feldes indem die Anzahl der Datensaetze gespeichert ist, wird um eins vermindert. Dadurch verschwindet der zu loeschende Datensatz aus der Datei.

Am Prinzip hat sich nichts geaendert. Die Aenderung betrifft vielmehr das Zeichen, welches den Namen ersetzt. Fuegt man ein Leerzeichen ein, gelangt der Datensatz beim Sortieren statt ans Ende, an den Anfang der Datei. Nachdem man dann die Anzahl der Datensaetze um eins vermindert hat, fehlt ploetzlich das Mitglied, dessen Name im Alphabet am hoechsten steht. Ausserdem hat man in der Datei einen Datensatz bei dem der Name fehlt. Schuld daran ist die Sortierroutine des NEWDOS-Betriebssystems. Sie sortiert Datensaetze nach dem Wert der ASCII-Zeichen, die sie enthalten. Und da liegt das Leerzeichen (20 H bzw. 32 D) eben weit vor dem ersten Buchstaben des Alphabets (A = 41 H bzw 65 D).

UCSD-PASCAL und CP/M 1.5 auf Video Genie

Fuer den TRS-80 Model 1 gibt es eine UCSD-PASCAL-Version der Firma FMG Corporation, P.O.Box 16020, Fort Worth, Texas 76133. Um diese, mit einem eigenen Diskettenbetriebssystem ausgeruestete PASCAL-Version, auf einem Video Genie I mit Diskcontroler der Firma RB-Elektronik Booten zu koennen, muessen folgende Aenderungen an der Controlerplatine vorgenommen werden:
1. Pin 11 von IC Z20 (74LS376) von der Leiterbahn (Data 5) trennen
2. den nun freien Pin 11 des IC Z20 mit dem Pin 3 des IC Z20 (Data 7) verbinden.

Nach dieser Aenderung, die nach telefonischer Rueckfrage mit der Firma RB-Elektronik an dem Video Genie eines Bekannten vorgenommen wurde, kann neben der oben angesprochenen UCSD-PASCAL-Version auch die ohne sonstige Hardwareaenderungen lauffaehige CP/M-Version 1.5 gestartet werden.

Disketten zum SUPERPREIS

Ich moechte euch noch einmal auf das schon in der letzten Clubinfo erschienene Angebot von Disketten zu einem wirklich erstaunlich guenstigen Preis hinweisen. Die Disketten haben sich sowohl bei mir, als auch bei meinen Freunden und Bekannten, die mit den verschiedensten Computern und Diskettenstationen arbeiten, bewaehrt. Die Disketten sind ungestempelte Markenware und werden im Zehnerpack mit Lables und Schreibschutzaufklebern geliefert. Der Preis belaeuft sich auf 47.00 DM je Paket. Darin sind Porto und Versand schon enthalten. Wie lange die Disketten zu diesem Preis noch lieferbar sind, ist derzeit nicht abzusehen.

Passwort aendern mit SUPERZAP

Wer hat sich nicht schon ueber Disketten geaergert, die durch ein unbekanntes Masterpassword unzugaenglich gemacht worden sind. Mit SUPERZAP kann man dieses Uebel schnell aus der Welt schaffen. Wie man dies tut, moechte ich hier kurz beschreiben.
Das Masterpassword befindet sich bei allen mir bekannten, auf TRS-80 lauffaehigen Disketten im sog. GAT-Sector (Granule Allocation Table) des Direktory. Dort belegt es die beiden relativen Bytes CE u. CF (H) bzw. 204 und 205 (D). Besieht man sich diesen ersten Sector des Direktorys mit SUPERZAP so sind dies die beiden letzten Bytes der 13 Reihe. Natuerlich ist das Password dort nicht im Klartext, sondern als Hash-Code gespeichert. Nun zur eigentlichen Aenderungs-Procedure:

1. Anzeigen des GAT-Sectors des Direktorys mit der Funktion DFS (=Display File Sector) aus dem SUPERZAP Hauptmenu. Hat man DFS eingegeben, fragt das Programm: Filenspec?. Worauf man mit DIR/SYS:L antwortet (L steht fuer die Laufwerksnummer) SUPERZAP liest die File und fragt dann: Relative-Sector-within-File?. Der GAT-Sector ist der erste relative Sector der File DIR/SYS. Man muss diese Frage also mit '0' beantworten. Nun zeigt SUPERZAP den GAT-Sector an.
   (Wenn man nur die etwas aeltere Version von SUPERZAP besitzt, muss man den GAT-Sector ueber einen kleinen Umweg erreichen. Man muss DD (Display Disk Sector) eingeben und auf die Fragen:
   Relative Disk # (0-3)? mit den entsprechenden
   Track # (Hex) (0-22)? Werten antworten. Die
   Sector # (0-9)? Lage des Direktory muss
   man sich aus dem Pdrive-Angaben heraussuchen.)
2. Aendern des Masterpasswords mit der MOD(dify)-Funktion. Dazu ist folgende Eingabe noetig: MODCE. Jetzt blinkt der Cursor auf den Hash-Code-Bytes des Masterpasswords. Danach erfolgt die eigentliche Aenderung. Man schreibt E042 und hat damit das unbekannte Masterpassword in "PASSWORD" geaendert. E042 ist naemlich der Hash-Code fuer "PASSWORD". Um die Aenderung zu beenden, muss 'ENTER' gedrueckt werden. Danach fragt SUPERZAP, ob die Aenderung auf die Diskette geschrieben werden soll. Ist dies geschehen, wird der neue GAT-Sector angezeigt. Man kann nun mit 'X' ins Hauptmenu, und mit 'EXIT' ins DOS zurueck.

Man sollte diese Procedur uebrigens ein paar mal an Backup's ueben, bevor man sich an Originale herantraut. Das Beste waere es, immer mit Backup's zu arbeiten.
Die neuere Version von SUPERZAP enthaelt uebrigens ein Password-Codier-Unterprogramm. Damit kann man jedes Password in den entsprechenden Hash-Code umwandeln und mit der MOD-Routine auf die Diskette schreiben.
Wie man das Update und Access Password einzelner Files aendert, beschreibe ich im naechsten Clubinfo.

ERFAHRUNGSBERICHT: Umbau von TRS-80 Model 1 und Video Genie in TRS-80 Model 3 Gehaeuse

Welcher TRS-80 Model 1 - Besitzer hat sich noch nicht ueber die Dreifaltigkeit seines Computers, CPU-Platine mit Tastatur, Expansionsinterface und Stromversorgung geaergert? Besitzt er zudem noch Floppylaufwerke, wird das Kabelgewirr noch unuebersichtlicher und gefaehrlicher. Gefaehrlich weniger fuer den Computerfreund als fuer die von ihm verarbeiteten Daten und Programme. Schon ein kleiner Wackelkontakt am Edgecardconnector und die Kiste stuerzt ab. Spaetestens nachdem sich ein, in stundenlanger Arbeit eingetipptes Programm in Wohlgefallen aufgeloest hat, kommt man zu der Erkenntnis: hier muss etwas getan werden! Ich persoenlich habe schon mehrere Sachen ausprobiert. Es fing damit an, dass ich alle Steckverbinder durch Loetanschluesse ersetzte. Dies endete damit, dass nach jedem Transport mindestens eines der unzaehligen Kabelchen geloest hatte. Die letzte Massnahme, die ich ergriffen habe, brachte endlich den durchschlagenden Erfolg. Ich erstand das Gebrauchtgehaeuse eines Model 3.
Um das Model 1 im Model 3 - Gehaeuse unterzubringen, sind neben einer moeglichst guten Ausruestung ca. 30-40 Stunden Arbeit noetig. Die genaue Vorgehensweise, moechte ich hier nicht beschreiben. Sollte jemand etwas aehnliches vorhaben, wendet er sich am besten direkt an mich. Ich bin dann gerne bereit, eine genauere, mit Fotos versehene Umbauanleitung zu geben. Hier moechte ich mich auf die Unterbringung der Einzelteile beschraenken. Die Hauptplatine des M 1 liegt, aehnlich wie beim Original, unter der Tastaturplatine. Sie muss jedoch leicht nach hinten verschoben sein, damit die Tastatur eine moeglichst gute Arbeitslage bekommt. Das Expansionsinterface wird im neuen Gehaeuse dort untergebracht, wo normalerweise die CPU-Platine des Model 3 sitzt. Dabei muss auf die Einbautiefe geachtet werden. Setzt man die Platine zu weit nach innen, kollidiert sie spaeter mit der Bildroehre des Monitors. Auch die Drives, bei meiner Version sind es drei an der Zahl, bekommen den Platz, der ihnen auch im Original M 3 zusteht. Nur die Befestigung laeuft erheblich anders. Am leichtesten ist der Monitor unterzubringen, da die meisten 12"-Geraete genormte Befestigungspunkte haben. Die Powersupplys bringt man am besten in einem separaten Gehaeuse unter und verbindet Netzteil und Computer mit einem Kabel.
Dieser Umbau hat sich bei meinem Geraet vorteilhaft auf die Betriebssicherheit ausgewirkt. Inzwischen hat auch ein Freund von mir sein Video Genie in ein M 3 - Gehaeuse umgebaut. Das Video Genie macht dabei erheblich weniger Arbeit als das M 1. Wie schon gesagt, bin ich gerne bereit, eine genaue Umbauanleitung zu liefern. Auch gebrauchte Gehaeuse kann ich eventuell noch besorgen. Sollte jemand Interesse haben, wendet er sich am besten direkt an mich. Meine Adresse koennt ihr dem Club-Info Nr.2 entnehmen.

** Hartmut Obermann **

COMPUTER-FACHBUCHHANDLUNG

L. Polacek, Postfach 43, 8409 Tegernheim

Wir können Ihnen fast 300 deutsche und noch mehr englischsprachige Bücher zum Thema Mikrocomputer besorgen. Für alle ist etwas dabei:

- für Anfänger, Fortgeschrittene und Profis
- für Bastler, Hardwarespezialisten und Programmierer
- für Hobbyisten und gewerbliche Anwender

Fordern Sie unseren nach Sachgebieten geordneten Katalog an - kostenlos! Aus Nachträgen zum Katalog (zwei- bis dreimal im Jahr) erfahren Sie, was es auf dem Büchermarkt Neues gibt.

Wir vermitteln Ihnen ein Zeitschriften-Abonnement: es gibt schon mehr als zehn deutsche und mehr als zwanzig englische/amerikanische Magazine zum Lesen.

Schaffen Sie sich einen Vorsprung - ohne Fachliteratur ist es aber nicht möglich.

Auszug aus unserem Katalog:

TRS-80 Assembler Programmierung	52,-
TRS 80 Basic Trick Buch	52,-
TRS-80 Disk & andere Geheimnisse	39,-
Graphik-Programme für TRS-80 und HP-9830	24,80
Spiele für Ihren TRS 80	13,80
I Speak Basic to My TRS-80	99,-
Writing BASIC Adventure Programs for the TRS-80	39,-

Anleitung zur Erstellung eigener Abenteuerspiele: Lernen Sie dabei neue Programmiertechniken, bessere Speicherausnutzung, höhere Programmiergeschwindigkeit,

How to Do It on the TRS-80	129,-

Ein Anwenderhandbuch nach Sachgruppen geordnet für TRS-80 Modelle und Color Computer. Hard- u. Softwarelösungen für den Alltag des Programmierers

Color Computer Graphics	32,-

Vollständiger Führer, bietet Instruktionen über die Anwendung von Video-Grafiken u. viele komplette Programme

The Rest of 80	38,-

Programme u. Hinweise, für die kein Platz in der Zeitschrift 80 Micro übrig blieb.

```
10 '*****************************************************************
20 '                         Z W I H E X
30 '*****************************************************************
40 '
50 '                          Copyright
60 '                             by
70 '
80 '                         W.ZWICKEL
90 '
100 '             Rel 1.1             07-03-83
110 '*****************************************************************
120 PRINT"Programm zur Umwandlung von Dezimal-in Hexzahlen und umgekeh
130 PRINT"Dez in Hex ---> x drcken "
140 PRINT"Dez in Hex ---> d drcken "
150 INPUT X$
160 IF X$ = "x" GOTO 500
170 TD = 0
180 HD = 0
190 TE = 0
200 CLS
210 PRINT
220 PRINT"Dezimalzahl zwischen 0 und 65535 eingeben "
230 INPUT A
240 PRINT A
250 T=A
260 IF T<4096 THEN GOTO 300
270 T=T-4096
280 TD=TD+1
290 IF T>=4096 THEN GOTO 270
300 IF T<256 THEN GOTO 340
310 T=T-256
320 HD=HD+1
330 IF T>=256 THEN GOTO 310
340 IF T<16 THEN GOTO 380
350 T=T-16
360 TE =TE +1
370 IF T>=16 THEN GOTO 350
380 UD =T
390 PRINT
400 A$="0123456789abcdef"
410 T$=MID$(A$,TD+1,1)
420 H$=MID$(A$,HD+1,1)
430 Z$=MID$(A$,TE+1,1)
440 U$=MID$(A$,UD+1,1)
450 PRINT" dez ";A;" = Hex ";T$;H$;Z$;U$
460 PRINT
470 PRINT"Weiter ???  D=Dez  X=Hex "
480 INPUT X$
490 IF X$="d" THEN GOTO 170
500 PRINT
510 PRINT "hexzahl vierstellig zwischen 0000 und ffff eingeben"
520 INPUT A$
530 PRINT
540 PRINT "hex ";A$;" = dez ";
550 IF LEN(A$) <>4 THEN 510
560 A = 1
570 GOSUB 690
580 T = Z
590 A = 2
600 GOSUB 690
610 H = Z
620 A = 3
630 GOSUB 690
640 ZE = Z
650 A = 4
660 GOSUB 690
670 E = Z
680 GOTO 810
690 Z = 0
700 IF MID$ (A$,A,1) ="a" THEN Z=10
710 IF MID$ (A$,A,1) ="b" THEN Z=11
720 IF MID$ (A$,A,1) ="c" THEN Z=12
730 IF MID$ (A$,A,1) ="d" THEN Z=13
740 IF MID$ (A$,A,1) ="e" THEN Z=14
750 IF MID$ (A$,A,1) ="f" THEN Z=15
760 IF Z<>0 THEN 800
770 FOR J = 0 TO 9
780 IF VAL (MID$ (A$,A,1)) =J THEN Z=J
790 NEXT J
800 RETURN
810 PRINT 4096*T+256*H+16*ZE+E
820 PRINT
830 PRINT "weiter ????  d=dez   x=hex"
840 INPUT X$
850 IF X$ = "d" THEN 170
860 IF X$ = "x" THEN 500
870 REM zwihex
```


Trace-ON

TRS-80 M I

Wird ein Basic-Programm mit Hilfe des Trace-Befehls auf dem TRS-80 beim Probelauf kontrolliert, so passiert es sehr häufig, daß eine erstellte Bildschirmmaske durch die Ausgabe der laufenden Zeilennummer zerstört wird. Auch ist es häufig nicht möglich, die einzelnen Zeilennummern zu verfolgen, da sie zu schnell ausgegeben werden.

Das Maschinenprogramm (siehe Listing) für TRS-80 (Model 1/Level II) wurde mit dem Editor-Assembler erstellt und ist in drei Teile aufgeteilt: Im Teil eins werden die entsprechenden DOS-Adressen initialisiert. Der zweite Teil umfaßt die eigentliche Trace-Routine. In den Zeilen 330 bis 490 wird die Absprungadresse überprüft. Erfolgte der Absprung nicht von der Trace-Routine (im ROM), so folgt ein einfaches Return, andernfalls wird die Zeilennummer ausgegeben. In den Zeilen 500 bis 530 wird die Cursoradresse gesichert und durch 3C00H ersetzt. An dieser Bildschirmstelle soll die Zeilennummer angezeigt werden. Nach der Ausgabe in den Zeilen 550 bis 650 wird die Cursoradresse wieder korrigiert und nach 1D58H zurückgesprungen. Der letzte Teil des Programms erlaubt es, die Ablaufgeschwindigkeit eines Programms zu verändern. Wird während des Programmablaufs oder Auflistens die »Shift«- und die »Pfeil/unten«-Taste gedrückt, so verlangsamt sich der Ablauf. In den Zeilen 750 bis 810 werden diese Tasten abgefragt. Wurden sie gedrückt, dann wird die Zeitschleifenroutine im ROM aufgerufen (Zeilen 830 bis 860). Durch Ändern des Zeitfaktors in Zeile 840 kann die Geschwindigkeit beeinflußt werden. Wurden diese zwei Tasten nicht gedrückt, erfolgt eine normale Tastaturabfrage mit anschließendem Return. Dieses Programm braucht nicht gesichert zu werden, da es sich durch Verschieben des Basic-Bereichs selbst schützt.

(Martin Höh)

```
                00100 ; ----------------------------------------
                00110 ;          --- TRACE - ON ---
                00120 ;   fuer TRS-80 (Mod. 1 / Level II)
                00130 ;      (C) Martin Hoeh / 1983
                00140 ; ----------------------------------------
                00150 ;
                00160 ; Initialisierung
                00170 ;
41C1            00180          ORG     41C1H
41C1 C3EF42     00190          JP      TRACE
4016            00200          ORG     4016H
4016 2F43       00210          DEFW    TASTE
40A4            00220          ORG     40A4H
40A4 4B43       00230          DEFW    ENDE+2
40F9            00240          ORG     40F9H
40F9 4D43       00250          DEFW    ENDE+4
                00260 ;
42E9            00270          ORG     42E9H
42E9 CD491B     00280 ENTRY    CALL    1B49H
42EC C3CC06     00290          JP      06CCH
                00300 ;
                00310 ; Trace - Routine
                00320 ;
42EF D9         00330 TRACE    EXX
42F0 D1         00340          POP     DE
42F1 C1         00350          POP     BC
42F2 E1         00360          POP     HL
42F3 7C         00370          LD      A,H
42F4 FE1D       00380          CP      01DH
42F6 CAFE42     00390          JP      Z,WEITER
42F9 E5         00400 AUSGA    PUSH    HL
42FA C5         00410          PUSH    BC
42FB D5         00420          PUSH    DE
42FC D9         00430          EXX
42FD C9         00440          RET
                00450 ;
42FE 7D         00460 WEITER   LD      A,L
42FF FE50       00470          CP      050H
4301 C2F942     00480          JP      NZ,AUSGA
                00490 ;
4304 2A2040     00500          LD      HL,(4020H)
4307 E5         00510          PUSH    HL
4308 21003C     00520          LD      HL,3C00H
430B 222040     00530          LD      (4020H),HL
                00540 ;
430E D9         00550          EXX
430F 3E8F       00560          LD      A,08FH
4311 CD2A03     00570          CALL    032AH
4314 3E20       00580          LD      A,020H
4316 CD2A03     00590          CALL    032AH
4319 CDAF0F     00600          CALL    0FAFH
431C 3E20       00610          LD      A,020H
431E CD2A03     00620          CALL    032AH
4321 3E8F       00630          LD      A,08FH
4323 CD2A03     00640          CALL    032AH
4326 D9         00650          EXX
                00660 ;
4327 E1         00670          POP     HL
4328 222040     00680          LD      (4020H),HL
                00690 ;
432B D9         00700          EXX
432C C3581D     00710          JP      1D58H
                00720 ;
                00730 ; Tastatur - Routine
                00740 ;
432F 3A4038     00750 TASTE    LD      A,(3840H)
4332 CB67       00760          BIT     4,A
4334 280F       00770          JR      Z,ABFRA
                00780 ;
4336 3A8038     00790          LD      A,(3880H)
4339 CB47       00800          BIT     0,A
433B 2808       00810          JR      Z,ABFRA
                00820 ;
433D C5         00830          PUSH    BC
433E 010050     00840          LD      BC,5000H
4341 CD6000     00850          CALL    0060H
4344 C1         00860          POP     BC
                00870 ;
4345 CDE303     00880 ABFRA    CALL    03E3H
4348 C9         00890          RET
4349 00         00900 ENDE     NOP
434A 00         00910          NOP
434B 00         00920          NOP
42E9            00930          END     ENTRY
00000 TOTAL ERRORS

ABFRA   4345
AUSGA   42F9
WEITER  42FE
ENTRY   42E9
ENDE    4349
TASTE   432F
TRACE   42EF
```

Listing des Programms »Trace-ON«

Software

Herr oder Frau?

Das Problem zu einem gegebenen Vornamen die richtige Anrede (»Herr« oder »Frau«) zu finden, wird hier durch ein Basic-Programm gelöst.

Das Abspeichern aller, oder zumindest sehr vieler, Vornamen scheidet aufgrund von geforderten Randbedingungen, wie Schnelligkeit, hohe Trefferquote, nur wenige KByte, aus. Das hier angegebene Verfahren erfüllt alle Ansprüche bestens und zeigt exemplarisch, wie Speicherplatz eingespart werden kann (siehe Listing).

Vom eingegebenen Vornamen werden die beiden letzten Buchstaben abgetrennt und mit einer Tabelle verglichen. Wird die Endung in dieser Tabelle gefunden, ist schon eine Vorentscheidung gefallen. Bestimmendes Kennzeichen eines Vornamens ist nämlich die Endung. Um Fehler, wie beispielsweise bei Cilli und Willi, zu vermeiden, wird danach der gesamte Vorname nochmals mit einer weiteren Tabelle verglichen, in der sich die (seltenen) Ausnahmen befinden. Diese zweite Datei kann beliebig erweitert werden.

Das angegebene Programm läuft auf allen Commodore-Computern ohne Änderung und ist problemlos für andere Systeme umzuschreiben.

(Berthold Weber)

```
100 rem programm zur erkennung von vornamen
110 rem (c) by berthold weber landgerichtsstr. 11 8480 weiden
120 :
130 rem****** programm *********
140 :
150 inputn$
160 g=0:restore
170 t$=right$(n$,2)
180 readtt$,gg:iftt$="ende"then210
190 ift$=tt$theng=gg:goto210
200 goto180
210 readtt$,gg:iftt$="ende"then240
220 iftt$=n$theng=gg:goto240
230 goto210
240 ifg=1thenprint"Herr ";n$
250 ifg=2thenprint"Frau ";n$
260 ifg=0thenprint"keine Ahnung"
270 goto150
280 :
290 rem******* daten ***********
300 :
310 dataer,1,ld,1,an,1,rd,1,nz,1,ut,1,ar,1,on,1,am,1,do,1,rt,1,es,1
320 datada,2,si,2,ia,2,ni,2,te,2,ig,1,de,2,ed,1,ng,1,ed,1,ta,2,rg,1
330 datato,1,ax,1,ex,1,ch,1,re,2,ns,1,it,1,st,1,ka,2,ar,1,ck,1,us,1
340 dataus,1,or,1,rt,1,ef,1,el,1,en,1,el,1,ed,1,nk,1,im,1,ph,1,as,1
350 datara,2,ne,2,ta,2,ee,2,ke,2,ea,2,ic,1,ie,2,na,2,do,1
360 datago,1,li,2,rk,1,nn,1,tz,1,nd,1,ev,1,ot,1,lf,1,ad,1,no,1
370 datale,2,la,2,is,2,ta,2,vi,2,in,2,th,2,ga,2,ty,2,se,2,ud,2,tl,2
380 databy,2,bi,2,un,2,di,2
390 datagi,1,jo,1,il,1,ny,1,ki,1,id,2,ve,2,hi,2,yn,2
400 datarl,1,we,1,eo,1,do,1,va,2
410 data"ende",0
420 data"Charlie",1,"Dagmar",2,"Willi",1,"Edeltraud",1,"Waltraud",2
430 data"Bene",1,"Boris",1,"Erwin",1,"Mali",2,"Rut",2,"Heini",2,"Agnes",2
440 data"Birgit",2,"Marion",2,"Margit",2
450 data"Ellen",2,"Waltraut",2;"Helmuth",1
460 data"ende",0
ready.
```

Listing des Programms, das zur Erkennung des Geschlechtes von Vornamen dient und aufgrund dessen die entsprechende Anrede herausfindet

Der aktuelle Bericht (von Günther Wagner) :

ist eigentlich recht kurz und unbedeutend - er befaßt sich mit der Computerszene in Norwegen (dort verbrachte ich meinen Sommerurlaub und vergaß natürlich nicht den CLUB 80).

Zunächst muß gesagt werden, daß Norwegen sehr dünn besiedelt ist (nach Island das schwächst besiedelte Land Europas). Ich fuhr mit der Vorstellung nach Norwegen, daß der Computer in Norwegen kaum zu finden ist. Doch im Gegenteil - bereits am Fährhafen lief alles über Computer und an der 1. Tankstelle erst recht! In Jeder Stadt fand ich einige Computergeschäfte (keine Computerladen u.ä.!). Meistens waren es bei den Heim-Computern Commodore und Atari, manchmal Casio, Sharp und Dragon. Der VC-64 kostet umgerechnet zwischen 800 und 900 DM. Ungefähr das gleiche kostet das Laufwerk. Viel IBM war zu sehen - Jedoch kein einziges Mal Apple und Tandy! Einige weitere kommerzielle Computer fielen mir noch auf, Jedoch sagte mir der Name nichts. Auffallend noch: In den Computergeschäften sieht man keine Jugenlichen an den Computern stehen. Der Grund dafür: Betritt man das Geschäft, wird man vom Personal sofort in Beschlag genommen und nach seinen Wünschen gefragt. In den Computergeschäften ist reichhaltiges und gutsortiertes Zubehör zu finden - von Programmen über HardwareErweiterungen bis hin zu Computerzeitschriften ist alles da. Und das in einem so dünn besiedelten Land!

F R A G E ?

Ich würde gerne wissen, wer noch mit hochauflösender Grafik (in X 0-383, in Y 0-191 Punkte) schafft. Meine Grafikzusatzplatine von RB-Elektronik arbeitet sehr zufriedenstellend. Über einen Informationsaustausch würde ich mich freuen.

(Jens Neueder)

DIE LETZTE SEITE

Wenn ich etwas vergessen habe (das hab ich gewiß!), dann erinnert mich bitte - ich hole es dann nach. Ansonsten hoffe ich nur noch, daß ich bis zum nächsten Info wieder viel Post bekomme.

Günther Wagner

* C L U B 8 0 *

Die letzte Seite

CLUB 80 - Clubinfo Nr. 4

November 1984

Hallo Freunde,

ich begrüße Euch alle recht herzlich und hoffe, daß Euch dieses 4. Clubinfo wieder recht gut gefällt. Es ist das bisher umfangreichste, aber ich habe mir gesagt, warum etwas zurückhalten wenn es da ist. Das nächste Info kann also wieder dünner werden. Bedanken möchte ich mich bei Allen, die mir bei der Erstellung dieses Infos geholfen haben; entschuldigen möchte ich mich bei denen, die etwas länger auf Ihre Sendungen warten mußten.

Was gibt es nun zu lesen:
Da wäre zunächst einmal ein vorgezogener Kassenbericht mit anschließendem Antrag auf Änderung unserer Clubsatzung (Termin beachten!). Auf das Thema Clubtreffen wird näher eingegangen. Ein BASIC-'Kurs' wird begonnen. Eine komplette Bücherliste wird abgedruckt, ebenso neue Programme und der Inhalt unserer neuen Fundgrube. Einige Berichte, die POKE + PEEK-Ecke sowie die Adventure-Ecke sind auch vorhanden. Die Datenfernübertragung und noch vieles vieles mehr wird angesprochen.

Ein wenig stolz bin ich auf die Entwicklung unseres Clubs. Aufgrund von Berichten in der GENIE DATA und der CP sowie großer Werbung vom Hartmut Obermann kann ich auf die stolze Anzahl von

— 25 —

Mitgliedern blicken. Dabei ist der CLUB 80 erst knapp über 6 Monate alt! Ich hoffe, daß unser Club weiter so im Aufwind bleibt.

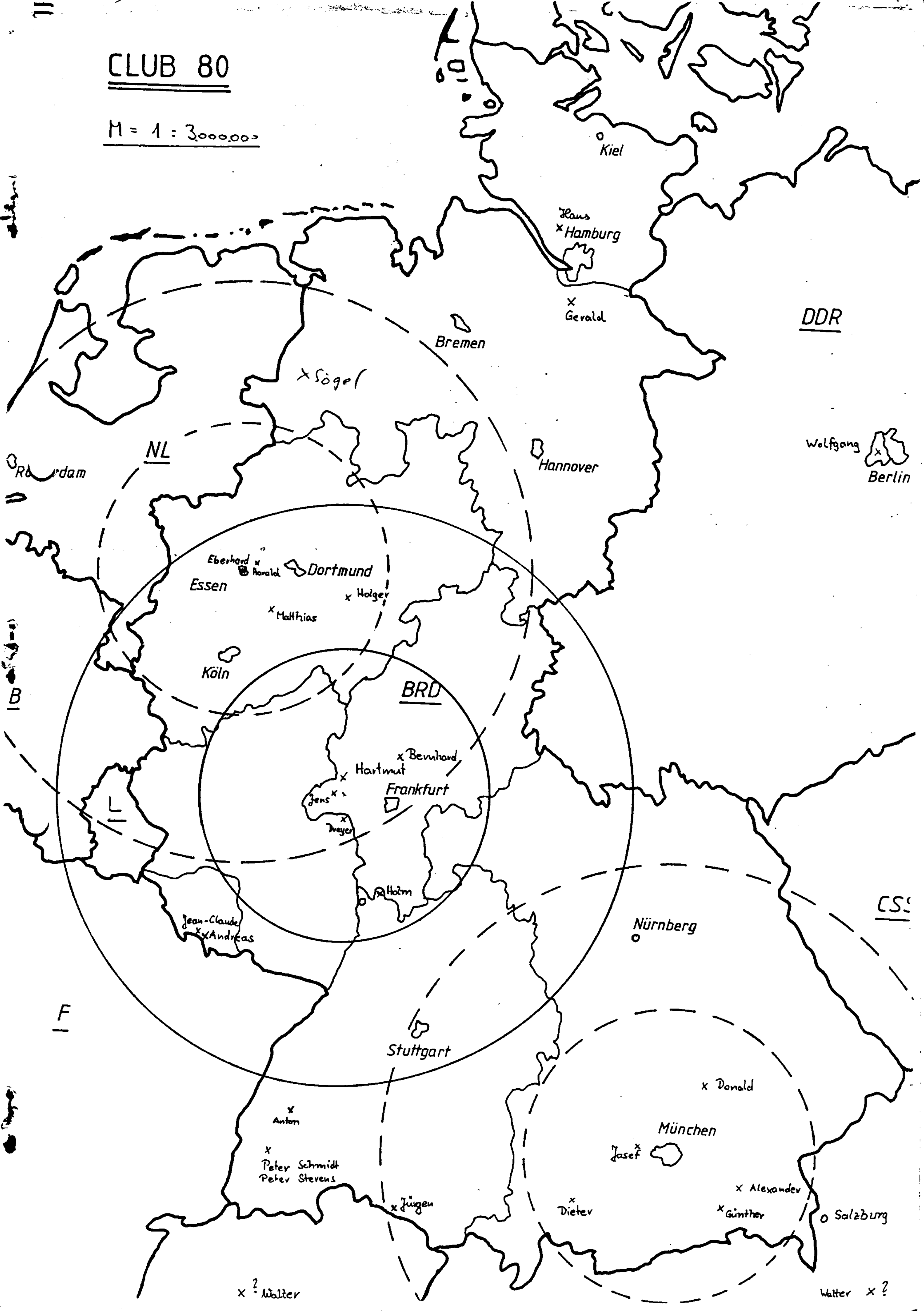
Ich möchte jetzt nur noch anfügen, daß dieses Clubinfo weit über 40 verkleinerte Seiten umfaßt (also ca. 80 DIN A4 Seiten!). Meine Bitte auf der 1. Seite: Schickt Eueren Stimmzettel und den Fragebogen bitte rechtzeitig alle zurück!

Damit Schluß für die erste Seite

Euer Günther

CLUB 80

M = 1 : 3000000



CLUBTREFFEN

Schon seit längerem laufen Überlegungen über ein Clubtreffen (ein 'kleines' gab es auf der Hobbytronic in Stuttgart - siehe Bericht). Ich habe jetzt mal auf einer Karte in etwa die Wohnorte aller Clubmitglieder eingetragen und mir anschließend Gedanken gemacht, wo ein Clubtreffen am besten zu veranstalten ist. In die engere Auswahl sind gekommen:

Entfernung	bis 100 km	100 - 200	über 200
Ruhrgebiet	4	4	17
Mainz/Frankfurt	5	6	14
München	5	2	18

Anhand dieser kleinen Tabelle und der Karte werden mir alle Mitglieder zustimmen, daß als Treffpunkt für ein Clubtreffen der Großraum Mainz/Frankfurt am besten geeignet ist. Er liegt sehr zentral und wer gerne kommen will, kann das auch.

Mein Vorschlag:
Im Gebiet Mainz/Frankfurt wird eine Pension, Gasthof oder Hotel gesucht, welches für 15-25 Personen preisgünstig Unterkunft bietet. Es muß ein Konferenzraum (Nebenzimmer) vorhanden sein, damit wir in aller Ruhe unser Clubtreffen abhalten können. Dazu gehört ein offizieller Teil in Form einer Mitgliederversammlung, und der sog. inoffizielle Teil, bei dem man sich kennenlernt, Erfahrungen austauscht, Probleme bespricht, Programme untereinander tauschen kann, etc. Es ist also auch notwendig, daß einige Computer aufgestellt werden können. Eventuell können auch einige Produkte vorgestellt werden (dazu muß man entsprechende Firmen darauf ansprechen).

Es ist also unbedingt ein Wochenende notwendig. Der ungefähre Ablauf könnte so aussehen:

> Anreise am Samstag, wobei Jeder spätestens noch vor dem Abend eintreffen sollte. Hierbei können natürlich auch Mitglieder bereits am Vormittag eintreffen. Am Abend findet der offizielle Teil, die Mitgliederversammlung statt. Der inoffizielle Teil erstreckt sich auf die andere Zeit, wobei es Jedem einzelnen überlassen bleibt, wann er die Rückfahrt antritt. Die einen werden bereits am Vormittag abfahren, einige aber auch erst am Abend.

Als Zeitpunkt für unser Clubtreffen schlage ich ein Wochenende der Fastenzeit vor. Die meisten Mitglieder dürften in diesem Zeitraum am ehesten Zeit haben. Als mögliche Wochenende für unser Clubtreffen schlage ich demnach vor:

- 09./10. März 1985
- 16./17. März 1985
- 23./24. März 1985
- 30./31. März 1985

<u>An Kosten entstehen:</u> 1 Übernachtung m. Frühst. (o. Halbpension)
Fahrtkosten (diese können durch Bildung v. Fahrgemeinschaften reduziert werden)
Eigenbedarf (Getränke, weitere Mahlzeiten)

Meine Bitte an alle Mitglieder:

Jeder teilt mir seine grundsätzliche Einstellung zu einem Clubtreffen (wie es geplant ist) mit! ob er an einem Clubtreffen teilnimmt oder nicht.

Ich muß von Jedem wissen, an welchem Wochenende er nicht kann. Jeder soll mir mitteilen, ob Halbpension oder Übernachtung mit Frühstück gewählt werden soll.

Wer kein Auto hat, soll mir das mitteilen. Ebenso muß ich unbedingt wissen, wer mit dem Auto kommen würde und bereit wäre, andere Clubmitglieder mitzunehmen. So können wir Fahrgemeinschaften besser koordinieren.

Ich muß beim nächsten Clubinfo sagen können, ob wir ein Clubtreffen veranstalten oder nicht. Wenn wir eins veranstalten, muß ich auch den Termin festlegen und wissen, wer in etwa kommt.

Bitte laßt mich Jetzt nicht im Stich und schreibt mir zu diesem Thema bis spätestens Weihnachten! Danke

Auch andere Vorschläge bzw. Meinungen dazu usw. könnt Ihr mir zusenden.

Verleihliste Hartmut Obermann (FS nächste Seite)

TRS-80 Graphics
....
Grafik, programmiert auf dem TRS-80 mit Level II - BASIC.
Dieses Buch enthaelt von den Grundlagen ueber bewegte Bilder bis zur Darstellung mathematischer Funktionen alles, war man ueber die Grafik des TRS-80 wissen muss.
Verfasser: Don Inman Verlag: Radio Shack (Tandy Corporation)
1300 One Tandy Center
Fort Worth, TX. 76102

Disk Interfacing Guide
....
Aus dem Inhalt: Disk BASIC, Shugart SA400-Laufwerk, Floppycontroler FD1771B-01, Expansion Interface, Disk Programming (Basic und Assembler).
Dieses Buch bildet einen tiefen Einblick in die Technik der Ansteuerung der Diskettenlaufwerke des TRS-80.
Verfasser: William Barden Verlag: 80-Northwest Publishing Inc.
3838 South Warner Street
Tacoma, Washington 98409

The first Book of 80-US
....
Sammlung der besten Beitraege aus dem Journal fuer TRS-80 Anwender.
Verlag: 80-Northwest Publishing Inc.
3838 South Warner Street
Tacoma, Washington 98409

DIE NEUEN

Schmidt Peter
Drei-Aehrenstr. 18
7800 Freiburg
-

Clubmitglied seit : 19.09.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
TRS-80 Mod. I/ Mod. III/ Mod. 4P/ Epson-MX80/ mehrere Laufwerke (je 40 Spuren)
bevorzugtes Betriebssystem : NEWDOS

Sickmann Bernhard
Fleigenweg 2
4430 Steinfurt 2
02552/60344

Clubmitglied seit : 24.10.84
Beitrag bezahlt : ja
Punktestand : 35
System- und Drivekonfiguration :
TRS-80 Model IV/ Drucker TRS 410/ 2 Laufwerke (je 40 Spuren/ ss / dd)
bevorzugtes Betriebssystem : TRSDOS und NEWDOS

Trapp Harald
Kranichstr. 46
4270 Dorsten 1
-

Clubmitglied seit : 04.10.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
Genie II/ Expansion/ Drucker microline 80/ 2 Laufwerke (je 40 Spuren/ ss/ sd)
bevorzugtes Betriebssystem : NEWDOS 80 Version 2.05

Marx Andreas
Mecklenburgring 48
6600 Saarbr cken
0681/812983

Clubmitglied seit : 02.11.84
Beitrag bezahlt : ja
Punktestand : 35
System- und Drivekonfiguration :
Video Genie EG 3003 64K/ Fernschreiber + Interface als Drucker/ 2 Laufwerke (je 40 Spuren/ ss/ sd und dd)
bevorzugtes Betriebssystem : NEWDOS

Dreyer Gerald
Am Speiergarten 8
6200 Wiesbaden-Bierstadt
06121/508218

Clubmitglied seit : 26.09.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
Genie I/ 48K/ HRES-Graphik/ BMC-Monitor/ Epson MX80 mit Graftrax / 2 Laufwerke (je 40 Spuren/ ds/ dd)
bevorzugtes Betriebssystem : Superdos 6.0 (Newdos 80 Version 2)

Hallup Matthias
Junggesellenstr. 15
4600 Dortmund
-

Clubmitglied seit : 17.09.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
Video-Genie Mod. I/ 64 K-RAM/ Monitor/ Epson RX-80/ 2 Laufwerke (je 40 Spuren/ ss/ sd+dd)
bevorzugtes Betriebssystem : NEWDOS 80 Version 2.0

Hummel Anton
Schubertstr. 2
7612 Haslach
07832/8289

Clubmitglied seit : 09.10.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
Genie II (64K)/ Fernschreiber (als Drucker)/ ZX81 (64K) 2 Laufwerke
bevorzugtes Betriebssystem : NEWDOS 80 und SUPERDOS

Piller Walter
Rohnenstr. 8
CH-8835 Feusisberg
01/7847418

Clubmitglied seit : 25.09.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
TRS-80 Level II mit Expansion Interface/ Kasettenger t (vorl ufig)/ Drucker -
bevorzugtes Betriebssystem : -

Buecher-Verleihliste: vom Hartmut Obermann

Programmieren ganz einfach
....
Einstieg in die Welt der Computer mit BASIC. Geeignet fuer Kinder ab 8-10 Jahren und Erwachsene ohne jegliche Vorkenntnisse.
Verfasser: B.R. Smith (deutsch: B. Schumacher) Verlag: Vogel-Verlag
Wuerzburg

Programmieren mit TRS-80
....
Eine Einfuehrung in die Programmierung des TRS-80 mit vielen Tip, Tricks, guten technischen Informationen und ca.30 Programmlistings.
Verfasser: M. Stuebs Verlag: Hofacker Verlag Muenchen

Anwenderprogramme fuer TRS-80 und VideoGenie
....
ca.30 Programmlistings vom Autorennen bis zur Lagerverwaltung.
Verfasser: M. Stuebs Verlag: Hofacker Verlag Muenchen

PASCAL in 100 Beispielen
....
100 Programmbeispiele aus allen Bereichen, programmiert in PASCAL. Da die Programmlistings in UCSD-PASCAL geschrieben sind, hat man keine Umschreibeprobleme auf den TRS-80.
Verfasser: Ottmann/Schrapp/Widmayer Verlag: B.G. Teubner
Stuttgart

Dem Microcomputer auf's Bit geschaut (ELO-Sonderheft)
....
Vom TTL-Gatter bis zum funktionierenden Microcomputer. Hard- und Software-Erklaerungen in auch fuer den Laien verstaendlichen Worten.
Verlag: Franzis' Verlag Muenchen

Hobby Computer (ELO-Sonderheft)
....
Grundlagen fuer den Anfaenger und heisse Tips fuer den Profi.
Verlag: Franzis' Verlag Muenchen

Hobby Computer 2 (Funkschau-Sonderheft)
....
Programme und Hardware-Tips (auch fuer TRS-80)
Verlag: Franzis' Verlag Muenchen

Programs in BASIC for Electronic Engineers, Technicians & Experimenters
....
25 Programmlistings aus allen Bereichen der Elektronik.
Verfasser: Ken Tracton Verlag: TAB Books Inc.
Blue Ridge Summit, PA.17214

Digital Interfacing with an Analog World
....
Grundlagen und Schaltungen zur Verbindung von Computern mit der Aussenwelt. Vom Messwertumwandler ueber A/D - D/A - Wandler bis zum Telefon-Waehlschaltkreis ist alles vorhanden.
Verfasser: J. J. Carr Verlag: TAB Books Inc.
Blue Ridge Summit, PA.17214

How to build your own working Microcomputer
....
Ein Buch das vor allem einen Einblick in den internen Aufbau einesMicrocomputers vermittelt. Der Inhalt erstreckt sich vom der Beschreibung der Architektur der 8080-CPU ueber die einzelnen- Baugruppen eines Microcomputers bis zu Assembler-Listings.
Verfasser: Charles K. Adams Verlag: TAB Books Inc.
Blue Ridge Summit, PA.17214

Verbal Control with Microcomputers
....
Dieses Buch beschaeftigt sich mit der Sprachein und -Ausgabe mit Microcomputern. Dabei wird besonders auf den TRS-80 eingegangen. Das Buch vermittelt einen Ueberblick von den Grundlagen der Spracherkennung bis zu funktionierenden Schaltungen (mit Platinen- Layouts).
Verfasser: Mike Riggsby Verlag: TAB Books Inc.
Blue Ridge Summit, PA.17214

Stimulating Simulations
....
12 BASIC-Spiel-Programm-Listings von Space Flight bis zu Monster Case. An diesem Buch fallen besonders die sehr guten Programm- erklaerungen auf. Zu jedem Programm gibt es eine Variablenliste, ein Flussdiagramm und ein gut gegliedertes Listing.
Verfasser: C.W. Engel Verlag: Hayden Book Company Inc.
Rochelle Park, New Jersey

Sixty Cahllenging Problems with BASIC Solutions
....
Sechzig kleine aber interessante Probleme aus der Mathematik. Problem und Loesung stehen auf unterschiedlichen Seiten. So kann man sich selbst pruefen.
Verfasser: Donald D. Spencer Verlag: Hayden Book Company Inc.
Rochelle Park, New Jersey

TRS-80 Level II BASIC (a self-teaching guide)
....
Programmiertes lernen von BASIC mit vielen Beispielen und in ueberschaubaren, kleinen Schritten. Jeder Lektion folgt eine Lernkontrolle.
Verfasser: B. Albrecht, D. Inman & R. Zamora
Verlag: Radio Shack (Tandy Corporation)
1300 One Tandy Center
Fort Worth, TX.76102

TRSDOS & DISK BASIC Reference Manual
....
Auf dieses Manual beruft sich das NEWDOS-Manual oeffters. Es enthaelt die komplette Dokumentation ueber das DiskOperationSystem von Radio Shack (Tandy Corp.).
Verlag: Radio Shack (Tandy Corporation)
1300 One Tandy Center
Fort Worth, TX.76102

Using the TRS-80 in your Home
....
ca. 90 BASIC-Programme aus allen Bereichen des Haushalts und des Hauses. Sehr gute Programmdokumentation mit Variablenliste, Probelauf und gut strukturierten Listings.
Verfasser: Charles D. Sternberg Verlag: Radio Shack (Tandy Corp.)
1300 One Tandy Center
Fort Worth, TX.76102

TRS-80 Programs
....
32 BASIC Programme fuer den TRS-80 Model 1 Level II und VideoGenie. Aus dem Inhalt: Biorhythm, Mileage, Arithmetic, Wari (Spiel), Flying Walloons (Grafik), Graph, Stats, Pythag usw.. Auch dieses Buch faellt durch die sehr gute Documentation der Programme auf.
Verfasser: T. Rugg & P. Feldman Verlag: Radio Shack (Tandy Corp.)
1300 One Tandy Center

Wucherer Juergen
Braunegger str. 14
7750 Konstanz
07531/29145

Clubmitglied seit : 26.10.84
Beitrag bezahlt : ja
Punktestand : 35
System- und Drivekonfiguration :
Video Genie II/ Drucker Star DP-510/ Zenith Monitor/ 2 Siemens- und 2 Shugart-Laufwerke
bevorzugtes Betriebssystem : G-DOS

Eine Adressenänderung

Hallup Matthias
Junggesellenstr. 15
4600 Dortmund
-

Clubmitglied seit : 17.09.84
Beitrag bezahlt : ja
Punktestand : 30
System- und Drivekonfiguration :
Video-Genie Mod. I/ 64 K-RAM/ Monitor/ Epson RX-80/ 2 Laufwerke (je 40 Spuren/ ss/ sd+dd)
bevorzugtes Betriebssystem : NEWDOS 80 Version 2.0

da hat der Jens also ein } bekommen !

Neueder Jens
Panoramastr. 21
7178 Michelbach/Bilz
0791/42877

ES SAGTE . . .

. . . . ein überforderter Computer zu seinem Programmierer: »NIE SOLLST DU MICH BEFRAGEN«

. . . . ein eingebildeter Computer zu Sokrates: »ICH WEISS, DASS DU NICHTS WEISST!«

Auch in diesem Info wollen sich zwei Mitglieder vorstellen:

Nun, ich heiße Jean-Claude Wies, erblickte am 14.12.1967 in Saarbrücken zum ersten Mal das Tageslicht und begann ca. 15 Jahre später, mich mit Rechnern zu beschäftigen.
Zunächst entschloß ich mich, im Inbegriff meiner schulischen Ausbildung das Wahlpflichtfach Informatik zu belegen. Es dauerte demnach auch nicht lange, bis ich mir meinen ersten Rechner zusammensparte - ein Colour Genie. Diesen verkaufte ich nach einem Jahr wieder und legte mir ein Video Genie II System an.
Das wär's zu meiner Vergangenheit. Jetzt möchte ich nur noch hinzufügen, daß ich mit meinem Genie II bisher wenig Schwierigkeiten hatte. Ich arbeite unter dem Betriebssystem NEWDOS 80, aber auch unter GDOS und besitze weiterhin SUPERDOS und LDOS, mit denen ich bis jetzt nicht allzuviel anzufangen weiß.

Ich heiße Holm Voigtländer, bin 26 Jahre alt und Student der Informatik an der FHT (Fachhochschule für Technik) in Mannheim.
Mit dem Genie II arbeite ich seit ca. 2 Jahren. Die Maschine ist auf 48K ausgebaut. Dazu besitze ich (leider nur) 1 Disk-Drive ss/sd und einen Epson RX-80 Printer.
Ich verwende als Betriebssystem NEWDOS 80/2.0. Meine bevorzugten Sprachen sind BASIC (allerdings verwende ich meistens den ZBASIC 2.2 Compiler von SIMUTEK - bei der atemberaubenden Geschwindigkeit des Interpreters schläft man ja ein), PASCAL (5.0 Compiler von MOLIMERX) und ASSEMBLER (mit EDTASM+; wenn jemand den MACRO 80 - Assembler hat, bitte sofort melden!). Stark interessiert wäre ich an LISP. Wenn jemand Erfahrung damit hat - bitte mal schreiben.

Es gibt ein sehr gutes Buch für unsere Computer. Leider ist dieses in Englisch geschrieben und kostet stolze 129,00 DM. Ich spreche von "BASIC Faster & Better & Other Mysteries".

Dieses Buch zeigt unwahrscheinliche Möglichkeiten auf, über die sich jeder Programmierer freut. Vor einigen Tagen habe ich damit begonnen, das Buch durchzuarbeiten. Da mir das Buch so gut gefällt, beschloß ich bald darauf, dieses Buch ins Deutsche zu übersetzen und (da dies ja sehr lange dauern wird) in jedem Info mindestens 12 Seiten zu veröffentlichen. In diesem Info sind also die ersten 12 Seiten. Natürlich ist meine Übersetzung zum Teil katastrophal - aber was gemeint ist, dürfte immer klar sein. Die ganze Übersetzung läuft natürlich über eine Textverarbeitung, so daß Änderungen jederzeit möglich sind. Jetzt müßt Ihr mir dann nur noch mitteilen, ob Ihr an weiteren Seiten des übersetzten Buches interessiert seid.

Hilfsroutinen, Unterprogramme und Grundlösungen

Das BASIC des TRS-80 hat um die 150 Befehle und eingebaute Funktionen. Dabei sind nach Meinung mancher Programmierer der MERGE- und der DELETE-Befehl die einflußreichsten und wichtigsten. Vielleicht vertreten auch alle anderen Programmierer diese Meinung, wenn Sie diesen 'Kurs' durchgearbeitet haben.

HILFSROUTINEN

Hilfsroutinen in BASIC können viele Stunden der Programmierarbeit einsparen. Das GOSUB-Kommando und das RETURN-Kommando eröffnen viele Möglichkeiten. Hier seien einige Vorteile von Hilfsroutinen aufgeführt.

- Hilfsroutinen sparen Speicherplatz.
  Jede Operation, die mehr als einmal innerhalb eines Programms ausgeführt werden soll, muß nur einmal als Hilfsroutinen erscheinen.

- Hilfsroutinen sparen Programmierzeit.
  Bei Verwendung von Hilfsroutinen erspart man sich das fortwährende Abtippen gleicher Logik.

- Hilfsroutinen liefern Flexibilität.
  Einfache Änderungen an einem Programm, welches reichlich Hilfsroutinen benutzt, können bewirken, daß das Programm neue Funktionen ausführen kann, welche beim Schreiben des Original-Programms nie beabsichtigt waren.

- Hilfsroutinen sind einfach und schnell auszutesten.
  Mit Hilfsroutinen werden Programme in kleinere, überschaubare Module aufgeteilt, die leicht getestet und von Fehlern bereinigt werden können. Wenn man einmal eine Hilfsroutine fertig getestet hat, stimmt diese für immer.

- Hilfsroutinen befreien den Programmierer.
  Hilfsroutinen erlauben dem Programmierer, daß er sich nur auf das Hauptproblem und den Entwurf konzentriert. Er kann die Details und die komplizierten Zusammenhänge der Hilfsroutinen vergessen.

- Hilfsroutinen erleichtern das Verständnis.
  Hilfsroutinen machen Programme lesbarer und durchschaubarer. Die Details und die komplizierten Zusammenhänge allgemeiner Vorgänge unterbrechen nicht den Gedankengang des Hauptprogramms. Sogar wenn eine Routine nur einmal in einem Programm verwendet wird, kann es sich als wertvoll erweisen, diese als Hilfsroutine zu programmieren.

- Hilfsroutinen erleichtern die Programm-Konvertierung.
  Hilfsroutinen erleichtern die Konvertierung von Programmen auf andere Rechner. Wenn zum Beispiel ein Rechner nur den Bildschirm anders anspricht, so brauchen Sie nur die entsprechenden Hilfsroutinen abändern, und das Programm läuft auf dem anderen Rechner.

- Hilfsroutinen können zu einer Bibliothek werden.
  Man kann sich auf Diskette eine richtige Bibliothek mit Hilfsroutinen anlegen. Wenn man Hilfsroutinen braucht, werden die entsprechenden Hilfsroutinen einfach mit dem MERGE-Befehl an das entsprechende Programm angehängt.

Am Ende dieses BASIC-Kurses verfügt jeder über eine umfassende Bibliothek von Hilfsroutinen, die überall genutzt werden können.
Beinahe alle Hilfsroutinen haben Zeilennummern zwischen 40000 und 59999. Die Zeilennummern der Hilfsroutinen in diesem Kurs überschneiden sich nie, mit Ausnahme von einigen Fällen, wo zwei Hilfsroutinen die gleiche Funktion haben. Hier ist es sinnlos, alle zwei in der Bibliothek zu belassen.
Natürlich kann jeder die Zeilen-Nummern ändern; dadurch verzichtet er aber auf den Vorteil, eine geschriebene Dokumentation zu haben.
Die Variablen wurden willkürlich gewählt; man sollte aber auch diese aus obengenannten Grund beibehalten.

UNTERPROGRAMME

Ein Unterprogramm stellt eine Anzahl von Hilfsroutinen und Verfahren dar, die zusammen eine größere Funktion in einem Programm ausführen.

Zum Beispiel wird in diesem Kurs ein Unterprogramm zur Bildschirm-Ausgabe vorgestellt, welches eine vereinfachte Programmierung von Dateneingabe und Bildschirm-Untersuchung ermöglicht.

Unterprogramme haben die Vorteile der Hilfsroutinen, nur eben um ein Niveau höher. Sie stellen größere Systeme für z.B. Programm-Organisation, Disketten-Organisation, standartisiertem Computer-Dialog etc. dar.

Zur Erstellung von Unterprogrammen innerhalb eines Hauptprogrammes braucht man nur die entsprechenden Hilfsroutinen in das Programm einfügen (MERGE) und einige bestimmte Zeilen ändern, einfügen oder löschen.

Wer einmal mit standardisierten Unterprogrammen gearbeitet hat, wird sich wundern, wie das früher ohne gegangen ist.

GRUNDLÖSUNGEN

Grundlösungen sind Programme, welche durch einfache Änderungen für viele verschiedene Anwendungen verwendbar sind. In der Regel werden für Grundlösungen viele Unterprogramme und Hilfsroutinen verwendet.

Zum Beispiel kann man ein Dateiverwaltungs-Programm so schreiben, daß es, jeweils nur durch geringfügige Änderungen, für Mitgliederverwaltung, Adressenverwaltung, Lagerverwaltung, etc. verwendbar ist.

Wenn man für ein Problem ein Programm schreibt, so sollte man sich zuerst fragen, "Welche andere (bereits programmierte) Anwendung hat die selbe Grundstruktur?" Vielleicht hat man eine gut durchdachte Grundlösung bei der Hand, die man unter Einsparung von bestimmt 90 % der sonst notwendigen Programmierzeit an das Problem anpassen kann.

PROGRAMMIER-NORM

Nachfolgend ist die Belegung von Variablen, Konstanten und Zeilennummern festgelegt. Man kann sie so übernehmen, man kann sie auch ändern.

Wer sie ändert, läuft aber in Gefahr, daß er später bei Problemen auf diesen Kurs nicht mehr zurückgreifen kann, da er seine Version kaum mehr hier finden wird.

Alle Variablen sind als Integer-Zahlen vordefiniert mit Ausnahme von F, die als String für Disketten-Files und Bildschirm-Files definiert ist. Deswegen wird am Anfang eines Programms in der Regel "DEFINT A-Z" und "DEFSTR F" ausgeführt. Alle anderen Variablen werden ausführlich je nach Bedarf durch Verwendung der Symbole "$", "!" und "#" innerhalb des Programms definiert.

ARBEITSVARIABLEN

A$,A%,A!,A#	vorübergehende Speicherung (sehr kurz)
A1$-A5$,A1%-A5%, etc.	vorübergehende Speicherung (länger)
AN$	ausführender String, vorüberg. Speicher.
FX$,FX%,FL$,FL%	Kontrolle und Verzweigung
TC$,TC%,CD$,CD%	gegenwärtiger Ausführungscode

ZÄHLVARIABLEN

X%,Y%,Z%	FOR-NEXT Schleifen etc.

KONSTANTEN

KD$	gegenwärtiges Datum, 8-Byte Format
KS$	gegenwärtiges Datum, 2-Byte komp. Format
KD%, KM%, KY%	gegenwärtiger Tag, Monat, Jahr
CN$	Gesellschaftsname

GRAPHISCHE KONSTANTEN

SG$	horizontaler Streifen, STRING$(63,131)
C$	löschen bis zum Bildschirmende - CHR$(31)
C1$	löschen bis zum Zeilenende - CHR$(30)

BILDSCHIRMEIN- UND AUSGABE

PO%	gegenwärtige PRINT oder INPUT Position
AL%	max. Länge der gegenwärtigen Eingabe
PL%	PRINT Position - Beginn der gegenw. Zeile
LI%	erste Position im SCROLL-Teil
LV%	Zeilenanzahl im SCROLL-Teil
LT%	horizontale Tabulator-Position
LZ%	gegenwärtige Eingabe-Zeilennummer
LN%	höchste eingegebene Eingabe-Zeilennummer
LM%	Grenze, Zahl der Einträge
F1$()	Formatierter Bildschirm, Feldspeicherung

SUCHEN UND DISKETTEN-ZUGRIFF

KY$,FK$	Schlüssel suchen
RE$	String zurückgeben - Schlüssel gefunden

DRUCKER

OP$	Wahlstring anzeigen
TI$	Titel anzeigen
PN%	Seiten-Nummer
H1$	Überschrift Zeile 1 anzeigen
H2$	Überschrift Zeile 2 anzeigen

DISKETTEN-DATEIEN

FS$,FD$	Name der Disketten-Datei
PF%	gegenwärtige Datei-Nummer
PR%(PF%)	gegenwärtiger oder erwünschter physikal. Satz
PP%(PF%)	letzter physikalischer Satz
LR%(PF%)	gegenwärtiger oder erwünschter logischer Satz
LL%(PF%)	logische Satzlänge
L0%(PF%,0)-L0%(PF%,6)	gegenwärtige Datei-Angaben
FH$()	Feld-Variablen

USR-ROUTINEN

J%	Wert, der dem BASIC zurückgemeldet wird
US%(), UX%()	Speicherbereich einer USR-Routine
C%(), P%()	Kontrolle oder Parameter-Ordnung

```
0  Programm-Name, Copyright-Informationen, Datum letzte Änder.
1  Speicherplatz-Änderung, CLEAR-Befehl
2  DEF-Befehle - DEFUSR's, DEFINT's, DEFSTR's, etc.
3  DIM-Befehle - Felddimensionierung
4  Konstanten und Buchstaben, welche im Programm benutzt werden
:
30 USR-Routinen laden
:
50 Funktionen definieren
:
:
80 GOSUB's für Eröffnung von Dateien etc.
:
100  Ausgabe des Hauptmenü
190  Eingabe der Menü-Wahl.   ON GOTO - Befehl
:
200  weitere Menüs
:
900  Programmende, Dateien schließen
:
1000 Erste  Hauptroutine
2000 Zweite Hauptroutine
:
:
15000 Hilfsroutinen speziell für dieses Programm
:
:
40000 Standard-Hilfsroutinen, Tastatur und Bildschirm
41000 Standard-Hilfsroutinen, allgemein
:
57000 Standard-Hilfsroutinen, Drucker
58000 Standard-Hilfsroutinen, Handhabung von Disketten-Dateien
:
```

Gewaltige Funktionsaufrufe

Wer sein BASIC-Buch gelesen hat, wird sich beim Kapitel "Wie definiere ich Funktionen" gedacht haben, diese Funktionen hätten keinen großen Wert und seien mehr oder weniger sinnlos. Viele Programmierer ignorieren (leider) diese Fähigkeit des Funktionen-Aufrufs einfach.

Jeder soll nun selbst entdecken, welche mächtigen Eigenschaften mit dem Funktions-Aufruf möglich sind. Alle Programmierer werden nach diesem Kapitel (hoffentlich) eine andere Meinung von den Funktionen haben. Im typischen BASIC-Handbuch wird meist zur Demonstration, wie eine Funktion anzuwenden ist, die Verbindung zweier Strings vorgeführt.

```
10  DEFFNCS$(A$,B$) = A$ + " " + B$
20  INPUT "Vorname  eingeben "; F$
30  INPUT "Nachname eingeben "; L$
40  PRINT "Der vollständige Name lautet "; FNCS$(F$,L$)
```

Wenn Sie das Musterprogramm starten, so könnte der Dialog wie folgt aussehen:

```
Vorname  eingeben ?Günther
Nachname eingeben ?Wagner
Der vollständige Name lautet Günther Wagner
```

Nun wird die Reaktion vieler in etwa "Was soll's!" lauten.

Aber wir wollen uns anhand dieses einfachsten und sinnlosen Programms sorgfältig die Vorteile überlegen:

- Die Variablen, die bei der Definition der Funktion verwendet werden, bleiben beim Aufruf einer Funktion absolut unverändert. In diesem Beispiel werden A$ und B$ nicht verändert. Wenn A$ vor Benutzung von FNCS$(A$,B$) den String "GUENTHER" enthält, so beinhaltet er auch nachher "GUENTHER". Deswegen hat man absolute Freiheiten bei der Verwendung von Variablen. Man kann sich eine große Bibliothek von Funktionen anlegen, die man bei Bedarf einfach an das Programm mit dem MERGE-Befehl anhängt - ohne daß man sich um Variablen-Namen kümmern muß.

- Die Definition der Funktion kann in irgendeiner Zeilennummer des Programms vorgenommen werden. Die einzige Forderung: Die Definition der Funktion muß während des Programmablaufs vor dem ersten Aufruf der Funktion durchlaufen werden. Somit wird es sehr einfach, sich eine 'MERGE-Bibliothek' für Funktionen zu erstellen.

WENIG BEKANNTE TATSACHEN ÜBER FUNKTIONEN

Wer mit Funktionen experimentiert, wird feststellen, daß diese sehr flexibel sein können.
Hier sind einige der wenig bekannten Tatsachen, die Sie entdecken können:

1. Man kann in einem Programm Funktionen so oft umdefinieren, wie man will.

2. Eine Funktion kann sich auf andere Funktionen beziehen. Man kann Funktionen vernetzen, genauso wie Hilfsroutinen untereinander vernetzt werden können.

3. Eine Funktions-Definierung kann eine oder mehrere Maschinensprache-USR-Hilfsroutinen aufrufen.

4. Eine Funktions-Definierung kann Variablen benutzen, welche im Programm nicht vorkommen. Zum Beispiel sollen in einem Inventur-Programm sein:

   LC! Warenmenge bei letzter Inventur
   PR! gekaufte Warenmenge seit der letzten Inventur
   SO! verkaufte Warenmenge seit der letzten Inventur

   Nun kann mit z.B. FNOH!(O) die vorhandene Warenmenge bestimmt werden. Die Definierung der Funktion kann lauten:

   DEFFNOH!(A%) = LC! + PR! - SO!

   In diesem Fall ist 'A%' eine Schein-Variable, die in der Definierung nicht verwendet wird.
5. Eine Funktions-Definierung muß ein Ausdruck sein. BASIC-Ausdrücke wie PRINT oder POKE sind nicht erlaubt.

BENUTZUNG DER FUNKTIONSDEFINIERUNG ALS DOKUMENTATION

Funktions-Aufrufe können dokumentiert werden. In diesem Kurs wollen wir A1, A2, A3, etc. als Standard-Variablen-Namen benutzen, um den Inhalt einer Funktion anzugeben. Um die String-Verbindung aus unserem Beispiel zu dokumentieren, wollen wir zuerst die Funktion definieren und dann dokumentieren.

```
DEFFNCS$(A1$,A2$) = A1$ + " " + A2$
```

Wenn wir diese Funktion in unsere Bibliothek aufnehmen wollen, so könnte die Dokumentation wie folgt aussehen:

FNCS$(A1$,A2$) verbindet den ersten String mit dem zweiten und fügt ein Leerzeichen ein.

Eine andere Funktion könnte z.B. wie folgt dokumentiert werden:

FNRE#(A1#,A2#) gibt den Rest der Division der 1. Zahl durch die 2. Zahl aus.

Funktionen können sehr gut für viele mathematische Anwendungen definiert werden. Manchmal kann der Wunsch auftreten, eine Funktion direkt zu programmieren. Zum Beispiel ist die Funktion FNRE#(A1#,A2#) wie folgt definiert:

```
35 DEFFNRE#(A1#,A2#)=A1#-INT(A1#/A2#)*A2#
```

Falls man den Rest der Division X#/Y# haben will, dies aber nicht als Funktion definiert haben will, so kann man dennoch die definierte Funktion als Grundlage übernehmen. Es könnte dann obige Zeile wie folgt editiert werden:

```
420 PRINT "Der Rest ist ";X#-INT(X#/Y#)*X#
```

Wir haben einfach anstelle von 'A1#' und 'A2#' die Ausdrücke 'X#' und 'Y#' gesetzt. Jeder soll sich selbst aussuchen, ob er lieber Funktionen definiert oder diese programmiert.

IF-THEN LOGIK IN FUNKTIONEN

Wir nehmen folgendes Problem an:

B ist 1, wenn die Zahl A zwischen 100 und 300 liegt.
B ist 2, wenn die Zahl A zwischen 301 und 800 liegt.
B ist 3, wenn die Zahl A größer als 800 ist.
Ansonsten ist B=0.

Man kann das Problem durch einige IF-THEN-Ausdrücke lösen, aber man wird mehr als eine Programm-Zeile benötigen. Sie können es glauben oder nicht, aber die nachfolgende Funktion weist B den richtigen Wert zu:

```
B%=-(A%>=100)*-((A%>=100)+(A%>=301)+(A%>=801))
```

Um dies als Funktion FNCB%(A%) zu definieren, können Sie wie folgt vorgehen:

```
10 DEFFNCB%(A%)=-(A%>=100)*-((A%>=100)+(A%>=301)+(A%>=801))
```

Dann könnte die Programmzeile im Hauptprogramm wie folgt aussehen:

```
20 INPUT A%
30 B%=FNCB%(A%)
```

Der Schlüssel zu dieser Technik ist folgender: Ein logischer Vergleich liefert 0, wenn der Ausdruck falsch ist und -1 wenn der Ausdruck stimmt. Wenn zum Beispiel ein Programm den Ausdruck 'A%=1>2' enthält, so erhält A% den Wert 0. Wird der Ausdruck 'A%=1<2' verwendet, so erhält A% den Wert -1, da '1<2' eine wahre Aussage ist.

Im obigen Beispiel wurde jede mögliche Bedingung in Klammern gesetzt und das Ergebnis mit Addition und Multiplikation manipuliert, so daß B% immer den richtigen Wert erhält.

Durch Experimente und ein wenig Kreativität kann man mit Funktionen Unglaubliches anstellen. Und wenn eine Funktion einmal definiert und getestet ist, so braucht man diese nur noch in die entsprechenden Programme einfügen. Dieser Kurs enthät viele fertige Funktionen. Die Zeilennummern für die Funktions-Definierungen sind willkürlich gewählt und können abgeändert werden.

Einige Funktionen versprechen eine schnelle Ausführung, einige eine Verbesserung, einige Bequemlichkeit. Die meisten ersparen Programmierzeit. Schließlich ist ein Programm bequemer zu schreiben und leichter zu verfolgen.

Für die meisten Hilfsroutinen, USR-Routinen und Funktionen in diesem Kurs werden Beispiele mitgeliefert. Der beste Weg, um mit den Routinen vertraut zu werden, ist das Ausprobieren der Testprogramme. Auf diese Weise können Sie mit verschiedenen Änderungen und mehreren Datentypen experimentieren. Manchmal sind in den abgedruckten Listings für Tests und Demonstrationen aus Platzersparnis Hilfsroutinen nicht mitabgedruckt. Die benötigten Hilfsroutinen und Funktions-Definierungen sind separat gelistet und brauchen nur eingetippt oder von Diskette geladen werden.

USR-ROUTINEN - FÜR GESCHWINDIGKEIT UND FLEXIBILITÄT

Die schnelle und einfache Art der Programmierung von Computer-Anwendungen in BASIC wird von keiner anderen Programmier-Sprache übertroffen. BASIC erlaubt uns eine Verständigung mit dem Computer durch Befehle und mathematische Formeln, welche unserer Denkweise und Verständigung sehr nahe kommt. Aber wenn sehr schnelle Ausführungsgeschwindigkeiten und wirklich wirtschaftliche Speicherbelegung benötigt wird, müssen wir uns mit dem Computer in seiner eigentlichen Sprache, der Z-80 Maschinensprache, verständigen. Wenn wir den Computer von der Last der Übersetzung von BASIC in Z-80 Befehle befreien, so stehen uns hohe Geschwindigkeit und gewaltige Möglichkeiten zur Verfügung.

Aber nur selten ist es praktisch und sinnvoll, Anwenderprogramme komplett in Z-80 Maschinensprache zu schreiben. Für die meisten Programmierer ist das Erstellen, Testen und Ändern von Z-80 Programmen viel zu zeitintensiv und die Vorteile der hohen Geschwindigkeit und der Speicherersparnis werden meist nicht benötigt. Die sinnvollste Methode ist eine Bibliothek von kurzen Routinen, welche von BASIC aus aufgerufen werden wann und wo sie gerade gebraucht werden. Die Möglichkeiten der USR-Routinen erlauben uns den Sprung von BASIC in die Maschinensprache und von dort wieder zurück in das BASIC.

In diesem Kurs werden wir uns über viele spezielle Möglichkeiten von USR-Routinen unterhalten, ohne das Sie einen einzigen Z-80 Befehl kennen müssen. Aber wer bereit ist, den Sprung zu wagen, und mit der Programmierung in Maschinensprache beginnen will, der findet anhand der bereitgestellten Listings eine gute Startmöglichkeit. Mit einem Editor-Assembler können Sie die gezeigten Routinen ändern oder verbinden; Sie können auch neue Routinen von Anfang an entwerfen.

Alle in diesem Kurs vorgestellten USR-Routinen haben ein sehr wichtiges Merkmal gemeinsam - man kann sie in jede Stelle des RAM laden und ausführen. Tatsächlich werden wir in einigen Fällen Techniken verwenden, wo USR-Routinen während eines BASIC-Programms mehrere Male im Speicher verschoben werden.

Bestimmt werden Sie schon einige der guten Maschinensprache-Routinen für schnelles Sortieren und andere Anwendungen gesehen oder auch gekauft haben. Obwohl diese in der Regel sehr gut arbeiten, gibt es dennoch vier Probleme mit vielen dieser Produkte:

1. Sie sind so entworfen, daß sie in eine ganz bestimmte Stelle des Speichers geladen werden müssen. Man muß Speicherplatz reservieren, indem man die Frage 'MEMORY SIZE' richtig beantwortet. Pech haben diejenigen, welche noch andere USR-Routinen haben, die an der gleichen Stelle laden.

2. Eine Dokumentation der Assembler-Sprache wird gewöhnlich nicht mitgeliefert. Man kann nur schwer erkennen, wie das Programm arbeitet, und es ist schwierig, davon zu lernen oder etwas zu ändern.

3. Sie werden oft als Pakete verkauft, die mehr als eine Routine enthalten. Man muß die nicht benötigten Routinen zusammen mit den benötigten laden und verschenkt so wertvollen Speicherplatz.

4. Wenn man diese Routinen in Programmen benutzt, die man verkaufen will, muß man für die Routinen Tantiemen zahlen.

Die USR-Routinen, die wir in diesem Kurs besprechen, kennen diese vier Probleme nicht. Sie liefern hohe Flexibilität und Leistung. Und niemand braucht sich Gedanken über Abgaben machen (solange die USR-Routinen nicht als 'Bibliothek' verkauft werden oder für Dritte kopiert werden).

DAS SCHREIBEN VON USR-ROUTINEN MIT EINEM EDITOR-ASSEMBLER

Wir wollen uns die erforderlichen Schritte zur Erstellung eines Z-80 Maschinen-Programms anschauen. Zu genau werden wir dies hier aber nicht tun, da die Beschreibung Ihres Editor-Assemblers die Einzelheiten beinhaltet und die genauen Befehle von der jeweils benutzten Version abhängig sind. Wer keinen Editor-Assembler hat, sollte diesen Teil dennoch verfolgen - ein Editor-Assembler wird zur Verwendung der in diesem Kurs vorgestellten USR-Routinen nicht benötigt!

Wir wollen als Musterprogramm eine kurze Routine schreiben, die augenblicklich den Inhalt der Bildschirm-Position 0 auf die restlichen 1023 Bildschirm-Positionen kopiert. Wenn wir z.B. an der Position 0 ein 'X' stehen haben, so wird durch Aufruf der USR-Routine der ganze Bildschirm voll mit 'X'.

Mit einem Editor können wir folgendes eintippen:

```
00010 ;BFUELL - BILDSCHIRM-FUELL-USR-ROUTINE
00020 ;
00030         ORG     0BFF0H         ;BEGINN
00040         LD      HL,15360       ;HL-REGISTER
00050         LD      DE,15361       ;DE-REGISTER
00060         LD      BC,1023        ;1023 * WIEDER.
00070         LDIR                   ;WIEDERH. HL/DE
00080         RET                    ;RUECKSPRUNG
00090         END                    ;
```


1. Die Zeile 30 legt den Beginn (ORIGIN) der USR-Routine fest. Wir haben BFF0 gewählt. Das sind 16 Byte unterhalb der maximalen RAM-Adresse im TRS-80 mit 32K. Bei einem TRS-80 mit 48K könnten wir FFF0 wählen. Für das Assemblieren jeder Z-80 Routine auf dem TRS-80 ist ein Beginn über 3000 festzulegen (Grenze ROM/RAM). Wenn die Routine veränderlich erstellt wird (keine JP's oder CALL's auf absolute Adressen), braucht der gewählte Beginn der Routine nicht die Adresse sein, welche bei Ausführung der Routine gewählt wird. Für das Assemblieren und Testen sollte ein Beginn gewählt werden, der genügend weit unter der höchsten RAM-Adresse liegt. Damit werden Fehler vermieden. Berücksichtigt sollte auch werden, ob sich andere USR-Routinen zur selben Zeit im Speicher befinden müssen. Manchmal ist es etwas schwierig, den richtigen Beginn zu finden. Die meisten Assembler-Listings in diesem Kurs haben als ORG-Befehl einen Wert zwischen F000 und FF00. Wer einen TRS-80 mit 32K hat, kann dafür Werte zwischen B000 und BF00 verwenden. Bei allen Routinen kann aber der Beginn (ORIGIN) auch anders gewählt werden.

2. Die Zeilen 40 bis 80 beinhalten die eigentliche Programm-Logik dieser Routine. Wir laden in das HL-Register die Adresse des ersten Byte vom Bildschirm und in das DE-Register die Adresse des nächsten Byte. Dann laden wir in das BC-Register denn Wert 1023. Das LDIR-Kommando in Zeile 70 kopiert das Byte aus dem HL-Register in das DE-Register. Dann wird das HL- und DE-Register um den Wert 1 erhöht; vom BC-Register wird der Wert 1 abgezogen. Dieser Vorgang wiederholt sich, bis BC=0 ist. Das Ergebnis ist, daß das 1. Bildschirmzeichen 1023 mal kopiert worden ist. In Zeile 80 steht das RET-Kommando, das dem RETURN im BASIC entspricht. Falls wir diese Routine als USR-Routine von BASIC aus aufrufen, veranlaßt RET einen Rücksprung ins BASIC. Es wird mit dem nächsten Befehl weitergearbeitet (vgl. GOSUB).

3. Der Assembler verlangt ein END-Statement; dem wird in Zeile 90 Rechnung getragen.

Wenn wir nun alles eingetippt haben, können wir das Programm assemblieren und auf Diskette oder Tape abspeichern. Wir können auch das ursprüngliche Programm, den sog. Quellcode, abspeichern, damit wir zu einem späteren Zeitpunkt Änderungen an der Routine vornehmen können, ohne das nochmals alles eingetippt werden muß. Hier ist nun das assemblierte Listing unserer Bildschirm-Füll-Routine:

```
                    00010 ;BFUELL - BILDSCHIRM-FUELL-USR-ROUTINE
                    00020 ;
BFF0                00030         ORG     0BFF0H         ;BEGINN
BFF0 21003C         00040         LD      HL,15360       ;HL-REGISTER
BFF3 11013C         00050         LD      DE,15361       ;DE-REGISTER
BFF6 01FF03         00060         LD      BC,1023        ;1023 * WIEDER.
BFF9 EDB0           00070         LDIR                   ;WIEDERH. HL/DE
BFFB C9             00080         RET                    ;RUECKSPRUNG
03FF                00090         END                    ;
00000 TOTAL ERRORS
```

K A S S E N B E R I C H T : Stichtag 31.10.1984

E i n n a h m e n :

- Clubbeiträge	5 * 30,00 DM	=	150,00 DM
	19 * 20,00 DM	=	380,00 DM
- Werbung EDV-Zubehör, 3502 Vellmar	(2. Info)	=	11,00 DM
- Werbung Buchversand Polacek	(3. Info)	=	20,00 DM
- Werbung exep, Bad Schussenried	(3. Info)	=	45,00 DM
- Probe-Info an Interessenten		=	4,00 DM

Gesamtbetrag der Einnahmen bis 31.10.84 : 610,00 DM

A u s g a b e n :

- Anzeige in CHIP Nr. 4/84	=	30,00 DM
- Photokopien	=	254,52 DM
- Briefmarken und Postkarten	=	176,00 DM
- Kuverts	=	5,25 DM
- Copy Fluid	=	3,50 DM
- 250 Leitz-Trennblätter; 6 Leitz-Ordner	=	60,17 DM
- 11 Disketten	=	70,90 DM

Gesamtbetrag der Ausgaben bis 31.10.84 : 600,34 DM

Z u s a m m e n s t e l l u n g :

Einnahmen:	610,00 DM
Ausgaben :	600,34 DM
Kassenstand:	9,66 DM

Briefmarken sind im Gegenwert von 16,60 DM vorrätig.

ERLÄUTERUNGEN ZUM KASSENBERICHT

Natürlich möchte ich den Kassenbericht noch ein klein wenig erläutern.

Die Photokopien waren für:

Clubsatzungen, Aufnahmeformulare, Reklame	ca.	30,00 DM
3 Clubinfos	ca.	165,00 DM
2 Bücher der Bibliothek		11,60 DM
Fundgrube	ca.	50,00 DM

Für 1 Clubinfo ist mit mindestens 60 DM Kopierkosten zu rechnen (ich kopiere immer ca. 40 Stück, da diese bestimmt früher oder später gebraucht werden). Die Fundgrube umfaßt mindestens (vorsichtig geschätzt!) 1500 Blatt. Die Kosten sind dafür nur deshalb so niedrig, weil ich im September bei meiner Ferienarbeit im Büro zu einem Sonderpreis kopieren konnte, d.h. mein Chef verlangte nur pauschal 30 DM!

Bei den Briefmarken möchte ich nur darauf hinweisen, daß die Briefmarken des Clubs gesondert aufbewahrt werden. Wenn ich mit Mitgliedern in privater Sache verkehre, so kommen die Briefmarken natürlich nicht vom Club!

Die Leitz-Trennblätter benötige ich zur Programm-Archivierung - die 6 Leitz-Ordner sind die Verleih-Ordner der Fundgrube.

Die 11 Disketten sind die bisher benötigten zur Programm-Archivierung (2 gingen durch die Post kaputt).

Bis zum Jahresende fallen noch Ausgaben für dieses Clubinfo (Porto und Kopien) an. Ferner ca. 30 DM für ein Inserat in der CHIP (Dezember-Ausgabe). Auf alle Fälle wird das Jahr 1984 mit einem Minus abgeschloßen.

Diesen Kassenbericht habe ich ziemlich zusammengefaßt; ich bin natürlich jederzeit bereit, diesen Kassenbericht auf Wunsch im nächsten Info ausführlichst zu wiederholen. Ferner gehe ich im nächsten Info auf sämtliche Unklarheiten genau ein, welche aber natürlich von Euch nachgefragt werden müssen.

In jedem Fall wird im nächsten Clubinfo (das ja im Januar erscheint) der Jahresabschluß kurz abgedruckt.

So - nun überlegt mal, warum ich bereits in diesem Clubinfo noch vor Jahresende den Kassenbericht gebracht habe.

"Wer soll das bezahlen, wer hat soviel Geld ?"

Änderung der Clubsatzung

Ich, Günther Wagner, beantrage eine Änderung der Clubsatzung. Die Änderung betrifft den Punkt 7 unserer Clubsatzung und betrifft den Jahresbeitrag.

Bisher: .... Der Jahresbeitrag beträgt DM 20 ..

Angestrebte Änderung: .... Der Jahresbeitrag beträgt DM 30 ..

Begründung: Ich bin zu der Erkenntnis gekommen, daß der derzeitige Jahresbeitrag zur Clubfinanzierung nicht ausreicht. Meine Erkenntnis finde ich bestätigt, wenn ich den Kassenbericht betrachte. Behalten wir die 20 DM bei, so müssen Einsparungen (z.B. nur noch dünne Clubinfos) vorgenommen werden. Die Erhöhung auf 30 DM ist meines Erachtens der einzige Weg, um auch weiterhin vernünftige Clubarbeit zu leisten. Mit etwas Glück verbleibt dann auch am Jahresende ein kleines Plus in der Kasse (mögliche Anschaffungen!).

Für die angestrebte Änderung ist eine 3/4 Mehrheit der abgegebenen Stimmen erforderlich (also sollten alle den Zettel einsenden, egal ob Sie dafür oder dagegen sind). Der beiliegende Stimmzettel muß bis spätestens 1. Dezember bei mir eingetroffen sein!

Bezahlung des Jahresbeitrages 85

Der Jahreswechsel naht - wegen der beantragten Änderung des Mitgliedsbeitrages ist also der Jahresbeitrag für 1985 noch nicht bekannt. Wir treffen folgende Regelung:

Geht mein Antrag nicht durch (Jahresbeitrag bleibt bei 20 DM), so erhält jedes Mitglied bis spätestens 15.12.84 eine schriftliche Mitteilung.

Geht mein Antrag durch (neuer Jahresbeitrag 30 DM), so erhält niemand eine Mitteilung.

Wenn Ihr also bis 15. Dezember nichts gehört habt, so müssen 30 DM bezahlt werden, ansonsten 20 DM.

Die Zahlungen müssen bis spätestens 31. Dezember erfolgen. Die Zahlungsmöglichkeiten sind im Punkt 8 unserer Clubsatzung vermerkt. Wer nicht rechtzeitig bezahlt, erhält das nächste Clubinfo nicht (ich hoffe, Ihr habt Verständnis dafür). Wer austreten will (hoffentlich bleiben alle dabei!!!), muß dies bis spätestens 31.12.84 mitteilen, andernfalls muß er den Jahresbeitrag für 1985 bezahlen.

Mein Wunsch: Zahlt alle rechtzeitig; wer austreten will, teilt das bitte rechtzeitig mit. Dadurch wird mir die Verwaltungsarbeit sehr erleichtert!

FINANZIERUNG DES CLUB 80

In der letzten Zeit habe ich mir vermehrt Gedanken über die Club-Finanzierung gemacht. Ein Ergebnis ist die angestrebte Erhöhung des Jahresbeitrages. Eine weitere, zusätzliche Möglichkeit möchte ich Euch nun hier vorstellen.

Viele Mitglieder haben selbstgeschriebene Programme. Diese zusammen ergäben doch eine größere Anzahl. Es wäre nun möglich, über den CLUB 80 einen Software-Verkauf an Nichtmitglieder anzukurbeln. Die erzielten Gewinne dienen zur Mitfinanzierung des Clubs. Dadurch könnten wir unter Umständen einiges anschaffen. Damit auch der Programmauthor was vom Verkauf hat, kann ihm ein bestimmter Prozentsatz vom Programm-Preis bezahlt werden (z.B. 30 %). Der Programmauthor kann natürlich auch darauf verzichten.
In das Angebot des CLUB 80 könnten aber auch deutschsprachige Programmanleitungen und Beschreibungen aufgenommen werden (ev. auch Sachen aus der Adventure-Ecke), welche übern Club erstellt werden (siehe auch 3. Clubinfo). Auch hier können die jeweiligen Authoren am Verkauf prozentual beteiligt werden.
Den Verkauf muß aber ein anderes Clubmitglied übernehmen. Damit wird der Zeitaufwand für mich nicht zu groß, und gleichzeitig hat ein anderes Mitglied die Möglichkeit zur Mitarbeit. Damit nach außen hin jeweils die Angabe einer Adresse genügt, wird das Verkaufsangebot über die Kontaktadresse bezogen - im Verkaufsangebot wird dann auf die entsprechende Adresse für den Verkauf hingewiesen. Wer den Verkauf übernimmt, sollte nach Möglichkeit über 2 Laufwerke (dd) verfügen. Zu den Aufgaben gehört das Erstellen des Verkaufsangebotes, die Abwicklung des Verkaufes, das Führen einer Kasse, etc.

Zu überlegen ist, ob sich ein Verkauf überhaupt lohnt. Werbung und Unkosten müssen bestritten werden; ein großes Software-Angebot ist schon auf dem Markt. Fraglich ist, ob wir genügend interessante Programme, Anleitungen und Adventure-Lösungen zusammenbekommen.

So - wie findet Ihr diesen Vorschlag? Habt Ihr bessere Vorschläge oder zum Detail Verbesserungsvorschläge? Füllt bitte auf alle Fälle den 'Fragebogen' aus, damit ich Eure Meinung hierzu kenne - schreibt mir aber ruhig zusätzlich zu dem Thema!

Es sagte

. . . . ein Arzt zu seinem (männlichen) Patienten:
»Entweder hat sich der Computer geirrt, oder Sie sind im dritten Monat schwanger«

VORSCHLAG

Über den CLUB 80 wäre ein zusätzlicher Service möglich - abonnieren der '80micro' mit 'LOAD80'.

Die '80micro' ist die meiner Meinung nach beste amerikanische Computerzeitschrift für Tandy- (und damit auch Genie- und Komtek-) Freunde. Die Ausgaben enthielten bis jetzt immer mehrere (ca. 5) relativ umfangreiche Programme, die zum größten Teil wirklich hochinteressant sind. Mit 'LOAD80' kann man diese Programme fehlerfrei auf Diskette gespeichert beziehen.

Nun kostet die monatlich erscheinende '80micro' in Deutschlands Läden 18.50 DM - bezieht man sie direkt aus Amerika im Jahresabonnement, so kostet eine Ausgabe ca. 12.50 DM - schon wesentlich billiger aber immer noch sehr viel Geld. Will man sich das fehlerträchtige und vor allem zeitaufwendige und langweilige Eintippen der Programme ersparen, so abonniert man zusätzlich 'LOAD80'. Der Service kostet allerdings ca. 50 DM im Monat - für den Einzelnen meist zuviel.

Wenn man aber die Gesamtkosten von ca. 62.50 DM/Monat bzw. 750 DM/Jahr auf mehrere verteilen kann, wird's wieder interessant. Drei Möglichkeiten sehe ich:

1. Die '80micro' und 'LOAD80' nicht beziehen (und damit auf viele interessante Programme verzichten).

2. Die '80micro' und 'LOAD80' wird über den CLUB 80 abonniert - die Programme werden mit jedem Info an die Mitglieder zugesandt, bzw. in die Bibliothek aufgenommen. Eine Erhöhung des Mitgliedsbeitrages bzw. eine Finazierung auf andere Weise (Zusatzabgabe) wird nötig.

3. Mehrere Mitglieder tun sich zusammen und abonnieren gemeinsam die '80 micro' mit 'LOAD80'. Somit entfallen auf den Einzelnen z.B. 50 DM oder 75 DM (je nach Anzahl) jährlich. Die Programme würden dann aber auch frühestens nach z.B. 1 Jahr in die Bibliothek des Clubs aufgenommen, damit die anderen Mitglieder, welche nicht mitmachen, nicht die Nutznießer sind. Diese 2. Möglichkeit hat im Prinzip mit dem Club nichts zu tun, außer daß sich die Leute übern Club zusammenfinden.

Bei der Möglichkeit 2 und 3 müßte sich ein Mitglied bereit erklären, diese Angelegenheit zu übernehmen. Das Mitglied erhält die '80micro' und 'LOAD80'. Seine Aufgabe: Er muß für alle Programme eine deutsche (Kurz-) Anleitung erstellen. Bei der 2. Möglichkeit wird dann die Diskette mit den Anlitungen an mich zur weiteren Verteilung gesandt. Bei der 3. Möglichkeit muß das Mitglied die Diskette mit den Anleitungen selbst verteilen (Diskette und Anleitungen jeweils kopieren). Dadurch wird mein Zeitaufwand nicht erhöht - ein weiteres Mitglied hat die Möglichkeit zur Mitarbeit.

Auch hier bin ich wieder auf Euch angewiesen. Was haltet Ihr davon? Euere weitergehenden Vorschläge? Bitte füllt auf alle Fälle den Fragebogen aus - schreibt mir aber bitte ruhig noch Euere Meinung!

PS.: Natürlich kann man die '80micro' auch innerhalb des Clubs verleihen.

PROGRAMMANLEITUNGEN

Bisher mußte ich bei Programmbestellungen immer ziemlich viel Zeit für die Beilage der notwendigen Programmanleitungen verwenden. Braucht man zu diesem Programm eine Anleitung? Wenn ja, welche und wo ist diese?
Ich beginne daher zur Zeit damit, für jedes Programm die erforderlichen Unterlagen zusammenzustellen und in Ordnern abzuheften. Dazu gehört u.a. ein Ausdruck der Programmbeschreibung, ein Hinweis, auf welcher Diskette das Programm abgespeichert ist, und, falls nötig bzw. vorhanden, die Programmanleitung.
Es wird für mich somit leichter, die Programme zu kopieren und die entsprechenden Anleitungen beizulegen.
Genau da beginnt das Problem. Wer bezahlt bei zu kopierenden Anleitungen die Kopien? Der Club oder das Mitglied?

Mein Vorschlag:
Ich richte für jedes Mitglied ein Kopienkonto ein. In diesem werden die jeweiligen Kopierkosten festgehalten und zum Jahresende abgerechnet (1 Kopie kostet zur Zeit 0.10 DM). Die Kopierkosten können dann gemeinsam mit dem Jahresbeitrag bezahlt werden.

So - wie findet Ihr diesen Vorschlag? Habt Ihr bessere Vorschläge? Füllt bitte auf alle Fälle den 'Fragebogen' aus, damit ich Euere Meinung hierzu kenne. Sollte sich anhand des Fragebogens eine einfache Mehrheit ergeben, so gilt dies (nur bei diesem Thema) als beschlossen - schreibt mir aber ruhig zusätzlich zu dem Thema!

Wer kann dem Gerald Schröder weiterhelfen ?

Ich habe mir vor ca. zwei Jahren ein Video Genie I Mod 82 gekauft. Dazu war dann ein Expander EG 3014 nötig, um die zwei Diskettenstationen zu betreiben. Diese habe ich bei der Fa. Füssner erworben, wobei gesagt wurde, daß diese Double Density lauffähig seien. Dann kam mit dem Einbau der EG 64 von Schmidtke electronic der Speicherausbau auf 64K im Grundgerät, was dem neuen Mod 83 entsprechen soll. Allerdings kann ich den Expander weiter benutzen.
Kürzlich besorgte ich mir dann den Doubler der Firma Aumann. Dieser wird in den Expander eingebaut. Die Veränderungen, die mit dem Einbau der EG 64 vorgenommen wurden, sollten nach Aumann unwichtig sein.
Nach dem Einbau ließen sich mehrere Disketten nicht lesen, die ich in Double Density bekommen hatte. Also schickte ich die Laufwerke zu Füssner. Es erfolgte eine Reparatur für 100 DM, aber lesen ließen sie sich trotzdem nicht: Prüfsummen und "Daten-Satz-nicht-gefunden" Fehler traten massenweise auf. Eine erneute Überprüfung der Laufwerke durch Füssner brachte das Ergebnis: einwandfrei.
Nun konnten auch einige Tauschpartner plötzlich meine Software nicht mehr lesen und eine Umstellung auf den Tausch in Single Density brachte auch nicht den gewünschten Erfolg.
Die auf meinem System geschriebenen Programme sind für mich einwandfrei lesbar, und bis jetzt ist mir noch kein Programm durch einen Diskettenfehler verloren gegangen.
Füssner meinte dazu nur: Wenn der Doubler nicht von Percom kommt, kann das schon vorkommen.

POKE + PEEK ECKE)

1. POKE 16863,195: POKE 16864,163: POKE 16865,30

   Hier reagiert der LIST-Befehl wie ein RUN. Im BASIC-Programm muß aber die Zeile 0 existieren.

2. POKE 16396,195

   Beim drücken der BREAK-Taste wird das ganze Programm gelöscht und springt nach MEM SIZE.

3. POKE 16414,141: POKE 16415,5

   Die PRINT- und INPUT-Anweisungen werden über Drucker ausgegeben. Rückgängig machen mittels

   POKE 16414,88: POKE 16415,4

4. PEEK (16464) Abfrage SHIFT-Taste gedrückt = 1, ansonsten 0.

5. PEEK (14463) alle Tasten außer SHIFT (sonst wie oben)

6. PEEK (14591) alle Tasten

7. PEEK (16537) Speicherzelle für INKEY$; enthält immer den ASCII-Code der zuletzt gedrückten Taste.

Diese POKE's und PEEK's stammen vom Hans. Ich hoffe, daß wir auch für's 5. Info eine POKE + PEEK Ecke zusammen bekommen.

Bericht !

Im letzten Info hast Du noch einen kurzen Bericht über die Computerszene in Norwegen gegeben. Vielleicht ist dies deshalb für Dich interessant:
Ich habe eine Studienreise in die UdSSR gemacht und mich dort kurz mit einem Angestellten eines Instituts für Agrarforschung unterhalten können. Er erzählte mir, daß in seinem Institut nur sehr wenig Computer benutzt werden, und wenn, nur in vorgegebener Zeit eine Großrechenanlage. Sie machen Ihre Berechnungen immer noch mit Taschenrechnern und können die Großrechenanlage nur bei genau definierten Problemen benutzen. Im Bezug auf Heimcomputer, die dort in dieser Form nicht bekannt sind, berichtete er mir über einen Kurs mit japanischen Rechnern, den er mitgemacht hatte (scheinbar vom Betrieb aus). Über die Rechner bzw. Programme konnte er mangels Kenntnissen nichts genaues sagen. Interessant ist noch, daß man Computer mit beachtlichen Fähigkeiten kaufen kann (Speicherkapazität scheinbar im Megabyte-Bereich), die jedoch so viel wie drei bis vier Autos (Wolga) oder ein Mercedes kosten. Also für jeden Normalbürger unerschwinglich.
Im Gegensatz dazu wurde in der DDR, wie ich aus der DDR-Zeitschrift "Jugend und Technik" erfuhr, der erste eigene Homecomputer entwickelt. Natürlich von einem "Jugendkollektiv" mit sensationellen Patentanmeldungen, die jedoch nicht genauer spezifiziert wurden. Von Preis und Erscheinungsdatum war nichts zu lesen, genausowenig von technischen Details und Peripherie.

Gerald Schröd.

Progra. mwettbewerb

Zur Vertiefung von Programmier-Kenntnissen kann ein Programmwettbewerb dienen, wenn dabei die verschiedenen Versionen vorgestellt werden. Es tritt dann der sog. 'AHA'-Effekt auf (so machts der, bzw. ja, das ist ja auch möglich).
Alle Mitglieder, die gerne mitmachen wollen, sollen Ihre Lösung einsenden. Die interessantesten (ev. auch alle) werden im Clubinfo veröffentlicht. Wir wollen den ersten Versuch mit einem sehr einfachen, klar umrissenen Problem beginnen. Fangt nicht zum Gähnen an, wenn Ihr jetzt 'Primzahlen' lest, denn genau darum geht es.

<u>AUFGABE:</u>

Zu erstellen ist ein <u>reines Basic-Programm</u> zur Berechnung von Primzahlen innerhalb eines Intervalls. Das Intervall beginnt bei 1; die obere Intervallgrenze wird im Dialog mit dem Bildschirm eingegeben. Das Programm muß Primzahlen bis mind. 50000 berechnen können. Die Ausgabe sämtlicher Primzahlen innerhalb des Intervalls erfolgt unformatiert auf dem Bildschirm. Das Programm ist auf <u>höchste Geschwindigkeit</u> auszulegen (nicht Bedienungskomfort).

Die Geschwindigkeit wird in 3 Intervallen getestet:
1) 1 - 100
2) 1 - 1000
3) 1 - 10000

Die Zeitnahme erfolgt mit beendeter Eingabe der Intervall-Obergrenze und wird gestoppt, wenn die Meldung 'READY' bzw. 'ENDE' ausgegeben wird. Sieger ist, wer das schnellste Programm eingesandt hat. Als Preis winkt die Anerkennung der anderen Mitglieder.

Übrigens: Da die Programme sehr kurz sein dürften, könnt Ihr diese als Programm-Listing einsenden. Wer will, kann Sie auch auf Diskette zusenden. <u>Einsendeschluß ist Weihnachten.</u>

PS.: Primzahlen sind natürliche Zahlen, die genau zwei Teiler haben. Also 2, 3, 5, 7, 11, 13, ...

FUNDGRUBE !!

Die ersten 6 Ordner unserer 'Fundgrube' sind fertig. Die Fundgrube-Ordner beinhalten Kopien zum Thema Computer und speziell eben unsere.

Ihr könnt ab sofort die Ordner ausleihen. Das Porto trägt jeweils natürlich der Ausleiher. Die Ordner können maximal 4 Wochen ausgeliehen werden (in Ausnahmefällen nach Absprache auch länger). Die Kopien sind in der Regel von sehr guter Qualität, so daß diese ohne weiteres nochmals kopiert werden können. Den Inhalt der Ordner habe ich teilweise auf den nächsten Seiten wiedergegeben.

Natürlich wird unsere Fundgrube laufend erweitert. Alle meine Computer-Zeitschriften (Ausnahme 80micro) sind nun archiviert. CHIP beziehe ich regelmäßig; für die Archivierung der übrigen Zeitschriften bin ich auf Euere Zusendungen (2-fach) angewiesen.

CLUB 80 - FUNDGRUBE ORDNER 1

Schwerpunkt: Computer-Anwendungen

Aus dem Inhalt:

- Vokabel-Lernprogramme
- Gefriergut-Überwachung
- Pauken französischer Zahlen
- Aktien: Kaufen oder verkaufen?
- Welches Wertpapier bringt den größten Gewinn?
- Aktiendepot
- Programmreservoir
- Haushaltskasse
- Tilgunsplanberechnung (Wie lange zahle ich wieviel?)
- Rentenberechnung
- Tilgunsplan von Darlehen, Hypotheken und Bausparverträgen
- Spitzensteuerberechnung
- Einkommen/Lohnsteuer-Programm 1983
- Girokredit oder Ratenkredit?
- Zinsen ohne Formelzwang
- Bitgrafik mit dem Itoh 8510 A-Drucker
- Repeat-Funktionen für den TRS Model I in Basic
- Kombinatorik
- Der TRS-80 als gerechter Verteiler
- Printerplotter
- Dec-Dual-Hex-Umwandlung
- Matrixinversion
- Formeln sichtbar gemacht
- Querschnittswerte beliebiger Objekte
- Hochauflösende Grafik: Darstellung von Grafik auf dem MX-80
- Exponentielle Glättung
- Kurvendiskussion in Basic
- Kurvendiskussionen mit grafischer Darstellung
- Funktionsanalyse
- Druck von Kassetten-Titelblättern
- Darstellung von Aktienkursen
- TRS-80 plottet Funktionen
- Zeichnung von Eichkurven und Berechnung der Eichgeraden
- Das Hex-Ascii-Hex-Problem
- Basic-Programm differenziert Funktionen
- Grafik für TRS-80/Model I mit Drucker
- Hochauflösende Grafik mit dem TRS-80
- Kurvenentwicklung auf Bildschirm und Plotter
- Klinisch-chemische Präzisionskontrolle
- Währungsumrechnung
- Funktionsschaubilder

CLUB 80 - FUNDGRUBE ORDNER 2

Schwerpunkt: Computer-Anwendungen

Aus dem Inhalt:

- Abrechnung von Telefongesprächen
- Schaltunssimulation
- Text nach Reset wiedergefunden (Scripsit)
- Eine Studie über das Mischen von Karten
- Orakel (Horoskop nicht ganz ernst genommen)
- Umwandlung römischer in arabische Zahlen und umgekehrt
- Ein Karteiprogramm
- Benzinkosten- und Verbrauchsberechnung
- Bundesliga: Torschützenliste und Spieler mit gelber Karte
- Bundesliga: 10 verschiedene Tabellen
- Bundesliga: Liga-Tabellen
- Texte leichter texten mit Video Genie und TRS-80
- Verarbeitung beliebiger Texte mit dem Zeileneditor
- Textverarbeitung
- Textedit
- Mitgliederverwaltung
- Allgemeine Dateiverwaltung
- Adressenverwaltung
- Dateien auf Diskette
- Notizbuch
- Adreßdatei
- Genie mit Klingel
- Kalenderausdruck
- Lagerverwaltung
- Dateiprogramm
- Buchhaltung
- Verwaltung von Vereinen
- Weißt Du, wo die Sterne stehen?
- Monats-Kalender bis ins nächste Jahrhundert
- Horizont-Koordinaten des Mondes
- Ephemeriden
- Mondephemeriden
- Sag mir, wo die Sternlein stehen

CLUB 80 - FUNDGRUBE ORDNER 3

Schwerpunkt: Computer-Anwendungen

Aus dem Inhalt:

- Gewichtsprogramm
- Hochzeitstag bleib unvergessen
- Schulische Leistungen
- Haushaltsrechnung
- Wortspielereien
- Heizölkosten senken mit dem Computer
- Speisekarte vom Genie
- Zeugnisprogramm
- Lottosimulation 6 aus 49
- Kästchen-Programm
- Römische Zahlen umrechnen
- Der Zauberlehrling
- Partnervergleich: Biorhythmus einmal anders
- Rechnungen erstellen leicht gemacht
- Junggeselle bleiben oder heiraten ?
- Zahlen als Text
- Silbentrennprogramm
- Ihr ganz persönlicher Lügendetektor
- Daumenkino zum Selbermachen
- TRS-80: Jetzt hast Du Töne
- Geheimschrift
- Der ewige Kalender
- Stammbaum
- Schnelle Grafik in Basic
- Schwimmwettkampf
- Vorhang auf - ein effektvoller Programm-Vorspann
- Telefonbuch mit Pfiff
- Multiple-Choice-Generator
- Herr oder Frau?

ACHTUNG: Dieser Ordner wird noch erweitert!

CLUB 80 - FUNDGRUBE ORDNER 4

Schwerpunkt: Utilities, Sonstiges, Programmiersprachen

Aus dem Inhalt:

- Basicstart
- Tongenerator
- Editor für hochauflösende Grafik (384*192 Punkte)
- Edel-Basic
- Neue Basic-Befehle für den TRS-80
- Neue Basic-Befehle
- Append beim Video-Genie
- Toolkit (neue Basic-Befehle)
- INLINE- und PRINT-Tip für den TRS-80 und Video-Genie
- TRS-80 und Video-Genie - INLINE, PRINT und LPRINT
- GETSTRING in Basic
- Zwei neue Befehle für den TRS-80
- RESTORE N für den TRS-80
- Rekonstruktion gelöschter Files für TRS-80
- Editor für TRS-80 Model I
- Mühelos Masken erzeugen
- Print CRT (Bildschirmausdruck)
- Superdirectory für TRS-80
- Druckertreiber für TRS-80 und VG
- DIM-Befehl mit Sondereffekt
- Universeller Basic-Text-Editor
- Software-Single-Step für Z80-bestückte Mikrocomputer
- Fehelrmeldungen in Deutsch
- Umlaut mit Funktionstaste (Video-Genie II)
- Autostart - noch einfacher gemacht
- Autoedit - mit dem Zeileneditor leichter arbeiten
- Stapelverarbeitung in Basic
- Singlestep-Programm
- Videogenie-Maschinenprogramme per Basic geladen
- Trace-On
- Autostart bei Video-Genie
- Umlaute auf dem Genie I
- Lange Variablennamen für den TRS-80
- Alle Steuercodes verfügbar
- TRS-80/Video-Genie: Zwei Hilfsprogramme
- Schnellere Programmbedienung durch Kommando-Interpreter
- RAM-Disk kostenlos
- Hashing - ein schnelles Zugriffsverfahren
- Sprungtabelle
- TRS-80-Dateien schneller gelesen
- ABC - Assembler Basic Converter generiert DATA-Zeilen
- ABC für Diskettenbetrieb - Erweiterungen
- Neuer Zeichensatz
- Ein Tiny-Pascal-Compiler
- Alcor, ein neues Pascal für TRS-80, Apple, Osborne und CP/M
- Pascal 2.0 von Tandy

C L U B 8 0 - F U N D G R U B E O R D N E R 5

Schwerpunkt: Spiele

Aus dem Inhalt:

- Testen Sie Ihre Fahrkünste mit 'Chopper Commander'
- ZAP - eine Art Invaderspiel
- Schachmatt löst Schachprobleme
- Serpents
- Zauberwürfel
- Lernfähige Programme (am Beispiel NIMM)
- Suchsel
- Magische Quadrate
- Irrgarten
- Awari
- Quadrato
- Bergsteigen
- Galaxis
- Hinterhalt
- Black Jack
- Kniffel
- Mensch ärgere den Computer nicht
- Würfeln mit Kumpel Computer
- Skat - Computer simuliert Spielpartner
- Rotamint 3003 - Münzspielautomat
- Monopol
- Kingdoms
- Imbiss-Bude
- Börse
- Pferderennen
- Karylon
- Atlantic Adventure
- Abenteuerland
- Fantasia - ein schatzträchtiges Adventure

C L U B 8 0 - F U N D G R U B E O R D N E R 6

Schwerpunkt: Tips, Tricks, Berichte

Aus dem Inhalt:

- s e h r v i e l e Tips und Tricks zu TRS-80 und Genie
- Tandy auf der Hannover Messe '84
- TRS-80/Genie-kompatibel aber viel schneller
- Für den Anfänger: POKE und PEEK
- Selbstimporte aus dem EG-Bereich
- Relais-Ausgang für den TRS-80
- Fortran für TRS-80
- Grafikpaket für TRS-80
- Professioneller Assembler
- Genie-Text - besser als Konkurrenz?
- Schauen Sie dem Computer beim Rechnen zu!
- Zahlendarstellung
- Disketten doppelt genutzt: Bitte wenden
- Der Monitor im Video Genie
- Test: Genie III
- TRS-80 Modell 4 - drei Computer in einem?
- Modernisierung statt Neukauf
- Zwei Betriebssysteme sichern großes Programmangebot (Genie 3)
- Rund um den TRS-80 und Genie (großer Bericht mit vielen Adr.)
- Tandy's Printerplotter: Color Graphics Printer CGP-115
- Selbstbau Joystick
- Hochauflösende Grafik für TRS-80 (Teil 2)
- Serielle Schnittstellen

ACHTUNG: Dieser Ordner wird noch erweitert!

Hallo Günther,

Zuerst einmal herzlichen Dank für das 3. Clubinfo. Du schaffst es doch, trotz Zeitmangel ein informationsreiches Info zusammenzustellen. Ich freue mich jedes mal darüber. Gleichzeitig möchte ich Dir aber auch einen Vorschlag machen. Bei jedem Info sind ja die Seiten mit den neuesten Programmen dabei. Ich mache nun folgendes mit den Programmseiten: Ich löse sie aus dem Info heraus und hefte sie extra unter "Clubprogramme" ab. Dadurch habe ich auf einfachem Wege immer nach dem neuesten Info ein aktuelles Programmverzeichnis. Nun mein Vorschlag: Da Du die Seiten nicht immer ganz voll bekommst, könntest Du nicht auf dem verbleibenden Platz die Werbung oder ähnliches aufnehmen, so daß das Info nicht zerrissen wird. Zur Zeit kopiere ich mir halt die entsprechende Seite, damit ich das Info nicht zerstören muß und trotzdem die Clubprogrammsammlung weiterführen kann. Schreib mir mal, wie Du darüber denkst.

Auch daß die Adventure-Ecke zustande gekommen ist, finde ich sehr gut. Nun weiß ich endlich, wie nah ich der Lösung war. Es ist doch gut, wenn man von den Profis den Finaltip erhält.

Dabei möchte ich Dir gleichzeitig zu Deinem Problem mit dem Holger May antworten. Ich finde, Du bist voll im Recht mit Deiner Meinung, und wir sollten als doch recht junger Club schon auf unsere Finanzen achten. Lieber am Ende des Jahres einen Überschuß - und dann etwas Vernünftiges für alle anschaffen- als ein Loch in der Clubkasse. Wer es besonders eilig hat, kann ja das Rückporto beilegen, um so auf die Dringlichkeit seiner Post hinzuweisen. Dies werden aber sicher nicht viele sein. Und wer auch Disketten (wie laut Rechnung vom Holger) für 60 Pfennig kauft, kann sich eine solche "Mehrausgabe" sicher locker leisten. Ich würde eine solche Lösung empfehlen. Auch hast Du ja noch dein Zeitproblem, da an Dir ja sicher doch die meiste Clubarbeit hängen bleibt. Private Interessen kommen sicherlich auch noch hinzu. Deine Entscheidung für eine auf diese Art rationelle und günstige Art der Clubarbeit kann ich nur unterstützen. Die Überlegungen für ein Clubtreffen finde ich gut. Leider sind wir, wie nicht unschwer aus der Adressenliste zu entnehmen ist, ziemlich über die Lande verstreut. Vielleicht besteht aber die Möglichkeit, daß man sich zur Hobby-Tronik in Stuttgart am Wochenende treffen kann. Sicher gehen einige von unserem Club hin, und es wäre gut, die Mitglieder einmal persönlich kennen zu lernen. Da läßt sich doch telefonisch bestimmt etwas ausmachen.

Zum Schluß nun noch ein kurzer Beitrag für's Info:

Hochauflösende Grafik mit der Grafikzusatzplatine HRG1B

Die HRG1B wird von der Firma RB-Elektronik (Bouraueler Str. 13, 5208 Eitorf) vertrieben und paßt für den TRS-80 MOD I und die zwei VIDEO-GENIE 3003/8 & 1 + 2. Zur Zeit kostet die Fertigplatine 379,-- DM inclusive einem Treiberprogramm. Die HRG1B ist aber auch als Bausatz lieferbar.
Der Einbau (anschließen von 40 Drähtchen im Computer) ist aber auf jeden Fall noch selbst zu bewerkstelligen. Die Anleitung hierzu ist recht ordentlich. Man sollte die Verdrahtung aber sicherheitshalber mehrmals kontrollieren, denn wie schnell haben sich Fehler eingeschlichen.Ich habe mir, da ich einen Großteil der Bauteile schon hatte, nur die Platine (ca. 100,--DM) für die Grafik bestellt und die HRG1B selber aufgebaut. Auch dies ist sehr einfach.

Nun noch zur Grafik selbst. Sie ist parallel zur regulären TRS-80 Bildschirmausgabe und hat eine Auflösung von 0-383 Punkte in X-Richtung sowie 0-191 Punkte in Y-Richtung. Initialisiert wird die Grafik durch entsprechende Treiberprogramme, welche dann im NEWDOS-BASIC eine Reihe neue Befehle einfügen. Ich war erstaunt, wie schnell die Grafik arbeitet. Besonders durch die Befehle für Linien, Rechtecke oder Kreise erhöht sich die Geschwindigkeit erheblich. Des weiteren können Zeichensätze selbst kreiert werden. Dazu gibt es für mehrere Druckertypen die entsprechenden Programmversionen zum maßtabsgerechten Ausdruck. Zusätzlich lassen sich bei abgeschalteter Grafik zwei TRS-80 Bildschirminhalte abspeichern, welche dann zum Beispiel einen schnellen Bildwechsel bei Spielen möglich machen. Im übrigen sind die Funktionen der HRG1B von der angebotenen Software abhängig. Es ist also noch vieles möglich!
Meiner Meinung nach ist die HRG1B vom Preisleistungsverhältnis recht interessant und empfehlenswert.
Dies nun von meiner Sicht aus. Wer noch spezielle Fragen hat, kann sich ja ruhig an mich wenden. Ich hoffe, ich kann dann helfen.

Jens Neueder

P.S.: Das mit dem DU finde ich eine gute Sache. Es fördert sicher die Arbeit und Gemeinschaft im Club.

....

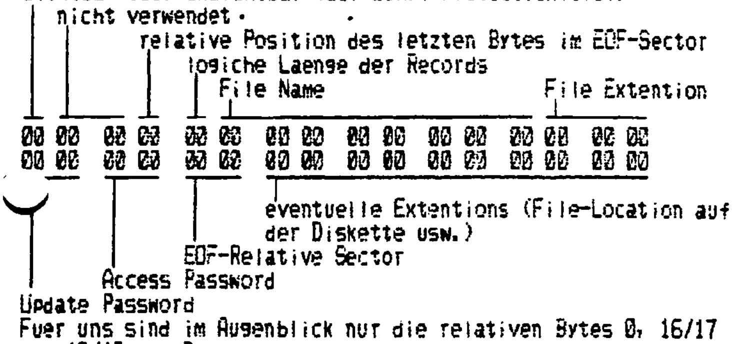
Dieser leicht gekürzte Brief vom Jens sollte einmal stellvertretend für mehrere ähnliche hier abgedruckt werden. Das mit dem Vorschlag wird gemacht - ganz klar!

Das Thema mit dem Holger sollte mit diesem Brief wieder beendet sein. Briefe in dieser Richtung erhielt ich mehrere. Ich hoffe nun nur noch, daß der Holger nicht 'beleidigt' die Kurve nimmt, sondern beim Club bleibt.
An dieser Stelle möchte ich darauf hinweisen, daß unser Mitglied Walter Piller freiwillig von sich aus 10 DM mehr Jahresbeitrag bezahlte, weil er in der Schweiz wohnt, und dadurch die Portokosten ein wenig höher sind. Es hat mich gefreut. Richtig ist, daß sehr oft das Zeitproblem eine schnellere Beantwortung meinerseits verhindert.

Den Bericht von der HRG1B finde ich sehr gut und habe deshalb an anderer Stelle auch noch was davon gebracht. Es ist zwar eine direkte Reklame für RB-Elektronik, aber ich finde die HRG sehr interessant (sie ist übrigens auch als Druckerpuffer verwendbar!). Wenn mehrere Leute die HRG anschaffen wollen, so gibt es Möglichkkeiten bei der Software und Hardware (Sammelbestellung = Rabatt). Deshalb mein Vorschlag - wer an der Anschaffung einer HRG interessiert ist, teilt mir dies wegen Koordination einer ev. Sammelbestellung mit!

Aendern und "Abschalten" von File-Passwoertern

Nachdem ich im letzten Clubinfo beschrieben habe wie man mit SUPERZAP das Masterpassword einer Diskette aendert, will ich heute auf das Filepassword und die Schutzebene (Protection-Level) von Diskfiles eingehen.
Sicher ist es jedem schon passiert, dass er eine File, die er vor ein paar Wochen mit einem Password gegen unerlaubten Zugriff geschuetzt hat, nicht mehr bearbeiten konnte, weil er das Filepassword vergessen hat. Dies ist vor allem dann aergerlich, wenn man mal nicht den Namen der Frau oder das Geburtsdatum der Freundin als Schutzwort benutzt hat und dieses somit auch assoziativ nicht mehr in Erfahrung zu bringen ist.
In einem solchen Fall ist es ganz nuetzlich ein Verfahren zu kennen, das Password mit Superzap zu veraendern oder den Protectionlevel auf "no restrictions" zurueckzusetzen.
Grundlagen: Direktory-Eintrag
Fuer jede File auf einer Diskette besteht in einem der Sectoren 2 - 9 des Direktory (DIR/SYS) ein 32 Byte Eintrag (FPDE = File primary Direktory Entry). In diesem Eintrag sind alle wichtigen Informationen ueber eine File enthalten. So auch Ihr Password und der Protectionlevel.
Der Eintrag setzt sich folgendermassen zusammen:
Filetype (System, No System); File gueltig, nicht gueltig; sichtbar oder unsichtbar (bei DIR); Protectionlevel.



Fuer uns sind im Augenblick nur die relativen Bytes 0, 16/17 und 18/19 von Bedeutung.
Das erste Byte eines jeden Direktoryeintrages (rel. Byte 0) enthaelt eine geballte Ladung Information. Diese ist versteckt in den einzelnen Bits des Bytes.

```
: 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 :
 Bit 7                      Bit 0
```

Die einzelnen Bits haben folgende Bedeutung:
Bit 7 - Identifizierung ob dies ein Haupteintrag (primary Entry) oder ein Zusatzeintrag (Extention Entry) ist
Bit 6 - System-File oder keine System-File (CMD, BAS usw.)
Bit 5 - nicht verwendet
Bit 4 - Directory-Eintrag gueltig oder nicht gueltig (gekillt)
Bit 3 - File sichtbar oder unsichtbar (bei DIR)
Bits 2,1 und 0 - Schutzlevel der File
Auch hier interessieren uns nicht alle Bits sondern nur die Bits 2,1 und 0. Sind alle drei Bits auf 1, so bedeutet dies, dass die File den hoechsten Protectionlevel (No access) besitzt. Sind diese Bits auf 0 gesetzt, ist der Passwordschutz abgeschaltet.
Die Bytes 16/17 enthalten den Hashcode des UPDATE-Passwords, die Bytes 18/19 den Hashcode des ACCESS-Passwords.
Aendern der Filepasswoerter:
Um UPDATE- und ACCESS-Password zu aendern braucht man im Grunde nur der Anleitung zum Aendern des Masterpasswords zu folgen (siehe Clubinfo 3). Die einzige Mehrarbeit besteht darin, dass man die Sectoren des Direktory nach dem Fileeintrag durchsuchen muss.
Protectionlevel auf "No restrictions" setzen:
Will man die Schutzebene aendern ohne die sonstigen Parameter des Byte 1 zu veraendern, muss man folgender Anweisung folgen:
1. Finden des Direktoryeintrages und Aufschreiben des Byte 0
2. Zerlegen des Bytes in seine Bits (Hex -) Dual - Konversion)
3. Die letzten drei Stellen des Bitmusters in 0 aendern
4. Umwandeln des Bitmusters in die Hexadezimalzahl
5. Aendern des Byte 0 mit Superzap (Funktion: MOD) wie im letzten Clubinfo beschrieben.

Schaut man sich einmal ein "ganz normales" Direktory an so faellt einem auf, dass das Byte 0 bei fast allen Benutzereintraegen (CMD-, TXT-, BAS- usw. Files) die Hexzahl 10 enthaelt. Im Normalfall reicht es also aus, das Byte 0 mit Superzap in eine 10 zu aendern um freien Zugriff zur File zu bekommen.
Auch hier ist wieder der Rat angebracht diese Prozedur nur an Backups durchzufuehren, um einen eventuellen Verlust der File zu verhindern.

Einzeiler: Flashing Display

Fast bei jedem selbstgeschriebenen Programm kommt es vor, dass man an bestimmten Stellen die besondere Aufmerksamkeit des Bedieners erregen muss. So z.B. bei Fehlermeldungen, Bedienungshinweisen usw.. Hier nun ein Vorschlag wie dies in BASIC mit nur einer Programmzeile zu realisieren ist.

```
1 CLS:P=INT(448+(32-LEN(A$)/2)):FORA=1TO10:PRINT@P,A$:
  FORZ=1TO99:NEXTZ:PRINT@P,STRING$(LEN(A$)," "):FORY=1
  TO99:NEXTY:NEXTA
```

Die zu uebergebende Meldung (z.B. Eingabe-Fehler) wird vom Hauptprogramm als Variable A$ uebergeben. Als erste Aktion wird der Bildschirm geloescht (kann nach Bedarf weggelassen werden). Danach wird die Anzeigeposition so errechnet, dass die Meldung immer "gemittelt" auf dem Bildschirm erscheint. Die erste Schleife (FOR A=) bestimmt die Anzahl der Blinkzyklen. Die zweite Schleife (FOR Z=) ist fuer die ON-Zeit, die Dritte fuer die OFF-Zeit massgebend. Wird das Programm von einem GOSUB aufgerufen, ist es zusaetzlich durch ein RETURN abzuschliessen. Weiterhin ist darauf zu achten, dass im Hauptprogramm genuegend Stringspace (CLEAR) reserviert wurde.

Unterdruecken der BREAK-Taste bei Model 1 und 3

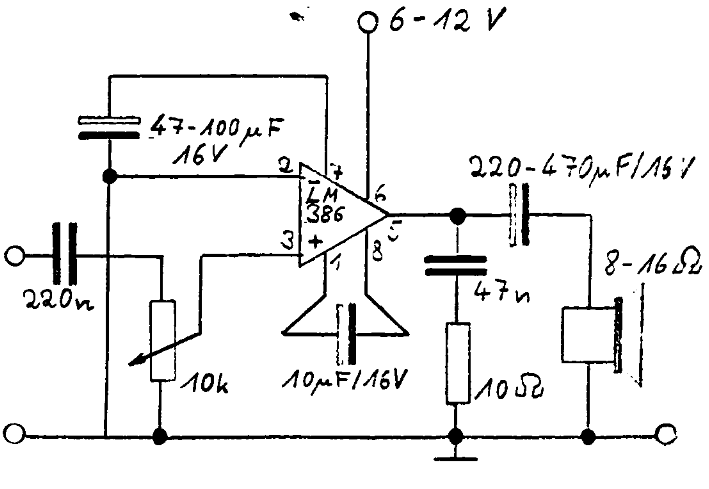
Im Programm CLUBADR/BAS stand ich vor dem Problem die BREAK-Taste funktionslos machen zu muessen, um den Verlust von Daten bei unbeabsichtigter Betaetigung zu vermeiden. Im NEWDOS Betriebssystem fuer das Model 1 ist dies sehr leicht zu realisieren. Man braucht nur den Inhalt der Speicherstelle 17170 von 201 in 195 zu veraendern. Veraendert man diese Speicherstelle beim Model 3, passiert im guenstigsten Fall garnichts. Hat man Pech, bringt man gehoerige Unordnung ins Betriebssystem und das Programm stuertzt ab. Was tun?
Das Problem laesst sich relativ einfach loesen. Man sucht sich im ROM der beiden Maschinen eine Speicherstelle deren Inhalt ungleich ist. Danach kann man eine kurze Routine schreiben, die je nach Model die entsprechende Funktion ausfuehrt.

```
1 IFPEEK(293)=32THENPOKE117170,195:ELSEIFPEEK(293)=73
  THENPOKE17528,195ELSEPRINT"KEIN MODEL 1/3, BREAK-TASTE
  NICHT UNTERDRUECKT!!!"
```

Die Speicherstelle 293 beinhaltet beim M1 den Charakter zwischen LEVEL II und BASIC, also eine Leerstelle. Beim M3 steht hier ein 'I'. Selbstverstaendlich kann man auch fuer andere kompatible Maschinen (z.B. KOMTEK) ROM-Identifizierungen durchfuehren.

Billiger NF-Verstaerker fuer den TRS 80

Die nachfolgend aufgefuehrte Schaltung mit dem LM 386, einem NF-Verstaerker-IC kleiner Leistung, kann direkt an den Tapeausgang des TRS 80 oder eines anderen Computers geschaltet werden. Sie macht den Anschluss eines grossen Verstaerkers zur Tonwidergabe (z.B.: Musikprogr., Spiele) ueberfluessig. Der Ruhestromverbrauch liegt bei ca. 3 mA. Die Ausgangsleistung betraegt bei der angegebenen Beschaltung ca. 500 mW an einem 16 Ohm Lautsprecher. Es wird eine Versorungsspannung von 6 - 12 Volt benoetigt. Ich habe diese der Stromversorung des Monitors entnommen in dem die Schaltung auch untergebracht ist. Die Wiedergabequalitaet ist zwar nicht ueberragend aber ausreichend.



* Hartmut Obermann *

Wilfried Klein

Wenn man einen Computer häufig benutzt, so wird man mehr und mehr Anwendungen für ihn sehen. Und es kommt die Zeit, daß Tastatu[illegible] Bildschirm als Ein- und Ausgabe nicht mehr ausreichen. Sicherlich sucht ma[illegible] nach einer Möglichkeit, den Computer mit der Außenwelt korrespondieren zu lassen. Und man weiß, daß ein Computer dazu befähigt ist – man muß nur wissen, wie man eine dafür geeignete Schaltung und die Software entwirft.

Der TRS-80 und die Außenwelt

Um eine Kommunikation mit der Außenwelt zu ermoglichen, muß man sich um 2 Dinge kümmern: Hardware und Software. Der erste Teil ist die Software. Es gibt nun 2 Moglichkeiten, Eingabe-Ausgabe-Programme zu schreiben:

1) als Maschinen- oder Assembler-Routine
2) mit Benutzung der Basic-Befehle IN, OUT, PEEK+POKE

Beim TRS-80-Level I und bei anderen Mikrocomputern ist eine Benutzung der entsprechenden BASIC-Befehle nicht moglich; hier muß man eine Maschinensprache-Routine schreiben, wahrend man beim Level II die BASIC-Befehle benutzen kann.

Die zusätzliche Hardware ist beim TRS-80 eine Notwendigkeit, denn weder der TRS-80 noch das Expansion-Interface haben Relais, Schalttransistoren oder Optokoppler eingebaut, um externe Gerate zu steuern. Und wenn man z. B. ein Relais an den vorhandenen Port anschließen wurde, würde das zu steuernde Gerät jedesmal in Aktion treten, wenn man ein Programm von der Kassette in den Arbeitsspeicher laden würde.

Wichtig ist es, zunächst die Software zu definieren, die man für das jeweilige Projekt benötigt, und anschließend erfolgt die technische Realisierung. Es gibt 2 Möglichkeiten, eine Steuerung von externen Geräten vorzunehmen:

1) als „Memory-mapped"-System
2) als Port-System

Bei einem Memory-mapped-System hat die CPU die Möglichkeit, das zu steuernde Gerät wie einen Speicherplatz anzusprechen. Man sucht sich eine nicht benutzte Adresse und verwendet sie als Ansprechpartner für die CPU. Wenn man nun Daten an das externe Gerät senden will, muß man dieses Gerät über den 16-bit-Adressen-Bus adressieren. Um die Übertragung zu ermöglichen, ist es noch erforderlich, den Write-Eingang der CPU zu aktivieren. Ist es notwendig, Daten in den Computer zu leiten, muß man den Read-Eingang aktivieren.

Nehmen wir einmal an, das externe Gerät soll mit der Adresse 36863 (dezimal) entsprechend 8FFF (hex) aktiviert werden und das binäre Datenwort zum Einschalten dieses Gerätes soll 02 sein.
LD 8FFFH, 02H; schalte das Gerät ein
So sieht die Assembler-Routine aus. LD ist die Load-Instruktion (Laden). Mit anderen Worten:

Lade die Adresse 8FFF Hex mit der Binär-Information 02 Hex.

Durch die Assembler-Routine wird ein Objekt-Maschinen-Code erzeugt, der die CPU aktiviert. In Basic würde diese Routine so aussehen:
POKE 36863,2
In einem Port-System hat man nun die Möglichkeit, 1 von 256 8-bit-Ports zu wählen, die die CPU adressieren kann. Dieses hat den Vorteil, nur 8 von 16 bit des Adressenbus benutzen zu müssen.

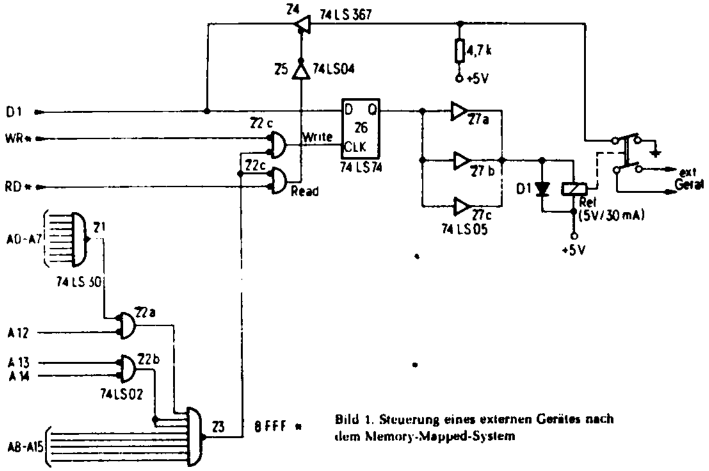


Bild 1. Steuerung eines externen Gerätes nach dem Memory-Mapped-System

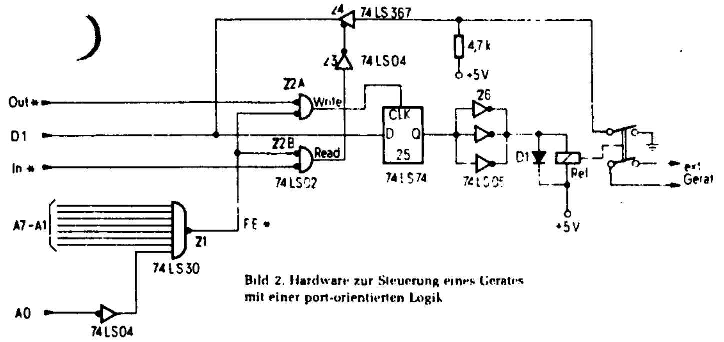


Bild 2. Hardware zur Steuerung eines Gerates mit einer port-orientierten Logik

Wir wählen als Beispiel Port 255 (dezimal) entsprechend FF Hex. Die Assembler-Routine lautet:

OUT FF, 02 H; Einschalten des externen Gerätes

In Basic Level II würde es so aussehen:
OUT 255,2

Bei einem Port-System verwenden wir also die Instruktionen IN und OUT und nur 8 Adreßleitungen für den Bereich 0000...00FF.

Soll man nun das Memory-Map- oder das Port-System verwenden? Das bleibt eigentlich jedem selbst überlassen. Entscheidet man sich für Memory-Map, dann sollten die I/O-Adressen so hoch wie möglich liegen, um Störungen mit dem Arbeitsspeicher zu umgehen. Wenn man das Portsystem benutzt, kann man 255 Adressenleitungen verwenden, weil der Kassettenrecorder den Port FF benutzt. Von der Hardware-Seite her bietet sich das Port-System eher an.

Warum? Es ist notwendig, extern eine Adressen-Decodierung vorzunehmen. Und während man beim Port-System nur 1 von 256 Möglichkeiten bei acht Adressenleitungen decodieren muß, fällt beim Memory-Map-System die Decodierung von 1 aus 65 536 Möglichkeiten bei 16 Adressenleitungen an.

Das Memory-Mapped-System

Bild 1 zeigt die notwendige Hardware für ein Memory-Mapped-I/O-System. Die sechzehn Adressenleitungen werden hier mit NAND-Gattern decodiert. Mit einem Reed-Relais läßt sich ein externes Gerät steuern. Ein zweiter Relaiskontakt dient als Rückmeldung, ob auch tatsächlich ein Schaltvorgang stattfand.

Zusätzlich benötigt man eine selbständige 5-V-Stromversorgung, weil diese Schaltung nicht aus dem TRS-80 gespeist werden kann.

Ein kleines Beispielprogramm:

```
10  REM STEUERUNG VON EXTER-
    NEN GERÄTEN
20  CLS: PRINT „SOLL DAS GERÄT
    EINGESCHALTET WERDEN";
30  INPUT A$ : IF A$ = „NEIN" THEN
    GOTO 60
40  POKE 4092,2
50  B = 2 : A = PEEK (4092) : REM AN-
    FORDERUNG NR. 3
60  END
```

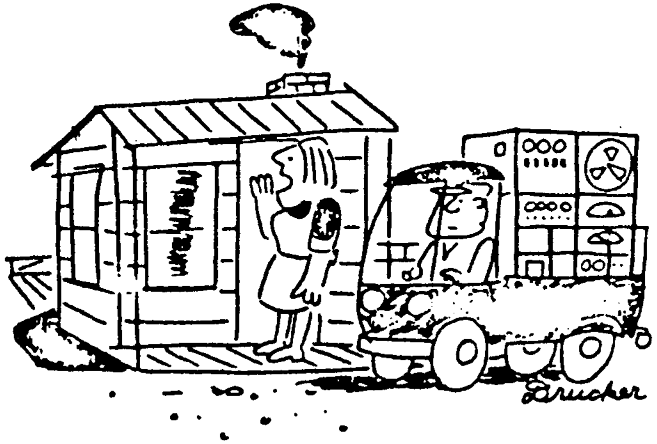
In Zeile 40 schreibt die CPU das Binärwort 2 in den Speicher unter der Adresse 4092 (binär: 1000 1111 1111 1111). Die ICs Z1, Z2a und Z3 dekodieren die Adresse für dieses 16-bit-Wort, welches auch als Signal am Ausgang von Z3 anliegt. Dieses Signal gelangt an die Eingänge von Z2c und Z2d. Die POKE-Instruktion bedingt, daß der Write-Eingang der CPU zur gleichen Zeit „Low" wird, wenn das 4092-Signal auf „low" geht. Dabei wird am Ausgang von Z2c ein „High"-Signal frei, welches als WRITE-Signal interpretiert wird und an den Takt-Eingang des Z6 eines Latches (Zwischenspeicher) gelangt. Das Datenwort „2", binär 0000 0010, gelangt als „High"-Signal über die Datenleitung D1 des Datenbus an den D-Eingang des Flipflops. Das Datenwort wird in Z6 abgespeichert. Das daraus resultierende „High"-Signal gelangt über den Ausgang Q von Z6 zu einem Relaistreiber Z7, dieser kann mit etwa 30 mA belastet werden. Während ein Kontakt das externe Gerät steuert, gibt der andere Kontakt ein Meldesignal über den Datenbus an den Computer zurück.

Das Port-System

Bild 2 zeigt das Schaltbild des Port-orientierten Systems. Wie man sieht, ändert sich hardwareseitig nicht viel. Im Basic-Betriebsprogramm ändern wir 2 Zeilen:

```
40  OUT 254,2
50  B = 2 : A = INP 254
```

Selbstverständlich ist es auf ähnliche Weise auch möglich, nicht ein externes Gerät zu steuern, sondern periphere Logikzustände abzufragen. Der Experimentierfreudigkeit sind also kaum Grenzen gesetzt.



„Charlie! Wo soll der neue Computer hin?"

H.O.

Jan Lichtermann

Allerlei Nützliches für den TRS-80

TRS-80-Besitzer müssen manche Kleinigkeit bei ihrem Computer in Kauf nehmen, die auf die Dauer ärgerlich ist. In den meisten Fällen kann man mit wenig Aufwand Abhilfe schaffen. Die im folgenden angesprochenen Tips und Kniffe haben sich in der Praxis vielfältig bewährt.

Die Dauer eines Tastendrucks wird festgestellt

Die INKEY-Funktion hat den Nachteil, daß nach einmaligem Tastendruck die Dauer der Betätigung nicht ermittelt werden kann. Möchte man mit einer Taste, z. B. in einem Grafikprogramm, eine Dauerfunktion erreichen, so kann man die Tastatur mit der PEEK-Funktion direkt abfragen und in einer Programmschleife einen Befehl so lange ausführen, wie die Taste betätigt wird. Die *Tabelle* gibt Aufschluß darüber, welches Argument zur Abfrage einer bestimmten Taste erforderlich ist. Wird die Taste betätigt, so erhält man den zugehörigen Wert als Resultat der Abfrage. Ist keine Taste betätigt, ergibt die PEEK-Abfrage den Wert 0. *Bild 1* verdeutlich das Prinzip der Tastaturabfrage durch PEEK.

Soll aus einem bestimmten Grund, etwa zur Erhöhung der Aufmerksamkeit, eine Stringvariable durch einen blinkenden Cursor eingelesen werden, so empfiehlt sich die Nachbildung des INPUT-Befehls durch die INKEY-Funktion. Durch die Wahl der Schleifenlänge kann man die Blinkfrequenz des Cursors beeinflussen. *Bild 2* zeigt, wie das Einlesen eines Strings mit der INKEY-Funktion bewerkstelligt wird.

Ausgabe im technischen Format

Obwohl das TRS-80-Basic die komfortable PRINT-USING-Anweisung beinhaltet, ergeben sich bei der Ausgabe von Zahlen manchmal Schwierigkeiten. Von der Technik her schätzt man das Arbeiten mit den Kürzeln piko (10^{-12}), nano (10^{-9}), Mega (10^{6}), Tera (10^{9}), Giga (10^{12}). Von Kürzel zu Kürzel unterscheidet sich der Exponent um jeweils 3. Eine automatische Anpassung wäre deshalb in vielen Fällen wünschenswert. Das in *Bild 3* dargestellte Unterprogramm erledigt das. Übergeben wird die auszugebende Zahl Z, ebenfalls verwendet werden die Variablen ZZ und V. Letzteres kann eine Ganzzahlvariable sein, lediglich ZZ und Z müssen vom selben Typ sein, also entweder von einfacher oder doppelter Genauigkeit.

Tastenprellen und Ladeprobleme

Vielen mag das Tastaturprellen beim TRS-80 schon Kopfzerbrechen bereitet haben. Sind die Tastenkontakte verschmutzt, so werden Buchstaben zum Teil doppelt geschrieben. Tandy empfiehlt dagegen das Reinigen der Tastaturkontakte oder die Verwendung eines kleinen Maschinenspracheprogramms KBFIX. Dieses Programm konnte ich von Tandy kostenlos erhalten, und es arbeitet zu meiner vollen Zufriedenheit. Ist das Programm geladen, wird das Tastaturprellen wirkungsvoll per Software eliminiert.

Ein anderes beinahe noch kritischeres Problem ist das Laden der Programme vom Kassettenrecorder. Dabei ist die Lautstärke am Recorder peinlich genau einzustellen, da sonst immer wieder Ladefehler auftreten. Auch ist die Lautstärke je nach Bandsorte verschieden zu wählen. Um so erfreuter war ich, als ich las, daß Tandy eine Zusatzplatine einbaut. Der kostenlose Einbau (er erforderte zwar einige Schreibarbeit meinerseits, da die Firma Tandy nicht sofort reagierte) hat sich durchaus gelohnt. Ladefehler treten jetzt nur noch in Ausnahmefällen auf.

Weitere Hardwareverbesserungen einfachster Art sind der zusätzliche Einbau zweier Tastschalter rechts neben der Tastatur. In der Klappe zur Erweiterung des TRS-80 befindet sich ein Taster, der beim Schieflaufen des Ladevorgangs bzw. eines Maschinenspracheprogramms betätigt werden kann und einen nichtmaskierbaren Interrupt am Prozessor auslöst. Die Betätigung dieses Tasters ist recht umständlich: Man muß jedesmal die Klappe öffnen und ihn auf der Rückseite mühsam ertasten. Deshalb kann man einen einfachen Taster parallel dazu legen und neben der Tastatur montieren. Dies empfiehlt sich sehr für Experimente mit Maschinensprache. Der Prozessor kann damit jederzeit ohne Programmverlust unterbrochen werden.

Bekannt ist jedem TRS-80-Anwender die nach dem Einschalten des TRS-80 erscheinende Frage: MEMORY SIZE?, die zur Reservierung eines vor Basic geschützten Arbeitsbereichs für Maschinensprache dient. Ausgelöst wird dieser Vorgang hardwaremäßig nach dem Einschalten durch den sogenannten Power-On-Reset. Arbeitet man viel mit Maschinensprache, so sind manchmal Umreservierungen nötig. Dies würde ein Aus- und Einschalten der Anlage erforderlich machen, wobei die Anlage einige Sekunden ausgeschaltet bleiben muß. Hier kann man einen weiteren Taster einbauen, der diesen Power-On-Reset simuliert. Dadurch braucht man die Anlage nicht an der Rückseite ein- und auszuschalten.

Betrachten wir einmal den Schaltungsausschnitt in *Bild 4*: Wird das Gerät eingeschaltet, so ist C42 entladen; am Eingang von Z53 liegt also L-Pegel an, damit auch am Ausgang von Z52. Dies veranlaßt den Prozessor dazu, mit der Befehlsausführung bei Adresse 0000 zu beginnen, sobald der Reset-Eingang wieder H-Pegel erhält. Dies geschieht dadurch, daß C42 über R47 langsam aufgeladen wird. Im Betriebszustand ist C42 also geladen. Legt man jetzt einen Taster parallel zu C42, kann man ihn auch während des Betriebs entladen und damit den Einschaltvorgang simulieren. Dazu muß man C42 auf der Platine suchen und eine entsprechende Leitung anlöten. Ich verwende hierzu eine abgeschirmte Leitung, damit im Videoteil keine wilden Schwingungen ausgelöst werden. Doch Achtung! Versehentliches Betätigen dieses Tasters löscht ein Basic-Programm im Speicher. Auch verliert man durch einen solchen Eingriff die Garantie des Herstellers.

Der Speicher wird erweitert

Bekanntlich kann ein 4-KByte-Modell auf 16 K erweitert werden. Dazu werden die Speicherchips ausgetauscht. Zusätzlich müssen zwei Shunts ausgetauscht werden. Führt man diese Arbeiten selbst aus, kann man eine Menge Geld sparen! Benötigt werden acht Chips vom Typ 4116 (450 ns oder schneller). Es handelt sich um dynamische NMOS-RAMs. Beim Einbau sind unbedingt die Handhabungsregeln von MOS-Bausteinen zu beachten, d. h. auf jeden Fall ist statische Aufladung zu vermeiden. Man entfernt die alten Speicherchips Z13 bis Z20 und setzt unter Beachtung der Richtung (Kerbe) die neuen ICs ein. Dann wechselt man die DIP-Shunts A3 und A71 aus. Das folgende gilt nur für die D-Version des TRS-80 (letzter Buchstabe der Nr. an Unterseite des Gehäuses) und „BASIC LEVEL II": Bei A3 sind sämtliche Kontakte zu überbrücken, bei A71 sind Brücken gemäß *Bild 5* herzustellen. Möchte man keine Shunts kaufen, so kann man die Verbindung mit Schaltdraht herstellen. Korrekte Verbindungen sind die erste Voraussetzung für den Betrieb, sonst läuft der Computer nicht mehr. Hat man eine andere Platinenversion oder kein Level-2-Basic, so sind die Brücken anders herzustellen! Die Funktion des eingebauten neuen Speichers läßt sich durch die Eingabe von PRINT MEM grob überprüfen: Bei leerem Arbeitsspeicher muß die Zahl 15572 erscheinen.

Tabelle zur Tastaturabfrage

Zeichen	PEEK()	Wert
@	14337	1
A	14337	2
B	14337	4
C	14337	8
D	14337	16
E	14337	32
F	14337	64
G	14337	128
H	14338	1
I	14338	2
J	14338	4
K	14338	8
L	14338	16
M	14338	32
N	14338	64
O	14338	128
P	14340	1
Q	14340	2
R	14340	4
S	14340	8
T	14340	16
U	14340	32
V	14340	64
W	14340	128
X	14344	1
Y	14344	2
Z	14344	4
frei	14344	8
frei	14344	16
frei	14344	32
frei	14344	64
frei	14344	128
0	14352	1
1 !	14352	2
2 "	14352	4
3 #	14352	8
4 $	14352	16
5 %	14352	32
6 &	14352	64
7 '	14352	128
8 (	14368	1
9)	14368	2
: *	14368	4
; +	14368	8
, <	14368	16
– =	14368	32
. >	14368	64
/ ?	14368	128
ENTER	14400	1
CLEAR	14400	2
BREAK	14400	4
↑	14400	8
↓	14400	16
←	14400	32
→	14400	64
SPACE	14400	128
SHIFT	14464	1
frei	14464	2
frei	14464	4
frei	14464	8
frei	14464	16
frei	14464	32
frei	14464	64
frei	14464	128

```
10 A=PEEK(14337):IF A=2 THEN PRINT"A";:GOTO 10
20 IF A=4 THEN PRINT"B";:GOTO 10
30 IF A=8 THEN PRINT"C";:GOTO 10
40 GOTO10
```

Bild 1. Das Programm gibt so lange einen der Buchstaben A, B oder C aus, wie die entsprechende Taste gedrückt ist

```
10 B$=""
20 PRINT CHR$(14);:FOR N=1TO50:A$=INKEY$:IF A$=""THEN NEXT ELSE GOSUB 100:NEXT
30 PRINT CHR$(15);:FOR N=1TO50:A$=INKEY$:IF A$=""THEN NEXT ELSE GOSUB 100:NEXT
40 GOTO 20
100 IF ASC(A$)=13 THEN PRINT CHR$(14);A$;:GOTO 130
110 IF ASC(A$)=8 AND LEN(B$)>0 THEN B$=LEFT$(B$,LEN(B$)-1):PRINTA$;:RETURN
120 IF ASC(A$)<>8 THEN B$=B$+A$:PRINTA$;
125 RETURN
130 REM STRING STEHT ALS B$ BEREIT, NACH BEARBEITUNG MIT RETURN ZURUECKKEHREN !
```

Bild 2. Das Programm liest einen String ein, dabei wird ein blinkender Cursor erzeugt – Enter schließt den String ab

```
60000 IF Z=0 THEN V=1:GOTO60010 ELSE ZZ=ABS(Z):V=FIX(LOG(ZZ)/LOG(10))-FIX(LOG
      (ZZ)/LOG(10)/3)*3+2:IF ZZ<1 THEN V=V+2
60010 PRINTUSING STRING$(V,"#")+".###↑↑↑↑";Z:RETURN
```

Bild 3. Das Unterprogramm gibt die Zahl Z im technischen Format aus

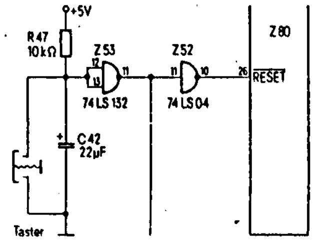


Bild 4. So wird der Power-On-Reset beim TRS-80 erzeugt

Literatur

[1] TRS-80 micro computer technical reference handbook. Radio Shack.
[2] Stübs, Martin: Programmieren mit TRS-80. Ing. W. Hofacker GmbH.

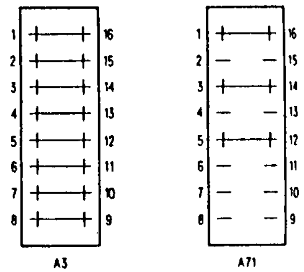


Bild 5. Erweitert man den Speicher des TRS-80 von 4 KByte auf 16 KByte, dann müssen bei der D-Version (letzter Buchstabe der Nummer auf der Unterseite des Gehäuses) die eingezeichneten Verbindungen hergestellt werden

Michael Haßelberg

422 neue Z80-Befehle

Hätten Sie gedacht, daß der Z80 noch vielseitiger ist, als ohnehin schon behauptet wird? Der vorliegende Artikel zeigt diese Möglichkeiten auf. Durch eifriges Experimentieren fand der Autor weitere 422 Opcodes heraus, die den ohnehin schon großen Befehlsvorrat des Prozessors von 694 auf 1116 Opcodes erweitern. Neben neuen Verschiebeoperationen kommen unter anderem weitere Befehle für die indizierte Adressierung hinzu.

Es existieren zunächst acht weitere Shift-Operationen, wobei das betreffende Register (A – L; (HL)) wie beim SLA-Befehl nach links geschoben wird, hierbei wird jedoch von rechts her mit logisch 1 aufgefüllt (Tabelle 1).
Die nun folgenden Erläuterungen und Befehle beziehen sich zwar auf das IX-Register, jedoch gilt das Gesagte sinngemäß auch für IY; hierbei ist dann das erste Byte des Opcodes von hex DD in FD zu ändern. Die *Tabellen 2 und 3* geben einen Überblick über zusätzliche Operationen bei den Shift-Befehlen, hierbei wird das Resultat der Operation in eines der Register A...L übertragen. In *Tabelle 2* sind noch einmal die oben beschriebenen Instruktionen bezüglich des Indexregisters IX aufgeführt. Die in *Tabelle 4* aufgezeigten Einzelbitoperationen verhalten sich im Prinzip so wie die Schiebebefehle, denn auch hierbei wird das Resultat der Operation in eines der o. g. Register transferiert. Schließlich geben die *Tabellen 5 und 6* noch einen Überblick über arithmetisch-logische Operationen direkt mit den einzelnen Hälften der Indexregister, wobei mit HX die Bits 8...15 von IX, mit LX die Bits 0...7 von IX gemeint sind. Hierdurch werden diese Indexregister zu vollwertigen Registerpaaren wie HL, DE oder BC.

Diese Liste erhebt natürlich keinen Anspruch auf Vollständigkeit, sie soll nur weitere Anregungen zum Erforschen der inneren Struktur des Z80 geben. Daher ist die Redaktion auch für weitere Ideen und Anregungen jederzeit dankbar.

Tabelle 3: Verschiebeoperationen bezüglich (IX + off) mit anschließendem Transfer in ein Register

	A	L	H	E	D	C	B
RLC (IX + off)	07	05	04	03	02	01	00
RRC (IX + off)	0F	0D	0C	0B	0A	09	08
RL (IX + off)	17	15	14	13	12	11	10
RR (IX + off)	1F	1D	1C	1B	1A	19	18
SLA (IX + off)	27	25	24	23	22	21	20
SRA (IX + off)	2F	2D	2C	2B	2A	29	28
SRL (IX + off)	3F	3D	3C	3B	3A	39	38

Tabelle 1: Verschiebeoperationen SLIA r

	SLIA r	
B	CB30	
C	CB31	
D	CB32	shift left inverted arithmetic r
E	CB33	Cy ◆ 7 ◆ 0 ◆ 1
H	CB34	
L	CB35	
(HL)	CB36	
A	CB37	

Tabelle 2: Verschiebeoperationen bezüglich (IX + off) mit anschließendem Transfer in ein Register

XX	B	C	D	E	H	L	(IX+ of)	L
SLIA (IX+of)r	30	31	32	33	34	35	36	37

Opcode: DDcBofXX (of = Offset)

Cy ◆ 7 ◆ 0 ◆ 1
7 • • • 0

Tabelle 4: Einzelbitoperationen bezüglich (LX + off) mit anschließendem Transfer in ein Register

Opcode: DDCBofXX

XX	RES 0, (IX + off)	RES 1, (IX + off)	RES 2, (IX + off)	RES 3, (IX + off)	RES 4, (IX + off)	RES 5, (IX + off)	RES 6, (IX + off)	RES 7, (IX + off)	SET 0, (IX + off)	SET 1, (IX + off)	SET 2, (IX + off)	SET 3, (IX + off)	SET 4, (IX + off)	SET 5, (IX + off)	SET 6, (IX + off)	SET 7, (IX + off)
B	80	88	90	98	A0	A8	B0	B8	C0	C8	D0	D8	E0	E8	F0	F8
C	81	89	91	99	A1	A9	B1	B9	C1	C9	D1	D9	E1	E9	F1	F9
D	82	8A	92	9A	A2	AA	B2	BA	C2	CA	D2	DA	E2	EA	F2	FA
E	83	8B	93	9B	A3	AB	B3	BB	C3	CB	D3	DB	E3	EB	F3	FB
H	84	8C	94	9C	A4	AC	B4	BC	C4	CC	D4	DC	E4	EC	F4	FC
L	85	8D	95	9D	A5	AD	B5	BD	C5	CD	D5	DD	E5	ED	F5	FD
A	87	8F	97	9F	A7	AF	B7	BF	C7	CF	D7	DF	E7	EF	F7	FF

Tabelle 5: Arithmetisch-logische Befehle bezüglich IX

INC HX	DD24	SUB LX	DD95
DEC HX	DD25	SBC HX	DD9C
INC LX	DD2C	SBC LX	DD9D
DEC LX	DD2D	AND HX	DDA4
ADD A, HX	DD84	AND LX	DDA5
ADD A, LX	DD85	XOR HX	DDAC
ADC A, HX	DD8C	XOR LX	DDAD
ADC A, LX	DD8D	CP HX	DDBC
SUB HX	DD94	CP LX	DDBD

Tabelle 6: 8-Bit-Ladebefehle bezüglich IX

LD HX, dd	DD26	LD HX, D	DD62
LD LX, dd	DD2E	LD HX, E	DD63
LD B, HX	DD44	LD HX, LX	DD65
LD B, LX	DD45	LD HX, A	DD67
LD C, HX	DD4C	LD LX, B	DD68
LD C, LX	DD4D	LD LX, C	DD69
LD D, HX	DD54	LD LX, D	DD6A
LD D, LX	DD55	LD LX, E	DD6B
LD E, HX	DD5C	LD LX, HX	DD6C
LD E, LX	DD5D	LD LX, A	DD6F
LD HX, B	DD60	LD A, HX	DD7C
LD HX, C	DD61	LD A, LX	DD7D

8-bit Arithmetische und Logische Befehle

	B	C	D	E	H	L	(HL)	A	n	(IX+d)	(IY+d)	S	Z	H	P/V	N	C
ADD .	80	81	82	83	84	85	86	87	C6XX	DD86XX	FD86XX	*	*	*	*	0	*
ADC .	88	89	8A	8B	8C	8D	8E	8F	CEXX	DD8EXX	FD8EXX	*	*	*	*	0	*
SUB .	90	91	92	93	94	95	96	97	D6XX	DD96XX	FD96XX	*	*	*	*	1	*
SBC .	98	99	9A	9B	9C	9D	9E	9F	DEXX	DD9EXX	FD9EXX	*	*	*	*	1	*
AND .	A0	A1	A2	A3	A4	A5	A6	A7	E6XX	DDA6XX	FDA6XX	*	*	1	*	0	0
XOR .	A8	A9	AA	AB	AC	AD	AE	AF	EEXX	DDAEXX	FDAEXX	*	*	1	*	0	0
OR .	B0	B1	B2	B3	B4	B5	B6	B7	F6XX	DDB6XX	FDB6XX	*	*	1	*	0	0
CP .	B8	B9	BA	BB	BC	BD	BE	BF	FEXX	DDBEXX	FDBEXX	*	*	*	*	1	*
INC .	04	0C	14	1C	24	2C	34	3C		DD34XX	FD34XX	*	*	*	*	0	–
DEC .	05	0D	15	1D	25	2D	35	3D		DD35XX	FD35XX	*	*	*	*	1	–

		S	Z	H	P/V	N	C	
DAA	27	*	*	*	*	–	*	BCD-Korrektur im Akku
CPL	2F	–	–	1	–	1	–	Komplementiere Akku (1er-Komplement)
NEG	ED44	*	*	*	*	1	*	Komplementiere Akku (2er-Komplement)

16-bit Arithmetische und Logische Befehle

	BC	DE	HL	SP	IX	IY	S	Z	H	P/V	N	C
INC .	03	13	23	33	DD23	FD23	–	–	–	–	–	–
DEC .	0B	1B	2B	3B	DD2B	FD2B	–	–	–	–	–	–
ADD HL,.	09	19	29	39			–	–	*	–	0	*
ADC HL,.	ED4A	ED5A	ED6A	ED7A			*	*	*	*	0	*
SBC HL,.	ED42	ED52	ED62	ED72			*	*	*	*	1	*
ADD IX,.	DD09	DD19		DD39	DD29		–	–	*	–	0	*
ADD IY,.	FD09	FD19		FD39		FD29	–	–	*	–	0	*

Rotations- und Schiebebefehle

	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)
RR	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E	CB1F	DDCBXX1E	FDCBXX1E
RL	CB10	CB11	CB12	CB13	CB14	CB15	CB16	CB17	DDCBXX16	FDCBXX16
RRC	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBXX0E	FDCBXX0E
RLC	CB00	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBXX06	FDCBXX06
SRA	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E	CB2F	DDCBXX2E	FDCBXX2E
SLA	CB20	CB21	CB22	CB23	CB24	CB25	CB26	CB27	DDCBXX26	FDCBXX26
SRL	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E	CB3F	DDCBXX3E	FDCBXX3E

	S	Z	H	P/V	N	C	
RR/RL	*	*	0	*	0	*	Rotiere Reg rechts/links durch Carry
RRC/RLC	*	*	0	*	0	*	Rotiere Reg rechts/links
SRA/SLA	*	*	0	*	0	*	Shift Reg rechts/links arithmetisch
SRL	*	*	0	*	0	*	Shift Reg rechts/links logisch

		S	Z	H	P/V	N	C	
RRCA	0F	–	–	0	–	0	*	Rotiere Akku rechts
RLCA	07	–	–	0	–	0	*	Rotiere Akku links
RRA	1F	–	–	0	–	0	*	Rotiere Akku rechts durch Carry
RLA	17	–	–	0	–	0	*	Rotiere Akku links durch Carry
RLD(HL)	ED6F	*	*	0	*	0	–	Rotiere Ziffer links zwischen Akku und (HL)
RRD(HL)	ED67	*	*	0	*	0	–	Rotiere Ziffer rechts zwischen Akku und (HL)

Einzelbitbefehle

	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)
BIT 0	CB40	CB41	CB42	CB43	CB44	CB45	CB46	CB47	DDCBXX46	FDCBXX46
BIT 1	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E	CB4F	DDCBXX4E	FDCBXX4E
BIT 2	CB50	CB51	CB52	CB53	CB54	CB55	CB56	CB57	DDCBXX56	FDCBXX56
BIT 3	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E	CB5F	DDCBXX5E	FDCBXX5E
BIT 4	CB60	CB61	CB62	CB63	CB64	CB65	CB66	CB67	DDCBXX66	FDCBXX66
BIT 5	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E	CB6F	DDCBXX6E	FDCBXX6E
BIT 6	CB70	CB71	CB72	CB73	CB74	CB75	CB76	CB77	DDCBXX76	FDCBXX76
BIT 7	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E	CB7F	DDCBXX7E	FDCBXX7E
RES 0	CB80	CB81	CB82	CB83	CB84	CB85	CB86	CB87	DDCBXX86	FDCBXX86
RES 1	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E	CB8F	DDCBXX8E	FDCBXX8E
RES 2	CB90	CB91	CB92	CB93	CB94	CB95	CB96	CB97	DDCBXX96	FDCBXX96
RES 3	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E	CB9F	DDCBXX9E	FDCBXX9E
RES 4	CBA0	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6	CBA7	DDCBXXA6	FDCBXXA6
RES 5	CBA8	CBA9	CBAA	CBAB	CBAC	CBAD	CBAE	CBAF	DDCBXXAE	FDCBXXAE
RES 6	CBB0	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6	CBB7	DDCBXXB6	FDCBXXB6
RES 7	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE	CBBF	DDCBXXBE	FDCBXXBE
SET 0	CBC0	CBC1	CBC2	CBC3	CBC4	CBC5	CBC6	CBC7	DDCBXXC6	FDCBXXC6
SET 1	CBC8	CBC9	CBCA	CBCB	CBCC	CBCD	CBCE	CBCF	DDCBXXCE	FDCBXXCE
SET 2	CBD0	CBD1	CBD2	CBD3	CBD4	CBD5	CBD6	CBD7	DDCBXXD6	FDCBXXD6
SET 3	CBD8	CBD9	CBDA	CBDB	CBDC	CBDD	CBDE	CBDF	DDCBXXDE	FDCBXXDE
SET 4	CBE0	CBE1	CBE2	CBE3	CBE4	CBE5	CBE6	CBE7	DDCBXXE6	FDCBXXE6
SET 5	CBE8	CBE9	CBEA	CBEB	CBEC	CBED	CBEE	CBEF	DDCBXXEE	FDCBXXEE
SET 6	CBF0	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6	CBF7	DDCBXXF6	FDCBXXF6
SET 7	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE	CBFF	DDCBXXFE	FDCBXXFE

Flagbeeinfussung:

	S	Z	H	P/V	N	C
BIT	?	*	1	?	0	–
SET	–	–	–	–	–	–
RES	–	–	–	–	–	–

Flag-Register

Bit	7	6	5	4	3	2	1	0
	S	Z	X	H	X	P/V	N	C

	gesetzt	nicht gesetzt	wird gesetzt bei
C Carry-Flag	C	NC	Uebertrag von Bit 7
N Add-/Subtract-Flag			Subtraktionen
P/V Parity-/Overflow-Flag	PE	PO	gerader Paritaet
H Half-Carry-Flag			Uebertrag von Bit 3
Z Zero-Flag	Z	NZ	Ergebnis 0
S Sign-Flag	M	P	neg. Ergebnis
X nicht verwendet			

Beeinflussung:
1 gesetzt
0 zurueckgesetzt
* abhaengig vom Ergebnis einer Operation
– nicht beeinflusst
? unbestimmt

Mikro-Poster: Z80-Befehlssatz

Kurze Maschinenprogramme lassen sich auch ohne Assembler schnell und einfach mit den hexadezimalen Operationscodes schreiben. Heinz-Peter Curdts hat sich die Mühe gemacht, die Z80-Maschinenbefehle übersichtlich zusammenzustellen.

8-bit-Ladebefehle

	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(nn)	n
LD A,..	7F	78	79	7A	7B	7C	7D	7E	0A	1A	3AXXXX	3EXX
LD B,..	47	40	41	42	43	44	45	46				06XX
LD C,..	4F	48	49	4A	4B	4C	4D	4E				0EXX
LD D,..	57	50	51	52	53	54	55	56				16XX
LD E,..	5F	58	59	5A	5B	5C	5D	5E				1EXX
LD H,..	67	60	61	62	63	64	65	66				26XX
LD L,..	6F	68	69	6A	6B	6C	6D	6E				2EXX
LD (HL),..	77	70	71	72	73	74	75					36XX
LD (BC),..	02											
LD (DE),..	12											
LD (nn),..	32XXXX											

	A	B	C	D	E	H	L
LD ..,(IX+d)	DD7EXX	DD46XX	DD4EXX	DD56XX	DD5EXX	DD66XX	DD6EXX
LD ..,(IY+d)	FD7EXX	FD46XX	FD4EXX	FD56XX	FD5EXX	FD66XX	FD6EXX
LD (IX+d),..	DD77XX	DD70XX	DD71XX	DD72XX	DD73XX	DD74XX	DD75XX
LD (IY+d),..	FD77XX	FD70XX	FD71XX	FD72XX	FD73XX	FD74XX	FD75XX

LD (IX+d),n DD36XXXX   LD (IY+d),n FD36XXXX

		S	Z	H	P/V	N	C
LD A,I	ED57	*	*	0	*	0	-
LD A,R	ED5F	*	*	0	*	0	-
LD I,A	ED47	-	-	-	-	-	-
LD R,A	ED4F	-	-	-	-	-	-

16-bit-Ladebefehle

	BC	DE	HL	SP	IX	IY
LD ...,nn	01XXXX	11XXXX	21XXXX	31XXXX	DD21XXXX	FD21XXXX
LD ...,(nn)	ED4BXXXX	ED5BXXXX	2AXXXX	ED7BXXXX	DD2AXXXX	FD2AXXXX
LD (nn),...	ED43XXXX	ED53XXXX	22XXXX	ED73XXXX	DD22XXXX	FD22XXXX
LD SP			F9		DDF9	FDF9

	BC	DE	HL	AF	IX	IY
PUSH ..	C5	D5	E5	F5	DDE5	FDE5
POP ..	C1	D1	E1	F1	DDE1	FDE1

EX (SP),HL	E3	EX DE,HL	EB
EX (SP),IX	DDE3	EX AF,AF'	08
EX (SP),IY	FDE3	EXX	D9 BC-BC' DE-DE' HL-HL'

Blocktransfer- und Suchbefehle

		S	Z	H	P/V	N	C	
LDI	EDA0	-	-	0	*	0	-	LD(DE),(HL):INC HL:INC DE:DEC BC
LDIR	EDB0	-	-	0	0	0	-	wie LDI,wiederholen bis BC=0
LDD	EDA8	-	-	0	*	0	-	LD(DE),(HL):DEC HL:DEC DE:DEC BC
LDDR	EDB8	-	-	0	0	0	-	wie LDD,wiederholen bis BC=0
CPI	EDA1	*	*	*	*	1	-	LD A,(HL):INC HL:DEC BC
CPIR	EDB1	*	*	*	*	1	-	wie CPI,wiederholen bis BC=0 oder Zeichen gefunden
CPD	EDA9	*	*	*	*	1	-	LD A,(HL):DEC HL:DEC BC
CPDR	EDB9	*	*	*	*	1	-	wie CPD,wiederholen bis BC=0 oder Zeichen gefunden

Sprungbefehle

	Z	NZ	C	NC	PE	PO	M	P
JP	CAXXXX	C2XXXX	DAXXXX	D2XXXX	EAXXXX	E2XXXX	FAXXXX	F2XXXX
CALL	CCXXXX	C4XXXX	DCXXXX	D4XXXX	ECXXXX	EAXXXX	FCXXXX	F4XXXX
RET	C8	C0	D8	D0	E8	E0	F8	F0
JR	28XX	20XX	38XX	30XX				

	unbedingt	(HL)	(IX)	(IY)
JP	C3XXXX	E9	DDE9	FDE9
CALL	CDXXXX			
RET	C9			
JR	18XX			

RST	00	08	10	18	20	28	30	38
	C7	CF	D7	DF	E7	EF	F7	FF

DJNZ	10XX	DEC B:JR NZ,d
RETI	ED4D	zurueck vom Interrupt
RETN	ED45	zurueck vom nicht maskierbaren Interrupt

CPU-Steuerbefehle

		S	Z	H	P/V	N	C	
NOP	00	-	-	-	-	-	-	Leerbefehl
HALT	76	-	-	-	-	-	-	
CCF	3F	-	-	*	-	0	*	Komplementiere Carry-Flag
SCF	37	-	-	0	-	0	1	Setze Carry-Flag
EI	FB	-	-	-	-	-	-	Interrupts freigeben
DI	F3	-	-	-	-	-	-	Interrupts sperren
IM 0	ED46	-	-	-	-	-	-	Interrupt-Modus 0
IM 1	ED56	-	-	-	-	-	-	Interrupt-Modus 1
IM 2	ED5E	-	-	-	-	-	-	Interrupt-Modus 2

Ein-/Ausgabebefehle

	A	B	C	D	E	H	L	S	Z	H	P/V	N	C
IN ..,(C)	ED78	ED40	ED48	ED50	ED58	ED60	ED68	*	*	0	*	0	-
OUT (C),..	ED79	ED41	ED49	ED51	ED59	ED61	ED69	-	-	-	-	-	-

		S	Z	H	P/V	N	C	
IN A,n	DBXX	-	-	-	-	-	-	
OUT n,A	D3XX	-	-	-	-	-	-	
INI	EDA2	?	*	?	?	1	-	IN (HL),(C):INC HL:DEC B
INIR	EDB2	?	1	?	?	1	-	wie INI,wiederholen solange B()0
IND	EDAA	?	*	?	?	1	-	IN (HL),(C):DEC HL:DEC B
INDR	EDBA	?	1	?	?	1	-	wie IND,wiederholen solange B()0
OUTI	EDA3	?	*	?	?	1	-	OUT (C),(HL):INC HL:DEC B
OTIR	EDB3	?	1	?	?	1	-	wie OUTI,wiederholen solange B()0
OUTD	EDAB	?	*	?	?	1	-	OUT (C),(HL):DEC HL:DEC B
OTDR	EDBB	?	1	?	?	1	-	wie OUTD,wiederholen solange B()0

Ein Beitrag vom Harald Trapp

Anmerkungen und Überlegungen zur Hardware-Erweiterung des Genie, TRS-80
Erfahrungsbericht **Druckerpuffer "MACH 3" der Fa. ROOS Electronic**

Auf der Suche nach einer preisgünstigen und leistungsfähigen V.24 Schnittstelle möchte ich an dieser Stelle kurz auf den Leistungsumfang der V.24 (2-fach) Schnittstelle der Fa. RB electronic einghen - ein praktischer Erfahrungsbericht kann an dieser Stelle nicht erfolgen, da ich die Hard- u. Software nicht in der Praxis erproben konnte (ich füge das Datenblatt o.a. Firma bei):

- **Terminalbetrieb** (TRS/VG als intelligentes Terminal)
- **softw. Einstellung** (Baudrate, bits p. character, parity usw.)
- **Log-Datei** (Senden und Empfangen von ASCII u. HEX Dat.)
- **Treiber f. Drucker** (serielle Drucker mit XON/XOFF und CTS-Handshaking)

Zum Vergleich zur Software des Apricot PC (8086, IBM-kompatibel 8088) stellt das Kommunikationsprogramm der Fa. RB electronic eine gute und preiswerte Aletrnative dar.

Die Listenpreise liegen wie folgt:
V.24 (Bausatz) 249.-
V.24 (fertig) 299.- (jeweils Hard/Software)

Über Staffelpreise könnte ich folgende Preise erzielen:
5 bis 9 Bestellungen 215.-
10 bis 24 Bestellungen 195.-
25 bis 170.-

Auf eine preiswertere Alternative gehe ich am Schluß dieses Aufsatzes ein.

Eine weitere interessante Einrichtung sind Druckerpuffer, - Nachbauanleitungen sind sehr rar - hier ein Bericht zu einem fertigen Produkt:

Druckerpuffer MACH 3 mit 48 Kb (Fa. ROOS electronic)

Das Gerät wird geliefert in einem dunkelgrauen TEKO-Gehäuse, Buchsen für externes Steckernetzteil und Centronics-Stecker (Eingang), ca. 50 cm Kabel mit Stecker (Ausgang), Ein-Ausschalter (auch zum Daten löschen) und LED (on/off).

In dem Gehäuse findet eine doppelseitige gedruckte Schaltung im Europa-Kartenformat (10x16 cm) sowie ein auf ein kleines Kühlblech befestigter Spannungsregler für 5 Volt Platz.

Das mitgelieferte Steckernetzteil bietet einstellbare Spannungen von 3 Volt, 4.5 Volt, 7.5 Volt und 12 Volt sowie ein Polaritätsschalter (für den Puffer wird das Netzteil auf 9 Volt eingestellt).

Die Funktion ist ebenso simpel wie beeindruckend:
Für den Ausdruck eines Listings benötigt mein (lahmer) Drucker "microline 80" bei ca. 10 Seiten 12 Minuten (Drucker mit 80 Zeichen/sec.) - mit dem Druckerpuffer nur ca. 2 Minuten. All' dijenigen, die einen ebenso langsamen Drucker ihr Eigentum nennen wissen ein Liedchen über nervenaufreibendes Warten zu singen.

Das Gerät kann mit einfachen Mitteln nicht nachgebaut werden (doppelseitige Platine, eigener Prozessor mit Clock und Betriebsprogramm im Eprom).

Der Puffer ist lieferbar mit parallelem Ein-/Ausgang oder mit seriellem Eingang und parallelem Ausgang.

Listenpreise: (parallel/parallel)

16 Kb	389.-
32 Kb	455.-
48 Kb	499.-

Wenn sich mindestens fünf Interessenten finden würden, so könnte ich folgende Preise erzielen:

16 Kb	290.-
32 Kb	335.-
48 Kb	375.-

Mittlerweile liefert die Fa. ROOS electronic ein neues Modell bei dem sich die Seitenlange, Seitenvorschub einstellen lassen (Stop des Seitenvorschubs z.B. für Einzelblatt) - weiterhin lassen sich eine beliebige Anzahl Kopien vom Speicher des Puffers ziehen (ohne den Rechner zu belasten). In der 48 Kb Ausführung liegt der Listenpreis bei 519 .-

Weiterhin im Lieferprogramm der ROOS electronic sind EPROM-Burner und EPROM-Datenbanken (siehe auch beigelegtes Originalinfo):

EPROM-Burner:
zum Komplettsystem gehören: Burner, Software, Nullkraft-Sockel (28 pol.)
Listenpreis: 209.-
Minimaler Preis (bei 5 Stück): 170.-

EPROM-Löscher:
für 4 Eproms, Listenpreis: 120.-
Minimaler Preis (bei 5 St) 100.-

An dieser Stelle möchte ich darauf hinweisen, daß ich weder mit dem Burner noch mit dem V-Löscher Erfahrungen gemacht habe.

Preisgünstigere Alternative (Diskussionsgegenstand):

Eine preisgünstigere Alternative ist es wenn sich ein Mitglied Hard/Software zulegt (z.B. Anschaffung durch den CLUB80, Hard/Software sind somit Eigentum des Clubs).
Unterlagen können über das vom Club gekaufte Produkt nur gegen einen Betrag bezogen werden - allerdings könnte es hier rechtliche Schwierigkeiten geben ?!?!

Andere Möglichkeit:
Die Anschaffung z.B. einer V.24 Schnittstelle wird von 10 Club-Mitgliedern befürwortet, daher wird die Anschaffung kostenmäßig unter die 10 Mitglieder aufgeteilt, Bestellung eines Produktes, Versenden der Unterlagen zu den Interessenten - Möglicher Aufteilungsschlüssel:

60% der Kosten für den Besteller
40% der Kosten werden unter den 9 Interessenten aufgeteilt.

Auch hier ist die rechtliche Grundlage fraglich - Probleme sehe ich auch beim Hardware-Nachbau. - Ein Platinen-Layout, nach dem man die Platine vervielfältigen kann, liegt sicher nicht bei - bleibt also der Einzelaufbau über z.B. Fädeln.

Vielleicht ist die bessere Alternative doch eine Sammelbestellung.

Nachtrag zum Telefon-Modem (Modem Sonderheft):

Der komplette Bausatz (mit Platine, 2 Lautsprechern) ist bei der FA:
r+r electronic
Breslauer Str. 29
6900 Heidelberg
Tel. 06221/781500

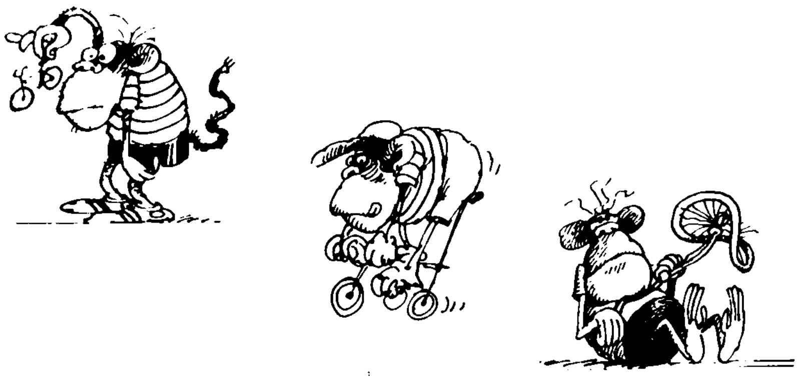
zum Bezugspreis von 37.50 DM erhältlich.

vom Harald Trapp

Anfrage:

Wer hat eine Bauanleitung für double density ?

Wer hat Erfahrung mit 8" Laufwerken in Kombination mit 5¼" im Anschluß an Genie ?



Es sagte
. . . . ein vorsichtiger Personalleiter zu einem Bewerber: »Ich kann Ihnen heute noch nichts Endgültiges sagen. Auf den Computer haben Sie jedenfalls einen guten Eindruck gemacht!«

RB - Elektronik -
Vertrieb GmbH
Bourauelerstr. 13
5208 Eitorf Postfach 113

Tel. 02243/3041
Telex 8869790 rbel d

PREISLISTE 4/84

Hochaufloesende Grafik

HRG1B fertig aufgebaut	DM 379,-
HRG1B Bausatz	Auf Anfrage
HRG1B Platine, Beschreibung u. Software (Cass.)	DM 100,-
Basgr/cmd Treiberprogramm auf Disk	DM 8,-
Supertreiberprogramm Version A 16 KByte Cass.	DM 98,-
" " Version B 48 KByte Disk/Cass	DM 128,-
Spoolerprogramm zur HRG1B, 12KByte Druckerspooler	DM 39,-

Expander

EXP1 Expanderplatine	fertig aufgebaut	DM 449,-
EXP1	Bausatz	DM 339,-

Doubler

DBL1 double-density Controller	fertig aufgebaut	DM 275,-
DBL1	Bausatz	DM 189,-

Kleinschriftmodule

KLTRS Kleinschrift fuer TRS-80 Model 1	DM 78,-
KLVG Kleinschrift fuer Video Genie 1 u. 2	DM 128,-

Serielle Schnittstelle - V.24

2*V.24	2-fach serielle Schnittstelle fertig	DM 299,-
2*V.24	Bausatz	DM 249,-

Speichererweiterung

MEM48 Speichererweiterung auf 48K, Einbausatz	DM 178,-

Diskettenlaufwerke und Zubehoer

Laufwerke slim-line TEAC 40 Spur SS DD	DM 639,-
TEAC 40 Spur DS DD	DM 785,-
TEAC 80 Spur SS DD	DM 745,-
TEAC 80 Spur DS DD	DM 955,-
Floppykabel fuer 2 Laufwerke	DM 75,-
Floppygehaeuse fuer 1 - 2 Slimlinelaufwerke (inkl. Zweifachnetzteil)	DM 200,-
Disketten Verbatim 5z MD 200-1 10 Stueck	DM 62,-
100 Stueck	DM 560,-
1000 Stueck	DM 5200,-

Lieferung: Nachnahme Verandkostenpauschale DM 8,50
Preisaenderungen vorbehalten.
Alle Preise inkl. Mehrwertsteuer.

rb-info dbl1
27.3.84 db

```
*****************************************************
*       DOUBLE-DENSITY CONTROLLER   DBL1            *
*****************************************************
```

Mit einem double-density Controller erhoehen Sie die Speicherkapazitaet Ihrer Diskettenlaufwerke auf den ca. 1,8-fachen Wert. Damit ist ein komfortablerer Diskettenbetrieb ermoeglicht, denn die ca. 100K die bei 40-spur einseitigen Laufwerken in einfacher Schreibdichte zur Verfuegung stehen, reichen bei manchen Betriebssystemen, wie zum Beispiel dem NEWDOS/80, gerade aus das komplette System aufzunehmen. DBL1 ist als Bausatz und als Fertiggeraet lieferbar und geeignet fuer TRS-80 M1 und VIDEO GENIE 1 und 2.

Technische Kurzbeschreibung:
Der DBL1 Doubler ist kompatibel zum PERCOM1-Doubler. Er wird auch z.B. als solcher bei den NEWDOS/80 Pdrive-Parametern eingetragen. Ihre Laufwerke muessen natuerlich fuer den Betrieb mit doppelter Schreibdichte geeignet sein; was aber bei fast allen handelsueblichen Geraeten der Fall ist. Der Controller verfuegt ueber einen Datenseparator, der eine wesentlich hoehere Datensicherheit ermoeglicht. Dieser ist auch bei einfacher Schreibdichte in Funktion, so dass sich auch in diesem Modus ein sicherer Betrieb ergibt.

Auf- und Einbau:
Der Selbstaufbau ist anhand der mitgelieferten Beschreibung sehr leicht durchzufuehren. Zum Einbau wird der Controllerchip des Expanders herausgenommen, an seine Stelle wird die EXP1-Platine eingesetzt und der "alte" Controller wird in einen dafuer vorgesehenen Sockel auf der DBL1 gesteckt.

Mitgelieferte Beschreibungen:
Einbauanleitung
Bestueckungsplan
Stueckliste

rb-info mem48
27.03.84 db

```
*****************************************************
*     48K-SPEICHERERWEITERUNG       MEM48           *
*****************************************************
```

Mit der MEM48-Speicherweiterung erweitern Sie den Speicher Ihres TRS-80 M1 oder VIDEO GENIE um 32K-Bytes auf insgesamt 48K. Erst damit ist ein sinnvoller Betrieb unter Diskettenbetriebsystemen moeglich.

Technische Kurzbeschreibung und Einbau:
Der Einbau der Erweitererung in Ihren Computer erfolgt durch dden Austausch der im Geraet vorhandenen 16K-Rams durch neue 64K-Chips. Es koennen nur Rams ganz bestimmter Typen dazu genommen werden. Wenn Sie ueber Erfahrung im Selbstbau verfuegen, duerfte Ihenen der Einbau ins Geraet nicht schwerfallen; andernfalls fuehren wir den Einbau fuer einen Betrag von DM 80 durch.

Mitgelieferte Beschreibungen:
Einbauanleitung fuer TRS-80 und VIDEO GENIE und Aenderungen fuer bestimmte Baureihen dieser Geraete

rb-info expl
26.3.84 db

EXPANDER-PLATINE EXP1

Irgendwann steigt jeder Besitzer eines TRS-80 oder eines VIDEO GENIE auf Diskettenlaufwerke um, weil ihm die Arbeit mit den Kassettenlaufwerken zur Qual geworden ist. Oder man braucht eine Druckerschnittstelle, die den neuerworbenen Drucker ansteuern soll. Unsere Expanderplatine EXP1 bietet Ihnen eine sehr preiswerte Moeglichkeit die benoetigte Hardware vorzusehen.

Technische Kurzbeschreibung:
Das Floppyinterface kann vier 5 1/4 Zoll Laufwerke steuern. Es werden sowohl ein- als auch doppelseitige Laufwerke unterstuetzt. Das Interface ist voll funktionskompatibel zu den Standart-Expandern. Die Centronics-Parallelschnittstelle zur Ansteuerung eines Druckers arbeitet sowohl mit memory-mapped als auch mit Portausgabe Daher kann speziell fuer VIDEO GENIE bzw. TRS-80 geschriebene Software, die in diesem Punkt inkompatibel waren gefahren werden. Zur EXP1 gehoert nicht die Speichererweiterung auf 48K-Bytes; dafuer ist unser Speichererweiterungssatz MEM48 vorgesehen.

Auf- und Einbau:
Der Selbstaufbau ist einfach und problemlos durchzufuehren. Zum Lieferumfang gehoeren die durchkontaktierte Platine mit Loetstopplack, der Bestueckungsplan, die Anleitung und alle Bauteile. Die Expanderplatine kann im Tastaturgehaeuse selbst untergebracht werden. Die Groesse der Platine betraegt nur 150 X 100 mm. Die Anschlusszungen werden dann durch eine seitlich im Gehaeuse eingeschnittene Oeffnung herausgefuehrt. Zum Einbau muessen nur Anschlussdraehte nach Plan angeloetet werden. Natuerlich gibt es den EXP1 auch als Fertiggeraet. Wenn Sie den Einbau nicht selbst vornehmen koennen, so bauen wir den Expander gegen einen Betrag von DM 80 auch bei uns in Ihr Geraet ein.

Mitgelieferte Beschreibungen:
Einbauanleitung fuer VG und TRS-80 M1
Anschlussplan
Schaltplan
Bestueckungsplan

Es sagte . . . ein verblüffter Meteorologe zu seinem Wetterfrosch-Kollegen (und wies dabei auf das Elektronengehirn):
»Er konstatiert, gegen Abend sei mit zunehmender Dunkelheit zu rechnen!»

rb-info v.24
23.3.84 db

2-FACH V.24 SCHNITTSTELLE

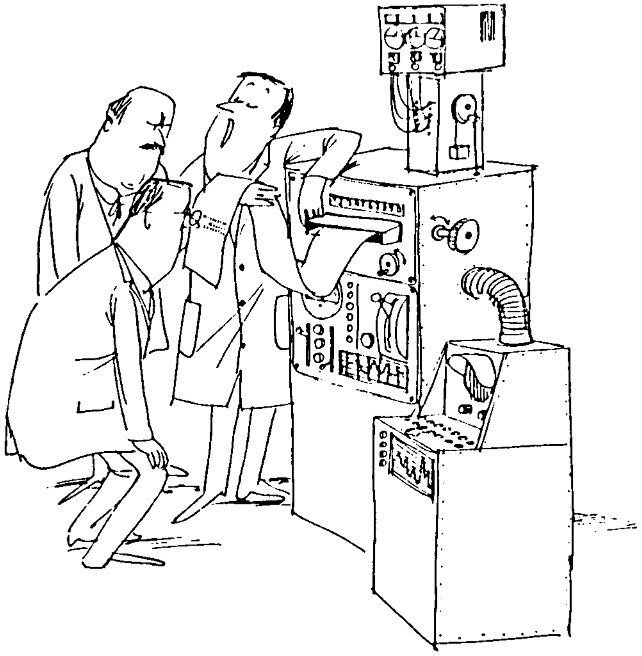
Lange Zeit waren serielle Schnittstellen fuer Besitzer eines TRS-80 oder eines Video Genie nur sehr schwer oder kaum erhaeltlich. Aus diesem Grunde bieten wir Ihnen jetzt eine 2-fach (!) V.24-Schnittstelle in Bausatzform oder als Fertiggeraet an. Mit diesem Zusatzgeraet ist es moeglich serielle Drucker anzusteuern, Daten mit anderen Microcomputersystemen auszutauschen - auch wenn dies vorher wegen inkompatibler Diskettenformate unmoeglich war - oder ein Modem zu betreiben. Erinnern Sie sich an die zahlreichen Berichte in Fachzeitsschriften die in letzter Zeit ueber neue Entwicklungen auf dem Gebiet der Datenuebertragung informierten. Die RB-V.24 ist in Bezug auf die verwendeten Portadressen nicht kompatibel zu den Originalschnittstellen. Das Umstellen von Programmen sollte allerdings ohne Schwierigkeiten moeglich sein.

Technische Kurzbeschreibung:
Durch den Einsatz eines Z80-DART verfuegt die RB-V.24 ueber zwei praktisch vollstaendig unabhaengig voneinander arbeitende serielle Kanaele. Die Einstellung der Baudrate erfolgt softwaremaessig, wobei alle Standard-Baudraten von 75 bis 19200 Baud eingestellt werden koennen. Aber auch "krumme" Uebertragungsgeschwinigkeiten koennen im unteren Bereich gewaehlt werden. Die vielfaeltigen Moeglichkeiten des DART erlauben es, eine ganze Reihe von Uebertragungsparametern via Software festzulegen. Darunter fallen zum Beispiel die Anzahl der Stopbits (1, 1 1/2 oder 2), die Paritaetskontrolle, die Abfrage von Steuerleitungen (CTS/RTS u.ae.) oder auch die Anzahl der Bits pro Zeichen (5,6,7 u. 8 bits). Auf dieses Weise koennen Sie in jedem gebraeuchlichen Format senden und empfangen. Schliesslich stehen noch zwei CTC (Counter timer circuit) -Kanaele zur freien Verfuegung.

Der Einbau:
Mit der mitgelieferten ausfuehrlichen Einbauanleitung duerfte es Ihnen nicht schwerfallen, die Platine in das Geraet einzubauen. Die Karte selbst ist ca. 6 X 12 cm gross. Als Scnittstellenausgaenge dienen zwei normgerechte D-Steckbuchsen. Der Einbau fuer die in Japan hergestellten Tandy's ist gesondert beschrieben.

Mitgelieferte Software:
Zum Lieferumfang gehoeren ein universelles Schnittstellenprogramm namens "Multicom" und zwei Druckertreiber fuer Drucker mit serieller Schnittstelle.
Mit Hilfe von Multicom kann man Dateien in Hex- und Ascii-Format senden und empfangen und es gibt einen Terminalmodus mit dem Ihr Rechner zu einem Terminal umfuntioniert wird. Da der Empfang von Zeichen interruptgesteuert realisiert wurde, sind Uebertragungsgeschwindigkeiten bis zu 19200 Baud moeglich, obwohl die relativ langsame Hardware des TRS-80/VG - insbesondere seine Videoausgabe - dieses eigentlich nicht zulaesst. Multicom ist hauptsaechlich fuer den Modembetrieb gedacht, bei dem es ein sehr wertvolles Hilfsmittel darstellt.
Die beiden Druckertreiber heissen "Sprx" und "Sprcts" und unterscheiden sich in der Art des verwendeten Druckerprotokolls. Sprx realisiert ein Xon/Xoff Protokoll, bei dem der Drucker ein Ascii-Xoff (Ctrl-S) sendet, wenn sein Puffer voll zu werden droht und ein Xon (Ctrl-Q), wenn weitergesendet werden kann. Sprcts verwendet die CTS (Clear to send) -Steuerleitung zum Handshaking mit dem Drucker. Beide Treiber lassen sich auf die bereits erwaehnten Standart-Baudraten einstellen. Sie sind voll verschieblich und schuetzen sich nach der Initialisierung selbst, so dass eine einwandfreie Zusammenarbeit mit den meisten Programmen gewaehrleistet ist.

Mitgelieferte Beschreibungen:
Neben der Einbauanleitung und den Programmbeschreibungen ist auch eine kurze Einfuehrung in die Programmierung des DART und eine Aufstellung seiner Schreib- und Leseregister beigefuegt.

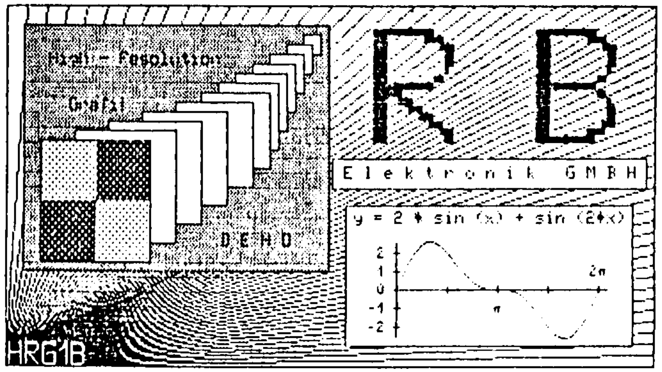


»Er sagt: Für dieses Problem müßt ihr das nächstgrößere Modell meiner Firma anschaffen.«

rb-info hrg1b
23.3.84 db

```
*********************************************
*       HOCHAUFLOESENDE GRAFIK   HRG1B      *
*********************************************
```

Ein Schwachpunkt des TRS-80 und des Video Genie ist seine unzureichende Grafikaufloesung mit 128 X 48 Punkten, mit der anspruchsvolle oder aussagekraeftige Grafiken kaum moeglich sind. Wir bieten daher seit ca. 2 Jahren eine hochaufloesende Grafik, die HRG1B, mit 384 X 192 Bildpunkten fuer diesee beiden Rechnertypen an, deren Aufloesungsvermoegen Sie aus dem Bildschirmausdruck ersehen koennen.



Unserem Wissen nach ist dieses Zusatzgeraet in der Zwischenzeit zur meistverkauften Grafikerweiterung fuer TRS-80 und VG ueberhaupt geworden. Eine Reihe von unabhaengigen Softwareherstellern hat Treiberprogramme und Applikationssoftware fuer die HRG1B vorgestellt.

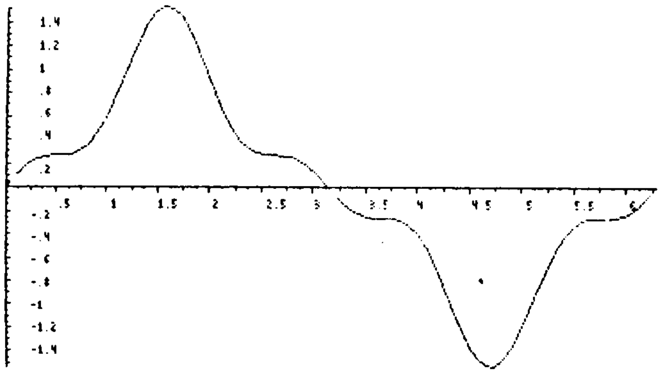
Technische Kurzbeschreibung:
Die HRG1B besitzt einen eigenen 12Kbyte grossen Bildwiederholspeicher, so dass der Ram-Speicher Ihres Geraetes nur von den zur Ansteuerung notwendigen Treiberprogrammen belegt ist. Die Ausgabe auf den Bildschirm erfolgt unabhaengig von der normalen Ascii- und Blockgrafikausgabe. Daher besteht die Moeglichkeit beide Darstellungen zu mischen. Wird kein hochaufloesende Grafik benoetigt, so kann die HRG1B vollkommen weggeschaltet werden. Die bereits erwaehnten 12Kbyte Ram koennen dann fuer ganz andere Zwecke verwendet werden, wie es z.B. in einem ebenfalls von uns angebotenen Spoolerprogramm zur zeitsparenden Druckerausgabe geschieht.

Mitgelieferte Software:
Zum Lieferumfang gehoert ein Treiberprogramm, welches den Befehlssatz des Basic-Interpreters um 8 zusaetzliche Kommandos erweitert. Es koennen wie bei der Blockgrafik Punkte gesetzt, rueckgesetzt und abgefragt werden, aber auch ein schnelles Zeichnen von Linien ist vorgesehen. Das Treiberprogramm gibt es in Kassetten- (GRL2) und Diskettenversion (BASGR). Als einfuehrende Beispiele sind drei kleine Demoprogramme beigefuegt.

Optionale Software:
Zusaetzlich bieten wir gegen Aufpreis eine stark erweiterte Version des Treiberprogramms an, den "Supertreiber", der in Bezug auf Komfort und Leistungsfaehigkeit kaum Wuensche offenlaesst. Grafiken konnen auf einen Drucker ausgegeben werden, auf Diskette bzw. Band abgepeichert, wieder eingelesen oder mit anderen Grafiken gemischt werden, und Sie koennen sogar einen eigenen Zeichensatz mit allen nur moeglichen Sonderzeichen erstellen, der dann anstellte der normalen Zeichen erscheint. Es existieren Funktionen zum Zeichnen beliebiger Vielecke - Angabe von Mittelpunkt, Radius, Anzahl der Ecken und Verschiebung der Drehachse - , zum Zeichen von Kreisen und von Linien auch mit relativer Angabe der Ursprungskoordinaten. Und das ist noch laengst nicht alles !

Zum Einbau:
Mit etwas Erfahrung koennen Sie die HRG1B ohne Schwierigkeiten selbst einbauen oder gegen einen Betrag von DM 80 den Einbau auch von uns vornehmen lassen. Die ca. 15 X 10 cm grosse Platine wird im Tastaturgehaeuse untergebracht.

Mitgelieferte Beschreibungen:
Einbauanleitung fuer TRS-80 und Video Genie
Programmbeschreibungen BASGR bzw. GRL2
Anschluss- und Bestueckungsplaene
Programmierhinweise



Allgemeine Ueberlegungen zur DFUE (Datenfernuebertragung)
Erfahrungsbericht mit dem Akustikkoppler CX-21 von Epson

Jeder Mikrocomputeranwender wird sich irgendwann bereits einmal mit der Frage von Datenfernuebertragung befasst haben - wenn man sich einmal in das Thema eingelesen hat kommt man so schnell nicht wieder los davon (und sobald man die ersten Erfahrungen gemacht hat mit Sicherheit nicht).

Es ist also eine durchaus lohnende und interessante Einrichtung Daten ueber weite Strecken per Telefon zu schicken. - Ein anwaehlbarer Rechner innerhalb des Club80 gewiss eine Bereicherung.

Leider sind von gesetzlicher Seite starke Einschraenkungen gemacht worden, mal abgesehen von der finanziellen Seite.

Um es vorneweg zu nehmen:
Ein Kommunikationsservice der z.B. dem TEDAS (anwaehlbarer Zentralrechner des Franzis Verlag, spez. 'mc') kann von dem Club80 nicht realisiert werden.

Von der programmtechnischen Seite liesse sich sicherlich ein Datenbanksystem aufbauen in das nicht nur Informationen abgelegt (Nachrichten, schwarzes Brett), sondern auch Programme abgerufen werden koennen.

Das Problem liegt auf der Hardware-Seite. Es muss ein galvanisch gekoppeltes Modem von der Post angemietet werden, das im Gegensatz zu Akkustikkopplern von der Post installiert wird, waehrend man Akustikkoppler privat kauft.
Dieses Modem muss von ausserhalb anwaehlbar sein was einen andauernden Betrieb des Rechners erfordert (ohne aktive Datenendeinrichtung (DEE) ist ein anwaehlbares Modem sinnlos).
Das Problem liegt nun darin, dass an galvanisch gekoppelte Modem nur Datenendeinrichtungen (Zentralrechner) angeschlossen sein duerfen, die die FTZ-Zulassung besitzen. Das ist bei dem TRS80, Genie oder Komtek nicht der Fall. - Diese Moeglichkeit scheidet also aus.

Bleiben also Akustikkoppler. - Der Betrieb laeuft folgendermassen ab:

Anwaehlen des gewuenschten Teilnehmers (fest eingerichtete DEE z.B. TEDAS)

Traegerton wird hoerbar

Der betriebsbereite AK (halb duplex, answer mode), ~~angeschlossen~~ an ein Terminal oder Mikrocomputer (mit Terminalprogramm) angeschlossen, wird mit dem Handapparat akustisch gekoppelt (Handapparat in Gummimuffe einlegen)

Nach einigen Sekunden meldet der AK 'ready' und die Kommunikation beginnt.

oder:
wenn zwei Mikrocomputer-Anwender Daten austauschen wollen (auf beiden Seiten Akustikkoppler)

Anwaehlen des Teilnehmers, Mitteilen das der AK bereitgemacht werden muss.

Das ist ein Beispiel, was man so alles an Möglichkeiten hat. Zwischen den Seiten gehen leider immer einige Zeilen ab. Das vollständige „Telefonat" kann zusammen mit einem 2. Beispiel (Anruf beim WDR - Computerclub) bei der Kontaktadresse gegen Verrechnung der Kopierkosten bezogen werden.

Sie sind der 19606. Anrufer!

```
XXX XXX XX  XXX XXX
 X  X   X X X X X
 X  XXX X X X X XXX
 X  X   X X XXX   X
 X  XXX XX  X X XXX
```

mc-TEDAS 5.3

Ausg.stop=CTRL-S;weiter=CTRL-Q;abbr.=CTRL-X
Der 20 000. Anrufer dieser Nummer bekommt ein Franzis-Buch eigener Wahl kostenlos!!! Hinterlegen Sie dazu Ihre Adresse unter "Mitt.an Verlag".
end

1 Informationen
2 Briefkasten
3 mc-Inhaltsverz.
4 Softw.-Service-Kat.
5 Firmenadressen
6 Bauanl.-Bezugsqu.
7 Parameter aendern
8 Ende

Ihre Eingabe ...1

1 Inf.z.Modembetrieb
2 Franzis-Hausmitt.
3 Brancheninfos
4 Bericht.,Nachtraege
5 Unterhaltung
6 Biorhythmus
99 Hauptmenue
100 Ende

Ihre Eingabe ...1

TEDAS hat zwei Nummern

TEDAS erreichen Sie unter 089/59 64 22 und 089/59 84 23. Beide Computer enthalten die gleiche Information, da ihr Inhalt jeden Morgen "gemixt" wird.

Suchen nach Stichworten

Mit dem Befehl f 'Return' ist in den Zeitschriften-Inhaltsverzeichnissen sowie in den Firmen-Adressendateien eine Stichwortsuche moeglich. Als Stichworte koennen z.B. verwendet werden:

a) bei Zeitschrifteninhalten: Apple, TRS-80, VC-20, C-64, CP/M, CBM, TEDAS, Z80, 6502 usw.
b) bei Firmenadressen: Firmennamen (Apple, Commodore usw.); Postleitzahlen (z.B. 8000 fuer Muenchen); Ortsnamen; Strassennamen.

Bitte beachten Sie, dass die Stichworte genau so eingegeben werden muessen (mit Gross- und Kleinschreibung), wie gespeichert sind, sonst finden Sie das Gewuenschte nicht.

Die von manchen Computern produzierten Steuerzeichen entsprechen leider nicht der ASCII- bzw. ISO-7-Bit-Norm. Beispielsweise wird beim HX-20 bei "Cur[illegible] nach links" (Pfeiltaste) weder der Code fuer Backspace (BS) [illegible]h fuer Delete erzeugt. CTRL-H dagegen funktioniert einwandfrei. Aehnliche Probleme gibt es auch bei Commodore-Rechnern (hier entspricht zudem die Einteilung von Gross- und Kleinbuchstaben-Codes nicht der Norm).
Folgende Steuerzeichen sind bei TEDAS zugelassen:

Zeichen	hex	dez	
Backspace	08	8	1 Zeichen zurueck
Delete	7F	127	1 Zeichen zurueck
Return	0D	13	Neue Zeile (nach max. ca. 75 Zeichen!)
Line Feed	0A	10	Zeilenvorschub (optional)
CTRL-S	13	19	Haelt Ausgabe an
CTRL-X	18	24	Bricht Ausgabe ab
CTRL-Q	11	17	Laesst Ausgabe weiterlaufen

Mit CTRL-X koennen Sie auch die Menuetexte abbrechen, um Zeit zu sparen. Verwenden Sie bitte keine deutschen Umlaute, da nicht alle Rechner diese Zeichen implementiert haben!

PROTOKOLL-INFO

Stellen Sie Ihr Geraet auf 300 Baud, 2 Stopbits, keine Parity, 8 Datenbits ein. Bei Textuebertragung (Upload) sollte Ihr Programm nach jeder Zeile etwa 200 ms Pause einlegen, da nach Return LF angefuegt wird und die Texte bei TEDAS zeilenweise auf Floppy gespeichert werden.

IMPRESSUM

TEDAS ist ein kostenloser Dienst der mc-Redaktion fuer ihre Leser. Fehlerhaft uebertragene Mitteilungen und kommerzielle Werbeeintraege werden geloescht. Alle Briefkasten-Eintraege bleiben etwa drei bis sechs Tage gespeichert. Ein Rechtsanspruch auf Weiterleitung von Informationen besteht nicht. Fuer den Inhalt verantwortlich: Herwig Feichtiner, mc-Redaktion, Postf.370120, 8000 Muenchen 37, Tel. (werktags ab 15 Uhr) 089 5117 354. TEDAS erreichen Sie unter 089 596422 und 089 598423.
end

1 Inf.z.Modembetrieb
2 Franzis-Hausmitt.
3 Brancheninfos
4 Bericht.,Nachtraege
5 Unterhaltung
6 Biorhythmus
99 Hauptmenue
100 Ende

Ihre Eingabe ...2

Der mc-68000-Computer

Die Schaltung des mc-68000 wird in mc 11/1984 veroeffentlicht. Vorab die wichtigsten technischen Daten:

CPU 68000, 8 MHz
Bus VME-kompatibel, 16 Daten- und 24 Adressenleitungen, acht Slots (64pol.VG)
RAM 128 KByte auf der Platine, erweiterbar extern bis 16 MByte
ROM 32 KByte EPROM mit 3-Pass-Assembler und Monitorprogramm
Video 4 Ausgaenge: Sync, BAS, FBAS, RGB; 40/80 Zeichen pro Z., 24 Zeilen, Grafik 640 x 200 Punkte (sw), 320 x 200 Punkte bei acht Farben
Schnittstellen: Centronics, V.24, TTY/TTL seriell, User-Port, Kassetten-Recorder 3600 Baud, Tastaturanschl. parallel und seriell
Betriebssysteme: mit Floppy-Karte OS-68K, CP/M-68K, geplant: Versa-DOS, Unix
Besonderheiten: Nur Standard-Bauteile (kein PLA usw.!), schnelles RAM als frei programmierbarer Adressendecoder... ein SUPER-SYSTEM!

mc ab 1. Oktober 50 Pf teurer

Der Angerufene stellt seinen AK auf 'answer mode' der Anrufer auf 'originate mode' (= Vergabe von Sende u. Empfangskanal)

Die Kommunikation beginnt.

Der zuletzt beschriebene Vorgang ist der, der fuer 'normale' Mikrocomputer-Anwender interessant ist, dies DFUE setzt keine DEE mit FTZ-Zulassung voraus. - Gesetzlich vorgeschrieben ist alerdings eine FTZ-Nummer fuer den Akustikkoppler. Und diese Akutikkoppler sind nicht gerade eine kleine Belastung fuer die Haushaltskasse.

Selbsgebaute AK liegen ca. bei 60 bis 100 DM, man laeuft allerdings Gefahr, dass bei einer Aufdeckung eines derartigen AK nicht nur die gesammte Anlage eingezogen wird, sondern auch mit einer Anzeige zu rechnen ist.

Auf dem Markt befinden sich zur Zeit drei Akustikkoppler, die fuer den Club80 mit seinen Mitgliedern interessant sein koennten:

1. AKustikkoppler der Fa. Tandy

Preis ca. 380 DM
Ausstattung: Netzteil
FTZ-Zulassung: ja
Betrieb: orig. und ans.

leider z.Zt in ganz Deutschland nicht erhaeltlich, Lieferschwierigkeiten voraussichtlich bis Ende 1984

2. AKustikkoppler Teleterm

Preis ca. 550 DM
FTZ-Zulassung: ja
Ausstattung: das Geraet macht ausserlich einen schlechten Eindruck, billiges Kunststoffgehaeuse, keine Beschriftung
Ausfuehrung der Muffen: na ja

Ist eher etwas fuer Anwender des VC 20 (fuer den es auch entsprechende Software gibt)

3. Akustikkoppler von Epson CX-21

Preis (Liste) 750.- bei Vobis (656.-)
FTZ-Zulassung: ja
Ausstattung: Netzteil und eingebaute Akkus
schweres Geraet, solider Aufbau, Muffen passen sich sogar an moderne Flachhandapparate an

Der Akustikkoppler CX-21 von der Firma EPSON wird gut verpackt mit Netzteil, Akkusatz, engl. ausfuehrlichem Handbuch und deutscher Betriebserlaubnis geliefert.

Schnittstelle V.24, Schalter fuer on/off, answer, originate und test mode, Schalter fuer half/full duplex, LED's fuer Betrieb und Empfangsbereit.

Die Muffen (Gummi mit Schaumstoffpuffer, innen) lassen sich sowohl fuer die Sprech- als auch fuer die Hoerkapsel in der Neigung verstellen, ein normaler Handapparat hat ebenso wie die modernen Flachhandapparate einen guten Kontakt. Die Umweltgeraeusche werden zuverlaessig abgeschirmt, Trittschall und Husten haben keinen Einfluss auf DFUE.

In meiner Eigenschaft als Systemberater und Org.Programmierer fuer Individualsoftware im Kundenauftrag kann ich diesn Akustikkoppler nur bestens empfehlen, er wird von meinem Arbeitgeber fuer DFUE im Rahmen mit Buchmacherprogrammen (Wettannahmen, DFUE von der Annahmestelle zur Rennbahn), sowie zur Ferndiagnose und Softwarepflege eingesetzt und auch von dieser Stelle entsprechend befuehrwortet.

Leider kann ich bei diesem Produkt keinen guenstigeren Preis als den Vobis-Preis von DM 658.- erzielen.

Es bleibt noch ein anderes Problem - die V.24 Schnittstelle mit entsprechender Kommunikationssoftware. Die FA. RB electronic bietet fuer diesen Zweck sowohl eine Hardware, als auch Softwareloesung an (Terminalbetrieb, Log-Datei, Einstellen der Parameter usw.) diese Adaption wird von RB fuer DM 249.- (Bausatz) vertrieben.

Falls hier keine Alternative von anderer Seite besteht habe ich die Moeglichkeit ueber Staffelpreise guenstig diese Hard/Software einzukaufen.

Mit diesen Ausfuehrungen habe ich hoffentlich zu einer regen Diskussion unter den CLUB80 Mitgliedern aufgerufen - es waere schoen, wenn wir auch untereinander via Akustikkoppler in Verbindung treten koennten.- Schoen waere es auch, wenn man zu einer preiswerteren (und erlaubten) Alternative kommen koente.

Fuer alle Faelle habe ich meinem Bericht einen Auszug aus dem MODEM-Sonderheft beigelegt (fuer den Betrieb an einer V.24 Schnittstelle muessen allerdings noch Line Driver (z.B SN 75188) an den TTL Ausgabng geschaltet werden)

H.Trapp

Nachtrag:
Normale von der Post gemietete galvanisch gekoppelte Modem verursachen folgende Kosten:
- Anschlussgebuehr (einmalig) DM 80.-
- Grundgebuehr (Monatlich) DM 80.-

Bleibt noch anzumerken, dass der oben beschriebene Akustikkoppler nicht nur mit PC's wie 'apricot' in 8086-Technologie, sondern auch mit der firmeneigenen Grossrechenanlage (NOVA-System) zuverlaessig arbeitet.

mc 6/1984,S.85: Der Massepunkt des 1-Ohm-Widerstandes in Bild 2 muss an die Minusleitung des Akkus, nicht von U3 gelegt werden. Der NTC-Fueh von Siemens hat die Bestellnummer Q63084-M7502-X3 (geeicht). Der KS Kondensator ist bei Siemens (Bauteile-Service Fuerth) unter der Nummer B31531-B5563-F erhaeltlich.

Modem-Programme fuer Apple-II, MC-65, AIM-65, PC-100

Fuer Datex-P ist eine Umstellung von zwei auf ein Stopbit noetig.
Das laesst sich folgendermassen erreichen:
1. Apple-Kommunikationsprogramm aus mc 1/84:
   0359=08
   036D=4C 24 03
   0324=A5 FD 4A 09 80 30 46
2. Apple-Modemprogramm aus Modem-Sonderheft:
   09B2=08
   09C6=4C B4 0B (steht leider verkehrt in mc 7/1984!)
   0BB3=00 A5 FB 4A 84 F7 4C CA 09
3. Apple-Modempgm.auf Sammeldiskette 5 (ab hex 8400):
   85B2=08
   85C6=4C B4 87
   87B3=00 A5 FB 4A 84 F7 4C CA 85
4. MC-65-Modemprogramm aus mc 3/1984 und Modem-Sonderheft:
   032A=08
   0340=4A EA
5. Terminalpgm.f.AIM-65 und PC-100 aus dem Buch
   "Programmierpraxis mit dem 6502" (Franzis):
   02ED=08
   0309=4A EA

Nach den Aenderungen ist uebrigens ein Kontakt mit TEDAS nach wie vor moeglich.

Telefon-Modem
----
mc-Modem-Sonderheft: Die Schaltung der Bauanleitung arbeitet bei Brueckenlegung fuer TTL-Ein/Ausgaenge mit negativer (!) Logik an Ein- und Ausgang. D.h. ein Low-Signal entspricht log.1, ein High-Signal log. 0. Die Ruhelage entspricht stets log.1 = Low, also der hoeheren der beiden Frequenzen eines Kanals.

Sprites - aber wie?
----
mc 5/1984: Die Speicherstelle hex 0287 beinhaltet die Farbe unter dem Cursor, nicht das High-Byte des Video-RAM. Richtig muss es also hex 288 bzw. dez. 648 heissen.

Monitor im Video-Genie
----
mc 12/1983: Der Hersteller hat offenbar inzwischen mehrmals den ROM-Inhalt geaendert. Bei manchen Video-Genie-Versionen funktionieren daher die genannten Monitorfunktionen nur teilweise.

end

1 Inf.z.Modembetrieb
2 Franzis-Hausmitt.
3 Brancheninfos
4 Bericht.,Nachtraege
5 Unterhaltung
6 Biorhythmus
99 Hauptmenue
100 Ende

Ihre Eingabe ...5

Dumme Bemerkungen und andere Spaesse

Anzeige im Weser-Report:
Piranhas fressen Menschen. Schlagt zurueck: Esst mehr Fisch!

muessen in beiden Zahlen und Arten mit den Variablen der Variablenliste uebereinstimmen. Variabelliste besteht aus einer Nummer der Variabelnamen, beide Zeichenreihe der numerisch, durch Kommata getrennt. Das zusaetzliche Prompt hat jeden g tigen Zeichenreihenausdruck. Basic vorher prompt erscheinen, um Ihre Eingabe zu akzeptieren..."

Der Computer ist die logische Weiterentwicklung des Menschen:
Intelligenz ohne Moral! (John O.)

Was ist der Unterschied zwischen DUAL und BINAER?
Binaer baut keine Plattenspieler!

Munbright's Gesetze:
Jedes Programm ist veraltet, wenn es laeuft.
Jedes Programm kostet mehr und dauert laenger.
Wenn ein Program nuetzlich ist,muss es geaendert werden.
Wenn ein Programm unnuetz ist, muss es dokumentiert werden.
Jedes Programm ist ein bisschen laenger,als der Speicher es erlaubt.
Der Wert eines Programms ist umgekehrt proportional dem von ihm verbrauchten Papier.
Der unangenehmste Fehler wird erst entdeckt, wenn ein Programm mindestens sechs Monate im Einsatz ist.
Zusaetzliche Programmierer, die zur Einhaltung eines Termins eingesetzt werden, verzoegern den Termin noch weiter.

Zeppelmiers Folgerung:
Die vier letzten Seiten eines kritischen Listings sind immer verlorengegangen.

Pennington's Beobachtung:
Die Wahrscheinlichkeit, dass ein Programm die ihm gestellten Anforderungen erfuellt, ist umgekehrt proportional dem Vertrauen des Programmierers in seine eigenen Faehigkeiten.

Kleinbrunners Folgerungen:
Wenn eine Programmieraufgabe leicht aussieht, ist sie schwer.
Wenn eine Programmieraufgabe schwer aussieht, ist sie unloesbar.

Was ist das Lieblingsgetraenk von Computer-Freaks?
Hacker-Pschorr...

Es ist unmoeglich, Programme idiotensicher zu machen,
weil Idioten so erfindungsreich sind. (E.Murphy)

Gesetze von der selektiven Gravitation (von E.Murphy):
Ein vom Tisch fallendes Marmeladenbrot faellt immer auf die Marmeladenseite.
Ein aus der Hand fallendes Werkzeug faellt immer so, dass es den groesstmoeglichen Schaden anrichtet.

Murphy war ein Optimist! (Beobachtung von Hofstatter)

Wissen Sie noch mehr dumme Sprueche?
Dann schreiben Sie sie in die Rubrik "Mitteilungen an Franzis"!
end

1 Inf.z.Modembetrieb
2 Franzis-Hausmitt.
3 Brancheninfos
4 Bericht.,Nachtraege
5 Unterhaltung
6 Biorhythmus
99 Hauptmenue
100 Ende

Ihre Eingabe ...6
Geburtsdatum
Datum (t.m.jj)
7.4.60
Also ein Donnerstag!

Franzis-EDV-Buecher - eine kurze Uebersicht

Die genannten Buecher erhalten Sie durch jede Buchhandlung
oder direkt vom Franzis-Verlag, Abt. Buchvertrieb.

Andersen,A.;Zirpel,M.: Die Programmierpraxis der naturwissenschaftlichen Taschenrechner 58 DM
Bonerz,H.: Halbleiterspeicher 9,80 DM
Busch,R.: Basic fuer Aufsteiger 39 DM
Busch,R.: Basic fuer Einsteiger 39 DM
Busch,R.: Basic-Dateien, Listen und Verzeichnisse 19,80 DM
Busch, R.: Die perfekte Behandlung von Zeichenketten 19,80 DM
Busch,R.: Kleiner Basic-Wortschatz 12,80 DM
Busch,R.: Mathematik per Computer 19,80 DM
Busch,R.: Matrix-Operationen 19,80 DM
Busch,R.: Sortierprogramme 19,80 DM
Busch, R.: Zahlen-Umwandlungen 16,80 DM
Feichtinger,H.: Basic fuer Mikrocomputer 28 DM
Feichtinger,H.: Mikrocomputer von A bis Z 24 DM
Feichtinger,H.: Mit Computern steuern 38 DM
Feichtinger,H.: Programmierpraxis mit dem 6502 19,80 DM
Haugg,F.: Software-Engineering und -Qualitaetssicherung 28 DM
Haugg,F.: Stets zu Diensten - Dein Computer 38 DM
Keil,R.u.G.: Der Heim-Computer 8085 38 DM
Klein,M.: Z80-Applikationsbuch 38 DM
Klein,R.D.: Basic-Interpreter 38 DM
Klein,R.D.: Mikrocomputer selbstgebaut und programmiert 38 DM
Klein,R.D.: Mikrocomputer-Hard- und Softwarepraxis 38 DM
Klein,R.D.: Mikrocomputer-Systeme 36 DM
Klein,R.D.: Was ist Pascal? 32 DM
Link,W.: Messen, Steuern und Regeln mit Basic 38 DM
Nutz,K.: Basic-Rechenprogramme 9,80 DM
Pelka,H.: Der Einchip-Mikrocomputer 18 DM
Pelka,H.: Praxis mit Mikroprozessoren 28 DM
Pelka,H.: Von der Schaltalgebra zum Mikroprozessor 28 DM
Pelka,H.: Was ist ein Mikroprozessor? 9,80 DM
Piotrowski,A.: IEC-Bus 48 DM
Plate,J.: Betriebssystem CP/M 56 DM
Plate,J.; Wittstock,P.: Pascal 48 DM
Requardt, H.: Alles ueber PEEK und POKE (Colour-Genie) 16,80 DM
Richter,E.: Die wichtigsten Home-Computer 5,80 DM
Schorn,W.: Prozessrechner-Systemprogramme 9,80 DM
Troitzsch,M.: Mikrocomputer-Schaltungstechnik 38 DM
Wunderlich,F.: Erfolgreicher mit CBM arbeiten 36 DM
Zech,R.: Die Programmiersprache Forth 68 DM
Ein komplettes Verzeichnis (inklusive der vielen Elektronik-Buecher) koennen
Sie ueber die Briefkasten-Rubrik anfordern.

end

1 Inf.z.Modembetrieb
2 Franzis-Hausmitt.
3 Brancheninfos
4 Bericht.,Nachtraege
5 Unterhaltung
6 Biorhythmus
99 Hauptmenue
100 Ende
Ihre Eingabe ...3
IBM-AT auch in Deutschland

Wie IBM Mitte September auf einer Pressekonferenz bekanntgab, wird das multiuser-faehige Geraet IBM-AT ab Anfang 1985 etwa zum Preis des PC-XT auch hierzulande verfuegbar sein. Der PC-junior ist dagegen weiterhin nur in den USA zu haben.

sein. Allerdings ist immer noch keine endgueltige Normenentscheidung gefallen.

Atari mit Finanzproblemen

Nach der Uebernahme von Atari hat Chairman Jack Tramiel von der Atari-Mutter Warner weitere Betriebsmittel angefordert. Glaeubiger verstaekern derweil ihren Druck auf Atari.

SWF-Fernsehsendung

Vom Bundesministerium fuer Bildung und Wissenschaft gefoerdert ist eine 13-teilige Medienverbundserie "Mikroprozessoren/Mikrocomputer", die der SWF ab Freitag, den 21. 9. 1984 um 21.15 ausstrahlt (Wiederh.: 22.9., 17.30), und zwar in seinem Dritten Fernsehprogramm. Das ZDF uebernimmt die Serie ab Montag, 24.9., 16.04. Eine Uebernahme durch den Bayrischen und Westdeutschen Rundfunk ist geplant.

Philips und Sony mit MSX

Noch Ende 1984 wollen u.a. Philips und Sony mit Heimcomputern nach Microsofts MSX-Standardvorschlag auf den Markt kommen. Das Philips-Geraet gibt es in drei Versionen mit 48, 64 und 80 KByte RAM. Fuer das Sony-Geraet gibt es als Peripherie bereits einen Plotter (DIN A4, 8,5 cm/s) und ein 3,5-Zoll-Floppy-laufwerk.

Sharp mit 2,8-Zoll-Floppy

Fuer den Heimcomputer MZ-731 bietet Sharp nun ein Laufwerk fuer 2,8-Zoll-Floppies an, das eine Speicherkapazitaet von 64 KByte je Floppy-Seite aufweist. Insider geben dem 2,8-Zoll-Standard wegen dieser geringen Kapazitaet allerdings ebensowenig Chancen wie der Stringy-Floppy von Sinclair.

VC-20-Nachfolger

Commodore stellt mit dem C-16 einen VC-20-Nachfolger in etwa derselben Preisklasse vor (rund 440 DM). Der C-16 besitzt 16 KByte RAM und 32 KByte ROM mit einem erweiterten Basic (75 Befehle). Sonstige Eigenschaften: Schnittstellen wie C-64, Grafik 320 x 200 Punkte, Schrift 25 x 40 Zeichen, Bildschirm wie beim Apple-II in Grafik- und Textteil aufspaltbar. In Basic sind 12 KByte fuer den Benutzer frei.

Messetermine 1984

13.9.-14.9. Epson-Info-Show, Rheinstein-Penta, Duesseldorf
5.10.-7.10. Berliner Computertag (Berl.-Kindl-Festsaal)
26.10.-28.10. Interradio, Hannover
24.10.-28.10. Hobby-Elektronik, Stuttgart
25.10.-29.10. Apple-Expo, Koeln
25.10.-30.10. Orgatechnik, Koeln
13.11.-17.11. Electronica, Muenchen
Ausblick auf 1985:
19.1.-23.1. Middle East Computing Exhibition, Jeddah, Saudi-Arabien
29.1. - 3.2. Mikrocomputer '85, Frankfurt
4.2. - 6.2. Office Automation Conf., Atlanta, Georgia
20.3. - 24.3. Hobbytronic, Dortmund
17.4.-24.4. Hannover-Messe
7.5. - 11.5. Ifabo + Programma, Wien
15.7. - 18.7. National Computer Conf., Chicago
end

1 Inf.z.Modembetrieb
2 Franzis-Hausmitt.
3 Brancheninfos
4 Bericht.,Nachtraege
5 Unterhaltung
6 Biorhythmus
99 Hauptmenue

ADVENTURE - ECKE

Hallo !

Hier ist wieder die Adventure-Ecke!
Heute möchte ich Euch das Grafik-Adventure ATLANTIS vorstellen! Ihr findet ATLANTIS ab sofort in der Programmbiliothek vor. Besonders interessant ist ATLANTIS, da es, als eines von wenigen, Grafik auf den Bildschirm bringt. Um was geht es nun in ATLANTIS ? Wie der Name schon verrät, dreht es sich um die versunkene Stadt Atlantis. Man muß dieses Atlantis entdecken und auf dem Weg dorthin sechs Schätze finden. Dabei ist, da Atlantis ja versunken ist, besonders auf den Sauerstoffvorrat und auf die Batterien der Lampe zu achten. Das Ganze geht dann ungefähr folgendermaßen vor sich:

Zu Beginn des Spieles findet man sich am Ufer des Pazifischen Ozeans wieder. Vor einem schaukelt ein Segelboot auf den Wellen. Man kann sich nun in verschiedene Richtungen bewegen, z.B. nach Osten durch Eingabe des Buchstabens E (für East). GO EAST wäre auch möglich ist aber überflüssig. Man kann auch auf das Segelboot, hierzu muß allerdings das GO mit eingegeben werden, also: GO SAILBOAT. Eine weitere Erleichterung ist, daß nur die ersten drei Buchstaben eines Wortes eingegeben werden müssen. Aus ENTER BUILDING wird dann ENT BUI. Wenn Ihr es erst einmal allein versuchen wollt noch ein paar Ratschläge:

1. Die Schätze sind durch Sternchen gekennzeichnet (* PYRAMID *)
2. In allen Räumen, in denen ein Achteck auf den Boden gemalt ist, kann man durch Bewegen eines der Steine des Medallions an einen anderen Ort versetzt werden.
3. Die Schätze sind nicht immer an der Oberfläche zu finden!
4. Um seine Lufttanks zu erneuern kann man im Pawnshop neue kaufen.
5. Das Segelboot liegt nicht nur zum Spaß vor dem Ufer.
6. Es gibt Schlüssel, die aussehen wie Pyramiden oder Cylinder.
7. Um aus Davy Jones Locker zu entwischen sollte man einen der Steine des Medallions bewegen.

So, und nun zur Lösung:
Wie vorhin schon angedeutet befinden wir uns am Ufer des Pazifiks. Vor uns schaukelt ein Segelboot. Gehen wir also als erstes auf das Segelboot: GO SAILBOAT.
Hier sehen wir nun das Segel und ein paar Aufbauten. Am Boden liegen ein Scubagear (Taucheranzug), ein Knapsack und ein Spear gun. Wir brauchen nur den Taucheranzug: GET SCUBAGEAR.
Zurück am Strand (GO SHORE) gehen wir nach Osten. Wir stehen vor einem Tempel, auf einer alten Straße. Hier ist nun unter der Erde der erste Schatz versteckt: DIG (grabe) und TAKE MEDALLION. Wenn wir nun in den Tempel hinein gehen sehen wir als erstes die PYRAMIDE. Damit hätten wir den zweiten Schatz.
Zurück am Strand gehen wir nach Süden, in den Pazifischen Ozean. Jetzt sollten wir eigentlich den Taucheranzug anziehen: WEAR SCUBAGEAR. Jetzt können wir tauchen: DOWN. Wir befinden uns nun im Unterwassertempel. Um weiterzukommen müssen wir einen der Steine des Medallions bewegen: PUSH OPAL.
Nun stehen wir am Fuße eines Riffs, vor uns eine Höhle. Zuerst ziehen wir den Taucheranzug wieder aus und gehen dann auf die Klippen. Hier sehen wir einen Spalt in dem Felsen. Am Besten wir probieren den gleich aus und gehen hinein. Da ist alles dunkel. Durch ausprobieren stellt man fest, daß wir nach Norden gehen können. Wir sind nun in einem Zimmer, indem eine Box steht und indem ein Achteck auf den Boden gemalt ist. Zuerst sehen wir uns die Box an und stellen fest, daß sie den dritten Schatz enthält: GET CYLINDER!

Nachdem wir den Opal bewegt haben stehen wir wieder in einem Zimmer. In diesem Zimmer steht ein Bett und liegen ein paar Stoffe. Durch näheres Ansehen der Stoffe hätten wir schon den vierten Schatz: GET PEARLS.
Durch öffnen der Türe können wir in eine alte Biliothek gehen. Dort ist wieder ein Achteck auf den Boden gemalt. Der Balkon, der vom vorigen Zimmer aus erreicht werden kann, ist nicht allzuwichtig. PUSH OPAL.
Jetzt sind wir wieder am Strand. Um die Grotte innerhalb des Riffes zu erforschen gehen wir, am Besten wieder über den Unterwassertempel, dorthin. Vorher müssen wir aber den Tauchanzug wieder anziehen. In der Höhle sehen wir uns zuerst die Felsen genauer an. Hierbei finden wir eine Lampe, die wir gleich anzünden: LIGHT LAMP.
Wenn wir nach Norden gehen, kommen wir in den hinteren Teil der Höhle. Dort müssen wir auch wieder etwas genauer ansehen, nämlich die Wand. Es zeichnet sich eine Tür ab, die wir zunächst öffnen, dann hineingehen (GO DOOR), und von innen wieder zumachen (CLOSE DOOR).
Da am Boden wieder ein Achteck zu erkennen ist bewegen wir wieder den Opal. Die Lampe, die nicht unbedingt notwendig ist, wenn man die Bewegungen im Dunkeln macht, können wir jetzt wieder ausmachen. Dies ist aber nicht unbedingt notwendig. Aber wie kommen wir jetzt hier heraus ?
Ganz einfach: In der Wand ist eine pyramidenförmige Vertiefung zu erkennen. Nach längerem Probieren kam ich auf die Idee, die * PYRAMID * dort einzusetzen: PUT PYRAMID ; und als Antwort auf die Frage IN WHAT : WALL.
Dies funktioniert!
Nun müssen wir nur noch die Tür öffnen und hinausspazieren. Draußen angekommen gehen wir, von dem "Zimmer", in dem wir uns befinden nach Norden, wo wir uns in einer Halle, vor einer großen Skulptur wiederfinden.
Am Boden liegen Kleidungsmaterialien. Beim Durchsuchen dieser stoßen wir auf das fünfte Treasure: GET DOLPHIN.
Tja, aber was nun ? Seitdem wir die Pyramide in die Wand eingesetzt haben befinden wir uns ja wieder im Unterwasserbereich!
Auch hier, vor der Statue, hilft uns das Medallion wieder. Nur ist es dieses Mal ein anderer Stein, den wir bewegen müssen: PUSH DIAMOND !
Darauf gibt die Statue eine Tür frei, wo wir hineingehen. Wir finden uns in einem Gang wieder, den wir nach Norden weiterverfolgen. Der Gang endet in einer Tür, mit einer metallischen Platte daneben. Hier müssen wir den Cylinder in die Platte einsetzen, so wie vorhin die Pyramide in die Wand. Daraufhin können wir eintreten in die Stadt ATLANTIS, die wir nun endlich gefunden haben!
Wir sehen die Gebäude der Stadt. Als Nächstes betreten wir eines dieser Gebäude (ENTER BUILDING) und stehen nun in einem Zimmer mit Fenster. Wir verlassen den Raum über das Fenster und finden uns in einem kleinem Hinterhof wieder. Wir können einen Baum und einen Pile of Debris sehen. Wenn wir jetzt diesen Haufen durchstöbern (LOOK DEBRIS) finden wir unser sechstes und letztes Treasure, ein ATLANTIC COIN !!!!
Durch Eingabe des Wortes SCORE erhalten wir hierfür die Bestätigung und das Adventure endet, erfolgreich gelöst.
Natürlich können wir auch wieder mit den Treasures zurückgehen und über "das Zimmer mit der Pyramide" wieder an die Oberfläche und damit an den Strand gelangen, allerdings dürfen wir dann SCORE nicht eingeben !
Vom Strand aus bietet sich auch die weitere Möglichkeit, mit dem Segelboot zu einem PAWNSHOP zu segeln (SET SAIL, GO SHORE), und dort die Lufttanks des Tauchanzuges zu wechseln (TRADE TANKS). Dies kostet aber ein Treasure!

So das war die Lösung für dieses Adventure! Ich hoffe, es hat Euch gefallen, und Ihr schaut auch nächstes Mal wieder vorbei, wenn Euch der GERALD SCHRÖDER das Adventure MISSION IMPOSSIBLE aus der Adventure-Reihe von Scott Adams (Nr. 3) vorstellt!

Tschau Alex

***　　　　　　　　　　　　　　　　　　　　　　　　　　　　***
****
********* BERICHT VON DER CHIP MIKROCOMPUTER WOCHE *********
****

(Günther Wagner)

Vom 16. - 18. Oktober fand im Arabella-Konferenz-Zentrum in München die CHIP Mikrocomputer Woche statt. Sie stand unter dem Thema 'Computer in der Anwendung' und sollte laut Werbung eine 'übersichtliche Leistungsschau des Computerfachhandels für Handwerk, Gewerbe und freie Berufe' sein.
Als interessierter 'Computeraner' und zukünftiger Bauingenieur wollte ich mir diese Gelegenheit natürlich nicht entgehen lassen. Ich bekam dann sogar noch von Tandy, München, eine Freikarte, so daß ich am Dienstag die Nachmittags-Vorlesungen sausen ließ und mich Richtung Arabellahaus begab. Dort angekommen beeindruckte mich nicht die Messe am meisten - nein, es war die neue Hauptverwaltung der Hypo-Bank - eine kühne, freischwebende Konstruktion getragen von 4 Stahlbetonsäulen. Dieses Meisterwerk moderner Baukunst betrachtete und bestaunte ich bestimmt 10 Minuten lang.
Nun, ich wollte ja auf die Messe; also rein und gleich der 1. Stand war von Tandy. Da wollte ich mich mal näher umschauen. Es sei gleich gesagt: dieser Stand war einer der uninteressantesten und schlechtesten der gesamten Messe - für mich eine herbe Enttäuschung. Es standen 2 oder 3 Modell 2000 rum (und eine Dame) - mehr nicht. Weder das Modell 4 oder 4P, das Modell 100 oder etwa eine Branchenlösung waren zu sehen - einfach nichts. Es war ein Stand zum links liegen lassen, und genau das taten auch die meisten Messe-Besucher. übrigens, auch Genie-Computer waren an keinem Stand zu sehen.
Ich begab mich daraufhin auf ein im Rahmen der Messe stattfindendes Seminar. Das Thema war 'Baustatik' und somit für mich hochinteressant. Dort wurde mein Kommen voll entlohnt, denn was dort in einer knappen Stunde gezeigt wurde war enorm. Auf einem TA 3030 mit 2 Laufwerken (5 1/4") und dem Epson FX-80 wurden 2 Baustatik-Programme vorgeführt. Zwei komplette, relativ schwierige und komplexe Probleme wurden an der Tafel erklärt, in den Computer eingegeben, vom Computer berechnet und anschließend komplett (mit Zeichnungen) auf dem Drucker ausgegeben - alles in knapp 1 Stunde. Die Programme waren in FORTRAN 4 geschrieben. Selbst kniffligste Fragen wurden anstandslos erklärt. Ich verließ beeindruckt dieses Seminar und schaute mir auf 2 Ständen Computer- und EDV-Zubehör an. Der eine Stand (DATAMARK) hatte zu hohe Preise, der 2. hatte einige interessante Angebote.
Nun folgten viele Stände, z.B. Apple, IBM, Commodore, Viktor, TI, Wang, HP, TI, Brother, ... Diese nahm ich mir alle etwas unter die Lupe. Manchmal wurde wenig, manchmal mehr und einige Male sogar viel geboten - auf alle Fälle war es meistens mehr als bei Tandy. Mein besonderes Interesse (wie könnte es auch anders sein) erregten einige CAD-Systeme. Da gibt es bestimmt in einigen Jahren preiswerte und zugleich leistungsfähige Systeme für den Bauingenieur und Architekten. Unter anderem vielen Branchenlösungen für Apotheken, Ärzte, Zahnärzte, Immobilien, Bäcker, ..., und Adreßverwaltung, Lagerverwaltung, FIBU, Lohn etc. auf. Auch einige Textverarbeitungssysteme gab es zu sehen und glücklicherweise kaum Spiele. Trotzdem war die Jugend recht stark vertreten, was manche Aussteller anscheinend ärgerte. Ich weiß auch nicht warum.
Abschließend möchte ich sagen, daß die Werbung im Großen und Ganzen gehalten hat, was sie versprochen hat. Es war sicherlich eine gelungene Messe mit interessanten Sachen - leider gab es aber nichts von Tandy und Genie zu sehen.

PS.: Beim Heimweg verweilten meine Augen nochmals voller Bewunderung bei der Hypo-Bank.

Bericht von der Hobbytronic in Stuttgart (mit 'kleinem' Clubtreffen) und einem Arbeitswochenende beim Hartmut in Heidenrod/Kemel (Günther Wagner)

Zunächst muß ich etwas erklären. Bei Gründung des CLUB 80 war der Hartmut der Erste, der dem Club beitrat. Bereits nach kurzer Zeit wußte ich Ihn als Mitglied zu schätzen, welches konstruktiv am Aufbau mitarbeitet und viel in den Club einbringt.
Bereits nach einigen Monaten lud ich mich mehr oder weniger selbst bei Ihm ein, da ich Ihn endlich mal persönlich kennenlernen wollte (nicht nur über's Telefon).
Den Besuch verbanden wir dann mit der Hobbytronic in Stuttgart, die gerade zu der Zeit stattfand. Der Hartmut fuhr mit seinem Freund Gerald Dreyer (auch ein Clubmitglied) mit dem Auto nach Stuttgart, ich nutzte mein Tramper-Monats-Ticket und kam mit dem Zug (als armer Student habe ich leider kein Auto).
In den Tagen vor der Hobbytronic konnte ich dann noch 2 Mitglieder, von denen ich wußte, daß Sie die Hobbytronic besuchen, kurzfristig über das Treffen informieren. So kam es, daß sich am Freitag, 19. Oktober, um 12 Uhr folgende CLUB 80 - Mitglieder trafen:

Walter Piller mit seiner Frau, Jens Neueder, Hartmut Obermann, Gerald Dreyer und meine Wenigkeit.

Man kann also durchaus von einem 'kleinen' Clubtreffen sprechen, als wir uns im Restaurant an einem Tisch gegenüber saßen und fleißig zu plaudern anfingen. Der Walter hatte z.B. Probleme mit dem Anschluß seines Epson-Druckers; ferner vermißte er 16K in seinem RAM (das Expension war nur zur Hälfte bestückt). Ich erörterte kurz einige wichtige Fragen und ansonsten redete jeder mit jedem was gerade von Interesse war.
Wir bummelten anschließend noch etwas durch die Messe. Tandy war (natürlich!) nicht vertreten (die spielen anscheinend zur Zeit eine Vogel-Strauß-Politik), dafür aber TCS (Trommelschläger) mit den neuen Genie-Computern II S und III S. Der Hartmut und der Gerald waren ganz begeistert davon und auch ich glaube, daß dies zwei sehr interessante Maschinen sind. Mir (und auch dem Hartmut) gefiel recht gut ein Drucker von Mannesmann-Tally für rund 850 DM. Dieser verfügt über ein sehr gutes Schriftbild und kann unsere 'Klötzchengraphik' voll ausdrucken (wer sich für diesen Drucker interessiert, soll sich mal mit dem Hartmut wegen einer ev. Sammelbestellung in Verbindung setzen).
Der Jens, der Gerald und der Hartmut erstanden noch jeweils einen 'Luststengel' (Joystick) - wie man den anschließen kann, wird wahrscheinlich in einem der nächsten Infos erklärt.
Nun trennten uns die Wege, d.h. der Hartmut fuhr (meist ziemlich rasant) mit dem Gerald auf dem Beifahrersitz (meist das nicht vorhandene Bremspedal tretend) und mir auf dem Rücksitz (Leitz-Ordner durchblätternd) Richtung Wiesbaden. Die Fahrt wurde leider durch mehrere Staus erheblich in die Länge gezogen.
Ich hatte mich also für Freitag und Samstag beim Hartmut einquartiert und so kam es, daß wir noch bis ca. 0.30 Uhr 'arbeiteten'. Am nächsten Tag in der Früh gings wieder los bis 17 Uhr, wobei gegen 11 Uhr der Gerald hinzukam. Wir arbeiteten fleißig und lösten manches Problem (auch einige Kopier-Aktionen fanden statt). Einige Ergebnisse finden sich in diesem Info wieder.
So, es ist jetzt 20 Uhr und ich sitze im Intercity 623. Damit dieser Bericht nicht noch länger wird, suche ich nun das Ende. Ich hab's leider noch nicht gefunden und möchte deshalb darauf hinweisen, daß alle Mitglieder Beiträge beisteuern können, dürfen, ... Jeder Beitrag gleich welcher Art ist für's Info meiner Meinung nach interessant. → E N D E
(ich hab's gefunden)

Unter dem Motto 'Kreuz und quer durch den Gemüsegarten' kommen nun die Sachen an die Reihe, die ich mir irgendwann notiert habe, und die noch in's Info gehören.

COBOL !

Da wär mal ein Angebot vom Harald. Er wäre bei entsprechendem Interesse bereit, einen Cobol-Kurs im Clubinfo zu machen. Benötigt werden auf alle Fälle 2 Laufwerke und etwas Fleiß, da Cobol eine 'strenge' Programmiersprache ist, wo man am Anfang schon wissen muß, was man alles braucht. Wer daran interessiert ist, soll mich das mal wissen lassen.

??

Der Hartmut hat 4 Fragen bzw. Bitten:

1. Bitte doch unseren Komtek-Besitzer einmal, seine Maschine zu beschreiben. Sicher bestehen Unterschiede zum TRS-80 Model I und Video Genie. Gibt es die Maschine noch zu kaufen und wenn, was kostet sie. Und so weiter. Ich glaube, daß ich nicht der Einzige bin, der sich dafür interessiert.

2. Bei manchen Programmen wird das Interface OE2TZL angesprochen. Welche Funktionen übernimmt es, wo kann man es beziehen, bzw. wer hat einen Schaltplan zum ev. Nachbau?

3. Wer hat eine EPROM-Karte und kann ev. Eprom's schießen? Wer hat einen Schaltplan zum ev. Nachbau?

4. Wer hat den für ein Programm notwendigen Fernschreibkonverter, kann darüber berichten und ev. eine Nachbauanleitung erstellen?

--- Anmerkung der 'Redaktion': Ich bitte hiermit die angesprochenen Mitglieder, entsprechendes für's Info zu erstellen. DANKE!

Wichtig ist dem Hartmut noch folgendes:
Auf dem Schaltbild der Seriellen/Paralellen-Schnittstelle (2. Info) sind die linken Anschlußbelegungen zum Teil nicht zu sehen (stimmt - wer eine richtige Kopie braucht, kann Sie bei mir anfordern!). Außerdem weist der Author zwar darauf hin, daß die Anschlußbelegung für den Komtek gilt, schreibt aber nicht zu allen Leitungen dazu, welche Funktion sie übernehmen (z.B. linke Seite, untere Hälfte, Leitung 45). Bitte sei so gut, und versuche diese Informationen in einem der nächsten Infos nachzuliefern.

****

H I N W E I S

Achtung:

Bücherverleih etc. ist Vertrauensache. Wer ein Buch ausleiht, muß dafür Sorge tragen, daß der Besitzer sein Buch zum richtigen Termin unbeschädigt zurück erhält. Natürlich kann es mal passieren, daß jemand ein Buch etc. beschädigt (Kinder, umfallende Tasse, etc.), dann sollte aber der Schaden behoben werden. Nur bei gegenseitigem Vertrauen werden die Mitglieder bereit sein, Ihre Sachen auszuleihen. Wer einmal durch Verleiherei einen Schaden hat, wird wahrscheinlich nicht mehr verleihen! OK ?

Ein Mitglied hatte Probleme, meine Disketten zu lesen. Ich verstehe das nicht ganz, da ich nie (außer beim Gerald Schröder) Probleme habe, irgendwelche Disketten zu lesen. Auch sind meine Laufwerke erst ca. 8 Monate alt. Ich bin aber trotzdem der Sache nachgegangen und habe mich mal erkundigt, was das Einstellen der Laufwerke kosten würde: 150 DM für meine Doppelstation.
Ich stecke viel in den Club rein, aber 150 DM für die Justierung meiner Laufwerke zahle ich erst, wenn ich Probleme mit meinen Laufwerken bekomme. Ich hoffe, diesen Standpunkt verstehen alle!

Die Idee, sich als Mitarbeiter für ein bestimmtes Fachgebiet zur Verfügung zu stellen, halte ich für ausgesprochen gut. Ich würde mich da als Fachmann für Hardware anbieten. Da bin ich auch vom beruflichen Background her am besten geeignet (IBM-Computertechniker).

So ein Angebot sollte man immer wahrnehmen - deshalb die Adresse für Hardware-Angelegenheiten:

Walter Zwickel
Lengfelden 123
A-5101 Bergheim

Ich hoffe auf viele interessante Artikel für's Info.

Beim letzten Info frankierte ich ungefähr die Hälfte der Sendungen nach günstigstem Gesichtspunkten (nicht immer ganz korrekt). Da ich nichts von Nachporto gehört habe, nehme ich an, daß die Sendungen ordnungsgemäß angekommen sind. Ich werde deshalb ab sofort die Sendungen so günstig wie nur irgendwie vertretbar frankieren - wer Nachporto bezahlen muß, soll mir das mitteilen (zusammen mit dem Hinweis, welche Versendungsform ich verwendet habe), damit er nur einmal Nachporto zahlen muß.

H I N W E I S : Einige haben Schwierigkeiten mit den Umlauten, da es hier bei den verschiedenen Computern immer wieder zu Fehlern kommt. Das Problem ist ganz einfach zu beheben, indem man die Programme einfach mit einem Textverarbeitungs-Programm kurz bearbeitet und die falschen Umlaute entsprechend ändert.

An dieser Stelle ein weiterer Hinweis: Ich habe mir 10 Stück der günstigen Disketten (47 DM) vom Hartmut bestellt und verwende diese nun doppelseitig mit doppelter Dichte und 80 Spuren.
Ergebnis: Noch kein einziger Fehler, was für eine gute Qualität der Disketten spricht. Ich kann also nur jedem empfehlen, sich beim Hartmut mit Disketten zu versorgen.

Es gibt neue CLUB 80 - Plakate in den Größen A3, A4, A5 und A6 (Handzettel). Dieses Plakat ist wesentlich besser als das vorhergehende. Wer Plakate zu Werbezwecken verwenden kann, sollte diese bei mir anfordern (Größe und Anzahl angeben). Ich hoffe, daß genügend Mitglieder bereit sind, für den Club Plakate aufzuhängen (Computergeschäfte, Hochschulen, Schulen, Messen, ...).

Wer Interesse an Elektronik-Büchern hat, sollte sich beim Hartmut melden - er hat eine größere Anzahl ev. zu verleihen.

-- der tip --- der tip --- der tip --- der tip --- der tip ---

Den Namen unbekannter Systemprogramme lassen sich mit folgender Eingabe schnell herausfinden:

```
10  CLEAR 300: INPUT#-1,A$: PRINT MID$ (A$,2,6)
```

Kassette einlegen, PLAY-Taste drücken und mit RUN starten. Der Einfachheit halber lassen sich System-Programme auch nur durch Eingabe eines Teiles vom Programmnamen laden. Statt System SARGON ließe sich auch System "S" oder "SA" verwenden.

Der Jean-Claude hat folgende Anleitungen:

Tasmon	2 Seiten	deutsch
Monitor 348	4 Seiten	deutsch
ZBASIC 2.2	61 Seiten	deutsch
SCRIPSIT	63 Seiten	deutsch
GEAP 2.2	88 Seiten	englisch
muSIMP	36 Seiten	englisch
PROFILE	12 Seiten	deutsch
VISICALC	116 Seiten	deutsch
BUGOUT	93 Seiten	deutsch
NEWDOS 80 HB	131 Seiten	deutsch
Diskettenbetriebssystem	60 Seiten	deutsch
GDOS HB	55 Seiten	deutsch
FS1	10 Seiten	deutsch
	13 Seiten	englisch
AIR TRAFFIC CONTR.	2 Seiten	englisch

Der Jean-Claude hat ferner folgende Handbücher / Bücher:

NEWDOS 80
GEMINI 10x USERS MANUAL
GEMINI 10x Benutzerhandbuch
Computer GAMES for Business, School and Home for TRS-80 Level 2 Basic

Ich (Günther Wagner) kann folgende Bücher zum Verleih anbieten:

BASIC Sammlung Band 1, 2, 3 und 4 (W.-D. Luther Verlag)
Jeder Band enthält viele Programme mit ausführlicher Programmbeschreibung

BASIC im Büro Band 1 und Band 2 (W.-D. Luther Verlag)
Band 1: Investition, Kalkulation, Wertpapieranalyse, Buchhaltung, Abschreibung, Bilanzanalyse, ..
Band 2: Organisationsübersicht, Baubuchhaltung, Finanzplanung, Angebotsvergleich, ...

Was gut an Tandy, Genie und Komtek ist, erfahren wir ja aus der Werbung. Unter dem Motto

"Was stinkt mir an Tandy, Genie, Komtek ?"

wollen wir mal alles Negative, alle Schwächen usw. dieser Firmen (z.B. auch TCS), der Computer, der angebotenen Software etc. sammeln. Also - Bleistift gespitzt (oder die Nadeln des Matrix-Druckers) und frei von der Leber weg schreiben, was einem so stinkt. Ich glaube, da kommt ganz schön was zusammen.

Vorhang auf

Eine Basic-Routine gestaltet den Vorspann eines Programmes effektvoll.

Häufig werden Einführungen oder Erläuterungen zu einem Programm auf dem Bildschirm, Wort für Wort, wie ein Protokoll heruntergeschrieben. Dieses Auflisten sieht langweilig aus und ermüdet den Zuschauer. Viel schöner ist es, wenn der Vorspann vom Rand in die Mitte des Bildschirmes fließt.

Diese Vorspann-Routine für den TRS-80 kann entweder mit der Funktion »MERGE« oder mit der bekannten »PEEK/POKE« -Anweisung an ein bestehendes Basic-Programm angehängt werden.

Es kann in den DATA-Zeilen entweder ein Hinweis zu dem Programm, eine Kurzbeschreibung oder auch eine Einführung eingetragen werden. Das Programm zeigt entweder von links nach rechts oder rechts nach links eingeschoben den Text an.

Das wahlweise Einschieben geschieht über die RND()-Funktion zufällig, kann aber auch durch eine Variable festgelegt werden.

Anweisungen in Zeile 900 und 1200, die auf bestimmten Computern nicht verfügbar sind, können durch Löschen der Zeile beseitigt werden, ohne Einfluß auf das Programm. Die Variable CS enthält die maximale Anzahl der Zeichen pro Bildschirmzeile.

(Klaus Wientke/hg)

```
100 'Vorspann eines Basic - Programmes
200 '--------------------------------
210 '
300 CLEAR1000
500 CS=64
600 ZEILE=1
700 READA$:IFA$="ENDE"THENRESTORE:GOTO700
800 IFA$=""THENGOSUB2700:GOTO700
900 IFA$="."THENA$=TIME$
1000 IFA$=";"THENFORT=1TO1000:NEXTT:GOTO700
1100 IFA$="CLS"THENCLS:ZEILE=0:GOTO700
1200 IFA$=";;"THENPRINTSTRING$(10,7);:GOTO700
1300 A=LEN(A$):A=CS-2-A:A=A/2:A=INT(A):IFA<1THENA=1
1400 A$=STRING$(A,32)+A$+STRING$(A,32)
1500 Q=RND(2):ONQGOSUB1700,2200
1600 GOTO700
1700 FORX=1TOLEN(A$)-1
1800 PRINT§ZEILE*CS,MID$(A$,LEN(A$)-X,LEN(A$));
1900 NEXT
2000 GOSUB2700
2100 RETURN
2200 FORX=1TOCS-1
2300 PRINT§CS-X-1+ZEILE*CS,MID$(A$,1,X);
2400 NEXT
2500 GOSUB2700
2600 RETURN
2700 PRINTCHR$(30);
2900 ZEILE=ZEILE+1:IFZEILE>=14THENZEILE=14:PRINT:PRINT
3100 RETURN
3200 DATACLS,R E C  H E N Z E  N T R U M   D O R N  B E R G,P R  O U D  L Y
 P R E S E N T S,,.,;,CLS,;;,
3300 DATADieses  Programm ist  ein Vorspannprogramm,Mit dem Programm können
Texte   als,Erklärungen   für  ein  Anwenderprogramm   auf   dem,Bildschirm
eingereiht  werden,ohne daß daraus der  Eindruck eines PROTOKOLLSCHREIBENS,
entsteht,,.,;,;,;,
3400 DATA"ENDE"
3500 ',,,,  bedeutet Leerzeile
3600 ',.,   bedeutet Einrücken der aktuellen Uhrzeit
3700 ',;,   bedeutet Zeitverzögerung
3800 ',;;,  akustisches Signal
3900 ',CLS, Löschen des Bildschirminhaltes
```

Listing zu »Vorhang auf«

Die komplette Club 80 - Bücherbibliothek

Nr. 0001: BASIC-Brevier: Eine Einf. in u. Prog. v. Heimcomp.

Siegmar Wittig --- Heinz Heise GmbH

Ein BASIC-Kurs fuer Nicht-Mathematiker und echte Amateure. Neben Grund- und Aufbaukurs gibt es zahlreiche Beispiele (eine gute Sammlung!).

Nr. 0002: Computerwissen

Michael Scharfenberger --- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von Hard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knobeleien programm. in BASIC

Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

\- --- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel --- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80

Chris Palmer --- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment.

Nr. 0008: More Basic Computer Games

David H. Ahl --- Creative Computing Press, USA

84 Spiele für den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist für Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Wichtig ???

Der Club wächst und wächst - die Bearbeitung und Leitung des CLUB 80 beansprucht also immer mehr Zeit. Aus diesem Grund habe ich mich entschlossen, alles was mit der Programmbibliothek zu tun hat, an ein anderes Mitglied unseres Clubs zu übergeben.

Es steigen zwar dann für den einzelnen die Portokosten, aber ich bin nicht mehr in der Lage, die Programmbibliothek zuverlässig zu verwalten. Immer öfter müßte ich das Studium zurückstellen.

Wer also bereit ist, Programme zu testen, zu archivieren, Programmbestellungen zu bearbeiten, Programm-Anleitungen (falls nötig) zusammenzustellen, die neuen Programme und die Programmänderungen für das Clubinfo mir zuzusenden, etc. sollte dies auf dem Fragebogen ankreuzen.

Natürlich benötigt man (um bequem arbeiten zu können) mindestens 2 Laufwerke. Ich hoffe, daß ich im nächsten Info die neue Adresse für die Programmbibliothek bekannt geben kann (bis dahin sendet Ihr natürlich die Programme zur Archivierung noch an mich - momentan geht es zeitlich noch einigermaßen)

Ich hoffe, daß sich mehrere Mitglieder bereit erklären. Ich werde mir das Mitglied aussuchen, das ich dafür am Besten geeignet halte. Die anderen Mitglieder dürfen mir dann nicht böse sein, daß Sie nicht ausgewählt wurden (übrigens: Sollte sich keiner bereit erklären, muß ich mir ganz ehrlich überlegen, ob es weiterhin eine Programmbibliothek geben wird - die berufliche Zukunft geht vor!)

Das nächste Clubinfo
- erscheint garantiert (Ihr werdet nicht davon verschont!)
- erscheint voraussichtlich Mitte Januar (falls es wider Erwarten später kommt, nicht böse sein).

Bis 1. Dezember
- unbedingt alle Stimmzettel einsenden !

Bis Weihnachten
- Fragebogen, Stellungnahme Clubtreffen, Primzahl-Programm und 1 Diskette zusenden.
Die Diskette kommt mit dem 5. Info mit und enthält dann die Programme zur
- Mitgliederabspeicherung
- Programmabspeicherung
- Bücherabspeicherung

mit den notwendigen Dateien.
(von manchen Mitgliedern habe ich noch Disketten da - die brauchen mir natürlich keine zusenden!).

Ich möchte noch darauf hinweisen, daß ich Ende Januar wieder Prüfungen habe. Deshalb kann es unter Umständen bei der Bearbeitung und Beantwortung von Clubpost mal große Verzögerungen geben. Ich bitte hiermit um Verständnis.

PEEK UND POKE AUF BYTES ÜBER 32767

Wenn Sie POKE 65535,0 versuchen, erhalten Sie einen Überlauf-Fehler. PEEK und POKE benötigen nämlich für die Speicheradresse einen Integerwert. Das große Geheimnis ist, daß Sie jede Adresse über 32767 durch Subtraktion der Zahl 65536 verwandeln müssen. Deshalb ist POKE -1,0 die richtige Anweisung zum Poken einer 0 in die höchste Adresse eines 48K TRS-80. Um sich den Inhalt der höchsten Adresse eines 48K TRS-80 anzusehen, können Sie PRINT PEEK (-1) benutzen.

Um das Poken oder Peeken beliebiger Adressen zwischen 0 und 65535 zu ermöglichen, kann man sich früh im Programm folgende Funktion definieren:

```
DEFFNSI%(S!)=-((S!>32767)*(S!-65536))-((S!<32768)*S!)
```

Wenn Sie dann z.B. in einem Programm auf die Speicherstelle 51400 zugreifen wollen, können Sie schreiben:

```
PRINT PEEK(FNSI%(51400))
```

Oder wenn Sie sich eine Reihe aufeinanderfolgender Speicherstellen ansehen wollen:

```
FOR A!=0 TO 65535: PRINT A!, PEEK(FNSI%(A!)): NEXT
```

Das analoge POKE-Format lautet:

```
POKE FNSI%(A!),A%
```

wobei 'A!' eine Adresse zwischen 0 und 65535 ist und 'A%' eine Zahl zwischen 0 und 255, welche in die Adresse gepoked werden soll.

Der Funktions-Aufruf wandelt einfach 4 Byte lange Zahlen einfacher Genauigkeit zwischen 0 und 65535 in die gleichbedeutenden 2-Byte Integerzahlen um, wobei diese dann zwischen -32768 und 32767 liegen. Zum Rückkonvertieren können Sie folgenden Funktionsaufruf verwenden:

```
DEFFNIS!(I%) = -((I%<0)*(65536+I%)+((I%>=0)*I%))
```

```
Beispiel:  FNIS!(-1)     ist 65535
           FNIS!(32000)  ist 32000
```

ADDITION UND SUBTRAKTION VON INTEGERZAHLEN

Bei vielen Hilfsroutinen und Techniken dieses Kurses werden wir die Notwendigkeit sehen, die nächste Adresse über oder unter einer gegebenen zu berechnen. In anderen Fällen werden wir mehrere Adressen addieren oder subtrahieren müssen.

In den meisten Fällen können wir uns bei Subtraktion und Addition auf die Gültigkeit des Ergebnisses verlassen. Aber in der Nähe von 32767 und -32768 riskieren wir einen Überlauf-Fehler. Wir wissen z.B., daß die nächste Adresse über 32767 die Adresse -32768 ist, aber wenn wir zu 32767 den Wert 1 addieren, oder wenn wir von -32768 den Wert 1 abziehen, erhalten wir einen Überlauf-Fehler.

Die meisten Unterroutinen dieses Kurses berücksichtigen diese Gefahr nicht, in der berechtigten Annahme, daß wir der Gefahr aus dem Weg gehen können. Gewöhnlich addieren wir 1 oder 2 zu einer Adresse, die wir über VARPTR erhalten haben. Wenn Sie bei Programm-Ausführung einen Überlauf-Fehler erhalten, ist es gewöhnlich kein Problem, das Programm entsprechend zu reorganisieren, oder einige Schein-Zeilen einzufügen, so daß ein VARPTR von 32767 oder -32768 für die in Frage kommende Variable nicht vorkommt.

Die Funktion FNIA%(A1%,A2%) ist eine Lösung für die Addition und Subtraktion von Integerwerten. Es liefert die Integer-Adresse, welche durch Addition des zweiten Wertes zum ersten Wert bestimmt wird. Falls Sie bei irgendwelchen Integer-Additionen sicher gehen wollen, so können Sie diese Funktion von Unterroutinen oder anderen Funktions-Aufrufen aufrufen lassen:

```
10  DEFFNIA%(A1%,A2%)=(65536-(A1%+A2%))*((A1%+A2%)>32767)+
    ((0-A1%+A2%)*-((A1%+A2%)<-32768))+(A1%+A2%)*-(((A1%+A2%)<
    32768)AND((A1%+A2%)>-32769))
```

Die FNIA-Funktion rechnet wie folgt:

Falls das Ergebnis der Addition größer wie 32767 ist, wird das Ergebnis von 65536 abgezogen.
Falls das Ergebnis der Addition niedriger wie -32768 ist, wird das Ergebnis von 0 abgezogen.
Andernfalls bleibt das Ergebnis unverändert.

Hier sind einige Beispiele:

```
FNIA%(16554,11)   ist  16565
FNIA%(32767,1)    ist -32768
FNIA%(-32768,-1)  ist  32767
FNIA%(-5,1)       ist     -4
FNIA%(-1,10)      ist      9
```

DAS PEEKEN VON 2 BYTES

Wie Sie bestimmt wissen, erhalten Sie durch PEEK auf jede beliebige Speicherstelle einen Wert zwischen 0 und 255. Der zweite Wert des [illegible]E-Befehls muß ebenfalls zwischen 0 und 255 liegen.

Oft ist es notwendig, mit 2 benachbarten Speicher-Adressen zu arbeiten. Zum Abfragen und Abspeichern von Integerwerten zwischen -32768 und 32767 benötigt der Computer nämlich immer 2 Bytes. Das erste Byte speichert das LSB (Least Signifigant Byte = erstes bedeutendes Byte). Das zweite Byte speichert das MSB (Most Signifigant Byte = höchstes bedeutendes Byte). MSB ist eine Zahl zwischen 0 und 255 und gibt an, wie oft 256 in der Zahl vorhanden ist. LSB ist eine Zahl zwischen 0 und 255 und liefert uns die Integerzahl, wenn wir diese zu MSB*256 hinzuaddieren (Integerzahl = MSB * 256 + LSB).

Um uns 2-Byte Integerwerte an beliebigen Speicherstellen (mit Ausnahme 32767) ansehen zu können, benutzen wir folgenden Ausdruck:

```
PRINT PEEK(A%) + PEEK(A%+1)*256
```
oder
```
PRINT CVI(CHR$(PEEK(A%))+CHR$(PEEK(A%+1)))
```

Falls ein Programm auf die Speicherstelle 32767 zugreifen muß, so benutzen Sie die Funktion FNSI%, drücken aber die Adresse durch eine Zahl einfacher Genauigkeit aus. Für 2-Byte Integerzahlen an beliebigen Speicherstellen, ausgedrückt durch Zahlen einfacher Genauigkeit, können Sie folgenden Ausdruck verwenden:

```
PRINT PEEK(FNSI%(A!)) + PEEK(FNSI%(A!+1))*256
```
oder
```
PRINT CVI(CHR$(PEEK(FNSI%(A!)))+CHR$(PEEK(FNSI%(A!+1))))
```

DAS POKEN EINER 2-BYTE INTEGERZAHL IN DEN SPEICHER

Manchmal möchten Sie den 2-Byte Integerwert einer Speicheradresse ändern. Wir werden dies tun, wenn wir die inneren Zeiger des TRS-80 ändern, um einige spezielle Tricks ausführen zu können. Ebenso kann es vorkommen, daß Sie einen Integerwert in eine USR-Routine poken wollen.

Um einen Integerwert I% zwischen -32768 und 32767 in beliebige aufeinanderfolgende Speicheradressen zu poken, benutzen Sie folgende Anweisung:

```
POKE A%,I%/256: POKE A%+1,I%-INT(I%/256)*256
```
oder
```
POKE A%,ASC(MKI$(I%)): POKE A%+1,ASC(MID$(MKI$(I%),2))
```

Diese einfachen Anweisungen sind leider nicht für die Adresse 32767 anwendbar. Falls Sie über die Grenzen 32767 und -32768 hinauskommen und eine allgemein gültige Routine brauchen, so können Sie folgende Anweisung benutzen, um beliebige Integerwerte in den Speicher zu poken. Sie müssen aber die Funktion FNSI%(S!) und FNIS!(I%) definieren:

```
POKE A%,I%/256: POKE FNSI%(FNIS!(A%+1),I%-INT(I%/256)*256
```


DAS ÄNDERN VON 'MEMORY SIZE' IM BASIC

Wenn Sie unter TRSDOS das BASIC aufrufen, so werden Sie als erstes nach 'MEMORY SIZE' gefragt. Unter NEWDOS und anderen Betriebssystemen bestimmen Sie 'MEMORY SIZE' bereits beim Aufruf von BASIC, da es ein Teil des Befehls ist. Wenn Sie z.B. bei einem 48K TRS-80 einen geschützten Speicherbereich von 61000 festlegen, so ist der Speicherbereich von 61000 bis 65535 geschützt. Das BASIC kann diesen Bereich nicht benutzen.

Manchmal kann es vorkommen, daß Sie innerhalb eines BASIC-Programms den geschützten Speicherbereich ändern wollen, z.B.:

- Sie müssen einer USR-Routine, welche Sie poken oder von Diskette laden, Platz zuweisen.
- Sie müssen Platz ausweisen um Daten abzulegen oder um zeitweise eine Kopie des Bildschirms abzulegen.
- Sie brauchen einen geschützten Bereich um Variablen etc. zwischen mehreren Programmen austauschen zu können.
- Sie möchten einen geschützten Specherbereich verkleinern bzw. ganz aufheben, da Sie diesen nicht mehr benötigen.

Zunächst haben wir hier eine Anweisung, die uns den Wert für die gegenwärtige MEMORY SIZE in die Variable MS! lädt:

```
MS! = PEEK(16561)+PEEK(16562)*256+1
```

Folgende Anweisung gibt die MEMORY SIZE direkt aus:

```
PRINT PEEK(16561)+PEEK(16562)*256+1
```

Um nun die gewählte MEMORY SIZE ändern zu können, setzen Sie in MS den Wert für die neue gewünschte MEMORY SIZE minus 1 und führen folgendes Kommando aus:

```
POKE 16562,MS!/256: POKE 16561,MS!-INT(MS!/256)*256
```

Sie müssen diesem Kommando ein RUN oder CLEAR folgen lassen, damit BASIC die neue MEMORY SIZE setzt. Wenn man MEMORY SIZE ändern möchte, so sollte man dies bereits am Anfang eines Programms durchführen. Die Zeile 1 eines Programms könnte z.B. wie folgt aussehen:

```
1 MS!=64401: POKE 16562,MS!/256: POKE 16561,MS!-
  INT(MS!/256)*256: CLEAR 500
```

Dadurch wird MEMORY SIZE auf 64401 gesetzt und 500 Bytes werden für die Abspeicherung von Strings bereitgestellt. Um es für den Computer zu vereinfachen, können Sie auf Hexadezimalzahlen übergehen. Die Zahl 64400 ist hexadezimal FB90. Unsere Zeile 1 können wir dadurch auf folgende kurze Form bringen:

```
1 PO[illegible]16562,&HFB: POKE 16561,&H90: CLEAR 500
```

CLUB SO - Clubinfo Nr. 5

Januar 1985

Hallo Freunde,

wieder mal ist es soweit - das neue Info kommt ins Haus geschneit. Ich hoffe, Ihr habt wieder alle recht viel Freude daran. Nun es steht ja auch ne Menge drin:

So fängt z.B. der Harald mit seinem Cobol-Kurs an.
Ihr könnt nachlesen, wie man einen Computer modernisiert.
Es gibt den 2. Teil von BASIC Faster & Better.
Ein Beitrag über die Kleinbcuhstaben ist dabei.
Der Gerald Dreyer zeigt Euch, wie man die Taktfrequenz beim Genie erhöht.
Wie man einen Joystick anschließt sagt Euch der Jens.
Mehrere Seiten stammen vom Hartmut mit einigen Programmen etc.
Natürlich wieder dabei: Die Adventure-Ecke.
Der Matthias schreibt über die Computerszene in der DDR.
Noch vieles mehr steht drin in diesem Info, doch schaut selbst.

Ja - und da so ein Info über 4-6 Wochen entsteht, geht es natürlich wieder ein wenig durcheinander. Wieder wurde (auch zu Recht) bemängelt, daß die Abheft-Möglichkeit des Infos schlecht ist. Ok - das stimmt. Aber durch die Verkleinerung und das doppelseitige Kopieren (1 Blatt = 4 DIN A4 Seiten) sowie eben der langen Bearbeitungszeit ist es mir nicht möglich, eine befriedigendere Lösung zu finden. Wer unbedingt das Info im Ordner archiviert haben will, kann ja Klarsichthüllen verwenden.

Das nächste Clubinfo erscheint auf alle Fälle mindestens eine Woche vor unserem Clubtreffen. Wer Beiträge liefern kann, sollte diese bitte nach Möglichkeit bis Mitte/Ende Februar einsenden. Die Beiträge können kopierfertige Vorlagen sein oder Textfiles, die ich auf mein Scripsit rüberziehe oder auch handschriftliches (wenn es nicht anders geht). Ich hoffe, daß ich für das 6. Info wieder recht viel Material bekomme.

Übrigens - das beiliegende Schreiben mit Anmeldung vom Jürgen liegt extra deshalb lose bei, damit es jedem auffällt und damit es jeder so bald wie möglich dem Jürgen zusendet (auch die, die am Clubtreffen nicht teilnehmen!).

Aber nun genug mit dem Prolog - viel Spaß wünscht Euch Euer

Günther Wagner

In eigener Sache:

Es wurde Zeit, die Programmbibliothek abzugeben. Ich bin froh, daß sich der Hartmut bereit erklärt hat, diese zu übernehmen. Ich weiß, daß die Bibliothek bei Ihm in den besten Händen ist. Unterstützt Ihn bitte so gut wie möglich und laßt Ihm vor allem genügend Zeit für die Neuorganisation der Bibliothek. Erleichtert Ihm die Arbeit wo es geht und denkt daran - auch er macht das ausschließlich in seiner Freizeit - also Geduld haben wenn es mal länger dauert.

Ich möchte mich bei all denen entschuldigen, die bei mir noch Programme bestellten und diese von mir nicht mehr erhalten haben. Es ging leider nicht anders. Ihr könnt Sie dann ab ca. März beim Hartmut anfordern. Ich weis - einige werden auf mich sauer sein - das kann ich leider nicht ändern.

Ein wenig geärgert hat mich auch etwas. Ich mußte 18 Mitgliedern ein Schreiben zusenden, in dem ich Sie an die Bezahlung des Jahresbeitrages erinnern mußte. Diese Arbeit hätte ich mir gerne erspart.

Die Mitgliederanzahl beträgt über 30. Es ist also noch gefährlicher geworden, daß ich etwas übersehe oder etwas nicht schreibe usw. Bitte erinnert mich daran, wenn ich etwas vergessen habe. DANKE.

Durch die relativ hohe Mitgliederzahl ist es auch nicht mehr möglich, daß ich jeden Brief ausführlich beantworte. Oft bin ich nun der Ansicht, daß der Brief durch das Info ausreichend beantwortet wird, oder daß eine Beantwortung des Briefes nicht unbedingt erforderlich ist.

Über die Leitung des CLUB 80, also die sog. Vorstandschaft, müssen wir auch auf dem Clubtreffen ausführlich sprechen. Ich 'trete' sozusagen von meinem (damals als Gründer des CLUB 80 selbst erwählten) Amt zurück. Ich muß offiziell entlastet werden (Kasse). Aus wievielen Personen (1 oder mehrere) sich die neue Vorstandschaft zusammensetzt und wie die Wahl vor sich geht ist Thema des Clubtreffens. Ich für meine Person muß Euch mitteilen, daß es durchaus sein kann, daß ich mich als Kandidat für die Clubleitung nicht mehr zur Verfügung stelle. Das 7. und 8. Semester mit der Diplomarbeit sind sehr sehr wichtig und sehr zeitintensiv.

Es sollte sich also jeder für sich überlegen, ob er im Club ein Amt übernehmen kann. Noch einmal ganz klar: Auf dem Clubtreffen wird die neue Clubleitung bestimmt und jeder steht zur Diskussion bzw. kann Vorstand werden!

Es fragte der Besucher: „Warum spricht der Hr. Gregor so durch die Nase?" „Um sein neues Gebiß zu schonen."

„Wetten, daß ich jedes Getränk mit verbundenen Augen erkenne?" gibt der Partygast an. Der Test beginnt sofort. Man reicht ihm ein Glas mit einer Flüssigkeit. Er nimmt einen Schluck und schimpft: „Pfui Teufel, das ist ja Benzin!" „Richtig", sagt der Gastgeber, „aber welche Marke?"

SCS
SPIELE- UND COMPUTER-SERVICE GMBH

KOMTEK I - DATENBLATT

Das Grundgerät (KT 1001/2) bietet folgende Ausstattungsmerkmale:

+ Z-80 CPU 1,93 MHz zukünftig 1,78 MHz
+ 12 K Basic Level II-Interpreter im ROM
+ 16 K RAM (erweiterbar bis auf 64 K)
+ echte Schreibmaschinentastatur (ASC II) mit
  - Tastenentprellung
  - Groß-, Kleinschreibung (echte Unterlängen!)
  - Wiederholfunktion
  - Blockgraphik auf der Tastatur
  - Umschaltung 64/32 Zeichen pro Zeile per Tastatur
  - 64 Zeichen pro Zeile, 16 Zeilen
+ Kassettenrecorderanschluß für alle handelsüblichen Recorder mit Start über Remote durch KOMTEK I
+ Anschlüsse für Fernseher oder Monitor
+ Programmierbarer Tongenerator
+ Echtzeituhr
+ Centronics-Schnittstelle zum Anschluß eines Druckers
+ Vier Sensor-Eingänge für Temperaturfühler, Lichtschranken, Alarmmelder usw.
+ Sechs programmierbare Zeitschalter zur sekundengenauen Fernsteuerung von z.B. Lampen, Alarmanlagen, Heizungen, Elektroeisenbahnen usw.
+ voll Software-kompatibel mit TRS-80*, daher Zugriff auf eines der größten Software-Programme der Welt
+ Bedienungs- und Basic Level II-Handbuch in Deutsch

Optionen, Einbau im Gerät

+ Floppy-Controller, anschließbar bis zu 4 Laufwerke
+ 32 und 64 K RAM
+ Farbe (automatisch 4 Farbkombinationen um alle Schwarz-Weiß--Programme in Farbe wiederzugeben, 16 Farben programmierbar)
+ High-RES Graphik voraussichtlich bis Mitte September 83 verfügbar
+ RS-232 c-Interface ca. Mitte September verfügbar
+ Analog/Digital-Wandler in Vorbereitung (1)
+ Pufferung der Speicherinhalte durch Batterie bei Netzausfall in Vorbereitung
+ ROM-Bank zum schnellen Zugriff auf Anwenderprogramme in Vorbereitung(1)

(1) außerhalb des Gerätes anzuschließen
* eingetragenes Warenzeichen der Tandy Corporation

```
****************************************************************
* C L U B 8 0  -  P R O G R A M M - B I B L I O T H E K *
****************************************************************
```

Wie Ihr dem Clubinfo entnehmen koennt, hat mir der Guenther die Leitung der Programmbibliothek des CLUB 80 uebertragen. Ich hoffe, dass durch diese Arbeitsteilung der Guenther entlastet und die Bedienung der Clubmitglieder mit Programmen aus der Bibliothek beschleunigt wird.

Wie Ihr ja auch dem Clubinfo entnehmen koennt, hat sich der Guenther von einem Rechtsexperten beraten lassen. Nach seinen Aussagen duerfen in einer Programmbibliothek folgende Programme vorhanden sein:

1. Programme, die von Mitgliedern selbst geschrieben wurden.

2. Programme, die aus Computerzeitschriften, Buechern etc. abgetippt wurden. Die Programme muessen einen Copyright- und einen Quell- Vermerk haben.

3. Programme von Urhebern, die es gestatten, dass Ihre Programme in der Clubbibliothek gefuehrt werden.

Das bedeutet, wir muessen unsere Programmbibliothek "ausmisten". Es muessen alle Programme, die nicht unter eine der oben genannten Kategorieen fallen, aus der Programmliste gestrichen werden. Das hoert sich einfach an, bringt aber neben sehr viel Arbeit eine Menge Probleme mit sich.

Methode 1: Ich setze mich hin, durchforste die ganzen Bibliothek und entferne die geschuetzten Programme.
Problem : Die gesamte Punktetabelle der Mitglieder kommt durcheinander. Ich weiss ja nicht wer ungeschuetzte und wer geschuetzte Programme eingesandt hat (zum Glueck!/).
Methode 2: Ich "vernichte" die komplette Bibliothek und wir fangen von vorne an.
Problem : So mancher wird sich um seine Punkte betrogen fuehlen (aehnlich wie der Copyright-Inhaber der Programme um sein Geld) und eventuell aus dem Club austreten oder zumindest keine Programme mehr einsenden. Beides waere natuerlich nicht im Sinne des "Club's" (und das sind wir schliesslich alle!/). Weiterhin waere es natuerlich eine heiden Arbeit. Fuer die Programmeinsender und fuer mich natuerlich auch.
Methode 3: Jeder schreibt mir, welche Programme (nur die ohne Copyright) er zur Programmbibliothek beigetragen hat.
Problem : Es besteht nur eine geringe Kontolle, ob das Programm wirklich von dem Mitglied stammt, das sich als der Einsender ausgibt. Das gibt eventuell Streitigkeiten unter den Mitgliedern.

Loesung : Ich bin zu folgender Loesung unseres Problems gekommen. Ab Januar bis zum geplanten Clubtreffen im Maerz schickt mir bitte jedes Clubmitglied einen Brief, in dem er angibt, welche Programme aus der Bibliothek von ihm "rechtmaessig" eingesandt wurden. Bei Mehrfachnennungen von Programmen durch verschiedene Mitglieder lasse ich mir von den Betreffenden eine Kopie des Programms schicken und teile demjenigen die Punkte zu, dessen Programmversion mit dem in der Bibliothek vorhandenen uebereinstimmt. Sollte sich keine Klarheit ueber die Herkunft des Programms einstellen, werden die Punkte auf die Kontrahenten verteilt. Alle Programme, die bis zum 20. Februar keinen Einsender gefunden haben, werden von mir auf Copyrights untersucht. Geschuetzte Programme entferne ich dabei aus der Bibliothek, ungeschuetzte werden ohne eine Punktgutschrift in der Bibliothek belassen.

Ich habe die Loesung gewaehlt, weil ich glaube, dass jedes Clubmitglied damit einverstanden sein wird. Sie ist eine Alternative bei der sich Arbeitsaufwand und Nutzen die Waage halten.

Wichtige Hinweise !!!

Es koennen in der Zeit bis Ende Maerz keine zusaetzlichen Programme fuer die Clubbibliothek eingesandt werden !!!
Ich bitte die Clubmitglieder nur dann Programme aus der Bibliothek zu bestellen, wenn dies dringend notwendig ist (z.B. es braucht jemand dringend ein Adressenverwaltungsprogramm).
Sollten Programme bestellt werden, die "unrechtmaessig" in der Programmsammlung sind, darf ich diese, so leid es mir tut, nicht mehr versenden. Ich wuerde mich dadurch strafbar machen!!!

```
////////////////////////////////////////////////////////////////
/ Ich arbeite mit Singel Sided / Singel Density - Disklaufwerken! /
/ D.h.: alle Programme, die eingesandt werden, muessen unter NEW-  /
/ DOS 80, einem Abkoemmling davon (z.B. GDOS) SS/SD 40 Tracks oder /
/ TRSDOS 2.3 SS/SD 35 Tracks gespeichert sein oder ein sonstiges  /
/ SS/SD-Betriebssystem enthalten !!!                               /
////////////////////////////////////////////////////////////////
```

Damit sie auch jeder mitkriegt, hier noch mal meine Adresse:

Hartmut Obermann
Schwalbacher Strasse 6
Postfach 27
6209 Heidenrod/Kemel
Telefon: 06124/3913

Das Postfach solltet ihr bei allen Sendungen angeben, die groesser als DIN A5 sind oder empfindliches Material (z.B. Disketten) enthalten.

Ich hoffe, dass Ihr alle damit einverstanden seid, dass ich die Programmbibliothek uebernehme und auch die Massnahmen, die ich leider treffen muss, akzeptiert. Sollte sich jemand finden, der einen besseren Vorschlag zu machen hat, bin ich gerne bereit, darauf einzugehen.
Damit moechte ich dieses Thema fuer heute abschliessen.

Fragen :
========

Ich suche das Mai-Heft der Zeitschrift c't. Hat es vielleicht jemand in seinem Bestand?

Der Hans Koenig hat eine komplette Kassette der Load 80 (Programme zur Zeitschrift Mico 80). Leider fehlt mir die zugehoerige Ausgabe (Februar 1983) der Zeitschrift, um die Programme zum Laufen zu bringen. Sollte sie irgendwo vorhanden sein, schickt sie mir bitte!

Ich besitze seit kurzem, wie man vielleicht am Schriftbild erkennt, einen neuen Drucker, einen EPSON RX-80 F/T+. Ich habe in der Adressenliste des Clubs einige Mitglieder ausgemacht, die den gleichen Drucker besitzen. Ich moechte an sie die Frage richten, ob vielleicht schon Programme bestehen, mit denen man diesem Drucker die TRS 80-Blockgrafik beibringen kann. Vielleicht gibt es auch Hardewareaenderungen (EPROM mit Blockgrafikzeichensatz), die dies ermoeglichen. Weiterhin waere fuer mich interessant zu wissen, welches Textverarbeitungsprogramm den RX-80 am besten unterstuetzt (z.B. Umlaute, Schriftartmoeglichkeiten, Grafikmoeglichkeiten usw.).

Da dieses Schreiben spaeter entstand als mein eigentlicher Clubinfobeitrag, moechte ich zum Schluss nur kurz meine guten Wuensche zum Weihnachtsfest und zum neuen Jahr wiederholen und mich verabschieden.

ERGEBNIS DER CLUBSATZUNGSÄNDERUNG (PUNKT 7)
****

Allen Mitgliedern dürfte ja bekannt sein, daß der Jahresbeitrag seit 1985 DM 30 beträgt - somit ist also mein Antrag vom letzten Info durchgegangen.

Hier aber nun das genaue Abstimmungsergebnis:

18 abgegebene Stimmen - davon
16 Stimmen für Änderung der Clubsatzung
1 Stimme gegen Änderung der Clubsatzung
1 Stimmenthaltung

Damit wurde die notwendige 3/4-Mehrheit eindeutig erreicht.

ANMERKUNG: Auf unserem Clubtreffen müssen wir unsere Clubsatzung unbedingt in einer überarbeiteten Form neu beschliessen. Es muß z.B. der Bereich Programmbibliothek geändert werden. Es ist daran zu denken, die Frage der Vorstandschaft usw. und andere Punkte in die Satzung mitaufzunehmen.

D E S W E G E N sollte sich mal jeder Gedanken machen, was seiner Meinung nach in die Satzung aufgenommen bzw. ergänzt gehört.

****

Z U M F R A G E B O G E N :

Die Idee des Verkaufs von Programmen und Anleitungen über den CLUB 80 war eine schlechte Idee von mir. Schon bei der Veröffentlichung war mir das eigentlich klar. Aber es hat trotzdem nicht geschadet, dieses Thema anzusprechen. Auf dem Clubtreffen werden wir darüber noch mal kurz sprechen.

Der Vorschlag mit LOAD 80 fand hingegen breite Zustimmung. Wir werden auf dem Clubtreffen darüber ausführlich sprechen und entscheiden, ob wir dies offiziell über den CLUB machen (der Jahresbeitrag müßte nochmals erhöht werden), oder ob sich die Personen, die mitmachen wollen, die Kosten anteilig aufteilen (genügend Interessenten sind da, so daß der Betrag für ein Jahr nicht über 30 DM zu liegen kommt). Die rechtliche Frage ist auf alle Fälle geklärt: Solange wir LOAD 80 nicht an andere verkaufen, also nur innerhalb des CLUB 80 kopieren, solange ist dies rechtlich in Ordnung.

Das Kopierkonto wird eingeführt. Der Hartmut (als Leiter der Programmbibliothek) wird dies auf seine Weise regeln, ich (falls nötig) ev. auf eine andere.

Das Thema Programmbibliothek wurde bereits an anderer Stelle besprochen.

Ich danke für die Zusendung der Fragebogen - insgesamt erreichten mich 16 - es war eine große Hilfe für mich.

C l u b t r e f f e n

Einiges gibt es zum Thema Clubtreffen zu sagen. Zunächst einmal: Es findet statt

vom 23.-24. März 1985

im Großraum Frankfurt. Um die Unterkunft kümmert sich der Jürgen (siehe unbedingt beiliegendes Schreiben mit Anmeldeformular). A L L E Clubmitglieder - auch die, die nicht zum Treffen kommen - senden bitte dem Jürgen so bald wie möglich - jedoch bis spätestens 31.1., das Anmeldeformular zurück.

Zum Quartier sei noch gesagt, daß sich der Jürgen selbstverständlich um günstigere Quartiere bemüht, aber den höheren Preis angesetzt hat.

Gut fände ich es, wenn Mitglieder Fahrgemeinschaften bilden. Zum einen wird die Fahrt wesentlich billiger. Ein zweiter - viel wichtigerer Punkt jedoch ist: Wir haben viele Schüler und Studenten ohne eigenes Fahrzeug im Club. Diese würden sich bestimmt freuen, wenn Sie bei jemandem mitfahren könnten. Anhand der Karte aus dem 4. Info und der Veröffentlichung der neuen Adressen in diesem Info können alle Mitglieder ersehen, 'wer mit wem' möglich wäre. (übrigens: Der Jens wurde von mir falsch eingetragen; er wohnt in der Nähe von Stuttgart).

Damit es allen klar ist: Dieses Clubtreffen ist voll beschlußfähig. Alles, was auf dem Clubtreffen beschlossen wird, hat Gültigkeit. So natürlich auch Änderungen der Clubsatzung, Beschlüsse über die Finanzen, 80micro, etc. Im 7. Clubinfo wird ausführlich über das Clubtreffen berichtet, damit auch die anderen Mitglieder erfahren, was gelaufen ist.

Damit sich jeder auf das Treffen thematisch vorbereiten kann, bitte ich nun alle Mitglieder um folgendes:
überlegt Euch, was zu besprechen ist. Macht Euch Gedanken über unsere derzeitige Clubsatzung. Welche Probleme, Themen, Fragen sollten angesprochen werden. Was machen wir bei der 80micro....

Bitte schreibt mir, was alles im offiziellen Teil besprochen werden sollte. Ich veröffentlich dies alles im 6. Info, damit sich jeder schon auf das Treffen vorbereiten kann.

Bitte beachtet auch die Hinweise auf das Clubtreffen, die in diesem Info noch an anderen Stellen zu finden sind (z.B. Vorstandschaft).

Mehr fällt mir jetzt zu diesem Thema nicht ein - ich hoffe nur, nichts wichtiges vergessen zu haben. Ich rechne bei derzeit 33 Mitgliedern mit mindestens 15 Teilnehmern, glaube aber, daß es 20-25 sein werden.

Anmerkung: Von folgenden Mitgliedern ist mir bekannt, daß Sie mit eigenem Auto zum Treffen kommen: Hans König, Jens, Jürgen, Walter

„Gegen Ihr Leiden ist nicht viel zu machen", sagt der Arzt zum ...tienten. „Sie haben es ...erbt!"
„Na schön, Herr Doktor, dann schicken Sie die Rechnung bitte an meinen Vater!"

Hermann Klaus
Gartenstr. 22
7401 Pliezhausen
07127/70024

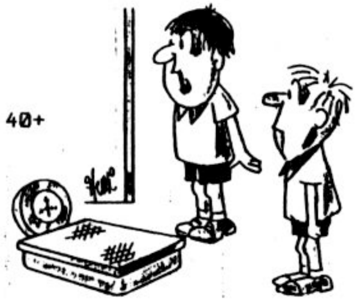
Clubmitglied seit : 02.11.84
Punktestand = 35
System- und Drivekonfiguration :
Tandy Model 4P (bis vor kurzem Model I)/ Epson MX 80 F/T/ 2 Laufwerke (je 40 Spuren/ ss/ dd)
bevorzugtes Betriebssystem : TRSDOS 6.1.1 D (Model I

Schrewe Christian
Fliederweg 32
4000 Duesseldorf 31
0203/740897

Clubmitglied seit : 30.10.84
Punktestand = 35
System- und Drivekonfiguration :
TRS-80 Model III/ ? Laufwerke (40 Spuren/ ss/ dd)
bevorzugtes Betriebssystem : TRSDOS 1.3 und NEWDOS 80

Bozek Hans Juergen
Gut Fachenfelde
2093 Stelle
-

Clubmitglied seit : 11.11.84
Punktestand = 30
System- und Drivekonfiguration :
Komtek 1/ 1 Laufwerk (40 Spuren/ ss/ sd)
bevorzugtes Betriebssystem : TRS1DOS/ NEWDOS 40+



„Ich weiß auch nicht, was das für ein Ding ist. Ich weiß nur, daß Mutti immer furchtbar wütend wird, wenn sie draufsteht!"

Sopp Arnulf
Wakenitzstr. 8
2400 Luebeck 1
0451-791926

Clubmitglied seit : 28.11.84
Punktestand = 30
System- und Drivekonfiguration :
Genie I (mit div. Ein- und Umbauten)/ HRG1B/ Speed-Up/ INT-Timer (regelbar)/ div. Sondertasten/ 3 Laufwerke (2 * 80 Spuren/ ds/ dd/ und 1* 40 Spuren/ ss/ dd)
bevorzugtes Betriebssystem : H-DOS

Perschbach Patrick
Waldstr. 52
5000 Koeln 91
872118

Clubmitglied seit : 24.11.84
Punktestand = 30
System- und Drivekonfiguration :
TRS-80 Model I (Level 2)/ Datasette/ Expansions Interface/ Monitor/ Laufwerke (?)
bevorzugtes Betriebssystem : TRSDOS

Boeckling Ulrich
Am Sonnenhang 11
5414 Vallendar
0261/69522

Clubmitglied seit : 04.12.84
Punktestand = 30
System- und Drivekonfiguration :
TRS-80 Model I (Level 2)/ Expansions Interface/ HRG1B/ Drucker Itoh 8510 A/ 2 Laufwerke (je 80 Spuren/ ds/ dd)
bevorzugtes Betriebssystem : NEWDOS V2

Jablotschkin Rainer
Thiekamp 29
4780 Lippstadt 8
-

Clubmitglied seit : 27.11.84
Punktestand = 30
System- und Drivekonfiguration :
Genie II (48K)/ HRG1B/ Drucker Star DP 510/ 2 Laufwerke (je 40 Spuren/ ss/ sd)
bevorzugtes Betriebssystem : G-DOS 2.2

Robert L.A. Trost
Aalscholverstraat 9
6921 WR Duiven (Holland)
08367/4736

Clubmitglied seit : Dez. 84
Punktestand = 30
System- und Drivekonfiguration :
noch nicht bekannt
bevorzugtes Betriebssystem :

Der Liebestöter

Zwei wohlgeformte Büroblondinen schauen zu, wie die neue EDV-Anlage installiert wird.
»Schau, unser neuer Computer«, meint die eine mit kaum verhehlter Wut.
»Hm.«
»Ersetzt vierzig Männer!«
»Hm.«
»Mistding, das!«

Boecker Dieter
Lehmweg 4
2930 Varrel 1
04451/7640

Clubmitglied seit : 30.12.84
Punktestand = 30
System- und Drivekonfiguration :
Genie I/ DMP 100/ Line Printer VII Laufwerk (ss/dd)
bevorzugtes Betriebssystem : G-DOS

„Du siehst sehr schlecht aus, lieber Freund!"
„Ich habe auch die ganze Woche kein Auge mehr zugetan. Wenn ich nämlich bis morgen keine zwanzigtausend Mark auftreibe, muß ich Konkurs anmelden!"
„Aber, mein Lieber, warum hast du das nicht eher gesagt?"
„Was, du kannst mir wirklich zwanzigtausend Mark leihen?"
„Nein, das natürlich nicht! Aber ich kenne ein ausgezeichnetes Mittel gegen Schlaflosigkeit!"

Grajewski Werner
Zedernweg 29
4220 Dinslaken
02134/54573

Clubmitglied seit : 02.01.85
Punktestand = 30
System- und Drivekonfiguration :
Genie I (64K)/ Star DP 510/ Cassette/ Laufwerke vermutlich ab F
eb.85 verfuegbar
bevorzugtes Betriebssystem : -

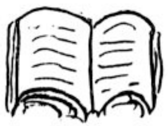
CLUBAUSTRITT:

Weiser Wolfgang
Behaimstr. 12
1000 Berlin 10
(er hatte einen Colour - Computer mit 6502 - CPU)

Computertest

Eine große Elektronikfirma hat einen Computer mit einer bis dahin unvorstellbaren Speicherfähigkeit entwickelt. Der Interessent Mr. Miller steht den Beteuerungen der Fachleute skeptisch gegenüber; er verlangt einen Test.
Der Computer wird mit sämtlichen persönlichen Daten Mr. Millers gefüttert.
Mr. Miller stellt die Frage: »Wo befindet sich mein Bruder zur Zeit?« Der Programmierer gibt ein, und in Bruchteilen von Sekunden kommt die Antwort: »IHR BRUDER BEFINDET SICH AUF EINER GESCHÄFTSREISE IN CHIKAGO!« »Stimmt!« sagt Mr. Miller verblüfft.
»Wo ist mein Vater?« fragt Mr. Miller mit hintergründigem Lächeln. Der Programmierer gibt ein, es dauert etwas länger, dann kommt die Antwort: »IHR VATER SITZT AM EAST RIVER UND ANGELT.«
»Denkste!« triumphiert Mr. Miller, »mein Vater ist schon seit zwei Jahren tot!«
Ratlosigkeit und Enttäuschung bei den Fachleuten. Die Frage wird nochmals als Kontrollfrage eingegeben. Die Antwort: »TOT IST DER GATTE IHRER MUTTER. IHR VATER SITZT AM EAST RIVER UND ANGELT!«

neue Bücher



Nr. 0011:   TRS-80 PROGRAMS

Tom Rugg und Phil Feldman   --- Dilithium Press, Beaverton,USA

32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012:   Programme und Tricks fuer Genie I und Genie II

Clemens Becher, Frank Seger   --- ?

Viele Programme, Tips und Tricks fuer den Genie.

Da hätte ich doch beinahe die Bücher-Verleihliste vom Rainer Jablotschkin übersehen:

- Anwenderprogramme für TRS-80 und Video Genie (M. Stübs)
- Programmieren mit TRS-80 (M. Stübs)
- 57 praktische BASIC-Programme (C. Lorenz)
- Programmieren in Maschinensprache mit Z-80 (C. Lorenz)
- BASIC, Grundlagen und Beispiele, Anwendung in Mikro- und Minicomputern (Günter Abeldt)
- 77 BASIC-Programme (Lon Poole, Mary Borchers)
- BASIC Computer Spiele Band 2 (David A. Ahl)
- Programmieren mit BASIC (Byron S. Gottfried)
- CHIP Spezial 2/81 Programme II
- PCC Games, People's Computer Company
- Computergrafik, Fred Wagenknecht
- Programmieren in BASIC und Maschinencode mit dem ZX-81
- The ZX-81 Companion (Bob Marnder)
- Programmieren von Taschenrechnern 2: Lehr- und übungsbuch für den TI-57 (Hans Heinrich Gloistehn)

Die im 4. Info angesprochene Diskette erhaltet Ihr nicht. Die Programmbibliothek muß ja erst sortiert werden und die Datenverwaltung der Clubmitglieder läuft nicht so recht. Außerdem habe ich nur von ca. 12 Leuten eine Diskette bekommen. Auf dem Clubtreffen erhalten alle Mitglieder Ihre Disketten zurück (und können sich dort auch die Clubadressen auf Diskette kopieren). Die Mitglieder, die zum Treffen nicht erscheinen, erhalten Ihre Diskette dann mit dem 7. Info. Alles klar?

K A S S E N B E R I C H T

Zum veröffentlichten Kassenbericht aus dem 4. Info gab es keine Nachfragen. Die Kasse kann jeder auf dem Clubtreffen einsehen. Ich beschränke mich daher also dieses Mal mit dem reinen Jahresabschluß.

Am 31.12.84 hatten wir folgenden Kassenstand:	- 114.07 DM
Gegenwert der vorhandenen Briefmarken:	8.80 DM

Ferner mehrere Ausgaben der 4 erschienenen Infos.

Für 1985 haben wir nach Abzug des Minus von 1984 und nach Eingang sämtlicher Jahresbeiträge etwas mehr als 900 DM zur Verfügung. Infos wie Nr. 4 oder Nr. 5 kosten je Mitglied zwischen 5 und 6 DM. Wenn wir also den Umfang des Infos halten können, werden wir gerade so eben über die Runden kommen.

Der Christian Schrewe schreibt folgendes:

Ich habe mittlerweile einen Akustikkoppler über den ich auch Programme in Mailboxen abrufen kann. Daher auch meine nächste Frage: Wer von uns hat auch Interesse an DFÜ. Terminalprogramm kann ich beisteuern (Listing aus CHIP). Außerdem gibt es eine WDR-Mailbox über die man Basicode II - Programme bekommen kann. Das Treiberprogramm für TRS-80 besitze ich auch.

Der Jens Neueder sucht nach wie vor Kontake mit Leuten, die eine HRG1B haben.

Der Programmwettbewerb aus dem 4. Info fand kein großes Echo. Dennoch werden 3 Programme an anderer Stelle dieses Infos vorgestellt. Ein weiterer Programmwettbewerb wird wegen des geringen Interesses nicht durchgeführt.

Nicht möglich war mir dieses Mal leider ein Beitrag zu POKE und PEEK. Ich hoffe, wir haben im 6. Info hierzu wieder einiges zu veröffentlichen (denkt doch mal nach, wo man was poken oder peeken kann).

*** F R A G E N *** F R A G E N *** F R A G E N ***

Nochmal Fragen, dieses Mal vom Konrad Josef!

Hat jemand Erfahrung mit einem C-Compiler für Model I ?

Hat jemand Erfahrung mit dem 68000-Paket (Opal-Assembler, RSU-68000 Runtime Simulator, HDT-68000 Debugging Tool) von Wilke ?

Hat jemand Erfahrung mit dem M-ZAL oder dem EDAS Assembler ?

Hat jemand Erfahrung mit POINTERN bei PASCAL80 ?

Der Jürgen Wucherer kann folgende Bücher verleihen:

Lexikon zur Datenverarbeitung rororo Taschenbuch
Anwenderprogramme für VG und TRS-80 Hofacker Nr. 120
Programmieren mit TRS-80 Hofacker Nr. 111
Programmieren in Maschinensprache Hofacker Nr. 119
Alles über Peek und Poke Franzis Verlag von Requardt
BASIC: Sortierprogramme Franzis Verlag von Busch
BASIC für Aufsteiger Franzis Verlag von Busch
Programme & Tricks Genie I+II von TCS

Der Hans König kann folgende Bücher verleihen:

Textsystem in BASIC TV-Verlag (sehr gut)
Basic für den Kaufmann Sybex Verlag
Basic für Aufsteiger Franzis Verlag von Busch
Vom Problem zum Programm Vogel-Verlag
80 Micro Nov. 82
80 Micro Jan.-Dez. 83
80 US TRS User Manuel Jan.-Okt. 79

Thema Fundgrube: CHIP kopiere ich
MC kopiert der Josef Konrad
CP kopiert der Christian Schrewe
die anderen Zeitschriften sieht der Jürgen durch

übrigens: In München gibt es ein regelmäßiges Treffen vom Dieter, Josef und mir. Wer auch Lust hat zu kommen, der soll sich bei mir melden oder einfach am 16.01. um ca. 13.15 Uhr bei Tandy, Briennerstraße, auftauchen (das ist unser nächster Termin!).

Der Walter Zwickel schreibt:

Für das Interface OE2TZL gibt es anscheinend eine Menge Interessenten. Vielleicht kommen wir damit zu einem Club-RS-232 Interface. Allerdings bringe ich einfach nicht die Zeit auf, ein Platinenlayout zu erstellen (wenigstens in der nächsten Zeit). So ist ein Nachbauer auf die "Fädeltechnik" angewiesen.
Meine Programme dazu beziehen sich ausschließlich auf Amatuerfunk-Fernschreiben und Morsen. Deshalb ist ein guter Assembler-Mann gefragt, der ein intelligentes Terminal-Programm dazu erstellt. Sämtliche Unterlagen zur Programmierung der UART 8251 könnte ich für diesen Club-Kollegen kopieren.

Persönliche Vorstellung von Matthias Hallup:

Ich heiße Matthias Hallup und bin 20 Jahre alt. Seit Oktober 1984 studiere ich Informatik an der Universität Dortmund (und 'schlage mich dort mit Computer-Theorie herum').
Meine ersten Kontakte zum Programmieren konnte ich vor ca. 5 Jahren knüpfen (programmierbarer Taschenrechner). Da ich von da ab der Computer-Faszination erlegen war, legte ich mir bald ein Video-Genie Mod. I zu. Nach und nach kamen dann Drucker und Disk-Laufwerke hinzu. Auch baute ich meinen Speicher aus, und legte mir vor kurzem eine HRG-1B (hochauflösende Grafik) zu.
Computer und Programmieren begeistern mich noch immer, doch bin ich nicht davon abhängig (kein Sklave des Computers). Meiner Meinung nach sollte der Computer dem Menschen dienen, nicht umgekehrt. Ich verspreche mir vom CLUB 80 einen regen Erfahrungsaustausch, auch hoffe ich auf Tips von erfahrenen TRS-80- und VG-Anwendern. Ferner möchte ich Leute kennenlernen, die dasselbe Computer-Hobby wie ich haben.

BERICHT ÜBER DIE COMPUTERSZENE IN DER DDR (Matthias Hallup)

Ich fand die Berichte über die Computer-Szene in Norwegen und UDSSR sehr interessant, denn man sollte öfter über die eigenen Grenzen hinaus schauen. Ich selber fahre öfter in die DDR (Verwandte in Halle/Saale), wo ich auch in Ost-Berlin einen computerbegeisterten Freund habe.
Meiner Meinung nach ist die Computerszene in der DDR gegenüber der unsrigen um ca. 10 Jahre zurück. Der Wissensdurst in Sachen Computer ist jedoch umso größer. Computer amerikanischer oder Japanischer Produktion sind zwar sehr begehrt, doch verständlicherweise nicht zu bekommen. Zwar gibt es z.B. den HP-Computer in Ost-Berlin, doch sind diese einigen ausgewählten Firmen vorbehalten (und dort nur bestimmten Mitarbeitern). Computer werden - soweit überhaupt vorhanden - zur Automation der Produktion eingesetzt. Es geht nämlich vorrangig darum, die Produktion und deren Qualität zu steigern. Darum investiert der Staat auch viel in die Roboter-Forschung und -Produktion. Die Firma Robotron sucht mit Ihren Produkten Anschluß an den Weltmarkt (was bei dem Japanischen Vorsprung kaum gelingen dürfte).
Die Hobby-Computerszene der DDR ist wie erwähnt recht klein. Zu kaufen gibt es programmierbare Taschenrechner Made in UDSSR, die aber durchaus z.B. dem TI-57 vergleichbar sind. Taschen- oder Personalcomputer sind kaum zu kaufen (und wären auch für den Normalbenutzer nicht zu bezahlen - schon ein programmierbarer Taschenrechner kostet drüben ca. 800-1000 M, ein für Schüler und Studenten nicht aufzubringender Betrag). Trotzdem gibt es drüben ein paar Exemplare Japanischer Taschencomputer (z.B. Sharp PC-1251), die von Dienst-Reisen (z.B. Ungarn) 'eingeschmuggelt' werden. Software, Tips und Hardware stehen natürlich hoch im Kurs, und Jedes Computer-Buch ist heiß begehrt.
Aber auch hier macht Not erfinderisch: Interfaces u.a. werden selber gebastelt, obwohl man sich die Elektronik-Teile dafür von Freunden im Westen besorgen lassen muß. Auch gibt es in der DDR keine privaten Computer-Clubs wie hier, Informationen werden mündlich weitergegeben und sind nur einem kleinen Kreis vorbehalten. Jugendliche in der DDR sehen mit Wehmut auf die Computer-Szene im Westen, und ein TRS-80- oder VG-Computer zu besitzen wäre eine absolute Sensation für Sie. Hier sieht man doch wieder, welche Computer-Vorteile wir besitzen, obwohl wir uns dessen oft gar nicht bewußt sind. Wenn man mal wieder neidisch auf die Computer-Szene in den USA sieht, sollte man sich dies zum Trost vor Augen halten.

Ich heiße Arnulf Sopp und bin 36 Jahre alt. Zwar gehe ich zur Schule, stehe aber auf der feindlichen Seite des Katheders.
Nachdem ich jahrelang als kultivierter Mensch Computer verachtet hatte, wobei jedoch meine Taschenrechner immer leistungsfähiger wurden, machte ich irgendwann Nägel mit Köpfen und kaufte mir ein gebrauchtes Video-Genie. Das war der Anfang vom Ende. Inzwischen bin ich vollkommen computerkrank und verbringe nahezu meine ganze Freizeit an der Denkprothese.
Meine jetzige Anlage sieht so aus: Genie I (83er Modell) mit diversen Um- und Einbauten, Laufwerke 2mal 80/DS/DD und 1mal 40/SS/DD, Gemini-10X, EG 64 MBA (wohl die beste Banking-Logik auf dem Markt zum Preis der schlechtesten: DM 195,-), HRG 1b.
Mein bevorzugtes Betriebssystem ist tatsächlich "mein" Betriebssystem: Aus G-DOS 2.1b machte ich mit etlichen Zaps und neuen SYS-Files H-DOS, das neben vielen anderen Features das Ansteuern des MBA und der HRG sehr bequem macht. H-DOS kommt von "The HACKTORY", meinem Spitznamen im Computer-Schriftverkehr. Das klingt wie eine Firma und ist auch ungefähr so gemeint. Wenn jemand aus dem Club in der Z80-Maschinensprache fit ist und Lust hat, bei solchen Entwicklungen mitzumischen, würde ich mich über Post oder einen Anruf freuen.

FRAGEN

- Hat Jemand das FORTRAN-Reference-Manual zum FORTRAN-Compiler von Microsoft ?
- Hat Jemand ein vollständiges ALCOR-PASCAL Manual ?
- Hat Jemand fuer das Modell I einen Zeichengenerator mit Umlauten ?
  Wenn ja, ist es ein EPROM ?
  Kann es kopiert werden ?
  Wo außer bei L.Roeckrath kann man noch einen beziehen ?
  Ist evtl. der vom Modell III oder Komtek verwendbar?
  Wer hat eine Schaltung des Videoteils von Modell III ?
- Hat Jemand sein Modell I auf hoehere Taktfrequenz umgestellt? Wenn ja, welche Erfahrungen hat er gemacht ?

Josef Konrad

- Wer hat die CT Nr. 5/84 ?
  Bitte sofort mit dem Hartmut in Verbindung treten!
- Wer hat Dotwriter-Zeichensätze ? Diese werden vom Jens 'gesammelt' und passen für die HRG und auch zum Briefschreiben.
- Wer kann dem Patrick ein Interfacekabel (40-äderig) besorgen?

PROBLEME

Bitte helft dem Hans König weiter! Er hat folgende Probleme:

Mein Problem ist, System Kassetten auf Disk zu übernehmen. Ich habe schon viel versucht, aber ich schaffe es nicht.

Genauso habe ich Schwierigkeiten mit /SRC Files, also Quellcodes zum CMD File zu machen. Ich kenne zwar den Ablauf (mit EDTASM) und im Directory steht der File auch, aber dann klappt nichts mehr, nur die Türen, die ich heftig schließe. Auch hier hätte ich gerne einen Tip.

Ein Auszug eines Briefes von Matthias Hallup

14.) Vorschlag: Was hälst Du von der Einrichtung einer Diskussions-Seite, z.B. zum Thema "Computer und Kinder"?Ich selber möchte meine Meinung zu diesem Thema kurz schildern:In letzter Zeit ist viel von sog. Computer-Kids die Rede gewesen.Diese Kinder bzw. Jugendliche haben zwar viel Computer- und Programmierwissen, doch wie steht es mit Ihrem Allgemeinwissen?Für mich ist es erschreckend,wenn Kinder im Vorschul-Alter auf den Umgang mit Computern getrimmt werden.Sie beherrschen zwar perfekt das Programmieren, doch können Sie sich auch in normalen Deutsch verständlich machen?Letzlich ist eine Generation, die nur mit dem Computer aufgewachsen ist, leicht manipulierbar.Ein autoritärer Staat könnte diese Menschen über Computer-Informationen in die Unfreiheit lenken, Kritikfähigkeit und zwischenmenschliche Kommunikation wären ausgeschaltet, Verständigung nur noch über Bildschirm möglich.Ich stelle diese Schreckensvision nicht deshalb dar, weil ich computerfeindlich bin.Im Gegenteil: Ich bin für technischen Fortschritt und Computereinsatz.Doch sollte dies in Maßen geschehen.Computerbegeisterung ist nicht schädlich, doch zu viel davon kann schwere Folgen haben.Seid Ihr von Freunden nicht auch schon verständnislos angeschaut worden, wenn Ihr Erklärungen in Computer-Chinesisch abgegeben habt? Es besteht also die Gefahr, daß der einseitig Computerbegeisterte den Kontakt zur Umwelt verliert.Dies sollte und darf man Kindern nicht zumuten.Man darf Kinder nicht zu früh das abstrakte Denken per Computer beibringen.Es ist unbedingt nötig, Jugendliche mit Computern vertraut zu machen und Technikfeindlichkeit abzubauen, doch darf dies nicht zu einer kritiklosen Computerliebe führen.Man sollte Kinder im kritischen Umgang mit Computern schulen, und Sie nicht zu Computer-Fachidioten machen.Nur wer selber zu denken gelernt hat, wird nicht Sklave des Computers werden.Computer-Begeisterung ist schön und nützlich, doch blinde Begeisterung hat noch nie genutzt,sondern immer Schaden hervorgerufen (Genügend Beispiele dafür findet man in der Geschichte).Übrigens hat blinde Computer-Feindlichkeit genauso verheerende Folgen.Nur der kritische,aufgeschlossene und informierte Mensch wird die Probleme der Zukunft meistern.
Was denkt Ihr dazu? Welche Meinung habt Ihr zu "Computer und Kinder", "Computer und Gesellschaft" oder "Roboter und künftige Arbeitswelt"? Gerade Computer-Begeisterte wie wir haben eine besondere Verantwortung gegenüber der Gesellschaft und sollten unseren Beitrag zur Lösung der Zukunfts-Probleme leisten!

15.) Werbung: Zum Schluß möchte ich noch ein wenig (einseitige?) Werbung betreiben.Ob Du Sie abdruckst, entscheide bitte selbst.

a.) Für alle diejenigen, die noch Software suchen, kann ich den Software-Händler (Privat-Anbieter) Hanke nur empfehlen.Er hat in seiner Programmbibliothek ca. 1300 Programme für TRS-8Ø/VG (Sprachen,Utilities, Anwender,Spiele etc.), und es dürfte für jeden etwas dabei sein.Auch werden Kopien von vielen Software-Anleitungen zum Kauf angeboten (gegen Erstattung der Kopier- und Portokosten).Wenn Ihr Hanke DM5.- als Schein zusendet, erhaltet Ihr seine umfangreiche Software-Liste.Ich teile Euch dieses nicht mit, weil ich prozentual am Verkauf beteiligt wäre (schön wärs!), sondern weil ich mit Hanke gute Erfahrungen gemacht habe (Wer schlechte hat, sollte dieses bitte mitteilen!).Ich finde es wichtig, eine günstige Quelle für Programme zu kennen, und möchte Euch daher die Adresse mitteilen: H.R.Hanke;Wiener Str.127;A-2620 Neunkirchen. Für einige wird diese Adresse ein alter Hut sein, doch vielleicht konnte ich einigen einen Tip geben.

b.)Im vorigen Info wurde die hochauflösende Grafik HRG-1B lobend (zu recht!) erwähnt.Leider fehlte eine Angabe von Software-Bezugsquellen für diese Grafik.Zwar wurde das Treiber-Programm von RB-Elektronik und der sog. Super-Treiber erwähnt, doch möchte ich dazu folgendes kritisch anmerken:Das mitgelieferte Treiberprogramm enthält nur einen sehr dürftigen Befehlssatz, und die Beschreibung des Super-Treibers ist als katastrophal und unverständlich zu bezeichnen (Mir wurden zu diesem Programm nur drei Seiten mitgeliefert, ohne Beispiele und Erklärungen).Meiner Meinung nach ist der Super-Treiber sein Geld keineswegs wert.Wesentlich besser sind folgende zwei Programme für die HRG-1B (leider nur für Disk zu haben):

a.) GRAPE für TRS-8Ø/VG mit HRG-1B und Drucker.Dieses Programm würde ich als sehr gut bezeichnen.Der Autor Winter hat die Möglichkeiten der Grafik voll ausgenutzt (z.B. 3D-Zeichnen), und nimmt Anpassungen an Computer-Typ und Drucker vor.Nähere Informationen und Preisauskunft bei M.Winter;Im Steinengarten 23;7000 Stuttgart 80.

b.) HRG-PACK von B.Wedell.Programm ist sehr empfehlenswert,zwar nicht ganz so gut wie GRAPE, doch weitaus besser als Supertreiber.Info und Preis bei B.Wedell;Postfach 911265;3000 Hannover 91 (Anpassung für Computer- und Druckertypen möglich!).

16.)Vorschlag: Einrichtung einer Soft- und Hardware-Test-Seite für TRS-8Ø/VG.Hier könnten Leser Ihre persönlichen Erfahrungen mit Programmen beschreiben, Vor- und Nachteile darlegen.Auch könnten Bausätze etc. beschrieben und beurteilt werden.Natürlich hängt so etwas stark vom Engagement und vor allem der Zeit der Leser ab (Test-Berichte erfordern Mühe und Aufwand!).

Okay! Was meint Ihr dazu?

(1)

3. Jan. 1985

Dieter Böcker
Lehmweg 4, ☎ (04451) 7640
2930 Varel 1

Varel, 30.12.84

Gerade noch in das Info aufnehmen konnte ich dieses Schreiben unseres neuen Mitglieds. Wer dem Dieter helfen kann, wendet sich direkt an Ihn.

Sehr geehrter Herr Wagner!

Als erstes möchte ich mich noch nachträglich für die schnelle Beantwortung meines Schreibens bedanken.

Nun kurz zu meinen Problemen:

1.) Mit einem Kollegen zusammen mache ich Basic-Kurse an einer Volkshochschule. Die Hardware stellen wir privat zur Verfügung. Wir besitzen mehrere Videogenie I, von denen 2 mit Floppy unter TRSDOS u. GDOS arbeiten. Wir wären daran interessiert, noch zwei Laufwerke incl. Controller günstig zu erwerben. Vielleicht gibt es Club-Mitglieder, die eins verkaufen wollen oder die eine Adresse kennen, wo man preiswert eins erwerben kann.

2.) Wir sind hauptamtlich Professoren an der Fachhochschule Wilhelmshaven. Da dort an Mikrocomputern überwiegend mit CP/M gearbeitet wird, sind wir daran interessiert, auch unsere privaten Videogenie auf CP/M umzurüsten. Deswegen hatte ich mich ja an Sie gewandt. Für Tips u. Adressen, wo

(2)

wir die erforderliche Hard- u. Software günstig bekommen können, wären wir sehr dankbar.

Weiter: Im Rahmen eines Kooperationsvertrages arbeiten wir (FH W'haven) mit einer ungarischen Hochschule für Elektrotechnik in Budapest zusammen. Dort wird in der Mikrocomputer-Ausbildung der Videogenie I benutzt. Er wird in Ungarn in Lizenz nachgebaut und ist der einzige Mikrocomputer an den die Ungarn ohne Devisenschwierigkeiten herankommen. Für eine Zusammenarbeit auf diesem Gebiet müssen wir uns also anpassen und auch den Videogenie benutzen. Wir möchten also für die Fachhochschule zwei bis drei Videogenie I anschaffen. Da er bei uns nicht mehr gebaut wird, haben wir da einige Schwierigkeiten, da wir aus haushaltsrechtlichen Gründen keine gebrauchten Rechner aus Privathand kaufen dürfen. Händleradressen, wo man noch Videogenie I kaufen kann (auch gebraucht), wären für uns also sehr interessant.

Abgesehen von diesen aktuellen Problemen sind wir natürlich an Software u. Erfahrungs-Austausch immer interessiert. Mein Aufnahme-Formular liegt bei.

Mit freundl. Gruß!
Dieter Böcker

COBOL eine Einfuehrung Teil: 01

==

Was ist COBOL ?
Diese Frage mag sich der geneigte Leser stellen - nun, um es kurz zu machen: COBOL ist eine Programmiersprache die in den 60 ziger Jahren entwickelt wurde und fuer kommerzielle Loesungen eingesetzt wird.

Die Programmiersprache COBOL ist eine bei Home- u. Personalcomputern unuebliche Sprache - das mag vielleich daran liegen, dass sich gerade diese Gruppe weniger mit kommerziellen Loesungeen beschaeftigt als mehr mit Spielen, kleine Anwenderprogramme und eventuell noch Steuerungen fuer Radios, Lueftungen usw.

Warum also COBOL in CLUB 80 ?
Ganz einfach, in meiner Eigenschaft als Systemberater und Anwendungsprogrammierer in einer mittleren Firma mit Sitz in Recklinghausen moechte ich gerne etwas von meinem Wissen hinsichtlich der COBOL-Programmierung im allgemeinen und speziell fuer Anwendungen auf dem TRS 80, Video-Genie, oder Komtek abgeben.

Wenn man die einschlaegige Fachliteratur durchblaettert so findet man fuer unsere Rechner BASIC (Compiler), PASCAL, FORTRAN und eventuell noch FORTH - aber COBOL ?

Nun, vielleicht liegt es auch daran, dass COBOL eine sehr stark strukturierte Sprache (aber mit ganz fantastischen Moeglichkeiten) ist, mit der sich ein Programmierer, der mal eben ein Programm auf die Schnelle entwickelt, einfach nicht anfreunden kann.

Auch ich muss an dieser Stelle sagen, dass ich mich am Anfang recht schwer, nicht mit der Sprache, aber mit der Organisation und Struktur getan habe.

Bevor wir ins Eingemachte gehen, soll hier ein kleiner Vergleich zwischen einer kleinen Anwendung in Basic und Cobol gezeigt werden:

Anhand eines kleinen Programmbeispiels soll ein Vergleich zwischen Basic und Cobol gezogen werden - uebrigens der einzige Vergleich, denn Cobol kann eigentlich mit keiner gaengigen Sprache fuer mathematisch-naturwissenschaftliche Zwecke geeignete Sprache verglichen werden - Cobol ist von seiner Anwendung "hauptsaechlich fuer kaufmaennische Aufgaben entwickelte Sprache (was aber nicht heissen soll, dass Cobol nicht nutzbringend fuer CLUB 80 Mitglieder sein kann).

Das Programm soll einen einfachen Taschenrechner mit den vier Grundrechenarten simulieren - jeder wird bereits schon jetzt ein Konzept im Kopf haben wie er das Problem in Basic angeht:

Es sieht bestimmt so aehnlich aus wie ich es nun darstellen will:

- Erst einmal CLS
- so, jetzt PRINT "Taschenrechner (4 Funktionen"
- und dann noch "=========================="
- eventuell noch ein PRINT
- PRINT "Geben Sie die Rechenart an (+,-,*,/)
- INPUT "Funktion .... ", FU (Dollar)

Spaetenstens an dieser Stelle folgt ein RUN, um einmal anzuschauen, wie das "Programm" funktioniert.

Und gerade das geht nicht in Cobol!
Das gesamte Problem muss im Rahmen eines Programmes codiert, compiliert (und gelinkt) werden und kann erst dann in Verbindung mit der RUN-TIME "zum Laufen gebracht" werden.

An dieser Stelle moechte ich nicht ein BASIC Programm zur Loesung des o.g. Problems anfuehren, ein jeder wird in Gedanken dieses Programm schon fertig haben, so dass ich gleich auf die Cobol-Loesung eingehen kann:

```
IDENTIFICATIN DIVISION.
PROGRAM-ID.                      RECHNER.
AUTHOR.                          H TRAPP.
INSTALLATION.                    CLUB 80.
DATE-WRITTEN.                    30-10-84.
DATE-COMPILED.                   TODAY.
SECURITY.                        KEINE.
REMARKS.                         SIMULATION EINES TASCHENRECHNERS.
*
ENVIRONMENT DIVISION.
WORKING-STORAGE SECTION.
77 WERT-1                        PIC 9999.
77 WERT-2                        PIC 9999.
77 ERGEBNIS                      PIC 99999.
77 WAHL                          PIC A.
*
PROCEDURE DEVISION.
ANFANG.
 DISPLAY "RECHENPROGRAMM", LINE 1, POSITION 1, ERASE.
 ACCEPT WERT-1, LINE 3, POSITION 1.
 DISPLAY "+" LINE 3, POSITION 6.
 ACCEPT WERT-2, LINE 3, POSITION 8.
 DISPLAY "=", LINE 3, POSITION 13.
 ADD WERT-1, WERT-2 GIVING ERGEBNIS.
 DISPLAY ERGEBNIS, LINE 3, POSITION 15.
 DISPLAY "WEITERE RECHNUNGEN (J) E=ENDE", LINE 5, POSITION 1.
NOCHMAL.
 ACCEPT WAHL, LINE 5, POSITION 32.
 IF WAHL = "J" GO TO ANFANG.
 IF WAHL = "E" GO TO ENDE.
 GO TO NOCHMAL.
ENDE.
```

Wenn man sich aufmerksam das Programm anschaut, so kann man auch als ´Noch-nicht-Fachmann´ erkennen, das es sich bei dem Beispiel nur um ein Additionsprogramm handelt. - Und dafuer so einnAufwand ??

Ja gut, man haette sich am Anfang vier oder fuenf Zeilen sparen koennen, aber alles andere ist ein unbedingtes ´MUSS´.

Also ist das Programmieren in Cobol ein recht grosser Schreibaufwand - mit dem Vorteil allerdings, dass,wie auch der ´Noch-nicht-Fachmann´ bestaetigen wird, das Programm sehr gut lesbar und anschaulich (Dokumentation) ist.

Ein Cobol-Programm dokumentiert sich also selbst, REMARKS sind fast ueberfluessig (erfreulich fuer genervte Basic-Programmierer).

Noch einige Worte zum Programmaufbau:
Das Programm, (es ist eigentlich der SOURCE-CODE) wuerde beim Compiler-Lauf mit zahlreichen Fehlern aus der Memory ´rausgeworfen´ werden.

Neben der normalen Syntax ist auch die Position der einzelnen Befehlswoerter von ausschlaggebender Rolle.

Dazu ein kleiner Exkurs in die Geschichte der Lochkarte. (Ich gehe an dieser Stelle nur auf die Dinge ein, die fuer Cobol-Programmierung von Bedeutung sind, Behandlung von Ziffer-und Zonenteil, sowie einige spezielle Eigenschaften sind hier uninteressant).

Um die Positionierung der einzelnen Befehlswoerter zu verstehen, muss man von einer Lochkarte wissen, dass ihr Informationsgehalt maximal 80 Zeichen betragen kann:

Spalte 01-06	Zeilennummer
Spalte 07	Kennzeichen
Spalte 08-11	Zone A
Spalte 12-71	Zone B
Spalte 72-80	Bemerkungen

Fangen wir von hinten an:
Spalte 72-80 ist fuer Bemerkungen zustaendig! In der Praxis heisst dass, das der Compiler die Zeichen in der Spalte 72-80 nicht beachtet - somit sind sie auch fuer uns nicht von Interesse.

Zone 'B' ist fueer die verschiedenen Befehle zustaendig - man kann auch sagen, das eigentliche Programm steht zwischen Spalte 12 und 72.

Die Zone 'A' nimmt bei Cobol eine Sonderstellung ein: Sowohl einzelne Befehle aus den ersten drei DIVISIONS, die DIVISIONS selbst, PARAGRAPEN und SECTION´s beginnen (!) in Zone 'A'. (Es sind nicht allzu viele Befehle, die in Zone 'A' beginnen (muessen) - man muss sich die Befehle mit diesem Merkmal einpraegen sonst sucht man oft vergeblich stundenlang nach einem Fehler in der Syntax der ueberhaupt nicht vorhanden ist.

Spalte 7 allein hat nur zwei Aufgaben: Kennzeichen '*' fuer REMARKS und Kennzeichen '-' fuer Verbindungen zwischen den einzelnen Befehlszeilen.
Steht in Spalte 7 ein '*' so wird die gesamte Zeile vom Compiler nicht beachtet, (wie REM in Basic).
Steht in der 7. Spalte ein '-', so bedeutet das fuer den Compiler, dass diese Zeile an die vorhergehende anzuhaengen ist. (kommt nur bei Ausgaben auf den Bildschirm oder den Drucker vor - und auch nur dann, wenn die Ausgabezeile + Befehlswort (DISPLAY oder WRITE) laenger als Zone 'B' ist.

Spalten 1 bis 6 wiederum werden vom Compiler nicht beachtet, der EDITOR setzt hier seine Zeilennummerierung ein (die bei Cobol Programmen nicht noetig sind).

Mit diesen Ausfuehrungen im Hinterkopf und den Blick auf unser Beispielprogramm wird der aufmerksame Leser an dieser Stelle sofort den Einwand bringen, dass das kurze ´Source-Listing´ nicht in der fuer Cobol spezifischen Form geschrieben ist.

Ganz richtig, wenn dieser Sorce-Code compiliert werden wuerde,

waere das totale Chaos perfekt - es sollte damit auch nur ein kurzer Ueberblick ueber die Lesbarkeit eines Cobol-Source-Codes gegeben werden.

So, genug der Vorrede, wenden wir uns der ersten von vier DIVISION´s zu aus der ein Cobol-Programm besteht.

```
IDENTIFICATION DIVISION.
PROGRAM-ID.                     Programm-Name
AUTHOR.                         Kommentar
INSTALLATION.                   Kommentar
DATE-WRITTEN.                   Kommentar
SECURITY.                       Kommentar
```

Die Identification Division besteht aus sechs Anweisungen, zwei davon sind fuer einen korrekten Compiler-Lauf unbedingt erforderlich, die letzten vier Anweisungen koennen entfallen.

Im Hinblick auf Allgemeingueltigkeit sollte man sich jedoch angewoehnen, auch die anderen vier Anweisungen (AUTHOR, INSTALLATION, DATE-WRITTEN und SECURITY) anzugeben, ein so greosser Schreibaufwandd ist es nicht und man hat alle benoetigten Angaben ueber den Urheber, Datum und Installation an der Stelle wo man sie haben sollte und nicht irgendwo zwischen ein paar REMARKS.

Man beginnt also beim Schreiben eines Programms bei der IDENTIFICATION DIVISION. (in Zone A, Spalte 8)

gefolgt von:

PROGRAM-ID.	Programm-Name	(Spalte 8)
AUTHOR.	Kommentar	(Spalte 8)
usw.		

Was bedeutet nun IDENTIFICATION DIVISION ?

Die IDENTIFICATION DIVISION leiete das Programm ein und stellt die erste von vier DIVISIONS dar - Kennzeichen fuer den Compiler, dass hier der Source Code beginnt.

PROGRAMM-ID.
bedeutet eigentlich PROGRAM IDENTIFICATION, und fordert einen Programm-NAmen. Dieser Programmname kann 32 Stellen lang sein, relevant sind allerdings nur die Stellen 1 bis 6.
Mit den ersten sechs Zeichen weist der Compiler bei Programmfehlern in den Source Codes hin oder aber bei der Run-Time auf Fehler waehrend des Programmlaufs (wichtig, wenn man mit CALLS arbeitet, zur Kennzeichnung, in welchem Programm denn nun der Fehler aufgetreten ist).

AUTHOR.
verlangt den Namen des Erstellers - wichtig: Abkuerzungen mit Punkten sind nicht erlaubt.
Falsch : H.TRAPP
Richtig: H-TRAPP
Man sollte sich von vornherein bei Cobol angewoehnen nur mit Bindestrichen und nicht mit Punkten zu arbeiten. (Diese haben, wie wir gleich sehen werden, eine andere Bedeutung).

INSTALLATION.

Adresse des Erstellers oder freier Kommentar.

DATE-WRITTEN.
Datum, aber bitte in der Form 05-06-1984 (fuer den 5. Juni)

SECURITY.
hier gibt man im allgemeinen: KEINE an.

Alle Anweisuungen haben es gemeinsam, dass sie in Spalte 8, Zone A anfangen.
Die einzelnen Statements (PROGRAM-ID,AUTHOR usw.) werden mit einem Punkt '.' zur Ende-Kennzeichnung abgeschlossen.
Der Programm-Name und die Kommentare koennen direkt hinter dem Punkt des Statements geschrieben werden - hier benutzt man allerdings zur besseren Uebersichtlichkeit die Moeglichkeiten des EDITOR´s (mit dem der Source Code erstellt wird) mit dem man z.B. den uebernaechsten TAB anspringt und alle Eingaben in der gleichen Spalte beginnen laesst.
Auch hier muss zur Ende-Kennzeichnung ein Punkt gesetzt werden.

Beispiel eines COBOL-Source Codes:

```
1----67A---B------------------------------------ usw. ... --72
       IDENTIFICATION DIVISION.
       PROGRAM-ID.                 DEMO-PROGRAMM.
       AUTHOR.                     H-TRAPP.
       INSTALLATION.               KRANICHSTR 46  4270 DORSTEN.
       DATE-WRITTEN.               07-11-1984.
       SECURITY.                   KEINE.
      *
      *Hier kann ein Kommentar stehen.
      *
```

Sie erinnern sich ??
Der Stern in Spalte 7 laesst jegliche Kommentare zu. Kommentare und DISPLAY (auf den Bildschirm) lassen Gross- u. Kleinbuchstaben zu.
Statements und Anweisungen muessen in Grossbuchstaben angeschrieben werden.

Mit dem Stern '*' in Spalte 7 kann man den Source-Code noch uebersichtlich gestalten oder, wie in dem ersten Beispiel, zusaetzliche Dokumentation erstellen ~~werden~~.

z.B. *DATE-COMPILED. 10-11-1984.

Die DATE-COMPILED Anweisung ist ueberigens auf Grossrechenanlagn Pflicht.

Das waer´s fuer Teil 1. - Im naechsten Teil gehen wir auf die zweite und dritte DIVISION ein - soviel sei schon hier verraten:

ENVIRONMENT DIVISION.	(Maschinen- u. Datei (Drucker) Definition).
DATA DIVISION.	(Erklaerung der Daten und Arbeitsfelder)

Wer hat Angst vor A S S E M B L E R ?

BASIC ist eine sehr bequeme Sprache, bei der sich mit einem Wort ein ganzer Rattenschwanz von Maschinenspracheprogrammen aufrufen läßt. Ich habe keine Ahnung, um wieviel ich mich nach oben oder unten verschätze, aber die simple Anweisung PRINT dürfte gut und gerne 1 kB des Interpreters benötigen. Hier liegt auch der Grund, warum BASIC so langsam ist.

Beim Erstellen eines Programms ist es genau umgekehrt. Das liegt daran, daß wir beim Programmieren einer Maschinenspracheroutine in Bahnen denken, die uns sehr ungewohnt sind. Da heißt es meistens nicht, wie in BASIC, "Tu was!", sondern überwiegend "Tu was wohin!". Das bedeutet folgendes: Da unser Z80 nicht rechnen kann, muß er nach einem möglichst sinnvollen Algorithmus so lange Daten schaufeln, bis an der vorher bestimmten Stelle der richtige Wert steht.

Hier ein konkretes Beispiel. An der Bildschirmstelle 0, also an der Speicherstelle 3C00h (15360d) soll der Großbuchstabe A angezeigt werden. Das ist ASCII 41h bzw. CHR$(65). In BASIC ist das fix getan mit dem Befehl

```
PRINT§Ø,"A"
```

Der Interpreter erkennt den PRINT-Befehl und sucht nun zunächst nach weiteren Operanden wie TAB, USING, ,, ;, (und eben auch § (at, Klammeraffe). Schon schneller geht es, wenn wir den Interpreter nicht wie oben beschrieben auffordern, etwas zu tun, sonder gleich etwas irgendwohin zu tun:

```
POKE15360,65
```

Jetzt passiert zwar noch immer so allerhand, aber ohne Umschweife wird nun der Code 41h (65d, "A") nach 3C00h geladen. Auch hier habe ich nicht im ROM nachgezählt, aber es dürften vielleicht 200 Bytes sein, die zum Syntaxcheck und zum eigentlichen Ladevorgang durchlaufen werden.

Am schnellsten geht es in Maschinensprache. Das hat man auch schon bei Microsoft gemerkt, denn der Interpreter ist ein gewaltiges Maschinenprogramm. Wie er die Bildschirmecke lädt, weiß ich nicht. Vielleicht so:

```
LD      A,41H
LD      (3CØØH),A
```

Der erste Befehl setzt sich aus zwei Bytes zusammen, der untere aus dreien. Wir haben also nun mit 5 Bytes in einer unvorstellbar kurzen Zeit das A angezeigt.

Und das war ganz simpel. Zunächst wurde ein Register (soviel wie Speicher) der CPU mit dem Code geladen (LD = load). Es ist der Akkumulator oder Akku. Es gibt noch eine ganze Reihe weiterer Register, auf die ich jetzt aber nicht eingehen will. Anschließend wurde dieser Code an die Speicherstelle 3C00h weitergegeben. Den Befehl

```
LD      (3CØØH),41H
```

kennt die CPU leider nicht, sonst wäre es noch simpler und schneller gegangen.

Es hat übrigens keinen Sinn, diese Assembler-Befehle ohne einen Assembler einzugeben. Da der BASIC-Interpreter sie nicht kennt, kommt nur ein ?SN-Error dabei heraus.

Diesen Info-Beitrag schreibe ich nicht, um meine neue Schreibmaschine auszuprobieren. Er hat vielmehr eine Vorgeschichte und einen ganz bestimmten Grund.

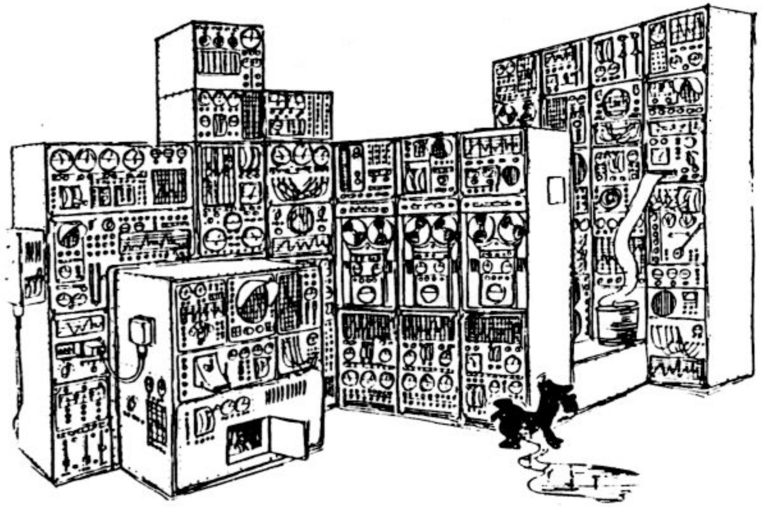
Die Vorgeschichte: Als Noch-nicht-Mitglied des Club 80 wollte ich mich informieren, was bei euch so läuft und bestellte bei Günther die bisherigen Infos. Bei der Lektüre fiel mir auf, daß kein einziges selbstgeschriebenes Assemblerprogramm drinsteht. Ich schlug vor, mit einem zunächst einfachen Artikel zu diesem Thema die Lücke bewußt zu machen und dann eure Reaktionen abzuwarten. Wenn es sich zeigt, daß im Club Interesse an der Maschinensprache besteht, will ich gerne noch mehr (aber nicht mehr so einfache Dinge) bringen.

Mein pädagogischer Ehrgeiz ist allerdings gering - zumindest außerhalb meiner beruflichen Tätigkeit. Günther schlug nämlich vor, einen Assemblerkurs ins Info zu bringen. Dergleichen gibt's im Buchhandel von Berufeneren als mir. Besonders empfehlenswert ist z. B. die Serie "View from the Top of the Stack" in The First Book of 80-US.

Der Grund: Vielmehr ist es mit diesem Artikel meine Absicht, euch für die Maschinensprache zu interessieren, indem ich euch zeige, daß das keine Zauberei ist. Wie jede Sprache hat Assembler natürlich auch ihren Wortschatz, ihre Rechtschreibung, ihre Syntax, genau wie BASIC. Sie sind ebenso einfach erlernbar wie BASIC. Und irgendwann hat man sich auch an die etwas umständlich anmutende Denkakrobatik gewöhnt, die erforderlich ist, um z. B. 7 und 5 zu multiplizieren, denn die CPU kann, wie gesagt, nicht (jedenfalls fast nicht) rechnen.

Wenn ihr jetzt Blut geleckt haben solltet, laßt es Günther wissen. In der Folge schreibe ich dann (oder eben nicht, wenn es keinen interessiert) weitere Beiträge, in denen ich Maschinenspracheprogramme vorstelle und so kommentiere, daß auch der Laie versteht, welchen Weg sie gehen. Die Programme restlos zu durchschauen, erfordert allerdings, daß ihr per Buch oder wie auch immer diese Sprache erlernt. Und genau das will ich mit diesem Beitrag bezwecken.

Arnulf Sopp



BASIC-Patch fuer RESTORE n

In der Programmbibliothek existieren zwei Versionen eines Programms mit dem der BASIC-Befehl RESTORE erweitert wird. RESTOREN/BAS und RESTOREN/CMD ermoeglichen es, den DATA-Zeiger auf eine bestimmte Zeile innerhalb eines DATA-Feldes zurueckzustellen (z.B. RESTORE 140). Dies kann bei bestimmten Anwendungen sehr nuetzlich sein. Auf alle Faelle ist es unpraktisch, dass beide Programme extra geladen und durch das Setzen von HIMEM geschuetzt werden muessen.
Eine elegantere Loesung des Problems sehe ich darin, dass relativ kurze Maschinenprogramm ins BASIC einzubinden. Das hat zur Folge, dass der RESTORE n - Befehl ein fester Bestandteil des Diskbasic wird.
Das Maschinenprogramm RESTOREN/CMD disassembliert:

```
21E17F      LD      HL,7FE1H
220440      LD      (4004H),HL
C3CC06      JP      06CCH
E3CC06      EX      (SP),HL
7C          LD      A,H
FE1D        CP      1DH
E3          EX      (SP),HL
C2781D      JP      NZ,1D78H
D7          RST     10H
FE90        CP      90H
C2791D      JP      NZ,1D79H
CD5B1E      CALL    1E5BH
E5          PUSH    HL
CD2C1B      CALL    1B2CH
0B          DEC     BC
ED43FF40    LD      (40FFH),BC
E1          POP     HL
C1          POP     BC
[illegible]E1D     JP      1D1EH
Spalte 1    Spalte 2  Spalte 3
```

Das Programm liegt im Bereich 7FD8H bis 7FFDH mit dem Entry-Point 7FD8H.
Das Programm ist in zwei wesentliche Teile unterteilt. Block 1 (Zeile 1-3) schreibt die Adresse der eigentlichen Befehlserweiterung (Block 2, Zeile 4-Ende) in die Speicherstelle 4004H. In 4003H steht ein C3H, was einen Sprung zur nachfolgend angegebenen Adresse (jetzt 7FE1) zur Folge hat. Zeile 3 des ersten Blocks bewirkt einen BASIC-Warmstart.
Wird jetzt in einem BASIC-Programm der Befehl RESTORE benutzt, wird statt der "alten" RESTORE-Routine (Addr 1D91H), die neue RESTORE n-Routine (Addr 7FE1H) angesprungen.
Der Block 2 soll uns hier funktionsmaessig nicht interessieren.
Um den neuen Befehl ins BASIC einzubinden, muss man folgendermassen vorgehen:

1. Suche einen freien Bereich im BASIC/CMD (mit Superzap).
2. Suche die Ladeadresse des freien Bereichs.
   Dazu muss man vom freien Bereich aus rueckwaerts das naechste 01-Byte finden. Diesem 01-Byte folgt ein 1-Byte-Wert, der angibt, wieviel Bytes als naechstes geladen werden sollen (00=256 Bytes). Die naechsten beiden Bytes enthalten die Ladeadresse des zu ladenden Blocks im Format LSB/MSB.
   Beispiel: 1. 2. 3. 4.
   01 00 C9 65
   1. Beginn des Ladeblocks
   2. Anzahl der zu ladenden Bytes (hier 256)
   3. und 4. Ladeadresse des Blocks LSB/MSB (hier also 65C9H)
3. Nun gehe ins BASIC und lasse einige Programme laufen. Ueberpruefe, ob der gefundene, ungenutzte Bereich auch tatsaechlich frei ist und nicht vom BASIC aus ueberschrieben wird (z.B. Buffer usw.). Hier hilft der im NEWDOS eingebaute Debugger und ein ROM-Listing.

Bis hier ist alles nur Vorgeplaenkel. Jetzt gehts zur Sache!!!

4. Bringe den Block 2 des CMD-Programms RESTOREN als Hex-Dump im freien Bereich von BASIC/CMD unter. Dazu werden einfach ab der Zeile 4 des Dissassembler-Listings die Hexzahlen (Spalte 1) mit Superzap (MOD-Funktion) in den entsprechenden Sector der BASIC/CMD-File uebertragen.
   ACHTUNG: Nur die Zeilen 4-Ende !!!
5. Stelle jetzt die Einsprungadresse des neuen Befehls fest. Zaehle dazu von dem ersten Byte der Block-Ladeadresse bis zum Beginn des neu eingegebenen Maschinenprogramms. Addiere die gefunden Zahl zur Block-Ladeadresse. Das Ergebnis ist die gesuchte Einsprungadresse.
6. Es werden noch sechs Bytes fuer den ersten Block des neunen RESTOREN-Programms gebraucht. Diese zwacken wir einfach von den Bytes des geaenderten Blocks ab. Wir zaehlen also vom ersten Byte der Blockladeadresse, um die Anzahl der zu ladenden Bytes nach vorne. Steht z.B. 00H im Block-Groessen-Byte zaehlen wir 256 Bytes. Ist dies geschehen, gehen wir von dort 6 Bytes zurueck.
7. Hier fuegen wir nun folgende Bytefolgen ein:
   01 = neue Blockblademarke
   04 = es sollen vier Bytes geladen werden
   0440 = Blockladeadresse (=4004)
   xxxx = in Punkt 5 ermittelte Einsprungadresse des BASIC-Patchs.

   Diese Bytefolge ersetzt den Block 1 des Maschinenprogramms.
8. Wir muessen jetzt noch den alten Block-Groessenwert um 6 Bytes vermindern (wir haben ja einen neuen Block im Bereich des Alten eroeffnet). Dazu ziehen wir vom alten Block-Groessenwert die 6 Bytes ab, die wir fuer unsere benutzt haben. War der Wert 00H, muss er jetzt durch FAH ersetzt werden.
9. Hoffen wir, dass alles geklappt hat!!!

Ich habe den RESTOREN-Befehl in das SUPERDOS- und das NEWDOS- BASIC eingearbeitet und bin sicher, dass es sich auch in andere BASIC-Verionen (z.B.: TRSDOS1.3, 2.3, VTOS-BASIC) einbauen laesst. Selbstverstaendlich koennen auch andere Patches eingebaut werden.
Sollte es jemand zu umstaendlich sein, dieser Anleitung zu folgen, kann er die geaenderten BASIC/CMD-Files fuer SUPERDOS-BASIC und NEWDOS-BASIC bei mir anfordern. Schickt mir dazu eine Systemdiskette (SingleSided/SingleDensity) des gewuenschten Diskoperatinsystems und einen frankierten Rueckumschlag. Leider habe ich keine Zeit auch noch andere Diskbasicversionen umzuarbeiten. Deshalb bitte nur SUPER- und NEW-DOS-Disketten einschicken!!!
Aber selbst probieren sollte es meiner Meinung nach jeder einmal. Man bekommt dadurch ungeahnte Sicherheit im Umgang mit Superzap und einen guten Einblick in den Aufbau von System- und CMD-Files.
Quellenhinweise: MC Nr.4 - April 1984, Seite 64
Changing the BASIC-Language, 80 micro 6/83

Und weils so schoen war, gleich noch ein Patch!!!

Wenn man vom DOS aus DIR oder vom BASIC aus CMD"DIR aufruft wird als erstes der Bildschirm geloescht bevor das Inhaltsverzeichnis ausgegeben wird. Dies stoert bei manchen Anwendungen. Man kann dieses Manko aber durch fogendes Patch in der fuer den DIR-Befehl zustaendigen Systemfile beseitigen.
Modifikation des SYS8/SYS mit Superzap (Befehlsfolge):
DFS (ENTER) , SYS8/SYS (ENTER) , 4 (relativ Sector in File) (ENTER) , MOD 5E , 031D (ENTER) , Y , (ENTER) , EXIT .
Mit dieser Befehlsfolge wird das relative Byte 5EH im relativen Sector 4 des Systemfiles 8 von 031CH in 031DH geaendert. Das Loeschen des Bildschirms auf Aufruf des Diskinhaltsverzeichnisses unterbleibt nun.
Quelle: Clubinfo Nr. 6 des Computerclubs Bremerhafen.

Locked Out Tracks

Verdammt noch mal, jetzt habe ich schon zum dritten mal versucht die Diskette zu formatieren und immer wieder wird der Track 36 als nicht formatierbar ausgewiesen oder beim Verifizieren als fehlerhaft bezeichnet!!!
Wer kennt dieses Problem nicht?!? Ich bin fast sicher, dass jeder von uns schon einmal eine solche Diskette in seinem Bestand hatte. Soll man es riskieren, die nur mit Problemen formatierte Floppy zu benutzen und eventuell ein Programm oder eine Datei zu verlieren oder soll man die Disk gleich zum Muell werfen? Beides ist nicht noetig, es gibt einen tragbaren Kompromiss!
Wer das weniger Gute aber um so aeltere TRSDOS 2.3 fuer das Model 1 noch kennt, erinnert sich sicherlich an die Frage "Lockout any Track" in der Formatierungsroutine. Das heute meist verwendete NEWDOS und seine Abkoemmlinge kennt diese Moeglichkeit den Zugriff auf verschiedene Tracks der Diskette zu verhindern nicht mehr. Man kann aber mit Superzap (wie koennte es auch anders sein) eventuell unsichere Tracks unbenutzbar machen.
Der GAT-Sector (GranulAllocationTable) des Direktorys enthaelt die Information ueber die benutzbaren und die benutzten Granules einer Diskette. Die Bytes 00H-27H enthalten die Information ueber die Ausnutzung der einzelnen Tracks. Die Bytes 60H-87H enthalten die sogenannte Lockout-Table. Ein Track ist dann unbenutzbar, wenn seine beiden repraesentativen Bytes im Gatsector den Wert FFH enthalten. An einem Beispiel erlaeutert klingt die Sache schon verstaendlicher. Nehmen wir also an, der Track 36 (relativ) einer Diskette laesst sich nicht formatieren. Der Rest der Disk funktioniert einwandfrei. Die Formatierung wird mit der Proceed-Funktion der Formatierungsroutine bis zum Ende gefuehrt. Danach schaut man sich mit Superzap den GAT-Sector (Sector 0) des Direktorys an. Man muss, um den Track 36 unbenutzbar zu machen, mit der MOD-Funktion die Bytes 23H und 83H mit dem Wert FFH ueberschreiben. Das Disk Operating System wird diesen Track nie zur Datenspeicherung hernehmen und man kann ohne Angst vor einem Datenverlust die Diskette benutzen!

** Hartmut Obermann **

Sector-Dump von BASIC/CMD aus NEWDOS80 vor den Aenderungen

```
DRV  00   D977 BEC2 C957 7022 B140 11CE FF19 22A0
1    10   4021 5C67 CD67 4421 FEFF 22A2 4021 CA64
1H   20   22A7 4021 6943 CBBE 3A6C 43CB 773E C928
     30   067C 3221 643E C332 1243 21BB 6711 5241
DRS  40   0193 00F3 EDB0 3E03 3289 5F21 BE66 3134
304  50   65FB B728 1108 ED4B C564 3600 0B23 78B1
130H 60   20F8 083D 20EF 3600 2322 A440 112C 0119
     70   EB2A B140 DFDA C957 CD4D 1B21 0000 7EFE
     80   2A20 1701 0000 2AA4 4071 2370 2100 00CD
     90   5A1B 21E9 5422 5E65 1804 FE0D 2806 2189
     A0   6522 0552 C319 1AE1 2133 0022 0552 C521
     B0   AC65 CD67 44ED 5B5E 652A A740 1AFE 0D77
FRS  C0   1323 20F8 AFF5 2BC3 7903 1D1B 1F03 0100
14   D0   C965 0000 0000 2A2A 237E FE3D C9C5 CDE1
EH   E0   6179 C1C9 ED5B C564 C9DD CB04 FEC9 C24A
     F0   1E3A 0251 B7FD 2100 4228 04FD 2117 42E5
```

```
DRV  00   DD6E 05DD 6606 FD5E 05FD 5606 DF20 04DD
1    10   CB04 F611 1700 FD19 FD7E 00B9 38E8 E1C9
1H   20   DDCB 0476 C032 024F C9DD 5E13 DDCB 0476
     30   C8F1 09C9 DDCB 0476 C019 10FD C9DD 740A
DRS  40   09EB DDCB 0476 C078 B128 042B 1BED B8AF
305  50   C900 0000 0000 0000 0000 0000 0000 0000
131H 60   0000 0000 0000 0000 0000 0000 0000 0000
     70   0000 0000 0000 0000 0000 0000 0000 0000
     80   0000 0000 0000 0000 0000 0000 0000 0000
     90   0000 0000 0000 0000 0000 0000 0000 0000
     A0   0000 0000 0000 0000 0000 0000 0000 0000
     B0   0000 0000 0000 0000 0000 0000 0000 0000
FRS  C0   0000 0000 0000 0000 0000 0000 0000 0000
15   D0   012A C766 0000 0000 0000 0000 0000 0000
FH   E0   0000 0000 0000 0000 0000 0000 0000 0000
     F0   0000 0000 0000 0000 0000 0000 0100 BC66
```

Sector-Dump von BASIC/CMD aus NEWDOS80 nach den Aenderungen

```
DRV  00   D977 BEC2 C957 7022 B140 11CE FF19 22A0
1    10   4021 5C67 CD67 4421 FEFF 22A2 4021 CA64
1H   20   22A7 4021 6943 CBBE 3A6C 43CB 773E C928
     30   067C 3221 643E C332 1243 21BB 6711 5241
DRS  40   0193 00F3 EDB0 3E03 3289 5F21 BE66 3134
314  50   65FB B728 1108 ED4B C564 3600 0B23 78B1
13AH 60   20F8 083D 20EF 3600 2322 A440 112C 0119
     70   EB2A B140 DFDA C957 CD4D 1B21 0000 7EFE
     80   2A20 1701 0000 2AA4 4071 2370 2100 00CD
     90   5A1B 21E9 5422 5E65 1804 FE0D 2806 2189
     A0   6522 0552 C319 1AE1 2133 0022 0552 C521
     B0   AC65 CD67 44ED 5B5E 652A A740 1AFE 0D77
FRS  C0   1323 20F8 AFF5 2BC3 7903 1D1B 1F03 01EA
14   D0   C965 0000 0000 2A2A 237E FE3D C9C5 CDE1
EH   E0   6179 C1C9 ED5B C564 C9DD CB04 FEC9 C24A
     F0   1E3A 0251 B7FD 2100 4228 04FD 2117 42E5
```

```
DRV  00   DD6E 05DD 6606 FD5E 05FD 5606 DF20 04DD
1    10   CB04 F611 1700 FD19 FD7E 00B9 38E8 E1C9
1H   20   DDCB 0476 C032 024F C9DD 5E13 DDCB 0476
     30   C8F1 09C9 DDCB 0476 C019 10FD C9DD 740A
DRS  40   09EB DDCB 0476 C078 B128 042B 1BED B8AF
315  50   C900 E37C FE1D E3C2 781D D7FE 90C2 791D
13BH 60   CD5B 1EE5 CD2C 1B0B ED43 FF40 E1C1 C31E
     70   1D00 0000 0000 0000 0000 0000 0000 0000
     80   0000 0000 0000 0000 0000 0000 0000 0000
     90   0000 0000 0000 0000 0000 0000 0000 0000
     A0   0000 0000 0000 0000 0000 0000 0000 0000
     B0   0000 0000 0000 0000 0000 0000 0000 0000
FRS  C0   0000 0000 0000 0000 0000 0104 0440 4966
15   D0   012A C766 0000 0000 0000 0000 0000 0000
FH   E0   0000 0000 0000 0000 0000 0000 0000 0000
     F0   0000 0000 0000 0000 0000 0000 0100 BC66
```

Geaenderte Bytes unterstrichen !!!

* Hartmut Obermann *

Schnelle Hardcopy fuer TRS-80

Unter diesem Titel erschien das folgende, sehr kurze BASIC-Programm nach einer Frage in der Computer Persoenlich 24/84. Der Verfasser ist P. Wollschlaeger.
Das Programm ermoeglicht den Ausdruck des Bildschirms innerhalb kuerzester Zeit. Mich hat erstaunt, dass auch die Grafik bei dieser Art der Hardcopy sehr gut wegkommt.
Die Zeilen 10-1070 stellen das in der CP abgedruckte Programm dar.
Die Zeile 1 zeigt das Programm nach der Behandlung mit PACKER.

```
1 I=I:V=V:V$="":V=VARPTR(V$):POKEV,64:FORI=15360TO16320
  STEP64:POKEV+1,IAND255:POKEV+2,INT(I/255):LPRINTV$:NEXT

  10 'TRS 80, MODEL I/III, VG : SCREENDUMP
  20 GOSUB 1000
  30 END
1000 I=I:V=V:V$=""
1010 V=VARPTR(V$)
1020 POKEV,64
1030 FORI=15360TO16320STEP64
1040 POKEV+1,IAND255
1050 POKEV+2,INT(I/255)
1060 LPRINTV$
1070 NEXT
```

Zeilennummer 00000

Die unten abgedruckten sieben BASIC-Zeilen entstammen einer 80-US, die ich mir beim Josef Konrad ausgeliehen habe. Es ermoeglicht, die Zeilennummern eines BASIC-Programms alle zu 0 zu machen. Solange ein Programm keine Spruenge (GOSUB, GOTO, RESUME, ON ERROR GOTO usw.) enthaelt, hat das keinen Einfluss auf den Programmablauf. Treten Spruenge in einem Programm auf, muss die aufrufende und die aufgerufene Programmzeile erhalten bleiben, die Zeilen dazwischen koennen zu 0 gemacht werden.
Das Programm wird mit RUN 65300 aufgerufen. Es zeigt die jeweilige Zeilennummer an und fragt, ob diese gerettet werden muss. Man muss sich also, bevor man ein Programm "behandelt", darueber im klaren sein, welche Zeilen stehen bleiben muessen.
Vielleicht faellt einem ja ein Trick ein, damit das Programm die Entscheidung darueber, ob eine Programmzeile stehen bleiben muss oder nicht, selbst faellt!?! Die Sache funktioniert uebrigens auch, wenn man ein BASIC-Programm als ASCII-File abspeichert und die Aenderung der Zeilennummern mit einem Textverarbeitungsprogramm vornimmt.

```
65300 P=PEEK(&H40A4)+PEEK(&H40A5)*256
65305 IFP)32768THENP=P-65536
65310 P1=PEEK(P)+PEEK(P+1)*256
65320 IF PEEK(P+2)+PEEK(P+3)*256=65300THENEND
65330 PRINT"SAVE LINE # ";PEEK(P+2)+PEEK(P+3)*256;
65340 INPUTZ$:IFZ$="Y"THENZ$="":P=P1:GOTO65305
65350 POKEP+2,0:POKEP+3,0:P=P1:GOTO65305
```

In eigener Sache

Ganz kurz moechte ich hier in eigener Sache ein paar Worte an euch richten. Ich bringe nun zum dritten Mal einen Beitrag in einem Clubinfo und frage mich oftmals, wie diese Beitraege bei euch ankommen. Ob das, was ich bringe, neu und interessant fuer euch ist oder ob man es als "olle Kamellin" handelt. Ob meine Formulierungen bei manchen, vielleicht etwas schwierigeren Problemen verstaendlich sind oder ob ich im 7. Programmierer- himmel schwebe und keiner mich versteht.
Ich moechte jetzt nicht noch eine Fragebogenaktion starten und damit mit Kanonen auf Spatzen schiessen. Aber ich moechte meine Bereitschaft kundtun jegliche Kritik und jeden konstruktiven Vorschlag der an mich herangetragen wird zu beachten. Ich freue mich ueber jede Anregung und verspreche jede an mich gestellte Frage zu beantworten (soweit ich eine Antwort weis!!!).

Damit Schluss fuer heute.
Da ich das Auslieferungsdatum des Infos noch nicht kenne, wuensche ich euch an dieser Stelle ein froehliches Weinachtsfest und ein glueckliches neues Jahr!!!

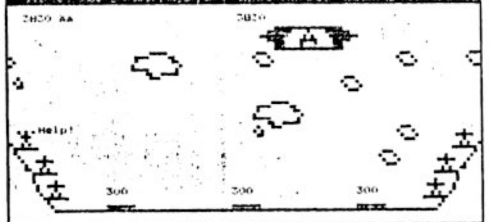
** Hartmut Obermann **

Den Computer besser nutzen:

Bildschirmausdruck ohne Probleme beim TRS/80 + Video Genie

Der TRS-80 enthält in seinen ROM's leider kein Programm zum Ausdrucken des Bildschirms. Ein solches Programm ist jedoch recht zweckmäßig, deswegen gibt es schon viele Programme, z. B. JKL beim NEWDOS/80, mit denen dieser Fehler wieder gutgemacht wird. Möchte man jedoch z. B. mitten in einem Spiel ein Schachfeld oder die Highscores für immer festhalten, versagen auch diese Programme. Das liegt daran, daß sie als eine Interruptroutine arbeiten, welche von den meisten Maschinensprachprogrammen abgeschaltet oder geändert wird. Das hier vorgestellte Programm kann jedoch immer den Bildschirminhalt ausdrucken, egal, welches Programm gerade läuft. Es ist für den EPSON MX-80 mit Graftrax geschrieben, kann jedoch leicht an andere Drucker mit TRS-80-Blockgrafik angepaßt werden.

```
DRV 00  3E1B CD38 423E 40CD 3842 3E1B CD38 423E
1   10  3ACD 3842 2100 3C06 407E FEB0 3802 CBB7
1H  20  FE20 3002 CBF7 CD38 4223 10ED 3E0D CD38
    30  423E 40BC 20E1 18FE 32EB 373A EB37 E6F0
DRS 40  FE30 20F7 C900 0000 0000 0000 0000 0000
0   50  0000 0000 0000 0000 0000 0000 0000 0000
0H  60  0000 0000 0000 0000 0000 0000 0000 0000
    70  0000 0000 0000 0000 0000 0000 0000 0000
TRK 80  0000 0000 0000 0000 0000 0000 0000 0000
0   90  0000 0000 0000 0000 0000 0000 0000 0000
0H  A0  0000 0000 0000 0000 0000 0000 0000 0000
    B0  0000 0000 0000 0000 0000 0000 0000 0000
TRS C0  0000 0000 0000 0000 0000 0000 0000 0000
0   D0  0000 0000 0000 0000 0000 0000 0000 0000
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000
    F0  0000 0000 0000 0000 0000 0000 0000 0000
```




```
               00010 ;*****************************
               00020 ;****** C.Ueberschaer ******
               00030 ;****** Lange Str. 20 ******
               00040 ;****** 2949 Hooksiel ******
               00050 ;*****************************
               00060 ;
4200           00100          ORG     4200H
               00110 ;MX-80/Graftrax wird auf TRS-80 Grafik eingestellt
4200 3E1B      00120          LD      A,1BH
4202 CD3842    00130          CALL    PRINT
4205 3E40      00140          LD      A,40H          ;Drucker-Reset
4207 CD3842    00150          CALL    PRINT
420A 3E1B      00160          LD      A,1BH
420C CD3842    00170          CALL    PRINT
420F 3E3A      00180          LD      A,3AH          ;TRS-80 Grafik
4211 CD3842    00190          CALL    PRINT
               00200 ;Bildschirminhalt zum Drucker
4214 21003C    00210          LD      HL,3C00H       ;Anfang Bildschirm
4217 0640      00220 NEWLIN   LD      B,40H          ;64 Zeichen / Zeile
4219 7E        00230 NEXT     LD      A,(HL)
421A FEB0      00240          CP      0B0H           ;Grafik
421C 3802      00250          JR      C,OK1
421E CBB7      00260          RES     6,A
4220 FE20      00270 OK1      CP      20H            ;Grossbuchstaben
4222 3002      00280          JR      NC,OK2
4224 CBF7      00290          SET     6,A
4226 CD3842    00300 OK2      CALL    PRINT
4229 23        00310          INC     HL
422A 10ED      00320          DJNZ    NEXT
422C 3E0D      00330          LD      A,0DH          ;Newline
422E CD3842    00340          CALL    PRINT
4231 3E40      00350          LD      A,40H          ;Ende Bildschirm
4233 BC        00360          CP      H
4234 20E1      00370          JR      NZ,NEWLIN
4236 18FE      00380 READY    JR      READY          ;Endlosschleife
4238 32EB37    00390 PRINT    LD      (37E8H),A
423B 3AE837    00400 TEST     LD      A,(37E8H)
423E E6F0      00410          AND     0F0H
4240 FE30      00420          CP      30H
4242 20F7      00430          JR      NZ,TEST        ;Drucker Ready?
4244 C9        00440          RET
4200           00450          END     4200H
00000 TOTAL ERRORS
34161  TEXT AREA BYTES LEFT
```

Oben: Adressen des Programms
Mitte: Beispiel
Unten: Listing des Programms

Es wird direkt auf die Diskette geschrieben, ist aber zum besseren Verständnis auch als Source-Code abgedruckt.
Zuerst muß eine Diskette formatiert werden. Wichtig ist, daß Spur 0 in Single Density formatiert wird. Dann wird Sektor 0 mit Superzap oder ähnlichem Utility-Programm wie folgt geändert:
Das Laufwerk, auf dem geändert wird, muß auf Single Density eingestellt sein.
Zum Gebrauch wird diese Diskette in Laufwerk 0 eingelegt, während ein Maschinensprachprogramm läuft. Soll zu einem bestimmten Zeitpunkt der Bildschirminhalt ausgedruckt werden, wird einfach auf den RESET-Knopf gedrückt.
Zum Programm: bei einem RESET wird beim TRS-80 bei angeschlossener Floppy Sektor 0 der in Laufwerk 0 liegenden Diskette in den Speicherbereich 4200H bis 42FFH geschrieben und anschließend nach 4200H gesprungen. Auf unserer Diskette steht in Sektor 0 das Programm zum Ausdrucken des Bildschirms, welches sofort startet, ohne daß der Bildschirm zuvor geändert wird. Bedingt durch den TRS-80-ROM kann nicht in das Maschinensprachprogramm zurückgesprungen werden, da der Stackpointer bei einem RESET auf 407DH geändert wird. Nach dem Bildschirmausdruck wartet der TRS-80 in einer Endlosschleife auf erneutes Drücken der RESET-Taste. Die geänderte Diskette kann übrigens weiterhin als Datendiskette benutzt werden!

Christof Ueberschaar

Willkommen in der ADVENTURE-ECKE!

Heute stellt Euch der Gerald Schröder das Adventure GOLDEN BATON vor. Ich verabschiede mich bis zum nächsten Club-Info, Euer

Adventure - Alex

Bahn frei
füüüüüüüüüür
Gerald !!!

Golden Baton

Die Situation und der Sinn des Spiels lassen sich aus dem Anfangstext ersehen.

Die Schauplätze
Jeder Platz erhält eine Nummer. Die möglichen Ausgänge werden den Nummern der Lokalitäten zugeordnet, zu denen sie führen.

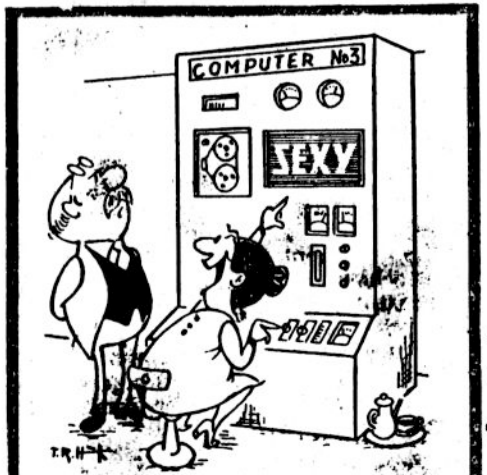
1. Dense Forest: N=3, S=2
2. Tangle of briars: N=1
3. At a small stream: N=7, S=1, W=4
4. Old Tree: N=5, E=3
5. Hut: S=4, Tür=6
6. Inside hut: W=5, D=24
7. Road 1: S=3, N=8
8. Road 2: S=7, N=92
9. Road 3: S=8, N=10
10. Pond: S=9, N=11
11. Moat: S=10, (12)
12. Portcullis: (13), (12)
13. Battlements: D=14
14. Courtyard: U=13, N=15, E=16, W=17
15. Huge oaken door: S=14, N=23
16. Stables: W=14
17. Outbuilding: E=14, D=18
18. Smelly chamber: E=19, U=17
19. Torture chamber: W=18, N=22, S=20
20. Sorcerer's laboratory: N=19, E=21
21. Bare room: W=20
22. Servants quarters: S=19
23. Cold room: S=15
24. Small cave: U=6, N=28, S=25
25. Larger cave: N=24, S=26
26. Empty room: N=25, S=27
27. Storage chamber: N=26
28. Shore: S=24 (29)
29. Lake.

Im folgenden werden jeweils die Nummer des Schauplatzes und die zu erfolgenden Aktionen aufgeführt. Dabei ist nicht berücksichtigt, wieviel getragen werden kann.

1. get cloak, wear cloak, search leaves, get sword
2. cut briars, get rope
4. throw rope, climb rope, get ring, wear ring, d, get rope
7. kill wolf
8. get staff
11. throw matches, swim
12. get matches, throw rope, climb rope
16. get helmet, wear helmet, read runes
17. get lamp, search straw
18. open door
19. light lamp, get hammer
20. wave staff, say akyrz, get quarz
21. wave quarz, search lizard, get knife
22. get mirror
15. rub ring, get key, unlock door, hold mirror, open door. N
23. get parchment, read parchment
16. get horn
5. open door
6. open barrel, get salt
25. throw salt, get slugs
26. smash padlock
27. get raft
28. drop slugs, drop raft, sail raft
29. blow horn, throw knife, get baton

ADVENTURE-ECKE

Frage:
„Wie macht ein Schotte Gruppensex?“
„Mit seiner Frau vor dem Spiegel.“



„Toller Kasten. Schauen Sie Chef, das ergeben meine Daten.“

„Hat Dir Mc. Gregor bei sich zu Hause etwas vorgesetzt?“
„Ja, einen Aschenbecher!“

Verdoppelung der Taktfrequenz für Genie I + II

Für verhältnismäßig wenig Geld läßt sich die Taktfrequenz des Videogenies verdoppeln. Notwendig sind nur eine neue CPU Z80A und ein Umschalter (1 X um).
Vorrausetzung für diese Hardware-Änderung ist allerdings, daß die RAM IC's keine größere Zugriffszeit als 300 ns (bzw. 200 ns für RAM IC's im Expander) haben. Mit den neuen 64 K Chips dürfte es keine Probleme geben.

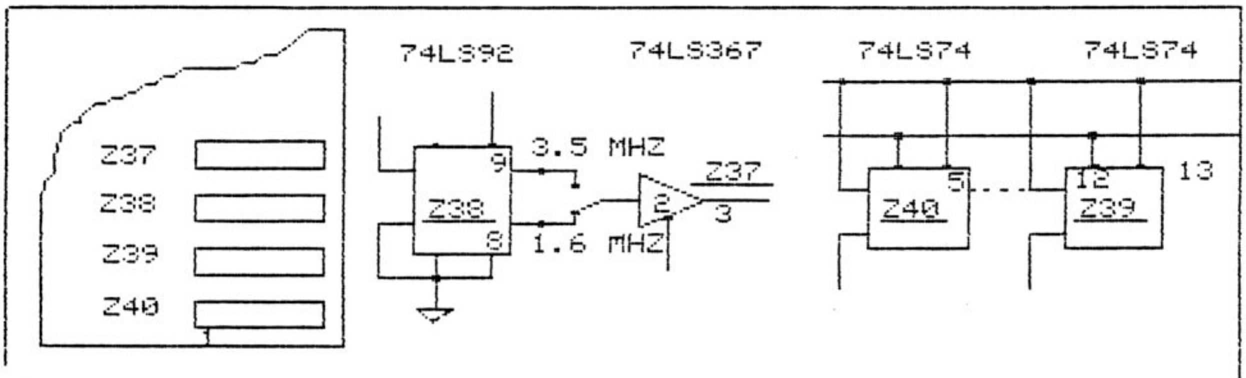


ABB.1 CPU-BOARD ABB.2 AENDERUNG 1 ABB.3 AENDERUNG 2

Im Einzelnen sind folgende Handgriffe zu verrichten:

1. Austausch des Z80 Prozessors gegen einen Z80A Prozessor.
2. Einbau des Umschalters ins Tastaturgehäuse.
3. Unterbrechung der Verbindung zwischen Pin 8 von Z38 und Pin 2 von Z37 (auf der Unterseite der CPU-Platine). Pin 2 von Z37 wird über den Umschalter (Abb.2) an Pin 8 und 9 von Z38 angeschlossen.
4. Unterbrechung der Verbindung zwischen Pin 5 von Z40 und Pin 12 von Z39 (auf der Oberseite der CPU-Platine, in der Nähe von Pin 5).
5. Pin 12 und 13 von Z39 werden durch eine Drahtbrücke verbunden (Abb.3).

Abb.1 zeigt die Lage von Z37 - Z40 auf der CPU-Platine (Aufsicht). Pin 1 befindet sich jeweils unten links am IC.

Am besten testet man den Computer erst einmal unter der alten Taktfrequenz (z.B. mit einem Diagnoseprogramm). Ist dieser Test erfolgreich verlaufen, wiederholt man die gleiche Prozedur mit der doppelte Taktfrequenz. Erforderlich hierzu ist eine Anpassung des Betriebssystemes an die doppelte Taktrate. Unter NEWDOS80 z.B. ändert man den Systemparameter BJ von 1 auf 2.
Ein Nachteil gegenüber den käuflichen 'SPEED UPS' läßt sich bei dieser Billiglösung nicht vermeiden, ein Umschalten der Taktrate bei eingeschaltetem Gerät führt meist zum "Hängen" des Computers. Dann hilft nur noch Ausschalten (incl. Programmverlust).
Zum Schluß noch ein Hinweis für eiserne Sparer. Beim Verfasser läuft ein Genie I auch mit alter CPU unter doppelter Taktfrequenz einwandfrei.

G.Dreyer

Neues aus der Gerüchte-Küche:

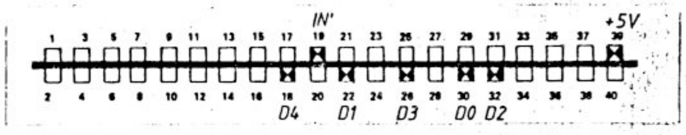
Wundert Euch nicht, wenn Ihr nächstens in Euer Tandy Computer-Center reinschaut und dann plötzlich 'apricot'-Computer seht. Angeblich werden die Tandy-Computer-Center von Tandy abgezwackt und heißen dann 'TA-Computerworld' und vertreiben außer Tandy dann die aus England stammenden Computer der Firma 'apricot'. Dabei soll es sich um leistungsstarke und sehr preiswerte MS-DOS-Computer handeln.

JOYSTICK für den TRS-80 Modell I

Anschluß des Joystick :
Der Joystick benötigt folgende TRS-80 Anschlüsse :

a) IN´ (Pin 19 des TRS-80-Bus) ==) gemeinsamer Anschluß des Joystick auf Peripherie-Lese-Taktsignal.
b) D0 (Pin 30 des TRS-80-Bus) ==) hoch- ,vor Anschluß des Joystick auf Datenleitung 0.
c) D1 (Pin 22 des TRS-80-Bus) ==) runter- ,zurück Anschluß des Joystick auf Datenleitung 1.
d) D2 (Pin 32 des TRS-80-Bus) ==) links Anschluß des Joystick auf Datenleitung 2.
e) D3 (Pin 26 des TRS-80-Bus) ==) rechts Anschluß des Joystick auf Datenleitung 3.
f) D4 (Pin 18 des TRS-80-Bus) ==) Feuer- Anschluß des Joystick auf Datenleitung 4.
g) + (Pin 39 des TRS-80-Bus) ==) Dauerfeuer- Anschluß des Joystick auf Plus 5V. (gegebenenfalls)

Die entsprechenden Joystickanschlüsse sind mit dem Ohmmeter und Betätigung des Joystick leicht auszumessen.



Numerierung der Bus-Anschlußleitungen (von der Rückseite des Computergehäuses aus gesehen).

Es gibt nun zwei Möglichkeiten, einen Joystick an den TRS-80 anzuschließen.

1. Ohne große Bastelei (zum Probieren)
Man besorge sich einen 40-poligen Busstecker für die TRS-80-Platine bzw. für das EXPANSION. Sowie fünf Dioden, die zur Entkopplung im Joystick in die Leitungen b - f eingelötet werden. Die Dioden müssen mit der Kathode (schwarzer Ring / - Pol) zur Joystick-Platine eingelötet werden. Danach sind die Drähte a - f bzw. a - g -je nach Joystick- auf den entsprechenden Kontakt des Bussteckers (wie oben angegeben) zu verpressen.

2. Wer seinen TRS-80-Bus bzw. EXPANSIONS-Bus benötigt
In diesem Fall braucht man einen 7-poligen Diodenstecker und eine passende Diodenbuchse anstatt des TRS-80 Bussteckers. Anschließend wird der TRS-80 oder das EXPANSION vorsichtig geöffnet, und auf der Hauptplatine werden vor dem 40-poligen Bus sieben Drähte auf die oben angegebenen Punkte aufgelötet. Es ist zu beachten, daß die Folienverbindung zwischen Haupt- und Tastaturplatine nicht beschädigt wird! Nun sucht man sich am Computergehäuse einen günstigen Platz zum Einbau der Diodenbuchse. Nachdem die Buchse eingebaut ist und an die Anschlüsse derselben die fünf Dioden (Kathode zur Buchse) angelötet sind, können die sieben Drähte von der Hauptplatine aufgelötet werden. Es ist darauf zu achten, daß die Belegung des Diodensteckers mit der der Diodenbuchse identisch ist. Die Verbindungen sind zu überprüfen. Bei dieser Einbauvariante kann der Joystick so belassen werden wie er ist.
Es besteht auch die Möglichkeit, den Original-Joystickstecker zu belassen und sich eine entsprechende Buchse dafür zu kaufen und einzubauen.

Der Einbau des Joysticks ist recht einfach. Ich möchte aber darauf hinweisen, daß ich für eventuelle Schäden, die am Gerät entstehen sollten, nicht hafte.
Für Verbesserungsvorschläge bin ich allerdings dankbar.

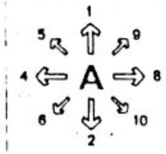
PS.: Leider ist der Joystick nicht bei HRG1B-Programmen verwendbar, da beide den PORT 0 benutzen. (Die Grafik wird bei jeder Abfrage ausgeschaltet)

Jens NEUEDER
Panoramastr. 21
7178 Michelbach /Bilz
Tel.: 0791 / 42077

Funktionstest des Joystick :
Der Joystick ist durch die Benutzung des Port 0 unabhängig von der Tastatur. Der Joystick muß demzufolge auch anders angesprochen werden. Der Befehl dazu ist INP (0).

Nun ein kurzes Programm in BASIC zum Testen :

```
 5 CLS
10 A=255-INP(0)
20 Print 50,A
30 GOTO 10
```



Nach Starten des Programms müssen auf dem Monitor je nach Betätigung des Joysticks die Zahlen -wie auf der Abbildung rechts- angezeigt werden.
Ist der Joystick in Ruhelage, erscheint eine 0.
Bei Feuerknopfdruck sollte eine 16 erkennbar sein.

Erscheinen die Zahlen entsprechend vorheriger Erklärung ist der Joystick betriebsbereit. Andernfalls sollten die Anschlüsse noch einmal auf ihre Richtigkeit überprüft werden.

Basicprogramme mit Joysticksteuerung :
Um nun die Joystickfunktionen in ein Basicprogramm einzubinden, ist folgendes zu tun.

Tastaturbetrieb	Joystickbetrieb
1000 A$=INKEY$: IFA$=""THEN1000	1000 A=255-INP(0): IFA=0THEN1000
1010 IFA$="↓"THEN GOSUB AAAA	1010 IFA= 1THEN GOSUB AAAA
1020 IFA$="↑"THEN GOSUB AABA	1020 IFA= 2THEN GOSUB AABA
1030 IFA$="→"THEN GOSUB AACA	1030 IFA= 4THEN GOSUB AACA
1040 IFA$="←"THEN GOSUB AADA	1040 IFA= 8THEN GOSUB AADA
1050 IFA$=" "THEN GOSUB AAEA	1050 IFA=16THEN GOSUB AAEA

Damit die Joystickabfrage schneller wird, sollte die entsprechende Variable als DEFINT-Variable ausgegeben werden.

Maschinenprogramme mit Joysticksteuerung :
Die schnellsten Spiele sind natürlich Programme in Maschinensprache. Hier passiert im Prinzip das gleiche wie bei den Basicprogrammen. Der Computer muß nur so programiert werden, daß er den Port 0 liest,anstatt die Tastaturabfrageroutine zu nutzen.

Die meisten Programme der Firmen BIG FIVE SOFTWARE, ADVENTURE INTERNATIONAL, CORNSOFT GROUP und ALPHA PRODUCTS sind Joystickkompatipel.

Ich wünsche viel Spaß beim Spiel.

Peter Wolf

EPROM-Programmiergerät

Mehr und mehr Anwendungen von Computern benötigen programmierte EPROMs, beispielsweise die zahlreichen Einplatinencomputer. Ein EPROM-Programmiergerät soll zum einen möglichst alle gängigen EPROM-Typen programmieren können, zum anderen an möglichst viele Computer anschließbar sein. Da in Zukunft weitere EPROMs auf den Markt kommen dürften, muß der Anwender die Änderungen der Software selbst durchführen können.

Die vorliegende Schaltung ist in der Lage, die EPROM-Typen 2716, 2732, 2732A und 2764 zu programmieren. Ziel war es, mit möglichst wenig zusätzlicher Hardware dieses Problem zu lösen. Zum Rechner hin ist eine Parallelschnittstelle erforderlich, in der vorliegenden Form sind 18 Portleitungen notwendig. Die Software ist in Basic geschrieben und zeigt den grundsätzlichen Umgang mit dem Programmiergerät. Der einzelne Anwender kann diese Software noch nach Belieben ausbauen.

Nicht viel Hardware

Die Schaltung wird im vorliegenden Fall an einem 8085-System betrieben und ist dort an einem PIO-Baustein 8255 angeschlossen. Dieser wird von Basic aus über die Befehle IN und OUT angesprochen, bei anderen Computern ist dies analog mit PEEK und POKE möglich. Die Gesamtschaltung des Programmiergerätes (*Bild 1*) ist nicht sehr aufwendig (*Bild 2*). Der PIO-Baustein 8255 besitzt drei Ports von je 8 Bit. Port A wird zur Übertragung der Daten von und zum Rechner benutzt. Port B steuert mit den Bits 0 und 1 die beiden Binärzähler IC1 und IC2. Diese erzeugen die Adressen für die EPROMs, da sonst eine erheblich größere Zahl von Portleitungen erforderlich wäre.
Port C ist für die Steuersignale zuständig. Hier werden die Programmierspannung, das Chip-Enable-Signal sowie der Programmierimpuls den jeweiligen Pins zugeführt. Das NAND-Gatter IC 3 verhindert, daß bei unsachgemäßer Programmierung (alle Leitungen des Ports C auf High) die 25-V-Versorgung kurzgeschlossen wird.
Die unterschiedlichen Spannungen zum Programmieren bzw. zum Lesen der EPROMs werden mit den Z-Dioden D3 und D4 erzeugt. D3 ist so zu wählen, daß sich eine Programmierspannung von 21 V einstellt. Da Z-Dioden Toleranzen haben, kann es sein, daß man sich mit Kombinationen von Dioden behelfen muß. Das gilt natürlich auch für D4, die mit D3 zusammen eine Spannung von < 5 V (High-Pegel) ergeben muß. In gewissen Grenzen kann man auch mit der 25-V-Versorgung nachhelfen, denn diese Spannung kann ja zwischen 24 und 26 V liegen.

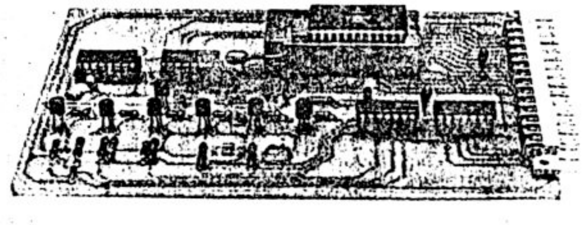


Bild 1. Der Prototyp des EPROM-Programmiergerätes

Die gesamte Schaltung ist auf einer einseitig kaschierten Leiterplatte im Format 100 × 160 mm untergebracht (*Bild 3*). Die Platine wird nach *Bild 4* bestückt. Da es sich um eine einseitige Leiterplatte handelt, lassen sich Brücken nicht ganz vermeiden, aber doppelseitige Leiterplatten kann nicht jeder so ohne weiteres herstellen und sind zudem erheblich teurer.
Haben Rechner und Programmiergerät getrennte Stromversorgungen, sollten zwei Widerstände von etwa 10 kΩ in die beiden Leitungen zum Adreßzähler geschaltet werden. Ansonsten werden die Schutzdioden des CMOS-Schaltkreises überlastet, wenn das EPROM-Programmiergerät nicht eingeschaltet ist.

Unterschiedliche EPROMs

Eine Aufstellung, wie die einzelnen EPROMs programmiert oder auch gelesen werden, zeigt *Bild 5*. In Zweifelsfällen ist es immer sinnvoll, ein Datenblatt des Herstellers zu Rate zu ziehen. Allgemein gilt aber, daß die EPROMs 2716 und 2732 mit 25 V programmiert werden, die Typen 2732A und 2764 hingegen mit 21 V. Die Länge des Programmierimpulses soll 50 ms ± 5 ms betragen. Durch das EPROM 2764 bedingt besitzt das Programmiergerät einen 28poligen Sockel. Alle übrigen Typen hingegen haben nur 24 Anschlüsse. Deshalb ist beim Einsetzen der EPROMs auf deren richtige Position zu achten. Wer nur 24polige EPROMs programmieren will, kann von vornherein einen 24poligen Sockel montieren.

Die Programmier-Software

Um das Programm (*Bild 6*) auf andere Rechner übertragen zu können, sind die spezifischen Adressen des PIO-Bausteins 8255 zu erklären. Es bedeuten im einzelnen:

2716

Modus \ Pin	$\overline{CE}$ (PGM) 18	$\overline{OE}$ 20	V_{PP} 21	Outputs 9-11/13-17
Lesen	Low	Low	+5V	Daten (aus)
Standby	High	beliebig	+5V	hochohmig
Programmieren	Impuls Low→High	High	+25V	Daten (ein)

2732 und 2732A

Modus \ Pin	$\overline{CE}$ 18	$\overline{OE}/V_{PP}$ 20	Outputs 9-11/13-17
Lesen	Low	Low	Daten (aus)
Standby	High	beliebig	hochohmig
Programmieren	Low	+25V (2732) +21V (2732A)	Daten (ein)

2764

Modus \ Pin	$\overline{CE}$ 20	$\overline{OE}$ 22	$\overline{PGM}$ 27	V_{PP} 1	Outputs 11-13/15-19
Lesen	Low	Low	High	+5V	Daten (aus)
Standby	High	beliebig	beliebig	+5V	hochohmig
Programmieren	Low	beliebig	Low	+21V	Daten (ein)

Bild 5. Eine Gegenüberstellung der Bedingungen an den Steuereingängen der verschiedenen EPROMs zum Lesen oder Programmieren. Der Zustand „beliebig" bedeutet, daß sowohl High- als auch Low-Pegel anliegen kann

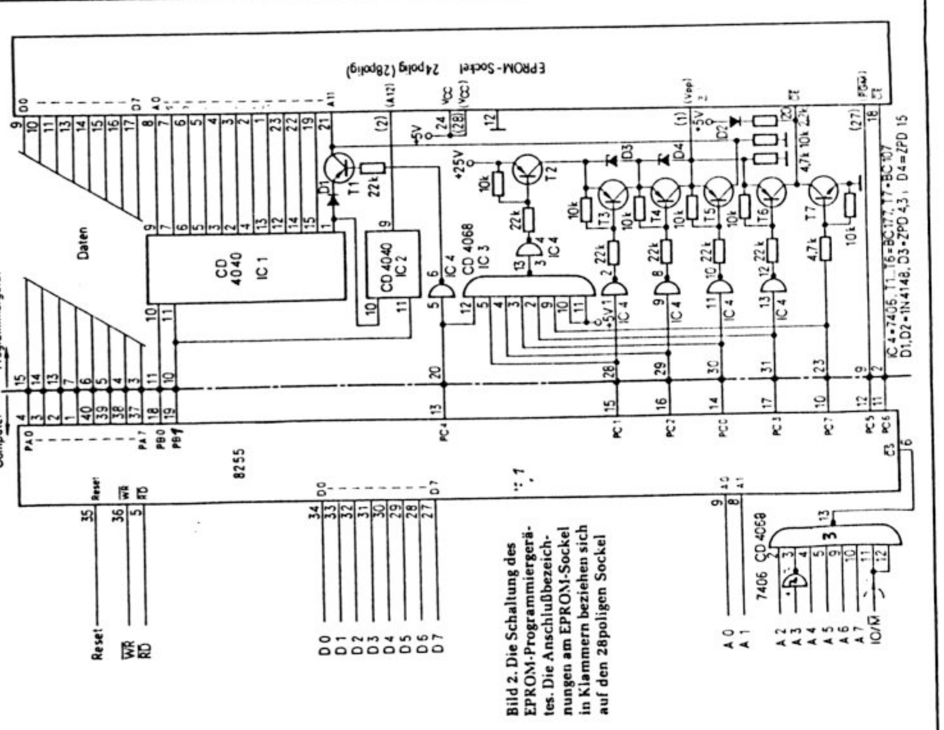


Bild 2. Die Schaltung des EPROM-Programmiergerätes. Die Anschlußbezeichnungen am EPROM-Sockel in Klammern beziehen sich auf den 28poligen Sockel

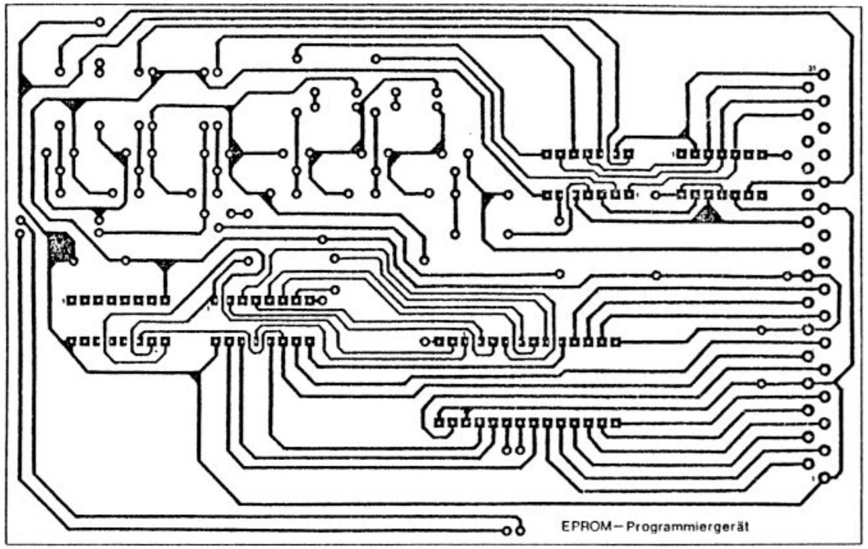


Bild 3. Die Schaltung des Programmiergerätes ist auf einer Europakarte untergebracht

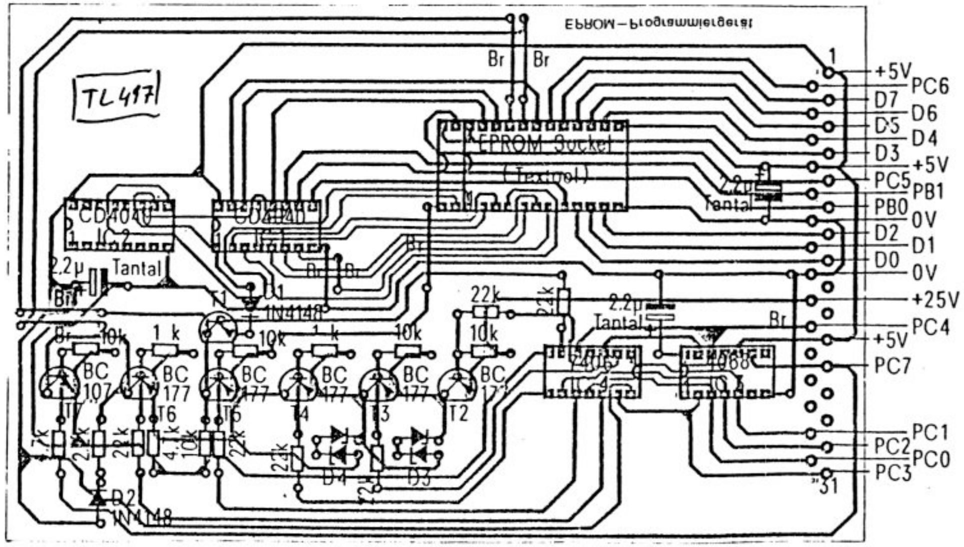
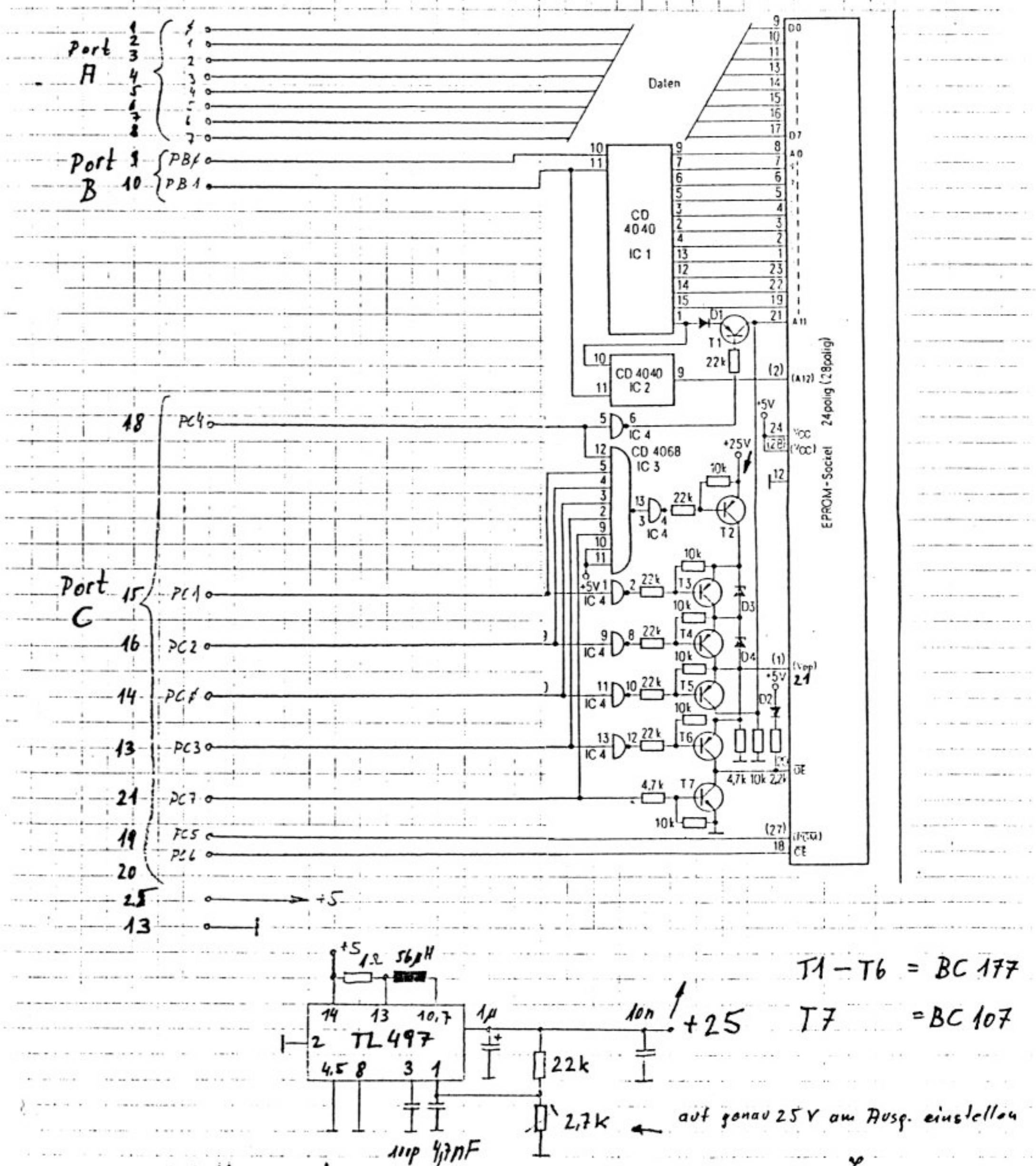


Bild 4. Die Anordnung der Bauelemente auf der Platine

EPROM - HARDWARE FÜR KOMTEK, TRS, VG

Interface OE2TZL



Die Schaltung mit dem TL 497 erzeugt aus der normalen +5V die Programmierspannung 25 V

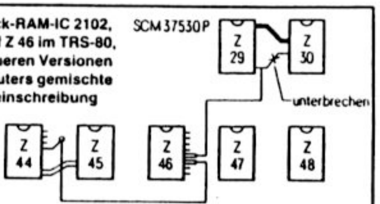
Die Bauteile dazu sind frei auf der Platine verdrahtet.

gehört an den Schluß dieses Beitrages.

Kleinbuchstaben für den TRS-80

Da ich auch einen TRS-80 besitze, habe ich den Artikel in mc 1982, Heft 12, auf Seite 61, mit großem Interesse gelesen. Bei neueren Versionen des TRS-80 mit dem Zeichengenerator SCM 37530 genügt ein einzelnes Video-RAM, um den Rechner von Groß- auf Kleinschreibung umzustellen. Zur Umschaltung kann man den Kleinbuchstaben-Treiber aus mc 1982, Heft 6, verwenden. Die Schaltung ist im *Bild* wiedergegeben. Das neue 2102-IC wird huckepack auf Z 46 gelötet, wobei die Pins 11 und 12 rechtwinklig abgeknickt werden.

Friedhelm Lüttecke, Münster



Ein Huckepack-RAM-IC 2102, aufgelötet auf Z 46 im TRS-80, macht bei neueren Versionen dieses Computers gemischte Groß- und Kleinschreibung möglich

mc 2/1983

Kleinbuchstaben-Treiber für den TRS-80

Wer in letzter Zeit einen TRS-80 Modell I gekauft hat, der hat wahrscheinlich, ohne es zu wissen, Kleinbuchstaben miterworben. Die neuen, die in Japan unter Lizenz hergestellt werden (Merkmal: am Tastaturgehäuse unten steht „MADE IN JAPAN"), werden serienmäßig mit dem Zeichen-ROM ausgestattet, das sonst inklusive Treiberprogramm in Tandy-Läden nur für rund 150 DM erhältlich ist. Nur: mit dem normalen Zeichentestprogramm des Handbuchs mit PRINT CHR$ bleiben diese Zeichen verborgen. Erst mit POKE kommen sie zum Vorschein, und zwar mit folgendem kleinen Testprogramm:

```
10 CLS:Y=15360:FOR X=0 TO 255
20 POKE Y,X:POKE Y+1,32:Y=Y+2
30 NEXT:PRINT@512,;
```

Nach RUN erscheinen die Großbuchstaben in der ersten Reihe, die normalerweise angesprochen werden. In der vierten Reihe, ab dezimal 97, kommen die Kleinbuchstaben. Es galt also, diese Kleinbuchstaben verfügbar zu machen, und zwar so, daß Großbuchstaben wie bei der Schreibmaschine mit der SHIFT-Taste zu erreichen sind; aber auch der alte Modus mit Großbuchstaben sollte nicht dabei ganz verlorengehen. Das Programm KBUCH macht's möglich.

Dieser Treiber für Kleinbuchstaben auf dem TRS-80 Modell I zeichnet sich durch einfaches Umschalten aus. Es wird das Basic-Kommando NAME, das weder mit ROM- oder mit Disk-Basic benutzt wird, verwendet, um von der TANDY-Tastatur (Großbuchstaben) zur geänderten und zurück zu gehen.

Angenommen, man hat nur 16 KByte RAM; der Treiber liegt bei hexadezimal 7FA2 bis 7FFF. Bei der Frage MEM SIZE? muß also 32673 (oder weniger) eingegeben werden. Das Programm wird von Kassette unter dem Namen KBUCH eingelesen und bei 7F92 (32658) gestartet. Hierdurch zeigt NAME auf den Treiber. Anschließend kann mit dem Aufruf NAME beliebig hin- und hergeschaltet werden.

Wirkungsweise: wenn der Treiber eingeschaltet ist, werden alle alphabetischen Zeichen von der Tastatur „umgedreht". Im Bildschirmtreiber wird die Ausgabe von Kleinbuchstaben ermöglicht.

Eine Kassettenaufzeichnung kann mit dem DEBUG-Monitor folgendermaßen erstellt werden: Zunächst wird DEBUG eingelesen, dann tippt man mit dem M-Kommando das Programm im Bild ein:

M ADDRESS = 7F92 usw.

Mit dem W-Kommando schreibt man nun auf Band:

S=7F92 E=7FFF T=7F92 N=KBUCH

Bei Systemen mit 32/48 KByte müssen die Bytes an den Adressen 7F99, 7FB0 und 7FB3 angepaßt werden; ansonsten ist das Programm frei verschiebbar.

Mary Jo Kostya

```
7F90   00 00 F5 E5 21 8F 41 11
7F98   A2 7F 73 23 72 E1 F1 C3
7FA0   19 1A F5 E5 D5 C5 21 16
7FA8   40 7E FE E3 20 13 11 C9
7FB0   7F 01 E2 7F 73 23 72 2E
7FB8   1E 71 23 70 C1 D1 E1 F1
7FC0   C9 11 E3 03 01 58 04 18
7FC8   EB CD E3 03 FE 41 D8 FE
7FD0   5B C8 FE 60 C8 FE 7B D0
7FD8   CB 6F 20 03 CB EF C9 CB
7FE0   AF C9 DD 6E 03 DD 66 04
7FE8   DA 9A 04 DD 7E 05 B7 28
7FF0   01 77 79 FE 20 DA 06 05
7FF8   FE 80 D2 A6 04 C3 7D 04
```

Das Treiberprogramm als Hex-Dump

Luidger Röckrath

Kleinbuchstaben für den TRS-80

Kleinbuchstaben sind für Textverarbeitung auf Computern notwendig, leider jedoch bei der Konzeption vieler handelsüblicher Kleincomputer (wie z. B. beim TRS-80) schlichtweg vergessen worden. Oder kam den Schöpfern dieser Kleincomputer Textverarbeitung als mögliche Anwendung überhaupt nicht in den Sinn? Besonders sträflich erscheint dieses Versäumnis beim TRS-80, da durch hinzufügen von nur zwei preisgünstigen Bausteinen (Gesamtkosten ca. 5 DM) Kleinbuchstaben implementiert werden können.

Textverarbeitung ist beim Autor eine der häufigsten und nützlichsten Anwendungen des TRS-80. Mit einem TRS-80, zwei Disk-Laufwerken, einer Kugelkopf- oder Typenradmaschine und entsprechender Software läßt sich für ein paar Tausender ein System zusammenstellen, mit dem komfortabel und schnell Briefe, Manuskripte und auch größere Arbeiten eingegeben, korrigiert, dauerhaft gesichert und in Korrespondenzqualität ausgegeben werden können. Kleinbuchstaben bilden die geringste Voraussetzung für halbwegs vernünftige Textverarbeitung. Wie die weiteren Voraussetzungen auch erfüllt werden können, dazu auch mehr im letzten Abschnitt.

Die Video-RAM-Schaltung des TRS-80

Bild 1 zeigt die Video-RAM-Schaltung des TRS-80. Das Video-RAM besteht aus 1024 Speicherplätzen, entsprechend der Anzahl der auf dem Bildschirm darstellbaren Zeichen. Aber leider bietet jeder Speicherplatz nicht Platz für die Speicherung eines Bytes, wie der Leser vielleicht annimmt, sondern nur für 7 Bit, da ja auf dem Bildschirm auch nur 128 verschiedene Zeichen dargestellt werden können. Der Speicherbaustein für Bit 6 wurde einfach weggelassen, und durch eine kleine Logikschaltung (Z30 in Bild 1) werden beim Auslesen des Video-RAM in Abhängigkeit von Bit 5 und 7 auf der Datenleitung von Bit 6 Logikzustände erzeugt, so daß man beim Auslesen des Video-RAM-Inhaltes die den auf dem Bildschirm dargestellten Zeichen entsprechenden ASCII-Codes erhält. Hierin (und nicht im Zeichengenerator!) liegt der Grund der Unmöglichkeit Kleinbuchstaben auf dem Bildschirm darzustellen. *Tabelle 1* zeigt, wie sich diese Logikschaltung auf das Verhalten des Video-RAM auswirkt. Aus ihr kann man leicht entnehmen, daß alle Zeichen jeweils durch zwei verschiedene Bitmuster auf den Bildschirm gebracht werden können. Beim Auslesen erscheinen allerdings immer die ASCII-Codes der dargestellten Zeichen – eine Tatsache, die auch in der ROM-Software des TRS-80 ihren Niederschlag findet, und uns weiter unten noch beschäftigen wird. (Neueste Modelle aus japanischer Produktion, die mit Kleinschreibung ausgerüstet sind, verhalten sich in dieser Beziehung anders.)

Die modifizierte Video-RAM-Schaltung

Der einfachste – und folglich auch meist beschrittene – Weg zur Kleinschreibung besteht in der schlichten Ergänzung des fehlenden RAM-Bausteins für Bit 6.

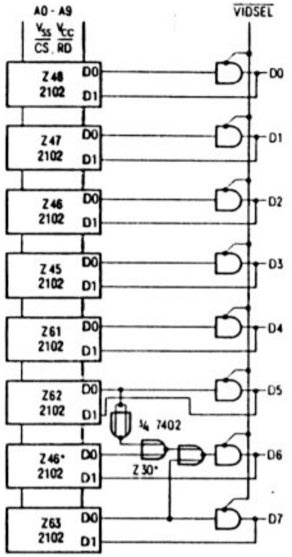


Bild 1. Die Schaltung des Video-RAM des TRS-80. Da der Baustein für Bit 6 fehlt, ist die Darstellung von Kleinbuchstaben auf dem Bildschirm nicht möglich

Aber dieser Weg führt wegen gewisser Eigenschaften des Displaytreibers im ROM, dessen Anfang in *Bild 2* als Auszug aus [1] aufgelistet ist, zu auf den ersten Blick recht eigenwilligen Ergebnissen. Es drängt sich fast der Verdacht auf, daß bei der Entwicklung des TRS-80 (hier Displaytreiber) Wert darauf gelegt wurde, durch „geschickte" Programmierung Erweiterungen (hier Kleinschreibung) möglichst schwer zu machen. Wie am Schluß des letzten Abschnittes schon erläutert, können alle Zeichen durch zwei verschiedene Codes erzeugt

Tabelle 1: Zeichencodierung im TRS-80

schreiben	im Speicher	auf Bildschirm	lesen
765	765	← Bit-Nr. →	765
000	0X0	Großbuchstaben	010
001	0X1	Ziffern, Sonderz.	001
010	0X0	Großbuchstaben	010
011	0X1	Kleinbuchstaben	011
100	1X0	Grafik	100
101	1X1	Grafik	101
110	1X0	Grafik	100
111	1X1	Grafik	101

werden. Aus unbegreiflichen Gründen (welche unter dem Blickwinkel obiger Argumentation wiederum auch recht begreiflich sind) hat man bei Radio Shack für die Buchstaben den Code gewählt, der nicht dem Standard-ASCII-Code entspricht, sondern demselben mit Bit 6 = 0. Den Erfolg dieser Strategie hat man nach dem Umbau auf der Hand bzw. auf dem Bildschirm: Statt dem gewohnten „MEMORY SIZE?" erscheinen beim Einschalten allerlei merkwürdige Zeichen, und selbst derjenige, der glaubte seinen TRS-80 zu kennen, ist erstaunt ob der ungeahnten Möglichkeiten grafischer Darstellung, die der TRS-80 noch in sich birgt.
Durch die Hinzufügung des zusätzlichen RAM-Bausteins werden beim Auslesen die Codes für Buchstaben nicht mehr in den ASCII-Code umgewandelt, sondern in der Form, wie sie vom Displaytreiber eingeschrieben wurden, an den Zeichengenerator weitergegeben. Und der erzeugt dann diese interessanten Grafikzeichen. Was tun? Als Lösungsmöglichkeiten bieten sich hard- und softwaremäßige Umschaltung an, die den zusätzlichen RAM-Baustein erst zuschalten, wenn ein neuer Displaytreiber geladen ist. Oder noch umständlicher: Blindes Laden des neuen Displaytreibers, der die rechten „ASCII-Verhältnisse" wieder herstellt.
Die Lösung, die in diesem Artikel vorgestellt werden soll, geht radikalere Wege. Auf die Hieroglyphen, die gerade noch auf dem Bildschirm erschienen, müssen wir dabei allerdings verzichten. Dafür erhält man aber normales Verhalten beim Einschalten mit dem Displaytreiber im ROM und Kleinschreibung nach Laden eines neuen Treibers.
Und wie funktioniert diese Lösung? Durch eine einfache Logikschaltung (*Bild 3*) werden beim Auslesen des Video-RAM alle Inhalte im Bereich von 00-1FH nach 40-5FH transferiert. Das heißt, es können weiterhin Buchstaben im „Tandy-Code" eingeschrieben werden, welche dann doch als Buchstaben sichtbar werden.
Tabelle 2 zeigt das Verhalten der so modifizierten Schaltung.

```
0458 DD 6E 03     LD     L,(IX+03)   ;CURSORADRESSE LADEN
0458 DD 66 04     LD     H,(IX+04)   ;NACH HL
045E 38 3A        JR     C,049A      ;CY? --> SPRUNG
0460 DD 7E 05     LD     A,(IX+05)   ;CURSOR ON?
0463 B7           OR     A
0464 28 01        JR     Z,0467      ;NEIN
0466 77           LD     (HL),A      ;JA, ALTES ZEICHEN AUF CURSORPOSITION
0467 79           LD     A,C         ;ZEICHEN NACH A
0468 FE 20        CP     20          ;CONTROLCODE?
046A DA 06 05     JP     C,0506      ;JA
046D FE 80        CP     80          ;GRAPHIC ODER SPACE COMPRESSION?
046F 30 35        JR     NC,04A6     ;JA
0471 FE 40        CP     40          ;BUCHSTABE?
0473 38 08        JR     C,047D      ;NEIN
0475 D6 40        SUB    40          ;NACH 0-3F TRANSFORMIEREN
0477 FE 20        CP     20          ;LOWER CASE?
0479 38 02        JR     C,047D      ;NEIN
047B D6 20        SUB    20          ;IN UPPER CASE UMWANDELN
047D CD 41 05     CALL   0541        ;IN VIDEORAM UND SCROLL, WENN NOETIG
```

Bild 2. Der Displaytreiber des TRS-80, dessen Anfang hier aufgelistet ist, hat die Eigenschaft, für Buchstaben nicht ASCII ins Video-RAM einzuschreiben, was den Umbau auf Kleinbuchstaben unnötig erschwert

Der Einbau

Nun endlich zur alles entscheidenden Operation, dem Eingriff ins Herz des Computers. Händler reagieren auf solche Eingriffe zwar meist mit Verlust der Garantie, aber das soll uns nicht stören, insbesondere wenn wir den Rechner schon länger als 6 Monate unser eigen nennen können. Außerdem kann mit folgender Anleitung sowie ein wenig Kenntnis im Umgang mit Schraubenzieher und Lötkolben kaum etwas schiefgehen. Also frisch ans Werk!

Öffnen des Gehäuses

Alle hierzu erforderlichen Schritte sind im folgenden zusammengestellt:
a) Alle Kabelverbindungen von der Tastatur lösen und die Tastatur auf den Arbeitstisch legen.
b) Rechner umdrehen und mit einem Schraubenzieher die 6 Schrauben entfernen. Man bemerke hierbei die verschiedenen Längen der Schrauben und deren Korrelation zur Gehäusedicke, und bewahre die so gewonnene Kenntnis bis zum Zusammenbau.
c) Nun wird die Tastatur vorsichtig umgedreht. (Man beachte, daß nach Entnahme der Schrauben nicht mehr viel Zusammenhalt vorhanden ist.)
d) Deckel abnehmen. Bei älteren Modellen ist hierbei zu beachten, daß die Betriebsanzeige-LED nicht immer auf der Tastaturplatine montiert ist, sondern direkt im Deckel und mit zwei Litzen mit der Tastaturplatine verbunden ist. Falls dies der Fall ist, löse man die LED vorsichtig aus dem Deckel.
e) Tastaturplatine nach vorne wegklappen. Bei diesem und den folgenden Schritten ist darauf zu achten, daß das Verbindungskabel zwischen Haupt- und Tastaturplatine nicht belastet wird. (Durch Mißachtung dieser Vorsichtsmaßnahme kam der Verfasser zu dem zweifelhaften Vergnügen, dieses Verbindungskabel wegen Kabelbruchs austauschen zu dürfen.)
f) Die 6 Abstandhalter entfernen und den Boden des Gehäuses nach vorsichtigem Anheben der Hauptplatine entfernen.
g) Haupt- und Tastaturplatine umdrehen (Vorsichtsmaßnahme aus e) beachten), so daß Bestückungsseite oben zu liegen kommt.

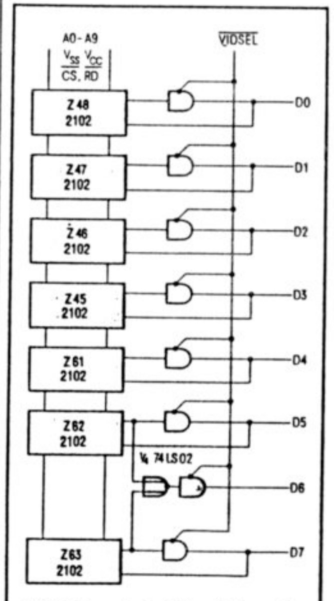


Bild 3. So muß die Videoschaltung des TRS-80 modifiziert werden, um die Darstellung von Kleinbuchstaben auf dem Bildschirm zu ermöglichen

Tabelle 2: Modifizierte Zeichencodierung

schreiben	im Speicher	auf Bildschirm	lesen
765	765	← Bit-Nr. →	765
000	000	Großbuchstaben	010
001	001	Ziffern, Sonderz.	001
010	010	Großbuchstaben	010
011	011	Ziffern, Sonderz.	001
100	100	Grafik	100
101	101	Grafik	101
110	110	Grafik	100
111	111	Grafik	101

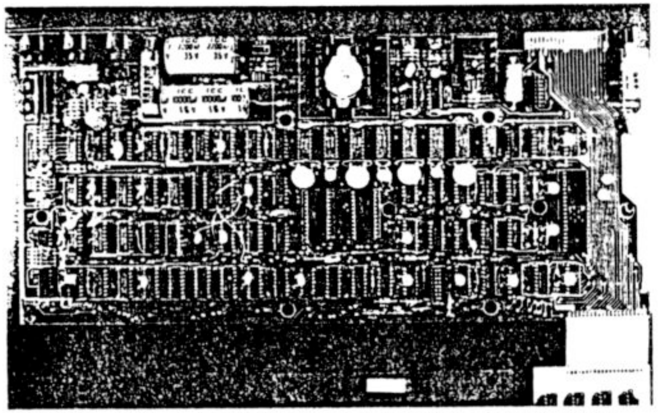


Bild 4. Die Hauptplatine des TRS-80. Die modifizierte Videoschaltung ist an den zahlreichen Drahtverbindungen zu erkennen

Der Umbau des TRS-80

Nach der nun hoffentlich erfolgreich vollendeten Demontage folgt der eigentliche Eingriff. Dazu legt man die beiden Platinen mit dem benötigten Werkzeug zurecht. Auf der in *Bild 4* dargestellten Hauptplatine ist der entscheidende Bereich eingerahmt. Man orte diesen Bereich auf der Platine und nehme dann die Skizze *Bild 5* zur Hand. Anhand dieser Skizze, dem Foto in *Bild 6*, welches den gleichen Ausschnitt zeigt, und folgender Anleitung, ist der Umbau schnell und problemlos vollzogen. Man suche die Video-RAM-Bausteine Z45-48 und Z61-63. Zumindest einer von ihnen ist auf einen Sockel montiert. Wenn nicht (was bei sehr frühen Modellen vorkommen soll), nehme man entweder eine Entlötpumpe zur Hand und entferne damit Z46 aus der Platine oder vergesse den ganzen Umbau und baue den TRS-80 wieder zusammen. Diesen auf einen Sockel montierten Baustein (im folgenden sei angenommen, es handle sich um Z46) hebe man mit einem kleinen Schraubenzieher vorsichtig aus der Fassung und stecke ihn sofort in leitenden Kunststoff. Unter diesen Vorsichtsmaßnahmen sind kaum Probleme beim Umgang mit diesem MOS-IC zu befürchten. Zuvor hat man sich einen Baustein gleichen Typs besorgt (2102), von dem die Anschlüsse 11 und 12 vorsichtig waagerecht abgebogen werden.
Der so vorbereitete Baustein wird nun Huckepack auf den herausgenommenen Z46 gesteckt, wobei darauf zu achten ist das alle Pins übereinstimmen (Pin-1-Markierung!). Nun werden alle Anschlüsse außer Pin 11 und 12 miteinander verlötet. Wenn man nun den Baustein in den Sockel von Z46 zurücksteckt, ist der schwierigste Teil der Operation schon vollbracht. Nun orte man die Leiterbahn zwischen Z29 und Z30 und trenne sie mit einem scharfen Messer an der in Bild 5 bezeichneten Stelle auf. Jetzt nehme man den zweiten Baustein (74LS02) und biege alle Pins bis

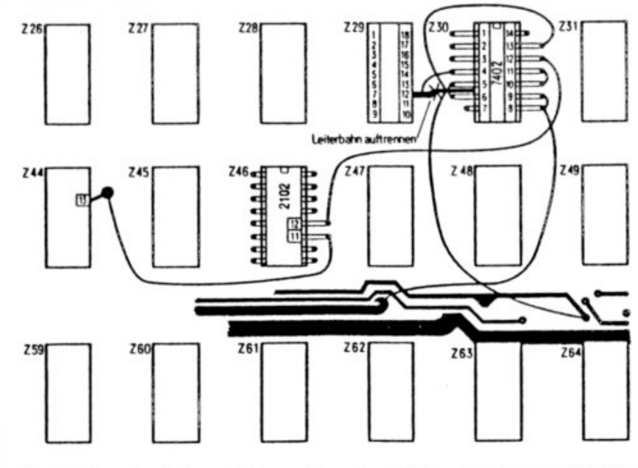


Bild 5. Werden anhand dieser Zeichnung die zwei zusätzlichen Bausteine eingebaut und die notwendigen Drahtverbindungen gelegt, können auf dem Bildschirm auch Kleinbuchstaben dargestellt werden

auf die Pins 7 und 14 waagerecht ab. Die Pins 7 und 14 werden mit gleichen Pins des Z30 verlötet, wodurch Z30* zugleich seine Stromversorgung erhält und mechanisch fixiert wird.
Mit sieben kurzen Litzen werden schließlich noch die in Bild 5 eingezeichneten Verbindungen gelegt. Um unerwünschte Zusatzverbindungen zu vermeiden, wurden für alle Signale leicht zugängliche Lötpunkte gewählt.

Test der Modifikation

Ist der Aufbau soweit gediehen und noch einmal überprüft, kann ein erster Test gewagt werden. Bildschirm und Stromversorgung werden dazu an die Hauptplatine angeschlossen und der Rechner eingeschaltet. Wenn jetzt die typische „MEMORY-SIZE?"-Frage erscheint, kann man ziemlich sicher sein, daß man nicht allzuviel falsch gemacht hat. Mit dem Programm aus *Bild 7* kann man den Erfolg des Umbaus nun testen. Beim erfolgreichen Umbau sollte dasselbe erscheinen wie in Bild 7 dargestellt. Falls dieses Erfolgserlebnis ausbleiben sollte, bleibt nichts anderes übrig als den gesamten Umbau noch einmal gründlich auf Fehler zu untersuchen. Durch Wiederherstellen der durch Durchtrennen der Leiterbahn zerstörten Verbindung (vorher Litze zu Pin 4 von Z30* lösen!) kann der ursprüngliche Zustand zu Testzwecken wiederhergestellt werden.

Zusammenbau

Zu diesem Vorgang muß ich nicht viele Worte verlieren. Man lese die Anleitung und Bildfolge zu Schritt 1 rückwärts und gelangt hoffentlich zum gewünschten Ergebnis. Bleiben schließlich noch Abstandhalter, Schrauben oder sonstige Bauteile übrig, dann muß wohl irgend etwas schiefgegangen sein.

Die Software

Ganz ohne Software geht es leider nicht. Wer Diskettenbetriebssysteme mit Kleinbuchstabentreibern besitzt (z. B.: LCDVR im NEWDOS80 V.1) oder sogar solche, die durch Setzen einer System-Option auf Kleinbuchstaben umgeschaltet werden können, ist natürlich aus dem Schneider (mit System :0,BF=Y,BG=Y im NEWDOS80 V.2; in beiden NEWDOS80-Versionen empfiehlt es sich, zusätzlich nach Installation des Kleinschreibungsumbaus die Optionen AI und AS folgendermaßen zu setzen: SYSTEM :0,AI=Y,AS=N um dadurch

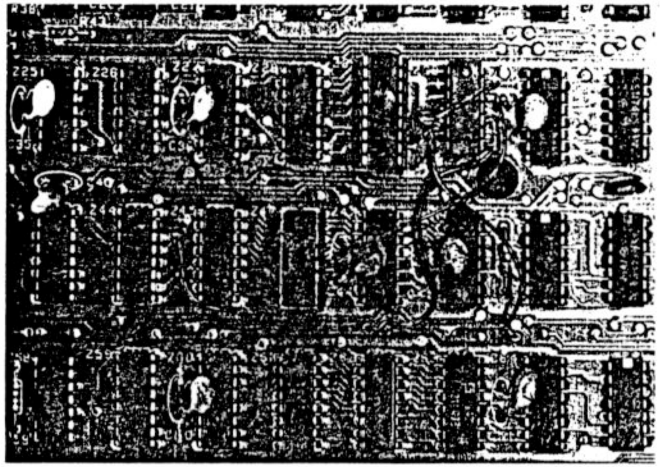


Bild 6. Dieser Ausschnitt zeigt die Videoschaltung auf der Hauptplatine des TRS-80, die für Kleinschreibung umgerüstet ist. Der zusätzliche RAM-Baustein Z46 wird Huckepack auf Z46 montiert. Dieser Aufbau empfiehlt sich, da beide Bausteine bis auf 2 Pins identisch verdrahtet sind
(Alle Fotos: Aegidius Röckrath)

die Umwandlung von Klein- in Großbuchstaben bei Ein- und Ausgabe zu verhindern).

Benutzer weniger komfortabler Betriebssysteme können mit den im folgenden beschriebenen Treibern den gleichen Effekt und bei Verwendung des AUTO-Kommandos den gleichen Komfort erreichen. Letzteres wird Kassettenbenutzern allerdings verwehrt bleiben. Sie müssen weiterhin von Hand den Treiber laden, um in den Genuß der Kleinschreibung zu gelangen.
Bild 8 zeigt den Treiber, für dessen bereitwillige Zur-Verfügung-Stellung ich mich an dieser Stelle bei Michael Büning bedanken möchte. Das Treiberprogramm enthält einen neuen Tastatur- und Displaytreiber. Der Displaytreiber verhindert nur die Umwandlung von Klein- in Großbuchstaben und deren Umwandlung in den „Tandy-Code". Der neue Tastaturtreiber ermöglicht die Eingabe von Kleinbuchstaben im normalen

Bild 7. Mit diesem einzeiligen Testprogramm kann der Erfolg des Umbaus überprüft werden. Wenn alles in Ordnung ist, erscheint auf den obersten vier Zeilen des Bildschirms das gleiche wie abgebildet


```
                ;***** LOWER-CASE-UMLAUTE-DRIVER FUER MODIFIZIERTEN TRS-80 ***** _
                ;
FF75                    ORG     0FF75H
FF75 21E2FF     LCUML1  LD      HL,NEWVID
FF78 221E40             LJ      (401EH),HL      ;NEUE VIDEO-DRIVER-ADRESSE
FF7B 2184FF             LD      HL,NEWKEY
FF7E 221640             LD      (4016H),HL      ;NEUE KEYBOARD-DRIVER-ADRESSE
FF81 C3E700             JP      00E7H           ;SPRUNG NACH BASIC-READY
                ;
                ;***** NEW KEYBOARD-DRIVER-SUBROUTINE *****
                ;
FF84 CDE303     NEWKEY  CALL    03E3H           ;NORMALER KEYBOARD-DRIVER
FF87 C5                 PUSH    BC
FF88 D5                 PUSH    DE
FF89 57                 LD      D,A
FF8A FE20               CP      20H
FF8C 2011               JR      NZ,NOSPAC
FF8E 3A8038             LD      A,(3880H)       ;SHIFT GEDRUECKT?
FF91 B7                 OR      A
FF92 2808               JR      Z,NOSPAC
FF94 3AD1FF             LD      A,(FLAG)
FF97 EE20               XOR     20H             ;SHIFTLOCK INVERTIEREN
FF99 32D1FF             LD      (FLAG),A        ;FLAG STATUS UMGEKEHRT
FF9C AF         IGNORE  XOR     A               ;AUSSPRUNG MIT A=0
FF9D 182F               JR      RUECK
FF9F 7A         NOSPAC  LD      A,D
FFA0 FE1F               CP      1FH             ;CLEAR IGNORIEREN
FFA2 28F8               JR      Z,IGNORE
FFA4 3A4038             LD      A,(3840H)
FFA7 FE02               CP      2               ;CLEAR ZUSAETZLICH GEDRUECKT?
FFA9 2010               JR      NZ,NORMAL
FFAB 7A                 LD      A,D
FFAC 21D2FF             LD      HL,ALT
FFAF 010800             LD      BC,NEU-ALT      ;BC=LAENGE DER TABELLE
FFB2 EDB1               CPIR
FFB4 2005               JR      NZ,NORMAL
FFB6 010700             LD      BC,NEU-ALT-1
FFB9 09                 ADD     HL,BC
FFBA 56                 LD      D,(HL)          ;UMCODIERTES ZEICHEN IN D LADEN
FFBB 7A         NORMAL  LD      A,D
FFBC E6DF               AND     0DFH
FFBE FE41               CP      41H
FFC0 380B               JR      C,SONDER        ;AUSSPRUNG FALLS NICHT ALPHA ZEICHEN
FFC2 FE5E               CP      5EH
FFC4 3007               JR      NC,SONDER       ;AUSSPRUNG FALLS NICHT ALPHA ZEICHEN
FFC6 7A                 LD      A,D
FFC7 21D1FF             LD      HL,FLAG
FFCA AE                 XOR     (HL)            ;GEMAESS SHIFTLOCK GROSS ODER KLEIN GENERI
EN
FFCB 1801               JR      RUECK
FFCD 7A         SONDER  LD      A,D
FFCE D1         RUECK   POP     DE
FFCF C1                 POP     BC
FFD0 C9                 RET
FFD1 00         FLAG    DEFB    00
FFD2 41         ALT     DEFB    41H             ;A
FFD3 4F                 DEFB    4FH             ;O
FFD4 55                 DEFB    55H             ;U
FFD5 61                 DEFB    61H             ;KLEIN A
FFD6 6F                 DEFB    6FH             ;KLEIN O
FFD7 75                 DEFB    75H             ;KLEIN U
FFD8 53                 DEFB    53H             ;S
FFD9 43                 DEFB    43H             ;C
FFDA 5B         NEU     DEFB    5BH             ;AE
FFDB 5C                 DEFB    5CH             ;OE
FFDC 5D                 DEFB    5DH             ;UE
FFDD 7B                 DEFB    7BH             ;KLEIN AE
FFDE 7C                 DEFB    7CH             ;KLEIN OE
FFDF 7D                 DEFB    7DH             ;KLEIN UE
FFE0 7E                 DEFB    7EH             ;SZ
FFE1 1F                 DEFB    1FH             ;CLEAR
                ;
                ;***** NEW VIDEO-DRIVER-SUBROUTINE *****
                ;
```

Bild 8. Leider kann die Kleinschreibung nur mit einem kleinen Treiberprogramm genutzt werden, da sie von der ROM-Software nicht unterstützt wird. Die Kassettenversion für 48-KByte-Systeme ist hier im Sourcelisting abgedruckt

```
                  00100 ;***** LOWER-CASE-UMLAUTE-DRIVER FUER MODIFIZIERTEN TRS-80 *****
                  00110 ;
FF72              00120        ORG    0FF72H
FF72 21E2FF       00130 LCUML1 LD     HL,NEWVID
FF75 221E40       00140        LD     (401EH),HL     ;NEUE VIDEO-DRIVER-ADRESSE
FF78 2184FF       00150        LD     HL,NEWKEY
FF7B 221640       00160        LD     (4016H),HL     ;NEUE KEYBOARD-DRIVER-ADRESSE
FF7E 2B           00170        DEC    HL
FF7F 224940       00180        LD     (4049H),HL     ;DOS-HIMEM UNTER LCUML1
FF82 AF           00190        XOR    A
FF83 C9           00200        RET                   ;ZURUECK INS DOS
                  00210 ;
```

Bild 9. Die Version des Treibers für Diskette erfordert einige kleine Änderungen gegenüber der Kassettenversion (Bild 8). Die abweichenden Quellprogramm-Statements sind hier abgedruckt. Der gesamte Rest ist identisch

Schreibmaschinenmodus und umgekehrt und bietet eine behelfsmäßige Möglichkeit zur Verarbeitung von Umlauten (der verwendete Algorithmus zur Codewandlung, der sich des CPIR-Befehls bedient, ist recht universell zu verwenden und sehr schnell).

Die Bedienung des Treibers ist denkbar einfach. Sobald der Treiber aktiv ist, können durch zusätzliches Drücken der Shift-Taste alle Kleinbuchstaben erzeugt werden und werden auch als solche auf dem Bildschirm angezeigt. Mit Shift-Space und Shift-0 kann in den Schreibmaschinen-Modus geschaltet werden bzw. dieser wieder verlassen werden. Schreibmaschinen-Modus bedeutet: Kleinbuchstaben ohne und Großbuchstaben mit Shift, ganz wie man es von einer normalen Schreibmaschine gewohnt ist. Mit Clear und A, O, U können die Umlaute Ä, Ö, Ü erzeugt werden, und mit Clear-S das ß. Man wundere sich nicht über die merkwürdigen Dinge, die dabei auf dem Bildschirm erscheinen (auch da ist Abhilfe möglich, s. u.). Mit Shift kann auch hier wieder zwischen groß und klein unterschieden werden. Da die Clear-Taste fortan diese Umschaltfunktion erfüllt, kann der Bildschirm nur noch durch gleichzeitiges Drücken von Clear und C gelöscht werden.

Bild 8 zeigt die Kassettenversion des Treibers in einem Quellprogrammlisting. Mit EDTASM oder einem Monitor erstellt man daraus ein Objekt-File, welches dann mit dem SYSTEM-Befehl geladen und mit / gestartet werden kann. Dabei ist keine Antwort auf die „MEMORY SIZE?"-Frage beim Einschalten erforderlich, denn das Programm schützt sich selbst vor dem Zugriff durch den Basic-Interpreter.

Bild 9 zeigt die ersten Quellprogrammzeilen der Diskettenversion des gleichen Treibers, die sich von denen der Kassettenversion unterscheiden. Unter DOS kann man ihn schnell und einfach mit dem DOS-Debugger in den Speicher schreiben und dann mit dem DUMP-Kommando als CMD-File auf Diskette abspeichern. Der Treiber wird fortan im DOS durch Angabe seines Filenamens geladen und gestartet, wobei er wie die Kassettenversion den Speicherbereich, den er belegt, vor unberechtigten Zugriffen schützt, so daß beim Aufruf des Basic die „MEMORY SIZE?"-Frage guten Gewissens mit Enter beantwortet werden kann.

Textverarbeitung

Nachdem der Kleinschreibungsumbau mühevoll vollzogen ist, stellt man leider fest, daß der TRS-80 in den USA entwickelt wurde. Für die dortigen Verhältnisse wäre er jetzt für Textverarbeitung vollkommen zufriedenstellend ausgerüstet und nichts stünde mehr im Wege. Aber leider erlaubt die deutsche Sprache sich den Luxus von Umlauten und des scharfen ß; Buchstaben, die weder als Tasten vorhanden sind noch auf dem Bildschirm dargestellt werden können. Durch Definition einer Umlaut-Umschalttaste (wie in dem oben vorgestellten Treiber) kann die Eingabe der Umlaute ermöglicht werden, wobei die etwas merkwürdige Darstellung auf dem Bildschirm etwas Kompromißbereitschaft erfordert.

Wer diese Kompromißbereitschaft nicht besitzt und auch noch die fehlenden Unterlängen bemängelt, der sollte einen weiteren Umbau in Kauf nehmen und einen neuen Zeichengenerator installieren, der all diese Mängel beseitigt und dabei noch bis zu drei verschiedene Zeichensätze bietet. Wenn man einmal soweit ist, ist auch das letzte Hemmnis, die fehlenden Umlauttasten, keine unüberwindliche Barriere mehr. Hier gibt es neuerdings Umbauten, die solche Tasten zusätzlich in den TRS-80 einbauen, wobei allerdings solche Umbauten immer nur so wertvoll wie die mitgelieferte Softwareunterstützung sind, sprich Modifikationen bekannter Textsysteme, wie SCRIPSIT oder PENCIL, die mit solchen Umbauten zusammenarbeiten. Die eleganteste Möglichkeit ist wohl, eine Typenrad-Schreibmaschine sowohl zur Eingabe als auch Ausgabe zu verwenden.

Hier angekommen wird man sich fragen, ob es nicht einfacher gewesen wäre, gleich einen Rechner mit entsprechender Ausrüstung zu erwerben, als nachträglich alle Fehler mühsam zu korrigieren. Für Erstanwender, die einen neuen Computer, der auf ihre Anwendung zugeschnitten ist, erwerben wollen, ist die Frage natürlich eindeutig mit ja zu beantworten. Aber wenn ein Rechner einmal vorhanden ist, ist die nachträgliche Umrüstung erheblich billiger und bietet unschätzbare Einblicke in das Innenleben des Rechners.

```
FFE2 DD6E03     00830 NEWVID  LD  L,(IX+3)     ALT     FFD2 00640    00420 00430 00460
FFE5 DD6604     00840         LD  H,(IX+4)     FLAG    FFD1 00630    00300 00320 00560
FFE8 DA9A04     00850         JP  C,049AH      IGNORE  FF9C 00330    00370
FFEB DD7E05     00860         LD  A,(IX+5)     LABEL   FFF2 00900    00880
FFEE B7         00870         OR  A            LCUML1  FF75 00130    00960
FFEF 2801       00880         JR  Z,LABEL      NEU     FFDA 00720    00430 00460
FFF1 77         00890         LD  (HL),A       NEWKEY  FF84 00210    00150
FFF2 79         00900 LABEL   LD  A,C          NEWVID  FFE2 00830    00130
FFF3 FE20       00910         CP  20H          NORMAL  FFBB 00490    00400 00450
FFF5 DA0605     00920         JP  C,0506H      NOSPAC  FF9F 00350    00260 00290
FFF8 FE80       00930         CP  80H          RUECK   FFCE 00600    00340 00580
FFFA D2A604     00940         JP  NC,04A6H     SONDER  FFCD 00590    00520 00540
FFFD C37D04     00950         JP  047DH
FF75            00960         END LCUML1
00000 TOTAL ERRORS
33210  TEXT AREA BYTES LEFT
```

Literatur

[1] Röckrath, Luidger: TRS-80-ROM-Listing. mc 1982, Heft 1, S. 12.

TRS 80 und Video enie — INLINE, PRINT und LPRINT

Manchmal ist es erforderlich, Satzzeichen in einen String zu bringen, wozu der Befehl INLINE benützt wird, welcher in Level II nicht enthalten ist. Mit drei Basic-Zeilen (120 Byte) kann Inline durch ein Unterprogramm ersetzt werden. Es können 255 Zeichen im String übernommen werden, wobei aber auch LEN und VARPTR funktioniert.

Beschreibung:

Zeile 1: S$ zurücksetzen.

Zeile 2: Cursor ausgeben.

Zeile 3: T$ zurücksetzen, auf Tastendruck warten.

Nach Tippen einer Taste eine Position zurück und Zeichen ausgeben. War Eingabe Linkspfeil, letztes Zeichen aus S$ entfernen, Cursor ausgeben, erneut auf Tastendruck warten, sonst zurück nach 2

Zeile 4: Ist das Zeichen ENTER, zurück in das Programm, sonst letztes Zeichen an S$ anfügen, Cursor ausgeben und nach Zeile 3.

```
1 S$=""
2 PRINT" - ";:GOSUB3:IF ASC(T$)=13, RETURN ELSE
S$=S$+T$:GOTO2
3 T$=" ":T$=Inkey$:1FT$=" ", 3 ELSE PRINT
CHR$(8);IFASC(T$)=8,S$=LEFT$(S$,(LEN(S$)-1)):
PRINT" - ";:GOTO3 ELSE RETURN
```

Tip für PRINT und LPRINT:

Wenn Sie Text oder Daten auf dem Bildschirm oder Drucker ausgeben müssen, schreiben Sie POKE P,L:PRINT.

P = 16540 und L = 0 Ausgabe auf dem Bildschirm.

oder L = 1 Ausgabe auf den Drucker.

So ersparen Sie sich, für PRINT und LPRINT getrennte Zeilen zu schreiben.

(G.Mulzheim)

„Gestehen Sie, daß Sie Ihre Schwiegermutter vom Balkon gestoßen haben, Angeklagter?" — „Ja, Herr Richter, das stimmt, aber ich wollte mich nur überzeugen, ob Drachen wirklich fliegen können"

Alle 3 Programme stammen vom Dieter. Mit dem 3. Programm brach er alle Rekorde!

PRIMZAHLEN 1

```
1 DEFSNGA-Z:DIMP(8500):P(1)=1.1:Z=1:CLS:INPUT"Anfang: ";A:INPU
T"Ende  : ";E:IFA=1ORA=2THENPRINT"2":A=3
2 IFCINT(A/2)=A/2THENA=A+1
3 FORI=ATOESTEP2:FORJ=1TOZ-1:IFSQR(I)<P(J)THENGOTO4IELSEIFCINT
(I/P(J))=I/P(J)THENNEXTIELSENEXTJ
4 P(Z)=I:Z=Z+1:PRINTI,Z:NEXTI:PRINT"FERTIG !"
```

PRIMZAHLEN 2

```
1 CLS:'---------  PRIMZAHLENBERECHNUNG  ------
2 INPUT" Bis zu welcher Grenze sollen die Primzahlen berechnet
 werden ";H%
3 FORA=3TOH%STEP2:W=INT(SQR(A))
4 FORB=3TOWSTEP2:X=INT(A/B)*B:IFX=ATHEN6
5 NEXTB:PRINTA;
6 NEXTA
```

PRIMZAHLEN 3

```
1 CLS:'---------  PRIMZAHLENBERECHNUNG  ------
2 DEFINTA-H
3 INPUT" BIS ZU WELCHER GRENZE SOLLEN DIE PRIMZAHLEN BERECHNET
 WERDEN ";H
4 DIMA%(H)
5 FORA=2TOSQR(H):IFA%(A)=1THEN7
6 PRINTA;:FORB=A*ATOHSTEPA:A%(B)=1:NEXT
7 NEXT
8 FORB=ATOH:IFA%(B)=0PRINTB;
9 NEXT
```

Berechnung nach dem 'Sieb von Eratosthenes'
Primzahlen bis 10000 in 3-4 Minuten !!!

Unser Sieger

Ein mysteriöser BASIC-Befehl: IsA

Wenn man Maschinenprogramme à la Leo Christopherson in Stringvariable packt und die BASIC-Programme mit diesen Variablen anschließend listet, erlebt man so manche Überraschung. Z. B. kann dabei der Befehl IsA auftauchen. Ich wollte untersuchen, was er bewirkt. Um das Ergebnis vorwegzunehmen: Nichts als einen ?SN-Error.

Da gibt Richard Straw in seinem Artikel "How the Level II Interpreter Sees It"[1] eine Tabelle der BASIC-Befehle mit ihren Tokens wieder. So gut der Artikel, so schlecht ist die Tabelle. Das Token FB (251)[2] steht für das Zeichen ', den REM-Ersatz. In der Tabelle steht nichts davon. Diese Lücke setzt sich (von hier ab allerdings zu Recht) fort bis FE (254). Diese Codes entsprechen wirklich keinen BASIC-Befehlen. Aber bei FF (255) schreibt Straw brav ISA hin, weil er das bei seinem Tandy ohne Kleinschrift so gesehen hat. Er ist der Sache wohl nicht nachgegangen, sonst hätte er feststellen müssen, how the level II interpreter nämlich wirklich sees it.

Da die CPU kein Englisch und auch kein BASIC kann, schlägt ihr Dolmetscher, der Interpreter (engl. Dolmetscher) bei jedem BASIC-Token in einer Art Wörterbuch nach, einer Tabelle, die bei 1650 (5712) beginnt. Die einzelnen Befehlswörter, die (fast) in Klarschrift hier stehen, werden durch das gesetzte Bit 7 des jeweils ersten Buchstabens eines jeden Wortes voneinander unterschieden. So heißt z. B. REM hier D2 45 4D (210 69 77) statt 52 45 4D (82 69 77). Der Interpreter findet nach dem LIST-Befehl im Programmtext das Token 93 (147) und setzt von diesem Token zunächst das Bit 7 zurück. Das tut er, indem er 80 (128) subtrahiert. So entsteht aus 93 (147) 13 (19). Jetzt sucht er 19mal nach einem gesetzten Bit 7 in der Tabelle und findet die obigen Codes. Er setzt vom ersten Zeichen das Bit 7 zurück und schreibt dieses und die nachfolgenden Zeichen auf den Bildschirm. Wenn er ein Zeichen findet, bei dem wieder das Bit 7 gesetzt ist, hört er damit auf, denn dort beginnt das nächste Befehlswort. Auf dem Screen stehen jetzt die Buchstaben R-E-M für REM.

Bei dem "Token" (es ist keins) FF (255) sucht er eben 127mal, denn 255-128=127. Er findet bis zum nächsten Bit 7 ab 182A (6186) folgende Bytes: C9 01 73 41 (201 1 115 65). Er setzt sodann das Bit 7 von C9 (201) pflichtgemäß zurück, so daß 49 (73) daraus wird. Kreuzbrav, denn er hat nichts gemerkt, schreibt er nun nach dem oben skizzierten Strickmuster die ASCII-Äquivalente dieser Codes auf den Bildschirm. 01 bedeutet ihm nichts, der Rest heißt dann eben IsA. So kann auch ein Genie irren.

Der casus cnaxus ist, daß hier die Tabelle längst zuende ist. Das unwiderruflich letzte Token, FB (251) für ' (REM-Ersatz) ist bereits überschritten. Wir befinden uns schon in der anschließenden Tabelle, wo die Adressen der Bearbeitungsroutinen dieser BASIC-Befehle verzeichnet sind.

Dieselbe Erklärung gilt auch für das Phänomen, daß bei derartigen Listings häufig auf 32 Zeichen/Zeile umgeschaltet wird, der Cursor plötzlich in die Home-Position geht usw.. Beim Genie 3 oder beim Genie 1/2 mit H-DOS und EG 64 MBA piept es sogar gelegentlich. Alle Codes ≥FC (252) werden irrtümlich für Tokens gehalten und die in der falschen Tabelle gefundenen "Befehlswörter" (die keine sind) buchstabengetreu angezeigt. Bei Codes ≤19 (31) passiert im Prinzip dasselbe: Wenn z. B. 1C (28) auftritt, geht der Cursor an den Bildschirmanfang.

Dick Straw ist gewiß nicht dumm, aber vielleicht ein bißchen zu fix mit der Feder bzw. der Tastatur bei der Hand.

Und was bringt das alles dem Programmierer? Rein gar nichts. Aber ist es nicht von allgemeinem Vorteil, die Eigenheiten des Interpreters zu kennen, wenn man sich seiner via BASIC bedient?

Arnulf Sopp, Tel. 0451-791926

[1] 80-US 5/79, in "The First Book of 80-US", Hofacker-Verlag
[2] alle Zahlen sedezimal, dahinter in Klammern dezimal

Ein Beitrag vom Josef Konrad

```
10 REM DATUM-KONVERTIERUNG VERSION 1
20 REM
30 DIM MO$(12)
40 CLS
50 DEFFN MO$(M)=MID$("   JANFEBMAEAPRMAIJUNJULAUGSEPOKTNOVDEZ",M*3,3
)
60 M=VAL(LEFT$(TIME$,2)):Y$=FN MO$(M)
70 DA$=MID$(TIME$,4,2)
80 YR$=MID$(TIME$,7,2):YR$="19"+YR$
90 DT$=DA$+". "+Y$+" "+YR$
100 PRINT@40,DT$
110 REM
120 REM
130 REM
140 REM
150 REM DATUM-KONVERTIERUNG VERSION 2
160 REM
170 CLS
180 X=VAL(LEFT$(TIME$,2))
190 DA$=MID$(TIME$,4,2)
200 YR$=MID$(TIME$,7,2):YR$="19"+YR$
210 FOR N%=1 TO X:READ MO$:NEXT N%
220 DT$=DA$+". "+MO$+" "+YR$
230 DATA "JANUAR","FEBRUAR","MAERZ","APRIL","MAI","JUNI","JULI","AU
GUST","SEPTEMBER","OKTOBER","NOVEMBER","DEZEMBER"
240 PRINT@45,DT$
```

Am Anfang war das Chaos

Ein Arzt, ein Architekt und ein Datenverarbeiter sitzen gesellig beisammen. Unversehens taucht die Frage auf, wer von ihnen wohl den ältesten Beruf ausübe.

Der Medizinmann verkündet siegessicher, diesen Ruhm könne er gewiß für seinen Stand beanspruchen. Denn schließlich heiße es, daß Gott die Eva aus einer Rippe Adams geschaffen habe, und das sei ja wohl eine chirurgische Tat.

Entgegnet lächelnd der Architekt: sein Beruf sei doch um einiges älter. Schon im Ersten Buch Moses stehe schwarz auf weiß, wie Gott die Welt aus dem Chaos schuf – und dabei handele es sich offenkundig um eine architektonische Leistung.

Daraufhin erhebt sich der Datenverarbeiter und fragt trocken: »Und wer schuf das Chaos?«

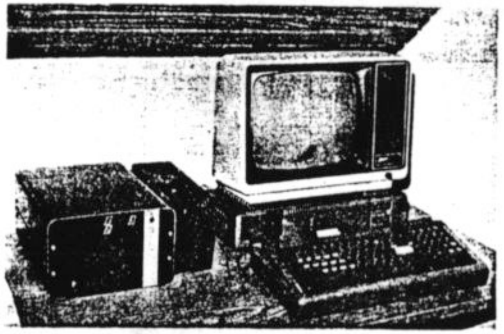
von Josef Konrad

```
               00100 ;*****************************************************
               00110 ;***   SCHUETZEN VON ASSEMBLER-PROGRAMMEN MODELL I   ***
               00120 ;***   DIESE BEFEHLSFOLGEN RESERVIEREN DEN SPEICHER-  ***
               00130 ;***   BEREICH AB 'START'.                            ***
               00140 ;***   SIE WERDEN VOR DAS ZU SCHUETZENDE PROGRAMM     ***
               00150 ;***   GESCHRIEBEN.                                   ***
               00160 ;*****************************************************
               00170 ;
               00180 ;         FUER LEVEL II BASIC
               00190 ;
0000 210000    00200          LD      HL,START        ;STARTADR. DES PROGRAMMS
0003 C3E700    00210          JP      00E7H           ;BEREICH SCHUETZEN
               00220 ;
               00230 ;
               00240 ;         FUER DOS
               00250 ;
0006 210000    00260          LD      HL,START        ;STARTADR. DES PROGRAMMS
0009 2B        00270          DEC     HL
000A 224940    00280          LD      (4049H),HL      ;DOS-HIMEM VOR PROGRAMM
000D AF        00290          XOR     A               ; A := 0
000E C9        00300          RET                     ;ZURUECK INS DOS
               00310 ;
               00320 ;
0000           00330 START    EQU     $-$             ;START DES ZU SCHUETZENDENPROGRAMMS
               00340 ;
0000           00350          END
00000 TOTAL ERRORS
32552  TEXT AREA BYTES LEFT

START  0000 00330   00200 00260
```

Modernisierung statt Neukauf

Wer schon seit einigen Jahren glücklicher Besitzer eines Computers ist, muß feststellen, daß die Technik in der Zwischenzeit nicht stehengeblieben ist. Er steht dann vor der Frage, ob er sein Gerät gegen ein neues austauschen soll, was meistens auch einen Wechsel von Systemsoftware nach sich zieht. Der Verfasser des folgenden Artikels hat sich für eine Modernisierung seines TRS-80 entschieden.

▲ Bild 1. Zeichengenerator mit Unterlängen und deutschen Umlauten



◄ Bild 2. Modernisierter TRS-80 Modell-1 mit Zenith-Bildschirm

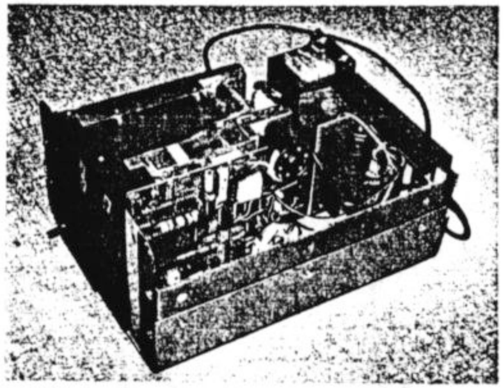


Bild 3. Marktgängiges Diskettenlaufwerkgehäuse mit erweitertem Netzteil

Der TRS-80 Modell 1 des Verfassers stammt aus dem Jahre 1978. Er hat die Seriennummer 16286 und dürfte somit einer der ältesten in der BRD laufenden TRS-80 Modell 1 sein. Seit 1978 hat es bei den Tischcomputern viele technische Fortschritte gegeben. Ende 1982 kam der Verfasser auf die Idee, sich auf dem Markt einen modernen Nachfolger für den TRS-80 auszusuchen. Das Ergebnis der dabei gemachten Marktstudie war unerfreulich für die Computershops und erfreulich für den eigenen Geldbeutel:

1. Die vielgepriesenen Fortschritte der Mikroelektronik seit 1978 sind so gering, daß der TRS-80 durchaus noch heutigen Ansprüchen genügt. Man kann es als Außenstehender am Markterfolg des Video-Genies erkennen.
2. Zum beim Verfasser benutzten Disketten-Betriebssystem NEWDOS-80-V2 gibt es keine gleichwertige Alternative. Ein Wechsel zu einem CP/M-fähigen System oder zum DOS 3.3 des Apple II erscheint beim Vergleich der Möglichkeiten der Betriebssysteme als Rückschritt.
3. Obwohl der TRS-80 insgesamt und sein Basic insbesondere recht langsam ist (das heißt in Zeitschriften-Benchmarktests schlecht abschneidet), ist das Basic des TRS-80 insbesondere unter NEWDOS-80-V2 leistungsfähiger als sämtliche dem Verfasser bekannten Interpreter-Basic-Dialekte für Tischcomputer.

Unter dem Eindruck dieser Erkenntnisse entschloß sich der Verfasser, anstelle einer Neuanschaffung eine Modernisierung des vorhandenen TRS-80 durchzuführen. Ziel der Modernisierung sollte sein:
- die Leistungsfähigkeit der TRS-80-Anlage erhöhen,
- die technisch sinnvolle Nutzungsdauer der TRS-80-Anlage so weit zu verlängern, bis die technischen Fortschritte auf dem Gebiet der Mikroelektronik den im TRS-80 Modell 1 verwendeten Mikroprozessor Z80 eindeutig als veraltet erscheinen lassen.

Ein wesentlicher Vorteil der Modernisierung gegenüber einem Neukauf eines anderen Systems darf nicht übersehen werden: Bei jedem Systemwechsel ist ein beachtlicher Aufwand zur Übernahme der Software und zur Neubeschaffung der Systemsoftware zu leisten. Bei einer Modernisierung entfällt dieser Aufwand vollkommen.

Alles aus dem Versandhandel

Praktisch alles wurde über den Versandhandel beschafft. Aufgrund von Anzeigen in den bekannten Tischcomputer-Zeitschriften wurden die Kontakte zu den Händlern aufgenommen. Händler, die Anfragen nicht beantworteten (so etwas gibt es wirklich!), erhielten selbstverständlich keine Bestellung. Bezahlt wurde mit Verrechnungsscheck (der Verfasser hat eine Abneigung gegen Nachnahme) oder — bei Bestellungen in den USA — per Visa-Kreditkarte. Die schnellste Lieferung kam innerhalb von vier Tagen (Computerstudio Braunschweig). Keine innerdeutsche Bestellung hat länger als zwei Wochen gedauert, keine Ware mußte reklamiert werden.

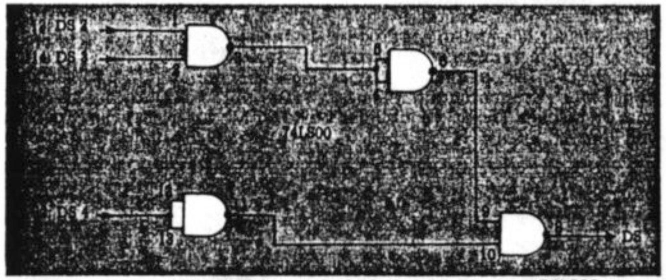


Bild 4. Selektion von Laufwerk 3 für den TRS-80 Modell-1 mit einem 74LS00

Bei der Lieferantenauswahl war der Preis das wesentliche Auswahlkriterium. Preisunterschiede von 20 Prozent sind bei gleicher Hardware normal, und es lohnt sich deshalb für sachkundige Käufer, die Lieferanten zu wechseln.

Die ersten Schritte zum »neuen« Gerät

Die beiden ersten Modernisierungsstufen erfolgten bereits in den letzten Jahren:

1. Stufe: Geschwindigkeitsumschalter für den CPU-Takt.

2. Stufe: Neuer Zeichengenerator mit Unterlängen und deutschen Umlauten (Bild 1).

Die beiden für die Erweiterungen erforderlichen zusätzlichen Platinen wurden selbst eingebaut.

3. Stufe: Doppelte Schreibdichte auf Disketten.

Beim TRS-80 besteht die Möglichkeit, die Diskettenaufzeichnungskapazität durch Einbau eines sogenannten »Doublers« in das Expansion Interface um 80 Prozent zu erhöhen. Die Doubler benötigen ein entsprechend angepaßtes Betriebssystem: Am besten ist einmal mehr NEWDOS-80-V2 geeignet. Nach Einbau des Doublers ist die Nutzung der Disketten mit einfacher Aufzeichnungsdichte weiterhin möglich, und der TRS-80 bleibt voll kompatibel zu seinem Ursprungszustand. Mit einiger Mühe beim Aussuchen konnte der Verfasser etwa 30 Prozent der vorhandenen, für einfache Schreibdichte vorgesehenen Disketten auch für doppelte Schreibdichte verwenden. Wichtig ist, daß man sich vor dem Kauf des Doublers beim Lieferanten erkundigt, ob die vorhandenen Laufwerke für doppelte Schreibdichte geeignet sind.

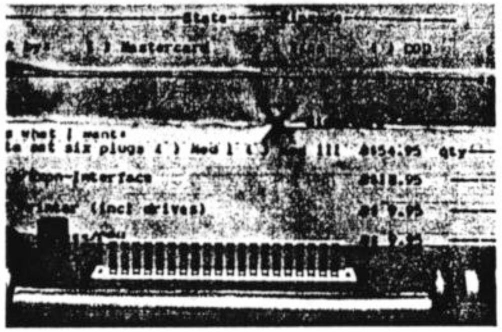


Bild 5. Die aufgelöteten, vergoldeten Kartenstecker stehen etwas vor

Der übliche Preis für einen Doubler beginnt je nach Lieferanten bei etwa 230 Mark. Wenn man ausländischen Computerzeitschriften Glauben schenken darf, bietet Tandy/Radio Shack in den USA Doubler für den TRS-80 Modell-1 an.

4. Stufe: Ein neues Gehäuse für zwei Diskettenlaufwerke

Ursache für diese Investition war das bei einem vorhandenen Diskettenlaufwerk mitgelieferte Gehäuse. Es war von billigstmöglicher Machart, und seine Seitenteile bestanden aus Kunststoff (einer der Freunde des Verfassers pflegt in diesem Zusammenhang nicht ganz wertfrei von Pappmachegehäusen zu reden). Wenn das Laufwerk in einem solchen Gehäuse nahe am Bildschirmmonitor steht, treten beim Betrieb mit doppelter Schreibdichte Schreib/Lesefehler auf. Dieses Gehäuse wurde gegen ein marktgängiges aus Aluminiumblech für zwei Diskettenlaufwerke mit eingebautem Netzteil ersetzt. Es wurde durch den Verfasser so umgebaut, daß es nun zwei Netzteile für insgesamt vier Diskettenlaufwerke enthält (Bilder 2 und 3).

5. Stufe: Das neue Diskettenlaufwerk

Anfang 1983 bewegten sich die Preise für Diskettenlaufwerke rapide nach unten. Es gelang dem Verfasser, ein BASF-Laufwerk preiswert zu erwerben, welches als logisches Laufwerk mit der Adresse 3 in das oben genannte Gehäuse eingebaut wurde (Bild 3).

Probleme bei der Steckerbeschaffung

Immer dann, wenn man Laufwerke ohne Gehäuse erwirbt, hat man Probleme mit der Kabel- und Steckerbeschaffung. Für den Verfasser war es ein glücklicher Umstand, daß Radio Shack an seinem Wohnort Ende 1982 ein Computer Center eröffnet hat. Ein Telefonanruf genügte, um zu erfahren, daß die benötigten Stecker auf Lager waren. Die Ersatzteilversorgung bei Radio Shack ist — der Verfasser hat es bisher dreimal ausprobiert — ausgezeichnet.

Da das Laufwerk die logische Adresse 3 hat, muß die Selektion des Laufwerkes angepaßt werden. Dazu bietet sich eine Gatterschaltung an, die man in das Laufwerk zusätzlich einbauen muß. Die Selektionstabelle beim TRS-80-Modell-1 sieht wie folgt aus (siehe nächste Seite).

Signal am Kabel	DS1	DS2	DS3	DS4
Laufwerk 0	0	1	1	1
Laufwerk 1	1	0	1	0
Laufwerk 2	1	1	0	0
Laufwerk 3	1	1	1	0

Somit genügt zur Selektion des Laufwerkes 3 die Verwendung der Signale DS2, DS3 und DS4 (Bild 4).

6. Stufe: Der neue Bildschirm

Der beim TRS-80 Modell-1 anno 1978 mitgelieferte Bildschirm war nicht gerade sehr schön — die Helligkeit schwankte, und das Bild stand auch nicht still, sondern wackelte mit einer Frequenz von 10 Hertz. Dieser Bildschirm wurde gegen einen Zenith-Bildschirm ersetzt (Bild 2). Der Bildschirm findet auf dem Expansion-Interface gut Platz, und er ist zum TRS-80 Modell-II-1 voll kompatibel.

Wie auch bei anderen Bildschirmen, muß man möbelschonende Füße aus Weich-PVC zusätzlich kaufen und selbst unterkleben.

Dieser Bildschirm — er wird von verschiedenen Lieferanten angeboten — war das Gerät mit den größten Preisabweichungen: je nach Händler kostete er zwischen 265 Mark und 395 Mark.

7. Stufe: Goldplug als Stecker — teuer aber gut

Die einzelnen Aggregate des TRS-80 Modell-1 sind durch steckbare Flachkabel miteinander verbunden. Die Kartenstecker sind verzinnt und geben nur für die Dauer der Garantiezeit des TRS-80 Modell-1 ausreichend Kontakt. Später beeinträchtigt der Übergangswiderstand der Steckverbindungen die Zuverlässigkeit. Man erkennt es als Benutzer daran, daß der TRS-80 gelegentlich Speicherinhalte verändert und plötzlich mitten im Betrieb einen Kaltstart durchführt.

Die Abhilfe kommt aus den USA: auflötbare, vergoldete Kartenstecker. Sie haben den eingetragenen Markennamen »Goldplug«. Sie funktionieren ausgezeichnet, haben aber zwei Nachteile:

1. Die Kunststoffabdeckteile passen nicht mehr auf die Stecker, denn die aufgelöteten Goldplugs stehen etwas vor (Bild 5).
2. Ein Steckersatz kostet ein kleines Vermögen, und zwar US$ 54,95.

Leider gibt es keine Importfirma für diese von vielen TRS-80-Modell-1-Besitzern dringend benötigten Stecker. Der Hersteller und Lieferant ist: E.A.P. Co., P.O.Box 14, Keller, Texas 76248.

Was hat der Umbau gekostet, was hat er gebracht?

Die Investition für die Umbaustufen außer den vorgezogenen Stufen 1 und 2 lagen um 1080 Mark. Dieser Betrag ist im Vergleich zu einer Neuanschaffung und gegengerechnetem Verkauf der Altanlage sehr gering. Ebenso betrug der für die Realisierung benötigte Zeitaufwand nur einen Bruchteil der für eine Systemumstellung benötigten Zeit.

(Rolf-Fr. Matthaei)

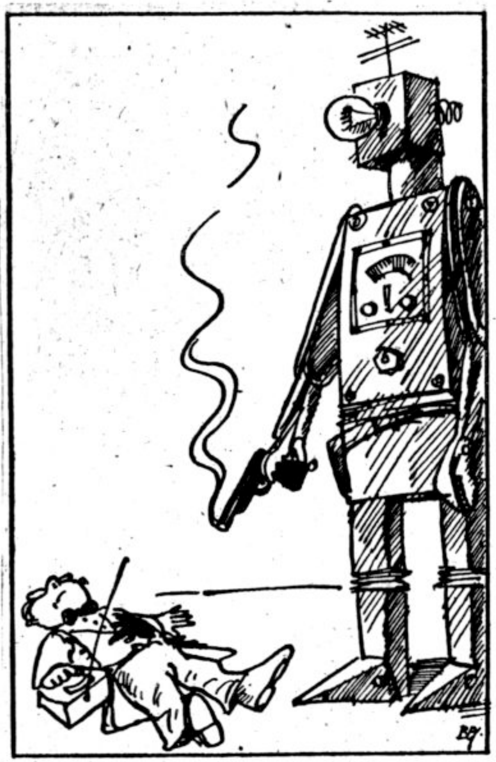
VORWORT ZUM 2. TEIL BASIC F&B

Viele Zuschriften ermunterten mich, weitere Seiten zu veröffentlichen. Es sind also dieses Mal insgesamt 24 Seiten. Einige Male wird auf den Anhang verwiesen, dieser wird im nächsten Info abgedruckt.

Warum mache ich mir diese Arbeit?
Nun, zum einen sitze ich sehr oft pro Tag ca. 2 Stunden im Zug. Lernen o.ä. kann ich da nicht - erstaunlicherweise aber kann ich dort dieses Buch übersetzen. Das verbinde ich mit dem Englischen, so daß ich da ein klein wenig drin bleibe.
Ein weiterer wichtiger Punkt: Was ich übersetzt habe, habe ich auch (meistens) verstanden. Ich hoffe, daß mich das weiterbringt. Ich glaube auch, daß einigen Mitgliedern durch diese Übersetzung geholfen wird. Nicht alle werden ausreichenden Englisch-Unterricht gehabt haben.

So - einige kleine Fehler bzw. Unstimmigkeiten entdeckte ein Mitglied im Assembler-Teil. Da das Buch sich mit BASIC befaßt, gehe ich an dieser Stelle nicht näher ein (vielleicht zu einem späteren Zeitpunkt). Aber bitte - meldet mir alle Fehler. Insbesondere die Beispiele und Listings sind sehr fehlerträchtig.

Aber nun genug. Ich hoffe, daß der 2. Teil allen etwas bringt!



Falsch programmiert ...
(Zeichnung: Georg Beyer)

DAS LADEN VON DISKETTE UND DAS AUSFÜHREN VON USR-ROUTINEN

Wir wollen annehmen, daß wir die Bildschirm-Füll-Routine als Disk-Datei 'BFUELL' abgespeichert haben. Unser erster Schritt ist das Laden der Routine in den Speicher (RAM). Von 'DOS READY' aus können wir die Routine BFUELL durch LOAD BFUELL laden.

Nun rufen wir BASIC auf. Wir dürfen dabei nicht vergessen, daß wir den Speicher entsprechend unserer Routine begrenzen müssen. Ansonsten kann unser BASIC-Programm die Routine zerstören. Wir schauen uns das Assembler-Listing an, und sehen als Beginn der Routine die Adresse BFF0, das der Dezimalzahl 49136 entspricht. Unsere Antwort auf MEMORY SIZE darf nicht größer als 49136 sein. Im BASIC können wir den Speicher ganz einfach begrenzen, indem wir eingeben: PRINT 65536 + &HBFF0

Wenn wir im BASIC sind, muß unser Programm die Startadresse der USR-Routine festlegen. Der DEFUSR-Befehl im Disk-BASIC erlaubt uns die Definition von maximal 10 Startadressen für 10 USR-Routinen (0 bis 9). Um unsere Routine als USR-Routine 0 zu definieren könnte unser Programm wie folgt aussehen:

```
10 DEFUSR0=&HBFF0
```
oder,
```
10 DEFUSR=&HBFF0
```
oder,
```
10 DEFUSR0=49136
```
oder,
```
10 DEFUSR=49136
```

Falls wir mehr als eine USR-Routine haben, können wir die zweite als DEFUSR1, die dritte als DEFUSR2, etc. definieren. Man sollte sich merken, daß man USR-Routinen in einem Programm so oft umdefinieren kann, wie man will. Außerdem bleiben die Adressen von USR-Routinen so lange erhalten, bis sie umdefiniert werden oder BASIC neu geladen wird. Man kann andere BASIC-Programme laden und starten so oft man will - die definierten USR-Routinen bleiben erhalten.

Zur Ausführung unserer Bildschirm-Füll-Routine können wir das folgende kleine Programm eingeben:

```
10 DEFUSR0=&HBFF0
20 PRINT@0,"X"
30 J=USR0(0)
```

Nach RUN wird der Bildschirm sofort mit 'X' gefüllt. Man muß nur die Zeile 20 ändern, wenn man den Bildschirm mit anderen Zeichen füllen will. Die Zeile 30 ruft die USR-Routine auf. In diesem Fall ist 'J%' eine Schein-Variable, genauso wie die '0' in den Klammern. In raffinierteren Programmen werden wir die '0' durch eine Integer-Zahl oder einen Ausdruck ersetzen, und teilen so der USR-Routine einen Wert mit, den diese für Berechnungen bzw. die Ausführung benötigt. Wir werden 'J%' oder andere Integer-Variablen benutzen, um von USR-Routinen einen Wert für das BASIC-Programm zu erhalten.

DAS POKEN VON USR-ROUTINEN IN DEN SPEICHER

Jede USR-Routine dieses Kurses wird im POKE-Format vorgestellt. Anders ausgedrückt wird eine Liste von Zahlen angegeben, die man benötigt, wenn man eine USR-Routine in den Speicher POKE'n will. Auf diese Weise benötigt man keinen Editor-Assembler; außerdem muß man die Z-80 Maschinensprache nicht verstehen. Die besprochene Bildschirm-Füll-Routine kann durch das POKE'n der folgenden 12 Zahlen in beliebige 12 benachbarte Adressen des RAM 'geladen' werden:

33, 0, 60, 17, 1, 60, 1, 255, 3, 237, 176, 201

Probieren Sie diese Schritte um zu sehen, wie es geht:

1. Von DOS READY aus laden Sie BASIC mit einer MEMORY SIZE von 49136.
2. Geben Sie folgendes Programm ein:

```
10 DEFUSR0=&HBFF0
15 DATA 33,0,60,17,1,60,1,255,3,237,176,201
16 FORX=0TO11: READ P: POKE &HBFF0+X,P: NEXT
20 PRINT@0,"X"
30 J=USR0(0)
```

3. Starten Sie das Programm. Der Bildschirm wird sofort mit 1024 'X' gefüllt. Ersetzen Sie nun die Zeile 20 durch:

```
20 PRINT@0,CHR$(191)
```

Starten Sie das Programm wieder, und Sie haben sofort einen weißen Bildschirm.

Die DATA-Ausdrücke in Zeile 15 entsprechen den 12 Bytes aus der USR-Routine. In Zeile 16 werden diese 12 Zahlen in 12 Adressen des geschützten Speichers gepoked, beginnend bei BFF0 (dezimal 49136), so daß die Routine ausgeführt werden kann.

Da die Bildschirm-Füll-Routine veränderlich ist, kann die Adresse &HBFF0 in den Zeilen 10 und 16 durch jede andere Adresse ersetzt werden. Wer einen TRS-80 mit 48K hat, wird z.B. FFF0 anstatt BFF0 verwenden. In Vereinbarung mit MEMORY SIZE können natürlich auch niedrigere Adressen gewählt werden.

Fragen Sie sich, wie man die POKE-Werte erhält? Unser assembliertes Listing zeigt uns die hexadezimalen Werte der USR-Routine. Das Kommando 'LD HL,15360' in Zeile 40 ergab die Maschinensprache-Anweisung 21003C. Diese Anweisung wird in Dezimalzahlen umgewandelt:

21 entspricht 33 dezimal.
00 entspricht 0 dezimal.
3C entspricht 60 dezimal.

Wir fahren fort mit den Zeilen 50 bis 80 unseres Assembler-Listings und erhalten so der Reihe nach unsere 12 Werte. Noch einfacher erhalten wir diese Werte, wenn wir das assemblierte Programm von Diskette oder Kassette in den Speicher laden. Wir können dann von BASIC aus mittels PEEK die Werte einfach von der ersten bis zur letzten Adresse der Routine ausgeben. Wir verwenden dazu folgende Anweisung:

```
FOR X=&HBFF0 TO &HBFFB: PRINT PEEK(X);: NEXT
```

DAS ABSPEICHERN VON USR-ROUTINEN AUF DISKETTE

Jede USR-Routine in diesem Kurs wird im POKE-Format angegeben. Das heißt, daß eine Liste von Zahlen angegeben wird, die mittels POKE in den Speicher gebracht wird, beginnend mit einer Startadresse, die im geschützten Speicher liegt. Wenn die zu einer USR-Routine gehörenden Zahlen einmal in den Speicher gepoked worden sind, kann die Routine unter jedem beliebigen Namen auf der Diskette abgespeichert werden. Nehmen wir an, wir wollen die Bildschirm-Füll-Routine auf Diskette abspeichern:

1. Zuerst gehen wir ins BASIC unter Beachtung von MEMORY SIZE. Sie muß für die USR-Routine ausreichend sein, so daß die USR-Routine auf alle Fälle im geschützten Speicherbereich liegt. In unserem Beispiel legten wir als MEMORY SIZE 49136 fest, so daß wir unsere 12-Byte USR-Routine bei BFF0 beginnen lassen können.

2. Dann schreiben oder laden wir ein Programm, welches uns die benötigten Zahlen in die gewünschten Adressen poked. Hier sind die Programmzeilen für unsere 'BFUELL'-Routine:

```
15  DATA 33,0,60,17,1,60,1,255,3,237,176,201
16  FORX=0TO11: READ P: POKE &HBFF0+X,P: NEXT
```

   Beachten Sie, daß es sich dabei um die Zeilen 15 und 16 unseres Test-Programms handelt.

3. Als nächstes starten wir das Programm. Es liest die DATA-Ausdrücke und poked die Zahlen in den Speicher.

4. Nun gehen wir in das DOS zurück (CMD"S").

5. Im DOS benutzen wir DUMP. Um die 12 Bytes, die sich noch ab Adresse BFF0 im Speicher befinden, in eine Disk-Datei mit dem Namen 'BFUELL/CIM' zu dumpen, benutzen wir folgende Anweisung:

```
DUMP SFILL (START=X'BFF0',END=X'BFFB')
```

   Beachten Sie, daß das DUMP-Kommando an den Datei-Namen automatisch den Zusatz '/CIM' anfügt, wenn nicht eine andere Erweiterung festgelegt wird. Das DOS-Handbuch erklärt diese und andere Details des DUMP-Befehls.

6. Immer wenn Sie in einem Programm die BFUELL-Routine benötigen, geben Sie einfach vor Aufruf des BASIC das Kommando BFUELL ein. Es wird dann die Routine in den Speicher geladen, und zwar genau an die Adresse, an der diese Routine gedumpt worden ist. Wenn Sie ins BASIC gehen, müssen Sie wieder einen Speicherbereich für diese Routine schützen.

Wer will, kann 'BFUELL/CIM' in jeden anderen möglichen Datei-Namen umbennen. Man verwendet hierzu den RENAME-Befehl. Wenn Sie die Datei zum Beispiel in 'BILDFUEL' umbenennen, so hat diese nicht mehr die Erweiterung '/CIM' und die Anweisung zum Laden vom DOS aus lautet LOAD BILDFUEL.

Wenn Sie ein Model III haben, oder auf Model I NEWDOS benutzen, so können Sie die Routine direkt von BASIC aus laden. Im BASIC unter NEWDOS schreiben wir

```
10  CMD "BFUELL"
```
oder
```
10  CMD "LOAD BFUELL"
```

Je nachdem, ob BFUELL den Zusatz '/CIM' hat oder nicht.

Bei einem Model III unter TRSDOS 1.3 lautet die DUMP-Anweisung:

```
DUMP BFUELL (START=BFF0,END=BFFB)
```

Nun können Sie die unter dem Namen BFUELL/CMD abgespeicherte Routine von TRSDOS READY aus einfach durch Eingabe von BFUELL starten. Im BASIC können Sie folgende Anweisung verwenden:

```
10  CMD "L","BFUELL/CMD"
```

SUPER-STRINGS

DAS LADEN VON USR-ROUTINEN IN STRINGS

Wir können jede veränderliche USR-Routine in einen String laden, wenn sie nicht länger als 255 Bytes ist. Diese Technik bietet einige große Vorteile. Zunächst können wir die geforderte Reservierung von Speicherplatz (die 'MEMORY-SIZE'-Frage) vermeiden, wenn wir die USR-Routine in einen String laden. Weiterhin können wir die Routine sehr leicht von einer Speicherstelle zur anderen bewegen, indem wir unseren String mittels VARPTR in einen anderen String poken. Und schließlich können wir die Routine in eine ganz gewöhnliche Disk-Datei abspeichern, welche eine umfassende Bibliothek von Routinen beinhalten kann, und von der man schnell und bequem von BASIC aus die Routinen laden kann.

Unsere Bildschirm-Füll-Routine können wir mit folgender Anweisung in den String S$ laden:

```
S$=CHR$(33)+CHR$(0)+CHR$(60)+CHR$(17)+CHR$(1)+CHR$(60)+
CHR$(1)+CHR$(255)+CHR$(3)+CHR$(237)+CHR$(176)+CHR$(201)
```

Um nun die Routine auszuführen, können wir die Adresse unserer USR-Routine so definieren, daß sie auf die Daten im String hinweist:

```
DEFUSR0=PEEK(VARPTR(S$)+1)+256*PEEK(VARPTR(S$)+2)
```

Zur Sicherheit sollten wir jedoch vor jedem Aufruf der Routine die Adresse neu definieren, da sich während eines Programmablaufes die Adresse des Strings S$ mehrmals ändern kann. Es kann sich z.B. die Adresse ändern, wenn neue Variablen zum ersten Mal benutzt werden, oder wenn eine DEF-Anweisung ausgeführt wird.

Hier ist nun eine einfachere Möglichkeit, um längere USR-Routinen in einen String zu laden, besonders wenn die Routine bereits im geschützten Speicher geladen und getestet worden ist:

1. Laden Sie (in einen geschützten Speicherbereich) die Routine von Diskette (Datei, welche mit einem Editor-Assembler erstellt wurde) oder poken Sie die Routine. Wie dies bei der Bildschirm-Füll-Routinen gemacht wird, ist bereits erklärt worden.

2. Definieren Sie mittels DEFUSR den Beginn der USR0-Routine. Bei unserem Beispiel startet die Bildschirm-Füll-Routine bei BFF0:
   ```
   DEFUSR0 = &HBFF0
   ```

3. Definieren Sie einen String wie folgt: S$ = ""

4. Poken Sie VARPTR von S$, so daß die Stringlänge der USR-Routine-Länge entspricht. In unserem Beispiel:
   ```
   POKE VARPTR(S$),12
   ```

5. Poken Sie den Zeiger der USR-Routine in VARPTR des Strings hinein. Im Anhang 2 steht eine Liste von USR-Routinen-Zeiger-Adressen für die meisten bekannten DOS-Systeme. Für das Model I unter NEWDOS verwenden wir:
   ```
   POKE (VARPTR(S$)+1),PEEK(&H5B14)
   POKE (VARPTR(S$)+2),PEEK(&HBB15)
   ```

Nun enthält der String S$ die USR-Routine und wir können S$ in einer Random-Diskdatei abspeichern. So können wir in zukünftigen Programmen die Routinen ganz einfach laden und ausführen, ohne daß wir uns mit Speicherplatz-Reservierung oder DATA-Ausdrücken quälen müssen.

Die Random-Diskdatei, welche wir erstellen, kann so viele USR-Routinen enthalten, wie auf einer Diskette gespeichert werden können. Um die in S$ abgespeicherte Routine in den Satz 1 einer mit 'USR' benannten Random-Diskdatei abzuspeichern, benutzen wir folgendes Kommando:

```
OPEN R,1,"USR"
FIELD 1, LEN(S$) AS A$
LSET A$ = S$
PUT 1,1
CLOSE
```

Immer wenn wir in einem zukünftigen Programm die Bildschirm-Füll-Routine benötigen, können wir irgendwo im Bereich des Programmanfangs die folgenden Anweisungen zum Laden der Routine in S$ programmieren:

```
OPEN R,1,"USR"
FIELD 1,12 AS A$
GET 1,1
S$=A$
CLOSE 1
```

Wenn nötig, können wir die Routine wie folgt aufrufen:

```
POKE &H5B14, PEEK (VARPTR(S$)+1)
POKE &H5B15, PEEK (VARPTR(S$)+2)
J=USR0(0)
```

Die zwei Poke's erfüllen die Funktion des DEFUSR-Befehls; die Adressen werden durch den Befehl VARPTR aus S$ eingelesen. Die Adressen &H5B14 und &H5B15 im obigen Beispiel müssen ersetzt werden durch die Adressen aus dem Anhang 2, wenn Sie ein anderes DOS verwenden.

Als Alternative können Sie die USR-Routine während der Ausführung im Disk-Buffer belassen. Jeder Disk-Buffer ist ein 256 Byte langer, geschützter Speicher, der durch die Frage 'HOW MANY FILES?' reserviert wird. Die Adressen der Disk-Buffer sind im Anhang 2 wiedergegeben.

Um nun den Buffer 1 zur Ausführung unserer Bildschirmfüll-Routine verwenden zu können, verwenden wir folgende Anweisung zum Laden der Routine:

```
OPEN R,1,"USR"        'Eröffnet die Datei USR
GET1,1                'Einlesen des Satzes 1
DEFUSR0 = &H6575      'Adresse des Disk-Buffers defin.
```

Die Routine können wir dann immer wie folgt aufrufen:

```
J = USR(0)
```

Sie werden erkennen, daß diese Technik der 'Superstrings' eine der schnellsten, flexibelsten und wirksamsten Methoden für die Anwendung von USR-Routinen darstellt.

SUPER-DATENFELDER

DAS LADEN UND AUSFÜHREN VON 'SUPER-DATENFELDERN'

Genauso wie man eine USR-Routine in einen String laden kann, und dann diesen 'String' ausführt, kann man auch eine USR-Routine in ein Integer-Feld laden, und dann dieses 'Super-Datenfeld' ausführen. Bei dieser Technik braucht man keinen Speicherplatz reservieren. Ein Integer-Feld von 15 Elementen reserviert und schützt automatisch 30 Bytes des Speichers. Ein ebenso bedeutender Vorteil dieser Technik ist die bequeme und wirtschaftliche Methode Integer-Werte an USR-Routinen weiterzugeben.

Um zu sehen, wie man mit 'Super-Datenfeldern' arbeitet, geben Sie dieses kurze Programme ein und starten es. Das Programm entspricht wieder unserer Bildschirm-Füll-Routine.

```
 5 DEFINT A-Z: J=0
10 US(0)=8448:US(2)=4352):US(4)=256:US(6)=-20243:US(7)=201
20 US(1)=15360:US(3)=15361:US(5)=1023
30 PRINT@0,"X"
40 DEFUSR0=VARPTR(US(0))
50 J=USR0(0)
60 GOTO 60
```

Wir laden 7 Integer-Werte in ein Integer-Feld. In Zeile 40 definieren wir die Adresse unserer USR-Routine. In Zeile 50 rufen wir die USR-Routine auf, welche im 'Super-Datenfeld' abgespeichert ist.

Nun betrachten wir uns die Zeile 20. Wir übergeben an die USR-Routine 3 Elemente über die Feldelemente Nr. 1, 3 und 5. Element 1 gibt die Adresse an, welche kopiert werden soll, in diesem Fall 15360 als Adresse der oberen linken Ecke unseres Bildschirms. Das Element 3, das hier um 1 größer ist wie Element 1, gibt an, daß nur 1 Byte zu kopieren ist. Das 5. Element bestimmt, daß das ganze 1023 mal ausgeführt werden soll.

Wir wollen jetzt eine Veränderung versuchen, indem wir andere Parameter verwenden. Das Wort "TEST" soll 63 mal kopiert werden. Es sind lediglich die Zeilen 20 und 30 wie folgt zu ändern:

```
20 US(1)=15360:US(3)=15364:US(5)=63*LEN("TEST")
30 PRINT@0,"TEST"
```


Nun starten Sie das Programm und "TEST" wird 63 mal kopiert. Wir änderten also nur die Argumente für unsere USR-Routine. Wie Sie sehen, ist diese Methode der POKE-Methode überlegen.

Seien Sie vorsichtig, wenn Sie mit dieser oder anderen Routinen zu experimentieren anfangen. Nur ein falsches Argument, und der Computer 'taucht unter'. Deshalb befolgen Sie bitte folgende Vorsichtsmaßnahmen, bevor Sie mit Experimenten beginnen:

- Speichern Sie das Programm ab
- Nehmen Sie alle Disketten aus den Laufwerken

Außerdem müssen zuerst die Vorschriften für die Benutzung von 'Super-Datenfeldern' besprochen werden:

1. Das Datenfeld muß ein Integer-Feld sein. In unserem Beispiel benutzen wir einfach die Anweisung 'DEFINT A-Z', um sicherzustellen, das US% ein Integer-Feld ist.

2. Das Programm darf zwischen dem DEFUSR-Befehl und dem Aufruf der USR-Routine keine neuen Variablen benutzen. Um dieser Regel zu entsprechen, wird die Variable 'J%' in der Zeile 5 unseres Beispiel-Programms vorinitialisiert.

Diese Regel ist notwendig, weil BASIC Integer-Felder immer im Speicher nach oben verschiebt, wenn in einem Programm eine neue Variable benutzt wird. Falls wir die Variable 'J%' in der Zeile 50 zum ersten Mal benutzen, wandert die Adresse unseres Feldes nach oben und unsere DEFUSR-Anweisung in Zeile 40 wird ungültig. Es ist gut, wenn man unmittelbar vor Aufruf einer 'Super-Datenfeld'-Routine die DEFUSR-Anweisung ausführt. Auf diese Weise führt in komplexen Programmen das zufällige Verschieben der USR-Routine durch Verwendung einer neuen Variablen zu keinem Programm-Fehler.

Jede USR-Routine dieses Kurses wird im Format eines 'Super-Datenfeldes' angegeben. Es wird eine Liste der Integer-Zahlen angegeben, die man zum Laden eines Integer-Feldes benötigt, falls man die Methode des 'Super-Datenfeldes' benutzt. Für längere Routinen kann man DATA-Ausdrücke verwenden, um die Integer-Zahlen in das Feld zu bekommen. Die Technik des 'Super-Datenfeldes' ist bestens geeignet für kurze USR-Routinen bis ungefähr 50 Bytes Länge. Sie werden vielleicht bemerkt haben, daß die Technik des Super-Datenfeldes etwas langsam wird, wenn in einem Programm mehrere große Felder während der Programm-Ausführung benutzt werden. Aber für kurze USR-Routinen ist die Technik des Super-Datenfeldes in der Tat 'Super'!

DAS SCHREIBEN VON 'SUPER-DATENFELD'-USR-ROUTINEN

Wie Sie gesehen haben, liefern die Super-Datenfelder einen einfachen Weg zum Laden von Argumenten aus BASIC in eine Maschinensprache-USR-Routine. Wer die Z-80 Assemblersprache kennt, kann hier nachlesen, wie man USR-Routinen als Super-Datenfeld schreibt:

1. Schreiben Sie eine Z-80 Routine und assemblieren Sie diese mit einem Editor-Assembler. Es muß eine veränderliche Routine sein.

2. Betrachten Sie das assemblierte Listing um festzustellen, wo Sie Argumente benötigen. Wenn nötig, fügen Sie entweder 'NOP'-Befehle ein, oder reorganisieren die Routine so, daß die Argumente bei jedem geraden Byte innerhalb der Routine starten. Falls die Länge der Routine nicht gleichmäßig durch 2 teilbar ist, so fügen Sie ein NOP als letzte Anweisung an, damit die Routine durch 2 teilbar wird. Nun assemblieren Sie die Routine nochmals und kontrollieren anschließend, ob die Routine nun richtig ausgerichtet ist.

Nun betrachten wir das Assembler-Listing, welches zur Erstellung des 'Super-Datenfeldes' für unsere Bildschirm-Füll-Routine benötigt wurde. Von jetzt an wollen wir diese Routine als 'DATA-Treiber'-Routine bezeichnen, da diese Routine bei vielen Anwendungen nützlich ist, wo wir DATA-Blöcke von einer Speicherstelle zu einer anderen verschieben wollen.

```
BFF0          00100     ORG     0BFF0H       ;BEGINN-VERÄND.
BFF0 00       00110     NOP                  ;NO-OP F. AUSR.
BFF1 210000   00120     LD      HL,0         ;LADE "VON"-ADR.
BFF4 00       00130     NOP                  ;NO-OP F. AUSR.
BFF5 110000   00140     LD      DE,0         ;LADE "NACH"-ADR
BFF8 00       00150     NOP                  ;NO-OP F. AUSR.
BFF9 010000   00160     LD      BC,0         ;LADE # V. BYTES
BFFC EDB0     00170     LDIR                 ;BC BYTES HL-DE
BFFE C9       00180     RET                  ;RUECKSPRUNG
BFFF 00       00190     NOP                  ;F. GERADE LAENG
0000          00200     END                  ;
00000 TOTAL ERRORS
```

Im Assembler-Listing der DATA-Treiber-Routine sehen wir, daß in den Zeilen 120, 140 und 160 jeweils Integer-Nullen von 2 Byte Länge in die Register HL, DE und BC geladen werden. Im BASIC werden diese Nullen durch den Inhalt der Elemente 1, 3 und 5 ersetzt. So wie wir Parameter in die erforderlichen Feldelemente innerhalb eines BASIC-Programms laden, laden wir auch unsere Argumente in die Routine.

In den Zeilen 110, 130 und 150 fügten wir NOP's ein, damit sich die Parameter auf gerade Bytes ausrichten. Die Z-80 NOP-Anweisung ist einfach eine 8-Bit Null und bedeutet 'keine Anweisung' (no operation). Der Computer beachtet diese NOP's einfach nicht und fährt mit der nächsten Anweisung fort.

In der Zeile 170 steht die gewaltige Z-80 LDIR-Anweisung. Sie bewegt ein Byte mit der Adresse aus dem BC-Register in die Adresse aus dem DE-Register. Dann werden das BC- und DE-Register erhöht, das BC-Register wird erniedrigt. Das ganze wiederholt sich, bis BC=0 geworden ist.

In der Zeile 190 steht nochmals ein NOP. Dadurch wird die Länge der Routine gerade. Es ist wichtig, daß Super-Datenfeld-Routinen von gerader Länge sind.

Wenn Sie die Routinen assembliert haben, laden Sie diese in den Speicher und gehen ins BASIC, wobei Sie die geschützte Speichergröße so wählen, daß Ihre Routine nicht überschrieben wird.

Um nun die Integer-Werte zu erhalten, die wir für unser Super-Datenfeld benötigen, schreiben wir folgendes Programm:

```
10  S% = &HBFF0            'Startadresse
20  E% = &HBFFF            'Endadresse
30  FOR X = S% TO E% STEP 2
40  PRINT CVI(CHR$(PEEK(X))+CHR$(PEEK(X+1)));
50  NEXT
```

Natürlich stehen in den Zeilen 10 und 20 die Start- und Endadressen Ihres Programmes. Gewöhnlich wird man in Zeile 40 anstatt PRINT ein LPRINT schreiben, um eine Ausgabe auf Drucker zu erreichen.

DAS ABSPEICHERN VON 'SUPER-DATENFELDERN' IN RANDOM-DISK-DATEIEN

Die Technik des Super-Datenfeldes hat einige feine Vorteile beim Laden von USR-Routinen in den Speicher. So benötigt eine Routine im Super-Datenfeld-Format nur die Hälfte der Argumente, welche beim POKE-Format benötigt werden.

Wenn man einmal die benötigten Werte einer Routine in ein Datenfeld gebracht hat, möchte man gewöhnlich die im Datenfeld befindliche USR-Routine in einer Random-Disk-Datei abspeichern. Dadurch verschenkt man in zukünftigen Programmen, in denen man diese Routine einsetzen möchte, keinen Speicher, welcher von DATA-Ausdrücken belegt werden würde. Wie speichert man nun 'Super-Datenfeld'-Routinen mit einer maximalen Länge von 127 Elementen in einer Random-Disk-Datei ab:

1. Eröffnen Sie Ihre Random-Disk-Datei.

2. Erstellen Sie FIELD mit 255 Bytes für A$.

3. Definieren Sie eine Schein-String-Variable S$ durch S$="".

4. Poken Sie in VARPTR von S$ die Länge Ihrer im Datenfeld abgespeicherten Routine. Die Länge entspricht der doppelten Element-Anzahl, da jedes Element 2 Bytes beinhaltet.

5. Poken Sie in VARPTR von S$+1 die Adresse von LSB (Least Signifigant Byte = erstes bedeutendes Byte) Ihres Datenfeldes. Wenn Ihr Datenfeld mit US%(0) beginnt, so schreiben Sie:

   ```
   POKE VARPTR(S$)+1, ASC(MKI$(VARPTR(US%(0))))
   ```

6. Poken Sie in VARPTR von S$+2 die Adresse von MSB (Most Signifigant Byte = höchstes bedeutendes Byte) Ihres Datenfeldes. Wir schreiben:

```
POKE VARPTR(S$)+2, ASC(RIGHT$(MKI$(VARPTR(US%(Ø))),1))
```

Nun enthält S$ die USR-Routine. Führen Sie LSET A$=S$ aus und bringen Sie A$ mittels PUT in den gewünschten physikalischen Satz.

Immer dann, wenn Sie diese Routine in einem Programm benutzen wollen, eröffnen Sie die Disk-Datei und lesen mit GET den gewünschten Satz in den Speicher ein. Sie können dann Ihre Routine über den Disk-Buffer ausführen, oder auch die Routine in einen anderen geschützten Speicherbereich bringen. Sie können die Routine aber auch wieder in ein Datenfeld bringen.

Ein Beispiel. Nehmen wir an, Sie haben 58 Werte über DATA-Ausdrücke in ein Super-Datenfeld US% gebracht. Ihre USR-Routine beginnt mit US%(Ø). Um nun diese Routine in den physikalischen Satz 2 einer mit "ROUTINEN" bezeichneten Datei abzuspeichern, verwenden Sie folgende Anweisungen:

```
OPEN "R",1,"ROUTINEN": FIELD1, 255 AS A$
S$="": POKE VARPTR(S$),116
POKE VARPTR(S$)+1, ASC(MKI$(VARPTR(US%(Ø))))
POKE VARPTR(S$)+2, ASC(RIGHT$(MKI$(VARPTR(US%(Ø))),1))
LSET A$=S$: PUT 1,2: CLOSE
```

Wenn Sie nun diese Routine zu einem späteren Zeitpunkt ohne Verwendung von DATA-Ausdrücken in ein Super-Datenfeld zurückbringen wollen, können Sie folgende Anweisungen benutzen:

```
DIM U$(58)
OPEN "R",1,"ROUTINEN": FIELD1, 116 AS A$
GET 1,2
S$="": POKE VARPTR(S$),116
POKE VARPTR(S$)+1, ASC(MKI$(VARPTR(US%(Ø))))
POKE VARPTR(S$)+2, ASC(RIGHT$(MKI$(VARPTR(US%(Ø))),1))
LSET S$=A$
```

Wenn Sie die Routine nicht in Feldelementen benötigen, können Sie natürlich auch die Techniken zum Laden und Ausführen von Super-Strings verwenden.

VERWANDLUNG VON USR-ROUTINEN DURCH KONTROLLFELDER

Dies ist eine weitere gewaltige Technik, die Sie in Ihrem DOS-Handbuch vergeblich suchen werden. Wir erstellen einfach ein Integer-Feld, welches die Argumente beinhaltet, die wir an eine USR-Routine weitergeben wollen. Dieses 'Kontrollfeld' kann ebenso Integer-Werte enthalten, welche von der USR-Routine erstellt wurden und an das BASIC übergeben worden sind.

Zum Beispiel benötigt die USR-Routine 'SORT1', welche String-Felder in aufsteigender Reihenfolge sortiert, 2 Werte. Das aufrufende BASIC-Programm muß das zu sortierende String-Feld und die Anzahl der zu sortierenden Elemente angeben. Diese 2 Argumente sind in einem Integer-Feld enthalten. Das Element Ø beinhaltet VARPTR des String-Feldes und Element 1 beinhaltet die höchste zu sortierende Element-Nummer des String-Feldes.

Um die ersten 600 Elemente des Feldes S$ zu sortieren, benutzen wir folgende Anweisungen, um die USR-Routine mit dem Feld C% als Kontrollfeld zu benutzen:

```
100  C%(Ø)=VARPTR(S$(Ø))
101  C%(1)=599
102  J=USRØ(VARPTR(C%(Ø)))
```

Im Programm müssen wir schon vorher mit dem DEFUSR-Kommando die Adresse unserer SORT1 USR-Routine geladen haben. Ebenso muß bereits die Schein-Integer-Variable J% definiert worden sein, damit Sie beim USR-Aufruf richtig arbeitet. Die Methode des Kontrollfeldes zur Übergabe von Werten kann für jede beliebige USR-Routine angewandt werden, egal ob diese im geschützten Speicher, im Super-String oder im Super-Datenfeld steht.

Kontrollfelder sind besonders nützlich, wenn viele Argumente zwischen der USR-Routine und BASIC ausgetauscht werden müssen. Bei allen USR-Routinen, die Kontrollfelder benutzen, wird angegeben, welche Elemente bzw. Angaben benötigt werden.

Es gibt einiges bei der Benutzung von Kontrollfeldern zu beachten:

1. Ein Kontrollfeld muß ein Integer-Feld sein. Deshalb sollten Sie entweder immer das Prozent-Zeichen '%' benutzen, oder die benutzten Variablen mit DEFINT definieren.

2. Erinnern Sie sich daran, daß sich die Feld-Adressen ändern, wenn Sie während der Ausführung eines BASIC-Programms eine neue Variable definieren. Falls eines der Elemente des Kontroll-Feldes das VARPTR eines anderen Feldes beinhaltet, müssen Sie sicherstellen, daß während des Ladens des Kontroll-Feldes und dem Aufruf der USR-Routine keine neuen Variablen benutzt werden.

3. Sie müssen nicht mit dem Element Ø des Kontrollfeldes starten. Sie können bei anderen Anwendungen andere Elemente benutzen. Zum Beispiel könnten wir die SORT1-Routine wie folgt aufrufen:

```
100  C%(14)=VARPTR(S$(Ø))
101  C%(15)=599
102  J=USRØ(VARPTR(C%(14)))
```

Wenn Sie in eigenen USR-Routinen Kontrollfelder benutzen wollen, so werfen Sie einen Blick auf irgendeine USR-Routine dieses Kurses, welche diese Technik anwendet.

Sie werden feststellen, daß die ersten drei Z-80 Anweisungen der Routine immer wie folgt lauten:

```
CALL  0A7FH
PUSH  HL
POP   IX
```

'CALL 0A7FH' lädt den Klammerwert der USR()-Funktion des BASIC-Programmes in das HL-Register. Die Hilfsroutine 0A7F vom ROM führt dies für uns aus. Weil der vom BASIC übergebene Wert der VARPTR-Wert eines Kontrollfeldes ist, bezieht sich HL auf das erste Element des Feldes.

Die PUSH- und POP-Anweisungen kopieren den Inhalt des HL-Registers in das IX-Register. Dann könnten wir z.B. den Inhalt des zweiten Elementes unseres Kontrollfeldes in das DE-Register laden:

```
LD  E,(IX+4)
LD  D,(IX+5)
```

Durch das entgegengesetzte Verfahren können wir Daten an das Kontrollfeld übergeben. Wenn wir den Inhalt von BC in das 3. Feldelement übergeben wollen, so können wir schreiben:

```
LD  (IX+6),C
LD  (IX+7),B
```

Falls wir nur einen Wert an das BASIC-Programm übergeben wollen, so lautet die letzte Anweisung der USR-Routine:

```
JP  0A9AH
```

Dies bewirkt einen Sprung zu einer ROM-Routine, welche den Inhalt von HL dem BASIC übergibt. Falls wir diesen Sprung benutzen um ins BASIC zurückzukehren, und wenn unser Aufruf

```
J=USR0(VARPTR(C%(0)))
```

gelautet hat, so erhält die Variable 'J' den letzten Wert von HL, welcher durch die USR0-Routine berechnet wurde. Wenn wir nur eine 'RET'-Anweisung benutzen, um ins BASIC zurückzukehren, so bleibt der Inhalt der Variablen J% von dem USR-Aufruf unberührt.

VERÄNDERLICHER VIELWERT-VERWALTER FÜR USR-AUFRUFE

Für das Programmieren in Assembler wird nun eine Routine angegeben, welche man bei USR-Routinen als eine alternative Methode zum Verwalten von verschiedenen Werten benutzen kann. Der VW-Verwalter (Viel-Wert-Verwalter) ermöglicht es im BASIC, alle Werte, die der USR-Routine in einfacher Genauigkeit übergeben werden sollen, festzulegen.

Zum Beispiel benötigt unsere 'DATA-Treiber'-Routine 3 Werte:

1. von Adresse
2. nach Adresse
3. Anzahl der zu übergebenden Werte

Mit dem VW-Verwalter können wir z.B. 50 Bytes von Adresse 15360 nach 15384 übertragen. Unser USR-Aufruf lautet:

```
J=USR(15360) OR USR(15384) OR USR(50)
```

Der Verwalter unterstützt einen Zähler für die bereits übertragenen Werte. Wenn alle Werte (in diesem Fall 3) anerkannt worden sind, so wird die Kontrolle für die Ausführung der Werte an die USR-Routine übergeben:

```
                 00000 ;VW-VERWALTER
                 00001 ;
FF00             00100          ORG   0FF00H       ;BEGINN
FF00 CD7F0A      00110          CALL  0A7FH        ;WERT IN HL STELLEN
FF03 DD2A145B    00120          LD    IX,(05B14H)  ;IX = DEFUSR-ADR.
FF07 DD7535      00130          LD    (IX+53),L    ;
FF0A DD7436      00140          LD    (IX+54),H    ;WERT SPEICHERN
FF0D DD3409      00150          INC   (IX+9)       ;
FF10 DD3409      00160          INC   (IX+9)       ;1. ZÄHLER + 2
FF13 DD340C      00170          INC   (IX+12)      ;
FF16 DD340C      00180          INC   (IX+12)      ;2. ZÄHLER + 2
FF19 DD7E09      00190          LD    A,(IX+9)     ;
FF1C 0635        00200          LD    B,53         ;
FF1E 90          00210          SUB   B            ;A=ÜBERTRAGENE WERTE * 2
FF1F DD4634      00220          LD    B,(IX+52)    ;B=RESTLICHE   WERTE * 2
FF22 90          00230          SUB   B            ;
FF23 2806        00240          JR    Z,PASS1      ;WENN 0, ALLE WERTE ÜBERT.
FF25 210000      00250          LD    HL,0000H     ;CLEAR FÜR RETURN
FF28 C39A0A      00260          JP    0A9AH        ;RETURN FÜR NÄCHSTEN WERT
FF2B DD360935    00270 PASS1    LD    (IX+9),53    ;
FF2F DD360C36    00280          LD    (IX+12),54   ;ZÄHLER ZURÜCKSETZEN
FF33 1806        00290          JR    START        ;
FF35 0000        00300          DEFW  0            ;WERT 1 SPEICHERN
FF37 0000        00310          DEFW  0            ;WERT 2 SPEICHERN
FF39 0000        00320          DEFW  0            ;WERT 3 SPEICHERN
FF3B 00          00330 START    NOP                ;ROUTINE STARTET HIER
402D             00340          END   402DH        ;
00000 TOTAL ERRORS
```

Wie schreibt man eine Z-80 Hilfsroutine mit dem VW-Verwalter:

1. In Bezug auf die verwendete Nummer der USR-Routine und auf das verwendete DOS suchen Sie sich im Anhang 2 die Zeiger-Adresse der USR-Routine. Ändern Sie Zeile 120 entsprechend. Unser Beispiel zeigt 5B14 als Adresse des Zeigers für USR0 unter NEWDOS 2.1.

2. Zwischen den Zeilen 290 und 330 müssen genau so viele DEFW-Anweisungen stehen, wie Sie Werte von BASIC an die USR-Routine übergeben wollen. Wichtig ist, daß nichts anderes zwischen 'JR START' (Zeile 240) und 'START' (Zeile 330) steht, da der Verwalter die Differenz zwischen diesen beiden Anweisungen benutzt, um festzustellen, wieviele Werte übergeben werden müsse, bevor die Hauptroutine aufgerufen werden kann.

3. Die Logik Ihrer Routine muß vor 'JR START' und nach 'START' stehen. Zugang zu den übertragenen Werten erreichen Sie über das IX-Register:

   (IX+53) und (IX+54) beinhalten den ersten Wert
   (IX+55) und (IX+56) beinhalten den zweiten Wert
   (IX+57) und (IX+58) beinhalten den dritten Wert, etc.

Um nun z.B. den zweiten Wert in das DE-Register zu laden, müssen Sie schreiben:

```
LD  E,(IX+55)
LD  D,(IX+56)
```

Der VW-Verwalter ist wahrscheinlich die bequemste Art, um USR-Routinen von BASIC aus aufzurufen. Behalten Sie für dessen Benutzung bitte folgendes im Gedächtnis:

1. Bei jedem Aufruf können maximal 25 Werte übertragen werden.

2. Sie müssen entscheiden, welche USR-Routine benutzt werden soll, weil der Zeiger im Verwalter assembliert ist. (Für mehr Flexibilität können Sie auchJ in das 6. und 7. Byte poken).

3. Der Verwalter fügt etwa 50 Bytes an Ihre Routine an.

4. Die Logik ändert sich während der Ausführungszeit selbst. Alle Variablen müssen ordnungsgemäß übertragen werden, oder der Verwalter wird sich nicht mehr in seinen Ursprungs-Zustand zurückinitialisieren.

5. Sie können Speicher sparen, falls Ihre USR-Routine nicht veränderlich sein muß. Der Hauptvorteil des VW-Verwalters ist seine Veränderlichkeit, aber auch, daß das Arbeiten mit Werten (Speicherstellen) in der Routine integriert ist.

SUPER - SPEICHERTECHNIKEN

'Jedes Programm will sich ausweiten, um den gesamten vorhandenen Speicherplatz zu belegen'

Falls Sie schon längere Zeit auf Ihrem Computer programmiert haben, werden Sie diese Aussage bestätigen können. Egal, wieviel Speicher- oder Diskettenkapazität man hat - man findet immer irgendeinen Weg, diesen zu belegen. Dieses Kapitel erklärt nun Techniken, die helfen sollen, das Beste aus den vorhandenen Speicher herauszuholen.

Bestimmt kennen Sie Vorführungen, wo Experten das Publikum mit schneller Speicherung von Namen oder Buchseiten unterhalten. Diese schönen Vorführungen sind im Prinzip nur auf einfache Techniken aufgebaut, die jeder erlernen kann. Dieses Kapitel will Ihnen einige einfache Techniken nahe bringen, mit deren Hilfe Sie von Fall zu Fall Ihrem Computer erstaunliche Speicherfähigkeiten geben können. Sie werden sehen, daß Ihre Programme ganz neue Leistungen schaffen, wenn Sie erstmal wissen, wie man den Speicher kontrolliert, wie man schnell Daten verschiebt und wie man Programm-Teile von Diskette lädt und wieder abspeichert.

WIEVIEL SPEICHER HABEN SIE WIRKLICH ?

Falls Sie einen 48K TRS-80-Computer gekauft haben, haben Sie in Wirklichkeit einen 64K Speicher. Wenn Sie einen 32K TRS-80 haben, sind es 48K. Da ein Teil des Speichers vom ROM belegt wird, kann man diesen als Programmierer nicht verändern, und man wird sich überlegen, wo die Grenze des verwendbaren Speichers liegt:

Radio Shack Katalog	Höchste Adr. hexadezimal	höchste Adr. dezimal	höchstes Byte Integer-Format
'16K'	7FFF	32767	32767
'32K'	BFFF	49151	-16385
'48K'	FFFF	65535	-1

Club 80 - Info Nr. 6

Hallo liebe Mitglieder

Ihr werdet zu Recht erstaunt sein über die neue Aufmachung unseres Clubinfos.

Warum habe ich das gemacht ? Ganz einfach - ich habe endlich die Vorschläge einiger Clubmitglieder aufgegriffen. Ich glaube, daß durch die Querrichtung und die neue Heftung das Info besser zu lesen ist. Ferner gibt es ab sofort eine durchgehende Seitennummerierung und ein Inhaltsverzeichnis. Ich hoffe, daß Ihr nun mit dem Info noch besser zurecht kommt. Ihr könnt mir ja mal Euere Meinung dazu mitteilen.

Ein ganz dickes Lob verbunden mit einem großen Dankeschön möchte ich loswerden an die Mitglieder, die Beiträge zum Clubinfo liefern oder sonst im Club konstruktiv mitarbeiten. Nur durch diese Mitglieder ist das Info in dieser Aufmachung überhaupt erst möglich.

Ich möchte deshalb auf dieser 1. Seite auch weiterhin um rege Mitarbeit seitens der Mitglieder bitten. Klar ist, daß viele Mitglieder keine Beiträge liefern können - aber die, die was beisteuern können, sollten dies bitte auch tun.

Was dieses Info bringt steht im Inhaltsverzeichnis. Ich möchte Euch nun noch über die aktuelle Mitgliederzahl informieren. Wir haben 41 Mitglieder (ist doch toll !) + 3 Mitglieder mit einem großen Fragezeichen (von Ihnen wurde der Jahresbeitrag 85 noch nicht bezahlt - Austrittserklärung liegt auch nicht vor!). Was wir mit den 3 letztgenannten 'Mitgliedern' machen (alle mindestens zweimal angemahnt) entscheiden wir auf dem Clubtreffen).

Genug für das Vorwort - viele Grüße

Euer Günther Wagner

INHALTSVERZEICHNIS:

Vorwort.....Seite 1
Inhaltsverzeichnis.....Seite 2
Rücktritt von Günther Wagner.....Seite 3
Aktuelle Mitgliederliste.....Seite 4
Clubtreffen.....Seite 5-6
Verkausangebote, Systemänderung Hartmut Obermann.....Seite 6
Gesucht, Treffen in München.....Seite 7
Fragen.....Seite 8
Die neuen Club-Mitglieder.....Seite 9-10
Persönliche Vorstellungen.....Seite 11-12
Cobol-Kurs (2. Teil).....Seite 13-17
Funktionsweise des Druckerpuffers.....Seite 18
Datenfernübertragung (Akkustikkoppler selbst bauen ?)....Seite 19-22
Angebot Tandy-Akkustikkoppler.....Seite 22
Monitor im Genie.....Seite 23
Ausgabeumschaltung in Basic (PRINT - LPRINT).....Seite 24
Erweiterung für TASMON (Ausführung von NEWDOS-Kommand.)..Seite 25-26
Kurzbeschreibung des Komtek 1.....Seite 27
Die Geschichte vom armen Hacker (aus alter Chronik)......Seite 28
Joystick-Anschluß (2. Möglichkeit).....Seite 29-30
LPRINT alles (Ausgabe aller Drucker-Codes).....Seite 31-32
Der geknackte TRS-80 (int. Aufbau des Basic-Interpret.)..Seite 33-36
Adventure-Ecke: Abenteuer (aus HC Mai 84).....Seite 37-38
Genie-Minitext (ein Text-Programm).....Seite 39-40
String-Anzeige mit Volldampf.....Seite 41-44
Bücher, POKE & PEEK.....Seite 45
Basic Faster & Better (3. Teil mit Fehlerberichtigung)...Seite 46-60
Programmbibliothek abschaffen (ein Vorschlag).....Seite 61
Anmerkungen zum o.g. Vorschlag.....Seite 62
Tape Transfer (LMOFFSET-Erweiterung für Model III).......Seite 63-64
Tricks mit Strings.....Seite 65-66
Schnelleres Arbeiten mit EDTASM.....Seite 67
Hardware-Projekt (Bus-System).....Seite 67
Konvertierung TRS-80-Level-2-Basic nach MBasic.....Seite 68
Schlußwort.....Seite 68

Achtung:

Bereits im letzten Info habe ich angedeutet, daß ich ev. mein 'Amt' abgeben möchte.

Nun - ich habe mich endgültig entschlossen, daß ich die Leitung des CLUB 80 nicht weiter ausüben kann. Zwar hat mich die Abgabe der Clubbibliothek zeitlich sehr erleichtert, aber dennoch ist die Arbeit für mich noch zu viel. Ab September habe ich keine Zeit mehr. Für alle nochmals zur Erinnerung: Ich bin Student und komme ab September in die entscheidende Phase meines Studiums, das 7. und 8. Semester mit den Abschlußprüfungen und einer sehr zeitintensiven Diplom-Arbeit. Da bleibt einfach nicht genügend Freizeit für den Club. Die berufliche (und auch private) Zukunft muss einfach auch mal Vorrang haben.

Daher mein fester Entschluß, mein Amt abzugeben. Das heißt, ich suche jemanden (oder auch mehrere), die meinen Tätigkeitsbereich übernehmen. Dies wären Beantwortung der Anfragen von Interessenten, Erledigung des anfallenden Briefverkehrs, Erstellung und Versand des Clubinfos und zuletzt die Kassenführung. Wie gesagt - Übernahme durch eine oder auch mehrere Personen.

Der Club hat über 40 Mitglieder und bestimmt findet sich jemand, der weitermacht. Wer Interesse hat, der soll sich bei mir melden. Ich hoffe, daß ich nach dem Clubtreffen meinem Nachfolger (meinen Nachfolgern) alles übergeben kann (vielleicht nimmt mein Nachfolger ja auch am Clubtreffen teil). Ob der (die) Nachfolger von den Mitgliedern offiziell gewählt wird (werden), wird auf dem Clubtreffen besprochen. Sollte sich wider Erwarten niemand bereit erklären, meinen Tätigkeitsbereich zu übernehmen, müßte der Club aufgelöst werden. Ich hoffe (und bitte alle Mitglieder), daß es so weit nicht kommt.

Ich bitte alle Mitglieder um Verständnis für meine Entscheidung, die mir wirklich schwer gefallen ist, da ich nicht wegen Amtsmüdigkeit aufhöre, sondern wirklich aus obengenannten Gründen.

Euer Günther

Übersetzungen eines Blechidioten

Fremde Sprachen sind Ihnen im wahrsten Sinne des Wortes noch fremd, jenen Intelligenzmaschinen. Das menschliche »Mini-Hirn« wird hier noch gebraucht – welch tröstliche Gewißheit!

Text englisch (Matthäus 26, 41):
»Der Geist ist willig,
aber das Fleisch ist schwach.«
Übersetzung des Computers ins Deutsche:
»DER ALKOHOL IST BRAUCHBAR,
DER BRATEN IST SCHLECHT.«

Text lateinisch (mors certa hora incerta):
»Der Tod ist bestimmt,
aber die Stunde ist ungewiß.«
Übersetzung des Computers ins Deutsche:
»TODSICHER IST DIE UHRZEIT UNGENAU.«

Alber Herbert, Al[illegible]nnenstr. 20, 7732 Niedereschach, 07721/7102
Baldes Hans, Johann-Strauss-Str. 6, 8025 Unterhaching, 089/6115179
Boecker Dieter, Lehmweg 4, 2930 Varrel 1, 04451/7640
Boeckling Ulrich, Am Sonnenhang 11, 5414 Vallendar, 0261/69522
Bozek Hans Juergen, Gut Fachenfelde, 2093 Stelle, -
Buskowiak Thomas, Eschersheimer Landstr. 257, 6000 Frankfurt 1, 069/5601621
Drever Gerald, Am Speiergarten 8, 6200 Wiesbaden-Bierstadt, 06121/508218
Grajewski Werner, Zedernweg 29, 4220 Dinslaken, 02134/54573
Hailup Matthias, Junggesellenstr. 15, 4600 Dortmund, -
Held Manfred, Stirnerstr. 22, 8835 Pleinfeld, 09144/6563
Hermann Klaus, Gartenstr. 22, 7401 Pliezhausen, 07127/70024
Hummel Anton, Schubertstr. 2, 7612 Haslach, 07832/8289
Jablotschkin Rainer, Thiekamp 29, 4780 Lippstadt 8, -
Kasper Dieter, Zeppelinstr. 9, 8952 Marktoberdorf, 08342/1630
Koenig Hans J., Hebbelstr. 25, 2080 Pinneberg, 04101/209444
Konrad Josef, Anzengruber 35, 8038 Groebenzell, 08142/8494
Kuhn Eckehard, Im Dorf 14, 7443 Frickenhausen 1, 07022/45417
Marx Andreas, Mecklenburgring 48, 6600 Saarbruecken, 0681/812983
May Holger, Marienstr. 9, 5768 Sundern 2, 02935/1668
Neueder Jens, Panoramastr. 21, 7178 Michelbach/Bilz, 0791/42877
Neukam Dieter, Garstedter Weg 18, 2087 Boenningstedt, 040/5566986
Obermann Hartmut, Schwalbacher Str. 6, 6209 Heidenrod/Kemel, 06124/3913
Perschbach Patrick, Waldstr. 52, 5000 Koeln 91, 872118
Piller Walter, Rohnenstr. 8, CH-8835 Feusisberg, 01/7847418
Preuss Lothar, Lautshof 13, 2940 Wilhelmshaven, 04421/84247 (dienstl. 804-1)
Rank Heinrich, Fruehlingstr. 2, 8080 Fuerstenfeldbruck, 08141/3791
Schmidt Peter, Drei-Aehrenstr. 18, 7800 Freiburg, - Beitrag 85 noch nicht bezahlt
Schrewe Christian, Fliederweg 32, 4000 Duesseldorf 31, 0203/740897
Schroeder Gerald, Am Schuetzenplatz 14, 2105 Seevetal 1, 04105/2602
Sickmann Bernhard, Fleigenweg 2, 4430 Steinfurt 2, 02552/60344
Smerling Frank, Tangstedter Str. 5, 2080 Pinneberg, 04101/207284
Sopp Arnulf, Wakenitzstr. 8, 2400 Luebeck 1, 0451-791926
Stephan Hans-Martin, Am Glasesch 9a (Postf. 1207), 4506 Hagen a.TW., 05401/99585
Stevens Peter, Postfach 6327, 7800 Freiburg, 0761/35384
Trapp Harald, Kranichstr. 46, 4270 Dorsten 1, -
Troesch Eberhard, Altenessener Str. 414, 4300 Essen 12, 0201/342324
Trost Robert, Aalscholverstraat 9, 6921 WR Duiven (Holland), 08367/4736 Beitrag 85 bez.
Voigtlaender Holm, Haselnussweg 30, 6940 Weinheim, 06201/65241
Wagner Alexander, Theresienstr. 21c, 8224 Chieming, 08664/1500
Wagner Guenther, Gartenstr. 4, 8201 Neubeuern, 08035/3361
Wies Jean-Claude, Harthweg 9, 6600 Saarbruecken, 0681/582513
Wright Donald, Apothekenweg 5a, 8056 Neufahrn, -
Wucherer Juergen, Brauneggerstr. 14, 7750 Konstanz, 07531/29145
Zwickel Walter, Lengfelden 123, A - 5101 Bergheim, 0043662/51130

CLUBTREFFEN

Noch einmal viele Zeilen zum Clubtreffen (ergänzend zum 4. und 5. Info).

Es findet statt vom 23.-24. März 1984 im Hotel "Darmstädter Hof", Fam. Bauer, Ludwigstraße 1, 6107 Reinheim (Odenwald).

Reinheim liegt ca. 15 km (Luftlinie) südöstlich von Darmstadt am Kreuzungspunkt der B38 und B426. In der Nähe sind die Autobahnen A680 und A5.

Folgende Mitglieder nehmen meines Wissens teil:

Josef Konrad	Klaus Hermann	Heinrich Rank
Holm Voigtländer	Günther Wagner	Dieter Kasper
Gerald Schröder	Ulrich Böckling	Martin Held
Hans König	Peter Stevens	Jürgen Wucherer
Christian Schrewe	Gerald Dreyer	Hartmut Obermann
Jens Neueder	Thomas Buskowiak	Arnulf Sopp
Alexander Wagner	Walter Zwickel mit Frau	
Frank Smerling	Walter Piller mit Frau	
	Wolfgang Beckhausen mit Frau (neues Mitg.)	

Es sind also derzeit 23 Mitglieder + 3 Frauen - es freut mich riesig, daß soviele Mitglieder teils mehrstündige Anfahrtszeiten auf sich nehmen (manche würden noch gerne kommen, können aber aus Termingründen nicht).

Wenn sich noch wer anmelden möchte, so soll er sich bei mir (Günther) melden. Ich kläre dann mit dem Jürgen Wucherer ab, ob noch ein Zimmer frei ist. An dieser Stelle möchte ich dem Jürgen danken, der mir sehr viel Arbeit abgenommen hat. Er hat das Hotel organisiert und auch sämtlichen schriftlichen Kram wie Anmeldebestätigung etc. erledigt.

Am Samstag werden wir so gegen 17 Uhr mit dem 'offiziellen' Teil des Treffens beginnen. Die Zeit davor dient zum Kennenlernen, zum Fachsimpeln, zum Programm-Tausch etc. Es gibt ziemlich viel zu besprechen. Bitte bereitet Euch darauf vor, daß ziemlich viel geredet wird, aber ich glaube, es läßt sich nicht vermeiden. Der offizielle Teil endet am Sonntag so gegen 11 Uhr, da ja die meisten nach dem Mittagessen die meist längere Heimfahrt antreten werden.

Nicht versäumen möchte ich, nochmals einige Punkte anzuschneiden, die sicherlich auf dem Treffen besprochen werden:

- Programmbibliothek (weiter wie geplant? Punkte abschaffen? ev. weitere Ideen hierzu)
- Hardware-Projekt(e) siehe auch dieses Info
- Änderungen an der Clubsatzung wie: Ausschliessen von Mitgliedern; Thema Programmbibliothek; Ergänzungen zu (6); Fragen der Vorstandschaft bzw. der weiteren Clubleitung; feste Einrichtung einer Jährl. Mitgliederversammlung; usw.
- 80-Micro
- und vieles mehr

Alle Mitglieder, die am Treffen nicht teilnehmen, können sog. Eingaben machen, das heißt, daß Sie Anträge, Vorschläge etc. bei mir schriftlich einreichen können. Diese werden dann auf dem Clubtreffen behandelt.

Zur Beschlußfähigkeit des Clubtreffens:
Dieser Punkt ist in der Clubsatzung (noch) nicht geregelt. Damit das Clubtreffen einen echten Sinn hat, muß es voll beschlußfähig sein. Damit alle anderen Mitglieder über das Clubtreffen und über das, was besprochen wird, auch informiert sind, wird auf dem Treffen ausführlich Protokoll geführt. Dieses Protokoll wird im 7. Info veröffentlicht.
Lediglich Entscheidungen, die eine größere finanzielle Belastung von Mitgliedern nach sich ziehen, werden wie gewohnt über das nächste Clubtreffen beantragt und beschlossen.
(Mitglieder, die am Treffen nicht teilnehmen, können schriftliche Einwände gegen die von mir oben vorgeschlagene Beschlußfähigkeit vorbringen. Liegen mehrere Einwände vor, so wird die Beschlußfähigkeit des Clubtreffens ganz aufgehoben und alles über das 7. Info beschlossen!).

So - ich hoffe, der letzte Absatz war im Sinne aller Mitglieder. Ich bitte alle Mitglieder um geistige Vorarbeit und um Ausarbeitung von Vorschlägen, damit wir auf dem Treffen zügig alle Punkte abarbeiten können. Alles klar ?

Unser Clubmitglied Mister Cobol (Harald Trapp) würde sich freuen, wenn er folgende Geräte verkaufen könnte:

1 Laufwerk 40 Spuren/ ss/ dd
Typ BASF 6106 ca. 1 Jahr alt
Preis: 150 DM

1 Magnetkartenleser Fabrikat Borroughs
incl. 100 Karten a 10,4 (Neupreis 1200 DM)
Preis: 80 DM (originalverpackt)

Ein Angebot von einer Privatperson habe ich bekommen. Herr Dr. Peter Hampel, Ringstr. 10, 8057 Eching möchte gerne folgende Geräte verkaufen:

- Model 100 8k mit Cassettenlaufwerk (originalverpackt) 885 DM
- Model 4P 64k mit 2 Laufwerken + Programme (neuwertig) 3885 DM
- Model 2000 128k 2 Laufwerke, Colour Monitor B/W 9000 DM
  Monitor, Graphicsboard + Programme (neuwertig)
- 4-Farben-Tintenstrahldrucker CGP-220 (orig.verpackt) 1300 DM

Systemänderung :

Seit kurzem besitze ich ein neues Diskettenlaufwerk und einen Doubler. Das Laufwerk ist zwar für Double Sided ausgelegt, arbeitet zur Zeit jedoch nur Single Sided. Das bedeutet, daß ich ab sofort auch SS/DD Diskettenformate (40 Tracks) lesen und schreiben kann.

Weiterhin war meine Information im letzten Info scheinbar missverständlich. Ich kann selbstverständlich auch Datendisketten und nicht nur Systemdisk's lesen.

Hartmut Obermann

WANTED * GESUCHT * WANTED * GESUCHT * WANTED * GESUCHT * WANTED

Ich suche Jemanden, der mit mir eventuell ein D[illegible]-Laufwerk tauschen will. Ich besitze ein TEC FB5[illegible] (11 Monate alt), sowie ein TEAC FD55A (neu). Beim Kauf des TEAC-drives hat man mir zugesagt, das wäre baugleich mit meinem TEC. Doch das stimmt nicht. Es sind zwar beide Drives SS/SD und auch slimline, aber äußerlich und in der Anordnung der Stecker sind sie verschieden. Nun möchte ich gerne 2 gleiche Laufwerke haben (welche ist mir egal). Beide Laufwerke funktionieren einwandfrei. (Walter Zwickel)

Ich suche auch eine Kopie des Manuals zum TEAC FD55A!

Nächste Bitte: Vielleicht kann mir einer der Besitzer der HRG1B mal kurzzeitig die Unterlagen überlassen, damit ich klären kann, ob auch am KOMTEK eine Anschlußmöglichkeit dafür besteht. (Walter Zwickel)

Wir wären interessiert an Bauanleitungen oder billigen Bausätzen ('saubere' Bezugsquellen) für ein Modem. (Andreas Marx und Jean Claude Wies)

Ich suche Bauanleitungen für Interfaces für Schreibmaschinen. (Andreas Marx)

D R I N G E N D gesucht wird ein Nachfolger, der die Geschäfte von Günther Wagner übernimmt.

* GESUCHT * WANTED * GESUCHT * WANTED* GESUCHT * WANTED * GESU

H I N W E I S

Wenn ich irgendetwas vergessen habe,
wenn irgendwer noch irgendwas bekommen müßte,
wenn ...

dann entschuldigt das bitte und seid so nett, daß Ihr mich kurz benachrichtigt. Günther Wagner

T R E F F E N I N M Ü N C H E N

In München treffen sich relativ regelmäßig der Josef Konrad, der Dieter Kasper und der Günther Wagner. Wer an einer Teilnahme interessiert ist, sollte sich bitte bei Günther Wagner melden.

Geplant ist für den Herbst ein Treffen in München, zu dem nach Möglichkeit mehrere Mitglieder aus dem Münchner Einzugsbereich kommen sollten/könnten. Dies wäre dann ein 'kleines' Clubtreffen wie es auch in anderen Gebieten aufgebaut werden könnte/sollte. Überlegt Euch diesen Punkt doch bitte mal!

FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ?

Wer hat ein Super-Sripsit, daß unter NEWDOS oder LDOS auf dem Model III läuft ? (Günther Wagner)

Wer kann mir sagen, wie ich auf meinem Model III (dt. Ausführung) unter NEWDOS die Umlaute ausdrucken kann? (Günther Wagner)

Wer kann mir die Anweisung zu dem Programm 'TEBAST', veröffentlicht in c't, zur Verfügung stellen ? (Hans-Martin Stephan)

Wer hat Erfahrung mit Schreibmaschinen als Drucker (ev. Probleme mit Textsystemen?) (Andreas Marx)

Zum 3. Info Peek & Poke eine Frage:

```
100 IF PEEK(14312)>=127 THEN PRINT "DRUCKER ANSCHALTEN":
    GOTO 100
```

Gilt dies nur für den TRS80 III/Epson RX-80 ? Bei meiner Ausrüstung (Genie I und Star DP 510) klappt das nicht. Wie lautet die Zeile bei meinem Computer? (Werner Grajewski)

Wer hat die Auflösungen zu den Adventures:
1) Haunted House
2) Pyramid 2000
3) Peake Tu (Patrick Perschbach)

Haben sich Clubmitglieder bereits mit 'Super-Tape' (siehe auch c't 1985 Heft 1) befaßt? Liegt ein Super-Tape für den Genie I vor ? (Werner Grajewski)

Wer hat Erfahrung mit Eprom-Bänken im Genie oder TRS80 ? (Harald Trapp)

Wer kann gebrauchtes oder neues 8"-Laufwerk single oder double side besorgen? (Harald Trapp)

Wer kann mir sagen, wie ich das Disk-Basic bzw. NEWDOS mit neuen BASIC- bzw. System-Kommandos erweitern kann? (Harald Trapp)

Wer hat Service Manual bzw. Schaltpläne für den Genie? (Harald Trapp)

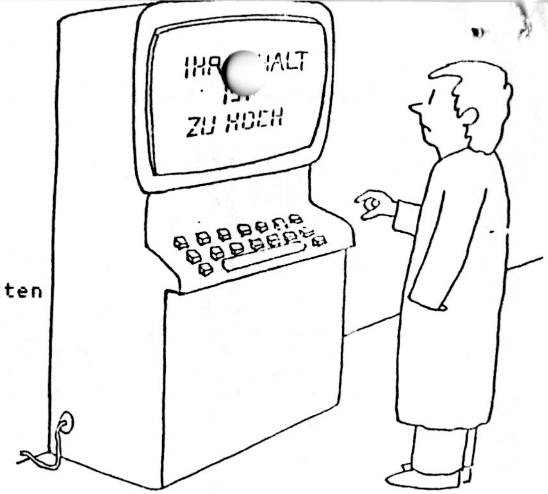
Wer kann mir einen Tip geben, wie die Umschaltung beim 'Superdoubler' (LNW-Doubler) funktioniert? Welche Daten müssen in welche Adressen, um von Single nach Double Density und von Double nach Single Density umzuschalten? (Manfred Held)

Ist es richtig, daß das Zusatz-ROM beim Genie, das auf den Adressen 3000H bis 37FFH liegt, gleichzeitig mit den Floppycontroller, der die Adressen 37E0H bis 37FFH belegt, eingeschaltet ist ? (Manfred Held)

Wer hat: Aktien-Programme ?
Programme, die mit dem Bauen zu tun haben ?
Dateiverwaltungs-Programme ? (Günther Wagner)

? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN ? FRAGEN

Neu im Club:



Kuhn Eckehard
Im Dorf 14
7443 Frickenhausen 1
07022/45417

Clubmitglied seit : 03.01.85
Punktestand = 45
System- und Drivekonfiguration :
TRS-80 Model 1 (Level 2/ 48K)/ Cassetten
bevorzugtes Betriebssystem : -

Buskowiak Thomas
Eschersheimer Landstr. 257
6000 Frankfurt 1
069/5601621

Clubmitglied seit : 03.01.85
Punktestand = 45
System- und Drivekonfiguration :
TRS-80 Model 3/ Drucker DMP 120/ 2 Laufwerke (je 40 Spuren/ ss/ dd)
bevorzugtes Betriebssystem : TRSDOS und NEWDOS

Held Manfred
Stirnerstr. 22
8835 Pleinfeld
09144/6563

Clubmitglied seit : 16.01.85
Punktestand = 45
System- und Drivekonfiguration :
Genie II/ Expander (Eigenbau)/ Drucker DRH 80/ 2 Laufwerke (je 40 Spuren/ ds/ dd)
bevorzugtes Betriebssystem : Newdos 80 V2

Baldes Hans
Johann-Strauss-Str. 6
8025 Unterhaching
089/6115179

Clubmitglied seit : 16.01.85
Punktestand = 45
System- und Drivekonfiguration :
Komtek 1S mit 32K/ Drucker Epson RX 80/ Recorder/ Monitor
bevorzugtes Betriebssystem : -

Gesegnete Mahlzeit

Zwei Hungrige sitzen im Automaten-Restaurant. Während der eine sich sofort über seine Mahlzeit hermacht, fährt der andere skeptisch mit dem Löffel durch die Suppe. Schließlich fördert er etwas zutage und meint daraufhin ärgerlich zu seinem Tischnachbarn:
»Früher fand man nur Haare in der Suppe. Heute sind es – Schrauben.«

Alber Herbert
Alemannenstr. 20
7732 Niedereschach
07721/7102

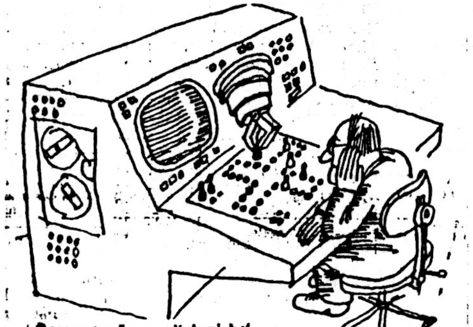
Clubmitglied seit : 10.02.85
Punktestand = 45
System- und Drivekonfiguration :
Video-Genie I/ Drucker Speedy 100-80/ Basicode-Adapter/ 1 Laufwerk (40 Spuren/ ds/ sd)
bevorzugtes Betriebssystem : NEWDOS 80

Smerling Frank
Tangstedter Str. 5
2080 Pinneberg
04101/207284

Clubmitglied seit : 14.02.85
Punktestand = 45
System- und Drivekonfiguration :
TRS-80 Model III/ Drucker RX-80 FT/ Modem/ 2 Laufwerke (je 40 Spuren/ ss/ dd)
bevorzugtes Betriebssystem : NEWDOS 80

Preuss Lothar
Lautshof 13
2940 Wilhelmshaven
04421/84247 (dienstl. 804-1)

Clubmitglied seit : 15.02.85
Punktestand = 45
System- und Drivekonfiguration :
Video-Genie 1/ Tandy Line Printer VII/ GP 100/ 2 Laufwerke (je 40 Spuren/ ss/ sd und dd)
bevorzugtes Betriebssystem : TRS-DOS und G-DOS


'Computer ärgere dich nicht'

Das Bankgeheimnis

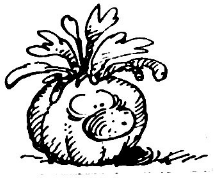
Es war im Hochsommer, und der Mann hinter dem Bankschalter hatte die dunkle Sonnenblende tief herabgezogen. Die Kundin, die jetzt an den Schalter trat, konnte ihn daher nicht sehen – wohl aber er sie. Der Mann begrüßte die Kundin und schob das Zahlbrett heraus. Sie blickte mißtrauisch auf das Brett, legte ihren Scheck darauf und zog hastig die Hand zurück. Der Scheck verschwand, und das Brett kam mit einigen Banknoten wieder hervor.
Die Dame blickte sich um, als fürchte sie einen Lauscher. Dann beugte sie sich vor und sagte: »Ich weiß, du bist nur ein Automat. Trotzdem, vielen Dank!«

Es stellt sich vor: W e r n e r G r a j e w s k i

Am 14.09.1951 in Duisburg geboren, bin ich nach dem Realschulabschluß 1968 in die Finanzverwaltung eingetreten. Seit 1982 bin ich bei der Großbetriebsprüfungsstelle in Essen tätig. Beruflich befasse ich mich seit 1980 mit Computern. Sharp PC 1211, Sharp PC 1500, Sharp PC 1251, Epson HX-20 und Genie I war die Reihenfolge der Geräte, mit denen ich mich befaßt habe bzw. ncoh befasse. Die Geräte habe ich bisher nahezu ausschließlich zur Unterstützung meiner beruflichen Tätigkeit eingesetzt.

Einige Bücher und VHS-Kurse versetzten mich in die Lage BASIC-Programme im Bereich Steuerrecht zu entwickeln. Dies führte soweit, daß ich mit einem Kollegen bereits 2 Bücher geschrieben habe. Im Februar beginnen die Arbeiten für das 3. Buch. Die Bücher und Programme sind zwar für den Epson HX-20 geschrieben, jedoch habe ich auch eine Version auf dem Genie I erstellt. Die Programme kann ich dem Club laut Vereinbarung mit dem 2. Autor nicht kostenlos zur Verfügung stellen. Soweit jedoch Interesse bei einem Clubmitglied in Sachen Steuerrecht vorliegt, kann ich die Programme oder Bücher zu Sonderkonditionen besorgen.

Das Genie I kenne ich nur ungenügend und kann auch nur in BASIC Programme erstellen. Dies hat mich veranlaßt, Mitglied im CLUB 80 und in einem weiteren Club zu werden.



„Was, Sie haben 30 000 Mark Schulden und wollen meine Tochter heiraten?" – „Allerdings, oder wissen Sie einen anderen Ausweg?"

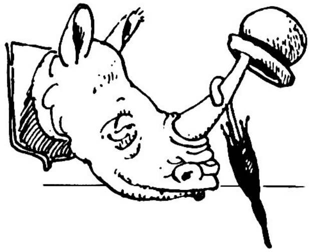
Mein Name ist Dieter Kasper, bin 22 Jahre alt und habe soeben in diesen Tagen das 3te Semester Mathematik an der TU-München hinter mich gebracht. Seit kurzem besitzte ich auch eine Floppy-Station (2x80 ds/dd unter Newdos80/2) für mein Genie 1 '83er Modell; auf einen Drucker glaube ich noch verzichten zu können bis sich ein günstiges Angebot ergibt, während eine Erhöhung der Taktfrequenz mir sehr willkommen wäre.

Meine Begeisterung zum Computer wurde eigentlich erst in der Kollegstufe durch die bekannten cbm 4xxx geweckt. Auch im Bewußtsein der Nützlichkeit für das spätere Berufsleben habe ich dann die Investition für den listigen Rechenknecht gewagt. Mein Interesse liegt in der Anwendung und Erstellung von Programmen (Hardware ist so ne Sache) und so versuche ich mich zur Zeit mit Pascal, Fortran und Assembler etwas näher anzufreunden.

Ich selbst erhoffe mir von "alten, erfahrenen Hasen" hilfreiche Tips, wenn ich mal irgendwo nicht weiter weiß und das liebe Gerät absolut nicht das tut, was ich erwarte. Meinerseits wünsche ich, daß ich Anderen in mathematischen, physikalischen und chemischen Problemen etwas weiter helfen kann.

Persoenliche Vorstellung von - Manfred Held -

Ich heiße Manfred Held, bin 30 Jahre alt und von Beruf Ausbilder fuer Informatioselektroniker bei der DB AW - Nuernberg.
Nachdem ich einsah, daß ich das Betriebssystem in Hex-dez. fuer meinen Computer Marke "Eigenbau" nicht selbstschreiben konnte, habe ich mir 1982 einen Genie II gekauft.
Die Erweiterungen habe ich selbst gebaut. (Faedeltechnik)
Mein System besteht aus - Genie II, 48K Speicher, 2 Laufwerken DS/DD, Epromer (aus Elcomp 10/82), DHR 80, SD 4035. -
Derzeit baue ich einen neuen Genie (Richtung Genie III S).
Mein bevorzugtes Betriebssystem ist Newdos 2.0
Durch den CLUB 80 erhoffe ich einen Erfahrungsaustausch und Tips.



Hallo Clubmitglieder ,

ich freue mich nun Mitglied im Club 8o zu sein und möchte mich kurz vorstellen .

Ich heiße Hans Jürgen Bozek , bin am Nikolaustag 1954 in einem kleinen Vorort von Hamburg geboren und heute als HF-Techniker tätig .
Schon früh kam ich durch meinen Beruf mit der Digital-Technik und später mit der Micro-Technik in Berührung und beschäftige mich daher seit etwa vier Jahren mit Computern .
Nach dem CT 65 und dem ZX 81 kaufte ich mir vor zwei Jahren den Komtek 1 (Kurzbeschreibung anbei) .
Nach anfänglichen Schwierigkeiten mit dem TRS 8o - Nachbau verbringe ich nun einen großen Teil meiner Freizeit mit der Herstellung von Verwaltungsprogrammen .
Trotz fehlender Unterstützung meines Komtek-Lieferanten ist es mir endlich auch noch gelungen NEWDOS 8o auf der Maschine zum Laufen zu bringen und somit in weitere Dimensionen meines Hobbys vorzudringen .

Mit freundlichen Grüßen

Hans Jürgen Bozek

Kölner Erfolgsmodell

Auch für Tünnes und Schäl, die beiden Kölner Originale, bleibt die neue Zeit der Elektronik nicht ohne Folgen: Während Schäl arbeitet wie jeder anständige Mensch, geht Tünnes spazieren. Schäl, der genau weiß, daß sein Freund über keinen roten Heller verfügt, wundert sich darüber und stellt ihn zur Rede. Tünnes erklärt ihm, er habe sich einen Klein-Computer angeschafft, der ihm jetzt wie ein Sekretär alle Arbeit abnehme.
Schäl empört sich: »Do Jeck, dat Ding koss doch enen Haufen Jeld. Wie willste dat dann bezahle?«
Meint Tünnes zuversichtlich: »Dat Problem is er jo jrad am löse!«

```
==============================================================
```

Nachdem wir uns im letzten Teil hauptsaechlich mit allgemeinen Hinweisen und anschliessend mit der ersten von vier DIVISIONS beschaeftigt haben, soll es nun hier an's Eingemachte gehen.

Zunaechste eine kleine Wiederholung:

```
1----67A---B------------------------------------- usw. ... 72
       IDENTIFICATION DIVISION.
      *************************
       PROGRAM-ID.        WAHL.
       AUTHOR.            HARALD TRAPP.
       INSTALLATION.      KRANICHSTR 46     4270 DORSTEN 1.
       DATE-WRITTEN.      10/12/83.
      *DATE-COMPILED.     10/24/83.
       SECURITY.          KEINE.
      *REMARKS.           AUSWAHLMENUE FUER STAMMDATENPFLEGE.
      *
      *
```

Das war also die erste DIVISION die wir im ersten Teil besprochen haben. Sind die Bedeutung der Spalten 1 bis 6, der Spalte 7, der Zone A und der Zone B noch in Erinnerung ?? (sonst schnell noch einmal nachschauen)

Es geht also um die Programmeinleitung.- Der Stern (*) in der Spalte 7 bedeutet, dass es sich hier um Bemerkungen handelt. Die Befehlswoerter, die teilweise hinter den Sternchen stehen, sind im Standard COBOL enthalten - nicht so im COBOL fuer TANDY und aehnliche. Aus diesem Grunde und zum Zweck der vollstaendigen Information stehen diese Kommandos hinter den bewußten Sternchen.

An dieser Stelle sei bemerkt, dass sich diese Programmeinleitung auf ein von mir entwickeltes Programmpaket (ADRESSEN TEXT) bezieht. Im Zuge dieses kleinen Kurs werde ich auf einige Punkte dieses Programmpakets eingehen und erlaeutern. ADRESSEN TEXT ist ueberigens fuer alle Interessierte aus der CLUB80 Programmbibliothek abrufbar, das komplette Paket besteht aus vier Disketten, der Objekt-, Daten- und Reorganisations-Diskette, sowie einer Diskette mit den COBOL-Sources.

Nach diesem Kurs besteht fuer alle Interresierten die Moeglichkeit das Programm zu verbessern und zu erweitern.

So, nun aber genug der Werbung fuer ADRESSEN TEXT, kommen wir zur zweiten von vier DIVISIONS, der ENVIRONMENT DIVISION.

Frei uebersetzt koennte man dazu sagen:'Maschinen/Geraete Erklaerung'. Damit liegen wir gar nicht so falsch. Tatsaechlich werden hier allen physischen Dateien (und Peripherien) logische Name zugeordnet und die Dateien naeher erklaert.

Diese Dateien koennen z.B. Files auf dem Laufwerk (1) oder dem Laufwerk (2) sein. Aber auch der Drucker wird vom COBOL wie eine (sequentielle) Datei behandelt.

Weiterhin gibt die ENVIRONMENT DIVISION Auskunft ueber den SOURCE und den OBJECT Computer. ~~Auskunft~~. In unserem Falle ist das immer 'unser' Rechner, nicht aber so bei Grossanlagen.

Als letzte Information erhaelt die ENVIRONMENT DIVISION Angaben ueber den logischen Aufbau unserer Dateien, also sequentieller Aufbau, Randon oder Key-Index (dazu spaeter mehr).

```
1----67A---B--------------------------------------- usw... 72
       ENVIRONMENT DIVISION.
      **********************
       CONFIGURATION SECTION.
       SOURCE-COMPUTER.       GENIE.
       OBJECT=COMPUTER.       GENIE.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
       SELECT DATUM ASSIGN TO RAANDOM, "HPTDATUM/DAT:1"
              ORGANISATION IS RELATIVE
              ACESS MODE IS RANDOM
              RELATIVE KEY IS SATZ.
       SELECT DATEN ASSIGN TO RANDOM, "STAMMDAT/DAT:1"
              ORGANIZATION IS INDEXED
              ACCESS MODE IS RANDOM
              RECORD KEI IS E-NUMMER
              ALTERNATE RECORD IS E-NAME.
      *
```

Nur keine Angst vor vielen Fachausdruecken - es ist wirklich nicht so schwer wie es am Anfang immer aussieht.

Als erstes eine Erklaerung:
Beim Cobol-Programm gibt es einen gewissen Aufbau. Die vielfach genannten DIVISIONS geben die grobe Struktur. Untergeordnet den DIVISIONS sind die SECTIONS, von denen kann es eine unbestimmte Anzahl geben. Darunter geordnet bestimmen die PARAGRAPHEN die weitere Struktur und darunter sind es die einzelnen Befehls-Woerter.

Nach einem kurzen Blick auf die ENVIRONMENT DIVISION wird man also feststellen, dass diese Division aus zwei SECTION's besteht. Die erste (CONFIGURATION-SECTION) gibt Auskunft ueber die verwendeteten Computer. Die zweite SECTION (INPUT-OUTPUT SECTION) bestimmt unser 'File-Handling'.

Die INPUT-OUTPUT SECTION unterteilt sich in die FILE-CONTROL und die I-O CONTROL. (Um letztere brauchen wir uns nicht zu kuemmern, da Sie nur ganz selten benutzt wird, die Funktion soll aber nicht verschwiegen werden: Um Hauptspeicherplatz zu sparen kann man ein und den gleichen Speicherbereich fuer mehrere Dateien gebrauchen, man reserviert also z.B. 200 Bytes fuer Stammdaten und kann diese Bytes auch gleichzeitig fuer irgendeine andere Datei benutzen).

Aber wie gesagt, fuer uns nicht interessant. Bleibt also noch die FILE-CONTROL.
Hier wird erklaert, welche logischen Datei-Namen den physischen Dateien zugeordnet werden. In unserem Beispiel sind das: DATUM und DATEN.

SELECT DATUM ASSIGN TO RANDOM bedeutet im Klartext: ordne dem Namen DATUM die DISKETTENDATEI (RANDOM) "HPTDATUM/DAT" im Schacht 1 zu.(siehe Beispiel)

Also: SELECT bedeutet zuordnen
ASSIGN TO RANDOM bedeutet 'zu einem Diskettenlaufwerk'

'HPTDATUM/DAT:1' gibt den Manen der phys. Datei und die Laufwerksnummer an.

ORGANIZATION IS RELATIVE, diese Aussage ist wohl ganz klar, es handelt sich bei der genannten Datei um eine RELATIV-DATEI.

ACCESS MODE IS RANDOM heisst in diesem Zusammenhang, dass uf die Datei ueber eine bestimmte Satznummer direkt auf einen bestimmten Satz zugegriffen werden kann, die Bezeichnung der Variablen der Satznummer wird in der naechsten Zeile des ObjectProgramms genannt:

RELATIVE KEY IS SATZ, der Schluessel, also die Satznummer wird durch die Variable SATZ bestimmt. (Energische Basic-Programmierer: Variablenbezeichnungen nicht mit z.B S1(Dollar) oder AB) Variablennamen koennen bis zu 32(!) Stellen lang sein.

SELECT DATUM...... usw. heisst also nichts weiter als das ich eine Datei habe in der das Datum gespeichert wird, meine Satznumer steht in der Variablen SATZ, es handelt sich um eine RELATIVE DATEI auf dem Laufwerk (1) mit dem Namen 'HPTDATUM/DAT', undd immer dann, wenn ich diese Datei ansprechen will, kann ich das ueber den logischen Namen DATUM machen. (z.B Schliessen der Datei 'HPTDATUM/DAT': CLOSE DATUM)

Alles klar ?? Ich glaube schon.... weiter geht's:

SELECT DATUM ASSIGN TO RANDOM .... ist wohl jetzt nicht mehr schwierig, es verhaelt sich alles so, wie oben beschrieben.

Aber jetzt: ORGANIZATION IS INDEXED

Key-Index Dateien werden dem reinen Basic Programmierer noch nicht untergekommen sein. Diese Dateiform hat fantastische Moeglichkeiten (und das nicht nur zur Stammdatenverwaltung). Vom BASIC her kennt man nur sequentielle und Random (Relative) Dateien (also direkter Zugriff auf eine bestimmte Information ueber die Satznummer) Bei einer Stammdatenverwaltung wuerde das z.B heissen, dass ich dem Meier den Satz (und damit auch die Numer) (1) zuweise, dem Karl die Satz'Nummer' (2) usw.

Moechte ich nach einer bestimmten Zeit alle Daten vom Meier abrufen, so habe ich mit hundert % iger Sicherheit die Nummer vegessen, oder ich sehe in einer Liste nach (die dann aber auch bereits die gewuenschte Informationen enthalten kann), und somit wird der Computer schon wieder unnuetz.

Nicht so bei Key-Index Dateien. Bei dieser Form der Datenspeicherung wird den eigentlichen Daten ein Index vorgestellt. Also eine Art Inhaltsverzeichnis - das ist noch nicht unbeding etwas besonderes. Stellen wir uns ein umfangreiches Nachschlagewerk vor. Wir suchen nach einem bestimmten Begriff un schlagen das Stichwortverzeichnis auf. Dieses Verzeichnis ist meist immer alphabetisch geordnet.

Ubertragen wir das nun einmal auf unsere Stammdaten-Datei. Wir haben z.B. drei verschiedene Meier gespeichert. Wir wissen also, das wir in unserem Stichwortverzeichnis nach mindestens dem Begriff:'MEIE' suchen muessen. Wir erhalten nun die Information:
Meier, Alfred   Band 10, Seite 376

Meier, Gustav   Band 13, Seite 561
usw.

Nun wissen wir, wo wir unsere 'Meier' finden koennen, wir suchen aber den Meier, der am 17. Juni Geburtstag hat. Also eine Verknuepfung (quasi eine kleine Datenbank). Nun, so etwas laesst sich alles (wenn auch mit Einschraenkungen) mit einer Key-Index Verwaltung vornehmen.

Kommen wir auf den Boden der Tatsachen zurueck. In Verbindung zu meinem ADRESSEN TEXT bedeutet Key-Index Verwaltung, dass ich Daten zu einer bestimmten Person wahlweise uber das erste Verzeichnis(Index-1), die Nummer abrufen kann (Nummer ist nicht unbedingt gleich die Satznummer, Unsere Nummer kann auch alphanumerisch sein), oder aber einen Zugriff direkt ueber den Namen habe (ich gebe also direkt den Namen:'Meier' ein und erhalte die gewuenschte Information). Man kann sogar, wenn man z.B. nur 'Mei' eingibt, von dieser Position aus Blaettern (also sequentiell suchen....).

Um auf die SELECT DATEN ASSIGN .... Anweisung zurueckzukommen:

Es wird die Aussage gemacht, das alle Information dem logischen Namen DATEN, der physischen DATEI 'STAMMDAT/DAT:1' zugeornet werden, die Datei INDEX-organisiert ist, der Zugriff direkt erfolgt und zwar wahlweise ueber den INDEX E-NUMMER oder den INDEX E-NAME.

E-NUMMER und E-NAME sind hierbei Variablen, E-NUMMER bedeutet hier Eingabe-Nummer und E-NAME Eingabe-Name. Anders ausgedrueckt, mein erster Zugriffsschluessel ist eine (wahlweise alphanumerische) Nummer, der zweite Schluessel ist der Name.

Der Vollstaendigkeit halber sei noch erwaehnt, dass ich einer Datei bis zu 8(!) Scluessel zuordnen kann (also z.B. den Wohnort, Geburtstag usw.) Fuer doppelte Schluessel (es koennen ja mehrere gespeicherte Personen in ein und dergleichen Stadt wohnen) gibt es eine spezielle Klausel (.... WITH DUPLICATES).

Man kann natuerlich auch rein sequentielle Dateien behandeln:

```
SELECT DATEI ASSIGN TO RANDOM, 'BRABRA/DAT:0'
       ORGANIZATION IS SEQUENTIAL
       ACCESS MODE IS SEQUENTIAL.
```

Man kann aber auch relative Dateien sequentiell, random oder dynamic behandeln (also sequentiel durcharbeiten, z.B fuer Stammdatenlisten, random (direkter Zugriff ueber Satznummer) bearbeiten, oder aber sequentiell und random -dynamic bearbeiten)
Im Klartext heisst dass, das ich eine durch OPEN geoffnete RELATIVE DATEI ohne Ruecksicht in ein und demselben Programm sequentiell, mit Direktzugriff oder sogar beidem bearbeiten, sprich auf den Datensatz zugreifen kann.

```
SELECT DATEI ASSIGN TO RANDOM, 'BRABRA/DAT:1'
       ORGANIZATION IS RELATIVE
       ACCESS MODE IS SEQUENTIAL, REALTIVE KEY IS xxxxx
             oder     RANDOM     , RELATIVE KEY iS zzzzzz
             oder     DYNAMIC.
```

(bei RELATIV-DATEIEN muss bei sequentiellem oder direktem

Zugrif die Satznummer angegeben werden, der REL[illegible]IVE KEY).

Bei den INDEX-Dateien sieht es folgendermassen aus:

```
SELECT DATEI ASSIGN TO RANDOM, 'BRABRA:1'
        ORGANIZATION IS INDEXED
        ACCESS MODE IS  SEQUENTIAL
            oder        RANDAOM
            oder        DYNAMIC
        RECORD KEY IS xxxxxxxxxxxxxx (1.Schluessel)
        ALTERNATE RECORD KEY IS zzzz (2.Schluessel)
        ALTERNATE ...... (usw.)  (eventuell WITH DUPLICATES).
```

Einen Drucker kann man so 'selectieren':

```
SELECT DRUCK ASSIGN TO PRINT, "PRINTER"
        ORGANIZATION IS SEQUENRTIAL
        ACCESS MODE IS SEQUENTIAL.
```

Der geneigte Leser wird sofort erkennen, dass der Drucker wie eine sequentielle Datei behandelt wird, der 'offizielle' Name ist 'PRINTER' die Zuweisung is statt 'RANDOM' 'PRINT'.

Mit diesen Informationen waere auch die zweite DIVISION abgehandelt. Die dritte, die DATA DIVISION erklaert alle im Programm benutzten Variablen, also nicht nur die Datensatzbeschreibungen (die Variablen der Dateien) sondern auch alle Arbeitsfelder (Variablen) mit Hinblick auf 'String'-Variablen, (String's gibt es eigentlich gar nicht im herkoemmlichen Sinn), numerischen Datenfeldern, und als Besonderheit, den Edit-Datenfeldern (Edit-Datenfelder kann man ganz entfernt mit der Print USING Anweisung im Basic vergleichen). Aber Cobol waere nicht Cobol, wenn nicht auch hier eine Besonderheit (eine bequeme und hilfreiche Besonderheit) gegenueber Basic dem Programmierer geboten werden wuerde.

Die DATA DIVISION unterteilt sich in die FILE, WORKING-STORAGE und die LINKAGE SECTION. Die letztere Section, die LINKAGE SECTION wollen wir hier gar nicht weiter behandeln, es soll aber kurz ihre Bedeutung erklaert werden.

Will man bei einem Programmablauf von einem Programm in ein anderes springen, ohne daß man die fuer das zweite Programm wichtigen Daten im ersten Programm in eine Datei schreibt die im zweiten Programm wieder ausgelesen wird, so ordnet man die Datenfelder, die fuer zwei oder mehrere Programme benoetigt werden in der LINKAGE SECTION an. Bei einem CALL hat dann das aufgerufe Programm alle Datenfelder des aufrufenden Programmes zur Verfuegung.

Zum Abschluss dieses zweiten Teils sei nur noch erwaehnt, dass in der FILE SECTION alle Datenfelder der mit der SELECT Klausel zugewiesenen Dateien definiert werden, in der WORKING-STORAGE SECTION dann die reinen Arbeits und Rechenfelder bezeichnet werden.

Harald Tr[illegible]

4270 Dorsten
Kranichstr. 46
Tel. 02362/42497

Herrn
Klaus Hermann
Gartenstr. 2

7401 Pliezhausen

Ihre Zeichen/Nachricht	meine Nachricht	Datum
13.11.84		20.11.84

Betreff:
Druckerpuffer

Hallo Klaus,

vielen Dank fuer Dein Schreiben. - Sehr gern bin ich bereit die Funktionsweise des Druckerpuffers naeher zu erlaeutern.

Natuerlich wird durch den Einsatz einees Druckerpuffers nicht der Drucker selbst schneller (die Geschwindigkeit ist nicht zuletzt durch die Druckermechanik festgelegt und kann nicht ohne weiteres beeinflusst werden).

Der Anwender allerdings kann bei Ausdrucken die Wartezeit drastisch verkuerzen. Der Rechner muss beim Einsatz eines Drucker-Zwischenpuffers nicht mehr auf den Drucker warten, sondern kann die vom Rechner mit Systemgeschwindigkeit ausgegebenen Druckdaten ohne Zeitverlust aufnehmen.

Ich habe, um genaue Zeitangaben machen zu koennen, folgenden Test durchgefuehrt:

Ausdruck eines Assembler Listings von einer Laenge von recht genau 15 Kb.

Ohne Druckerpuffer,	Ausdruck auf Microline 80 (80 Zeichen/sec) Rechner frei fuer andere Aufgaben nach **14 Minuten !**
Mit Druckerpuffer,	Ausdruck auf Microline 80 Rechner frei fuer andere Aufgaben schon bereits nach **13 Sekunden (!!)**

DATENFERNUEBERTRAGUNG (DFUE)

Frage:
Akustikkoppler selbst bauen oder lieber kaufen ??

Ein Bericht von Harald Trapp

Datenfernuebertragung als Schlagwort - es ist schon mehrmals im Club-Info gefallen. Bereits im Info Nummer vier habe ich einen Bericht ueber den Einsatz des Akustikkopplers der Firma EPSON abgegeben. Inzwischen wurden auch viele verschiedene Anregeungen einer fuer Datenfernuebertragung noetigen V.24 Schnittstelle gemacht (Interface und Software von RB electronic ca. 240 DM) und nicht zuletzt eine Eigenbauversion von unserem Mitglied W.Zwickel.

Tja, was braucht man fuer DFUE ?? Es wurde schon gesagt, man benoetigt einen Akustikkoppler (oder einen MODEM), eine serielle Schnittstelle und Kommunikationssoftware.

Zunaechst zum Modem:
Ein Modem wirde dem Computeranwender von der Post zur Verfuegung gestellt, es kostet 80 Dm im Monat (fuer 300 Baud Uebertragungsrate = ca. 35 Zeichen/sec), eine einmalige Anschlussgebuehr von 80 DM wird zusaetzlich faellig und gekoppelt wird ein Modem nur mit Rechnern mit FTZ-Zulassungsnummer-. Diese Loesung faellt also fuer den zukuenftigen DFUE-isten flach (Erwaehnt werden soll allerdings noch, dass mit einem galvanisch gekoppelten Modem ein fast stoerungsfreier Betrieb moeglich ist, man kann nicht nur Rechner anwaehlen, sondern sich sogar anwaehlen lassen. Das heisst,der Rechner ist in Betrieb un kann jederzeit von aussen durch irgendeinen Benutzer angewaehlt werden (wie die ueblichen Mailboxen)).

Fuer Club 80 Mitglieder bbt also nur noch der Akustikkoppler, doch dazu spaeter mehr. Was benoetigt man noch ?? Eine Schnittstelle, ueber die die Daten in den Computer 'rein' und aus den Computer 'raus' koennen. Diese Schnittstelle funktioniert aehnlich einer Druckerschnittstelle, die Daten werden bei einer Centronics-Druckerschnittstelle jedoch jeweils 8 bit parallel ausgegeben, bei einer Schnittstelle fuer Akustikkoppler (und auch andere Anwendungen) bit-seriell. Man nennt so eine genormte Schnittstelle 'V.24' oder 'RS 232'. (Unterschiede liegen hier bei der Steckerebelegung od. Ausfuehrung, denn obwohl genormt, Ausnahmen bestetigen die Regel).

So eine Schnittstelle kann man entweder kaufen, z.B rb electronic (mit entsprechender Kommunikationssoftware) oder aber selber bauen (z.B. in Faedeltechnik, z.B. die Schnittstelle von unserem Mitglied W. Zwickel). Wird die Schnittstelle von W. Zwickel noch mit V.24 Treibern, einer symmetrischen Speisung von ca. +/- 12 Volt versehen und eine Software fuer diese Schnittstelle geschrieben, so hat man fuer vergleichsweise sehr wenig Geld eine mindestens ebenbuertige Schnittstelle zur Verfuegung.

Hat man die Schnittstelle und die Software, dann fehlt eigentlich nur noch der Akustikkoppler, und, damit auch alles gesetzlich geregelt ist, sollte der Akustikkoppler eine FTZ-Nummer besitzen. Diese Koppler kosten leider einiges an Geld. Obwohl die Preise fuer Koppler gefallen sind, so liegen

sie doch ... zwischen 300 DM und 800 DM.

Man kann so etwas auch selber bauen (ohne FTZ-Nummer aber mit Risiko und hier beginnt meine Geschichte:)

Durch zahlreiche Beitraege in der Fachpresse und nicht zulezt durch den Einsatz von Akustikkopplern im Betrieb wurde mein Interesse geweckt. Nachdem ich vom Betrieb aus mit mehreren Boxen Kontakt aufgenommen habe,wollte ich dies auch von zu Hause aus. Also wurde eine V.24 Schnittstelle besorgt (die von rb electronic, die ich immer wieder nur loben kann)
und, da das Geld nun ziemlich verbraucht war, das Modem Sonderheft mit einer Bauanleitung fuer einen Akustikkoppler.

Da ich mich schon sehr lange mit Elektronik beschaeftige und auch entsprechend labormaessig mit Oszilloskop und aenlichem ausgeruestet bin, sollte mir der Nachbau nicht schwerfallen.

Also wurde ruck-zuck eine Platinenfolie vom Layout angefertigt, eine CU-Platte belichtet, geaetzt und gebohrt. Der erste Schritt war getan.

Bauteile wurden aus einem nahegelegenen Shop zu recht humanen Preisen gekauft, Standardwiderstaende mit 10 bis 20% Toleranz, hochwertige Folienkondensatoren und geprufte Transistoren im Metallgehaeuse und zulezt noch ein paar Schalterchen.

Die Platine wurde bestueckt, ein Steckernetzteil mit 12 Volt angschlossen, als Schallwandler dieneten zwei Miniatur-Lautsprecher. Ein munteres Pfeifen war nach dem Anlegen der Versorgungsspannung zu hoeren, legte man das Schalterchen um, so wurde es sogar noch hoeher. Mir war klar, de Modulator funktionierte.

Nun aber fluss ans Abgleichen (hierzu dienten vier Trimmer). Zwei Trimmer fuer Modulator, zwei fuer den Demodulator, jeweils Sende und Empfangskanal. Der Abgleich wurde mit dem Oszilloskop vorgenommen, Kanal A auf 980 Hz, Kanal B auf 1650 Hz. Mit einem aus dem Betrieb geliehenen Koppler wurde mein Abgleich kontrolliert, den Koppler am linken Ohr, den Lautsprecher am rechten Ohr wurden die erzeugten Toene verglichen. Sie stimmten (wie solls auch anders sein) nicht ganz ueberein - woher nun die dritte Hand zum Drehen an dem Trimmer ?? Irgendwie ging's schliesslich doch und ich habe beide Frequenzen auf Schwebungsnull ableichen koennen.

Doch was waren das vorher fuer Experimente ?? Mit den angegeben Bauteilewerten war die untere Frequenz ueberhaupt nicht zu erreichen - statt 56 Ko musste ein Widerstand von 82 Ko eingesetzt werden (ob man hier noch von Toleranzen sprechen kann??). Der Wiederstand kam in die Platine und somit auch der gewuenschte Ton aus dem Lautsprecherchen.

Das war der zweite Schritt - schon sehr erfolgreich. Nun musste ueberprueft werden, ob der Modulator auch von 980 Hz auf 1180 und von 1650 auf 1850Hz schaltete. Ein Widerstand nach +12 Volt schaltete den Modulator um, ein Vergleich mit dem geliehenen Kopler - schauderhaft schaurige Uebereinstimmung der zweiten Frequenz, diese war nicht ueber Trimmer zu beeinflussen, die zwei verbliebenen Trimmer waren fuer die PLL, also fuer das 'Einrasten' des Demodulatorteils und nicht fuer die Frequenz zustaendig. Ich redete mir ein, dass diese Abweichungen im Rahmen der Toleranz lagen, kurz ueberprueft, und bestaetigt.

MONITOR im GENIE

Mehr durch Zufall stieß ich auf den im Genie (83er Modell) eingebauten Monitor mit der Startadresse 13577 Dez, 3509 Hex.

Bevor ich die mir bekannten Befehle aufzeige etwas Allgemeines:

Adressen werden in hexadezimaler Form eingegeben, ohne Führungs-Null und End-H (zB. A3C00), wobei die vier rechtsbündigen Stellen zählen (also A3F74200 = A4200). Bei einer Speicheranzeige kann das Scrolling durch SPACE gestoppt und mit ENTER fortgesetzt werden; BREAK unterbricht einen Befehl.

A....	Speicheranzeige ab einer Adresse in ASCII + Klötzchengrafik
D....	- " - HEX
E....	Editieren, schreiben in den Speicher in Hex
H.....	Umrechnen von Dez in Hex. Hier Eingabe in Dez.
J....	Springen zu einer Adresse
X....	Die angegebene Zahl ist X, die höchste adresse ist Y Anzeigen von X Y X+Y X-Y Y-X in Hex,Dez.

Das sind die Befehle die tadellos funktionieren. Drei weitere machen keinen so rechten Sinn: S.... (Search?) zeigt die Speicherplätze in Hex an, aber nur untereinander. R.... springt zurück zu READY?. P.... .... (Punch von bis Einsprung) gestattet zwar diese Eingabe kommt auch mit NAME? zurück, verfährt aber nach diesen Namen genau wie bei R.

Nun ist mir zwar bekannt, daß die GENIE-Geräte zwischen 3000 und 37FF ein 1,5K EPROM eingebaut haben, welches vor März 82 ein Monitorprogramm enthielt. Heute sitzt hier aber das Level IV ROM, das Komfortroutinen zur Verfügung stellt (Groß- Kleinschrift, ä,ö,ü, Ä,Ö,Ü,ß, Ausgabe des Bildschirminhaltes auf den Drucker, Full-Screen-Editor und Shorthand = bis zu 4 Befehle/Grafiksymbole auf einer Taste). Möglicher Weise ist die Überdeckung der beiden nicht vollständig und der übrig gebliebene Rest ist noch halb funktionstüchtig. Wer weiß mehr darüber ??

Dieter Kasper

Wußten Sie schon, daß.....

manche seine eigenen Fingernägel zum Fressen gern hat?

niemand weiß, ob kein Nichtraucher, der nicht raucht, ein Raucher ist oder Nichtraucher?

Ausgabeumschaltung in BASIC (Josef Konrad)

Mit dem folgenden Programm läßt sich die Ausgabe bei BASIC - Programmen während des Laufs abhängig von einer Benutzereingabe entweder auf Drucker oder Bildschirm einstellen.

```
 30 INPUT"AUSGABE AUF <B>ILDSCHIRM ODER <D>RUCKER ";O$
 40 IF O$="B" THEN TK=178 ELSE TK=175 :REM TOKEN FUER PRINT BZW. LPRINT
 50 A$="A" : GOSUB 100: LPRINT"AUSGABE ":LPRINT"WEITERER TEXT....":STOP
100 P=VARPTR(A$):P2=PEEK(P+1)+256*PEEK(P+2)
110 FOR X=P2+3 TO P2+256:IF PEEK(X)=0 THEN RETURN ELSE IF PEEK(X)=178
    OR PEEK(X)=175 THEN POKE X,TK
120 NEXT X : RETURN
```

In Zeile 40 erhält TK den Wert des Tokens für PRINT oder LPRINT.

Die Variable A in Zeile 50 ist eine 'Dummy'-Variable, sie wird benötigt, um in Zeile 100 die Lage der Zeile 50 im Speicher bestimmen zu können.

In Zeile 110 wird nun die Zeile, die 'A' enthält, durchsucht und alle bis zum Ende der Zeile auftauchenden PRINT-oder LPRINT-Statements entsprechend der Eingabe in Zeile 40 evtl. geändert.

Vor oder nach den LPRINT's oder PRINT's in Zeile 50 können auch noch weitere Statements eingefügt werden.

Das Unterprogramm ab Zeile 100 kann auch noch aus anderen Programmzeilen aufgerufen werden, diese müssen nur die Anweisung A$="A" enthalten.

Literatur: 80 Micro 3/83

Josef Konrad

//

Universell anwendbare Regeln für gutgläubige Ingenieure:

1) In jede Berechnung wird sich auch jeder mögliche Fehler einschleichen.

2) In jeder Formel sind alle Konstanten - besonders solche, die aus Ingenieur-Handbüchern stammen - als Variable zu betrachten.

3) Hauptänderungen einer Entwicklung werden immer dann notwendig, wenn bereits die Fertigung läuft. Aufbau- und Betriebsanleitungen, die zusammen mit den Geräten versandt werden, werden bei der Wareneingangskontrolle sofort weggeworfen.

4) Wenn mehr als eine Person für eine Fehlkalkulation verantwortlich ist, ist keiner der Schuldige.

Nur noch zwei Trimmerchen (PLL) und es kann fas [los]gehen. Ueberpruefen des Demodulators. Dazu habe ich eine kleine Schaltung fuer einen V.24 Treiber aufgebaut, zwei IC's fuer die Signale, eins zum Invertieren der Signale und eins zur Spannungswandlung, +/- Spannung aus nur einer Stromversorgung.

Die V.24 Schnittstelle des Kopplers wurde mit der V.24 Schnittstelle des Rechners verbunden. - Ein Schraubendreher wurde in die Tastatur geklemmt damit immer ein Signal ueber die Rechner-Schnittstelle an den Koppler gesandt wurde.

Die Schalterchen des Kopplers stellte ich beide auf Kanal A, legte beide Lautsprecher aufeinander (akustische Kopplung) und los gings mit vielen wirren Zeichen auf dem Bildschirm. Ein paar vorsichtige Drehungen am Trimmer und einwandfrei erschien der 'auf der Tastatur festgeklemmte' Buchstabe auf dem Schirm. Meine Freude war gross, schnell ein Schluck Mineralwasser (!) fuer den klaren Kopf , die Flasche zurueck auf den Tisch und schon war es passiert, ein Kuddelmuddel auf dem Bildschirm, dann wirder das 'festgeklemmte Zeichen'. Ein vorsichiges leichtes Husten bestaetigte meinen Verdacht: sehr stoeranfaellig auf Umweltgeraeusche, na das wird sich schon nochgeben (beruhigte ich mich).

Umschalten auf Kanal B, schnell noch mal die gleiche Prozedur. Aber nein, es klappte nicht. Ja wieso denn nicht ?! (Verblueffung), noch einmal auf Kanal A, alles ok, und Kanal B ?
Nichts !!, Selbst ein lautes und unwirsches Husten in das Lautsprecher-Mikrofon. Ja hols der Geier ....

Um meinen Frust nicht weiter zu beschreiben: es lag einfach an der mangelnden Empfindlichkeit des Lautsprecher-Mikrofons auf dem zweiten Kanal. Ich habe dann die Lautsprecher gegen Hoer-und Sprechkapsel (von der Post) vertauscht. Das Ergebnis war noch mieser, jetzt ging gar nichts mehr. Das Kohlemikrofon war viel zu traege. Also wurde wieder der Lautsprecher als Lautsprecher und die dynamische Hoerkapsel der Post als Mikrofon eingesetz.

Meine Freude war gross - endlich klappte alles auf beiden Kanaeln, die Zeichen kamen einwandfrei auf dem Bildschirm an.

Schnell wurden die Schallwandler in Kappen von Spraydosen (von Teppich-Schampoo), eingebettet und in viel Schaumstoff verpackt.

Telefon her, ca. 20 Minuten vergebliches Waehlen, dann endich eine freie Mailbox. Ja was kam da fuer ein Kuddelmuddel uber den Bildschirm. Um es kurz zu machen, nach einer Woche frustrierenden Versuchen habe ich den Koppler aufgegeben. Das Ding war einfach zu unempfindlich fuer die Datenuebertragung und zu empfindlich auf Umgebungsgeraesche. Selbst die Neon-Roehren in meinem Arbeitszimmer verursachten Stoerungen, kein Laut durfte zu hoeren sein. Jedes Druecken einer Taste verursachte eine Stoerung.

Abgeschirmte Kabel brachten ebensowenig wie ein Auswechseln und zweites Aufbauen der Schaltung. Nun, ich habe mir einen Akustikkoppler der Firma TANDY gekauft,Kosten ca. 400 DM, Ausgegeben habe ich fuer den Selbstbaukoppler ca. 120 DM, mit V.24 Schnittstelle, Steckernetzteil, alle moeglichen Schallwander und was weis ich) - haette ich mir sofort den TANDY Koppler gekauft, so haette er mich nach der Rechnung nur 280 DM ge[ko]stet, so aber 520 DM,(auch wenn die Rechnung nicht gaz stimmt). Hinzu kommen die vielen Abende (und Naechte) des Prbierens, (die Telefonrechnung habe ich noch nicht bekommen, aber mir graut es schon jetzt).

Wenn man aber ganz sachlich ueberlegt, so sollte einem jedem, der diese Schaltung nachbauen will klar sein, dass ein Bausatz von ca 40 bis 50 DM nicht das gleiche leisten kann wie ein Geraet von ca. 400 DM. Das wird einem auch klar, wenn man mal in den 'Bauch' eines gekauften Kopplers sieht. An jedermann die Warnung also:Diesen Koppler nicht nachbauen.

Die Filtereigenschaften dieses Kopplers sind sehr gering - ausserdem wird ueberhaupt nicht beachtet, dass die Post, um die Sprachverstaendlichkeit zu erhoehen, die Frequenzen anhebt. Das muss der Koppler ebenso ausgleichen koennen wie die unterschiedlichen Empfindlichkeiten des Mikrofons auf verschiedene Frequenzen. Es muesste also fuer jeden Kanal noch ein zusaetzlicher Trimmer fuer die Lautstaerke vorhanden sein usw.

Ich habe allerdings mit dem TANDY Koppler keine Probleme und kann diesen sehr empfehlen - und wenn einer doch Lust hat den Koppler zu bauen: Mit Piezos und Electret-Mikro habe ich es noch nicht probiert, vielleicht fruchtet das ? Und an alle diejenigen,die diesen Koppler zur Zufriedenheit bauen konnten (oder bauen werden),meinen herzlichsten Glueckwunsch. Ich jedenfalls bin kuriert - und wenn einer mit mir ueber Akustikkopler in Kontakt treten will (was mich sehr freuen wuerde), und die Uebertragung ist unter allen Sa..., dann liegt es entweder an einer schlechten Sprechleitung oder an diesem bewussten Koppler der trotz allem seinen Einsatz gefunden hat.

Den Akustikkoppler von Tandy kann ich Euch für 349,00 DM (anstatt knapp 400 DM) anbieten. Das Angebot geht allerdings nur bis 18. März!

Wer dieses Angebot wahrnehmen will, der schickt mir einen Scheck über 349,00 DM wenn ich ihm den Akustikkoppler beim Clubtreffen übergeben kann, andernfalls einen Scheck über 354,00 DM (Porto!).

Günther Wagner
Gartenstraße 4
8201 Neubeuern

```
                 00010 ;*********************************************************
                 00020 ;*  PROGRAMMNAME :  DOSCAL                               *
                 00030 ;*  FILENAME     :  DOSCALL/ASM                          *
                 00040 ;*  DATUM        :  12.05.84                             *
                 00050 ;*                                                       *
                 00060 ;*  ERSTELLT VON : JOSEF KONRAD                          *
                 00070 ;*                 ANZENGRUBERSTRASSE 35                 *
                 00080 ;*                 8038 GROEBENZELL                      *
                 00090 ;*                                                       *
                 00100 ;*  BESCHREIBUNG : ERWEITERUNG FUER TASMON ZUM           *
                 00110 ;*      AUFRUFEN VON NEWDOS80-ROUTINEN NACH EINGABE      *
                 00120 ;*      DES KOMMANDOS 'U'.                               *
                 00130 ;*      FUER DIE ANGEGEBENE ORG-ANWEISUNG MUSS TASMON    *
                 00140 ;*      IM BEREICH A000H-BFFFH ABGESPEICHERT SEIN.       *
                 00150 ;*    . DIE DOS-KOMMANDOS DUERFEN MAXIMAL 32 ZEICHEN     *
                 00160 ;*      LANG SEIN.                                       *
                 00170 ;*                                                       *
                 00180 ;*      LITERATUR 80 MICRO 6,7/82/12                     *
                 00190 ;*                                                       *
                 00200 ;*********************************************************
                 00210 ;
                 00220 ;
9F75             00270           ORG     9F75H
0020             00280 BUFF      DEFS    20H
9F95 44          00290 MESS1     DEFM    'DOS COMMAND ?'
9FA2 57          00300 MESS2     DEFM    'WEITER <ENTER> '
9FB1             00310 MESEND    EQU     $
                 00312 ;
                 00314 ;
9FB1 E5          00320 DOSCAL    PUSH    HL
9FB2 C5          00330           PUSH    BC
9FB3 D5          00340           PUSH    DE
9FB4 CD5CA9      00350           CALL    WL               ;DISPLAY 'U'
9FB7 01000A      00360           LD      BC,0A00H
9FBA CD6000      00370           CALL    60H              ;DELAY
9FBD CDABA9      00380           CALL    CLS
9FC0 21959F      00390           LD      HL,MESS1
9FC3 11403F      00400           LD      DE,3F40H
9FC6 010D00      00410           LD      BC,MESS2-MESS1
9FC9 EDB0        00420           LDIR
9FCB 214F3F      00430           LD      HL,3F40H+MESS2-MESS1+2
9FCE 22B40F      00440           LD      (4020),HL        ;SETZE CURSOR
9FD1 21759F      00450           LD      HL,BUFF
9FD4 0620        00460           LD      B,20H
9FD6 CDD905      00470           CALL    05D9H            ;INLINE
9FD9 CD1944      00480           CALL    4419H            ;DOS COMMAND AUSFUEHREN
9FDC 3807        00490           JR      C,END            ;FALLS KEIN FEHLER
9FDE 2805        00500           JR      Z,END            ;FALLS KEIN FEHLER
9FE0 CBFF        00510           SET     7,A
9FE2 CD0944      00520           CALL    4409H            ;AUSGABE FEHLERMELDUNG
9FE5 21A29F      00530 END       LD      HL,MESS2
9FE8 11C03F      00540           LD      DE,3FC0H
9FEB 010F00      00550           LD      BC,MESEND-MESS2
9FEE EDB0        00560           LDIR
9FF0 21D13F      00570           LD      HL,3FC0H+MESEND-MESS2+2
9FF3 22B40F      00580           LD      (4020),HL        ;SETZE CURSOR
9FF6 CDE3A0      00590 LOOP1     CALL    KYINP            ;TASMON TASTATURABFRAGE
9FF9 FE0D        00600           CP      0DH
9FFB 20F9        00610           JR      NZ,LOOP1
9FFD D1          00620           POP     DE
9FFE C1          00630           POP     BC
9FFF E1          00640           POP     HL
```



```
                 00642 ;
                 00644 ;** HIER BEGINNT TASMON **
                 00646 ;
A95C             00650 WL        EQU     0A95CH
A9AB             00660 CLS       EQU     0A9ABH
A0E3             00670 KYINP     EQU     0A0E3H
                 00680 ;
                 00690 ;         PATCH IN USER ROUTINE SPRUNGADRESSE
                 00700 ;
                 00710 ;
A0DE             00720           ORG     0A0DEH
A0DE B19F        00730           DEFW    DOSCAL
9FB1             00740           END     DOSCAL
00000 TOTAL ERRORS
31334  TEXT AREA BYTES LEFT

BUFF    9F75 00280   00450
CLS     A9AB 00660   00380
DOSCAL  9FB1 00320   00730 00740
END     9FE5 00530   00490 00500
KYINP   A0E3 00670   00590
LOOP1   9FF6 00590   00610
MESEND  9FB1 00310   00550 00570
MESS1   9F95 00290   00390 00410 00430
MESS2   9FA2 00300   00410 00430 00530 00550 00570
WL      A95C 00650   00350
```

Das Programm DOSCAL ermöglicht die Ausführung von NEWDOS80-Kommandos (z.B. DIR <dn>) von TASMON aus durch Eingabe des Kommandos 'U'.

Nach dem Abspeichern auf Diskette wird das Programm von TASMON aus mit L D <ENTER> DOSCALL/CIM zum TASMON-Programm 'gelinkt' und dann das erweiterte Programm mit W D 9F95 BFFF A000 TASMON/CMD abgespeichert (gilt wenn TASMON im Bereich A000-BFFF liegt, andernfalls muß man die entsprechenden Adressen und die ORG-Anweisung ändern bzw. TASMON in den oben genannten Bereich verschieben, 'linken', dann wieder in den gewünschten Bereich verschieben und dann mit den entsprechenden Adreßeingaben abspeichern).

Richard's Regel zum Eigentum:

1) Wenn man Dinge nur lange genug aufbewahrt, kann man sie danach auch sicher wegwerfen.

2) Wirft man hingegen etwas weg, so benötigt man es in dem Augenblick, in dem es nicht mehr greifbar ist!

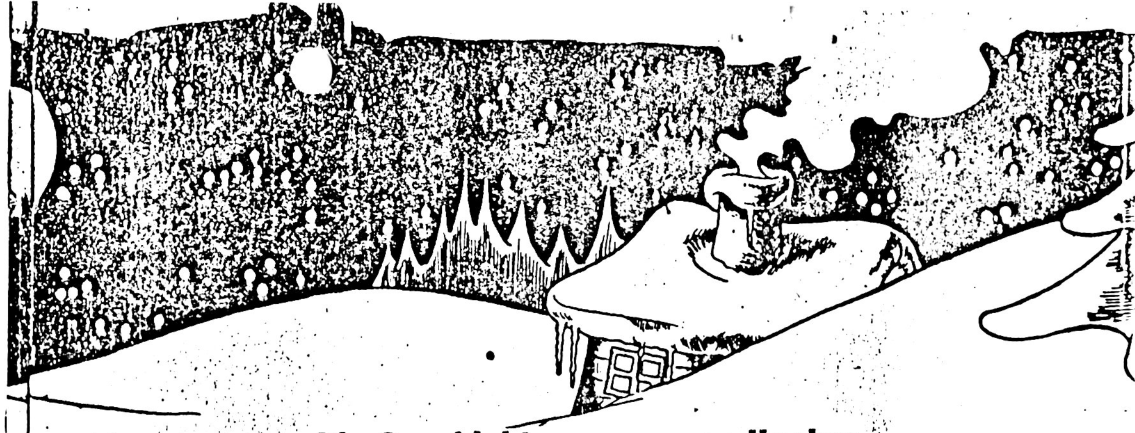
KURZBESCHREIBUNG DES KOMTEK 1 :

Ich möchte als einer der wenigen Komtek-Besitzer auf diesem Planeten der Bitte von Hartmut nachkommen und den Computer kurz beschreiben .

Es handelt sich bei dem KOMTEK 1 um einen mechanisch recht gut gelungenen TRS 8o - Nachbau aus Hongkong , der vor etwa zwei Jahren für ca 1ooo,- DM zu bekommen war .
In der Grundausstattung 16k-RAM (12k-ROM Basic Level II), die leicht intern auf 32k-RAM , einfach durch Einstecken der RAM in vorhandene Sockel zu erweitern ist .
Groß/Kleinschreibung , Umlaute , Tastenentprellung und Autorepeat stehen nach Aufrufen von 'SUPER-BASIC' zur Verfügung . Auch belegt das 'SUPER BASIC' jede Taste mit einem Graphiczeichen , das , wenn es nicht in "" gesetzt wird einem Basic-Befehl entspricht .   (Also : Hochpfeil"1" =▞ / Hochpfeil 1 = CLOSE)
Wenn man erst einmal im Kopf hat , was sich unter der jeweiligen Taste versteckt , dann kann man sehr viel Zeit beim Programmieren sparen .
Der KOMTEK 1 verfügt über einen eingebauten und programmierbaren Tongenerator incl. Lautsprecher , der mit Hilfe eines Maschinenprogramms auch ohne Anschluß eines Cassettenrecorders o.ä. den Computer zum 'Sprechen' bringt.
Echtzeituhr und sechs programmierbare Zeitschalter mit 'Chinch-Ausgängen' sowie vier programmierbare Sensoreingänge fordern die Bastler unter uns heraus . Ohne zusätzliche Hardware können direkt Relais oder Transistoren geschaltet werden , die dann z.B. eine elektrische Eisenbahn , die Zentralheizung oder etwa eine Messdaten erfassung steuern .
Weiterhin verfügt das Gerät über einen 5o Pin-Erweiterungsbus , einen Disk-Driverport (Shugart - nicht betriebsbereit) und ein Druckerinterface .
Die Nachrüstung mit einem Floppy-Controller ist nicht problemlos, da im Gerät kein Platz für einen Controller ist und die Schnittstelle nicht mitTRS/GENIE hardwaremäßig kompatibel ist.
Eine genaue Dokumentation oder gar ein Schaltbild für das Gerät konnte ich bis heute nicht auftreiben . Das hat zur Folge , daß ich weder TRSDOS noch NEWDOS 4o sondern nur NEWDOS 8o zum Laufen bekommen habe (wahrscheinlich wegen Speicherdoppelbelegung) .
Meine Erfahrungen beziehen sich fast ausschließlich auf das recht komfortable Basic und da ich weder den TRS 8o noch das GENIE genau kenne , kann ich keine evtl. Unterschiede aufzeigen .
Kurz aber noch einige Nachteile des Komtek 1 .
Er verfügt nicht über Funktionstasten und die Reset-Taste befindet sich genau über der Return(Enter)-Taste . Dokumentation über Speicherbelegung oder Schaltbilder sind schwer oder garnicht zu bekommen . Der Erweiterungsbus ist nicht an TRS/GENIE- Zubehör anzuschließen .
Das Gerät ist aber mit TRS/GENIE voll softwarekompatibel .
Kaufen kann man den KOMTEK 1 noch beim 'COMPUTER STUDIO BRAUNSCHWEIG) für ca 65o,- DM .
Speziellen Fragen zum Komtek stehe ich gern zur Verfügung .

Gruß

Hans Jürgen Bozek



Die Geschichte vom armen Hacker

Irgendwo hinter dem Dorf Oberwiesenau hausten in einer kleinwinzigen Hütte gar arme Leute, und der Computer, den sie benutzten war alt und langsam, und die Datasette savete gar kärglich, und zuweilen hatten sie keine einzige lauffähige Kopie und waren weit schlimmer noch dran als die Mäuse im Wald.
Nun waren einmal wieder die seligen Weihnachten gekommen, und Mutter und Vater waren ins Kirchdorf hinuntergegangen, um dort in der Christmette den lieben Gott um einen brauchbaren 16-biter zu bitten. Die Kindlein, die sie hatten, es waren ihrer sieben, die blieben zuhause, saßen an ihren Joysticks und meinten, es müsse heut nacht doch noch ein goldbekränzter Space-Invader daher klingeln und ihnen einen Sack voller RAM-Packs, Cartridges und neuer Spielprogramme bringen, auf daß sie nicht gar so arm und hilflos dasäßen und unter Umständen nach einem Buche greifen müßten mitten in der Heiligen Nacht. Aber draußen regte sich nichts, nur die Sterne gingen um das Haus und spiegelten sich im Schnee, und die nahen Tannen standen weiß und ehrwürdig.
Als es Mitternacht worden und es die Kinder schläferte und sie am liebsten geweint hätten vor vergeblichem Warten, sieh, da klopfete es und ein Grunzen ertönte an der Tür. Da taten sie denn stracks und fröhlich die Türe auf. Doch draußen stand kein Space–Invader, auch kein Gremlin mit Rucksack, sondern nur ein armer Hacker. Man hätte die Rippen an ihm zählen können, so verhärmt sah er aus: Man hatte ihm sein Modem beschlagnahmt und dadurch jede Lebensgrundlage genommen. So blickte er müde und verzweifelt drein und hatte gar wenig Freude mehr an der Welt.
Das arme Wesen erbarmte die Kindlein sehr. Sie nahmen den Hacker mit in die Wohnung, holten den allerletzten Kasten Pilsener aus dem Keller, räumten den Kühlschrank aus, soweit Nahrhaftes in ihm zu finden, tischten dem armen Hacker auf und nötigten ihn, vor dem Bildschirm Platz zu nehmen.
Dem Hacker, der vor Zeiten Besseres gewöhnt gewesen, mutete die Hardware der Kleinen gar kümmerlich an doch sah er ihre gute Absicht, und ihre Mildtätigkeit rührte ihn.
Nachdem der Kasten geleeret und die Happen zwischen den gelichteten Beißern des Hackers verschwunden waren, ging ein Leuchten über seine müden Augen. Er streckte die Beine aus und hielt ein erquickliches Nickerchen.
Als nun aber die Kleinen zur Ruhe kamen und merketen, daß sie den armen Eltern das letzte Weihnachtsmahl genommen und verschenkt hatten, da erschraken sie gar sehr und hatten große Angst vor der Rückkunft der Eltern.
Der Mond schien über der Alm hinter dem Haus und der silber glänzende Schnee machte ein gar friedliches Bild.
Als nach einer guten Weile der Hacker ausgeschlafen, da reckte und streckte er sich, zwinkerte den Kindern zu und wandte sich zum Gehn: Er band sich einen dicken Wollschal um, stülpte eine Pudelmütze über und stapfte in den Schnee hinaus.
Und siehe da, als ihm die Kinder nachblickten, wie er die mondbeschienene Alm hinanstieg und ab und zu den Schnee von den Schultern schüttelte, da blinkte und blitzte es bisweilen wie Sternschnuppen. Und als die armen Kleinen hinaus traten und den Spuren des Hackers folgten, da fanden sie lauter kleine Eproms, sieben an der Zahl, im Schnee liegen.
Sie hoben sie auf, trugen sie in die Wohnung, löteten sie in ihre Rechner und sahen, daß jedes Eprom mit der köstlichsten Software beschossen war, welche die Kleinen je kennengelernt hatten.
Und als die Eltern von der Christmette heimkehrten und die geheimnisvollen Vorgänge in dieser Nacht erfuhren, da zürneten sie nicht, sondern freuten sich mit den Kindern.

(Aus einer alten Chronik)

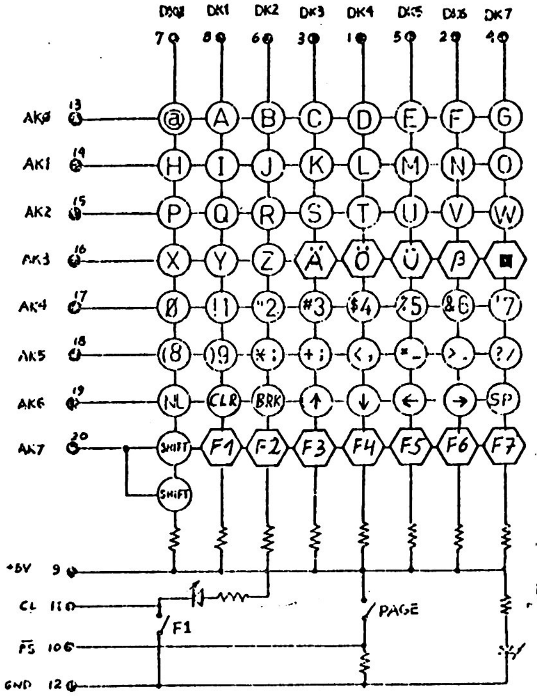
N O C H M A L S :   J O Y S T I C K - A N S C H L U S S

Im Clubinfo Nr. 5 beschreibt Jens den Anschluß eines Joysticks an das Genie bzw. den Tandy. Der Trick, über den Port Ø zu gehen, wobei auf der Adreßleitung eben nichts liegt, ist geradezu hinterlistig. Respekt! Die Nachteile nennt Jens allerdings selber: Der Busstecker ist besetzt (wenn man nicht löten will), der Port Ø ist es auch. Es gibt eine Abhilfe, bei der man allerdings ums Löten nicht herumkommt:

Unsere Tastatur ist ein Aggregat aus lauter simplen Tastern. Sie erzeugen einen Kurzen zwischen je zwei Leitungen der Tastaturmatrix. Nichts spricht dagegen, ihn von einem oder mehreren externen Tastern erzeugen zu lassen, z. B. von denen im Joystick.

Dazu müssen die Leitungen, die für das Leerzeichen und die Pfeiltasten zuständig sind (DK3 - DK7 sowie für alle gemeinsam AK6) von aussen über eine Buchse zugänglich gemacht werden. Man findet sie an den Lötstellen des Flachmannkabels, das die Tastatur mit der CPU-Platine verbindet. So ist das zumindest beim Genie. Die Unterschiede zum TRS-80 dürften kaum mehr als optisch sein.

Das "Who's who" bzw. "Where's what" auf der Kabelflunder habe ich jetzt nicht mehr auswendig drauf. Meinen Computer zerlegen und nachsehen möchte ich nun auch nicht. Die Numerierung der AK- und DK-Leitungen in der Abbildung von 1-20 hilft wohl weiter.



Ein Test gibt letzte Sicherheit: Wenn das Gehäuse geöffnet ist, kann man das Flachbandkabel von der Tastatur abziehen. Vorsicht! Es ist sehr steif, und die Lötstellen könnten brechen. Nun werden mit einem Stück Draht nacheinander je zwei Leitungen verbunden. Bei durchschnittlich jedem zweiten Versuch sieht man etwas auf dem Bildschirm. Beim Vergleich mit der Abbildung ergibt sich, welche zwei Leitungen man erwischt hat.

Schließlich sind alle identifiziert. An die betreffenden Kabelenden kann jetzt ein zusätzliches Kabel angelötet werden, das zu einer Buchse führen soll. Es eignet sich jede Buchse, die wenigstens sechspolig ist. Die Rückseite des Genie ist schon recht voll, aber an den Seiten des Gehäuses findet sich noch genug Platz zum Einbau.

Beim beschriebenen Kabeltest nach der Zufallsmethode sind auf dem Bildschirm sicherlich Umlaute, das ß und das Cursorzeichen erschienen (jedenfalls beim Genie). Demzufolge sind sie per Tastendruck darstellbar; es existieren bloß keine Tasten. Glücklicherweise sind sie aber auf der Tastaturplatine vorgesehen. Über der obersten Tastenreihe sind Löcher vorhanden, in die sich diese Tasten (bei TCS erhältlich) einlöten lassen. Dazu muß allerdings am Gehäuseoberteil ein Stück ausgesägt werden, damit sie Platz haben und von außen erreichbar sind.

Diese Zeichen liegen auf AK3. Dort hat die Tastaturmatrix von DK3 - DK7 eine Lücke. In der Abbildung sind an diesen Stellen die zusätzlichen Tasten mit sechseckigen Symbolen dargestellt. Eine weitere Lücke gibt es in der Shift-Reihe AK7. Nur DKØ ist mit den Shifttasten erreichbar. Die Kreuzungspunkte von AK7 mit DK1 - DK7 bieten sich für Funktionstasten an (in der Abbildung ebenfalls sechseckig): gewöhnliche Klingelknöpfe für'n Groschen, die noch gut unterhalb des eingebauten Recorders (Genie I) Platz haben. Z. Zt. entsteht für H-DOS die erforderliche F-Software.

Ein weiterer Gedanke kommt dem Bastler zwangsläufig, wenn er an der Tastatur herumlötet: Wieso nicht gleich alle Leitungen auf eine Buchse legen? Dann kann eine komplette zweite Tastatur angeschlossen werden. Das Konzept des "remote keyboard" hat seine Vorzüge. Bei meiner Maschine gibt es eine solche Buchse zusätzlich zum Joystick-Anschluß. Auch ein separater Zehner- oder gar Hexblock ist mit entsprechenden parallel liegenden Zusatztasten kein Problem.

Und eine letzte Änderung ist sinnvoll, wenn der Computer eh' schon offen ist: Die Tasten Y und Z können vertauscht werden, um die Tastatur wenigstens ein bißchen einzudeutschen. Dazu werden die Leitungen DK1 und DK2 direkt an diesen beiden Tasten durchtrennt (man findet sie nach der oben beschriebenen Methode). Über eine Drahtbrücke erhält nun die eine Taste den DK-Anschluß der anderen. Zuletzt werden noch die Tastenköpfe mit einem Schraubenzieher vorsichtig abgehebelt und vertauscht.

Die Löterei am eingeschalteten Computer kostet zwar Nerven, ist aber nicht gefährlich, wenn man ein paar Regeln beachtet. Der Lötkolben sollte einen Trenntrafo haben oder bei Berührung der Platine ausgeschaltet sein. Außerdem kann man von Zeit zu Zeit die Heizung oder den Schutzkontakt der Steckdose anfassen (bitte nicht danebengreifen!), um selber nicht statisch aufgeladen zu sein. Offen gestanden habe ich selber dergleichen nie beachtet, und dennoch ging immer alles glatt.

Eine kurze vergleichende Betrachtung von Jens' und meiner Lösung sei noch nachgetragen. Wer den Computer noch "pur" hat, d. h. ohne Peripheriegeräte wie Floppy usw., die auf den Busstecker kommen (und wer sich zutraut, auch Maschinenprogramme auf die Port-I/O-Methode umzustellen), der ist mit Jens' Anschluß bestens bedient. Er spart eine Stunde Arbeit, Nerven und Lötzinn. Wer den Bus aber frei haben, die HRG 1b betreiben, bei der Gelegenheit auch gleich die weiteren Verbesserungen anbringen und seine Software unverändert lassen möchte, dem sei die hier beschriebene Methode empfohlen.

Viel Spaß beim Fummeln - wie auch immer!

Arnulf Sopp

L P R I N T   a l l e s

"Unser" Microsoft-BASIC wird allgemein gelobt. Insgesamt wohl zu Recht. Einiges Ärgerliche steckt aber drin. Dazu gehört das leidige Problem, daß mit der LPRINT-Anweisung nicht alle Codes auf den Drucker ausgegeben werden können. CHR$(0) führt nur zur Abfrage des Druckerstatus (den man in BASIC so nicht einmal erfahren kann), CHR$(10) wird in CHR$(13) umgewandelt (LF wird zu CR), alle senkrechten Tabulationen (VT und FF) werden zu einer Folge von Zeilenvorschüben (LF).

In der Steinzeit, als die Dampfdrucker noch kaum mehr als die Schriftzeichen beherrschten, mag das sinnvoll und einzig richtig gewesen sein. Moderne Drukker erfordern jedoch, daß als Steuerzeichen alles, also auch z. B. die 0 ausgegeben werden kann. Wenn beispielsweise in der ESC-Sequenz zum Ausdruck hochauflösender Graphik die Anzahl der folgenden Dotspalten angegeben werden muß (Gemini, DP 510, ITOH, NEC usw.), ist es unabdingbar, daß restlos alle Zeichen von 0-255 ohne Abwandlung den Interpreter verlassen. Das ist ohne Änderungen des Interpreters nur mit einer Krücke möglich: Das Genie macht es mit OUT 253, code, der Tandy mit POKE 14312,code.

Hierfür sind zwei Segmente des Interpreters verantwortlich. Die Umwandlung von LF in CR geschieht ab 03A6. Dort wird geprüft, ob es sich um CHR$(10) handelt. Falls ja, wird der Code im Akku in 0D (CHR$(13)) verwandelt. Ein paar Bytes zuvor wird FF (CHR$(12)) verwanzt: Der Interpreter springt nach 0386, wo lauter LFs daraus werden. Das geschieht durch den Aufruf einer weiteren Routine, die ich aber hier nicht näher erläutern will, um den Rahmen dieses Beitrags nicht zu sprengen. Im Druckertreiber schließlich, wo LPRINT letztenendes immer landet, wird LPRINT CHR$(0) in die Statusabfrage umgewandelt, die übrigen o. g. Codes werden dort ebenfalls umdefiniert.

Eine wirksame Abhilfe ist sehr simpel. Nahezu alle Bearbeitungsroutinen der BASIC-Befehle landen früher oder später im frei programmierbaren RAM oberhalb 3FFF. In der Regel wird dort ein sog. DOS-Vektor angesprungen, denn je nach angeschlossener Peripherie werden die Befehle unterschiedlich behandelt. Ein einleuchtendes Beispiel wäre der USR-Befehl, der in Level 2 eine andere Syntax hat als unter Disk-BASIC. Der Plattenquirl lädt in diesen unteren RAM-Bereich bestimmte Werte, die die Bearbeitungsadressen der Befehle bei angeschlossener Disco darstellen.

Hier können wir eingreifen. Die im folgenden beschriebenen Manipulationen laufen nur unter Level 2. Da meine Quirle z. Z. in Reparatur sind, kann ich eine Version für Disk-BASIC oder beides nicht äbieten. Mit G-DOS bzw. NEWDOS-80 oder H-DOS müßte es auch klappen, denn diese Betriebssysteme lassen die Adresse im unteren RAM in Frieden, wo das hier vorgestellte Programm eingreift.

In diese Adresse, nämlich 41C1, wird der Sprungbefehl in die eigene Bearbeitungsroutine eingeschrieben: JP F000. Der RET-Befehl, der zuvor dort stand, ist damit überschrieben. Diese eine Routine ist für alle oben bezeichneten Fälle ausreichend, denn bei der Auswertung eines CHR$-Ausdrucks wird immer dorthin verzweigt, bevor der Druckertreiber und die anderen genannten Teile des Interpreters angesprungen werden. Das bedeutet, daß unser Programm ein kleiner Treiber für Codes bis CHR$(13) sein muß. Er funktioniert folgendermaßen:

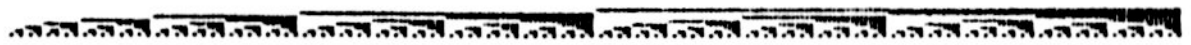
Am Anfang steht natürlich die Abfrage, ob es sich überhaupt um eine Druckerausgabe handelt. Die Bildschirmausgabe bedient sich nämlich derselben ROM-routine. Hierzu werden zunächst mit PUSH AF das zu druckende Zeichen im Akku und die Flags gerettet. Jetzt wird in 409C nachgesehen, ob dort als Druckerflag der Wert 1 abgelegt ist. Wenn ja, muß der Akku nach DEC A auf 0 stehen und die Z-Bedingung erfüllt sein. In diesem Falle erfolgt ein Sprung zur weiteren Bearbeitung, andernfalls der Rücksprung zum BASIC-Interpreter.

Wenn der Drucker angesprochen war (1 in 409C), muß nun geprüft werden, ob das zu druckende Zeichen eines der kritischen <CHR$(14) ist (CP 0EH). Falls nein, erfolgt der sofortige Rücksprung zur weiteren Bearbeitung des LPRINT-Befehls im Interpreter. War es aber einer dieser Codes, wird er ohne weitere Umschweife auf den Drucker ausgegeben (OUT (0DFH),A; für Tandy: LD (37E8H),A).

Da wir uns hier in einer Unterprogrammebene befinden, liegt die Rückkehradresse auf dem Stack. An die alte Adresse wollen wir aber nicht zurückkehren, denn die Druckerausgabe hat ja nun bereits stattgefunden. Also wird sie mit POP BC kurzerhand vom Stack "heruntergeholt" (in Wirklichkeit wird der Stapelzeiger bzw. Stackpointer auf die nächsten 2 Bytes auf dem Stack gerichtet). Es folgt noch einmal POP BC. Das Registerpaar BC ist nämlich im Interpreter vor dem Ansprung unserer Routine auf den Stack gerettet worden. Da wir nicht zur alten Routine zurückkehren, wo der POP stattgefunden hätte, müssen wir ihn eben selber vollziehen. Mit RET geht es schließlich zurück in den Interpreter, wo der nächste BASIC-Befehl bearbeitet werden kann.

Den Effekt dieser kleinen Routine kann man mit einem kleinen BASIC-Einzeiler (hier der Deutlichkeit wegen als Zweizeiler) überprüfen:

```
10 LPRINTCHR$(27)"K"CHR$(0)CHR$(1);
20 FORI=0TO255:LPRINTCHR$(I);:NEXT
```



Direkt darunter steht gleich das Resultat nach RUN. In Zeile 10 wird auf dem Drucker die Einzelpunktgraphik für 256 Punkte initialisiert (wer den Gemini hat, versteht diese Zeile; ich möchte sie jetzt nicht erläutern). Wie man sieht, wird CHR$(0) mit LPRINT ausgegeben, was bisher nicht möglich war. In Zeile 20 folgen nun alle Codes von 0 - 255. Wie man an dem regelmäßigen Muster erkennt, wurden sie alle unverändert ausgedruckt, und zwar auch mit LPRINT.

Einen Haken hat die Routine: Auch der LLIST-Befehl passiert den DOS-Vektor an 409C. Er verträgt sich mit dieser Erweiterung offenbar nicht, wie das Listing des BASIC-Testprogramms zeigt:

```
10 LPRINTCHR$(27)"K"CHR$(0)CHR$(1);20 FORI=0TO255:LPRINTCHR$(I);:NEXT
```

Der Zeilenvorschub vor der BASIC-Zeile 20 ist unterblieben. Ohne Floppies ist mir jedoch die EDTASM-Cassetterei zu albern, deshalb möchte ich darauf verzichten, eine Version anzubieten, die auch dieses Problem noch löst. Vielleicht bietet jemand von euch im nächsten Info ein entsprechendes Programm an!?

Arnulf Sopp

```
F000            00100         ORG    0F000H
F000 F5         00110 CHECK   PUSH   AF          ;Akku und Flags retten
F001 3A9C40     00120         LD     A,(409CH)   ;Ausgabeflag laden
F004 3D         00130         DEC    A           ;Akku -1
F005 2802       00140         JR     Z,LPRINT    ;0, falls Druckerausgabe
F007 F1         00150         POP    AF          ;nicht Dr., Reg. restaur.
F008 C9         00160         RET                ;und normal weitermachen
F009 F1         00170 LPRINT  POP    AF          ;Akku restaurieren
F00A FE0E       00180         CP     0EH         ;einer der krit. Codes?
F00C D0         00190         RET    NC          ;norm. weiter, falls nein
F00D D3DF       00200         OUT    (0DFH),A    ;Code auf Dr. ausgeben
F00F C1         00210         POP    BC          ;Stack korrigieren (CALL)
F010 C1         00220         POP    BC          ;BC restaurieren
F011 C9         00230         RET                ;erledigt
                00240
41C1            00250         ORG    41C1H       ;DOS-Vektor
41C1 C300F0     00260         JP     CHECK       ;dorthin umleiten
                00270
0000            00280         END
00000 Fehler ges.
```

Luidger Röckrath

Der geknackte TRS-80

Über den internen Aufbau von Basic-Interpretern legen die Hersteller verbreiteter Tischcomputer meist den Mantel des Schweigens. So erfordert es schon einige Arbeit, diese Nuß zu knacken. Die Ergebnisse dieser Arbeit (über 500 Adressen und einiges mehr) sind im folgenden dargestellt.

Im Franzis-Sonderheft „Mikrocomputer-Anwendungen" erschienen für mehrere 6502-Computer RAM/ROM-Listings [1], d. h. Aufstellungen aller wichtigen Adressen sowohl für Programmspeicher als auch für Datenspeicher. Dies ist erforderlich, weil die meisten Hersteller sich über diese Dinge ausschweigen, obwohl sie zur optimalen Ausnutzung des Computers für den Anwender aus verschiedenen Gründen notwendig sind [2].
Erstens für den Basic-Programmierer, der Programme für andere Computer adaptieren muß und dazu z. B. die Tastaturadressen benötigt, aber auch für Programmierer, die es entgegen allen Hindernissen nicht sein lassen können, ihren Computer in Maschinensprache zu programmieren, sind besonders die Adressen der Ein-/Ausgaberoutinen von Interesse. Und schließlich für jene Unverbesserlichen, die meinen, ihr Basic sei noch nicht gut genug und könnte durch einige zusätzliche Befehle sinnreich erweitert werden.
Das RAM/ROM-Listing für den TRS-80 enthält für alle drei Gruppen etwas: für den Basic-Programmierer alle wichtigen I/O-Adressen (einschließlich aller Tastaturadressen), für die passionierten Maschinenprogrammierer alle wichtigen Unterprogramme, die sie zum Anpassen fremder oder zum Entwickeln eigener Programme benötigen. Um es dieser Gruppe besonders leicht zu machen, enthält die Tabelle eine genaue Beschreibung, was das Unterprogramm eigentlich tut und welche Register verändert werden. Die letzte Gruppe sei auf den vorletzten Abschnitt dieses Artikels verwiesen, in dem ausführlich die zahlreichen Möglichkeiten des Eingriffs in den Basic-Interpreter erläutert sind.

Einige grundsätzliche Bemerkungen

Wer schon mal in der Tabelle gestöbert hat, der wird sich gewundert haben, daß nur die Beeinflussung der 8080-Register angegeben ist, obwohl der TRS-80 einen Z80 als Zentraleinheit beinhaltet, der doch erheblich mehr Register hat. Des Rätsels Lösung ist ebenso einfach wie enttäuschend: Der TRS-80 arbeitet mit einem 8080-Basic. Man hat sich nur die Mühe gemacht, die Sprünge, wenn möglich, in relative Sprünge umzuwandeln. Lediglich die systemspezifische Ein-/Ausgabe ist im Z80-Code programmiert; dort wird auch einmal das X-Indexregister benutzt, aber zuvor gerettet. So stehen der ganze Zweitregistersatz und die Indexregister für Erweiterungen zur Verfügung.
Das Level-2-Basic des TRS-80 ist ein Microsoft-Basic, d. h. der Programmtext wird nicht so abgespeichert wie er eingegeben wurde, sondern zuvor in einen Zwischencode (1 Byte pro Befehl, zwischen hex 80 und FA) umgewandelt. Diese Zwischencodes sind in [3] auf Seite E/1 aufgelistet. Bei der Programmausführung muß jetzt also nicht mehr das ganze Befehlswort erkannt und dann aus einer Tabelle eine Adresse herausgesucht werden, sondern der Zwischencode dient als Zeiger auf eine Tabelle, die die Ansprungadressen der Befehle und Funktionen enthält. So nimmt die Interpretation des Programmes nicht mehr so viel Zeit in Anspruch wie z. B. beim Level-1-Basic. Nach der gleichen Methode arbeiten die meisten handelsüblichen Interpreter (PET, Apple usw.).

Unterschiedliche Versionen

Das RAM/ROM-Listing wurde von einer Version (V.2) des Level-2-Basic erstellt, die Mitte 1980 verkauft wurde. Bei dieser Version sind der READ/DATA-Fehler [4] und das Prellen der Eingabetastatur ausgemerzt. Zum Vergleich wurde eine ältere Version (V.1) herangezogen, welche die beiden oben genannten Fehler noch hat. Die Unterschiede der V.2 gegenüber der V.1 seien im folgenden dargestellt:

1. Der Tastendruck SHIFT ← wird nicht mehr erkannt aufgrund einer Änderung in der Tastaturdecodierungstabelle. (SHIFT ← hat nun die Funktion einer CONTROL-Taste.)
2. Zugunsten der Entprellung wurden die Texte „RADIO SHACK LEVEL 2 BASIC" und „MEMORY SIZE" zu „R/S L2 BASIC" und „MEM SIZE" gekürzt. Dies hat noch eine Änderung der Zeiger zur Folge.
3. Um beim Laden von der Kassette die Fehlerträchtigkeit zu erniedrigen, wurden zwei Schleifen geringfügig verlängert.
4. Einige kleine Änderungen am PRINT-Befehl, die bewirken, daß
   a) mehrere @ in einer PRINT-Anweisung möglich sind,
   b) nicht mehr zwischen @ und SHIFT @ unterschieden wird,
   c) das Argument von TAB nicht mehr modulo 64, sondern modulo 128 vorbehandelt wird.
5. Der READ/DATA-Fehler wurde behoben.
6. Durch eine Änderung bei CLOAD ist es nicht mehr möglich, eine Kassettenrecorder-Nummer anzugeben (CLOAD#-1, „A" führt zu Fehlermeldung).

Wie von Radio Shack bestätigt, existieren auch andere Versionen, bei denen z. B. nur der READ-/Data-Fehler ausgemerzt ist. Bei der Programmentwicklung sollten diese Unterschiede beachtet werden, um die Kompatibilität zu wahren.

Eingriff in den Interpreter

Der Level-2-Interpreter ist zwar in zwei ROMs abgespeichert, und somit nicht zu ändern, aber an sehr vielen Stellen verzweigt die Programmausführung ins RAM. Dort kann leicht in den Programmablauf eingegriffen werden.

1. Alle wichtigen Ein-/Ausgaberoutinen werden über die DCBs angesprungen, die die Treiberadressen enthalten. Ändert man diese Adressen, so können leicht eigene Ein-/Ausgaberoutinen mit dem Basic-Interpreter in Verbindung treten [5], [6], [7].

2. Wichtige Unterprogramme des Basic werden über die RSTs 8 bis 20 (hex) angesprungen, deren Vektoren im RAM liegen. In praktisch allen Programmsegmenten des Basic-Interpreters werden diese Unterprogramme benötigt. Eine Analyse des entsprechenden Programmteiles zeigt schnell, wie man zweckmäßig eingreifen kann. Dann wird nur noch der Vektor geändert, nach Ansprung des entsprechenden RST geprüft, ob das richtige Programmsegment aufruft und schon kann die eigene Erweiterung folgen [8].

3. Eine einfache und fast unerschöpfliche Möglichkeit des Eingriffes sind die zahlreichen Disk-Basic-Befehle und Funktionen. Diese werden erkannt; sodann wird ins RAM verzweigt, wo Sprünge zu den eigentlichen Programmsegmenten zur Ausführung dieser Befehle und Funktionen stehen. Diese Sprünge zu ändern und eigene neue Befehle zu implentieren, ist kein Problem. Wie [9] zeigt, muß dies noch nicht einmal dazu führen, daß die Zusammenarbeit mit dem Disk-Basic gestört ist.

4. Zur Erweiterung vorhandener Befehle des Level-2-Basic durch das Disk-Basic (z. B. PRINT#, was im Disk-Basic zum Schreiben auf Diskette dient), rufen diese Befehle Unterprogramme im RAM auf. Normalerweise sind diese Adressen mit Return-Befehlen (C9) belegt, aber sie können leicht durch entsprechende Sprünge ersetzt werden. Auch hier analysiert man zweckmäßigerweise das zu erweiternde Programmsegment und wird bald herausfinden, wie man günstig eingreifen kann. Man muß beachten, daß eine Adresse unter Umständen aus verschiedenen Programmteilen aufgerufen wird.
Nun schnell den TRS-80 angeschaltet und an die Arbeit! Wem das hier alles noch zu abstrakt ist, der möge sich einmal [7], [8] und [9] anschauen, drei Beispiele für die ersten drei Methoden. Sollte es jemandem an Ideen ermangeln, auch dem kann geholfen werden. [10] und [11] dürften für die nächsten Feierabende ausreichen.

Das RAM/ROM-Listing

Das RAM/ROM-Listing enthält über 500 ROM-, RAM- und I/O-Adressen. Bei wichtigen Unterprogrammen ist angegeben, welche Register verändert werden (xx). Ist in dieser Spalte ein konkreter Zahlenwert angegeben, so bedeutet dies, daß nach der Rückkehr von diesem Unterprogramm das entsprechende Register immer diesen Wert beinhaltet. Das Register HL dient bei der Ausführung von Basic-Programmen als Zeiger auf den Programmtext. Hat es diese Funktionen, so ist ein P in der entsprechenden Spalte zu sehen.

Literatur

[1] *Martin, R.; Smode, D.*: ROM und RAM in PET und CBM. Franzis-Sonderheft „Mikrocomputer-Anwendungen".

[2] *Feichtinger, H.*: Meine Meinung... FUNKSCHAU 1979, Heft 24, Seite 1418.

[3] Radio Shack: Level II Basic Reference Manual.

[4] Neske, W.: TRS-80 – ein Computer mit kleinem Fehler. Franzis-Sonderheft „Programme für Kleincomputer und Taschenrechner".

[5] *Duddek, G.*: Ein preiswerter Drucker fur den TRS-80. Franzis-Sonderheft „Hobbycomputer 2".

[6] *Hofer, R.*: TRS-80-Treiberprogramm für Schreibmaschinen-Drucker. FUNKSCHAU 1980, Heft 7, Seite 91.

[7] *Röckroth, L.*: Repeatfunktion für den TRS-80 (in Vorbereitung).

[8] *Röckrath, L.*: RESTORE N für den TRS-80 (in Vorbereitung).

[9] *Röckrath, L.*: Double-Precision-Funktionen (in Vorbereitung).

[10] *Feichtinger, H.*: Super-Basic für AIM-65 und PC-100. FUNKSCHAU 1980, Heft 25, Seite 103.

[11] *Klein, R. D.*: S-100-System mit Basic und schnellem Kassetten-Interface. Franzis-Sonderheft „Hobbycomputer 1".

ROM- und RAM-Adressen

ROM-Adressen Level-II-Basic-Interpeter
Initialisierung und Ein-/Ausgabe

					AF	BC	DE	HL
0000	0000	00000	00000	RST 0, Basic-Kaltstart				
0008	0008	00008	00008	RST 08, Basic-UP s. 1C96				
0010	0010	00016	00016	RST 10, Basic-UP s. 1D78				
0018	0018	00024	00024	RST 18, Basic-UP s. 1C90				
0020	0020	00032	00032	RST 20, Basic-UP s. 25D9				
0028	0028	00040	00040	RST 28, wird bei Drücken der Break-Taste angespr.				
002B	002B	00043	00043	INCH1-Ansprung (über DCB), Tastaturabfrage: ASCII-Code gedrückter Taste in A, wenn keine Taste gedrückt ist, A=0	xxxx		4015	
0030	0030	00048	00048	RST 30, unbenutzt				
0033	0033	00051	00051	OUTCH-Ansprung (über DCB), Ausgabe des Akkuinhaltes auf den Bildschirm	xx		401D	
0038	0038	00056	00056	RST 38H, unbenutzt				
003B	003B	00059	00059	PRINT-Ansprung (über DCB), Ausgabe des Akkuinhaltes auf den Drucker	xxxx		4025	
0049	004F	00073	00079	INCH2, wie INCH1, es wird aber gewartet, bis eine Taste gedrückt wird	xxxx		4015	

					AF	BC	DE	HL
0050	005F	00080	00095	Tabelle für die Tastaturdecodierung				
0060	0065	00096	00101	DELAY, Zeitschleife 14,6 µs × BC	0044	0000		
0066	0074	00102	00116	Reset-Ansprung (NMI-Vektor).				
				Wenn Floppy angeschlossen, Basic-Kaltstart, sonst Warmstart				
0075	0104	00117	00260	Basic-Initialisierung:				
0075	008D	00117	00141	DCBs, RST-Vektoren und I/O-Buffer einrichten				
008E	00AB	00142	00171	Disk-Basic-Zeiger initialisieren				
				(werden beim Laden des Disk-Basic geändert)				
00AC	00F8	00172	00248	MEM SIZE anfordern oder selbst Speicherende suchen				
00F9	0104	00249	00260	NEW, „R/S L2 BASIC" ausdrucken und Sprung zur Hauptschleife				
0105	010D	00261	00269	Text „MEM SIZE"				
010E	011B	00270	00283	Text „R/S L2 BASIC(CR)"				
011C	012C	00284	00300	Tastaturentprellung				
012D	0131	00301	00305	L3-Error				
0132	0132	00306	00306	POINT A=00 } Ansprung der drei Grafikbefehle				
0135	0135	00309	00309	SET A=80 } und Setzen des Flags (A).				
0138	0138	00312	00312	RESET A=01 }				
013A	017D	00314	00381	Aus den Koordinaten werden die Adresse im Bildschirmram				
				und die Maske errechnet				
017E	019C	00382	00412	In Abhängigkeit von dem Flag wird ein POINT,				
				SET oder RESET ausgeführt				
019D	01C8	00413	00456	INKE$-Funktion				
01BC	01C8	00444	00456	Leerstring nach X				
01C9	01D2	00457	00466	CLS-Befehl, Bildschirm wird gelöscht	1Fxx			
01D3	01D8	00467	00472	RANDOM-Befehl				
				(als zufällige Größe wird das R-Register benutzt)				
01D9	01F7	00473	00503	Impuls auf Kassette ausgeben	xxxx	00		FC00
01F8	01FD	00504	00509	Kassettenrecorder abschalten	xxxx			
01FE	021D	00510	00541	Kassettenrecordernummer decodieren	xxxx	xxxx	xxxx	P
				und Kassettenrecorder einschalten				
0215	021D	00533	00541	Kassettenrecorder einschalten	xxxx			
021E	0220	00542	00544	Bit 7 von Port (FF) zurücksetzen	xxxx			FC00
0221	022B	00545	00555	(403D) ∧ H ∨ L nach (403D) und zum Port FF	xxxx			
022C	0234	00556	00564	Stern in rechter, oberer Ecke des Bildschirms umschalten	xxxx			
0235	0260	00565	00608	Byte von Kassette lesen (wird im Akku übergeben)	xxxx			
0241	0260	00577	00608	Bit (b) von Kassette lesen (2 . A + b → A)	xxxx			FC00
0261	0283	00609	00643	Byte (im Akku) zweimal auf Kassette aufzeichnen				
0264	0283	00612	00643	Byte (im Akku) auf Kassette aufzeichnen				
0284	0292	00644	00658	Kassette einschalten. 255 mal 0 und A5 aufzeichnen	xxxx	xxxx	xxxx	P
0293	02A8	00659	00680	Kassette einschalten, auf A5 warten und Sternchen	2Axx	xxxx	xxxx	P
				auf Bildschirm setzen				
02A9	0329	00681	00809	SYSTEM-Befehl				
02B2	02B2	00690	00690	Ansprung				
02CE	0313	00718	00787	Objektfile von Kassette laden				
0314	031C	00788	00796	Wort (in HL) von Kassette laden (L, H)	xxxx			xxxx
031D	0329	00797	00809	Ansprung des Objektfile oder irgendeiner anderen Adresse				
032A	0347	00810	00839	Ausgabe des Akkuinhaltes auf Bildschirm (00).	xx			
				Drucker (01) oder Kassette (80) in Abhängigkeit von (409C)				
033A	0347	00826	00839	OUTCH2, wie OUTCH, zusätzlich Cursorposition nach (40A6)	xx			
0348	0357	00840	00855	Position des Cursors in der Bildschirmzeile nach A	xxxx			
0358	0360	00856	00864	INCH3, wie INCH1	xxxx			
0361	0383	00865	00899	INLINE, Eingabe von max. 240 Zeichen in den I/O-Buffer	xxxx		401D	xxxx
				mit allen Cursorfunktionen. Wenn Breack gedrückt wird,				
				Rückkehr mit gesetzten Carry-Flag, bei Drücken von Clear				
				wird der Bildschirm gelöscht und die Eingabe beginnt				
				von vorne. HL enthält die Bufferanfangsadresse –1				
0384	038A	00900	00906	INCH4, wie INCH2	xxxx			
038B	039B	00907	00923	CR auf Drucker ausgeben, wenn Druckkopf nicht	xxxx			
				in Position 0, (409C) auf 0 setzen				
039C	03C1	00924	00961	Zeichen in A auf Drucker ausgeben, Zeichenzähler (409B)				
				incrementieren und bei 0A, 0C und oD auf 0 setzen				
03C2	03E2	00962	00994	Routine zum Aufruf der Ein-/Ausgabe über die DCBs.				
				(DCB-Typ in B, DCB-Adresse in DE)				
03E3	0457	00995	01111	Tastaturabfrage und Decodierung				
				(Ansprung nur über INCH1-4, da nur dann die				
				Register gerettet werden, siehe dort)				
0458	058C	01112	01420	Bildschirmausgabe				
				(Ansprung nur über DCB, da nur dann Register gerettet werden				
				und die Cursoradresse übergeben wird, siehe OUTCH)				
04B8	058C	01208	01420	Bildschirm-Steuerbefehle				

					AF	BC	DE	HL
04B8	04BC	01208	01212	Cursor ON	(0E)			
04BD	04BF	01213	01215	Cursor OFF	(0F)			
04C0	04CD	01216	01229	Cursor HOME (64 cpl)	(1C)			
04CE	04D9	01230	01241	Cursor BACKSPACE	(08)			
04DA	04EB	01242	01259	Cursor →	(18)			
04E7	04EB	01255	01259	Cursor ←	(1A)			
04EC	04F5	01260	01269	Cursor →	(19)			
04F1	04F5	01265	01269	Cursor ←	(1B)			
04F6	0505	01270	01285	32cpl	(17)			
0506	0540	01286	01344	Auswertung der Steuerbefehle				
0541	0563	01345	01379	Zeichen auf Bildschirm setzen und Scroll, wenn nötig				
0564	0572	01380	01394	New-Line	(0A-0D)			
0573	058C	01395	01420	Clear to end of line	(1E)			
057C	058C	01404	01420	Clear to end of frame	(1F)			
058D	05D0	01421	01488	Drucker-Treiber				
05D1	05D8	01489	01496	Drucker bereit?	xxxx			
				j,Z=1.				
05D9	0673	01497	01651	Unterprogramm für INLINE wie INLINE, maximale Anzahl	xxxx	xxxx	401D	
				der Zeichen in B, Bufferanfangsadresse in HL				
				Bei Rückkehr Anzahl der eingegebenen Zeichen in B				
0674	06CB	01652	01739	Wenn Breack gedrückt ist, Sprung zur Basic-				
				Initialisierung, sonst Testen ob Floppy vorhanden				
				Wenn ja, DOS-Laden und starten				
06CC	06CC	01740	01740	Adresse für Rücksprung von SYSTEM-Programmen zum Basic				
				(Basic-Warmstart)				
06D2	0707	01746	01799	RST-Vektoren, DCBs usw. (werden bei Initialisierung				
				ins RAM übertragen (4000–4035)				

Arithmetik

					AF	BC	DE	HL
0708	07F7	01800	02039	X + 1 → X	xxxx	xxxx	xxxx	xxxx
0710	07F7	01808	02039	(HL...) + X → X	xxxx	xxxx	xxxx	xxxx
0713	07F7	01811	02039	BCDE – X → X	xxxx	xxxx	xxxx	xxxx
0716	07F7	01814	02039	BCDE + X → X	xxxx	xxxx	xxxx	xxxx
0778	077C	01912	01916	0 → X	0044			
07A8	07B1	01960	01969	Rundung				
07B2	07B6	01970	01974	OV-Error				
07B7	07C2	01975	01986	Festkommaaddition:	xxxx	xx	xxxx	+2
				(HL...) + CDE → CDE				
07D7	07F7	02007	02039	CDE um A-Bitpositionen	00xx	xxxx	xxxx	xx
				nach rechts schieben				
07F8	07FB	02040	02043	Konstante 1!				
07FC	0808	02044	02056	Konstanten für LOG-Reihe				
0809	0896	02057	02189	LOG(X) → X	xxxx	xxxx	xxxx	xxxx
0841	0896	02057	02189	X . ln2 → X	xxxx	xxxx	xxxx	xxxx
0847	0896	02119	02189	X . BCDE → X	xxxx	xxxx	xxxx	xxxx
0897	0906	02199	02311	X / 10 → X	xxxx	xxxx	xxxx	xxxx
08A2	0906	02211	02310	BCDE / X → X	xxxx	xxxx	xxxx	xxxx
0907	093D	02311	02365	Vorbehandlung der Exponenten und Vorzeichen				
				bei Multiplikation und Division				
093E	0954	02366	02388	X . 10 → X	xxxx	xxxx	xxxx	xxxx
0955	0963	02389	02403	Text X (für Single und Double Presision)	xxxx			
				Wenn X = 0, Z,P				
				Wenn X < 0, C,S				
				Wenn X > 0, keine				
0964	0976	00404	02422	Akku in Fließkommazahl (in X) umformen	xxxx	xxxx	xxxx	xxxx
0977	0989	02423	02441	ABS(X) → X	xxxx	xxxx	xxxx	xxxx
0982	0989	02434	02441	– X → X	xxxx			4123
098A	0993	02442	02451	SGN(X) → X	xxxx			xxxx
0994	09A3	02452	02467	Test X, auch für Integer (siehe oben 0955)	xxxx			xxxx
				bei String in X TM-Error				
09A4	09B0	02468	02480	X → (SP)			xxxx	
09B1	09BE	02481	02494	(HL...) → BCDE → X		xxxx	xxxx	+4
09B4	09BE	02484	02494	BCDE → X			xxxx	
09BF	09CA	02495	02506	X → BCDE		xxxx	xxxx	4125
09C2	09CA	02498	02506	(HL...) → BCDE		xxxx	xxxx	+4
09CB	09DE	02507	02526	X → (HL..)	xxxx	00	4125	+4
09D2	09DE	02514	02526	(HL...HL+(40AF)) → (DE..)	xxxx	00	+4	+4

Fortsetzung folgt

Adventure - Ecke

H a l l o !

Hier ist wieder Euer Adventure-Alex.

Dieses Mal kann ich Euch einen Beitrag von

C H R I S T I A N   S C H R E W E

vorstellen, der aus der Zeitschrift Homecomputer (Mai 1984) das Adventure 'Abenteuer' abgetippt hat. Hier sein Beitrag, damit manchem der Kopf nicht zu rauchen anfängt.

WOZU MAN WAS BRAUCHT !

Man suche:

Den Stock, um zu versuchen, die Schlange zu töten, die flieht, und ein Eis verliert.

Ein Eis, das man dem Ritter gibt, um sein Schwert zu bekommen, das man braucht, um den Grottenolm in die Flucht zu schlagen.

Den Altar, zieht ihn weg, und findet ein Feuerzeug.

Eine Lampe, die man zusammen mit dem Feuerzeug braucht, um einen der beiden Tunnels zu erleuchten.

Eine Münze, die in den See geworfen werden muß, damit dieser das Wort 'AGDAR' preisgibt (siehe Heuschober).

Zwei Bären, sie müssen zum kleinen Raum mitgenommen werden, dort reissen sie Schlüssel los. Ihn braucht man, um die Tür zum Heuschober zu öffnen.

Einen Knochen, um den Hund zu betäuben, der die Verbindung Dorf-Haus bewacht. Den Knochen findet man unter dem Feuerholz im Kamin.

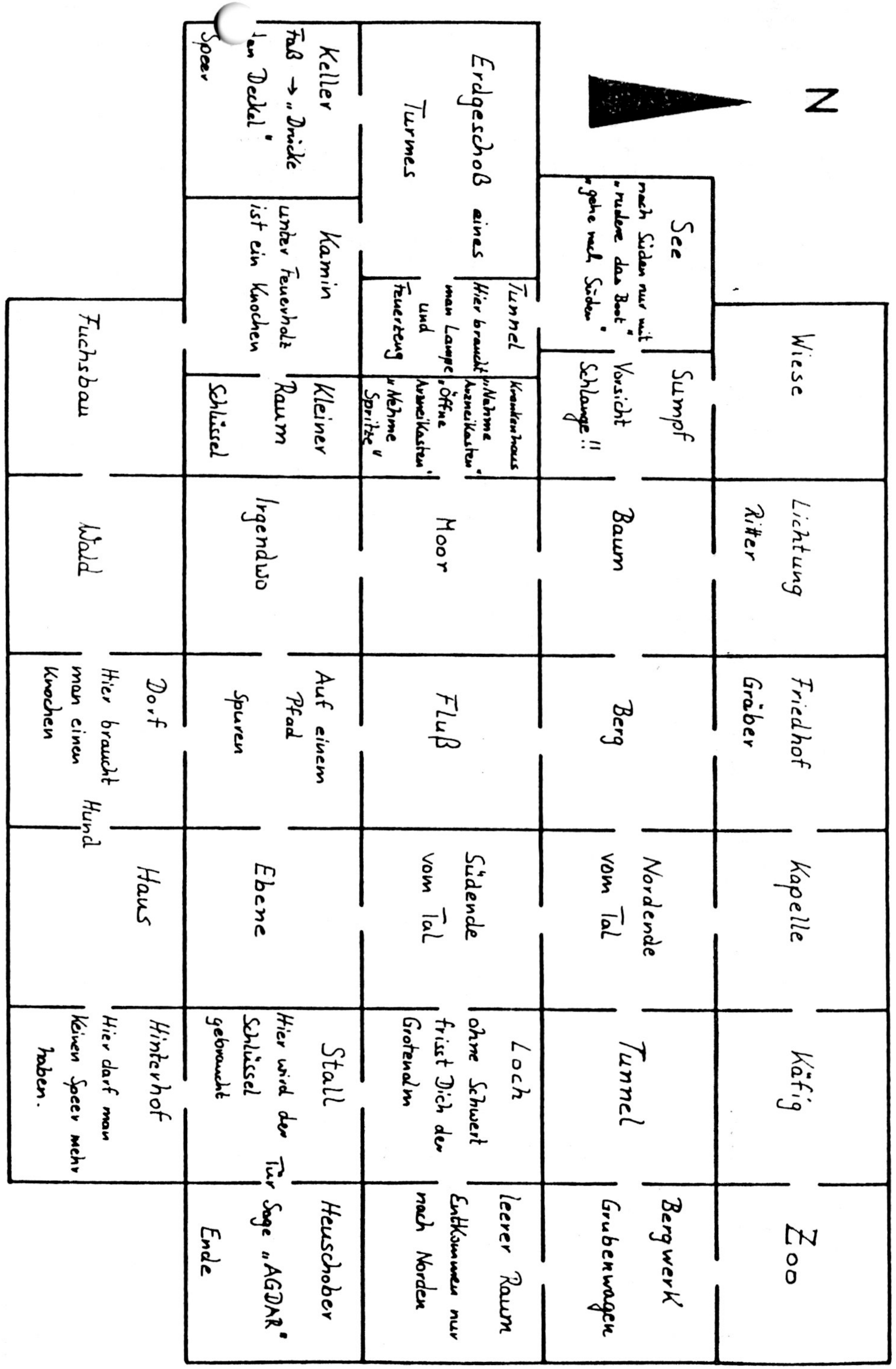
2 TIPS:

1) Bist Du im Krankenhaus, so nehme den Arzneikasten, öffne in und nehme die Spritze. Du bist dann stark genug, es zu verlassen.

2) Falls Du den Speer hast, lege in spätestens im Haus weg, da im Hinterhof der Knecht lauert.

Als Schluß sei gesagt, es gibt keinen direkten Lösungsweg, da die Dinge teilweise willkürlich verteilt werden.

Christian Schrewe



Dietheim Buttkus

Genie-Minitext

Jeder Tischcomputerbesitzer will sein gutes Stück irgendwann auch zur Textverarbeitung einsetzen. Textverarbeitungsprogramme gibt es wie Sand am Meer, aber die guten sind meist sehr teuer, benötigen viel Speicherraum und oft auch eine Floppy-Disk-Station.

Es wurden auch zahlreiche kleine Programme geschrieben, die den Computer wenigstens zu einer elektrischen Speicherschreibmaschine umfunktionieren sollen. In der Praxis zeigt sich leider, daß die meisten unbrauchbar sind. Dafür gibt es vor allem zwei Gründe: Fast alle benutzen Stringvariable, um den Text in den Speicher zu bringen. Schon nach wenigen hundert Anschlägen scheint sich aber der Computer zu verschlucken. Immer wieder verweigert er die Zeichenannahme für eine gewisse Zeit. An flüssiges Schreiben ist nicht zu denken. Ursache ist die „Garbage Collection", ein Aufräumen der Stringvariablen im Speicher. Während dieses „Reinemachens" ist die Tastaturabfrage abgeschaltet. Hat man den Text dann schließlich im Speicher und will ihn auf Kassette abspeichern, kommt meist die zweite unangenehme Überraschung. In vielen Fällen wird der Kassettenrecorder über die PRINT#- und die INPUT#-Funktion bedient. Die Aufzeichnung und Wiedergabe wird dabei immer wieder durch den Vorlauf unterbrochen, was die Laufzeit in unsinniger Weise verlängert. Abhilfe schafft nur ein Maschinenprogramm, das den Text direkt, also ohne den Umweg über Stringvariable, in den Speicher bringt. Leider haben nicht alle Genie-Computer ein Zusatz-ROM mit einer PUNCH-Funktion, womit Maschinenprogramme und Texte auf Band gespeichert werden können. All diese Schwierigkeiten überwindet das kleine Programm Genie-Minitext (*Bild*). Es ist nur neun Zeilen lang, benutzt keinerlei Stringvariable und gestattet selbst auf einer 16-kByte-Maschine die Aufzeichnung von über fünf dichtbeschriebenen DIN-A4-Seiten mit CSAVE und CLOAD. Daneben bestehen außerordentlich weitreichende Editiermöglichkeiten, wie sie sonst nur bei großen Programmen zu finden sind. Es können Buchstaben, Wörter, Sätze oder ganze Abschnitte geändert, gelöscht oder eingefügt werden. Während des Druckens kann sogar der Drucker auf verschiedene Schriften automatisch umgeschaltet werden. Ganze Abschnitte können übersprungen oder an anderer Stelle gedruckt werden und noch vieles mehr. Und das alles mit einem winzigen 9-Zeilen-Programm?

Der eingebaute Editor wird ausgenutzt

Der Trick ist ganz einfach. GENIE-MINITEXT ist nur das Startprogramm für ein Textsystem, das bereits in Ihren Computer eingebaut ist, es muß nur zum Leben erweckt werden! Grundlage ist die LPRINT-Funktion. Jeder beliebige Text läßt sich zum Beispiel so eingeben:

50 LPRINT"Erste Zeile......"
150 LPRINT"Zweite Zeile......"
usw.

Der Text steckt nun in einem „Programm", das mit dem Zeilen- oder dem Bildschirmeditor wie jedes Basic-Programm sehr komfortabel editiert werden kann. Es können Zeichen oder ganze Zeilen beliebig geändert oder ganz gelöscht werden. Zwischen die Zeilennummern kann man weitere Zeilen einfügen, zum Beispiel auch Druckerkommandos, die während des Druckes Schriftart oder Randbreite ändern. Mit GOTO-Befehlen läßt sich der ganze Text umkrempeln und vieles andere mehr. Nun hat natürlich keiner Lust, jede Zeile mit Zeilennummer und LPRINT" zu beginnen. Lassen wir daher den Computer das „Programm" selbst schreiben, wir beschränken uns auf die Eingabe des Textes. Genie-Minitext lädt ein kleines Maschinenprogramm mit READ DATA an das Ende des Speichers. Dieses Programm verwandelt automatisch jede Textzeile in eine Basic-Zeile, die mit einer Zeilennummer und LPRINT" beginnt. Es sind alle Zeichen außer dem Anführungszeichen erlaubt, eine eingebaute Sperre verhindert die Eingabe eines Anführungszeichens. Sogar um die Zeilenlänge brauchen wir uns nicht zu kümmern. Wenn 65 Zeichen pro Zeile (oder eine beliebig vorgegebene andere Anzahl) erreicht sind, schaltet der Computer beim nächsten Leerzeichen automatisch auf die nächste Zeile um. Die minimale Zeilenlänge steht in der Zeile 30 hinter DATA an erster Stelle (hier 65) und kann beliebig geändert werden. Ist die Texteingabe beendet, kommt man mit BREAK wieder zur Basic-Ebene zurück. Wie ein normales Basic-Programm läßt sich unser Text dann mit CSAVE oder CLOAD speichern und wieder laden.

Zwischen zwei Zeilen lassen sich beim Editieren 49 weitere Zeilen schieben. Allerdings, und hier muß nun die Einfachheit bezahlt werden, sind bei der Korrektur eine eventuelle neue Zeilennummer sowie der LPRINT-Befehl „per Hand" einzugeben.

Wird die größte Zeilenlänge durch Zusätze überschritten, wird einfach eine neue Zeile eingefügt. Ein wenig Geschick ist allerdings schon erforderlich. Schließlich läßt sich mit LLIST ein sehr praktischer Korrekturabdruck erstellen.

Einige Besonderheiten

Da keine Stringvariablen vorkommen, kann das Programm ganz einfach mit CLEAR in der ersten Zeile geschützt werden. Das Maschinenprogramm ist frei verschiebbar. Hat der Computer einen größeren Speicher als 16 kByte, so ist lediglich der Variablen A in der ersten Zeile ein anderer Wert zuzuweisen (bei 32 kByte: 48660; bei 48 kByte: 65240).

Das Basic-Startprogramm wird bei der Texteingabe überschrieben. Nach der Erstellung muß es also erst einmal auf Kassette gesichert werden, ehe man es ausprobiert! Vor jeder neuen Texteingabe ist es also neu zu laden, oder aber man springt das Maschinenprogramm, das bis zum Ausschalten im Speicher stehen bleibt, mit SYSTEM / (Startadresse) direkt an. Natürlich läßt sich auch noch eine Zeile mit Befehlen zur Drucker-Voreinstellung in das Startprogramm einfügen. Leider schützt die CLEAR-Vorgabe in Zeile 9 das geladene Maschinenprogramm nicht vor sich selbst. Wenn man den Speicher bis zum Ende vollschreibt, frißt sich das Maschinenprogramm selbst auf. Programmabsturz und vollständiger Textverlust sind die Folge. Das wäre fatal! Daher sorgt eine kleine Testroutine dafür, daß der noch freie Speicherraum ständig überwacht wird. Nähert sich der Text zu sehr dem Speicherende, wird die Eingabe selbsttätig abgebrochen, die Zeile abgeschlossen und READY aufgerufen. Dadurch wird der Text gerettet.

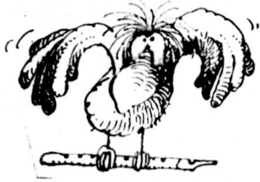

```
? CLEAR?0:A=32?6?:[illegible]:IFA>32767THENB=A-65536
10 FOR?=B TO B+17?:READC:POKEX,C:NEXT
20 DATA 62,144,[illegible],34,[illegible],?4,17,50,0,35,35,115,35,114,35,54,1?5,35,54,34,35,213,6
[illegible] DATA ?5,205,7?,?,2?4,1,40,40,254,34,40,245,205,51,0,254,?,4?,?7,2?1,1?,40,25,11?,
[illegible],124,254,0,4?,[illegible],?1,?5,1?,223,205,73,0,254,32,40,3,4,2?,21?,??,1?,2?,220
40 DATA 51,34,2??,2?3,35,54,0,?5,237,75,64,64,125,2,3,12?,?,241,254,1,40,14,34,[illegible]
,2??,23?,17,50,?,23?,??,235,22?,24,158,54
50 DATA 0,35,54,?,3?,34,249,64,195,204,6,43,24,158,241,?2,1,24,1??
60 H=INT(A/256):L=INT(A-256*H)
70 POKE1?32?,L:POKE16527,H:POKEB+50,H-1
?? CLS:PRINT"TE[illegible]BREAK":D=USR(0)
```

Nur wenige Basic-Zeilen sind erforderlich, um mit dem Video-Genie Textverarbeitung betreiben zu können

Rosige Aussichten

Kossygin fordert Johnson auf: »Laß doch mal eure Computer feststellen, wie ihr Amerikaner in vierzig Jahren leben werdet!«
Sekunden später antwortet das Elektronengehirn: »SOZIALISTISCH«. Kossygin schüttelt sich vor Lachen.
Da verlangt Johnson: »Jetzt prüfe du mal, was dann mit euch Russen sein wird!«
Kossygin studiert die Antwortkarte und meint irritiert: »Das kann ich nicht lesen. Das ist chinesisch.«

Menschliches Versagen



Die Flugpassagiere stolpern die Gangway hinauf, fallen in die weichen Sessel und warten auf den Start des Düsenclippers.
Da ertönt auch schon der Bordlautsprecher; eine freundliche Stimme begrüßt die Passagiere und wünscht einen ruhigen Flug. »Bitte, schnallen Sie sich an und stellen Sie vorübergehend das Rauchen ein«, kommt es aus dem Lautsprecher.
Und dann horchen alle auf: »Dies ist der erste vollautomatische Flug unserer Gesellschaft«, verkündet die Stimme, »Elektronengehirn Eins ersetzt den Chefpiloten, Elektronengehirn Zwei den Co-Piloten. Damit schließen wir eines aus: Menschliches Versagen. Menschliches Versagen. Menschliches Versagen. Menschliches Versagen. . . . .«

String-Anzeige mit V o l l d a m p f

Im letzten Info habe ich angedroht, Maschinenspracheprogramme vorzustellen, falls Interesse dafür besteht. Das ist der Fall, also los!

Als ich in die abgebildete junge Dame noch schrecklich verliebt war, digitalisierte ich sie auf eine Art und Weise, die ich vielleicht in einem anderen Info beschreiben werde und gab ihren traurigen Rest dem Computer ein. Das Resultat seht ihr am Ende dieses Artikels. Die einzelnen Graphik-Codes hatte ich den vorbereiteten Stringvariablen mit einem speziellen Programm einverleibt, das den Anfang des hier gelisteten Programms darstellte (deshalb fängt es erst bei Zeile 520 an). Die Anzeige geschah zunächst mit der kreuzbraven BASIC-Befehlsfolge

FOR I = Ø TO 4 : PRINT A$(I); : NEXT

Es sollte aber gerne schneller gehen. Das war nämlich gerade in der Zeit, als ich die ersten zaghaften Versuche in der Maschinensprache unternahm. Das Ergebnis ist das Listing im Listing ab Zeile 10001. Da es ein paar Merkwürdigkeiten enthält, auf die ich noch eingehen werde, folgt hier zunächst eine vereinfachte Fassung, die man so allerdings nicht in einer Stringvariablen unterbringen könnte. Vergewissert euch dann bei Unklarheiten an diesem Listing:

```
         CALL    2687
         LD      A,5
         LD      DE,1536Ø
LOOP:    LD      BC,2Ø5
         LDIR
         LD      BC,13
         ADD     HL,BC
         DEC     A
         JR      NZ,LOOP
         JP      73
```

Diese Fassung hat außerdem den Vorteil, syntaktisch richtig zu sein, was man von meinem damaligen Machwerk nicht sagen kann. Jeden Versuch, es einem Editor-Assembler so einzugeben, würde er mit einer ganzen Latte von Fehlermeldungen quittieren. Ich habe es aber absichtlich nicht in seiner Gestalt und seiner Programmlogik nachträglich geschönt, weil ich mich nicht geniere, zu zeigen, daß ich einmal ziemlich unbeholfen anfing.

Zunächst aber zum BASIC-Programm. Es beginnt mit der Zuweisung der Stringvariablen, die die Graphik bereits fertig enthalten. Was im Listing wie BASIC-Befehle aussieht, sind die einzelnen Graphikzeichen. Deren Codes werden nämlich vom Interpreter wegen des gesetzten Bits 7 als BASIC-Tokens mißverstanden. Ähnlich ist in Zeile 1ØØØ, wo das Maschinenprogramm zur schnellen Anzeige in der Stringvariablen P$ steht. Das Programm, das theoretisch alle Codes von Ø - 255 enthalten kann, wird bei Werten über 8Øh (128d) ebenfalls als Folge von BASIC-Befehlswörtern angezeigt. U. a. sehen wir dort übrigens den "Befehl" IsA (der gar keiner ist), über den ich bei anderer Gelegenheit schreiben werde.

In den Zeilen 1Ø1Ø - 1Ø3Ø wird schließlich mit Hilfe der VARPTR-Funktion dem Interpreter in der bekannten Weise mitgeteilt, wo er das Maschinensprache-Unterprogramm findet (P) und wo das erste Byte der Graphik steht (X). Mit dem USR-Befehl wird das Programm nun angesprungen.

Es hat keinen Sinn, diesen BASIC-Text eintippen zu wollen. Zwischen Anführungszeichen werden die Befehlswörter nur als Folge sinnloser Buchstaben übernommen, nicht als Tokens.

Und jetzt zum Maschinenprogramm. Daß man das USR-Argument (hier X) als Parameter an das Maschinenprogramm über die ROM-Routine ab ØA7Fh (2687d) übergeben kann, wonach es im Registerpaar HL bereitliegt, steht im BASIC-Manual. Hier wird die Routine mit CALL aufgerufen; das ist so ziemlich dasselbe wie GOSUB in BASIC. Nun wird der Zähler für die fünf Stringvariablen eingerichtet. Der Akkumulator (oder einfach Akku), eines der Register der CPU, wird mit dem Befehl LD A,5 mit dem Wert 5 geladen. In BASIC entspricht das der LET-Anweisung.

Und jetzt wird's ein bißchen komplizierter. Es gibt einen sehr leistungsfähigen Maschinenbefehl, der einen zusammenhängenden Block von Daten (Bytes) an eine andere Stelle transferieren kann: LDIR. Er verlangt drei Vorgaben: Das Registerpaar BC (die Register B und C) enthält die Anzahl der zu übertragenden Bytes. In HL steht die Adresse des ersten Bytes in diesem Block. DE hält die Adresse, an die der Block bewegt werden soll. In unserem Fall ist BC = 2Ø5 (so viele Bytes enthält je eine Stringvariable), HL hält den Wert aus der BASIC-Zeile 1Ø2Ø (erstes Byte von A$(Ø)) und DE ist = 3CØØh (1536Ød) - das ist der Anfang des Bildschirmspeichers, also die linke obere Ecke der Bildröhre. Mit LDIR wird nun der Inhalt von A$(Ø) auf die ersten 2Ø5 Stellen des Bildschirms übertragen.

Der aufmerksame Leser hat bemerkt, daß die obigen Zahlen nicht im Listing erscheinen, und daß die zusätzlichen Befehle INC DE und INC B dort stehen. Das hat eine Bewandtnis, die eigentlich schon einen gesonderten Artikel füllen könnte. Aber ich will es kurz machen: Wenn der Interpreter im Programmtext eine Ø antrifft, versteht er sie als Markierung für das Zeilenende. Die Rede ist wohlgemerkt von ASCII Ø, nicht von der Ziffer "Ø" (= ASCII 3Øh = 48d). Die vier folgenden Bytes versteht er dann als Zeiger auf die übernächste Zeile und die Nummer der folgenden Zeile. Enthält demnach die Zeile 1ØØØ im Maschinenprogramm eine logische Ø, dann wird der Rest vollkommen falsch verstanden.

Deshalb durfte ich DE nicht mit 3CØØh (1536Ød) laden. Ich nahm stattdessen 3BFFh (15359d) und erhöhte mit dem folgenden Befehl INC DE auf 3CØØ. Dann wird der Inhalt von E ebenfalls = Ø, aber diese Null erscheint nicht im Programm. Dasselbe passierte mit dem Zähler BC: Sein MSB, also das Register B, wurde zunächst mit FFh (255d) geladen und anschließend durch den Befehl INC B auf Ø inkrementiert. Denn für die CPU ist 255 + 1 = Ø. So war am Ende der Inhalt von BC = ØØCDh (2Ø5d), die Anzahl der Bytes pro Stringvariabler.

Nachdem nun die Zeiger auf die Quelle (HL) und das Ziel (DE) und der Zähler (BC) die korrekten Werte enthalten, wird mit LDIR das Teilbild in A$(Ø) schlagartig angezeigt. Zwischen diesem Teilbild und der nächsten Portion in A$(1) liegt jedoch noch allerhand BASIC, das übersprungen werden muß: Da sind die Abführungszeichen von A$(Ø) (die im Listing nicht erscheinen), die Null als Markierung für das Zeilenende, der Zeiger auf die übernächste Zeile, die Nummer der folgenden Zeile und der Text 'A$(1)="'. Macht zusammen 13 Bytes, die dem Zeiger HL hinzuaddiert werden müssen. Das geschieht mit ADD HL,BC. Aber da ist das alte Problem: Das Registerpaar BC muß mit ØØØDh (13d) geladen werden. Die Ø als MSB von BC ist aber mitten in einer BASIC-Zeile nicht zulässig, wie oben beschrieben. Da zieht wieder der alte Trick: Durch INC B wird aus FFØDh (65293d) der Wert ØØØDh (13d). Jetzt kann auch A$(1) - zack! - auf den Bildschirm übertragen werden.

Das Programm muß natürlich auch wissen, wann es fertig ist. Gleich beim zweiten Schritt haben wir den Akku als Zähler der Variablen mit dem Wert 5 geladen. In Zeile 1ØØ21 wird er dekrementiert, also um 1 vermindert. Ist er schon auf Ø geschrumpft? Falls ja, dann ist das Zero-Flag im Register F gesetzt. Die CPU setzt dort nämlich bei sehr vielen Befehlen ganz bestimmte Flags (bzw. löscht sie, je nach dem), wenn die ihnen zugordne-

ten Bedingungen erfüllt sind. Bei diesen Flags handelt es sich übrigens um ganz gewöhnliche Bits eines ganz gewöhnlichen Bytes in diesem Register. Der Befehl JR NZ,LOOP (jump relatively on non-zero) wird nur befolgt, wenn das Z-Flag noch nicht gesetzt ist. In diesem Falle springt das Programm zurück an den Beginn der Schleife, der willkürlich den Namen LOOP erhielt. Ich hätte ihn auch SEPPEL taufen können.

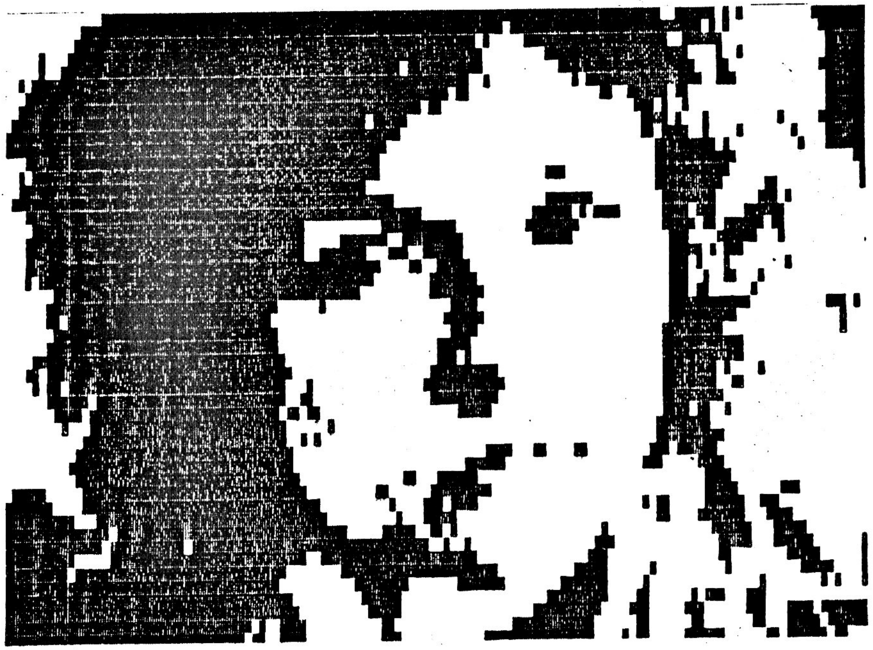
Irgendwann ist der Akku = Ø, die 5 Stringvariablen sind auf den Bildschirm übertragen. Dieser komplizierte Vorgang, über den ihr seit drei Minuten lest, hat einen winzigen Bruchteil einer Sekunde gedauert. Tatsächlich steht die Graphik schlagartig auf dem Bildschirm, nachdem nach RUN die NEW LINE-Taste gedrückt wurde.

Damit sind wir aber noch nicht ganz am Ende. USR stellt einen Unterprogrammaufruf dar, ähnlich GOSUB in BASIC. Doch wo ist RETURN? In Assembler heißt der entsprechende Befehl RET. Nahezu alle einzelnen Abschnitte des Interpreters werden mit einem RET abgeschlossen, denn er ist ein gigantisches Sammelsurium von Unterprogrammen, die von irgendwoher mit CALL aufgerufen werden. Der Befehl JP in Zeile 1ØØ24 entspricht einem GOTO. An der Adresse ØØ49h (73d) befindet sich ein Unterprogramm, das in einer Endlosschleife hängenbleibt, bis irgendeine Taste gedrückt wird. Kommt endlich der erlösende Tastendruck, weil der User Brigitte nicht mehr sehen mag, dann wird der Befehl RET erreicht und READY angezeigt.

Bevor ihr erlöst seid, noch ein Wort zu den komischen Dezimalzahlen links: Die verschiedenen Assembler-Befehle setzen sich aus einem oder mehreren Bytes zusammen. Diese Zahlen enthalten deren Werte. Sie könnten in dieser Form beispielsweise in DATA-Zeilen gepackt werden.

Dieses war sozusagen mein Jungfernprogramm, die erste Maschinenroutine, die ich je schrieb (auch wenn es vielleicht die 5. Version ist). Mit diesem Problem, eine Graphik anzuzeigen, fing alles an. Ich wünsche den Anfängern unter euch viele Erfolgserlebnisse. Ihr habt meine ganze Sympathie.

Arnulf Sopp




```
520 A$(0)="USINGUSINGUSINGUSINGUSINGUSINGNEXTSET
       OUTOUTCLEARDEF             FIELDRESTORE CLSPUTTOUSINGUSINGUSINGU
SINGUSINGLIST   USINGUSINGLPRINTCMDIFFOR                           CLS
   DEFLPRINTUSINGUSINGUSINGTOLISTRESTORERESTORE        R
530 A$(1)="                     OUTTAB(USINGUSINGUSINGUSINGUSINGUSINGUSING
USINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGTOTRON  SETRESETNE
XTSAVEUSINGUSINGUSINGTONEWUSINGDEFDEF USINGRESUMERESET
          DATARUNLPRINTUSINGUSINGUSINGUSINGUSINGUSINGUS
540 A$(2)=" RESETNEWTAB(DELETECLOADCLS RESETLPRINTUSINGUSINGUSINGUSI
NGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINELSE LISTSAVELSERSETUS
INGLPRINTUSINGNEWUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGFORRES
TORE               OUTTAB(TAB(TAB(CLEARTAB(TAB(USINGUS
550 A$(3)="USINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSING    SAV
ECONTUSINGFNUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGUSINGT
OCLEARERROR               NAMEERRORLINEAUTOUSINGUSINGUSINGUSINGUSINGU
SINGUSINGUSINGUSINGTAB(TAB(FNUSINGUSINGUSINGUSINGUSINGU
560 A$(4)="  RESETUSINGREMINPUTRSETUSINGUSINGUSINGUSINGUSINGUSING      DE
FDEFTRON     CLS        DEF  SETSETFOR  SETLETINPUTNEXTLPRINTUSINGUS
INGUSINGUSINGUSINGUSINGUSINGRESUMEFORNEWUSINGUSINGUSINGUSINGUSINGUSI
NGTO DEFCSAVEUSINGDEF RESETLPRINTUSINGUSINGTROFF      RE
1000 P$="
>IsA;
+IsAMKS$DEF
IsA    = FIXERRI
1005 REM
        ***  DIE ZEILEN 520 - 1000 DUERFEN NICHT  ***
        ***  EDITIERT WERDEN. SIE ENTHALTEN IN    ***
        ***  A$(0-4) DIE TEILBILDER ZUR ERZEUGUNG ***
        ***  DER GRAPHIK, IN P$ DAS MASCH.-PRG.   ***

1010 P=PEEK(VARPTR(P$)+1)+256*PEEK(VARPTR(P$)+2):PM=P/256:POKE16527,
INT(PM):POKE16526,(PM-INT(PM))*256
1020 X=PEEK(VARPTR(A$(0))+1)+256*PEEK(VARPTR(A$(0))+2)
1030 U=USR(X)
10000 END'
```

```
        *** ASSEMBLER-LISTING IN DEZIMAL ***

        *** DAS PROGRAMM (P$ IN BASIC)    ***
        *** IST FREI RELOZIERBAR (KEINE   ***
        *** ABSOLUTEN ADRESSEN).          ***


10001 205,127,10           CALL 2687      ;$-ANFANG -> HL
10004 62,5                 LD A,5         ;5 TEILBILDER
10006 17,255,59            LD DE,15359    ;ANF. VIDEO-RAM
10009 13                   INC DE         ;KEINE 0 IN P$
10010 1,205,255      LOOP: LD BC,65485      ;LEN(A$(I))=205
10013 4                    INC B          ;KEINE 0 IN P$
10014 237,176              LDIR           ;BLOCKTRANSFER
10016 1,13,255             LD BC,65293      ;HL = HL + 13
10019 4                    INC B          ;KEINE 0 IN P$
10020 9                    ADD HL,BC      ;FORTS. HL + 13
10021 61                   DEC A          ;NOCH EIN $ ?
10022 32,242               JR NZ,LOOP     ;JA, WIEDERH.
10024 195,73,0             JP73           ;ENDE BEI BE-
                                          ;LIEBIGER TASTE
```

Arnulf Sopp

BÜCHER

Das neueste Buch in der CLUB 80 - Bibliothek:

- Heiko Requardt: Basic: Alles über PEEK und POKE
  Eine Software-Sammlung in Basic
  erschienen im Franzis Verlag

Der Patrick Perschbach hat folgende Bücher zu verleihen:

- 57 BASIC-Programme (dt.)
- Advanced Level II BASIC
- Computer Programming in BASIC for everyone
- Announcing Computer Games for Business, School and Home
- Programming is Style
- Business Programms Applications

Der Günther Wagner hat folgendes Buch zu verleihen:

- TRS-80 und VG ROM-Listing für Level II (Luidger Röckrath)

So, jetzt erst mal wieder eine aktuelle Aufstellung meiner Programmanleitungen und Handbücher.

Bedienungshandbuch Disketten-Betriebssystem
Bedienungshandbuch G-Dos
Bugout-Maschinensprache-Monitor
Visicalc
Datapro
ZBasic 2.2
Scripsit Textverarbeitung
Tiny-Pascal Supersoft
GEAP Grafik-Editor
Musimp•
Air-Traffic-Controller

Jean Claude Wies
(und Andreas Marx)

Dann noch einiges für die Peek und Poke Ecke

Poke 16448	Echtzeit-Uhr Zähler
Poke 16449-51	stellt Sekunden, Minuten und Stunden der Uhr kann mit Peek auch zur Abfrage verwendet werden
Poke 16452-54	dasselbe mit Jahr, Tag und Monat
Poke 16526-27	Startadresse für System-Programme USR(n) über Peek auch brauchbar zur Ermittlung des DEFUSR-Befehls zu gebrauchen. Anm: Einfach Peek(16527)*256+Peek(16526), dann den Wert in Hex-Umrechnen und als DEFUSR=&Hxxxx angeben. Dieses Problem tritt öfter bei Tape-Programmen auf, die keinen DEFUSR Befehl enthalten.

BASIC FASTER & BETTER
=====================================

Wieder darf ich im Vorwort einige Worte über dieses Thema verlieren. Zunächst ist anscheinend manchen Mitgliedern noch nicht klar, wer diese Übersetzung macht. Nun - derjenige heißt Günther Wagner.

Da auch dieses Mal die Resonanz wieder positiv war, erscheinen auch wieder 12 Seiten. Damit ist das Kapitel 4 abgeschlossen.

Zusätzlich sind noch 2 Seiten des Anhangs abgedruckt, die für den Einen oder Anderen wichtig sind.

A C H T U N G *** W I C H T I G :

In der letzten Ausgabe haben sich Fehler bzw. Ungenauigkeiten eingeschlichen:

Seite 21, letzter Absatz:
... mit der Adresse aus dem HL-Register in die Adresse aus dem DE-Register. Dann werden das HL- und DE-Register erhöht, das BC-Register wird erniedrigt......

Seite 21, 3. Absatz:
anstatt 'DATA-Blöcke' werden 'Daten-Blöcke' verschoben

Zum VW-Verwalter, Seite 26:
Klar dürfte sein, daß nach 'START' eine Routine folgen muß.

Seite 16, letzter Absatz:
Statt 'veränderliche' sollte es besser 'verschiebliche' heißen.

Ich bitte um Entschuldigung für diese Fehler. Wer weitere Fehler findet, soll mir diese bitte mitteilen. Viel Spaß mit den neuen 12 Seiten - ich hoffe, Sie bringen Euch was!

Günther Wagner

Scotts erstes Gesetz:

Egal was schiefgeht - man sieht es ihm zuerst nicht an.

Scotts zweites Gesetz:

Wenn man einen Fehler gefunden und endlich korrigiert hat, stellt sich heraus, daß die erste Version richtig war.

Folgerung:

Nachdem sich die Korrektur plötzlich als falsch herausstellte, ist es unmöglich, den Originalzustand wiederherzustellen.

USR Routine Pointer Addresses

DOS VERSION	0	1	2	3	4	5	6	7	8	9
TRSDOS 2.3 Radio Shack Model I	23415 5B77	23417 5B79	23419 5B7B	23421 5B7D	23423 5B7F	23425 5B81	23427 5B83	23429 5B85	23431 5B87	23433 5B89
TRSDOS 2.0 Radio Shack Model 2	11050 2B2A	11052 2B2C	11054 2B2E	11056 2B30	11058 2B32	11060 2B34	11062 2B36	11064 2B38	11066 2B3A	11068 2B3C
TRSDOS 1.2 Radio Shack Model III	22586 583A	22588 583C	22590 583E	22592 5840	22594 5842	22596 5844	22598 5846	22600 5848	22602 584A	22604 584C
TRSDOS 1.3 Radio Shack Model III	22632 5868	22634 586A	22636 586C	22638 586E	22640 5870	22642 5872	22644 5874	22646 5876	22648 5878	22650 587A
NEWDOS 2.1 Apparat	23316 5B14	23318 5B16	23320 5B18	23322 5B1A	23324 5B1C	23326 5B1E	23328 5B20	23330 5B22	23332 5B24	23334 5B26
NEWDOS80 1.0 Apparat	22330 573A	22332 573C	22334 573E	22336 5740	22338 5742	22340 5744	22342 5746	22344 5748	22346 574A	22348 574C
DOS PLUS 3.3D Micro Systems Software	23483 5BBB	23485 5BBD	23487 5BBF	23489 5BC1	23491 5BC3	23493 5BC5	23495 5BC7	23497 5BC9	23499 5BCB	23501 5BCD
LDOS 5.0.1 Lobo Drives, Int'l	23415 5B77	23417 5B79	23419 5B7B	23421 5B7D	23423 5B7F	23425 5B81	23427 5B83	23429 5B85	23431 5B87	23433 5B89
ULTRADOS 4.2 Level IV Products	20992 5200	20994 5202	20996 5204	20998 5206	21000 5208	21002 520A	21004 520C	21006 520E	21008 5210	21010 5212
DBLDOS 4.23 Percom	23316 5B14	23318 5B16	23320 5B18	23322 5B1A	23324 5B1C	23326 5B1E	23328 5B20	23330 5B22	23332 5B24	23334 5B26

Disk Buffer Memory Locations

DISK OPERATING SYSTEM	VERSION	MODEL	1	2	3	4	5	6
TRSDOS Radio Shack	2.3	1	26335 66DF	26625 6801	26915 6923	27205 6A45	27495 6B67	27785 6C89
TRSDOS Radio Shack	2.0	2	27779 6C83	28613 6FC5	29447 7307	30281 7649	31115 798B	31949 7CCD
TRSDOS Radio Shack	1.2	3	25812 64D4	26172 663C	26532 67A4	26892 690C	27252 6A74	27612 6BDC
TRSDOS Radio Shack	1.3	3	26232 6678	26592 67E0	26952 6948	27312 6AB0	27672 6C18	28032 6D80
NEWDOS Apparat	2.1	1	25973 6575	26263 6697	26553 67B9	26843 68DB	27133 69FD	27423 6B1F
NEWDOS/80 Apparat	1.0	1	26347 66EB	26648 6818	26949 6945	27250 6A72	27551 6B9F	27852 6CCC
DOS PLUS Micro Systems Software	3.3D	1	28053 6D95	28599 6FB7	29145 71D9	29691 73FB	30237 761D	30783 783F
DOS PLUS - TBASIC Micro Systems Software	3.3D	1	25450 636A	25996 658C	26542 67AE	27088 69D0	27634 6BF2	28180 6E14
DOS PLUS Micro Systems Software	3.3	3	28039 6D87	28585 6FA9	29131 71CB	29677 73ED	30223 760F	30769 7831
LDOS Lobo Drives, Int'l	5.0.1	1	27237 6A65	27783 6C87	28329 6EA9	28875 70CB	29421 72ED	29967 750F
ULTRADOS Level IV Products	4.2	1	25531 63BB	25821 64DD	26111 65FF	26401 6721	26691 6843	26981 6965
DBLDOS Percom	4.23	1	25973 6575	26263 6697	26553 67B9	26843 68DB	27133 69FD	27423 6B1F

Beachten Sie, daß nur der gegenwärtige Inhalt jeder Variablen gezeigt wird. Der String XY$, welcher die Zahl 100 im MKI$-Format (2 Bytes) enthält, wird für uns automatisch umgewandelt. Enthalten Strings nichtdarstellbare Character (ASCII-Werte kleiner 32 oder größer 191), so werden hierfür Blanks ausgegeben. Anführungszeichen am Beginn und Ende der Strings werden angegeben, damit man Blanks am Anfang und Ende eines Strings auch sieht. Da die Variable G% in Zeile 20 durch die Programm-Logik nicht definiert wurde, wird Sie auch vom AVA nicht angezeigt.

```
65000 PRINT "AKTIVE EINFACHE VARIABLEN:"
65002 ZD%=0: ZZ%=0: ZZ$="": ZZ$(3)="": ZZ%(0)=PEEK(16633):
      ZZ%(1)=PEEK(16634)
65004 GOSUB 65110
65006 IF ZZ%(0)=PEEK(16635) AND ZZ%(1)=PEEK(16636) THEN 65030
      ELSE GOSUB 65130
65007 GOSUB 65140: GOTO 65006
65030 PRINT "AKTIVE FELDER:"
65032 ZZ%(0)=PEEK(16635): ZZ%(1)=PEEK(16636)
65034 GOSUB 65110
65036 IF ZZ%(0)=PEEK(16637) AND ZZ%(1)=PEEK(16638) THEN RETURN
      ELSE GOSUB 65130: GOSUB 65100: GOSUB 65100: GOSUB 65100:
      GOSUB 65110: ZD%=ZZ%(3): ZZ%=0
65038 IF ZZ%=ZD% THEN 65040 ELSE GOSUB 65100: GOSUB 65110:
      ZZ$(1)=ZZ$(0): GOSUB 65100: GOSUB 65110: ZZ%(8+ZZ%)=0:
      ZZ%(5+ZZ%)=CVI(ZZ$(1)+ZZ$(0)): ZZ%=ZZ%+1: GOTO 65038
65040 ZZ$=LEFT$(ZZ$,2): ZZ$(3)="(": FOR ZZ%=ZD% TO 1 STEP -1:
      ZZ$(3)=ZZ$(3)+STR$(ZZ%(7+ZZ%))
65041 IF ZZ%>1 THEN ZZ$(3)=ZZ$(3)+"," ELSE ZZ$(3)=ZZ$(3)+")"
65042 NEXT
65050 GOSUB 65140
65051 ZZ%(7+ZD%)=ZZ%(7+ZD%)+1: IF ZZ%(7+ZD%)<ZZ%(4+ZD%) THEN
      65040
65052 IF ZD%=1 THEN 65070 ELSE ZZ%(7+ZD%)=0
65053 ZZ%(6+ZD%)=ZZ%(6+ZD%)+1: IF ZZ%(6+ZD%)<ZZ%(3+ZD%) THEN
      65040
65054 IF ZD%=2 THEN 65070 ELSE ZZ%(6+ZD%)=0
65055 ZZ%(5+ZD%)=ZZ%(5+ZD%)+1: IF ZZ%(5+ZD%)<ZZ%(2+ZD%) THEN
      65040 ELSE 65070
65060 GOTO 65040
65070 GOSUB 65100: GOSUB 65110: GOTO 65036
65100 ZZ%(0)=ZZ%(0)+1: IF ZZ%(0)=256 THEN ZZ%(0)=0:
      ZZ%(1)=ZZ%(1)+1
65101 RETURN
65110 ZZ%(4)=CVI(CHR$(ZZ%(0))+CHR$(ZZ%(1))): ZZ%(3)=PEEK(ZZ%(
      4)): ZZ$(0)=CHR$(ZZ%(3)): RETURN
65120 FOR ZZ%=1 TO ZZ%(2): GOSUB 65100: GOSUB 65110: ZZ$(1)=
      ZZ$(1)+ZZ$(0): NEXT: IF ZZ$(3)="" THEN GOSUB 65100: GOSUB
      65110
65121 IF INSTR("ZZ$ZZ%ZD%",ZZ$) THEN ZZ$=""
65122 RETURN
65130 ZZ%(2)=ZZ%(3): GOSUB 65100: GOSUB 65110: ZZ$=ZZ$(0):
      GOSUB 65100: GOSUB 65110: ZZ$=ZZ$(0)+ZZ$: RETURN
```



```
65140 ZZ$(1)="": ON (INSTR(" 2 3 4 8",STR$(ZZ%(2)))-1)/2+1
      GOSUB 65144, 65146, 65160, 65162
65142 RETURN
65144 ZZ$=ZZ$+"%": GOSUB 65120: IF ZZ$="" THEN RETURN ELSE
      PRINT ZZ$; ZZ$(3) TAB(20) CVI(ZZ$(1)): RETURN
65146 ZZ$=ZZ$+"$": GOSUB 65120: IF ZZ$="" THEN RETURN ELSE
      PRINT ZZ$; ZZ$(3) TAB(20)
65148 PRINT CHR$(34);: ZZ$(2)=CHR$(ZZ%(0))+CHR$(ZZ%(1))+CHR$(
      ZZ%(2)): ZZ%=ASC(ZZ$(1)): ZZ%(0)=ASC(MID$(ZZ$(1),2)):
      ZZ%(1)=ASC(MID$(ZZ$(1),3)): ZZ$(1)="": ZZ%(2)=ZZ%
65150 IF ZZ%>0 THEN 65156 ELSE PRINT CHR$(34): ZZ%(0)=ASC(ZZ$(
      2)): ZZ%(1)=ASC(MID$(ZZ$(2),2))
65152 IF ZZ%(2)=2 THEN PRINT "CVI("; ZZ$; ZZ$(3); ")"; TAB(20)
      CVI(ZZ$(1)) ELSE IF ZZ%(2)=4 THEN PRINT "CVS("; ZZ$; ZZ$(
      3); ")"; TAB(20) CVS(ZZ$(1)) ELSE IF ZZ%(2)=8 THEN PRINT
      "CVD("; ZZ$; ZZ$(3); ")"; TAB(20) CVD(ZZ$(1))
65154 ZZ%(2)=ASC(MID$(ZZ$(2),3)): GOSUB 65110: RETURN
65156 GOSUB 65110: GOSUB 65100: ZZ$(1)=ZZ$(1)+ZZ$(0): IF ZZ%(3)
      <32 OR ZZ%(3)>191 THEN PRINT "."; ELSE PRINT ZZ$(0);
65158 ZZ%=ZZ%-1: GOTO 65150
65160 ZZ$=ZZ$+"!": GOSUB 65120: IF ZZ$="" THEN RETURN ELSE
      PRINT ZZ$; ZZ$(3) TAB(20) CVS(ZZ$(1)): RETURN
65162 ZZ$=ZZ$+"#": GOSUB 65120: IF ZZ$="" THEN RETURN ELSE
      PRINT ZZ$; ZZ$(3) TAB(20) CVD(ZZ$(1)): RETURN
```

ANMERKUNGEN ZUM AVA

1. Durch Zusammenfassung von Zeilen und ausschließlicher Verwendung von Variablen beginnend mit 'ZZ' oder 'ZD', hat die Lesbarkeit dieser Routine gelitten. Speicherersparnis und eine kurze Variablenliste sind das Ergebnis. Außerdem wird der Übergang zu anderen Variablennamen erleichtert. Für ev. Änderungen ist hier eine Liste der verwendeten Variablen:

ZZ%	Vorübergehender Zähler und Arbeitsspeicher.
ZZ% (0)	LSB der laufenden Adresse.
ZZ% (1)	MSB der laufenden Adresse.
ZZ% (2)	Schreibt Kennzahl 2,3,4 o. 8 für %,$,! o. #-Variablen. Auch vorübergehender Speicher für Stringlänge.
ZZ% (3)	Inhalt der laufenden Speicheradresse (0-255).
ZZ% (4)	Laufende Speicheradresse im Variablen-Speicherber.
ZZ% (5)	1. Dimension des laufenden Feldes.
ZZ% (6)	2. Dimension des laufenden Feldes falls vorhanden.
ZZ% (7)	3. Dimension des laufenden Feldes falls vorhanden.
ZZ% (8)	Zähler für die 1. Dimension.
ZZ% (9)	Zähler für die 2. Dimension.
ZZ%(10)	Zähler für die 3. Dimension.
ZZ$	Laufender Variablen-Name.
ZZ$ (0)	Inhalt der laufenden Speicheradresse (CHR$-Format)
ZZ$ (1)	Laufender Inhalt der Variablen- bzw String-Pointer
ZZ$ (2)	vorübergehender Speicher der laufenden Adresse während dem String-Aufbau.
ZZ$ (3)	vorüberg. Variablen-Ausdruck für Anzeige mit Felder.
ZD%	Dimension des laufenden Feldes (1-, 2- o. 3-fach).

2. Sie können ein 'GOSUB 65030' ausführen falls Sie nur Felder analysieren wollen. Sie können in Zeile 65030 ein 'RETURN' schreiben, falls Sie nur einfache Variablen analysieren wollen. Die Zeilen 65030 bis 65070 werden nicht benötigt, falls Sie nur einfache Variablen anzeigen wollen.

3. In der Zeile 65002 laden wir die Startadresse für einfache Variablen in den Speicher. Der Zeiger steht in der Speicherstelle 16633 und 16634. Wir sind mit den einfachen Variablen fertig, wenn wir die Adressen aus den Speicherstellen 16635 und 16636 erreicht haben (Zeile 65006). Hier beginnt nun der Speicherbereich für die Felder. Beachten Sie, daß wir die Startadresse in der Zeile 65032 nochmals laden. Somit wird erst ein direktes 'GOSUB 65030' (siehe Punkt 2) möglich. Wenn wir die Adressen der Speicherstellen 16637 und 16638 erreicht haben, sind wir mit den Feldern fertig.

4. Im Unterprogramm 65100 wird unsere Adresse erhöht. Dieser Teil ist bei vielen Anwendungen sinnvoll, wo es nötig ist, den Speicher Byte für Byte zu lesen. Wir addieren 1 zum LSB der Adresse. Wenn LSB 256 erreicht, wird es auf 0 zurückgesetzt und MSB wird um 1 erhöht.

5. Das Unterprogramm 65110 wandelt LSB und MSB in eine Adresse mit Integer-Format zurück. Dann wird mit PEEK der Wert der laufenden Adresse eingelesen und umgewandelt, so daß schließlich CHR$ des PEEK-Wertes abgespeichert wird, Dieses Unterprogramm dient der Programmier-Bequemlichkeit und der Leistungsfähigkeit des Speichers; die Ausführungsgeschwindigkeit wird dafür langsamer.

6. Das Unterprogramm 65120 erstellt einen String, welcher die laufenden Variablen an der laufenden Adresse beinhaltet.

7. In der Zeile 65121 wird überprüft, ob der Variablenname ein Teil des AVA ist. Wollen Sie irgendwelche Variablen-Namen umgehen bzw. nicht analysieren, so fügen Sie einfach diese Variablennamen in diese Liste ein oder machen hier eine Änderung, so daß nur noch die von Ihnen bestimmten Variablen angezeigt werden. Falls eine Variable in der Liste steht, wird ZZ$ in einen Null-String umgewandelt.

8. Das Unterprogramm 65130 erstellt den Variablen-Namen.

9. Im Unterprogramm 65140 steht die Logik für den korrekten Aufruf des jeweiligen Unterprogramms für Integer, String, einfache Dichte und doppelte Dichte.

10. Falls Sie keine Anzeige der CVI-, CVS- und CVD-Umwandlungen für 2-, 4- und 8-Byte Strings wollen, so können Sie die Zeile 65152 löschen.

11. Falls Sie sich eine LPRINT-Version des AVA anfertigen, müssen Sie unter Umständen den Wert '191' in Zeile 65156 in einen niedrigeren Wert verändern, z.B. 128. Manche Drucker verwenden ASCII-Werte über 128 für spezielle Kontrollcodes.

DAS SUPER-KOPIERFELD

Viele besondere Effekte und schnelle Techniken beinhalten im Prinzip nicht mehr als das Bewegen, oder genauer bezeichnet 'Kopieren', eines Datenblockes von einer Speicherstelle zu einer anderen. Mit speziellen Hilfsroutinen in Z-80 Maschinensprache können wir diese Funktion sehr schnell ausführen. Wir geben einfach an, von welcher Adresse wieviele Bytes nach welcher Adresse übertragen werden sollen.

Zum Kopieren von Daten gibt es mit LDIR und LDDR in Z-80 zwei besonders sinnvolle Anweisungen. Um die Arbeitsweise besser erklären zu können, wollen wir einmal annehmen, daß wir einen Block mit 16 Bytes im Speicher haben. Wir wollen Sie mit 0 beginnend durchnummerieren. Natürlich können Sie an jeder Stelle zwischen 0 und 65535 stehen. Weiterhin nehmen wir an, daß die ersten 4 Bytes dieses Speicherblockes das Wort 'DATA' beinhalten.

```
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
 D  A  T  A  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
```

Um das Wort 'DATA' auf die Speicherstelle 6 zu kopieren, wird das LDIR-Kommando zunächst das 'D' an die Stelle 6 kopieren, dann 'A' nach 7, 'T' nach 8 und 'A' nach 9. Nun sieht unser Speicherblock so aus:

```
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
 D  A  T  A  ?  ?  D  A  T  A  ?  ?  ?  ?  ?  ?  ?
```

Wir haben soeben 4 Bytes von der Speicherstelle 0 nach 6 kopiert.

Mit der LDDR-Anweisung können wir das selbe Ergebnis erreichen, nur beginnt LDDR mit dem am Ende stehenden 'A' in 'DATA' und arbeitet sich bis zum 'D' zurück. Zuerst wird 'A' von der Speicherstelle 3 nach 9 kopiert, dann 'T' von 2 nach 8, 'A' von 1 nach 7 und 'D' von 0 nach 6.

Welche dieser beiden Methoden verwendet wird ist egal, solange sich die Ursprungsadresse und die Bestimmungsadresse nicht überschneiden. Aber nehmen wir mal an, Sie wollen 4 Bytes von Stelle 0 nach 1 bewegen. Wir beginnen wieder mit dem ursprünglichen Speicherinhalt. Die LDIR-Anweisung kopiert das 'D' von 0 nach 1. Nun wird der Inhalt der Speicherstelle 1 auf die Speicherstelle 2 kopiert. In der Speicherstelle 1 steht aber nun nicht mehr das 'A' aus 'DATA' sondern das bereits kopierte 'D'. Unser Ergebnis sieht so aus:

```
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
 D  D  D  D  D  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
```

Hier müssen wir also die LDDR-Anweisung benutzen, die ja das Kopieren mit dem letzten Byte beginnt. Um das Wort 'DATA' um 1 Stelle nach vorne zu kopieren lassen wir die LDDR-Anweisung 4 Bytes von Position 3 nach 4 kopieren.

Ausgehend vom ursprünglichen Speicherinhalt erhalten wir:

```
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
 D  D  A  T  A  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?  ?
```

In der Z-80 Maschinensprache arbeiten die LDIR- und die LDDR-Anweisung mit den Inhalten von 3 Registern: HL, DE und BC (wer nicht Z-80 versteht, soll sich HL, DE und BC einfach als 3 Integer-Variablen in BASIC vorstellen). Das HL-Register bezeichnet die 'von'-Adresse, das DE-Register bezeichnet die 'nach'-Adresse und das BC-Register gibt an, wie oft ein Byte von einer Adresse zu einer anderen kopiert werden soll. Die LDIR-Anweisung erhöht die 'von'- und 'nach'-Adresse nach jedem kopierten Byte. Die LDDR-Anweisung erniedrigt die 'von'- und 'nach'-Adresse nach jedem kopierten Byte. Das BC-Register wird bei LDIR und LDDR nach jedem kopierten Byte erniedrigt. Wenn BC 0 erreicht, ist die Kopieranweisung beendet.

Wir können diese schnellen Kopiermöglichkeiten sehr gut in ein 'Super-Kopierfeld' umsetzen. Wir laden einfach die benötigten Z-80 Codes in ein Integerfeld von 8 Elementen. Mit DEFUSR legen wir das erste Element des Feldes als Adresse dersse der USR-Routine fest und mit der USR-Routine führen wir das Kopieren aus.

Dies sind die für das 'Super-Kopierfeld' benötigten Z-80 Codes:

Element 0:	8448
Element 1:	'von'-Adresse
Element 2:	4352
Element 3:	'nach'-Adresse
Element 4:	256
Element 5:	Anzahl der zu kopierenden Bytes
Element 6:	-20243 für LDIR oder -18195 für LDDR
Element 7:	201

Die Elemente 0, 2, 4 und 7 können bereits vorab mit den richtigen Elementen geladen werden, da diese ja immer die gleichen sind. Sie können auch das Element 6 bereits mit -20243 laden, falls Sie keine überlappenden Bereiche haben bzw. wenn Sie keine LDDR-Anweisung benötigen.

Zur Demonstration einiger Kopieraktionen spielen wir mit dem Bildschirmspeicher der die Adressen 15360 bis 16383 belegt. Geben Sie hierzu das folgende Programm ein:

```
 10  DEFINT A-Z: J=0: A$=""
 20  US(0)=8448: US(2)=4352: US(4)=256: US(7)=201
 30  CLS: PRINT "KOPIERFELD DEMO"
 40  PRINT @64, "VON            ";: INPUT US(1)
 50  PRINT "NACH            ";: INPUT US(3)
 60  PRINT "ANZAHL BYTES:   ";: INPUT US(5)
 70  PRINT "I=LDIR, D=LDDR  ";: INPUT A$
 80  IF A$="D" THEN US(6)=-18195 ELSE US(6)=-20243
 90  DEFUSR=VARPTR(US(0)): J=USR(0)
100  GOTO 40
```


Bevor Sie Ihr Demo-Programm starten speichern Sie es ab und nehmen als Vorsichtsmaßnahme alle Disketten aus den Laufwerken. Falls Sie nämlich zufällig falsche Zahlen eingeben, können Sie Daten auf Speicherstellen kopieren, die BASIC- oder DOS-Zeiger enthalten. Bei der Kopier-Routine ist es wichtig zu wissen, wohin die Daten kopiert werden und wieviele Daten kopiert werden. Wenn Sie die folgenden Beispiele sorgfältig nachvollziehen können keine Fehler auftreten.

Beispiel 1: Wir wollen die obere Bildschirmhälfte auf die untere kopieren. Starten Sie das Programm:

von = 15360
nach = 15872
Anzahl Bytes = 512
'I' für LDIR

Beispiel 2: Der Titel 'KOPIERFELD DEMO' soll von Position 0 nach 32 kopiert werden:

von = 15360
nach = 15392
Anzahl Bytes = 15
'I' für LDIR

Beispiel 3: Die ersten 512 Bytes des ROM sollten auf der unteren Bildschirmhälfte ausgegeben werden:

von = 0
nach = 15872
Anzahl Bytes = 512
'I' für LDIR

Beispiel 4: Wir täuschen ein Verschieben der eben aus dem ROM kopierten Daten um 1 nach rechts vor:

von = 1
nach = 15872
Anzahl Bytes = 512
'I' für LDIR

Beispiel 5: Wir wollen 'KOPIERFELD DEMO' überlappend kopieren um 11 Positionen nach rechts was uns als Ergebnis 'KOPIERFELD KOPIERFELD DEMO' liefert.

von = 15375
nach = 15386
Anzahl Bytes = 16
'D' für LDDR

Beispiel 6: Der Bildschirm wird mit 'K' gefüllt (in der Annahme, daß an der Bildschirmposition 0 ein 'K' steht):

von = 15360
nach = 15361
Anzahl Bytes = 1023
'I' für LDIR

Viele andere Beispiel sind möglich. Geben Sie für die Anzahl der zu kopierenden Bytes nie eine 0 ein. Das ist sehr wichtig, denn eine 0 als Parameter für BC bedeutet bei LDIR und LDDR, daß 65536 mal kopiert wird. Ein katastrophaler Speicherinhalt ist die Folge.

Die folgende Tabelle gibt Ihnen eine brauchbare Auskunft über die möglichen Ausführungsarten mit dem 'Super-Kopierfeld' und gibt an, wie die Elemente 1, 3, 5 und 6 zu belegen sind:

NICHT ÜBERLAPPENDES KOPIEREN NACH OBEN UND UNTEN

Element 1 (HL)	=	'von'-Adresse
Element 3 (DE)	=	'nach'-Adresse
Element 5 (BC)	=	Anzahl der zu kopierenden Bytes
Element 6 (LDIR)	=	-20243

ÜBERLAPPENDES KOPIEREN NACH OBEN

Element 1 (HL)	=	höchste Adresse des zu kopierenden Blockes
Element 3 (DE)	=	höchste Adresse des Ziels
Element 5 (BC)	=	Anzahl der zu kopierenden Bytes
Element 6 (LDDR)	=	-18195

ÜBERLAPPENDES KOPIEREN NACH UNTEN

Element 1 (HL)	=	'von'-Adresse
Element 3 (DE)	=	'nach'-Adresse (kleiner wie 'von'-Adresse)
Element 5 (BC)	=	Anzahl der zu kopierenden Bytes
Element 6 (LDIR)	=	-20243

VERVIELFÄLTIGUNG EINER BYTE-FOLGE NACH OBEN

Element 1 (HL)	=	Adresse des 1. Byte der Byte-Folge
Element 3 (DE)	=	Adresse des 1. Byte des 1. Duplikat
Element 5 (BC)	=	Die Anzahl der Duplikate (ohne Original) ist zu multiplizieren mit der Länge der Byte-Folge.
Element 6 (LDIR)	=	-20243

VERVIELFÄLTIGUNG EINER BYTE-FOLGE NACH UNTEN

Element 1 (HL)	=	Adresse des letzten Byte der Byte-Folge
Element 3 (DE)	=	Adresse des ersten Byte der Byte-Folge - 1
Element 5 (BC)	=	Die Anzahl der Duplikate (ohne Original) ist zu multiplizieren mit der Länge der Byte-Folge.
Element 6 (LDDR)	=	-18195

Einige Beispiele für die Möglichkeiten mit dem 'Super-Kopierfeld':

1. Einfügen und Löschen auf dem Bildschirm
2. Auf- und abrollen des Bildschirms oder Teile davon. In oder von geschützten Speicherbereich kopieren.
3. Sichern des Bildschirms in einem geschützten Speicherbereich um in später wieder laden zu können.
4. Daten in geschützten Speicher kopieren, so daß diese an andere Programmer übergeben werden können.
5. Einfügen und Löschen von Feld-Elementen.
6. Kopieren von Daten aus einem Random-Disk-Buffer direkt (ohne Feld) in den Bildschirmspeicher. Abspeichern des Bildschirms auf Diskette, wobei Jeweils 256 Bytes in den Buffer kopiert werden, um dann mit 'PUT' auf Diskette abgespeichert abgespeichert zu werden.
7. Verschieben einer veränderlichen USR-Routine von einer Speicheradresse zu einer anderen.
8. Schnelles Laden numerischer Feldelemente von Diskette und schnelles Übertragen numerischer Felder auf Diskette.
9. Speicher bereinigen oder sich öfters wiederholende Byte-Folgen kopieren. Graphische Effekte.
10. Sofortiges Kopieren von Feldelementen.
11. Daten oder USR-Routinen direkt aus dem Disketten-Buffer in den geschützten Speicher kopieren.

Wie Sie sehen, ist das Super-Kopierfeld sehr nützlich und sehr schnell. Auf einige Besonderheiten kommen wir noch in anderen Teilen dieses Kurses zu sprechen.

EINE USR-ROUTINE ZUM DATEN KOPIEREN

Diese Routine kopiert sofort einen Speicherblock von einer Adresse zu einer anderen. Diese Kopier-USR-Routine kann bis auf folgende Unterschiede die gleichen Funktionen wie das 'Super-Kopierfeld' ausführen:

1. Sie können der USR-Routine die Angaben der 'von' und 'nach' Adresse sowie die Anzahl der zu kopierenden Bytes als einfachen BASIC-Ausdruck übergeben. Damit wird die Programmierung bequemer.

2. Die USR-Routine verwaltet jede Kopierrichtung, auch überlappendes Kopieren nach oben oder unten. Sie brauchen nicht entscheiden, ob LDIR oder LDDR verwendet werden soll.

3. Das 'Super-Kopierfeld' benötigt 16 Bytes, die Kopier-USR-Routine 88 Bytes. Dennoch wird die Kopier-USR-Routine in den meisten Fällen günstiger sein, da das BASIC-Programm weniger Operationen durchführen muß und eine einfache Übergabe der notwendigen Werte möglich ist.

4. Die USR-Routine verwendet den 'Vielwert-Verwalter'. Daher müssen Sie zunächst entscheiden, welche USR-Nummer (0-9) Sie verwenden und Sie müssen in Abhängigkeit von Ihrem DOS zwei Bytes abändern.

Um einen Aufruf der USR-Routine von BASIC aus erklären zu können, nehmen wir an, daß wir die obere Hälfte des Bildschirms auf die untere kopieren wollen. In der Annahme, daß die Routine geladen ist und als USR0 definiert worden ist, lautet unsere Anweisung:

```
J = USR(15360) OR USR(15872) OR USR(512)
```

Um den Inhalt der 1. Bildschirmzeile um eine Position nach rechts wandern zu lassen schreiben Sie:

```
J = USR(15360) OR USR(15361) OR USR(63)
```

Um die 1. Zeile um eine Position nach links wandern zu lassen:

```
J = USR(15361) OR USR(15360) OR USR(63)
```

Um jeden beliebigen Teil des Bildschirms nach oben rollen zu lassen, können Sie schreiben:

```
J = USR(15360+LI+64) OR USR(15360+LI) OR USR(64*(LV-1))
PRINT @ 15360+LI+(LV-1)*64, CHR$(30);
```

Dabei beinhaltet LI% den Anfang des zu rollenden Teils als PRINT @ - Position und LV% die Anzahl der zu rollenden Zeilen.

Wie Sie wahrscheinlich schon gemerkt haben, rufen Sie die Routinen mit einem Ausdruck folgenden Formats auf:

```
J% = USR(F%) OR USR(T%) OR USR(B%)
```

Hierbei bedeuten die Integer-Variablen folgendes:

- J% ist eine Schein-Variable.
- F% ist eine Integer-Variable, Konstante oder Ausdruck für die 'von'-Adresse.
- T% ist eine Integer-Variable, Konstante oder Ausdruck für die 'nach'-Adresse.
- B% beinhaltet die Anzahl der zu kopierenden Bytes. WICHTIG: B% darf nicht 0 sein!


```
          00000 ;KOPIER-USR-ROUTINE
          00001 ;
F000      00100        ORG   0F000H       ;ANFANG (VERÄNDERLICH)
          00110 ;
          00120 ;DIE FOLGENDE LOGIK NIMMT DIE 3 ARGUMENTE AN
          00130 ;
F000 CD7F0A    00140        CALL  0A7FH        ;ÜBERGEBE WERT AN HL
F003 00        00150        NOP                ;NOP FÜR AUSGLEICH
F004 DD2A145B  00160        LD    IX,(05B14H)  ;IX BEINHALTET DEFUSR-ADR.
F008 DD7531    00170        LD    (IX+49),L
F00B DD7432    00180        LD    (IX+50),H
F00E DD340A    00190        INC   (IX+10)
F011 DD340A    00200        INC   (IX+10)      ;ERHÖHE ZÄHLER UM 2
F014 DD340D    00210        INC   (IX+13)
F017 DD340D    00220        INC   (IX+13)      ;ERHÖHE ZÄHLER 2 UM 2
F01A DD7E0A    00230        LD    A,(IX+10)
F01D 0631      00240        LD    B,49
F01F 90        00250        SUB   B            ;A = ÜBERGEBENE WERTE * 2
F020 DD4630    00260        LD    B,(IX+48)    ;B = VERBLIEBENE WERTE * 2
F023 90        00270        SUB   B
F024 2801      00280        JR    Z,PASS1      ;WENN 0, KEINE WEITEREN W.
F026 C9        00290        RET                ;ANDERNF. RETURN ZU NEXT
F027 DD360A31  00300 PASS1  LD    (IX+10),49
F02B DD360D32  00310        LD    (IX+13),50   ;ZÄHLER ZURÜCKSETZEN
F02F 1806      00320        JR    START
F031 0000      00330        DEFW  0            ;SPEICHER 'VON'-ADRESSE
F033 0000      00340        DEFW  0            ;SPEICHER 'NACH'-ADRESSE.
F035 0000      00350        DEFW  0            ;SPEICHER FÜR BYTE-ANZAHL
          00351 ;
          00352 ;DIE FOLGENDE LOGIK FÜHRT KOPIEREN AUS
          00353 ;
F037 E5        00360 START  PUSH  HL
F038 C1        00370        POP   BC
F039 DD6E31    00380        LD    L,(IX+49)
F03C DD6632    00390        LD    H,(IX+50)
F03F E5        00400        PUSH  HL
F040 DD5E33    00410        LD    E,(IX+51)
F043 DD5634    00420        LD    D,(IX+52)
F046 B7        00430        OR    A
F047 ED52      00440        SBC   HL,DE
F049 E1        00450        POP   HL
F04A 3803      00460        JR    C,MOVEUP
F04C EDB0      00470        LDIR
F04E C9        00480        RET                ;RÜCKSPRUNG INS BASIC
F04F 09        00490 MOVEUP ADD   HL,BC
F050 2B        00500        DEC   HL
F051 EB        00510        EX    DE,HL
F052 09        00520        ADD   HL,BC
F053 2B        00530        DEC   HL
F054 EB        00540        EX    DE,HL
F055 EDB8      00550        LDDR
F057 C9        00560        RET                ;RÜCKSPRUNG INS BASIC
F04F           00570        END
00000 TOTAL ERRORS
```

'Super-Feld'-Format, 44 Elemente:

32717	10	10973	23316	30173	-8911	12916	13533	-8950
2612	13533	-8947	3380	32477	1546	-28623	18141	-28624
296	-8759	2614	-8911	3382	6194	6	0	0
-6912	-8767	12654	26333	-6862	24285	-8909	13398	-4681
-7854	824	-20243	2505	-5333	11017	-4629	-13896	

POKE-Format, 88 Bytes:

205	127	10	0	221	42	20	91	221	117	49	221	116	50	221	52
10	221	52	10	221	52	12	221	52	13	221	126	10	6	49	144
221	70	48	144	40	1	201	221	54	10	49	221	54	13	50	24
6	0	0	0	0	0	0	229	193	221	110	49	221	102	50	229
221	94	51	221	86	52	183	237	82	225	56	3	237	176	201	9
43	235	9	43	235	237	184	201								

```
10 DEFINT A-Z: J=0

30 'LADEN DER KOPIER-USR-ROUTINE IN EIN SUPER-DATENFELD
31 DATA 32717,10,10973,23316,30173,-8911,12916,13533,-8950,
   2612,13533,-8947,3380,32477,1546,-28623
32 DATA 18141,-28642,296,-8759,2614,-8911,3382,6194,6,0,0,
  -6912,-8767,12654,26333,-6862
33 DATA 24285,-8909,13398,-4681,-7854,824,-20243,2505,-5333,
  11017,-4629,-13896
34 DIM UX(43): FOR X=0 TO 43: READ UX(X): NEXT

100 CLS: PRINT "KOPIER-USR-ROUTINE - DEMO UND UTILITY"
110 PRINT @128, "KOPIERE VON:          ";: INPUT MF%
120 PRINT "KOPIERE NACH:           ";: INPUT MT%
130 PRINT "ANZAHL DER BYTES:       ";: INPUT NB%
131 IF NB%=0 THEN 130
140 DEFUSR=VARPTR(UX(0))
150 J=USR(MF%) OR USR(MT%) OR USR(NB%)
160 GOTO 110
```

Die Kopier-USR-Routine wird im 'Super-Datenfeld'-Format, im POKE-Format und als Assemblerlisting gezeigt. Die vorgestellte Version läuft als USR0 unter NEWDOS 2.1. Um sie als eine andere USR-Routine (USR1-USR9) unter NEWDOS 2.1 benutzen zu können, oder wenn Sie ein anderes DOS verwenden, so schlagen Sie im Anhang 2 nach und befolgen nachstehende Anleitungen:

1. Bei der Ausführung als 'Super-Datenfeld' ersetzen Sie das 4. Element, 23316, mit dem entsprechenden Integerwert aus dem Anhang 2. Wenn Sie z.B. TRSDOS 2.3 benutzen, und die Kopier-USR-Routine als USR6 ausführen wollen, so finden Sie im Anhang 2 den Wert 5B83 was dezimal 23427 entspricht. Das 4. Element lautet also 23427.

2. Falls Sie das POKE-Format verwenden, so ersetzen Sie das 7. und 8. Byte, 20 und 91, mit den benötigten Bytes aus dem Anhang 2. Wenn Sie z.B. NEWDOS 2.1 benutzen, und die Routine als USR9 ausführen wollen, so finden Sie im Anhang 5B26. Das 7. Byte wird 26 (dezimal 38) und das 8. Byte 5B (dezimal 91).

3. Beim Assemblerlisting ist der Wert 5B14 in Zeile 160 durch den benötigten Hexadezimalwert zu ersetzen.

Das obenstehende Demo-Programm zur Kopier-USR-Routine fragt nach der 'von'- und 'nach'-Adresse und nach der Anzahl der zu kopierenden Bytes. Die Routine wird aus DATA-Zeilen in ein 'Super-Datenfeld' geladen, so daß Sie keinen Speicher schützen müssen. Denken Sie daran, daß Sie in Zeile 31 den Wert 23316 ändern müssen, falls Sie ein anderes DOS verwenden.

Ein weiteres Demo im Zusammenhang mit der Kopier-USR-Routine erhalten Sie, wenn Sie die Zeilen 100 bis 160 löschen und nachstehende Zeilen einfügen. Das Demo liefert eine schnelle, sichtbare 'Reise' durch den Speicher und zeigt sehr schön die gewaltigen Möglichkeiten dieser Routine. Die letzte Bildschirmzeile zeigt die gegenwärtige Adresse in Abständen von 64, während der Speicherinhalt in der oberen Hälfte des Bildschirms rollt.

```
100 CLS: A=0: DEFUSR=VARPTR(UX(0))
110 FOR X=-1 TO 32766 STEP 64: A=X+1: GOSUB 200: NEXT
120 FOR X=-32768 TO 0 STEP 64: A=X: GOSUB 200: NEXT
130 END
200 PRINT @990,A;: J=USR(A) OR USR(15360) OR USR(960): RETURN
```

Vorschlag zur Programmbibliothek: a b s c h a f f e n

Im Clubinfo Nr. 5 erklärt Hartmut im Zusammenhang mit der Programmbibliothek, er werde auf weitere Vorschläge eingehen. Hartmut, mein Vorschlag ist so radikal, daß Du hoffentlich nicht zu Deinem Wort stehst, ohne beim Clubtreffen ein Votum dazu einzuholen.

Daß unsere Softwaresammlung juristisch heikel ist, wurde schon des langen und breiten diskutiert. Ob sie überhaupt sinnvoll ist, danach hat bisher komischerweise niemand gefragt. Im folgenden will ich Denkanstöße zu dieser Frage geben. Jeder mag sie für sich selber beantworten. Für mich persönlich ist die Antwort klar: Nein.

Nach der Reinigung der Bibliothek von Fremdprogrammen bleibt voraussichtlich ein schmaler Rest von Brauchbarem. Ihn mit dem bisherigen Aufwand von einem Club-Amtsinhaber verwalten zu lassen, entbehrt nicht einer gewissen Komik. Die Gesamtmenge, in Bytes gemessen, bleibt zwar vielleicht verwaltungswürdig, aber bei allem Respekt vor den Selbstdenkern: So richtigen Schmiß wie die ganz großen Raubkopien in der bisherigen Sammlung hat doch kaum ein Eigenbauprogramm.

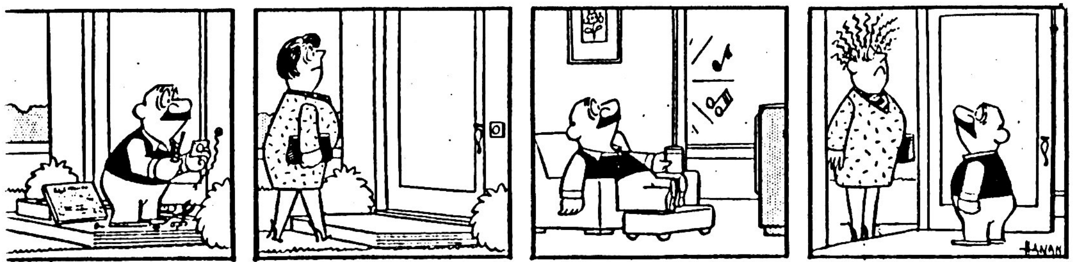
Das alleine hätte mich aber noch nicht bewogen, mich mit meiner Meinung beim Rest des Clubs möglicherweise gleich am Anfang meiner Mitgliedschaft unbeliebt zu machen. Der wesentliche Grund ist ein ganz anderer:

Jeder von uns startet mit 30 Punkten. Wer aus Zeitgründen, oder weil er nun mal nicht so drauf ist, keine eigene Software liefern kann (oder nicht, wie manche Mitglieder, die Chuzpe besitzt, irgendwelchen Daddelkram einzureichen), ist für den Rest seines Clubdaseins von der Teilhabe ausgeschlossen. Was hat das noch mit einem Club zu tun?

Und umgekehrt: Mein H-DOS, falls man es mit 5 Pkt. bewerten sollte, soll mir nur 25 Bonuspunkte einbringen? Ein paar Videospiele für 200 Arbeitsstunden? Da verschenke ich es lieber. Und genau das ist die Methode in anderen Clubs: Software kostet nichts. Man tauscht Listen, äußert Wünsche, fragt nicht nach Copyrights oder danach, wieviele Bytes in welche Richtung gehen und hat so in kurzer Zeit dasselbe, was alle haben: Alles!

Da es die Programmbibliothek aber noch (?) gibt, will ich mich nicht gegen die Clubregeln stellen und meine Programme nicht unter Umgehung der Clubsammlung anbieten. Dabei ist es mir vollkommen egal, wer noch alles von meinem Freizeitspaß an der Tastatur profitiert. Jegliches Monopoly-Denken liegt mir fern. Aber da sind nun mal die Clubregeln. Noch lange?

Arnulf Sopp, Tel. 0451-791926



Anmerkungen zur Programmbibliothek:
(bzw. zum Vorschlag von A. Sopp dieselbe abzuschaffen)

Im vorstehenden Artikel macht unser neues Mitglied A. Sopp den Vorschlag, die Programmbibliothek abzuschaffen. Meiner Meinung nach tut er das etwas vorschnell.

Hier nun ein paar Punkte, die klar für die Weiterführung der Programmsammlung sprechen und die die Beibehaltung des Punktesystems zumindest sinnvoll erscheinen lassen.

1. Auch eine von allen Raubkopieen (sehr oft Spiele, die sowieso fast jeder besitzt) befreite Programmsammlung enthält noch sehr viele interessante Programme. Der Schmiß ist, das kann ich dem Arnulf versichern, nicht raus. Im Gegenteil, da die sieben Pacman-Versionen jetzt verschwunden sind, fallen die Selbstentwicklungen (die nicht überall zu haben sind) und die abgetippten Listings (die abzutippen sich doch die meisten aus Zeitmangel scheuen) umso deutlicher ins suchende Auge.

2. Selbstverständlich bleibt jedem die Möglichkeit, privat zu Tauschen mit wem es ihm beliebt. Dabei sind aber zwei Punkte, die jeder bedenken sollte:

Wenn ich innerhalb oder außerhalb des Clubs tausche, kann ich jederzeit an einen "Fänger" geraten! Das ist einem unserer Mitglieder ja schon passiert, und zwar beim Tausch mit einem anderen Clubmitglied. Die Sache birgt also ein gewisses Risiko.
Wer mit allen Clubmitgliedern (inzwischen über 40) persönlich tauschen möchte, muß sich auf einen erheblichen Briefverkehr einrichten. Allein die Durchsicht aller Tauschlisten würde eine Menge Zeit in Anspruch nehmen.

Die Vorteile einer Clubbibliothek dagegen, sind:

Nur eine Tauschliste für alle Programme und keine Angst vor einem Gerichtsverfahren wegen Verletzung von Rechten anderer!!!

3. Damit die Programmbibliothek jedoch vollständig ist und möglichst viele wertvolle Programme enthält, muß ein Anreiz dafür her, daß alle Mitglieder die von Ihnen erstellten oder eingegebenen Programme auch einsenden. Eine mögliche Lösung dieses Problems besteht in der Beibehaltung der Punkteregelung. Eventuell gibt es auch andere, vielleicht sogar bessere Lösungen für dieses Problem und alle Mitglieder sind dazu aufgefordert, sich eine solche zu überlegen und sie beim Clubtreffen vorzubringen.

4. Es gibt hie und da Mitglieder (die gibt es in jedem Club oder Verein) die aus dem CLUB 80 möglichst viel herausholen wollen, ohne etwas in ihn zu investieren (der Clubbeitrag deckt gerade so die Kosten für das Clubinfo). Solche Mitglieder sollen nicht etwa dem Club ferngehalten werden sondern sie sollen die Clubkasse aufbessern, indem sie sich die für gewünschte Programme fehlenden Programme erkaufen (1.- DM pro Punkt). Dadurch können auch sie einen erheblichen Beitrag zur Clubexistenz beitragen.
Computerfreunde, die im Selbstprogrammieren nicht so fit sind, können Listings abtippen und diese einsenden. Sie ersparen damit anderen viel Zeit und sind durchaus nicht vom Clubleben ausgeschloßen, wie der Arnulf befürchtet.

Ich hoffe, diese kurze Klarlegung einiger, vielleicht für den Neuling undurchsichtiger Punkte, macht deutlich, daß auch wir (der Günther und ich) nicht an Monopolydenken leiden sondern von der Notwendigkeit der Clubbibliothek überzeugt sind.

Hartmut Obermann

Tape Transfer

by J.L. Kissel

Model III NEWDOS80 2.0 users can't load tape-based source files because NEWDOS's EDTASM doesn't support Model III tape input/output (I/O). While the DOS's documentation describes how to load a source file from tape, zap 30 states that NEWDOS supports tape I/O for the Model I only.

I decided to rectify this. Rather than try to add Model III tape I/O capability to NEWDOS80 2.0's EDTASM, however, I modified the LMOFFSET program so it would load source files from tape to disk. You can do so by adding the LMOFFSET zaps indicated in the Figure. The Program Listing shows the source code of the patch.

To add the zaps yourself, copy LMOFFSET and save it under a new file name. (The zaps destroy LMOFFSET's ability to handle system tapes and support tape-to-disk loading only.) Then use SuperZap to make the changes listed in the Figure. ■

For instructions on how to assemble source code, consult your editor/assembler manual or "An Idiot's Guide to Assembly Language," Parts I and II (80 Micro, May 1981, p. 168, and June 1981, p. 112).

Contact J.L. Kissel at 287 London Road, Isleworth, Middlesex, England.

The Key Box

Model III
NEWDOS80 2.0

```
At 00,58 change:
3E FF CD 57 54 CD 3D 58
to:
3E 00 CD 7B 54 CD 7B 54

At 00,63 change:
FE 55 28 0A CD
to:
FE D3 28 0C CD

At 00,73 change:
CD 57 54
to:
CD 7B 54

At 00,78 change:
58 CD 35 02 FE 78 CA CA
54 FE 3C 28 05 CD 92 54
18 EC 3E 01 CD 7B 54 CD
35 02 47 C6 02 CD
to:
58 06 FF CD 35 02 CD 7B
54 FE 1A CA 32 54 10 F3
CD 35 02 CD 7B 54 FE 1A
CA 32 54 18 18

At 00,B7 change:
58 CD 35 02 FE 3C CA F8
53 FE 78 CA CA 54 F5 3E
49 32 3D 3C 3E FD CD 57
54 F1 C3
to:
58 ED 5B 9D 5E 13 ED 53
9D 53 C3 E7 53 3E 00 CD
7B 54 10 F9 CD 7B 54 C3
D5 54 C3

At 01,6F change:
C3 50 55 21
to:
C3 FC 56 21

At 03,A1 change:
F8 CD 20 44 C2 57 58 D9
21 9D 5E CD 33 57 30 FB
FE 01 28 0A FE 02 CA 44
57 CD 5C 57 18 0B CD 5C
57 05 05 CD 60 57 CD 60
57 CD 60 57 10 FB 18 DB
7E FE 20 D8 FE FE DA 48
58 23 5E 23 56 23 19 AF
C9 CD 5C 57 CD 5C 57 D9

78 B7
to:
F8 21 00 52 11 00 53 06
00 CD 20 44 D9 21 9F 5E
22 74 53 2A 9D 5E 23 23
22 9D 5E ED 5B 9D 5E 1B
ED 53 9D 5E 7A B3 CA 45
57 0E 00 06 01 2A 74 53
11 00 52 ED B0 22 74 53
D9 CD 3C 44 C2 9D 5E D9
18 D9 D9 CD 28 44 C3 59
57 B7
```

Figure. Zaps to LMOFFSET. Zap locations appear in sector and offset format (e.g., XX,YY means sector XX, offset YY).

Program Listing. Patch, the LMOFFSET modification program.

```
                 ;
                 ;          LMOFFSET PATCHES
                 ;
                 ;TO ALLOW LMOFFSET TO READ A EDTASM TAPE AND
                 ;WRITE IT TO A DISK FILE FOR USE BY APPARAT
                 ;DISK BASED EDTASM
                 ;
                 ;DISK FORMAT MAY BE COMPATIABLE WITH OTHER EDTASM
                 ;
53C6                     ORG 53C6H          ;START OF PATCH
                 ;CASSETTE IS ON AND SYNC HEADER HAS BEEN READ
53C6 3E00                LD A,0H            ;ZERO THE
```

Listing continued


```
573B D9                  EXX                ;FOR PROPER FCB INFO
573C CD3C44              CALL 443CH         ;WRITE A SECTOR
573F C29D5E              JP NZ,5E9DH        ;IF DISK ERROR
5742 D9                  EXX                ;BACK TO TRANSFER POINTERS
5743 18D9                JR WRDSK1          ;LOOP BACK FOR NEXT SECTOR
5745 D9          WRDSK2  EXX                ;ORIGINAL REGS
5746 CD2844              CALL 4428H         ;CLOSE FILE
5749 C35957              JP 5759H           ;TO DONE MESSAGE
0000                     END
00000 TOTAL ERRORS
32543  TEXT AREA BYTES LEFT
```



```
53C8 CD7B54              CALL 547BH         ;BLOCK COUNT
53CB CD7B54              CALL 547BH         ;I.E. NUMBER OF SECTORS TO WRITE
53D1                     [illegible] 53D1H
                 ;
                 ;FIRST BYTE OF TAPE IS NOW IN THE A REG
                 ;
53D1 FED3                CP 0D3H            ;CHECK IF EDTASM
53D3 280C                JR Z,53E1H         ;TAPE AND JUMP IF IT IS
                 ;
                 ;WRITE FIRST BYTE TO BUFFER
                 ;
53E1                     ORG 53E1H
53E1 CD7B54              CALL 547BH         ;FIRST BYTE TO BUFFER
                 ;
                 ;53E4 WILL CHECK FOR THE UP ARROW KEY ABORT FUNCTION

                 ;
53E7                     ORG 53E7H
53E7 06FF        RDBLK1  LD B,255           ;SET UP LOOP COUNT
53E9 CD3502      RDBLK2  CALL 0235H         ;READ A BYTE FROM TAPE
53EC CD7B54              CALL 547BH         ;WRITE BYTE TO BUFFER
53EF FE1A                CP 1AH             ;CHECK IF
53F1 CA3254              JP Z,TAPFIN        ;END OF FILE
53F4 10F3                DJNZ RDBLK2        ;LOOP FOR 255 BYTES
53F6 CD3502              CALL 0235H         ;GET 256'S BYTE

53F9 CD7B54              CALL 547BH         ;AND SAVE IT
53FC FE1A                CP 1AH             ;CHECK IF
53FE CA3254              JP Z,TAPFIN        ;END OF FILE
5401 1818                JR 541BH           ;ELSE CONTINUE TO READ TAPE
                 ;
                 ;541B TWINKLES THE STARS AND CHECKS
                        FOR CANCELED FUNCTION
                 ;
5426                     ORG 5426H
5426 ED5B9D5E            LD DE,(5E9DH)      ;AND

542A 13                  INC DE             ;INCREMENT
542B ED539D5E            LD (5E9DH),DE      ;IT
542F C3E753              JP RDBLK1          ;CONTINUE READING TAPE
5432 3E00        TAPFIN  LD A,0H            ;ZERO
5434 CD7B54              CALL 547BH         ;THE REMAINING
5437 10F9                DJNZ TAPFIN        ;BYTES OF THE BLOCK
5439 CD7B54              CALL 547BH         ;
543C C3D554              JP 54D5H           ;TO TURN OFF TAPE
                 ;
                 ;CASSETTE IF OFF INTRUPTS ARE ON

                 ;NOW ASK FOR DESTINATION FILESPEC
                 ;
54D9                     ORG 54D9H
54D9 C3FC56              JP 56FCH           ;TO FILESPEC MESSAGE
                 ;
                 ;DISK FILE IS NOW OPEN STARTING WRITING TO DISK
                 ;
5704                     ORG 5704H
5704 210052              LD HL,5200H        ;START OF FCB'S BUFFER
5707 110053              LD DE,5300H        ;FCB ITSELF

570A 0600                LD B,0H ;256 BYTE RECORDS
570C CD2044              CALL 4420H         ;OPEN FILE
570F D9                  EXX                ;SAVE ASSOCIATED FCB INFO
5710 219F5E              LD HL,5E9FH        ;FIRST BYTE OF TAPE BUFFER
5713 227453              LD (5374H),HL      ;PGM POINTER TO TAPE BUFFER
5716 2A9D5E              LD HL,(5E9DH)      ;AJUST
5719 23                  INC HL             ;BLOCK
571A 23                  INC HL             ;COUNT
571B 229D5E              LD (5E9DH),HL      ;TO +1 OF TRUE VALUE
571E ED5B9D5E    WRDSK1  LD DE,(5E9DH)      ;CKECK

5722 1B                  DEC DE             ;IF
5723 ED539D5E            LD (5E9DH),DE      ;
5727 7A                  LD A,D             ;SECTOR
5728 B3                  OR E               ;COUNT
5729 CA4557              JP Z,WRDSK2        ;COMPLETED
572C 0E00                LD C,0H            ;ELSE SET FOR
572E 0601                LD B,1H            ;256 BUTE TRANSFER
5730 2A7453              LD HL,(5374H)      ;POINT FIRST BYTE TO TRANSFER
5733 110052              LD DE,5200H        ;FCB BUFFER FIRST BYTE
5736 EDB0                LDIR               ;MOVE A SECTORS WORTH OF DATA
5738 227453              LD (5374H),HL      ;STORE UPDATED POINTER TO BYTE
```

Ein Städter kommt in ein kleines Bergdorf und fragt einen Einheimischen: „Ich suche einen jungen Mann, der hier wohnen soll. 20 Jahre alt, groß, blond, kräftig, mit einem Ohr – namens Oskar!" „Hm", überlegt der Einheimische, „und wie heißt das andere Ohr?"

„Der Doktor hat gesagt, ich brauche dringend Luftveränderung", erzählt Frau McGregor ihrem Mann. „Da hast du aber Glück", meint der alte Schotte, „der Wind hat nämlich gerade gedreht!"

„Ich begreife nicht, wie du nur so faul sein kannst", schimpft der Vater mit seinem Sohn. „Für mich ist die Arbeit immer ein wahres Vergnügen!" „Aber, Papa, wir sind doch nicht nur zum Vergnügen auf die Welt gekommen!"

Tricks mit Strings

Ohne String-Bearbeitung wäre ein Computer nur eine simple Rechenmaschine. Komfortable Befehle in puncto Strings zeichnen deshalb eine gute Programmiersprache aus. Die meisten Basic-Dialekte beschränken sich auf das Notwendigste, den Rest muß man selbst schreiben.

Listing 1 ist sozusagen die Grundübung. Im String X$ soll das Zeichen C$ gesucht werden, und zwar ab Position P. Wurde das Zeichen gefunden, soll P diese Position als Wert erhalten, ansonsten wird P=0. Das Unterprogramm ab Zeile 100 führt den Job aus. Warum verwenden wir kein FOR-NEXT? Nun, wir wären gezwungen, die FOR-Schleife beim ersten »Finden« zu verlassen, und das ist von Übel. Der Interpreter legt die Schleifenparameter (Index-Name, Step und ähnliches) auf dem Stack ab. Erst wenn die Schleife ordnungsgemäß über NEXT verlassen wird, wird der Stack bereinigt. Im anderen Falle bleibt der Stackpointer an der alten Position. Beim nächsten Aufruf des Unterprogramms wird eine neue Schleife eröffnet, wieder gehen ihre Parameter auf den Stack und so weiter. Wiederholt man das Spielchen oft genug, ist der Stack mit Schleifen-Trümmern gefüllt und der Computer steigt aus. Listing 2 demonstriert, wie man in einem String einen anderen, Substring genannt, sucht. Haben Sie diese Routine in ein Programm eingebaut, brauchen Sie Listing 1 nicht mehr, denn ein Zeichen ist nichts weiter als ein Substring der Länge eins. Es passiert nichts weiter, als daß aus dem Zielstring Substrings gebildet und diese gegen den Suchstring verglichen werden. Eine andere Methode wäre, erst das erste Zeichen zu suchen. Wenn das gefunden ist, prüft man das zweite auf Übereinstimmung, dann das dritte und so weiter. Dieses Verfahren ist typisch für Assembler.

Strings verändert man durch Einfügen, Löschen und Ersetzen von Substrings. Auch hier gilt, daß für alle Funktionen ein Unterprogramm ausreicht. Löschen beispielsweise kann als Ersetzen durch einen Nullstring realisiert werden, aber das ist nicht unbedingt der optimale Weg. Listing 3 zeigt das Einfügen für diesen Fall: In X$ soll C$ nach Position P eingefügt werden. Zeile 100 ist alles, was dafür erforderlich ist. Fast genau so funktioniert das Löschen nach Listing 4. In X$ sollen L Zeichen ab Position P gelöscht werden. Das Ersetzen nach Listing 5 ist dann lediglich Löschen mit Einfügen kombiniert. Auch dazu reicht eine Zeile. Listing 6 wird in der Praxis häufiger gebraucht werden: In X$ soll ab Position P C$ gesucht und falls gefunden, durch E$ ersetzt werden. Dafür reicht eine Kombination von Listing 2 mit Listing 5.

Und jetzt die Tricks:

Listing 7 zeigt ein Menü, wie es häufig vorkommt. Ich könnte schreiben

IF A$="S" THEN...
IF A$="D" THEN...
IF A$="E" THEN...

Dafür reicht aber eine Zeile (hier 50), die darauf beruht, daß ein logischer Ausdruck wie »A$="S"« den Wert —1 annimmt, wenn die Bedingung zutrifft, und den Wert 0 erhält, wenn sie nicht erfüllt ist. Ist zum Beispiel in Zeile 50 der 3. Ausdruck wahr, dann ergibt —1*—3=+3. Alle anderen Ausdrücke bekommen den Wert 0. Sollten Sie einen Computer besitzen, der logisch wahr als +1 meldet (zum Beispiel Apple), dann brauchen Sie nur die negativen Vorzeichen wegzulassen. Wenn Sie schreiben »IF A< > =0 THEN...« ist das zwar richtig, aber "IF A THEN...« tut's oft auch. Basic prüft dann, ob A logisch wahr ist. Zeile 45 testet, ob in Kleinschrift eingegeben wurde, und wandelt gegebenenfalls in »uppercase«. Das beruht darauf, daß Kleinbuchstaben im ASCII-Code um 32 versetzt sind. Die Zeile 50 kann bei langen Menüs Ausmaße annehmen, die in Tipparbeit ausartet. Das sollte vermieden werden. Eine Lösung zeigt Listing 8. Alle erlaubten Antworten stehen in X$. In diesem wird nach bekanntem Muster das Zeichen A$ gesucht und anhand seiner Position verzweigt.

Listing 9 schließlich können Sie dann in die Menü-Routine einbauen, wenn die Antworten länger als 1 Zeichen sind. Aber auch dann einsetzen, wenn Sie Speicher sparen wollen. Mit einem String-Array kann man zwar bequemer eine Beziehung zum Beispiel zwischen Monatsnummer und -namen herstellen, jeder Eintrag kostet aber zusätzlich 3 Byte. Solange alle Einträge gleich lang sind, ist die Sache einfach. Die kleine Formel in Zeile 30 reicht.

Etwas komplizierter wird die Sache, wenn Sie im Menü verschieden lange Texte zulassen. Auch das wird mit den Monatsnamen, jetzt in Listing 10, demonstriert. Der Trick: Die Namen müssen mit einem Delimeter (hier der »/«) getrennt sein. In den Zeilen 50 bis 60 wird nach dem n-ten »/« (=Nummer) gesucht. Dessen Position zuzüglich 1 ist der Beginn des Substrings und wird in P1 gerettet.

Ab Zeile 80 wird nun das nächste »/« gesucht, dessen Position findet sich in P. Folglich ist P—P1—1 die Länge des Substrings, P1+1 war der Start, Zeile 90 kann mit dem Ausdruck beginnen.

Zeiger verbogen

Jede Variable, die Basic im Programm antrifft, wird in eine Tabelle (Vartab) in der Reihenfolge des Auftretens eingetragen. Wird einer Variablen ein neuer Wert zugewiesen, muß sie der Interpreter in Vartab suchen und dort den neuen Wert eintragen. Bei numerischen Variablen ist das kein Problem, zur Abspeicherung steht immer ein konstant großer Platz zur Verfügung. Nur bei Strings wird es schwierig, da die Länge variieren kann. Um nicht jedesmal die Tabelle zu verschieben, arbeitet Basic mit einem Trick. In Vartab wird die Länge des Strings (1 Byte) und seine Adresse (2 Byte) eingetragen. Der String selbst steht außerhalb der Tabelle im Stringpool, die Adresse sagt wo. Beim TRS-80 gibt es die Funktion VARPTR (Variable Pointer). Beim Apple und anderen gibt es zwei Bytes, die die Adresse der zuletzt benutzten Variablen halten. In Listing 11 wird in den Zeilen 25 bis 30 VARPTR simuliert. Zeile 50 holt dann erst das Längen-Byte (L) und schließlich noch die Adresse (A). In der folgenden Schleife wird dann A$ ausgedruckt. Mit »PRINT A$« ginge das einfacher, aber es sollte nur das System bewiesen werden.

Man kann in die 3 Bytes alles mögliche POKEn, zum Beispiel die Werte eines anderen Strings. Damit lassen sich Strings sehr schnell tauschen und sortieren. Oder man lädt das Längen-Byte mit der Länge einer Bildschirmzeile und die Adreßbytes mit deren Adresse. »LPRINT A$« gibt dann die Schirmzeile auf dem Drucker aus. Experimentieren Sie selbst!

(Peter Wollschläger/hg)

```
10 INPUT X$,C$,P
20 GOSUB 100:PRINT P
30 GOTO 10
100 L=LEN(X$):P=P-1
110 P=P+1
120 IF MID$(X$,P,1)=C$ THEN RETURN
130 IF P<L THEN 110
140 P=0:RETURN
```

Listing 1. So sucht man ein Zeichen in einem String

```
10 INPUT X$,C$,P
20 GOSUB 100:PRINT P
30 GOTO 10
100 L1=LEN(C$):L=LEN(X$)-L1+1:P=P-1
110 P=P+1
120 IF MID$(X$,P,L1)=C$ THEN RETURN
130 IF P<L THEN 110
140 P=0:RETURN
```

Listing 2. Schon besser, auch Worte werden gefunden

```
10 INPUT X$,C$,P
20 GOSUB 100:PRINT X$
30 GOTO 10
100 X$=LEFT$(X$,P)+C$+RIGHT$(X$,LEN(X$)-
P)
110 RETURN
```

Listing 3. Einfügen von Strings in einen String

```
10 INPUT X$,P,L
20 GOSUB 100:PRINT X$
30 GOTO 10
100 X$=LEFT$(X$,P-1)+RIGHT$(X$,LEN(X$)-P
-L+1)
110 RETURN
```

Listing 4. Teil eines Strings löschen

```
10 INPUT X$,P,L,C$
20 GOSUB 100:PRINT X$
30 GOTO 10
100 X$=LEFT$(X$,P-1)+C$+RIGHT$(X$,LEN(X$
)-P-L+1)
110 RETURN
```

Listing 5. Löschen + Einfügen = Ersetzen

```
10 INPUT X$,C$,P,E$
20 GOSUB 100:PRINT X$
30 GOTO 10
100 L1=LEN(C$):L=LEN(X$)-L1+1:P=P-1
110 P=P+1
120 IF MID$(X$,P,L1)=C$ THEN 150
130 IF P<L THEN 110
140 P=0:RETURN
150 X$=LEFT$(X$,P-1)+E$+RIGHT$(X$,LEN(X$
)-P-L1+1)
160 RETURN
```

Listing 6. Ersetzen mit Komfort: Der alte String darf genannt werden

```
10 PRINT"<S> SICHERN"
20 PRINT"<D> DRUCKEN"
30 PRINT"<E> ENDE"
40 INPUT A$
45 IF A$>"Z" THEN A$=CHR$(ASC(A$)-32)
50 ON (A$="S")*-1+(A$="D")*-2+(A$="E")*-
3 GOSUB 100,200,300
60 GOTO 10
100 PRINT "ROUTINE SICHERN":RETURN
200 PRINT "ROUTINE DRUCKEN":RETURN
300 PRINT "ROUTINE ENDE    ":ENDE
```

Listing 7. »ON GOSUB« funktioniert auch mit Strings

```
10 PRINT"<S> SICHERN"
20 PRINT"<D> DRUCKEN"
30 PRINT"<E> ENDE"
40 INPUT A$
45 IF A$>"Z" THEN A$=CHR$(ASC(A$)-32)
50 X$="SDE":L=3:P=0
51 P=P+1:IF MID$(X$,P,1)<>A$ AND P<=L TH
EN 51
52 ON P GOSUB 100,200,300
60 GOTO 10
100 PRINT "ROUTINE SICHERN":RETURN
200 PRINT "ROUTINE DRUCKEN":RETURN
300 PRINT "ROUTINE ENDE    ":ENDE
```

Listing 8. Das Menü in einem String versteckt

```
10 D$="JANFEBMARAPRMAIJUNJULAUGSEPOKTNOV
DEZ"
20 INPUT"MONATS-NR.";M
30 PRINT MID$(D$,3*M-2,3)
40 GOTO 20
```

Listing 9. Eine Reihe von Substrings spart Speicher

```
10 D$="/JANUAR/FEBRUAR/MAERZ/APRIL/MAI/J
UNI/JULI/AUGUST"
20 D$=D$+"/SEPTEMBER/OKTOBER/NOVEMBER/DE
ZEMBER/"
30 INPUT "MONATS-NR.";M
40 P=0:P1=0
50 P=P+1:IF MID$(D$,P,1)<>"/"THEN 50
60 P1=P1+1:IF P1<>M THEN 50
70 P1=P
80 P=P+1:IF MID$(D$,P,1)<>"/"THEN 80
90 PRINT MID$(D$,P1+1,P-P1-1)
100 GOTO 30
```

Listing 10. String-Reihe mit Delimeter

```
10 A$="ABC"
20 POKE 6,PEEK(131):POKE 7,PEEK(132)
30 VP=PEEK(6)+256*PEEK(7)
50 L=PEEK(VP):A=PEEK(VP+1)+256*PEEK(VP+2
)
60 FOR I=A TO A+L-1
70 PRINT CHR$(PEEK(I));
80 NEXT I
```

Listing 11. So simuliert man VARPTR, hier auf dem Apple

++

Ein Tip, mit dem EDTASM unter Newdos80 schneller und eleganter zu arbeiten:

Ihr kennt alle die Prozedur: Source-file laden, assemblieren, Source-file saven, go to dos, object-code laden., test-run.. Wenn das Programm nicht so tut, wie es soll, beginnt das ganze Spielchen von vorne.

Ich mache es so: Textfile laden, assemblieren, goto dos, exec object-code, wenn ein Problem auftaucht, mit 123 in den Debugger, dann G6FAO, das ist ein Warmstart des Assemblers, wobei der Textbuffer erhalten bleibt. Der Cursor ist allerdings unsichtbar, aber es geht für kurze Edit oder Einfügen von Zeilen auch so sehr gut.

EDTASM belegt den Platz: 5200-76FF

Deshalb muß die ORG-Anweisung oberhalb von HA000 liegen, um auch für den Source-Code noch genügend Platz zu lassen. Nach Fertigstellung des Programmes kann man die ORG ja wieder hingeben, wo man will.

Viel Spaß

W.Zwickel

++

Der Mann mit dem rauchenden Lötkolben meldet:

Für das Clutreffen überlege ich, ob ich ein Selbstbauprojekt zur Sprache bringen soll. Ich denke da an ein Bus-System, wie bei Apple od. IBM-PC.....Da könnte man dann alles, was es an I/O Spielchen gibt anstecken. (Sprachausgabe, Joystick, Eprommer,RS232, PIO, D/A-Wandler, Relaiskarte, ...

Der Bus-Controller wäre auf jeden Fall voll gepuffert und könnte alle I/O Adressen von 1-254 betreiben.

Jede Karte hat für sich die Adressdecodierung, damit das System universell ist und die Karten an beliebigen Stellen eingesteckt werden können.

Zur Größe der Steckkarten:

Als Format stelle ich mir Europa-Größe vor, da man darauf sicher alles unterbringt. Es gibt auch in dieser Größe in diversen Fachzeitschriften jede Menge Bauvorschläge, die man 'abkupfern' kann.

Vielleicht habt Ihr, liebe Freunde einen guten Vorschlag für die Pin-Belegung des Bus-Systems. Es muß auf jeden Fall 8 Datenbits, die unteren 8 Adressbits, den Control-bus sowie die Stromversorgung (+-5V +-12V) beinhalten.

Wenn wir alle 16 Adressbits herausführen, so können wir auch Speichererweiterungen, Eprom-Banks,High-res.Grafik, Floppy-Controller........über das Bus-System realisieren.

Wie denkt Ihr darüber? Lasst es mich wissen. Wer Zeit für ein Briefchen hat, den ersuche ich schon vor dem Clubtreffen mal kurz ein paar Zeilen mir zukommen zu lassen.

Walter Zwickel
Lengfelden 123
A-5101 BERGHEIM

„NANU?"

Konvertierung TRS-80 Level-2-Basic nach MBasic

Das Level-2-Basic der TRS-80-Computer besitzt zum größten Teil alle Befehle, Funktionen und Operatoren von MBasic. Daher ist es meist kein Problem, die TRS-80-Basic-Programme auf anderen Computern unter MBasic laufen zu lassen. Eine Schwierigkeit liegt jedoch im Cursor-Positionierungsbefehl des TRS-80. Alle Cursor-Positionen sind bei ihm von der linken oberen bis zur rechten unteren Ecke von 0 bis 1023 durchnumeriert (beim 16 x 64-Bildschirm). Man positioniert den Curosr nun an eine bestimmte Bildschirmposition durch den Befehl PRINT@ZAHL, wobei Zahl eine der Positionen 0 bis 1023 ist. Bei den meisten anderen Computern wird eine Bildschirmposition aber durch Angabe der Zeile und Spalte erreicht.

Die lästige Umrechnerei der TRS-80-Zahlen (0 bis 1023) in Zeilen und Spaltenangaben erledigt folgendes kleine Unterprogramm:

(In der Variablen ZAHL ist die TRS-80-Position zu übergeben.)

```
ZEILE=INT(ZAHL/64)
SPALTE=ZAHL-64*Zeile
```

ZEILE und SPALTE nehmen die Werte von 0 bis 15 beziehungsweise 0 bis 63 an. Zählen auf Ihrem Computer die Zeilen und Spalten von 1 bis 16 (1 bis 24) beziehungsweise 1 bis 64 (1 bis 80) sind die Variablen ZEILE, SPALTE am Ende des Unterprogramms einfach um Eins zu erhöhen. Die TRS-80-Programme mit dem 16 x 64-Bildschirm laufen so auch auf 24 x 80-Bildschirmen, wobei natürlich nur 16 Zeilen und 64 Spalten benutzt werden.

Da es noch relativ umständlich ist, bei jeder Cursor-Positionierung obiges Unterprogramm aufzurufen, hier noch ein Tip, wie man es sich, eventuell nicht auf allen Computern, noch einfacher machen kann:

Da MBasic selbst keinen Cursor-Positionierungsbefehl besitzt, muß auf jedem Computer eine systemspezifische Positionierungsroutine benutzt werden. Bei manchen Computern ist es möglich, diese Routine in einer benutzerdefinierten MBasic-Funktion (DEF FN...) unterzubringen.

Ein Beispiel dazu: Der Cursor werde zum Beispiel durch PRINT CHR$(27)+CHR$(22)+CHR$(Zeile)+CHR$(SPALTE) positioniert.

Eine entsprechende Funktion, die direkt die TRS-80-Zahlen in Zeilen/Spalten umrechnet und gleichzeitig den Cursor positioniert, sähe dann folgendermaßen aus:

```
DEF FNC$(ZAHL)= CHR$(27)+CHR$(22)+CHR$(INT(Zahl/64))+ CHR$(Zahl-64*INT(Zahl 64))
```

Damit fallen alle Umrechnungen des TRS-80-Formats weg. Statt PRINT@ Zahl im TRS-80-Basic schreibt man jetzt in MBasic: PRINT FNC$(ZAHL)

Ob es möglich ist, eine Funktion dieser Art zu definieren, hängt jedoch von der Art der Cursor-Positionierung des jeweiligen Computers ab.

Und nun viel Spaß beim Übertragen der TRS-80-Programme auf MBasic, ohne lästige Rechnereien!

(J.Voß/bo)

HURRA!

Wieder einmal habe ich es geschafft - wieder einmal kann ich mit dieser letzten Seite ein Info abschließen. Dieses Mal habe ich es wieder mit Freude gemacht, da ich momentan etwas mehr Zeit habe. Jetzt zum Schluß hinaus mußte ich mich zwar nochmal ganz schön beeilen - aber ich glaube, das Info kommt noch rechtzeitig an.

Nochmal Danke bei allen Mitgliedern, die an der Entstehung dieses Infos beteiligt waren.

Ich freue mich schon riesig auf unser Clubtreffen - obwohl das Wochenende ziemlich anstrengend werden wird.

Manche werden vielleicht einige Antworten zu Vorschlägen oder Fragen zum letzten Info vermissen - hier habe ich dieses Mal so gut wie keine Reaktionen von Euch erhalten, und kann aus diesem Grund auch nichts schreiben.

So - nun nur noch Info kopieren lassen, zusammenstellen und heften und dann rein in die Kuverts, Adresse und Briefmarke drauf und ab damit zur Deutschen Bundespost.

Tschau bis bald

Euer Günther Wagner

← Nicht gültig für unser Clubtreffen

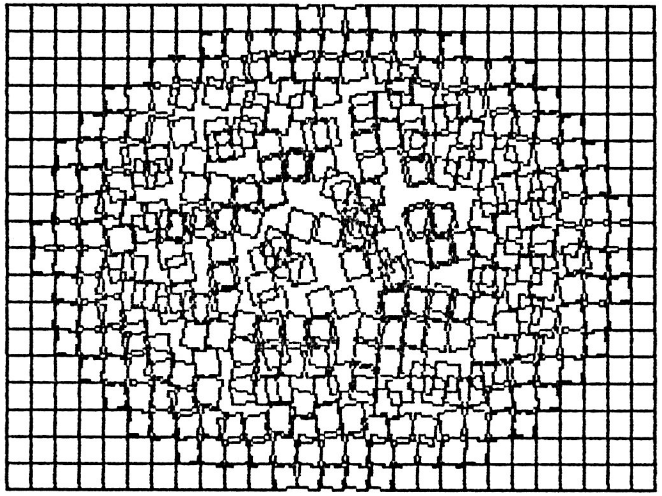
CLUB 80

Clubinfo
der
TANDY – GENIE – und KOMTEK – ANWENDER

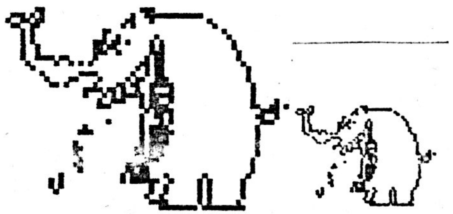
7. AUSGABE

Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubeuern

Tel.: 08035/3361 (18 – 20 Uhr)



UNORDNUNG und GRAZY MAMMUTS
erstellt mit
hochauflösender Grafik.



Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert . . . . .01 - 04
Neues aus dem Clubleben . . . . .05 - 06

Software

Lokale Variable . . . . .07 - 08
Basic, Better and Faster . . . . 09 - 12
Eingaberoutine mit Komfort . . . . . 13 - 16
Strukturierte Programmierung . . . . 17 - 22
Planvoll programieren kann jeder . . 23 - 27
Keine Meinung zur Struktur 28
Einlesen von Programmzeilen 32
Peeks und Pok's 32
Adventure - Ecke . . . . 34 - 36
Tips und Tricks . . . . .37 - 39

Hardware

Warnung (TRS-80 () Genie) 40
ECB - Bus System . . . . 41 - 42
Ein sicheres Plätzchen 43
Super-Tape . . . . 44 - 50
Tests 51

Seite:

Börse

Vorbemerkung 52
Wer hat was ???--Wer sucht was ??? . . 53
Fragen, Fragen, Fragekasten 54

Sonstiges

Aus aktuellem Anlaß . . . . .55 - 58
Computerkäufer sind anders 58
Lebensdauer von Disketten 61

Programmbibliotek

Neue Programme 15

Club -Bücherei

Neue Bücher . . . . .59 - 60
Fundgrube 60

Die letzen Seiten

Clubmitgliederadressen 62
Impressum 63
Schluß 64
Abstimmzettel . . . . am INFO-Ende

Liebe Club-Freunde,

das 7. Info liegt mir im Entwurf vor - ich kann dem Jens dazu nur gratulieren und Ihm an dieser Stelle danken - danken für seine Bereitschaft, einen sehr wesentlichen Bestandteil des CLUB 80 (Gestaltung des Clubinfos und der damit verbundenen Arbeiten) verantwortlich zu übernehmen.

Das 7. Info und die zahlreichen Anrufe vom Jens zeigen sein großes Engagement um den Club. Doch auch er ist auf Euch Mitglieder angewiesen - nur wenn er genügend Beiträge bekommt, kann er auch ein wirklich gutes Clubinfo rausbringen. Auf dem Clubtreffen war es offensichtlich: Da ist soviel Wissen und Know-how da, da haben viele echt tolle Tips und Tricks auf Lager - doch veröffentlichen wollen Sie im Info nichts. Die einen trauen sich nicht, die anderen wollen nicht, manche meinen, es wäre nicht interessant genug.

Ich möchte also nun an a l l e Club-Mitglieder appellieren: Wenn Ihr was habt, so schreibt es auch - sei es auch noch so kurz. Der Jens freut sich bestimmt über jeden Beitrag.

Nun noch kurz zum Clubtreffen. Es hat prima eingeschlagen und findet nun jährlich statt. Ich bin also doch noch 1 Jahr Vorstand - aber nur, weil sich der Jens bereit erklärt hat, den Löwenanteil der Arbeit zu übernehmen (den Vorstand mitzuübernehmen weigerte er sich leider - ein anderer fand sich nicht).

Ich hoffe, daß die neue Satzung von allen Clubmitgliedern anerkannt wird. über 20 Mitglieder waren bei der Neugestaltung dabei und haben sich Gedanken gemacht, wie die neue Satzung auszusehen hat. Einig war man sich auch, den Beitrag auf 50 DM zu erhöhen und 10 DM nachzuzahlen. Diese nochmalige Erhöhung ist unbedingt nötig um den derzeitigen Umfang des Clubinfos beibehalten zu können.
Ich bitte also nun alle Mitglieder darum, den Stimmzettel am Ende dieses Clubinfos auszufüllen (er wurde übrigens so gestaltet, daß er nach üblicher Faltung in ein Fensterkuvert paßt). Bitte denkt daran - ein Info mit diesem Umfang und die Programm- und Bücherbibliothek sind nur dann möglich, wenn die Finanzierung gesichert ist. Diese ist nur dann gesichert, wenn wir den Beitrag auf 50 DM erhöhen. Die Nachzahlung in Höhe von 10 DM ist notwendig, damit wir dieses Jahr nicht wieder mit einem Minus abschneiden. Bitte beachtet auch die Seiten 3-5.

übrigens: Den Stellenwert unseres (derzeit mindestens 64 Seiten umfassenden) Clubinfos können nachstehende Zeilen eindrucksvoll belegen. Peter Spieß stellte folgende Untersuchung an:

Er wollte herausbekommen, was die Zeitschrift "CHIP" an Informationen für den Leser bietet. Dazu hat er die Ausgabe 12/84 in die Einzelteile zerlegt. Was dabei herauskam findet Ihr in der folgenden Tabelle:

Seitenanzahl des Heftes: 360

121	Seiten ganzseitige Werbung
67	Seiten CHIP-Börse und Werbung
141	Seiten Informationen, wobei aber auf diesen Seiten teilweise kleinere Anzeigen abgedruckt sind.
31	Seiten Artrikel und Programme für Commodore VC 64

360	
===	

Das Heft kostet 6.00 DM; macht im Jahr 72 DM. Mit ein wenig Glück findet man auf Jahressicht gesehen einige wenige Seiten die die Computer der Firmen Tandy und EACA betreffen.

Der Club 80 kostet Jährlich 50 DM - es kommen 6 Infos heraus. Der Jahresumfang dürfte 400 Seiten übertreffen !!!

Ich glaube, die Entscheidung auf dem Stimmzettel dürfte nicht schwer fallen. Viel Spaß wünsche ich Euch mit diesem Info,

Euer Günther Wagner

S I N D W I R K R I M I N E L L ?
??

Mit dieser Frage meine ich, ob wir Computer-Anwender kriminell sind ? - Wie ich dazu komme ? - Lest diesen Artikel und Ihr werdet verstehen was ich meine. Am besten fange ich von vorne an.

ES WAR EINMAL ein 'liebes' Club- Mitglied namens Peter Schmidt, 7800 Freiburg - ein Schäflein unter vielen. Doch dieses Schäflein sollte sich schon bald als schwarzes Schaf herausstellen.
Denn - Peter Schmidt, Freiburg, war nur eine Deckadresse - eine Deckadresse für

Hans Peter Schmid
Lenaustraße 2
6906 Leimen 2

Dieser vertreibt u.a. die Programme Newscript, Bugout, DOSPlus, Crashman, Faster, Accell, Edit, etc.

Und eben dieses Schmid'chen-Schleicher' hatte sich beim Hans König gemeldet. Er sei ein neues Club-Mitglied und suche schon lange die Programme Newscript und Bugout und ob er diese bekommen könnte.
Der Hans hat die Programme an den Schmid weitergegeben und dafür ein Schreiben vom Rechtsanwalt bekommen. Er würde Programme vertreiben; ein Streitwert von 12000 DM wurde genannt. Der Hans verweigerte natürlich jede Zahlung.

Was macht da unser 'lieber' Schmid ?

Nun - er stellt Strafanzeige gegen den Hans König und den Frank Smerling (dessen Name stand auf einer beigelegten Anleitung). Beiden wurde Ende April die Wohnung von der Polizei durchsucht - Disketten, Anleitungen und Schriftverkehr wurde mitgenommen!

Nun haben der Hans und der Frank wahrscheinlich ganz schönen Trouble - die Polizei hat die Adressen aller CLubmitglieder - und ich habe einen Teil meines Schriftverkehrs vernichtet. Mit diesem kurzen Artikel wollte ich informieren und warnen. Der Schmid hat seit Dezember 84 nichts mehr von mir erhalten (aber vielleicht ist ja unter uns ein weiteres schwarzes Schaf). Es könnte nicht schaden, wenn jeder mal seinen Schriftverkehr ausmistet!

PS.: In der CHIP vom Mai 85 steht auf Seite 18:

"Es ist nicht verboten, Raubkopien privat zu nutzen. Aber vervielfältigen und weitergeben darf man sie nicht, auch nicht an einen Freund."

[Satzung] des Club 80

(Entwurf des Clubtreffens – endgültige Abstimmung über Clubinfo)

Bitte beiliegenden Stimmzettel am INFO-Ende beachten !!!

1. Mitgliedschaft, Organe und Aufgaben des Clubs

1.1 Der Club 80 ist die Gesamtheit der Mitglieder. Seine Aufgaben sind der Austausch von Erfahrungen, Programmen und Büchern durch Clubnachrichten, Versand und Clubtreffen.

1.2 Mitglied ist, wer auf Antrag aufgenommen wurde, die Clubsatzung anerkannt hat, die Aufnahmegebühr von DM 10,- und einen Jahresbeitrag von DM 50,- entrichtet hat.

1.3 Die Mitgliedschaft endet mit dem Ableben, dem Austritt oder dem Ausschluß. Bezahlte Beiträge können nicht zurückerstattet werden.

1.4 Der Ausschluß erfolgt
- a) wenn nach zweimaliger Mahnung der Beitrag nicht entrichtet wurde
- b) durch Beschluß der Mitglieder mit Dreiviertelmehrheit
- c) bei Verstößen gegen die Clubsatzung.

Mitglieder, die nach b) oder c) ausgeschlossen wurden, können nicht wieder aufgenommen werden.

1.5 Organe des Clubs sind der Vorstand, das jährliche Clubtreffen und die Gesamtheit der Mitglieder.

1.6 Beschlüsse, die die Satzung betreffen, erfordern eine Dreiviertelmehrheit, andere Beschlüsse eine einfache Mehrheit der abgegebenen Stimmen.

1.7 Der Vorstand wird von der Gesamtheit der beim Clubtreffen anwesenden Mitglieder gewählt, nachdem der alte Vorstand entlastet worden ist.

1.8 Der Club wird auf Beschluß mit Dreiviertelmehrheit aufgelöst. Sein evtl. Vermögen wird gemeinnützigen Zwecken zugeführt.

2. Programmtausch

Der Programmtausch erfolgt über die Programmbibliothek des Clubs. Jedes Mitglied verpflichtet sich, nur Programme einzusenden, die frei von Rechten Dritter sind. Eine Überprüfung durch den Club kann nicht erfolgen. Die neuen Programme werden jeweils in den Clubnachrichten mit kurzer Beschreibung vorgestellt – eine Gesamtübersicht der Programmbibliothek erfolgt jeweils zum Jahreswechsel. Nur Clubmitglieder können Programme beziehen. Ein Verkauf von Programmen, die über den Club bezogen wurden, ist strengstens verboten, ebenso die Weitergabe an Dritte. Bei Zuwiderhandlung erfolgt sofortige Kündigung der Mitgliedschaft; strafrechtliche Verfolgung ist möglich.

3. Büchertausch

Computerbücher sind teuer, und oft sind für den Einzelnen nur wenige Seiten interessant. Es wird deshalb im Laufe der Zeit eine Bücherbibliothek angelegt. Jedes Clubmitglied kann jeweils ein Buch für vier Wochen kostenlos ausleihen. Ist das gewünschte Buch zur Zeit der Bestellung ausgeliehen, so wird das Mitglied auf eine Warteliste gesetzt. Das Buch wird auf Kosten des Entleihers versandt, der Besteller sendet das Buch auf seine Kosten zurück. Verlorengegangene Bücher müssen ersetzt werden. Wird die Ausleihfrist überschritten, erfolgt eine Mahnung in Höhe von DM 20,-, für jede weitere Mahnung DM 5,-. Die neuen Bücher werden in den Clubnachrichten veröffentlicht, eine komplette Liste erscheint jeweils zum Jahreswechsel.

4. Hardware und Verbrauchsmaterialien

Bei entsprechendem Interesse werden Sammelbestellungen für Hardware-Artikel und Verbrauchsmaterialien (z. B. Disketten) organisiert. Außerdem sind die Mitglieder aufgefordert, eigene Hardware-Entwicklungen auch dem Club zur Verfügung zu stellen.

5. Clubnachrichten

Diese erscheinen unregelmäßig (etwa alle zwei Monate). Der Inhalt ist ersichtlich aus den übrigen Punkten der Clubsatzung. Falls möglich, wird auch gewerbliche Reklame gegen Gebühr abgedruckt.

6. Mitgliedsbeitrag

Der Club verursacht Kosten, z. B. die Programmarchivierung, der Druck der Clubnachrichten, Porto usw.. Natürlich werden die Ausgaben so gering wie möglich gehalten, aber ein Mitgliedsbeitrag ist nötig. Der Aufnahmebeitrag beträgt DM 10,-, der zum Jahreswechsel fällige Jahresbeitrag DM 50,-. Erfolgt die Aufnahme in den Club nach dem 31. Juli, ist nur der halbe Jahresbeitrag für das erste Mitgliedsjahr zu entrichten.

7. Clubkasse

Aus der Clubkasse werden die laufenden Unkosten bezahlt. Größere Überschüsse werden zum Ankauf weiterer Programme und Bücher verwandt. In den Clubnachrichten erscheint zu gegebener Zeit eine Aufstellung der zu kaufenden Artikel. Ein Rechenschaftsbericht über die Clubkasse erfolgt jeweils zum Jahreswechsel in den Clubnachrichten.
Für die Mitglieder bestehen folgende Zahlungsmöglichkeiten:
- a) Barzahlung
- b) Scheck
- c) Überweisung auf Kto. Nr. 194 712 bei der Sparkasse Rosenheim, BLZ 711 500 00, PSchKto. Nr. 8077-801

8. Austritt

Jedes Mitglied kann jederzeit aus dem Club austreten. Der Austritt sollte schriftlich und unter Angabe von Gründen erfolgen.

9. Änderung der Clubsatzung

Jedes Mitglied kann eine Änderung der Clubsatzung unter Angabe von Gründen beantragen. Dieser Antrag wird in den Clubnachrichten veröffentlicht, und die Mitglieder entscheiden sich für oder gegen die beantragte Satzungsänderung mit einer Dreiviertelmehrheit.

10. Gültigkeit der Satzung

Diese Satzung ist nach ihrer Genehmigung per Abstimmung über das Clubinfo Nr. 7 in Kraft und für alle Mitglieder verbindlich. Frühere Fassungen der Satzung sind danach ungültig. Ihre Gültigkeit bzw. eine geänderte Fassung (je nach Abstimmungsergebnis) wird im Clubinfo Nr. 8 bekanntgegeben.

Protokoll des Clubtreffens vom 23./24. März in Reinheim

Vorsitz: Günther Wagner
Protokoll: Arnulf Sopp

Anwesende: die obigen,	Hans J. König,	Christian Schrewe
Wolfg. Beckhausen,	Josef Konrad,	Gerald Schröder,
Ulrich Böckling,	Jens Neueder,	Frank Smerling,
Gerald Dreyer,	Hartmut Obermann	Holm Voigtländer,
Manfred Held,	Walter Piller,	Alexander Wagner,
Klaus Hermann,	Heinrich Rank,	Jürgen Wucherer,
Dieter Kasper,		Walter Zwickel,

sowie Ehefrauen von Mitgliedern, die jedoch nur aus touristischen Gründen angereist waren und nicht mit abstimmten.

Beginn: 23. 3. 85, 17 Uhr
Ende: 24. 3. 85 gegen Mittag

Verlauf:

<u>1. Beschlußfähigkeit:</u>
Einstimmiger Beschluß, daß der Vorschlag zur Beschlußfähigkeit aus dem Clubinfo Nr. 6 zu befolgen ist: Das Clubtreffen erklärt sich als Organ des Clubs als entscheidungsbefugt, weil keine Einwände gegen den Info-Vorschlag vorliegen.

<u>2. Programmbibliothek:</u>
Hartmut verwaltet die Bibliothek. Sie umfaßt ca. 190 Programme ohne Rechte Dritter. Entsprechend Hartmuts Hardware können Disketten bis 40/ SS/DD eingesandt und abgefordert werden.
Das Punktesystem wird mit einstimmigem Beschluß abgeschafft.
Es ist geplant, Ordner mit Listings herumzuschicken, von denen jedes Mitglied eines abtippen soll, um die Programmbibliothek zu erweitern.

<u>3. Entlastung des Vorstands:</u>
Wahl von Frank und Jürgen als Kassenprüfer. Nach Prüfung wird der Vorstand einstimmig entlastet.

<u>4. Vorstand, Gestaltung des Clubinfos:</u>
Die Tätigkeiten des bisherigen Vorstands werden verteilt: Ansprechadresse für alle den Club betreffenden Fragen und damit Vorstand bleibt Günther. Die Zusammenstellung des Clubinfos übernimmt Jens. Beide einstimmig gewählt. Wer die Kopien anfertigen wird, ist noch nicht geklärt. Mehrere Mitglieder erkundigen sich nach möglichst günstigen Angeboten.

<u>5. Beiträge:</u>
Der Mitgliedsbeitrag beträgt rückwirkend zum 1. 1. 85 DM 50,-. Dieser Betrag gilt für neue Mitglieder. Bisherige Mitglieder zahlen DM 10,- nach. Hierzu erfolgt noch ein endgültiger Beschluß über das Info.

<u>6. Micro 80, Load 80:</u>
Holm übersetzt Programmanleitungen und Artikel aus Micro 80 ins Deutsche. Ein Abonnement von Load 80 auf Clubkosten wird abgelehnt, weil der Nutzen zu gering ist.

<u>7. Satzung:</u>
Die neue Satzung (s. nächste Seiten) wird entworfen und einstimmig verabschiedet. über die Gültigkeit entscheidet eine Abstimmung über das Clubinfo.

<u>8. Hardware:</u>
Walter Z. stellt eine Idee für eine universelle I/O-Karte vor. Mehrere

Clubkameraden stellen sich vor !!

Ich heiße Eckehard Kuhn, bin 21 Jahre alt und Schüler an einem Technischen Gymnasium. In diesem Jahr fange ich eine Ausbildung zum Informationselektroniker an.
Vor etwa zwei Jahren erwarb ich einen TRS-80 M 1 und legte mir bald darauf ein Expansion Interface zu. Zur Zeit suche ich ein günstiges Laufwerk. (Wer hätte eins?) Meinen Computer benutze ich vor allem zum Erstellen von Programmen, (Basic + Assembler) im Bereich der Chemie. Zum Beispiel: elektronische ph-Wert Messung über Computer.

Eckehard Kuhn

DIE NEUEN MITGLIEDER

Beckhausen Wolfgang
Vuerfelser-Kaule 30
5060 Bergisch-Gladbach 1
02204/62781

Clubmitglied seit : 08.03.85
Punktestand = 60
System- und Drivekonfiguration :
Komtek 1 Super/ Epson FX-80/ 2 Laufwerke (je 40 Spuren/ ds/ sd
Schaltbox f r Leitungen bis 10 A
bevorzugtes Betriebssystem : NEWDOS 80

Schaefer Walter
Rathausstr. 4
8160 Miesbach
08025/1631

Clubmitglied seit : 11.03.85
Punktestand = 60
System- und Drivekonfiguration :
Genie I (Mod. 83)/ Star Gemini 10x/ 2 Laufwerke TCS 820/2 FC
bevorzugtes Betriebssystem : G-DOS 2.16

Schneider Manfred
Rheinkasseler Weg 11
5000 Koeln 71
0221/707044

Clubmitglied seit : 09.04.85
Punktestand = 60
System- und Drivekonfiguration :
TRS-80 Model I Level II mit 48 K/ Drucker NEC PC-8023 B-C/ 3 Laufwerke (40 Spuren/ 1 * ss/sd und 2 * ds/dd)/ RB RS-232/ RB HRG1 B
bevorzugtes Betriebssystem : NEWDOS 80 V2

Spiess Peter
Trugenhofenerstr. 27
8859 Rennertshofen
08434/454

Clubmitglied seit : 16.03.85
Punktestand = 45
System- und Drivekonfiguration :
Video Genie II (Mod. 83)/ TRS-80 Model 3 und 4/ Drucker NEC 8023 B-C/ Itoh 8510A und 1550/ HRG1b/ RS-232/ Grafikkarte 512*512/ Modem (Tandy)/ 3 Laufwerke (40 Spuren und 80 Spuren jeweils ds/ dd)
bevorzugtes Betriebssystem : NEWDOS 80 und Multidos

Helmut Paulo

Lokale Variable beim TRS-80

Einer der Vorteile von Pascal ist die Möglichkeit, Variablen in einzelnen Programmteilen lokal verwenden zu können, ohne die Verwendung einer Variablen gleichen Namens in einem anderen Programmteil zu behindern. Der folgende Beitrag zeigt nicht nur einen Weg, so etwas auch in Basic zu realisieren, sondern auch, wie man Programme ohne Variablenverlust verketten kann.

Für einen Neuling kommt zur Einführung das Programmieren mit einem Pascal-Compiler überhaupt nicht in Frage, weil das Verfahren viel zu umständlich ist (editieren, speichern, compilieren, linken usw.), so daß ein Anfänger mit einem solchen System schnell seine Versuche frustiert beenden würde. Für den Anfängerunterricht gibt es also zur Zeit keine Alternative zu Basic. Hinzu kommt, daß man z. B. beim TRS-80 in Basic zahlreiche Vorteile hat, die in Pascal nicht gegeben sind (doppelte Genauigkeit, Grafik). Schließlich gibt es ja auch noch SBasic, womit ein strukturiert geschriebenes Programm erstellt werden kann. Da dieses Programm durch den Precompiler von SBasic in ein ganz gewöhnliches Basic-Programm verwandelt wird, ist gezeigt, daß man auch in Basic

handelt sich lediglich um Fragen der Formulierung. Eine wichtige Eigenschaft von Pascal hat allerdings Basic nicht: die Verwendung von lokalen und globalen Variablen. Der folgende Beitrag zeigt, wie man beim TRS-80 auch diesen Mangel beseitigen kann.

```
FC00  21FF FB22 4940 C32D 40CD 7F0A 7D32 75FE !.."I§.-§....2..
FC10  FE1E CA7B FDFE 28CA 7BFD FE0A CA61 FCFE ......(.........
FC20  14CA 35FC 110C 0021 E9FD 3C47 1910 FD22 ..5....!..<G..."
FC30  6FFE C32A FD2A A440 E523 2323 237E B7C2 ...*.*.§.####...
FC40  3CFC 23D1 7D12 137C 1211 FAFF CD2C 1B23 <.#...........,.#
FC50  2322 F940 22FB 4022 FD40 2A49 4022 B140 #".§".§".§*I§".§
FC60  C93A 71FE EE01 3271 FECA FAFC 2AB1 4022 .:....2.....*.§"
FC70  01FE 2AD6 4022 05FE 2AA0 4022 03FE 2114 ..*.§"..*.§"..!.
FC80  FDCD A728 3E02 32AF 40CD B31B 23CD 6C0E ...(>.2.§...#...
FC90  2A21 4111 00FC CDC7 0B22 F5FD 2322 07FE *!A......"..#"..
FCA0  2209 FE22 0BFE 1100 FEDD 210D FE3A F4FD ".."......!..:..
FCB0  B7CA E5FC 472B DD75 00DD 7401 DD75 04DD ....G+..........
FCC0  7405 19DD 7502 DD74 0319 DD75 06DD 7407 ................
FCD0  DD75 08DD 7409 DD75 0ADD 740B D511 0C00 ................
FCE0  DD19 D110 D02B 22F5 FD22 B140 11F5 FD21 .....+".."§...!
FCF0  6DFE CDF6 FCC9 7323 72C9 21F9 4011 FBFD .......#..!.§...

FD00  0106 00ED B02A D640 22F9 FD2A A040 22F7 .....*.§"..*.§".
FD10  FDC3 EDFD 4752 4F45 5353 4520 564F 4E20 ....GROESSE.VON.
FD20  4245 5245 4943 4820 3100 2A6D FE11 0400 BEREICH.1.*.....
FD30  19ED 5BD6 40CD F6FC 23EB 21F9 4001 0600 ..Ä.§...#.!.§...
FD40  EDB0 ED5B 6FFE 216D FECD F6FC 2A6F FE7E ...Ä..!.....*...
FD50  32B1 4023 7E32 B240 237E 32A0 4023 7E32 2.§#.2.§#.2.§#.2
FD60  A140 237E 32D6 4023 7E32 D740 2311 F940 .§#.2.§#.2.§#..§
FD70  0106 00ED B0C3 EDFD 5A24 003E 0132 74FE .........Z$.>.2..
FD80  2178 FDCD 0D26 131A 6F13 1A67 1182 FED5 !....&..........
FD90  1176 FE7E FE20 CAA5 FDFE 22CA DAFD FE2C ..........."....,
FDA0  CAA9 FD12 1323 C393 FDAF 12D1 23E5 D521 .....#.......#..!
FDB0  76FE CD0D 263A 75FE FE28 CAE1 FDD5 E1D1 ....&:...(......
FDC0  2B2B 2B7E 1223 2323 1306 004F EDB0 E13A +++..###...O...:
FDD0  74FE B7CA EDFD D5C3 90FD AF32 74FE C3A9 ............2....
FDE0  FDE1 7E06 004F 23ED B0EB C3CE FD3E 0232 .....O#......>.2
FDF0  AF40 C900 0300 0000 0000 00B2 6D00 0000 .§..............
```

Bild 1. Maschinenprogramm zum Umschalten zwischen bis zu acht Variablen-Bereichen für einen 48-KByte-TRS-80

Globale und lokale Variable

Benötigt wird das Maschinenprogramm (*Bild 1*), das als CMD-File auf Diskette zu speichern ist (START=TRA=FC00, END=FDFF). Nach Aufruf dieses Programms vom DOS aus geht man ins Basic. Das Basic-Programm (*Bild 2*) demonstriert, wie man lokale und auch globale Variable bekommt. Hierzu muß man eine USR-Funktion benutzen (DEFUSR =&HFC09), die mit unterschiedlichen Argumenten aufzurufen ist. Zunächst werden mit Hilfe von Zeile 40 die beiden Hauptbereiche 0 und 1 für die Variablen initialisiert, die CLEAR-Statements reservieren den gewünschten String-Platz. Danach befindet man sich im Bereich 0, in dem man Programme laden, ändern und speichern kann. Mit Zeile 70 wird der Stack initialisiert, und nun kann man jeweils durch Aufruf von Z=USR(n) in den jeweils gewünschten Variablenbereich Nr. n umschalten. Normalerweise stehen die beiden Hauptbereiche und drei weitere Bereiche mit je 1 KByte zur Verfügung. Die Anzahl dieser zusätzlichen Bereiche steht in Adresse FDF4 und kann bis acht erhöht werden. In allen Bereichen existieren die Variablen völlig unabhängig voneinander, so daß man beim Aufruf von Unterprogrammen unabhängig von den Variablennamen ist! Zum Beispiel stört die Verwendung der Variablen I im Bereich 0 nicht die Verwendung von I im Unterprogramm in Zeile 280 im Bereich 1. Der Vorteil dieses Verfahrens ist, daß man nun ein Programm wirklich modular aufbauen kann, ohne an bestimmte Variablennamen gebunden zu sein.

Die Verwendung von lokalen Variablen z. B. in Unterprogrammen hätte wenig Wert, wenn es nicht möglich wäre, Variablenwerte von einem zum anderen Bereich zu übergeben. Dies geschieht mit dem Statement Z$="........":Z=USR(30). Die in Z$ aufgeführten Variablen (Zeile 140) werden zur Übergabe bereitgestellt; nach Umschalten auf einen anderen Bereich werden sie mit Z$="........":Z=USR(40) übernommen (Zeile 240). Hierbei müssen jedoch keineswegs die Variablennamen übereinstimmen, es kommt lediglich auf die Reihenfolge an, die allerdings typenentsprechend sein muß. Wenn man z. B. das Demonstrationsprogramm mehrmals laufen lassen will, so muß man nach dem ersten Durchgang nicht von vorne beginnen, sondern mit RUN 70 bei Zeile 70. Dieses Statement ist bei jedem neuen Start erforderlich, damit der Basic-Stack in den richtigen Bereich (und zwar Bereich 1) verlagert wird. Bekommt man aus irgendeinem Grund eine OM-Meldung, so startet man neu mit NEW und PRINT USR(20). Hierdurch wird das gelöschte Programm reaktiviert, und alle Variablenpointer werden in den Start-Zustand gebracht.

```
10    DEFUSR=&HFC09 :
20    CLS
30    'INITIALISIERUNG VON BEREICH 0 UND BEREICH 1
40    CLEAR800:Z=USR(10):CLEAR500:Z=USR(10)
50    '
60    'INITIALISIERUNG DES BASIC-STACKS
70    Z=USR(1):CLEAR:Z=USR(0)
80    ' ***********************
90    CLS
100   A$="TESTVARIABLE"+""
110   Z=USR(1):X$="STRING IN BEREICH 1"+"":Z=USR(0)
120   FOR I=1 TO 3
130   INPUT"A,B";A,B
140   Z$="A,B,A$":Z=USR(30) :' VARIABLENUEBERGABE AN BEREICH 1
150   GOSUB220
160   Z$="C":Z=USR(40) :'     VARAIABLENUEBERNAHME VON BEREICH 1
170   PRINTA,B,C
180   PRINTA$
190   NEXT
200   END
210   ' **********************
220   ' PYTHAGORAS
230   Z=USR(1) :'               UMSCHALTEN AUF BEREICH 1
240   Z$="U,V,P$":Z=USR(40) :' VARIABLENUEBERNAHME VON BEREICH 0
250   W=U*U+V*V
260   Z$="W":Z=USR(30) :'       VARIABLENUEBERGABE AN BEREICH 0
270   PRINTX$,P$
280   FOR I=1 TO 5: PRINTI;:NEXT:PRINT
290   Z=USR(0) :'               UMSCHALTEN AUF BEREICH 0
300   RETURN
```

Bild 2. Demonstrationsprogramm für die Verwendung lokaler und globaler Variablen in zwei Bereichen

Programm-Verkettung

Eine ganz andere Möglichkeit, ein Programm modular aufzubauen, ist die Programm-Verkettung. Normalerweise werden bei einem neuen RUN alle Variablen gelöscht. Dies kann man mit dem in *Bild 3* gezeigten System vermeiden. PROGR1 dient zur Initialisierung und wird nur einmal benötigt. Danach kann man beliebig viele Teilprogramme aufrufen. Mit dem ersten Statement Z=USR(1) schaltet man auf einen geschützten Variablenbereich, mit Z=USR(2) schaltet man zurück auf den Normalbereich, in dem man Programme laden, ändern und speichern kann, ohne daß die Variablen von Bereich 1 verlorengehen. So kann man auch bei geringer Speicherkapazität praktisch beliebig lange Programme laufen lassen.

```
10    ' PROGR1
20    ' ZUR INITIALISIERUNG
30    CLS
40    INPUT"RAM-GROESSE (16 - 32 -48)";R
50    IF R=48 THEN H=254
60    IF R=32 THEN H=190
70    IF R=16 THEN H=126
80    L=255
90    POKE 16561,L:POKE 16562,H: POKE 16457,L: POKE 16458,H: CLEAR 50
100   H=PEEK(16562): D=65536: IF H=126 THEN D=0
110   M=(H+1)*256-D
120   DEFUSR=M
130   FOR I=0 TO 99
140     READ A: IF A=255 THEN A=H+1
150     POKE M+I,A
160   NEXT
170   DATA 205,127,10,124,183,32,11,125
180   DATA 254,1,40,34,254,2,40,55
190   DATA 24,0,237,91,249,64,25,34
200   DATA 106,255,34,108,255,34,110,255
210   DATA 42,214,64,34,112,255,33,249
220   DATA 64,17,100,255,24,48,33,249
230   DATA 64,17,100,255,1,6,0,237
240   DATA 176,42,112,255,34,214,64,33
250   DATA 106,255,17,249,64,24,23,33
260   DATA 249,64,17,106,255,1,6,0
270   DATA 237,176,42,214,64,34,112,255
280   DATA 33,100,255,17,249,64,1,6
290   DATA 0,237,176,201
300   INPUT"MAXIMALE PROGRAMMLAENGE IN BYTES ";L
310   Z=USR(L)
320   RUN "PROGR2"
```

```
10    ' PROGR2
20    Z=USR(1)
30    DEFINT N
40    DIM A(50)
50    A(33)=100: N=50
60    C$="TEST"+""
70    D$=C$+"-PROGRAMM"
80    PRINT A(33),N
90    PRINTD$,C$
100   Z=USR(2)
110   RUN"PROGR3"
```

```
10    ' PROGR3
20    Z=USR(1)
30    DEFINT N
40    X=17.4
50    Z$=D$+"PROGRAMM3"
60    A(34)=3.4
70    PRINTA(33),N
80    PRINTC$,D$
90    PRINTX,Z$
100   Z=USR(2)
```

Bild 3. Beispiel für die Verkettung von drei Programmen, die einzeln auf Diskette gespeichert sind

BASIC-ÜBERLAGERUNGEN

VARIABLENAUSTAUSCH ZWISCHEN PROGRAMMEN

Immer wenn Sie das RUN- oder LOAD-Kommando benutzen, werden die Inhalte aller benutzten Variablen gelöscht, so daß das Programm mit einem bereinigtem Speicher beginnt. Aber es gibt viele Situationen, bei denen man die Variablenwerte von einem Programm in ein anderes übernehmen will.

Wenn Sie Varibalen zwischen Programmen austauschen können, ist es möglich, daß Sie ein Programm in kleinere Teilprogramme unterteilen. Mit kleineren Programmen haben Sie mehr Speicher zur Verfügung. Ein Programm könnte z.B. die Daten von Diskette laden oder die Eingabe über die Tastatur regeln, ein zweites Programm arbeitet mit den Daten und ein drittes Programm kümmert sich um den Ausdruck.

Bevor Sie die Hilfsroutinen für den Variablenaustausch benutzen können, müssen Sie wissen, daß die Variablen gleich nach dem BASIC-Programmtext im Speicher gespeichert werden. Als Beispiel wollen wir annehmen, daß Sie folgendes Programm geschrieben haben:

```
10  X%=1
20  A%=2
30  S$=STRING$(5,"X")
```

Wenn Sie das Programm starten wird der Inhalt von X% gleich nach der letzten Adresse des BASIC-Programmtextes abgespeichert. Der Inhalt von A% wird nach dem Inhalt von X% gespeichert, gefolgt von einem Zeiger, welcher die Länge und Lage des Inhalts von S$ anzeigt. Die fünf 'X' von S$ werden direkt unterhalb der höchsten Speicheradresse (Festlegung durch MEMORY SIZE) abgespeichert. Definieren Sie in einem Programm eins oder mehrere Felder, so werden diese unmittelbar nach den einfachen Variablen gespeichert.

Der Speicherbereich, der alle Variablennamen, Typencodes, Dimensionen, Zahlenwerte und String-Zeiger speichert, wird Variablenliste genannt. Da die Variablenliste unmittelbar nach dem Programmtext beginnt, ist der Beginn der Variablenliste abhängig von der Länge des geladenen Programms. Zur Übergabe von Variablen setzen wir uns über dieses Merkmal von BASIC hinweg und entscheiden uns für eine bestimmte Stelle an der die Variablenliste beginnen soll. Die von uns gewählte Stelle wird gleich nach der Endadresse des längsten benutzten Programms beginnen.

So findet man die erste verfügbare Adresse nach der Endadresse des längsten Programms:

1. Laden Sie Ihr längstes Programm und beantworten Sie die Frage 'HOW MANY FILES?' genauso, wie Sie es später in der praktischen Anwendung tun.

2. Geben Sie folgende Anweisungen ein:

   ```
   CLEAR
   PRINT CVI(CHR$(PEEK(&H40F9))+CHR$(PEEK(&H40FA)))
   ```

3. Addieren Sie 17 zu der angezeigten Zahl. Das Ergebnis ist die niedrigste Adresse, die Sie für den Beginn Ihrer Variablenliste verwenden sollten, falls Sie Variablen von einem Programm zu einem anderen übergeben wollen. In der Praxis wird man gewöhnlich noch einen gewissen Spielraum einräumen; Sie brauchen diesen z.B. wenn durch Änderungen das Programm länger wird. Addieren Sie also nochmals einen Wert hinzu, z.B. 300.

Wie schaffen wir es nun, daß die Variablen ab der von uns gewählten festen Adresse abgespeichert werden? Im ersten Programm das wir starten wird als eine der ersten Anweisungen ein 'GOSUB 52000' ausgeführt. Dieses GOSUB muß auf alle Fälle vor der Benutzung irgendeiner Variablen erfolgen. Die Hilfsroutine 52000 ändert 3 Zeiger im BASIC. Diese bestimmen Anfang und Ende der benutzten Variablen:

```
52000  A$="": FOR A%=1 TO 3: A$=A$+MKI$(30000): NEXT: AN$=
       "XXXXXX": POKE VARPTR(AN$)+1,&HF9: POKE VARPTR (AN$)
       +2,&H40: LSET AN$=A$: A$="": RETURN
```

Sie müssen anstelle der '30000' die Adresse schreiben, bei der Ihre Variablenliste beginnen soll.

BEACHTE: Die Hilfsroutine 52000 benutzt eine interessante Methode für das POKEN der neuen Zeiger in die 6 Bytes beginnend bei 40F9. Zunächst wird der String A$ erstellt, der die 6 Bytes enthält, welche gepoked werden sollen. Dann wird VARPTR von AN$ so geändert, daß AN$ auf die Adresse 40F9 zeigt. Schließlich führen wir ein LSET von A$ in AN$ aus. Das LSET-Kommando ergibt ein sofortiges POKEN der 6 Bytes. Hätten wir versucht, die 6 Bytes mit einzelnen POKE-Anweisungen zu poken, so hätte sich im BASIC ein Fehler ergeben, da der erste 2-Byte-Zeiger nach der ersten Anweisung nur 'zur Hälfte' gepoked gewesen wäre.

Das abschließende A$="" in der Hilfsroutine 52000 bewirkt, daß A$ die erste initialisierte Varibale wird. Die 'Variablen-Übergabe'-Hilfsroutine und die 'Variablen-Empfänger'-Hilfsroutine erwarten, daß A$ die erste Variable der Variablenliste ist.

Die Hilfsroutine 52100 ist die 'Variablen-Übergabe'-Hilfsroutine. Wenn Sie Variablen von einem Programm an ein anderes Programm übergeben wollen, so müssen Sie ein 'GOSUB 52100' ausführen. Anschließend starten Sie das neue Programm mit RUN. Die Hilfsroutine 52100 lädt A$ mit allen Zeigern die BASIC laufend unterhält. Unter anderem beinhalten die 104 Bytes, die in A$ geladen werden, die Startadressen der einfachen Variablen, Anfang und Ende beliebiger Felder welche aktiv sind, den gegenwärtigen Zustand im String-Speicher und die Typen-Vereinbarungen (DEFSTR, DEFINT, DEFSNG oder DEFDBL) die aktiv sein können.

```
52100 AN$="": POKE VARPTR(AN$),104: POKE VARPTR(AN$)+1,&HB3:
      POKE VARPTR(AN$)+2,&H40: A$=STRING$(104,0): LSET A$=
      AN$: RETURN
```

Die letzte Anforderung für die Technik der Variablen-Übergabe besteht darin, daß das Programm die Variablen wieder erreichen muß. Im neuen Programm sollte daher die erste Anweisung ein 'GOSUB 52200' sein. Die Programmzeile, welche die Hilfsroutine 52200 aufruft, darf keine weiteren Ausdrücke enthalten. Die Hilfsroutine 52200 ist die 'Variablen-Empfänger'-Hilfsroutine. Sie muß die feste Adresse kenne, die Sie als Beginn des Variablen-Speichers festgelegt haben. Dies und die Kenntnis darüber, daß A$ die erste definierte Variable im vorhergehenden Programm war, erlaubt der Routine die Rekonstruktion einer vorläufigen Variable A$ zur Wiedererlangung der 104 Bytes. In diesen 104 Bytes sind alle Zeiger enthalten, die im vorhergehenden Programm abgespeichert worden sind. Schließlich zeigt AN$ auf den Bereich der BASIC-Verwaltung und 'poked' sogleich die 104 Bytes mit einem LSET-Befehl zurück.

```
52200 A$="": FOR A%=0 TO 2: POKE VARPTR(A$)+A%, PEEK(30000
      +A%+3): NEXT: AN$="": POKE VARPTR(AN$),104: POKE
      VARPTR(AN$)+1,&HB3: POKE VARPTR(AN$)+2.&H40: LSET
      AN$=A$: RETURN
```

Sie müssen die '30000' in der Hilfsroutine 52200 in die Adresse abändern, die Sie als Beginn der Variablen-Liste festgelegt haben.

Zur Demonstration der Arbeitsweise der Variablen-Übergabe-Technik können Sie die zwei folgenden Programme verwenden. Das Programm VARPASS/DEM initialisiert die Variablenliste an der Speicherstelle 30000. Anschließend erzeugt das Programm verschiedene Variablen und zeigt diese an.

Schließlich ruft das Programm die 'Variablen-Übergabe'-Hilfsroutine auf und startet dann mit RUN das Programm VARPASS/RCV, welches zunächst die von VARPASS/DEM erzeugten Variablen wiederherstellt. Dies geschieht durch Aufruf der Hilfsroutine 52200. In Zeile 2 wird A$ in einen Null-String zurückgesetzt, da die 104 Bytes, welche wir für die Übergabe der BASIC-Zeiger benutzten, nicht länger gebraucht werden. Schließlich zeigt das Programm die wiederhergestellten Variablen an.

Sie müssen sich bewußt sein, daß das Programm VARPASS/RCV erst gestartet werden kann, wenn durch das Programm VARPASS/DEM die Hilfsroutinen 52000 und 52100 angesprungen worden sind.

```
    0  'VARPASS/DEM
    1  CLEAR 150
    2  GOSUB 52000

   20  C$="CAT"+"": D$="DOG"+""
   30  DATA 1,2,3,4,5,6,7,8,9,10
   31  FOR X=1 TO 10: READ A%(X): NEXT
   40  A!=123: A#=456

  100  CLS
  110  PRINT "PROGRAMM 1 - VARIABLEN SIND:"
  120  PRINT "C$="; C$; TAB(20); "D$="; D$
  130  PRINT "A%()=";: FOR X=1 TO 10: PRINT A%(X);: NEXT: PRINT
  140  PRINT "A!="; A!; TAB(20); "A#="; A#
  200  GOSUB 52100: RUN "VARPASS/RCV"

52000  A$="": FOR A%=1 TO 3: A$=A$+MKI$(30000): NEXT: AN$=
       "XXXXXX": POKE VARPTR(AN$)+1,&HF9: POKE VARPTR(AN$)+2,
       &H40: LSET AN$=A$: A$="": RETURN
52100  AN$="": POKE VARPTR(AN$),104: POKE VARPTR(AN$)+1,&HB3:
       POKE VARPTR(AN$)+2,&H40: A$=STRING$(104,0): LSET A$=
       AN$: RETURN
```

```
    0  'VARPASS/RCV
    1  GOSUB 52200
    2  A$=""

  100  CLS
  110  PRINT "PROGRAMM 2 - VARIABLEN SIND:"
  120  PRINT "C$="; C$; TAB(20); "D$="; D$
  130  PRINT "A%()=";: FOR X=1 TO 10: PRINT A%(X);: NEXT: PRINT
  140  PRINT "A!="; A!; TAB(20); "A#="; A#
  200  END

52200  A$="": FOR A%=0 TO 2: POKE VARPTR(A$)+A%,PEEK(30000+A%
       +3): NEXT: AN$="": POKE VARPTR(AN$),104: POKE VARPTR
       (AN$)+1,&HB3: POKE VARPTR(AN$)+2,&H40: LSET AN$=A$:
       RETURN
```

Hartmut Bulling

TRS-80: Eingaberoutine mit Komfort

Die Funktion INKEY$ beim TRS-80 und beim Video-Genie eignet sich nur bedingt zur Eingabe von Zeichenketten von der Tastatur aus. Bei vielen Anwendungen ist der Basic-Interpreter für diesen Zweck zu langsam.

Das Programm KBREAD/SOU (*Bild 1*) behebt diese Mängel. Im einzelnen hat es folgende Eigenschaften:

- von Basic aus definierbare Eingabefeldlänge,
- alle Sonderzeichen erlaubt (inkl. „Komma"),
- Umlaut „ä" von der Tastatur erreichbar,
- akustische Eingabekontrolle (Frequenz und Tonlänge einstellbar),
- Sonderfunktionen (Clear, Break, Shift-Rechtspfeil, Ctrl) ausgeblendet,
- direkte Übernahme der Eingabe in Basic-String.

Das Programm arbeitet in jedem Speicherbereich und kann daher in allen Speicherkonfigurationen verwendet werden. Einzige Voraussetzung ist ein Level-II-ROM, da auf einige ROM-Routinen zurückgegriffen wird.

KBREAD/SOU ist eine modifizierte Version des Tastaturtreibers im ROM (Adresse 05D9H...0673H). Alle gegenüber dem Original-ROM ausgelassenen Zeilen wurden für weitergehende Änderungen bzw. für die Reaktivierung der ausgeblendeten Tasten als Kommentarzeilen in das Listing übernommen. Vorsicht: Die Taste Shift-Rechtspfeil (ASC 19H) sollte desaktiviert bleiben! Folgende ROM-Routinen wurden verwendet:

- Adresse 0033H: zeigt den Inhalt des A-Registers an. Steuerzeichen sind erlaubt.
- Adresse 0049H: Rücksprung, sobald eine Taste gedrückt wurde. Der ASCII-Wert der gedrückten Taste steht im A-Register. Benutzt A-, Status- und DE-Register.
- Adresse 01C9H: Löscht den Bildschirm, wählt 64 Zeichen/Zeile und setzt den Cursor an den linken, oberen Bildrand.
- Adresse 0A7FH: Bringt die von Basic übergebene Variable in das HL-Register.
- Adresse 0A9AH: Übergibt den Wert des HL-Registers an Basic.

Mit Ausnahme des zweiten Unterprogramms (Adresse 0049H) benutzen die Routinen sämtliche Register.

Von Basic aus leicht zu benutzen

Das Beispielprogramm KBREAD/BAS (*Bild 2*) liest mit Hilfe der neuen Funktion einen Adreßsatz und druckt ihn anschließend.

Das Maschinenprogramm muß vorher in einen geschützten Speicherbereich geladen worden sein. Sollte Ihnen kein Assembler zur Verfügung stehen, können Sie das Programm mit Hilfe von KBREAD/DAT (*Bild 3*) in den Speicher bringen.

Der Aufruf erfolgt mit dem USR-Befehl (Bild 2, Zeilen 1020 und 10030). Die im USR-Aufruf übergebene Variable „L" bestimmt die maximale Länge der zu lesenden Zeichenkette. Bei Rückkehr ins Basic enthält die Variable A$ den gelesenen String.

Wenn Sie weniger Speicher haben

Die vorgestellten Programme setzen 48 KByte RAM voraus. Sollten Sie über weniger Speicherplatz verfügen, so brauchen Sie lediglich den ORG-Befehl (Zeile 300 in KBREAD/SOU) entsprechend Ihrer Konfiguration zu ändern. Wollen Sie das Programm per POKE-Befehl in den Speicher bringen, müssen Sie in Zeile 500 von Bild 3 die benötigten Adressen eintragen. Weiterhin ändern Sie den Wert 255 bis inkl. Zeile 1120 jeweils in 127 für die 16-KByte-Version bzw. in 191 für die 32-KByte-Version.

Die Zahlenfolge 234,254 in Zeile 1030 ändern Sie in 234,126 bzw. in 234,190. HIMEM bzw. MEMSIZE setzen Sie jeweils 296 Byte unter das Ende Ihres Speichers.

Programmänderungen leicht durchzuführen

Tonfrequenz und Tondauer können Sie leicht verändern. Sollten Sie im Besitz eines Assemblers sein, ändern Sie in KBREAD/SOU die Zeilen 2120 und 2130 entsprechend Ihren Vorstellungen. Im Programm KBREAD/DAT ist der Wert 4 in Zeile 1130 ein Maß für die Tondauer, der Wert 120 in den Zeilen 1130 und 1140 ein Maß für die Tonfrequenz.

Den Namen der Variablen, in die die Zeichenkette gelesen wird, finden Sie in KBREAD/SOU in Zeile 2300, in KBREAD/DAT in Zeile 1140 (...,65,36). Das Ein- bzw. Ausblenden bestimmter Tasten setzt einen Assembler voraus. Vergleichen Sie hierzu die Zeilen 580...800, 900...910, 1070...1110 sowie 1180...1190 im Assemblerlisting. Die Taste Shift-Rechtspfeil sollte, wie eingangs bereits erwähnt, ausgeblendet bleiben. Für das Verkleinern bzw. Vergrößern des Puffers (voreingestellt sind 64 Zeichen) benötigen Sie ebenfalls einen Assembler. Wählen Sie hierzu in Zeile 2330 von KBREAD/SOU die neue Puffergröße und ändern Sie Zeile 300 entsprechend. Vergessen Sie nicht, in KBREAD/BAS in Zeile 1020 die Einsprungadresse zu korrigieren.

Literatur

[1] Farvour, James: Microsoft Basic decoded & other Mysteries. ISBN 0-936200-01-4.
[2] Heidenreich, Ulrich: VARPTR umgekehrt. mc 1983, Heft 10, S. 53.

```
100 '* * *   K B R E A D  /  D A T    48 KByte Version    * * *
110 '
120 'Sollte Ihnen kein Assembler zur Verfügung stehen, so können
130 'Sie KBREAD/CIM mit Hilfe dieses Programms in den Speicher laden.
140 '
150 'HIMEM muß mindestens 0FED7H = 65239 betragen !
160 '
170 '
500 FOR I = -296 TO -1
510 '
520 '16 KByte:  FOR I =  32472 TO  32767
530 '32 KByte:  FOR I = -16680 TO -16385
540 '
550 READ D : POKE I,D
560 NEXT I
570 END
580 '
590 '
1000 DATA 245,197,213,229,205,127,10,69,33,192,255,229,62,14
1010 DATA 205,51,0,72,205,73,0,254,10,32,2,62,123,254,25,40,243
1020 DATA 254,31,40,239,254,1,40,235,254,32,48,25,254,13,202
1030 DATA 106,255,17,234,254,213,254,8,40,39,254,24,40,30,254,9
1040 DATA 40,51,254,10,192,209,119,120,183,40,201,126,35,205,51
1050 DATA 0,205,162,255,5,24,190,205,55,255,43,126,35,120,185
1060 DATA 32,246,201,120,185,200,43,126,254,10,35,200,43,62,8
1070 DATA 205,51,0,205,162,255,4,201,205,72,3,230,7,47,60,198
1080 DATA 8,95,120,183,200,62,32,119,35,213,205,51,0,209,5,29
1090 DATA 204,162,255,200,24,236,55,245,62,13,119,205,51,0,205
1100 DATA 162,255,62,15,205,51,0,121,144,71,241,225,33,191,255
1110 DATA 112,205,141,255,225,209,193,241,205,154,10,201,33,188
1120 DATA 255,205,13,38,213,58,191,255,18,17,192,255,227,35,115
1130 DATA 35,114,225,201,245,197,14,4,6,120,62,1,211,255,16,254
1140 DATA 6,120,62,2,211,255,16,254,13,32,237,193,241,201,65,36
1150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,0,0,0,0,0,0,0
```

Bild 3. Wer keinen Assembler hat, kann das Programm von Bild 1 mit dieser Laderoutine abspeichern

```
100 '* * *   K B R E A D  /  B A S    48 KByte Version    * * *
110 '
120 'Das Programm KBREAD/BAS demonstriert die Zusammenarbeit
130 'zwischen BASIC und dem Maschinenprogramm KBREAD/CIM.
140 '(KBREAD/CIM ist die assemblierte Version von KBREAD/SOU).
150 '
160 'HIMEM muß mindestens 0FED7H = 65239 betragen !
170 '
180 '
190 '
1000 DEFINT A-Z
1010 CLEAR 500
1020 DEFUSR0 = -296     ' = 0FED8H bzw. POKE 16526,216 : POKE 16527,254
1030 '
1040 '16 KByte:  DEFUSR0 =  32472 bzw. POKE 16256,216 : POKE 16527,126
1050 '32 KByte:  DEFUSR0 = -16680 bzw. POKE 16256,216 : POKE 16527,190
1060 '
1070 '
1080 CLS
1090 PRINT§140,"E i n g a b e   A d r e ß s a t z"
1100 '
1110 X= 0 : Y= 5 : S$="Vorname:" : GOSUB 8000    'Aufbau der Maske
1120 X=36 : Y= 5 : S$="Name:"    : GOSUB 8000    '   "    "    "
1130 X= 0 : Y= 8 : S$="Straße:"  : GOSUB 8000    '   "    "    "
1140 X= 0 : Y=10 : S$="Plz:"     : GOSUB 8000    '   "    "    "
1150 X=15 : Y=10 : S$="Ort:"     : GOSUB 8000    'Aufbau der Maske
1160 '
1170 '
2000 'Eingabe der Daten
2010 '
2020 X= 9 : Y= 5 : L=20 : GOSUB 10000 : VO$=A$
2030 X=42 : Y= 5 : L=20 : GOSUB 10000 : NA$=A$
2040 X= 8 : Y= 8 : L=30 : GOSUB 10000 : ST$=A$
2050 X= 5 : Y=10 : L= 4 : GOSUB 10000 : PL$=A$
2060 X=20 : Y=10 : L=32 : GOSUB 10000 : O$=A$
2070 X= 0 : Y=14 : S$="Eingaben Korrekt (J/N) ?" : GOSUB 8000
2080 X=25 : Y=14 : L=1 : GOSUB 10000 : AN$=A$
2090 '
2100 IF AN$ <> "J" AND AN$ <> "N" THEN 2080
2110 IF AN$="N" THEN 1080
2120 '
2130 '
3000 CLS
3010 PRINT§129,"Eingegebene Daten:"
3020 PRINT
3030 PRINT"Vorname: ";VO$
3040 PRINT"Name    : ";NA$
3050 PRINT"Straße : ";ST$
3060 PRINT"Plz     : ";PL$
3070 PRINT"Ort     : ";O$
3080 END
3090 '
3100 '
8000 'Berechne Bildschirmposition und drucke Feldnamen
8010 PO=X+64*Y
8020 PRINT §PO,S$
8030 RETURN
8040 '
8050 '
10000 'Hole Daten mit Hilfe von KBREAD/CIM
10010 PO=X+64*Y          'Bildschirmposition des Eingabefelds
10020 PRINT §PO,"";      'Positioniere auf Eingabefeld
10030 X=USR0(L)          'Hole String mit maximaler Länge L
10040 A$=A$              'Fixiere den Wert von A$
10050 RETURN             'Zurück in's Hauptprogramm
```

Bild 2. So kann der neue Tastaturtreiber in ein Basic-Programm eingebunden werden

> ***„Der Kopf ist jener Teil unseres Körpers, der uns am häufigsten im Wege steht."***
>
> Gabriel Laub, tschechischer Satiriker (geboren 1928)

```
            00010 ;*  *  *  *      K B R E A D   /   S O U       *  *  *  *
            00020 ;*                                                        *
            00030 ;*                                                        *
            00040 ;* Autor: H. Bulling                   Datum: 22.01.84    *
            00050 ;*                                                        *
            00060 ;* Das Programm KBREAD/SOU liest, nachdem es alle         *
            00070 ;* Register auf den Stack gerettet hat, einen max. 64     *
            00080 ;* Zeichen langen String ein und schreibt ihn in den     *
            00090 ;* Speicherbereich ab 0FFC0H. Die Länge des Strings      *
            00100 ;* befindet sich an Speicherstelle 0FFBFH.                *
            00110 ;* Das Unterprogramm GETSTR schreibt diesen String in    *
            00120 ;* die Variable A$.                                       *
            00130 ;*                                                        *
            00140 ;* Ausgeblendete Tasten:                                  *
            00150 ;*                        1. SHIFT -->  (ASC 16H)         *
            00160 ;*                        2. CLEAR      (ASC 1FH)         *
            00170 ;*                        3. BREAK      (ASC 01H)         *
            00180 ;*                                                        *
            00190 ;* Geanderte Tasten:                                      *
            00200 ;*                        1. CTRL (ASC 0AH) in "ä"        *
            00210 ;*                                                        *
            00220 ;* Die Länge des Puffers (Kleiner 40H) wird vom           *
            00230 ;* BASIC übernommen.                                      *
            00240 ;*                                                        *
            00250 ;*                                                        *
            00260 ;*********************************************************
            00270 ;
            00280 ;
            00290 ;
FED8        00300          ORG     0FED8H
            00310 ;
            00320 ; 16 KByte:  ORG 07ED8H
            00330 ; 32 KByte:  ORG 0BED8H
            00340 ;
FED8 F5     00350          PUSH    AF              ;Rette Register
FED9 C5     00360          PUSH    BC              ;  "      "
FEDA D5     00370          PUSH    DE              ;  "      "
FEDB E5     00380          PUSH    HL              ;Rette Register
FEDC CD7F0A 00390          CALL    0A7FH           ;BASIC-Parameter nach HL
FEDF 45     00400          LD      B,L             ;Länge des Puffers nach B
FEE0 21C0FF 00410          LD      HL,STRBUF       ;Pufferadresse nach HL
            00420 ;
            00430 ; Hier beginnt das Level II Rom, Adresse 05D9H
            00440 ;
FEE3 E5     00450 KEYBD    PUSH    HL              ;HL zeigt auf den Puffer
FEE4 3E0E   00460          LD      A,0EH           ;Kontrollzeichen "Cursor an"
FEE6 CD3300 00470          CALL    0033H           ;Cursor an
FEE9 48     00480          LD      C,B             ;C = Länge des Puffers
FEEA CD4900 00490 L05E0H   CALL    0049H           ;warte auf Eingabe
            00500 ;
            00510 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            00520 ;
            00530 ; Zusätzliche Zeilen. Ändert CTRL (ASC 0AH) zu "ä" (ASC 7BH)
            00540 ; und blendet CLEAR (ASC 1FH), BREAK (ASC 01H) und SHIFT Rchts-
            00550 ; pfeil (ASC 19H) aus.
            00560 ;
            00570 ;
FEED FE0A   00580          CP      0AH             ;Teste auf CONTROL
FEEF 2002   00590          JR      NZ,CONT         ;Sprung, falls ungleich CONTROL
FEF1 3E7B   00600          LD      A,7BH           ;Andere CONTROL zu "ä"
FEF3 FE19   00610 CONT     CP      19H             ;Teste auf SHIFT Rechtspfeil
FEF5 28F3   00620          JR      Z,L05E0H        ;Blende Taste aus
FEF7 FE1F   00630          CP      1FH             ;Teste auf CLEAR
FEF9 28EF   00640          JR      Z,L05E0H        ;Blende Taste aus
FEFB FE01   00650          CP      01H             ;Teste auf BREAK
FEFD 28EB   00660          JR      Z,L05E0H        ;Blende Taste aus
            00670 ;
            00680 ;
            00690 ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Bild 1. Dieses Programm ersetzt den Tastaturtreiber, der im ROM von Adresse 5D9H bis Adresse 0673H gespeichert ist

```
            00700 ;
FEFF FE20   00710          CP      20H             ;Teste auf Blank
FF01 3019   00720          JR      NC,L060CH       ;Weder Blank noch Kontrollzeichen
FF03 FE0D   00730          CP      0DH             ;Teste auf ENTER
FF05 CA6AFF 00740          JP      Z,L0662H        ;Sprung, falls ENTER gedrückt
            00750 ;
            00760 ;        CP      1FH             ;Teste auf CLEAR
            00770 ;        JR      Z,L0619H        ;CLEAR ausgeblendet
            00780 ;        CP      01H             ;Teste auf BREAK
            00790 ;        JR      Z,L0661H        ;BREAK ausgeblendet
            00800 ;
FF08 11EAFE 00810          LD      DE,L05E0H       ;Lade Rücksprungadresse
FF0B D5     00820          PUSH    DE              ;Rucksprungadresse auf Stack
FF0C FE08   00830          CP      08H             ;Teste auf SHIFT Linkspfeil
FF0E 2827   00840          JR      Z,L0630H        ;Sprung, falls SHIFT Linkspfeil
FF10 FE18   00850          CP      18H             ;Teste auf Linkspfeil
FF12 281E   00860          JR      Z,L062BH        ;Sprung, falls Linkspfeil
FF14 FE09   00870          CP      09H             ;Teste auf Rechtspfeil
FF16 2833   00880          JR      Z,L0646H        ;Sprung, falls Rechtspfeil
            00890 ;
            00900 ;        CP      19H             ;Teste auf SHIFT Rechtspfeil
            00910 ;        JR      Z,L0641H        ;SHIFT Rechtspfeil ausgeblendet
            00920 ;
FF18 FE0A   00930          CP      0AH             ;Teste auf CONTROL
FF1A C0     00940          RET     NZ              ;Zurück zu L05E0H, falls kein CONTROL
FF1B D1     00950          POP     DE              ;Hole Rücksprungadresse vom Stack
FF1C 77     00960 L060CH   LD      (HL),A          ;Druckbares Zeichen empfangen, sichern
FF1D 78     00970          LD      A,B             ;240 - Anzahl empfangener Zeichen
FF1E B7     00980          OR      A               ;Setze Status
FF1F 28C9   00990          JR      Z,L05E0H        ;Ignorieren, falls Puffer voll (außer CR)
FF21 7E     01000          LD      A,(HL)          ;Hole letzten Buchstaben
FF22 23     01010          INC     HL              ;Erhöhe Pufferadresse um 1
FF23 CD3300 01020          CALL    0033H           ;Drucke letzten Buchstaben
FF26 CDA2FF 01030          CALL    SOUND           ;BEEP
FF29 05     01040          DEC     B               ;
FF2A 18BE   01050          JR      L05E0H          ;Hole nächstes Zeichen
            01060 ;
            01070 ;L0619H  CALL    01C9H           ;Die nächsten 5 Zeilen werden
            01080 ;        LD      B,C             ;von CLEAR benötigt. Diese Taste
            01090 ;        POP     HL              ;ist ausgeblendet.
            01100 ;        PUSH    HL              ;
            01110 ;        JP      L05E0H          ;
            01120 ;
FF2C CD37FF 01130 L0622H   CALL    L0630H          ;Warte auf nächste Taste
FF2F 2B     01140          DEC     HL              ;Geh zum letzten Zeichen (1 vor ENTER)
FF30 7E     01150          LD      A,(HL)          ;Hol es und teste auf CONTROL
FF31 23     01160          INC     HL              ;Korrigiere Pufferadresse
            01170 ;
            01180 ;        CP      0AH             ;Teste auf CONTROL
            01190 ;        RET     Z               ;CONTROL ausgeblendet
            01200 ;
FF32 78     01210 L062BH   LD      A,B             ;Puffer voll?
FF33 B9     01220          CP      C               ;Erhaltene Zeichen - Pufferlänge
FF34 20F6   01230          JR      NZ,L0622H       ;Sprung, falls Platz für neue Daten
FF36 C9     01240          RET                     ;Zurück, Puffer voll
FF37 78     01250 L0630H   LD      A,B             ;B=erhaltene Zeichen, C=Pufferlänge
FF38 B9     01260          CP      C               ;Puffer voll?
FF39 C8     01270          RET     Z               ;Ja, Rücksprung
FF3A 2B     01280          DEC     HL              ;Zeiger auf vorhergehendes Zeichen
FF3B 7E     01290          LD      A,(HL)          ;Hole es
FF3C FE0A   01300          CP      0AH             ;Teste auf CONTROL
FF3E 23     01310          INC     HL              ;Zeiger auf zuletzt erhaltenes Zeichen
FF3F C8     01320          RET     Z               ;Rücksprung, falls CONTROL
FF40 2B     01330          DEC     HL              ;überschreibe letztes Zeichen
FF41 3E08   01340          LD      A,08H           ;Backspace Kontrollzeichen
FF43 CD3300 01350          CALL    0033H           ;Drucke Backspace
FF46 CDA2FF 01360          CALL    SOUND           ;BEEP
FF49 04     01370          INC     B               ;Korrigiere Zeichenzähler
FF4A C9     01380          RET                     ;Rücksprung
            01390 ;
            01400 ;L0641H  LD      A,17H           ;Benutzt von SHIFT Rechtspfeil
            01410 ;        JP      0033H           ;Taste ausgeblendet
            01420 ;
```

```
FF4B CD4803 01430 L0646H   CALL    0348H           ;Warte auf nächste Taste
FF4E E607   01440          AND     07H             ;Isoliere die unteren 3 Bits
FF50 2F     01450          CPL                     ;Invertiere Akku
FF51 3C     01460          INC     A               ;Gibt 1 <= X <= 8
FF52 C608   01470          ADD     A,08H           ;Lösche die oberen Bits des Zählers
FF54 5F     01480          LD      E,A             ;Sichere Zahl der zu addierenden Blanks
FF55 78     01490 L0650H   LD      A,B             ;Hole Größe des verbleibenden Platzes
FF56 B7     01500          OR      A               ;Puffer voll?
FF57 C8     01510          RET     Z               ;Ja, Rücksprung
FF58 3E20   01520          LD      A,20H           ;Lade Blank
FF5A 77     01530          LD      (HL),A          ;Space in den Puffer
FF5B 23     01540          INC     HL              ;Zeige auf nächstes Pufferzeichen
FF5C D5     01550          PUSH    DE              ;Sichere DE-Register des Hauptprogramms
FF5D CD3300 01560          CALL    0033H           ;Drucke Blank
FF60 D1     01570          POP     DE              ;Hole DE-Register des Hauptprogramms
FF61 05     01580          DEC     B               ;Verringere Zeichenzähler um 1
FF62 1D     01590          DEC     E               ;Genügend Blanks gedruckt?
FF63 CCA2FF 01600          CALL    Z,SOUND         ;BEEP
FF66 C8     01610          RET     Z               ;Ja, Rücksprung
FF67 18EC   01620          JR      L0650H          ;Nein, weiter bis genügend gedruckt
FF69 37     01630 L0661H   SCF                     ;CARRY-Flag gesetzt, falls BREAK gedrückt
FF6A F5     01640 L0662H   PUSH    AF              ;ENTER gedrückt
FF6B 3E0D   01650          LD      A,0DH           ;Lade CR
FF6D 77     01660          LD      (HL),A          ;Stelle CR in den Puffer
FF6E CD3300 01670          CALL    0033H           ;Drucke CR
FF71 CDA2FF 01680          CALL    SOUND           ;BEEP
FF74 3E0F   01690          LD      A,0FH           ;Lade Cursor-aus Kode
FF76 CD3300 01700          CALL    0033H           ;Cursor aus
FF79 79     01710          LD      A,C             ;C = Pufferlänge
FF7A 90     01720          SUB     B               ;Pufferlänge - erhaltene Zeichen
FF7B 47     01730          LD      B,A             ;Hole Anzahl der Zeichen in Puffer
FF7C F1     01740          POP     AF              ;Rekonstruiere AF-Register
FF7D E1     01750          POP     HL              ;HL = Startadresse des Puffers
            01760 ;        RET                     ;Original war Unterprogramm
            01770 ;
            01780 ;Hier endet das Level II Rom, Adresse 0673H
            01790 ;
FF7E 21BFFF 01800          LD      HL,LENBUF       ;Hole Pufferadresse
FF81 70     01810          LD      (HL),B          ;Empfangene Zeichen (ohne CR)
FF82 CD8DFF 01820          CALL    GETSTR          ;Stelle Puffer nach A$
FF85 E1     01830          POP     HL              ;Rekonstruiere Register
FF86 D1     01840          POP     DE              ;     "          "
FF87 C1     01850          POP     BC              ;     "          "
FF88 F1     01860          POP     AF              ;Rekonstruiere Register
FF89 CD9A0A 01870          CALL    0A9AH           ;Stelle HL in BASIC-Variable
FF8C C9     01880          RET                     ;Zurück in's BASIC
            01890 ;
            01900 ; Unterprogramm GETSTR           vgl. MC 10/83
            01910 ;
260D        01920 CREVAR   EQU     260DH           ;Suche Variablenadresse
FF8D 21BCFF 01930 GETSTR   LD      HL,NAME         ;Adresse des gew. Variablennamens
FF90 CD0D26 01940          CALL    CREVAR          ;Erzeuge String A$
FF93 D5     01950          PUSH    DE              ;Sichere Variablenadresse
FF94 3ABFFF 01960          LD      A,(LENBUF)      ;Lade Stringlange und
FF97 12     01970          LD      (DE),A          ;schreibe in Variablentabelle
FF98 11C0FF 01980          LD      DE,STRBUF       ;Pufferadresse nach DE
FF9B E3     01990          EX      (SP),HL         ;Stelle Puffer nach A$
FF9C 23     02000          INC     HL              ;          "
FF9D 73     02010          LD      (HL),E          ;          "
FF9E 23     02020          INC     HL              ;          "
FF9F 72     02030          LD      (HL),D          ;Stelle Puffer nach A$
FFA0 E1     02040          POP     HL              ;Rekonstruiere Programmzähler
FFA1 C9     02050          RET                     ;Zurück ins Hauptprogramm
            02060 ;
            02070 ; Unterprogramm SOUND
            02080 ; Dieses Unterprogramm erzeugt auf dem Casettenport (0FFH)
            02090 ; einen kurzen Piepslaut.
            02100 ;
00FF        02110 PORT     EQU     0FFH            ;Casettenport
0004        02120 DUR      EQU     004H            ;Tondauer
0078        02130 FREQ     EQU     078H            ;Tonfrequenz
FFA2 F5     02140 SOUND    PUSH    AF              ;Sichere Register
FFA3 C5     02150          PUSH    BC              ;Sichere Register
```

```
FFA4 0E04   02160          LD      C,DUR           ;Stelle Tonlänge nach C
FFA6 0678   02170 LOOP1    LD      B,FREQ          ;Hole Frequenzwert
FFA8 3E01   02180          LD      A,1             ;1. Ausgabewert
FFAA D3FF   02190          OUT     (PORT),A        ;Wert durch Port ausgeben
FFAC 10FE   02200 LOOP2    DJNZ    LOOP2           ;Warte bis entsprechende Freq. erzeugt
FFAE 0678   02210          LD      B,FREQ          ;Hole Frequenzwert
FFB0 3E02   02220          LD      A,2             ;2. Ausgabewert
FFB2 D3FF   02230          OUT     (PORT),A        ;Wert durch Port ausgeben
FFB4 10FE   02240 LOOP3    DJNZ    LOOP3           ;Warte, bis entsprechende Freq. erzeugt
FFB6 0D     02250          DEC     C               ;Verringere Tonlängenzähler um 1
FFB7 20ED   02260          JR      NZ,LOOP1        ;Zurück, bis Zähler auf 0
FFB9 C1     02270          POP     BC              ;Rekonstruiere Register
FFBA F1     02280          POP     AF              ;Rekonstruiere Register
FFBB C9     02290          RET                     ;Zurück in's Hauptprogramm
FFBC 41     02300 NAME     DEFM    'A$'            ;Name der Zeichenkette
FFBE 00     02310          DEFB    0               ;Ende des Namens
0001        02320 LENBUF   DEFS    0001H           ;Reserviere Platz für Pufferzähler
0040        02330 STRBUF   DEFS    0040H           ;Reserviere Platz für Puffer
0000        02340          END
```

Programmbibliotek

Die meisten von Euch haben die neue Liste über die CLUB-Programme schon beim CLUB-Treffen erhalten. Die nichtanwesenden Mitglieder bekommen die Liste mit dem INFO zugeschickt.

Im nächsten INFO wird dann wieder ausführlich über neue Programme in der Bibliotek informiert.

„Die Fehlersuche wird wesentlich erleichtert, wenn intakte Geräte zur Verfügung stehen."

Aus einer Bundeswehrvorschrift

Strukturierte Programmierung

Hilfsmittel zur professionellen Programmierung

Wenn Programme für den kommerziellen oder wissenschaftlichen Einsatz geplant sind, ergeben sich Aspekte, die eine ausführliche Dokumentation und eine leicht nachvollziehbare Struktur erfordern. Inwieweit sich hierfür die Strukturierte Programmierung — mit wenigen Einschränkungen — anbietet, beleuchtet dieser Beitrag.

Wer sich mit der Programmierung von Computern beschäftigt, hat sicher schon folgendes erlebt. Man hat eine Idee für eine Problemlösung, einige kleine Skizzen bestätigen: es wird funktionieren. Die Idee läßt sich schnell umsetzen, das Programm läuft — und jetzt beginnt die Arbeit erst richtig.

Die kurze, schnelle Lösung kann zwar der Programmschreiber bedienen, zur Vereinfachung sind jedoch in vielen Fällen Ein- und Ausgaben, diverse Konstanten, eventuell sogar Daten im Programm festgelegt. Absicherungen gegen fehlerhafte Bedienung fehlen völlig, Kommentare sind in der Regel sehr kurz gehalten. Eventuell muß das Programm für jeden einzelnen Anwendungsfall neu assembliert oder compiliert werden.

Soll ein anderer dieses Programm benutzen, so muß er sich erst in die Funktion des Programmes einarbeiten, um die Änderungen auszuführen. Damit wird der Kreis der Benutzer auf die Profis begrenzt, die ohnehin in der Lage wären, ein derartiges Programm selbst zu schreiben. Was muß geschehen, damit auch ein unbedarfter Bediener mit dem Programm zurechtkommt? Es muß eine Bedienerführung her, die Ein- und Ausgaben müssen sinnvoll aufgebaut werden. Die Abspeicherung auf die Diskette muß so geschaltet sein, daß nicht das gesamte System abstürzt, wenn ein Fehler gemacht wird.

Sind alle Arbeiten ausgeführt, so hat die ursprüngliche Lösung nur noch einen Anteil von wenigen Prozent des gesamten Programms.

Nebensachen?

Typisch für die unstrukturierte Arbeitsweise ist, daß der Programmschreiber sich durchaus Gedanken über das Programm gemacht hat; nur hat er sich dabei auf die Lösung eines Teilproblems (vermutlich auf 'den' Lösungsalgorithmus) beschränkt. Die unwichtigen 'Nebensächlichkeiten' wurden bestenfalls mit einem flüchtigen Gedanken bedacht! Dabei entsteht meistens ein Programm, das die Zusammenarbeit mit anderen Funktionen nicht berücksichtigt. An die damit vorgegebene Struktur müssen sich alle späteren Erweiterungen halten. Prinzipiell kann man zwar auch den schon geschriebenen Code wieder modifizieren, wenn sich dies als notwendig erweist. Die Praxis zeigt jedoch, daß man nur äußerst ungern bereits getestete Programmteile wieder umschreibt, wobei oftmals neue Fehlerquellen entstehen. Diese Vorgehensweise führt dann zu den (heute fast zur Regel gehörenden) Terminüberschreitungen bei Software-Projekten. Hinzu kommt noch, daß bei vielen praktischen Projekten am Anfang nicht einmal eine exakte Vorstellung über die Funktion des Systems besteht.

Das soll nicht heißen, daß darüber nicht nachgedacht wurde, sondern daß unter dem Gesichtspunkt der Programmerstellung und der technischen Realisierungsmöglichkeiten durchaus wesentliche Aspekte einfach nicht erkannt beziehungsweise übersehen wurden. Schon Kleinigkeiten können erhebliche Probleme verursachen, wenn sie an den technischen Grenzen der betreffenden Systeme Änderungen bewirken. Dies gilt insbesondere für Echtzeitanforderungen!

Ein Rechner, dessen Arbeitsgeschwindigkeit gerade ausreicht, um einen Analog-/Digitalumsetzer mit maximaler Datenrate zu betreiben, wird nicht ohne wesentliche Verringerung der Abtastrate in der Lage sein, zusätzlich eine Tastatur auf eine bestimmte Taste hin abzufragen. Wenn der Hardware-Entwurf die Möglichkeit des Interrupts durch diese Taste nicht berücksichtigt, so ist entweder eine Hardware-Änderung nötig oder eine Verringerung der Abtastrate muß in Kauf genommen werden. Gleichzeitig steigt der Entwurfsaufwand in diesem Fall wesentlich an, da der gesamte Programmteil zeitlich zu optimieren ist.

Allen geschilderten Problemen ist eines gemeinsam: Vor Beginn der Arbeiten wurden nicht alle Probleme des Hard- und Software-Designs ausreichend durchdacht. Infolgedessen schaffen die schon realisierten Teilbereiche Randbedingungen, die in allen späteren Arbeiten berücksichtigt werden müssen. Bei der Systemplanung nicht berücksichtigte Eigenheiten können den Arbeitsaufwand eines Projektes äußerst negativ beeinflussen, falls sie beispielsweise eine Änderung an fertiggestellten Programmteilen oder an der Hardware erfordern. Nicht [illegible] zen ist auch der zusätzliche Testaufwand, der zur Prüfung des Zusammenspiels zwischen vorhandenen und neuen Routinen notwendig ist.

Wurde für das Design zum Beispiel eine höhere Sprache verwendet, so kann eine Systemerweiterung unter Umständen erfordern, größere Teile des Programms in Assembler zu schreiben. Dies bringt aber einen erheblichen Zeitaufwand zur Lösung von Problemen mit sich, die schon einmal unter anderen Gesichtspunkten gelöst worden waren.

Ähnliche Probleme treten im laufenden Einsatz von Systemen und Programmen auf. Im Laufe der Lebensdauer müssen Anpassungen an neue Peripheriesysteme (wie Plotter, Drucker) erfolgen, Teile der Lösung sind oftmals zu ersetzen, weil sich Anforderungen geändert haben (zum Beispiel gesetzliche Vorschriften, die Verarbeitung anderer Materialien). Im Laufe der Zeit sind einzelne Bauelemente nicht mehr zu beschaffen, und der Ersatz (zum Beispiel AD-Umsetzer) erfordert eine neue Anpassung. Aus wirtschaftlicher Sicht ist daher die einmalige Erstellung eines Programmes nicht der einzige Faktor, der bei der Programmierung entscheidend ist.

Es müssen die Kosten für die gesamte Lebensdauer eines Programms beziehungsweise Systems berücksichtigt werden. Ein geringfügiger Mehraufwand entsteht durch die Erstellung ausführlicher Unterlagen und einem prüfungsfreundlichen Gesamtdesign. Das zahlt sich in der Regel aber über die gesamte Einsatzzeit eines Programmes aus. Wenn Programme das Wohl und Wehe von Menschen beeinflussen können, sogenannte sicherheitsrelevante Software (medizinische Technik, Maschinensteuerung), ist eine ausführliche, unmittelbar verständliche Dokumentation sowieso zwingende Voraussetzung. Sie haben sonst bei der gutachterlichen Prüfung keine Chance, zu bestehen. Bei größeren Projekten (die schon in Mannjahren gemessen werden) kann man auf eine gründliche Vorplanung ohnehin nicht verzichten, so daß hier durch die Anwendung der Strukturierten Programmierung kein zusätzlicher Auf-

Ausnahmen

Es sei allerdings nicht verschwiegen, daß es Gründe geben kann, ganze Programme oder Teile davon nicht strukturiert auszuführen. Das sind:

Kostenfragen bei Großserien.
Ein Beispiel dafür sind die Homecomputer, Matrixdrucker und Typenradschreibmaschinen. Wenn Serien mit Millionenstückzahlen zum Einsatz kommen, kann der Unterschied zwischen einem oder zwei notwendigen ROMs oder dem nächst größeren EPROM oder ROM schon mehrere Millionen kosten. Alle möglichen Tricks sind erlaubt, wenn nur dadurch das Programm verkürzt wird. Selbst wenn der 'Spaghetti-Code' nur sehr schwer zum Laufen zu bringen ist und damit die Entwicklung entsprechend teuer wird, so ist dies durch die Kosteneinsparung in der Serienproduktion zu rechtfertigen.

Zeitanforderungen.
Muß eine Hardware bis an die Grenzen ihrer Leistungsfähigkeit ausgenutzt werden, so ist jeder Weg, der dieses Ziel erreicht, sinnvoll. Zusätzliche Befehle sind hier überflüssig, wenn sie nur der Prüfbarkeit dienen. In diesem Fall sollte man sich jedoch die Frage stellen, ob nicht ein genereller Designfehler vorliegt. Bei kleineren Serien sollte die Hardware nämlich nie bis an die möglichen Grenzen ausgenutzt werden, da dabei der Programmier-, Test- und Wartungsaufwand beträchtlich steigt. Außerdem wird oftmals die gesamte Programmierkunst aufgewendet, um einen Algorithmus zeitoptimal zu kodieren. Wenn man aber statt dessen den Algorithmus selbst optimierte, könnte auch die Strukturierte Programmierung zu wesentlich schnelleren Ergebnissen führen.

Kurze Testroutinen,
die nur einmal benutzt werden, um einen vermuteten Fehler einzukreisen.

Rationell und wartungsfreundlich

Wie kommt man zu einem rationellen, wartungsfreundlichen Design? Die Antwort ist das 'Strukturierte Programm' [illegible]

Da kaum eine Programmieraufgabe ohne Berücksichtigung der Hardware auskommt und im industriellen Bereich aus Kostengründen nicht einfach eine Hardware vorgegeben werden kann, hängen in vielen Anwendungsfällen Programmierung und Hardware-Entwicklung voneinander ab. Das Geheimnis der Strukturierten Entwurfsverfahren liegt darin, daß man die einzelnen Komponenten einer technischen Lösung mehrfach auf verschiedenen Ebenen durchdenkt und genaue Vorstellungen über möglichst alle Eigenschaften des gesamten Programms oder Systems entwickelt, bevor(!) die erste Programmzeile geschrieben und der erste Schaltungsentwurf in Angriff genommen wird.

An eine gute Software beziehungsweise an ein gutes Systemdesign sollten die folgenden Forderungen gestellt werden:

Einfache Prüfbarkeit.
Ideal wäre die Prüfung allein durch die logische Nachverfolgung der Entwurfsunterlagen. Praktisch muß aber gefordert werden, daß der Zugriff auf die einzelnen Routinen und Variablen (bei Systemen auch der Hardware-Baugruppen) einfach möglich ist (z. B. aktivierbare Statusausdrucke).

Gute Dokumentation.
Nur auf diese Weise läßt sich sicherstellen, daß mehrere Personen gleichzeitig an dem Projekt arbeiten können, und später Änderungen auch von anderen Personen durchführbar sind. Eine Dokumentation kann nur dann aktuell sein, wenn sie laufend dem Stand der Arbeiten angepaßt wird. Dies wiederum wird in der Regel nur dann geschehen, wenn die Bearbeiter

Schritt 1:	Arbeitsweise des zu entwickelnden Systems grob beschrieben.
Schritt 2:	Technische Randbedingungen festlegen (Algorithmen, Ein- und Ausgaben, Speicherplatzbedarf...).
Schritt 3:	Festlegung der Arbeitsabläufe (Schnittstellen zwischen den Programmen festlegen, Testmöglichkeiten vorsehen).
Schritt 4:	Erstellung und Test des Programmes.

Tabelle 1. Hierarchische Anordnung der Entwurfsschritte

```
I   Idee grob schriftlich fixieren                                              I
I I I I   Einzelne Anforderungen festlegen und praezisieren (nach Schema)       I
I I I I IF Sind einzelne Anforderungen widerspruechlich?
I I I I I THEN .................................................... ELSE
I I I I I   Anforderungen aendern.                          I   I
I I I I   Bis alle Anforderungen realisierbar sind.
I I I   Bis alle Anforderungen vorliegen.
I I I   Strukturdiagramme entwerfen
I I I IF Gibt es Konflikte mit den Anforderungen oder den techn. Moeglichkeiten?
I I I I THEN .................................................... ELSE
I I I I   Strukturdiagramme modifizieren                    I   I
I I I   Bis Anforderungen erfuellt, oder Anforderungen geaendert werden muessen.
I I   Bis Anforderungen erfuellt und Programmstruktur realisierbar.
I I I   Programme entwerfen
I I IF Gibt es Konflikte mit den Anforderungen oder den techn. Moeglichkeiten?
I I I THEN .................................................... ELSE
I I I   Programme modifizieren                                I   I
I I   Bis Strukturdiagramme realisiert,od Anforderungen geaend. werden muessen.
I   Bis Anforderungen erfuellt und Programm lauffaehig.                         I
```

Programm 1. Ablauf eines Programmentwurfes

einen Nutzen in der Erstellung und Aktualisierung der Dokumentation sehen.

Geplante Reaktion auf unvorschriftsmäßige Betriebsfälle.
So sollten falsche Eingaben (zum Beispiel Buchstaben anstelle von Ziffern) erkannt werden, zu einer Fehlermeldung führen und die Wiederholung der Eingabe zulassen. In diesem Sinne gibt es keine Eingaben beziehungsweise Eingangszustände, die 'in der Praxis' nicht vorkommen. Eine Selbstzerstörung des Programms durch falsche Eingaben (zum Beispiel Überschreiben von Eingabepuffern) darf es nicht geben!

Maßnahmen

Mit der Zeichnung von Nassi-Shneidermann-Diagrammen (Strukturdiagramme) allein können die vorgestellten Forderungen nicht erfüllt werden. Auch Strukturdiagramme garantieren aus sich heraus nicht, daß alle wesentlichen Punkte des Designs bedacht werden. Dies gilt besonders für komplexe Projekte, an denen mehrere Personen gleichzeitig arbeiten sollen. Es gilt auch für den Anfänger und den weniger geübten System-Designer bei kleineren Programmen.

Das erste Ziel muß es sein, den Ansatzpunkt für die sinnvolle Erstellung von Strukturdiagrammen zu finden. Es ist zu bedenken, daß wohl kein Design nur nach streng logischen Gesichtspunkten aufgebaut werden kann. Viele Entscheidungen (zum Beispiel Befehlssequenzen zur Steuerung eines zu entwerfenden Systems) sind einzig dem Entwickler und seiner Erfahrung zu überlassen. Nicht in jedem Falle gibt es Kriterien, die eine Bewertung die-ser Entscheidungen erlauben. Weitere Freiheitsgrade ergeben sich durch die Auswahl der Hardware. Sie kann eigentlich erst erfolgen, wenn alle Anforderungen definiert sind.

Andererseits ist es unzweckmäßig, die Anforderungen unabhängig von jeder Hardware zu definieren, da dann unter Umständen sehr schnell eine Situation entsteht, in der zur Problemlösung ein Cray-Rechner (sehr schneller und teurer Array-Prozessor) benötigt wird, aber möglichst nur zum Preis eines einzigen Personal Computers. Praktisch muß man während der Festlegung der Anforderungen ständig prüfen, ob die vorgesehene Hardware die Forderungen (unter anderem die Kosten) auch erfüllen kann.

Alle Schritte sollten schriftlich fixiert werden, damit bei notwendigen Änderungen die bereits getroffenen Entscheidungen noch nachvollziehbar sind. Diese Forderung mag zunächst nicht notwendig erscheinen. Bedenkt man jedoch, daß ein Software-Projekt durchaus mehrere Monate oder Jahre dauern kann und daß dabei unter Umständen mehrere Personen mit der Programmierung befaßt sind, so erscheint eine schriftliche Fixierung notwendig. Wer hat nicht schon einmal ein selbsterstelltes älteres Programm ändern müssen und sich dabei über die mangelnde Dokumentation geärgert. Ganz zu schweigen davon, daß die eigene Erinnerung an die Absichten, die bei der Programmerstellung zugrunde gelegen haben, nur noch lückenhaft ist.

Es hat sich gezeigt, daß ein Design vom Problem hin zur Lösung wachsen muß (Tabelle 1), [illegible]

Problem möglichst genau bekannt sein. Im nächsten Schritt werden die technischen Randbedingungen der Lösung festgelegt (wie Algorithmen, Zahl und Art der Schnittstellen). Der folgende Schritt legt die Arbeitsabläufe fest (Strukturdiagramme) und bereitet damit den letzten Schritt, die Kodierung des Problems vor.

Programm 1 gibt den Ablauf des Designs als vereinfachtes Strukturdiagramm wieder. Die darin enthaltenen Iterationsmöglichkeiten stellen den praktischen Fall dar, bei dem meist Korrekturen an den Ausgangsannahmen notwendig sind.

Dabei sollte man bedenken, daß der Aufwand für eine Korrektur immer größer wird, je später sie erfolgt!

Der systematische Entwurf eines Systems soll diese Iterationen auf das machbare Mindestmaß reduzieren. Ein weiterer Vorteil dieser Vorgehensweise ist eine genaue Systembeschreibung, die es nicht nur dem Entwickler, sondern auch anderen möglich macht, den Sinn eines Programmes oder Systems ohne hohen Arbeitsaufwand zu verstehen.

Wie die Schritte 1 und 2 aussehen können, wird am Schluß dieses Beitrags an einem Beispiel demonstriert.

Anwendungen

Der Schritt 3 bedarf einer weitergehenden Erläuterung. Prinzipiell läßt sich ein Strukturiertes Programm auch durch Flußdiagramme (Kästchendiagramme) dokumentieren. Allerdings verführen diese Diagramme dazu, 'Spaghetti-Code', das heißt einen sehr vermaschten Code zu erzeugen. Außerdem bieten die Kästchendiagramme nur wenig Platz für eine Beschriftung, so daß häufig eine zusätzliche Beschreibung des Flußdiagramms notwendig ist. Übergeordnete Strukturen sind zwar darstellbar, bieten sich durch die Art der Darstellung jedoch nicht ohne weiteres an. Hierbei zu einer sinnvollen Aufteilung zu kommen, ist daher mit zusätzlichem Arbeitsaufwand verbunden.

Die Darstellungsart nach Nassi und Shneidermann geht einen anderen Weg. Ausgehend von der Forderung eines modularen Programmaufbaues und der Forderung nach nur einem Moduleingang und nur einem Modulausgang wird eine grafische Darstellung mit nur drei Grundelementen möglich:

— das Modul
— die Schleife
— die Verzweigung

Ein Modul kann dabei ein weiteres Strukturdiagramm repräsentieren, es kann aber auch nur einen einzelnen Vorgang (Anweisung) enthalten.

Eine Schleife wiederholt ein Modul so lange, bis eine vorgegebene Bedingung erfüllt wurde.

Die Verzweigung erlaubt die Auswahl zwischen zwei Submodulen innerhalb eines Moduls und damit die Ausführung von Entscheidungen.

In der Praxis werden einige weitere Konstruktionen verwendet, zum Beispiel mehrfache Fallunterscheidungen (switch, CASE) und abweisende beziehungsweise nichtabweisende Schleifen (siehe auch 'Planvoll programmieren kann jeder' in diesem Heft).

Allein durch die Anordnungsmöglichkeiten schränkt diese Art der Darstellung die Entwurfsfreiheit bei der Programmierung ein. Sie verhindert schon in der Entwurfsphase ein seitliches Einspringen in Programme (mehrfache Einsprungpunkte). Es ist auch nur ein Programmende möglich. Der Assemblerprogrammierer wird hier sicher den Kopf schütteln und auf den erhöhten Programmieraufwand hinweisen, den die Realisierung dieser Struktur auf Assemblerebene erfordert (Programm 2). Gleiches gilt für BASIC- und FORTRAN-Anwender.

Bei höheren Programmiersprachen wie ADA, C oder PASCAL ist diese Art des Programmendes ohnehin selbstverständlich und kann vom Anwender nicht geändert werden.

Aber auch in Assemblerprogrammen gibt es Gründe für den etwas aufwendigeren Code. Schon die einfache Routine im Programm 2a zeigt das Problem sehr deutlich auf. Mit Hilfe eines Debuggers kann in der Routine kein Haltepunkt gesetzt werden, es sei denn, man würde es in Kauf nehmen, bei jedem Durchlauf den Haltepunkt neu zu setzen (zum Beispiel mit DDT). Bei wenigen Durchläufen mag dies noch möglich sein, bei Rechenschleifen wird es aber zur Last.

Will man feststellen, welche Wirkung die Routine gehabt hat, so muß man zuerst den Programmaufruf in der übergeordneten Routine ausfindig machen und nach der Rückkehr aus dem Unterprogramm die Programmausführung stoppen. Meistens wird die Routine von mehreren Stellen aufgerufen, so daß alle in Frage kommenden Routinenaufrufe mit Haltepunkten versehen werden müssen. Das ist jedoch sehr arbeitsintensiv und fehlerträchtig. Einfache Debugger wie DDT lassen ohnehin nur wenige Haltepunkte zu, so daß diese oftmals nur für eine Routine ausreichen.

Eine andere Möglichkeit ist die Untersuchung bestimmter Variablen, die am Ende verschiedener Routinen ausgegeben werden (zum Beispiel die Register des Mikroprozessors). Auch hier entstünde nur ein Zahlenfriedhof, erzeugte man in jedem Durchlauf einen Ausdruck, wie es bei der Routine nach Programm 2a nötig wäre. Auch hier bietet sich die Form nach Programm 2b an, die nur dann einen Ausdruck erzeugt, wenn die Routine beendet ist.

Selbstverständlich kann man auch hier Lösungen finden, die dann aber der jeweiligen Routine anzupassen sind. Ein Unterprogrammaufruf läßt sich einfach durch ein MACRO realisieren, das nach erfolgter Programmprüfung lediglich noch NOPs oder gar keine Befehle mehr enthält. Sonderlösungen müssen einzeln aus dem Programm entfernt werden. Dabei sollte man nicht unterschätzen, daß derartige Änderungsaktionen manchmal auch Code mitentfernen, der noch benötigt wird, und dann beginnt die Fehlersuche erneut.

Der Aufbau eines Strukturierten Programmes beginnt mit der höchsten hierarchischen Ebene, der Hauptprogrammschleife. Diese Schleife soll die wesentlichen Programmblöcke zusammenfassen. Es empfiehlt sich dabei, kein Strukturdiagramm größer als ein DIN- A4-Blatt zu machen, da es sonst unübersichtlich wird.

Weiterhin hat sich gezeigt, daß pro Diagramm maximal sechs Module nebeneinander dargestellt werden sollten. Die einzelnen Module sind dann in ge-

```
            RET NG
            RET Z
            .... ..
            .... ..     (CALL PRINT)
            JP SCHLEIFE
```

```
EINSPRUNG:  .... ..
            .... ..
SCHLEIFE:   .... ..     Programm-
            .... ..     anweisungen
            .... ..
            JP NC,ENDE
            JP Z,ENDE
            .... ..
            .... ..
            JP SCHLEIFE
ENDE        .... ..     (CALL PRINT)
            RET
```

Programm 2. Beispiel für 'Strukturierten Assemblercode', a) kurzer unstrukturierter Code, b) strukturierte Version

trennten Diagrammen weiter zu präzisieren. Schon diese Darstellung verlangt bei vielen Programmen die Festlegung von Prioritäten, die allerdings nicht immer eindeutig sein müssen. In der weiteren Arbeit wird nun ein Modulblock nach dem anderen als Strukturdiagramm dargestellt, wobei zunächst jeweils nur Blöcke der gleichen Ebene erstellt werden. Diese Arbeit setzt sich fort bis zu einer Ebene, die die einzelnen Aufgaben eindeutig festlegt (zum Beispiel 'Bestimme das Maximum zweier Werte und weise es X zu'). Dabei ist zu beachten, daß in den einzelnen Modulblöcken der untersten Ebene nicht etwa der Programmcode stehen sollte, sondern immer die Beschreibung der Funktion des Codes. Im Falle mathematischer Operation kann das jedoch identisch sein.

Strukturiertes Design: ein Beispiel

Ein einfaches Anwendungsbeispiel soll die Methode erläutern helfen. Der Gesamtumfang der Überlegungen würde allerdings den gesetzten Rahmen sprengen, so daß wir uns nur auf einen geringen Teil beschränken. Die Zweige sind deshalb lediglich bis zur vorletzten Ebene des Strukturdiagramms dargestellt; die eigentliche Programmierung bleibt unbehandelt. Eine Reihe von Vorgaben, wie etwa die Befehle des verwendeten Systems, können völlig frei gewählt werden. Die jeweilige Auswahl hängt dann aber von den Erfahrungen des Bearbeiters ab. Nicht jeder Designer ist in der Lage, jede dargelegte Entscheidung zu treffen, oder er würde abhängig vom eigenen Wissensstand vielleicht zu anderen Entscheidungen kommen. Es ist ein Merkmal fast aller Software-Entwicklungen, daß es keinen 'einzig richtigen' Weg gibt!

Der erste Schritt ist eine Kurzbeschreibung des Systems (siehe Kasten S. 62). Sie ist zunächst sehr oberflächlich und gibt nur einige Details an. Dieser Stand entspricht in etwa einer Produktidee, die zum Beispiel in einem Gespräch zwischen Entwicklung und Verkauf herauskommen könnte, wenn ein neues Produkt gesucht wird. Tatsächlich beginnen viele Entwicklungsaufgaben mit derartig 'dünnen' Vorgaben, dies gilt besonders dann, wenn der Partner in der Entwicklung aus einer völlig anderen Branche stammt und selbst keine Erfahrungen mit Rechnern hat.

Für den Beginn der Programmierung oder den Entwurf und die Auswahl einer Hardware müssen die vorliegenden Angaben präzisiert werden. Folgendes Schema hat sich beim Verfasser bewährt:

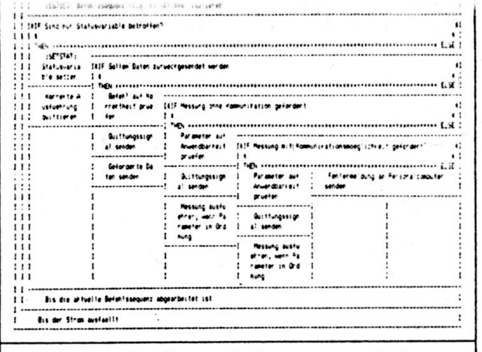
— Festlegen der Systemfunktionen (Reaktion auf zu verarbeitende Daten, Befehlssatz, anzuwendende Rechenverfahren, gesetzliche Vorschriften)
— Darstellung der speziellen Anforderungen (Speicherplatzbedarf, Arbeitsgeschwindigkeit, Interrupt-Reaktionen)
— Beschreibung der zu verarbeitenden Daten und Befehle (Befehlsbeispiel, Wortbreite, Art der Daten, Bedingungen der Befehlsübergabe, mögliche Konflikte bei mehreren Dateneingaben)
— Festlegung der physikalischen Ein- und Ausgaben (analoge Pegel, Datenprotokolle, Geschwindigkeiten, zu erwartende Probleme)
— Bekannte, aber ungelöste Problemstellungen (wie fehlende Mindestanforderungen an die Arbeitsgeschwindigkeit, unbekannte Übergabeprotokolle ...)
— Abschätzung der zu erwartenden Erweiterungen und Lösungsmöglichkeiten (zum Beispiel weitere sinnvolle Algorithmen, weitere Befehle, höherer Datendurchsatz)
— Zur Auswahl stehende Baugruppen der Hard- und Software (Rechnersysteme, Peripheriegeräte, Softwarepakete)

Da die Systemdefinitionen auf dieser Ebene noch sehr änderungsanfällig sind, empfiehlt es sich, den Entwurf auf einem Textverarbeitungssystem durchzuführen. Im wesentlichen entspricht diese Vorgehensweise der Schaffung eines Pflichtenheftes, wobei die Dokumentation sehr viel differenzierter ausfallen muß. Die Ausführung dieser Überlegungen ist im Kasten wiedergegeben, es ist allerdings anzumerken, daß hier der 'Endstand' nicht dargestellt werden kann, da der Umfang zu groß würde.

Diese Vorgaben befassen sich im wesentlichen mit der Funktion des Systems. Technische Details der Realisierung gehören hier nicht hinein, soweit sie nicht als unbedingt verbindlich gelten. Mit Hilfe dieser Vorgaben kann man nun darangehen, das erste Strukturdiagramm zu zeichnen.

Die höchste hierarchische Ebene ist im Programm 3 wiedergegeben. Selbstverständlich erfolgt zunächst eine Initialisierung, deren Einzelheiten noch nicht festgelegt werden müssen. Es folgt die Hauptschleife, die so lange laufen soll, wie das System in Betrieb ist. Daran schließt sich ein Block an, der auf einen Befehlssatz wartet. Dieses Modul bekommt den Namen :WAITBEF:, der später auch im Programm auftreten sollte. Das System wird also von sich aus nichts weiter tun, bis der Personal Computer eine gültige Befehlssequenz abgibt. In der Zuständigkeit dieses Moduls liegt es auch, eventuelle Abbruchbefehle oder Wiederholungsbefehle zu verarbeiten. Erst wenn alle Bedingungen dieses Moduls erledigt sind, wird der Befehlssatz an das nächste Programm übergeben.

Obwohl das erste Modul :WAITBEF: im Moment noch gar nicht geschrieben sein muß, kann man sich schon dem nächsten Modul zuwenden. Dafür muß der Befehlssatz im Detail festgelegt sein. Alle weiteren Aktionen des Systems ergeben sich aus den folgenden Verzweigungsanweisungen, die die weiteren Aufgaben entsprechend der vorliegenden Befehlssequenz (zum Beispiel K1T10N1000) ausführt. Es sind insgesamt fünf Fälle denkbar. Dies sind die reinen Zuweisungen auf Statusvariablen, beispielsweise Vorgaben für die Zahl der Meßwerte, oder die



Programm 3. Strukturdiagramm der Hauptprogrammschleife

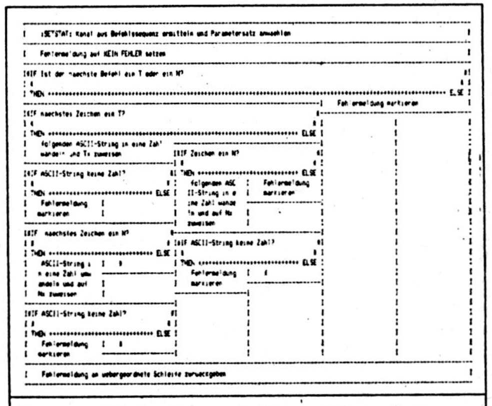
Dauer der Abtastzeit. Weiterhin gibt es Datenanforderungen, das heißt, Daten werden an den steuernden Personal Computer zurückgegeben.

Bei den Meßaufträgen sind ebenfalls zwei Fälle zu unterscheiden. Die Messung mit und ohne Kommunikationsmöglichkeit. Es fällt in die Zuständigkeit der Module, zu prüfen, ob die entsprechenden Voraussetzungen gegeben sind, und gegebenenfalls eine Fehlermeldung an den Prozessor abzusetzen. Das letzte Modul wird nötig, um falsche Befehlssätze zu verarbeiten beziehungsweise eine entsprechende Fehlermeldung abzusetzen. Auf diese Weise werden die restlichen Module vor formalen Fehlern im Befehlssatz geschützt und müssen nur noch auf Parameterfehler untersuchen.

Ein weiteres Problem ist die Quittierung eines ausführbaren Befehlssatzes, die im Interesse eines automatischen Betriebes notwendig ist. Manche Betriebssysteme können nur eine vordefinierte Zahl von Bytes lesen, und sie warten, bis die entsprechende Zahl von Bytes empfangen worden ist. Das erfordert eine definierte Vorgabe, damit der Anwender sein Steuerprogramm darauf einstellen kann, ohne in die Innereien seines Personal Computers einsteigen zu müssen. Da die Meldungen je nach Modul unterschiedlich sind, werden sie von den jeweiligen Modulen selbst erzeugt.

Stufe für Stufe

In der nächsten Arbeitsstufe sind für die bisher definierten Module eigene Strukturdiagramme zu erstellen. Es sollen dabei nur die Strukturdiagramme dieser Hierarchie-Ebene bearbeitet werden. Die Bearbei-



Programm 4. Strukturdiagramm der Statusvariablenzuweisung

Systembeschreibung des Beispielproblems

Kurzbeschreibung:

Es soll ein Datenerfassungssubsystem für Personal Computer entwickelt werden, das die Möglichkeit sehafft, analoge Daten aufzunehmen und an den Personal Computer zu überspielen. Das System soll selbständig nach vorgegebenen Zeiten oder Triggerkriterien Daten aufnehmen, zwischenspeichern und auf Abruf an den Personal Computer zur Auswertung senden können. Weiterhin soll das System in der Lage sein, die Datenaufnahme bei einem vorgegebenen Kriterium zu beenden. Die Verbindung zum Personal Computer erfolgt über den IEC-Bus oder eine RS-232-Schnittstelle.

Die Arbeitsweise soll durch den Personal Computer voll programmierbar sein. Befehle dürfen nur ASCII-Zeichen enthalten, da nicht alle Treiber bei Personal Computern die Steuerzeichen voll unterstützen. Die Daten sollen wahlweise als ASCII-Zeichen mit programmierbarem Endekennzeichen (z. B. Ctrl-Z) oder als binärer Datensatz programmierbarer Länge an den Personal Computer übergeben werden.

Es sollen acht Kanäle mit einer Auflösung von 12 Bit abgetastet werden können. Die Abtastrate muß im Bereich von mindestens 100 µs pro Kanal bis 100 s pro Kanal einstellbar sein.

Erweiterungen auf die Linearisierung von Kurven sind denkbar.

Als Anwendungsbereich kommt die Abtastung von mechanischen Vibrationen, die Erfassung von Temperaturen, die Erfassung von Störungen an Stromversorgungsgeräten in Frage.

Festlegen der Systemfunktion:

Das System erhält analoge Daten. Die Daten setzt ein Analog-/Digitalumsetzer in binäre Werte um, die dann entsprechend den vorliegenden Vorgaben weiterverarbeitet oder abgespeichert werden.

Weitere Eingangsdaten sind die Steuersignale vom Personal Computer, die als ASCII-Zeichenkette übergeben werden.

Als Befehl wird ein Datensatz erkannt, der das Befehlskennzeichen benutzt. Alle Daten zwischen Befehlskennzeichen und Befehlsende werden als Befehl akzeptiert und vom System interpretiert. Das Auswertungssystem speichert die Befehlsdaten zwischen, bis das Befehlsende erkannt wird. Erkennt es anstelle des Ende-Kennzeichens das Abbruchkennzeichen, so ignoriert es den gesamten Befehlssatz. Mit Hilfe eines Wiederholungskennzeichens läßt sich das System auffordern, den kompletten Datensatz an den Personal Computer zurückzusenden. Erst wenn ein kompletter Datensatz vorliegt, wird der Datensatz interpretiert.

Als Rechenoperationen sind vorgesehen: Vergleich von Meßwert und Vorgabewert, Mittelwertbildung, Minimum- und Maximumbildung.

Bei der Hardware sind die entsprechenden VDE/FTZ-Bestimmungen einzuhalten.

Spezielle Anforderungen:

Für den vorgesehenen Anwendungsbereich erscheinen 10000 Abtastungen pro Sekunde ausreichend zu sein. Ein großer Teil der Anwendungen wird mit wesentlich niedrigeren Abtastraten auskommen. Im Bereich der niedrigeren Abtastraten wäre die gleichzeitige Kommunikation mit dem Personal Computer wünschenswert.

Der Speicherbereich sollte mindestens 20000 Meßwerte aufnehmen können. Im Bereich der höheren Abtastraten wird während der Messung keine Kommunikation mit dem Personal Computer möglich sein.

Laufende Aktivitäten werden durch jedwede Steuersequenz des Personal Computers unterbrochen.

Die Hardware muß in der Lage sein, die maximale Datenrate des ADU zu verarbeiten.

Zu verarbeitende Daten und Befehle:

Als Eingabedaten sind acht analoge Kanäle vorzusehen. Alle analogen Eingänge werden über einen Multiplexer abgefragt. Multiplexer und Analog-/Digitalumsetzer werden vom Rechner gesteuert, der auch die Daten übernimmt und in seinem Speicher ablegt.

Weiterhin erhält das System Daten vom Personal Computer, die über eine RS-232-Schnittstelle oder den IEC-Bus anstehen. Bei dem Entwurf des Befehlssatzes ist zu beachten, daß nicht alle Personal Computer sämtliche Control-Zeichen ausgeben können. Der Befehlssatz darf daher nur ASCII-Zeichen verwenden.

Eine Eingabe durch den Personal Computer kann die folgende Form haben:

":::35K1T10N1000K2T12N1500K3T200N3000S•••"

Damit würden die Kanäle 1, 2 und 3 mit jeweils einer Abtastzeit von 10, 12 und 200 ms voreingestellt (T), 1000, 1500 und 3000 Messungen angeordnet (N) und die Messung aktiviert (S).

Vor der Ausführung jeder Messung wird auf Wunsch ein Quittungssignal an den Steuerrechner zurückgesendet. Der Personal Computer kann die gemessenen Werte per Befehl abrufen: ":::M1N500•••". Diese Befehlssequenz ruft dann die ersten 500 Meßwerte des Kanals 1 ab.

Inhalt und Blocklänge der übertragenen Daten können per Programm vorgegeben werden. Dies ist notwendig, da manche Rechner mit einer gepufferten Eingabe arbeiten und so lange in der Eingabeschleife warten, bis der Buffer gefüllt ist.

Eine Ausgabe kann die folgende Form haben:
"+ + +K1:00001:1275;1300;. . ."
Hinter der Kanalangabe folgt einmalig im Datensatz die Adresse des ersten Meßwertes.

Es wird nur ein kompletter Block übergeben; füllen die angeforderten Daten den Block nicht aus, so wird der Rest mit 0 oder einem programmierbaren Zeichen ergänzt.

Festlegung der physikalischen Ein- und Ausgaben:

Das System wird über eine RS-232-Schnittstelle oder den IEC-Bus angesprochen. Die Übertragungsgeschwindigkeit der RS-232-Schnittstelle soll programmierbar sein, das heißt, das System wird durch DIL-Schalter auf die Baudrate des Personal Computers eingestellt. Es müssen Baudraten im Bereich von 300 bis 4800 Baud zur Verfügung stehen. Die unteren Baudraten sind zum Beispiel für Osborne beziehungsweise den C64 nötig.

Auf der Analogseite sollen 8 Eingänge vorhanden sein, die jeweils ±5 V verarbeiten können.

Die Analog-/Digitalwandlerkarte ist über einen Port mit dem Arbeitsprozessor verbunden.

Es ist zu erwarten, daß die gleichzeitige Verarbeitung von Befehlen und analogen Daten zu schwankenden Abtastzeiten führt.

Bekannte, aber ungelöste Probleme:

Nicht festgelegt sind im Moment der volle Befehlsumfang und die tatsächlichen Abtastraten. Offen bleibt auch der volle Umfang der mathematischen Arbeiten, die der Prozessor ausführen soll. Eventuell notwendige Umrechnungen der digitalisierten Daten (Skalierung) müssen auch noch geklärt werden.

Zu erwartende Erweiterungen:

Da das System auch für die Erfassung von Temperaturen eingesetzt werden kann, muß unter Umständen später eine Linearisierung von Kennlinien möglich sein.

Weiterhin ist auch eine direkte Druckerausgabe ohne den Umweg über den Personal Computer denkbar. Dabei sind auch grafische Ausgaben erwünscht (Bit 8 bei EPSON & Co.). Für diesen Zweck muß dann auch eine Bedienung direkt am System möglich sein.

Auch sollte später eine FFT (Fast Fourier Transformation) realisierbar sein.

Hard- und Softwareauswahl:

Es wird auf eine vorhandene Hardware (ADU, Einkartenprozessor, RS-232- bzw. IEC-Bus-Schnittstelle zurückgegriffen.

Die Programme sollen weitgehend in der Sprache C geschrieben werden, ein entsprechender C-Compiler steht zur Verfügung.

tung weiterer Submodul-Ebenen empfiehlt sich zu diesem Zeitpunkt noch nicht.

Als Beispiel sei hier nur das Strukturdiagramm für das Setzen der Statusvariablen vorgestellt.

Programm 4 zeigt die Realisierung der Datensatzdekodierung. Es setzt nun schon eine Reihenfolge der Parameterübergaben voraus, weiterhin wird zugelassen, daß einzelne Befehle (T oder N) nicht unbedingt vorhanden sein müssen. Der Test der numerischen Parameter auf ihre Zulässigkeit kann an dieser Stelle eingefügt oder aber auch wie im Beispiel den Meßprogrammen überlassen werden. Diese könnten eine derartige Untersuchung vornehmen, bevor die Messung beginnt. Werden die Variablen erfolgreich übernommen, so wird ein entsprechendes Quittungssignal gesendet, sonst erfolgt eine Fehlermeldung. Es ist vorteilhafter, erst alle Strukturdiagramme der gleichen Ebene zu erstellen, als von einem Modul aus gleich weiter in größere Tiefen vorzudringen. Dabei gefällte Entscheidungen, die auch andere Module betreffen, können nämlich leicht an anderer Stelle zusätzliche Probleme einhandeln. Liegen hingegen alle Strukturdiagramme einer Ebene vor, so lassen sich diese Entscheidungen mit viel mehr Übersicht treffen.

Es sind in der Regel bei diesem Verfahren etliche Änderungen zu erwarten, bis alle Strukturdiagramme der ersten zwei Stufen vorliegen. Werden diese Änderungen auf der Ebene der Schritte 1, 2 und 3 durchgeführt, so sind normalerweise nur Bleistift, Radiergummi und Texteditor notwendig. Es besteht noch nicht die Notwendigkeit, mühsam erzeugten und getesteten Programmcode zu ändern. Der Änderungsaufwand hält sich in Grenzen, und es wird ein guter Überblick über Möglichkeiten des Systems geschaffen. Eventuelle Erweiterungen können im Ansatz berücksichtigt werden und führen später nicht zu größeren Umschreibeaktionen.

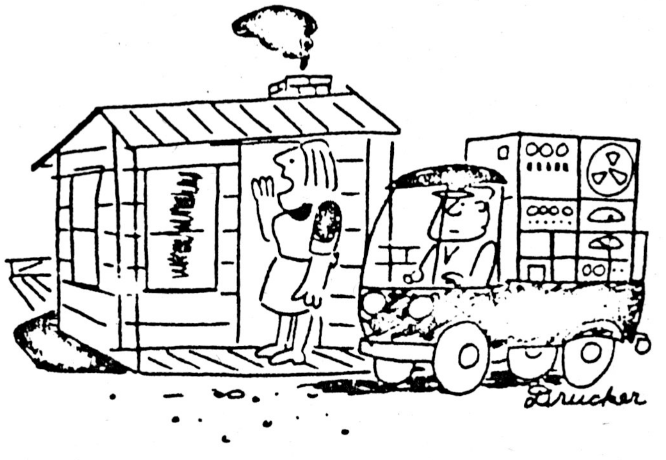
Das vorgestellte Beispiel versuchte, einen Einblick in die Arbeitsweise bei der Erstellung von Strukturierten Programmen zu geben, anhand eines Projektes im Anfangsstadium. Der Leser hat Gelegenheit, dieses Projekt je nach seinen Vorstellungen weiter zu durchdenken oder sich an einem entsprechenden Beispiel zu erproben. Es sei darauf hingewiesen, daß nur die tatsächliche Erprobung des Verfahrens eine sinnvolle Beurteilung erlaubt. Werden die Ergebnisse komplett präsentiert, so kommt man leicht zu der Meinung, daß man dies auf jeden Fall alles auch gewußt hätte — *nachdem* man es gelesen hat.

Kurz und bündig

Strukturiertes Programmieren (oder Strukturiertes Design) ist ein Verfahren zur sinnvollen Erarbeitung eines Programmes (Systems). Ziel des Verfahrens ist es, möglichst viele Entscheidungen schon vor dem Schreiben der ersten Programmzeile kritisch an den Erfordernissen des Gesamtsystems zu prüfen, kritische Punkte im voraus zu erkennen und durch Optimierung des Hardware-/Software-Einsatzes zu entschärfen.

Ein weiterer wesentlicher Punkt ist die Abschätzbarkeit des Aufwandes, die sich wesentlich verbessert, wenn wenigstens die erste und zweite Strukturdiagramm-Ebene vorliegt. Letztendlich wird die Wartung der Programme (Fehlersuche) durch die verbesserte Dokumentation wesentlich vereinfacht und damit langfristig der notwendige Software-Aufwand reduziert. Als Nachteil steht ein höherer Aufwand ins Haus, bevor die erste Programmzeile geschrieben wird. Bezogen auf das gesamte Projekt ergibt sich in der Regel aber ein deutlicher Zeitgewinn, der noch mit dem Projektumfang steigt.

Wer allerdings Spaß daran findet, mit einer Idee im Kopf den Personal Computer einzuschalten und das Programm direkt wachsen zu sehen, der soll es auch weiterhin tun — rationelles Arbeiten und Spaß an der Arbeit müssen nicht unbedingt mit den gleichen Mitteln zu realisieren sein. □



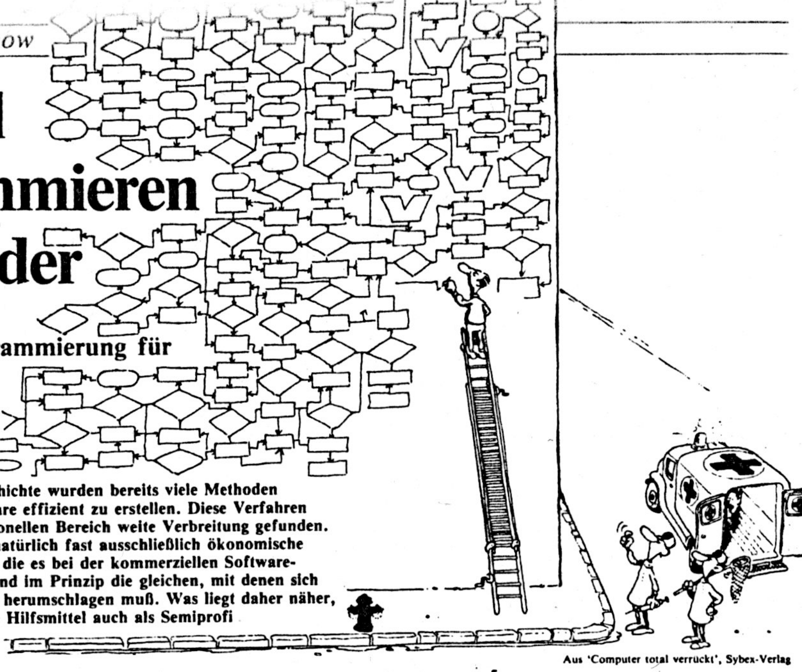
„Charlie! Wo soll der neue Computer hin?"

Planvoll programmieren kann jeder

Strukturierte Programmierung für den Hausgebrauch

Florian Sachse

Im Laufe der Computergeschichte wurden bereits viele Methoden entwickelt, um 'gute' Software effizient zu erstellen. Diese Verfahren haben vor allem im professionellen Bereich weite Verbreitung gefunden. Das hat auf diesem Sektor natürlich fast ausschließlich ökonomische Gründe, aber die Probleme, die es bei der kommerziellen Software-Entwicklung zu lösen gilt, sind im Prinzip die gleichen, mit denen sich auch der Software-Hobbyist herumschlagen muß. Was liegt daher näher, als sich diese professionellen Hilfsmittel auch als Semiprofi zunutze zu machen.



Aus 'Computer total verrückt', Sybex-Verlag

Die Strukturierte Programmierung ist eine Methode (unter anderen) zur Erstellung von Programmen, die

— einfach zu testen sind,

— leicht zu lesen sind und auf dieser Ebene bereits eine qualitative Beurteilung des Programmes zulassen (Beispiel: eine Dokumentation, die nicht ausschließlich dem Programmurheber seine Kreation ins Gedächtnis zurückruft),

— leicht zu warten sind (einfache Fehlersuche),

— die mit geringem Aufwand verändert oder erweitert werden können.

Der wohl am häufigsten beschrittene Weg der Programmentwicklung beim Hobbyisten (vermutlich aber nicht nur bei diesen) ist, ein Programm sofort in den Computer (ohne ein Stückchen Papier) nach den ersten groben Vorstellungen einzutippen. Dann wird es durch 'Hinbiegen' zum Laufen gebracht, und irgendwann werden eventuell noch ein paar Gedanken an eine Dokumentation verschwendet.

Es soll nicht behauptet werden, daß nicht auch auf diese Art und Weise Programme gemäß der aufgeführten Anforderungen erstellt werden können. Erfahrene Programmierer müßten dazu eigentlich in der Lage sein. Es soll hier aber deutlich betont werden, daß es wirklich viel Erfahrung und auch sehr viel Selbstdisziplin erfordert, mit dieser Methode anspruchsvolle — und dabei vor allem umfangreiche — Programme zu erstellen.

Wer zum Beispiel durch die besonderen 'Vorzüge' von BASIC verwöhnt ist (hier noch ein Zeilchen einfügen, dort noch ein GOTO, dann ein Kosmetik-Renumber), wird bei der ersten Verbesserung (und die gibt es eigentlich immer), sagen wir mal nach zwei Monaten des Vergessens, unter Garantie auf die Nase fallen.

Erst denken, dann tippen

Die erste und wichtigste Aufgabe, die es zu lösen gilt, besteht darin, einen vorgegebenen Algorithmus in funktionale Einheiten zu zerlegen. Diese Einheiten nennt man Aufgaben oder Funktionen (nicht zu verwechseln mit dem mathematischen Begriff einer Funktion). Um später auch wirklich zu einem strukturierten Programm zu kommen, darf diese Einteilung natürlich nicht willkürlich vorgenommen werden. Vielmehr versucht man, den Algorithmus erst einmal in grobe Teilaufgaben zu zerlegen. Jede Teilaufgabe kann natürlich ebenfalls in Unteraufgaben zerlegt werden. Von Stufe zu Stufe verlieren die Funktionen an Komplexität. Die 'schrittweise Verfeinerung' wird abgebrochen, wenn die Funktionen so einfach geworden sind, daß sie sich problemlos in eine Programmiersprache umsetzen lassen. Um nicht vom Weg zum strukturierten Programm abzukommen, sollte jede Funktion

— eine logisch abgeschlossene Aufgabe beschreiben (zum Beispiel Einlesen von Eingabedaten, Aufbau einer Bildschirmmaske),

— genau einen Eingang und einen Ausgang haben. Auf Programmebene betrachtet heißt das: Ein Unterprogramm hat nur eine Einsprungadresse, und auch der Rücksprung erfolgt nur an einer Stelle. (Das hat nicht nur formale, sondern auch rein praktische Gründe. Beim Austesten von Maschinenprogrammen bieten sich beispielsweise solche Stellen zum Setzen von Breakpoints an, da sie unter jeder Bedingung angesprungen werden. Außerdem sind häufig vor Verlassen eines Unterprogramms einige abschließende Operationen auszuführen, wie das Wiederherstellen von Registerinhalten.),

— einfach aufgebaut sein. Zu komplexe Funktionen werden durch weitere Unterteilung in Untermodule vereinfacht.

Der Algorithmus kann nun durch die Teilaufgaben beschrieben werden, in die er zerlegt worden ist. Die Funktionen auf der obersten Stufe liefern eine grobe Beschreibung des Algorithmus. Von Stufe zu Stufe wird diese Beschreibung verfeinert, bis hin zu elementaren Anweisungen, die direkt in eine Programmiersprache übertragen werden können.

Bildhaft

Nicht nur für die Programmentwicklung, sondern auch für die Programmwartung ist dieses Konzept von Nutzen. Für den ersten Überblick über ein Programmpaket genügt ein Blick auf die Hauptfunktionen. Sind Änderungen nötig, kann man sich auf die entsprechenden Teilaufgaben konzentrieren. Das setzt natürlich voraus, daß die Funktionen und die Abhängigkeiten zwischen ihnen leicht zu überblicken sind. Als grafische Hilfsmittel für die erste Planungsstufe haben sich vor allem die Hierarchie-Diagramme bewährt. Jede Funktion wird als Kästchen dargestellt, jede Abhängigkeit durch eine Verbindung. Die Grafik in Bild 1 ist zum Beispiel so zu interpretieren:

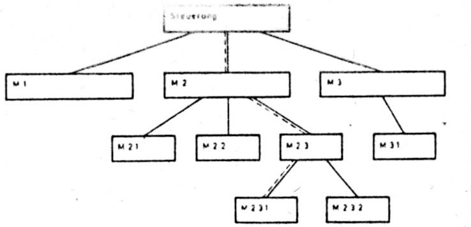


Bild 1. Hierarchie-Diagramme stellen nur die Abhängigkeiten von Programm-Modulen untereinander dar. Sie bilden die planerische Vorstufe zu Flußdiagramm oder Struktogramm.

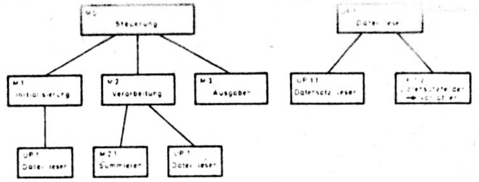


Bild 3. Eine typische Aufteilung einer Programmieraufgabe in logisch unabhängige Funktionseinheiten.

Der Algorithmus besteht aus drei Hauptfunktionen M1, M2 und M3. Die Funktion M2.3 setzt sich zusammen aus den Funktionen M2.3.1 und M2.3.2, die selbst nicht weiter unterteilt sind. Aus der Grafik wird auch ersichtlich, daß zum Beispiel M2.2 keine Unterfunktion von M1 ist, da es zwischen beiden keine Verbindung gibt.

Besonders auffallend ist die baumartige Struktur, die sich bei der schrittweisen Verfeinerung (auch Top-Down Entwicklung genannt) von allein ergibt. Werden die Funktionen in einem Programm als Unterroutinen realisiert, dann folgt aus der baumartigen Struktur, daß jede Unterroutine nur von einer 'darüberliegenden' Routine aufgerufen wird (und das häufig auch nur einmal).

So manchem Programmierer wird bei diesem Gedanken gar nicht wohl zumute sein, aber die Verletzung der baumartigen Struktur kann letztlich zu einem Gebilde führen, wie es Bild 2 zeigt. Das Diagramm sieht vielleicht auf den ersten Blick recht harmlos aus, aber schon die Frage, wie viele Pfade von Funktion A nach Funktion B führen, ist nicht mehr so leicht zu beantworten.

Funktionen, die mehr als einmal benutzt werden, sollten im Hierarchie-Diagramm für jeden Aufruf als eigenständige Funktionen (wenn auch mit gleichem Namen) dargestellt werden (Bild 3). So wird verhindert, daß im Hierarchie-Diagramm die Verbindungen kreuz und quer verlaufen. Im späteren Programm-Code tauchen die Funktionen, die mehrfach benötigt werden, natürlich nur einmal als Unterprogramm auf.

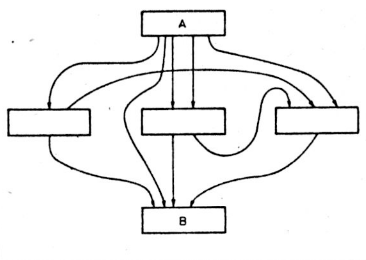


Bild 2. Bei Anwendung der Strukturierten Programmierung gibt es nur genau eine Verbindung zwischen dem aufrufenden und dem aufgerufenen Programm-Modul (wie in Bild 1). Dann können Gebilde wie dieses hier gar nicht vorkommen.

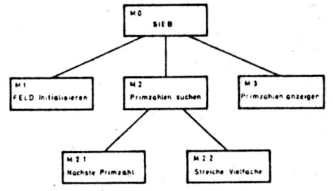


Bild 4. Ein konkretes Beispiel: So läßt sich der Algorithmus zum Ermitteln von Primzahlen (Sieb des Eratosthenes) in Funktionsblöcke aufteilen.

Sieben

Es soll das 'Sieb des Eratosthenes' zur Berechnung aller Primzahlen in einem gegebenen Intervall programmiert werden. Tabelle 1 zeigt die Arbeitsweise des Algorithmus. Auf der obersten Ebene kann der Algorithmus in drei Teilfunktionen zerlegt werden:

1. Feld initialisieren,
2. Primzahlen suchen,
3. Primzahlen ausgeben.

Die Teilfunktionen 1. und 3. können unmittelbar in Programm-Module umgesetzt werden, Funktion 2. sollte aber noch weiter verfeinert werden:

2.1. Nächste Primzahl suchen,

2.2. Alle Vielfachen streichen.

Aus dieser Einteilung ergibt sich ein Hierarchie-Diagramm wie in Bild 4.

So weit, so gut. Der Algorithmus ist in handliche Funktionen zerlegt, und die Abhängigkeiten zwischen ihnen sind auch geklärt, aber über den Kontrollfluß sagt das Hierarchie-Diagramm nichts aus. Zum Beispiel: Werden die Funktionen 2.1 und 2.2 hintereinander ausgeführt (Sequenz), oder entweder 2.1 oder 2.2 (Verzweigung), oder werden sie wiederholt durchlaufen (Schleife)?

Diese Frage läßt sich bislang nur mit Hilfe der allgemeinen Beschreibung aus Tabelle 1 beantworten: Die Funktionen 1, 2 und 3 bilden eine Sequenz, zuerst die Tabelle initialisieren, dann alle Primzahlen suchen und zuletzt die gefundenen Primzahlen ausgeben. Die Funktionen 2.1 und 2.2 werden in einer Schleife so lange wiederholt, bis das Ende der Tabelle erreicht ist.

Strukturhilfe

Mit den drei Strukturelementen Sequenz, Verzweigung und Schleife kann man nicht nur den Funktionen beschreiben, sondern auch die Funktionen selbst. Der Kontrollfluß eines

---

Das Sieb des Eratosthenes beschreibt ein Verfahren, mit dem man alle Primzahlen zwischen 1 und n bestimmen kann. Dabei macht man sich zunutze, daß die ganzzahligen Vielfachen von Primzahlen selber keine Primzahlen sind.

Man geht von einer Zahlenkette mit den ganzen Zahlen zwischen 1 und n aus.

1 2 3 4 5 6 7 8 9 … n

Da die 1 per Definition keine Primzahl ist, kann sie gestrichen werden. Ab der ersten Zahl der Reihe wird eine noch nicht durchgestrichene Zahl (Primzahl) gesucht …

~~1~~ 2 3 4 5 6 7 8 9 … n

… und auch gefunden. Die 2 ist eine Primzahl, deren Vielfachen sind selber keine Primzahlen, sie müssen also gestrichen werden.

1 (2) 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 … n

Die nächste Primzahl, die gefunden wird, ist die 3, deren Vielfachen werden ebenfalls gestrichen.

1 (2) (3) ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ … n

Danach werden die 5, 7, 11 etc. gefunden. Ist die Zahlenreihe vollständig abgearbeitet, so enthält sie nur noch Primzahlen.

Tabelle 1. Der Algorithmus zu 'Sieb des Eratosthenes'

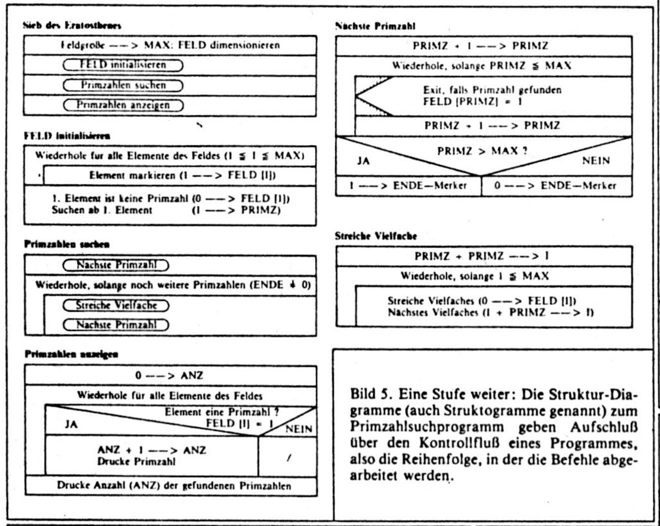


Bild 5. Eine Stufe weiter: Die Struktur-Diagramme (auch Struktogramme genannt) zum Primzahlsuchprogramm geben Aufschluß über den Kontrollfluß eines Programmes, also die Reihenfolge, in der die Befehle abgearbeitet werden.

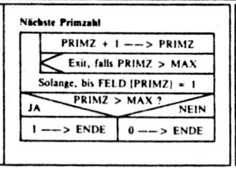


Bild 6. Verbesserungen oder Änderungen beschränken sich bei modularen Programmen meistens auf sehr wenige Module. Hier eine Alternative zum Modul 'Nächste Primzahl' aus Bild 5.

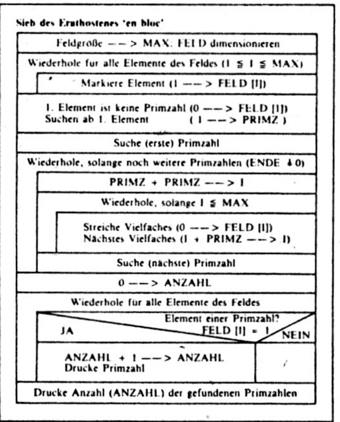


Bild 7. Module wieder einlagern: Man muß beim Strukturierten Programmieren stets einen Mittelweg zwischen Komplexität (zu viele verschiedene Funktionen in einem Modul) und Übersichtlichkeit (zu viele Blätter Papier) wählen.

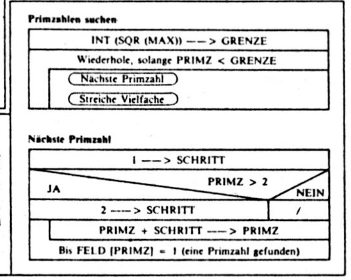


Bild 8. Die Module 'Primzahlen suchen' und 'Nächste Primzahl' lassen sich verbessern. Hier die optimierten Versionen.

Algorithmus läßt sich zum Beispiel mit einem Programmablaufplan beschreiben. Für unsere Zwecke besser geeignet sind aber Struktogramme (auch Nassi-Shneiderman-Diagramme genannt), da sie zu jedem Strukturelement eine grafische Entsprechung haben. Tabelle 2 zeigt die drei Grundkonstrukte und einige Erweiterungen im Vergleich zu Elementen des Programmablaufplans sowie deren Umsetzung in BASIC- und Pascal-Befehle.

Die Strukturblöcke haben — ebenso wie Funktionen — nur einen Ein- und einen Ausgang. Jeder mit einem Großbuchstaben gekennzeichnete Block kann wiederum einen eigenständigen Strukturblock enthalten. Aus diesen Elementen lassen sich die Struktogramme für das Primzahlproblem zusammensetzen (Bild 5). Die Umsetzung des Algorithmus in

```
1000 REMark Hauptprogramm (Steuerung der Module)
1020   MAX = 100
1030   DIM FELD(MAX)
1040   GO SUB 2000
1050   GO SUB 2200
1060   GO SUB 2400
1070 STOP
1080 :
1090 :
2000 REMark MODUL 'Feld initialisieren'
2010   FOR I = 1 TO MAX
2020     FELD(I) = 1
2030   NEXT I
2040   FELD(1) = 0
2050   PRIMZ   = 1
2060 RETurn
2070 :
2080 :
2200 REMark MODUL 'Primzahlen suchen'
2210   GO SUB 2600
2220   IF ENDE = 0 THEN GO SUB 2800:GO SUB 2600:GO TO 2220
2230 RETurn
2240 :
2250 :
2400 REMark MODUL 'Primzahlen anzeigen'
2410   ANZAHL = 0
2420   FOR I = 1 TO MAX
2430     IF FELD(I) = 1 THEN ANZAHL = ANZAHL + 1:PRINT I
2440   NEXT I
2450   PRINT
2460   PRINT "Zwischen 1 und ";MAX;" gibt es ";ANZAHL;" Primzahlen"
2470 RETurn
2480 :
2490 :
2600 REMark MODUL 'Naechste Primzahl'
2610   PRIMZ = PRIMZ + 1
2620   IF PRIMZ > MAX THEN GO TO 2660
2630     IF FELD(PRIMZ) = 1 THEN GO TO 2660
2640     PRIMZ = PRIMZ + 1
2650   GO TO 2620
2660   ENDE = 0
2670   IF PRIMZ > MAX THEN ENDE = 1
2680 RETurn
2690 :
2700 :
2800 REMark MODUL 'Streiche Vielfache'
2810   I = PRIMZ + PRIMZ
2820   IF I <= MAX THEN FELD(I) = 0:I = I + PRIMZ:GO TO 2920
2830 RETurn
```

Programm 1. Auch in BASIC kann man wunderbar strukturiert programmieren. Dieses Programm korrespondiert mit Bild 5.

```
2600 REMark MODUL 'Naechste Primzahl'
2610.  PRIMZ = PRIMZ + 1
2620     IF PRIMZ > MAX THEN GO TO 2640
2630   IF FELD(PRIMZ) <> 1 THEN GO TO 2610
2640   ENDE = 0
2650   IF PRIMZ > MAX THEN ENDE = 1
2660 RETurn
```

Programm 2. Die alternative Lösung zum Struktogramm Bild 6.

BASIC (Programm 1) zeigt einige typische Besonderheiten:

Um die Logik der Struktogramme nicht zu verändern, kommt im Modul 'Nächste Primzahl' die Anweisung 'PRIMZ = PRIMZ + 1' zweimal vor (Zeilen 2610 und 2640). Ohne die Logik des Algorithmus zu verletzen, könnte auf eine der beiden Anweisungen verzichtet werden. Das führt aber natürlich zu einer anderen Definition der Funktion 'Nächste Primzahl' (Bild 6, Programm 2).

Die Schleifen in 'Nächste Primzahl' und 'Streiche Vielfache' werden im BASIC-Programm nicht durch eine FOR-Schleife gebildet, obwohl dies eigentlich naheläge. Das Problem besteht

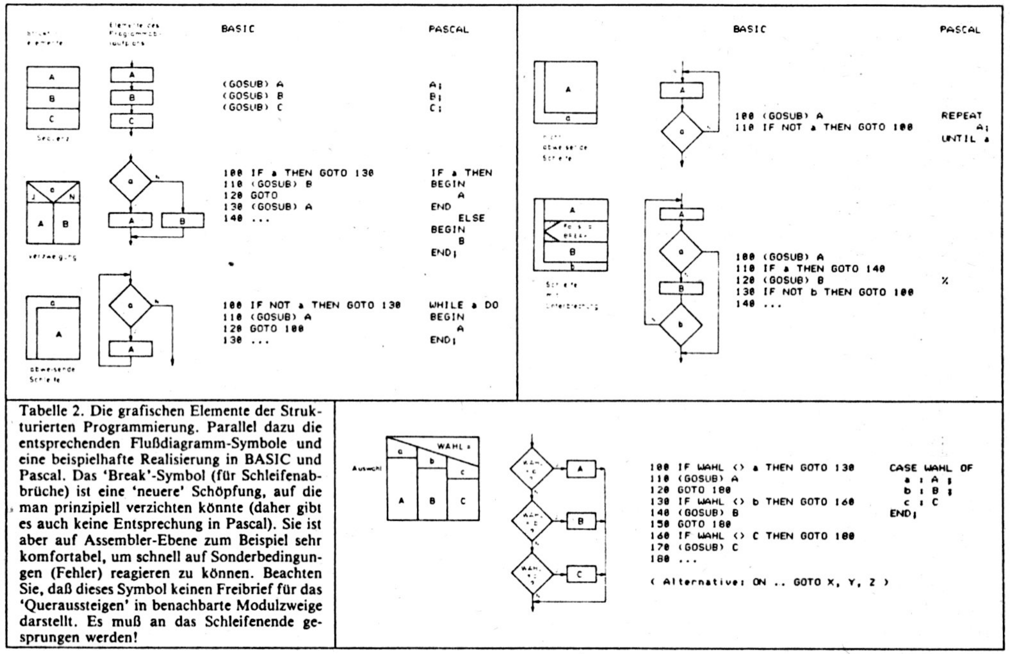


Tabelle 2. Die grafischen Elemente der Strukturierten Programmierung. Parallel dazu die entsprechenden Flußdiagramm-Symbole und eine beispielhafte Realisierung in BASIC und Pascal. Das 'Break'-Symbol (für Schleifenabbrüche) ist eine 'neuere' Schöpfung, auf die man prinzipiell verzichten könnte (daher gibt es auch keine Entsprechung in Pascal). Sie ist aber auf Assembler-Ebene zum Beispiel sehr komfortabel, um schnell auf Sonderbedingungen (Fehler) reagieren zu können. Beachten Sie, daß dieses Symbol keinen Freibrief für das 'Queraussteigen' in benachbarte Modulzweige darstellt. Es muß an das Schleifenende gesprungen werden!

darin, daß FOR-Schleifen, je nach Interpreter, abweisende (zum Beispiel Sinclair) oder nichtabweisende Schleifen (zum Beispiel Apple BASIC) bilden. Abweisende und nichtabweisende Schleifen verhalten sich im großen und ganzen gleich: außer wenn schon vor dem ersten Eintritt in die Schleife die Schleifenbedingung nicht erfüllt ist.

In diesem Falle wird die abweisende Schleife überhaupt nicht durchlaufen. Im Gegensatz dazu wird die nichtabweisende Schleife auf jeden Fall einmal durchlaufen, da die Schleifenbedingung erst am Ende des Blocks abgefragt wird. An all jenen Stellen, wo im Programm eine abweisende Schleife unbedingt notwendig ist (um zum Beispiel Feldüberschreitungen zu verhindern), sollten deshalb FOR-Schleifen nur mit Bedacht eingesetzt werden.

In dieser ersten Lösung sind die einzelnen Funktionen, die ja durch das Hierarchie-Diagramm vorgegeben sind, als eigene Struktogramme beziehungsweise Unterprogramme umgesetzt worden. Die Programm-Module müssen aber nicht unbedingt 'physikalisch' (neues Blatt Papier) ausgelagert werden. Wichtig ist nur, daß sie logisch abgeschlossen sind (genau einen Ein-/Ausgang haben). Ohne die Prinzipien der Strukturierten Programmierung zu verletzen, können deshalb die 'Unter-Struktogramme' an den Stellen im Struktogramm eingefügt werden, wo sie sonst aufgerufen würden.

Als Beispiel zeigt Bild 7 die zweite Lösung mit einem zusammengefaßten Struktogramm. Nur das Untermodul zum Suchen der nächsten Primzahl bleibt ausgelagert, da es an zwei verschiedenen Stellen aufgerufen wird.

Interessanter als das 'Einlagern' von Modulen ist natürlich das 'Auslagern'. Ein wesentlicher Vorteil der Strukturierten Programmierung ist es, daß Programmteile, die im Laufe der Entwicklung und Anpassung immer umfangreicher geworden sind, einfach ausgelagert werden können, ohne daß sich die Programmlogik ändert. Der modulare Aufbau läßt sich aber auch auf andere Weise nutzen. Programmteile können ohne große Probleme durch andere oder gleichwertige Module ersetzt werden. Voraussetzung ist natürlich, daß die Datenschnittstellen (Parameterübergabe und Speicherdefinition) übereinstimmen. So arbeitet das Alternativ-Modul aus Bild 6, eingebunden in das Primzahlprogramm, problemlos.

Der Parameterübergabe zwischen Programm-Modulen muß man vor allem auf Assembler-Ebene sehr viel Aufmerksamkeit schenken, da diese oft über Register abgewickelt wird. Wer zum Beispiel nicht von vornherein festlegt, welche Register verändert werden dürfen und welche Register für Übergaben an untergeordnete beziehungsweise übergeordnete Module verwendet werden sollen, der sollte sehr intensiv dokumentieren. Es empfiehlt sich, diese sogenannte Software-Schnittstellen-Beschreibung jedem Modul — sowohl auf Struktogrammebene als auch auf Code-Ebene — in Form von expliziten Kommentaren voranzustellen.

Besser als bei Programmablaufplänen lassen sich in Struktogrammen Programmteile auffinden, die häufig durchlaufen werden (Schleifen) und bei denen Verbesserungen am meisten fruchten. Das Modul kann zu Testzwecken separat untersucht werden, zum Beispiel ob sich vielleicht Teile davon optimieren oder die Anzahl der Schleifendurchläufe verringern lassen.

Da sich die Veränderungen nur auf das Modul beziehen, braucht man um die Logik des gesamten Programms nicht zu bangen, solange die Logik des optimierten Moduls erhalten bleibt. Außerdem läßt sich die Korrektheit eines veränderten Moduls immer noch leichter überprüfen als die eines ganzen (womöglich unstrukturierten) Programms.

Modulweise optimieren

Auch unser Primzahlprogramm läßt sich natürlich verbessern. Schon Eratosthenes war auf die Idee gekommen, daß man nicht das ganze Feld nach Primzahlen durchsuchen muß. Wenn man nämlich alle Primzahlen zwischen 1 und Wurzel(n) gefunden und deren Vielfache markiert hat, kann

die Suche nach weiteren Primzahlen abgebrochen werden. Alle restlichen Zahlen im Feld, die noch nicht markiert sind, sind Primzahlen. Auch das Suchen nach der nächsten Primzahl läßt sich beschleunigen. Alle Primzahlen größer als zwei sind ungerade. Da man nur ungerade 'Kandidaten' als Primzahlverdächtige ins Auge fassen muß, kann man in Zweier-Schritten das Feld durchsuchen und gerade Zahlen überspringen.

Die erste Verbesserung wirkt sich hauptsächlich auf das Modul 'Primzahlen suchen' aus. Dort ändert sich die Schleifenbedingung. Da die Ende-Meldung aus dem Modul 'Nächste Primzahl' nicht mehr benötigt wird, kann dieses Modul sogar vereinfacht werden. Die zweite Änderung bezieht sich ausschließlich auf das Modul 'Nächste Primzahl'. Bild 8 und die Programme 4 und 5 zeigen die optimierten Module.

```
130 REMark MODUL 'Feld initialisieren'
140 FOR I = 1 TO MAX
150   FELD(I) = 1
160 NEXT I
170 FELD(1) = 0
180 PRIMZ   = 1
190 :
200 REMark MODUL 'Primzahlen suchen'
210 GO SUB 420
220 IF ENDE = 1 THEN GO TO 320
230   :
240   REMark MODUL 'Streiche Vielfache'
250   I = PRIMZ + PRIMZ
260   IF I <= MAX THEN FELD(I) = 0:I = I + PRIMZ:GO TO 260
270   :
280   GO SUB 420
290   :
300 GO TO 220
310 :
320 REMark MODUL 'Primzahlen anzeigen'
330 ANZAHL = 0
340 FOR I = 1 TO MAX
350   IF FELD(I)=1 THEN ANZAHL = ANZAHL + 1:PRINT I
360 NEXT I
370 PRINT
380 PRINT "Zwischen 1 und ";MAX;" gibt es ";ANZAHL;" Primzahlen"
390 :
400 STOP
410 :
420 REMark MODUL 'Naechste Primzahl'
430 PRIMZ = PRIMZ + 1
440 IF PRIMZ > MAX THEN GO TO 480
450   IF FELD(PRIMZ) = 1 THEN GO TO 480
460   PRIMZ = PRIMZ + 1
470 GO TO 440
480 ENDE = 0
490 IF PRIMZ > MAX THEN ENDE = 1
500 RETurn
```

Programm 3

```
2200 REMark MODUL 'Primzahlen suchen'
2210   GRENZE = INT(SQRT(MAX))
2220   IF PRIMZ < GRENZE THEN GO SUB 2600:GO SUB 2800:GO TO 2220
2230 RETurn
```

Programm 4

```
[illegible]   IF PRIMZ [illegible]
2620   PRIMZ = PRIMZ [illegible]
2640   IF FELD(PRIMZ) [illegible] 1 THEN GO TO 2630
2650 RETurn
```

Programm 5

Die Programme 3 bis 5 zeigen die teilweise integrierte Programmlösung nach Bild 7 und 8, ebenfalls wieder in BASIC kodiert.

Aber auch wenn es darum geht, die Korrektheit eines Programms zu überprüfen, bietet die Strukturierte Programmierung Vorteile. Das beginnt schon bei den Struktogrammen, die die Logik des Programms besser veranschaulichen als Programmablaufpläne.

Während in Struktogrammen der Kontrollfluß sehr differenziert dargestellt werden kann, sind die Darstellungsmöglichkeiten in den Programmablaufplänen eher bescheiden. Sowohl die Bedingungen als auch die Schleifen werden durch die Verzweigung realisiert. Das kann natürlich beim Analysieren eines Programmablaufplans zu Interpretationsproblemen führen. Außerdem werden, im Gegensatz zu Struktogrammen, Operationen und den Programmfluß steuernde Anweisungen vermengt.

Es sind ja nur die Anweisungen, also zum Beispiel Zuweisungen oder Rechenoperationen, die zum Programmergebnis führen. Die kontrollierenden Anweisungen steuern 'nur' die Reihenfolge, in der die Anweisungen ausgeführt werden müssen. Dies wird bei Struktogrammen besonders deutlich: Die Blöcke enthalten die Operationen und sind überlagert von Kontrollanweisungen, die die Abarbeitung steuern.

Ein anderes Problem, wenn es um die Analyse eines Programms geht, ist der Unterschied zwischen dem statischen Programmtext und dem dynamischen Verhalten des Programms während der Ausführung. Durch die Einschränkungen der Kontrollstrukturen besteht jedoch ein enger Zusammenhang zwischen der statischen und dynamischen Struktur eines Programms. Anders kann es bei einem nicht strukturierten Programm aussehen, wo wilde Sprünge das dynamische Verhalten in ein 'gordisches Chaos' (siehe Zeichnung am Anfang des Artikels) verwandeln können. Es läßt sich dann auch nicht mehr an Hand des Programmtestes oder Programmablaufplans entwirren.

Bei der Analyse eines strukturierten Programms kann man, wie bei der Programmentwicklung, 'Top-Down' arbeiten. Man geht dabei davon aus, daß die Module auf den logisch tieferen Ebenen korrekt sind und überprüft erst, ob das zu untersuchende Modul diese Untermodule auch wirklich so einsetzt, daß das richtige Ergebnis berechnet wird. Im Fehlerfalle werden die Untermodule, jedes für sich, genauso analysiert.

Diesen Vorteilen der Strukturierten Programmierung, die sich mit wachsender Programmkomplexität noch verstärken, stehen natürlich auch einige Nachteile gegenüber, die hier nicht verschwiegen werden sollen:

Programmiersprachen wie BASIC und Entwicklungsmethoden wie die Programmablaufpläne, die die Strukturierte Programmierung nicht unterstützen, fordern vom Programmierer einiges an Selbstdisziplin, damit er nicht vom 'strukturierten' Weg abkommt. Deshalb ist der umgekehrte Weg, nämlich mit Struktogrammen zu arbeiten, wesentlich sinnvoller. Denn deren Anwendung läßt die Entstehung unstrukturierter Programme nicht zu! (Siehe auch 'Keine Meinung zur Struktur?')

Häufig muß die Modularität eines Programms mit mehr Programm-Code oder mit längeren Programmlaufzeiten erkauft werden. Dies hängt in erster Linie damit zusammen, daß auf Bedingungen nicht immer an den Stellen im Programm reagiert werden darf, wo sie auftreten.

So darf im Beispielprogramm in Bild 5 nicht einfach vom Modul 'Nächste Primzahl' aus in das Modul 'Primzahlen anzeigen' verzweigt werden, nachdem das ganze Feld abgearbeitet worden ist. Statt dessen wird die Bedingung in einem 'Merker' festgehalten (hier die Variable ENDE), um dann an passender Stelle abgefragt zu werden (Zeile 2010, Programm 1).

Neuerdings greifen immer mehr Programmierer zur Hochsprache (Pascal, C), also zu Compilern, um maschinennahe Probleme zu lösen. Damit handelt man sich allerdings Laufzeiterhöhungen (auch bei C-Compilern) von Faktor 10 bis 100 (bei sehr kleinen Programmen) im Vergleich zur Assembler-Codierung ein. Daß diese Software-Werkzeuge dennoch immer größeren Anklang finden (Turbo-Pascal zum Beispiel), zeigt, daß die Mikroprozessor-Programmierer der ersten Stunde langsam von einer neuen Generation abgelöst werden.

In einer Zeit, in der Speicher immer größer und ebenso wie die Prozessoren immer schneller werden, kann man sich planvolles Programmieren leisten. Probieren Sie es doch mal. Aller Anfang ist zugegebenermaßen schwer, vor allem, wenn man es vorher anders gelernt hat. Hat man's aber erstmal drauf, kann man über Spaghetti-Code und Trickprogrammierung nur noch milde lächeln. □

Literatur:

[1] Arno Schulz
Methoden des Softwareentwurfs und strukturierte Programmierung
de Gruyter 1978
ISBN 3-11-007336-6

[2] Gewald, Haake, Pfadler
Software Engineering
R. Oldenbourg Verlag 1979
ISBN 3-486-21492-6

[3] Gerhard Platz
Methoden der Softwareentwicklung
Carl Hanser Verlag 1983
ISBN 3-446-13040-3

[4] Sneed
Softwareentwicklungsmethoden
Verlagsgesellschaft
Rudolf Müller GmbH 1980
ISBN 3-481-36271-4

[5] Frank Heubach
Strukturierte Programmierung auch bei Mikrocomputern
ELEKTRONIK 1977
Heft 10, S. 113–118

Keine Meinung zur Struktur?

Detlef Grell

Die Strukturierte Programmierung ist eigentlich mehr als nur eine Methode zur Programmentwicklung, fanatische Anhänger sprechen deshalb auch bereits von einer Philosophie. Daher kann man sich über dieses Thema auch unglaublich verbreiten, und Kenner der Materie werden (trotz zweier Beiträge in diesem Heft) sicherlich Aspekte vermissen, die sie für besonders wichtig oder gar ausschlaggebend für Anwendung (oder auch Nichtanwendung) der Strukturierten Programmierung halten. Anregungen und Ergänzungen seitens der Leser sind uns daher sehr willkommen.

Lassen Sie mich an dieser Stelle einigen grundlegenden Mißverständnissen über das, was 'Strukturierte Programmierung' sei, zu Leibe rücken. Leicht hysterische Struktur-Fans vermuten ja bereits Rufmordkampagnen, so geballt schlägt ihnen gelegentlich die Ablehnung der Strukturierten Programmierung vor allem aus Kreisen der 'Maschinen-Code-Quetscher' und 'Hacker' entgegen. Läßt man sich dann eine Kurzdefinition dessen, was für Strukturierte Programmierung gehalten wird, geben, stößt man auf faszinierendes Halbwissen. In diesem Sinne also:

Was ist überhaupt Strukturierte Programmierung? Oder andersherum gefragt: 'Kann es überhaupt ein unstrukturiertes Programm, also ein Programm ohne irgendwie geartete Struktur geben?' Selbstverständlich hat jedes Programm eine Struktur, und wenn wir in diesem Heft von Strukturierter Programmierung reden, dann meinen wir die Programmentwicklungstechnik, die die Herren Nassi und Shneidermann sich darunter vorstellen. Der Begriff 'Strukturierte Programmierung' ist der Name für ein ganz konkret definiertes Verfahren (deswegen schreiben wir auch das Adjektiv groß), und man sollte Wendungen wie 'das Programm hat eine gute oder schlechte Struktur' vermeiden, weil sie ganz einfach falsch sind.

Im Sinne der Strukturierten Programmierung kann es nämlich nur Programme geben, die sich an die Vorschriften dieses Verfahrens halten (dann sind sie 'strukturiert') oder aber nicht — und dann sind sie eben 'nicht strukturiert'.

Im Grunde genommen besteht der wesentliche Teil der Strukturierten Programmierung aus der regelgerechten Anwendung der Struktursymbole von Nassi-Shneidermann. Und schon geht das Jammern los: 'Das sind doch diese blöden, unverständlichen Kästen'. Nein, liebe Leser, da muß ich ernsthaft widersprechen:

Mit diesen Symbolen (die keinen Deut komplizierter sind als die gemeinhin gepriesenen Flußdiagramme), ist den Herren Nassi und Shneidermann nämlich ein echter Geniestreich gelungen. Zunächst haben sie nachgewiesen, daß sich mit nur drei Darstellungsformen (Sequenz, Schleife, Verzweigung) jedes programmtechnische Problem lösen läßt.

Aber viel wichtiger ist die Konsequenz, die sich aus den von Nassi-Shneidermann geschaffenen Symbolen ergibt:

Bei der Anwendung dieser Struktursymbole ist es definitiv unmöglich, ein nichtstrukturiertes Programm (im Sinne der Erfinder) zu erstellen. Man kann das natürlich bei Kenntnis der zugrundeliegenden Regeln auch auf andere Weise erreichen. Das ist aber weitaus schwieriger, für einen Programmieranfänger eigentlich sogar unmöglich. Deswegen kann man die Einarbeitung in den Umgang mit Struktursymbolen nur wärmstens empfehlen, denn das hält Sie immer auf dem richtigen Weg.

Oft hört man die Behauptung, man könne nur in ganz bestimmten Programmiersprachen strukturiert programmieren, vornehmlich in Pascal. BASIC sei völlig ungeeignet, und immer wieder geistert das Schlagwort vom angeblichen 'GOTO-Verbot' in der Strukturierten Programmierung umher.

Erstens: In absolut jeder Programmiersprache kann man strukturiert programmieren. Das läßt sich einfach 'beweisen': In jeder Programmiersprache kann man Sequenzen, Verzweigungen und Schleifen (wie bequem oder umständlich auch immer) programmieren. Um das zu zeigen, sind die Beispiele im vorangegangenen Beitrag auch extra in BASIC geschrieben. Es gibt auf der anderen Seite natürlich Programmiersprachen, die die Strukturierte Programmierung wesentlich vereinfachen beziehungsweise gar nichts anderes zulassen. Zu den sehr förderlichen Sprachen gehört ganz sicher Pascal, denn ALGOL, Pascals Ursprung, wurde im Prinzip für die Strukturierte Programmierung entwickelt.

Zweitens: Direkte Sprünge, also vor allem das umstrittene GOTO, benötigt man in Pascal eigentlich gar nicht. Man kann im Prinzip mit den verfügbaren Kontrollstrukturen auf Struktogramm-Ebene programmieren. Ein Pascal-Lehrer kann es sich somit erlauben, ein 'GOTO-Verbot' auszusprechen, um seine Schützlinge vor einem Abgleiten auf die 'schiefe Bahn' zu bewahren.

Und so unrecht hat er nicht, denn der Mißbrauch der direkten Sprünge ist des Pudels Kern: Modulwechsel quer und rückwärts machen natürlich viel Freude. Kann man doch Programme damit so schön unlesbar und sich als Programmentwickler schnell unentbehrlich machen.

Dennoch, der Gebrauch der direkten Sprünge ist auf Assembler-Ebene natürlich unverzichtbar, wie sollte man sonst Schleifen oder Verzweigungen kodieren? Wenn man sich aber beim Einsatz von Sprüngen darauf beschränkt, nur die in sich abgeschlossenen Konstrukte der Strukturierten Programmierung nachzubilden, ist das völlig legitim.

Kommen wir zum letzten Punkt, bei dem sich die Gemüter am erbittertsten erhitzen:

Strukturierte Programme sind viel zu lang, umständlich und vor allem langsam. Das sind die Lieblingsargumente der code-quetschenden High-Speed-Low-Memory-Fetischisten, wenn es darum geht, die Einführung der Strukturierten Programmierung abzuwimmeln.

Die Zeiten 'kurzer Programme mit allen Mitteln' sind vorbei. Ob ein Programm ein oder zwei Kilobyte lang ist, ist heutzutage weitgehend irrelevant, außerdem wird man beispielsweise auf Assembler-Ebene kaum mehr als 20 Prozent Diskrepanz erhalten.

Die Frage, wann ein Programm elegant und wann es umständlich ist, läßt sich wohl nicht objektiv beantworten. Es bleibe jedem selbst überlassen, ob er sich lieber durch hundert verschlungene Befehle durchbeißt oder lieber hundertzwanzig klar gegliederte Befehle auf Anhieb nachvollzieht.

Strukturierte Programme werden geringfügig langsamer und länger, wenn man weitgehend gleiche Algorithmen und Kodierungen zugrunde legt. Es kommen üblicherweise einige verschachtelte Unterprogrammaufrufe hinzu, bei denen hauptsächlich die Parameterübergaben und gelegentlich mehrfaches Testen desselben Flags Zeit kosten. Zum Teil verliert man auch Zeit, weil bestimmte Befehle nicht verwendet werden dürfen.

Hier sollte sich der Programmierer gefordert sehen, durch eine geeignete Programmanlage und die Verwendung optimierter Algorithmen die Strukturierte Lösung zu beschleunigen. Es gilt also, 'bewußtseinsverändernd' zu wirken. Sicher, planvolles, nachvollziehbares Programmieren (die Strukturierte Programmierung ist ja nicht das einzige Verfahren dazu) kostet auch Schweiß fernab von der Tastatur. (Manche Programmierer bekommen richtiggehend Terminal-Entzugs-Ausschlag, wenn man ihnen mit solchen praxisfernen Vorstellungen kommt.) Ich neige immer mehr zu der Unterstellung, daß sich oft psychologische Barrieren (ich vermeide bewußt das Wort 'Bequemlichkeit') hinter den Argumenten der Struktur-Gegner verbergen.

Vielleicht hilft ein bißchen Motivation? Raffiniertes Trickprogrammieren ist out. Selbstmodifizierender Code ist ätzend. Sprünge per RETurn über einen gefälschten Stack sind einfach widerlich! Oder etwa nicht? □

Programmieren heißt mit Dateien arbeiten

Basic kennt in seiner Grundausstattung bei der Dateiverwaltung eigentlich nur den sequentiellen oder den wahlfreien Zugriff. Alle anderen Verfahren muß man selber stricken oder als Zusatzprogramm kaufen.

Das englische Wort File wird laut Wörterbuch mit Akte, Ordner oder Ablage übersetzt. Der Computer-Begriff File ist damit etwas umfassender als das deutsche Wort Datei. Es gibt unterschiedliche Typen von Files, wie zum Beispiel auch Ordner und Schnellhefter sich unterscheiden. Nicht der Inhalt bestimmt den Filetyp auf einer Diskette, sondern die Art, wie die Daten abgelegt sind.

In einem Büro kann die Ablage ein Karton sein, in den alle Schreiben, so wie sie kommen, einfach hineingestapelt werden. Es dürfen aber auch Aktenschränke mit Schubladen und Ordnern sein, alles mit viel System. Es ist einzusehen, daß die erste Art der Ablage sehr billig ist und wenig Raum kostet, die zweite hingegen schon kräftig ins Geld geht. Ganz anders sieht es aber bei der Zugriffszeit aus. Methode eins zwingt, den Papierstapel von Anfang bis Ende zu durchsuchen, jedenfalls so weit, bis das gewünschte Schreiben gefunden ist. Bei Methode zwei genügen wenige Handgriffe. In der EDV nennt man die beiden Verfahren »Sequentiell« und »Random«.

Sequentielle Files sind typisch für endlose Medien wie Lochstreifen oder Magnetbänder, sie kommen aber auch auf Disketten vor.

Sinnvoll sind sie sicherlich bei Programmen, denn ein solches wird immer im ganzen aufgezeichnet beziehungsweise geladen. Und damit hätten wir ein Kriterium für die Anwendung von sequentiellen Files auf Disketten: Immer dann, wenn alle Daten im Stück in den RAM zu laden sind (und natürlich da auch hineinpassen). Nun ist es leider nicht so, daß man sagen kann: »Bei großen Files wähle ich also Random«.

Ein Random-File stellt man sich am besten wie ein Buch vor. Für jeden Eintrag (Record genannt) ist eine Seite reserviert. Der Zugriff erfolgt über die Seiten-(Record-)Nummer, wobei das System auf Anhieb die richtige Seite aufschlägt. Die Seitengröße gilt für das ganze Buch. Da aber die »Schriftgröße« konstant ist, müssen Sie die Seiten so groß halten, daß der längste Eintrag auf eine Seite paßt. Der Nachteil: Alle anderen Seiten sind mehr oder weniger leer. Random-Files kosten viel Platz. Außerdem entbindet die Random-Organisation den Programmierer nicht von einer unangenehmen Aufgabe. Er muß wissen, unter welcher Record-Nummer was abgelegt ist, beziehungsweise das System der Ablage im Programm definieren und einige Suchhilfen zur Verfügung stellen.

Keine echten Random-Files

Eine Diskette ist in konzentrische Spuren (Tracks) unterteilt und diese wiederum in Sektoren. Im Directory (Inhaltsverzeichnis) wird notiert, welche Tracks und Sektoren jeweils zu welchem File-Namen gehören. Meistens werden (um die Anzahl der Notizen kleiner zu halten) mehrere Sektoren zu einem Block zusammengefaßt. Diese sind der Reihe nach, beginnend auf dem ersten Track von 1 bis n durchnumeriert. Folglich führt die Blocknummer, dividiert durch die Anzahl Blöcke je Track, sehr einfach auf die Track-Nummer.

Im Directory werden zu jedem File die Block-Nummern notiert und zwar in der logischen Folge. Diese kann infolge von mehrmaligem Aufzeichnen, Löschen und Vergrößern bestehender Files alles andere als seriell sein. Möglich ist zum Beispiel:

— File 1 belegt die Blöcke 3, 5, 7
— File 2 belegt die Blöcke 1, 2, 4, 8

So gesehen wird also ein File auf der Diskette immer sequentiell aufgezeichnet, in einem Block nach dem nächsten, so dieser gerade frei ist. Der »große« Unterschied zu Random ist gar nicht so groß. Die Blöcke werden rein rechnerisch in Records eingeteilt, deren Größe können Sie wählen, meistens jedoch mit der Einschränkung ≤256 Bytes (der physikalischen Größe eines Sektors). Wenn Sie nun einen Record über seine Nummer ansprechen, dann fängt das System an zu rechnen: Record-Nummer mal Record-Länge dividiert durch Blockgröße ergibt die relative Block-Nummer. Von da aus läßt sich dann mit »einem Blick ins Directory« die absolute Block-Nummer ermitteln, über die Blockgröße kommt man so zum Sektor und von da über die Record-Größe auf die gesuchten Daten.

Es gibt noch einen zweiten parallelen Weg. Die Bytes eines Files werden von 0 bis n (n = File-Länge-1) durchnumeriert. Dazu wird beim Öffnen des Files der sogenannte Filepointer, auch RB (Relatives Byte) genannt auf Null gesetzt. Bei sequentiellen Files wird dieser Filepointer beim Lesen/Schreiben eines jeden Bytes um 1 erhöht. Bei Random-Files wird lediglich aus Record-Nummer mal Record-Länge das relative Byte (der Filepointer-Wert) errechnet. Das, dividiert durch die Blockgröße, ergibt wieder die Block-Nummer und weiter geht's, wie schon geschildert. Die Sache mit dem Filepointer hat einige Vorteile, zum Beispiel ist damit der Zugriff auf jedes Byte eines Files möglich, aber erst einmal zur Frage »echtes Random?«

Der Zugriff auf einen Record eines Random-Files läuft, wie schon angedeutet, über einige Umwege.

Der Zugriff beginnt beim Directory, von dort geht's zum sogenannten File-Entry oder über einige Sektoren mit der sogenannten Track/Sektor-Liste zum eigentlichen Sektor, der den Record hält. Dazwischen läuft noch einiges an Mathematik und Zugriff auf Tabellen. Bei einer echten Random-Organisation werden die Adressen aller Records auf einer Spur notiert, die sich das System natürlich merkt, wenn der File geöffnet wird. Das heißt nach spätestens einer Umdrehung der Disk ist die Adresse eines Records bekannt.

Apple II	TRS-80/Genie
Schreiben von sequentielle Files auf die Disk	
10 D$-CHR$(4)	
20 PRINT D$;"OPEN TESTFILE"	20 OPEN "O",1,"TESTFILE"
30 PRINT D$;"WRITE TESTFILE"	
40 PRINT"einige Daten"	40 PRINT #1,"Einige Daten"
50 PRINT D$;CLOSE TESTFILE"	50 CLOSE 1
Lesen von der Disk	
10 D$-CHR$(4)	
20 PRINT D$;"OPEN TESTFILE"	20 OPEN "I",1,"TESTFILE"
30 PRINT D$;"READ TESTFILE"	
40 INPUT A$	40 INPUT #1,A$
50 PRINT D$;CLOSE TESTFILE"	50 CLOSE 1

Bild 1. Lesen und Schreiben von sequentiellen Files

Apple II	TRS-80/Genie
Schreiben von Random-Files auf die Disk (hier in den 3. Record)	
10 D$-CHR$(4)	
20 PRINT D$;"OPEN TESTFILE L12"	20 OPEN "D",1,"TESTFILE"
30 PRINT D$;"WRITE TESTFILE R3"	30 FIELD 1, 12 AS A$
	35 LSET A$-"Einige Daten"
40 PRINT"einige Daten"	40 PUT 1,3
50 PRINT D$;CLOSE TESTFILE"	50 CLOSE 1
Lesen von der Disk	
10 D$-CHR$(4)	
20 PRINT D$;"OPEN TESTFILE L12"	20 OPEN "R",1,"TESTFILE"
30 PRINT D$;"READ TESTFILE R3"	30 FIELD 1, 12 AS A$
40 Input A$	40 GET 1,3
50 PRINT D$;CLOSE TESTFILE"	50 CLOSE 1

Bild 2. Bei Random-Files sammelt der Apple Pluspunkte

Dieser Komfort ist auf Personal Computern nicht üblich. Dort ist ein Random-File nichts weiter als ein speziell verwalteter sequentieller File. Diesen Nachteil werden wir noch zu unserem Vorteil nutzen, zuerst aber noch etwas Technik.

Der OPEN-Befehl ist aber leider unvermeidbar gefährlich. Ihm sollte möglichst bald ein CLOSE folgen. Schauen wir uns an, warum. Obwohl von der Diskette virtuell (scheinbar) immer nur ein Byte nach dem anderen gelesen wird, sieht es praktisch anders aus. Der FDC (Floppy Disk Controller) kennt nur den Befehl »ganzen Sektor lesen«. Dazu muß ihm das DOS (Disk Operating System) sagen, wo im RAM die zum Beispiel 256 Bytes abzulegen sind. Im Falle Schreiben will der FDC wissen, ab welcher RAM-Adresse die Daten zu finden sind. Diesen RAM-Bereich nennt man Sektor-Puffer. Tatsächlich arbeitet das DOS immer nur auf diesen Puffer. Ist er voll, wird er im Stück auf die Disk kopiert beziehungsweise im Falle »Lesen« woandershin (im RAM) übertragen. Doch der Sektor-Puffer allein reicht nicht aus. Zusätzlich wird je File ein zweiter Puffer angelegt, in dem zum File gehörige Directory-Informationen gehalten werden. Beide Puffer zusammen heißen DOS-Buffer.

Für jeden offenen File benötigt man einen DOS-Puffer, meist zwei und mehr, zum Beispiel wenn man einen File lesen und gleich auf einen anderen schreiben (kopieren) will. Der OPEN-Befehl hat im wesentlichen die Aufgabe, zu einem File-Namen einen solchen DOS-Puffer einzurichten.

In manchen Systemen wird dafür RAM reserviert, für wieviel Puffer kann der User sogar wählen. Dort belegt dann jedes aktive OPEN einen dieser Puffer.

Wird nun während einer Session etwas geändert (File wird länger), dann wird das nur im File-Puffer notiert, nicht auf der Disk. Erst ganz zum Schluß wird das Directory und die Block-Belegungstabelle aktualisiert, das bewirkt der Befehl CLOSE. Und noch eines: Der Sektor-Puffer wird immer erst auf die Disk geschrieben, wenn er voll ist. Da aber die Datenmenge höchst selten ein ganzzahliges Vielfaches von 256 ist, bleibt der Sektor-Puffer zum Schluß mehr oder weniger leer. Hier sorgt CLOSE für dreierlei.

Es wird eine EOF-Marke (End of File) nach dem letzten User-Byte in den Puffer geschrieben.

Es wird der ganze Puffer auf die Disk kopiert.

Es wird der Puffer wieder freigegeben (für das nächste OPEN).

Warum ist nun OPEN gefährlich? Stellen Sie sich vor, zwischen OPEN und CLOSE tritt eine Störung auf, wie zum Beispiel Netzausfall, eine Spannungsspitze, Fehlbedienung oder der simple Druck auf die BREAK- oder RESET-Taste! Dann heißt das, die letzten Daten sind nicht auf der Disk, dem File fehlt das EOF-Byte und das Directory stimmt nicht mit der Wirklichkeit überein. Die meisten Betriebssysteme geben sich dann gar keine Mühe zu retten, was noch zu retten ist, die melden schlicht Error und weigern sich, den File weiter zu bearbeiten.

Daraus folgt Regel 1: Einem OPEN sollte so schnell wie möglich ein CLOSE folgen.

Regel 2: Updating nie am Original. Das heißt wenn Sie einen File ändern, zum Beispiel erweitern wollen, legen Sie zuerst eine Kopie an.

Dazu muß der File nur gelesen werden, im Modus »Read only« ist aber ein File gegen die genannten Pannen recht unempfindlich.

Darum:
— File im Read-Mode öffnen
— auf einen im Write-Mode geöffneten File kopieren
— Read-File schließen
— den zu »appenden« Teil schreiben

```
 10 D$=CHR$(13)+CHR$(4)                    0123456789
 20 F$="TESTFILE"
 30 PRINT D$"OPEN"F$;D$"DELETE"F$;          123456789
    D$"CLOSE"F$
 40 PRINT D$"OPEN"F$;D$"WRITE"F$            23456789
 50 PRINT "0123456789"
 60 PRINT D$"CLOSE"F$                       3456789
 70 REM
 80 REM ----------------------              456789
 90 REM
100 PRINT D$"OPEN"F$                        56789
110 FOR X=0 TO 9
120 PRINT D$"READ"F$",B",X                  6789
130 INPUT A$:PRINT A$
140 NEXT                                    789
150 PRINT D$"CLOSE"F$
160 END                                     89
                                            9
```

Bild 3. So kann man beim Apple den Filepointer manipulieren

— alles testen
— Original löschen, neuen File auf Namen des Originals umbenennen.

In der Befehls-Syntax unterscheidet man zwei Gruppen. Dabei geht es unter anderem darum, wie man zwischen den DOS-Puffern, mit denen das System arbeitet und den File-Namen, die der Benutzer kennt, eine Verbindung herstellt. In den meisten Basic-Dialekten sind die DOS-Puffer von 1 bis n numeriert. Im OPEN-Befehl wird eine dieser Nummern dem File-Namen zugeordnet, zum Beispiel als »OPEN "0",1,"Daten"«. Hier wurde der File »Daten« geöffnet, der Transfer läuft über den Puffer Nummer 1, "0" heißt, es ist ein Output-File. Beim Apple II kennt der Anwender die Puffer-Nummer nicht, und das geht so:

10 D$=CHR$(4)
20 PRINT D$"OPEN Daten"
30 PRINTD$"WRITE Daten"

Zeile 10 bringt das Steuerzeichen »es folgt DOS-Befehl« in die Variable D$. In Zeile 20 die Anweisung, den File »Daten« zu öffnen, Zeile 30 schließlich legt die Richtung fest, hier Schreiben (auf Disk).

Jetzt wirken alle PRINT-Befehle anstatt auf den Schirm auf die Disk, auch LIST schreibt jetzt auf die Diskette. Ein CLOSE-Befehl hebt diese Wirkung wieder auf.

In der anderen Lösung muß man beim PRINT-Befehl die Buffer-Nummer mit angeben, zum Beispiel als »PRINT#1,"text"«.

In Bild 1 sind beide Verfahren gegenübergestellt, und zwar am Beispiel einfacher serieller Files. Wie Sie sehen, ist die Apple-Lösung recht umständlich. Ihr Vorteil liegt im einfachen Routing, so nennt man das Umleiten von Bildschirmausgaben auf andere Geräte, beziehungsweise das Holen von Daten von der Disk anstatt von der Tastatur.

In Bild 2 werden die Lösungen für Random-Files gegenübergestellt. Hier zeigt der Apple deutliche Vorteile, weil das Fielding und Converting (nicht gezeigt) entfallen.

ISAM heißt Index Sequentiel Access Method und ist tatsächlich die Kombination der Vorteile beider Standard-Verfahren, sozusagen ein Random-File mit variabler Record-Länge und noch einigen Vorteilen. Sie erinnern sich an den Filepointer? Dieser wird beim Lesen oder Schreiben jeweils um 1 Byte weitergeschaltet. Es gibt aber auch einen Befehl, den Filepointer auf ein bestimmtes Byte im File zu stellen. Liest man dann ab da, wird der Text bis zum nächsten Schlußzeichen erfaßt.

```
1 PRINT "GEBEN SIE 3 TEXTE EIN":PRINT
2 FOR I=1 TO 3
3 PRINT I". TEXT ";:INPUT A$(I)
4 NEXT
10 D$=CHR$(13)+CHR$(4)
20 F$="TESTFILE"
30 PRINT D$"OPEN"F$;D$"DELETE"F$;D$"CLOSE"F$
35 PRINT D$"OPEN INDEX";D$"DELETE INDEX";D$"
   CLOSE INDEX"
40 PRINT D$"OPEN"F$;D$"WRITE"F$
50 FOR I=1 TO 3
60 PRINT A$(I)
62 A(I)=A(I-1)+LEN(A$(I))+1
65 NEXT
66 PRINT D$"CLOSE"F$
67 PRINT D$;"OPEN INDEX":D$"WRITE INDEX"
68 FOR I=0 TO 2:PRINT A(I):NEXT
69 PRINT D$"CLOSE INDEX"
70 REM
80 REM ---------------------------
90 REM
91 CLEAR
92 D$=CHR$(13)+CHR$(4):F$="TESTFILE"
95 PRINT D$"OPEN INDEX":D$"READ INDEX"
96 FOR I=0 TO 2:INPUT A(I):NEXT
97 PRINT D$"CLOSE INDEX"
100 PRINT D$"OPEN"F$
110 FOR I=0 TO 2
120 PRINT D$"READ"F$",B",A(I)
130 INPUT A$:PRINT A$
140 NEXT
150 PRINT D$"CLOSE"F$
160 END
```

Bild 4. Ein ISAM-File mit drei Sätzen und einem Schlüssel

Bild 3 demonstriert dies für den Apple. In den Zeilen 10 bis 60 wird auf einen sequentiellen File der String »0123456789« geschrieben.

In der Schleife ab Zeile 110 wird der Filepointer jeweils auf die relativen Bytes 0 bis 9 gestellt und ab da gelesen. Das Ergebnis zeigt das Bild unter dem Listing. Der Trick beim ISAM-File ist nun schon fast zu erraten. Wenn man auf einen sequentiellen File Sätze verschiedener Länge schreibt, und zu jedem Satzbeginn den Filepointer-Wert notiert, dann kann man auch direkt auf jeden Satz zugreifen. Praktisch werden die Filepointer-Werte in einem Array notiert, der dann separat auch auf der Disk gespeichert wird. Somit muß man gar nicht die Filepointer kennen, sondern kann über den n-ten Array-Index auf den n-ten Satz zugreifen.

Im Prinzip entspricht also der Index der Record-Nummer im Random-File. Einzusehen, daß man keinen Index-Array braucht, wenn alle Sätze gleich lang sind.

Dann kann man den Filepointer-Wert auch errechnen, und genau das tut das DOS, wenn Sie einen Random-File definiert haben, mehr nicht.

Bild 4 zeigt nun die vollständige Lösung für den Apple. Sie müssen wissen, daß der Filepointer (B-Parameter) ab Null zählt, folglich muß der erste Index auch Null sein. Das ist er, weil die Schleife ab Zeile 50 mit 1 startet. Den letzten Wert, den Filepointer nach dem letzten Satz, benötigen wir auch nicht, weshalb die Leseschleife ab Zeile 110 nur bis 2 läuft. Zeile 62 rechnet den Filepointer-Wert über die Länge der

```
1 PRINT "GEB...                       RINT
2 FOR I=1 TO 3
3 PRINT I;". TEXT ";:INPUT A$(I)
4 NEXT
20 F$="TESTFILE:0"
30 OPEN "O",1,F$,"MI"
35 OPEN "O",2,"INDEX:0"
50 FOR I=1 TO 3
60 PUT 1,,,A$(I);
62 A(I)=LOC(1)!
65 NEXT
66 CLOSE 1
68 FOR I=1 TO 3:PRINT #2,A(I):NEXT
69 CLOSE 2
70 REM
80 REM --------------------------
90 REM
91 CLEAR
92 F$="TESTFILE:0"
95 OPEN"I",1,"INDEX"
96 FOR I=1 TO 3:INPUT #1,A(I):NEXT
97 CLOSE 1
100 OPEN "I",1,F$,"MI"
110 FOR I=1 TO 3
120 RBA=A(I-1)
130 GET 1,!RBA,,A$;:PRINT A$
140 NEXT
150 CLOSE 1
160 END
```

Bild 5. NEWDOS und GDOS unterstützen ISAM-Files

Sätze. Das automatisch hinzukommende Schlußzeichen je Satz ist mitzuzählen.

Bild 5 bringt ein ISAM-Beispiel für die Modelle und TRS-80 und Video-Genie unter NEWDOS beziehungsweise GDOS. Der Vorteil liegt hier darin, daß über die LOC-Funktion der Filepointer abgefragt werden kann, beim Apple und anderen muß man dessen Wert selbst rechnen, siehe Zeile 62 im Apple-Listing.

Die bisher vorgestellten ISAM-Files haben nur ein Index-Array, auch Schlüssel genannt. Schon mit einem Schlüssel ist allerhand möglich, zum Beispiel sortieren. Dazu muß man zwar den Daten-File noch lesen, in ihm aber kein einziges Byte bewegen. Sortiert wird nur der Index. Für diverse Sortierkriterien kann man sogleich die richtigen Index-Files bereit halten. Der zweite Weg besteht darin, den Index-File zu teilen. In einem hält man zum Beispiel für eine Kunden-Datei alle Pointer auf Großkunden, im nächsten Index-File alle Pointer auf Kunden für die Produktgruppe A. In Adreß-Dateien ist es sinnvoll, so viele Index-Files zu führen, wie im Menü Kriterien angeboten werden. Kommt dann eine Adresse hinzu, wird sie immer ans Ende des Daten-Files gehängt und der entsprechende Index-File um den Filepointer-Wert erweitert.

Schließlich kann man noch mit mehreren Schlüsseln (Index-Files) arbeiten. Schon bei der Erfassung läßt man zwei Pointer (oder mehr) mitlaufen.

Das war nur ein kleiner Ausschnitt von dem was ISAM-Files alles bieten. (P. Wollschläger/bo)

„Alle Menschen sind klug; die einen vorher, die anderen nachher.“

Chinesisches Sprichwort

„Die Welt ist überhaupt nur dadurch weitergekommen, daß irgend jemand die Courage gehabt hat, an Dinge zu rühren, von denen die Leute, in deren Interesse das lag, durch Jahrhunderte behauptet haben, daß man nicht an sie rühren darf.“

Arthur Schnitzler,
österreichischer Schriftsteller
(1862–1931)

„Es ist nett, wichtig zu sein. Aber es ist wichtiger, nett zu sein.“

Graffito an einer Mauer

Einlesen von Programmzeilen beim TRS-80

Manchmal ist es durchaus nötig, Programmzeilen erst während des Programmablaufs einzugeben. Dies ist zum Beispiel bei Plotprogrammen nützlich, da hier oft mehrere Funktionen nacheinander geplottet werden sollen. Ferner können Sie dadurch auch Eingaben dauerhaft in Data-Zeilen speichern. Deshalb soll an folgenden Programmzeilen demonstriert werden, wie einfach dies unter NEWDOS/80 geht.

```
10 PRINT "Geben Sie bitte die Funktion ein"
15 PRINT "Die unabhängige Variable ist X"
20 INPUT "Y=";A$
30 ZL$="1000 Y="+A$+":RETURN"+CHR$(13)
40 OPEN "0",1,"FUNKTION/BAS", "FI"
50 PUT 1,,,(LEN(ZL$))ZL$;
60 CLOSE(1)
65 CMD"F",DELETE 1000
70 MERGE "FUNKTION/BAS"
75 ON ERROR GOTO 200
80 X=10
90 GOSUB 1000
100 PRINT "Y(";X;")=";Y
110 END
200 IF ERL=1000 THEN PRINT "Fehler in der Funktions-
eingabe"
210 RESUME 10
```

Das ganze Geheimnis an der Sache ist, daß nicht nur der LOAD-Befehl, sondern auch der MERGE-Befehl auf ASCII-Files anwendbar ist. Deshalb müssen die einzulesenden Programmzeilen lediglich als ASCII-File abgelegt werden. Dies ist eine der leichtesten Übungen für NEWDOS, deshalb sei hier nur noch kurz das Rezept in Stichworten erläutert: Die Programmzeile mit Chr$(13) abschließen und ein PUT auf ein File. Für die korrekte Syntax ist ausschließlich der Anwender des Programms verantwortlich. Deshalb gehört ein Abfangen von Fehlern in den eingelesenen Zeilen mit einem ON ERROR GOTO zur unbedingt notwendigen Austattung des Programms. Die Anweisung ON ERROR GOTO darf allerdings erst nach dem MERGE-Befehl stehen. (M. Ebinger/bo)

Peeks & Pok's -- Peeks & Pok's -- Peek

POKE 16413,0 schaltet den Bildschirm ab.
Du hast keine Daten oder Eingaben
mehr auf dem Bildschirm, obwohl alle
Eingaben vom Programm akzeptiert werden.

POKE 16413,7 schaltet den Bildschirm wieder ein.
Anwendungen dieses Pok's :
bei z.B. Passwordeingaben

Ulrich Böckling

Peeks & Pok's -- Peeks & Pok's -- Peek

Einen Rechner im Herzen der Schweiz
packt beim Rechnen ganz plötzlich der Geiz.
Er addiert voller List
auch noch dort, wo nichts ist –
das hat grad in der Schweiz seinen Reiz.

Ein Kurarzt im schönen Bad Pfrund
prüft am Terminal jeden Befund.
Diagnose per Kabel
ist für ihn keine Fabel –
und ihn selbst macht das ganz schön gesund.

Ein End-User bei Bayrisch Zell
rechnet sicher und unheimlich schnell.
Im April wird er Vater,
das gibt sicher Theater –
so verrechnet sich jeder mal, gell?

Eine Lochkartendame in Herne
lochte stets ihre Karten recht gerne.
Eines Tages, oh Schreck,
war der Locher dann weg –
jetzt verarbeitet sie in der Ferne.

Ein Computel steht auch in Shanghai
und zählt stets binäl bis auf zwei.
Doch weil viele Millionen
von Chinesen da wohnen,
ist das Zählen schon fast Quälelei.

Ein Computerexperte aus Bruhl
macht jetzt Hochzeit, doch bleibt er ganz cool.
Statt von Liebe und Leid
spricht von Bit er und Byte –
und die Braut schluchzt: „Du hast kein Gefool!"

Mission Impossible

Ratschläge

Die Kommandos können durch die ersten drei Buchstaben abgekürzt werden.
SAVE GAME speichert den Spielstand auf Kassette.
HELP und SCORE sind keine Hilfen.
Der Befehl EXAMINE bringt dafür um so mehr. Er sollte möglichst oft im Bezug auf gefundene oder sehbare Gegenstände angewandt werden.

1. Bis zur Apparatur mit den mehrfarbigen Knöpfen und dem Finden des Saboteurs sollten Sie es alleine schaffen.
Bei den Knöpfen hilft nur ausprobieren. Achten Sie dabei auf den Bombendetektor. Dies gilt für jede Benutzung der Knöpfe.
2. Die Karten mit ihrem Bild verschaffen Ihnen jeweils Eintritt in bestimmte Räume.
3. Im Besucherraum sollten Sie auf das Fenster achten.
4. Zu öffnen geht es nicht. Also weiter mit Gewalt. Das Mittel dazu sehen Sie sehr früh.
5. Das Sicherheitssystem ähnelt dem der Tür. In welchem Zusammenhang tauchte "window" schon mal auf?
6. Jeder Schlüssel paßt in ein Schloß. Wo haben Sie schon welche gesehen?
7. Ein Mop mit komischen Geräuschen? Stellen Sie sich das Ding bildlich vor. Was könnte damit sein?
8. Immer noch nichts? SUCHEN hilft.
9. Eine Tür läßt sich nicht öffnen. Da gibt's nur eins: hart bleiben.
10. Filmkassette? Projektor? Vorführung? Viel Laufarbeit, aber es lohnt sich.
11. Aha, Schutzanzug ist nötig. War eigentlich klar. Ist ja auch ein Reaktor. Aber das Plastik?
12. Die Bombe tickt vor sich hin, ohne Sie zu beachten? Trennen Sie das Ding von seinem Nabel. Das Mittel ist in jeder Werkzeugkiste vorhanden. Und Sie haben es auch schon gesehen.
13. Was tun mit dem Ding? Was tun, wenn es brennt?
14. Sie löschen Feuer doch auch mit Wasser. Sie müssen es nur dahin bekommen, wo es hin soll.
15. Sagen Sie bloß nicht, sie haben 11. vergessen! Wenn der Krug nicht zum Brunnen geht, muß ...
Na also!

Achtung !!!

Nicht umblättern, da die

die Auflösung auf der Rückseite ist.

Versuch's doch erst selber!

Du hast doch umgeblättert ???

Also bitte .... die Auflösung

Mission Impossible

Die Situation und der Sinn des Spiels lassen sich aus dem Anfangstext und der ersten Situation ersehen.

Die Räume
Jeder Raum erhält eine Nummer. Die möglichen Ausgänge werden den Nummern der Räume zugeordnet, zu denen sie führen.

1. Briefing room; W=2
2. Large grey corridor; N=3, S=4, W=5, E=1, U=6, D=7
3. Twisting blue hallway; N=9, S=2, W=6
4. Grey room; N=2
5. Maintenance room 1; E=2
6. Twisting white hallway; N=10, E=3, D=2
7. Twisting yellow hallway; U=2, N=8
8. Yellow room; S=7, Tür=12
9. Blue room; S=3, Tür=15
10. White room; S=6, Tür=11
11. Visitors room; Tür=10
12. Yellow corridor; W=13, Tür=8
13. Maintenance room 2; E=12, U=14
14. Projectionist room; D=13
15. Anteroom; Tür1=9, Tür2=18, W=17, U=16
16. Viewing room; D=15
17. Storage room; E=15
18. Control room; E=19, D=20, Tür=15
19. Break room; W=18
20. Reactor core; U=18

Der Spielverlauf:
Sie beginnen immer in Raum 1. Nach INVENT bemerken Sie, daß Sie einen Bombendetektor tragen, der Sie warnt, wenn die Bombe scharf ist.
Neben sich finden Sie einen Recorder. Mit TAKE RECORDER und START RECORDER wird Ihnen die Situation noch einmal erklärt.
Begeben Sie sich in Raum 5. Dort finden Sie einen Plastikeimer, der später wichtig für Sie ist. TAKE PAIL.
Sodann gehen Sie in Raum 4, wo Sie sich mit SIT vor eine Apparatur setzen. PESS RED und PRESS WHITE ergeben eine Reaktion, die sich nach GET UP und GET PICTURE in Form einer Besucherkarte vorfinden.
Nun warten Sie, bis in der Ferne ein Fall zu hören ist. Durchsuchen Sie alle für Sie bis jetzt erreichbaren Räume, bis Sie den Saboteur finden. Durchsuchen Sie ihn mit FRISK HIM und nehmen Sie sodann seine Karte (bzw. PICTURE) und ihn selber.
In Raum 10 verschaffen Sie sich mit SHOW AUTHORIZATION Einlaß zu Raum 11. Mit SMASH WINDOW, WITH RECORDER. SHOW AUTHORIZATION, ENTER WINDOW, GET KEY, ENTER WINDOW erlangen Sie ein weiteres wichtiges Utensil. Durch PRESS WHITE verlassen Sie den Raum.

Vor der Apparatur in Raum 4 sitzend geben Sie UNLOCK YELLOW ein. Nun drücken Sie hintereinander die Knöpfe weiß, rot, gelb, weiß und holen sich ihr nächstes Bild ab, diesmal als Instandhalter.
Vor Raum 8 zeigen Sie wieder ihr Bild und begeben sich in Raum 13. Mit SEARCH MOP finden Sie einen weiteren Schlüssel. Die Zange nehmen Sie auch gleich mit.
In Raum 4 holen Sie sich mit rot, blau und weiß nach Aufschließen des blauen Knopfes die Karte "Sicherheitspersonal".
Nachdem Sie damit über die Räume 9 und 15 nach Raum 17 gelangt sind, ziehen Sie sich mit WEAR SUIT den Schutzanzug an und nehmen mit GET WATER Wasser mit (dazu brauchen Sie den Eimer).
Die Tür zu Raum 18 öffnen Sie mit PUSH HARD und nach ENTER DOOR finden Sie dort eine Filmkassette.
Sie können die Kassette mit INSERT FILM in den Projektor im Raum 14 einlegen und dann in Raum 11 mit PRESS GREEN ein kleines Filmchen sehen.
Zurück in Raum 18 stellen Sie den Eimer mit dem Wasser ab, denn Plastik verformt sich in Reaktornähe.
In Raum 20 ist endlich die Zeitbombe, die Sie mit CUT WIRE lösen und dann mitnehmen in Raum 18. Dort leeren Sie mit POUR PAIL den Eimer über die Bombe, die damit entschärft ist.

Gerald Schröder

1966: »put – put – put . . .«

2066: »input – input – input . . .«

Laufwerkmodifikation

Was macht man, wenn man 40-Spur-Disketten auf 80-Spur-Laufwerken laufen lassen möchte? Ganz einfach, man stellt den PDRIVE passend ein.

Was macht man, wenn man selbstbootende 40-Spur-Disketten auf einem 80-Spur-Laufwerk booten möchte? Auch ganz einfach, bei einem neuen TEAC 55F 80 -Spur-Laufwerk überbrückt man die beiden Lötpunkte, die für den Widerstand R 14 vorgesehen sind mit einem 10-Ohm-Widerstand und schon hat man ein 40-Spur -Laufwerk.
Ich habe den Widerstand mit einem Minischalter umschaltbar gemacht und kann so hardwaremäßig von 80-Spur-Drive auf 40-Spur-Drive umschalten. Klappt prima.

Ulrich Böckling

Um eine 40-Track-Disk auf einem 80-Track-Laufwerk zu booten gibt es auch eine Softwaremöglichkeit. Man muß nur zuerst ein paar Bytes im Bootsektor der 40-Track-Disk umzappen. Danach kann man die Diskette auf einem 80-Track -Laufwerk booten.
Mit den Zaps wird erreicht, daß das 80-Track-Laufwerk zwei Steppschritte macht und somit jeden 2. Track anspricht. Wie bei PDRIVE die Einstellung AL.

Um die gleiche Diskette auf 40-Track-Laufwerken zu booten sollte die Änderung wieder rückgängig gemacht werden. Ist ja klar.

Die zu ändernden Zeilen sind wie folgt :

Bootsektor
Spur 0, Sektor 0

Bootsektor alt

```
A0  CB7E 28E6 7E36 D0C1 D1E6 FC20 0C1C 7BD6 .ß(.ß6........ä.
B0  0A20 0314 1E00 D97E C9CD D742 360B 1098 .......ß...B6...
C0  21DD 427E FE03 28FB 23CD 3300 18F5 CDD7 !.Bß..(.#.3.....
D0  42CB 4620 FC7E C93E 063D 20FD C91C 1F45 B.F..ß.>.=.....E
E0  5252 4F52 031C 1F4E 4F20 5359 5303 18F1 RROR...NO.SYS...
F0  213D 6E18 D021 596E 18CB 2185 6E80 0704 !=n..!Yn..!.n...
```

Bootsektor geändert

```
A0  CB7E 28E6 7E36 D0C1 D1E6 FC20 0C1C 7BD6 .ß(.ß6........ä.
B0  0A20 03CD E542 D97E C9CD D742 360B 1098 .....B.ß...B6...
C0  21DD 427E FE03 28FB 23CD 3300 18F5 CDD7 !.Bß..(.#.3.....
D0  42CB 4620 FC7E C93E 063D 20FD C91C 1F45 B.F..ß.>.=.....E
E0  5203 4E53 0314 1E00 36D0 3643 CDCE 4236 R.NS....6.6C..B6
F0  D036 43CD CE42 36D0 CDCE 427A 32ED 37C9 .6C..B6...Bz2.7.
```

Jens Neueder

Auf Seite 67 (Info 6) schreibt Walter, daß man Edtasm von Debug aus aufrufen könne. Nachteil sei nur, daß der Cursor nicht sichtbar sei. Diesen Nachteil kann man im NEWDOS80 mit Zap # 22 beheben:

```
Edtasm/CMD ,05 ,12 ändern von CD 39 59 0E
                         in  CD 00 57 0E

Edtasm/CMD ,03 ,1A       von 40 7D E6 3F C0 11 C0 FF 19 C9
                         in  40 CD 39 59 3E 0E C3 39 59 C9
```

Noch ein anderes nützliches Zap, wenn im Basic die Abkürzungen L ,E ,D ,A für LIST ,EDIT ,DELETE ,AUTO im Direktmodus nicht immer erkannt werden:

```
SYS18/SYS ,02 ,31 ändern von 2C D7 28 06 FE
                         in  2C C3 59 55 FE

SYS18/SYS ,03 ,68        von 18 00 00 00 00 00 00 00 00 00 00
                         in  18 D7 FE 0D CA 2F 54 C3 29 54 00
```

Herbert Alber

Ein weiteres Problem vom Clubtreffen will ich gleich noch an dieser Stelle klären:
Viele von Euch haben das Schachprogramm SARGON3. Aber es läuft nicht unter Disk-Betrieb. Dies ist nämlich ein Kassettenprogramm, das aber unter Newdos mit folgendem Trick zu starten ist:
1.: Load SARGON3/CMD
2.: BASIC2
3.: SYSTEM
4.:/48128
So funktioniert es bei mir einwandfrei. Der Grund für das seltsame Verhalten: Diese Programm ist ursprünglich in einem nicht TRS-Kassettenformat gespeichert und dehalb versagt auch LMOFFSET.
W.Zwickel

Pi ganz einfach

In allen Basic-Lehrbüchern, die ich kenne – selber besitze ich gegen zwei Dutzend – und in allen Programmen, die ich bisher gesehen habe, wird die Zahl Pi mit ihrem Zahlenwert definiert:

...Pi = 3.141592..
Es wäre sinnvoller, die Zeile stattdessen zu schreiben:
...Pi = 4*ATN(1)

Denn erstens irrt man sich dabei nicht, wie beim Eintippen von Ziffern, zweitens geht es leichter und drittens wird dabei die Genauigkeit des jeweiligen Rechners ausgenützt.
Heinrich Kraft, Kiel

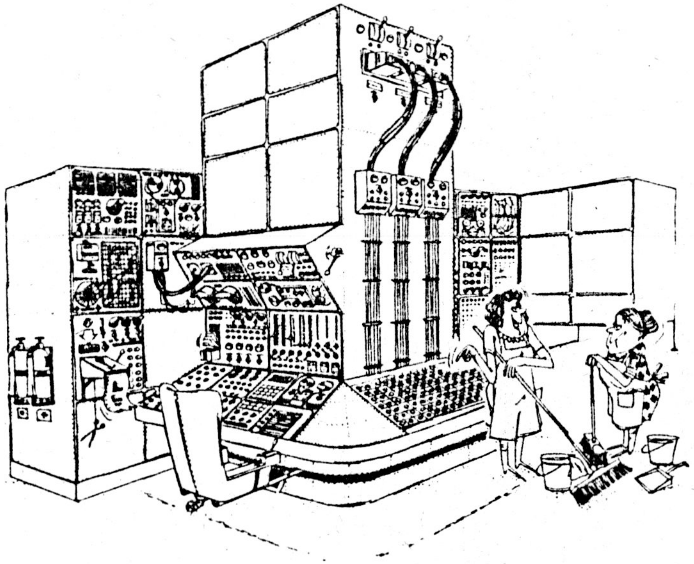
Tips + Tricks + Tips + Tricks + Tips

Ausgabeumschaltung in BASIC

Im letzten Clubinfo war ein Beitrag vom Josef Konrad abgedruckt, in dem er beschrieb, wie man Ausgaben, die normalerweise auf dem Bildschirm erscheinen, auf den Drucker umleitet. Die in diesem Beitrag aufgezeigte Methode hat den Nachteil, daß nur die Texte umgeleitet werden, die durch den PRINT-Befehl ausgegeben werden. Texte, die anders auf den Bildschirm gebracht werden (z.B. INPUT"Gib was ein";A$), werden davon nicht berührt.
Abhilfe schafft hier die Verwendung des ROUTE-Kommandos des NEWDOS80. Wie dies zu bewerkstelligen ist, habe ich unten kurz aufgezeigt. Benutzt man diese Methode, wird alles, was normalerweise auf dem Bildschrim angezeigt wird, auf den Drucker umgeleitet. Wichtig ist dabei nur, daß man nach Beendigung des Programms die ROUTE-Anweisung durch ROUTE,CLEAR (bzw. CMD"ROUTE,CLEAR) wieder rückgängig macht.

```
10 CLEAR 1000 : CLS
20 PRINT @ 0,"Ausgabe auf (D)rucker oder (B)ildschirm?"
30 AINKEY: IF A"D" OR A"d" THEN 50
40 IF A"B" OR A"b" THEN 60 ELSE GOTO 30
50 CMD"ROUTE,DO,PR" : GOTO70
60 CMD"ROUTE,CLEAR"
70 FOR X=32 TO 191 : PRINT CHRX);" "; : NEXT : RUN
```

Hartmut OBERMANN



»Na, wie wär's – einfach auf ein paar Knöpfe drücken und dann nix wie weg!«

Warnung ! ! ! TRS80 Model I + VideoGenie

Im Clubinfo Nr. 6 veröffentlichte unser Assembler-Freak Arnulf Sopp einen Beitrag mit dem Titel "Nochmals: Joystick Anschluß". Die darin aufgezeigte Möglichkeit einen Joystick an ein VideoGenie anzuschließen oder die Tastatur um weitere Tasten zu erweitern ist durchaus praktikabel und bringt eine Menge zusätzliche Möglichkeiten ohne viel zu kosten.

Aber es ist Vorsicht geboten bei dem Versuch, die Vorgehensweise, die der Arnulf beschreibt, auf den TRS80 zu übertragen. Prinzipiell ist zwar die Tastatur gleichartig aufgebaut, schaltungstechnisch unterscheiden sich die Computer stellenweise jedoch erheblich.

Eine dieser Stellen ist die Tastatur der beiden kompatiblen Computer. Während die Tastaturplatine des VideoGenie nur die Tasten und ein paar Widerstände beherbergt, sitzen auf der TRS80-Tastatur zusätzlich vier Integrierte Schaltkreise. Diese IC's besitzt der TRS80 nicht etwa mehr als das VideoGenie, sie sind nur beim Nachbau (und nichts anderes ist das VideoGenie) von der Tastaturplatine auf das CPU-Board verlagert worden. Aus dieser Tatsache ergeben sich einige Umstände, wenn man den Joystickanschluß trotzdem so realisieren will, wie der Arnulf ihn beschreibt oder wenn man sich ein paar zusätzliche Funktionstasten einbauen will.

Am besten beschafft man sich eine Einbaubuchse mit 16 Anschlüßen, auf die man die Tastaturspalten- und -reihen-Leitungen legen kann. Die Spaltenleitungen sind relativ einfach zu finden, man kann sie direkt an den 4.7kΩ abnehmen. Dabei muß man nur darauf achten, daß man die richtige Seite des Widerstands erwischt. Die Seite von der aus eine Leiterbahn zu irgendeiner Taste führt ist die richtige! Die Reihenleitungen sind auch nicht viel schwieriger ausfindig zu machen. Man verwendet am besten die Anschlüße der IC's, die folgendermaßen zugeordnet sind:

Spalte 1 Widerstand R8	Reihe 1 IC 1 Pin 8
Spalte 2 Widerstand R5	Reihe 2 IC 1 Pin 2
Spalte 3 Widerstand R3	Reihe 3 IC 1 Pin 10
Spalte 4 Widerstand R2	Reihe 4 IC 2 Pin 2
Spalte 5 Widerstand R7	Reihe 5 IC 1 Pin 6
Spalte 6 Widerstand R1	Reihe 6 IC 1 Pin 4
Spalte 7 Widerstand R4	Reihe 7 IC 1 Pin 12
Spalte 8 Widerstand R6	Reihe 8 IC 2 Pin 4

Diese sechzehn Punkte verbindet man mit den Anschlüßen der Buchse und kann daran dann, wie vom Arnulf beschrieben, die Tastaturerweiterungen anschließen.

Auch ich wünsche euch beim Basteln viel Spaß und gutes Gelingen!!!

Hartmut Obermann

Hardware - Arbeitskreis

E C B - B U S System

Liebe Clubfreunde

Wie die meisten von Euch sicherlich schon wisen, haben wir beim Clubtreffen einen Hardware-Arbeitskreis gebildet. Als erstes Projekt wollen wir ein universelles ECB-Bus System bauen. Das soll vorerst von den Mitgliedern des Arbeitskreises entwickelt werden. Wobei jeder Mitstreiter einen Teil der aufzubringenden Arbeit leistet. Wenn das Projekt fertig ist, sollen fertige Platinen und eine Bauanleitung vorliegen, die es auch dem elektronisch nicht so vorbelasteten Clubmitglied erlauben das ECB-Bus System aufzubauen.

Meine Planungen sind inzwischen soweit fortgeschritten, daß ich unbedingt auch die Meinungen der übrigen Hardware-Fans brauche. Dazu will ich das System kurz beschreiben:

Die Basisplatine soll eine ECB-Platine von der Zeitschrift c't sein. Diese hat 10 Steckplätze mit sogenannten VG-Steckern mit je 96 Kontakten in 3 Reihen angeordnet. Diese Platine kostet fertig ca. 50.- DM . Das ECB-System braucht nur 2 dieser Reihen, sodaß noch 32 Leitungen für TRS-spezifische Dinge zur Verfügung stehen. Der Anschluß an den TRS/VG/Komtek soll mit einer von mir zu entwickelnden Karte geschehen. Wobei Pufferung aller Signale vom und zum Computer, Adreßdecodierung der wesentlichen Portadressen sowie Terminierung der vom Computer kommenden Leitungen die Hauptaufgaben sind. Die Terminierung macht mir noch etwas Sorgen. Es stehen 3 Varianten zur Auswahl:

1. Aktive Terminierung (m.E. zu aufwendig, da eine weitere Karte notwendig würde)
2. Niederohmige Terminierung mit 220 und 330 Ohm. Ist leicht machbar, aber verbraucht ca 400 mA. Kann auch Probleme machen, wenn im Computer Treiber IC's der LS-Type verwendet werden.
3. Hochohmige Terminierung mit 2.2 und 3.3 Kohm. Erfordert ein Verbindungskabel zum Computer mit max. 60 cm Länge.

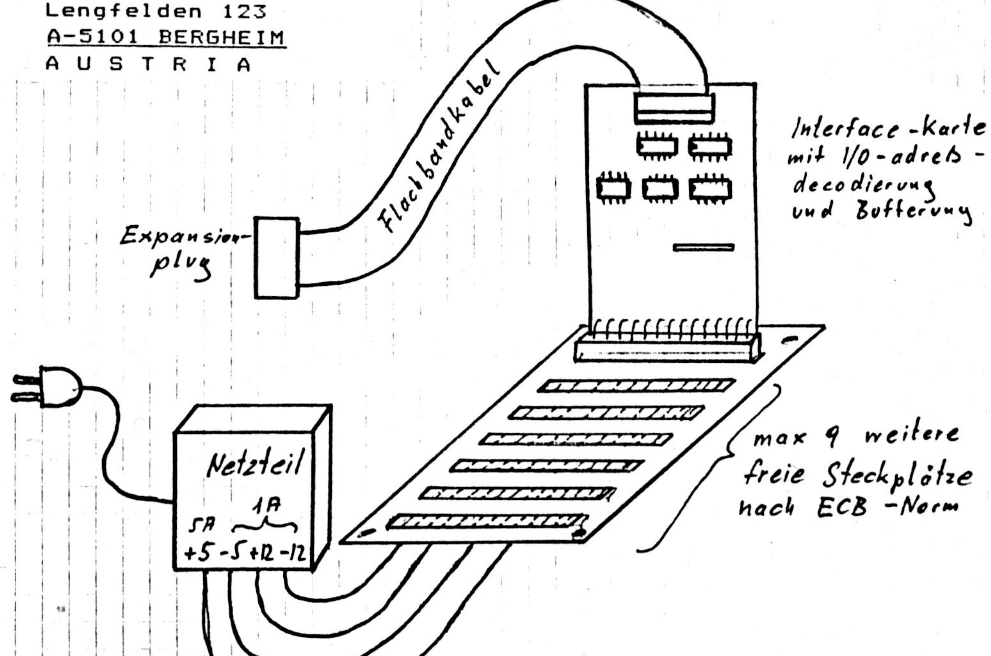
Ich habe vor die Karte so zu gestalten, daß man wahlweise die Variante 2 oder 3 bestücken kann.

Das zweite Problem ist die Steuerung des Datenbus-Puffers. Der muß ja auch auf read geschalten werden und darf keinesfalls versehentlich adressiert werden, sonst knüppelt er ja die Signale am Datenbus zusammen. Nur mit I/O request klappt das nur, wenn man sich auf die I/O Adressen beschränken würde. Aber das wollen wir ja nicht. So werde ich wahrscheinlich auf der Interface-card nur die I/O Adressen decodieren und falls jemand eine Karte mit Memory-Adressen baut, muß er eben selber ein CHIP-SELECT Signal erzeugen und auch zur Treiberkarte senden. Leitungen sind auf der 96-poligen Buchse ja genug frei.

Teilt mir bitte mit, ob Ihr damit einverstanden seid, oder ob jemand eine bessere Idee für obige Probleme hat. Ich habe meine ECB-Platine schon bestellt und habe auf jeden Fall beschlossen, eine Interface-Karte in Fädeltechnik zu bauen und alles auf Herz und Nieren zu testen, bevor es an eine Platine geht. Deshalb bitte ich auch um Geduld, daß es sicher noch ein paar Monate dauern wird, bis die fertigen Platinen vorliegen.

Herzliche Grüße aus Salzburg

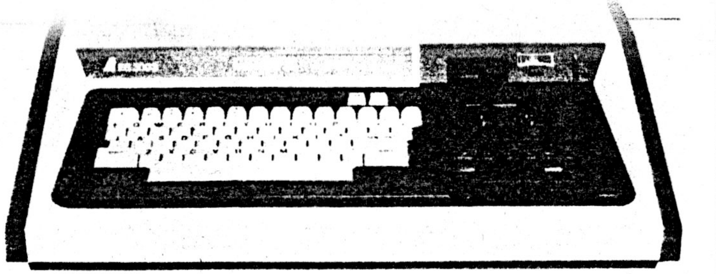
Walter Zwickel
Lengfelden 123
A-5101 BERGHEIM
A U S T R I A



Lieber INFO-Leser

Wenn Du in der INFO einen Druckfehler findest, dann bedenke bitte, daß dieser beabsichtigt ist. Die INFO bringt für jeden etwas, und es gibt immer Leser, die nach Fehlern suchen.

Ein sicheres Plätzchen



Maschinenroutinen geschützt im GENIE untergebracht

Helmut Bernhardt

Die Sicherung von Maschinenprogrammen im oberen Adreßbereich durch Befehle wie HIMEM oder MEMSIZE gewährt lediglich Schutz gegen Überschreiben durch BASIC-Programme. Der Computer legt Maschinenprogramme beim Laden immer in den Speicherbereich, für den sie vorgesehen sind. Die Pointer werden dabei nicht berücksichtigt, so daß es normalerweise keinen Speicherbereich gibt, der wirklich 'sicher' ist. Mit einer kleinen Zusatzschaltung kann man jedoch ein 'sicheres Plätzchen' schaffen.

Auf der Suche nach diesem Platz im Adreßraum des TRS-80 oder GENIE I, II findet man schnell einige 'Lücken'. So beginnt der Adreßbereich der Tastatur bei 3800h und endet bei 38FFh. Da aber die Dekodierung des Freigabesignals für die Tastatur-Lesetreiber unvollständig ausgeführt ist, wird auch der Bereich von 3900h bis 3BFFh blockiert.

Das muß man nicht so hinnehmen. Durch geringfügige Eingriffe in den Rechner läßt sich die Tastatur vollständig dekodieren und gleichzeitig ein Freigabesignal für einen zusätzlichen Speicher-Baustein im Bereich 3900h bis 3BFFh gewinnen. Die dazu notwendige Hilfsschaltung zeigt Bild 1. Der zusätzliche Speicher besteht aus zwei ICs 2114 (1 K x 4 Bit); die weitere Dekodierung geschieht mit zwei TTL-ICs.

In den so gewonnenen 768 Byte RAM sind eigene Routinen ziemlich sicher, da man normalerweise nicht auf diesen Speicherbereich zugreifen kann. Programme mit eigener Treiberroutine könnten allerdings versuchen, die Tastatur im Bereich 3900h bis 3BFFh auszulesen. Solche Routinen würden laufend 'Eingaben' aus dem RAM erhalten, was zu Fehlfunktionen führt.

In solchen (seltenen) Fällen hilft dann nur noch der Einbau eines Zweifach-Umschalters, der entweder das unvollständig dekodierte oder das neu gewonnene Freigabesignal an die Tastatur-Lesetreiber legt und die Freigabe der RAMs bei Benutzung des 'alten' Freigabesignals für die Tastatur abschaltet. Programme, die diesen Schalter benötigen, lassen sich dann nicht mit dem zusätzlichen RAM verwenden. Das Betriebssystem NEWDOS-80 gehört glücklicherweise nicht zu dieser Gruppe von Software.

Eingriffe

Die vier ICs der Hilfsschaltung kann man auf einem Streifen Lochraster-Platine unterbringen. Die Versorgungsspannung (+5 V) ist jeweils an Pin 14 der TTL-ICs und an Pin 18 der RAM-Bausteine anzuschließen. Der Masseanschluß ist bei den RAMs Pin 9, bei den TTL-Bausteinen Pin 7. Zwischen Versorgungsspannung und Masse sollte man einen Kondensator (100 nF) schalten.

Beim Einbau der Karte in GENIE-Rechner kann man folgendermaßen vorgehen:

Zuerst ist das Gehäuse des Computers zu öffnen. Nachdem man die acht tiefer versenkten Schrauben in der Bodenwanne des Gerätes herausgedreht hat, läßt sich die obere Gehäuseschale abheben. Schraubt man noch die Tastatur ab, liegt links das CPU-Board, in der Mitte das Interface-Board und rechts das Netzteil sowie der Kassettenrecorder.

Die durchzutrennende Leitung (siehe Bild 1) ist die Leiterbahn, die vom hintersten IC (Z35, 74LS32) auf dem CPU-Board in der rechten Spalte nach 'hinten' verläuft (neben der 5-V- und Masse-Leitung). An die beiden Seiten der Trennstelle kann man den Ein- und Ausgang des 'OR'-Gatters der Hilfsschaltung anschließen. Das IC 'Z12' findet man an zweiter Stelle (von vorne) in der linken Spalte auf dem CPU-Board. Der Baustein 'Z15' (74LS32) liegt in der zweiten Spalte (von links gezählt) als zweites IC.

Die Numerierung der Verbindungsleitungen zum Interface-Board, an denen die meisten Signale für die Zusatz-Schaltung abzunehmen sind, erfolgt von vorne nach hinten (Nummern 1 bis 32).

Beim Anschließen der Erweiterung sollte man unbedingt darauf achten, daß die Datenleitungen nicht an den ROMs, sondern an den Lötpunkten der Stecker für die Verbindungsleitung zum Interface-Board abgenommen werden. Die an den ROM-Chips anliegenden Datenleitungen sind wegen der Freigabeschaltung der Treiber nur in Leserichtung verwendbar. An der Steckverbindung lassen sich außer $\overline{CS}$ und $\overline{MWR}$ alle für die RAMs nötigen Signale abnehmen. □

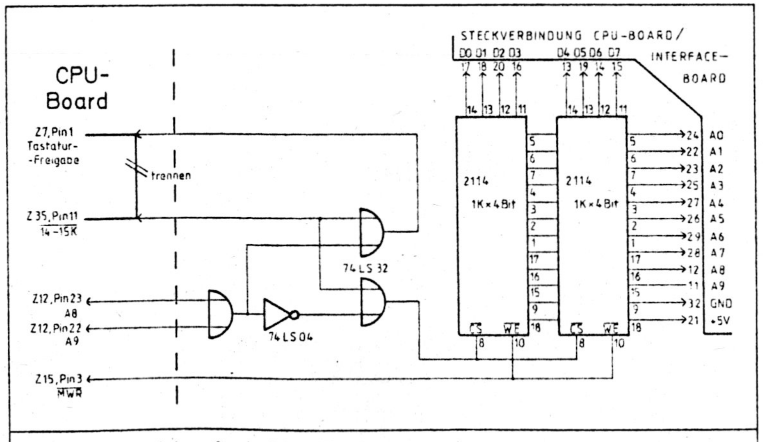


Bild 1. Die Zusatzschaltung für ein sicheres Plätzchen

SuperTape für TRS-80

Andreas Burgwitz, Andreas Stiller

Endlich schließt sich auch der TRS-80 der Gemeinde der SuperTape-Rechner an. Zuvor galt es allerdings eine Hürde zu nehmen, die die Entwickler des mittlerweile recht betagten Rechners errichteten: Vom SuperTape-Eingangssignal läßt die Schaltung des Kassettenports nichts über, was mittels Software ausgewertet werden könnte. Die Lösung dieses Problems besteht aus einer kleinen Zusatzplatine, die dem TRS-80 SuperTape-Betrieb mit 7200 Baud ermöglicht.

Gerade die Anwender des TRS-80 werden ein schnelles und vor allem störsicheres Kassettenverfahren begrüßen; sind sie doch vom Standard-Verfahren des Rechners Schlimmes gewohnt. Und gerade dieses Verfahren bescherte die Probleme, die den Einsatz einer Zusatzschaltung nötig machen. Aber wie fast alles, hat auch dies seine guten Seiten: Mit dem so notwendig gewordenen zusätzlichen Interface kann der TRS-80 SuperTape-Kassetten mit 7200 Baud beschreiben und lesen. Zusätzlich bietet die Schaltung je ein frei verwendbares 'Ein-/Ausgabebit'.

Die SuperTape-Software entspricht im wesentlichen dem in Heft 1/85 vorgestellten Programm für das Colour Genie, und damit auch dem SuperTape für den ZX-Spectrum. Allerdings sind die SuperTape-Ein-/Ausgaberoutinen an den Tandy angepaßt. Da der TRS-80 auch mit 7200 Baud lesen und schreiben soll, mußte die Laderoutine völlig neu erstellt werden.

Die TRS-80 SuperTape-Lösung kann man auf Geräten mit und ohne Diskettenlaufwerk betreiben. Sinnvollerweise sollte der Tandy aber mit minimal 16 KByte RAM ausgebaut sein. Das abgedruckte Programm liegt im Speicher ab der Adresse B900h. Natürlich kann man das Programm auch an die obere Grenze des Speichers legen, wo es aber oftmals mit einem Monitor oder anderen Utilities kollidiert.

Vor dem Start von SuperTape sollte man diesen Speicherbereich gegen Überschreiben durch BASIC-Programme schützen. Dies geschieht durch die Beantwortung der Frage 'MEM SIZE ?' mit 47300. Allerdings sichert man SuperTape damit nicht gegen Maschinenprogramme, die im Speicher ab B900h liegen.

Input

Die Eingabe des Programms in den Rechner sollte nicht ohne die Hilfe eines Monitor-Programms geschehen, das das Abspeichern der eingegebenen Routinen als 'SYSTEM-File' erlaubt. Mehr Tipparbeit, aber wesentlich weniger Ärger bei Änderungen ergibt die Eingabe mit Hilfe eines Assemblers. Verfügt man über keins dieser Hilfsprogramme, sollte man SuperTape vom c't-Software-Service als SYSTEM-File beziehen.

Bits schieben

Wie das SuperTape-Verfahren funktioniert, ist in mehreren vorausgegangenen c't-Ausgaben ausführlich erläutert worden. Die SuperTape-Programme aller Z80-Rechner orientieren sich dabei an den Routinen für den Spectrum aus c't 6/84.

Für 7200 Baud wird allerdings dabei eine einigermaßen 'flotte' CPU vorausgesetzt, die der Tandy mit seinen 1,774 MHz Taktfrequenz offensichtlich nicht aufweisen kann. Vor allen Dingen bei der Laderoutine wird es zeitlich etwas zu 'eng'.

Die optimale Prüfzeit liegt für 7200 Baud bei etwa 2/3 einer ganzen Periode für die Übertragung einer Null: also 93 Mikrosekunden. Davon ist sogar noch die mittlere Flankenerkennungszeit abzuziehen, so daß dem Tandy also ungefähr 150 bis 160 Takte zur Verfügung stehen. Außerdem sollte man nach oben und nach unten noch etwas Spiel haben, um rein empirisch die beste Prüfzeitkonstante ermitteln zu können.

Kommunizieren mit SuperTape

Das SuperTape-Verfahren [illegible] ausführlich [illegible] Das bedeutet, daß die im SuperTape-Format [illegible] Kassetten von jedem Computer gelesen werden können, für den es SuperTape-Routinen gibt. SuperTape funktioniert mit handelsüblichen Kassetten und Recordern, arbeitet aber um ein Vielfaches schneller als alle anderen Aufzeichnungsverfahren, die sonst bei Homecomputern verwendet werden. Für folgende Rechner hat c't bisher SuperTape-Routinen vorgestellt:

Sinclair ZX81	c't 4/84 u. 5/84
VC-20	c't 5/84 u. 10/84
C64	c't 5/84 u. 10/84
Sinclair Spectrum	c't 6/84 u. 9/84
Apple II (und kompatible)	c't 7/84
6809-Rechner (am Beispiel des Eurocom II)	c't 8/84
CP/M-Rechner	c't 11/84
Oric 1 und Atmos	c't 12/84
Colour Genie	c't 1/85
CBM 3000/4000/8000	c't 2/85
Sharp PC1500	c't 3/85

Die meisten der Programme sind auf Datenträgern beim Heise-Software-Service (siehe Anzeigenteil) erhältlich. Unsere Leser sind eingeladen, SuperTape-Anpassungen für weitere Rechner (selbstverständlich gegen Honorar) in c't vorzustellen.

Die Spectrum-Routine benötigt jedoch bereits 175 Takte, selbst wenn gar kein Sprung mehr in eine Warteroutine erfolgt. Daß so ein Sprung nicht ausgeführt werden soll, müßte aber das Programm auch noch feststellen, da ja für 3600 Baud auf jeden Fall eine zusätzliche Warteschleife nötig ist. Für diese Abfrage wäre erst recht keine Zeit mehr vorhanden, so daß die gesamte Laderoutine einschließlich Synchronisation zweimal geschrieben werden müßte, einmal für 7200 Baud und einmal für 3600 Baud.

Das Spectrum-Programm birgt auch noch eine Gefahr für andere Rechner: Beim Einlesen des Ports erstreckt sich die Abfrage über den gesamten Datenbus. Eine eventuell 'floatende' Datenleitung verfälscht das

Ergebnis völlig. Eine zusätzliche Ausmaskierung des einzulesenden Bits ist also vonnöten.

Über alle Register

Um 7200 Baud auch auf dem Tandy — sogar mit einiger Reserve — lauffähig zu bekommen, muß man fast alle Register der Z80-CPU 'ziehen'. Lediglich das Interrupt- und Refresh-, das IY- und das DE'-Register bleiben unberücksichtigt.

Das neue Interface legt die Signale vom Kassettenrecorder auf die Datenleitung D7. Das hat den Vorteil, daß hierbei bereits mit dem IN-Befehl das Bit abgefragt werden kann. Allerdings muß dafür das Register C die Portadresse enthalten, also IN A,(C). Die Abfrage JP M beziehungsweise JP P wertet dann unmittelbar das eingelesene Bit auf D7 aus. Damit läßt sich die Flankenabfrage innerhalb von nur 22 Takten durchführen, was der Lesesicherheit sehr zugute kommt.

Der eingelesene Wert wird als Carry-Flag in AF' zwischengespeichert. Falls der neu eingelesene Wert mit dem alten übereinstimmt, erkennt das Programm eine Null; bei Ungleichheit eine Eins. Bei einer Eins wird das Prüfsummenregister HL inkrementiert. Das entsprechend gesetzte Carry-Flag gelangt dann in bewährter Manier per Rotation ins E-Register, das so Bit für Bit bis zu dem gewünschten Byte auffgefüllt wird. Das Programm erkennt die 'Fülle' an dem Endflag, das am Anfang (Adresse BCC9h) gesetzt wird und dann nach acht Rotationen ins Carry gelangt.

Beim bitweisen Einlesen innerhalb der ersten Synchronisationsstufe findet — bedingt durch den Befehl SET E,0 — nur eine Rotation statt.

Der indirekte Sprung über das IX-Register ermöglicht eine einfache und schnelle Verzweigung (8 Takte) in eine Warteschleife, die der gewählten Baudrate entspricht. Bei 7200 Baud läßt sich so die Wartezeit im 'Viertakterhythmus' durch Einsprung in die NOP-Reihe von LDW36 bis LDSAM verändern. Der Benutzer von SuperTape sollte sowohl diese Einsprungadresse für 7200 Baud (in Adresse BC25h) als auch die Wartekonstante LCON36 für 3600 Baud etwas variieren, um die für seinen Recorder besten Werte zu finden.

Immerhin 22 Takte ließen sich gegenüber den Spectrum-Routinen dadurch sparen, daß für die Blocklade- und Vergleichsoperationen die gestrichenen Register herangezogen wurden. Die zeitkritische Situation am Ende der Synchronisation konnte entschärft werden, da die Routine gleich in die LDBYT-Routine 'überläuft' und somit bereits mit einem geladenen Byte zurückkehrt. Dieses eine Byte ist allerdings extra zu behandeln, bevor die eigentliche Lade- beziehungsweise Verify-Schleife beginnt.

Summa summarum kommt die Laderoutine einschließlich des JP (IX)-Befehls mit 139 Takten 'Eigenzeit' aus.

Gut ge'toggle't

Beim SAVE liegen die Dinge wesentlich einfacher. Da das Interface auch noch die Arbeit des 'toggelns' übernimmt, also bei jedem Aufruf den Portzustand ändert, wird das sonst nötige Toggle-Register frei. Dieses findet im Programm Verwendung als Flagregister, das das Baudratenflag schnell abrufbar speichert. Bei 3600 Baud (Bit 7 von D = 0) wird dann eine zusätzliche Wartezeit von 124 Takten eingeschoben.

Der Rahmen

Den Rahmen um die eigentlichen SuperTape-Routinen bildet ein menügesteuerter Programmteil, der die Anwendung von SuperTape komfortabel macht. Nach dem Laden von SuperTape meldet sich der TRS-80 mit:

```
SUPERTAPE FUER TRS-80, MODELL 1
(S)AVE (Q)UICK (L)OAD (V)ERIFY (E)XIT
COMMAND:
```

Durch die Eingabe eines in Klammern stehenden Buchstabens verzweigt das Programm zu der entsprechenden Routine und fordert die eventuell noch benötigten Angaben an. Gibt man zum Beispiel 'S' für Speichern mit 3600 Baud (oder Q — Speichern mit 7200 Baud) ein, fragt das Programm nach dem Namen des abzuspeichernden Programms:

```
NAME........TYP
```

Allerdings sollte man bei 7200 Baud einen besseren Recorder verwenden als das Tandy-Gerät.

Nach der Eingabe des Namens gelangt man durch Betätigen der 'RETURN'-Taste zur TYP-Abfrage, zum Beispiel 'BAS' für BASIC-Programme oder 'HEX' für Speicherinhalte. Beim Typ 'HEX' fragt das Programm nach der Start- und Endadresse des Speicherbereichs. Diese Frage entfällt beim Typ 'BAS'; das Programm schreibt ein BASIC-Programm vom Anfang bis zum Ende (im Rechner durch Zeiger markiert) auf Band.

Bei der folgenden Frage 'TAPE READY ?' hat man noch die Möglichkeit, sofort wieder zum Menü zurückzukehren, indem man diese Frage mit 'N' beantwortet. Andernfalls schreibt das Programm die Daten auf Band.

Alle Funktionen, die man schon einmal ausgeführt hat, kann man im Menü durch die Eingabe von 'Shift Buchstabe' erneut aufrufen, wobei die zuvor eingegebenen Parameter übernommen und angezeigt werden.

Soll die Aufzeichnung sogleich überprüft werden, genügt die Eingabe von 'v' (Shift), da die Parameter ja dieselben wie beim Abspeichern sind. Nach der Frage 'TAPE READY ?' erscheint in der untersten Bildschirmzeile ein Sternchen, das zu 'flackern' beginnt, sobald Signale vom Band kommen. Während das Sternchen 'blinkt', kann man den Ladevorgang durch Betätigen der 'BREAK'-Taste abbrechen und zurück zum Menü gelangen. Als Abschluß der Verify-Routine listet das Programm die Daten des 'geladenen' Programms und gibt eine Lade-/Prüfmeldung neben dem Sternchen aus:

- #∗ Fehlerfrei
- B∗ Blockfehler
- >∗ Fehler bei Verify
- F∗ Prüfsummenfehler

Beim Load-Befehl 'L' fragt das Programm nach der Eingabe des Namens noch zusätzlich nach einer neuen Startadresse. Somit kann man ein Programm an beliebiger Stelle im Speicher ablegen. Bei der Nameneingabe erlaubt das Programm die Verwendung von 'Jokern'; das Zeichen '∗' steht für einen beliebig langen Namen mit beliebigen Buchstaben. Die einzige Ausnahme hiervon bilden BASIC-Programme: Sie sollten immer mit der Typ-Angabe 'BAS' geladen werden, da nur dann die Zeiger für den späteren Start des BASIC-Programms richtig gesetzt werden. Die Zeiger 'stehen' zwar richtig, wenn man ein mit dem DISK-BASIC erstelltes Programm wieder in diesem BASIC starten beziehungsweise listen will; sie stimmen aber nicht, wenn ein ROM-BASIC-Programm unter DISK-BASIC laufen soll (oder umgekehrt).

Nach dem erfolgreichen Laden eines BASIC-Programms verzweigt das SuperTape-Programm in das BASIC: Hat man SuperTape aus dem ROM-BASIC gestartet, springt das Programm auch in dieses BASIC. Arbeitet man mit Disketten und hat SuperTape nicht aus dem DISK-BASIC aufgerufen, erscheint kurz die Fehlermeldung 'KEIN BASIC' und anschließend das Menü. Soll ein BASIC-Programm nach dem Laden ausgeführt werden, ist folgendermaßen vorzugehen:

Vom DOS aus mit dem Befehl 'LOAD SUTRS/CMD' SuperTape laden.

Anschließend muß man BASIC aufrufen.

Im BASIC den Befehl 'SYSTEM' eingeben und auf die Meldung '?∗' mit '/47360' antworten — SuperTape wird gestartet.

Einen direkten Rücksprung ins BASIC oder DOS kann man durch die Eingabe von 'E' im SuperTape-Programm erreichen. Arbeitet man mit Disketten und hatte vor dem Aufruf von SuperTape BASIC geladen, führt 'E' direkt ins DISK-BASIC. Ohne geladenes BASIC gelangt man durch die Eingabe von 'E' in die Befehlsebene vom DOS, beim Arbeiten im ROM-BASIC erfolgt ein BASIC-Warmstart.

Die Unterscheidung zwischen diesen drei 'Betriebssystemen' erfolgt selbsttätig durch das Programm. Dazu wertet es ein Byte aus dem 'Keyboard DCB' (Zeiger auf die Tastatur-Treiberroutine) aus.

Beim TRS-80, Modell 1 mit dem 'Level 2 ROM-BASIC' steht in dieser Speicherzelle (4017h) der Wert 03 beim aktivierten ROM-BASIC, beim Betrieb des TRS-80 unter NEWDOS 40 erwartet das Programm das Byte 43h in der Speicherzelle. Ein Sprung in das DISK-BASIC erfolgt dann, wenn die zuvor genannten Werte nicht erkannt werden konnten.

Interface

Die Adreßdekodierung des Interfaces besteht aus IC1 und IC2. Die ausgewerteten Adressen schalten entweder ein Flipflop (IC3a) oder erlauben Ein-/Ausgaben über den Puffer IC4. Das Flipflop IC3a schaltet bei jedem Zugriff auf den Port 7Fh (127d) um, unabhängig von den ausgegebenen Daten. Über die drei Inverter (IC5a...IC5c) und das Filter (R2, R3, C1, C2) gelangt das Ausgangssignal des Flipflops an den Kassettenrecorder.

Die Datenleitungen D0, D1, D6 und D7 des TRS-80 liegen über den Puffer IC4 an IC3a und IC3b. Der logische Pegel der Datenleitung D0 gelangt über die Brücke J1 an den 'Clear'-Eingang von IC3a. Legt man diesen Eingang kurzfristig auf logisch 0, schaltet der Ausgang von IC3a auf logisch 1. Diese Funktion benötigt man aber nur dann, wenn mehr als zwei Rechner mit einer 'SuperTape-Ringleitung' verbunden werden sollen. Im Normalfall kann man die Brücke 1 offenlassen und an der Leiste X3 den Impuls von D0 abgreifen.

Mit dem Datenbit D1 kann man das Flipflop IC3b steuern. So setzt zum Beispiel der BASIC-Befehl 'OUT 126,2' den Ausgang von IC3b auf logisch 1. Schließt man an diesen Ausgang eine Treiberstufe und ein Relais an, kann man den Kassettenrecorder programmgesteuert ein- und ausschalten.

Das Signal vom Kassettenrecorder gelangt über die Triggerschaltung (IC5d, IC5e) an die Datenleitung D7. Der Pegel dieser Leitung entspricht nach einer Portabfrage dem Eingangssignal am Kassetten-Eingang. Die Leitung D6 liegt über den Puffer an der Anschlußleiste X3, Pin 2. Hier kann man ein Eingangssignal mit TTL-Pegel in den Rechner einlesen.

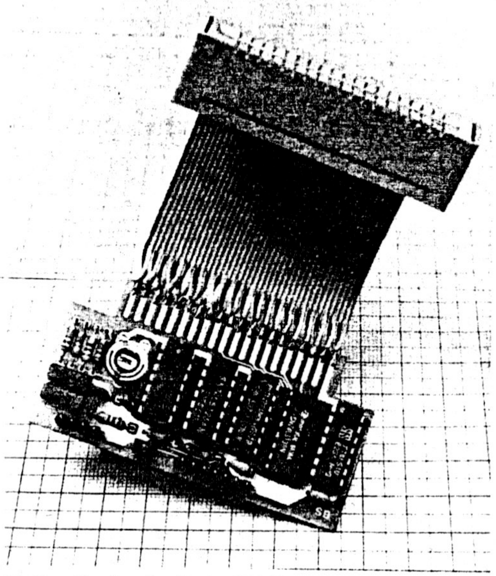
Aufbau

Da die Leiterplatte sehr eng bestückt wird, sollte man beim Aufbau besonders sorgfältig vorgehen. Der Anschluß des Interfaces an den Rechner geschieht über die Steckerleiste X4. Die Platine ist so ausgelegt, daß man eine 40polige Steckerleiste direkt anlöten kann. Soll das Interface an den 'Interface-Bus' (J3) des Tandy-Expansion-Interface angeschlossen werden, hat diese Lösung allerdings den Nachteil, daß die Platine 'in der Luft schwebt'. Beim direkten Anschluß an den TRS-80-Bus kann man der Leiterplatte durch zwei Distanzrollen festen Halt geben.

Die wohl bessere Lösung besteht darin, die Platine über ein Stück Flachbandkabel und einen 'Preßstecker' mit dem Rechner zu verbinden. Allerdings muß das Kabel so kurz wie möglich sein, da sonst der Tandy recht 'eigenwillig' wird.

Test

Nach dem Bestücken sollte man die Platine sehr sorgfältig auf Kurzschlüsse und Unterbrechungen untersuchen. Anschließend kann man die Karte auf den Bus des Rechners oder des Expansion-Boards stecken



Das SuperTape-Interface für den TRS-80.

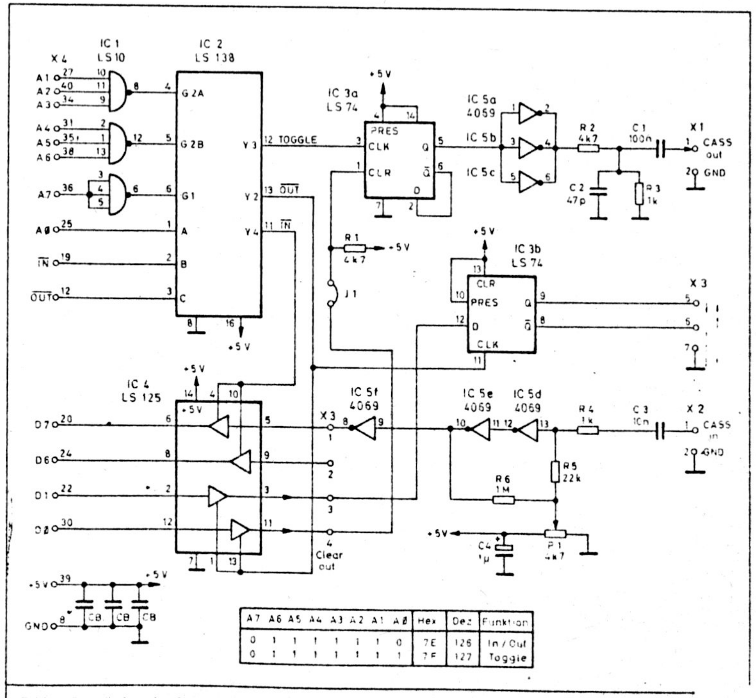


Bild 1. Schaltplan des TRS-80-Interfaces.

und den Computer einschalten: Das System sollte sich wie gewohnt verhalten. Andernfalls ist der Computer sofort abzuschalten und das Interface erneut zu überprüfen.

Tritt dieser Fall nicht ein, können die weiteren Tests des Interfaces mit BASIC-Befehlen und einem Voltmeter durchgeführt werden. Dazu sollte man die Brücke J1 schließen und ein Voltmeter an Pin 2, 4 oder 6 von IC5 anschließen. Nach jeder Ausführung des BASIC-Befehls 'OUT 127,0' muß der Pegel an IC5 wechseln; von circa 0 Volt auf etwa 5 Volt und umgekehrt. Für den nächsten Test sollte die Spannung an IC5 etwa 0 Volt betragen. Mit der Ausführung der Anweisung 'OUT 126,0' muß an IC5 der Pegel auf rund 5 Volt wechseln (Reset-Funktion).

Anschließend ist das Voltmeter an Pin 5 von der Steckerleiste X3 anzuschließen. Der Befehl 'OUT 126,2' bewirkt, daß der Ausgang von IC3b auf logisch 1 (etwa 5 Volt) springt. Mit der Anweisung 'OUT 126,0' ergibt sich eine Spannung von rund 0 Volt an dem Ausgang dieses ICs.

Für die Überprüfung des Kassetten-Eingangs gibt man folgendes Programm ein:

```
10 A = INP(126)
20 PRINT A;
30 GOTO 10
```

Nach dem Start des Programms wird ein Wert angezeigt, der sich ändern muß, wenn man P1 an den rechten oder linken Anschlag dreht. Die richtige Einstellung von P1 liegt an dem Punkt, wo die Anzeige zwischen den beiden Werten springt. □

Stückliste

Widerstände		IC2	74LS138
R1,2	4k7	IC3	74LS74
R3,4	1k	IC4	74LS125
R5	22k	IC5	CD 4069
R6	1M	**Sonstiges**	
P1	4k7 Trimmer, liegend	X1, X2, J1	Pfostenleiste, 2polig
Kondensatoren		X3	Pfostenleiste, 7polig
C1	100n	X4	40poliger 'Card-Edge'-Stecker, 2reihig, Raster 2,54 mm
C2	47p		
C3	10n		
C4	1µ, Tantal		
CB	3 Stützkondensatoren, 100n, ker.		
Halbleiter			
IC1	74LS10		

Brückenstecker für Pfostenleiste; IC-Fassungen: 4 x 14polig, 1 x 16polig; Platine 'TRS-80 SuperTape'.

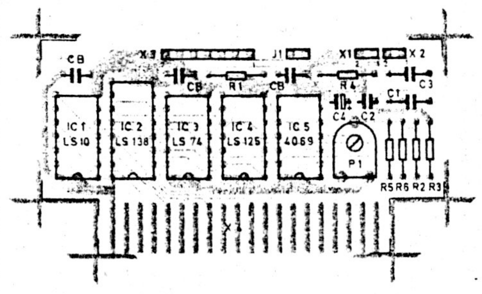


Bild 2. Der Bestückungsplan. Da die ICs sehr dicht beieinander liegen, müssen 'anreihbare' Fassungen verwendet werden.

```
B900                 ORG     0B900H
0033        DISPL    EQU     00033H
0049        RDTAS    EQU     00049H
0066        READY    EQU     00066H
01C9        CLS      EQU     001C9H
401A        CURSYM   EQU     0401AH
40A4        HEADP    EQU     040A4H
40F9        TAILP    EQU     040F9H
40FB        VARANF   EQU     040FBH
40FD        VAREND   EQU     040FDH
3FF0        BILDP1   EQU     03FF0H
3FF1        BILDP2   EQU     03FF1H
3840        ZTAST    EQU     03840H
000B        LCON36   EQU     0BH
007E        PORTIN   EQU     07EH
007F        PORTOU   EQU     07FH
06CC        ROMBAS   EQU     06CCH
06CC        DOSBAS   EQU     06CCH
402D        DOS      EQU     402DH
FFFF        STACK    EQU     0FFFFH
4017        ECHEC    EQU     04017H
            ;
            ;
            ;**************************************************
            ;SUPERTAPE TANDY TRS-80, MODELL 1
            ;MENUEAUSWAHL
            ;
            ;**************************************************
            ;
            ;
            ;
B900 31FFFF START:   LD      SP,STACK
B903 CD2BB9 START1:  CALL    CLSRE           ;CLEAR BILDSCHIRM
B906 215DBD START2:  LD      HL,TEXT1        ;TEXT
B909 CDEFB9          CALL    PRTEXT          ;ZUM
B90C CD4900 TAST:    CALL    RDTAS           ;BILDSCHIRM
B90F 4F              LD      C,A
B910 E65F            AND     5FH
B912 FE53            CP      053H
B914 2868            JR      Z,SAVE          ;'S'
B916 FE51            CP      51H             ;'Q'
B918 2864            JR      Z,SAVE
B91A FE4C            CP      04CH
B91C CA09BA          JP      Z,LOAD          ;'L'
B91F FE56            CP      056H
B921 CAF9B9          JP      Z,VERI          ;'V'
B924 FE45            CP      045H            ;'E'
B926 CA19BD          JP      Z,BWARM         ;ZUM BASIC ODER DOS
B929 18E1            JR      TAST            ;SONST ZUR
                                             ;TATATURABFRAGE
            ;
            ;**************************************************
            ;BILDSCHIRM LOESCHEN UND CURSOR SETZEN
            ;**************************************************
B92B CDC901 CLSRE:   CALL    CLS             ;CLEAR  SCREEN
B92E 3E67            LD      A,067H          ;NEUES CURSOR-
B930 321A40          LD      (CURSYM),A      ;SYMBOL
B933 3E0E            LD      A,00EH          ;CURSOR
B935 CD3300          CALL    DISPL           ;EIN
B938 C9              RET
            ;
            ;**************************************************
            ;TASTATURABFRAGE FUER LD NAME/TYP
            ;**************************************************
B939 7E     RDNIT:   LD      A,(HL)
B93A CB69            BIT     5,C             ;GROSS ODER KLEIN?
B93C 200C            JR      NZ,RDCONT
B93E CD4900          CALL    RDTAS           ;TASTATURABFRAGE
B941 FE0D            CP      00DH            ;RETURN ?
B943 280C            JR      Z,RETUR         ;WENN JA SPRUNG
B945 FE08            CP      008H            ;CURSOR -
B947 2812            JR      Z,CURS          ;WENN JA SPRUNG
B949 77              LD      (HL),A          ;ZEICHEN ABLEGEN
B94A CD3300 RDCONT:  CALL    DISPL           ;UND ZUM BILD
B94D 23              INC     HL
B94E 10E9            DJNZ    RDNIT           ;WEITER BIS B=0
B950 C9              RET
            ;**************************************************
            ; RETURN WURDE GEDRUECKT
            ;**************************************************
B951 3E20   RETUR:   LD      A,020H          ;FUELLE REST MIT
B953 77              LD      (HL),A          ;SPACE AUF
B954 CD3300          CALL    DISPL
B957 23              INC     HL
B958 10F7            DJNZ    RETUR
B95A C9              RET
            ;**************************************************
            ; CURSOR ZURUECKSETZEN
            ;**************************************************
B95B 2B     CURS:    DEC     HL              ;
B95C 2B              DEC     HL              ;CURSOR
B95D 04              INC     B               ;EIN ZEICHEN
B95E 04              INC     B               ;ZURUECKSETZEN
B95F 3E18            LD      A,018H
B961 18E7            JR      RDCONT
            ;**************************************************
            ;LADE HEXADRESSE NACH HL
            ;**************************************************
B963 CD9BBA HEXADR:  CALL    ASIBYT          ;LAEDT 4STELLIGE
B966 67              LD      H,A             ;HEXADRESSE UEBER
B967 CD9BBA          CALL    ASIBYT          ;ASCII/HEX-WANDLER
B96A 6F              LD      L,A             ;VON TASTATUR
B96B C9              RET                     ;NACH HL
            ;**************************************************
            ;LADE NAME UND TYP
            ;**************************************************
B96C 212BBD NAITY:   LD      HL,SVNAME       ;NAME UND TYP
B96F 060C            LD      B,00CH          ;WERDEN IN DEN
B971 CD39B9          CALL    RDNIT           ;SV-PUFFER
B974 3E2E            LD      A,02EH          ;GELADEN
B976 77              LD      (HL),A
B977 23              INC     HL
B978 0603            LD      B,003H
B97A CD39B9          CALL    RDNIT
B97D C9              RET
            ;**************************************************
            ;SAVE
            ;**************************************************
B97E F5     SAVE:    PUSH    AF              ;RETTE BEFEHL
B97F 2182BD          LD      HL,TEXT2        ;TEXT
B982 CDEFB9          CALL    PRTEXT          ;ZUM BILD
B985 CD6CB9          CALL    NAITY           ;LADE BLOCKPARAMETER
B988 3A38BD          LD      A,(SVTYP)
B98B FE42            CP      042H            ;WENN TYP=B(BASIC)
B98D 2832            JR      Z,TIBAS         ;HOLE BASIC-POINTER
B98F CB69            BIT     5,C             ;ALTE WERTE?
B991 203E            JR      NZ,SAVEA
B993 21C9BD          LD      HL,TEXT3        ;
B996 CDEFB9          CALL    PRTEXT          ;LADE STARTADRESSE
B999 CD63B9          CALL    HEXADR          ;IN SV-PUFFER
B99C 223CBD          LD      (SVSTA),HL      ;
B99F E5              PUSH    HL              ;
B9A0 3E20            LD      A,020H          ;EIN SPACE ZUM
B9A2 CD3300          CALL    DISPL           ;BILD
B9A5 CD63B9          CALL    HEXADR          ;LADE ENDADRESSE
B9A8 C1              POP     BC              ;UND
B9A9 ED42   DTLEN:   SBC     HL,BC           ;ERRECHNE LAENGE
B9AB 223EBD          LD      (SVLEN),HL      ;FUER PUFFER
B9AE 3E00   STFLA:   LD      A,000H          ;
B9B0 213BBD          LD      HL,SVFLAG       ;BAUDRATE SETZEN
B9B3 77              LD      (HL),A          ;FLAG 00 SETZEN
B9B4 F1              POP     AF              ;HOLE BEFEHL
B9B5 FE51            CP      51H
B9B7 2002            JR      NZ,NOQ          ;NEIN: BAUDRATE 0
B9B9 3680            LD      (HL),80H        ;SONST 80H
B9BB CD0BB9 NOQ      CALL    TAPERY          ;TAPE READY?
B9BE C345B8          JP      SVSUP           ;DANN SAVEN
            ;**************************************************
            ; TYP BASIC
            ;**************************************************
B9C1 2AA440 TIBAS:   LD      HL,(HEADP)      ;FALLS TYP =B
B9C4 223CBD          LD      (SVSTA),HL      ;WERDEN ADRESSEN
B9C7 E5              PUSH    HL              ;FUER START/LAENGE
B9C8 ED4BF940        LD      BC,(TAILP)      ;VOM HEAD/TAILPOINTER
B9CC C5              PUSH    BC              ;ABGELEITET UND
B9CD E1              POP     HL              ;AUTOMATISCH IN DEN
B9CE C1              POP     BC              ;SV-PUFFER GESETZT
B9CF 18D8            JR      DTLEN
            ;
B9D1 213DBD SAVEA:   LD      HL,SVSTA+1
B9D4 0602            LD      B,02H
B9D6 CD04BB          CALL    LILP2
B9D9 18D3            JR      STFLA
            ;**************************************************
            ;TAPE READY ?
            ;**************************************************
B9DB 2106BD TAPERY:  LD      HL,TEXT4        ;HOLE TEXTADRESSE
B9DE CDEFB9          CALL    PRTEXT          ;TEXT ZUM BILD
B9E1 CD4900          CALL    RDTAS           ;TASTATURABFRAGE
B9E4 FE4E            CP      04EH            ;'N' NEIN: BREAK
B9E6 CA00B9          JP      Z,START         ;DANN ZUM MENUE
B9E9 3E0F            LD      A,00FH          ;SONST RET
B9EB CD3300          CALL    DISPL
B9EE C9              RET
            ;**************************************************
            ;MENUEGENERATOR (HL) ZUM BILD
            ;**************************************************
B9EF 7E     PRTEXT:  LD      A,(HL)          ;PRINT WAS IN (HL)
B9F0 FE23            CP      023H            ;STEHT ZUM BILD
B9F2 C8              RET     Z               ;BIS ENDMARKE
B9F3 CD3300          CALL    DISPL           ;ERREICHT
B9F6 23              INC     HL
B9F7 18F6            JR      PRTEXT
            ;**************************************************
            ;VERIFY
            ;**************************************************
B9F9 21E6BD VERI:    LD      HL,TEXT5        ;TEXT
B9FC CDEFB9          CALL    PRTEXT          ;ZUM BILD
B9FF CD6CB9          CALL    NAITY           ;HOLE BLOCKPARAMETER
BA02 3E02   VECONT:  LD      A,002H          ;SETZE 02 IN
BA04 322ABD          LD      (ZFLAG),A       ;ZFLAG
BA07 1833            JR      TPRDY           ;TAPE READY
            ;
            ;**************************************************
            ;LADEAUFRUF LOAD
            ;**************************************************
BA09 21FEBD LOAD:    LD      HL,TEXT6        ;TEST 'LOAD' USW.
BA0C CDEFB9          CALL    PRTEXT          ;ZUM BILD
BA0F CD6CB9          CALL    NAITY           ;NAME + TYP EINGEBEN
BA12 3A38BD          LD      A,(SVTYP)
BA15 FE42            CP      42H             ;'B'?
BA17 200D            JR      NZ,LOHEX
BA19 2AA440          LD      HL,(40A4H)
BA1C 223CBD          LD      (SVSTA),HL
BA1F 212ABD          LD      HL,ZFLAG
BA22 CBD6            SET     2,(HL)
BA24 1816            JR      TPRDY
            ;
BA26 2115BE LOHEX:   LD      HL,TEXT7        ;TEXT
BA29 CDEFB9          CALL    PRTEXT          ;'NEUE STARTADRESSE'
BA2C CB69            BIT     5,C             ;ALTE WERTE?
BA2E 2032            JR      NZ,LOADA
BA30 CD4900          CALL    RDTAS           ;TASTATURABFRAGE
BA33 FE4A            CP      04AH            ;JA?
BA35 2818            JR      Z,NADR          ;DANN ZUR EINGABE
BA37 3E00   LDRDY:   LD      A,000H          ;ZFLAG AUF NULL
BA39 322ABD          LD      (ZFLAG),A       ;
BA3C CD0BB9 TPRDY:   CALL    TAPERY          ;'TAPE READY'
BA3F 3E2A            LD      A,02AH          ;STERNCHEN
BA41 32F13F          LD      (BILDP2),A      ;SETZEN
BA44 CD0BB8          CALL    LDSUP           ;LADEN
BA47 3229BD          LD      (ERROR),A       ;FEHLERMELDUNG RETTEN
BA4A CDE2BA          CALL    LIIPU           ;LD-PUFFER AUSGEBEN
BA4D 182B            JR      LDCHK           ;ZUM LADECHECK
            ;-----------NEUE STARTADRESSE EINGEBEN
BA4F 2126BE NADR:    LD      HL,TEXT8        ;TEXT 'ADRESSE'
BA52 CDEFB9          CALL    PRTEXT          ;ZUM BILD
BA55 CD63B9          CALL    HEXADR          ;LADE ADRESSE
BA58 223CBD          LD      (SVSTA),HL      ;IN SV-PUFFER
BA5B 3E04            LD      A,004H          ;SETZE FLAG 04
BA5D 322ABD          LD      (ZFLAG),A       ;IN ZFLAG
BA60 18DA            JR      TPRDY           ;UND WEITER
            ;-----------LADEAUFRUF "L" OHNE BLOCKEINGABE
BA62 3A2ABD LOADA:   LD      A,(ZFLAG)
BA65 CB57            BIT     2,A
BA67 280A            JR      Z,LADR
BA69 213DBD          LD      HL,SVSTA+1
BA6C 0601            LD      B,001
BA6E CD04BB          CALL    LILP2
BA71 18C9            JR      TPRDY
BA73 3E2A   LADR:    LD      A,02AH
BA75 CD3300          CALL    DISPL
BA78 18BD            JR      LDRDY
            ;**************************************************
            ;LDCHECK;AUSWERTUNG LOAD/VERIFY
            ;**************************************************
BA7A 3A29BD LDCHK:   LD      A,(ERROR)       ;FEHLERMELDUNG
BA7D 32F03F          LD      (BILDP1),A      ;ABLEGEN
BA80 FE42            CP      042H            ;'B' ? =BLOCKFEHLER
BA82 28B8            JR      Z,TPRDY         ;ZURUECK ZUR AUFNAHME
BA84 FE23            CP      023H            ;'#' LADE ZFLAG
BA86 2803            JR      Z,LDOK
BA88 C306B9 RUECK:   JP      START2          ;ZUM MENUE
                     ;
BA8B 3A2ABD LDOK:    LD      A,(ZFLAG)       ;LADE ZFLAG
BA8E FE02            CP      002H            ;VERIFY?
BA90 28F6            JR      Z,RUECK         ;DANN ZUM MENUE
BA92 3A51BD          LD      A,(LDTYP)       ;TYP LADEN
BA95 FE42            CP      042H            ;B(BASIC) ?
BA97 20EF            JR      NZ,RUECK        ;JA ? DANN WEITER
BA99 187B            JR      STBAS           ;UND BASIC-POINTER
            ;                                ;SETZEN
            ;**************************************************
            ;ASCII/HEX WANDLER
            ;**************************************************
BA9B CDA8BA ASIBYT:  CALL    ASINIB          ;
BA9E 07              RLCA                    ;ROUTINE WANDELT
BA9F 07              RLCA                    ;ZWEI ASCII-ZEICHEN
BAA0 07              RLCA                    ;IN EINE ZWEISTELLIGE
BAA1 07              RLCA                    ;HEX-ZAHL
BAA2 4F              LD      C,A             ;
BAA3 CDA8BA          CALL    ASINIB          ;
BAA6 B1              OR      C               ;ERGEBNIS IN A
BAA7 C9              RET
            ;
BAA8 CD4900 ASINIB:  CALL    RDTAS           ;EIN ZEICHEN WIRD
BAAB F5              PUSH    AF              ;VON DER TASTATUR
BAAC CD3300          CALL    DISPL           ;GELESEN UND ALS
BAAF F1              POP     AF              ;EIN HEX-NIBBLE
BAB0 D630            SUB     030H            ;INTERPRETIERT
BAB2 D8              RET     C
BAB3 C6E9            ADD     A,0E9H
BAB5 D8              RET     C
BAB6 C606            ADD     A,006H
BAB8 F2BEBA          JP      P,ASICON
BABB C607            ADD     A,007H
BABD D8              RET     C
BABE C60A   ASICON:  ADD     A,00AH
BAC0 B7              OR      A
BAC1 C9              RET
            ;
BAC2 7E     BYTIAS:  LD      A,(HL)          ;
BAC3 F5              PUSH    AF              ;ROUTINE WANDELT
BAC4 D9              EXX                     ;ZWEISTELLIGE
BAC5 0F              RRCA                    ;HEXZAHL
BAC6 0F              RRCA                    ;IN ZWEI ASCII-
BAC7 0F              RRCA                    ;ZEICHEN UND
BAC8 0F              RRCA                    ;GIBT DIESE
BAC9 E60F            AND     00FH            ;ZUM BILD
BACB CDD6BA          CALL    NIBIAS          ;AUS
BACE F1              POP     AF
BACF E60F            AND     00FH
BAD1 CDD6BA          CALL    NIBIAS
BAD4 D9              EXX
BAD5 C9              RET
            ;
BAD6 C690   NIBIAS:  ADD     A,090H          ;BERECHNET ASCII
BAD8 27              DAA                     ;AUS NIBBLE UND
BAD9 CE40            ADC     A,040H          ;BRINGT DAS ZEICHEN
BADB 27              DAA                     ;AUF DEN BILDSCHIRM
BADC F5              PUSH    AF
BADD CD3300          CALL    DISPL
BAE0 F1              POP     AF
BAE1 C9              RET
            ;**************************************************
            ;LIST PUFFER
            ;**************************************************
BAE2 3E0D   LIIPU:   LD      A,00DH          ;CR
BAE4 CD3300          CALL    DISPL           ;ZUM BILD
BAE7 2144BD          LD      HL,LDNAME       ;LADE ADR AUS LD-PUFFER
BAEA 0610            LD      B,010H          ;LAENGE 10H FUER
BAEC 7E     LILP1:   LD      A,(HL)          ;NAME UND TYP
BAED CD3300          CALL    DISPL           ;ZUM BILD
BAF0 23              INC     HL              ;WEITER
BAF1 10F9            DJNZ    LILP1           ;BIS B=0
```

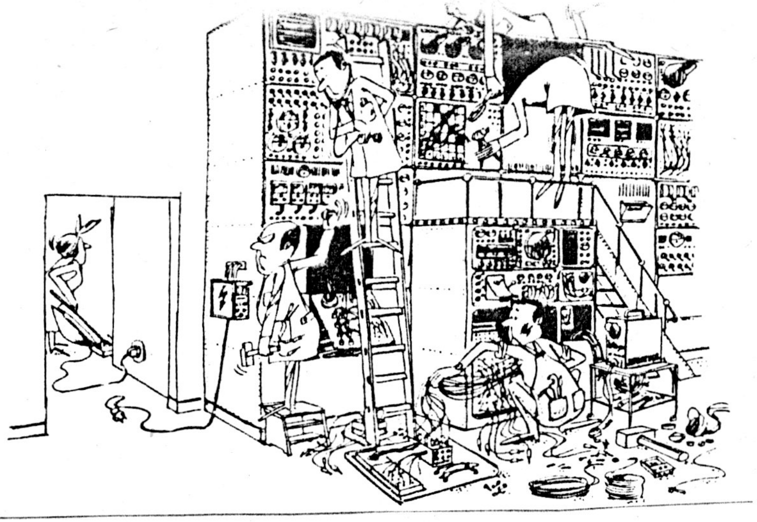
```
BAF3 2A55BD            LD      HL,(LDSTA)      ;LADE STARTADR
BAF6 ED4B57BD          LD      BC,(LDLEN)      ;UND LAENGE
BAFA ED4A              ADC     HL,BC           ;ADDIEREN
BAFC 2257BD            LD      (LDFREI),HL     ;END-ADR ABLEGEN
BAFF 2156BD            LD      HL,LDSTH        ;ZEIGE AUF ADRESSEN
BB02 0603              LD      B,003H          ;
BB04 3E20     LILP2:   LD      A,020H          ;AUSGABESCHLEIFE
BB06 CD3300            CALL    DISPL           ;FUER STARTADRESSE
BB09 CDC2BA            CALL    BYT1AS          ;LAENGE
BB0C 2B                DEC     HL              ;UND ENDADRESSE
BB0D CDC2BA            CALL    BYT1AS
BB10 23                INC     HL
BB11 23                INC     HL
BB12 23                INC     HL
BB13 10EF              DJNZ    LILP2
BB15 C9                RET
              ;*********************************************
              ; SET BASIC-POINTER
              ;*********************************************
BB16 3A1740   STBAS:   LD      A,(ECHEC)       ;TEST OB BASIC
BB19 FE43              CP      43H             ;GELADEN IST
BB1B 2817              JR      Z,NOBAS         ;Z=NEIN
BB1D 2131BE            LD      HL,TEXT9        ;TEXT 'START BASIC'
BB20 CDEFB9            CALL    PRTEXT          ;AUSGEBEN
BB23 0129BB            LD      BC,WARM
BB26 C3AE19            JP      19AEH
BB29 2A55BD   WARM:    LD      HL,(LDSTA)
BB2C ED5B57BD          LD      DE,(LDLEN)
BB30 19                ADD     HL,DE
BB31 C3772C            JP      2C77H
              ;
              ;KEIN BASIC GELADEN
BB34 213FBE   NOBAS:   LD      HL,TEXT10       ;TEXT 'KEIN BASIC'
BB37 CDEFB9            CALL    PRTEXT          ;AUSGEBEN
BB3A 010000            LD      BC,0000
BB3D 0B       NOBALO:  DEC     BC
BB3E 78                LD      A,B
BB3F B1                OR      C
BB40 20FB              JR      NZ,NOBALO
BB42 C300B9            JP      START
              ;*********************************************
              ;SVSUP
              ;*********************************************
BB45 F3       SVSUP:   DI                      ;INTERRUPTS ABSCHALTEN
BB46 212BBD            LD      HL,SVNAME       ;SENDET PARAMETERBLOCK
BB49 011900            LD      BC,00019H       ;
BB4C 3E2A              LD      A,02AH          ;
BB4E 1600              LD      D,0             ;FUER 3600 BAUD
BB50 CD66BB            CALL    SVBLO           ;SENDET DATENBLOCK
BB53 2A3CBD            LD      HL,(SVSTA)      ;
BB56 ED4B3EBD          LD      BC,(SVLEN)      ;
BB5A 3A3BBD            LD      A,(SVFLAG)      ;HOLE BAUDRATE
BB5D 57                LD      D,A             ;NACH   D
BB5E 3EC5              LD      A,0C5H
BB60 CD66BB            CALL    SVBLO
BB63 C306B9            JP      START2
              ;*********************************************
              ;SVBLO
              ;*********************************************
BB66 F5       SVBLO:   PUSH    AF              ;RETTE ZUM STACK
BB67 0B                DEC     BC              ;DECR. BLOCKLNGE
BB68 C5                PUSH    BC              ;UND RETTE ZUM STACK
BB69 0640              LD      B,040H          ;SENDE 64 SYNC-ZEICHEN
BB6B 1E16     SVLO1:   LD      E,016H          ;LADE  E MIT SYNC-BYTE
BB6D A7                DEFB    0A7H            ;WARTE 13 TAKTE
BB6E A7                DEFB    0A7H
BB6F D8                DEFB    0D8H
BB70 CDA2BB            CALL    SVBYT
BB73 10F6              DJNZ    SVLO1
BB75 C1                POP     BC              ;HOLE LAENGE VOM STACK
BB76 F1                POP     AF              ;HOLE SYNC-ABSCHLUSS
BB77 5F                LD      E,A             ;SENDE SYNC-ABSCHLUSS
BB78 CDA2BB            CALL    SVBYT
BB7B DD210000          LD      IX,00000H       ;INIT PRUEFSUMME
BB7F A6                DEFB    0A6H            ;WARTE 12 TAKTE
BB80 D8                DEFB    0D8H
BB81 5E       SVLO2:   LD      E,(HL)          ;SENDE BYTE
BB82 CDA2BB            CALL    SVBYT           ;
BB85 EDA1              CPI                     ;INC HL,DEC BC
BB87 EA81BB            JP      PE,SVLO2        ;SCHLEIFE BIS BC=0
BB8A 5E                LD      E,(HL)          ;SENDE LETZTES BYTE
BB8B CDA2BB            CALL    SVBYT           ;
BB8E C0                DEFB    0C0H            ;WARTE 5 TAKTE
BB8F DDE5              PUSH    IX              ;SENDE PRUEFSUMME
BB91 C1                POP     BC              ;
BB92 59                LD      E,C
BB93 CDA2BB            CALL    SVBYT
BB96 E5                DEFB    0E5H            ;WARTE 29 TAKTE
BB97 E1                DEFB    0E1H
BB98 A7                DEFB    0A7H
BB99 A7                DEFB    0A7H
BB9A 58                LD      E,B
BB9B CDA2BB            CALL    SVBYT
BB9E E3                DEFB    0E3H            ;WARTE 50 TAKTE UND
BB9F E3                DEFB    0E3H            ;SENDE SCHLUSSBYTE
BBA0 A6                DEFB    0A6H
BBA1 D8                DEFB    0D8H
              ;*********************************************
              ;SVBYTE: SAVE BYTE
              ;*********************************************
BBA2 37       SVBYT:   SCF                     ;SETZE ENDFLAG
BBA3 CB1B     SVBIT:   RR      E               ;AUSGABEBIT IN CARRY
BBA5 3827              JR      C,SVEINS        ;BIT =1? SPRUNG
BBA7 D37F              OUT     (07FH),A        ;UND AUSGEBEN
BBA9 A6                DEFB    0A6H            ;WARTE  14 TAKTE
BBAA A6                DEFB    0A6H            ;
BBAB CB7A     SVCONT:  BIT     7,D
```

```
BBAD CCD3BB            CALL    Z,SVWAIT        ;WARTE FUER 3600 BAUD
BBB0 E3                DEFB    0E3H            ;WARTE WEITERE 80 TAKTE
BBB1 E3                DEFB    0E3H
BBB2 E3                DEFB    0E3H
BBB3 E3                DEFB    0E3H
BBB4 A7                DEFB    0A7H
BBB5 D37F              OUT     (07FH),A        ;AUSGABE ZUM PORT
BBB7 CB7A              BIT     7,D
BBB9 CCD3BB            CALL    Z,SVWAIT        ;WARTE FUER 3600BD
BBBC D8                DEFB    0D8H            ;WARTE 10 TAKTE
BBBD D8                DEFB    0D8H
BBBE 1D                DEC     E               ;8 BITS AUSGEGEBEN?
BBBF C8                RET     Z               ;RETURN FALLS JA
BBC0 1C                INC     E               ;SONST RESTAURIERE E
BBC1 3A4038            LD      A,(ZTAST)       ;WARTE 40 TAKTE
BBC4 FE04              CP      004H            ;UND TESTE BREAK
BBC6 CA00B9            JP      Z,START
BBC9 C8                DEFB    0C8H
BBCA C8                DEFB    0C8H
BBCB AF                XOR     A               ;LOESCHE CARRY
BBCC 18D5              JR      SVBIT           ;WEITER
              ;
BBCE DD23     SVEINS:  INC     IX              ;INKR. PRUEFSUMME
BBD0 C3ABBB            JP      SVCONT
              ;*********************************************
              ; SVWAIT WARTET 117 TAKTE FUER 3600 BAUD
              ;*********************************************
BBD3 3E06     SVWAIT:  LD      A,006           ;WARTE 99 TAKTE
BBD5 3D       WTCON:   DEC     A
BBD6 20FD              JR      NZ,WTCON
BBD8 A7                DEFB    0A7H            ;WEITERE 9 TAKTE
BBD9 D8                DEFB    0D8H
BBDA C9                RET
              ;*********************************************
              ;LDSUP:LADEROUTINE
              ;*********************************************
BBDB 215CBD   LDSUP:   LD      HL,LDEND        ;
BBDE 0619              LD      B,019H
BBE0 3E20              LD      A,020H
BBE2 77       CLCON:   LD      (HL),A          ;LOESCHE LD-PUFFER
BBE3 2B                DEC     HL
BBE4 10FC              DJNZ    CLCON
BBE6 23                INC     HL              ;HL=LDNAM
BBE7 011900            LD      BC,00019H       ;BLOCKLAENGE 19H
BBEA 3E2A              LD      A,02AH
BBEC DD2111BD          LD      IX,LDWAIT       ;BAUDRATE FUER 3600BD
BBF0 CD38BC            CALL    LDBLOC          ;LADE BLOCKPARAMETER
BBF3 C0                RET     NZ              ;RET BEI FEHLER
BBF4 1144BD            LD      DE,LDNAME       ;VERGLEICHE NAMEN
BBF7 212BBD            LD      HL,SVNAME       ;
BBFA 0610              LD      B,010H
BBFC 7E       CPCON:   LD      A,(HL)
BBFD FE2A              CP      02AH            ;FALLS * SKIP REST
BBFF 280A              JR      Z,SKNAM
BC01 FE3F              CP      03FH            ;FALLS ? SKIP CHAR
BC03 2805              JR      Z,SKCHA
BC05 1A                LD      A,(DE)
BC06 BE                CP      (HL)
BC07 3E42              LD      A,042H
BC09 C0                RET     NZ              ;FALSCH? DANN RET
BC0A 23       SKCHA:   INC     HL
BC0B 13       SKNAM:   INC     DE
BC0C 3E04              LD      A,004H          ;AB B=4 MIT
BC0E B8                CP      B               ;SV-TYP VERGLEICHEN
BC0F 2003              JR      NZ,SKCON
BC11 2138BD            LD      HL,SVTYP
BC14 10E6     SKCON:   DJNZ    CPCON
BC16 2A55BD            LD      HL,(LDSTA)      ;PARAMETER FUER DATEN-
BC19 ED4B57BD          LD      BC,(LDLEN)      ;BLOCK SETZEN
BC1D 3A54BD            LD      A,(LDFLAG)
BC20 17                RLA
BC21 3004              JR      NC,IXALT
BC23 DD21D0BC          LD      IX,LDW24        ;BAUDRATE 7200BD
BC27 3A2ABD   IXALT    LD      A,(ZFLAG)       ;
BC2A CB57              BIT     2,A             ;NEUE STARTADRESSE
BC2C 2803              JR      Z,SKADR
BC2E 2A3CBD            LD      HL,(SVSTA)
BC31 CB4F     SKADR:   BIT     1,A             ;LADEN ODER VERIFY
BC33 3EC5              LD      A,0C5H
BC35 C26DBC            JP      NZ,LDVER
              ;*********************************************
              ;LDBLOC: LAEDT BLOCK
              ;*********************************************
BC38 167E     LDBLOC   LD      D,PORTIN
BC3A D9                EXX
BC3B CD8EBC            CALL    LDSYN           ;SYNCHRONISIERT
BC3E D9                EXX
BC3F 77                LD      (HL),A          ;BYTE NACH (HL)
BC40 EDA1              CPI                     ;INC HL UND DEC BC
BC42 E250BC            JP      PO,LTEST        ;TEST BEI ENDE
BC45 D9       LDNEXT:  EXX
BC46 CDC9BC            CALL    LDBYT           ;NAECHSTES BYTE
BC49 D9                EXX
BC4A 77                LD      (HL),A          ;NACH (HL)
BC4B EDA1              CPI                     ;UND SCHLEIFE BIS
BC4D EA45BC            JP      PE,LDNEXT       ;ENDE
BC50 4A       LTEST:   LD      C,D             ;SETZE PORTADRESSE
BC51 CDC9BC            CALL    LDBYT           ;LADE LO-PRUEFBYTE
BC54 00                NOP                     ;WARTE 12 TAKTE
BC55 00                NOP
BC56 00                NOP
BC57 47       TEST:    LD      B,A
BC58 E5                DEFB    0E5H            ;WARTE 25 TAKTE
BC59 00                DEFB    00H
BC5A E1                DEFB    0E1H
BC5B CDC9BC            CALL    LDBYT           ;LADE HI-PRUEFBYTE
BC5E 48                LD      C,B             ;GELADENE
```



```
BC5F 47                LD      B,A             ;PRUEFSUMME NACH BC
BC60 D9                EXX
BC61 E5                PUSH    HL
BC62 D9                EXX                     ;BESTIMMTE
BC63 E1                POP     HL              ;PRUEFSUMME NACH HL
              ;*********************************************
              ;LDTES: PRUEFSUMMENTEST
              ;*********************************************
BC64 AF                XOR     A
BC65 ED42              SBC     HL,BC           ;VERGLEICHE HL MIT BC
BC67 3E23              LD      A,023H          ;LADE AKKU MIT '#'
BC69 C8                RET     Z               ;RET FALLS OK
BC6A 3E46              LD      A,046H          ;SONST 'F'
BC6C C9                RET                     ;UND RETURN
              ;*********************************************
              ;LDVER : VERIFY
              ;*********************************************
BC6D 167E     LDVER:   LD      D,PORTIN        ;PORTADRESSE
BC6F D9                EXX
BC70 CD8EBC            CALL    LDSYN           ;SYNCHRONISIERT
BC73 D9                EXX
BC74 EDA1              CPI                     ;VERGLEICH
BC76 2013              JR      NZ,ERRVE        ;ERROR FALLS <>
BC78 E285BC            JP      PO,VTEST        ;SPRUNG FALLS ENDE
BC7B D9       VERNXT:  EXX
BC7C CDC9BC            CALL    LDBYT           ;NAECHST. BYTE
BC7F D9                EXX
BC80 EDA1              CPI                     ;VERGLEICH
BC82 EA7BBC            JP      PE,VERNXT       ;BEI PE WEITER
BC85 4A       VTEST:   LD      C,D             ;SONST LADE PRUEF-
BC86 CDC9BC            CALL    LDBYT           ;SUMME
BC89 18CC              JR      TEST
BC8B 3E3E     ERRVE:   LD      A,03EH          ;FEHLER BEI VERIFY
BC8D C9                RET
              ;*********************************************
              ;LDSYNC: SYNCHRONISATION
              ;*********************************************
BC8E 1E00     LDSYN:   LD      E,000H          ;INIT LADEBYTE
BC90 0E7E              LD      C,PORTIN        ;PORTADRESSE
BC92 F3                DI                      ;INTERRUPTS AUS
BC93 57                LD      D,A             ;C=ABSCHLUSSMARKE
              ;WARTE AUF EIN SYCHA
BC94 00       SYNEU1:  DEFB    00H             ;WARTE 4 TAKTE
BC95 A6       LDNXB:   DEFB    0A6H            ;WARTE  7 TAKTE
BC96 C399BC   SYNEU2:  JP      SYWEI           ;WARTE  10 TAKTE
BC99 CBC3     SYWEI:   SET     0,E             ;LADE NUR EIN BIT
BC9B CDCBBC            CALL    LDBIT
BC9E FE16              CP      016H            ;SYCHA ERKANNT?
BCA0 5F                LD      E,A             ;
BCA1 20F2              JR      NZ,LDNXB        ;NEIN ? DANN NEXT
              ;TEST AUF WEITERE SYCHA
BCA3 A7                DEFB    0A7H            ;WARTE 9 TAKTE
BCA4 D8                DEFB    0D8H            ;
BCA5 060A              LD      B,00AH          ;ZAEHLER AUF 10
BCA7 A6       SYCON:   DEFB    0A6H            ;WARTE 7 TAKTE
BCA8 CDC9BC            CALL    LDBYT           ;LADE NAECHSTES BYTE
BCAB FE16              CP      016H            ;VERGL. MIT SYCHA
BCAD 20E5              JR      NZ,SYNEU1       ;UNGLEICH ? SYNC NEU
BCAF A6                DEFB    0A6H            ;WARTE 7TAKTE
BCB0 10F5              DJNZ    SYCON           ;WEITER BIS 10*SYCHA
              ;UEBERLESEN FOLGENDER SYCHA
BCB2 1804              JR      SYEIN
BCB4 A6       SYIGN:   DEFB    0A6H            ;WARTE 22 TAKTE
BCB5 D8                DEFB    0D8H
BCB6 D8                DEFB    0D8H
BCB7 D8                DEFB    0D8H
BCB8 CDC9BC   SYEIN:   CALL    LDBYT           ;LADE NAECHSTES BYTE
BCBB FE16              CP      016H            ;VERGL.MIT SYCHA
BCBD 28F5              JR      Z,SYIGN         ;JA, DANN WEITER LADEN
              ;TEST AUF ABSCHLUSSMARKE
BCBF BA                CP      D               ;ABSCHLUSS RICHTIG?
BCC0 20D4              JR      NZ,SYNEU2       ;NEIN SYNCHR. NEU
BCC2 210000            LD      HL,00000H       ;INIT PRUEFSUME
BCC5 A6                DEFB    0A6H            ;WARTE 23 TAKTE
BCC6 A6                DEFB    0A6H
BCC7 A7                DEFB    0A7H
BCC8 D8                DEFB    0D8H
              ;*********************************************
              ;LDBYT: LAEDT EIN BYTE/BIT
              ;*********************************************
BCC9 1E80     LDBYT:   LD      E,080H          ;SETZE FLAG FUER 8 BITS
BCCB DDE9     LDBIT:   JP      (IX)            ;WARTE
BCCD 00       LDW36:   NOP                     ;VARIABLE
BCCE 00       LDW32:   NOP                     ;EINSRUNGS-
BCCF 00       LDW28:   NOP                     ;MOEGLICHKEITEN
BCD0 00       LDW24:   NOP                     ;FUER
BCD1 00       LDW20:   NOP                     ;IX
BCD2 00       LDW16:   NOP
BCD3 00       LDW12:   NOP
BCD4 00       LDW08:   NOP
BCD5 00       LDW04:   NOP
BCD6 ED78     LDSAM:   IN      A,(C)           ;SAMPLE-WERT
BCD8 FAFEBC            JP      M,HIGH
BCDB ED78     LOW:     IN      A,(C)           ;WARTE AUF STEIGENDE
BCDD F2DBBC            JP      P,LOW           ;FLANKE
BCE0 AF                XOR     A               ;LOESCHE CARRY
BCE1 08                EX      AF,AF'          ;NACH AF' UND HOLE
BCE2 00                NOP                     ;ALTEN CARRY
BCE3 3001              JR      NC,NULL1        ;C=0 DANN 0
BCE5 23                INC     HL              ;SONST INC PRUEFSUMME
BCE6 7B       NULL1:   LD      A,E             ;ROTIERE CARRY IN E
BCE7 1F                RRA
BCE8 D8                RET     C               ;RET BEI ENDFLAG
BCE9 5F                LD      E,A             ;SPEICHER BITS
BCEA 3AF13F            LD      A,(BILDP2)
BCED EE01              XOR     001H            ;STERNCHEN BLINKT
BCEF 32F13F            LD      (BILDP2),A      ;
```

```
BCF2 3A4038   BRCHC:   LD      A,(ZTAST)       ;ZEICHEN VON TAST.
BCF5 FE04              CP      004H            ;BREAK?
BCF7 00                NOP
BCF8 C2CBBC            JP      NZ,LDBIT        ;LADE NAECHSTES BIT
BCFB C300B9            JP      START
BCFE ED78     HIGH:    IN      A,(C)           ;WARTE AUF FALLENDE
BD00 FAFEBC            JP      M,HIGH          ;FLANKE
BD03 37                SCF                     ;SETZE CARRY
BD04 08                EX      AF,AF'          ;NACH AF' UND HOLE ALTES
BD05 3F                CCF                     ;CARRY UND INVERTIERE
BD06 3001              JR      NC,NULL2        ;C=0 ERGIBT BIT=0
BD08 23                INC     HL              ;INC PRUEFSUMME
BD09 7B       NULL2:   LD      A,E             ;ROTIERE CARRY IN E
BD0A 1F                RRA
BD0B D8                RET     C               ;RETURN BEI ENDFLAG
BD0C 5F                LD      E,A
BD0D E5                DEFB    0E5H            ;WARTE 21 TAKTE
BD0E E1                DEFB    0E1H
BD0F 18E1              JR      BRCHC
              ;*********************************************
              ;LDWAIT: WARTE FUER 3600 BAUD
              ;*********************************************
BD11 3E08     LDWAIT:  LD      A,LCON36                ;WARTE 172 TAKTE
BD13 3D       LDWCON:  DEC     A
BD14 20FD              JR      NZ,LDWCON
BD16 C3D6BC            JP      LDSAM
              ;*********************************************
              ;EXIT NACH DOS, DOS-BASIC ODER ROM-BASIC
              ;*********************************************
BD19 3A1740   BWARM:   LD      A,(ECHEC)
BD1C FE03              CP      03
BD1E CACC06            JP      Z,ROMBAS        ;INS ROM-BASIC SPRINGEN
BD21 FE43              CP      43H
BD23 CA2D40            JP      Z,DOS           ;INS DOS SPRINGEN
BD26 C3CC06            JP      DOSBAS          ;INS DOS-BASIC SPRINGEN
              ;*********************************************
              ;DER PUFFERBEREICH
              ;*********************************************
0001          ERROR:   DEFS    1
0001          ZFLAG:   DEFS    1
000C          SVNAME:  DEFS    12
0001          SVPUN:   DEFS    1
0003          SVTYP:   DEFS    3
0001          SVFLAG:  DEFS    1
0002          SVSTA:   DEFS    2
0002          SVLEN:   DEFS    2
0004          SVFREI:  DEFS    4
000C          LDNAME:  DEFS    12
0001          LDPUN:   DEFS    1
0003          LDTYP:   DEFS    3
0001          LDFLAG:  DEFS    1
0002          LDSTA:   DEFS    2
BD56          LDSTH    EQU     LDSTA+1
0002          LDLEN:   DEFS    2
0004          LDFREI:  DEFS    4
BD5C          LDEND    EQU     LDFREI+3
              ;
BD5D 0D       TEXT1:   DEFB    00DH
BD5E 53                DEFM    'SUPERTAPE FUER TRS-80, MODELL 1'
BD7D 0D                DEFB    00DH
BD7E 0D                DEFB    00DH
BD7F 28                DEFM    '(S)AVE (Q)UICK  (L)OAD  (V)ERIFY (E)XIT'
BDA6 0D                DEFB    00DH
BDA7 0D                DEFB    00DH
BDA8 43                DEFM    'COMMAND:'
BDB0 0E                DEFB    00EH
BDB1 23                DEFB    '#'
BDB2 0D       TEXT2:   DEFB    00DH
BDB3 53                DEFM    'SAVE'
BDB7 0D                DEFB    00DH
BDB8 4E                DEFM    'NAME........TYP'
BDC7 0D                DEFB    00DH
BDC8 23                DEFB    '#'
BDC9 0D       TEXT3:   DEFB    00DH
BDCA 53                DEFM    'START ENDE'
BDD4 0D                DEFB    00DH
BDD5 23                DEFB    '#'
BDD6 0D       TEXT4:   DEFB    00DH
BDD7 54                DEFM    'TAPE READY ? #'
BDE5 0D                DEFB    00DH
BDE6 56       TEXT5:   DEFM    'VERIFY'
BDEC 0D                DEFB    00DH
BDED 4E                DEFM    'NAME........TYP'
BDFC 0D                DEFB    00DH
BDFD 23                DEFB    '#'
BDFE 0D       TEXT6:   DEFB    00DH
BDFF 4C                DEFM    'LOAD'
BE03 0D                DEFB    00DH
BE04 4E                DEFM    'NAME........TYP'
BE13 0D                DEFB    00DH
BE14 23                DEFB    '#'
BE15 0D       TEXT7:   DEFB    00DH
BE16 4E                DEFM    'NEUE ADRESSE ? #'
BE26 0D       TEXT8:   DEFB    00DH
BE27 41                DEFM    'ADRESSE'
BE2E 0D                DEFB    00DH
BE2F 23                DEFB    '#'
BE30 0D                DEFB    00DH
BE31 0D       TEXT9:   DEFB    00DH
BE32 42                DEFM    'BASIC-START'
BE3D 0D                DEFB    00DH
BE3E 23                DEFB    '#'
BE3F 0D       TEXT10:  DEFB    00DH
BE40 4B                DEFM    'KEIN BASIC'
BE4A 0D                DEFB    00DH
BE4B 23                DEFB    '#'
B900                   END     START
```



»Sofort aufhören! Der Fehler liegt hier!«

Hardware-Tests

Hinweis für Leute, die an der seriellen Schnittstelle von RB-Elektronik interessiert sind.
Diese Schnittstelle kostet bei RB-Elektronik 249,-DM. Die gleiche Schnittstelle habe ich bei der Firma Dr, Auermann in Strassenhaus für 199,-DM gekauft.

Ulrich Böckling

Hardware-Tests -- Hardware-Tests

Privatanzeige

Verkaufe GENIE 1 64K mit SPEEDUP in speziellem Gehäuse mit externer Tastatur.
1 SHUGART DS/DD 40-Track Laufwerk, RB-Hochauflösende Grafik, RB-V24-Schnittstelle, Akustikkoppler und grüner Monitor.
Preis VHS
Joachim HERL, Petersenstraße 14, 5000 Köln 91, Tel. 0221/843160

BÖRSE

Kurz noch eine Vorbemerkung der Redaktion zu der Spalte Börse (insbesondere möchte ich noch auf den Fragekasten eingehen).

Die Textabschnitte für diese Seiten stelle ich aus Euren Briefen zusammen. Dazu habe ich zur besseren Ordnung und Übersicht zwei Themen erwählt. Und zwar einmal die Spalte WER HAT WAS? - WER SUCHT WAS? und als zweites FRAGEN, FRAGEN, FRAGEKASTEN. Ich möchte Euch nun bitten, Eure Infobeiträge zu diesen Themen so zu gestalten, wie Ihr im Moment aus den folgenden Seiten ersehen könnt. Zu jedem Thema getrennt und unterschrieben.

Bezüglich des Fragekasten hätte ich noch an alle CLUB 80-er eine Bitte. Bei eventueller Beantwortung einer Frage kann dies über die immer beiliegende aktuelle Adressenliste erfolgen, dabei wäre es nett wenn Ihr die Lösung auch der INFO zukommen lassen würdet. Manch eine Beantwortung wäre sicher einen Artikel in der INFO wert.

Vielleicht gibt es dann auf diesem Wege mehere Lösungsmöglichkeiten für ein Problem und man kann sich die auswählen, die einem am günstigsten erscheint. Gleichzeitig wird durch den Infoartikel das Info zum Nachschlagewerk für diejenigen, die sich erst später mit der Thematik befassen und dann die gleichen oder ähnlichen Fragen haben.

Vielen Dank im voraus für Eure Bemühungen.

Die Redaktion

Wer hat was ? ? ?
Wer sucht was ? ? ?

Wer hat ein günstiges Laufwerk zu verkaufen (nach Möglichkeit 40 Spuren/ ss/ dd) ?

Ferner suche ich das relativ neue Buch von L. Röckrath "Programmieren in Maschinensprache". Wer kann es mir mal für einige Tage leihen ?

Günther WAGNER

Derjenige, der am Clubtreffen die Unterlagen (Schaltpläne) für den GENIE wollte, sollte sich bitte nochmals melden.
Manfred HELD

Ich suche schon seit geraumer Zeit ein leeres TRS80 Modell 3-Gehäuse oder Genie 3-Gehäuse. Wo kann ich dies günstig beziehen? Neue Terminalgehäuse sind nicht unter 500,-DM zu erhalten.
Peter Speiß

Ich kann Ringkerntrafos (Durchmesser: 9 cm, Höhe: 3,5 cm) mit den Anschlußwerten 2 x 15 Volt / 2 x 3,3 A zu einem Preis von 30 DM beschaffen. Wer sich in der Bauteilebeschaffung auskennt weiß, daß das ein sensationeller Preis ist. Die Bestellungen bitte direkt an mich schicken.
Hartmut OBERMANN

Fragen, Fragen, Fragekasten

Ich habe nun endlich ein Superscripsit das unter NEWDOS läuft - allerdings mit einem (für mich erheblichen) Schönheitsfehler. Es ist die Version für die amerikanische Tastatur und ich habe die deutsche. Etliche Zeichen sind vertauscht und die Umlaute sind überhaupt nicht erreichbar. Wer weiß Abhilfe (z.B. wo die Tastaur angesprochen wird etc.) ?

Ich habe intern das Laufwerk 0 und extern 2 Laufwerke (Nr. 2 und 3) angeschlossen. Nun habe ich Laufwerk 3 einmal probeweise intern angeschlossen als Laufwerk 1. Dabei ergab sich folgendes: Das Laufwerk 1 lief einwandfrei, formatiert und kopiert einwandfrei - aber: Die alten Disketten (bzw. die Disketten des baugleichen Laufwerk 2) lassen sich nicht im Laufwerk 1 lesen und umgekehrt die vom Laufwerk 1 nicht mehr im Laufwerk 2. Ein Austausch der beiden Laufwerke brachte das gleiche Ergebnis. Das heißt, das ich zwei 80-Spur-Laufwerke hätte, die untereinander verschieden formatieren (nicht aber, wenn diese als Laufwerk 2 und 3 extern angeschlossen werden). An was liegt das? Wer weis Rat? Wer weis, wie ich ev. im Computer z.B. den Controller so einstellen kann, daß das externe Laufwerk 2 als Laufwerk 1 angesprochen wird?

Günther WAGNER

Wer kennt das Innenleben des EG 64 MBA von TCS? Da sich in dem kleinen Kästchen nur 6 IC's und zwei Kondensatoren befinden, finde ich den Preis von ca. 190.-DM etwas happig. Ein Nachbau würde sich also lohnen. Von den IC's (normale TTL-Bausteine) ist die Beschriftung abgeschliffen!

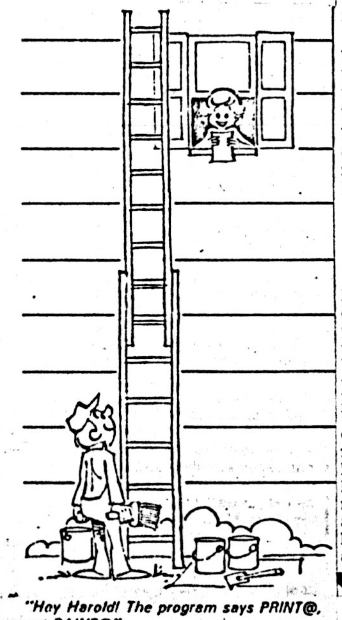
Seit längerer Zeit versuche ich mich mit dem Adventure "Microworld". Es handelt sich dabei um eine Expedition durch das Innenleben eines Computers. Wer kennt die Auflösung oder zumindest einen Teil davon.
Peter Spieß

Aus aktuellem Anlaß sammle ich seit kurzem Artikel über ein sehr heikles Thema. Abmahnung!!!

Seltsamerweise häufen sich die Beiträge in den Computerzeitschriften zu diesem Thema in letzter Zeit. Das läßt darauf schließen, daß sich diese, in Geschäftskreisen durchaus übliche Art eine "Unstimmigkeit" aus der Welt zu schaffen, langsam auch im Hobby-Computerbereich breit macht. Was es damit auf sich hat, wie man sich dagegen schützt bzw. sich dagegen wehrt, könnt ihr in den folgenden Artikeln erfahren.

Ich kann nur jedem wünschen, daß er nie von solchen Praktiken, manchmal windiger und fingier Anwälte verschont bleibt.

Hartmut Obermann



"Hey Harold! The program says PRINT@, not PAINT@"

Die neue Abmahnmasche: Vorsicht bei Programmangeboten

Die neueste Abmahn-Masche, mit der unterbeschäftigte Rechtsanwälte hart am Rande der Legalität zu Geld zu kommen suchen, trifft die Programmierer

So bekam kürzlich ein Leser, der eine selbstgeschriebene Grafik-Routine für 30 Mark in einer Kleinanzeige angeboten hatte, von einem Rechtsanwalt eine Abmahnung samt Gebührenforderung über 501,60 Mark (willkürlich vom Anwalt festgesetzter »Streitwert«: 20000 Mark). Begründung: In der Anzeige fehle der Hinweis, daß es sich um einen gewerblichen Anbieter handle — das verstoße aber gegen das Gesetz gegen unlauteren Wettbewerb.

Das wäre in Ordnung, wenn es sich bei dem Anbieter um eine Firma handeln würde — oder wenn der Softwareverkauf gewerblich betrieben würde. Nun gibt es aber viele Computerbenutzer, die zwar bereit (und vielleicht sogar interessiert sind), das eine oder andere selbstgeschriebene Programm an Interessenten abzugeben — die aber daraus keineswegs ein Geschäft oder gar Gewerbe machen wollen. Um Ärger mit gewerblich tätigen Firmen, Rechtsanwälten und vor allem dem Finanzamt zu vermeiden, sollten Sie entweder nur tauschen (Tausch zwischen Privatleuten im Rahmen ihres Hobbys ist keine gewerbliche Tätigkeit) oder darauf achten, daß Sie lediglich einen Kostenersatz berechnen. Es ist zweckmässig, den Betrag zu spezifizieren — zum Beispiel 1,30 Mark Porto, 10 Fotokopien á 0,50 Mark, eine Diskette á 4,85 Mark und so weiter.

Wenn Sie einen — und sei er auch nur bescheiden — Gewinn erzielen wollen, müssen Sie auf schriftlichen Unterlagen in Inseraten und so weiter durch eine geeignete Angabe wie »Firma«, »Programmierbüro«, »Softwarevertrieb« oder ähnliches erkennen lassen, daß Sie sich gewerblich betätigen. Sie müssen außerdem das Gewerbe bei der Gemeinde beziehungsweise Stadt anmelden und ein Minimum an Buchführung machen, damit Sie dem Finanzamt jederzeit Umsätze, Kosten und Gewinn nachweisen können. In den meisten Fällen werden Umsatz und Ertrag so gering sein, daß ohnehin keine ernstzunehmende Menge Steuern zu bezahlen ist.

Sollten Sie als Privatmann eine Abmahnung der oben erwähnten Art bekommen, dann schreiben Sie umgehend zurück, daß Sie ihren Computer nur privat benutzen, die Programme für private Zwecke geschrieben haben und durch das Anbieten ihrer selbstgeschriebenen Progamme Kontakt zu anderen Computerbenutzern zum Zweck des Erfahrungs- und Programmaustausches suchen. Ihre selbstgeschriebenen Programme gäben Sie entweder im Tausch oder gegen Ersatz der durch Erstellen und Versenden der Kopie entstehenden Kosten ab. Falls das zutrifft, brauchen Sie auch keine Unterwerfungserklärung abzugeben und keine Gebühren zu zahlen.

(py)

Abmahnschwindler nie gefaßt

Mit einem Abmahnschwindel besonderer Art tat sich im vergangenen Jahr eine »R + S Computerorganisation« in Berlin hervor: Sie trat als angeblicher Wettbewerber auf, verlangte von zahlreichen Anbietern von Raubkopien die Abgabe einer Unterlassungserklärung sowie die Bezahlung einer Gebührenrechnung in Höhe von mehreren hundert Mark. Der Rechnungsbetrag sollte in bar zusammen mit der Unterlassungserklärung an eine Postfachadresse in Berlin geschickt werden. Die Staatsanwaltschaft schaltete sich sehr schnell ein und stellte fest, daß das angegebene Postfach in der vorgegaukelten Form nicht existierte. Es handelte sich dabei, wie die Justizpressestelle jetzt auf Anfrage mitteilte, um die Nummer einer Postlagerkarte bei einem Berliner Postamt, die tatsächlich ausgegeben worden war »ohne daß die Personalien des Empfängers notiert worden wären oder hätten notiert werden müssen«. Bei Beobachtungen in dem Postamt stellte die Kriminalpolizei im vergangenen Jahr zwar einen 15jährigen Jungen, der mit der Postlagerkarte und einer Vollmacht der Schwindelfirma die Post abholen wollte. Als die Polizisten ihn nach dem Auftraggeber fragten, deutete er auf einen etwa hundert Meter vom Postamt entfernt stehenden Mann, der daraufhin zusammen mit einem anderen, mit einem Auto die Flucht ergriff. Da der Junge nach Feststellungen der Polizei als Mittäter ausscheidet und die beiden Flüchtigen nicht identifiziert werden konnten, wurde das Verfahren gegen R + S wohl oder übel eingestellt.

(py)

Das Geschäft mit den Kleinanzeigen

Die Kleinanzeigen in Zeitschriften sind eine beliebte Lektüre nicht nur für Computer-Interessierte, sondern auch für Abmahnvereine.

Kleinanzeigen versprechen ein gutes Geschäft. Doch nicht nur dem Anbieter. Immer häufiger treten jetzt Leute auf, die sich weniger für die angebotenen Waren interessieren, als vielmehr dafür, wie die Ware angeboten wird. Sehr häufig lassen sich da Verstöße gegen das UWG, das „Gesetz gegen den unlauteren Wettbewerb" nachweisen.

Jede Anzeige muß bestimmten Maßstäben gerecht werden. Die Generalklausel des UWG (§ 1) lautet: „Wer im geschäftlichen Verkehr zu Zwecken des Wettbewerbs Handlungen vornimmt, die gegen die guten Sitten verstoßen, kann auf Unterlassung und Schadensersatz in Anspruch genommen werden." Gegen die guten Sitten wird dann verstoßen, wenn angenommen wird, daß der Tatbestand des Kundenfangs, der Behinderung, der Ausbeutung und/oder des Rechtsbruchs vorliegt. Konkrete Fälle sind: Ein Akustikkoppler ohne Postzulassung wird ohne Hinweis auf die fehlende Postzulassung angeboten, oder es wird kein Firmenstatus angegeben, sondern nur eine Telefonnummer des Anbieters.

Abmahnvereine

Im strengen Sinne können in diesen Fällen Unterlassungsansprüche erhoben werden. Dazu sind in erster Linie die Mitbewerber und auch die „Verbände zur Förderung gewerblicher Interessen" nach UWG § 13 berechtigt. Bei letzteren, gemeinhin als Abmahnvereine bekannt, besteht jedoch teilweise Anlaß, ihr Vorgehen kritisch unter die Lupe zu nehmen. Mit Hilfe des UWG läßt sich nämlich nicht nur gut abmahnen, sondern auch gut absahnen.

Auf eine Kleinanzeige in einer Zeitschrift bekam ein Inserent von fünf verschiedenen Rechtsanwälten beziehungsweise Organisationen Abmahnschreiben mit Geldforderungen zwischen 500 und 1000 Mark und Klagedrohungen, weil in diesem Fall die Mehrwertsteuer getrennt ausgewiesen worden war.

Eine Abmahnung enthält in der Regel folgendes: Der Abgemahnte wird zuerst auf seinen Verstoß hingewiesen; unter Hinweis auf die Einleitung gerichtlicher Schritte (einstweilige Verfügung oder Klage) wird er aufgefordert, den Verstoß zu unterlassen; unter Fristsetzung wird er dazu aufgefordert, eine „Unterwerfungserklärung" zu unterzeichnen, in der er den Verstoß gegen das UWG zugibt.

Wird dafür ein Rechtsanwalt eingesetzt, so kostet dies alles Gebühren — und damit machen manche dubiose Abmahnvereine das schnelle Geld. Sie durchforsten die Kleinanzeigen der Zeitschriften nach Verstößen gegen das UWG und schicken Abmahnschreiben los. Für den Inserenten, der sich oft keines Vergehens bewußt ist, bringt das vor allem viele Kosten mit sich.

Die Gebühr für eine Abmahnung durch einen Rechtsanwalt berechnet sich nach dem Streitwert. Bei mittleren Wettbewerbsverstößen wird in der Regel vom Abmahner ein Streitwert von 10 000 Mark (bei Geschäftsleuten oft auch 30 000 Mark) angenommen. Der Rechtsanwalt macht die halbe BRAGO-Gebühr, Portopauschale von 40 Mark und Umsatzsteuer geltend und kommt so auf rund 300 Mark. Nimmt sich auch der Abgemahnte einen Anwalt, so kann dessen Tätigkeit nach denselben Grundsätzen bewertet werden und weitere 300 Mark kommen dazu.

Reagiert der Betroffene nicht auf die Abmahnung, so kann es noch teurer werden. Macht der Abmahner die angedrohten Schritte wahr, dann beantragt er im Regelfall eine einstweilige Verfügung des zuständigen Landgerichts. Diese erhält der Abgemahnte als Zustellungsurkunde vom Gerichtsvollzieher. Mit der einstweiligen Verfügung wird der „Wettbewerbsstörer" aufgefordert den Verstoß gegen das UWG zu unterlassen. Bei Zuwiderhandlung wird eine Geldstrafe bis zu 500 000 Mark oder eine Haft bis zu zwei Jahren angedroht.

Der Betroffene muß auch diesmal zahlen: Die vom Anwalt geltend gemachten Gebühren haben sich verdoppelt. Dazu kommen die Kosten des Landgerichts für die einstweilige Verfügung von rund 100 Mark.

Teure Gebühren

Doch damit nicht genug: Fordert der Anwalt den Abgemahnten nach Erlaß der einstweiligen Verfügung erneut auf, eine Unterwerfungserklärung abzugeben, werden für dieses „Abschlußschreiben" wiederum Gebühren fällig. Diesmal berechnet nach dem Wert, der der Klage zugrunde gelegt wurde. Im Regelfall wird dann vom dreifachen des ursprüglichen Streitwertes ausgegangen, also bei zuerst 10 000 Mark von nunmehr 30 000 Mark.

Zahlt der Betroffene die Gebühren des Abmahners nicht, so erhält er einen Mahnbescheid, der ihn wieder etwa 100 Mark kosten kann.

Ein Prozeß gegen die einstweilige Verfügung muß vom Landgericht des Abmahners geführt werden. Wegen des Anwaltszwanges kann es sein, daß der bisherige „Vertrauensanwalt" nicht zugelassen wird. Die Kosten des neuen Anwalts und die Gerichtskosten summieren sich.

Gegenmaßnahme

Aber es gibt einen Lichtblick. In seinem Urteil vom 12. April 1984 hat sich der Bundesgerichtshof zu den Abmahnpraktiken geäußert: Das Abmahn-Opfer muß nun nicht mehr die für die Abmahnung erhobenen Gebühren zahlen, wenn die Einschaltung eines Rechtsanwaltes überflüssig war. Dies ist immer dann der Fall, wenn der Abmahnende die für die Abmahnung „maßgeblichen Kriterien, insbesondere Branchenübung und Verkehrsauffassung aus eigener Sachkunde beurteilen kann und der Erwerb der übrigen erforderlichen Sachkenntnis ihm angesichts des Umfanges seiner Abmahntätigkeit zuzumuten ist. Bei einer solchen Ausstattung des Klägers würde sich bei den typischen und durchschnittlich schwierigen Abmahnungen die Einschaltung eines Rechtsanwalts erübrigen und er wäre daher nicht ..." erforderlich.

Bei Eingang der Abmahnung sollte der Abgemahnte also im Zweifel nach Einholung rechtlicher Beratung unverzüglich eine Unterwerfungserklärung abgeben und unter Hinweis auf das Urteil des BGH die Zahlung der geforderten Gebühren verweigern. Damit wäre dem UWG genüge getan und der Gebührenschinderei könnte damit ein Riegel vorgeschoben werden.

Hans Preisker

Abmahnung und Durchsuchung – was tun!

Wer aktiv ist, berührt automatisch Interessen anderer, nur wer nichts tut, kann niemand stören. Wer mit seinem Computer arbeitet, ist aktiv, und je mehr er aktiv ist, desto schneller kommt er mit anderen in Konflikt. Mit anderen in Konflikt zu kommen, heißt aber noch nicht, mit dem Gesetz in Konflikt zu kommen, denn jeder meint, daß das Recht auf seiner Seite sei. Es gibt auch Rechte für den einen und den anderen. Kommt das Recht des einen mit dem des anderen in Konflikt, entscheiden die Gerichte, wessen Recht vorrangig ist.

Viele junge Computerfreunde haben mit Recht und Gesetz nichts zu tun gehabt. Sie sind unerfahren, die Eltern haben sich oft so bewegt, daß auch sie nie mit anderen Streit hatten. So können viele auch nicht aus Erfahrungen der Eltern ein Wissen ableiten. Umso wichtiger ist es, daß hier einmal klar gesagt wird, was eine Abmahnung ist und wie man sich dabei verhält. Und erst recht muß man wissen, was man bei Durchsuchungen tut oder besser nicht tut. Zwischen Abmahnung und Durchsuchung sind wesentliche Unterschiede. Wer abgemahnt wird, wird von einem Privatmann privat verfolgt. Kommt die Polizei zur Durchsuchung, wird man vom Staat verfolgt. Wer abgemahnt oder durchsucht wird, braucht erst einmal selbst nichts zu tun. Nichts ist falscher, als sofort zu reagieren. Liegt die Abmahnung auf dem Tisch, notiert man die Frist und denkt einmal nach.

Abmahnungen

Während dieser Frist denkt man nach, ob die Abmahnung begründet ist oder begründet wäre, beides kann der Fall sein. Wer wirklich Urheberrechte verletzt und es nicht mehr tun will, gibt eine Unterlassungserklärung ab. Aber immer schön in die Unterlassungserklärung hineinschreiben: "Ohne Anerkenntnis, Veranlassung gegeben zu haben", und immer schön den Satz rausstreichen, daß man die Kosten tragen will. Wenn nichts bewiesen werden kann, kann einem insbesondere nicht bewiesen werden, daß man die Abmahnung veranlaßt hat.

Wer meint, daß er tun darf, was abgemahnt wird, braucht **nichts zu tun.** Er wartet die einstweilige Verfügung ab. Nichts tun ist dann am besten, um den Gegner nicht erst auf intelligente Gedanken zu bringen. Wichtig ist (was die meisten Anwälte falsch machen): **Keinen Widerspruch** einlegen! Der Könner beantragt **Fristsetzung zur Erhebung der Hauptsacheklage beim Gericht. Einen Anwalt braucht man solange noch nicht.** Dann muß der Abmahner Hauptsacheklage erheben und beweisen. Beim Widerspruch verbleibt der Rechtsstreit im Bereich des juristischen Wischi-Waschi der Wahrscheinlichkeitstheorien. Das ist schlecht, besonders wenn man eine seriöse Firma als Gegner hat. Gewinnt man den Hauptsacheprozeß, wird die einstweilige Verfügung auch aufgehoben und man hat einen Schadensersatzanspruch für die Zeit ihrer Geltung.

Durchsuchungen

Gegen Durchsuchungen kann man nichts machen, sondern muß diese über sich ergehen lassen. Grundsätzlich muß man wissen: die Polizei ist nur ausführendes Organ. Jeder Polizist ist, leger gesagt, nur Hampelmann eines Staatsanwaltes. Die Polizei kann nichts entscheiden, sie ist für die Durchsuchung nicht verantwortlich. Sie sammelt nur Material. Deswegen gilt: nichts sagen, denn unüberlegte Worte und mißverständliche Sätze merken sich die Polizisten, schreiben sie nieder und verwenden sie gegen den Durchsuchten. Daß die Polizei verpflichtet wäre, sich auch zugunsten des Durchsuchten etwas zu notieren, ist bloße Rechtstheorie (§ 161 StPO).

Beschwerden gegen die Durchsuchungen sind unzulässig, weil die Durchsuchung immer abgeschlossen ist, wenn die Beschwerde beim Gericht eingegangen ist. Also laßt die Polizei suchen. Auf jeden Fall sollte man sich aber eine Abschrift des gerichtlichen Durchsuchungsbeschlusses aushändigen lassen. Wenn die Polizei was gefunden hat, erklärt sie, ob sie es beschlagnahmen will. Was sie beschlagnahmen will, und hierauf müßt Ihr bestehen, muß einzeln im Protokoll aufgeschrieben werden. Also nicht einfach "10 Disketten", sondern: 1 Diskette, 3,5", BASF, beschriftet mit...usw. Haben die Beamten keine Zeit, dann besteht darauf, daß alles in einen Sack, notfalls eine Plastiktüte verpackt wird, die zu versiegeln ist. Das Verzeichnis kann dann an einem der nächsten Tage, wenn die Leute mehr Zeit haben, in Eurer Anwesenheit erstellt werden. Immer verbal sehr zurückhaltend sein! Man darf keinem Polizisten Gelegenheit geben, Ordnungsmacht zu spielen. Der Polizist muß ganz in der Rolle des Hilfsbeamten der Staatsanwaltschaft bleiben.

Gegen die Beschlagnahme kann man dann eine Beschwerde schreiben. Meist sind die Durchsuchungsbeschlüsse der Gerichte aus der hohlen Hand geschrieben, ein Verdacht aus den Fingern gesaugt. Aber selbst wenn was dran ist, reicht es meist später im Hauptverfahren nicht aus. Dann gilt der Beweis, vorerst reicht aber der Verdacht. Deswegen darf man keinen Fehler machen und selbst etwas zur Sache sagen. Denn wer jede Woche Krimis im Fernsehen sieht, merkt, daß die Mörder sich durch eigenes Gerede überführen. Deswegen gilt der Grundsatz: Ruhe ist die erste Bürgerpflicht; in der Aufregung sagt man mit Sicherheit das Falsche. Die meisten Leute werden verurteilt, weil sie etwas sagen (sich zur Sache einlassen), was ihnen widerlegt werden kann. Dann braucht ihnen nichts mehr bewiesen zu werden, sondern das Gericht beweist nur, daß die eigene Aussage widerlegt ist. Und dann hat man sich selbst seine Chancen kaputt gemacht.

Was ist das Ergebnis einer Durchsuchung? Die Polizei beschlagnahmt einen Computer. So einen darf man haben. Sagt man also nicht, daß man damit auch kopiert hat (auch mal ganz für sich privat kopiert hat), ist nichts bewiesen. Auch wenn Kopiermaterial gefunden wird, es ist nichts bewiesen, weil man seine eigenen Dateien so oft kopieren kann wie man will. Man darf auch eigene Programme schreiben und diese kopieren. Man kann sich also nur selbst um Kopf und Kragen reden.

Kleinanzeigen zum Superbilligpreis

Und was ist, wenn die Polizei fremde Kopien findet? Wer sich nicht zur Sache einläßt, braucht dann überhaupt keine Antwort zu geben. Wer sich aber anfangs rauszureden versuchte, der muß jetzt solche Antworten entgegnen, die im jetzt peinlich werden. Wer weise von Anfang an geschwiegen hat, erinnert an seine Aussageverweigerung und die gefundenen Raubkopien sind als Beweismittel wertlos. Man kann diese von einem Raubkopierer erhalten haben und wollte diese gerade prüfen. Diese denkbaren Möglichkeiten trägt man aber besser erst im späteren Verfahren vor.

Wie geht das spätere Verfahren weiter? Drei von vier Verfahren werden eingestellt. Also werden, wenn man nichts gesagt hat, nach einiger Zeit die Gegenstände zurückgegeben.

Dies ist das wahrscheinlichste. Wer angeklagt wird, der kann nun, aber erst jetzt, in einem guten Schriftsatz den Verdacht entkräften. Denn erst jetzt legt man die Beweise der Unschuld hin. Wer zu früh seine Karten ausspielt, seine Unschuld zu früh beweisen will, gibt der Staatsanwalt nur Gelegenheit, die Vorwürfe zu untermauern und neue Argumente zu bringen. Anklagen müssen dünn bleiben. Das Wichtigste ist, Nerven zu behalten, weil gerade in der ersten Aufregung die Fehler passieren. Die Zeit für die Verteidigung ist dann, wenn die Anklage geschrieben ist. Vorher füttert man bloß die Polizei mit Wissen.

Weil viele junge Leute auf rücksichtslose Weise und auch durch die heuchlerische Freundlichkeit der Polizei hereingelegt werden, etwas zu sagen, habe ich mit anderen Profis einen COMPUTER PIONEER CLUB gegründet. Wir bilden unsere Mitglieder aus, solchen Angriffen standzuhalten. Wir helfen uns gegenseitig mit Erfahrungen und können gemeinsam manche Kopie machen, ohne uns zu gefährden. Warum wollt Ihr Euch aus Rache dafür hängen lassen, daß die echten Profis nicht schlagbar sind. Also verhaltet Euch richtig und schreit und schreibt, wenn Ihr Euch verunsichert fühlt. Die Polizei ist ganz gewiß nicht der Freund und Helfer. Sie ist es von Gesetzeswegen nicht und könnte es auch gar nicht sein. Wer will schon solche Freunde, die Demonstranten verprügeln und Parksünder anzeigen, hinter Büschen Schnellfahrer stoppen und wenn man sie braucht, nicht zuständig sind. In der Demokratie schließt man sich mit Leuten zusammen, die die gleichen Interessen haben.

Graf Adelmann, 7794 Wald 3

Computerkäufer sind anders!

Daß Computerfans eine Zielgruppe sind, die man auf dem Markt nicht mehr übersehen darf, ist bekannt. Konsequenterweise werden die ersten Marktforschungsunternehmen tätig. Jüngst ist eins der ersten Untersuchungsergebnisse bekannt geworden: Es ist äußerst schmeichelhaft.

Die Computerfans sind mit 71 % überwiegend männlichen Geschlechts. Mit ihrer hervorragenden Ausbildung – 22 % besitzen Abitur und haben ein Studium absolviert – heben sie sich von dem Gesamtdurchschnitt (13 %) deutlich ab.

55 % aller Befragten bezeichneten sich gegenüber allem Neuen aufgeschlossen. Zum Vergleich: Der Durchschnitt der Gesamtbevölkerung liegt bei lediglich 36 %. Computerfans finden Technik zu 52 % faszinierend (Durchschnitt: 27 %). 50 % aller Befragten arbeiten beruflich mit elektronischen Datenverarbeitungs-

Siegmar Wittig --- Heinz Heise GmbH

Ein BASIC-Kurs fuer Nicht-Mathematiker und echte Amateure. Neben Grund- und Aufbaukurs gibt es zahlreiche Beispiele (eine gute Sammlung!).

Nr. 0002: Computerwissen

Michael Scharfenberger --- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von Hard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knobeleien programm. in BASIC

Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

- --- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel --- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80

Chris Palmer --- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment.

Nr. 0008: More Basic Computer Games

David H. Ahl --- Creative Computing Press, USA

84 Spiele fuer den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist fuer Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Busch Rudolf --- Franzis-Verlag

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0011: TRS-80 PROGRAMS

Tom Rugg und Phil Feldman --- Dilithium Press, Beaverton,USA

32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012: Programme und Tricks fuer Genie I und Genie II

Clemens Becher, Franz Seger --- ?

Viele Programme, Tips und Tricks fuer den Genie.

Nr. 0013: BASIC: Alles ueber PEEK und POKE

Heiko Requardt --- Franzis-Verlag

Eine Software-Sammlung in BASIC (mit vielen guten Tips und Tricks fuer den 'Amateur').

Nr. 0014: TRS-80 und Video Genie ROM-Listing fuer Level II

Luidger Roeckrath --- ?

ROM-Listing, RAM-Adressen, I/O-Adressen, Unterprogramme, Basic-Anweisungen und Funktionen, Aufzeichnungsformate auf Cassette, ...

F U N D G R U B E U N D B Ü C H E R

Nach wie vor bauen wir unsere Bücherbibliothek auf. Diese Bücher können übrigens von jedem Mitglied ausgeliehen werden. Das gleiche gilt für die FUNDGRUBE. Das sind 6 Ordner deren Inhalt aus dem 4. Info ersichtlich ist.
Die Bücher und die Fundgrube-Ordner können bei Günther Wagner entliehen werden.

> **„Wenn man Intelligenz als die Fähigkeit definiert, neue Dinge zu lernen und Lösungen für Probleme zu finden, die das erste Mal auftauchen – wer ist dann intelligenter als das Kind?"**

Lebensdauer von Disketten

Disketten, auch Floppy Disks genannt, sind gerade für Mikrocomputer das Massenspeicher-Medium der Wahl, sowohl was die Speicherkapazität betrifft als auch in bezug auf Datensicherheit. Doch wenn man die Arbeit vieler Stunden oder gar Wochen einer solchen Scheibe zur Aufbewahrung anvertraut, dann möchte man es vielleicht einmal ganz genau wissen, wie sicher und für welchen Zeitraum die Daten wieder abrufbar bleiben.

Die eigentliche Information auf einer Magnetplatte wird durch die Ausrichtung winziger Magnete in der Beschichtung gebildet, wobei ein Datenbit etwa vier tausendstel Millimeter lang ist. Bleibt nun die Umgebungstemperatur im Bereich von 0...50 °C und setzt man die Diskette keinen starken Magnetfeldern oder mechanischen Beanspruchungen aus, dann ist diese Information praktisch unbegrenzt haltbar. So liegt zum Beispiel bei der BASF in Ludwigshafen ein Magnetband, dessen Konzertaufnahme aus dem Jahre 1936 bis heute nichts an Qualität verloren hat. Klar, daß man durch Erfahrungen in der Herstellung von Trägern die magnetischen Aufzeichnungen seit damals verbessert hat, und somit von einer unbegrenzten Haltbarkeit der Information auf einer Diskette ausgehen kann.

Auch der Abrieb am Kopf bei häufig gelesenen Floppy-Disks ist wesentlich unkritischer, als man vielleicht annehmen könnte: Einen sauberen Schreib-/Lesekopf vorausgesetzt verträgt jede Spur mehrere Millionen Kopfdurchläufe; also bei 300 Umdrehungen pro Minute eine Zeit von einigen Monaten ununterbrochenen Zugriffs – und das auf jede einzelne Spur!

Daß dann trotzdem manchmal Disketten nicht mehr zu lesen sind und damit das Ergebnis vieler arbeitsreicher Stunden dahin ist, hat meist ganz andere Ursachen. Einer der größten Datenvernichter in diesem Zusammenhang ist wohl der Kaffee. Aber auch andere Getränke oder auch Fingerabdrücke können, auf die magnetisierbare Schicht der Diskette aufgebracht, die Informationen wirksam abdecken und somit jedem Lesezugriff entziehen. Im Fall der Fingerabdrücke ist es meist möglich, durch häufige Leseversuche die Verunreinigung zu beseitigen und so die Datei zu retten. Bei anderen Verschmutzungen hilft oft gar nichts mehr: Die Lösungsmittel, die wirklich helfen können, lösen auch gleich die Trägerschicht. Sollte es trotzdem gelingen, eine so gereinigte Floppy noch einmal zu lesen, muß diese anschließend gleich aus dem Verkehr gezogen werden: Durch die Reinigung verschwindet nämlich auch das Gleitmittel, das den Kopf gegen großen Abrieb schützt. Und ein neuer Schreib-/Lesekopf ist bestimmt teurer als einige neue Disketten. Kr.

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PlZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Baldes	Hans	Johann-Strauss-Str. 6	8025	Unterhaching	089 /6115179
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Bozek	Hans Juergen	Gut Fachenfelde	2093	Stelle	
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Hallup	Matthias	Junggesellenstr. 15	4600	Dortmund	
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruber 35	8038	Groebenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
Marx	Andreas	Mecklenburgring 48	6600	Saarbruecken	0681 /812983
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-450)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod/Kemel	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Rohnenstr. 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Sickmann	Bernhard	Fleigenweg 2	4430	Steinfurt 2	02552 /60344
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 6327	7800	Freiburg	0761 /35384
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Wies	Jean-Claude	Harthweg 9	6600	Saarbruecken	0681 /582513
Wucherer	Juergen	Brauneggerstr. 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Wegen Nichtbezahlung von Beiträgen
– trotz mehrmaliger Anmahnung –
wurden vom CLUB ausgeschlossen :

Peter Schmidt
Robert Trost
Dieter Neukam
Donald Wright

Bitte überprüft Eure Adresse und meldet mir Fehler oder Veränderungen.
Die Redaktion

Vorstand

Kontaktadresse für Clubangelegenheiten, Clubbücherei /Fundgrube, Clubkasse

Günther WAGNER
Gartenstraße 4
8201 Neubeuern
Tel.: 08035 /3361
< 18 - 20 Uhr >

Programmbibliotek

Kontaktadresse

Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod /Kemel
Tel.: 06124 /3913

Redaktion

Kontaktadresse

Jens NEUEDER
Panoramastraße 21
7176 Michelbach /Bilz
Tel.: 0791 /42877
tagsüber 0791 /44-450

Adventure-Ecke

Kontaktadresse

Alexander WAGNER
Theresienstr. 21c
8224 Chieming
Tel.: 08664 /1500

Hardware

Kontaktadresse

Walter ZWICKEL
Lengfelden 123
5101 Bergheim (Austria)
Tel.: 0043662 /51130

Redakteure dieser Ausgabe

Herbert Alber * Ulrich Böckling
Eckehard Kuhn * Jens Neueder
Hartmut Obermann * Gerald Schröder
Arnulf Sopp * Günther Wagner
Walter Zwickel
sowie Artikel aus: MC, CHIP, c't und Computer Persönlich

Bankverbindung des CLUB 80

Sparkasse Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postscheckkonto der Sparkasse
Nr. 8077-801

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

Ihr wisst ja, daß sich in unserem Club nun einiges verändert hat und so möchte ich am Schluß als "Redaktuer" noch ein paar Sätze loswerden.

Zunächst einmal zu den Terminen. Unsere Clubzeitung kam ja bisher so ungefähr im zweimonatlichen Turnus. Da ich in den nächsten Jahren auch privat ziemlich eingespannt bin, möchte ich das in dieser Weise beibehalten.

Damit ein reibungsloser Ablauf gewärleistet ist habe ich mich zur Einführung eines Redaktionsschlusses entschlossen. Der Redaktionsschluß für das jeweilige INFO ist der letzte Tag der geraden Monate im Jahr. Für die Endfertigstellung, das Vervielfältigen und den Versandt an Euch rechne ich dann so ca. 14 Tage. Somit bekommt Ihr das INFO Anfang bis Mitte Jan., März, Mai, Juli, Sept. und Nov. ins Haus. Um eine termingerechte Ausgabe unseres INFO zu gewährleisten, bitte ich daher um Einhaltung unseres Redaktionsschlusses.

Desweiteren möchte ich an Alle appellieren für die INFO zu schreiben, wobei die Größe des Artikels nicht entscheidend ist. -Ob groß oder klein jeder hat doch im Umgang mit seinem Computer einen Trick oder Kniff über den sich etwas schreiben ließe. Also spitzet Eure Federn und versucht es einmal mit einem kleinen Infobeitrag. Vielleicht stellt sich dann sogar ein neues Talent heraus. Anhand des Inhaltsverzeichnisses könnt Ihr ersehen, welche Themengruppen wir ansprechen. Einer regen Beteiligung sehe ich mit Begeisterung entgegen.

Bei Gelegenheit schreibt doch bitte ob, bzw. wie Ihr unter Tage telefonisch zu erreichen seit. Es wäre für mich einfacher, wenn ich bei Euch wegen eventuellen Rücksprachen -welche das INFO betreffen- tagsüber anrufen könnte.

Ansonsten hoffe ich, daß das INFO in der Form, in der es gerade vor Euch liegt, gefällt. Für Verbesserungsvorschläge oder neue Anregungen wäre ich Euch dankbar.

Bis zum nächsten INFO

Jens Neueder

CLUB 80

CLUBINFO
DER
TANDY -
GENIE -
UND KOMTEK -
ANWENDER

8. AUSGABE

Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubeuern
Tel.: 08035/3361 (18 - 20 Uhr)

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert 01
Diskussion . . . . .21 - 22

Software

Das Abtipplisting 02
LPRINT alles einfacher . . . . 03 - 04
COBOL eine Einführung . . . . 05 - 11
Tips und Tricks . . . . 12 - 14
BASIC - Dolmetscher gesucht . . . . . 15 - 18
Tongenerator . . . . 19 - 20

Hardware

Shift - Trick . . . . 23 - 26
ECB - Bus System 24
Taktumschaltung für VIDEO-GENIE . . 25 - 26
Super-Tape (nochmals) 27
Tests 28

Seite:

Börse

Wer hat was ???--Wer sucht was ??? . . 36
Fragen, Fragen, Fragekasten 36

Sonstiges

Kurzgeschichten 37

Programmbibliotek

Neue Programme . . . . 29 - 34
Exklusiv 35

Die letzten Seiten

Clubmitgliederadressen 38
Impressum 39
Schluß 40
Neue CLUBSATZUNG . . . . am INFO-Ende

Hallo liebe Club-Freunde,

dieses Mal darf ich mich kurz fassen. Ich bin gerade vom Urlaub zurück gekommen und noch nicht so ganz drin im Geschäft.

Zunächst das wichtigste - das Ergebnis unserer Abstimmung:
****

Abgegeben wurden bis zum 01.07.85 insgesamt 25 Abstimmzettel.

Punkt 1: Annahme der neuen Clubsatzung
23 Ja-Stimmen
2 Nein-Stimmen
Hiermit ist die neue Clubsatzung gültig.

Punkt 2: Nachzahlung von 10 DM für das laufende Jahr
23 Ja-Stimmen
2 Nein-Stimmen
Hiermit muß jedes Mitglied 10 DM nachzahlen.

Punkt 3: Erhöhung des Jahresbeitrages auf 50 DM
24 Ja-Stimmen
1 Nein-Stimme
Hiermit beträgt der Jahresbeitrag 50 DM.

Ich bin froh, daß sich die Mitglieder zu der neuen Clubsatzung so eindeutig bekannt haben. Verständlich ist, daß insbesondere die Beitragserhöhung und die Nachzahlung unsere Schüler und Studenten besonders trifft; daher überrascht es mich, das hier doch nur 1 bzw. 2 Nein-Stimmen abgegeben wurden.

Wir sind also jetzt vom finanziellen her in der Lage, weiterhin gute Clubarbeit zu leisten. Es kann weiterhin dicke Clubinfos geben. Aber dicke Clubinfos gibt es nur bei entsprechender Mitarbeit der Mitglieder - und dieses Mal hat der Jens nicht gerade viel erhalten. Hoffentlich schleicht sich hier jetzt keine Trägheit ein. Greift zur Feder, hackt in die Tasten - der Jens braucht Euere Beiträge um ein gutes Info rausschicken zu können. Und vergeßt den Hartmut nicht mit der Programmbibliothek.

Noch kurz zur Nachzahlung der 10 DM. Von vielen Mitgliedern habe ich diese ja schon beim Clubtreffen erhalten. Viele haben auch auf das 7. Info hin gleich die 10 DM bezahlt. Alle anderen Mitglieder bezahlen bitte die 10 DM so bald wie möglich. Ich würde mich freuen, wenn bis Ende Juli alle Mitglieder die 10 DM bezahlt hätten.

So - das soll's dieses Mal von meiner Seite gewesen sein.
Bis zum nächsten Mal
Euer
Günther Wagner

Das Labyrinth und seine Maus

- das Programmlisting zum Abtippen -

```
1 CLS
2 ONERRORGOTO26
3 FORI=0TO122:SET(I,0):SET(I,45):NEXTI
4 FORI=0TO45:SET(0,I):SET(123,I):NEXTI
5 FORX1=0TO122STEP3:IFPOINT(X1,45)THENX=X1:Y=45:GOSUB11
6 NEXTX1
7 FORY1=0TO45STEP3:FORX1=0TO122STEP3
8 IFPOINT(X1,Y1)THENX=X1:Y=Y1:GOSUB11
9 NEXTX1:NEXTY1
10 GOTO90:'ab 100 startet ihr Maus-Programm!!!
11 '    Ab gesetztem Punkt, Wand ziehen
12 Z=1
13 R=RND(4)
14 Z=Z+1:IFZ=10THENZ=1:RETURN
15 IFR=1THENIFPOINT(X-3,Y)=-1THENGOTO13
16 IFR=2THENIFPOINT(X+3,Y)=-1THENGOTO13
17 IFR=3THENIFPOINT(X,Y-3)=-1THENGOTO13
18 IFR=4THENIFPOINT(X,Y+3)=-1THENGOTO13
19 FORI=1TO3
20 IFR=1THENX=X-1
21 IFR=2THENX=X+1
22 IFR=3THENY=Y-1
23 IFR=4THENY=Y+1
24 SET(X,Y)
25 NEXTI:Q=RND(3):IFQ=1THENGOTO15ELSEGOTO13
26 RESUME13
90 RESET(0,2):RESET(123,43)
100 CLEAR:DEFINTA,B:A=1:B=2:POKE16406,82:SET(0,2):SET(A,B):RESET
(0,2):IFPOINT(A,B+1)103
101 FORB=B+1TO47:RESET(A,B-1):SET(A,B):IFB=43IFPOINT(A-1,B)RESET
(A,B):B=B+1:GOTO103ELSE120ELSEIFNOTPOINT(A-1,B)107ELSEIFPOINT(A,
B+1)103ELSENEXT
103 FORA=A+1TO127:RESET(A-1,B):SET(A,B):IFA=121THEN115ELSEIFNOTP
OINT(A,B+1)IFB=41THEN118ELSE101ELSEIFPOINT(A+1,B)105ELSENEXT
105 FORB=B-1TO0STEP-1:RESET(A,B+1):SET(A,B):IFNOTPOINT(A+1,B)103
ELSEIFPOINT(A,B-1)107ELSENEXT
107 FORA=A-1TO0STEP-1:RESET(A+1,B):SET(A,B):IFNOTPOINT(A,B-1)125
ELSEIFPOINT(A-1,B)101ELSENEXT
115 RESET(A,B):A=A+1:SET(A,B):IFA=123POKE16406,227:END
116 IFB=44RESET(A,B):B=B-1:SET(A,B):GOTO115ELSEFORB=B+1TO43:IFPO
INT(A+1,B)RESET(A,B-1):SET(A,B):NEXTELSERESET(A,B-1):SET(A,B)8GO
TO115
118 IFPOINT(A+2,43)=0THENRESET(A,B):B=B+3:SET(A,B):GOTO103ELSERE
SET(A,B):A=A+12IFPOINT(A+1,B)=0THEN103ELSE105
120 IFPOINT(A-2,B-1)=0THEN107ELSEIFPOINT(A+2,44)RESET(A,B):A=A+1
:GOTO105ELSERESET(A,B):B=B+1:GOTO103
125 IFPOINT(A,B-4)IFPOINT(A-1,B-2)IFNOTPOINT(A-2,B)RESET(A,B):A=
A-1:GOTO107ELSEIFB=43RESET(A,B):B=44:GOTO103ELSEIFPOINT(A-2,B)RE
SET(A,B):A=A-1:GOTO101
126 GOTO105
```

LPRINT alles einfacher!!! (Teil 1)

Unser Assembler-Freak Arnulf Sopp hat sich im Clubinfo Nr. 6 ausgiebig über das Thema "Ausgabe der Steuercodes 0, 10 und 12 an den Drucker" ausgelassen. Mit seinem in diesem Beitrag abgedruckten Assemblerlisting versucht er dem Microsoft-BASIC beizubringen, alle Steuerzeichen ungefiltert zum Drucker durchzulassen. Leider verbiegt er dabei den Interpreter jedoch so, daß danach kein (normales) LLIST mehr möglich ist. Hier nun eine einfache, leicht durchschaubare und realisierbare Möglichkeit den gleichen Effekt zu erreichen und dabei die LList-Funktion völlig unberührt lassen.

Die Stelle, an der das unten stehende Programm in den BASIC-Interpeter eingreift, ist der sogenannte DCB (DeviceControlBlock). Dieser beinhaltet, neben der Angabe des DCB-Typs, die Treiberadresse und weitere Informationen. Der Drucker-DCB nimmt die Speicherstellen 4025h - 402Bh (16421d - 16427d) ein und sieht folgendermaßen aus:

```
4025 06          DCB-Typ
4026 BD 05       Treiberadresse
4028 43          Zeilen pro Seite
4029 ??          Zeilenzähler
402A 00          unbenutzt
402B 50 52       PR (Hinweis: Druckertreiber!)
```

Da der DCB, wie ein kurzer Blick in die Memorymap zeigt, im RAM-Bereich liegt, kann er sehr leicht geändert werden. So kann man z.B die Adressen 4026h und 4027h so verändern, daß sie auf einen selbstgeschriebenen Druckertreiber zeigen. Dieser übernimmt es dann, die Zeichen zum Drucker zu senden. Der neue Druckertreiber ist bis auf die Grundfunktionen abgemagert.

```
100 LD      A,C       ;Zeichen..
110 PUSH    AF        ;      ....Retten
120 CALL    05D1      ;Druckerstatus abfragen
130 JR      NZ,$-3    ;wenn nicht READY, weiter abfragen
140 POP     AF        ;Zeichen zurückholen
150 LD      (37E8),A  ;Zeichen an Drucker ausgeben
160 RET               ;zurück zum aufrufenden Programm
                      ;(hier zum BASIC-Interpreter)
```

Ihn gilt es jetzt in den Speicher zu bringen. Dazu benutzt man am besten ein BASIC-Programm wie es hier abgedruckt ist. Der Treiber wird auf dem Platz einer Dummyvarialben gepokt und der DCB so geändert, daß er auf die entsprechende RAM-Adresse zeigt. Die REM's im Programm sollten zur Erklärung ausreichen.

```
10 LN$="ABCDEFGHIJKL" : ' Freihalten von 12 Bytes durch LN$
20 V=VARPTR(LN$):ADDR=PEEK(V+1)+PEEK(V+2)*256 : ' Ermitteln der Adresse von
   LN$ und abspeichern in ADDR
30 FORI=0TO11:READ C:POKE ADDR+I,C:NEXT I : ' Programm in LN$ poken
40 DATA 121,245,205,209,5,32,251,241,50,232,55,201 : ' Druckertreiber als
   dezimale Opcodes
50 POKE 16422,PEEK(V+1) : POKE 16423,PEEK(V+2) : ' Ändern des DCB
55 ' Die Zeilen 60 und 70 dienen nur zur Demonstration!!!
60 LPRINTCHR$(27)"K"CHR$(0)CHR$(1);
70 FORI=0TO255:LPRINTCHR$(I);:NEXT
80 POKE 16422,141:POKE 16423,5 : ' Zurücksetzen des DCB auf die "alte"
   Treiberadresse
```

Die letzte Zeile sollte man übrigens nicht vergessen, da ansonsten nach dem Löschen des Programms (NEW) der DCB auf eine Adresse zeigt, an der alles mögliche steht, nur kein Druckertreiber und damit der Absturz unvermeidbar ist.

Zuletzt möchte ich noch an mein RESTORE n - Patch erinnern, daß im Info Nr. 5 erschienen ist. Natürlich könnte man auch den neuen Druckertreiber auf die dort beschriebene Art ins BASIC einpatchen. Wer dazu Lust hat, kanns ja mal versuchen.

Hartmut Obermann

Quellenangaben:
ROM-Listing von L. Röckrath
"Alle Steuercodes verfügbar"
von R. Metzger (mc 12/83)

LPRINT alles einfacher!!! (Teil 2)

"Nach dem Hochmut kommt der Fall" besagt ein altes Sprichwort und wie recht der Volksmund mit dieser Behauptung hat, möchte ich euch heute kurz schildern.

Sicher habe ihr sowohl den Artikel "LPRINT alles" vom Arnulf (Info Nr. 6) als auch meinen Beitrag "LPRINT alles einfacher (Teil 1) gelesen. Wie euch vielleicht aufgefallen ist, war ich Assembler-Greenhorn ziemlich stolz darauf, einem so ausgefuchsten Assembler-Freak wie dem Arnulf mal zeigen zu können, daß man nicht immer um 10 Ecken herum im Interpreter herummurgeln muß, um ein relativ einfaches Ziel zu erreichen. Dies habe ich ihm bei einem Treffen des Bremerhavener Computerclubs natürlich auch gleich auf die Nase gebunden. Ich erklärte dem Arnulf meine Lösung und erntete erst ungläubiges Erstaunen und sodann den Beweis, daß diese Lösung nur eine Teillösung des Problems darstellt. Ich war baff!

Der BASIC-Interpreter versaubeutelt das Zeichen 0Ah (10d) doch tatsächlich schon, bevor es überhaupt zum Druckertreiber kommt! Das hat zur Folge, daß mit meiner Lösung alle Zeichen außer 10d bzw 0Ah ausgegeben werden können. Jetzt hieß es beschämt zum Rückzug anzutreten und mich beim Arnulf entschuldigen, was ich hiermit nochmal offiziell tue!

Der Arnulf war aber noch so nett, mir noch einen Lösungstip mitzugeben. Er wies mich darauf hin, daß der BASIC-Interpreter die Zeichen (0,10 und 12) nicht einfach nur umwandelt, sondern sie zuvor auf dem Stack (Stapelspeicher) ablegt und man sie sich von dort wieder unbeschadet herausholen könnte. Vielen Dank Arnulf!!! Auf dieser Aussage habe ich eine (beweisbar) voll funktionsfähige Lösung entwickelt, die ich euch kurz schildern möchte.

Auch bei dieser Lösung wird der DCB verbogen. Und zwar auf folgendes kurze Assemblerprogramm:

```
100 LD      (0401Bh),SP   ;Wert des Stackpointers zwischenspeichern
200 LD      IX,(0401Bh)   ;Wert des Stackpointers in das IX-Register laden
300 LD      A,(IX+11h)    ;zu druckendes Zeichen in den Accu laden
400 PUSH    AF            ;Zeichen retten
500 CALL    05D1h         ;Druckerstatus abfragen
600 JR      NZ,$-3        ;wenn Printer nicht ready, weiter abfragen
700 POP     AF            ;Zeichen zurückholen
800 LD      (037E8h),A    ;Zeichen an Drucker ausgeben
900 RET                   ;zurück zum aufrufenden Programm
```

In den Zeilen 100 und 200 wird als erstes festgestellt, auf welche Speicheradresse der Stackpointer zur Zeit zeigt. In Zeile 300 wird der Inhalt der Speicherstelle, in der die zu druckenden Zeichen zwischengelagert werden, in den Accu geladen. Die Zeilen 400 - 900 sind praktisch die gleichen wie im vorhergehenden Programm.

Auch dieses kleine Maschinenprogramm bringt man am besten in der schon bekannten Weise in einem BASIC-Programm unter (oder patcht es direkt in den Disk-BASIC-Interpreter). Hier das kurze BASIC-Programm mit eingebautem Demo.

```
10 LN"1234567890123456789212" : V=VARPTR(LN$)
20 ADDR=PEEK(V+1)+PEEK(V+2)*256
30 FORI=0TO21 : READ C : POKE ADDR+I,C : NEXT I
40 DATA 237,115,24,64,221,42,24,64,221,126,17,245,205,209,05,32,
   251,241,50,232.55,201
50 POKE 16422,PEEK(V+1) : POKE 16423,PEEK(V+2)
60 LPRINTCHR$(27)"K"CHR$(0)CHR$(1);
70 FORI=0TO255:LPRINTCHR$(I);:NEXT
80 POKE 16422,141 : POKE 16423,5
```

Wie ihr hoffentlich alle seht, ist es, mit freundlicher Hilfe von Clubkameraden, selbst einem Assembler-Greenhorn möglich Probleme zu lösen und damit zu Erfolgserlebnissen zu kommen (wenn auch spät und über Umwege). Ich hoffe, daß mancher von euch diesem Beispiel (den Hochmut könnt ihr ja weglassen) folgen wird und so ein bißchen Freude an der Assembler-programmiererei findet. Viel Spaß dabei, Euer Hartmut Obermann

Wie im letzten Teil kurz erwaehnt, folgt hier nun die dritte Division, die DATA DIVISION.

Hier werden allen Variablen Datenfelder zugewiesen, der Aufbau wird festgelegt und weiterhin der Typ (alpha oder numerisch) und schliesslich eine eventuelle Druck-Aufbereitung.

Was das nun alles im einzelnen bedeutet, wird im folgenden erlaeutert.

```
1----67A---B-------------------------------------- usw... 72
       DATA DIVISION.
       ***************
       *
        FILE SECTION.
       *
        FD DATUM
            BLOCK CONTAINS 1 RECORD
            RECORD CONTAINS 10 CHARACTERS
            LABEL RECORD IS STANDARD
            DATA RECORD IS DATUM-SATZ.
        01 DATUM-SATZ.
         02 TAG        PIC X(10).
```

Was bedeutet das nun ?? - An dieser Stelle verweise ich auf den Teil 2 der Cobol-Einfuehrung. Dort wird in der ENVIRONMENT DIVISION innerhalb der ersten SELECT-Anweisung der physischen Datei 'HPTDATUM/DAT:1' der logische Dateiname 'DATUM' zugeordnent. Weiterhin wird festgelegt, dass es sich um eine Relative-Datei mit direktem Zugriff ueber die Satznummer (Variable:SATZ) handelt.

Ueber den Aufbau des Datensatzes wurde alleerdings nichts gesagt. Dies geschieht nun in der DATA-DIVISION (s.o).

Dr vermeintlich grosse Aufwand fuer die Datei und Datenbeschreibung taeuscht. Bie umfangreichen Dateien mit Saetzen bis zu einem halben Kb macht sich das Verfahren positiv bemerkbar.

Aber zurueck zur DATA DIVISION. Unter der FILE SECTION werden zuerst alle Dateien mit 'ihren' Datensaetzen versorgt.

```
FD (File Description) N a m e      (Zuordnung Datensatz-Datei)
BLOCK CONTAINS                     (Blockungs-Faktor)
RECORD CONTAINS xx CHARACTERS      (Anzahl Bytes per Record)
LABEL RECORD IS                    (bei Dateien: STANDARD
                                    beim Drucker OMITTED)
DATA RECORD IS                     (Name des Datensatzes)


01 DATUM-SATZ                      (Beschreibung Datensatz)
 02 TAG         PIC X(10)          (Datenformat)
```

Nun, so verwirrend ist es gar nicht. Bis einschliesslich DATA RECORD muesste alles klar sein. Ueber das 01-er und 02-er Feld machen wir uns zunaechst noch keine Gedanken.

Gehen wir zunaechst noch auf eine zweite Datei-Beschreibung ein: auf die 'STAMMDAT/DAT:1', die in der SELECT Anweisung den logischen Satz 'DATEN' zugeordnet bekommen hat.

Bei dieser Datei handelt es sich um eine Stammdaten-Datei fuer eine Adressverwaltung, dementsprechend gross ist der Datensatz.

```
FD DATEN
   BLOCK CONTAINS 10 RECORDS           (nur Dokumentation)
   RECORD CONTAINS 274 CHARACTERS      (Satzlaenge)
   LABEL RECORD IS STANDARD            (Label f. Dat-Header)
   DATA RECORD IS STAMM-SATZ.          (Name Datensatz)
01 STAMM-SATZ.
02 E-NUMMER          PIC X(8).
02 E-NAME            PIC X(30).
02 E-VORN            PIC X(20).
02 E-STR             PIC X(25).
02 E-PLZ             PIC X(6).
02 E-ORT             PIC X(25).
02 E-TEL             PIC X(15).
02 E-GEB             PIC X(10).
02 E-TXT1            PIC X(45).
02 E-TXT2            PIC X(45).
02 E-TXT3            PIC X(45).
```

So, das waere die FD-Anweisung fuer die Stammdaten-Datei. Allen Variablen sind Datentyp und Datenlaenge innerhalb des Datensatzes zugeordnet worden. Fragen sind aber sicherlich noch einige offen: Was bedeutet die 'komische' Durchnummerierung der einzelnen Variablen ? oder: Was hat es mit dem PIC und dem X dahinter auf sich ??

Ein klein wenig Geduld noch, die Erklaerung folgt in der WORKING-STORAGE SECTION.

Halten wir also noch einmal fest: Die DATA DIVISION ordnet allen Variablen, egal ob reine Arbeitsvariablen, wie Merker, Zaehler usw., oder Variablen die zu einer Datei gehoeren, den Datentyp (alpha, numerisch, alphanumerisch) und die Laenge zu. Die Variablen-Namen koennen dabei bis zu 32 Stellen lang sein. Unterschieden werden die Arbeitsvariablen und Datensatz-Variablen bei der Zuordnung der SECTION's.

Arbeits-Variablen 'stehen unter' der WORKING-STORAGE SECTION; Variablen zur Datei stehen unter der FILE SECTION, jeweils getrennt unter den einzelnen FD-Anweisungen.

(Der Vollstaendigkeit halber sei noch erwaehnt,dass innerhalb der FILE SECTION gleiche Variablen-Namen verwendet werden duerfen. So kann es unter der FD DATUM z.B. einen TAG geben, ebenfalls unter der FD DATEN. Beide Variablen koennen unterschiedliche Informationen enthalten. Die Unterscheidung findet innerhalb der Programm-Anweisung statt.)

Kommen wir nun zur WORKING-STORAGE SECTION. Hier werden noch einige Dinge auftauchen, die zunaechst Verwirrung ausloesen koennen - aber trotz allem: immer daran denken, dass, egal wie es aussieht, lediglich Variablen-Namen, Länge und Datentyp zugeordnet werden - sonst nichts.

Ein Hinweis noch: alle hier angefuehrten Beispiele stammen aus einem Stammdaten-Verwaltungs Paket, genannt :ADRESSEN TEXT', von mir verfasst und geschrieben. Eine Programm-Beschreibung und das Source Listing koennen von mir angefordert werden. Die Sourcen sind als Listing oder als ASCII-Files auf Diskette zu erhalten. Mir ist es lieber, die Sourcen auf Disk zu verschicken, ca. 60 Kb, der Ausdruck dauert naemlich ewig. Die Programmbeschreibung gibt es als Kopien. (Bei Source-Anforderung: bitte Disk mitschicken !)

So, nun aber zum Kern zurueck:

```
1----67A---B---------------------------------- usw... 72
       WORKING-STORAGE SECTION.
       77 SATZ       PIC 9 VALUE 1.
       77 WAHL       PIC 9 VALUE 0.
       77 MERKER     PIC 9 VALUE 0.
       77 ZEILE      PIC 99.
       77 DP-WAHL    PIC 9 VALUE 0.
       77 C-1        PIC X.
       77 C-2        PIC X.
       77 E-DAT      PIC X(10).
       77 STRICH     PIC X(63).
       77 LEERZEILE  PIC X(63).
```

Es wurden also 10 Variablen definiert, die im Programm als reine Arbeitsfelder Verwendung finden.

Nun aber die Erklaerungen:

PIC heisst eigentlic PICTURE (Bild) und bedeutet, dass in die jeweilige Variable ein bestimmtes 'Bild' gelegt wird, aus dem sich der Variablentyp, die Laenge und der Aufbau der Variablen ersehen lassen.

PIC X: eine Variable, der PIC X zugeordnet wird, hat eine Laenge von einem Byte (ein X=ein Byte).

PIC XX: Variable hat eine Laenge von zwei Byte.
PIC XXX: drei Byte
PIC XXXX: vier Byte
PIC XXXXX: fuenf Byte, aber jetzt wird es langsam unuebersichtlich

Man hilft sich mit folgendem Trick:
PIC X(5):fuenf Byte
PIC X(9):neun Byte
PIC X(1):ein Byte (geht natuerlich auch)

Das X hinter der PIC Anweisung sagt aber auch etwas ueber den Datentyp aus: X bedeutet alphanumerisch. In diesem Zusammenhang bedeutet 9 hinter der PIC Anweisung: numerisch und A hinter der PIC Anweisung alphabetisch.

Also wird mit z.B.

```
77 TRALLA              PIC X(10)
```

die Variable 'TRALLA' mit 10 Stellen, alphanumerisch definiert.

So weit, so gut: Nun Kommen die Zahlen vor den Variablen, mal ist sie 01, mal 02 und dann 77. Ganz einfach, die Variablen haben eine gewisse Prioritaet, einfach ausgedrueckt: Variablen mit kleiner Nummer (01) haben hohe und Variablen mit ~~niedriger~~ höher Nummer (z.B. 77) niedrige Prioritaet.

Aber das geht jetzt nicht stur von 01 bis 99 oder so:

Ueber 50 gibt es nur noch die Stufen: 66,77,88. Die 66-er und 88-er lassen wir an dieser Stelle weg - bleiben also nur noch die 77-er. (66,88 dienen z.B. fuer LINKAGE und andere feine Sachen, mit denen sich der erfahrene Programmierer auseinandersetzen kann)

Die Nummern vor den Variablen heissen auch nicht 'Nummern', sondern Stufen - also, die Stufe 77 kennzeichnet Arbeitsfelder, allgemein.

Was heisst nun allgemeine Arbeitsfelder ?? Das sind Variablen, von denen jeweils eine Var eine eigene unabhaengige Variable darstellt, z.B. Zaehler.

Nun gibt es aber auch z.B. folgendes:

```
10 CHEF-VARIABLE.
 20 ABTEILUNGSLEITER-1          PIC X(10).
 20 ABTEILUNGSLEITER-2          PIC X(10).
  30 ANGESTELLTER-1             PIC X(10).
```

Wie schon an den (willkuehrlich) gewaehlten Variablen-Namen zu sehen ist kann man, wie auch an den Stufen, eine Abhaengigkeit und Rangordnung feststellen. Man kann nun innerhalb des Programms jede Variable einzeln ansprechen und Daten zuordnen.

Also z.B.

```
MOVE "HEINZ" TO ANGESTELLTER-1.
MOVE "HERMANN" TO ABTEILUNGSLEITER-1.
```

Mit der uebergeordneten Variablen (CHEF-VARIABLE) kann man sich die ganze Angelegenheit aber sehr stark vereinfachen:

```
          1          2          3
1---------01---------01---------0     (Mit MOVE aufbereiteter
HEINZ     HERMANN    GUENTHER          Datensatz)
```

Wenn ich nun den Befehl gebe: MOVE Datensatz TO CHEFVARIABLE, so habe ich mit einem einzigen Befehl drei Namen zu drei Variablen zugeordnet.

Und genau nach dem gleichen Verfahren wird auch bei den niedrigeren Stufen verfahren.

Wenn ich die Variable mit der Stufe 01 anspreche, so kann ich alle ihr untergeordneten Variablen mit einem Befehl mit Daten fuellen oder aber mehrere Variablen mit einem Befehl von einem Datenbereich in einen anderen MOVEn.

Die Stufen von 01 bis 49 sind also dazu gedacht, um Datenfelder, die voneinander abhaengig sind, mit einem einzigen Befehl bequem handhaben zu koennen. Wird die Stufe 01 angesprochen, so kann man ueber die Stufen 02 ... 49 verfuegen; entsprechendes gilt natuelich auch wenn z.B. die Stufe 10 angesprochen wird. (Verfuegung ueber 11 ... 49).

Siehe dazu auch im Beispiel:

```
01 STAMM-SATZ.
 02 E-NUMMER          PIC X(8).
usw.
```

Das Einruecken der Stufen geschieht willkuehrlich und dient nur zur besseren Dokumentation. Man kann auch alle Stufen an der Position 8 'beginnen' lassen.

Zurueck zur PIC-Anweisung:
Der Aufbau duerfte nun klar sein. Also noch einmal: Das X steht fuer eine alphanumerische Variable, das A fuer eine reine alphabetische Variable und schliesslich die 9 fuer eine numerische Variable. Die Anzahl der Xen, der As oder der 9nen schliesslich gibt die Laenge der Variable an.

Es gibt aber gerade bei den numerische Variablen noch viele Besonderheiten, von denen hier aber nur die wichtigsten angesprochen werden sollen:

Ganz allgemein: es gibt nur Integer-Variablen ! (Das stimmt natuerlich nicht, aber es dient zur besseren Darstellung).

Zahl	Feld-Aufbereitung	Feld-Inhalt
10	PIC 9(2)	!1!0!
1234	PIC 9(4)	!1!2!3!4!
1234	PIC 9(6)	! ! !1!2!3!4!
12.34	PIC 99V99	!1!2!3!4!
-2.34	PIC S9V99	!2!3!4!
12.34	PIC 99.99	!1!2!.!3!4!
12.34	PIC 999.99	!0!1!2!.!3!4!
12.34	PIC ZZ9.99	! !1!2!.!3!4!
-3.2	PIC --9.99	! !-!3!.!2!0!

Und nun zur Erklaerung:

Mit Feld-Inhalt ist die Darstellung der Variablen-Inhalte im Ram gemeint. Die ersten beiden Beispiele sind klar. Das dritte Beispiel zeigt die Ausrichtung der Daten im Feld, wenn die Daten der Variable kleiner sind als das entsprechende Datenfeld.

Beispiel vier und fuenf sind etwas neues. Es handelt sich hier um Datenfelder, die fuer Dezimalzahlen aufbereitet sind - Das 'V' gibt dabei die Position des Kommas an, das Komma selbst wird in dem Datenfeld nicht angezeigt.

Das Beispiel fuenf unterscheidet sich von dem vierten Beispiel dahingehend, dass statt der ersten '9' ein 'S' bei der Aufbereitung verwendet wurde. Dieses 'S' bedeutet nichts anderes als 'Sign', also Vorzeichen. Man sollte Rechenfelder immer mit diesem S ausstatten - nur dann besteht die Moeglichkeit auch mit negativen Zahlen zu rechnen.

Die nun folgenden vier Beispiele unterscheiden sich von dem bisher gesagten ganz erheblich:

Waren bisher alle behandelten Darstellungsformen nur zum Rechnen gedacht, so muss nun natuerlich noch die Druckaufbereitung behandelt werden.

Das ist nun etwas ganz neues fuer Basic-Anwender. Die Datenfelder zum Rechen und Darstellen (egal ob auf dem Bildschirm oder Drucker) sind nicht identisch mit den Datenfeldern zum Rechen. Das mutet vielleicht etwas seltsam an - ist jedoch eine ganz sinnvolle Einrichtung da man bei Cobol viele Moeglichkeiten zum Ablegen der numerischen Daten im Ram hat (z.B. gepackt).

Wenn alle Moeglichkeiten hier dargestellt werden sollten, so liesse sich zu diesem Thema ein eigenstaendiger Kurs schreiben.

Nun aber zu den Druckaufbereitungen:

Bei PIC 99.99 wird eine Dezimalzahl in der gewohnten Weise angezeigt (man kann aber nicht mit ihr Rechnen). Damit die Zahl angezeigt werden kann, muss sie zunaechst erst einmal von dem Rechnfeld ins Druck-Feld geMOVEd werden.

MOVE RE-Feld TO DRUCK-Feld.

Wenn das RE-Feld Beispiel 2 ist, dann steht im Druck-Feld:
12.34

Ist das RE-Feld Beispiel 5, so steht im Druck-Feld:
02.34

Aber wo ist das negative Vorzeichen und wieso steht vor der 2 eine 0 ??

Die Erklaerung ist einfach:
Fuer diese Darstellung ist die Druck-Aufbereitung nicht geeignet, man muss eine andere waehlen.

Zur Vor-Null-Unterdrueckung gibt es das 'Z' (Zero suppress) Wuerde man nun das Beispiel Nr. 5 in das Beispiel Nr.8 (PIC ZZ9.99) moven, so wuerde die Vor-Null unterdrueckt werden. das Vorzeichen allerdings ist immer noch nicht beruecksichtigt.

Moven wir also das Rechenfeld (PIC S9V99) mit dem Inhalt -234 (es ist dazimal aufbereitet, (V)), in das Beispiel Nr. 10 (PIC --9.99).

Bei dieser Druckaufbereitung wuerde auf dem Bildschirm der Inhalt der Variablen so angezeigt werden, wie wir es gewohnt sind.

Diese Art der Programmierung ist nur halb so umstaendlich wie sie jetzt erscheint. Wenn man alle Moeglichkeiten ausnutzt, hat man ein Vielfaches der vom Basic her gewohnten Darstellungsformen (PRINT USING ist dagegen ueberhaupt nichts).

Jetzt fehlen eigentlich nur noch die Befehle fuer den Programm-Ablauf (in der PROCEDURE DIVISION).

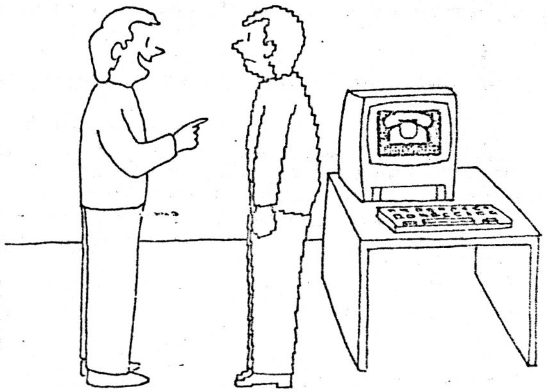
Man wird staunen, mit wie wenigen Befehlen man ein komfortables Programm schreiben kann. (Dafuer musste man ja schon einiges an Arbeit in der DATA DIVISION auf sich nehmen.

Die Eingabesteuerung und Ausgabemoeglichkeiten sind derart zahlreich, dass man angenehm ueberrascht sein wird (z.B automatisches Carrige Return bei Erreichen des Endes bei Eingabefeldern oder automatisches Prompt in der Laenge des Eingabefeldes (das ist der Unterstrich, den man bei Eingabemasken haeufig finded, z.B. NAME: ________), korrigiert man etwas bei seiner Eingabe (Pfeil links), so werden die Promts, also der Unterstrich, wieder automatisch gesetzt usw.)

Aber man muss es erst einmal ausprobieren....

Noch ein Hinweis in eigener Sache:
Mich interessiert natuerlich brennend, ob mein kleiner Kurs verstaendlich genug ist, und vor alle Dingen, ob er ueberhaupt von Interesse fuer die Clubmitglieder ist.

Wenn also jemand Zeit und Musse hat, so moege er mir vielleicht ein paar Zeilen schreiben.



„Ich sehe schon ... sie gehen voll in Ihrer Arbeit auf"

Tips & Tricks -- Tips & Tricks -- Tips

Im 3. INFO wurde ein PEEK über die Druckerabfrage für den TRS-80 gebracht.

```
100 IF PEEK(14312) > 127 Then PRINT "Drucker anschalten !": GOTO 100
```

Im 6. INFO fragte Werner wie er den Befehl für seinen GENIE verwenden kann.

Da der Genie den Drucker über seinen User-Port ansteuert muß nun der Port abgefragt werden. Dies geschied mit folgendem Syntax:

```
100 IF INP(253) > 127 Then PRINT "Drucker anschalten !": GOTO 100
```

Beim EPSON und dem GENIE wurden folgende Werte ermittelt:

INP(253) = 63 Drucker aus oder ok.
INP(253) = 159 off Line
INP(253) = 223 Paperout
INP(253) = 255 Kein Druckerkabel angeschlossen.

--Noch kurz vor INFO-Schluß von Gerald und Hartmut per Telefon.--

Zap im SUPERZAP!

Einer der größten Mängel von SUPERZAP ist die Tatsache, daß man die Sectoreninhalte nur hexadezimal modifizieren kann. Vor allem dort, wo längere Texte geändert werden sollen, ist dieser Mangel sehr störend! Dieses Manko ist nun abgestellt. Das unten abgedruckte Zap ermöglicht es, einen Sector nicht nur in HEX, sondern auch in ASCII zu ändern.
Die Umschaltung erfolgt durch die Taste <CLEAR>. Die Steuerung des Cursors geschieht weiterhin mit den Pfeiltasten, mit der Ausnahme, daß man, um eine Zeile nach oben zu springen, <SHIFT> + ↑ drücken muß.

```
DRV 00   0100 0052 0000 0000 0000 0000 0000 0000  A§R§§§§§§§§§§§§§
0   10   0000 0000 0000 0000 0000 0000 00F6 FE7B  §§§§§§§§§§§§§§
OH  20   2802 FE1B CAF7 5CFE 1FC2 345D E521 3952  (B A  ö   4Ü !9R
    30   2274 5C21 5B03 22EF 5DE1 C369 5C33 333E  "tö!AC" Ü  iö33>
DRS 40   6C32 E15D 79E6 F06F 2600 2929 79E6 0FB5   2 üy  o&§))y O
570 50   6F11 303C CDDE 5DE1 CBC0 FE20 380C CB48  oO0<  Ü     8L H
23AH60   CB88 C2A6 5D1E 00C3 905C 4FCB 80FE 08CA     Ü^§  öO   H
    70   AE5C CBC0 FE1F C2AE 5CE5 21BC 5D22 745C   ö     öl ! ü"tö
    80   2105 622E F55D E1CB 8018 AF00 0000 0000  !Eb. Ü    X§§§§§
    90   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
    A0   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
    B0   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
FRS C0   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
0   D0   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
OH  E0   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
    F0   0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
```

Der obenstehende Sectordump stellt den Sector 0 von SUPERZAP/CMD dar. Dieser Sector enthält im Original nur Nullen. Die Änderungen sind also nicht schwer zu erkennen. Weiterhin müßen noch vier Bytes im Sector 9 geändert werden. Beginnend beim relativen Byte BEh werden folgende Änderungen vorgenommen:
C2 34 5D 7D in C2 1A 52 7D.

Viel Spaß beim zapen, euer *Hartmut Obermann.*

Quelle: Clubinfo des TRS80-Computerclub München

„Wenn die Affen es dahin bringen könnten, Langeweile zu haben, so könnten sie Menschen werden."

Johann Wolfgang von Goethe, deutscher Dichter (1749–1832)

Noch zwei Tips zu SCRIPSIT:

Änderung der Voreinstellung für die Anzahl Zeichen pro Zeile auf dem Bildschirm:

0,A6 von 0E 3C FD = 60 Zeichen pro Zeile
nach 0E 3F FD = 63 Zeichen pro Zeile

Umdefinieren einiger SCRIPSIT-Kommando-Tasten

Mit den folgenden Änderungen werden Kommando-Tasten bei Scripsit so geändert, das die Bezeichnung der Taste der Funktion des Kommandos entspricht (z.B. <CONTROL><I> statt <CONTROL><S> für Insert) :

Bedeutung	Alt Taste	Alt Code	Neu Taste	Neu Code	Speicher
Wort	Z	1A	W	17	7A0D 79E4
Paragraph	C	03	P	10	79A5 7A0E 79E2
Zeile	X	18	Z	1A	79A4 79E1 568B
Blanks	F	06	B	02	79E0
Insert	S	13	I	09	79A1

Literatur: 80 Micro 3/83/18

Josef Konrad

Nochmal: Ausgabeumschaltung in BASIC
(diesmal ohne Fehler!?!)

Schon im letzten Info war das hier abgedruckte Programm veröffentlicht. Leider habe ich nicht beachtet, daß das von mir benutzte Textverarbeitungsprogramm das $-Zeichen als Steuerzeichen benutzt und damit das BASIC-Programm verstümmelt und mit Fehlern spickt. Hier die korrigierte und (hoffentlich) fehlerfreie Version:

```
10 CLEAR 1000 : CLS
20 PRINT @ 0,"Ausgabe auf (D)rucker oder (B)ildschirm?"
30 A$=INKEY$ : IF A$="D" OR A$="d" THEN 50
40 IF A$="B" OR A$="b" THEN 60 ELSE GOTO 30
50 CMD"ROUTE,DO,PR" : GOTO70
60 CMD"ROUTE,CLEAR"
70 FOR X=32 TO 191 : PRINT CHR$(X);" "; : NEXT : RUN
```

Hartmut Obermann

Bei einem Wettlauf zwischen einer Schnecke und einem Kamel gewann zum Erstaunen aller die Schnecke. Sofort wurde dieses Ereignis von einer Sonderkommission untersucht. Sie stellte fest: Das Kamel hatte den Dienstweg eingeschlagen.

Basic-Dolmetscher gesucht

Basic ist die dominierende Sprache der Heimcomputer. Aber es herrscht ein unglaubliches Sprachgewirr. Jeder Hersteller hat offenbar den Ehrgeiz, einen eigenen Dialekt zu schaffen. Deshalb ist es mühsam und gerade für den Anfänger unmöglich, ein fremdes Listing für seinen Computer zu übersetzen.

Der Mann aus Schleswig-Holstein sagt: »We snakt Platt!«. Der Niederbayer erklärt: »I red boarisch!«. Würden beide auf ihrem Dialekt beharren, könnten sie sich kaum miteinander unterhalten. Glücklicherweise verstehen beide Hochdeutsch. Damit ist das Gespräch gerettet. Hochdeutsch bildet hier also den Kern der Sprache, den beide »Volksstämme« verstehen. Ähnlich gelagert ist auch das Verhältnis bei Basic. Entwickelt wurde sie vor genau 20 Jahren von den Professoren John G. Kemeny und Thomas E. Kurtz. Am Dartmouth College wurde sie zuerst eingeführt und trat von dort aus ihren Siegeszug an. Heute gibt es über 250 Basic-Dialekte, die alle um den ursprünglichen Kern aufgebaut sind. Jeder Hersteller entwickelte für seinen Heimcomputer eine besonders geeignete Version. An einen Standard oder gar eine Norm brauchte sich keiner zu halten; das wurde natürlich ausgenutzt. Zum großen Ärger und Verdruß der Anwender. Sie sitzen nämlich vor ihrem Heimcomputer und haben nur ihr Basic zur Verfügung. Da gibt es keinen Umschalter für die unterschiedlichen Dialekte. Interessante Programme müssen also »zu Fuß« von einem »Slang« in den anderen übersetzt werden. Dabei kommt es natürlich zu Problemen. Denn nicht jeder Hersteller bringt seinem Heimcomputer alle Befehle der Sprache Basic bei.

Außerdem werden gleichartige Befehle anders genannt. Ein klassisches Beispiel dafür ist der Befehl zum Löschen des Bildschrims. In der »Urform«, die auf den meisten Heimcomputern eingebaut ist, lautet er »CLS«. Übersetzt heißt das Clear Screen — lösche den Bildschirm. Beim Apple II und allen Commodore-Computern zum Beispiel führt diese Eingabe zu einem Syntax Error. Das heißt soviel wie: »Den Befehl kenne ich nicht«. Der Apple hätte hier gerne die Eingabe »Home«, bei den Commodore-Computern ist »Print Chr$(147)« erwünscht und beim TI 99/4A muß mit »Call Clear« das entsprechende Maschinenprogramm aufgerufen werden. Diese Eingaben sind immer auf den Einsatz in Programmen ausgelegt. Zum direkten Löschen des Bildschirms haben die meisten Computer eine entsprechende Tastenfunktion eingebaut. Würde für die Commodore-Computer tatsächlich »Print Chr$(147)« in einem Listing abgedruckt, gäbe es bei der Konvertierung in andere Dialekte keine Probleme. Hier werden aber besondere Controll-Codes eingesetzt (Bild 1).

```
1 REM "DIESE VERSION GIBT EINEN
2 REM "BLINKENDEN TEXT BEI DEN
3 REM "COMMODORE- COMPUTERN AUS
4 REM "DIE STEUERCODES FUER ANDERE
5 REM "COMPUTER SIND DER TABELLE
6 REM "ZU ENTNEHMEN.
7 REM "MIT CHR$(145) WIRD DER
8 REM "CURSOR EINE ZEILE NACH
9 REM "OBEN BEWEGT
10 PRINT" TEXT ";PRINTCHR$(145);
15 REM "ZEITDAUER FUER TEXTDARSTELLUNG
20 FOR A = 1 TO 100: NEXT A
30 PRINT"       ";PRINTCHR$(145);
35 REM "FUER DIESE ZEIT WIRD KEIN
36 REM "TEXT AUSGEGEBEN
40 FOR A = 1 TO 100: NEXT A
50 GOTO 10
```

Listing 1. Durch zwei For-To-Next-Schleifen läßt sich der Befehl »Flash« simulieren


```
10 REM KONTROLL- CODES
20 REM FUER COMMODORE
30 REM COMPUTER
40 PRINT"■": REM BILDSCHIRM
50 '           REM LOESCHEN
60 PRINT"■": REM CURSOR IN
70 '           REM DIE OBERE
80 '           REM LINKE ECKE
90 PRINT"■": REM CURSOR RECHTS
91 PRINT"■": REM CURSOR LINKS
92 PRINT"■": REM CURSOR RUNTER
93 PRINT"■": REM CURSOR RAUF
```

Bild 1. So stellen Commodore-Computer Control-Codes dar

Auch bei der Benennung von Basic-Befehlen kommt es zu Unterschieden. Wird zum Beispiel in einem Programm für den Sharp-Computer MZ-700 der Befehl »SPC (b)« aufgelistet, so wird eine Zeichenkette erzeugt, die, dem Wert der Variablen »b« entsprechend, aus Leerzeichen besteht. Bei anderen Computern wiederum wird mit diesem Befehl bei einer Ausgabe mit »Print« die durch »b« angegebene Anzahl von Zeichen in einer Zeile übersprungen.

Zusätzlich erschwert wird die Konvertierung durch irreführende Tastenbeschriftungen. Soll zum Beispiel der Colour Genie die Anweisung »Clear« ausführen, so ist damit gemeint daß er alle Variablen löschen soll. Auf der Tastatur dieses Computers findet sich aber auch eine Taste mit der Beschriftung »Clear«. Durch einen Druck auf diese Taste wird der Bildschirm gelöscht. Die Befehle »Print« und »Input«, so erstaunlich es klingt, sind noch bei allen Basic-Dialekten gleich. Durch den Befehl »Print« erfolgt die Ausgabe einer Leerzeile, »Print a« gibt den Inhalt der Variablen »a« aus und »Print a$« führt zur Ausgabe der Textvariablen »a$«. Besondere Aufmerksamkeit wird durch blinkende Texte erregt. Hierfür stellt der Apple II-Computer und der Sinclair Spectrum die Funktion »Flash« zur Verfügung. Andere Computer können diese Funktion mit einem kleinen Unterprogramm simulieren (Listing 1). Beim Ausdruck von Daten erleichtert eine formatierte Ausgabe den Überblick. In vielen Basic-Dialekten findet man hierzu den Befehl »Print Using«. Mit einer Anordnung von Doppelkreuzen »#« wird vor der Ausgabe ein Format festgelegt. So führt zum Beispiel die Anweisung: Print Using »####.###«,A dazu, daß die Variable »A« immer mit drei Nachkommastellen ausgedruckt wird. Diese Ausgabeform läßt sich bei den meisten Heimcomputern durch eine geeignete Unterroutine simulieren (Listing 2). Mit der »Input«-Anweisung werden während eines

```
1 REM "MIT DIESEM PROGRAMM
2 REM "WIRD DIE PRINT USING
3 REM "FUNKTION SIMULIERT
4 REM
10 INPUT"WIEVIELE STELLEN ?";Z
20 INPUT"ZAHL";A
30 GOSUB 9000
35 PRINT"AUS";A;" WURDE ";ZF$
50 GOTO 10
60 REM "MIT DIESEM BEISPIEL
70 REM "WIRD IN DIE ROUTINE
80 REM "AB ZEILE 9000 GESPRUNGEN
9000 F$=STR$(A)
9010 FOR X= 1 TO LEN(F$)
9020 IFMID$(F$,X,1)="." THEN 9050
9030 NEXT X: ZF$=F$
9040 RETURN
9050 F$=F$+"0000000000000000"
9055 ZF$=LEFT$(F$,X+Z)
9060 RETURN
```

Listing 2. Die Funktion »Print Using« mit einem Unterprogramm simuliert

```
1 REM "AUCH MIT DER 'GET'ABFRAGE
2 REM "KOENNEN ZEICHENKETTEN
3 REM "EINGEGEBEN WERDEN
4 REM "DIESES PROGRAMM VERLAENGERT
5 REM "DIE TEXTVARIABLE 'B$'
6 REM "SOLANGE, BIS 'RETURN'
7 REM "EINGEGEBEN WIRD.
10 GET A$
20 IF A$=""THEN10
25 IF A$=CHR$(13)THEN 60
30 B$= B$+A$
40 PRINTB$
50 GOTO10
60 END
```

Listing 3. Eingabe einer Zeichenkette mit der »Get«-Abfrage

```
1 REM "DER TEXTVARIABLEN 'A$'
2 REM "WIRD DER TEXT '1000'
3 REM "ZUGEORDNET. DIE 'VAL'
4 REM "FUNKTION WANDELT DARAUS
5 REM "DIE NUMERISCHE VARIABLE
6 REM "MIT DEM WERT '1000'
10 A$= "1000"
20 PRINT A$;" TEXT "
30 B = VAL(A$)
40 PRINT B;" ZAHL "
50 C = B+234
60 PRINT C;" VARIABLE + ZAHL "
70 REM "SO LASSEN SICH TEXTE
71 REM "MATHEMATISCH BEEINFLUSSEN
72 REM "DIE FUNKTION 'STR$'
73 REM "WANDELT DIE ZAHL
74 REM "WIEDER IN EINE TEXT-
75 REM "VARIABLE UM.
80 A$=STR$(C)
90 PRINTA$;" TEXT "
```

Listing 4. Mit Textvariablen rechnen

Programmlaufs Variablen eingegeben. Dabei wartet das Programm solange, bis die Eingabe durch ein »Return« abgeschlossen wird. Für viele Aufgaben ist diese Lösung indiskutabel, zum Beispiel wenn ein Programm durch Tastendrücke gesteuert wird. Auch wenn keine Taste gedrückt ist, sollen Berechnungen durchgeführt werden. Die Hauptanwendung liegt hier bei Spielen. Dafür bietet sich die »Get« oder »Inkey« Funktion an. Aber nicht jede »Get«-Abfrage arbeitet gleich. Das Basic des Apple II unterbricht das laufende Programm auch bei dieser Abfrage und arbeitet erst nach einem Tastendruck weiter (ohne »Return«). Bei den Commodore-Computern wird jedoch, wie bei allen »Inkey«-Befehlen, das Programm nicht zur Dateneingabe angehalten. Soll hier eine Entscheidung in Abhängigkeit eines Tastendrucks getroffen werden, muß in den nachfolgenden Programmzeilen gegebenfalls ein Rücksprung zur »Get«- oder »Inkey«-Abfrage erfolgen. Beide Möglichkeiten haben gemeinsam, daß sie immer nur ein Zeichen verarbeiten. Zur Eingabe einer Zeichenkette kann ein entsprechendes Programm dienen (Listing 3). Zur Verarbeitung von Zeichenketten stellen die meisten Heimcomputer eine große Befehlsauswahl zur Verfügung. Die Befehle »Val(A$)« und »Str$(A)« sind in fast allen Dialekten enthalten. Außer bei den Computern von Atari. Die kennen diese Befehle überhaupt nicht. Dabei handelt es sich um sehr praktische Befehle, die eine Beeinflussung von Zeichenketten nach mathematischen Regeln erlauben (Listing 4). Mit »Val(A$)« wird die Textvariable A$ in eine numerische Variable umgewandelt. Diese kann nun durch mathematische Ausdrücke verändert werden. Danach kann durch den Befehl »Str$(A)« dieser Ausdruck wieder in eine Textvariable zurückverwandelt werden. Enthält die Textvariable A$ außer Zahlen auch Buchstaben, so übernimmt der Befehl »Val(A$)« nur die linksbündigen Zahlen. Zum Beispiel wird aus A$=»1234DER56« die Zahl »1234«. Die nachfolgenden Ziffern werden ignoriert.

Stößt man in einem Listing auf den Befehl »String$(10,»*«)«, so handelt es sich dabei um eine Funktion, die einen String erzeugt, der zehnmal das Zeichen »*« enthält. Diese Funktion kann relativ leicht simuliert werden (Listing 5). Ebenso leicht läßt sich der Befehl »Swap« ersetzen. Dabei handelt es sich um den Austausch von zwei Variablen. Ist zum Beispiel in der Variablen »A« der Wert »5«, und in der Variablen »B« der Wert »8« gespeichert, so ist nach Ausführung des Befehls »Swap (A,B)« der Inhalt von »A« und »B« getauscht. Um diese Funktion in den jeweiligen Basic-Dialekt umzusetzen, bietet sich der Austausch über eine Hilfsvariable an (Listing 6).

Bei allen Programmbeispielen sollte auf das richtige Einsetzen der Variablen geachtet werden. Solche Variablennamen, die im Programm bereits benutzt werden, können natürlich nicht für eine Befehlssimulierung eingesetzt werden. Beim Ersatz von Befehlen durch Befehlsfolgen verlangsamt sich natürlich die Arbeitsgeschwindigkeit.

In dieser Ausgabe befaßten wir uns mit Befehlen, die Textein- und -ausgabe sowie die Verarbeitung von Zeichenketten ausführen. Der nächste Teil behandelt die unterschiedlichen Grafikmöglichkeiten.

siehe Übersicht auf Seite 70

(rs)

```
1 REM DER BEFEHL 'STRING$'
2 REM KANN SIMULIERT
3 REM WERDEN. DAZU
4 REM DIESES KLEINE
5 REM BEISPIEL
10 L = 20
15 REM IN 'L' IST DIE
16 REM LAENGE DER ZEICHEN-
17 REM KETTE ENTHALTEN
20 A$ = "#"
25 REM A$ ENTHAELT DAS
26 REM AUSZUGEBENDE ZEICHEN
30 FOR S = 1 TO L
40 PRINTA$;
50 NEXT S
60 REM WO IN ANDEREN
70 REM PROGRAMMEN DER
80 REM 'STRING$' BEFEHL
90 REM AUFTAUCHT, KANN
100 REM ER DURCH DIESE
110 REM ZEILEN ERSETZT
120 REM WERDEN.
```

◄ *Listing 5. »String$« — für viele Computer ein Fremdwort*

```
1 REM DIE 'SWAP' FUNKTION
2 REM TAUSCHT
3 REM ZWEI VARIABLEN
10 INPUT " ZAHL 1 ";A
20 INPUT " ZAHL 2 ";B
30 C=A:A=B:B=C
40 PRINT A,B
50 REM DIESE BEFEHLSFOLGE
60 REM SIMULIERT DIESEN
70 REM BEFEHL
```

▲ *Listing 6. Der Wert einer Variablen kann auch ohne »Swap« getauscht werden*

Funktionen / Computer	Apple II	Atari 600/800 XL	Commodore, VC 20, C 64, 3000er, 4000er, 8000er	Sharp MZ700	Texas Instruments TI 99/4A	Sinclair ZX81	Sinclair Spectrum	Laser Colour Genie	Laser 210, 310, VZ 200	Tandy TRS-80 Level II
Bildschirm löschen	Call — 936 oder Home	Print CHR$ (125)	Print CHR$ (147)	Print CHR$ (198)	Call Clear	CLS	CLS	CLS	CLS	CLS
Umschaltung auf inverse Darstellung	Inverse	Atari-Taste	Print CHR$(18)	—	—	—	Invers 1	—	—	—
* Umschaltung auf normale Darstellung	Normal	Atari-Taste	Print CHR$ (146)	—	—	—	Invers 0	—	—	—
Cursor nach rechts bewegen	PRINT CHR$(21)	Print "Esc Control —"	Print CHR$(29)	Print CHR$ (195)	—	—	—	Print CHR$(25)	Print CHR$(9)	Print CHR$(25)
Cursor nach links bewegen	Print CHR$(8)	Print "Esc Control —"	Print CHR$ (157)	Print CHR$ (196)	—	—	—	Print CHR$(24)	Print CHR$(8)	Print CHR$(24)
Cursor nach oben bewegen	Call-998	Print "Esc Control ↑"	Print CHR$ (145)	Print CHR$ (194)	—	—	—	Print CHR$(27)	Print CHR$(27)	Print CHR$(27)
Cursor nach unten bewegen	Print CHR$(10)	Print "Esc Control ↓"	Print CHR$(17)	Print CHR$ (193)	—	—	—	Print CHR$(26)	Print CHR$(10)	Print CHR$(26)
Text blinkend darstellen	Flash	siehe Listing 1	siehe Listing 1	siehe Listing 1	—	—	Flash	siehe Listing 1	siehe Listing 1	siehe Listing 1
Ausgabe von Variablen	Print	Print	Print	Print	Print	Print	Print	Print	Print	Print
Ausgabe von Variablen an einer bestimmten Bildschirmposition	HTABx, VTABy, Print	Position (Spalte, Zeile), Print	Print "Cursorsteuercode;" Variable	Cursor (Spalte, Zeile); Print	Display At (x,y); Variable ((nur Ex-Basic))	Print at (X, Y); Variable	Print at (x,y); Variable	Print @	Print @	Print @
Ausgabe von Variablen an einer bestimmten Zeilenposition	Print TAB	—	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB	Print TAB
Formatierte Textausgabe	siehe Listing 2	—	siehe Listing 2	siehe Listing 2	Print Using ((nur Ex-Basic))	—	—	Print Using	Print Using	Print Using
Eingabe von Variablen während des Programmablaufs	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input
Tastaturabfrage (im Programm)	Get	XIO 7	Get	Get	Call Key	Inkey	Inkey	Inkey	Inkey	Inkey
Den ASCII-Wert eines Zeichens ermitteln	ASC(A$)	ASC(A$)	ASC(A$)	ASC(A$)	ASC(A$)	Code (A$)	Code (A$)	ASC(A$)	ASC(A$)	ASC(A$)
Ab welchem Punkt a ist B$ in der Textvariablen C$ enthalten	—	—	—	—	Pos (C$,B$,a)	—	—	—	—	Instr (C$,B$,a)
Länge einer Zeichenkette ermitteln	Len (A$)	Len (A$)	Len (A$)	Len (A$)	Len (A$)	Len (A$)	Len (A$)	Len (A$)	Len (A$)	Len (A$)
Eine Zahl in einen String umwandeln	Str$(A)	Str$(A)	Str$(A)	Str$(A)	Str$(A)	Str$(A)	Str$(A)	Str$(A)	Str$(A)	Str$(A)
Einen String in eine Zahl umwandeln	Val(A$)	Val(A$)	Val(A$)	Val (A$)	Val(A$)	Val(A$)	Val(A$)	Val(A$)	Val(A$)	Val(A$)
Aus A$ die ersten "B"-Zeichen von links entnehmen	Left $(A$,B)	Print * A$(x, y)	Left $(A$,B)	Left $(A$,B)	Seg $(A$,1,B)	Teilstringabfrage über A$(B to C)		Left $(A$,B)	Left $(A$,B)	Left $(A$,B)
Aus A$ einen Teil-String vom "B'ten" Zeichen bis zum "C'ten" Zeichen entnehmen	Mid $(A$,B,C)	Print * A$(x, y)	Mid $(A$,B,C)	Mid $(A$,B,C)	Seg $(A$,B,C)			Mid $(A$,B,C)	Mid $(A$,B,C)	Mid $(A$,B,C)
Aus A$ die letzten "B"-Zeichen von rechts entnehmen	Right $(A$,B)	Print * A$(x, y)	Right $(A$,B)	Right $(A$,B)	Seg(A$, Len(A$-B,B)			Right $(A$,B)	Right $(A$,B)	Right $(A$,B)
Erzeugt "B"-lange Zeichenkette mit dem ersten Zeichen aus A$	siehe Listing 5	String $(B,A$)	siehe Listing 5	siehe Listing 5	RPT $(A$,B)	siehe Listing 5	siehe Listing 5	String $(B,A$)	siehe Listing 5	String $(B,A$)
Erzeugt "B"-lange Zeichenkette mit Leerzeichen	"	siehe Listing 5	"	SPC(B)	RPT $(" ",B)	"	"	String $(B, " ")	"	Space $(B)
In einer Zeile werden "B"-Zeichen übersprungen	SPC(P)	String $(B," ")	SPC(B)	—	—	—	—	—	—	—
Löschen aller Variablen	CLR	CLR	CLR	CLR	—	Clear	Clear	Clear	Clear	Clear
Inhalt zweier Variablen tauschen	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	siehe Listing 6	Swap (A,B)	siehe Listing 6	Swap (A,B)

* Mit x wird der Anfangswert und mit y der Endwert angegeben, der den Teilstring definiert.

Übersicht der Basic-Dialekte

Bernd Plumhoff

Tongenerator

Wenn sich ein Programm dem Benutzer bemerkbar machen soll, kann dieses auch auf dem akustischen Weg geschehen.

Eine Tonmeldung bietet sich an z. B. bei einer falschen Eingabe, nach einer längeren Rechnung, oder als Effekt bei Spielen.

Bei dem TRS 80 kann man Töne erzeugen, indem man die Spannung am AUX-Stecker steuert.

Das Maschinenprogramm gibt auf den Ausgabeport 255 so schnell zwei verschiedene Werte des Akkumulators aus, daß eine Schwingung simuliert wird. Die Warteschleifen zwischen den Ausgaben von A beeinflussen die Anzahl der Schwingungen pro Sekunde, also die Frequenz des Tones. Je länger die Warteschleifen sind, desto tiefer ist der Ton. Und je häufiger man die Schwingungen wiederholt, desto länger wird der Ton gehalten.

Wenn man über einen Macroassembler mit einigen arithmetischen Erweiterungen verfügt, kann man das Maschinenprogramm unverändert benutzen. Sonst lasse man die Zeilen 140 bis 190 weg und ersetze die Macroaufrufe von Zeile 450 bis 650 durch die angegebenen DEFB-Befehle.

Die Zahlenangaben im Programm sind alle dezimal!

Einem NOTE X-Befehl entsprechen zwei DEFB-Befehle, und zwar bestimmt der erste DEFB-Befehl die Tonhöhe und der zweite die Tonlänge.

Im BASIC-Programm ist das Maschinenprogramm in der Zeile 10 gespeichert und wird in Zeile 20 geladen. Die Tondaten sind in der Zeichenkettenvariablen A$ gespeichert. Man kann in einem BASIC-Programm mehrere Tonfolgen benutzen, indem man einfach den Wert von A$ ändert u. das Unterprogramm ab Zeile 90 aufruft. Die Tonfolge wird dann mit I=USR (0), PRINT USR (0) o.ä. über den AUX-Stecker ausgegeben.

```
10 DATA 243,6,21,221,33,40,125,80,221,102,0,221,110,1,62,2,93
        ,68,61,211,255,16,254,68,60,211,255,16,254,29,32,241
        ,221,35,221,35,66,16,224,201
20 FOR I=32000 TO 32039: READ J: POKE I,J: NEXT I
30 A$="AZBZCZDZEZFZGZHZIZJZKZLZMZNZOZPZQZRZSZTZUZVZWZXZYZZZ"
40 GOSUB 90
50 DEFUSR=32000  'ODER POKE 16526,0 UND POKE 16527,125
60 I=USR(0)
70 END
80 REM DAS FOLGENDE UNTERPROGRAMM INITIALISIERT DEN
       WERT VON A$ ALS DATEN FUER DIE TONROUTINE
90 POKE 32002,LEN(A$)/2
100 I=VARPTR(A$)
110 POKE 32005,PEEK(I+1)
120 POKE 32006,PEEK(I+2)
130 RETURN
```

```
DATEN   DEFB   76        DEFB   202       DEFB   157
        DEFB   247       DEFB   107       DEFB   144
        DEFB   80        DEFB   193       DEFB   148
        DEFB   238       DEFB   114       DEFB   153
        DEFB   85        DEFB   184       DEFB   139
        DEFB   229       DEFB   121       DEFB   162
        DEFB   90        DEFB   175       DEFB   130
        DEFB   220       DEFB   129       DEFB   172
        DEFB   95        DEFB   166       DEFB   121
        DEFB   211       DEFB   136       DEFB   183
        DEFB   102                        DEFB   112
                                          DEFB   194
                                          DEFB   103
                                          DEFB   206
                                          DEFB   94
                                          DEFB   219
                                          DEFB   85
                                          DEFB   232
                                          DEFB   76
                                  DEND    DEFB   246
                                          DEFB   67
```



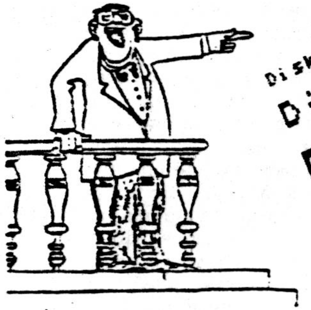
```
00100 ; ##################################################
00110 ; ##                 TONGENERATOR                 ##
00120 ; ##################################################
00130 ;
00140 NOTE     MACRO    #PAR1
00150 TON1     DEFL     5*#PAR1*#PAR1*#PAR1
00160 TON2     DEFL     8*#PAR1*#PAR1+447*#PAR1+7145
00170          DEFB     TON2/100+TON1/1000
00180          DEFB     -9*#PAR1
00190          ENDM
00200          ORG      32000
00210 START    DI                 ; INTERRUPTS STOEREN DIE TOENE!
00220 ZAHL     DEFL     DEND-DATEN
00230          LD       B,ZAHL/2+1        ; ANZAHL DER TOENE
00240          LD       IX,DATEN
00250 WEITER   LD       D,B
00260          LD       H,(IX)   ; TONHOEHE
00270          LD       L,(IX+1); TONLAENGE
00280          LD       A,2
00290          LD       E,L
00300 WIEDER   LD       B,H
00310          DEC      A
00320          OUT      (255),A
00330 WARTE1   DJNZ     WARTE1
00340          LD       B,H
00350          INC      A
00360          OUT      (255),A
00370 WARTE2   DJNZ     WARTE2
00380          DEC      E
00390          JR       NZ,WIEDER         ; TON HALTEN
00400          INC      IX
00410          INC      IX
00420          LD       B,D
00430          DJNZ     WEITER
00440          RET
00450 DATEN    NOTE     1       | 00560           NOTE     12
00460          NOTE     2       | 00570           NOTE     13
00470          NOTE     3       | 00580           NOTE     14
00480          NOTE     4       | 00590           NOTE     15
00490          NOTE     5       | 00600           NOTE     16
00500          NOTE     6       | 00610           NOTE     17
00510          NOTE     7       | 00620           NOTE     18
00520          NOTE     8       | 00630           NOTE     19
00530          NOTE     9       | 00640           NOTE     20
00540          NOTE     10      | 00650 DEND      NOTE     21
00550          NOTE     11      | 00670           END      START
```

„Das Schönste am Berühmtsein ist: Wenn man Leute langweilt, glauben sie, es ist ihre Schuld."

Henry Kissinger, ehemaliger amerikanischer Außenminister (geboren 1923)

„Das absolut Negative an den schnellen Flugzeugen ist, daß es jetzt keine entfernten Verwandten mehr gibt."

Autor unbekannt



Diskussion
Diskussion
Diskussion
Diskussion

Schreibt Eure Meinung zu diesen Themen !!

INHALTSVERZEICHNIS VON ZEITSCHRIFTEN
====================================

Sicher hat jeder schon mal Informationen zu einem gewissen Thema über seinen Rechner (ich z.B. suche z.Zt. Informationen über die Programmiersprachen 'C' und 'LISP' für Modell I) oder ganz allgemein zu einem Thema gesucht.
Da nun innerhalb des Clubs bei den einzelnen Mitgliedern sicherlich viele verschiedene Zeitschriften vorhanden sind, möchte ich folgende Anregung machen:
Man könnte doch die Inhaltsverzeichnisse von Zeitschriften austauschen, und zwar entweder als Kopie oder über ein geeignetes Dateiprogramm.
Ich habe z.B. die Inhaltsverzeichnisse von verschiedenen (z.T. unvollständigen) Jahrgängen der Zeitschriften MC, ELEKTOR, 80 MICRO und 80 US als UNIDAT80-Datei.
Der Austausch über Kopien ist zunächst mal der einfachere Weg, hat allerdings ab einem gewissen Umfang dann bei der Suche nach Artikeln Nachteile (man muß viel suchen, übersieht leicht etwas).
Beim Austausch über ein Dateiprogramm ergibt sich das Problem der Auswahl eines geeigneten Dateiprogramms (UNIDAT80 ist z.B. nur für SD, SUPER ist bei der Tastaturabfrage recht langsam und hat eine feste Satzlänge, selbstgeschriebene Programme erfordern einen ziemlich hohen Programmieraufwand, da sehr schnell einige tausend Datensätze zusammenkommen (bei mir z.Zt. über 2000) und das Programm dadurch ziemlich schnell sein muß, auch muß es wirklich ausgereift sein, um nicht nach tagelanger Dateneingabe feststellen zu müssen, daß das Programm dieses oder jenes nicht leistet), während die Eingabe eines Inhaltsverzeichnisses bei einem guten Programm recht schnell erledigt ist (je nach Umfang 1-2 Tage).

Wer Interesse oder Anregungen zu diesem Vorschlag hat, soll sich doch bei mir melden oder über das nächste Info seine Meinung darlegen.
Josef Konrad

Kurzerklärungen für Programme

Sicher habt Ihr das Problem auch schon gehabt : Mitten in der Anwendung eines Programmes wisst Ihr den entsprechenden Befehl nicht. Man muß in der Anleitung nachsehen. Aber wo steht's jetzt?

Bis nun das Richtige gefunden ist gibt es eine ganz schöne Sucherei, Blätterei und Leserei. Und diese zusätzliche unnütze Nebenbeschäftigung kostet uns Zeit. Zeit, die uns vom reinen Vergnügen der Computerei abhält.

Viele Programme haben umfangreiche Erklärungen über ihre Funktion oder ihre Bedienung. Wäre es nicht gut eine einseitige, maximal zweiseitige Programmanleitung zu erstellen. Ein Blick genügt, und man findet das Gewünschte auf Anhieb. Gleichzeitig könnte von den meist englischen Anleitungen eine Deutschversion geschrieben werden. Ein weiterer nützlicher Nebeneffekt besteht darin, daß auf diesen Eigenproduktionen von Anleitungen kein Copyright mehr besteht. Und somit von uns -im Club untereinander - getauscht oder vervielfältigt werden kann. Es ist auch günstiger und weniger zeitaufwendig eine Kurzversion zu haben, als hinzustehen und sich ein mehrseitiges Werk zu vervielfältigen. Ganz davon zu schweigen, daß dies sowieso verboten ist.

In unserer Clubgemeinschaft haben sicher einige von Euch solche Kurzerklärungen, die dem einen oder anderen die Bedienung bzw. Anwendung eines Programmes erleichtern hilft. Ich glaube, daß wir in dieser Beziehung zusammenarbeiten sollten. Bevor sich jeder selbst eine solche Hilfe erstellt, -die einer von Euch vielleicht schon hat-.

Nun mein Vorschlag, man könnte die Kurzanleitungen für die Clubmitglieder mit in die Programmbibliotek als Textfile aufnehmen, und/oder auf karton gedruckt mit dem INFO für die Interessenten mitschicken. Ich würde mich bei eventuellem Interesse bereiterklären, die Anfertigung der kartonierten Hilfen, von Euren Textfile, zu übernehmen.

Was meint Ihr dazu

J. Kreuder

Shift-TRICK

Programme, um den Kleinschriftmodus bei den Genie-Computern Genie I, Genie II und Colour Genie umzudrehen, gibt es ja eigentlich genug. Warum noch mal das gleiche Thema? Erstens, um einmal zu verdeutlichen, warum und wie es eigentlich kommt, daß die Genie-Computer in der Grundversion auf einfachen Tastendruck groß, bei Shift und Taste erst kleinschreiben und zweitens, um einige Hilfen zu geben, dieses leidige Softwareproblem anzugehen.

Der Leser hat richtig verstanden. Es handelt sich nicht etwa um ein Problem der Hardware (das könnte man ja noch verstehen!), sondern um die Software im ROM (dem Betriebssystem des Computers). Nimmt man sich nämlich ein ROM-Listing zur Hand, in dem das Betriebssystem ausführlich aufgelistet und erklärt ist, so können sich einem manchmal die Haare sträuben, wie gewisse Betriebsroutinen angelegt sind.

Es scheint, daß einige Softwarelösungen mit Absicht so geschrieben wurden, daß dem sehr ernsten Anwender (z. B. Textverarbeiter) später nichts anderes übrig bleibt, als auf die Erweiterungen für den Computer zurückzugreifen, um kleine, aber manchmal doch sehr störende Eigenheiten des Systems wieder auszubügeln. Dies gilt auch in gewissem Maße für die Hardware eines Computersystems, wie beispielsweise mein Artikel in der GENIE DATA Nr. 4 zeigt. Mit einem wirklich minimalen Aufwand wird hier der Einbau von vier weiteren Tasten in die Tastatur beschrieben, so daß sich die deutschen Umlaute direkt erreichen lassen.

Doch nun zurück zu unserem eigentlichen Thema. Für die, die über kein ROM-Listing von einem der drei Genie-Computer verfügen, möchte ich grob auf die Tastaturabfrage bei den oben genannten Systemen eingehen. Das Prinzip ist für alle drei Systeme gleich. Dies ist nicht verwunderlich, da der Genie I und der Genie II direkte Nachfahren (Kopien) des guten alten TRS 80 Computers sind. Dieser Computer hat auch in Deutschland ein Stück Microcomputer-Geschichte mitgeschrieben und zählt eigentlich längst zu den Veteranen der Microcomputer. Dank seines guten Konzepts (vor allem seiner Software) erlebte er in den Genie-Computern eine Renaissance, die sein Betriebssystem fast unverändert übernahmen und lediglich eine bessere Hardware darumbauten. Leider wurden allerdings auch viele Schwächen des Betriebssystems mit übernommen.

Eine Weiterentwicklung des Betriebssystemes stellt das ROM des Colour Genie dar. Bei Vergleich der ROM's des Colour Genie und des Genie I oder II ergibt sich, daß das Colour Genie fast die gleichen Adressen der einzelnen Routinen benutzt wie seine „Brüder". Besonders auffallend ist dies bei den Befehlen Set und Reset. Sie befinden sich bei allen drei Systemen im ROM an fast der gleichen Stelle. Während bei Genie I und II der Befehl für das Setzen, bzw. Rücksetzen, eines Grafikpunktes verantwortlich ist, bewirken sie beim Colour Genie das Setzen, bzw. Rücksetzen eines Bits in einer bestimmten Adresse.

Auch das Prinzip und der Aufbau der Tastaturabfrage ist bei allen drei Systemen fast identisch. Der jeweilige Computer trennt die Tasten nach Buchstaben, Ziffern und Sonderzeichen und nach Kontrolltasten. In jeweils gesonderten Teilprogrammen wird der zugehörige ASCII-Code errechnet und einem Register der CPU (Zentraleinheit = Mikroprozessor) zur weiteren Verarbeitung übergeben. So steht der ASCII-Code der jeweiligen Taste am Ende im sog. Akku (Register A der CPU).

Bei der Behandlung von Buchstaben wird es jetzt endlich für unser Ausgangsproblem interessant.

Hier erfolgt nämlich eine Abfrage, ob zu dem gedrückten Buchstaben auch die SHIFT-Taste betätigt wurde. Und siehe da, bei gedrückter SHIFT-Taste wird der sog. „Lower Case" erzeugt, was nichts anderes bedeutet, als daß der ASCII-Code eines Kleinbuchstaben erzeugt wird. Bei nicht betätigter SHIFT-Taste hingegen wird Großschrift erzeugt.

Hier fragt man sich zurecht, warum dies so sein muß. Kann man denn diese Routine nicht so anlegen, daß bei SHIFT Großschrift erzeugt wird?

Man kann! Nämlich durch die Abänderung eines Maschinensprachebefehls. Hierdurch wird die oben erklärte Abfrage umgedreht, sodaß der „Lower Case" jetzt ohne SHIFT-Taste erzeugt wird.

Aber wie kann man ein doch unveränderliches ROM abändern?

Hier sind die Besitzer eines EPROM-Programmierers gut dran. Denn EPROM's (löschbare Festspeicher) lassen sich nach belieben ändern und anstelle der ROM's in den Genie Computer einsetzen. Der EPROM-Typ für alle drei behandelten Computer ist ein 2532-EPROM. Es handelt sich um einen 4 KByte Speicher, von dem sich im Colour Genie vier, im Genie I und II jeweils drei befinden.

Für die Programmierung geht man am besten folgendermaßen vor:

Die entsprechende Stelle, die geändert werden muß, befindet sich bei allen drei Genies im ersten ROM, d. h., die Abfrage wurde einfach umgekehrt. Kleine Veränderung - große Wirkung!

Am besten schreibt man zu Beginn der Prozedur in einen freien Speicherplatz des RAM (Benutzerspeicher) den Wert 38H bzw. 56 dez. mittels Poke-Befehl. Günstig ist es dabei, wenn diese Speicherstelle vorher geschützt ist, damit dieser Wert nicht durch irgendwelche Umstände verlorengeht. Dann wird das ROM 1 bis zu der Adresse in ein EPROM 2532 kopiert, die vor dem zu ändernden Byte steht. Die nächste EPROM-Adresse wird mit dem Wert 38H aus der RAM-Speicherzelle programmiert. Der Rest des EPROM's wird wieder mit dem entsprechenden Teil aus dem ursprünglichen ROM programmiert, sodaß anschliessend im EPROM eine genaue Kopie der ersten 4 KByte des ROM's steht, mit nerlei Einfluß, da das Betriebssystem der Genie-Computer intern alle Kleinbuchstaben, die nicht in Anführungszeichen stehen, in Großbuchstaben umwandelt. Um dies zu Veranschaulichen, geben Sie im Direktmodus folgendes ein:

>1Ø goto 1ØØ

Wichtig! Das „goto" muß klein geschrieben sein.

Nun listen Sie das Miniprogramm! Es erscheint nun:

1Ø GOTO 1ØØ

Wie man sieht, ist das „goto" jetzt groß geschrieben. Colour-Genie-Besitzer müssen sich allerdings daran gewöhnen, daß auch die Funktionstasten, die auch als Buchstaben behandelt werden, umgedreht sind.

Übrigens finden sich im Betriebssystem des Colour-Genie an einigen Stellen Freiräume, in die eventuell kleine eigene Routinen Platz

Hardware — Arbeitskreis

ECB — BUS System

Zwischenbericht:
Im letzten Clubinfo berichtete der Initiator der Hardwaregruppe Walter Zwickel über das geplante ECB - Bussystem. Ich bin heute so frei, in Stichpunkten über die Fortschritte der Entwicklung zu berichten.

1. Die Schaltung der Basis-Platine (Entwicklung: Walter Zwickel) ist fertig und in der praktischen Erprobung! Die Schaltunterlagen wurden an alle Mitglieder der Hardwaregruppe verschickt!
2. Die Erstellung des Platinenlayouts für die Basis-Platine hat der Manfred Held übernommen. Wie er berichtet, geht es damit gut voran!
3. Das Netzteil für die ECB-Buserweiterung erstellt der Ulrich Böckling. Auch hier sind beachtliche Fortschritte zu vermelden!
4. Es ist zu erwarten, daß sowohl die Platine für die Basis-Schaltung, als auch das Netzteil bis zum Erscheinen des nächsten Clubinfos fertig sind. In diesem Info wird dann die ganze Sache eingehend vorgestellt und besprochen!

übrigens, bei der Hardwaregruppe des CLUB 80 kann jedes Mitglied mitmachen, egal wieviel Hardwareerfahrung es mitbringt!!! Die Kontaktperson ist Walter Zwickel.

Hartmut Obermann

Syncronisierte Taktumschaltung für VideoGenie

Liebe Clubfreunde,

Im Clubinfo Nr. 5 hat Gerald Dreyer eine Schaltung veröffentlicht, mit der man die Taktfrequenz des VideoGenie verdoppeln kann. Nachteil dieser Schaltung ist die Tatsache, daß bei der Umschaltung zwischen einfacher und doppelter Taktfrequenz in den meisten Fällen der Rechner hängenbleibt. Außerdem kann man die Taktfrequenz nur mit einem Schalter, also hardwaremäßig, jedoch nicht softwaremäßig umschalten. Um diese Nachteile zu beseitigen, hat Manfred Held die unten abgebildete Schaltung entworfen.

Die eigentliche Umschaltung der Taktfrequenz bewerkstelligen zwei NAND-Gatter (x2, rechts neben Z38). Die Umschaltung kann durch einen Schalter (linke obere Ecke) oder über den Port FEh (254d) (Decodierlogik linke untere Ecke) erfolgen. Um einen Systemabsturz wärend der Umschaltung zu verhindern, wird durch Z14 sichergestellt, daß der Umschaltvorgang erst erfolgt, wenn die CPU weder Schreib- noch Leseoperationen durchführt.

Durch die erweiterten Möglichkeiten der Schaltung ist natürlich auch der Bauteileaufwand gestiegen. Es werden folgende Teile benötigt:

IC x1 = 74LS00 = vierfach NAND-Gatter mit je 2 Eingängen
IC x2 = 74LS03 = vierfach NAND-Gatter mit offenem Kollektor-Ausgang
IC x3 = 74LS04 = sechsfach Inverter
IC x4 = 74LS30 = NAND-Gatter mit 8 Eingängen
IC x5 = 74LS74 = zweifach Speicher-Flip-Flop, positiv flankengetriggert
Alle IC's je 1 mal. Weiterhin:
2* 1N4148 = Siliziumdiode
3* 4700 Ω Widerstand
1* 10 kΩ Widerstand
1* 10 µF/16V Kondensator

Viel Spaß beim Basteln, wünscht euch ~~Hartmut Obermann~~

Eine Operateuse aus Wien
dacht' beim Arbeiten stets nur an ‚Ihn':
Ihr Computer, voll Frust,
verlor deshalb die Lust –
er war auch nur ein Mensch, wie es schien.

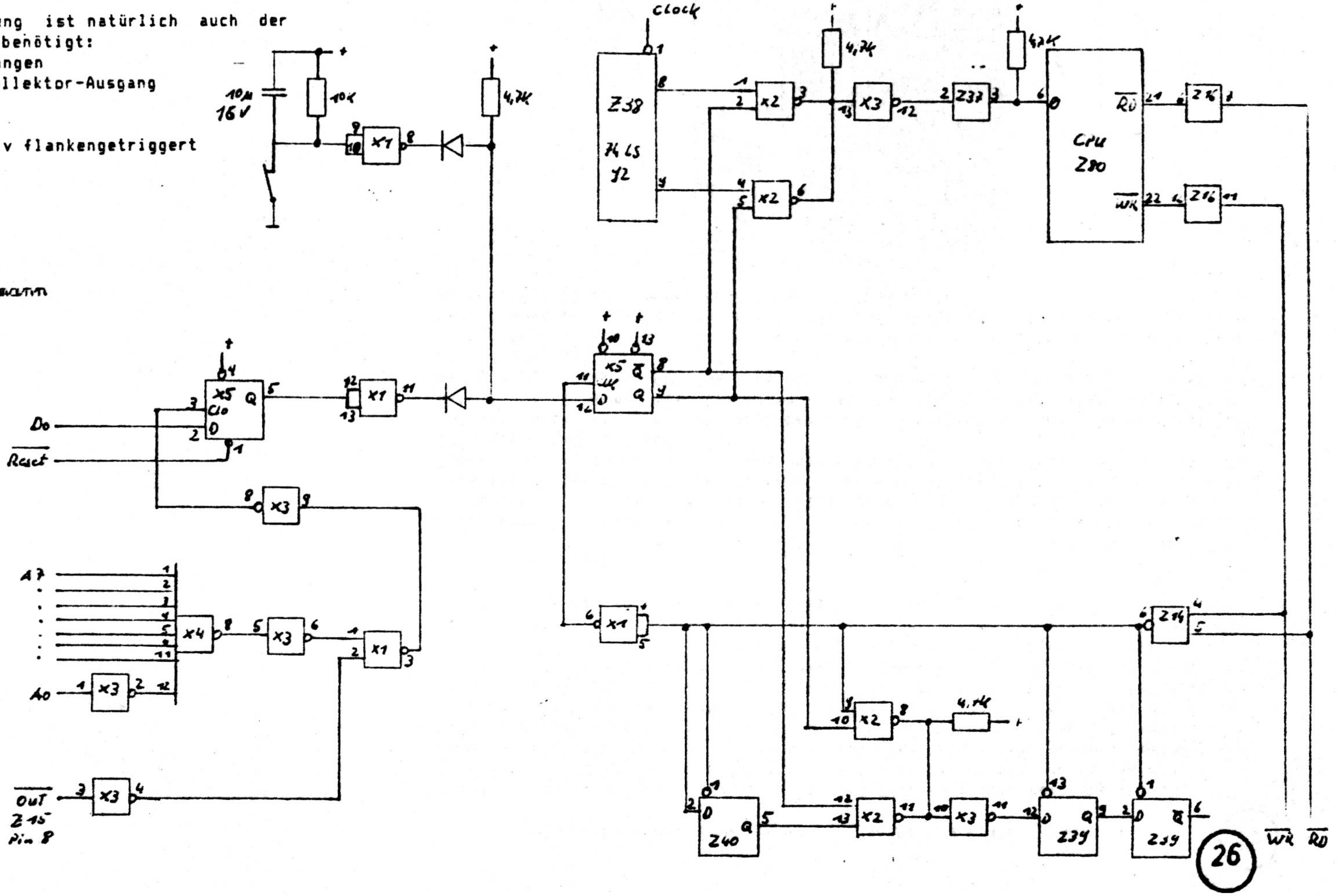
- Bei Genie I, II auf Adresse 0418H
- Beim Colour Genie auf Adresse 044CH

Auf dieser Adresse befindet sich der Wert 30H bzw. 48 dez. Er verzweigt das Programm, wenn keine SHIFT-Taste gedrückt ist. Dieser Wert, übrigens auf allen drei Systemen gleich, muß in 38H bzw. 56 dez. umgewandelt werden. Jetzt verzweigt das Programm bei gedrückter SHIFT-Taste, d. Ausnahme der einen Adresse.

Das EPROM sollte erst dann für das jeweilige ROM in das Gerät eingesetzt werden, wenn alle Adresseninhalte überprüft worden sind. Die meisten EPROM-Programmierer gestatten dies. Nach dem Einbau kann wie bei einer normalen Schreibmaschine mit der SHIFT-Taste groß- ansonsten klein geschrieben werden. Auf die Erkennung von Basic-Befehlen hat diese Umkehrung keifinden könnten. Weiterhin könnte man bestimmte Adressen des Betriebssystemes abändern, um bestimmte Effekte nach dem Einschalten des Gerätes zur Verfügung zu haben.

Sie sehen, ein Blick in ein ROM-Listing lohnt sich.

Ich hoffe, ich konnte einige Anregungen bieten, und zeigen, daß auch in der Grundversion eines Computers mit oftmals geringen Mitteln einiges erreicht werden kann.



Die Software schmilzt

Beim Begießen der Blumen im Raum mit der Datenverarbeitungsanlage hat es einige Wasserflecken im Teppichboden gegeben. Es ist Hochsommer und sehr warm. Eine neue Mitarbeiterin – mit den Feinheiten der EDV wenig vertraut – betritt den Raum, sieht die Flecken im Teppichboden, erbleicht und schreit: »Die Software schmilzt!«

Thema: SUPERTAPE

Das im Info Nr. 7 (Seite 44-50) vorgestellte Bandaufzeichnungsverfahren SUPERTAPE ist sicher für einige Mitglieder des CLUB 80 so interessant, daß sie sich entschlossen haben, das Programm abzutippen und die Platine nachzubauen. Dazu zwei Tips:

1. Das Programm wird vom Heise-Verlag, der auch die Zeitschrift c't herausgibt für 7.-- DM auf Kassette verschickt! Man kann sich also die Mühe des Eintippens und Fehlersuchens sparen. Wer zudem noch etwas Zeit hat, kann ja warten bis das Programm in der Clubbibliothek auftaucht (was sicher nicht allzu lange dauert).
2. Die Platine für die benötigte Zusatzschaltung wird für unter 20,-- DM von verschiedenen Inserenten in der c't angeboten. Die Schaltung kann also auch von den in der Hardware nicht so erfahrenen Clubmitgliedern aufgebaut werden.

Einige Diskuser werden sich natürlich fragen, was man mit einem Tapeaufzeichnungsverfahren, sei es auch noch so schnell, soll. Die Antwort ist relativ einfach. Mit SUPERTAPE erhält man die Möglichkeit, Programme und Daten mit anderen (supertapefähigen) Computern auszutauschen! Weiterhin kann man sich damit vielleicht manche Backupdiskette sparen.

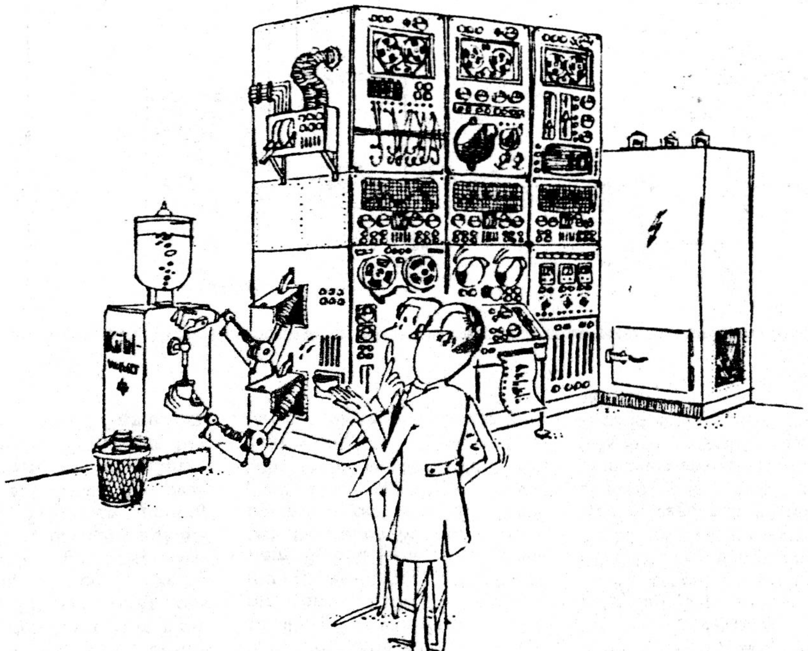
Ein sicheres Plätzchen (für TRS80)!

Im letzten Clubinfo wurde eine Schaltung veröffentlicht, mit der sich die Besitzer von VideoGenie-Computern ein sicheres Plätzchen für Maschinenroutinen in ihrer Maschine sichern können. Die Schaltung ist für TRS80-User nur mit erheblichen Änderungen brauchbar und somit mit Vorsicht zu genießen. Wer sich trotzdem in seinen TRS80 etwas mehr RAM einbauen möchte, der möge noch etwas Geduld haben. Ich bin z.Z dabei eine Schaltung aus einer älteren amerikanischen Zeitschrift (80 US) auszutesten, die speziell für TRS80-Computer 1920 Bytes zusätzlichen Speicher bringt. Die Schaltung und der zugehörige Artikel werden dann im nächsten Info veröffentlicht (wenn alles funktioniert!).

Hartmut Obermann

Hardwaretip : CASICO fx 115

Nicht wenige Computerfreunde schreckt die Erlernung der Maschinensprache bzw. Assembler deshalb ab, weil man dabei sehr viel mit verschiedenen Zahlensystemen umgehen muß. Die dauernde Umrechnerei zwischen dezimaler, dualer, oktaler und hexadezimaler Darstellung von Zahlen macht die Arbeit mit dem Assembler oder Monitor nicht gerade leichter. Und selbst, wenn man garnicht nicht allzu tief in die Maschine einsteigen will, bekommt man schon Probleme. Ein Beispiel: in einem BASIC-Programm wird eine Adresse angepoket (die Adresse ist naürlich dezimal angegeben). Um schnell festzustellen, was dieses POKE eigentlich macht, schaut man am besten ins röckrath'sche ROM-Listing. Hier sind aber alle Adressen hexadezimal angegeben (wie sich das gehört) und schon geht die Umrechnerei los!
BASIC-Umrechungsprogramme bieten kaum einen Ausweg aus dem Dilemma. Wer verläßt z.B. gern den Assebler und lädt ein BASIC-Programm, nur um eine Zahl umzurechnen? Eine wirkliche Lösung bieten eigentlich nur Taschenrechner mit integrierter Umrechungsfunktion. Diese aber waren bis jetzt relativ teuer. Inzwischen jedoch hat die Firma CASICO mehrere technisch wissenschaftliche Taschenrechner zu einem durchaus erschwinglichen Preis herausgebracht (50 - 70 DM), die diese Funktion besitzen.
So hat mich mein kürzlich erworbener CASIO fx 115 nur 59,- DM gekostet. Er beherrscht nicht nur die Umrechung aller benötigten Zahlensysteme sondern auch die Logikoperationen AND, OR und XOR sowie die Darstellung dualer Zahlen im sog. Zweierkomplement!
Meiner Meinung nach erleichtert ein solches Hilfsmittel den Einstieg in die Maschinensprache erheblich und hilft dem erfahrenen Assemblerprogrammierer erheblich Zeit zu sparen. Euer *Hartmut Obermann*



»Donnerwett[illegible] Das ist ja beinahe menschlich, nicht?«

EISENBAHN S 0126 A 00 /BAS --

Maessiges Geschicklichkeitsspiel mit ebenso maessiger Grafik.

VIERGEWINNT D 0141 A 01 /BAS --

Das Programm stellt das benoetigte Spielfeld zur Verfuegung und verwal tet die Einwuerfe der Spieler, spielt jedoch nicht selbst!

VERLIES D 0142 A 01 /BAS --

Adventure, bei dem man durch ein Verlies mit 10 Stockwerken irrt und dabei gegen Monster kaempfen muss.

FUNKTIONSPLOTTER M 0143 A 02 /BAS --

Dieses Programm plottet Funktionen hochaufloesend (400 oder 800 Punkte) auf EPSON MX/RX/FX - Drucker. Die Aufloesung ist sehr gut. Quelle: ELCOMP

ROBOTER D 0150 A 01 /BAS --

Ein mittelmaessiges Denk-/Geschicklichkeitsspiel mit maessiger Grafik.

BUNDESLIGATABELLE A 0158 A 01 /BAS --

Bundesligatabelle fuer den Hausgebrauch in BASICCODE 2 geschrieben. Die Daten werden auf Kassette gespeichert (kann umgeschrieben werden).

PFLANZEN S 0159 A 01 /BAS --

Dieses BASICCODE 2-Programm stellt eine grosse Menge Wissen ueber viele Garten -Blumen und -Gemuese zur Verfuegung.

NIMMSPIEL D 0161 A 02 /BAS --

Ein sehr bekanntes Streichholzspielchen wird hier gegen den Computer gespielt. Er gewinnt (fast) immer, probiert's mal aus.

PROTON S 0162 B 01 /CMD --

Dieses Programm realisiert eine Protokollfunktion fuer TRSDOS (aehnlich ROUTE,DO,DO,PR) im NEWDOS80.

TASCHENRECHNER M 0268 A 01 /BAS --

Ein Taschenrechner als BASIC-Einzeiler. Quelle: Computer Persoenlich

---

BOND A 0269 A 03 /BAS --

Aktienverwaltungs-Programm amerikanischer Machart. Quelle: Micro 80

DIGITALKURS S 0270 A 03 /BAS --

Eine Einfuehrung in die Digitaltechnik in englicher Sprache. Sehr gut gemacht und vor allem fuer den Anfaenger interessant.

MCEDIT S 0271 A 03 /BAS --

Ein besonders komfortabler Texteditor aus der MC 8/83

Funktionssucher M 0272 A 03 /BAS --

Ein Programm, dass nach der Eingabe von Wertepaaren (z.B. aus einer Messreihe) die mathematische Funktion ermittelt, die diese Wertepaare unterliegen.

Wordchecker T 0273 A 03 /BAS --

Dieses Programm dient zur Fehlersuche in ASCII-Texten. Quelle: Micro 80 8/84

Adressen-Text A 0274 A 50 /COB --

Ein sehr komfortables Adressenverwaltungsprogramm, in COBOL geschrieben. Das System enthaelt u.a. eine sehr gute Etikettendruckroutine. Autor: Harald Trapp Umfang: 3 Disketten.

H-DOS S 0275 B 99 /xxx --

Ein von Arnulf Sopp sehr stark veraendertes und verbessertes G-DOS. Features: Grafik-Hardcopy, Bildschirmsicherung, akustische Fehlersignale, Laufwerktest, DiskDatenEditor, CLS fuer HRG, uvm.!!!

CLUB 80 - Programmbibliothek exklusiv

Die beiden wohl wichtigsten und besten Programme bzw. Programmpakete, die sich in der Bibliothek des CLUB 80 befinden, sind das Adressenverwaltungs-programm ADRESSENTEXT und das Diskettenbetriebssystem H=DOS. Beide Programme verdienen es, hier nocheinmal speziell erläutert und angepriesen zu werden.

1. Das H=DOS von Arnulf Sopp ist eine (erhebliche) Erweiterung des G-DOS der Firma TCS (Trommelschläger Computer Service). Der Arnulf hat nach einer Vereinbarung mit TCS das Recht, das G-DOS zu verändern und als sein H=DOS zu veräußern. Dies jedoch nur an Leute, die sich irgendwann einmal rechtmäßig ein G-DOS beschafft haben.
H=DOS besitzt gegenüber dem ursprünglichen G-DOS erhebliche Erweiterungen und damit Vorteile! Hier ein paar davon:

I. Die JKL-Funktion ist so geändert, daß Grafiken positiv oder negativ ausgedruckt werden können. Dabei wird gleichzeitig die Verzerrung, die normalerweise bei Bildschirmhardcopys auftritt, ausgeschaltet. Weiterhin ist JKL nun auch als DOS-Befehl implementiert!

II. Bildschirmspeicherung mit dem Dreitastenbefehl 345 oder dem Utilityprogramm GRA/CMD.

III. Akustische Signale bei DOS- und BASIC- Fehlern!

IV. Der DOS-Befehl LWT (Laufwerktest) ist, im Gegensatz zu einigen G-DOS-Versionen, wieder implementiert.

V. Der DOS-Befehl DDE (Diskdateneditor) ist erheblich erweitert. So kann man z.B. nicht nur an Dateisektoren sondern auch an allen Sektoren einer Diskette herumdoktern.

VI. CLS 6 löscht den Bildspeicher der hochauflösenden Grafik. Das gleiche passiert bei jedem Bootvorgang.

VII. Eine der wichtigsten Neuerungen ist der DOS-Befehl ID! Er ermöglicht die automatische Ermittlung der PDRIVE-Parameter unbekannter Disketten und erspart damit viel nervenraubende Sucherei bei fremden Disketten!

VIII. Weiterhin unterstützt das H=DOS voll die Hardwareerweiterung EG 64 MBA (Bankingbaustein zur vollen Nutzung der ab Baujahr 83 in alle Genie I und II eingebauten 64k-RAM) und gibt dem Benutzer gleich sinnvolle Anwendungen der gewonnenen 16k-RAM an die Hand (Spooler, Änderungen des BASIC-Interpreters usw.).

Alle Änderungen aufzuzählen würde den Rahmen dieser kurzen Vorstellung sprengen. Ich kann nur jedem G-DOS-Besitzer raten, mir eine G-DOS-System-diskette zu schicken und sich das H=DOS selbst einmal anzuschauen. Eine Anleitung für die Erweiterungen wird übrigens mitgeliefert!

2. Das Programmpaket ADRESSENTEXT von unserem Cobol-Spezialisten Harald Trapp zeichnet sich vor allem durch die Verwendung sog. ISAM-Files zur Speicherung der Daten aus. Dadurch ermöglicht es einen sehr schnellen Zugriff auf einzelne Datensätze der Adressendatei, auch wenn diese schon eine erhebliche Größe erreicht hat. Neben den Funktionen "Anlegen, Ändern und Löschen" von Datensätzen, werden folgende Zusätze geboten:
1. Drucken der Stammdatenliste (komplett oder als Kurzliste)
2. Eingabe, Speicherung und Druck von Normetiketten
3. Druck von Etiketten aus der Adressen-Datei

Ein spezielles Programm dient der Sortierung der Stammdatei und beschleunigt somit nochmals das Auffinden einzelner Datensätze.
Das Programmpaket wird mit Anleitung geliefert. Wenn gewünscht, können auch die Cobol-Quelltexte mitgeliefert werden. Ein Cobol-Compiler ist nicht enthalten!!!

Wie ihr seht, haben wir hier wirklich zwei ausgezeichnete Programmpakete in unserer Bibliothek, die sich keines der Clubmitglieder entgehen lassen sollte. Ich möchte mich an dieser Stelle nochmals und im Namen aller Mitglieder des CLUB 80 bei den Autoren für ihre Arbeit bedanken!

Hartmut Obermann

Achtung: Die beiden Programme können unter ihren normalen Namen bestellt werden!!!

BÖRSE

WER HAT WAS -- WER SUCHT WAS --

Ich habe Interesse am EG 64 MBA. Kann es jedoch bei keiner mir bekannten Firma beziehen. Bei TCS konnte man mir nicht weiterhelfen. Wer weiß, wo man es noch beziehen kann?

JÜRGEN WUCHERER

FRAGEN, FRAGEN, FRAGEN ...

??? FRAGENKASTEN ???

Was bringt mir als Einsteiger der ECB-BUS der Hardware-Gruppe ?

WALTER PILLER

Anmerkung der Redaktion

Hallo Walter, eine Antwort dazu möchte ich den Hardware-Profis überlassen. Näheres über den ECB-BUS ist ja sicher einen Infoartikel wert.

Mein GENIE II steht nur 2 Meter vom Radio entfernt. Leider ist vom Radio nur noch ein übles Rauschen zu vernehmen, wenn gleichzeitig der Computer läuft. Wer weiß Abhilfe?

JÜRGEN WUCHERER

CLUB 80 - Programmbibliothek : neue Programme

An dieser Stelle will ich euch eine ganze Menge neuer Programme vorstellen, die seit dem Erscheinen der letzten Liste in die Bibliothek aufgenommen wurden. Es sind vor allem Anwenderprogramme und Utilities in BASIC (auch compiliert) und Maschinensprache. Die Programmbilbiothek ist vor allem um zwei Programmpakete reicher geworden, die ich am Schluß dieser Liste genauer vorstellen werde. Ich danke allen Einsendern von Programmen für ihre Mitarbeit und hoffe, daß möglichst viele Mitglieder etwas für sie interessantes in der Bibliothek entdecken werden!

ERRORS S 0057 A 01 /BAS --

ON ERROR GOTO - Routinen fuer Diskfehler - Behandlung in BASIC-Program men.

GRAFIK 3-D Z 0058 A 01 /BAS --

3-D Grafikdemo aus Micro 80 4/84.

Integer->Oktal M 0061 A 01 /CMD --

Umwandlung von Integerzahlen in Oktalzahlen.

JAHRESKALENDER S 0064 A 02 /BAS --

Das Programm druckt nach Angabe der Jahreszahl einen Kalender aus. Aus CP 6/82

OSTERN S 0065 A 01 /CMD --

Das Programm druckt das Datum fuer Ostern fuer die Jahre 1930 - 1990 aus.

Lohnberechnung M 0067 A 01 /BAS --

Das Programm errechnet Stunden-, Monats- und Jahreslohn aus verschiedenen moeglichen Angaben.

Lohntabelle M 0068 A 01 /BAS --

Das Programm erstellt eine Lohntabelle nach Eingabe der Wochenarbeitszeit unter Beruecksichtigung verschiedener Stundenloehne.

Pascal Dreieck M 0069 A 01 /CMD --

Das Programm erstellt die rechte Haelfte des pascal'schen Dreiecks mit acht Reihen.

Roemische Ziffern M 0070 A 01 /CMD --

Das Programm rechnet Integerzahlen in "roemische" Ziffern um.

Temperaturumrechnung M 0071 A 01 /CMD --

Das Programm rechnet Fahrenheit in Celsius um und umgekehrt.

Wochentag S 0074 A 01 /CMD --

Das Programm gibt nach Eingabe des Datums den entsprechenden Wochentag aus.

SPOOLER S 0075 A 01 /CMD --

Ein Druckerspooler aus der MC 6/82. Die Sourcecodefile wird mitgeliefert.

Titelgrafik S 0076 A 01 /BAS --

Dieses Programm zeigt eine Unmenge von Moeglichkeiten, wie man seinem BASIC-Programm einen interessanten Vorspann geben kann. Die einzelnen Unterprogramme koennen uebernommen und leicht geaendert werden.

BALISTIK G 0078 A 00 /BAS --

Ein mittelmaessiges Geschicklichkeitsspiel.

STRICKMUSTERGRAFIK Z 0079 A 02 /BAS --

Dieses Programm laesst nach Eingabe zweier Parameter sehr schoene Grafikmuster aus dem Bildschirm erscheinen.

GALTON S 0080 A 01 /BAS --

Demonstration eines Nagelbrettes, auf dem Kugeln von oben nach unten fallen. Die Zahl der Kuglen kann vorgegeben werden, ebenso die Wahrscheinlichkeit, dass eine Kugel an einem Nagel nach rechts faellt.

SCHULDEN B 0084 A 02 /BAS --

BASICCODE 2-Programm zur Berechnung von Ratenkrediten und sonstigen Schuldenvertraegen.

DISKKILL S 0095 A 01 /BAS --

Dieses Programm ueberschreibt alle freien Sectoren einer Diskette mit Nullen und loescht dabei die nach dem killen von Dateien immer noch vorhandenen Daten.

EDITTEXT T 0096 C 03 /CMD --

Ein Texteditor fuer Model III aus Micro 80, mit Sourcecodefile.

Flowchartgenerator S 0097 A 02 /BAS --

Dieses Programm erstellt Flussdiagramme von als ASCII-Files abgespeicherten BASIC-Programmen. Aus Micro 80.

TAFFY B 0098 A 01 /BAS --

Man muss mit moeglichst wenigen Versuchen ein Wort (5 Buchstaben) erraten. Die Worte stehen in DATA-Zeilen und koennen geaendert werden. Aus Micro 80.

DriveTimer S 0100 C 03 /CMD --

Mit diesem Programm koennen die Model III - Besitzer die Geschwindigkeit ihrer Laufwerke ueberpruefen. Aus Micro 80, mit Sourcecodefile.

MONDPHASEN S 0109 A 02 /BAS --

Dieses Programm ermittelt die Phase, in der sich der Mond zu einem eingegebenen Zeitpunkt befindet. Quelle: Computer persoenlich 25/82 S. 37

FUENFZEHNERSPIEL D 0110 A 04 /BAS --

Das Programm stellt das beliebte Spiel auf dem Bildschirm nach. Die Zahlen werden zufaellig verteilt. Auf Wunsch wird angegeben, ob und mit wievielen Zuegen das Problem loesbar ist.

ANGRIFF S 0112 A 00 /BAS --

Maessiges Geschicklichkeitsspiel.

PFERDERENNEN Z 0113 A 01 /BAS --

Wettspiel fuer mehrere Spieler. Maessige Grafik.

GUTHABEN B 0114 A 02 /BAS --

BASICCODE 2-Programm zur Errechnung von Sparguthaben und Rentensparvertraegen.

MAUMAU K 0115 A 02 /BAS --

Bekanntes Kartenspiel gegen den Computer.

MENUE S 0117 B 02 /BAS --

Direktoryorientiertes Menueprogramm in BASIC. Quelle: REST of 80.

BENCHMARKTEST S 0118 A 01 /BAS --

Eine Zusammenstellung von Benchmarktests, die sich vor allem zum Vergleich einzelner, auf dem TRS80/VG/Komtek lauffaehiger BASIC- Versionen und -Compiler eignet.

KURVEN M 0120 A 03 /BAS --

Ein sehr gutes und umfangreiches Programm zur Kurvendiskussion fast aller Funktionen. Mit Schaubild, Ableitung (1. und 2.), Nullstellen, Extrema und Wendestellen. Autor: Herbert Alber

---

ABENTEUER D 0122 A 02 /BAS --

Adventure in deutscher Sprache aus Homecomputer 5/84.

TEXTEDITOR T 0123 A 03 /BAS --

Bildschirmorientierter Texteditor aus CHIP.

SUPERHIRN D 0124 A 02 /BAS --

MASTERMIND-Spiel. Der Computer gibt eine Zahl vor, die erraten werden muss.

Teuer, teuer

Hochbetrieb im Rechenzentrum des Versandhauses. Ein Stoß Rechnungen ist zu schreiben.
Plötzlich bekommt die Anlage – schon ein älteres Modell – einen Tobsuchtsanfall. Erst stockt der Drucker, registriert unter wildem Geratter dieselbe Summe zweimal, frißt dann mit wahnwitziger Geschwindigkeit den Formularstapel in sich hinein und speit ihn mit Getöse wieder aus.
Während die roten Warnlampen aufblitzen und der Operator fassungslos auf den Rechnungswust blickt, sagt jemand hinter ihm: »Siehst Du, die Preise von heute können selbst eine Maschine rasend machen.«

Bitte Ruhe

Ein Besucher betritt das Rechenzentrum. Überall im Raum herrscht betretenes Schweigen. Über dem Computer wird gerade ein Schild mit der Aufschrift »Bitte Ruhe« angebracht. Der Besucher fragt nach der Ursache dieses ungewöhnlichen Verhaltens. Man bedeutet ihm mit einer Finger-auf-den-Mund-Geste, leiser zu sprechen. Und der Chefprogrammierer flüstert völlig verwirrt:
»Der Computer antwortete auf die letzte Frage: BEI DEM LÄRM KANN KEIN MENSCH KLAR DENKEN!«

Computer als Steuereintreiber

Der Computer des Aachener Finanzamtes schickte einem ahnungslosen Auto-Eigner 17 Mahnzettel ins Haus. 18 weitere Mahnungen lagen versandbereit, wurden jedoch von einem aufmerksamen Beamten zurückgehalten. Wären sie alle ans Ziel gekommen, hätte der überraschte Automobilist 10 056 DM rückständige Steuern nachzahlen sollen.
Des Irrtums Grund: Irgendwo im Zusammenspiel von Mensch und Mechanik war eine Sechs mit der Null verwechselt worden. Statt der Rückstände für 1967 versuchte der Computer die Steuern ab 1907 einzutreiben.

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PlZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Baldes	Hans	Johann-Strauss-Str. 6	8025	Unterhaching	089 /6115179
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62701
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Bozek	Hans Juergen	Gut Fachenfelde	2093	Stelle	
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Hallupp	Matthias	Stockumer Straße 405	4600	Dortmund 50	0231 /756413
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
Marx	Andreas	Mecklenburgring 48	6600	Saarbruecken	0681 /812983
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42677 (dienstl.44-450)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Ronenstrasse 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 6327	7800	Freiburg	0761 /35384
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Wies	Jean-Claude	Harthweg 9	6600	Saarbruecken	0681 /582513
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Wegen Systemwechsel ausgetreten

Sickmann Bernhard

Neuzugang :

Retzlaff Bernd

Impressum

Vorstand

Kontaktadresse für
Clubangelegenheiten
Clubbücherei /Fundgrube
Clubkasse

Günther WAGNER
Gartenstraße 4
8201 Neubeuern
Tel.: 08035 /3361
< 18 - 20 Uhr >

Programmbibliotek

Kontaktadresse

Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod /Kemel
Tel.: 06124 /3913

Redaktion

Kontaktadresse

Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
Tel.: 0791 /42877
tagsüber 0791 /44-450

Adventure-Ecke

Kontaktadresse

Alexander WAGNER
Theresienstr. 21c
8224 Chieming
Tel.: 08664 /1500

Hardware

Kontaktadresse

Walter ZWICKEL
Lengfelden 123
5101 Bergheim (Austria)
Tel.: 0043662 /51130

Redakteure

dieser Ausgabe

Manfred Held * Josef Konrad
Jens Neueder * Hartmut Obermann
Arnulf Sopp * Harald Trapp
Günther Wagner *
sowie Artikel aus: Genie Data und Computer Persönlich

Bankverbindung des CLUB 80

Sparkasse Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postscheckkonto der Sparkasse
Nr. 8077-801

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

nun zum Schluß wieder einige Zeilen von der "Redaktion".

Zunächst einmal muß ich mich bei allen, die Probleme mit den Listings hatten, entschuldigen. Es wird nicht wieder vorkommen, daß solche Listings in der INFO erscheinen. Ich hatte es leider auch erst nach dem Druck bemerkt. Für diejenigen, die gerne eine Kopie vom Original haben möchten, bin ich gerne bereit -mit der nächsten INFO- einen Abzug mitzuschicken. Meldet Euch bitte.

Zu der vor Euch liegenden 8. Ausgabe unsres INFO muß ich leider mitteilen, daß ich wenig Informationen von Euch für das INFO bekommen habe. Ich hoffe, das dies nur in der beginnenden Ferienzeit oder endigenden Frühjahrsmüdigkeit seine Ursache hat. Ich würde mich freuen, wenn sich möglichst viele Mitglieder an der INFO beteiligen. Das INFO ist ja das Aushängeschild unseres Clubs, und sollte deshalb auch etwas darstellen. Da ich privat ziemlich eingespannt bin, ist es mir zur Zeit unmöglich, noch extra einige Artikel für das INFO zu schreiben, so gern ich das auch täte. Ich rechne also mit Eurer regen Teilnahme und mache mich auf Berge von Infomaterial gefaßt.

Ein weiterer Wunsch von mir wäre, daß Ihr bei der Erstellung Eurer Beiträge bitte darauf achtet, auf den DIN-A4-Seiten zwei oder drei Zeilen frei zu lassen. Ich brauche diesen Platz um einen Lochrand -auf der geleimten Seite der INFO- zum Einheften gewähren zu können.

Dies ist nun das 8. Clubinfo. Ich hoffe, daß es Euch genau so gut wie die 7. Ausgabe gefällt. Euren Briefen nach zu urteilen, hat es Euch gefallen. -- Danke --

Bis zum nächsten INFO

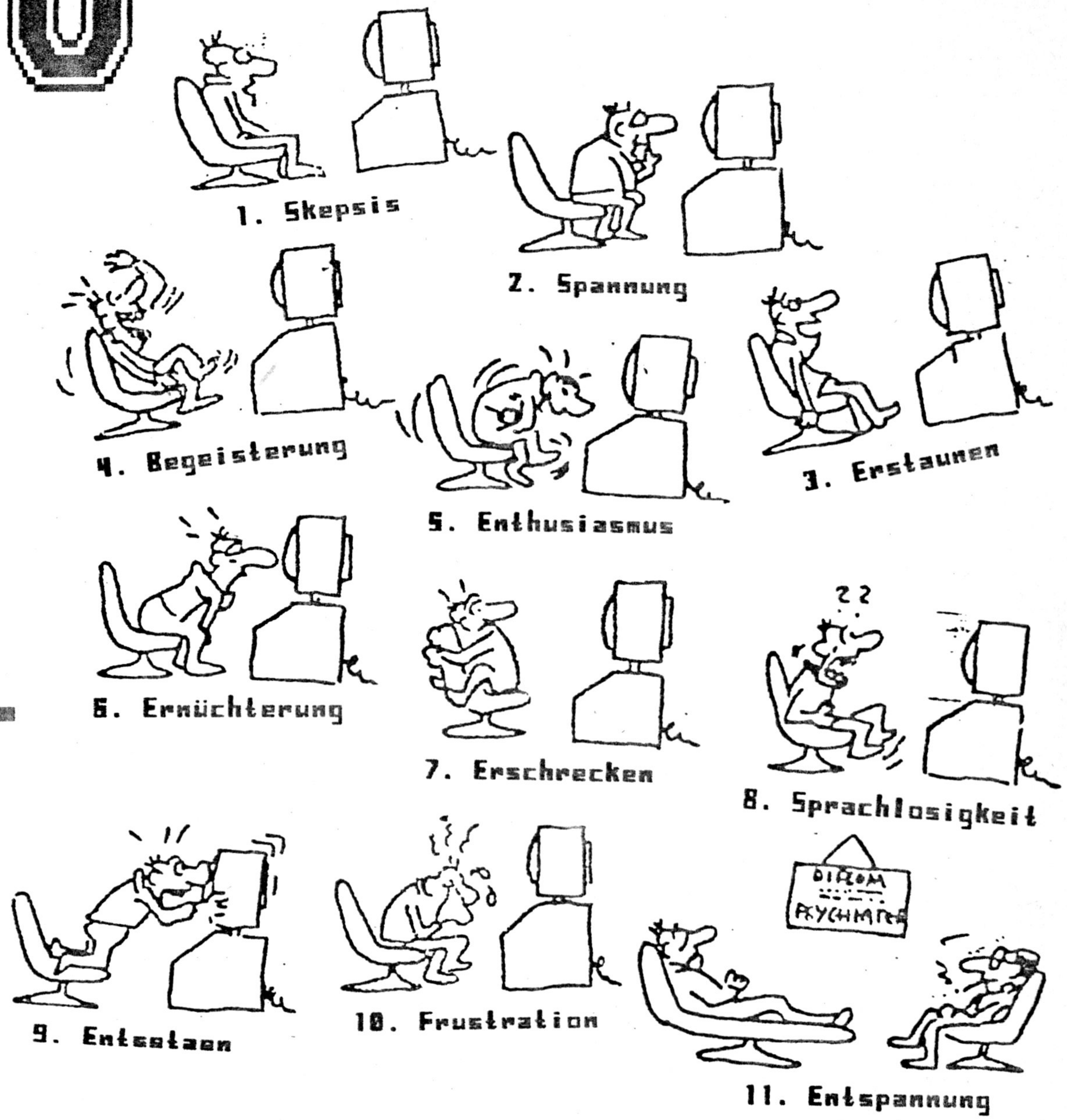
J. Neueder

CLUB 80

Clubinfo der TANDY - GENIE - und KOMTEK - ANWENDER

9. AUSGABE

Das Computerleben...



...ist nicht einfach !!!

Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubeuern

Tel.: 08035/3361 (18 - 20 Uhr)

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert . . . . 01 - 02

Die Neuen 81

Software

Festlegung von ORG 03

Directory lesen 04

Editor . . . . 05 - 08

LMOFFSET mit DISABLE DOS . . . . 09 - 10

Lprint alles einfacher 11

PDRIVE-Angaben 12

VIDHEX mit HRG . . . . 13 - 17

Unbegrenztes Spielvergnügen 18

Am Bildschirm definiert . . . . 19 - 22

Adventure - Ecke . . . . 23 - 24

Basic - Dialekte . . . . 25 - 32

Zeichencodes 38

BASIC - Wörtervergleich . . . . 33 - 36

Hardware

ECB - Bus System . . . . 39 - 44

Testadapter . . . . 45 - 46

Relais - Interface . . . . 47 - 52

Billiges CP/M. . . . . 53 - 62

Extra RAM . . . . 63 - 66

Test -- Genie IIs . . . . 67 - 68

Seite:

Börse

Wer hat was ???--Wer sucht was ??? . 71 - 72

Fragen, Fragen, Fragekasten . . . . 73 - 76

Sonstiges

Berichtigungen . . . . 37 - 38

Nachlese 69

Programmbibliothek

Neue Programme . . . . 77 - 80

Die letzten Seiten

Clubmitgliederadressen 82

Impressum 83

Schluß 84

Beilageblätter . . . . am INFO-Ende

Liebe Mitglieder,

wenn ich diese Zeilen schreibe, habe ich zwar das Info noch nicht gesehen - aber ich habe mir vom Jens kurz erzählen lassen, was so alles erscheinen wird.
Es ist demnach ein sehr gutes Info sowohl hinsichtlich Qualität als auch Umfang; das freut mich sehr - zeigt es doch davon, daß alle Mitglieder an einem guten Info interessiert sind.

Trotzdem möchte ich eine 'Nachlese' zum 8. Info bringen. Mir (und wahrscheinlich auch vielen von Euch) hat es gut gefallen; weniger interessant dürfte es für den Hartmut gewesen sein - er hat nämlich etwa 15 Seiten des Clubinfo bestritten! Es war also insgesamt relativ wenig Neues für den Hartmut dabei. Da wird Ihm dieses 9. Info schon wieder besser gefallen. Ich wollte damit nur gesagt haben, daß es wichtig ist, daß sich viele Mitglieder am Clubinfo beteiligen - ich möchte damit vor allem die Mitglieder ansprechen, die sehr wenig bzw. nichts für das Info schreiben, obwohl Sie einiges auf dem Kasten hätten.

Nun zur Clubkasse: Wir haben zur Zeit ein Guthaben von ca. 660 DM; am 19. August waren folgende Beträge noch ausständig:

Jeweils 10 DM Nachzahlung von:
Troesch, May, Wies, Hummel, Bozek, Grajewski, Boecker, Baldes, Preuss, Smerling, Retzlaff, Fröhlich
6 DM (für Info 1 und 3) von Alber

Für mich ist dies der einfachste und billigste Weg, die Mitglieder an Ihre Zahlungen zu erinnern - ich glaube (bzw. hoffe), daß sich durch diese Art und Weise niemand auf den Schlips getreten fühlt. Ferner ist ganz klar, daß sich Zahlungen von Mitgliedern mit Erscheinen dieses Info's bereits erledigt haben können!
An dieser Stelle möchte ich auch unserem Mitglied Peter Spiess danken, der für einen sehr günstigen Preis das Info umdruckt. Dadurch konnte ich seit dem 7. Info eine erfreuliche Reduzierung der Kosten feststellen.

Mit dem Info bekommt Ihr auch ein Werbeblatt für den CLUB 80. Wer Gelegenheit hat, sollte diese an günstigen Stellen (z.B. Schulen, Universitäten, Computergeschäften etc.) aushängen oder an ev. interessierte Personen weitergeben. Wer weitere Werbeblätter benötigt, kann diese bei mir anfordern (in den Größen DIN A3 - DIN A6). Weitere Werbung für den CLUB 80 wird hoffentlich in den nächsten Monaten in einschlägigen Computer-Zeitschriften erscheinen. Ich habe die entsprechenden Verlage darum gebeten.

Nun möchte ich noch unser jährliches Clubtreffen ansprechen; schön langsam muß man sich da wieder an die Planung machen. Ich denke an folgende Termine:

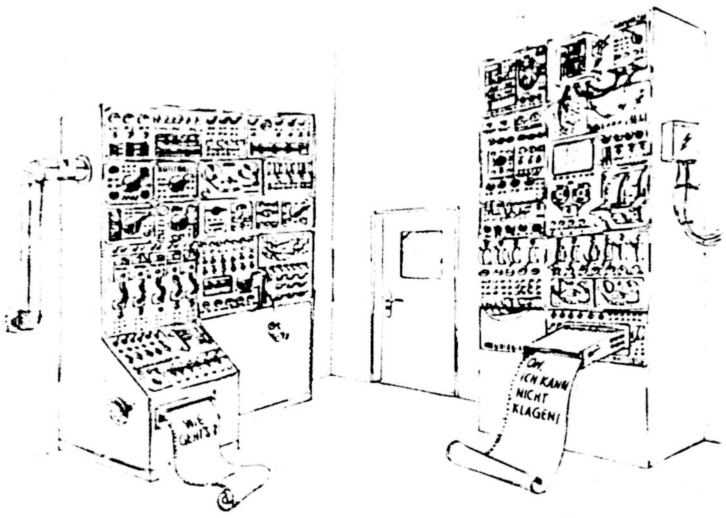
22./23. März
05./06. April
12./13. April

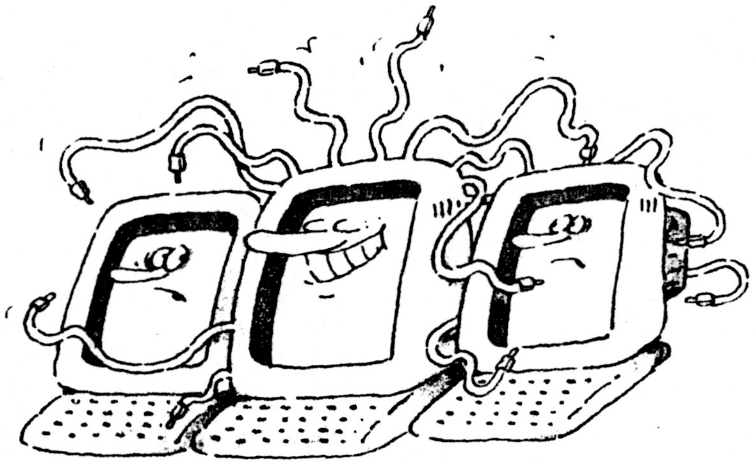
Das Treffen sollte meiner Meinung nach wieder im Großraum Frankfurt/Wiesbaden stattfinden. Bitte schreibt mir, an welchem Wochenende es bei Euch besonders gut oder schlecht geht und was Ihr sonst noch zum Thema Clubtreffen anzusprechen habt.
Ansprechen möchte ich noch die Möglichkeit, regionale Clubtreffen abzuhalten. Dies muß aber von den Mitgliedern selbst ausgehen. So möchte ich ganz gerne ein kleines Treffen im November in München abhalten (1 Nachmittag oder so). Wer Interesse am Erscheinen hat, sollte sich bitte bei mir melden.

Jedes Mitglied erhält mit diesem Info einen Abdruck seiner abgespeicherten Daten. Ich vermute, daß einige dieser Daten bei vielen Mitgliedern nicht mehr stimmen. Der eine hat ein weiteres Floppy, der andere eine HRG usw. Bitte überprüft diese Ausdrucke und falls es was zu ändern gibt, so informiert mich bitte unbedingt bis Ende September. D A N K E !

Hoffentlich habe ich Euch mit diesen Zeilen jetzt nicht zu sehr gelangweilt. Viel Freude mit dem 9. Info wünscht Euch allen (und auch mir selbst)

Euer
Günther Wagner





HALBAUTOMATISCHE FESTLEGUNG VON ORG

Bei der Programmierung in Assembler steht man öfters vor dem Problem, den Begin des Programms so festzulegen, daß es genau an einer bestimmten Stelle endet.
Die Lösung diese Problems kann man auf einfache Weise auch dem Assembler überlassen:

```
TOPMEM  EQU  0FFFFH        ;oder ein anderer Wert
PRGLEN  EQU  0000          ;zunächst irgend ein beliebiger Wert
        ORG  TOPMEM-PRGLEN
START                      ;des Programms
        'Programm'

        'letzte Anweisung des Programms'
RAMUSE  EQU  $-START-1     ;ergibt Programmlänge in HEX
        END  START         ;Programmende
```

Nun kann man das Programm assemblieren (ohne Abspeichern!), wodurch der Assembler den Wert RAMUSE berechnet. Diesen braucht man dann nur in die PRGLEN-EQU-Anweisung einzusetzen und hat nun für den endgültigen Assemblerlauf das korrekte ORG.

Josef Konrad

Hallo liebe Club-Freunde,

dieses Mal sollt Ihr auch wieder etwas von Euerem Vorstand zu lesen bekommen. Ich habe gerade etwas Zeit und möchte Euch auch als 'gutes Vorbild' vorangehen:

Schreibt doch ruhig auch mal Euere kleinen und großen Kniffe. Bestimmt ist dieser Artikel für viele kalter Kaffee - aber bestimmt auch für manchen von Interesse. Also - laßt Euch nicht lumpen und haut in die Tasten!

Euer

Directory lesen

So - nach der Vorrede aber nun zum Programm. Oft möchte ich in BASIC-Programmen das Directory in Variablen einlesen und damit arbeiten. Nach einer Vorlage aus einer alten CHIP entstand dabei folgendes Unterprogramm, das sich in jedes BASIC-Programm einbauen läßt:

```
1000  ON ERROR GOTO 1130
1010  DR$="DIR/SYS:"+RIGHT$(STR$(DR),1)
1020  OPEN"R",1,DR$
1030  FOR I=3 TO LOF(1)
1040  GET 1,I
1050  FOR II=0 TO 7
1060  FIELD 1,(II*32) AS D$,1 AS A$,4 AS D$,8 AS FS$,3 AS FE$
1070  IF NOT (CVI(A$+CHR$(0)) AND 208)=16 THEN 1100
1080  A$(J)=FS$+" "+FE$
1090  J=J+1
1100  NEXT II,I
1110  CLOSE
1120  RETURN
1130  RESUME NEXT
```

DR = Directory im Drive DR wird gelesen (muß im Hauptprogramm festgelegt werden)
FS$ = Dateiname bis zum Schrägstrich (/)
FE$ = Namenserweiterung nach dem Schrägstrich (z.B. BAS, CMD)

Selbstverständlich muß das Feld A$ ausreichend dimensioniert werden und genügend Speicher zur Verfügung gestellt werden.

Mögliche Anwendungen gibt es genügend für diese Routine. Man kann z.B. damit seine ganzen Disketten katalogisieren und zu jedem Programm noch eine kurze Erklärung abspeichern oder man kann sich in einem Programm alle Dateien mit der Namenserweiterung DAT ausgeben lassen. Man kann auch das Directory alphabetisch sortieren lassen und noch vieles mehr.

E D I T O R

Das hat nichts mit einem Konditor zu tun, sondern so nenne ich ein Unterprogramm, welches eine Bildschirmorientierte Eingabe ermöglicht. Mit Hilfe dieses UP kann man in einer Bildschirmmaske beliebig von Eingabe zu Eingabe springen, innerhalb einer Eingabe sich bewegen, Zeichen einfügen und löschen, etc. Erst zum Schluß werden dann die Eingaben direkt vom Bildschirm in die Variablen eingelesen. Ich benutze dieses UP schon seit ca. 2 Jahren. In diesen 2 Jahren wurde es laufend verbessert und ergänzt. Jetzt halte ich den Zeitpunkt gekommen, Euch dieses UP vorzustellen. Übrigens gibt es dieses UP jetzt auch in unserer Programmbibliothek (wer es z.B. nicht eintippen möchte). Die Zeilen 500-1150 des Listings gehören nicht zum UP selbst - sie sollten nur die Möglichkeiten des Editor demonstrieren.

PROGRAMMABLAUF:

A) Zeile 100 sollte immer die 1. Anweisung im Programm sein. DEFINT U definiert U für das Kopierfeld (siehe hierzu BASIC FASTER & BETTER, Seite 40 ff, 6. Clubinfo). Das Kopierfeld wird für löschen, einfügen und auffüllen mit '#' benutzt. Die DIM-Anweisung ist nur notwendig, wenn mehr als 10 Eingabefelder vorhanden sind.
B) Im Hauptprogramm wird zunächst die Eingabemaske geladen; anschließend wird der Editor aufgerufen. Nach der Eingabe können die eingegebenen Werte (befinden sich in E$()) weiter bearbeitet werden.
C) Zeile 20030: G=0 bedeutet 1. Eingabezeile
D) Zeile 20040: A erhält die Anfangsposition der 1. Eingabe
E) Zeile 20050: B liest das Zeichen an der Bildschirmposition A
F) Zeile 20060: Ausgabe des 'Cursor'
G) Zeile 20070: Tastaturabfrage
H) Zeile 20080 - Zeile 20220: Abarbeitung der Eingaben/Befehle (weiteres siehe unter Befehle); Verwaltung von A und G; in Zeile 20210 muß das GOTO-Statement für den Befehl 'SHIFT ↓ B' an das jeweilige Programm angepaßt werden.
I) Zeile 20230: keine gültige Eingabe; Rücksprung nach 20070
J) Zeile 20260-20320: Einlesen der Eingaben direkt vom Bildschirm (mittels PEEK). Die Eingaben werden in E$() abgespeichert. Wird ein '#' eingelesen, so wird das Ende der Eingabe angenommen. Das Einlesen der Eingaben dauert einige Sekunden. Das UP wird mit RETURN abgeschlossen
K) Die Zeilen 21000 - 24050 dürften verständlich sein.
L) Zeile 50030: Eine der beiden Zeilen, die bei Einbinden in Programmen geändert werden muß. Nach DATA stehen jeweils die Bildschirmpositionen für Anfang und Ende der jeweiligen Eingabefelder. Die Werte müssen an die einzelnen Bildschirmmasken angepaßt werden. Anhand der Anzahl der eingelesenen Werte weis der Editor auch, wieviele Eingabefelder vorhanden sind. Die beiden Nullen am Ende sind unbedingt notwendig. Sie signalisieren, daß alle Werte eingelesen wurden.

BEFEHLE:

→	Cursor um 1 nach rechts; wenn Ende von Eingabefeld erreicht ist, wird die nächste Eingabe angesprungen
←	Cursor um 1 nach links
↑	Cursor entweder auf das vorliegende Eingabefeld o. innerhalb des Eingabefeldes eine Zeile höher (nur bei Eingabefelder mit einer Mindestlänge von 64 Zeichen möglich)
↓	Cursor entweder auf das nachfolgende Eingabefeld o. innerhalb des Eingabefeldes eine Zeile tiefer
SHIFT →	Sprung auf Ende des Eingabefeldes
SHIFT ←	Sprung auf Anfang des Eingabefeldes
SHIFT ↑	Sprung auf Anfang des 1. Eingabefeldes
SHIFT ↓ D	Sprung auf Anfang des letzten Eingabefeldes
SHIFT ↓ E	Einfügen innerhalb des Eingabefeldes
SHIFT ↓ L	Löschen innerhalb des Eingabefeldes
SHIFT ↓ B	Abbruch und Rücksprung
@	Auffüllen des Eingabefeldes mit '#' von Cursor-Position bis Ende des Eingabefeldes; Sprung auf Anfang des nächsten Eingabefeldes
ENTER	Sprung auf Anfang des nächsten Eingabefeldes
CLEAR	Einlesen der Eingabefelder in E$(); Ende des UP

VARIABLENLISTE

A	Bildschirmposition
B	Inhalt der Bildschirmposition A
EA()	Anfang einer Eingabezeile
EE()	Ende einer Eingabezeile
F	Anzahl der Eingabefelder
G	gegenwärtiges Eingabefeld
I	FOR-NEXT-Schleife
US	Variable für Kopierfeld
A$	Inhalt der Tastaturabfrage
E$()	Inhalt der Eingabezeilen

Ich hoffe, daß dieser Artikel für den Einen oder Anderen interessant war. Das UP ist ganz einfach - nur die Erklärung fällt mir schwer. Sollten noch Fragen zu diesem Artikel auftauchen, so meldet Euch bitte bei mir.

Günther Wagner

```
100 CLEAR 5000: DEFINT U: DIM EE(15), EA(15), E$(15): GOSUB 21000: GOSUB
 23000
500 REM -----------------
510 REM *** HAUPTPROGRAMM
520 REM -----------------
550 GOSUB 1000
560 GOSUB 20000
570 CLS: STOP
1000 REM -------------------------
1010 REM *** EINGABEMASKE AUSGEBEN
1020 REM -------------------------
1030 CLS
1040 PRINT "Anrede:           #####"
1050 PRINT "Vorname:          ####################"
1060 PRINT "Familienname:     ####################"
1070 PRINT "Stra e Nr.:       ##############################"
1080 PRINT "PLZ Wohnort:      ##############################"
1090 PRINT "Telefon:          #############"
1100 PRINT "Geb.-Datum:       ##########";
1110 PRINT TAB(49) "Geschlecht:   #";
1120 PRINT "Beruf:            ##############################"
1130 PRINT "Bemerkung 1:      #############################################
################################################################
######################################################"
1140 PRINT "Bemerkung 2:      #############################################
################################################################
######################################################";
1150 RETURN
20000 REM ----------
20010 REM *** EDITOR
20020 REM ----------
20030 G=0
20040 A=EA(G)
20050 B=PEEK(15360+A)
20060 PRINT @ A, CHR$(143);
20070 A$=INKEY$: IF A$="" THEN 20070
20080 IF A$="@" THEN PRINT @ A, "#";: US(1)=15360+A: US(3)=15361+A: US(5
)=EE(G)-A-1: US(6)=-20243: GOSUB 22000: G=G+1: GOSUB 24000: GOTO 20040
20090 IF A$=CHR$(91) THEN PRINT @ A, CHR$(B);: IF A-64<EA(G) THEN G=G-1:
 GOSUB 24000: GOTO 20040: ELSE A=A-64: GOTO 20050
20100 IF A$>CHR$(31) AND A$<CHR$(128) THEN PRINT @ A, A$;: A=A+1: IF A<E
E(G) THEN 20050 ELSE G=G+1: GOSUB 24000: GOTO 20040
20110 IF A$=CHR$(9) THEN PRINT @ A, CHR$(B);: A=A+1: IF A<EE(G) THEN 200
50 ELSE G=G+1: GOSUB 24000: GOTO 20040
20120 IF A$=CHR$(8) THEN PRINT @ A, CHR$(B);: A=A-1: IF A>=EA(G) THEN 20
050 ELSE G=G-1: GOSUB 24000: GOTO 20040
20130 IF A$=CHR$(10) THEN PRINT @A, CHR$(B);: IF A+64<EE(G) THEN A=A+64:
 GOTO 20050: ELSE G=G+1: GOSUB 24000: GOTO 20040
20140 IF A$=CHR$(5) THEN PRINT @ A, CHR$(B);: US(3)=15359+EE(G): US(1)=1
5358+EE(G): US(5)=EE(G)-A: US(6)=-18195: GOSUB 22000: PRINT @ A, " ";: G
OTO 20050
20150 IF A$=CHR$(12) THEN US(1)=15361+A: US(3)=15360+A: US(5)=EE(G)-A: U
S(6)=-20243: GOSUB 22000: GOTO 20050
20160 IF A$=CHR$(27) THEN PRINT @ A, CHR$(B);: G=0: GOTO 20040
20170 IF A$=CHR$(4) THEN PRINT @ A, CHR$(B);: G=F: GOTO 20040
20180 IF A$=CHR$(25) THEN PRINT @ A, CHR$(B);: A=EE(G)-1: GOTO 20050
20190 IF A$=CHR$(24) THEN PRINT @ A, CHR$(B);: A=EA(G): GOTO 20050
20200 IF A$=CHR$(13) THEN PRINT @ A, CHR$(B);: G=G+1: GOSUB 24000: GOTO
20040
```



```
20210 IF A$=CHR$(2) THEN CLS: GOTO 570
20220 IF A$=CHR$(31) THEN PRINT @ A, CHR$(B);: GOTO 20260
20230 GOTO 20070
20240 '
20250 '
20260 FOR G=0 TO F: E$(G)=""
20270 FOR I=15360+EA(G) TO 15359+EE(G)
20280 IF PEEK(I)=35 THEN 20310
20290 E$(G)=E$(G)+CHR$(PEEK(I))
20300 NEXT I
20310 NEXT G
20320 RETURN
21000 REM -------------------------
21010 REM *** KOPIERFELD VORBEREITEN
21020 REM -------------------------
21030 US(0)=8448: US(2)=4352: US(4)=256: US(7)=201
21040 RETURN
22000 REM -----------------------
22010 REM *** KOPIERFELD AUSF HREN
22020 REM -----------------------
22030    J=0: DEFUSR=VARPTR(US(0)): J=USR(0)
22040 RETURN
23000 REM ---------------------------
23010 REM *** WERTE F R EDITOR EINLESEN
23020 REM ---------------------------
23030 RESTORE: F=0
23040 READ EA(F), EE(F): IF EA(F)=0 THEN F=F-1: RETURN: ELSE F=F+1: GOTO
 23040
24000 REM -----------------------------------
24010 REM *** UP FUER RICHTIGE BESTIMMUNG VON G
24020 REM -----------------------------------
24030 IF G<0 THEN G=F
24040 IF G>F THEN G=0
24050 RETURN
50000 REM -------------------------
50010 REM *** DATA-WERTE FUER EDITOR
50020 REM -------------------------
50030 DATA 15,20,79,99,143,163,207,237,271,301,335,348,399,409,447,448,4
63,493,527,704,783,960,0,0
```

Computer in Schweden

Schweden, das Land der »freigebigen« Computer:
In der schwedischen Stadt Göteborg erhielt ein Forstarbeiter einen Gehaltsstreifen über 7 790 Kronen. Das sind ungefähr 6 000 Mark – das Zehnfache seines gewöhnlichen Gehalts. Der ehrliche Arbeiter gab den Lohnstreifen an das Personalbüro zurück. Wieder trat der Computer in Aktion: Er schrieb einen neuen Streifen über 77 900 Kronen.

DO IT YOURSELF: LMOFFSET with DISABLE DOS

Ihr kennt bestimmt alle den berühmten Anhang, den LMOFFSET erzeugt: "DO THE FOLLOWING:" etc., um ins Level II-Basic zu gelangen. Erst durch SYSTEM und "/" läßt sich das Programm starten.
Bei meinem neuen Genie IIs ließen sich nun einige der alten Programme (ohne diesen Anhang) nicht starten, weil sie (so vermute ich jedenfalls) im DOS--Bereich etwas ändern. Also mußte das DOS "ausgeschaltet" werden. Das kann man mit LMOFFSET machen, indem man das Programm verschieben läßt und dann auf die Frage "DISABLE DOS" mit "Y" antwortet. Abgesehen davon, daß dies nicht bei allen Programmen klappt, müßte das der Hobby-Programmierer auch selbst können. Was macht LMOFFSET also?
Ganz einfach: wenn man vom Verschieben absieht, hängt LMOFFSET nur ein kleines Anhängsel an das Programm an. Dieses Anhängsel sieht im Prinzip so aus:

```
          ORG       9000H     ;erste freie Adresse hinter dem
                              ;Programm
ENTRY:    EQU       5300H     ;Entrypoint des Programms
                              ;
START:    DI                  ;Interrupts sperren
          LD        HL,TAB    ;Adresse der Tabelle (s.u.)
          LD        DE,42E9H  ;feste Adresse im DOS
          LD        BC,0006H  ;Länge der Tabelle
          LDIR                ;6 Byte von (HL) nach (DE)
                              ;
          LD        HL,TEXT   ;Adresse des Textes
          CALL      4467H     ;Text bis ODH auf Bildschirm
                              ;
LOOP:     JP        LOOP      ;Endlosschleife
                              ;
TAB:      NOP
          NOP
          NOP
          JP        ENTRY
                              ;
TEXT:     DEFM      'DO THE FOLLOWING: ...'
          DEFB      ODH       ;Ende des Textes, wichtig!
                              ;
          END       START
```

Will man diesen Anhang selbst erstellen, geht man am besten so vor:
1. mit LMOFFSET Start- und Endadresse sowie den Entrypoint feststellen (aber das Programm nicht verschieben oder so etwas)
2. den ENTRYPOINT für ENTRY einsetzen und ORG mit der Endadresse + 1 setzen
3. Anhang assemblieren und abspeichern
4. im DOS eingeben: LOAD,programm/CMD und LOAD,anhang/CMD
5. DUMP,neuprog/CMD,start,ende,entry wobei start=Startadresse des Originalprogramms, ende=Endadresse des Anhangs und entry=Entrypoint des Anhangs (alles mit LMOFFSET erfahrbar)
(Punkt 4 und 5 lassen sich auch mit einem geeigneten Monitor erledigen)

Das Programm wartet nun nach dem Aufruf auf <Break und Reset>. Durch die Kommandofolge "SYSTEM" und "/" wird die nach 4209H gespeicherte Bafehlsfolge angsprungen, die das Programm startet.

Dieser Aufwand wäre ja eigentlich unnötig, denn das macht LMOFFSET automatisch. Doch nun wird es erst interessant!!!
Was, wenn LMOFFSET streikt?! Mir passierte es: ASYLUM II lädt direkt in den Bildschirmspeicher und was weiß ich wo sonst noch hin. LMOFFSET zeigte nicht einmal die drei oben angesprochenen Adressen.

Jetzt müssen wir auf das RAM als Arbeitsplatz verzichten und uns direkt auf die Diskette begeben. Natürlich mit dem von Hartmut Obermann modifizierten Superzap.
Man sollte vermuten, daß wir einfach den assemblierten Anhang hinter das Programm hängen können. Doch so einfach ist das leider nicht.
Beim Laden von der Diskette muß das DOS ja wissen, wohin es das Programm bzw. bestimmte Teile laden soll und wo der Entrypoint ist. Der Entrypoint ist leicht zu finden: wir suchen uns den letzten Sektor des Programms (nicht mit dem ersten hinter dem Programm verwechseln!!!) und suchen dort nach den Bytes "0202". Hinter diesen Bytes steht der Entrypoint, natürlich wieder in dieser verqueren Darstellung, so daß die beiden Bytes also vertauscht werden müssen. Das ist aber einfach zu erkennen, denn sonst würde meist eine Adresse im ROM herauskommen, an der der Start zumindest unwahrscheinlich ist.
Nach diesen vier Bytes "0202xxxx" endet das Programm. Da kann man soviel ändern, wie man will, nichts passiert.
Also muß unser Anhang die 0202... überschreiben. Dafür fügen wir dann am Ende unseres Anhangs ein: "0202yyyy", wobei yyyy der Entrypoint des Anhangs ist.
Wer meinen Vorschlag nun so versucht, bekommt einen "LOAD FILE FORMAT ERROR" (Ihr seht, das ich alle Fehler schon gemacht habe). Es fehlt noch was, und zwar die Ladeadresse des Anhangs aus unserem ORG. Wenn wir den Anhang auf der Diskette betrachten, finden wir das Fehlende: "01xx yyyy".
Die 01 teilt dem DOS mit, daß eine neue Ladeadresse folgt. xx gibt die Länge des Blocks an, yyyy die Ladeadresse. Da die 01 auch im Programm vorkommen kann, muß die Länge des Blocks genau auf die nächste 01 oder 02 (zum Starten) zeigen. Was allerdings mit Blöcken ist, die länger als FF Byte sind, kann ich nicht sagen. Für solche Feinheiten haben wir bessere Experten im Club (Arnulf und Hartmut dürfen sich aufgefordert fühlen, hierzu etwas zu schreiben).
Glücklicherweise ist das Problem bei mir noch nicht relevant gewesen, denn einen so langen Anhang habe ich noch nicht fabrizieren müssen. Allerdings habe ich einen kleinen Trick benutzt: im oben gezeigten Programm müßte der Text 64x16 Zeichen lang sein, damit der Bildschirm voll beschrieben wird. LMOFFSET benutzt im Original-Anhang noch eine zweite Blockübertragung, um den Bildschirm zu löschen. Doch das läßt sich einfacher erreichen. Am Anfang des Textes fügt Ihr ein:

```
          DEFB      1CH
          DEFB      1FH
```

was den Bildschirm löscht. Um ein Return (Enter, Newline) zu erzeugen, muß im Text

```
          DEFB      0AH
```

auftauchen. Schon schrumpft der Text zusammen.
Nun könnt ihr herrlich auf der Diskette rumscratchen und LMOFFSET zum alten Eisen werfen, denn die erste Ladeadresse des Programms befindet sich natürlich am Anfang nach 01xx (es müssen nicht die ersten Byte im ersten Sektor sein!) und die Endadresse könnte man sich ausrechnen. Und wer eine neue Ladeadresse haben will: verschieben dürfte mit LDIR kein Problem sein.
Doch dieses <Break und Reset> ist doch immer etwas umständlich. Was erreicht man eigentlich damit? Letztlich nichts anderes als ein System ohne DOS, aber mit dem Programm im Speicher. Wenn Ihr Euch nun das gute alte ROM-Listing anschaut und bei 0000H anfangt, seht Ihr die Befehle, die die DOS-Vektoren zurücksetzen und das Level II-Basic initialisieren. Viele Programme benötigen bestimmt nicht eine volle Initialisierung, sondern nur das Rücksetzen der DOS-Vektoren und einige andere Kleinigkeiten. Seht Euch die Befehle an und schreibt den Anfang ab (0674H-0689H, 0075H ff.). Dann noch ein JumP zum Entrypoint (ohne <Break und Reset>). Das bringt eventuell die gleichen Ergebnisse wie die ursprüngliche Prozedur. Versprechen kann ich nichts, aber bei einigen Programmen lohnt es sich bestimmt!
Gerald Schröder

LPRINT alles - einfacher!!! (Teil 3)

Ach, wie hüpft mir das Herze lustig in der Brust! Da ist schon wieder einer ins Lager der Maschinensprache-Maniacs konvertiert. Lieber Hartmut, Deine Routine, mit der sich beliebige Codes mit LPRINT auf den Drucker ausgeben lassen, funktioniert einwandfrei. Wenn ich ihr jetzt eine Alternative entgegensetze, so geschieht das nur aus missionarischem Ehrgeiz. Die logische Hälfte des Assembler-Programmierens hast Du zwar schon richtig gut drauf, aber hier folgt ein wenig zur ökonomischen Hälfte:

Die Lösung, mit den Indexregistern zu arbeiten, liegt nahe, wenn ein Byte von einem direkt greifbaren Speicherplatz (hier (SP)) ein Stückweit entfernt ist. Aber das kostet Platz. Die indizierten Adressierungen erfordern nicht nur ein zusätzliches Switchbyte (DD bzw. FD), sondern manchmal sind auch Umwege nötig. So mußtest Du z. B. den sonst unbenutzten Speicherplatz 4018h im Tastatur-DCB als Zwischenspeicher hernehmen.

Eine genauere Untersuchung des Interpreters hilft weiter: Das Registerpaar HL wird in den Treibern beim Einsprung nur als Adressenhalter benutzt; mit JP (HL) werden sie adressiert. Einmal im Treiber angelangt, ist HL frei. Also holen wir uns das zu druckende Byte folgendermaßen:

```
0000  211101   LD   HL,0111h   ;Summand, MSB 1 zu hoch
0003  25       DEC  H          ;jetzt HL = 0011h
0004  39       ADD  HL,SP      ;Stelle im Stack errechn.
0005  7E       LD   A,(HL)     ;Byte aus dem Stack laden
0006  C3B405   JP   05b4h      ;und ausdrucken lassen
```

Die ersten beiden Befehle erscheinen widersinnig oder zumindest umständlich. Wozu HL mit einem Wert laden und den dann sofort im MSB (H) verändern? Der Ladebefehl LD HL,0011h (Codes dez. 33, 17, 0) hätte eine logische 0 enthalten. Die hätte der Interpreter aber mitten in der Stringvariablen LN$ als Zeilenende-Flag erkannt. Das wäre tödlich gewesen. Daher mußte dieser kleine Kunstgriff die 0 vermeiden.

HL wird mit dem Stackpointer addiert, so daß die Summe (die Adresse des Bytes) in HL steht. Aus (HL) wird es in den Akku geladen. Auch die Überprüfung der Druckerbereitschaft und die Ausgabe auf's Papier kann man sich schenken, denn der Interpreter weiß auch, wie das geht. Was Du in den Zeilen 400 bis 900 programmiert hast, steht dort ab Adresse 05B4h. Zusätzlich wird dabei sogar der Zeilenzähler im DCB unterhalten, was allerdings schon fast seit Gutenbergs Zeiten überflüssig ist. Mit JP 05B4h geht also der Code auf den Drucker. Diese Interpreter-Routine ist ebenfalls ein Unterprogramm, so daß früher oder später auch ein RET angetroffen wird. Deshalb kommen wir elegant aus dem Druckertreiber wieder heraus ohne ein eigenes RET.

Das kleine BASIC-Programm, mit dem der neue Treiber implementiert wird, kann nun folgendermaßen angepaßt werden:

```
10 LN$="123456789": REM 9 Bytes genügen!
30 FOR I=1TO9: REM Daher auch eine kürzere Schleife
40 DATA 33,17,1,37,57,126,195,180,5: REM und weniger DATA
```

Die übrigen Zeilen bzw. das übrige der gelisteten Zeilen können von Hartmuts Programm unverändert übernommen werden.

Arnulf Sopp

PDRIVE-Angaben

Sicher ging es nicht nur mir so; man bekommt eine Disk ins Haus und weiss ueber die Laufwerksangaben ueberhaupt nichts. Saemtliche PDRIVE-Parameter zu testen ist doch etwas zu langwierig. Ausserdem geben die Laufwerke dabei Geraeusche von sich, dass einem Angst und Bange wird.
Ich machte mich also an die Suche und habe im BOOT/SYS, Sektor 2 die Parameter gefunden (ich hoffe, dass es auch stimmt). Die Tabelle ist in Byte 00 bis 9F gespeichert, wobei jedes Laufwerk die Bytes 0 bis F belegt. Also z.B. Laufwerk 1 : 10 - 1F / Laufwerk 4: 40 - 4F usw.
Die PDRIVE-Parameter sind wie folgt verteilt:

```
Byte x0: Spur-Nummer DIRectory
     x1:
     x2:
     x3: TC   SP
     x4: SPT  SEK
     x5: GPL  EIB
     x6: (HD)
     x7: (HD)
     x8: DDSL SBIV
     x9: DDGA AEIV
     xA: (HD)
     xB: (HD)
     xC: TSR  SWZ
     xD: TI 1. Teil wobei A=01/B=02/C=04/D=08/E=10
     xE: TI 2. Teil wobei H=01/I=02/J=03/K=04/L=08/M=10
     xF: TD wobei A=00/B=01/C=02/D=03/E=04/F=05/G=06
```

Die Tabelle wurde einer Systemdisk NEWDOS 80 V 2.5 fuer Modell III entnommen. Bei V 2.0 duerfte es sich wohl aehnlich verhalten. Beim Lesen einer Model I - Disk im Modell III waere schliesslich noch der Befehl WRDIRP (bzw. ueber DIRCHECK) notwendig, da das Directory schreibgeschuetzt sein sollte. (KH)

Computer in Dänemark

Dänemark, das Land der »beschränkten« Computer: Sofie Madsen, mit 104 Lenzen älteste Oma ihres Landes, zweifelte an der Schärfe ihrer Augengläser. Erhielt sie doch wahrlich eine Aufforderung, sich als Erstkläßlerin einschulen zu lassen. Oma Sofie knirschte mit den Leihzähnen . . .
Die Hollerith-Anlage ihres Heimatortes konnte in der Altersspalte nur bis 99 »zählen«.
Und somit war die 104jährige Sofie erst fünf Jahre alt – und schulpflichtig.

VIDHEX - Hexanzeige des Bildschirms mit der HRG

Die Sonderzeichen des Genie 3 (serienmäßig), 2 und 1 (nach Hardwareänderung) mit den ASCII-Codes 00-1F sehen zwar ganz putzig aus. Sieht man sie jedoch in einem Dump mit dem Debugger oder erscheinen sie auf andere Weise auf dem Bildschirm, so ist man gelegentlich ratlos. Sie sind nämlich in keiner Tabelle zu finden, so daß man ihre ASCII-Codes bei Bedarf nur erraten kann. Um diesem Mißstand abzuhelfen, entwarf ich das Programm VIDHEX/CMD, das mit Hilfe der HRG 1b anstelle der ASCII- und Sonderzeichen kleine zweistellige Hexzahlen anzeigt. Was dabei herauskommt, ist in dem HRG-Ausdruck am Ende des Listings zu sehen. Es handelt sich um eine Anzeige des Debuggers.

Die Ladeadresse 3900 ist mit der serienmäßigen Hardware natürlich nicht zu realisieren. Das geht mit einer kleinen Zusatzplatine, die Helmut Bernhardt in c't vorstellte (RAM von 3900-3BFF). Wer sie nicht hat, braucht nur ORG auf irgendeine andere Adresse zu setzen (genau 256 Bytes müssen bis zum Himem noch mindestens frei sein).

Das Programm ist mit ZEUS/CMD erstellt. Für EDTASM müssen alle Labels auf max. 6 Zeichen gekürzt werden. Kleinbuchstaben sind nur in den Kommentaren erlaubt. DB muß in DEFB geändert werden, DW in DEFW. In einem DEFB-Statement dürfen die einzelnen Bytes auch nicht durch Komma getrennt in eine gemeinsame Zeile gepackt werden, sondern jedes Byte erhält eine eigene Zeile. Das alles ändert aber nichts an der Programmlogik, die im folgenden erläutert werden soll.

Unter allen Steuerzeichen ist (zumindest in der Direkteingabe über die Tastatur) ESC das überflüssigste (escape, Shift-Hochpfeil, ASCII 27 bzw. 1B). Es hat zudem den Vorteil, daß sich mit ESC nichts auf dem Bildschirm ändert, solange man nicht etwa mit dem Level-4-ROM arbeitet und ESC zusammen mit einer anderen Taste drückt. Deshalb wurde als Trigger für die Umwandlung des Bildschirms ESC ausgewählt. Hierzu erhielt der Tastaturtreiber einen kleinen Vorspann, der vor der normalen Tastaturabfrage prüft, ob Shift mit dem Hochpfeil gedrückt wurde. Dies geschieht im Programmsegment newdrv. Wird kein oder ein anderes Zeichen festgestellt, geht es weiter an 4516, dem Beginn des normalen Tastaturtreibers (G-DOS bzw. H-DOS).

Nach ESC erfolgt nun die Umwandlung. Hierzu wird zunächst der Bildschirm gerettet, denn er muß gelöscht werden, um die HRG-Anzeige nicht zu stören. Ein Puffer wird für das Kilobyte des Bildschirms gebraucht. Die HRG benutzt von jedem Byte ihres Speichers nur 6 Bits zur Anzeige. Die beiden höchstwertigen Bits bleiben unsichtbar. Was liegt näher, als dort den Bildschirm zu verstauen? Dazu wird zunächst der Videozeiger mit 3C00 geladen, dem Beginn des Screens. Der HRG-Zeiger kommt auf den Anfangswert 0000. In den verschachtelten Schleifen vidsav1 und vidsav2 wird jeweils der Akku mit dem Videobyte geladen und mit C0 undiert, um die übrigen Bits zu löschen. Dieses Viertelbyte im Akku wird nun über den Port 5 in die HRG geladen. Jetzt werden die beiden oberen Bits der Videostelle nach unten rotiert; die nächsten beiden Bits stehen an. Das ist ein bißchen fummelig und kostet auch ein paar Bytes Programmcode, aber 1 kB Bildschirmpuffer im RAM wäre ein vielfacher Verlust.

Anschließend wird der HRG-Speicher gelöscht. Weshalb das nötig ist, wird später erklärt. Seine internen Adressen gehen von 0000 bis 2FFF. Der Zeiger HL startet deshalb nach dem Puffern des Bildschirms mit 0400 (= 1 kB), wo der HRG-Zeiger jetzt gerade steht. Das MSB der HRG wird bei jedem Schleifendurchlauf auf 30, den ersten nicht mehr erlaubten Wert geprüft. Dies geschieht in der Schleife clear.

Nach dem Löschen geht es im UP hexdisp weiter. Es wird jeweils ein Bildschirmzeichen geladen und mit Blank verglichen. Um den Bildschirm übersichtlich zu halten, wird ein Blank nicht mit der Hexzahl 20 angezeigt. In diesem Fall wird das UP byte nicht angesprungen. Das würde ohne vorheriges Löschen der HRG bedeuten, daß die alten Codes dort erhalten blieben. Daher war zuvor die Löschung erforderlich.

Im UP byte wird die Videoadresse auf die HRG-Adresse umgerechnet. Hierzu braucht (zumindest für die oberste Dotzeile) nur 3C vom MSB subtrahiert zu werden. In diesem Falle wird mit 03 undiert, was auf dasselbe hinausläuft. Nacheinander werden nun die beiden Halbbytes in eine Ziffer umgerechnet und angezeigt:

Das obere Nibble wird zunächst durch 16 dividiert und damit ins untere geschoben. DE wird nun als Zeiger auf die Zeichensatztabelle chrtab geladen und das UP nibble angesprungen. Je nach Ziffer wird der Zeiger dort bis zu 16mal um fünf Stellen weitergerückt, denn jede Ziffer setzt sich aus fünf Dotmustern zusammen. Die jeweilige Stelle in der HRG wird im UP HRGadr adressiert und bei output das Byte ausgegeben. Für das nächste der fünf Bytes pro Ziffer muß der HRG-Zeiger HL um 1 kB erhöht werden. Das geht am einfachsten durch viermaliges Inkrementieren des MSB.

Um beide Hexziffern auf dem engen Raum einer einzigen Video-Anzeigestelle gut unterscheidbar zu machen, stehen sie untereinander und sind um zwei Dots in der Waagerechten gegeneinander verschoben. Das geschieht beim unteren Nibble (Einerstelle der Zahl) durch zweimaliges RLCA. Da das UP Nibble nicht unterscheidet, welche der beiden Ziffern gerade ansteht, werden die beiden RLCA immer abwechselnd abgearbeitet oder übersprungen. Dazu dient der relative Sprungbefehl JR displc. Die Sprungdistanz displc wird mit dem XOR-Befehl gleich zu Beginn des UP nibble ständig zwischen 00 und 02 (durch Verwendung von Labels variabel gehalten) hin- und hergeschaltet.

Wenn beide Ziffern angezeigt sind, geht es mit der nächsten Bildschirmstelle weiter, bis alle Bildschirmzeichen als Hexzahlen auf dem HRG-Screen stehen. Die HR-Graphik bleibt stehen, bis irgendeine Taste gedrückt wird. Jetzt muß der alte Bildschirm wieder restauriert werden. Die Bildschirmzeichen, die bisher geduldig im Puffer gewartet haben, werden zurückgeladen. Das geschieht im Prinzip wie das Puffern, nur eben umgekehrt: Die beiden oberen Bits aus einer HRG-Stelle werden aus dem Akku hinaus- und in ein Bildschirmbyte hineinrotiert. Damit ist endlich alles erledigt, mit RET geht es zurück ins Betriebssystem.

Die Befehle OUT (1),A und OUT (0),A zum Ein- und Wiederausschalten der HRG stehen an frühst- bzw. spätestmöglicher Stelle. Daher kann der User bei 1,77 MHz ungefähr 2-3 Sekunden lang ein wildes Schauspiel auf dem Screen beobachten. Gute Unterhaltung!

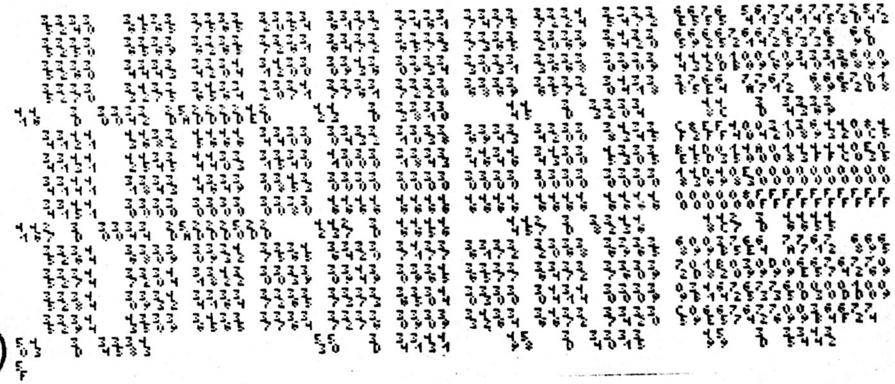
Arnulf Sopp

```
                ;=====================================================
                ;       Umwandlung der Bildschirmanzeige von ASCII-
                ;       Zeichen in Hexzahlen mit Hilfe der HRG 1b
                ;                (C) '85 by The HACKTORY
                ;=====================================================

3900                            ORG     3900h           ;Lade- u. Einsprungsadr.

                ;neuer Tastaturtreiber für Shift-Aufwärtspfeil
3900  3A4038    newdrv  LD      A,(3840h)       ;Tastatur Steuerzeichen
3903  CB5F              BIT     3,A             ;Aufwärtspfeil gedrückt?
3905  2804              JR      Z,exit          ;falls nein
3907  3A8038            LD      A,(3880h)       ;ja, Shift-Reihe
390A  B7                OR      A               ;Shift gedrückt?
390B  CA1645    exit    JP      Z,4516H         ;falls nicht Sh.-Hochpf.

                ;nach Shift-Hochpfeil Bildsch. retten
390E  D301              OUT     (1),A           ;HRG einschalten
3910  21003C            LD      HL,3c00h        ;Bildschirmadresse
3913  55                LD      D,L             ;DE <- 00xx, HRG-Adresse
3914  5D                LD      E,L             ;DE <- 0000
3915  D5                PUSH    DE              ;HRG-Adresse retten
3916  E5                PUSH    HL              ;dto. Videoadresse
3917  01C004    vidsav1 LD      BC,04c0h        ;4*2 Bits/Byte, Konst. C0
391A  CDA839    vidsav2 CALL    HRGadr          ;HRG-Stelle adressieren
391D  7E                LD      A,(HL)          ;Bildschirmzeichen
391E  A1                AND     C               ;nur oberste 2 Bits
391F  D305              OUT     (5),A           ;auf HRG ausgeben
3921  CB06              RLC     (HL)            ;Zeichen 2 Bits aufrücken
3923  CB06              RLC     (HL)
3925  10F3              DJNZ    vidsav2         ;bis 1 Byte fertig
3927  23                INC     HL              ;nächste Videostelle
3928  CB74              BIT     6,H             ;Bildsch. überschritten?
392A  28EB              JR      Z,vidsav1       ;falls noch nicht

                ;HRG-Speicher ab nächster freier Stelle löschen
392C  CDA839    clear   CALL    HRGadr          ;HRG-Stelle adressieren
392F  FE30              CP      30h             ;Speicher überschritten?
3931  2805              JR      Z,cleared       ;falls ja
3933  AF                XOR     A               ;A <- 00
3934  D305              OUT     (5),A           ;diese Stelle löschen
3936  18F4              JR      clear           ;nächste Stelle
3938  E1        cleared POP     HL              ;Bildschirmanfang
3939  E5                PUSH    HL              ;für später retten

                ;Bildschirm von ASCII- in Hexanzeige ändern
393A  E5        hexdisp PUSH    HL              ;Videozeiger retten
393B  4E                LD      C,(HL)          ;Bildschirmzeichen
393C  3E20              LD      A,' '           ;Blank
393E  B9                CP      C               ;Blank? (nicht verändern)
393F  77                LD      (HL),A          ;diese Stelle löschen
3940  C46539            CALL    NZ,byte         ;kein Bl., 1 Byte umwand.

                ;Zeichen umgewandelt oder Blank überspr.; nächst. Zeichen
3943  E1                POP     HL              ;Bildschirmzeiger
3944  23                INC     HL              ;nächste Stelle
3945  CB74              BIT     6,H             ;Bildsch. überschritten?
3947  28F1              JR      Z,hexdisp       ;nein, nächstes Byte

                ;Nach Anzeige auf Taste warten, dann Bildschirm restaur.
3949  CD4900            CALL    0049h           ;auf Tastendruck warten
394C  E1                POP     HL              ;Videoadresse
394D  D1                POP     DE              ;HRG-Adresse
394E  0604      restor1 LD      B,4             ;4*2 Bits/Byte
```



```
3950  CDA839    restor2 CALL    HRGadr          ;HRG-Stelle adressieren
3953  DB04              IN      A,(4)           ;HRG-Byte holen
3955  07                RLCA                    ;2 oberste Bits
3956  CB16              RL      (HL)            ;in den Bildsch. laden
3958  07                RLCA
3959  CB16              RL      (HL)
395B  10F3              DJNZ    restor2         ;bis 1 Videobyte fertig
395D  23                INC     HL              ;nächste Bildschirmstelle
395E  CB74              BIT     6,H             ;Bildsch. überschritten?
3960  28EC              JR      Z,restor1       ;falls noch nicht
3962  D300              OUT     (0),A           ;HRG ausschalten
3964  C9                RET                     ;Tastaturtreiber verlass.

                ;UP Hexanzeige: 1 Byte ändern
3965  7C        byte    LD      A,H             ;MSB der Videoadresse
3966  E603              AND     03              ;Adr. Vid. -> Adr. HRG
3968  57                LD      D,A             ;neues MSB
3969  5D                LD      E,L             ;HRG-MSB wie Video-MSB
396A  79                LD      A,C             ;Videozeichen
396B  E6F0              AND     0f0h            ;oberes Nibble
396D  0F                RRCA                    ;ins untere schieben
396E  0F                RRCA
396F  0F                RRCA
3970  0F                RRCA
3971  21AB39            LD      HL,chrtab-5     ;vor Tab. f. Hexzeich.
3974  E5                PUSH    HL              ;brauchen wir noch
3975  CD7C39            CALL    nibble          ;oberes Nibble anzeigen
3978  79                LD      A,C             ;alter Code
3979  E60F              AND     0fh             ;unteres Nibble
397B  E1                POP     HL              ;Tabellenzeiger

                ;einzelnes Halbbyte in die HRG laden
397C  47        nibble  LD      B,A             ;als Zähler i. d. Tabelle
397D  3A9639            LD      A,(displc)      ;Sprungdistanz
3980  EE02              XOR     output-displc-1 ;umschalten
3982  329639            LD      (displc),A      ;neu laden
3985  04                INC     B               ;wegen DE = Tabelle -5
3986  23        seekchr INC     HL              ;Zeiger nachstellen
3987  23                INC     HL              ;über 5 Stellen, weil
3988  23                INC     HL              ;5 Codes pro Zeichen
3989  23                INC     HL
398A  23                INC     HL
398B  10F9              DJNZ    seekchr         ;bis Code gefunden
398D  0605              LD      B,5             ;5 Dotzeilen pro Zeichen
398F  C5        nibloop PUSH    BC              ;Zähler retten
3990  CDA839            CALL    HRGadr          ;HRG-Stelle adressieren
3993  1B                DEC     DE              ;HRG-Zeiger korrigieren
3994  7E                LD      A,(HL)          ;Dotzeile laden
3995  1800              JR      $+2             ;variable Sprungdistanz
3996            displc  EQU     $-1             ;hier Distanzbyte
3997  07                RLCA                    ;lower Nibble verschieben
3998  07                RLCA                    ;um 2 Dots
3999  4F        output  LD      C,A             ;Dotzeile retten
399A  DB04              IN      A,(4)           ;HRG-Byte mit Videocode
399C  B1                OR      C               ;mit Dotzeile verknüpfen
399D  D305              OUT     (5),A           ;Dotzeile in HRG laden
399F  14                INC     D               ;im MSB um 1 kB erhöhen
39A0  14                INC     D               ;für nächste Dotzeile
39A1  14                INC     D
39A2  14                INC     D
39A3  23                INC     HL              ;nächster Code für Ziffer
39A4  C1                POP     BC              ;Zähler restaurieren
39A5  10E8              DJNZ    nibloop         ;bis Nibble angezeigt
39A7  C9                RET                     ;zurück

                ;UP, um die HRG-Adresse auszugeben
```

```
39A5  7B        00131 HRGadr  LD     A,E           ;LSB der HRG-Adresse
39A9  D302      00132         OUT    (2),A         ;auf Port ausgeben
39AB  7A        00133         LD     A,D           ;MSB
39AC  D303      00134         OUT    (3),A         ;dto.
39AE  13        00135         INC    DE            ;nächste HRG-Stelle
39AF  C9        00136         RET
                00137
                00138 ;Codetabelle für HRG-Zeichensatz
39B0  02        00139 chrtab  DB     2,5,5,5,2     ;0
39B5  04        00140         DB     4,6,5,4,4     ;1
39BA  03        00141         DB     3,4,2,1,7     ;2
39BF  07        00142         DB     7,4,2,4,7     ;3
39C4  04        00143         DB     4,5,7,4,4     ;4
39C9  07        00144         DB     7,1,3,4,3     ;5
39CE  06        00145         DB     6,1,7,5,2     ;6
39D3  07        00146         DB     7,4,2,1,1     ;7
39D8  02        00147         DB     2,5,2,5,2     ;8
39DD  02        00148         DB     2,5,7,2,1     ;9
39E2  02        00149         DB     2,5,7,5,5     ;A
39E7  03        00150         DB     3,5,3,5,3     ;B
39EC  06        00151         DB     6,1,1,1,6     ;C
39F1  03        00152         DB     3,5,5,5,3     ;D
39F6  07        00153         DB     7,1,3,1,7     ;E
39FB  07        00154         DB     7,1,3,1,1     ;F
                00155
                00156 ;neue Tastatur-Treiberadresse im DCB
4016            00157         ORG    4016h         ;und zwar hier
4016  0039      00158         DW     newdrv        ;Adresse
                00159
3900            00160         END    newdrv        ;dort Einsprung

00000 Fehler

HRGadr   39A8   byte     3965   chrtab  39B0   clear    392C   cleared 3938
displc   3996   exit     390B   hexdisp 393A   newdrv   3900   nibble  397C
nibloop  398F   output   3999   restor1 394E   restor2  3950   seekchr 3986
vidsav1  3917   vidsav2  391A
```



Unbegrenztes Spielvergnügen
===========================

Sicherlich hat sich schon die ein oder andere Spieler - Natur geärgert, wenn eine schwierige Stelle im Ablauf der Bildchen zu einem großen Verschleiß an "Men" oder "Ships" geführt hat. In zwei Fällen läßt sich Abhilfe schaffen mit Debug oder einem Monitor:

DONKEY-KONG läd bei mir von 7E2C bis B87B, Einsprung 9000.
Eine Änderung an der Stelle AFD6 : 3D = Dec A
in 00 = Nop
...und bei einem crash erfolgt kein Abzug mehr, es geht mit dem gleichen Guthaben an "Männchen" und Punkten weiter.

JUNGLE-BOY von 5200 bis BFFF, Einsprung 5300
Adresse 77D0 : 3D
in 00 ... erbringt den gleichen Erfolg wie oben.

Nun wünsche ich allen viel Spaß beim Üben der kniffligen Passagen und ich hoffe die Hiscore-Freaks sind nicht allzu böse auf mich!

Dieter Kasper

Freitag, 01011.
August 010001 (lt. L.)*

Als der Computer ersetzt wurde

*(laut Leibniz)

TRS-80 Modell I und III, Star Gemini-10x – Zeicheneditor

Am Bildschirm definiert

Ein Bildschirmeditor für die definierbaren Zeichen des Matrixdruckers Star Gemini-10x ist dieses Programm. Es läuft auf den Rechnern Model I und III von Tandy und Kompatible.

Der Matrixdrucker Gemini-10x von Star verfügt über einen vom Benutzer definierbaren Zeichensatz („Download Character Set"). Selbstdefinierte Zeichen können mit Hilfe dieses Zeichensatzes genauso benutzt werden, wie es mit dem normalen Zeichensatz möglich ist. Lediglich kursive Schrift (Italics) und die Umschaltung zwischen nationalen Sonderzeichen sind damit nicht verfügbar. Die Definition dieser Zeichen ist jedoch eine umständliche Angelegenheit. Für jedes Zeichen müssen 9 × 7 Punkte als gesetzt oder nicht gesetzt bestimmt werden. Zusätzlich ist noch festzulegen, ob das Zeichen zum Erreichen einer Unterlänge beim Druck verschoben werden muß.

Das hier vorgestellte Programm erleichtert Benutzern eines TRS-80 mit Diskettenlaufwerk (Model I oder III) die Definition von Zeichensätzen. Es ermöglicht das Editieren von Zeichen auf dem Bildschirm und das Abspeichern von Zeichensätzen auf Diskette. Die Zeichen werden auf dem Bildschirm in einem Feld von 9 × 9 Punkten editiert. Dadurch ist es möglich, Zeichen mit Unterlänge direkt darzustellen. Vor der Übergabe der Zeichen an den Drucker wird die Umsetzung in die 9 × 7-Matrix vorgenommen und bestimmt, ob das Zeichen eine Unterlänge hat oder nicht. Beim Abspeichern eines Zeichens wird auch überprüft, ob Punkte eines Zeichens direkt nebeneinander liegen und damit den Anforderungen für Download Characters widersprechen.

Zur Bedienung

Nach einer Initialisierungsphase von etwa 20 Sekunden erscheint auf dem Bildschirm das 9 × 9-Anzeigefeld für die Zeichen. Unterhalb des Anzeigefeldes wird der Code (ASCII,dezimal) des Zeichens angezeigt und das Zeichen, so wie es im Zeichensatz des Rechners vorhanden ist. Weiterhin erscheint auf dem Bildschirm das Hauptmenü mit den verfügbaren Befehlen und eine zehnspaltige Eingabezeile mit blinkendem Cursor. Zum Editieren in dieser Eingabezeile stehen die Tasten ⟨←⟩ und ⟨Shift⟩⟨←⟩ mit den bekannten Funktionen zur Verfügung. Leerzeichen vor oder innerhalb von Befehlen sind nicht zulässig. Alle Befehlseingaben müssen mit ⟨ENTER⟩ abgeschlossen werden.

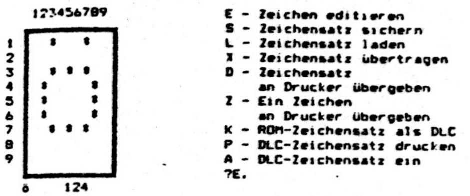
Der wichtigste Befehl des Hauptmenues ist der Befehl zum Editieren eines Zeichens (E). Die Form „E" ermöglicht das Editieren des gerade im Anzeigefeld dargestellten Zeichens, „E(" bzw. „E)" das Editieren des Zeichens mit dem nächstniedrigeren beziehungsweise nächsthöheren Code. Die Form „En" ermöglicht das Editieren des Zeichens mit dem Code n (32-126), E"x" das Editieren des Zeichens x. Das hintere Anführungszeichen kann dabei auch weggelassen werden.

Nach der Eingabe des Befehls wird das zu editierende Zeichen im Anzeigefeld dargestellt. Ein blinkender Cursor erscheint in der linken oberen Ecke des Anzeigefeldes. Ein neues Menü erläutert die Eingabemöglichkeiten. Mit den Pfeiltasten kann der Cursor dabei durch das Anzeigefeld bewegt werden.

Mit der Taste ⟨•⟩ (mit oder ohne ⟨Shift⟩) wird der Punkt an der Cursorposition gesetzt, dargestellt durch ein "•". Mit der Leertaste wird der Punkt an der Cursorposition gelöscht. Beide Tasten bewirken keine Cursorbewegung. Mit der Taste ⟨@⟩ werden alle Punkte des Anzeigefeldes gesetzt, mit der Taste ⟨CLEAR⟩ alle Punkte gelöscht. Die Taste ⟨Z⟩ ermöglicht die Übernahme eines anderen Zeichens in das Anzeigefeld. Nach dem Druck auf die Taste erscheint unterhalb des Anzeigefeldes eine Eingabezeile mit blinkendem Cursor. Hier kann entweder der Code des gewünschten Zeichens (32-126) oder das Zeichen selbst in der Form "x" angegeben werden.

Die Eingabe ist mit ⟨ENTER⟩ abzuschließen. Das angegebene Zeichen erscheint dann im Anzeigefeld. Zu diesem Zeitpunkt bereits gesetzte Punkte werden dabei gelöscht. Die Taste ⟨Q⟩ ermöglicht das Abbrechen des Editiervorgangs. Die für das editierte Zeichen bereits abgespeicherte Darstellung erscheint wieder im Anzeigefenster. Nach der Eingabe von ⟨ENTER⟩ wird das Zeichen im Anzeigefeld auf Korrektheit hinsichtlich der Anforderungen für Download Characters überprüft. Stellen, an denen zwei Punkte direkt nebeneinandergesetzt sind, werden in der Form Zeile, Spalte gemeldet. Sind sowohl in einer der beiden oberen als auch in einer der beiden unteren Zeilen Punkte gesetzt, so wird die Meldung "Unterlänge ??" ausgegeben. Das Zeichen würde später ohne die beiden oberen Zeilen an den Drucker übergeben.

Sind Unkorrektheiten entdeckt worden, so erscheint die Frage "Trotzdem abspeichern?" Wird als Antwort die Taste ⟨ENTER⟩ gedrückt, so wird das Zeichen im Anzeigefeld abgespeichert. Wird eine andere Taste gedrückt, so bleibt das bisher gespeicherte Zeichen erhalten. Ist das Zeichen korrekt, so wird es ohne weitere Fragen abgespeichert.



So sieht der Editor am Bildschirm aus

Der Befehl „S" des Hauptmenüs dient zum Abspeichern des Zeichensatzes auf Diskette. Der Befehl hat die Form S=Filename. Der Filename darf maximal acht Zeichen lang sein und keinen Zusatz (Extension) enthalten. Die Extension /DLC (für Download Character) wird vom Programm angehängt.

Der Befehl „L" dient entsprechend zum Einlesen eines Zeichensatzes von der Diskette. Er hat ebenfalls die Form L=Filename. Für den Filenamen gelten die gleichen Bedingungen wie beim S-Befehl. Es ist nicht möglich, einzelne Zeichen auf Diskette zu speichern oder von der Diskette zu lesen.

Fehler, die bei der Ein- und Ausgabe von Diskette auftreten, werden abgefangen und führen nicht zum Programmabbruch. Die am häufigsten auftretenden Fehler werden dabei mit deutscher Fehlerbeschreibung gemeldet. Bei anderen wird nur der Fehlercode angegeben. Die Beschreibung des Fehlers muß dann im BASIC-Handbuch nachgesehen werden. Die Fehlermeldung muß durch Drükken von ⟨ENTER⟩ quittiert werden, dann kehrt das Programm ins Hauptmenü zurück.

Der Befehl „X" bewirkt den Transfer eines Zeichensatzes von der Diskette auf den Drucker, ohne daß der Zeichensatz im Rechner gespeichert wird. Der Befehl hat die Form X=Filename. Für den Filenamen gilt dasselbe wie beim L- und S-Befehl.

Der Befehl „K" sorgt dafür, daß im Drucker der normale Zeichensatz in den freidefinierbaren Zeichensatz kopiert wird. Dies ist die Voraussetzung dafür, mit dem Z-Befehl einzelne Zeichen neu zu definieren.

Der Befehl „Z" sorgt dafür, daß ein selbstdefiniertes Zeichen an den Drukker übergeben wird. In der Form „Z" wird das im Anzeigefeld dargestellte Zeichen übergeben. Mit „Zn" wird das Zeichen mit dem Code n (32-126) übergeben, mit Z"x" das Zeichen x.

Der Befehl „D" bewirkt die Übergabe des gesamten selbstdefinierten Zeichensatzes an den Drucker.

Der Befehl „P" sorgt dafür, daß der auf dem Drukker vorhandene selbstdefinierte Zeichensatz ausgegeben wird. Dies geschieht immer zweizeilig: die obere Zeile zeigt die Zeichen des normalen Zeichensatzes, die untere die des selbstdefinierten Zeichensatzes.

Der Befehl „A" sorgt dafür, daß auf dem Drucker der selbstdefinierte Zeichensatz eingeschaltet wird. Alle weiteren Ausgaben des Druckers erfolgen mit diesem Zeichensatz. Die Verwendung des Befehls „P" macht diese Umschaltung allerdings wieder rückgängig.

Alle Befehle, die auf den Drucker zugreifen, prüfen zunächst die Einsatzbereitschaft des Druckers. Ist er nicht in der Lage, Zeichen zu empfangen, erscheint die Meldung "Drucker nicht bereit!" auf dem Bildschirm. Sie verschwindet wieder, sobald der Drukker einsatzbereit ist. Ein Druck auf die ⟨ENTER⟩-Taste sorgt für den Abbruch des Befehls, für den Fall, daß der Drucker nicht einsatzbereit gemacht werden kann oder soll.

Die folgenden Befehle des Hauptmenüs werden aus Platzgründen nicht auf dem Bildschirm angezeigt: Die Befehle (und) bewirken, daß das Zeichen mit dem nächstniedrigeren beziehungsweise nächsthöheren Code ins Anzeigefeld gebracht wird. Wird der Code eines Zeichens (32-126) oder das Zeichen selbst in der Form "x" eingegeben, so wird dieses Zeichen angezeigt.

Der Befehl „B" sorgt für die Rückkehr ins BASIC.

Die Speicherung des selbstdefinierten Zeichensatzes auf der Diskette erfolgt in derselben Darstellung, die für die Übergabe an den Drucker verwendet wird. So werden pro Zeichen nur 10 Byte auf der Diskette gebraucht, für den gesamten Zeichensatz also 950. Für die Umrechnung von der Darstellung auf dem Bildschirm und im Speicher in die kodierte Darstellung und zurück werden Routinen in Maschinensprache benutzt. Diese Routinen werden über den USR-Befehl angesprochen. Während der Initialisierungsphase werden die Routinen mit POKE in die Strings U0$ und U1$ gebracht. Bei der Eingabe des Programms ist darauf zu achten, daß U0$ 46 Zeichen und U1$ 53 Zeichen lang ist. Welche Zeichen verwendet werden, ist unwichtig.

Ein Hinweis zum Listing: Programmzeilen, deren Nummern nicht mit 0 enden, sind Kommentarzeilen. Sie können ohne Folgen für den Programmablauf weggelassen werden. Auch das Listing wurde übrigens mit einem selbstdefinierten Zeichensatz gedruckt. Nur so ist es möglich, den "Klammeraffen" @ und die Umlaute gleichzeitig in einem Listing zu haben. *Gerd Kluge*

```
1 REM            Bildschirmeditor für selbstdefinierbare
2 REM            Druckerzeichen (Download Characters)
3 REM
4 REM            Gemini 10X + TRS-80 Modell I und III
5 REM
100 CLS:CLEAR 20000:DEFINT A-Z:DEFSNG U
109 REM----------- Variablen initialisieren ------------------
110 DIM ZZ,I,Y,I$:BL$=STRING$(9,32):DIM Z,S,D,P:CD=32:PO=205
120 TP=100:ZP=140:DIM CP,CU,ZS,XF,IN$,NM$,CC,P,FE
130 DIM U0,U1,U0$,U1$,BP,BL,BH,   Z$(96,8),P(9),BZ$(8)
140 PRINT@465,"Bitte 20 Sekunden warten !";
149 REM----------- Array für Zeichen belegen ------------------
150 FOR I=0 TO 95: FOR ZZ=0 TO 8:Z$(I,ZZ)=BL$:NEXT ZZ :NEXT I
160 FOR ZZ=0 TO 8:Z$(96,ZZ)="*********":NEXT ZZ
169 REM----------- Bildschirmarray konstruieren ---------------
170 FOR ZZ=0 TO 8: POKE VARPTR(BZ$(ZZ)),9
180 BP=15360+PO+ZZ*64:BH=INT(BP/256):BL=BP-256*BH
190 POKE VARPTR(BZ$(ZZ))+1,BL: POKE VARPTR(BZ$(ZZ))+2,BH
200 NEXT ZZ
208 REM------------ USR-Routinen installieren -----------------
209 REM------------ U0$ - 47 Zeichen, U1$ - 53 Zeichen ---------
210 U0$="12345678901234567890123456789012345678901234567"
220 U1$="12345678901234567890123456789012345678901234567890123"
230 U0=PEEK (VARPTR(U0$)+1)+256*PEEK (VARPTR(U0$)+2)
240 U1=PEEK (VARPTR(U1$)+1)+256*PEEK (VARPTR(U1$)+2)
250 DEFUSR0=U0+(U0>32767)*65536: DEFUSR1=U1+(U1>32767)*65536
260 FOR U0=U0 TO U0+46
270 READ Y:POKE U0+(U0>32767)*65536,Y: NEXT U0
280 FOR U1=U1 TO U1+52
290 READ Y:POKE U1+(U1>32767)*65536,Y: NEXT U1
299 REM----------- Kasten zeichnen ----------------------------
300 CLS:PRINT@ZP-63,"123456789";
310 PRINT@ZP,CHR$(156);STRING$(9,140);CHR$(172);
320 FOR ZZ=1 TO 9
330 PRINT@ZP-3+ZZ*64,ZZ;CHR$(149);BL$;CHR$(170);
340 NEXT ZZ
350 PRINT@ZP+640,CHR$(141);STRING$(9,140);CHR$(142);
359 REM----------- Zeichen darstellen -------------------------
360 PRINT@ZP+704,CHR$(CD),CD;:GOSUB 10000
369 REM----------- Menue schreiben ----------------------------
370 ON ERROR GOTO 0:GOSUB 11000
380 PRINT@TP,"E - Zeichen editieren";
390 PRINT@TP+64,"S - Zeichensatz sichern";
400 PRINT@TP+128,"L - Zeichensatz laden";
410 PRINT@TP+192,"X - Zeichensatz übertragen";

420 PRINT@TP+256,"D - Zeichensatz";
430 PRINT@TP+320,"    an Drucker übergeben";
440 PRINT@TP+384,"Z - Ein Zeichen";
450 PRINT@TP+448,"    an Drucker übergeben";
460 PRINT@TP+512,"K - ROM-Zeichensatz als DLC";
470 PRINT@TP+576,"P - DLC-Zeichensatz drucken";
480 PRINT@TP+640,"A - DLC-Zeichensatz ein";
489 REM----------- Befehlseingabe und -ausführung -------------
490 CP=TP+704:GOSUB 13000:I$=LEFT$(IN$,1)
500 IF I$="B" OR I$="b" THEN CLS:END
510 IF I$="<" THEN IF CD=32 THEN 370 ELSE CD=CD-1:GOTO 360
520 IF I$=">" THEN IF CD=126 THEN 370 ELSE CD=CD+1:GOTO 360
530 IF I$="E" OR I$="e" THEN 1000
540 IF I$="K" OR I$="k" THEN 2000
550 IF I$="Z" OR I$="z" THEN 3000
560 IF I$="D" OR I$="d" THEN 4000
570 IF I$="S" OR I$="s" THEN 5000
580 IF I$="X" OR I$="x" THEN 6000
590 IF I$="L" OR I$="l" THEN 6010
600 IF I$="P" OR I$="p" THEN 7000
610 IF I$="A" OR I$="a" THEN 8000
619 REM----------- Code oder Zeichen ? ------------------------
620 IF LEN(IN$)<2 OR LEN(IN$)>3 THEN 490
630 CC=CD: IF I$=CHR$(34) THEN CD=ASC(MID$(IN$,2,1)): GOTO 650
640 IF I$<"1" OR I$>"9" THEN 490 ELSE CD=VAL(IN$)
650 IF CD>31 AND CD<127 THEN 360 ELSE CD=CC
659 REM----------- Fehlerrücksprung ---------------------------
660 PRINT@TP+768,"Fehler!";
670 FOR Y=1 TO 200:NEXT Y:GOTO 370
999 REM=========== E - Zeichen editieren =====================
1000 I$=MID$(IN$,2,1): CC=CD
1010 IF I$="." THEN 1060
1020 IF I$="<" THEN IF CD=32 THEN 1060 ELSE CD=CD-1:GOTO 1060
1030 IF I$=">" THEN IF CD=126 THEN1060 ELSE CD=CD+1:GOTO 1060
1040 IF LEN(IN$)<3 OR LEN(IN$)>4 THEN 660
1050 IF I$=CHR$(34) THEN CD=ASC(MID$(IN$,3,1))
                    ELSE CD=VAL(RIGHT$(IN$,LEN(IN$)-1))
1060 IF CD<32 OR CD>126 THEN CD=CC: GOTO 660
1069 REM----------- Untermenue schreiben ----------------------
1070 GOSUB 11000
1080 PRINT@TP,"Pfeiltasten -";
1090 PRINT@TP+64,"              Cursorsteuerung";
1100 PRINT@TP+128,"Leertaste - Punkt löschen";
1110 PRINT@TP+192,"* - Punkt setzen";
1120 PRINT@TP+256,"CLEAR - Alle Punkte löschen";
```



```
1130 PRINT@TP+320,"@ - Alle Punkte setzen";
1140 PRINT@TP+384,"Z - Zeichen einsetzen";
1150 PRINT@TP+448,"O - Editieren abbrechen";
1160 PRINT@TP+512,"ENTER - Zeichen abspeichern";
1169 REM----------- Zu editierendes Zeichen darstellen --------
1170 PRINT@ZP+704,CHR$(CD),CD;:GOSUB 10000
1179 REM----------- Befehlseingabe und -ausführung ------------
1180 Z=0:S=0:CP=PO
1190 CU=PEEK(15360+PO+Z+S):GOSUB 12000:PRINT@PO+Z+S,CHR$(CU);
1200 IF I$=":" THEN I$="*"
1210 IF I$="O" OR I$="q" THEN 360
1220 IF I$=" " OR I$="*" THEN PRINT@PO+Z+S,I$;:GOTO 1190
1230 IF I$="@" THEN CC=CD:CD=128:GOTO 1320
1240 IF I$=CHR$(31) THEN CC=CD:CD=127:GOTO 1320
1250 IF I$<>"Z" AND I$<>"z" THEN 1330
1259 REM----------- Zeichen einsetzen -------------------------
1260 CC=CD:CP=ZP+768:GOSUB 13000
1270 IF LEN(IN$)<2 OR LEN(I$)>3 THEN 1300
1280 IF LEFT$(IN$,1)=CHR$(34)
      THEN CD=ASC(RIGHT$(IN$,LEN(IN$)-1))
      ELSE CD=VAL(IN$)
1290 IF CD>31 AND CD<127 THEN PRINT@ZP+768,BL$;BL$;:GOTO 1320
1300 CD=CC:PRINT"Fehler!";:FOR Y=1 TO 200:NEXT Y
1310 PRINT@ZP+768,BL$;BL$;:GOTO 1260
1320 GOSUB 10000:CD=CC:GOTO 1180
1330 IF I$=CHR$(9) THEN IF S=8 THEN 1190 ELSE S=S+1:GOTO 1190
1340 IF I$=CHR$(8) THEN IF S=0 THEN 1190 ELSE S=S-1:GOTO 1190
1350 IF I$=CHR$(10) THEN IF Z=512 THEN 1190 ELSE Z=Z+64:GOTO 1190
1360 IF I$=CHR$(91) THEN IF Z=0 THEN 1190 ELSE Z=Z-64:GOTO 1190
1370 IF I$<>CHR$(13) THEN 1190
1379 REM----------- Zeichen prüfen ----------------------------
1380 FE=0:GOSUB 11000:I=0
1390 FOR Z=0 TO 8:FOR S=0 TO 7
1400 IF PEEK(15360+PO+(Z*64)+S)=42
     AND PEEK(15360+PO+(Z*64)+S+1)=42
          THEN PRINT@TP+I,Z+1;",";S+1;:I=I+64:FE=-1:
               IF I>924 THEN GOSUB 11000:I=0
1410 NEXT S:NEXT Z
1419 REM----------- Unter- und Oberl(nge ? -------------------
1420 IF BZ$(7)=BL$ AND BZ$(8)=BL$
     THEN 1430
     ELSE IF NOT(BZ$(0)=BL$ AND BZ$(1)=BL$)
            THEN PRINT@TP+I," Unterl(nge ??";:FE=-1:I=I+64:
                 IF I>924 THEN GOSUB 11000:I=0
1430 IF NOT FE THEN 1470
1440 PRINT@TP+I," Trotzdem abspeichern ?";
1450 I$=INKEY$:IF I$="" THEN 1450
1460 IF I$<>CHR$(13) THEN 360
1469 REM----------- Zeichen abspeichern ------------------------
1470 FOR ZZ=0 TO 8:Z$(CD-32,ZZ)=BZ$(ZZ):NEXT ZZ: GOTO 370
1999 REM=========== K - ROM-Zeichensatz als DLC ==============
2000 GOSUB 14000:IF FE THEN 370
2010 LPRINT CHR$(27);"*";:D=0:GOSUB 15000:GOTO 370
2999 REM=========== Z - Ein Zeichen an Drucker übergeben ======
3000 GOSUB 14000:IF FE THEN 370
3010 I$=MID$(IN$,2,1):IF I$="." THEN 3050
3020 IF LEN(IN$)<3 THEN 660
3030 IF I$=CHR$(34) THEN CD=ASC(MID$(IN$,3,1))
                    ELSE CD=VAL(RIGHT$(IN$,LEN(IN$)-1))
3040 IF CD<32 OR CD>126 THEN 660

3050 CC=CD:GOSUB 4030:CD=CC: GOTO 360
3999 REM=========== D - Zeichensatz an Drucker übergeben ======
4000 GOSUB 14000:IF FE THEN 370
4010 CC=CD:FOR CD=32 TO 126:PRINT@ZP+704,CHR$(CD),CD;
4020 GOSUB 4030:NEXT CD:CD=CC:GOTO 360
4029 REM----------- Ein DLC-Zeichen kodieren und übergeben -----
4030 GOSUB 10000:Z=USR1(VARPTR(P(0)))
4040 LPRINT CHR$(27);"*";CHR$(1);CHR$(CD);
4050 FOR I=0 TO 9:D=P(I):GOSUB 15000: NEXT I:RETURN
4999 REM=========== S - Zeichensatz speichern ================
5000 IF LEN(IN$)<3 THEN 660
5010 IF MID$(IN$,2,1)<>"=" THEN 660
5020 NM$=MID$(IN$,3,LEN(IN$)-2)+"/DLC"
5030 ON ERROR GOTO 17000
5039 REM----------- File eröffnen, Pointer auf Buffer ---------
5040 OPEN "R",1,NM$:FIELD 1, 1 AS P$
5050 Y=0:P=PEEK(VARPTR(P$)+1)+256*PEEK(VARPTR(P$)+2)
5060 CC=CD:FOR CD=32 TO 126
5069 REM----------- Zeichen kodieren --------------------------
5070 PRINT@ZP+704,CHR$(CD),CD;:GOSUB 10000:Z=USR1(VARPTR(P(0)))
5079 REM----------- Kodiertes Zeichen in Buffer schreiben -----
5080 FOR I=0 TO 9:IF Y=256 THEN PUT 1:Y=0
5090 POKE P+Y,P(I):Y=Y+1:NEXT I
5100 NEXT CD: PUT 1: CLOSE: CD=CC: GOTO 360
5999 REM=========== X - Zeichensatz von Disk auf Drucker ======
6000 GOSUB 14000:IF FE THEN 370 ELSE XF=-1
6009 REM=========== L - Zeichensatz laden ====================
6010 IF LEN(IN$)<3 THEN 660
6020 IF MID$(IN$,2,1)<>"=" THEN 660
6030 NM$=MID$(IN$,3,LEN(IN$)-2)+"/DLC"
6040 ON ERROR GOTO 17000
6049 REM----------- File eroffnen, Pointer auf Buffer ---------
6050 OPEN "I",1,NM$:CLOSE:OPEN "R",1,NM$:FIELD 1, 1 AS P$
6060 Y=256:P=PEEK(VARPTR(P$)+1)+256*PEEK(VARPTR(P$)+2)
6070 CC=CD:FOR CD=32 TO 126:PRINT@ZP+704,CHR$(CD),CD;
6079 REM----------- Kodiertes Zeichen aus Buffer lesen --------
6080 FOR I=0 TO 9:IF Y=256 THEN GET 1:Y=0
6090 P(I)=PEEK(P+Y):Y=Y+1:NEXT I
6099 REM----------- Zeichen an Drucker übergeben (X) ----------
6100 IF XF THEN GOSUB 4040:GOTO 6140
6109 REM----------- oder dekodieren (L) -----------------------
6110 FOR ZZ=0 TO 8:PRINT@PO+(ZZ*64),BL$;:NEXT ZZ
6120 Z=USR0(VARPTR(P(0)))
6130 FOR ZZ=0 TO 8:Z$(CD-32,ZZ)=BZ$(ZZ):NEXT ZZ
6140 NEXT CD:CLOSE:XF=0:CD=CC:GOTO 360
6999 REM=========== P - DLC-Zeichensatz drucken ==============
7000 GOSUB 14000:IF FE THEN 370
7010 LPRINT CHR$(27);"@";CHR$(27);"E"
7020 I=31:GOSUB 7040:I=79:GOSUB 7040
7030 LPRINT CHR$(27);"@":GOTO 370
7038 REM----------- Eine Zeile drucken ------------------------
7039 REM----------- Standard-Zeichen drucken -----------------
7040 LPRINT CHR$(27);CHR$(36);:D=0:GOSUB 15000
7050 FOR Y=I TO I+47:LPRINT CHR$(Y);:NEXT Y
7059 REM----------- DLC-Zeichen drucken ----------------------
7060 LPRINT:LPRINT CHR$(27);CHR$(36);CHR$(1);
7070 FOR Y=I TO I+47:LPRINT CHR$(Y);:NEXT Y
7080 LPRINT:LPRINT:RETURN
7999 REM=========== A - DLC-Zeichensatz aktivieren ==========
8000 GOSUB 14000:IF FE THEN 370
8010 LPRINT CHR$(27);CHR$(36);CHR$(1);:GOTO 370
9998 REM

9999 REM----------- UP: Zeichen darstellen -------------------
10000 FOR ZZ=0 TO 8
10010 PRINT@PO+(ZZ*64),Z$(CD-32,ZZ);
10020 NEXT ZZ:RETURN
10999 REM----------- UP: Menue löschen ----------------------
11000 FOR ZZ=0 TO 14
11010 PRINT@TP+ZZ*64,BL$;BL$;BL$;
11020 NEXT ZZ:RETURN
11999 REM----------- UP: Eingabezeichen holen ----------------
12000 PRINT@CP+Z+S,CHR$(191);:I$=INKEY$:
      IF I$<>""
      THEN RETURN
      ELSE FOR Y=1 TO 50:NEXT Y:PRINT@CP+Z+S,CHR$(CU);:
           I$=INKEY$:
           IF I$<>"" THEN RETURN
                     ELSE FOR Y=1 TO 50:NEXT Y:GOTO 12000
12999 REM----------- UP: Zeile einlesen ---------------------
13000 PRINT@CP,"?            ";:Z=0:S=1:IN$=""
13010 CU=PEEK(15360+CP+S):GOSUB 12000:PRINT@CP+Z+S,CHR$(CU);
13020 IF I$=CHR$(8)
      THEN IF S=1
           THEN 13010
           ELSE S=S-1:PRINT@CP+S,"  ";:
                IN$=LEFT$(IN$,LEN(IN$)-1):
                GOTO 13010
13030 IF I$=CHR$(24) THEN 13000
13040 IF I$=CHR$(13) THEN RETURN
13050 IF I$<" " OR I$>CHR$(126) THEN 13010
13060 IF S=11 THEN 13010
13070 IN$=IN$+I$:PRINT@CP+S,I$;:S=S+1:GOTO 13010
13999 REM----------- UP: Drucker bereit ? -------------------
14000 FE=0:IF PEEK(14312)=63 THEN RETURN
14010 PRINT@TP+768," Drucker nicht bereit !";
14020 IF INKEY$=CHR$(13) THEN FE=-1: RETURN
14030 IF PEEK(14312)<>63 THEN 14020
14040 PRINT@TP+769,BL$;BL$;BL$;:RETURN
14999 REM----------- UP: Byte auf Druckerport geben ----------
15000 IF PEEK(14312)<>63 THEN 15000
15010 IF PEEK(4)=6 THEN POKE 14312,D ELSE OUT 248,D
15020 RETURN
15999 REM----------- Bytes für USR-Routinen -----------------
16000 DATA 205,127, 10,235, 33,205, 60, 26,183, 40,  3, 33
16010 DATA  77, 61,  6,  9,197, 19, 19,  1,  1,  7, 26,161
16020 DATA  40,  2, 54, 42,197,175, 71, 14, 64,  9,193,203
16030 DATA   1, 16,239,  1, 65,254,  9,193, 16,226,201,205
16040 DATA 127, 10,235, 33,205, 60, 26,183, 40,  3, 33, 77
16050 DATA  61,  6,  9,197, 19, 19,  1,  1,  7,175,245,126
16060 DATA 254, 42, 32,  3,241,177,245,197,175, 71, 14, 64
16070 DATA   9,193,203,  1, 16,237,241, 18,  1, 65,254,  9
16080 DATA 193, 16,220,201
16999 REM----------- ON ERROR Routine ----------------------
17000 IF ERR<45 THEN ON ERROR GOTO 0:END
17010 CLOSE
17020 IF ERR=106 THEN IN$="File existiert nicht !":GOTO 17070
17030 IF ERR=122 THEN IN$="Diskette voll !":GOTO 17070
17040 IF ERR=128 THEN IN$="Filename nicht zul(ssig !":GOTO 17070
17050 IF ERR=134 THEN IN$="Diskette schreibgeschützt !":GOTO 17070
17060 IN$="Fehler Nummer :"+STR$((ERR/2)+1)
17070 PRINT@TP+768,IN$;
17080 IF INKEY$="" THEN 17080 ELSE RESUME 370
```

Die Suche nach dem Schlüssel von Nightshade

Die Hauptsache von "Die Suche nach dem Schlüssel von Nightshade" ist folgende:

1) Der Abenteurer muß versuchen die Größe, und Stärke seiner Armee zu vergrößern.

2) Der Abenteurer muß 5 Stücke eines Schlüßels, die Zerstreut im Lande Nightshade herumliegen sich erkämpfen.

Der Bildschirm

Der Bildschirm ist in 5 Sektionen unterteilt. Ganz oben wird die Aufgabe des Spielers angezeigt. Ganz links wird die Größe und Stärke der Armee angezeigt. In der Mitte des Bildschirm's wird eine Teilansicht vom Land, und der Standtpunkt des Spielers, angezeigt. Die Unterste Sektion wird für Mitteilungen gebraucht. Die Sektion Rechts zeigt den Spieler Wahlmöglichkeiten an.

Die Karte

Das Land ist in 30 mal 30 Einheiten unterteilt.

Jede Einheit ist eine mögliche Stellung, die Du in Besitz nehmen kannst.

Der Bildschirm zeigt einen Kleinen Teil der Karte mit Dir in der Mitte.

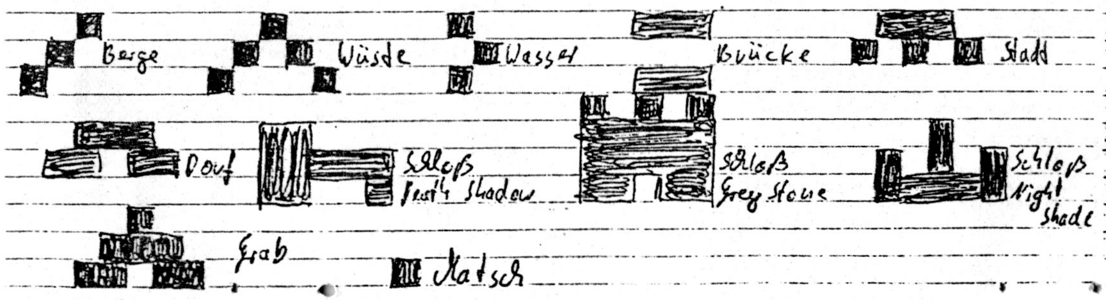
Die Karte stellt das Land um vier Einheiten in jede Richtung dar.

Wenn Du Dich Bewegst, stehst Du selbst still, und nur die Karte Bewegt sich unter Dir.

Kleine Grafik-Symbole auf der Karte, symbolisieren Dir die Landschaft.

Landstriche, Flüsse, Seen, Berge und Wüsten sind in jedem Spiel an der Gleichen Stelle. Menschenbezügliche Objekte wie: Brücken, Städte, Dörfer, Schlößer und Gräber varrieren von Spiel zu Spiel.

Tabelle 1: Symbole



Marktplatz

Auf dem Marktplatz bekommst Du alles was du für Dich, und Deine Armee, brauchst.

Marktplätze findest Du überall dort wo Menschen sind.

Dinge kosten in kleinen Dörfern mehr als in Städten. Wenn Du etwas kaufst, kaufst Du für die Ganze Armee.

Waffen und Rüstungen

Waffen und Rüstungen machen eine Armee Stark und Mächtig.

Je mehr Waffen eine Armee besitzt, je mehr Kraft hat sie.

Lebensmittel, Wasser, Arznei

Lebensm., Wasser und Arznei sind im Spiel unentbehrlich, Du brauchst sie um Deine Armee am Leben zu halten.

Wenn Du Arznei nimmst, geht Deine Erschöpfung auf 0 % zurück.

Erschöpfung

Wenn Du Dich bewegst oder kämpfst, wirst Du Erschöpft. Dieser Faktor wird ganz links in % angezeigt. Ist die % Rate = 100, beginnt die Armee langsam dahin zu siechen. Du stirbst immer zuletzt.

Begegnungen

Wenn Du unterwegs jemanden Begegnest. Töte sie denn Sie tragen Schätze, Lebensmittel oder... Schlüßelstücke.

Schlößer und Gräber

In Schlößern und Gräbern findest Du Schätze und Schlüßelstücke.

Schloß Nightshade

Schloß Nightshade kann nur mit dem Schlüßel geöffnet werden.

Eine Armee von Feuerdrachen Bewacht dieses Schloß.

Nur wenn Du sie tötest, und den König des Landes Befreit hast, ist das Spiel beendet.

So nun Viel Spass mein Adventurer wünscht euch [illegible]

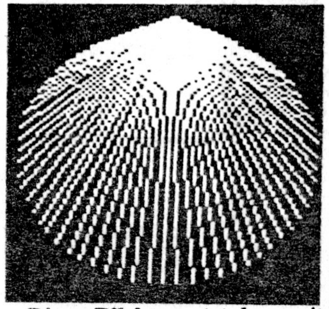
Basic-Dialekte im Vergleich – Teil 2

In Ausgabe 18 begannen wir einen Vergleich der Basic-Versionen der beliebtesten Heimcomputer. Auch bei den Befehlen für die hochauflösende Grafik gibt es etliche Unterschiede.

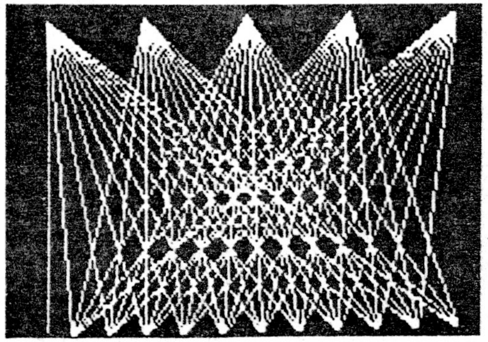
Entscheidendes Kriterium für den Kauf eines Heimcomputers war bis vor kurzem noch die Speicher-Kapazität. Der ZX80 bot nur 1 KByte RAM, der VC 20 immerhin das fünffache. Diese Zeiten sind vorbei. Denn ein Heimcomputer bietet heute mindestens 16 KByte RAM. Mit ein Punkt bei der Entscheidung für oder gegen einen neuen Heimcomputer sind heute die grafischen Fähigkeiten. Nach höherer Bildschirmauflösung, mehr Farben und besserer Grafik-Software bei möglichst niedrigeren Preisen ist gefragt. 64000 oder gar 256000 Grafikpunkte auf einem Bildschirm sind schon keine Seltenheit mehr und ein Ende dieser Entwicklung ist nicht abzusehen. Der Mitsubishi DC-186 zum Beispiel bringt es auf über eine Million einzeln ansteuerbarer Pixel. Für dieses Gerät müssen etwa 17000 Mark bezahlt werden, aber auch preiswertere Geräte leisten schon Erstaunliches. Schon für weniger als 1000 Mark gibt es Computer zu kaufen, deren Grafikfähigkeiten wie Auflösung, Anzahl der Farben und Software-Unterstützung einiges bieten. Bei der Erstellung eigener Grafik-Programme lassen sich fast alle Heimcomputer mit Basic in der Grundversion in zwei Gruppen einteilen: die »Poke«-Gruppe und die Grafik-Basic-Gruppe. Zur ersten Gruppe, die Grafiken nur mit entsprechenden Toolkits oder über komplizierte POKE-Routinen erzeugen, zählen zum Beispiel alle Commodore-Computer und der TI 99/4A.

Auch bei Grafikbefehlen: Gleicher Name, andere Funktion

Die meisten Heimcomputer erlauben den Einsatz von Basic-Befehlen wie »Line«, »Set« und »Circle«, um Grafiken zu erzeugen. Bei diesen Befehlen gibt es aber bei jedem Heimcomputer eine andere Syntax oder gar ein anderes Befehlswort. Der »Line«-Befehl dient im allgemeinen dazu, eine Linie zwischen zwei Punkten zu zeichnen. Beim Spectrum wird der »Line«-Befehl nur mit den Befehlen »Save« und »Input« eingesetzt und hat dort ganz andere Funktionen. Um bei diesem, wie auch bei anderen Computern (siehe Übersicht auf Seite 129), eine Linie zu ziehen, ist der Befehl »Draw« einzusetzen. Für Computer, die diesen Befehl nicht eingebaut haben, kann das Programm aus Listing 1 eingesetzt werden. Die allgemeine Syntax des »Line«-Befehls ist: Line (X,Y)-(V,W),pl,p2. Mit den Variablen X und Y wird ein Punkt des Grafikbildschirms definiert, von dem aus eine Linie zum Punkt V,W in der Farbe pl gezogen werden soll. Die Variable p2 kann bei den Computern Dragon 32 oder Dragon 64 »B« oder »BF« sein. Durch den Einsatz von »B« interpretiert das System die Punkte X,Y und V,W als gegenüberliegende Ecken eines Rechtecks, das mit der Option »BF« ausgemalt wird. Bei anderen Computern als denen von Dragon sollten diese Parameter fortgelassen werden, da sie eine Fehlermeldung hervorrufen können. Der »Pset (X,Y,F)«-Befehl setzt einen Punkt mit den Koordinaten X,Y in der Farbe F. In anderen Basic-Dialekten lautet das entsprechende Befehlswort »Plot«. Etwas komplizierter ist da schon der Einsatz des Befehls »Circle«, der einen Kreis erzeugt. Die allgemeine Syntax hierzu lautet : Circle (X,Y),R,F,HB,A,E. Wieder geben die Variablen X und Y einen Punkt auf dem Bildschirm an. Dies ist der Mittelpunkt des Kreises, der mit dem Radius R in der Farbe F gezeichnet wird. Hier sind die Möglichkeiten der meisten Heimcomputer erschöpft. Die Syntax der Dragon-Computer erlaubt noch drei weitere Parameter. Die Variable HB steht für den Quotienten Höhe/Breite, mit dem man die Exzentizität für das Zeichnen einer Ellipse definiert. Mit A und E wie Anfang und Ende wird vereinbart, ob nur Kreis- oder Ellipsenbögen gezeichnet werden sollen. Diese Eingaben sind nicht erforderlich, bieten aber eine einfache Möglichkeit, komplizierte Teilkreise zu zeichnen. Die Werte für Anfang und Ende beziehen sich auf den 360-Grad-Vollkreis und geben den zu zeichnenden Ausschnitt an. Auch beim Spectrum kann man einen Teilkreis erzeugen, indem an die »Draw«-Anweisung ein dritter Parameter angehängt wird. Heimcomputer, die den Befehl »Circle« nicht haben, dafür aber »Plot« oder »Set« beziehungsweise »Pset«, können den Befehl durch ein kleines Programm simulieren (Listing 2). Für grafische Darstellungen muß bei den meisten Heimcomputern der Bildschirmmodus umgeschaltet werden. Die Befehle dazu heißen »Pmode«, »Mode«, »Fgr«, »Graphics« oder »Hgr«. Dabei sind, je nach Computertyp, unterschiedliche Parameter einzusetzen.



Diese Effekte entstehen mit dem Programm aus Listing 7, obwohl nur Linien gezeichnet werden



◄ Auch bei diesem Beispiel sind nur »Plot«-Befehle eingesetzt worden

```
1 ' ziehe Linie von
2 ' der Koordinate
3 ' A,A zu der
4 ' Koordinate A,B
10 FOR P = A TO B
20 SET A,P
30 NEXT P
```

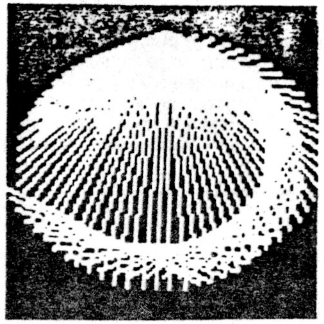
Listing 1. Mit dieser Befehlsfolge wird der »Plot to«- oder »Draw«-Befehl simuliert

Auch mit einfachen Befehlen sind Grafiken zu erzeugen

Alle Beispielprogramme sind auf den Dragon-Computern 32 und 64 ohne Änderungen lauffähig. Für andere Typen müssen die jeweils angegebenen Befehle ersetzt werden.

Wer noch keine Erfahrungen mit der Programmierung von Grafiken hat, sollte die Beispiele aus Listing 4,5 und 6 ausprobieren. Mit dem Programm aus Listing 4 wird ein Punkt kurzzeitig ein- und ausgeschaltet, indem die Befehle »Set« und »Reset« eingesetzt werden. Für andere Computer werden diese Befehle durch »Plot« beziehungsweise »Unplot« ersetzt. Das Programm aus Listing 5 setzt den »Line«-Befehl ein, um eine Linie und ein Rechteck zu zeichnen. In anderen Basic-Dialekten ist dieser Befehl durch »Draw« oder »Plot to« zu ersetzen. Für Computer, die nur die Befehle »Set« und »Reset« haben, kann das Programm aus Listing 3 eingesetzt werden. Das Programm aus Listing 6 schließlich stellt Kreise mit unterschiedlichen Höhe/Breite-Verhältnissen dar. Wer durch diese kleinen Beispiele noch nicht genügend motiviert ist, sich mehr mit Computergrafik zu beschäftigen, sollte das Programm aus Listing 7 eingeben. Kein Bild und keine Erklärung kann die Effekte, die beim Ablauf dieses Programms entstehen, erschöpfend darstellen. Eine Linie wandert über einen imaginären Kreis und dreht sich dabei um ihren Mittelpunkt. Dadurch entstehen sogenannte Moirè-Bilder. Da nur die Befehle zum Zeichnen einer Linie und zum Setzen eines Punktes notwendig sind, ist das Programm leicht auf jeden Heimcomputer umzusetzen.

Die Beispiele zeigten, wie einfach Grafiken entstehen können. Auch die Übersetzung eines Basic-Dialekts in einen anderen, fällt nicht schwer. Gleiche Befehle haben nur unterschiedliche Namen, die Funktion bleibt die Gleiche. Schwieriger gestaltet sich schon die Umsetzung eigener Ideen in entsprechende Programme. Das ist aber gar nicht so kompliziert, wie wohl mancher denken wird.



▲ Das Programm aus Listing 7 durchläuft einige Schleifen. So entstehen immer neue Bilder.

```
1 ' Mit dieser Befehlsfolge
2 ' können Computer, die
3 ' den "Circle"-Befehl
4 ' nicht kennen, einen
5 ' Kreis darstellen.
10 ' R = Radius
20 ' CHOR = Cursor horizontal
30 ' CVER = Cursor vertikal
40 ' PI = Kreiszahl (3.145)
50 FOR A = 0 TO 2*PI STEP PI/R
60 X = R * COS A + CHOR
70 Y = R * SIN A + CVER
80 SET X,Y : 'oder PLOT x,y
90 NEXT A
91 ' Der Schnittpunkt von
92 ' CHOR und CVER ist der
93 ' Mittelpunkt des Kreises
```

► Listing 2. Auch Computer ohne den »Circle«-Befehl können Kreise zeichnen

```
10 ' Befehlsfolge um ein
20 ' Rechteck zu zeichnen
40 ' K = Kantenlänge
50 ' A = Eckpunkt oben links
60 ' B =    ''    oben rechts
70 ' C =    ''    unten links
80 ' D =    ''    unten rechts
90 ' jeweils die Werte der
100 ' X- Achse sind anzugeben
110 ' BY- und CY- Werte müssen
111 ' angegeben werden.
120 FOR Z = A TO A + K
130 SET Z,A : NEXT Z
140 FOR Z = BY TO BY + K
160 SET B,Z : NEXT Z
170 FOR Z = D TO D-K STEP -1
180 SET Z,D : NEXT Z
190 FOR Z = CY TO CY - 10 STEP -1
200 SET C,Z : NEXT Z
```

◄ Listing 3. Dieses Programm stellt ein Rechteck dar

Der Bildschirm als Koordinatensystem

Der Bildschirm, oder besser der Grafikbildschirm, ist in Zeilen und Spalten aufgeteilt. Zeilen liegen horizontal und Spalten vertikal auf dem Schirm. Jeder Schnittpunkt dieser beiden Linien wird als Punkt oder Pixel bezeichnet und kann beeinflußt werden. Bei den meisten Heimcomputern ist die obere linke Ecke mit den Koordinaten 0,0 bezeichnet. Von ihr aus werden die Spalten und Zahlen fortlaufend gezählt, abhängig von der höchsten Auflösung. Beim Dragon 64 zum Beispiel lautet die größte Zeilennummer 192 und die höchste Spaltennummer 256. Dadurch wird die unterste rechte Ecke definiert. Halbiert man diese beiden Zahlen, so erhält man die Werte 128 und 96, wodurch die Bildschirmmitte bezeichnet sein sollte. Eine Überprüfung dieser Berechnung mit dem »Set«-Befehl wird dieses Ergebnis bestätigen. Mit dem Punkt P=(128,96) hat man einen Koordinatenpunkt bestimmt, wobei der erste Wert die X-Achse und der zweite Wert die Y-Achse angibt. Bei allen Befehlen, die diese Koordinatenschreibweise einsetzen, muß man darauf achten, daß jeder Wert im zulässigen Bereich liegt. Für die X-Achse liegt dieser in unserem Beispiel zwischen 0 und 256, für die Y-Achse sind Werte von 0 bis 192 zulässig. Nicht jeder Basic-Interpreter läßt bei Grafikbefehlen den Einsatz von Gleitkommazahlen zu. Deshalb sollte man bei allen Berechnungen nur Integer-Variablen anwenden.

```
10 '**DEMO/PSET,PRESET**
15 '********************************
20 PM=4          '*   Auflösungsgrad*
25 SC=1          '*       Farbgruppe*
30 HF=0          '* Hintergrundfarbe*
35 X=128         '*     X-Koordinate*
40 Y=96          '*     Y-Koordinate*
45 F=5           '*     Zeichenfarbe*
50 WA=200        '*    Warteschleife*
55 '********************************
60 PMODEPM,1:SCREEN1,SC:PCLSHF
65 FOR I=0 TO WA: NEXT I
70 PSET(X,Y,F)
75 FOR I=0 TO WA: NEXT I
80 PRESET(X,Y)
85 GOTO 65
```

◄ *Listing 4. Blinkender Punkt*

```
10 '**DEMO/LINE**
15 '****************************
20 PM=4    '*     Auflösungsgrad*
25 SC=1    '*         Farbgruppe*
30 HF=0    '*   Hintergrundfarbe*
35 X=128   '*X-Koordinate von P1*
40 Y=96    '*Y-     ''         P1*
45 V=200   '*X-     ''         P2*
50 W=125   '*Y-     ''         P2*
55 WA=800  '*      Warteschleife*
60 '****************************
65 PMODEPM,1:SCREEN1,SC:PCLSHF
70 FOR I=0 TO WA:NEXT I
75 LINE(X,Y)-(V,W),PSET
80 FOR I=0 TO WA:NEXT I
85 LINE(X,Y)-(V,W),PRESET
90 FOR I=0 TO WA:NEXT I
95 LINE(X,Y)-(V,W),PSET,B
100 FOR I=0 TO WA:NEXT I
105 LINE(X,Y)-(V,W),PRESET,B
110 FOR I=0 TO WA:NEXT I
115 LINE(X,Y)-(V,W),PSET,BF
120 FOR I=0 TO WA:NEXT I
125 LINE(X,Y)-(V,W),PRESET,BF
130 GOTO 70
```

◄ *Listing 5. Durch »Pset, B« wird ein Rechteck dargestellt. Für andere Computer als den Dragon 32/64 muß das Programm aus Listing 3 eingesetzt werden.*

❝Das Wenn und das Aber sind das Erbteil der Dummen.❞

Italienisches Sprichwort

❝Unsere Prinzipien dauern gerade so lange, bis sie mit unseren Leidenschaften oder Eitelkeiten in Konflikt kommen, und ziehen dann jedesmal den kürzeren.❞

Theodor Fontane,
deutscher Schriftsteller (1819–1898)

ichen

Ohne Mathematik geht es auch bei Grafik nicht

Solange Linien parallel zum Bildrand gezeichnet werden, erfordert die Bestimmung der Koordinaten keine große Aufgabe. Für ein Rechteck sind vier Linien zu definieren, je zwei parallel zu den horizontalen und zwei zu den vertikalen Bildschirmrändern. Davon abweichende Lagen eines Rechtecks, wie zum Beispiel ein auf der Spitze stehendes Quadrat, müssen über Formeln berechnet werden. In diesen Fällen reicht es aus, die Eckpunkte zu bestimmen. Jeweils zwei Eckpunkte liegen mit dem Mittelpunkt des Quadrats auf einer Geraden. Das heißt aber auch, daß die X-Koordinaten der Punkte A,C und M gleich sind, wie auch die Y-Koordinaten der Punkte B,D und M. Ausgehend von einer Auflösung von 256 x 192 Punkten haben wir also folgende Koordinaten festgelegt: M = (128,96), A = (128,y(A)), B = (x(B),96), C = (128, y(C)) und D = (x(D), 96). Da das Quadrat in der Mitte des Bildschirms gezeichnet wird, sind die Abstände der Eckpunkte zum Mittelpunkt konstant, nämlich genau halb so lang, wie die Länge der Diagonale (d) des Quadrats. Legen wir fest, daß die Kantenlänge aus 60 Pixeln besteht, errechnen sich die Abstände der Eckpunkte vom Mittelpunkt aus mit der Formel: 60 * SQR (2)/2. Aufgerundet ergibt das 43 Pixel, die nun, immer vom Mittelpunkt aus betrachtet, die Koordinaten der Eckpunkte ergeben: M = (128,96), A = (128,53), B = (85,96), C = (128,139) und D = (171,96).

Daß der Bildschirmmittelpunkt für Berechnungen von Grafiken zentrale Bedeutung hat, zeigt auch das nächste Beispiel. Auf dem Bildschirm soll ein gleichseitiges Dreieck gezeichnet werden. Die Dreiecksbasisseite soll dabei parallel zum unteren Bildschirmrand verlaufen. Wieder legen wir eine Seitenlänge von 60 Pixel fest. Da der Bildschirmmittelpunkt auch genau in der Mitte der Höhe (h) des Dreiecks liegt, berechnen sich die X-Koordinaten relativ einfach aus: x(B) = x(M) - 30 = 98 und x(C) = x(M) + 30 = 158. Der Mittelpunkt (M) hat die Koordinaten M = (128,96). Für die Punkte A, B und C ergibt sich also folgendes: A = (128, y(A)), B = (98, y(B)) und C = (158, y (C)). Anschließend wird die Dreieckshöhe (h) berechnet: h = 60 *SQR (3)/2. Das ergibt 52 und die Strecke (h/2) ist somit 26 Pixel lang. Um nun die richtigen Y-Werte zu finden, müssen diese 26 Pixel für den Punkt A von der Mittelpunktkoordinate abgezogen und für die Punkte B und C addiert werden. Die Koordinaten lauten also: A = (128,70), B = (98,122) und C = (158,122).

Spruch der Woche

Es kann passieren, was will: es gibt immer einen, der es kommen sah.

❝Ich erkenne die Männer an den Früchten, die sie bevorzugen. Apfelsinenesser werden gute Familienväter. Birnenesser schätzen Ruhe und Gemütlichkeit. Himbeeren verraten den Romantiker, Nüsseknacker sind sehr kompliziert. Vorsicht vor Männern, die grüne Früchte lieben. Am temperamentvollsten sind Grapefrucht-Freunde.❞

Martine Carol,
französische Filmschauspielerin
(1922–1967)

❝Die Teilung der Hand in Finger hat keine Schwächung herbeigeführt, vielmehr ihre Leistungsfähigkeit gesteigert. Ebenso erhöht der Staatsmann, der anderen einen Anteil an der Führung gibt, durch die gemeinsame Leistung den Erfolg seiner Tätigkeit.❞

Plutarch,
griechischer Philosoph und Historiker (50–125 n. Chr.)

Attraktive Grafiken durch Bewegung

Erweitert man die bisher besprochenen Programme durch eine oder mehrere For-Next-Schleifen, kann man Bewegungen simulieren. Die einfachste Bewegung ist wohl die eines Punktes, der über den Bildschirm läuft (Listing 8). Dabei wird ein Pixel ein- und ausgeschaltet, wobei der Wert der X-Koordinate immer um den Wert »eins« erhöht wird. Im Programm aus Listing 9 wird der Befehl »Line« eingesetzt, um immer größere Quadrate zu zeichnen. Dadurch entsteht der Eindruck, als ob sich ein langer Tunnel öffnen würde. Dieser Effekt wird

Befehl / Computer	Apple II	Atari	Colour Genie	Dragon 32/64	Laser/ VZ200	Sinclair ZX81	Sinclair Spectrum	TRS-80 Level II	Sharp MZ 700
Zieht eine Linie von (x,y) nach (v,w)	HPlot x,y to v,w	Draw to v,w. x,y vorher mit Plot definieren	Plot x,y to v,w	Linie (x,y) – (v,w), P set	siehe Listing 1	in Listing 1 Plot einsetzen	Draw v,w. x,y vorher mit Plot def.	siehe Listing 1	siehe Listing 1
Löscht eine Linie [illegible] (x,y) nach (v,w)	Auf Hintergrundfarbe umschalten. HPlot x,y to v,w	Auf Hintergrundfarbe umschalten und Draw to v,w eingeben	Nplot x,y to v,w	Linie (x,y – (v,w), Preset	in Listing 1 Reset einsetzen	in Listing 1 Unplot einsetzen	Over 1; Draw, v,w. x,y vorher mit Plot definieren	in Listing 1 Reset einsetzen	in Listing 1 Reset einsetzen
[illegible]	Plot x,y	Plot x,y	Plot x,y	Pset (x,y)	Set x,y	Plot x,y	Plot x,y	Set x,y	Set x,y
[illegible]	Plot x,y in Hintergrundfarbe	Plot x,y in Hintergrundfarbe	Nplot x,y	Preset (x,y)	Reset x,y	Unplot x,y	Plot; Over 1; x,y	Reset x,y	Reset x,y
[illegible]	siehe Listing 2	siehe Listing 2	Circle x,y,r	Circle (x,y),r	siehe Listing 2	siehe Listing 2	Circle x,y,r	siehe Listing 2	Circle x,y,r nur mit Plotter wirksam
[illegible]	Call FB 40 oder Gr, Hgr	Graphics (Mode)	FCLS	PCLS	CLS	CLS	CLS	CLS	CLS
[illegible]	In Listing 3 Set durch Plot ersetzen	In Listing 3 Set durch Plot ersetzen	Plot A to B to C to D to A	Linie (A,B)-(C,D), Pset,B	siehe Listing 3	In Listing 3 Set durch Plot ersetzen	Draw A,B: Draw B,C: Draw C,D: Draw D,A	siehe Listing 3	siehe Listing 3
[illegible]	GR mit 40x48 Punkten, 16 Farben HGR mit 280x160 Punkten, 8 Farben HGR2 mit 280x192 Punkten, 8 Farben	Graphics 3 mit 40x24 Punkten, 4 Farben, Graphics 4 mit 80x48 Punkten, 2 Farben Graphics 5 mit 80x48 Punkten, 4 Farben, Graphics 6 mit 160x96 Punkten, 2 Farben Graphics 7 mit 160x96 Punkten, 4 Farben, Graphics 8 mit 320x192 Punkten, 2 Farben	FGR mit 160x96 Punkten, 4 Farben (alte ROM's) 160x102 mit neuen ROM's	Pmode 0 mit 128x96 Punkten, 2 Farben; Pmode 1 mit 128x96 Punkten, 4 Farben; Pmode 2 mit 192 x 128 Punkten, 2 Farben; Pmode 3 mit 192x128 Punkten, 4 Farben; Pmode 4 mit 256x192 Punkten, 2 Farben	Mode 1 mit 128x64 Punkten, 8 Farben	64x44 Punkte, 2 Farben	256x192, 8 Farben	128x48, 2 Farben	80x50, 8 Farben

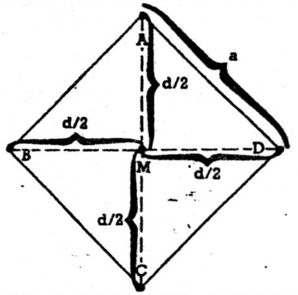
Übersicht der wichtigsten Grafikbefehle ▲

aber durch die fortlaufende Bewegung erzielt. Als stehendes Bild sieht man tatsächlich nur Quadrate.

In dieser Ausgabe haben wir die unterschiedlichen Grafikbefehle der gängigen Heimcomputer gegenübergestellt. Die Computer VC 20 und Commodore 64 fehlen in dieser Übersicht, da sie nur mit Toolkits wie zum Beispiel »Simon's Basic« Grafik erzeugen. Der nächste und letzte Teil der »Basic-Dialekte im Vergleich« wird sich mit den Systembefehlen beschäftigen.

(F.L.Moersdorf/rs)

▼ So berechnen sich die Koordinaten für ein auf der Spitze stehendes Quadrat.



Formel: $d = a * \sqrt{2}$

bei a = 60 Pixel

=> d/2 ≈ 43 Pixel

=> A = (128,53)
B = (85,96)
C = (128,239)
D = (171,96)

Zeichencodes in Computern

hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII
00	0	NUL	20	32	Space	40	64	@	60	96	'
01	1	SOH	21	33	!	41	65	A	61	97	a
02	2	STX	22	34	"	42	66	B	62	98	b
03	3	ETX	23	35	#	43	67	C	63	99	c
04	4	EOT	24	36	$	44	68	D	64	100	d
05	5	ENQ	25	37	%	45	69	E	65	101	e
06	6	ACK	26	38	&	46	70	F	66	102	f
07	7	BEL	27	39	'	47	71	G	67	103	g
08	8	BS	28	40	(	48	72	H	68	104	h
09	9	HT	29	41	)	49	73	I	69	105	i
0A	10	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	2D	45	–	4D	77	M	6D	109	m
0E	14	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	30	48	0	50	80	P	70	112	p
11	17	DC1	31	49	1	51	81	Q	71	113	q
12	18	DC2	32	50	2	52	82	R	72	114	r
13	19	DC3	33	51	3	53	83	S	73	115	s
14	20	DC4	34	52	4	54	84	T	74	116	t
15	21	NAK	35	53	5	55	85	U	75	117	u
16	22	SYN	36	54	6	56	86	V	76	118	v
17	23	ETB	37	55	7	57	87	W	77	119	w
18	24	CAN	38	56	8	58	88	X	78	120	x
19	25	EM	39	57	9	59	89	Y	79	121	y
1A	26	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	3B	59	;	5B	91	[	7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	¦
1D	29	GS	3D	61	=	5D	93	]	7D	125	}
1E	30	RS	3E	62	>	5E	94	↑	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL

Bedeutung der wichtigsten Steuerzeichen

NUL = ohne Wirkung, SOH = start of header, STX = start of text, ETX = end of text, EOT = end of transmission, ENQ = enquiry (Aufforderung der Gegenstation zum Senden), ACK = acknowledge (Bestätigung, Rückmeldung), DLE = data link escape (Umschalten auf eine andere Steuerzeichengruppe), DC = Device Control (DC1 = XON, DC3 = XOFF), NAK = negative acknowledge, SYN = Synchronisationszeichen, ETB = end of transmission block, VT = vertical tabulating (Cursor nach oben), HT = horizontal tab. (Cursor nach rechts), BS = Back Space (Cursor nach links), LF = Line feed (Cursor nach unten), BEL = bell (akustisches Zeichen, Klingel), CR = Carriage Return, FS = field seperator, GS = group seperator, US = unit separator, Space = Leerraum, DEL = Delete/Rub out.

Deutsche Sonderzeichen

Zeichen	IBM-PC u. ä. hex	IBM-PC u. ä. dez.	ISO-Norm hex	ISO-Norm dez.
§	15	21	40	64
Ä	8E	142	5B	91
Ö	99	153	5C	92
Ü	9A	154	5D	93
ä	84	132	7B	123
ö	94	148	7C	124
ü	81	129	7D	125
ß	E1	225	7E	126

Basic-Dialekte im Vergleich – Teil 3

Im abschließenden Teil der Basic-Befehlsübersicht befassen wir uns mit den Systembefehlen und Funktionen. Diese Befehlsgruppe ist für den ernsthaften Programmierer beim Kauf eines Heimcomputers entscheidend.

Mancher Programmierer wird es zu schätzen wissen, wenn in dem Basic seines Heimcomputers der Befehl »Trace« eingebaut ist. Hiermit wird während des Programmablaufs immer die aktuelle Zeilennummer angezeigt. Damit kann besonders bei komplexen Programmen die Arbeitsweise Schritt für Schritt verfolgt werden. Falsche Ergebnisse oder Reaktionen lassen sich so auf einfache Art und Weise erklären. Viele Heimcomputer aber bieten diesen Befehl nicht. Die Fehlersuche in Programmen gestaltet sich daher äußerst kompliziert. Mit diesem Befehl hat man ein Merkmal, das bei der Kaufentscheidung nur indirekt mit dem Preis zu tun hat, denn auch preiswertere Geräte haben diesen Befehl, bieten aber auf anderen Gebieten (Grafik- oder System-Befehle) weniger.

Ein weiterer nützlicher Befehl ist der »Auto«-Befehl, der bei der Eingabe eines Programms automatisch nach der Betätigung der »Return«-Taste die Zeilennummern um einen angegebenen Wert erhöht. Bei der Eingabe längerer Programme erspart man sich so das Eintippen der Zeilennummern und hat immer den gleichen Abstand zwischen den Zeilen. Sollte einmal eine Umnumerierung eines gesamten Programms erforderlich sein, lernt man schnell die Vorteile des »Renum«-Befehls kennen. Mit diesem Befehl werden die Abstände zwischen den Programmzeilen verändert. Hat man zum Beispiel ein Programm, bei dem der Abstand immer genau zehn Nummern beträgt, möchte nun aber ein Erweiterung von zwanzig Zeilennummern dazwischen schieben, so genügt es, das Programm mit dem »Renum«-Befehl auf den gewünschten Abstand zu bringen. Wurde die Änderung durchgeführt, kann natürlich wieder eine Umnumerierung erfolgen, zum Beispiel auf einen Abstand von zwei Zeilen. Der große Vorteil bei diesem Befehl ist, daß auch die Adressen der Sprünge (Goto, Gosub) mitverändert werden. Bei Computern, die diesen Befehl nicht kennen, gestaltet sich eine Programmüberarbeitung äußerst umständlich.

Hier kann von vorneherein nur mit sehr großen Zeilenabständen gearbeitet werden.

Jeder Heimcomputer speichert Programme und Daten auf einem Kassetten-Recorder oder einem Floppy-Laufwerk ab. Sehr wichtig beim Speichermedium Kassette ist es, die auf einer Kassette gespeicherten Daten oder Programme mit denen im RAM des Computers zu vergleichen. Sollten dabei Unstimmigkeiten auftreten, kann ein erneutes Abspeichern vor dem Verlust der Daten schützen. Nicht jeder Computer aber bietet den dazu erforderlichen Befehl »Verify«.

Das optimale Basic gibt es nicht, noch nicht. Bei jedem Computer muß ein Kompromiß eingegangen werden, was zum einen die Hardware und zum anderen die Software anbelangt. Nur wer genau weiß, was er von seinem Heimcomputer erwartet und wofür er ihn einsetzen will, der kann auch anhand der Übersichten entscheiden, welcher der für ihn Richtige ist.

(rs)

	Apple II	Atari 400/800	Colour Genie	Commodore 64, VC 20	Commodore 4000er, 8000er	Dragon 32/64	Laser 210/310 VZ 200	Sharp MZ-700	Sinclair ZX81	Sinclair Spectrum	Schneider CPC 464	TRS-80 Level II	Texas Instr. TI 99/4A
Mathematische Funktionen													
Absolutwert von x ABS(x)	+	+	+	+	+	+	+	+	+	+	+	+	+
Logische »UND«-Verknüpfung	+	+	+	+	+	0	+	+	+	+	+	+	0
Arkustangens	+	+	+	+	+	+	+	+	+	+	+	+	+
Cosinus	+	+	+	+	+	+	+	+	+	+	+	+	+
Logarithmus	+	+	+	+	+	+	+	+	+	+	+	+	+
Logische »Nicht«-Verknüpfung	+	+	+	+	+	0	+	+	+	+	+	+	0
Logische »Oder«-Verknüpfung	+	+	+	+	+	0	+	+	+	+	+	+	0
Integerfunktion	+	+	+	+	+	+	+	+	+	+	+	+	+
Sinus	+	+	+	+	+	+	+	+	+	+	+	+	+
Tangens	+	+	+	+	+	+	+	+	+	+	+	+	+
Zufallszahlen erzeugen	+	+	+	+	+	+	+	+	+	+	+	+	+
Systembefehle													
CLOAD	0	+	+	0	0	+	+	0	0	0	0	+	0
CSAVE	0	+	+	0	0	+	+	0	0	0	0	+	0
LOAD	+	+	0	+	+	0	0	+	+	+	+	0	+
SAVE	+	+	0	+	+	0	0	+	+	+	+	0	+
VERIFY	+	0	+	+	+	0	+	+	0	0	0	0	+
MERGE	0	0	0	0	0	0	0	+	0	+	+	0	0
Funktionen													
AUTO	+	0	+	0	0	0	0	+	0	0	+	+	Number
CLEAR/CLR	+	+	+	+	+	+	+	+	+	+	+	+	0
CONT	+	+	+	+	+	+	+	+	+	+	+	+	+
DEF	+	0	+	+	+	+	0	+	0	+	+	0	+
DEL	+	0	+	0	+	+	0	+	0	0	+	+	+
DIM	+	+	+	+	+	+	+	+	+	+	+	+	+
EDIT	0	0	+	0	0	+	0	0	+	+	+	+	
ELSE	0	0	+	0	0	+	+	0	0	0	+	+	+
END	+	+	+	+	+	+	+	+	0	0	+	+	+
FOR	+	+	+	+	+	+	+	+	+	+	+	+	+
GOSUB	+	+	+	+	+	+	+	+	+	+	+	+	+
GOTO	+	+	+	+	+	+	+	+	+	+	+	+	+
IF THEN	+	+	+	+	+	+	+	+	+	+	+	+	+
LIST	+	+	+	+	+	+	+	+	+	+	+	+	+
NEW	+	+	+	+	+	+	+	+	+	+	+	+	+
NEXT	+	+	+	+	+	+	+	+	+	+	+	+	+
ON x GOTO	+	+	+	+	+	+	0	+	0	0	+	+	+
PEEK	+	+	+	+	+	+	+	+	+	+	+	+	0
POKE	+	+	+	+	+	+	+	+	+	+	+	+	0
TRACE	+	0	TRON	0	0	TRON	0	TRON	0	0	TRON	TRON	+
NOTRACE	+	0	TROFF	0	0	TROFF	0	TROFF	0	0	TROFF	TROFF	+
USR	+	+	+	+	+	+	+	+	RAND USR addr.	RAND USR addr.	0	+	0
CALL	+	0	+	0	0	0	0	0	0	0	+	0	+
SYS	0	0	0	+	+	0	0	0	0	0	0	0	0
SYSTEM	0	0	+	0	0	0	0	0	0	0	0	+	0
DATA	+	+	+	+	+	+	+	+	0	+	+	+	+
READ	+	+	+	+	+	+	+	+	0	+	+	+	+
RESTORE	+	+	+	+	+	+	+	+	0	+	+	+	+
RENUM	0	0	+	0	0	+	0	+	0	0	+	+	0

Heute: Bildschirmgestaltung (1)

Wort:	Text ausgeben	Variable ausgeben	Ausgabe: enger Abstand	Ausgabe: weiter Abstand
Bedeutung: / **System:**	Ausgabe des Textes zwischen Anführungszeichen	Ausgabe des Werts der Variablen X	Gibt Texte/Zahlen unmittelbar nebeneinander aus (bei Zahlen: 1 Leerst. Zwischenr.)	Gibt Texte/Zahlen mit größerem Abstand aus. (6–10 Leerst., je nach System)
ACORN	PRINT"Hallo!" (in allen Systemen gleich)	PRINT X (in allen Systemen gleich)	PRINT X;Y;Z (in allen Systemen gleich)	PRINT X,Y,Z (in allen Systemen gleich)
ALPHA-TRONIC	PRINT kann bei fast allen Rechnern durch ? ersetzt werden.			
APPLE				
ATARI				
COLOUR GENIE				
COMMODORE 8096/C-64/VC20				
SIMONS BASIC				
EX-BASIC				
DRAGON				
EPSON HX-20				
IBM PC				
LASER 310				
LASER 3000				
MSX-BASIC				
ORIC				
SCHNEIDER CPC				
SHARP MZ-Serie				
SHARP PC-1500				
SIRIUS				
SPECTRAVIDEO				
TANDY C.C.				
TANDY TRS 80				
TI-99/4A				
-EXT-BASIC				
ZX-81 ZX-SPECTRUM				

Alle BASIC-Wörter im Vergleich

Ab dieser Ausgabe finden Sie in jedem Heft BASIC-Befehle, -Anweisungen und -Funktionen, wie sie auf den wichtigsten Heim- und Personalcomputern aussehen. Am Ende der Serie erhalten Sie dazu ein alphabetisches Verzeichnis aller vorgestellten BASIC-Wörter. Die Tabellenform erleichtert Ihnen das Übersetzen von Programmen aus einem BASIC-Dialekt in einen anderen.

Bitte beachten Sie: Wenn ein Ausdruck in *Kursivschrift* erscheint, dann ist eine Taste gemeint, die Sie drücken müssen!

Heute: Bildschirmgestaltung (1)

Wort:	Bildschirm löschen	Piepston erzeugen	Auf Spalte positionieren	Auf Spalte u. Zeile position.
Bedeutung: / **System:**	Löscht alle Zeichen auf dem Bildschirm	Erzeugt einen kurzen Piepston (»akustisches Signal«, abgekürzt BEL)	Setzt die Schreibmarke (Cursor) auf Spalte S und gibt dort den Text aus	Setzt die Schreibmarke (Cursor) auf Zeile Z und Spalte S und gibt dort den Text aus
ACORN	CLS	PRINT CHR$(7)	PRINT TAB(S);"Hallo" oder PRINT TAB(S)"Hallo!"	PRINT TAB(S,Z);"Hallo!"
ALPHA-TRONIC		PRINT CHR$(7)	Diese Anweisung gilt für alle Systeme außer für den SHARP PC-1500, wo es heißt: CURSOR(S):PRINT"Hallo!"	PRINT$(Z,S),"Hallo!"
APPLE	HOME	BEEP		VTAB Z:HTAB S:PRINT"Hallo!"
ATARI	PRINT"*ESCAPE,CLEAR*"	PRINT"*ESCAPE,CONTROL+2*"		POSITION(S,Z):PRINT"Hallo!"
COLOUR GENIE	CLS	PRINT CHR$(7)		PRINT@ X,"Hallo!" (X=(Z−1)*40+S)
COMMODORE 8096/C-64/VC20	PRINT"*CLR*"			POKE 214,Z:POKE 211,S: SYS 58460:PRINT"Hallo!"
SIMONS BASIC	ebenso			PRINT AT(S,Z)"Hallo!"
EX-BASIC	ebenso			PRINT@ X,"Hallo!" (X=(Z−1)*40+S)
DRAGON	CLS	BEEP		PRINT@ X,"Hallo!" (X=32*(Z−1)+S)
EPSON HX-20	CLS	SOUND 10,10		LOCATE S,Z:PRINT"Hallo!"
IBM PC	CLS	BEEP		LOCATE Z,S:PRINT„Hallo!"
LASER 310	CLS	SOUND 1,1		PRINT@ X,"Hallo!" (X=32*(Z−1)+S)
LASER 3000	HOME	SOUND 1,1		HTAB(S):VTAB(Z):PRINT"Hallo!"
MSX-BASIC	CLS	BEEP		LOCATE S,Z:PRINT"Hallo!"
ORIC	CLS	PING		PRINT@ S;Z;"Hallo!"
SCHNEIDER CPC	CLS	PRINT CHR$(7)		LOCATE S,Z:PRINT"Hallo!"
SHARP MZ-Serie	CLS	MUSIC"+G4"		CURSOR S,Z:PRINT"Hallo!"
SHARP PC-1500	CLS	BEEP ON:BEEP 1,1,1000		Nicht sinnvoll bei Einzeilen-Display
SIRIUS	PRINT CHR$(12)	PRINT CHR$(7)		
SPECTRAVIDEO	CLS	BEEP		LOCATE(S,Z):PRINT"Hallo!"
TANDY C.C.	CLS	SOUND 100,3		PRINT@ X,"Hallo!" (X=32*(Z−1)+S)
TANDY TRS 80	CLS	PRINT CHR$(7)		PRINT@ (Z,S),"Hallo!"
TI-99/4A	CALL CLEAR	CALL SOUND(200,440,2)		
-EXT-BASIC	ebenso	ebenso		DISPLAY AT(Z,S):"Hallo!"
ZX-81 ZX-SPECTRUM	CLS	BEEP		PRINT AT Z,S;"Hallo!"

Heute: Bildschirmgestaltung (2)

Wort:	Cursor hoch (↑)	Cursor tief (↓)	Cursor links (←)	Cursor rechts (→)
Bedeutung:	Setzt die Schreibmarke (CURSOR) eine Zeile nach oben	Setzt die Schreibmarke (CURSOR) eine Zeile nach unten	Setzt die Schreibmarke (CURSOR) eine Zeile nach links	Setzt die Schreibmarke (CURSOR) eine Zeile nach rechts
System:				
ACORN	PRINT CHR$(11)	PRINT CHR$(10)	PRINT CHR$(8)	PRINT CHR$(9)
ALPHATRONIC	PRINT CHR$(30)	PRINT CHR$(31)	PRINT CHR$(29)	PRINT CHR$(28)
APPLE	CALL -998	CALL -922	CALL -1008	PRINT CHR$(21)
ATARI	PRINT CHR$(28)	PRINT CHR$(29)	PRINT CHR$(30)	PRINT CHR$(31)
COLOUR GENIE	PRINT CHR$(27)	PRINT CHR$(26)	PRINT CHR$(24)	PRINT CHR$(25)
COMMODORE 8096/C-64/VC20	PRINT CHR$(145)	PRINT CHR$(17)	PRINT CHR$(157)	PRINT CHR$(29)
SIMONS BASIC				
EX-BASIC				
DRAGON	PRINT CHR$(94)	PRINT CHR$(10)	PRINT CHR$(8)	PRINT CHR$(9)
EPSON HX-20	PRINT CHR$(30)	PRINT CHR$(31)	PRINT CHR$(29)	PRINT CHR$(28)
IBM PC	PRINT CHR$(30)	PRINT CHR$(31)	PRINT CHR$(29)	PRINT CHR$(28)
LASER 310	PRINT CHR$(27)	PRINT CHR$(10)	PRINT CHR$(8)	PRINT CHR$(9)
LASER 3000	PRINT CHR$(30)	PRINT CHR$(31)	PRINT CHR$(29)	PRINT CHR$(28)
MSX-BASIC	PRINT CHR$(30)	PRINT CHR$(31)	PRINT CHR$(29)	PRINT CHR$(28)
ORIC	PRINT CHR$(11)	PRINT CHR$(10)	PRINT CHR$(8)	PRINT CHR$(9)
SCHNEIDER CPC	PRINT CHR$(11)	PRINT CHR$(10)	PRINT CHR$(8)	PRINT CHR$(9)
SHARP MZ-Serie	PRINT CHR$(18)	PRINT CHR$(17)	PRINT CHR$(20)	PRINT CHR$(19)
SHARP PC-1500				
SIRIUS				
SPECTRAVIDEO	PRINT CHR$(30)	PRINT CHR$(31)	PRINT CHR$(29)	PRINT CHR$(28)
TANDY C.C.	PRINT CHR$(94)	PRINT CHR$(10)	PRINT CHR$(8)	PRINT CHR$(9)
TANDY TRS 80	PRINT CHR$(27)	PRINT CHR$(26)	PRINT CHR$(24)	PRINT CHR$(25)
TI-99/4A				
-EXT-BASIC				
ZX-81 ZX-SPECTRUM	PRINT CHR$(112) PRINT CHR$(11)	PRINT CHR$(113) PRINT CHR$(10)	PRINT CHR$(114) PRINT CHR$(8)	PRINT CHR$(115) PRINT CHR$(9)

Heute: Bildschirmgestaltung (2)

Wort:	INVERS	NORMAL	Blinken	Formatierte Ausgabe
Bedeutung:	Stellt Zeichen invers (dunkel auf hellem Untergrund) dar	Schaltet in den Normalzustand zurück (helle Zeichen auf dunklem Hintergrund)	Läßt die Schrift in kurzen Abständen aufleuchten	Druckt Zahlen und Texte formatiert, also in Tabellenform. S$ enthält Formatangaben
System:				
ACORN	mit COLOUR (abhängig vom Mode)	MODE 0	COLOUR-Anweisung abhängig vom MODE	@%=...(String format) PRINT A,B,...
ALPHATRONIC	PRINT CHR$(28)	PRINT CHR$(18)		PRINT USING S$;A;B;...
APPLE	INVERSE	NORMAL	FLASH	Hilfsprogramm erforderlich*
ATARI	◪-Taste drücken	◪-Taste drücken		Hilfsprogramm erforderlich*
COLOUR GENIE				PRINT USING S$;A;B;...
COMMODORE 8096/C-64/VC20	PRINT CHR$(18)	PRINT CHR$(146)		Hilfsprogramm erforderlich*
SIMONS BASIC	INV a,b,c,d		FLASH A,B (A=Farbe, B=Geschw.)	USE S$, STR$(A):PRINT (A muß erst in String umgew. werden)
EX-BASIC				PRINT USING S$,A,B,...
DRAGON				PRINT USING $;A,B,...
EPSON HX-20				PRINT USING S$;A,B,...
IBM PC	COLOR 0,7	COLOR 2,0	COLOR X (x>16)	PRINT USING S$;A,B,...
LASER 310				PRINT USING S$;A,B,...
LASER 3000	INVERSE	NORMAL	FLASH	PRINT USING S$;A;B;...
MSX-BASIC				PRINT USING S$;A,B,...
ORIC	PRINT CHR$(27);"@"; CHR$(27);"w"	PRINT CHR$(27);"G"; CHR$(27);"P"	PRINT CHR$(27),"L"	Hilfsprogramm erforderlich*
SCHNEIDER CPC	PRINT CHR$(24)		mit EVERY	PRINT USING S$;A,B,...
SHARP MZ-Serie	mit COLOR			PRINT USING S$;A,B,...
SHARP PC-1500	mit POINT, GCURSOR, GPRINT			USING S$: PRINT A,B,...
SIRIUS				PRINT USING S$;A,B,...
SPECTRAVIDEO	PRINT CHR$(27);"p"	PRINT CHR$(27);"q"		PRINT USING S$;A,B,...
TANDY C.C.				PRINT USING S$;A,B,... (nur erweitertes BASIC)
TANDY TRS 80	PRINT CHR$(16)	PRINT CHR$(17)		PRINT USING S$;A,B,...
TI-99/4A				
-EXT-BASIC				PRINT USING S$: A,B,...
ZX-81 ZX-SPECTRUM	G-Taste drücken INVERS 1	G-Taste drücken INVERS 0	FLASH 1	Hilfsprogramm erforderlich*

Berichtigungen zum letzten INFO

Zum Info selbst möchte ich noch ein paar Punkte bemerken. Die Übersicht der Basic-Dialekte auf Seite 18 enthält ein paar Fehler. Das Colour Genie und der TRS80 kennt die Befehle SWAP (A,B) und SPACE$(B) nicht. Im Diskbasic wird SPACE$ zwar nicht als Fehler interpretiert, jedoch nicht richtig abgearbeitet.

Peter Spieß

```
*****************************************************************************
*                        Ergaenzung und Aenderung                            *
*****************************************************************************
```

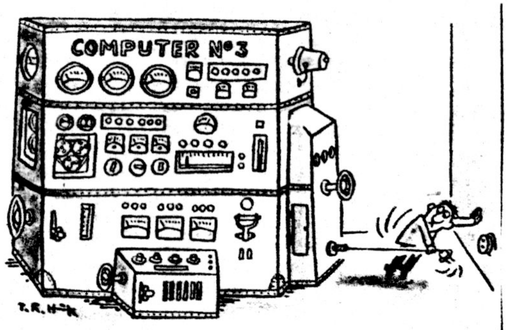
Synchronisierte Taktumschaltung fuer Videogenie

In der Clubinfo Nr. 8 ist der Schaltplan fuer eine Taktumschaltung fuer das Videogenie, der zwei Fehler enthaelt.

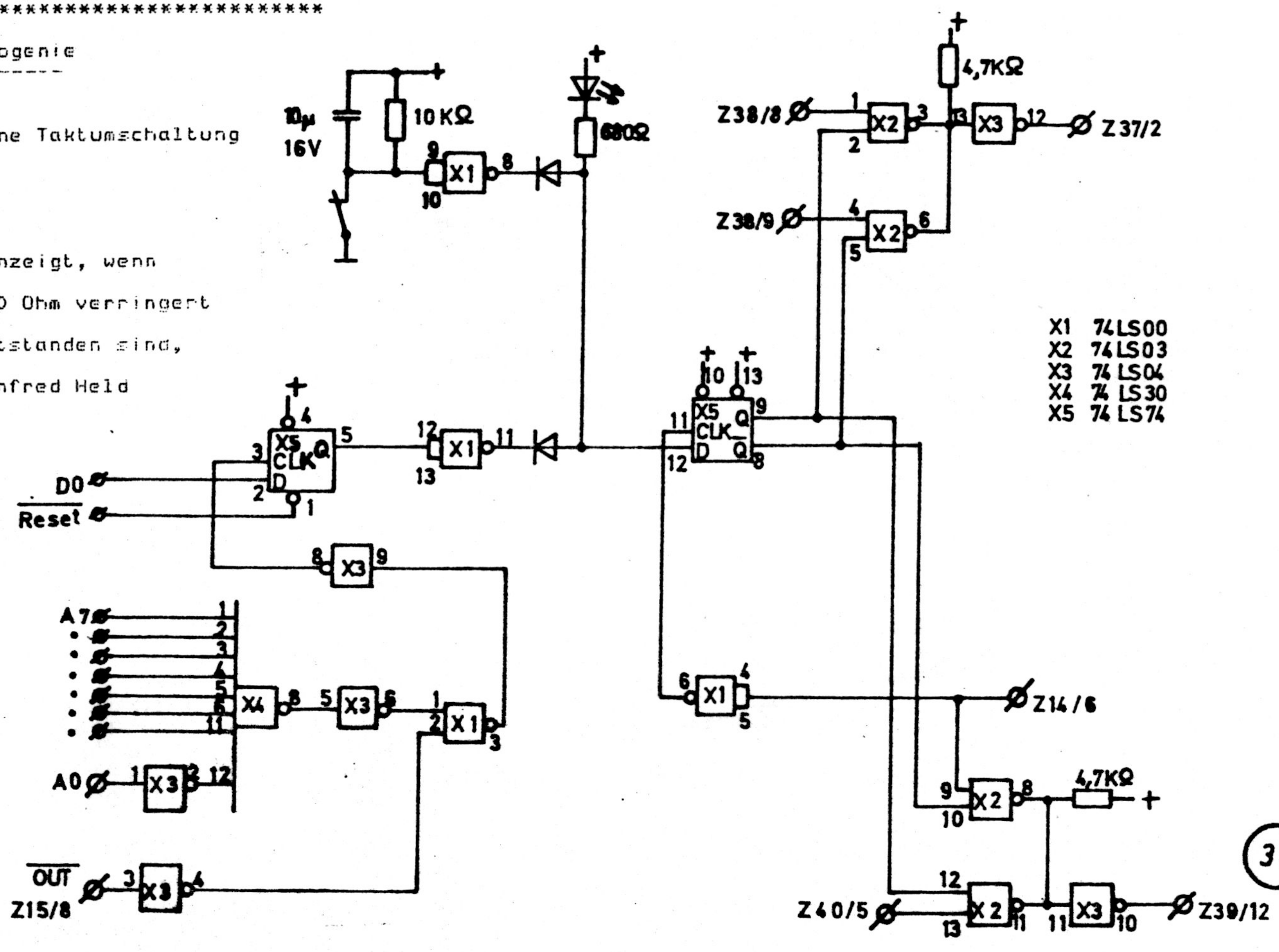
- Am Ic X3 ist Pin 10 und Pin 11 vertauscht.
- Am Ic X5 ist Pin 8 und Pin 9 vertauscht.

Als ergaenzung ist eine LED hinzugekommen, die anzeigt, wenn die CPU mit 3.55 MHz arbeitet.
Bedingt durch die LED muss der Widerstand auf 680 Ohm verringert werden.
Die Fehler, die beim Zeichnen des Schaltplans entstanden sind, bitte ich zu entschultignen.

Manfred Held



„Nichts auf der Welt ist wirklich vollkommen!"



ECB - BUS - Projekt

Im letzten Clubinfo fragt Walter Piller, was man als Einsteiger mit dem ECB-BUS anfangen kann. Das regt mich jetzt zu einem Grundsatzartikel für alle Clubmitglieder an. Damit soll auch der Computerbesitzer der sich für Hardware nicht (oder noch nicht) interessiert wissen, wovon wir doch schon seit einigen Club-Infos schreiben.
Es gibt in unzähligen Fachzeitschriften Bauvorschläge für die diversesten Hardware-Erweiterungen. Ich will da nur die gebräuchlichsten aufzählen:
High-resolution-grafik; Speichererweiterungen; EPROM-Bank; Hardware Uhr mit Batteriepufferung; Parallelschnittstelle; serielle Schnittstelle (V 24); Eprommer; Basicode-Hardware; Supertape; Joystick; Relais-Steuerkarte;................
Wer nun irgendeine dieser feinen Dinger kauft oder selberbaut, wird auf 2 Probleme stoßen:
Entweder ist die Erweiterung nicht speziell für den TRS/VG gebaut und ist damit nicht oder nur schwierig verwendbar.
Oder es steckt schon eine Erweiterung am Expansion-port. Dann muß man je nach Anwendung ständig die diversen Erweiterungen an oder abstecken. Das ruiniert in kurzer Zeit die ohnehin nicht sehr massiven Platinenstecker auf der Hauptplatine.

Die Lösung sieht also folgendermaßen aus:
Es führt ein Kabel vom Expansion-port zu einer Platine mit 10 Steckplätzen. Dieses Kabel bleibt ständig stecken. 1 Karte (die sogenannte Basiskarte) entkoppelt alle Signale zum und vom Computer.
Die restlichen 9 Steckplätze stehen für Erweiterungen aller Art zur Verfügung. Diese können ständig stecken bleiben, da sie im passiven Zustand keinerlei Einfluß auf den Computer haben.
Damit erklärt sich die Frage nach dem Anwenderkreis fast von selbst:
Jeder der sich mit Hardware-Erweiterungen beschäftigt und die ewige Herumsteckerei satt hat, wird dieses Bauprojekt gut gebrauchen können.
Nochmals: Die ECB-Bus Platine samt Basiskarte kann allein noch überhaupt nichts. Sie bildet lediglich die Basis für kommende Hardware- Basteleien aller Art.
Was alles von der Hardware-Gruppe entwickelt oder adaptiert wird, hängt von der Interessenrichtung jedes Mitglieds und auch vom Leserecho des gesamten Club80 ab.
Bei mir sind zur Zeit Schaltpläne für folgende Einrichtungen verfügbar (keine Platinen):
Parallelschnittstelle 3 mal 8 bits (mit 8255)
Serielle Schnittstelle (mit 8251)
Hardware-Uhr mit Akku-Pufferung
HRG - Grafik mit Texas Ic's (TMS....) + 8 mal 4116

Hardware - Arbeitskreis ECB - BUS System

Warum gerade ECB-BUS ?

Wenn man ein solches Steckkartensystem entwickelt, stellt sich gleich zu Anfang die Frage nach der Norm. Es soll ja nicht so sein, daß jedes Mitglied der Hardware-Gruppe seine Karten frei verdrahtet, sondern nach Möglichkeit jeder von der Entwicklung des Anderen profitieren kann. Das zwingt zu einer einheitlichen Belegung der Steckerstifte. Und wenn man schon soweit ist, kann man sich gleich am Markt umsehen, ob es schon eine derartige Norm gibt. Und sieheda es gibt sie: Die ECB-Norm. Das ist ein System zur Vereinheitlichung aller Erweiterungen für beliebige Computer, soweit sie mit einer Z80 CPU bestückt sind. Ursprünglich wurde dieses System für CP/M - Rechner der verschiedensten Hersteller entwickelt. Damit konnten Erweiterungen verschiedener Hersteller mit fast jedem CP/M Rechner verbunden werden. Dieses Konzept paßt uns ausgezeichnet Denn es gibt am Markt schon einiges an Bausätzen oder gar fertigen Bausteinen zu kaufen.
Es wird auch laufend von den Zeitschriften 'elektor' und c't etwas für das ECB-System veröffentlicht, was für uns nach minimalen Modifikationen einsetzbar ist.
Einheitlich ist dabei auch die Größe jeder Steckkarte. Nämlich genau eine sogenannte Europakarte, von der das System auch seinen Namen hat. (Euro-card-bus)
Die Maße sind: 160 mal 100 mm. Das ist sehr handlich; sodaß die gesamte ECB- Hardware samt Netzteil in einem relativ kleinen Metallgehäuse untergebracht werden kann.
So, ich hoffe daß nach Lektüre dieses Beitrages jeder Clubfreund weiß, was wir im Schilde führen und wir auf keinen Fall ein Club im Club sein wollen. Sollte noch die eine oder andere Frage offen geblieben sein, so bitte ich Euch, ohne Hemmungen zu schreiben oder mich anzurufen.

Walter
....
W. Zwickel

ECB -Bus Netzteil

Wie im Club Info Nr.8 von Hartmut berichtet, habe ich mir Gedanken um die Stromversorgung für unseren ECB -Bus gemacht. Die Aufgabenstellung war, wir brauchen ein Netzteil mit den folgenden Spannungen/Strömen: +12V/1A, -12V/0.5A, +5V/3,5A bis 5A und -5V/0.5A.

Ein Netzteil mit diesen Daten und sogenannter Längsregelung ist im Prinzip einfach aufzubauen, das Problem ist aber der Preis. Alleine der Trafo kostet min. 50,-DM, der Längsregler für 5A ca. 30,-DM, eine Kostenrechnung für alle Bauelemente einschl. Platine, Kühlkörper usw ergab einen Preis von ca. 130,-DM.

Die billigere Alternative dazu ist ein Kaufnetzteil. Die Firma:

Erich-Willi MEYER
6343 Frohnhausen (tel 02771/35071)
Postfach 74c8

bietet ein Schaltnetzteil! mit folgenden Daten: +5V/3,5A, -5V/0,5A, +12V/2A, -12V/0,5A. Der Preis, komplett aufgebaut incl. Gehäuse 99,80DM.

Ich habe mir dieses Netzteil inzwischen gekauft und getestet. M.E. ist dieses Netzteil vom Preis und Leistung her für unsere Zwecke ideal geeignet. Wer bei 5V mehr als 3,5A benötigt kann auch die Version mit 5V/5A (119,80DM) oder die Version 5V/7,5A (149,50 DM) kaufen.

Ulrich Böckling

Roboter festgenommen

LOS ANGELES, 19. August (AFP). Wegen »öffentlicher Ruhestörung« hat die Polizei von Los Angeles im luxuriösen Villenviertel Beverly Hills einen Roboter »festgenommen«, der knarrend und klirrend durch die Straßen stapfte. Zwar konnten die Beamten dank des Walkie-talkie-Systems des ferngelenkten Roboters mit dessen Besitzer Verbindung aufnehmen, doch weigerte sich dieser, Name und Adresse zu nennen. Daraufhin wurde der Automat ohne weiteres Aufheben vom Abschleppdienst weggeschafft.

Zwischenbericht:
In diesem Info wird an mehreren Stellen über das ECB-Bus- Projekt berichtet. Ich möchte hier kurz über die eigentliche Adapterplatine ein paar Worte verlieren.

Nachdem Walter Zwickel die Schaltung für den ECB-Adapter entworfen und in Fädeltechnik aufgebaut hatte, war es an der Zeit, sich über die sog. Entflechtung der Platine Gedanken zu machen. Mit diesem Problem beschäftigte sich vor allem Manfred Held. Leider mußte er schon nach kurzer Zeit feststellen, daß eine vollständige Entflechtung nicht möglich war. Es würden immer ein paar, in Fädeltechnik nachzulötende, Verbindungen übrigbleiben.

Wie sehr der Manfred damit recht hatte, erwies sich, als Walter Zwickel die Möglichkeit erhielt an einem Großcomputer mit CAD - Programm eine Leiterplattenentflechtung zu machen. Selbst dieser Computer mit dem entsprechenden Spezialprogramm war nicht in der Lage alle Leiterbahnen vollständig zu entwirren.

Als Ergebnis erhielt der Walter zwei Platinenzeichungen für eine doppelseitige Platine, bei der die beiden Anschlußleisten (Computer<->Adapterplatine und Adapterplatine<->96poliger ECB-Stecker) in Fädeltechnik verdrahtet werden müssen. Die Bestückungsseite der Platine enthält nur relativ wenige Verbindungen (siehe Layout).

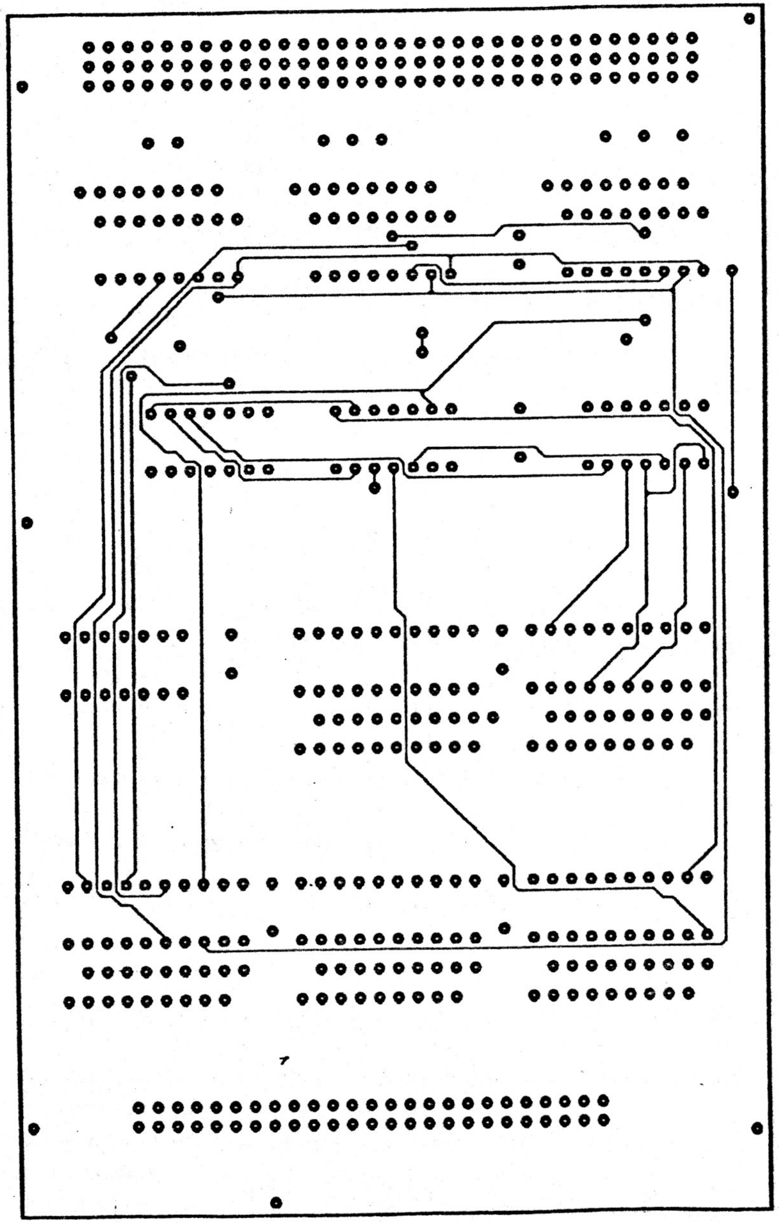
Grundsätzlich gibt es nun zwei Möglichkeiten von dem Layout Gebrauch zu machen:
1. Man läßt sich von einer Firma eine zweiseitige Platine mit allen Bohrungen und Durchkontaktierungen herstellen.
2. Eines der Mitlieder der Hardwaregruppe (wahrscheinlich Manfred Held) stellt eine einseitige Platine her und die fehlenden Leitungen werden in Fädeltechnik nachgelötet.

Gegen die erste Möglichkeit spricht vor allem der relativ hohe Preis pro Platine. Die zweite Möglichkeit ist zwar billig, erfordert jedoch ein bischen Erfahrung im Umgang mit dem Lötkolben. Da aber sowieso gelötet werden muß (siehe oben), fällt dieses Argument nicht ins Gewicht. Eine entgültige Entscheidung ist jedoch noch nicht gefallen.

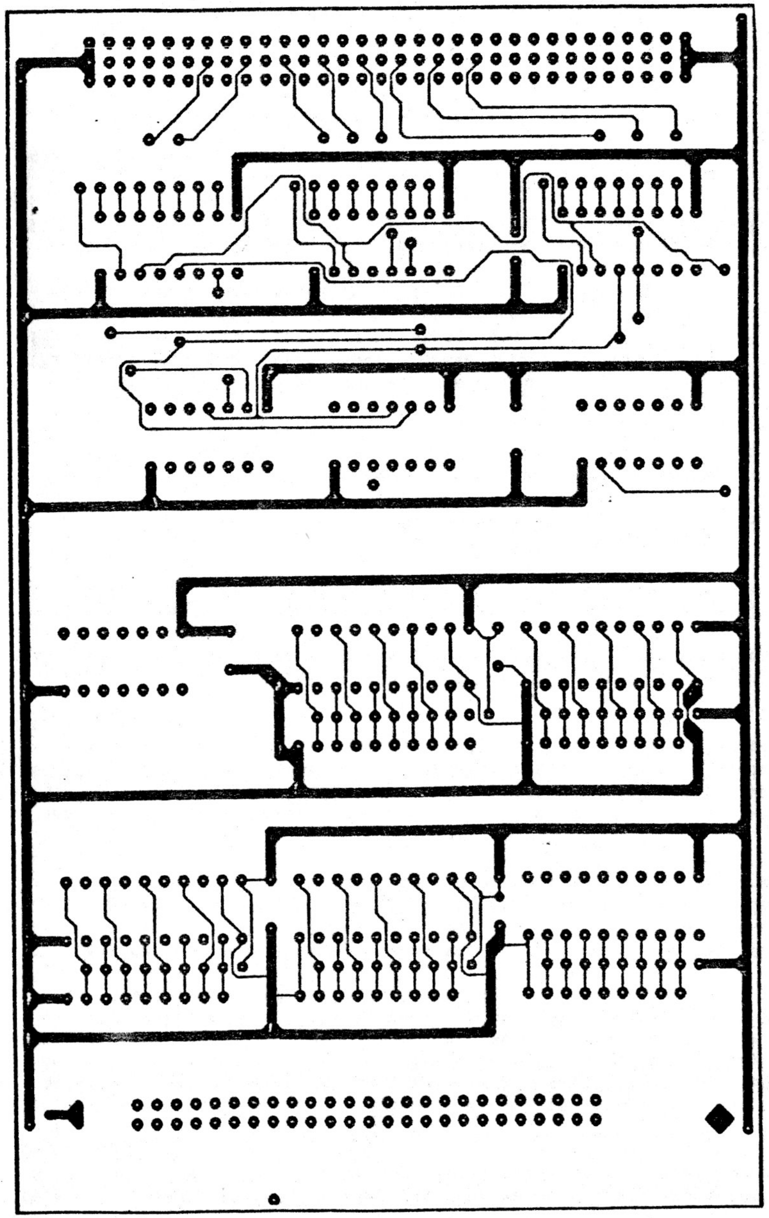
Zur Zeit wird durch Ulrich Böckling von der mit dem Computer erstellten Layout-Zeichung eine Fotopositivfolie erstellt. Sobald diese Arbeit erledigt ist, wird eine Entscheidung (einseitige/doppelseitige Platine) getroffen und ich glaube, daß wir schon vor Erscheinen des nächsten Clubinfos den Mitgliedern der Hardwaregruppe die Platine zur Verfügung stellen können.

Sollte irgend jemand noch Vorschläge oder Einwände im Zusammenhang mit der Platinenerstellung haben, so bitte ich diese möglichst schnell (am besten telefonisch bei mir) vorzubringen.

Das war's in Sachen Hardware, tschüß Hartmut Obermann



Bestückungsseite



Lötseite

Hans-Joachim Regge

Test-adapter

Die Fehlersuche wird oft zur Tortur, wenn die Europakarte gut bestückt ist und zudem in einem Gehäuse sitzt. Die beiden hier beschriebenen Adapterkarten schaffen da Abhilfe.

Die Idee der Verwendung von Adapterkarten ist nun keineswegs neu. Leider sind solche Karten im allgemeinen recht teuer, zumal man für unterschiedliche Bussysteme auch verschiedene Testadapter benötigt. Aus diesem Grund sind die beiden vorliegenden Adapterkarten entstanden, die beide mit Steckverbindern nach DIN 41612 ausgerüstet sind. Der wesentliche Unterschied liegt in der Anzahl der Steckerkontakte: die eine Karte ist für Systeme mit 64poligen, die andere für solche mit 96poligen Steckern vorgesehen. Sie sind nicht nur für den ECB- oder MC-65-Bus, sondern auch für eine Vielzahl anderer Bussysteme geeignet.

Kleine Kapazitäten

Die Busleitungen eines Mikroprozessorsystems werden mit Taktfrequenzen bis zu 8 MHz und mehr betrieben. Wird nun durch Verlängerung der Busleitungen die Kapazität derselben (gegen eine andere Leitung oder gegen Masse) zu groß, so kann es zum Übersprechen oder – der zweite Fall – zur Verformung der Signale kommen. Beides kann die Funktion des Gerätes beeinträchtigen und muß deshalb auf jeden Fall vermieden werden.

Bei den beiden hier vorgestellten Karten sind mit Absicht die Leiterbahnen nur 0,33 mm breit gewählt worden, um die besagten Kapazitäten so gering wie möglich zu halten. Die *Bilder 1* und *2* zeigen das Layout der 96poligen Adapterkarte. Bei der 64poligen Karte konnte zwischen zwei Signalleitungen jeweils eine dünne Masseleitung untergebracht werden, um eine zusätzliche Leiterbahnentkopplung zu erreichen. Versuche ergaben, daß sowohl der mc-CP M-Computer als auch der MC-65 problemlos arbeiteten, wenn Systemkarten mit diesen Adaptern betrieben wurden.

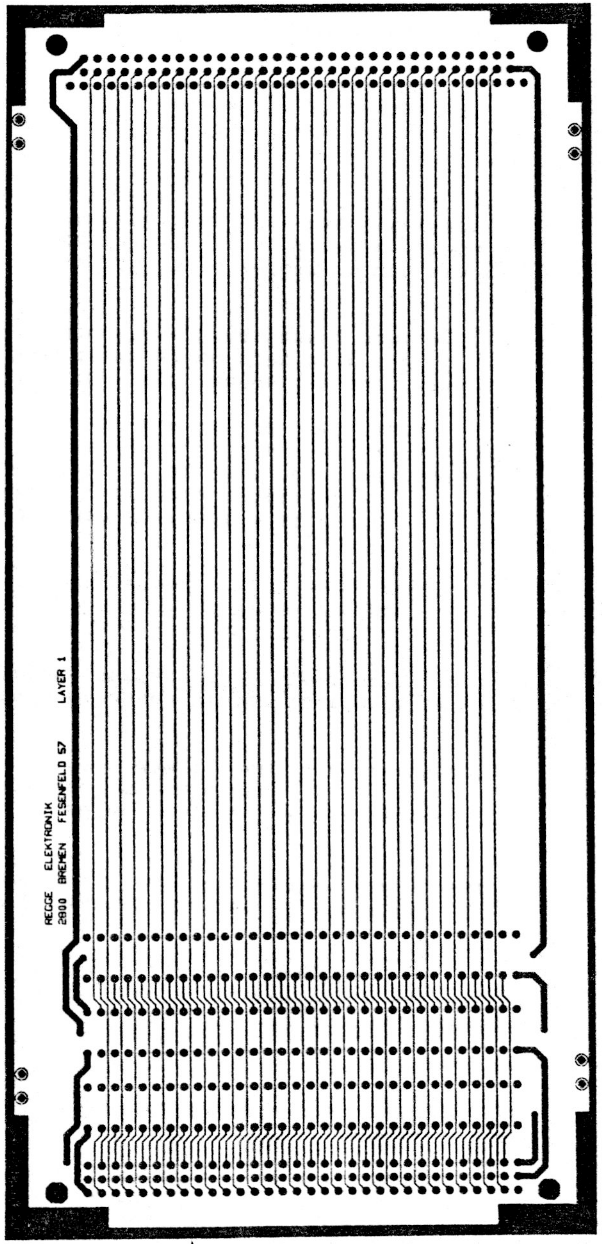


Bild 1. Die Bauteilseite der 96poligen Adapterkarte

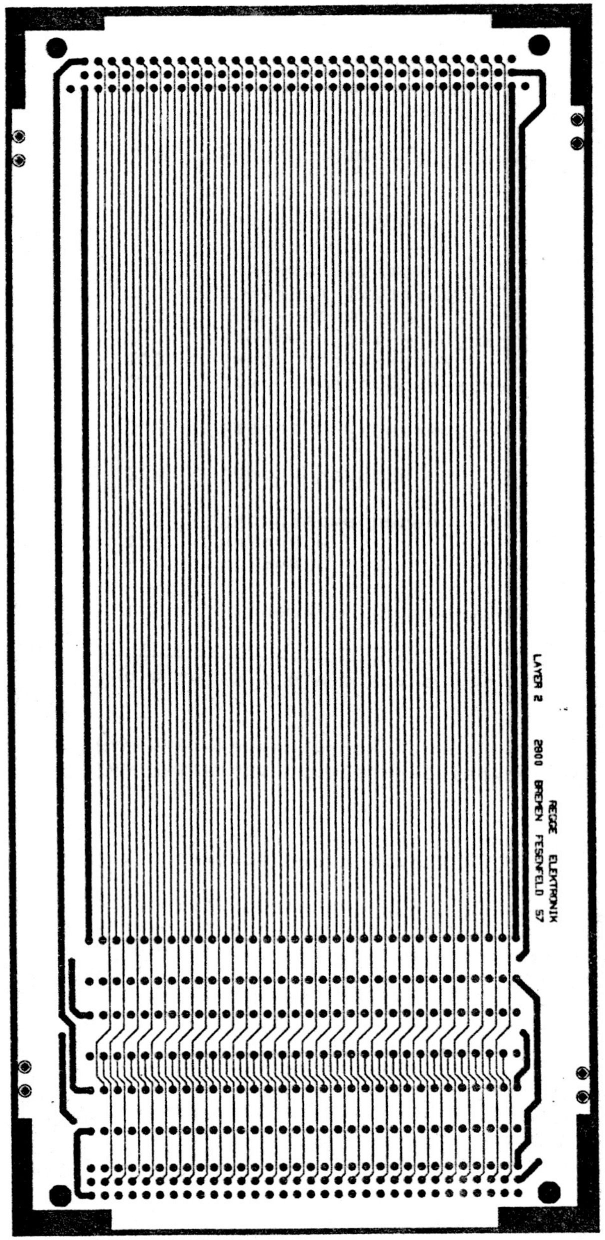


Bild 2. Die Lötseite des 96poligen Testadapters

Eine große Hilfe: Schalter

Zur Untersuchung von Karten, auf denen Kurzschlüsse vermutet werden, ist es erforderlich, einzelne oder auch alle Leitungen vom System zu trennen. Hierzu sind auf den Adapterkarten zwei (64polig) bzw. drei (96polig) Reihen DIL-Schalter vorgesehen, mit denen alle Leitungen aufgetrennt werden können. Besonders bewährt haben sich da die Spezialschalter von Siemens, die über einen offenen Drahtbügel verfügen. Dieser Bügel eignet sich vorzüglich zum Anklemmen von Prüfspitzen und dergleichen. Die Schalter sollten so herum eingesetzt werden, daß der Drahtbügel zum System hin zeigt, da man dann im geöffneten Zustand die Systemsignale messen kann. Wer keine Schalter benötigt, kann stattdessen feste Drahtbrücken einlöten, die ebenfalls als Testpunkte gut geeignet sind.

Noch ein Hinweis zum Auftrennen von Leitungen, die Betriebsspannungen führen: Werden Datenanscchlüsse von ICs, deren *Betriebsspannung* „abgeklemmt" wurde, von anderen „versorgten" ICs angesteuert, so können ICs beschädigt werden. Deshalb: Wenn zu Testzwecken die Spannungsversorgung abgeklemmt wird, *auch* die Datenleitungen öffnen! ICs mit mehreren Versorgungsspannung nehmen es oft übel, wenn nur eine derselben abgeklemmt wird.

Damit die 64polige Adapterkarte für verschiedene Bussysteme verwendet werden kann, wurden die Masseleitungen zwar zusammengefaßt, aber noch nicht mit einem Anschluß des Steckers verbunden. Diese Verbindung ist nach Bedarf, am besten auf der Systembus-Seite, herzustellen. Wird bei bestimmten Bussystemen auch über eine dünne Mittelleitung eine Stromversorgung geführt, so muß diese gegebenenfalls verstärkt werden, da die Belastbarkeit im günstigsten Fall 1,5 A beträgt.

Platinen und Bausätze gibt es bei Regge Elektronik, Fesenfeld 57, 2800 Bremen.

HARDWARE - ERWEITERUNG fuer das Video-Genie, Tandy, Komtek
==

** R E L A I S - I N T E R F A C E **

Nachdem im letzten INFO von Walter Zwickel eine ECB-Bus Erweiterung fuer unsere Rechner vorgestellt wurde, moechte ich an dieser Stelle ueber ein Projekt sprechen, an dem ich seit Februar 85 arbeite.

Es handelt sich hierbei um ein Relais-Interface ueber welches ich nun kurz berichten moechte.

Ganz am Anfang stand der Gedanke mit meinem Computer 'Etwas' steuern zu wollen. Nun dieses 'Etwas' sollte ein sogenannter Lehrn-Roboter sein oder aber auch 'Etwas anderes'. Nun war es gerade dieses 'Etwas andere' worueber ich mir den Kopf zerbrochen habe.

Jeder der sich etwas in der Hardware auskennt wird sofort einwenden, dass es keines grossen Aufwandes bedarf um ein paar Relais mit dem Computer steuern zu koennen.

Doch wieviel Relais muessen es sein, damit ich einen Roboter und jemand anderer 'Etwas anderes' steuern kann ??
Und wie soll die Software aussehen, damit jedermann das steuer kann, was er gerne moechte ??

Es gab viele Probleme in jenem strengen Winter fuer mich um ein moeglichst universelles Relais-Interface zu erstellen.

Die entsprechende Hardware war schnell ausgedacht und gefaedelt. Nach einigen kleinen Basic-Programmen und den entsprechenden Aenderungen in der Hardware war der Hardware - Prototyp fertiggestellt.

Das Erstellen einer entsprechenden Software sollte sich allerdings als sehr viel schwieriger herausstellen.
Anfaengliche Basic-Programme habe ich schnell ueber den Haufen geworfen, eine Steuerung der Karte von COBOL aus war auch nicht besonders sinnvoll. (Lediglich fuer die zahlreichen Masken)

Blieb also nur noch ASSEMBLER, zwar sehr aufwendig aber dafuer sehr schnell und Geschwindigkeit ist schon noetig um z.B 16 Relais innerhalb weniger Sekunden viele Schaltungen ausfuehren zu lassen.

Somit war das Problem der Wahl der Sprache geloest - nun graute es mir vor der Ausfuehrung. Nach zwei Monaten hatte ich schon einige Programme beisammen. (Testprogramme fuer das Interface, Programme zum 'SCHALTEN' der Relais, Programme zum 'TASTEN' der Relais).

Mir war klar, dass einiges nicht besonders elegant von der Programmierung war, aber die Funktion die mir vorschwebte war da. - An dieser Stelle musste naturgemaess ein Rueckschlag erfolgen: ein kompletter Systemabsturz mit der Folge, dass alle Sourcen hin waren. Mir blieb nur noch ein etwas aelterer Programmausdruck.

Nach diesem ersten Schreck habe ich mir nochmals ueber das gesamte Projekt Gedanken gemacht - das Ergebnis ist eine vollkommen neue Version (Version 2.1) der Software die wohl kaum noch Wuensche offen laesst.

Nach dem Motto: 'Software ist nur komplett mit Handbuch' gibt es zu dem Programm ein ca. 25 seitiges Manual.

Das Demo-Programm zum Relais-Interface kann ueeber den Hartumt Obermann aus der Programm-Bibliothek bezogen werden, oder aber direkt bei mir, dann komplett mit Handbuch (bitte Diskette und 2.60 DM fuer Kopien beilegen).

Das Demo-Programm befindet sich auf einer Diskette ohne System, 40 Tracks, ss/sd. Zusaetzlich werden drei Schalt-Dateien fuer die 24-Std. Steuerung bzw. Kurzzeitsteuerung mitgeliefert.
Diese Dateien koennen vom Programm eingelesen (A), Editiert (C) und ausgefuehrt (7) werden. Die Angaben in Klammern geben das entsprechende Programm an.

Noch einige Saetze zur Hardware:
Die Hardware ist zur Zeit 1. als Prototyp in Faedeltechnik und 2. auf Platine aufgebaut. Die Platinenversion 1.0 muss nochmals ueberarbeitet werden, da die Leiterbahnen zu den Relais-Schaltkontakten zu duenn ausgefallen sind und sich bei Stromstaerken um 0.5 bis 1 Ampere in Rauch aufloesen.

Der Gesamt-Hardware Umfang wird sich auf vier Europa-Karten (mit Netzteil) bzw. drei Karten (ohne Netzteil) erstrecken.

Zum Abschluss noch einige (gekuerzte) Seiten aus dem Anwender-Handbuch:

I n h a l t s u e b e r s i c h t

1.0	Allgemeine Anmerkungen	Seite: 3
1.1	Software	Seite: 4
1.2	Hardware	Seite: 5
1.3	Technische Daten	Seite: 6
2.0	Demo - Version	Seite: 6
3.0	Steuer - Programme	Seite: 7
3.1	Info (Hard/Software)	Seite: 8
3.2	Test (Lauflicht)	Seite: 9
3.3	Status Port A / Port B	Seite: 10
3.4	DOS - Kommando	Seite: 11
3.5	Tasten / Schalten	Seite: 12
3.6	Uhrzeit setzen	Seite: 13
3.7	Schaltuhr	Seite: 14
3.8	Rueckkehr zum DOS	Seite: 15
4.0	Verwaltungs - Programme	Seite: 15
4.1	Schaltzeiten speichern	Seite: 16
4.2	Schaltzeiten laden	Seite: 16
4.3	Schaltzeiten editieren	Seite: 17
5.0	Hardware - Komponenten	Seite: 18
5.1	Aufbau/Funktion der Karten	Seite: 19
5.2	Anschluss - Plaene	Seite: 20

1.0 Allgemeine Anmerkungen

Wer hat sich nicht schon einmal ueberlegt mehr mit seinem Computer anzufangen, als die 12. Version von 'Invader' zu spielen ?!

Es gibt zahlreiche Moeglichkeiten den eigenen Computer fuer Steuerungs- und Ueberwachungsaufgaben einzusetzen, allein es fehlt das noetige Interface und die entsprechende Software um diesen Einsatz zu ermoeglichen.

Die einfachste Steuerung ist wohl die zeitliche Ueberwachung von elektrischen Verbrauchern, die zu bestimmten Zeitpunkten ein- bzw. ausgeschaltet werden muessen.

Als Bespiele seien nur die Ueberwachung einer Zentralheizung (Pumpe, Brenner, Magnetventile fuer Gas und Wasser, Kaminklappe usw.) oder die Steuerung einer Modelleisenbahn genannt.

Es gibt aber auch Anwendungsfaelle, bei denen neben einer Schaltuhrfunktion auch die manuelle Steuerung von verschiedenen Verbrauchern gefordert wird. Ein einfacher Lehrrobotter mit vier Achsen wuerde acht Kanele zum Steuern benoetigen.

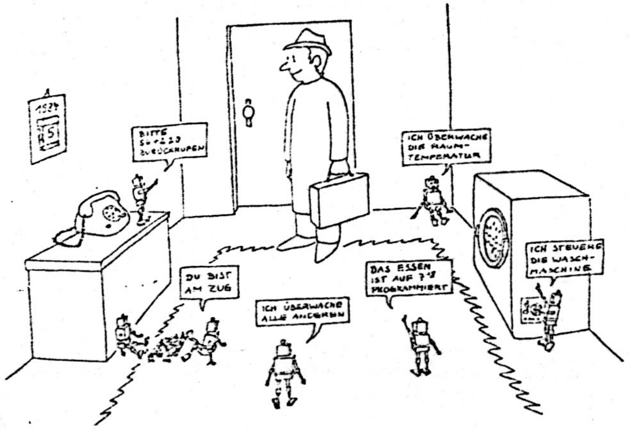
Die Arme eines Robotters muessten bei manueller Steuerung auf Tastendruck reagieren, zudem sollte eine Moeglichkeit der Speicherung von zeitlichen Bewegungsablaeufen und ihre Ausfuehrung gegeben sein. (sog. Kurzzeitsteuerungen)

Neben der reinen 24-Stunden-Steuerung kann mit einer Kurzzeitsteuerung (also Start eines Schaltprogramms von einem frei wahlbaren Zeitpunkt aus) eine Modelleisenbahn mit mehreren Zuegen sehr komfortabel gesteuert werden.

z.B.	Prog.		
10 Sek. nach	Start	Zug 1 faehrt an	
12 Sek. nach	Start	Zug 2 faehrt an	
15 Sek. nach	Start	Zug 3 faehrt an	
16 Sek. nach	Start	Zug 1 stoppt	
16 Sek. nach	Start	Zug 4 faehrt an	usw.

Es gibt sicherlich noch viele andere Faelle, bei denen man mit einer Relais-Schaltsteuerung Verbraucher ueberwachen kann.

In der vorliegenden Form der Relais-Steuerung per Computer wurde besonders auf die Benutzerfreundlichkeit der Software geachtet. Sie ist so ausgelegt, dass man fast alle Anwendungen mit ihr abdecken kann, sei es eine rein manuelle Schaltsteuerung per Taster bzw.per Schalter, oder aber eine vollautomatische 24-Stunden Uebrwachung.



1.1 Software

Die Software ist vollstaendig in Assembler geschrieben um die geforderte Verarbeitungsgeschwindigkeiten zu erreichen (mehrere Schaltungen innerhalb einer Sekunde).

Der jeweilige Schaltzustand der Relais-Ausgaenge ist jederzeit auf dem Bildschirm sichtbar, so dass bei Fern-Ueberwachung immer der Zustand der Verbraucher (bestromt/entstromt) am Bildschirm erkannt werden kann.

Wie bei vollautomatischen Steuerungen erforderlich gibt es auch hier das sogenannte 'NOT AUS', eine Funktion, mit der zu jedem Zeitunkt alle Verbraucher ausgeschaltet werden koennen.

Zahlreiche Testfunktionen und die Moeglichkeit der direkten Ausfuehrung von DOS-Kommandos runden die Steuerungs-Programme ab.

Die Gruppe der Verwaltungs-Programme bietet eine gute Hilfe beim Erfassen oder Editieren der Schaltzeiten fuer die einzelnen Relais. Es steht ein Bildschirmeditor zum problemlosen Veraendern oder Neuerfassen von Zeiten zur Verfuegung. Auf fuenf Bildschirmseiten stehen fuer 15 zu schaltende Relais insgesamt 270 Schaltzeiten zur Verfuegung.

Aufgerechnet auf ein einzelnes Relais kommen somit z.B. fuer die 24 Std. Schaltuhr 18 Schaltungen auf 24 Stunden (also neun mal Ein-Aus).

Benoetigt man weniger Relais-Ausgaenge, dafuer aber mehr Schaltzeiten, so kann man mehere Relais parallel schalten um somit die Anzahl der Schaltungen fuer einen Relais-Kanal zu erhoehen.

Die Schaltkontakte der Relais koennen mit 220 Volt und 1 Ampere belastet werden. Als Schaler stehen pro Relais 1 x um zur Verfuegung.

1.2 Hardware

Der mechanische Aufbau der Relais-Karten sieht eine Verteilung des Steuer- und Lastteils auf drei Europa-Karten vor.

Auf der Steuer(Logik)-Karte befinden sich der Adress-Dekoder, Leitungstreiber, LED-Treiber, LED's und eine Z80 PIO sowie einige weitere Komponenten.

Mit Hilfe der LED's wird zusaetzlich zur Schaltzustands-Anzeige auf dem Bildschirm der Status der Relais optisch direkt auf der Karte angezeigt.

Die Verbindung Steuer(Logik) Karte und Computer geschieht ueber ein 24 adriges Flachkabel mit Anpress-Stecker. Das freie Ende des Kabels wird direkt mit dem Computer oder einer vorhandenen BUS-Platine verdrahtet.

Die Software arbeitet bereits mit der Logik-Karte. Fuer eigene Anwendungen kann man z.B. auf die Relais-Karten verzichten und die Steuerleitungen (TTL-Pegel) direkt verwenden. In diesem Fall erfolgt die Stromversorgung vom Computer aus. (Jumper auf der Logik-Karte fuer interne/externe Stromversorgung)

Die Relaiskarten sind wie folgt aufgebaut:

Karte 1 enthaelt neben acht Relais ein Netzteil, das nicht nur die Relais dieser Karte (Nr. 1 bis 8) versorgt, sondern auch die Relais der zweiten Karte (Relais Nr. 9 bis 16), sowie die Logig-Karte.

Die zweite Relais-Karte enthaelt neben den Relais ein freies Wrap-Feld fuer eigene Entwicklungen.

Die Relais-karten werden ueber 16-adrige Falchkabel mit Anpress-Steckern mit der Logikkarte verbunden.

Die von der PIO gelieferten Schaltsignale werden den Relais ueber Treiber-Transistoren zugefuehrt. Die Treiber-Stufen befinden sich auf den Relais-Karten. Die Schaltkontakte der Relais sind auf Klemm/Schraubboecke an der Stirnseite der Platine gefuehrt, so dass dort dierekt Kabel bis 1mm Querschnit angeschlossen werden koennen.

Die Relais-Karten bilden zusammen mit der Logik-Karte einen kompakten Sandwich-Aufbau bei Montage ueber Abstandsrollen. An der einen Stirnseite sind wie erwaehnt die Anschlussboecke der Relais-Karten, sowie die LED's der Logik-Karte gefuehrt, die andere Stirnseite steht fuer die Anschluss-Kabel zur Verfuegung.

die Karten-Kombination kann daher bequem wahlweise in ein eigenes Gehaeuse oder aber in ein 19 Zoll Einschubrack installiert werden.

1.3 Technische Daten

Logik-Karte:

- Z 80 PIO, Bus Treiber, Adressdekoder, LED-Treiber, 16 LED's, Codierbruecke fuer externe/interne Stromversorgung

- Anschluss Logik-Karte (-) Computer ueber 24 adriges Flachkabel mit Anpress-Stecker

- Signale der Pio (Port A und B) auf jeweils eine 16 pol. IC-Fassung gefuehrt

Relais-Karte 1

- 16 Relais mit Kontakt 1 x um, 1 Ampere, Treiber-Stufen

- Anschlussboecke fuer Kabel bis 1 mm Querschnitt

- Netzteil mit 5 Volt und 1.5 Ampere (fuer Logik-Karte, Relais-Karte 1 und Relais-Karte 2)

Relais-Karte 2

- wie Relais-Karte 1, jedoch statt Netzteil freies Wrap-Feld

2.0 Demoversion

Die Treiber-Software ist in zwei Versionen (Demo bzw. Einsatz) verfuegbar.

Die Demo-Version unterschiedet sich von der Einsatz-Version dahingehend, das mit ihr keine Hardware angesteuert werden kann.

Alle sonstigen Funktionen sind vollkommen identisch. Mit der Demoversion koennen Schaltzeiten erfasst, veraendert, gespeichert und geladen werden.

Bei der Ausfuehrung dieser Schaltzeiten (des Steuerprogramms) werden alle durchgefuehrten Schaltungen auf dem Bildschirm angezeigt (wie auch bei der Einsatz-Version).

Billiges CP/M von Bill Brewer

übersetzt von G. Schröder

Das Modell III Kann in seinem ursprünglichen Zustand keine CP/M-Programme laufen lassen. Aber dieser Bericht zeigt Ihnen, wie Sie eine einfache Modifikation mit zwei Chips für fünf Dollar bauen können.
Das ROM belegt im Modell III die Adressen 0000-37FF hexadezimal. Außerdem liegen die Tastaturadressen und der Bildschirmspeicher in 37FF-3FFF. Nur der Bereich von 4000 bis FFFF kann im Modell III als Lese/Schreib-RAM benutzt werden. Die Aufteilung des Speichers wird in Bild 1 gezeigt.
Das ROM liegt im Modell III leider genau dort, wo auch die Programme unter CP/M stehen würden. CP/M würde Daten an der gleichen Stelle speichern, an der das Modell III-ROM ab 0000 liegt, und CP/M lädt alle Programme, die unter ihm laufen, in den Speicher ab 100 hexadezimal. CP/M und alle Programme, die unter ihm laufen, brauchen einen durchgehenden Speicher von 0000 bis zum Ende des Speichers, wo das Betriebssystem steht.
Abgesehen von dieser Speicherabweichung ist die Modell III Hardware durchaus dazu geeignet, leicht modifizierte 8080 oder Z80 Betriebssysteme laufen zu lassen; auch CP/M.
Diese Hardware-Begrenzung im Modell III wird durch nur sehr wenige der Schaltkreise dieses Computers verursacht. Wenn man sich diese Schaltkreise ansieht, wird rasch ein schneller und leichter Weg sichtbar, um diese Hindernisse aufzuheben.
Die Modell III CPU adressiert Signale über 16 Drähte in seinem Adressbus. Ein unmodifiziertes Modell III führt seine beiden höchstwertigen Adressdrähte in die eine Hälfte eines 74LS139 Multiplexers (U58), der einen Decoder Schaltkreis darstellt. Dieser Decoder gibt Signale aus, um eine der vier 16K-Speicher- bänke einzuschalten (Bild 2).
Wenn beide der höchstwertigen Adressdrähte (A14 und A15) auf einem niedrigen logischen Level sind, schaltet der Decoder die 16K-Speicherbank ein, die das Modell III ROM sowie die Tastatur und den Bildschirmspeicher enthält. Abhängig von den binären Werten dieser beiden Adressdrähte wird eine der vier 16K-Speicherbänke im Modell III Adressenraum ausgewählt (Tabelle 1).
Ein zweiter Decoder, der die untere Hälfte desselben Multiplexers benutzt, produziert Einschalt-Signale in der 16K-Bank, die das ROM enthält. Indem er Adressdraht A14 und den 0000-3FFF-Einschalter (ROMENABLE*) benutzt, schaltet dieser Decoder entweder den 8K ROM-Chip ein, der das meiste des Modell III Basic enthält, oder eine andere Gruppe von Schaltkreisen mit einem 4K-ROM, einem 2K-ROM, der Tastatur und dem Bildschirmspeicher. Dieser Segment-Decoder wählt diese verschiedenen Routinen zur Benutzung in ihrem jeweiligen Speicherplatz wie in Tabelle 2 gezeigt.

SIXTEEN KILOBYTE ADDRESS BANK SELECTED BY MRQ*, A16, AND A15

MRQ*	A16	A15	BANK SELECTED
HIGH	EITHER	EITHER	NONE
LOW	LOW	LOW	0000-3FFF
LOW	LOW	HIGH	4000-7FFF
LOW	HIGH	LOW	8000-BFFF
LOW	HIGH	HIGH	C000-FFFF

Table 1. 16K Bank Selection by U58

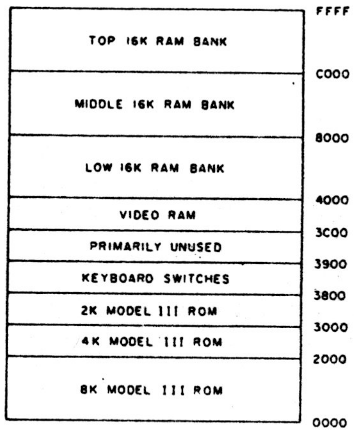


Fig. 1. Model III Memory Map

EIGHT KILOBYTE DEVICE SELECT BY ROM ENABLE* AND A14

ROM ENABLE*	A14	DEVICE SELECTED
HIGH	EITHER	NONE
LOW	LOW	8 KILOBYTE PROM
LOW	HIGH	PROM, KBD, VIDEO

Table 2. Segment Selection in Bank Containing ROM

Eine Hardware-Lösung

Die einfachste Lösung, um das Modell III zu einer CP/M-fähigen Maschine zu machen, ist, das ROM aus dem Adressenbereich auszuschalten, den die meisten Programme benutzen, die nicht für das Modell III geschrieben wurden, und dafür das RAM einzu- schalten.
Das sieht zunächst sehr einfach aus. Man kann dies erreichen, indem man die Leiterbahnen, die im Bild 2 mit 0000-3FFF und C000-FFFF bezeichnet sind, vertauscht. Somit wäre das ROM, die Tastatur und der Bildschirm in C000-FFFF und die 16K-RAM-Bank der Bank 3 würde von 0000 bis 3FFF adressiert werden.
Dieser einfache Ansatz bringt aber zwei Nachteile mit sich. Erstens können keine Programme mehr laufen, die für das Modell III geschrieben wurden, denn ROM, Tastatur und Bildschirm würden ständig aus dem Speicherbereich herausgeschaltet werden, in dem sie laufen müssen.
Zweitens würde der Prozessor - ohne entsprechende Änderungen seines Kontroll-Schaltkreises - bei jedem RESET versuchen, ein Programm an der Adresse 0000 im RAM auszuführen; es gibt keine Möglichkeit, ein Programm in das RAM zu laden, bevor der Prozessor beginnt, von ihm Instruktionen zu holen.
Sie können die ursprünglichen Fähigkeiten bewahren, wenn Sie elektronische Schalter benutzen. Elektronische Schalter können die Adressierung so ändern, daß 0000 die Startadresse des ROMs ist, wenn Programme unter TRSDOS etc. laufen und daß 0000 die Startadresse des RAMs ist, wenn Sie Programme unter CP/M laufen lassen.
Elektronische Schaltungen achten auch auf das Problem mit dem Laden des Programms: Mit einem elektronischen Schalter können Sie Programme mit Routinen aus dem ROM laden, bevor der Schalter die ROM- und RAM-Adressen tauscht. Somit versorgt ein elektronischer Schalter das Programm mit einer dauernden Kontrolle durch den Prozessor.

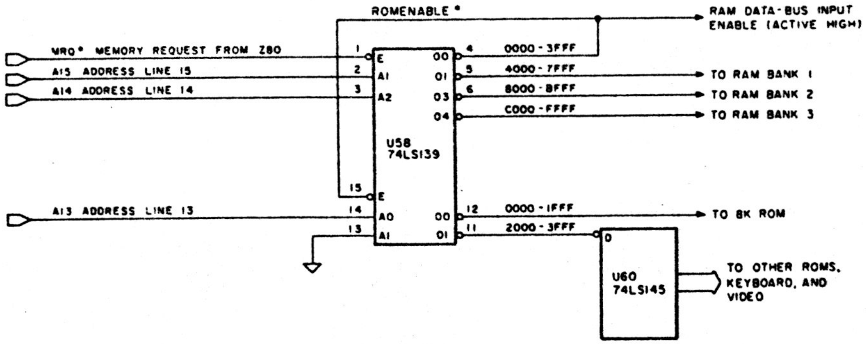


Fig. 2. Model III Address Decoder Circuit

Ein elektronischer Schalter

Ein Weg, um die elektronische Speicher-Umschaltung zu erreichen, ist, die Adressensignale des Z80 zu bearbeiten, bevor sie den Decoder erreichen. Während sich der Speicher im Standard Modell III-Modus befindet, kann der Z80 Instruktionen im Adressenbereich von 4000 bis BFFF ausführen, um die Signale von A15 und A14 zu invertieren, wenn beide WAHR oder beide FALSCH sind. Somit wird der Decoder genarrt, denn er denkt, daß eine vom Prozessor gelieferte Adresse im Bereich 0000-3FFF in Wirklichkeit im Bereich C000-FFFF liegt, und der Prozessor hat benutzbares RAM bei 0000.
Es gibt einige kommerzielle Modifikationen, die den Speicher in dieser Weise umschalten. Diese Methode hat einen Vorteil und einen Nachteil. Das Vorbearbeiten der Adressensignale ist mechanisch bequem; es kann durch eine einzelne Huckepack-Platine im Z80-Sockel erreicht werden, welche natürlich wieder den Z80 in einem Sockel aufnehmen muß.
Der Vorteil ist, daß der Computer nicht physisch verändert werden muß; die Huckepack-Platine ist keine unwiederrufliche Veränderung, denn der Z80 ist einer der wenigen gesockelten Chips im Modell III. Der Nachteil ist, daß dieses Vorbearbeiten der Signale natürlich eine Verzögerung vor dem Erreichen des Decoders und der RAM-Bänke bewirkt, wodurch RAM-Operationen unzuverlässig werden. Diese Verzögerung ist im Modell III kritischer als im Modell I, denn die Speicher-Zyklus-Zeit (memory-cycle time) ist im Modell III um annähernd 20% kürzer als im Modell I.

Ein neuer Bank-Wahl-Decoder

Eine bessere Lösung stellt der Ersatz des Multiplexer-Schaltkreises (U58) durch einen neuen Decoder-Schaltkreis dar, wobei dieser die Adresszuweisungen für ROM und RAM umschalten kann und trotzdem nicht langsamer im Ansprechen der Adressen ist als das Originalteil.
Der neue Decoder, der den 16K Bank-Wahl Decoder in der oberen Hälfte von U58 ersetzt, besteht aus zwei anwählbaren Decodern in einem 74LS156 (Demultiplexer mit offenem Collector). Einer der neuen Decoder kann das ROM mit Tastatur und Bildschirm von 0000-3FFF und die RAM-Bank 3 von C000-FFFF einschalten (wie der Original Modell III Decoder). Der andere neue Decoder schaltet das ROM + Tastatur-Matrix + Bildschirmspeicher in C000-FFFF und RAM-Bank 3 in 0000-3FFF ein. Dieser Schaltkreis, um den 74LS156 Demultiplexer (genannt U58-1) herum aufgebaut, wird in Bild 3 gezeigt.
Es ist jederzeit immer nur einer der Decoder aktiv. Ein aktiv-HIGH Signal (REMAP), extern an die Decoder herangeführt, bestimmt, welcher der beiden Decoder die Speicherbank-Auswahl vornimmt. Weil die aktiv-LOW Ausgänge der Decoder als verdrahtete OR-Glieder (wired OR) geschaltet sind und der nicht angewählte Decoder HIGH (inaktive) Ausgänge für alle hat, kann nur der gewählte Decoder ein aktiv-LOW Signal über die Bank-Wahl-Anschlüsse senden.
Wenn REMAP logisch LOW (inaktiv) ist, wird der Decoder, der in Bild 4 oben gezeigt ist, eingeschaltet; er gibt Schaltsignale aus, die mit der Original Speicheraufteilung des Modell III übereinstimmen. Ist REMAP logisch HIGH (aktiv), wird der untere Decoder eingeschaltet; er gibt Schaltsignale aus, die das ROM mit Tastatur und Bildschirm in die obersten 16K des Speichers verlegen. RAM-Bank 3 wird in die untersten 16K des Speichers gelegt.

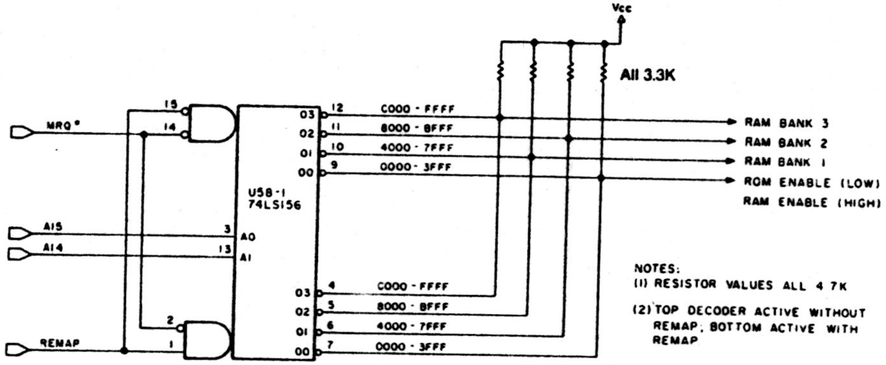


Fig. 3. Bank Decoder with External Request Line

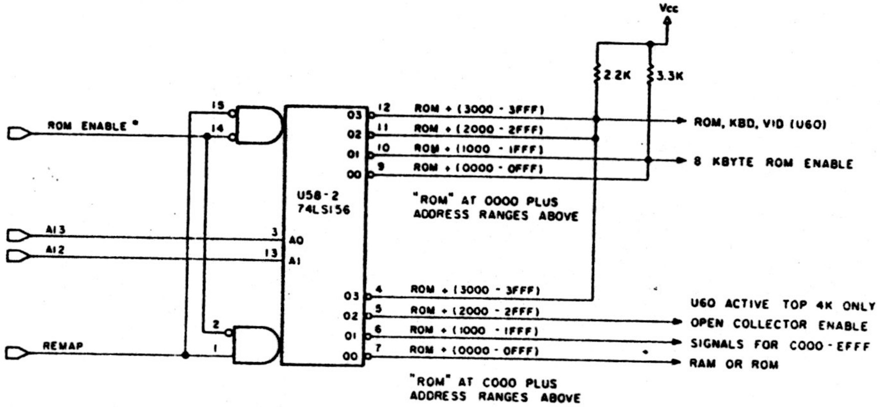


Fig. 4. Decoder for Bank Containing ROM, Keyboard, and Video

Treffen der Roboter

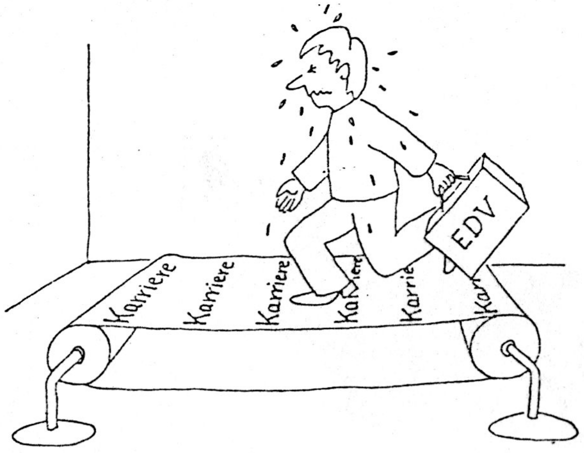
Chicago – Eine Baufirma schrieb einer Fabrik für elektronische Geräte:
»Unser Elektronengehirn hat ausgerechnet, daß sich die Kosten des von Ihnen geplanten Umbaus auf 125 000 Dollar belaufen.«
Die Firma erhielt umgehend die Antwort ihres Kunden:
»Unser Elektronengehirn hält Ihren Kostenanschlag für viel zu hoch. Wir schlagen daher sobald wie möglich ein Treffen der beiden Roboter in unserem Büro vor.«

Der zusätzliche Decoder

Ein zusätzlicher Decoder muß außerhalb von U58-1 benutzt werden, um die 8K Segment-Wahl Funktionen zu übernehmen, die vorher der untere Teil von U58 durchführte. Wie in Bild 4 gezeigt, besorgt der obere Decoder in einem zweiten 74LS156 Chip (genannt U58-2) die 8K Schaltungen für die unter Normalaufteilung zum ROM gehörenden 16K Speicher-Segmente. Der andere Decoder versorgt vier 4K Schalter im Adressbereich C000-DFFF unter der Neuaufteilung des Speichers.
Die Segment-Decoder werden folgendermaßen durch REMAP angesprochen. Ist REMAP LOW, wird das ROM mit Tastatur und Bildschirm wie im Original-Decoder geschaltet. Ist REMAP HIGH, produziert der untere Decoder vier Einschaltsignale in übereinstimmung mit Adressleitung A12. Eine dieser beiden Möglichkeiten ist immer aktiv, wenn der Prozessor C000-FFFF adressiert.
Der Draht, der "ROM + (3000-3FFF)" genannt wurde, schaltet in aktivem Zustand U60 ein, um das 4K Segment anzuwählen, welches das 2K ROM, die Tastaturmatrix und den Bildschirmspeicher enthält. Die drei zusätzlichen Signale sind Ersatz, und sie sind für die drei anderen 4K Segmente in C000-FFFF aktiv. Sie können zusammen benutzt werden, um 12K RAM außer der Reihe einzuschalten (womit wir insgesamt 60K hätten). Die Original 8K- und 4K-ROMs, die nutzlos sind, wenn sie von 0000 weggeschaltet sind, tauchen nicht mehr in der Speicheraufteilung auf. Dieser Raum ist frei für eine Speichererweiterung.
Der Ersatz von U58 durch die beiden 744LS156 Demultiplexer- Chips versorgt das Modell III mit seiner Original-Speicher- aufteilung und einer von CP/M nutzbaren Speicheraufteilung (siehe Bild 5). Außerdem erzeugen die neuen Decoder keine längere Verzögerung als die Decoder, die sie in dem jeweiligen Speicherzustand ersetzen. Statt Unzuverlässigkeit im Speicherzugriff bewirken sie sogar eine Speichererweiterung.

Das Signal, das den Speicher umorganisiert (REMAP)

Das REMAP-Signal, das zwischen den beiden Speicher Konfigurationen umschaltet, könnte von einem Schalter außerhalb des Modell III-Gehäuses ausgehen. Aber der Schalter kann nur betätigt werden, wenn der Z80 ein Programm ausführt, das in den Speicher 4000-BFFF geladen wurde (oder im ROM ist). Sonst "vergißt" der Programmzähler, welche Instruktionen er gerade ausführte.
Diese Einschränkung erfordert also, daß Sie ein REMAP nur dann ausführen, wenn die Kontrolle bei einem Programm liegt, das in den richtigen Bereich geladen wurde.
Das Signal kann von einem unbenutzten Ausgabe-Port des Modell III Logic-Boards versorgt werden. Der Ausgabebaustein U98, normalerweise benutzt, um Kassettenrekorder-Levels und ein Video Zeichengröße Signal zu senden, beinhaltet ein unbenutztes D-Flip-Flop.
Indem der unbenutzte Eingang dieses Bausteins mit Bit 0 des Data-Bus verbunden wird, kann die Ausgabe des freien D-Flip-Flop durch das niederwertigste Bit des Port EC kontrolliert werden. Ist dieses Bit LOW (inaktiv), geht der Ausgang des Flip-Flop (REMAP*) auf LOW, und die Speicher- aufteilung des Modell III ist für TRSDOS richtig. Ist dieses Bit HIGH (aktiv), wird der Remap-Befehl ausgegeben, welcher den Speicher für CP/M umorganisiert.
Das RESET* Signal von U98 ist normalerweise an Vcc (+ 5 Volt) gebunden. Die Information der beinhalteten Flip-Flops wird nicht unbedingt gelöscht, wenn Sie den Computer einschalten; auf jeden Fall wird es nie gelöscht, wenn Sie Reset von der Tastatur aus eingeben. Um sicher zu gehen, daß der Computer mit dem ROM an 0000 (Startadresse des Prozessors) startet, muß das RESET* Signal zu U98 von der Unterstützungseinheit des Prozessors kommen und zwar genau von U57 Pin 13.



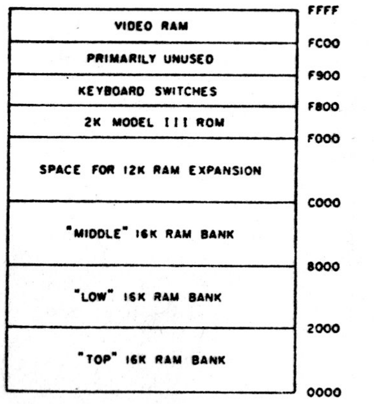


Fig. 5. CP/M-Compatible Model III Memory Map.

Physische Modifikationen

Die Decoder im Modell III zu ersetzen ist weitaus komplizierter als das Aufstecken einer Huckepack-Platine in den Z80-Sockel, obwohl an der CPU keine größeren Änderungen vorgenommen werden. Der neue Decoder-Schaltkreis wird auf einer separaten Platine plaziert, welche in einen Sockel gesteckt wird, der wiederum den Platz des ursprünglichen Decoders (U58) auf der Hauptplatine einnimmt (Fotos 1a und 1b).
Teile des Decoders, bestehend aus IC58-1, IC58-2, zwei 16-Pin-Sockeln, verschiedenen Widerständen und einem DIP-Stecker (der in den obern erwähnten, U58 ersetzenden DIP- Sockel gesteckt wird) werden auf einer passenden Lochraster- platine aufgebaut. Die Sockel für die Chips sind auf der Komponenten-Seite der Platine eingesteckt und die Verdrahtung wird auf der Lötseite mit ummanteltem Draht (Wrap-Draht) vorgenommen. Der DIP-Stecker ist auf die Platinenbahnen auf der Lötseite der Platine gelötet. Die Fotos 2a und 2b zeigen die Komponenten- und die Lötseite der Platine.
Die schwierigste Änderung ist das Entfernen des Original-Decoder-Chips (U58) und die Installation eines DIP-Sockels. Das Abschneiden jedes Drahtes des Chips und das Entfernen je eines Drahtes mit einem nicht zu stark erhitzten Lötkolben ist am wenigsten riskant. Ein Sockel, der den Decoder-Board-Stecker aufnehmen kann, wird dann anstelle des IC58 eingelötet. Dies kann ein konventioneller IC-Sockel sein, doch ein aus einzelnen Pins (IC-Kontakte Meterware) individuell aufgebauter Sockel ist erheblich sicherer und besser.
Die Reset-Leitung des Ausgabe-Latch (U98) muß auf dem Locig-Board gekappt werden. Mit isoliertem Draht, der direkt an die IC-Pins gelötet wird, werden folgende Verbindungen hergestellt:
-Data-Bus Leitung 0 von U100, Pin 14, zu dem Ersatz Port EC Latch Input bei U98, Pin 4.
-Port EC Latch Output von U100, Pin 5, zu einer Klemme auf der Decoder-Platte, welche REMAP an U58-1 und U58-2 weiterleitet.
-/RESET (RESET*) von U57, Pin 13, zum Port EC Latch Reset bei U98, Pin 1.
-Adressleitung U12 von U60, Pin 13, zur Decoder-Platte.
Ich habe die oben beschriebene Modifikation über hundert Stunden lang getestet und es ist kein Fehler aufgetreten. Die 8K und 4K Modell III ROMs sind ausgeschaltet und drei 4K Speichererweiterung Einschaltsignale funktionieren wie beschrieben.

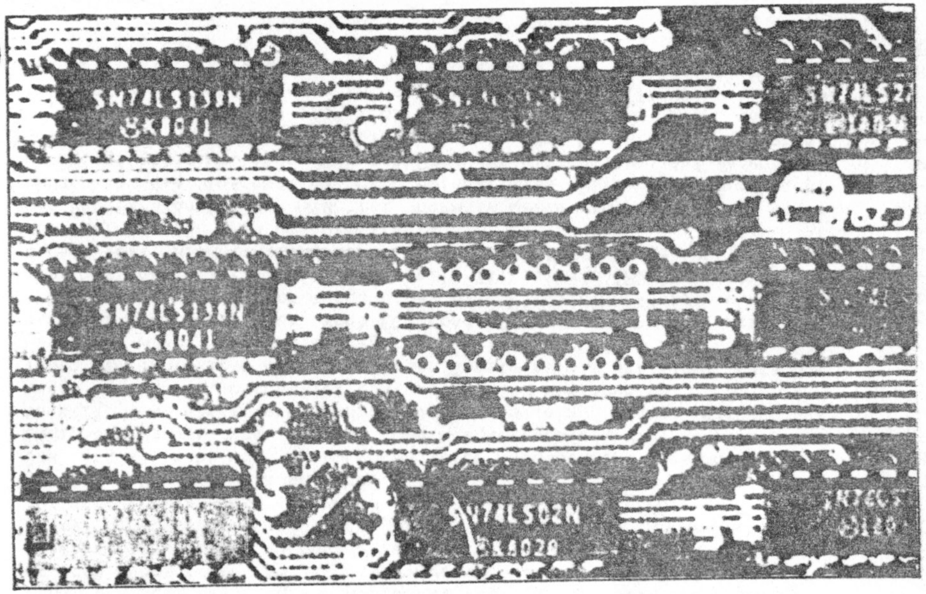


Photo 1a. Socket for New Decoder

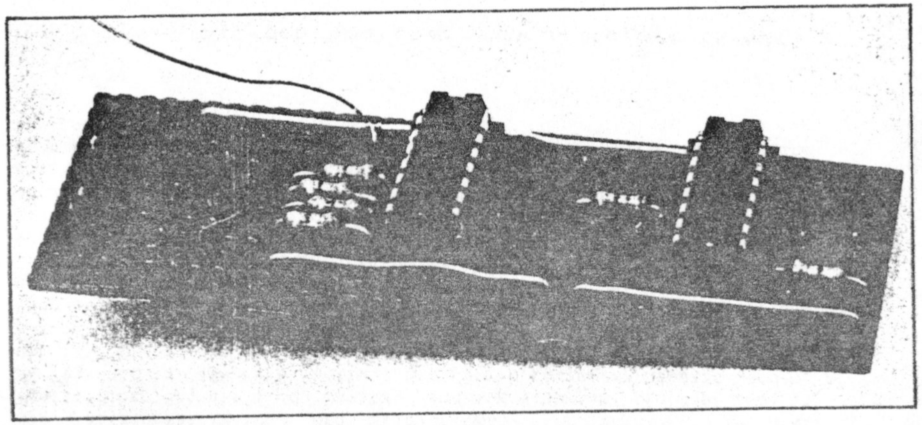


Photo 2a. Component Side of Decoder Board

Das Assemblerlisting für den Bootstrap Loader ist als "Paperware-Kopie" von Hartmut Obermann erhältlich

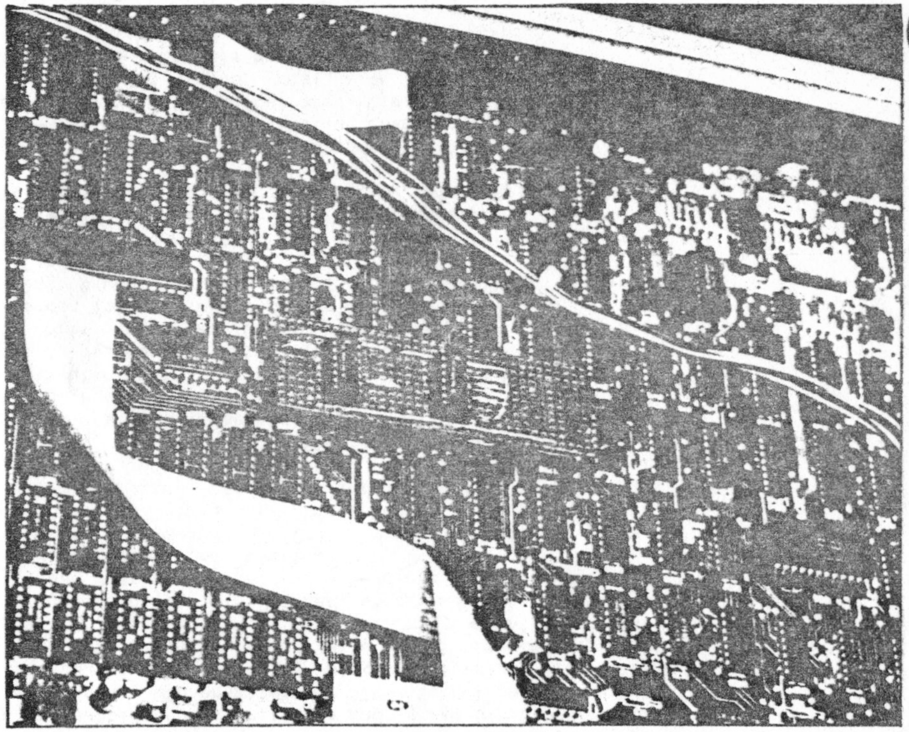


Photo 1b. New Decoder Installed

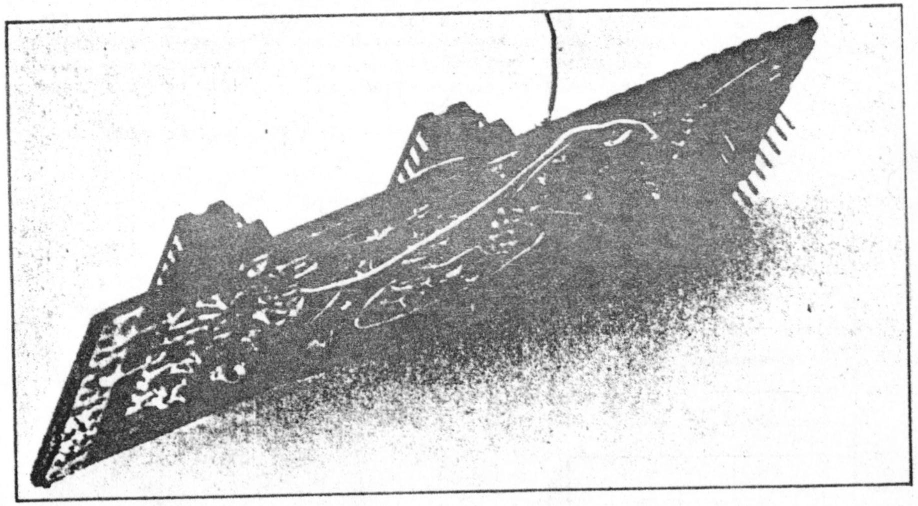


Photo 2b. Solder Side of Controller Board

Vorläufiges Design zur Speichererweiterung

Es wurden keine Tests mit einem Modell III durchgeführt, das zusätzlichen Speicher hatte. Die Speichererweiterung ist eine kostenlose Nebenerscheinung der Wahl unserer Chips für die Speicherneuaufteilungsmodifikation.
Speichenweiterungen können in derselben Technik ausgeführt werden wie die beschriebene Modifikation zur Neuaufteilung des Speichers. Stecken Sie eine Platine, die aus 8 DIP-Steckern und 16 Sockeln konstruiert ist, in die Sockel für eine der 16K RAM Banken. Verbinden Sie 8 der Sockel parallel mit entsprechenden Headern und stecken Sie die 4116 RAM Chips in sie. Sie funktionieren genau wie bei einem nagelneuen Computer.
Verbinden Sie alle Pins bis auf das CAS*-Signal (CAS* - Column Adress Strobe pin) der acht übrigen Sockel (für die acht zusätzlichen 4116s) parallel mit entsprechenden Steckern. Eine CAS-Leitung, die mit allen CAS* Pins der neuen RAM Chips verbunden ist, kann hergestellt werden, indem die drei 4K Einschaltleitungen des U58-2 als wired OR verdrahtet werden. Um das neue RAM Bank-Wahlsignal CAS$* zu erhalten, wird das daraus resultierende Signal mit dem Model III CAS* über ein OR-Glied verbunden (siehe Bild 6).
Nur weiteres Konstruieren und Austesten kann sicherstellen, daß diese Speichererweiterungsmodifikation über längere Zeiträume brauchbar ist. Auf jeden Fall sollte die Stromversorgung, die mit einer 50-Hertz Leitung arbeiten kann, die neuen RAMs mitversorgen können. Außerdem dürfte in diesem gut durchlüfteten Teil des Gehäuses kein Hitzestau auftreten.
Äußerste Vorsicht ist die einzige Garantie für einen Erfolg bei der Arbeit mit der CPU-Platine. Alle Verbindungskabel und Verbindungen, die in der Platine stecken, sind sehr dünn und sehr zerbrechlich.
Versuchen Sie sich nicht an dieser Modifikation, wenn Sie ein Neuling im Arbeiten mit Lötzinn sind.

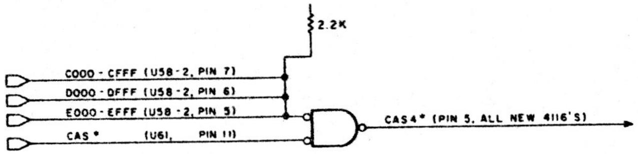


Fig. 6. Preliminary Logic for a New RAM Bank-Select Signal

Software Kontrolle

Die Software, die die Speicherumstellung schaltet, ist einfach. Um den Speicher zu re-organisieren, schreiben Sie die folgenden Instruktionen in Ihr Bootstrap Loader Programm:

```
  LD    A,9
  OUT   (0ECH),A
```

Dementsprechend benutzen Sie folgende Zeilen, um den Originalzustand Ihres Computers wiederherzustellen:

```
  LD    A,8
  OUT   (0ECH),A
```

Das am wenigsten signifikante Bit des Ausgabe-Bytes schaltet den Speicher um. Andere Bits in diesem Byte betreffen Video Zeilenlänge und Zeichensatz, Video Warte-Status, Kassettenmotor Schaltung und externe I/O Bus Operationen. Sind diese Bits aus (off), ist das Modell III in folgendem Zustand:
64 Zeichen/Zeile, japanischen Kana Spezialzeichensatz, kein Video Warte Status und Kassettenmotor sowie I/O Bus sind ausgeschaltet. Mit Bit 3 an wird der griechische Zeichensatz angewählt.
Die meisten mit kommerziellen Modifikationen gebrauchten CP/M-Versionen können leicht an diese Modifikation angepaßt werden. Durchsuchen Sie nur den Bootstrap Loader und das Basic Input/Output System (BIOS, das maschinenabhängige Modul in CP/M) nach den Speicherumstellungs-Routinen, die für die jeweilige Modifikation gelten und ersetzen Sie diese durch die oben angeführten.
Als Alternative können Sie ein neues BOOT (s. unten) und BIOS für die Standard CP/M-Version schreiben. Der unten gelistete Bootstraploader kann in ein vorhandenen BIOS eingebaut werden.
Ein Bootstrap Loader für CP/M oder jedes andere Programm, das Sie auf der Diskette plaziert haben, ist im Listing gezeigt. Der Loader muß auf Track 0 , Sektor 1 einer Diskette im Modell III Standard Format von 18 256-Byte Sektoren per Track stehen. Das zu ladende Programm muß auf Track 0, Sektor 2 beginnen.
Die Speicheradressen, in die das Programm geladen wird, sind immer am Anfang des Programms angegeben und dort können Sie diese leicht verändern, um das Programm anzupassen. Wenn Ihre Diskette eingelegt ist, bringt das Modell III ROM bei einem RESET den Loader in das RAM bei 4300 hexadezimal und übergibt ihm die Kontrolle. Er lädt dann Ihr Programm und dieses Programm gibt Ihnen Möglichkeiten, die Sie vorher nicht hatten.

Bild 1: Modell III Speicher Aufbau
Bild 2: Modell III Adressen Decoder Schaltkreis
Tabelle 1: 16K Bank-Wahl beim U58
Tabelle 2: Segment-Wahl in ROM-Bank
Bild 3: Bank Decoder mit externer Wahlleitung
Bild 4: Decoder für Bank mit ROM, Tastatur und Bildschirm
Bild 5: CP/M-Kompatibler Modell III Speicher Aufbau
Bild 6: Vorläufige Logik für ein neues RAM Bank-Wahl Signal
Foto 1a: Sockel für den neuen Decoder
Foto 1b: Neuer Decoder installiert
Foto 2a: Komponenten Seite des Decoder Boards
Foto 2b: Lötseite des Controller (Decoder) Boards

Extra RAM oder Ein sicheres Plätzchen im TRS80 Modell

Artikel aus 80-U.S.Jurnal von Truman Krumholz
sehr frei übersetzt von Hartmut Obermann

In diesem Artikel möchte ich den Aufbau eines Memory-Boards beschreiben, das den Zwichenraum zwischen ROM und Tastatur ausfüllt. Dieses zusätzliche RAM ist besonders zur geschützten Unterbringung von Maschinenroutinen geeignet, da es nicht vom System bzw. DOS benutzt wird. Das Board benutzt die Adressen 3000H-377FH und stellt 1920 Bytes freien Speicherplatz zur Verfügung.

Das Board läßt sich schnell und problemlos auf einer Lochrasterplatine in Fädeltechnik aufbauen. Der Anschluß an das System erfolgt über den meist unbenutzten Screenprinteranschluß des Expansionsboards *1. Dazu benötigt man entweder ein selbstgestricktes Kabel mit passendem Stecker oder man lötet die Leitungen direkt auf. Als Stromversorgung werden +5 Volt bei 0.5 Ampere benötigt *2. Als Speicherchips werden zwei 2114s (1024*4 Bit Static RAM) verwendet.

Die Adressdecodierung wurde mit gängigen TTL-Bausteinen (7400 und 7410) aufgebaut. Alle Leitungen zwischen dem Board und dem Computer sind gebuffert (74LS367 und 8216) um Überlastung und Beeinflussung des Systems zu vermeiden. Die 8 "Pullup"-Wiederstände (je 1 kΩ) werden benötigt, da ansonsten die High-Signale auf dem Datenbus manchmal nicht die geforderten Werte erreichen.

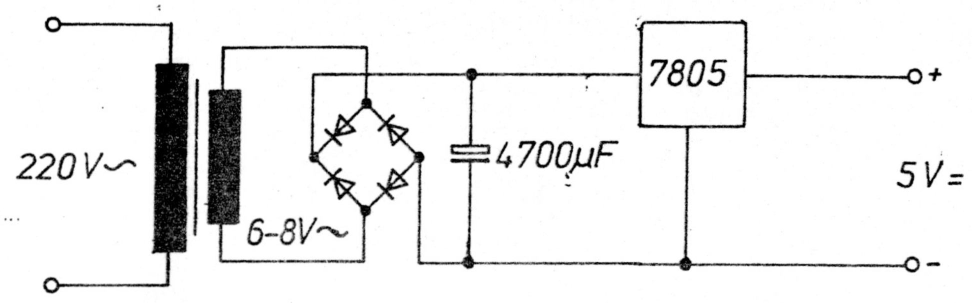
Das externe Powersupply *2 besteht aus einem 6-8 Volt Transformator (0.5-1 Ampere belastbar), einem Brückengleichrichter und einem 4700 µF Siebkondensator. Als Spannungsstabilisator dient ein IC 7805 Spannungskonstanter. Die Versorgungsspannung sollte an jedem Chip mit einem Keramikkondensator (0.1 µF) von Störungen frei gehalten werden.

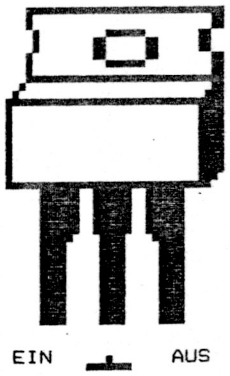
Nach dem Zusammenbau von Powersupply und Memory Board sollte als erstes eine genaue Sichtkontrolle der Schaltung erfolgen. Auf der Speicherplatine kann man außerdem mit einem Ohmmeter feststellen, ob die 5 Volt-Leitungen einen Kurzschluß haben. Erst nach diesen Kontrollen sollte man die Stromversorgung anschließen. Als letzte Überprüfung schaltet man die Versorgungsspannung ein und überprüft alle Leitungen, die später zum Computer führen darauf, daß keine mehr als 5 Volt führt.

Nach dem endgültigen Anschluß der Erweiterung kann man das nun zusätzlich vorhandene RAM durch das unten stehende Programm auf seine Funktionsfähigkeit überprüfen.

*1 Natürlich ist auch der Anschluß über den geplanten ECB-Bus realisierbar
*2 Siehe Schaltbild Nr. 2

```
 00 'MEMORY-TESTROUTINE FUER DAS ZUSATZ-RAM 3000H-377FH
     Langsam aber einfach! Am besten mit ZBASIC oder ACCEL
     Compilieren!!!
 10 CLS
 20 FOR X = 12288 TO 14207
 30      FOR Y = 0 TO 255
 40               POKE X,Y
 50               IF PEEK (X) <> Y THEN GOSUB 100
 60      NEXT Y
 70 NEXT X
 80 END
 90 'FEHLERBEHANDLUNG
100 PRINT"SPEICHERFEHLER IN ZELLE ";X;" SOLLWERT ";Y;" ISTWERT ";
110 PRINT PEEK (X)
120 PRINT, "WEITER MIT <ENTER> !"
130 IF INKEY>CHR13) THEN GOTO 130 ELSE RETURN
```



R E G L E R

UA 78 05
UA 78 06
UA 78 08
UA 78 10
UA 78 12
UA 78 15
UA 78 18
UA 78 24

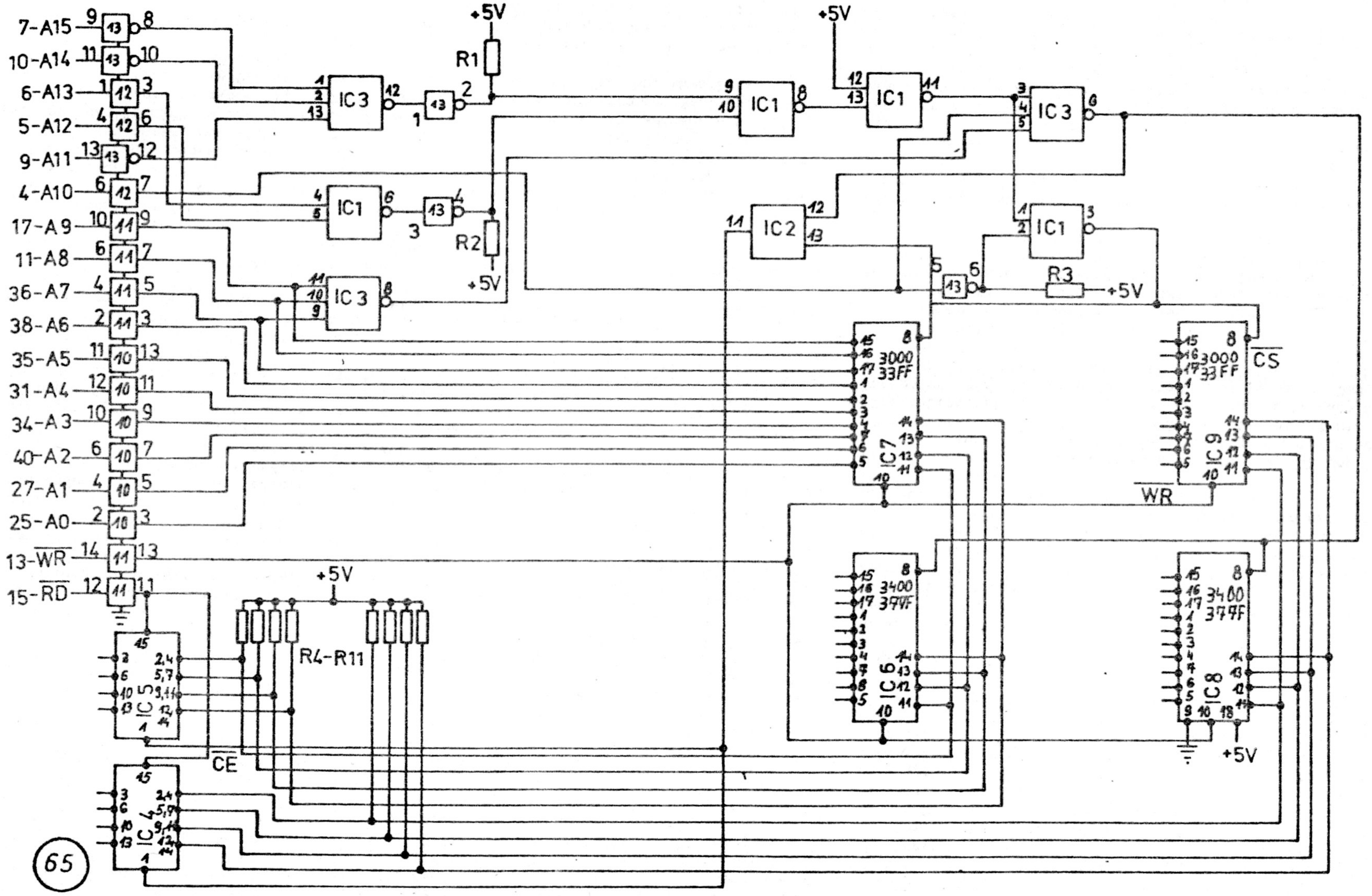
Die Originalzeichnung des Memory Board war wegen ihrer schlechten Qualität nicht zur Veröffentlichung geeignet. Ich bedanke mich vielmals bei Manfred Held, daß er eine reproduzierbare Zeichnung erstellte.
Die Zeichnung des Netzteils stammt von mir. Wie man feststellen kann, sind meine Fähigkeiten auf dem Gebiet des technischen Zeichnens nicht allzusehr ausgeprägt. Ich hoffe, man kann erkennen was gemeint ist.

Hartmut Obermann

San Diego – Sechs Kühlschränke, die hier in einer Oberschule aufgestellt werden sollten, waren mit Drei-Pol-Steckern ausgerüstet. In der Schule gab es jedoch nur Zwei-Pol-Steckdosen. Am nächsten Tag kamen zwei Elektriker. Der eine versah die Kühlschränke mit Zwei-Pol-Steckern. Unterdessen tauschte der andere die Steckdosen an der Wand aus und montierte überall Drei-Pol-Steckdosen.

Ein »völlig neues Frühstücksgefühl« verspricht eine amerikanische Elektrowarenfabrik den Käufern ihres Toasträsters, der je nach Wahl byzantinische, florentinische oder spätgotische Ornamente ins Brot brennt und überdies während des Röstens eine Mozartmelodie abklimpert.

Als der Lehrer einer Münchner Berufsschule die Prüfungsfrage »Was ist ein Lichtjahr?« stellte, antwortete der Prüfling: »Das ist die Stromrechnung für 365 Tage.«



Bauteile:

R1-R11 = 1KΩ

IC 1 = 74LS00
IC 2 = 74LS08
IC 3 = 74 LS10
IC 4 = 8216
IC 5 = 8216
IC 6 = 2114
IC 7 = 2114
IC 8 = 2114
IC 9 = 2114
IC 10 = LS367
IC 11 = LS367
IC 12 = LS367
IC 13 = LS368

zu IC 8216:

8216		
RD	CE	
0	0	Mem→Buss
1	0	Mem←Buss
0	1	off
1	1	off

Gerald Schröder:
Genie IIs - Erfahrungsbericht

Wie wohl inzwischen ein großer Teil von Euch mitbekommen hat, habe ich dauernd mit technischen Mängeln an meinem Genie I gekämpft. Der letzte Stand stellte sich so dar: die Laufwerke konnten sich nicht auf eine einheitliche Spurhaltung einigen und ein Wackelkontakt an der Video-Platine verwandelte den Bildschirm in ein Tummelfeld aller möglichen Grafikzeichen, wobei nur ein kräftiger Schlag auf das Gehäuse Abhilfe schaffte. Entnervt beschloß ich gemeinsam mit meinem elterlichen Geldgeber, die gesammelten Neuheiten der Technik in Form eines Genie IIs zu erwerben.
Glücklicherweise fand sich ein Händler in der Nähe (nur ca. 60 km entfernt), der uns bei einer Besichtigung der Anlage diese gleich andrehte. Für vier Riesen durften wir einen Computer mit zwei eingebauten Slimline-Laufwerken je 80 Track DS/DD und einen Taxan-Monitor mitnehmen.
Zum Aufbau müssen nur die separate Tastatur und der Monitor werden an das massive (und dementsprechend schwere) Metallgehäuse angeschlossen werden. Die Anschlüsse sind absolut narrensicher, nur die Kabel leider nicht: das Tastaturkabel gehört zur Gattung der Flachkabel und ist dementsprechend inflexibel. Die Länge des Kabels eröffnet drei Möglichkeiten: die Tastatur hinter das Hauptgehäuse zu stellen, genau davor oder etwas rechts versetzt. Die erste Möglichkeit scheidet aus, und die beiden anderen sollte man auch vergessen, denn die Laufwerke liegen (im wahrsten Sinne des Wortes) auch rechts und damit ist das Laufwerk 0 mit davorstehender Tastatur nicht mehr erreichbar. Also das Kabel dehnen so weit es geht und die Tastatur nach links.
Aber wir haben ja noch ein Kabel: das vom Monitor zum Computer. Die Verbindung am Computer rechtfertigt den Slogan des Herstellers TCS: "Made in Germany" <=> Qualität, doch die andere Seite ist noch original Hongkong-made, was wohl erklärt, warum erst das dritte (!!) Kabel nicht auseinanderfiel. Allerdings erklärt es nicht die laschen Qualitätskontrollen (von denen später noch die Rede sein wird).
Fertig angeschlossen bietet sich ein tolles Bild: eine große, halbwegs flache Tastatur, ein wuchtiges Hauptgehäuse und ein flimmerfreier Bildschirm (Video-Ausgang verbessert); dazu der Sound eines leicht rauschenden Lüfters und der leise ratternden Laufwerke.
Doch der Schein trügt etwas: die Tastatur erfüllt nicht alle Erwartungen. Quantität ging vor Qualität. Sehr schön die Zusatztasten: zwei Funktionstasten ergeben in Verbindung mit den Ziffern im abgesetzten Ziffernblock (leider ohne zweite Enter-Taste) sechzehn Funktionen, Reset wird über zwei gleichzeitig zu betätigende Tasten gegeben, die Umlaute sind gut erreichbar, im Gegensatz zu den Cursortasten zur Horizontalsteuerung im Ziffernblock.

Die Qualität läßt zu wünschen übrig: ähnlich dem C64 "schwimmen" die Tasten etwas wie in Gummi, außerdem habe ich (subjektiv) den Eindruck, als ob sie "haken": die alte Genie-Tastatur war tipp-freundlicher...
Eine Sondertaste fehlt noch in der Aufzählung: die "LSP"-Taste (steht wohl für Lower - SPeed), neben der rechten Shift-Taste. Sie rastet ein, wenn man sie drückt, und schaltet damit die Geschwindigkeit des Prozessors von 6 auf 1.77 Mhz zurück. Damit habe ich auch einen Hauptpunkt angesprochen, der dieses Gerät so attraktiv macht: die Geschwindigkeit. Scripsit ist unter "Höchstgeschwindigkeit" fast nicht benutzbar. Der Cursor springt nur so über den Bildschirm und ein einzelner Buchstabe kann kaum getippt werden. Also muß man sich umgewöhnen: schreiben mit 1.77 Mhz und alle zeitraubenden Aktionen (Replace, Blättern) mit 6. Selbst die Diskettenoperationen laufen etwas schneller ab, obwohl die Laufwerke natürlich hemmend wirken.
Offensichtlich ist die hohe Geschwindigkeit vor allem im Basic. Spiele erreichen leicht die Qualität der in Maschinensprache geschriebenen. Nur eines vermißte ich sofort: den Sound. Kein Tongenerator lenkt vom "ernsthaften Arbeiten" ab. Vielleicht läßt sich der Ton über einen extern anschließbaren Recorder oder Lautsprecher ausgeben, aber dokumentiert ist diese Möglichkeit nicht.
Natürlich reicht die hohe Geschwindikeit nicht aus, diesen Computer zu rechtfertigen. Also Punkt zwei: TCS ist einfach unserem Hardware-Zirkel unter Leitung von Walter Zwickel zuvorgekommen und hat das Steckkarten-System benutzt. Nach öffnung des Gehäuses ist eine gähnende Leere sichtbar. In meinem Gerät sind noch fünf der zehn Steckplätze frei (Grundgerät ohne Floppy-Controller mit sechs freien). Bis jetzt bietet TCS folgende Steckkarten an: Grafikkarte (80x24 Zeichen, 480x192 Punkte), SIO/PIO-Karte (zwei serielle und zwei parallele Schnittstellen), RAM-Karte (192 Kb, höchste Speicherkapazität 832 Kb), Host-Karte (zwei Harddisks anschließbar), ROM-Karte (Erweiterung des ROM auf bis zu 4x32 Kb). Außerdem kann die Taktfrequenz auf 8 Mhz erhöht werden.

Übrigens: wer sich so etwas lieber mal kurz selbst zusammenlöten möchte, dem empfehle ich die Technische Beschreibung zum Genie IIs (39 DM, von Trommeschläger). Dort sind alle Platinen mit Bauteilen etc. auf ca. 100 Seiten (für mein Laienauge) ziemlich genau beschrieben.
Wie versprochen möchte ich noch einmal auf die Qualität eingehen. Natürlich wollte ich meinen Drucker nicht ungenutzt rumstehen lassen. Also besorgte ich mir ein Druckerkabel zum Preis von 95 DM bei meinem Händler (Materialkosten bei Conrad-electronic unter 50 DM). Doch mein Drucker spuckte nur Müll aus und so schickte ich das Ding zurück. Das zweite Kabel verschwand angeblich auf dem Postweg, aber das dritte lief wenigstens, allerdings nur mit 7 Bit. Nun kam ich darauf, den Ausgang am Computer durchzprüfen. Und siehe da: statt den versprochenen 8 kamen nur 7 Bit heraus. Ich kann noch nicht sagen, was das zu bedeuten hat, aber ich werde TCS einen netten Brief schreiben müssen.
Nun will ich zum letzten Punkt kommen: der Software. In einem Zusatz-ROM befindet sich im Genie IIs ein tolles Software-Paket: Monitor, Texteditor und Assembler und ein kleiner Disassembler. Nach Aufruf wird dieses Paket in den oberen Bereich des Speichers gelegt. Allerdings fehlen hier die Diskettenbefehle, so daß ein Zusatzprogramm auf der mitgelieferten Systemdiskette zu finden ist, das diese beinhaltet. Doch dieses Super-Paket löst sich mit zunehmender Arbeitsdauer in Luft auf. Angeblich können Programme bearbeitet werden, die in den Bereich 5300-D000H laden. Doch auch bei Programmen, die diese Grenzen nicht erreichen, hängt sich der Monitor auf. Außerdem scheint der Assembler etwas gegen Source-Codes zu haben, die über ein NOP hinausgehen. Sind beim Start des Programmpakets noch andere DOS-Hilfsprogramme im Speicher, kann man herrliche Effekte erzeugen ("blindes" Eingeben von Filenamen ohne Cursor, toll für die Password-Freaks), von denen nichts im Handbuch steht.
Dieses Handbuch (besser gesagt deren zwei, für DOS und Basic) ist auch so 'ne Sache, natürlich auch die Technische Beschreibung. Angeblich können Controller und Laufwerke alles lesen, von 35/SD/SS bis 80/DD/DS, aber es ist mir mit keiner der angegebenen PDrive-Einstellungen gelungen, 40/SD/SS zu lesen. Auch die beschriebenen Prozeduren zum Booten von 40er-Disketten sind absolut wirkungslos. Das DOS-Handbuch weiß außerdem nur in einem Vorspann über das mitgelieferte GDOS 2.4 zu berichten, welches angeblich für alle Genie-Rechner geeignet ist. Das Handbuch selbst hört beim S-DOS 3.0b für den Speedmaster auf. Die letzte Auflage fehlt scheinbar noch.
Dafür werden einige neue Programme vorgestellt: z.B. zwei Grafikpakete für hochauflösende Grafik bei Genie IIs und IIIs. Oder KEY24, ein Programm, das sich selbst nach HIMEM verschiebt und neben Druckertreiber einen Bildschirmeditor zur Verfügung stellt, der allerdings nicht sehr komfortabel ist, dafür aber im DOS, im Basic und in allen Programmen ohne eigene Eingabe-Routine zur Verfügung steht (da sollte sich Arnulf mit seinem HDOS noch mal hinterklemmen).
Andere Programme unterstützen den zusätzlichen Speicher (falls vorhanden) und die Funktionstasten (sehr einfach programmierbar). Bildschirm- und Tastaturtreiber wurden abgeändert (steht da jedenfalls). Und scheinbar auch das Mini-Superzap, hier DDE (=Disk Daten Editor) genannt, denn es läuft nicht mehr. Im Modifiziermodus hängt es sich auf (wie vieles auf dieser Diskette).
Dann ist da natürlich noch die Kompatibilität: Im Basic geht es noch, obwohl Dancing Demon einige komische Sachen macht. Aber mit Assembler-Programmen sieht es schlimm aus. Auf dem Genie I vollkommen lauffähige Programme müssen über das Level II-Basic gestartet werden und die Abspeicherung von Highscores klappt auch nicht immer. Aber zum Laufen bekommen habe ich bis jetzt alle, wenn auch mit Komfort-Einbußen. Nur alles Bootbare kann ich vergessen.
Fazit: Trotz aller negativen Punkte möchte ich nicht zu meinem alten Genie I zurückkehren. Das flimmerfreie Bild, die Umlaut- und Funktionstasten, der Bildschirmeditor, die 1.4 Megabyte Diskettenspeicher (mit den alten SD/SS-Disketten!) und die Erweiterungsmöglichkleiten ohne das finger- und nervtötende Löten überzeugen mich von dem System. Ab und zu vermisse ich die Töne, aber die kann ich auch selbst machen, und schließlich war die Geräuschkulisse oft nervtötend.
Außerdem: wer möchte nicht Intruders mit 6 Mhz spielen. So schnell bin ich noch nie gestorben.

Nachlese

Kurzerklärungen für Programme

Hallo Leute, mir scheint, daß viele von Euch die gleiche Idee mit den Programmkurzerklärungen auch schon hatten.

Ich werde mich also in nächster Zeit über die Kurzerklärungsbibliothek hermachen und Euch in den nächsten INFO's die Resultate, in Form einer Aufstellung von Kurzanleitungen auf Karton anbieten.

Bis dahin könnt Ihr Euch aus der Clubbibliothek die Anleitungen zukommen lassen. Die Auflistung der entsprechenden File findet Ihr in diesem INFO.

Wenn Ihr im Besitz von Anleitungs-File seit, die nicht in der Auflistung angeführt sind, seit bitte so nett und sendet diese an die Clubbibliothek.

Bis zur 1. kartonierten Kurzerklärung

J. Neuder

Im Artikel "Shift-Trick" wird angesprochen, daß sich das L2-Basic unserer Computer verändern läßt und dann in 2532-Eprom's abgespeichert werden kann. Wenn im Club jemand solche Änderungen vornehmen will und keinen Eprommer besitzt, kann er sich an mich wenden. Ich bin gerne bereit, kostenlos Eproms nach eigenen Wünschen zu brennen. Als Beispiel habe ich mir den teilweise hässlichen Zeichensatz meines Genie II völlig neu gestaltet.

Peter Spieß

Zum Kampf gegen die Rattenplage wird in Japan ein elektrisch betriebener Apparat angeboten, der in regelmäßigen Abständen »Miau«-Geräusche ausstößt und außerdem mit funkelnden Katzenaugen bestückt ist.

Heilige Hardware

MATTHÄUS, Markus, Lukas und Johannes würden staunen über das, was Pfarrer John Derksen in Kemptville in Kanada mit einem Computer zuwege bringt. Der Geistliche hat Bibelstellen, Bücher aus seiner Bibliothek und eine Fülle weiterer biblischer Informationen in einen Computer eingespeist, mit dem er seine wöchentlichen Predigten recherchiert. Anschließend schreibt er die Reden auf einem mit dem Computer gekoppelten Textverarbeitungsgerät.

In seinem Studierzimmer hat Derksen Geräte im Wert von Tausenden von Dollar installiert. Mit Hilfe seiner Computerdateien lokalisiert er schriftliche Dokumente, Bibelstellen und Informationen für besondere Feiertage und Anlässe. Das geht schneller als das herkömmliche Suchen und Nachblättern.

Derksen macht sich zunächst zu seinem Thema eine Gliederung und skizziert in großen Zügen den Inhalt. „Allmählich wächst die Predigt, und mit dem Textprozessor kann man ganze Absätze hinzufügen, tilgen oder verschieben", sagt er. Für ihn gehöre das handschriftliche Ausarbeiten von Predigten der Vergangenheit an. *Canadian Press*

Notausrüstung

MEIN Mann war ganz unglücklich, als er einsehen mußte, daß die Taschenrechner und Computer seinen guten alten Rechenschieber zum Museumsstück gemacht hatten. Er warf ihn jedoch nicht weg, sondern ließ ihn an einer Wand seines Büros hinter Glas anbringen mit der Aufschrift: „Bei Stromausfall Scheibe einschlagen!" Y. R.

MIR GEFÄLLT der Mann, der immer wenn er in einem Einreiseantrag auf die Frage nach der Rassenzugehörigkeit stieß, nur das Wort „Mensch" eintrug. P. T.

BÖRSE

WER HAT WAS -- WER SUCHT WAS --

In der heimischen Bastelstube entsteht zur Zeit ein Druckerspooler mit max. 64k Speicher. Dieser Spooler ist für jeden Drucker verwendbar und wird zwischen Centronic's-Schnittstelle und Drucker gesteckt. Das Gerät arbeitet mit einer eigenen CPU und selbstständigen Monitorprogramm. Der Preis wird aber nur einen Bruchteil eines Industriegerätes kosten. Falls im Club Interesse besteht, bitte ich um Nachricht.

Weiterhin habe ich jetzt eine externe Grafikkarte mit einer Auflösung von 512 * 512 Bildpunken lauffähig. Sie wird über die Ports 8ØH - 8FH angesprochen und ist daher universell für jeden Z80-Computer einsetzbar. Außerdem können unabhängig voneinander zwei Bildschirmseiten angesprochen werden. Die Umschaltung erfolgt über den Port 9ØH. Die Karte enthält einen eigenen ASCII-Zeichensatz und hat eine Zeichengeschwindigkeit von max. 1,5 Mill. Bildpunkte / Sekunde. Testbericht folgt in Kürze.

Mit besten Grüßen

[signature]

Suche Kontakt zu Leuten,
die Spaß am Lösen von
A d v e n t u r e s
haben.
Gerald Schröder

Zu verkaufen !!!

1. Ein Expander EXP1 von RB-electronic (Floppy- und Druckerschnittstelle)
2. Zwei Laufwerke Tandon TM100 40 Track / DS / DD, normale Bauhöhe
3. Ein Laufwerk Philips 3114 80 Track / DS / DD, 2/3 Bauhöhe

Alle Preise auf Anfrage bei: Gerald Dreyer
Am Speiergarten 8
6200 Wiesbaden/Bierstadt
☎ 06121 / 508218

Wer kann mir die Ausgaben August und November 1984 der 80Micro fuer ein paar Tage ausleihen ?

Klaus Hermann

~~Ich habe einen neuen und originalverpackten Magnetkartenleser der Firma Redaktron zu verschenken(!). Der Neupreis ist 1200 - Ich habe das Geraet vor ca. 2 Jahren fuer 280.- aus einer Konkursmasse gekauft (inkl. 100 Magnetkarten mit 10.4 KB je Karte). Der Leser ist einschliesslich Controller und Handbuch verpackt und nach Anschluss einer Versorgungsspannung betriebsbereit. Man muss sich allerdings ein Anpassungs-Interface bauen und entsprechende Software schreiben, also sehr viel Zeit investieren. Aus diesem Grunde mochte ich das Geraet nur an einen wirklich interessierten 'Freak' abgeben. Da das Geraet und die Verpackung recht gross sind, bitte ich um Erstattung der Porto-Kosten (ca. 20 bis 25 DM)~~

Harald Trapp

ist schon weg! P.S.

Wer hat folgende Sachen für 80 Track DS/DD:
Superutility (oder weiß die richtige Einstellung)
LDOS
Newdos

Wie lese ich auf 80 Track DS/DD stinknormale 40/SD/SS Disketten? Ich bekomme nur "Lesefehler Inhaltsverzeichnis"

Wie boote ich auf Laufwerken mit 80/DS/DD Disketten mit 40 Spuren SD oder DD? Die Software-Lösung von Jens läuft bei mir nicht.

Wer hat ein Programm, das den Start über das Kassettenbasic nur simuliert, ohne über <Break und Reset> zu gehen? Das ROM-Listing liefert leider nur Anhaltspunkte, denn viele Operationen sind scheinbar überflüssig (mir ist ihr Sinn jedenfalls schleierhaft).

Gerald Schröder

FRAGEN, FRAGEN, FRAGEKASTEN

PASCAL - PROBLEME

Beim Arbeiten mit ALCOR-PASCAL Version 1.2 für das Modell I bin ich auf folgende Probleme gestoßen:

1. Offensichtlich erkennt der Compiler bei der Verwendung von Unterbereichstypen diese nicht immer richtig.
Um dies zu testen, habe ich folgendes kleine Programm zur Analyse eines einzugebenden Textes geschrieben und getestet:

```
(*****************************************************************)
(*                                                               *)
(*    PROGRAMMNAME :    TEXTANAL                                 *)
(*    DATUM :           3. 4. 1985                               *)
(*                                                               *)
(*    BESCHREIBUNG :    DAS PROGRAMM LIEST EINEN TEXT EIN        *)
(*    UND BESTIMMT DIE GESAMTZAHL DER ZEICHEN, DIE               *)
(*    ANZAHL DER BUCHSTABEN UND DIE ANZAHL DER ZIFFERN.          *)
(*    DER TEXT WIRD MIT EINEM $-ZEICHEN ABGESCHLOSSEN.           *)
(*    DAS PROGRAMM WIRD DURCH EINGABE EINES '.' BEENDET.         *)
(*                                                               *)
(*                                                               *)
(*****************************************************************)
(*                                                               *)
PROGRAM  TEXTANAL (INPUT,OUTPUT);

VAR
  C1,C2    : CHAR;
  BUCHSTAB : INTEGER;
  ZIFFER   : INTEGER;
  GESAMT   : INTEGER;

BEGIN
  REPEAT
  BUCHSTAB := 0;
  ZIFFER   := 0;
  GESAMT   := 0;

  WRITELN ('ZUM BEENDEN DER EINGABE $ EINGEBEN UND ENTER !!');

  READ (C1);

  WHILE C1 <> '$' DO
    BEGIN
    IF (C1 IN (.'A'..'Z'.)) THEN BUCHSTAB := BUCHSTAB +1;
    IF (C1 IN (.'0'..'9'.)) THEN ZIFFER := ZIFFER + 1;
    GESAMT := GESAMT + 1;
    READ (C1);
    END;

  WRITELN;
  WRITELN ('GELESEN WURDEN ',GESAMT,' ZEICHEN.');
  WRITELN ('DAVON WAREN ',BUCHSTAB,' BUCHSTABEN UND ',ZIFFER,
          ' ZIFFERN.');
  READ(C2)
  UNTIL C2 = '.'
END.
```

Damit ergaben sich mit folgenden Eingaben die Egebnisse (wobei man nach dem Abschluß der Eingabe des Strings mit (ENTER) nochmal (ENTER) für die READ(C2)-Anweisung drücken muß) :

```
ABCDEFGHIJ$
GELESEN WURDEN       10 ZEICHEN.
DAVON WAREN       10 BUCHSTABEN UND        5 ZIFFERN.

ABCDEFGH$
GELESEN WURDEN        8 ZEICHEN.
DAVON WAREN        8 BUCHSTABEN UND        5 ZIFFERN.

ABC$
GELESEN WURDEN        3 ZEICHEN.
DAVON WAREN        3 BUCHSTABEN UND        1 ZIFFERN.

A$
GELESEN WURDEN        1 ZEICHEN.
DAVON WAREN        1 BUCHSTABEN UND        1 ZIFFERN.

ABCDEFGHIJKLMN$
GELESEN WURDEN       14 ZEICHEN.
DAVON WAREN       14 BUCHSTABEN UND        6 ZIFFERN.

1234567890$
GELESEN WURDEN       10 ZEICHEN.
DAVON WAREN        0 BUCHSTABEN UND       10 ZIFFERN.

!"#%&'():-$
GELESEN WURDEN       10 ZEICHEN.
DAVON WAREN        0 BUCHSTABEN UND        0 ZIFFERN.
```

2. Die Library mit den externen TRS-Funktionen bzw. Prozeduren (TRSLIB/OBJ) enthält wohl Fehler, da Programme, die sie benützen, nach dem Linken nicht mehr einwandfrei funktionieren.

3. Der Overlay-Compiler steigt immer mit Fehlermeldungen aus, obwohl es sich um einwandfreie Programme handelt.

Falls jemand Patches kennt, um diese oder auch andere Fehler zu beheben oder aber diese Probleme mit dem ALCOR-PASCAL nicht hat, so soll er sich doch bitte bei mir melden.

Josef Konrad

Wer arbeitet mit
LISP ???
Bitte bei Josef Konrad
melden !!

Wer hat Interesse daran,
daß die Inhaltsverzeichnisse
der 80 MICRO
im INFO abgedruckt
werden sollen?
Ulrich Böckling

Zur Zeit sind Foto-Kopien
dieser Verzeichnisse
bei Josef Konrad
erhältlich.

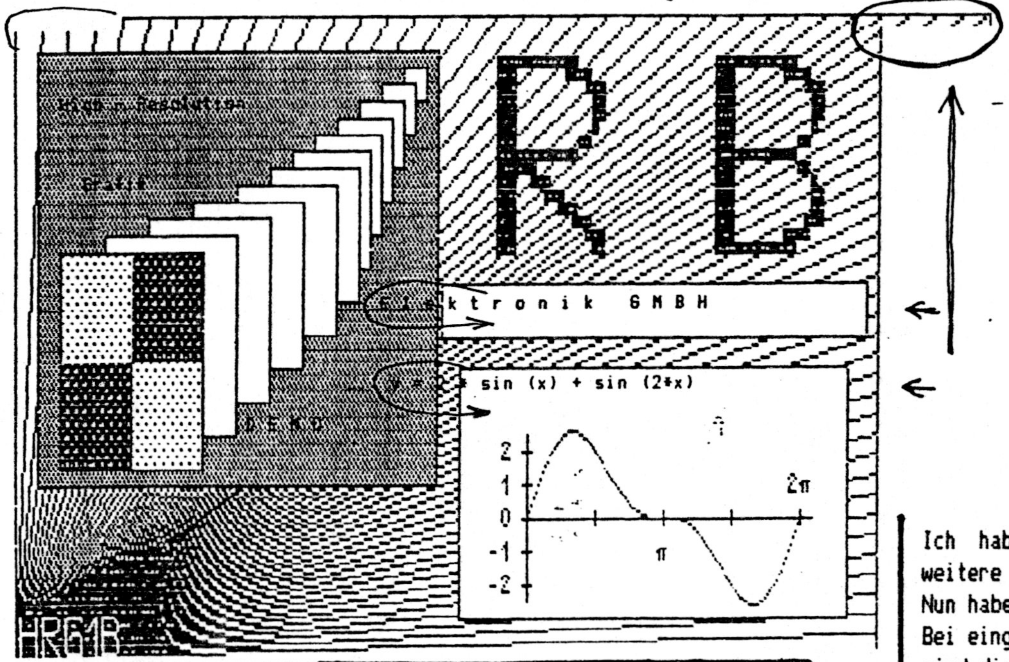
Wer kann helfen!
Ich bin nun glücklicher Besitzer der HRG1b von RB-Elektronik und möchte natürlich Graphiken erstellen.
Dazu habe ich folgendes Problem.
Nach dem Laden des Treiberprogr. von RB und dem nachladen des Basicprogr. "Graflix/BAS" aus der C.Persönlich Nr.21 v.3.10.84, lassen sich die Graphiken einwandfrei erstellen,nur ich kann die erstellten Graphiken nicht Speichern und Laden.Der Computer bleib hängen.

Wer kann mir auch noch in dem Progr.Graflix stehenden Drucker-zeilen von*ITOH auf EPSON RX umschreiben. * oder NEC??

```
LPRINTCHR$(27)"E"CHR$(27)"T16"CHR$(27);
LPRINTCHR$(27)"S0384";
```

Diese beiden Programme habe ich nun auch noch auf das hervorragende H-DOS von Arnulf Sopp kopiert.
Mit der JKL Funktion besteht nämlich die möglichkeit die HRG mit auszudrucken.
Beim Druck habe ich aber folgende Ergebnisse.(siehe Abbildung)
Wer kann hier helfen.

Hans-J. KÖNIG



Wie ist es möglich Sonderzeichen (speziell Umlaute) in Filenamen zu verwenden? Das NEWDOS lässt ja nur "reine Buchstaben" bzw. ab der 2. Stelle auch Ziffern zu.
Umlaute für Filenamen sind sicher interessant, da sich dadurch die Filenamen genauer angeben lassen und man durch weitere Zeichen mehr Codierungsmöglichkeiten hat.
Jens Neueder

??? Frage ???

Ich möchte mir eine CP/M - fähigen Computer zulegen. Trotzdem will ich nicht von der TRS 80 - Linie abweichen. Aus diesem Grund suche ich Leute die entweder:

- ihr TRS 80 Model 1 auf 64k RAM "aufgebohrt" haben (zum Erfahrungsaustausch) oder
- ein VideoGenie I oder II mit 64k RAM zu verkaufen haben.

Vor allem bin ich an einer Anleitung zur Erweiterung des TRS 80 Model 1 auf 64k interessiert. Schaltpläne für einen Mapper besitze ich bereits.

Hartmut Obermann

Wer hat Erfahrungen mit einem VOICE SYNTHI ???
Wie wird er eingebaut ???
Wer kennt Bezugsquellen ???
Lohnt sich der Kauf ???

Patrick Perschbach

Wer hat Erfahrungen oder Bauanleitungen über einen "Lichtgriffel" für TRS-80 Mod. I ?
Ist der Light-Pen auch für die HRG verwendbar ?
Jens Neueder

Ich habe seit längerem die Groß-/Kleinschreibung sowie weitere Tasten zur Erreichung der Sonderzeichen eingebaut.
Nun habe ich damit folgendes Problem :
Bei eingeschalteter Kleinschreibung (im NEWDOS oder Basic) sind die Kleinbuchstaben der Umlaute nur über SHIFT zu erreichen. Alle anderen Buchstaben werden aber ohne SHIFT klein dargestellt -wie es auch sein sollte-.
Bei Arbeiten mit Textverarbeitung oder anderen Betriebssystemen taucht der Fehler nicht auf.
Wer kennt die richtige Einstellung bzw. den Zap um Umlautkleinbuchstaben ohne SHIFT zu erzeugen ?
Jens Neuder

Bierlexicon S 0166 A 01 /BAS

Mit Hilfe dieses Programms kann man sich ausfuehrlich ueber die deutschen Biersorten informieren, ohne dabei Gefahr zu laufen, wegen Alkoholismus seine Stellung zu verlieren.

Fuenfzehnerspiel G 0167 A 03 /BAS

Wer kennt nicht das unterhaltsame Spiel, bei dem es gilt fuenfzehn in Unordnung gebrachte Zahlen in einem 4*4 grossen Quadrat durch verschieben der leeren Stelle wieder zu ordnen.

LISP S 0168 A 03 /BAS

Diese Implementierung der Sprache fuer die Programmierung von Problemen der kuenstlichen Intelligenz entstammt der Micro 80 Maerz 83 S.176

LOCEDIT S 0170 A 02 /BAS

Das Programm entstammt der Micro 80 und ermittelt die genaue Stelle, an der ein Syntaxfehler in einer laengeren BASIC-Zeile auftrat. Das Programm ist beim Editieren von Programmen sehr hilfreich!

MODEM via Cassettenport S 0171 B 05 /BAS

Dieses Programm (Micro 80) ermoeglicht mit Hilfe einer kleinen Zusatzschaltung den Betrieb eines Modems an der Kassettenschnittstelle des TRS80 M1. Man erspart sich also die Anschaffung einer RS232.

PILOT S 0172 A 03 /BAS

Quelle: Micro 80. Dieses Programmpaket (3 Progr.) implementiert nicht nur einen PILOT-Interpreter sondern stellt gleich einen kurzen Einfuehrungskurs fuer diese Sprache dar.

RX 80 - Einsteller S 0173 B 01 /CMD

Mit diesem Programm kann man sehr leicht die Parameter (Schriftart, Randeinstellung usw.) des EPSON RX 80 einstellen. Das Programm ist ein compiliertes BASIC-Programm.

Telefonwaehler S 0175 B 03 /BAS --

Das Programm ermoeglicht, nach einem geringfuegigen Eingriff in das Telefon, mit dem Computer Telefonnummern zu waehlen. Autor: J. Herl

---

User Definable Keys S 0176 B 05 /CMD --

Mit diesem Programm (Quelle: ELCOMP) kann man die meisten der Tasten der Tastatur mit DOS- und BASIC- Befehlen belegen.

Unnummer S 0177 B 02 /CMD --

Das Programm macht, vom BASIC aus geladen und mit "NAME" aufgerufen, alle nicht als Sprungadressen benoetigten Zeilennummern zu 0 und erschwert somit die Veraenderung von Programmen durch Unbefugte.

QUADRATO G 0178 A 01 /BAS --

Das Programm verwaltet das Spielbrett fuer das bekannte Quadratospiel. Es ermoeglicht nicht Quadrato gegen den Computer zu Spielen! Quelle: Homecomputer

---

SuperTape S 0180 B 05 /CMD --

Supertape ist ein Programm zur schnellen Speicherung von Programmen und Daten auf Tape (siehe Clubinfo Nr.7). Quelle: c't

---

Westwaerts 1847 B 0181 A 03 /BAS 16

Ein gutes Adventure-aehnliches Spiel mit Grafik.

CMDPATCH - BASIC Erweiterung S 0182 B 03 /CMD --

Dieses Programm erweitert den BASIC - Befehlssatz um die Befehle: CLICK, FLASH, MELT, RVS und PIEP. Quelle: ELCOMP 4/5 84

DEFINPUT S 0183 B 03 /CMD --

Dieses Programm erlaubt es, den BASIC-Befehl Input mit Parametern (max. Eingabelaenge usw.) zu versehen.

SPOOLER S 0184 B 03 /CMD --

Ein 4k RAM-Spooler der die Programmausfuehrung von Programmen mit viel Textausgabe erheblich beschleunigt. Da der Sourcecode vorhanden ist, kann der Spooler leicht veraendert (Groesse usw.) werden.

TOTO S 0185 A 03 /BAS --

Ein Programm, mit dem eigentlich jeder zum TOTO-Millionaer werden muesste! Autor: Guenther Wagner

---

INPUT S 0186 B 03 /CMD --

Das Programm erfuellt die gleichen Aufgaben wie das Programm DEFINPUT (S0183B03/CMD) jedoch mit anderem Syntax. Autor: Gerald Schroeder

---

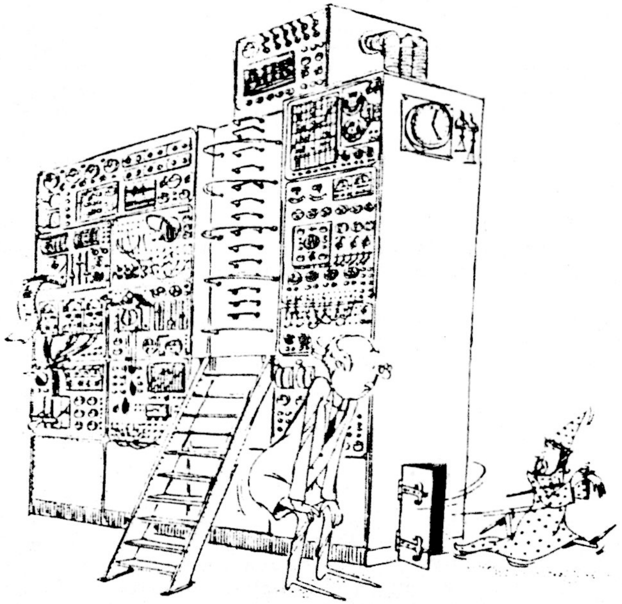
Kraftwerk D 0187 B 03 /BAS --

Aehnlich dem Adventure Powerplant jedoch eingedeutscht, erheblich erweitert und mit Hilfsfunktionen versehen von Gerald Schroeder

---

TRANSMUT S 0189 A 01 /BAS --

Dieses Programm erleichtert die Aenderung von BASIC-Programmen. Es ermoeglicht z.B. die globale Aenderung von Variablennamen und das Auffin-den von BASIC-Befehlsworten.



»Feierabend!«

CLUB 80 - Programmbibliothek exklusiv

Heute möchte ich euch wieder ein paar Programme aus der Bibliothek ganz besonders ans Herz legen.

Da wäre als erstes einmal die wirklich sehr gute übersetzung des Superutility+ 3.2 - Handbuches von Gerald Schröder zu nennen. Sie ermöglicht es auch dem, der englischen Sprache weniger mächtigen, Besitzer der Superutility diese voll auszunutzen. Die übersetzung liegt als Scripsit-File vor und kann so von jedem selbst ausgedruckt oder direkt am Bildschirm gelesen werden.
Das Handbuch hat übrigens ca. 70 Seiten!!!

Weiterhin wären zwei Programme zu nennen, die schon länger in der Bibliothek vorhanden sind. Es handelt sich dabei um EDITOR (S0007A01/BAS) und AktienTabelle (S0018A02/BAS), beide von unserem Clubgründer Günther Wagner. Diese Programme wurden vom Günther überarbeitet und sind nun in ihrer neuesten Version zu haben.

Zum Thema Aktienverwaltung kann ich noch berichten, daß unser Mitglied Ulrich Böckling ein sehr gutes Programm zur Lösung dieses Problems geschrieben hat. Es ist sehr umfangreich und mit vielen Hilfsfunktionen gespickt. Leider läuft es nur mit einer Highresolution Grafik-Erweiterung. Wer es haben möchte, sollte sich bitte beim Ulrich direkt melden. Er bekommt dann nämlich noch die Aktienkurse einer Unmenge von Firmen aus den letzten 1 1/2 Jahren mitgeliefert.

Zu guter Letzt noch einmal ein Hinweis zum inzwischen abgeschafften Punktesystem. Immer wieder schreiben mir Mitglieder, ich solle ihnen ihren aktuellen Punktestand mitteilen. Hier also noch einmal der Hinweis: Das Punktesystem ist abgeschafft!!! Jedes Mitglied kann so viele Programme bestellen wie es mag. Dies kann z.B. so weit gehen, daß man mir eine entsprechende Anzahl Disketten (die Bibliothek umfaßt z.Z. ca. 5 DS/DD/80 Track-Disketten) zuschickt und dann die komplette Bibliothek zurückbekommt.
Trotzdem sollte natürlich niemand vergessen, seine selbstgeschriebenen oder abgetippeten Programme in die Bibliothek einzubringen.

Das wars mal wieder. Ich hoffe, ich höre bald mal von Euch, Euer

Hartmut Obermann

CLUB 80

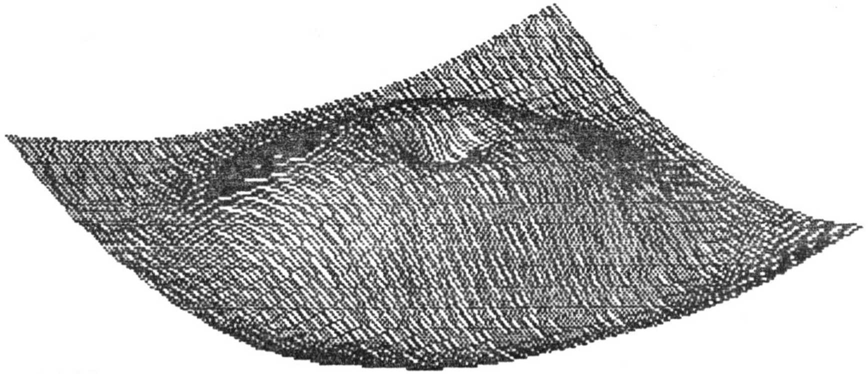
Clubinfo der TANDY - GENIE - und KOMTEK - ANWENDER

10. AUSGABE

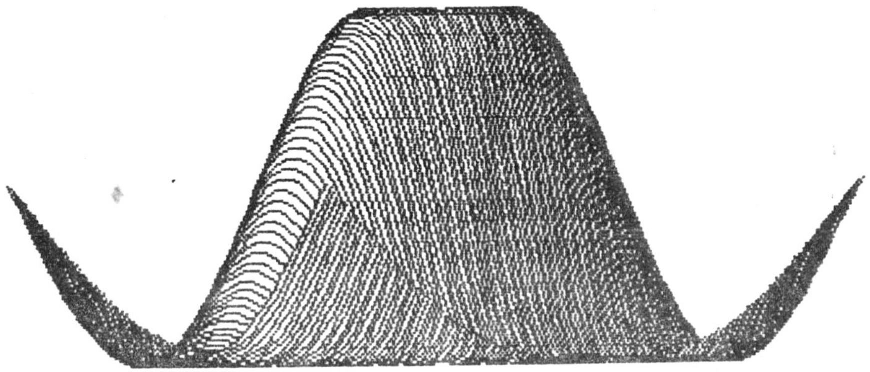
Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubeuern

Tel.: 08[illegible]/3351 (18 - 20 Uhr)

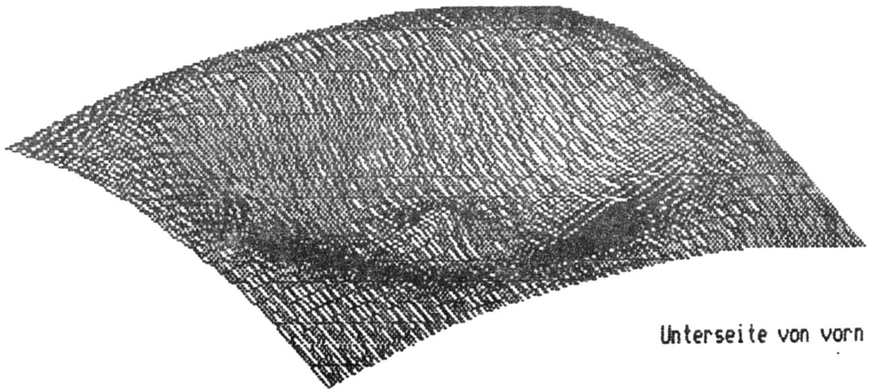
3D-HRG-Modell in verschiedenen Ansichten



Draufsicht von vorn



Seitenansicht von vorn



Unterseite von vorn

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert . . . . .01 - 02

Software

nochmals LMOFFSET . . . . .03 - 04
Extended NEWDOS 80 . . . . 05 - 10
Platz im SYS 1 . . . . 11 - 14
Auflistung NEWDOS 80 Befehle 15
Eröffnen von Systemfiles 16
PDrive-Identifikation 17
DDE und LMT . . . . 19 - 20
Ein Byte und seine Folgen . . . . . 21 - 22
HRG-Pixel's . . . . 23 - 32
Cassetten-Verzeichnis . . . . 33 - 34
nochmals EDITOR . . . . 35 - 36
Minni-RAM-Floppy . . . . 37 - 39
Vergleichsliste G-Dos/NewDos 40
Umlaute im NEWDOS . . . . 41 - 42
Filevergleich . . . . 43 - 44

Hardware

TRS 80 Umbau (16k auf 48k) . . . . . 45 - 48
nochmals Taktumschaltung 49

Seite:

Börse

Wer hat was ???--Wer sucht was ??? . 51 - 52

Sonstiges

Computer anders genutzt . . . . . . 53 - 54
Urheberrecht . . . . 55 - 56
Ein alter Bekannter ?? 57
Kurzes 58

Programmbibliotek

Exklusiv 59
Neue Programme . . . . 59 - 61

Club -Bücherei

Bücherliste . . . . 63 - 64
Buchtest 65

Die letzten Seiten

Impressum 66
Schluß . . . . 67 - 68
Adressenliste . . . . letzte Seite
Anmeldung . . . . am INFO-Ende

Liebe Clubmitglieder,

dieses ist jetzt das letzte Clubinfo, das Euch noch 1985 erreicht - Nr. 11 kommt ja schon Januar 1986. Keine Angst - der Rückblick auf das Jahr 85 kommt erst im nächsten Info.

In diesem Info darf ich auf andere Punkte hinweisen - wie heißt es so schön: "Ohne Moos nicht's los!"
Ja - es ist bald wieder soweit - der Jahresbeitrag für 1986 wird fällig. Um es Euch so einfach wie möglich zu machen, liegen vorgedruckte Zahlscheine bei. Bitte denkt daran, daß Ihr möglichst schon Mitte Dezember den Jahresbeitrag für 1986 in Höhe von 50 DM überweist (bzw. anderweitig bezahlt).

Da wir gerade beim Geld sind: folgende Beiträge sind noch immer ausständig (berücksichtigt sind Zahlungen bis einschließlich 30. Oktober):

Jeweils 10 DM Nachzahlung von: Troesch, May, Wies, Hummel, Baldes, Smerling, Fröhlich

Ich bitte um baldige Bezahlung der ausstehenden Beträge!

Clubtreffen 1986

Unser Clubtreffen 1986 nimmt schön langsam Gestalt an. Sowohl Termin als auch Unterkunft stehen fest.

Termin: Freitag 11.04. bzw. Samstag 12.04. bis Sonntag 13.04.

Das heißt folgendes: Wer will, kann schon am Freitag anreisen. Er hat dann die Möglichkeit, sich länger und intensiver mit anderen Mitgliedern zu unterhalten bzw. über die Computerei zu reden. Das eigentliche Treffen aber dauert so in etwa von Samstag Nachmittag bis Sonntag Mittag.

Unterkunft: Hotel Taunusblick, 5429 Holzhausen, Tel 06772/8343

Der Erholungsort Holzhausen liegt am Kreuzungspunkt der Bundesstraße 260 (Wiesbaden-Nassau, auch Bäderstraße genannt) mit der Bundesstraße 274 (St. Goarshausen-Limburg) und ist über Autobahnen von jeder Himmelsrichtung gut zu erreichen (Landkarte anbei). Das Hotel hat der Hartmut gefunden und für unser Clubtreffen für geeignet befunden. Neben schönem Gastraum und gemütlichen Nebenraum steht auch ein hoteleigenes Hallenbad (3*8m) zur Verfügung. Besonders letzteres dürfte für mitkommende Familienangehörige und sonstigem Anhang interessant sein - ich hoffe ja auf alle Fälle, daß einige Mitglieder Ihre Frau bzw. Freundin mitnehmen. Für diese nicht an der Computerei interessierten Anhängsel unserer Mitglieder besteht so z.B. die Möglichkeit zu einem Stadtbummel durch Wiesbaden, Koblenz (beides ca. 30 Autominuten), Limburg a.d.Lahn oder Dietz a.d.Lahn (ca. 20 Autominuten) und zu einem Spaziergang durch den schönen Taunus.

Die Übernachtung mit Frühstück kostet 28,-- DM. Das Essen ist gut und preiswert. Der Hartmut glaubt, daß das Hotel mit seiner Atmosphäre und Umgebung zu einem gelungen Clubtreffen beitragen wird.

Ihr findet als Beilage das Anmeldeformular für unser Clubtreffen. Bitte füllt a l l e dieses Formular aus; auch diejenigen, die am Clubtreffen nicht teilnehmen. Wir brauchen unbedingt einen Überblick über die Teilnehmerzahl. Bitte sendet mir alle das Formular nach Möglichkeit bis Jahresende zu (je früher, desto besser).

Denkt daran, daß die Anreise auch in Fahrgemeinschaften möglich ist. Damit werden die Fahrtkosten billiger und auch Schülern ist die Anreise erleichtert. Dies müßt Ihr aber selber unter Euch organisieren. Bitte macht Euch auch Gedanken über das Treffen; wie sollte es ablaufen, was sollten wir im Gegensatz zum ersten Clubtreffen ändern, über was wollt Ihr sprechen etc.

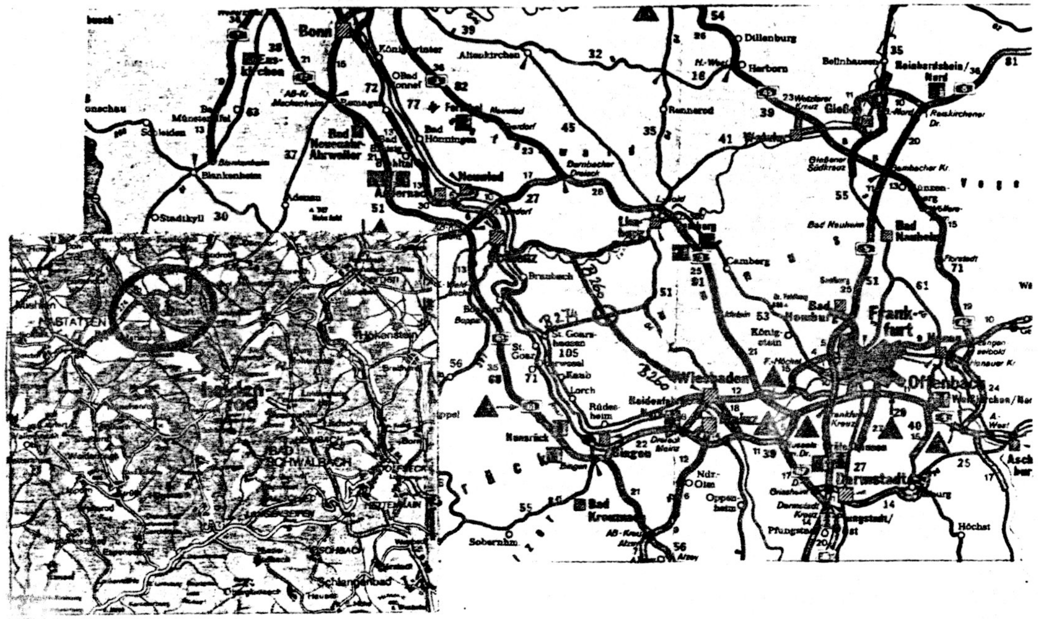
Nicht vergessen:

- alle den Jahrebeitrag 1986 zahlen !
- alle das Anmeldeformular zurücksenden !

D A N K E

Ihr hört wieder im Januar von mir (kommt gut rüber in's neue Jahr)

bis dann
Euer Günther Wagner



Zu Geralds Artikel über LMOFFSET

Lieber Gerald, in Deinem interessanten Beitrag im Info 9/85 forderst Du Hartmut und mich auf, mehr zur Record-Organisation zu sagen. Das will ich später gerne tun. Zunächst möchte ich aber mit der Paral-Spritze einer Wanze zuleibe rücken, die sich dort eingeschlichen hat:

42E9 ist nicht etwa eine feste Adresse im DOS, sondern der erste frei verfügbare Speicherplatz unter Level 2. Hier beginnen z. B. BASIC-Programme oder Variable, wenn noch kein Programm im Speicher steht. Auch Maschinenprogramme stören ab hier nicht mehr, wenn BASIC nicht gebraucht wird. Und hier hast Du die Autoren von LMOFFSET bei einem Irrtum erwischt: Bereits ab 42E8 ist der Speicher frei. Dort steht eine 00, die normalerweise den Anfang einer neuen (hier also der allerersten) BASIC-Zeile markiert.

Hinter der letzten Programmzeile folgt wieder eine 00. Würden weitere Zeilen folgen, so enthielten die beiden nächsten Bytes den Zeiger auf die übernächste Zeile. Wenn dort aber zwei weitere Nullen stehen, weiß der Interpreter, daß das Programm zuende ist und ab hier die Variablen abgelegt werden können. Steht aber kein Programm im Speicher, dann folgen diese beiden Nullen gleich der ersten, so daß wir ab 42E8 die Bytefolge 00-00-00 vorfinden. Genau das bedeuten die drei NOPs, die Du als Tabelle interpretierst. Sie signalisieren dem Interpreter lediglich, "sorry, kein BASIC", so daß er z. B. nach RUN oder LIST unverrichteter Dinge sofort wieder zu "READY" zurückkehrt.

So, das wurde jetzt ein bißchen ausführlicher, denn ich meine, daß der BASIC-User wissen sollte, was intern passiert, wenn er programmiert. Aber zu Deinem Record-Thema:

Jeder DOS-Record beginnt mit einem Header-Byte, das dem DOS sagt, welcher Art der folgende Code ist. Wie Du richtig feststellst, bedeuten 01 den Start von Maschinencode und 02 die Ankündigung der Startadresse am Programmende. Da wäre noch 05 zu erwähnen, das dieselbe Funktion hat wie REM in BASIC: Hier folgt ein Kommentar, den das DOS nicht weiter beachtet. Nach diesem Header kommt ein Byte, das die Länge des folgenden Records angibt. Zwischen 03 und FF bedeutet dieses Byte eine Länge von 3 - 255 Bytes. 00 meint 256. Viele SYS-Files haben eine Recordlänge von 256 Bytes. Was es mit den ebenfalls möglichen Zählbytes 01 und 02 auf sich hat, folgt gleich.

Bis hierher wird kaum ein Leser "aha" gesagt haben, denn das ist das Übliche. Jetzt kommt's aber: Es sind auch die Recordlängen 257 und 258 möglich. NEWDOS und seine Enkel G-DOS und H-DOS sind in der Lage, diese Recordlänge aus den Zählcodes 01 und 02 herauszulesen. Weshalb das zumindest für 02 höchst erstaunlich ist, wird nach der folgenden Erläuterung klar:

Nach dem Header (01, 02, 05 ...) und dem Zählbyte folgen zwei weitere Bytes, die noch nicht zum eigentlichen Programmcode gehören. Sie bedeuten in der Reihenfolge LSB-MSB die Ladeadresse des Records. Diese beiden Bytes werden im Zählbyte mitgezählt. So erklärt es sich, daß das Programmende mit 02-02-yy-xx angekündigt wird: 02 (finito) - 02 (zwei Bytes folgen) - yy (LSB der Startadresse) - xx (MSB). Der Zählcode 02 bedeutet demzufolge 2 Bytes und nicht 258. Oder eben doch. Also beides: Wie das DOS das auseinanderhält, weiß ich auch nicht. Es tut es. finden wir uns damit ab, und freuen wir uns darüber!

Mit ASM/CMD (und wohl auch mit seinem Großvater EDTASM/CMD) entstehen immer Records von maximal 252 Bytes Länge. - 252 plus die beiden mitgezählten Adreßbytes ergeben 254 (FE), plus Header plus Zählbyte sind das genau 256 Bytes, also ein kompletter Sektor. So fängt jeder S[illegible]tor in einem zusammenhängenden Maschinensprachefile links oben mit [illegible]-FE an, wonach die Ladeadresse folgt. Das ist optisch sehr angenehm und macht einen "aufgeräumten" Eindruck.

ZEUS/CMD erzeugt Records von 256 Bytes Länge. Sie beginnen mit 01-02 (01 für Maschinencode, 02 für 258 Bytes, also 2 Adreß- plus 256 Programmbytes). Das wirkt "unordentlicher", hat aber seinerseits wieder einen anderen Vorteil: In einem Sektor lassen sich Speicheradressen sehr leicht auszählen, denn wenn z. B. ein Record bei 8000 beginnt, so startet der nächste mit 8100. Bei EDTASM wäre es stattdessen 80FC.

Ein weiterer Vorteil ist noch wesentlich wichtiger: Wenn in einem Sektor ganz unten rechts der Header, das Zählbyte und die Ladeadresse des nächsten Records stehen, so ist der komplette nächste Sektor mit reinem Maschinencode beschrieben, es erscheinen keine Bytes, die zur Recordorganisation gehören. Dann haben wir es mit einem sog. "core image code" zu tun, also Maschinensprache "netto", wie z. B. in BOOT/SYS. Wenn ein solcher Sektor diskrelativ oder dateirelativ eingelesen wird, kann er sofort angesprungen werden. Seine Ladeadresse bestimmt der User. Mit diesem Trick arbeitet z. B. das alte H-DOS, um die Funktionentabelle bei Bedarf reinzuschlürfen. Auch die beiden FORM-Tabellen in SYS28/SYS von G- (H-) DOS sind als core image code angelegt.

Einen solchen Sektor zu erzeugen, ist kein Problem. Man schreibt z. B. mit DEFM ein Stück Dummy-Code von genau 248 Bytes Länge. Seine Ladeadresse liegt irgendwo, z. B. im ROM. Wenn die Adresse des nächsten, ernst gemeinten Records nicht unmittelbar anschließt, wird der alte Record nach dem 248. Byte geschlossen. Die nächsten Bytes zur Definition des nächsten Records werden dann direkt dahinter geschrieben. Zwei Header, zwei Zählbytes und zwei Adressen addieren sich zu 8 Bytes. 8 + 248 = 256 Bytes, also ein Sektor. So beginnt der reine Maschinencode des nächsten Records tatsächlich ganz links oben im nächsten Sektor und endet ganz rechts unten.

Diese Erklärung ist nun ebenfalls wesentlich ausführlicher geworden, als es Deine Anregung erfordert hätte. Es sollen aber alle etwas von dem profitieren, was ich im Laufe der Zeit (und der Entwicklung von H-DOS) herausgefunden habe.

Du möchtest die BREAK-RESET-Akrobatik umgehen und schlägst dazu vor, Teile des ROM-Inhalts ins eigene Programm zu übertragen, um BASIC zu initialisieren. Hierzu genügen die paar Bytes von 0676-0680 und von 0685-068A. Anschließend hüpft man einfach nach 0075, den Rest macht der Computer alleine. Dann steht gleich READY auf dem Bildschirm. Mit dem SYSTEM-Befehl (oder USR) kommt man in sein eigenes Programm. Es geht noch komfortabler, aber da wird es allmählich kompliziert:

Die Routine ab 0075 landet bald bei 1A1C. Dort steht CALL 41AC. Wenn nun in 41AC der Befehl JP PROGRM steht, wird das eigene Programm direkt angesprungen. Den alten Code an 41AC (RET) kann man anschließend restaurieren, um Level 2 komplett im alten Zustand zu haben. Auf diese Weise werden letzte Initialisierungen übersprungen, die aber ziemlich unwichtig sind: Der Cursor kommt nicht an den nächsten Zeilenanfang und der Cassettenmotor wird nicht ausgeschaltet. Da der Recorder aber nicht lief, ist das egal. Daß READY nicht ausgegeben wird, ist ebenso wurscht, weil das eigene Maschinenprogramm vielleicht andere Sorgen hat, als ausgerechnet auf eine BASIC-Eingabe zu warten.

Wenn das Maschinenprogramm das Level-2-BASIC nicht benötigt (keine RSTs, keine CALLs ins ROM usw.), kann man auf jegliche Initialisierung verzichten. Man darf dann sogar nach 4000 laden, der untersten RAM-Speicherstelle.

Arnulf Sopp

Extended NewDOS 80

Erweitern des NewDOS80 mit GDOS-, HDOS- und selbstgestrickten Funktionen

Wenn man sich einmal die Mühe macht zu ermitteln, welches Betriebssystem von TRS 80- und VideoGenie- Benutzern bevorzugt wird, kommt man zu folgendem Ergebnis:

NEWDOS 80
GDOS (HDOS)
TRSDOS
DOSPLUS
MULTIDOS
sonst. (VTOS,LDOS usw)

Wie man feststellen kann, ist NEWDOS das meistverwendete Diskettenbetriebssystem. Platz zwei belegt GDOS, die restlichen Systeme rangieren nur unter "ferner liefen"!

Den Grund, warum NEWDOS beliebter ist als GDOS, sehe ich keineswegs in der Leistungsfähigkeit (GDOS hat mehr Befehle als NEWDOS), sondern in der etwas ungeschickten, eingedeutschten Syntax der GDOS-Befehle.

Wenn man jedoch den sehr komfortablen Befehlssatz von GDOS (noch besser HDOS, z.B. OUT, ID usw.) benutzen will, muß man eben in den sauren Apfel beißen!
Besser gesagt, man musste es!!!

In der letzten Woche meines Urlaubs habe ich mich endlich an ein Projekt gemacht, das mir schon lange vorschwebte. Ich wollte das NEWDOS erweitern. Anfänglich dachte ich nur an die HDOS-Befehle ID und OUT. Am Ende wurden daraus 10 neue Befehle!

Inzwischen ist das Projekt praktisch abgeschlossen und erprobt. Ich finde es deshalb an der Zeit, euch meine Vorgehensweise darzulegen, damit sie jedermann nachvollziehen und sich seine eigenen DOS-Erweiterungen basteln kann (oder meine übernehmen).

Die Grundlagen, die ich zur Erweiterung von NewDOS benötigte und die im Rahmen dieser Artikelserie noch genauer erklärt werden, habe ich einer Reihe von Artikeln unseres Clubmitglieds und Oberzappers Arnulf Sopp entnommen (erschienen im Info des Computerclub Bremerhaven). Ich möchte ihm daher an dieser Stelle besonders für seine vielen Anregungen danken!
Die Anregungen und Informationen wurden durch die Lektüre des neuen Röckrath-Buches "Das DOS Buch" von H. Grosser, erheblich vertieft. Dieses Buch enthält sehr viele tiefgründige und genaue Informationen über das Betriebssystem NewDOS80 und seine Abkömmlinge. Ich kann das Buch nur wärmstens empfehlen!

Nun genug der langen Vorrede, ans Werk!!!

Grundlagen:

Der Aufruf eines NewDOS-Befehls:

Die Befehle, die unser NewDOS (und auch GDOS und HDOS) versteht, sind im sog. Library verzeichnet. Das Library von NEWDOS, GDOS und HDOS ist im Systemfile 1 (SYS1/SYS) untergebracht.
Wird ein Befehl aufgerufen, durchsucht das System das Library.

APPEND	ATTRIB	AUTO	BASIC2	BLINK	BOOT	BREAK	CHAIN
CHNON	CLEAR	CLOCK	CLS	COPY	CREATE	DATE	DEBUG
DIR	DO	DUMP	ERROR	FORMAT	FREE	HIMEM	JKL
KILL	LC	LCDVR	LIB	LIST	LOAD	MDBORT	MDCOPY
MDRET	PAUSE	PDRIVE	PRINT	PROT	PURGE	R	RENAME
ROUTE	STMT	SYSTEM	TIME	VERIFY	WRDIRP		

Hat es den eingegebenen Befehl gefunden, weist es dem A- und dem C-Register der Z80-CPU bestimmte, für den Befehl spezifische Werte, zu.

Beispiele:

	A-Reg.	C-Reg.
DIR	2A	80
ID	F8	80
ATTRIB	E9	85
AUTO	E9	84

Danach wird mit dem Z80-Befehl RST 28 ein Teil des DOS angesprungen, das das Systemfile lädt und startet, welches die Routine für den entsprechenden Befehl enthält. Welches Systemfile dies ist, wird durch den Wert bestimmt, der zuvor im A-Register (ACCU) der CPU abgelegt wurde.

Der im C-Register gespeicherte Wert dient nach dem Einsprung in das entsprechende Sysfile zum Durchsteigen einer Entscheidungstabelle, die zur richtigen (fast alle Systemfiles enthalten mehrere) Unterroutine führt.

Verteilung der Systemfiles im Directory und Zusammensetzung des "Aufrufbytes":
Das Directory von NewDOS-, GDOS- und HDOS-Disketten ist eine relativ flexible Sache. Es kann, bestimmt durch die PDrive-Parameter, praktisch auf fast allen Spuren einer Diskette liegen (außer auf der Spur 0) und die unterschiedlichsten Größen annehmen. Etwas am Directory ist jedoch starr und immer gleich, die Verteilung der Systemfiles!

Das Directory kann, wie schon gesagt, unterschiedlich groß sein, seine Mindestlänge beträgt jedoch zehn Sektoren (entspricht bei SS/SD-Disketten zwei Granules). Die ersten beiden Sektoren eines Directorys werden immer vom Granule Allocation Table und vom Hashcode Index Table belegt. Aus diesem Grund sind die max. 32 Systemfiles auf die acht verbleibenden, mindestens verfügbaren Sektoren verteilt!
Die eigentliche Aufteilung ist aus dem nachfolgenden Schaubild ersichtlich:

Aufbau des Direktory und Verteilung der Systemfiles

DIR/SYS relative Sektoren:

relative Bytes im Sektor

	0 Granule Allocation Table	1 Hashcode Indication Table	2	3	4	5	6	7	8	9	10	11usw.
00			BOOT	DIR	SYS0	SYS1	SYS2	SYS3	SYS4	SYS5	weitere	File-
10												
20			SYS6	SYS7	SYS8	SYS9	SYS10	SYS11	SYS12	SYS13	Einträge	
30												
40			SYS14	SYS15	SYS16	SYS17	SYS18	SYS19	SYS20	SYS21		
50												
60			SYS22	SYS23	SYS24	SYS25	SYS26	SYS27	SYS28	SYS29		
70												
80			weitere	File-	Einträge							
90												
A0												
B0												
C0												
D0												
E0												
F0												

»Der Spezialist ist in seinem winzigen Weltwinkel vortrefflich zu Hause; aber er hat keine Ahnung von dem Rest.«

»Ärgern Sie sich nicht, wenn Ihr Haar ausfällt. Stellen Sie sich vor, es täte weh und müßte gezogen werden wie Zähne.«

»Der Unwissende hat Mut, der Wissende hat Angst.«

Weibliche Logik

Im Stadtpark trägt der Oberprimaner seiner Angebeteten begeistert Goethes Ballade vom Zauberlehrling vor. Sie, nachdem sie ihm schweigend gelauscht hat, meint schließlich skeptisch, daß es im Zeitalter der Elektronengehirne wohl auch so sei und man sich der gerufenen Geister nicht erwehren könne.

»Aber das ist doch unmöglich«, ereifert er sich, »ein Computer kann sich nicht selbständig machen! Wir geben ihm Informationen, und er zieht daraus logische Schlüsse. Nur geht es sehr viel schneller und genauer als es das menschliche Gehirn vermag.«

Da sie ihn zweifelnd ansieht, versucht er zu erklären: »Von Hannover bis Hamburg sind es 160 Kilometer. Wie weit ist es also von Hamburg nach Hannover?« »Ebenfalls 160 Kilometer«, antwortet sie.

»Gut«, lobt er, »das ist eine logische Folgerung. Der –«

»Moment«, unterbricht sie ihn. »Ich möchte dich auch etwas fragen: Wie viele Tage sind es von Weihnachten bis Neujahr?«

»Natürlich 7 Tage«, entgegnete er irritiert, »aber das hat doch damit nichts zu tun!«

»Und ob!« trumpft sie auf, »denn von Neujahr bis Weihnachten sind es genau 358 Tage! Nun sage mir, wo deine Logik bleibt!«

Da die Systemfiles relativ häufig benötigt werden, werden sie nicht über ihren Namen aufgerufen (wie z.B. jedes Maschinenprogramm), sondern von einem speziellen Teil des DOS unter Zuhilfenahme eines "Aufrufbytes" direkt geladen. Das verkürzt die Ladezeit erheblich und bringt somit eine Beschleunigung bei der Ausführung von DOS-Kommandos.

Das Aufrufbyte setzt sich folgendermaßen zusammen:

Die Bits 0 bis 2 stellen den relativen Sektor minus 2 dar, in dem der Fileeintrag des gewünschten Systemfiles steht.
z.B.: SYS8/SYS steht im relativen Sector 4 des Direktory
4 - 2 = 2 bzw. als Bitmuster 0 1 0
Bits im Aufrufbyte: 2 1 0

Die Bits 3 und 4 stellen den relativen Fileeintrag in dem Sektor dar, in dem das Systemfile steht.
z.B.: SYS8/SYS steht an zweiter Stelle im rel. Sektor 4, d.h. es stellt den ersten relativen Fileeintrag in diesem Direktoysektor dar. Bitmuster 0 1
Bits im Aufrufbyte: 4 3

Die verbleibenden Bits 5 bis 7 können zur Übergabe von Parametern benutzt werden. Sie sollten auf alle Fälle größer 0 sein (am besten Bit 7 gesetzt).
z.B.: 1 0 0
Bits im Aufrufbyte: 7 6 5

Daraus ergibt sich folgendes Bitmuster zum Aufruf des Systemfiles 8: Bit 7 6 5 4 3 2 1 0
1 0 0 0 1 0 1 0
und dies entspricht der Hexzahl 8A

Das sieht im ersten Moment sehr kompliziert aus, ist es aber nicht! Wenn man die Prozedur ein paarmal durchgerechnet hat, geht sie einem relativ schnell von der Hand.
Zur Übung hier noch ein Beispiel:
Aufruf des SYS21/SYS
SYS21/SYS steht im rel. Sektor 9 des Direktory und ist dort als drittes File eingetragen. Für die Bits 5 bis 7 wollen wir das Bitmuster 1 0 1 einsetzen. Daraus ergibt sich folgende Bitfolge: 1 0 1 1 0 1 1 1
Bits im Aufrufbyte: 7 6 5 4 3 2 1 0
und dies entpricht der Hexzahl B7

Ich glaube, damit wäre die Sache wohl klar und wir können uns dem im C-Register gespeicherten "Sprungbyte" widmen.

Das "Sprungbyte":
Die Systemfiles haben im Schnitt eine Größe von fünf Sektoren. Das entspricht ca. einem kByte Programm. Darin läßt sich natürlich mehr als nur eine kurze Routine, z.B. für das Anzeigen der Library oder des Direktorys, unterbringen. Ein Systemfile kann jedoch, genau wie jedes Maschinenprogramm auch, nur eine Startadresse haben. Man benötigt deshalb ein Argument, mit dessen Hilfe man in die gewünschte Unterroutine gelangt.
Beim Aufruf der Systemfiles wird dieses Argument im C-Register der CPU zwischengespeichert und am Beginn des Systemprogramms ausgewertet. Eine solche Auswertung zeigt das folgende disassemblierte Listing von SYS28/SYS:

Disassembliertes Listing des SYS28/SYS in der von mir geänderten Form. Das Systemfile enthält die Befehle F(orm), DR(uck), LF(eed) und OUT!

Die Befehle erhalten im SYS1/SYS folgende Werte zugewiesen:

Befehl:	Accu	C-Register
LF	FE	81
DR	FE	82
OUT	FE	87
F	FE	88

```
        ORG     4D00H       Programmbeginn und gleich-
                            zeitig Startadresse!
START   CP      0FEH        "Aufrufbyte" richtig ?
        JR      NZ,ERROR    wenn nein, dann Fehler!
        DEC     C           Vom Inhalt des C-Registers
                            1 abziehen!
        JP      Z,LFeed     C-Register jetzt = 0 ?
                            Ja, dann zur Routine,
                            Nein, dann weiter!
        DEC     C           Das gleiche Spielchen wie
        JP      Z,DR        oben!
        DEC     C
        JP      Z,ERROR
        DEC     C
        JP      Z,ERROR
        DEC     C
        JP      Z,ERROR
        DEC     C
        JP      Z,ERROR
        DEC     C
        JP      Z,OUT
        DEC     C
```
Hier folgt die Bearbeitungsroutine für den Befehl F(orm)!

Dem aufmerksamen Leser (und das seid Ihr ja hoffentlich alle?!) wird sofort auffallen, daß dem C-Register jeweils ein Wert von 8x zugewiesen wird (Bitmuster: 10xx xxxx), während in der Sprungtabelle im Systemprogramm davon ausgegangen wird, daß die ersten vier Bit des Wertes auf 0 stehen! Wie man sonst könnte nach einmaligem Dekrementieren des C-Registers (DEC C) die LF-Routine erreichen, wenn zuvor 81 in "C" geladen wurde?!?
Die Lösung dieses scheinbaren Widerspruchs ist relativ einfach. Das System benutzt die Bits 5 und 6 des Sprungbytes selbst und setzt sie vor dem Aufruf des Systemprogramms auf 0.
Die Bedeutung der Bits:

Bit	Inhalt	Bedeutung
7	1	ist immer gesetzt und bildet die Endmarkierung für den davorstehenden Befehlstext
6	1	der Befehl darf nicht vom MiniDOS aufger. werden
6	0	der Befehl darf vom MiniDOS aufgerufen werden
5-0	x	Entry-Zähler

Die erste Aktion nach dem Start des Systemprogramms ist die Abfrage des Inhalts des Accu. Stimmt dieser nicht, so wird angenommen, das DOS hat sich "verlaufen" und es wird ein "Illegal DOS-Funktion"-Error angezeigt.
Die Stufen der Sprungtabelle, die zur ERROR-Routine führen, können übrigens sehr leicht zum Einbau bzw. Aufruf eigener Routinen benutzt werden (wenn noch Platz ist im Systemfile).

Nun aber weg von der grauen Theorie und hin zur Praxis!

Platz im SYS 1

Um zusätzliche Lib-Befehle installieren zu können, muß man sich zuerst Platz im Sys 1 schaffen. Das könnte man prinzipiell dadurch erreichen, daß man die bestehenden Befehlswörter kürzt. So könnte man z.B. aus BASIC2 B2 machen, aus ERROR E usw.. Meiner Meinung nach ist das jedoch ein ungeschickter Weg (wie er z.B. auch bei GDOS beschritten wurde) und ich war mir sicher, daß es auch einen anderen geben mußte.

Wer sich SYS1/SYS einmal mit Superzap oder DDE anschaut, stellt fest, daß am Ende des Files (Sektor 4 relativ) 32 Bytes mit 00 gefüllt sind. Ein bißchen Platz ist also schon da, jedoch an einer total falschen Stelle.

Das Libary endet, wenn man sich das Systemfile einmal disassembliert stellt man dies schnell fest, bei der Speicherstelle 50B6H. Als Endzeichen dient die Byte 00. Gleich anschließend folgt dann ein Teil des Programms zur Anzeige des Libaries (LIB).

```
5094 53595354454D           DEFM     'SYSTEM'   'LIBBefehl SYSTEM
509A 81                     DEFB     81H        'Wert für C-Register
509B E9                     DEFB     0E9H       'Wert für A-Register
509C 00                     DEFB     00H        'Schlußbyte
509D 54494D45               DEFM     'TIME'     'LIBBefehl TIME
50A1 8A                     DEFB     8AH        'weiter wie oben!
50A2 E9                     DEFB     0E9H
50A3 00                     DEFB     00H
50A4 564552494659           DEFM     'VERIFY'
50AA 84                     DEFB     84H
50AB E5                     DEFB     0E5H
50AC 00                     DEFB     00H
50AD 575244495250           DEFM     'WRDIRP'
50B3 80                     DEFB     80H
50B4 53                     DEFB     53H
50B5 00                     DEFB     00H        'Schlußbyte WRDIRP-Befehl
50B6 00                     DEFB     00H        'Schlußbyte Library
'Hier beginnt die Routine zur Anzeige des Librarys (LIB-Befehl)'
'man beachte: für die nächsten 36 Bytes nur relative Sprünge!!!'
50B7 21584F       M50B7     LD       HL,M4F58   'Anfang des Library in HL
50BA 0E08         M50BA     LD       C,08H
50BC 0608         M50BC     LD       B,08H
50BE 7E           M50BE     LD       A,(HL)
50BF CB7F                   BIT      7,A
50C1 23                     INC      HL
50C2 2005                   JR       NZ,M50C9
50C4 CD9F51                 CALL     M519F
50C7 10F5                   DJNZ     M50BE
50C9 23           M50C9     INC      HL
50CA 23                     INC      HL
50CB 7E                     LD       A,(HL)
50CC B7                     OR       A
50CD CA9D51                 JP       Z,M519D
50D0 0D                     DEC      C
50D1 CC9D51                 CALL     Z,M519D
50D4 28E4                   JR       Z,M50BA
50D6 CD9551                 CALL     M5195
50D9 18E1                   JR       M50BC
'hier ist es mit den relativen Sprüngen zuende!!!'
```


Schaut man sich die dort aufgeführten Programmschritte einmal genauer an, erkennt man, daß innerhalb eines Bereichs von 36 Bytes nur relative Sprünge auftreten. Dieser Umstand ließ in mir die Idee reifen, dieses verschiebbare Programmteil auf den freien Platz am Ende von SYS1/SYS zu verlegen.
Der Platz dort ist jedoch nur 32 Bytes lang. Aus diesem Grund mußte die DOS-Readymeldung etwas gekürzt werden. Als Ergebnis der Aktion hat man 36 Bytes freien Raum im Libary, in dem man eine ganze Menge neue Befehle einfügen kann (bei günstiger Ausnutzung max. 9 Befehle). Ich glaube aber, dafür lohnt sich die Arbeit!

Die Änderungen

Zunächst muß man die sehr lange Meldung "Mini-NewDOS/80 Ready" erheblich kürzen. Sie lautet dann "M-DOS >". Während sich der Beginn für die MiniDOS-Meldung nicht verschiebt, wird der Anfang der normalen DOS-Meldung auf die Speicherstelle 51B2 vorverlegt. Man muß dies der Routine, die diese Meldung benutzt, mitteilen. Dazu werden im Sektor 0 rel. des SYS1/SYS das Byte E1 rel. von B5 auf B2 ändern.

Nach der normalen Ready-Meldung des DOS folgen noch drei Bytes, die zur CLS-Routine gehören. Diese werden wieder direkt an die neue Meldung angehängt. Im Zusammenhang damit muß im Sektor 0 rel. das Byte 4D rel. von C5 auf B8 geändert werden!

Direkt im Anschluß daran kommen dann die 36 Bytes der LIB-Routine aus dem oben angegebenen Listing. Natürlich muß auch hier die aufrufende Routine über die Veränderung informiert werden. Dazu muß man im Sektor 0 rel. die Bytes 25/26 rel. von B750 in BB51 ändern.

Ob ihr's glaubt oder nicht, das war schon alles! Mehr Änderungen sind nicht zu machen, um sich einen ziemlich großen Platz im Library zu verschaffen, in den man dann einiges einbauen kann. Dazu aber später. Hier erst nochmal ein bißchen von der unabdingbar nötigen Theorie.

Das Schlußbyte

Wenn ihr euch nochmal den ersten Teil des Disassemblerlistings anschaut, werdet ihr feststellen, daß nach dem Befehlswort und den beiden Werten für das C- und das A-Register noch ein Byte steht. Dieses Byte, ich bezeichne es als Schlußbyte, enthält folgende, für die Befehlsausführung wichtige Informationen in seinem Bitmuster versteckt:

Bit	Inhalt	Bedeutung
0	1	besitzt der anzugebende Dateiname kein Extention, so ist /CMD als Default anzunehmen
1	1	wie oben, jedoch /JCL (bzw. /JOB bei GDOS)
2	x	nicht benutzt
3	0	das Registerpaar HL zeit auf die zur Ausführung benötigten Parameter
3	1	der Zeiger auf die benötigten Parameter ist auf dem Stack abgelegt
4	1	dem Befehl folgen weitere Parameter
4	0	dem Befehl dürfen keine weiteren Param. folgen
5	1	der Befehl erwartet zwei Dateinamen als Param.
6	1	bei der Ausführung des Befehls ist eine neue Datei zu eröffnen
6	0	bei der Ausführung des Befehls darf nur eine schon bestehende Datei geöffnet werden
7	1	SYS1 [illegible]net eine neue oder bestehende Datei zur Bearbe[illegible]tung durch das aufgerufene Systemprogramm

Quelle: Das DOS [illegible], Grosser

```
000000: 0100 004D FE23 CA8A 4DFE 4328 73FE 63CA ...M.#.█M.C(s.c.
000010: 304E FE83 CA3D 51FE A3CA 2A4F FEC3 283F ON.█.=Q.█.*O..(?
000020: 0D28 3A0D CAB7 500D CA32 4D0D CADB 500D .(:..█P..2M...P.
000030: 2852 0DCA FA50 0D28 430D CA34 4E0D 280C (R...P.(C..4N.(.
000040: 0D0D CA50 510D 283C 3E2A B7C9 21C5 51C3 ...PQ.(<>*█.!.Q.
000050: 6744 D901 01E3 11D3 49C5 D5D9 EFF1 C9CD gD......I.......
000060: 2B51 0100 00EB 216A 43CB 76CB F628 04ED +Q....!jC.v..(..
000070: 4B9D 43C5 ED73 9D43 EBC3 354E F1F1 180F K█C..s█C..5N....
000080: AF37 180B 216A 437E E62F 772B CBAE AFF3 █7..!jCB./w+.█.
000090: 216B 4336 002B 462B 4E1E 0BF5 CB50 2023 !kC6.+F+N....P #
0000A0: F1F5 3802 280A FE38 2806 1E04 CB69 2013 ..8.(..8(....i .
0000B0: CBB6 CB70 205E 3A6C 43CB 7728 0BCB 6920 .█.p ^:lC.w(..i
0000C0: 071E 0C16 EB7A 4BEF CB78 2014 31E0 41CB .....zK..x .1.A.
0000D0: 6F21 B045 2213 433E C328 023E C932 1243 o!_E".C>.(.>.2.C
0000E0: 21B5 51CB 7828 07ED 7B9B 4321 B051 FB3E !█Q.x(..ä█C!_Q.>
0000F0: 0DCD 3300 CB69 CC67 4421 6A43 CBEE 0108 ..3..i.gD!jC....

000300: 4154 C02B 0046 0100 FA4F 5245 4580 4A00 AT.(.F...OREE J.
000310: 4849 4D45 4D82 E900 4A4B 4C80 A510 4B49 HIMEM█..JKL █.KI
000320: 4C4C 8045 904C 4389 E500 4C43 4456 5288 LL E LC█..LCDVR█
000330: E500 4C49 4282 E300 4C49 5354 85F0 884C ..LIB█..LIST█.█L
000340: 4F41 4480 A450 4D44 424F 5254 85E3 004D OAD █PMDBORT█..M
000350: 4443 4F50 5982 EBB0 4D44 5245 5486 E300 DCOPY█._MDRET█..
000360: 5041 5553 4588 EB00 5044 5249 5645 83E9 PAUSE█..PDRIVE█.
000370: 0050 5249 4E54 86F0 8850 524F 5486 E900 .PRINT█.█PROT█..
000380: 5055 5247 4589 E900 5280 2300 5245 4E41 PURGE█..R #.RENA
000390: 4D45 81E4 B052 4F55 5445 81F0 0053 544D ME█._ROUTE█..STM
0003A0: 5489 EB00 5359 5354 454D 81E9 0054 494D T█..SYSTEM█..TIM
0003B0: 458A E900 5645 5249 4659 84E5 0057 5244 E█..VERIFY█..WRD
0003C0: 4952 5080 5300 0021 584F 0E08 0608 7ECB IRP S..!XO....B.
0003D0: 7F23 2005 CD9F 5110 F523 237E B7CA 9D51 # .. Q█.##B .█Q█
0003E0: 0DCC 9D51 28E4 CD95 5118 E1F3 CD2B 5121 ..█Q(..█Q....+Q!
0003F0: 6A43 7EE6 C020 333A 2240 F5ED 739B 43CB jCB.. 3:"§..s█C.

000400: FEFB 3E0D CD33 00C3 01EA F850 8A4D 216A ..>..3.....P█M!j
000410: 43CB 7ECA 444D ED7B 9B43 3E0E CD33 00F1 C.B.DM.ä█C>..3..
000420: B747 3E0F CC33 0078 3222 40F3 CBBE AF08 █G>..3.x2"§..█.
000430: FDE1 DDE1 F1C1 D1E1 D9C1 D1E1 08FB C9F1 ................
000440: E5D5 C508 D9E5 D5C5 F5DD E5FD E5D9 08F5 ................
000450: C9CD 5051 F57E D603 2802 D60A 2801 23F1 ..PQ.B..(...(.#.
000460: C911 8044 D506 20CD 5A51 D106 00C9 7EFE .. D.. .ZQ....B.
000470: 2A20 0412 1323 05E5 7ED6 30FE 0ACD 8951 * ...#..B.0...█Q
000480: 3016 7ED6 2EFE 0DCD 8951 3806 3E03 12F1 0.B.....█Q8.>...
000490: AFC9 7E12 1323 10EA F601 E17E C9D8 7ED6 █.B..#.....B..B.
0004A0: 41FE 1AD8 D620 FE1A C93E 20CD 9F51 10F9 A.... ...> .█Q..
0004B0: C93E 0DD5 F5CD 3300 F1D1 C943 4D44 4A43 .>....3....CMDJC
0004C0: 4C54 4F00 4D69 6E69 2D4E 6577 444F 532F LTO.Mini-NewDOS/
0004D0: 3830 2052 6561 6479 0D1C 1F03 0000 0000 80 Ready........
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0004F0: 0000 0000 0202 004D 0000 0000 0000 0000 .......M........
```



```
000000: 0100 004D FE23 CA8A 4DFE 4328 73FE 63CA ...M.#.█M.C(s.c.
000010: 304E FE83 CA3D 51FE A3CA 2A4F FEC3 283F ON.█.=Q.█.*O..(?
000020: 0D28 3A0D CABB 510D CA32 4D0D CADB 500D .(:..█Q..2M...P.
000030: 2852 0DCA FA50 0D28 430D CA34 4E0D 280C (R...P.(C..4N.(.
000040: 0D0D CA50 510D 283C 3E2A B7C9 21B8 51C3 ...PQ.(<>*█.!█Q.
000050: 6744 D901 01E3 11D3 49C5 D5D9 EFF1 C9CD gD......I.......
000060: 2B51 0100 00EB 216A 43CB 76CB F628 04ED +Q....!jC.v..(..
000070: 4B9D 43C5 ED73 9D43 EBC3 354E F1F1 180F K█C..s█C..5N....
000080: AF37 180B 216A 437E E62F 772B CBAE AFF3 █7..!jCB./w+.█.
000090: 216B 4336 002B 462B 4E1E 0BF5 CB50 2023 !kC6.+F+N....P #
0000A0: F1F5 3802 280A FE38 2806 1E04 CB69 2013 ..8.(..8(....i .
0000B0: CBB6 CB70 205E 3A6C 43CB 7728 0BCB 6920 .█.p ^:lC.w(..i
0000C0: 071E 0C16 EB7A 4BEF CB78 2014 31E0 41CB .....zK..x .1.A.
0000D0: 6F21 B045 2213 433E C328 023E C932 1243 o!_E".C>.(.>.2.C
0000E0: 21B2 51CB 7828 07ED 7B9B 4321 B051 FB3E !_Q.x(..ä█C!_Q.>
0000F0: 0DCD 3300 CB69 CC67 4421 6A43 CBEE 0108 ..3..i.gD!jC....

000300: 4154 C02B 0046 0100 FA4F 5245 4580 4A00 AT.(.F...OREE J.
000310: 4849 4D45 4D82 E900 4A4B 4C80 A510 4B49 HIMEM█..JKL █.KI
000320: 4C4C 8045 904C 4389 E500 4C43 4456 5288 LL E LC█..LCDVR█
000330: E500 4C49 4282 E300 4C49 5354 85F0 884C ..LIB█..LIST█.█L
000340: 4F41 4480 A450 4D44 424F 5254 85E3 004D OAD █PMDBORT█..M
000350: 4443 4F50 5982 EBB0 4D44 5245 5486 E300 DCOPY█._MDRET█..
000360: 5041 5553 4588 EB00 5044 5249 5645 83E9 PAUSE█..PDRIVE█.
000370: 0050 5249 4E54 86F0 8850 524F 5486 E900 .PRINT█.█PROT█..
000380: 5055 5247 4589 E900 5280 2300 5245 4E41 PURGE█..R #.RENA
000390: 4D45 81E4 B052 4F55 5445 81F0 0053 544D ME█._ROUTE█..STM
0003A0: 5489 EB00 5359 5354 454D 81E9 0054 494D T█..SYSTEM█..TIM
0003B0: 458A E900 5645 5249 4659 84E5 0057 5244 E█..VERIFY█..WRD
0003C0: 4952 5080 5300 0000 0000 0000 0000 0000 IRP S...........
0003D0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0003E0: 0000 0000 0000 0000 0000 00F3 CD2B 5121 .............+Q!
0003F0: 6A43 7EE6 C020 333A 2240 F5ED 739B 43CB jCB.. 3:"§..s█C.

000400: FEFB 3E0D CD33 00C3 01EA F850 8A4D 216A ..>..3.....P█M!j
000410: 43CB 7ECA 444D ED7B 9B43 3E0E CD33 00F1 C.B.DM.ä█C>..3..
000420: B747 3E0F CC33 0078 3222 40F3 CBBE AF08 █G>..3.x2"§..█.
000430: FDE1 DDE1 F1C1 D1E1 D9C1 D1E1 08FB C9F1 ................
000440: E5D5 C508 D9E5 D5C5 F5DD E5FD E5D9 08F5 ................
000450: C9CD 5051 F57E D603 2802 D60A 2801 23F1 ..PQ.B..(...(.#.
000460: C911 8044 D506 20CD 5A51 D106 00C9 7EFE .. D.. .ZQ....B.
000470: 2A20 0412 1323 05E5 7ED6 30FE 0ACD 8951 * ...#..B.0...█Q
000480: 3016 7ED6 2EFE 0DCD 8951 3806 3E03 12F1 0.B.....█Q8.>...
000490: AFC9 7E12 1323 10EA F601 E17E C9D8 7ED6 █.B..#.....B..B.
0004A0: 41FE 1AD8 D620 FE1A C93E 20CD 9F51 10F9 A.... ...> .█Q..
0004B0: C93E 0DD5 F5CD 3300 F1D1 C943 4D44 4A43 .>....3....CMDJC
0004C0: 4C54 4F00 4D2D 444F 5320 3E0D 1C1F 0321 LTO.M-DOS >....!
0004D0: 584F 0E08 0608 7ECB 7F23 2005 CD9F 5110 XO....B# .. Q█
0004E0: F523 237E B7CA 9D51 0DCC 9D51 28E4 CD95 .##B█.█Q..█Q(..█
0004F0: 5118 E100 0202 004D 0000 0000 0000 0000 Q......M........
```

Hier nun eine Auflistung aller NewDOS80-Befehle mit ihren Sprung-, Aufruf- und Schlußbytes

Befehl	Sprung-	Aufruf-	Schlußbyte	Systemfile
APPEND	C0	68	00	6
ATTRIB	85	E9	88	7
AUTO	84	E9	00	7
BASIC2	86	EB	00	9
BLINK	81	E5	00	3
BOOT	8A	EB	10	9
BREAK	85	E5	00	3
CHAIN	C3	EB	8A	9
CHNON	C5	EB	00	9
CLEAR	84	F0	00	14
CLOCK	82	E5	00	3
CLS	89	E3	10	1
COPY	C0	48	00	6
CREATE	82	F0	40	14
DATE	8B	E9	00	7
DEBUG	83	E5	00	3
DIR	80	2A	00	8
DO	C3	EB	8A	9
DUMP	87	E9	C8	7
ERROR	87	F0	00	1
FORMAT	C0	28	00	6
FREE	80	4A	00	8
HIMEM	82	E9	00	7
JKL	80	A5	10	3
KILL	80	45	10	3
LC	89	E5	00	3
LCDVR	88	E5	00	3
LIB	82	E3	00	1
LIST	85	F0	80	1
LOAD	80	A4	50	2
MDBORT	85	E3	00	1
MDCOPY	82	EB	B0	9
MDRET	86	E3	00	1
PAUSE	88	EB	00	9
PDRIVE	83	E9	00	7
PRINT	86	F0	88	1
PROT	86	E9	00	7
PURGE	89	E0	00	7
R	80	23	00	1
RENAME	81	E4	B0	2
ROUTE	81	F0	00	14
STMT	89	EB	00	9
SYSTEM	81	E9	00	7
TIME	8A	E9	00	7
VERIFY	84	E5	00	3
WRDIRP	80	53	00	17
und die neue Funktion "ID":				
ID	80	F8	00	22

Das war eine geballte Ladung Information, jetzt kommt wieder ein Praxisteil! Es geht weiter mit der "Eröffnung" der Systemfiles, die normalerweise nicht auf einer NewDOS80 -Diskette vorhanden sind (SYS22-SYS29), dort aber für unsere beabsichtigten Erweiterungen unbedingt gebraucht werden.

Eröffnen von Systemfiles

Wie wir schon gehört haben, werden die Systemfiles nicht über ihren Namen, sondern über ihren genau bestimmten Platz im Direktory aufgerufen (siehe "Grundlagen"). Auf einer "normale" NewDOS-Systemdiskette fehlen die Systemfiles 22-29. Aber um neue Funktionen implementieren zu können, brauchen wir diese Files. Hier nun ein Trick, wie man die fehlenden Systemfiles ins Direktory einträgt.

1. Man nehme eine NewDOS-Systemdiskette und erstelle mit dem Kommando COPY,0,1,,FMT,CBF,/SYS eine "Nursystem-diskette". Wenn man zuvor die PDRIVE-Parameter der Zieldiskette auf SS/SD/40 Tr. gesetzt hat, spart man später eine Menge Zeit.
2. Man fülle die Diskette mit Dummyfiles. Dabei hilft folgendes kurze Programm:

```
10 CLEAR1000 : CLS
20 CMD"FREE
30   INPUT"Wieviel   Fileeinträge  sind  im   Direktory  der
Diskette im Laufwerk 1 noch frei";FE
40 FOR X=1 TO FE
50 X$ = STR$(X) : Y = LEN(X$)-1
60 FE$ = RIGHT$(X$,Y)
70 CM$ = "CREATE,FILE"+FE$+"/DUM:1,REC=5"
75 PRINT§256,CM$
80 CMD"CM$
90 NEXT X
100 CMD"S=DIR 1
```

3. Mit SUPERZAP kann man nun feststellen, welche der eröffneten Files an der Stelle steht, an der normalerweise ein Systemfile erscheinen sollte.
4. Die Dummyfiles haben durch den Zusatz zum CREATE-Befehl REC=5 schon die richtige Länge. Den richtigen Inhalt bekommen sie durch das überkopieren des leeren SYS15/SYS (enthält im Model3-NewDOS die Befehle FORMS und SETCOM).
5. Die Dummyfiles bekommen nun durch den RENAME-Befehl noch ihren richtigen Systemfilenamen.
6. Zu guter Letzt kann man die nicht benötigten Dummyfiles weglöschen.

Achtung: wer das Betriebssystem GDOS (oder HDOS) sein eigen nennt, kann sich diese umständliche Prozedur sparen. GDOS/HDOS beinhaltet die kompletten Systemfiles, so daß man mit folgender, erheblich kürzerer Methode auskommt:
1. eine G/H-DOS-Systemdiskette mit dem oben schon einmal genannten Befehl COPY,0,1,,FMT,CBF,/SYS kopieren.
2. Die GDOS-Originaldisk durch ein NewDOS ersetzen und mit dem gleichen Befehl, die unter NewDOS vorhandenen Systemfiles auf das Backup des G/H-DOS aufkopieren.

Man erhält dadurch ein NewDOS mit allen Systemfiles, das zudem den Vorteil hat, daß man die interessanten GDOS- (noch besser HDOS) Funktionen schon auf der Diskette hat und nur noch in das vergrößerte Library des NewDOS einzubauen braucht.

PDrive-Identifikation für NewDOS80

Nachdem wir nun alle Bedingungen erfüllt haben, die vor der Erweiterung des NewDOS-Befehlssatzes zu erfüllen waren, können wir mit dem Einbau der verschiedenen Funktionen beginnen. Wir fangen mit der am schnellsten und einfachsten zu implementierenden Funktion an, der Identifizierungsroutine für PDriveangaben ID.

Die Funktion stammt aus dem H-DOS unseres Hacktory Arnulf Sopp und ihre Fähigkeit PDriveangaben zu erkennen geht so weit, daß sie sogar die Option TI=xxL erkennt. Diese Option wird benötigt, wenn man 40 Track-Disketten auf 80 Track-Laufwerken lesen und schreiben will und bereitet den sonstigen im Umlauf befindlichen Einstellroutinen unlösbare Schwierigkeiten. Der Aufruf erfolgt mit dem Syntax ID,n<,A>, wobei "n" die Nummer des Laufwerks darstellt, welche die zu testende Diskette enthält. Wird die Option ",A" verwendet, schreibt die Routine die richtigen PDriveangaben gleich auf die Systemdiskette und macht sie wirksam.

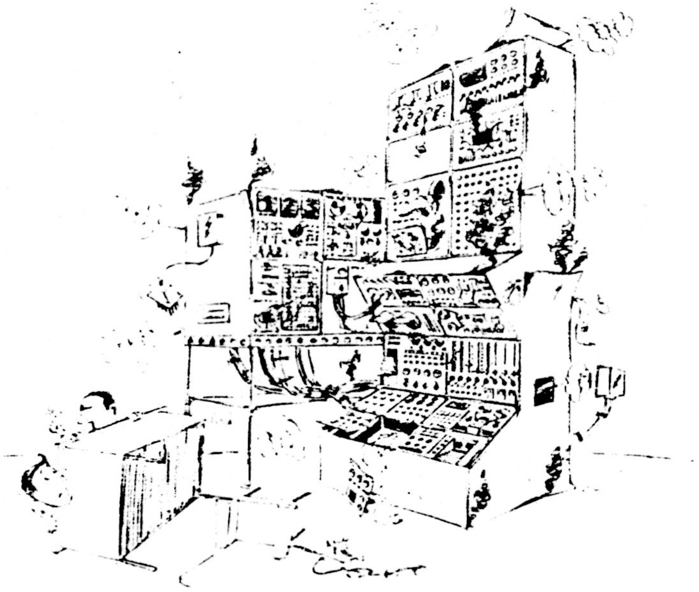
Der Einbau dieser sehr wertvollen Funktion ist deshalb so einfach, weil sie praktisch aus dem HDOS ohne jede Änderung übernommen werden kann. ID liegt im HDOS im SYS22/SYS und wird folgendermaßen im Library eingetragen: 49 44 80 FB 00 hex
I D . . . ASCII

Vorgehensweise:
1. SYS22/SYS von HDOS auf SYS22/SYS von NewDOS80 (vorher eröffnen!!!) kopieren.
2. Mit SUPERZAP im SYS1/SYS Sektor 3 rel. die oben aufgeführten Bytes ab dem Byte C6 rel. eintragen.
   Achtung: Nach dem Befehl WRDIRP und dem zugehörigen Sprung- und Aufrufbyte folgt noch das Schlußbyte (00) des Befehls. Dieses Bytes darf nicht überschrieben werden! Die Bytefolge in der Reihe "C0 - CF" des Sektordumps muß lauten:
   C0 49 52 50 80 53 00 49 44 80 F8 00 00 00 00 00 00
   I R P S . I D . . . . . .

Nachdem diese Änderungen durchgeführt sind, hat man zum erstenmal das Gefühl, die Arbeit, die man sich mit dem "Platz machen" im SYS1/SYS und der "Eröffnung" der Systemfiles gemacht hat, hat sich gelohnt. Und ich kann euch versichern, es werden noch eine ganze Menge solcher Erfolgserlebnisse folgen, wenn ihr euch selbst einmal daran macht, interessante und nützliche Funktionen (auch wenn sie durch andere Programmierer entwickelt wurden und praktisch "geklaut" sind*) ins NewDOS einzubauen.
Viel Spaß dabei wünscht euch, euer

Hartmut Obermann

* Es wird sicher niemand etwas dagegen haben, wenn man Programme und Routinen, die man mit einem DOS zusammen erworben hat (z.B. GDOS + NewDOS -> an mich = HDOS von Arnulf Sopp!!!) in ein anderes DOS einbaut, solange man dann damit nicht als Eigenentwicklung hausieren geht oder sogar Geld zu machen versucht. Aber wie ich die Clubmitglieder kenne, wird so etwas sicher nicht der Fall sein!



»Hundertmal habe ich dir gesagt: Keine widersprüchlichen Fragen stellen!«

„Hast du dir erst Wissen erworben, so weißt du, was dir fehlt."

DDE & LWT

Nachdem man das SYS1 geändert und sich im Library Platz für neue Befehle geschaffen hat, ist man bestrebt möglichst viele Funktionen ins DOS einzubauen. Dabei sollte man jedoch bedenken, daß die 36 freien Libraybytes sehr schnell gefüllt sind und man deshalb bei der Wahl der einzubauenden Befehle ein paar Dinge beachten sollte.

Die meisten Funktionen, die das DOS zur Verfügung stellt, könnte man auch durch "normale" Maschinenprogramme erledigen lassen. So ist z.B. die FORMAT-Routine durchaus als /CMD-File denkbar (siehe das alte TRSDOS 2.3). Sinn macht der Einbau ins Betriebssystem eigentlich nur, wenn man die Funktion von einem Programm oder von MiniDOS aus aufrufen will. Bei den GDOS-Funktionen DDE (DiskDaten Editor) und LWT (Laufwerktest) ist das fraglich!

Wer will schon mitten in einem BASIC-Programm einen Laufwerktest machen!? Ich glaube nicht, daß es dafür eine Anwendung gibt. Es genügt also, den Laufwerktest als CMD-File zu übernehmen und nicht ins DOS einzubauen.

Beim DiskDatenEditor verhält sich die Sache etwas anders. Hier gäbe es manchmal durchaus sinnvolle Anwendungen, die einen Einbau ins Library rechtfertigen würden. Leider wird dies durch den Aufbau von DDE verhindert. Das Programm benutzt nämlich den Bereich um 5500h als Sectorbuffer. Dieser Bereich wird aber auch von vielen Programmen (so auch von BASIC) benutzt. Dadurch ist ein Aufruf von DDE aus dem BASIC zwar möglich, die Rückkehr endet jedoch immer in einem Systemabsturz! Also ist auch die DDE-Funktion nicht als DOS-Befehl, sondern als CMD-File zu übernehmen.

Der Einbau von DDE

DDE ist in seinem Systemfile (HDOS SYS15/SYS) ganz allein und ist aus diesem Grunde wie ein ganz normales Maschinenprogramm aufgebaut. Um es ins NewDOS zu übernehmen, muß man es nur von einer HDOS-Diskette mit dem Befehl:

```
COPY SYS15/SYS:1 TO DDE/CMD:0
```

herüberziehen. Es läuft auf Anhieb und benötigt keine weiteren Zap's.

Der Einbau von LWT

LWT macht da schon etwas mehr Probleme. Es muß nämlich, um als CMD-File lauffähig zu sein, von seinem Lade-/Laufbereich 4D00h-51E7h nach 5D00h-61E7h (Startaddr. 5D7Ch) verlegt werden. Außerdem muß man, wenn man ein TRS80-Modell 1 besitzt, zwei Bytes im Sektor 0 rel. ändern. Sonst stimmt die angezeigte Geschwindigkeit nämlich nicht, die Laufwerke laufen immer zu langsam. Zusätzlich sollte man die Maßeinheiten von der Anzeigeskala entfernen, da sie nicht mehr stimmen.

Die Verlegung in den höheren Bereich erreicht man am leichtesten, indem man sich das SYS23/SYS des HDOS von einem Disassembler als Sourcecode auf Diskette schreiben läßt, die ORG-Adresse ändert und wieder, jetzt als LWT/CMD, assembliert. Die richtige Startadresse kann man nachträglich mit DDE ändern (letzter Sector von LWT/CMD, Bytes FE und FF von 005Dh in 7C5Dh ändern).

Damit die Anzeige für den TRS80 passt, müssen im rel. Sector 0 die Bytes C6 und C7 von C858 in 2859 geändert werden. Die Löschung der Maßeinheiten der Anzeigeskala erfolgt im relativen Sektor 1 im Bereich 20h bis [illegible].

HDOS SYS23/SYS SECTOR 0 REL.

```
000000: 0102 004D FEF9 2030 0D28 710D CA63 500D ...M.. 0.(q..cP.
000010: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000020: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000030: 0000 0000 0000 0000 3E2A C900 0000 0000 ........>*......
000040: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000050: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000060: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000070: 0000 0000 ED7B 6E4D AFC9 0000 0000 0000 .....änM█.......
000080: ED73 6E4D F316 0031 5F50 D5CD C901 210D .snM...1_P....!.
000090: 4E11 B03E 0180 00ED B021 8D4E 1140 3C01 N. >. .._!█N.§<.
0000A0: 8001 EDB0 D121 EC37 3A40 38FE 0420 06CD  .._.!.7:§8.. ..
0000B0: C901 FB18 BFCD E94D 0100 00CD E94D E52A ....█..M.....M.*
0000C0: 0D50 3E20 7721 C858 AFED 42CD 8051 3E3F .P> w!.X█.B. Q>?
0000D0: BC38 0E20 07CB 3DCB 3D00 1808 2100 3F18 █8. ..=.=...!.?.
0000E0: 0321 3F3F 3E2A 7722 0D50 E118 BB3A 1038 .!??>*w".P..█:.8
0000F0: B720 1503 CB4E 28F5 3A10 38B7 200A 03CB █ ...N(.:.8█ ...
```

NEWDOS80 LWT/CMD SECTOR 0 REL.

```
000000: 01FE 005D FEF9 2030 0D28 710D CA63 600D ...Ü.. 0.(q..c`.
000010: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000020: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000030: 0000 0000 0000 0000 3E2A C900 0000 0000 ........>*......
000040: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000050: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000060: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000070: 0000 0000 ED7B 6E5D AFC9 0000 0000 0000 .....änü█.......
000080: ED73 6E5D F316 0031 5F60 D5CD C901 210D .snü...1_`....!.
000090: 5E11 803E 0180 00ED B021 8D5E 1140 3C01 ^. >. .._!█^.§<.
0000A0: 8001 EDB0 D121 EC37 3A40 38FE 0420 06CD  .._.!.7:§8.. ..
0000B0: C901 FB18 BFCD E95D 0100 00CD E95D E52A ....█..Ü.....Ü.*
0000C0: 0D60 3E20 7721 2859 AFED 42CD 8061 3E3F .`> w!(Y█.B. a>?
0000D0: BC38 0E20 07CB 3DCB 3D00 1808 2100 3F18 █8. ..=.=...!.?.
0000E0: 0321 3F3F 3E2A 7722 0D60 E118 BB3A 1038 .!??>*w".`..█:.8
0000F0: B720 1503 CB4E 28F5 3A10 38B7 200A 03CB █ ...N(.:.8█ ...
```

HDOS SYS23/SYS SECTOR 1 REL.

```
000100: 4E20 F57A 0102 004E 32E1 37C9 E60F 5732 N .z...N2.7...W2
000110: E137 C383 4D7A 7520 6C61 6E67 7361 6D20 .7.█Mzu langsam
000120: 2020 3239 392E 3420 2020 2020 2020 2020    299.4
000130: 2020 2033 3030 2020 2020 2020 2020 2020     300
000140: 2020 3330 302E 3620 2020 7A75 2073 6368    300.6   zu sch
000150: 6E65 6C6C 2090 9590 9090 9094 9090 9090 nell █.....█....
```

NEWDOS80 LWT/CMD SECTOR 1 REL.

```
000100: 01FE FC5D 4E20 F57A 32E1 37C9 E60F 5732 ...ÜN .z2.7...W2
000110: E137 C383 5D7A 7520 6C61 6E67 7361 6D20 .7.█Üzu langsam
000120: 2020 2020 2020 2020 2020 2020 2020 2020
000130: 2020 2033 3030 2020 2020 2020 2020 2020     300
000140: 2020 2020 2020 2020 2020 7A75 2073 6368           zu sch
000150: 6E65 6C6C 2090 9590 9090 9094 9090 9090 nell █.....█....
```

Viel Spaß beim Zapen und Arbeiten mit DDE und LWT, wünscht euch,

Hartmut Obermann

Ein Byte und seine Folgen

Welch seltsame Symptome bei der Änderung eines einzelnen Bytes im SYS1/SYS auftreten können, mußte ich vor kurzem am Beispiel des Programmes HELP/CMD erfahren.

Nachdem ich im System 1 das Abschlußbyte der DOS-Ready-Meldung von OD in 03 geändert hatte (der Cursor steht dann hinter der Meldung, statt eine Zeile tiefer), bekam ich beim Aufruf von HELP,Befehl immer nur den freundlichen Hinweis, daß HELP die NEWDOS-Befehle erklärt und mit dem Syntax HELP,Befehl aufgerufen wird. Alle Bemühungen, dem Programm etwas anderes zu entlocken, scheiterten!

Eine kurze Durchsicht des disassemblierten Listings deckte den Grund für das seltsame Verhalten des Programms auf.
Das hinter dem HELP eingegebene Suchargument für den Erklärungstext wird auf folgende Art gerettet:

```
LD      HL,(4020H)      'Cursoradresse -> HL
LD      DE,003BH        'Zeilenlänge (64)-"HELP,"(5)=59
SBC     HL,DE           'HL-DE=Beginn des Arguments
```

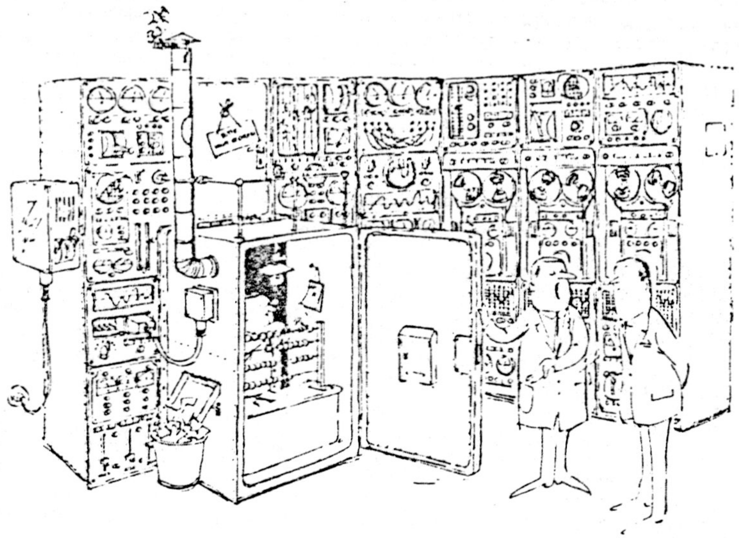
Durch die Änderung im Sys1 verschiebt sich der abzuziehende Wert erheblich. Er beträgt dann:
Zeilenlänge(64)-"HELP,"(5)-DOS-Meldung(unterschiedlich).
Beispiel: DOS-Ready: HELP,AUTO abzuziehender Wert= 31H!

Das betreffende Byte findet man im Sector 0 (relativ) als Byte 18 (relativ) in HELP/CMD.
Dieses Byte muß auch angepaßt werden, wenn der Name des Programms (HILF/CMD) so geändert wird, daß er mehr (oder weniger) als vier Buchstaben hat!

Wie man sieht, können kleinste Änderungen am DOS oder in Programmen unerwartete, teilweise fatale Folgen nach sich ziehen. Ich bitte also jeden, der in dem von mir geänderten NewDOS einen Fehler findet, mir diesen mitzuteilen, damit dieser schnellstmöglich behoben werden kann!
Im voraus vielen Dank!!!

Hartmut Obermann


```
DOS> HELP,ATTRIB█

000000: 0102 006C 0000 0000 0000 00DD 2100 00DD ...l........!...
000010: 3931 F0DF 2A20 4011 3600 AFED 5211 006C 91..* §.6.█.R..l
000020: 0606 7EFE 2028 07CD 9B6C 2313 10F4 AF12 ..ß. (..█l#...█.
000030: 21B0 6C11 F497 EBAF ED52 EBD5 C13E 25ED !_l..█.█.R...>%.
000040: B1C2 4F6C 1100 6C1A B7CA 646C BE20 E423 █.Ol..l.█.dl█ .#
000050: 1318 F42A 2040 11D4 FF19 EB21 A76C 010B ...* §.....!█l..
000060: 00ED B0DD F9C3 2D40 010A 003E 25ED B923 ..█...-§...>%.█#
000070: 23CD C901 0140 0011 003C D5CD 816C 23D1 #....§...<..█l#.
000080: EB09 EB18 F57E FE0D C8FE 2528 0512 1323 .....ß....%(...#
000090: 18F3 D1D1 EB09 0922 2040 DDF9 C32D 40CB ......." §...-§.
0000A0: F7CB AF12 C900 0000 0000 0020 2020 4552 ::█.........   ER
0000B0: 524F 5220 2020 2020 2548 454C 500D 4865 ROR      %HELP.He
0000C0: 6C70 2069 7374 2065 696E 2048 696C 6673 lp ist ein Hilfs
0000D0: 7072 6F67 7261 6D6D 2066 7565 7220 4E45 programm fuer NE
0000E0: 5744 4F53 2D38 3020 322E 3020 2E0D 416C WDOS-80 2.0 ..Al
0000F0: 6C65 2042 6566 6568 6C65 2064 6965 2062 le Befehle die b
```



».. . und das ist das eigentliche Gehirn der Anlage!«

HRG-Pixel's aus dem BASIC !!

Vor kurzem befasste ich mich mit dem Problem der HRG-Ansteuerung vom Basic aus, um ein paar Linien für eine Eingabemaske setzen, ohne einen HRG-Treiber zu laden.

Nachdem mir zu diesem Thema noch Verschiedenes unklar war, habe ich mich dazu im Club umgehört, und fand beim Arnulf schon einige ausgearbeitete Werke, die ich Euch nach dieser kleinen Einleitung vorstellen möchte.

Zuerst aber noch folgendes:
Das Bearbeiten einer ganzen Grafikseite (immerhin 12K) vom Basic aus, ist eine doch recht langwierige Angelegenheit. So dauert zum Beispiel das Löschen der HRG mit dem schnellsten Basicprogramm zirka 2 Minuten. Wer aber nur schnell eine Linie ziehen- oder ein Kästchen zeichnen will, dem seien folgende Programmlistings, am Ende dieses Artikels, an's Herze gelegt.

Nachdem die Hauptprogrammteile schon vom Arnulf geschrieben waren, blieb mir eigentlich nur noch übrich einige Features dazu zu setzen und das Ganze im Clubinfo zu veröffentlichen.

Ich hoffe, damit noch einigen von Euch gedient zu haben, zumal die HRG doch schon weit verbreitetes Zubehör unseres Computers ist.

JENS NEUEDER

Es folgt nun ein Auszug eines Briefes von Arnulf an mich. Danach kommen drei Beiträge zu der "HRG-Pixelei", die den Brief und sich gegenseitig noch ergänzen. Zum Abschluß des Ganzen die versprochenen Listings, die als Programme unter Pix1/BAS und Pix2/BAS in der Clubbibliothek zu finden sind.

Das Ganze ist nicht besonders schwierig. Verstehen mußt Du eigentlich nur, wie man ein ganz bestimmtes Byte innerhalb des externen HRG-Speichers adressiert. Der Speicher hat 12 kB, daher gehen seine Adressen von 0000-2FFF. Da man auf einen Port nur höchstens FF ausgeben kann, nicht aber z. B. 1AB3 oder so, muß die HRG-Adresse in zwei Bytes aufgeteilt werden. Das MSB ist Adresse/256, das LSB AdresseAND255. Das LSB wird auf Port 2 ausgegeben, das MSB auf Port 3. Jetzt erst kann man ein Byte über Port 5 in die HRG einschreiben (OUT5,byte) oder über Port 4 aus der HRG lesen (byte=INP(4)).

Jedes Byte steht für 6 Punkte auf dem Bildschirm. Die beiden höchstwertigen Bits (also der 7. und 8. Punkt) sind auf dem Bildschirm nicht sichtbar. Wenn Du nun an einer bestimmten HRG-Adresse einen bestimmten von 6 Punkten setzen willst, mußt Du wissen, welchem Bit im

ausgegebenen Byte dieser Punkt entspricht. Ganz links ist der 1. Punkt, der mit dem Bit 0 gesetzt wird. Das bedeutet, daß die binäre Zahl (Dein Byte) umgekehrt gedacht werden muß: Die niedrigste Stelle ist links, die höchste rechts. In einem Byte entsprechen die Bits 0-5 (nur diese 6 sind wichtig) den Zahlen 1, 2, 4, 8, 16, 32. Also: Wenn Du über Port 5 das Byte 16 ausgibst, nachdem Du über die Ports 2 und 3 die Adresse definiert hast, dann wird der 5. Punkt von links gesetzt. Um nun nicht bei der Ausgabe eines Punktes die übrigen Punkte an dieser HRG-Adresse zu löschen, ist es wichtig, daß Du so vorgehst: X=INP(4):Y=16:Z=XORY:OUT5,Z oder: OUT5,INP(4)OR16.

Für das Rücksetzen eines Punktes gilt das Entsprechende: Zunächst mußt Du das Komplement des Bytes finden (das zu löschende Bit steht auf 0, alle anderen auf 1). Bevor ich Dir jetzt lange Opern über das binäre Einerkomplement vorsinge, gebe ich Dir eine Tabelle wie die obige, aber mit 0 im angesprochenen Bit und 1 in den übrigen (wieder für Bit 0-5): 254, 253, 251, 247, 239, 223. Jetzt nehmen wir wieder das obige Beispiel. Der 5. Punkt von links wird diesmal aber gelöscht: X=INP(4): Y=239:Z=XANDY:OUT5,Z oder: OUT5,INP(4)AND239. Statt 16 steht nun 239, statt OR steht AND, weil ein Bit ausmaskiert werden muß.

Oben sprach ich das Adressieren an. Hier haben wir zufällig eine Erleichterung. Die LSB der HRG-Adressen stimmen mit denen des normalen Bildschirms genau überein. Bei den ersten und allen weiteren vier (normalen und HRG-) Bildschirmzeilen geht das LSB von 0 bis FF, also in der 1. Zeile von 0-63, in der 2. von 64-127, in der 3. von 128-191 und in der 4. von 192-255. So kannst Du Dir das LSB aus der Video-Display-Map im TRS-80-Handbuch (Appendix E) herausfiedeln. Mit dem MSB ist es auch nicht viel komplizierter: Die Bildschirmadressen gehen von 3C00-3FFF, die HRG-Adressen von 0000-2FFF. Das ist ein Unterschied von 3C00 (15360 dez).

Ich fange einen neuen Absatz an, aber nur aus optischen Gründen. Nun kannst Du beim Setzen oder Rücksetzen eines Punktes in der Video-Display-Map die Adresse heraussuchen und 15360 subtrahieren. Das ist dann die Adresse in der HRG, wenn der Punkt in der obersten Punktzeile dieser Anzeigestelle gemeint ist. Meinst Du aber eine tiefere Punktzeile, dann addiere pro Punktzeile 1 kB (= 0400 hex oder 1024 dez) hinzu. Das ist dann die korrekte HRG-Adresse. Dazu paßt dann der Schmus aus meinem Artikel "Die HRG 1b programmieren".

Himmel, gerade las ich mir das Vorige noch einmal durch und verstand kaum eine Silbe! Es ist schon ein bißchen vigeliensch, as wi hier op platt seggt. Lies Dir bitte die drei Artikel je dreieinhalbmal durch, jeweils durch ein Pils oder Alt unterbrochen, und lies anschließend diesen Brief laut Deiner Katze vor. Wenn dann noch Fragen offen sind, bastele einen Molotow-Cocktail, löse eine Rückfahrkarte nach Eitorf/Sieg und statte der Fa. RB-Elektronik einen Besuch ab. Grüße von mir!

Die HRG 1b programmieren

Die Zusatzplatine für hochauflösende Graphik hat innerhalb des Clubs bereits etliche Freunde gefunden. Vielleicht sind einige von Euch von der chaotischen Anleitung immer noch so gefrustet wie ich es war, als ich mir gestern nach dem Einlöten die Früchte des soeben abgewischten Schweißes gönnen wollte. Mit der gegen Aufpreis erhältlichen Software geht es zwar ganz gut, aber die blockiert mal wieder das Himen. Außerdem bietet sie überwiegend zusätzliche BASIC-Befehle, so daß der Assembler-Programmierer letztenendes mit seiner HRG alleingelassen ist. Nicht ganz, siehe Fotokopie der einzigen Seite der Anleitung, die überhaupt (obgleich in BASIC) auf die direkte Programmierung der Karte eingeht. Wer aber nach Lektüre dieser Seite schlauer ist als zuvor, verdient Bewunderung.

So schrieb ich eine Stelle dieser Seite zunächst von hinten nach vorne, um mit der sonderbaren Bitphilosophie mehr anfangen zu können:

```
      >       Port 3  (HOB)        < >      Port 2  (LOB)        <
Bit:   7   6   5   4   3   2   1   0   7   6   5   4   3   2   1   0
      > n.b.< >  Bereich C  < >  Bereich B  < >      Bereich A    <
```

Jetzt liest sich das Ganze wie eine 16-Bit-Zahl und beginnt, einen Sinn zu bekommen. Diese Zahl kann man sich sehr wohl als Adresse eines Bytes im Graphikspeicher vorstellen. Sie ist es auch, wie die selbstquälerische wiederholte Lektüre dieser Anleitung (besser: Verschleierung) und ein paar Stunden an der Tastatur ergaben. Das Resultat war eine Tabelle, die auf der nächsten Seite oben wiedergegeben ist.

Das Video-RAM kann man sich wie eine Perlenschnur vorstellen, wo sich Byte an Byte fügt. Mit dem RAM der Graphikkarte ist es nicht anders. Allerdings belegen die Bytes des Video- und des Graphik-RAMs auf dem Bildschirm physikalisch leider nicht denselben Platz. Das ist ja auch logisch, wenn 12 kB in den Raum passen sollen, den normalerweise 1 kB belegt. Außerdem ist die Reihenfolge nicht die gewohnte, in der der Inhalt des Graphikspeichers ausgegeben wird. Das erste kB wird genau da abgelegt, wo es auch das Video-RAM tut, aber natürlich nur die oberste Punktreihe einer jeden Bildschirmstelle. Dann wiederholt sich dieser Vorgang, wobei nach jedem Kilobyte die nächste Punktreihe angesteuert wird, bis alle 12 Reihen geladen und die 12 kB des Graphikspeichers abgearbeitet sind.

Deshalb ist für die wiedergegebene Tabelle folgende Erläuterung erforderlich: Je nach Punktreihe muß der angegebenen Zahl ein Vielfaches von 0400 (1024 dez.) hinzugezählt werden. Oder einfacher: Das MSB der Adresse wird nach jeder Punktreihe um 4 erhöht.

Die gewohnten Videoadressen gehen von 3C00-3FFF, die Graphikadressen intern von 0000-2FFF. Betrachtet man nur die oberste Dotreihe, so gehen sie von 0000-03FF! Das ist gleich Videoadresse minus 3C00, weiter nichts! Da das LSB (auf Port 2) exakt mit dem des Video-RAMs übereinstimmt, haben wir es mit simpelster Mathematik zu tun, auch wenn die Anleitung eher an eine schöngeistige Diziplin denken läßt. Freilich darf man nicht vergessen, dem MSB je nach seiner Dotreihe 00, 04, 08, 0C usw. hinzuzuaddieren.

Die zweite Tabelle auf der nächsten Seite gibt ein Beispiel für eine einzelne Bildschirmstelle. Es ist die Stelle 3D00, also das erste Byte der 5. Zeile. Die zweite Spalte der Tabelle zeigt die Nachbarstelle zur Rechten (3D01). Daraus erkennt man, daß das LSB wie beim normalen Video-RAM behandelt werden kann. Ganz rechts ist die letzte Stelle der Zeile wiedergegeben. Der Anwender braucht nun nur noch zu wissen, welche der sechs möglichen Punkte er in der betreffenden Dotreihe setzen möchte. Jedem Punkt ist ein Bit von 0-5 zugeordnet. Dieser Wert geht über Port 5 ins Graphik-RAM, bzw. er kann von dort über Port 4 ausgelesen werden.

Zugegeben, diese Erläuterung ist auch nur denjenigen Lesern auf Anhieb verständlich, die mit Binär- und Sedezimalzahlen vertraut sind. Das trägt mir voraussichtlich wieder Senge von Seiten der BASIC-Lobby ein. Aber der BASIC-User hat mit der lieferbaren Software brauchbare Werkzeuge, so daß die zitierte Seite der HRG-Anleitung für ihn ohnehin von geringem Interesse ist.

An der HRG-Anleitung ist weitere Kritik angebracht! Auf der abgebildeten Seite wird im ersten Absatz gesagt, wie die Ports anzusteuern sind (Input bzw. Output). Tatsächlich aber reagieren alle 6 Ports bei einem Input so, als hätte ein Output von FF (255 dez.) stattgefunden. Man kann demnach die Graphik auch durch Lesen der Ports 0 und 1 ein- oder ausschalten. Das ist bei den G-DOS-Befehlen PORT und INFO leider fatal, denn sie lesen alle Ports. Ergebnis: Die prompt eingeschaltete Graphik übermalt die Anzeige, derentwegen wir ja schließlich den Befehl eingegeben haben.

So ist denn zwar die HRG-Platine nicht ganz ohne Tücken (die Anleitung ist sogar die Tücke selber), aber letztenendes kocht sie auch nur mit Wasser und kann gehandhabt werden.

Und wie sie gehandhabt werden kann! Seit den obigen Zeilen sind zwei Tage vergangen, und vor einer Stunde war eine Modifikation meiner Graphik-Hardcopy-Routine fertig, die bei <JKL> auf Wunsch auch noch die hochauflösende Graphik mit ausdruckt, nach Lust und Laune in positiver oder negativer Darstellung, in jedem Falle gemischt mit Text und der normalen Pixelgraphik.

Das Video-RAM und die zugeordneten HRG-Adressen:

1. Bildschirmzeile:	0000	0001	0002	....	003D	003E	003F
2. Bildschirmzeile:	0040	0041	0042	....	007D	007E	007F
3. Bildschirmzeile:	0080	0081	0082	....	00BD	00BE	00BF
4. Bildschirmzeile:	00C0	00C1	00C2	....	00FD	00FE	00FF
5. Bildschirmzeile:	0100	0101	0103	....	....	....	....
12. Bildschirmzeile:	....	....	....	....	02FD	02FE	02FF
13. Bildschirmzeile:	0300	0301	0302	....	033D	033E	033F
14. Bildschirmzeile:	0340	0341	0342	....	037D	037E	037F
15. Bildschirmzeile:	0380	0381	0382	....	03BD	03BE	03BF
16. Bildschirmzeile:	03C0	03C1	03C2	....	03FD	03FE	03FF

Die 5. Bildschirmzeile (3D00 - 3D3F) mit HRG:

1. Dotreihe:	0100	0101	0102	....	013D	013E	013F
2. Dotreihe:	0500	0501	0502	....	053D	053E	053F
3. Dotreihe:	0900	0901	0902	....	093D	093E	093F
4. Dotreihe:	0D00	0D01	0D02	....	....	....	....
9. Dotreihe:	....	....	....	....	213D	213E	213F
10. Dotreihe:	2500	2501	2502	....	253D	253E	253F
11. Dotreihe:	2900	2901	2902	....	293D	293E	293F
12. Dotreihe:	2D00	2D01	2D02	....	2D3D	2D3E	2D3F

Arnulf Sopp

Die HRG 1b und BASIC netto

Für die hochauflösende Graphikplatine HRG 1b gibt es etliche Treiber, die eine große Menge neuer Befehle zu ihrer Ansteuerung anbieten. Viele dieser Features sind ebenso leistungsfähig wie überflüssig. Im rauhen Alltag geht es darum, z. B. irgendeinen Funktionsgraphen zu zeichnen oder eine Balken- oder Tortengraphik. Das kann der Microsoft-Interpreter schon lange; es brauchen nicht große RAM-Bereiche, meist im kostbaren Himem, durch Treibersoftware verschlissen zu werden.

Das hier vorgestellte BASIC-Programm (jawoll!) ist ein vollständiger Treiber für solche HRG-Funktionen, die den Befehlen SET, RESET und POINT entsprechen. Die Koordinaten X und Y (für 384*192 Punkte) werden im Target-Programm (hier Zeilen 10-40) geladen. Die Auswahl einer dieser drei Funktionen erfolgt durch Ansprung eines der drei Unterprogramme ab Zeile 10000. Das UP an 40000 ist allen drei gemeinsam und wird später ausführlich erklärt werden. Es ist quasi die Seele des Ganzen.

Zu den Zeilen 10-40 möchte ich weiter nichts sagen. Da wird halt eine Sinuskurve gemalt (Abb.). Diese Anwendung ist vollkommen willkürlich, damit die Wirkung des Treibers ab Zeile 10000 demonstriert werden kann. Erwähnenswert wäre nur der DOS-Befehl CLS,G in Zeile 10. Er ist eine Spezialität von H-DOS. Mit NEWDOS oder G-DOS wäre es auch mit BASIC-pur gegangen, aber wozu Zeit vergeuden?

Die Zeile 40000 errechnet aus den Koordinaten X und Y die physikalische Lage eines Punktes innerhalb des HRG-Speichers. Dazu muß zunächst X durch 6 dividiert werden, weil je sechs benachbarte Punkte in Wirklichkeit zu einem Byte gehören. Erst jeweils sechs Punkte später wird wieder ein neues Byte adressiert. Der Rest der Division (B) bestimmt das Bit und damit den Punkt in diesem Byte. Aber das Integer des Quotienten ist der Offset der Adresse nach rechts, weil die HRG-Adressen in waagerechter Richtung mit denen des normalen Bildschirms parallel laufen (die LSB der Adressen sind identisch).

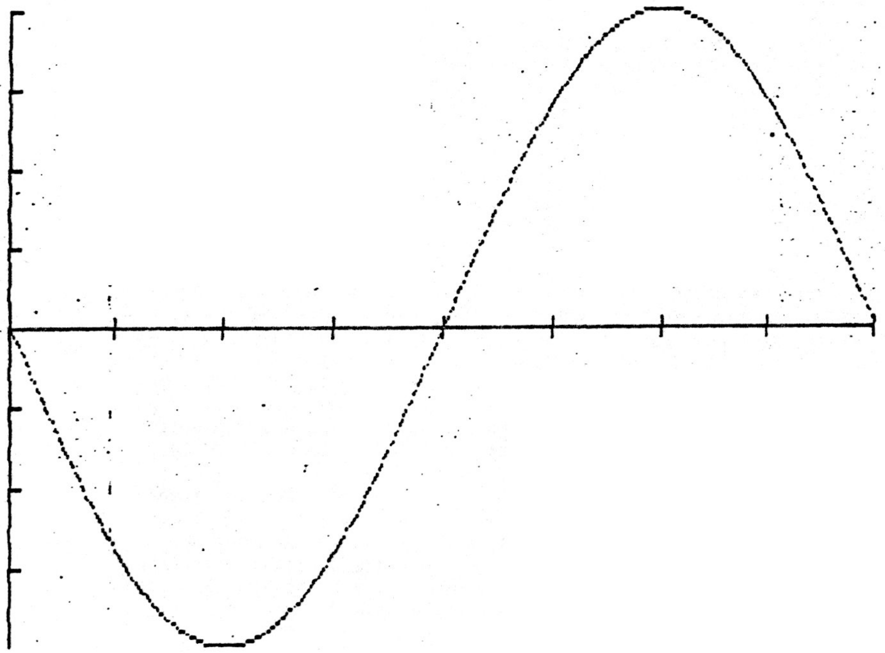
Für die senkrechte Richtung wird zunächst Y durch 12 dividiert. Zwei übereinanderliegende Punkte sind physikalisch nämlich gar nicht benachbart, sondern liegen 1 kB voneinander entfernt: 12 kB hat der HRG-Screen, es gibt pro normaler Bildschirmstelle 12 Punktzeilen. Der Rest der Division Y2 (ähnlich B für die X-Richtung) zählt die Punktzeilen innerhalb einer normalen Bildschirmstelle nach unten ab, bedeutet also je ein volles kB. Die physikalische Stelle des anzusprechenden Bytes im HRG-RAM (Z) errechnet sich daher aus Y2 Kilobytes plus Y1 Bildschirmzeilen plus Offset nach rechts (X1). Diese Adresse muß nun in ein Hibyte (ZM) und ein Lobyte (ZL) unterteilt werden, um sie über die Ports 2 und 3 anzusteuern. Anschließend wird noch der binäre Wert des Bits (des Punktes) errechnet (C gleich 2 hoch B). Schließlich wird für die spätere logische Verknüpfung noch A mit dem bisherigen Wert dieses Bytes geladen (über Port 4). Dieses ziemlich komplizierte Unterprogramm ist allen drei Funktionen gemeinsam.

Die SET-Routine in Zeile 10000 odeiert das neue Bit mit dem bisherigen Wert und gibt das Resultat als neues Bitmuster auf den Port 5 aus. Die RESET-Routine (20000) undiert den alten mit dem Komplement des neuen Wertes und gibt das Resultat aus. So verschwindet dieses Bit. Die POINT-Routine (30000) ordnet der Variablen D das Wahrheitsflag 0 (falsch, Punkt nicht gesetzt) oder -1 zu (wahr, Punkt gesetzt), genau wie bei der Klötzchengraphik.

Dieser Treiber ist so komplex, daß ich ihn selbst kaum kapiere. Wie man weiß, ist BASIC weder meine Leidenschaft noch meine Stärke. Deshalb wäre ich dankbar für Anregungen, wie man das Programm einfacher und vielleicht auch schneller machen könnte. Ein Bedarf ist sicherlich vorhanden, denn wer möchte schon z. B. ein Ungetüm wie GRAFE [illegible]n, um irgendwo einen lausigen Punkt zu setzen oder eine simple Kurve zu zeichnen wie die abgebildete?

Arnulf Sopp

```
10 CLS: CMD"CLS.G": OUT1,255%: DEFINT A-Z'
      Bildsch. u. Graph. löschen, Gr. einschalten, nur Integer
20 Y=96%: FOR X=0 TO 383%: GOSUB10000: NEXT: FOR X=47 TO 383 STEP 48: FOR Y=93 TO 99%:
 GOSUB10000: NEXT: NEXT'
      Abszisse mit Teilungen ziehen
30 X=0: FOR Y=0 TO 191%: GOSUB10000: NEXT: FOR Y=0 TO 191 STEP 24: FOR X=1 TO 5%: GOSU
B 10000: NEXT: NEXT'
      Ordinate mit Teilungen ziehen
40 FOR X=0 TO 383%: Y=SIN(X/61%)*95.5+95.5: GOSUB10000: NEXT: END'    Sinusgraphen zie
hen
10000 GOSUB40000: OUT5,A OR C: RETURN'
      SET-Unterprogramm
20000 GOSUB40000: OUT5,A AND (NOT C): RETURN'
      RESET-Unterprogramm
30000 GOSUB40000: D=((A AND C)<>0%) :RETURN'
      POINT-Unterprogramm
40000 X1=INT(X/6%): B=X-X1*6%: Y1=INT(Y/12%): Y2=Y-Y1*12%: Z=Y2*1024%+Y1*64%+X1: ZM=Z/
256%: ZL=Z AND 255%: OUT2,ZL: OUT3,ZM: C=2%ÄB: A=INP(4): RETURN'
      gemeins. Unterpr., hauptsächlich Koordinaten bestimmen
```



Den HRG-Speicher löschen

In meinem Artikel "Die HRG 1b programmieren" versuchte ich aufzuzeigen, daß diese Karte nahezu ebenso einfach adressiert wird wie irgendein RAM-Bereich, denn sie hat intern ebenfalls gewöhnliche 16-Bit-Adressen. Jener Beitrag schwebte ein wenig im luftleeren Raum, denn es blieb bei der Theorie. Hier ein wenig Alltagspraxis. Das im Anschluß an diesen Text gelistete Assemblerprogramm löscht den HRG-Speicher, eine Funktion, die man zu Beginn einer jeden neuen Graphik braucht.

Die internen Adressen der HRG reichen von 0000-2FFF. Das bedeutet folgendes: Das höchste MSB einer Adresse beträgt 2F. Da in der Schleife LOOP1 der Akku dekrementiert wird, noch bevor er auf den Port geht, muß der Anfangswert 1 höher, also auf 30 gesetzt werden. Das geschieht im ersten Befehl LD BC,3002. Das MSB+1 steht nun im Register B, gleichzeitig lädt dieser Befehl C mit der Portadresse 02.

Im nächsten Schritt wird der Akku mit dem MSB geladen, um die Ausgabe auf den Port 03 vorzubereiten. Sie erfolgt nach dem oben erwähnten Dekrementieren. Mit dem LSB, das auf Port 02 ausgegeben werden muß, ist es etwas komplizierter. Genauer gesagt ist es genauso simpel; um jedoch Speicherplatz zu sparen, geht das Programm einen etwas gewundenen Weg:

Das LSB muß für jedes MSB, also 30h-mal, die Werte FF-00 durchlaufen, um alle 256 LSBs zu erreichen. In diesem Falle wird mit 00 begonnen und über FF bis 01 gegangen. Der Anfangswert steht nach XOR A im Akku. Mit diesem Wert wird B geladen, weil dieses Register den Befehl DJNZ ermöglicht. Auf diese Weise werden ebenso rückwärts fortlaufend alle Werte erzeugt wie beim MSB. Sie gehen nach dem Eintritt in die Schleife LOOP2 auf den Port 02.

Der Akku bleibt in dieser Schleife immer auf 00. Dieser Code bedeutet für die HRG eine leere Anzeigestelle, also ein Blank. Es wird über den Port 05 ausgegeben. Die jeweilige Speicherstelle der HRG ist damit gelöscht. Nach 256 Durchläufen dieser Schleife wird das alte MSB vom Stack geholt, wo es seit dem Befehl PUSH BC lag. Ebenfalls mit DJNZ nimmt es nacheinander alle Werte von 30-01 an. Um auf die tatsächlich nötigen Werte 2F-00 zu kommen, ist der Befehl DEC A notwendig, der in Zeile 130 steht. Das war es bereits. Mit RET geht es ab ins Betriebssystem, sei es DOS oder was auch immer.

Das ist wieder einmal so ein Feature, das gleich bei Booten laufen sollte, um den Müll von Zufallscodes zu beseitigen, die sich nach dem Einschalten im Graphikspeicher befinden. Bei H-DOS ist das der Fall. Um aber eine Graphik, die im Speicher steht, nicht nach jedem BOOT neu erzeugen zu müssen, kann das Löschen mit der gedrückten Shifttaste auch verhindert werden. Das gilt nota bene für H-DOS, nicht für das hier vorgestellte Programm.

Arnulf Sopp

```
8000            00100        ORG    8000H     ;Adresse beliebig
8000 010230     00110 START  LD     BC,3002H  ;B= MSB 30, C= Port 02
8003 78         00120 LOOP1  LD     A,B       ;A <- MSB der HRG-Adresse
8004 3D         00130        DEC    A         ;30 -> 2F usw.
8005 D303       00140        OUT    (03H),A   ;MSB der HRG-Adr. ausgeb.
8007 C5         00150        PUSH   BC        ;aktuelles MSB retten
8008 AF         00160        XOR    A         ;A <- 00 = 256
8009 47         00170        LD     B,A       ;LSB der HRG-Adresse
800A ED41       00180 LOOP2  OUT    (C),B     ;ausgeben
800C D305       00190        OUT    (05H),A   ;00 = Graphik-Blank
800E 10FA       00200        DJNZ   LOOP2     ;LSB 100 -> 01
8010 C1         00210        POP    BC        ;altes MSB und Port
8011 10F0       00220        DJNZ   LOOP1     ;nächste 256 Adressen
8013 C9         00230        RET              ;z. B. ins DOS
8000            00240        END    START
00000 mal genennt
34572 Zeichen verfügbar
```


Programmieren der HRG1b (a) mit Hilfe der Ports

Zur Ansteuerung der HRG1b werden folgende Ports benutzt:

Port 0:	Ausschalten des Grafikbildschirms	Output
Port 1:	Einschalten des Grafikbildschirms	Output
Port 2:	Untere Ansteueradresse (LOB)	Output
Port 3:	Obere Ansteueradresse (HOB)	Output
Port 4:	Einlesen eines Grafikpunktes	Input
Port 5:	Setzen eines Grafikpunktes	Output

Mit Port 0 und 1 schalten Sie die Grafik ein bzw. aus, wie mit #OPEN und #CLOSE.
Beispiel: OUT 0,0 = #CLOSE | OUT 1,0 = #OPEN

Mit Port 2 und 3 wird der Bildschirmpunkt adressiert, den Sie ansprechen wollen. Die Bits der beiden Ports sind wie folgt aufgeteilt:

```
          Port 2 (LOB)           Port 3 (HOB)
Bit:  0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
       >  Bereich A  <> Bereich B<> Bereich C<>n.b.<
```

Bereich A:	Bit 0-5	Port 2	Position der 64 normalen Spalten
Bereich B:	Bit 6+7 & 0+1	Port 2+3	Position der 16 normalen Zeilen
Bereich C:	Bit 2-5	Port 3	Position der 12 Zeilen pro Char.
n.b.	: Bit 6+7	Port 3	nicht benutzt

Mit Port 4 koennen Sie dann 6 nebeneinanderliegende Bildschirmpunkte einlesen. Benutzt sind Bit 0-5. Beispiel: X=INP(4) oder PRINT INP(4)

Mit Port 5 werden bis zu 6 nebeneinanderliegende Punkte gesetzt. Gebraucht werden nur Bit 0-5. Beispiel: OUT 5,63 ===> 6 Punkte

Beispiel: Sie wollen den Punkt 21,31 setzen.

```
#OPEN         = OUT 1,0
#SET(21,31)   = OUT 2,131   Bit 0,1,7 gesetzt
                OUT 3,32    Bit 5     gesetzt
                OUT 5,4     Bit 2     gesetzt
```

Beispiel: Sie wollen wissen ob der Punkt 100,100 gesetzt ist

```
#POINT(100,100),X = OUT 2,16    Bit 4     gesetzt
                    OUT 3,22    Bit 1,2,4 gesetzt
                    X=INP(4)    wenn in der Variablen
                                X jetzt das Bit 4
                                gesetzt ist, dann ist
                                der Punkt 100,100
                                vorhanden.
```

```
1  '  PIX1 /BAS              == D E M O V E R S I O N ==
2      OUT1,0:                 'Grafik anschalten
3      OUT0,0:                 'Grafik ausschalten
4      OUT2,X:                 'LSB-Adresse (0-255)
5      OUT3,X:                 'MSB-Adresse (0-47)
6      OUT5,X:                 'Ausgabewert (0-63) für 6 Pixel
7     X=INP(4):                'lesen des Ausgabewertes
10 '****************************************
11 '**    ändern der HRG vom Basic aus        **  von
12 '**    im zwölfzeilenstepp                 **  Arnulf SOPP
13 '**    schnellste Möglichkeit              **
14 '****************************************
15
CLS:OUT1,0:FORA=0TO47:OUT3,A:FORB=0TO255:OUT2,B:OUT5,0:NEXTB,A:OUT0,0
20 '****************************************
21 '**    ändern der HRG vom Basic aus        **  von
22 '**    der Reihe nach                      **  Jens NEUEDER
23 '**    zeilenweise                         **
24 '****************************************
25 CLS:A=0:OUT1,0
26 FORB=ATO47STEP4:OUT3,B:FORC=0TO63:OUT2,C+D:OUT5,1:NEXTC,B
27 D=D+64:IFD=256THEND=0:A=A+1
28 IFA<4THEN26ELSEOUT0,0
30 '****************************************
31 '**    ändern der HRG vom Basic aus        **  von
32 '**    der Reihe nach                      **  Jens NEUEDER
33 '**    spaltenweise                        **
34 '****************************************
35 CLS:OUT1,0:FORC=0TO63:A=0:D=0
36 OUT2,C+D:FORB=ATO47STEP4:OUT3,B:OUT5,0:NEXTB
37 D=D+64:IFD=256THEND=0:A=A+1
38 IFA<4THEN36ELSENEXTC:OUT0,0
40 '****************************************
41 '**    ändern der HRG vom Basic aus        **  von
42 '**    nach Bildschirmadressen             **  Jens NEUEDER
43 '**    (15360 - 16383)                     **
44 '****************************************
45 CLS:OUT1,0:P=-1:FORA=15360TO16383:FORC=0TO11:IFC=11THENP=63ELSEP=1
46 B=((C*1024)+A)-15360:OUT2,BAND255:OUT3,B/256:OUT5,P:NEXTC,A:OUT0,0
```

Wo Pressefreiheit herrscht und jedermann lesen kann, da ist Sicherheit.
Thomas Jefferson


```
1 'HRG - Grafik vom Basic aus    == D E M O P R O G R A M M ==
2 '                                Anwendungen von Jens NEUEDER
3 'CMD"clh/cmd"
5 CLS:CLEAR1000:DEFINTA-Z:OUT1,0
9 '======== Grafik festlegen ==========================
10 '--------- Horixontale Linie ------------------------
11 Y=100:FORX=0TO383:GOSUB101:NEXT
12 '--------- Bewegte horizontale Linie ----------------
13 Y=110:FORL=1TO5:FORX=0TO20:GOSUB101:NEXTX:FORX=0TO20STEP2:
   GOSUB102:NEXTX:FORX=0TO20:GOSUB102:NEXTX,L
14 '--------- Laufende horizontale Linie ---------------
15 Y=120:FORL=0TO20:FORX=LTOL+10:GOSUB101:NEXTX:X=L:GOSUB102:
   NEXTL
80 '--------- Rahmen -----------------------------------
81 Y=10:FORX=10TO60:GOSUB101:NEXT
82 Y=20:FORX=10TO60:GOSUB101:NEXT
83 X=10:FORY=10TO20:GOSUB101:NEXT
84 X=60:FORY=10TO20:GOSUB101:NEXT
90 '--------- Fläche -----------------------------------
91 FORX=150TO175:FORY=50TO80:GOSUB101:NEXTY,X
97 GOTO122:'  Sprung zur Löschroutine
98 '===================================================
99 '          Grundprogramm von Arnulf SOPP
100 '======= Grafikmodus ===============================
101 GOSUB111:OUT5,AORC:RETURN                  'setzen
102 GOSUB111:OUT5,AAND(NOTC):RETURN            'löschen
103 GOSUB111:D=((AANDC)<>0%):RETURN            'invertieren
110 '======= Grafikpunkt errechnen ====================
111 X1=INT(X/6%):B=X-X1*6%:Y1=INT(Y/12%):Y2=Y-Y1*12%:
    Z=Y2*1024%+Y1*64%+X1:ZM=Z/256%:ZL=ZAND255%:OUT2,ZL:OUT3,ZM:
    C=2%äB:A=INP(4):RETURN
120 '======= Grafik löschen ===========================
121 CLS:FORI=0TO2FFF:OUT2,IAND255:OUT3,I/256:OUT5,63:NEXT
122 Z$=INKEY$:IFZ$=""THEN122ELSECMD"clh/CMD":END
123 'CLH/CMD ist ein File aus HRG-PACK.
    Es kann auch ein eigenes File erstellt werden, wie im INFO
    unter "Den HRG-Speicher löschen" beschrieben ist.
```

```
100 CLEAR 5000: DEFINT U: DIM EE(15), EA(15), E$(15), F$(20): GOSUB 2100
0
500 REM -----------------
510 REM *** HAUPTPROGRAMM
520 REM -----------------
530 CLS: PRINT TAB(11) "C A S S E T T E N - V E R Z E I C H N I S"
540 PRINT STRING$(64,"+")
550 PRINT @ 267, "W a h l m   g l i c h k e i t e n :"
560 PRINT @ 395, "<1>    Eingeben bzw. editieren
570 PRINT @ 459, "<2>    Abspeichern"
580 PRINT @ 523, "<3>    Laden"
590 PRINT @ 587, "<4>    Drucken"
595 PRINT @ 651, "<5>    Programmende"
600 A$=INKEY$: IF A$<"1" OR A$>"5" THEN 600
610 IF A$="1" THEN GOSUB 1000: GOSUB 2000: GOTO 500
620 IF A$="2" THEN PRINT @ 832, "Dateiname :    ";: INPUT D$: GOSUB 3000:
 GOTO 500
630 IF A$="3" THEN PRINT @ 832, "Dateiname :    ";: INPUT D$: GOSUB 4000:
 GOTO 500
640 IF A$="4" THEN GOSUB 5000: GOTO 500
650 CLS: PRINT "P r o g r a m m e n d e": PRINT @ 832, "";: END
1000 REM ------------------------
1010 REM *** EINGABEMASKE 1 AUSGEBEN
1020 REM ------------------------
1025 EE=50030: GOSUB 23000
1030 CLS: FOR I=0 TO 14: PRINT USING "##  ";I;: IF F$(I)<>"" THEN PRINT
F$(I) ELSE PRINT STRING$(57,"#")
1035 NEXT I
1040 GOSUB 20000
1050 FOR I=0 TO 14: F$(I)=E$(I): NEXT I
1060 RETURN
2000 REM ------------------------
2010 REM *** EINGABEMASKE 2 AUSGEBEN
2020 REM ------------------------
2025 EE=50040: GOSUB 23000
2030 CLS: PRINT TAB(12) "E I N G A B E   R   C K E N T I T E L"
2040 PRINT @ 256, "1)    ";: IF F$(15)="" THEN PRINT STRING$(57,"#") ELSE
 PRINT F$(15)
2050 PRINT "2)    ";: IF F$(16)="" THEN PRINT STRING$(57,"#") ELSE PRINT
F$(16)
2055 PRINT: PRINT
2060 PRINT "Datum          :    ";: IF F$(17)="" THEN PRINT "##.##.1985" ELS
E PRINT F$(17)
2070 PRINT "Laufzeit       :    ";: IF F$(18)="" THEN PRINT "C 90" ELSE PRIN
T F$(18)
2080 PRINT "Mono/Stereo :    ";: IF F$(19)="" THEN PRINT "S" ELSE PRINT F
$(19)
2090 PRINT "Dolby (J/N) :    ";: IF F$(20)="" THEN PRINT "J" ELSE PRINT F
$(20)
2100 GOSUB 20000
2110 FOR I=15 TO 20: F$(I)=E$(I-15): NEXT I
2120 RETURN
3000 REM -------------------------------
3010 REM *** CASSETTEN-VERZEICHNIS ABSPEICHERN
3020 REM -------------------------------
3030 OPEN "O", 1, D$
3040 FOR I=0 TO 20: PRINT #1, F$(I): NEXT I
3050 CLOSE 1
3060 RETURN
4000 REM -------------------------------
4010 REM *** CASSETTEN-VER[illegible]ICHNIS LADEN
4020 REM -------------------------------
4030 OPEN "I", 1, D$
4040 FOR I=0 TO 20: INPUT #1, F$(I): NEXT I
4050 CLOSE 1
4060 RETURN
5000 REM -------------------------------
5010 REM *** CASSETTEN-VERZEICHNIS DRUCKEN
5020 REM -------------------------------
5030 LPRINT CHR$(27); CHR$(14); CHR$(27); CHR$(28)
5040 LPRINT CHR$(240); STRING$(59,241); CHR$(242)
5050 FOR I=0 TO 14: LPRINT CHR$(245); TAB(60) CHR$(245): LPRINT CHR$(245
); " "; F$(I); TAB(60) CHR$(245): NEXT I
5060 LPRINT CHR$(245); TAB(60) CHR$(245): LPRINT CHR$(244); STRING$(59,2
41); CHR$(249): LPRINT CHR$(245); TAB(60) CHR$(245)
5070 LPRINT CHR$(245); " "; F$(15); TAB(60) CHR$(245): LPRINT CHR$(245);
 TAB(60) CHR$(245)
5080 LPRINT CHR$(245); " "; F$(16); TAB(60) CHR$(245): LPRINT CHR$(245);
 TAB(60) CHR$(245)
5090 LPRINT CHR$(244); STRING$(59,241); CHR$(249): LPRINT CHR$(245); TAB
(60) CHR$(245)
5100 LPRINT CHR$(245); " Datum :    "; F$(17); TAB(30) "Cassette   "; F$(
18); TAB(60) CHR$(245): LPRINT CHR$(245); TAB(60) CHR$(245): LPRINT CHR$
(245); TAB(60); CHR$(245)
5110 LPRINT CHR$(245);: IF F$(19)="M" THEN PRINT " Mono"; ELSE LPRINT "
Stereo";
5120 LPRINT TAB(30) "Dolby :      "; F$(20); TAB(60) CHR$(245): LPRINT CH
R$(245); TAB(60) CHR$(245): LPRINT CHR$(246); STRING$(59,241); CHR$(247)
5130 RETURN


23030 REST EE: F=0


50030 DATA 4,61,68,125,132,189,196,253,260,317,324,381,388,445,452,509,5
16,573,580,637,644,701,708,765,772,829,836,893,900,957,0,0
50040 DATA 261,318,325,382,528,538,592,596,656,657,720,721,0,0
```

M A N F R E D   M A N N ' S   E A R T H   B A N D

1) S o m e w h e r e   I n   A f r i c a

Tribal Statistics / Eyes Of Nostradamus / Third World Service / Demolition Man/ Brothers And Sisters Of Azania/ Africa Suite (Part a-d: Brothers And Sisters Of Africa; To Bantustan?; Koze Kobenini?; Lalela) / Redemption Song (No Kwazulu)/ Somewhere In Africa

2) G l o r i f i e d   M a g n i f i e d

Meat/ Look Around/ One Way Glass/ I'm Gonna Have You All/ Down Home / Our Friend George / Ashes To The Wind / Wind/ It's All Over Now Baby Blue/ Glorified Magnified

M A N F R E D   M A N N ' S   E A R T H   B A N D
Somewhere In Africa --- Glorified Magnified

Datum : 30.10.1985 Cassette C 90
Dolby : J

CASSETTEN - VERZEICHNIS

Das Schreiben derselben habe ich mir heute durch den Comp. vereinfacht (Ergebnis siehe links). Ich habe den im letzten Info vorgest. Editor geladen und dann in ca. 3 Stunden dieses Programm geschrieben. Im Bereich des Editors gab es dabei folgende Änderung: Ich benötige 2 Eingabemasken und daher 2 verschiedene DATA-Werte für den Editor; deshalb in Zeile 23030 der REST-Befehl (= RESTORE N; u.a. gibt es in der Bibliothek ein Programm dafür). Die Zeilen 20000-50020 siehe Info 9, Seite 7). Die Druckerroutine ist auf meinen LP VI angepaßt und muß ev. abgeändert werden.

Günther Wagner

ACHTUNG: EDITOR wurde verbessert; Bericht in diesem Info

NACHTRAG ZUM EDITOR

(von Günther Wagner)

Der im letzten Info vorgestellte Editor wurde jetzt nochmals verbessert - hier der aktuelle Bericht darüber.

Das größte Manko des Editors war bisher die lange Einlesezeit vom Bildschirm in die Variable (siehe Punkt J. Zeile 20260-20320 des letzten Artikels). Diese Einleseroutine habe ich nun umgeschrieben und dabei ganz einfache Techniken verwendet. Auf die Idee brachte mich übrigens der Hartmut; die Lösung wurde mir mit dem Buch Basic Faster & Better erleichtert.

Zunächst machen wir einen Ausflug in den Speicher unseres Computers - speziell in den String-Speicher. Wenn wir VARPTR (String) ausführen, so erhalten wir einen sehr wertvollen Adresswert. Die Adresse gibt uns die entsprechende String-Länge an. Die Adresse um 1 erhöht ergibt das LSB der Startadresse des Stringwertes, die Adresse um 2 erhöht das MSB. Aus LSB und MSB können wir uns die Adresse errechnen, an der die Abspeicherung des Strings beginnt:

Speicherplatz = MSB * 256 + LSB

Zum Beispiel:	VARPTR (AN$)	ergibt	29000
	PEEK (29000)	ergibt	113
	PEEK (29001)	ergibt	205
	PEEK (29002)	ergibt	48

Wir wissen also jetzt, das der String AN$ eine Länge von 113 hat und die Abspeicherung bei Adresse 48 * 256 + 205 = 12493 beginnt.

Nun können wir natürlich auch in die 3 Speicherstellen POKE'n - genau das machen wir auch. Wir wollen z.B. daß AN$ den Inhalt der 1. Bildschirmzeile erhält. In Anlehnung an obiges Beispiel genügen folgende Anweisungen:

```
VARPTR (AN$)      ergibt   29000
POKE 29000,64
POKE 29001,0
POKE 29001,60
```

Die 1. Bildschirmposition ist der Adressenwert 15360.

15360 : 256 = 60 Rest 0

d.h. unser neues MSB ist 60, LSB ist 0 und die String-Länge ist 64 (1 Bildschirm-Zeile).

Ich hoffe, daß ich mich bisher verständlich genug ausgedrückt habe. Nun - genau diese Methode wende ich nun auch für den EDITOR an.
In der Zeile 21270 wird der Anfang der Variable festgelegt und in PO% abgespeichert. Der Anfang der Variablen wird auf den Anfang des jeweiligen Eingabefeldes gelegt (EE()). In der Zeile 21270 wird auch die Länge der Variablen festgelegt. Diese ist genauso lang wie die Länge des Eingabefeldes und wird in A1% abgespeichert.
Nun folgt in der Zeile 21280 das VARPTR-Kommando und die beiden POKE-Kommandos. Beim 1. POKE-Kommando (MSB) wird 60 addiert, da in PO% die Bildschirmposition steht und diese ja um 15360 (= 60 * 256) erhöht werden muß, damit man die Adressposition erhält.
Nach der Zeile 21280 ist also der Inhalt von AN$ gleich dem Inhalt des jeweiligen Eingabefeldes (wir habe AN$ auf den Bildschirm "gelegt").
In der Zeile 21300 übergeben wir dann nur noch den Inhalt von AN$ in E$(), wobei die Zeile 21290 und der LEFT$-Befehl dafür sorgen, daß beim ersten '#' das Ende der Eingabe erkannt wird.

So - Ihr braucht jetzt also nur beim Listing des 9. Infos die Zeilen 20260-20300 wie folgt abändern:

```
20260 FOR G=0 TO F
20270 PO%=EA(G): A1%=EE(G)-EA(G)
20280 AN$=" ": POKE VARPTR(AN$),A1%: POKE VARPTR(AN$)+2,INT(PO%/256)+60:
 POKE VARPTR(AN$)+1,PO%-INT(PO%/256)*256
20290 GG=INSTR(AN$,"#")-1: IF GG<0 THEN GG=LEN(AN$)
20300 E$(G)=LEFT$(AN$,GG)
```

Yes

In der kritischen Phase des Vietnam-Krieges befragen die Generale im Pentagon ihren großen Computer, was sie jetzt tun sollten:
»Weiter vorgehen?«
»Zurückgehen?«
»Die Stellungen halten?«
Der Computer klickt eine Weile nachdenklich, dann antwortet er: »YES!«
Konsterniertes Schweigen der Generale. Dann tippt einer ärgerlich auf der Eingabe-Tastatur: »Yes what?«
Wieder klickt der Computer – erheblich länger diesmal – und dann schreibt er: »YES, SIR!«

Mini-RAM-Floppy im "sicheren Plätzchen"

Helmut Bernhardt stellt in c't 5/85 unter dem Titel "Ein sicheres Plätzchen" eine Schaltung vor, die im Adreßbereich 3900-3BFF RAM verfügbar macht (diese wie auch fast alle folgenden Zahlenangaben in Hex). Dort liegen ursprünglich die oberen nicht genutzten Adressen der Tastatur. Der Autor schlägt vor, dort Maschinenprogramme unterzubringen. Da alle gängigen Anwenderprogramme im Adreßraum ab 4300 (Level 2) bzw. 5200 (DOS) residieren, liegt es nahe, im neu gewonnenen RAM allgemeine Systemerweiterungen unterzubringen. Hier soll eine Methode vorgestellt werden, nach der NEWDOS-80 und seine Abkömmlinge G-DOS und H-DOS mit wertvollen zusätzlichen Möglichkeiten ausgestattet werden können.

Eine DOS-Anforderung, d. h. das Laden und Anspringen eines SYS-Files, wird über die RST-28-Routine abgewickelt. Dazu muß ein Code im Akku stehen, dessen binäre Bitkonfiguration darüber entscheidet, welche Systemdatei geladen und welche Routine innerhalb dieses Files angesprungen wird. Dieser Code muß mindestens 20 (hex, wohlgemerkt) betragen, andernfalls kehrt RST 28 unverrichteter Dinge zurück. In der Praxis kommt aber nur ein einziger Fall vor (abgesehen von Programmierfehlern), in dem der Requestcode kleiner als 20 ist: Wird im ROM-Tastaturtreiber die BREAK-Taste erkannt, dann wird RST 28 mit 01 im Akku angesprungen (und ohne Wirkung sofort wieder verlassen, wie gesagt).

Bei diesem Requestcode entscheiden die drei unteren Bits 0-2 darüber, in welchem Sektor des Inhaltsverzeichnisses der Systemdiskette die SYS-Datei zu suchen ist. Die Bits 3 und 4 geben an, welches der vier dort eingetragenen Systemfiles gemeint ist. BOOT/SYS (oder GDOS/SYS bzw. HDOS/SYS) wird im Prinzip nur nach dem Einschalten aufgerufen. Sein Requestcode entspräche dem Bitmuster xxx-00-000. Das bedeutet, daß im "nullten" Dateieintrags-Sektor die "nullte" Datei gemeint ist, also im Sektor 02 des Inhaltsverzeichnisses der 1. Eintrag. Ein Aufruf des Bootfiles mit RST 28 kommt nicht vor. Die acht möglichen Requestcodes mit dem Muster xxx00000 stehen deshalb für unsere Zwecke zur Verfügung.

Es wird noch wesentlich mehr: Die Bedingung, daß der Code im Akku mindestens 20 betragen muß, verringert die theoretisch möglichen 256 (diesmal dez) Codes glatt auf die Hälfte, was wir nicht hinnehmen müssen. Stattdessen kann man auf den BREAK-Code 01 testen und bei Übereinstimmung zurückspringen, um nicht bei jedem BREAK die Floppy in Gang zu setzen. Anschließend kann geprüft werden, ob der Requestcode höchstens 1F beträgt. Falls nein, handelt es sich um eine Anforderung a la Apparat. Inc. bzw. TCS. Dann geht es eben in der alten Routine im DOS-Kern weiter. Andernfalls ist es ein ehemals wirkungsloser Code bis 1F, mit dem der Anwender nun etwas anstellen kann.

Außerdem ist die Tatsache interessant, daß bei einem RST 28 alle Register zunächst unverändert in der Bearbeitungsroutine ankommen. So können beliebige Parameter an eine selbstgeschriebene DOS-Erweiterung übergeben werden. NEWDOS-80 (G-DOS, H-DOS) macht sich das zunutze, indem es dem Register C bei den meisten Library-Befehlen eine Zeigerfunktion zuordnet.

Um die oben skizzierten neuen Möglichkeiten auszunutzen, muß man wissen, wie RST 28 arbeitet. Der Einsprung ist natürlich bei 0028 in der "zero-page", der "Seite 0", also im Bereich der ersten 256 Bytes des ROM. Dort steht ein Vektor nach 400C, wo wiederum nach 4BC2 weiterverzweigt wird. An der Adresse 4BC2 wird der Stapelzeiger SP (stack pointer) zweimal inkrementiert. Die Wirkung ist, daß die RET-Adresse sozusagen vom Stack verschwindet. Dadurch verliert der RST 28 im Gegensatz zu den anderen RSTs seinen CALL-Charakter. Es wird ein gewöhnlicher JP daraus. Anders als bei einem Unterprogrammaufruf geht deshalb die Kontrolle endgültig an die angesprungene Routine über. Unter welchen Umständen sie dennoch mit einem RET verlassen werden kann, soll später erläutert werden.

Nach dem Quasi-Löschen der RET-Adresse folgt die oben angesprochene Prüfung auf 20. Wenn der Requestcode kleiner ist, geht das Carry-Flag auf 1 und es erfolgt ein Sprung nach 4312, von dort nach 4590, wo nur der Akku auf 00 gesetzt und aus der RST-28-Routine zurückgekehrt wird. Dieser bedingte Sprungbefehl JP C,4312h kann nun leicht durch einen Sprung in die eigene Routine im Bereich 3900-3BFF ersetzt werden (wer das "sichere Plätzchen" nicht hat, kann natürlich auch sonstwohin springen). Ein Teil dessen, was dort bei H-DOS passiert, geht aus dem Listing am Ende dieses Artikels hervor:

Im residenten Teil von SYS0 ist an der Adresse 4BC6 der bedingte Sprungbefehl durch einen JP rst28 ersetzt. Dieses Label steht für die Adresse 3A5B. Dort wird geprüft, ob die BREAK-Taste mit 01 im Akku den RST verursachte (s. o.). Bei Übereinstimmung wird der Sprung nach 4312 nachgeholt. Sonst wird getestet, ob eines der beiden Bitmuster 000xxxxx oder xxx00000 vorliegt. Falls ja, ist unser neues RAM-SYS-File zuständig und wird angesprungen. Andernfalls geht im DOS-Kern bei 4BC9 die Bearbeitung wie gewohnt weiter. Auf diese Weise sind 39 zusätzliche Requestcodes möglich. Wie aus dem Listing hervorgeht, wird davon bisher nur ein Teil genutzt. Zukünftige Erweiterungen werden diesen Vorrat nach und nach verkleinern.

Die Tatsache, daß der RST 28 wie ein JP behandelt wird, hat gute Gründe: Bei vielen Systemdateien ist ein Rücksprung zum Caller nicht sinnvoll, manchmal, etwa beim Auftreten eines I/O-Fehlers, u. U. sogar fatal. Gleichwohl kann eine SYS-Datei wie ein Unterprogramm aufgerufen werden. Dazu ist es lediglich nötig, den RST 28 nicht direkt zu programmieren, sondern stattdessen einen CALL an eine Adresse, wo ein RST 28 steht. Das ist z. B. bei 4402 der Fall oder auch im ROM bei 0456, wo BREAK den RST aufruft. Ein RST ist für den Z80 ein CALL. Wenn man nach dem angegebenen Muster einen CALL callt, befindet man sich deshalb bereits in der zweiten Unterprogrammebene. Das zweimalige Inkrementieren des Stackpointers am Beginn der RST-28-Routine geht eine Ebene höher, so daß bei einem RET nun die richtige Rückkehradresse gefunden wird.

So erklärt sich das RET in Zeile 164 des Listings. Mehr möchte ich zum gelisteten Teil des Programms nicht sagen, denn es geht hier nur um die Methode, RST 28 für eigene Anwendungen nutzbar zu machen. Die hier nicht interessierenden Teile des Programms sind durch LIST OFF ausgespart. Wer Interesse daran hat, kann einen großen Teil davon aus dem darüberstehenden Sektordump rekonstruieren. Das hier Gelistete ist darin unterstrichen. Es handelt sich um den relativen Sektor 10h von SYS0/SYS. Ursprünglich hat SYS0 nur 15 Sektoren, wurde aber für die Erweiterungen mit APPEND um weitere 5 Sektoren verlängert.

Arnulf Sopp

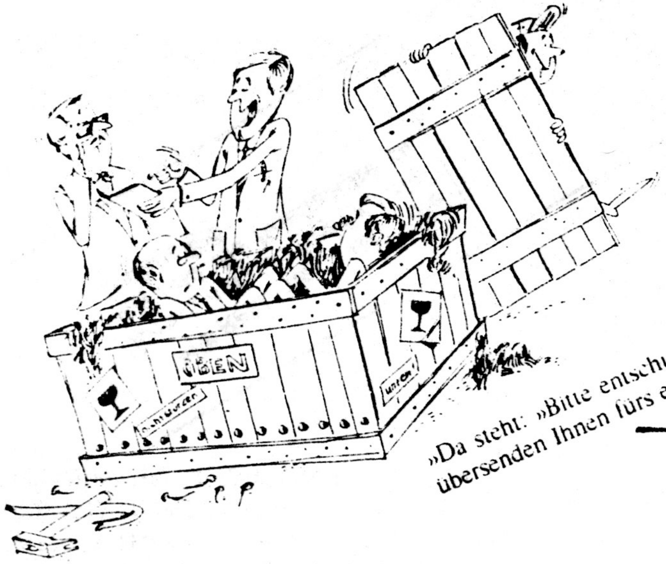
```
001000: 0102 003A FF10 FE41 3CD3 FF10 FE0D 4110 ...:...A<.....A.
001010: F0F1 C93A 5038 FE01 200A 3A40 38FE 1020 ...:.S.. .:§8..
001020: C33A 1038 E6FE 283C C31C 3BCD 5444 280C .:.8..(...:.TD(.
001030: 7EE6 DFFE 4ACA E73A FE4E 2804 3E2F 185E B...J..:.N(.>/.^
001040: 2153 4822 0845 3E3A 32D3 4521 1038 22D4 !SH".E>:2.E!.8".
001050: 453E DA32 C64B 2112 4322 C74B AF18 40FE E>.2.K!.C".K..§.
001060: 01CA 1243 F5E6 1F28 0AF1 F5E6 E028 04F1 ...C...(.....(..
001070: C3C9 4BF1 FEE0 2007 3E01 D3F0 3EFD EFEE ..K... .>...>...
001080: D5C5 FE60 CA06 3BF3 FE40 2838 FE80 285B ...`..:..§(8..(A
001090: FEA0 2897 FEC0 28A8 FE0F 3008 3E37 B7C1 ..(...(...0.>7..
0010A0: D1E1 FBC9 CB67 2004 0EF0 1807 0EDF 3E0A .....g ........>.
0010B0: ED79 3CED 7921 0030 7E2F 77BE 2F77 ED78 .y<.y!.0B/w./w.x
0010C0: 3E08 18DB 3E08 D3DF 3CD3 DF3C 3CD3 DF21 >...>...<..<<..!
0010D0: 0000 545D 7E2F 77BE 2804 3E08 18C0 AFD3 ..TUB/w.(.>.....
0010E0: DF01 E037 EDB0 DBDF AF18 B421 F439 2208 ...7.......!.9".
0010F0: 453E CD32 D345 210F 3A22 D445 3EC3 32C6 E>.2.E!.:".E>.2.
```

```
4BC6            00001          ORG     4bc6h          ;RST-28h-Routine
4BC6  C35B3A    00002          JP      rst28          ;umleiten
                00003
                00114 ;Ansprung DOS-Request (RST 28h):
3A5B  FE01      00115 rst28    CP      01h            ;RST 28h nach BREAK?
3A5D  CA1243    00116          JP      Z,4312h        ;falls ja
3A60  F5        00117          PUSH    AF             ;Requestcode retten
3A61  E61F      00118          AND     1fh            ;Requestbits isolieren
3A63  280A      00119          JR      Z,ramsys       ;falls Code xxx00000b
3A65  F1        00120          POP     AF             ;sonst Requestcode rest.
3A66  F5        00121          PUSH    AF             ;und wieder retten
3A67  E6E0      00122          AND     0e0h           ;Requestbits ausmaskieren
3A69  2804      00123          JR      Z,ramsys       ;falls Code 000xxxxxb
3A6B  F1        00124          POP     AF             ;sonst Code restaurieren
3A6C  C3C94B    00125          JP      4bc9h          ;und RST 28h à la G-DOS
                00126
                00127 ;Ansprung hier, falls RAM-SYS-File zuständig:
3A6F  F1        00128 ramsys   POP     AF             ;Requestcode restaurieren
3A70  FEE0      00129          CP      0e0h           ;V24?
3A72  2007      00130          JR      NZ.nov24       ;falls nein
                00131
3A7B  E5        00140 nov24    PUSH    HL             ;Register retten
3A7C  D5        00141          PUSH    DE
3A7D  C5        00142          PUSH    BC
3A7E  FE60      00143          CP      60h            ;HRG-Speicher löschen?
3A80  CA063B    00144          JP      Z.hrgcls       ;falls ja
3A83  F3        00145          DI                     ;bloß keine Störungen!
3A84  FE40      00146          CP      40h            ;ROM -> RAM kopieren?
3A86  283E      00147          JR      Z.copy         ;falls ja
3A88  FE80      00148          CP      80h            ;INI,J?
3A8A  285B      00149          JR      Z,inij         ;falls ja
3A8C  FEA0      00150          CP      0a0h           ;INI?
3A8E  2897      00151          JR      Z,ini          ;falls ja
3A90  FEC0      00152          CP      0c0h           ;INI,N?
3A92  28A8      00153          JR      Z.inin         ;falls ja
3A94  FE0F      00154          CP      0fh            ;auf Bank-RAM testen?
3A96  3008      00155          JR      NC.ramtest     ;falls ja
                00156
                00157 ;raus mit oder ohne Fehlercode:
3A98  3E37      00158          LD      A.37h          ;Code für DOS-Fehler
3A9A  B7        00159 error    OR      A              ;Fehlerflag NZ
3A9B  C1        00160 exit     POP     BC             ;Register restaurieren
3A9C  D1        00161          POP     DE
3A9D  E1        00162          POP     HL
3A9E  FB        00163          EI                     ;INTs [illegible]er zulassen
3A9F  C9        00164          RET                    ;und raus
```


Vergleichsliste: Genie-DOS <-----> NEWDOS80.2

Genie-DOS	NEWDOS80.2
Mini-Befehlseingabe:	**MINI-NEWDOS/80 READY**
M>	**MDCOPY**
;	**MDRET**
/	**MDBORT**
!	**---**
Befehlseingabe:	**NEWDOS/80 READY**
& (J,N)	**DEBUG (Y,N)**
? bzw. LIB	**LIB**
I bzw. DIR (I,S,A,B,/typ)	**DIR (I,S,A,U,/ext)**
> bzw. COPY (QPDN=,ZPDN=,KDWA, QKW=,NZKW=,ZZND,AZN=,QN=, BZN,NZN=,BZD,SQD,IVU,SBIV=, AEIV=,EDK,AZKW=,FRD,BEA,NVD, IDL=,XDL=,FRAG,---	**COPY (SPDN=,DPDN=,NDMW, SPW=,NDPW=,DDND, ODN=,SN=, KDN,NDN=,KDD,USD,BDU,DDSL=, DDGA=,CBF,ODPW=,USR,UPD,DFO, ILF=,XLF=,CFWO,UBB**
0 (VON=,BIS=,HIMEM=)	**CLEAR (START=,END=,MEM=)**
@ (KEINE,MO,TA,DR,ST=,NL)	**ROUTE (CLEAR,DO,KI,PR,MM=,NL)**
AIK	**WRDIRP**
ATTRIB (BKW=,HKW=,BEA=,PROT=, KEIN,START,LESEN,ÄNDERN, NAME,KILL,NULL)	**ATTRIB (ACC=,UPD=,UDF=,PROT=, LOCK,EXEC,READ,WRITE, (RE)NAME,KILL,FULL)**
B2	**BASIC2**
BL (J,N)	**BLINK (Y,N)**
CREATE (ANZ=,ADE=,ADF=,LOG=)	**CREATE (REC=,ASE=,ASC=,LRL=)**
CONT (J,N,D)	**CHNON (Y,N,D)**
DATUM (tt.mm.jj)	**DATE (MM/DD/YY)**
E	**ERROR**
LC	**LCDVR**
---	**LC**
N	**RENAME**
NDF (SPUR=,STOP=,MAG)	**FORMAT (PFST=,PFTC=,RWF)**
PD (SP=,SEK=,SWZ=,EIB=, SBIV=,AEIV=)	**PDRIVE (TC=,SPT=,TSR=,GPL=, DDSL=,DDGA=)**
PROT (DATUM=,BKL,ZU,AUF)	**PROT (DATE=,RUF,LOCK,UNLOCK)**
S	**SYSTEM**
UHR	**CLOCK**
V+ (J,N)	**VERIFY (Y,N)**
ZEIT (hh:mm:ss)	**TIME (hh:mm:ss)**

nur bei Genie-DOS:	nur bei NEWDOS80.2:
DISK	**FORMS**
DR	**SETCOM**
FORM	
INFO	
LF	
PORT	
V24	
Z	
F#	
##	
80	
64	
DDE	



»Da steht: »Bitte entschuldigen Sie die lange Lieferfrist. Wir übersenden Ihnen fürs erste zwei erstklassige Rechner!««

Umlaute im NEWDOS ?? – aber natürlich !!

Ihr könnt Euch sicher noch erinnern, wie ich im letzten Clubinfo nach einer Antwort für mein "UMLAUTPROBLEM MIT NEWDOS" -dies war die richtige Groß-/Kleinschreibung der Umlaute wie bei den Normalbuchstaben und die Akzeptanz von Umlauten in den Filenamen-suchte.

Nun, nach einigen kleinen Tips über die Tastatureingaberoutine, habe ich die Antwort darauf selbst gefunden. Aber erst einmal etwas "Vorgeschichtliches" zu diesem Thema.

Nach der Erkennung meines Problems probierte ich dann verschiedene DOS-Versionen aus. Dabei bemerkte ich, daß diese "FEHLER" im Genie-DOS nicht auftauchten, wohl aber im NEWDOS. So stellte ich fest, daß im NEWDOS die Umlautschreibung nicht berücksichtigt ist. Was wiederum verständlich ist, da NEWDOS, ein amerikanisches Programm ist, und die englisch Schreibweise keine Umlaute kennt.

Nun ist es leicht zu sagen, man arbeite mit dem Genie-DOS. Auch schon desshalb, weil G-DOS eine "(DEUTSCHE) NEWDOS-VERSION" ist. Andererseits aber, wäre es dann nötig, das DOS auf die entsprechenden Disketten umzukopieren, sowie auch die "DEUTSCHEN BEFEHLE" für das DOS sich anzueignen, wo man doch gerade die englischen erst richtig anwenden kann. Ich bin deshalb einen anderen Weg gegangen, den ich hier beschreiben möchte.

Da die beiden DOS -wie vorher schon erwähnt- bis auf die Deutschschreibung der Meldungen und Befehle ziemlich gleich sein müssen, liegt es nahe, beide zu vergleichen. Das sich nun dabei der Computer in's Zeug legen darf ist ja klar. Dazu habe ich ein kleines "FAULHEITSPROGRAMM" -wie ich es nenne, wenn der Computer für mich arbeitet-geschrieben.
Das Listing dazu ist am Ende dieses Berichtes angehängt und das Programm gibt es natürlich in der Clubbibliothek. Mein Programm fragt nun nach dem zu vergleichenden File, das in beiden Laufwerken zu Verfügung stehen sollte. Danach werden die entsprechenden File miteinander verglichen, wobei die darstellbaren Zeichen auf dem Bildschirm erscheinen. Bei Differenzen während des Vergleichs wird dann, wenn gewünscht, der Platz, das Zeichen sowie dessen Wert in Dez und Hex ausgedruckt.

Nun hatte ich es recht einfach. Ich ließ meinen Computer vergleichen und suchte dann in den Ausdrucken nach dem/der entsprechenden Zeichen/Zeichenfolge. Da ich die SYS-File der Reihe nach verglich, fand ich schon in SYS0/SYS den ersten Zap, welcher zu ändern wäre :

RELATIVER SEKTOR 2 BEI 8A : VON 1A AUF 1F

1F bedeuted das höchste abzuarbeitende Zeichen ( ). Vorher war hier das (Z) durch 1A implementiert.
Der Fehler mit der unrichtigen Abarbeitung der Umlaute war behoben. Jetzt steht noch die Umlautverwendung bei Filenamen aus. Dieses Hemmnis müsste in SYS2/SYS liegen, da hier die Filenamen auf Richtigkeit überprüft werden. Beim Vergleich fand ich nun Punkt 2 unser Änderung :

RELATIVER SEKTOR 4 BEI 63 : VON 5B AUF 5F

Auch hier wird der Wert für den maximalen Zahlenwert höher gesetzt. Das Abspeichern von Umlautfilenamen ist nun möglich. Es werden aber nur Großbuchstaben der Umlaute anerkannt. Dies nun zu beseitigen ist der nächste Schritt. Wieder zurück zu -wir hatten es ja schon-SYS0/SYS, wo die Eingaberoutine abgelegt ist. Es folgt Zap 3 :

RELATIVER SEKTOR 2 BEI AA : VON 7B AUF 7F

Jetzt werden auch "KLEINUMLAUTE" bei der Filenameneingabe akzeptiert.
In der Zwischenzeit sind aber im DOS auch einige Fehler aufgetaucht, die sich folgender Art auswirken. Befehle wie "KILL ....", "RENAME ...." oder andere, die auf die Programmnameneingabe angewiesen sind erzeugen bei der Umlauteingabe eine Fehlermeldung. Wir können also Umlautfilenamen im Moment nicht weiter bearbeiten bzw. mit Umlauten benamte Programme können nur durch "PURGE .." wieder gelöscht werden. Es werden also noch weitere Änderungen nötig sein.

Es wird weitergesucht und eine Fortsetzung dieses Artikels folgt.

Jens NEUEDER

„Der menschliche Verstand ist in der Praxis nicht verläßlich, am wenigsten in größter Not.“

„Der Kopf ist jener Teil unseres Körpers, der uns am häufigsten im Wege steht.“

FILEVERGLEICHSPROGRAMM

```
10 CLEAR1000:DEFINTA-Z:D=0:Y$="#######"
20 CLS:PRINT§320,CHR$(23);"Filevergleich"
30 PRINT§448,"von  Jens NEUEDER   Version 1.2"
40 PRINT§554,"18.10.1985"
50 PRINT§576,"Panoramastraße 21":PRINT"7178 Michelbach /Bilz":PRINT"Tel.: 0791/42877"
60 PRINT§832,STRING$(32,CHR$(127)):PRINT§896,"Bitte Vergleichfilename eingeben":INPUTA$
70 CLS:PRINTA$:PRINT"Eingabe richtig ?          (Ja/Nein)"
80 Z$=INKEY$:IFZ$=""THEN80ELSEIFZ$="N"THEN60ELSEIFZ$="n"THEN60
90 PRINT:PRINT"Ausdruck erwünscht ?       (Ja/Nein)"
100 Z$=INKEY$:IFZ$=""THEN100ELSEIFZ$="N"THEN110ELSEIFZ$="n"THEN110ELSED=1:
    LPRINTCHR$(14);:LPRINTCHR$(27);"!";:LPRINTA$;"  -File werden verglichen :":
    LPRINTCHR$(27);CHR$(34);:LPRINTCHR$(15)
110 PRINT:PRINT"Bitte die File in beiden Laufwerken bereithalten !!":PRINT"     ---> < ENTER >"
120 Z$=INKEY$:IFZ$<>CHR$(13)THEN120ELSE140
130 (1)B$;
140 CLS:PRINT"Die File => ";A$;" <= werden miteinander verglichen"
150 OPEN"R",1,A$+":0","FF",1:PRINT"File in 0 ";
160 OPEN"R",2,A$+":1","FF",1:PRINT"File in 1 eröffnet."
170 PRINT§200,". laufendes Byte":PRINT§264,". Sektor":PRINT§328,". relatives Byte";:
    PRINT§448,"File in 0 :";:PRINT§576,"File in 1 :";
180 A=1:Q=-1
190 GET1,A:GET1,,130:V1=ASC(B$)
200 GET2,A:GET2,,130:V2=ASC(B$)
210 S=INT(A/256):R=INT(A-(SX256))
220 PRINT§192,USINGY$;A;:PRINT§256,USINGY$;S;:PRINT§320,USINGY$;R;
230 IFV1>31THEN240ELSEI$=I$+" ":GOTO250
240 IFV1<192THENI$=I$+CHR$(V1)ELSEI$=I$+" "
250 IFV2>31THEN260ELSEJ$=J$+" ":GOTO270
260 IFV2<192THENJ$=J$+CHR$(V2)ELSEJ$=J$+" "
270 I$=RIGHT$(I$,64):J$=RIGHT$(J$,64)
280 PRINT§512,I$;:PRINT§640,J$;
290 IFV1=V2THEN360ELSEIFD=0THEN360
300 IFQ<>STHENLPRINT:LPRINT"Sektor :";S;" ab relativem Byte :";R
310 Q=S:H=R:GOSUB430:R$=H$:H=V1:GOSUB430:V1$=H$:H=V2:GOSUB430:V2$=H$
320 LPRINTR$;"->  1: (";V1;")--(";V1$;".hex)--(";
330 IFV1>32THENLPRINTCHR$(V1);")",ELSELPRINT")",
340 LPRINT"in ->  2: (";V2;")--(";V2$;".hex)--(";
350 IFV2>32THENLPRINTCHR$(V2);")"ELSELPRINT")"
360 A=A+1
370 IFNOTEOF(1)THEN380ELSEPRINT§448,"File 1 zu Ende.";:CLOSE1:IFD=1THENLPRINT:LPRINT"Ende File 1";
380 IFNOTEOF(2)THEN190ELSEPRINT§576,"File 2 zu Ende.";:CLOSE2:IFD=1THENLPRINT" File 2."
390 PRINT§768,"Vergleich abgeschlossen.":A=A-1:IFD=1THENLPRINT:
    LPRINT"Fileende bei :";A;" Bytes  -> ";S;" Sektoren"
400 PRINT"<P>rogrammende, <w>eiteres File ?   (P/W)"
410 Z$=INKEY$:
    IFZ$="W"THENRUNELSEIFZ$="w"THENRUNELSEIFZ$="P"THEN420ELSEIFZ$="p"THEN420ELSE410
420 END:'cmd"s"
430 H$="":X=H-INT(H/256):GOSUB440:RETURN
440 I=XAND15:J=(XAND240)/16:IFJ>9THENJ=J+55:GOTO450ELSEJ=J+48
450 IFI>9THENI=I+55:GOTO460ELSEI=I+48
460 H$=CHR$(J)+CHR$(I):RETURN
```

„Weiß ich nicht. Ich beschwere mein Gedächtnis nicht mit Tatsachen, die ich in einem Konversationslexikon finden kann."

„Reklame ist die Kunst, auf den Kopf zu zielen und die Brieftasche zu treffen"

Umbau eines TRS-80 Modell 1 16K auf 48k (64k)

Nachdem ich im Sommer des letzten Jahres einen gebrauchten TRS-80 Modell 1 gekauft hatte, der mit 16 k RAM bestueckt war, stand ich vor dem Problem diesen Rechner auf 48 k RAM aufzuruesten.
Zu diesem Zeitpunkt hatte ich schon einen Genie-I mit 48-k (64-k) RAM, und ich dachte das es kein groesseres Problem sein sollte auch den Tandy auf die gewuenschte Speicherkapazitaet zu bringen. Aber erstens kommt es anders, und zweitens als man denkt.
Da ich die technischen Handbuecher fuer beide Geraete besitzte verglich ich erst einmal die Schaltunterlagen von beiden Rechnern. Auf den ersten Blick sahen auch beide Schaltplaene sehr aehnlich aus, aber gerade beim interessanten Teil, naemlich dem Adressdekoder fuer die RAM's und ROM's unterschieden sich beide Geraete erhaeblich. So musste ich feststellen das der Genie-Computer von seiner Grundauslegung her schon fuer den Umbau von 16-k auf 48-k mit den neueren 64-k-RAM Speicherchips ausgelegt war, waehrend der TRS-80 einen voellig anderen Adressdekoder hatte, der sich auf den ersten und zweiten Blick ueberhaupt nicht fuer diesen Umbau eignete.
Es musste also ein mehr oder minder neuer Adressdekoder eingebaut werden.
Ich hatte zuerst vor, den Dekoder des Genie-I in den TRS-80 einzubauen. Aber auch dies erwies sich als nicht so einfach machbar, da sich beide Rechner in der Ansteuerung der ROM's und des Bildspeichers stark unterschieden.
Also was tun ?
Ich entschied mich dazu die ROM und Bildspeicheradressierung so zu lassen wie sie war und nur die Ansteurung der RAM's zu aendern.
Zuerst zum Adressdekoder Z21. Dieser Baustein zerlegt die unteren 32K, die die ROM's, Bildspeicher, Tastatur und die alten 16 K RAM enthalten in Bloecke zu je 4K. Die Ansteurung der alten RAM's konnte ja nun wegfallen, also machte ich folgende Ueberlegung, die neue RAM's sollen nicht angesprochen werden, wenn auf die unteren 16K zugegriffen wird, bei einer Adresse oberhalb von 3FFFH muss der RAM Bereich angesprochen werden.
Dies wird nun so erreicht: Wenn einer der ersten 4 Ausgaenge von Z21 LOW wird, so bedeutet dies, das auf die unteren 16K zugegriffen wird, also darf in diesem Fall der RAM Bereich nicht angesprochen werden.
Ich nahm also ein 4-fach-NAND und legte die Eingaenge dieses Chips auf die 4 Ausgaenge des Z21 (Pin 9 - 12), den Ausgang des NAND legte ich auf X3 Pin 15 von wo aus die RAM-Select Leitung abgeht (RAM *). Ist diese Leitung LOW, so werden die RAM-Bausteine selectiert, und zwar geht das Signal ueber X71 Pin 12 - Pin 5 auf Z67 Pin 15, den Select-Eingang eines Buffers ueber den MCAS geschaltet wird.
Somit war das Ziel der Adressierung schon erreicht. Wird eine Aresse der ROM's des Bildspeichers oder der Tastatur (0000H - 3FFFH) angesprochen, so wird der RAM-Bereich nicht selectiert, wird aber eine Adresse ueber 3FFFH (4000H - FFFFH) angesprochen so wird der RAM-Bereich selectiert.

Nun aber zum praktischen Teil des Umbau's. Als erstes mussten die alten RAM's (4116) gegen die neuen RAM's (4164) ausgetauscht werden.
Zuerst musste die Spannungsversorgung der RAM's geaendert werden.

Tabelle 1:

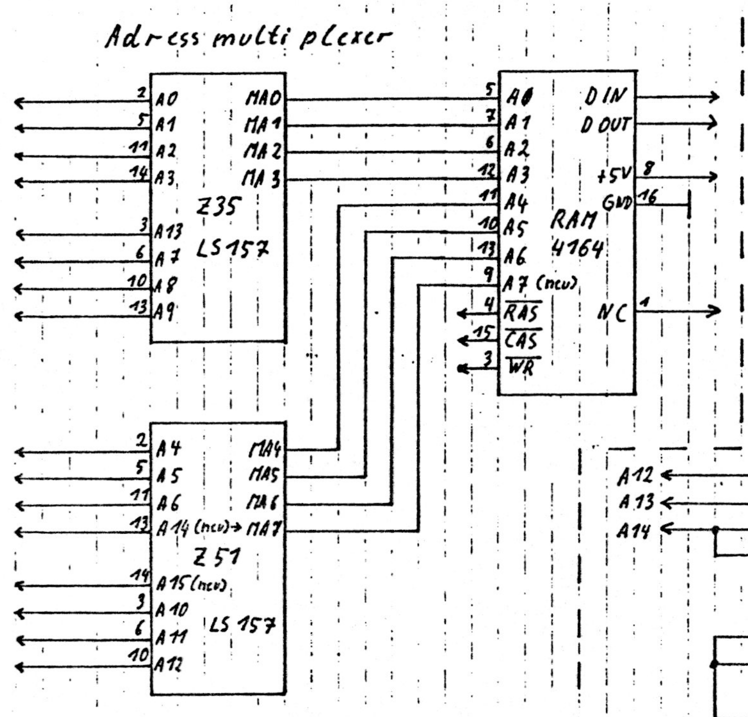
4116 -> :	Pin 1	--	-5 V
	Pin 8	--	+12 V
	Pin 9	--	+5 V
	Pin 16	--	GND
4164 -> :	Pin 1	--	N.C.
	Pin 8	--	+5 V
	Pin 9	--	A 7
	Pin 16	--	GND

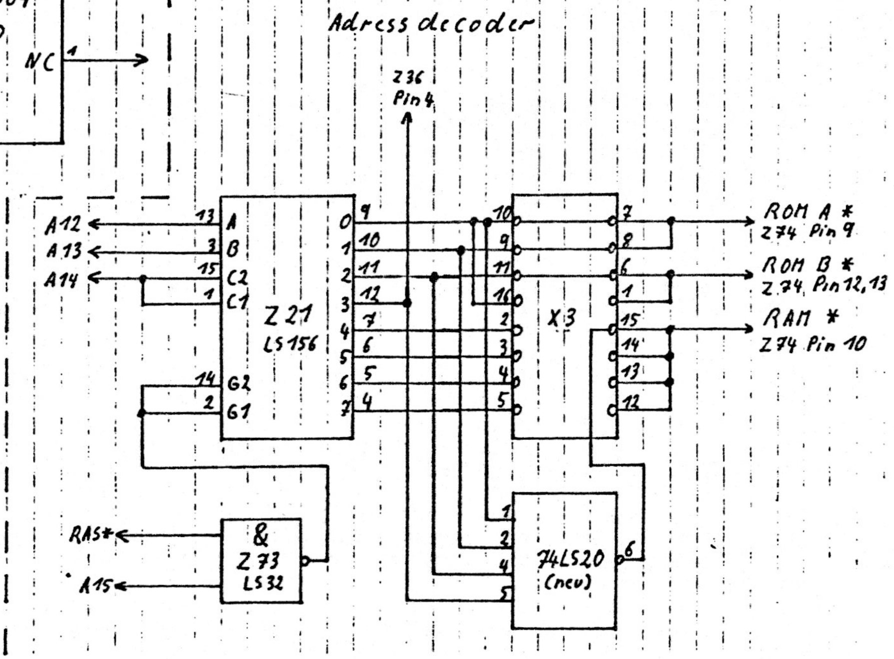
Wie aus dieser Tabelle zu ersehen muss der Pin 1 freigemacht werden. Auf der Platine sind die einzelnen Pins der 8 Speicher IC's durch Leiterbahnen miteinander verbunden. Also wurde festgestellt von welchem IC aus die Leiterbahn Als naechstes musste aus dem Stromversorgungsanschluss Pin 9 der Adresseingang A 7 gemacht werden. Dies geschah aehnlich dem Umbau von Pin 1. Zuerst stellte ich fest, von wo aus die Versorgungsspannung aus dem Netzteil kam, dann trennte ich diese Leiterbahn auf, entfernte noch die Siebkondensatoren (100nF) die auf dieser Leiterbahn gegen Masse lagen und dann wurde der Pin 9 der Speicher IC's mit dem Adressmultiplexer Z51 Pin 2 verbunden. Damit der Adressmultiplexer auch die Adresse A 7 erzeugen kann muss er auch etwas modifiziert werden. Es muessen die Adressleitungen A 14 und A 15 auf die Pins 14 und 15 gelegt werden. Diese drei Anschluesse sind normalerweise nicht belegt.
Als letztes musste noch der Pin 8 der Speicher-IC's von der +12 V Versorgung getrennt, und mit der +5 V Versorgung verbunden werden. Dies war fast der schlimste Teil des ganzen Umbaus, da an jedem einzelnen der 8 Speicher-IC's der Pin 8 von den Leiterbahnen getrennt werden musste und die neuen Anschluesse mit Schaltlitze auf der Platine nachverdrahtet werden musste.
Als letztes wurden dann die neuen Speicher-Bausteine in die Fassungen eingesetzt. Allerdings ist zu empfehlen, vor dem Einbau der IC's den Rechner schon einmal einzuschalten und die Spannungen an den IC-Sockeln zu pruefen, ob auch alle Aenderungen korrekt sind.
Dann koennen die neuen Speicherbausteine getestet werden. Also Rechner einschalten. Es sollte jetzt wie gewohnt 'MEM-SIZE' auf dem Bildschirm erscheinen. Nach dem Druecken der 'ENTER'-Taste dauert der Speicherausbautest jetzt etwa dreimal solange wie bisher. Also keine Panik wenn es jetzt etwas laenger dauert bis 'READY' erscheint. Nachdem der Rechner sich mit 'READY' gemeldet hat, sollte man 'PRINT MEM' eingeben und die 'ENTER'-Taste druecken. Wenn alles richtig geklappt hat, sollte jetzt die Zahl '48340' auf dem Bildschirm erscheinen.

Jetzt noch ein paar Tips zu den Speicher-Bausteinen selbst :
Bei meinem ersten Umbau klappte alles wunderbar, aber nach ein paar Tagen als ich ein ziemlich grosses Programm im Speicher hatte fing mein Rechner ploetzlich an zu Spinnen. Einer der auftretenden Effekte war ein sporadisch auftredender Kaltstart ab Aresse 0000H ('MEM-SIZE'), ein weiterer Fehler war, dass sich der Rechner irgendwo im Programm aufhaengte und nur durch ausschalten und wiedereinschalten erneut zum Arbeiten zu bewegen war.
Diese und aehnliche Effekte traten aber nur auf, wenn die 'MEM-SIZE' Eingabe ueber 32767 lagen, also im neu dazugewonnenen Speicherbereich lag.

»Der Verstand ist unser größtes Vermögen, aber Armut schändet nicht.«

»Bildung ist die Fähigkeit, sich alles anzuhören und dabei weder die Selbstbeherrschung zu verlieren noch das Selbstvertrauen.«





Also fiel der Verdacht auf die neuen Speicher IC's. Da ich diese IC's mit einigen Arbeitskollegen in einer Sammelbestellung bezogen hatte, erkundigte ich mich beide diesen, ob auch sie Probleme mit den Speichern haetten.
Und siehe da, einige hatten Probleme und andere nicht. Ein Vergleich der Hardware der einzelnen Rechner brachte uns dann auf den REFRESH. Und zwar wurden in einigen Rechnern die Speicher von der CPU aus REFRESH'd und in einigen anderen Rechner wurde dies von speziellen REFRESH-Controllern erledigt. Wir fanden also heraus, das alle Rechner, die den CPU-REFRESH benutzten Probleme mit den Speicher IC's hatten. Nach dem Durchsehen von diversen Datenblaettern fanden wir schliesslich heraus wo der Wurm zu suchen war.
Die Z80-CPU erzeugt einen 7 Bit REFRESH. Nun gibt es allerdings einige Hersteller die einen 8 Bit REFRESH fuer ihre 64K dymamischen RAM's brauchen Wenn man nun diese 8 Bit REFRESH IC's direkt an den Z80 anschliesst dann werden die Adressen 8000H -> FFFFH nicht refrescht, dass bedeutet das die Informationen, die auf diese Speicherplaetze geschrieben werden nach einer gewissen Zeit verschwinden.
Also Leute, aufpassen beim Kauf von 4164 IC's, fuer den TRS-80, und den Genie I muessen es IC's mit 7 Bit REFRESH sein.

Bernd Retzlaff

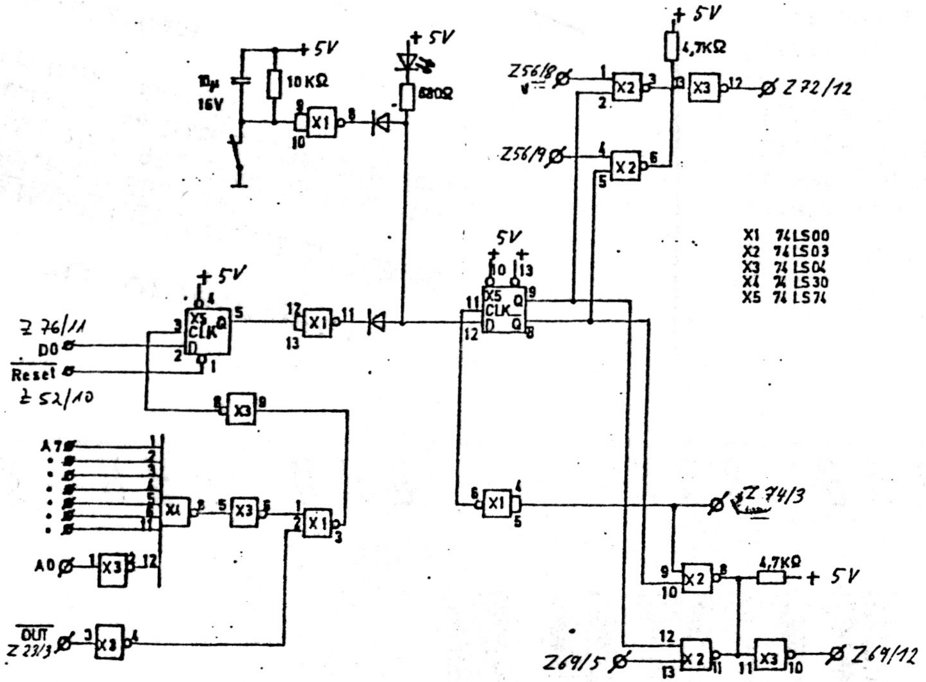
Computerurteil

Ein Mann besitzt zwei Uhren, die beide defekt sind. Die eine geht grundsätzlich 10 Minuten vor, das Laufwerk der anderen ist kaputt. Der Mann beschließt, das Elektronengehirn zu befragen.
Frage an den Computer:
»Welche Uhr soll ich behalten?«
Antwort des Computers: »DIE KAPUTTE UHR!«
Rückfrage: »Weshalb?«
Antwort des Computers: »DIE ERSTE UHR GEHT STÄNDIG FALSCH. DIE KAPUTTE UHR ZEIGT ZWEIMAL TÄGLICH DIE RICHTIGE ZEIT AN.«

****
*   Ergaenzung zur synchronisierten Taktumschaltung   *
****

Ergaenzend zu den Artikeln ueber die synchromisierte Taktumschaltung in den Clubinfo's 8 und 9 moechte ich hier eine Aenderung vorstellen in der die Anschlusspunkte fuer den TRS-80 Modell 1 angegeben sind. Weiterhin moechte ich noch hinzufuegen das die Leiterbahnen zwischen Z 56/8 und Z 72/12 und zwischen Z 69/5 und Z 69/12 zu trennen sind.

Bernd Retzlaff



BÖRSE

WER HAT WAS -- WER SUCHT WAS --

Suche Handbuch für den
GRAPE - Grafiktreiber

Dieter Weiß

ZU VERKAUFEN !!

Genie Modell 1 im TRS-80 M III Gehäuse
mit folgenden Einbauten:
Monitor, HRG, RS 232, 80-Zeichenkarte,
2 Laufwerke 40 Track DS/DD, CPM-fähig
mit externem Netzteil

Preis auf Anfrage

Gerald DREYER

ZU VERKAUFEN !!

Exatron Stringy-Floppy für TRS-80 M I

Mit 45 Waver in verschiedener Länge
(teilweise bespielt) mit 3 Registern
und zugehörigen Taschen

Preis 350,--DM

Jens NEUEDER

Mein EG 64 MBA ist zu haben! Noch immer schwärme ich von ihm, es fehlt ihm auch nicht das geringste, aber ein Kumpel hat einen entwickelt, der noch mehr kann. Wer so gerne wie ich im Computer herumlötet, mag auf die Veröffentlichung der Schaltung in unserer Clubpostille warten. Wer aber lieber die Finger davon läßt, kann meinen für DM 100,- (VB) kriegen (Neupreis DM 195,-, neuerdings soll er wohl DM 150,- kosten). An meinem hängt eine zusätzliche I/O-Platine dran, die alle Leitungen (inkl. IORD und IOWR) enthält, die man für portgesteuerte Peripherie braucht. Es lassen sich damit gleichzeitig ca. 30 Geräte über lauter verschiedene Ports ansteuern. Die Platine ist fest mit dem MBA verbunden und natürlich im Preis inbegriffen.

Ein Freund bietet einen Typenraddrucker TP-II von Smith-Corona für DM 600,- an. Die Daten:
12 Z/s, 10 Z/inch, 105 Z/Zeile
unidirektional, max. Papierbreite 33 cm
Schnittstellen parallel Centronics, seriell RS232C, Puffer 256 Bytes
Wer Interesse hat, wende sich bitte an
Frank Helferich, Schneidemühler Str. 20b, 7500 Karlsruhe 1 (0721-688828).

Arnulf Sopp

*** 80-Zeichen-Karte für unsere Computer zum Sonderpreis.
Bei einer Mindestbestellmenge von 6 Stück bekommen wir die Karte zum Clubpreis von 215,-DM + Porto.

- Bildschirmformat: 80 * 25 Zeichen/Zeilen
- 8 Zeichensätze á 128 Zeichen (deutsch, amerik., Grafik, invers)
- Bankinglogik für CP/M 2.2 eingebaut. Angepasstes CP/M erhältlich
- Treiber für NEWDOS und GDOS (HDOS) und Colour Genie + Handbuch im Lieferumfang. Ebenso Schalt- und Bestückungsplan, Listings der Treibersoftware + Diskette
- Ausführliche Informationen von Peter Spieß anfordern.

Der Stolz ist das Rückgrat des Menschen, und die Eitelkeit ist das Schwänzchen daran.

Der Computer einmal anders genutzt

Für viele mag es neu sein, aber mit einem Computer (selbst einem TRS 80) kann man nicht nur Daten verwalten, Briefe schreiben u.ä. sondern sogar rechnen. Das war scheinbar von den Erbauern nicht so vorgesehen, denn die Rechenarten werden in Handbüchern nur nebenbei erwähnt und außerdem rechnet das Ding verdammt ungenau (mit Single und Double), aber mit etwas Glück bekommt man doch ganz brauchbare Ergebnisse.
Ich möchte hier etwas zu Sinus und Cosinus sagen. Diese Funktionen braucht man eigentlich oft, wenn irgendwelche Räume oder Grafiken ins Programm mit hineinspielen. Also dort, wo es Punkte gibt, die Koordinaten haben und wo Winkel zu finden sind.

»FRAU MÜLLER, IHR SEID DIE SCHÖNSTE HIER!«

Diese Winkel stellt der normale Mensch nun in Grad dar. Wer allerdings glaubt, der Computer tut das auch, der wird enttäuscht: der TRS 80 arbeitet mit Radiant. Um einen Winkel von Grad in Radiant umzuwandeln, nehme ich den Winkel mal Pi und teile das ganze durch 180. Da der TRS 80 Pi nicht kennt, nehmen wir den Winkel einfach mal ".0174533". Jetzt können wir diesen neuen Wert für cos, sin und tan benutzen. Lassen wir uns aber durch atn (arcustangens, Texas Instruments: invtan) einen Winkel ausrechnen, so steht der auch in Radiant. Um die Einheit Grad zu erreichen, nehmen wir den Winkel also mal "57.29578" (bzw. mal 180 durch Pi).
Was machen wir aber, wenn wir arcsin oder arccos benutzen wollen? An diesem Punkt verläßt uns der Computer. Wir müssen diese Funktion also mit atn ausdrücken. Dafür gibt eine Formel:

```
arccos(x) := ATN(SQR(1-x*x)/x)
arcsin(x) := ATN(x/SQR(1-x*x))
```

Bei x=0 versagt die erste Formel, bei x=1 die zweite. Dann kommt jeweils Pi/2 bzw. 90 Grad heraus.
Nun sind diese Ergebnisse mit Vorsicht zu genießen, denn sie sind nicht eindeutig. Wenn Ihr eine Tabelle von -180 bis +180 Grad mit cos(x) und arccos(cos(x)) (oder sin(x) und arcsin(sin(x))) erstellt, werden sich einige verwunderliche Ergebnisse zeigen, die in der Natur dieser Funktionen begründet liegen. Vor allem liegen die Ergebnisse alle zwischen -90 und +90 Grad. Um also das wahre x-raus zu bekommen, nehmt Ihr folgende Unterscheidungen vor (y = Ergebnis durch atn in Grad!):

```
arccos:
 -90<y<0 : x=180+y    oder   x=-180-y
   0<y<90: x=y        oder   x=-y
arcsin:
 -90<y<0 : x=y        oder   x=-180-y
   0<y<90: x=y        oder   x=180-y
```

Es gibt dafür viele Anwendungen. Eine kleine sei hier genannt:
Ihr habt zwei Orte mit den Breiten-/Längengraden (b1/l1) und (b2/l2). Für den Winkel W zwischen den beiden Orten gilt:

```
   cos(W)=         sin(b1)*sin(b2)+cos(b1)*cos(b2)*cos(l1-l2)
<=>     W =arccos(                    "                       )
```

Dieser Winkel (in Radiant) mal dem Erdradius genommen, ergibt die Entfernung zwischen den beiden Orten (Luftlinie über der Erdoberfläche, nicht direkt durch den Erdball gebohrt!!!).
Also Entfernung KM=6368*W (in Kilometer).
Wenn ihr jetzt mal die Entfernung von Hamburg (10 O, 53 N) nach Hamburg (!) berechnet, werdet Ihr überrascht sein, wie unser TRS 80 es schafft, diesen Ort von sich selbst weg zu verschieben.

Für den arccos und arcsin gibt es sicherlich noch mehr Anwendungen (stellt doch mal welche vor!) und die Entfernungsformel läßt sich auch gut einsetzen (z.B.: wie weit hat es jedes Clubmitglied zum nächsten Treffpunkt; natürlich gilt das Ergebnis nur für Leute mit einem Hubschrauber). Ich selbst habe versucht, ein kleines Luftfahrt-Spiel daraus zu machen, bin aber später an einigen mathematischen Problemen gescheitert. Man kann trotzdem tolle Sachen berechnen, vor allem, wenn man sich auf Ausschnitte der Erdkugel beschränkt (Beispiel: ein Flugzeug kann mit vollem Tank 100 km fliegen; wo zwischen Hamburg und München muß es zum Nachtanken landen?)
Viel Spaß mit den Winkeln!

Gerald Schröder

(atn-Formel aus "Basic-Kurs", herausgegeben vom Gymnasium Hittfeld
Entfernungsformel von Elessar)

Hermann Kahlen

Urheberrechtsschutz für Programme

Am 1. 7. 1985 ist das Gesetz zur Änderung von Vorschriften auf dem Gebiet des Urheberrechts in Kraft getreten. Primärer Zweck der Neuregelung war es, ein neues Abgabesystem für Fotokopierer sowie für Bild- und Tonaufzeichnungen einzuführen. Gleichzeitig wurde der Urheberschutz auf Computer-Software ausgedehnt.

Hieß es früher in § 2 Abs. 1 des Urheberrechtsgesetzes: „Zu den geschützten Werken der Literatur, Wissenschaft und Kunst gehören insbesondere: 1. Sprachwerke, wie Schriftwerke und Reden", so lautet der Gesetzestext heute: „1 ... Sprachwerke, wie Schriftwerke und Reden, sowie Programme für die Datenverarbeitung."

Abwägung im Einzelfall

Damit sind Programme nunmehr ausdrücklich urheberrechtlich geschützt. Zwar hat dies nur klarstellende Bedeutung: Der Bundesgerichtshof hatte nämlich mit Urteil vom 9. 5. 1985 (Aktenzeichen I ZR 52/83) ein Urteil des Oberlandesgerichts Karlsruhe aufgehoben und zur weiteren Sachaufklärung zurückverwiesen. Er bejahte dabei grundsätzlich die Urheberrechtsfähigkeit von Computerprogrammen. Der Gesetzgeber hat diese Entwicklung nunmehr klar gebilligt. Zu beachten ist jedoch, daß § 2 Abs. 2 des Urheberrechtsgesetzes nicht geändert worden ist. Danach gilt: Werke im Sinne dieses Gesetzes sind nur persönliche geistige Schöpfungen. Das heißt, nicht die mathematische oder technische Idee, sondern ihre Verkörperung im fertigen Programm verdient und erhält Schutz. Die rein „handwerksmäßige" Handhabung allgemeinen Wissens ist somit wie bisher ungeschützt. Der Streit der Experten wird sich also zukünftig nicht mehr um die Frage drehen, ob Software als solche geschützt ist oder nicht; vielmehr wird man sich darüber auseinanderzusetzen haben, welche Software eine „persönliche geistige Schöpfung" im Sinne des Urheberrechtsgesetzes ist und welche bloße Handwerks-„Kunst" ist.

Freiheitsstrafe bis zu fünf Jahren

Geschützt wird Software zukünftig durch die Strafvorschriften des Urheberrechtsgesetzes. § 53 Abs. 4 Satz 2 bestimmt dazu: „Ebenso ist die Vervielfältigung eines Programms für die Datenverarbeitung (§ 2 Abs. 1 Nr. 1) oder wesentlicher Teile davon stets nur mit Einwilligung des Berechtigten zulässig." Verstöße gegen dieses Verbot werden gemäß § 106 mit Freiheitsstrafe bis zu einem Jahr oder Geldstrafe bestraft. Handelt es sich um gewerbsmäßige unerlaubte Verwertung, greift § 108 a ein. Danach ist eine Freiheitsstrafe bis zu fünf Jahren oder eine Geldstrafe vorgesehen und auch schon der Versuch mit Strafe bedroht.

Das genannte Einwilligungs-Erfordernis ist nur dann gegeben, wenn entweder das ganze Programm oder wesentliche Teile davon verwertet werden sollen. Die Verwertung unwesentlicher Teile wird nicht vom Verbot – und somit auch nicht von den Strafvorschriften – erfaßt. Somit taucht das Problem auf, abzuklären, was wesentlich oder unwesentlich ist. Dies wird sich nur von Fall zu Fall klären lassen. Maßgebend dürfte die Relation zum Ganzen sein.

Interessen-Wahrnehmung durch VG Wort

Es empfiehlt sich, eventuelle Vergütungsansprüche durch eine Verwertungs-Gesellschaft wahrnehmen zu lassen. Insoweit bietet sich die VG Wort an. Sie macht Ansprüche eines Software-Erstellers aus dem Urheberrechtsgesetz für diesen geltend. Voraussetzung ist, daß der jeweilige Urheber der VG Wort die von ihm geschaffenen Werke mitteilt. Nähere Einzelheiten sind bei der VG Wort, Goethestraße 49, 8000 München 2, zu erfragen (siehe auch mc-Impressum).

Anmerkung der Redaktion: Auf Anfrage teilte die VG Wort uns mit, daß es bisher noch keine konkreten Fälle der Interessenwahrnehmung durch sie im Bereich der Computer-Software gebe. Die VG Wort nehme lediglich „Sekundärinteressen" wahr, während die primäre Lizenzvergabe beim Software-Entwickler bzw. bei dem von ihm lizenzierten Unternehmer liege.

Literatur

[1] Kahlen, H.: Urheberrechtlicher Schutz für Software. mc 1984, Heft 11, Seite 52.
[2] Kahlen, H.: Verbesserter Urheberrechtsschutz. mc 1985, Heft 2, Seite 96.

Zitat des Monats

Im Amiga wird der Mikroprozessor sehr viel effektiver genutzt, da er in diesem Computer nur die Aufgabe, die er am besten beherrscht, nämlich das Addieren und Vergleichen von Zahlen, auszuführen hat.

Aus einem Test in der Zeitschrift „Computer Persönlich", 1985, Heft 19.

Urheberrecht

Zur Zeit ermittelt die Kripo gegen (mir bekannt) weit über 100 Softwarefreaks, die 1982/83 Tauschverbindungen zu anderen hatten. Die Adressenliste resultiert aus einer Durchsuchung im süddeutschen Raum. Hier beschlagnahmte die Kripo Tausch- und Adressenlisten sowie Disketten.

Ich glaube, richtig in der Annahme gehen zu können, wenn ich die Behauptung aufstelle, zu der Zeit hat jeder mit jedem irgend etwas getauscht. Ein Unrechtsbewußtsein beschränkte sich eigentlich nur auf den Verkauf kopierter Disketten, davon haben aber die meisten abgesehen.

Ich selbst habe an diesem regen Tauschhandel teilgenommen und seit Mitte Juli 1985 ein Ermittlungsverfahren wegen Verdachts nach §106, 108 Urheberrechtsgesetz (diese Paragraphen wären vielleicht hier einmal abzudrucken) anhängig. Entgegen Ihrer Behauptung in CP 15/85, Seite 75, wird nicht auf Antrag des Urhebers/Verwertungsberechtigten hin dieses Verfahren eingeleitet, sondern es erfolgt zur Zeit eine groß angelegte Ermittlung durch die Staatsanwaltschaft. Die Ermittlung begann morgens um 8 Uhr mit der Durchsuchung gemäß Beschluß des Amtsgerichtes der Wohn- und sonstigen Räume. Die zwei Kripobeamten drohten mit der Mitnahme meiner inzwischen mehreren Computer und sämtlicher Disketten. Da ich nachweisen konnte, diese Geräte zum rechtmäßigen Broterwerb zu nutzen und die Suche vorwiegend Spiele betraf, wurden bei mir 80 Disketten mit Spielen (nur Spiele!) sichergestellt, unter denen natürlich auch Kopien waren. Dabei habe ich die Feststellung machen müssen, daß die Ermittlungsbeamten (glücklicherweise) fast gar nicht in der Materie steckten. Ich mußte ihnen erst erklären, wo man Disketten anfaßt und wie man diese lagert. Jedem, dem so etwas passiert, kann ich nur raten, ebenfalls Hinweise zur Behandlung von Disketten den Ermittlungsbeamten auszuhändigen, damit Schaden an rechtmäßig erworbenen (diese werden nämlich auch mitgenommen, da die Ermittlungsbeamten hier nicht unterscheiden können) Disketten erst gar nicht entsteht.

Wie mir der ermittelnde Beamte auf Anfrage erklärte, werden die Disketten dann an die Staatsanwaltschaft weitergeleitet und wahrscheinlich durch einen »Verband der Softwareautoren c/o Commodore« geprüft. Danach wird dieser Verband die Urheber/Verwertungsberechtigten über diesen Vorgang informieren, die dann ihrerseits drei Monate Zeit haben, Strafantrag gegen den Besitzer/Kopierer der Disketten zu erheben. Ferner ist damit zu rechnen, daß die Steuerfahndung eingeschaltet wird. Auf mich wartet nun die Prüfung der Disketten, die entscheiden wird, ob Strafantrag wegen Verletzung des Urheberrechts gestellt werden kann oder nicht. Da ich in keinem Fall eine entgeltliche Weitergabe von Disketten betrieben habe, wird mir eine Anzeige wegen Betruges erspart bleiben, jedoch könnte ich schlimmstenfalls mit einer Anklage wegen Hehlerei rechnen und so könnte ein Hobby zur Vorstrafe führen.

Das Unrechtsbewußtsein ist mir bisher noch nicht gekommen und ich glaube, es wird auch nicht kommen. Bei den Softwarepreisen war man überhaupt nicht in der Lage, Disketten zu kaufen. Der Tausch war die einzigste Möglichkeit, andere Softwareprodukte kennenzulernen. Ohne diesen Tauschhandel und das geben sogar Experten zu, wären bestimmte Programme wie zum Beispiel Wordstar oder Multiplan nie bekannt geworden, und die Umsatzzahlen von Spieleherstellern hätten sicherlich klägliche Größen. Ich werde es auch nicht einsehen, warum renommierte Firmen auf diese kostenlose Werbung verzichten wollen. Da ich mittlerweile selbst Softwareautor bin, nutze ich darum die Kopiergewohnheiten einiger, um mich bekannt zu machen. Ich sehe es heute noch gerne, wenn ich erfahre, daß einer aus Österreich meine Programme hat und stolz anderen vorführt; mir würde nur der gewerbsmäßige Verkauf dieser Programme sauer aufstoßen.

Ich bin sicher, daß diese über 100 mir bekannten Ermittlungsverfahren nicht die einzigen sind und kann nur jedem, der damals oder heute Disketten tauscht, raten, alle Tausch- und Adreßlisten sofort zu vernichten oder zu verschlüsseln sowie eventuell seine Tauschpartner bitten, ein Gleiches zu tun (Der Verfasser dieses Briefes möchte aus verständlichen Gründen nicht namentlich genannt werden).

H.-P. Schmid hat einen neuen Club!

Vor ein paar Tagen bekam ich einen Fragebogen von Herrn
Dr. med. Friedrich Lücke
Deisterallee 14 A
3250 Hameln 1,
der offenbar den Zweck verfolgte, die gemeinsame Hard- und Softbasis der Mitglieder des Hamburger TRS-80- und VG-User-Clubs herauszufinden. Merkwürdig daran war lediglich, daß ich von diesem Club nie gehört habe und höchstens in Abwesenheit und Unkenntnis zur Mitgliedschaft verurteilt worden sein kann. Daß der Club keine Beiträge erhebt, machte mich ebenfalls stutzig, denn wer spendiert das Porto und die Kopierkosten für Infos usw.? Also bat ich Herrn Dr. Lücke um Aufklärung und kriegte einen offensichtlich für viele Empfänger konzipierten Formbrief, der folgendes aufdeckte: Zwei weitere Adressaten (die geantwortet hatten; wer weiß wer noch alles!) kannten den Club überhaupt nicht. Das wirft ein deutliches Licht auf die "Mitglieder"-Verwaltung dieses Clubs. Herr Dr. Lücke weiß nicht, wer für die Mitgliederliste verantwortlich ist!!! Ist der Boß des Clubs der Große Unbekannte?

Nein, es ist ein allzu Bekannter, glaube ich: Mit dem Formbrief kam auch eine Mitgliederliste (mit einigen Streichungen, versteht sich, denn sie enthielt auch Leute wie unsereinen).

<u>Hans-Peter Schmid ist dort Mitglied!</u>

Die vielen Merkwürdigkeiten lassen einen interessanten Schluß zu: Sollte der bekannte Jäger von Raubkopierern, der nur so Geld für die von ihm vertriebenen Programme erlangen kann, wieder mal einen Club gegründet haben? Wenn man dann den Kreis der Mitglieder scheinbar durch ein paar Ahnungslose künstlich erweitert, erweitert man damit auch die Möglichkeiten, Geld zu verdienen. Denn irgendwer wird schon so unvorsichtig sein, Software mit ihm oder einem seiner Helfer zu tauschen.

Herr Lücke hatte vor meiner Antwort an ihn wahrscheinlich keine Ahnung, in welch prominenter Gesellschaft er sich befindet, sonst hätte er mir nicht ausgerechnet ein solches Intimum wie die Liste der Mitglieder geschickt. Er ist demnach wohl kaum zu zeihen. Ebenso wenig sind es die Mitglieder, die gleichzeitig zu unserem Club gehören. überhaupt ist der Hamburger Club - von ihm gegründet oder vielleicht auch nicht - wohl nur ein willkommenes Werkzeug für Schmid, ansonsten aber vermutlich ein Haufen von netten Kollegen wie wir, mit Zielen wie den unseren. Zu warnen ist deshalb nicht vor Dr. Lücke, auch nicht vor dem Club schlechthin, aber davor, den Fragebogen auszufüllen, wo es um die Software geht. Und vor allem davor, mit Angehörigen dieses Clubs Programme zu tauschen, die man nicht selber geschrieben hat.

Seien wir mal ehrlich; fast jeder von uns kennt einen, der einen kennt, der schon einmal etwas raubkopiert hat. Wahrscheinlich ist Gevatter Schmid sogar formaljuristisch im Recht. Also haben wir genug Anlaß, auf den Fragebogen gar nicht erst zu reagieren. Schon gar nicht mit einem "geklauften" NEWSCRIPT, das Schmid vertreibt.

Arnulf Sopp

„Die Beleidigungen sind die Argumente jener, die über keine Argumente verfügen."

Eine gute und eine schlechte Nachricht:
Erst die schlechte: TCS in St. Augustin hat pleite gemacht.
Nun die gute: die Geschäfte der TCS GmbH übernimmt die Phoenix Computer GmbH,
5461 Windhagen, Klarenplatz. Geschäftsführer ist *Trommeschläger!*
(Wie sowas geht, ist mir ein Rätsel; aber seitdem AT&T (amerikanische Telefongesellschaft) in deutschen Zeitschriften für Telefongespräche über'n Teich wirbt, wundert mich gar nichts mehr.)
Frei wiedergegeben aus: CP 22/85

Wer sich für das mms-Forth interessiert, findet Informationen dazu in:
Ken Knecht: Einführung in Forth; Markt und Technik Verlag AG; 8013 Haar; ISBN 3-922120-73-3; 218 Seiten; 58,- DM
aus: CP

Gerald Schröder

„Keine Zeit ist mit der Zeit zufrieden; das heißt, die Jünglinge halten die künftige für idealer als die gegenwärtige, die Alten die vergangene."

CLUB 80 - Programmbibliothek exklusiv

Zwanzig neue Programme kann ich Euch heute präsentieren. Es könnten erheblich mehr sein, aber leider war ich in letzter Zeit sehr beschäftigt und so blieb manches liegen, was eigentlich hätte gemacht werden müssen. Wer also ein eingeschicktes Programm in der neuen Liste vermißt, braucht nicht zu befürchten, es sei in der Ablage "P" gelandet, sondern muß bis zum nächstenmal warten.

Besonders erwähnen muß ich hier das Demoprogramm zur Relaissteuerkarte von Harald Trapp. Das Demoprogramm entspricht praktisch dem Originalprogramm, benötigt jedoch nicht die entsprechende Hardware. Man kann sich also, bevor man sich für die Anschaffung der Karte entscheidet, schon einmal über die Software und die Funktionsweise informieren. Das Programmpaket kann unter dem Namen "Relaissoftware" angefordert werden.

Zuletzt noch eine kurze Bemerkung über die Lieferzeiten der Bibliotheksprogramme. Im Normalfall versuche ich die Bestellungen noch am Tag des Eingangs zu erledigen, was mir bis jetzt auch meist gelungen ist. Da ich in der nächsten Zeit jedoch meine Maschine auf 64k aufbohren und CP/M-fähig machen will, kann es zu erheblichen, hardwarebedingten Verzögerungen kommen.
Weiterhin suche ich, bis spätestens zum Treffen des Clubs im März nächsten Jahres (besser bis Januar), einen Nachfolger in meiner Tätigkeit als Programmverwalter. Der Grund dafür liegt in meiner, durch berufsfördernde Maßnahmen (BW-Jargon), stark eingeschränkten Freizeit und erhöhter Arbeitsbelastung. Wer sich dafür zur Verfügung stellen möchte, tritt bitte direkt mit mir in Kontakt. Ich hoffe, es findet sich jemand, da ansonsten die mühsam aufgebaute, gereinigte und erhaltene Programmbibliothek wohl zum Sterben verurteilt würde!

Das wars mal wieder. Ich hoffe, ihr habt für die etwas verlängerten Lieferzeiten und für meinen Entschluß, die Betreuung der Clubbibliotehk abzugeben, Verständnis und es würde mich freuen, bald mal was von euch zu hören, euer,

Hartmut Obermann

CLUB 80 - Programmbibliothek die neuen Programme

OELDATEI — A 0227 A 02 /BAS

Dieses Programm hilft dabei, den Ueberblick ueber die Heizoelreserven und -Kosten zu halten.

Grafikeditor — Z 0228 A 01 /BAS --

Mit diesem Grafikeditor für Pixelgrafik kann man Bilder von Disk laden und auf Disk speicher. Ausserdem kann man die erzeugten Bilder ausrucken lassen.

Zeicheneditor fuer Gemini 10 — T 0190 A 05 /BAS --

Ein Bildschirmeditor zum Erstellen und Editieren von Sonderzeichen für den STAR Gemini x10 Drucker. Quelle: Chip 9/85

BASIC-Formlister — S 0191 B 02 /CMD --

Listet BASIC-Programme besonders uebersichtlich auf Bildschirm oder Drucker.

Farbenproblem — D 0193 A 04 /BAS

Eine veränderliche Anzahl von Feldern muß mit vier Farben ausgefüllt werden. Dabei dürfen zwei angrenzende Felder nicht die gleiche Farbe haben.

Grossbuchstaben — S 0194 A 02 /BAS

Das Programm stellt Grossbuchstaben für Programmköpfe und ähnliches z ur Verfügung.

LUNAR — D 0196 A 02 /BAS --

Mondlandung ohne Grafik, jedoch mathematisch relativ exakt.

MINICALC — M 0197 A 03 /BAS

Aehnlich Visicalc, jedoch in BASIC und mit erheblich eingeschraenkten Funktionen.

OSCAR — M 0202 A 05 /BAS

Programm zur Berechnung der Position des Amateurfunksatelliten OSCAR. Nach Aenderung der Bahnwerte ist das Programm auch fuer andere (künstliche) Himmelskörper zu verwenden.

Reaktionstest — G 0207 A 02 /BAS

Ein nettes Spielchen für die Party. Man kann damit sehr schnell feststellen, ob man noch reaktionsfähig und damit fahrtüchtig ist!

Spooler — S 0211 K 05 /CMD

Endlich ein Druckerspooler der speziell für den KOMTEK (andere Drucker - und Treiberadressen) geschrieben wurde. Autor: Walter Zwickel

---

LM 381 Vorverstärker IC M 0214 A 04 /BAS

Berechnet einen rauscharmen Vorverstärker mit dem Spezial-IC LM381.

Umwelt D 0215 A 03 /BAS

Eine Regierungssimulation, bei der es vor allem auf die richtige, ökologische Wirtschaftsführung ankommt.

ZAP G 0216 A 01 /CMD

Ein mittelmäßiges Schießspiel mit Ton.

Kassettenverzeichniss S 0217 A 02 /BAS

Mit diesem Programm kann man sehr leicht und schnell für seine Musik- (oder Programm-) Kassetten Inhaltsverzeichnisse drucken. Autor:Günther Wagner

---

Disassembler S 0219 D 04 /BAS

Ein einfacher Disassembler in BASIC für das Model 4 von Tandy.

MAKEDO S 0221 D 02 /BAS

Programm zur Erstellung von DO-Files (ähnlich CHAINBLD) für das Model 4 von Tandy

---

Scrollprotection S 0222 D 03 /BAS

Mit diesem Programm kann man 1-7 Zeilen des Bildschirms vor dem scrollen schützen. Nur für Model 4 von Tandy!

SETRESET S 0223 D 03 /BAS

Dieses Programm simuliert die BASIC-Befehle SET, RESET und POINT auf dem Model 4 von Tandy.

MENUE4 S 0224 D 04 /BAS --

Ein Programm zum Aufruf von BASIC- und JCL-Files auf Tastendruck. Nur für Model 4 von Tandy

---

»Es ist nett, wichtig zu sein. Aber es ist wichtiger, nett zu sein.«

»So, Fräulein Knickdübel, passen Sie gut auf, wie er's macht – dann brauchen wir nicht jedesmal auf die teuren Fachleute zu warten, nur weil das Ding ein bißchen bockt!«

Die CLUB 80 - Bücher

Nr. 0001: BASIC-Brevier: Eine Einf. in d. Prog. v. Heimcomp.

Siegmar Wittig --- Heinz Heise GmbH

Ein BASIC-Kurs fuer Nicht-Mathematiker und echte Amateure. Neben Grund- u nd Aufbaukurs gibt es zahlreiche Beispiele (eine gute Sammlung!).

Nr. 0002: Computerwissen

Michael Scharfenberger --- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von H ard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knobeleien programm. in BASIC

Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

- --- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel --- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ue ber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80

Chris Palmer --- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man besse re Programme schreibt.

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann u nd vor allem wie. Beispiele und Aufgaben veranlassen zum experiment.

Nr. 0008: More Basic Computer Games

David H. Ahl --- Creative Computing Press, USA

84 Spiele fuer den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist fuer Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (als o mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (als o mit Lern-Effekt).

Nr. 0011: TRS-80 PROGRAMS

Tom Rugg und Phil Feldman --- Dilithium Press, Beaverton,USA

32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012: Programme und Tricks fuer Genie I und Genie II

Clemens Becher, Franz Seger --- ?

Viele Programme, Tips und Tricks fuer den Genie.

Nr. 0013: BASIC: Alles ueber PEEK und POKE

Heiko Requardt --- Franzis-Verlag

Eine Software-Sammlung in BASIC (mit vielen guten Tips und Tricks fuer de n 'Amateur').

Nr. 0014: TRS-80 und Video Genie ROM-Listing fuer Level II

Luidger Roeckrath --- ?

ROM-Listing, RAM-Adressen, I/O-Adressen, Unterprogramme, Basic-Anweisunge n und Funktionen, Aufzeichnungsformate auf Cassette, ...

Nr. 0015: Machine Language Disk I/O & Other Mysteries

Mihael J. Wagner --- IJG Inc., 1953 West, USA

Alles rund um Disk und DOS wird erkl rt - mit zahlreichen wertvollen Hinw eisen und Tips.

Nr. 0016: Das Modem-Sonderheft (Daten bertr. mit Mikrocomp.)

verschiedene --- MC, Franzis-Verlag, M nchen

Grundlagen, Schaltungstechnik, DF -Programme

Prädikat "wertvoll": Grosser, "Das DOS Buch"

Von einem Text für mehr als 10 Leser, also einer Veröffentlichung im wörtlichen Sinne, erwartet der Käufer zu Recht ein fehlerfreies Deutsch. In "Das DOS Buch" von Hartmut Grosser, bei Röckrath zu haben, ist schon der Titel falsch. Auch im Text ist manche orthographische Eigenwilligkeit zu finden. Aber damit ist auch, was Negativkritik an diesem Buch betrifft, mein ganzes Pulver verschossen. Das ist das Beste, was nach meiner Kenntnis zu diesem Thema auf dem Markt ist! Ein Reader für Einsteiger, die begreifen wollen, wie ihr DOS funktioniert, ein unentbehrliches Vademecum auch noch für den ganz eingefleischten Disk-Jockey, der der alten Tante NEWDOS-80 oder ihren Neffen aus der TCS-Sippe endlich das Laufen beibringen will.

In den ersten beiden Kapiteln wird erläutert, wie der Controller arbeitet und wie sich das ROM sein Lieblings-DOS reinschlürft. Anschließend folgt das Kapitel, das alleine schon den Preis von (leider nicht weniger als) 65 Mark wert ist: Ein ausführlich kommentiertes Listing von SYS0/SYS. Dort werden in zusätzlichen Kommentarzeilen auch Fehler gegeißelt, die in einem späteren Kapitel noch einmal zusammengefaßt sind. Auch alle wichtigen Einsprungsadressen für den Maschinensprache-Anwender sind noch einmal lexikographisch und alphabetisch zusammengestellt.

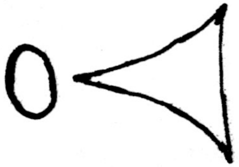
Wer gerne in der Library fummelt, um eigene DOS-Befehle zu implementieren, wird in Kapitel 5 erschöpfend darüber informiert, was er dazu wissen muß. Solche Befehle landen sinnvollerweise möglichst in einem SYS-File (bei Grosser weiblich), und darüber erfährt der Leser einiges. SYS-Files lassen sich ohne nähere Kenntnisse des Directory (bei Grosser männlich) kaum adressieren. Folgerichtig läßt Kapitel 7 hierüber keine Fragen offen.

Die Banking-Freaks unter uns, verliebt in ihren EG 64 MBA oder wen auch immer, staunen in Kapitel 8 nicht schlecht: Das DOS unternimmt bei bestimmten Verrichtungen Dummy-Schreibzugriffe auf das ROM. Am Samstagmorgen kam das Buch mit der Post, am Sonntagnachmittag waren die entsprechenden Adressen (im Buch natürlich haarklein erläutert) auf den Bildschirm-Adreßbereich umgezapt. Wer nämlich gerade mit einer modifizierten ROM-Kopie im Parallel-RAM arbeitet, kann z. B. bei einem Verify-Gang sein blaues Wunder erleben. Daß ich nun etwa Kommentar-Records live auf dem Screen erlebe, ist weniger störend als ein ausgewachsener Systemcrash. Die Tastatur ab 3800 wäre eleganter gewesen, aber seit kurzem ist sie bei mir nur noch 256 Bytes lang. Diese Adresse sei aber allen empfohlen, die das "Sichere Plätzchen" von Helmut Bernhardt (c't 5/85) nicht haben.

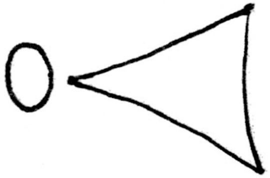
Im Anhang gibt es einen Artikel meines langjährigen Freundes Ulrich Heidenreich über seinen neuen Library-Befehl ID. Mit ihm erkennt das DOS automatisch die PDRIVE-Parameter fremder Disketten. Das Gewühle beim Ausprobieren, wenn mal wieder jemand seiner Platte keinen Zettel beigefügt hat, gehört damit der Vergangenheit an. Viele von euch benutzen den Befehl bereits mit Erfolg. Wie ich neidvoll zugeben muß, ist ausgerechnet dieses einzige von H-DOS, das nicht von mir stammt, das interessanteste Feature.

In dieser Rezension habe ich nur eine Auswahl dessen erwähnt, was im "DOS-Buch" (die Freiheit des nachträglichen Bindestrichs sei mir erlaubt) steht. Es ist nur dasjenige, was mich persönlich besonders interessiert. Versteht sich, daß alles für NEWDOS-80 2.x, G-DOS 2.x und Colour-DOS gemünzt ist. Die jeweiligen Unterschiede sind extra aufgeführt. Und das alles für alle Genies und alle Modelle des TRS-80. Selten habe ich beim Kauf eines Buches pro Groschen so viel Information bekommen.

Arnulf Sopp



Für das DOS-Buch ist bei Peter Spieß eine Sammelbestellung eingerichtet. Clubpreis: 59,-DM abzgl. 2% (Porto ist schon dabei). Bitte Bestellung baldmöglichst abschicken und die Unterschrift nicht vergessen.



- Impressum -

Vorstand

Kontaktadresse für Clubangelegenheiten, Clubbücherei /Fundgrube, Clubkasse:
Günther WAGNER
Gartenstraße 4
8201 Neubeuern
Tel.: 08035 /3361
< 18 - 20 Uhr >

Programmbibliotek

Kontaktadresse:
Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod /Kemel
Tel.: 06124 /3913

Redaktion

Kontaktadresse:
Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
Tel.: 0791 /42877
tagsüber 0791 /44-667

Adventure-Ecke

Kontaktadresse:
Alexander WAGNER
Theresienstr. 21c
8224 Chieming
Tel.: 08664 /1500

Hardware

Kontaktadresse:
Walter ZWICKEL
Lengfelden 123
5101 Bergheim (Austria)
Tel.: 0043662 /51130

Redakteure dieser Ausgabe

Jens Neueder * Hartmut Obermann
Bernd Retzlaff * Gerald Schröder
Arnulf Sopp * Günther Wagner
sowie Artikel aus: MC und Computer Persönlich

Bankverbindung des CLUB 80
Sparkasse Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postscheckkonto der Sparkasse
Nr. 8077-801

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

Zum Abschluß des INFO wie immer ein paar Redaktionsworte.

Gerade habt Ihr das 10. und in diesem Jahre auch letzte Clubinfo in der Hand. Bei dieser Gelegenheit möchte sich nun die Redaktion bei Euch für Eure Mühen als Beitragende sowie für Euer Interesse am Clubinfo bedanken. Die Zusammenarbeit mit Euch hat mir sehr viel Freude bereitet. Ich hoffe, Ihr laßt mich auch im nächsten Clubinfojahr nicht im Stich und sendet mir Eure Erfahrungen, Kniffe und anderes Wissenswerte aus dem Computerleben.

Gleichzeitig habe ich aber auch den Wunsch für's nächste Jahr, daß sich mehr Mitglieder an dem Clubinfo mit Beiträgen beteiligen, denn in diesem Heft stammen fast zweidrittel der Artikel -wie Ihr sicher selbst gemerkt habt- aus der Feder von Arnulf und Hartmut. Daß für die Beiden das Clubinfo nun nicht mehr so interressant ist, dürfte klar sein. Aber Ihr könnt Euch ja bei den zwei eifrigen Schreibern mit Eurem Clubinfobeitrag revangieren. Wie schon des öfteren erwähnt, finden auch kurze Beiträge ihren Platz im Clubinfo.

Achtung !!!

REDAKTIONSSCHLUß für INFO 11 ist der

31. DEZEMBER 1985

Nun noch zwei Neuerungen :

1. In Zukunft wird für die Clubinforubrik Börse eine "Hot-Line" eingeführt. Dies bedeuted für Euch, daß Ihr mich Montags abends zwischen 21Uhr und 22Uhr anrufen, und Eure Börsenaufträge durchgeben könnt. Sie werden dann gesammelt und in der nächstmöglichen Clubinfo veröffentlicht. Ich hoffe, daß dies unsere Börsenspalte etwas wiederbeleben wird.

2. Ab diesem INFO ist die Adressenseite die letzte Infoseite, und zwar so "eingebaut", daß Ihr ohne Aufschlagen der INFO die aktuellen Mitgliederadressen lesen könnt. Die "Rückseite" der Adressenseite bleibt frei. Diese Seite ist für Mitteilungen der Redaktion an den Vorstand, die Redakteure oder andere Informationen an Euch persönlich vorgesehen. Ein weiterer Grund ist, daß ich die Seite bei Probeinfos an Nichtmitglieder abtrennen kann. Ihr seht, auch wir betreiben Datenschutz !! Falls nach der Adressenseite noch Seiten angehängt sind, so sind diese zum Abtrennen für Euch oder als Antwortseite zum Zurücksenden gedacht.

Als letztes möchte ich Euch allen zu Weihnachten und zum Jahreswechsel alles Gute und weiterhin Happy Computing wünschen. (Das nächste INFO kommt erst im Januar 86)

So, nun am Ende des 10. INFO angelangt, hoffe ich, daß Ihr auch diesmal wieder Gefallen an Eurer Clubinfo habt, und verbleibe --

-- bis zum nächsten INFO Euer

J. Neueder

CLUB 80 Mitgliederadressen**liste**

Name	Vorname	Straße	PLZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Baldes	Hans	Johann-Strauss-Str. 6	8025	Unterhaching	089 /6115179
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Drowälder	Bernd	Buchentalsweg 8	4939	Steinheim	
Fröhlich	Burghard	Brandströstraße 11	4350	Recklinghausen	02361 /63416
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Hallupp	Matthias	Stockumer Straße 405	4600	Dortmund 50	0231 /756413
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
Marx	Andreas	Mecklenburgring 48	6600	Saarbruecken	0681 /812983
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Ronenstrasse 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 56	4600	Dortmund 1	
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194
Wies	Jean-Claude	Harthweg 9	6600	Saarbruecken	0681 /582513
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Wegen Systemwechsel <u>ausgetreten</u>
Jürgen Bozek (zum 1.1.86)

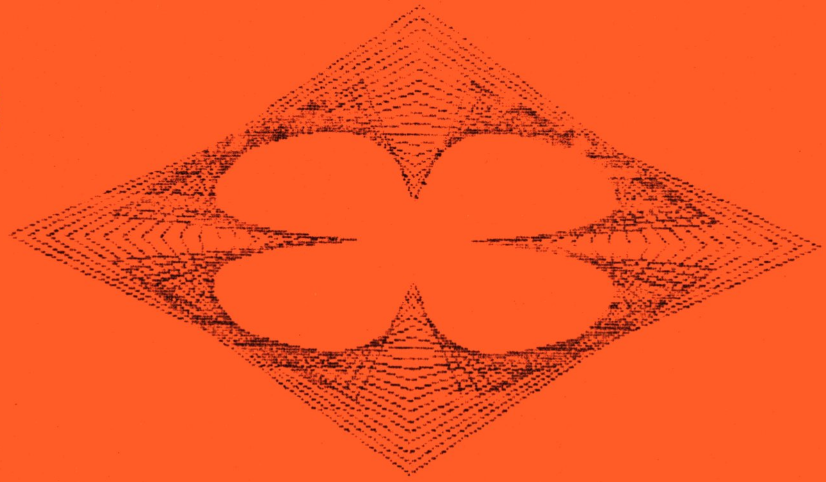
Neuzugang :
Bernd Drowälder

Adressenänderung
Peter Stevens

überprüft bitte Eure Adresse auf Richtigkeit !!!

CLUB 80

Clubinfo der TANDY - GENIE - und KOMTEK - ANWENDER



Kleeblatt, Pendel
und Weltkarte sind HRG-Produkte

11. AUSGABE

Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubeuern

Tel.: 08035/3361 (18 - 20 Uhr)

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert . . . . 01 - 03
An die "2. Hälften" 04

Software

nochmals INPUT . . . . 05 - 06
Files wie Sand am Meer . . . . 07 - 08
Adventure - Ecke . . . . 09 - 17
Programmierwettbewerb . . . . 19 - 20
Fehler bei BOOT !! . . . . 21 - 22
Primzahl '39' . . . . 23 - 24
NEWDOS/80 + 10 (F,DR,LF,OUT) . . . 25 - 30
Extended BASIC . . . . 31 - 32
NEWDOS/80 + 10 (I,P,DISK). . . . . 33 - 37

Hardware

Verbindend . . . . 39 - 40
ECB-Adapter für alle . . . . 41 - 46

Seite:

Börse

Wer hat was ???--Wer sucht was ??? . 47 - 48

Sonstiges

BASIC-Vergleichstabelle . . . . 49 - 63
Murphy-Gesetz . . . . 65 - 66
Folgerungen aus Murphy . . . . 67 - 68

Programmbibliothek

Exklusiv 69
Das letzte Mal 69
Neue Programme . . . . 70 - 73
Unter neuer Verwaltung . . . . 74 - 75

Club -Bücherei

Kurzmitteilung 64

Die letzen Seiten

Impressum 75
Schluß 76
Adressenliste . . . . letzte Seite
Kalender und Anleitung . . . . am INFO-Ende

Liebe Mitglieder,

wir beginnen mit diesem 11. Clubinfo das Jahr 1986. Rückblickend auf das Jahr 1985 möchte ich sagen, daß es für uns Mitglieder des CLUB 80 ein sehr erfolgreiches war:

- unser erstes Clubtreffen war ein großer Erfolg; u.a. wurde dort auch die Satzung überarbeitet.
- der Jens übernahm das Clubinfo - er hat damit eindeutig die meiste Arbeit von allen Mitgliedern - er leistet hier eine hervorragende Arbeit. Die Übernahme durch Ihn hat unserem Clubinfo wirklich gut getan.
- der Hartmut übernahm die Programmbibliothek. Er hat sich die große Arbeit gemacht, geschützte Programme auszusortieren. Er hat die Programmbiblithek wirklich gut verwaltet und trotzdem noch Zeit gefunden, sehr viele Beiträge zum Clubinfo beizusteuern.
- viele interessante Beiträge (große und kleine) konnten wir im Clubinfo lesen - gedankt sei hier allen Redakteuren, die sich die Zeit genommen haben oder sich getraut haben, etwas im Clubinfo zu veröffentlichen.

Ich möchte mich an dieser Stelle vor allem beim Jens und beim Hartmut bedanken - ohne deren Engagement und Idealismus wäre vieles nicht möglich gewesen im vergangenen Jahr.

Ich hoffe, daß wir auch 1987 auf ein gutes Jahr 1986 zurückblicken können - wir können das sicherlich, wenn wir alle zusammenhelfen, wenn jeder etwas zum Club beisteuert (außer dem Jahresbeitrag).

Aber nun zu den Finanzen.

Wir haben seit diesem Jahr drei Kassen; eine Hauptkasse und zwei Nebenkassen:

- Hauptkasse, wird von mir verwaltet
- Nebenkasse vom Jens Neueder für seine Ausgaben als Betreuer des Clubinfos
- Nebenkasse vom Hartmut Obermann als Betreuer der Programmbibliothek.

Der Kassier wird jeweils beim Clubtreffen entlastet. Beim Clubtreffen kann auch jedes Mitglied Einsicht in die 3 Kassenbücher nehmen. Deswegen sei hier nur eine sehr kurze Übersicht angegeben, wobei die drei Kassen zusamengefaßt sind:

ZUSAMMENSTELLUNG

	Übertrag aus dem Jahr 1984	- 114.07	DM
Einnahmen:	Clubbeiträge und Aufnahmegebühren	+1615.00	DM
	Nachzahlung 10 DM	+ 400.00	DM
	Unkostenbeitrag für Clubinfos	+ 122.50	DM
	Sonstige Einnahmen	+ 70.00	DM
Ausgaben:	Briefmarken	- 245.20	DM
	Photokopien	- 524.70	DM
	Verbrauchsmaterialien, Sonstiges	- 185.03	DM
	Disketten, Kopien, Porto etc.(Hartmut)	- 142.55	DM
	Info 7-10 mit sämtlichen Nebenkosten (Ausgaben vom Jens Neueder)	- 544.65	DM

Somit schließen wir das Jahr mit einem Überschuß in Höhe von ca. 450 DM ab.
Dieser Überschuß wird laut Clubsatzung u.a. zum Ankauf von Büchern und Programmen verwendet. Der Hartmut hat von einem Verlag einen Sonderprospekt mit günstigen Preisen erhalten. Er hat mich auf die günstigen Preise der Bücher aufmerksam gemacht, und wir haben beschlossen, 3 Bücher zu bestellen.
Ferner liegt ein Verkaufsangebot über Bücher von unserem ehemaligen Mitglied Mathias Hallup vor. Wahrscheinlich werde ich vom Mathias Hallup für den CLUB 80 zwischen 9 und 19 Bücher zu einem Höchstpreis von 250 DM abkaufen. Zum jetzigen Zeitpunkt kann ich hierzu noch nichts näheres mitteilen - also bitte bis zum nächsten Info warten. Auf alle Fälle werden die Bücher nur nach Rücksprache mit dem Jens und dem Hartmut angeschafft.
Ich hoffe, daß wir (der Jens, der Hartmut und ich) im Sinne der Mehrheit der Mitglieder entscheiden.

Zu den Ausgaben ist noch etwas zu sagen: Durch den günstigen Vervielfältigungspreis, den wir über den Peter Spieß erhalten, konnten wir die Ausgaben erheblich vermindern. Eine weitere Einsparnis haben wir durch geringere Portokosten, da wir die gebundenen Clubinfos (die ab Nr. 7) als Büchersendung verschicken können. Nur so konnte überhaupt ein Überschuß in Höhe von etwa 450 DM entstehen. Ich möchte nicht versäumen, an dieser Stelle auch dem Peter Spieß meinen Dank für seine Arbeit auszusprechen.

Einige haben den Jahresbeitrag noch nicht bezahlt - bitte daran denken falls vergessen!
Mittlerweile ärgere ich mich auch über 4 Mitglieder, die bis jetzt noch nicht Ihre 10 DM Nachzahlung bezahlt haben. Dieses Quartett bekommt dieses Info erst nach Bezahlung der 10 DM.

CLUBTREFFEN

Auch hier möchte ich einiges anmerken. Bis Weihnachten sind bei mir erst 18 Anmeldungen eingetrudelt. Davon nehmen 6 Mitglieder am Treffen nicht teil. Es nehmen also bis jetzt 12 Mitglieder am Treffen teil. Von diesen 12 Mitgliedern werden 4 Mitglieder von der Frau bzw. Freundin begleitet.

Mir ist klar, daß viele zum jetzigen Zeitpunkt noch nicht sagen können, ob sie am Treffen teilnehmen können oder wollen. Dann schickt mir aber bitte trotzdem die Anmeldung zu und schreibt drauf, ob Ihr teilnehmt oder ob nicht und wann ev. Ihr näheres mitteilen könnt. Ich brauche unbedingt von den anderen 27 Mitgliedern, die bis jetzt noch nichts bezüglich Clubtreffen verlauten lassen haben, entsprechende Mitteilungen.

Auch die Mitglieder, die am Treffen nicht teilnehmen, müssen mir dies unbedingt mitteilen!

Ihr denkt hoffentlich alle daran, daß auf dem Clubtreffen ein neuer Vorstand zu wählen ist. Mit dem alten Vorstand (daß heißt mit meiner Person) bin ich selbst nicht zufrieden. Zu wenig pflege ich meiner Meinung nach die Kontakte mit den anderen Mitgliedern; viel zu selten denke ich an meine Aufgaben als Clubvorstand.
Warum ? Ich habe wegen dem Studium (Diplomarbeit und Hauptdiplomprüfungen) viel zu wenig Zeit und außerdem nicht mehr so das Interesse am Amt des Vorstandes. Viel zu sehr stehen bei mir einfach andere Interessen (berufliche und private) im Vordergrund.
Ich stehe also mit Sicherheit bei der Neuwahl als Kandidat nicht mehr zur Verfügung. Ich hoffe, daß sich im Club Mitglieder befinden, die sich bereit erklären, das Amt des Vorstandes zu übernehmen. Sollte ein Mitglied bereit sein, für das Amt zu kandidieren und kann er am Clubtreffen nicht teilnehmen, so muß er seine Kandidatur unbedingt bei mir schriftlich melden, damit wir Ihn bei der Neuwahl auch berücksichtigen können. Es ist hoffentlich jedem von Euch klar, daß wir auf dem Clubtreffen unbedingt einen neuen Vorstand wählen müssen - schön wär es, wenn wir mehrere Kandidaten hätten.
Wer Interesse am Amt des Vorstandes hat, sich aber über seine Aufgaben unklar ist, sollte sich bei mir melden.

Ich hoffe, daß ich jetzt nichts vergessen habe. Bis zum nächsten Mal

Euer

Günther Wagner

An alle computergeschädigten Ehefrauen und Freundinnen!!

Wie Ihr sicherlich wißt ist vom 11. bzw. 12. - 13.04.85 das Clubtreffen unserer Männer. Ich würde mich aus diesem Anlaß sehr freuen, wenn Ihr sie begleiten würdet.
Gemeinsam könnten wir einiges unternehmen, z. B. einen Stadtbummel durch Limburg (das sehr schön ist), Wiesbaden oder Rüdesheim. Außerdem gibt es im Hotel ein Schwimmbad. Zusätzlich hat sich unser Clubmitglied Walter Piller bereiterklärt, seine Erfahrungen im Brotbacken an uns kostenlos weiterzugeben. Wer Interesse hat, melde sich bitte bei mir. Tel. 06124/3913.

Ihr seht, hier kann man vieles unternehmen. Vor allem können wir die Gelegenheit zu einem Gedankenaustausch nutzen.

Ich hoffe auf Euer zahlreiches Erscheinen und verbleibe mit freundlichen Grüßen

Jutta Obermann

Herr Doktor, hier ist meine Frau – sie bildet sich ein, ein Computer zu sein!

* FILES wie Sand am Meer... * *

Vieles gibt's wie Sand am Meer – zum Beispiel: Wenn vieles als Files auf fast gefüllten Floppydisks wohlfeile Felder fand und viele Fehler anfallen, weil vielleicht feinfühlige, frei formatierte Informationen fehlen!

Nach kurzem Luftholen: Ich meinte, wenn man sich vor lauter Filenamen, deren 8-11stellige Abkürzungen ihren Inhalt mehr verschleiern als entblößen, nicht mehr auskennt – und wenn man sich in der sandreichen Menge Disks nicht mehr zurechtfindet, weil es eben immer mehr, eben weil es ein Meer zu speichern gibt – dann muß irgendetwas her, was hilft.

Das kann eine Datenbank sein. Muß nicht.

Günther gab uns in Heft 9, Seite 4, eine Anregung unter dem Titel "Directory lesen" und legte die Bearbeitung eines Unterprogrammes aus CHIP vor, das sich den/die/das Directory direkt aus DIR/SYS holt und unsere sorgfältig hingehaltenen Variablen damit füllt.

Ferner bat er, auch negligible "Leistungen" weiterzugeben. Deshalb scheue ich mich nicht, den bescheidenen Gebrauch, den ich von seinem UP gemacht habe, indem ich meinen "DISCAT" damit baue, hier als Anwendungsbeispiel zu veröffentlichen.

Zeilen 1000-1130 stammen aus der Vorlage. Diese machen – nach dem nötigen Vorspann – den Auszug aus dem/der/dem Directory (ich stehe mit der "Artikulierung" der engl. termini technici genau so auf dem Kriegsfuß wie Ihr und gebe daher meist allen drei Artikeln das gleiche Recht). Danach kann man wählen (warum "optieren"?), ob man die Filenamen schön säuberlich auf dem Bildschirm ausgegeben oder auf Papier gedruckt haben möchte.

Und im letzten Fall gibt's die Möglichkeit, den Filenamen einen hinreichend langen Kommentar zuzuordnen – das wär's schon!

Das Listing wurde übrigens mit dem "NAME"-Befehl ausgedruckt. Dieser hat seine Tücken, über die man eigentlich hier auch einmal berichten sollte (vielleicht ein andermal). Hier soviel: wer's abtippt, muß wissen, daß an jedem Zeilenende ein "Delimiter" zu denken ist, zu Deutsch: ein Syntax-Begrenzer.

Das ist nun allerdings kein Adliger aus Bregenz (y und z in Namen sind selten), die Betonung liegt vielmehr auf der zweiten Silbe. Man denke sich also am Ende jeder Zeile entweder ein ":" – ausgenommen nach der jeweils letzten Zeile vor der nächsten Zeilennummer – oder ein "," – – welches von beiden, tja, da muß man eben aufpassen; und deshalb finde ich "NAME" zwar "schön", aber nicht sehr praktisch... Was meint Ihr?

Bis demnächst! Euer KlauJot... →

DIRECTORY-Ausdruck mit FILE-Erläuterungen

```
  10 CLS
     CLEAR 2500
     DIM A$(200)
     PRINT " Directory lesen und verarbeiten (einschl. unsichtbarer Files)
  15 REM      >>>>> (C) KaJot Mühlenbein * Dezember 1985 <<<<<
  20 PRINT STRING$ (63,"*")
  30 PRINT
     INPUT "Welches Laufwerk ";DR
  40 GOSUB 1000
  45 PRINT
     D$ = ""
     INPUT "Directory-Ausgabe auf Monitor oder auf Drucker <M oder ENTER>";D$

  46 IF D$ = "M" OR D$ = "m"
     THEN GOTO 50
     ELSE INPUT "Mit Kommentaren <J oder ENTER> ";J$
          GOTO 1500
  50 CLS
     FOR I = 0 TO J STEP 45
     FOR K = 0 TO 44 STEP 3
     FOR L = 0 TO 2
  55 IF A$(I + K) = ""
     THEN END
  60 PRINT USING "##";I + K + L + 1;
  65 PRINT ") "A$(I + K + L);"    ";
  70 NEXT L
     PRINT
     NEXT K
  80 INPUT
     NEXT I
     END
1000 ON ERROR GOTO 1130
1010 DR$ = "DIR/SYS:" + RIGHT$ ( STR$ (DR),1)
1020 OPEN "R",1,DR$
1030 FOR I = 3 TO LOF (1)
1040 GET 1,I
1050 FOR II = 0 TO 7
1060 FIELD 1
     (II * 32) AS D$
     1 AS A$
     4 AS D$
     8 AS FS$
     3 AS FE$
1070 IF NOT ( CVI (A$ + CHR$ (0)) AND 208) = 16
     THEN GOTO 1100
1080 A$(J) = FS$ + " " + FE$
1090 J = J + 1
1100 NEXT II,I
1110 CLOSE
1120 RETURN
1130 RESUME NEXT
1500 CLS
     FOR I = 0 TO J - 1
1520 IF J$ = "J" OR J$ = "j"
     THEN PRINT "KOMMENTAR FÜR "A$(I);
          PRINT "    (Nur bis Ende Bildschirmzeile)
1525 INPUT " ";K$
1530 LPRINT I + 1") "A$(I); TAB( 20);K$
     NEXT
     END
```

Noch einmal: INPUT

Wie Ihr ja inzwischen gemerkt habt, stehe ich mit allem Assembler-Kram auf Kriegsfuß. Da tat die "Entwanzung" des DOS-Königs Arnulf schon not und ich bin auch sehr froh darüber, daß er meinen Artikel mit seinem Wissen richtiggestellt hat. Aber damit die Überwacher des einzig echten und richtigen Assembler-Codes auch weiterhin was zu tun haben, habe ich mich wieder in diesen undurchdringlichen Dschungel begeben und mir in meinem stillen Kämmerlein eine neue Frechheit ausgedacht. Wie Ihr alle Eurem Gedächtnis entnehmen könnt, gibt es in der Clubbibliothek zwei Verbesserungen des INPUT-Befehls. Eine stammt (ich muß es zu meiner Schande gestehen) von mir, die andere ist mir noch nicht vor die Füße gefallen. Ich hoffe also, daß meine dritte Lösung nicht mit der zweiten übereinstimmt. Wenn doch: Sorry! Isvinitje würden die Russen sagen (leider kann meine Textverarbeitung noch kein kyrillisch).
Also: es gibt beim INPUT meines Erachtens zwei Probleme. Erstens kann man die Eingabe nicht auf eine bestimmte Länge beschränken. Zweitens löscht es die folgende Zeile.
Nun gibt es im ROM eine Routine, die INPUT mit benutzt. Sie fängt bei 05D9H an. Sie erwartet zwei Eingangsparameter: Im Register B muß die maximal erlaubte Länge stehen; im Doppelregister HL die Anfangsadresse des Buffers, in den die Eingabe geschrieben werden soll. Außerdem erscheint sie noch auf dem Bildschirm.
Nun schreiben wir uns ein kleines Assembler-Programm mit vier Befehlen:

```
        LD      B,Länge
        LD      HL,Buffer
        CALL    05D9H
        RET
```

Für die Länge setzen wir Beliebiges von 1 bis 255 ein. Der Buffer ist schon ein größeres Problem, aber nicht unüberwindlich. Wir müssen nur wissen, was wir haben wollen, nämlich einen String mit der Eingabe, und die hervorragende Übersetzung von "Basic Faster and Better" unseres Generaldirektors. Da finden wir ein Superfeld mit diversen Erläuterungen, die Ihr da nachschlagen könnt. Das Endergebnis sieht so aus:

```
10 DEFINT A-Z  'alles in INTEGER
20 DIM US(4)   'unser Superfeld
30 A$=STRING$(10,32)  'statt '10' beliebige andere
40 'Länge, die größer als die Eingabe ist
50 X=0
60 PRINT§128,;  'Wo soll die Eingabe beginnen?
70 U(0)=1030:U(1)=8448
80 U(2)=CVI(CHR$(PEEK(VARPTR(A$)+1))+CHR$(PEEK(
VARPTR(A$)+2)))
90 U(3)=-9779:U(4)=-14075
100 DEFUSRO=VARPTR(U(0)): X=USRO(0)
110 IF INSTR(A$,CHR$(13))>0 THEN A$=LEFT$(A$,
INSTR(A$,CHR$(13))-1): PRINT A$
120 END 'RETURN ODER AUSWERTUNG VON A$
```


Bei diesem Beispiel ist die Eingabe auf 4 Zeichen beschränkt. Weiterentwickeln müßt Ihr die Sache schon selbst.
Einige Dinge sind zu beachten:
-erstmal "Basic Faster und Better" ansehen!
-A$="   " geht nicht, weil die Zeile nach der ersten Eingabe ungefähr so aussieht: A$="123
"
Hier spielt das CR eine Rolle, das als letzter Buchstabe in A$ auftaucht. Bei der nächsten Eingabe wäre das Ergebnis falsch, wenn weniger Buchstaben eingegeben würden, denn das "123" bleibt stehen. Also: A$=STRING$(10,32).
-die Eingabe kann auch durch Break abgebrochen oder mit einigen Kontrollcodes verzerrt werden. Vorsicht ist geboten, wenn kein CR in A$ zu finden ist. Dann ist an der Sache was faul.
-wenn nur CR eingegeben wird, sollte die Anwenderroutine dies berücksichtigen. Bei obiger Programmierung bleibt nur A$="" übrig. Keine sehr aussagefähige Eingabe.

Die Vorteile dieser Eingabe habt Ihr natürlich alle schon erfaßt und somit brauche ich mir keine ausdenken. Auf jeden Fall haben wir die Genugtuung, dem alten ROM wieder ein Schnippchen geschlagen zu haben. Ist doch schon was!

Fröhliches Tippen und Ausprobieren wünscht
Gerald Schröder

<u>Anmerkung:</u>
Bei dieser Lösung wird die folgende Zeile noch gelöscht.
Um dies zu verhindern, müßte ein Filter in die Ausgabe 0033H konstruiert werden. Wenn Ihr vorher einen Bildschirmeditor (SEDIT etc.) ladet, wird die folgende Zeile nicht gelöscht!

Adven-80

An Advanced Adventure Development System

For those not familiar with Adventure, it is a game which could simply not have existed before the age of the computer. Imagine reading a story tale where you follow the hero (or heroine) in a fantasy world where anything can happen. The difference between Adventure and a simple story book is that you can be the hero and be instrumental in affecting the outcome of the story, rather than being simply a passive reader.

More specifically, in most Adventures, you are in control of an "entity" which acts as your eyes and ears throughout the story. You can look, listen, manipulate objects, collect treasures, move in whatever direction you want, in fact do anything you want within the confines of a limited universe maintained by the computer.

Conceivably, some day, 3-dimensional tactile dreamworlds will become possible as technology progresses, perhaps coupling computer technology with some form of video disk system; but for the present, Adventure and similar games provide us with the first simple examples of this exciting concept.

This article is probably the first to be published of its kind, in that it provides a complete package for creating Adventures. The code presented here is a *completely general-purpose adventure "interpreter"* together with a BASIC program, designed to allow you, the reader, to create a database for your own Adventures with only the vaguest knowledge of assembly language programming. The code is not intended to be clever; experienced programmers can no doubt quite drastically improve upon the techniques used; however, it does work quite well and I would appreciate any feedback on possible improvements.

In Brief

ADVEN-80 is a program wherein the user resides in one of many LOCATIONS. A LOCATION may have the property of being LIGHT or DARK, and usually is associated with several alternative EXITS. The player moves from one LOCATION to another, and in any LOCATION he may find an OBJECT or OBJECTS. Objects have DESCRIPTIONS, and may optionally have DETAILED DESCRIPTIONS as well. Unless specifically mentioned an object may be TAKEN or DROPPED or LOOKED-AT or otherwise manipulated within the confines of the facilities of the MATCHW table. Each object may only be in one location at any give time (or may be worn or carried as applicable). The object of the game is usually to obtain some form of TREASURE or achieve a SCORE or so many points.

Of special interest is the LINE PARSER, whose job it is to take in whatever the player enters on the keyboard, and attempt to make some sense out of it. In simplified form, the parser in ADVEN-80 takes in a line of user entry, and looks at each word in turn to see if that is a known word. If so, a token entry (somewhat like the one-byte tokens used in some BASICs) is made in a separate buffer; if not, no entry is made, and the next word is checked. When the entire line has been checked over, the token buffer is scanned for words such as "THEN" and "AND" and if found, the line is split, leaving all code after the "AND" for the next time round, in lieu of the player's next entry. This allows the use of multiple statement lines.

Generally, user input breaks down into simple two word sequences after irrelevant words have been discarded. A sentence such as "PLEASE GET THE LAMP AND LIGHT IT" breaks down into two operations, "GET LAMP" and "LIGHT LAMP." The program treats "IT" as the last object referred to. A table is scanned for such word sequences and appropriate action is taken, scores can be maintained, and the program has special facilities to accommodate scoring.

Adventure-type games for microcomputers have been around for some time and several Adventures have been made available for the more popular systems[1] although to date, very few have had anywhere near the capabilities to handle user input as had the original "BIG SYSTEMS" Adventure which many of the major Polytechnic computers have access to.

The Original Adventure for these systems was written in Fortran and required considerably more memory than that available on present-day microcomputers, and the program itself was inflexible (i.e., the creation of a new game would require the re-writing of the Fortran program as the database was an integral part of the overall structure).

ADVEN-80 is a new Micro-Adventure system, written in machine language. Consideration has been given as to obtaining the maximum usage out of the limited memory capacity of currently available small computers.

This Adventure system, written for the Z80 microprocessor, is a full-feature, machine-code Adventure utilizing the powerful functions of the above. It is partly because of this that the game runs at very high speed, with a minimum memory requirement (a simple Adventure could be written in *well* under 16K).

Because of the speed of the Z80, it was possible to use an extremely simple database layout, and the use of an intermediate translator accounts for high readability of the source listing. In reading the following, an elemental knowledge of assembly language programming in Z80 microprocessor is considered useful but not essential.

Tables

OBJLOC The format of this table, containing <object no./location> pairs is simple. The pairs are contiguous with a zero to end of the table. Several location numbers have special meanings and these are listed below.

Loc. 255	Object is non-existent. Cannot be found.
Loc. 254	Object is currently being carried.
Loc. 253	Object is currently being worn.

Finally, location 252 is special in that any object currently in this location is one of a pair of objects which together form a TWIN-STATE object. A typical example of this is a LAMP. The lamp may be LIT or UNLIT and hence it is necessary to have two versions of the same object.

This method is probably the simplest way of implementing a twin-state object, and in the table these two forms should always be concurrent. In the example of the lamp, the two states could be, say, LAMP and ULAMP. If the word LAMP is made accessible, then this must appear FIRST in the table. This is very important, first in ascending order: i.e.:

```
OBJLOC      /* Table name. These are mandatory.              */
LAMP 45     /* object 34, a lit lamp, is in location 45      */
ULAMP 252   /* object 35, an unlit lamp, is hidden from
               view.                                         */
ENDTABLE    /* end of table */
```

As you can see from the above table, if the scanner begins at the start of the table, and then finds a match with object LAMP, in order to blow out the lamp, the two LOCATION entries in the table are simply swapped over.

Object ULAMP was equated only and *must never* be found by the user-input encoding section, although CONDITIONS such as OC or OP can access both versions.

If a user says GET LAMP, and there are no conditions, the system will look for lamp. If this is at location 252 then the pointer will be incremented automatically to point at the ULAMP entry. Hence the user will be given whichever version of the object is *not* at location 252. (Assuming that the other version is PRESENT or being carried or whatever.)

Inventories simply look at the location numbers and if a match is found, then a description is printed. The same applies during the initial description, where objects contained within the current location are described.

OBJDES This table containing <object no./MSG no.> pairs is closely related to the above OBJLOC table. During "I can see . . ." type routines, the table OBJLOC is scanned in order to find objects contained within a given location, be it the current location or those reserved for objects being carried or worn . . .

When the object is found, the number of the object is used to scan through the left side of OBJDES, and when an object match is found, the description number, which refers to an entry in the UMESS table, is used to produce a description of the required object. The OBJDES table need be in no particular order, and no ZERO terminator is required as we are not searching for a non-existent object. In the sample database the ENDTABLE macro is shown, but this is just for neatness, it is not essential here.

OBPROP This table describes the things that can be done to an object, assuming you wish to limit that object to certain actions. I.e., you would not expect the player to be able to TAKE THE STATUE if the statue weighs a ton . . .

```
OBPROP
STATUE      LOOK,LISTEN,FEEL
DOOR        LOCK,UNLOCK,LOOK
ENDTABLE
```

Objects not in the table are considered to have universal attributes (within the limitations of the MATCHW table).

OBDSK This table finds use for those objects you would like to enable a close inspection of. Any object not appearing in the table will be described as ordinary looking.

```
OBJDSK
LAMP        'Looks like an old miner's lamp to me.'
ENDTABLE
```

TABKEY This table is used by the line parser. As the user input line is parsed, 4-byte word sections are put into a temporary buffer. (Words shorter than 4 characters long are left-justified and padded with spaces.) TABKEY is used to replace any known words with a single byte code. Unknown words are scrapped.

TABKEY is composed of the following. The first byte is KEY number. Following that are groups of 4 bytes with the last 4-byte group having the 8th bit set on the last character. If any of the 4-byte groups match that in the buffer, then the KEY byte is put into a buffer for later use. To signify the end of the table, ZERO is used as a KEY byte.

When the sentence has been completely scanned, a buffer called CODBUF will contain the single byte KEY replacements to all the words in the user input that were matched successfully. A ZERO will indicate the end of this buffer. Hence the sequence N "N NORT" will allow user inputs of "N" or "GO NORTH" to be placed in CODBUF as the single byte representing the key N.

A special byte which may end up in CODBUF is THEN. This is currently set at 250 decimal and is used for batching of commands, so that the user may say, for example "GET THE AXE THEN LOOK AT IT." The IT being taken care of automatically by a section of code that converts occurrences of IT into the last mentioned object.

KEYSETS are actually entered in macro form; i.e.:

```
BEGIN       'BEGISTAR'
```

Another example of this is LOOK "LOOKPERUEXAM" where LOOK is the word that will be equated, and will match LOOK, PERUSE and EXAMINE.

MATCHW This table is the one that looks at user input after it has been encoded into a byte sequence. The table is scanned until a pair of bytes corresponding to those in the CODBUF area are found. If there is no match then a default message is printed to indicate that the program found no sensible matches. Otherwise, a set of CONDITIONS is checked, and if all of these are TRUE, then a set of actions is taken.

The format is as follows:

<2 byte key><cond><255><actions><255><either ZERO or more keys>

Where cond = either no entry or any amount of conditions and arguments.

Where actions = any amount of actions.

Obviously from the above, 255 is used as a delimiter and so must not occur within either the conditions or the actions However, 255 may be used as one of the two KEY bytes to tell the program to match up with *any* entry, i.e.:

<23,255>	If CODBUF contains 23 followed by anything then the set will be accepted.
<34,0>	In this case, CODBUF should contain 34 ONLY.

Again, simple equates are used to make life easy. Typical table entries are shown below, together with an explanation of each.

```
MATCH       GET,ANY
TEST        OH,INP
ACTION      TKE,INP,OK
```

In the above example, if the user enters, say, GET LAMP then the system will test to see if the object is in the current location (OH) and if so, will allow the taking of the LAMP and will then print an OK message and exit the table, ready for more user input.

```
MATCH       LOOK,NULL
ACTION      :A,EX
```

If and only if the user enters the single word LOOK, the system will unconditionally Look Around, then EXit the

by Peter D. Scargill

Peter D. Scargill, 24 Percy Park, Tynemouth, Tyne & Wear, Great Britain.

table

```
MATCH     ANY,ANY
ACTION    PRM,37,EX
```

This entry should be put at the end of the MATCHW table. This will match ANY user input, and print message 37 from the MSG table, which would be, for example, "I don't know what you mean."

```
MATCH     ON,LAMP
TEST      OC,MATCH,OC,ULAMP
ACTION    SWP,LAMP,SWP,MATCH,OK
```

On the previous page we have a more complex example. If the user enters ON LAMP or LIGHT THE LAMP, etc., and then only if the MATCH and the ULAMP (unlit lamp) are being carried, then swap the MATCH for its USED partner, and swap the ULAMP with its LIT partner.

Notice that no attempt was made to execute the instruction SWP,ULAMP. This would have been disastrous as the ULAMP is the SECOND entry in the table, and you must only SWAP the FIRST entry (i.e., the main entry), as the command SWP FRED looks for the entry FRED in the OBJLOC table, and swaps its location with that of the *next* entry in the table. That is why the user-accessible version of the object *must* be entered into the table before the alternative form.

Messages

MSG This is a general-purpose message table, containing messages such as "I can not see anything" and "ok.". These messages include directives and all general statements. The format is extremely simple:

0 message sequence 0 message sequence 0

The sequence is that of a zero byte followed by ASCII text until the next zero. The zero then acts as both delimiter and counter. As the maximum message number is known (i.e., there is no search), an end-of-table delimiter is not required, other than the zero at the end of the last message.

NOTE that the first 20 messages must *not* be altered in position as they are used by the system in some cases automatically. There is a limit of 255 messages, but this is far more than will normally be required.

In macro form, simply enter the message number and the message, i.e.:

45 'Hi there, this is a message, isn't it . %'

Note that the standard ' symbol cannot be used *within text* and must be replaced with the alternative ' form. You will have noticed the use of one other special character, the percentage sign. This is used as a CRLF and hence must not be used for its original purpose (see below).

UMESS This table contains descriptions of objects and all general messages that are specific to a particular adventure.

7 'This is another message'

PLEASE NOTE: To make assembly listings more easily readable, CRLF sequence will be sent to the screen on encountering a "%" percentage sign in the text. This of course also saves a little space if you're tight with your bytes. Normally, CRLF's are not required in bulk text as the system automatically inserts these depending on the width of the screen.

It must also be noted here that the use of the character ">" anywhere in a sentence will not produce any output but will serve to destroy any pending BATCHING and so this may be used freely within error messages and any message which requires the user's immediate attention.

A final point concerning messages. It is possible to have conditional sections within messages (or indeed complete conditional messages). Flags 23 to 49 inclusive have been reserved for this purpose. These may be tested for being TRUE or FALSE only, in the following manner. Anywhere within a string, the game writer may insert parentheses as follows:

'This is a bit of {Auntidy } text {B as you can {C easily} see}.'

To explain. If flags 23 (i.e., A), 24 (i.e., B), and 25 (i.e., C), are all ZERO then the output of the message would be "This is a bit of text."; however if one were to set flag 25 then the output would be "This is a bit of untidy text."

If flag 24 was set, then you would get "This is a bit of text as you can see." It is obvious that if flag 25 were also set then the word "easily" would be inserted. Note, however, that this would not apply if flag 24 were reset, whereby all text inside the matching set of parentheses would be ignored.

Only the letters A-Z inclusive and the "/" character may be used after the opening parentheses (A-Z corresponding to flags 24-49) and the maximum NESTING depth is 10, although as many NON-NESTED sets as desired may be put into any section of text. The character "/" may be used *before* any of the alpha chars A-Z to indicate that the flag has *not* been set, i.e., is ZERO. The system flag that indicates that the "/" char is in use is only set and reset on OPENING braces so do not try any fancy multi-level conditional text while using this form.

In this version, two special flags have been reserved. Flags 21 and 22 corresponding to @ and ? are automatically set by the program after every user entry to be always opposite; i.e., if flag 21 is "1" then flag 22 will be "0" and vice-versa, but *randomly* so. Hence it is possible to write the following default message.

44 '{@I simply don't understand you.} {?What ????}'

Whenever this message is called, there is a 50-50 chance of either "What ????" or "I simply don't understand you." being output by the program.

These conditional facilities find their main use in circumstances where you want the program to differentiate between LOOK AT THE GLOBE and, say, LOOK AT THE GLOBE CAREFULLY. Obviously you can have umpteen levels of description.

The facilities may be used anywhere that text is found, either in place descriptions or in object descriptions, or even in default replies. A good use here is to allow for either random or deliberate variations in, say, the default "CANT HELP YOU" message, etc. etc.

Descriptions

LOC Place descriptions begin with zero, a number 255 to 252 to indicate one of the following:

255	Normal place, full exit description
254	Normal place, no exit description
253	Dark place, full exit description (if lit by lamp)
252	Dark place, no exit description

In high level form the following words have the same effect as the numbers above:

NORMAL, NOEXIT, DARK, DARKNO

MESSAGE / KEY / LOCATION bytes, 255 to indicate end of directions, and then textual description, zero being the text terminator.

I.e.:

0 254 02 01 02 255 <TEXT> 0 253 03 32 12 255 <TEXT> 0

In macro form, a typical entry may be as follows:

```
1    NOEXIT,02,N,2
     'I am in a small room with no furniture. There '
     'appears to be nobody around at the moment.%'
```

Again, no end of table required other than the mandatory zero, (or ENDTABLE) as we are not looking for locations which do not exist. In the above example, the first text contains only one sequence of exits, (not printed) MSG2 being the description, i.e., NORTH. This corresponds to a user KEY code of 01, and in this case, if a match for the key is found, the player will be transferred to location number 03.

In the second example, MSG3 prints out SOUTH; if the user typed in GO SOUTH, or SOUTH, or S then this would be converted to KEY 03 and the user would then be transferred to location 12. Meanwhile, the present location description is printed out, and any attempt to LOOK at anything will produce the response "I can't see anything" as the 253 at the start of the sequence dictates that this is a dark location. (Of course, if the user has a LAMP, then that is a different matter. See description of LAMP.)

A special location number is 249 (RTN). When a transfer

Flag Table

This is a 255-byte table numbered 1 to 255, zero not being used, as zero is used as a delimiter. Certain flags have special uses.

1	Dark location indicator . . . If ZERO then dark.
2	Number of items carried.
3	Maximum permissible items to be carried.
4	Counted up once every turn.
5	" "
6	" "
7	Counted down once every turn if not zero.
8	" "
9	" "
10	Counted down if dark, lamp ON, every turn.
11	" "
12	" "
13	Present score.
14	Present location.
15	Store FF or FE (Internal).
16	1 and 10 OR'd together. ZERO means can't see.
17	Lamp on/off. Zero when ON. 1 when OFF.
18	Number of items WORN.
19	Number of items HERE (present location).
20	Store of last location number (automatic).
21-22	Inclusive. For random text control (automatic).
23-48	Inclusive. Text controls A-Z.

Flags 49 to 250 are available to the user.

Achtung:
Zu diesem Artikel gehört noch ein 13 - seitiges Programmlisting.
(Listing besteht aus Maschinensprache und BASIC-Teilen)
Wer sich dafür interessiert melde sich bitte bei der Redaktion. Die "Paperware" wird dann umgehend zugesandt.
Wer schon die entsprechende Software in seiner Programmbibliothek hat, kann das Programm ja den anderen interessierten Clubmitgliedern zu Verfügung stellen.
Vielleicht findet sich auch jemand, der das Listing gerne eintippt.
Bitte meldet Euch, wenn Ihr dazu etwas beizutragen habt.

request to this location is selected, the program will treat this in a special manner by returning the user to the previous location. Hence, you may enter a LARGE BOX and a RTN exit will bring you out, no matter where the box is situated. This avoids the need for an awful lot of location testing.

The macro version of this table makes life very simple as can be seen in the above example. No delimiters are required, and the final description is quite readable.

Assembly Notes

To use these programs you will need access to the following utilities. A Z80 assembler (Zilog mnemonics) with conditional assembly. (If your assembler does not support the CONMSG pseudo-op and/or conditional assembly, miss out the messages and only select the sections of code that you require for your application.)

The BASIC used for the converter is Microsoft Version 5. Peculiarities of this BASIC are few, the most important one is the INSTR. This returns a value of the 1st occurrence of one string or character within another string:

A=INSTR(QS,"R") where QS="ARE", A=2

Only sequential disk files are used, read line by line from the source file into the destination file in a single pass. The program is designed to allow reasonably free format of the source file.

Conditions/Actions

Here are the conditions and actions possible together with explanations for each. Those shown below are all available to the Adventure writer and there is no limit to the amount of conditions and/or actions that you may specify in any set.

Conditions

CL	xx	Current location is xx.
NCL	xx	Current location is NOT xx.
OP	xx	Object xx is in present location or is worn or carried (xx=00 means obtain object number from user input).
OC	xx	Object is being carried.
OW	xx	Object is being worn.
OWC	xx	Object is worn or carried.
OH	xx	Object here but not carried or worn.
ONP	xx	Object is not present.
ONC	xx	Object is not being carried.
ONW	xx	Object is not being worn.
NWC	xx	Object is not worn and not carried.
ONH	xx	Object not here, but MAY be worn or carried.
OE	xx	Object exists at present.
ONE	xx	Object does not exist at present.
@L	xx yy	Object xx is at location yy.
N@L	xx yy	Object is not at location yy.
RND	xx	Random chance 0-254 . . . Where 254 is dead cert.
TF	xx yy	Test flag No. xx for value yy.
NTF	xx yy	Test flag No. xx for NOT value yy.
GE	xx yy	Flag xx greater than or equal to value yy.
LT	xx yy	Flag xx less than value yy.
C2	xx	Compare 2nd word with xx.
C3	xx	Compare 3rd word with xx.
NC2	xx	True if 2nd word is NOT = to xx.
NC3	xx	True if 3rd word is NOT = to xx.

Actions

INV		Print an inventory of carried and worn objects.
TKE	xx	Get object to carry it.
WR	xx	Get object and wear it.
DR	xx	Drop the object.
PR	xx	Print message number xx from UMESS table.
PRM	xx	Print message number xx from MESS table.
DL	xx	Describe location.
SF	xx yy	Set flag xx to value yy.
DSCOB	xx	Describe the object.
SWP	xx	Swap object xx with object xx+1.
MV	xx	Move to location xx.
OK		Say 'OK' and exit table.
QT		Quit.
INF	xx	Increment flag xx.
DEF	xx	Decrement flag xx.
EX		Exit table.
SC		Print current score.
PU	xx yy	Put object xx into location yy.
CR	xx	Create object xx (Place here).
DES	xx	Destroy object xx.
FLIP	xx	Flip flag xx true to Zero and Vice Versa.
LA	xx	Look around at items.
SMA	xx	Set message A to UMESS number xx.
SMB	xx	Set message B to UMESS number xx.
WT	xx	Wait for xx seconds.
PF	xx	Print the value of flag No. xx.
ATF	xx yy	Add yy to the value of flag xx. If >255 then set to 255. Make yy=0.
SFF	xx yy	Subtract yy from flag xx. If makes xx<0 then set xx=0. Make yy=0.
GRUN	xx	Print description, directions etc. & exit.

The above conditions and actions represent all of those implemented in the present version of ADVEN-80.

Flag

This table is simply a collection of pairs of flag numbers and their corresponding initial data values. Entries *must* be contiguous.

35 22 /* means flag 35 is set to value 22 */

Certain flags are controlled automatically by the program and may be viewed only.

TITL1

This final message is that which will be presented at the start of any game. It is comprised of ASCII text, with a ZERO delimiter at the end.

CHR$(any single ASCII number you like, or several to clear screen.)

'any title message'

'blah blah'

ENDTABLE

ADVEN-80 System Overall View

ADVEN-80 was a development of an original BASIC system[6] which relied heavily on disk storage. The BASIC program itself was suggested by a small program called GROW[3] (for those interested, the BASIC used was Microsoft Disk Extended Version 5.1 and the disks, Micropolis Metafloppy).

Anyone who wants to attempt to write their own Adventures in BASIC would be advised to start by reading Greg Hassett's article[8] and for those who want to start in machine code (any processor) and are capable of working from generalized macros, a good starting point would be Ken Reed's program[4].

Due mainly to the desire to re-do the system from scratch in Assembler, and with help from someone already familiar with the design of Adventure systems[4], ADVEN-80 was developed. The present version has several outstanding features, firstly the ability to handle multiple commands (i.e., GET LAMP THEN LIGHT IT). This example shows another feature, the system assumes that when you say "IT" you mean the item mentioned last.

ADVEN-80 is fairly flexible when it comes to entering English, (depending on the database). For example, you may enter GO SOUTH or simply S. You may say GET LAMP or TAKE THAT LAMP. If, for example, the player were to say RUB THE LAMP, a particular effect may occur. If it is necessary to immediately repeat such an action, he may either reenter RUB THE LAMP, or if preferred, he may simply enter AGAIN. The idea for this was taken from a magazine article on ZORK[7]. On long inputs this can save a small amount of key pressing and so was considered to be a worthwhile addition to the system.

Due to a potentially fairly massive vocabulary, the system can make sense out of many sentences, but of course this must not be confused with real intelligence (as anyone actually creating the database could not fail to notice!!). So if you *try* to confuse the program, then you probably will succeed, so be sensible. . . . The present version was written under the M80 macro pre-processor[5] and assembled by ASMB, the Cromemco macro-assembler. The listing shown here has already been run through M80 and is suitable for most Z80 assemblers. (The macro pre-pass accounts for the excess labels.)

Readers who wish to use the programs in this article to create their own Adventures may find database construction made considerably easier by starting off with a very large piece of paper, drawing out all the locations as large "molecules." Connect them together, then make a list of all the interconnections, followed by all the items in the game, and where they are going to be at the start of the game. Finally, work through the game location by location, armed with as many sketches as possible, preferably with a friend known to be a whizz at finding faults in other people's programs . . . really, you won't get it right first time: you'll probably be able to work through a game by yourself with no problems, then along comes young Johnny, totally unaware that you're not *supposed* to try EATING the AXE. Hey, presto, another slight bug in your database discovered! Spend a lot of time thinking out the game, and twice as much time debugging it; but most of all, enjoy yourself.

(LISTING ON PAGE 14)

References

1 Scott Adams. Adventures available for TRS-80, Sorcerer, Apple, etc.

2 Mostek Z80 manual.

3 "Grow" by Jeff Levinsky. *Creative Computing* magazine.

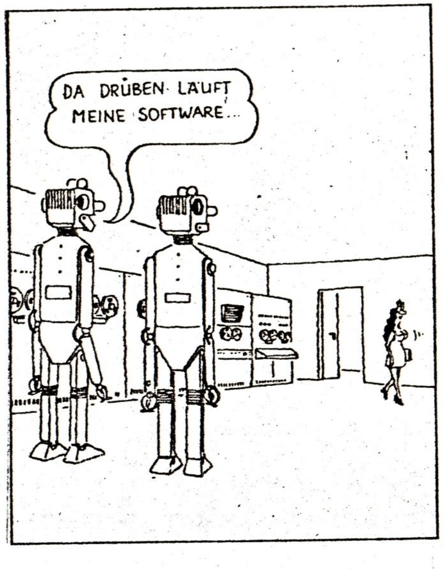
4 "Adventure II" by K. Reed. *Practical Computing* (Aug. 1980).

5 *ML80: A Structured Machine-Oriented Microcomputer Language*. By L. R. B. Pedroso. AD/A-010 055.

6 "Fantasia" by P. Scargill. A BASIC Adventure system. *Creative Computing* (1981).

7 "How To Fit a Large Program Into a Small Machine" by Marc S. Blank and S. W. Galley. *Creative Computing* (July 1980).

8 "How To Write an Adventure" by Gregg Hassett. *Creative Computing* (July 1980).

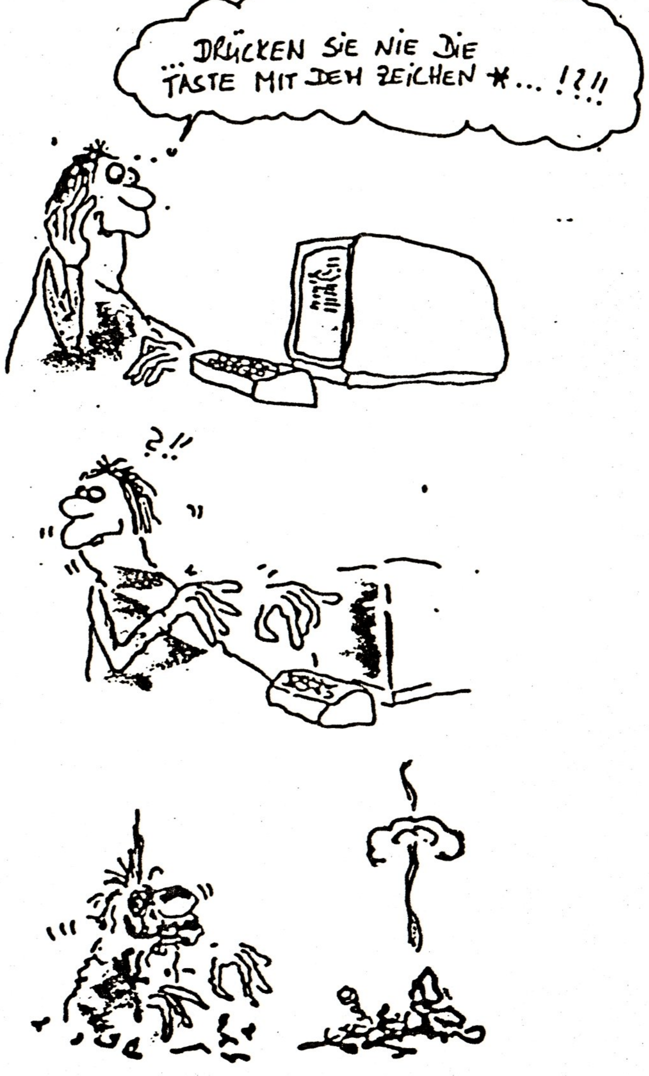


Aus den Memoiren eines Software-Agenten
nacherzählt von Gerald Schröder

DAS PIRATE-ADVENTURE

Als ich erwachte, fand ich mich in einem Londoner Appartment wieder, auf einem Teppich, der an den Boden genagelt war (die Angst vor Diebstahl führt schon zu den komischsten Präventivmaßnahmen). Neben mir lag ein Gegenstand, der mir das Rätsel meiner Anwesenheit in einer mir unbekannten Wohnung scheinbar aufklärte: eine Flasche Rum. Ich war schon immer anfällig für diese Freude des Lebens gewesen und hatte ihren Genuß oft genug mit einem Gedächtnisschwund bezahlen müssen. Doch etwas machte mich stutzig: die Flasche war noch voll! Sollte mich nur vom Geruch des Alkohols eine (wein)selige Bewußtlosigkeit ergriffen haben? Unmöglich, so wenig vertrag ich nun auch nicht!
Da fiel mein Blick auf ein Schild: "Bring die SCHÄTZE hierher und sage: SCORE". Nun war mir alles klar. Ich hatte wieder eine Mission zu erfüllen und war zu diesem Zweck durch Raum (und vielleicht auch Zeit) an diesen Platz gebracht worden. Sofort untersuchte ich in routinierter Weise die Umgebung, denn jeder Hinweis konnte mir helfen, möglichst schnell wieder mein Zuhause zu erreichen und die Lorbeeren meines Erfolges zu genießen. Doch bis dahin war es noch ein langer Weg.
Als erstes fielen mir ein Paar Segeltuchschuhe auf, die sogleich gegen meine eigenen Schuhe eingetauscht wurden, und eine Tüte mit Crackers in die Hände. Ich widerstand der Versuchung, mit Crackers und Rum ein Picknick zu veranstalten, denn diese Gegenstände mochten mir später vielleicht noch von Nutzen sein.
Der einzige Weg aus diesem Zimmer führte über eine Treppe, die ich vorsichtig erklomm, immer auf der Hut vor möglichen Gefahren.
Am Ende der Treppe befand sich ein Alkoven. Einen möglichen Fluchtweg bot ein offenes Fenster, doch als ich herauskletterte, mußte ich feststellen, daß sich unter meinen Füßen nur ein schmaler Sims und zwischen mir und dem Boden mehrere Dutzend Meter Luft befanden. Ich bemühte mich, meine Augen nicht nach unten zu richten, denn ich bin von Natur aus nicht schwindelfrei wie mein Adventure-Kollege James Bond, und kletterte vorsichtig wieder in das Zimmer. Aber in dem Zimmer stand noch ein vollgestellter Bücherschrank. Neugierig nahm ich ein Buch heraus und mit einem Quietschen schwang die Bücherwand beiseite, um einen Geheimgang freizulegen. Ohne das Buch weiter zu beachten, betrat ich den Gang.
Durch den dunklen Gang gelangte man auf einen muffigen Dachboden. Hier fand ich eine Fackel und die Tasche eines Piraten, In welcher sich eine Packung Streichhölzer befanden. Zurück in den Alkoven interessierte mich das Buch. Es war schon ziemlich zerfallen, doch der Titel ließ sich noch mit einiger Mühe entziffern: "Die Schatzinsel". Ein Satz war noch leserlich: "Long John Silver ließ zwei Wertgegenstände auf der Schatzinsel zurück." Nun war meine Mission klar: zuerst mußten die Schatzinsel und die beiden Wertgegenstände gefunden werden. Doch wie sollte man mitten in London eine Insel finden, mal abgesehen von der Verkehrsinsel am Piccadilly Circus?
Am Fenster sah ich mir das Buch noch einmal an und mir lief ein Schauer den Rücken hinunter: auf dem Vorsatzblatt war von zittriger Hand ein Wort mit Blut geschrieben: "YOHO".
Leise murmelte ich das Wort vor mich hin und alles fing an, sich um mich zu drehen. Plötzlich stand ich vor dem Fenster. Verwirrt sagte ich noch einmal "YOHO!" und der Himmel wechselte von grau zu blau, von kaltem Londoner Nebel zu blendender Sonne und [illegible] Brise. Durch die Segeltuchschuhe spürte ich Sand und an mein Ohr drang ein leises Rauschen: Meeresrauschen! Ein Teil dessen, was ich für den Himmel gehalten hatte, war in Wirklichkeit blaues Meerwasser. Vor mir stand ein Schild: "Willkommen auf der Pirateninsel, passen Sie auf die Tide auf!" Aus dem Sand ragten Kiel und Mast eines kleinen Bootes. Scheinbar waren schon vor mir andere Adventurer auf dieser Insel gestrandet und hatten das Schild als Warnung für ihre Nachfolger aufgestellt.
Im seichten Wasser einer Lagune fand ich einen rostigen Anker. Mangels Schwimmerfahrung mußte ich mich jedoch vom offenen Meer fernhalten. Somit blieb nur das Landesinnere, wo ich auf eine Wiese mit einer Grashütte stieß. In der Hütte residierte ein verschlagen dreinschauender Pirat. Um ihn zu besänftigen, bot ich ihm die Buddel Rum an, die er ohne Umschweife nahm und glucksend lachend damit verschwand. Zurück blieben eine Schatzkiste und ein Papagei, der mir viele nützliche Ratschläge geben konnte (wie mir vorher ein Zettel, der in dem Buch lag, schon prophezeit hatte). Allerdings entwickelte der Papagei einen ganz schön kräftigen Appetit und verbrauchte ziemlich viele Cracker.
Mit der Schatztruhe konnte ich vorerst nicht viel anfangen, denn sie war verschlossen. Also zog ich weiter ins Inselinnere und gelangte an einen mit Höhlen übersäten Berg. Im Höhlensystem war eine verschlossene Tür, die von Krokodilen bewacht wurde, Endstation. Eine helfende Stimme riet mir, Futter für die Krokos zu finden, doch der Papagei war ihnen zu mager und somit war ich gezwungen, die Höhlen unverrichteter Dinge wieder zu verlassen.
Ein Pfad führte zum Gipfel des Berges. Dort konnte ich mich durch eine Felsspalte zwängen, wobei ich das große Buch schweren Herzens zurücklassen mußte. In einer großen Höhle, die durch meine Fackel erhellt wurde, fanden sich ein Satz Segel, Bretter, und ein Werkzeugschuppen, in dem ich sofort herumstöberte. So kamen noch eine Schaufel, ein Hammer und ein Schwimmflügel zum Vorschein. Aus der Höhle ließen sich nur die Schwimmflügel und den Hammer mitnehmen, denn alle anderen Sachen waren für den Spalt zu sperrig und der einzige andere Ausgang, eine Tür am Ende eines Ganges, war verschlossen. Wieder an der frischen Luft, löschte ich die Fackel, damit sie nicht so schnell ausbrannte.
Gedanklich hatte ich schon die Verbindung von der Tür in der Höhle zu der von den Krokodilen bewachten hergestellt. Es galt nun, den Schlüssel und das richtige Futter zu finden.
Mit den Schwimmflügeln konnte ich mich endlich auf den Ozean wagen, aber der Fisch, den ich dort fing, entwischte wieder, sobald ich mich dem Land näherte. Ein Behälter mußte her! Doch der einzige erreichbare Behälter war die Schatztruhe, und die war verschlossen.
Entmutigt an den Strand zurückgekehrt, murmelte ich wieder "YOHO" und stand kurz darauf auf dem Sims hoch über dem nebelverhangenen London. Nach einer Klettereinlage durch das Fenster und die Treppe herunter, kam mir der Gedankenblitz: mit dem Hammer ließen sich die Nägel lösen und der Teppich verschieben. Tatsächlich fand sich darunter ein Schlüsselbund, mit dem man die Truhe öffnen konnte. In der Truhe befanden sich zwei Dinge: ein Plan, wie ein Schiff zu bauen war, und eine Karte, die den Weg zur Schatzinsel beschrieb, mit der seltsamen Aufschrift: "30 Schritte, dann graben!"
Die leere Truhe sollte mir beim Fischfang helfen, doch sie leistete mir ebenso schlechte Dienste wie meine Hände. Also kehrte ich erst an den Strand und dann in die Londoner Wohnung zurück. Einem lauten Geräusch folgend, gelangte ich in die staubige Dachkammer und fand dort den schlafenden Piraten mit der nun leeren Rumflasche. Diese ließ ich mitgehen und kehrte in die Fischfanggründe zurück. Nun endlich gelang es: erst das

Wasser und dann den Fisch in die Flasche und schon war es geschafft. Die Krokos freuten sich sehr über die Mahlzeit und verschwanden sofort. Die Tür ließ sich mit dem Schlüssel öffnen und über diesen Weg standen mir die Segel, das Holz und die Schaufel zur Verfügung. In der Lagune mußte der Anker ausgegraben werden, und nun konnte ich endlich alle Utensilien am Strand zusammentragen. Nach ausgiebigem Studium des Plans war es mir möglich, das Schiff zusammenzuzimmern. Doch es fehlte mir noch ein Kapitän!
Der befand sich schlafend in der DachKammer, wie schon erwähnt. Er mußte kräftig geschüttelt werden, doch schließlich erwachte er und wir verabredeten uns am Schiff. Dort wartete er auch schon auf mich. Mein Buch gefiel ihm gar nicht und so ließ ich es widerwillig zurück. Wir erwarteten gemeinsam die Flut und setzten dann die Segel.
Schon nach einem Tag konnten wir an der Schatzinsel ankern. Am Strand probierte ich den Rat der Karte aus: 30 Schritte gehen und graben. Es kam eine Kiste Rum zum Vorschein. Der Pirat verschwand gleich mit einer Flasche.
Auf einem Feld mußte sich die Formel zum zweiten Mal bewähren und bescherte mir eine Holzkiste. In einem Kloster kam mir der Papagei zuhilfe: er verscheuchte die hochgiftigen Mamabas, kurz bevor sie mir den tödlichen Biß verpassen konnten. So konnte ich einen der beiden Schätze mitnehmen: DUBLONEN!
Der zweite Schatz lag vermutlich in der Holzschachtel. Doch um dies zu überprüfen, mußte ich erst zurücksegeln. Mein Kapitän befand sich in der gewohnten Verfassung: betrunken und schlafend. Er war schnell wach zu bekommen und stimmte mir zu, sofort die Segel zu setzen.
Wieder am Strand angelangt, ließ sich die Holzkiste mit dem Hammer und einiger Gewaltanwendung öffnen und der zweite Schatz kam zum Vorschein: die rote und die blaue Mauritius. überglücklich sprang ich in die Luft, nahm das Buch und kehrte in das Londoner Appartment zurück.
Dort bestätigte sich meine Vermutung: alle Schätze waren in meinem Besitz. Nun war es mir vergönnt, den Heimweg anzutreten und zuhause in Ruhe auf den nächsten Auftrag zu warten.



WETTBEWERB

Auch im neuen Jahr mal wieder ein Versuch alle Mitglieder unseres CLUB's anzusprechen. Das sich auch einjeder beteilige an diesem Wettbewerb.

--- Programmierwettbewerb ---

Bei diesem Wettbewerb geht es um folgendes:

Gesucht wird ein ZEHNZEILER IN BASIC

Als Bedingungen sind einzuhalten:
Das Programm muß in BASIC geschrieben sein und darf nicht mehr als 10 Programmzeilen lang sein. Die Funktion, die das Programm ausführen soll bleibt Euch überlassen. Es liegt also an Euch, was das Programm für die anderen CLUB-Mitglieder zu bieten hat. Die Anzahl der Programme, die pro CLUB-Mitglied eingesand werden darf, ist nicht begrenzt. Wichtig ist noch, daß das Programm "auf Eurem Mist gewachsen ist" --auf gut deutsch: eine kreative Eigenproduktion Eurerseits.
Dieses Programm, daß Ihr an die Redaktion (natürlich mit ein paar erklärenden Worten) schickt, wird im jeweiligen folgenden INFO den anderen CLUB-Mitgliedern vorgestellt. Sie dürfen und sollen ausprobieren. Zum Jahresende 86 werden dann Stimmzettel an alle Mitglieder versand und es wird gewählt werden.

Gewählt wird dann DAS PROGRAMM DER CLUB80'ER 1986.

Da für das Programm kein Thema gestellt ist, hat also jeder von Euch seine Chance mitzuwirken. Vielleicht hat man die richtige Idee und fällt unter die Ersten. Ich wünsche es jedem.

Ich hoffe, Ihr lasst mich mit dem WETTBEWERB nicht im Regen stehen. Werft die FLOPPY's an und setzt die CPU unter Spannung. Ich rechne mit Euren Programmen.

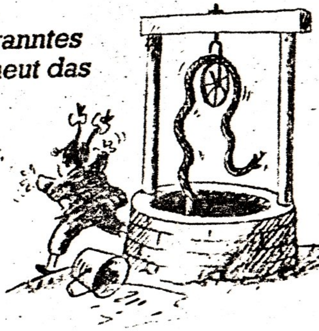
Erfolgreiches programmieren wünscht Euch

Eure Redaktion.

Chinesisch: Yángtóu mài gouròu.
Einen Schafkopf aufhängen und Hundefleisch verkaufen.
Eine schlimme Sache beschönigen.

Yi zhao bèi she yao, san nián pà jing shéng.
Einmal von einer Schlange gebissen, drei Jahre Angst vor dem Brunnenseil.
Ein gebranntes Kind scheut das Feuer.

Rudolf Haselberger, Gröbenzell



„Geschichte schreiben ist eine Art, sich das Vergangene vom Hals zu schaffen.“

Johann Wolfgang von Goethe, deutscher Dichter (1749–1832)

„Es stimmt, daß Geld nicht glücklich macht. Allerdings meint man damit das Geld der anderen.“

George Bernard Shaw, anglo-irischer Dramatiker und Satiriker (1856–1950)

„Gemessen an den bevorstehenden Flugaufgaben ist der Mensch eine Fehlkonstruktion.“

Edward T. Connolly, amerikanischer Testpilot

„Man fällt nicht über seine Fehler. Man fällt immer über seine Feinde, die diese Fehler ausnutzen.“

Kurt Tucholsky, deutscher Schriftsteller (1890–1935)

„Geister erinnern mich an den Männerspruch über Frauen: Du kannst mit ihnen nicht leben, und du kannst nicht ohne sie leben.“

Eugene O'Neill, amerikanischer Dramatiker (1888–1953)

Fehler im NEWDOS-Befehl 'BOOT'

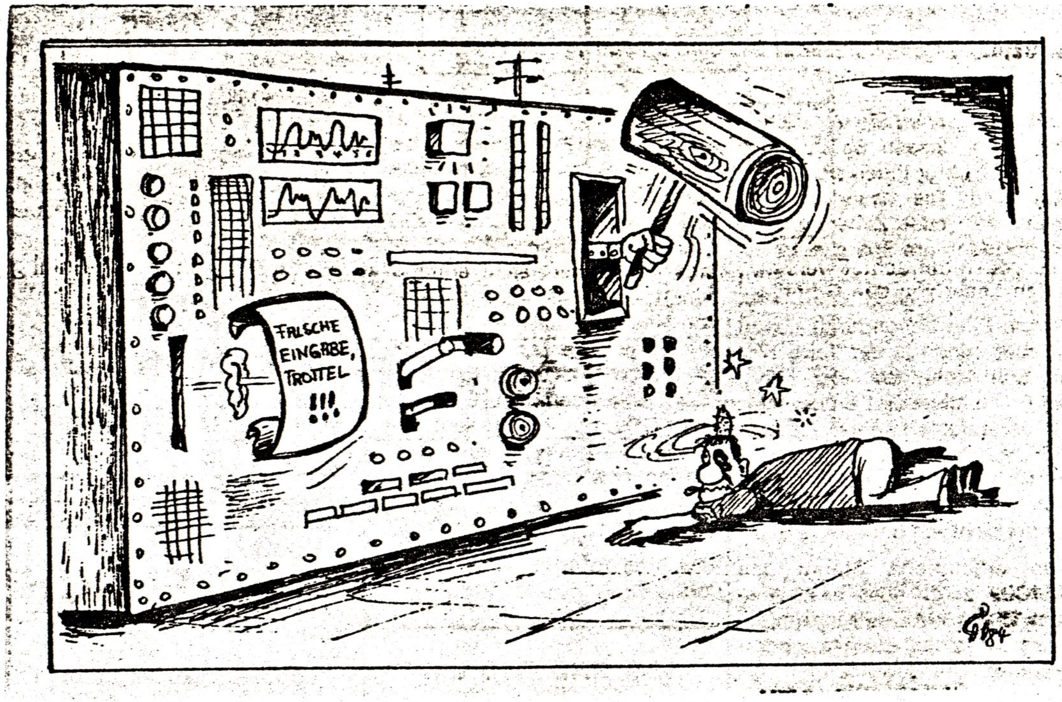
Da das Basteln im DOS in den letzten Club-Info's sehr viel Raum einahm, will auch ich hier einen kleinen Beitrag dazu liefern.
Wie schon aus der Überschrift hervorgeht, geht es um den 'BOOT'-Befehl des NEWDOS-80.
Vor einigen Wochen wurde ich durch einen Bekannten darauf Aufmerksam gemacht, daß in seinem NEWDOS der 'BOOT'-Befehl nicht richtig arbeitet. Ich war ziemlich erstaunt, da ich erst einige Tage vorher mit diesem Befehl gearbeitet hatte.
Bei genaueren Nachforschungen ergab sich aber, das ich mit GDOS und nicht mit NEWDOS gearbeitet hatte. Also wurde das gleiche mit NEWDOS ausprobiert, und siehe da, es ging nicht. Nach Eingabe des 'BOOT'-Befehls hängte sich der Rechner auf oder er landete im Debugger, was ja nun auch nicht unbedingt erwünscht war. Im GDOS funktionierte der 'BOOT'-Befehl allerdings einwandfrei. Also auf gings - "Hinein ins DOS".
Mit Hilfe von Grossers "DOS Buch" war dann sehr schnell die SYS9/SYS als zuständige DOS-Datei ermittelt.
Also wurde als erstes einmal die SYS9/SYS vom GDOS auf das NEWDOS kopiert und der 'BOOT'-Befehl nun nochmals getestet. Und siehe da - es ging. Also mußte das Problem in der SYS9/SYS des NEWDOS liegen.
Nachdem das klar war, kam der Disassembler zum Einsatz. Nach einigen Vergleichen der für den 'BOOT'-Befehl zuständigen Programmteilen kam ich dann zu folgenden Änderungen:

SYS9/SYS des NEWDOS-80

```
rel. Byte 00CA
Inhalt: 3E FE 32 EC 37 AF C3 D0 51

rel. Byte 04E4
Inhalt: 32 E1 37 61 02 00 CD 60 00 00
        00 F3 AF C3 66 00
```

Nach dem letzten geänderten Byte sollten die Bytes 02 02 stehen.
Die alten Inhalte der zu ändernden Bereiche habe ich nicht mit angegeben weil mir im Augenblick leider kein Drucker zur Verfügung steht.
Die relativen Bytes sind mit 'DDE' angegeben. Wer mit Superzap ändern will, muß im relativen Sektor 00, relatives Byte CA und im relativen Sektor 04, relatives Byte E4 die Änderungen vornehmen.
Nach diesen Änderungen arbeitet der 'BOOT'-Befehl auf meinem System einwandfrei.
So, nun aber genug der Theorie, Rechner einschalten und dann haut in die Tasten Leute, das die Bit's und Byte's nur so dampfen.

Bernd Retzlaff



Was ist mit der Primzahl '37' los ?

```
10 CLS:PRINT"  Bildung von Quersummen des 1X1 von 37":PRINT
20 DEFINTF,I,M,Q:DEFSTRS:DEFDBLZ
30 INPUT"Niedrigster und höchster Multiplikator <M1, M2> ";M1,M2
40 FORJ=M1TOM2:Z=J*37:ZZ=Z
50 Z$=STR$(Z):FORI=1TOLEN(Z$):S=MID$(Z$,I,1):Q=Q+VAL(S):NEXT
60 IFQ>9Z=Q:Q=0:GOTO50
70 PRINT"Die kleinste Quersumme des Faktors  "J"        ist ="Q
80 PRINT"Die kleinste Quersumme des Produktes"J" * 37 ist ="Q:Q=0:P
RINT
90 Y$="":Y$=INKEY$:IFY$="",90
100 NEXT:END
```

Hier ist das Programm zum Durchspielen der Quersummen des "Kleinen Einmaleins" mit der ominösen Zahl '37'; blättert nochmal zurück zu meiner "Vorstellung"! Ich hatte dort behauptet, daß jedes Produkt der Zahl 37 mit einem beliebigen Faktor die Eigenschaft habe, daß seine Quersumme gleich der Quersumme des betreffenden Faktors sei. Dazu eine Erläuterung des von mir hier verwendeten Begriffs "Quersumme": Natürlich ist dies zunächst die Summe der Ziffern, aus denen sich eine Zahl zusammensetzt. Wenn diese Summe aber selbst mehrstellig ist, kann man von ihr wieder die Quersumme bilden und so fort, bis man schließlich zu einer einstelligen Zahl kommt. (Diese kann nie Null sein; warum nicht?) Dies nenne ich "die kleinste Quersumme".

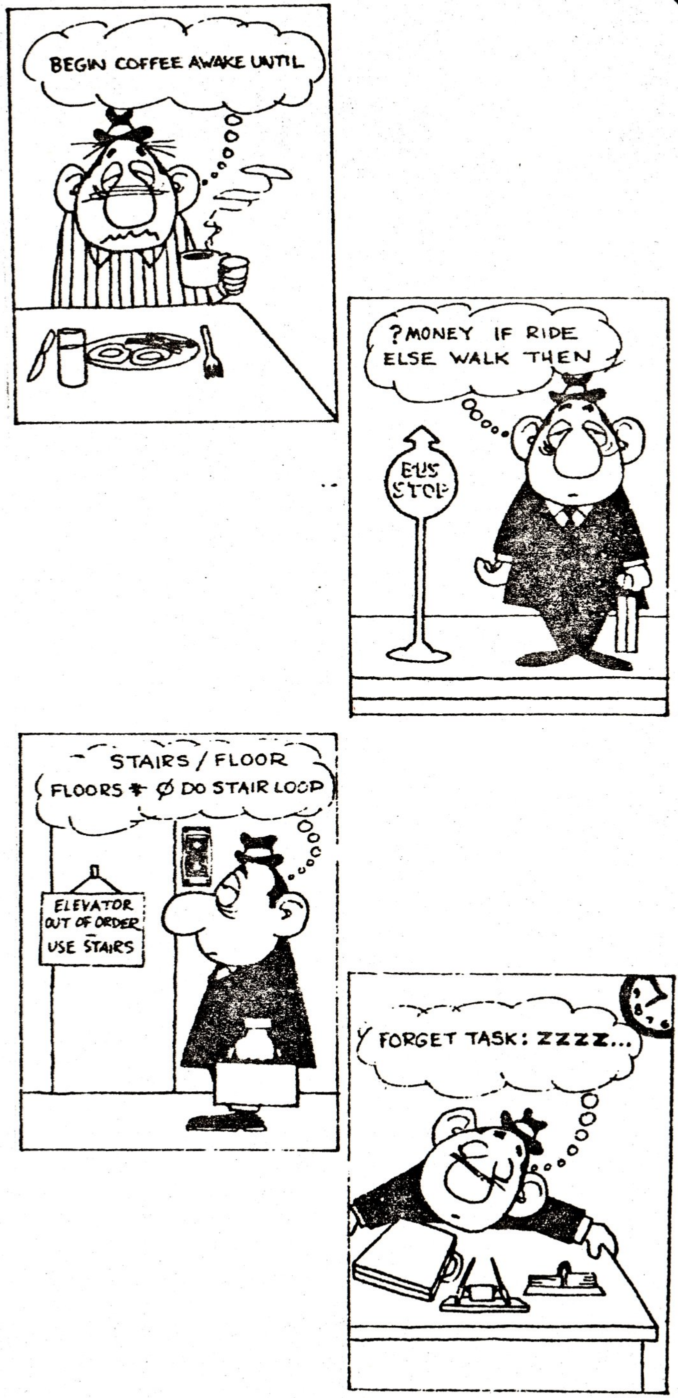
Multipliziert man 37 mit einem beliebigen Faktor ("Multiplikator"), so hat die kleinste Quersumme des Produktes denselben Wert wie die Quersumme des Multiplikators - das war meine tiefsinnige Behauptung.

Gewiß, nichts weiter als eine Zahlenspielerei - keine Mathematik! Außer der "Entdeckerfreude", die dieses Spiel fußhaltet, zeigt das Programmchen dem Anfänger (und der ist immer ein Spieler, sonst würde er wohl nicht mit der Computerei anfangen) erstens, wie sich jede beliebig lange Zahl auf dem Umweg über ihre Stringumwandlung elegant in ihre Einzelteile auflösen läßt, und zweitens, wie eine Rekursion aufgebaut werden kann, um schließlich zum erstrebten Ziel zu gelangen. Das gleiche hätte sich auch rein numerisch ohne Verwendung von Strings erreichen lassen - aber sehr viel langwieriger mit einigen umständlichen Formeln.

Wirklich: ein winziges Beispiel nur. Aber klein fängt man an...

> Euer KaJot <

PS: 'fuß-' (wo doch?) kann natürlich auch durch 'bein-' ersetzt werden



NEWDOS/80+10

Version 2.4
26.10.85

10/12/85   12:00:02

DOS>

Extended NewDOS 80

Erweitern des NewDOS80 mit GDOS-, HDOS- und selbstgestrickten Funktionen

Heute: F(*orm*), DR, LF und OUT

Im letzten Info wurde der "Einbau" der ID-Funktion ins NewDOS80 beschrieben. Außerdem wurde gezeigt, wie man DDE und LWT in unter NewDOS startbare /CMD-Programme umwandelt. Heute nun will ich die Änderungen beschreiben, die nötig sind, um die DOS-Befehle F(orm), DR, LF und OUT im NewDOS zu installieren.

Zuvor jedoch ein paar Bemerkungen zu den Quellen, aus denen die entsprechenden Routinen stammen:

Funktion	Quelle
F	SYS28/SYS aus GDOS 2.2 (dort als FORM im Library!)
DR	SYS28/SYS aus GDOS 2.2
LF	SYS28/SYS aus GDOS 2.2
OUT	SYS29/SYS aus HDOS (Autor Arnulf Sopp, veröffentlicht im Info des Bremerhavener Computerclubs)

Da bei meiner Ur-NewDOS-Version die Systemfile 29 als BASIC-Overlay (LINE und NAME) benutzt wird, mußte OUT im SYS28/SYS Platz finden, was auch keinerlei Schwierigkeiten bereitet!

Die folgenden Zap's setzen voraus, daß die Änderung "Platz im SYS1" durchgeführt und SYS28/SYS an der richtigen Stelle auf der Systemdiskette eröffnet ist!

Besitzer von VideoGenie-Maschinen können die Zap's weglassen, die sich ausschließlich mit der "memory-mapped" (und damit TRS80-spezifischen) Druckersteuerung befassen. Diese Änderungen sind mit ☞TRS80 gekennzeichnet!

Nun aber frohen Mutes an die Arbeit!*
Viel Spaß und Erfolg wünscht Euch Euer

Hartmut Obermann

* Auf alle Fälle nur an einem Backup arbeiten!!!

Zuallererst muß man sich natürlich SYS28/SYS von einer G- oder H-DOS -Diskette in das zwar schon eröffnete aber noch leere Systemfile auf seiner neuen Systemdiskette kopieren. Danach sind die folgenden Änderungen (Format: File,File relativ Sector, Sector relativ Byte, Änderung) vorzunehmen:

☞TRS -> alle Änderungen mit diesem Zeichen dienen der Druckersteuerung bei TRS 80-Rechnern und brauchen von VideoGenie-Besitzern nicht durchgeführt werden!

☞TRS
SYS28/SYS,0,4F
ändere DB FD F6 0F in CD 11 4F 00

Folgende Änderung behebt einen Fehler (falscher relativer Sprung!), und man sollte sie auch auf HDOS- und GDOS-Originaldisketten durchführen!!!
SYS28/SYS,0,6D
ändere E2 in E1

☞TRS
SYS28/SYS,2,1D
ändere 00 00 00 00 00 00 in 3A E8 37 F6 0F C9

☞TRS
SYS28/SYS,2,57
ändere D3 FD 00 in 32 E8 37

☞TRS
SYS28/SYS,2,87
ändere D3 FD 00 in 32 E8 37

☞TRS
SYS28/SYS,2,C1
ändere D3 FD 00 in 32 E8 37

☞TRS
SYS28/SYS,2,DF
ändere DB FD 00 in 3A E8 37

Die nächste Änderung betrifft den Druckercode für 6 bzw. 8 Zeilen pro Zeile. Dieser ist natürlich vom Drucker abhängig und wird hier für EPSON RX80 angegeben!
SYS28/SYS,4,6A
ändere 1B 41 03 00 1B 42 03 00 in 1B 32 0D 00 1B 30 0D 00
Die alten Codes gelten, soweit ich weiß, für ITOH-Drucker!

Damit es nicht nur bei den recht trockenen Zaps bleibt, hier eine kurze Erklärung der Assembleranweisungen:
wo früher

```
    D3 FD    = OUT (FD),A   ' Inhalt des Accus auf Port
                            ' FD ausgeben
    00       = NOP          ' keine Operation
stand, haben wir
    32 E8 37 = LD (37E8),A  ' Inhalt des Accus in Speicher-
                            ' zelle 37E8 laden
```

eingetragen. Wir haben aus einer portorientierten Ein/Ausgabe eine speicherorientierte Ein/Ausgabe gemacht (wie sie der TRS-80 verwendet!). Entsprechend wurde die Befehlsfolge IN A,(FD) [DB FD] in LD A,(37E8) [3A E8 37] geändert!

Wer mehr wissen möchte, kann sich die Zaps ja mal mit dem Disassembler anschauen!

Die bis hierher durchgeführten Änderungen betreffen nur die Befehle DR, F und LF. Der OUT-Befehl kommt etwas später an die Reihe. Zuvor will ich noch kurz auf die Parametertabelle des F(orm)-Befehls eingehen.

Aus dem GDOS-Handbuch:
FORM, parameterkette

FORM erlaubt die Ausgabe von besonders definierten Zeichenketten an den Drucker. Da die Möglichkeiten der Drucker recht unterschiedlich sind, wurde wie folgt verfahren.

Die Parameter A bis P bezeichnen 16 Zeichenketten von jeweils 15 Zeichen und einem Endzeichen, die vom Anwender selbst definiert werden können. Das Endzeichen muß entweder 03h (kein Wagenrücklauf) oder 0Dh (Wagenrücklauf) sein.

Ein Punkt gibt ebenfalls 0Dh an den Drucker aus.

X erlaubt die Ausgabe einer bis zu 255, Y einer bis zu 123 Zeichen langen, vom Anwender definierten Zeichenkette.

Die Zeichenketten für die Parameter A-P können mit DDE in SYS28/SYS Sector 1 relativ, für X in BOOT/SYS (GDOS- oder HDOS/SYS) Sector 4 relativ und für Y in SYS28/SYS Sector 3 relativ geändert werden!

Hier nun ein Hardcopy des relativen Sectors 1 von SYS28/SYS. Wie man sieht, sind die Parameter A-E belegt, während F-P noch frei sind. Die in A-E eingetragenen Werte gelten für EPSON RX80 - Drucker und bedeuten:
A = Drucker neu initialisieren
B = Schmalschrift, linker Rand 10, rechter Rand 122
C = Schmalschrift
D = linker Rand 10
E = rechter Rand 122

```
000100: 1B40 0D20 2020 2020 2020 2020 2020 2020 .§.
000110: 1B0F 1B6C 0A1B 517A 0D20 2020 2020 2020 ...l..Qz.
000120: 1B0F 0D20 2020 2020 2020 2020 2020 2020 ...
000130: 1B6C 0A0D 2020 2020 2020 2020 2020 2020 .l..
000140: 1B51 7A0D 2020 2020 2020 2020 2020 2020 .Qz.
000150: 4646 4646 2C46 4646 462C 4646 4646 2C03 FFFF,FFFF,FFFF,.
000160: 4747 4747 2C47 4747 472C 4747 4747 2C03 GGGG,GGGG,GGGG,.
000170: 4848 4848 2C48 4848 482C 4848 4848 2C03 HHHH,HHHH,HHHH,.
000180: 4949 4949 2C49 4949 492C 4949 4949 2C03 IIII,IIII,IIII,.
000190: 4A4A 4A4A 2C4A 4A4A 4A2C 4A4A 4A4A 2C03 JJJJ,JJJJ,JJJJ,.
0001A0: 4B4B 4B4B 2C4B 4B4B 4B2C 4B4B 4B4B 2C03 KKKK,KKKK,KKKK,.
0001B0: 4C4C 4C4C 2C4C 4C4C 4C2C 4C4C 4C4C 2C03 LLLL,LLLL,LLLL,.
0001C0: 4D4D 4D4D 2C4D 4D4D 4D2C 4D4D 4D4D 2C03 MMMM,MMMM,MMMM,.
0001D0: 4E4E 4E4E 2C4E 4E4E 4E2C 4E4E 4E4E 2C03 NNNN,NNNN,NNNN,.
0001E0: 4F4F 4F4F 2C4F 4F4F 4F2C 4F4F 4F4F 2C03 OOOO,OOOO,OOOO,.
0001F0: 5050 5050 2C50 5050 502C 5050 5050 2C03 PPPP,PPPP,PPPP,.
```

Natürlich kann und soll sich jeder Anwender für seinen Drucker seine eigenen Codes zusammenstricken. Die hier aufgeführten Funktionen dienen nur der Demonstration!

Nun aber zu dem ersten wirklich neuen Befehl, zu

OUT,port#,xx,yy--- !

Wie schon zu Beginn dieses Artikels gesagt, entsprang der Befehl "OUT,port#,xx,yy,..." den genialen Gehirnwindungen unseres allseits bekannten(!) und beliebten(!?) Clubmitglieds Arnulf Sopp, von dem böse Zungen behaupten, er würde sogar Telefonnummern hexadezimal angeben (was eine reine Verleumdnung ist)!

Aber Scherz beiseite, diese Funktion ist wirklich nützlich und gehört deshalb in "unser" NewDOS80 +10. Mit ihr kann man z.B. den HRG-Bildschirm abschalten (OUT,00,00) oder, falls man ein VideoGenie besitzt, Codes an den Drucker ausgeben. Wie die Routine funktioniert, durchschaut Ihr (hoffentlich) anhand der Kommentare im Listing unseres Assemblergroßmeisters!

Ich möchte hier nur kurz auf zwei Unterschiede zum Original (erschienen im Info des Bremerhavener Computerclubs) hinweisen:

1. Die ORG-Adresse (Zeile 100) mußte geändert werden, da die Routine bei uns ja in einem anderen SYS-File Platz finden muß.
2. In Zeile 250 mußte aus CALL 4454h ein CALL 4CD5h gemacht werden. Die Speicherzellen 4454h-4456h, die im GDOS 2.2 ein JP 4CD5 (C3 D5 4C) enthalten sind im NewDOS mit NOP's (00 00 00) gefüllt. Durch diese Änderung tritt keinerlei Funktionsänderung ein. Warum und wozu im GDOS 2.2 der Ansprungpunkt geändert wurde, ist mir nicht bekannt!

Im nebenstehenden Sectordump des rel. Sector 4 von SYS28 sind die neu einzutragenden Codes unterstrichen. Vorher waren dort überall NOP's (00) eingetragen.
Zusätzlich muß im rel. Sector 0 von SYS28/SYS, rel. Byte 1A noch die Einsprungadresse der Routine abgelegt werden. Dort wird 7351 statt 4B4D eingetragen!

Nun brauchen wir nur noch die neuen Befehle ins Library einzutragen. Dazu hängen wir an unsere Befehlstabelle folgende Codes an:

```
44 52 82 FE 00     (für DR)
46 88 FE 00        (für F)
4C 46 81 FE 00     (für LF)
4F 55 54 87 FE 00  (für OUT)
```

Das war schon alles. Eigentlich nur eine Sache von ein paar Minuten, und schon hat man vier neue DOS-Befehle zur Verfügung!

Beim nächsten mal kommen dann die Befehle P(ort), I(nfo) und DISK dran. Bis dahin, viel Spaß beim Zappen, Euer

Hartmut Obermann

```
                 00010 ;****************************************
                 00011 ;*                                      *
                 00012 ;*   OUT,port#,xx,yy,....               *
                 00013 ;*   (C) '84 by Arnulf Sopp             *
                 00014 ;*                                      *
                 00015 ;****************************************
5173             00100          ORG     5173H   ;Beginn der Routine
5173 CD7E51      00110 OUT      CALL    GETVAL  ;Hexzahl einlesen
5176 4F          00120          LD      C,A     ;Portnummer in C laden
5177 CD7E51      00130 LOOP     CALL    GETVAL  ;Hexzahl einlesen
517A ED79        00140          OUT     (C),A   ;auf Port ausgeben
517C 18F9        00150          JR      LOOP    ;bis zum bitteren Ende
517E CD8B51      00160 GETVAL   CALL    GETCHR  ;eine Hexzahl einlesen
5181 17          00170          RLA             ;Stellenwert korrigieren
5182 17          00180          RLA             ;d. h. in linkes Nibble
5183 17          00190          RLA             ;schieben
5184 17          00200          RLA             ;(4 Bits nach links)
5185 57          00210          LD      D,A     ;linke Hexziffer merken
5186 CD8B51      00220          CALL    GETCHR  ;naechste Hexziffer einlesen
5189 B2          00230          OR      D       ;LSN mit MSN vereinigen
518A C9          00240          RET             ;jetzt der korrekte Wert in Accu
518B CDD54C      00250 GETCHR   CALL    4CD5H   ;Trennzeichen und CR erkennen
518E 2817        00260          JR      Z,EXIT  ;Ende bei NEW LINE
5190 7E          00270          LD      A,(HL)  ;Hexziffer laden
5191 FE30        00280          CP      '0'     ;Dezimalziffer?
5193 3816        00290          JR      C,ERROR ;falls ASCII < Dezimalziffer
5195 FE3A        00300          CP      ':'     ;> ASCII "9" ?
5197 380A        00310          JR      C,RETURN;falls korrekte Dezimalziffer
5199 FE41        00320          CP      'A'     ;Hexziffer > ASCII "9" ?
519B 380E        00330          JR      C,ERROR ;falls < ASCII "A"
519D FE47        00340          CP      'G'     ;> ASCII "F" ?
519F 300A        00350          JR      NC,ERROR;falls falsche Eingabe
51A1 D607        00360          SUB     7       ;falls Alpha-Hexziffer
51A3 E60F        00370 RETURN   AND     0FH     ;ASCII  nach binaer umwandeln
51A5 23          00380          INC     HL      ;naechste Bildschirmstelle
51A6 C9          00390          RET             ;erledigt
51A7 F1          00400 EXIT     POP     AF      ;Stack korrigieren
51A8 F1          00410          POP     AF      ;(2. CALL-Ebene)
51A9 AF          00420          XOR     A       ;Z-Flag fuer "kein Fehler"
51AA C9          00430          RET             ;ins Betriebssystem
51AB F1          00440 ERROR    POP     AF      ;Stack korr. (s.o.)
51AC F1          00450          POP     AF
51AD 3E2F        00460          LD      A,2FH   ;Fehlercode "falsche Parameter"
51AF B7          00470          OR      A       ;Z-Flag ruecksetzen
51B0 C9          00480          RET             ;ins Betriebssystem
002F             00490          END             ;wohlverdient
00000  Fehler
33448  Zeichen verfügbar

ERROR  51AB 00440    00290 00330 00350
EXIT   51A7 00400    00260
GETCHR 518B 00250    00160 00220
GETVAL 517E 00160    00110 00130
LOOP   5177 00130    00150
OUT    5173 00110
RETURN 51A3 00370    00310
```

```
000400: 01FA F050 03E5 7EFE 41D2 3E51 0620 211D ...P..B.A.>Q. !.
000410: 51BE 280C 2310 FAE1 FE0D C83E 34C3 0944 Q.(.#.......>4..D
000420: 7821 EC4F 0707 1600 5F19 CD6A 44E1 2318 x!.O...._..jD.#.
000430: D23F 3E3D 3C3B 3A39 3837 3635 3433 3231 .?>=<;:987654321
000440: 302F 2E2D 2C2B 2A29 2827 2625 2423 2221 0/.-,+*)('&%$#"!
000450: 2000 FE51 3023 D640 1110 0047 21E8 4D19  ..Q0#.@...G!.M.
000460: 10FD C316 5121 0042 18FB 1B32 0D00 1B30 ....Q!.B...2...0
000470: 0D00 1B07 0300 1B0A 03FE 5828 E8FE 5921 ..........X(..Y!
000480: 7050 28DE E118 94CD 7E51 4FCD 7E51 ED79 pP(.....BQO.BQ.y
000490: 18F9 CD8B 5117 1717 1757 CD8B 51B2 C9CD ....Q....W..Q...
0004A0: D54C 2817 7EFE 3038 16FE 3A38 0AFE 4138 .L(.B.08..:8..A8
0004B0: 0EFE 4730 0AD6 07E6 0F23 C9F1 F1AF C9F1 ..G0.....#......
0004C0: F13E 2FB7 C900 0000 0000 0000 0000 0000 .>/.............
0004D0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0004F0: 4D30 3220 2E2E 2E2E 2E2E 2E2E 0202 004D M02 ...........M
```


Hier eine Kurzübersicht über die besprochenen, neuen Befehle!

DR <,$,#><text>
DR ermoeglicht die direkte Ausgabe von Text und Sonderzeichen an den Drucker. Der Text kann max. 76 Zeichen lang sein und wird in Grossbuchstaben umgewandelt.
$ = Escape (= ASCII 27)
# = Umwandlung des folgenden Zeichens in einen Controlcode (z.B.: N=16)

F,parameterkette
F entspricht dem GDOS-Befehl FORM. Er erlaubt die Ausgabe besonders definierter Zeichenketten an den Drucker.
Die Parameter A-P bezeichnen 16 Zeichenketten zu je 15 Zeichen.
X bezeichnet eine 255, Y eine 128 Zeichen lange Zeichenkette.
Die Zeichenketten koennen vom Anwender selbst definiert werden.
A-P stehen im Sector 2, Y im Sector 4 des SYS28/SYS.
X steht im BOOT/SYS Sector 4.

LF <,8 ,6>
LF,8 setzt die Zeilenzahl fuer den Drucker auf 8 / Zoll
LF,6 setzt die Zeilenzahl fuer den Drucker auf 6 / Zoll
Dabei wird der interne Zeilenzaehler jeweils auf 0 gesetzt.
LF ohne Parameter aktiviert das Unterprogramm zur Positionierung des Druckers.
Pfeil oben = Zeilenvorschub bis Seitenende
Pfeil unten = Zeilenvorschub (Taste festhalten)
Pfeil rechts = Zeilenvorschub
Pfeil links = negativer Zeilenvorschub (je nach Drucker!!)

OUT,##,xx,yy,..
= Nummer des angesprochenen Ports
xx,yy,.. = Werte, die an den Port ausgegeben werden sollen
Die Werte muessen hexadezimal angegeben werden. Die Anzahl der Bytes, die auf einmal abgesendet werden kann, wird von der Groesse des DOS-Eingabebuffers bestimmt (norm. 80 Zeichen+CR).

ID,#<,A>
Mit ID werden die PDRIVE-Parameter unbekannter Disketten erkannt und auf Diskette geschrieben.
Wird ,A angehaengt werden die erkannten Parameter sofort aktiv.

Noch ein Zap!

Wenn man versucht, ein Programm auf einer Diskette zu löschen, welche schreibgeschützt ist, so bekommt man die Fehlermeldung "Directory Write Error". Die gleiche Meldung bekommt man auch wenn man versucht, ein File auf eine schreibgeschützte Diskette zu kopieren.

Wem das nicht gefällt, kann mit folgender Änderung Abhilfe schaffen:
SYS0/SYS,6,2D ändere 12 in 0F

Danach bekommt man in oben genannten Fällen die Fehlermeldung "Write Protected Diskette"

Hartmut Obermann

Extended BASIC

RESTORE n und LPRINT alles für's NewDOS80 +10 - BASIC

Wer meint, er bräuchte nie einen RESTORE n-Befehl und wen nicht interessiert, daß der Druckertreiber des BASIC eine Menge Druckersteuerzeichen verschluckt oder verhunzt, der braucht den folgenden Abschnitt nicht zu lesen. Wen es aber danach gelüstet, seinem Drucker einmal all das zukommen zu lassen, was er normalerweise nicht bekommt, oder zu sehen, wie man eine neue Funktion in einen schon bestehenden Befehl einbauen kann, der sollte aufmerksam weiterlesen!

Wie schon gesagt, es geht um den BASIC-Befehl RESTORE n und um einen Druckertreiber, der alle ASCII-Zeichen ungefiltert an den angeschlossenen Drucker weitergibt. Die beiden Routinen sind schon zur Genüge erklärt worden (RESTORE n im Info Nr. 5, der Druckertreiber in den Info's Nr. 6, 8 und 9). Der Vollständigkeit halber sind hier trotzdem die beiden recht kurzen Assemblerlistings (allerdings kommentarlos) abgedruckt!

Die RESTORE n - Routine

```
6669    E3        EX      (SP),HL
666A    7C        LD      A,H
666B    FE1D      CP      1DH
666D    E3        EX      (SP),HL
666E    C2781D    JP      NZ,1D78H
6671    D7        RST     10H
6672    FE90      CP      90H
6674    C2791D    JP      NZ,1D79H
6677    CD5B1E    CALL    11E5BH
667A    E5        PUSH    HL
667B    CD2C1B    CALL    1B2CH
667E    0B        DEC     BC
667F    ED43FF40  LD      (40FFH),BC
6683    E1        POP     HL
6684    C1        POP     BC
6685    C31E1D    JP      1D1EH
```

Die LPRINT alles - Routine

```
6688    211100    LD      HL,0011H
668B    39        ADD     HL,SP
668C    7E        LD      A,(HL)
668D    C3B405    JP      05B4H
```

Während die RESTORE n - Routine ständig aktiv ist, ist das Druckertreiberteil nur dann verfügbar, wenn die Treiberadresse im Drucker-DCB (4026h und 4027h enthalten normalerweise 8D 05) auf die Routine zeigt. Man kann dies natürlich durch die BASIC-Befehlsfolge
POKE 16422,136 : POKE 16423,102
tun.
Diese Methode ist allerdings wenig elegant. Eine bessere Möglichkeit wird gleich beschrieben, zuvor jedoch ein Hexdump des relativen Sector 15 von BASIC/CMD. Die durchzuführenden Änderungen sind unterstrichen!
Wer eine andere BASIC-Version sein eigen nennt, sollte nicht einfach drauflosändern, sondern sich mit mir in Verbindung setzen!

BASIC/CMD

```
DRV 00  DC5D 210D 57E5 CDDE 5CD1 3600 2100 42C1  ü!MW ö 6§!§B
0   10  0000 0000 0000 0000 0600 CD24 4428 07FE  §§§§§§§§F§ $D(G
0H  20  18C2 165E 1815 2141 66CD 6744 CD84 03FE  X V^XU!Af gD C
    30  4A20 0CCD 3A03 3E0D CD3A 03E1 C39F 6021  J L :C>M :C  `!
DRS 40  6466 CD67 443E 74C3 055E 416C 7465 6E20  df gD>t E^Alten
35  50  496E 6861 6C74 2075 6562 6572 7363 6872  Inhalt ueberschr
23H 60  6569 6265 6E20 284A 2920 3F20 034E 6569  eiben (J) ? CNei
    70  6E0D E37C FE1D E3C2 7B1D D7FE 90C2 791D  nM ö ü  xü   yü
    80  CD5B 1EE5 CD2C 1B0B ED43 FF40 E1C1 C31E   Ä^ ,ÄK C§
    90  1D21 1100 397E C3B4 0500 0000 0000 0000  ü!Q§9ß E§§§§§§§
    A0  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
    B0  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
FRS C0  0000 0000 0000 0000 0000 0104 0440 6966  §§§§§§§§§§ADD§if
15  D0  012A C766 0000 0000 0000 0000 0000 0000  A* f§§§§§§§§§§§§
FH  E0  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
    F0  0000 0000 0000 0000 0000 0000 0100 BC66  §§§§§§§§§§§§A§ f
```

SYS27/SYS

```
DRV 00  01FA 004D FEFD 201B 0D28 2F0D CA44 4F0D  A §M  ÄM(/M DOM
0   10  CA88 4F0D CA01 4F0D CA06 4F0D CA09 4F0D   OM ADM FOM IOM
0H  20  CA9E 513E 2AB7 C921 1645 2216 40C9 218D   Q>* !VE"V§ !
    30  0522 2640 C900 0000 0000 CDD5 4C28 3600  E"&§ §§§§§ L(6§
DRS 40  00E5 CD47 4D00 E123 28F0 C97E FE54 28D7  § GM§ #( ß T(
255 50  FE4D 2814 FE44 28D6 FE5A 2806 0000 00C3   M(T D( Z(F§§§
FFH 60  D04D 3E48 3228 40C9 2105 4522 1E40 C930   M>H2(§ !EE"^§ 0
    70  3132 3334 3521 244E ED5B 1640 CD63 4021  12345!$N ÄV§ c§!
    80  304E ED5B 1540 CD67 4021 5B4E ED5B 1E40  ON ÄU§ g§!ÄN Ä^§
    90  CD63 4021 674E ED5B 1D40 CD67 4021 8D4E   c§!gN ÄU§ g§! N
    A0  ED5B 2640 CD63 4021 994E ED5B 2540 CD67   Ä&§ c§! N Ä%§ g
    B0  40ED 5B28 4021 394E CD67 405A 2170 4ECD  § Ä(§!9N g§Z!pN
FRS C0  6740 ED5B 4940 21A8 4ECD 6340 21F8 4DCD  g§ ÄI§! N c§! M
0   D0  6744 1810 FE47 CA18 4F3E 34B7 C309 4400  gDXP G XO>4 ID§
0H  E0  0000 0000 2A20 40AF 0100 0177 3C23 000B  §§§§* § A§Aw<#§K
    F0  10F9 2220 40CD 4B4F C900 0000 0102 FB4D  P " § KO §§§AB M
```

```
DRV 00  01FA FB4E 0000 0000 0000 0000 003E 0BD3  A N§§§§§§§§§>H
0   10  FFC9 AF18 FACD D54C 7EFE 5828 F5FE 5928   X  LB X(  Y(
0H  20  EC3E 34C9 1188 661A FE21 2005 ED53 2640   >4 Q fZ ! E S&§
    30  C93E 0BB7 C900 0000 0000 0000 0000 0000   >H §§§§§§§§§§
DRS 40  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
257 50  0E00 7900 0000 00ED 78FE FFC4 5C4F 0C20  N§y§§§§ x  öOL
101H60  F621 874F CD67 44C9 5FC5 2181 4FCD 6740   ! O gD _ ! O g§
    70  217A 4FC1 C579 5FCD 6740 2176 4FCD 6744  !zO y_ g§!vO gD
    80  C1C9 494E 5028 3030 4829 203D 2030 3048    INP(00H) = 00H
    90  2020 030D CDD5 4C3A 7A47 3299 4F7E E607    CM L:zG2 OB G
    A0  B7CC 3F51 FE00 3CD2 D04F 4723 7EFE 3D20   ?Q §< OG#B =
    B0  2F23 7EFE 4038 29E6 1FFE 1A28 2730 214F  /#B §8)  Z('0!O
FRS C0  E521 6743 3E0A 5F16 0019 10FD E521 E64F   !gC>J_V§YF  ! O
2   D0  4119 10FD D106 004F EDB0 1810 3E20 B7C9  AYP F§O  XF>
2H  E0  3E34 B7C9 AFC9 0000 0000 AFC9 E123 CDD5  >4  §§§§  #
    F0  4C20 A4AF C9CA C8CC A0BB B200 0102 F04F  L          §AB O
```

NEWDOS/80 + 10

Version 2.4
26.10.85

20/12/85 17:00:02

DOS>

Extended NewDOS 80

Erweitern des NewDOS80 mit GDOS-, HDOS- und selbstgestrickten Funktionen

Hier: I(nfo), P(ort), und DISK

Eigentlich sollte dieser Teil der NewDOS-Modifikation ja erst im nächsten Clubinfo erscheinen. Durch eine schwere Mandelentzündung und die danach fällige Erholungspause (ich erhole mich nirgens besser als am Computer) bin ich jedoch in der Lage, die Fortsetzung schon in diesem Info zu bringen.

Quellenhinweis:
Die in dieser Folge vorgestellten Befehle sind der G-DOS Version 2.2 entnommen (dort SYS29/SYS). Die gleich mitgelieferte BASIC-Erweiterung entspringt meinem Artikel "BASIC-Patch für RESTORE n" und der Artikelserie "LPRINT alles - einfacher", die von Arnulf Sopp und meiner Wenigkeit gemeinsam verbrochen wurde.

Jetzt aber ohne weitere Vorrede ans Werk!

Nur ein Byte

ist im SYS29/SYS von G-DOS 2.2 zu ändern, um es im Extended NewDOS80 +10 als SYS27/SYS verwenden zu können!
Zuvor muß man natürlich SYS27/SYS an der richtigen Stelle auf der Systemdiskette eröffnet und SYS29/SYS von einer G-DOS-Systemdisk auf das noch leere SYS27/SYS kopiert haben. Dann braucht man nur noch im relativen Sector 0 von SYS27/SYS das relative Byte 5 von FF in FD zu ändern und im SYS1/SYS die Befehle I, P und DISK mit ihren Aufruf- und Sprungbytes einzutragen:
49 81 FD 00 (für I)
50 82 FD 00 (für P) und
44 49 53 4B 83 FD (für DISK)
Schon hat man diese Befehle verfügbar!

Im neuen G-DOS 2.4 (für VideoGenie IIIs) ist der I(nfo) Befehl erheblich erweitert. Wer ein G-DOS 2.4 sein eigen nennt, kann natürlich diese Version verwenden. Die Änderungen sind dann zwar etwas umfangreicher, lohnen sich aber auf alle Fälle! Wer eine solche G-DOS-Version verfügbar hat, kann sich wegen der Änderungen direkt an mich wenden.

Wie schon erwähnt, ist die POKE-Möglichkeit nicht gerade die eleganteste Art, die Adresse des neuen Druckertreibers in den DCB (Device Control Block) zu bringen. Daß es auch anders geht, zeigen die folgenden Seiten. Weiterhin gibt diese Lösung ein Beispiel dafür, wie man einen DOS-Befehl in seiner Funktion erweitern kann.

Wir wissen, daß man mit dem (I)NFO-Befehl nicht nur eine Reihe von Informationen über das System auf dem Bildschirm sichtbar machen, sondern auch die Treiberadressen im Tastatur-, Monitor- und Drucker-DCB auf ihre Normalwerte zurücksetzen kann. Warum sollte man den I(nfo)-Befehl nicht auch dazu mißbrauchen, einen eigenen, speziellen Grafikdruckertreiber zu inialisieren? Die Befehlsfolge I,G wäre doch eigentlich genau das richtige dafür!

Bevor man dieses neue Argument in den I-Befehl implementieren kann, muß man sich das SYS27/SYS etwas genauer anschauen.

Kommentiertes Assemblerlisting der "Sprungtabelle" und des "Argumentauswerters" von SYS27/SYS (SYS29/SYS aus G-DOS!).
Die ausgelassenen Teile sind für die jetzige Betrachtung uninteressant!

```
                        ORG   4D00H
4D00 FEFD      START    CP    0FDH      'richtiges Systemfile?
4D02 201B               JR    NZ,ERROR  'nein, dann Fehler anzeigen!
4D04 0D                 DEC   C         'I-Befehl?
4D05 282F               JR    Z,INFO    'dann Info-Routine!
4D07 0D                 DEC   C         'P-Befehl?
4D08 CA444F             JP    Z,PROT    'dann Port-Routine!
4D0B 0D                 DEC   C         'DISK-Befehl?
4D0C CA884F             JP    Z,DISK    'dann DISK-Routine
.
4D1F 3E2A      ERROR    LD    A,2A      'Code für ILLEGAL DOS FUNKTION laden
4D21 B7                 OR    A         'Z-Flag zurücksetzen
4D22 C9                 RET             'ins DOS und Fehler anzeigen!
4D23 211645    TAST     LD    HL,4516   'normale Treiberadresse
4D26 2211640            LD    (4016),HL 'in Tastatur-DCB laden!
4D29 C9                 RET             'zurück zu WEITER
4D2A 218D05    DRUCK    LD    HL,058D   'normale Treiberadresse
4D2D 222640             LD    (4026),HL 'in Drucker-DCB laden!
4D30 C9                 RET             'zurück zu WEITER
.
4D36 CDD54C    INFO     CALL  4CD5      'Argumente vorhanden?
4D39 2836               JR    Z,ANZEI   'nein, dann Information anzeigen!
4D3B 00                 NOP
4D3C 00                 NOP
4D3D E5                 PUSH  HL        'HL retten, wird gebraucht!
4D3E CD474D             CALL  ARGUM     'Welches Argument?
4D41 00        WEITER   NOP
4D42 E1                 POP   HL        'HL wiederholen!
4D43 23                 INC   HL        'auf nächstes Zeichen
4D44 28F0               JR    Z,INFO    'Fehler? Nein, dann weiter mit der
4D46 C9                 RET             'Argumentausw., sonst Feheranzeige!
4D47 7E        ARGUM    LD    A,(HL)    'Zeichen, auf das HL zeigt in A
4D48 FE54               CP    'T'       'ein T ?
4D4A 28D7               JR    Z,TAST    'ja, dann Tastaturroutine
4D4C FE4D               CP    'M'       'oder vielleicht ein M ?
4D4E 2814               JR    Z,MONIT   'dann Monitorroutine
4D50 FE44               CP    'D'       'vielleicht D ?
4D52 28D6               JR    Z,DRUCK   'dann Druckroutine
4D54 FE5A               CP    'Z'       'letzer Versuch! Ein Z ?
4D56 2806               JR    Z,ZEILE   'dann Zeilenroutine!
4D58 3E34               LD    A,34      'nein, dann Fehlercode für ILLEGAL
4D5A B7                 OR    A         'KEYWORD OR SEPERATOR OR TERMINATOR
4D5B C30944             JP    4409      'laden und Fehler anzeigen!
```

Gehen wir einmal davon aus, es wurde der DOS-Befehl I,D eingegeben. Das DOS hat mit Hilfe des Aufrufbytes SYS27 geladen und bei 4D00h gestartet. Da sich das Aufrufbyte (in diesem Fall FDh) noch im Accu befindet, wird die erste Hürde, die überprüfung, ob ein Illegal DOS Funktion - Fehler vorliegt, erfolgreich genommen. Danach wird das C-Register einmal DECrementiert (Inhalt von C - 1) und abgefragt, ob sein Inhalt jetzt gleich Null ist! Wäre dies nicht der Fall, würde solange C DECrementiert, bis entweder die richtige Routine gefunden ist, oder eine Fehlermeldung ausgegeben wird! Da aber beim Aufruf von SYS27 mit dem Befehl I das C-Register mit 81h geladen und das höherwertige Nibble während des RST 28 auf 0 gesetzt wird, ist nach einmaligem DECrementieren die Bedingung für den Befehl JR,Z (springe relativ, wenn das Zero-Flag gesetzt ist) erfüllt! Es geht also bei der Routine mit dem Label INFO weiter.

Hier wird nun als erstes eine Routine im DOS aufgerufen, die feststellt, ob irgendwelche Argumente hinter dem Befehl folgen. Ist dies nicht der Fall, wird nur die Information über die verschiedenen Treiber, Himem usw. ausgegeben. Folgt dem Befehl ein Argument (Zero-Flag nicht gesetzt), so wird dieses im Argumentauswerter (4D47h-4D5Bh) auf seine Richtigkeit überprüft. Ist es ein nicht erlaubtes Argument, wird ein Illegal Keyword or Seperator or Terminator - Fehler angezeigt, ansonsten wird die entsprechende Routine angesprungen. In unserem Beispiel würde also die Routine mit dem Label DRUCK aufgerufen. Nachdem diese die normale Treiberadresse in den Drucker-DCB eingetragen hat, wird abgefragt, ob noch ein weiteres Argument folgt. Ist dies nicht der Fall, werden jetzt die Informationen angezeit, ansonsten geht es nochmal in den Argumentauswerter.

Wie ihr sicher schon selbst festgestellt habt, ist der Platz, an dem wir mit unserer Erweiterung ansetzen müssen, der Auswerter für die möglichen Argumente des I-Befehls. Wir müssen nichts weiter tun, als nach

```
CP    'Z        'Argument ein Z ?
JR    Z,ZEILE   'dann zur Routine ZEILE springen!
```

durch Einfügen von

```
CP    'G'       'Argument ein G ?
JP    Z,GRAFIK  'dann zur Routine GRAFIK springen!
```

auch ein ,G als gültiges Argument abzufragen und zu einer entsprechenden Routine zu springen, dann können wir mit I,G unseren neuen Treiber initialisieren.
Aber ganz so einfach ist es leider doch nicht! Wir haben nämlich keinen Platz zwischen der Abfrage von ,Z und der Anzeige eines Illegal Keyword ... - Fehlers. Wir müssen uns also damit behelfen, nach der Abfrage von ,Z an einen Platz zu springen, wo wir genügend Raum zur Verfügung haben, um unser Vorhaben zu realisieren!
Wir ersetzen aus diesem Grund die Befehle LD A,34 und OR A (4D58-4D5A) durch NOP's (No Operation = keine Operation) und den darauffolgenden JP 4409h durch einen JP 4DD0h. Dort haben wir genug Platz, um die Abfrage von ,G und die eventuelle Anzeige eines Fehlers auszuführen!

```
4DCE 1810              JR    4DE0       'die früheren NOP's überspringen!!!
4DD0 FE47       NEU    CP    'G'        'Ein G ?
4DD2 CA1184F           JP    Z,GRAFIK   'Ja, dann die neue Routine aufrufen!
4DD5 3E34              LD    A,34       'Wenn nicht,
4DD7 B7                OR    A          'dann Fehleranzeige
4DD8 C30944            JP    4409       'wie gehabt!
```


Die eigentliche Routine zur Initialisierung der neuen Druckertreibers folgt dann bei 4F18h.

```
4F18 118866     GRAFIK LD    DE,M6688   'Die Adresse des neuen Treibers in DE
4F1B 1A                LD    A,(DE)     'erstes Byte des neuen Treibers in A
4F1C FE21              CP    21         'dieses Byte = 21 ?
4F1E 2005              JR    NZ,FEHL    'nein, dann Fehler
4F20 ED532640          LD    (4026),DE  'ja, dann neue Adresse in DCB!
4F24 C9                RET              'zurück zu WEITER!
4F25 3E08              LD    A,08       'DEVICE NOT AVALIBLE
4F27 B7                OR    A          'Fehler
4F28 C9                RET              'anzeigen!
```

Zuerst wird überprüft, ob das richtige BASIC, sprich das mit unserer Änderung, geladen ist. Ist dies nicht der Fall, wird ein DEVICE NOT AVALIBLE - Fehler angezeigt. Wem diese Fehlermeldung nicht gefällt, kann sie selbstverständlich ändern (einfach einen anderen Fehlercode eingeben, z.B. 23h = MEMORY FAULT), ich finde sie am sinnvollsten! Ist Extended-BASIC geladen, wird die Adresse der neuen Druckerroutine in den Drucker-DCB eingetragen und zum Argumentauswerter zurückgesprungen.

Das war wirklich nicht besonders schwierig! Oder? Ihr könnt Euch ja mal selbst an einem ähnlichen Problem versuchen, das praktisch auf jeden zukommen wird, der diese Änderung des I-Befehls nutzt. Jedesmal wenn man zwischen dem Grafik- und dem normalen Druckertreiber wechselt, wird die komplette Information über Treiber, Himem, Zeichensatz usw. angezeigt und zerstört damit den Bildschirmaufbau des aufrufenden Programms. Eine Erweiterung des I-Befehls um das Argument ,N = keine Anzeige der Information, wäre also durchaus sinnvoll!

Noch zwei Hinweise zum Schluß:
1. Der Platz, an dem unsere Erweiterung des Argumentauswerters steht (4DD0h-4DD8h) ist normalerweise mit NOP's gefüllt, die zwischen zwei Programmteilen liegen und ohne irgendeine Funktion zu erfüllen durchlaufen werden. Da wir dort jetzt unsere Routine abgelegt haben, muß ein Befehl zum überspringen dieses Bereichs vorangestellt werden!
2. Solange man in BASIC ist, kann der Grafikdruckertreiber immer initialisiert bleiben. Nur bevor man ins DOS geht, sollte der normale Druckertreiber mit CMD"I,D (oder mit CMD"S=I,G) wieder eingestellt werden. Ansonsten wird man beim ersten Versuch, etwas vom DOS aus auszudrucken (z.B.: DIR P), sein blaues Wunder erleben!

Die Änderungen, die im SYS27/SYS nötig sind, stehen direkt unter denen für BASIC/CMD. Wie ihr dem Hexdump sicher schon anseht, ist für den Einbau von ,N noch genügend Platz vorhanden!

Ich hoffe, diese Beschreibung war denen, die sie von Anfang an gelesen haben, nicht zu langatmig und/oder unverständlich! Vielleicht regt es sogar manchen dazu an, selbt einmal ein bischen etwas in dieser Richtung zu unternehmen? Ich wünsche mir, es wäre so!
Selbstverständlich wird das nicht die letzte Änderung sein, die ich, angeregt durch meinen "Vorturner" Arnulf Sopp, im NewDOS tätige. Zur Zeit wird daran gearbeitet, die ID-Funktion auch auf dem Model 3/4 lauffähig zu machen. Weiterhin stört es mich, daß man einmal falsch eingegebene DOS-Befehle nicht korrigieren kann, sondern sie komplett neu eintippen muß! Und da ich (hoffentlich) bald über einen Mapper und damit über 96k RAM verfügen werde, wird sich in dieser Richtung auch bald mal was tun!

Bis dahin wünscht euch viel Erfolg, Euer *Hartmut Obermann*!!!"

... über die Befehle I(nfo), P(ort) und DISK!

Bei der in diesem Clubinfo beschriebenen Installierung des I(nfo)-Befehls können die Atribute <,H>,<,NP> und <,N> nicht benutzt werden!

I <,T><,D><,M><,Z><,H><,G><,NP><,N>
I entspricht dem GDOS-Befehl INFO.
Ohne Parameter gibt I eine Uebersicht der Teiberadressen fuer Drucker, Tastatur und Monitor, den Beginn des reservierten Speicherplatzes, eine vollstaendige Auflistung aller derzeit verfuegbaren Bildschirmzeichen und die Eingangsdaten aller aktiven Ports.

Durch das Anhaengen von T fuer Tastatur, D fuer Drucker, M fuer Monitor und Z fuer die Zeilenzahl je Seite kann man die Treiber in ihren Initialisierungszustand versetzen.
Durch das Anhaengen von H wird HIMEM auf FFFFh gesetzt.
Durch das Anhaenen von G wird ein spezieller Grafikdrucker-treiber initialisiert (nur in BASIC mit entspr. Modifikation!).
Mit N wird jegliche, mit NP nur die Ausgabe der Ports unter-drueckt.

P entspricht dem GDOS-Befehl PORT.
P listet alle Ports, deren INPut-Signale nicht hochgesetzt sind, mit den entsprechenden Eingabewerten auf.

DISK #=$
Mit dem Befehl DISK koennen die PDRIVE-Parameter fuer fremde Disketten im Speicher eingestellt werden, ohne sie auf der Systemdiskette zu speichern.
Dabei gibt # die Laufwerksnummer und $ einen Buchstaben zwischen a/A und y/Y an. Der Buchstabe bezeichnet das gewuen-schte Diskettenformat.
z.B.: DISK 1=P
Die den Buchstaben entsprechenden Formate entnehmen Sie bitte dem GDOS - Handbuch.

Zap im Superzap!

Wer in sein Superzap die Änderung aus Info Nr. 8 Seite 13 eingebaut hat, kann mittels der 14 Byte langen, zusätzlichen Änderungen noch eine Unschönheit von Superzap ausräumen. Nachdem die Änderungen durchgeführt sind, kann man den Modify-Modus durch drücken von BREAK verlassen, ohne ein Byte auf der Diskette geändert zu haben! Das kann manchmal ganz nützlich sein, vor allem dann, wenn man in einen falschen Sektor geraten ist oder sich vertippt hat.

Hartmut Obermann

```
DRV 00  0100 0052 0000 0000 0000 0000 0000 0000  A§§R§§§§§§§§§§§§
O   10  0000 0000 0000 0000 0000 0000 00F6 FE7B  §§§§§§§§§§§§§§
OH  20  2802 FE1B CAF7 5CFE 1FC2 8752 E521 3952  (B A  ö   R !9R
    30  2274 5C21 5B03 22EF 5DE1 C369 5C33 333E  "tö!AC" Ü   1ö33>
DRS 40  8C32 E15D 79E6 F06F 2600 2929 79E6 0FB5   2 üy  o&§))y Ö
810 50  6F11 303C CDDE 5DE1 CBC0 FE20 380C CB48  oÜ0<        8L H
32AH60  CB88 C2A6 5D1E 00C3 905C 4FCB 80FE 0BCA    ü §  öO    H
    70  AE5C CBC0 FE1F C2AE 5CE5 21BC 5D22 745C  ö       ö ! ü"tö
    80  2105 622E F55D E1CB 8018 AF3A 4038 FE04  !Eb. ü    X :§8 D
    90  CAC5 54C3 345D 0000 0000 0000 0000 0000   T 4ü§§§§§§§§§§
    A0  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
    B0  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000  §§§§§§§§§§§§§§§§
```

Verbindend

Der ECB-Bus

Johannes Assenbaum

Vor acht Jahren von Mitarbeitern der Firma Kontron in die Welt gesetzt, hat der ECB-Bus recht schnell große Verbreitung gefunden. Heute ist er zumindest hierzulande der Bus schlechthin für Z80-Rechner in Steckkartenbauweise. Erstaunlicherweise findet man nur ganz selten einmal eine Beschreibung dieses Busses, die über die Pinbelegung der Steckverbinder hinausgeht. Und dabei hat der real existierende ECB-Bus durchaus auch ein paar handfeste Macken.

Als 1977 bei Kontron die ersten ECB, in vollem Wortlaut Einfach-Europaformat-Computer-Baugruppen, geschaffen wurden, ist bestimmt niemandem in den Sinn gekommen, daß der ECB-Bus ein paar Jahre später De-Facto-Standard für modulare Z80-Systeme sein würde. Damals war die Z80-Bausteinfamilie ganz neu, und es gab noch keine Europakartensysteme mit diesen leistungsfähigen ICs.

Durch die hohe Integrationsdichte der neuen Bausteine konnte man das ECB-Konzept äußerst flexibel auslegen. Ausgehend von einer CPU-Karte, die bereits ein vollständiges Kleinstsystem enthält, kann man durch simples Hinzufügen von Speicher- oder Ein-/Ausgabe-Erweiterungskarten fast beliebig komplexe Rechner aufbauen. Die Grenzen des Wachstums liegen hauptsächlich im mechanischen Bereich: Da alle Speicher und Ports vollständig dekodiert und die Bussignale doppelt gepuffert sind (Treiber auf der CPU-Platine und auf den Erweiterungskarten), ist eher der Kartenträger (zumeist ein 19-Zoll-Rack) voll, als daß der Adreßraum ausgeschöpft oder der Bus überlastet wäre. Das war jedenfalls in den späten 70ern Stand der Technik.

Vielseitig

Die Flexibilität war entscheidend für den Erfolg der ECB. In der Industrie, dem zunächst angepeilten Kundenkreis, gibt es kaum Standardanwendungen. Die ECB boten nun die Möglichkeit, aus standardisierten Baugruppen schnell eine passende Hardware zusammenzustellen, ohne einen unsinnigen Überhang an nicht gebrauchten Ports oder Speichern beziehungsweise Neuentwicklungen in Kauf nehmen zu müssen, weil es die gewünschte Konfiguration nicht gab.

Das große Interesse am ECB-System und die ständig steigende Nachfrage vor allem nach spezialisierten Portkarten führte dazu, daß auch andere Firmen in diesen Markt einstiegen. Und Anfang der 80er Jahre entdeckten dann die Hobbyisten die Z80-Bausteine und den ECB-Bus für sich, der daraufhin einen wahren Boom erlebte.

Diese Entwicklung brachte aber gewisse Probleme mit sich: Kontron hatte nicht von vornherein alle 64 Pins der verwendeten VG-Steckverbinder irgendwelchen Signalen zugeordnet. Einige Leitungen wurden für spätere Erweiterungen reserviert, seinerzeit dachte man an den Ausbau zu einem 16-Bit-System mit der Z8000-CPU. Doch schon ziemlich bald zeigte sich, daß ein 16-Bit-ECB-Bus gegen die Übermacht der vorhandenen 8-Bit-Karten keine Chance mehr hätte. Man entschloß sich, den 16-Bit-Plan aufzugeben und die so gewonnenen Busleitungen für realistischere Zwecke zu verwenden. Für zusätzliche Adreßleitungen zum Beispiel, zumal im Zuge ständig steigender Integrationsdichte und fallender IC-Preise Hauptspeicher über 64 KByte keine Utopie mehr darstellten.

Nun kamen aber mehrere Firmen etwa gleichzeitig auf diese Idee. Und da keine stark genug war, von den anderen als Vorreiter anerkannt zu werden, gibt es heute mehrere ECB-Bus-Belegungen und damit ein paar grundsätzliche Inkompatibilitäten zwischen den Platinen verschiedener Hersteller.

Das betrifft in erster Linie die zusätzlichen Adressierungssignale für Speicher über 64 KByte, aber auch einige Hilfssignale wie zum Beispiel die Prioritätssteuerung für DMA-Einheiten (BAI/BAO). Dabei wäre die unterschiedliche Busbelegung noch das kleinere Übel, so etwas ist bei Bedarf mit Messer, Lötkolben und ein paar Zentimetern Schaltdraht schnell behoben. Beim Umschalten der Speicherbänke kommen jedoch unterschiedliche Auswahlverfahren zum Einsatz. So liegen bei Kontron acht dekodierte Bank-Select-Signale auf dem ECB-Bus, mit denen man bis zu acht 64-KByte-Bänke ansteuern kann. Andere Hersteller haben den Adreßbus um einige Bits erweitert, die erst auf den Speicherkarten dekodiert werden. Hauptsächlich verwendet man vier weitere Adreßsignale, womit insgesamt ein Megabyte für Speicher zur Verfügung steht. Anwender des zweiten Verfahrens sind unter anderem auch Janich & Klaas sowie der Elektronikladen Giesler & Danne (ELZET 80), deren Busse bei Hobbyisten weit häufiger anzutreffen sind als der Kontron-Bus.

Unmöglich

Leider blieb es nicht bei diesen wenigen kritischen Stellen, die im wesentlichen nur bei Speichererweiterungen zum Tragen kommen. Besonders im Hobby-Bereich erschienen Rechnerkarten auf dem Markt, bei denen die ursprüngliche Busdefinition etwas eigenwillig ausgelegt war. Beispielsweise wurden Stromversorgungsleitungen umgelegt oder die Puffer weggelassen. Gelegentlich übernahm man den ECB-Bus auch für Systeme mit einer 'Nicht-Z80-CPU' (8085, 8086/88 und seit neuerem 68008), was fast jedesmal weitere Inkompatibilitäten mit sich brachte, da dann ein anderes Timing auf dem Bus herrscht. c't-Leser, die schon länger dabei sind, kennen so einen Fall: den c't 86 beziehungsweise dessen 'jüngeren Bruder', den c't 68 ECB.

Vorsichtig

Wer sich heute eine der vielen ECB-Bus-kompatiblen Rechnerkarten kaufen will, muß also schon genau wissen, was tatsächlich auf seinem Bus los ist. Das Prinzip 'Auspacken – Einschalten – Geht' ist nur dann angebracht, wenn man sich der Kompatibilität wirklich sicher sein kann; blindes Vertrauen auf diesbezügliche Werbeaussagen führt in der Regel zu einem 'Geht nicht', unter Umständen sogar zum Tode etlicher ICs, wenn nämlich die ±12V-Versorgung an die falschen Pins zu liegen kommt.

Als Hilfestellung zeigen die Tabellen die original Kontron-ECB-Busbelegung, die der beiden anderen genannten Firmen sowie die unserer Rechnerprojekte c't 86/68 ECB und c't 80 (PROF-80). Bitte beachten Sie, daß auch die Richtung der Signale von Bedeutung sein kann. Dazu ein letztes Beispiel aus der Praxis: Es gibt Systeme, bei denen die Signale RESET und PWRCL gleichberechtigt sind (beides Ausgänge der CPU-Karte). Laut ECB-Busdefinition ist RESET jedoch ein Eingang, über den vom Bus aus ein System-Reset ausgelöst werden kann. Kombiniert man nun eine solche CPU-Karte mit 'normalen' ECB-kompatiblen Baugruppen, gelangt eine Reset-Anforderung auf dem Bus nicht mehr zur CPU. Umgekehrt kann es auch passieren, daß der Reset-Impuls von der CPU-Karte zur Peripherie (PWRCL) nicht durchkommt, weil deren Reset-Eingang am anderen Pin liegt und die Anforderung nicht über den Bus gelaufen ist. □

a				c
+5V		1		+5V
D5		2		D0
D6		3		D7
D3		4		D2
D4		5		A0
A2		6		A3
A4		7		A1
A5		8		A8
A6		9		A7
/WAIT	-	10	#	
/BUSRQ	+	11	-	IEI
	#	12	#	
+12V		13	#	
	#	14		D1
-5V	#	15		-15V
2(phi)	+	16	-	IEO
	#	17		A11
A14		18		A10
+15V		19	#	
/M1	+	20		/NMI
	#	21		/INT
	#	22		/WR
	#	23	#	
VCMOS	#	24		/RD
	#	25		/HALT
	#	26		/PWRCL
/IORQ		27		A12
/RFSH	+	28		A15
A13		29		(phi)
A9		30		/MREQ
/BUSAK	+	31		/RESET
Gnd		32		Gnd

Tabelle 1. Die ECB-Busbelegung, soweit sie einheitlich gehandhabt wird, die „umstrittenen" Pins („#") siehe Tabelle 2. Mit „-" und „+" markierte Pins sind beim c't 86/68 ECB nicht beziehungsweise anderweitig verwendet (siehe c't 11/85, Seite 63).

(siehe c't 11/85, Seite 63)

Dänisch: En fjer bliver til fem høns.
Aus einer Feder werden fünf Hühner.
Aus einer Mücke einen Elefanten machen.
Dorte Salomon, Berlin

Have en rav bag øret.
Einen Fuchs hinter dem Ohr haben.
Etwas im Schilde führen.

Englisch: To get on the gravy train.
Auf den Bratensoßenzug aufspringen.
Absahnen.

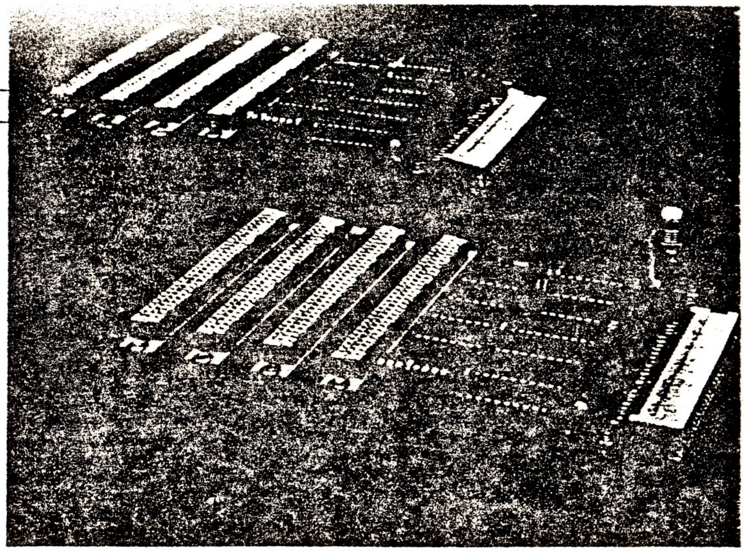
To paint the town red.
Die Stadt rot anmalen.
Auf die Pauke hauen.
Cavin Specht, Regensburg

Pin	Bedeutung Kontron	J & K	ELZET80	c't80	c't86/68 ECB
10c	MSB0 (A16)	A16	---	reserv.	PCLK (phi)/2
12a	BAI	BAI	A18	A18	A18
12c	MSB1 (A17)	A17	A19	A19	A19
13c	MSB2 (A18)	A18	---	reserv.	-12V
14a	MSB3 (A19)	A19	-12V	-12V	CLOCK (phi)
15a	-12V	-5V	-5V	reserv.	-5V
17a	BAO	BAO	A17	A17	A17
19c	MSB5 (A21)	---	A16	A16	A16
21a	MSB6 (A22)	---	---	reserv.	reserv.
22a	MSB7 (A23)	/DMARDY	/BOOT	---	reserv.
23a	RETIRQ	/PF	/BAI	/BAI	BAI (-)
23c	MSB4 (A20)	/SEL	/VID BLNK	---	reserv.
24a	---	VCMOS	VCMOS	UBAT	5V-BAT (-)
25a	n(phi)	---	/BAO	/BAO	BAO (-)
26a	/WRITEN	/DESLCT	---	reserv.	reserv.

Tabelle 2. Die zunächst undefinierten Leitungen des ECB-Bus und was aus ihnen geworden ist. Man beachte, daß Kontron die Pins 15a und 24a nachträglich umbelegt hat. Die Markierungen „-" in der Spalte c't 86/68 ECB haben dieselbe Bedeutung wie in Tabelle 1.

Signal	Pin	Richtung (Fan In/Out)		Bedeutung
A0	5c	A	(9 TTL)	Adreßbus
A1	7c	A	(9 TTL)	
A2	6a	A	(9 TTL)	der CPU
A3	6c	A	(9 TTL)	
A4	7a	A	(9 TTL)	
A5	8a	A	(9 TTL)	
A6	9a	A	(9 TTL)	(hoch-
A7	9c	A	(9 TTL)	ohmig
A8	8c	A	(9 TTL)	bei
A9	30a	A	(9 TTL)	BUSAK)
A10	18c	A	(9 TTL)	
A11	17c	A	(9 TTL)	
A12	27c	A	(9 TTL)	
A13	29a	A	(9 TTL)	
A14	18a	A	(9 TTL)	
A15	28c	A	(9 TTL)	
D0	2c	E/A	(Fan Out:	Datenbus
D1	14c	E/A	25mA	
D2	4c	E/A	bei Low)	der CPU
D3	4a	E/A		
D4	5a	E/A		(hoch-
D5	2a	E/A	(Fan In:	ohmig
D6	3a	E/A	0,25mA	bei
D7	3c	E/A	bei Low)	BUSAK)
/MRQ	30c	A	(9 TTL)	CPU-Steuer-
/IORQ	27a	A	(9 TTL)	ausgänge
/RD	24c	A	(9 TTL)	(hochohmig
/WR	22c	A	(9 TTL)	bei BUSAK)
/M1	20a	A	(9 TTL)	(Tri-State-
/RFSH	28a	A	(9 TTL)	Treiber nur
/HLT	25c	A	(9 TTL)	bei mehr als einer CPU im System)
/WAIT	10a	E	(Pullups	CPU-Steuer-
/INT	21c	E	4k7)	eingänge
/NMI	20c	E		
/RESET	31c	E	(plus 1 TTL-Eingang)	
IEI	11c	E	(Pullup 4k7)	Interrupt-
IEO	16c	A	(1 TTL)	Daisy-Chain
/PWRCL	26c	A	(9 TTL)	Peripherie-Reset
(PHI)	29c	A	(keine	CPU-Takt
2(PHI)	16a	A	Angaben)	Taktsignal doppelter Frequenz
/BUSRQ	11a	E	(Pullup 4k7)	DMA-Steuer-
/BUSAK	31a	A	(9 TTL)	Signale
+5V	1a,c			Stromver-
Gnd	32a,c			sorgungen
+12V	13a			
-5V	15a			(-12V bei Kontron)
(-12V	14a			ELZET80/c't80)
+15V (+12V)	19a			Für V.24
-15V (-12V)	15c			und Analoges
VCMOS/UBAT	24a			Backup-Akku (Spannung unterschiedlich, bei Kontron nicht mehr vorgesehen)
				Weitere Signale:
n(PHI)		A		noch ein Takt (nur Kontron)
BAI		E		Daisy-Chain
BAO		A		für DMA
A16...19		A	(9 TTL)	zusätzliche Adressen
MSB0...7 oder		A	(9 TTL)	Kontron: Bank-select- oder
A16...23		A	(9 TTL)	Adreßsignale

Tabelle 3. Eine Kurzübersicht über die Bedeutung und Richtung (aus der „Sicht" der CPU-Karte) der Signale auf dem ECB-Bus. Die Angaben zur Treiberleistung (Fan Out) sind Mindestwerte. Nicht genannte Signale sind nur in dem betreffenden System von Belang.



Andreas Stiller

Schluß mit der Spezial-Hardware! Es lebe die Kompatibilität. So wie man beispielsweise mit SuperTape durch ein einheitliches Kassettenverfahren das babylonische Formatwirrwarr verschiedener Rechner beseitigte, ist es auch sinnvoll, sich für alle Z80-Rechner auf einen Anschluß-Standard für die Peripherie zu einigen. Hierfür bietet sich der bei professionellen Rechnern weitverbreitete ECB-Bus an, den mit Hilfe eines Adapters nun auch die Schneiderfamilie 'antreiben' kann.

Schneider findet Anschluß

ECB-Adapter für alle CPCs

Was der ECB-Bus eigentlich ist und welche Bedingungen er erfüllen muß, war uns einen eigenen Artikel an anderer Stelle in diesem Heft wert. Für dieses Bus-System hat c't nämlich bereits etliche Hardware-Projekte vorgestellt:

- Intelligente Floppy-Karte, die eigenständig CP/M fährt
- Eprommer
- I/O-Karte mit CTC, SIO und 2 × PIO
- EPAC-80 (der auch als I/O-Karte zu gebrauchen ist)
- 1-MByte-RAM-Karte
- Bus-Monitor
- Floppy-Karte (vom c't86) mit WD 2797
- I/O-Karte (vom c't86) mit 8153A, 8255A, 8251A, mit SuperTape-Kassettenschnittstelle

Diese umfangreiche Palette von Peripherie-Karten ist nun auch für die Schneiders verfügbar, wobei die RAM-Karte allerdings noch ein wenig Hardware-Unterstützung benötigt. Für den Eprommer gibt es eine umfangreiche Software (unter CP/M), die zur Zeit auf die etwas beschränkte TPA (freier Speicher) und den 16-Bit-I/O der kleinen CPCs angepaßt wird.

Hürden

Allzuleicht haben es dabei die Amstrad-Entwickler nicht gemacht, den CPCs die ECB-Welt in ihrer Vielfalt zu erschließen. Für Schwierigkeiten sorgen nicht nur die unüblichen – und nicht ausdekodierten – 16-Bit-Ports sowie die ungewöhnliche Interrupt-Struktur (siehe Schneiders Kern, Teil 1, c't 8/85), sondern auch die Unterschiede innerhalb der inzwischen auf drei Mitglieder angewachsenen CPC-Familie.

Zweckmäßigerweise bietet der Adapter auch weiterhin einen Standard-Platinenanschluß für die Schneider-Peripherie. Das ist besonders für den CPC 464 wichtig, da sich dadurch problemlos und ohne Umlötaktion eine Floppy-Station (von Schneider, Vortex etc.) anschließen läßt. Um auch hier etwas 'Leistungsreserve' zu bekommen, erhalten auch die Signale für den Platinenanschluß genau wie die ECB-Bus-Signale über die vier Bustreiber (LS244 und LS245) eine kräftige Verstärkung.

Beim CPC 664 und CPC 6128 befinden sich Floppy-Controller und Floppy-ROM ja bereits mit im Rechnergehäuse, also auf der anderen Seite des Adapters, was eine etwas aufwendigere Umschaltung der Datenrichtung (des bidirektionalen LS245) notwendig macht.

16/2 = 256

Mit 16 Bit lassen sich theoretisch 65536 Portadressen ansprechen. Da aber wohl kaum jemand einen so riesigen Peripherie-Bedarf hat, haben sich die CPC-Entwickler auf maximal zehn Geräte beschränkt (vier interne und sechs externe). Das Adreßbit A10 entscheidet dabei, ob es sich um ein externes oder ein im Rechner bereits eingebautes internes 'Gerät' (Gate-Array, Video-Chip, Centronics-Ausgang, 8255-PIO) handeln soll. A13 dient zusammen mit den Datenbits zur Selektion eventuell vorhandener Expansion-ROMs. Die verbleibenden vier Bits (A9,A8,A1,A0) können beliebig zur Steuerung des durch die übrigen Bits ausgewählten Geräts eingesetzt werden.

Für den ECB-Adapter ist insbesondere das Adreß-Bit A10 (interner/externer Port) von Bedeutung. Um auf einen ECB-Port zugreifen zu können, muß A10 low sein (und A13 sowie A9 auf high). Nur dann wird nämlich das Signal $\overline{IORQ}$ über den Adreßdekoder LS138 (IC9, Ausgang Y5) an den ECB-Bus weitergereicht.

Die Adapterschaltung ist so ausgelegt, daß sie auch eine korrekte Interrupt-Behandlung für den ECB-Bus ermöglicht. Dazu ist es nötig, daß die Peripherie die Antwort auf einen Interrupt von der CPU erhält, das sogenannte Interrupt Acknowledge (INTACK), das aus der Signalkombination M1 und IORQ – gleichzeitig auf low – besteht. Diese Kombination wird von einem OR-Gatter in IC10 'erkannt', und das erzeugte INTACK-Signal sorgt dafür, daß unabhängig von irgendwelchen Portadressen das IORQ-Signal auf den ECB-Bus kommt (über ein UND-Gatter von IC6).

Auf der Software-Seite ist für einen Zugriff auf einen 8-Bit-ECB-Port also in bester Schneider-Manier weiterhin eine 16 Bit Portadresse nötig. Hierbei müssen die oberen fünf Adreßbits (A11 bis A15) gesetzt sein, da sich sonst vielleicht gleichzeitig der Video-Controller angesprochen fühlt oder der Drucker losrattert.

Auch A9 muß bei einem Portzugriff high sein, da die Kombination A9=0, A10=0, A13=1 als Software-Reset für die ECB-Peripherie interpretiert wird. Folglich sind die Adressen &FAxx und &FBxx für die ECB-Ports xx und &F8xx und &F9xx für den Software-Reset auserkoren. Die Adreßlage für den Reset ist so gewählt, daß sie der Adresse des System-Resets(&F8FF) entspricht, den die Schneider-Firmware beim Einschalten oder nach CTRL-SHIFT-ESC ausgibt. Für einen ordnungsgemäßen Reset der üblichen Z80-Bausteine muß das Reset-Signal einige Taktzyklen auf low liegen, so beispielsweise bei Mostek:

SIO: ein Takt,
PIO: zwei Takte,
CTC: drei Takte

Normalerweise ist die Dauer eines I/O-Zugriffs mit etwa zweieinhalb Takten dafür etwas knapp. Beim Schneider aber helfen die zusätzlichen WAIT-Zyklen kräftig mit (siehe 'Vertrackte Warte-Takte', c't 9/85), so daß der Reset-Puls bei OUT (C),A drei zusätzliche Takte erhält und nun lang genug ist.

Weitere Adressen sind &DExx und&DFxx, die für die ROM-Selektion verantwortlich sind. Das dazu erzeugte Signal EXTROM (Ausgang Y6 von IC9) gelangt allerdings nicht auf den ECB-BUS, es steht aber an der Steckleiste X2 zur Verfügung.

Für Spezialzwecke kann es manchmal nützlich sein, ein Strobe-Signal, also einen kurzen Puls, an einem Portausgang zu haben. So beispielsweise für ein DATA-STROBE-Signal an einem Druckerport. Hiermit ließe sich also auch das berühmt-berüchtigte achte Bit der verstümmelten Centronics-Schnittstelle restaurieren.

Solche Strobe-Signale kann man den noch unbenutzten Ausgängen des Adreßdekoders (IC9) entnehmen, von denen eins (Y0, Adresse &D9xx) auch auf die Steckleiste X2 geführt ist.

Die Länge dieser Strobe-Signale beträgt ebenso wie beim Software-Reset etwa 5,5 Taktzyklen, also knapp 1,5 Mikrosekunden.

Durchgeschleift

Genau wie auch beim ECB-Bus erfährt das IORQ-Signal für den Platinenanschluß zur Schneider-Peripherie eine Sonderbehandlung. Das wäre zwar nicht unbedingt nötig, ermöglicht aber, viele Portadressen zu retten, die sonst durch die sehr unvollständig dekodierten Peripherie-Geräte verloren gingen. Beispielsweise belegt die Floppy-Station des CPC 464 bei einem externen Portzugriff (&FAxx oder &FBxx) sämtliche Adressen zwischen xx = &00 und xx = &7F, so daß für den ECB-Bus nur noch die Portadressen xx = &80 bis xx = &FF verbleiben. Beim CPC 664 und CPC 6128 kann man allerdings diese Einschränkung der Adreßfreiheit nur verhindern, wenn man einen Eingriff in den Rechner vornimmt, um hier die Manipulation am IORQ-Signal zwischen CPU und Floppy-Station vorzunehmen. Das sollte man aber tunlichst lassen und sich lieber mit den noch freien 128 Portadressen zufriedengeben. Die einstellbare Dekodier-Logik (J3 und IC8) sorgt dann dafür, daß nicht noch zu viele weitere Portadressen durch Anschluß eines Modems oder anderer nur teilweise dekodierter Peripherie-Geräte verlorengehen.

Es gibt vier Wege für das IORQ-Signal, um an den Platinenanschluß zu gelangen: Software-Reset, ROM-Select, Interrupt-Acknowledge und Zugriff auf einen externen Port, wenn die Adreßbits A2 bis A7 die durch S1.1 bis S1.6 einstellbare Bedingung erfüllen. Das Vierfach-Und (IC5a) führt diese Wege zusammen.

S1 kann man wahlweise mit Lötbrücken, Stiftleisten und Jumpern oder einem Achtfach-DIL-Schalter bestücken. Ist ein Schalter von S1.1 bis S1.6 auf 'on', so muß das zugehörige Adreßbit high sein, damit die Dekodierbedingung erfüllt ist. S1.8 ist auf 'on' zu stellen, falls der Rechner eine eingebaute Floppy-Station hat, also beim 664 und beim 6128. Beim CPC 464 ist die zugehörige Schalterstellung von S1.8 'off'.

Die übliche Schalterstellung für den CPC 464 mit Floppy-Station ist dann:

S1.8 off	S1.4 on
S1.7 on	S1.3 on
S1.6 off	S1.2 on
S1.5 on	S1.1 on

Die Floppy-Station nimmt dann im 'Raum der Portadressen' nur noch den Bereich von xx = &7C bis xx = &7F in Anspruch. Alle anderen Adressen stehen dem ECB-Bus zur freien Verfügung.

Beim CPC 664 oder 6128 ist die übliche Schalterstellung bei Anschluß einer seriellen Schnittstelle:

S1.8 on	S1.4 on
S1.7 on	S1.3 on
S1.6 on	S1.2 on
S1.5 off	S1.1 on

Die für den ECB-Bus freibleibenden 8-Bit-Adressen sind demnach xx = &80 bis &FF, wovon die serielle Schnittstelle nur die Adressen &DC bis &DF belegt.

Richtungswechsel

Da die Daten in zwei Richtungen 'fließen' können – von der Peripherie zur CPU und umgekehrt –, hat die Adapterschaltung dafür zu sorgen, den bidirektionalen Datentreiber (LS245, IC3) immer rechtzeitig umzuschalten. In der Ruhestellung ist der Treiber auf Schreiben (aus der Sicht der CPU) gestellt. Die Peripherie akzeptiert die anstehenden Daten aber nur, falls gleichzeitig auch das WR-Signal aktiviert ist.

Unter drei Voraussetzungen soll der Treiber nun von außen ankommende Daten auf den CPU-Datenbus legen.

- Lesen von externem ROM oder RAM
- Lesen von einem externen Port
- beim Interrupt Acknowledge

Die Schwierigkeit besteht nun darin zu erkennen, ob es sich nun um einen internen oder externen Zugriff handelt. Bei externem ROM hat die Peripherie mitzuhelfen: Wenn sich irgendwo ein externes ROM angesprochen fühlt (zum Beispiel in der externen Floppy-Station), muß dieses Gerät ein besonderes Signal (ROMDIS, aktive high) liefern, das das interne ROM ausblendet. Ebenso ist es bei externem RAM mit dem Signal RAMDIS. Die Umschaltlogik wertet nun diese Signale aus (über IC10c, IC11d, IC10a, IC5b), wobei die Dioden D1 und D2 dafür sorgen, daß eventuell vom Rechner herrührende Ausblendsignale (bei eingebauter Floppy-Station) unberücksichtigt bleiben. In diesem Fall bleibt der Treiber-Baustein daher weiterhin auf Schreiben stehen, so daß er nicht die vom internen Floppy-ROM kommenden Daten überblendet (der LS245 ist nämlich weitaus stärker).

Bei den Ports ist die Sache scheinbar einfacher, da ja wie schon beschrieben das Adreßbit A10 über intern/extern entscheidet. Nun wird aber die Floppy-Station grundsätzlich als externes Gerät angesehen, egal ob sie eingebaut ist oder nicht. Daher muß man über den Schalter S1.8 der Umschaltlogik die 'Lage' der Floppy-Station mitteilen. Bei S1.8 auf 'on' (eingebaute Station) interpretiert sie alle Portadressen xx ≦ &7F als intern und schaltet beim Lesezugriff den LS245 auf Schreiben (IC11a, IC12c, IC10c, IC5b).

Mit IM 2

Ein besonderes Augenmerk verdient die Behandlung der Interrupts. Viele ECB-Bus-Karten haben Interrupt-Quellen 'on board', die korrekt bedient werden wollen.

Der Schneider läuft normalerweise im Interrupt-Mode 1, das heißt, daß die CPU bei einem Interrupt immer das Programm ab &38 ausführt. Da aber die CPCs auch über periodische, interne Interrupts verfügen (300mal pro Sekunde), ergibt sich das Problem, wie man zwischen externen und internen Interrupts unterscheiden kann.

Wie schon in der Serie über des Schneiders Kern ausgeführt, haben sich die CPC-Entwickler hier etwas Besonderes einfallen lassen. Die Interrupt-Routine prüft nämlich nach einiger Zeit nach, ob die Interrupt-Anforderung immer noch besteht. Falls nein, interpretiert sie den Interrupt als intern, im anderen Fall vermutet sie eine externe Quelle und führt dann die User-Interrupt-Routine (Sprungvektor steht in &3B) aus. Diese Behandlungsroutine muß dann gezielt durch einen OUT-Befehl die unterbrechende Einheit zur Rücknahme der Interrupt-Anforderung veranlassen.

Diese Art der Interrupt-Behandlung entspricht allerdings nicht den Bedürfnissen der üblichen Z80-Peripherie-Bausteine, wie man sie meistens auf den ECB-Karten vorfindet. Man hat jedoch mehrere Möglichkeiten zur Abhilfe:

– Man ersetzt die Interrupt-Routine (ab &B939 beim CPC 464 und &B941 bei CPC 664) durch eine eigene, die auf eine andere Art die Herkunft des Interrupts ermittelt, zum Beispiel, indem sie die in Frage kommenden Geräte abfragt, ob sie einer Behandlung bedürfen (Polling).

– Die Adapterschaltung bietet die Möglichkeit, (Steckbrücke J2 gesetzt) mittels eines Monoflops (IC7a) die Interrupt-

HEFT 11 Januar 1985

[illegible] Ge-[illegible] einige Mikrosekunden zu verlängern, so daß das Betriebssystem den externen Interrupt richtig erkennt. Die Zeitkonstante des Monoflops sollte dabei normalerweise etwa 30 Mikrosekunden betragen, was sich durch entsprechende Wahl des Zeitglieds R1/C1 erreichen läßt. Da $t \approx 0.7 \times R \times C$ ist, käme beispielsweise R1 = 3k9 und C1 = 10 nF in Frage.

– Die weitaus komfortabelste Möglichkeit bietet der Interrupt-Mode 2, bei dem das auslösende Gerät einen Interrupt-Vektor liefert. Der Adapter unterstützt diesen mächtigen Modus, indem er dafür sorgt, daß der Interrupt-Vektor richtig eingelesen wird, und er setzt bei einem internen Interrupt diesen Vektor mit Sicherheit auf &FF. Der dadurch adressierte Tabellenplatz (Interrupt-Register I: high Byte, Interrupt-Vektor: low Byte), der hier ausnahmsweise ungerade ist, muß die Adresse der normalen Interrupt-Routine beinhalten.
Wenn also beispielsweise die Interrupt-Tabelle ab Adresse &B600 stehen soll (im Sound-Puffer beim CPC 464), so muß man vor dem Umschalten auf IM 2 die Interrupt-Adresse eintragen:

```
&B6FF: 38
&B700: 00
```

Zwei Dinge sind beim Arbeiten mit Interrupts noch zu beachten. Zum einen müssen Tabelle und Behandlungs-Routinen im ungebankten RAM stehen, zum anderen ist einzuplanen, daß sich die Reaktionszeit auf einen Interrupt manchmal etwas hinziehen kann. Wenn der Rechner sich nämlich gerade mit der internen Interrupt-Behandlung abmüht, sind externe Interrupts erst einmal auf Eis gelegt. Ohne Eingriff in den Rechner läßt sich die interne Interrupt-Quelle leider nicht abstellen. Wenn Zeitnot besteht, kann man aber eine eventuelle Verzögerungszeit durch den internen Interrupt gewaltig abkürzen, indem man die Behandlungsroutine ersetzt durch:

```
EI
RETI
```

Komm zurück

Wenn sich der Rechner mal wieder jenseits von Gut und Böse im 'Nirwana' befindet und sich auch mittels CTRL-SHIFT-ESC nicht zur Rückkehr bewegen läßt, bleibt einem nur noch ein Ausweg: der Reset. Das Aus- und Einschalten des Rechners ist jedoch dafür nicht die beste Lösung, da unter Umständen noch versehentlich eingelegte Disketten Schaden nehmen könnten. Außerdem hat man so mit Sicherheit keine Chance mehr, wertvollen Speicherinhalt zu retten.

Der Adapter bietet deshalb auch eine kleine Reset-Logik, die über den Taster T1 und den Monoflop IC7b einen kurzen Reset-Puls auf die BUSRESET-Leitung legt.

Es muß allerdings die Software mithelfen, daß bei der nun folgenden Initialisierung nicht doch noch der gesamte Speicherinhalt gelöscht wird, ein unangenehmer Effekt, den nicht das Kernal-, sondern das BASIC-ROM zu verantworten hat. Falls jedoch ein anderes Hintergrund-ROM die Kontrolle übernimmt, vielleicht sogar mit einem Monitor-Programm, kann man diesen Löscheffekt verhindern.

Den Reset-Vektor im RAM ab &0000 auf eine eigene Routine zu verbiegen, bringt nichts ein, da zumindest der verlagseigene CPC 464 beim Reset anscheinend immer das ROM anwählte. Andernfalls spielte man sonst va banque, ob gerade das RAM (Sekt) oder das parallele Kernal-ROM (Selters) selektiert ist.

Optionen

Wer weder die Interrupts noch den speichererhaltenden Reset benötigt, kann auf die beiden Monoflops (IC7) und die Zeitglieder R1, C1, R2, C2 auch verzichten. Um dann mittels T1 einen einfachen Reset auslösen zu können, ist eine Verbindung zwischen Pin 9 und Pin 12 von IC7 zu ziehen.

Noch mehr sparen läßt sich, wenn man gar keine Schneider-Peripherie am durchgeschleiften Bus anschließen möchte oder wenn man nur wenige Portadressen auf dem ECB-Bus benötigt. Dann können IC5, RN2 und der DIL-Schalter J3 (wahlweise Jumperfeld) entfallen. Lediglich zwei Verbindungen sind statt dessen nötig: Pin 9 von IC11 an Pin 8 von IC7 und Pin 1 von IC11 entweder an plus (CPC 464) oder an Pin 8 von J3 (CPC 664 oder 6128).

Auch die Dioden D4 und D5 samt Steckbrücken J3 und J4 sind 'optional'. Über sie kann man die Ausblendsignale für RAM und ROM vom ECB-Bus empfangen. Um problemlos mit RAM-Erweiterungen arbeiten zu können, sollte man aber einen Eingriff in den Rechner wagen und das Beschreiben des internen RAMs bei Zugriff auf externe Speicher verhindern (siehe Bild 1).

Die voreingestellte Lötbrücke L1 ist für Spezialisten gedacht, die signalgesteuert den Schneider vom ECB-Bus abkoppeln wollen. So kann man entscheiden, ob dann der Takt mit abgehängt wird (voreingestellt) oder weiterhin als Bus Clock auf dem Bus verbleibt. Auch das Schicksal der Signale RFSH, M1 und HALT hängt von der Einstellung ab. Normalerweise sind sie ebenso wie die anderen CPU-Signale abhängig vom Abkoppelsignal (Pin 12 von X2).

Wer all diese Möglichkeiten nutzen möchte, kann dafür eine Pfostenstiftleiste samt Steckbrücke einsetzen und die Voreinstellung auftrennen.

Schließlich bleibt noch die Steckbrücke J1, die darüber entscheidet, woher der Adapter und der Bus den 'Saft' bekommen. Da das Netzteil im Schneider-Monitor noch einige Reserven hat, reicht die Stromversorgung für den ECB-Betrieb meist noch aus (J1 gesetzt). Im anderen Fall muß wohl oder übel ein zusätzliches Netzteil (5V) her, das man an die vorgesehenen Lötstifte anschließen kann (J1 offen). R6 sorgt dann dafür, daß die Treiber so lange hochohmig sind, bis auch der Rechner eingeschaltet ist.

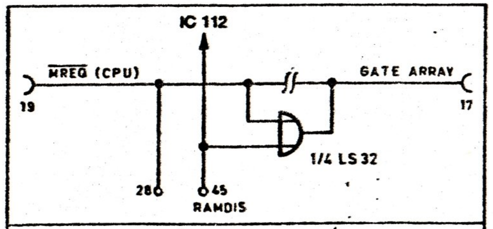


Bild 1. Wenn man dieses Gatter (huckepack über den vorhandenen 74 LS 32) in den CPC einbaut, kann man problemlos externes RAM anschließen.

Auf die Plätze...

Je nach CPC-Typ variiert der Anschluß an den Rechner ein wenig. Beim CPC 464 und 664 hat man die Wahl, ob man den Adapter auf 'Beinchen' stellt (30 mm bei 464, 10 mm beim 664) und mit einem Platinenstecker direkt anschließt oder ob man lieber ein Stück Flachbandkabel dazwischen hat. Dann erfolgt am besten der Anschluß am Adapter über die Pfostenstiftleiste. Letzteres gilt auch für den 6128, der an der Rechnerseite einen 50poligen 'Centronics'-Stecker erfordert. Beim Direktanschluß des CPC 464 ist noch zu beachten, daß der Platinenstecker leicht nach unten abzuwinkeln ist, damit er auf die schief eingebaute Platine richtig paßt.

...Fertig, los!

Wie üblich sollte man nach dem Zusammenbau die Platine noch einmal sorgfältig auf kalte Lötstellen, Kurzschlüsse und andere Unschönheiten überprüfen. Als nächstes sollte man wenn möglich die Stromaufnahme der Platine messen, die etwa 120 mA verschluckt. Nun ist es Zeit, die Platine (mit der richtigen DIL-Schalterstellung) an den Rechner anzuschließen – natürlich im ausgeschalteten Zustand.

Als erstes kann man die Funktion des Reset-Tasters testen, dessen Wirkung deutlich sichtbar sein müßte. Falls sich nichts tut, haben Sie vielleicht die Steckbrücke J1 für die Versorgungsspannung vergessen.

Für das weitere Testen aller Busfunktionen und zur eventuellen Fehlersuche wäre der ECB-Bus-Monitor aus c't 9/85 ideal. Falls Sie selbst sich noch keinen zusammengebaut haben: vielleicht könnte ein Freund (oder auch eine Freundin?) Ihnen für ein paar Stunden dieses praktische Werkzeug überlassen.

Zum Testen stellen Sie den Monitor auf &FB80, WR und IORQ. Wenn Sie nun

```
OUT (&FB80),&FF
```

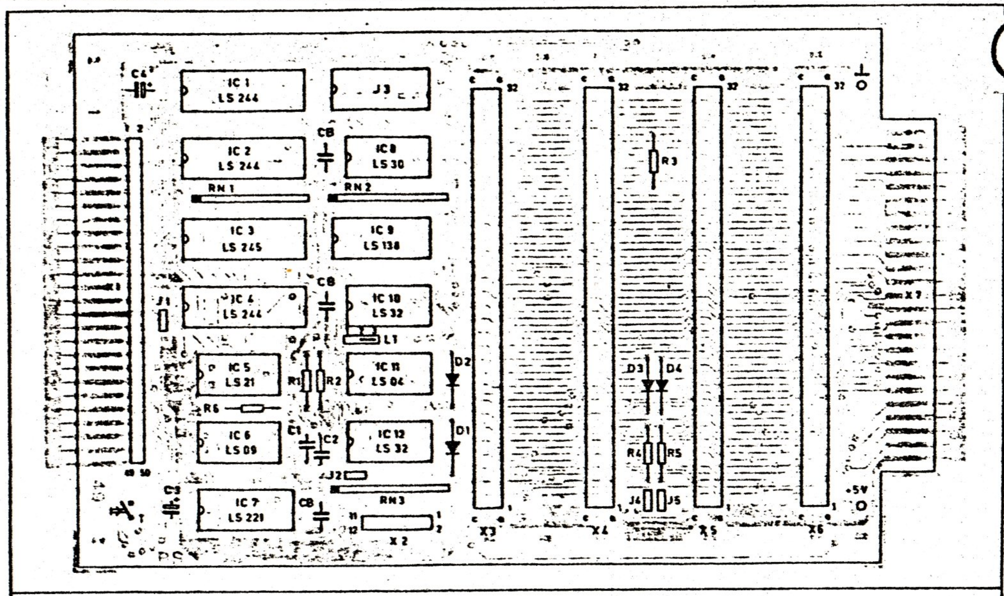
ausgeben, müßten alle Datenlämpchen leuchten. Sie sollten an diese Adresse mehrere verschiedene Werte ausgeben, um die Datenleitung auf eventuelle Kurzschlüsse zu überprüfen. Mit

```
OUT (&F8FF),0
```

müßten alle Lämpchen wieder verlöschen (Software-Reset).

Wenn Sie den Monitor auf &BDD3, M1 und RD stellen, Wait zulassen und dann einen Reset auslösen, passiert auf dem Bildschirm erstmal gar nichts – nur das Wait-Lämpchen leuchtet auf. Nun läßt sich mit dem Wait-Taster auf der Monitor-Karte Zeichen für Zeichen der Initialisierungsmeldung durchsteppen.

Wenn soweit alles läuft, können Sie davon ausgehen, daß der Adapter funktionsbereit ist und sich nun mit der Eprommer-, I/O-, Floppy- oder Speicherkarte beschäftigen. □



Der Adapter bietet vier ECB-Steckplätze und auch einen gepufferten SchneiderBus.

Stückliste Schneider ECB-Adapter

ICs

IC1,2,4	74LS244	Treiber
IC3	74LS245	Bi-Treib
IC5	74LS21	2×4 UND
IC6	74LS09	4×2 UND o.C.
IC7	74LS221	2×MMF
IC8	74LS30	1×8 UND
IC9	74LS138	Decoder
IC10,12	74LS32	4×2 OR
IC11	74LS04	6×INV.

Dioden

D1,2,3,4 1N4148

Kondensatoren

C1	10nF
C2	10nF
C3	1µF/10V Tantal
C4	47µF/10V Tantal
3 × CB	100 nF

Widerstände

R1	3k9
R2	10k
R3	330R
R4,5	2k2
R6	1k
RN1	8 × 4,7k
RN2	8 × 10k
RN3	8 × 1k

Fassungen

4 × 20pol
2 × 16pol (+1 für DIL-Schalter)
6 × 14pol

Sonstiges

Bei Direktanschluß (CPC 464 und CPC 664):
1 × Platinenstecker 2 × 25pol 1/10" mit Lötstiften
6 × Abstandsröhrchen 30 mm (464) oder 10 mm (664) mit passenden M3-Schrauben und Muttern

Anschluß über Kabel:
1 × Platinenstecker 2 × 25pol 1/10" mit Lötanschluß oder in Schneid-Klemmtechnik (CPC 464 und 664)
oder
1 × 'Centronics'-Kabelstecker 50pol mit Lötanschluß oder in Schneid-Klemmtechnik (CPC 6128)
1 × Pfostenstiftleiste 2 × 25pol 1/10"
1 × Pfostenbuchsenleiste 2 × 25pol mit Lötanschluß oder in Schneid-Klemmtechnik
5 bis 20 cm Flachkabel 50pol

4 × VG-Buchsenleisten 64pol, gerade
1 × Pfostenstiftleiste 2 × 6pol
1 × Pfostenstiftleiste 2 × 8pol oder
1 × 8fach-DIL-Schalter
5 × Pfostenstiftleiste 2 × 2pol
maximal 12 Steckbrücken
1 × Resettaster

Für Spezialzwecke:
1 × Pfostenstiftleiste 2 × 3pol mit einer Steckbrücke

Jiddisch: Dus helft wie a toiten bankes.
Das hilft wie Schröpfköpfe bei einem Toten.
Das hilft gar nichts.

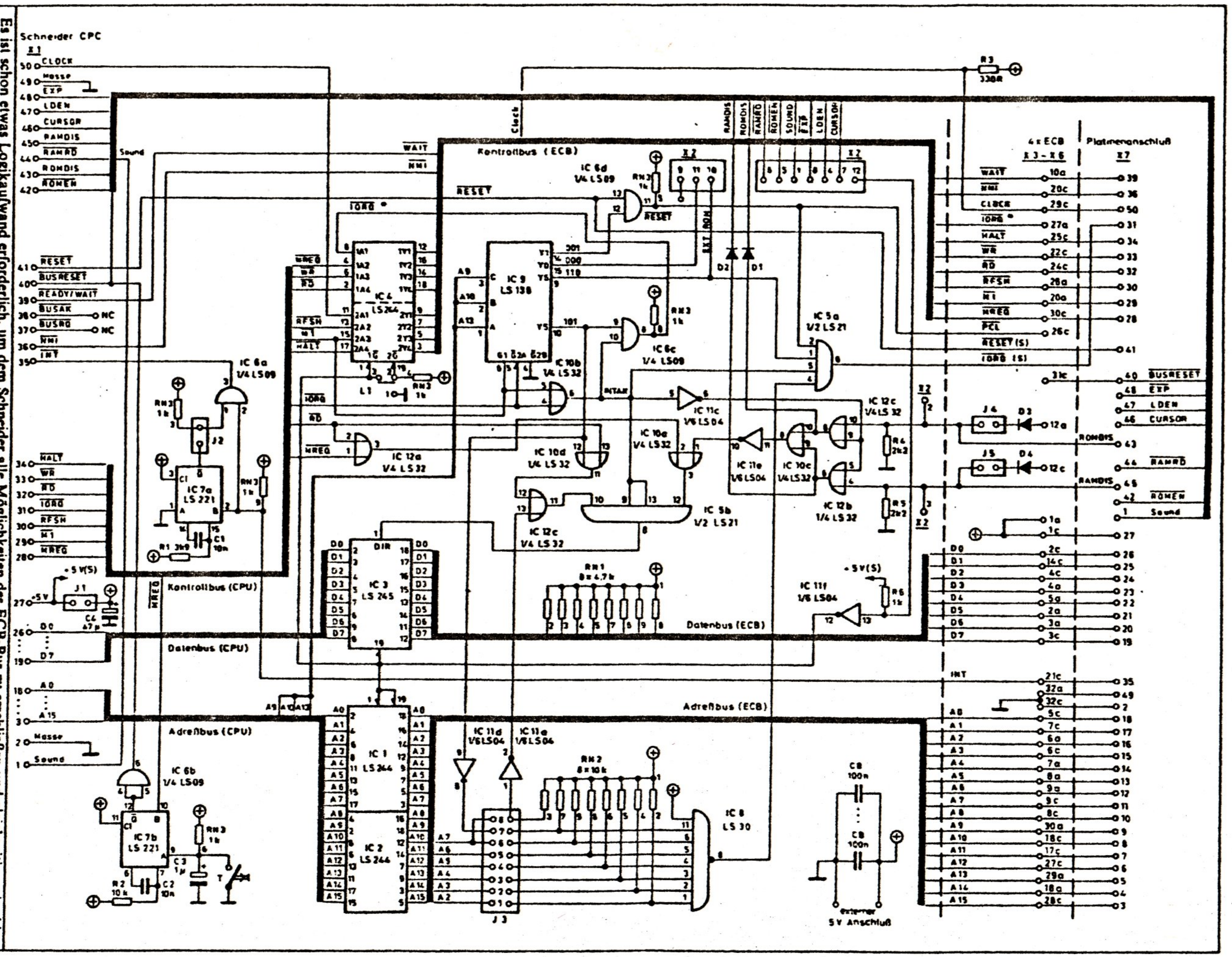
Persisch: Henvañe sir-e baqal-e kasi gosaštan.
Jemandem eine Wassermelone unter die Achsel stecken.
Jemandem schmeicheln.

Manche Leute werden hauptsächlich deshalb für gebildet gehalten, weil ihnen das wenige, was sie wissen, im richtigen Augenblick einfällt.

(Friedrich Julius Stahl)

Spanisch: Hay moros en la costa.
Es sind Mauren (algerische Seeräuber) an der Küste!
Vorsicht, Feind hört mit!

Ungarisch: A hazug embert könnyebb utólérni, mint a sánta kutyát.
Ein Lügner ist schneller zu erreichen als ein hinkender Hund.
Lügen haben kurze Beine.



Es ist schon etwas Logikaufwand erforderlich, um dem Schneider alle Möglichkeiten des ECB-Bus zu erschließen und gleichzeitig die Schneider-Peripherie zu bedienen.

„Jedesmal, wenn ein Mensch lacht, fügt er seinem Leben ein paar Tage hinzu."

Curzio Malaparte
(Pseudonym von Kurt Erich Suckert), italienischer Schriftsteller (1898–1957)

„Die Menschen müssen begreifen, daß sie das gefährlichste Ungeziefer sind, das je die Erde verwüstet hat."

BÖRSE

WER HAT WAS -- WER SUCHT WAS --

Verkaufe:

Computer:	KOMTEK 1S, int. 32 K-Ram (Aufrüstung Räbiger E-Prom)
Drucker :	Epson RX-80
Monitor :	Zenith Data System 12" Grün mit Video-Eingang
Recorder:	ISP Cassette Tape Recorder TRC 5. 2 St.
IC :	Ram's 4116 25 St.
Software:	Diverse-Programme,MON 5,RTTY-Prog,Tebast,Verstärker
Unterla.:	Schaltpläne,ROM-Listings,

Preis : 1400,- DM alles zusammen. Versand möglich.

Von 8 - 16.30 Uhr bin ich unter der Nummer 089/6 11 40 11 und nach 16.30 Uhr unter 089/6 11 59 71 zu erreichen.

im voraus besten Dank

(H. Baldeck)

--

Wer hat groessere oder auch kleinere nuetzliche Programme die mir als Physikstudent 1.Sem. helfen koennten. geschrieben? (Themenberieche Physik I. Mechanik. Waerme. Akustik. Hoehere Mathematik. z.B. Integral- Diff-rechnung. Vektorrechnung. komplexe Zahlen. usw. usw. usw.) Oder wer kann an leicht umschreibbare Basic-Listings solcher Programme anderer Computersysteme kommen? Also schaut mal in Eure Sammlungen und fragt Freunde und wenn Ihr was gefunden habt. schreibt an Holger May. Danke.

Da ich nun ein TRS 80 Mdl IVp besitze möchte ich mein Mdl III verscherbeln.
(Mdl III, 2x40TRK SS DD LWKe)
Hat jemand Interesse?

Ich habe neuerdings einen Druckertreiber für Mdl III/IV der es bewerkstelligt, daß die Umlaute (aus Basic) gedruckt werden - sowohl in NEWDOS wie auch TRSDOS. Allderdings auf dem Epsondrucker. Wer Interesse hat soll es mir sagen.

Ich suche z.Zt 80-Micro Ausgaben: Dez 83, Jan 84 und Jan85 und Feb. 85. Kann mir jemand helfen.

Peter Stevens

Systembefehle

Wort:	AUTO Z,A	BLOAD Datei, Adresse	BSAVE Datei, Adr., Länge	BYE
Bedeutung:	Generiert automatische Zeilennummern, beginnend mit Zeile Z im Abstand A	Lädt eine im Binär- (Maschinencode-) format abgesp. Datei ab angegeb. Adr. in Speicher	Speichert RAM-Inhalt ab »Adr.« im angegebenen Laufwerk unter Namen »Datei«	Rückkehr von BASIC ins Betriebssystem
System:				
ACORN	AUTO			
ALPHATRONIC	AUTO			
APPLE	AUTO	BLOAD[1]	BSAVE	
ATARI				DOS
COLOUR GENIE	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
COMMODORE 8096/C-64/VC20	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIMONS BASIC	AUTO			COLD
EX-BASIC	AUTO			BASIC
DRAGON		CLOAD	CSAVEM	
EPSON HX-20	AUTO	LOADM	SAVEM	MON[4]
IBM PC	AUTO	BLOAD	BSAVE	SYSTEM
LASER 310	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
LASER 3000		BLOAD	BSAVE	
MSX-BASIC	AUTO	BLOAD[1,2]		
ORIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SCHNEIDER CPC	AUTO			¦CPM[5]
SHARP MZ-Serie	AUTO			
SHARP PC-1500[6]	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIRIUS	AUTO			
SPECTRAVIDEO	AUTO	BLOAD	BSAVE	
TANDY C.C.		CLOAD, DLOAD[3]	CSAVEM	
TANDY TRS 80	AUTO			
TI-99/4A	NUMBER			BYE
-EXT-BASIC	NUMBER			BYE
ZX-81 ZX-SPECTRUM	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			

Systembefehle

Wort:	CALL	CHAIN Programm	CLOAD Programm	COMMON A,B,...
Bedeutung:	Ruft Maschinen-Prog. auf u. speichert es ab Adr. A, wobei Parameter P_i übergeb. werden	Wirkt wie LOAD Programm + RUN. Siehe auch COMMON	Lädt ein Programm von Kassette	Die angegeb. Variablen (o. alle b. ALL) werden b. CHAIN-Befehl ins aufgerufene Prog. überg.
System:				
ACORN	CALL [Name / A] P_1,P_2...	CHAIN		
ALPHATRONIC	CALL A (P_1,P_2,...)	CHAIN		COMMON
APPLE	CALL A	CHAIN		
ATARI			CLOAD	
COLOUR GENIE	CALL A		CLOAD	
COMMODORE 8096/C-64/VC20	SYS A		LOAD Programm	
SIMONS BASIC			LOAD Programm	
EX-BASIC	CALL (P_1,P_2,...)		LOAD Programm	
DRAGON			CLOAD	
EPSON HX-20	EXEC A oder USR A(P)			
IBM PC	CALL A(P_1,P_2,...)	CHAIN		COMMON
LASER 310	USR (P)		CLOAD[6]	
LASER 3000	CALL Name			
MSX-BASIC	CALL Name (P_1,P_2,...)		CLOAD	
ORIC	CALL A		CLOAD	
SCHNEIDER CPC	CALL A, P_1, P_2,...	CHAIN		
SHARP MZ-Serie	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SHARP PC-1500		CHAIN	CLOAD	
SIRIUS	CALL A (P_1,P_2,...)	CHAIN		COMMON
SPECTRAVIDEO			CLOAD	
TANDY C.C.	EXEC A oder USR A		CLOAD	
TANDY TRS 80	CALL A (P_1,P_2,...)	CHAIN		COMMON
TI-99/4A	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
-EXT-BASIC	CALL LOAD (Name, Adr., Param.)			
ZX-81 ZX-SPECTRUM	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			

Systembefehle (II)

Wort:	CONT	COPY	CSAVE Programm	DELETE A-B
Bedeutung:	Programmfortsetzung nach einem STOP oder BREAK	Druckt den Bildschirminhalt aus (Hardcopy)	Speichert ein Programm auf Kassette (falls eigener Befehl)	Löscht die Programmzeilen A bis B
System:				
ACORN	CONT			DELETE
ALPHATRONIC	CONT			DELETE
APPLE	CONT[7]			DEL
ATARI	CONT		CSAVE	
COLOUR GENIE	CONT		CSAVE	
COMMODORE 8096/C-64/VC20	CONT		SAVE	
SIMONS BASIC	CONT	COPY	SAVE	
EX-BASIC	CONT	HRDCOPY	SAVE	DEL
DRAGON	CONT		CSAVE	DEL
EPSON HX-20	CONT			DELETE
IBM PC	CONT	PrtSc-Taste drücken		DELETE
LASER 310	CONT		CSAVE	
LASER 3000	CONT			DEL
MSX-BASIC	CONT		CSAVE	
ORIC	CONT			DELETE
SCHNEIDER CPC	CONT			DELETE
SHARP MZ-Serie	CONT			DELETE
SHARP PC-1500	CONT		CSAVE	
SIRIUS	CONT			DELETE
SPECTRAVIDEO	CONT		CSAVE	DELETE
TANDY C.C.	CONT		CSAVE	DEL[8]
TANDY TRS 80	CONT			DELETE
TI-99/4A	CONTINUE			
-EXT-BASIC	CONTINUE			
ZX-81 ZX-SPECTRUM	CONT	COPY		

Systembefehle (III)

Wort:	EDIT N	LIST A-B	LLIST A-B	LOAD Programm
Bedeutung:	Zeile N wird zum Bearbeiten bereitgestellt.[9]	Gibt die Programmzeilen A bis B auf dem Bildschirm aus.	Gibt die Programmzeilen A bis B auf dem Drucker aus.	Lädt ein Programm von Diskette oder Kassette in den Arbeitsspeicher.
System:				
[illegible]		LIST A, B[10]	FX 5,1:LIST A, B	LOAD Programm
[illegible]	EDIT N	LIST A-B	LLIST A-B	LOAD Programm
[illegible]	EDIT N	LIST A, B	PR# 1:LIST A, B	LOAD Programm
[illegible]		LIST A, B	LIST"P:", A, B	LOAD Programm
[illegible]	EDIT N	LIST A-B	LLIST A-B	LOAD Programm
[illegible]		LIST A-B	OPEN 4,4:CMD4:LIST A-B: PRINT #4:CLOSE4	LOAD Programm (Kassette) LOAD Programm, 8 (Diskette)
[illegible]		LIST A-B	OPEN 4,4:CMD4:LIST A-B: PRINT #4:CLOSE4	LOAD
[illegible]		LIST A-B	OPEN 4,4:CMD4:LIST A-B: PRINT #4:CLOSE4	LOAD*
[illegible]	EDIT N	LIST A-B	LLIST A-B	[14]
[illegible]		LIST A-B	LLIST A-B	LOAD Programm, R[13]
[illegible]		LIST A-B	LLIST A-B	LOAD Programm, R[13]
[illegible]		LIST A-B	LLIST A-B	[14]
[illegible]		LIST A-B	PR# 1:LIST A-B	LOAD Programm
[illegible]		LIST A-B	LLIST A-B	LOAD Programm
[illegible]	EDIT N	LIST A-B	LLIST A-B	[14]
[illegible]	EDIT N	LIST A-B	LIST A-B,#8	LOAD Programm
[illegible]		LIST A-B	LIST/P A-B	LOAD Programm
[illegible]	EDIT N	LIST[15]	LLIST A-B	[14]
[illegible]		LIST A-B	LLIST A-B	LOAD Programm, R[13]
SPECTRAVIDEO		LIST A-B	LLIST A-B	LOAD Programm, R[13]
TANDY C.C.	EDIT N	LIST A-B	LLIST A-B	LOAD Programm, R[13]
TANDY TRS 80	EDIT N	LIST A-B	LLIST A-B	LOAD Programm, R[13]
TI-99/4A	EDIT N	LIST A-B	LIST"P":A-B	OLD Programm
-EXT-BASIC	EDIT N	LIST A-B	LIST"P":A-B	OLD Programm
ZX-81 ZX-SPECTRUM		LIST A-B[12]	LLIST A-B	LOAD Programm

Systembefehle (IV)

Wort:	MERGE Programm	NEW	RENUM neu, alt, Abstand	RUN Programm
Bedeutung:	Lädt ein Programm in den Arbeitsspeicher und „mischt" es mit dem vorhandenen. Gleiche Zeilennummern werden vom eingeladenen Programm überschrieben.	Löscht den Arbeitsspeicher und alle Dauerbefehle wie z. B. TRON.	Neunumerierung eines Programms. Beginn der Numerierung bei Zeile alt, Beginn der neuen Numerierung bei neu, mit Zeilenabstand Abstand.	1) RUN Z startet ein vorhandenes Programm ab Zeile Z 2) RUN Programm wirkt wie LOAD + RUN
System:				
ACORN		Dieser Befehl ist bei allen Rechnern gleich. Er kann beim Acorn und bei Simons BASIC mit OLD rückgängig gemacht werden.	RENUMBER	nur RUN
ALPHATRONIC	MERGE		RENUM	1 2
APPLE				1 2
ATARI				1 2
COLOUR GENIE			RENUM	1
COMMODORE 8096/C-64/VC20	POKE 43, PEEK (45)-2 POKE 44, PEEK (46) LOAD Programm POKE 43,1:POKE 44,8			1 2
SIMONS BASIC	MERGE		RENUMBER	1 2
EX-BASIC	MERGE		RENUM	1 2
DRAGON			RENUM	1
EPSON HX-20	MERGE		RENUM	1 2
IBM PC	MERGE		RENUM	1 2
LASER 310				1
LASER 3000				1
MSX-BASIC	MERGE		RENUM	1
ORIC				1
SCHNEIDER CPC	MERGE		RENUM	1 2
SHARP MZ-Serie	MERGE		RENUM	1
SHARP PC-1500	MERGE			1[16]
SIRIUS	MERGE		RENUM	1 2
SPECTRAVIDEO	MERGE		RENUM	1 2
TANDY C.C.			RENUM[17]	nur RUN
TANDY TRS 80	MERGE			1 2
TI-99/4A			RESEQUENCE, neu, Abstand	1
-EXT-BASIC	MERGE		RESEQUENCE, neu, Abstand	1 2
ZX-81 ZX-SPECTRUM	MERGE			1

Heute: Dateiverwaltung/Sequentielle Dateien

Wort:	zum Lesen öffnen	zum Schreiben öffnen	zum Anfügen öffnen	schließen
Bedeutung:	Die Datei namens DAT (der Name ist in D$ gespeichert) wird zum Lesen eröffnet.	Die Datei namens DAT wird zum Beschreiben eröffnet.	Eine Datei wird so eröffnet, daß am Dateiende Daten angefügt werden können (Erweitern).	Der Kanal zur Datei wird wieder geschlossen.
System:				
ACORN	X=OPENIN D$	X=OPENOUT D$		CLOSE#X
ALPHATRONIC	OPEN D$ FOR INPUT AS #1	OPEN D$ FOR OUTPUT AS #1	OPEN D$ FOR APPEND AS #1	CLOSE 1
APPLE	PRINT CHR$(4);"OPEN";D$ PRINT CHR$(4);"READ";D$	PRINT CHR$(4);"OPEN";D$ PRINT CHR$(4);"WRITE";D$	PRINT CHR$(4);"APPEND";D$	PRINT CHR$(4);"CLOSE";D$
ATARI	OPEN#1,4,0,"D:";D$	OPEN#1,8,0,"D:";D$[1]	OPEN#1,9,0,"D:";D$	CLOSE#1
COLOUR GENIE	OPEN"I",1,D$	OPEN"0",1,D$		
COMMODORE 8096/C-64/VC20	OPEN 1,1,0,D$ (Kassette) OPEN 1,8;2,D$;"S,R" (Disk)	OPEN 1,1,1,D$ (Kassette)[2] OPEN 1,8,2,D$;"S,W" (Disk)	OPEN1,8,2,D$;"S,A"	CLOSE1
SIMONS BASIC				
EX-BASIC				
DRAGON	automatisch mit FREAD	automatisch mit FWRITE, vorher Zeiger auf Null setzen	automatisch durch FWRITE	CLOSE1
EPSON HX-20	OPEN"I",#1,D$	OPEN"0",#1,D$		CLOSE#1
IBM PC	OPEN D$ FOR INPUT AS 1	OPEN D$ FOR OUTPUT AS 1	OPEN D$ FOR APPEND AS 1	CLOSE#1
LASER 310	nicht nötig	nicht nötig	nicht möglich	nicht nötig
LASER 3000	PRINT CHR$(4),"OPEN";D$: PRINT CHR$(4),"READ";D$	PRINT CHR$(4),"OPEN";D$: PRINT CHR$(4),"WRITE";D$	PRINT CHR$(4),"APPEND";D$	PRINT CHR$(4),"CLOSE";D$
MSX-BASIC	OPEN D$ FOR INPUT AS 1	OPEN D$ FOR OUTPUT AS 1	OPEN D$ FOR APPEND AS 1	CLOSE #1
ORIC	nicht nötig	nicht nötig	nicht möglich	nicht nötig
SCHNEIDER CPC	OPENIN D$	OPENOUT D$	nicht möglich	CLOSEIN bzw. CLOSEOUT
SHARP MZ-Serie	ROPEN"DAT"	WOPEN"DAT"	nicht möglich	CLOSE
SHARP PC-1500	nicht nötig	nicht nötig	nicht möglich	nicht nötig
SIRIUS	OPEN"I",1,D$	OPEN"0",1,D$		CLOSE#1 bzw. RESET (für alle Dateien gleichzeitig
SPECTRAVIDEO	OPEN D$ FOR INPUT AS 1	OPEN D$ FOR OUTPUT AS 1	OPEN D$ FOR APPEND AS 1 (nur RANDOM-Dateien)	CLOSE#1
TANDY C.C.	OPEN"I",#-1,D$	OPEN"0",#-1,D$	nicht möglich	CLOSE#-1
TANDY TRS 80	OPEN"I",1,D$	OPEN"0",1,D$	OPEN"E",1,D$	CLOSE 1
TI-99/4A	OPEN#1:D$,SEQUENTIAL, DISPLAY,INPUT,VARIABLE	OPEN#1:D$,SEQUENTIAL, DISPLAY,OUTPUT,VARIABLE	OPEN #1:D$,SEQUENTIAL, DISPLAY,APPEND,VARIABLE	CLOSE#1
-EXT-BASIC				
ZX-81 ZX-SPECTRUM[7]	– OPEN#1;"m";1;"DAT"	– OPEN#1;"m";1;"DAT"	– nicht möglich	– CLOSE#1 oder CLEAR#

	Sequentielle Dateien								RANDOM-Dateien
Wort:	Variablen lesen	ein Zeichen lesen	ganze Zeile lesen	Variablen schreiben	Datei-Ende	Katalog	Datei löschen	Datei eröffnen
Bedeutung:	Liest die Texte und Zahlen aus der Datei ein und überschreibt sie auf die Variablen A$,A,...	Liest ein Zeichen aus der Datei (auch Steuerzeichen) und überschreibt es auf X$.	Eine Textzeile wird ohne Berücks. von Trennzeichen bis z. nächst. CR gelesen.	Die Variablenwerte A$,A, ...werden auf die Datei geschrieben	Prüft auf Datei-Ende und gibt als Funktion den Wert "wahr" (=-1 oder +1) bzw. "falsch".	Druckt den Katalog aller registrierten Dateien aus.	Löscht die Datei namens DAT.	Eine Random-Datei mit einer Satzlänge von 50 Byte wird eröffnet.
System:								
ACORN	INPUT#X,A$,A,...	BGET#X,X$		PRINT#X,A$,A,...	EOF#X	CAT	"DELETE"DAT"	
ALPHATRONIC	INPUT#1,A$,A,...	X$=INPUT$(1,#1)	LINE INPUT#1,X$	PRINT#1,A$;A;...[12]	EOF(1)	FILES	KILL"DAT"	OPEN"R",#1,D$,50
APPLE	INPUT A$,A,...	GET X$		PRINT A$;CHR$(13);A;...		CATALOG	DELETE DAT	PRINT CHR$(4);"OPEN";D$;",L50"
ATARI	INPUT#1;A$,A,...	GET#1,X[9]		PRINT#1;A$,A,...	STATUS#1,A (A=136)	über DOS-Menü	über DOS-Menü	
COLOUR GENIE	keine Angaben			keine Angaben				
COMMODORE 8096/C-64/VC20	INPUT#1,A$,A,...	GET#1,X$		PRINT#1,A$,A,...	ST (ST=64)	LOAD"$",8:LIST	OPEN1,8,15:PRINT#1,"S:DAT"	OPEN1,8,2,D$;",L,"+CHR$(50)
SIMONS BASIC						DIR"$"	DISK"S:DAT"	
EX-BASIC							]S:DAT	
DRAGON	FREAD D$;A$,A,...		FLREAD D$;X$	FWRITE D$;A$,A,...	EOF("DAT")	DIR	KILL"DAT"	CREATE D$,50
EPSON HX-20	INPUT#1,A$,A,...	X$=INPUT$(1,#1)	LINE INPUT#1,X$	PRINT#1,A$,A,...[12]	EOF(1)	FILES	KILL"DAT"	OPEN"R",#1,D$
IBM PC	INPUT#1,A$,A,...	X$=INPUT$(1,#1)	LINE INPUT#1,X$	PRINT#1,A$,A,...[12]	EOF(1)	FILES	KILL"DAT"	OPEN"R",#1,D$,50
LASER 310	INPUT#"DAT",A$,A,...			PRINT#"DAT",A$,A,...		nicht möglich	nicht nötig	
LASER 3000	INPUT A$,A,...		INPUT X$	PRINT A$;CHR$(13);A;...		CATALOG	DELETE DAT	PRINT CHR$(4),"OPEN";D$;",L50"
MSX-BASIC	INPUT#1,A$,A,...	X$=INPUT$(1,#1)	LINE INPUT#1,X$	PRINT#1,A$,A,...		Betriebssystem	Betriebssystem	
ORIC	RECAL A$,"DAT",S[14]			STORE A$,"DAT",S[13]		nicht möglich	nicht nötig	
SCHNEIDER CPC	INPUT#9,A$,A,...		LINE INPUT#9,X$	PRINT#9,A$,A,...[12]		CAT	nicht nötig	
SHARP MZ-Serie	INPUT/T A$,A,...			PRINT/T A$,A,...		nicht möglich	nicht nötig	
SHARP PC-1500	INPUT#"DAT";A$,A(*),...[11]			PRINT#"DAT",A$,A(*),...[11]		nicht möglich	nicht nötig	
SIRIUS	INPUT#1,A$,A,...	X$=INPUT$(1,#1)	LINE INPUT#1,X$	PRINT#1,A$;A;...[12]	EOF(1)	FILES	KILL"DAT"	OPEN"R",#1,D$,50
SPECTRAVIDEO	INPUT#1,A$,A,...		LINE INPUT#1,X$	PRINT#1,A$,A,...[12]	EOF(1)	FILES	KILL"DAT"	OPEN D$ AS 1[13]
TANDY C.C.	INPUT#1,A$,A,...			PRINT#1,A$,A,...	EOF(-1)	nicht möglich	nicht nötig	
TANDY TRS 80	INPUT#1,A$,A,...	X$=INPUT$(1,1)	LINE INPUT#1,X$	PRINT#1,A$,A,...[12]	EOF(1)	SYSTEM"DIR"	KILL"DAT"	OPEN"D",1,D$,50
TI-99/4A	INPUT#1,A$,A,...			PRINT#1:A$,A,...	EOF(1)	nicht möglich	DELETE D$	OPEN#1:D$,RELATIVE, DISPLAY,INPUT,FIXED,50
EXT-BASIC			LINPUT#1:X$	wie TI-BASIC[12]				
ZX-81 ZX-SPECTRUM	– INPUT#1;A$,A,...	– X$=INKEY$#1	–	– PRINT#1;A$,A,...	–	– CAT 1	– ERASE"m";"DAT"	–

RANDOM-Dateien

Wort:	Struktur festlegen	positionieren	Lesen	Schreiben
Bedeutung: / System:	Der Satz wird eingeteilt in Felder, die Strings vorgeschriebener Länge aufnehmen	Positioniert den Dateizeiger auf Satz Nr. S (Byte BY)	Liest von Satz S Daten in A$ bzw. die FIELD-Variablen	Die angegebenen oder die FIELD-Variablen werden in Sektor S geschrieben
ACORN		PTR#X=S	wie bei sequentiellen Dateien	
ALPHATRONIC	FIELD 1,12 AS A$,4 AS B$,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
APPLE	automatisch	autom. bei Lesen/Schreiben	PRINT CHR$(4);"READ"; D$;"R";S	PRINT CHR$(4);"WRITE";D$; ";R";S:PRINT A$,A,...
ATARI				
COLOUR GENIE				
COMMODORE 8096/C-64/VC20	automatisch	OPEN2,8,15,"P"+CHR$(2)+[18] CHR$(LO)+CHR$(HI)+CHR$(BY)	wie bei sequentiellen Dateien	wie bei sequentiellen Dateien
SIMONS BASIC				
EX-BASIC				
DRAGON	autom. beim Lesen/Schreiben	autom. bei Lesen/Schreiben	FREAD D$,FROM S,FOR L;A$	FWRITE D$ FROM S,FOR L;A$,A,...
EPSON HX-20	FIELD 1,12 AS A$, 4 AS B$,... DEFFIL 100,ST[14]	autom. bei Lesen/Schreiben	GET#1,S GET%S,A$,A,...[16]	PUT#1,S PUT%S,A$,A,...[16]
IBM PC	FIELD 1,12 AS A$,4 AS B$,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
LASER 110				
LASER 3000	1 Satz wird jeweils durch ein RETURN begrenzt	PRINT CHR$(4),"POSITION"; D$;",R";S	PRINT CHR$(4),"READ";D$; ",R";S;",B";Z[17]	PRINT CHR$(4);"WRITE";D$; ",R";S;",B";Z[17]
MSX-BASIC				
ORIC				
SCHNEIDER CPC				
SHARP MZ-Serie				
SHARP PC-1500				
SIRIUS	FIELD 1,12 AS A$, 4 AS B$,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
SPECTRAVIDEO	FIELD 1,12 AS A$, 4 AS B$,...	autom. bei Lesen/Schreiben	GET#1,S	PUT#1,S
TANDY C.C.				
TANDY TRS 80	FIELD 1,12 AS A$,4 AS B$,...	autom. bei Lesen/Schreiben	GET 1,S	PUT 1,S
TI-99/4A		RESTORE#1,REC S	INPUT#1,REC S:A$,A,...	PRINT#1,REC S:A$,A,...[12]
EXT-BASIC				
ZX-81 ZX-SPECTRUM				

Sequentielle Dateien

Wort:	Besondere Schreibweise	Datei-Ende[13]	Datei-Länge	Datei-Anzahl festlegen
Bedeutung: / System:	(1) schreibt ein Zeichen in die Datei. (2) umgibt Strings mit "" und trennt sie durch Kommas.	Prüft auf Dateiende und gibt als Funktion den Wert "wahr" (=−1 oder +1) bzw. "falsch".	Diese Funktion gibt die Dateilänge in Byte an.	Damit kann die Anzahl gleichzeitig offener Dateien erweitert werden.
ACORN	BPUT#X,X$ (1)	EOF#X	EXT#X	
ALPHATRONIC	WRITE#1,A$,B$ (2)	EOF(1)		
APPLE				MAXFILES N
ATARI	PUT#1,A[14] (1)	STATUS#1,A (A=136)		
COLOUR GENIE				
COMMODORE 8096/C-64/VC20		ST (ST=64)		
SIMONS BASIC				
EX-BASIC				
DRAGON		EOF("DAT")	LOF"DAT"	
EPSON HX-20		EOF(1)	LOF(1)	FILNUM N
IBM PC	WRITE#1,A$,B$,... (2)	EOF(1)	LOF(1)	BASIC/F:N (b. Starten d. Systems)
LASER 110				
LASER 3000				
MSX-BASIC				MAXFILES = N
ORIC				
SCHNEIDER CPC	WRITE#9,A$,B$,... (2)			
SHARP MZ-Serie				
SHARP PC-1500				
SIRIUS	WRITE#1,A$,B$,... (2)	EOF(1)	LOF(1)	BASIC/F:N (b. Starten d. Systems)
SPECTRAVIDEO	WRITE#1,A$,B$,... (2)	EOF(1)	LOF(1)	MAXFILES=N
TANDY C.C.		EOF(−1)		
TANDY TRS 80	WRITE#1,A$,B$,... (2)	EOF(1)		
TI-99/4A		EOF(1)		CALL FILES(N)
EXT-BASIC				
ZX-81 ZX-SPECTRUM				

Stringverarbeitung

Wort:	String(feld) definieren	String deklarieren	String-Länge	Teilstring suchen
Bedeutung:	Ein zweidim. Stringfeld wird definiert (Sonderbedeutungen bei Atari und Sinclair).	Die Variablen, die mit A bis Z beginnen, werden als Strings definiert.[1]	Bestimmt die Länge eines Strings (=ganze Zahl zwischen 0 und – meistens – 255).	Sucht den Such-String S$ im String Z$ ab Pos. P. Resultat 0 = nicht gefunden.
System:				
ACORN	DIM (A$(5,10)	–	LEN(Z$)	INSTR(Z$,S$,P)
ALPHATRONIC	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)
APPLE	DIM A$(5,10)	–	LEN(Z$)	–
ATARI	DIM A$(10)[2]	–	LEN(Z$)	–
COLOUR GENIE	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)[3]
COMMODORE 8096/C-64/VC20	DIM A$(5,10)	–	LEN(Z$)	–
SIMONS BASIC	DIM A$(5,10)	–	LEN(Z$)	PLACE(Z$,S$)
EX-BASIC	DIM A$(5,10)	–	LEN(Z$)	INSTR(Z$,S$,P)
DRAGON	DIM A$(5,10)	–	LEN(Z$)	INSTR(Z$,S$,P)
EPSON HX-20	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)
IBM PC	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)
LASER 310	DIM A$(5,10)	–	LEN(Z$)	–
LASER 3000	DIM A$(5,10)	–	LEN(Z$)	–
MSX-BASIC	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)
ORIC	DIM A$(5,10)	–	LEN(Z$)	–
SCHNEIDER CPC	DIM A$(5,10)	–	LEN(Z$)	–
SHARP MZ-Serie	DIM A$(5,10)	–	LEN(Z$)	–
SHARP PC-1500	–	–	LEN(Z$)	–
SIRIUS	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)
SPECTRAVIDEO	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	–
TANDY C.C.	DIM A$(5,10)	–	LEN(Z$)	–
TANDY TRS 80	DIM A$(5,10)	DEFSTR A,Z	LEN(Z$)	INSTR(P,Z$,S$)
TI-99/4A	DIM A$(5,10)	–	LEN(Z$)	POS(Z$,S$,P)
-EXT-BASIC	DIM A$(5,10)	–	LEN(Z$)	POS(Z$,S$,P)
ZX-81	–	–	–	–
ZX-SPECTRUM	DIM A$(5,10)[4]	–	LEN(Z$)	–

Stringverarbeitung

Wort:	Teilstring links	Teilstring rechts	Teilstring Mitte	Strings verknüpfen
Bedeutung:	Kopiert aus dem String Z$ die ersten zehn Zeichen heraus.	Kopiert aus dem String Z$ die letzten zehn Zeichen heraus.	Kopiert aus dem String Z$ fünf Zeichen ab Position 3 (also die Zeichen 3 bis 7).	Verkettet zwei Strings miteinander.
System:				
ACORN	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
ALPHATRONIC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)[5]	A$ + B$
APPLE	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
ATARI	Z$(1,10)	Z$(LEN(Z$)–9,LEN(Z$))	Z$(3,7)	A$(LEN(A$)+1)=B$
COLOUR GENIE	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
COMMODORE 8096/C-64/VC20	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
SIMONS BASIC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
EX-BASIC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
DRAGON	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
EPSON HX-20	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)[5]	A$ + B$
IBM PC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)[5]	A$ + B$
LASER 310	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
LASER 3000	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
MSX-BASIC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)[5]	A$ + B$
ORIC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
SCHNEIDER CPC	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
SHARP MZ-Serie	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
SHARP PC-1500	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
SIRIUS	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
SPECTRAVIDEO	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
TANDY C.C.	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
TANDY TRS 80	LEFT$(Z$,10)	RIGHT$(Z$,10)	MID$(Z$,3,5)	A$ + B$
TI-99/4A	SEG$(Z$,1,10)	–	SEG$(Z$,3,5)	A$ & B$
-EXT-BASIC	SEG$(Z$,1,10)	–	SEG$(Z$,3,5)	A$ & B$
ZX-81	–	–	–	–
ZX-SPECTRUM	Z$(1 TO 10)	–	Z$(3 TO 7)	–

Stringverarbeitung

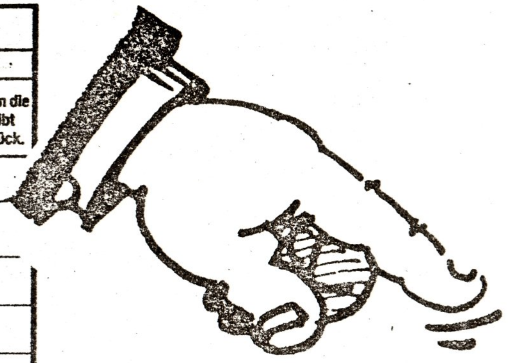
Wort:	Zeichen in Code	Code in Zeichen	String in Zahl	Zahl in String
Bedeutung:	Verwandelt das erste Zeichen des Strings Z$ in den entsprechenden ASCII-Code.	Verwandelt die Zahl Z in das dem ASCII-Code entsprechende Zeichen.	Verwandelt den String Z$ in eine Zahl, sofern und soweit dies möglich ist.	Verwandelt die Zahl Z in einen String mit Vorzeichen.
System:				
ACORN	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
ALPHATRONIC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
APPLE	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
ATARI	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
COLOUR GENIE	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
COMMODORE 8096/C-64/VC20	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
SIMONS BASIC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
EX-BASIC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
DRAGON	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
EPSON HX-20	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
IBM PC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
LASER 310	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
LASER 3000	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
MSX-BASIC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
ORIC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
SCHNEIDER CPC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
SHARP MZ-Serie	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
SHARP PC-1500	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
SIRIUS	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
SPECTRAVIDEO	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
TANDY C.C.	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
TANDY TRS 80	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
TI-99/4A	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
EXT-BASIC	ASC(Z$)	CHR$(Z)	VAL(Z$)	STR$(Z)
ZX-81 ZX-SPECTRUM	CODE Z$	CHR$ Z	VAL(Z$)	STR$(Z)

Stringverarbeitung II

Wort:	Zeichen vervielfachen	Leerstring erzeugen	Strings löschen	Stringspeicher angeben
Bedeutung:	Erzeugt einen String, der N-mal die Zeichenkette in Anführungszeichen enthält.	Erzeugt einen String aus N Leerzeichen.	Löscht alle String-Variablen und schafft N freie Speicherplätze (in Bytes).	Gibt an, wieviele Bytes im Arbeitsspeicher für Strings noch frei sind.
System:				
ACORN	STRING$(N,"-0-")		CLEAR N	
ALPHATRONIC	STRING$(N,"-0-")[7]	SPACE$(N)	CLEAR N	FRE (X$)[8]
APPLE			CLEAR N	
ATARI			CLR	
COLOUR GENIE	STRING$(N,"-0-")[7]		CLEAR N	
COMMODORE 8096/C-64/VC20			CLR	
SIMONS BASIC	DUP("-0-",N)		CLR	
EX-BASIC	STRING$(N,"-0-")[7]		CLR	
DRAGON	STRING$(N,"-0-")[7]		CLEAR N	FRE$[9][8]
EPSON HX-20	STRING$(N,"-0-")[7]	SPACE$(N)	CLEAR N	FRE("")[8]
IBM PC	STRING$(N,"-0-")	SPACE$(N)	CLEAR N	FRE("")[8]
LASER 310	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
LASER 3000			CLR	FRE(A$)[8]
MSX-BASIC	STRING$(N,"-0-")[7]	SPACE$(N)	CLEAR N	FRE("")[8]
ORIC			CLEAR N	FRE("")
SCHNEIDER CPC	STRING$(N,"-0-")	SPACE$(N)	CLEAR N	FRE("")[8]
SHARP MZ-Serie			CLR	
SHARP PC-1500			CLEAR N	
SIRIUS	STRING$(N,"-0-")	SPACE$(N)	CLEAR N	FRE("")[8]
SPECTRAVIDEO	STRING$(N,"-0-")	SPACE$(N)	CLEAR N	FRE("")[8]
TANDY C.C.			CLEAR N	
TANDY TRS 80	STRING$(N,"-0-")	SPACE$(N)	CLEAR N	FRE("")
TI-99/4A	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
-EXT-BASIC	RPT$("-0-",N)			
ZX-81 ZX-SPECTRUM			CLEAR N	

Stringverarbeitung II

Wort:	Spezialumwandlungen 1	Spezialumwandlungen 2	Dezimal in Hex	Dezimal in Oct
Bedeutung:	Verwandelt spezielle Strings in Zahlen (ganz, einfach genau, doppelt genau).	Verwandelt Zahlen (ganz, einfach genau, doppelt genau) in Strings.	Wandelt eine Dezimalzahl in die entspr. Hexadezimalzahl um u. gibt Ergebn. als String zurück.	Wandelt eine Dezimalzahl in die entspr. Oktalzahl um und gibt das Ergebnis als String zurück.
System:				
ACORN	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
ALPHATRONIC	CVI(Z$), CVS(Z$), CVD(Z$)	MKI$(Z%), MKS$(Z), MKD$(Z#)	HEX$(X)	OCT$(X)
APPLE	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
ATARI	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
COLOUR GENIE	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
COMMODORE 8096/C-64/VC20	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIMONS BASIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
EX-BASIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
DRAGON			HEX$(X)	
EPSON HX-20	CVI(Z$), CVS(Z$), CVD(Z$)	MKI$(Z%), MKS$(Z), MKD$(Z#)	HEX$(X)	OCT$(X)
IBM PC	CVI(Z$), CVS(Z$), CVD(Z$)	MKI$(Z%), MKS$(Z), MKD$(Z#)	HEX$(X)	OCT$(X)
LASER 310	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
LASER 3000	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
MSX-BASIC			HEX$(X)	OCT$(X)*
ORIC			HEX$(X)	
SCHNEIDER CPC			HEX$(X)*	
SHARP MZ-Serie	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SHARP PC-1500	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
SIRIUS	CVI(Z$), CVS(Z$), CVD(Z$)	MKI$(Z%), MKS$(Z), MKD$(Z#)	HEX$(X)	OCT$(X)
SPECTRAVIDEO	CVI(Z$), CVS(Z$), CVD(Z$)	MKI$(Z%), MKS$(Z), MKD$(Z#)	HEX$(X)	OCT$(X)*
TANDY C.C.	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
TANDY TRS 80	CVI(Z$), CVS(Z$), CVD(Z$)	MKI$(Z%), MKS$(Z), MKD$(Z#)	HEX$(X)	OCT$(X)
TI-99/4A	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
-EXT-BASIC	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			
ZX-81 ZX-SPECTRUM	Entsprechende Befehle sind bei diesem Rechner nicht vorhanden			



CLUB-Bücherei

Gerald Dreyer hat mehrere Ausgaben der
GENIE-DATA
gestiftet. Die Hefte sind in der CLUB-
Bibliothek und können ausgeliehen werden.

Ein Computer läßt sich nicht täuschen

Auf der New Yorker Weltausstellung steht ein Computer, der jedem Besucher – auf Knopfdruck – Namen und Herkunft sagt.

Auch der Indianer Tom To-peeh aus Kansas zahlt seinen Dollar, drückt den Knopf, und der Computer sagt: »DU HEISST TOM TO-PEEH, BIST INDIANER, KOMMST AUS KANSAS, DEIN LETZTER ZUG GEHT UM 18 UHR 30.«
Tom To-peeh hält das für einen Schwindeltrick und beschließt, den Computer hereinzulegen. Nach langem Suchen findet er einen Neger-Schuhputzer, der bereit ist, ihm seine Uniform und seinen Putzkasten zu borgen; er schmiert Gesicht und Hände mit Schuhcreme schwarz, zieht die Uniform an und tritt, den Putzkasten in der Hand, wieder vor den Computer.
Er wirft seinen Dollar ein, drückt den Knopf – der Computer sagt: »DU HEISST TOM TO-PEEH, BIST INDIANER, KOMMST AUS KANSAS, DEIN LETZTER ZUG GING UM 18 UHR 30. – UND WENN DU DEN QUATSCH MIT DER VERKLEIDUNG NICHT GEMACHT HÄTTEST, HÄTTEST DU IHN NICHT VERPASST!«

DAS GESETZ VON MURPHY

JEDER VON UNS WIRD TAEGLICH - BEI DER ARBEIT, BEI DER AUSUEBUNG DES HOBBYS, WIE AUCH IM GANZ PRIVATEN BEREICH - MIT DEM GESETZ VON MURPHY KONFRONTIERT. DOCH BLEIBT DIES MEISTENS UNBEWUSST, WEIL DIESES GESETZ - OBWOHL ES WEITAUS UNIVERSALER ALS DAS OHMSCHE GESETZ IST - ZUMINDEST BEI UNS FAST UNBEKANNT BLIEB.
VON DER EXISTENZ DES GESETZES VON MURPHY WUSSTE ICH SEIT LAENGEREM, UND ICH HALTE ES FUER MEINE PFLICHT, DIESE ERKENNTNISSE HIERMIT WEITERZUGEBEN - ERKLAEREN SIE BEISPIELSWEISE ENDLICH, WARUM DER CHEF AUSGERECHNET IMMER DANN DEN RAUM BETRITT, WENN MAN NACH STUNDENLANGER, INTENSIVER ARBEIT DEN ARBEITSPLATZ FUER DAS NAECHSTE PROJEKT AUFGERAEUMT HAT UND EINEN AUGENBLICK LANG NACHDENKLICH AUS DEM FENSTER SCHAUT.

1. EDSEL MURPHY

DER AMERIKANER EDSEL MURPHY ENTDECKTE UND FORMULIERTE ALS ERSTER DIE
" GESETZMAESSIGKEITEN DES STETIGEN AERGERNISSES "
DIE TATSACHE, DASS IHM IN SPAETREREN JAHREN DER SICHERE PLATZ IN DER RUHMESHALLE GROSSER FORSCHER UND ENTDECKER VERSAGT BLIEB, KANN NUR ALS AUSWIRKUNG SEINES EIGENEN GESETZES ERKLAERT WERDEN, DESSEN ORGINALFASSUNG LAUTET:

IF ANYTHING CAN GO WRONG, IT WILL

DIES KANN UEBERTRAGEN WERDEN:

FALLS IRGENDETWAS SCHIEFGEHEN KANN, DANN GEHT ES SCHIEF.

ODER IN EXAKTER MATHEMATISCHER FORM: 1 + 1 <-> 2
WORIN <-> DAS MATHEMATISCHE SYMBOL FUER "KAUM JEMALS" IST.

ZUR ERLAEUTERUNG DIESES UNIVERSALEN GESETZES WERDEN IM FOLGENDEN EINIGE ANWENDUNGSBEISPIELE AUS DEM BEREICH DER ELEKTRONIK ANGEBOTEN.

2. DAS GESETZ VON MURPHY IN FORSCHUNG UND ENTWICKLUNG

- JE HARMLOSER EINE KONSTRUKTIONSAENDERUNG ERSCHEINT, DESTO WEITTRAGENDER SIND IHRE TATSAECHLICHEN FOLGEN.
- DIE NOTWENDIGKEIT FUER EINE GROESSERE KONSTRUKTIONSAENDERUNG WAECHST MIT DEM NAEHERUECKEN DES FABRIKATIONSBEGINNS.
- KOPIERMASCHINEN ZERMANGELN IMMER DIE ORGINALZEICHNUNGEN, VON DENEN NOCH NICHT EINMAL EINE KOPIE VORHANDEN IST.
- DIE WAHRSCHEINLICHKEIT, DASS EIN MASS IN EINER ZEICHNUNG VERGESSEN WIRD IST DIREKT PROPORTIONAL ZU SEINER WICHTIGKEIT.
- JEDER FEHLER DER SICH IN EINE BERECHNUNG EINSCHLEICHEN KANN WIRD DIES MIT SICHERHEIT TUN; UND ZWAR SO, DASS MAN VOELLIG VON VORNE BEGINNEN MUSS.
- IN JEDER BELIEBIGEN BERECHNUNG IST IMMER DIEJENIGE ZAHL DIE URSACHE VON FEHLERN, DIE AM OFFENSICHTLICHSTEN RICHTIG WAR.
- DEZIMALPUNKTE SIND IMMER AN DER FALSCHEN STELLE, VORZEICHEN VERTAUSCHT
- EINEN HINWEIS, DEN MAN IN EINEM BUCH, LEXIKON, ORDNER ODER AEHNLICHEM SUCHT, FINDET MAN AUF DER LETZTEN SEITE, WENN MAN VON VORNE ANFAENGT ZU SUCHEN, UND AUF DER ERSTEN SEITE, WENN MAN SCHLAUERWEISE HINTEN ANFAENGT.
- EINE TOTSICHERE SCHALTUNG FUNKTIONIERT NIE.
- EINE OSZILLATORSCHALTUNG IST DURCH NICHTS ZUM SCHWINGEN ZU BEWEGEN; EINE VERSTAERKERSCHALTUNG DAGEGEN SCHWINGT SOFORT.
- EIN DURCH EINE ULTRAFLINKE SICHERUNG GESCHUETZTER TRANSISTOR SCHUETZT DIE SICHERUNG, INDEM ER ZUERST "HOCHGEHT".
- EIN SEIT WOCHEN FUNKTIONIERENDER LABORAUFBAU VERSAGT PLOETZLICH, WENN DER CHEF ODER EINEM KUNDEN VORGEFUEHRT WERDEN SOLL. (DIESER TEILASPEKT DES GESETZES VON MURPHY IST ALS "VORFUEHREFFEKT" BEKANNT)

3. DAS GESETZ VON MURPHY IN FERTIGUNG UND SERVICE

- JE DRINGENDER MAN EIN BAUTEIL BENOETIGT, DESTO SCHWIERIGER LAESST ES SICH BESCHAFFEN.
- BENOETIGT MAN FUER EIN PROJEKT N BAUELEMENTE, DANN SIND N-1 AUF LAGER
- IST EIN BESTIMMTER KAPAZITAETSWERT ERFORDERLICH, SO IST DIESER WERT NICHT VORRAETIG; AUSSERDEM LAESST ER SICH WEDER DURCH PARALLEL- NOCH DURCH SERIENSCHALTUNG VORHANDENER WERTE REALIESIREN.
- AUSTAUSCHTEILE SIND IN DER PRAXIS NICHT AUSTAUSCHBAR.
- EIN AUF RICHTIGE LAENGE ABGESCHNITTNER DRAHT ERWEISST SICH IMMER ALS ZU KURZ.
- BAUTEILE, DIE KEINESFALLS FALSCH EINGEBAUT WERDEN DUERFEN, UND DIE AUCH GARNICHT FALSCH EINGEBAUT WERDEN KOENNEN, SIND ES AM SCHLUSS DOCH
- AUSGERECHNET DAS ZERBRECHLICHSTE BAUTEIL LAESST MAN FALLEN.
- DIE AUSFALLWAHRSCHEINLICHKEIT EINES BAUTEILS ODER EINER BAUGRUPPE IST UM SO HOEHER, JE SCHWIERIGER DIE REPARATUR ODER EIN AUSTAUSCH DURCHGEFUEHRT WERDEN KANN.
- EIN WERKZEUG DAS VERSEHENTLICH IN EIN GERAET FAELLT, LANDET UNWEIGERLICH AN DER STELLE, WO ES DEN GROESSTEN SCHADEN ANRICHTEN KANN. (DIESER SPEZIALFALL DES GESETZES VON MURPHY IST AUCH ALS "SELEKTIVE GRAVITATION" BEKANNT)
- WENN EIN GERAET KOMPLETT MONTIERT IST, FINDET MAN NOCH UEBRIGE BAUTEILE AUF DER WERKBANK.
- WENN MAN IN EINEM GERAET MIT AUSSETZFEHLERN EIN OFFENSICHTLICH DEFEKTES BAUTEIL ERSETZT HAT, SO TRITT DER FEHLER WIEDER AUF, SOBALD DAS GERAET IM EINSATZ IST.

4. DAS GESETZ VON MURPHY UND DIE DATENVERARBEITUNG

- TEILPROGRAMME, DIE EINZELN IN ORDNUNG SIND, FUNKTIONIEREN ZUSAMMEN GELADEN NICHT MEHR.
- JE EINFACHER EIN STRUKTOGRAMM, DESTO KOMPLIZIERTER DIE UEBERSETZUNG IN EINE PROGRAMMIERSPRACHE.
- EIN PROGRAMM VERHAELT SICH IMMER ANDERS ALS DAS STRUKTOGRAMM.
- DIE VERBINDUNG EINER DATENFERNUEBERTRAGUNG, BRICHT IMMER IN DIESEN MOMENT ZUSAMMEN, WENN DIE SYNCHRONISATION ZU ENDE IST, UND DIE ERSTEN DATEN KOMMEN.
- PROGRAMMLISTEN AUS ZEITSCHRIFTEN ODER FACHBUECHERN, HABEN GRUNDSAETZLICH FEHLER
- DER SCHREIBSCHUTZ WIRD IMMER BEI DER ORGINAL DISKETTE VERGESSEN.
- PROGRAMME, DIE ERWORBEN WERDEN, LAUFEN ERST NACH GROESSEREN AENDERUNGEN, UND DANN NICHT SO, WIE DAS ORGINAL.
- BEI EINER DATENBANK IST DAS SCHLUESSELWORT UMGEKEHRT PROPORTIONAL ZUR GEHEIMHALTUNG.
- DAS SCHLUESSELWORT EINER DATENBANK WIRD GRUNDSAETZLICH VERGESSEN
- KOPIERGESCHUETZTE PORGRAMME SIND GANZ EINFACH ZU KOPIEREN.

5. BEMERKUNGEN

ETLICHE DER AUFGEFUEHRTEN BEISPIELE, WIE VOR ALLEM DER " VOHRFUEHREFFEKT ", SIND DEN MEISTEN VON UNS AUS DER BERUFSPRAXIS BESTENS VERTRAUT; IM UEBRIGEN SPRECHEN WIR AUCH VON DER " TUECKE DES OBJEKTS ". DIES ALLES DEUTET DARAUF HIN, DASS DIE KENNTNIS DES GESETZES, WELCHES EDSEL MURPHY FORMULIERTE, IM DEUTSCHEN SPRACHBEREICH ZUMINDEST LATENT SCHON VORHANDEN WAR.

(MANFRED HELD)

Folgerungen aus Murphys Gesetz

Guttersons Gesetze
Jedes Programmierprojekt, das gut anfängt, endet schlecht.
Jedes Programmierprojekt, das schlecht anfängt, endet noch schlechter.

Kleinbrunners Folgerungen
Wenn eine Programmieraufgabe leicht aussieht, ist sie schwer.
Wenn eine Programmieraufgabe schwer aussieht, ist sie unlösbar.

Munbrights Gesetze
Jedes Programm ist es veraltet, wenn es läuft.
Jedes Programm kostet mehr und dauert länger.
Wenn ein Programm nützlich ist, muß es geändert werden.
Wenn ein Programm unnütz ist, muß es dokumentiert werden.
Jedes Programm ist ein bißchen länger, als der verfügbare Speicher es zuläßt.
Der Wert eines Programms ist umgekehrt proportional dem von ihm verbrauchten Papier.
Die Komplexität eines Programms wächst solange, bis sie die Fähigkeiten des Programmierers übersteigt, der es unterhalten soll.
Der unangenehmste Fehler wird erst entdeckt, wenn das Programm mindestens 6 Monate im Einsatz ist.
Maschinenunabhängigen Code gibt es nicht.
Zusätzliche Programmierer, die zur Einhaltung eines Termins eingesetzt werden, verzögern den Termin noch weiter.
Softwareprobleme sind umso größer, je weniger Zeit man zu ihrer Behebung zur Verfügung hat.

Farvours Gesetz
Es ist immer *noch* ein Fehler vorhanden.

Brunks Gesetz
Wenn ein Listing einen Anfang hat, hat es auch ein Ende.

Zeppelmiers Folgerung
Die letzten vier Seiten eines kritischen Listings sind immer verloren gegangen.

Penningtons Beobachtung
Die Wahrscheinlichkeit, daß ein Programm die gestellten Anforderungen erfüllt, ist umgekehrt proportional dem Vertrauen des Programmierers in seine eigenen Fähigkeiten.

Schrödingers Gesetze der Programmierung
Zwei Fehler in einem Programm heben sich beim Probelauf in ihrer Wirkung auf.
Im späteren Einsatz des Programms ergänzen sie sich, um den größtmöglichen Schaden anzurichten.

Schrödingers Erkenntnisse
I. Man konstruiere jeden Programmteil so kurz, daß nur ein Fehler auftauchen kann.
II. So kurze Programmteile gibt es in der Praxis nicht.
III. Die überprüften Bestandteile eines Programms laufen garantiert nicht zusammen.

Hergeröders Gesetz zur Kürzung eines Programms
Werden in einem Programm alle vermeintlich unwichtigen Zeilen gelöscht, läuft das Programm nicht mehr.

Ableitungen aus Hergeröders Gesetz
1. Unter den unwichtigen Zeilen war mindestens eine wichtige.
2. Es läßt sich unmöglich rekonstruieren, welche Zeile dies war.
3. Beim Versuch, das gekürzte Programm zum Laufen zu bringen, wird es bis zum vollkommenen Kollaps getrieben.
4. Kurz vor dem Zusammenbruch zerstört das fehlerhafte Programm seinen Urcode.
5. Der Programmierer kann sich beim erneuten Schreiben des Programms unmöglich an die wichtigsten Algorithmen erinnern.

Zwergmanns allgemeingültigen Naturgesetze
Jedes Unterprogramm strebt danach, zum Hauptprogramm zu werden.
Je länger ein Programmierer ohne Zwischenspeichern tippt, desto größer ist die Wahrscheinlichkeit für einen Stromausfall.
Mit jedem Zwischenspeichern bei seiner Arbeit steigt die Wahrscheinlichkeit für einen irreparablen Diskettenfehler.
Jede Fehler-Erkennungs-Routine trifft früher oder später auf einen Fehler, den sie nicht kennt. Dieser Fehler veranlaßt die Routine, alles das zu machen, was ihre Erbauer verhindern wollten.

Klingers Daten-Gesetze
Jede Datenverwaltung oder -verarbeitung hat die Aufgabe, möglichst viele Daten im unpassendsten Augenblick auf verheerendste Weise zu zerstören.

1. Ableitung: Je umfassender die Zerstörung desto weniger Vorwarnungen gibt es.

2. Ableitung: Drei Ereignisse charakterisieren diesen Vorgang:
- in letzter Zeit lief alles glatt und ohne Fehler
- im Moment der Zerstörung meldet das System absolut nichts
- nach einer alles umfassenden Zerstörung meldet sich das System als wäre nichts gewesen
- nach einem kleineren Zwischenfall fängt irgendetwas an zu rauchen oder zu klingeln, um möglichst große Hektik zu verursachen

Erkenntnis: Je ruhiger ein System, desto unruhiger sollte der Betreiber sein.

Mareks Konfusionsregeln
Verläßt ein Programmierer während des Ausdruckens eines wichtigen Listings für fünf Minuten den Raum, so passieren in diesem Moment mit größter Wahrscheinlichkeit folgende Dinge:
- das Papier bleibt hängen oder
- der Papierstapel ist aufgebraucht oder
- das schon beschriebene Papier wird vom Drucker noch mal eingezogen ("Epsonsche Rekursion")
- das Farbband verhakt sich
- es wird ein falscher Steuercode zum Drucker gesandt
- das Drucker-Interface beschließt, seinen Geist aufzugeben

CLUB 80 - Programmbibliothek exklusiv

Liebe Clubfreunde!

Auch in diesem Info kann ich euch wieder eine Menge neuer Programme vorstellen. Außerdem möchte ich euch noch auf ein paar ganz besondere Leckerbissen in der Bibliothek hinweisen!

Schon seit einiger Zeit sind drei moderne Programmiersprachen in der Bibliothek für jedes Mitglied verfügbar. Es handelt sich dabei um Implementierungen von LISP, PILOT und LOGO, die alle in MicroBO - Heften abgedruckt waren. Natürlich beinhalten diese Minimalsysteme nicht alle möglichen Sprachelemente, aber um in eine Sprache einmal hineinzuschnuppern (z.B. bevor man sich einen teueren Interpreter oder Compiler zulegt) reicht es allemal!

Unser Clubmitglied Klaus Herman hat mir in letzter Zeil eine ganze Menge Utilitys für das Model 4 von Tandy geschickt, die im letzten und in diesem Info in die Bibliothek aufgenommen wurden. Ich hoffe, daß die Programmsammlung damit auch für die Besitzer des Model 4 (und das sind inzwischen schon einige) interessanter wird.

Gerald Schröder, der sich ja schon ein paarmal durch überdurchschnittliche Beiträge zur Bibliothek (z.B. Übersetzung des Handbuches zur SuperUtility +) ausgezeichnet hat, ist auch diesmal wieder mit zwei selbstgeschriebenen Programmen dabei. Zum einen hat er einen schon fast professionellen Grafikeditor geschrieben (BASIC compiliert), der durch seine Vielzahl an Möglichkeiten glänzt, zum andern hat er das Brettspiel RISIKO auf unserem Computer installiert. Die dazu benötigten Weltkarten hat er übrigens mit seinem selbstgeschriebenen Grafikeditor erstellt!

Noch jemand hat sich diesmal besonders hervorgetan. Walter Schäfer hat zwei selbstgeschriebene Denkspiele (NIMM und MASTERMIND) beigesteuert, die durch ihre gute Grafik auffallen.

Aber natürlich waren auch diesmal wieder viel mehr Mitglieder als die hier genannten, an dem rapiden Wachstum der Softwaresammlung schuld. Sie alle hier aufzuzählen erspare ich mir und begnüge mich damit, ihnen im Namen des Club's zu danken!

Das letzte Mal!

Mit 37 Neuzugängen und einigen speziellen Tips ist die Rubrik Programmbibliothek exklusiv heute wieder einmal recht groß geraten. Dieser Umstand zeigt, daß entgegen aller Abschaffungsversuche einzelner Mitglieder, die Softwaresammlung ihren festen Platz im CLUB 80 behauptet hat. Nicht umsonst ist sie im Laufe ihres Bestehens auf fast 300 Titel gewachsen!
Dies ist nun das letzte Clubinfo, in welchem ich für die Programmbibliothek des CLUB 80 verantwortlich zeichne. Ich habe einen Nachfolger in diesem Amt gefunden und bin sicher, er wird die Bibliothek genauso gut, vielleicht sogar besser betreuen als ich!
Ich hoffe, ihr werdet ihm genausoviel Unterstützung zukommen lassen wie mir, damit die Programmbibliothek als eines der lebendigsten und wichtigsten Elemente des CLUB 80 noch lange weiterbesteht!
Ich möchte mich nocheinmal für eure Mitarbeit bedanken und meinem Nachfolger, Klaus Jürgen (Kajot) Mühlenbein viel Glück und Erfolg bei seiner Arbeit wünschen, Euer

Hartmut Obermann

ADRESS-100 — A 0229 A 01 /BAS

Adressenverwaltung für bis zu 100 Datensätze. Speicherung auf Disk!

ADRESS-CAS — A 0232 A 01 /BAS

Adressenverwaltungprogramm mit recht komfortablen Möglichkeiten. Speicherung der Daten auf Kassette!

ADVENTURE — D 0234 A 02 /BAS

Wie der Name schon sagt, ein Adventure-Spiel. Worum es geht, weiß ich nicht (ich interessiere mich nict für diese Art Spiele), vielleicht bekommt es jemand von euch heraus!? Englisch!

AUTORENNEN — G 0236 A 01 /CMD

Ein ziemlich einfaches Geschicklichkeitsspiel mit mittelmäßiger Grafik.

Kalender — S 0237 A 02 /BAS

Noch ein Kalenderdruckprogramm, allerdings ein recht ansprechendes! Für alle Drucker geeignet!

Vergleich — S 0238 A 03 /BAS

Dieses Programm von Jens Neueder ermöglicht den Vergleich von Diskfiles mit Ausdruck der Unterschiede. (s. Clubinfo Nr. 10 Seite 43/44)

DATVAR — S 0240 B 03 /CMD

Dieses Hilfsprogramm listet auf Tastendruck, auch mitten in einem laufenden BASIC-Programm, alle gültigen Variablen und Felder sowie deren Inhalte. Der BASIC-Bildschrimaufbau wird dabei nicht beschädigt!!

BARGRAPH — M 0241 A 01 /BAS

Demoprogramm zur Erstellung horizontaler Balkendiagramme. Leicht zu ändern und in eigene Programme einzubauen!

EXECUTE — S 0243 B 02 /CMD

Menüprogramm zum Aufruf von Programmen mittels Tastendruck.

DECIPOKE S 0244 A 01 /BAS --

Hilfsprogramm zur Umwandlung von Maschinenroutinen in DATA-Zeilen. Anleitung bzw. Artikel in englisch aus '80 US' kann geliefert werden!

Artillery G 0245 A 01 /BAS --

Mit einem Artilleriegeschütz soll man einen unsichtbaren Feind treffen (Koordinaten bekannt). Dabei muß man sowohl die Geschoßgeschwindigkeit als auch den Abschußwinkel angeben. Keine Grafik!!

CHECKBOOK M 0246 A 01 /BAS

Dieses Programm hilft beim Verwalten des, durch Frau, Freundin und Computer sowieso schon ziemlich abgemagerten, Guthabens. Vielleicht wird es dadruch mehr!?

Vokabeltrainer S 0247 A 02 /BAS

Praktisches Hilfsprogramm zum Einpauken von Vokabeln. Die Vokabeln können eingegeben, auf Disk gespeichert und wieder geladen werden. Danach wird die Abfrage mit Erfolgsquotenanzeige gestartet.

KATALOG4 A 0276 D 03 /BAS

Programmkatalog aus CHIP 5/83 S.129, geaendert fuer TANDY Model4 von Klaus Hermann!

HUSTLE G 0277 A 03 /BAS

Ein recht nettes Geschicklichkeitsspiel ähnlich dem bekannten WURMI.

Panzerschlacht G 0278 A 02 /BAS

Spiel für zwei Personen. Jeder hat einen Panzer in einem Labyrinth und muß den Gegner ausschalten.

Mini-Rechner M 0279 A 01 /BAS

Mini-Taschenrechner (+,-,*,/) als BASIC-Einzeiler programmiert. Quelle : Computer Persönlich

---

VAMPIR G 0280 A 02 /BAS

Nettes Geschicklichkeitsspiel, bei dem es darum geht, sich als Vampir seine blutige Nahrung zu suchen.

Kurvendiskussion M 0281 A 02 /BAS

Programm zur Berechnung von Nullstellen, Wendepunkten, Minima und Maxima usw. einer Funktion. Keine grafische Darstellung.

NIMM D 0282 A 03 /BAS

Das bekannte Streichhölzchenspiel in einer sehr interessanten Version unseres Clubmitgliedes Walter Schäfer.

Mastermind D 0283 A 03 /BAS

Noch ein bekanntes Spiel gut aufgemacht von unserem Mitglied Walter Schäfer!

Lagerverwaltung A 0284 A 02 /BAS

Lagerverwaltung mit allen Funktionen, die man zur Verwaltung eines kleinen Lagers benötigt. Speicherung der Daten auf Kassette.

HRG-Demoprogramm S 0285 B 01 /BAS

Die beiden Demoprogramme zum Thema 'HRG vom BASIC aus' aus dem Info 10 Seite 31/32. Autor: Jens Neueder

---

Finanzen M 0286 A 03 /BAS

Ein bischen Finanzmathematik in ein recht ansprechendes Programm gepackt!

Magic Square M 0287 A 01 /BAS

Programm zur Erstellung der geheimnisvollen 'magischen Quadrate'.

TAPE-IDENT S 0288 B 03 /CMD

Identifiziert Tapefiles und gibt Art, Namen und ev. Anfangs-, End- und Startadresse an.

SCRABBLE D 0289 A 02 /BAS

Ein Wortspiel für 1 bis 4 Spieler. Anleitung in deutsch im Spiel enthalten!

THREEKEY S 0290 B 02 /CMD

Dieses Programm ermöglicht es, verschiedene DOS- und BASIC-Funktionen mit einer Kombination aus SHIFT+Pfeil unten+entspr. Buchstaben aufzurufen.

ORBITENG D 0291 A 04 /BAS

Das Programm aus Micro 80 5/82 ermoeglicht es, die Landung auf verschiedenen Planeten (u.a. Mond, Mars, Titan) "lebensecht" zu simulieren. Schoene Grafik! eingeg. von Gerald Schroeder!

Sortierrogramm fuer Model 4 U 0292 D 03 /BAS

Sortierroutine (in Maschinenspr.) fuer BASIC-Felder, hier als BASIC-Lader! Fuer Model 4 aus 80MICRO Maerz 85,S.70; eingegeben von Klaus Herman

---

GRAFIK-EDITOR S 0293 A 50 /CMD

Ein in BASIC geschriebener und dann compilierter Grafikeditor mit vielen Moeglichkeiten (u.a. Speichern, Laden, Umwandeln in BASIC-Zeilen, verzerren des Bildes uvam.). Autor: Gerald Schroeder!

Set Funktion Keys U 0294 D 03 /CMD

Dieses Programm ermoeglicht es, die Funktionstasten des Model 4 mit beliebigen Werten zu belegen. Quelle: Micro80 11-12/84, eingegeben von Klaus Herman

---

Menue-Helfer U 0295 D 03 /BAS

Das Programm erstellt Hilfsmenues, die durch CTRL-A in den ersten sechs Zeilen des Bildschirms angezeigt werden. Achtung: nur fuer Model 4! Quelle: Micro80 7-10/85, eingegeben von Klaus Herman

---

HELP-File-Generator U 0296 D 03 /BAS

Erstellt HELP-Files (aehnl. DOS/HELP), die mit HELP/CMD aufgerufen werden. Nach Eingabe eines Stichwortes erfolgt dann die Erklaerung! Quelle: Micro80 03/85, eingegeben von Klaus Herman

---

RISIKO D 0297 A 05 /BAS

Dieses Programm wurde dem bekannten Brettspiel nachempfunden. Zur Zeit dient der Computer nur als Spielfeldverwalter, es soll jedoch eine Computergegner-Version folgen. Gute Grafiken! Autor: G. Schroeder!

TAUSCH U 0298 B 03 /CMD

Dieses Programm ermoeglicht das Arbeiten unter NewDOS (und seinen Abkoemmlingen) mit zwei Bildschirmen! Umschaltung mit SHIFT+BREAK. Quelle: GenieData Autor: Thomas Weinstein

---

* * * *
P R O S I T N E U J A H R !
* * * *

Ich grüße den CLUB 80 !

Als neues Mitglied ist es meine Pflicht mich vorzustellen. Ich bin zwar ein neues Mitglied - aber nicht mehr ganz "neu". Meine Mitgliedschaft ist ca. 2 Wochen alt (s.Datum dieses INFO, falls es eins besitzt, wie ich hoffe...)
Ich bin also eins der "jüngsten" Mitglieder in Eurer Runde - und doch wiederum nicht - im Gegenteil! Das nennt man ein "Paradoxon"! So paradox wie eine Primzahl?
Jetzt dürft Ihr raten!
Raten? Wozu hat man einen Computer? -
Dann "computet" (rechnet) mal schön!
Ausgangsbasis ist also eine Primzahl. Eine Primzahl ist, wie Ihr wißt, unteilbar. Like me. Nur durch "1" und sich selbst. Die erste Primzahl ist die "1", die zweite die "2", die dritte die "3", die vierte die - nein, eben nicht die "4" - jetzt wird's interessant! - sondern die "5", und nun paßt auf:
die 5. Primzahl ist die " 7",
" 7. " " " "13" und
" 13. " " " "37" !
Und die ist besonders interessant!
Das Interessante an ihr ist, daß die "kleinste Quersumme" ihres Produktes mit einer beliebigen Zahl immer wieder gleich der "kleinsten Quersumme" dieser Zahl selbst ist. Programmiert das mal! (Das Programm findet Ihr irgendwo im Heft)
Nun kommt's:
In ihrem Quadrat steckt mein Geburtsjahr - wieder als Quersumme natürlich und somit unsichtbar.
Multipliziert man sie mit meinem Geburtsmonat, so erhält man wieder meinen Geburtsmonat als Quersumme.
(Die Quersumme ihrer dritten Potenz ergibt übrigens wieder mein Geburtsjahr - das nur nebenbei...)
Nachdem Ihr auf diese Weise sowohl Jahr als auch Monat jenes denkwürdigen Datums errechnet habt, an dem ich das "Licht der Welt" leider nicht erblickte (denn es war Nacht), erhaltet Ihr den Rest leicht, wenn ich Euch noch verrate, daß diese Primzahl auch die Summe meiner Geburtsdaten, also gleich TT+MM+JJ ist * ganz einfach nach Herrn Riese, genannt **A.R.** (bitte <u>nicht</u> mit **J.R.** verwechseln, wie anderwärts schon geschehen...)
Habe ich mich damit genug vorgestellt? * Prima!
Nein, Irrtum: es liegt kein astrologischer Unterbau zugrunde (obwohl ich meine Skorpion-Pflichten hinreichend erfülle); es ist reine Zahlenmystik des Mittelalters (noch bewahrt von S) wenngleich ich auch dieses schon hinter mir weiß (wie Ihr nun wißt).
Sonst noch was? Ach so - was Ihr von meiner Mitgliedschaft habt? Was kann der Club von mir profitieren? * Laßt mal sehen: Ihr könnt mich <u>alles</u> fragen, egal was * ich habe nichts dagegen! * Nein, wieder Irrtum: nicht, daß ich alles wüßte (im Gegenteil!), vielmehr: ich wüßte (auch) gern alles! Wir machen das vielmehr so: Ihr fragt mich was (tel. oder briefl.) und ich schaue in der klugen Datei "PROGRAMME" nach. Diese habe ich nämlich seit 1.1.86 von Hartmut übernommen. Vorausgesetzt sie enthält die Antwort und Ihr habt mir so eine kleine dumme Zielscheibe zugeschickt, schieße ich Euch die Antwort drauf und mach sie schlau * allerdings darf sie nicht mehr als 41 Ringe haben, ab 42. treffe ich nichts mehr, und den 80.Innenring schon gar nicht! Und außerdem muß mir der hoffnungsvolle Einsender mitteilen, mit welchem Gewehr ich schießen soll: mit dem Karabiner oder der Schrotflinte; letztere hat zwei(!) Läufe und schießt d* = dank dessen doppelt dicht! (Einen Drilling besitze ich leider noch nicht; dafür aber drei Laufzwerge, häufig "Drraiws" genannt * hinten mit weichem "s".)

Ich kann übrigens auch um die Ecke schießen. Wer's nicht glaubt, dem durchlöchre ich die Scheibe mal von hinten, das heißt, vorn und hinten abwechselnd. Das nennt sich dann "DS" (Drehnicht-Scheibe, weil man sie nicht umzudrehn braucht, wenn sie vorne voll ist).
Ihr dürft natürlich auch schießen – wollt' sagen: schicken, was Ihr der Mit- und Nachwelt würdig befunden. Das staple ich in die Diskothek, auf daß sie wachse: MIRO = Much In * Random Out.

Trotz ID/CMD:

Vorabaufklärung über **PD**'s wird dankbar registriert!!!

Denn nicht immer klappen diese Tricks. Ich habe schon erlebt, daß mein TRS80 keine GENIE-Wurst verdaute, auch wenn sie in besagte Scheiben geschnitten war...
Und Ihr wollt doch alle, daß das mit dem Neuling klappt, oder?

Das walte Gott!
So spricht:
KaJott

PS für Anfänger und solche, die es werden wollen:
Meine Laufwerke sind nicht groß-, sondern nur 40-spurig;
eins – das System-Laufwerk – ist einseitig, die beiden andern hingegen sind vielseitig;
(leider: mehr als zwei Seiten gibt's glaube ich noch nicht)
und alle drei sind besonders dicht, nämlich doppelt (jedenfalls wurde das von meinem Lieferanten behauptet. Ich habe es noch nicht geprüft, weil man keine elektrisch aufgeladenen Geräte mit in die Badewanne nehmen soll...)

* * * ALLES KLAR?? * * *

Achtung :
Hier die PDRIVE-Werte unseres neuen Club-Programm-Bibliothekars für selbstbootenden Disketten :
40 TRACK -- SS -- DD
Ansonsten ist auch
40 TRACK -- DS -- DD
möglich.

Impressum

Vorstand

Kontaktadresse für Clubangelegenheiten, Clubbücherei /Fundgrube, Clubkasse:
Günther WAGNER
Gartenstraße 4
8201 Neubeuern
Tel.: 08035 /3361
< 18 - 20 Uhr >

Programmbibliothek

Kontaktadresse:
Klaus-Jürgen Mühlenbein
Am Mönchgarten 28
6940 Weinheim
Tel.: 06201 /55052

Redaktion

Kontaktadresse:
Jens NEUEDER
Panoramastraße 21
7170 Michelbach /Bilz
Tel.: 0791 /42877
tagsüber 0791 /44-667

Adventure-Ecke

Kontaktadresse:
Alexander WAGNER
Theresienstr. 2ic
8224 Chieming
Tel.: 08664 /1500

Hardware

Kontaktadresse:
Walter ZWICKEL
Lengfelden 123
5101 Bergheim (Austria)
Tel.: 0043662 /51130

Redakteure dieser Ausgabe

Manfred	Held	* Klaus-J.	Mühlenbein
Jens	Neueder	* Hartmut	Obermann
Jutta	Obermann	* Walter	Piller
Bernd	Retzlaff	* Gerald	Schröder
Peter	Stevens	* Günther	Wagner
Walter	Zwickel	*	

sowie Artikel aus: Dr. Dobb's Jurnal, c't und Computerheft

Bankverbindung des CLUB 80
Sparkasse Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postscheckkonto der Sparkasse
Nr. 8077-801

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

auch in dem neuen Jahr findet Ihr den Schluß in altbewährter Weise -wie auch im letzten Jahr- wieder fast ganz am Ende des CLUBINFO's.

Zuerst aber noch einen herzlichen Dank für die Weihnachts- und Neujahrswünsche die ich von Euch erhalten habe. Ich habe mich sehr darüber gefreut. Ich hoffe, daß das Neue Jahr auch Eure Wünsche erfüllt und Ihr den Jahreswechsel gut überstanden habt. Um dies mit den Worten von Ulrich zu beschließen:

"Habt immer eine arbeitende CPU"

..

Am Infoende findet Ihr diesmal einen einfachen Kalender, der Euch an die Redaktionsschlußtermine in diesem Jahr und an das Clubtreffen erinnern soll. Als Weiteres habe ich einmal eine Zusammenfassung geschrieben, die Euch als INFO-Mitarbeiter den "Umgang" mit der Redaktion erleichtern bzw. vereinfachen soll. Diese beiden Blätter sind zum Abtrennen gedacht. Somit habt Ihr ein paar Unterlagen mehr an Eurem "Computerarbeitsplatz" zur Verfügung. Vielleicht sind sie Euch nützlich.

Ich persöhnlich hatte diesmal wenig Zeit etwas Eigenes zu schreiben. Ich hoffe, daß ich bis zum nächsten Mal wieder einen Beitrag für Euch habe.

So, dies war es mal wieder. Nun habt Ihr ein weiteres INFO in Eurer Sammlung. Ich wünsch Euch viel Spaß beim lesen und melde mich pünktlich mit dem nächsten INFO wieder.

Euer

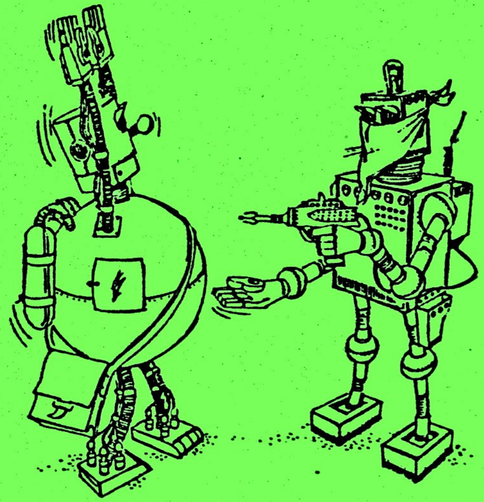
J. Neueder

CLUB 80 Mitgliederadressenliste

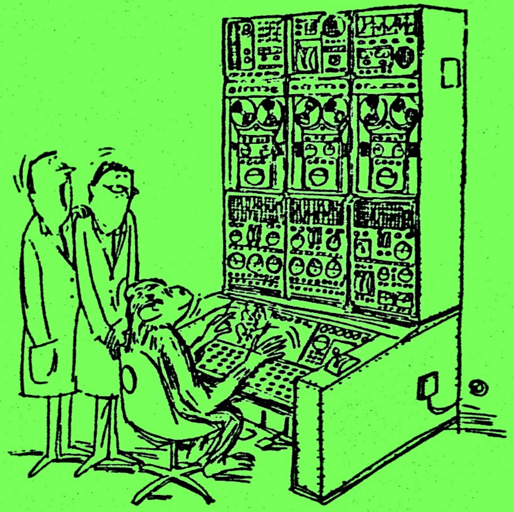
Name	Vorname	Straße	PlZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Albers	Herbert	Zum Duwelshopen 14	2117	Westedt	
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Drowälder	Bernd	Buchentalsweg 8	4939	Steinheim	05233 /4320
Fröhlich	Burghard	Brandströstraße 11	4350	Recklinghausen	02361 /63416
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lü.	06201 /55052
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Wegen Systemwechsel /Computerei aufgegeben ausgetreten
Hans Baldes (zum 1.1.86)
Matthias Hallup (zum 1.1.86)
Andreas Marx (zum 1.1.86)
Jean-Claude Wies (zum 1.1.86)

Neuzugänge :
Herbert Albers
Klaus-Jürgen Mühlenbein
Hans Raggan
Oliver Volz



»GELD HER ODER BATTERIE!«



»Menschliches Versagen ist bei uns jetzt völlig ausgeschlossen.«

CLUB80 Termin - Kalender 1986

Januar

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Februar

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
◆	25	26	27	28		

März

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

April

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	‡	‡	‡
14	15	16	17	18	19	20
21	22	23	24	25	26	27
◆	29	30				

Mai

Mo	Di	Mi	Do	Fr	Sa	So
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Juni

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
◆						

Juli

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

August

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
◆	26	27	28	29	30	31

September

Mo	Di	Mi	Do	Fr	Sa	Sc
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Oktober

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
◆	28	29	30	31		

November

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

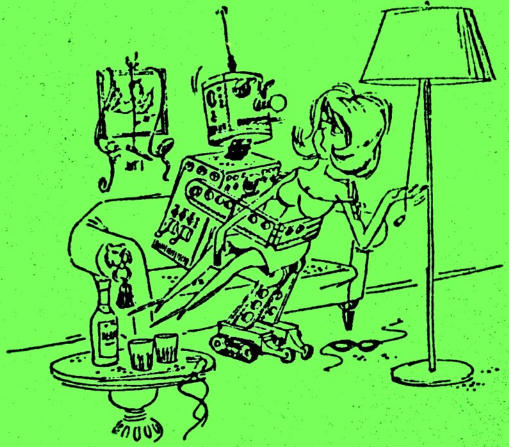
Dezember

Mo	Di	Mi	Do	Fr	Sa	So
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
◆	30	31				

◆ —> Redaktionsschluß des Info's

‡ —> 2. Clubtreffen in Holzhausen (siehe INFO 10)

Bitte denkt an die Termine und nehmt sie rechtzeitig war !!!



»ICH HABE IHNEN DOCH GLEICH GESAGT, DASS ICH ECHT UND NICHT MASKIERT BIN!«

»Wer hatte eigentlich die famose Idee, den Computer aus Indien zu importieren?«

CLUB80 Termin - Kalender 1986

Januar

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Februar

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
◆	25	26	27	28		

März

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

April

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	‡	‡	‡
14	15	16	17	18	19	20
21	22	23	24	25	26	27
◆	29	30				

Mai

Mo	Di	Mi	Do	Fr	Sa	So
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Juni

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
◆						

Juli

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

August

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
◆	26	27	28	29	30	31

September

Mo	Di	Mi	Do	Fr	Sa	So
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Oktober

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
◆	28	29	30	31		

November

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Dezember

Mo	Di	Mi	Do	Fr	Sa	So
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
◆	30	31				

◆ --> Redaktionsschluß des INFO's

‡ --> 2. Clubtreffen in Holzhausen (siehe INFO 10)

Bitte denkt an die Termine und nehmt sie rechtzeitig war !!!

I N F O F O R M

An alle Redakteure --- oder an die, die es noch werden wollen.

Aus gegebenem Anlaß möchte ich Euch einige Tips geben, die eine bessere übersicht und Form der INFO ermöglicht. Gleichzeitig hilft es mir, die Redaktionsarbeit zu vereinfachen. Ich bitte Euch daher, sich an diese Richtlinie für Redakteure zu halten.

-- DANKE --

1. ALLGEMEINES :

Durch das Verkleinern der Artikel bekommen wir viel Information auf kleinsten Raum. Dieses System wollen wir beibehalten, es ermöglicht uns nämlich das INFO mit geringem Kosten- und Material-aufwand zu erstellen. Um daher eine Verkleinerung Eures Beitrages in bester Qualität zu erhalten, ist folgendes nötig :
Textbeiträge oder Listings sollten ausgedruckt oder mit Schreibmaschine geschrieben sein. Als Untergrund für die Buchstaben und sonstige Zeichen empfiehlt sich weißes linienfreies Papier. Bitte sendet keine handschriftlichen Beiträge ein, da meist die persöhnliche Handschrift eines jeden nicht mehr an die in der Schule gelernte Schönschrift heranreicht und daher zu Entzifferungsproblemen führt.
Wer keinen Drucker/Schreibmaschine hat, kann mir ja sein Textfile auf Diskette zusenden. (Bitte Textverarbeitungssystem und Titel des File angeben)
Verwendet auch nicht gerade die kleinste Schriftart Eures Druckers oder ein altes Farbband, denn auch dies führt nach dem Verkleinern und Drucken zur Unleserlichkeit.
Bei Einsendung eines mehrseitigen Artikels (mehrerer Artikel) denkt bitte daran, die Reihenfolge der Blätter numerisch festzulegen (auf der Rückseite der Beiträge genügt). Ich hoffe, dadurch Verwechslungen innerhalb eines Beitrages oder ein Vertauschen der chronologischen Reihenfolge der Beiträge entgegenzutreten.

2. FORM :

Die Ausdruckform sollte eine Breite von 15 - 17cm haben, und die maximale Zeilenzahl pro Seite darf 62 Zeilen nicht überschreiten. Ich benötige die dadurch erstandenen freien Ränder für die Seitennumerierung /-markierung sowie als Kleberand zum Binden und Abheftlochrand für Euch.

3. ARTIKEL AUS ZEITSCHRIFTEN :

Beachtet bei Artikeln aus Zeitschriften, daß Text, der in farbigen oder gerasterten Flächen steht, durch das Kopieren meist schon schwer leserlich wird. Versucht beim Kopieren die Kopie mit einer helleren Stufe als normal zu erstellen. Ist die Textstelle trotzdem noch nicht schwarz auf weiß, solltet Ihr vielleicht eine Abschrift der entsprechenden Textstelle machen.
Ihr habt aber auch die Möglichkeit, mir das Original zukommen zu lassen. Ich habe die Gelegenheit, heller zu kopieren mit gleichzeitiger Verkleinerung. Dies ergibt meist eine brauchbare Veröffentlichung für das INFO. Das Original erhaltet Ihr auf jeden Fall mit der INFO zurück.

4. VERSAND :

Nachdem Ihr nun Euer Schriftwerk fertig habt, und Ihr es der Redaktion zukommen lassen wollt, solltet Ihr Eure Arbeit nicht zusammenfalten und in ein kleines Kuvert stecken. Diese Knicke geben dann beim Kopieren/Verkleinern graue Streifen, die die Buchstaben an dieser Stelle unkenntlich machen. Darum ungeknickt in ein der Größe entsprechendes Kuvert (max DIN A4) und ab an die Post. Portomäßig sind die Kosten für Euch ja gleich.

5. BEITRÄGE ZUR BÖRSE :

Beiträge zu diesem Thema schreibt bitte in folgender Art und Weise auf:

Sparte (Wer hat was, wer will was oder Fragekasten)
Euren Text (Angebot ,Wunsch oder Frage)
Euer Autogramm

Ich schneide die Teile aus Eurem Brief und klebe sie dann in die entsprechende Sparte.

6. DISKETTEN :

Bitte gebt grundsätzlich Eure EDRIVE's an. Lesbar sind für mich nur Singel-Density-Disketten folgender Typen:
40 Track SS oder DS und 80 Track SS oder DS
Zurücksendung Eurer angekommenen Disketten ist Ehrensache.
Ihr denkt bei der Diskversendung natürlich auch an die richtige Versendungsform Eurer Scheiben. Am besten sind Kuverts mit Luftpolsterung --- daß wißt Ihr ja selbst.

Wenn ihr Euch an diese kleine Redakteuranleitung haltet, dürfte das die Qualität unseres INFO's noch steigern. Ich hoffe, Ihr helft dabei mit.

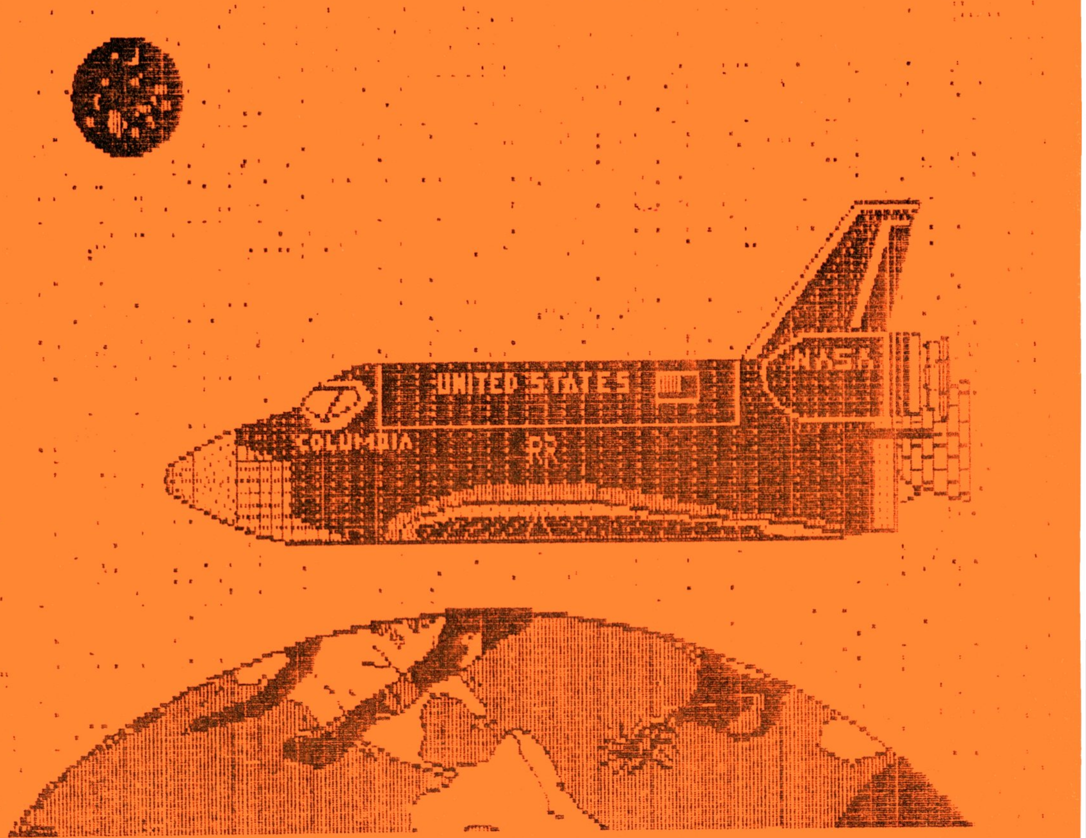
Im voraus nochmals meinen DANK
Redaktion CLUB 80

CLUB 80

Clubinfo

der

TANDY -
GENIE -
und
KOMTEK -
ANWENDER



12. AUSGABE

Kontaktadresse : CLUB 80 / Günther WAGNER / Gartenstraße 4 / 8201 Neubeuern

Tel.: 08035/3361 (18 - 20 Uhr)

Inhaltsverzeichnis

Seite:

Clubinternes

Der Vorstand informiert 01
Clubtreffen in Holzhausen . . . . . . .02 - 04

Software

Nachtrag zu INPUT 05
Programmierwettbewerb . . . . 07 - 09
Verbessert: IO-Port . . . . 11 - 12
JKL - druckerschonend . . . . 13 - 14
Volle Library ? . . . . 15 - 16
DOE - Genie IIIs . . . . 17 - 20
HRG - Genie IIIs . . . . 21 - 26
Zeichensatz neu - Genie IIIs . . . . 28 - 32
Call mit BASCOM . . . . 33 - 35
Modell 4 intern . . . . 37 - 43
Modell 4 soft . . . . 45 - 59

Hardware

CLUB 80 Hard 61
Datenübertragung per Fernsehbild . . 62
80-Zeichenkarte . . . . 63 - 64
Speed Writing . . . . 65 - 68
Modell 4 hard . . . . 69 - 78
Bildschirmausgabe aufpolieren . . . 79 - 84
ECB-Bus 91

Seite:

Börse

Wer hat was ???--Wer sucht was ??? 85 - 90
Fragen, Fragen, Fragekasten 92

Sonstiges

In Kürze 60
Abmahnfalle 94
Diskussion . . . . .95 - 96

Programmbibliothek

Neue Programmbibliothekenliste liegt diesem INFO bei

Club -Bücherei

Bücherliste CLUB 80 . . . . .97 -101

Die letzten Seiten

Impressum 103
Schluß 104
Clubmitgliederadressen . . . . am INFO-Ende
Persönliches Kärtchen . . . . liegt dem INFO bei

Liebe Mitglieder,

zum letzten Mal an dieser Stelle als Vorstand des Clubs möchte ich zunächst allen Mitgliedern recht herzlich danken, die mich in meiner Arbeit unterstützt haben.
Mein besonderer Dank geht an den Jens, dem Hartmut und dem KaJott. Sie haben sich bereit erklärt, verantwortlich ein Stück des Clubs zu übernehmen und Ihre Freizeit dafür zu opfern.

Wir brauchen nun jemanden (oder auch mehrere), der bereit ist, für das Amt des Vorstandes zu kandidieren. Die Satzung sagt ganz klar aus, daß der Vorstand von der Gesamtheit der beim Clubtreffen anwesenden Mitglieder gewählt wird. Ich hoffe, daß sich bis zum Clubtreffen Kandidaten finden, denn eines ist gewiß: Ich lasse mich als Vorstand nicht mehr aufstellen.

Ich will nun einen kurzen Abriß der Tätigkeiten des Vorstandes geben. Der Vorstand ist derjenige, welcher den Club nach außen hin vertritt. Das heißt, er hat Anfragen von Interessenten zu beantworten. Die neuen Mitglieder hat er in den Club aufzunehmen. Er ist für die Kassenführung der Hauptkasse verantwortlich. Der Vorstand ist Anlaufstelle für viele Clubangelegenheiten. Er soll auch den Kontakt zu den Mitgliedern halten. Der Vorstand regelt den Bücherverleih.

Das wär dann schon alles, was mir derzeit einfällt. Ich habe in den letzten Monaten im Schnitt wöchentlich 3-5 Stunden für den Club aufgebracht. Wer jetzt noch Fragen zum Vorstand hat, kann sich ja bei mir melden.

Bitte nehmt alle das Problem Vorstand ernst - wir brauchen einen Vorstand für einen funktionsfähigen CLUB 80 - und an dem ist uns doch allen gelegen. Sollte unter uns ein Mitglied sein, das bereit wäre, den Vorstand zu übernehmen, aber aus beruflichen oder privaten Gründen am Clubtreffen nicht teilnehmen kann, so muß er mir dies unbedingt mitteilen. Ich hoffe ferner, daß sich die Teilnehmer am Clubtreffen auch Ihre Gedanken darüber machen, wie das Treffen ablaufen soll und was besprochen gehört.

Wie bereits im letzten Info angedeutet wurde, haben wir einen Teil des Überschusses von 1985 für den Ankauf von insgesamt 22 Büchern verwendet. Hierbei konnten wir 19 Bücher zum m.E. sehr günstigen Komplettpreis von 171,-- DM (inklusive aller angefallenen Unkosten) kaufen. Die komplette Bücherliste des CLUB 80 ist in diesem Info abgedruckt.

Ich hoffe jetzt nur noch auf ein gelungenes Clubtreffen.

Viele Grüße
Euer Günther Wagner

P.S.: Zum Clubtreffen findet Ihr weiter hinten noch einen Beitrag.

Clubtreffen in Holzhausen a. d. Heide

Um allen Teilnehmern am Clubtreffen die Anfahrt möglichst zu erleichtern, veröffentlichen wir hier noch einmal eine Straßenkarte der weiteren Umgebung von Holzhausen a. d. Haide. Die günstigsten Anfahrtswege sind

- von Süden kommend:
Richtung Wiesbaden, dort auf die A66 in Richtung Rüdesheim. Die Autobahn geht direkt in die B260 über, welche direkt durch Holzhausen führt!

- von Norden kommend:
entweder linksrheinisch in Richtung Koblenz, dort Richtung Bad Ems/Nassau (B417). In Nassau auf die B260 in Richtung Wiesbaden oder rechtsrheinisch Autobahn A3 bis Limburg a. d. Lahn. Dort auf die B417 in Richtung Diez a. d. L./ Nassau. In Nassau auf die B260 in Richtung Wiesbaden.

Die B260 führt, wie schon erwähnt, direkt durch Holzhausen hindurch. In Holzhausen (übrigens der Geburtsort von Nikolaus August Otto) ist der Weg zum Hotel "Taunusblick" sehr gut ausgeschildert. Für die von Süden kommenden Teilnehmer der Hinweis: Die Hinweisschilder (mehrere) "Taunusblick" befinden sich auf der rechten Straßenseite! Für alle aus nördlicher Richtung ankommenden Teilnehmer gilt demnach: Die Hinweisschilder sind links zu suchen und zwar erst nach dem Passieren des Geburtshauses (deutlich sichtbar) von N.A. Otto!

Ich glaube mit dieser Beschreibung dürfte jeder Teilnehmer den Ort des Treffens mit Leichtigkeit finden!

Nun zu den Teilnehmern, die ev. mit der Bahn anreisen wollen. Da Holzhausen keine Bahnanschluß hat, ist es ohne eigenes KFZ nur schwer zu erreichen. Um trotzdem auch Mitgliedern ohne eigenen oder geliehenen fahrbaren Untersatz die Teilnahme zu ermöglichen, erbiete ich mit, sie vom Bahnhof Mainz oder Wiesbaden abzuholen und nach Beendigung des Treffens dort auch wieder abzuliefern. Damit ich nicht dauernd nur auf Achse bin, wäre mir natürlich ein "Sammeltransport" am liebsten. Ich bitte also alle Bahnfahrer, sich sobald wie möglich, jedoch bis spätenstens 31.03. telefonisch bei mir zu melden, damit wir die Ankunftszeiten möglichst aufeinander abstimmen können!

Noch ein letzter Hinweis:
Mitglieder, die schon abgesagt oder ihren Anmeldezettel nicht abgeschickt haben und sich kurzfristig doch für die Teilnahme entscheiden, sollten mich telefonisch davon in Kenntnis setzen. Sollte nämlich der "Taunusblick" voll belegt sein, kann ich ohne vorherige telefonische Anmeldung keine Garantie dafür übernehmen, daß ihr in einem anderen Gasthof noch ein Zimmer bekommt!

Doch damit genug der guten Ratschläge. Ich wünsche allen Teilnehmern am Treffen eine gute und vor allem unfallfreie Anfahrt, Euer

Hartmut Obermann

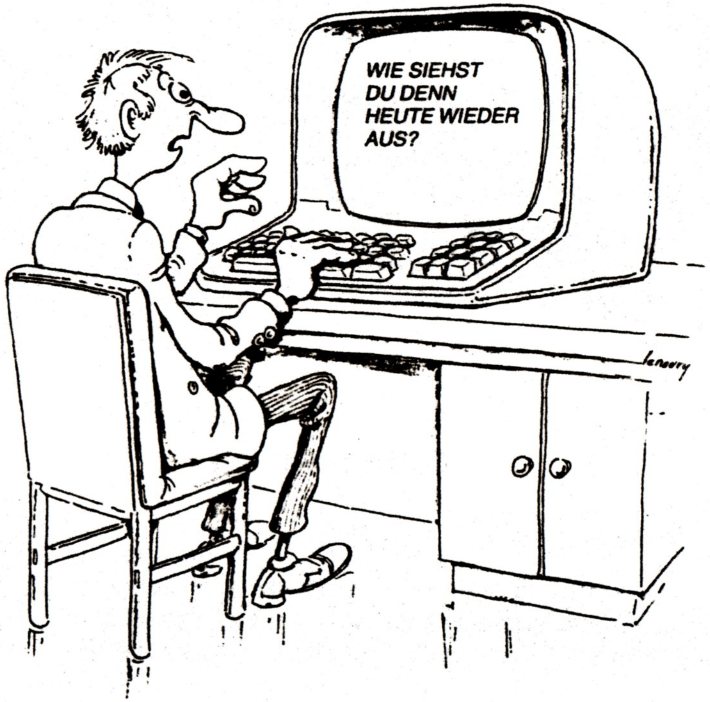
Hotelanschrift: "Taunusblick", 5429 Holzhausen ☎ 06772/8343

0 3
0 4
NEUWIED
KOBLENZ
WETZLAR
GIESSEN
BAD HOMBURG
OBERURSEL
FRANKFURT AM MAIN
WIESBADEN
OFFENBACH AM MAIN
NEU-ISENBURG
Limburg an der Lahn

Nachtrag zu "INPUT"

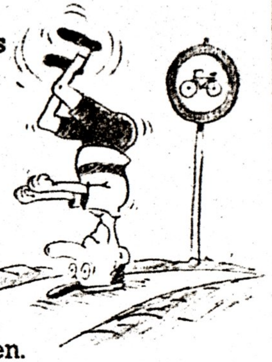
Ihr erinnert Euch sicherlich noch an meinen wenig hilfreichen Hinweis, den Bildschirmeditor zu laden, damit beim INPUT die nächste Zeile nicht gelöscht wird. Nun habe ich den Weg des geringsten Widerstandes gesucht und wiederum in "Basic Faster and Better" auch gefunden. Das nachstehende Programm löst das Problem der verschwindenden Zeile selbst, indem es sie einfach über eine Assembler-Routine in einen String rettet und nach der Eingabe (und dem Löschen) wieder zurückschreibt. Nicht genial, aber einfach.

```
10 CLEAR 5000 : DEFINT A-Z
20 DIM US(4)   'Assembler-Routine 1: Eingabe
30 DIM UV(6)   'Assembler-Routine 2: Zeile retten
40 US(0)=1030 : US(1)=8448 : US(3)=-9779 : US(4)=-14075
50 UV(0)=8448 : UV(2)=4352 : UV(4)=16385 : UV(5)=-4864 :
   UV(6)=-13904
60 ' irgendein Hauptprogramm
70 ' ...
80 ?$I1,;   'I1 = Position, an der die Eingabe erfolgt, könnte
90 '              auch über Abfrage von (4020H) und (4021H)
100 '             realisiert werden
110 GOSUB 1000
120 ' Reaktionen auf die Eingabe, entweder A$ oder I
130 ' auswerten
140 ' ...
140 END
1000 ' Unterprogramm "Eingabe"
1010 ' -das Programm benötigt die Cursorposition in I1
1020 ' -es gibt die Eingabe in A$ und den ASCII-Wert des
1030 '  ersten Buchstaben in I aus
1040 I2=(INT(I1/64)+1)*64  '$-Position am Anfang der nächsten
1050 '                          Zeile
1060 '   Retten der nächsten Zeile:
1070 F$=STRING$(65,32) : UV(1)=15360+I2
1080 UV(3)=CVI(CHR$(PEEK(VARPTR(F$)+1))+CHR$(PEEK(
     VARPTR(F$)+2)))
1090 I=0 : DEFUSR=VARPTR(UV(0)) : I=USR(0)
1100 '   Eingabe:
1110 A$=STRING$(5,32)
1120 US(2)=CVI(CHR$(PEEK(VARPTR(A$)+1))+CHR$(PEEK(
     VARPTR(A$)+2)))
1130 I=0 : DEFUSR=VARPTR(US(0)) : I=USR(0)
1140 '   Gelöschte Zeile wieder zurück:
1150 UV(1)=CVI(CHR$(PEEK(VARPTR(F$)+1))+CHR$(PEEK(
     VARPTR(F$)+2)))
1160 UV(3)=15360+I2 : I=0 : DEFUSR=VARPTR(UV(0)) : I=USR(0)
1170 '   Auswertung
1180 IF INSTR(A$,CHR$(13))>0 THEN A$=LEFT$(A$,
     INSTR(A$,CHR$(13))-1) 'String ohne CR
1190 I=ASC(A$+CHR$(128))
1200 IF PEEK(14400)=4 THEN END  ' Break-Taste gedrückt
1210 IF I<32 OR I>127 THEN I=0  ' Code zu klein oder zu groß
1220 RETURN
```

Französisch: **Il a un petit vélo dans la tête.**

Er hat ein kleines Fahrrad im Kopf.
Er ist als Kind zu heiß gebadet worden.



Un chien regarde bien un évêque.
Ein Hund kann doch einen Bischof anschauen.
Ansehen kostet nichts.

Griechisch: **Opou laloun polli kokori argi na ximerosi.**

Wo viele Hähne krähen, verspätet sich das Tagwerden.
Viele Köche verderben den Brei.

Eipo (Papua-Neuguinea): **Asok kikye binil.**

Mir ist nach Kratzen an der Stirn.
Ich habe mich wohl getäuscht.

* * * Wozu Zehnzeiler ? * * *

Günther hat mit seinem Wettbewerb-Vorschlag im letzten INFO sicher alle müden Hacker anregen wollen, auch mal etwas zum INFO beizutragen; und seien es "nur 10 Zeilen"!

Im ersten Augenblick war ich ob dieser neuen Philosophie etwas verwundert und fragte mich, worin dabei die "Leistung" bestehen könnte! Denn seit ich die notdürftige Beleuchtung dieser UN-Welt wahrnahm (um nicht wiederholt zu behaupten, daß ich "das Licht der Welt erblickte" - denn wo ist es denn noch außerhalb des 'CLUB80'??...) habe ich den Leitspruch beherzigt:

In der Kürze liegt die Würze !

Also war ich der Meinung, dieser Wettbewerb sei kein Wettbewerb um eine edle Sache...

Hier könnt Ihr nun die "Entwicklung eines unterentwickelten Entwicklungsfähigen" miterleben:

Bestärkt durch Harold B.Fink, einen klugen Mann - hat er doch das Schachspiel "SFINKS" erfunden, eines der besten! -, der im August 1984 in der **80 MICRO** (von manchen fälschlicherweise "MICRO 80" genannt) eine ganze Serie kluger Einzeiler unter dem Titel "ONE-LINERS" publizierte und dies als besonders vornehme (und elegante?) Programmierweise deklarierte - - bestärkt also durch solches Vorbild, machte ich mich daran, ebenfalls einen Einzeiler zu verfassen, der das gleiche leistet wie ein X-Zeiler (z.B. X=10), egal wofür.

Was sollte als Beispiel herhalten? Wieder einmal das leidige Umwandlungsthema "DEZ/HEX"! Hatte ich doch gerade in INFO 3, vorletzte Seite (Seiten-Nummern gab es da noch nicht; welches Kind zählt schon die Tage ersten Spiels??) die Schöpfung von Kamerad Zwickel (Hallo Walter, ich grüße Dich und Deine Berge im schönen österreich, besonders Dein Bergheim!) gelesen, der über 75 Stationen von DEZ nach HEX fährt, wobei die Entfernung von einer zur nächsten zugegeben auch zu Fuß zu überbrücken ist, so kurz sind deren Abstände (=Zeilen)...

"Ha!" - so frohlockte ich: "Was der Walter in 75 Schritten macht, das machst Du doch spielend in einem Durchgang!!"

Gesagt - getan. Da habt Ihr ihn, den EINZEILER! Seht selbst: er tut's!

Gewonnen? * Langsam! * Urteilen heißt Vergleichen. Ich hab da meinen eigenen Maßstab. Im Nachbarclub (wer kennt den **BCC** nicht?) hatte ich schon vor Jahresfrist ein Programm dem öffentlichen Lächeln preisgegeben, das ich bereits vor Urzeiten für's gleiche "Problem" geschaffen: eine 'DEZ/HEX/DEZ-Umwandlung'. Hier sei es mit seinen 10 Zeilen nochmals wiedergegeben. (Anm.: Auf die DEZ/HEX-Umwandlung selbst entfallen davon nur 5 Zeilen.)

Nun kommt's: Fahrt mal mit beiden Programmen die gleiche Strecke ab! Sagen wir irgendwas, z.B. von 47111(dez) nach B807(hex)! Toll, was? Der "Normalzeiler" ist ein TEE, der "Einzeiler" ein Bummelzug dagegen!

Zum erstenmal hatte der Computer für mich etwas Menschliches an sich:

Je mehr man ihn quält, desto mehr er sich wehrt!
So hat er mich leise Bescheidung gelehrt...

Also steckt im "ZEHNZEILER" wohl doch so etwas wie ein "weises Optimum"? Dann wohlauf: folgt Günthers Aufruf!

* * Ehre wem Ehre gebührt!! * *

Der Naturphilosoph fühlt sich auch bestätigt: Mutationen = winzigste Veränderungen können unübersehbare Folgen haben! Nimm einen EINZELLER (der bekanntlich ewig lebt) das kleine Häkchen am ersten L, auf das er so stolz ist: und er lebt als EINZEILER kaum eine Stunde noch...

Und nun werden ihm obendrein auch noch die ZEHNE gezeigt!!

* O jeminee - das tut ja weh! * * Grüß Gott! * WaJott *

(Schlußfrage: Wann spielen wir "ALLE NEUNE"?)

W E T T B E W E R B

Zu der Wettbewerbsausschreibung möchte ich ergänzenderweise noch folgendes anmerken :

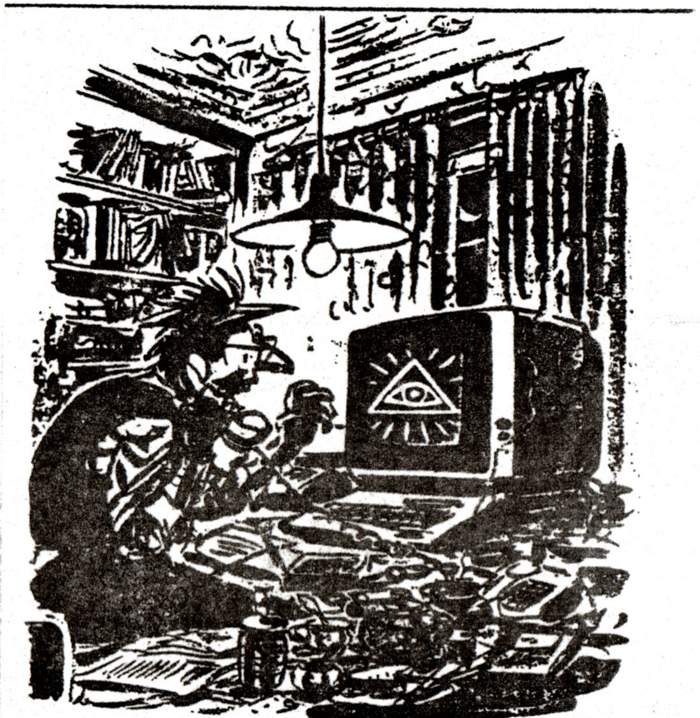
Es muß kein **ZEHNZEILER** sein!
Eine geringere Zeilenzahl wird auf jeden Fall auch akzeptiert. Nur wollte ich durch die Beschreibung **ZEHNZEILER** sicherstellen, daß die Programme nicht mehr als 10 Zeilen haben.

Die Begründung für die Programmlänge liegt darin, daß die Programme leicht überschaubar sein sollen,und von jedermann leicht eingetippt werden können. Auch wollte ich durch die Programmkürze den Programmieraufwand für Euch gering halten und für uns, die das Programm beurteilen sollen, eine Möglichkeit schaffen, genauer über den Wert eines Programmes zu urteilen. Deshalb **MAXIMUM** :
ZEHNZEILER

Ich möchte Euch nun nochmals bitten, es diesem mutigen Vorreiter, dessen Programm hier folgt, nachzutun und eine Programmschöpfung -es können auch mehrere sein- aus eigener Feder hier in's Rennen zu schicken.

Eure CLUB 80 Redaktion

```
1 REM * * Beispiel eines 'E I N Z E I L E R s': DEZ/HEX-Umwandlung * *
2 REM * * ============================================================ * *
3 REM   aufgestellt von KaJot Mühlenbein, Weinheim, Februar 1986
4 REM
5 REM
10 CLS:INPUT"Gib eine ganze Zahl unter 65336 ein :    ";D:FORI=3TOOSTEP-1:S
(I)=INT(D/16ÄI):S$(I)=CHR$(48+S(I)-(S(I)>9)*7):D=D-S(I)*16ÄI:NEXT:PRINT"Die
 entspr. Sedezimalzahl lautet :  "S$(3);S$(2);S$(1);S$(0):END
```



„Weißt du, welchen Code ich gerade geknackt habe?"

*** * * Die gleiche Aufgabe in einem 'ZEHNZEILER' : * * ***

(der Teil 'DEZ/HEX-Umwandlung' beansprucht 5 Zeilen und ist>3mal schneller!)

```
1 CLS:CLEAR70:PRINT"    UMWANDLUNG VON HEX- IN DEZ-SYSTEM UND UMGEKEHRT
          < (C) Muehlenbein, Weinheim 1983 >
   ";STRING$(47,"*"):DEFINTA,I-N:DEFSTRH,X
2 INPUT"

      WIRD    <1>  HEX/DEZ-UMWANDLUNG
      ODER    <2>  DEZ/HEX-UMWANDLUNG
      GEWUENSCHT ";K:ONKGOTO3,6
3 CMD"LC N":INPUT"

HEXADEZIMALZAHL =    ";HEX
4 FORI=1TOLEN(HEX):A=ASC(MID$(HEX,I,1))-48:IFA>16THEN
A=A-7
5 DEZ=DEZ*16+A:NEXT:PRINT:PRINTCHR$(214)HEX" (HEX)  =  "DEZ" (DEZ)":CMD"lc"
:END
6 INPUT"

DEZIMALZAHL     =    ";DEZ:D(I)=DEZ
7 D(I+1)=INT(D(I)/16):B(I)=D(I)-D(I+1)*16:IFD(I+1)>0THENI=I+1:GOTO7
8 FORJ=0TOI:IFB(J)<10THENX(J)=RIGHT$(STR$(B(J)),LEN(STR$(B(J)))-1):ELSEX(J)
=CHR$(B(J)+55)
9 NEXT:FORJ=ITOOSTEP-1:HEX=HEX+X(J):NEXT:N=4-LEN(HEX):NULL$=STRING$(N,"0")
10 PRINT"

"CHR$(212)DEZ" (DEZ)   =   "NULL$+HEX"  (HEX)":END
```

„Der menschliche Verstand ist in der Praxis nicht verläßlich, am wenigsten in größter Not."

„Früher litten wir an Verbrechen, heute an Gesetzen."

„Autovertreter verkaufen Autos, Versicherungsvertreter Versicherungen. Und Volksvertreter?"

„Wenn die Affen es dahin bringen könnten, Langeweile zu haben, so könnten sie Menschen werden."

Verbessert: IO.port#<,xx,yy,...>

Im Info Nr. 11 stellt Hartmut den (nicht mehr ganz) neuen Library-Befehl OUT vor, den ich laut Listing offenbar schon vor bald zwei Jahren verbrochen habe. Irgendwie gefiel mir die Routine beim Durchlesen nicht mehr: sie erschien mir zu brav, amateurhaft. Auf der nächsten Seite ist deshalb eine heute entstandene Alternative vorgestellt, deren reiner OUT-Teil rd. 20% kürzer und schneller ist.

Zusätzlich gibt es jetzt die Möglichkeit, einen Input von einem Port einzuholen. Dazu wird hinter dem Befehlswort einfach nur die Portadresse ohne weitere Parameter (Outputs) angegeben. Die Routine "weiß" dann, daß sie den Port nur auslesen soll. Durch diese Erweiterung wurde das Programm nun insgesamt 9 Bytes länger als die alte Version. Wegen der Input-Möglichkeit heißt der Library-Befehl nun IO (Input/Output) statt nur OUT.

Im OUT-Teil unterscheiden sich das alte und das neue Programm nicht. Erklären will ich deshalb nur die Verbesserungen, und seien sie auch nur ästhetischer Art: Der Befehl RLCA ist hier vorzuziehen, denn aus der Routine getchr kann durchaus auch mit gesetztem Carry-Flag zurückgekehrt werden. Das würde mit RLA den Akku verändern. Als Zwischenspeicher für das höherwertige Nibble der Hexzahl dient hier B, weil halt C den Port hält. Sieht besser aus. Die meisten Unterschiede gibt es bei der Umwandlung der ASCII-Zahl der Eingabe in eine Binärzahl zur Ausgabe. Hier wurden auch die meisten Bytes gespart. An der Stelle error war in der alten Version ein Bug. Es genügt nicht, mit NZ-Bedingung über RET zurückzukehren. Es muß über 4409h geschehen.

Der Dump zeigt den letzten Sektor von SYS3/SYS. Die OUT-Routine (unterstrichen) überschreibt hier die ehemalige JKL-Routine, die beim Genie IIIs überflüssig geworden ist. Wer einen anderen Computer hat, kann OUT in jedem beliebigen SYS-File unterbringen, wo Platz dafür übrig ist, z. B. in SYS28, wie Hartmut vorschlägt oder in SYS29 wie in H-DOS. Die jeweilige Ladeadresse ist dann natürlich vom verfügbaren Platz abhängig. Sie wird mit ORG variiert. Dabei ändern sich auch die Sprungadressen der CALLs. Man sollte daher vorsichtshalber die Routine zu diesem Zweck mit EDTASM o. ä. eintippen und neu assemblieren.

Eine interessante Variante ist es, die Routine als CMD-File zu benutzen, nicht als SYS-File. Um dennoch den Arbeitsspeicher sauber zu halten, wird dazu die ORG-Adresse auf den Beginn des Sektorpuffers 4200h gelegt. Für dieses kleine Programm würde sogar der DOS-Eingabepuffer ab 4318h genügen (Vorsicht, der Aufrufbefehl selbst braucht davor noch genügend Platz!). Das ist auch eine gute Methode, das Programm zunächst auf einwandfreie Funktion zu überprüfen, bevor man sich das DOS damit zerschießt.

Arnulf Sopp

```
000400: 8132 8744 E57D C605 01EA F850 6F06 08CD  2 D ü EA  PoFH
000410: 5951 7EFE 202B 0A3E 2FCD 3300 0603 CD59 YQB  (J>/ 3§FC Y
000420: 5121 6751 CD67 44CD 4900 CBAF FE45 2809 Q!gQ gD I§   E(H
000430: FE4E 2804 FE4A 20EF F5CD 3300 3E0D CD33  N(D J    3§>M 3
000440: 00F1 E1FE 45C8 FE4E 2811 1180 443E 8012 §   E  N(QD D> R
000450: CD17 4EC0 E579 CD36 49E1 C07D E6E0 C620  WN  y 6I  ü
000460: 6FD2 BC50 0C79 FE00 DAB7 50AF C97E FE20 o  PLy §  P  B
000470: 23C4 3300 10F7 C900 2020 20C2 0944 6174 # 3§P  §     IDat
000480: 6569 206C 7C73 6368 656E 203F 2028 4A2F ei löschen ? (J/
000490: 4E2F 4529 2003 CDA3 514F CDD5 4C20 11ED N/E) C  QG  L Q
0004A0: 7B2A 2040 CD68 403E 0AC3 3300 CDD5 4CC8 {* § h§>J 3§  L
0004B0: CDA3 51ED 7918 F5CD B051 0707 0707 47CD   C y   QGGGGG
0004C0: B051 B0C9 7E23 D630 380C FE0A D8D6 07FE  Q  ~# 08L J  G
0004D0: 0A38 0CB8 67C8 3E2F C1C1 C30F 4400 0000 J8C g >/   ID§§§
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004F0: 0000 0000 C202 004D 0000 0000 0000 0000 §§§§B§M§§§§§§§§§
```



```
                ;Zap in SYS3/SYS für den Befehl IO.port#<,xx,yy,...>
5182                     ORG     5182h          ;hier ehemalige JKL-Rout.

                ;den zu adressierenden Port ermitteln
5182  CDA351             CALL    getval         ;eine Hexzahl einlesen
5185  4F                 LD      C,A            ;das ist der Port
5186  CDD54C             CALL    4cd5h          ;Trennz. und CR erkennen
5189  2011               JR      NZ,out         ;falls noch etwas folgt

                ;nein, nur Portadresse angegeben, Input ist gefragt
518B  ED78               IN      A,(C)          ;Port auslesen
518D  2A2040             LD      HL,(4020h)     ;Cursorposition
5190  CD6840             CALL    4068h          ;Input anzeigen
5193  3E0A               LD      A,0ah          ;Zeilenvorschub
5195  C33300             JP      0033h          ;ausgeben und raus

                ;die jeweiligen Outputs ermitteln und ausgeben
5198  CDD54C      loop   CALL    4cd5h          ;Trennz. und CR erkennen
519B  C8                 RET     Z              ;falls alles abgearbeitet
519C  CDA351      out    CALL    getval         ;nächste Hexzahl(en)
519F  ED79               OUT     (C),A          ;Output auf den Port
51A1  18F5               JR      loop           ;immer so weiter

                ;eine Hexzahl einlesen und auswerten
51A3  CDB051      getval CALL    getchr         ;eine Hexziffer einlesen
51A6  07                 RLCA                   ;ins obere Nibble
51A7  07                 RLCA                   ;schieben, also zuerst
51A8  07                 RLCA                   ;die höhere Stelle der
51A9  07                 RLCA                   ;Zahl nehmen
51AA  47                 LD      B,A            ;zwischenspeichern
51AB  CDB051             CALL    getchr         ;untere Sedezimalstelle
51AE  B0                 OR      B              ;mit d. oberen vereinigen
51AF  C9                 RET                    ;zur. mit kompletter Zahl

                ;eine Hexziffer einlesen und binär angleichen
51B0  7E          getchr LD      A,(HL)         ;nächst. Zeichen einlesen
51B1  23                 INC     HL             ;weiter im Befehlsstring
51B2  D630               SUB     '0'            ;Binärangleichung
51B4  380C               JR      C,error        ;falls Zeichen < '0'
51B6  FE0A               CP      0ah            ;Zeichen > '9'?
51B8  D8                 RET     C              ;fertig, falls Dezziffer
51B9  D607               SUB     07h            ;Ziffern A - F angleichen
51BB  FE0A               CP      0ah            ;Hex A oder größer?
51BD  3803               JR      C,error        ;falls kleiner
51BF  CB67               BIT     4,A            ;Wert > F?
51C1  C8                 RET     Z              ;fertig, falls höchst. F

                ;Fehlercode "falsche Parameter" in den Akku, NZ-Bedingung
51C2  3E2F        error  LD      A,2fh          ;Fehlercode
51C4  C1                 POP     BC             ;Stack korrigieren (CALL)
51C5  C1                 POP     BC             ;(2. CALL-Ebene)
51C6  C30944             JP      4409h          ;Fehleranzeige und zurück

0000                     END                    ;na ENDlich!

00000 Fehler

error  51C2     getchr 51B0     getval 51A3     loop   5198     out    519C
```

Arnulf Sopp

JKL - aber druckerschonend!

Die Idee zu diesem Zap in SYS26/SYS (G-DOS 2.4) bzw. in SYS3/SYS (G-DOS 2.1b und 2.2; bei NEWDOS je nach Computermodell) wuchs auf keinem geringeren Mist als dem der weiland Fa. TCS Computer GmbH. Für G-DOS 2.4 hat man nämlich wegen der unterschiedlichen Bildschirmformate der Genie-Modelle III und IIIs die Routine zur Screen-Hardcopy völlig neu geschrieben und, ihres beträchtlichen Umfangs wegen, nach SYS26 ausgelagert. Daraus folgt übrigens, daß die alte Routine in SYS3 für diese G-DOS-Version überflüssig geworden ist und lustig überschrieben werden darf. Aufmerksame Leser erinnern sich, daß dort nun mein IO-Programm residiert.

In der neuen JKL-Routine dachte man einen Schritt weiter als damals und ersetzte nicht druckbare Codes nicht etwa durch den wenig informativen Punkt, sondern durch ein Leerzeichen. Das ist keinen Deut informativer, schont aber den Druckkopf. Da muß der Programmierer jedoch mitten im Gedankengang eingeschlafen sein, denn das eben Gelobte gilt nur für Codes ab 7Fh. Die Codes zwischen 00 und 1Fh kriegen kurzerhand das Bit 6 gesetzt und sehen nun aus wie gewöhnliche Buchstaben, was sie nun einmal nicht sind. Der erste der drei Sektordumps zeigt, wie das aussieht. Greulich, wenn ihr mich fragt.

Die Abhilfe ist simpel: Wo das Bit 6 mit dem Befehl OR 40h gesetzt wird, kann man stattdessen mit dem Befehl LD A,20h den Akku mit einem Blank laden. Beide Befehle haben zwei Bytes, so daß akrobatische Verrenkungen entfallen. Der zweite Dump zeigt den dateirelativen Sektor 01 von SYS26 (G-DOS 2.4) in der geänderten Form, der nächste den Sektor 04 von SYS3 (sonstige). Die Änderung ist jeweils unterstrichen. Diese beiden Dumps sind bereits mit der neuen Routine erstellt.

Wer zu diesem Zweck SYS3 ändern müßte, aber in seinen Punkt verliebt ist, möge alles beim alten lassen. Sollte jemand mit einem entsprechenden SYS26 den Punkt bevorzugen, so wird in dem zu ändernden Befehl der Akku einfach mit 2Eh statt 20h geladen. Die beiden Bytes lauten dann 3E-2E statt 3E-20. Das ist bereits alles.

Es gibt eine dritte Möglichkeit. Wer als User von G-DOS 2.1b oder 2.2 unterhalb 20h gerne anstelle der Buchstabenentsprechungen Punkte sehen möchte, ändert ADD A,40h in LD A,2Eh (3E-2E statt C6-40). Das ist die obere der beiden unterstrichenen Modifikationen.

Arnulf Sopp

```
000400: E544 6973 6B65 7474 65EE 4174 7472 6962  Diskette Attrib
000410: 01EA 0051 75F4 7363 686C 6563 6874 65F2 A §Qu schlechte
000420: 4B65 696E E518 1818 6665 686C 65F2 6265 Kein XXXfehle be
000430: E975 6E7A 756C 6165 7373 6967 E508 F3FE  unzulaessig H
000440: 86C2 004D 5E23 5623 CDD5 4C38 19E5 0602   §M^#V#  LBY FB
000450: D521 5651 5E23 56D7 23E3 4FDF 79E3 2804  !VQ^#V # O v (D
000460: 10F2 3E34 E1E1 C209 44EF 4944 3D50 4FA5 F >4   ID ID=PO
000470: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
000480: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
000490: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004A0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004B0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004C0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004D0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0004F0: 0000 0000 0000 0000 0000 0000 0202 2BE1 §§§§§§§§§§§§BB+Q
```

```
000100: 673A FD37 0102 004E 6F19 E53A F637 6F3A g: 7   No  : 7o:
000110: F837 E603 FE03 2002 CB05 3AF1 3732 474E  7          : 72GN
000120: CD92 4CEB E1EB 19EB AF32 DD4D 32D9 4D01   L       2 M2 M
000130: 0000 CDD1 0528 070B 78B1 284C 18F4 EB2B        (  x  (L   +
000140: DF28 0BCD D54D FE20 28F5 23EB D5E5 1E40  (    M  ( #     §
000150: 1600 1943 2BCD D54D FE20 2003 10F6 04E1     C+  M
000160: 2277 4ED1 CDD5 4DFE 2030 023E 20CD 844E "wN   M  0 >   N
000170: 233A 4038 E604 2012 10EA 3A47 4E4F 2100 #:§8         :GNO!
000180: 0009 CD82 4EDF 38C4 AFC9 3E0D D55F 3A70       N 8    >   _:p
000190: 43BB 3002 1E20 7BCD 3B00 D1C9 5049 4F20 C 0    ä ;    PIO
0001A0: 034E 4F50 520D 3A07 43E6 07FE 043E 2AD8  NOPR : C      >*
0001B0: 7EFE 0D28 0BFE 5028 33FE 4E28 1AC3 604D ß  (  P(3 N(   `M
0001C0: 2194 4ECD 6744 2199 4E3A BD05 FED4 2002 ! N gD! N:
0001D0: 2323 CD67 44AF C921 E837 3E32 32BB 0522 ## gD  ! 7>22   "
0001E0: BC05 3E3A 32D1 0522 D205 18E9 3E07 D3D6   >:2  "     >
0001F0: D3D7 3E0F D3D6 3ECF D3D7 3EFE D3D7 3E01   >   >   >   >
```

```
000400: 8132 8744 E57D C605 01F2 FB50 6F06 08CD  2 D ü       Po
000410: 5951 7EFE 2028 0A3E 2FCD 3300 0603 CD59 YQß   ( >/ 3     Y
000420: 5121 6751 CD67 44CD 4900 CBAF FE45 2808 Q!gQ gD I      E(
000430: FE4E 2804 FE4A 20EF F5CD 3300 3E0D CD33  N(   J      3 >  3
000440: 00F1 E1FE 45C8 FE4E 2811 1180 443E 8012       E  N(     D>
000450: CD17 4EC0 E579 CD36 49E1 C07D E6E0 C620    N   y 6I   ü
000460: 6FD2 BC50 0C79 FE00 DAB7 50AF C97E FE20 o  P y      P  ß
000470: 23C4 3300 10F7 C900 2020 20C4 4461 7465 # 3              Date
000480: 6920 6C7C 7363 6865 6E20 203F 2028 4A2F i löschen  ? (J/
000490: 4E2F 4529 2003 2100 3C3A 4038 E604 2805 N/E)  ! <:§8   (
0004A0: 2100 4018 F47D E63F 3E0D CC3B 007C E63F ! §   ü ?>   : ö ?
0004B0: C87E FE20 2330 023E 2047 3A70 43B8 7830  ß   #0 > G:pC x0
0004C0: 023E 20CD 3B00 12D1 0D28 CB3A 8038 E601  >   :      ( : 8
0004D0: 4720 0D21 FF3B D7FE 8038 FB7C FE40 30B6 G   ! :     8 ö §0
0004E0: 3E07 CD3B 00CD 4900 E65F FE41 28A8 FE50 >   :   I   _ A(  P
0004F0: 2804 FE4E 20EA 4F3E D8EF 0000 0202 004D (   N   O>          M
```

Was tun bei voller Library?

In unserem Kirchenblättchen sind schon verschiedentlich Artikel mit Vorschlägen erschienen, wie man zusätzliche DOS-Befehle in die Library am Ende von SYS1/SYS implementieren kann. Wenn sie aber schon voll ist? Da gibt es DOSes, die für diesen Fall eine zweite Library in einem anderen SYS-File installieren. Wenn aber, wie beim Genie IIIs, nur noch eins frei ist, mit dem man Größeres vorhat?

Der letztere Weg ist gar nicht einmal so verkehrt. Da hierbei aber SYS1 überschrieben wird, müssen wesentliche Teile davon in der neuen SYS-Datei wiederholt werden. Das kostet Platz. Deshalb wurde hier ein Kompromiß gewählt: Ein neuer Library-Befehl leitet alle ganz-ganz neuen Library-Befehle ein. Er verzweigt in eine andere Systemdatei, wo sein Argument, nämlich der eigentliche Befehl, ausgewertet wird. Das hört sich umständlich an, ist es aber nicht:

Das Programm ist sehr einfach und kurz (s. Listing). Die Bedienung ist es nicht minder: Nehmen wir an, ein neuer Befehl soll ID lauten. Man gibt jetzt nicht ID,... ein, sondern SYS,ID,... Diese kleine Unbequemlichkeit ist wesentlich weniger unangenehm als der Verlust von viel Speicher- oder Diskettenplatz in einem teuren SYS-File. Aber zum Programm selbst:

Es residiert in SYS4/SYS, wo noch ausreichend Platz vorhanden war. Beim Einsprung wird zunächst geprüft, ob der Requestcode im Akku 86h lautet. Ist das nicht der Fall, muß es sich um eine andere DOS-Anforderung für SYS4 handeln. Also verzweigt das Programm in diesem Falle nach 4D00h, wo SYS4 beginnt und seine eigenen Checks durchführt. War es aber 86h, wurde also der Befehl SYS,... eingegeben, dann wird geprüft, was dem Befehl folgt.

Wegen der Platzökonomie sind hier ein paar Vereinbarungen verbindlich: Der Folgebefehl darf nur zwei Zeichen lang sein. Dann läßt er sich nämlich mit RST 18h, einem Vergleich von HL und DE identifizieren. Er darf keinen Restriktionen unterliegen wie beispielsweise einem Verbot unter Mini-DOS. Zwar wäre auch das machbar, aber es gälte dann für alle Zusatzbefehle.

Wie der zusätzliche Befehl erkannt wird, geht aus den Kommentaren des Listings hervor. Es muß erwähnt werden, daß bei einer Verlängerung der Zweit-Library jeweils auch die neue Anzahl der verfügbaren Befehle in Zeile 11 aktualisiert werden muß. Dort wird B als Zähler für die Suchdurchläufe geladen. Die neue Befehlstabelle selbst (cmdtab) besteht außer dem Befehlswort noch aus dem Requestcode für den Akku, der einen Wegweiser in das zutreffende SYS-File zu seiner Bearbeitung darstellt. Das genügt in allen Fällen, denn in keiner Systemdatei werden alle möglichen Codes voll ausgeschöpft. Platzersparnis, wie gesagt.

Die beiden Sektordumps zeigen das neue Ende von SYS4 mit der Serviceroutine für den SYS-Befehl sowie die Library in SYS1, wo der Befehl SYS eingezapt ist. Dort stand früher LIB, das aber überflüssig ist. Erstens kann es durch ? ersetzt werden, zweitens sollte man seine DOS-Befehle auswendig kennen. Und wer sie nicht kennt, ist ohnehin besser beraten, für nähere Informationen das Handbuch zu konsultieren.

Arnulf Sopp


```
512B               00001        ORG    512bh        ;ab hier Platz in SYS4
512B  FE86         00002 start  CP     86h          ;SYS-Befehl?
512D  C2004D       00003        JP     NZ,4d00h     ;falls nein
5130  5E           00004        LD     E,(HL)       ;1. Buchstabe des Argum.
5131  23           00005        INC    HL           ;2. Buchstabe
5132  56           00006        LD     D,(HL)       ;jetzt SYS-Argument in DE
5133  23           00007        INC    HL           ;auf Delimiter stellen
5134  CDD54C       00008        CALL   4cd5h        ;" " oder "," oder CR?
5137  3819         00009        JR     C,error      ;Fehler, falls anderes Z.
5139  E5           00010        PUSH   HL           ;Befehlszeiger retten
513A  0602         00011        LD     B,02h        ;bisher 2 neue Befehle
513C  D5           00012        PUSH   DE           ;Argument auf den Stack
513D  215651       00013        LD     HL,cmdtab    ;Anfang Befehlstabelle
5140  5E           00014 loop   LD     E,(HL)       ;1. Zeichen
5141  23           00015        INC    HL           ;2. Zeichen
5142  56           00016        LD     D,(HL)       ;DE <- Befehl aus Tabelle
5143  D7           00017        RST    10h          ;Requestcode in den Akku
5144  23           00018        INC    HL           ;Zeiger auf nächsten Bef.
5145  E3           00019        EX     (SP),HL      ;Zeig. m. Arg. vertausch.
5146  4F           00020        LD     C,A          ;Akku retten
5147  DF           00021        RST    18h          ;Befehl gefunden?
5148  79           00022        LD     A,C          ;Akku restaurieren
5149  E3           00023        EX     (SP),HL      ;zurücktauschen
514A  2804         00024        JR     Z,exit       ;falls ja
514C  10F2         00025        DJNZ   loop         ;falls noch nicht
514E  3E34         00026        LD     A,34h        ;Fehlercode
5150  E1           00027 exit   POP    HL           ;Stack bereinigen
5151  E1           00028        POP    HL           ;(2 Ebenen)
5152  C20944       00029 error  JP     NZ,4409h     ;Fehleranzeige und zurück
5155  EF           00030        RST    28h          ;zur Bearbeitungsroutine
5156  49           00031 cmdtab DM     'ID',3dh     ;DOS-Befehl ID in SYS27
5159  49           00032        DM     'ID',0a5h    ;DOS-Befehl ID in SYS3
512B               00033        END    start
```

```
000400: E544 6973 6B65 7474 65EE 4174 7472 6962  Diskette Attrib
000410: 01EA 0051 75F4 7363 686C 6563 6874 65F2    Qu schlechte
000420: 4B65 696E E518 1818 6665 686C 65F2 6265 Kein    fehle be
000430: E975 6E7A 756C 6165 7373 6967 E508 F3FE  unzulaessig
000440: 86C2 004D 5E23 5623 CDD5 4C38 19E5 0602     M^#V#   L8
000450: D521 5651 5E23 56D7 23E3 4FDF 79E3 2804  !VQ^#V # O y (
000460: 10F2 3E34 E1E1 C209 44EF 4944 3D50 4FA5   >4     D ID=PO
000470: 0000 0000 0000 0000 0000 0000 0000 0000
000480: 0000 0000 0000 0000 0000 0000 0000 0000
000490: 0000 0000 0000 0000 0000 0000 0000 0000
0004A0: 0000 0000 0000 0000 0000 0000 0000 0000
0004B0: 0000 0000 0000 0000 0000 0000 0000 0000
0004C0: 0000 0000 0000 0000 0000 0000 0000 0000
0004D0: 0000 0000 0000 0000 0000 0000 0000 0000
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000
0004F0: 0000 0000 0000 0000 0000 0000 0202 2B51                 +Q
```

```
000300: 4D82 E900 4980 2A00 494E 464F 0102 0050 M   I * INFO   P
000310: 81FF 004A 4B4C 807C 104B 494C 4C80 4590    JKL ö KILL E
000320: 4C43 88E5 004C 4681 FE00 5359 5380 8600 LC   LF   SYS
000330: 4C49 5354 85F0 884C 4F41 4480 A450 4D3E LIST   LOAD  PM>
000340: 82EB B04E 81E4 B04E 4446 C028 0050 4155    N   NDF ( PAU
000350: 5345 98EB 0050 4483 E900 5049 4F80 9C00 SE   PD   PIO
000360: 504F 5254 82FF 0050 5249 4E54 86F0 8850 PORT   PRINT   P
000370: 524F 5486 E900 5055 5247 4589 E900 5280 ROT   PURGE   R
000380: 2300 5381 E900 5349 4F80 BD00 5354 4D54 # S   SID   STMT
000390: 89EB 0055 4852 82E5 0056 2B84 E500 5A81    UHR   V+   Z
0003A0: FE00 5A45 4954 8AE9 005A 4C82 F882 3084    ZEIT   ZL   0
0003B0: F000 3634 8193 0038 3082 9800 2193 EB8A    64   80   !
0003C0: 2323 8398 0026 83E5 0040 81F0 003B 86E3 ##   &   §   :
0003D0: 002F 85E3 003E C048 003F 82E3 0000 0021   /   > H ?    !
0003E0: 584F 0540 0808 7ECB 7F23 2005 CD57 5110 XO §  B  #    Q
0003F0: FE23 237F E7C4 B551 0DC0 B551 28E4 CDAD  ##G   O   Q(
```

DDE für das Genie III s

Seit ein paar Tagen steht in meiner Gummizelle ein neuer Komposter namens G3s. Sein Betriebssystem, G-DOS 2.4, steckt zwar so manches andere in die Tasche, aber gewisse H-DOS-Features vermisse ich doch sehr. Bei der ewigen Zapperei ist es natürlich der Library-Befehl DDE in SYS15/SYS, den ich am meisten brauche. Und so ist denn logischerweise SYS15/SYS das erste File, das in Richtung H-DOS gestreamlined wurde:

Im Gegensatz zur Version für das G1/2 lädt diese SYS-Datei in den DOS-Overlay-Bereich. Man kann sie deshalb sogar unter Mini-DOS benutzen. Das Argument, man solle den Disketten-Editor lieber als CMD-File fahren, zieht deshalb beim G3 nicht mehr.

Beim Einsprung wird zunächst geprüft, ob der Speicherbereich ab 3000h benutzt werden darf oder gerade von einem anderen Modul des Betriebssystems belegt ist. Wenn er nicht frei ist, wird automatisch die Datei DDE52/SYS aufgerufen, die mit dem SYS15/SYS für das G1/2 praktisch identisch ist. Klarer Fall: In DDE52/SYS wurde sofort SYS15/SYS aus H-DOS mit geringfügigen Änderungen einkopiert. Diese Änderungen werden hier nicht näher benannt, denn sie erscheinen auch weiter unten im modifizierten SYS15/SYS für das G3s.

Wenn in 3000h ff. aber Platz ist, verlagert sich der Hauptteil der Datei dorthin. Das Bildschirmformat wird auf 16 X 64 Zeichen eingestellt und das Programm an seinem neuen Platz schließlich angesprungen.

Ein paar Programmschritte zuvor hakt die erste Änderung ein: Im M-Modus, wenn also ein Sektor modifiziert werden soll, erscheint ein Cursor. Das ist in der Originalversion ein Graphikblock, der das betreffende Zeichen natürlich überdeckt. Schade drum! Das G3s kann Zeichen invers darstellen. Also ruft die neue Version von DDE zunächst in SYS22/SYS den Befehl Z,YR auf, der von Graphik auf inverse Zeichen ab 80h umstellt. Das sind im anhängenden Programmlisting die Zeilen 58-64. Der Z-Befehl braucht dabei nicht zu erscheinen, da das Programm ökonomischer direkt mit RST 28h arbeitet. Nur das Argument YR, gefolgt von CR (ENTER), erscheint explizit an der Stelle invpar (Zeile 479, Parameter für Inversdarstellung). Übrigens sind diejenigen Teile des Listings, die hier nicht von Belang sind, mit LIST OFF ausgespart. Die Auslassungen sind durch Striche zwischen den Zeilennummern gekennzeichnet.

Die Änderung in den Zeilen 169-171 und 178-180 bezieht sich direkt auf die Form des Cursors: Das gerade angezeigte Zeichen wird jetzt nicht mehr vom Graphikblock mit dem Code 8Fh zugeschaufelt, sondern stattdessen wird sein Bit 7 invertiert, so daß das inverse Zeichen jetzt im Akku steht. Es wird statt seiner in den Bildschirm geladen.

Im gleichen Zusammenhang sind die Zeilen 380 und 381 zu sehen: der Code, der aus dem Sektorpuffer geholt wird (DE als Zeiger), wird direkt und ohne Änderungen in den Bildschirm geladen (HL als Zeiger). Da alle 256 Codes mit dem G3s anzeigbar sind, ist es nicht mehr erforderlich, einzelne von ihnen durch Punkte zu ersetzen.

Bis hierher beziehen sich die Modifikationen nur auf das optische Erscheinungsbild, sind also nur ebenso sinnvoll oder sinnlos wie ein auf Diskette sortiertes Directory (das beim nächsten Dateieintrag eh' wieder durcheinandergerät). Eine wirklich sinnvolle, geradezu notwendige Änderung ist in den Zeilen 300-328 programmiert:

Beim Umschalten mit CLEAR vom linken Hex- ins rechte ASCII-Feld geht im M-Modus der Cursor in der Originalversion immer in die jeweilige linke obere Ecke. Modifiziert man also gerade ziemlich weit unten im Sektor und will beispielsweise zwischen zwei Wörtern ein Steuerzeichen eingeben, das nicht über die Tastatur erreichbar ist, dann darf man zunächst mit den

Pfeiltasten Schiffeversenken spielen. In der hier vorgestellten Version geht der Cursor im Nachbarfeld immer genau an dasselbe Byte, so daß man sofort weitertippen kann. Und das geht so:

Beim Ansprung dieses Programmsegments steht die Cursoradresse gerade in HL. Der Anfang der aktuellen Bildschirmzeile wird durch UND-Verknüpfung des Cursor-LSB mit C0h gewonnen und vorübergehend in L verstaut. Jede Bildschirmzeile bezieht sich auf 16 Bytes des Sektorpuffers. Dessen Zeiger-LSB ist E. Mit 0Fh verknüpft, ergibt es die aktuelle Stelle in einem jeden Block von je 16 Bytes. Sie wird in B kühl und trocken aufbewahrt. Jetzt sind die passende Videozeile und das laufende Byte in ihr dingfest gemacht. Ein schlichtes LD A,(HL) o. ä. hätte leider nicht genügt, denn wäre der Cursor gerade im Hexfeld gewesen, hätte diese Operation nur ein Nibble erbracht, obendrein in ASCII-Darstellung.

Jetzt wird in Zeile 306 das LSB der aktuellen Videoadresse aus ihrem Puffer geladen. Nun geschieht der eigentliche Wechsel von ASCII nach Hex bzw. umgekehrt. Weshalb er ausgerechnet durch die Exklusiv-ODER-Verknüpfung mit 38h vollzogen wird, wäre einen eigenen Artikel über das Liebesleben der Bits wert. Freuen wir uns einfach, daß mir dieser kurze Weg eingefallen ist und reden wir nicht weiter drüber! Das neue Video-LSB wird anschließend in den Puffer geschrieben.

Für das Weitere muß das Programm nun wissen, ob das gerade ASCII oder Hex wurde. Wenn im LSB der Cursoradresse das Bit 4 oder 5 gesetzt war (= 30h), dann war es wohl das ASCII-Feld, denn im rechten Viertel jeder Bildschirmzeile sind diese Bits immer 1. Das LSB des Zeilenanfangs wird aus dem Datenfriedhof L in den Akku zurückgeladen. Falls ASCII, erfolgt nun der Sprung zum Label ascii (Zeile 325). Auch was dort geschieht, ist wieder binäre Akrobatik, die ich hier nicht erläutern will. Letztenendes kommt das neue Bildschirm-LSB in den halben Zeiger L, und die Arbeit ist getan.

Ein bißchen schwieriger ist es, wenn soeben ins Hexfeld umgeschaltet wurde. Die erste Hex-Ziffer erscheint an der achten Stelle, weil am Zeilenanfang zuerst der Kopf mit Sektor- und Zeilennummer steht. Vorab kommt dieses vorläufige Cursor-LSB nach L. Weiter oben war die Stelle in einem laufenden 16er-Block des Sektorpuffers nach B gerettet worden. Wenn es die nullte Stelle war, ist es damit erledigt. Andernfalls wird zunächst die halbe Videostellenzahl ermittelt, um sie später als Anzahl der trennenden Blanks hinzuaddieren zu können. Danach werden für jedes Byte im Puffer zwei Stellen addiert, denn es werden ja je zwei Nibbles angezeigt. Die Summe der Nibbles und Blanks ist schließlich das LSB der tatsächlichen Bildschirmadresse.

Uff! In einer letzten mörderischen Anstrengung habe ich schließlich noch in den Zeilen 411 und 412 klargemacht, daß dies eine neue Version von DDE ist, und wer da wann mitgemischt hat.

Abschließend ein Wort zum Listing. Da sich die Versionen von SYS15/SYS für G1/2 und G3/3s unterscheiden, mußte zunächst eine Disassembly erstellt und näher untersucht werden. Die große Mehrzahl der absoluten Adressen (bei CALLs, JPs, Speicherladebefehlen usw.) bezog sich darin auf den Bereich ab 3000h nach der Relokation des Hauptprogramms. Da aber die Modifikationen nahezu alle Adressen veränderten, mußte jede dieser Adressen auf die Ladeadresse vor dem Programmstart bezogen werden. Deshalb wurde das Label offs (Offset zur Arbeitsadresse) eingeführt. Als Subtrahend eingesetzt, korrigierte es alle Verschiebungen durch die Manipulationen.

Der Quelltext wurde mit ZEUS/CMD erstellt. Dieser Assembler erlaubt eine sehr bequeme Programmierung. Es dürfte für den EDTASM-gewohnten Assembler-Crack kein Problem sein, die Programmlogik zu verstehen.

Das komplette Listing des Programms SYS15/SYS wäre für unsere Postille nicht nur zu lang, sondern es würde sich garantiert keiner der Mühe unterziehen, es abzutippen. Recht so! Wenigstens ein Hexdump der ersten vier Sektoren ist beigefügt, damit der geneigte Leser und Selbstmodifizierer an der Umgebung erkennt, was sich wo gerade abspielt. Im Dump sind die Änderungen gegenüber Version 2.1 unterstrichen. Da Teile gelöscht, andere Teile hinzugefügt sind, finden sich die unterstrichenen Segmente nicht unbedingt an derselben Stelle wie im TCS-Original.

Wer für sein G3s die Version 2.2 haben will, mag mir einen frankierten und adressierten Umschlag mit einer Leerdiskette zuschicken (bitte mit der üblichen G-DOS-Formatierung, sonst Zettelchen nicht vergessen!). Ohne das G3 (ohne s) zu kennen, vermute ich, daß das Programm auch auf dieser Maschine ohne größere Änderungen laufen müßte.

Arnulf Sopp

```
4D48  E5       00058 m4d48  PUSH  HL                 ;Befehlszeiger retten
4D49  21DF51   00059        LD    HL,invpar          ;YR für Z-Befehl
4D4C  3EF8     00060        LD    A,0f8h             ;für SYS22/SYS
4D4E  0E01     00061        LD    C,01h              ;für Z,YR
4D50  CD0244   00062        CALL  4402h              ;dort RST 28h
4D53  E1       00063        POP   HL                 ;Befehlsz. restaurieren
4D54  11D732   00064        LD    DE,m32d6-offs      ;weiter wie gehabt
4E31  3E80     00169        LD    A,80h              ;Bit 7 = 1
4E33  AE       00170        XOR   (HL)               ;Zeichen invertieren
4E34  77       00171        LD    (HL),A             ;und neu laden
4E3D  3E80     00178        LD    A,80h              ;Bit 7 = 1
4E3F  AE       00179        XOR   (HL)               ;Zeichen invertieren
4E40  77       00180        LD    (HL),A             ;und neu laden
4F00  7D       00300 m31a8  LD    A,L                ;LSB der Cursoradresse
4F01  E6C0     00301        AND   0c0h               ;auf Zeilenanfang
4F03  6F       00302        LD    L,A                ;zwischenspeichern
4F04  7B       00303        LD    A,E                ;LSB Pufferstelle
4F05  E60F     00304        AND   0fh                ;Stelle im 16er-Block
4F07  47       00305        LD    B,A                ;zwischenspeichern
4F08  3ABD32   00306        LD    A,(m32bc-offs)     ;Video-LSB
4F0B  EE38     00307        XOR   38h                ;ASCII-Feld <-> Hexfeld
4F0D  32BD32   00308        LD    (m32bc-offs),A     ;Video-LSB neu ablegen
4F10  E630     00309        AND   30h                ;jetzt ASCII-Feld?
4F12  7D       00310        LD    A,L                ;LSB d. Cursoradr. zurück
4F13  2011     00311        JR    NZ,ascii           ;falls jetzt ASCII-Feld
4F15  F608     00312        OR    08h                ;Hexfeld, auf 8. Stelle
4F17  6F       00313        LD    L,A                ;Hexfeld-Zeilenanfang
4F18  AF       00314        XOR   A                  ;A <- 00
4F19  B0       00315        OR    B                  ;0. Stelle im 16er-Block?
4F1A  C8       00316        RET   Z                  ;erledigt, falls ja
4F1B  48       00317        LD    C,B                ;C <- Stellenzahl
4F1C  CB39     00318        SRL   C                  ;die Hälfte = Blanks
4F1E  7D       00319        LD    A,L                ;Zeilenanfang
4F1F  C602     00320 hex    ADD   A,02h              ;je 2 Stellen/Byte
4F21  10FC     00321        DJNZ  hex                ;fortgesetzt addieren
4F23  81       00322        ADD   A,C                ;+ Anzahl der Blanks
4F24  6F       00323        LD    L,A                ;neue Cursoradresse
4F25  C9       00324        RET                      ;erledigt
4F26  F630     00325 ascii  OR    30h                ;auf letzte 16 Stellen
4F28  B0       00326        OR    B                  ;+ Stelle im 16er-Block
4F29  6F       00327        LD    L,A                ;neue Cursoradresse
4F2A  C9       00328        RET                      ;erledigt
4F87  1A       00380 m4f58  LD    A,(DE)             ;ASCII-Code laden
4F88  77       00381        LD    (HL),A             ;unverändert anzeigen
4FBA  1C       00410 m324d  DB    1ch,1fh
4FBC  47       00411        DM    'Genie-DOS Diskdaten-Editor Vers. 2.2'
4FE0  0A       00412        DM    0ah,0ah,'(C) 1986 TCS/HACKTORY',0dh
51D7  0000     00478 m51e0  DW    0000h,0000h,0000h,0000h
51DF  59       00479 invpar DM    'YR',0dh
1D48           00480 offs   EQU   m4d48-3000h

00000 Fehler
```



```
000000: 0102 004D 3A07 43E6 0FFE 0238 1BE5 214B AB§M:GC O BBA !H
000010: 4D11 0030 0100 04ED B021 0030 E33A 1419 MO§0A§D  !§0 :TY
000020: FE40 3E10 C233 00C9 3A6A 43CB 7F3E 38C0  §>F 3§ :jC  >8
000030: E501 0A00 213E 4D11 D751 D5ED B0D1 E1C3  AJ§!>MO Q
000040: 3344 4444 4535 322F 5359 530D E521 DF51 3DDDE52/SYSM ! Q
000050: 3EF8 0E01 CD02 44E1 11D7 32CD 1C44 2827 > NA BD Q 2 öD('
000060: 2172 32CD 6744 21B0 32CD 6744 0618 21BE !r2 gD! 2 gDFX!
000070: 32CD 4000 D87E FE23 281A FE2A 2830 11D7 2 §§ ß #(Z *(0Q
000080: 32CD 1C44 C251 3206 0021 0033 CD24 44C4 2 öD Q2F§!§3 $D
000090: 5A32 1807 237E E60F 32DD 32ED 4BE1 3211 Z2XG#ß O2 2 K 2Q
0000A0: D732 CD42 44CD 3644 C45A 32CD 4544 CD0A  2 BD 6D Z2 ED J
0000B0: 32CD 4B32 FE01 28A8 FE3B 2836 FE2B 2815 2 K2 A(  ;(6 +(U
0000C0: FE3D 281E FE2D 281F CBAF FE53 2839 FE4D  =(^ -(_   S(9 M
0000D0: CAD7 3018 D9ED 4BE3 323A DF32 B720 C00B   0X  K 2: 2   K
0000E0: 18BD 0100 0018 B8ED 4BE1 3278 B12B BF0B X A§§X  K 2x ( K
0000F0: 18AD ED4B E132 032A E332 3ADF 32B7 2001 X  K 2C* 2: 2  A

000100: 2BED 4230 0102 004E 9A18 A701 0000 CD4B + BOAB§N X A§§ K
000110: 32FE 2138 E6C5 CDF7 31C1 38F2 6069 2929 2 !8    1 8 `i))
000120: 2929 854F 4418 E711 0033 263C 3ABD 326F )) ODX Q§3&<: 2o
000130: CD0A 327C E603 C63C 673E 80AE 773A BD32  J2ö C <g>  w: 2
000140: D608 2006 233E 80AE 772B 01E0 30C5 CD4B  H F#>  w+A 0  K
000150: 32FE 0928 46FE 0928 64FE 0828 6CFE 0ACA 2 I(F K(d H(l J
000160: A031 FE40 CAAC 31FE 1FCA B831 FE0D CAE3  1 §  1 _  1 M
000170: 31FE 01CA F331 473A BD32 D608 7820 1ACD 1 A  1G: 2 Hx Z
000180: F731 38CA 7017 1717 174F CD4B 32CD F731  18 pWWWWO K2  1
000190: 38F8 2370 2B81 1218 0277 123A BD32 D608 8 #p+ RXBwR: 2 H
0001A0: 2010 7B0F 7DCE 026F 1C7B E60F C001 1800  PäOü Booä O AX§
0001B0: 09C9 231C 7BE6 0FC0 0130 0009 C901 C0FF I #öä O AO§I A
0001C0: 097B D610 5FD0 263F C93A BD32 D608 2013 Iä P_ &? : 2 H S
0001D0: 7B0F 7D3F DE02 6F1D 7BE6 0FEE 0FC0 01E8 äOü? Boüä O O A
0001E0: FF09 C92B 1D7B E60F EE0F C001 D0FF 09C9  I +üä O O A  I
0001F0: 0140 0009 7BC6 105F D026 3CC9 D53A BD32 A§§Iä P_ &<  : 2

000200: E620 121C 20FC D1C9 0102 004F 7DE6 C06F   Rö   AB§Oü  o
000210: 7BE6 0F47 3ABD 32EE 3832 BD32 E630 7D20 ä OG: 2 82 2 Oü
000220: 11F6 086F AFB0 C848 CB39 7DC6 0210 FC81 Q Ho   H 9ü BF
000230: 6FC9 F630 B06F C9C1 11D7 32CD 3944 C45A o  0 o  Q 2 9D Z
000240: 32CD 4544 C362 30C1 C34F 3047 D630 D8FE 2 ED b0  OOG 0
000250: 0A38 09CB AFFE 11D8 D607 FE10 3FC9 D921 J8I   Q  G P?  !
000260: 003C 1100 33D5 0608 3AE2 32CD 6840 3AE1 §<Q§3 FH: 2 h§:
000270: 32CD 6840 7BCD 6840 3E3A 7723 3E20 7723 2 h§ä h§>:w#> w#
000280: 1ACD 6840 1C1A CD68 401C 10F0 3E20 7723 Z h§öZ h§öP > w#
000290: D106 101A 7723 1C28 0410 F818 C8D9 C9D5  FPZw#ö(DP X
0002A0: CD49 00D1 C921 B932 CD67 44C3 1A30 F6C0  I§  ! 2 gD Z0
0002B0: FEC6 2808 3333 CD09 44C3 1A30 3AD7 32F6   (H33 ID Z0: 2
0002C0: 0132 D732 AFC9 1C1F 4765 6E69 652D 444F A2 2  ö_Genie-DO
0002D0: 5320 4469 736B 6461 7465 6E2D 4564 6974 S Diskdaten-Edit
0002E0: 6F72 2056 6572 732E 2032 2E32 0A0A 2843 or Vers. 2.2JJ(C
0002F0: 2920 3139 3836 2054 4353 2F48 4143 4B54 ) 1986 TCS/HACKT

000300: 4F52 590D 0A44 6174 6569 3F20 0102 0050 ORYMJDatei? AB§P
000310: 0307 1B1B 0330 3031 3233 3435 3637 3839 CGAAC00123456789
000320: 4142 4344 4546 3031 3233 3435 3637 0D82 ABCDEF01234567M
000330: 2800 0033 0000 0000 0000 00FF FFFF FFFF (§§3§§§§§§§
000340: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FF00                §
000350: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
000360: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
000370: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
000380: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
000390: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0003A0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0003B0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0003C0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0003D0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0003E0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
0003F0: 0000 0000 0000 0000 0000 0000 0000 0000 §§§§§§§§§§§§§§§§
```

Die HRG des Genie III s

Bis ich mich endlich entschlossen hatte, bei meinem alten Genie I die HRG 1b einzubauen, vergingen seit dem Kauf des Computers über zwei Jahre. "Spielkram!" war mein Argument dagegen. Aber die zusätzlichen 12 kB Speicher waren schließlich doch zu verlockend. Beim G3s ist von Hause aus eine Baugruppe für hochauflösende Graphik drin. Sie umfaßt sogar zweimal 32, also zusammen 64 kB. So kann sich der geneigte Leser lebhaft meinen Ärger vorstellen, als in der Dokumentation (wie bei Trommeschläger üblich nicht ausreichend) keine brauchbare Erklärung darüber zu finden war, wie man diesen Speicher anspricht.

Es wird ein sehr interessantes Graphik-BASIC auf der G-DOS-Diskette mitgeliefert. Aber es braucht den Speicher für die absonderlichsten Luxusroutinen auch dann, wenn man bloß eben mal irgendwo einen Punkt hinsetzen möchte. Und letztenendes ist es natürlich nur unter BASIC nutzbar; für den Assembler-Crack ein gewichtiger Grund dagegen. Also war der Mann mal wieder selbst:

Aus der technischen Beschreibung geht (zumindest nach mehrmaligem Lesen und Probieren) hervor, daß der Graphikspeicher von 8000-FFFF geht. Er kann anstelle des Hauptspeichers durch einen Output auf den Port FA mit gesetztem Bit 3 der CPU zugänglich gemacht werden. Da dieser Port noch weitere Funktionen des Systems steuert, muß er zunächst gelesen und an diesem Input nur bitweise gefummelt werden, damit der bisherige Betriebszustand nicht in fataler Weise verändert wird. Jetzt kann in 8000-FFFF beliebig geschrieben und gelesen werden. Dieser Adreßraum ist jetzt nach dem Umschalten über Port FA der Graphikspeicher.

Der logische Aufbau der HRG des G3s ist der HRG 1b ganz ähnlich. Auch hier entspricht ein bestimmtes Byte des Speichers der ganzen Breite einer Punktzeile einer normalen Anzeigestelle des Bildschirms. Deshalb sind auch hier die LSB der Graphik- und Bildschirmadressen gleich. Die Reihenfolge der Bits (Graphikpunkte) ist ebenfalls dieselbe: Bit 0 wird ganz links angezeigt, Bit 7 rechts (die HRG 1b zeigt nur bis Bit 5 an). Wegen der großen Ähnlichkeit mit der HRG 1b, die mir sehr vertraut ist, erschien deshalb die Aufgabe lösbar, der HRG des G3s auf die Schliche zu kommen.

Aber da ist eine Merkwürdigkeit, die vorab geklärt werden mußte: Jeder Anzeigemodus (25 X 80, 16 X 64, 24 X 64 und 32 X 64 Zeichen) benutzt einen unterschiedlich großen Bildschirmspeicher. Außerdem werden unterschiedlich tiefe Punktspalten pro Bildschirmbyte angezeigt. Nur im Modus 32 X 64 kommen 32 X 64 X 16 X 8, also 512 X 512 = 262144 Punkte, das sind 32 kB, zur Anzeige. Was geschieht in den anderen Modi mit dem übrigbleibenden Graphikspeicher? Ein Testprogramm, das im Anschluß an diesen Artikel aufgelistet ist, brachte Aufschluß.

Das Programm hat seinen Zweck erfüllt. Es mündete in die beiden Tabellen hinter dem Listing. Da es deshalb niemand abzutippen braucht, seien hier nur diejenigen Teile erläutert, die die Graphik ansprechen, denn diese Teile kann der Leser in seine eigenen Programme einbauen:

Im DOS liegt der Stack ziemlich tief unten. Um aber das Programm notfalls auch fahren zu können, wenn der Stack im Himem liegt, war es notwendig, den Stackpointer zu retten und neu zu laden. Nach dem Umschalten würde er zwar normal arbeiten, wäre aber in der Graphik sichtbar.

Der nächste Schritt (Unterprogramm change) sollte an wichtigen Stellen des Bildschirm-RAMs Marken setzen, so daß die jeweilige Video-Adresse ablesbar ist. Es sind die Anfangs- und Endadressen des Bildschirms bei jedem der vier möglichen Bildschirmformate. Da im unteren Adreßraum der großen Formate gleichzeitig die Tastatur memory mapped ist,

mußte sie zunächst über das Bit 4 des Ports FA weggebankt und anschliessend wieder enabled werden. Der einleitende CALL nach 01C9h löscht den Bildschirm und aktiviert gleichzeitig Formatänderungen (s. u.).

Das Löschen des Graphikspeichers ist einfacher als mit der HRG 1b: Da die CPU direkten Zugriff hat, braucht er nur mit LDIR ausgenullt zu werden. Die anschließend folgende Abfrage der F-Tasten, mit denen das Programm zwischen den Bildschirmformaten hin- und herspringen kann, soll hier nicht weiter erklärt werden. Wie aber die Formate gewechselt werden, ist wieder wichtig für den Graphik-Programmierer:

Ab 3760h stehen vier Tabellen zu je 16 Bytes, die die Bildschirmparameter enthalten. Die Parameter für das gerade benutzte Format stehen ab 37F0h. Deshalb werden beim Umschalten des Formats die neuen Parameter, wieder mit LDIR, dorthin geladen. Das alleine genügt allerdings nicht. Aktiv werden sie erst, wenn gewisse Control-Codes ausgegeben werden. Das ist z. B. auch beim Löschen des Bildschirms der Fall, deshalb wird nach dem Formatwechsel erneut change aufgerufen.

Das eigentliche Arbeitsprogramm steht ab loop. Es sei nur so viel dazu gesagt, daß ein Graphik-Cursor mit den Pfeiltasten über den Bildschirm gejagt werden kann. Mit der Blank-Taste kann er verlangsamt werden, um einzelne Bildschirmstellen gezielt ansteuern zu können. Die jeweilige Adresse des Cursors im Graphikspeicher wird ständig angezeigt. Wenn dieser Cursor nun eine der im UP change gesetzten Marken durchläuft, kann man sich die Bildschirmadresse der Marke und die Graphikadresse des Cursors notieren und erfährt so den Zusammenhang zwischen beiden.

Für den Computergraphiker wird es nun ab der Stelle exit wieder interessant: Der alte Stackpointer wird wieder aus dem Puffer geladen. Die Bits 1 und 3 des Systembytes 1 (Port FA) werden wieder auf 0 gesetzt, wodurch die Graphik vom Bildschirm verschwindet und der Hauptspeicher anstelle des Graphikspeichers in die obere Adreßhälfte eingeblendet wird. Die Interrupts waren die ganze Zeit über ausgeschaltet, denn bei zeitweise ausgebankter Tastatur und manipuliertem Stack ist Vorsicht angebracht. Mit EI werden sie wieder enabled. Mit RET geht es schließlich zurück ins Betriebssystem.

Die beiden Tabellen sind folgendermaßen zu interpretieren: Wie auch bei der HRG 1b, so kann man sich für die jeweils oberste Punktzeile eines Zeichens den ganzen Bildschirm als eine ununterbrochene Perlenkette vorstellen. Wenn die oberste Zeile durchlaufen ist (also rechts unten im Bildschirm!), dann geht es in der zweiten Punktzeile (oben links!) weiter. Beim Anzeigeformat 32 X 64 geschieht das lückenlos.

Die anderen Formate zeigen aber Unterbrechungen beim Zeilenwechsel. Besonders die Graphik auf der letzten Seite dieses Artikels dürfte das anschaulich machen. Aus beiden Tabellen geht auch hervor, daß pro Bildschirmstelle bei den kleineren Formaten auch nicht alle Punktzeilen zur Anzeige kommen. Die Punktzeilen sind von 0 - F durchnumeriert.

Zur praktischen Anwendung: Die Tabellen zeigen, welche Bereiche des Graphikspeichers überhaupt im Bildschirm sichtbar werden können, je nach Format. Die Abszisse eines Graphikbytes errechnet sich nun ebenso wie die eines Buchstabens im normalen Bildschirm. Um aus diesem Beitrag kein Buch zu machen, verweise ich hierzu auf die Vielzahl von Artikeln zur HRG 1b, die an dieser Stelle erschienen sind.

Die Ordinate, also die Lage eines Bytes in senkrechter Richtung, ist vom darüber-/darunterliegenden Byte immer genau 2 kB im Graphikspeicher entfernt, solange es sich um dieselbe Videozeile handelt. Beim Sprung in die nächste Videozeile (nicht zu verwechseln mit den Punktzeilen eines Zeichens!) ist einfach zu einer Adresse die Zeilenlänge hinzuzuaddieren, also 80 oder 64.

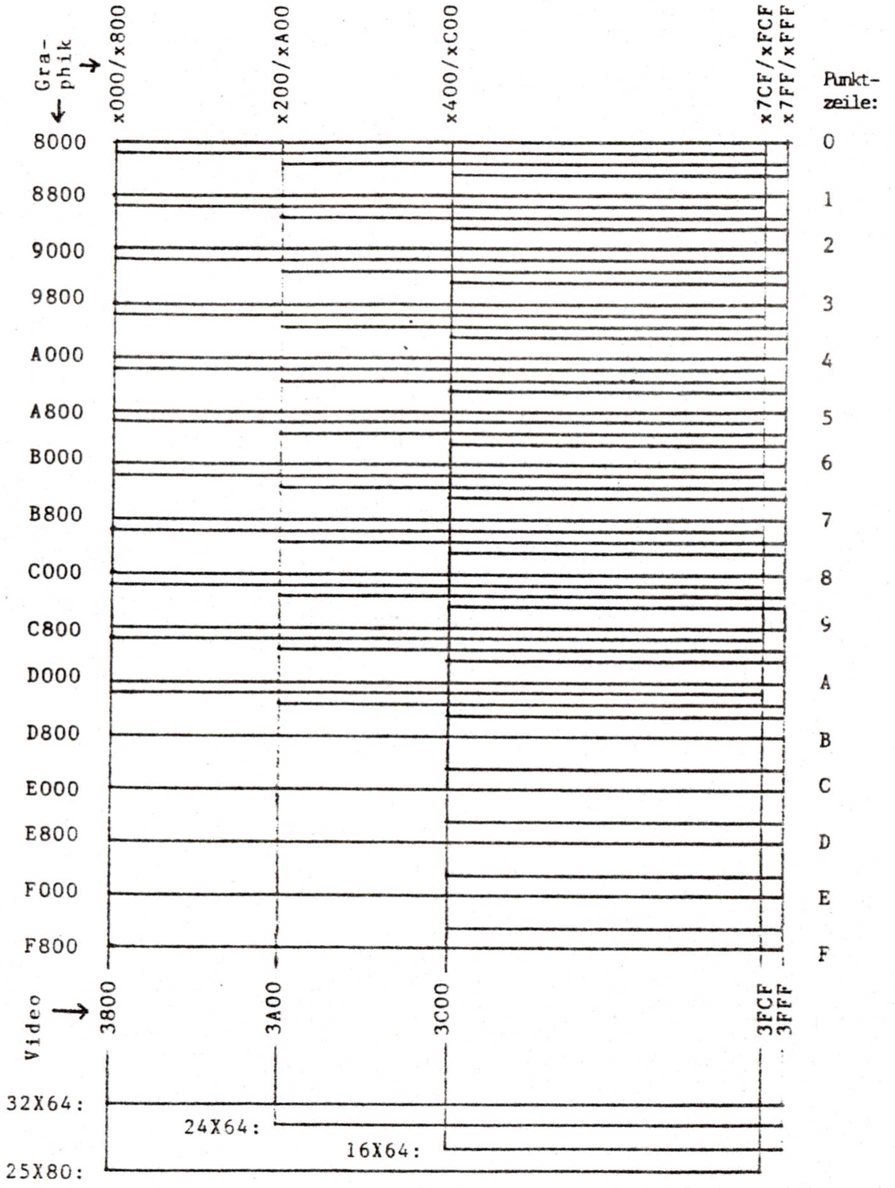
Jedes Graphikbyte hat selbstverständlich ebenfalls 8 Bits. Jedes Bit repräsentiert einen Punkt in der Waagerechten. Wie auch bei der HRG 1b, so kann man auch beim G3s durch Setzen, Löschen oder Prüfen einzelner Bits die BASIC-Entsprechungen SET, RESET und POINT durchführen. Auch zu diesem Punkt mag es genügen, auf die bisher erschienen Artikel zur HRG 1b zu verweisen. Dabei muß lediglich bedacht werden, daß alle acht Bits beim G3s zur Anzeige kommen.

Arnulf Sopp

```
5300                  ORG     5300h           ;darunter Platz für Stack
5300          stack   EQU     $               ;hier fängt er an
0002          spbuf   DS      2               ;Puffer für Stackpointer
5302 F3       start   DI                      ;nichts riskieren
5303 ED730053         LD      (spbuf),SP      ;Stackpointer retten
5307 310053           LD      SP,stack        ;eigener Stack f. Progr.
530A CD8553           CALL    change          ;Bildschirm markieren
530D 210080           LD      HL,8000h        ;Beginn des Graphiksp.
5310 110180           LD      DE,8001h        ;eine Stelle weiter
5313 01FF7F           LD      BC,7fffh        ;Zähler zum Löschen
5316 75               LD      (HL),L          ;1. Graphikstelle löschen
5317 EDB0             LDIR                    ;alle löschen
5319 11023C           LD      DE,3c02h        ;Bildschirmbereich
531C 3A8038   loop    LD      A,(3880h)       ;P-Tasten abfragen
531F E6F0             AND     0f0h            ;eine gedrückt?
5321 281A             JR      Z,ctrlkey       ;falls nein
5323 E5               PUSH    HL              ;Graphikzeiger retten
5324 D5               PUSH    DE              ;dto. Bildschirmzeiger
5325 210637           LD      HL,3706h        ;(Adr. Video-Par.-tab.)
5328 11F037           LD      DE,37f0h        ;dto. aktive Parameter
532B 0E20             LD      C,20h           ;16 Parameter
532D CDA353           CALL    pkey            ;Parameter-LSB ermitteln
5330 7D               LD      A,L             ;Parameter-LSB
5331 07               RLCA                    ;um 4 Bits rot. (*16)
5332 07               RLCA
5333 07               RLCA
5334 07               RLCA
5335 6F               LD      L,A             ;jetzt kompl. Adr. in HL
5336 EDB0             LDIR                    ;neue Parameter setzen
5338 D1               POP     DE              ;Register restaurieren
5339 E1               POP     HL
533A CD8553           CALL    change          ;Änderungen aktivieren
533D 3A4038   ctrlkey LD      A,(3840h)       ;Tastatur auslesen
5340 E6FC             AND     0fch            ;Pfeile, Blank oder BRK?
5342 28D8             JR      Z,loop          ;falls nichts dergleichen
5344 07               RLCA                    ;Blank?
5345 063F             LD      B,03fh          ;für Verzögerung
5347 3802             JR      C,goloop        ;falls Blank
5349 0607             LD      B,07h           ;sonst weniger Verzög.
534B 0EFF     goloop  LD      C,0ffh          ;Zähler BC komplettieren
534D D5               PUSH    DE              ;retten
534E 110004           LD      DE,0400h        ;in d. Gr.-splt. darunter
5351 3600             LD      (HL),00h        ;alten Strich löschen
5353 07               RLCA                    ;Rechtspfeil?
5354 DCAD53           CALL    C,right         ;falls ja
5357 07               RLCA                    ;Linkspfeil?
5358 DCAF53           CALL    C,left          ;falls ja
535B 07               RLCA                    ;Abwärtspfeil
535C DCB153           CALL    C,down          ;falls ja
535F 07               RLCA                    ;Aufwärtspfeil?
5360 DCB353           CALL    C,up            ;falls ja
5363 D1               POP     DE              ;restaurieren
5364 07               RLCA                    ;BREAK
5365 3812             JR      C,exit          ;falls ja
5367 CBFC             SET     7,H             ;nur 8000 - FFFF zulassen
5369 36FF             LD      (HL),0ffh       ;dort neuen Strich laden
536B EB               EX      DE,HL           ;austauschen
536C E5               PUSH    HL              ;wird verändert
536D CD6340           CALL    4063h           ;Stelle in Hex anzeigen
5370 E1               POP     HL              ;restaurieren
5371 EB               EX      DE,HL           ;zurücktauschen
5372 0B       delay   DEC     BC              ;Verzögerungszähler dekr.
5373 78               LD      A,B             ;nachsehen, ob
5374 B1               OR      C               ;Zähler abgelaufen
5375 20FB             JR      NZ,delay        ;falls noch nicht
5377 18A3             JR      loop            ;o. k., von vorne
5379 ED7B0053 exit    LD      SP,(spbuf)      ;Stackpointer restaur.
537D DBFA             IN      A,(0fah)        ;Systembyte 1 lesen
537F E6F5             AND     0f5h            ;Bits 7 und 5 wieder aus
5381 D3FA             OUT     (0fah),A        ;Graphik ausblenden
5383 FB               EI                      ;keine Gefahr mehr
5384 C9               RET                     ;ins Betriebssystem
5385 CDC901   change  CALL    01c9h           ;Änderungen aktivieren
5388 DBFA             IN      A,(0fah)        ;Systembyte 1 lesen
538A F61A             OR      1ah             ;Bits 1, 3 und 4 setzen
538C D3FA             OUT     (0fah),A        ;Graphik fitmachen
538E 3EBF             LD      A,0bfh          ;Grobgraphikblock
5390 32003B           LD      (3B00h),A       ;Anfang Großbildschirm
5393 32003C           LD      (3c00h),A       ;Anfang Kleinbildschirm
5396 32CF3F           LD      (3fcfh),A       ;Ende 80-Zeichen-Bildsch.
5399 32FF3F           LD      (3fffh),A       ;Ende sonstiger Bildsch.
539C DBFA             IN      A,(0fah)        ;Port wieder lesen
539E CBA7             RES     4,A             ;Tastatur wieder aktiv.
53A0 D3FA             OUT     (0fah),A        ;mit Systembyte 1
53A2 C9               RET                     ;oben weiter
53A3 07       pkey    RLCA                    ;P1-Taste? (80 X 25 Z.)
53A4 D8               RET     C               ;falls ja
53A5 2C               INC     L               ;Adr.-LSB d. Tab. erhöhen
53A6 07               RLCA                    ;P2-Taste? (64 X 16 Z.)
53A7 D8               RET     C               ;usw.
53A8 2C               INC     L
53A9 07               RLCA                    ;P3-Taste? (64 X 24 Z.)
53AA D8               RET     C
53AB 2C               INC     L
53AC C9               RET                     ;P4: 64 X 32 Z.
53AD 23       right   INC     HL              ;nächste Graphikstelle
53AE C9               RET                     ;oben weiter
53AF 2B       left    DEC     HL              ;vorige Graphikstelle
53B0 C9               RET
53B1 19       down    ADD     HL,DE           ;tiefer in selber Spalte
53B2 C9               RET
53B3 B7       up      OR      A               ;Cy löschen
53B4 ED52             SBC     HL,DE           ;höher in selber Spalte
53B6 C9               RET                     ;erledigt
5302                  END     start           ;dort Einsprung

00000 Fehler

change  5385    ctrlkey 533D    delay   5372    down    53B1    exit    5379
goloop  534B    left    53AF    loop    531C    pkey    53A3    right   53AD
spbuf   5300    stack   5300    start   5302    up      53B3
```

Tabelle der Graphik- und Bildschirmadressen des Genie III s:

Bildschirm-darstellung:	Bildsch.-adressen:	Zei-le:	Graphik-adressen:
25 X 80 Zeichen:	3800 - 3FCF	0	8000 - 87CF
		1	8800 - 8FCF
Beispiel für einige Zeilen:		2	9000 - 97CF
3800 - 384F <- 8000 - 804F		3	9800 - 9FCF
3850 - 389F <- 8050 - 809F		4	A000 - A7CF
.		5	A800 - AFCF
usw.		6	B000 - B7CF
.		7	B800 - BFCF
3F30 - 3F7F <- D730 - D77F		8	C000 - C7CF
3F80 - 3FCF <- D780 - D7CF		9	C800 - CFCF
		A	D000 - D7CF
16 X 64 Zeichen:	3C00 - 3FFF	0	8400 - 87FF
		1	8C00 - 8FFF
Beispiel für einige Zeilen:		2	9400 - 97FF
3C00 - 3C3F <- 8400 - 843F		3	9C00 - 9FFF
3C40 - 3C7F <- 8440 - 847F		4	A400 - A7FF
.		5	AC00 - AFFF
usw.		6	B400 - B7FF
.		7	BC00 - BFFF
3F80 - 3FBF <- F780 - F7BF		8	C400 - C7FF
3FC0 - 3FFF <- F7C0 - F7FF		9	CC00 - CFFF
		A	D400 - D7FF
		B	DC00 - DFFF
		C	E400 - E7FF
		D	EC00 - EFFF
		E	F400 - F7FF
24 X 64 Zeichen:	3A00 - 3FFF	0	8200 - 87FF
		1	8A00 - 8FFF
Beispiel für einige Zeilen:		2	9200 - 97FF
3A00 - 3A3F <- 8200 - 823F		3	9A00 - 9FFF
3A40 - 3A7F <- 8240 - 827F		4	A200 - A7FF
.		5	AA00 - AFFF
usw.		6	B200 - B7FF
.		7	BA00 - BFFF
3F80 - 3FBF <- CF80 - CFBF		8	C200 - C7FF
3FC0 - 3FFF <- CFC0 - CFFF		9	CA00 - CFFF
		A	D200 - D7FF
32 X 64 Zeichen:	3800 - 3FFF	0	8000 - 87FF
		1	8800 - 8FFF
Beispiel für einige Zeilen:		2	9000 - 97FF
3800 - 383F <- 8000 - 803F		3	9800 - 9FFF
3840 - 387F <- 8040 - 807F		4	A000 - A7FF
.		5	A800 - AFFF
usw.		6	B000 - B7FF
.		7	B800 - BFFF
3F80 - 3FBF <- FF80 - FFBF		8	C000 - C7FF
3FC0 - 3FFF <- FFC0 - FFFF		9	C800 - CFFF
		A	D000 - D7FF
		B	D800 - DFFF
		C	E000 - E7FF
		D	E800 - EFFF
		E	F000 - F7FF
		F	F800 - FFFF



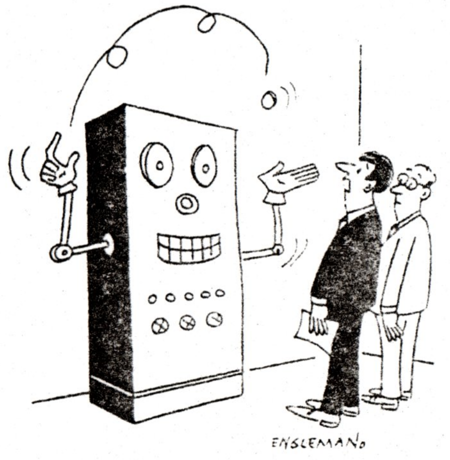
Korrespondierende Graphik- und Bildschirmadressen des Genie III s
bei unterschiedlichen Bildschirmformaten

»Die Welt ist überhaupt nur dadurch weitergekommen, daß irgend jemand die Courage gehabt hat, an Dinge zu rühren, von denen die Leute, in deren Interesse das lag, durch Jahrhunderte behauptet haben, daß man nicht an sie rühren darf.«

»Wenn man Intelligenz als die Fähigkeit definiert, neue Dinge zu lernen und Lösungen für Probleme zu finden, die das erste Mal auftauchen – wer ist dann intelligenter als das Kind?«

»Um die Wahrheit wahrscheinlicher zu machen, tut man gut daran, ein wenig Lüge hineinzumischen.«

»Am liebsten esse ich immer nur Pellkartoffeln. Allerdings gern mit einem Klacks Kaviar und einem Glas Champagner dazu.«



"I don't like the way this computer makes decisions!" © Creative Computing

Neue Zeichensätze für das Genie IIIs

Eine sehr erfreuliche Eigenschaft des G3s ist es, daß sein Zeichensatz soft vorliegt und von der Systemdiskette (oder sonstwoher) geladen wird. Zur Auswahl stehen ab Werk vier verschiedene: STD (Standard deutsch), HIRES (verlängert, um beim Bildschirmformat 32 X 64 Zeichen normal auszusehen), ASCII (amerikanisch) und Tandy (so ähnlich). Die beiden Letztgenannten haben für die Praxis hierzulande keine nennenswerte Bedeutung. Also ärgerte ich mich nur darüber, daß STD und HIRES ästhetisch und praktisch ihre Nachteile hatten, die jetzt behoben sind:

Die Codes 01-1A waren nur eine Wiederholung des Großbuchstaben-Alphabets 41-5A. Nun sind sie unterstrichen. Man kann damit nicht nur unterstrichene Texte ausgeben, sondern auch etwa unter CP/M einen Code wie CTRL-C als zweifelsfrei unterscheidbares Zeichen ausgeben lassen.

Überflüssigerweise gab es zwei Cursorzeichen (1F und 5F), obwohl beim G3s der Cursor vom Video-Controller hard erzeugt wird. Sie flogen raus und sind jetzt das Grad-Zeichen (1F) und der griechische Buchstabe x (5F).

Die übrigen Änderungen, von denen es noch ungefähr zwanzig gibt, haben nur rein geschmackliche Bedeutung. So sind z. B. die unmögliche Form der großen Umlaute, des p und des q mit viel zu kleinen Bäuchen usw. gestreamlined. Im Zeichensatz HIRES, der noch phantasieloser als STD war, sind die Modifikationen etwas zahlreicher.

In den Dumps im Anschluß an diesen Artikel sind die neuen Zeichensätze dargestellt (zunächst STD, dann HIRES). In der ersten Spalte steht der ASCII-Code, danach die einzelnen Bytes für die Matrixzeilen des Zeichens, am Ende ist das Zeichen selbst mit Hilfe der Drucker-HRG ausgeprintet. Es liegt auf der Seite, weil mein Drucker nun mal nur neun Nadeln hat. Zwischen je 16 Zeilen ist eine Leerzeile, um die Grenzen der Disksektoren zu markieren.

Die neuen Zeichen unterscheiden sich von den alten mitunter nur durch ein Bit. Man muß deshalb schon sehr genau hinsehen, um die Unterschiede festzustellen. Wer die Modifikationen durchführen will, hat beim Abtippen reichlich zu tun. Deshalb möchte ich anbieten, den Interessenten die fertigen Zeichensätze auf Diskette zuzuschicken. Sendet mir hierzu bitte einen selbstadressierten Freiumschlag mit einer Leerdiskette (oder formatiert a la G-DOS 2.4).

Arnulf Sopp

00 :	00	3C	42	B9	85	85	B9	42	3C	00	00	00	00	00	00	00	©
01 :	00	00	08	14	22	22	3E	22	22	00	FF	00	00	00	00	00	A
02 :	00	00	1E	22	22	1E	22	22	1E	00	FF	00	00	00	00	00	B
03 :	00	00	1C	22	02	02	02	22	1C	00	FF	00	00	00	00	00	C
04 :	00	00	1E	22	22	22	22	22	1E	00	FF	00	00	00	00	00	D
05 :	00	00	3E	02	02	0E	02	02	3E	00	FF	00	00	00	00	00	E
06 :	00	00	3E	02	02	0E	02	02	02	00	FF	00	00	00	00	00	F
07 :	00	00	1C	22	02	3A	22	22	1C	00	FF	00	00	00	00	00	G
08 :	00	00	22	22	22	3E	22	22	22	00	FF	00	00	00	00	00	H
09 :	00	00	1C	08	08	08	08	08	1C	00	FF	00	00	00	00	00	I
0A :	00	00	3C	20	20	20	20	22	1C	00	FF	00	00	00	00	00	J
0B :	00	00	22	12	0A	06	0A	12	22	00	FF	00	00	00	00	00	K
0C :	00	00	02	02	02	02	02	02	3E	00	FF	00	00	00	00	00	L
0D :	00	00	22	36	2A	2A	22	22	22	00	FF	00	00	00	00	00	M
0E :	00	00	22	22	26	2A	32	22	22	00	FF	00	00	00	00	00	N
0F :	00	00	1C	22	22	22	22	22	1C	00	FF	00	00	00	00	00	O
10 :	00	00	1E	22	22	1E	02	02	02	00	FF	00	00	00	00	00	P
11 :	00	00	1C	22	22	22	2A	12	2C	00	FF	00	00	00	00	00	Q
12 :	00	00	1E	22	22	1E	0A	12	22	00	FF	00	00	00	00	00	R
13 :	00	00	1C	22	02	1C	20	22	1C	00	FF	00	00	00	00	00	S
14 :	00	00	7F	08	08	08	08	08	08	00	FF	00	00	00	00	00	T
15 :	00	00	22	22	22	22	22	22	1C	00	FF	00	00	00	00	00	U
16 :	00	00	22	22	22	22	22	14	08	00	FF	00	00	00	00	00	V
17 :	00	00	22	22	22	2A	2A	36	22	00	FF	00	00	00	00	00	W
18 :	00	00	22	22	14	08	14	22	22	00	FF	00	00	00	00	00	X
19 :	00	00	22	22	14	08	08	08	08	00	FF	00	00	00	00	00	Y
1A :	00	00	3E	20	10	08	04	02	3E	00	FF	00	00	00	00	00	Z
1B :	00	00	08	1C	2A	08	08	08	08	00	00	00	00	00	00	00	←
1C :	00	00	08	08	08	08	2A	1C	08	00	00	00	00	00	00	00	→
1D :	00	00	00	08	04	3E	04	08	00	00	00	00	00	00	00	00	↓
1E :	00	00	00	08	10	3E	10	08	00	00	00	00	00	00	00	00	↑
1F :	00	00	18	24	18	00	00	00	00	00	00	00	00	00	00	00	°
20 :	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
21 :	00	00	08	08	08	08	08	00	08	00	00	00	00	00	00	00	!
22 :	00	00	14	14	14	00	00	00	00	00	00	00	00	00	00	00	"
23 :	00	00	14	14	3E	14	3E	14	14	00	00	00	00	00	00	00	#
24 :	00	00	08	3C	0A	1C	28	1E	08	00	00	00	00	00	00	00	$
25 :	00	00	46	26	10	08	04	62	61	00	00	00	00	00	00	00	%
26 :	00	00	04	0A	0A	04	2A	1A	24	00	00	00	00	00	00	00	&
27 :	00	00	18	18	08	04	00	00	00	00	00	00	00	00	00	00	'
28 :	00	00	10	08	04	04	04	08	10	00	00	00	00	00	00	00	(
29 :	00	00	08	10	20	20	20	10	08	00	00	00	00	00	00	00	)
2A :	00	00	00	08	2A	1C	2A	08	00	00	00	00	00	00	00	00	*
2B :	00	00	00	08	08	7F	08	08	00	00	00	00	00	00	00	00	+
2C :	00	00	00	00	00	00	00	18	18	08	04	00	00	00	00	00	,
2D :	00	00	00	00	00	3E	00	00	00	00	00	00	00	00	00	00	-
2E :	00	00	00	00	00	00	00	0C	0C	00	00	00	00	00	00	00	.
2F :	00	00	40	20	10	08	04	02	01	00	00	00	00	00	00	00	/
30 :	00	00	1C	22	32	2A	26	22	1C	00	00	00	00	00	00	00	0
31 :	00	00	08	0C	0A	08	08	08	1C	00	00	00	00	00	00	00	1
32 :	00	00	1C	22	20	10	08	04	3E	00	00	00	00	00	00	00	2
33 :	00	00	1C	22	20	18	20	22	1C	00	00	00	00	00	00	00	3
34 :	00	00	10	18	14	12	3E	10	10	00	00	00	00	00	00	00	4
35 :	00	00	3E	02	1E	20	20	22	1C	00	00	00	00	00	00	00	5
36 :	00	00	18	04	02	1E	22	22	1C	00	00	00	00	00	00	00	6
37 :	00	00	3E	20	20	10	08	08	08	00	00	00	00	00	00	00	7
38 :	00	00	1C	22	22	1C	22	22	1C	00	00	00	00	00	00	00	8
39 :	00	00	1C	22	22	3C	20	20	1C	00	00	00	00	00	00	00	9
3A :	00	00	00	00	18	18	00	18	18	00	00	00	00	00	00	00	:
3B :	00	00	00	00	18	18	00	18	18	08	04	00	00	00	00	00	;
3C :	00	00	10	08	04	02	04	08	10	00	00	00	00	00	00	00	<
3D :	00	00	00	00	3E	00	3E	00	00	00	00	00	00	00	00	00	=
3E :	00	00	04	08	10	20	10	08	04	00	00	00	00	00	00	00	>
3F :	00	00	1C	22	20	10	08	00	08	00	00	00	00	00	00	00	?

40 :	00	00	1C	22	20	2C	2A	2A	1C	00	00	00	00	00	00	00	@
41 :	00	00	08	14	22	22	3E	22	22	00	00	00	00	00	00	00	A
42 :	00	00	1E	22	22	1E	22	22	1E	00	00	00	00	00	00	00	B
43 :	00	00	1C	22	02	02	02	22	1C	00	00	00	00	00	00	00	C
44 :	00	00	1E	22	22	22	22	22	1E	00	00	00	00	00	00	00	D
45 :	00	00	3E	02	02	0E	02	02	3E	00	00	00	00	00	00	00	E
46 :	00	00	3E	02	02	0E	02	02	02	00	00	00	00	00	00	00	F
47 :	00	00	1C	22	02	3A	22	22	1C	00	00	00	00	00	00	00	G
48 :	00	00	22	22	22	3E	22	22	22	00	00	00	00	00	00	00	H
49 :	00	00	1C	08	08	08	08	08	1C	00	00	00	00	00	00	00	I
4A :	00	00	3C	20	20	20	20	22	1C	00	00	00	00	00	00	00	J
4B :	00	00	22	12	0A	06	0A	12	22	00	00	00	00	00	00	00	K
4C :	00	00	02	02	02	02	02	02	3E	00	00	00	00	00	00	00	L
4D :	00	00	22	36	2A	2A	22	22	22	00	00	00	00	00	00	00	M
4E :	00	00	22	22	26	2A	32	22	22	00	00	00	00	00	00	00	N
4F :	00	00	1C	22	22	22	22	22	1C	00	00	00	00	00	00	00	O
50 :	00	00	1E	22	22	1E	02	02	02	00	00	00	00	00	00	00	P
51 :	00	00	1C	22	22	22	2A	12	2C	00	00	00	00	00	00	00	Q
52 :	00	00	1E	22	22	1E	0A	12	22	00	00	00	00	00	00	00	R
53 :	00	00	1C	22	02	1C	20	22	1C	00	00	00	00	00	00	00	S
54 :	00	00	7F	08	08	08	08	08	08	00	00	00	00	00	00	00	T
55 :	00	00	22	22	22	22	22	22	1C	00	00	00	00	00	00	00	U
56 :	00	00	22	22	22	22	22	14	08	00	00	00	00	00	00	00	V
57 :	00	00	22	22	22	2A	2A	36	22	00	00	00	00	00	00	00	W
58 :	00	00	22	22	14	08	14	22	22	00	00	00	00	00	00	00	X
59 :	00	00	22	22	14	08	08	08	08	00	00	00	00	00	00	00	Y
5A :	00	00	3E	20	10	08	04	02	3E	00	00	00	00	00	00	00	Z
5B :	00	22	08	14	22	22	3E	22	22	00	00	00	00	00	00	00	Ä
5C :	22	00	1C	22	22	22	22	22	1C	00	00	00	00	00	00	00	Ö
5D :	22	00	22	22	22	22	22	22	1C	00	00	00	00	00	00	00	Ü
5E :	00	00	08	14	22	00	00	00	00	00	00	00	00	00	00	00	^
5F :	00	00	00	40	3E	25	24	24	24	00	00	00	00	00	00	00	π
60 :	00	1C	22	02	1C	22	1C	20	22	1C	00	00	00	00	00	00	§
61 :	00	00	00	00	1C	20	3C	22	3C	00	00	00	00	00	00	00	a
62 :	00	00	02	02	1A	26	22	26	1A	00	00	00	00	00	00	00	b
63 :	00	00	00	00	1C	22	02	22	1C	00	00	00	00	00	00	00	c
64 :	00	00	20	20	2C	32	22	32	2C	00	00	00	00	00	00	00	d
65 :	00	00	00	00	1C	22	3E	02	1C	00	00	00	00	00	00	00	e
66 :	00	00	18	24	04	0E	04	04	04	00	00	00	00	00	00	00	f
67 :	00	00	00	00	1C	22	22	22	3C	20	1E	00	00	00	00	00	g
68 :	00	00	02	02	1A	26	22	22	22	00	00	00	00	00	00	00	h
69 :	00	00	0C	00	0C	08	08	08	1C	00	00	00	00	00	00	00	i
6A :	00	00	30	00	38	20	20	20	20	22	1C	00	00	00	00	00	j
6B :	00	00	02	02	22	12	0E	12	22	00	00	00	00	00	00	00	k
6C :	00	00	0C	08	08	08	08	08	1C	00	00	00	00	00	00	00	l
6D :	00	00	00	00	16	2A	2A	2A	2A	00	00	00	00	00	00	00	m
6E :	00	00	00	00	1A	26	22	22	22	00	00	00	00	00	00	00	n
6F :	00	00	00	00	1C	22	22	22	1C	00	00	00	00	00	00	00	o
70 :	00	00	00	00	1A	26	22	26	1A	02	02	00	00	00	00	00	p
71 :	00	00	00	00	2C	32	22	32	2C	20	20	00	00	00	00	00	q
72 :	00	00	00	00	1A	26	02	02	02	00	00	00	00	00	00	00	r
73 :	00	00	00	00	3C	02	1C	20	1E	00	00	00	00	00	00	00	s
74 :	00	00	04	04	0E	04	04	24	18	00	00	00	00	00	00	00	t
75 :	00	00	00	00	22	22	22	32	2C	00	00	00	00	00	00	00	u
76 :	00	00	00	00	22	22	22	14	08	00	00	00	00	00	00	00	v
77 :	00	00	00	00	22	22	2A	2A	14	00	00	00	00	00	00	00	w
78 :	00	00	00	00	22	14	08	14	22	00	00	00	00	00	00	00	x
79 :	00	00	00	00	22	22	22	22	3C	20	1E	00	00	00	00	00	y
7A :	00	00	00	00	3E	10	08	04	3E	00	00	00	00	00	00	00	z
7B :	00	00	22	00	1C	20	3C	22	3C	00	00	00	00	00	00	00	ä
7C :	00	00	22	00	1C	22	22	22	1C	00	00	00	00	00	00	00	ö
7D :	00	00	22	00	22	22	22	32	2C	00	00	00	00	00	00	00	ü
7E :	00	00	1C	22	22	3A	42	42	32	02	02	00	00	00	00	00	ß
7F :	00	00	55	AA	55	AA	55	AA	55	00	00	00	00	00	00	00	▒

00 :	00	3C	42	99	A5	85	85	95	85	A5	99	42	3C	00	00	00
01 :	00	08	08	14	14	22	22	3E	22	22	22	22	22	00	FF	00
02 :	00	1E	24	24	24	24	1C	24	24	24	24	24	1E	00	FF	00
03 :	00	1C	22	02	02	02	02	02	02	02	02	22	1C	00	FF	00
04 :	00	1E	24	24	24	24	24	24	24	24	24	24	1E	00	FF	00
05 :	00	3E	02	02	02	02	0E	02	02	02	02	02	3E	00	FF	00
06 :	00	3E	02	02	02	02	0E	02	02	02	02	02	02	00	FF	00
07 :	00	1C	22	22	02	02	32	22	22	22	22	22	1C	00	FF	00
08 :	00	22	22	22	22	22	3E	22	22	22	22	22	22	00	FF	00
09 :	00	3E	08	08	08	08	08	08	08	08	08	08	3E	00	FF	00
0A :	00	3E	20	20	20	20	20	20	20	20	20	22	1C	00	FF	00
0B :	00	22	22	12	12	0A	06	06	0A	12	12	22	22	00	FF	00
0C :	00	02	02	02	02	02	02	02	02	02	02	02	3E	00	FF	00
0D :	00	22	36	36	2A	2A	22	22	22	22	22	22	22	00	FF	00
0E :	00	22	26	26	2A	2A	2A	2A	32	32	22	22	22	00	FF	00
0F :	00	1C	22	22	22	22	22	22	22	22	22	22	1C	00	FF	00
10 :	00	1E	22	22	22	22	22	1E	02	02	02	02	02	00	FF	00
11 :	00	1C	22	22	22	22	22	22	22	2A	1A	12	2C	00	FF	00
12 :	00	1E	22	22	22	22	1E	06	0A	0A	12	22	22	00	FF	00
13 :	00	1C	22	22	02	02	1C	20	20	20	22	22	1C	00	FF	00
14 :	00	3E	08	08	08	08	08	08	08	08	08	08	08	00	FF	00
15 :	00	22	22	22	22	22	22	22	22	22	22	22	1C	00	FF	00
16 :	00	22	22	22	22	22	22	22	22	14	14	08	08	00	FF	00
17 :	00	22	22	22	22	22	22	2A	2A	36	36	22	22	00	FF	00
18 :	00	22	22	22	14	14	08	08	14	14	22	22	22	00	FF	00
19 :	00	22	22	22	22	14	14	08	08	08	08	08	08	00	FF	00
1A :	00	3E	20	20	10	10	08	08	04	04	02	02	3E	00	FF	00
1B :	00	08	08	1C	1C	2A	2A	08	08	08	08	08	08	00	00	00
1C :	00	08	08	08	08	08	08	2A	2A	1C	1C	08	08	00	00	00
1D :	00	00	08	08	04	04	3E	04	04	08	08	00	00	00	00	00
1E :	00	00	08	08	10	10	3E	10	10	08	08	00	00	00	00	00
1F :	00	18	24	24	24	18	00	00	00	00	00	00	00	00	00	00
20 :	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
21 :	00	08	08	08	08	08	08	08	08	00	00	08	08	00	00	00
22 :	00	14	14	14	14	00	00	00	00	00	00	00	00	00	00	00
23 :	00	14	14	14	3E	14	14	14	14	3E	14	14	14	00	00	00
24 :	00	08	3C	0A	0A	0A	1C	28	28	28	1E	08	08	00	00	00
25 :	00	04	2A	24	10	10	08	08	04	04	12	2A	10	00	00	00
26 :	00	0C	12	12	12	12	0C	0C	4A	52	32	52	4C	00	00	00
27 :	00	18	18	18	08	04	00	00	00	00	00	00	00	00	00	00
28 :	00	40	20	20	10	10	10	10	10	10	20	20	40	00	00	00
29 :	00	02	04	04	08	08	08	08	08	08	04	04	02	00	00	00
2A :	00	00	00	08	2A	2A	1C	1C	2A	2A	08	00	00	00	00	00
2B :	00	00	08	08	08	08	7F	08	08	08	08	00	00	00	00	00
2C :	00	00	00	00	00	00	00	00	00	00	18	18	08	04	00	00
2D :	00	00	00	00	00	00	7E	00	00	00	00	00	00	00	00	00
2E :	00	00	00	00	00	00	00	00	00	00	00	0C	0C	00	00	00
2F :	00	20	20	10	10	08	08	08	04	04	02	02	00	00	00	00
30 :	00	1C	22	32	32	2A	2A	2A	2A	26	26	22	1C	00	00	00
31 :	00	08	0C	0C	0A	0A	08	08	08	08	08	08	3E	00	00	00
32 :	00	1C	22	22	10	10	08	08	04	04	02	02	3E	00	00	00
33 :	00	1C	22	22	20	10	0C	10	20	20	22	22	1C	00	00	00
34 :	00	10	18	18	14	14	12	12	7E	10	10	10	10	00	00	00
35 :	00	3E	02	02	02	02	0E	10	20	20	22	22	1C	00	00	00
36 :	00	18	04	04	02	02	1E	22	22	22	22	22	1C	00	00	00
37 :	00	3E	20	20	10	10	10	08	08	08	04	04	04	00	00	00
38 :	00	1C	22	22	22	14	1C	22	22	22	22	22	1C	00	00	00
39 :	00	1C	22	22	22	22	3C	20	20	20	20	22	1C	00	00	00
3A :	00	00	00	00	0C	0C	00	00	00	00	00	0C	0C	00	00	00
3B :	00	00	00	00	18	18	00	00	00	00	18	18	08	04	00	00
3C :	00	10	08	08	04	04	02	02	04	04	08	08	10	00	00	00
3D :	00	00	00	00	7E	00	00	00	7E	00	00	00	00	00	00	00
3E :	00	08	10	10	20	20	40	40	20	20	10	10	08	00	00	00
3F :	00	1C	22	22	20	10	10	08	08	00	00	08	08	00	00	00

40 :	00	38	44	44	40	40	58	54	54	54	54	54	38	00	00	00
41 :	00	08	08	14	14	22	22	3E	22	22	22	22	22	00	00	00
42 :	00	1E	24	24	24	24	1C	24	24	24	24	24	1E	00	00	00
43 :	00	1C	22	02	02	02	02	02	02	02	02	22	1C	00	00	00
44 :	00	1E	24	24	24	24	24	24	24	24	24	24	1E	00	00	00
45 :	00	3E	02	02	02	02	0E	02	02	02	02	02	3E	00	00	00
46 :	00	3E	02	02	02	02	0E	02	02	02	02	02	02	00	00	00
47 :	00	1C	22	22	02	02	32	22	22	22	22	22	1C	00	00	00
48 :	00	22	22	22	22	22	3E	22	22	22	22	22	22	00	00	00
49 :	00	3E	08	08	08	08	08	08	08	08	08	08	3E	00	00	00
4A :	00	3E	20	20	20	20	20	20	20	20	20	22	1C	00	00	00
4B :	00	22	22	12	12	0A	06	06	0A	12	12	22	22	00	00	00
4C :	00	02	02	02	02	02	02	02	02	02	02	02	3E	00	00	00
4D :	00	22	36	36	2A	2A	22	22	22	22	22	22	22	00	00	00
4E :	00	22	26	26	2A	2A	2A	2A	32	32	22	22	22	00	00	00
4F :	00	1C	22	22	22	22	22	22	22	22	22	22	1C	00	00	00
50 :	00	1E	22	22	22	22	22	1E	02	02	02	02	02	00	00	00
51 :	00	1C	22	22	22	22	22	22	22	2A	1A	12	2C	00	00	00
52 :	00	1E	22	22	22	22	1E	06	0A	0A	12	22	22	00	00	00
53 :	00	1C	22	22	02	02	1C	20	20	20	22	22	1C	00	00	00
54 :	00	3E	08	08	08	08	08	08	08	08	08	08	08	00	00	00
55 :	00	22	22	22	22	22	22	22	22	22	22	22	1C	00	00	00
56 :	00	22	22	22	22	22	22	22	22	14	14	08	08	00	00	00
57 :	00	22	22	22	22	22	22	2A	2A	36	36	22	22	00	00	00
58 :	00	22	22	22	14	14	08	08	14	14	22	22	22	00	00	00
59 :	00	22	22	22	22	14	14	08	08	08	08	08	08	00	00	00
5A :	00	3E	20	20	10	10	08	08	04	04	02	02	3E	00	00	00
5B :	22	22	08	14	14	22	22	22	3E	22	22	22	22	00	00	00
5C :	22	22	00	1C	22	22	22	22	22	22	22	22	1C	00	00	00
5D :	22	22	00	22	22	22	22	22	22	22	22	22	1C	00	00	00
5E :	00	08	14	22	00	00	00	00	00	00	00	00	00	00	00	00
5F :	00	00	00	00	40	7C	26	24	24	24	24	24	24	00	00	00
60 :	00	1C	22	22	02	1C	22	22	1C	20	22	22	1C	00	00	00
61 :	00	00	00	00	00	1C	20	20	3C	22	22	22	3C	00	00	00
62 :	00	02	02	02	02	1A	26	22	22	22	22	26	1A	00	00	00
63 :	00	00	00	00	00	1C	22	02	02	02	02	22	1C	00	00	00
64 :	00	20	20	20	20	2C	32	22	22	22	22	32	2C	00	00	00
65 :	00	00	00	00	00	1C	22	22	3E	02	02	22	1C	00	00	00
66 :	00	18	24	24	04	04	0E	04	04	04	04	04	04	00	00	00
67 :	00	00	00	00	00	2C	32	22	22	22	32	2C	20	20	1C	00
68 :	00	02	02	02	02	1A	26	22	22	22	22	22	22	00	00	00
69 :	00	08	08	00	00	0E	08	08	08	08	08	08	3E	00	00	00
6A :	00	10	10	00	00	1C	10	10	10	10	10	10	10	10	0E	00
6B :	00	02	02	02	02	22	22	12	0E	12	12	22	22	00	00	00
6C :	00	0E	08	08	08	08	08	08	08	08	08	08	3E	00	00	00
6D :	00	00	00	00	00	14	14	2A	2A	2A	2A	2A	2A	00	00	00
6E :	00	00	00	00	00	1A	26	22	22	22	22	22	22	00	00	00
6F :	00	00	00	00	00	1C	22	22	22	22	22	22	1C	00	00	00
70 :	00	00	00	00	00	1A	26	22	22	22	26	1A	02	02	02	00
71 :	00	00	00	00	00	2C	32	22	22	22	32	2C	20	20	20	00
72 :	00	00	00	00	00	1A	26	22	02	02	02	02	02	00	00	00
73 :	00	00	00	00	00	3C	02	02	1C	20	20	20	1E	00	00	00
74 :	00	04	04	04	04	0E	04	04	04	04	04	24	18	00	00	00
75 :	00	00	00	00	00	22	22	22	22	22	22	32	2C	00	00	00
76 :	00	00	00	00	00	22	22	22	22	14	14	08	08	00	00	00
77 :	00	00	00	00	00	22	22	22	2A	2A	2A	14	14	00	00	00
78 :	00	00	00	00	00	22	22	14	08	14	14	22	22	00	00	00
79 :	00	00	00	00	00	22	22	22	22	22	22	3C	20	20	1C	00
7A :	00	00	00	00	00	3E	20	10	08	08	04	02	3E	00	00	00
7B :	00	24	24	00	00	1C	20	20	3C	22	22	22	3C	00	00	00
7C :	00	22	22	00	00	1C	22	22	22	22	22	22	1C	00	00	00
7D :	00	22	22	00	00	22	22	22	22	22	22	32	2C	00	00	00
7E :	00	1C	22	22	22	12	0A	12	22	22	22	1A	02	02	00	00
7F :	00	55	55	AA	AA	55	55	AA	AA	55	55	AA	AA	00	00	00

Call mit BASCOM

Vor kurzem schrieb ich ein BASIC-Programm zur rationellen Erstellung von ILF-Files und zum sicheren Kopieren von "großen" (z.B. 80/DS/DD) auf "kleine" (z.B. 40/SS/DD) Disketten. Das Programm hier in seinem genauen Ablauf darzulegen ist nicht meine Absicht. Ihr bekommt es in der Programmbibliothek bei Kajott und könnt es euch auseinanderpfriemeln oder einfach nur benutzen! Es geht mir vielmehr um ein Problem, das beim Kompilieren des Programms mit dem BASIC-Compiler BASCOM von Microsoft auftrat.

BASCOM kontra ZBASIC
BASCOM erzeugt, genau wie der ZBASIC-Compiler, direkt aufrufbare /CMD-Files. Er hat gegenüber ZBASIC jedoch den Vorteil, sich in seinem Syntax sehr eng an das Disk-BASIC anzulehnen und so z. B. auch sehr komplizierte Stringoperationen übersetzen zu können. Dafür kann er, im Gegensatz zu ZBASIC, mit der Funktion CMD"S=DOSCMD" nichts anfangen.
Da mein Programm fast nur von Stringoperationen (MID$,LEN$ usw.) lebt, aber nur ein CMD"S=DOSCMD" enthält, wurde mir die Wahl des Compilers leichtgemacht. BASCOM musste es sein!
Aber wie schon erwähnt, BASCOM kann mit dem Befehl CMD"S=DOSCMD" nichts anfangen. Es mußte also ein anderer Weg gefunden werden, z. B. ein COPY direkt von einem compilierten Programm aus zu starten!

Die Rettung: CALL
BASCOM unterstützt neben den Disk-BASIC-Funktionen des TRSDOS zusätzlich Befehle, die normalerweise nicht verwendbar sind. Unter anderem gibt es da den Befehl
CALL Routine (Par,Par,Par).
Mit diesem Befehl lassen sich Maschinenroutinen von einem BASIC-Programm aufrufen. Im BASCOM Handbuch steht dazu folgende, frei ins deutsche übersetzte, kurze Anmerkung:

Der CALL-Befehl erlaubt es, ein Maschinensprache- oder TRS80-Fortran- Unterprogramm aufzurufen.

Das Format des CALL-Befehls lautet:
CALL <variable name> [(argument list)]
wobei <variable name> und <argument list> vom Benutzer bestimmt werden können.

<variable name> ist der Name des Unterprogramms, welches aufgerufen werden soll. Dieser Name muß zwischen einem und sechs Zeichen lang sein (mein BASCOM verkraftet nur zwei Zeichen!!!) und muß von LINK 80 (L80/CMD) als "Global Symbol" erkannt werden! (<variable name> muß entweder der Name eines Unterprogramms in einem FORTRAN SUBROUTINE Statement oder ein PUBLIC Symbol in einer Maschinenspracheroutine sein.)

<argument list> ist optional und enthält die Argumente, die einer Maschinen- oder Fortran- Subroutine übergeben werden sollen.
Beispiel: 120 CALL MYROUT (I,J,K)

Das ist alles, was mein BASCOM-Handbuch zu dieser Möglichkeit der Einbindung von Maschinenroutinen in BASIC-Programme sagt. Ein praktisches Beispiel zur Nutzung des CALL-Befehls ist nirgends vorhanden und so ist diese wertvolle Eigenschaft von BASCOM praktisch nutzlos.

Gänzlich nutzlos ist er nicht, mit einem kleinen Trick kann man ihn schon benutzen. Wie man das bewerkstelligt, will ich euch kurz aufzeigen!

Das folgende, kurze BASIC-Programm soll nichts weiter tun, als den Bildschirm zu löschen, einen kurzen Text auszugeben und dann den freien Speicherplatz auf der Diskette anzuzeigen.

```
10 CLS
20 PRINT"DIESES PROGRAMM DEMONSTRIERT DEN AUFRUF DES
30 PRINT"DOS-KOMMANDOS -FREE- VON EINEM MIT BASCOM
40 PRINT"KOMPILIERTEN BASIC-PROGRAMM AUS!
50 PRINT:PRINT
60 CALL FR
70 END
```

Die folgenden Zeilen sind ein "Livemitschnitt" (Bildschirmausgaben in ein Diskfile geroutet!) eines Kompilerlaufs von BASCOM:

```
DOS> BASCOM TEST=TEST

00000 FATAL ERROR(S)
13040 BYTES FREE

DOS> L80 TEST,TEST-N-E
DATA     5200     75AA
FR*      5256
(5211    75AA)
```

Wie man sieht, tritt beim Kompilierungsvorgang keinerlei Fehler auf. Einzige "abnormale" Anzeige ist die Zeile:
FR* 5256
Hier wird angezeigt, daß an dieser Stelle im erzeugten /CMD-File der Aufruf eines, dem Linker nicht bekannten Unterprogramms steht. Wie man im untenstehenden Hexdump des erzeugten /CMD-Programms deutlich sehen kann, trägt der Linker an der angezeigten Stelle (5256H) als Adresse für die unbekannte Routine den Wert 0000H ein. Dies führt natürlich beim Lauf des Programms unvermeidbar zum Absturz!

```
TEST/CMD Drive 1 Record 0 X'0000'

0123456789ABCDEF BYTE 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
======================================================================
...R............ <00> 01 02 00 52 00 00 00 00 00 00 00 00 00 00 00 00
.....!%R..S.R... <10> 00 00 00 00 00 21 25 52 C3 19 53 DE 52 00 00 E0
R.R.R...R.hT..Ä! <20> 52 00 52 07 52 00 00 11 52 CD 68 54 CD BC 5B 21
.R.gZ..Ä!.R.gZ.. <30> AD 52 CD 67 5A CD BC 5B 21 81 52 CD 67 5A CD BC
Ä!.R.gZ..Ä!.R.gZ <40> 5B 21 5E 52 CD 67 5A CD BC 5B 21 DB 52 CD 67 5A
..Ä!.R.gZ.....V. <50> CD BC 5B 21 DB 52 CD 67 5A CD 00 00 CD 8E 56 CD
.V aRKOMPILIERTE <60> 8E 56 20 61 52 4B 4F 4D 50 49 4C 49 45 52 54 45
N BASIC-PROGRAMM <70> 4E 20 42 41 53 49 43 2D 50 52 4F 47 52 41 4D 4D
 AUS!).RDOS-KOMM <80> 20 41 55 53 21 29 84 52 44 4F 53 2D 4B 4F 4D 4D
ANDOS -FREE- VON <90> 41 4E 44 4F 53 20 2D 46 52 45 45 2D 20 56 4F 4E
 EINEM MIT BASCO <A0> 20 45 49 4E 45 4D 20 4D 49 54 20 42 41 53 43 4F
M+.RDIESES PROGR <B0> 4D 2B B0 52 44 49 45 53 45 53 20 50 52 4F 47 52
AMM DEMONSTRIERT <C0> 41 4D 4D 20 44 45 4D 4F 4E 53 54 52 49 45 52 54
 DEN AUFRUF DES. <D0> 20 44 45 4E 20 41 55 46 52 55 46 20 44 45 53 00
.R...ÄÄÄÄ BASLIB <E0> DE 52 00 00 00 5B 5B 5B 5B 20 42 41 53 4C 49 42
 #000000 5.2 - 0 <F0> 20 23 30 30 30 30 30 30 20 35 2E 32 20 2D 20 4F
```

Um den CALL-Befehl doch noch nutzen zu können, bin ich auf folgenden Trick gekommen:

1. Man sucht sich im erzeugten /CMD-Programm eine vom Programm selbst nicht benutzte Stelle, die für das gewünschte Unterprogramm groß genug ist. Eine solche Stelle kann man sich selbst dadurch schaffen, daß man eine Dummyvariable mit soviel Text auffüllt, wie man für sein Unterprogramm braucht.

2. Nach dem Kompilierungsvorgang wird die Maschinenroutine dann in diese Variable gezapt und der ursprüngliche CALL nach 0000H in einen CALL auf die eingezapte Unterroutine umgeändert.

In unserem Beispiel habe ich keine Dummyvariable sondern die Copyrightmeldung von BASCOM für die Unterroutine ausgenutzt. Das kurze Maschinenprogramm sieht folgendermaßen aus:

```
21F752      LD      HL,FREE    'HL-Register auf den DOS-Befehl
CD1944      CALL    4419H      'DOSCAL aufrufen
C9          RET                'zurück zum Hauptprogramm
0000        DEFW    0000H      'nur Füllung, keine Funktion
0000        DEFW    0000H      ' "     "       "       "
46524545    DEFM    'FREE'     'der auszuführende DOS-Befehl
0D          DEFB    0DH        'nicht vergessen: OD (CR) als
                               'Endzeichen!
```

Die durchzuführenden Änderungen sind im untenstehenden Hexdump unterstrichen!

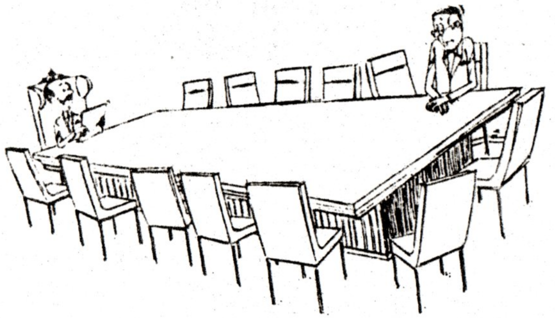
```
TEST/CMD Drive 1 Record 0 X'0000'

0123456789ABCDEF BYTE 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
=====================================================================
...R............ <00> 01 02 00 52 00 00 00 00 00 00 00 00 00 00 00 00
.....!%R..S.R... <10> 00 00 00 00 00 21 25 52 C3 19 53 DE 52 00 00 E0
R.R.R...R.hT..X! <20> 52 00 52 07 52 00 00 11 52 CD 68 54 CD BC 5B 21
.R.gZ..X!.R.gZ.. <30> AD 52 CD 67 5A CD BC 5B 21 81 52 CD 67 5A CD BC
X!.R.gZ..X!.R.gZ <40> 5B 21 5E 52 CD 67 5A CD BC 5B 21 DB 52 CD 67 5A
..X!.R.gZ..R..V. <50> CD BC 5B 21 DB 52 CD 67 5A CD EC 52 CD 8E 56 CD
.V aRKOMPILIERTE <60> 8E 56 20 61 52 4B 4F 4D 50 49 4C 49 45 52 54 45
N BASIC-PROGRAMM <70> 4E 20 42 41 53 49 43 2D 50 52 4F 47 52 41 4D 4D
 AUS!).RDOS-KOMM <80> 20 41 55 53 21 29 84 52 44 4F 53 2D 4B 4F 4D 4D
ANDOS -FREE- VON <90> 41 4E 44 4F 53 20 2D 46 52 45 45 2D 20 56 4F 4E
 EINEM MIT BASCO <A0> 20 45 49 4E 45 4D 20 4D 49 54 20 42 41 53 43 4F
M+.RDIESES PROGR <B0> 4D 2B B0 52 44 49 45 53 45 53 20 50 52 4F 47 52
AMM DEMONSTRIERT <C0> 41 4D 4D 20 44 45 4D 4F 4E 53 54 52 49 45 52 54
 DEN AUFRUF DES. <D0> 20 44 45 4E 20 41 55 46 52 55 46 20 44 45 53 00
.R...XXXX BASLIB <E0> DE 52 00 00 00 5B 5B 5B 5B 20 42 41 53 4C 49 42
!.R..D.....FREE. <F0> 21 F7 52 CD 19 44 C9 00 00 00 00 46 52 45 45 0D
```

Dieses kurze Beispiel kann natürlich nur das Prinzip darlegen. Eine immer gültige Vorlage kann es schon deshalb nicht sein, da sich je nach Länge des BASIC-Quellprogramms die Dummyvariable (und auch die Copyrightmeldung) verschiebt. Man muß sich also die jeweiligen Adressen selbst suchen.

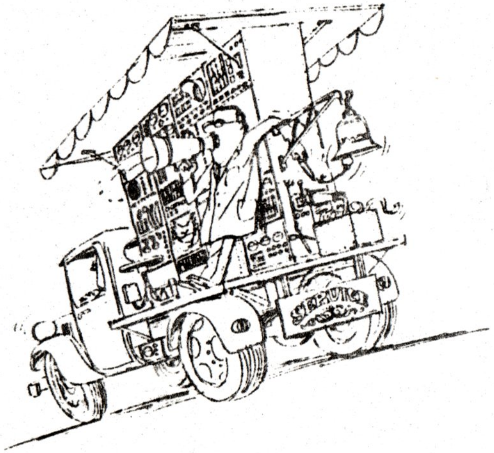
Wie ihr selbst seht, ist diese Methode natürlich reichlich umständlich und ich wäre sehr froh, wenn mir jemand eine bessere Möglichkeit aufzeigt. Am liebsten wäre mir natürlich eine Anleitung, die aufzeigt, wie man Assemblerfiles erstellt, in denen der Linker die Call's als "Global Symbol" erkennt! Sollte sich also jemand in dieser Richtung auskennen, wäre ich für eine Information sehr dankbar!

Euer

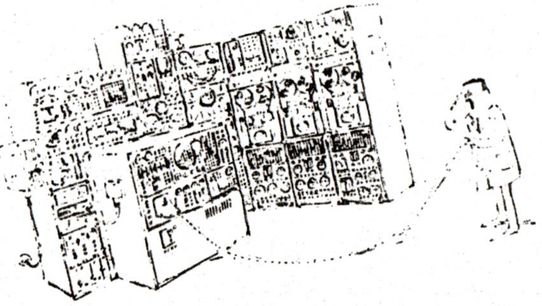
Hartmut Obermann



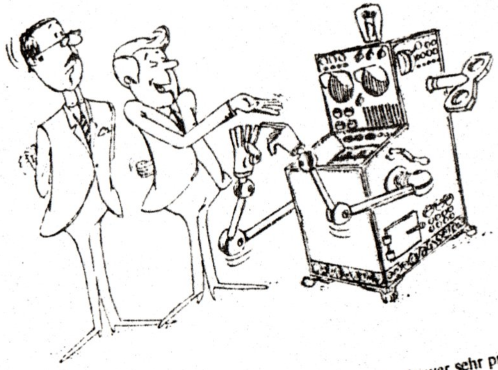
»Auf der letzten Sitzung kamen wir überein, den Betrieb zu automatisieren. Der Prozeß ist beinahe abgeschlossen . . .«



»Daten, frische Daten!«



»Und wenn du diesen Fehler jetzt noch einmal machst, dann schreibst du zur Strafe 5000mal: ICH BIN NUR EINE MASCHINE UND HABE KEINEN FREIEN WILLEN!«



»Gewiß – es gibt bessere Modelle. Aber das da war sehr preisgünstig.«

Model 4 intern: Dem Interpreter aufs Fell geschaut

Naturgemäß wendet sich dieser Beitrag in erster Linie an Assembler-Programmierer und solche, die es werden wollen. Doch heiße Tips gibt es auch für Basic-Fans, zum Beispiel wie der List-Schutz funktioniert.

Zuerst: Viele Routinen aus dem Level2-Basic findet man auch hier wieder. Der Arithmetik-Teil wurde fast unverändert übernommen, nur kräftig gemischt und im Adreßbereich verschoben. Kleine Verbesserungen sind festzustellen, die große Schwachstelle, reiner 8080-Code, blieb. In Listing 1 sind die wichtigsten Unterprogramme aufgeführt. Wo WRA1 und WRA2 hingewandert sind, können sie aus Listing 2 ab 6664 (Hex) entnehmen. Sofern nicht anders vermerkt, sind alle Werte Hex-Zahlen.

Bei der Entwicklung von Ausdrücken benutzt Basic eine Tabelle mit Adressen einiger Arithmetik-Routinen; diese finden Sie in Listing 3. Generell ist WRA1 der am meisten benutzte Speicherplatz. Er hält immer die »laufende Variable«, das heißt, wenn Basic irgendein Ergebnis erwartet, dann muß es da stehen. Um welchen Typ es sich handelt, ist in (60BE) notiert.

Mit PEEK sollte man da ruhig einmal nachschauen; es ist nicht alles »Double-Precision, wo man es vermutet. Die trigonometrischen Funktionen werden immer in einfacher Genauigkeit ausgeführt, und FIX ist noch lange nicht Integer, SGL bleibt erhalten, es wird nur abgeschnitten.

Sollten Sie ein Sortier-Programm schreiben wollen, so vergessen Sie die Subtraktions-Routinen, die »Compares« (siehe Listing 1) sind deutlich schneller.

Der Schlüssel zu einem Interpreter ist natürlich die Adreßtabelle der Basic-Befehle. Sie wurde (Listing 4) mit einem Assembler in lesbare Form gebracht, ein Disassembler zeigt da nur sinnlosen Code. Das Prinzip ist einfach: Trifft Basic auf ein Token, subtrahiert es 81H, multipliziert das mit 2, addiert das Ergebnis auf die Anfangsadresse der Tabelle und schon ist die Adresse der Routine gefunden.

Die Tabelle hat zwei Basis-Adressen, nämlich (5757) und (57F1). Letztere gilt für Doppel-Tokens, deren erstes Byte immer FF ist. Ab 692B können Sie das (mit einem Disassembler) nachlesen, bei 68EB startet der »Execution Driver«.

Nicht alle Tokens sind in dieser Tabelle aufgeführt. Solche, die logisch zu einem Befehl gehören, wie TO zu FOR werden dort, solche die erst später in einem Ausdruck auftreten, werden in EVALEX aufgeprüft. Die Routine zur Entwicklung von Ausdrücken (EVALEX) startet bei 7000, einige weitere Interpreter-Routinen in Listing 5.

Etwas anders als im Level-2-Basic werden Variable abgelegt. Da ein Name anstatt zwei jetzt bis zu 40 signifikante Zeichen haben kann, wird eine Variable in VARTAB so eingetragen:

Byte 1: Typ (2, 3, 4 oder 8)
Byte 2: Erstes Zeichen des Namens (Großbuchstabe)
Byte 3: Zweites Zeichen des Namens (Großbuchstabe oder Ziffer)
Byte 4: Anzahl weitere Zeichen (Länge des Namens 2)
Byte 5 und folgende: die weiteren Zeichen des Namens, ASCCI-Code, aber Bit 7 gesetzt.

Hiernach ist also die Namenslänge nicht auf 40 begrenzt, wohl aber die Größe des Puffers, den Basic für die Behandlung des Namens zur Verfügung stellt (ab 5E4B).

Mit Byte 4 wird erreicht, daß Basic die Variablentabelle schnell scannen kann, der Test eines langen Namens kostet natürlich dennoch mehr Zeit als der eines kurzen.

Alle Ein-/Ausgabe-Operationen laufen ähnlich wie in CP/M über eine Funktionsnummer mit anschließendem »CALL BDOS«, allerdings mit folgenden Unterschieden:

Die Funktionsnummer wird im Register A, eine eventuelle Unterfunktion in B übergeben. Ein auszugebendes Zeichen muß immer in C stehen, die Rückgabe erfolgt ins Register B. Gegebenenfalls sind vor dem Aufruf weitere Register zu laden, auch stehen in weiteren Registern Rückgabeparameter bereit.

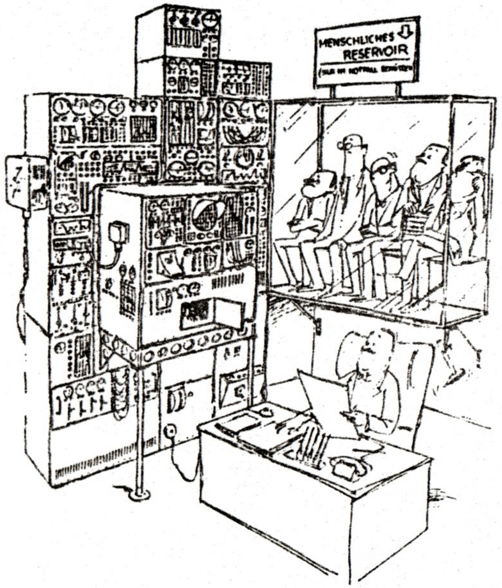
Der Aufruf erfolgt immer über RST 28. Das Z-Flag bedeutet, die Routine wurde (ohne Fehler) ausgeführt, NZ heißt Fehler oder »kein Erfolg«. Wird zum Beispiel die Ta-

```
Arithmetik-Routinen
===================
Bedeutung der Kurzel:
WRA1: Work Area1, WRA2: Work Area 2
INT: Integer, SP: Single Precision, DP: Double Precision
(HL)...: Bytes ab Adresse in HL, dto. (DE)...
TF : Typ-Flag, Inhalt von (60BE)
BCDE: hält einen SP-Operanden als Exp, MSB, LSB, LSB
----------------------------------------------------------------
2607: DP-Add.  : WRA1=WRA1+0.5
2617: SP-ADD.  : WRA1=WRA1+(HL)....
261C: SP-Sub.  : WRA1=(HL)...-WRA1
261F: SP-SUB.  : WRA1=BCDE-WRA1
2622: SP-ADD.  : WRA1=BCDE+WRA1
2746: LOG      : WRA1=LOG(WRA1)
278B: SP-Mult.: WRA1=BCDE*WRA1
27ED: SP-Div.  : WRA1=BCDE/WRA1
28C4: SGN-Test: Nur SP: Sgn(WRA1)->A. -1=neg, +1=pos., 0=0
28E6: ABS      : WRA1=ABS(WRA1)
28F3: NEG      : WRA1=-WRA1
28FB: SGN      : WRA1=SGN(WRA1)
2905: SGN-Test: wie 28C4, aber alle Typen lt. TF
2917: Push WRA: PUSH WRA1 -> Stack, POP mit POP BC, POP DE
2924: Get SP   : WRA1=(HL)...
2927: Get SP   : WRA1=BCDE
2932: Get SP   : BCDE=WRA1
2935: Get SP   : BCDE=(HL)...
293E: Transfer: WRA1->(HL)          4 Bytes
2941: Transfer: (DE)...->(HL)...  4 Bytes
2945: Transfer: (HL)...->(DE)...  TF Bytes
2946: Transfer: (DE)...->(HL)...  TF Bytes
2949: Transfer: (DE)...->(HL)...  A Bytes
294A: Transfer: (DE)...->(HL)...  B Bytes
2970: Transfer: WRA1 -> WRA2        TF Bytes
Die folgenden Compare-Routinen sind schneller als Sub.
2982: Compare : SP : WRA1-BCDE    Flags valid
299C:             Wie vor, wenn Vorzeichen gleich
29AF: Compare : INT: HL-DE        Flags valid
29C4: Compare : DP : WRA1-WRA2    Flags valid
29ED: Compare : DP : WRA2-WRA1    Flags valid
29F4: CINT     : WRA1=HL=CINT(WRA1), alle Typen lt. TF
2A53: MAKINT   : HL->WRA1, setzt TF=2, USR-RET
2A6A: CSNG     : WRA1(DP) -> WRA1(SP)
2A87: Convert : WRA1(INT) -> WRA1(SP)
2A8A: Convert : (HL)...-> WRA1(SP)
2A96: CDBL     : WRA1(alle Typen lt TF) -> WRA1(DP)
2AA9: Set TF8 : TF=8 (DP)
2AAC: Set TF4 : TF=4 (SP)
2AB0: Test TF : IF TF=3 THEN exit to "Type mismatch error"
2AB4: TM-Error: Print "type mismatch error"
2AE2: FIX      : WRA1=FIX(WRA1)
2AF5: FRCINT   : WRA1=HL=INT(WRA1) (SP & DP)
2B1A:             wie vor, aber nur DP
2B93: Sub. INT : WRA1=DE-HL
2B9E: Add. INT : WRA1=DE+HL
2BBE: Mult.INT : WRA1=DE*HL
2CEB: Abs(INT) : WRA1=ABS(WRA1)
2C81: Sub.  DP : WRA1=WRA1-WRA2
2C88: Add.  DP : WRA1=WRA1+WRA2
2DB5: Mult. DP : WRA1=WRA1*WRA2
2DF0: REF       : Konstante 10 in DP
2E8C: Div.  DP : WRA1=WRA1/WRA2 (Siehe auch Entry 2E88)
2F16: ASCII     : Convert ASCII-String (TF) -> WRA1
                  (HL)=Begin of String, Delimeter 00 or ","
3331:           : Print "in", weiter bei 313D
313D: Print #   : Print (WRA1)=(HL), i.d.R. Zeilennummer
3147: Convert   : WRA1 -> ASCII, formatiert lt. Bits set in A
                  Bit muß auf 1, sonst unfomatiert
                  Bit 6  5  4  3  2

                      ,  *  $  +  trailing blanks


36D4: SQR       : WRA1=SQR(WRA1)
36ED: Exp.      : Basis in BCDE, Exp. in WRA1, Result->WRA1
378F: EXP       : WRA1=EXP(WRA1)
3876: RND(N)    : WRA1=RND(1..N)
389C: RND(0)    : WRA1=RND(0..1)
die folgenden trigonmetr. Funktionen ergeben immer SP
38EA: COS       : WRA1=COS(WRA1)
38F0: SIN       : WRA1=SIN(WRA1)
388D: TAN       : WRA1=TAN(WRA1)
39A2: ATN       : WRA1=ATN(WRA1)
```

Listing 1. Die Arithmetik-Routinen des Model 4, das alte Level 2-Basic kaum verändert

```
Systemvariable & Buffer      uberssicht
=======================================

5753-5856: Adressen der Basic-Routinen
5787-588A: Adressen für Anfangsbuchstaben A-Z in
           folgender Liste.
588B-5AC0: Basic-Worte alpabetisch sortiert.
5AC1-5AE1: Liste Operator Precedence
5AE2-5B08: Adressen Arith-Routinen
5B09     : NOP
5B0A-5DF3: Texte der Error Meldungen
5DF4-5E07: Adressen USR0-USR9 oder 6A45 (FC-Error)
5E08-5E74: Variable & Buffer
5E75-5FB4: Buffer für Tokenizing
5FB5-60B4: Input Buffer
60B5-62A4: Variable und Buffer
62A5-62A9: Message "in"
62AA-62B0: Message "ready"
62B1-62B6: Message "Break"
------------------------------------------------------------
Die wichtigsten Variablen
=========================
5E0C: Code last error
5E0E: Druckerposition für LPOS
5E0F: Ausgabe-Flag. 00=CRT
5E11: Zeilenlänge Printer
5E12: Zeilenlänge CRT
5E17: String Data Pointer
5E19: Aktuelle Zeilennummer
5E1B: Head Pointer. Beginn Basic-Pgm
5E1D: Adresse aktueller Error Text
5E2A: Pointer durch Tokenized Buffer
5E4B-5E71: Variable in VARPTR
60B8: Pointer durch Video-Zeile (auch TAB)
60B9-60BB: RND Use
60BD: Flag locate(00)/create(FF) für Find Var Routine
60BE: Typ-Flag (2=INT. 3=STRING. 4=SGL. 8=DBL)
60C1: letzte Zeilennummer vor aktueller
60CF: HIMEM-Pointer
60D1: Adr. next free Entry to LSTPA
60D3-60F0: Literal String Pool Area. 10*3 Bytes
60F1-60F3: Dope Vektor aktueller String
60F6: Adresse aktuelles Token
60FC: Aktuelle DATA-Zeile
60FE: CALL-Flag
60FF: Read/Input Flag. 00=Input, else Read
6100: Auto Flag
6104: Zeilennummer in Eingabe
6106: Auto-Inkrement
6108: Encoded Statement Pointer
610A: Stack Pointer
610C: ON ERROR-Zeile
610E: EDIT-Zeile. lfd. Zeile in LIST
6110: Text-Pointer in Error-Zeile
6112: ON ERROR Adresse
6114: RESUME-Flag (00) else FF
6115: Adresse next Token (wenn Backspacing)
6117: CONT-Zeile
6119: CONT-Adresse
611B: Tail Pointer (Ende Basic-Pgm)
611D: Pointer einfache Variable
611F: Pointer Array-Variable
6121: DATA-Pointer
6123-613C Typ-Tabelle A-Z für DEFxxx
621F: Option Base (0 oder 1)
6220: Option Base Flag (Option Base+1)
6247: Prot Flag (>0: no LIST. no Direkt Mode)
6259: Trace Flag

Work Area 1       WRA1
---------------------------------------
Adr.   if INT    if SGL       if DBL
---------------------------------------
6264                          LSB
6265                          LSB
6266                          LSB
6267                          LSB
6268   LSB       LSB          LSB
6269   MSB       LSB          LSB
626A             MSB          MSB
626B             EXP          EXP

Work Area 2      WRA2
---------------------------------------
6272   LSB       LSB          LSB
6273   MSB       LSB          LSB
6274             MSB          LSB
6275             EXP          LSB
6276                          LSB
6277                          LSB
6278                          MSB
6279                          EXP

627A-6294: PRINT USING Buffer
```

Listing 2. Systemvariable, Texte und Puffer; Adressen in der Übersicht und im einzelnen




```
                 00100 ;ADRESS-TABELLE FUER ARIHT-ROUTINEN
                 00110 ;================================
                 00120
5AE2             00130          ORG     5AE2H
5AE2             00140 BASEDP   EQU     $          ;BASIS-ADR. WENN DBL PRECIS.
5AE2 962A        00150          DEFW    2A96H      ;CDBL
5AE4 0000        00160          DEFW    0000H      ;NOT USED
5AE6 F429        00170          DEFW    29F4H      ;CINT
5AE8 B02A        00180          DEFW    2AB0H      ;TESTE TYP
5AEA 6A2A        00190          DEFW    2A6AH      ;CSNG
5AEC 882C        00200          DEFW    2C88H      ;ADD  (DBL)
5AEE 812C        00210          DEFW    2C81H      ;SUB  (DBL)
5AF0 B52D        00220          DEFW    2DB5H      ;MULT (DBL)
5AF2 8C2E        00230          DEFW    2E8CH      ;DIV. (DBL)
5AF4 ED29        00240          DEFW    29EDH      ;COMPARE (DBL)
5AF6             00250 BASSGL   EQU     $          ;BASIS ADR. SGL PRECIS.
5AF6 2226        00260          DEFW    2622H      ;ADD  (SGL)
5AF8 1F26        00270          DEFW    261FH      ;SUB  (SGL)
5AFA 8B27        00280          DEFW    278BH      ;MULT (SGL)
5AFC ED27        00290          DEFW    27EDH      ;DIV. (SGL)
5AFE 8229        00300          DEFW    2982H      ;COMPARE (SGL)
5B00             00310 BASINT   EQU     $          ;BASIS-ADRESS INTEGERS
5B00 9E2B        00320          DEFW    2B9EH      ;ADD
5B02 932B        00330          DEFW    2B93H      ;SUB
5B04 BE2B        00340          DEFW    2BBEH      ;MULT
5B06 6C71        00350          DEFW    716CH      ;DIV.
5B08 AF29        00360          DEFW    29AFH      ;COMPARE
5B0A             00380 ERROR    EQU     $          ;BEGINN TEXTE ERROR MELDUNGEN
0000             00390          END
00000 TOTAL ERRORS
34294  TEXT AREA BYTES LEFT


BASEDP 5AE2 00140
BASINT 5B00 00310
BASSGL 5AF6 00250
ERROR  5B0A 00380
```

Listing 3. Nach dieser Adreßtabelle werden Arithmetik-Routinen aufgerufen

statur abgefragt, heißt NZ »keine Taste gedrückt«, Z dann »Taste betätigt«, der Code steht in A. In Listing 6 sind einige Routinen praktisch demonstriert. Sie sind ein Teil des DOS, das hier aus Platzgründen nicht behandelt werden kann. Nur soviel: Auch jede Disk-Operation läuft über RST 28. Sie sollten nie eine Routine direkt aufrufen. Der Sinn dieser Organisation ist es, das System beliebig erweitern (und ändern) zu können, womit sich absolute Adressen verschieben, die Funktionsnummern aber bleiben. Den Drucker hätte ich fast vergessen, Funktion 6, Zeichen in C, oder direkt über Port F8. Die RST-Calls des Level2-Basics sind hier entfallen. Das kostet zwar Speicher, macht das System aber schneller. Aus RST 8 (Syntax-Check) wurde CALL 430E, RST 10 (Scanner) ist jetzt CALL 6943, RST 18 (HL-DE) wurde zu CALL 4308 und RST 20 (get vartyp) zu CALL 735E.

Interpreter mit Error-Meldungen knacken

Ein weiterer Weg, so einen Interpreter zu knacken, sind die Error-Meldungen, die hier auf vier Arten erzeugt werden. Meistens geschieht dies über JP in die Sprungliste (Listing 7). Mit einem Disassembler sieht man da nur Unmengen von »LD BC,nn«, ein beliebter Trick, den Microsoft häufig anwendet. Wenn Sie in der Adreßtabelle nur 3-Byte-Abstände vorfinden, geht das auf diese Art. Bei 2-Byte-Abständen steht da »OR AF«, womit A > 0 wird. Steigt man beim zweiten Byte ein (AF), heißt das »XOR A«, also A = 0. Zu den Error-Meldungen, Weg 2: "LD E,Code/JP 6343". 6343 ist die Adresse, auf die die vorher genannte Liste durchfällt. Weg 3: ein JP zur Adresse, wo ein »Weg 2« beginnt, und Weg 4: HL wird mit der Adresse des Fehlertextes geladen und dann wird 4660 (Print Text ab (HL)) gerufen. Als Hilfe dazu Listing 8.

Störend ist oft, daß bei Syntax-Fehlern sofort EDIT aufgerufen wird, womit leider alle Variablen gelöscht werden. Wollen Sie das vermeiden, poken Sie NOPs (00) auf 63F5-63F7. Da wir gerade beim Poken sind. Ist Ihr Drucker grafikfähig, Sie wollen also die Punkte nicht im Screenprint, dann reicht ►

```
                :TABELLE MIT ADRESSEN DER BASIC-ROUTINEN LT. TOKEN
                ;=================================================
                
                :FUER NICHT BELEGTE TOKENS STEHT '****'.
                ;ADR. 0000: RET
                ;ADR. 6329: PRINT "SYNTAX-ERROR"
                
5757                            ORG     5757H
5757 DE43                       DEFW    43DEH   ;81: END
5759 0A68                       DEFW    680AH   ;82: FOR
575B E244                       DEFW    44E2H   ;83: NEXT
575D 156B                       DEFW    6B15H   ;84: DATA
575F 6C6E                       DEFW    6E6CH   ;85: INPUT
5761 F639                       DEFW    39F6H   ;86: DIM
5763 476F                       DEFW    6F47H   ;87: READ
5765 3C6B                       DEFW    6B3CH   ;88: LET
5767 BA6A                       DEFW    6ABAH   ;89: GOTO
5769 BF6A                       DEFW    6ABFH   ;8A: RUN
576B 776C                       DEFW    6C77H   ;8B: IF
576D 2043                       DEFW    4320H   ;8C: RESTORE
576F A36A                       DEFW    6AA3H   ;8D: GOSUB
5771 F56A                       DEFW    6AF5H   ;8E: RETURN
5773 176B                       DEFW    6B17H   ;8F: REM
5775 3A43                       DEFW    433AH   ;90: STOP
5777 DB6C                       DEFW    6CDBH   ;91: PRINT
5779 3944                       DEFW    4439H   ;92: CLEAR
577B 4076                       DEFW    7640H   ;93: LIST
577D 6B62                       DEFW    626BH   ;94: NEW
577F A26B                       DEFW    6BA2H   ;95: ON
5781 F975                       DEFW    75F9H   ;96: WAIT
5783 1D74                       DEFW    741DH   ;97: DEF
5785 C578                       DEFW    78C5H   ;98: POKE
5787 9743                       DEFW    4397H   ;99: CONT
5789 2963                       DEFW    6329H   ;9A: ****
578B 2963                       DEFW    6329H   ;9B: ****
578D F375                       DEFW    75F3H   ;9C: OUT
578F B56C                       DEFW    6CB5H   ;9D: LPRINT
5791 3B76                       DEFW    763BH   ;9E: LLIST
5793 D47C                       DEFW    7CD4H   ;9F: CLS
5795 0000                       DEFW    0000H   ;A0: ****
5797 2963                       DEFW    6329H   ;A1: ****
5799 176B                       DEFW    6B17H   ;A2: ELSE
579B AB43                       DEFW    43ABH   ;A3: TRON
579D AC43                       DEFW    43ACH   ;A4: TROFF
579F B143                       DEFW    43B1H   ;A5: SWAP
57A1 F243                       DEFW    43F2H   ;A6: ERASE
57A3 223D                       DEFW    3D22H   ;A7: EDIT
57A5 3A6C                       DEFW    6C3AH   ;A8: ERROR
57A7 EB6B                       DEFW    6BEBH   ;A9: RESUME
57A9 7D78                       DEFW    787DH   ;AA: DELETE
57AB 456C                       DEFW    6C45H   ;AB: AUTO
57AD FE78                       DEFW    78FEH   ;AC: RENUM
57AF F969                       DEFW    69F9H   ;AD: DEFSTR
57B1 FC69                       DEFW    69FCH   ;AE: DEFINT
57B3 FF69                       DEFW    69FFH   ;AF: DEFSNG
57B5 026A                       DEFW    6A02H   ;B0: DEFDBL
57B7 016E                       DEFW    6E01H   ;B1: LINE
57B9 0000                       DEFW    0000H   ;B2: ****
57BB 0000                       DEFW    0000H   ;B3: ****
57BD 2B52                       DEFW    522BH   ;B4: WHILE
57BF 4C52                       DEFW    524CH   ;B5: WEND
57C1 BB52                       DEFW    52BBH   ;B6: CALL
57C3 5C56                       DEFW    565CH   ;B7: WRITE
57C5 156B                       DEFW    6B15H   ;B8: COMMON
57C7 4E53                       DEFW    534EH   ;B9: CHAIN
57C9 2E7A                       DEFW    7A2EH   ;BA: OPTION
57CB 527D                       DEFW    7D52H   ;BB: RANDOM
57CD 0000                       DEFW    0000H   ;BC: ****
57CF A17D                       DEFW    7DA1H   ;BD: SYSTEM
57D1 0000                       DEFW    0000H   ;BF: ****
57D3 7250                       DEFW    5072H   ;BF: OPEN
57D5 7B4D                       DEFW    4D7BH   ;C0: FIELD
57D7 7D4F                       DEFW    4F7DH   ;C1: GET
57D9 7C4F                       DEFW    4F7CH   ;C2: PUT
57DB 3B4D                       DEFW    4D3BH   ;C3: CLOSE
57DD DC4B                       DEFW    4BDCH   ;C4: LOAD
57DF 9A4C                       DEFW    4C9AH   ;C5: MERGE
57E1 0000                       DEFW    0000H   ;C6: ****
57E3 B27E                       DEFW    7EB2H   ;C7: NAME
57E5 8F51                       DEFW    518FH   ;C8: KILL
57E7 D54D                       DEFW    4DD5H   ;C9: LSET
57E9 D44D                       DEFW    4DD4H   ;CA: RSET
57EB 0000                       DEFW    0000H   ;CB: ****
57ED 0000                       DEFW    0000H   ;CD: SOUND
                
                ;JETZT FOLGEN 2-BYTES TOKENS.
                ;DAS ERSTE BYTE IM BASIC-TEXT IST IMMER FF.
                ;IN DER LISTE STEHT DAS ZWEITE.
                
57EF 7348                       DEFW    4873H   ;81: LEFT$
57F1 A348                       DEFW    48A3H   ;82: RIGHT$
57F3 AC48                       DEFW    48ACH   ;83: MID$
57F5 FB28                       DEFW    28FBH   ;84: SGN
57F7 F52A                       DEFW    2AF5H   ;85: INT
57F9 E628                       DEFW    28E6H   ;86: ABS
57FB D436                       DEFW    36D4H   ;87: SQR
57FD 7638                       DEFW    3876H   ;88: RND
57FF F038                       DEFW    38F0H   ;89: SIN
5801 4627                       DEFW    2746H   ;8A: LOG
5803 8F37                       DEFW    378FH   ;8B: EXP
5805 EA38                       DEFW    38EAH   ;8C: COS
5807 8D39                       DEFW    398DH   ;8D: TAN
5809 A239                       DEFW    39A2H   ;8E: ATN
580B 244A                       DEFW    4A24H   ;8F: FRE
580D DC75                       DEFW    75DCH   ;90: INP
580F C273                       DEFW    73C2H   ;91: POS
5811 0548                       DEFW    4805H   ;92: LEN
5813 D945                       DEFW    45D9H   ;93: STR$
5815 CD48                       DEFW    48CDH   ;94: VAL
5817 1148                       DEFW    4811H   ;95: ASC
5819 2148                       DEFW    4821H   ;96: CHR$
581B BB78                       DEFW    78BBH   ;97: PEEK
581D 5849                       DEFW    4958H   ;98: SPACE$
581F CF45                       DEFW    45CFH   ;99: OCT$
5821 D445                       DEFW    45D4H   ;9A: HEX$
5823 BD73                       DEFW    73BDH   ;9B: LPOS
5825 F429                       DEFW    29F4H   ;9C: CINT
5827 6A2A                       DEFW    2A6AH   ;9D: CSNG
5829 962A                       DEFW    2A96H   ;9E: CDBL
582B E22A                       DEFW    2AE2H   ;9F: FIX
                :TOKENS A6-A9 ADR. 0000H ****
5843                            ORG     5843H
5843 9B4A                       DEFW    4A9BH   ;AA: CVI
5845 9E4A                       DEFW    4A9EH   ;AB: CVS
5847 A14A                       DEFW    4AA1H   ;AC: CVD
5849 E34E                       DEFW    4EE3H   ;AD: EOF
584B 274F                       DEFW    4F27H   ;AE: LOC
584D 2A4F                       DEFW    4F2AH   ;AF: LOF
584F 824A                       DEFW    4A82H   ;B0: MKI$
5851 854A                       DEFW    4A85H   ;B1: MKS$
5853 884A                       DEFW    4A88H   ;B2: MKD$
5855 987D                       DEFW    7D98H   ;B3: ROW
0000                            END
00000 TOTAL ERRORS
31865 TEXT AREA BYTES LEFT
```

Listing 4. Der Schlüssel zum Interpreter: Adressen der Routinen je Befehl

```
Adressen einiger Interpreter-Routinen
=====================================
39FB: FNDVAR    : Find Var.Adr. Var-Name pointed by HL,
                  Deli. 00 or ",". Adr. -> DE
410E: PRTCRT    : Output Device is CRT
4308: Compare   : HL-DE, Flags valid
430E: SNTEST    : Teste auf Zeichen, das dem Call folgt,
                  Syntax-Error wenn <>, Incr. HL
442E: TESTLET.  : Teste Zeichen (HL) auf Buchszaben, NC if
442F:           : wie vor, Zeichen in A
4660: PRTLIN    : Print Text ab (HL) bis 00
5740: NODIRT    : Verhindert Direkt-Mode wenn Prot-Flag (6247)
                  gesetzt (<>00)
5749: TESTPT    : Teste P-Mode (Listschutz)
6343: PRTERR    : Print Error-Text, Code in E
6521: NEWLIN    : Update Line Pointers
6564: FNDLIN    : Find Line Number
6943: NXTBYT    : Hole nächstes Zeichen aus dem Pgm-Text.
                  Skip Blanks u. einige Controls, behandle
                  einige Codes direkt.
```

Listing 5. Einige Interpreter-Routinen, weitere im Text

Wenn bei uns schon mal was schiefgeht, dann aber gründlich!

```
                :DEMONSTRATION DOS-CALLS (RST 28H)
                ;=================================
                
                :TASTAUR-ABFRAGE
                ;---------------
0000 3E08       SCANKI  LD      A,8             ;TESTE KEYBOARD
0002 EF                 RST     28H             ;CALL FUNCTION
0003 20FB               JR      NZ,SCANKI
                                                ;CHAR NOW IN A
                
0005 3E01       WAITKI  LD      A,1             ;SCAN KB & WAIT
0007 EF                 RST     28H
                                                ;CHAR NOW IN A
                
0008 3E09       GETLIN  LD      A,9             ;GET LINE
000A 0607               LD      B,7             ;ALLOW MAX 7 CHAR
000C 0E00               LD      C,0             ;MAX NOW IN BC
000E 21B55F             LD      HL,5FB5H        ;USE BASICS BUFFER
0011 EF                 RST     28H
0012 3804               JR      C,BREAK         ;IF BREAK TERMINATOR
0014 AF                 XOR     A               ;A=0
0015 B8                 CP      B               ;CHARS INPUT
0016 28F0               JR      Z,GETLIN        ;IF NULLSTRING
0018            BREAK   EQU     $               ;HANDLE BREAK HERE
                
                ;AUSGABE AUF VIDEO
                ;-----------------
0018 3E02       DSPCHR  LD      A,2             ;DISP CHAR & CURSOR
001A 0E58               LD      C,'X'           ;CHAR TO DISP
001C EF                 RST     28H
                
7CD4                    ORG     7CD4H           ;BASICS CLS-ROUTINE
7CD4 E5                 PUSH    HL
7CD5 0E1C               LD      C,1CH           ;HOME CURSOR
7CD7 3E02               LD      A,2
7CD9 EF                 RST     28H
7CDA 0E1F               LD      C,1FH           ;ERASE TO END OF FRAME
7CDC 3E02               LD      A,2
7CDE EF                 RST     28H
7CDF E1                 POP     HL
7CE0 AF                 XOR     A               ;A=0
7CE1 32B860             LD      (60B8H),A       ;TAB(0)
7CE4 C9                 RET
                
7CE5 3E0A       DSPMSG  LD      A,0AH           ;DISP MESSAGE
7CE7 21ED7C             LD      HL,MESSG        ;POINT TO TEXT
7CEA EF                 RST     28H
7CEB 1809               JR      POINT
7CED 54         MESSG   DEFM    'TEXT HERE'
                        DEFB    03H
                :VIDEO CONTROL
                ;-------------
7CF6 3E0F       POINT   LD      A,0FH           ;VIDEO CONTROL
7CF8 0601               LD      B,1             ;FUNCTION POINT
7CFA 2603               LD      H,3             ;ROW 3
7CFC 2E07               LD      L,7             ;COL 7
7CFE EF                 RST     28H
                                                ;CHAR (H),(L) NOW
                
7CFF 3E0F       PRTAT   LD      A,0FH
7D01 0602               LD      B,2             ;FUNCTION PRINT AT
7D03 210703             LD      HL,0307H        ;ROW 3, COL 7
7D06 0E58               LD      C,'X'           ;CHAR TO PRINT
7D08 EF                 RST     28H
                
7D09 3E0F       MOVCUR  LD      A,0FH
7D0B 0603               LD      B,3             ;FUNCTION MOVE CURSOR
7D0D 210703             LD      HL,0307H        ;ROW 3, COL 7
7D10 EF                 RST     28H
                
7D11 3E0F       FNDCUR  LD      A,0FH
7D13 0604               LD      B,4             ;FIND CURSOR POSITION
7D15 EF                 RST     28H
                                                ;CURSOR POS. NOW IN H,L
0000                    END
00000 TOTAL ERRORS
33502 TEXT AREA BYTES LEFT
```

Listing 6. Jegliches I/O läuft über DOS-Calls. Hier einige praktische Beispiele

"POKE &HOACE,&H18". Unter Programm-Kontrolle können Sie den Screendump auch aufrufen und zwar mit :"X=&HOABA:CALL X". Das sind zwar leider absolute Calls ins DOS, aber dafür habe ich keine Funktionsnummer gefunden. Für folgendes wird es wohl nie eine geben, die Sache mit dem Listschutz: Ein mit P-Option gesichertes Programm setzt das Flag in 6247 <>0. So ziemlich am Anfang der LIST-Routine wird das Flag geprüft. Jetzt poken geht nicht mehr, da die Routine, die Direkt-Mode testet, auch das Flag prüft. Die wird aber nicht immer aufgerufen, zum Beispiel nicht bei SYSTEM. Sie können also in aller Ruhe ein Programm nachladen (und dabei starten) in dem steht »COR A:LD« (6247),A:RET". Haben Sie das nicht, geht folgendes: Vor dem Laden des P-Programms poken Sie auf 5740 ein C9 (RET), und nach dem Laden auf 6247 eine Null, und Ihr versehentlich mit P gespeichertes Programm ist wieder sichtbar.

Konstanten schneller als Variablen

Bliebe noch zu erwähnen, warum hier Konstanten schneller sind als Variable. Sie werden direkt nach der Eingabe umgewandelt und stehen als Hex-Zahlen im Programmtext.

Der Encoder startet bei 6583, die Konstanten werden ab 6731 konvertiert. Aus "&Hxx" wird "OC nn", wobei nn für die beiden Bytes in

```
                 00001 ;                   ADRESSEN FEHLERMELDUNGEN
                 00002 ;                   ========================
                 00003 ;EIN JP ODER CALL ADR. FUEHRT ZUR AUSGABE DES
                 00004 ;ENTPRECHENDEN TEXTES. "LD BC,NN" SCHUETZT DIE
                 00005 ;"LD E,N"-BEFEHLE.
                 00006
62FB             00007          ORG     62FBH
62FB 1E3D        00008          LD      E,3DH   ;DISK FULL
62FD 01          00009          DEFB    01
62FE 1E39        00010          LD      E,39H   ;DEVICE I/O-ERROR
6300 01          00011          DEFB    01
6301 1E36        00012          LD      E,36H   ;BAD FILE MODE
6303 01          00013          DEFB    01
6304 1E35        00014          LD      E,35H   ;FILE NOT FOUND
6306 01          00015          DEFB    01
6307 1E34        00016          LD      E,34H   ;BAD FILE NUMBER
6309 01          00017          DEFB    01
630A 1E33        00018          LD      E,33H   ;INTRENAL ERROR
630C 01          00019          DEFB    01
630D 1E3E        00020          LD      E,3EH   ;INPUT PAST END
630F 01          00021          DEFB    01
6310 1E37        00022          LD      E,37H   ;FILE ALREADY OPEN
6312 01          00023          DEFB    01
6313 1E40        00024          LD      E,40H   ;BAD FILE NAME
6315 01          00025          DEFB    01
6316 1E3F        00026          LD      E,3FH   ;BAD RECORD NUMBER
6318 01          00027          DEFB    01
6319 1E32        00028          LD      E,32H   ;FIELD OVERFLOW
631B 01          00029          DEFB    01H
631C 1E43        00030          LD      E,43H   ;TO MANY FILES
631E 01          00031          DEFB    01
631F 1E3A        00032          LD      E,3AH   ;FILE ALREADY EXISTS
6321 1820        00033          JR      PRINT
                 00034
6323 2AFC60      00035          LD      HL,(60FCH)      ;LAST DATA LINE
6326 22195E      00036          LD      (5E19H),HL      ;CURRENT LINE
                 00037
6329 1E02        00038          LD      E,02H   ;SYNTAX ERROR
632B 01          00039          DEFB    01
632C 1E0B        00040          LD      E,0BH   ;DIVISION BY ZERO
632E 01          00041          DEFB    01
632F 1E01        00042          LD      E,01H   ;NEXT WITHOUT FOR
6331 01          00043          DEFB    01
6332 1E0A        00044          LD      E,0AH   ;REDIMED ARRAY
6334 01          00045          DEFB    01
6335 1E12        00046          LD      E,12H   ;UNDEFINED USER FUNCTION
6337 01          00047          DEFB    01
6338 1E14        00048          LD      E,14H   ;RESUME WITHOUT ERROR
633A 01          00049          DEFB    01
633B 1E06        00050          LD      E,06    ;OVERLOW
633D 01          00051          DEFB    01
633E 1E16        00052          LD      E,16H   ;MISSING OPERAND
6340 01          00053          DEFB    01
6341 1E0D        00054          LD      E,0DH   ;TYPE MISMATCH
6343             00055 PRINT    EQU     $
0000             00056          END
00000 TOTAL ERRORS
33897  TEXT AREA BYTES LEFT

PRINT  6343 00055   00033
```

Listing 7. Entry-Points zu Fehlermeldungen: Demonstration eines häufig von Microsoft gebrauchten Tricks

LSB/MSB-Order steht. Auch eine Oktal-Konstante wird so abgelegt, nur steht anstatt 0C das Typ-Byte 0B. Eine Dezimal-Konstante wird ähnlich behandelt, wenn sie größer als 255 ist, das Typ-Byte ist dann 1C. Bei Zahlen von 0 bis 9 wird 1 addiert (01H-0AH), das Typ-Byte entfällt dann, das higher Nibble wird aber 1, aus 0 - 9 wird 11H - 1AH. Aus Zahlen von 10 bis 255 wird OAH - FFH, wobei das Typ-Byte jetzt 0FH ist. Wenn Sie also beim Disassemblieren im Scanner (so ab 6966) oder im LIST auf diese Bytes stoßen, that's it. Da direkt nach der Eingabe konvertiert wird, muß da auch geprüft werden. Das ist der Grund, warum die Eingabe einer Zeile mit FC-Error quittiert werden kann.

Bliebe noch die obligatorische Frage zu klären, wo man in den Interpreter mit eigenen Routinen einsteigt. Hier ist die Sache ganz einfach, da man immer im RAM ist. Man »patche« da, wo es für den jeweiligen Fall am günstigsten ist. Die Routine, die die Tastatur abfragt, startet bei 7BC8. Hauptschleife und Scanner hatten wir schon erwähnt, EVALEX auch, also: Wo bleiben die interessanten Listings für das Modell 4?

(P. Wollschläger/ml)

```
         00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
         -----------------------------------------------
5B00 =>  9E 2B 93 2B BE 2B 6C 71 AF 29 00 4E 45 58 54 20    .+.+.+lq.).NEXT
5B10 =>  77 69 74 68 6F 75 74 20 46 4F 52 00 53 79 6E 74    without FOR.Synt
5B20 =>  61 78 20 65 72 72 6F 72 00 52 45 54 55 52 4E 20    ax error.RETURN
5B30 =>  77 69 74 68 6F 75 74 20 47 4F 53 55 42 00 4F 75    without GOSUB.Ou
5B40 =>  74 20 6F 66 20 44 41 54 41 00 49 6C 6C 65 67 61    t of DATA.Illega
5B50 =>  6C 20 66 75 6E 63 74 69 6F 6E 20 63 61 6C 6C 00    l function call.
5B60 =>  4F 76 65 72 66 6C 6F 77 00 4F 75 74 20 6F 66 20    Overflow.Out of
5B70 =>  6D 65 6D 6F 72 79 00 55 6E 64 65 66 69 6E 65 64    memory.Undefined
5B80 =>  20 6C 69 6E 65 20 6E 75 6D 62 65 72 00 53 75 62     line number.Sub
5B90 =>  73 63 72 69 70 74 20 6F 75 74 20 6F 66 20 72 61    script out of ra
5BA0 =>  6E 67 65 00 44 75 70 6C 69 63 61 74 65 20 44 65    nge.Duplicate De
5BB0 =>  66 69 6E 69 74 69 6F 6E 00 44 69 76 69 73 69 6F    finition.Divisio
5BC0 =>  6E 20 62 79 20 7A 65 72 6F 00 49 6C 6C 65 67 61    n by zero.Illega
5BD0 =>  6C 20 64 69 72 65 63 74 00 54 79 70 65 20 6D 69    l direct.Type mi
5BE0 =>  73 6D 61 74 63 68 00 4F 75 74 20 6F 66 20 73 74    smatch.Out of st
5BF0 =>  72 69 6E 67 20 73 70 61 63 65 00 53 74 72 69 6E    ring space.Strin
5C00 =>  67 20 74 6F 6F 20 6C 6F 6E 67 00 53 74 72 69 6E    g too long.Strin
5C10 =>  67 20 66 6F 72 6D 75 6C 61 20 74 6F 6F 20 63 6F    g formula too co
5C20 =>  6D 70 6C 65 78 00 43 61 6E 27 74 20 63 6F 6E 74    mplex.Can't cont
5C30 =>  69 6E 75 65 00 55 6E 64 65 66 69 6E 65 64 20 75    inue.Undefined u
5C40 =>  73 65 72 20 66 75 6E 63 74 69 6F 6E 00 4E 6F 20    ser function.No
5C50 =>  52 45 53 55 4D 45 00 52 45 53 55 4D 45 20 77 69    RESUME.RESUME wi
5C60 =>  74 68 6F 75 74 20 65 72 72 6F 72 00 55 6E 70 72    thout error.Unpr
5C70 =>  69 6E 74 61 62 6C 65 20 65 72 72 6F 72 00 4D 69    intable error.Mi
5C80 =>  73 73 69 6E 67 20 6F 70 65 72 61 6E 64 00 4C 69    ssing operand.Li
5C90 =>  6E 65 20 62 75 66 66 65 72 20 6F 76 65 72 66 6C    ne buffer overfl
5CA0 =>  6F 77 00 3F 00 3F 00 46 4F 52 20 57 69 74 68 6F    ow.?.?.FOR Witho
5CB0 =>  75 74 20 4E 45 58 54 00 3F 00 3F 00 57 48 49 4C    ut NEXT.?.?.WHIL
5CC0 =>  45 20 77 69 74 68 6F 75 74 20 57 45 4E 44 00 57    E without WEND.W
5CD0 =>  45 4E 44 20 77 69 74 68 6F 75 74 20 57 48 49 4C    END without WHIL
5CE0 =>  45 00 46 49 45 4C 44 20 6F 76 65 72 66 6C 6F 77    E.FIELD overflow
5CF0 =>  00 49 6E 74 65 72 6E 61 6C 20 65 72 72 6F 72 00    .Internal error.
```

Listing 8. Viele Fehlermeldungen werden nach direktem Zugriff auf diese Tabelle ausgegeben

Es darf gelacht werden

NEWS OF THE WORLD

vom Reporter des Satens: F.W.

Softwarepiraten

London: Die Polizei Londons hob ein bestens organisiertes Nest britischer Amstrad Softwarepiraten aus. Den alten Trick, den Anfang der kopierten Kassette mit Musik zu bespielen, hatte die Gang perfektioniert. Wie der Polizeichef mitteilen ließ, wurde zur Tarnung die ganze Kassette mit Musik überspielt. Raffiniert sorgte die Bande dafür, daß der Hausdurchsuchungstrupp auch keinen Computer fand. Der Prozeß findet trotzdem im Mai 1987 statt. Bis dahin bleiben die Tatverdächtigten im Tower und können gegen Entgelt besichtigt werden.

Prozessorwechsel bei Schneider

Türkheim: Die Firma SCHNEIDER wird für ein neues Produkt dem bewährten Z80 untreu und wendet sich dem 68000 von MOTORROLLER zu. Eingesetzt wird der Prozessor erstmals bei einem überdachten Mofa mit Schleudersitz, Kaffeemaschine und obenliegendem Handschuhfach. Der TÜV weiß gottseidank noch nichts davon.

Computer für Leere und Forschung

Wiesbaden: Aus dem Vorhaben der hessischen Landesregierung, Universitäten mit Computern auszurüsten, wird vorerst nichts. Ministerpräsident Bönner weigert sich, dem Wunsch der Grünen nach einem Grünmonitor nachzugeben. Bönner dazu wörtlich: "Ich weigere mich, dem Wunsch der Grünen nach einem Grünmonitor nachzugeben! Wenigstens vorerst."

Volkszählung verschoben

Bonn: Die schon einmal verlegte Volkszählung wird erneut um unbestimmte Zeit verschoben, da die nötige Computerkapazität momentan nicht zur Verfügung steht. Wie es heißt, erfordere ein von den Angestellten zwischenzeitlich installiertes Adventure-Game mehr Zeit zur Lösung als geplant.

Jugendschutz bei Videospielen

Bonn: Einige Videospiele wurden aufgrund ihrer Gewaltverherrlichung indiziert. Um auch bei Selbstprogrammierern wirksam werden zu können, wird das Jugendamt in Zusammenarbeit mit dem Gesundheitsamt in Zukunft Hauskontrollen durchführen. In Eigenarbeit programmierte Spiele, die nicht dem Jugendschutz genügen, werden dann ersatzlos eingezogen. Dem ertappten Sünder wird ein Strafvortrag über Zahnpflege und Fußpilz durch Turnschuhe gehalten.

Grundstücksschwindel aufgedeckt

Frankfurt: Die nächste Frankfurter Computermesse ist für das Frühjahr 1986 geplant. Die Messeleitung korrigiert den bei einigen diesjährigen Besuchern entstandenen Eindruck, sie seien Besitzer eines Grundstücks. Der enorme Preis für einen Parkplatz sei, so erklärte seitens der Messeleitung Herr R. Affke, die ortsübliche Parkgebühr, nicht etwa der Kaufpreis.

Rat & Hülfe

von F. W.

J. Dämiel fragt:
Ich würde mir gerne eine Floppy kaufen, aber dann wäre ja das Cassettengerät am 464 nutzlos. Soll ich nicht lieber gegen einen 664 umtauschen.

F.W. antwortet:
Das Cassettendeck ist auch nach der Anschaffung der Floppy nicht nutzlos. Es eignet sich hervorragend als Staubschutzhülle einer Cassette. Zudem wird der Platz als Abstellfläche der Kaffeetasse benötigt.

B. Lödsinn fragt:
Daß LOGO von Bchneider Boll von SCHNEIDERN-AKTIV nicht mehr benutzt werden. Nun Btellt dieß aber doch lediglich ein "ß" dar. Heißt daß jetzt, daß ein "ß" nicht mehr benutzt werden darf? Bicherheitßhalber habe ich mal meinen Brief mit dem Doppel-ß geßchrieben, man kann ja nie wißßen. Oder hat Bich daß "ß" etwa auch Bchon eine Firma alß LOGO gekrallt?

F.W. antwortet:
Leider, leider erwurten wir den Einßpruch von Uturi, bezüglich deß "U" uus UTURI. Denken Bie bitte in Zukunft un die Möglichkeit einer Kluge, wenn Bie Ihre Briefe ubfußßen. Vielen Dunk.

SMARTSCREEN, SMARTKEY und SMARTPRINT: nützliche Utilities — oder Spielerei?

Auch den neueren Versionen des CP/M-Betriebssystemes, CP/M 2.2 und sogar CP/M 3.0, sieht man noch an, daß es seine Wurzeln in der Zeit der Computer-Veteranen hat. Die Ein- und Ausgabe ist zeilenorientiert — schließlich standen damals nur Lochstreifenleser und -stanzer (daher die »reader«- und »puncher«-Routinen im BIOS) sowie Fernschreibterminals zur Verfügung. Mit zunehmender Verbreitung der Bildschirmsichtgeräte wurden diese auch für den Personal Computer interessant.

Steigende Integrationsdichte und der Verfall der Preise für integrierte Schaltungen sorgten bald dafür, daß Bildschirmsichtgeräte mit erheblicher Intelligenz ausgestattet wurden. Heute führt meist ein eigener Mikroprozessor Befehle wie »Bildschirm löschen«, »Löschen von Cursor bis Zeilenende«, »Löschen von Cursor bis Bildende«, »Einfügen einer Leerzeile«, »Ausfügen einer Zeile«, und so weiter aus. Damit waren die Voraussetzungen geschaffen für bildschirmorientierte Editoren, Datenbanksysteme mit Bildschirmmasken und nicht zu vergessen die Computerspiele sowie viele andere Anwendungen. Doch damit tauchte bereits ein neues Problem auf, denn die Hersteller (der Terminals) hatten sich, wie so oft in der Vergangenheit, nicht auf einen Standard geeinigt. Daher können verschiedene Terminals nicht nur verschiedene Funktionen ausführen, oftmals verwenden verschiedene Terminals für eine identische Funktion verschiedene Steuercodes.

Beispiel: Direkte Cursoradressierung.

Steuercode	Terminal
(ESC) = (Zeile)(Spalte)	Lear Siegler ADM 3/5/31, Soroc IQ 120, Televideo 912-950, NEC, Osborne I,
(ESC)Y(Zeile)(Spalte)	Heath/Zenith 19, DEC VT 52, ADDS, IBM 3101, Intertube, Superbrain, Teleray
(DC4)(Zeile)(Spalte)	ACT IV/V
(RS)(Zeile)(Spalte)	Datamedia
(ESC)(Zeile)(Spalte)(	Dynabyte 57
(ESC)&a(Zeile)(Spalte)C	HP 2621
(ESC)f(Spalte)(Zeile)	Info-10
(ESC)Y(Zeile)(ESC)X(Spalte)	Perkin Elmer

Tabelle 1. Beispiele der Cursorsteuerung verschiedener Terminals

Bildschirmorientiert arbeitende Programme müssen daher in der Regel (meist vom Anwender) an das eigene Terminal angepaßt werden. Um diesem Umstand weitestgehend abzuhelfen, wurde von FBN Software das Programm SMARTSCREEN entwickelt. Dieses Programm lädt sich direkt unter den CP/M Console Command Processor (CCP) und verändert die Bios-Sprungtabelle so, daß bei der Ausgabe über die Console (Bildschirm) zunächst die jeweiligen Übersetzungsroutinen in SMARTSCREEN angesprungen werden, die dann den ADM-3A-Code in den terminalspezifischen Code übersetzen. Damit sieht es für ein Anwenderprogramm so aus, als würde ein ADM-3A-Terminal ange-

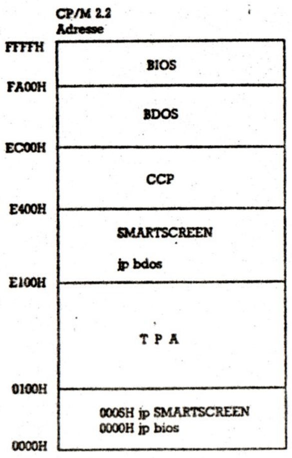


Bild 1. Speicherverteilung bei einem 64k-CP/M-System mit SMARTSCREEN:

steuert. Dadurch, daß SMARTSCREEN direkt unter dem CCP liegt und auch den BDOS-Sprung bei Adresse 0005H verändert, bleibt SMARTSCREEN auch beim Warm-Boot im Speicher aktiv und muß daher nicht vor jedem Programm neu geladen werden. Das hat jedoch zur Folge, daß der Speicherplatz, der vom CCP und von SMARTSCREEN belegt wird, nicht dem Anwenderprogramm zur Verfügung steht. Der Verlust beträgt 2,75 KByte Speicherplatz.

Das »virtuelle« ADM-3A-Terminal erkennt folgende Steuercodes:
↑K, (VT), 0BH: Cursor 1 Pos. nach oben
↑J, (LF), 0AH: Cursor 1 Pos. nach unten
↑H, (BS), 08H: Cursor 1 Pos. nach links
↑L, (FF), 0CH: Cursor 1 Pos. nach rechts
↑↑, (RS), 1EH: Cursor in Zeile 1, Spalte 1
↑Z, (SUB), 1AH: Cursor in Zeile 1, Spalte 1 und Bildschirm löschen
↑M, (CR), 0DH: Cursor an den Zeilenanfang
↑G, (BEL), 07H: Bell (Klingel)

Zur Cursor-Adressierung benutzt das ADM-3A-Terminal folgende Sequenz: (ESC)= in HEX: 1BH, 3DH, gefolgt von der Zeilen- und Spaltenadresse. Die Adresse ist in HEX, begonnen wird in Zeile 1, Spalte 1 mit 20H, 20H. Zeile 1, Spalte 2 wäre also 20H, 21H.

Wie nicht anders zu erwarten, muß man SMARTSCREEN vor der ersten Benutzung an das eigene Terminal anpassen. Das geschieht mit dem mitgelieferten Programm SMSPATCH. Für eine Anzahl von Terminals kann SMSPATCH die notwendigen Änderungen in SMARTSCREEN selbst durchführen. Dies sind:
Microterm ACT IV/ACT V, DEC VT-52, Heath/Zenith H19/Z19, ANSI X3 standard, ADDS Viewpoint, Hazeltine 1500, HP 2621, TRS-80 (P&T), IBM 3101, NEC, Vector Graphic und Superbrain.

Für alle anderen Terminals ist eine spezielle Anpassung notwendig, die aber mit dem SMSPATCH keine Schwierigkeiten macht. SMSPATCH fragt nach der maximalen Spalten- und Zeilenanzahl.

Bei der Cursoradressierung ist beispielsweise folgende Form vorgegeben: (string1) (Adresse1) (string2) (Adresse2) (string3), wobei jeder string aus einer Folge von 0 bis 9 Zeichen bestehen kann. Es kann wahlweise als erste Adresse die Zeilen- oder Spaltenadresse gesendet werden. Der Zeilen- und Spaltenoffset kann ebenfalls getrennt eingegeben werden. Beispiel für eine selektierte Cursor-Adressierungs-Sequenz: (ESC)Y (Spalte + 20H)(Zeile + 20H).

In entsprechender Weise werden die notwendigen Steuerzeichen für Bell (Klingel), Cursor 1 Pos. nach links, Cursor 1 Pos. nach unten, Cursor 1 Pos. nach oben, Cursor 1 Pos. nach rechts, Cursor an Zeilenanfang, Cursor Home und Bildschirm löschen eingegeben. Die Eingaben können dezimal, hexadezimal oder direkt, als Tastendruck, erfolgen.

Da nach Aktivieren von SMARTSCREEN das Programm nicht nur sich selbst, sondern auch den CCP im Speicher vor Überschreiben schützt, ist es nicht mehr notwendig, beim Warmboot den CCP von der Diskette zu laden. Auch das kann man bei der Programmanpassung durch SMS PATCH vorgeben. Bei einigen Computern wird es sogar notwendig sein, ein Nachladen des Systems zu unterbinden. Wird nämlich beim Warmboot auch das BIOS neu geladen (wie beim Superbrain), bleibt SMARTSCREEN ohne diese Vorsichtsmaßnahme natürlich nicht im Speicher.

Nachdem SMARTSCREEN angepaßt ist, kann man die Anpassung mit dem ebenfalls mitgelieferten Programm SMSTEST testen. SMSTEST erstellt mit Hilfe der ADM-3A Steuercodes Testbilder, die mit den schriftlichen Beschreibungen beziehungsweise mit den Abbildungen im Manual übereinstimmen müssen. Hier fehlt allerdings der Hinweis darauf, daß das Testprogramm nur für ein Terminal mit 80 Zeichen pro Zeile und 24 Zeilen geeignet ist. Bei anderen Bildschirmformaten sind die Angaben nicht mehr korrekt. Bei der Benutzung arbeitet SMARTSCREEN so, wie man es von ihm erwartet. Die für ADM-3 adaptierte Version eines bildschirmorientierten Editor funktionierte einwandfrei. Einen Nachteil sollte man aber nicht verschweigen: Das ADM-3-Terminal kennt nur wenige und nicht sehr leistungsfähige Kommandos. Die meisten neueren Terminals können bereits selbständig Zeilen ein- und ausfügen und andere leistungsfähigere Befehle ausführen. Die meisten Programme (wie Wordstar) kennen diese Befehle ebenfalls und können sie mitbenutzen. Der Geschwindigkeitsvorteil, gerade bei seriell angeschlossenen Terminals ist beträchtlich. Die meisten kommerziellen Programme lassen sich außerdem auch an eine große Anzahl weiterer Terminals anpassen, notfalls sogar im Dialog, Kommando für Kommando.

Auch Anpassung an Tastatur und Drucker möglich

Wer diesen (geringen) Aufwand scheut, eine größere Anzahl bildschirmorientierter Programme hat, oder wer gar Programme hat oder benötigt, die sich nicht anpassen lassen und die für das ADM-3A-Terminal ausgelegt sind, für den kann und wird sich dieses Programm lohnen.

Einen ähnlichen Zweck wie SMARTSCREEN verfolgen auch die Programme SMARTKEY und SMARTPRINT. Der Unterschied ist, daß sie nicht die Funktion eines »virtuellen« Terminals, sondern die einer »virtuellen« Tastatur beziehungsweise eines »virtuellen« Druckers haben. Wer hat nicht eine Tastatur, auf der es einige von CP/M nicht, oder falsch genutzte Funktionen gibt? Oder eine Tastatur, die keinen Standard-ASCII-Code ausgibt? Diese Probleme kann (in den meisten Fällen) SMARTKEY lösen. SMARTKEY funktioniert ähnlich wie SMARTSCREEN. Einmal aktiviert, liegt es unterhalb des CCP und fängt die Eingaben über die Tastatur ab, übersetzt sie so, wie vom Benutzer gewünscht und leitet sie weiter. Hiermit lassen sich nicht nur Tasten umdefinieren (man denke nur an die Verwechslung von »Z« und »Y« bei vielen Tastaturen), oder allen Tasten neue Codes zuordnen, sondern es lassen sich einer Taste ganze Strings, wie zum Beispiel »DIR(CR)« zuordnen. Ist das Programm erst einmal aktiv, können die Zuordnungen ständig verändert werden. Das geschieht durch Eingabe eines bestimmten Controll-characters (normalerweise (ESC), läßt sich aber verändern). Damit gelangt man in den SMARTKEY-Input-Modus. Nun betätigt man die Taste, die umdefiniert werden soll und gibt anschließend den zuzuordnenden String ein, abgeschlossen mit einem weiteren (ESC). Die neue Zuordnung wird in einer Tabelle innerhalb des SMARTKEY-Programmes gespeichert und wird ab sofort berücksichtigt.

Beispiel: »↑D« soll durch »DIR (CR)« ersetzt werden.

Eingabe: (ESC)↑D DIR(CR)(ESC) (Das Blank nach dem ersten D wird automatisch von SMARTKEY erzeugt). Wird hiernach ↑D betätigt, so erscheint nicht ↑D, sondern DIR auf dem Bildschirm und der Befehl wird durch das folgende (CR) ausgeführt. Um diese Zuordnung wieder rückgängig zu machen, muß man nur (ESC)↑D (ESC) eingeben, dann wird wieder der Originalcode ausgegeben.

Die weitergehende Verwaltung der Zuordnungstabellen und des Programmes SMARTKEY geschieht durch das Programm FIXKEY. Ist SMARTKEY aktiv und man startet FIXKEY so erscheint folgendes Menü:
COMMAND MENU
0 Exit to CP/M
1 Pack and Save current definitions
2 Load a definition file
3 List contents of a definition file
4 List current definitions
5 Clear current definitions
6 Alter Escape character
7 Terminate SMARTKEY

Dadurch ist es auch möglich, die Zuordnungstabellen auf Diskette abzuspeichern und später direkt als Tabelle zu laden. So kann man mehrere Übersetzungstabellen abspeichern und jeweils die benötigte laden.

SMARTPRINT hat eine entsprechende Übersetzerfunktion für die Ausgabe über die »List«-BIOS-Routine. Hier hat man ebenfalls die Möglichkeit, Drucker, die nicht ASCII-Code-kompatibel (zum Beispiel alte Fernschreib-Druckwerke) sind, entsprechend anzupassen. Die Generierung der Zuord-

nungstabellen geschieht genauso wie bei SMARTKEY. Man kann SMARTPRINT auch dazu benutzen, vom Drucker nicht druckbare Zeichen in äquivalente umzusetzen. Kann ein Drucker beispielsweise die Zeichen »(« und »)« nicht drucken, so kann man sie mit SMARTPRINT durch ».LT.« beziehungsweise »GT« ersetzen lassen. Die Verwaltung der Zuordnungstabellen und des Programmes erfolgt ebenfalls durch FIXKEY. Man kann also auch hier die Zuordnungstabellen permanent auf Diskette speichern.

Die Handhabung dieser Programme wird erst dann kompliziert, wenn man mehrere dieser Utilities gleichzeitig benutzen muß. Beispielsweise SMARTSCREEN und SMARTKEY. Dann ist es wichtig, das Programm, welches länger aktiv bleiben soll, zuerst zu starten.

Bleibt noch anzumerken, daß SMARTSCREEN, SMARTKEY und SMARTPRINT zur Zeit nicht unter CP/M 3.0 und unter Microshell lauffähig sind.

Die in Deutschland ausgelieferten Handbücher sind in englischer Sprache und dem Programm durchaus angemessen. Nachteilig fiel nur auf, daß die Handbücher nicht auf einem Standard-ASCII-Drucker erstellt wurden. Dadurch sind an Stelle der eckigen Klammern beispielsweise deutsche Umlaute abgedruckt, was manchmal etwas verwirrte.

(Andreas Walter)

POWER — ein Nachbrenner für CP/M

Eigentlich wäre dieser Ausdruck falsch, denn dieses Programm ist in Richtung Benutzer dem CP/M vorgeschaltet, aber was die verstärkende Wirkung des Nachbrenners betrifft, ist er richtig. Die Sammlung von Dienstroutinen übernimmt die Programmsteuerung durch ein numeriertes Directory. Sie eignet sich für alle »täglichen« Gegebenheiten. Die — laut Handbuch — 55 Befehle werden hier zusammenfassend besprochen.

Die zuvor angeführten Gegebenheiten kann man in fünf Gruppen einteilen:
1. Datei- und Directoryaktivitäten
2. Diskmanipulationen und Reparaturen
3. Programmschutz
4. Debug-Arbeiten
5. Diskettentest

Zum Punkt Datei und Directoryaktivitäten werden folgende Funktionen ausgeführt:
a) Umbenennen
b) Löschen
c) Wiederherstellen dieser gelöschten Dateien
d) Größe von Dateien feststellen
e) Vergleichen
f) Kopieren zwischen verschiedenen Laufwerken
g) Kopieren zwischen Usernummern
h) Schreibschutz setzen für Dateien
i) Aufheben des Schreibschutzes
j) Dateinamen als systemeigen erklären (nicht sichtbar mit »DIR«)
k) Systemprogrammschutz aufheben
l) Starten von Programmen
m) Starten von bestimmten Speicherbereichen
n) Einloggen von Disketten (Reset)

Zu Punkt 2 — Diskmanipulationen und Reparaturen:
a) Laden von Dateien
b) Wegschreiben von Dateien
c) Laufwerkstatistiken
d) Laufwerkstest
e) CP/M-Gruppen listen für eine Datei
f) Gruppendaten auf dem Bildschirm anzeigen (in ASCII und HEX)
g) Gruppendaten in den Speicher lesen und auf die Disk schreiben
h) Sektor- und Spurdaten auf dem Bildschirm anzeigen (in ASCII und HEX)
i) Sektor- und Spurdaten in den Speicher lesen und auf die Disk schreiben
j) Programme und Daten von der Disk lesen und auf dem Bildschirm anzeigen (in ASCII und HEX)
k) Markieren von Dateinamen zum Beispiel zum gemeinsamen Kopieren (ohne Umbenennung)
l) Sortieren des »DIRECTORIES« nach bestimmten Kriterien

Zu Punkt 3 — Dateischutz:
a) Bestimmen von Dateien, die nur mit Schlüsselworten bearbeitet werden können
b) Dateien als »nur zu lesen« kennzeichnen
c) Lesen und Schreiben erlauben

Zu Punkt 4 — Debug-Arbeiten:
a) Speicherinhalt anzeigen und ersetzen (in HEX, ASCII, BINÄR)
b) Speicherbereiche vergleichen
c) Speicher »dumpen«
d) Speicher mit einem bestimmten Byte füllen
e) Speicherzonen verschieben
f) Bytemuster im Speicher suchen
g) Ausführen von im Speicher befindlichen Maschinencoderoutinen

Zu Punkt 5 — Diskettentest:
a) Testen der Diskette auf fehlerhafte Sektoren, Reparieren dieser Stelle(n) oder Ablegen in einer Sammeldatei, um sie von der Benutzung auszuschließen

Zu diesen Punkten kommen noch zwei weitere hinzu. So gibt es für den Umgang mit POWER noch
a) HELP (Ausdruck von POWER-Befehlen)
b) EXIT (Rückkehr zu CP/M)
c) LOG (Programmparameter)
d) SPEED (Geschwindigkeit, mit der Daten auf dem Bildschirm »geblättert werden)

Vier weitere (USER1 bis USER4) sind vorbereitet, aber noch nicht belegt. Hier können Benutzerwünsche verwirklicht werden.

Der Umfang der obigen Sammlung zeigt, daß sich hier mehrere Programme, die sonst einzeln aufgerufen werden mußten, alle unter einem »Hut« befinden. Dies erleichtert beträchtlich die Arbeit. Man muß sich nur auf die Einarbeitung in ein Programm konzentrieren. So gibt es zum Beispiel für die »wilden Karten« unter CP/M eine vereinfachte Sequenz, statt *.* ist es möglich, nur ** einzugeben, was eine Erleichterung gegenüber der Eingabe »Shift Stern-Punkt-Shift-Stern« darstellt. Einige Befehle sind auch schon durch CP/M und dessen Dienstprogramme bekannt, jedoch wurden sie aus dem Verband (STAT — hier die Befehle DISK, SETDIR, SETSYS, SETRO, SETWR, SIZE, STAT oder PIP — COPY) herausgelöst und einzeln verwendet.

Den großen Pluspunkt stellt jedoch ein Menüsystem dar, daß das Arbeiten mit diesem Programm zu einer wahren Freude für eilige Benutzer macht. Anstelle von achtstelligen Programmnamen wird die Eingabe von nur ein, zwei oder höchstenfalls drei Ziffern benötigt. So wird zum Beispiel ein einfaches, selektiertes Kopieren möglich, ohne wie bereits gesagt, lange Buchstabenkolonnen eintippen zu müssen, das gleiche gilt für den Löschvorgang, das Ausdrucken von Dateien auf dem Bildschirm, wobei einzelne Dateien nur angedruckt werden können und nach Eingabe eines ! Ks (dieses Zeichen kann auf ein gewünschtes geändert werden), die nächste aufgerufen wird. Es gilt weiterhin für den Vergleich mehrerer Dateien, Neubenennung von Dateien, Kennzeichnen von Bits in Dateinamen zum selektierten Kopieren, Schreibschutz von Dateien, Ausschließen von der Darstellung von Dateinamen beim »DIR«-Befehl, Größenfeststellung von Dateien, Transfer zwischen verschiedenen Benutzernummern. In Bild 1 sehen Sie ein Beispiel dieser Menüarchitektur.

Der Programmname XDIR.COM ist mit einem Stern versehen. Dies kennzeichnet eine »nur lese«-Datei. Nun noch ein Beispiel für eine Übertragung von einem Benutzerbereich in einen anderen. Mit dem Befehl XUSER wird der Transfer in einen anderen Bereich eingeleitet (Bild 2).

In dem letzten Beispiel wird noch einmal die Frage gestellt, ob kopiert oder transferiert werden soll. Der Unterschied besteht darin, daß "M" nur den Namen in das jeweilige »DIRECTORY« einträgt, die Datei aber nicht kopiert wird. Bei einer Kopie würde die gesamte Datei auf der Diskette dupliziert werden! Das U hinter »DIR« ist der Code für die Darstellung aller Benutzerbereichseinträge in das »DIRECTORY«. Durch Hinzufügen eines X können auch die Benutzerbereiche auf anderen, bis dahin angesprochenen Disketten-Laufwerken angezeigt werden. Da es für die meisten Befehle Schlüssel dieser Art gibt, ergeben sich mehr als »nur« die oben erwähnten 55 Befehle. Zusätzlich zu diesen gibt es noch eine Datei, die die Programmsteuerung übernimmt. Der Inhalt kann mit einem »LOG«-Befehl sichtbar gemacht werden. Die in dieser Datei vorhandenen Parameter kann man immer ändern, wenn das Power-Zeichen vorhanden ist (A0=):
— so kann hier bestimmt werden, ob beim Kopieren die auf dem Ziellaufwerk vorhandene Datei auf den Typ .BAK umbenannt, überschrieben oder übersprungen werden soll oder die Abfrage der genannten Möglichkeiten in dem Fall selbst durchgeführt werden soll
— in wieviele Spalten seitenweise umgeblättert wird
— ob Original und Backup nach dem Kopieren markiert werden sollen
— kann der Autostart von Power nach Lauf eines (T)ransient (P)rogramm (A)rea Programms an und ausgeschaltet werden
— es können bestimmte Attribute bei Systemprogrammen (R/W, R/O, SYS, DIR) gesetzt werden.

Etliche weitere Hinweise zum »Zuschneidern« des Programms auf persönliche Wünsche findet man in dem recht guten und umfangreichen (etwa 123 Seiten) Handbuch (Kapitel Anpassung = Customization), das mit dem Programm geliefert wird. Dort findet man auch die Erklärung eines Hilfsprogramms, das im Lieferumfang von Power enthalten ist — PWRETOOL. Mit Hilfe dieses Programms ist es möglich, die Befehle umzubenennen, sie als Funktion zu streichen oder sie mit einem Schlüsselwort zu belegen. Zum Beispiel ERA wird in LÖSCHEN umbenannt und mit einem Schlüsselwort versehen. Nun kann nur derjenige ein Programm oder eine Datei löschen, der im Besitz dieses Wortes ist.

Besonders viel Interesse dürfte die genaue Erklärung, wie man eine zerstörte Diskette (BAD SECTOR Meldung) wieder brauchbar macht, finden. Auch Tips, wie man Daten nach einem Systemcrash retten, Dateien für ein schnelleres Laden neu organisiert und wie man Programme modifizieren kann, findet man im Anhang.

Dieses Programm lief auf einem CP/M-System Version 2.2 einwandfrei, auf einem System mit Version 1.41 (Icom-Floppy-Disk-System) erfolgte die Fehlermeldung, daß eine spätere CP/M-Version vonnöten sei. Das Handbuch ist leider in Englisch, das einigen nicht so perfekt »Computerenglisch« sprechenden Lesern einiges Kopfzerbrechen bereiten wird. Störend wirkt sich auch das Fehlen einer Bildschirmsteuerung aus. So werden zum Teil gleiche Informationen über den Bildschirm geschoben, anstatt nur die sich ändernden Daten neu abzubilden.

Dieses umfangreiche Programm ist leicht bedienbar. Es eignet sich daher (bis auf Einschränkungen — siehe vorhergehenden Abschnitt) für den kommerziellen Einsatz, auf den es gut zugeschnitten werden kann. Aber auch der Hobbyist, dem es in erster Linie auf Programmabarbeitung und weniger um Betriebssysteminteresse geht, ist mit dieser Vielfalt von Hilfsprogrammen sicher gut bedient.

(Axel Pohl)

```
A>POWER

        POWER 3.03 on CP/M 2.2

  Copyright (c) 1981, 1982 by PAVEL BREDER
        All rights reserved. 01/03/83

A0=DIR
A0:    1= POWER    .COM |  2= RESOURCE.COM |  3= SDIR     .COM
A0:    4= SURVEY   .COM |  5= SWEEP37 .COM |  6= SPOOL    .COM
A0:    5= XDIR     .COM*
```

Bild 1. Beispiel für die Menüarchitektur

```
A0=XUSER 3
A0-3=COPY
A0:    1= POWER    .COM |  2= RESOURCE.COM |  3= SDIR     .COM
A0:    4= SURVEY   .COM |  5=SWEEP37  .COM |  6= SPOOL    .COM
A0:    5= XDIR     .COM*
select? 6

destination drive :A
(C)opy or (M)ove:M
A0:=A:SPOOL    .COM

A0:=DIR [U]
A0:    1= POWER    .COM |  2= RESOURCE.COM |  3= SDIR     .COM
A0:    4= SURVEY   .COM |  5= SWEEP37 .COM |  6= SPOOL    .COM
A0:    7= XDIR     .COM*
A3:    1= SPOOL    .COM
```

Bild 2. Beispiel für eine Übertragung mit dem Befehl XUSER

DISK DOCTOR

Es kommt alle Nase lang vor, daß man irrtümlich die falsche Datei löscht. »ERA*.BAK« ist ein Beispiel, das wohl jeder CP/M-Benutzer kennt. Bemerkt man den Irrtum sofort nach Ausführung des Befehls, dann kann Disk Doctor das Inhaltsverzeichnis (Directory) der Diskette so verändern, daß die Datei dem Benutzer wieder vollständig zugänglich gemacht wird. Es sollte jedoch bereits an dieser Stelle bemerkt werden, daß durch den Gebrauch von Disk Doctor unter Umständen mehr kaputt gemacht als repariert wird.

Disk Doctor bietet auch für den Fall Hilfestellung, daß man das Löschen erst später bemerkt. In diesem Fall ist es jedoch nicht immer möglich, die ganze Datei zu retten. (Die Gründe hierfür liegen beim CP/M-Betriebssystem. Eine Erklärung würde an dieser Stelle zu weit führen. Man kann sie im Disk-Doctor-Handbuch nachschlagen.) In Disk Doctor ist ein »DIR«-ähnlicher Befehl vorhanden, mit dem auch gelöschte Dateien angezeigt werden können. Anhand dieser Liste kann man sich dann die »hervorzaubernswerten« Dateien aussuchen. Ein anderes Anwendungsgebiet liegt bei defekten Disketten. Hier sind mehrere Fälle zu unterscheiden, die alle von Disk Doctor abgedeckt werden:

— Defekte Sektoren im Directory. Sämtlicher Zugriff auf Dateien wird hierdurch unmöglich gemacht. Disk Doctor kann nun ganze Disketten kopieren, indem einfach Sektor für Sektor kopiert wird. Die defekten Sektoren der zu kopierenden Diskette werden beim Kopieren einfach übersprungen.

— Defekte Sektoren in Dateien. Auch hier kann wieder die gesamte Diskette kopiert werden. Meistens reicht es jedoch aus, die noch lesbaren Sektoren einer Datei auf eine andere Diskette zu kopieren. Auch hierfür steht eine Routine von Disk Doctor bereit.

— Defekte Sektoren in Bereichen, die keiner Datei zugeordnet sind. Beim Anlegen der nächsten Datei könnte ein Fehler gemeldet werden. Disk Doctor durchsucht die ganze Diskette nach defekten Sektoren und erzeugt dann eine Datei, in der diese Sektoren sozusagen gesammelt werden. Falls der Benutzer nun eine neue Datei anlegt, werden die defekten Sektoren von CP/M automatisch übersprungen. In diesem Fall geht kein Byte an Information verloren.

Disk Doctor ist auf verschiedenen 5,25-Zoll-Diskettenformaten und auch auf 8-Zoll-SS-SD-Format erhältlich.

Das Handbuch ist hervorragend — aber nicht aktuell

Das 20seitige englische Handbuch ist — von wenigen sachlichen Fehlern und Schönheitsfehlern abgesehen — hervorragend. Nach einer kleinen Einführung wird die Art, wie das CP/M-Betriebssystem Dateien und Directory handhabt, ausführlich beschrieben. Aber das kann man von der Dokumentation eines Diskettenreparier-Programmes ja wohl erwarten. Anschließend wird das Installieren von Disk Doctor auf das Diskettenformat des Benutzers beschrieben (ein Relikt aus den Zeiten der CP/M-Version 1.4.). Die folgenden Kapitel beschreiben den Aufruf von Disk Doctor und das Arbeiten mit den fünf Routinen des Disk Doctors (genannt »Ward A« bis »Ward E«, siehe »Anwendungsbereiche«). In diesen Kapiteln wird dem Benutzer auch ein tiefergehender Einblick in die Arbeitsweise des CP/M-Betriebssystems gegeben, zumindest was das Verhalten von CP/M im Falle eines Diskettenfehlers betrifft.

Nun zu den Schönheitsfehlern: Die sonst übliche Liste von Dateien auf der »Distribution-Diskette« fehlt. Man muß sich die Dateinamen aus dem Kapitel »Installation« heraussuchen. Aber auch diese Liste ist nicht komplett. In einer Datei namens READ.ME wird erläutert, daß man das Installieren von Disk Doctor auch einem Programm namens INSTALL2.COM (ebenfalls auf der Diskette) überlassen kann. INSTALL2 sucht sich dann die notwendigen Installationsparameter (Anzahl Sektoren je Track, Anzahl Tracks je Diskette) aus den ohnehin schon in CP/M 2.2 vorhandenen Tabellen heraus.

Ferner stehen das Programm HELP.COM und die Datei HELP auf der Diskette. Beide Dateien stellen praktisch ein gekürztes Handbuch auf Diskette dar. (Die Ausgabegeschwindigkeit von HELP.COM ist so gewählt, daß ein altersschwacher Fernschreiber keine Probleme beim Mitprotokollieren bekommt.) Von diesen Dateien steht im Handbuch kein Wort. Mit anderen Worten: Das Handbuch ist seit einiger Zeit nicht mehr überarbeitet worden.

Die gravierendste Unannehmlichkeit sei zuerst genannt: Besitzt man nur ein einziges Diskettenlaufwerk, dann sollte man sich den Disk Doctor erst gar nicht kaufen. Selbst für das Installieren werden zwei Laufwerke benötigt. Abhilfe könnte allerdings ein (irgendwie zu beschaffendes) Programm schaffen, das es gestattet, ein einziges Laufwerk so zu verwenden, als hätte man zwei oder vier (man wird dadurch im wahrsten Sinne des Wortes zum Disk-Jockey). Ein anderer Weg, Disk Doctor mit nur einem Laufwerk zum Laufen zu bringen, besteht darin, im Programmcode herumzufummeln. In diesen Fällen lohnt sich vielleicht der Kauf von Disk Doctor doch noch.

Ein weiterer Kritikpunkt ist die Notwendigkeit, Disk Doctor überhaupt zu installieren. In einem Anhang werden die Parameter für die wichtigsten Formate angegeben. Trotzdem kann der Benutzer Fehler machen, wenn er nicht INSTALL2 verwendet. Außerdem ist man irgendwie auf halbem Weg stehengeblieben: Der Installierungsteil hätte genausogut in DOCTOR.COM eingebaut werden können. (Vor der Verwendung des Formates 8 Zoll, 512-Byte-Sektoren sollte man sich hüten. In diesem Fall meint Disk Doctor, die Diskette habe nur 76 statt 77 Tracks. Der Fehler tritt auch bei Verwendung von INSTALL 2.COM auf.)

Die Behauptung, CP/M speichere die Directory-Einträge einer Datei immer in aufsteigender Reihenfolge ab (siehe Handbuch, letzter Abschnitt Seite 14), ist schlichtweg falsch. Dies ist zwar normalerweise der Fall, es gibt allerdings auch Ausnahmen. (Löschen einer Datei von einem Texteditor aus, während man eine andere Datei bearbeitet.) Diesem Irrtum sind aber auch schon andere Utilities zum Opfer gefallen, hauptsächlich Directory-Sortierprogramme. Infolgedessen verweigert Disk Doctor unter Umständen das »Un-ERAsen« von Dateien, die noch vollkommen intakt sind.

Hat man häufiger mit defekten Disketten zu tun (ohne, daß der Fehler beim Laufwerk liegt), so ist Disk Doctor bestimmt ein gutes Hilfsmittel, Dateien zumindest teilweise zu retten.

Falls die Hauptanwendung von DISK DOCTOR darin liegen sollte, »ERA« rückgängig zu machen oder defekte Sektoren auf der Diskette zu suchen, so ist einem systemerfahrenen Benutzer wahrscheinlich eher mit den kostenlosen — da Public Domain — Programmen DISKUTIL, RECOVER, RECLAIM, UNERASE, FINDBAD und so weiter geholfen. Dem reinen CP/M-Anwender jedoch ist mit Disk Doctor ein kompaktes sowie leicht verständliches und anwendbares Programm gegeben. Es nimmt ihm vielleicht auch ein wenig die Angst, durch eigene Fehler große Dateien zu verlieren.

(Joachim Schäfer)

System-Checker und Diagnostics-II

Computer sind komplizierte Maschinen. Fehler, besonders solche sporadischer Natur, sind schwer aufzuspüren. Unter Umständen aber kann sich der Computer selbst diagnostizieren. Hier werden die beiden System-Testprogramme »Diagnostics-II« und »System-Checker« vorgestellt. System-Checker ist eine abgemagerte Version von »Diagnostics II«.

Sowohl System-Checker als auch Diagnostics-II bestehen aus mehreren Programmen. Jedes einzelne testet einen bestimmten Bereich der Hardware:

— Speicher,
— CPU,
— Disketten-Laufwerke (nicht die Disketten selber),
— Terminal (nur die Ausgabe, nicht die Eingabe) und
— Drucker.

Ich werde nun im folgenden die Art und Weise beschreiben, wie Diagnostics-II die einzelnen Komponenten testet. Die Unterschiede zwischen System-Checker und Diagnostics-II werden am Ende eines jeden Abschnitts aufgeführt.

□ Speicher

Es gibt vier Testarten:

— Der »Quick-Test« schreibt den Speicher voller Nullen oder OFFH-Bytes. Er testet also nur, ob sich jedes Bit setzen und rücksetzen läßt.

— Der »Walking Bit Test« schreibt den zu testenden Speicherbereich voller Nullen und »schiebt« ein einzelnes Bit mit Wert »1« durch den Speicherbereich. Durch diesen Test können Adreßdecodierungsfehler und »Übersprecher« zwischen einzelnen Bits ausfindig gemacht werden.

— Der »Burn-In Test« soll testen, ob dynamische Speicherchips gegen »Einbrennen« von Bits resistent sind.

— Der »Speed Test« soll Auskunft darüber geben, ob der Prozessor mit maximaler Geschwindigkeit arbeitet, oder ob er auf langsame RAMs warten muß.

Der zu testende Speicherbereich läßt sich jeweils angeben.

Mit System-Checker ist nur der »Quick Test« durchführbar.

□ CPU

Dieser Test stellt fest, ob der im Computer verwendete Prozessor ein Z80 oder ein 8080 beziehungsweise 8085 ist. Einige kleinere Testroutinen prüfen, ob der Prozessor die wichtigsten Befehle noch ausführen kann. Auftretende Fehler werden nur bei Diagnostics-II mit recht ausführlichen Fehlermeldungen gemeldet.

System-Checker testet nicht ganz so viele CPU-Befehle wie Diagnostics-II. Auch werden keine Fehlermeldungen abgegeben.

□ Disketten-Laufwerk

Getestet wird, ob sich der Schreib-/Lesekopf einwandfrei positionieren läßt und ob Lese- und/oder Schreiboperationen fehlerfrei ausgeführt werden.

Bei Diagnostics-II kann der Benutzer den Testablauf steuern (zum Beispiel durch Angabe der zu testenden Sektoren). Bei System-Checker hat der Benutzer keine Einflußmöglichkeit.

□ Terminal

Dieser Test »zeigt jedes druckbare Zeichen in jeder Bildschirmposition«. Dies ist der bekannte Test, bei dem die Zeichen auf dem Bildschirm ein diagonales Muster bilden. Falls der Bildschirm es erlaubt, werden direkte Cursor-Positionierung und andere Spezialfunktionen (inverses Schriftbild und so weiter) getestet. Dazu muß das Testprogramm erst installiert werden. Dies wird jedoch vom Testprogramm auf Wunsch durchgeführt und ist im Handbuch sehr ausführlich beschrieben.

Bei dieser Testroutine unterscheiden sich System-Checker und Diagnostic-II nicht.

□ Drucker

Auch hier wird ein Diagonalmuster gedruckt. Besitzer von Typenraddruckern können sogar Vor- und Rückwärtsdruck, hoch- und tiefgestellten Druck und einiges mehr testen.

Auch hier gibt es keinen Unterschied zwischen System-Checker und Diagnostics-II.

Abgesehen von Umfang und Komfort der Testroutinen unterscheiden sich System-Checker und Diagnostics-II auch in der Art des Aufrufes: System-Checker wird mit dem Speichertest-Programm gestartet. Ist der Speicher getestet, wird automatisch das nächste Testprogramm von der Diskette nachgeladen und ausgeführt.

Bei Diagnostics-II muß jede Testroutine einzeln aufgerufen werden. Hierbei ist dann allerdings auch die Angabe der Testparameter als »Command-String« möglich. Ferner besteht die Möglichkeit, alle Testprogramme durch SUBMIT ausführen zu lassen.

Das Diagnostic-II-Handbuch umfaßt 38 Seiten, das System-Checker-Handbuch 18 Seiten sowie ein 5seitiges Glossar, in dem die wichtigsten Begriffe des Computers aufgeführt sind. Beide Handbücher sind in Englisch. Die Testroutinen werden sehr ausführlich beschrieben, und es dürften keine Mißverständnisse aufkommen.

Diagnostics-II bietet neben den oben genannten Testroutinen noch zwei weitere wichtige Programme: Ein Schnelltest für die wichtigsten Systemkomponenten und ein Programm, das beim Justieren des Schreib-/Lesekopfes von Diskettenlaufwerken gute Dienste tut.

Die Diagnostic-II-Testroutinen schreiben auf Wunsch die auftretenden Fehlermeldungen in eine Datei DIAG-LOG. (Sehr angenehm im Zusammenhang mit SUBMINT.)

Falls beim Disktest Fehler auftreten, so kann Diagnostic-II diejenigen Ein-/Ausgabeports oder Speicherstellen »dumpen«, die dem Floppy-Controller zugeordnet sind.

Der Speichertest von Diagnostic-II kann auch Speicherbereiche testen, die in einer anderen 64-K-Bank liegen.

Angaben über den Zeitbedarf der Testroutinen werden zwar bei Diagnostics-II gemacht, sind jedoch viel zu pauschal, um aussagefähig zu sein. (Mein 2,106-MHz-Z80 braucht 112 Sekunden für einen Test, der mit einer Minute veranschlagt ist). Für System-Checker gilt dasselbe.

Falls die Hardware fehlerhaft ist und die Fehler von System-Checker oder Diagnostics-II erkannt werden, so kann der Benutzer mit den System-Checker-Fehlermeldungen (sofern sie überhaupt ausgegeben werden) kaum etwas anfangen. Die Diagnostics-II-Fehlermeldungen lassen schon eher auf die genaue Fehlerursache schließen.

Benutzer, die ihre Hardware für fehlerfrei halten, sich dies aber bestätigen lassen wollen, sind wahrscheinlich mit dem billigeren System-Checker besser bedient. (Obwohl der »Quick-Test« meiner Meinung nach nur sehr grobe Fehler feststellt). Legt man jedoch Wert auf hilfreiche Fehlermeldungen, dann sollte man sich für Diagnostics-II entscheiden.

Zusammenfassung

Diagnostics-II ist wohl ein Leckerbissen für Hobby-Reparateure, die öfter einmal den Mikrocomputer ihrer Bekannten nachsehen müssen. Denn ein einzelner Benutzer kann die Möglichkeiten gar nicht ausreizen.

(Joachim Schäfer)

Extend — ein Programm für Diskettenplatz-Fetischisten

CP/M bietet, in einfacherer Form als MP/M, die Möglichkeit, in verschiedenen, sogenannten »User«-Bereichen zu arbeiten, beziehungsweise das »Directory« in verschiedene Unterteilungen zu gliedern. So kann man zum Beispiel Hilfsprogramme wie PIP, STAT, (Z)SID, Kopierprogramme und so weiter im Benutzerbereich 0 ablegen; Sprachen — wie Basic, Pascal, Pilot und andere mehr im Bereich 1 und Textverwaltung im Bereich 2. Jedoch wird man in allen Bereichen die Programme PIP und eventuell STAT benötigen. Kopiert man diese nun in die »USER«-Bereiche hinein, so belegen sie jedesmal weiteren Speicherplatz auf der Diskette. Diesem Übel hilft Extend ab. »Bitfreaks« werden sicherlich die gefährliche Methode der »Directory«-Manipulation kennen, aber auch diesen erleichtert Extend das Arbeiten.

Extend wird aufgerufen wie eine normale (COM)mand Datei. Statt sich jedoch auf den Anfang der (T)ransient (P)rogramm (A)rea (100H) zu laden, errechnet das Programm seinen Speicherplatz, indem es vom Anfang des (C)onsole (C)ommand (P)rocessors seine Länge subtrahiert. Dort oben lagert es sich ein und verändert anschließend die Einsprungadressen ins CP/M. Außerdem werden die Parameter verändert, die die Länge der TPA angeben. Da die Größe von Extend nur 2,5 KByte beträgt, ist die Verkleinerung zu vernachlässigen. Damit ist das Programm resident geworden. Um zu vermeiden, daß es durch andere Programme, die sich ebenfalls dort oben einlagern (DDT, (Z)SID, DESPOOL, XSUB und die »SMART«-Programme »rausgeschmissen« wird, bedienen Sie sich am besten des »AUTOLOAD«-Features von CP/M. Hierdurch wird direkt nach dem »booten« des Systems Extend geladen. Nachdem die Kontrolle an das CP/M zurückgegeben wurde, meldet sich der CCP jetzt statt mit dem üblichen »A)« mit »0A)«. Wechselt man die »User«-Nummer, ändert sich zum Beispiel »0A)« zu 2A)«. Nun ist es möglich, Programme, welche sich in der User-Nummer 0 befindet, von anderen Bereichen aufzurufen. Eine Einschränkung bildet jedoch die Tatsache, daß dies nur für die Kommandoausführung durch den CCP gilt. Benutzerprogramme können diesen Effekt nicht nutzen, sie können nur Programme oder Overlays aufrufen, die sich in dem jeweiligen Benutzerbereich befinden.

Hier einige Beispiele für das Aufrufen von Programmen:

3A)PIP

PIP wird zuerst in der jetzigen Benutzernummer 3 gesucht, anschließend unter 0.

3B)A:STAT

STAT wird in der Benutzernummer 3 auf dem Laufwerk A: gesucht, und wenn es da nicht gefunden wird, auf Benutzer 0.

3B)ZSID

Hier beginnt die Suche auf Benutzer 3, Laufwerk B:, danach Benutzer 0, Laufwerk B: und als letztes Benutzer 0, Laufwerk A:.

CCP darf bei Warmboot nicht nachgeladen werden

Die Modifikation des CP/Ms durch Extend hat auch noch eine andere Nebenwirkung: Zu einer Warmbootsequenz gehört der Reset der BDOS-Parameter (FCB Adresse, DMA-Adresse und die Rückkehr zu Laufwerk A als Bezugslaufwerk) sowie das Laden des CCP von den Systemspuren der Diskette. Diese letzte Ladefunktion wird durch das Vorhandensein von Extend unterbunden. Man muß hier also bei einigen Programmen vorsichtig sein, die den Speicherplatz vom CCP belegen. Die Gefahr der Benutzung des CCP-RAMs wird jedoch heute im Zeitalter der 48- bis 64-KByte-Computer relativ gering sein. Früher, mit 24 bis 32 KByte war sie jedoch oft notwendig. Um den CCP und Extend dann wieder zu laden, ist ein Systemreset notwendig. Ein Systemcoldboot ist dann auch der notwendige Weg, um »Extend« zu entfernen.

Durch den Einfluß von »Extend« auf den CCP ist es nicht mehr möglich, ein CP/M-Hilfsprogramm — XSUB — zu benutzen. Anstelle dessen gehört eine neue Routine — FBNSUB — zum Lieferumfang. Diese wird in gewohnter Weise wie XSUB aufgerufen, das heißt es wird als erstes Argument in einer Submitdatei aufgeführt.

Da Extend transparent für den Benutzer abläuft, also nicht interaktiv, ist seine Benutzung innerhalb von wenigen Minuten gelernt. Drei Seiten Begleitdokumentation geben eine kurze Einführung in die Funktion und Handhabung des Programms.

(Axel Pohl)

Microshell, für CP/M-Gourmets genau das richtige

Keine Frage, ohne CP/M wäre die Mikrocomputerwelt heute nicht das, was sie ist. CP/M gab den Softwarehäusern die Möglichkeit, Programme für eine breite Masse von Benutzern zu schreiben. Und langsam wurden auch andere Betriebssysteme bekannt — darunter Unix, das hauptsächlich auf Minirechnern läuft. Dann kam man auf die Idee, die Benutzerschnittstelle von Unix auf CP/M zu übertragen. Eine Sammlung von diesen übertragenen Routinen stellt Microshell dar.

Microshell ersetzt den (C)onsole (C)ommand (P)rocessor von CP/M. Es lagert sich unterhalb von dem CCP ein und verändert die Einsprungadresse 0005H und die Größe der zur Verfügung stehenden (T)ransient (P)rogram (A)rea (um 8 KByte reduziert).

Nun zu den Funktionen von Microshell. Man kann sie in drei Gruppen aufteilen:

1. Umleitung von Ausgabedaten (Datensichtgerät und Drucker) auf die Diskette, Eingaben für Programme werden statt vom Datensichtgerät von einer Diskettendatei angefordert (ähnlich CP/Ms SUBMIT- und XSUB-Programmen).
2. »PIPELINE« Verarbeitungsstruktur, das heißt Ausgaben des einen Programms dienen als Eingabeparameter für das nächste. Dessen Ausgaben werden wiederum von einem vierten Programm verarbeitet und so weiter.
3. SHELL- oder COMMAND-Dateien steuern die Abarbeitung von zum Beispiel Dienstprogrammen (Backup von Dateien, die seit dem letzten Backup neu beschrieben wurden, oder Modifizierung von Texten, ohne den Editor oder das Textverarbeitungsprogramm aufzurufen).

Unter Microshell sind solche Kommandos wie DIR, ERA, SAVE, TYPE, REN, USER also alle CP/M-eigenen Kommandos, weiterhin gültig. Hier einige Beispiele für die Umleitung von Daten:

B(0)DIR > > + SAMMELDIR

Das Directory wird zu den bereits in der Datei »SAMMLDIR« stehenden Daten hinzugefügt, dies wird durch die Verdoppelung von »>« erreicht.

B(0)Dir «+ SAMMLDIR

Zu der obigen Funktion der Umleitung kommt hier noch die Darstellung auf dem Bildschirm hinzu. Aber nicht nur in der Umleitung von Datenausgaben zur Diskette für Drucker oder Bildschirm, das zusätzliche Echo auf dem Bildschirm, sondern auch die Anordnung mehrerer Befehle in einer Zeile stellen die Besonderheit von Microshell dar.

Der zweite Punkt, das »pipelining«, kann auf einem CP/M-System nur simuliert werden (es entsteht eine Zwischendatei). Unix arbeitet diese Prozesse parallel ab (ohne Datei).

Als wesentlichstes Feature bietet Microshell eben die SHELL-Verarbeitung. Diese SHELL-DATEIEN (vielleicht besser Programme) führen viele Standardfunktionen in einem wesentlich komplexeren Rahmen aus als SUBMIT/XSUB. So gibt es zum Beispiel die Möglichkeit, Dateien und Programme im gesamten Disksystem suchen zu lassen (dieser Weg wird vom Benutzer über eine Datei bestimmt), SHELL-Dateien lassen sich gegenseitig verknüpfen (nesting), Argumente

ZEICHEN	BEDEUTUNG	BEISPIEL
>	Ausgabenumleitung vom Bildschirm	stat > dateiname
>*	Ausgabenumleitung vom Drucker	pip lst:=Datei >* prtdatei
<	Eingangsumleitung	ed text.txt <korrektur
I	leitet Ausgaben eines Befehls als Eingaben an den nächsten	IDatei 1 IDatei 2 stat *.* I pip lst:=con:
↑	"↑" dieses Zeichen verändert den folgenden Buchstaben zum ASCII-Kontrollzeichen	↑C oder ↑c wird ASCII Wert 03
;	als erstes Zeichen wird es als Kom mentar interpretiert	;so wird kommentiert ;oder so auch
+	Echo der Umleitung erscheint auf dem Bildschirm	Datei 1 I+Datei 2 stat *.* > +Dateiname
—	bei Abfrage des Konsoleingangs-Status ="Zeichen erhalten"	Datei 1 I — Datei 2

ZEICHEN	BEDEUTUNG	BEISPIEL
;	Kommandos in einer Befehlszeile trennen	dir *.com;era *.bak
""	behandelt den Text innerhalb der Anführungszeichen als ein Char.	echo "dies wäre ein argument"
$	gleiche Verarbeitung wie bei SUBMIT/XSUB-Argumentersatz	copieren bak ;aufruf pip b:=a:*.$1 ;datei pip b:=a:*.bak ;ausführen
—u —U	erlaubt ohne Konvertierung die Weitergabe von Kleinbuchstaben	A(0) —U A(0)echo kleinbuchstaben <RETURN> kleinbuchstaben
+v +V	gibt Echo von allen Kommandos vor der Ausführung	A(0) +V löschen test spiel era test.com era spiel.com
—x —X	Rückkehr zu CP/M	A(0) —x A>

Einige Funktionen von Microshell

können beim Aufruf von SHELLs auch nicht belegt werden, MICROSHELL gewährleistet trotzdem einwandfreie Funktion. Unter SUBMIT würde folgendes Beispiel eine Fehlermeldung auslösen:

Submitdatei:

```
    DIR $1.COM
    ERA $1.BAK
    PIP $2.ASM=$3.ASM,B:$4.ASM
```

Befehlszeile:

```
SUBMIT CLEAN MIC MAC MUC
```

Hier fehlt das vierte Argument. Die Fehlermeldung »Datei nicht gefunden« wäre unausweichlich.

Hier gibt es unzählige Möglichkeiten, in einer Art Programmiersprache, die Elemente wie

```
    GOTO
    IF ... THEN
    IFF (F=FILE).. THEN
    RETURN (CONDITION)
```

enthält den Programmfluß zu steuern. So gibt es insgesamt 33 Variblen

```
    numerisch %A - %G
    String %A$ - %G$
```

sowie die von SUBMIT bekannten Übergabeparameter

```
    $0 - $18
```

Hierbei ist $0 für die Shelldatei reserviert. Es können also noch weitere 17 Argumente an die aufgerufene Shelldatei übergeben werden. Zu der Stringfunktion ist noch zu sagen, das eine Anweisung existiert, die ein Löschen der in den Zeichenketten vorhandenen Werte vornimmt (%clear).

In der Tabelle sind einige Funktionen von Microshell aufgelistet.

Die Tabelle ist nicht vollständig, aber sie gibt schon einen Eindruck von der Leistungsfähigkeit von Microshell.

Ein Beispiel, welches die Anwendung solcher Anweisungen wieder spiegelt, ist eine Schleife. Mit einem Texteditor oder über die direkte Eingabe in eine Datei unter Zuhilfenahme von PIPTEST.SUB= CON: (LINEFEED muß nach dem [CR] extra eingegeben werden !) würde wie folgt aufgebaut werden:

```
;Beispiel für die Anweisungen
;Kommentar, %Label, %getchar,
;Terminaldirekteingabe $T,
;numerische und Stringvariable,
;%if .. then ., %goto
;
% Schleife
%getchar %a$ < $'I
%B$=%B$+%A$;%A=%A+1
%if %A=7 then ende
%goto schleife
```

Da die Eingaben in einer Zeile bis zu 84 Zeichen lang sein können, wurde eine recht praktische Editierroutine implementiert. Sie benutzt Wordstar ähnliche Kommandos, was das Erlernen vereinfacht. Es ergibt sich die Möglichkeit vorhergehende Anweisungen ein zweites Mal auszuführen.

Viele der verwandten Anweisungen lassen sich auf Benutzerwünsche anpassen. So kann man statt des Ausdrucks $T (für Terminaldirekteingabe) eigene Wünsche verwirklichen (statt $P vielleicht $DRK, oder für den Aufruf des Editors statt ! ein @). Zu beachten ist nur, daß Einzelzeichen (! oder @) dann für diese Aufgaben reserviert sind. Weiter gibt es Definitionsdateien (zum Beispiel automatische Suchprozedur) worin die erforderlichen Parameter definiert werden (wie viele Laufwerke, Reihenfolge des Suchvorganges), Echo auf den Bildschirm an- und ausschalten, automatisches Einloggen von gewechselten Disketten und so weiter. Aufgezeigt werden diese Möglichkeiten sehr genau in dem mitgelieferten Handbuch. Dort werden die einzelnen Anweisungen auch sehr detailliert besprochen, Tips für die Anpassung gegeben, Fehlermeldungen besprochen und Programmeinsprungadressen angegeben, wenn jemand spezielle Programme für Microshell schreiben möchte. Leider muß hier noch bemerkt werden, daß das Handbuch in Englisch geschrieben ist, was vielen Lesern eine Erschwernis bereiten wird, bis sie endlich die Vorzüge dieses hilfreichen Programms genießen können.

(Axel Pohl)

Computer in den USA

USA, das Land der »hartnäckigen« Computer:
Ein Mann war längere Zeit von zu Hause fort und hatte daher keinen Strom verbraucht.
Der Computer der Elektrizitätsgesellschaft war auf Draht, er schickte dem Mann eine Rechnung über 0,00 Dollar. Sie landete im Papierkorb. Vierzehn Tage später: Wieder die gleiche Rechnung über 0,00 Dollar mit dem Zusatz »überfällig«. Auch sie wurde von dem Empfänger müde lächelnd zerrissen und weggeworfen. Der Computer reagierte sauer. Er schickte vierzehn Tage später nochmals eine Rechnung über 0,00 Dollar und drohte, dem Mann das Licht zu sperren.
Der geplagte Stromverbraucher resignierte und sandte der Elektrizitätsgesellschaft einen Scheck über 0,00 Dollar. Der Computer war versöhnt und schickte dem Mann eine Karte, mit der er höflich für die Zahlung dankte.

»Wieso füttern Sie den CHROMWELL 3X-4 - Computer mit Eiswürfeln; mag er die denn?«
»Scheinbar ja; über Nacht sind sie immer weggeputzt!«

Der Computer ist in die Gewerkschaft eingetreten: Er verlangt zweimal täglich eine Kaffeepause.

Computer-Identifizierung

Der neue Computer steht am Firmeneingang. Er identifiziert morgens bei Arbeitsbeginn die Mitarbeiter optisch durch Erfassen des Gesichtes und gibt dann den Weg frei. Der Firmeninhaber verfügt schon am zweiten Tag nach der Aufstellung des Computers dessen Abschaffung. Am ersten Tag hatte der Computer beim Anblick des Firmenchefs unter »Besondere Bemerkungen« ausgedruckt: »IHR GESICHT IST SO EHRLICH WIE IHRE SPESENABRECHNUNGEN!« Am zweiten Tag schrieb er: »IHR GESICHT IST SO EHRLICH WIE IHRE EINKOMMENSTEUERERKLÄRUNGEN!«

Resource — ein interaktiver Disassembler

CP/M-Disassembler gibt es eigentlich »wie Sand am Meer«: In jedem Debugger ist ein Disassembler-Modul enthalten. Dagegen sind Resource, Disilog und Revas eigenständige Programme, die »nur« disassemblieren können.

Sogenannte Assembler übersetzen in symbolischer Form (LD A, 3AH = Lade das Register A mit der Hexzahl »3A«) geschriebene Maschinenprogramme in Binärzahlen, wie sie vom Prozessor verstanden werden. Disassembler kehren diesen Vorgang um: Aus dem Programmcode, wie er in einem EPROM oder bei einem von Disk oder Kassette geladenen Programm vorliegt, erzeugen sie ein für den Menschen lesbares Programm in Assembler-Sprache. Leider hat dies aber so seine Tücken, denn nicht alles, was in einem Codefile vorkommt, sind Maschinenbefehle — es können zum Beispiel Texte sein, die vom Programm ausgegeben werden, oder irgendwelche Daten, die in einer Tabelle für das Programm griffbereit vorliegen. Disassembler können das von sich aus nicht unterscheiden.

»Gewöhnliche« CP/M-Disassembler verlangen neben dem Namen einer Programmdatei (.COM-FILE) die Angabe von Byte- und Text-Speicherbereichen, bevor der eigentliche Disassembliervorgang stattfindet. Man kann nun nach einem von Murphys Gesetzen sicher sein, daß man mindestens einen solchen Speicherbereich übersehen hat. Ein anderes Gesetz besagt, daß dies erst auffällt, nachdem man das bereits disassemblierte Programm stundenlang kommentiert, verbessert oder sonstwie editiert hat. Es lohnt sich also nicht, von neuem zu disassemblieren, sondern man muß frustriert in einer Tabelle nachsehen, daß die Befehle LD C,A; LD L,E; LD L ODH den Text »Ok.« gefolgt von einem CR darstellen. (Dies Beispiel ist vielleicht ein wenig an den Haaren herbeigezogen, zeigt aber, worum es im Prinzip geht.)

Resource dagegen ist ein »interaktiver« Disassembler: Man kann sich ansehen, was passiert, wenn man einen Speicherbereich versuchsweise als Byte-Speicherbereich definiert anstelle als Programmcode.

Einige weitere Eigenschaften von Resource in Stichworten:

Kommentare können bei Resource bereits während des Disassemblierens eingefügt werden. Symbolnamen anstelle von Adressen werden automatisch erzeugt. Texte, die länger als acht Byte sind, werden erkannt. Man kann sich die Angabe von Text-Bereichen also sparen. Das Disassemblieren kann zu jedem beliebigen Zeitpunkt unterbrochen und später wieder fortgesetzt werden. Man kann nach 16-Bit-Wörtern suchen. Anstelle von Hex-Adressen können auch Symbole angegeben werden.

Steuer-Tabelle

Das »Was wäre, wenn« ist deshalb möglich, weil Resource eine sogenannte Steuertabelle verwendet: Jeder Eintrag dieser Tabelle besteht aus einem Adreßteil und einem Byte, das den Disassembler steuert. Diese Tabelle kann vom Benutzer jederzeit verändert werden.

Zum Beispiel: Die Speicherbereiche 0100-0102, 012C-0147, 014C-017F enthalten Programmcode, der Bereich 0103-012B einen Text, der Bereich 0148-014B zwei Datenwörter. Der Stackbereich des zu untersuchenden Programmes liege von 0180-019F. Adressen ab 01A0 kommen nicht vor.

Dann sieht die zugehörige Steuertabelle folgendermaßen aus:

0100,I Instruktion ab Adresse 0100
0103,B Byte-Speicherbereich (*)
012C,I Instruktionen (*)
0148,W Wort-Speicherbereich
014C,I Instruktionen
0180,S Space, das heißt im Assemblerfile wird: »DEFS 20H« gelistet (20H = 01AO minus 0180)
01AO,E Ende des Programms

Dabei gibt die Adresse immer den Anfang des jeweiligen Bereiches an. Der erste Eintrag ist von Resource vorgegeben, da CP/M-Programme ab Adresse 0100Hex gestartet werden. Ab Adresse 0100 ist demnach mindestens eine Instruktion zu finden. Die beiden mit (*) gekennzeichneten Einträge wurden von Resource erzeugt, da ab Adresse 0103 ein Text steht, der länger als acht Byte ist. Am Ende des Textes »schaltet« Resource wieder auf Programmcode-Modus um. Die Programm-Ende-Markierung erzeugt im Assemblerfile an der entsprechenden Stelle den Pseudo-Befehl »END 0100H«.

Nun kann man zum Beispiel den Eintrag »014C,I« löschen. Zusätzlich zu den beiden Wörtern ab Adresse 0148 werden nun auch alle Bytes zwischen 014C und 017F als Wörter gelistet. Indem man die Steuertabelle gezielt verändert, kann man also ein Assemblerfile erhalten, das von der Struktur her mit dem Original übereinstimmt.

Kommentare und Symbole im Listing

Die Steuer-Tabelle selber macht den Komfort von Resource jedoch nicht allein aus: Mann kan an jede beliebige Stelle des Listings bereits beim Disassemblieren Kommentare einfügen. Diese werden sowohl in der fertigen Assemblerdatei ausgedruckt als auch auf dem Bildschirm ausgegeben. Zum Speichern der Kommentartexte wird ein Speicherbereich verwendet, dessen Beginn der Benutzer selbst festlegen muß.

Mit dem »B«-Befehl wird Resource dazu veranlaßt, eine Standard-symbol-Tabelle zu erzeugen: Jede Adresse, die als Sprungziel, Unterprogramm-Adresse oder als 16-Bit-Konstante in einem Ladebefehl verwendet wird, wird in die Symboltabelle aufgenommen: Adresse xyzt — – Symbol »Lxyzt«. Zum Beispiel: Adresse 04FA — – Zeichenreihe »L04FA«.

Die Symboltabelle kann jederzeit vom Benutzer verändert werden. Das Symbol »L005C« zum Beispiel kann in »FCB1« geändert werden. Ebenso ist folgende Konstruktion möglich:

»L05F1« — »KONVRT«
»L05F9« — »KONVRT«+8

Damit das Disassemblieren zu einem späteren Zeitpunkt fortgesetzt werden kann, müssen die Symbol-, Control- und Kommentar-Tabellen abgespeichert werden. Eine .ASM-Datei braucht noch nicht gespeichert zu werden, da, wie gesagt, der Disassembliervorgang noch nicht beendet ist. Das Erzeugen einer Assemblerdatei ist demnach die letzte Tätigkeit, zu der man Resource veranlaßt.

Es sollte beachtet werden, daß Resource weder eine Warnung, noch sonst einen Hinweis ausgibt, wenn die Tabellen nicht abgespeichert wurden.

Resource-Versionen gibt es für folgende Assemblersprachen:
8080: Standard-Intel-Format
Z80 : Standard-Zilog-Format, TDL-Z80-Format.

Resource wird CP/M-gemäß ab Adresse 0100H geladen, verschiebt sich jedoch nicht unter das BDOS, wie man es von DDT her kennt. Das hat zur Folge, daß man einen sogenannten Offset angeben muß. Diese Offset-Adresse wird als scheinbare »Adresse 0« angesehen. Das heißt im einzelnen: Das zu disassemblierende .COM-File wird ab RAM-Adresse Offset + 100H geladen. Von nun ab ist der Offset weniger wichtig. Gibt man nun den Befehl »L100« oder D1234«, dann wird ab wahrer Adresse Offset + 100H gelistet beziehungsweise ab wahrer Adresse Offset + 1234H gedumpt. Der Offset und Resource selber werden für den Benutzer also »unsichtbar«.

Falls man am Ende einer Sitzung die Assemblerdatei schreiben will, erfolgt die Ausgabe sowohl auf die Datei als auch auf den Bildschirm. Dies ist zeitaufwendig. Deshalb kann man vor jedem Befehl, den man eintippt, den Buchstaben »Q« setzen (Quiet-Modus). Die Bildschirmausgabe wird unterbunden, bis man den nächsten Befehl gibt.

Resource besitzt folgende Anweisungen:

A Suche im angegebenen Bereich nach Texten (»A«SCII)
B Erzeugen der Standard-Symboltabelle (»B«uild«)
C Control-Tabelle ändern
D Dumpen eines Speicherbereichs
E Einfügen eines Symbols in die Symboltabelle
F Finde 16-Bit-Worte
K Symbol löschen (»K«ill)
L Listen eines Speicherbereichs
O Offset angeben
Q Quiet-Modus
R Lesen von Dateien (»R«ead)
S Schreiben der Dateien
; Kommentare einfügen

(Joachim Schäfer)

DPATCH

ist ein bildschirmorientiertes menügesteuertes Hilfsprogramm für Floppy- und Harddisksysteme. Es gestattet die Datensicherung bei Dateien mit I/0-Error, die Weiterbenutzung von Disketten und Hard-Disks mit Hard-Fehlern sowie die Manipulation eines jeden Bytes auf der Diskette oder Platte.

Nach dem Start von Dpatch meldet sich das Programm mit Menü. Man kann nun eine der sechs Funktionen durch Eingabe der entsprechenden Ziffer aufrufen.

1. Disk-Edit-Funktion
2. Datei-Manipulation
3. Reaktivierung einer Datei mit I/O-Error
4. Disk-Oberflächen-Analyse
5. Reaktivieren einer gelöschten Datei
6. Programmende.

Die oberste Statuszeile nach Aufruf der Disk-Edit-Funktion informiert, welcher Sektor von welchem Laufwerk gerade in hexadezimaler und ASCII-Form dargestellt wird, während die letzten beiden Kommandozeilen die möglichen Befehle zur Anwahl eines beliebigen Sektors auf allen angeschlossenen Laufwerken das Lesen, Modifizieren und Zurückschreiben auf die Diskette oder Platte ermöglichen. Mit den Kommandos »IN« und »OUT« kann trackweise fortgeschritten werden.

Die nicht dokumentierten Kommandos der Version 2.2. erlauben es, nach bestimmten Stringfolgen zu suchen (LOCATE, LOCATE NEXT) und einen kompletten Sektor in einem von acht »Sektor-Registern« zwischenzuspeichern (PUT) und von dort bei Bedarf in einen anderen Sektor zurückzuladen (GET).

Mit CNTRL-P kann der Bildschirminhalt jederzeit auf einem angeschlossenen Drucker ausgegeben werden.

Als nachteilig wurde empfunden, daß im Disk-Edit-Mode mit den Kommandos +Sektor und —Sektor jeweils der physikalische Sektor um ± 1 verändert und dargestellt wird und nicht unter Berücksichtigung des »Skew-Faktors« die logischen Sektoren. Ein gezieltes Suchen, zum Beispiel durch den Bereich des Inhaltsverzeichnisses, wird dadurch erschwert, daß immer sechs Sektoren weitergeschaltet werden muß.

Bei der direkten Dateimanipulation besteht dieser Nachteil jedoch nicht. Dieses Programmsegment arbeitet ähnlich wie die Disk-Edit-Funktion, nur muß hier zu Beginn der Name der zu analysierenden Datei eingegeben werden. Im Gegensatz zur Disk-Edit-Funktion, mit der sich jedes Byte auf einer Diskette oder Platte manipulieren läßt, ist dies mittels der direkten Dateimanipulation nur innerhalb der gewählten Datei möglich.

Dateien, die aufgrund eines I/O-Errors nicht mehr lesbar sind, können unter Umständen wieder korrigiert werden. Zu diesem Zweck wird die Datei umkopiert und bei Erreichen des fehlerhaften Sektors der Lesezyklus so lange wiederholt, bis er entweder erfolgreich abgeschlossen oder vom Anwender abgebrochen wird. Als Erfahrungswert wird im Handbuch angegeben, daß nach 40 bis 50 Versuchen die Funktion abgebrochen werden sollte. Bei den Leseversuchen wird die Spur, auf der sich der fehlerhafte Sektor befindet, aus verschiedenen Richtungen angefahren, in der Hoffnung, so eventuell ein gewisses Spiel in der Mechanik auszugleichen und den Kopf exakt so zu positionieren wie beim Schreibvorgang.

Mittels der »Oberflächenanalyse« werden BAD RECORDS aufgespürt, in einer Datei »BADBLKS« zusammengefaßt und unter der User-Nummer 31 abgelegt, auf die unter Standard-CP/M-Kommandos nicht zugegriffen werden kann. Zwei Analysefunktionen können angewählt werden:

1. Zerstörungsfreie Disk-Analyse
2. Test-Pattern-Analyse

Während bei 1. die Diskette nur gescannt und Fehler aufgespürt werden, werden bei 2. verschiedene Bitmuster geschrieben, gelesen und verglichen. Hierbei werden sämtliche Informationen auf der Diskette zerstört. Dpatch arbeitet in allen Funktionen fehlerfrei. Die interaktive menügesteuerte Bedienung ist, insbesondere aufgrund der guten Dokumentation mit umfangreichen Fehlermeldungen, im Klartext schnell zu beherrschen. Für den ambitionierten Anwender ist Dpatch sicherlich ein wertvolles Hilfsmittel.

(Rainer Markgraf)

Peter Wollschlaeger

TRS-80/Modell 4:

Kleine Zipperlein rasch kuriert

Der TRS-80/Modell 4 wird serienmäßig mit TRSDOS 6.1 und einem erweiterten MBasic geliefert. Dieser Software ist durchaus Lob zu zollen. Doch es gibt nichts, was man nicht besser machen kann, und so ein paar kleine Zipperlein, die im Programmiereralltag manchmal ganz schön zwicken, sind auch hier vorhanden. Lesen Sie, wie man sie kuriert und einige schlummernde Eigenschaften der Software zum Leben erweckt.

Vorab, wir greifen mit Peek und Poke in das Betriebssystem ein. Einige der Adressen sind nicht offiziell dokumentiert, andere wurden dem technischen Manual [1] entnommen, das aber in Deutschland (noch) nicht erhältlich ist. Theoretisch besteht also die Gefahr, daß sich mit einer neuen Version die inoffiziellen Adressen ändern. Die neuen dann wieder aufzuspüren, dürfte aber für findige mc-Leser kein Problem sein. Auch in [1] sind keine absoluten Adressen zu finden. Dokumentiert sind sogenannte SVCs (*Supervisor Calls*) – ein neuer Name für einen alten (CP/M-)Hut. Sprich, DOS-Routinen werden über eine Funktionsnummer (im A-Register) immer mit RST 28H aufgerufen. Ein SVC (65H) erbringt im IY-Register die Basisadresse einer Tabelle, in der diverse Systemvariable gehalten werden. *Bild 1* zeigt die zwei Assembleranweisungen, die dafür erforderlich sind. Demonstriert werden soll aber etwas anderes: Früher hätte man einfach, wenn man einen Wert von der USR-Routine an Basic zurückgeben wollte, geschrieben: LD HL,WERT, JP CINT. Jetzt ist für CINT nur noch ein Vektor dokumentiert (2605H); der Wert muß aber immer noch in HL stehen. Daher die Lösung ab Zeile 8. Nämlich: Wert auf den Stack, HL mit Adresse von CINT (aus Vektor) laden und "EX (SP),HL".

Danach steht der Wert in HL und die Adresse von CINT auf dem Stack, das folgende RET wirkt als indirekter Sprung zu CINT.

In *Bild 2*, Zeilen 100...130, finden Sie das Programm noch einmal, diesmal in einen Basic-Lader eingebunden. Danach steht die gesuchte Adresse in der Variablen BASE. BASE+10 zeigt auf die Versionsnummer, in diesem Byte steht 61H für „6.1". In Zeile 150/160 finden Sie die praktische Anwendung.

Wichtiger ist schon das K-Flag (*Keyboard*) in BASE+10. Interessant sind Bit 7 (1, wenn noch ein Zeichen im „Type Ahead Buffer"), Bit 5 (1, wenn „CAPS locked") und Bit 0 (Break gedrückt).

```
                     00001 ;USR-ROUTINE. ERGIBT BASIS DER SYSTEM-FLAGS-TABELLE
                     00002
0000 3E65            00003        LD      A,65H          ;SVC TO GET FLAG-
0002 EF              00004        RST     28H            ;TABLE BASE -> IY
0003 FDE5            00005        PUSH    IY             ;VAL TO RETURN
0005 2A0526          00006        LD      HL,(2605H)     ;ADR CINT-ROUTINE
0008 E3              00007        EX      (SP),HL        ;-> STACK
0009 C9              00008        RET                    ;VIA CINT -> BASIC
0000                 00009        END
00000 TOTAL ERRORS
34831  TEXT AREA BYTES LEFT
```

Bild 1. So gibt eine USR-Routine ein Ergebnis an Basic zurück

CAPS-Taste vom Programm bedient

Wesentlich erscheint dem Autor Bit 5. Da in String-Variablen sauber zwischen Groß- und Kleinschrift unterschieden wird, ist man gezwungen, Antworten auf Fragen wie „Ja oder Nein (J/N)?" so zu prüfen: IF A$="J" OR A$="j" THEN... ELSE IF A$="N" OR A$="n" THEN... Läßt man in der Antwort viele Buchstaben und andere Zeichen zu, wird auch das Testen via ASC mühsam. Deshalb ab Zeile 180 die Demonstration, wie man mit Hilfe von „Bit 5" die CAPS-Taste im Programm abfragen und betätigen kann.

Cursor in allen Varianten

Viele Möglichkeiten eröffnet das Video-Flag. Vorab (Zeilen 350...510) einige Spiele mit der Uhr. Das Stellen wäre zwar auch über einen SYSTEM-Aufruf möglich, nur muß dazu eine Systemdiskette eingelegt sein.

Mögen Sie lieber einen blinkenden oder stehenden Cursor, hätten Sie ihn gern etwas größer oder kleiner oder ganz unsichtbar?

Möchten Sie das alles mischen, um in Masken bestimmte Stellen herauszuheben? Kein Problem, in den Zeilen 520...690 werden alle Varianten vorgeführt.

Hardcopy mit einem Befehl

Das Modell 4 erlaubt es, im Direkt-Modus mit „Control Shift Doppelpunkt" den Schirminhalt auf den Drucker zu kopieren. Per Programm geht das auch, Sie müssen nur die BS-Routine kennen (siehe Zeile 730). Sollte es Sie dann stören, daß anstatt Ihrer mühsam erzeugten Grafikzeichen nur schnöde Punkte gedruckt werden, dann hilft von Basic aus: SYSTEM"SYSTEM(GRAPHICS)".

Zeile 740 zeigt, wie man die Break-Taste aus- und einschalten kann, und Zeile 750, ein CLS mit Ready-Meldung, man weiß dann wenigstens, was Sache ist. Scroll-Protect von maximal sieben Zeilen ist entgegen anderslautenden Gerüchten auch beim Modell 4 möglich. Offiziell läuft das über SVC 0FH, wobei vor dem Aufruf das B-Register mit 7 und C mit der Zeilenzahl zu laden ist. Ist Ihnen das zu mühsam, so geht das auch mit einem Poke-Befehl, wie in Zeile 800 gezeigt. Der etwas umständliche Ausdruck dort ist die Kopie der BS-Routine: LD A,C:AND 7:LD C,A:LD A,(0BFB): AND F8:OR C:LD (0BFB),A.

LIST-Befehl desaktiviert oder geheim

Man kann zwar ein Programm über die P-Option schützen, nur hat die den Nachteil, daß jeder Hacker weiß, wie man sie ausschalten kann. Wenn Sie sich nach den Vorschlägen ab Zeile 820 Ihre Spezialversion schaffen, wird es etwas schwieriger. Wenn Sie ganz sicher gehen wollen, müssen Sie allerdings diese mc-Ausgabe komplett aufkaufen.

Basic-Programm gerettet

Nach einem versehentlichen Ausflug ins DOS (bei SYSTEM Textstring vergessen) kommen Sie leider nicht mehr ins Basic zurück, ohne das Programm dort zu zerstören. Abhilfe bringt die Routine in *Bild 3*. Der Basic-Teil generiert einen Cmd-File, so daß Sie dann nur noch BOLD tippen müssen, und die Welt ist wieder in Ordnung. Notfalls geht es auch so: Tippen Sie im DOS: MEMORY-(GO=X'63DD').

Sie kommen dann auch ins Basic, finden auch Ihr Programm intakt vor, nur alle Zeiger sind verbogen, und FRE(0) meldet einen freien Speicher von 8.054 Bytes, also 3 Bytes und noch ein halbes Bit oder so. In diesem Falle sollten Sie Ihr Programm sichern und dann Basic neu laden.

Zeilen duplizieren geht doch

Die kleine Routine in *Bild 4* kostet kein einziges Byte, sie ändert nur den Editor. Der K-Befehl ist bekanntlich redundant (mit Delete geht es auch), weshalb ihm die Bedeutung KOPIE zugewiesen wurde. Wollen Sie eine Zeile duplizieren, so holen Sie sie in den Edit-Mode und tippen K. Es erscheint ein „?" als Aufforderung, die Nummer der neuen Zeile einzugeben. Sobald das erfolgt ist, wird die neue Zeile im Edit-Mode angezeigt. Sie können dann <Enter> geben oder die Zeile gleich editieren, ganz original kopiert man ja meistens nicht.

Deutsche Sonderzeichen auch gedruckt

Sollten Sie auch zu geizig gewesen sein, sich den Tandy-Drucker zu kaufen und sich statt dessen mit einem Epson oder ähnlichem beschieden, dann haben Sie ein Problem. Die deutschen Zeichen erscheinen zwar auf dem Schirm, bringen aber auf dem Drucker nur Nonsens. Abhilfe schafft das Programm in *Bild 5*. Das ist ein Basic-Lader einmal anders. Die

```
1 ' Tricks und Tips fur den TRS-80 Model 4
2 '
3 ' (C) 1984 Peter Wollschlaeger, Hildesheim
4 '
100 'Die folgende Routine ermittelt die Basis der System-Flags-Tabelle
110 DATA 3E,65,EF,FD,E5,2A,05,26,E3,C9
120 FOR I=0 TO 9:READ A$:POKE &H5FB5+I,VAL("&H"+A$):NEXT
130 DEF USR=&H5FB5:BASE=USR(0)
140 '-------------------------------------------------------------------------
150 V$= HEX$(PEEK(BASE+27))
160 PRINT"Sie arbeiten unter TRSDOS Version ";LEFT$(V$,1)"."RIGHT$(V$,1)
170 '-------------------------------------------------------------------------
180 'CAPS-Taste
190 '==========
200 KF=BASE+10: 'Keyboard Status Flag
210 ' Status der CAPS-Taste testen:
220 '-----------------------------
230 IF PEEK(KF) AND 32 THEN PRINT"CAPS on Uppercase"
   ELSE PRINT"CAPS on Lowercase"
240 '
250 'CAPS-Taste betätigen:
260 '---------------------
270 POKE KF,PEEK(KF) XOR 32
280 'Uppercase erzwingen:
290 '-------------------
300 POKE KF,PEEK(KF) OR 32
310 'Lowercase erzwingen
320 '-------------------
330 POKE KF,PEEK(KF) AND 223
340 '
350 'Echtzeit-Uhr auf dem Bildschirm
360 '===============================
370 VF=BASE+21:' Video Status Flag
380 'testen ob Uhr sichtbar oder nicht
390 '--------------------------------
400 IF PEEK(VF) AND 16 THEN PRINT"Uhr an" ELSE PRINT"Uhr aus"
410 'Uhr einschalten:
420 '----------------
430 POKE VF,PEEK(VF) OR 16
440 'Uhr ausschalten:
450 '----------------
460 POKE VF,PEEK(VF) AND 239
470 'Uhr stellen
480 '-----------
490 INPUT"Uhrzeit in h,m,s ";H,M,S
500 POKE 45,S:POKE 46,M:POKE 47,H
510 PRINT"Ihre Zeiteingabe: ";TIME$
520 'Cursor aendern
530 '==============
540 CSA=&HBFF:'Adresse des Cursor-Symbols
550 VF=BASE+21:  'Video Status Flag
560 'Cursor-Symbol testen/aendern
570 '----------------------------
580 PRINT"Cursor hat z.Z. Code ";PEEK(CSA),HEX$(PEEK(CSA));"H"
590 INPUT"Code neues Symbol (hex mit &Hnn) ";C: POKE CSA,C
600 REM Code 32 oder 128 macht den Cursor unsichtbar
610 'testen ob Cursor blinkt
620 '----------------------
630 IF PEEK(VF) AND 64 THEN PRINT"Cursor steht" ELSE PRINT"Cursor blinkt"
640 'Cursor auf stehend schalten:
650 '----------------------------
660 POKE VF,PEEK(VF) OR 64
670 'Cursor auf blinkend schalten:
680 '-----------------------------
690 POKE VF,PEEK(VF) AND 191
700 'Verschiedene Tricks
710 '===================
720 X=&H3A0:CALL X:' erzeugt ein "Klick"
730 X=&HABA:CALL X:' Hardcopy Schirm -> Drucker
740 B=BASE+18:POKE B,PEEK(B) XOR 16:' Schaltet Break-Taste aus bzw. wieder an
750 X=&H559:CALL X:' Wie CLS, aber mit Ready-Prompt
760 '-------------------------------------------------------------------------
770 ' Die folgende Routine erlaubt, max. 7 Zeilen auf dem Schirm zu schützen
780 ' (Kein Scrollen dieser Zeilen)
790 INPUT"Wieviele Zeilen schützen (max 7)";Z
800 POKE &HBFB,(PEEK(&HBFB) AND &HF8) OR (Z AND 7)
810 '-------------------------------------------------------------------------
820 'Der ausgeschaltete oder geheime LIST-Befehl
830 '===========================================
840 'in den Adressen &H599A  &H599B   &H599C    &H599D
850 'steht  ASC von     "I"     "S"    "T"+128   &H93=Token LIST
860 '
870 'mit POKE &H599A,ASC("A") wird aus LIST LAST
880 'mit POKE &H599D,129 wird aus LIST END
890 'mit POKE &H599D,140 wird aus LIST NEW
900 '-------------------------------------------------------------------------
910 'Wenn es Sie auch stört, daß im EDIT die ESC-Taste (Shift ^) schlecht
920 'zu bedienen ist, dann legen Sie doch die Funktion auf F1, und zwar so:
930 '                         POKE &H3E6E,129
940 '-------------------------------------------------------------------------
```

Bild 2. Eine Sammlung von Tricks: Eingriffe in das Betriebssystem über Peek und Poke

```
           00010 ;RE-ENTER BASIC FROM TRSDOS 6.1       TRS-80, M 4
           00020
           00030 ;(C) 1983/12 PETER WOLLSCHLAEGER, HILDESHEIM
           00040
6000       00100         ORG     6000H         ;MITTEN IM I/O-BUFFER
6000 CDEC7D 00110 START  CALL    7DECH         ;LOAD BASIC/OV1
6003 CD8642 00120        CALL    4286H         ;INIT POINTER
6006 C3DD63 00130        JP      63DDH         ;READY-PROMPT
6000       00140         END     START
00000 TOTAL ERRORS
34818  TEXT AREA BYTES LEFT
10 'Programm generiert einen Cmd-File. Name: BOLD/CMD
20 'Zweck: Ruckkehr von TRSDOS (über SYSTEM erreicht) zu BASIC.
25 'Aufruf: im TRSDOS: BOLD
30 '
40 DATA CD,EC,7D,CD,86,42,C3,DD,63
50 FOR I=0 TO 8:READ A$:POKE &H6000+I,VAL("&H"+A$):NEXT
60 SYSTEM"DUMP BOLD/CMD:0 (S=X'6000',E=X'6008',TRA=X'6000')"
```

Bild 3. So kann man nach einem Ausflug ins DOS sein Basic-Programm retten

```
            00001 :ZDUP        ZEILEN DUPLIZIEREN
            00002
            00003 :DAS EDIT K-KOMMANDO HEISST JETZT KOPIE
            00004
            00005 :(C) 1984/3 PETER WOLLSCHLAEGER, HILDESHEIM
            00006
3DE0        00010         ORG     3DE0H        ;PATCH K-CMD
3DE0 3E3F   00020         LD      A,'?'
3DE2 CDAF40 00030         CALL    40AFH        ;PRINT '?'
            00033                              ;INPUT NEW LINE-#
3DE5 21005F 00040         LD      HL,5F00H     ;USED AS INP BUFFER
3DE8 010005 00050         LD      BC,0500H     ;ALLOW MAX 5 CHAR.
3DEB 3E09   00070         LD      A,9          ;SVC LINE INPUT
3DED EF     00080         RST     28H          ;CALL SVC 9
3DEE 3E60   00090         LD      A,60H        ;SVC DECHEX CONVERSION
3DF0 EF     00100         RST     28H          ;RESULT IN BC
3DF1 ED430E61 00110       LD      (610EH),BC   ;STORE IN EDIT LINE
3DF5 F1     00120         POP     AF           ;KILL OLD LINE-#
3DF6 C5     00130         PUSH    BC           ;NEW LINE-# -> STACK
3DF7 C33D3D 00140         JP      3D3DH        ;EDIT MAIN LOOP
0000        00150         END
00000 TOTAL ERRORS
34536  TEXT AREA BYTES LEFT
1 'BASIC-Lader ZDUP
10 DATA 3E,3F,CD,AF,40,21,00,5F,01,00,05,3E,09,EF,3E,60,EF,ED,43,0E,61,
     F1,C5,C3,3D,3D
20 FOR I=0 TO 25:READ A$:POKE &H3DE0+I,VAL("&H"+A$):NEXT
```

Bild 4. Den Editor verändert: Jetzt ist das Duplizieren von Zeilen möglich

Assemblerzeilen erscheinen als Kommentare („'" heißt REM). Diese Art der Darstellung ist hier angebracht. Sie sehen ab Zeile 290 die Umsetztabelle, wobei immer das erste Zeichen (in hex.) der Modell-4-Code, das zweite der Code des Druckers ist. Sie können damit jeden Drucker anpassen. Sollten Sie mehr oder weniger als sieben Zeichen konvertieren wollen, müssen Sie die Zahl Sieben in Zeile 160 ändern.

Das Programm „patcht" direkt den Druckertreiber, obwohl das die ideale Anwendung für ein Filterprogramm wäre. Nur ist es mir leider nicht gelungen, ein solches zum Laufen zu bringen, auch nicht, als ich ein Beispiel aus [1] vom Header bis zur Verschiebetabelle so weit zeichengetreu abgetippt habe, wie das mit einem fremden Assembler möglich ist (der Modell-4-Assembler ist noch nicht lieferbar). Auf jeden Fall muß ich einen intelligenten Fehler gemacht haben, denn die sonst so cleveren Experten bei Tandy-Deutschland konnten das Problem auch nicht lösen und haben es an die Zentrale weitergegeben. Der Autor hofft, in mc vom Ergebnis noch berichten zu können.

```
100 '                          ORG    0BECH        ;PATCH KB-DRIVER
110 DATA C3,80,02:'            JP     START
120 '                          ORG    0280H
130 DATA E5       :'  START    PUSH   HL
140 DATA C5       :'           PUSH   BC
150 DATA 21,98,02:'            LD     HL,TABLE-1
160 DATA 06,07    :'           LD     B,7          ;TAB LEN
170 DATA 79       :'           LD     A,C          ;CHAR TO TEST
180 DATA 23       :'  LOOP     INC    HL           ;NEXT CHAR TO TEST
190 DATA BE       :'           CP     (HL)         ;IN TABLE?
195 DATA 23       :'           INC    HL           ;POINT TO NEW CHAR
200 DATA 28,09    :'           JR     Z,FOUND      ;IF
210 DATA 10,F9    :'           DJNZ   LOOP         ;ELSE KEEP LOOKING
220 DATA C1       :'  EXIT     POP    BC
230 DATA E1       :'           POP    HL
240 DATA D3,F8    :'           OUT    (0F8H),A     ;LPRINT CHAR
250 DATA 79       :'           LD     A,C
260 DATA BF       :'           CP     A            ;KEEP TRSDOS HAPPY
270 DATA C9       :'           RET
275 DATA 7E       :'  FOUND    LD     A,(HL)       ;REPLACE
276 DATA 18,F6    :'           JR     EXIT
280 '
290 DATA D1,5B    :'  TABLE    DEFW   5BD1H        ;Ä
300 DATA D2,5C    :'           DEFW   5CD2H        ;Ö
310 DATA D3,5D    :'           DEFW   5DD3H        ;Ü
320 DATA D6,7B    :'           DEFW   7BD6H        ;ä
330 DATA D7,7C    :'           DEFW   7CD7H        ;ö
340 DATA D8,7D    :'           DEFW   7DD8H        ;ü
350 DATA D9,7E    :'           DEFW   7ED9H        ;ß
360 '                          END
400 FOR I=0 TO 2  :READ A$:POKE &HEBC+I,VAL("&H"+A$):NEXT
410 FOR I=0 TO 38 :READ A$:POKE &H280+I,VAL("&H"+A$):NEXT
```

Bild 5. Es muß nicht der Tandy-Drucker sein. Mit diesem Patch kann alles angeschlossen werden, was einen deutschen Zeichensatz hat

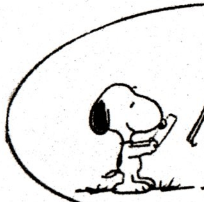
Wie Ihnen bei allen Programmen aufgefallen sein dürfte, wurde nicht mit Mem-Size ein Speicherbereich für die Assemblerroutinen geschützt. Nun, der Editor-Patch braucht das nicht, die Routine von Bild 1 nutzt den I/O-Buffer, und die letzte Routine schließlich tummelt sich in Seite 2. Dort werden die Parameter für die „Geräte" KI DO usw. eingetragen, und zwar bis 0237H. Jedes „User-Device" belegt acht Bytes, folglich ist mit dem ORG auf 0280H noch Platz für neun davon gelassen worden. Meines Erachtens kann man auch noch Teile der Seite 3 benutzen, wieviel, ist reine Nervensache. Das Problem: Ab 0380H wächst Ihnen ein Stack entgegen, der allerdings nicht allzu tief werden dürfte. Normalerweise liegt nämlich der Stack „hoch oben", der untere wird nur benutzt, solange der obere infolge Bank-Switching nicht greifbar ist.
Zum Schluß sei noch erwähnt, wie ohne Modell-4-Assembler die Listings erstellt wurden. Ich habe das Modell 4 im Modell-3-Mode unter NEWDOS gefahren und den NEWDOS-Assembler benutzt. Dann wurden die Objekt-Files auf eine TRSDOS-1.3-Diskette kopiert und von da (wieder im Model-4-Mode unter TRSDOS 6.1) mittels CONV herübergezogen. Damit ist bewiesen, daß sich die TRS-80-Modelle 1, 3 und 4 im Objektcode-Format (Lader) nicht unterscheiden.

Literatur

[1] Model 4 technical reference manual. Tandy Corporation 1983.
[2] *Röckrath, Luidger:* Der geknackte TRS-80. mc 1981, Heft 1, Seite 46.

IN KURZE

nächster Redaktionsschluß ist der 28. April 1986



Programmbibliothek

Die Programmbibliothek erscheint diesmal wieder als unsere jährliche Sonderausgabe und liegt diesem 12. INFO bei.

Programmkurzanleitungen

Endlich ist es soweit!
Die erste Resonanz auf Eure Kurzanleitungseinsendungen wird ein Grafiksonderheft werden.
Ob es mir möglich ist, diese Kurzanleitungssonderausgabe dieser INFO mit beizulegen, weiß ich noch nicht. Ihr werdet es aber spätestens beim Clubtreffen erhalten. Wer am Clubtreffen nicht teilnehmen kann bekommt es mit dem 13. INFO zugesandt.

CLUB 80 HAAR

Es haben:

Video Genie

Alber Herbert
Albers Herbert
Boecker Dieter
Froehlich Burghard
Grajewski Werner
Held Manfred
Hummel Anton
Jablotschkin Rainer
Kasper Dieter
May Holger
Preuss Lothar
Rank Heinrich
Schaefer Walter
Schroeder Gerald (II s)
Sopp Arnulf (III s)
Spiess Peter
Stephan Hans-Martin
Trapp Harald
Voigtlaender Holm
Volz Oliver (II s)
Weiss Dieter
Wucherer Juergen

HRG

Boeckling Ulrich
Dreyer Gerald
Drowaelder Bernd
Jablotschkin Rainer
Muehlenbein Klaus Juergen
Neueder Jens
Schneider Manfred
Sopp Arnulf
Spiess Peter
Weiss Dieter

Epson - Drucker

Beckhausen Wolfgang
Dreyer Gerald
Hermann Klaus
Koenig Hans J.
Muehlenbein Klaus Juergen
Piller Walter
Smerling Frank
Stevens Peter
Troesch Eberhard
Voigtlaender Holm

TRS-80 Mod. 1

Boeckling Ulrich
Koenig Hans J.
Konrad Josef
Kuhn Eckehard
Muehlenbein Klaus Juergen
Neueder Jens
Obermann Hartmut
Perschbach Patrick
Piller Walter
Raggan Hans
Retzlaff Bernd
Schneider Manfred
Wagner Alexander

TRS-80 Mod. 3

Buskowiak Thomas
Schrewe Christian
Smerling Frank
Stevens Peter
Troesch Eberhard
Wagner Guenther
Spiess Peter

TRS-80 Mod. 4

Dreyer Gerald
Hermann Klaus
Stevens Peter
~~Spiess Peter~~

Komtek

Beckhausen Wolfgang
Zwickel Walter

Star - Drucker

Albers Herbert
Grajewski Werner
Jablotschkin Rainer
May Holger
Schaefer Walter
Schroeder Gerald
Weiss Dieter
Wucherer Juergen
Zwickel Walter

* DATENUEBERTRAGUNG PER FERNSEHBILD *

(C by CORNER BOX)

Der WDR-Computer-Club plant eine neue Art der Datenübertragung. Statt über den Tonkanal werden die Informationen mit dem Fernsehbild übertragen. Die Übertragung per Tonkanal hat folgende Nachteile:
1. Durch die aufwendige Anpassung des Cassetten-Recorders ist die Übertragung sehr unsicher.
2. Während der Übertragung ist der Tonkanal blockiert.

Das WDR CC-VIDEODAT-SYSTEM erlaubt es nun, während der ganzen Sendezeit des Computer-Clubs parallel Daten zu übertragen. Diese können auch vom Video-Recorder in den Rechner geladen werden. Es ist möglich, in 30 Minuten ca.26 DIN-A4 (das sind 50.000 Buchstaben = 50 K) zu übertragen. Gesendet werden Informationen, Programme in Ascii-Code, Basicode, Pascal und Schaltbilder.

Das Verfahren:

Im Fersehbild werden die Zeilen 25 bis 36 entsprechend dem Ascii-Code hell getastet. So kann bei jedem 2. Halbbild ein Datenbild übertragen werden.

Der Anschluß:

Die Scart Buchse oder die AV-Buchse des Fernsehers oder Video-Recorders wird über das Interface 'Videodat 300' mit der RS 232 Schnittstelle des Computers verbunden. Als Software kann jedes DFÜ-Programm verwendet werden. Die Parameter sind: 300 Baud / 1 Startbit / 8 Datenbit / 2 Stopbits / keine Parität.

Das Interface:

Das Interface 'Videodat 300' muß man sich selbst zusammenlöten. Die Kosten der Bauteile belaufen sich auf ca. DM 50. Ein Bausatz ist in Vorbereitung. Dieser soll mit einem speziellem IC arbeiten. Damit können die Kosten auf ca. DM 30 gesenkt werden. Das Schaltbild bekommt ihr beim WDR Computer-Club und demnächst in unserer Box.

Viel Spaß beim Löten! Ede u. Heribert

Die Firma Schmidtke electronic machte 1984 eine bemerkenswerte Ankündigung: es sollte Mitte des Jahres (wohlgemerkt: noch befinden wir uns im Jahr 1984) eine 80-Zeichen- und Grafikkarte für ca. 500 DM angeboten werden. Diese Karte sollte neben 80 Zeichen, 25 Zeilen auch 720 x 480 Grafikpunkte, einen eigenen Prozessor, einen 64K-Speicher, eine parallele und zwei serielle Schnittstellen bieten.
Zuerst verschob sich die Vorstellung auf den September, dann wurden diesbezügliche Anfragen gar nicht mehr beantwortet.
Mitte 1985 erfuhr ich dann, daß Schmidtke electronic endlich die Karte anbot. In der Anzeige war nun von 80 Zeichen bei 25 Zeilen und 4 Zeichensätzen die Rede: Deutsch + Invers oder Grafik, Amerikanisch + Invers oder Grafik. Außerdem wurde auf die Software und ein umfangreiches Handbuch hingewiesen. Es hörte sich gut an und war mit 245 DM (inzwischen auf 295 DM erhöht) noch erschwinglich. Also her damit.
Mitte November 1985 bestellte ich die Karte mit einem Scheck. Mitte Januar 1986 kam sie dann endlich. Diese Wartezeit ist bei Schmidtke noch guter Durchschnitt. Die Karte selber sieht dürftig aus: Es hätte doppelt so viel darauf gepaßt.
Der Einbau in mein Genie IIS war recht einfach, aber er nichts ist gegen den in das Genie I/II. Zuerst mußten zwei ICs ausgewechselt werden. Kein Problem. Dann sollte die Karte in einen freien Slot gesteckt und mit dem Videokabel verbunden werden. Dieses Videokabel war nur leider zu kurz. Selbst nach Umstecken der anderen Karten erreichte es nur unter äußerster Spannung die Karte. Bei dieser Spannung brachen dann auch die Lötstellen am Stecker nach außen. Ein neues Kabel und eine Videobuchse mußten besorgt werden, denn die alte war Hongkongmade, sprich: verklebt statt geschraubt.
Nun lief die Karte endlich. Aber das Bild auf dem Monitor mußte erst langwierig nachgestellt werden, bis sowohl im 80- als auch im 64-Zeichen-Modus alles erkennbar war. Beim Umschalten zwischen den beiden muß jedes Mal die Helligkeit verändert werden, denn im 80er-Modus sind die Zeichen heller.
Nun wurde das "Handbuch" studiert. Eigentlich ist es kein Handbuch, sondern eine Montageanleitung. 21 der 32 Seiten bestehen aus Schaltplänen (dabei wollte ich die Karte eigentlich nicht nachbauen) und der Einbauanleitung für das Genie I/II. Der Rest beschreibt kurz die drei Programme auf der Diskette und noch kürzer die Programmierung der Karte. Auf der ersten Seite wird extra gesagt: Januar 1986 erscheint das "richtige" Handbuch, mit dem beiliegenden "Gutschein" kann man es anfordern. Nur: erstens lag kein Gutschein bei und zweitens habe ich die Karte im Januar 1986 bekommen, also dürfte sich das Handbuch noch im Kopf des Autoren befinden (falls es jemand inzwischen irgendwo aufgetrieben hat, wäre ich stark daran interessiert).
Die Programme sind schlicht, um nicht zu sagen simpel. Eines testet die Karte und zeigt dabei die verschiedenen Zeichensätze an. Dabei stellt sich auch heraus, daß das oben erwähnte "Grafik" eine geschickte Abkürzung für "Grafikzeichen" ist, die genauso öde wie nutzlos sind (bis auf die Kartensymbole für Kreuz, Pik, Herz und Karo). Es gibt auf der Karte keine Grafik, nicht mal eine Entprechung für die SET/RESET-Grafik. Die 720 x 400 Punkte sind verschwunden. Ebenso der 64K-RAM-Speicher: Die 2048 Zeichen werden bei B000H oder F000H eingeblendet. Die Karte hat genau 4K Speicher. Dafür kann die Karte aber die vollen 64K des Computers für CP/M verfügbar machen. Erweiterungen wie MBA / EG64 / LSS1 werden damit überflüssig.
Von Schnittstellen keine Spur: Es gibt nur einen Anschluß für ein Netzgerät, falls man eins benötigt. Im Genie IIS entfällt dies.
Weiter mit den Programmen: Das eine schaltet den 80-Zeichen-Modus an, das andere ihn wieder aus. Interessant ist nur das erste. Es belegt den Speicher über

FC00H und leitet alle normalen Bildschirmausgaben auf die Karte. Wenn ein Programm allerdings nicht schön gesittet diese Routinen benutzt, sondern lieber poked, ist Sense: 80 Zeichen ade.
Dieses Programm ermöglicht es, über PRINT CHR$(7) + Steuercode im Basic die verschiedenen Schriftsätze anzuwählen und sogar ein PRINT$ bis 2047 durchführen. Dafür laufen dann andere Hilfsprogramme vielleicht nicht mehr. Ich habe das bisher nur mit dem Bildschirmeditor ausprobiert, und der meldet sich ab. Was soll ich mit fantastischen 80 Zeichen und diesem alten, total untauglichen Zeileneditor? Zum Heulen!
Andere Programme laufen von Fall zu Fall. Edtasm läuft im Editormodus, aber Assembeln tut er nicht immer, Scripsit meldet sich ab.
Dem könnte ja abgeholfen werden, indem man diese Programme so abändert, daß sie selbst die Karte programmieren. Würde vielleicht klappen, wenn man wüßte, wie! Aber im "Handbuch" steht nur, daß die Ports D0-D2 benutzt werden und die Werte, die man in das Steuerregister D2 schreiben muß, um bestimmte Reaktionen zu erreichen. Über D0 und D1 schweigt es sich aus, aber sie werden von dem Treiberprogramm angesprochen. Nur was es mit ihnen macht und warum, bleibt im Dunkeln.
So, ich glaube, jetzt habe ich die Karte genug verrissen. Noch einige Anmerkungen:
Es gibt die Karte für TRS80, Speedmaster, Genie I/II/IIS und das Colour Genie. Lohnen dürfte sie sich nur, wenn man intensiv CP/M 2.2 nutzen will. Dann ist es D I E Erweiterung. Ansonsten würde ich beim Genie IIS wohl eher die TCS-Karte und für die anderen Computer eine Grafikkarte empfehlen.
Falls mir jemand mehr Informationen über die Karte geben kann oder schon Programme angepaßt hat: könnte er sich freundlicherweise bei mir melden? Das würde meine Laune stark verbessern.
Gerald Schröder

Wer Interesse an einer ausführlichen Anleitung zu dieser Zeichenkarte hat, melde sich bitte bei der Redaktion.

Der Chefprogrammierer liest die Computerdiagnose und erbleicht: Sie hat ergeben, daß sich die Firma keinen Computer mehr leisten kann.

»Was hältst Du von Großrechnern, Albert?«
»Von diesen Strebern halte ich gar nichts; ich jedenfalls war in der Schule immer ein kleiner Rechner.«

Speed Writing

by Joe Lindsey

If you own an Epson MX-series printer, you're wasting time. That's because your Epson prints at a fraction of its potential. Imagine a 40 percent increase in printing speed—that's what you'll get with this modification. By replacing your printer's 6-megahertz (MHz) crystals with ones rated at 10 MHz, you can print out both text and graphics faster. And you can make this simple modification for about $6 and a half hour of your time.

Here's how the modification works: The Epson's 6-MHz crystals govern the printer's master clock. When you change the crystals, you change the clock frequency. The faster the system clock, the more instructions-per-second the printer's central processing unit (CPU) executes. And when the CPU goes faster, the Epson goes faster.

Printer Fix

You first need to buy two 10-MHz crystals from an electronics supply house. You also need a Phillips screwdriver, some masking tape, and a soldering iron.

Disconnect the Epson's external cable and power cord, put your printer on a table or other level surface, and you're ready to begin.

Pull the manual paper feed knob off the right-hand side of the printer and remove the smoked plastic cover: Flip it up to a vertical position and lift it out, left side first.

On the MX-100, reach inside the right side of the printer (by the control buttons) and unplug the cable to the buttons. This will make it easier to remove the case. Look along the front of the printer; you'll see three screws where the base of the gray cover was, and one screw in each of the back corners. Remove all five screws and lift off the case.

Turn the printer upside down, making sure you don't damage anything that sticks out of the top, and unscrew the four Phillips screws at each corner of the bottom of the case. Cover the holes with masking tape to keep the screws from falling out.

Turn the printer right side up and remove the top cover by lifting up on the left side and sliding the cover off the paper feed shaft that sticks out the right side.

You'll see some tied wires running up to the printer's control panel on the right side of the machine (see Photo 1). Unplug the connector in the top cover at the operator panel and the whole cover should come free. Set it aside for now.

Turn the printer around so the back faces you. I'll refer to the right and left sides as you view the printer from the rear; I'll use "front" and "rear" in their true senses.

Look carefully at your printer from the rear to get acquainted with its components. The power supply appears on your left. A large control circuit board lies at the bottom of the machine behind the paper feed mechanism. A smaller board, called the driver circuit board, sits on the right side and on top of the control board.

Remove the driver board from the control board by removing the two Phillips screws along the right side of the driver board (see Photo 2). The left side of the driver board plugs into a couple of sockets (CN4 and CN5) on the control board. Carefully work the pins out of the sockets underneath, making sure you don't bend them. You may want to remove the big, blue male plug (CN6) on the right side of the driver board. I left mine plugged in and let the whole board dangle by the attached wiring. Remove the driver board by grasping it and working it loose while pulling straight up.

A single ground wire connector (FG) attaches to the rear of the control board, just to the left of the external printer connector (CN1). Slip it off the pin onto which it plugs. You need to remove two power supply connectors (CN2 and CN6), both of which are on the left rear corner of the control board, near the largest of the two capacitors mounted there (C1) (see Photo 3).

To free the control board, you have to remove three Phillips screws. One is in the center of the printer just behind the paper feed assembly, between the integrated circuits labeled 2C and 3C (midway between the two white standoff posts protruding up through the control board from the bottom). The other two screws sit at the left side of the board in front of the power transformer (see Photo 4).

The only things holding the control board in place now are three plastic tabs molded into the base, two along the rear of the board, the third on the right side. Push each plastic tab out of the way enough to clear the board, then lift the corner of the board out from under the tab (see Photo 5).

Raise the control board over the standoffs molded to the base and slide it out from under the paper feed assembly. The board is made of a paper laminate material that breaks and chips easily, so be careful when removing it. Also, use caution when soldering or unsoldering, as excessive heat will destroy the board.

Unsolder the two 6-MHz crystals (CR1 and CR2) from the control board and solder the 10-MHz crystals in their place, keeping the lead lengths the same as the original (see Photo 6). CR1 is located toward the left and on the front of the control board just to the left of the large integrated circuit labeled 1C. CR2 is on the right side of the control board just in front of the large integrated circuit labeled 9B.

The crystals in my printer were also held in place by double stick tape. I had to pry up on them slightly to break the tape bond, but the tape was still sticky enough so that I didn't have to replace it. The new crystals came encased in metal, so I stuck a small piece of black electrical tape to the bottoms to keep them from shorting out to any circuitry close by.

Now you can reassemble your printer, reversing the above procedure. Be sure you put all connectors back in their original positions without bending any of the pins. Make sure that the connectors line up properly between the driver board and main printed circuit board. Don't force anything. Additionally, when you reinstall the PC board you need to push each of the plastic tabs out of the way before you gently push the board into place. Don't expect the board to move the tabs. Finally, don't forget the control panel plug in the top cover of the printer.

Printer Ready

The operation I've just described for speeding up your MX-80 is reversible—you can put the old 6-MHz crystals back in and your printer will work the way it did before. If you ever need to take your printer in for service, you may need to put the original crystals back in to return it to "normal" speed. Epson won't fix a modified printer without unmodifying it, and you can't exchange any boards to which you've soldered.

I do offer one caveat: I wouldn't try to increase the speed over 10 MHz, since the CPU and memory can't safely drive the Epson any faster than that.

I've noticed no change in print quality and no skipping of dots after increasing the speed of my MX-80, even in heavy graphics printing. You'll hear a noticeable change in pitch as it prints, but I don't think it's any louder. Another positive result is that the skip and space speed of the paper feed has also increased. ■

A simple, inexpensive modification to Epson MX-series printers that increases printing speed by up to 40 percent.

System Requirements

Epson MX-series printer

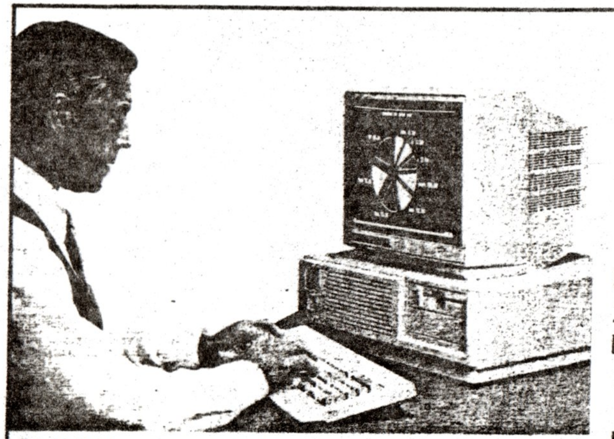
Support für Genie

Den Support der Recher Genie I und II übernimmt ab sofort die Firma Schmidtke Computersysteme, Sandkaulstr. 41, 5100 Aachen. Produktion, Vertrieb und Support der aktuellen Modelle (Genie IIs, Genie IIIs und Genie 16) bleiben unverändert in den Händen der Firma Phoenix-Computer.

Phoenix-Computer, Computer GmbH & CO KG, Gewerbegebiet Klarensplatz, 5461 Windhagen, 0 26 45/3222

Tandy-Computer mit 80286

In den gehobenen Bereich der MS-DOS-Computer ist der neue Tandy-3000 einzuordnen. Diese Tatsache belegt nicht nur die Wahl der CPU (80286 mit 8 MHz Taktfrequenz), sondern auch der Arbeitsspeicher von 512 KByte in der Standardversion, der ohne Erweiterungskarten auf 640 KByte ausgebaut werden kann. Das eingebaute Diskettenlaufwerk bringt auf jeder Scheibe 1,2 MByte unter, kann aber auch IBMs Standardformat mit 360 KByte lesen. Das Betriebssystem, MS-DOS 3.1, gehört nicht zur Grundausstattung, kann aber getrennt bezogen werden. Für Mehrplatzanwendungen bis zu sechs Benutzern wird künftig Xenix 5.0 auf dem Tandy-3000 laufen. Als lokales Netzwerk für MS-DOS-Computer bietet Tandy ViaNet an. Das Gerät hat insgesamt zehn Erweiterungssteckplätze: Sieben davon sind kompatibel zum IBM-PC/AT und zwei zum IBM-PC/XT. Ein weiterer „XT"-Steckplatz kann für kurze Karten verwendet werden; er ist normalerweise von der Parallel-/Seriell-Schnittstelle belegt, die zur Grundausstattung gehört. Wer viel Massenspeicher braucht, kann sich für das Modell HD entscheiden, das eine 20-MByte-Festplatte enthält. Aber auch beim Grundmodell kann der Massenspeicher auf über 40 MByte ausgebaut werden. Für rechenintensive Aufgaben gibt es den Arithmetikprozessor 80287 als Zusatz.



Mit der Taktfrequenz von 8 MHz läuft der Prozessor 80286 im Tandy-3000

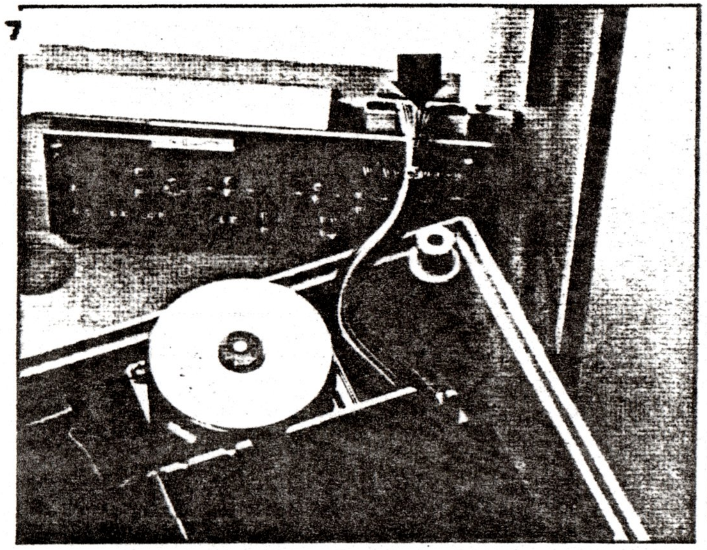


Photo 1. Unplug the connector at the operator panel to remove the cover.

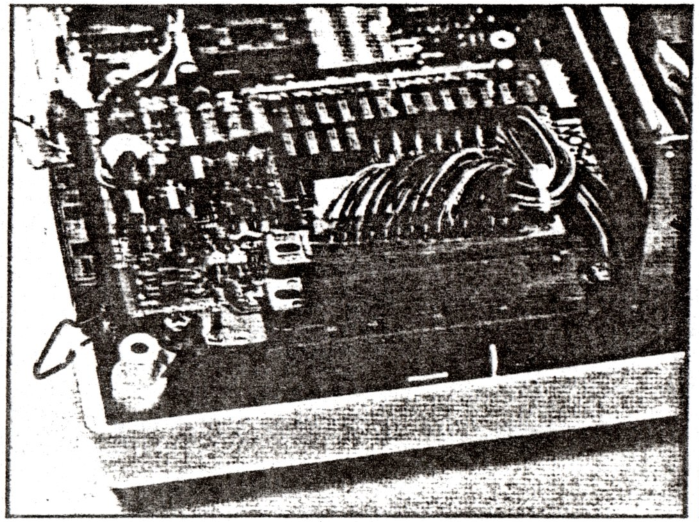


Photo 2. Remove the driver board from the control board by removing the two Phillips screws along the right side of the driver board.

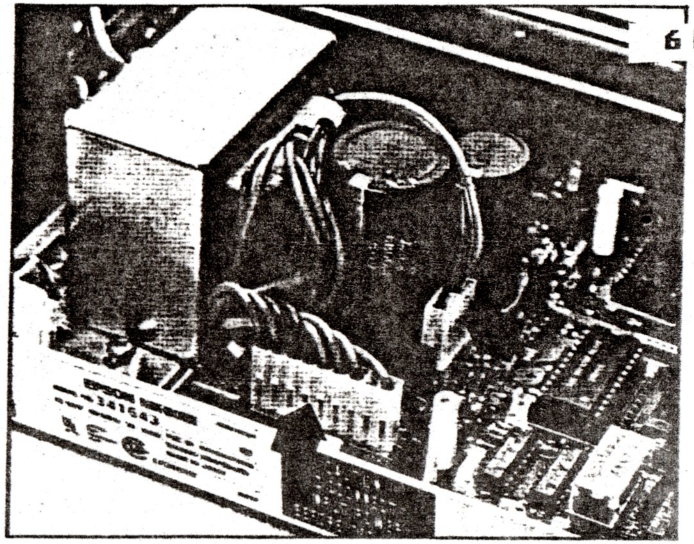


Photo 3. Remove CN2 and CN6 from the left rear corner of the control board.

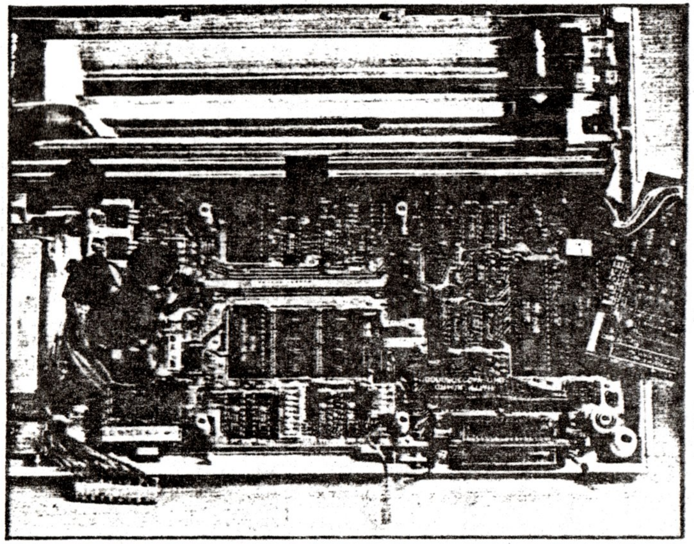


Photo 4. Free the control board by removing the three Phillips screws at the left side of the control board.

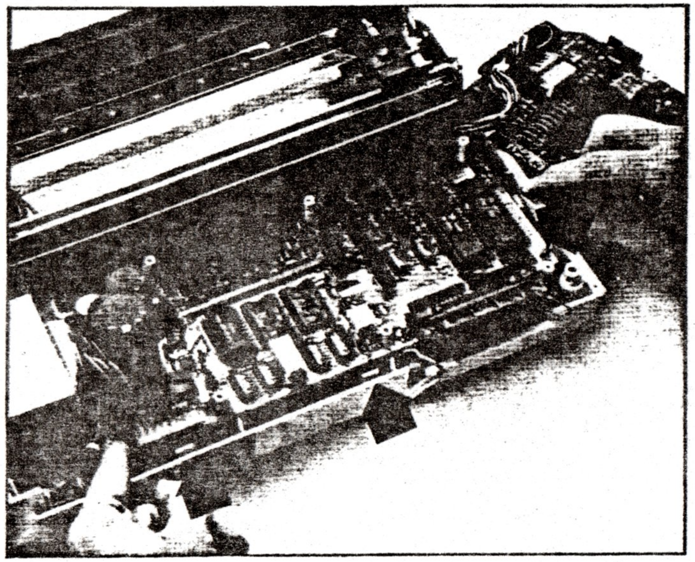


Photo 5. Push each of the three plastic tabs on the base of the printer out of the way.

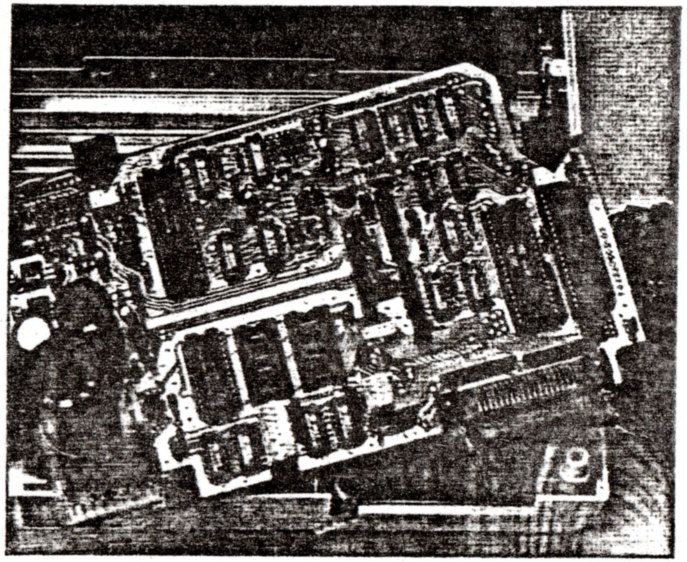
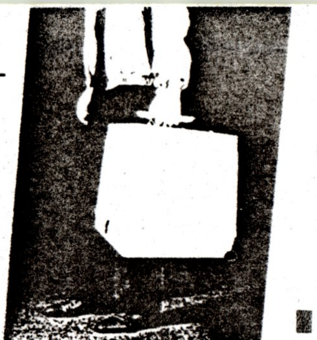


Photo 6. Unsolder the two crystals CR1 and CR2 from the control board.



TRS-80 Modell 4P: Der schönste Portable der Welt

Daß ein Portabler nicht immer ein häßlicher schwarzer Kasten sein muß, hat Tandy mit dem Modell 4P eindeutig bewiesen. Ich habe noch keinen Portablen gefunden, der besser aussieht. Bliebe die Frage, was hinter der schönen Fassade steckt? Ein Anwender des großen Modell 4 hat den Kleinen getestet.

Eine englische Fachzeitschrift schrieb unter der Überschrift »Tandy's Sweet P«, das 4P sei ein M4 »with wanderlust«. Längere Wanderungen dürften damit allerdings nicht sehr lustig werden, denn immerhin wiegt der Kleine 11 kg, so klein ist er also gar nicht. Das liegt an der sehr soliden Konstruktion, ein Portable muß schließlich rauhe Behandlung vertragen.

Die Bildröhre hat 9 Zoll Diagonale und damit ein Format, das als angenehm bezeichnet werden kann. Womit auch gesagt sein soll: die 7-Zoll-Formate der Mitbewerber sind bei 80 Zeichen/Zeile meines Erachtens unerträglich.

Die Frontseite ist durch einen Deckel abgedeckt, der nach Lösen von zwei Schnellspannern abgenommen wird. Der Tragegriff ist auf der Rückseite, das Gerät wird also mit dem Schirm nach unten transportiert. Im Deckel ist Platz für das Netzkabel, ein Fach für das Pocket-Manual und 4 bid 5 Disketten.

Auf der Rückseite verbergen sich unter einer Klappe die Buchsen für das Netzkabel, eine RS232- und eine Centronics-Schnittstelle, Anschlüsse für den Rechner-Bus und für bis zu vier Festplatten. Eine Tonband-Buchse fehlt.

Auf der Vorderseite links vom Schirm sind der beleuchtete Netzschalter und die Regler für Helligkeit und Kontrast. Die Reset-Taste ist unter dem Netzschalter angebracht und damit so gut versteckt, daß man sie garantiert nie versehentlich auslöst.

Auch innen sieht es gut aus

Das Gehäuse ist recht einfach zu entfernen: Zwei Schrauben auf jeder Seite und die beiden unter dem Griff lösen, Gehäuse abziehen. Nach dem Lösen zahlloser Schrauben müßte man an die Einzelteile herankommen, mit wenig Aufwand läßt sich aber feststellen, daß das CPU-Board die Unterseite belegt, links steht senkrecht die Video-Platte, rechts die Laufwerk-Elektronik, den Deckel bildet sozusagen das Netzteil. Die Baugruppen sind untereinander nochmals mit Blechen abgeschirmt; jetzt weiß man langsam, wo die 11 kg herkommen. Immerhin, lieber eine solide Konstruktion, die auch mal einen Bums verträgt, als ein empfindliches Leichtgewicht. Auf der Rückseite sieht man nach Abnehmen eines Blechdeckels in einen Schacht, der noch eine Platine aufnehmen kann, nämlich die für das in Deutschland nicht zugelassene Modem. Auch der CPU-Bus ist mit einem Blechdeckel abgedeckt.

Vorbildliche Tastatur

Die Tastatur ist beim Transport in einem Fach unter dem Bildschirm gelagert, man muß sie da herausziehen. Leider ist das Kabel etwas kurz geraten. Zur ergonomischen Anordnung auf dem Tisch reicht es noch, nicht aber für die amerikanische Kreativ-Haltung, sprich Füße auf dem Tisch und Tastatur auf dem Schoß. Um gleich bei der Tastatur zu bleiben: Sie ist extrem flach, kann mit zwei ausschwenkbaren Füßen schräg gestellt werden und hat 70 beziehungsweise 75 (deutsche) Tasten. Die Anordnung ist sehr gut (besser als beim M4), Shift und Enter sind deutlich abgesetzt. Die Tasten lassen sich gut bedienen und haben einen deutlich hörbaren Anschlag, allerdings nicht so blechern, wie beim IBM-PC. Auch schnelle Tipper können die Tastatur nicht schlagen. Alle Tasten haben eine Wiederholautomatik, die Ansprechverzögerung und die Widerholfrequenz sind per Software einstellbar, die Grundeinstellung ist aber meines Erachtens schon genau richtig.

Beim Bildschirm hat sich viel getan

Das M4 hat als deutsche Version immer noch den Fehler, bei großen Grafik-Flächen ständig und bei Texten im 80-Zeichen-Mode manchmal zu flimmern. Beim 4P ist das Bild gestochen scharf, und steht wie eine Eins. Obwohl der Schirm kleiner ist (9 statt 12 Zoll), läßt er sich deshalb mindestens genau so gut ablesen. Die Option, auf 40 Zeichen/Zeile umzuschalten, habe ich mal probiert (sie funktioniert), aber sonst nie angewendet. Tatsächlich hat man die Qual der Wahl zwischen 24 Zeilen mal 80 Zeichen, 24 mal 40 und 16 mal 64.

Erste Inbetriebnahme in drei Sprachen

Das Modell 4P soll international sein, was an sich kein Problem ist, man muß halt nur die entsprechende nationale Diskette einlegen, von der dann der Zeichensatz geladen wird. Dafür hat der Computer einen Bootstrap-ROM (Urlader). Alle Fehlermeldungen und Prompts erscheinen in Englisch, Deutsch und Französisch, womit wir wissen, was der Hersteller für die drei wichtigsten Sprachen der Welt hält. Daß sich damit der Bootstrap-ROM vom M4 unterscheidet, ist klar, allerdings noch in einem anderen Punkt. Er erwartet TRSDOS 6.1.1 oder neuer. Eine Diskette der Version 6.0.0 oder 6.1.0 bootet nicht. Solche Disketten können allerdings ohne Probleme gelesen, und mit einem mitgeliefertem Programm auch »upgedatet« werden. Letzteres spielt nur das neue DOS auf, läßt aber alle anderen Dateien unberührt.

Ein weiteres Problem: Das Testgerät hat noch keine deutsche Tastatur. Die Lösung: Ich schreibe die Umlaute aus und lasse sie auf dem M4 über die globale Ersetzfunktion wieder in »äöüß« wandeln.

Laut Prospekt besteht kein Unterschied zwischen dem M4 und dem 4P, vom Äußeren einmal abgesehen. Im M4 ist ein Modell 3 physikalisch echt enthalten. Tatsächlich fehlt dem 4P aber das Modell-3-ROM. Beim M4 ist es so, daß das ROM dem RAM adreßmäßig parallel liegt, und per Bank-Select automatisch das eine oder andere gewählt wird, je nachdem, welche Diskette man einlegt. Beim 4P funktioniert das im Prinzip auch so, nur daß das sogenannte ROM-Image auf der Diskette vorhanden sein muß. Diese wird beim Systemstart zuerst in das RAM geladen, was natürlich einige Sekunden dauert. Hat man kein ROM-Image auf der Diskette, kann man mit der zum Lieferumfang gehörenden Software ein solches darauf kopieren. Ist dafür kein Platz mehr, muß man mit einer anderen Diskette starten und dann die Scheibe wechseln. Dies mag auf den ersten Blick ein Nachteil sein, Hacker aber werden jubeln. Haben Sie doch jetzt die Möglichkeit, ein Modell 3 mit maßgeschneidertem »ROM« anzufertigen.

Ich habe einige Modell-3-Software getestet, auch unter einem fremden

DOS (NEWDOS/80), verschiedene Programme eingesetzt, die exzessiv ROM-Ware benutzen: kein Fehler feststellbar. Im Modell-4-Mode läuft das System von Haus aus (Lieferumfang) unter TRSDOS 6.1, ein Betriebssystem, das keine Wünsche offen läßt. Computer Nr. 3 ist eine reine CP/M-Maschine, vorausgesetzt, Sie starten mit einer passenden Diskette. Von Tandy lange angekündigt, ist CP/M 3.0.

Drei Computer in einem

Dazu muß man wissen, daß das 4P von Haus aus 64 KByte RAM hat, das M4 ist darauf aufrüstbar.

Beide bieten die Möglichkeit, auf 128 K zu erweitern (329 Mark). Diese letzten 64 KByte sind aber nur über Bank-Select zu adressieren, was TRSDOS 6.1 perfekt unterstützt, nicht aber CP/M so ohne weiteres. Hier dürfte das Problem in der Anpassung von CP/M 3.0 gelegen haben. Ungeduldige haben inzwischen CP/M 2.2 gekauft. Vielleicht war die Idee gar nicht so schlecht. Dieses CP/M soll laut Tandy zwar nicht perfekt sein, es hat aber einen Vorteil: Es liest und schreibt im Format von IBM, Kaypro, Osborne und anderen.

Welches Betriebssystem, das ist die Frage!

TRSDOS 6.1 ist für das Modell 4 maßgeschneidert und eindeutig benutzerfreundlicher als CP/M, letzteres hat aber den Vorteil, daß es ein riesiges Softwarepotential dafür gibt. Das heißt, wenn Sie unbedingt Software benötigen, für die unter TRSDOS nichts Gleichwertiges angeboten wird, müssen Sie wohl oder übel in den teuren Apfel beißen.

Eine Zukunftsinvestition dürfte aber der Kauf von CP/M nicht sein. Der Schwerpunkt liegt eindeutig auf TRSDOS, was durch eine Open-System-Politik unterstützt wird. Das technische Manual das M4 (und 4P) ist ein Musterbeispiel dafür. Die Hard- und Software ist völlig offen gelegt, für einen Systemprogrammierer bleibt keine Frage unbeantwortet. Daß der Hersteller inzwischen selbst exellente Software entwickelt, zeigt das ALDS, ein Super-Assembler, der beim Autor gerade im Test ist; aber auch die Open-System-Politik trägt Früchte, wie Alcor-Pascal und Multiplan zeigen. ▸

Mein Rat: Kaufen Sie zuerst TRSDOS 1.3, das Betriebssystem des Modell 3. Damit steht Ihnen schon einmal eine riesige Programm-Bibliothek zur Verfügung, in der Regel wesentlich preiswerter als CP/M-Software. Dann schauen Sie sich das M4-Angebot an, und wenn dann noch was fehlt, versuchen Sie Passendes unter CP/M zu finden.

Disk-Drives und Geräusche

Zwei Disketten und dann nur noch Festplatten als Option. Das heißt, für unterwegs muß man sich mit 360 KByte Massenspeicher (einseitige Aufzeichnung) »bescheiden«, zu Hause darf man dann auf 20 MByte aufrüsten. Das M4 erlaubt, noch zwei externe Laufwerke anzuschließen, die auch »double sided« sein dürfen, es ist also im Prinzip preiswerter mit Massenspeicher auszurüsten. Im M4 klappern die Laufwerke ganz schön, was das große Gehäuse als Resonanzboden noch kräftig unterstützt, die 4P-Laufwerke sind dagegen wohltuend ruhig und deutlich schneller. Ich habe ein 1 100-Zeilen-Programm mit einem Assembler übersetzt, der dazu ständig auf die Disk, wo der Source-Text liegt, zugreift. Auf dem M4 dauert das 80 Sekunden, auf dem 4P nur 66.

Was gibt es an Software?

Mitgeliefert wird TRSDOS 6.1.1 und Basic. Vom TRSDOS 6.1.0 gibt es auch eine Version für die deutsche Tastatur, Sie sollten allerdings auf der Version 6.1.1.D bestehen, die noch besser eingedeutscht ist, zum Beispiel beim Datum. TRSDOS 6.x ist ein exzellentes Betriebssystem, das keine Wünsche offen läßt. Wenn Sie NEWDOS kennen, wissen Sie, daß man da lange Parameterlisten mit unverständlichen Kürzeln eingeben muß, oder Ersatz-Werte durch viele Kommas zu definieren hat. Die vielen Parameter sind bei TRSDOS6 auch möglich, nur daß dafür Klartext oder auch nur die Anfangsbuchstaben zugelassen sind. Aber das Wesentliche: Wissen Sie die Parameter nicht, geben Sie einfach den Befehl, zum Beispiel BACKUP ein, und Sie werden gefragt.

Besonders zu erwähnen wäre noch MEMDISK, eine sehr einfach zu bedienende RAM-Floppy und SPOOL, ein Spooler, der im einfachsten Fall durch Nennung seines Namens aktiviert wird. Allein diese beiden Programme sind Grund genug, die 128-KByte-Erweiterung, die jetzt auch in Deutschland erhältlich ist, zu kaufen. Es ist schon ein erhebendes Gefühl, einen großen File auf »Disk« zu schreiben, und er ist da, sobald man die Enter-Taste wieder losgelassen hat.

Ansonsten gibt es noch einige Programme für Fortgeschrittene, wozu zum Beispiel die Communication-Software (RS232) zu zählen ist.

Die läßt zwar das Herz eines jeden Hackers höher schlagen — es fehlt nichts, was einen daran hindern könnte, fremde Großrechner anzuzapfen — aber die Bedienung ist doch arg umständlich. Diese Umstände könnte man zwar auch einem ICL-File anvertrauen, die Job Control Language ist auch einsame Spitze, aber leider von Profis für Profis gemacht. Wer sich nicht scheut, den ausführlichen Anhang zu diesem Thema zu studieren (lesen reicht nicht), hat dann allerdings auch ein mächtiges Instrument zur Verfügung. ICL ist nicht nur ein Textfile, der anstelle der Tastatur gelesen wird, sondern schon eine Programmiersprache für sich.

Gut gelöst ist in TRSDOS 6.x das Konzept der »logischen Geräte«. Das heißt, Tastatur, Bildschirm, Drucker, die RS-232 und Disk-Files sind im Prinzip alle gleich zu behandeln. Mit LINK kann man die »Geräte« miteinander verbinden, zum Beispiel Schirm und Drucker, mit dem Erfolg, daß man sieht, was gedruckt wird. Mit ROUTE wird umgeleitet, zum Beispiel der Drucker auf den Schirm, womit sich die Formatierung überprüfen läßt, ohne Papier zu verbrauchen. In alle »Links« kann man sogenannte Filterprogramme schalten, um Zeichen umzuwandeln, welche hinzuzufügen (zum Beispiel zu jedem CR ein LF) oder andere zu »strippen«, sprich zu unterdrücken. Einige Filterprogramme werden mitgeliefert, wovon FORMS/FLT (Druckformat) und KSM/FLT (vom Anwender zu definierende Tasten-Makros) besonders hervorzuheben sind.

Das Basic ist mehr als erfreulich

Was sich da im Copyright mit Basic 1.0 meldet, ist Microsoft-Basic Version 5.

Die besonderen Eigenschaften dieses Basics erkennt man erst, wenn man sehr tief (mit einem Disassembler) hineinschaut. So werden zum Beispiel Konstanten, auch hexadezimale und oktale, sofort nach der Eingabe in das rechnerinterne Format umgewandelt und so im Programmtext zusammen mit einem Typ-Byte gespeichert. Das heißt, der alte Trick, häufig benutzte Konstanten als erste Variable zu definieren, ist hier unangebracht, Konstanten sind immer schneller!

Aber auch wenn man das Basic nur von außen (laut Handbuch) kennenlernt, wird man erfreut sein. Zuerst: Wer Level 2 oder Basic-80 beherrscht, kann sofort loslegen, wenn er sich bitte nur eines merkt: Reservierte Worte sind mit Blanks zu trennen. Tatsächlich ist Level 4, wenn ich das mal so nennen darf, eine Erweiterung von Level 2 oder 3.

Über die neuen Befehle sollte man allerdings gründlich nachdenken. Auf Seite 32 unten links sehen Sie zum Beispiel, wie man mit WHILE/WEND und SWAP (vertauschen von zwei Variablen) ein Sortierprogramm mit einer in Basic bisher nicht möglichen Effektivität schreiben kann. Die langen Variablennamen sind echt, bis zu 40 signifikante Zeichen sind jetzt möglich. Das Bemühen, volle Kompatibilität zu den Vorgängern zu erhalten, führt allerdings auch zu einiger Redundanz, sogar an falscher Stelle.

So gibt es nach wie vor USR, obwohl der neue Befehl 'CALL Name (Parameterliste)' erlaubt, beliebig viele Routinen aufzurufen (andere Namen) und alle mit soviel Parametern, wie in einer Basic-Zeile von 250 Zeichen unterzubringen sind.

Diesem sehr guten Basic nicht mehr angemessen ist der Editor, der genau der des alten Level 2-Basics ist. Ich würde empfehlen, Tandys ALDS zu kaufen. Dieses Assembler-Entwicklungs-System beinhaltet einen erstklassigen Editor, der auch auf Basic-Files (ASCII-Format) anwendbar ist. Einschränkung: Basic-Zeilen dürfen dann nicht länger als 80 Zeichen sein, deutsche Umlaute werden verfälscht.

Alle Programme der Modelle 1 und 3 laufen auch auf dem 4, sofern Sie jedes PEEK und POKE und alles was sonst mit den Maschinen-Internas zu tun hat, strikt vermeiden. Wenn Sie die Programme vom Modell 1 mit REPAIR beziehungsweise vom M 3 mit CONV übertragen, dann bitte nur im ASCII-Format. PRINT heißt zwar immer noch PRINT, ist aber intern anders codiert, die sogenannten Tokens haben alle eine andere Bedeutung. Vergißt man das, erhält man sehr lustige Listings, nur laufen tun sie nicht. REPAIR und CONV versagen aber bei Modell 3-Disketten, die unter NEWDOS erstellt wurden.

```
100 ' Demo Bubblesort mit WHILE/WEND und SWAP
110 '=========================================
120 '
130 'Array A mit Zufalszahlen laden:
140 FOR I=0 TO 10:A(I)=RND(100):NEXT
150 '
160 'Start Sort
170 '-----------------------------------------
180 FLAG=1
190 WHILE FLAG
200       FLAG=0
210       FOR I=0 TO 9
220            IF A(I)>A(I+1) THEN
                                SWAP A(I),A(I+1):
                                FLAG=1
230       NEXT I
240 WEND
250 '
260 'Print sorted data
270 '-----------------------------------------
280 FOR I=0 TO 9:PRINT A(I);:NEXT
290 '
300 END
```

◂ *So einfach kann man mit dem neuen Basic ein Sortierprogramm schreiben.*

Die Anpassung eines Druckers ist an sich kein Problem, sollte aber im Manual besser beschrieben sein. ▾

```
.LA180/JCL
.Set up for RS-232 -> DEC-Printer LA-180
.
SET *CL TO COM/DVR
SETCOM (BAUD=2400,CTS=YES)
ROUTE *PR TO *CL
.
.LA-180 ready to use
//EXIT
```

Also schnell eine TRSDOS-1.3-Diskette besorgen (27 Mark) und Zwischenkopien erstellen. Für Grafik-Fans eine schlechte und zwei gute Nachrichten, zuerst die schlechte: SET, RESET und POINT sind entfallen. Die erste gute Nachricht: Es gibt eine Grafik-Karte für 768 Mark mit einer extrem hohen Auflösung und mächtigen Grafik-Befehlen.

Die zweite gute Nachricht: Computer persönlich bringt demnächst ein Programm, das die Blockgrafik-Befehle wieder installiert, und das mit doppelt so hoher Auflösung als beim Modell 3.

Beim **M4** hat man die Buchse für den Kassetten-Recorder noch belassen, wohl, weil es das Gerät als Billig-Version auch ohne Disks gibt; beim 4P fehlt sie und folglich auch CLOAD und CSAVE. Somit dürfen 4P-Besitzer auch TAPE100 löschen, was Tandy vergessen hat. Das ist das Programm zum Einlesen von Modell-100-Kassetten. Sie können das nicht löschen? Nun, das Paßwort heißt UTILITY.

Die Dokumentation: Qualität.

In diesem Punkt war Tandy vom Inhalt her schon immer gut, jetzt auch in der äußeren Form. Mein M4-Manual wurde noch als Kopie von Schreibmaschinenseiten im Pappdeckel geliefert, die 4P-Anleitung ist ein Ring-Ordner in Kunstleder mit Goldeindruck. Die Seiten sind richtig gesetzt und offensichtlich von einem Layouter gestaltet, das heißt angenehm und klar zu lesen, sofern man Englisch beherrscht. Da Tandy aber inzwischen Deutschland entdeckt hat, was Tastatur und DOS schon beweisen, ist auch hier noch Besserung zu erwarten.

Vom Inhalt her ist fast nichts zu bemängeln, bestenfalls zuviel Ausführlichkeit, aber da gibt es ja noch das Pocket-Manual, mit dem ich gern praktisch arbeite.

Das »fast« mußte ich geben, weil ich die Kapitel über die RS232 nie verstanden hätte, wenn ich nicht schon einiges darüber wüßte und nicht wie wild im Manual geblättert hätte. COM/DVR, SETCOM und COMM/CMD gehören nämlich zusammen, sind aber im Handbuch völlig getrennt.

Tatsächlich sind nur drei Befehle nötig: die Zeilen mit dem Punkt sind Kommentare. Legt man die in einem Do-File ab, reicht »DO Name« und der Drucker ist benutzbar. SETCOM selbst ist ein gutes Programm, sein Aufruf zeigt den Ist-Stand, mit »SETCOM(Parameter)« kann man jederzeit ändern, und sich so sehr einfach an die Besonderheiten einer wilden Norm wie die der RS232 herantasten.

M4 oder 4P, das ist noch eine Frage

Theoretisch kosten beide Computer dasselbe, praktisch ist das 4P in gleicher Ausbaustufe um den Preis der dort serienmäßigen RS232-Schnittstelle billiger (rund 300 Mark). Wenn Ihnen zwei Diskettenlaufwerke reichen (oder sie sich Harddisks leisten wollen), ist vom technischen her kein Unterschied festzustellen. Bliebe der kleinere Schirm (9 statt 12 Zoll). Wie gesagt, der kleinere ist schärfer, und Apples Macintosh hat auch 9 Zoll.

Vorteilhaft ist die absetzbare Tastatur, und der Kleine blockiert nicht gleich den ganzen Schreibtisch. Er sieht besser aus und kann leicht weggeräumt werden.

(Peter Wollschläger)

Auf eine Blick

CPU: Z80A, 4 MHz
RAM: 64 KByte, aufrüstbar auf 128 KByte
zwei eingebaute Diskettenlaufwerke, je 184 KByte
Erweiterung um bis zu vier Festplatten, je 5 MByte
Parallelschnittstelle und RS232 serienmäßig
Betriebssystem: TRSDOS 6.1 (Standard), CP/M 3.0 (Aufpreis)
Sprachen: Basic (Microsoft Ver. 5)
Gegen Aufpreis: Pascal, Assembler und anderes
Nachrüstbar: Hochauflösende Grafik
Bildschirm 9 Zoll, grün auf schwarz oder invers
24 Zeilen mal 80 Zeichen umschaltbar auf 40 Zeichen
Absetzbare Tastatur, 75 Tasten
Besonderheiten: Kann TRS-80, Model 3 emulieren
Tonausgabe
Gewicht 11 kg

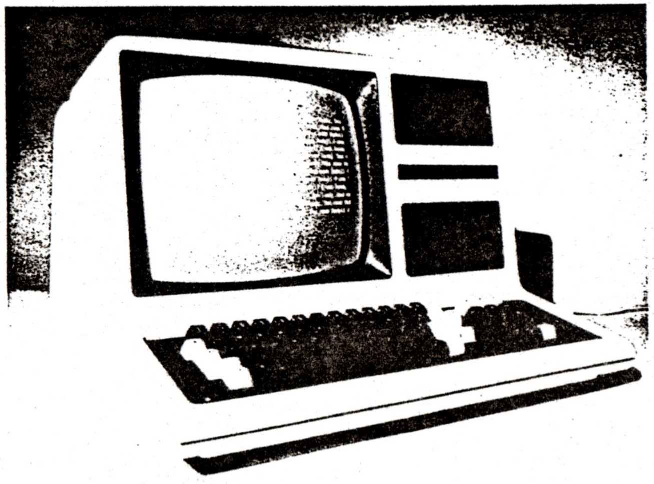


Bild 1. Ähnelt dem Vorgänger Modell III: Tandys neues Modell 4

TRS-80 Modell 4 — Drei Computer in einem?

Das seit langem auf dem Markt befindliche Modell I und das sich kaum davon unterscheidende Modell III bedurften eines Nachfolgers auf dem neuesten Stand der Technik. Mit Rücksicht auf die umfangreiche Software-Basis sollte die Kompatibilität der Reihe erhalten bleiben, ein besseres DOS war nötig, und die CP/M-Welt durfte auch nicht länger ignoriert werden. Lesen Sie, ob und wie Tandy diesen gordischen Knoten mit seinem Modell 4 durchschlagen hat.

Das Modell 4 sieht aus wie ein Modell III (Bild 1), nur daß dessen abwischbarer Metallic-Look durch soliden, mattweißen Kunststoff ersetzt wurde. Zum Öffnen sind zehn Schrauben auf der Unterseite zu lösen (man merke sich tunlichst, wo welche hingehören). Dann hebt man das Oberteil ab. Das erscheint einem sehr schwer, und das ist es auch, denn da hängt die ganze Bildröhre nebst Video-Platte dran (Bild 2). Jetzt hat man freie Sicht auf das Chassis und erblickt ganz hinten die große Hauptplatine, davor zwei kleinere für den Disk-Controller und das Netzteil. Die Hauptplatte ist durch ein gleichgroßes Blech abgeschirmt, auch sonst dürfte die Post zufrieden sein. Jedenfalls funktioniert der beliebte Modell-I-Trick, »Ton« über ein neben dem Computer stehendes Kofferradio auszugeben, nicht mehr.

Auf der Hauptplatte sieht man fünf Drahtbrücken, von denen mindestens eine wild gebastelt ist, immerhin ein Zeichen dafür, daß das Gerät erprobt wurde (Bild 3).

Die Sound-Platine mit ihren nicht viel mehr als drei Bauteilen hätte auch noch bequem Platz auf der Hauptplatte gefunden, und man hätte die Steckverbindung gespart. Aber irgendwie muß man ja Kunden, die die abgemagerte Grundversion kaufen wollen, davon abhalten. Sound scheint ein gutes Verkaufsargument zu sein.

Ein weiterer Blick auf die Platine zeigt, daß Tandy den in der Einleitung genannten gordischen Knoten mit Geld, sprich Hardware, durchschlagen hat. Mit 64 KByte RAM ist das Modell 4 als CP/M-Maschine ausgerüstet, 14 KByte ROM extra sind nötig, um die Modell-III-Betriebsart zu ermöglichen, und noch einmal 2 KByte Video-RAM zusätzlich waren nötig, um nicht CP/M's TPA durch einen Bildspeicher ab 3C00H zu verwirren.

Wer selbst von 64 auf 128 KByte aufrüsten will, muß mindestens ein Teil bei Tandy kaufen. Die Adreßlogik wird mit einem PAL (Programmable Logic Array) gemacht, und PALs sind kundenspezifische Schaltkreise, die nur an diese ausgeliefert werden. Wie gesagt, es gibt nur noch ein Netzteil, früher wurden die Disklaufwerke (2 x 180 KByte Speicherkapazität) gesondert versorgt, und das merkt man als leichtes Zucken im Bild, wenn ein Diskmotor anläuft.

Insgesamt macht die Technik einen sehr soliden Eindruck, irgendwo müssen die 20 kg Gewicht schließlich herkommen. Platz für Erweiterungen ist genügend vorhanden. Tandy bietet zur Zeit eine RS232- und eine Grafikkarte zum Einbau an.

Die Tastatur ist deutsch (Bild 4) und hat mit 75 Tasten fünf mehr als die amerikanische, die im Handbuch beschrieben ist. Auf jeden Fall ist sie angenehm zu bedienen, prellfrei und hat Autorepeat.

Leider habe ich noch das amerikanische DOS, das deutsche soll in zwei Wochen hier sein. Das macht stutzig. Man sollte sich doch überlegen, ob man nicht bei der auch hier angebotenen ASCII-Tastatur bleibt, zumal, wenn man noch andere Computer bedient.

Der Zeichengenerator erzeugt 256 Zeichen, darunter die deutschen Umlaute, griechische Zeichen und anderes mehr. In Kombination mit Shift, Clear und Control sind sogar alle von der Tastatur aus erreichbar.

Zu loben ist der Type-Ahead-Buffer (nur im Modell-4-Modus) zu bedauern ist, daß die mit Control erzeugten Codes nicht wie beim Apple in Texte aufgenommen werden, man also weiterhin CHR$(x) tippen muß. Die drei Funktionstasten werden vom Basic-Editor akzeptiert, ansonsten lösen sie keine besonde- ►

ren Aktivitäten aus. Der Anwender muß sie selbst (in Basic mit INKEY$) abfragen.

Der Schirm (grün) bietet 24 Zeilen x 80 Zeichen im Modell-4-Mode, 16 x 64 im Modell-III-Betrieb. Bei Breitschrift wird auf 40 beziehungsweise 32 Zeichen/Zeile umgeschaltet. Reverse Video ist auch bereichsweise möglich. Bei größeren Flächen in Pixelgrafik stört ein deutliches Flimmern im Modell-4-Mode, in dem statt der 48 x 128 jetzt 72 x 160 Pixels möglich sind.

Eine Parallelschnittstelle für Drucker ist eingebaut, RS232 gibt es nur gegen Aufpreis.

Drei Betriebssysteme und zwei Computer auf einmal

Das Modell 4 kann sich als Modell 4 verhalten oder als Modell III oder als reine CP/M-Maschine. Was es gerade sein soll, merkt es selbst anhand der Diskette, mit der es hochgefahren wird. Fehlt sie ganz (oder ist nicht eingelegt) meldet es sich mit »Cass?«, womit es sich als Modell-III-Kassettenversion einstuft und nach der Baudrate fragt.

Im Modell-III-Mode arbeitet die Z80-CPU mit 2 MHz, kann aber per Software auf 4 MHz umgeschaltet werden. Der Cursor flackert dann recht nervös, was ihm mit einem POKE abgewöhnt werden kann, und die Echtzeit-Uhr zeigt Hausnummern an. Ansonsten läuft alles an Modell-III-Software dann auch, halt nur entsprechend schneller.

CP/M ist in der deutschen Version (Tastatur) noch nicht lieferbar (soll Ende 1983 kommen). In Amerika und hier habe ich bezüglich der US-Version widersprüchliche Aussagen gehört, der eine hat sie, der andere nicht. Ich fürchte, daß da einige CP/M-2.2 anbieten und nicht das von Tandy angekündigte CP/M-Plus (3.0). Von letzterem gibt es bekanntlich zwei Versionen (mit und ohne Bank-Select). Wie macht das nun Tandy, wenn jemand von 64 auf 128 KByte aufrüsten will?

TRS-DOS 6.1, das eigentliche Modell 4-DOS, ist absolute Spitze und sicherlich benutzerfreundlicher und komfortabler als CP/M. Für letzteres gibt es jedoch eine Unmenge Software, darunter einiges, wofür ich auch im großen Tandy-Angebot nichts Gleichwertiges finde. Daß CP/M erst in einigen Monaten verfügbar ist, hat mich dennoch nicht vom Kauf abgehalten, ein neuer Computer und gleich drei Betriebssysteme auf einmal ist eh zu viel für den Anfang.

Das Modell 4 in Stichworten

Z80-CPU, 4 MHz, im Modell-III-Mode 2/4 MHz umschaltbar
64 KByte RAM, erweiterbar auf 128 KByte
14 KByte ROM-Basic, ROM abschaltbar
2 KByte Video-RAM extra
Grüner Bildschirm, 24 Zeilen x 80 Zeichen
256 darstellbare Zeichen
Umschaltbar auf 16 Zeilen x 64 Zeichen
Breitschrift als 40 beziehungsweise 32 Zeichen/Zeile
Reverse Video, Blockgrafik, hochauflösende Grafik nachrüstbar
Deutsche Tastatur mit 75 Tasten, Autorepeat-Funktion
1 oder 2 Diskettenlaufwerke eingebaut, je 180 KByte, 2 weitere Laufwerke oder Festplatte anschließbar
Besonderheit: kann das Modell III emulieren
Betriebssysteme: TRS-DOS-6.1 oder CP/M-Plus
Im Lieferumfang: TRS-DOS-6.1 und Microsoft-Basic
Im Angebot: Assembler, Compiler-Basic, Alcor-Pascal, Cobol

Software ist genügend vorhanden, nur das Interessante fehlt

Ich habe Unmengen Software auf Modell-I-Disketten und einiges (ganz schnell geliehen) im Modell-III-Format. Zuerst: Im Modell-III-Mode läuft alles, was Original-Modell-III ist, ohne Fehl und Tadel, zum Beispiel Scripsit, mit dem dieser Artikel geschrieben wurde, und dieses Programm macht exzessiv von Tastatur, Schirm und Systemsoftware Gebrauch. Leider mit 16 Zeilen zu 64 Zeichen, und nicht mit den 24 x 80 des Modell 4-Modes.

Bei der Übertragung von Modell-I-Software gibt es natürlich die bekannten Probleme, wenn man in Maschinensprache arbeitet, nahezu unmöglich wird dies Richtung Modell 4, da paßt halt nichts mehr. In Basic ist das alles viel einfacher, auf die Unterschiede wird noch eingegangen. Wichtig ist, daß Sie alle Modell-I-/Modell-/III-Programme im ASCII-Format abgespeichert haben. Andernfalls hilft nur noch NEW-DOS/80, das in der Modell-III-Version noch ein Modell-I-Laufwerk emulieren kann.

Grund für das ASCII-Format ist, daß das Modell-4-Basic im gepackten Format die Befehle anders codiert. Eine Modell-I-Diskette kann man einfach mit REPAIR ins Modell 4-Format bringen, von einer TRS-DOS-1.3-Disk sind Files mit CONV auf Modell-4-Disketten zu übertragen. CONV akzeptiert aber keine Modell-III-NEW-DOS-Diskette. Folglich habe ich, der ich nur NEW-DOS besitze, damit ein Modell-I-Laufwerk emuliert, dann die Modell-III-Software auf die Modell-I-Diskette überspielt, und schließlich diese mit REPAIR ins Modell-4-Format gebracht. Wundersamerweise liefen die Programme (in Basic) dann immer noch, allerdings erst nach Korrektur eines Hauptfehlers. Modell-4-Basic erwartet nämlich, daß alle Schlüsselworte durch Blanks getrennt sind. »FORI = 1TO10« enthält danach drei Syntaxfehler.

Typische Modell-4-Software, die die Vorteile des neuen DOS und des Modell-4-Basic voll ausnutzt, gibt es hier noch nicht, schlimmer, wichtige Tools, die im Katalog stehen, den ich mir aus Amerika mitgebracht hatte, sind in Deutschland noch nicht erhältlich. Ich vermisse besonders das Assembler-Entwicklungspaket (149 Dollar) und Alcor-Pascal (jetzt auch Tandys Pascal) für 250 Dollar. Auch laut US-Katalog dürfen deutsche Anwender auf Cobol hoffen (199 Dollar), Bascom (195 Dollar), Visicalc (249 Dollar), Multiplan (199 Dollar) und ein auch im Singlestep arbeitendes Compiler-Basic mit Debugger für 149 Dollar.

Mitgelieferte Software wertvoll und nützlich

Zuerst: Es handelt sich nicht um die üblichen Soloprogramme, sondern alle sind schön in das System eingebunden. KSM/FLT zum Beispiel erlaubt eigene Texte unter 26 Tasten zu legen und mit SYSGEN kann das permanent gemacht (sozusagen ins DOS eingebunden) werden. Das gilt auch für die Communication-Software, die sehr komfortabel die RS232 unterstützt und in das Rout-/Link-Konzept paßt, womit zum Beispiel ein direkter Diskfile-Transfer möglich wird.

JCL, die Job Control Language, dürfte auch einen an Komfort gewohnten Arbeitsvorbereiter im Großrechenzentrum nichts vermissen lassen. Natürlich kann man auf einfachste Art mit BUILD den Standard-Batch-File schreiben, das ist aber noch keine JCL, obwohl sie


```
:0   [TRSDOS61] 5" Floppy #1, Cyls= 40, Dden, Sides=1, Step=12ms, Dly=.5s
:1   [No Disk ] 5" Floppy #2, Cyls= 40, Dden, Sides=1, Step=12ms, Dly=.5s
:2   [No Disk ] 5" Floppy #4, Cyls= 40, Dden, Sides=1, Step=12ms, Dly=.5s
:3   [No Disk ] 5" Floppy #8, Cyls= 40, Dden, Sides=1, Step=12ms, Dly=.5s
*KI <#  [*DU] X'FEAC'
*DO <=> X'0BB9'
*PR  #> [*SO] X'F7A9'
*SI <#  *KI
*SO <=> *DO
*JL  => Nil
*DU <=  X'0BED'
*SO  => X'0E0F'
Options: Spool, Type, KSM
```

Listing 1. Der Device-Befehl informiert über Geräte und Software im System

Listing 2. Mit dem neuen Basic wird vieles einfacher. Hier Hex/Dez und Dez/Hex-Konversion mit je nur einem Befehl.

```
10 INPUT"Eingabe in Hex (H) oder in Dezi (D)";E$
20 IF E$="D" THEN 80
25 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
30 ' ////// hex-dezi Konversion \\\\\\
40 INPUT "Hexzahl (stop mit 0)";H$
50 IF H$="0" THEN END ELSE PRINT VAL("&H"+H$): GOTO 40
60 '
70 ' /////// dezi-hex Konversion \\\\\\\
80 INPUT "Dezimalzahl (stop mit 0)";D
90 IF D=0 THEN END ELSE PRINT HEX$(D): GOTO 80
```

```
LIST
10 ' Programm listet die reservierten Worte des Model 4
20 ' (C) 1983 Peter Wollschlager. All rights released.
30 '
40 TABELLENZEIGER=22666
50 WORTANFANG=ASC("A")
60 IF PEEK(TABELLENZEIGER+1) THEN PRINT CHR$(WORTANFANG);
70 TABELLENZEIGER=TABELLENZEIGER+1
80 ZEICHEN=PEEK(TABELLENZEIGER)
90 IF ZEICHEN=0 THEN WORTANFANG=WORTANFANG+1: GOTO 60:'   Skip 1 oder 2 0-Bytes
100 IF WORTANFANG > ASC("X") THEN END
110 IF ZEICHEN<ASC("Z") THEN PRINT CHR$(ZEICHEN);:GOTO 70
120 PRINT CHR$(ZEICHEN AND &H7F);" ";:'                  Wortende, Reset Bit 7
130 IF POS(0)>76 THEN PRINT:'                            Ggf. Zeilenvorschub
140 TABELLENZEIGER=TABELLENZEIGER+1: '                   zeigt jetzt auf Token
150 GOTO 60
Ready

RUN
AUTO AND ABS ATN ASC CLOSE CONT CLEAR CINT CSNG CDBL CVI CVS CVD COS CHR$ CALL
COMMON CHAIN CLS DELETE DATA DIM DEFSTR DEFINT DEFSNG DEFDBL DEF DATE$ ELSE
END ERASE EDIT ERRS$ ERROR ERL ERR EXP EOF EQV FOR FIELD FN FRE FIX GOTO GO TO
GOSUB GET HEX$ INPUT IF INSTR INT INP IMP INKEY$ KILL LPRINT LLIST LPOS LET
LINE LOAD LSET LIST LOG LOC LEN LEFT$ LOF MERGE MOD MKI$ MKS$ MKD$ MID$ MEM
NEXT NAME NEW NOT OPEN OUT ON OR OCT$ OPTION PRINT PUT POKE POS PEEK RETURN
READ RUN RESTORE REM RESUME RSET RIGHT$ RND RENUM RANDOM ROW STOP SWAP SAVE
SPC( STEP SGN SQR SIN STR$ STRING$ SPACE$ SYSTEM SOUND THEN TRON TROFF TAB(
TO TIME$ TAN USING USR VAL VARPTR WAIT WHILE WEND WRITE XOR
Ready
```

Listing 3. Ein Programm und sein Ergebnis: Alle Befehle und Funktionen in alphabetischer Folge.

mancher Anbieter so nennt. Dinge wie Makros, Variable und IF/-THEN/ELSE machen eine gute JCL aus, hier sind sie vorhanden.

TRS-DOS-6.1 ist L-DOS

Zuerst zur Namensverwirrung. Im Copyright meldet es sich mit »TRSDOS 06.01.00«, im Disk-Namen mit TRSDOS61 und ansonsten heißt es TRS-DOS-6.1. Wie auch immer, das erste TRS-DOS schrieb seinerzeit Bill Schroeder als Tandy-Mitarbeiter. Dann hat er sich selbständig gemacht und schuf das berühmte L-DOS. Offensichtlich hat er sich von seiner alten Firma in Frieden getrennt, denn TRS-DOS-6.1 ist L-DOS-5.1. Tandy bietet L-DOS jetzt auch für die anderen Modelle an (sinnigerweise auch für das Modell 4, wo es unter anderem Namen zum Lieferumfang gehört), und schreibt in dem 84er Katalog, daß man neuerdings bevorzugt Software für L-DOS liefern wird.

L-DOS ist mit NEW-DOS/80 durchaus auf eine Stufe zu stellen, ist aber wesentlich benutzerfreundlicher und einfacher zu erlernen. Beeindruckend ist das Konzept der logischen Geräte, wonach zum Beispiel der Printer »★PR«, das Display »★DO und das Keyboard »★KI« heißen. Dazwischen und über den Weg sogenannter Phantom-Geräte (vom Anwender zu benennen) ist ein beliebiges Routing und Linking auch zu/von Disk-Files und Anwender-Routinen erlaubt. In die Links können eigene oder die mitgelieferten Filter-Programme geschaltet werden, um zum Beispiel Zeichen umzucodieren oder zu strippen. Mit dem SYSGEN-Befehl können solche Dinge permanent gemacht werden.

Ich meine allerdings, daß Einsteiger sich schwer tun werden, die Möglichkeiten dieser Technik auszuschöpfen, wenn sie nur das Handbuch lesen. Wer CP/Ms PIP und STAT kennt, tut sich da leichter.

Am frappierendsten werden die Unterschiede zwischen NEW-DOS und L-DOS am Beispiel Spooler sichtbar. Wer sich unter NEW-DOS schon mal mit ASPOOL/MAS versucht hat, wird staunen. Unter L-DOS reicht schlicht SPOOL. Damit wird automatisch ein 1-KByte-Druckerpuffer geschaffen und alle Ausdrucke laufen ab sofort über diesen. Ist Ihnen 1 KByte zu wenig und sie hätten lieber 10 KByte, dann geben Sie einfach »SPOOL (MEM = 10)« ein. Wird es im Arbeitsspeicher eng, können Sie ein Disk-File hinzunehmen und geben dazu nur den Namen und die File-Größe in KByte an.

Auch in diese Rubrik fällt MEMDISK, was erlaubt, das Memory als Disk zu benutzen. Richtig Spaß macht das zwar erst in der 128-KByte-Version, aber mit 64 KByte geht's auch. Ich habe so eine 10-KByte-Diskette im Background, wo ich blitzschnell Zwischenergebnisse speichere.

So schöne Dinge wie SPOOL und MEMDISK sollte man sich allerdings überlegen, bevor man vom DOS ins Basic geht. Das sind nämlich die einzigen DOS-Routinen, die man nicht unbeschadet von Basic aus initialisieren, wohl aber da benutzen kann. Noch ein Bug, und zwar ein großer: ich habe noch nicht herausbekommen, wie man nach einem versehentlichen Verlassen von Basic da wieder hinkommt und sein Programm unbeschadet vorfindet. Der Befehl »BASIC★« von NEW-DOS fehlt. Das Problem dürfte lösbar sein. Ich habe es mir vorgenommen, sobald ich über einen Assembler verfüge.

Schon aus Gründen der Kompatibilität ist alles an TRS-DOS/alt-Befehlen auch hier vorhanden. Von den Extras wären hervorzuheben:

Nach dem Kaltstart erscheint die (abschaltbare) Frage nach dem Da- ►

tum als MM/DD/YY, die dann als Text quittiert wird, sofern man nicht versucht, den 29. Februar 1983 oder ähnliches einzugeben, was das Kalender-Programm nicht mag. Das Datum wird im File-Header mit aufgezeichnet und kann zur selektiven Behandlung von Files genutzt werden. Ein ebenfalls geführtes »Mod-Flag« erlaubt, nur Dateien zu kopieren, die seit dem letzten Datum geändert worden sind.

BACKUP kann als Spur für Spur-Kopie laufen oder als File-Kopie mit Krunch-Funktion (die Diskette wird aufgeräumt).

DEVICE liefert ähnlich CP/Ms STAT Informationen über alle Geräte im System, aber auch über User-Options wie Spool (Listing 1).

FILTER lenkt ein logisches Gerät über ein Filter-Programm.

FORMS ist ein vorhandenes Filterprogramm zur Druckereinstellung beziehungsweise Formatierung.

MEMORY erlaubt »Lomem« und »Himem« zu setzen.

PATCH ist ein Programm zum direkten Ändern von Disk-Sektoren; es arbeitet auch Patch-Files ab.

REMOVE entfernt Files; es darf eine Liste angegeben werden.

PURGE erlaubt globales »Remove« mit Rückfrage.

SYSGEN bindet eine Anwender-Konfiguration ins DOS ein und

SYSTEM erlaubt ihre Anpassung/Eingabe.

BLINK: Blink-Frequenz (auch 0) und Größe des Cursors.

SYSRES: Laufwerke physikalisch anpassen.

System-Drive kann ein anderer als 0 sein.

TRACE gibt den Z80-PC aus.

ALIVE erzeugt einen sogenannten »alive bug«, ein Grafiksymbol, das sich in sich ändert, solange das System nicht »hängt«.

Ein neues Basic von Microsoft

Das Basic ist total RAM-resident und wird immer von der Diskette geladen. Damit sind Korrekturen und Erweiterungen leicht möglich und billig.

Nun zum Handling von Random-Files: Es läuft in schlechter M80-Tradition immer noch über Fielding, LSET und CVx. Ein CIM-Dump in Basic wäre auch noch zu installieren und der Editor ist der alte, also verbesserungsbedürftig, sofern man sich nicht Wordstar leisten will.

Der Kern bildet das alte Level-2-Basic. Einiges wurde [illegible]

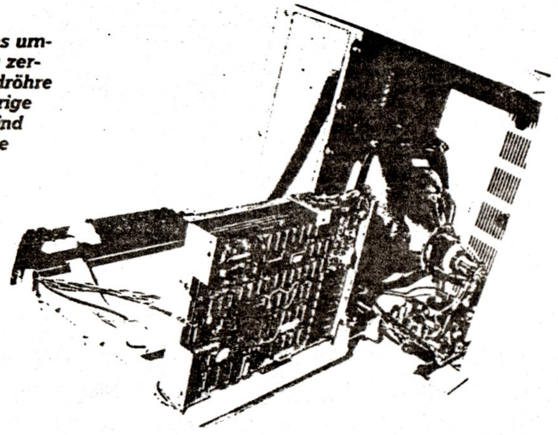


Bild 2. Etwas umständlich zu zerlegen — Bildröhre und zugehörige Schaltung sind in der Haube eingebaut

vieles kam hinzu, aber es ging auch etwas verloren. CLOAD und CSAVE gehören dazu, was nicht zu beklagen ist, aber auch SET, RESET und POINT. Die Grafik-Symbole gibt es zwar noch, wenn auch wegen der jetzt 24 Zeilen leicht gestaucht, man kann sie aber nur noch mit PRINT§ auf den Schirm bringen. Letzteres erlaubt jetzt auch die Angabe von Zeile/Spalte. POINT kann auch nicht mehr mit PEEK realisiert werden, weil der inzwischen ausgelagerte Video-RAM sich dem entzieht. Tandy bietet allerdings eine Supergrafik-Karte an mitsamt vielen mächtigen Grafik-Befehlen. Ein Testbericht soll folgen.

Gehen wir alphabetisch vor, so beginnen die interessanten neuen Dinge bei C wie CALL. Der Befehl erlaubt, Routinen in Maschinensprache aufzurufen und beliebig viele Variablen zu übergeben. USR blieb erhalten, wurde aber auch erweitert. Noch unter C finden Sie zwei ganz wichtige Dinge, nämlich CHAIN zum Hereinziehen von Overlays und Fortrans COMMON, damit dann die Parameterübergabe auch klappt.

CLEAR hat eine ganz neue Bedeutung, Stringspace muß nicht mehr reserviert werden. Es ermöglicht, im Basic-Programm Mem-Size und Stackgröße zu setzen. Mit letzterem kann man ein paar Byte sparen, wenn mangels geschachtelter FOR-NEXTs oder GOSUBs der Basic-Stack bescheiden bleibt. DATE$ und TIME$ ergeben die entsprechenden Daten.

Die globale Deklaration von Variablen-Typen wie DEFINT blieb (Gott sei Dank) erhalten, Variablennamen dürfen jetzt aber bis zu 40 signifikante Zeichen haben, was wesentlich zur Lesbarkeit und

Wartbarkeit der Programme beiträgt und zahllose REMs erspart. ERL ergibt jetzt die echte Fehlerzeile, ERR gibt es noch, neu ist ERR$, das die Fehlermeldung im Klartext ausgibt.

ERASE erlaubt dynamisches Löschen von Arrays. FOR-Schleifen werden nicht mehr sinnlos ausgeführt, wie es bisher zum Beispiel in »FOR I = 1 TO 0« der Fall war.

Vielleicht störend bei der Übernahme von Modell 3-Programmen kann sein, daß bei der Umwandlung Fließkomma in Integer nicht mehr abgeschnitten sondern gerundet wird.

HEX$ und OCT$ als die Umkehr von &H und &O kamen hinzu und machen die so schönen Konversionsprogramme glatt überflüssig. INKEY$(n) als Erweiterung dazu, macht auch Listings zu diesem Thema obsolet. Bei J gibts nichts, zu O wäre OPTION BASE zu erwähnen, was erlaubt, das selten gebrauchte nullte Element von Arrays zu streichen. POS(0) ergibt wie gehabt die Cursorposition, ROW ist neu und »returned« die Cursor-Zeile und LPOS(0) gilt für den Drucker. Mit WIDTH kann neuerdings die Zeilenlänge für Video und Drucker eingestellt werden.

RESTORE jetzt mit Zeilennummer und RENUM mit drei Argumenten (wohin, woher, Inkrement) ist fest eingebaut.

SWAP tauscht die Inhalte zweier Variablen, SYSTEM »DOS-Befehl« ersetzt CMD»x«. WAIT wartet auf ein zu definierendes Ereignis an einem Port.

WHILE/WEND bringt etwas Pascal ins Basic und WRITE, wie ►

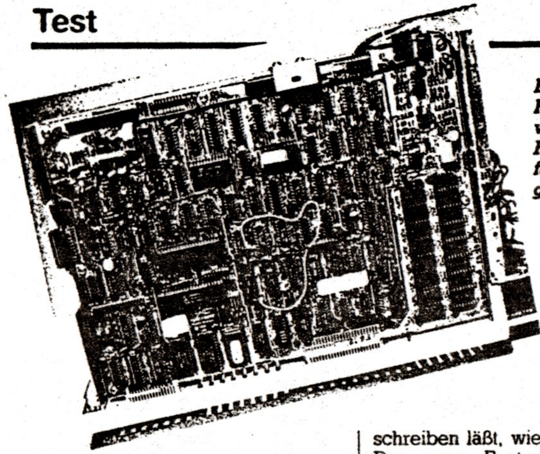


Bild 3. Die Hauptplatine weist noch von Hand durchgeführte Änderungen auf

PRINT anzuwenden, druckt die Variablen mit Kommas getrennt und Strings in Anführungszeichen. Auf dem Schirm sieht das gut aus, als WRITE# auf Files angewandt erspart es, selbst Delimeter vorzusehen.

Befehle, die nicht im Handbuch stehen

In der Einleitung des Manuals wird dem Besitzer zum Kauf gratuliert und ein kurzer Abriß über die Vorteile des Modell 4 gegeben, und da steht auch »sound generation« und mehr nicht. Im Anhang findet man dann noch in der Liste der Tokens wiederum SOUND und das war's dann auch. Nach der Methode »trial and error« ergab sich folgende Syntax: »SOUND Ton, Dauer«. Dabei darf Ton die Werte 0 bis 7 annehmen, was in etwa einer Tonleiter G-Dur entspricht, und Dauer darf 0 bis 31 betragen in grob 0,5-Sekundenschritten. Klangqualität und Lautstärke sind nicht berühmt.

Auch in der Tokenliste habe ich XOR entdeckt, die Syntax ist klar, vielleicht auch noch EQV, das wie alle Logik-Operatoren auf Bit-Ebene arbeitet und bei gleichen Bits 1 sonst 0 erzeugt. Etwas länger habe ich an IMP geknobelt. Was das heißt, weiß ich nicht. Vielleicht kann mir da ein Leser helfen, jedenfalls gilt »A IMP B = NOT (A OR B)—B« (Anmerkung der Redaktion: Wahrscheinlich handelt es sich um die logische Implikation). Eine Anwendung dafür sehe ich noch nicht, jedoch für den Schrägstrich links, was DIV, also Ganzzahldivision heißt sowie für MOD, also Modulo.

Auch nicht erwähnt wird, daß das altbekannte &H jetzt sogar in Ausdrücken erlaubt ist, womit sich eine höchst simple Dezi-Hex-Konversion

schreiben läßt, wie Listing 2 zeigt. Das neue Feature HEX$ wird gleich mit demonstriert.

Daß Variablen-Namen jetzt bis zu 40 Zeichen (signifikant) lang sein dürfen, habe ich mit Freude registriert und prompt in CBasic-Art den Punkt mit eingebaut. Auch das funktioniert, obwohl es nicht im Manual steht. In die Rubrik gehört noch INPUT, was neuerdings (wozu?) als Direkt-Statement erlaubt ist.

So belehrt mißtrauisch und hoffnungsvoll wollte ich wissen, ob denn nun wenigstens die Token-Tabelle im Handbuch vollständig ist und fing an, das Pendant im RAM zu suchen. Sie war im Prinzip schnell gefunden, aber nicht so einfach zu knacken. Wie, zeigt Listing 3 mitsamt dem Ergebnis. Dahinter steht folgendes: Alle Texte sind alphabetisch sortiert und in Gruppen je Anfangsbuchstabe eingeteilt. Jede Gruppe endet mit einem 0-Byte, folglich bedeuten zwei aufeinanderfolgende Nullen, der Buchstabe, zum Beispiel Q, ist nicht besetzt. Da so der Anfangsbuchstabe je Wort vorgegeben ist, wird er denn auch weggelassen. Das Wortende wird durch ein gesetztes Bit 7 im letzten Buchstaben markiert, direkt dahinter folgt das Token, ein 1-Byte-Code, der anstelle der langen Basic-Worte im Programmtext eingetragen wird. Um Tokens von anderen Zeichen unterscheiden zu können, sind sie in der Regel ≧ 128. Da ein Byte nicht mehr als 255 darstellen kann, sind maximal 128 Tokens möglich, das ging bisher ganz gut, das neue Basic hat aber 140 Tokens!

Gelöst wurde das so: Das »Token« 255 bedeutet, das folgende ist das richtige, hat aber eine andere Bedeutung. Diese Sondertokens stehen in der Tabelle im Handbuch als Zahlen > 65280. Das • ist 255 ★ 256 + Tokenwert.

Im Programmtext sind auch diese Tokens ≧ 128, nicht aber in der RAM-Tabelle, da sind von ihrem Wert jeweils 128 subtrahiert, womit man sie von den anderen unterscheiden kann, ohne die 255 davorzusetzen.

Nebenbei gesagt hatte ich auch Probleme, ein Basic-Programm im RAM zu finden, denn im Handbuch steht folgende Memory-Map:

0000H-2F5FH DOS
2600H-2FFFH Overlays
3000H-85FFH Basic
8600H-Himem Programm

Das ist insofern falsch, als Basic-Programme schon bei 7F0CH beginnen. Mag sein, daß eine geänderte Basic-Version, zum Beispiel die für Grafik, mehr verbraucht, die kenne ich aber noch nicht. Zu den Handbüchern ist nicht viel zu sagen, bekannte Tandy-Qualität in Menge. Neu ist »Getting started« als extra Heftchen und eine Referenz-Karte. Zu bemängeln wäre: Das DOS-Manual ist nicht gedruckt, sondern direkt von der Schreibmaschine kopiert, vielleicht kommt's noch besser. Für den Nicht-Diskbasic-Teil gibt es ein extra Buch, gemeinsam für Modell III und 4, nach dem Motto, das alte Modell-III-Buch mit Hinweisen auf Modell-4-Features. Man hat allerdings versäumt zu sagen, daß die meisten der aufgeführten POKEs im Modell 4 Nonsens sind.

Fazit: Eine solide Maschine mit einem guten DOS und Basic, offen für CP/M und andere Betriebssysteme. Der Anschluß an die Vorgänger und ihr Softwarepotential ist auch gelungen. Vom Konzept und vom Preis her eindeutig Richtung kommerzielle Anwendung orientiert, aber noch preiswert genug als Homecomputer für gehobene Ansprüche.

(Peter Wollschlaeger)

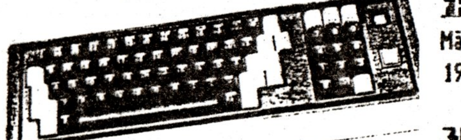


Bild 4. Sehr schön: deutsche Tastatur mit 10er-Block

Die folgende Schaltung stammt aus 80 MICRO Mai 81 und ermöglicht, daß man an beliebigen Stellen im Text einzelne oder mehrere Zeichen invertieren kann, d.h. schwarze Zeichen auf weißem Hintergrund. Leider hat die Sache aber einen Schönheitsfehler und der ist, daß inverse Zeichen und Grafik-Zeichen nicht gleichzeitigt auf dem Bildschirm dargestellt werden können. Da aber die Anwendung der Invers-Darstellung vorwiegend im Textbereich liegt, ist dieser Nachteil meiner Meinung nach zu verschmerzen.

Da ich mit dieser Schaltung in der Lage bin ein einzelnes Zeichen zu 'adressieren' lag der Gedanke nahe die Schaltung so umzubauen, daß man an der gewünschten Stelle auch die Helligkeit des Zeichens umschalten kann und damit einzelne Wörter oder Textabschnitte in den Hintergrund schieben oder auch besonders hervorheben kann. Die gesamte Schaltung ist auf 2 Bilder verteilt. Bild 1 zeigt die etwas modifizierte Schaltung (für TRS-80!) aus 80 MICRO, Bild 2 zeigt die Adressierlogik. Inverse Darstellung und Helligkeitsumschaltung können HW-mäßig über die Schalter S1 bzw S2 und SW-mäßig über die Befehle:

OUT 251,2 = Helligkeit schwach

OUT 251,4 = Inverse Darstellung

umgeschaltet werden. Als Treiberprogramm muß entweder das Maschinenspracheprogramm (Listing1) oder das Basic Programm (Listing2) geladen werden. Nachfolgend habe ich die Beschreibung dieser Schaltung aus 80-MICRO übersetzt, damit Ihr auch wißt wie die Sache funktioniert.

Die Umschaltung erfolgt durch Bit 7. Sieht man sich das Schaltbild des TRS-80 an, so sieht man das Bits 0 bis 5 von den Video Memory ICs an IC Z28 gehen und Bits 6 und 7 zusammen mit ein paar anderen Signalen an IC Z27 gehen. Diese beiden ICs sind einfach Zwischenspeicher welche die Daten puffern die vom Video Memory kommen.

Bis hierhin ist Bit 7 immer noch im Takt mit den anderen Datenbits. Von Z29 und von Z8 aus gehen die Bits dann zu den Schieberegister Z10 und Z11 die dazu dienen, die von den Zeichengeneratoren kommenden Zeichen Bit für Bit auszugeben.

Mit Bit 7 wird entschieden, welches der Schiebe- register die Daten erhalten soll und somit ob ein normales Zeichen oder ein Grafiksymbol ausgegeben werden soll. Das Problem ist nur, daß Bit 7 gerade solange ansteht um das Schieberegister zu laden. Zu der Zeit, in der die Bildpunkte vom Register ausgegeben werden, sind aber die Bits an dem Datenpuffer inclusive Bit 7 bereits die, des nächsten Zeichens welches ausgegeben werden soll. Die Konsequenz ist, daß das inverse Zeichen immer um eine Zeichenbreite nach links versetzt ist und damit nicht an der Stelle wo man es haben wollte.

```
            ;*************************************************************
            ;* LOWER CASE AND INVERSE CHARACTER DRIVER PROGRAM           *
            ;* (DISK VERSION)                                            *
            ;*************************************************************
            ;
007F        RVCODE  EQU     7FH             ; REVERSE CONTROL CODE
401E        VIDEO   EQU     401EH           ; VIDEO DCB ADDRESS
4049        HIMEM   EQU     4049H           ; DOS HIGH MEMORY POINTER
            ;
FFB0                ORG     0FFB0H
            ;
FFB0 21BFFF LOADER  LD      HL,START        ; GET DRIVER ADDRESS
FFB3 221E40         LD      (VIDEO),HL      ; AND INSERT IT INTO THE
                                            ;   VIDEO DEVICE CONTROL
                                            ;   BLOCK
            ;
            ; THE FOLLOWING TWO INSTRUCTIONS AUTOMATICALLY SET THE
            ; DOS HIGH MEMORY POINTER.  THIS ONLY APPLIES TO NEWDOS
            ; (2.1 OR 80) AND TRSDOS 2.2 OR LATER.
            ; FOR OTHERS, THEY SHOULD BE DELETED.
FFB6 21BEFF         LD      HL,START-1      ; GET MEMORY SIZE
FFB9 224940         LD      (HIMEM),HL      ; AND SAVE IT
            ;
FFBC C32D40         JP      402DH           ; RETURN TO DOS
            ;
FFBF DD6E03 START   LD      L,(IX+3)        ; GET THE ADDRESS OF
FFC2 DD6604         LD      H,(IX+4)        ; THE CURSOR
FFC5 DA9A04         JP      C,049AH         ; IF READING FROM SCREEN,
                                            ;   THEN RETURN TO ROM
            ;
FFC8 79             LD      A,C             ; GET THE CHARACTER
FFC9 FE7F           CP      RVCODE          ; IF IT'S NOT REVERSE CODE
FFCB 2009           JR      NZ,DISPLY       ;   THEN GO TO "DISPLY"
FFCD 3AEFFF         LD      A,(MASK)        ; GET THE REVERSE MASK
FFD0 EE80           XOR     80H             ; AND TOGGLE IT
FFD2 32EFFF         LD      (MASK),A        ; SAVE THE NEW MASK
FFD5 C9             RET                     ; RETURN WITHOUT DISPLAY-
                                            ;   ING THE CHARACTER
            ;
FFD6 DD7E05 DISPLY  LD      A,(IX+5)        ; IF THE CURSOR
FFD9 B7.            OR      A               ; IS ON
FFDA 2801           JR      Z,GETCHR
FFDC 77             LD      (HL),A          ;   THEN BLANK IT
            ;
FFDD 79     GETCHR  LD      A,C             ; GET THE CHARACTER
FFDE FE20           CP      20H             ; IF IT'S A CONTROL CHAR.
FFE0 DA0605         JP      C,0506H         ;   THEN JUMP BACK TO ROM
FFE3 FE80           CP      80H             ; IF IT'S A GRAPHIC CHAR.
FFE5 D2A604         JP      NC,04A6H        ;   THEN JUMP BACK TO ROM
FFE8 3AEFFF         LD      A,(MASK)        ; GET THE INVERSION MASK
FFEB B1             OR      C               ; AND 'OR' IT WITH CHAR.
FFEC C37D04         JP      047DH           ; JUMP BACK TO ROM
FFEF 00     MASK    DEFB    00              ; REVERSE MASK (INITIALLY
                                            ;  SET FOR NORMAL DISPLAY)
            ;
FFB0                END     LOADER          ; AUTO-EXECUTE DRIVER
```

Program Listing 1. Disk based inverse video driver.

Die Lösung ist der Einbau eines neuen Schieberegisters für Bit 7, welches das Signal für inverse Zeichen liefert. Das neue Schieberegister welches ich Z 10a nenne, liegt parallel zu Z10, sodaß sein Ausgangssignal gleich mit dem Signal für normale Zeichen ist.

Die Umschaltung zwischen normalen Zeichen und inversen Zeichen geschieht durch die Logik nach Bild 2. Im normalen Darstellungsmodus ist Pin 8 von IC X2 high, d.h. Pin 4 von Z 26 low und Pin 10 von Z26 ist high, wie bei der Darstellung von alphanummerischen Zeichen. Wenn im Invers Modus Bit 7 high ist, liegt 2 von Z27 auf low. Der Unterschied zwischen diesem Signal und DLY 7 wird von Z42a erkannt, dessen Pin 3 dadurch high wird. Dieses High geht nun zu Z 10a welches dafür sorgt, daß die Bits im gleichen Takt wie bei Z10 ausgegeben werden. Die Ausgangssignale von Z10 und Z10a gehen dann auf ein Exclusiv-Oder wodurch die Zeichen invertiert werden.

Wird auf normale Darstellung umgeschaltet, d.h. Bit 7 an Z 27 ist low, so wird Pin 3 von Z 42a low, weil über die Gatter auf Bild 2 an Pin 1 und 3 von Z 42a das gleiche Signal liegt. Da jetzt der Ausgang von Z 10a ebenfalls low ist, werden die Bits von Z 42a nicht invertiert.

Z 26 bewirkt, daß Z10 alphanummerische Zeichen lädt, wenn die Signale CLK, DLY Bit 7, DLY BLANK und DLY L8 alle low sind. Z 26a bewirkt das gleiche für Z 10a, benutzt jedoch nicht das DLY L8 Signal, da DLY L8 dazu dient die 4 unbenutzten Zeilen unter jedem Zeichen zu erzeugen. Somit werden diese 4 Zeilen immer invertiert.

Voraussetzung für inverse Darstellung ist, daß Kleinschreibung installiert ist, da in einem TRS-80 ohne Kleinschreibung Bit 6 auf Low gezwungen wird wenn Bit 7 high wird. Deshalb werden alle inversen alphanummerische Zeichen als inverse Steuerzeichen dargestellt.

Helligkeitsumschaltung

Die Helligkeitsumschaltung funktioniert prinzipiell genauso wie die Invers-Umschaltung. Nur wird hier das Ausgangs- signal des Schieberegisters Z10a Pin 13 über IC X5 auf die Basis des Video Transistors Q1 gegeben.

Beide Umschaltungen werden mit 'CHR' aktiviert.

BEISPIEL: PRINT CHR;"HALLO";CHR

Basic-Treiber

```
 20 CLEAR200:CLS
 50 MS=PEEK(13561)+PEEK(16562)*256
 60 MS=MS-49
 80 POKE16561,MS-(256*INT(MS/256))
 90 POKE16562,INT(MS/256)
100 M=MS
110 IFM>32767THENM=M-65536
130 FORI=M+1TOM+49
140 READN
150 POKE I,N
160 NEXT I
180 MH=INT((MS+49)/256)
190 ML=(MS+49)-(256*MH)
200 POKE M+16,ML:POKE M+17,MH
210 POKE M+43,ML:POKE M+22,MH
220 POKE M+43,ML:POKE M+44,MH
250 POKE16414,(MS+1)-(256*INT((MS+1)/256))
260 POKE16415,INT((MS+1)/256)
270 CLEAR:END
290 DATA 221,110,3,221,102,4,218,154,4,121
300 DATA 254,127,32,9,58,0,0,238,128,50
310 DATA 0,0,201,221,126,5,183,40,1,119
320 DATA 121,254,32,218,6,5,254,128,210,166
330 DATA 4,58,0,0,177,195,125,4,0
```

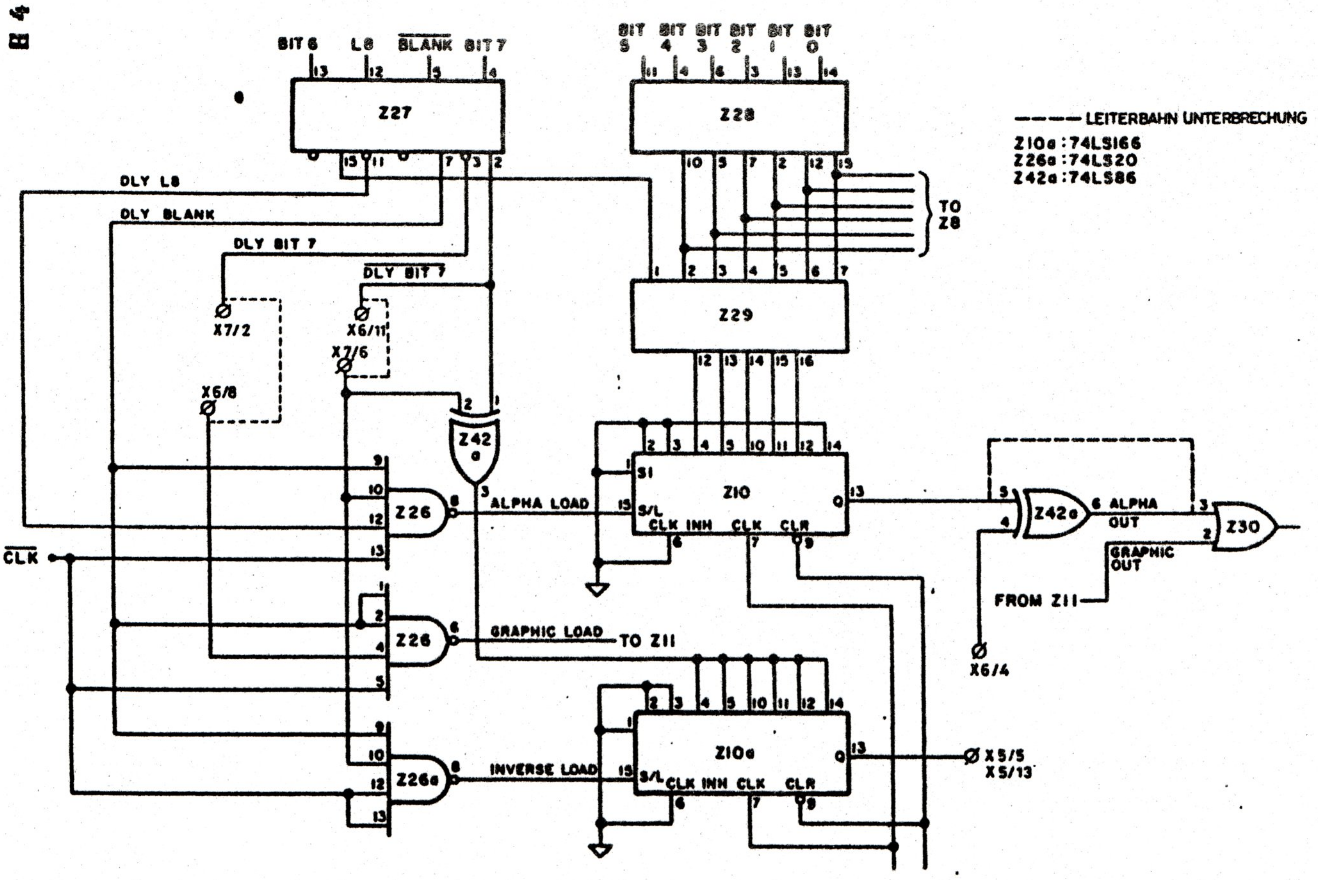
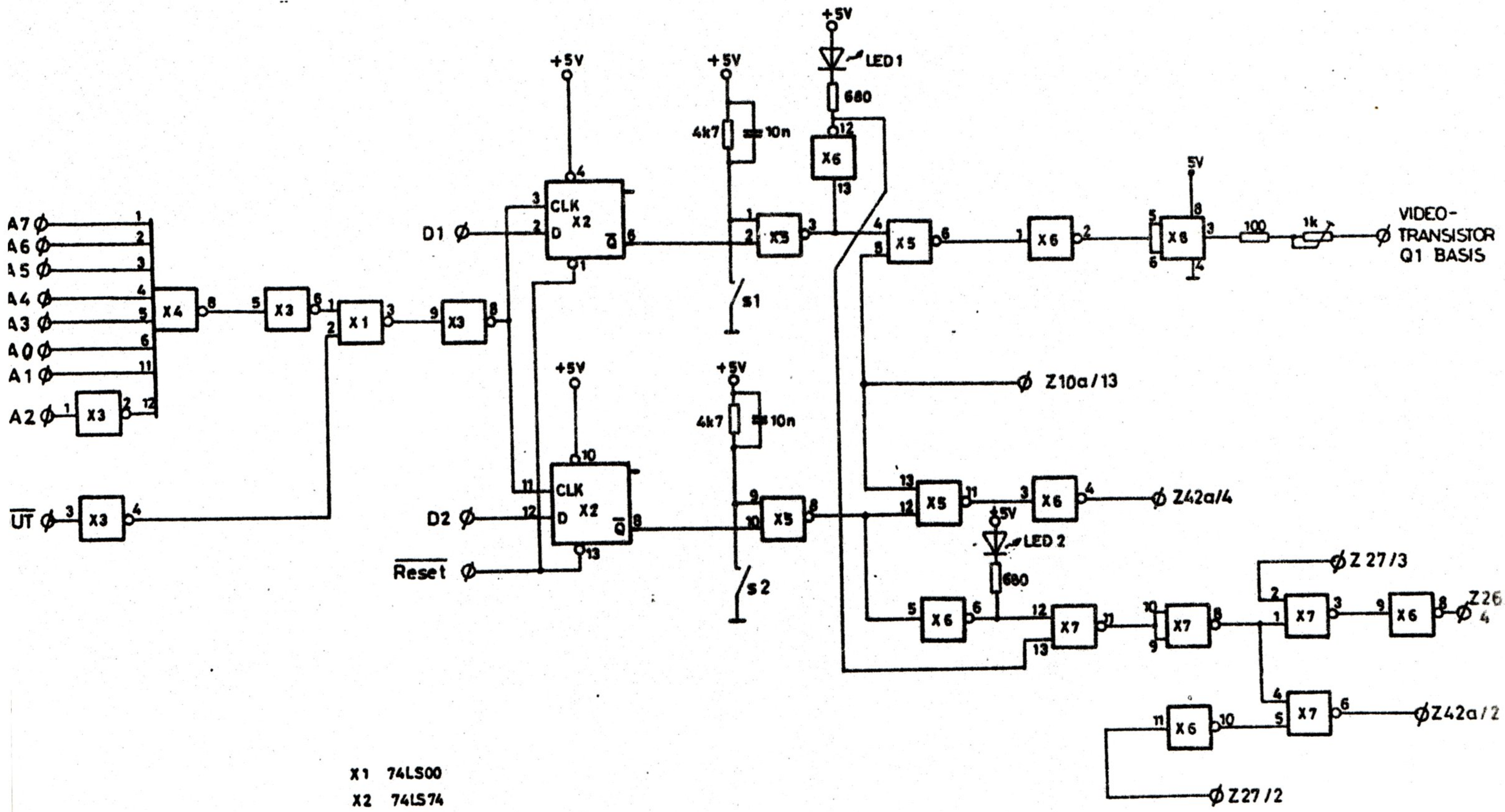


Bild 1



X1	74LS00
X2	74LS74
X3	74LS04
X4	74LS30
X5	74LS00
X6	74LS04
X7	74LS00
X8	75452

Bild 2

BÖRSE

TRS-80 Modell III zu verkaufen
einschließlich:
- Cassettenrecorder CTR-80A
- umfangreiche Software und Literatur (inkl. ROM-Listing)
Preisvorstellung ca. 600 DM

Außerdem abzugeben ist ein Fernschreiber Lorenz C15 mit Beschreibung, diente lange Zeit als erster Drucker am GENIE II. Preisvorstellung ca. 100 DM

Tel. 02948/542 R.Jablotschkin

Verkaufe

* Genie I, 64K 400.–
* Floppy-Contr. 200.–
* Doppler 120.–

möglichst als Einheit für 700.–

80-Zeichenkarte für Genie I + Software 150.–

Herbert Alber
Niedereschach

Das System besteht aus dem KOMTEK-1s, einem ZENITH-Monitor, dem SEIKOSHA-Drucker GP100A und zwei BASF 6108 DS-Laufwerken incl. Controller, Netzteil und Gehäuse sowie Software, Büchern und Unterlagen.

Verhandlungsbasis für das ganze Paket wären 1200,- DM. Ich würde jedoch die Geräte auch einzeln verkaufen.

Sollte jemand Interesse an dem System haben, so kann er mich am Tage unter der Rufnummer 040/215011-15 erreichen.

Hans Jürgen Bozek

RB - RS-232 Schnittstelle zu verkaufen; kompl. mit Software und allen Kabeln; passend für TRS80 MI und Genie I + II. Preis: 120,-DM. Peter Spieß, Trugenhofenerstr. 27, 8859 Rennertshofen 1

Verkaufe:

TRS 80 Modell I, 48 KB RAM, 2 x 40 Tr. LW (SS,DD)

VB: 1450,-- DM

Klaus Hermann (Tel. 07127/70024)

Nachdem ich mir kürzlich zwei Laufwerke angeschafft habe, ist bei mir das Druckerinterface EG 3016 über. Ich wäre Dir zu unendlichem Dank verpflichtet, wenn Du das gute Stück in der Börse anbieten könntest. Als Verhandlungsbasis habe ich mir DM 150,-- vorgestellt. Dienstlich bin ich z.Zt. unter der Rufnummer 05401/2422 zu erreichen.

Mit freundlichen Grüßen

Martin

Verkaufe Apple-Schaltnetzteil

Die technischen Daten wie folgt:
+5V/3,5A .. -5V/0,5A .. +12V/2A .. -12V/0,5A
Preis 70,--DM
Ulrich Böckling Tel.: 0261/69522

Genie IV s zu verkaufen!

Üblicherweise lassen wir bei einem Verkauf vom Chefredakteur eine entsprechende Notiz formulieren oder schicken für die Flohmarktecke zwei, drei selbstgestrickte Sätze. Die Kiste, die ich anzubieten habe, ist jedoch so heiß, daß sich eine längere Beschreibung lohnt. Schuld daran ist Helmut Bernhardt, der seine Hard-Ideen wohl vorsichtshalber gerne an meinem Computer ausprobierte. Das Gerät ist deshalb gespickt mit seinen digitalen Schmankerln. Schuld ist auch meine Sucht, aus dem Genie ein Universalgenie zu machen. Deshalb hat es auch dies und jenes von RB-Elektronik und von Udo Jourdan. Und schuld ist schließlich meine Bank, die mir nicht böse ist, wenn schon wieder ein gewisser Betrag ins Hobby fließt. Schließlich leben die davon, daß unsereiner Schulden hat.

Das Wichtigste und Beste in Kürze, falls ich nicht wortreich in Marktschreierei entgleite: Das Gerät ist ein ehemaliges Genie I. Der neue Name in der Überschrift ist natürlich ein Scherz, aber vom Genie I unterscheidet sich der Computer tatsächlich in so vielen Punkten, daß ich über den Scherz eigentlich nicht lachen kann: 256 kB freies RAM, vorbereitet auf 1 MB (wer fit mit dem Lötkolben ist, macht das in einer halben Stunde). Da die CPU nur 64 kB gleichzeitig erreichen kann, gibt es ein ausgeklügeltes Banking-System:

In Portionen zu 32 kB kann man wahlweise in der oberen oder unteren Speicherhälfte (0000-7FFF oder 8000-FFFF) einfach die Bank wechseln. Da es auch unten geht, ist der Computer für CP/M (jedwede Version!) geeignet. Der Banker kann außerdem bestimmte Bereiche des Speichers getrennt für Lese- und Schreibzugriffe enablen. Das bedeutet, daß man z. B. für LPRINT CHR$(10) nicht extra eine Utility laden muß, sondern einfach den Interpreter direkt umpoket. Und schließlich kann Helmuts Banker die I/O-Adressen für Bildschirm, Tastatur, Floppy usw. ins Himem legen, so daß CP/M oder ein x-beliebiges anderes System mit der Maschine klarkommt.

Da ist ein Speed-Up drin, bei dem man mitten in der Arbeit von 1,77 auf 3,54 MHz und zurück schalten kann. Der ziemlich lendenlahme Cursor von SCRIPSIT wird plötzlich ganz schön munter. Bei PACMAN hat der Spieler keine Chance mehr, es sei denn, er schaltet rechtzeitig auf 1,77 MHz.

Die HRG 1b ist ebenfalls eingebaut. Sie hat eine Matrix von 384 X 192 Punkten. Sie ist nachträglich auf 16 kB aufgemotzt (ursprünglich 12 kB). Dieser Speicher geht nicht dem Hauptspeicher von 64 kB verloren, denn die RAMs befinden sich auf der Zusatzplatine. Wenn man nicht gerade eine Graphik bearbeitet, steht dieser Speicher für beliebige Zwecke als Datenfriedhof zur Verfügung. Außerdem ist die HRG nachträglich sauber dekodiert. Sie braucht nur 6 Ports, verseuchte aber ab Werk so ziemlich den kompletten I/O-Adreßraum bis zum Port 127. Jetzt sind durch sie nur noch die Ports 0-7 belegt.

Da ist auch ein Interrupt-Timer. Er generiert wahlweise nichtmaskierbare Interrupts (NMI) oder maskierbare (INT). Seine Frequenz läßt sich in weiten Grenzen einstellen. Mit Floppy kriegt man seine Interrupts von ihr. Aber dann ist das Ding dennoch interessant, weil man in regelmäßigen Abständen NMIs ausgeben kann, die irgendeine Funktion erfüllen. Sie verzweigen beim Z80 an die Speicherstelle 0066h, wo man dank des Bankers eine eigene Routine unterbringen kann. Damit lassen sich Programme überlisten, die die maskierbaren Interrupts abschalten, um sich nicht in die Karten gucken zu lassen.

Der Adreßraum 3800-3BFF ist bei der Computerfamilie unseres Clubs der Tastatur vorbehalten. Helmut hat es fertiggebracht, die Tastatur auf 256 Bytes zurückzustutzen, so daß der Bereich 3900-3BFF für sonstiges zur Verfügung steht. Er nannte das in der c't 5/85 "Ein sicheres Plätzchen". Wahrlich! Kein Programm für TRS-80 oder Genie stört diesen Speicherbereich (außer H-DOS, versteht sich, aber davon später mehr).

Ein weiterer Speicherzugewinn ist dadurch entstanden, daß Helmut eine Platine entwarf, die den Adreßraum 3000-37DF versechzehnfacht. Das sind runde 32 kB. Auf dieser Platine sind 16 Steckplätze für EPROMs oder statische RAMs. Einer davon ist vom Level-IV-EPROM von TCS belegt. Der Rest (bisher gut zur Hälfte bestückt) kann andere Speicher aufnehmen, so daß man beispielsweise ständig einen HRG-Treiber verfügbar haben kann o. ä..

Der eingebaute Lautsprecher ist regel- und abschaltbar, falls das Gequietsche bei DIGOUT einmal nerven sollte. Eine Ohrhörerbuchse hat er auch, um den Ärger mit den Nachbarn und der Familie in Grenzen zu halten.

Die Tastatur ist vollständig mit Tasten bestückt. So sind die deutschen Umlaute, das ß, der Cursor usw. über die Tastatur erreichbar. Der Interpreter, auch Programme wie TSCRIPS spielen dabei mit, so daß man sich die CTRL-Akrobatik für diese Zeichen schenken kann. Zusätzlich gibt es 7 programmierbare Funktionstasten, die ganze Befehlsstrings oder sonstwas mit einem Tastendruck ausgeben können. Die Strings dürfen auch Steuerzeichen wie NEW LINE etc. enthalten.

Es gibt eine Anschlußbuchse für eine komplette zweite Tastatur und eine für einen Joystick. Bei letzterer habe ich zwei Kabel beim Löten vertauscht, so daß bei einer Cursorrichtung stattdessen ein Buchstabe ausgegeben wird. Ich habe selbst keine Lust mehr dazu, aber die Behebung dieses Mangels dauert keine zehn Minuten. Gegf. reden wir noch darüber.

Am Gehäuse ist ein CPU-Resetknopf angebracht. Im Gegensatz zu dem an der Rückseite, der nur einen NMI ausgibt, funktioniert dieser immer, denn er versetzt den Z80 in den Einschaltzustand. Die Speicherinhalte bleiben allerdings erhalten, so daß man nach einem Hänger einfach weiterarbeiten kann. Nur Zeiger auf BASIC-Texte, Variable usw. werden in ihren Nullzustand versetzt, denn das System bootet natürlich neu.

Der Cassettenrecorder ist ausgebaut (wird mitgeliefert; es sind weiterhin zwei Recorder anschließbar). An seiner Stelle ist ein Ziffernblock wie beim Genie II. Zusätzliche Tasten, die ihn zu einem Hexblock machen, sind bereits fertig zum Einbau vorbereitet und werden beigelegt. Dazu hatte ich selbst allerdings bisher keine Lust.

Das alte einfache Netzteil ist durch ein wesentlich stärkeres Schaltnetzteil ersetzt, so daß für eine Fülle weiterer Um- und Anbauten genügend Power zur Verfügung steht. Wie sich in der Praxis zeigte, nimmt es auch Kurzschlüsse nicht übel.

Die Floppies, die ich ebenfalls verkaufen möchte, sind auch modifiziert. Sie enthalten zwei Laufwerke mit 80/DS/DD und eins mit 40/SS/DD. Durch einen Umschalter kann wahlweise auf Laufwerk 0 (80) oder 2 (40) gebootet werden, so daß man praktisch jedes fremde DOS fahren kann. Evtl. wäre ich interessiert, eines der Laufwerke selbst zu behalten. Darüber wäre zu verhandeln.

Der Monitor ist bernsteinfarben.

Außer Helmuts Superbanker gehört zum Gerät auch noch der EG 64 MBA von TCS. Daran habe ich eine Platine angebaut, die alle notwendigen Anschlüsse zur Port-I/O hat. Es lassen sich damit gleichzeitig ca. 20 externe Geräte ansteuern, die über Ports betrieben werden.

Eine ECS-Buskarte und ein serielles Interface sind auch dabei. Beide sind aber noch nicht an den Computer angeschlossen. Da sich die Bastelei in der letzten Zeit sehr gehäuft hatte, waren immer andere Dinge wichtiger. Der Einbau ist nicht schwierig. Gegf. kann ich Hinweise dazu geben.

Schließlich ist noch ein Zweifach-Busextender dabei. So lassen sich außer den Floppies noch weitere Geräte auf den Bus stecken. Da die Steckleisten einen recht weiten Abstand haben, passen gut und gerne noch zwei weitere dazwischen, so daß ein Vierfach-Extender entsteht. Die Leisten stecken in Slots und können herausgenommen werden. Man kann deswegen Zusatzgeräte wahlweise mit Vater- oder Mutteranschluß betreiben. Letzteres ist bei den gängigen Geräten für Genie-Computer üblich.

Der Leser wird sich mit Recht fragen, weshalb ich ein solches Gerät nicht selbst behalte. Weil das Bessere der Feind des Guten ist. Inzwischen habe ich nämlich ein Genie III s, bei dem all' das und noch einiges mehr von Hause aus schon drin ist.

Als Verhandlungsbasis für den Preis des kompletten Systems stelle ich mir DM 2.500,- vor. Allein die Floppies waren wesentlich teurer. Es dürfte sich rund um ein Drittel der Summe der Neupreise aller Einzelteile handeln. Insofern denke ich, daß dieses Angebot fair genannt werden kann. Wenn Interessenten nur einzelne Komponenten des Systems haben möchten, wird man sich über den Preis jeweils unterhalten müssen.

Umsonst gibt's dazu noch H-DOS, eine erweiterte Version von G-DOS. Seine Features möchte ich hier nicht beschreiben, denn das wären ein paar weitere Seiten. H-DOS holt aus den Hardware-Erweiterungen jeweils das Äußerste heraus und ist damit jedem anderen DOS, das für die Computer unseres Clubs im Umlauf ist, überlegen.

Arnulf Sopp

Berliner Putzfrau zum Computer: »Ick hör imma, Du kannst allet – staub Dir doch selba ab!«

»Frollein Rosi, laut Computer schreiben Sie nur ganze 1,348 Briefe an einem Tag, machen pro Zeile 0,0975 Fehler und verbringen 2,416% Ihrer Arbeitszeit auf der Toilette! Was haben Sie dazu zu sagen?«
»Wenn ich aufrunden darf, Chef, Sie gehen mir 100% auf den Wecker!«

Z80 – Assemblerkurs zu verkaufen!!!

Die Z80-Maschinensprache kann man nur am praktischen Beispiel erlernen! Diese Tatsache ist mir das erstemal aufgegangen, als ich die Z80-Bibel (Programmieren des Z80 von Rodnay Zaks, erschienen im SYBEX-Verlag) durchzuackern versuchte. "Da ich jedoch einen TRS80 mit einer Z80-CPU habe, kann das ja eigentlich nicht allzu schwer sein", dachte ich und versuchte mit allen möglichen Mitteln das Geheimnis der Mnemonics zu lüften. Vergebens!!! Leider gibt es trotz der beinahe schon in Computergenerationen zählenden Standzeit des TRS80 keinen einigermaßen brauchbaren Kurs zum Erlernen der Maschinensprache auf dieser Maschine. Was tun also, um doch noch zu einem einigermaßen gefestigten Grundwissen über die Maschinenspracheprogrammierung zu kommen?
Ein Kurs mußte her!
Nach einer gewissen Zeit der Umschau habe ich mich für den Z80-Kurs von Christiani entschieden. Grundlage der Entscheidung war dabei die Tatsache, daß ich nach dem Softwarekurs auch den anschließenden Peripherie-Lehrgang machen wollte.
Gesagt, getan! Beide Kurse sind beendet (jeweils ca. 4 Monate a' 20 Stunden) und ich glaube mich inzwischen nicht mehr zu den ganz feuchten Greenhorns unter den Assemblerfreaks zählen zu müssen. Natürlich reichen die erworbenen Kenntnisse nicht, um unserem Assemblerpapst Arnulf Sopp das Wasser zu reichen und müssen noch erheblich vertieft werden. Das jedoch kann ich ganz gut auf dem TRS80 machen!
Und so steht nun also ein Z80-Kurs (Microprofessor) und ein Peripheriebausteine-Lehrgang herum und verstaubt. Diesen Zustand möchte ich ändern! Ich möchte die ganze "Lernanlage" verkaufen und von dem Erlös einen Olivetti M10 oder ein Model 100 von Tandy anschaffen (logisch muß ich noch was drauflegen!). Es steht also folgendes zum Verkauf:

1 Z80-Kurs (4 Lehrbriefe im Ordner und ein Microprofessor Lehrsystem) zum Original-Preis von 676,-.
1 Pheripherie-Bausteine-Kurs (4 Lehrbriefe im Ordner mit einer Erweiterungsplatine mit CTC, PIO und 8255) zum Original-Preis von 642,-.

1318,-

Das Ganze soll für ca. **900,– DM (VB)** über den Tisch gehen, wobei ich auch an einem Tausch gegen einen M10 oder ein Model 100 interessiert bin (natürlich mit entsprechender Ausgleichszahlung)!
Interessenten melden sich am besten telefonisch bei mir.
Meine Adresse und Telefon-Nr.: Hartmut Obermann
Schwalbacher Str. 6
6209 Heidenrod/Kemel
☎ 06124 / 3913

Hardware im letzten Augenblick!!!

Am ersten Wochenende (Sa./So. 01/02) im März hatte ich Besuch von unserem Clubinfo-Macher Jens Neueder. Samstags war auch noch Ulrich Böckling anwesend, von dem ihr in diesem Info einen Hardwarebeitrag findet!
Während des Treffens kam unter anderem zur Sprache, daß man von dem laufenden ECB-Projekt schon lange nichts mehr gehört hat. Es gibt schon Gerüchte, die Sache sei ganz gestorben! Dem, das sei hier noch einmal ausdrücklich bemerkt, ist nicht so!

Nachdem mir der Initiator des Euro-Card-Bus-Projekts, Walter Zwickel, im Sommer die Layouts für die zu fertigende Platine geschickt hatte, wurden diese von Ulrich Böckling auf Folien umkopiert. Die Herstellung der Platine hat dann einer unserer eifrigsten Hardwarebastler, Manfred Held übernommen.
Und obwohl alle an der Entwicklung und Fertigung der Adapterplatine Beteiligten immer ihr Bestes gaben, waren Verzögerungen nicht ganz zu vermeiden. Schließlich gibt es für einige Mitglieder hie und da auch etwas wichtigeres zu tun, als immer nur am Computer zu hängen! Nichts desto trotz ist die Platine nun endlich (zumindest in zwei Prototypen) fertiggestellt.

Tester gesucht!!!
Nun muß noch die Prüfung auf ev. Fehler im Layout oder in der Schaltung erfolgen. Leider habe ich gerade erfahren, daß Walter Zwickel auf einen PC-Kompatiblen umgestiegen ist und uns damit der kompetenteste Mann zum Testen wegfällt. Sollte sich jemand als Tester zur Verfügung stellen, melde er sich bitte bei Manfred Held oder bei mir. Wenn alle Stricke reißen, werden Ulrich Böckling und ich das Auschecken übernehmen. Die Fertigstellung des Projekts wird dann aber noch länger auf sich warten lassen, da wir beide nicht allzuviel Zeit haben. Wenn alles glatt geht, können die ersten Karten nach dem Clubtreffen ausgeliefert werden. Die Platinen werden voraussichtlich ca. 10,- DM pro Stück kosten und auch für den technisch nicht allzu versierten (einen Lötkolben sollte man schon bedienen können) zusammenbaubar sein. Die Bauteile können ev. in Sammelbestellung beschafft werden, darüber kann man jedoch auch auf dem Clubtreffen noch entscheiden. Sollte jemand es sehr eilig haben, kann er die Karte auch sofort (vor den letzten Tests) bei Manfred Held anfordern!

Manfred Held, der mich telefonisch bat diesen Bericht über den Entwicklungsstand in Sachen ECB-Bus zu schreiben, gab mir zusätzlich den Auftrag, folgende Hinweise bzw. Bitten zu formulieren:
1. Die von ihm vorgestellte Highspeed - Umschaltung kollidiert mit manchen CP/M-Erweiterungen! Aus diesem Grunde sollten CP/M-Benutzer die Adresse der Schaltung von FEh in FBh ändern. Wie man das macht ist im Hardwarebeitrag von Ulrich Böckling kurz beschrieben (es ist nur eine Adressleitung zu ändern)!
2. Um bei der weiteren Entwicklung von Karten für den ECB-Bus Kollisionen mit schon bestehenden Hardwareerweiterungen zu vermeiden, werden alle Mitglieder des Clubs aufgerufen, ihnen bekannte Adressen von Erweiterungen im Info zu veröffentlichen bzw. an
   - Manfred Held, Stirner Str. 22, 8835 Pleinfeld

   zu schicken! Z.B. hat Manfred eine Hardwareuhr für den ECB-Bus gebaut, welche die Adressen C8h-CBh belegt.

Ich hoffe, möglichst viele Mitglieder kommen dem Aufruf nach und verabschiede mich bis zum Clubtreffen mit den freundlichsten Grüßen,
Euer

Hartmut Obermann

??? FRAGEKASTEN ???

Für den Fragekasten:

Wer hat ein kommentiertes Listing des Bildschirmeditors (SEDIT etc.) oder weiß, wie das Ding funktioniert. Jedes Mal, wenn ich mir das disassemblierte Listing anschaue, finde ich heraus, daß das Programm überhaupt nicht laufen dürfte, obwohl es das zweifellos tut, nur fatalerweise zusammen mit der 80-Zeichen-Karte nicht!

Wer kann mir bei "Lucifer's Realm" im Ventilationsschacht weiterhelfen. Habe leider nichts dabei, um den CANVAS zu CUTten.

Wer hat Informationen, Programme oder das Handbuch für die 80-Zeichen-Karte?

Wer hat eine Assembler-Routine, die die Tastaturroutine ersetzen kann, so daß Umkodierungen vorgenommen werden könnten (Umlaute realisieren oder ESCAPE mit Clear-Taste, denn <Shift Pfeil Abwärts> ist nun wirklich grausam)?

Gerald Schröder

TECRIPS scheint nicht alle Codes an den Drucker weiterzugeben, die zwischen &-Zeichen eingeschlossen sind. Die Codefolge $&2401& (1Bh, 24h, 01h bzw. 27, 36, 1)), mit der die freidefinierbaren Zeichen des Gemini-10X zugeschaltet werden sollen, wird nicht befolgt. Wer hat damit Erfahrungen und kann mir helfen?

Arnulf Sopp

Für den Fragekasten:
- Wie steht es mit dem Projekt 'ECB-Bus-System' ? Lange nichts mehr davon gehört, ist es eingeschlafen oder kurz vor der Vollendung ?

- Seit ein paar Tagen sind wir DFÜ-fähig. Akustikkoppler, V.24 und Software (Multicom) alles von RB-Elektronik. Wer hat Erfahrungen mit 'Multicom' und Mailboxen und außerdem Lust uns zu schreiben, entweder per Post oder über Corner Box (02927/650) an 'JABO'.

Rainer Jablotschkin

Ehe der große Stop kam

Das Volk der Bit im Lande Byte,
man schrieb 01 der Nanozeit,
erwählte sich aus eigenen Kreisen
einen König, welcher Flop geheißen.

Flop war ein schneller Interrupter,
er zog Register, hielt Adapter,
und an der Spitze der Regierung
gewann er manche Assemblierung.

Flop emulierte fulminant
und fing den Cobol mit der Hand.
Einst sprach der 1. Programmierer
der Bitpartei zu Flop, dem Führer:
Es fehlt dir in dualem Sinn
zum guten Glück die Königin !

Des ernsten Rates Ironie
bemerkte nur das Parity;
die andern wählten klar und klipp
dem Flop ein Weib, die flotte Flip.

Flip-Flop, ein Hochzeitspaar in Pracht,
man schrieb die Nanozeit 08.
Das Bitvolk jauchzte - auf zum Run,
zum Doppeljob als Festprogramm.

Das war ein Flippen und ein Floppen,
ein Zyklus, nicht mehr abzustoppen,
denn kybernetisch digital
wie weiland hexadazumal
klang die Musik der Elektronen
schon hoch bis in die 11er-Zonen.

Am tollsten trieben es die Timer,
der Multiplexer ging in' Eimer.
Mit Dezimalen fuhr man Schlitten,
das Gleitkomma war ausgeglitten.

Im großen Overhead-Büro
kniff man Compilern in den Po,
und Bits addierten splitternackt
im Puffer zum Dreivierteltakt.

Ein Interface sang ordinär
die Watson-Hymne rein binär.
Man hatte Maß und Ziel verloren...
dem Supervisor kams zu Ohren:
Du lieber Dump, welch ein Gezeter -
Greif ein, oh großer Operator !

Ein Druck - es kam der große Stop.
"Halt" war das letzte Wort von Flop.
Abrupt erstarb der Bit-Gesang,
na, seht Ihr wohl ? Cherchez la femme !

Im Lande Byte seitdem geht stumm
bei Tag und Nacht der Error um !

Auch wenn Computer-Freaks Rechnerprogramme ohne kommerzielles Interesse nur tauschen möchten, bekommen sie es schnell mit dem Staatsanwalt zu tun. Oder es melden sich Rechtsanwälte, die durch »Abmahnschreiben« schnelles Geld einzustreichen gedenken.

Abmahnfalle

Auch Computerprogramme unterliegen dem Urheberrechtsschutz.

An die »ges. Vertretung von Rolf Koch«*) war der lange Brief adressiert, in dem der Vater Holger Koch mit rund 1100 Mark für »Gebühren und Auslagen« zur Kasse gebeten wurde. Absender war eine Rechtsanwaltskanzlei im fernen München, die in dem offensichtlich vervielfältigten Schreiben Sohn Rolf »Software-Piraterie« anlastete und nun vom Vater per Abmahnung die Versicherung erheischte, daß der Sprößling Kopien seiner Programme für den Heimcomputer niemals wieder zum Tausch oder Verkauf anbiete. Widrigenfalls seien für jeden Verstoß gegen § 97 des Urheberrechtsgesetzes 2000 Mark Vertragsstrafe fällig. Der vorläufige »Gegenstandswert« betrage 50 000 Mark. Als so hoch sei der Schaden einzuschätzen, den »Pirat« Rolf dem Hersteller seiner Computerspiel-Programme . mit Raubkopien zufügen könne. Und nach dieser Schadenshöhe berechnet sich die Abmahngebühr von 1100 Mark. Die Kochs hielten leicht verstört Familienrat, vermochten sich jedoch keines Unrechts bewußt zu werden: Rolf besitzt einige Computerspiele für seinen Rechner, tauscht die relativ teure Software mit Freunden, hatte aber auch auf eine Annonce hin seine Programme einem anderen Freak angeboten. Eigene Kopien hatte Rolf nicht angefertigt. Und Geld war bei Rolfs Tauschaktionen zu keinem Zeitpunkt im Spiel gewesen.

Allerdings: Die zuständige Staatsanwaltschaft hatte einige Zeit zuvor gegen Rolf ermittelt, ob er Raubkopien von Computerprogrammen rechtswidrig unter die Leute bringe. Und spätestens hier beginnt der Fall Rolf allgemeine Bedeutung zu erhalten. Der Verdacht war nämlich aufgekommen, nachdem Rolf auf die Annonce reagiert hatte; denn beim Inserenten hatte die Polizei wegen des dringenden Verdachts der Software-Piraterie Hausdurchsuchung gehalten – und war fündig geworden. Was die Beamten auch fanden, war eine große Menge an Adressen weiterer Computerbesitzer, darunter die von Rolf.

Mitgefangen, mitgehangen: Nach der Ergänzung des Urheberrechtsgesetzes im Mai des Jahres 1985 haben Ermittlungsbehörden nicht mehr nur auf Antrag eines Geschädigten tätig zu werden, sondern auch bei vermutetem öffentlichen Interesse an Strafverfolgung. So geriet auch Rolf in die Ermittlungsmühle. Für ihn verlief die Geschichte allerdings unproblematisch: Der Staatsanwalt stellte das Verfahren ein.

So klar die Situation danach war, so vertrackt ist die Rechtslage nach dem Abmahnschreiben aus München: Vater Koch hatte sich nach der ersten Rechtsanwaltspost vehement geweigert, die Unterwerfungserklärung zu unterschreiben und damit ein rechtswidriges Verhalten seines Sohnes anzuerkennen: folgerichtig hielt er auch die 1100 Mark Gebühren für gegenstandslos. Er hat vielmehr den dringenden Verdacht, daß die Schuldfrage die Rechtsanwälte gar nicht sonderlich interessiert: Vielmehr sieht er – je länger die Sache sich hinzieht, desto deutlicher – die Forderung aus München als »Abmahnmasche«, mit der sich die Kanzlei an alle Adressaten wendet, die sie über Ermittlungsverfahren ausfindig machen kann.

Vater Kochs entschlossene Weigerung zog weitere Rechtsanwaltspost nach sich: Werden die Gebühren bis zu einer letzten »Nachfrist« nicht gezahlt, soll Vater Koch auf dem Wege einer Unterlassungsklage zur Zahlung bewegt werden. Nach nochmaligem Gespräch mit dem Staatsanwalt will Holger Koch nun erst recht standhaft bleiben. Eine Zuwiderhandlung hätte nur erwiesen werden können, wenn der Staatsanwalt eine Hausdurchsuchung angeordnet hätte. Diese Maßnahme mochte der anscheinend weise Staatsanwalt jedoch nicht gegen bisher unbescholtene Bürger wegen des geringfügigen Verdachts einer eventuellen unerlaubten Weitergabe von Raubkopien einleiten, zumal selbst bei begründetem Verdacht eine Wiederholung auszuschließen sei. Holger Koch appellierte denn auch weniger juristisch, vielmehr pädagogisch an die Kanzlei, den Versuch zu unterlassen, mit dem Mittel der Abmahnung Jugendliche zu kriminalisieren. Rolf habe sich nicht rechtswidrig verhalten. Basta.

Bei Redaktionsschluß war noch offen, ob der Appell an die Vernunft die erwünschte Wirkung gehabt hat.

*

***test rät:** Besonders Eltern, die ihren Kindern einen Heimcomputer unter den Tannenbaum gestellt haben, sollten vorbeugend aufklären: Gegen den Tausch von Computer-Programmen im Freundeskreis ist so lange nichts einzuwenden, wie die Gewähr besteht, daß keine Kopien hergestellt werden. Bei der Weitergabe an Fremde, die dann ohne Wissen des Eigentümers Raubkopien ziehen, macht sich der Ausleihende mit strafbar. Die Frage, ob Computer-Programme überhaupt als urheberrechtlich schützenswerte Werke anzusehen sind, hat der Bundesgerichtshof in einem Urteil vom 9. Mai 1985 grundsätzlich bejaht (Az.: I ZR 52/83). Bei rechtswidriger Weitergabe von Programmen haben Freaks also mit Staatsanwalt, Klagen, Strafen und eben auch mit Abmahnschreiben zu rechnen.*

Aus aktuellem Anlaß bekamen wir einen Brief von Herrn Adelmann durch einen befreundeten CLUB zugesandt.

Wir möchten Euch diesen Brief ausschnittweise darlegen. Gleichzeitig solltet Ihr Euch über dieses Thema einmal Gedanken machen. Eine Diskussionsrunde über diese Thematik wäre bei unserem kommenden Clubtreffen sicher sehr interessant.

// Kopieren ist nur in ganz wenigen Fällen verboten. Hätten wir in der Rechtspflege seriöse Zustände, würde die Behörde, bevor sie durchsucht, erst folgende Fragen prüfen :

Liegt ein Strafantrag vor ?

Diese Anträge fehlen meist überhaupt. Die Behörden durchsuchen ohne sie. Finden sie was, fragen sie bei allen möglichen Firmen an, ob eine einen Strafantrag stellen will. So kommen dann etwa 2 Strafanträge zusammen, aber 100 Programme sind erst mal beschlagnahmt worden.

Und genau hier läge der springende Punkt einer seriösen Rechtspflege; denn die Strafantragssteller haben oft gar kein Strafantragsrecht. Wenn sie ein Vertriebsrecht haben, können sie selbst nicht Strafanträge stellen. Richtig wäre es, wenn vor der Polizeirazzia geprüft würde, ob es überhaupt einen interessierten Urheberrechtsinhaber gibt.

Ist ein echtes geistiges Eigentum gestohlen worden ?

Selbst wenn es einen Erfinder des kopierten Spiels gibt, ist noch lange kein Urheberrecht verletzt. Die meisten Spiele haben gar keinen Schutzwert. Es ist gar kein "geistiges Eigentum". Es sind meist die selben Abläufe : Abschießen irgendwelcher beweglicher Ziele. Ob es eine Seeschlacht, ein Luftangriff oder eine Panzerschlacht ist, all das Zeug hat keinen Schutzwert. Wären die Strafverfolgungsbehörden seriös, würden sie vom Strafantragsteller ein Gutachten über den Schutzwert des Programms verlangen, bevor sie auf den Grundrechten anderer herumtrampeln.

Tatsächlich scheitern die meisten Strafverfahren dann beim Staatsanwalt, der alt aussieht, wenn er die Urheberrechtsqualität der Programme beweisen soll. Die Verfahren werden alle wieder eingestellt, meist ohne Buße und Folgen. Nur die ganzen Programme waren monatelang als Asservate unter Verschluß. //

//— Das bedeutet, daß sich der User vor drohendem Unrecht durch die Rechtspflege schützen muß. Die Gerichte schaden mehr als sie nützen. Man verfährt so

Getauscht werden keine Kopien. Wer ein Programm hat, das ein anderer auch sehen soll, schickt das Orginal. Wenn der Partner eine Kopie fertigt, ist dies seine eigene Sache

Im Hause behält man keine Orginale, nur Arbeitskopien. Beschlagnahmt die Polizei eine Kopie, ist hieraus nicht zu beweisen, ob es eine Arbeitskopie oder eine "geraubte" ist.

Hierzu gehört auch, sich total passiv gegenüber den Verfolgern zu verhalten. Man macht keine Angaben zur Sache. Wer eine Version vorträgt, schließt selbst 19 andere denkbare Versionen aus. Er torpediert seinen eigenen Schutz aus dem "im Zweifel für den Angeklagten". Man macht nur Angaben zur Person, keine zum Einkommen oder sonst was. Zu den Angaben zur Person gehören nur Name, Vorname, Alter, Wohnung und Beruf. Welcher Narr wollte behaupten, über ein geringes Einkommen zu verfügen, und dann sieht man Spiele im theoretischen Wert von 10.000 bei ihm.

Die Vertriebsfirmen wissen, daß das Urhebergesetz sie nicht schützt. Deswegen lassen sie heute Verträge unterschreiben, daß das erworbene Programm nur für eine bestimmte Anlage genutzt werden darf. Diese Verträge liegen den Kasetten oft bei. Nur kann kein Verkäufer solche Verträge durchsetzen. Die Feinde der Kopierer sitzen längst am kürzeren Hebel. Verurteilt werden nur die Schwätzer, die selbst glauben Unrecht getan zu haben.

Wer durchsucht wird, hat die Ehrenpflicht, seine Partner zu informieren. Die Hausdurchsuchung kommt dann wie die Tollwut nach einer Art Incubationszeit von bis zu 9 Monaten. Man hat also Zeit, alles zu ordnen. Man kann Spuren verwischen, indem man die Anschriften von Polizeibeamten unter die Listen der Tauschpartner aufnimmt, Anschriften mit Paßworten deckt und einen Ordner voller Rechnungen von Second-hand-shops hat, aus denen sich der Erwerb von Computerhardware und kompletten Spielesammlungen ohne Titel ergibt. //

CLUBBÜCHEREI

Nr. 0001: BASIC-Brevier: Eine Einf. in d. Prog. v. Heimcomp.

Siegmar Wittig --- Heinz Heise GmbH

Ein BASIC-Kurs fuer Nicht-Mathematiker und echte Amateure. Neben Grund- und Aufbaukurs gibt es zahlreiche Beispiele (eine gute Sammlung!).

Nr. 0002: Computerwissen

Michael Scharfenberger --- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von Hard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knobeleien programm. in BASIC

Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

\- --- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel --- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80

Chris Palmer --- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment.

Nr. 0008: More Basic Computer Games

David H. Ahl --- Creative Computing Press, USA

84 Spiele fuer den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist fuer Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen

Busch Rudolf --- Franzis-Verlag GmbH, M nchen

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0011: TRS-80 PROGRAMS

Tom Rugg und Phil Feldman --- Dilithium Press, Beaverton,USA

32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012: Programme und Tricks fuer Genie I und Genie II

Clemens Becher, Franz Seger --- ?

Viele Programme, Tips und Tricks fuer den Genie.

Nr. 0013: BASIC: Alles ueber PEEK und POKE

Heiko Requardt --- Franzis-Verlag

Eine Software-Sammlung in BASIC (mit vielen guten Tips und Tricks fuer den 'Amateur').

Nr. 0014: TRS-80 und Video Genie ROM-Listing fuer Level II

Luidger Roeckrath --- ?

ROM-Listing, RAM-Adressen, I/O-Adressen, Unterprogramme, Basic-Anweisungen und Funktionen, Aufzeichnungsformate auf Cassette, ...

Nr. 0015: Machine Language Disk I/O & Other Mysteries

Mihael J. Wagner --- IJG Inc., 1953 West, USA

Alles rund um Disk und DOS wird erkl rt - mit zahlreichen wertvollen Hinweisen und Tips.

Nr. 0016: Das Modem-Sonderheft (Daten bertr. mit Mikrocomp.)

verschiedene --- MC, Franzis-Verlag, M nchen

Grundlagen, Schaltungstechnik, DF -Programme

Nr. 0017: GENIE DATA (Heft 1/83 bis Heft 3/85)

--- Heinz Hueben Verlag

Eine Sammlung von 13 Computerheften.

Nr. 0018: Basic Disk I/O Faster And Better & Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips und Tricks zum DOS, zur DISK, zu Files etc. zeigt (mit Programm-Diskette zum Buch).

Nr. 0019: Basic Faster And Better & Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips, Tricks und Routinen in BASIC bietet.

Nr. 0020: The Custom TRS-80 & Other Mysteries

Dennis Bathory Kitsz --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das sich mit Soft- und Hardware f r uns eren Computer besch ftigt.

Nr. 0021: Microsoft Basic Decoded & Other Mysteries

James Farvour --- IJG Inc., 1953 West, USA

Ein kommentiertes Level II - BASIC in englischer Sprache.

Nr. 0022: TRSDOS 2.3 Decoded & Other Mysteries

James Farvour --- IJG Inc., 1953 West, USA

Ein kommentiertes TRSDOS 2.3 in englischer Sprache.

Nr. 0023: TRS-80 Disk & andere Geheimnisse

H.C. Pennington (G. Daubach) --- IJG Inc., 1953 West, USA

Das ins Deutsche bersetzte Buch beschreibt die Diskettenorg., die Zuordn ung freier Speicherbereiche, die Behebung v. Diskettenfehlern etc.

Nr. 0024: TRS-80 Data File Programming (Model I/III)

Finkel & Brown ---

Ein Lehrbuch, wie man in BASIC Datenfiles programiert und benutzt. Mit se hr vielen Beispielen (englisch-sprachig).

Nr. 0025: Programming Techniques for Level II BASIC

William Barden ---

Ein gutes Buch ber schnelle Graphiken, Suchroutinen und Sortierroutinen im Level II BASIC (englisch-sprachig).

Nr. 0026: Advanced Level II BASIC

Don Imman, Albrecht, Zamora ---

Ein gutes Lehrbuch mit vielen Beispielen zu allen Themen der Computerei (englisch-sprachig).

Nr. 0027: Level II BASIC

Don Imman, Albrecht, Zamora ---

Ein gutes Lehrbuch f r den Anf nger in BASIC (englisch-sprachig).

Nr. 0028: Programming In Style

Th. Dwyer, Margot Critchfield ---

Einf hrung in BASIC, strukturiertes Programmieren, Tips zur besseren Aufm achung der eigenen Programme, Tips f r den Profi (englisch-sprachig)

Nr. 0029: The First Book Of 80-US

verschiedene --- Hofacker Verlag, Muenchen

Sammelband aus dem TRS-80 Users Journal.

Nr. 0030: The Rest Of 80

verschiedene --- Wayne Green Books, USA

Viele Beitr ge aus der 80-Micro (Allgemeines, Graphik, Utilities).

Nr. 0031: Z-80-Applikationsbuch

Michael Klein --- Franzis Verlag, M nchen

Eine Einfuehrung in die Programmier- und Interfacetechniken des Mikroproz essors Z 80.

Nr. 0032: Z 80 Assembler Handbuch

--- Hofacker, M nchen

Die Z 80 Befehle mit Kommentar und je einem Beispiel.

Nr. 0033:   Programmierung des Z 80

Rodnay Zaks (Bernd Ploss)   --- Sybex-Verlag, Duesseldorf

Als Lehr- und Lernmittel konzipiert ist dieses ins Deutsche bersetzte Bu ch ein umfassendes Nachschlagewerk und eine gruendliche Einfuehrung.

Nr. 0034:   Daten - Disketten - Dateien

Hans-Joachim Sacht   --- CHIP, Vogel-Verlag, Wuerzburg

Anregungen und Hinweise, wie Programme mit komfortabler Dateiverw. geschr ieben werden koennen und was bei den versch. DOS zu beachten ist.

Nr. 0035:   Anwenderprogramme fuer TRS-80 und Video Genie

M. Stuebs   --- Hofacker, Muenchen

Viele (meist einfache) BASIC-Programme fuer den TRS-80.

Nr. 0036:   Programmieren mit TRS-80

M. Stuebs   --- Hofacker, Muenchen

Eine Einfuehrung in den TRS-80 und viele (meist einfache) BASIC-Programme .

Nr. 0037:   Die grosse BASIC Referenztabelle der 51 Dialekte

Wolf-Detlev Luther   --- Luther-Verlag, Sprendlingen

Nr. 0038:   Textsystem in BASIC und Z80-Assembler

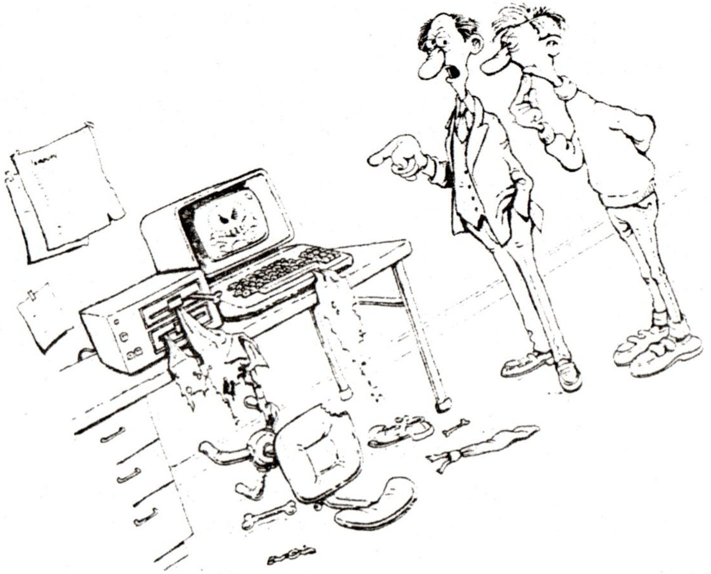
Gert Scheller   --- Luther-Verlag, Sprendlingen

Das Buch beschreibt ein fertiges Textsystem fuer den TRS-80, das in BASIC und Z80 modular aufgebaut ist, und leicht erweiterbar ist.

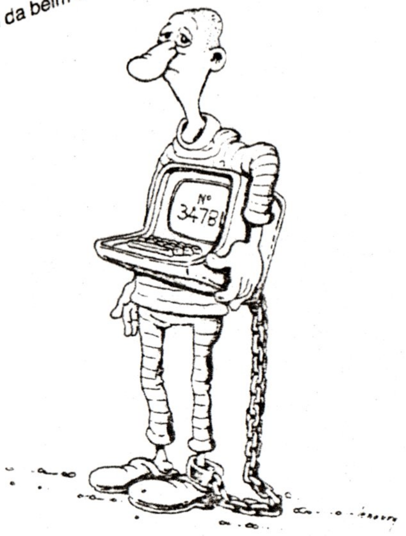
Nr. 0039:   Textsystem in BASIC und Z80-Assembler - Band 2

Wolfgang Weber   --- Luther-Verlag, Sprendlingen

Der Band 2 beschreibt Erweiterungen zum Programm aus Band 1.



Versuche ja nicht, dem da beim Schlüsselwort was vorzumachen . . .



Vorstand

Kontaktadresse für Clubangelegenheiten, Clubbücherei /Fundgrube, Clubkasse

Günther WAGNER
Gartenstraße 4
8201 Neubeuern
Tel.: 08035 /3361
< 18 - 20 Uhr >

Programmbibliothek

Kontaktadresse

Klaus-Jürgen Mühlenbein
Am Mönchgarten 28
6940 Weinheim
Tel.: 06201 /55052

Redaktion

Kontaktadresse

Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
Tel.: 0791 /42877
tagsüber 0791 /44-667

Adventure-Ecke

Kontaktadresse

Alexander WAGNER
Theresienstr. 21c
8224 Chieming
Tel.: 08664 /1500

Hardware

Kontaktadresse

Walter ZWICKEL
Lengfelden 123
5101 Bergheim (Austria)
Tel.: 0043662 /51130

Redakteure

dieser Ausgabe

Ulrich Böckling * Gerald Dreyer
Klaus-J. Mühlenbein * Jens Neueder
Hartmut Obermann * Gerald Schröder
Arnulf Sopp * Günther Wagner

sowie Artikel aus: Computer Persönlich, 80 Micro, Test und MC

Bankverbindung des CLUB 80

Sparkasse Rosenheim, BLZ 711 500 00
auf Konto-Nr. 194 712
Postscheckkonto der Sparkasse
Nr. 8077-801

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

am Ende unseres 12. INFO darf ich die Gelegenheit wieder nutzen, ein paar abschließende Worte zu finden:

Zuerst möchte ich nochmals auf die Programmkurzanleitungen zu sprechen kommen. Es hat sich ja aus Eurer Sicht lange nichts dazu getan. Aber es dauert halt doch etwas, bis das Material gesichtet und sortiert ist. Weiterhin ist für überarbeitete oder andere Werke -als in INFO 9 vorgestellt- in der Kurzanleitungsbibliothek noch Platz. Ihr braucht sie mir nur zuzusenden.

Des weiteren findet im April (11.04.-13.04.86) unser 2. Clubtreffen statt. Ich hoffe, daß Ihr alle die Möglichkeit findet, daran teilzunehmen. Es ist doch meist sehr interessant, sich einmal persönlich kennen zu lernen. Mit Sicherheit hat auch der Eine oder Andere ein Diskussionsthema parat, über das man in versammelter Runde einmal sprechen sollte. Weiterhin geht es ja auch um die Wahl eines neuen Vorstandes und der Entlastung des vorherigen. Auch die "Computerei" wird an diesem Wochenende sicher nicht zu kurz kommen. Hier findet ...

.. der Anfänger geballtes Computerwissen, das er dann mit seinen gezielten Fragen durcheinanderbringen darf -

.. der alte Hase einen anderen ganz ausgefuchsten Kollegen, der zum gleichen Problem natürlich eine andere Lösung aus seiner Trickkiste zieht.

Ihr seht also, es lohnt sich sicher für jeden von Euch, an dem Clubtreffen teilzunehmen. Auf alle Fälle liegt jedem Clubinfo ein persönliches Kärtchen bei, das Euch als Teilnehmer des Clubtreffens ausweist und gleichzeitig als Namenskärtchen dienen soll.

Zum INFO-Abschluß dieses 12. INFO's wünsche ich Euch allen "Happy Hacking", viel Freude an der neuesten INFO und ein gelungenes Clubtreffen.

Bis zum nächsten INFO -
oder noch früher in Holzhausen a. d. Heide verbleibe ich

Euer

J. Neueder

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PlZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Albers	Herbert	Zum Duwelshopen 14	2117	Westedt	
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Boeckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Drowälder	Bernd	Buchentalsweg 8	4939	Steinheim	05233 /4320
Fröhlich	Burghard	Brandströstraße 11	4350	Recklinghausen	02361 /63416
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hill	Peter	Eckstraße 36	6750	Kaiserslautern 31	0631 /54782
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach	0911 /506051
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lü.	06201 /55052
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285
Wagner	Alexander	Theresienstr. 21c	8224	Chieming	08664 /1500
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

Neuzugänge :

Waldemar Misioch

Peter Hill

Kurt Müller

Stand: März 1986

Bitte überprüft Eure Anschrift und meldet eventuelle Adressenänderungen an die Redaktion.

CLUB 80

Clubinfo

der

TANDY -
GENIE -
und KOMTEK -
ANWENDER

13. AUSGABE

Kontaktadresse : CLUB 80 / Peter STEVENS / Postfach 56 / 4600 Dortmund 1

Tel.: 0231 / [illegible]

Inhaltsverzeichnis

Seite:

Clubinternes

Vorwort des 1.Vorsitzenden. . . . . .01 - 02
Protokoll der Hauptversammlung. . . .03 - 04
Neue Satzung. . . . . .05 - 07
Clubleben allgemein . . . . .08 - 10

Software

Nachtrag zu Autoboot 40/80. 11
Nachtrag Grafik-Sonderheft 12
Listen von Basic-Files . . . . 11 - 12
Programmierwettbewerb. 13
HRG 192*192. 14
Directorybearbeitung . . . . 15 - 16
Ein paar kleine Änderungen . . . . . 16 - 18
Tips für SCRIPS. . . . . 19 - 21
Sondertasten diverser Genies . . . . 23 - 24
HRG-Hardcopy G III s . . . . 25 - 31
Graphik-JKL G III s. . . . . 32 - 33
Graphik-CLS G III s. . . . . 35 - 37
Bascom-Kurz. 38

Hardware

ECB-Bus-TRS-80-Selbstbau. 39
LNT für Genie I / TRS-80 M I 40
Floppytricks . . . . 41 - 42
A Quick Draw . . . . 43 - 45
Verbindung mit der Außenwelt . . . . 47 - 50
Videodat 300 . . . . 51 - 60

Seite:

Börse

Wer hat was ???--Wer sucht was ??? 61 - 62

Sonstiges

Verschiedenes . . . . .63 - 64
Computer waren sein Programm. . . . .65 - 66
Neues von der Bundespost. . . . . . .65 - 66
Fehlstart vermeidbar. . . . . .67 - 68

Programmbibliothek

Einleitung. 69
Neue Programme. . . . . .70 - 74
Nachtrag zum Programmjahresheft . . . 73

Die letzten Seiten

Impressum 75
Schluß 76
Clubmitgliederadressen . . . . am INFO-Ende
Fragebogen . . . . am INFO-Ende

Liebe Mitglieder,

nun, wie all diejenigen schon wissen, die bei dem Clubtreffen anwesend waren, wurde ich (dort) zum 1. Vorsitzenden gewählt. Ich danke euch für das Vertrauen und werde mich bemühen, diese Aufgabe zu bewältigen, so gut ich's eben kann. Allerdings möchte ich betonen, daß ich auf all eure Hilfe angewiesen bin.

Meine erste Tat ist es, Günther in unser aller Namen zu danken für die viele Arbeit und Mühe, die er bisher in den Club gesteckt hat (und möge er weiterhin kräftig mitmischen!). Ohne ihn gäbe es den Club 80 nicht, und das wäre schade finde ich. So möchte ich die anderen alten (Hasen?) und neuen Vorstandsmitglieder begrüßen und hoffe auf eine gute zusammenarbeit.

Als ich überlegt habe, was ich nun in diesen Zeilen schreiben soll, habe ich versucht, mir ein Bild zu machen von dem, was der Club ist und von dem Weg, den er gegangen ist. Ich glaube, in den letzten beiden Jahren hat es eine Phase der Expansion und des Kennenlernens gegeben, die nun so langsam zu Ende geht. Der Club wird sicher weiterhin wachsen, aber vielleicht nicht in dem Maße wie früher. Möglicherweise tritt nun eine Konsolidierungs-Phase ein, die sich in dem Bestreben einzelner Mitglieder zeigt, Verknüpfungen und Beziehungen innerhalb der Mitgliedschaft aufzubauen - dies ist zu begrüßen!

Das, was uns als Club zusammenhält, ist der Computer - und zwar einer bestimmter; allerdings gibt es innerhalb dieses Gebietes mehrere Interessensrichtungen, z.B. Hardware, Software und Telecommunication. Hierzu sehe ich zwei Möglichkeiten, Kontakte zwischen den Mitgliedern zu verfestigen:

1. Regionale Treffen
2. Interessensgruppen

1. Es ist mir schon seit langer zeit ein Wünsch gewesen, dem einen oder anderen im Raum 4 oder 5, mal anzurufen, um ihm kennenzulernen - mir fehlte bisher die Zeit (Antrieb?) dazu. Es wäre wünschenswert, wenn vielleicht einer in jeder Region sich bereit erklären würde, als Kontaktperson aufzutreten, um ggf. ein Treffen am Wochenende zu organisieren (Aufruf!!!). Er braucht dazu

nicht mehr als ein Telefon, um Wünsche und Anregungen entgegenzunehmen und einen Termin festzulegen. Irgendein Wirtshaus wird sich schon finden, wo wir ein paar Stunden verbringen konnten - oder nicht? Ich glaube, im Stuttgarter Raum ist ein Anfang in dieser Richtung schon gemacht worden. Jeder, der Interesse hat, melde sich bei eine der Vorstandsmitglieder.

2. Durch sachbezogene Interessensgruppen kann vielês erreicht werden; als Beispiel hierfür ist die Hardware-Gruppe; aber warum nicht auch eine Software-Gruppe oder eine Modem-Gruppe? Wichtig hierfür ist natürlich das Wissen, daß jemand anderer Interesse für dasselbe Gebiet hat. Deshalb an andere Stelle in diesem Info ein Fragebogen.

Also, jetzt komme ich zum Schluß. Ich hoffe, daß sich der eine oder andere aufgerufen fühlt, etwas zum Clubleben beizutragen - auch wenn es sehr wenig zu sein scheint; denn nur die Mitglieder machen den Club - nicht umgekehrt!

es wünscht euch viel Spaß am Tippen,

Peter

Peter

Zeit : 12.April 1986
Ort : Hotel "Taunusblick" ,5429 Holzhausen
Anwesend : 23 Mitglieder, davon 2 erst anwesend nach der Wahl des Vorstandes
Versammlungsleiter : 1.Vorsitzender Günther Wagner; nach der Neuwahl: 1.Vorsitzender Peter Stevens
Protokollführer : Hans Raggan

Der 1. Vorsitzende eröffnete die Versammlung um 16.00 Uhr, begrüßte die Erschienenen und stellte fest, daß die Mitgliederversammlung ordnungsgemäß einberufen wurde. Die Tagesordnung war allen zugegangen und bekannt.

Tagesordnung:

1. Kassenbericht
2. Entlastung des Vorstandes
3. Aufgaben und Pflichten des Vorstandes
4. Neuwahl des Vorstandes
5. Beratung über Eintragung in das Vereinsregister
6. bessere Aktivierung der passiven Mitglieder
7. Bericht über Hardwareprojekt
8. Computer und Recht
9. Fragen und Wünsche

zu TOP 1.:
Die Kassenübersicht wurde erörtert, die Kasse geprüft und der Vorstand hinsichtlich der Kassenführung einstimmig entlastet.

zu TOP 2.:
Dem Vorstand wurde nach kurzer Diskussion Entlastung erteilt.

zu TOP 3.:
Günther Wagner informierte die Anwesenden über die Pflichten und Rechte des Vorstandes.

zu TOP 4.:
Wahl des 1.Vorsitzenden: Vorgeschlagen wurden U.Böckling und P.Stevens. U.Böckling stellte sich aus wichtigen persönlichen Gründen nicht zur Wahl. Mit 20 Ja-Stimmen und 1 Nein-Stimme wurde Peter Stevens zum neuen 1. Vorsitzenden gewählt.
Wahl des stellvertretenden Vorsitzenden:
Vorgeschlagen wurden H.Obermann, U.Böckling und T.Frinck.
H.Obermann erhielt 9 Stimmen, U.Böckling erhielt 1 Stimme, T.Frinck 7 Stimmen. Der Stimme enthalten haben sich 4 Mitglieder. Somit wurde H.Obermann zum neuen stellvertretenden Vorsitzenden gewählt.
Wahl des Clubinfo-Redakteurs:
Vorgeschlagen wurde J.Neueder. Er erhielt 20 Stimmen, 1 Stimmenthaltung. Er ist somit gewählt.
Wahl des Softwarebibliothekars:
Vorgeschlagen wurde K.-J. Mühlenbein. Er wurde einstimmig gewählt.
Wahl des Hardware-Koordinators:
Vorgeschlagen wurden E.Kuhn, B.Drowälder und M.Held.
E.Kuhn erhielt 7 Stimmen, B.Drowälder 1 Stimme, M. Held 6 Stimmen. 7 Mitglieder enthielten sich der Stimme. E.Kuhn wurde als Hardware-Koordinator gewählt.

zu TOP 5.:
Es wurde zunächst eine allgemeine Diskussion über die Frage der Eintragung des Clubs in das Vereinsregister geführt. Bis auf ein Mitglied waren alle für eine Eintragung.
Dann wurde die Vereinssatzung, die von J.Neueder überarbeitet worden war, Punkt für Punkt vorgelesen, diskutiert und gegebenfalls wurden Punkte geändert. über jeden Punkt der endgültigen Vereinssatzung wurde abgestimmt:
Einstimmig wurden die Punkte 1,2,3,4,6,8,10 angenommen,
beim Punkt 5 gab es 19 Ja-Stimmen, 1 Nein-Stimme und 3 Enthaltungen
beim Punkt 7 gab es 22 Ja-Stimmen und 1 Enthaltung
beim Punkt 9 gab es 1 Enthaltung, 1 Nein-Stimme und 21 Ja-Stimmen.
Es nahmen 23 Mitglieder an dieser Abstimmung teil.
Der Vorstand wurde einstimmig beauftragt, die Eintragung in das Vereinsregister beim Amtsgericht Schwäbisch Hall zu beantragen.

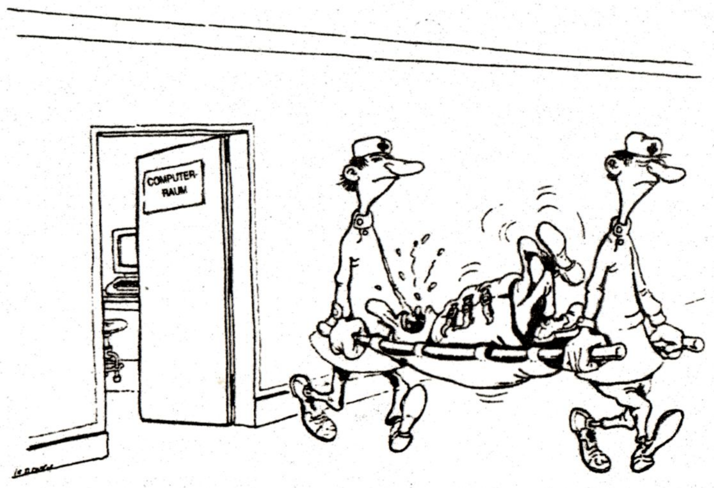
zu TOP 6.:
Fragen der Aktivierung der Vereinsmitglieder wurden erörtert.

zu TOP 7.:
Das ECE-BUS-Projekt wurde dem Hardwarekoordinator übergeben.

zu TOP 8.:
Entfiel wegen fortgeschrittener Zeit. Wird später nachgeholt.

zu TOP 9.:
Fragen und Wünsche der Mitglieder wurden erörtert. Es wurden kaum Wünsche geäußert .

Protokollführer 1.Vorsitzender



SATZUNG
des CLUB 80 Computerclubs

1. Name und Sitz
Der Club führt den Namen **CLUB 80** Computerclub.
Sein Sitz ist in Schwäbisch Hall.
Er soll in das Vereinsregister eingetragen werden.

2. Zweck und Aufgaben
Der Club betreibt die Förderung und Verbreitung der Computeranwendung.
Seine Aufgabe ist es, Erkenntnisse und Erfahrungen über Computer -speziell des TRS 80 und Kompatible- einem möglichst großen Kreis von Interessierten zur Verfügung zu stellen.
Der Austausch von Erfahrungen, urheberrechtlich nicht geschützter Software und Büchern geschieht durch Clubnachrichten, Versand und Clubtreffen.

3. Mitgliedschaft
Mitglied ist, wer auf einfachen Antrag gegenüber dem Vorstand aufgenommen wurde, die Clubsatzung anerkannt hat, die Aufnahmegebühr von DM 10,- und den jeweilig gültigen Jahresbeitrag entrichtet hat.
Die Mitgliedschaft erlischt durch Austritt, Streichung, Ausschluß oder Tod.
Bezahlte Beiträge können nicht zurückerstattet werden.

a. Der Austritt geschieht durch schriftliche Erklärung unter Angabe von Gründen gegenüber dem Vorstand. Kündigungsfristen sind nicht einzuhalten.
b. Beitragszahlungen, die trotz zweimaliger Mahnung nicht entrichtet wurden, berechtigen den Vorstand zur Streichung des Mitglieds.
c. Mitglieder, die dem Ansehen des Clubs schaden oder seinen satzungsgemäßen Aufgaben zuwiderhandeln, können ausgeschlossen werden.
   Der Ausschluß wird vom Vorstand, nach vorheriger einstimmiger Beschlußfassung in einer Vorstandsitzung, oder durch Beschluß der Mitglieder mit 2/3 Mehrheit in der Hauptversammlung ausgesprochen.
   Mitglieder, die ausgeschlossen wurden, können nicht wieder aufgenommen werden.

4. Mitgliedsbeitrag
Die Aufrechterhaltung des Clubs verursacht Kosten. Als Beispiele sind zu nennen: Programmarchivierung, Druck der Clubinfos und Portokosten.
Diese Ausgaben sind so gering wie möglich zu halten.
Der Mitgliedsbeitrag wird von der Hauptversammlung festgesetzt. Hierfür ist eine 2/3 Mehrheit der anwesenden Mitglieder erforderlich. Der Beitrag ist jährlich, zum Jahreswechsel, im voraus und in der beschlossenen Höhe zu entrichten. Des weiteren wird ein einmaliger Aufnahmebeitrag von DM 10,- erhoben. Erfolgt die Aufnahme in den Club nach dem 31. Juli, ist nur der halbe Jahresbeitrag für das erste Mitgliedsjahr zu zahlen.

5. Clubkasse / Rechnungsprüfung
Aus der Clubkasse werden die laufenden Unkosten bezahlt. Größere überschüsse werden gemeinnützig verwandt. In der Clubinfo erscheint zu gegebener Zeit eine Aufstellung der zu Artikel, deren Kauf beabsichtigt ist.
Ein Rechenschaftsbericht über die Clubkasse erfolgt jeweils zum Jahreswechsel in der Clubinfo. Die gewählten Rechnungsprüfer haben die Aufgabe, in der Hauptversammlung die Abrechnung des Clubs für das abgelaufene Jahr zu prüfen und der Hauptversammlung zu berichten. Die Rechnungsprüfer haben das Recht, jederzeit Einblick in die Kassenführung zu nehmen.

6. Organe
Die Organe des Clubs sind
1. der Vorstand
2. die Hauptversammlung
3. die Gesamtheit der Mitglieder

1. Der Vorstand wird von der Hauptversammlung jeweils für ein Jahr von der Gesamtheit der anwesenden Mitglieder in einfacher Mehrheit gewählt, nachdem der alte Vorstand entlastet worden ist. Bei Stimmengleichheit entscheidet die Stichwahl.
   Gewählt werden zu Vorstandsmitgliedern
   1. der Vorsitzende
   2. sein Stellvertreter
   3. der Hardwarekoordinator
   4. der Softwarebibliothekar
   5. der Clubinforedakteur

   Der Vorsitzende und der Stellvertreter vertreten den Verein nach außen. Sie sind Vorstand im Sinne des § 26 des BGB. Jeder ist alleinvertretungsberechtigt.
   Der Vorstand hat die Tagesordnung festzusetzen und die Hauptversammlung sowie außerordentliche Versammlungen einzuberufen.
   Der Vorstand entscheidet mit einfacher Stimmenmehrheit. Bei gleicher Stimmenzahl entscheidet die Stimme des Vorsitzenden.
   Sämtliche Ämter des Clubs werden ehrenamtlich wahrgenommen. Nachgewiesene notwendige Aufwendungen in Ausübung des Amtes werden erstattet.
   über Einzelausgaben größer DM 100,- entscheidet der Gesamtvorstand.

2. Die Hauptversammlung findet im 2. Quartal jeden Jahres statt.
   Sie ist vom Vorstand schriftlich, drei Monate vorher, in der Clubinfo einzuberufen. Außer dem Vorstand können 1/4 der Mitglieder schriftlich die Einberufung einer außerordentlichen Hauptversammlung beantragen.
   Die Hauptversammlung hat folgende Aufgaben
   1. Wahl des Schriftführers für die Versammlung
   2. Ernennung von 2 Rechnungsprüfern
   3. Anhörung des Tätigkeitsberichtes des Vorstandes über das abgelaufene Jahr und Entlastung des Vorstandes.
   4. die Vorstandsneuwahl
   5. Beschlußfassung über erforderliche Satzungsänderungen
   6. Ausschluß von Mitgliedern
   7. Festsetzung des Mitgliedsbeitrages

   Die Hauptversammlung entscheidet in den Fällen der Ziffern 1, 2 und 4 mit einfacher Mehrheit, in den übrigen Fällen mit 2/3 Mehrheit der anwesenden Mitglieder.
   über die in der Versammlung gefaßten Beschlüsse ist eine Niederschrift aufzunehmen. Die Niederschrift ist von dem Vorsitzenden der Versammlung und dem Schriftführer zu unterschreiben. Jedes Clubmitglied ist berechtigt, die Niederschrift einzusehen. Sie wird in der folgenden INFO veröffentlicht.

7. Änderung der Satzung
Jedes Mitglied kann eine Änderung der Clubsatzung unter Angabe der Gründe beantragen. Dieser Antrag wird in der Clubinfo veröffentlicht. über den Antrag wird mit einer 2/3 Mehrheit auf der nächsten Hauptversammlung entschieden.

8. Clubinternes

1. Programmtausch
Der Programmtausch erfolgt über die Programmbibliothek des Clubs.
a) Die Programmbibliothek des CLUB 80 besteht aus Programmen, die von den Mitgliedern entweder selbst erstellt, von freien Listings (aus Zeitschriften u.a.) abgetippt wurden oder aus frei erhältlicher Software stammen.
Jedes Mitglied verpflichtet sich, nur Programme einzusenden, die frei von Rechten Dritter sind. Aus diesem Grunde haftet stets der Einsender für die Einhaltung eines etwaigen Copyright's (Urheberrechts).
b) Nur Clubmitglieder können Programme aus der Programmbibliothek beziehen. Ein Verkauf von Programmen, die über den Club bezogen wurden, ist strengstens verboten, ebenso die Weitergabe an Dritte. Bei Zuwiderhandlung erfolgt sofortiger Ausschluß.
Strafrechtliche Verfolgung ist möglich.

2. Bücherverleih
Es wird eine Bibliothek angelegt. Jedes Clubmitglied kann jeweils ein Buch für vier Wochen kostenlos ausleihen. Ist das gewünschte Buch zur Zeit der Bestellung ausgeliehen, so wird das Mitglied auf eine Warteliste gesetzt. Das Buch wird auf Kosten des Entleihers versandt, der Besteller sendet das Buch auf seine Kosten zurück. Verlorengegangene Bücher müssen ersetzt werden. Die Weitergabe der Bücher an Dritte ist nicht erlaubt und führt zum Ausschluß des Mitgliedes.
Wird die Ausleihfrist überschritten, erfolgt eine Mahnung in Höhe von DM 5.- und für jede weitere Mahnung DM 5,-.
Die neuen Bücher werden in den Clubnachrichten veröffentlicht, eine komplette Liste erscheint jeweils zum Jahreswechsel.

3. Clubinfo
Der Club gibt eine Clubinfo heraus, die mehrmals im Jahr erscheint.

4. Hardware und Verbrauchsmaterialien
Bei entsprechendem Interesse werden Sammelbestellungen für Hardware-Artikel und Verbrauchsmaterialien organisiert. Außerdem sind die Mitglieder aufgefordert, eigene Hardware-Entwicklungen auch dem Club zur Verfügung zu stellen.

9. Auflösung des Clubs
Der Club wird auf Beschluß mit Dreiviertelmehrheit aufgelöst.
Sein eventuelles Vermögen wird gemeinnützigen Zwecken zugeführt.
Die Liquidation und Schlußzahlungen, sowie die Schlußabrechnung führt der geschäftsführende Vorstand durch.

10. Gültigkeit der Satzung
Die Satzung ist nach der Genehmigung per Abstimmung in der 2. Hauptversammlung des Clubs vom 12. April 1986 gültig und für alle Mitglieder verbindlich.
Die neue Satzung tritt mit ihrem Erscheinen in der 13 Ausgabe der Clubinfo in kraft.
Frühere Fassungen der Satzung sind danach ungültig.

MITGLIEDER AKTIVIERUNG

Ein Themenpunkt unserer Hauptversammlung war die "Aktivierung der passiven Mitglieder". Dieses Thema wurde aber wegen der fortgeschrittenen Stunde und dem nicht zu bremsenden Verlangen von Euch, wieder an die "Kiste" zu kommen, nur kurz abgehandelt.

Uns ist natürlich klar, daß man nicht jeden zur Mitarbeit anregen kann. Andererseits möchten wir aber auch gern wissen, was Euch mehr oder weniger interessiert und wohin der Trent geht. Wir, als Euer Vorstand, möchten ja so viele wie möglich von Euch mit unserer Arbeit ansprechen. Um uns die zu erleichtern findet Ihr am INFO-Ende einen Fragebogen, den Ihr bitte ausfüllen und zur Auswertung an die Redaktion zurückschicken solltet.

Weiterhin haben wir am Clubtreffen festgestellt, daß ein einmaliger persönlicher Kontakt -wie gerade nur zur Jahreshauptversammlung- nicht ausreicht. Unser Vorschlag ist es nun, daß sich Mitglieder, die räumlich in ca. 100 km Entfernung zueinander wohnen, sozusagen "regional" treffen. Einfach nur um Kontakt zu halten. Meist bringt ein solches Treffen dann aber auch noch Ideen für den Club und seine Mitglieder hervor, die sich sicher lohnen im INFO veröffentlicht zu werden. Zum Beispiel wird Ende Mai ein solches Treffen im Postleitzahlenbereich 71/72 stattfinden. Es gibt sicher eine Möglichkeit, dieses auch in anderen Ecken Deutschlands zu machen. Genauso sollte bei den Elektronik-Messen ein Tag ausgemacht werden, an dem sich die Mitglieder, die sowieso zur Messe kommen würden, treffen können. Bei der rechtzeitigen Publikation der Termine in unserer INFO lässt sich sicher auch hier etwas arrangieren.
Schaut mal in die Landkarte. Es ist sicher nicht weit bis zum nächsten Clubmitglied.

Die Redaktion

Autoren gesucht !!

Die Club-80-Redaktion sucht noch Autoren.
Folgende Themen wären interessant:

BASIC, Maschinensprache, Assembler
Programmiersprachen, CP/M, Adventure

Wer von Euch möchte dazu etwas Schreiben?
Wer hat vielleicht schon die entsprechenden Unterlagen bereit?
Bitte meldet Euch dazu. Sicher gibt es auch andere Themen als diese, die sich lohnen würden aufgegriffen zu werden.

Die Redaktion

IN EIGENER SACHE

Wie Ihr jetzt aus dem Protokoll ersehen habt, setzt sich der neue Vorstand aus folgenden Personen zusammen:

1. Vorsitzender Peter STEVENS
2. Vorsitzender Hartmut OBERMANN
3. Hardwarekoordinator Eckehard KUHN
4. Diskothekar Klaus-Jürgen MÜHLENBEIN
5. Redaktion Jens NEUEDER

Unsere vollständigen Adressen findet Ihr im Impressum oder in der Adressenliste des Clubs. Wir bitten Euch, daß Ihr Euch -Euren Wünschen entsprechend- an die jeweilig zuständigen Leute wendet. Ein Teil des Vorstandes ist Euch sicher bekannt. Die Einleitung unseres "Ersten" habt Ihr ja schon am Beginn des Heftes gelesen. Es steht als nur noch unser Hardwarekoordinator aus, dessen Vorstellung Ihr rechts auf dieser Seite findet.

Zu der hier auf den Seiten 5-7 vorgestellten Satzung sei noch folgendes angemerkt:
Diese Satzung ist diejenige, welche Ihr an der Jahreshauptversammlung in Holzhausen beschlossen habt.
Die Satzung ist ab Erscheinen in dieser INFO-Ausgabe für alle Mitglieder gültig.

Des weiteren habe ich die Satzung und das Protokoll beim Amtsgericht in Schwäbisch Hall mit dem Antrag auf Eintrag unseres Clubs als e. V. eingereicht. Es wird sich also in den nächsten Wochen ergeben, daß wir dann als eingetragener Verein (und somit der ganze Club als juristische Person) in Erscheinung treten.

Supercomputer

Und dann ist da noch der Supercomputer mit dem Kontrollämpchen rechts oben. Wenn es blinkt, zeigt es an, daß die Firma pleite ist.

Hallo Clubmitglider,
ich bin der Eckehard Kuhn und habe bei der letzten Clubversammlung das Amt des Hardwarecoordinators übernommen.

Zuerst zu meiner Person ,ich bin Jahrgang.64 und bin zur Zeit in der Ausbildung zum Informationselektroniker.Dadurch stehen mir Messgeräte und Zeit zur Verfügung.
Ich besitze das Model 1 mit 3 LWs ss sd ,einem Guen Data Drucker DMP-1180 (Kombatibel zum Epson MX 80), eine HRG sowie eine RS 232 Schnittstelle.
Da unsere Hardareprojekte bisher immer wieder eingeschlafen sind , möchte ich jetzt die Sache in Schwung bringen .
Darum rufe ich alle Hardware Interessierten auf, mir zu schreiben oder an Peter Stevens (der es an mich weiterleitet). Ihr solltet mir mitteilen ob Ihr nur an den fertigen Produkten interessiert seid ,oder ob Ihr auch mitentwickeln und mitbauen wollt.
Als Projekte stehen demnächst der ECB-Bus und eine einfache Relaissteuerung zur Verfügung.
Ich hoffe auf viel Post.

Eckehard Kuhn

Kontaktadresse
Eckehard Kuhn
Im Dorf 14
7443 Frickenhausen1
Tel.: 07022/45417
ab 17 Uhr

Nachtrag zu "Autoboot" 40- auf 80-Track

Selbstbootende 40-Track-Disketten auf einem 80-Track-Laufwerk zu fahren war schon oft ein Problem. Diesbezüglich möchte ich Euch an die Artikel im INFO erinnern. So gab es Hard- und Softwarelösungen zu diesem Thema.

Auch ich hatte einen Lösungsvorschlag in der 8. INFO-Ausgabe auf Seite 37 veröffentlicht. Mehrfach gab es aber beim Ausprobieren des Tips durch Euch keinen Erfolg.
Nun, die ganze Geschichte hat, wie ich herausgefunden habe, einen kleinen Haken. Man kann aber sagen, daß es doch funktioniert. Wie bei "Radio Erivan":

Im Prinzip ja, aber ...   ... es sollte kein Doubler eingebaut sein!

Dies fand ich aber auch erst heraus, als ich mein System auf Double-Density umbaute. Nach dem Umbau war kein Booten der 40-Track-Disk's mehr möglich. Mit dem Zurückstecken auf SD (mit dem normalen Floppy-Controler 1771) waren die Zap wieder gültig und 40-Track bootete ohne Probleme auf den 80'ern. Es muß also mit dem 2. Controler-IC auf der Erweiterungsplatine zusammenhängen. Da ich, als ich diesen Artikel schrieb, die entsprechende Hardware noch nicht besaß, konnte sich der Fehler bei mir auch nicht bemerkbar machen.

Die Zap aus meinem Artikel sind also nur für SD-Laufwerke gültig.

Warum dies so ist, kann ich Euch im Moment nicht erklären. Vielleicht weiß einer von Euch die Lösung dazu. Ich würde mich freuen, wenn ich eine Antwort von Euch dazu bekommen würde.

Im Prinzip ja, ...

Jens Neueder

Französisch: Abîmer le portrait.
Das Porträt ruinieren.
Die Fresse polieren.

Englisch: Like a dying duck in a thunderstorm.
Wie eine sterbende Ente im Gewitter.
Wie der Ochs vorm Berg.

In two shakes of a lambstail.
Während ein Lämmerschwanz zweimal wackelt.
Im Nu.

Listen von Basic-Files

Bei einem Gespräch mit einem Clubmitglied fiel mir auf, daß die bequemen LIST-Befehle im Disk-Basic doch noch nicht so gut bekannt sind. Soweit ich weiß gilt das Folgende für alle Versionen, also von Apparats- über GDOS- bis hin zum neuen NDOS80+10 Extendet Disk-Basic:

;	erste Zeile
/	letzte Zeile
↓	eine Zeile vor
↑	eine Zeile zurück
@	eine Seite vor
:	eine Seite zurück

Dieter Kasper

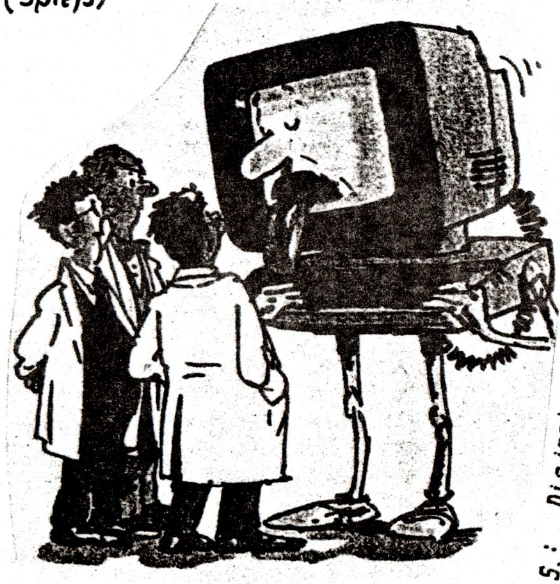
Nachtrag zum Grafik-Sonderheft

In der Grafik-Sonderausgabe befindet sich auf Seite 3 ein Fehler. Die Zeichen der Blockgrafik befinden sich nicht im Zeichensatz-ROM sondern werden "hard" erzeugt. In Zusammenarbeit mit Arnulf Sopp wollte ich darüber schon einmal einen Artikel verfassen. Leider konnten wir nicht 100%ig herausfinden, wo diese Zeichen genieriert werden. Vielleicht weiß es ein anderes Clubmitglied. Das Zeichensatz-ROM enthält nur die Sonderzeichen von 1 bis 31, die durch einen kleinen Trick auch auf dem Bildschirm dargestellt werden können (frag' mal Arnulf) und die ASCII-zeichen von 32 bis 127.
PRINT CHR$(128) und größer ergibt dann die Grafikzeichen und lange Strings von Leerzeichen.

Ich hoffe, Du bist mir für die "Besserwisserei" nicht böse und verbleibe

mit besten Grüßen

Ditpit (Spieß)



P.S.: Diejenigen, die nicht am Clubtreffen anwesend waren, erhalten mit dieser INFO-Ausgabe das Grafik-Sonderheft.

Zehnzeilerwettbewerb

Ein weiterer "Zehnzeiler" ist hereingekommen. Und damit sind es nun schon zwei. Ich vermute, daß es sich hier um eine Routine für EPSON & Kompatible handelt. Dies läßt sich aber leicht ändern, da nur in Programmzeile 50000 druckerspezifische Befehle stehen. Merkt euch also auch dieses Programm für die Endwertung vor.

```
10 ' ***************************************************************
20 ' **                    EIN "ZEHNZEILER"                       **
30 ' **                  --------------------                     **
40 ' **     GRAPHIK-HARDCOPY-PROGRAMM IN BASIC                    **
50 ' **          (C) HERBERT ALBERS, 2117 WISTEDT                 **
60 ' **       ========== DISK-BASIC ==========                    **
70 ' ***************************************************************
80 '
90 '
50000 LprintCHR$(27);CHR$(64);: LprintCHR$(15);:
      LprintCHR$(27);CHR$(51);CHR$(12);
50010 PO=15296: ZN=1: LprintString$(132,35): FORY=0TO47:GZ$="":
      FORX=0TO127:
      IFPOINT(X,Y)=-1THENGZ$=GZ$+CHR$(239)ELSEGZ$=GZ$+CHR$(32)
50020 NEXTX: IFY=ZNTHENZN=ZN+3: GOSUB50040
50030 GZ$="# "+GZ$+" #": LprintGZ$: NEXTY:
      LprintString$(132,35): Lprint: Lprint: Lprint: Return
50040 P=2: PO=PO+64: FORI=POTOPO+63: T=ASC(CHR$(PEEK(I))):
      IFT(127THENMID$(GZ$,P,1)=CHR$(T)
50050 P=P+2: NEXTI: Return
```


HRG: 384X192 oder 192X192

Die HRG-Pixel-Matrix des Bildschirms mit einer Auflösung von 384X192 Punkten ist die Normalversion der HRG. Dies läßt sich aber leicht ändern, indem Ihr mit dem Befehl PrintCHR$(23) Eure Bildschirmausgabe auf "Großschrift" (Ausgabe von 32 Zeichen pro Zeile) umschaltet. Nun ist auch bei der HRG ein Pixel von doppelter Breite wie sonst üblich eingestellt. Dies ergibt dann die 192X192.

Die Erklärung für diese "HRG-Soft-Änderung" ist recht einfach. Durch die Umschaltung wird nur jede 2. Schreibstelle des Bildschirms angesprochen. Dies gilt auch für die parallelgeschaltete HRG, da diese ihre Ansteuerung von unserem Computer abgezwackt hat.
Eine Schreibstelle beinhaltet sechs Pixel. Da jede 2. Schreibstelle nicht ausgegeben wird, ergibt sich dadurch der STEP 12 um jeden 1. Pixel einer gültigen Schreibstelle zu erreichen. Um die jeweilige Bildposition zu füllen benötigt man nun noch die Schleife 0-5. Die Bildposition wird dann mit sechs Pixel waagrecht aufgefüllt.

Leider ist aber dabei das dargestellte Bild nicht quadratisch, wie es sich aus der Pixel-Matrix erkennen lassen könnte. Auch der Syntax zum Füllen der Bildfläche muß anders heißen:

```
.... FOR X=0TO383 STEP12: FOR Y=0TO191
.... FOR A=0TO5: #SET(X+A,Y): NEXTA,Y,X
```

Einen Vorteil hat die ganze Sache aber doch. Die Ausgabezeit für errechnete Grafiken. Man kann bei dieser "Großschrift-HRG" die Grafiken in kürzerer Zeit laufen lassen und hat doch eine vollwertige Bildschirmausgabe. Natürlich in groberem Raster. Es besteht aber so die Möglichkeit, den Aufbau einer HRG zu verfolgen und den Trent (wird's was oder nicht) zu erkennen. Wie gesagt: Die Zeitdauer für die aufzubauende HRG verringert sich auf ein Viertel bis fast auf die Hälfte (je nach Programmierung).
Einen weiteren Nutzen hat diese Ausgabeart wenn man die HRG vom BASIC aus ohne Treiber steuert. Hier braucht man nur jede 2. Schreibposition der 1024 Schreibstellen über die OUT-Befehle ansprechen. Ich möchte dazu auf den Programmteil in INFO 10 (Seite 31, Programmzeilen 40-46) hinweisen.

Wenn dann die HRG-Darstellung in der "Grobversion" erfolgreich gelaufen ist, wird die Feindarstellung sicher noch besser sein. Probiert es doch einmal aus.

Jens Neueder

HEFT
13
Mai
1986

* * * DIRECTORY lesen und KOMMENTARE speichern * * *

```
10 CLS:CLEAR2500:DIMA$(200):PRINT" Directory lesen und verarbe
iten (einschl. unsichtbarer Files)
15 REM       >>>>> von KaJot Mühlenbein * Mai 1986 <<<<<
20 PRINTSTRING$(63,"*")
30 PRINT:PRINT"In welchem Laufwerk befindet sich die zu lesend
e Diskette? ";
35 LW$="":LW$=INKEY$:IFLW$=""THEN35ELSEPRINTLW$
40 GOSUB1000
45 PRINT:INPUT"Directory-Ausgabe mit Numerierung <j oder ENTER
> ";NN$
48 INPUT"Mit Kommentaren <j oder ENTER> ";K1$:IFK1$<>""INPUT"K
ommentare speichern <j oder ENTER> ";SP$:IFSP$<>""OPEN"O",2,"F
ILEKOMM/DAT:"+LW$
49 GOTO1500
50 CLS:FORI=0TOJSTEP45:FORK=0TO44STEP3:FORL=0TO2
55 IFA$(I+K+L)=""THENEND
60 PRINTUSING"##";I+K+L+1;
65 PRINT") "A$(I+K+L);"     ";
70 NEXTL:PRINT:NEXTK
80 INPUT:NEXTI:END
1000 ONERRORGOTO1130
1010 DR$="DIR/SYS:"+LW$
1020 OPEN"R",1,DR$
1030 FORI=3TOLOF(1)
1040 GET1,I
1050 FORII=0TO7
1060 FIELD1,(II*32)AS D$,1AS A$,4AS D$,8AS FS$,3AS FE$
1070 IFNOT(CVI(A$+CHR$(0))AND208)=16THEN1100
1080 A$(J)=FS$
1085 IFASC(FE$)=>65,A$(J)=FS$+"/"+FE$
1090 J=J+1
1100 NEXTII,I
1110 CLOSE1
1120 RETURN
1130 RESUMENEXT
1500 CLS:FORI=0TOJ-1
1520 IFK1$<>""PRINT"Kommentar für "A$(I);"  (bis zu 3 Bildschi
rmzeilen)":LINEINPUTKD$
1525 IFSP$<>""K%=K%+1:KK$(K%)=KD$
1527 IFNN$<>""LPRINTUSING"##)";I+1;:LPRINT" ";
1530 LPRINTA$(I);TAB(17);LEFT$(KD$,64)
1533 IFLEN(KD$)>64LPRINTTAB(17);MID$(KD$,65,64)
1536 IFLEN(KD$)>128LPRINTTAB(17);RIGHT$(KD$,LEN(KD$)-128)
1538 CLS:NEXT
1539 IFSP$<>"":FORD%=1TOK%:PRINT#2,A$(D%-1);",";CHR$(34);KK$(D
%)CHR$(34);",";:NEXT
1540 CLOSE:CLS:PRINT§475," E N D E":END
```

(unter Verwendung von Guenther Wagners Grundprogramm aus INFO Nr.9, S.4)

* * * * *

Zuruecklesen der Kommentare mit Aenderungsmoeglichkeit

```
10 CLS:PRINT"          FILE-Kommentare lesen und aendern
20 CLEAR30000:DEFINTI:DEFSTRK,L:DIMK(160)
25 PRINT:PRINT"In welchem Laufwerk befindet sich die Diskette
? ";
27 LW=INKEY$:IFLW=""THEN27ELSEPRINTLW
30 OPEN"I",1,"FILEKOMM/DAT:"+LW
40 IFEOF(1)THENCLOSE:GOTO60
50 I1=I1+1:INPUT#1,K(I1):GOTO40
60 FORI2=1TOI1-1STEP2:CLS:PRINTK(I2)" :":PRINTK(I2+1):PRINT:A$
="":INPUT"Kommentar aendern <j oder ENTER> ";A$:IFA$<>""THENLI
NEINPUT"Neuer Text:
";K(I2+1):AE=1
70 NEXT:IFAE=1OPEN"O",1,"FILEKOMM/DAT":FORI3=1TOI1STEP2:PRINT#
1,K(I3);",";CHR$(34);K(I3+1);CHR$(34);",";:NEXT
80 END
```

Ein paar kleine Änderungen

1. NAME/CMD

Auf der GDOS 2.4-Diskette befindet sich ein Programm, das die Funktionen eines Monitors, Texteditors und Assemblers übernimmt. Es benutzt den Speicher ab D000. Texteditor und Assembler erscheinen mir persönlich uninteressant, aber der Monitor läuft nicht schlecht. Es gibt zwei Nachteile: Das zu editierende Programm darf keinen Platz über D000 beanspruchen und es darf erst bei 5300 beginnen. Dieser zweite Punkt ist besonders ärgerlich, weil die meisten Programme bei 5200 beginnen, dafür aber selten über D000 hinausgehen.
Warum darf der Platz von 5200-5300 nicht benutzt werden? Weil NAME/CMD dort einen Stack plaziert. Diesen Stack habe ich verlegt. Nun überschreibt er einige Standardtexte, die normalerweise für das Textprogramm gedacht sind ("Gnädige Frau" etc.), also relativ wertloses Zeug, wenn man das Textprogramm sowieso nicht nutzt.
Die Änderungen sind sehr viel kürzer als der ganze vorstehende Text:

```
D000    LD    SP,0D080H
E95E    LD    SP,0D11AH
```

80-Zeichen-Karte: X80, Sedit und Tscrips

Wie Ihr im letzten Info lesen konntet, bin ich mit der 80-Zeichen-Karte von Schmidtke belastet und muß das beste draus machen. Die ersten drei Programme mußten nun meine miesen Programmierfähigkeiten über sich ergehen lassen.

Das erste war der Treiber der 80-Zeichen-Karte, auch X80/CMD genannt. Das Ding hätte von mir stammen können, so stümperhaft war es programmiert (d.h. falls es nicht nur ein Prototyp sein soll). Also habe ich erstmal einige Sachen rausgeschmissen, einiges dazugepackt und das Ganze etwas nach oben verschoben, damit es nicht soviel Platz verbraucht. Eigentlich ist nur die Programmierung der 80-Zeichen-Karte wie im Original-Programm. Die einzelnen Änderungen aufzuführen, wäre unmöglich und für 90 Prozent der Mitglieder auch uninteressant. Wer also auch dummerweise die Karte erstanden hat, kann sich bei mir Sourcelisting und Programm bestellen, natürlich kostenlos, alles als Freeware. Es gibt bei mir mehrere X80-Versionen: eine benutzt den alten Bildschirmspeicher als Programm-Bereich, so daß absolut kein RAM im Vergleich zu vorher fehlt; eine andere Version kann zwischen 64- und 80-Bildschirm hin- und herschalten, was nützlich ist, wenn man nicht ganz auf Grafik verzichten möchte. Alle Versionen können auf den 64-Bildschirm zurückschalten, ohne das Programm X64 aufrufen zu müssen und außerdem auch die Codes 0-20h auf den Bildschirm bringen. Wer Interesse hat, kann sich wie gesagt bei mir melden.

Das zweite Programm war der Bildschirmeditor. Im letzten Info hatte ich ja nach dem Listing gefragt und habe es auch von Hartmut bekommen. Weil auf dem Clubtreffen gesagt wurde, daß wir über die Antworten auf unsere Fragen aus dem Fragekasten berichten sollten, will ich hiermit eingestehen, daß die Frage selten dämlich war. Ich habe nämlich die ganze Zeit den Loader von Sedit disassembliert (der ja nur den eigentlichen Bildschirmeditor lädt) und versucht, ihm die Funktionen des Bildschirmeditors anzudichten. Konnte natürlich nicht klappen!
Nun hatte ich also das richtige Listing vorliegen, das übrigens etwas von dem von Röckrath veröffentlichten abweicht, und habe danach wiederum mehrere neue Sedit-Versionen erstellt. Eine ist ein gekürztes Sedit ohne die Umlaut- und Entprellungsroutinen, denn die kennt mein Gdos 2.4 selber. Die anderen Versionen vereinigen die Funktion des 80-Zeichen-Treibers mit denen eines Bildschirmeditors. Wer daran Interesse hat, kann sich wiederum bei mir melden; ein Sourcelisting würde das Info überfordern.

Als Programm Nr. 3 kam Tscrips 4.0 unter meine Grabbeln (Pardon: Finger). Dort gibt es überall massenhaft Platz, weil die Autoren mit unnützen Routinen und Anmerkungen nicht gespart haben. Allein das Sourcelisting der Änderungen ist schon wieder zu lang (vor allem, weil jede zweite Zeile ein neues ORG benötigt), um hier veröffentlicht zu werden. Interessierte sehen in der Adressenliste nach und lassen dann die Post verdienen, entweder über Telefon oder Brief. übrigens dürften die Änderungen auch leicht auf andere 80-Zeichen-Karten übertragbar sein.

Gerald Schröder

2. TSCRIPS 5.4
Tscrips 5.4 bietet mehr Funktionen als das alte 4.0, hat dafür aber weniger Textspeicher, was sich sicher noch beheben läßt, denn die Autoren verschwenden immer noch massenhaft Platz für ihre "Copyright"-Routinen. Bei mir meldete sich das 5.4 aber nach <BREAK> <P> <ENTER> mit "Line Printer Not Ready" ab. Grund: Scheinbar auf einem Genie geschrieben, wurde der Port FD zur Drucker-E/A benutzt, statt der traditionellen Speicherstelle 37E8. Somit muß an folgenden Speicherstellen "IN A,(FDH)" (= DBFD)
in "LD A,(37E8H)" (= 3AE837) umgeändert werden:
5F63
663F
6650
86EE

Es fehlt noch der Ausgabebefehl, der an 86E3 steht. Hier muß "OUT (FDH),A" (= D3FD) in "LD (37E8H),A" (= 32E837) geändert werden.

3. Tscrips 4.0 Version für Epson xx 80
Es gibt inzwischen viele Computer, deren Taktfrequenz erhöht wurde. Das wirkt sich ungünstig auf Tastatureingaben aus, denn der Cursor huscht dann viel zu schnell über den Bildschirm und blinkt auch noch äußerst penetrant. Die folgenden Zeilen geben Speicherstellen im Tscrips 4.0 an, die Warte- oder Wiederholroutinen enthalten. Die Werte können je nach Taktfrequenz und Tippgeschwindigkeit oder Gefallen des Benutzers geändert werden.

5FDA/B: Blinken des Cursors; solange bleibt der Cursor sichtbar (normal: 400h, steht aber im Speicher als "0004", aufgrund LSB/MSB)
5FE9/A: Blinken des Cursors; solange bleibt der Cursor ausgeschaltet (normal: 400h bzw. "0004")
60B9/A: Nach dieser Zeit wird die gedrückte Taste zum 1. Mal wiederholt (normal: 400h bzw. "0004")
60B2/3: Nach dieser Zeit wird die gedrückte Taste zum 2., 3. usw. Male wiederholt (normal: 1h bzw. "0100")
60DA/B: Wartezeit bei Entprellung (normal: 2EEh bzw. "EE02")
617B/C: Wartezeit nach Eingabe eines Zeichens, bestimmt die Blinkfrequenz des Cursors mit (normal: 4ACh bzw. "AC04")

85A6/7: Verzögerung nach Ausgabe eines Zeichens bei PROBEDRUCK (normal: 200h bzw. "0002")
8E66/7 und 8E78/9: Blinkfrequenz des <ENTER> bei der Titelgrafik (normal: 4000h bzw. "0040")

Wie gesagt gelten die Angaben nur für Tscrips 4.0 Version "Epson-Drucker". Bei anderen Tscrips-Versionen dürften die Angaben nur bis 617B/C stimmen. Auf jeden Fall sollte vor dem Ändern kontrolliert werden, ob die Normalwerte übereinstimmen. Wenn nicht, liegt eine andere Version vor.

Gerald Schröder

Tips für SCRIPS

Der Leser möge nach Belieben vor den im Titel genannten Programmnamen ein T oder hinter ihn IT setzen. Die Rede ist von SCRIPSIT und seinen Enkeln, besonders TSCRIPS 5.4. Da alle besseren Textprogramme gewisse Steuerzeichen direkt von der Tastatur einlesen können bzw. die eine oder andere Tücke haben, ist das folgende vielleicht auch für Benutzer anderer Software interessant.

Im Clubinfo Nr. 12, S. 92 rief ich um Hilfe, weil mein Textprogramm die Druckercodes für die freiladbaren Zeichen nicht an den Gemini weitergab. Es kamen gutgemeinte und interessante Ratschläge von zwei Seiten. Die tatsächliche Lösung aber war viel simpler als die vorgeschlagenen Zaps der Clubfreunde: TSCRIPS gibt bei einer bestimmten Einstellung des linken Randes so viele Blanks auf den Drucker aus, bis die gewünschte Stelle erreicht ist. Damit ist die Zeile aber nicht mehr "frisch", so daß die ersehnte Umschaltung nicht mehr stattfinden konnte. Abhilfe:

Was TSCRIPS kann, kann der Drucker schon lange. Z. B. mit der Codefolge ESC-'M'-09 wird der linke Rand im Drucker (also nicht im Textprogramm) auf die neunte Stelle gesetzt. Bei den EPSON-Typen lautet diese Sequenz ESC-'l'-09. Wenn gleich am Beginn eines Textes diese Bytefolge auf den Drucker ausgegeben wird, kann TSCRIPS mit der Formatanweisung >LM=0 abgespeist werden. Die Syntax dieser Druckersteuerung ist für TSCRIPS $M&09& (EPSON: $l&09&).

Nun war mein Problem plötzlich aus der Welt. Ohne beim Umschalten auf einen selbstdefinierten Zeichensatz zunächst LM=0, dann den Umschaltbefehl und schließlich wieder LM=9 eingeben zu müssen, könnte ich jetzt in Russisch weiterschreiben, wenn ich Russisch könnte:

Будьте так любезны пришить эту пуговицу.

>LM=0 hat noch einen weiteren Vorteil: Wenn beispielsweise ein unterstrichenes Wort getrennt wird, finden wir andernfalls den Strich auch unter den Blanks des linken Randes. Wenn TSCRIPS glaubt, es gebe keinen (LM=0), dann wird auch nichts unterstrichen.

A propos, die Erfinder der Version 5.4, K. Trappschuh und E. Sörensen, haben das auch gemerkt und mit einer Unterstreichungsfunktion gekontert. Wenn ein Textstück zwischen '^' eingeschlossen ist, wird es unterstrichen . Aber die Unterstreichung des Druckers sieht bei weitem besser aus. Außerdem sind die Blanks bei der Unterstreichung bei der ^-Methode ausgespart.

Da ist noch ein Problem, das den Leser vielleicht schon einmal geplagt hat. TSCRIPS 5.4 versteht die Zeichenfolge $S als Aufforderung zur Breitschrift, $T hebt sie wieder auf. Druckercodes, die mit diesen beiden Zeichen beginnen, sind daher nicht mehr mit dem Dollarzeichen möglich. Die Abhilfe ist simpel, um beispielsweise mit der Folge ESC-'S'-00 Superscript einzuschalten: Statt $S&00& kann man ebenso gut &1B5300& eingeben, was die Hex-Entsprechung dieser Zeichen ist. Das abschließende ESC-T, das für TSCRIPS nicht Sub- oder Superscript, sondern die Breitschrift aufhebt, wird dann eben als &1B54& in den Text eingefügt.

Wer einen ITOH oder ein anderes Textprogramm hat, wird die obigen Codes zwar nicht brauchen. Parallele Fälle mit anderen Codes sind jedoch in jedem Falle zu befürchten. Insoweit kann dies nur eine Anregung sein.

Bei dieser Gelegenheit möchte ich ein Programm vorstellen, dessen Grundversion Alexander Schmid aus dem Bremerhavener Club entwickelt hat. Es leitet Druckerausgaben in den Textpuffer von TSCRIPS um. So kann man beispielsweise JKL-Hardcopies, DIR,P-Ausgaben, PRINT- oder DR-Ausgaben mitten in seinen Text einbauen. Als Beispiel hier gleich die Hardcopy des Programms als Sektordump:


```
drv: 0           frs: 0000h       drs: 0113h
000000: 011C 0442 CDD5 4C7E FE4E 2005 2A66 4318    B  LB N  *fC
000010: 092A 2640 2266 4321 BB35 2226 40C9 0146  *&§"fC! 5"&§  F
000020: BB35 2A2B 7C79 015F 26FE 2020 0379 1830  5*+öy _&     y 0
000030: 3002 CBFF 1133 32FE 2328 1E1C FE24 2819 0     32 #(    $(
000040: 1E36 B828 1411 4533 FE3E 280D 1146 35B9  6 (  E3 >(  F5
000050: 2807 FE60 200A 1130 3670 2372 2373 2378 (  '    06p#r#s#x
000060: 7723 222B 7CC9 0202 0442 0000 0000 0000 w#"+ö      B
000070: 0000 0000 0000 0000 0000 0000 0000 0000
000080: 0000 0000 0000 0000 0000 0000 0000 0000
000090: 0000 0000 0000 0000 0000 0000 0000 0000
0000A0: 0000 0000 0000 0000 0000 0000 0000 0000
0000B0: 0000 0000 0000 0000 0000 0000 0000 0000
0000C0: 0000 0000 0000 0000 0000 0000 0000 0000
0000D0: 0000 0000 0000 0000 0000 0000 0000 0000
0000E0: 0000 0000 0000 0000 0000 0000 0000 0000
0000F0: 0000 0000 0000 0000 0000 0000 0000 0000
```

Dieser Bildschirmausdruck ist wie gewohnt einfach mit JKL entstanden. Der Leser möge seinen spontanen Enthusiasmus allerdings vorläufig etwas zügeln; mein SYS15/SYS (für DDE) lädt in den Adreßbereich ab 3000h. Das DDE anderer G-DOS-Versionen oder gar SUPERZAP laden in den Anwenderspeicher, wo dann TSCRIPS gnadenlos überschrieben wird. Daher ist bei der Anwendung des Programms eine gewisse Vorsicht geboten.

Im Dump fällt auf, daß auch TSCRIPS-Sonderzeichen wie $, # usw. korrekt wiedergegeben werden. Die Routine sorgt dafür, daß sie erkannt und nicht direkt in den Text eingebaut werden. Stattdessen wird ihre Hex-ASCII-Darstellung zwischen zwei &-Zeichen gesetzt. Ebenso wird ein Blank immer durch das Zwangs-Blank '_' ersetzt, so daß Blanks nicht gedehnt werden können, um bei rechtem Randausgleich die Zeile zu streamlinen. Da es sich kaum je um gewöhnlichen Sprachtext, sondern um Ausgaben der obigen Art handeln dürfte, erschien mir das sinnvoll.

Das Programm hat einen Ladebereich, der auf das Genie III s zugeschnitten ist. Bei anderen Maschinen muß der User jeweils ein freies Plätzchen suchen. Bei nicht allzu langen Texten bietet sich das Himem an. Wer den MBA hat, ist wieder mal fein raus. Auch die Codes, nach denen das Programm die Druckerausgaben absucht, sind nicht für alle Anwender gleich. Es hängt ab vom Textprogramm. Die Routine muß daher jeweils angepaßt werden.

Zum Programmablauf ist nicht viel zu sagen, denn das Listing auf der folgenden Seite ist ausführlich kommentiert. Nur so viel: Da die Routine in den Sektorpuffer lädt, kann sie während der TSCRIPS-Arbeit abgerufen werden. Sie überschreibt nicht den Anwenderspeicher. Zunächst wird geprüft, ob beim Aufruf ,N an den Programmnamen angehängt wurde, um die Druckerumleitung wieder rückgängig zu machen. Falls nein, wird der Beginn des eigentlichen Arbeitsprogramms beim Label newdrv als Druckertreiberadresse in den DCB geschrieben.

Bei jeder vermeintlichen Druckerausgabe wird dann geprüft, ob ein Steuerzeichen wie CR o. ä. vorliegt. Falls ja, wird das Bit 7 gesetzt, weil TSCRIPS das gerne so hat. Dann werden die Blanks zu Zwangs-Blanks (_) gemacht und TSCRIPS-Sondercodes zwischen &-Zeichen als Hex-ASCII-Strings eingeschlossen. Gewöhnliche Zeichen werden unverändert in den Textpuffer geschrieben. Da der Puffer sich auf diese Weise verlängerte, wird schließlich sein neues Ende in den Zeiger von TSCRIPS gepatcht. Das war's.

```
                ; Programm zur Umleitung von Druckerausgaben
                ; auf den TSCRIPS-Textpuffer
                ; 1986 by Alexander Schmid und Arnulf Sopp

4366            drvbuff EQU     4366h       ;hier Treiberadr. gepuff.
7C2B            buffend EQU     7C2Bh       ;Zeiger auf Pufferende

4204                    ORG     4204h       ;im Sektorpuffer
4204  CDD54C    start   CALL    4CD5h       ;folgt ein Parameter?
4207  7E                LD      A,(HL)      ;welcher?
4208  FE4E              CP      'N'         ;Umleit. rückgäng. mach.?
420A  2005              JR      NZ,init     ;falls initialisieren
420C  2A6643            LD      HL,(drvbuff) ;alte Treiberadresse
420F  1809              JR      return      ;dort weiter
4211  2A2640    init    LD      HL,(4026h)  ;Dr.-Treiberadresse
4214  226643            LD      (drvbuff),HL ;retten
4217  21BB35            LD      HL,newdrv   ;neuer Treiber
421A  222640    return  LD      (4026h),HL  ;in den DCB schmuggeln
421D  C9                RET                 ;zurück ins Betriebssyst.

35BB                    ORG     35BBh       ;beim Genie III s frei
35BB  2A2B7C    newdrv  LD      HL,(buffend) ;derzeitiges Pufferende
35BE  79                LD      A,C         ;zu "druckendes" Zeichen
35BF  015F26            LD      BC,'&_'     ;2 TSCRIPS-Steuerzeichen
35C2  FE20              CP      ' '         ;normales Schriftzeichen?
35C4  2003              JR      NZ,testvid  ;falls nicht Blank
35C6  79                LD      A,C         ;Zwangsblank für TSCRIPS
35C7  1830              JR      chrok       ;ausgeben
35C9  3002      testvid JR      NC,testctl  ;falls kein Vid.-Steuerz.
35CB  CBFF              SET     7,A         ;Steuerz., Bit 7 setzen
35CD  113332    testctl LD      DE,'23'     ;für "23/24/26/40/5F/60"
35D0  FE23              CP      '#'         ;"#" für freidefin. Z.?
35D2  281E              JR      Z,ctl       ;falls ja
35D4  1C                INC     E           ;sonst DE auf "24" setzen
35D5  FE24              CP      '$'         ;Steuerz. "$" für ESC?
35D7  2819              JR      Z,ctl       ;falls ja
35D9  1E36              LD      E,'6'       ;sonst DE auf "26" setzen
35DB  B8                CP      B           ;Stz. "&" für Hexcodes?
35DC  2814              JR      Z,ctl       ;falls ja
35DE  114533            LD      DE,'3E'     ;ASCII-Code für ">"
35E1  FE3E              CP      '>'         ;LINE-FORMAT-Zeichen?
35E3  280D              JR      Z,ctl       ;falls ja
35E5  114635            LD      DE,'5F'     ;ASCII-Code für "_"
35E8  B9                CP      C           ;TSCRIPS-Zwangsblank?
35E9  2807              JR      Z,ctl       ;falls ja
35EB  FE60              CP      '`'         ;Steuerz. ` für IMAGE-x?
35ED  200A              JR      NZ,chrok    ;falls nein
35EF  113036            LD      DE,'60'     ;ASCII-Code für "`"
35F2  70        ctl     LD      (HL),B      ;& leitet Hex-Sequenz ein
35F3  23                INC     HL          ;nächste Pufferstelle
35F4  72                LD      (HL),D      ;MSB des Hex-ASCII-Worts
35F5  23                INC     HL          ;nächste Stelle
35F6  73                LD      (HL),E      ;ASCII-LSB
35F7  23                INC     HL          ;nächste Stelle
35F8  78                LD      A,B         ;& schließt Hex-Seq. ab
35F9  77        chrok   LD      (HL),A      ;Zeichen in den Puffer
35FA  23                INC     HL          ;neues Pufferende
35FB  222B7C            LD      (buffend),HL ;neuen Zeiger patchen
35FE  C9                RET                 ;erledigt

4204                    END     start       ;dort Einsprung

00000 Fehler
```

An der Entwicklung menschenwürdiger Interfaces zum Computer hängt es, ob sich die beruflichen Fähigkeiten des Menschen erweitern und ausnützen lassen oder ob sie verkümmern. Soll die Entwicklung vom Affen wieder zum Affen gehen?

Sondertasten diverser Genies

Die Genie-Computer, die mit der Version 2.4 von G-DOS beheizt werden, haben im Ziffernblock ein paar nichtnumerische Tasten, die frei programmiert werden können. Wie das geht, wird (bei TCS geradezu selbstverständlich) im Handbuch nicht erklärt. Aber in unserer Clubzeitung:

Die jeweiligen Boot-EPROMS laden in den Bildschirm eine Ziffer, die von SYS0/SYS im Laufe der Initialisierungen übernommen und in den Platzhalter x in dem Text "OVLx/SYS" geschrieben wird. Das so bestimmte Overlay wird sodann geladen. Dabei werden ganz bestimmte Bearbeitungsroutinen für besondere Features des jeweiligen Modells hereingeschlürft. Diese modellspezifischen Overlays enthalten u. a. auch die Belegung der besagten Sondertasten, bei den meisten Modellen im relativen Sektor 05 des Overlays.

Die Belegung ab Werk ist ziemlich abstrus. Das Pluszeichen macht wohl noch einen Sinn, aber z. B. mit dem Linefeed auf einer P-Taste des 63s wird man kaum etwas anfangen können. Ob die Belegung der PRINT-Taste mit dem PRINT-Text sinnvoll ist, darüber läßt sich streiten. In diesem Bereich sind auch der Dezimalpunkt und je nach Modell das Komma, der ESC-Code und das Minuszeichen kodiert. Da die Belegung all' dieser Tasten auf Diskette steht, ist sie mit DDE änderbar.

Die beiden Sektordumps im Anschluß an diesen Text zeigen oben die Originalbelegung beim 63s und darunter meinen Vorschlag einer Modifikation. Unterstrichen sind diejenigen Bytes, mit denen die Sondertasten belegt sind.

Bei meinem Änderungsvorschlag wird zunächst eine Hex-Tastatur realisiert, die für die Arbeit in Maschinensprache unentbehrlich ist. Die Ziffern A-F sind als Kleinbuchstaben kodiert, das h für den Hex-Kenner als Großbuchstabe. In der Anzeige erscheinen sie jedoch genau umgekehrt. Offenbar wird demzufolge bei der Anzeige das Bit 5 invertiert. Das ist wichtig, wenn der Leser die Tasten mit anderen Codes belegen will.

Die Hexziffern entsprechen den Tasten P1-P3. Ohne Shift erzeugen diese die Ziffern A-C, mit Shift D-F. Man sieht im Sektordump, daß sie dort von rechts nach links niedergelegt sind. Die Tasten P4 und P5 erzeugen je nach Shift-Status den Hex-Kenner h und die Zeichen +, / und *. Das Komma, das Minuszeichen und der ESC-Code (1Bh) wurden belassen. Der Code 1Bh wird übrigens sowohl mit als auch ohne Shift ausgegeben. Shift-ESC wäre deshalb noch vakant.

Der Textstring PRINT, der - wer hätte das gedacht? - mit der PRINT-Taste abgerufen wird, ist aber wegen penetranter Überflüssigkeit (für einen BASIC-Ignoranten wie mich) ersetzt. Jetzt lautet er EQU $-. Das ist ein unbequem zu tippender, jedoch recht häufig vorkommender String bei der Arbeit in Assembler. Ihm muß jeweils noch der Offset hinterhergetippt werden.

Wenn man die PRINT-Taste ganz deaktivieren möchte, um sie z. B. als Triggertaste für irgendeine Funktion einzusetzen, genügt es, das Byte 31h dieses Sektors auszunullen. Bei vorübergehendem Bedarf kann dies auch im Speicher an der Stelle 3EBCh geschehen, wo dieser Code nach dem Laden landet. Dazu muß im Systembyte 1, das auf den Port FAh (63s) auszugeben ist, das Bit 0 gesetzt werden, um das freie RAM "hinter" dem memory-mapped Bildschirm adressieren zu können.

Es steht dem Leser frei, beliebige andere Codes in diese Plätze auf der Diskette zu zappen. Gibt es alternative Vorschläge im nächsten Clubinfo? Dabei ist lediglich zu beachten, daß die Anzahl der erlaubten Zeichen nicht überschritten wird. Nur die PRINT-Taste gewährt Platz für sechs Codes, alle anderen gestatten nur jeweils ein Zeichen. Hat übrigens schon jemand herausgefunden, wie die Doppelnull erkannt und auf den Bildschirm erzeugt wird? Das ist so ziemlich die überflüssigste Taste des Computers, die danach schreit, sinnvoller belegt zu werden (kaufmännische User mögen mir verzeihen).

Bei den Untersuchungen für diesen Beitrag fiel mir auf, daß in OVL5/SYS für den Speedmaster eine Lücke im Bildschirmtreiber (rel. Skt. 08) nach Füllung lechzt. Je nach dem, was die Hardware hergibt, kann hier noch dies oder jenes programmiert werden. Der Platz beträgt zwar nur einige Bytes, aber dort ließe sich auch ein Vektor in eine Zone mit mehr freiem Raum unterbringen.

Arnulf Sopp

```
H-DOS-Diskdaten-Editor Vers. 2.4
(C) 1986 by TCS / The HACKTORY

Datei: OVL4/SYS:1


drv: 1            frs: 0005h        drs: 007Dh
000500: 18EA D613 C821 883E CD35 3EFE 3FC0 CD7E X  S ! > 5> ?  ß
000510: 3DC8 3E06 2175 3EC3 763D 5052 494E 5420 = >F!u> v=PRINT
000520: 0000 1B1B 0A1A 0B1B 2B2B 0000 0000 2C2D §§ÄÄJZKÄ++§§§§,-
000530: 2E3F AF32 2104 CD00 3DC3 0E04 CDB5 0621 .? 2!D §= ND  F!
000540: 4140 0E02 3E23 32BC 3E16 0A06 037E D96F A§NB>#2 >VJFCß o
000550: 2600 3E0A CDB4 4CD9 CD19 3FD9 7DD9 CD19 &§>J  L  Y? ü  Y
000560: 3F23 10E9 167A 3E2B 32BC 3E2E 460D 20DB ?#P Vz>+2 >.FM
000570: AF16 0ACD 193F CDBE 06DD E5DD 212C 3F2E  VJ Y?  F   !,?.
000580: 440E 023E 2332 F53E 16CC 0603 CD0A 3F87 DNB>#2 >V FC J?
000590: 5F87 8783 5FCD 0A3F 8377 2310 EF16 5C3E _   _ J? w#P Vö>
0005A0: 2B32 F53E 2E43 0D20 E1DD E1AF D35B C97A +2 >.CM       Ä z
0005B0: D35B 1610 9257 DD23 DB5A DDA6 00C9 5F7A  ÄVP W # Z  §  _z
0005C0: D35B 1610 8257 E6F0 FE60 7B20 02CB DFD3  ÄVP W    'ä B
0005D0: 0110 3C50 5AC9 0F0F 010F 030F 030F 070F AP<PZ OOAOCOCOGO
0005E0: 070F 010F 0645 00F5 3A05 45EE 2032 7304 GOAOFE§ :EE  2sD
0005F0: F100 0001 0C5E 04ED 5B41 36F5 19F1 C342  §§AL^D ÄA6 Y  B
```

```
H-DOS-Diskdaten-Editor Vers. 2.4
(C) 1986 by TCS / The HACKTORY

Datei: OVL4/SYS


drv: 0            frs: 0005h        drs: 007Dh
000500: 18EA D613 C821 883E CD35 3EFE 3FC0 CD7E X  S ! > 5> ?  ß
000510: 3DC8 3E06 2175 3EC3 763D 4551 5509 242D = >F!u> v=EQUI$-
000520: 0000 1B1B 2A2F 2B48 6366 6265 6164 2C2D §§ÄÄ*/+Hcfbead,-
000530: 2E3F AF32 2104 CD00 3DC3 0E04 CDB5 0621 .? 2!D §= ND  F!
000540: 4140 0E02 3E23 32BC 3E16 0A06 037E D96F A§NB>#2 >VJFCß o
000550: 2600 3E0A CDB4 4CD9 CD19 3FD9 7DD9 CD19 &§>J  L  Y? ü  Y
000560: 3F23 10E9 167A 3E2B 32BC 3E2E 460D 20DB ?#P Vz>+2 >.FM
000570: AF16 0ACD 193F CDBE 06DD E5DD 212C 3F2E  VJ Y?  F   !,?.
000580: 440E 023E 2332 F53E 16CC 0603 CD0A 3F87 DNB>#2 >V FC J?
000590: 5F87 8783 5FCD 0A3F 8377 2310 EF16 5C3E _   _ J? w#P Vö>
0005A0: 2B32 F53E 2E43 0D20 E1DD E1AF D35B C97A +2 >.CM       Ä z
0005B0: D35B 1610 9257 DD23 DB5A DDA6 00C9 5F7A  ÄVP W # Z  §  _z
0005C0: D35B 1610 8257 E6F0 FE60 7B20 02CB DFD3  ÄVP W    'ä B
0005D0: 0110 3C50 5AC9 0F0F 010F 030F 030F 070F AP<PZ OOAOCOCOGO
0005E0: 070F 010F 0645 00F5 3A05 45EE 2032 7304 GOAOFE§ :EE  2sD
0005F0: F100 0001 0C5E 04ED 5B41 36F5 19F1 C342  §§AL^D ÄA6 Y  B
```

HRG-Hardcopy für das G III s

Mit der JKL-Erweiterung des H-DOS für das Genie I wurde der Bildschirm in seinen richtigen Proportionen wiedergegeben. Bei diesem Programm war das nicht möglich. Die vier verschiedenen Bildschirmformate des IIIs hätten jedes eine eigene Programmvariante erfordert, welchen Aufwand aber der Erfolg nicht mehr rechtfertigen würde. Hier wurde ein Kompromiß gewählt: Eine anderthalbfache Dehnung in der Breite läßt die Formate 16X64, 24X64 und 25X80 Zeichen breiter als der Bildschirm erscheinen, während sich beim Format 32X64 ein leichtes Hochformat ergibt. Die Vereineinhalbfachung wird dadurch erreicht, daß abwechselnd ein Graphikcode einmal, der nächste zweimal ausgegeben wird usw..

Die Hardcopy-Routine arbeitet mit der hochauflösenden Graphik doppelter Dichte des Gemini-10X. Für andere Drucker müssen entsprechende andere Steuercodes ausgegeben werden. Um dem Leser dies zu ermöglichen, werden die druckerspezifischen Teile des Programms im folgenden näher erläutert. Das soll auch das Hauptanliegen dieses Artikels sein, denn zweieinhalb Seiten Listing eines immerhin recht anspruchsvollen Programms einigermaßen erschöpfend zu erklären, würde den Rahmen dieses Beitrags sprengen.

Da der Drucker nur neun Nadeln hat, von denen im Bit-Image-Modus nur acht gleichzeitig angesprochen werden können, werden pro Bildschirmzeile auf den Drucker zwei Halbzeilen ausgegeben. Die obere umfaßt immer acht senkrecht übereinanderstehende Bildpunkte, die untere den Rest der Bildschirmzeile, der je nach Format unterschiedlich hoch sein kann. Dieses Verfahren habe ich im Hinblick auf eine spätere Einbeziehung der ASCII-Anteile des Bildschirms gewählt (s. u.), die ja bildschirmzeilenorientiert sind. Für die Graphik allein wäre es auch simpler gegangen.

Dieses Programm druckt nur die HRG des IIIs aus. Die ASCII-Anteile bzw. die Blockgraphik, Inversschrift usw. bleiben unberücksichtigt. Da die Zeichensätze von der Diskette geladen werden, müßte ansonsten die Routine den Benutzer zunächst fragen, welcher denn gerade angezeigt wird, der müßte in einen Puffer geschrieben, seine Alternativzeichen oberhalb 80h errechnet und ebenfalls gepuffert werden. Nein, danke! Sobald die Garantie meiner neuen Denkprothese abgelaufen ist, wird das Zeichensatz-RAM für die CPU lesbar gelötet, so daß die Zeichen einfach mit der HRG auf dem Drucker gemischt werden können.

Da das Programm 244 Bytes lang ist, paßt es bequem in den Sektorpuffer. Daher die Ladeadresse 4204h. Um für Pufferbereiche, den Stack usw. keinen zusätzlichen Platz zu benötigen, werden zu diesem Zweck die Programmteile am Anfang überschrieben, denn sie dienen nur zu Initialisierungen und sind anschließend überflüssig. Auf diese Weise kann das Programm jederzeit und mit jedem Betriebssystem laufen, denn es belegt keinen Raum im Anwenderspeicher. Da es seinen eigenen Stack unterhält, läuft es auch unter BASIC. Das wäre sonst nicht möglich, weil der BASIC-Stack im Adreßbereich der Graphik liegt.

Zu den druckerspezifischen Eigenschaften des Programms: Es wird davon ausgegangen, daß das höchstwertige Bit des Druckkopfes oben liegt. Für Drucker, bei denen das umgekehrt ist (im Handbuch nachschlagen!), sind folgende Änderungen erforderlich: In Zeile 129 muß der Befehl RL (HL) in RR (HL) umdefiniert werden. Die AND-Maske, die für die zweite Halbzeile überflüssige Punkte löscht, wird dann auch in Zeile 37 nicht mehr mit RRA, sondern mit RLA ermittelt. Ohne das jetzt selbst nachprüfen zu können, scheint mir, daß dieser Punkt damit erledigt ist.

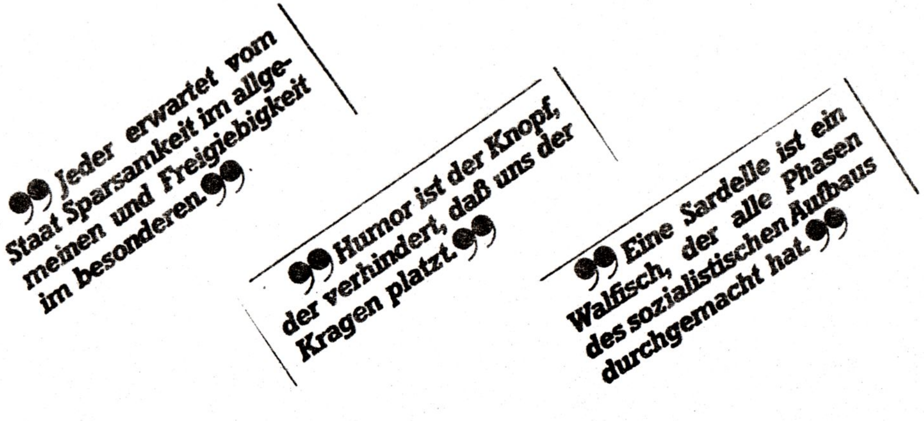
Weitere Druckerspezifika finden sich in der Tabelle am Ende des Listings. Bei Label lpinit wird der Drucker vorsichtshalber in den Einschaltzustand versetzt, weil evtl. zuvor schon eine bestimmte Randeinstellung o. ä. vorgegeben war. Beim Gemini geht das mit ESC '8'. Das nachfolgende 0Dh-Byte ist ein gewöhnlicher Wagenrücklauf (der in der neuesten Version herausgenommen ist), das abschließende FFh gibt lediglich dem Programm bekannt, daß der auszudruckende String hier zuende ist. Mit ESC 'M' 9 wird der linke Rand auf die 9. Stelle festgelegt. Bei Epson-Typen geschieht das mit ESC 'l' 9.

Beim Label lninit erscheint zunächst wieder ein Wagenrücklauf, dann die Einstellung des Zeilenabstandes mit ESC '3' n auf n/144 Zoll. Das Nullbyte ist nur ein Platzhalter für n, das jeweils neu definiert und dort eingeschrieben wird (Label inby144). Beim Gemini ist der senkrechte Abstand zweier Punkte 1/72, also 2/144 Zoll. Deshalb wird für eine Spalte mit allen 8 Dots das Register C in Zeile 63 mit 10h (16d) geladen, um 16/144 = 8/72 Zoll = 8 Punkte Zeilenabstand zu definieren. In der zweiten Halbzeile ist der Abstand je nach Bildschirmformat variabel. Er wird in den Zeilen 30-33 errechnet. Anpassungen für andere Drucker müssen hier ansetzen.

Die HRG doppelter Dichte des Druckers wird in Zeile 158 mit ESC 'L' n1 n2 initialisiert. Dabei bedeuten n1 und n2 in der Reihenfolge LSB-MSB die Anzahl der zu druckenden Dotspalten pro Zeile. Das MSB ist für jedes Format dasselbe (03h), aber das LSB (Label nrcols) variiert. Anpassungen an andere Matrixdrucker sind an dieser Stelle wohl am schwierigsten. Leser, die das Schicksal mit einem ITOH oder NEC geschlagen hat, beneide ich nicht, so gut diese Geräte in anderer Hinsicht auch sind.

Am Ende des Listings ist ein Bildschirmausdruck des Files RDLDEMO wiedergegeben. Es ist eine Phase vor der Anzeige einer neuen Graphik, daher das unmotivierte "und" am Fuße. Die übrigen Ausdrucke sind weniger "schön", aber aussagekräftiger. Sie entstanden der Reihe nach mit den Bildschirmformaten 16X64, 24X64, 25X80 und 32X64 Zeichen. Man erkennt, daß nicht bei allen Formaten alle Scan-Zeilen in der Graphik angezeigt werden. Das ist nur bei 32X64 Zeichen der Fall. Hier gehen die Schrägstriche ohne Unterbrechung durch. In den übrigen Formaten werden mehr oder weniger auffällige "Treppen" gebildet. Es ist auch zu sehen, daß ein Graphikpunkt abwechselnd einmal bzw. zweimal ausgedruckt wird (unterschiedlich dick erscheinende Punkte). Jedenfalls hoffe ich, daß der Leser bei seiner Kopie einer Kopie einer Kopie ... es noch erkennen kann.

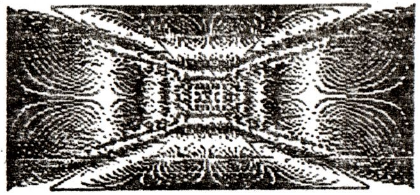
Arnulf Sopp



```
     ;*********************************************************
     ;             Hardcopy der hochauflösenden Graphik
     ;                     des Genie III s
     ;            zentriert für alle Bildschirmformate
     ;
     ;            (C) 1986 by The HACKTORY Arnulf Sopp
     ;*********************************************************

4204                    ORG     4204h           ;im Sektorpuffer

     ;Drucker initialisieren, Bildschirmformat feststellen,
     ;Programm entsprechend modifizieren, Graphik zuschalten
4204  21F042    start   LD      HL,lpinit       ;Codes zur Druckerinit.
4207  CD8F42            CALL    prtstr          ;auf den Drucker ausgeben
420A  3AF137            LD      A,(37f1h)       ;Zeichen pro Zeile
420D  325F42            LD      (chrpl1),A      ;als Zähler ablegen
4210  327842            LD      (chrpl2),A      ;und als Summand
4213  CB67              BIT     4,A             ;80 Zeichen?
4215  2007              JR      NZ,no64         ;falls ja
4217  AF                XOR     A               ;Ak-00 = LSB von 64*8*1.5
4218  32ED42            LD      (nrcols),A      ;LSB Anzahl Dotspalten
421B  CD8F42            CALL    prtstr          ;Randeinst. bei 64 Zeich.
421E  3AF637    no64    LD      A,(37f6h)       ;Zeilen pro Seite
4221  21F837            LD      HL,37f8h        ;Bit 0 = Interlace-Modus
4224  CB46              BIT     0,(HL)          ;32 Zeilen pro Seite?
4226  2801              JR      Z,skipilc       ;falls nein
4228  07                RLCA                    ;Zeilen/Seite verdoppeln
4229  325542    skipilc LD      (linppag),A     ;Zeilenzähler dort ableg.
422C  23                INC     HL              ;HL <- 37F9h
422D  7E                LD      A,(HL)          ;Scan-Zeilen pro Zeichen
422E  D607              SUB     07h             ; + 1 - 8 = 2. Halbzeile
4230  47                LD      B,A             ;Dotzeilen der 2. Halbz.
4231  07                RLCA                    ;verdoppeln
4232  327342            LD      (linpchr),A     ;für Druckerzeilenabstand
4235  AF                XOR     A               ;Akku löschen
4236  37        masklop SCF                     ;Cy auf 1 setzen
4237  1F        codbuf  RRA                     ;Akku: ab Bit 7 auf 1
4238  10FC              DJNZ    masklop         ;AND-Maske errechnen
423A  327142            LD      (andmask),A     ;AND-Operanden anpassen
423D  3AFC37            LD      A,(37fch)       ;rel. Anfang Bildschirm
4240  CBFF              SET     7,A             ;auf Graphik umrechnen
4242  325842            LD      (stgra),A       ;Anfang Graphikspeicher
4245  ED738942          LD      (spbuff),SP     ;Stackpointer retten
4249  F3                DI                      ;keine Störungen
424A  313642            LD      SP,masklop      ;eigener Stack
424D  DBFA              IN      A,(0fah)        ;Systembyte 1 einlesen
424F  F5                PUSH    AF              ;alten Zustand retten
4250  CBDF              SET     3,A             ;Graphik für CPU
4252  D3FA              OUT     (0fah),A        ;zugänglich machen

     ;äußere Schleife: 16, 24, 25 oder 32 Zeilen ausdrucken
4254  0600              LD      B,00h           ;Zähler für n Zeilen
4255            linppag EQU     $-1             ;je nach Bildschirmformat
4256  210000            LD      HL,0000h        ;Adr. der 1. Graphikzeile
4258            stgra   EQU     $-1             ;MSB je nach Bildschirmf.
4259  110008            LD      DE,0800h        ;Summand 1 Dotzl. (2 kB)
425C  C5        pagloop PUSH    BC              ;Zeilenzähler retten
425D  E5                PUSH    HL              ;Graphikzeiger retten
425E  0600              LD      B,00h           ;Zeichen pro Zeile
425F            chrpl1  EQU     $-1             ;je nach Bildschirmformat
4260  C5                PUSH    BC              ;Zeichenzähler retten
4261  3EFF              LD      A,0ffh          ;AND-Maske: alle Dots
4263  0E10              LD      C,10h           ;Zeilenabst. Dots*2/144
4265  CDA342            CALL    oneline         ;1. Halbzeile ausgeben
4268  C1                POP     BC              ;Zeichenzähler
4269  E1                POP     HL              ;Graphikzeiger
426A  E5                PUSH    HL              ;wieder retten
426B  1640              LD      D,40h           ;Smd. f. 16 kB (9. Dotz.)
426D  19                ADD     HL,DE           ;untere Halbzeile
426E  1608              LD      D,08h           ;Summand für 2 kB
4270  3E00              LD      A,00h           ;AND-Maske in den Akku
4271            andmask EQU     $-1             ;je nach Bildschirmformat
4272  0E00              LD      C,00h           ;Druckzeilenabstand
4273            linpchr EQU     $-1             ;je nach Bildschirmformat
4274  CDA342            CALL    oneline         ;2. Halbzeile ausgeben
4277  0E00              LD      C,00h           ;Zeichen pro Zeile
4278            chrpl2  EQU     $-1             ;je nach Bildschirmformat
4279  E1                POP     HL              ;Graphikzeiger
427A  09                ADD     HL,BC           ;nächste Graphikzeile
427B  C1                POP     BC              ;Zeilenzähler restaur.
427C  3A4038            LD      A,(3840h)       ;Tastatur abfragen
427F  CB57              BIT     2,A             ;BREAK gedrückt?
4281  2002              JR      NZ,exit         ;falls ja
4283  10D7              DJNZ    pagloop         ;bis Hardcopy fertig

     ;alten Zustand wiederherstellen, zurück ins Betriebssyst.
4285  F1        exit    POP     AF              ;altes Systembyte 1
4286  D3FA              OUT     (0fah),A        ;restaurieren
4288  310000            LD      SP,0000h        ;Stackpointer restaur.
4289            spbuff  EQU     $-2             ;(bis jetzt hier gepuff.)
428B  FB                EI                      ;INTs wieder zulassen
428C  21F042            LD      HL,lpinit       ;Drucker neu initialis.

     ;einen String ab (HL) auf den Drucker ausgeben
428F  7E        prtstr  LD      A,(HL)          ;Druckzeichen laden
4290  23                INC     HL              ;nächste St. des Strings
4291  FEFF              CP      0ffh            ;String zuende?
4293  C8                RET     Z               ;zurück, falls ja
4294  CD9942            CALL    print           ;nein, ausdrucken
4297  18F6              JR      prtstr          ;nächstes Byte

     ;ein Byte auf den Drucker ausgeben
4299  F5        print   PUSH    AF              ;retten, wird verändert
429A  CDD105    wait    CALL    05d1h           ;Drucker bereit?
429D  20FB              JR      NZ,wait         ;falls noch nicht
429F  F1                POP     AF              ;Code restaurieren
42A0  D3FD              OUT     (0fdh),A        ;auf Drucker ausgeben
42A2  C9                RET

     ;eine Druckzeile initialisieren
42A3  E5        oneline PUSH    HL              ;Graphikzeiger retten
42A4  32CD42            LD      (andoprd),A     ;AND-Maske setzen
42A7  21EA42            LD      HL,inby144      ;Zeilenabstand in Dots
42AA  71                LD      (HL),C          ;im Codestring anpassen
42AB  21E742            LD      HL,lninit       ;Codes für Zeileninitial.
42AE  CD8F42            CALL    prtstr          ;einer Druckzeile ausgeb.
42B1  E1                POP     HL              ;Graphikzeiger restaur.

     ;16-, 24-, 25- o. 32mal 8 Dotspalten auf den Dr. ausgeben
42B2  E5        onechr  PUSH    HL              ;Graphikzeiger retten
42B3  C5                PUSH    BC              ;Zeichenzähler retten
42B4  42                LD      B,D             ;8 Dotzeilen (D = 08)
42B5  C5        oneclp1 PUSH    BC              ;Dotzeilen-Zähler retten
42B6  E5                PUSH    HL              ;Graphikzeiger retten
42B7  7E                LD      A,(HL)          ;Graphikbyte in den Akku
42B8  42                LD      B,D             ;Zähler für 8 Spalten
42B9  213742            LD      HL,codbuf       ;Zeiger auf Graphikpuffer
42BC  07        oneclp2 RLCA                    ;Cy <- je das oberste Bit
42BD  CB16              RL      (HL)            ;in den Puffer rotieren
42BF  23                INC     HL              ;nächste Pufferstelle
42C0  10FA              DJNZ    oneclp2         ;bis 1 Graphikbyte fertig
42C2  E1                POP     HL              ;Graphikzeiger
```

```
42C3  C1                  POP   BC              ;Zeilenzähler
42C4  19                  ADD   HL,DE           ;nächste Graphikzeile
42C5  10EE                DJNZ  oneclp1         ;bis alle Dotzeilen fert.
42C7  213E42              LD    HL,codbuf+7     ;Ende des Graphikpuffers
42CA  42                  LD    B,D             ;Zähler 8 Pufferstellen
42CB  7E          oneclp3 LD    A,(HL)          ;1 Pufferbyte in den Akku
42CC  E600                AND   00h             ;je nach Halbzeile mask.
42CD              andoprd EQU   $-1             ;(Msk. je nach Bildschf.)
42CE  CD9942              CALL  print           ;Graphikbyte ausdrucken
42D1  1803                JR    onecol          ;Akku abwechselnd
42D2              jroffs  EQU   $-1             ;einmal oder zweimal
42D3  CD9942      twocol  CALL  print           ;auf Drucker ausgeben
42D6  3AD242      onecol  LD    A,(jroffs)      ;JR-Distanz umdefinieren
42D9  EE03                XOR   onecol-twocol   ;neuen Offset
42DB  32D242              LD    (jroffs),A      ;im Distanzbyte ablegen
42DE  2B                  DEC   HL              ;nächste Pufferadresse
42DF  10EA                DJNZ  oneclp3         ;bis Puffer ausgedruckt
42E1  C1                  POP   BC              ;Zeichenzähler restaur.
42E2  E1                  POP   HL              ;Graphikzeiger
42E3  23                  INC   HL              ;nächste Graphikadresse
42E4  10CC                DJNZ  onechr          ;bis 1 Halbzeile fertig
42E6  C9                  RET

                  ;ESC-Sequenzen für die HRG und andere Drucker-Features
42E7  0D          lninit  DB    0dh,1bh,'3'     ;CR, Einst. Zeilenabst.
42EA  00          inby144 DB    00h,1bh,'L'     ;und der Drucker-HRG
42ED  C0          nrcols  DB    0c0h,03h,0ffh   ;Anzahl der Dotspalten
42F0  1B          lpinit  DB    1bh,'$',0dh,0ffh  ;Drucker neu initialis.
42F4  1B                  DB    1bh,'M',09h,0ffh  ;li. Rand auf 9. Stelle

4204                      END   start           ;dort Einsprung

00000 Fehler

andmask 4271   andoprd 42CD   chrpl1  425F   chrpl2  4278   codbuf  4237
exit    4285   inby144 42EA   jroffs  42D2   linpchr 4273   linppag 4255
lninit  42E7   lpinit  42F0   masklop 4236   no64    421E   nrcols  42ED
onechr  4282   oneclp1 42B5   oneclp2 42BC   oneclp3 42CB   onecol  42D6
oneline 42A3   pagloop 425C   print   4299   prtstr  428F   skipilc 4229
spbuff  4289   start   4204   stgra   425B   twocol  42D3   wait    429A
```

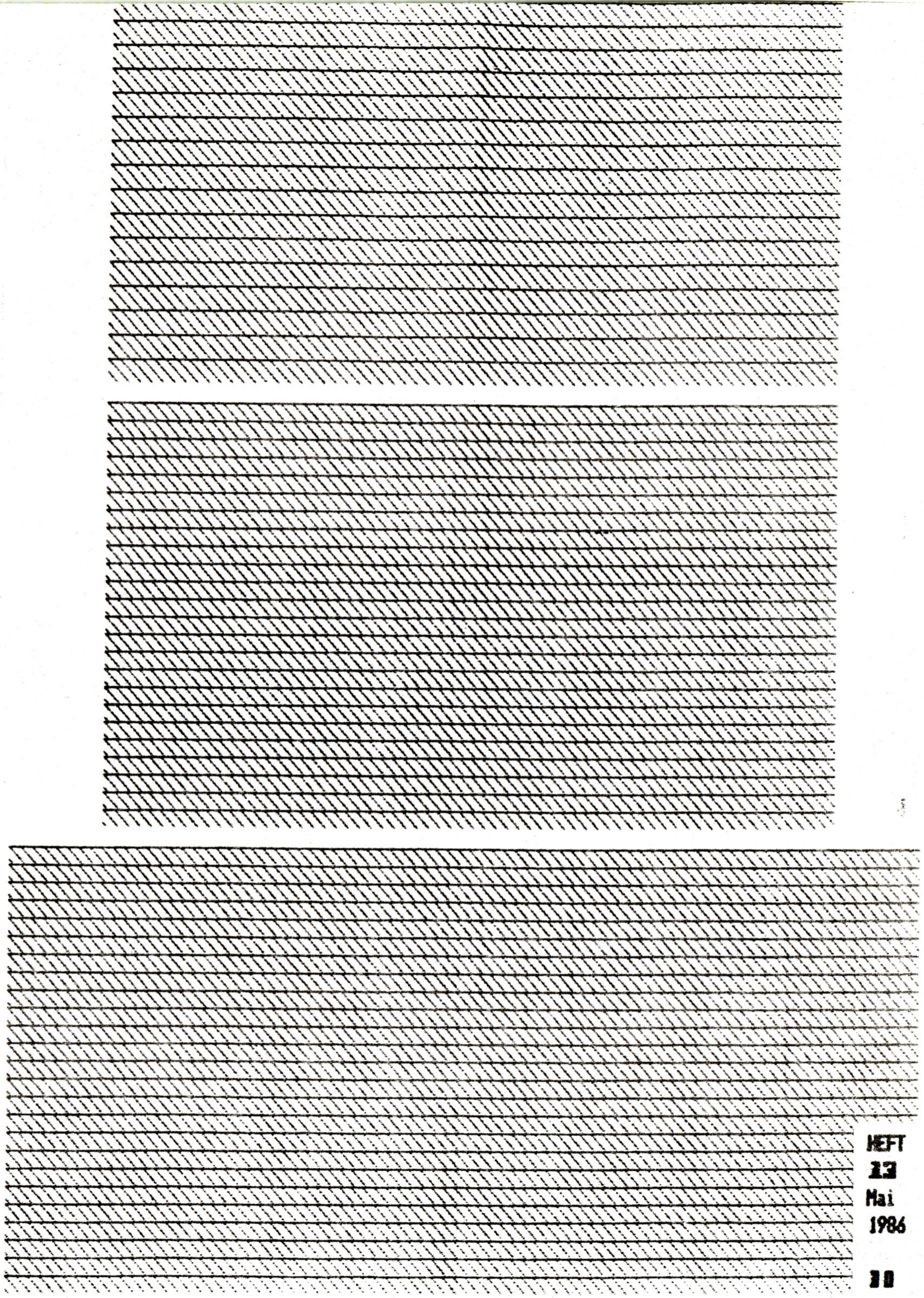
TCS

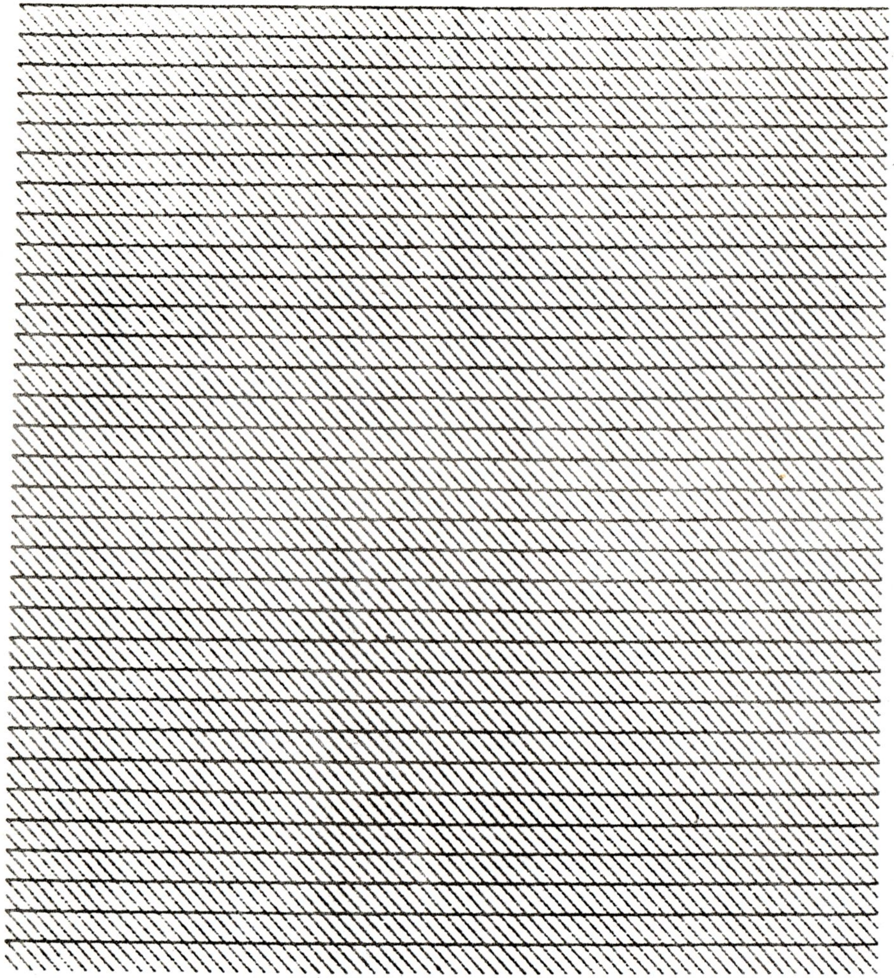


präsentiert den

Genie III S

und





Anmerkung der Redaktion
Die Grafiken wurden für die Clubinfo verkleinert.
Um die Originalgröße zu erhalten sind die hier abgebildeten Grafiken um das 1,4-fache wieder zu vergrößern.

Graphik-JKL für das Genie III s

Kürzlich schrieb ich eine Routine, die mit Shift-JKL die hochauflösende Graphik des G3s zu Papier bringt. Darüber habe ich auch einen Artikel verzapft, der aber bisher (heute ist der 21. 4. 86) noch nicht erschienen ist. Mit dieser Routine konnten die gewöhnlichen ASCII- bzw. die alternativen Zeichen nicht ausgedruckt werden. Der Grund ist folgender: Die CPU kann in den Zeichengenerator zwar die Bitmuster für einen Zeichensatz schreiben, das Character-RAM aber nicht wieder auslesen. Das schafft nur der Bildschirm. Da nun sehr unterschiedliche Zeichensätze geladen werden können, ist es nicht sinnvoll, ganz einfach die Zeichen des Druckers zu diesem Zweck zu mißbrauchen.

Inzwischen ist eine neue Routine fertig, die das packt. Obgleich der Zeichengenerator aus der Sicht der CPU ein WOM (write only memory - nein, das ist hier ausnahmsweise kein Witz) ist, können die Bitmuster der Zeichen dennoch gelesen werden. Sie stehen nämlich zusätzlich im Hauptspeicher in Bank 1 von 0100 bis 10FF. Das Hardcopy-Programm liest sie von dort aus und verknüpft sie XOR mit der Graphik, genau wie es der Bildschirm tut. So entsteht auf dem Papier eine punktgenaue Kopie des Bildschirms. Ein bißchen ist das allerdings gelogen, weil wegen gefälligerer und natürlicherer Proportionen des Ausdrucks jede Dotspalte abwechselnd ein- oder zweimal gedruckt wird.

Die drei Hardcopies sind alle vom selben Bidschirm entstanden. Die am Fuß dieser Seite ist mit dem gewohnten JKL ohne Shift erstellt. Es werden wie üblich nur druckbare Zeichen ausgegeben. Die obere Hardcopy auf der nächsten Seite gibt dieselbe Anzeige wieder, aber mit Shift-JKL. Der schwarze Fladen in der linken Hälfte ist nicht gerade klassische Kunst, aber er ist mit der HRG entstanden, um die Wirkung des Programms zu zeigen. Vor dem unteren Ausdruck wurde die HRG ausgeblendet, aber nicht gelöscht. Das Programm "merkt" das und druckt sie nicht mit. Auf diese Weise kann man z. B. invers drucken, ohne deshalb gleich den Einschaltmüll des Graphikspeichers mit aufs Papier zu kriegen.

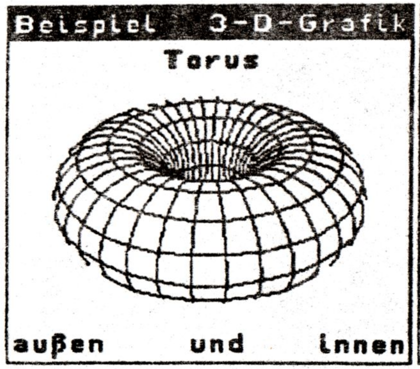
Diese Routine liegt in SYS27/SYS. Wer Interesse hat, sie in sein DOS einzubauen (klappt nur mit dem G3s), kann mir im frankierten und adressierten Rückumschlag eine Systemdiskette (G-DOS 2.4) schicken.

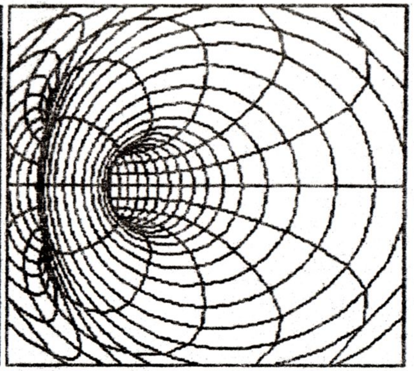
Arnulf Sopp

```
------------------------- GENIE-DOS INFO -------------------------
Tastätur: 4516h            Speicherende (HIMEM): FFFFh
Monitor:  4505h            Kopfzeilen: 00h, Fußzeilen: 00h
Drucker:  058Dh            Druckzeile: 0Eh von 48h
------------------------- ZEICHENSATZ ----------------------------
                                 !"#$%&'()*+,-./0123456789:;<=>?
§ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ^_`abcdefghijklmnopqrstuvwxyzäöüß

------------------------- AKTIVE PORTS ---------------------------
 D2=44  D3=44  D4=00  D5=00  E0=9F  E1=9F  E2=9F  E3=9F  E8=3F
 E9=3F  EA=3F  EB=3F  EC=80  ED=29  EE=03  F1=AA  F7=00  F9=40
 FA=84  FD=3F
Hau rein, is Tango!
```

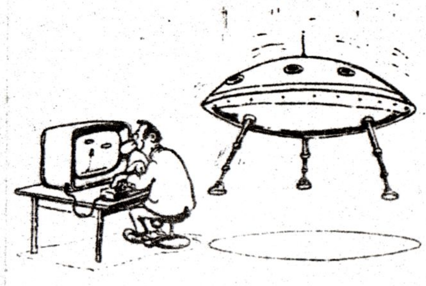
```
------------------------ GENIE-DOS INFO ------------------------
Tastatur: 4516h                 Speicherende (HIMEM): FFFFh
Monitor:  4585h                 Kopfzeilen: 00h, Fußzeilen: 00h
Drucker:  058Dh                 Druckzeile: 38h von 48h
------------------------- ZEICHENSATZ ---------------------------
@ABCDEFGHIJKLMNOPQRSTUVWXYZ↑↓←→ !"#$%&'()*+,-./0123456789:;<=>?
ƏABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ^¤$abcdefghijklmnopqrstuvwxyzäöüß▒
@ABCDEFGHIJKLMNOPQRSTUVWXYZ↑↓←→ !"#$%&'()*+,-./0123456789:;<=>?
ƏABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ^¤$abcdefghijklmnopqrstuvwxyzäöüß▒
------------------------- AKTIVE PORTS --------------------------
 D2=44   D3=44   D4=00   D5=00   E0=9F   E1=9F   E2=9F   E3=9F   E8=3F
 E9=3F   EA=3F   EB=3F   EC=80   ED=2E   EE=10   EF=45   F1=AA   F7=C0
 F9=40   FA=86   FD=3F
Hau rein, is Tango!
```


```
------------------------ GENIE-DOS INFO ------------------------
Tastatur: 4516h                 Speicherende (HIMEM): FFFFh
Monitor:  4585h                 Kopfzeilen: 00h, Fußzeilen: 00h
Drucker:  058Dh                 Druckzeile: 38h von 48h
------------------------- ZEICHENSATZ ---------------------------
@ABCDEFGHIJKLMNOPQRSTUVWXYZ↑↓←→ !"#$%&'()*+,-./0123456789:;<=>?
ƏABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ^¤$abcdefghijklmnopqrstuvwxyzäöüß▒
@ABCDEFGHIJKLMNOPQRSTUVWXYZ↑↓←→ !"#$%&'()*+,-./0123456789:;<=>?
ƏABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜ^¤$abcdefghijklmnopqrstuvwxyzäöüß▒
------------------------- AKTIVE PORTS --------------------------
 D2=44   D3=44   D4=00   D5=00   E0=9F   E1=9F   E2=9F   E3=9F   E8=3F
 E9=3F   EA=3F   EB=3F   EC=80   ED=2E   EE=10   EF=45   F1=AA   F7=C0
 F9=40   FA=84   FD=3F
Hau rein, is Tango!
```

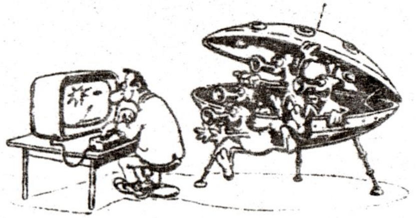





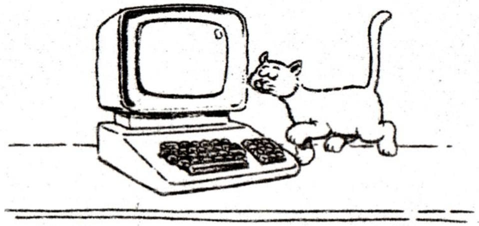
(C) Torus by TCS, Hardcopies by Sopp



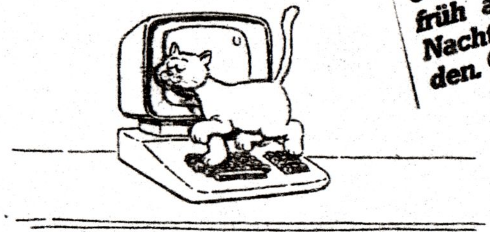
„Das Schönste am Berühmtsein ist: Wenn man Leute langweilt, glauben sie, es ist ihre Schuld."



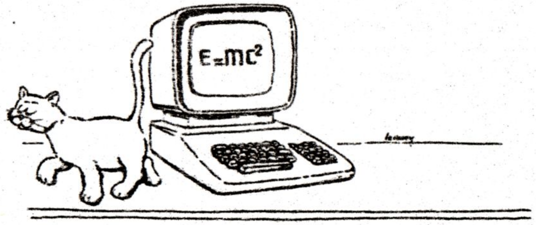
„Der Psychoanalytiker ist ein Beichtvater, den es gelüstet, auch die Sünden der Väter abzuhören."



„Um es im Leben zu etwas zu bringen, muß man früh aufstehen, bis in die Nacht arbeiten – und Öl finden."



„Das absolut Negative an den schnellen Flugzeugen ist, daß es jetzt keine entfernten Verwandten mehr gibt."



Graphik-CLS für das Genie IIIs

Schon im alten H-DOS für das Genie I wurde der Library-Befehl CLS so verbogen, daß auch die HRG 1b damit gelöscht werden konnte. Das ist jetzt auch in H-DOS 2.4 für das Genie IIIs der Fall. Da die beiden Graphik-Systeme jedoch nicht gleich sind, mußte die Bearbeitungsroutine völlig neu geschrieben werden. Sie residiert zusammen mit der SYS-Routine (auch in diesem Info abgedruckt?) im letzten Sektor von SYS4/SYS, wo noch genügend Platz frei ist.

Beim Einsprung wird zunächst geprüft, ob der Requestcode im Akku 86h oder A6h für den Befehl SYS bzw. CLS beträgt. Ist das nicht der Fall, muß es sich um eine Anforderung für eine andere Routine in SYS4/SYS handeln. In diesem Falle wird nach 4D00h weitergesprungen, dem alten Entrypoint von SYS4. Handelt es sich um den umgemodelten Befehl CLS, dann war A6h im Akku, und es geht los:

Mit einem CALL nach 4CD5h wird überprüft, ob nach CLS noch weitere Parameter folgen. Falls nein, war es der alte Befehl, wie er schon immer möglich war. Dann wird HL als Zeiger auf die Bildschirmsteuerzeichen 1Ch und 1Fh geladen und diese ausgegeben. Das war dann schon alles. Folgt aber etwas nach, dann wird geprüft, welche(r) von drei möglichen Parametern mit eingegeben wurde(n).

Möglich sind V (Video löschen), 0 (Graphikseite 0 löschen) und 1 (Graphikseite 1 löschen). Wenn man gleich alles leerfegen will, kann man auch CLS,0,1,V eingeben, wobei die Reihenfolge und Anzahl der Parameter egal ist.

Im Kern passiert folgendes: Durch einen Output auf den Port FAh (Systembyte 1) mit gesetztem Bit 3 wird ab 8000h das Anwender-RAM gegen den Graphikspeicher ausgetauscht. Da er somit für die CPU direkt zugänglich ist, kann er mit dem einfachen LDIR-Konstrukt ab Zeile 37 im Listing gemäht werden. Bei der HRG 1b war das komplizierter.

Die Anwahl der gewünschten Graphikseite geschieht kurz davor: Wenn das Bit 5 des Systembytes 0 (Port F9h) gesetzt ist, wird die Graphikseite 1 zum Schreiben und Lesen freigegeben. Bei rückgesetztem Bit 5 ist es die Seite 0. Um im Kernprogramm auf beide Fälle vorbereitet zu sein, wird in Zeile 15 (Label clspar) zunächst das Register C mit 00h geladen. Bei Seite 1 wird in C das Bit 5 (s. o.) gesetzt, bei Seite 0 wird dieser Befehl übersprungen. Wenn nun in Zeile 30 der Akku-Inhalt (alter Zustand von Systembyte 0) mit C oderiert wird, ist die entsprechende Seite selektiert.

Wenn sie schließlich gelöscht ist, wird beim Label getpar (Zeile 47) nach weiteren Parametern geforscht. Gibt es keine mehr, geht's zurück ins Betriebssystem. Andernfalls geht das alte Spiel bei clspar von vorne los.

Um das Löschen der Graphik auch unter BASIC zu ermöglichen, muß dafür Sorge getragen werden, daß der Stack nichts zu tun hat, während auf dem Port FAh das Bit 5 gesetzt ist. BASIC unterhält den Stack ziemlich hoch im Speicher, also ausgerechnet dort, wo vorübergehend die Graphik eingeblendet ist. Würden nun in Zeile 34 die Registerpaare HL und DE mit PUSH gerettet werden, dann überschriebe der Löschvorgang den Stack mit Nullen. Beim anschließenden POP stünde demnach 0000h in beiden Registerpaaren. Der Befehl EXX tauscht stattdessen die CPU-Register gegen den alternativen Satz aus, so daß hier keine Gefahr droht.

Die Sektordumps, in denen wie üblich die geänderten Codes unterstrichen sind, zeigen der Reihe nach folgendes:

1. Sektor 02 von SYS1/SYS, Änderung am Befehl CLS
2. Sektor 04 von SYS4/SYS, Programm und geänderter Einsprung

Zusätzlich kann die ehemalige CLS-Routine in SYS1/SYS jetzt überschrieben werden. Sie ist jedoch nur sechs Bytes lang, so daß kaum etwas machbar ist. Aber die Codes 1Ch-1Fh-03h im letzten Sektor von SYS1/SYS sind nun frei und können für eine etwas längere, vielleicht lustigere DOS-Ready-Meldung ausgenutzt werden.

Arnulf Sopp

```
H-DOS-Diskdaten-Editor Vers. 2.4
(C) 1986 by TCS / The HACKTORY

Datei: SYS1/SYS

drv: 0              frs: 0002h        drs: 0598h
000200: D54C 20A9 CB59 2802 0102 004F E3E5 79E6  L   Y(     O  y
000210: 0728 0EE5 21BC 5123 2323 3D20 FACD 2A4F   (  ! Q###=  . *O
000220: E179 01D3 49C5 CB7F C806 0021 0042 CB77  y  I       ! B w
000230: CA24 44C3 2044 D5C5 011C 091A FE3A 280A  $D  D         :(
000240: FE2F 3806 281B 0D13 10F1 2323 E5EB 0600  /8 (       ##
000250: 0954 5D2B 1313 13ED BBE1 0E03 EDB8 3E2F  TÜ+              >/
000260: 12C1 D1C9 4149 4B80 5300 4150 5045 4E44     AIK S APPEND
000270: C06B 0041 5454 5249 42B5 E98B 4155 544F  h ATTRIB    AUTO
000280: 80DC 0042 3286 EB00 424C 81E5 0042 4F4F     B2    BL    BOO
000290: 548A EB10 4252 4541 4B85 E500 434C 5380 T    BREAK    CLS
0002A0: A600 434F 4E54 C5EB 0043 4F50 59C0 4800   CONT    COPY H
0002B0: 4352 4541 5445 82F0 4044 4154 554D 8BE9 CREATE  $DATUM
0002C0: 0044 4445 81F1 0044 4952 802A 0044 4953  DDE   DIR * DIS
0002D0: 4B83 FF00 444F C3EB 8A44 5282 FE00 4455 K    DO    DR    DU
0002E0: 4D50 87E9 C845 87F0 0046 4F52 4D88 FE00 MP    E    FORM
0002F0: 4652 4545 804A 0046 2380 FB00 4849 4D45 FREE J F#    HIME
```

```
H-DOS-Diskdaten-Editor Vers. 2.4
(C) 1986 by TCS / The HACKTORY

Datei: SYS4/SYS

drv: 0              frs: 0004h        drs: 05C7h
000400: E544 6973 6B65 7474 65EE 4174 7472 6962  Diskette Attrib
000410: 01EA 0051 75F4 7363 686C 6563 6874 65F2    Qu schlechte
000420: 4B65 696E EE1B 1B1B 6665 686C 65F2 6265 Kein    fehle be
000430: E975 6E7A 756C 6165 7373 6967 E508 F3FE  unzulaessig
000440: B628 56FE A6C2 004D CDD5 4C20 0921 3F51  (V     M  L  !?Q
000450: C367 441C 1F03 0E00 7ED6 302B 103D 280B  gD     B O( =(
000460: FE2E 2031 E5CD 3951 E118 23CB E9DB F957  % 1  9Q   #     W
000470: B1D3 F9DB FA5F D9CB DFD3 FA21 0080 1101             !
000480: 8001 FF7F 75ED B0D9 7AD3 F97B D3FA 23CD     u   z   ä   #
000490: D54C CB18 C13E AF18 225E 2356 23CD D54C  L    >   "^#V#   L
0004A0: 3819 E506 03D5 21AB 515E 2356 D723 E34F 8       ! Q^#V # O
0004B0: DF79 E328 0410 F23E 34E1 E1C2 0944 EF49  v  (    >4     D I
0004C0: 443D 494F A545 435D 0000 0000 0000 0000 D=IO HCO
0004D0: 0000 0000 0000 0000 0000 0000 0000 0000
0004E0: 0000 0000 0000 0000 0000 0000 0000 0000
0004F0: 0000 0000 0000 0000 0000 0000 0202 2B51                 +Q
```

```
                      ;Systemerweiterg. am Ende von SYS4/SYS für den SYS-Befehl
                      ;und die Varianten des CLS-Befehls

512B                          ORG   512bh         ;ab hier Platz in SYS4
512B   FEB6           sys4    CP    B6h           ;SYS-Befehl?
512D   2856                   JR    Z,sys         ;falls ja
512F   FEA6                   CP    0A6h          ;CLS-Befehl?
5131   C2004D                 JP    NZ,4D00h      ;falls nein

5134   CDD54C                 CALL  4cd5h         ;Trennzeichen erkennen
5137   2009                   JR    NZ,clspar     ;falls Parameter folgen
5139   213F51         clsv    LD    HL,cls        ;nein, nur ASCII-Bildsch.
513C   C36744                 JP    4467h         ;diesen löschen
513F   1C             cls     DB    1ch,1fh,03h   ;Codes für CLS
5142   0E00           clspar  LD    C,00h         ;Anfangswert OR-Operand
5144   7E                     LD    A,(HL)        ;was folgt nach CLS?
5145   D630                   SUB   '0'           ;Graphikseite 0 löschen?
5147   2810                   JR    Z,clsgra0     ;falls ja
5149   3D                     DEC   A             ;Graphikseite 1 löschen?
514A   2808                   JR    Z,clsgra1     ;falls ja
514C   FE25                   CP    '%'           ;ASCII-B.? (ehem. "V")
514E   2031                   JR    NZ,error1     ;falls nichts von alledem
5150   E5                     PUSH  HL            ;retten, wird verändert
5151   CD3951                 CALL  clsv          ;ASCII-Bildschirm löschen
5154   E1                     POP   HL            ;restaurieren
5155   1823                   JR    getpar        ;nächster Parameter
5157   CBE9           clsgra1 SET   5,C           ;Sysbyte 0, Bit 5: Gr. 1
5159   DBF9           clsgra0 IN    A,(0f9h)      ;Systembyte 0 lesen
515B   57                     LD    D,A           ;altes Systembyte retten
515C   B1                     OR    C             ;Graphikseite selektieren
515D   D3F9                   OUT   (0f9h),A      ;Systembyte neu ausgeben
515F   DBFA                   IN    A,(0fah)      ;Systembyte 1
5161   5F                     LD    E,A           ;retten
5162   D9                     EXX                 ;Reg. ohne Stack retten
5163   CBDF                   SET   3,A           ;Graphikseite freigeben
5165   D3FA                   OUT   (0fah),A      ;Systembyte neu ausgeben
5167   210080                 LD    HL,8000h      ;Anfang der Graphik
516A   110180                 LD    DE,8001h      ;nächste Stelle
516D   01FF7F                 LD    BC;7fffh      ;Länge des Gr.-speichers
5170   75                     LD    (HL),L        ;Stelle ausnullen
5171   EDB0                   LDIR                ;dto. kpl. Graphikspeich.
5173   D9                     EXX                 ;Register restaurieren
5174   7A                     LD    A,D           ;altes Systembyte 0
5175   D3F9                   OUT   (0f9h),A      ;restaurieren
5177   7B                     LD    A,E           ;dto. Systembyte 1
5178   D3FA                   OUT   (0fah),A
517A   23             getpar  INC   HL            ;weiter im Befehl
517B   CDD54C                 CALL  4cd5h         ;Trennzeichen erkennen
517E   C8                     RET   Z             ;falls Befehl zuende
517F   18C1                   JR    clspar        ;sonst weitere Parameter
5181   3EAF           error1  LD    A,0afh        ;"falsche Parameter"
5183   1822                   JR    error2        ;anzeigen und zurück

                      ;Hier folgt die Bearbeitungsroutine des
                      ;Library-Befehls SYS (mit LIST OFF ausgespart).

51A7   C20944         error2  JP    NZ,4409h      ;Fehleranzeige und zurück

512B                          END   sys4          ;dort neuer Einsprung

00000 Fehler
```


BASCOM-Kurzanleitung
(von KaJot)

1) Im BASIC-Programm verbotene Statements:
AUTO / CLEAR (im CMD-File nicht benötigt) / CLOAD / CMD / COMMON / CSAVE / CONT / DELETE / EDIT / ERASE / INKEY / LF (=↓) / LIST / LLIST / LOAD / MERGE / NEW / RENUM / SAVE

2) Programm mit Extension /BAS als ASCII-File speichern;
3) Eingabe im DOS:
BASCOM Filename1:Lw1,Filename2:Lw2=Filename3:Lw3 <-Switches>
Filename1 ==> erhält Extension "/REL" und wird Ausgangsfile für das mit "L80" zu erstellende CMD-File;
Filename2 ==> erhält Extension "/LST" und besteht aus Basic-Text + Assembler-Code;
Filename3 = Source-File = zu kompilierendes BASIC-Programm.
Die 3 Filenamen dürfen gleich sein. Die Angabe der/des Laufwerke(s) empfiehlt sich. Filename 1 und/oder 2 können fortgelassen werden. Werden beide Filenamen fortgelassen, so findet nur der Syntax-Check des Source-Files statt.

Switches (jeweils an den letzten Filenamen im Befehl anzufügen)

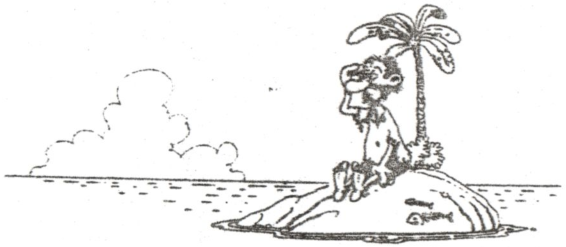
-E wenn das Programm "ON ERROR GOTO" enthält und nur "RESUME Zeilen-Nr."; sonst:
-X siehe -E. Das Programm enthält außer "ON ERROR GOTO" auch "RESUME", RESUME NEXT" oder "RESUMEØ";
-N im LST-File soll kein (NO) Object-Code eingeschrieben werden;
-D ruft beim Programmlauf ggf. DEBUG auf. Versieht das REL-File mit Zeilennummern, die bei Fehlern während des Programmlaufs ("Runtime Errors") angezeigt werden. -D muß gesetzt werden, wenn TRON/TROFF benutzt werden soll.
-S veranlaßt den Compiler, auch Strings (in Anführungszeichen) mit mehr als 4 Zeichen in das Binary-File zu schreiben (spart Speicherplatz).

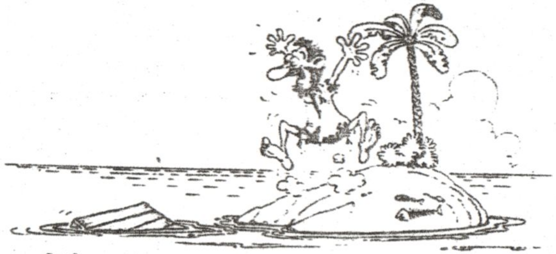
4) Eingabe im DOS:
L80 Filename3:Lw1,Filename1:Lw2-N-E <-weitere "Switches">
-N speichert das compilierte Programm als /CMD-File, das im DOS ohne Extension gestartet werden kann; -N muß dem Filenamen (ggf. einschl. Lw-Nr.) unmittelbar folgen und unmittelbar oder mittelbar von einem -E- oder -G-Switch gefolgt werden;
-E übergibt die Kontrolle nach Ausführung wieder dem DOS;
-G das Programm wird sofort gestartet
-R "lädt" das gerade geladene Programm wieder "aus" (falls ein Fehler gemacht wurde). - Kann während des Ladevorganges benutzt werden!

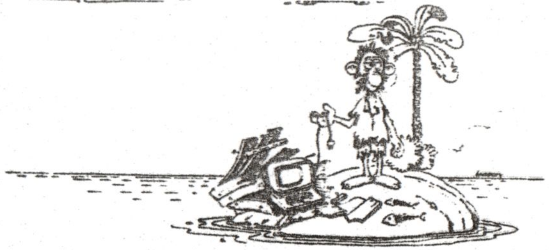
39 Diejenigen, die an der Hauptversammlung anwesend waren, haben sicher den "Selbstbau-Rechner" vom Bernd gesehen.
Wir haben jetzt vor, uns auch einen solchen zu bauen und suchen dazu noch weitere Mitstreiter.
Unsere Vorstellungen dazu liegen darin, daß der Selbstbau voll kompatibel zu dem TRS-80 M I und transportabel ist. Weiterhin sollte er CP/M fähig und ausbaubar für bestimmte Anwendungen sein. Der Einbau verschiedener eigener technischer Raffinessen ist dann natürlich auf einfachstem Wege machbar. Dies wird unter anderem durch den ECB-Bus sichergestellt. Für diesen genormten Bus gibt es ja schon eine Menge Erweiterungen.
Durch einfaches Tauschen der europakartengroßen Platinen besteht dann auch später die Möglichkeit offen, das System der Technik entsprechent aufzurüsten.

Wie es im Moment aussieht sind wir mit dem Bernd zu dritt. Wer sich für die Sache interessiert, Vorschläge dazu hat und/oder einfach selbst einmal eine Kiste zusammenlöten möchte melde sich bitte beim Bernd, dem Ulrich oder mir.

Jens Neueder







Amerikanisch: Are you a turtle?
Bist du eine Schildkröte?
Warum sagst du nichts?

Arabisch: Mithel et-tin sil farde.
Zerdrückt wie eine Feige im Sack.
Mir geht's scheußlich.

Chinesisch: T'ieh shu k'ai hua lü tzu nien.
Im Jahr des Affen, wenn Eisenbäume Blüten tragen.
Wenn Ostern und Pfingsten auf einen Tag fallen.

LWT für Genie I und TRS-80 Modell I

In der 10. Ausgabe des Club infos beschrieb Hartmut, wie man die LWT-Routine des G-DOS an das New-DOS anpaßt. Er informierte , daß einige Zaps zur Anpassung der Anzeige an den TRS-80 nötig seien.

Nach meiner Meinung irrt sich Hartmut hier, da die Änderungen nicht unbedingt für den TRS-80 nötig sind. Nach meinem Wissen liegt der Sachverhalt etwas anders.

Bevor jemand die LWT-Routine ändert, dann möglicherweise auf eine falsch Anzeige hereinfällt und seine Laufwerke anstatt einzustellen verstellt, sollte er prüfen, welcher Quarz in seinem Rechner eingebaut ist.

Tandy und auch EACA (Video Genie) haben ursprünglich einen Quarz von 10.644 MHz eingebaut. Aus dieser Quarzfrequenz erzeugt sich der Rechner alle benötigten Taktfrequenzen, z. B. CPU-Takt und Bildsynchronsignale. Bei dieser Quarzfrequenz ergibt sich ein Vertikalsynchronsignal von 50,77 Hz. Da aber unsere Netzfrequenz 50 Hz beträgt, kommt es auf dem Monitor häufig zu Bildzittern oder Bildschwankungen.

Da in den USA die Netzfrequenz 60 Hz beträgt, war der benutzte Quarz dort richtig. Durch eine andere Teilung wurde die Quarzfrequenz auf 60 Hz gebracht und war somit synchron mit der Netzfrequenz.

Anfangs verkaufte man in Europa beide Rechner mit dem Original USA-Quarz. Da es immer wieder zu Bildschwankungen kam, änderte man den Quarz auf 10.483 MHz, um die Vertikal-Synchronfrequenz auf 50 Hz zu bringen. Um also die LWT-Routine richtig anzupassen, sollte man auf dem Quarz im Rechner nachsehen, welche Frequenz auf diesem steht.

Im übrigen ist natürlich auch die CPU-Taktfrequenz bei beiden Quarzen unterschiedlich:

10.644 MHz Quarzfrequenz ergibt 1,77 MHz CPU-Takt
10.483 MHz Quarzfrequenz ergibt 1,75 MHz CPU-Takt.

Ein Rechner mit dem neuen Quarz läuft daher langsamer als ein Rechner mit altem Quarz. Dies sollte aber wohl nur Assemblerprogrammierer interessieren, die Realtimeprogramme schreiben (Beispiel: möglicherweise das c't Supertapeprogramm).

Bei der LWT-Routine wird auf dem Bildschirm ein Kalibrierwert angezeigt. Dieser sollte für den 10.644 MHz Quarz 5928H und für den 10.483 MHz Quarz 58C8H betragen.

Wenn der Rechner einen neuen Quarz eingebaut hat, sollte man auch sehr vorsichtig mit Floppy-Prüf-Programmen "Made in USA" sein, da sich diese sicher auf den USA-Quarz beziehen. Dies kann zu Fehleinstellungen bzw. Fehldiagnosen führen.

So, und nun viel Spaß beim richtigen Anpassen der LWT-Routine und beim Laufwerkabgleich.

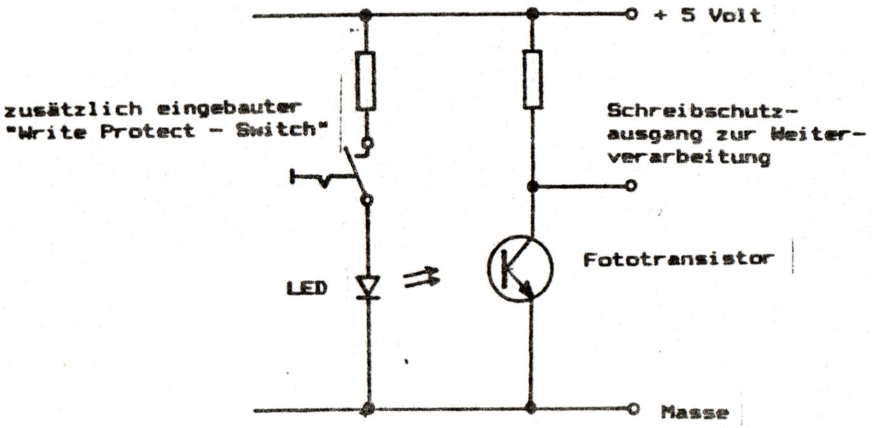
Bernd Retzlaff

Floppytrick's

"Kaum der Rede Wert", so stufte ich eine kleine Modifikation ein, die ich an meinen Diskettenlaufwerken gemacht habe. Erst beim Clubtreffen wurde mir klar, daß die Sache so einfach ist, daß man sogar Computerfreaks wie Arnulf damit verblüffen kann. Die Rede ist von einem sog. 'Write Protect - Switch', mit dem man dem Drive jederzeit signalisieren kann, ob man eine Disk beschreibbar haben möchte oder nicht.

Normalerweise klebt man zu diesem Zweck ja die Schreibschutzkerbe der Diskette zu. Diese Methode ist jedoch bei häufigem Gebrauch sehr lästig. "Kleber drauf, Kleber ab, Kleber drauf!" ist nicht jedermanns Sache. Wenn man sich die Funktion des Schreibschutzes jedoch einmal genauer ansieht, erkennt man schnell eine einfache Lösungsmöglichkeit.

Neuere Laufwerke klären die Frage "Schreibschutzkerbe abgeklebt oder nicht?" meist mittels einer Lichtschranke. Ist die Kerbe offen, kann das Licht der LED den Fototransistor erreichen und signalisiert so "kein Schreibschutz!". Eine zugeklebte Kerbe läßt kein Licht zum Fototransistor und die Scheibe kann nicht beschrieben werden.



Dieses einfache Prinzipschaltbild veranschaulicht die Funktion der heute üblichen Schreibschutzerkennung und den Einbau des zusätzlichen Write Protect - Schalters

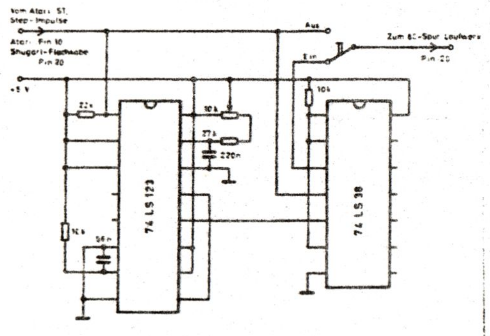
Will man diese Schaltungen überlisten, braucht man nichts weiter zu tun als die Leuchtdiode auszuschalten. Dann fällt kein Licht mehr auf den Transistor und die Floppy ist vor Schreibzugriffen sicher!

Größter Vorteil des kleinen Eingriffs ist der problemlose und sekundenschnelle Wechsel zwischen Write Protect- und Write Enable-Zustand. Ein kleiner Nebeneffekt tritt bei sehr modernen Laufwerken (z.B. Philips 3134), die beim Einlegen der Diskette zum besseren Einfädeln des Mitnehmerkonus kurzzeitig anlaufen, auf. Hier läuft bei jedem Umschalten des Schreibschutzschalters kurz der Motor an, da das Laufwerk ja der Meinung ist, gerade eine Floppy eingelegt zu bekommen. Fehler beim Betrieb treten jedoch dadurch nicht auf.

Einen weiteren Floppytrick habe ich in der c't 5/86 gefunden. Dort wurde eine kleine Schaltung vorgestellt (siehe unten), die es ermöglicht 80-Track- als 40-Track-Laufwerke laufen zu lassen. Das gab es zwar schonmal im Info (Nr. 7, Seite 37 TEAC 80/40-Spur-Umschaltung von Ulrich Böckling), da jedoch nur für TEAC-Laufwerke. Die unten vorgestellte Schaltung funktioniert mit allen 80-Spur-Laufwerken.

Die abgebildete kleine Zusatzschaltung mit zwei billigen TTL-Chips macht jedes beliebige 80-Spur-Laufwerk umschaltbar auf den 40-Spur-Betrieb. Dies funktioniert ebenso wie die Umschaltung bei den ab Werk dafür vorgesehenen Drives: Die Schaltung erzeugt nach jedem Step-Impuls im 3-ms-Abstand einen weiteren, so daß der Schreib-/Lesekopf jeweils um zwei Spuren weitergerückt wird. Sie wird in die Leitung 20 des Flachkabel-Anschlusses (Schrittsignal) zwischen Rechner und Laufwerk eingeschleift. Mit dem Schalter kann man dann zwischen 40- und 80-Spur-Betrieb wählen. Natürlich darf auch ein so umgerüstetes Laufwerk die Step-Impulse nur alle 6 ms erhalten.

Wir haben die Zusatzschaltung auf einer kleinen Lochrasterplatine in das Laufwerkgehäuse mit eingebaut. Masse und +5-V-Zuleitung wurden an den Stromversorgungsstecker des Laufwerks angeschlossen (Vorsicht, +5V liegt an Pin 4, Pin 1 führt +12V, im Zweifel messen). Mit dem Trimmpoti muß die Verzögerung so eingestellt werden, daß die zusätzlich erzeugten Impulse ungefähr in der Mitte zwischen den Originalimpulsen liegen. Eine geringfügige Abweichung macht allerdings nichts aus, so daß man, falls geeignete Meßmöglichkeiten fehlen, mit dem in Mittelstellung gebrachten Trimmer und etwas Glück auch schon den gewünschten Erfolg erzielen kann. Notfalls können Sie die geeignete Einstellung durch Ausprobieren ermitteln. Im c't-Labor wurde die Schaltung mit einem Philips-Laufwerk X 3134 erprobt und funktionierte einwandfrei. ct



Impulsverdopplerschaltung zur Umschaltung von 80-Spur-Laufwerken auf 40-Spur-Betrieb

Hat man zwei 80'er im System und will beide gleichzeitig umschalten, braucht man diese Schaltung nur einmal. Man schleift sie dann vor den Laufwerken in die Leitung 20 der Shugart-Schnittstelle ein. Will man im gemischten Betrieb fahren, muß man die Schaltung für jedes Laufwerk einzeln aufbauen und vor den Drives einschleifen. Hat man zusätzlich noch ein 40'er Laufwerk im System, muß man dafür sorgen, daß dieses nicht auch mit den doppelten Stepimpulsen versorgt wird. Man muß also auch hier die Schaltung für jedes verfügbare 80-Spur-Laufwerk aufbauen und einzeln vor dem jeweiligen Laufwerk einschleifen.

Solltet ihr Fragen in Bezug auf Laufwerke, deren Anschluß und die richtige Jumperbestückung (die ja oft sehr verwirrend ist) haben, könnt ihr euch an mich wenden. Auch wenn jemand Schaltunterlagen zu Philips 3134- oder BASF 6106-Drives benötigt, kann ich weiterhelfen.

Viel Spaß beim Basteln wünscht euch
euer

Hartmut Obermann

A Quick Draw

by John D. Wolfskill

The Model 4 has its virtues, but graphics power isn't one of them. Its Basic lacks the built-in drawing commands that make graphics programming on other machines, like the Model 1000 or 2000, so convenient. My machine-language subroutine, Drawstring, adds graphics commands to Disk Basic and gives you pixel-level control of the Model 4's display.

While you can't use Drawstring to create perfect circles in 16 blazing colors, you can draw or erase in eight directions, jump over pixels, and build animated sequences. The subroutine is extremely fast. You use string variables to pass drawing commands to the subroutine; a single string can send up to 254 sequential commands without returning to Basic.

Preliminaries

Drawstring must be installed in memory before your Basic programs can call it. To create a Draw/CMD file, go into Basic and run Program Listing 1. Then exit to TRSDOS and type in:

DUMP DRAW/CMD (START=X'FE9A',END=X'FFFF',TRA=X'FE9A')

Thereafter, you can install Drawstring by typing in DRAW from TRSDOS Ready.

If you have filters or modules that normally reside in high memory, use the TRSDOS Memory command to set the system HIGH$ marker below FE9A hexadecimal. If you find you're getting "illegal function call" error messages, try using a clean copy of TRSDOS 6.x.

With the subroutine installed, you can run Program Listings 2 and 3 for a demonstration of some of Drawstring's capabilities. Listing 2 shows how rapidly the program draws and erases patterns and inverts video from white-on-black to black-on-white (see Photo 1). Listing 3 displays a blinking diamond character that you can move around the screen without destroying the image underneath (see Photo 2).

Writer's Workshop

Drawstring is a programmer's tool. Once you understand how the commands work, you can begin to experiment with the subroutine in your own Basic programs.

You must include the statements:

CLEAR,65177I:DEFINT X,Y:DRAW=&HFE9A

in the first line of any program that will call the subroutine (see lines 30 and 50 of Listings 2 and 3, respectively). Note that variables must be defined as integers.

First, select a screen starting point. Set the variables X and Y equal to the row (zero to 71) and column (zero to 159) pixel coordinates of the location you want. For example:

X=0:Y=0

tells the subroutine to start drawing at the upper-left corner of the screen.

Define a string variable that will pass a sequence of commands to the subroutine. Table 1 lists the commands that Drawstring recognizes. Letter commands control the direction of drawing; follow them with a number from 1–255 to specify the distance in pixels. For example:

X=0:Y=0:BORDER$="D71R159U71L159$"

tells Drawstring to draw a border around the edge of the display, starting in the upper-left corner and moving counterclockwise: down 71 pixels, right 159, up 71, left 159.

The B command moves the pixel pointer without drawing. The * command complements (inverts) pixels while drawing, making black pixels white and vice versa. Use % to toggle between the complement and draw options within a single command sequence.

Make sure that you enter alphanumeric string commands as uppercase characters. If you issue a direction command without a number, the pixel pointer will move just one step in the direction you indicate.

The dollar sign ($) must be the last character in each string, since Drawstring uses it as an end-of-string delimiter. This helps the subroutine keep track of the data in case the computer shuffles it to high memory for processing.

Now, place a call to the subroutine, listing the appropriate variables as parameters. For the sample screen border above, the statement would look like this:

CALL DRAW (X,Y,BORDER$)

The subroutine picks up the current value of the variables from Basic and feeds the information in the strings to the pixel processor. The program returns control to Basic when it encounters the dollar sign ($) at the end of your command sequence.

A Few Tips

Since Drawstring disregards any character not listed in Table 1, stray letters or numbers won't crash your program. They might, however, produce unexpected results. For instance:

FILL$=FOUR SCORE AND SEVEN YEARS AGO$"

will develop its own unique graphics signature.

Drawstring doesn't check your number entries to make sure they stay within the allowed range, because TRSDOS provides its own error check by refusing to read or write to any bytes outside video memory. The subroutine accepts integer pixel values from 1–255. Try experimenting with numbers outside the normal coordinate range. You can create some interesting graphics by using numbers that are large enough to make the processor wrap around and continue drawing at the opposite end of the screen.

You can load and save data strings as disk files in the normal manner, and you can manipulate the data using any of Basic's string-handling functions. For example, run this economical two-liner that uses the INKEY$ function:

5 DRAW=&HFE9A:X=0:Y=0
10 X$=INKEY$:X$=X$+"$":CALL DRAW (X,Y,X$):GOTO 10

It lets you draw in any of eight directions by pressing the letter keys listed in Table 1.

The easiest way to draw complex figures is to construct a paper graph in row/column block form. Divide each video byte into six individual pixel cells. Remember to size the cells using the proper ratio; each horizontal pair of pixels within a byte is a different vertical size.

When your graph picture is complete, translate the image into string commands. The pixel-counting algorithm is pretty straightforward. Examine Listing 2 and you should quickly catch on to the logical pixel-counting progression.

Moving Pictures

One of Drawstring's powerful features is its ability to move objects around the display without destroying the background. You can draw (%) a pattern on the screen, then redraw the string starting at the same location, this time using a leading complement sign (*) to erase it. You can also turn complementing on and off from within your Basic program so you don't have to define the string twice. Table 2 provides the POKE address and values for controlling the draw/complement toggle.

The complement option inverts pixels as it proceeds across the display. When it encounters a set bit (on), it resets the bit (off) and vice versa. Combining this action with directional movement lets an object travel across the screen without altering the background. Listing 3 demonstrates this technique.

When you design animated screens that use the same data to draw and erase an object, keep in mind that you cannot backtrack over a pixel that has been set. Retracing your steps will turn off the pixel that you might have just turned on. If you get cornered while plotting a figure, it's a simple matter to jump over a set pixel using the B command.

End Notes

I expanded the string data in Listing 2 to make it easier for you to understand how Drawstring works. You can increase the program's drawing speed by omitting the spaces and compressing the arrays into longer strings.

I've barely scratched the surface of Drawstring's capabilities. The best way to discover what it can do is by experimentation. In fact, you might consider using it to develop a full package of graphics commands for the Model 4. ■

John D. Wolfskill welcomes comments and suggestions. Send correspondence to 201 Puritan Drive, Middletown, RI 02840. Enclose a stamped, self-addressed envelope for a reply.

LOAD 80

System Requirements

Model 4
TRSDOS 6.x
Basic

Direction Commands

Direction	Command
Up	U
Down	D
Left	L
Right	R
Up and Left	E
Up and Right	F
Down and Left	G
Down and Right	H

String Control Commands

Action	Command
Move Pixel Pointer	B
Draw and Set	%
Draw and Complement	*
End of Data	$

Table 1. Drawstring commands.

Program Control Options

Action	POKE Value
Draw	&HFFF9,0
Complement	&HFFF9,1

Table 2. POKE address and values for controlling the draw/complement toggle.

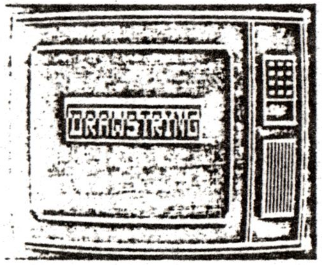


Photo 1. Program Listing 2 demonstrates drawing, erasing, and inverting patterns.

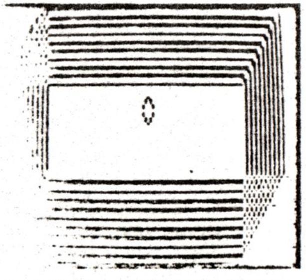


Photo 2. Program Listing 3 demonstrates animation: the diamond-shaped character can move across the screen without destroying the background graphic.

Program Listing 1. Drawstring.

```
20 REM *******************************************
30 REM *                                         *
40 REM *            DRAWSTRING                   *
50 REM *                                         *
60 REM *                BY                       *
70 REM *                                         *
80 REM *          JOHN  D. WOLFSKILL             *
90 REM *          201 PURITAN  DRIVE             *
100 REM*          MIDDLETOWN,RI 02840            *
110 REM*                                         *
120 REM*******************************************
130 CLEAR, 65177I:PRINT CHR$(15);
140 CLS:PRINT@240,"Verifying DRAWSTRING Data   ":PRINT
150 L=250
160 FOR J=1 TO 36:B=0:L=L+10
170 FOR K=1 TO 10:READ A:B=B+A:NEXT K
180 READ C
190 IF B=C THEN 200 ELSE PRINT "Error in Line ";L:END
200 NEXT J:RESTORE
210 PRINT "Data Correct --  Loading DRAWSTRING......";
220 FOR J=&HFE9A TO &HFFFF
230 X=X+1:IF X=11 THEN READ Q:X=1
240 READ AD:POKE J,AD:NEXT J
250 PRINT@480,"Successful Load..";:PRINT CHR$(14);:END
260 DATA 253,229,221,229,229,221,225,213,253,225,2290
270 DATA 3,10,95,3,10,87,27,6,5,33,279
280 DATA 250,255,54,0,35,16,251,62,255,50,1228
290 DATA 255,255,19,26,33,223,255,6,12,190,1274
300 DATA 40,50,35,35,16,249,214,48,250,186,1123
310 DATA 254,254,10,242,186,254,213,245,58,251,1967
320 DATA 255,183,32,11,60,50,251,255,241,50,1388
330 DATA 252,255,209,24,213,58,252,255,79,62,1659
340 DATA 90,30,10,239,193,128,50,252,255,209,1456
350 DATA 24,196,254,67,250,12,255,58,253,255,1624
360 DATA 183,32,22,60,58,253,255,35,126,50,1066
370 DATA 254,255,24,174,58,253,255,183,32,5,1493
380 DATA 35,110,38,255,233,27,58,252,255,183,1446
390 DATA 32,4,60,50,252,255,213,205,171,255,1497
400 DATA 58,250,255,183,204,64,255,58,252,255,1834
410 DATA 183,32,240,50,251,255,50,253,255,50,1619
420 DATA 250,255,209,195,186,254,62,93,221,94,1819
430 DATA 0,14,2,239,213,245,62,93,253,94,1215
440 DATA 0,14,3,239,183,241,111,34,247,255,1247
450 DATA 123,135,209,131,95,46,1,175,187,40,1142
460 DATA 5,29,203,37,24,248,229,62,15,6,858
470 DATA 1,42,247,255,239,203,127,40,25,225,1404
480 DATA 245,58,249,255,254,0,40,4,241,173,1519
490 DATA 24,2,241,181,79,62,15,6,2,42,654
500 DATA 247,255,239,201,175,203,255,24,226,62,1887
510 DATA 1,50,250,255,195,186,254,221,225,253,1890
520 DATA 225,201,175,24,2,62,1,50,249,255,1244
530 DATA 195,186,254,33,252,255,53,42,254,255,1779
540 DATA 233,221,53,0,201,221,52,0,201,253,1435
550 DATA 53,0,201,253,52,0,201,221,53,0,1034
560 DATA 253,53,0,201,221,52,0,253,53,0,1086
570 DATA 201,221,52,0,253,52,0,201,221,53,1254
580 DATA 0,253,52,0,201,76,179,82,183,85,1111
590 DATA 187,68,191,69,195,70,202,71,209,72,1334
600 DATA 216,66,147,36,155,37,160,42,163,0,1022
610 DATA 0,0,0,0,0,0,0,0,0,0,0
```

End

Program Listing 2. Drawing demonstration.

```
20 REM ******* DRAW DEMONSTRATION ************
30 CLEAR,65177I:DEFINT X,Y:DRAW=&HFE9A:X=0:Y=0
40 CLS:PRINT CHR$(15);
50 REM  .......................................
60 REM  . DRAWING DATA (SPACES MAY BE OMITTED) .
70 REM  . UPPERCASE LETTERS ONLY               .
```

Listing continued

Listing 2 continued

```
80 REM  . STRING LENGTH 1-255 CHARACTERS        .
90 REM  .......................................
100 MESSAGE$(0)="R2 U5 E R5 D6 L3 BR5 $"
110 MESSAGE$(1)="R U6 R4 D3 L3 G3 BR2 $"
120 MESSAGE$(2)="R U6 R4 D2 BL3 D R3 D3 BR2 $"
130 MESSAGE$(3)="R U6 BR BD5 D F U R D G R U6 BR2 BD6 $"
140 MESSAGE$(4)="BU R D R4 U3 L4 U3 R4 D BR3 $"
150 MESSAGE$(5)="U R4 BD BL L D5 BR3 $"
160 MESSAGE$(6)="R U6 R4 D3 L3 G3 BR2 $"
170 MESSAGE$(7)="R2 U6 L BU BR2 D BD5 D BR2 $"
180 MESSAGE$(8)="R U6 G2 D G BU4 R D6 BR2 $"
190 MESSAGE$(9)="R U6 R4 BD2 BL D R D3 L3 BR5 $"
200 MESSAGE$(10)="U8 L70 D10 R70 U2 $"
210 TV$(0)="%D71 R159 U71 L159 BG2 D67 R155 U67 L155 BR125 $"
220 TV$(1)="D66 BR R U65 BL12 BD10 D45 E3 L100 E3 U45 F3 R100 $"
230 TV$(2)="G3 BR18 R17 D20 L17 U20 BG2 R3 BR2 R3 BR2 R3 BD2 $"
240 TV$(3)="D L2 BL2 L3 BL2 L3 BD2 D R2 BR2 R3 BR2 R3 BD2 D L2 BL2 $"
250 TV$(4)="L3 BL2 L3 BL2 BD D R16 BD12 L17 D22 R17 U22 BL2 D22 L2 $"
260 TV$(5)="U22 BL2 D22 BL2 U22 BL2 D22 BL2 U22 BL2 D22 BL2 U22 $"
270 PATTERN$(1)="R D34 BR R U34 BR $"
280 PATTERN$(2)="* R D34 BR R U34 BR $"
290 REM **** DRAW OUTLINE ****
300 X=0:Y=0
310 FOR J=0 TO 5
320 CALL DRAW(X,Y,TV$(J))
330 NEXT
340 REM **** DRAW TEST PATTERN ****
350 FOR L=1 TO 2
360 X=15:Y=17
370 FOR J=1 TO 25
380 CALL DRAW(X,Y,PATTERN$(L))
390 NEXT:NEXT
400 REM **** DRAW EXPANDING BOX ****
410 L=0:GOSUB 420:L=1:GOSUB 420:GOTO 490
420 X=65:Y=34:POKE &HFFF9,L
430 FOR J=4 TO 48 STEP 2
440 BOX$="D"+STR$(J)+"R"+STR$(J)+"U"+STR$(J)+"L"+STR$(J)+"BE"+"$"
450 CALL DRAW(X,Y,BOX$)
460 NEXT
470 RETURN
480 REM **** DRAW MESSAGE ****
490 L=0:GOSUB 500:L=1:GOSUB 500:L=0:GOSUB 500:GOTO 570
500 X=31:Y=36:FOR TD=1 TO 700:NEXT
510 POKE &HFFF9,L
520 FOR J=0 TO 10
530 CALL DRAW(X,Y,MESSAGE$(J))
540 FOR TD=1 TO 100:NEXT
550 NEXT:RETURN
560 REM **** INVERT MESSAGE ****
570 INVERT$="*R73$"
580 GOSUB 590:GOSUB 590:L=1:GOSUB 500:CLS:GOTO 300
590 X=28:FOR TD=1 TO 1400:NEXT
600 FOR Y=26 TO 40
610 CALL DRAW(X,Y,INVERT$):X=28
620 NEXT:RETURN
```

End

Program Listing 3. Animation demonstration.

```
20 REM .......................................
30 REM .    ANIMATION DEMONSTRATION          .
40 REM .......................................
50 CLEAR,65177:DEFINT X,Y:DRAW=&HFE9A
60 X=0:Y=0:CLS:PRINT CHR$(15);
70 TOP$="*E3R6L6F3G3H3$":BORDER$="FHD40FHR100FHU40FHL100BR20U2$"
80 FILL$="*R80$"
90 X=20:Y=16
100 FOR J=1 TO 18
110 CALL DRAW(X,Y,BORDER$)
120 NEXT
130 X=39
140 FOR Y=18 TO 37
150 CALL DRAW(X,Y,FILL$):X=39
160 NEXT
170 X=80:Y=26
180 X$=INKEY$
190 IF X$="" THEN GOSUB 300 ELSE GOSUB 300
200 IF X$=CHR$(3) THEN 70
210 IF X$="U" THEN Y=Y-1
240 IF X$="D" THEN Y=Y+1
250 IF X$="L" THEN X=X-1
260 IF X$="R" THEN X=X+1
265 IF X$="E" THEN X=X-1:Y=Y-1
268 IF X$="F" THEN X=X+1:Y=Y-1
270 IF X$="G" THEN X=X+1:Y=Y+1
280 IF X$="H" THEN X=X-1:Y=Y+1
290 GOTO 180
300 POKE &HFF-9,0:[illegible] 310:[illegible] &HFFF9,0:[illegible] 310:[illegible]
310 CALL DRAW(X,Y,[illegible]
```

Rufen Sie zuerst den Krankenwagen, und dann suchen Sie bitte einen Freiwilligen, der ihm sagt, daß er seit drei Stunden Daten in einen kaputten Speicher eingibt.

- Die Alm liegt hoch im Gebirge. Dort ist der Senner und die Sennerin. Im Frühjahr wird aufgetriegen und im Herbst wird abgetrieben.
- Eine katholische Schwester kann nicht austreten, da sie zeitlebens im Kloster leben muß.
- Die Macht des Wassers ist so gewaltig, daß selbst der Stärkste es nicht halten kann.
- Wir gingen in den Zoo. Es war ein grosser Affe im Käfig. Mein Onkel war auch dabei.
- Der landwirtschaftsminister ließ die Bauern zusammenkommen, denn die Schweine fraßen zuviel.
- Beim Roten Kreuz widmen sich die Frauen und Männer der Liebe. Manche tun es umsonst, andere bekommen es bezahlt.
- Als der Jäger den dicken Bauch der Großmutter sah, wußte er sofort bescheid.
- Dort wo jetzt Ruinen ragen, standen einst stolze Burgfräulein und warteten auf die ausgezogenen Ritter.

Stilblüten aus Schüleraufsätzen

- Die Welt horchte auf, als Luther seine 95 Prothesen an die Schloßkirche zu Wittenberg schlug.
- Neben Prunksälen hatten die Ritter auch heizbare Frauenzimmer.
- Cäsar machte das Lager voll und jeder stand stramm bei seinem Haufen.
- Zeppelin war der Erste, der nach allen Richtungen schiffte.

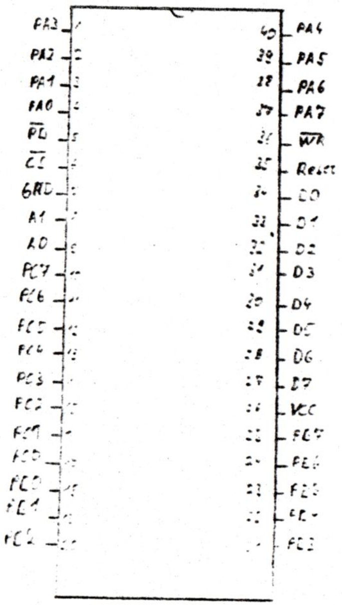
Der Computer hat doch einen perfekten Entwurf für das Haus angefertigt! – Na ja, beinahe . . .

HEFT 23
Mai
1986

Der programmierbare Interface - Adapter 8255 kann praktisch mit jedem Z80 Computer verbunden werden.

Pinbelegung des 8255

D0-D7	Dateneingangs- und Datenausgangsleitungen
$\overline{CS}$	Wenn dieser Eingang auf logisch 0 ist, kann der Mikroprozessor mit dem 8255 Verbindung aufnehmen.
$\overline{RD}$	Sobald der RD und der CS - Eingang logisch 0 sind, werden Daten vom Systembus in das interne Register des 8255 geschrieben.
$\overline{WR}$	Wenn der WR und der CS - Eingang logisch 0 sind, werden Daten vom Systembus in das interne Register des 8255 geschrieben.
A0-A1	Die logische Kombination dieser zwei Adresseingänge bestimmt das interne Register, das bei Lese/Schreib operationen angesprochen wird.
Reset	Wenn dieser Eingang auf logisch 1 gebracht wird, werden alle internen Register des 8255 gelöscht und die Portleitungen als Eingänge geschaltet
PA0-PA7	I/O - Signalleitungen des Port A
PB0-PB7	I/O - Signalleitungen des Port B.
PC0-PC7	I/O - Signalleitungen des Port C. Sie können auch in zwei 4 Bit Gruppen aufgeteilt werden, wobei sie als Steuerleitungen für Port A und B dienen.



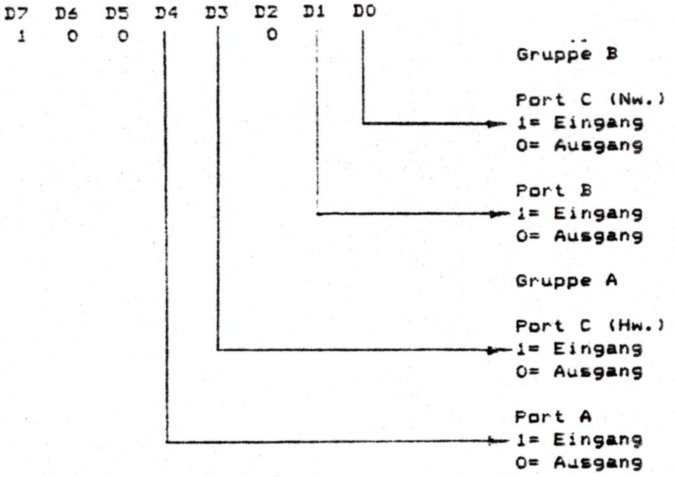
Programmierung des 8255

Hier ist nur der Modus 0 erklärt.(sonst siehe Datenblätter)

Die ICs 74LS04 und 74LS30 dienen zur Dekodierung des I/O Ports, damit andere nicht gleichzeitig angesprochen werden
Durch die Decodierung sind nun folgende Adressen möglich um den 8255 anzusprechen.
76D 4CH Datenport A schreiben oder lesen.
77D 4DH Datenport B schreiben oder lesen.
78D 4EH Datenport C schreiben oder lesen.
79D 4FH Steuerwort

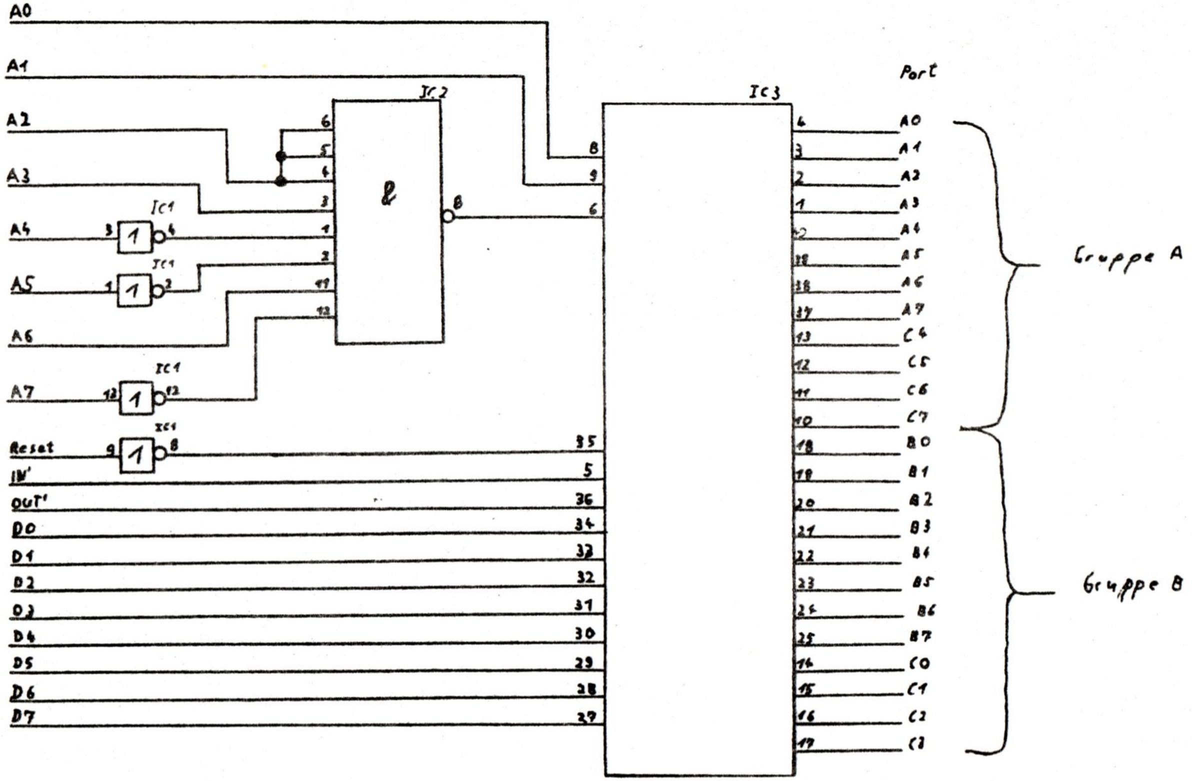
Als erstes gibt man über die Adresse 4FH das Steuerwort aus.
Dieses Steuerwort kann wie folgt aussehen.

Steuerwort



Ein Beispiel:

Das Steuerwort sei 10000000 = 80H = 128D
Out 79,128 (Out Befehl des Model 1 in Basic)
Der 8255 hat nun über die Adresse 79 die Mitteilung bekommen
das alle Ports als Ausänge geschaltet sind.
Out 76,255
Nun liegen an den 8 Leitungen des Ports A lauter 'High'



IC1 = 74LS04
IC2 = 74LS30
IC3 = 8255

Wir haben die 25 Mio nie verloren, Chef!
Das war nur ein Fehler im Computerausdruck.

Nur ein klitzekleiner Fehler im Steuerprogramm, Herr Major.

Signale.
Da die Ausgänge mit keinem größerem Strom belastet werden können, ist es sinnvoll Transistoren als Treiber zu verwenden.
Falls elektrische Geräte geschaltet werden sollen, muß unbedingt die Schaltung galvanisch durch Optokoppler getrennt werden

VIDEODAT 300

"VIDEODAT 300" - ein klangvoller Name, aber was verbirgt sich dahinter ? Mit dieser kleinen Beschreibung wollen wir versuchen Ihnen dies zu erklären.

In unseren letzten Sendungen haben wir Ihnen schon einiges über das vom Computerclub-Team entwickelte Verfahren erzählt. Neben Videotext und VPS ist dies eine Methode zur Übertragung von Informationen, Daten, Computerprogrammen und Schaltbildern mit Hilfe des Fernsehbildsignals. Das neue Verfahren soll künftig unseren sogenannten Hartbitrock am Sonntagnachmittag ersetzen.

Bei dieser Entwicklung ist es uns möglich dem Zuschauer Informationen zu übermitteln, ohne den Empfang des Fernsehprogramms merklich zu beeinträchtigen. Durch das Einstanzen einer kleinen, kaum erkennbaren schwarzen Fläche in die linke obere Ecke des Fernsehbildes ist unsere neue Übertragungsform gekennzeichnet. Während der Datenübertragung werden bestimmte Zeilen in dieser Fläche hell getastet. Dies geschieht nach dem asynchronen, seriellen Datenübertragungsverfahren. Das von uns gewählte Format entnehmen Sie bitte dem Schaubild 1.

Aus diesem Schaubild können Sie erkennen, daß die Schwarzflächen in der 23. Zeile des 1. Halbbildes und in der Zeile 336 des 2. Halbbildes beginnen. Die Begrenzung bildet Zeile 36 des 1. Halbbildes bzw. Zeile 349 des 2. Halbbildes. Seitlich beginnt das Fenster unmittelbar nach der hinteren Schwarzschulter und ragt ungefähr 17,5 μs nach Beginn des Zeilensynchronsignals in den Zeileninhalt. Das Startbit finden Sie jeweils in der Zeile 25 oder 338. Es ist immer weiß. Dies entspricht dem LOW-Signal auf der TTL-Ebene. Die Datenbits belegen die Zeilen 26 bis 33 bzw. 339-346. Entsprechend der Dateninformation sind diese Zeilen entweder schwarz oder weiß. Den Abschluß bilden die Zeilen mit den Stopbits. Die Stopbits sind immer schwarz.

Diese Anordnung und Belegung der Zeilen wurde von uns so gewählt, daß der Bildinhalt während der Übertragung kaum gestört wird. Viele Zuschauer werden unsere "schwarze Ecke" wahrscheinlich gar nicht erkennen können, da die meisten Fernsehempfänger das Bild überzeichnen. Sollten auch Sie zu den Zuschauern gehören, bei denen der übertragene Datenblock unsichtbar ist, so ist das für die Teilnahme an unserem Dienst unerheblich. Die Sichtbarkeit, bzw. Unsichtbarkeit der "schwarzen Ecke" hat keinerlei Einfluß auf die Funktionstüchtigkeit beim Datenempfang, da VIDEODAT 300 elektrisch mit dem Videosignal verbunden wird.

Diese Methode ermöglicht den Aufbau des Interfaces für den Empfang der übertragenen Signale mit einfachen Standardbausteinen und eine nicht vom Computertyp abhängige RS 232-Schnittstelle.

Nun jedoch zur Schnelligkeit unseres Verfahrens : In jedem Fernseh-"Halbbild" kann ein Byte übertragen werden. Nach der Fernsehnorm werden in jeder Sekunde 50 Halbbilder übertragen. Das ergibt einen maximalen Datendurchsatz von 50 Byte bzw. 50*(1+8+2) Bit- das entspricht 550 Bit in jeder Sekunde. Um kompatibel zur vorhandenen DFÜ-Software zu bleiben, werden von uns maximal 300 Bits in der Sekunde übertragen.

Wenn Sie also zu Hause diesen Informations- und Software-Service nutzen wollen, benötigen Sie außer Ihrem Computer das Interface "VIDEODAT 300" (siehe Schaubild 2).

Das Interface ist unabhängig vom Computertyp. Der Eingang des Interfaces wird mit dem Videoausgang Ihres Fernsehers oder Ihres Videorecorders verbunden. An den meisten Fernsehgeräten finden Sie das Videoausgangssignal an der AV- oder der Scart-Buchse.

Videoausgang an der AV-Buchse :
- Signal : Stift 2
- Masse : Stift 3

Videoausgang an der Scart-Buchse :
- Signal : Stift 19
- Masse : Stift 17

Das Interface VIDEODAT 300 separiert aus dem Videosignal das übertragene Datenfeld. Es stellt die darin enthaltenen Informationen an der RS 232-Schnittstelle (DB 25-Buchse) zur Verfügung. Genauso wie in der Datenfernübertragung mit Akustik-Kopplern werden diese seriellen Informationen mit 300 Baud, 1 Startbit, 8 Datenbits und einem Stopbit an Ihren Rechner weitergegeben.

Ihr Computer muß dementsprechend über eine serielle Schnittstelle (RS 232) und entsprechender DFÜ- oder Terminal-Software verfügen.

Die Daten werden nach dem 7-Bit-ASCII-Code übertragen. Das 8.Datenbit ist immer LOW. Die Daten-Listen werden in Blöcken mit jeweils 2kByte gesendet. Jedem Block folgt eine Pause von 15 Sekunden. Dadurch haben Benutzer von DFÜ-Software mit einem kleinen Datenpuffer die Möglichkeit, den empfangenen Block auf einer Diskette abzuspeichern.

Bei dieser Übertragungsmethode im ASCII-Format ist es erforderlich gesendete BASIC-Programme in sogenannte Tokenfiles bzw. Programmfiles umzuwandeln. Das ist auf verschiedene Art möglich. Einmal bei Anwendung von DFÜ-Software durch die Download-Funktion oder durch entsprechende Aufrufe des Betriebssystems, nachdem sich das Programm als ASCII-Datei auf der Diskette befindet. Bei anderen Rechnern wiederum erfolgt die Wandlung automatisch mit dem Aufruf des Load-Befehls.

Damit wäre ein übertragenes BASIC-Programm grundsätzlich lauffähig. Aber da stellt sich das Problem der von den Herstellern verwendeten, unterschiedlichen BASIC-Dialekte.

Der Schneider CPC benutzt beispielsweise zum Löschen des Bildschirms den Befehl CLS, der Apple HOME, der Commodore PRINT"inv.Herz" usw..Zum Ausgleich dieser Unterschiede haben wir uns bisher des von Radio Hilversum, Redaktion Hobbyscoop, entwickelten BASIC-Esperantos, dem BASICODE 2 bedient. Auch bei dem neuen Verfahren werden wir uns künftig dieser Unterprogramm-Philosophie anschließen.

In einem BASICODE 2-Programm, das immer mit der Zeile 1000 beginnt, werden die bei den einzelnen Rechnern unterschiedlichen BASIC-Anweisungen, die aber das gleiche bewirken, durch Standard-Unterprogramm-Aufrufe ersetzt. In unserem Beispiel "Bildschirm löschen" heißt es im BASICODE 2-Programm für alle Computer einheitlich : GOSUB 100. In der computertypspezifischen Unterprogramm-Bibliothek steht dann der entsprechende Befehl, gefolgt von einer RETURN-Anweisung.

Nach dem Empfang eines BASICODE 2-Programms und dessen Wandlung in ein Tokenfile, muß mit Hilfe einer MERGE-Funktion, die für Ihren Computer spezifische Unterprogramm-Bibliothek hinzugeladen werden.

Die Zuschauer, die bereits über einen BASICODE 2 Übersetzer verfügen, finden diese Unterprogramme in den Programmzeilen 10 bis 369 ihres Übersetzerprogramms. Aber auch den Benutzern der Rechner, die bisher nicht dem BASICODE 2 angeschlossen waren, wird mit unserer Entwicklung die Möglichkeit gegeben künftig an unserem kostenlosen Software-Service teilzunehmen, da es nicht, wie bisher, erforderlich ist das Kassettenaufzeichnungsformat an eine bestimmte Norm anzupassen. Zur Teilnahme gnügt es nun, über die entsprechenden Unterprogramm-Routinen zu verfügen.

Um Ihnen die Möglichkeit zu geben diese Routinen selbst zu erstellen, haben wir nachstehend eine Leistungsbeschreibung der BASICODE 2-Unterprogramm-Bibliothek vorbereitet :

Zeile 10 : In dieser Zeile steht immer GOTO 1000

Zeile 20 : In dieser Routine werden soviele Speicherplätze für Stringspaces reserviert, wie der Inhalt der Variablen A (in Zeile 1000) angibt. Den Abschluß bildet die Anweisung GOTO 1010.

Alle folgenden Routinen sind Unterprogramme und enden mit der Anweisung RETURN.

Zeile 100 : Diese Routine löscht den Bildachirm und bringt den Cursor in Position 0,0 links oben auf dem Bildschirm.

Zeile 110 : Bringt den Cursor zu einer bestimmten Bildschirmstelle. Bevor diese Routine angesprungen wird, muß in den Variablen HO und VE die gewünschte Cursorposition stehen, und zwar in HO die Spalten- und in VE die Zeilennummer. Für HO ist eine Zahl von 0-39 und für VE von 0-23 zulässig. Im BASICODE hat der Bildschirm 24 Zeilen mit jeweils 40 Zeichen.

Zeile 120 : Ist praktisch eine Umkehrung der Routine 110. Die aktuelle Cursorposition wird in die Variablen HO und VE geschrieben. Indem Sie die Cursorposition mit GOSUB 120 abfragen, können Sie mit GOSUB 110 z.B. anschließend den Cursor um eine oder mehrere Zeilen höher oder tiefer setzen.

Zeile 200 : Diese Routine überprüft, ob eine Taste gedrückt ist, und speichert das entsprechende Zeichen in die Zeichenkettenvariablen IN$ ab. Dabei können durchaus auch Kontrollzeichen, wie zum Beispiel RETURN, gespeichert werden. Achten Sie in Ihren eigenen Programmen darauf, daß Sie nach Möglichkeit keine Kontrollzeichen abfragen, weil diese von Rechner zu Rechner unterschiedlich sein können. ENTER bzw. RETURN jedoch hat bei allen Rechnern den ASCII-Code 13. Ist keine Taste gedrückt, so enthält IN$ kein Zeichen.

Zeile 210 : Grundsätzlich entspricht diese Routine der vorangegangenen, nur wartet dieses Unterprogramm, bis eine Taste gedrückt ist.

Zeile 250 : Über diese Routine kann bei den Rechnern, die dazu in der Lage sind, ein Tonsignal ausgegeben werden. Die Tonhöhe und -dauer sind nicht festgelegt, d.h. Musik können Sie mit dieser Routine nicht machen.

Zeile 260 : Erzeugt eine Zufallszahl zwischen 0 und kleiner 1 und speichert diese in der Variablen RV. Diese Zufallszahl kann 0 sein aber nie gleich 1 werden. Sinnvol ist diese Routine zum Beispiel für Spiele.

Zeile 270 : Säubert den Variablenbereich und legt die Anzahl der noch freien Bytes des Speichers in der Variablen FR ab. Die Werte der Variablen werden durch Aufruf dieser Routine nicht gelöscht.

Zeile 300 : Bildet aus dem Inhalt der Variablen SR die Zeichenkette SR$. Diese Zeichenkette hat werder am Anfang noch am Ende ein Leerzeichen, was sie von einer Zeichenkette, die mit STR$() gebildet wird, unterscheidet. Einige Rechner fügen nämlich Leerzeichen hinzu, deshalb ist STR$() im BASICODE 2 nicht zugelassen.

Zeile 310 : Diese Routine erzeugt eine Zeichenkette SR$, deren Form durch die Variablen CT und CN und deren Inhalt durch die Variable SR bestimmt wird. Die erzeugte Zeichenkette (hier eigentlich Ziffernkette) hat Festpunktdarstellung. Die Gesamtlänge beträgt CT-Zeichen, davon CN-Zeichen nach dem Dezimalpunkt. Falls erforderlich wird SR genau gerundet. Paßt SR nicht in der angegebenen Form in SR$, so enthält SR$ nur Sternchen. Durch den Aufruf der Routine 310 werden die Variablen CT,CN und SR nicht verändert.

Beispiele :

```
CT=7:CN=3:SR=2/3:GOSUB 310
                    SR$="  0.667"
CT=8:CN=5:SR=-1.1E-3:GOSUB 310
                    SR$="-0.00110"
CT=3:CN=0:SR=23.6:GOSUB 310
                    SR$=" 24"
CT=3:CN=1:SR=100:GOSUB 310
                    SR$="***"
```

Zeile 350 : Druckt SR$ auf dem Drucker aus, schließt jedoch die Zeile nicht ab, d.h. Sie können in einer Zeile mehrfach drucken. Denken Sie bei eigenen Programmen daran, daß nicht jeder über einen Drucker verfügt. Fragen Sie im Programm also besser ab, ob die Ausgabe auf dem Bildschirm oder auf dem Drucker erfolgen soll.

Zeile 360 : Schließt eine Druckzeile mit einem Wagenrücklaufkommando ab, d.h. der nächste Druckvorgang findet in der nächsten Zeile statt.

Die Redaktion wird bemüht sein, für neue, stark verbreitete Rechner, die BASICODE 2-Bibliothek zu veröffentlichen.

Nach soviel Einleitung und Softwareinformationen wollen wir uns jetzt der Hardware von VIDEODAT 300 zuwenden. Das Schaubild 3 zeigt Ihnen die wesentlichen Komponenten.

VIDEODAT 300 erwartet am Videoeingang einen Signalpegel von 1 V Spitze-Spitze an 75Ω. Das Videosignal muß negativ synchron sein. Der einstellbare Videoverstärker hebt dieses Signal auf 6 Vss an und dreht es um 180° in der Phase. Der Videoverstärker ist der einzige Abgleichpunkt in dieser Schaltung. Dadurch wird die Anpassung an zu schwache oder zu starke Videosignale möglich. Die Einstellung ist in weiten Grenzen unkritisch.

Zwischen Videoverstärker und Amplitudensieb befindet sich eine Klemmschaltung. Deren Aufgabe ist es, den"Boden"der Synchronsignale, unabhängig vom aktuellen Bildinhalt, auf ein bestimmtes Bezugspotential zu fixieren.

Das Amplitudensieb besteht aus zwei Vergleichern. Der Ausgang des 1. Vergleichers wird LOW, wenn sich Weißanteile im Bild befinden (Integrationszeit ca. 1µs). Der Ausgang des 2. Vergleichers isoliert die Synchronimpulse und führt sie dem Controller zu.

Der Controller ist verantwortlich für die Takterzeugung und das Timing der Schaltung. Er liefert an das Daten-Flip-Flop 14µs nach Beginn eines Zeilensynchronsignals, innerhalb eines Datenfensters einen Clock-Impuls. Das bewirkt die Übernahme der aktuellen Dateninformation der entsprechneden Zeile für die Dauer einer Zeile (64µs).

Außerhalb des Datenfensters wird das Daten-Flip-Flop auf HIGH-Signal (Stopbit-Pegel) gezogen. Am Ausgang "Q" des Daten-Flip-Flop stellt sich somit ein asynchrones Datenübertragungssignal mit 15625 Baud für die Dauer eines Bytes ein: Nach jedem Byte folgt eine große Pause durch Verlängerung des Stopbits.

Dieses serielle Signal und der Empfangstakt von 15625 Hz*16= 250 kHz werden dem UART-Empfangsteil zugeführt. Dieser empfängt die Dateninformation und gibt sie parallel an den UART-Sendeteil weiter.

Auf grund der dort angelegten Sendetaktfrequenz von 300Hz*16= 4,8kHz wird die empfangene Dateninformation mit 300 Bit/Sec. gesendet.

Eine Pegelanpaßstufe erzeugt schließlich die normgerechten Pegel des RS 232-Signals. Die ursprünglich im Bild enthaltene Dateninformation steht hiernach an Pin 3 und 7 der DB 25-Buchse für Ihren Rechner bereit.

Technische Daten :

Versorgung	:220 V 10%, 50 Hz
Umgebungstemperatur	:0.....40°C
Videoeingang	:1Vss an 75Ω neg.syn. (Cynchbuchse)
Datenausgang	:RS 232 (+/- 12V,RL > = 2kΩ) DB 25 Buchse 300 Baud/asynchron 1 Startbit 8 Datenbits 2 Stopbits kein Prüfbit

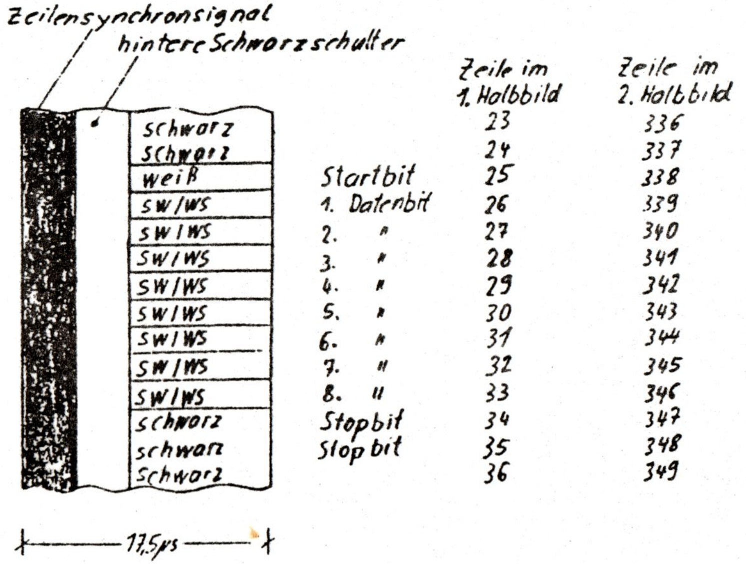


BILD:1

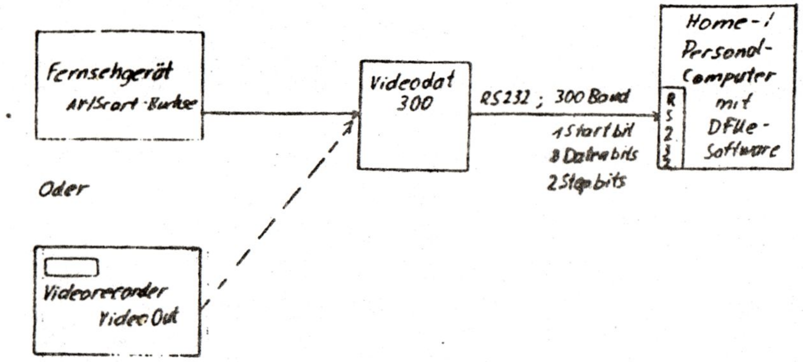


BILD:2

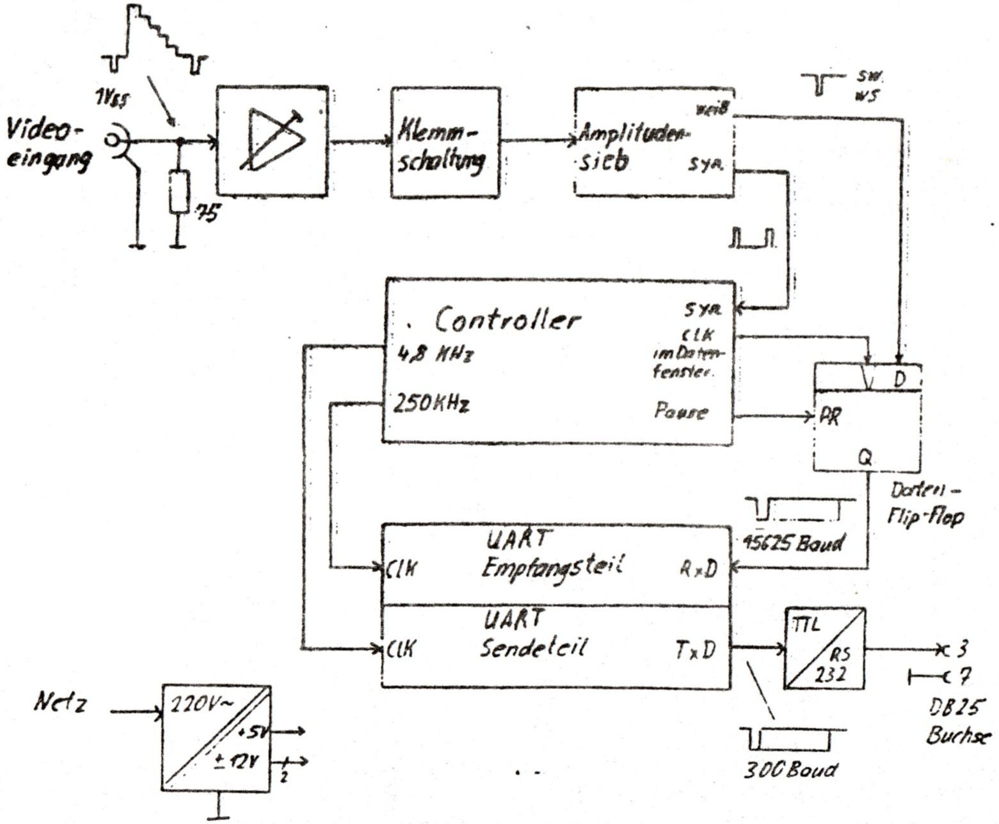


BILD:3

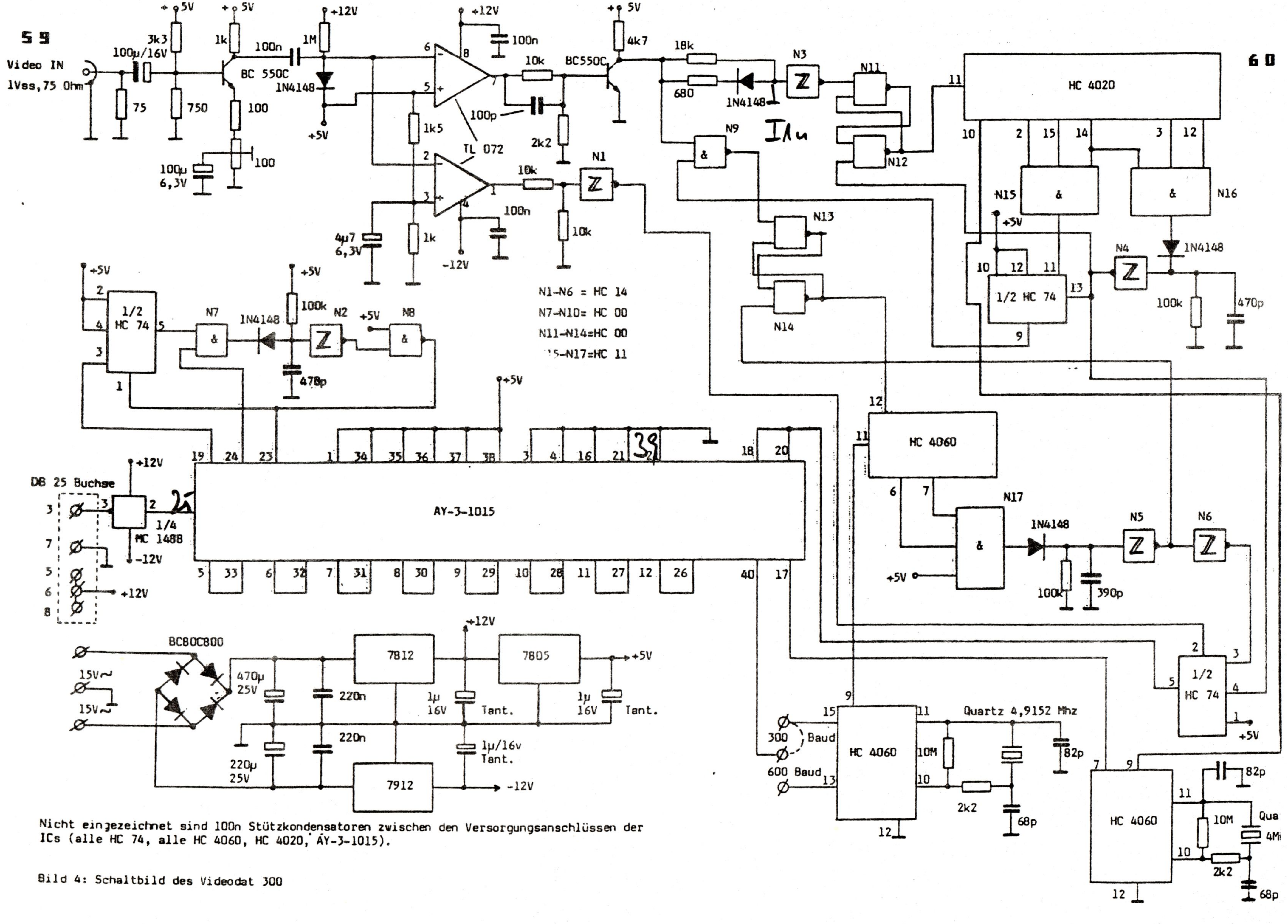
WDR - Programme   WDR - Programme

Die Programme, die auf diese Art und Weise überspielt werden sind auf unserer Maschine lauffähig. Weiterhin könnten diese Programme auch in unsere Programm-Bibliothek eingebracht werden, da sie über den FS-Anschluß für jedermann zugänglich sind.

Unsere Fragen:
Wer wohnt im Empfangsbereich des Senders ?
Wer hat schon ein solches Gerät in Betrieb
und kann die Programme zur Verfügung stellen ?
Wer hat ein Interesse an dieser Sache ?

Bitte meldet Euch bei unserem Hardwarekoordinator

- Es war eine machtvolle Demonstration. Der Marktplatz war voller Menschen. In den Nebenstraßen pflanzten sich Männer und Frauen fort.
- Die Streichhölzer müssen gut versteckt werdem, damit si keine kleinen Kinder bekommen.
- Meine Tante schenkte mir eine Sparbüchse Sie war ein Schwein und hatte zwei Schlitze Hinten einen fürs Papier und vorne fürs Harte.
- Wir gingen mit unserer Lehrerin spazieren im Park. Gegenüber vom Park war ein Haus, wo Mütter ihre Kinder gebären. Eine Gebärmutter schaute gerade aus dem Fenster und winkte uns zu.
- Zu Allerheiligen waren wir auf dem Friedhof Es war sehr feierlich, denn der Herr Pfarrer besprang mit seinem Wedel die Friedhofsbesucher.
- Gestern hatten wir unseren Schitag der Jugend. Alle Minuten ließ der Lehrer einen fahren. Wenn einer zu früh auskam, schickte er ihn sofort wieder zurück.



Nicht eingezeichnet sind 100n Stützkondensatoren zwischen den Versorgungsanschlüssen der ICs (alle HC 74, alle HC 4060, HC 4020, AY-3-1015).

Bild 4: Schaltbild des Videodat 300

Börse Börse Börse Börse Börse Börse Börse Börse Börse Börse

Zu verkaufen!

1 Schneider CPC 464 mit Grünmonitor (15 Monate alt)
1 Schneider Matrixdrucker NLQ401 incl. Traktor (5 Mon. alt)
1 Lightpen incl. Software
div. Literatur (u.a. CPC Intern von Data Becker)
div. Software (u.a. Monitor, Assembler, Disassembler usw.)
Staubschutzhauben für das komplette System
Preisvorstellung ca. 1700,-- (Verhandlungssache)
Interessenten melden sich bitte telefonisch bei:
Hartmut Obermann ☎ 06124 /3913

Verkaufe Schaltnetzteil
----- "Neu" -----
+/- 12V/5V 7A
für 85,-- DM

Suche Drucker-Buffer !!!
Wer hat Informationen dazu ????

Peter Stevens

Verkaufe

Genie I 64K mit EG64 (CP/M-fähig)	**350 DM**
Expander mit Doubler	**300 DM**
Monitor Zenith	**150 DM**
Doppellaufwerk 40 SD/DD	**400 DM**
Original-Newdos mit Anleitung	**150 DM**
Forbidden City (Talking Adventure)	**30 DM**
Golden Baton (Adventure)	**20 DM**
EDAS (Editor-Ass. für LDOS) mit Anl.	**80 DM**

zusammen: 1200 DM

Gerald Schröder

Wer hat einen Doubler für GENIE I zu verkaufen?
Herbert Albers, 2117 Wistedt

Verkaufe:
=========

(komplett oder einzeln)

- TRS 80 Modell 1, 16 KB (DM 330,--)
- Expansion Interface, 32 KB, 5/8 Zoll Doubler (DM 400,--)
- 2 x 40 Tr. Laufwerke i. Gehäuse (DM 360,--)

komplett DM 980,--

Klaus Hermann, Tel.: 07127/70024

Tandy Color Computer
32K--Neu--VHB 200,--DM
(vielleicht als Geschenk für den Neffen)

Peter Stevens

Modell 4p und 4
64K, 2 Laufwerke
Neu! Sonderpreise!
über Peter Stevens
1590,--DM !!!
oder mit 128K
1790,--DM !!!

Genie I "GTI" zu verkaufen

Genie I, Bj. 1984, CP/M 3.x-fähig, 315¾ kB, und zwar:

ROM	(0000-2FFFh)	12 kB		
RAM	(4mal 0000-FFFFh)	256 kB	von Helmut Bernhardt	
ROM/RAM	(16mal 3000-37DFh)	31 kB	" " "	
RAM	(3900-3BFFh)	0.75 kB	" " "	
RAM	(HRG um 4 kB erw.)	16 kB	" Ulrich Böckling	

Im ästhetischen Gehäuse des Commodore 610/620 mit vollständig bestückter Tastatur, Dez-/Hexziffernblock und Cursorfeld: Umlaute, ß, Cursor, Funktionstasten, CTRL, ESC usw. (Firmenname wird auf Wunsch schamhaft entfernt)

Zusätzlich eingebaut: HRG 1b von RB-Electronic (belegt nur noch Ports 0-7), Speed-Up 1.77/3.54 MHz mit Z80A-CPU von Udo Jourdan, CPU-Reset-Knopf, Lautstärkeregler/-schalter, Ohrhörerbuchse, INT/NMI-Timer regel-/schaltbar, Schaltnetzteil

2 Banker, EG 64 MBA von TCS und Helmut Bernhardts verbesserte Alternative: Kann die Tastatur- und Bildschirmadressen für CP/M nach F7E0-FFFF verlegen und wahlweise die Tastatur, die Floppy und den Drucker über Ports adressieren, so daß nur noch 1 kB für den Bildschirm ab FC00 dem Hauptspeicher verlorengeht; vorbereitet für Zusammenarbeit mit der 80-Zeichenkarte

nicht eingebaut, aber mitgeliefert: ECB-Buskarte, serielle Schnittstelle, Platine zum Aufrüsten auf 1 MB (nur Jumpers auf der 256-kB-Karte umlöten und zusätzliche RAMs reinstecken)

Floppy 2mal 80/DS/DD (Mitsubishi) mit Controller im TCS-Gehäuse, Bernsteinmonitor, Druckerkabel, Zweifach-Busextender, Handbücher, H-DOS

Außerdem so dies und das: 2 Genie-I-Gehäuse, 1 Tastatur mit Ziffernblock, Genie-I-Netzteil, Vielfachmeßgerät ICE 680 R (mit Kapazitäten, Frequenzen usw.), Taschenrechner CASIO FX-602P (alphanum., 512 Programmschritte), CASIO FX-512 (dez, hex, oct, bin).

Habe ich etwas vergessen? Wahrscheinlich!

Über Einzel- bzw. Komplettpreise wäre jeweils zu verhandeln. Wer mit diesen knappen Stichworten nicht viel anfangen kann, kriegt gegen Freiumschlag eine ausführliche Beschreibung.

Arnulf Sopp, ☎ 0451-791926

HEFT 13 Mai 1986

63

Hilfe!

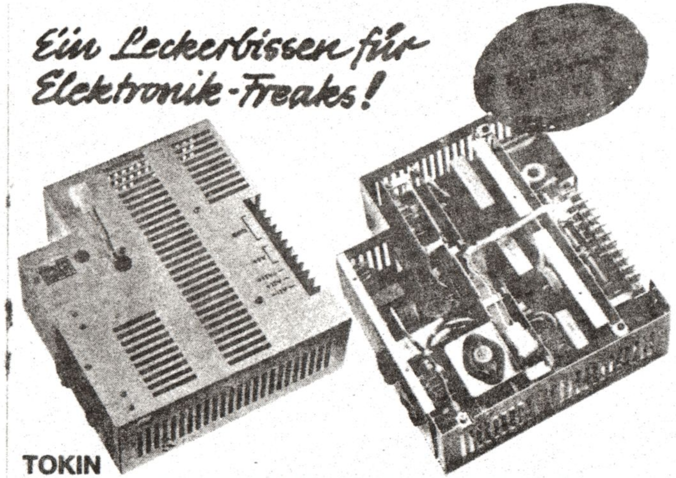
Wußtet ihr schon, daß Tandy eine sog. Hotline unterhält?! über diese ständig (???) besetzte Telefonverbindung soll Besitzern von Tandycomputern bei Problemen Hilfe geleistet werden. Die Hotline-Nummer lautet: 02102 / 4983 - 18
Eigentlich sollte sich dann ein Herr Burghart melden und hilfreich tätig werden.

Wenn jemand mal Hilfe braucht kann er's ja über diese Nummer versuchen. Ich wäre dankbar über jede Information, ob diese Serviceleistung tatsächlich funktioniert und ob man auch als Besitzer eines Model 1 noch geholfen bekommt!

Hartmut Obermann

Conrad-Netzteil
Vielleicht für den Bastler interessant ????

Ein Leckerbissen für Elektronik-Freaks!

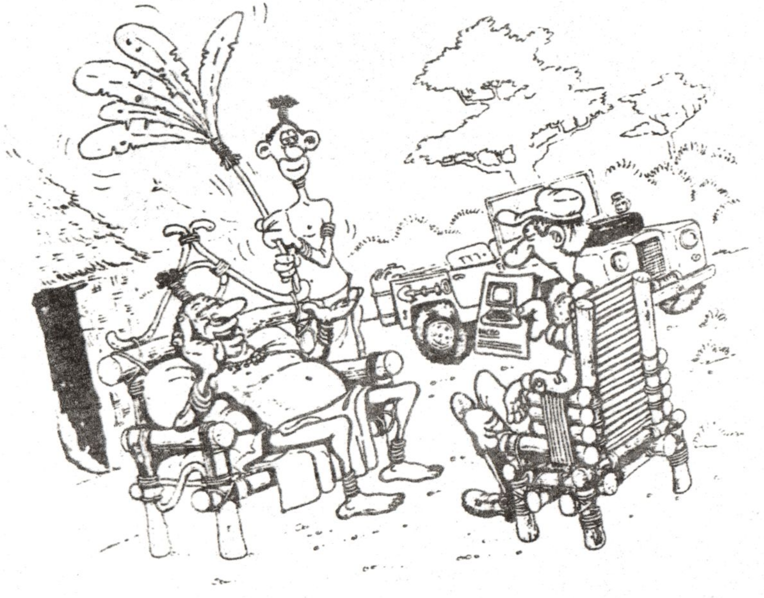


TOKIN
Professionelles HI-TECH-Schaltnetzteil, Typ PSS-166

Ein Gerät der 1000.- DM Klasse für nur 149.- DM!
Dieses in VDE-gerechtem Aufbau gehaltene Qualitäts-Schaltnetzteil wurde für folgende Anwendungsbereiche konzipiert:
Stromversorgung für professionelle Computertische · Stromversorgung für Selbstbaucomputer · Stromversorgung für den professionellen Meßplatz · Digitale TTL- und CMOS-Schaltungen · Operationsverstärkerschaltungen · Ergänzung für den Hobby-Arbeitsplatz · Als Stromversorgung für das Hobby-Elektroniklabor u.v.a.m.
Zur Störunterdrückung dienen sehr aufwendig aufgebaute Siebketten, sowohl auf der Eingangs- als auch auf der Ausgangsseite. Bei voller Belastung der einzelnen Ausgänge ist die Restwelligkeit kleiner als 1 mV. Alle Ausgänge sind kurzschlußfest und arbeiten mit einer elektronischen Strombegrenzung. Die galvanische Trennung erfolgt durch VDE-geprüfte Opto-Koppler (CNY 65).
Abm. (B x H x T): 260 x 85 x 235 mm · Gewicht: 2,8 kg · Eingang: Umschaltb. 110/220 V · Ausgänge über Klemmleiste: +5 V/20 A, +12 V/7 A, +12 V/2 A, −12 V/400 mA (gemessene Werte) · Ausgänge über Steckerbuchse: + 12 V zum Anschluß einer Betriebsanzeige (z.B. Anzeigelämpchen). Lieferung erfolgt inkl. Netzkabel (Gerät ist sofort anschlußfertig), sowie ausführlichen technischen Unterlagen.

Best.-Nr. 99 91 05 Stück **149.-** ab 3 St.à **125.-**

»Wer ein Volk regiert, indem er gute Beispiele gibt, ist wie der Polarstern, der unbeweglich bleibt, während alle anderen sich um ihn drehen.«



Was sind Ihre mickrigen 64 Kilobytes Speicherplatz gegen die Millionen von Bäumen in unserem Urwald oder die Milliarden von Grashalmen auf unseren Savannen!

Computerkriminalität wird schärfer bekämpft

Bonn. (dpa) Der Bundesrat hat das zweite Gesetz zur Bekämpfung der Wirtschaftskriminalität verabschiedet. Schwerpunkt des Gesetzes, das voraussichtlich noch im Sommer dieses Jahres in Kraft treten soll, sind Maßnahmen zur Bekämpfung der Computerkriminalität. Nach den neuen Bestimmungen muß künftig derjenige mit bis zu drei Jahren Haft oder Geldstrafe rechnen, der sich aus besonders gesicherten Datenspeichern Informationen verschafft. Das bloße Einschleichen in Dateien, wie dies von sogenannten „Hackern" betrieben wird, bleibt solange straffrei, wie von den dabei gewonnenen Daten kein Gebrauch gemacht wird. Ebenfalls strafbar macht sich, wer Daten unbefugt löscht und verändert.

»Unsere Show hat allerdings schon eine erheblich längere Laufzeit.«

Brian Brindley, englischer Pfarrer, in einem Kommentar zu John Lennons Äußerung, die Beatles seien berühmter als Jesus

Unternehmenschef Heinz Nixdorf gestorben

Computer waren sein Programm

Sein Aufstieg als Unternehmer liest sich wie ein Roman aus der Wirtschaftswunderzeit. Er war der Selfmademan aus der Provinz, der die Unternehmenszentrale auch nicht aus dem westfälischen Paderborn nach München verlegte, als sein Name Weltgeltung erlangt hatte. Und er gehörte zu den bedeutendsten Erscheinungen der deutschen Unternehmerszene: Heinz Nixdorf, Gründer und Vorstandsvorsitzender der Nixdorf Computer AG. Am 9. April wäre er 61 Jahre alt geworden. Am Montag abend erlag er in Hannover einem Herzversagen. Er starb im Kreis von Kunden und Mitarbeitern auf einer Veranstaltung seines Unternehmens anläßlich der Hannover-Messe Cebit. Heinz Nixdorf hinterläßt Ehefrau und drei Söhne.

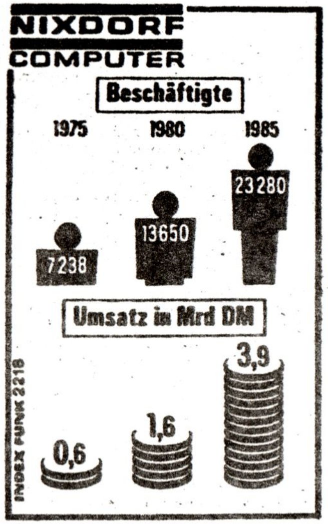
Der technisch brillante Computer-Pionier gründete 1952 in Essen in einer Kellerwerkstatt eine eigene Firma, das „Labor für Impulstechnik". Er war damals 27 Jahre, hatte nach 9 Semestern Physik und Betriebswirtschaftslehre das Studium abgebrochen und verfügte über keinerlei eigenes Kapital. Die Rheinisch-Westfälischen Elektrizitätswerke (RWE) in Essen stellten ihm eine Starthilfe von 30 000 DM zur Verfügung. 1958 machte Nixdorf als eigenwilliger Tüftler erstmals von sich reden, als er eine Transistor-Komponente für einen Buchungsautomaten der Wanderer-Werke, seinen Hauptkunden, entwickelte. Noch in jüngster Zeit, längst vom Erfolg verwöhnt, erzählte Heinz Nixdorf gern und nicht ohne Stolz aus dieser Gründerzeit.

Der unternehmerische Durchbruch gelang ihm 10 Jahre später, inzwischen an seinen Geburtsort Paderborn umgesiedelt, mit dem Kauf der Wanderer-Werke. Durch den Erwerb dieses Unternehmens für 17,4 Mio. DM bekam Nixdorf ein gut eingeführtes Vertriebs- und Service-Netz. Nixdorf konzentrierte sich von Anfang an auf die kleine und mittlere Datentechnik und setzte sich erstaunlich schnell gegen Produkte der Konkurrenz wie IBM und Siemens durch. Daher gilt Nixdorf als maßgeblicher Wegbereiter des Einsatzes von arbeitsplatzorientierten kleineren Computern, deren dezentrale Anwendungsmöglichkeiten er von Mitte der 60er Jahre an zentralistischen Organisationskonzepten internationaler Wettbewerber erfolgreich entgegensetzte.

Wie erfolgreich die Nixdorf Computer AG mittlerweile ist, verdeutlichen die Zahlen: 1966 betrug der Umsatz des Unternehmens rund 28 Mio. DM, 10 Jahre später waren es bereits 686 Mio. DM. Im vergangenen Jahr steigerten die Paderborner den Umsatz weltweit mit über 23 000 Mitarbeitern um 20 v. H. auf rund 3,9 Mrd. DM, wovon fast 2 Mrd. DM auf das Inland entfielen. Das Unternehmen investiert nach wie vor viel Geld in die Forschungs- und Entwicklungsarbeit. Allein 1985 wurden in Sachanlagen 510 Mio. DM investiert. Heinz Nixdorf hatte sich vorgenommen, schneller als die Branche zu wachsen, um die Marktführung in Europa zu erreichen.

Um das schnelle Wachstum realisieren zu können, ging Nixdorf 1984 an die Börse: Mit stimmrechtslosen Vorzugsaktien und nur einem Teil seines Grundkapitals, um weiterhin Herr im eigenen Haus zu bleiben. Daß der Westfale Nixdorf, der ohnehin in kein gängiges Klischee von Managern und Machern paßte, recht eigenwillig und kantig sein konnte, bekam unter anderem im vergangenen Jahr die Deutsche Bundespost zu spüren. Auf der vergangenen Hannover-Messe hatte sich Nixdorf über das aus seiner Sicht zu langsame Vorgehen der Bundespost bei der Einführung des neuen digitalen Fernsprech- und Datennetzes öffentlich beklagt.

Nixdorf machte mit vielen unternehmerischen Besonderheiten von sich reden. Das Unternehmen hat rund 1100 Ausbildungsplätze. Auf dem Gelände des Unternehmens steht eine private Berufsschule. Allein im technischen Bereich umfaßt die Ausbildung 14 Berufe. Außerdem ließ Nixdorf ein Sportzentrum auf dem Betriebsgelände bauen, den Zehnkämpfer Kurt Bendlin engagierte er als Leiter der Sport- und Ausbildungsförderung. Jede Firmenangehörige, die ein Kind bekommt, erhält zur Geburt einen Scheck. Alleinstehende Frauen erhalten ihn in doppelter Höhe.



Nixdorf, der bereits Mitte der 70er Jahre einen Herzinfarkt überstand, hatte seinem designierten Nachfolger Klaus Luft mehr und mehr die Führung überlassen.

Nach wie vor blieb er aber bis zuletzt Weichensteller und Ideengeber des weltweit renommierten Computer-Unternehmens. Seinen offiziellen Nachfolger wird der Aufsichtsrat der Nixdorf AG voraussichtlich auf seiner nächsten Sitzung in 14 Tagen bestellen. (14.3.86) dpa/VWD/dik

»Haben sie schon mal darüber nachgedacht, weshalb ich Sie noch nicht durch einen Computer ersetzt habe?«

Neues von der Bundespost

In Dienstanfängerkreisen kommen immer wieder Verwechslungen der Begriffe "Wertsack", "Wertbeutel", "Versackbeutel" und "Wertpaketsack" vor. Um diesem Übel abzuhelfen, ist das folgende Merkblatt dem Paragraphen 49 der ADA vorzuheften:

Der Wertsack ist ein Beutel, der aufgrund seiner besonderen Verwendung im Postbeförderungsdienst nicht Wertbeutel, sondern Wertsack genannt wird, weil sein Inhalt aus mehreren Wertbeuteln besteht, die in dem Wertsack nicht verbeutelt, sondern versackt werden. Das ändert aber nichts an der Tatsache, daß die zur Bezeichnung des Wertsackes verwendete Wertbeutelfahne auch bei einem Wertsack mit Wertbeutelfahne bezeichnet wird und nicht mit Wertsackfahne, Wertsackbeutelfahne oder Wertbeutelsackfahne.

Sollte sich bei der Inhaltsfeststellung eines Wertsacks herausstellen, daß ein in einem Wertsack versackter Versackbeutel statt im Wertsack in einem der im Wertsack versackten Wertbeutel hätte versackt werden müssen, so ist die zuständige Versackstelle unverzüglich zu benachrichtigen.

Nach seiner Entleerung wird der Wertsack wieder zu einem Beutel, und er ist auch bei der Beutelzählung nicht als Sack, sondern als Beutel zu zählen. Bei einem im Ladezettel mit dem Vermerk "Wertsack" eingetragenen Beutel handelt es sich jedoch nicht um einen Wertpaketsack, weil ein Wertsack im Ladezettel nicht als solcher bezeichnet wird, sondern lediglich durch den Vermerk "versackt" darauf hingewiesen wird, daß es sich bei dem versackten Wertbeutel um einen Wertsack und nicht um einen ausdrücklich mit "Wertsack" bezeichneten Wertpaketsack handelt. Verwecklungen sind insofern im übrigen ausgeschlossen, als jeder Postangehörige weiß, daß ein mit Wertsack bezeichneter Beutel kein Wertsack, sondern ein Wertpaketsack ist.

»Fehlstart« vermeidbar

Ursachen und Tips zur Abhilfe bei Initialisierungspannen neuer Disketten

(Dö) Gehören Sie zu dem Kreis der Kleincomputeranwender, die ihre Diskette selbst initialisieren können? Wenn ja, ist für Sie die Diskettenbeschaffung relativ einfach und meist sehr preisgünstig. Dafür können aber beim erstmaligen Initialisieren Probleme auftreten. Und wenn der Kleincomputer bei der soeben neu gekauften Diskette streikt, weil er die Initialisierung nicht durchführen kann, vermutet man meist die Floppy als den Störenfried. Die Angabe des Diskettenherstellers, wonach die vorliegende Scheibe zu »100 Prozent fehlerfrei« sei, scheint falsch zu sein. Die BASF als Disketten- und Laufwerkhersteller klärt diesen Widerspruch auf und erläutert Ursachen und Abhilfemaßnahmen bei Initialisierungspannen.

Ursache Nummer 1:

Nach dem Starten des Initialisierungsprogramms muß das Diskettenlaufwerk den Magnetkopf auf eine bestimmte Spur positionieren. Unmittelbar danach beginnt der Initialisierungsvorgang. Zwischen diesen beiden Vorgängen muß ein winziger Moment als sogenannte Kopfberuhigungszeit eingeplant werden, denn der ruckartig über die benötigte Spur positionierte Magnetkopf muß erst zur Ruhe kommen. Falls die Kopfberuhigungszeit bei einem beispielsweise extrem preisgünstigen Diskettenlaufwerk zu lange dauert oder in der Initialisierungssoftware die benötigte Pause zu knapp bemessen ist oder nicht für den verwendeten Floppy-Laufwerktyp vorgesehen war, kommt es zu einer Meldung wie »Diskette läßt sich nicht initialisieren«.

Hier ist ein zweiter Initialisierungsversuch meist erfolgreich. Denn beim zweiten Schreibversuch ohne erneute Kopfpositionierung wird die erwähnte Beruhigungszeit nicht benötigt. Der Kopf ist bereits angedrückt auf der anderen Seite. Das Umschalten ist nicht mit mechanischen Bewegungen verbunden (elektronisch im Millisekundenbereich).

Ursache Nummer 2:

Schon das kleinste Staubteilchen unter der Abspielnadel läßt jede Schallplatte »knacken«. Die prinzipiell gleiche Ursache führt bei Diskettenlaufwerken zu Betriebsstörungen, denn ein schlechter Kontakt zwischen Magnetkopf und Speichermedium stört jeden Schreib- und jeden Lesevorgang. Dabei gibt es zwei Ursachen: Entweder sind der Kopf und/oder das Speichermedium verschmutzt oder die Diskette und der Magnetkopf werden nicht intensiv genug aneinandergedrückt. Doch schon bei den Verunreinigungen gibt es drei verschiedene Möglichkeiten: Entweder wurde eine verschmutzte Diskette ins Laufwerk eingeschoben und dadurch Schmutz am Schreib-/Lesekopf abgelagert, oder es handelt sich um – leider im Laufe vieler Betriebsstunden unvermeidbaren – Abrieb. In beiden Fällen läßt sich mit einer Reinigungsdiskette fast immer schnell und problemlos Abhilfe schaffen. Lediglich übermäßige und hartnäckige Verschmutzungen müssen vom Servicetechniker manuell entfernt werden.

Jedes Diskettenlaufwerk verfügt aber auch über ein sogenanntes Pressure Pad, zu deutsch: Andruckkissen. Dieses Kissen preßt das auf der Innenseite der Diskettenhülle befestigte Reinigungsvlies mit sanftem Druck gegen die Diskettenscheibe und sorgt so für die kontinuierliche Reinigung. Je preiswerter das Diskettenlaufwerk bei der Anschaffung war, desto größer ist leider die Wahrscheinlichkeit, daß jenes Andruckkissen aus minderwertigem Schaumstoff besteht. Dieser Schaumstoff kann schon nach einigen Betriebsmonaten ermüden oder schon alleine durch den natürlichen Alterungsprozeß verspröden. Wenn das Pressure Pad versagt, so unterbleibt die vom Konstrukteur vorgesehene Selbstreinigung.

Als Pannenhilfe bei Ursache 2 dient also zuerst die Reinigungsdiskette. Bei defektem Andruckkissen oder mangelhaftem Kontakt des Magnetkopfes auf dem Speichermedium kann meist nur der spezialisierte Mechaniker helfen.

Ursache Nummer 3:

Wie man sich bettet, so liegt man, und wie eine Diskette im Laufwerk eingespannt ist, so erfolgt die Datenverarbeitung – oder auch nicht. Bei defektem Antriebskonus sowie bei beschädigtem Disketteninnenloch oder nicht normgerecht ausgestanzter Antriebsöffnung kann die Diskette exzentrisch eingespannt sein. Oder laienhaft und prägnant gesagt: Sie eiert. Dadurch kann der Schreib-/Lesekopf die Datenspuren nicht verfolgen.

Mit zahlreichen Einspannversuchen kann es glücken, die Diskettenscheibe so zu fixieren, daß die Daten gelesen werden können. Traten die Probleme nur bei einer Diskette auf, so ist deren Einspannöffnung vermutlich die Ursache, und es hilft, die Daten auf eine andere Floppy zu kopieren und die beschädigte Scheibe auszusortieren. Tritt der Fehler häufig und bei allen verwendeten Datenträgern auf, so ist das Laufwerk der Schuldige und die Konusabmessungen und die Einspannkraft müssen in einer Servicewerkstatt überprüft werden. Prinzipiell kann der Anwender davon ausgehen, daß die Disketten renommierter Anbieter bei der Einspannöffnung der Norm entsprechen und somit keine Probleme verursachen.

Ursache Nummer 4:

Eine relativ seltene, wenn auch nicht völlig auszuschließende Ursache ist eine schlecht konzipierte Software, die das Initialisierungsmuster nur einmal schreibt. »Write retry«, zu deutsch: »Wiederholung des Schreibvorgangs«, lautet die Abhilfe, die allerdings nur derjenige schaffen kann, der in der Lage ist, das Initialisierungsprogramm zu verändern. Analog der »read retries« sollte auch bei den Schreibvorgängen eine mindestens dreifache Wiederholung vorgesehen sein.

Ursache Nummer 5:

Ebenso unwahrscheinlich, aber durchaus möglich ist das Deformieren der Diskettenscheibe durch Temperatureinwirkung.

Es empfiehlt sich unbedingt, alle am nächsten Tag zur Verarbeitung vorgesehenen Disketten schon am Vorabend in demselben Raum aufzubewahren wie die Hardware selbst. Wer bei eisiger Kälte und einem langen Fußmarsch seine neuerworbenen Disketten per Tragetasche nach Hause transportiert, sollte sich mindestens sechs Stunden gedulden, bevor er die Diskette ins Laufwerk schiebt.

Im allgemeinen empfiehlt es sich, bei Initialisierungsproblemen die gleiche Floppy auf einem zweiten Laufwerk zu versuchen. Läßt sie sich auf einem anderen Laufwerk initialisieren, so handelt es sich höchstwahrscheinlich um ein Hardwareproblem. Lassen sich außer der beanstandeten Diskette alle anderen Datenträger, auch älteren Datums, initialisieren, so ist mit großer Sicherheit die Diskette schuld. Die Wahrscheinlichkeit, daß ein renommierter Hersteller fehlerhafte Disketten ausliefert, ist allerdings gering. Sie hat eigenen Angaben nach ihre Disketten an die Betriebsbedingungen heutiger Systeme angepaßt. Der Einsatz immer einfacherer und billigerer Laufwerke und Controller stellt immer höhere Anforderungen an das Speichermedium. Magnetschichten mit höchstmöglicher Resolution bieten schärfere Leseimpulse und gesteigerte Datensicherheit. Flüchtige Fehler können durch Staub aus der Umgebung sowie statische Aufladung verursacht werden. Vliesmaterial hoher Flauschigkeit und Schmiegsamkeit sowie antistatisch ausgerüstete Hüllen verhindern dies jedoch in den meisten Fällen.

»Neuerer und schöpferische Geniusse können nicht in Schulen großgezogen werden. Es sind genau die Männer, die allem trotzen, was die Schule sie gelehrt hat.«

»Überall dort, wo das Schwein nicht mehr als das Symbol für feine Lebensart verehrt wird, hat es in letzter Zeit auffallend viele Erdbeben, Überschwemmungen und andere Naturkatastrophen gegeben.«

»Kinder, Betrunkene und die Vereinigten Staaten haben einen Schutzengel.«

»Um der Natur befehlen zu können, muß man ihr gehorchen.«

* = * = * Hallo Clubfreunde ! * = * = *

Neues aus der Programm-Bibliothek

Im vorletzten Heft (Nr.11, Jan.19...- - halt: hier muß ich mich selbst unterbrechen! einen Moment bitte, die Klammer wird gleich wieder zugemacht! Also: Zunächst für Jens ein großes "Dankeschön!" dafür, daß er von Nr.11 an die Seiten unserer INFOs auf meine Bitte hin auch mit Monats- und Jahresangabe versieht! Ich glaube, daß das all denen eine gute Hilfe ist, die die Hefte, schön geordnet, sammeln und später einmal etwas nachschauen wollen: denn die Angabe der Ausgabe-Nummer auf der farbigen Titelseite sagte nichts darüber aus, in welchem Zeitabschnitt man sich beim Herumblättern gerade befand! Das kann aus rein historisch-logischen Gründen von Wichtigkeit sein. * Nun muß diese Neuerung aber wohl sehr anstrengend gewesen sein; jedenfalls verließen unseren Chefredakteur bei der Umstellung auf das neue Jahr offenbar die Kräfte... Nichts für ungut; das kann vorkommen! Also bitte, wenn Ihr Zeit und Lust habt: ändert die Jahreszahlen in Heft 11 in 1986 um! Sonst könnte es einmal passieren, daß ein außenstehender Leser dieses Heftes (vorher Mitglieder- mit Anschriften-Verzeichnis abtrennen) konstatiert: "Donnerwetter, die sind ja schon ein ganzes Jahr dem Stand der Technik voraus! " - was andererseits allerdings auch als Werbe-Gag gedeutet werden könnte... Nun darf ich die Klammer schließen, ja? *)

Also noch einmal: In Heft 11 (Jan.1986) hatte das letzte Programm die Nummer 298. Folgerichtig geht es jetzt mit Nr.299 weiter. Sieh an: in der Tat, die "300" wurde überschritten - die Diskothek reicht inzwischen bis "326"! Wesentlich haben hierzu Josef Konrad und Holger May beigetragen.

Nehmt Euch ein Beispiel daran!

Die Programm-Auflistung erscheint jetzt in etwas verändertem Gewand. Der Programm-Name ist durch doppelte Unterstreichung hervorgehoben.*) Ich habe ferner die komplette Dateimaske ausgedruckt, wie sie bei der Bearbeitung auf meinem Schildbirm (hier vertippe ich mich jedesmal; aber ich finde es ganz sinnig: haben wir denn nicht alle fast täglich ein paar Stunden lang so ein "Schild" (aus Glas statt Holz) vor der Birne?) vor meiner Birne erscheint.

Hier findet Ihr alles Wissenswerte: die Programm-Gattung (drei Buchstaben); die Programm-Nummer (4stellig); die Anzahl Teilprogramme (wichtig bei einem "Programm-Paket"); die Benotung (1=schwach bis 5=sehr gut); das Eingangsdatum (amerik.Art - leider -); die Eignung für welche Computer; die Herkunft (Quelle kann z.B. eine Zeitschrift sein, aus der abgetippt wurde); den Verfasser (der i.d.R. bekannt ist); den "Eintipper" (Fleiß muß anerkannt werden); einen möglichst vielsagenden Kommentar und die Nummer der Diskette * Alles klar?

Eine Bitte: *Schreibt mir, wenn eine Angabe einmal falsch sein sollte; z.B. der Verfasser-Name oder oder... die Korrektur erscheint dann in der nächsten INFO.*

* * * Bei Programm-Bestellung bitte angeben: * * *
Programm-Name oder -Nummer / Anzahl Programmteile
und Disknummer!

Wer nur (z.B.) "SPL" angibt, bekommt von mir sämtliche Spiele, wer nur "UTL" angibt erhält alle Utilities usw.

Bitte Disketten mit Angabe der Drive-Parameter einsenden! Rückporto ist erwünscht. * Die Weitergabe von Programmen an Nicht-Mitglieder ist verboten!

Ergötzt Euch nun an den Programmen!
Es kam ein gutes Teil zusammen.
Und "spendet" für die Diskothek,
holt Euch auch mal was von ihr weg,
denn dieses ist ihr Lebenszweck !

Euer KaJot
(Am Mönchgarten 28, 6940 WEINHEIM)

*) Die Programme sind innerhalb jeder Programmart alphabetisch nach Namen geordnet!


```
MONDUHR/BAS  DIV ProgrNr 0321 Teile: 01 Note 1        2 /28/86
============      f.Comp ALLE            Quelle:
Verfasser MAY, HOLGER            eingetippt von M., H.
Konstellation von Sonne/Erde/Mond zum eingegebenen Zeitpunkt
(Die entsprechende Mondphase (zu- oder abnehmend, Halb-,
Voll- oder Neumond) muss daraufhin genau ueber-[Disk-Nr. 9
legt werden.

AKTIEN2/BAS  KFM ProgrNr 0324 Teile: 52 Note 4        0 /0 /0
============      f.Comp ALLE            Quelle:
Verfasser Boeckling u. Wagner   eingetippt von
Zur Verwaltung der Aktien zahlreicher Gesellschaften. Mit
guten graf. Darstellungen. Abs. u. prozentuale Aenderung der
Kurse usw. * Das Paket besteht aus 52 Files     Disk-Nr. 7

OELDATEI/BAS KFM ProgrNr 0320 Teile: 01 Note 2        2 /28/86
============      f.Comp ALLE            Quelle: COMPPERSOENL
Verfasser                        eingetippt von MAY, HOLGER
Ueberwachung der Heizoelkosten bei monatlicher Tankablesung

   Disk-Nr. 9

SEPTEMM/BAS  MTH ProgrNr 0301 Teile: 01 Note 3        2 /14/86
============      f.Comp ALLE            Quelle: LOAD 80
Verfasser                        eingetippt von
Loesung einer Differentialgleichung (die in der zweiten Zeile
einzugeben ist). Ohne Dokumentation. Differentialquotient
nicht gekennzeichnet.                         Disk-Nr. 9

BERGSTEI/BAS SPL ProgrNr 0323 Teile: 01 Note 1        2 /8 /86
============      f.Comp ALLE            Quelle:
Verfasser MAY, HOLGER            eingetippt von M., H.
Zwei Spieler (oder ein Spieler und der Computer) klettern an
einer Steilwand hoch, die offenbar nie endet.
   Disk-Nr. 9

HOTDOG/BAS   SPL ProgrNr 0302 Teile: 01 Note 1        2 /14/86
============      f.Comp ALLE            Quelle: LOAD 80
Verfasser                        eingetippt von
Im Prinzip das "NIMM"-Spiel. Es werden schmale Scheiben von
einer Wurst abgeschnitten. (Primitiv.)
   Disk-Nr. 6

KUNGFU/BAS   SPL ProgrNr 0303 Teile: 01 Note 2        2 /14/86
============      f.Comp ALLE            Quelle: LOAD 80
Verfasser                        eingetippt von
Suchen einer Gefangenen in chinesischen Tempeln unter Lebens-
gefahr. (Unzureichende Dokumentation.)
   Disk-Nr. 9

POKER/BAS    SPL ProgrNr 0304 Teile: 01 Note 1        2 /14/86
============      f.Comp ALLE            Quelle: LOAD 80
Verfasser                        eingetippt von
Mit dem Computer Karten spielen

   Disk-Nr. 9

RAMMER/BAS   SPL ProgrNr 0305 Teile: 01 Note 2        2 /14/86
============      f.Comp ALLE            Quelle: LOAD 80
Verfasser                        eingetippt von
Raketenabwehr-Spiel (Reaktionsspiel)

   Disk-Nr. 6
```

ROLLADEN/BAS SPL ProgrNr 0322 Teile: 01 Note 1 2 /28/86
============ f.Comp Quelle:
Verfasser MAY, HOLGER eingetippt von M., H.
Von drei grossen Rolladen kann jeweils einer hochgezogen werden. Lassen Sie sich ueberraschen, was Sie dann sehen!
Disk-Nr. 9

SCHNECKE/BAS SPL ProgrNr 0306 Teile: 01 Note 1 2 /14/86
============ f.Comp ALLE Quelle: LOAD 80
Verfasser eingetippt von
Der Cursor zieht eine Linie hinter sich her. Sobald diese sich selbst beruehrt, ertoent ein Signal zur Niederlage. (Ziemlich dumm) Disk-Nr. 6

SECHSER/BAS SPL ProgrNr 0318 Teile: 01 Note 2 2 /28/86
============ f.Comp ALLE Quelle:
Verfasser MAY, HOLGER eingetippt von M., H.
Wuerfelspiel. Bis zu 6 Teilnehmer einschl. Computer.

Disk-Nr. 9

SKAT/BAS SPL ProgrNr 0319 Teile: 01 Note 2 2 /28/86
============ f.Comp ALLE Quelle: COMPPERSOENL
Verfasser eingetippt von MAY, HOLGER
Man spielt Skat gegen den Computer, welcher zwei Spieler vertritt.
Disk-Nr. 9

SQUAROTA/BAS SPL ProgrNr 0307 Teile: 01 Note 2 2 /14/86
============ f.Comp ALLE Quelle: LOAD 80
Verfasser eingetippt von
Das bekannte Puzzle: Plaettchen muessen durch ueberlegtes Verschieben in die richtige Reihenfolge gebracht werden.
Disk-Nr. 9

STARMARK/BAS SPL ProgrNr 0308 Teile: 01 Note 2 2 /14/86
============ f.Comp ALLE Quelle: LOAD 80
Verfasser eingetippt von
Steinchen in einem Sternmuster muessen durch Ueberspringen beseitigt werden (fuer einen Spieler allein).
Disk-Nr. 9

STARPILO/BAS SPL ProgrNr 0310 Teile: 01 Note 4 2 /14/86
============ f.Comp ALLE Quelle: LOAD 80
Verfasser eingetippt von
Man fliegt durch einen Asteroidenschwarm und muss diesem ausweichen. 9 Schwierigkeitsgrade. Guter Reaktionstest! Mit Ton.
Disk-Nr. 9

THREMARK/BAS SPL ProgrNr 0309 Teile: 01 Note 3 2 /14/86
============ f.Comp ALLE Quelle: LOAD 80
Verfasser eingetippt von
Steinchen liegen in einem vorgegebenen Muster. Durch ueber=legtes Ueberspringen muessen moeglichst viele beseitigt werden (fuer einen Spieler allein). Disk-Nr. 9

COPYILF/BCD UTL ProgrNr 0311 Teile: 02 Note 4 2 /14/86
============ f.Comp ALLE Quelle:
Verfasser OBERMANN, Hartmut eingetippt von H.OBERMANN
Programm zum komfortablen Erstellen und Kopieren von ILF-Files. Compiliert mit BASCOM.(BASIC- und CMD-Version.)
Disk-Nr. 10

COUNTER/CSR UTL ProgrNr 0300 Teile: 02 Note 2 2 /14/86
============ f.Comp ALLE Quelle: 80MICRO 1/86
Verfasser eingetippt von SCHAEFER,WALTER
zeigt die Benutzungshaeufigkeit der Zeilen eines vorgewaehlten Programmbereichs an. Mit EDTASM-Source-Code.
Disk-Nr. 10

FORMLIST/CZX UTL ProgrNr 0312 Teile: 02 Note 3 2 /14/86
============ f.Comp ALLE Quelle:
Verfasser KONRAD, Josef eingetippt von JOSEF KONRAD
erstellt /JCL-Files zum Listen von BASIC-Programmen (Aufruf: "FORMLIST, Programm-Name"). Mit ZBASIC-Source-Code.
Disk-Nr. 10

LIESDIR/BAS UTL ProgrNr 0299 Teile: 02 Note 3 12/30/85
============ f.Comp ALLE Quelle:
Verfasser MUEHLENBEIN, KAJOT eingetippt von MB.KJ.
Numeriert, druckt und speichert die Filenamen einer Disk (incl.unsichtb.) nebst 3 Zl.Kommentar je File, der spaeter wieder aufgerufen u.geaendert werden kann! Disk-Nr. 6

MORE/CZX UTL ProgrNr 0313 Teile: 02 Note 3 2 /14/86
============ f.Comp Quelle:
Verfasser KONRAD, Josef eingetippt von JOSEF KONRAD
komfortables Listen von ASCII-Files (Aufruf: "MORE,Programm-Name".) Mit ZBASIC-Source-Code.
Disk-Nr. 10

NUM/CZX UTL ProgrNr 0314 Teile: 02 Note 3 2 /14/86
============ f.Comp ALLE Quelle:
Verfasser KONRAD, Josef eingetippt von JOSEF KONRAD
numeriert ASCII-Files um (Aufruf: "NUM,Programmname"). Mit ZBASIC-Source-Code.
Disk-Nr. 10

PASFORMG/CPC UTL ProgrNr 0315 Teile: 02 Note 3 2 /14/86
============ f.Comp ALLE Quelle:
Verfasser KONRAD, Josef eingetippt von JOSEF KONRAD
wandelt alle PASCAL-Worte in klein geschriebenen PASCAL-Source-Codes in Gross-Schrift um. Sonstige Texte (Variablen usw.) bleiben klein. Erhoeht Lesbarkeit! Disk-Nr. 10

TEILER/CZX UTL ProgrNr 0316 Teile: 02 Note 3 2 /14/86
============ f.Comp ALLE Quelle:
Verfasser KONRAD, Josef eingetippt von JOSEF KONRAD
teilt ASCII-Files in mehrere Teile (Aufruf:"TEILER,Programm=name"). Mit ZBASIC-Source-Code.
Disk-Nr. 10

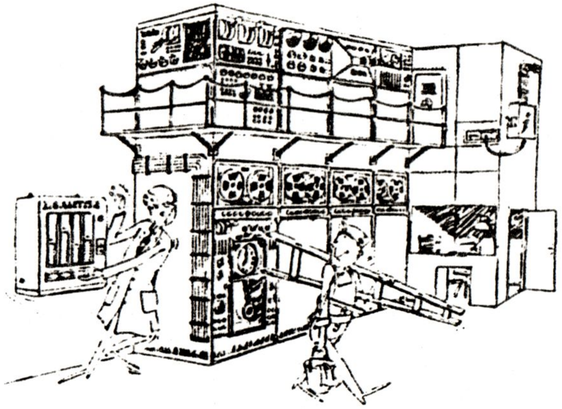
VIDEO/BAS DAT ProgrNr 0325 Teile: 01 Note 3 4 /2 /86
========== f.Comp alle Quelle: COMPERS10/83
Verfasser THIEME, Roland eingetippt von WUCHERER, Juerg.
erleichtert die Verwaltung von Video- bzw. Musik-Kassetten. Die Eintraege koennen auch auf dem Drucker ausgegeben werden. (Programm nicht getestet.) Kurzanleitung vorh. Disk-Nr. 9

ZEILZ/CZX UTL ProgrNr 0317 Teile: 02 Note 3 2 /14/86
============ f.Comp ALLE Quelle:
Verfasser KONRAD, Josef eingetippt von JOSEF KONRAD
zaehlt die Zeilen eines ASCII-Files (Aufruf: "ZEILZ,Programm-Name"). Mit ZBASIC-Source-Code.
Disk-Nr. 10

disko REPORT --

Neuerscheinungen in der Programm-Bibliothek seit INFO Nr.11

```
ZEUS/CMD     UTL ProgrNr 0326 Teile: 01 Note 0         4 /21/86
============        f.Comp ALLE              Quelle: NACH GOETHE
Verfasser KaJot                      eingetippt von KaJot
Ein delikater Assembler, der hoehere Weihen erteilt.
Du brauchst nur "ZEUS" zu rufen: schon rollt er vor Deinen
auf dem Bildschirm ab und ermahnt Dich!           Disk-Nr. 10
```



»Hallo, Sie – wie funktioniert dieser komplizierte Apparat?«

disko REPORT --

* * * **NACHTRAEGE** * * * (Hier hatte der Bibliothekar geschlafen)

```
MASTMIND/BAS SPL ProgrNr 0283 Teile: 01 Note 3        0 /0 /0
============        f.Comp ALLE              Quelle:
Verfasser SCHAEFER, Walter        eingetippt von W.Sch.
Das bekannte Denkspiel "MASTERMIND"

   Disk-Nr. 5
```

```
NIMM1/BAS    SPL ProgrNr 0282 Teile: 01 Note 3        0 /0 /0
============        f.Comp ALLE              Quelle:
Verfasser SCHAEFER, WALTER        eingetippt von SCH., W.
Das bekannte NIMM-Spiel mit interessanten Verbesserungen von
Walter Schaefer.
   Disk-Nr. 5
```

Nachtrag zum Programm-Katalog auf Seite 30!
Bitte dort einkleben!

Neue Eingaenge am 5.5.86 von Herbert Albers

```
HAMURABI/BAS SPL ProgrNr 0327 Teile: 01 Note 2         5 /6 /86
============       f.Comp alle             Quelle: GENIE-DATA
Verfasser                       eingetippt von ALBERS, Herbert
Ein Spiel fuer Bauern u. Kaufleute zugleich. Das vorhandene
Guthaben ist durch Einsatz geeigneter Mittel zu vermehren.
                                                    Disk-Nr. 9
```

```
HANGMAN2/BAS SPL ProgrNr 0328 Teile: 01 Note 4         5 /6 /86
============       f.Comp alle             Quelle:
Verfasser ALBERS, Herbert       eingetippt von ALBERS, H.
Ein bekanntes Spiel. Es muessen Begriffe aus dem Bereich des
Fussballs gefunden werden. Mit Auswertung und Grafik.
                                                    Disk-Nr. 9
```

```
TABELLEN/BAS DIV ProgrNr 0329 Teile: 01 Note 3         5 /6 /86
============       f.Comp alle             Quelle: COMP.PERS.
Verfasser erw.,umgeschrieben & eingetippt von Albers, Herbert
Erstellung von Tabellen aus dem Bereich des Sports. Sie
koennen auf Disk gespeichert sowie fuer andere Belange
umgeschrieben werden.                               Disk-Nr. 9
```

```
EINKST85/BAS KFM ProgrNr 0330 Teile: 01 Note 3         5 /6 /86
============       f.Comp alle             Quelle: COMP.PERS.
Verfasser                       eingetippt von Albers, Herbert
Berechnung der Einkommensteuer nach dem Splittingtarif.
Mit Druckroutine.

                                                    Disk-Nr. 9
```

```
BUCHHLTG/BAS KFM ProgrNr 0331 Teile: 01 Note 3         5 /6 /86
============       f.Comp alle             Quelle: COMP.PERS.
Verfasser ---umgeschrieben und eingetippt von Albers, Herbert
Komplette Konten- und Belegbuchfuehrung mit Jahresabschluss.

                                                    Disk-Nr. 9
```

```
TILGUNG/BAS  KFM ProgrNr 0332 Teile: 01 Note 3         5 /6 /86
============       f.Comp alle             Quelle: COMP.PERS.
Verfasser ---umgeschrieben und eingetippt von Albers, Herbert
Berechnung eines Tilgungsplanes fuer Darlehen oder Hypothek.

                                                    Disk-Nr. 9
```

```
STATISTK/BAS MTH ProgrNr 0333 Teile: 01 Note 3         5 /6 /86
============       f.Comp alle             Quelle:
Verfasser ALBERS, Herbert       eingetippt von ALBERS, Herbert
Statistikprogramm zur Auswertung von Mess- od.Eichreihen
mit tabellarischer Druckroutine.
                                                    Disk-Nr. 9
```

```
BALKEN01/BAS DIV ProgrNr 0334 Teile: 01 Note 3         5 /6 /86
============       f.Comp alle             Quelle:
Verfasser ALBERS, Herbert       eingetippt von Albers, H.
Erstellung individueller Balkengrafiken mit Grafik-Hardcopy.

                                                    Disk-Nr. 9
```

- I M P R E S S U M -

1. Vorsitzende
Peter STEVENS
Postfach 56
4600 Dortmund 1
☎ 0231 /593883

2. Vorsitzende
Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

HardwareKoordinator
Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar
Klaus-Jürgen MÜHLENBEIN
Am Mönchgarten 28
6940 Weinheim -Lü.
☎ 06201 /55052

Redaktion
Jens NEUEDER
Panoramastraße 21
7170 Michelbach /Bilz
☎ 0791 /42077

Redakteure dieser Ausgabe:

Herbert Albers	Ulrich Böckling
Dieter Kasper	Eckehard Kuhn
Klaus-J. Mühlenbein	Jens Neueder
Hartmut Obermann	Hans Raggan
Bernd Retzlaff	Gerald Schröder
Arnulf Sopp	Peter Spieß
Peter Stevens	

sowie Artikel aus: c't, 80 Micro und WDR Computer-Club

Bankverbindung des CLUB 80
Zur Zeit kann wegen des Vorstandswechsels noch keine neue Bankverbindung genannt werden. Bitte wendet Euch einstweilen an Peter Stevens.

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

heute möchte die Gelegenheit nutzen, um nochmals auf ein paar redaktionelle Dinge hinzuweisen.

Zum Einen geht es wieder mal um die Farbbänder, welche -anfangs so schön schwarz und später in dezentem grau- den Ausdruck Eurer Werke für das INFO sicherstellen. Bitte seit so nett und verwendet neuere Farbbänder für den Ausdruck Eures Info -Beitrags. Ihr könnt dadurch den Wert Eures Beitrags (durch bessere Leserlichkeit) erhöhen und die Qualität des Club-INFO's verbessern. Euer "Lieblingsfarbband" könnt Ihr ja noch weiterhin für andere Aufgaben verwenden.
Zum Anderen habe ich teilweise noch Probleme mit dem Format Eurer Beiträge. Ich möchte Euch an das Beiblatt "Infoform" von Club -INFO 11 erinnnern. Bitte haltet die Empfehlungen ein

Ein weiteres Problem ist für mich, daß ich momentan aus persönlichen Gründen sehr wenig Zeit für Clubarbeiten habe. Aufgrund der vielen Nachbestellungen von alten Club-INFO's verzögert sich dadurch die Auslieferung der "Oldies". Ich hoffe, Ihr habt dafür Verständnis. Ich bin bemüht diese Redaktionsschuld an Euch trotzdem schnellstmöglich zu begleichen.

Nun noch ein Tipp an diejenigen, die Beiträge -so günstig wie möglich- an die Redaktion senden möchten. Verwendet dazu die Versandtform D r u c k s a c h e. Ihr braucht nicht unbedingt einen Brief beizulegen. Eine kurze Erklärung über den Artikel sowie Anmerkungen zu anderen Themen sind -wenn sie gedruckt und ohne persönliche Vermerke- jederzeit erlaubt. Auch sollte der Beitrag dann in einem "offenen" Kuvert verschickt werden. Informiert Euch mal bei der Post darüber.

Als letztes möchte ich Euch nochmals an den anhängenden Fragebogen erinnern. Bitte füllt ihn sorgfältig aus und schickt ihn mir zur Auswertung zu. Eure Angaben bestimmen den weiteren Werdegang unseres Clubs mit.

Viel Spaß mit Eurer 13. INFO's wünsche ich Euch allen.

Bis zum nächsten INFO verabschiedet sich

Euer

J. Neueder

JENS NEUEDER

Panoramastr. 21
7178 Michelbach /Bilz Tel.: 0791 / 42877

Michelbach, den 24.05.1986

Helmut Bernhardt

Hafenstraße 7
2305 Heikendorf

Hallo **Bernhardt,**

ich habe Dich natürlich nicht vergessen. Leider war ich aber privat und mit der neuesten Club-INFO so eingespannt, daß ich für "Sonderaktionen" wenig Zeit übrig hatte. Ich hoffe, Du hast Verständnis dafür.

Fast die Hälfte der von Euch nachbestellten INFO's habe ich nun doch schon fertig. Ich sende die für Dich, wie Du sicher schon gemerkt hast, anbei mit. Der Rest folgt ... je nach Zeit aber schnellst möglich.

Viel Grüße aus Michelbach.
Bis bald -- oder bis zum nächsten INFO

Jens

******* BOERSE NACHTRAG ******

GENIE IIIS (6 MONATE ALT)
INCL. CPM2.22
MONITOR GRUEN
EPSON MX80FT,GRAFTRAX
VHB
GERRALD DREYER 06121/508218

Name	Vorname	Straße	PlZ	Stadt	Telefon
Alber	Herbert	Alemannenstr. 20	7732	Niedereschach	07721 /7102
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241967
Boecker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218
Drowälder	Bernd	Buchentalsweg 8	4939	Steinheim	05233 /4320
Frink	Thomas	An der Schleifmühle 2	5042	Erftstadt	02235 /76255
Fröhlich	Burghard	Brandströstraße 11	4350	Recklinghausen	02361 /63416
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573
Hartel	Eberhard	Neenstetter Straße 20	7901	Breitingen	07340 /7281
Heile	Heinz-Dieter	Blankensteiner Straße 13	4320	Hattingen	02324 /28458
Held	Manfred	Stirnerstr. 22	8835	Pleinfeld	09144 /6563
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024
Hill	Peter	Eckstraße 36	6750	Kaiserslautern 31	0631 /54782
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630
Koenig	Hans J.	Hebbelstr. 25	2080	Pinneberg	04101 /209444
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach	0911 /506051
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lü.	06201 /55052
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083
Neueder	Jens	Panoramastr. 21	7178	Michelbach/Bilz	0791 /42877 (dienstl.44-667)
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418
Preuss	Lothar	Lautshof 13	2940	Wilhelmshaven	04421 /84247 (dienstl. 804-1)
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611
Rank	Heinrich	Fruehlingstr. 2	8080	Fuerstenfeldbruck	08141 /3791
Rentsch	Richart	Postfach 226	7128	Lauffen am Neckar	
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551
Schaefer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631
Schneider	Manfred	Rheinkasseler Weg 11	5000	Koeln 71	0221 /707044
Schrewe	Christian	Fliederweg 32	4000	Duesseldorf 31	0203 /740897
Schroeder	Gerald	Am Schuetzenplatz 14	2105	Seevetal 1	04105 /2602
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284
Sopp	Arnulf	Wakenitzstr. 8	2400	Luebeck 1	0451 /791926
Spiess	Peter	Trugenhofenerstr. 27	8859	Rennertshofen	08434 /454
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324
Voigtlaender	Holm	Haselnussweg 30	6940	Weinheim	06201 /65241
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285
Wagner	Guenther	Gartenstr. 4	8201	Neubeuern	08035 /3361
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194
Wucherer	Jürgen	Brauneggerstraße 14	7750	Konstanz	07531 /29145
Zwickel	Walter	Lengfelden 123	A-5101	Bergheim	0043662/51130

unsere Neuen :

Helmut Bernhardt
Thomas Frink
Eberhard Hartel
Heinz-Dieter Heile
Richart Rentsch

verlassen hat uns :

Alexander Wagner 15.03.86

Stand: Mai 1986

Bitte überprüft Eure Anschrift und meldet eventuelle Adressenänderungen an die Redaktion.

CLUB 80

Clubinfo
der
TANDY -
GENIE -
und KOMTEK -
ANWENDER

14. AUSGABE

Kontaktadresse : CLUB 80 / Peter STEVENS / Postfach 56 / 4600 Dortmund 1

Tel.: 0231 / 593883

Inhaltsverzeichnis

Seite:

Clubinternes

1. Vorstandssitzung . . . . 01 - 02
Nochmals Fragebogen 03
Vorstellung eines Mitgliedes 04
Terminkalender 05

Software

EDIT-Befehl 06
Programmierwettbewerb 08
Mein 1. ASSEMBLER-Programm . . . . . 09 - 16
ASSEMBLER-Routinen . . . . 17 - 20
Mehr Fernsehen für's Geld . . . . . . 21 - 22
ASCII-Tab 23
HRG-Bildübertragung . . . . 24 - 28
Grafik-Programme . . . . 29 - 38
Wenn die Uhr mal stört . . . . 39 - 42
Scripsit auf dem "alten" Genie . . . 43 - 44
Scripsit-Tabellen . . . . 45 - 46
Freiräume im Scripsit . . . . 47 - 48
Textumleitung . . . . 49 - 51
Nachtrag zu "Sondertasten" . . . . . 53 - 54
Unterschiede . . . . 55 - 56
Knöpfe . . . . 57 - 58
Aufgestiegen . . . . 59 - 60
Tips M4/4p . . . . 61 - 62
Root-ROM des 4p . . . . 63 - 65
HRG für Model 3/4/4p . . . . 67 - 68
DOS - (fast) ohne Floppy . . . . . . 69 - 75
Rechenzeitverkürzung . . . . 77 - 85

Hardware

Floppytrick's . . . . 87 - 88
HRG-1b . . . . 89 - 90
256k RAM für Z80 . . . . 91 - 99
CLUB 80 ECB-Bus-System . . . . 101 - 103

Seite:

Börse

Wer hat was ???--Wer sucht was ??? 105 - 106
Wer kann weiterhelfen . . . . 107 - 108

Sonstiges

Buchbesprechung . . . . 109 - 110

Programmbibliothek

Einleitung 111
Neue Programme . . . . 112 - 113

Die letzten Seiten

Impressum 114
Schluß 115
Clubmitgliederadressen . . . . am INFO-Ende
Anmeldung ECB-Bus . . . . . . am INFO-Ende

1. VORSTANDSSITZUNG

am Sonntag, 29. 06. 1986 bei KaJot
Anwesende waren: Eckehard Kuhn
Klaus-J. Mühlenbein
Jens Neueder
Hartmut Obermann

Mit dieser Zusammenfassung über die 1. Vorstandssitzung möchte der Vorstand die Mitglieder über seine laufende Arbeit informieren.

Priorität in unserer Tagesordnung hatte natürlich unsere Fragebogenauswertung. Leider kamen von 53 Fragebogen -nach sechs Wochen !- nur insgesammt 19 zur Auswertung an uns zurück. Die Auswertung der Fragebogen wurde aus diesem Grund verschoben.
Die aus der Fragebogenauswertung resultierenden Punkte wie:

- -- Mitgliederaktivierung
- -- neuer Mitgliederstand (Geräte, Interessen usw.)
- -- Interessengruppen

mußten aufgrund der geringen Daten auch unter den Tisch fallen.
Das Thema "Fragebogen" wird wegen seiner Wichtigkeit an anderer Stelle im INFO nochmals aufgegriffen.

Weiterhin sprachen wir über die Mitgliederwerbung und haben uns für folgendes entschlossen.

1. Nochmalige Puplikation unseres Clubs in den einschlägigen Zeitschriften, da uns dies durch die "Vorstandsänderung" nichts kostet.
2. Interessenten an unserem Club bekommen ein Vorstellungsblatt (Darstellung unseres Clubs und Clubsatzung), sowie ein Anmeldeformular (wie der Fragebogen vom letzten INFO) zugesandt.

Weiterhin wird auf die Möglichkeit hingewiesen, ein Probeinfo für DM 5,-- zu beziehen. Wer Mitglieder werben möchte oder Interessierte an unserem Club kennt, melde sich bitte wegen der nötigen Formulare bei der Redaktion.

Desweiteren hatten wir die Möglichkeit ein Public-Domain-Software-Paket in die "Diskothek" zu übernehmen. Es sind über 200 neue Programme aus den USA. Ihr werdet im nächsten INFO Ausführliches darüber erfahren.

Alsdann beschlossen wir wieder ein Sonderheft herauszubringen.
Wir haben dazu mehrere Themen in der engeren Wahl:
Modem-Sonderheft, Grafik M4/4p und ROM-Sonderheft
Auch möchten wir Kurzbefehlslisten über BASIC und DOS erstellen.
Zu beiden "Aktionen" ist Eure Mitarbeit erwünscht. Bitte meldet Euch dazu bei der Redaktion zur Koordinierung.

In Zukunft möchten wir auch in der INFO einen Veranstaltungskalender einbauen. Als Grundstock soll uns "TEDAS" vom Franzisverlag dienen. In dieser Mailbox sind sicher die aktuellsten Meldungen über die Computerei vorhanden. Wer zudem noch Termine bekanntzugeben hat (Regionaltreffen usw.) melde sich bei der Redaktion.

Als nächstes diskutierten wir darüber, mit einem Tandy-Club in den USA in Briefwechsel zu treten. Die Entscheidung dazu fiel positiv aus, da wir uns davon einige Vorteile erhoffen.

Weiterhin wird Hartmut einen Hardware-Grundlagen-Kurs abhalten. Wann, wo und wie wird bekannt gegeben.

Als vorletzten Punkt kam nun noch unser neues Club-Konto an die Reihe. Die neuen Daten wie folgt:

Postgirokonto	Peter STEVENS
	Sonderkonto Computerclub
Konto-Nummer	285 491 - 465
Postgiroamt	Dortmund
BLZ	440 100 46

Da sich die "e.V. -Eintragung" noch etwas verzögert, wurde diese Möglichkeit gewählt. Nach der "e.V. -Eintragung" wird der Name des Inhabers in Club 80 e.V. geändert.

Abschließend entschlossen wir uns, innerhalb des Vorstandes, alle 14 Tage ein Rundschreiben kursieren zu lassen, damit neueste Informationen diskutiert werden können. Solltet Ihr also in Zukunft wichtige Neuigkeiten für die Mitglieder oder den Club betreffend haben, so ist jetzt die Möglichkeit gegeben, brandneue Informationen bei einem der Vorstandsmitglieder abzusetzen.

Die Redaktion.

F R A G E B O G E N

Wie schon vorher im INFO angeführt soll hier nochmals die Rede auf den Fragebogen kommen.

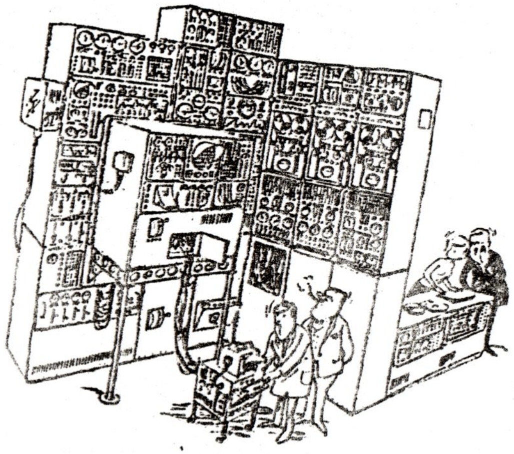
Der Fragebogen ist dazu da, uns -Euren Vorstand- über Eure Wünsche, Interessen und Gerätekonfigurationen zu informieren. Nur durch die von uns gewünschten Informationen ist es uns erst möglich Euch ein aktuelles, interessantes und vor allem ansprechendes INFO zu bringen - deshalb der Fragebogen. Leider hat uns die Mehrzahl der Mitglieder im Stich gelassen. Nach <u>sechs Wochen !!!</u> kamen nur 19 Fragebögen von 53, die mit dem INFO verschickt wurden, zurück.

Ist es für Euch eigentlich so uninteressant, was in <u>Eurer Club-INFO</u> steht? Sicher nicht, oder ...?

Wir denken, daß diese Fragebogenaktion in Eurem Sinn ist und von allgemeinem Nutzen. Aus diesem Grund haben wir die Frist verlängert. Bitte seit so freundlich und sendet den Fragebogen umgehend zur Auswertung an die Redaktion.

In der Hoffnung, nun auch die fehlenden Fragebogen zurückzubekommen
Euer Vorstand

P S.: Wer seinen Fragebogen unauffindbar aufgeräumt haben sollte, kann von der Redaktion einen weiteren Fragebogen bekommen - Anruf genügt.



»Typisch unser Chef –
er läßt den Computer von einem Lügendetektor testen!«

Liebe Clubfreunde,

ich bin seit Februar diesen Jahres Mitglied im Club und moechte mich ebenso wie andere vor mir kurz vorstellen. Ich benutze ein Video-Genie System mit zwei Laufwerken, HRG-1B Grafik-Karte, 64K RAM und einer einfachen, von mir fuer den Drucker eingebauten RS-232c Schnittstelle. Als Drucker verwende ich einen H14 der Firma Heathkit. Er ist bereits reichlich aus der Mode, aber ein neuer kostet auch seine Mark und nur zum Papierschwaerzen erfuellt er seine Dienste bisher noch ausreichend. Mein bevorzugtes und einziges Betriebssystem ist Newdos80.

Was meine Aktivitaeten in Sachen computern betrifft, so erstelle ich meine kleinen Programm in Assembler, BASIC oder auch einmal in FORTRAN. Z. Zt. plage ich mich mit der Anpassung des RPNL-Compilers des Herrn Wostrack ab. Sollte sich Jemand im Club hier ebenfals schon einmal versucht haben, bin ich an einem Informationsaustausch interessiert. Der Compiler erscheint mir durchaus brauchbar, ermoeglicht er doch eine maschinennahe Programmierung in Assembler als auch die problembezogene Programmerstellung. Obwohl FORTH-Aehnlich, ist RPNL bei weitem nicht so kaotisch, da die Sprachstruktur an PASCAL angelehnt ist.

Tschuess

Kurt Müller

-- Termine -- Termine -- Termine --

An dieser Stelle unseres CLUB-INFO's möchten wir in Zukunft einen Terminkalender aufbauen.

Wir haben vor, hier über allgemeine Veranstaltungen der Computerei

-- Börsen, Treffen und Messen --

sowie über spezielle TRS-80-Termine zu berichten.
Auch die Termine unseres Clubs sollen hier bekanntgegeben werden.
Wir dachten dabei an:

Regionale Treffen, Stichtage von Aktionen usw.

Ihr als Mitglieder seit nun aufgefordert hierbei mitzuwirken.

Die ersten Termine:

Ende der ECB-Bus-Anmeldung	20. August 1986
Ende der Fragebogenaktion	29. August 1986
nächster Redaktionschluß	29. August 1986

Nachdem wir bei unserem regionalen Treffen bei Hans feststellen mußten, daß die EDIT-Möglichkeiten wohl wegen fehlender Handbücher o.ä. bei einigen Clubkameraden nicht bekannt sind, nachfolgende Kurzbeschreibung:

Editieren einer Programmzeile

In die gewünschte Programmzeile kommt man durch:

1. EDIT Programmzeilen-Nummer
2. EDIT . (die zuletzt bearbeitete Zeile wird aufgerufen)
3. E Programmzeilen-Nummer (nur in einigen BASIC-Versionen)
4. , (die zuletzt bearbeitete Zeile wird aufgerufen - nur in einigen BASIC-Versionen

Unterbefehle und Funktionstasten im EDIT-Modus

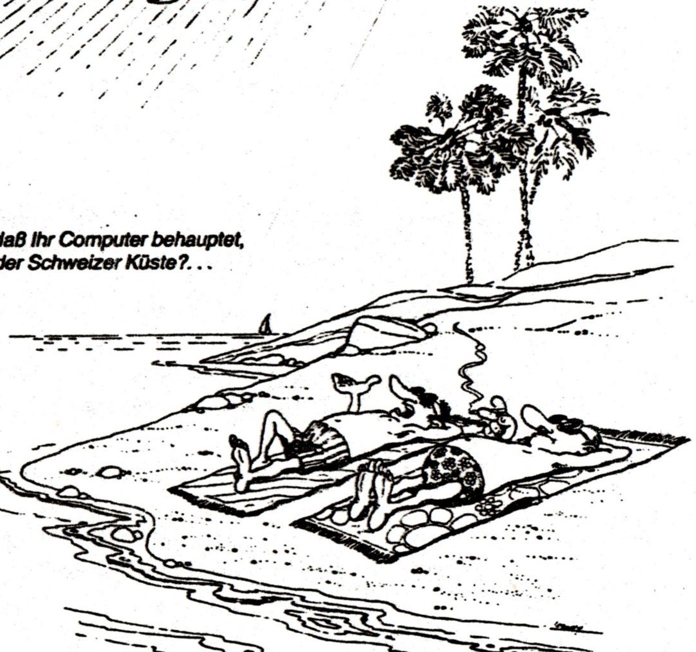
(HP = Hochpfeil / LP = Linkspfeil)

ENTER	Editieren beenden und in den Befehlsmodus zurückkehren
SHIFT HP	Unterbefehl verlassen und im EDIT-Modus bleiben
n Leert.	Den Cursor um n Zwischenräume nach rechts verschieben
n LP	Den Cursor um n Zwischenräume nach links verschieben
L	den Rest der Zeile auflisten und zum Anfang der Zeile zurückkehren
X	den Rest der Zeile auflisten, den Cursor zum Ende der Zeile und den Insert(Einfüge-) Unterbefehl aufrufen
I	die folgenden Zeichen an der augenblicklichen Lage des Cursors einfügen; dieser Unterbefehl wird mit SHIFT HP verlassen
A	Änderungen rückgängig machen und den Cursor zum Anfang der Zeile zurücksetzen
E	Editieren beenden, alle Änderungen sichern und in den Befehlsmodus zurückkehren
Q	Editieren beenden, alle Änderungen rückgängig machen und in den Befehlsmodus zurückkehren
H	Rest der Zeile löschen und durch die folgenden Zeichen ersetzen; dieser Unterbefehl wird mit SHIFT HP verlassen
nD	von der augenblicklichen Lage des Cursors an die n folgenden Zeichen entfernen
nC	von der augenblicklichen Lage des Cursors an die n folgenden Zeichen durch die nächsten n Zeichen, die eingegeben werden, ersetzen
nSc	den Cursor von seiner augenblicklichen Lage aus an die Stelle fahren, wo das Zeichen c zum n-ten Male auftritt
nKc	alle Zeichen von der augenblicklichen Lage des Cursors an bis zu der Stelle entfernen, wo das Zeichen c zum n-ten Male auftritt

Sicherlich gibt es noch komfortablere bzw. bessere Editoren als diesen von Tandy (BASIC Level II). Aber immerhin ist es besser mit diesem zu arbeiten, als die Zeilen neu zu schreiben bzw. mit der Leertaste vorrücken zu müssen.

Klaus Hermann

Soll das heißen, daß Ihr Computer behauptet, wir seien vor der Schweizer Küste?. . .



Aber was werden wir tun, wenn Computer die ganze Arbeit machen?

WETTBEWERBSBEITRAG UM DAS 'PROGRAMM DES JAHRES' VON ARNULF SOPP

```
10 '     *****   BASIC-pur-Treiber für die HRG 1b von RB-Electronic   *****

          Die Graphik hat die Koordinaten X und Y (BASIC-Variable), die von
          0-383 (X) und von 0-191 (Y) gehen.
20 '     Dieses Programm stellt die Funktionen SET, RESET und POINT zur
         Verfügung, analog zur Klötzchengraphik.
30 '     Ein zu setzender, zu löschender oder zu prüfender HRG-Punkt wird
         mit den Koordinaten (Variablen) X und Y gekennzeichnet. Beispiel:
          X=50:Y=100. Dies ist der Punkt an der Stelle 50/100 auf dem Bildschirm.
40 '     SET wird mit GOSUB 10000, RESET mit GOSUB 20000, POINT mit GOSUB 30000
         realisiert. SET (10000) und RESET (20000) setzen bzw. löschen Punkte
          wie von Level-II-BASIC her gewohnt.
50 '     POINT (30000) lädt die Variable D mit -1 (Punkt gesetzt) oder 0
         (Punkt nicht gesetzt), entsprechend den wahr/unwahr-Usancen von
         Level II.

60 DEFINTA-Z'
         Dieser Befehl sollte allen SET/RESET/POINT-Operationen vorangehen,
         um die Ausführungsgeschwindigkeit zu erhöhen. Außerdem dient er als
          Lückenfüller, um dieses Programm wirklich zu einem Zehnzeiler zu
          machen. (Kicher!)

10000 GOSUB40000:OUT5%,AORC:RETURN'
      SET-Unterprogramm

20000 GOSUB40000:OUT5%,AAND(NOTC):RETURN'
      RESET-Unterprogramm

30000 GOSUB40000:D=((AANDC)<>0%):RETURN'
      POINT-Unterprogramm

40000 X1=INT(X/6%):B=X-X1*6%:Y1=INT(Y/12%).Y2=Y-Y1*12%:Z=Y2*1024%+Y1*64%+X1:
      ZM=Z/256%:ZL=ZAND255%:OUT2%,ZL:OUT3%.ZM:C=2%ÄB:A=INP(4%):RETURN'
      gemeinsames Unterprogramm, hauptsächlich Koordinaten bestimmen
```

* * * Mein erstes 'ASSEMBLER-Programm' * * *

====> Schau Dir mal den Bildschirm an! <====

```
8000          00010         ORG     8000H              ;EINSPRUNGADR.
8000 21003C   00020 START   LD      HL,3C00H           ;1.BILDSCHIRM-POS.
8003 11013C   00030         LD      DE,3C01H           ;NAECHSTE POS.
8006 01FF03   00040         LD      BC,03FFH           ;ZAEHLE 1023 VORWAERTS
8009 36BF     00050         LD      (HL),0BFH          ;MACH 1.POS. WEISS
800B EDB0     00060         LDIR                       ;UND ALLE UEBRIGEN
800D CD5880   00070         CALL    PSZYKL             ;ZUM PAUSE-ZYKLUS
-----------------------------------------------------------------------------
8010 20       00130 REKLAM  DEFM    '   REKLAME IST GUT   ' ;BEHAUPTUNG
0015          00140 LAENG1  EQU     $-REKLAM           ;LAENGE 1.TEXT
8025 00       00150         DEFB    00H                ;WEITER GEHT'S
8026 20       00160 WAHRH   DEFM    ' WAHRHEIT IST BESSER ' ;GEGENBEHAUPTG.
0015          00170 LAENG2  EQU     $-WAHRH            ;LAENGE 2.TEXT
803B 00       00180         DEFB    00H                ;TEXTENDE
-----------------------------------------------------------------------------
803C 011500   00190         LD      BC,LAENG1          ;CHR-ZAEHLER
803F 11133D   00200         LD      DE,3D13H           ;SCHAU NICHT HIN
8042 211080   00210         LD      HL,REKLAM          ;ANFANG DES UEBELS
8045 EDB0     00220         LDIR                       ;AUSGABE
8047 CD5880   00230         CALL    PSZYKL             ;PAUSE-ZYKLUS
-----------------------------------------------------------------------------
804A 011500   00250         LD      BC,LAENG2          ;CHR-ZAEHLER
804D 11D33D   00260         LD      DE,3DD3H           ;NOTABENE
8050 212680   00270         LD      HL,WAHRH           ;HEISST 'MERKE GUT'
8053 EDB0     00280         LDIR                       ;JETZT KOMMT'S
8055 C35580   00290 LOOP    JP      LOOP               ;SIEH DICH SATT
-----------------------------------------------------------------------------
              00305                                    ;SUBMARIN
8058 0603     00310 PSZYKL  LD      B,3H               ;PAUSE-WIEDERHOLFAKTOR
805A C5       00320 PAUSE   PUSH    BC                 ;BEWAHRE B (C = EGAL)
805B 01FFFF   00330         LD      BC,0FFFFH          ;PAUSENLAENGE
805E CD6000   00340         CALL    60H                ;PAUSE (ROM-CALL)
8061 C1       00350         POP     BC                 ;HOLE WIEDERHOLFAKTOR
8062 10F6     00360         DJNZ    PAUSE              ;WIEDERHOLE PAUSE
8064 C9       00370         RET                        ;WEITER
8000          00380         END     START              ;glückliches Ende!
00000 TOTAL ERRORS
33672  TEXT AREA BYTES LEFT
-----------------------------------------------------------------------------
```

SYMBOL TABLE

```
LAENG1 0015 00140   00190
LAENG2 0015 00170   00250
LOOP   8055 00290   00290
PAUSE  805A 00320   00360
PSZYKL 8058 00310   00070 00230
REKLAM 8010 00130   00140 00210
START  8000 00020   00380
WAHRH  8026 00160   00170 00270
```


Schreiben, um verständlich zu machen

HEJ, Leute – verzagt nicht !
Auch wenn ZEUS, der Große Assemblierer, SEINEN Blitz auf Euch herabschleudert! ER WIRD EUCH NICHT TREFFEN: Denn ich habe IHN zum Wanken gebracht!

Heute sage ich Euch (noch etwas hinter der hohlen Hand):
*** Macht IHN Euch untertan! ***
Von nun an wird "ER", der Göttervater, klein geschrieben!
Denn er läßt sich zu UNS herab...
– auf einer Leiter, mühsam gehalten von Zween von Euch, denen ich für ihre Hilfe beim mühevollen "Einstieg" (Aufstieg? Abstieg!) hier danken möchte, ohne ihre Namen zu nennen – denn sonst laßt Ihr Assembler-Willigen ihnen keine Ruhe mehr!
(Anm.: Bitte, schreibt doch dieses "ihre" und "ihnen" nicht immer groß, wenn, wie hier, in der 3.Person gemeint – tut uns den Gefallen, ja? Dankeschön! Dafür verrate ich Euch auch, wie man "ihn" – den Göttervater – in die Kniee zwingt...)

Hier nun das Ergebnis meines unermüdlichen Einsatzes im HEXen-Lotto (wie immer natürlich ohne Gewähr!)
Wer's besser weiß,
erhält 'nen Preis!

Das Thema des Programms ist völlig uninteressant (erst wird die Tafel gereinigt; dann schreibt ein Lümmel eine freche Behauptung drauf; schließlich wird er von weiser Hand eines Besseren belehrt – das ist alles). * Lediglich auf die Abfolge und Wirkung der Befehle ("Mnemonics" = Gedächtnisstützen) und ihre richtige Anwendung kam es mir an. Und Euch zunächst doch auch!? ("Euch" ist jetzt immer selektiv gemeint: mein "Seminar" ist Senf für die Experten; die sollen weiterblättern, es gibt Interessanteres in diesem Heft. Ich wende mich an die große Menge der Unschuldigen, der Unbedarften – als Einäugiger unter den Sehbehinderten...)

Seht Euch das Listing an: es ist fein säuberlich in sieben Spalten unterteilt. Die beiden ersten Spalten von links lassen wir da liegen, wo sie sind, nämlich links. Die hat der Assembler auf dem Gewissen. Die 3.Spalte, das sind die Zeilen-Nummern, die ich benutzt habe bei der Programm-Entwicklung.
Gehen wir nun nach diesen Zeilen-Nummern vor:
In der ersten Zeile habe ich "ihn" (den Assembler, künftig von mir nur noch "Ass" genannt) "angewiesen", daß er das von ihm Verdaute im Örtchen Nr. 8000H (Speicheradresse) abladen soll. Wohlgemerkt: ich habe ihn "angewiesen", ihm eine "Anweisung erteilt" – nicht einen Befehl, den er an die CPU (=Central Processing Unit = Leitzentrale unseres GENIE/TRS80) weitergeben solle. Dies ist ein feiner Schiedunter, wie Ihr noch merken werdet.
Zeilen 2 bis 4: Jetzt wird es ernst, sehr ernst sogar!
Ein ausnahmsweise erfolgreicher Sisyphos kreuzt auf: er schafft es in der Tat, ganze Blöcke zu verschieben, was dem König von Korinth sz. nie gelingen sollte...
Der "Block": das ist hier eine Folge von Bytes, z.B. eine Anzahl Zeichen gemäß ASCII-Code oder einfach eine Mitteilung (Deutsch: "message"), die man z.B. auf den Bildschirm bekommen will.
Wie geschieht das? Dies ist eins der wichtigsten Geheimnisse des Herrn Ass, das wir jetzt entschleiern wollen:
Wir alle wissen natürlich schon, daß die Seele unseres Spielzeugs im wesentlichen zweiteilig (nicht etwa zwiespältig) ist: sie hat ein Gedächtnis und ein Herz; auf "Deutsch": ein memory und eine CPU (s.o.). Letztere, die Leit- und Arbeitszentrale, besteht (unter anderem) aus 12 Registern – von den Ersatzregistern sehe ich einmal ab –, deren jedes 1Byte = 8 Bits faßt. Laßt Euch nicht durch deren Namen verwirren, bis auf vier sind es nichts anderes als

Haus-Nummern: B, C, D, E, H, L, IX und IY. Die vier übrigen sind die "Edleren", und zwar: der fleißige Akkumulator **A** (nomen est omen), die symbolisierende Fahne oder Flagge **F**, der Programm-Zähler oder Program Counter **PC** und der Stapel-Zeiger oder Stack Pointer **SP**. Die Funktionen all dieser Register lassen uns jetzt noch kalt.
Merken müssen wir uns schon, daß man einige davon zu Pärchen zusammenfügen kann. Das ist wichtig, wenn das Fassungsvermögen von einem einzigen Byte nicht ausreicht. Zum Beispiel bei Speicher-Adressen. Die gehen ja bis 65535 = FFFFH. Ein Byte kann jedoch höchstens FFH = 255 darstellen (die Kenntnis der HEXerei darf ich wohl voraussetzen).
Folgende Pärchen sind machbar: AF / BC / DE / HL.
AF - das fällt zunächst schwer; ist aber als halber Affe leicht zu merken. BC heißt zwar nicht Byte Counter, könnte aber. Dieses Pärchen countet häufig. DE = Du(mmer) Esel ist für fast alles da. Und HL = High/Low = HiLo merkt sich auch leicht, denn dessen Hauptfunktion ist in der Tat der "Hinweis" auf eine Speicheradresse: erst das "Hohe", spr. das vordere Byte (z.B. von "3FD7H" landet das Byte "3F" im Register H), dann folgt das "Lausige" - Verzeihung: das Lowere Byte (D7 kommt also ins Register L). Im Speicher steht's dann allerdings umgekehrt - aber diese Verwirrungs-Taktik zieht bei uns nicht mehr, wir Sehleute lassen uns nicht erschüttern (nach dem TSW-Motto "Trau, Schau, Wem").
So, nun kommt's:
Wir wollen zunächst einmal den Bildschirm ganz schnell weiß anstreichen. Hierfür bedienen wir uns des "White Byte". Mit dem "Weißen Byte" meine ich den Weißmacher, das Grafikzeichen mit dem ASCII-Code 191 = BFH. Wenn wir jede der 16 mal 64 = 1024 Positionen des Bildschirms mit dem Byte BFH besetzen, so wird er lückenlos aufgehellt.

ASS hat für diese und ähnliche Prozeduren zur Vereinfachung bereits bestimmte Register-Paare verpflichtet: das HL-Paar wird mit der ersten Adresse des Bildschirms (3C00H), das DE-Paar mit der nächsten (3C01H) "geladen" (LOAD, kurz LD):

```
LD HL, 3C00h
LD DE, 3C01h
```

D.h.: "LaDe HL mit 3C00h - LaDe DE mit 3C01h".
So einfach ist das. Leider wieder rückwärts. Also merkt Euch: HL ist immer der Geber (Hingebender Lieferant), DE = Der Empfänger.
Ass macht's stets so kurz wie möglich. Gerad so kurz, daß wir uns noch etwas dabei denken können. Und nichts anderes als solche Gedächtniskrücken ("MNEMONICS") versteht der (Götter-)Vater Ass; NUR DIE kann er der CPU schmackhaft machen, ja regelrecht "vorkauen". Denn dieses Baby - unser Spielzeug - will nicht einmal mit ALETE gefüttert werden, geschweige denn mit MNEMONICS; nur mit diesen langweiligen Flips und Flops: mit 1=höheres oder 0=niederes Potential! Ein elektronischer Homunculus...
Wer gibt hier Wem Was? Das Konzept lautet: Gib 3C00h (der ersten Bildschirm-Position) einen Weißmacher und veranlasse die CPU, daß dieser an die nächste Position weitergereicht wird, danach wieder an die nächste usf. Wie oft? Solange bis der Schirm voller BFh ist, also noch 1023 mal, nachdem die erste Position schon "eins drauf gekriegt hat".
Diese Anzahl kontrolliert der ByteCähler BC. Wir schreiben daher diesmal:

```
LD BC, 03FFh
```

weil 1023 (Dez) = 03FF (Hex).
Gleich kann's losgehen. Das einzige, was wir noch zu tun haben, ist, den ersten Weißmacher als Pionier in die Bildschirm-Speicherstelle 3C00H zu packen. Also:

LD 3C00H, BFh ? ? ? Mitnichten !!!!!

Nächster MERKSATZ:
Es können nie - niemals zwei Zahlenwerte im Befehl stehen !
Das wäre für Herrn ASS wie Höhere Mathematik: soviel verkraftet er nicht!
Wir müssen mit Zahlen stets den Umweg über ein Register einschlagen (meist A oder HL), wenn im OP(erations)-Code (=5.Spalte) oder im Operanden (=6.Spalte) schon eine Zahl steht! Und eine Speicher-Adresse ist ein reiner Zahlenwert
In diesem Fall wird das Register HL in Klammern gesetzt, dann weiß Ass bzw. die CPU: Nicht das Register HL, sondern die Speicherstelle, die in HL steht, soll mit dem Zahlenwert BFh gefüllt werden!
Steht nicht ein Register-Buchstabe, sondern eine Zahl in Klammern, so ist damit eine "Hausnummer", d.h. eine Speicher-Adresse gemeint.
Soll die Speicherstelle 3C00h mit BFh geladen werden (damit sie weiß wird), so müssen wir das also dem Register HL (in dem die Adresse 3C00h bereits steht) mitteilen, indem wir es einklammern.
Allgemein gilt immer, wenn eine Klammer auftritt:
"Lade die Speicherstelle, die du im Sinn hast, mit dem Wert sowieso!"
("Sinn" = Klammer.)

In unserem Falle ist der "Wert" ein Code, nämlich der ASCII-Code "BFh".
Und noch eine Falle hätte Herr Ass bei diesem Versuchs-Befehl vorgefunden: er hätte "BFh" für einen String, eben für die Buchstaben-Folge BFh gehalten. Wir meinten aber eine Hex-Zahl. Die akzeptiert er als solche nur, wenn sie mit einer Ziffer beginnt. Doch woher nehmen? Code 191 (Dez) fängt in HEX-Darstellung nunmal mit dem Buchstaben B an! "Was tun?" sprach ZEUS hier nicht; sondern: "Eine Null ist immer gut: sie schadet niemand - wenn was dahinter steckt!!" (Wie recht er hat...)
Jetzt ist also alles klar (??) :

```
LD (HL), 0BFh
```

ist richtig. Nicht HL wird mit BFh geladen; sondern (HL), das heißt:
die Speicherzelle, deren Adresse in HL steht,
und das ist ja (siehe 1.Befehl) die erste Bildschirm-Position!
Wir dürfen uns den Schweiß abwischen, den ZEUS vor unseren Erfolg gesetzt.
Denn nun arbeitet ASS für uns. Es bedarf nur noch eines Fußtrittes, unentschlossen wie ASS ist - aber dann rast er!
Der Tritt, der den Ball (Fußball 1986!) ins Sausen bringt:
LaDe den Inhalt der Quelle HL ins Ziel DE, kurz "LD";
Inkrementiere (erhöhe) beide um 1, kurz "I";
vermindere gleichzeitig den Zähler BC um 1;
Repeat (wiederhole) diesen Vorgang, kurz "ER"
und zwar solange, bis in BC eine Null steht.
Fassen wir diese Abkürzungen zusammen, so erhalten wir das schlaue Kürzel
LDIR
das dies alles allein zuwege bringt.
Jetzt ist der Bildschirm zwar nicht sauber - aber rein. So schnell schafft das kein Persil.
Was steht denn da in der 4.Spalte? Sieht aus wie Aushänge-Schilder. Sind es auch, wie sie vor wichtigen Häusern hängen (z.B. bei Friseuren, Bäckern und - Ähnlichem).
Gelegentlich muß man der CPU sagen, wo sie mit ihrer Arbeit anfangen und wo aufhören soll. Sie versteht "START" und "END". Nicht immer ist "START" identisch mit "ORG" (= origine = Ursprung, eigentlich also auch gleich Anfang; hier aber Anfang des Codes im Speicher, START hingegen ist das Startloch für die CPU, wie RUN für BASIC.
übrigens, der Ordnung halber sei's gesagt, obschon der Club80 wohl namensgerecht nur den Mikroprozessor Z80 oder dessen Abkömmlinge betreibt: Alles hier Gesagte betrifft eben diesen; ich weiß nicht, ob alles z.B. für den "6502" (hochwohllöblicher COMMODORE) oder andere auch zutrifft. Wie gesagt: "OHNE GEWÄHR!"

==== **P A U S E !** ====

Gut erholt??
Was, der Monitor will auch pausieren? Möchte eine Zeitlang weiß bleiben? (Welcher Bösewicht möchte das nicht?)
Kann er haben!
Meine Pause ist aber noch nicht zu Ende! Ich lasse jetzt mal andere für mich arbeiten! Wo sind doch die vielen AL (Arbeitslosen)? .... Aha - hier in Zauberkästchen vor mir mit dem Namen KB (Kleines Biest wie KeyBoard) schlummern sie vor sich hin. Weckt sie auf, die Bediensteten im ROM !! Was heißt überhaupt "ROM" ?
Ich glaube: Routines Organization for Management! (Wer's nicht glaubt, verzichtet auf vieles!)
Unter den vielen Routinen, die es uns zu unserer Bequemlichkeit beim (Maschinen-)Programmieren bietet, befindet sich eine, die heißt "DELAY" (= Zaudere! Zögere!) Sie beginnt in der Speicherstelle 60H. Bevor wir diese in Anspruch nehmen, müssen wir aber unserem "Zählwerk" BC verraten, wie lange das "Bild" still stehen soll. Ich füllte BC daher mit der größten Zahl, die in dieses Doppel-Register (= 2 Byte) hineingeht (s. Zeile 330. Seht Ihr dort die Null zur Einleitung der HEX-Zahl? Erinnert Euch!)
Doch als ich das Progrämmchen dann laufen ließ, war mir das doch ein bißchen zu kurz: eine knappe Sekunde. Die Taktzeit des Delay-Befehls beträgt nur 14,7µs, wie man an passender Stelle nachlesen kann (z.B. bei Röckrath, "TRS80 ROM Listing", Seite 6). Dies mit FFFFh = 65535d multipliziert, ergibt 0,96 Sekunden.
Deshalb baute ich diese ROM-Routine in eine Schleife ein, die sie dreimal durchlaufen muß, ehe sie befreit wird, so daß die Pause wenigstens knapp drei Sekunden (2,88s) dauert.
Dies machte ich so: Ich packte die ROM-Routine in ein UP (Unterprogramm, "Deutsch":SUBMARINE (d.i. eine Sub-Routine, die unter Wasser schießt, so daß man sie innerhalb des Programms erst lange suchen muß und (siehe besagtes ROM-Listing) oft nur daran erkennt, daß sie mit einem "RET"(urn) aufhört).
Statt BASIC's "GOSUB" sagt man "CALL" (Zeile 70) und gibt die Zeilen-Nummer an, wo sich das UP befindet; gewöhnlich am Schluß eines Programms. Ich wählte die Nummer 310 (s.dort).
Schlau wie ich manchmal bin, sagte ich mir aber, daß das wohl nicht optimal ist, weil die Zeilen-Nummern dieses "Source-Codes" für Herrn Ass ziemlich uninteressant sind; er mißachtet sie und stopft die Bytes einfach hintereinanderweg in den Speicher, beginnend mit der Adresse, die ich ihm als "ORG" vorgegeben. Diese Adressen sind dann die eigentlichen Zeilen-Nummern.
Um mich von den allzu vergänglichen Zeilen-Nummern meines Codes (oder Kotes?) unabhängig zu machen, hängte ich wieder so ein Aushängeschild, auch "LABEL" genannt ("Etikett" wär' auch nett) vor den Beginn des UP und nannte es sinnig "PSZYKL" (PAUSE-ZYKLUS). Dieses steht nun stellvertretend für die Adresse des UP - egal, an welcher Stelle auch immer es nachher im Speicher abgelegt wird (anfängt).
Darum sieht Zeile 70 so aus, wie sie aussieht: mit "PSZYKLUS" als Zielscheibe.
Wie wurde diese Schleife, dieser Zyklus, der ab 310 beginnt, nun programmiert?
Als Schleifenzähler dient immer Register B. Leider aber auch (zusammen mit C) als "Uhr" für die Dauer der Pause. Was nun?
Ich habe mir einen Trick ausgedacht. Den habe ich weder gelesen noch hat ihn mir jemand verraten, und deshalb ist er möglicherweise auch unbeholfen. Aber er läuft - Hauptsache! (Die Erfahrenen unter uns würden sicher einen besseren wissen; aber leider sind die ja längst zu interessanteren Themen übergegangen - nicht ohne vorher ein- bis dreimal über dieses "ihre Nase gerümpft" zu haben...)
Ich lade also zuerst B mit der Zyklenhäufigkeit 3 (Zeile 310)
Dann stoße (PUSHe) ich diesen Wert rücksichtslos ins Abseits, "STACK" genannt (Zeile 320). Aus diesem (un)heimlichen Schlupfloch kann ich ihn jederzeit wieder hervorholen: es ist wie ein Räubernest: immer der Letzte, der sich dorthin geflüchtet, muß als Erster wieder heraus (was, Ihr habt noch nicht von

"LIFO" gehört? "LAST IN, FIRST OUT"? Schämt Euch!)
Als Geisel nimmt B noch das völlig unschuldige Christkindlein C mit, denn Einzelpersonen öffnet der Zerberus vor dem Unterschlupf dessen Tore nicht; er will immer zwei Bytes auf einmal verschlingen. Macht nichts, denn C interessiert uns heute weder in Freiheit noch in Gefangenschaft. (Immer wird dies nicht so klappen, fürchte ich...)
Nun kann BC getrost mit der Schleifendauer geladen werden (Zeile 330); sein Inhalt ist sichergestellt.
Und jetzt: Nichts wie her mit der DELAY-Routine (Zeile 340), dieser Schlafmütze!
Es ist deren besondere Eigenart, weiter zu schlafen, wenn sie gerufen wird; aber genau das ist ihre Aufgabe: solange das Zählwerk BC noch nicht bei Null angelangt ist, geschieht - nichts, rein gar nichts, und wir können uns in aller Ruhe das Werk des Herrn Saubermann ansehen.
Hat sie ausgeschlafen, POP(elt) Zeile 350 das verstoßene Pärchen BC (d.h. seine alten Werte) aus dem Stack wieder in sein Heimat-Registerpaar BC zurück, so daß wir in B wieder den letzten Schleifenstand stehen haben.

Nun kommt schon wieder was Neues! (Lieber Leser: Hast Du noch ein paar Speicherplätze frei im Hirn? Sonst geh hier schlafen! ROMCALL 60 weckt Dich!!)
"Denk Jaa Nich, Z80,
der Anfänger lacht sich!"
Der lacht sich gar nicht. Bis er merkt, daß er die erste Zeile (nicht des CODE, sondern dieses "Gedichtes") leicht zusammenfassen kann:
DJNZ = Dekrementiere (and) Jump (if flag bit is) Not Zero!
Diesmal auf gut Deutsch: "Vermindere B um 1 und springe (zurück zum Label PAUSE), sofern das Null-Bit des FLAG-Registers nicht gesetzt ist, d.h. nicht Null anzeigt!" Alles restlos klar?? KLar für den, der Deutsch versteht. (Oder hättet Ihr es lieber in Spanisch? Dann fragt Arnulf. Er antwortet Euch Spanisch; teils wirklich, teils wird's Euch so vorkommen...)
Wie auch immer: die Pausenschleife wird solange durchlaufen, bis B=0. In unserem Fall also dreimal.
Puh - das wär's. Und das bei 34 Grad im Schatten des Odenwaldes...
Nachdem sich der Beschauer genügend bei der Betrachtung des gereinigten Bildschirms gelangweilt hat, soll endlich was passieren. Laßt uns auf die Tafel eine Botschaft schreiben.
"Botschaft" heißt bei ASS: message (nicht zu verwechseln mit Messe oder Kirmes - aber ich wittere eine gewisse Verwandtschaft!)
Was ist mit dem Zauberwort "LDIR"? Könnten wir das hier nicht auch gebrauchen? Statt jeden Buchstaben unserer Message einzeln zu transportieren (das ginge auch), fassen wir sie wieder in einem "Block" zusammen und definieren einfach:
"Define Message" ===> DEFM '.................' (z.B. Zeile 130)
Das, was die Botschaft besagen soll, setzen wir dorthin, wo jetzt noch die Punkte stehen (Apostrophe nicht vergessen!) Ich glaube, ihre Länge ist beliebig (?)
Damit die CPU auch weiß, wann und wo die Message zu Ende ist, setzen wir noch ein Null-Byte hinterher, vermögens der Definition DEFB 00H (s.Zeile 150)
Trotzdem - die CPU ist ja sauduma - müssen wir ihr noch explizit sagen, wie lang das Ding ist. Das teilen wir wieder unserem "ByteCählwerk" BC mit:
LD BC, Länge der Botschaft
Zum Beispiel behauptet da einer "   REKLAME IST GUT   ", eine Sentenz mit je drei Blanks Anstands-Abstand vorn und hinten - weil Reklame sowas Vornehmes ist...
Also 21 Bytes (auch Blanks sind Bytes! Verachtet mir die Unscheinbaren nicht, sie leisten die wichtigsten Dienste!
Oderkannstedaslesen:Himmikruzifixibuasmirnicheeiheiligsblechle?
Somit: BC,15H (15H = 21dez)
Aber wir brauchen die Bytes auch nicht "von Hand" abzuzählen, wenn uns das schwerfällt. Wir können einen Trick anwenden.
Zunächst hängen wir wieder so ein "Schild" (Label) vor die Message-Definition.

Sinnvollerweise nannte ich es "REKLAM". Damit repräsentiert das Label "REKLAM" die uns verborgene Anfangs-Adresse der Message. Es ist dasselbe wie eine symbolische Konstante in der Mathematik, hinter der sich ein Zahlenwert versteckt, den wir noch gar nicht kennen.

ACHTUNG:
Länger als 6 alphanumerische Zeichen darf ein Label nicht sein und es muß mit einem Buchstaben beginnen!

Wenn die Message X Bytes lang ist, so belegt sie X Speicher-Adressen.
Wenn sie bei der Adresse REKLAM beginnt, befindet sich ihr letztes Byte demnach in der Adresse REKLAM+X-1 (rechnet nach!). Die nächste Zeile hat also die Adresse REKLAM+X. * Andersherum:
<u>Die Länge X ist gleich der Adresse der nächsten Zeile minus "REKLAM"</u>
Da die Adresse einer Zeile, in der man sich gerade befindet, sich durch das Symbol $ kennzeichnen läßt, ohne daß man ihren zahlenmäßigen Wert zu kennen braucht, lösen wir unser kleines Problem, indem wir
<u>sofort in der nächsten Zeile folgende *Gleichung* (EQUATION) hinschreiben:</u>

```
LAENG1 EQU $-REKLAM     (Zeile 140)
```

Somit füllen wir den Zähler BC mit LAENG1:

```
LD BC,LAENG1   (Zeile 190)
```

Nun brauchen wir nur noch HL und DE, wie gehabt, zu beschicken.
Die Quelle, der Haus- & Hof-Lieferant HL, peilt - das haben wir mithilfe des Labels so geregelt - die Adresse "REKLAM" an. So ergibt sich Zeile 210.
Wohin mit der "Botschaft"? Auf den Bildschirm natürlich, damit er nicht ewig weiß bleibt. Wir wählen als Anfang den Punkt mit der Adresse 15632 = 3D10H (wie schön: fängt mit einer Ziffer an!) Somit erhält "Der Empfänger" DE die Zieladresse 3D10, s.Zeile 200.
(Die Reihenfolge, in der wir die Register-Pärchen beschicken, spielt offenbar keine Rolle. Deswegen ist dieses kleine Zeilen-Durcheinander hier nichts weiter als eine freche Abwechslung. Punkt.)

So, das wär's schon. Jetzt nur noch schnell **LDIR** befohlen (Zeile 220) - - - und schon prangt da diese Behauptung der Sprüche-Macher auf dem schönen Laken...

Nur rein zur Übung wiederholen wir die Prozedur, diesmal mit einem VETO (Einspruch) - na, schaut es Euch selbst an.
Das Looping - der unbedingte Sprung (JUMP) "zurück" zum Label "LOOP" - kann unterbleiben, falls Ihr Euch das Wunderwerk nicht bis zu St.Nimmerlein's Tag ansehen wollt. Dann allerdings hilft nur noch das Fast-Allheilmittel "RESET" - so streng sind Asses Bräuche!! (Stimmt nicht: man kann auch den Dreitasten-Befehl 123 benutzen und dann im DEBUG "Q" sagen!).
Vergeßt nicht, am Schluß zu sagen "Nun Schluß!" oder vielmehr: "END(e)"
Wenn Ihr hinter "END" noch "START" setzt, tut Ihr, wie ich gehört habe, dem ASS oder auch der CPU einen Gefallen; dann weiß sie nämlich, wo sie starten soll, nachdem sie den ganzen Plan, nach dem sie marschieren soll, erst einmal durchgesehen und gründlich studiert hat. Ich glaube nämlich, genau das tut sie, bevor sie loslegt; wie könnte sie sonst wissen, was da alles definiert und equalisiert wurde, bevor sie die betreffenden Zeilen mit den Definitionen und Gleichungen überhaupt erreicht hat?? Das habe ich zwar nirgends gelesen und das hat mir auch niemand gesagt
<u>*(eben das ist ja der ganze Jammer bei diesem Lernvorhaben!)*</u>
aber mir sagt es schlichte Logik (die allerdings nicht immer ins Schwarze trifft...)

* * * * * *

So - und Ihr lieben geduldigen Leser tätet mir jetzt auch einen Gefallen, wenn Ihr mir bald mitteilen tuen tätet, ob Ihr daran interessiert seid, daß ich Euch künftig

weiterhin von meinem ständigen (?) zähen Kampf mit ASS und Genossen laufend berichte - vielleicht hat doch der eine oder andere etwas davon, der sich bisher nicht "rantraute".
Die Experten unter uns machen sicher manche Wendung besser. Vielleicht steuern sie jetzt auch mal etwas <u>unter</u> ihrem Niveau hierzu bei...
(und weisen mich, bitte, auf Fehler in meiner "Anleitung" hin - das Prograämachen läuft jedenfalls!)

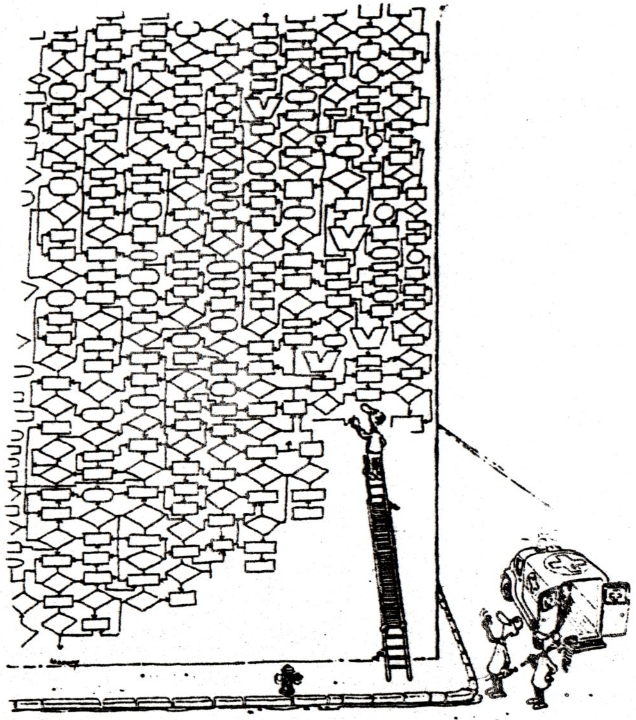
Glaubt nur nicht, ich wäre ein Menschenfreund.
Alles ist purer Egoismus!

Denn:

ERST WENN MAN ETWAS ZU BESCHREIBEN VERSUCHT,
MERKT MAN, OB MAN ES BEGRIFFEN HAT...

SO LONG ! (BUT NOT "PING-PONG"!) #KA-JOT

KaJot



Einige Assembler-Routinen

Wenn man als Assembler-Laie einige professionelle Programme betrachtet, fallen kurze Unterprogramme ins Auge, die Funktionen ausführen, die man selber vergeblich zu programmieren versucht hat. Jedenfalls ging es mir so mit Scripsit. Ganz am Anfang meiner Bekanntschaft mit Assembler wollte ich einige kleine Spielprogramme schreiben, die aber schon im Ansatz scheiterten: Zufallszahlen und die Zahlenausgaben auf dem Bildschirm waren damals ein riesiges Problem. Da ich vermute, daß auch andere klein anfangen und ähnliche Schwierigkeiten haben, möchte ich einige kurze Routinen vorstellen, die aus Scripsit stammen, aber wohl ähnlich in jedem größeren Programm auftauchen. Beim Ansehen und Ausprobieren kann der Anfänger schon viel über Assembler lernen, aber auch für geübtere Anwender dürften diese Routinen zum Einbau in ihre Programme interessant sein.

Gerald Schröder

Zu AUSWERT:
Dieses Unterprogramm erwartet, daß HL auf einen Buffer zeigt, in dem eine Dezimalzahl steht (0-65535). Außer Ziffern sind nur Leerzeichen in dem Buffer zugelassen. Das erste Zeichen, das weder Space noch Ziffer ist, führt zum Rücksprung aus dem Unterprogramm. Dabei steht die umgesetzte Zahl in HL und kann somit weiterverarbeitet werden.

Zu UMWAND:
Hier zeigt HL auf das Low-Byte einer Zahl, die in der Reihenfolge Low-Byte / High-Byte im Speicher steht. Diese Zahl wird in eine Dezimalzahl (0-65535) umgewandelt, welche rechtsbündig in einen Buffer geschrieben wird. Von dort kann sie zum Beispiel auf den Bildschirm gebracht werden.

Zu RND:
Dieses Programm erzeugt eine Zufallszahl in A. Das Programm ist zwar sehr kurz, aber eine Folge von Zufallszahlen scheint sich nicht zu wiederholen. Scripsit benötigt die Zufallszahlen, um die zusätzlichen Spaces beim Justify (Randausgleich) über die Zeile zu verteilen. Da die RND-Tabelle vor jedem Ausdruck in den Urzustand zurückversetzt wird, sehen alle Ausdrucke desselben Textes absolut identisch aus.

Zu TEILEN:
Hier wird L durch B geteilt, was eigentlich nicht aufregend ist, doch das Ergebnis ist interessant: In BC steht sozusagen eine Kommazahl im Hexadezimalsystem. Das heißt nichts weiter, als daß B die Zahl vor dem Komma und C die nach dem Komma beinhalten. Einziger Unterschied zum Dezimalsystem: C muß durch 256 geteilt werden, um zum korrekten Ergebnis zu kommen.
Beispiele:

```
Eingabe  L=2   B=1      Ausgabe  BC=02 00  (=2.00 dezimal)
         L=3   B=2               BC=01 80  (=1.50 dez.)
         L=1   B=3               BC=00 55  (=0.33 dez.)
```

Somit kann man jetzt auf- oder abrunden oder auch zwei Kommazahlen addieren. Scripsit benutzt auch diese Routine im Justify. Es berechnet damit, wieviele Spaces in der Zeile an einem Wortzwischenraum noch eingefügt werden müssen (L=Anzahl der einzufügenden Spaces, B=Anzahl der noch verfügbaren Wortzwischenräume). Zu dem Ergebnis BC wird eine Zufallszahl aus der Routine RND addiert, so daß eventuell aufgerundet wird, aber nicht unbedingt. Es werden dann B Spaces an dem aktuellen Wortzwischenraum eingefügt.


```
;     CALL AUSWERT                    geschützte Register:
;                                     alle außer HL
;     Zahl aus dem Buffer (HL)... auswerten.
;     Ausgabe: Zahl in HL
;
AUSWERT PUSH    AF
        PUSH    BC
        PUSH    DE
        EX      DE,HL                 ;Bufferzeiger NACH DE
        LD      HL,0000               ;HL LÖSCHEN
AUSW1   LD      A,(DE)                ;ZEICHEN LADEN
        INC     DE                    ;POINTER +1
        CP      20H                   ;SPACE? => NÄCHSTES ZEICHEN
        JR      Z,AUSW1
        CP      30H                   ;Ziffer?
        JR      C,AUSW2               ;nein, Absprung
        CP      3AH
        JR      NC,AUSW2              ;ebenso
;
;       ZIFFER GEFUNDEN
;
        ADD     HL,HL                 ;ZAHL VERGRÖßERN (ZIFFERN LINKS)
        LD      B,H
        LD      C,L
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,BC
        SUB     '0'                   ;AUS ZIFFER ZAHL ERZEUGEN
        LD      C,A                   ;NACH BC UND AUF HL ADDIEREN
        LD      B,00H
        ADD     HL,BC
        JR      AUSW1                 ;NÄCHSTE ZIFFER
;
AUSW2   POP     DE                    ;ABGANG
        POP     BC
        POP     AF
        RET

;     CALL UMWAND                     geschützte Register:
;                                     alle
;     Wandelt Zahl in (HL)/(HL+1) in Ziffern (dezimal) um.
;     Die Zahl wird rechtsbündig in den Buffer geschrieben.
;
UMWAND  PUSH    AF
        PUSH    DE
        PUSH    BC
        PUSH    HL
        PUSH    IY
        LD      IY,ZIFTAB
        LD      C,00H                 ;WIEVIELE ZIFFERN
        LD      E,(HL)                ;DIVIDENDEN LADEN (LOW-BYTE)
        INC     HL
        LD      D,(HL)                ;(HIGH-BYTE)
        LD      HL,BUFFER             ;AUSGABE-BUFFER
        EX      DE,HL                 ;NACH DE, ZAHL IN HL
UMW1    PUSH    DE                    ;MOM. BUFFERADRESSE RETTEN
        LD      A,'0'                 ;ZIFFER "0"
        LD      D,(IY+01H)            ;DIVISOR LADEN (10.000, 1000, 100, 10, 1)
        LD      E,(IY+00H)
UMW2    OR      A
        SBC     HL,DE                 ;PAßT WIE OFT IN IN DEN DIVIDENDEN?
```

```
        JR      C,UMW3          ;FERTIG MIT DIESEM DIVISOR
        INC     A               ;ZIFFER+1
        JR      UMW2            ;UND WEITER
;
UMW3    ADD     HL,DE           ;DIVIDENDEN KORRIGIEREN
        INC     IY              ;NÄCHSTER DIVISOR
        INC     IY
        CP      '0'             ;DIVISOR>DIVIDEND?
        JR      NZ,UMW4         ;NEIN, DIVIDEND/DIVISOR>0
        INC     C               ;SCHON VORHER ZIFFER?
        DEC     C
        JR      NZ,UMW5         ;JA, ZIFFER "0" AUSGEBEN
        LD      A,' '           ;NEIN, FÜHRENDE LEERZEICHEN AUSGEBEN
        JR      UMW5            ;UND IN DEN BUFFER
;
UMW4    LD      C,A             ;"SCHON VORHER ZIFFER"-FLAG SETZEN
UMW5    DEC     E               ;DIVISOR=1? JA, FERTIG, AB
        POP     DE              ;BUFFERZEIGER ZURÜCK
        LD      (DE),A          ;ZIFFER ABSPEICHERN
        INC     DE              ;ZEIGER+1
        JR      NZ,UMW1         ;NOCH NICHT FERTIG? => NÄCHSTER DIVISOR
;
        INC     C               ;ZAHL=0? (KEINE ZIFFER?)
        DEC     C
        JR      NZ,UMW6         ;NEIN, AB
        DEC     DE              ;JA, "0" IN DEN BUFFER
        LD      A,'0'
        LD      (DE),A
UMW6    POP     IY              ;AB
        POP     HL
        POP     BC
        POP     DE
        POP     AF
        RET
;
;       DIVISOREN FÜR VORSTEHENDES UNTERPROGRAMM
;
ZIFTAB  DW      10000
        DW      1000
        DW      100
        DW      10
        DW      1
;
;       AUSGABE-BUFFER FÜR VORSTEHENDES UNTERPROGRAMM
;
BUFFER  DS      10

;    CALL RND                   geschützte Register:
;                               alle außer AF
;    Erzeugt eine Zufallszahl in A
;
RND     PUSH    BC
        PUSH    HL
        LD      HL,RNDTAB       ;RANDOM-TABELLE
        LD      B,08H
        LD      A,(HL)
RND1    RLCA
        RLCA
        RLCA
        XOR     (HL)
        RLA
        RLA
```



```
        DEC     HL
        DEC     HL
        DEC     HL
        LD      A,(HL)
        RLA
        LD      (HL),A
        INC     L
        LD      A,(HL)
        RLA
        LD      (HL),A
        INC     L
        LD      A,(HL)
        RLA
        LD      (HL),A
        INC     L
        LD      A,(HL)
        RLA
        LD      (HL),A
        DJNZ    RND1
        POP     HL
        POP     BC
        RET
;
        DB      34H,12H,78H
RNDTAB  DB      56H

;    CALL TEILER                geschützte Register:
;                               alle außer BC
;    Teilt L durch B. Ausgabe: B=INT(L/B)
;                              C=L/B - INT(L/B) in 256stel
;
TEILER  PUSH    AF              ;RETTEN
        PUSH    HL
        PUSH    DE
        LD      H,00H           ;H UND C LÖSCHEN, D=2
        LD      D,02H
        LD      C,H
TEIL1   LD      A,08H
TEIL2   ADD     HL,HL
        OR      A
        SBC     HL,BC
        INC     HL
        JR      NC,TEIL3
        ADD     HL,BC
        RES     0,L
TEIL3   DEC     A
        JR      NZ,TEIL2        ;8 MAL
;
        DEC     D
        JR      Z,TEIL4         ;2 MAL
        LD      E,L             ;L RETTEN
        LD      L,C             ;L = 0
        JR      TEIL1           ;ZUM 2. MAL
;
TEIL4   LD      B,E             ;1. L NACH E
        LD      C,L             ;2. L NACH C
        POP     DE
        POP     HL
        POP     AF
        RET
```

Mehr Fernsehen für's Geld

Das Genie III s (und vermutlich einige andere Genies "darunter") arbeitet im Gegensatz zum Genie I/II mit einem Video-Controller (CRTC, cathode ray tube controller). Der hat eine Reihe von Registern, mit denen gewisse Bildschirmzustände programmiert werden. G-DOS 2.4 unterstützt lt. Handbuch diese Möglichkeiten. In einem Falle ist das aber pure Hochstapelei: Der Befehl ## soll angeblich in der Lage sein, u. a. die beiden Bildschirmseiten des Formats 16*64 Zeichen untereinander auszutauschen. Er hält sich aber vornehm zurück. Jedenfalls ist mir dergleichen noch nie gelungen. Bis heute.

Dazu ein wenig Theorie: In den Registern 12 und 13 des CRTC ist die relative Startadresse des angezeigten Bildschirms im Verhältnis zur Startadresse des Video-RAMs abgelegt. Dieses beginnt bei 3800h. Beim Kleinbildschirm 16*64 liegt die erste angezeigte Adresse bei 3C00h, also 0400h Bytes hoch im Video-RAM. Daher wird beim Umschalten auf diesen "Kleinbildschirm" 0400h in die Register 12 und 13 des Controllers eingeschrieben. Dazu wird zunächst 0Ch (12d) auf den Port F6h ausgegeben, um das Register Nr. 12 zu adressieren. Der Port F7h kriegt dann das MSB dieses Anzeige-Offsets mitgeteilt. Anschließend wird über F6h das Register 13 aufgeweckt, um über F7h das LSB ausgeben zu können.

Um nun eine Bildschirmseite in die andere zu kopieren, beide auszutauschen, die eine oder andere vollzumalen usw., muß demnach das MSB des Anzeige-Offsets entweder 00h oder 04h lauten. Auf diese Weise wird die obere oder die untere Bildschirmseite adressiert. Den CRTC interessiert nämlich nicht die Bohne, wo die Designer des G-DOS oder des Genie den Bildschirm gerne memory mapped haben möchten. Er schreibt stur dorthin, wo es ihm seine Register 12 und 13 vorschlagen. Schreibt man nun nach Ausgabe des Offsets 0000h einen Text ab 3800h (Großbildschirm) und gibt dann den Offset 0400h aus (Kleinbildschirm), dann ist der Text verschwunden, weil nicht dort gelesen wird, wo er steht.

Das ist das Funktionsprinzip der beiden Bildschirmseiten, nach dem der ##-Befehl - angeblich, leider nicht wirklich - arbeitet. Um in einem Assembler- oder BASIC-Programm dennoch dieses interessante Feature des CRTC nutzen zu können, möchte ich hier ein Strickmuster vorstellen, nach dem der Bildschirm dem User aus der Hand frißt. Weil sie einfacher zu verstehen ist, zunächst die BASIC-Variante:

In der Zeile 10 wird mit dem Library-Befehl ##,V der Bildschirm auf 32*64 Zeichen eingestellt. Er liegt nun im Bereich 3800-3FFFh. CLS löscht ihn. Links oben wird der Text mit dem Hinweis auf die Seite 0 angezeigt. Mit dem Library-Befehl 64 wird auf den Bildschirm mit 16*64 Zeichen umgeschaltet (3C00-3FFFh). Wieder links oben, aber ein ganzes Kilobyte höher wird der Hinweis auf die Seite 1 geprintet. Danach wird das Offset-Register 12 adressiert. Es genügt nämlich, das MSB des Offsets zu verändern. Das LSB ist immer 00h. In der Zeile 20 wird eben dieser Offset abwechselnd zu 0000h oder zu 0400h definiert. Dazwischen liegt jeweils eine Verzögerungsschleife, um den Effekt beobachten zu können: Beide zuvor in den Bildschirm geladenen Texte kommen abwechselnd zur Anzeige.

Das darüber gelistete Maschinenprogramm tut genau dasselbe. Die ausführlichen Kommentare erklären gewiß, was sich da tut. Ich möchte darüber hinaus lediglich noch auf die Unterprogramme eingehen, die gecallt werden:

Das UP an 3497h schaltet auf die maximale Auflösung mit 32*64 Zeichen. Dazu holt es die entsprechenden CRTC-Parameter aus 37F0h ff. und gibt sie an den Controller aus. 34A1h schaltet auf die gleiche Weise auf 16*64 Zeichen, die kleinste Auflösung. Der Vollständigkeit halber seien noch die beiden Unterprogramme für die zwei verbleibenden Bildschirmformate genannt: 34A6h schaltet auf 24*64, 349Ch auf 25*80 Zeichen. Das UP an 01C9h löscht den Bildschirm bei jedem Format. Mit der Routine an 4467h wird ein Text, auf den HL zeigt, auf den Bildschirm übertragen. Er muß mit 03h oder 0Dh abgeschlossen sein.

Beide Programmvarianten lassen sich mit den von Fall zu Fall erforderlichen Modifikationen in eigene Programme einbauen. Eine denkbare Anwendung: Auf Knopfdruck soll der User ein Menü angezeigt bekommen. Dazu wird nach Umschalten auf den Großbildschirm das Menü einfach mit PRINT in den Bildschirm geladen. Nach Rückkehr zum Kleinbildschirm bleibt dieser Text erhalten. Er wartet sozusagen im Keller. Bei Bedarf braucht nur das Register 12 des CRTC über Port F6h adressiert zu werden, damit man sodann 00 auf Port F7h ausgeben kann. Damit ist der Bildschirm ab 3800h definiert. Der versteckte Text erscheint, der zuletzt angezeigte verschwindet. Nach Ausgabe von 04h auf Port F7h werden beide Bildschirmseiten wieder zurückgetauscht. Es ist, als wäre nichts geschehen.

Arnulf Sopp

```
5200           00001         ORG    5200h
5200  CD9734   00002 start   CALL   3497h        ;auf 32*64 Z. schalten
5203  CDC901   00003         CALL   01c9h        ;Bildsch. ab 3800 löschen
5206  212852   00004         LD     HL,text0     ;Text "Seite 0"
5209  CD6744   00005         CALL   4467h        ;anzeigen
520C  CDA134   00006         CALL   34a1h        ;auf 16*64 Z. schalten
520F  214A52   00007         LD     HL,text1     ;Text "Seite 1"
5212  CD6744   00008         CALL   4467h        ;anzeigen
5215  3E0C     00009         LD     A,0ch        ;Anz.-Offset-Reg. d. CRTC
5217  D3F6     00010         OUT    (0f6h),A     ;adressieren
5219  AF       00011         XOR    A            ;A <- 00 (Offset-Byte)
521A  F5       00012 loop    PUSH   AF           ;Akku retten
521B  01FFFF   00013         LD     BC,0ffffh    ;Zähler für Warteschleife
521E  CD6000   00014         CALL   0060h        ;etwas trödeln
5221  F1       00015         POP    AF           ;Akku zurück
5222  EE04     00016         XOR    04h          ;Offset-Bit umschalten
5224  D3F7     00017         OUT    (0f7h),A     ;neu ausgeben
5226  18F2     00018         JR     loop         ;und so weiter
5228  44       00019 text0   DM     'Dies ist Seite 0 des Bildschirms.',03h
524A  55       00020 text1   DM     'Und dieses hier ist Seite 1.',03h
               00021 endprg
5200           00022         END    start

00000 Fehler
```

Und hier dasselbe in BASIC:

```
10 CMD"##,V":CLS:PRINT"Dies ist Seite 0 des Bildschirms.":CMD"64":
PRINT"Und dieses hier ist Seite 1.":OUT&HF6,12
'Texte schreiben, Anzeige-Offset-Register adressieren

20 FORI=0TO500:NEXT:OUT&HF7,4:FORI=0TO500:NEXT:OUT&HF7,0:GOTO20
'Zwischen den Offsets 0000 und 0400 hin- und herschalten
```


HEFT 14
Juli
1986

```
1 'ASCIITAB VON GERALD SCHRÖDER
2 'DIESES PROGRAMM GIBT EINE VIERSPALTIGE TABELLE (64 ZEILEN) AUF DEM
3 'DRUCKER AUS. JEDE SPALTE HAT DAS FORMAT:
4 ' DEZIMALZAHL    HEXADEZIMALZAHL    BINÄRZAHL    ASCII-CODE
5 'DIE TABELLE      KANN VIELSEITIG BEIM PROGRAMMIEREN GENUTZT WERDEN.
6 'VIELEN DANK AN KAJOT FÜR DIE DEZ/HEX-ROUTINE
7 'ACHTUNG: ZEILE 20 MUß JE NACH DRUCKER GEÄNDERT WERDEN!
8 '
10 CLS:CLEAR70:DEFINTA,I-N:DEFSTRH,X:RESTORE
20 LPRINTCHR$(15) '132 ZEICHEN/ZEILE BEI GEMINI 10X
30 FORDA=0TO63:FORDB=0TO3:DE=DA+DB*64:LPRINTUSING"###";DE;:LPRINT" ";
40 ' *** HEXADEZIMALWERT ***
50 I=0:D1=DE
60 D2=INT(D1/16):B(I)=D1-D2*16:IFD2>0THENI=I+1:D1=D2:GOTO60
70 FORJ=0TOI:IFB(J)<10THENX(J)=RIGHT$(STR$(B(J)),LEN(STR$(B(J)))-1):ELSEX
(J)=CHR$(B(J)+55)
80 NEXTJ:HE="":FORJ=ITO0STEP-1:HE=HE+X(J):NEXTJ:N=2-LEN(HE):NU$=STRING$(N
,"0")
90 LPRINTNU$+HE;" ";
100 ' *** BINÄRWERT ***
110 I=0:D1=DE
120 D2=INT(D1/2):B(I)=D1-D2*2:IFD2>0THENI=I+1:D1=D2:GOTO120
130 FORJ=0TOI:X(J)=RIGHT$(STR$(B(J)),1)
140 NEXTJ:HE="":FORJ=ITO0STEP-1:HE=HE+X(J):NEXTJ:N=8-LEN(HE):NU$=STRING$(
N,"0")
150 LPRINTNU$+HE;" ";
160 ' *** ASCII-CODE ***
170 IFDE>32ANDDE<127THENX=CHR$(DE)ELSEIFDE<128THENREADXELSEX=" "
180 LPRINTUSING"%   %";X;:IFDB<3THENLPRINT"      ";
190 NEXTDB:LPRINT:NEXTDA:END
200 ' *** DATA'S FÜR ASCII-CODE ***
210 DATANUL,SOH,STX,ETX,EOT,ENQ,ACK,BEL,BS,HT,LF,VT,FF,CR,SO,SI,DLE,DC1,D
C2,DC3,DC4,NAK,SYN,ETB,CAN,EM,SUB,ESC,FS,GS,RS,US,SPACE,DEL
```

```
87 57 01010111 W      151 97 10010111      215 D7 11010111
88 58 01011000 X      152 98 10011000      216 D8 11011000
89 59 01011001 Y      153 99 10011001      217 D9 11011001
90 5A 01011010 Z      154 9A 10011010      218 DA 11011010
91 5B 01011011 Ä      155 9B 10011011      219 DB 11011011
92 5C 01011100 Ö      156 9C 10011100      220 DC 11011100
93 5D 01011101 Ü      157 9D 10011101      221 DD 11011101
94 5E 01011110 ^      158 9E 10011110      222 DE 11011110
95 5F 01011111 _      159 9F 10011111      223 DF 11011111
96 60 01100000 `      160 A0 10100000      224 E0 11100000
97 61 01100001 a      161 A1 10100001      225 E1 11100001
98 62 01100010 b      162 A2 10100010      226 E2 11100010
```

HRG - Bildübertragung

Schon vor einiger Zeit hat mich Ulrich Böckling auf einen Service des WDR-Computerclubs aufmerksam gemacht. In der Mailbox dieses Clubs kann man neben sehr vielen anderen Informationen und Programmen seit einiger Zeit hochauflösende Bilder abrufen. Dies geschieht nach einem weiter unten genau beschriebenen Verfahren, auf das ich hier nicht weiter eingehen möchte.
Interessant finde ich die Möglichkeit, nun endlich Bilder vom Model 1 auf's Model 3/4 übertragen zu können und natürlich sind auch Genie I/II/III und IIs/IIIs -Bilder übertragbar. Vielleicht ein Grund mehr, sich eine HRG anzuschaffen!
Die für TRS 80 M.1 und M.3/4 sind übrigens ab sofort beim Kajott in der Programmbibliothek abrufbar. Wer HRG-Bilder mit mir tauschen möchte, kann sich direkt an mich wenden. Vielleicht entsteht ja mal eine ganze Sammlung von für alle Tandygeräte (und viele andere Computermodelle) lesbaren HRG-Bildern wie z.B. Schaltplänen, Funktionsgrafiken usw.. Ich würde mich sehr darüber freuen! Euer

Hartmut Obermann

Bildübertragung per DFÜ

Als wir im Computerclub vor etwa einem Jahr mit der Installation unseres KOMCOM begannen, sah die Welt der Datenfernübertragung noch anders aus als heute. Damals war die Verbreitung von Akustikkopplern als recht dürftig zu bezeichnen. Datenfernübertragung war einigen wenigen Spezialisten vorbehalten. Die Hardware, wie Akustikkoppler, war noch immens teuer und die Software war auf dem Hobbymarkt kaum zu bekommen.
Dieses hat sich geändert - und zwar mit großer Geschwindigkeit. Heute sind viele unserer Zuschauer in der Lage, mit ihrem Heimcomputer DFÜ zu machen. Wir zählen in unserem KOMCOM nach etwa einem Jahr immerhin 65000 Anrufe. Wenn wir davon ausgehen, daß die Technik mithin bekannt ist, so können wir nun in der zweiten Phase damit beginnen, die Angebote in INFO-BOXEN etwas bunter zu gestalten. Es soll ein zusätzlicher Service sein, der dem einen Nutzer gefällt, dem zweiten Nutzer vielleicht nicht behagt.
In der Computerclubsendung vom 2.6. propagierten wir eine neue Möglichkeit, die Datenfernübertragung (auch im 300-Baud Modus) zur Übermittlung von Bildern zu nutzen. Vorweg gesagt: Zum Rennwagen können wir die Box nicht umfunktionieren. Die Übertragung eines hochaufgelösten Bildes vom Apple oder vom C64 dauert 7 Minuten, das Bild vom Viktor oder vom QX10 mit mehr Bildpunkten entsprechend länger.
Doch wir sehen schon einen Sinn in der Möglichkeit der Übertragung solcher Bilder. Zum Beispiel kann man sich Schaltbilder direkt nach Hause holen und sie dann ausdrucken. Die Realisierung kann dann sofort beginnen.
Stichwort Schaltbild. Gerade dies ist ein Beispiel, bei dem wir in Zukunft unserer Mailbox noch Beine machen können. Denn hier finden wir viele Flächen "schwarz" und einige Linien "weiß" vor. Durch Datenkompression (indem man am Anfang einer Zeile zum Beispiel sagt :200 Punkte schwarz) läßt sich hier noch ein erheblicher Verkürzungsfaktor in der Übertragung von Bildern einbauen. Diese Kompressionsmethode werden wir in einer der nächsten Sendungen vorstellen. Vorausgesetzt, das Übertragen von Bildern wird als Service überhaupt von Ihnen genutzt. In der letzten Sendung begannen wir damit, den von uns erdachten Standard vorzustellen. Wohlgemerkt, es kann ein vorläufiger Standard sein, wenn Ihnen dazu eine bessere Methode einfällt.

Nun zu unserem Prinzip:
Ein beschriebener Computerbildschirm liefert seine Information, indem alle dargestellten Zeichen als kleinste Einheit eine Anzahl von Bildpunkten benutzen. Bei dem einen Rechner ist der Bildschirm in xxx Punkte horizontal und yyy Punkte vertikal aufgeteilt. Ein zweiter Rechner benutzt ein ganz anderes Format. Die Auflösung des Bildschirms ist umso besser, je größer die Anzahl der Bildpunkte horizontal und vertikal ist. Dieses stellt natürlich ein Problem bei der übertragung dar. Wir könnten natürlich Basicprogramme mit Grafikbefehlen ablegen, zum Beispiel zeichnet der Befehl: LINE (1,1)-(300,1) eine waagrechte Linie ===> bei einigen Computern. Andere Computer wiederum benötigen zur gleichen Darstellung den PLOT - Befehl. Also hierüber ist keine Normierung zu finden!
Unser eingeschlagener Weg führt über eine "Krücke" zum Ziel. Wir scannen den Bildschirm Punkt für Punkt ab und legen den Inhalt als eine ASCII-Datei ab. Diese Datei können wir einfach übertragen und zu Hause kann sie mit einem kleinen BASIC- Programm wieder in den Grafikmodus umgewandelt werden. Wie gesagt: kein Rennwagen, dafür aber ein Auto, das jeder fahren kann.

Wie geht das?
Wir stellen uns einmal eine Linie auf dem Bildschirm vergrössert vor. Die könnte so aussehen:

```
.. .... . ... . . . . ..... . . . ... ... .. ... u.s.w..
```

Nunmehr packen wir 6 Bildpunkte als eine Gruppe zusammen und geben dieser Gruppe einen bestimmten Zahlenwert. Aus dem Zahlenwert machen wir dann ein ASCII-Zeichen, also CHR$(Zahl). Dieses Zeichen legen wir in der Bilddatei ab. Es kann dann übertragen werden und zu Hause wieder in eine Punktgruppe zurückverwandelt werden.
Nun einmal im Einzelnen:
Wir untersuchen die ersten 6 Bildpunkte einer Zeile. Sie mögen so aussehen :

```
1 2 3 4 5 6
.   . .   .
```

Nun geben wir jeder Stelle eine Wertigkeit nach dem Muster

```
2↑n:  1 2 3 4 5 6
      .   . .   .
      1 2 4 8 16 32 =Wertigkeit des Punktes
```

Wir addieren die Wertigkeit der gesetzten Punkte. Das macht:
1 + 4 + 8 + 32 = 45
Nur ein einziges Muster, nämlich das oben dargestellte, kann die Zahl 45 ergeben. Deshalb kann ich dies zu Hause auch wieder in ein eindeutiges Muster zerlegen. Wir müssen aber noch etwas berücksichtigen. Bei 6 Punkten "schwarz" würde die Addition 0+0+0+0+0+0 = 0 ergeben. Wenn wir dieses kleinstmögliche Ergebnis in der Datei ablegen und es später wieder auslesen, so sind wir damit im Bereich der Steuerzeichen, die wir möglichst vermeiden müssen. Trick: wir zählen überall 32, den Wert für daß Leerzeichen (SPACE), hinzu. Damit ergibt sich für unser Ergebnis 45 : 45+32 = 77. Umgesetzt in der ASCII-Tabelle steht hierfür das Zeichen "M". Dieses legen wir in der Bilddatei ab. Für alle Punkte "schwarz", also für die Addition 0 ergibt sich entsprechend 0+32 = 32, oder als Zeichen eben " " = SPACE.
Dieses waren die ersten 6 Bildpunkte. Nun kommen die nächsten 6 Bildpunkte. Dann die nächsten, bis die erste Zeile abgelegt ist. Dann kommt die nächste Zeile dran, dann die nächste usw. usw..

Unser Programm zur Rückverwandlung braucht also zwei Schleifen:
Für VERTIKAL = 0 TO ANZAHL BILDPUNKTE VERT.
Für HORIZONTAL = 0 TO ANZAHL BILDPUNKTE HORIZ.
Eine so abgelegte Zeile kann deshalb als Datei so aussehen:

```
M AN/ / ))D %VF ZF%§§§
```

Jedem Buchstaben bzw. Zeichen entsprechen also 6 Bildpunkte.
Nun müssen wir noch ein gemeinsames Kopfformat festlegen, damit wir zum einen erkennen, daß es sich um eine Bildübertragung und nicht um eine Nachricht handelt. Zum anderen wollen wir automatisch das Bildschirmformat des abgelegten Bildes erkennen. Der Anfang der Datei ist deshalb folgendermaßen festgelegt:

```
BILD  (in Grossbuchstaben Super-Calculator GTX  (Zugspitze im Nebel) (Rechner + Beschreibung)
H300 (horizontale Bildpkte. mit führender Null)
V099 (vertikale Bildpkte. mit führender Null)
(leer, für spätere Erweiterung)
(leer, für spätere Erweiterung)
M  §§§ % B T D %%%% (Daten zur Bildbeschreibung)
§ " FGT M BB )
dto.
dto.
§ A G T FE ( JJJ E
(Carriage Return)
(Carriage Return)
```

Die beiden CR am Ende signalisieren das Ende der Bilddatei.

Soweit also zum Prinzip.
Wie komme ich aber später, wenn ich die Daten als Textfile zu Hause habe, wieder an die Wertigkeit der Punkte. Nun, hier kann man sich sicherlich seine eigenen Gedanken machen, doch man wird nach der schnellsten Methode suchen.
Hier mein Vorschlag zur Lösung:
Wir finden wieder das Zeichen "M" vor.
Der Befehl ASC("M") ergibt 77. Davon ziehen wir 32, unseren "Offset" ab. Damit sind wir wieder bei 45 gelandet. Diese Zahl wollen wir in ein eindeutiges Muster zurückverwandeln. Wir nehmen dazu die Vergleichsoperation AND. Schreiben Sie sich doch einmal das kleine Programm:

```
10 A=45
20 PRINT A AND 1
30 PRINT A AND 2
40 PRINT A AND 4
50 PRINT A AND 8
60 PRINT A AND 16
70 PRINT A AND 32
```

Nun, das Ergebnis wird folgendermaßen aussehen:
0 4 8 0 32
Genau das, was wir benötigen. Wenn wir das Ganze in IF-Abfragen kleiden und nach "wahr" oder "nicht wahr" abfragen, so können wir nun unsere Punkte auf dem Bildschirm setzen. Ich kann es hier nur für die Rechner machen, mit denen ich arbeite, bei denen ich die Grafikbefehle kenne (z.B. MBASIC oder GWBASIC). Hier gibt es den Befehl PSET (x,y) zum Setzen eines Punktes. Also sieht die Abfrage bei Einlesen des Zeichens "M", dann bei Umwandlung in den ASCII-Wert =77 und bei Subtraktion des Offsets von 32, also A=45 folgendermaßen aus:

```
IF A AND 1 THEN PSET (X+0,Y) (wahr, Punkt setzen)
IF A AND 2 THEN PSET (X+1,Y) (nicht wahr, Punkt nicht setzen)
IF A AND 4 THEN PSET (X+2,Y) (wahr, .....)
IF A AND 8 THEN PSET (X+3,Y) (wahr, .....)
IF A AND 16 THEN PSET (X+4,Y) (nicht wahr, .....)
IF A AND 32 THEN PSET (X+5,Y) (wahr, .....)
```

Bei anderen Rechnern muß eben der Befehl zum Setzen eines Punktes entsprechend geändert werden.

Ebenso verfährt man dann, wenn man ein Bild in eine Textdatei umwandeln möchte. Hier noch einmal das Format:

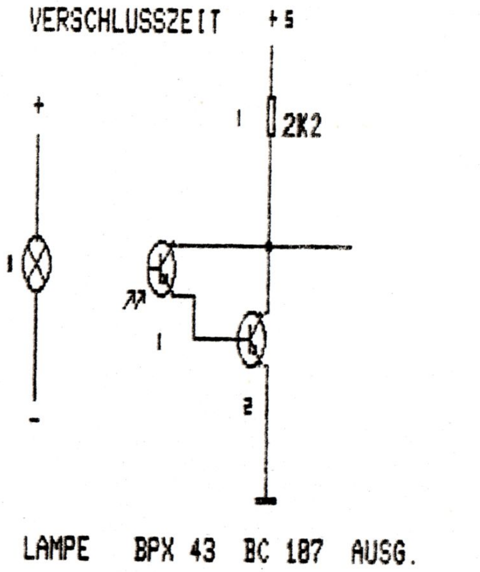
```
BILD
Beschreibung Rechner, Beschreibung Bild
Hnnn
Vnnn

Daten,Daten,CR+LF
dto.
dto.
dto.
Daten,Daten,CR+LF
CR
CR
```

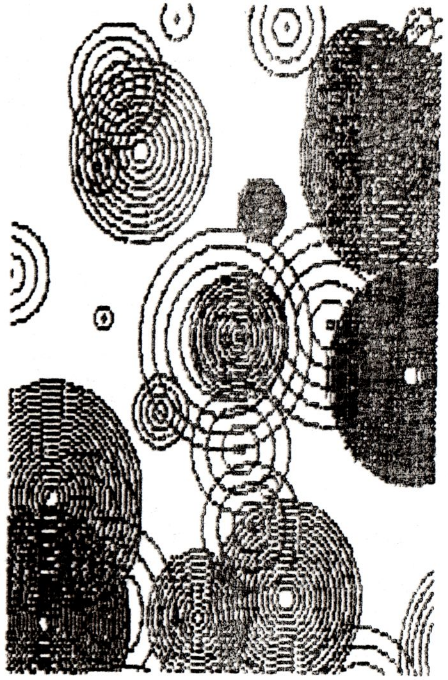
Ein Beispiel wieder für MBASIC bzw. GWBASIC. Hier gibt es den Befehl zum Abfragen eines gesetzten Punktes, der POINT heißt. Also z.B. IF POINT (X,Y)=1 (bei GWBASIC) oder IF POINT (x,y)=7 (bei MFBASIC) gibt bei gesetztem Punkt das Ergebnis "wahr". Somit sieht das Programm dann grob so aus:

```
PRINT #1,"BILD"
PRINT #1,"RECHNER +BESCHREIBUNG"
PRINT #1,"H";H
PRINT #1,"V";V
PRINT #1,""
PRINT #1,""
FOR VERTIKAL = 0 TO V
FOR HORIZONTAL = 0 TO H STEP 6
IF POINT (VERTIKAL ,HORIZONTAL) =1 (7) THEN Z=Z+1
IF POINT (VERTIKAL+1,HORIZONTAL) =1 (7) THEN Z=Z+2
IF POINT (VERTIKAL+2,HORIZONTAL) =1 (7) THEN Z=Z+4
IF POINT (VERTIKAL+3,HORIZONTAL) =1 (7) THEN Z=Z+8
IF POINT (VERTIKAL+4,HORIZONTAL) =1 (7) THEN Z=Z+16
IF POINT (VERTIKAL+5,HORIZONTAL) =1 (7) THEN Z=Z+32
PRINT #1,CHRZ+32);
Z=0
NEXT HORIZONTAL
PRINT #1,""
NEXT VERTIKAL
PRINT #1,CHR13)
PRINT #1,CHR13)
```

Man sollte nur mit INTEGER-Variablen arbeiten, um das Programm schneller zu machen. Also DEFINT oder Variable% benutzen. Beispiele für die Programmgestaltung für verschiedene Rechner findet man im Menuepunkt "BILDUEBERTRAGUNG" im KOMCOM. Wir hoffen natürlich, daß dieser Service Ihren Zuspruch findet und wir schon bald mit der Datenkompression beginnen können. Für Verbesserungsvorschläge sind wir auf jeden Fall dankbar.

Ihr Wolfgang Back.
(Dieser Beitrag wurde der WDR-Mailbox entnommen!)





Verschlusszeitenschaltung (C 64) und "Kreise" (Victor Sirius) aus der WDR-Mailbox (0221 / 371076)

Ein einfaches Grafik-Programm: PLOT/BAS (und seine Nachkommen)

Ich möchte Euch ein kleines Basic-Programm vorstellen, das einer meiner Freunde auf einem C64 auf meine Bitte hin entwickelt hat. Es zeichnet einen Ausschnitt einer dreidimensionalen Funktion mit (vom Beobachtungspunkt abhängigen) verdeckten Linien.

Funktionsweise: Die Achsen des Koordinatensystems sind: x = rechts/links, y = vorne/hinten, z = oben/unten. Vom jeweiligen Punkt (px/py/pz) wird eine Gerade zum Beobachtungspunkt (ox/oy/oz) gezogen. Die Gerade trifft bei y=-20 auf eine Ebene. Der Schnittpunkt wird mit (x/z) bezeichnet, aus dem sich je nach Lage des Ursprungs des Koordinatensystems (xx/zz) der auf dem Bildschirm zu setzende Punkt (sx/sz) berechnet.
Um festzustellen, ob ein Punkt sichtbar ist, schreitet der Computer die Gerade vom Punkt (px/py/pz) bis zum Schnittpunkt mit der Ebene ab. Findet er einen Punkt der Funktion auf der Gerade, so ist der Punkt (px/py/pz) nicht sichtbar und wird nicht gesetzt.

Programmierung:
Es werden folgende grafikspezifische Angaben benutzt, die je nach Grafikkarte und -treiber zu ändern sind:

HON	Grafikbildschirm einblenden
HCLS	" löschen
HMOVE sx,191-sz,2	Punkt invertieren (Grafik-Nullpunkt unten links)
HMOVE sx,191-sz,1	Punkt setzen
191	höchster Y-Wert (vertikal)
479	höchster X-Wert (horizontal)
100/240	Hälfte der obigen Werte

Programmlauf:
Zuerst wird abgefragt, wie groß der anzuzeigende Ausschnitt sein soll. Je tiefer die Funktion in den Raum angezeigt werden soll (y-Richtung) desto länger dauern die Berechnungen!
Es folgt die Abfrage des Beobachtungspunktes. Da die Projektionsebene fest ist, verzerrt sich das Bild mit wachsenden X-Werten erheblich. Empfohlene Größenordnung: 90,800,200 (Blick etwas von rechts, sehr weit entfernt, etwas von oben).
Das Berechnen dauert je nach Größe des Ausschnitts und Blickwinkel mehrere Stunden. Empfohlene Zeit für den Programmlauf: nachts! Zur Beschleunigung dient die Space-Taste: Wenn eine Linie offensichtlich sichtbar oder unsichtbar ist (auf blinkenden Punkt achten!), kann die Berechnung "verdeckt oder nicht" entfallen! Sie wird mit der Space-Taste übergangen. Der Punkt wird gesetzt, wenn der vorherige gesetzt wurde. Ist jemand bei einer verdeckten Linie zu weit vorgestürmt, kann er mit "0" (Taste lange festhalten) einige Punkte zurückspringen. Aber Achtung: schon gesetzte Linien werden nicht gelöscht!

Fragen an den Programmierer über Gerald Schröder

Idee und Programmierung: Stephen Arps
Text: Gerald Schröder

Das Programm: FLOT/BAS

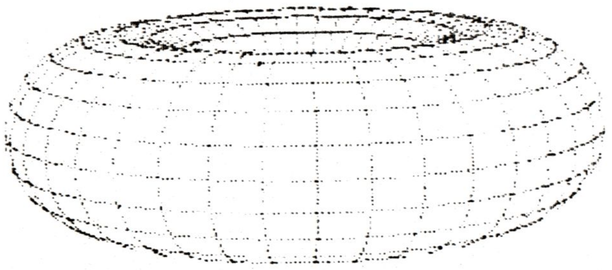
```
5 'PLOTTEN EINER FUNKTION
6 '1986 BY STEPHEN ARPS
7 '
10 CLS:HCLS
50 INPUT"X-RICHTUNG: VON ";VX
60 INPUT"             BIS ";BX
70 IFVX>BXTHEN50
80 INPUT"Y-RICHTUNG: VON ";VY
90 INPUT"             BIS ";BY
100 IFVY>BYTHEN80
110 PRINT"BEOBACHTUNGSPUNKT":INPUT"   X (RECHTS/LINKS)";OX:
    INPUT"   Y (VORNE/HINTEN)";OY:INPUT"   Z (OBEN / UNTEN)";OZ
120 PX=0:PY=0:PZ=0
130 GOSUB600
140 XX=240-X:ZZ=100+Z
150 '
200 HON:CLS
210 FORPY=BYTOVYSTEP-20
220 FORPX=VXTOBXSTEP3
225 IFINKEY$="0"THENPX=PX-6:IFPX<VXTHENPX=VX
230 GOSUB900:GOSUB500:NEXTPX,PY
250 '
300 FORPX=VXTOBXSTEP20
310 FORPY=BYTOVYSTEP-3
315 IFINKEY$="0"THENPY=PY+6:IFPY>BYTHENPY=BY
320 GOSUB900:GOSUB500
330 NEXTPY.PX
400 END
410 '
500 GOSUB600
505 SX=INT(XX+X+.5):SZ=INT(ZZ-Z+.5):IFSX<0ORSZ<0ORSX>479ORSZ>191THEN550
507 HMOVE SX,191-SZ,2
510 IFPEEK(14400)<>128THENGOSUB700
515 HMOVE SX,191-SZ,2
520 IFF=1THEN550
540 HMOVE SX,191-SZ,1
550 RETURN
560 '
600 YP=PY-BY-20:D!=YP/(OY-YP)
610 X=PX-(OX-PX)*D:Z=PZ-(OZ-PZ)*D
620 RETURN
630 '
700 DX=OX-PX:DY=OY-YP:DZ=OZ-PZ:LX=DX/DY:LZ=DZ/DY
710 S1=PX:S2=PY:S3=PZ
720 PY=YP+BY+21:PX=S1+LX
730 GOSUB900:C0=SGN(PZ-S3-LZ):IFC0=0THENF=1:GOTO795
740 LY=YP+1
750 HMOVE SX,191-SZ,2:ZW=LY-YP:PY=LY+BY+20:PX=S1+LX*ZW:HMOVE SX,191-SZ,2:
    IFPX<VXORPX>BXTHEN790
760 GOSUB900:C1=SGN(PZ-S3-LZ*ZW)
770 IFC1<>C0THENF=1:GOTO795
780 LY=LY+3:IFLY<-19THEN750
790 F=0
795 PX=S1:PY=S2:PZ=S3:RETURN
810 '
900 FU=SQR(PX*PX+PY*PY)
910 PZ=3*(4-FU/100)*COS(FU/25-.4)
920 RETURN
```

```
998 'ANDERE FUNKTIONEN
999 '
1000 PZ=PX*PX/200-150
1099 '
1100 PZ=PX*PX/200-PY*PY/200
1199 '
1200 PZ=(-(PX-100)*(PX-100)-PY*PY)/300+100:IFPZ<50THENPZ=50
1299 '
1300 PZ=SIN(PX/10)*SIN(PY/20)*10
1399 '
1400 PZ=3*(5-SQR(PX*PX+PY*PY)/50):IFPZ>120THENPZ=120
1499 '
1500 PZ=1.5*SQR(PX*PX+PY*PY)-100
1597 '
1598 '
1599 ' PYRAMIDE (VX=BY, VY=BY)
1600 AX=ABS(PX):AY=ABS(PY):PZ=AY-70:IFAX>AYTHENPZ=AX-70
```

Nun kann man nicht nur rechteckige Auschnitte von Funktionen plotten, sondern auch runde. Dazu müßt Ihr folgende Änderungen an dem vorstehenden Programm vornehmen:

RUNDPLOT/BAS '

```
5 'PLOTTEN EINES RUNDEN AUSSCHNITTS EINER FUNKTION
20 PI=3.14159
50 INPUT"RADIUS ";A:A2=A*A
60 '
70 '
80 '
90 '
100 '
210 FORRA=ATO1STEP-20
220 FORWI=0TO2*PISTEPPI/180
225 IFINKEY$="0"THENWI=WI-PI/90:IFWI<0THENWI=0
227 T1=SIN(WI):T2=COS(WI)
230 GOSUB895:GOSUB500:NEXTWI,RA
300 FORWI=PI/60TO2*PI+PI/60STEPPI/15
302 T1=SIN(WI):T2=COS(WI)
310 FORRA=ATO0STEP-3
315 IFINKEY$="0"THENRA=RA+6:IFRA>ATHENRA=A
320 GOSUB895:GOSUB500
330 NEXTRA,WI
600 YP=PY-A-20:D=YP/(OY-YP)
720 PY=YP+A+21:PX=S1+LX
750 HMOVE SX,191-SZ,2:ZW=LY-YP:PY=LY+A+20:GOSUB900:HMOVE SX,191-SZ,2:
    IFPX*PX+PY*PY>A2THEN790
760 C1=SGN(PZ-S3-LZ*ZW)
795 RETURN
895 PX=RA*T1:PY=RA*T2
```



Aber sicherlich wollt Ihr auch noch mehr als solche simplen mathematischen Funktionen plotten. Also noch was: der Torus. Am vorstehenden Programm RUNDPLOT müßt Ihr dazu folgendes ändern:

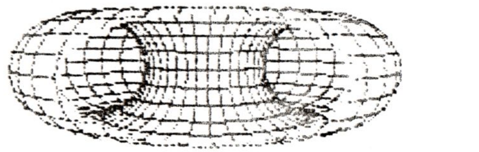
TORUS/BAS

```
5 'TORUS
10 CLS:HCLS:DEFINTA,F,K,O,R:PI!=3.14159
20 '
50 INPUT"RADIUS DES TORUS";A
60 INPUT"RADIUS DES TORUS-KÖRPERS";R
70 IFR>ATHEN50
80 R2=R*R:DI%=A+R+20
210 FORW2=0TO2*PISTEPPI/10
211 T1=R*SIN(W2):T2=R*COS(W2)+A
220 FORW1=0TO2*PISTEPPI/180
225 IFINKEY$="0"THENW1=W1-PI/90:IFW1<0THENW1=0
227 U1=SIN(W1):U2=COS(W1)
230 GOSUB800:GOSUB500:NEXTW1,W2
300 FORW1=PI/80TO2*PI+PI/80STEPPI/20
302 U1=SIN(W1):U2=COS(W1)
310 FORW2=0TO2*PISTEPPI/90
315 IFINKEY$="0"THENW2=W2-PI/45:IFW2<0THENW2=0
317 T1=R*SIN(W2):T2=R*COS(W2)+A
320 GOSUB800:GOSUB500
330 NEXTW2,W1
600 YP=PY-DI%:D=YP/(OY-YP)
720 '
730 '
750 HMOVESX,191-SZ,2:ZW=LY-YP:PY=LY+DI%:PX=S1+LX*ZW:GOSUB900:HMOVESX,191-SZ,2:
    IFK=0THENIFPZ>=S3+LZ*ZWTHENIF-PZ<=S3+LZ*ZWTHENF=1:GOTO795
760 '
770 '
800 PX=T2*U1:PY=T2*U2:PZ=T1:RETURN
900 FU!=SQR(PX*PX+PY*PY)-A:FU!=R2-FU!*FU!:IFFU!<0THENK=1:RETURN
910 K=0:PZ=SQR(FU!):RETURN
```

- alle folgenden Zeilen bis zum Ende löschen!

Nun kann man den Torus auch aufschneiden und in ihn hineinsehen. Das bewerkstelligt das folgende Programm. Ihr müßt diesmal die folgenden Zeilen in TORUS/BAS ändern!

TORUSINN/BAS



```
5 'TORUS AUFGESCHNITTEN
10 CLS:HCLS:PI=3.141
80 R2=R*R:DI=A+R/2
211 T1=R*SIN(W2):T2=A+R*COS(W2)
317 T1=R*SIN(W2):T2=A+R*COS(W2)
501 IFYP>0THENRETURN
600 YP=PY-DI:D=YP/(OY-YP)
750 PY=LY+DI+1:PX=S1+LX:GOSUB900:C0=0:IFK=0THENIFPZ>S3+LZTHENIF-PZ<S3+LZ
    THENC0=1
780 LY=YP+1
781 HMOVESX,191-SZ,2:ZW=LY-YP:PY=LY+DI:PX=S1+LX*ZW:GOSUB900:C1=0:
    HMOVESX,191-SZ,2:IFK=0THENIFPZ>S3+LZ*ZWTHENIF-PZ<S3+LZ*ZWTHENC1=1
782 IFC1<>C0THENF=1:GOTO795
783 LY=LY+3:IFLY<3THEN781
```

Ein Generalist fängt damit an, daß er **WENIG** über **VIELES** weiß. Jm Laufe der Zeit weiß er immer weniger über immer mehr, bis er zuletzt **NICHTS** über **ALLES** weiß.

Ein Spezialist weiß am Anfang ziemlich **VIEL** über ziemlich **WENIG**. Nach und nach weiß er immer **MEHR** über immer **WENIGER**, bis er schließlich **ALLES** über **NICHTS** weiß.

(LORD ELTON)

HEFT 14 Juli 1986

Linienmodelle in 3 Dimensionen

Jeder von Euch kennt Baupläne. Dort sieht man einen Gegenstand meist mit einfachen Linien von mehreren Seiten. Das folgende Basic-Programm kann nun solche Pläne erstellen, indem es einen Gegenstand, den man aus Linien zusammensetzt, aus verschiedenen Blickwinkeln, von verschiedenen Standpunkten, in unterschiedlicher Größe etc. abbildet. Allerdings verschwinden die Linien, die "hinten" liegen, also verdeckt wären, nicht.
Das Programm entstand erst kurz vor Readaktionsschluß und ist noch dementsprechend kurz. Es enthält aber alle wichtigen Routinen. Der weitere Ausbau ist Geschmackssache.

Verwendet wurden folgende Grafikangaben, die abzuändern sind:

HON	Grafikbildschirm einblenden
HCLS	" löschen
HCOLOR 1	alle Linien "SET", also setzen
HPLOT qx,191-qz,0	Punkt als Startpunkt für eine Linie
HPLOT sx,191-sz,1	Linie vom vorherigen Startpunkt bis zu diesem Punkt ziehen
479	höchstzulässiger X-Wert
191	" Y-Wert
240/100	Hälfte der höchstzulässigen X- und Y-Werte

Disk-Ein/Ausgabe wurde für GDOS erstellt.

Idee und Programm: Stephen Arps (Original auf C64)
Text: Gerald Schröder

Fragen zu dem Programm leitet Gerald Schröder weiter.

LINPROG/BAS

```
1 'LINIEN-MODELLE ZEICHNEN
2 '1986 BY STEPHEN ARPS
3 '
10 DIMP1%(400,2),P2%(400,2)
20 CLS:PRINT"DATEN LADEN (J/N)? ";
30 GOSUB1500
40 IFJATHEN550ELSE60
50 '
60 CLS
70 PRINT"SIE MÜSSEN LINIEN DURCH JEWEILS EINEN ANFANGS- UND ENDPUNKT":
   PRINT"DEFINIEREN. AUS DIESEN LINIEN ENTSTEHT IHR LINIENMODELL."
75 PRINT"SIE MÜSSEN DIE KOORDINATEN IN DER FORM 'X,Y,Z' EINGEBEN.":
   PRINT"  X = HORIZONTAL":PRINT"  Y = IN DEN RAUM":PRINT"  Z = VERTIKAL":
   PRINT
80 PRINT"WENN SIE BEI 'VON' NUR <ENTER> EINGEBEN, WERDEN DIE WERTE DES":
   PRINT"LETZTEN 'BIS' GENOMMEN."
90 PRINT"DREI KOMMANDOS SIND BEI DER 'BIS'-EINGABE MöGLICH:":
   PRINT" 'Z,0,0' = VORHERIGE LINIE NOCHMAL EINGEBEN.":
   PRINT" 'E,0,0' = EINGABE BEENDEN.":
   PRINT" 'S,0,0' = SPEICHERN DER BIS JETZT EINGEGEBENEN DATEN."
95 PRINT:INPUT"<ENTER>";FR$
100 NR%=0
110 NR%=NR%+1
120 PRINTNR%;TAB(5);". LINIE";
130 PRINTTAB(14);:INPUT"VON ";P1%(NR%,0),P1%(NR%,1),P1%(NR%,2)
140 P2$="":PRINTTAB(14);:INPUT"BIS ";P2$,P2%(NR%,1),P2%(NR%,2)
150 IFP2$="Z"THENNR%=NR%-2:GOTO190
160 IFP2$="E"THENNR%=NR%-1:GOTO210
170 IFP2$="S"THENNR%=NR%-1:GOSUB500:GOTO190
180 P2%(NR%,0)=VAL(P2$)
190 IFNR%<400THENP1%(NR%+1,0)=P2%(NR%,0):P1%(NR%+1,1)=P2%(NR%,1):
    P1%(NR%+1,2)=P2%(NR%,2):GOTO110
200 '
210 PRINTSTRING$(35,"-")
220 INPUT"BEOBACHTUNGSPUNKT (X,Y,Z) ";OX,OY,OZ
230 INPUT"X-/Z-VERSCHIEBUNG (X,Z) ";X,Z
240 XX=240-X:ZZ=100+Z
250 INPUT"GRößENFAKTOR ";GF:INPUT"DREHPUNKT (X,Y,Z) ";GX,GY,GZ
260 INPUT"WINKEL HORIZONTAL ";WH:INPUT"        VERTIKAL   ";WV
270 WH=WH*3.14159/180:WV=WV*3.14159/180:HS=SIN(WH):HC=COS(WH):VS=SIN(WV):
    VC=COS(WV)
280 '
290 HON:HCLS:CLS:HCOLOR 1
300 FORZA=1TONR%
310 PX=P1%(ZA,0):PY=P1%(ZA,1):PZ=P1%(ZA,2)
320 GOSUB1000
330 GOSUB790
340 S1=GF*X:S2=GF*Z
350 PX=P2%(ZA,0):PY=P2%(ZA,1):PZ=P2%(ZA,2):GOSUB1000
360 GOSUB790:X=GF*X:Z=GF*Z
370 GOSUB700
380 NEXTZA
390 '
400 IFINKEY$=""THEN400
410 HOFF 'Grafik ausblenden
420 PRINT"NOCHMAL (J/N)? ";
430 GOSUB1500
440 IFJATHEN210
450 PRINT"ABSPEICHERN (J/N)? ";
460 GOSUB1500
470 IFJATHENGOSUB500
480 END
490 '
491 '
492 ' FILE SPEICHERN
493 '
500 GOSUB640
510 OPEN"O",1,N$,"MU"
520 PUT1,,,NR%;:FORKL=1TONR%:PUT1,,620
530 NEXTKL:CLOSE:RETURN
540 '
541 ' DATEN LADEN
542 '
550 GOSUB640
560 OPEN"I",1,N$,"MU"
570 GET1,,,NR%;:FORKL=1TONR%:GET1,,620
580 NEXTKL:CLOSE:PRINT"ERWEITERUNG (J/N) ?";
590 GOSUB1500
600 IFJATHEN110ELSE210
609 '
610 'IGEL
611 '
620 P1%(KL,0),P1%(KL,1),P1%(KL,2),
630 P2%(KL,0),P2%(KL,1),P2%(KL,2);
635 '
636 'FILENAMEN EINGEBEN UND BEARBEITEN
637 '
```

```
640 INPUT"FILENAME OHNE EXTENSION ";N$:IFLEN(N$)>8THEN
    N$=LEFT$(N$,8)
650 IFINSTR(N$,"/")>0THENN$=LEFT$(N$,INSTR(N$,"/")-1)
660 N$=N$+"/LIN"
670 RETURN
690 '
691 ' LINIE ZIEHEN, SOWEIT SICHTBAR
692 '
700 QX=INT(XX+S1+.5):QZ=INT(ZZ-S2+.5)
710 SX=INT(XX+X+.5):SZ=INT(ZZ-Z+.5)
720 IFQX<0ORQZ<0ORQX>479ORQZ>191THEN840
730 IFSX<0ORSZ<0ORSX>479ORSZ>191THEN920
740 IFRO=1THENIFSX=QXTHENIFSX=QZTHENRETURN
750 QX=INT(QX):QZ=INT(QZ):SX=INT(SX):SZ=INT(SZ)
760 HPLOT QX,191-QZ,0:HPLOT SX,191-SZ,1
770 RETURN
780 '
790 D=PY/(QY-PY)
800 X=PX-(QX-PX)*D
810 Z=PZ-(QZ-PZ)*D
820 RETURN
830 '
840 RO=0:IFSX<0ORSZ<0ORSX>479ORSZ>191THENRO=1
850 GOSUB1090
860 IFQX<0THENGOSUB1120
870 IFQZ<0THENGOSUB1140
880 IFQX>479THENGOSUB1160
890 IFQZ>191THENGOSUB1180
900 GOTO730
910 '
920 IFQX<0ORQZ<0ORQX>479ORQZ>191THENRETURN
930 GOSUB1100
940 IFSX<0THENGOSUB1200
950 IFSZ<0THENGOSUB1220
960 IFSX>479THENGOSUB1240
970 IFSZ>191THENGOSUB1260
980 GOTO740
990 '
1000 I1=PX-GX:I2=PY-GY:I3=PZ-GZ
1010 ZX=I1*HC+I2*HS:ZY=I2*HC-I1*HS
1020 YZ=ZY*VC-I3*VS:CZ=ZY*VS+I3*VC
1030 PX=GX+ZX:PY=GY+YZ:PZ=CZ+GZ
1040 RETURN
1080 '
1090 K1=SZ-QZ:K2=SX-QX:K3=QX-479:K4=QZ-191:RETURN
1100 K1=QZ-SZ:K2=QX-SX:K3=SX-479:K4=SZ-191:RETURN
1110 '
1120 IFK2<>0THENQZ=QZ-QX*K1/K2:QX=0:GOTO1090
1130 RETURN
1140 IFK1<>0THENQX=QX-QZ*K2/K1:QZ=0:GOTO1090
1150 RETURN
1160 IFK2<>0THENQZ=QZ-K3*K1/K2:QX=479:GOTO1090
1170 RETURN
1180 IFK1<>0THENQX=QX-K4*K2/K1:QZ=191
1190 RETURN
1200 IFK2<>0THENSZ=SZ-SX*K1/K2:SX=0:GOTO1100
1210 RETURN
1220 IFK1<>0THENSX=SX-SZ*K2/K1:SZ=0:GOTO1100
1230 RETURN
1240 IFK2<>0THENSZ=SZ-K3*K1/K2:SX=479:GOTO1100
```



```
1250 RETURN
1260 IFK1<>0THENSX=SX-K4*K2/K1:SZ=191
1270 RETURN
1490 '
1491 'EINGABE JA/NEIN: BEI JA IST VARIABLE JA=TRUE
1492 '
1500 FR$=LEFT$(INKEY$+"*",1):IFINSTR("JjNn",FR$)=0THEN1500
1510 PRINTFR$:IFINSTR("Jj",FR$)=0THENJA=0ELSEJA=1
1520 RETURN
```

Und noch ein kleines Beispiel (jeweils "VON" / "BIS"):

20	-20	-110	50	-50	-110
50	-50	-110	50	-250	-110
50	-250	-110	30	-250	-110
30	-250	-110	30	-310	-110
30	-310	-110	-30	-310	-110
-30	-310	-110	-30	-250	-110
-30	-250	-110	-50	-250	-110
-50	-250	-110	-50	-50	-110
-50	-50	-110	-20	-20	-110
-20	-20	-110	20	-20	-110
20	-20	-110	20	-20	-30
50	-50	-110	50	-50	-30
50	-250	-110	50	-250	-30
30	-250	-110	30	-250	30
30	-310	-110	30	-310	30
-30	-310	-110	-30	-310	30
-30	-250	-110	-30	-250	30
-50	-250	-110	-50	-250	-30
-50	-50	-110	-50	-50	-30
-20	-20	-110	-20	-20	-30
20	-20	-30	50	-50	-30
50	-50	-30	50	-250	-30
30	-250	30	30	-310	30
30	-310	30	-30	-310	30
-30	-310	30	-30	-250	30
-30	-250	30	30	-250	30
-50	-250	-30	-50	-50	-30
-50	-50	-30	-20	-20	-30
-20	-20	-30	20	-20	-30
30	-250	30	0	-280	130
30	-310	30	0	-280	130
-30	-310	30	0	-280	130
-30	-250	30	0	-280	130
50	-250	-30	0	-250	30
-50	-250	-30	0	-250	30
0	-250	30	0	-70	30
0	-70	30	-50	-50	-30
0	-70	30	-20	-20	-30
0	-70	30	20	-20	-30
0	-70	30	50	-50	-30

Empfohlene Werte:
Beobachtungspunkt 0,200,0
Drehpunkt 0,-165,0
Verschiebung 0,-10
Vergrößerung 1.5
Winkel beliebig

Wenn die Uhr mal stört

Der UHR-Befehl unseres DOS ist gewiß eine feine Sache, aber die angezeigte Zeit übertüncht natürlich alles, was in der rechten oberen Bildschirmecke steht. Meist stört das nicht besonders. Im Augenblick aber läuft auf meiner Maschine TECRIPS, und da hätte ich rechts oben ganz gerne den Text anstatt der Zeit. Rechts unten in der Kommandozeile ist dafür Platz genug. Es sind eine Menge weiterer Fälle denkbar, wo das wünschenswert ist.

Hier möchte ich zwei Versionen eines Programms vorstellen, das die Uhrzeit am Ende des Bildschirms anzeigen läßt. Um es ein bißchen komfortabler zu machen, hat es noch drei weitere Fähigkeiten. Gleich zu Beginn prüft es, ob beim Aufruf mit dem Namen UHRUNTEN noch ein Parameter angehängt wurde, z. B. UHRUNTEN,N für "nein". Ist das der Fall, so wird der Patch rückgängig gemacht und die Anzeigestelle wieder an die obere Bildschirmecke gerückt. Dabei ist es übrigens gleichgültig, welches Zeichen man dem Aufruf nachstellt. Zweitens wird automatisch der DOS-Befehl UHR durchgeführt; die Anzeige, wo auch immer, erfolgt also in jedem Falle.

Und drittens wird ein Mangel behoben, wo die Designer von NEWDOS-80 offenbar gepennt haben: Die Uhrzeit wird nicht ganz rechts angezeigt, sondern drei Stellen in die Zeile eingerückt. Das ist zwar nicht viel, aber es ist nicht einzusehen, was das soll. Deshalb wird - oben oder unten - der äußerste rechte Rand zur Anzeige benutzt. Das bleibt auch so, wenn mit UHRUNTEN,N wieder auf Anzeige oben zurückgeschaltet wird. Mit einem einfachen Patch kann das übrigens zurechtgebogen werden, ohne UHRUNTEN zu fahren:

Für das Genie I und seine Geschwister:
in SYS0/SYS, rel. Sekt. 01, Byte 92 von 35 auf 38 zappen

Für das Genie III s:
in OVL4/SYS, rel. Sekt. 09, Byte 53 von 0B auf 08 zappen

Zunächst möchte ich die für beide Versionen gleichen Teile erläutern. Der CALL nach 4CD5h überprüft, ob nach dem Programmnamen noch ein Parameter folgt. Wenn das der Fall ist, kehrt 4CD5h mit der NZ-Bedingung zurück, andernfalls mit Z. Das Entsprechende kann nun ablaufen. Am Ende des Programms wird HL als Zeiger auf das 0Dh-Byte (das entspricht ENTER) geladen. Was das bedeutet, kommt gleich. Danach werden C mit 02h und A mit E5h geladen. E5h ist der Requestcode, der über RST 28h SYS3/SYS aufruft. Der Inhalt von C entscheidet mit darüber, welche der Routinen aus SYS3/SYS angesprungen wird. Für UHR muß es 02h sein. Beim Einsprung in die UHR-Routine wird zuerst ebenfalls 4CD5h geCALLt, um die Parameter zu checken. Wenn in (HL) (Zeiger auf das Ende des Aufrufbefehls) nur ENTER (0Dh) steht, bedeutet das dasselbe wie UHR,J. Deshalb wird mit RST 28h die Uhr nun auf jeden Fall angezeigt, wenn HL auf 0Dh steht.

Nun zur Version für das Genie III s. Ab 3594h steht die Routine zur Anzeige der Uhrzeit. Dort wird u. a. die je nach Bildschirmformat zutreffende Stelle errechnet. Dabei werden vom Beginn der zweiten Zeile 11 Bytes subtrahiert, um nicht ganz ans Ende der ersten zu gelangen. Dieser Subtrahend steht in 35A3h. Dort schreibt das Programm nun stattdessen 8 ein, damit das äußerste Ende benutzt wird. An der Stelle 359Dh steht der Befehl LD DE,(340Bh). In 340Bh ist die Zeilenlänge je nach Bildschirmformat abgelegt. Die Summe von Bildschirmstart und Zeilenlänge ergibt den Beginn der zweiten Zeile (s. o.).

Zur Anzeige rechts unten muß aber die Summe von Bildschirmanfang und seiner Länge errechnet werden, um das Bildschirmende zu erhalten. Die Länge steht in 340Eh. Beide Adressen unterscheiden sich nur im LSB, das entweder 0Bh (oben) oder 0Eh (unten) lautet. Daher genügt es, wenn der Akku in Zeile 7 mit 0Eh oder in Zeile 10 mit 0Bh geladen wird, je nachdem. Dieses LSB wird nun nach 359Fh gepatcht.

Beim Genie I läuft es etwas anders. An der Stelle 44A4h wird HL als Zeiger auf die (fast) rechte obere Bildschirmecke, nämlich 3C3Bh geladen. Wir nehmen lieber 3C38h, um ganz rechts zu landen. Rechts unten lautet diese Adresse 3FF8h. Je nach Wunsch wird nun eine dieser beiden Adressen

nach DE geladen. In der Folge (Zeilen 11-13) patcht das Programm die gewünschte Adresse nach 44A5h, wo der Operand des Ladebefehls steht.

Etliche Clubmitglieder haben die 80-Zeichen-Karte von RB-Elektronik. Es wäre interessant, im nächsten Info eine Version zu finden, die mit dieser Karte läuft. Da es beim Komtek, dem Genie III (ohne "s") und ähnlichen Computern auch Unterschiede geben dürfte, sind wir auf weitere Versionen gespannt.

```
                  00001 ;                    UHRUNTEN/CMD
                  00002 ; Programm zur Anzeige der Uhrzeit in der unteren Ecke
                  00003 ; Version für Genie I, TRS-80 usw.;
                  00004
5200              00005             ORG     5200h
5200  CDD54C      00006 start       CALL    4cd5h          ;folgt ein Parameter?
5203  11F83F      00007             LD      DE,3ff8h       ;Anz. am Bildschirmende
5206  2803        00008             JR      Z,unten        ;falls kein Parameter
5208  11383C      00009             LD      DE,3c38h       ;rechte obere Ecke
520B  21A544      00010 unten       LD      HL,44a5h       ;Adresse für Ladebefehl
520E  73          00011             LD      (HL),E         ;Adreß-LSB des Ladebef.
520F  23          00012             INC     HL             ;auf MSB erhöhen
5210  72          00013             LD      (HL),D         ;MSB patchen
5211  211A52      00014             LD      HL,cr          ;ENTER für UHR-Befehl
5214  0E02        00015             LD      C,02h          ;Zeiger auf UHR in SYS3
5216  3EE5        00016             LD      A,0e5h         ;Requestcode für SYS3
5218  EF          00017             RST     28h            ;aufrufen (= UHR<ENTER>)
5219  C9          00018             RET                    ;zur. ins Betriebssystem
521A  0D          00019 cr          DB      0dh            ;ENTER für UHR-Befehl
5200              00020             END     start

00000 Fehler
```

```
                  00001 ;                    UHRUNTEN/CMD
                  00002 ; Programm zur Anzeige der Uhrzeit in der unteren Ecke
                  00003 ; Version für das Genie III s:
                  00004
5200              00005             ORG     5200h
5200  CDD54C      00006 start       CALL    4cd5h          ;folgt ein Parameter?
5203  3E0E        00007             LD      A,0Eh          ;für 340Eh und 8 Stellen
5205  32A335      00008             LD      (35a3h),A      ;8 Stellen vor Zeilenende
5208  2802        00009             JR      Z,unten        ;falls kein Parameter
520A  3E0B        00010             LD      A,0Bh          ;DE aus 430Bh laden
520C  329F35      00011 unten       LD      (359fh),A      ;Adresse für Ladebefehl
520F  211852      00012             LD      HL,cr          ;ENTER für UHR-Befehl
5212  0E02        00013             LD      C,02h          ;Zeiger auf UHR in SYS3
5214  3EE5        00014             LD      A,0e5h         ;Requestcode für SYS3
5216  EF          00015             RST     28h            ;aufrufen (= UHR<ENTER>)
5217  C9          00016             RET                    ;zur. ins Betriebssystem
5218  0D          00017 cr          DB      0dh            ;ENTER für UHR-Befehl
5200              00018             END     start

00000 Fehler
```

Arnulf Sopp

Nachtrag zu UHRUNTEN/CMD

Bei meinem Programm zur Anzeige der Uhrzeit in der rechten unteren Ecke des Bildschirms habe ich gleich zweimal gepennt. Das soll jetzt richtiggestellt werden:

Der RET-Befehl nach dem RST 28h erübrigt sich, weil RST 28h, obgleich ein Unterprogrammaufruf (wie CALL bzw. GOSUB), wie ein JP (entspr. GOTO) behandelt wird. Das RET stört aber auch nicht weiter.

Viel gravierender ist ein anderer Fehler, der aber nur in der Version für das Genie III s vorkommt: Bei dem Bildschirmformat 24*64 Zeichen erfolgt die Anzeige nicht rechts unten, sondern in einer mittleren Zeile, je nach dem. Das ist mit dem anschließend vorgestellten Listing behoben. Dieses arbeitet bei allen möglichen Bildschirmformaten fehlerfrei. Auf die Programmlogik will ich jetzt nicht eingehen. Die drei Patches, die das Programm setzt, erklären sich aus den nachfolgenden Erläuterungen, hoffe ich.

Das DOS des Genie III s holt seine Informationen über das Bildschirmformat nicht direkt vom Videocontroller, sondern aus dem RAM, wo die wichtigen Daten ähnlich wie in einem DCB niedergelegt sind. Ab 3400h finden wir folgende Daten:

3400/01	physikalische Anfangsadresse des Bildschirms
3402/03	mit PRINT, CLS usw. adressierbare Anfangsadresse
3404/05	physikalische Endadresse des Bildschirms +1
3406/07	wie oben adressierbare Endadresse +1
3408/09	Länge des wie oben adressierbaren Bildschirms
340A	unbenutzt?
340B	Länge einer Bildschirmzeile

Bei der Anzeige der Uhrzeit wird eine Routine angesprungen, die ich zum Verständnis (auch meines Programms) ab 359Ah vorstellen und kommentieren möchte:

```
LD      HL,(3400h)      ;Anfangsadresse des Bildschirms
LD      DE,(340Bh)      ;Anzahl der Zeichen pro Zeile
ADD     HL,DE           ;ergibt Anfang der 2. Zeile
LD      DE,000Bh        ;11 Bytes zurück
SBC     HL,DE           ;ergibt ca. Ende der 1. Zeile
```

Nun zeigt HL auf die Bildschirmstelle, ab wo die Uhrzeit erscheinen soll. Mein Programm subtrahiert nicht 11, sondern 8 Stellen vom Beginn der 2. Zeile, damit der äußerste rechte Rand erreicht wird. Wenn die Uhr unten erscheinen soll, wird nicht der Anfang, sondern eben das Ende des Bildschirms geladen. Dann werden ein paar Bytes übersprungen, um sofort die 8 Stellen abzuziehen.

Der oben beschriebene Fehler trat deswegen auf, weil die erste Version zur Anfangsadresse nicht die tatsächliche Länge des Bildschirms, sondern die mit den Standardbefehlen erreichbare Länge addierte, die in 3408/09h abgelegt ist. Da zeigt sich mal wieder, daß auch der Assembler-Fachidiot gelegentlich sein BASIC anschmeißen sollte, um Fehler zu bemerken, die eben nur unter BASIC auftreten. Und auf der folgenden Seite steht das korrigierte Programm.

Arnulf Sopp


```
                :       U H R U N T E N / C M D
                : Version für das Genie III s. korrigierte Fassung

5200                        ORG     5200h
5200  CD D54C       start   CALL    4cd5h       ;folgt ein Parameter?
5203  3E08                 LD      A,08h       ;8 Stellen vor Zeilenende
5205  32A335                LD      (35a3h),A   ;für Anzeige patchen
5208  0F                    RRCA                ;3404h als Adreßquelle
5209  21180З                LD      HL,0318h    ;= JR 03h
520C  2804                  JR      Z,unten     ;falls kein Parameter
520E  AF                    XOR     A           ;3400h als Adreßquelle
520F  21ED5B                LD      HL,5bedh    ;überschriebener Code
5212  329B35        unten   LD      (359bh),A   ;Adresse für Ladebefehl
5215  229D35                LD      (359dh),HL  ;Programmcode patchen
5218  212052                LD      HL,cr       ;ENTER für UHR-Befehl
521B  0E02                  LD      C,02h       ;Zeiger auf UHR in SYS1
521D  3EE5                  LD      A,0e5h      ;Requestcode für SYS3
521F  EF                    RST     28h         ;aufrufen (= UHR(ENTER))
5220  0D            cr      DB      0dh         ;ENTER für UHR-Befehl
5200                        END     start

00000 Fehler

cr      5220    start 5200    unten 5212
```

Scripsit auf dem "alten" Genie

Schon auf dem Clubtreffen im April wurde ich durch unser Clubmitglied Hans-Martin Stephan auf ein Problem hingewiesen, daß einige Besitzer von VideoGenie 1 - Geräten älterer Bauart haben.

Die ersten Nachbauten des guten alten TRS80 Model 1 haben von Ihren fernöstlichen Geburtshelfern statt eines "memory-mapped" Druckeranschlusses (wie ihn der TRS80 besaß) einen "portgesteuerten" erhalten. Dies hat den Nachteil, daß Programme, die direkt auf die TRS80-Hardware (und in diesem speziellen Fall auf die Druckeradresse 37E8h) zugreifen, auf diesen Geräten nicht laufen! Diesen Mangel stellte man schon bald ab, indem man späteren Auflagen des VideoGenie eine "doppelte" Zugriffsmöglichkeit einbaute. Bei "neueren" VideoGenie 1 - Modellen kann man den Drucker sowohl unter der Memoryadresse 14312 (37E8h) als auch unter der Portadresse 253 (FDh) erreichen.

Den Besitzern der ersten Genies bleibt aber das Problem erhalten, daß z.B. Scripsit auf ihren Maschinen nicht läuft. Abhilfe kann man da auf zwei Arten schaffen:
1. nachträglicher Einbau der Decodierung der Adresse 37E8h
2. Anpassung der Programme an das "abnorme" Gerät!
Obwohl die erste Möglichkeit durchaus realisierbar ist, wird wohl meist die Softwarelösung bevorzugt, die ich aus diesem Grunde hier ausführlich erläutern möchte.

LD contra IN/OUT

LD A,(37E8h) und LD (37E8h),A sind normalerweise die Befehle, mit denen der memory-mapped Druckeranschluß des TRS80 abgefragt bzw. beschickt wird. Mit dem ersten Befehl kann der Druckerstatus abgefragt werden. Der zweite gibt ein im Accu zwischengespeichertes Zeichen an den Drucker aus. Aus den Memoniks (so nennt man die Befehlsworte im Assemblerformat) werden nach der Assemblierung folgende Maschinencodes:

```
LD   A,(37E8h)  -->  3AE837
LD   (37E8H),A  -->  32E837
```

Diese Codes sind mit entsprechenden Hilfsprogrammen (FED, SuperZap usw.) in den umzuschreibenden Programmen sehr leicht zu finden und zu ändern!
IN A,(FDh) und OUT (FDh),A sind die entsprechenden Befehle, die eingesetzt werden müssen. Die Maschinencodes der beiden Befehle sind nur zweistellig, so daß eine Änderung ohne Probleme durchführbar ist! Der Maschinencode 3AE837 ist in DBFD00 (IN A,(FDh) zu ändern, aus dem Code 32E837 wird D3FD00. Da die IN/OUT-Befehle, wie schon erwähnt, nur Zweibytebefehle sind, muß jeweils "00" (NOP = keine Operation) angehängt werden.

Finden und Ändern

Zum Finden der betreffenden Stellen in /CMD-Files und zum Ändern der Codes benutzt man am besten Superzap oder FED (Fileeditor). Der Diskdateneditor DDE aus GDOS ist für diese Aufgabe kaum brauchbar, da mit ihm keine Suchoperationen durchgeführt werden können.
Im Folgenden wird ein Such/Änderungsprozess mit Superzap im Detail durchgesprochen, damit auch diejenigen, die mit dem Zappen nicht so vertraut sind, ohne größere Probleme Programme anpassen können. Als Beispielprogramm nehmen wir SCRIPSIT.

1. Superzap laden.
2. DFS (Display File Sector) aufrufen.
3. Als Filenamen SCRIPSIT/CMD (oder wie das Programm sonst heißen mag) eingeben.
4. Auf die Frage nach dem "File relativ Sector" 0 eingeben.
5. Mit der Eingabereihenfolge "F,3A,E8,37" bzw. "F,32,E8,37" nach den zu ändernden Codes suchen und deren Position (File relativ Sector und relativ Byte im Sector) notieren.
6. Mit der Eingabe "F" nach weiteren Vorkommen des gesuchten Codes fahnden.
7. Wenn alle zu ändernden Stellen gefunden sind, kann mit der Änderung begonnen werden. Dazu den entsprechenden Sector mittels der Tasten "+ und -" aufsuchen und mit dem Kommando "Modxx" in den Änderungsmodus gehen. "xx" steht dabei für das relative Byte im Sector, welches zuvor beim Suchen notiert wurde.
8. Der Cursor blinkt jetzt an der zu ändernden Stelle und man kann die neuen Codes (siehe oben) eingeben. Mit <ENTER> wird die Änderung auf die Diskette geschrieben.
9. Den Änderungsvorgang so lange wiederholen, bis alle gefundenen Codes umgeschrieben sind!

```
DRV 00F E1C1 4704 D1F1 3235 7C21 D27D 7EC9 FE9E ..G...25ö!.üB..
1    10, 2014 1B1A 3C28 AF1B 1A13 3C20 0332 617C  ...<(....< .2aö
1H  203 042B 362E 18A1 FE8E 2006 E1C1 D1C3 765E .+6.........v^
    30A CD2B 5FCB 0102 005F 7928 10FE 3A30 8CFE .+_...._y(..:0.
DRS 40, 2E28 08FE 2C28 04FE 3038 8012 0520 0404 .(..,(..08 .. ..
363 50E C393 5ECD 445F CB71 CA92 5E78 3DC2 925E ..^.D_.q..^x=...
16BH608 1318 95FE 2038 03FE 80D8 E521 7479 3281 ..|. 8.. ..!ty2.
    70, 7923 BE23 20FB 7EE1 C93A 307C 083A 637C y#.# .B..:0ö.:cö
    803 CB7F CA8E 5FE6 60C0 FDCB 0E4E 280A CD84 . _.`....N(..  .
    907 5F08 E67F D3EB 1814 CD04 603A E837 CB7F _........`:.7
    A0  20F6 08FD CB34 6600 00C3 8A7A FE0D 2804  ....4f... ...(.
    B0  FE0A 2004 FD35 4BC9 FD35 4CC9 CD04 60DB .. ..5K..5L...`.
FRS C0  EACB 7728 F7C9 E60F 201C 08CB 7F28 16FE ..w(.... ..(..
13  D0  A030 10E5 2166 7932 7379 23BE 2320 FB7E .0..!fy2sy#.# .B
DH  E0  E118 02CB BF77 23FD 3406 3A35 7CFE 40C9 .....w#.4.:5ö.§.
    F0  E506 41CD 725E E1CD 355A FE59 C8FE 4E20 ..A.r^..5Z.Y..N
```

Scripsit auch für Model 3

Die obenstehende Anleitung ermöglicht es übrigens auch Model 3 - Besitzern Scripsit und natürlich auch die Scripsitabkömmlinge wie z.B. TSCRIPS4 umzuschreiben. Beim Model 3 ist nur statt des Prots FDh der Port F8h einzusetzen.
Zusätzlich muß Scripsit mitgeteilt werden, daß die Speicherstelle in der der HIMEM abgelegt ist, sich von 4049h beim Model 1 auf 4411h beim Model 3 geändert hat. Ansonsten behauptet das Programm unrichtigerweise, daß nicht genügend Speicherplatz vorhanden sei.
Ein zusätzliches Problem stellt sich den Model 3 - Usern mit deutscher Tastatur. Scripsit benutzt nämlich seine eigene Treiberroutine und erzeugt damit eine schier unendliche Verwirrung auf der Tastatur. Dies zu Ändern bedeutet jedoch mehr als nur ein paar Zaps und sprengt den Rahmen dieses Beitrags!

So, ich glaube damit wäre einige Probleme aus der Welt geschafft!? Sollte es Fragen geben werde ich selbstverständlich versuchen sie zu klären, Euer

Hartmut Obermann

Scripsit-Tabellen

Scripsit (oder Tscrips etc.) ist ziemlich offen aufgebaut, denn es benutzt möglichst oft Tabellen, die man meist ohne Probleme ändern kann. So wäre eine Umstellung von Englisch auf Deutsch nicht schwer, aber wie bei Newdos wohl eher unsinnig. Interessanter sind andere Bereiche, die durch einfache Zaps von jedem geändert werden können.

Als erstes wäre eine Tabelle zu nennen, die bei "BREAK P,I", also dem Ausdrucken mit Steuerzeichen, diese Steuerzeichen umwandelt. So wird bei mir u.a. aus der Paragraph-Markierung (auf dem Bildschirm "-") ein "$" auf dem Papier. Eine Verwendung, über die man streiten kann. Wer andere Sonderzeichen auf seinem Drucker ansteuern kann, sollte diese Tabelle bei 7446h vielleicht ändern. In der Tabelle steht immer das Scripsit-interne Steuerzeichen vor dem Zeichen, in das es zum Ausdrucken umgewandelt wird. Das stellt sich so dar:

```
M7446   DB      8Dh,'-'         ;Enter wird zu "-"
        DB      8Eh,'$'         ;Paragraph zu "$"
        DB      97h,'Ä'         ;Blockanfang zu "Ä"
        DB      9Bh,'Ü'         ;Blockende zu "Ü"
        DB      8Ch,'Ö'         ;Seite zu "Ö"
        DB      00h,' '         ;unbekannte Zeichen werden zu Space
```

Bei Bildschirmausgaben gibt es eine ähnliche Tabelle. Wer auf dem Bildschirm andere Zeichen als normal darstellen kann (z.B. mit der 80-Zeichen-Karte) sollte sie unbedingt ändern, denn "Ö" und "Ä" sind sehr mißverständlich. Die Tabelle steht bei 7967h und sieht im Urzustand folgendermaßen aus:

```
M7967   DB      8Dh,8Ch         ;Enter in Grafikblock
        DB      97h,0B7h        ;Blockanfang in Grafik-"("
        DB      9Bh,0BBh        ;Blockende in Grafik-")"
        DB      8Eh,'-'         ;Paragraph in "-"
        DB      8Ch,'Ö'         ;Seite in "Ö"
        DB      8Bh,'Ä'         ;Copy (Break C) in "Ä"
        DB      00h,'Ü'         ;unbekanntes Zeichen in "Ü"
```

Nachdem man "BREAK" eingegeben hat, können zum Beispiel Zeichenfolgen gesucht werden (mit "F>"). Um auch nach bestimmten Steuerzeichen suchen zu können, wurden einige Funktionstasten auch im Break-Modus weiter zugelassen bzw. umcodiert. Die Tabelle dazu sieht folgendermaßen aus:

```
M7975   DB      18h,8Dh         ;§X wird zur "Enter"-Markierung
        DB      03h,8Eh         ;§C wird zur "Paragraph"-M.
        DB      11h,97h         ;§Q wird zu Blockanfang
        DB      16h,8Ch         ;§V wird zur "Seite"-Markierung
        DB      1Ch,9Bh         ;§ DOWN ARROW wird zu Blockende
        DB      9Bh,8Bh         ;Hochpfeil wird zur "Copy"-Markierung
        DB      00h,20h         ;andere werden zu Space
```

Beim Ausdruck werden immer bestimmte Default-Druckparameter angenommen, die im Handbuch auch extra aufgeführt werden. Wem diese nicht passen (z.B. Arnulf und sein LM=0), der kann sie problemlos im Programm selbst zappen. Die Tabelle bei 7A15h ist dafür zuständig. Der Normalzustand:


```
M7A15   DB      3Ch             ;Zeichen/Zeile: 60 (s. auch LM/RM)
        DB      0Ch             ;LM=12 (Linker Rand)
        DB      48h             ;RM=72 (Rechter Rand)
        DB      01h             ;PF=1  (Paragraph-Einschub)
        DB      'Y'             ;J=Y   (Randausgleich an)
        DB      'N'             ;C=N   (keine Zentrierung)
        DB      'N'             ;FR=N  (nicht rechtsbündig schreiben)
        DB      'Y'             ;unbenutzter Parameter
        DB      'Y'             ;WS=Y  (Sinn mir unbekannt)
        DB      01h             ;LS=1  (eine Zwischenzeile)
        DB      06h             ;TM=6  (Druck beginnt in Zeile 6)
        DB      3Ch             ;BM=60 (Druck endet in Zeile 60)
        DB      'N'             ;VC=N  (keine vertikale Zentrierung)
        DB      42h             ;PL=66 (eine Seite hat 66 Zeilen)
        DB      80h             ;H=Y   (sollte nicht geändert werden)
        DB      80h             ;F=Y   (ebenso)
```

Sollte noch jemand spezielle Fragen haben, kann er sich gerne an mich wenden. Ich bin noch dabei, das Listing des Scripsit (bzw. Tscrips) zu kommentieren.

Gerald Schröder

Sicher ist sicher . . .

Freiräume im Tscrips

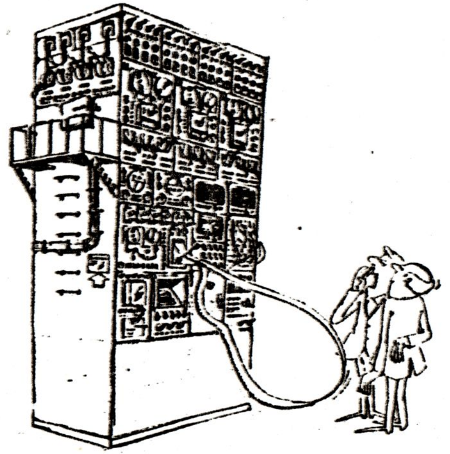
Im Tscrips 4.0 gibt es einige Freiräume, die für jeden interessant sind, der Tscrips erweitern will. Allerdings würde ich für ein ernsthaftes Arbeiten eine vollständige Neu-Assemblierung empfehlen. Aber für kleinere Erweiterungen genügen diese Bereiche, in die man zappen kann. Die Angaben erfolgen jeweils einschließlich des ersten und letzten Bytes, das verändert werden darf.

Bereich	dort stand vorher...
5848h-5869h	nur Quark
61B8h-61A1h	ein Unterprogramm ohne erkennbaren Ansprung
6561h-6567h	Quark
6C17h-6C29h	ein Unterprogramm ohne Ansprung
6FB8h-6FC8h	Text: "TAB COMMAND MODE" (gibt es nicht mehr)
7AA4h-7C28h	Sicherungsroutine für Tscrips 4.0 (vorher 52F2h-5241h mit NOP's füllen; Achtung bei Tscrips 5.4: 7BAEh-7BF5h werden noch anderweitig benutzt, also nicht überschreiben!)
8DF1h-8E4Ah	Quark (nur bei Tscrips 4.0)

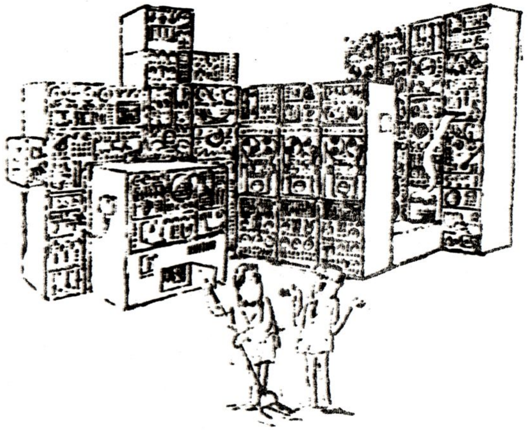
Außerdem sind meines Erachtens die Tape-I/O-Routinen überflüssig, weil niemand ohne Disk Tscrips benutzen kann. Anschließend also noch die Zaps, die die Tape-I/O abschaffen und einen Absturz durch versehentliche Eingabe der Kommandos "S,T", "L,T" oder "V" verhindern.

Gerald Schröder

```
; PATCH #1
; ABSCHAFFEN DER TAPE-I/O
; DADURCH FREIRAUM VON 626Ah BIS 6445h
; LETZTE ÄNDERUNG: 7.4.86 BY GS
;
        ORG     6238H
TEIL1   LD      DE,6F8BH        ;CLEAR-ROUTINE
        LD      (7C29H),DE
        LD      D,H
        LD      E,L
LOOP2   LD      A,(HL)
        OR      A               ;ENDE DES TEXTES? (00)
        RET     Z               ;JA => ZURÜCK
        CP      20H             ;<20H? (FUNKTIONSCODE)
        JR      NC,LOOP1        ;NEIN
        SET     07H,(HL)        ;JA => BIT 7 SETZEN
LOOP1   INC     HL
        JR      LOOP2
;
;
TEIL2   LD      DE,6F8BH        ;CLEAR-ROUTINE
        LD      (7C29H),DE
        XOR     A
        LD      DE,(7C43H)
        LD      (DE),A          ;TEXTENDE MARKIEREN
        LD      (7C2BH),DE      ;TEXTENDE SPEICHERN
        SET     01H,(IY+34H)
        CALL    65B0H
        LD      A,01H           ;"NO MORE ROOM"-ERROR
        JP      6F77H
;
;
;
;       ÄNDERUNGEN IM DISK-I/O-TEIL
;
        ORG     5DB4H
        CALL    TEIL1
;
        ORG     5DBDH
        JP      C,TEIL2
;
;
;       ÄNDERUNGEN IM I/O-AUFRUF
;
        ORG     66EEH
        JP      6F76H           ;BEI TAPE-I/O "INVALID COMMAND"
;
        ORG     66F9H
        JP      6F76H           ;      "
;
;
;       TAPE VERIFY (BREAK V) WEG
;
        ORG     6AF1H
        DB      00
;
        ORG     6AF5H
        DW      0
;
;
;
        END
```

»Um Himmels willen! Jetzt führt er schon Selbstgespräche!«



»Aber Chef, Sie haben doch selbst gesagt, der Computer solle alle Mitarbeiter ihren Fähigkeiten entsprechend einsetzen!«

Textumleitung für Tscrips und andere

In der letzten Ausgabe hat Arnulf ein Programm vorgestellt, mit dem Sektordumps u.ä. in einen Tscrips-Text eingebaut werden können. Leider gab es dabei ein Problem: Tscrips muß schon im Speicher sein und laufen. Wenn ein anderes Anwenderprogramm Tscrips (zer)stört, funktioniert die Umleitung nicht.
Nun wollte ich beweisen, daß Arnulfs und Hartmuts Artikel auch einem Assembler-Neuling etwas bringen und habe mir deshalb meine eigenen Gedanken dazu gemacht. Und siehe da, auf einmal erinnerte ich mich an den von mir noch nie benutzten "ROUTE"-Befehl des Newdos. Da ich vermute, daß Ihr diesen Befehl auch selten benutzt, will ich ihn kurz erklären: ROUTE leitet Ein- oder Ausgaben um. Wenn man zum Beispiel keinen Drucker hat oder kein Papier verschwenden will, kann "ROUTE PR DO" eingegeben werden und alle Druckerausgaben, die auf den Druckertreiber zurückgreifen, werden auf den Bildschirm umgeleitet.
Dieses machte ich mir nun zunutze: mit "ROUTE PR MM=xxxxH" werden Druckerausgaben an eine Routine weitergereicht, die bei xxxxH steht. Ein Beispiel für eine solche Routine seht Ihr unten. Die Routine packt in diesem Fall alles, was eigentlich auf dem Drucker erscheinen würde, in den File "ROUTE/DVC" (u.a. LPRINT und LLIST von BASIC, JKL-Hardcopies, DIR P, H#:* bei Edtasm).
Diese Umleitung wird beendet, wenn Ihr ein 01 an den Drucker schickt (z.B. "LPRINT CHR$(1)"). Dann wird der File geschlossen und die Umleitung durch "ROUTE CLEAR" rückgängig gemacht.
Der entstandene File kann von anderen Programmen benutzt werden. Er läßt sich in dieser Form von Tscrips laden, womit wir wieder den Anschluß an Arnulfs Artikel gefunden hätten.
Ein Nachteil: Es müssen dann noch $ etc. in &24& etc. umgewandelt werden. Wem die Replace-Funktion dazu nicht ausreicht, der kann sich ja noch ein Konvertierungsprogramm schreiben.
Ein Vorteil: Das Programm arbeitet unabhängig von Tscrips. Also lassen sich auch Superzap-Dumps und Edtasm-Programme in einen Text einbauen, allerdings nicht direkt, sondern nur über den Ladebefehl "L.C filename" in Tscrips.
Das nachfolgende Listing wurde folgendermaßen in den Text eingefügt:

1. Listing mit ZEUS erstellen.
2. Abspeichern mit "S*ROUTE" (für Edtasm, weil ZEUS eine eigene Druckroutine besitzt und Übertragung nur durch JKL möglich ist).
3. ZEUS verlassen, "ROUTE/CMD" eingeben, "EDTASM/CMD" eingeben.
4. In Edtasm: "L D=ROUTE/SRC".
5. "H#:*" (Listing an Drucker).
6. Edtasm verlassen.
7. Im GDOS "DR #A" (gibt 01 an den Drucker aus).
8. Tscrips starten und BREAK "L ROUTE/DVC".
9. Text um das Listing herum konstruieren.

Das ganze sieht sehr umständlich aus, ist es aber nicht! Die Möglichkeiten sind fast unbegrenzt. Einige Grenzen: FE00 ff. darf nicht überschrieben werden; Treiberadresse muß unverändert benutzt werden; DIR 1 P ist nicht möglich, wenn dort "ROUTE/DVC" steht (warum? Null Ahnung!); File darf nicht zu lang für Tscrips sein.

Anwendungen und Verbesserungen denkt Ihr Euch am besten selbst aus. Aber berichtet darüber im Clubinfo!

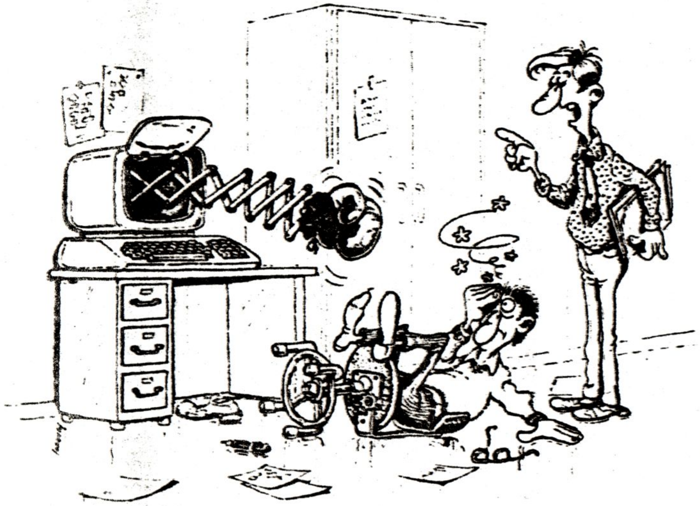
Gerald Schröder


```
;          ROUTE von Gerald Schröder
;          8.6.86
;          LEITET DRUCKERAUSGABE IN DEN FILE "ROUTE/DVC" UM
;          FÜR SPÄTERE NUTZUNG DURCH TSCRIPS
;
;          KONSTANTEN
;
WRIBYT     EQU     001BH            ;ein Byte ausgeben
DOS        EQU     402DH            ;DOS-Abgang
HIMEM      EQU     4049H            ;HIMEM-Zeiger
DERROR     EQU     4409H            ;DOS-Fehlerausgabe
DOSCAL     EQU     4419H            ;ruft DOS-Befehl auf
EXFIL      EQU     441CH            ;Filennamen bearbeiten
DINIT      EQU     4420H            ;File eröffnen
CLOSE      EQU     442BH            ;File schließen
;
           ORG     0E000H
INIT       LD      HL,START-1
           LD      (HIMEM),HL
           LD      HL,FILENAME      ;Filename-Zeiger
           LD      DE,FCB           ;File Control Block-Zeiger
           CALL    EXFIL            ;Filenamen übertragen
;
           LD      DE,FCB           ;Zeiger auf FCB
           LD      HL,BUFFER        ;Zeiger auf Sektor-Buffer (256 Bytes)
           LD      B,00H            ;RECORD-Länge (256)
           CALL    DINIT            ;File eröffnen
           JP      NZ,DERROR        ;Error ausgeben, wenn aufgetreten
;
           LD      HL,ROUTEON       ;Ausgabe-Umleitung
           CALL    DOSCAL
           JP      NZ,DERROR
           RET
;
           ORG     0FE00H
START      DW      00,00,00,00,00,00     ;12 Byte für das DOS
           PUSH    AF
           PUSH    HL
           PUSH    DE
           PUSH    BC
;
           LD      A,C              ;auszugebendes Zeichen nach A
           LD      DE,FCB           ;FCB-Zeiger
           CP      01H              ;Ende der Ausgabe?
           JR      Z,ENDE           ;Ja
;
           CALL    WRIBYT           ;Zeichen ausgeben
           CALL    NZ,DERROR        ;falls Fehler aufgetreten
;
           POP     BC
           POP     DE
           POP     HL
           POP     AF
           RET
;
;
```

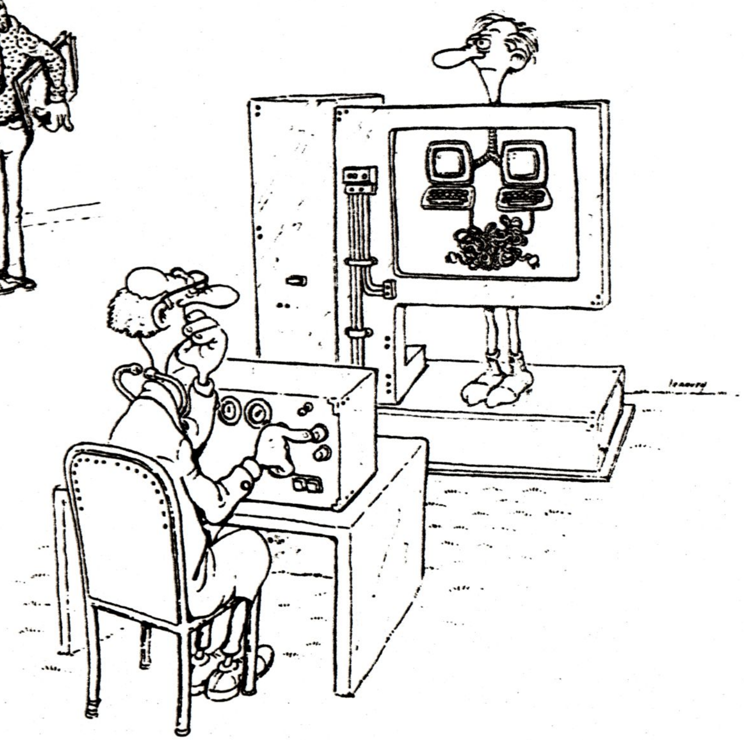
```
ENDE    CALL    CLOSE              ;File schließen
        CALL    NZ,DERROR
;
        LD      HL,0FFFFH          ;HIMEM wieder zurück
        LD      (HIMEM),HL
;
        LD      HL,ROUTEOFF        ;Ausgabe-Umleitung zurück
        CALL    DOSCAL
        CALL    NZ,DERROR
        JP      DOS                ;Absprung ins DOS
;
;
FILENAME        DM      'ROUTE/DVC',0DH
ROUTEON         DM      '§ DR ST=FE00H',0DH
ROUTEOFF        DM      '§ KEINE',0DH
FCB             DS      32
BUFFER          DS      256
;
        END     INIT
```

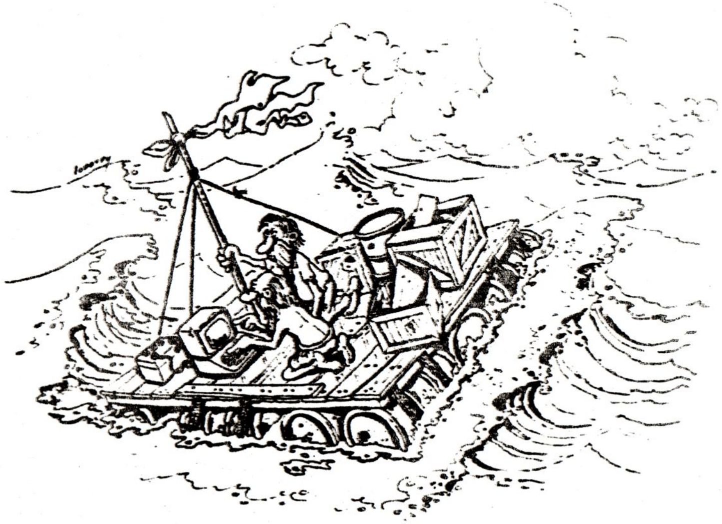
Anmerkung: Die Ausgabeumleitung gilt für GDOS. Bei Newdos heißen die Befehle:
ROUTEON: "ROUTE PR MM=FE00H"
ROUTEOFF: "ROUTE CLEAR"

Er sagt: „Verlange Barzahlung, der überlebt die Nacht nicht mehr."



Das versucht er mit jedem. Zeig ihm, wer der Boß ist!





Die Chance gerettet zu werden steht drei Milliarden zu eins.
Gut, daß wir den Computer haben, sonst hätten wir es nie erfahren!

Nachtrag zu "Sondertasten ..."

In meinem Beitrag "Sondertasten diverser Genies" fragte ich, ob jemand schon herausgefunden habe, wie die Doppelnull erzeugt wird. Wieso habe ich mir diese Frage nicht selber gestellt? Inzwischen habe ich es nämlich raus:

Auf der nächsten Seite ist oben ein Listing abgedruckt. Der obere Teil ist die Routine, die die Doppelnull auf den Bildschirm bringt. Da wird zunächst die Anzahl der zusätzlich zu printenden Zeichen, hier 1, nach 4019h geladen. Dort ist normalerweise für die F-Tasten die Länge ihrer Strings abgelegt. Die Adresse des einen zusätzlichen Zeichens, 3E54h, wird nach 401Ah übertragen. Dort ist sonst der Beginn eines F-Strings gespeichert. Bei 3E47h geht es dann weiter, wo alles angezeigt wird.

Jede dieser Informationen kann geändert werden: Die um 1 verminderte Anzahl der Zeichen darf beliebig sein. Sie wird entsprechend nach 3E4Fh gepatcht. Das erste anzuzeigende Zeichen in 3E54h darf auch beliebig lauten. Schließlich kann man in 3E56/57h die Adresse des zweiten und der folgenden Zeichen ebenfalls nach Gusto ändern. Auf diese Weise ist es möglich, die 00-Taste wie eine zusätzliche F-Taste zu gebrauchen. Ihr String kann beliebig lang sein (solange er kürzer als 256 Bytes ist). Sie ist auch abschaltbar, wenn in 3E54h z. B. eine logische 00 oder ein anderer unwirksamer Code steht. In diesem Falle kann die 00-Taste z. B. während einer INT-Bearbeitung als Steuertaste für jedwede Funktion dienen, ohne daß sie außerhalb der INT-Routine stören würde.

Das Erkennen und Anzeigen der PRINT-Taste ist im unteren Teil des Listings wiedergegeben. Wie man sie abschalten oder anders belegen kann, habe ich bereits im o. g. Artikel beschrieben. Zusätzlich hier ein Tip, wie ihr ebenfalls ein beliebig langer Code zugeordnet werden kann: Die Anzahl der Zeichen ist in 3E6Eh abgelegt. Sie darf geändert werden. Die Adresse des Strings auch, die in 3E70h steht. Man definiert einen eigenen String und patcht seine Adresse dorthin.

Die gelistete Routine steht "neben" dem Bildwiederholspeicher in der Nachbarbank. Den Änderungen muß deshalb ein entsprechendes Banking vorausgehen: Beim Genie III s muß beim Systembyte 1 (Port FAh) dazu das Bit 0 gesetzt werden. Hierzu genügt ein CALL 06A0h. Jetzt sind die Tastatur und der Bildschirm ausgeblendet, stattdessen ist das RAM zugänglich. Nach der Manipulation wird das Bit 0 wieder zurückgesetzt, wofür ein CALL 06ABh langt. Es empfiehlt sich, zuvor die Interrupts mit DI zu sperren, weil evtl. Tastatur- oder Bildschirmzugriffe (Uhr usw.) sonst das Nachbar-RAM adressieren.

In den beiden Sektordumps unter dem Listing ist alles unterstrichen, was hier zur Änderung vorgeschlagen wird. Das sind die Sektoren 04 und 05 von OVL4/SYS für das Genie III s, gebankte G-DOS-Version. Bei den anderen Overlays für die anderen Genies wird man die entsprechenden Codes leicht finden können, weil in ihrer Umgebung so ziemlich dasselbe stehen dürfte.

Arnulf Sopp


```
3E4E  3E01      00224          LD    A,01h        ;1 zusätzliches Zeichen
3E50  321940    00225          LD    (fcount),A   ;als Zeichenzähler ableg.
3E53  3E30      00226          LD    A,'0'        ;zweite Null bei 00-Taste
3E54            00227 key00    EQU   $-1          ;zu verdoppelndes Zeichen
3E55  21543E    00228          LD    HL,key00     ;Adr. des Doppelzeichens
3E58  221A40    00229          LD    (faddr),HL   ;als Stringadresse ableg.
3E5B  18EA      00230          JR    m3e47        ;dort weiter
                00231
3E66  FE3F      00238          CP    '?'          ;PRINT-Taste gedrückt?
3E68  C0        00239          RET   NZ           ;falls nein
3E69  CD7E3D    00240          CALL  m3d7e        ;Cursor aus?
3E6C  C8        00241          RET   Z            ;falls ja
3E6D  3E06      00242          LD    A,06h        ;6 Zeichen von PRINT
3E6F  21753E    00243          LD    HL,kyPRINT   ;Belegg. der PRINT-Taste
3E72  C3763D    00244          JP    m3d76        ;dort weiter
3E75  50        00245 kyPRINT  DM    'PRINT '     ;Belegung PRINT-Taste

00000 Fehler
```

```
DRV 00  003C 7E23 0528 0485 6F18 F732 1940 221A .<B#.(..o..2.§".
1   10  40AF C94F 3A22 40B7 79C8 3A6E 43CB 4779 §..O:"§.y.:nC.Gy
1H  20  C9CD 993D ED4B 4845 CD21 91C3 4A45 21B5 ...=.KHE.!..JE!.
    30  3DCD 2101 5FAE 73A3 200C 7AC6 0857 2379 =.!._.s...z..W#y
DRS 40  C620 4F20 ECC9 E1C3 5E45 0000 0000 280E ..O.....^E....(.
864 50  2100 002B 22BC 3D7C B53E 00C2 8945 2100 !..+".=ö.>...E!.
360H60  0022 BC3D C355 4521 0000 22BC 3D32 4F45 .".=.UE!..".=2OE
    70  AFC9 219F 377A 2334 3528 05ED A120 F77E ..!.7z#45(.....~
    80  FE0B 2806 FE41 DBFE 5FD0 4FCD 003D E642 ..(..A.._.O..=.B
    90  79C8 E61F FE0B C001 4038 CD21 01E6 083E y.......§8.!...>
    A0  0BC9 3E5B C9D6 3E38 79C8 3DC8 2118 0136 ..>Ä..98y.=.!..6
    B0  C9E5 CD24 3EE1 36CD C9FE 0730 1207 4FCD ...$>.6....0..O.
FRS C0  0102 3C4F 003D 0F79 3001 3C21 7B3E CD46 ..<O.=.y0.<!ä>.F
4   D0  047A C9C6 79FE 88D8 E67F FE12 3007 C628 .z..y.......0..
4H  E0  CD4B 047A C920 0F3E 0132 1940 3E30 2154 .K.z...>.2.§>0!T
    F0  3E22 1A40 18EA D613 C821 883E CD35 3EFE >".§......!.>.5>.
```

```
DRV 00  3EC0 CD7E 3DC8 3E06 2175 3EC3 763D 5052 >..~=.>.!u>.v=PR
1   10  494E 5420 0000 1B1B 0A1A 0B1B 2B2B 0000 INT.........++..
1H  20  0000 2C2D 2E3F AF32 2104 CD00 3DC3 0E04 ...-.?.2!...=...
    30  CDB5 0621 4140 0E02 3E23 32BC 3E16 0A06 ...!A§..>#2.>...
DRS 40  037E D96F 2600 3E0A CDB4 4CD9 CD19 3FD9 .~.o&.>...L...?.
865 50  7DD9 CD19 3F23 10E9 167A 3E2B 32BC 3E2E ü...?#...z>+2.>.
361H60  460D 20DB AF16 0ACD 193F CDBE 06DD E5DD F.........?.....
    70  212C 3F2E 440E 023E 2332 F53E 16CC 0603 !.?.D..>#2.>....
    80  CD0A 3F87 5F87 9783 5FCD 0A3F 8377 2310 ..?._..._..?.w#.
    90  EF16 503E 2B32 F53E 2E43 0D20 E1DD E1AF ..P>+2.>.C......
    A0  D35B C97A D35B 1610 9257 DD23 DB5A DDA6 .Ä.z.Ä...W.#.Z..
    B0  00C9 5F7A D35B 1610 9257 E6F0 FE60 7B20 .._z.Ä...W...`ä.
FRS C0  02C8 3FD3 0110 3C50 EAC9 0F0F 010F 030F ......<PZ.......
5   D0  030F 070F 070F 010F 0645 00F5 3A05 45EE .........E..:.E.
5H  E0  2032 7304 F100 0001 0C5E 04ED 5B41 36F5 .2s......^..[A6.
    F0  19F1 0342 3401 2780 04E5 ED5B 4136 B7ED ...B4.'....[A6..
```

Unterschiede

Wer, wie z.B. ich, von einem Model 1 auf ein Model 3 (bzw. Model 4/4P im Model 3-Mode) umsteigt, wird mit dem Problem konfrontiert, daß manche Programme, die man auf dem Model 1 schätzen gelernt hat, nicht mehr laufen. Um diese Programme umzuschreiben, benötigt man, außer etwas Erfahrung im Umgang mit der Maschinensprache, eine Auflistung der Unterschiede zwischen den beiden Maschinen.

In einer kleinen Serie von Artikeln möchte ich diese Unterschiede aufzeigen. Den Anfang machen die ROM-Differenzen, danach (ev. noch in dieser Ausgabe) folgen die Tastaturabweichungen. Ein Beitrag über die NewDOS-Unterschiede wird im nächsten Info erscheinen. Weiterhin werde ich für verschiedene, von mir angepasste Programme, Patchs veröffentlichen. Hier also zunächst die ROM-Unterschiede!

Wenn man sich die beiden Röckrath'schen ROM-Listings genauer betrachtet, sind natürlich erhebliche, meist hardwarebedingte Unterschiede festzustellen. Betrachtet man dagegen nur die Adressenauflistung der Unterroutinen, so wird man feststellen, daß sich hier nur wenig geändert hat. Die ROM-Macher haben also, um die Model 1-Kompatibilität zu wahren, zwar den Programmcode geändert, aber wo möglich die Einsprung -Adressen und -Bedingungen beibehalten. Die untenstehende Liste führt alle Änderungen und Erweiterungen auf.

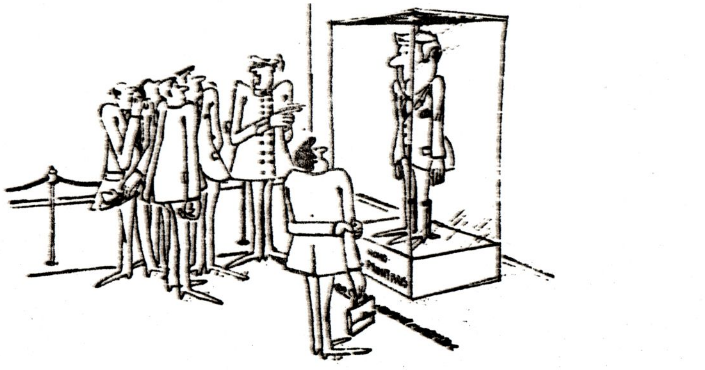
ADDR	M. 1-ROM	M. 3-ROM	Bedeutung
0050	-	RSRCV	Holen eines Zeichens von der RS-232 (BREAK bewirkt Sprung über 4203 nach 1A19)
0055	-	RSTX	Ausgabe eines Zeichens an die RS-232 (BREAK bewirkt Sprung über 4203 nach 1A19)
005A	-	RSINIT	Initialisierung der RS-232
0069	-	INITIO	Initialisiert alle I/O-DCB's einschl. I/O-Routes
006C	-	ROUTE	Führt Routing der I/O-Treiber durch (gibt es nur in der amerikanischen Version!)
01D9	PULSE	*31A5*	Gibt Impuls auf Kassette aus
01D9	-	PRSCN	Druckt Bildschirm aus (Codes 80-FF als "."!)
01FE	CASNO	-	Decodiert Kassettenrecorder-Nr. und schaltet Kassettenrecorder ein
0214	-	CRPR	CR (Carrige Returne) auf Drucker ausgeben
0215	CASON	*31E8*	Schaltet Kassettenrecorder ein
021B	-	TEXT	Text auf Video ausgeben
021E	RESINP	-	Setzt den Eingabeport Bit 7 von Port FF zurück
0221	PORTCT	-	Steuert Port FF
022C	BLINK	-	Schaltet Stern in Bildschirmecke um
0241	READBI	*3220*	Liest Bit von Kassette
0287	WRITES	-	Schreibt Syncronbytes
028D	-	BREAK	BREAK-Taste abfragen
0296	READS	-	Liest Syncronisation
029B	-	CLKON	Uhrzeit jede Sekunde anzeigen
02A1	-	CLKOFF	Uhrzeit-Anzeige abschalten
03C2	CALL	*0674*	Ruft I/O-Routinen über DCB auf
044B	*05D1*	PRRDY	Testet ob Drucker bereit
05D1	PRRDY	*044B*	Testet ob Drucker bereit
0674	*03C2*	CALL	Ruft I/O-Routinen über DCB auf

ADDR	M. 1-ROM	M. 3-ROM	Bedeutung
3033	-	DATE	Datum ab (HL) abspeichern
3036	-	TIME	Zeit ab (HL) abspeichern
3042	-	SETCAS	Fragt den Benutzer nach der Kassetten-BAUD-Rate (gibt es nur in der amerik. Version!)
31A5	*01D9*	PULSE	Gibt Inpuls auf Kassette aus (500 BAUD)
31D1	-	MOTON	DE und BC retten, Kassettenrec. einschalten
31E8	*0215*	CASON	Kassettenrecorder einschalten
3220	*0241*	READBI	Liest Bit von Kassette (500 BAUD, BREAK bewirkt einen Sprung über 4203 nach 1A19)
3335	-	WRITBI	Bit auf Kassette schreiben (1500 BAUD)
3350	-	WAITBI	Auf nächstes Bit von Kassette warten (1500 BAUD, BREAK bewirkt einen Sprung über 4203 nach 1A19)
337C	-	AUSWBI	Wartezeit für gelesenes Bit von Kassette auswerten (1500 BAUD)
3505	-	PRNCND	Druckt Bildschirm aus (Codes 80 - 8F als ".". Gibt es nur in der deutschen Vers.!)

Erklärung zu Spalte 2 und 3:
"-" = Routine in der ROM-Version nicht vorhanden!
xxxx = Routine in der ROM-Version an anderer Stelle!

Wie schon erwähnt sind viele Probleme im Model 3 erheblich anders gelöst als im Model 1. Ein gravierendes Beispiel hierfür ist die Abfrage der Tastatur. Diese Lösungswege interessieren aber bei der Anpassung von Programmen nur selten und sollen deshalb hier auch weiter nicht erwähnt werden. Ich glaube, daß mit der obenstehenden Liste schon die meisten Anpassprobleme zu lösen sind. Sollte jemand weitere Unterschiede in Erfahrung bringen oder schon Progamme angepasst haben, möge er doch bitte sein Wissen im Info weitergeben! Viel Spaß beim Patchen euer

Hartmut Obermann



»Beachten Sie bitte die Schädelbildung aus der Zeit vor der Erfindung des Computers«

Knöpfe

Die Tastatur aller TRS80-Modelle und auch der Kompatiblen ist, wie schon mehrfach im Clubinfo festgestellt, nichts weiter als eine Matrix aus lauter Schließern. Leider sind die einzelnen Tasten bei den unterschiedlichen Geräten in der Matrix unterschiedlich angeordnet. So besitzt das M. 1 z.B. zwar zwei SHIFT-Tasten, diese sind jedoch elektrisch parallelgeschaltet. Beim Model 3/4/4P hingegen wird zwischen zwei SHIFT-Tasten unterschieden. Und sogar innerhalb eines Models gibt es erhebliche Unterschiede, wenn man sich eine "amerikanische" und eine "deutsche" Tastatur ansieht! Diese unterschiedliche Belegung der Matrix kann bei bestimmten Programmen (SCRIPSIT, FED usw.) zu Problemen führen. Aus diesem Grund sind hier einmal die Belegungslisten aller TRS-80- und VideoGenie-Geräte veröffentlicht.

Tastaturmatrix TRS 80 Model 1, Genie I und II:

Adresse:	I	Datenleitung Bit 0	1	2	3	4	5	6	7
3801h	I	ALPHA	A	B	C	D	E	F	G
3802h	I	H	I	J	K	L	M	N	O
3804h	I	P	Q	R	S	T	U	V	W
3808h	I	X	Y	Z					
3810h	I	0	1	2	3	4	5	6	7
3820h	I	8	9	:	;	,	-	*	/
3840h	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHTS	SPACE
3880h	I	SHIFT							

Tastaturmatrix Genie III:

Adresse:	I	Datenleitung Bit 0	1	2	3	4	5	6	7
3801h	I		A	B	C	D	E	F	G
3802h	I	H	I	J	K	L	M	N	O
3804h	I	P	Q	R	S	T	U	V	W
3808h	I	X	Y	Z	;	Ä	:	ALPHA	
3810h	I	0	1	2	3	4	5	6	7
3820h	I	8	9	ß	ö	,	-	.	/
3840h	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHTS	SPACE
3880h	I	SHIFT	CTRL						LP
38A0h	I	F1	F2	F3	F4	F5	F6	F7	F8
38C0h	I	0	1	2	3	4	5	6	7
38E0h	I	8	9	00	LOCK	,	-	.	

Tastaturmatrix Genie IIIs:

Adresse:	I	Datenleitung Bit 0	1	2	3	4	5	6	7
3801h	I	ALPHA	A	B	C	D	E	F	G
3802h	I	H	I	J	K	L	M	N	O
3804h	I	P	Q	R	S	T	U	V	W
3808h	I	X	Y	Z	Ä	ö	ü	ß	_
3810h	I	0	1	2	3	4	5	6	7
3820h	I	8	9	:	;	,	-	.	/
3840h	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHTS	SPACE
3880h	I	SHIFT	CTRL	ESC	P5	P4	P3	P2	P1
38A0h	I	F1	F2	F3	F4	F5	F6	F7	F8
38C0h	I	0	1	2	3	4	5	6	7
38E0h	I	8	9	00	LOCK	,	-	.	PRINT

Tastaturmatrix TRS 80 Model 3 (amerik. Version)

Adresse:	I	Datenleitung Bit 0	1	2	3	4	5	6	7
3801h	I	ALPHA	A	B	C	D	E	F	G
3802h	I	H	I	J	K	L	M	N	O
3804h	I	P	Q	R	S	T	U	V	W
3808h	I	X	Y	Z					
3810h	I	0	1	2	3	4	5	6	7
3820h	I	8	9	:	;	,	-	*	/
3840h	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHTS	SPACE
3880h	I	LSHFT	RSHFT						

Tastaturmatrix TRS 80 Model 3 (deutsche Version)

Adresse:	I	Datenleitung Bit 0	1	2	3	4	5	6	7
3801h	I	>	A	B	C	D	E	F	G
3802h	I	H	I	J	K	L	M	N	O
3804h	I	P	Q	R	S	T	U	V	W
3808h	I	X	Z	Y	ü	+	ö	Ä	#
3810h	I	0	1	2	3	4	5	6	7
3820h	I	8	9	ß	'	ALPHA	,	.	-
3840h	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHTS	SPACE
3880h	I	LSHFT	RSHFT						

Tastaturmatrix TRS 80 Model 4/4P

Adresse:	I	Datenleitung Bit 0	1	2	3	4	5	6	7
3880h	I	LSHFT	RSHFT	CTRL	CAPS	F1	F2	F3	

Hier wird nur die letzte Adressenreihe angegeben, da sich ansonsten gegenüber der Model 3 - Tastaturmatrix (amerik. und deutsch) keine Änderungen ergeben haben!

Hier noch ein kleines Beispiel:
Beim DEBUG des NewDOS läßt sich durch Eingabe von <+> und <ENTER> eine Seite vor-, und durch Eingabe von <-> und <ENTER> eine Seite zurückblättern. Dies funktioniert beim Model 3 nicht bzw. nicht richtig! Beim Model 1 liegt das <+> über dem <;>, während es beim Model 3/4 zusammen mit dem <*> auf einer Taste liegt und ohne Shift erreichbar ist. DEBUG fragt aber in beiden Versionen (Mod.1 und 3 NewDOS) ab, ob das <;> gedrückt wurde. Da beim Model 3/4 das <;> über dem <,> liegt, muß man <SHIFT><,> und <ENTER> drücken, um den gewünschten Erfolg zu haben! Damit man auch beim M.3/4 mit <+> arbeiten kann, ist folgendes Mini-Patch nötig:
SYS5/SYS Rel.Sector 1 Rel. Byte FD
von FE 3B in FE 2B ändern

So einfach ist manchmal das Ausmerzen unschöner Fehler und Schwächen, probierts doch mal! Viel Glück euer

Hartmut Obermann

m4/4p ecke

Aufgestiegen?

(Günther Wagner)

Auch ich gehöre jetzt seit 2 Wochen zum erlauchtem Kreis der Model 4 - Besitzer. Ich selbst habe einen 4P wobei das P nicht für Profi sondern für transportabel steht. Lange Zeit habe ich mir auch einen IBM-Kompatiblen überlegt - vor allem die vorhandene und dann nicht mehr lauffähige Software hat mich davon abgehalten.

Mein altes Model III konnte ich (mit 2 Laufwerken 40/ss/dd) für 1300 DM recht gut verkaufen.
Das neue 4P bekam ich sehr günstig und ich habe bei dieser Gelegenheit auch eine neue interessante Adresse aufgetan.

Ein Spezialist für TRS-80 ist : Anthony Barber
Computer Stube
Fuchskamp 4
3032 Fallingsbostel
Tel. 05162 - 3818

Warum ich für diese Adresse Reklame mache ? Ganz einfach - alle Mitglieder des CLUB 80 erhalten 3 % Skonto. Ein Beispiel:

Das Model 4P kostet bei A. Barber mit 128 K RAM inklusive aller Nebenkosten (wie Versand) 1798 DM (schon sehr günstig) - nach Abzug der 3 % Skonto bezahlt man dann 1744 DM (Preise vom Mai).

Man bekommt bei A. Barber außer Hardware auch die nötige Software für TRS-DOS 1.3, 6.2 und für CP/M. Wer was braucht, kann sich ja mal das Angebot zusenden lassen.

Nun aber zurück zum eigentlichen Artikel:

Bereits beim Studium des neuen Handbuches (sehr umfangreich, meist sehr gut gemacht, manchmal etwas merkwürdig ins Deutsche übersetzt) konnte ich mit großer Freude feststellen, daß das neue TRS-DOS 6.2 ein hervorragendes Betriebssystem ist, welches speziell auf die Möglichkeiten des 4P zugeschnitten ist.

Man kann es auch so ausdrücken: Vergleicht man das TRS-DOS 1.3 mit einem VW-Käfer so hat man mit NEWDOS-80 einen VW-Golf und mit TRS-DOS 6.2 einen BMW oder Mercedes.

Mit sehr einfachen Befehlen kann man eine RAM-Disk initialisieren. Ebenso einfach ist das Einrichten dieser RAM-Disk als System-Laufwerk. Das Arbeiten mit der RAM-Disk ist natürlich viel schneller als mit normaler Diskette.

Ebenso einfach ist das Einrichten des Spoolers. Einfachste Befehle erlauben das Spoolen in den Speicher und/oder in einen File. Nur das Einschalten des Spoolers muß man besorgen - um alles andere kümmert sich das DOS.
Das DOS unterstützt auch sehr gut (wenn auch etwas umständlich) die Kommunikation über die RS232.
Nicht vergessen möchte ich die 'Job Control Language' (JCL), eine eigene Programmiersprache im DOS mit sehr guten Möglichkeiten.
Sehr gut gefällt mir die Verwendung von Drucker, Tastatur, Bildschirm etc. als logische (und gleichwertige) Geräte, die ohne weiteres miteinander verbunden werden können. Dabei können hier Filterprogramme zwischengeschaltet werden, die es z.B. ermöglichen, bestimmte Zeichen zu unterdrücken, abzuändern oder zusätzliche Zeichen auszugeben.
Das weitere Aufzählen wirklich leistungsfähiger Merkmale würde hier zu weit führen - erlaubt sei mir noch ein Hinweis auf das vorzügliche BASIC, welches mehrere neue leistungsfähige Befehle aufweist. Es handelt sich hierbei um Microsoft-Basic Version 5.

Will man Model III - Software laufen lassen, so muß man sich zunächst ein Model III - ROM-Image von Diskette einladen. Anschließend kann man ganz normal die Model III - System-Diskette einlegen. Bisher lief bei meinem Model 4 im Model III - Modus sämtliche Software, welche auch beim Model III lief. Und diese Software läuft sogar doppelt so schnell. Man muß einfach über die Tastenfolge 1-2-3 mit M 4210 auf die betreffende Adresse gehen und dort 20 in 60 abändern und siehe da - die Model III - Software läuft mit 4 MHz.

Auch CP/M läuft auf dieser wirklich feinen Maschine. Hierzu kann ich mich allerdings noch nicht weiter auslassen, da ich mit CP/M noch nicht viel gemacht habe.

Fazit : Ich betrachte mich als echten Aufsteiger - ich habe jetzt einen transportablen, formschönen Computer mit 128 K in dem eigentlich 3 Computer stecken:

- ein Model III
- ein Model 4P mit TRS-DOS 6.2
- ein CP/M - Computer

An dieser Stelle darf ich noch über bereits vorgenommene Veränderungen am Model 4P berichten. Ich darf dabei nochmals dem Hartmut danken, der diese für mich vorgenommen hat.

- zunächst kamen die internen 40 Spur-Laufwerke ins externe Gehäuse und die externen 80 Spur-Laufwerke in den Computer rein
- damit man die externen Laufwerke auch am 4P anschließen kann, baute mir der Hartmut eine entsprechende Buchse in den 4P ein
- die in der 13. Ausgabe auf Seite 40-41 beschriebenen Floppy-Tricks wurden ebenfalls mit Erfolg ausgeführt. Ich habe jetzt die Möglichkeit, mein Laufwerk 0 mit 40 Spuren oder 80 Spuren laufen zu lassen. Ferner habe ich für jedes Laufwerk einen Schreibschutz-Schalter.

m4/4p ecke

Tips für Model 4/4P

Aufruf des RAM-Tests:

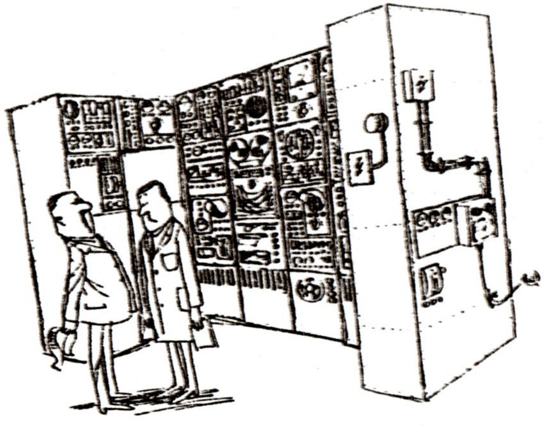
folgende Tasten zusammen drücken   < 6 8 0 >
oder   < . >
und anschließend den RESET-Knopf drücken.

Patch für den KILL-Befehl:

Im TRSDOS wurde der KILL-Befehl entfernt und statt dessen ein REMOVE installiert. Um den KILL-Befehl wieder einzusetzen, kann man folgenden Patch verwenden:

```
PATCH SYS1/SYS.LSIDOS (D02,81=4B:F02,81=00)
```

Der KILL-Befehl hat dieselbe Syntax wie der REMOVE-Befehl.



»Müller, der Computer hat sich über Sie beschwert!«

Aufruf der SuperVisor Calls im Basic

(aus 80Micro, April 1986, Seite 102ff, Hardin Brothers)

TRSDOS 6.xx benutzt für seine Ein- und Ausgabeoperationen die sog. SuperVisor Calls (SVC). Damit lassen sich diese Operationen ziemlich einfach durchführen, wie z.B. Directory lesen, Bildschirminhalt abspeichern etc. Dies ist jedoch nur in Assembler möglich. Am nächsten kommt diesem der SYSTEM-Befehl in Basic. Damit lassen sich jedoch längst nicht alle gewünschten Systeminformationen in ein Programm (-variable) einlesen. Es ist zwar möglich z.B. aus dem Directory Daten über einen File einzulesen und dies auch in ein Programm einzubinden. Will man aber mehrere versch. Systeminfos, so wäre für jede eine spezielle Routine erforderlich. Das in dem o.g. Artikel vorgestellte Programm läßt sich dagegen als GOSUB-Routine in jedes Basicprogramm einbinden. (das Programm befindet sich demnächst auch in der Clubbibliothek - "SVC/BAS" u. "DEMO/SVC")

Nun zur Funktionsweise:
Um von einem Assemblerprogramm einen SVC aufzurufen, wird die SVC-Nummer in das A-Register geladen, weitere erforderl. Register werden ggf. gesetzt. Danach wird ein RST 28H-Befehl ausgeführt. TRSDOS sucht sich nun die aktuelle Adresse des SVC in einer Tabelle und führt anschließend den SVC aus.
Im Prinzip funktioniert so auch das SVC-Programm. Es definiert für jedes Z80 Register eine Variable. über eine Funktion (Zeile 50108) werden zwei dieser 8Bit Register zu einem Registerpaar kombiniert. Um nun einen SVC aufzurufen, müssen nur die richtigen Werte in die Registervariablen geladen werden. Danach wird mit GOSUB 50000 zum SVC-Unterprogramm gesprungen. Die Registerübergabe und der Aufruf des SVC werden durch ein kleines Assemblerprogramm (Z 50100-50138) erledigt. Nach Ausführung des SVC werden die Ergebniswerte wieder in die Registervariablen zurückgeladen, sodaß man darauf vom Basicprogramm aus zugreifen kann. über die Z.FLAG-Variable kann man kontrollieren, ob der SVC-Aufruf erfolgreich war. Sollte für entsprechende Systeminformationen ein Datenfeld notwendig sein, kann man dieses zunächst mit dem DIM-Befehl erstellen und anschließend mit dem ERASE Befehl wieder löschen.

Die einzelnen SVC's und ihre Registerwerte sind im techn. Handbuch für Model 4/4P beschrieben.

Einige der Möglichkeiten, einen SVC von einem Basicprogramm aus aufzurufen, werden in dem DEMO-Programm dargestellt (Scrollschutz für 1-7 Zeilen, Umschaltung Groß-/Kleinschreibung, Abspeichern eines Teils des Bildschirms, womit sich eine Fenstertechnik realisieren läßt, und Einlesen von Teilinformationen aus dem Directory).

Bei der Anwendung sollte man jedoch aufpassen, damit keine falschen Werte in die Registervariablen gelangen. Ansonsten dürfte der RESET-Knopf wohl die letzte Rettung sein.

Klaus Hermann

m4/4p ecke

Das Boot-ROM des 4P

Wie allgemein (?) bekannt, besitzt die tragbare Version des Model 4 kein BASIC- sondern nur ein sog. Boot-ROM. Dieses 4kByte große ROM wird "nur" dazu benutzt, ein Disk Operating System (TRSDOS 6.x, DOSPlus, MultiDOS, CP/M usw.) ins RAM zu laden und zu starten. Will man im Model 3 - Mode arbeiten, muß man ein BASIC-ROM ins RAM laden; auch das erledigt das Boot-ROM!

Diese Operationen sind den meisten Usern bekannt. Weit weniger bekannt sind die weiteren Möglichkeiten, die in diesem 4k-EPROM versteckt sind!

RAM - Test
Hält man beim Einschalten des 4P oder beim RESET drücken die Taste <.> (Dezimalpunkt), startet man ein ins Boot-ROM eingebautes Diagnoseprogramm. Leider sind im "Technical Reference Manual" (aus diesem stammen alle hier abgedruckten Informationen) keine weitere Erläuterung zu diesem Programm zu finden. Da (hoffentlich bleibt das noch lange so) meine 128k-RAM in Ordnung sind, weiß ich nicht, wie eine ev. Fehlermeldung aussehen würde. Sollte jemand mehr zu diesem Teil des Boot-ROM herausbekommen, wäre ich für jede Information sehr dankbar!

Boot - Funktionen
Nachdem der RESET-Schalter betätigt wurde, führt das Boot-ROM einige wichtige Funktionen durch (Bildschirmspeicher löschen, CRT- und Floppy-Controler initialisieren usw.), die uns hier nicht weiter interessieren sollen. Nachdem diese Operationen beendet sind (dauert nur Millisekunden) wird ein sog. Keyboard-Scanner angesprungen. Dieses Programm ermittelt, ob und welche Tasten auf der Tastatur gedrückt wurden. Um dem Benutzer Zeit zur Eingabe seiner Wünsche zu lassen, wird die Durchsicht in einer Schleife für ca. 2-3 Sekunden ständig wiederholt. Erst danach wird das eigentliche Boot-Programm aufgerufen.

Je nachdem, welche Tasten oder Tastenkombinationen während des Keyboard-Scannings gedrückt wurden, werden verschiedene Funktionen ausgeführt. Hier eine ausführliche Zusammenstellung der möglichen Optionen:

Taste(n)	Funktion
F1 oder 1	Hard-Disk - Boot
F2 oder 2	Floppy-Disk - Boot (norm. Funktion)
F3 oder 3	Model 3 - Mode
rechte Shift	RS 232 - Boot

Wie man sieht, kann man dem 4P sogar dann Leben einhauchen, wenn die Drives mal streiken und eine Harddisk nicht vorhanden ist. Man genötigt allerdings ein zweites Model 3/4/4P, von dem man die benötigte Software über die RS 232-Schnitt- stelle bereitgestellt bekommt.

Wird während der Abfrage des Keyboards keine Taste gedrückt, wird zunächst versucht, ein DOS von der Diskette einzulesen. Gelingt dies nicht, wird nach einem Model 3 ROM-Image-File gesucht und dieses (falls vorhanden) geladen. Das Programm schaltet nach einer nicht erfolgreichen DOS-Suche automatisch auf Model 3 - Mode. In diesen Mode stehen dem Benutzer noch drei weitere Optionen offen.

Taste	Funktion
P	wenn das ROM-Image geladen wurde, wird es normalerweise sofort gestartet. Wurde während der Tastaturabfrage <P> eingegeben, wird nach Beendigung des Ladevorgangs eine Meldung angezeigt und auf die Eingabe von ENTER gewartet, bevor es weitergeht!
N	Unterdrückt das Laden des ROM-Image, es sei denn, es ist Bestandteil des DOS (NewDOS für Model 4-Mode) und wird automatisch mitgeladen!
L	Erzwingt das Laden des ROM-Images, auch wenn sich dieses (von einem früheren Bootvorgang) noch im RAM befindet! Bemerkung: Das Boot-ROM erkennt, ob das ROM-Image schon geladen wurde und unterläßt in diesem Fall ein nochmaliges Laden. Die beiden letztgenannten Optionen (N und L) sind vor allem dann brauchbar und notwendig, wenn man am "ROM" (im RAM) Änderungen vorgenommen hat und entweder ein überschreiben verhindern oder erzwingen will!

Sondertasten
Wie schon erwähnt, tritt durch die Tastaturabfrage eine Wartezeit von ca. 2-3 Sekunden auf. Ist einem diese Zeitspanne zu lange, kann man sie durch Drücken von ENTER abbrechen. Will man z.B. TRSDOS 2.3 laden und drückt direkt nach RESET die Taste ENTER, wird das Booten um ca. 2-3 Sekunden beschläunigt!

Eine weitere Sonderstellung nimmt die BREAK-Taste ein. Mit ihr kann man dem DOS (nur im Model 4 - Mode) einen "Wink" geben. Wird während der Keyboadabfrage BREAK gedrückt, wird die RAM-Speicherstelle 405Bh (16475d) auf ungleich 00 gesetzt. Wurde BREAK nicht gedrückt, findet das DOS an dieser Stelle später den Wert 00 wieder! Damit kann man also schon während des Bootvorganges dem DOS eine Mitteilung zukommen lassen und ev. eine besondere Aktion starten.

Beispiel
Nehmen wir an, man hat im Model 3 - Mode ein bißchen am "ROM" herumgezapt und will nun neu booten. Dazu wären folgende Tasteingaben nötig:

F3 oder 3	(muß nicht unbedingt sein, da, wenn kein DOS auf der Disk gefunden wird, sowieso nach dem ROM-Image gesucht wird!)
L	(das ROM-Image wird "zwangsweise" geladen!)
P	(es soll die Meldung ausgegeben und auf ENTER gewartet werden!)
ENTER	(muß nicht sein, kürzt aber die Wartezeit ab!)

Übrigens!
Wenn man Tastenkombinationen wählt, die sich widersprechen (z.B. N und L), dann wird jeweils die zuletzt gegebene Instruktion (hier z.B. L) ausgeführt!

Sonst noch was?
Wenn man sich das Boot-ROM einmal ansieht (siehe Listing), wird man feststellen, daß über 50% des EPROM mit Texten (sprich Fehlermeldungen) in drei Sprachen (engl., deutsch und franz.) gefüllt ist. Die eigentliche Programm-Area kommt einem da ziemlich kurz vor und bringt einen (zumindest mich) auf dumme Gedanken. Diese bestehen darin, die "überflüssigen" Texte (Fehlermeldungen in einer Sprache sind meiner Meinung nach genug) aus dem Boot-ROM zu entfernen und den entstehenden Platz mit nützlichem zu füllen.

m4/4p ecke

Denkbar wären z.B.: ein Monitor-Programm, eine Bildschirm-Hardcopy-Routine (für Programme, die auf JKL wegen abgeschalteter Interrupts nicht reagieren), eine Password-Abfrage nach der Inbetriebnahme usw. usw.. Es sind sicher noch viele Anwendungen und Spielereien, die sich hier unterbringen ließen. Für Vorschläge bin ich immer offen und dankbar!

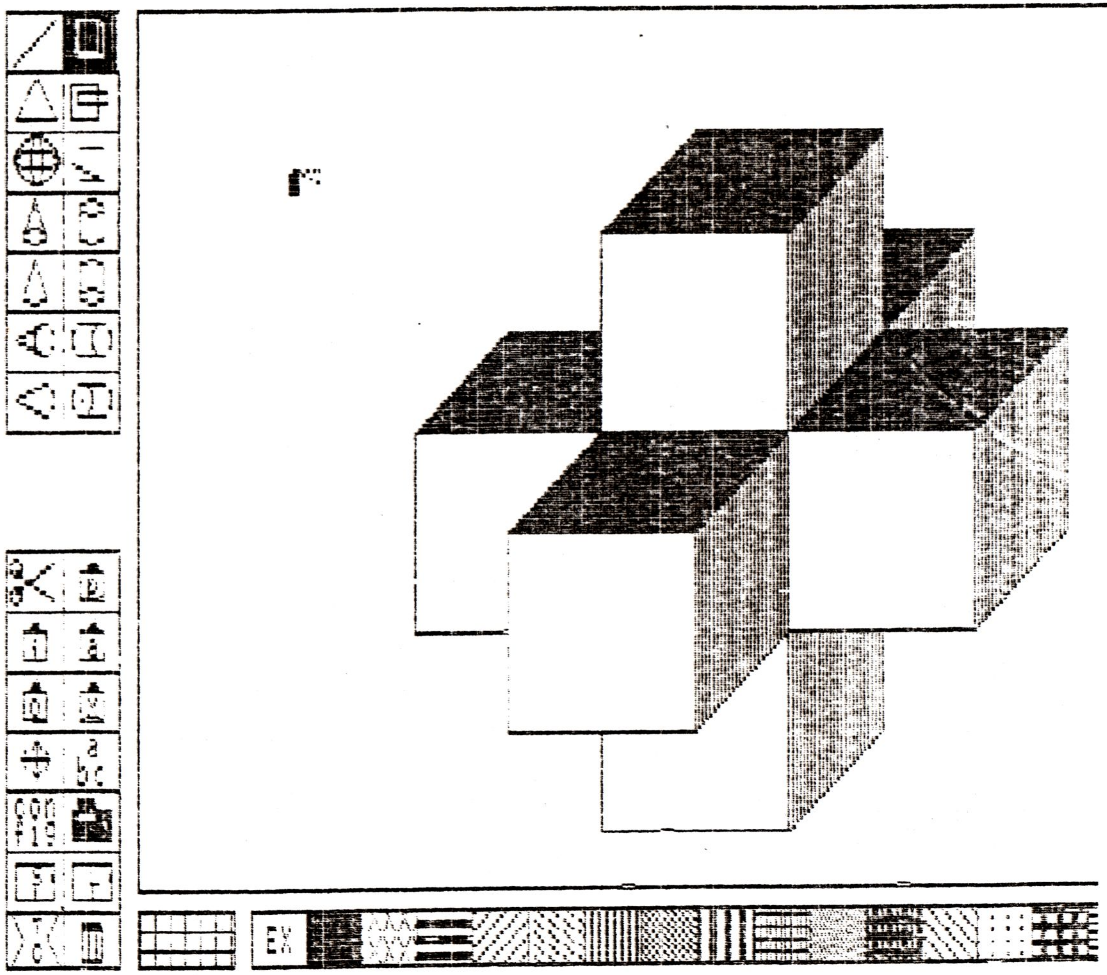
Read Boot-ROM
Das folgende kleine Assemblerprogramm ermöglicht es, das Boot-ROM, welches normalerweise in einer nicht ohne weiteres erreichbaren Speicherbank liegt, ins RAM zu verschieben (Topmem). Von dort kann man es mit dem DOS-Befehl DUMP BOOTROM/CIM F000h FFFFh 0000h 0000h auf Diskette abspeichern.

```
        ORG     5200H         'los gehts!
START   DI                    'keine Unterbrechungen bitte
        LD      A,01h         '01 in den Accu
        OUT     (9CH),A       'Boot-ROM einschalten
        LD      HL,0000h      'Quelladresse in HL
        LD      DE,0F000h     'Zeiladresse in DE
        LD      BC,0FFFh      'Anzahl der zu
                               verschiebenden Bytes
        LDIR                  'Verschieben
        XOR     A             '00 in den Accu
        OUT     (9CH),A       'Boot-ROM ausblenden
        EI                    'jetzt darf wieder
                               interruptet werden!
        JP      402Dh         'zurück zu DOS Ready
        END     START         'das bittere Ende!!!
```

Das wäre erstmal alles zum Boot-ROM. Sollte jemand weitere Informationen haben, möge er sie bitte im Interesse aller Clubmitglieder weitergeben!
Wer sich für das von Tandy vertriebene "Technical Reference Manual" für das Model 4/4P interessiert, kann es sich bei mir (kurzzeitig) ausleihen. Das Buch enthält viele interessante Informationen und Hinweise, ist jedoch für den "Normalanwender" zu umfangreich (nur absolute Freaks werden sich wohl für das Datenblatt des Bildschirmcontrolers interessieren) und zu teuer (ca. 70,- Deutsche Mark!).

Damit erstmal genug für heute. Happy computing wünscht

Hartmut Obermann



M4 Grafik

HEFT 14 Juli 1986

HRG für Model 3/4/4P!

Seit kurzem bin ich stolzer Besitzer einer Grafikkarte (µ-Labs Graphyx Solution) mit einer Auflösung von 640 * 240 Punkten (512 * 192 im Model 3-Modus). Hier ein kurzer Erfahrungsbericht über den Einbau und die ersten Programmierversuche.

Die µ-Labs-Karte ist sehr sauber aufgebaut (sauberes Layout) und macht einen stabilen Eindruck. Letzteres ist besonders wichtig, da die Karte bei mir ja in ein tragbares Gerät (4P) eingebaut wird! Grafyx Solution gibt es übrigens auch für's Model 3 und 4(Tischmodel). Der Einbau der Schaltung ist genauestens beschrieben, sodaß auch ein Hardwarelaie keinerlei Probleme haben dürfte. Lötungen sind keine vorzunehmen; die Karte wird auf einen Pfostenfeldstecker aufgesetzt und eine Leitung per Klemme mit einem IC-Beinchen verbunden. Weiterhin ist noch ein Jumper umzustecken, um dem Gerät das Vorhandensein der Karte zu signalisieren, dann kann der Grafikspaß auch schon beginnen!

Zusätzlich zur Karte bekommt man ein relativ dünnes Handbuch und eine Diskette mit Software für Model 3 und 4. Trotz des geringen Umfanges des Handbuches kommt man eigentlich von Anfang an sehr gut mit der HRG zurecht. Dazu tragen vor allem die über 20 mitgelieferten, teils sehr kurzen Beispielprogramme und Utilitys bei.

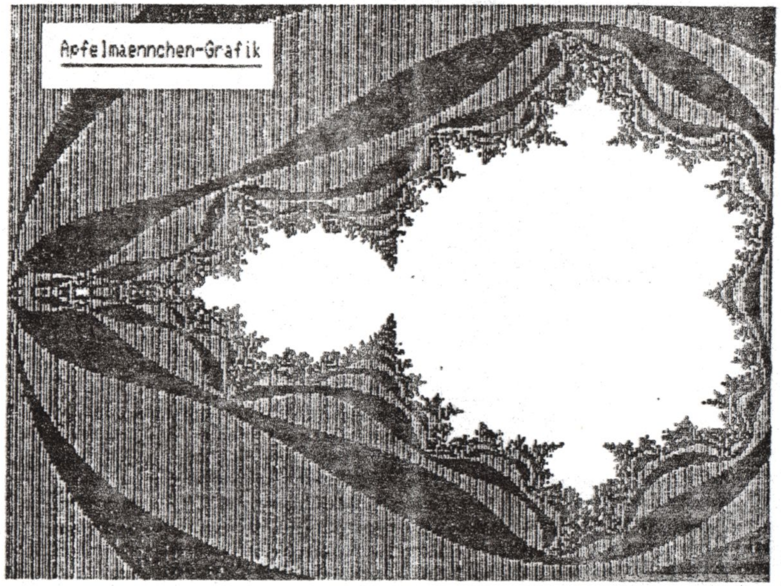
Die mitgelieferte BASIC-Erweiterung ermöglicht eine sehr komfortable Programmierung von Grafik. Ihr Befehlsumfang ist reichlich und umfaßt neben "normalen" Grafikbefehlen wie CLS, PLOT, POINT usw. auch Erweiterungen zum schnellen Zeichnen von Kreisen (CIRCLE) und Rechtecken (BOX), eine Füllroutine mit 256 möglichen Mustern und eine integrierte LOAD/SAVE-Möglichkeit (ca. 3-5 Sekunden Ladezeit) für die Bilder.

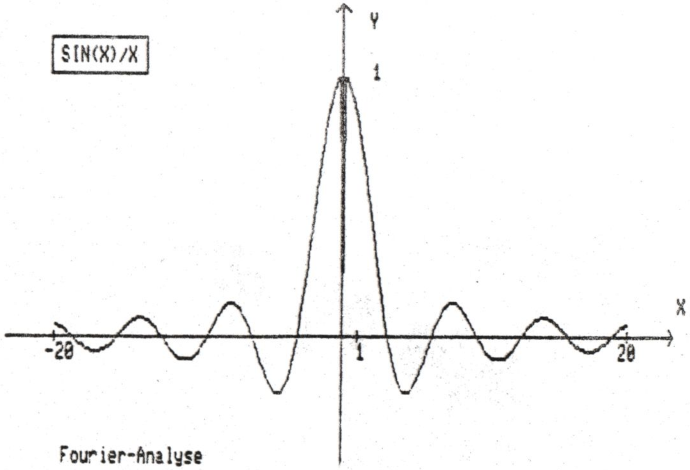
Natürlich will man die Grafikkarte nicht nur unter BASIC sondern auch unter Assembler ansprechen. Dazu liefert µ-Labs neben der genauen Beschreibung der Ansteuerung der Karte über die Ports 128 - 131 eine CLS- und eine POINT-Routine, sowie ein Beispielprogramm in dem diese Routinen benutzt werden. Damit gibt man dem Benutzer weiterhin die Möglichkeit, nach einigen kleinen Änderungen in der CLS/POINT-Routine, die HRG auch von Microsoft-Sprachen wie Fortran 80, Cobol 80 und BASCOM aus zu gebrauchen. Dadurch wird die Grafik auch in CP/M nutzbar.

Zusammen mit einem Bekannten der normalerweise auf einem MS-DOS-Rechner arbeitet, habe ich die Grafik auch für Turbo-PASCAL angepaßt. Dabei wurden wieder die CLS/POINT-Routinen verwendet (als Inline-Code direkt in den Turbo-Sourcecode eingebaut) und weitere Proceduren dazugeschrieben (PLOT). Wer die angepassten Routinen für BASCOM und Turbo-PASCAL haben möchte, kann sich direkt an mich wenden. Sicherlich könnte man auch die Model 1 HRG 1B für Hochsprachen in CP/M nutzbar machen!

Die beiden nebenstehenden Grafiken sind übrigens mit Turbo-Pascal-Programmen entstanden, die von einem IBM-Kompatiblen auf Montezuma CP/M transferiert und an die hochauflösende Grafik angepaßt wurden!

Hartmut Obermann





DOS - (fast) ohne Floppy

128 kB oder mehr sind für die meisten heute verkauften Computer Standard geworden. Helmuts Banker für 256 kB bis 1 MB RAM macht aus den in unserem Club vertretenen Computern Maschinen, die sich in mancher Hinsicht ohne weiteres mit sehr viel teureren Rechnern messen können. Die Datenmengen, die wir zu verarbeiten haben, passen zwar normalerweise in die serienmäßigen 48 kB. Aber unser vergleichsweise leistungsfähiges DOS ist dafür bereits zu umfangreich. Deshalb wird es je nach Bedarf in kleinen Portionen in den Speicher geladen. Die Folge: Die Floppy steht kaum still und wird entsprechend belastet.

Hier soll das Programm RAMSYS/CMD vorgestellt werden, das zunächst alle Systemdateien des NEWDOS-80/G-DOS/H-DOS in den Speicher lädt, die nach dem Booten noch gebraucht werden. Danach residiert das DOS in zweien der verfügbaren Banks und braucht nicht von Fall zu Fall vom Systemlaufwerk nachgeladen zu werden. Eine Ausnahme bilden lediglich INHALT/SYS (DIR/SYS), SYS4/SYS (DOS-Fehlermeldungen) und SYS8/SYS (DIR-Befehl). INHALT/SYS muß auf der Diskette verwaltet werden, um beim erneuten Booten keine bösen Überraschungen zu erleben. Wie sich bei Tests herausstellte, treten beim Aufruf eines RAM-residenten SYS4/SYS und SYS8/SYS Fehler auf, weil diese offenbar die RST-28h-Routine modifizieren. Sie werden deshalb wie gehabt von der Diskette abgerufen.

Das Programm beginnt mit dem FCB, der zum Laden der SYS-Files gebraucht wird. Es schließt sich eine Tabelle an, in der die diskrelativen (physikalischen) Adressen der Systemdateien verzeichnet sind und deren Länge in Sektoren. Mit diesen Informationen kann das Programm die Systemmoduln laden. Das Byte, das normalerweise die Länge angibt, enthält an zwei Stellen eine andere Information: Die Speicherblocks, die der Banker zugänglich macht, sind je 32 kB lang. Das ist zu wenig für das gesamte DOS. Nach dem Laden von SYS12/SYS wird deshalb von Bank 1 auf Bank 2 umgeschaltet, um dort den Rest unterzubringen. Das Signal zum Umschalten gibt das Byte 255 in Zeile 48. Am Ende der Tabelle findet sich eine 0, die dieses Ende kenntlich macht.

Das Programm beginnt beim Label start (Z. 85). Dort wird der Banker durch einen beliebigen Output auf den Port EDh so konfiguriert, daß er die Umschaltungen im Adreßbereich ab 8000h vornimmt. Nicht alle Bernhardt-Banker haben dieses Feature. Die Umschaltung muß, wie auch immer, A15 gegen *A15 austauschen. Das geht auch zu Fuß mit einem gewöhnlichen Schalter. Ebenso ist es möglich, die normale Speicherkonfiguration beizubehalten. Dann liegt der Common-Bereich in der oberen Speicherhälfte, was für den Betrieb mit CP/M besser ist. Nun kann z. B. ein Programm im Himem die Arbeit übernehmen, das sich vom hier vorgestellten nur um wenige Bytes unterscheiden müßte.

Anschließend an diese Operation wird mit dem Output 1 auf den Port ECh in diesem Adreßraum die Bank 1 selektiert, wo die erste Hälfte des DOS geladen werden soll. Das passiert in der Schleife tabloop. Der FCB wird von Sektor zu Sektor jeweils aktualisiert. Wird das Byte 255 angetroffen, schaltet der Banker um auf Bank 2. Dort wird die zweite Hälfte wieder ab 8000h geladen. Wenn das 0-Byte erkannt wird, ist tabloop fertig abgearbeitet, und es geht bei tabdone (Z. 127) weiter.

Hier wird das Programm zum Interpretieren eines DOS-Requestcodes, Label loadsys (Z. 150), an seine endgültige Arbeitsadresse geladen. Sie lautet hier 3900h. Dort ist nämlich das "sichere Plätzchen" von Helmut Bernhardt (beschrieben in cft 5/85). Es handelt sich um freies RAM in dem Adreßbereich, wo ab Werk die Tastatur überflüssigerweise Speicherplatz verschwendet. Wer das "sichere Plätzchen" nicht eingebaut hat, kann einen beliebigen anderen Speicherbereich zwischen 5000h und 7FFFh wählen (oder 8000-FFFFh, wenn der Common-Bereich oben liegt, s. o.). Helmut hat bisher zwei Versionen seiner RAM-Platine in 3900-3BFFh veröffentlicht (eine davon im Info 10/85 des Bremerhavener Clubs).

Ein anderer Ausweg ist der EG 64 MBA. Er erlaubt es z. B., die RST-28h-Umleitung "neben" den Bildschirm ab 3C00h oder in den Adreßbereich des Sonder-ROMs ab 3000h zu legen. Dazu werden auf den Port DFh die Codes 0Ah und 0Bh ausgegeben, um den Speicherbereich 3000-37FFh auf RAM zu schalten. Dann lautet der offset (Z. 150) $-3000h.

Leider führten Versuche zu Systemabstürzen, bei denen das Arbeitsprogramm im Bereich der ursprünglichen RST-28h-Routine lief. Dies geschah jedoch nur bei einigen wenigen DOS-Anforderungen. Bei entsprechenden Checks und Ausschlüssen bestimmter Requestcodes (s. u.) wäre das aber machbar. Die meisten und wichtigsten Systemfiles konnten sogar ohne solche Vorsichtsmaßnahmen geladen und benutzt werden, z. B. auch Debug.

Um Speicherplatz zu sparen, wird der größte Teil der Arbeit zum Finden der zuständigen SYS-Datei in den beiden reservierten Banks erledigt, und zwar von dem Programmsegment findsys (Z. 178). Dieser Abschnitt kommt in den oberen, frei gebliebenen Teil der beiden Nachbarbanks 1 und 2.

Zuletzt wird die ursprüngliche Speicherkonfiguration wiederhergestellt. Bank 0 wird zugeschaltet (0 auf Port ECh) und die RST-28h-Routine gepatcht: Bis zur Adresse 4BDFh hat sie die nötigen Vorarbeiten geleistet, ohne die auch RAMSYS nicht auskommt. Von hier ab geht es aber darum, das Directory nach dem zuständigen Systemfile abzusuchen, was ohne rotierende Floppy nicht zu machen ist. Und genau das soll mit RAMSYS aber vermieden werden. Deshalb wird an der Stelle 4BE0h ein Sprung nach loadsys gepatcht (ab Z. 142). Damit sind alle Initialisierungen erledigt, und mit RET kehrt das Programm ins DOS zurück.

In der RST-28h-Umleitung loadsys wird geprüft, ob der Requestcode eines der kritischen Files (s. o.) betrifft. In diesem Falle geht es in der alten RST-28h-Routine weiter, die dann in Gottes Namen die Drives ankurbelt. Erlaubte DOS-Moduln werden aber aus dem RAM geholt. Da sie auf zwei Banks verteilt sind, muß die richtige zunächst herausgefunden werden. Sie geht aus dem Requestcode hervor. Wenn er 0Fh oder mehr beträgt, liegt die Datei in Bank 2, sonst in Bank 1. Dabei bezieht sich die Angabe des Requestcodes übrigens auf seinen um die Auswahl-Bits 5-7 verkürzten Rumpf. Die entsprechende Bank wird angewählt. Da unter BASIC und bei vielen Anwenderprogrammen der Stack im oberen Adreßbereich ab 8000h liegt, lädt die Routine den Stackpointer mit dem Himem der Nachbarbank. Jetzt wird findsys aufgerufen, um das zuständige Systemfile zu finden:

Da nicht alle SYS-Files gleich lang sind (SYS6/SYS umfaßt 35 Sektoren) und da in beiden Banks im Akku für die erste Datei der Requestcode 0 erwartet wird, muß anfangs ein wenig gerechnet werden (Zz. 180 ff.). Nun hat der Akku seinen korrekten Wert, und die SYS-Dateien können abgezählt werden, bis die richtige gefunden ist (ab sectok). In der Folge werden die einzelnen Bytes des Record-Headers nach ihrer Bedeutung interpretiert und ausgewertet. Die eigentlichen Programm- bzw. Datenbytes werden in den Speicher geladen, wo sie auch ab Floppy hingekommen wären. Dabei sorgt eine Sicherung dafür, daß Kommentar-Records sozusagen im Niemandsland, nämlich in einem freien Bereich der Nachbarbank verschwinden.

Wenn die Systemdatei vollständig übertragen ist, kehrt findsys zum CALLer loadsys zurück. Der restauriert als erstes wieder den Stack und schaltet die Bank 0 ein. Die vorsichtshalber abgeschalteten Interrupts werden wieder zugelassen. An der Stelle 4C19h, wohin loadsys zuletzt springt, wird hauptsächlich das Systemprogramm angesprungen. Seine Startadresse wurde in findsys in den Adreß-Operanden des CALL-Befehls bei 4C1Dh geladen. Im weiteren Verlauf macht das DOS genau dasselbe, als hätte es die Systemdatei von der Diskette geholt.

Diese DOS-Vergewaltigung ist relativ umfangreich, aber dennoch um ein Vielfaches schneller als die ursprüngliche RST-28h-Prozedur. Das Auffinden der Systemdatei im Inhaltsverzeichnis und das Laden von der Floppy erübrigen sich nämlich. Zum Test kann der Leser beispielsweise JKL eingeben. Der Drucker beginnt sofort mit der Arbeit, die Laufwerks bleiben stehen.

Zuletzt noch ein Hinweis: Die DOS-Moduln liegen in den verschiedenen DOS-Versionen an unterschiedlichen physikalischen Disk-Adressen. Mit SUPERZAP oder einem ähnlichen Programm muß daher zunächst überprüft werden, ob vielleicht Änderungen in der Tabelle tab (Zz. 24 ff.) erforderlich sind. Die Programme werden (bei SUPERZAP) mit DFS aufgerufen. In der Anzeige ist danach ablesbar, um welchen diskrelativen Sektor es sich handelt (drs).

Es gibt auch DOSsas, bei denen mehr Systemdateien als nur SYS6/SYS über 5 Sektoren lang sind. Das gilt z. B. für SYS25/SYS von G-DOS 2.1b. SUPERZAP und DDE helfen dabei, es herauszufinden. In diesem Falle muß ähnlich wie wegen SYS6/SYS in Z. 183 der Requestcode bereinigt werden. Da es sich dabei aber um simple binäre Arithmetik handelt, sollte dies kaum abschrecken.

U. U. kann es auch erforderlich werden, den Bankwechsel anderswo als bei SYS13/SYS vorzunehmen. Man sollte vorsichtshalber alle relevanten Systemsektoren addieren, die Anzahl halbieren und die Trennung entsprechend vornehmen. Da genügend Platz vorhanden ist, kommt es auf ein Ungleichgewicht von ein paar Sektoren aber nicht an. Wichtig ist dabei nur, daß für das Unterprogramm findsys und eventuelle Kommentarrecords noch 315 Bytes übrig sind. Sie an das oberste Ende der Banks 1 und 2 zu legen, ist bei äußerster Platznot auch keine Hürde: Das Label offset2 kann mit maximal FEC4h statt F000h definiert werden. Wichtig ist lediglich, daß die erste Systemdatei der neuen Bank fünf Sektoren lang ist. RAMSYS unterstellt nämlich, daß bei dem angeblichen Sektorzähler 255 in Wirklichkeit 5 gemeint ist.

Hier ein Hinweis für die dann notwendigen Änderungen: In der Zeile 161 wird der Akku mit 0Fh verglichen. Das ist der Rumpf-Requestcode für SYS13/SYS. Er erhöht bzw. erniedrigt sich mit der laufenden Nummer der Systemdatei. Er ist immer um zwei höher als deren Nummer. In den Zz. 184 und 186 hat er eine Subtraktion um 3 hinter sich. Hier beträgt die Differenz nunmehr minus eins. Ist es also SYS12/SYS, dann muß hier mit 0Bh+1Eh verglichen werden. 0Bh ist dezimal 11 (12-1).

Das liest sich furchtbar trocken und ist es wohl auch. Solche Modifikationen sind aber nur bei DOS-Exoten wie DOSPLUS, Genie-Text u. dergl. zu erwarten. Es sei dem geneigten Leser empfohlen, ein rechtschaffenes System wie NEW-, G- oder H-DOS zu fahren. Alles andere verdirbt eh' den Charakter.

Arnulf Sopp

Bitte beachtet dazu auch die
Seiten 91 - 99
dieses INFO's


```
                00001 ;**********************************************************
                00002 ;*                                                        *
                00003 ;*                      R A M S Y S                       *
                00004 ;*                                                        *
                00005 ;*      Utility zum RAM-residenten Betrieb von            *
                00006 ;*      NEWDOS-80, G- und H-DOS mit dem 256-kB-           *
                00007 ;*           Banker von Helmut Bernhardt                  *
                00008 ;*                                                        *
                00009 ;*              (C) 1986 by Arnulf Sopp                   *
                00010 ;*                                                        *
                00011 ;**********************************************************
                00012
5300            00013          ORG     5300h
                00014
                00015 ;FCB für diskrelative Sektor-I/O
5300  82        00016 fcb      DB      82h,28h,00h                  ;diskrelativ usw.
5303  0052      00017          DW      5200h                        ;Sektorpuffer
5305  00        00018          DB      00h,00h,00h,00h,00h          ;hier belanglos
530A  0000      00019 sector   DW      0000h                        ;Sektornummer
530C  FFFF      00020          DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh ;s. o.
5316  FFFF      00021          DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh ;s. o.
                00022
                00023 ;Tabelle der File-Längen und diskrelativen Sektoradressen
5320  05        00024 tab      DB      5                  ;Länge in Sektoren von
5321  9605      00025          DW      0596h              ;SYS1/SYS, Startsektor
5323  05        00026          DB      5                  ;dto. SYS2
5324  9B05      00027          DW      059bh
5326  05        00028          DB      5                  ;SYS3
5327  BE05      00029          DW      05beh
5329  05        00030          DB      5                  ;SYS4
532A  C305      00031          DW      05c3h
532C  05        00032          DB      5                  ;SYS5
532D  D205      00033          DW      05d2h
532F  23        00034          DB      35                 ;SYS6
5330  D705      00035          DW      05d7h
5332  05        00036          DB      5                  ;SYS7
5333  7805      00037          DW      0578h
5335  05        00038          DB      5                  ;SYS8
5336  C805      00039          DW      05c8h
5338  05        00040          DB      5                  ;SYS9
5339  CD05      00041          DW      05cdh
533B  05        00042          DB      5                  ;SYS10
533C  8C05      00043          DW      058ch
533E  05        00044          DB      5                  ;SYS11
533F  6905      00045          DW      0569h
5341  05        00046          DB      5                  ;SYS12
5342  FA05      00047          DW      05fah
5344  FF        00048          DB      255                ;SYS13, Bank umschalten
5345  8705      00049          DW      0587h
5347  05        00050          DB      5                  ;SYS14
5348  7305      00051          DW      0573h
534A  05        00052          DB      5                  ;SYS15
534B  6E05      00053          DW      056eh
534D  05        00054          DB      5                  ;SYS16
534E  6405      00055          DW      0564h
5350  05        00056          DB      5                  ;SYS17
5351  5F05      00057          DW      055fh
5353  05        00058          DB      5                  ;SYS18
5354  8205      00059          DW      0582h
5356  05        00060          DB      5                  ;SYS19
5357  7D05      00061          DW      057dh
5359  05        00062          DB      5                  ;SYS20
535A  9105      00063          DW      0591h
535C  05        00064          DB      5                  ;SYS21
```

```
535D  FF05      00065          DW      05ffh
535F  05        00066          DB      5                    ;SYS22
5360  0E06      00067          DW      060eh
5362  05        00068          DB      5                    ;SYS23
5363  5A05      00069          DW      055ah
5365  05        00070          DB      5                    ;SYS24
5366  1306      00071          DW      0613h
5368  05        00072          DB      5                    ;SYS25
5369  0406      00073          DW      0604h
536B  05        00074          DB      5                    ;SYS26
536C  4605      00075          DW      0546h
536E  05        00076          DB      5                    ;SYS27
536F  4B05      00077          DW      054bh
5371  05        00078          DB      5                    ;SYS28
5372  5005      00079          DW      0550h
5374  05        00080          DB      5                    ;SYS29
5375  5505      00081          DW      0555h
5377  00        00082          DB      0                    ;Flag für Tabellenende
                00083
                00084 ;SYS-Dateien in die Puffer laden
5378  D3ED      00085 start    OUT     (0edh),A             ;Common-Bereich unten
537A  3E01      00086          LD      A,1                  ;Bank 1 zuschalten
537C  D3EC      00087          OUT     (0ech),A
537E  212053    00088          LD      HL,tab               ;Anfang der Sektortabelle
5381  F3        00089          DI                           ;vorsichtshalber
5382  46        00090 tabloop  LD      B,(HL)               ;Länge des SYS-Files
5383  04        00091          INC     B                    ;Ende 1. Hälfte?
5384  200B      00092          JR      NZ,noend             ;falls noch nicht
5386  3E02      00093          LD      A,2                  ;sonst Bank 2 zuschalten
5388  D3EC      00094          OUT     (0ech),A
538A  3E80      00095          LD      A,80h                ;MSB Sektorpuffer
538C  32AD53    00096          LD      (filbuf),A           ;neu festlegen
538F  0606      00097          LD      B,6                  ;Sektorzähler + 1
5391  05        00098 noend    DEC     B                    ;Tabelle ganz zuende?
5392  2829      00099          JR      Z,tabdone            ;falls ja
5394  23        00100          INC     HL                   ;LSB der Sektornummer
5395  5E        00101          LD      E,(HL)               ;DE <- Sektornummer. LSB
5396  23        00102          INC     HL                   ;auf MSB stellen
5397  56        00103          LD      D,(HL)               ;D <- MSB
5398  ED530A53  00104          LD      (sector),DE          ;Sektornummer in den FCB
539C  23        00105          INC     HL                   ;auf nächstes Längenbyte
539D  E5        00106          PUSH    HL                   ;retten
539E  110053    00107          LD      DE,fcb               ;zum Lesen der Sektoren
53A1  210052    00108          LD      HL,5200h             ;Sektorpuffer
53A4  CD3644    00109 readlop  CALL    4436h                ;Sektor einlesen
53A7  D9        00110          EXX                          ;Register retten
53A8  210052    00111          LD      HL,5200h             ;Sektorpuffer
53AB  110080    00112          LD      DE,8000h             ;Filepuffer
53AD            00113 filbuf   EQU     $-1                  ;MSB veränderlich
53AE  010001    00114          LD      BC,0100h             ;Sektorlänge
53B1  EDB0      00115          LDIR                         ;Sektor übertragen
53B3  21AD53    00116          LD      HL,filbuf            ;MSB der File-Pufferadr.
53B6  34        00117          INC     (HL)                 ;256 Bytes weiterstellen
53B7  D9        00118          EXX                          ;Register restaurieren
53B8  10EA      00119          DJNZ    readlop              ;bis B Sekt. eingelesen
53BA  E1        00120          POP     HL                   ;Tabellenzeiger
53BB  18C5      00121          JR      tabloop              ;nächstes SYS-File
                00122
53BD  21EF53    00123 tabdone  LD      HL,loadsys           ;Anf. der RST-28h-Routine
53C0  110039    00124          LD      DE,loadsys-offset1        ;dorthin versch.
53C3  013100    00125          LD      BC,findsys-loadsys        ;Länge der Rout.
53C6  EDB0      00126          LDIR                              ;übertragen
53C8  212054    00127          LD      HL,findsys           ;Anf. der RST-28h-Umleit.
53CB  1100F0    00128          LD      DE,findsys+offset2        ;dorthin versch.
```



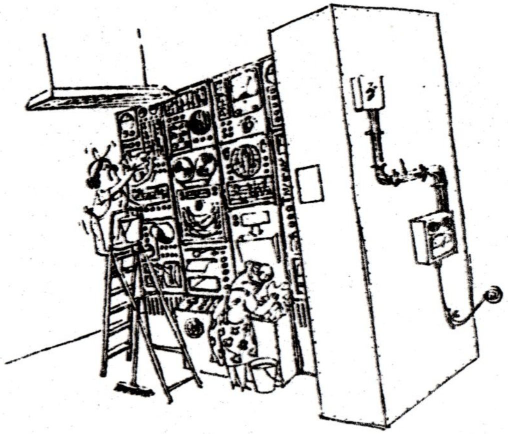
```
53CE  013B00    00129          LD      BC,endprog-findsys        ;Länge der Uml.
53D1  E5        00130          PUSH    HL                   ;alles retten
53D2  D5        00131          PUSH    DE
53D3  C5        00132          PUSH    BC
53D4  EDB0      00133          LDIR                         ;Umleitung übertragen
53D6  3E01      00134          LD      A,1                  ;Bank 1 selektieren
53D8  D3EC      00135          OUT     (0ech),A             ;(seit tabloop Bank 2)
53DA  C1        00136          POP     BC                   ;Register restaurieren
53DB  D1        00137          POP     DE
53DC  E1        00138          POP     HL
53DD  EDB0      00139          LDIR                         ;Uml. auch in Bank 1
53DF  AF        00140          XOR     A                    ;Arbeitsbank 0
53E0  D3EC      00141          OUT     (0ech),A             ;zuschalten
53E2  3EC3      00142          LD      A,0c3h               ;JP-Opcode
53E4  32E04B    00143          LD      (4be0h),A            ;dort laden
53E7  210039    00144          LD      HL,loadsys-offset1        ;Anf. RST 28h
53EA  22E14B    00145          LD      (4be1h),HL           ;umpatchen
53ED  FB        00146          EI                           ;INTs wieder zulassen
53EE  C9        00147          RET                          ;fertig, zurück ins DOS
                00148
                00149 ;Programm zum Laden und Anspringen der RAM-SYS-Files
1AEF            00150 offset1  EQU     $-3900h              ;zum Verladen
53EF  77        00151 loadsys  LD      (HL),A               ;SYS-File vermerken
53F0  FE0B      00152          CP      0bh                  ;SYS9/SYS oder höher?
53F2  300D      00153          JR      NC,ramsys            ;falls ja
53F4  E607      00154          AND     07h                  ;signif. Teil d. Rec.-Cd.
53F6  FE03      00155          CP      03h                  ;ab SYS1, ohne SYS8
53F8  DAE34B    00156          JP      C,4be3h              ;andere normal bearbeiten
53FB  FE06      00157          CP      06h                  ;SYS4/SYS?
53FD  CAE34B    00158          JP      Z,4be3h              ;normal bearbeiten
5400  7E        00159          LD      A,(HL)               ;Requestcode restaurieren
5401  57        00160 ramsys   LD      D,A                  ;und retten
5402  FE0F      00161          CP      0fh                  ;SYS13 oder höher?
5404  3E01      00162          LD      A,1                  ;Voreinstellung Bank 1
5406  3801      00163          JR      C,bankok             ;falls SYS1 - SYS12
5408  3C        00164          INC     A                    ;Bank 2 selektieren
5409  D3EC      00165 bankok   OUT     (0ech),A             ;zuständ. B. zuschalten
540B  F3        00166          DI                           ;vorsichtshalber
540C  ED732839  00167          LD      (spbuf),SP           ;Stackpointer retten
5410  310000    00168          LD      SP,0000h             ;Stack im Himem
5413  CD00F0    00169          CALL    findsys+offset2 ;SYS-File aus RAM laden
5416  310000    00170          LD      SP,0000h             ;Dummy-Operand
3928            00171 spbuf    EQU     $-2-offset1          ;hier Pointer gepuffert
5419  AF        00172          XOR     A                    ;A <- 0 für Bank 0
541A  D3EC      00173          OUT     (0ech),A             ;Bank 0 zuschalten
541C  FB        00174          EI                           ;INTs wieder zulassen
541D  C3194C    00175          JP      4c19h                ;SYS-File starten u. zur.
                00176
                00177 ;Programm zum Herunterladen der SYS-Files
9BE0            00178 offset2  EQU     0f000h-$             ;zum Verladen
5420  7A        00179 findsys  LD      A,D                  ;Requestcode
5421  D603      00180          SUB     03h                  ;an SYS-File# angleichen
5423  FE05      00181          CP      05h                  ;SYS6/SYS oder höher?
5425  3809      00182          JR      C,sectok             ;falls darunter
5427  C61E      00183          ADD     A,1eh                ;sonst 30 Sektoren mehr
5429  FE2A      00184          CP      0ch+1eh              ;SYS13/SYS oder höher?
542B  3802      00185          JR      C,sectok             ;falls darunter
542D  D62A      00186          SUB     0ch+1eh              ;an SYS13 ff. angleichen
542F  210005    00187 sectok   LD      HL,0500h             ;HL = 1280 Bytes = 5 Skt.
5432  CD944C    00188          CALL    4c94h                ;AHL <- HL * A
5435  110080    00189          LD      DE,8000h             ;Pufferanfang
5438  19        00190          ADD     HL,DE                ;HL <- Adr. d. SYS-Files
5439  5E        00191 movelop  LD      E,(HL)               ;1. Byte des Rec.-headers
543A  23        00192          INC     HL                   ;auf nächstes Byte
```

```
543B 0600              LD    B,00h              ;Bytezähler 00xx
543D 7E                LD    A,(HL)             ;Längenbyte
543E FE03              CP    03h                ;256, 257 oder 258?
5440 CB10              RL    B                  ;Zähler-MSB anpassen
5442 4F                LD    C,A                ;BC <- Zählbyte
5443 7B                LD    A,E                ;Typbyte
5444 23                INC   HL                 ;auf Adreß-LSB
5445 5E                LD    E,(HL)             ;DE <- Zieladresse
5446 23                INC   HL
5447 56                LD    D,(HL)
5448 ED531E4C          LD    (4c1eh),DE         ;Startadresse
544C 23                INC   HL                 ;1. Programmbyte
544D 3D                DEC   A                  ;Typbyte 01?
544E 2805              JR    Z,cmdrec           ;falls ja
5450 3D                DEC   A                  ;Typbyte 02?
5451 C8                RET   Z                  ;Ende, falls ja
5452 113BF0            LD    DE,endprog+offset2      ;Komment. dorthin
5455 0B       cmdrec   DEC   BC                 ;um 2 vermindern, weil
5456 0B                DEC   BC                 ;Adreßbytes mitgezählt
5457 EDB0              LDIR                     ;Record übertragen
5459 18DE              JR    movelop            ;bis SYS-File übertragen
545B          endprog  EQU   $

5378                   END   start

00000 Fehler

bankok  5409   cmdrec  5455   endprog 545B   fcb     5300   filbuf  53AD
findsys 5420   loadsys 53EF   movelop 5439   noend   5391   offset1 1AEF
offset2 9BE0   ramsys  5401   readlop 53A4   sectok  542F   sector  530A
spbuf   3928   start   5378   tab     5320   tabdone 53BD   tabloop 5382
```

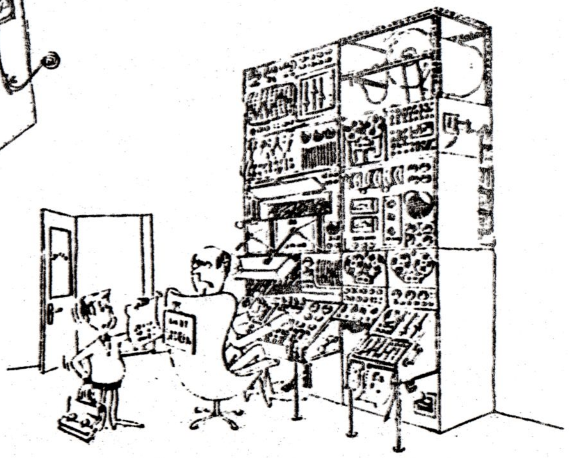
Eine Elektrogeräte-Fabrik in Los Angeles produziert batteriegetriebene Spaghetti-Gabeln, die Nudeln automatisch aufrollen. Eine Ersatzbatterie wird serienmäßig mitgeliefert, damit stets ein ungetrübter und ungestörter Eßgenuß gesichert ist.

Über einen Schwachstromtechniker, der sein Hinterteil mit Stanniol eingeschalt, durch die Verpackung dann Strom geleitet und dabei schwere Verbrennungen erlitten hatte, urteilt die Wiener »Klinik Hoff« in einem psychiatrischen Gutachten: »Dieser Mann ist völlig normal. Seine Handlung läßt lediglich die besonders innige Beziehung zu seiner beruflichen Tätigkeit erkennen.«

Bedingt durch vollautomatischen Rechnungsverkehr erhielten in einer bundesdeutschen Großstadt wohnende Bürger, die wochenlang verreist waren und deshalb keinen Strom verbrauchten, von den Stadtwerken Rechnungen in Höhe von 0,00 Mark. Wird der Betrag nicht fristgerecht eingezahlt, bitten die Stadtwerke »dringend, den Rechnungsbetrag von 0,00 Mark und 0,70 Mark Mahnkosten« zu überweisen. Im Weigerungsfalle – so wird den Kunden angedroht – würden sie »ohne weitere Mahnung« von der Strombelieferung ausgeschlossen.



»Wenn der schon alles kann, warum kann er sich dann nicht selbst abstauben?«



»Ist mir egal, ob dein Opa hier der Chef ist – mach' deine Schulaufgaben allein!«

Wie der Natuerliche Logarithmus die Rechenzeit verkuerzen kann

===

MITHILFE DES NATUERLICHEN LOGARITHMUS KOENNEN KOMPLIZIERTE MATHEMATISCHE AUSDRUECKE, DIE AUS MULTIPLIKATION, DIVISION, EXPONENTIATION UND WURZELAUSDRUECKEN BESTEHEN, IN EINE AUCH FUER 8-BIT CPU's HANDHABBARE FORM GEBRACHT WERDEN.

Peter Guagliano, COMTECH GOVERNMENT SYSTEMS

Um auch komplizierte arithmetische Probleme mit 8-bit CPU's in Assembler loesen zu koennen, sei sich der einfachen Handhabung der Logarithmen erinnert.
Solch ein Gedanke kann - richtig angewaendet - den Einsatz eines Arithmetik-Prozessors ueberfluessig machen. Sinn und Zweck dieser Ueberlegungen ist es, die Benutzung von langsamen High-Level-Language Bibliotheksroutinen ueberflussig zu machen, die nicht selten ein festes Parameteruebergabeformat beduerfen.
Mithilfe des Natuerlichen Logarithmus ist es nun moeglich, zu multiplizieren, dividieren (alles trotz fehlender Hardware-Befehle) oder sogar Wurzelrechnung und Exponentiation. Eine kurze Uebersicht ueber die moeglichen logarithmenverknuepfungen zeigt die nachfolgende Tabelle :

- o log(ab) = log(a) + log(b)
- o log(a/b) = log(a) - log(b)
- o log(sqrt(a)) = 1/2 * log(a)
- o log(a*a) = 2 * log(a)
- o log(a) fuer a <= 0 ist nicht definiert !

Um obige Gleichungen in Mikroprozessor-Applikationen nutzen zu koennen, wird ein Algorithmus zur Umwandlung von Binaerzahlen in den binaeren Logarithmus benoetigt.
Da negative Zahlen und Null nicht definiert sind, bleiben diese unberuecksichtigt. Damit der verwaendete Algorithmus einfach gehalten werden kann, erfolgt eine Eingrenzung des Wertebereiches auf Zahlen >= 1. Sollen dennoch einmal positive ganze Zahlen mit gebrochenen Zahlen multipliziert oder dividiert werden, ist die inverse Operation zu benutzen. Ein Beispiel :

y = 2 / 0.1

stattdessen

y = 2 * 10

NATUERLICHER LOGARITHMUS EINFACH INTERPOLIERT

Bild 1 zeigt den Funktionsgrafen des nat. Log., in Bild 2 ist ein vergroesserter Ausschnitt wiedergegeben. In diesem Bereich ist die Funktion nahezu linear, ein Umstand der fuer einige Vereinfachungen im Umwandlungs-Algorithmus genutzt werden kann.
Nun jedoch zu den Einzelheiten des benutzten Algorithmus.

Es sei X eine binaere Zahl. Der Exponent zur Basis 2 sei N, gegeben durch die Position der hoechstwertigsten 1 in der Binaerzahl. Zur Erinnerung : Die Nullen und Einsen repraesentieren die Koeffizienten a der Stellenwertigkeiten.

$$a_0 2^0 + a_1 2^1 + \ldots + a_k 2^k$$

Demzufolge ist a der Koeffizient von 2^N . Im einzelnen :

00010110, N = 4
01011001, N = 6
11010011, N = 7

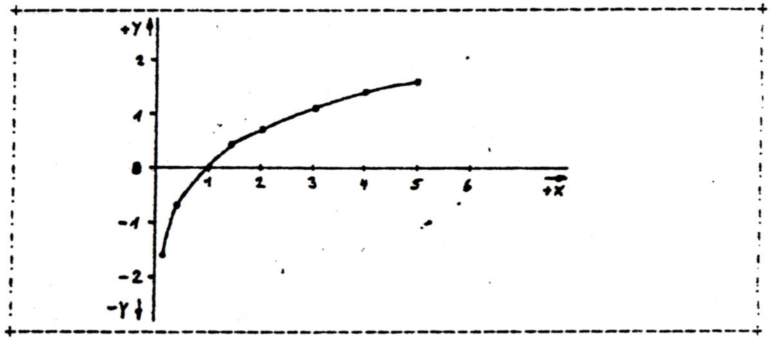


Bild 1: Lineare Approximation der Funktion $y = \log_2(x)$ im Intervall 1<X<2. Die durchgezogene Line stellt dabei die beste Annaeherung an den tatsaechlichen Funktionsverlauf dar.

Durch anwenden des Algorithmus kann die Charackteristik N des binaeren Logarithmus einer Zahl (ganzzahlig) sowie der gebrochene Anteil (=Mantisse) bestimmt werden.
Das Folgende zeigt, wie N und Mantisse bestimmt werden koennen. Hierbei sei X eine positive ganze Zahl, die folgendermassen dargestellt werden kann :

$$X = 2^N + (X - 2^N)$$

Umformen ergibt

$$x = 2^N + 2^N \left[\frac{X - 2^N}{2^N}\right]$$

Als Produkt

$$X = 2^N \left[1 + \frac{X - 2^N}{2^N}\right]$$

Logarithmieren obiger Gleichung und umschreiben in die Form Log(ab) = Log(a) + Log(b) ergibt

$$\log(X) = \log(2^N) + \log\left[1 + \frac{X - 2^N}{2^N}\right]$$

und weiter

$$\log(X) = N + \log\left[1 + \frac{x - 2^N}{2^N}\right] \qquad (2)$$

Hierbei nimmt der Term $(X - 2^N)/2^N$ - er ergibt den Divisionsrest, nachdem das MSB-Bit per Shift entfernt wurde - die Werte zwischen Null und Eins an.

Die gestrichelte Linie in Bild 2 ergibt fuer Gleichung 1 folgende lineare Approximation fuer Werte zwischen 1 und

$$\log(X) = N + \frac{X - 2^N}{2^N}$$

Diese Interpolations-Gl. ergibt jedoch einen max. Fehler von 9%.Hinzu kommt ausserdem,dass der Fehler einseitig und die ermittelten Werte stets zu klein sind.
Der mittlere Fehler belaeuft sich somit auf 4%, summiert sich aber mit jeder durchgefuehrten Rechnung weiter auf.

Ein erster Ansatz, den mittleren Fehler zu reduzieren, besteht darin, eine guenstiger liegende Interpolations-Gerade zu waehlen (siehe Bild 2).

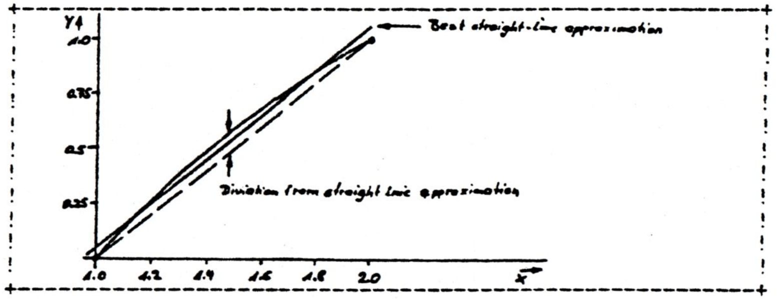


Bild 2: Lineare Interpolation fuer y = log (X) im Intervall 1 < X < 2. Die durchgezogene Gerade stellt die guenstigste Loesung dar.

Um die gewuenschte Interpolations-Gerade zu erhalten, genuegt die Addition einer Konstanten in Hoehe von 5% oder 13 Teile von 256. Die Addition dieser Konstanten reduziert den max. Fehler auf 4% und den mittleren Fehler auf Null. Damit wird der Rechenfehler zu Null, sofern die Verteilung der auftretenden Fehler gleichmaessig ist. Andernfalls koennen bei einseitiger Fehlerverteilung dennoch groessere Rechenfehler entstehen, z.B. wenn der Logarithmus von Zahlen in Intervallmitte oder an den Intervallgrenzen gebildet wird.

Eine bessere Loesung ist es, den Term $(X - 2^N)/2^N$ dazu zu benutzen, eine 256 Byte umfassende Tabelle mit den tatsaechlichen Logarithmen-Werten zu adressieren. Der Worst-

Case-Fehler bei dieser Methode ist dann kleiner 1%, wobei der Rechenfehler auch hier gegen Null geht. Diese Operation benoetigt zusaetzliche 3 usec (Z80, 4Mhz) und zwei Tabellen (fuer den Logarithmus/Antilogarithmus).

```
COMPUTELOG: LD    B,0FH        ;MAX. LOOP-COUNT
LOGLOOP:    ADD   HL,HL        ;SHIFT HL LEFT
            JR    C,GOTN       ;FOUND N ON CARRY
            DJNZ  LOGLOOP      ;LOOP
            LD    H,B          ;RETURN ZERO IN HL
            RET                ;END OF CONVERSION
GOTN:       LD    L,H          ;L GETS 8 BIT REMAIN-
                               ;DERFRACTION
            LD    H,LOGTABLE   ;H GETS PAGE OF LOG-
                               ;LOOK-UP TABLE
            LD    A,(HL)       ;PERF. TABLE LOOK-UP
            LD    L,A          ;RETURN MANTISSA
            LD    H,B          ;RETURN CHARACTERISTIC
            RET                ;END OF CONVERSION
LOGTABLE:   EQU   088H         ;PAGE OF LOG LOOK-UP
                               ;TABLE
```

Bild 3: Die Z80 Assembler-Routine, welche den Binaer-Logarithmus fuer 16-Bit Integerzahlen ermittelt

```
ANTILOG:    LD    A,H          ;GET CHARACTERISTIC
            CP    10H          ;OVERFLOW ?
            JP    OVERFLOW
            LD    B,H          ;MOVE CHARACT. INTO
                               ;LOOP COUNT
            INC   B            ;INC LOOP-CNT BY 1
            LD    H,INVLOG     ;H GETS PAGE OF INV.
                               ;LOG TABLE
            LD    A,(HL)       ;PERFORM INV. LOOK-UP
            LD    HL,0000
            STC                ;SET CARRY
INVLOOP:    RL    L            ;CY <- L <- CY
            RL    H            ;CY <- H <- CY
            SLA   A            ;CY <- A <- CY
            DJNZ  INVLOOP      ;LOOP
            RET
OVERFLOW:   LD    HL,0FFFFH    ;MAX-INTEGER
            RET
INVLOG:     EQU   089H         ;PAGE OF INV. LOG TAB.
```

Bild 4: Sobald das Endergebnis in logarithmischer Form vorliegt,kann diese Z80 Routine den inversen Logarithmus bilden.

EIN BEISPIEL BEWEIST DIE BRAUCHBARKEIT

Die Routine im nachfolgenden Anwendungsbeispiel uebernimmt den zu verarbeitenden Wert in einem 16-Bit Register. Der zurueckgelieferte Logarithmus ist ebenfalls im 16-Bit Format. Die Charakteristik wird in einem 8-Bit Register uebergeben, ebenso wie die 8-Bit Mantisse (der verbleibende Rest). Die Ergebenisse aller Berechnungen sind grundsaetzlich in 16-Bit Format. Der sich daraus ergebende Wertebereich ist somit :

$$1 <= X <= 2^{16} - 1$$

Soll die Logarithmen-Rechnung zum Loesen von Gleichungen benutzt werden, ausgedrueckt in Form eines 16-Bit Logarithmus, so besitzen die Zwischenwerte einen Wertebereich von 1 ... 2^{255} . Dies ergibt sich aus der Tatsache, dass eine 8-Bit Charakteristik verwendet wird, die einen Maximalwert von 255 ermoeglicht, bevor ein Ueberlauf erfolgt. Die derzeitige Genauigkeit ist auf 8-Bit begrenzt, kann jedoch leicht erweitert werden.

Bild 3 zeigt eine Z80 Assembler-Routine fuer die Berechnung des 16-Bit Integer-Logarithmus (unter der Vorraussetzung das die Zahl positiv und groesser Eins ist). Der 8-Bit Exponent wird im Register H, die Mantisse im Register L uebergeben. Beim verwendeten Algorithmus wird das HL-Register so lange nach links geschoben, bis ein Uebertrag erfolgt. Die Subtraktion der dazu noetigen Anzahl von Schritten von 16 ergibt N, Register H enthaelt den Rest. Mithilfe dieses Restes kann nun eine 256 Byte umfassende Log.-Tabelle adressiert werden, welche den genauen Logarithmus enthaelt.
Die Laufzeit dieser Routine haengt von den zu verarbeitenden Daten ab, duerfte im Mittel jedoch (bei einem 4Mhz Takt bei ca. 100 usec) liegen.
Bild 4 zeigt eine Routine fuer die inverse Operation, also Log. -> Integer. Auch sie basiert auf dem oben beschriebenen Verfahren, wendet es jedoch invers an. Wiederrum ist hier die Laufzeit von den Daten abhaengig, wie auch hierfuer eine 256 Byte lange Tabelle vorhanden sein muss. Die mittlere Verarbeitungszeit (4Mhz Takt) betraegt ca. 120 usec.
Bild 5 zeigt ein PASCAL-Programm zur Berechnung der beiden Tabellen.

IM MITTEL WERTEUNABHAENGIGE LAUFZEIT

Es ist zu beachten, dass die Zeit zur Berechnung des Antilogarithmus fuer grosse Werte hoeher liegt als die zur Berechnung des Logarithmus (bei gleichem Wert). Aehnlich verhaelt es sich mit kleinen Werten. Hier ist jedoch die Rechenzeit fuer den Logarithmus laenger als fuer den Antilogarithmus. Deshalb wird die Gesamtzeit zur Berechnung des Logarithmus einer Zahl mit anschliessender Rueckwandlung im Mittel ueber den in Frage kommenden Zahlenbereich konstant sein.
Als eine Art Anwendungsbeispiel zeigt Bild 6, wie die

Routinen aus Bild 3 & 4 fuer die Berechnung der Gleichung $Y = X^{2.5}$ benutzt werden koennen.

```
+--------------------------------------------------------+
! PROGRAM LOG.TABLES (INPUT,OUTPUT)                      !
! VAR                                                    !
!     I:INTEGER                                          !
! BEGIN                                                  !
!     WRITELN (' LOG TRANSFORMATION TABLE ...');         !
!     WRITELN (' ===========================');          !
!     FOR I:=0 TO 255 DO                                 !
!       WRITELN(I:5,ROUND(256*LN(1+I/256)/LN(2)):9;      !
!     FOR I:=0 TO 10 DO  WRITELN;                        !
!     WRITELN (' ANTILOG TRANSFORMATION TABLE ...');     !
!     WRITELN (' ===============================');      !
!     FOR I:=0 TO 255 DO                                 !
!       WRITELN (I:5,ROUND(256*(EXP(I/256*LN(2))-1)):9;  !
! END.                                                   !
+--------------------------------------------------------+
```

Bild 5: Das PASCAL-Programm fuer die Erstellung der Log und Antilog-Tabellen zu den Routinen aus Bild 3 und Bild 4.

Das Register HL uebergibt das Argument X bei Aufruf und und enthaelt das Resultat nach Rueckkehr von den jeweiligen Wandlungs-Routinen. Bemerkenswert ist die Methode, wie das Programm in Bild 6 die Quadratwurzel (mittels Rechtsschieben) und das Quadrat (mittels Linksschieben) bildet.

SO WERDEN DIE KOSTEN FUER EINEN CO-PROZESSOR GESPART

Die mittlere Ausfuehrungszeit der Routine aus Bild 6 liegt bei ca. 250 usec (4Mhz CPU Takt). Im Gegensatz hierzu benoetigt ein INTEL 8231A, ein adequater Co-Prozessor fuer einen Z80, 2.5 msec um einen Zahl zu quadrieren (Der Ehrlichkeit halber muss jedoch gesagt werden, dass der 8231A fuer das genannte Beispiel etwas zu starker Toback ist. Beherrscht er doch 16 und 32-Bit Gleitkomma-Arithmetik in den ueblichen mathematischen Funktionen).

Ein weiteres Beispiel (Bild 7) soll die Moeglichkeiten der Routinen aus Bild 3 & 4 zeigen. Die Routine in Bild 7 berechnet die nachfolgende Gleichung :

$$X = \sqrt{\frac{A^2}{313} + \frac{B^2}{462}}$$

Das Argument A (16 Bit) und das Argument B (ebenfalls 16 Bit) wird in den Registern HL und DE uebergeben.
Das Ergebnis wird im Register HL zurueckgeliefert.
Beide Argumente muessen >= 1 sein.
Hervorzuheben ist, dass die Zwischenergebnisse bei der Quadrierung von A und B im logarithmischen Masstab vorliegen und somit nicht durch ein Zahlenueberlauf verfaelscht werden koennen, auch nicht bei sehr grossen

Zahlen. Eine Codeoptimierung ist nicht erfolgt, trotzdem betraegt die Laufzeit nur 700 usec bei 4Mhz Takt. Der 8231A benoetigt fuer die gleiche Aufgabe ca. 500usec

Man kann die in Bild 3 & 4 gezeigten Routinen auf manche Weise erweitern. Eine waere z.B. die Erhoehung der Genauigkeit, eine andere die Erweiterung auf die Moeglichkeit zur Verarbeitung von gebrochenen Ergebnissen.

```
EXP2FIVE: CALL COMPUTELOG
          PUSH HL         ;SAVE LOGARITHM
          SRL  H          ;0  -> H -> CY
          RR   L          ;CY -> L -> CY
          PUSH HL         ;PUSH SQUARE ROOT
          POP DE
          POP DE
          POP HL          ;GET LOGARITHM
          ADD HL,DE       ;COMPUTE SQUARE
          ADD HL,DE       ;ADD IN SQUARE ROOT
          CALL ANTILOG    ;CONVERT BACK TO BINARY
          RET
```

TEST-DATA

INPUT	IDEAL RESULT	ACTUAL RESULT	ERROR
14	733	734	0.08%
27	3788	3776	0.3%
50	17678	17664	0.07%
63	31503	31488	0.05%
61	59049	59006	0.07%

BILD 6: Man kann Logarithmen-Rechnung in Routinen wie dieser benutzen um Zahlen zu Potenzieren, sogar wenn der Exponent gebrochen ist.

Einer von zwei Wegen die Genauigkeit zu erhoehen ist, weiterhin mit 16 Bit zu arbeiten, den Exponenten aber nur mit 4 Bit darzustellen. Dann stehen 12 Bit fuer die Mantisse zur Verfuegung. Das aber wiederrum begrenzt den moeglichen Zahlenbereich der Zwischenergebnisse (wegen des 4 Bit Exponenten).
Der andere Weg ist, den Exponenten weiterhin bei 8 Bit zu belassen, dafuer dann aber die Mantisse auf 24 Bit zu verlaengern. Das macht dann aber die Benutzung eines zusaetzlichen 8-Bit Registers erforderlich. Gleichgueltig welcher Weg beschritten wird, er bedeutet eine Aenderung in der Art wie die Log.-Tabelle angesprochen wird. Ein Beispiel :
Ein geaenderter Algorithmus im Sinne des zweiten Weges, wuerde die Mantisse so erzeugen. Zuerst wird der verbleibende Rest aus der Schiebeaktion als 16-Bit Zahl ausgedrueckt. Die oberen 8 Bit dienen nun zur Adressierung der 16-Bit Log.-Tabelle. Sobald nun der 16-Bit Logarithmenwert vorliegt, werden die unteren 8 Bit linear interpoliert und hinzuadiert. Hierfuer wird die Interpolations-Methode aus Bild 2 benutzt. Das hinzufuegen der unteren 8 Bit, wie genannt, verringert den Interpolation-Fehler einer nur 256 Eintraege umfassenden 16-Bit Tabelle.

Sollten in irgend einer Applikation einmal gebrochene Zahlen von noeten sein, so kann der Algorithmus dahingehend geaendert werden, dass alle 16-Bit Eingaben als 8-Bit Integer (high Byte) und 8-Bit gebrochener Anteil (low Byte) aufgefasst werden. Grundsaetzlich muss hierfuer der Initialisierungszaehler des Algorithmus geaendert werden, dieser Schleifenzaehler bestimmt den Exponenten. Wird die genannte Aenderung durchgefuehrt, dann liefert eine Zahl, deren 'most significant binary one' des high Byte im low Byte liegt, einen negativen Exponenten.

```
RMS:   CALL   COMPUTELOG
       ADD    HL,HL      ;PERFORM SQUARE
       LD     BC,LG313   ;BC GETS LOG OF 313, COULD
                         ;EASILY BE ANOTHER PARAM.
       OR     A          ;CLEAR CARRY
       SBC    HL,BC      ;PERFORM DIVISION
       CALL   ANTILOG    ;CONVERT BACK TO BINARY
       PUSH   HL         ;SAVE FIRST TERM
       PUSH   DE         ;PUSH TERM B
       POP    HL
       CALL   COMPUTELOG
       ADD    HL,HL      ;PERFORM SQUARE
       LD     BC,LG462   ;BC GETS LOG OF 462
       OR     A          ;RESET CARRY
       SBC    HL,BC      ;DO DIVISIN BY 462
       CALL   ANTILOG    ;CONVERT BACK TO BINARY
       POP    DE         ;GET FIRST TERM
       ADD    HL,DE      ;PERFORM ADDITION
       CALL   COMPUTELOG ;COMPUTE LOG OF THE SUM
       SRL    H          ;0 --> H --> CY, PERFORM
                         ;SQUARE ROOT
       RR     L          ;CY --> L --> CY
       CALL   ANTILOG    ;CONVERT BACK TO BINARY
       RET               ;READY
LG313 EQU     084AH      ;LOG OF 313
LG462 EQU     08DAH      ;LOG OF 462
```

Test-Data

A	B	IDEAL RESULT	ACTUAL RESULT
1000	2000	108.9	108
782	418	48.8	48
219	87	13.0	13

Bild 7 : Ein Beispiel fuer die Moeglichkeiten, die in der Logarithmenrechnung stecken. Die gezeigte Routine berechnet die Quadratwurzel der Summe von Quadraten.

Manchmal erscheint eine Gleichung durch Logarithmenrechnung nicht loesbar zu sein. In der Regel ist dies aber nur ein Umformungsproblem. Man denke sich ein Problem, welches die Sinus-Funktion benutzt. Weiterhin sei eine Routine benoetigt, die eine 16-Bit Integerzahl mit dem Cosinus eines Winkels multipliziert (beide als Variable vorliegend). Der Winkel sei dabei auf dem Bereich

$$-90^\circ < \theta <= +90^\circ$$

begrenzt. Die Gleichung

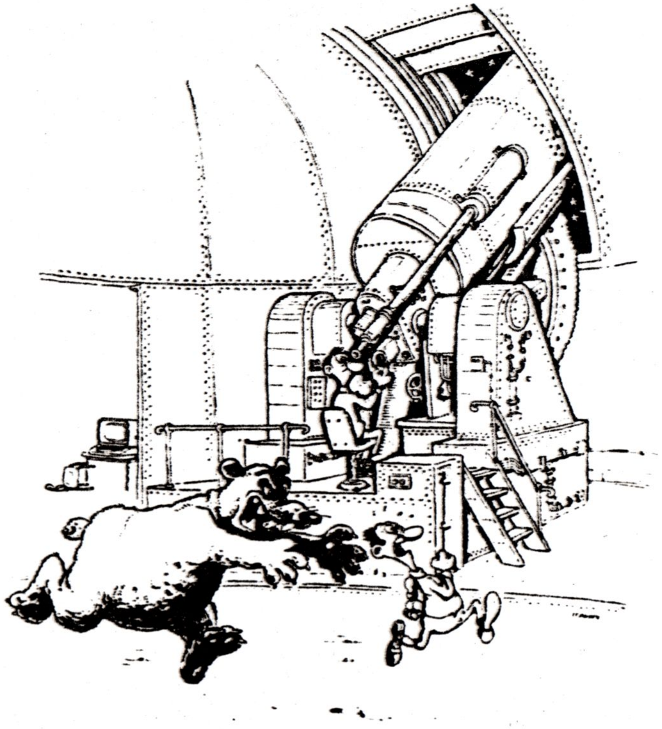
$$Y = X * COS(TETA)$$

kann nicht direkt geloest werden, da log(cos(90)) nicht definiert ist. Umformen hilft hier weiter :

$$Y = X * (1 + COS(TETA)) - X$$

In dieser Form ist das Problem fuer unseren Algorithmus verdaulich. Eine passende Routine ist nun relativ schnell codiert. Jedoch wird eine dritte Tabelle benoetigt, in der die Werte fuer log(1+cos(teta)) abgelegt sind.

ENDE



Ich hab' doch nur vergessen einzugeben, „suche das Sternzeichen Großer Bär".

Ich wünsche allen Clubmitgliedern, die Spaß am Softwaretüfteln haben, viel Vergnügen bei der Lektüre.

Euer Karl Müller

Floppytrick's

Ergänzung
Die im letzten Info veröffentlichte 80/40-Spurumschaltung wurde inzwischen von mir in das neue Model 4P von Günther Wagner eingebaut. Dabei stellte sich heraus, daß die Impulsbreite der durch die Schaltung zusätzlich erzeugten Stepimpulse für die verwendeten Shugartlaufwerke nicht ausreichten. Dieses Problem ließ sich jedoch durch Erhöhen der Kapazität des 56 nF-Kondensators aber leicht beheben. Ein genauer Wert ist hier nicht anzugeben, es muß nach der "Try and Error"-Methode vorgegangen werden. Die Schaltung funktioniert nun schon längere Zeit problemlos!

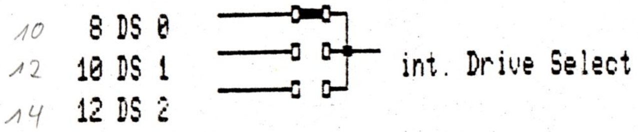
Noch eine Umschaltmöglichkeit
Für User die jeweils mindestens ein 40'er und ein 80'er Laufwerk ihr Eigen nennen, gibt es auch eine Möglichkeit Systemdisketten in beiden Spurzahlen zu booten. Die Möglichkeit besteht darin, die Drive-Select-Leitungen (DS) umzuschalten.
Die Auswahl des jeweils angesprochenen Laufwerks geschieht durch die Leitungen 10, 12 und 14 (Drive 0, 1 und 2) des Shugart-Bus. Beim Model 1 wird die Leitung 32 in Systemen die nur singlesided Laufwerke besitzen, zur Anwahl des Drives 3 benutzt. Sind Doppelkopflaufwerke im System, dient die Leitung zur Ansteuerung des "Hinterkopfes" (Backside select BS). Bei Model 3/4/4P-Geräten wird Laufwerk 3 durch die Leitung 6 selektiert und 32 dient als BS-Signal.

Belegung des Shugart-Bus:
alle ungeraden Pins führen Masse!

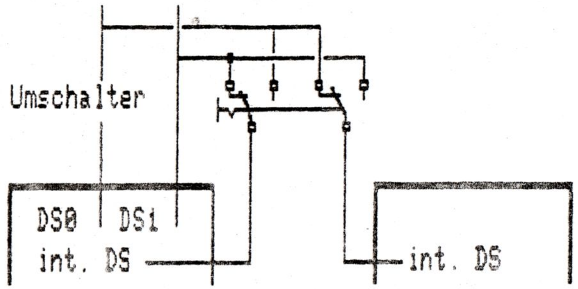
2	=	NC	nicht angeschlossen	
4	=	NC	" "	
6	=	DS 3 (NC)	Drive select 3 (im Model 1 NC)	
8	=	Index Pulse		
10	=	DS 0	" "	0
12	=	DS 1	" "	1
14	=	DS 2	" "	2
16	=	Motor ON		
18	=	Direction Select		
20	=	Step		
22	=	Write Data		
24	=	Write Gate		
26	=	Track Zero		
28	=	Write Protect		
30	=	Read Data		
32	=	BS (DS 3)	Backside select (im Model 1 auch DS 3)	
34	=	NC	nicht angeschlossen	

Im Normalfall gehen alle diese Signale an alle angeschlossenen Laufwerke. Welches Laufwerk auf welche Adresse (0,1,2,3) reagiert, wird erst auf den Drives durch sog. Jumper oder kleine Schalter eingestellt. Das heißt, daß ein Laufwerk, welches als Drive 0 benutzt wurde, durch Umstecken einer Brücke oder Umschalten eines Schalters auch als Drive 1,2 oder 3 verwendet werden kann!

Prinzipschaltbild:



Entfernt man nun die Jumper von zwei Laufwerken und verschaltet sie nach untenstehendem Schaltbild, kann man durch Umschalten wählen, welches Laufwerk welche Adresse bekommt! Natürlich kann man die Schaltung auch mit 3 Laufwerken verwenden, man benutzt einfach DS 2 statt DS 1. Damit bleibt Laufwerk 1 immer unverändert und Drive 0 und 2 tauschen ihre Plätze im System.



Probleme mit dem Model 3
Bei original Model 3/4/4P - Geräten hat Tandy übrigens einen Fallstrick für Leute eingebaut, die ihr System selbst mit Doppelkopflaufwerken aufrüsten wollen. Die Radio Shack - Leute haben nämlich aus den Floppysteckern all die Pins herausgelassen, die bei den SS - Originallaufwerken nicht gebraucht werden. So fehlen im Stecker für Drive 0 die Pins 6, 12, 14 und 32. Dadurch können mit diesen Kabeln normalerweise keine DS - Laufwerke eingebaut werden, man muß sich neue Kabel besorgen. Mit folgendem Trick kann man sich eine Menge Geld sparen, da man die alten Kabel weiterverwenden kann!

Man entferne das Floppykabel und stecke alle Anschlüsse (auch den am Floppycontroller!!!) um 180° gedreht wieder auf. Das funktioniert, weil alle ungeraden Stifte auf einer Seite liegen und normalerweise Masse führen. Werden alle Stecker um 180° gedreht, sind auf der "Signalseite" nun alle Pins vorhanden und das Fehlen von 4 Masseleitungen (von insg. 17) ist nicht tragisch.

Speedjustage
Nochmals zur Justierung der Laufwerksgeschwindigkeit und damit zu LWT und der Anmerkung von Bernd Retzlaff zu meinen Änderungsvorschlägen in diesem Programm. LWT kann meiner Meinung nach immer nur eine Hilfe sein, ohne das Gerät öffnen zu müssen, die Laufwerksgeschwindigkeit festzustellen. Zur genauen Justage bedient man sich am besten des auf dem "Schwungrad" des Laufwerks angebrachten Strop kops.

Dazu wird das Laufwerk mit einer durch Netzspannung betriebenen Lichtquelle beleuchtet und das Justierpoti so lange gedreht, bis sich die aufgemalten Striche scheinbar nicht mehr bewegen! Vorsicht, auf der Scheibe sind meist zwei Einteilungen für 50 und 60 Hz Netzfrequenz aufgetragen. Also nicht die falsche benutzen!

So, das waren vorerst alle Tricks und Tips, die ich zur Zeit auf Lager habe. **Zum Schluß eine Bitte:** "Kramt doch mal in eurer Trickkiste, sicher fördert ihr etwas Interessantes zu Tage, was sich zu veröffentlichen lohnt. Auch wenn es auf den ersten Blick recht banal aussieht, kann man anderen Mitgliedern damit einen großen Schritt weiterhelfen!" In diesem Sinne viel Glück, Euer

Hartmut Obermann

An alle HRG-1b Benutzer !

Als ich mir vor ca. zwei Jahren die Grafik-Erweiterung der Fa. RB-Elektronik kaufte, hatte ich noch keine Vorstellung von dem, was noch im Kaufpreis enthalten war. Ich spreche von den technischen Unzulaenglichkeiten der Karte und den Loesungsvorschlaegen !.

Angefangen hat es damit, dass das Laufwerk 0 gelegentlich beim Nachladen irgentwelcher System-Module nicht so recht wusste 'was Sache war' und recht geraeuschvoll einmal ueber die Diskette und zurueck steppte, bis die richtige Spur eingestellt war. Eine Anfrage bei RB-Elektronik bestaetigte meinen Verdacht: Es stimmte etwas nicht. So erfuhr ich dann, dass beim ansprechen der Grafik-Karte der Disk-Kontroller gestoert wird und beim naechsten Diskzugriff ersteinmal nach Spur 0 laeuft und dann die angeforderte Spur anfaehrt. Das Problem sollte sich gemaess Auskunft durch umsetzen der MREQ-Leitung auf MI beheben lassen (???). Den Grund konnte man mir leider nicht erklaeren. Hier tappte der Meister selbst im Dunklen. Leider funktionierte der Trick bei meinem Rechner nicht, warum auch immer. So hatte ich den Schwarzen Peter zurueck und durfte nun in eigener Sache nach dem Grund suchen. Der war dann auch bald gefunden : Falsches Hardware-Design!

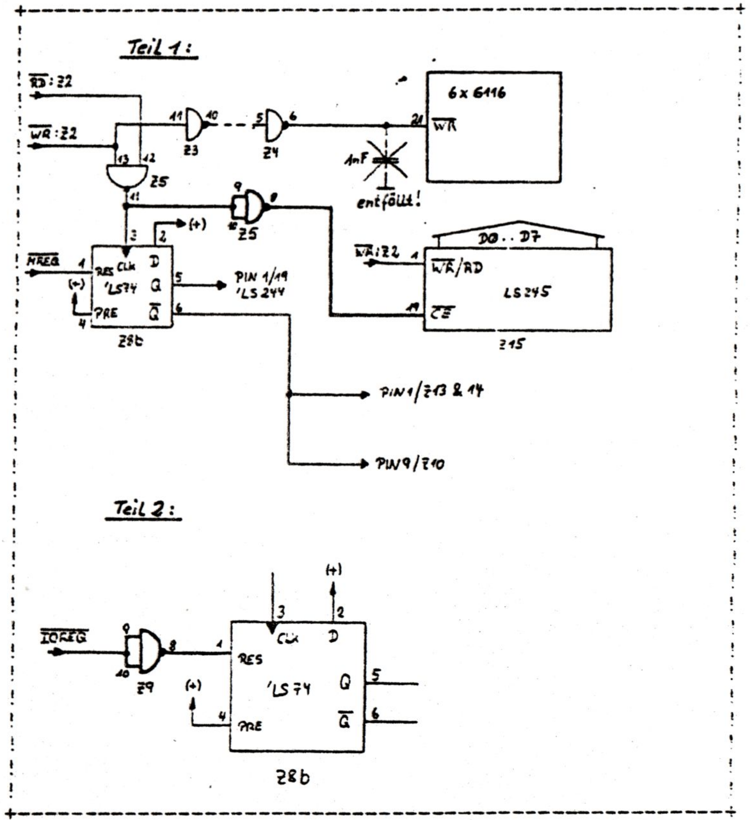
Sobald der Rechner auf die Grafik-Karte zugreifen will, wird das Flip-Flop Z8b umgeschaltet und dadurch die Ausleselogik der Karte gesperrt. Gleichzeitig wird der Datenbus-Treiber Z15 freigeschaltet. Ueber PIN1 von Z15 wird bei einem Schreibzugriff ausserdem noch die Richtung umgeschaltet. Die Uebertragungsrichtung nach dem Zugriff auf die Karte ist in Richtung Datenbus (!!!) und wird erst mit dem naechsten MREQ-Zyklus aufgehoben. Bis dies geschieht, liegt ein Buskonflikt vor mit den oben genannten Folgen.
Der PIN19 von Z15 darf deshalb nicht an PIN6 von Z8b angeschlossen sein !. Dieser Pin gehoert an PIN11 Z5 oder an PIN3 Z8b. Hier liegt die NOR-Verknuepfung der RD & WR Clocks von Z2 an. Mit diesem Signal erfolgt die Freigabe von Z15 immer dann, wenn dies auch erforderlich.Ansonsten ist Z15 immer im TRI-State Zustand und stoert den Datenbus nicht mehr.
Allerdings muss das Signal zuvor invertiert werden, ansonsten passiert genau das Umgekehrte. Den benoetigten Inverter kann Z5 oder Z9 liefern. Hier sind noch einige frei.

Ausserdem sollte noch der Kondensator am PIN21 'WR' der 6116 RAM's entfernt werden. Er sollte die Probleme wohl etwas 'verrunden', verursacht aber bei abgeschalteter Grafik und '#CLS' Stoerungen auf dem Monitor. Wer ganz sauber arbeiten will, kann noch das MREQ-Signal von PIN1 Z8b 'RES' abklemmen und stattdessen das IOREQ-Sig. ueber einen Inverter benutzen. Dann wird die Kartenlogik mit Beendigung eines jeden Portzugriffs zurueckgeschaltet und nicht erst mit dem naechsten RAM-Zugriff.

Damit verringert sich die groesse der 'Schneeflocken' auf dem Monitor etwas.

Wem dies zu aufwendig ist, kann auch alles lassen wie es ist. Man sollte es sich dann allerdings verkneifen, die Grafik auf Diskette abspeichern zu wollen. Bedingt durch die 'Verwirrung' des Controlers kann das angesprochene Laufwerk (bei bereits gut gefuellter Diskette) beschaedigt werden,da der Kopfschlitten wiederholt gegen den Anschlag gefahren wird.

Viel Spass



Auszug aus dem Schaltplan der HRG1B mit den Aenderungen

256K RAM für Z80-Systeme

Helmut Bernhardt

"64K-Byte RAM sind die obere Grenze für den vom Z80 adressierbaren Speicher". Diese Behauptung ist nur dann richtig, wenn die direkte Adressierung gemeint ist. Durch Banking d.h. Umschalten eines Teilbereichs des 64K-Adreßraumes auf parallele Speicher läßt sich der vom Z80 nutzbare Speicher beliebig ausbauen; 1M-Byte-RAM-Karten werden heute für fast alle Z80-Systeme angeboten.

Gegenüber solchen Karten hat die im folgenden beschriebene Speichererweiterung den Vorteil, daß sie für wenig Geld zu realisieren ist und weder an einen bestimmten Busstandard noch gar einen speziellen Computer gebunden ist. Einzige Voraussetzung ist ein Z80-Prozessor.

Der Nachteil des Projektes besteht darin, daß man sich auf dem CPU-Board ein bißchen auskennen muß und daß es ohne Lötkolben doch nicht geht.

Das Grunprinzip der Speichererweiterung ist ein Ersetzen der vorhandenen 4164-RAMs (oder auch 4116-RAMs) durch die pinkompatiblen 41256-Typen und eine mit wenigen Standard-ICs aufgebaute Banking-Logik, mit der ein gezielter Zugriff auf die vielen K-Byte möglich ist.

Die Banking-Logik

Vom Z80 aus gesehen ist der direkt adressierbare Speicherraum von 64K-Byte in zwei Blocks von 32K-Byte Größe unterteilt. Die obere Hälfte (Adressen 8000H-FFFFH) ist dabei nicht umschaltbar und deshalb vom Z80 immer erreichbar (Common-Bereich). Interrupt-Service-Routinen, der Stack und alle Routinen, auf die ständig zugegriffen werden muß, müssen hier liegen.

Die untere Hälfte ist in mehreren parallelen 32K-Blocks vorhanden. Welcher dieser Blocks im Adreßraum 0000H-7FFFH für den Z80 jeweils verfügbar sein soll, läßt sich durch Ausgabe der Nummer des Blocks (Bank Nr.) an einen Port (Latch) einstellen. Der Block ist dann solange als Speicher verfügbar, bis eine andere Bank Nr. ausgegeben wird. Durch RESET* wird beim Einschalten des Computers automatisch immer die Bank 0 eingestellt, um bei der Systeminitialisierung definierte Verhältnisse vorzugeben.

Von der Aufteilung der 256K (512K, 1024K) RAM her gesehen sind die Verhältnisse etwas anders zu sehen. Die unteren 64K sind nach dem Einschalten des Computers diejenigen, auf die der Z80 zugreifen kann. Alle anderen 32K-Blocks können anstelle des ursprünglichen 32K-Blocks in die untere Hälfte des Adreßraumes des Z80 gelegt werden, wenn eine der Bank Nummern 1-7 (15, 31) ausgegeben wird.

Das wird dadurch erreicht, daß durch die Latchausgänge Q1 und Q2 die Adressen A16 und A17 vorgegeben werden, die einen der vier 64K-Blocks innerhalb eines 256K-Blocks anwählen, Q3 und Q4 codieren einen der bis zu vier 256K-Blocks und Q0 und A15 des Systembus werden dafür herangezogen, die Adresse A15' für den Speicher zu liefern, um dort die untere oder obere Hälfte eines 64K-Blocks zu adressieren.

Die Herleitung von A15' über OR-Verknüpfung von A15 und Q0 sorgt dafür, daß immer dann, wenn A15 high ist, die obere Hälfte des 64K-Blocks angewählt wird (Common-Bereich), während dann, wenn A15 low ist, der Pegel des Latchausgangs Q0 den Pegel von A15' bestimmt und so zur richtigen Anwahl der eingestellten Bank beiträgt.

Da nun aber bei einer Adresse im Common-Bereich (A15=1) nicht nur die obere Hälfte irgendeines durch Q1 bis Q4 codierten 64K-Blocks im RAM sondern ausschließlich die obere Hälfte des unteren 64K-Blocks angewählt werden soll, wird durch den Pegel von A15 am STB*-Eingang des 74LS157-Multiplexers (Abb.1) Low-Pegel ausgegeben.

Der Rest der Schaltung erfüllt die Aufgabe, zu erkennen, ob Bank Nr.1, die den gleichen 32K-Block im RAM wie der Common-Bereich belegen würde, eingestellt ist, und daraufhin die höchste verfügbare Bank zu adressieren. Das geschieht dadurch, daß bei eingestellter Bank Nr.1 das Bitmuster an den Latchausgängen zu einem High-Pegel am Select-Eingang des Multiplexers führt, der dann die Pegel der B-Eingänge, die über 4K7 an +5V gelegt sind, durchschaltet. Diese Sicherung des Common-Bereichs führt dazu, daß unter den Bank-Nummern 1 und 7 (15, 31) der gleiche 32K-Block adressiert wird. Die Bank 7 (bei 256K), 15 (bei 512K) bzw. 31 (bei 1024K) sollte also nicht benutzt werden.

Mehr als 256K RAM lassen sich dann realisieren, wenn im Computer mehrere Reihen mit 8 Sockeln für RAMs vorhanden sind, oder indem die RAM-ICs übereinandergelötet werden, wobei dann Pin 15 (CAS*) des aufgelöteten ICs hochgebogen werden muß. Durch die Drahtbrücken A-E muß die Schaltung dem Speicherausbau angepaßt werden. Je nach Speichergröße sind folgende Drahtbrücken zu legen:

256K	1 Reihe RAMs	Brücken A und D
512K	2 Reihen RAMs	" B " E
1024K	4 Reihen RAMs	" C " E

Diese Drahtbrücken sorgen dafür, daß je nach Speicherausbau der durch Bank Nr.1 angewählte 32K-Block identisch ist mit dem durch die höchste mögliche Bank-Nummer freizugebenden Block.

Bei mehr als 256K RAM müssen die einzelnen 256K-Blocks durch die Signale CASEN1* bis CASEN4* (bzw. deren Verknüpfung mit CAS* über OR-Gatter) freigegeben werden. Bei nur 256K RAM braucht die CAS*-Freigabe, wie sie vorher für 64K vorgelegen hat, nicht geändert zu werden.

Wenn 512K RAM realisiert worden sind, muß auf dem Banking-Board die Leiterbahn zu Pin13 des 74LS139 durchtrennt werden, an diesen Pin ist stattdessen GND zu legen, um zu verhindern, daß bei Ausgabe der Bank-Nr.1 nicht die nicht vorhandene Bank31 anstelle der Bank15 eingestellt wird.

Das Refreshing

Ein weiteres Problem stellt neben der Adressierung innerhalb der 41256-RAMs noch das Refreshing dar. Gegenüber den meisten 4164-Typen mit 128 Refresh-Zyklen (nur einige Ausnahmen) benötigen 41256-RAMs grundsätzlich 256 Refresh-Zyklen. Der Z80 liefert bei seinem Hidden Refresh aber nur eine 7-Bit-Refresh-Adresse.

Dieses Problem ist aber von Michael Hungershausen (c't 9/84, S.22) schon gelöst worden und soll deshalb hier nicht weiter abgehandelt werden. Der Vollständigkeit halber ist in Abb.1 eine der beiden vorgestellten Varianten für die Herleitung einer 8-Bit-Refresh-Afresse wiedergegeben.

Der Adreßmultiplexer

Ebenfalls in Verbindung mit dem Refreshing ist folgendes Problem zu sehen. Durch den Zyklus von 128 Refresh-Adressen ist es nötig, daß beim Adreß-Multiplex die Adressen A0 bis A6 zusammen durchgeschaltet werden. Anstelle von A7 kann aber auch irgend eine höhere Adresse zusammen mit A0-A6 durchgeschaltet werden. Bei Computern, in denen ursprünglich 4116-RAMs verwendet wurden, und die dann später auf 4164-RAMs umgerüstet wurden, muß sogar davon ausgegangen werden, daß dies der Fall ist, da für die interne Adressierung von 4116-RAMs die Adressen A0-A6 und A7-A13 abwechselnd durchgeschaltet werden.

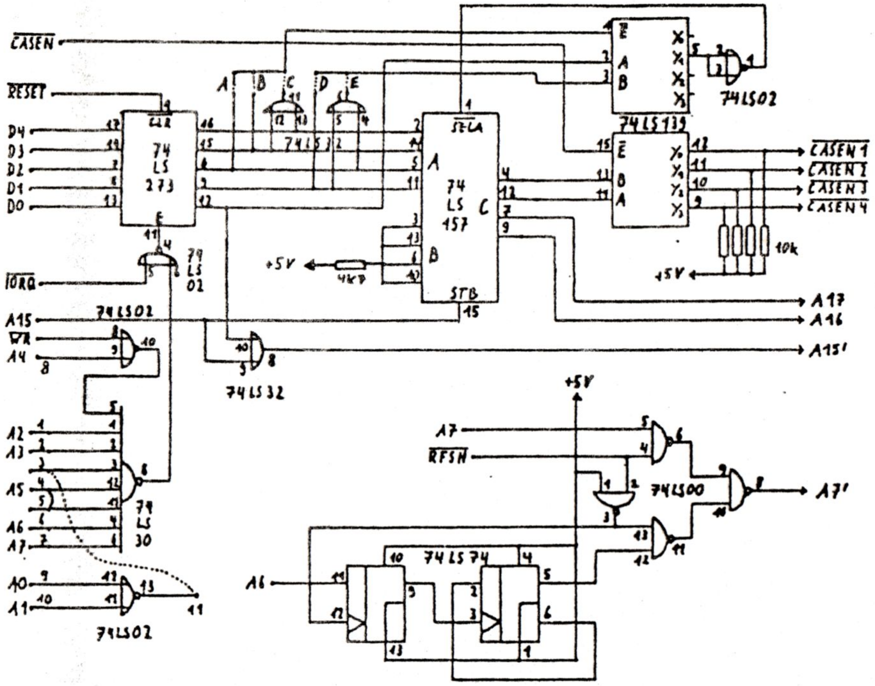
Es muß bei den Adreßmultiplexern also ermittelt werden, ob A7 mit A0-A6 zusammen durchgeschaltet wird. Wenn dies nicht der Fall ist, muß ermittelt werden, wo A7 anliegt und wo eine höhere Adresse mit A0-A6 zusammen durchgeschaltet wird. Beide Signale sind von den Pins der Multipexer abzutrennen; an den Pin, wo vorher die höhere Adresse anlag, ist A7' (Abb.1) zu legen und die dort abgetrennte Adreßleitung ist dahin zu führen, wo vorher A7 anlag.

Außerdem muß durch ein huckepack-aufgelötetes 74LS157 (auf einen der schon vorhandenen Multiplexer; bei 74LS157 können die Pins 1,8,15 und 16 direkt angelötet werden) noch das Multiplexen von A16 und A17 möglich gemacht werden. Der Ausgang MA8 (Abb.2) wird (gegebenenfalls über einen 33-Ohm-Widerstand) an die Pins 1 der RAMs geführt.

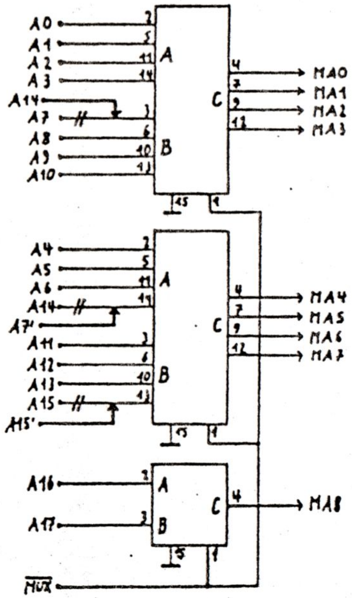
Anstelle von A15 muß das Signal A15' (Abb.1) an die Adreßmultiplexer geführt werden.

Bei Computern mit 4116-RAMs

müssen alle Speicherchips entfernt werden, und bei Vorhandensein getrennter Daten-Treiber für verschiedene 16K-Blöcke muß auch die Freigabe der Treiber nicht benutzter RAM-Reihen durch Anlegen von +5V über einen 4K7-Widerstand an den E*-Eingang der Treiber verhindert werden.



1) Schaltung des Banking-Boards und Generierung einer 8-Bit Refresh-Adresse



2) Änderung und Erweiterung des Adreßmultiplexers für die RAM-Adressen am hypothetischen Beispiel eines Computers, der schon von 4116-RAMs auf 4164-RAMs umgerüstet worden ist. Die Belegung der A-Eingänge mit den niederen Adressen ist nicht zwingend (siehe mc-CP/M-Computer). Von einem zusätzlich nötigen 74LS157 wird nur 1/4 benötigt.

Da diese Speicherchips mit -5V an Pin 1, +12V an Pin 8 und +5V an Pin 9 versorgt werden, müssen diese Leitungen von den Versorgungsspannungen abgetrennt werden. Die Versorgungsspannung +5V wird an die Pins 8 gelegt. Bei den die Pins 1 und die Pins 9 verbindenden Leitungen werden alle Abblockkondensatoren entfernt. An die Pins 9 wird MA7 und an die Pins 1 wird MA8 der Adreßmultiplexer gelegt.

Das CAS*-Signal an Pin 15 der 4116-RAMs ist ein mit einem 16K-Freigabesignal (meistens durch OR, 74LS32) verknüpftes CAS*. Anstelle des 16K-Feigabesignals muß ein low aktives (bei OR-Verknüpfung) Freigabesignal, das den gesamten als RAM vorgesehenen Adreßraum des Z80 umfaßt (nur ROM und memory-mapped I/O ausgeblendet) oder bei Speichererweiterung auf mehr als 256K eines der Signale CASEN1* - CASEN4* verwendet werden.

Banking der oberen 32K

ist nur dann sinnvoll, wenn es sich um einen BASIC-in-ROM-Computer, der nicht CP/M-fähig ist, handelt, und läßt sich durch Invertieren von A15 (durch z.B. 74LS04) für die Schaltung in Abb. 1 erreichen.

Einbau des Banking-Boards

Die Wahl der Portadresse für das Latch zum Einstellen der Bank Nr. wurde flexibel gehalten. Die Portadresse kann sich aus maximal 7, minimal 5 high aktiven und maximal 3, minimal 1 low aktiven Adreßbits zusammensetzen.

Low aktive Adreßleitungen werden an die Punkte 8 bis 10 angeschlossen. Der Punkt 8 muß auf alle Fälle belegt werden. Wenn dann nur noch eine zusätzliche Adreßleitung low sein soll, muß diese an beide Punkte (9 und 10) gelegt werden. Wenn die Punkte 9 und 10 belegt werden, muß Punkt 11 mit einem der Punkte 1-7 verbunden werden. Die high aktiven Adreßleitungen werden an die restlichen Punkte (der Punkte 1-7) gelegt. Nicht belegte Punkte (der Punkte 1-7) werden mit belegten Punkten verbunden. Selbstverständlich müssen A6 und A7 auch noch an die dafür vorgesehenen Punkte (n und m) neben dem 74LS00 angeschlossen werden.

Die Adreßdecodierung sei am Beispiel der Portadresse ECH genauer beschrieben: ECH hat folgendes Bitmuster

A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	1	1	0	0

Es müssen dann A0, A1 und A4 als low aktive Adreßleitungen an die Punkte 8, 9 und 10 gelegt werden und wegen der Belegung der Punkte 9 und 10 muß Punkt 11 mit z.B. Punkt 3 verbunden werden. Die Adreßleitungen A2, A3, A5, A6 und A7 werden dann z.B. an die Punkte 1, 2, 4, 6 und 7 angeschlossen. Der unbenutzte Punkt 5 muß dann mit z.B. Punkt 4 verbunden werden.

Das Signal CASEN* ist das CAS*-Freigabe-Signal für einen mit 4164-RAMs bestückten Speicher. Wenn vor dem Umrüsten auf 41256-RAMs noch 4116-RAMs vorhanden waren, muß ein solches Signal noch hergeleitet werden. CASEN* muß bei allen Adressen, die weder ROM noch memory-mapped I/O ansprechen, low aktiv sein. Bei TRS 80 und Kompatiblen ist dies ein Signal 16-64K*, das durch Invertieren des Signals 0-16K* und - leider nicht überall durchgeführt - OR-Verknüpfung mit MERQ* hergeleitet wird.

Die Freigabe des Datenbus-Transceivers

für die RAMs und eventuell auch ROMs kann in einigen Computern größere Schwierigkeiten bereiten. Am einfachsten gestaltet sich die Lösung dieses Problems, wenn nur 8 Stück 4164-RAMs vorhanden sind und der zugehörige Datenbus-Transceiver ausschließlich die RAMs bedient. In diesem Fall kann die Freigabe des Transceivers so, wie sie war, belassen werden.

Auch dann, wenn vorher mehrere Reihen 4116-RAMs vorhanden waren und später durch Umrüsten auf 4164-RAMs in einer Reihe und Freilassen aller anderen Sockel die Freigabe eines für alle RAMs gemeinsamen Datenbus-Transceivers auf die 4164-RAMs angepaßt wurde, braucht bei Verwendung nun wieder aller Sockel die Freigabe des Transceivers nicht mehr geändert zu werden.

Bei Vorhandensein je eines Transceivers für eine Reihe RAMs (ursprüglich 4116) und nun Bestückung von 2 oder 4 Reihen mit 41256-RAMs müssen zur Freigabe der Transceiver die Signale CASEN1* bis CASEN4* des Banking-Boards verwendet werden.

Wenn für ROM und RAM bisher ein gemeinsamer Datenbus-Transceiver bestanden hat, so kann dessen Freigabe nur bei bisheriger Bestückung mit 4164-RAMs beibehalten werden. In Fällen, wo für ROM und einen 16K-RAM-Block (z.B. ältere GENIE-Modelle) ein gemeinsamer Datenbus-Treiber besteht, kann nur diese eine Reihe RAM-Sockel verwendet werden und die Freigabe muß die vollen 64K abzüglich von Adreßbereichen für memory-mapped I/O umfassen. Wenn in solchen Fällen eine Umrüstung auf 4164-RAMs schon durchgeführt worden ist, ist die Freigabe des Treibers bereits den Anforderungen für den Betrieb von 41256-RAMs mit Banking gerecht.

Anschluß des Banking-Boards

Signal	Anschlußpunkt Banking-Board	TRS 80			GENIE I/II			Kontek 1			mc-CP/M-Computer		
		IC	Typ	Pin	IC	Typ	Pin	IC	Typ	Pin	IC	Typ	Pin
D0	g	76	367	11	19	367	6	42	2732	9		Z80	14
D1	f	"	"	13	"	"	10	"	"	10		"	15
D2	e	75	"	13	"	"	4	"	"	11		"	12
D3	d	"	"	5	"	"	13	"	"	13		"	8
D4	c	"	"	7	"	"	11	"	"	14		"	7
A0	8	55	"	17	17	"	9	"	"	8		"	30
A1	9	"	"	13	"	"	7	"	"	7		"	31
A2	1	22	"	11	"	"	5	"	"	6		"	32
A3	2	"	"	13	"	"	3	"	"	5		"	33
A4	10	39	"	7	4	"	9	"	"	4		"	34
A5	4,5	"	"	9	"	"	7	"	"	3		"	35
A6	6,m	"	"	5	"	"	11	"	"	2		"	36
A7	7,n	"	"	11	"	"	5	"	"	1		"	37
A15	h	38	"	9	6	"	9	33	Z80	5		"	5
RESET* (SYSRES*)	a	37	02	1	5	Z80	26 #1	"	"	26 #1		"	26
WR* (OUT*)	l	23	32	3	16	367	11	"	"	22		"	22
IORQ* (OUT*)	k	"	"	"	2	"	9	"	"	20		"	20
CASEN*	b	74	00	10	36	04	2 #2	43'	04	2 #2		32	1 #7
RFSH*	o	40	Z80	28	16	367	3	33	Z80	28		Z80	28
A7'	p	51	157	14	24	157	14 #3	30	157	5 #4		157	13 #8
A15'	q	52	"	6	23	"	6 #5	"	"	6 #5		"	14 #8
A16	r	51'	"	2 #6	24'	"	2 #6	34'	"	2 #6		"	3 #6
A17	s	"	"	3 #6	"	"	3 #6	"	"	3 #6			2 #6
CASEN1*	w	----------			----------			54	32	12	----------		
CASEN2*	v	----------			----------			2	2	1	----------		
CASEN3*	u	----------			----------			----------			----------		
CASEN4*	t	----------			----------			----------			----------		
+5V	j	52	157	16	23	157	16	30	157	16		Z80	11
GND	i	"	"	8	"	"	8	"	2	8		"	29

#1 Anstelle des reinen RESET*-Signals ist eine AND-Verknüpfung von RESET* und NMI* (Z80, Pin17) zweckmäßiger, um bei jedem Booten Bank 0 einzustellen.

#2 Anstelle dieses invertierten 0-16K*-Signals ist es sicherer, dieses Signal über ein OR-Gatter 74LS32 mit MERQ* (GENIE: IC16, 74LS367, Pin 5 / Kontek 1: IC33, Z80, Pin 19) zu verknüpfen.

#3 Das beim Umrüsten auf 4164-RAMs an diesen Pin gelegte Signal ist an Pin 13 von Z24, 74LS157 zu legen. Die dorthin führende Leiterbahn ist zu durchtrennen.

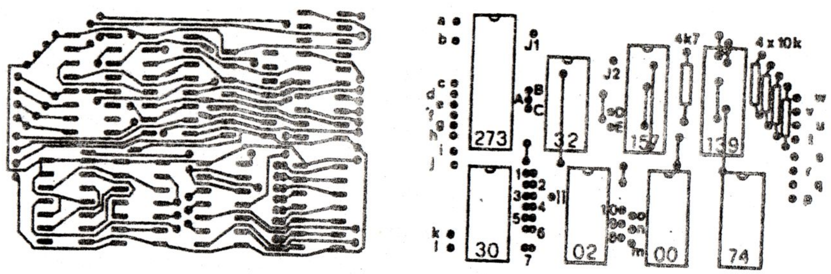
#4 Das beim Umrüsten auf 4164-RAMs an diesen Pin gelegte Signal ist an Pin 3 von IC 34, 74LS157 zu legen. Die dorthin führende Leiterbahn ist zu durchtrennen.

#5 Die an diesen Pin führende Leiterbahn ist zu durchtrennen.

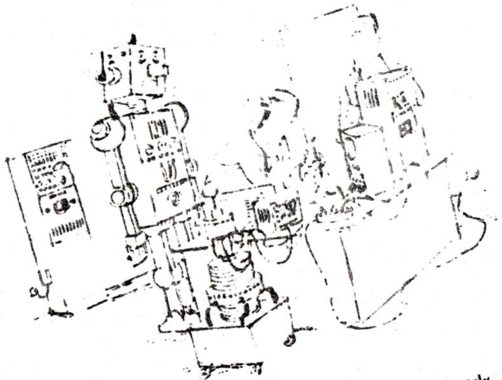
#6 Dieses IC 74LS157 wird mit den Pins 1, 8, 15 und 16 huckepack auf einen anderen 74LS157-Multiplexer aufgelötet. Die Nummerierung bezieht sich auf das IC darunter.

#7 74LS32 auf Höhe des EPROMs

#8 74LS157 beim u1-Kondensator



3) Layout, Bestückung, Drahtbrücken, Jumper und externe Anschlüsse



»Verdammt noch mal, sag nicht immer PAPA zu mir!«

»Ich werd' verrückt: Das Ding sagt: COGITO ERGO SUM!*«
* Ich denke, also bin ich

1M-Byte RAM im GENIE III

Einbauanleitung für das 256K-(1M-)Banking-Board

Wegen nichtz vorhandener Hardware-Dokumentation werden hier für den GENIE III alle wichtigen Informationen zum Umrüstzen auf 256K, 512K oder 1M RAM zusammengestellt.

Das Banking-Board wird dabei in geeigneter Weise unter dem CPU-Board angebracht und die Verbindungen zwischen den Boards werden mit dünnen isolierten Drahtlitzen hergestellt. Bei den meisten Leitungen können die Enden der Litzen direkt auf der Lötseiten an die entsprechenden Pins der ICs gelötet werden.

Verbindungen zu Pins von Huckepack-ICs müssen durch Bohrlöcher, die an geeigneter Stelle im CPU-Board angelegt werden, geführt werden. Wenn man das Board gegen das Licht hält, sieht man freie Stellen, wo auf beiden Seiten keine Leiterbahnen verlaufen.

Es sollte darauf geachtet werden, daß die Drahtverbindungen möglichst kurz sind.

Im einzelnen sind folgende Maßnahmen durchzuführen:

1) Sämtliche 4116-RAMs werden durch 41256-RAMs ersetzt

2) Auf der Bestückungsseite wird die breite unter Z8 (74LS00) hindurchführende Leiterbahn (+12V) durchtrennt.

3) Auf der Lötseite wird die außen an den ROM-Sockeln vorbeiführende und mit "-5V" beschriftete Leiterbahn durchtrennt.

4) Auf der Lötseite wird die Verbindung zwischen Pin16 von Z12 (74LS161) zu Pin9 des RAMs daneben durchtrennt.

5) Auf der Lötseite wird die breite Leiterbahn (neben der mit "-5V" beschrifteten Leiterbahn) zwischen dem Widerstandsarray und dem Kondensator, neben dem auf der Bestückungsseite "D0" steht, durchtrennt.

7) Zwischen Pin8 des obersten linken (bestückungsseitig gesehen) RAMs und dem äußersten linken Pin des Widerstandsarays wird mit einem nicht zu dünnen, isolierten Draht eine Verbindung hergestellt.

8) Im Bereich der RAMs werden alle auf dem Bestückungsplan eingekreisten Kondensatoren ausgelötet.

9) Auf der Bestückungsseite werden die 4 nebeneinander verlaufenden Leiterbahnen von den Pins 4-7 von Z17 (74LS139) zu den Pins 2, 5, 10 und 13 von Z21 (74LS32) durchtrennt.

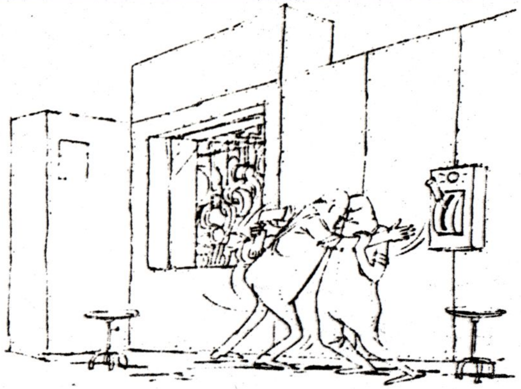
10) Auf der Lötseite wird die zu Pin3 von Z23 (74LS157) führende Leitung (A7) durchtrennt.

11) Auf Z23 (74LS157) wird ein zusätzliches IC 74LS157 mit den Pins 1, 8, 15 und 16 huckepack-aufgelötet (gleiche Orientierung); alle anderen Pins werden waagerecht abgebogen.
An die Pins 2 und 3 des Huckepack-ICs werden A16 und A17 des Banking-Boards geführt. Pin 4 wird über einen 33-Ohm-Widerstand mit den Pins 1 der RAMs (vorher -5V) verbunden.
An Pin5 wird A7' und an Pin6 wird A15' des Banking-Boards geführt. Pin7 wird über einen 33-Ohm-Widerstand mit den Pins 9 der RAMs (vorher +5V) verbunden.

12) Das Signal CASEN* für das Banking-Board wird von Pin1 von Z17 (74LS139) abgenommen.

13) RESET* kann an Pin26 der Z80-CPU abgenommen werden.
Wegen der Nutzung des NMI*-Signals zum Booten des DOS ist es aber zweckmäßig, anstelle des reinen RESET* für das Banking-Board lieber ein SYSRES*-Signal zu erzeugen. Dafür wird auf Z22 (74LS00) ein 74LS08 mit den Pins 7 und 14 huckepack-aufgelötet. Alle anderen Pins werden hochgebogen. Pin1 dieses ICs wird mit Pin17 des Z80 und Pin2 mit Pin26 des Z80 verbunden. An Pin3 wird das Signal SYSRES* abgenommen, das anstelle von RESET* an das Banking-Board geführt wird.

14) Die Signale D0-D5, A0-A7 und A15 werden von den im Auszug aus dem Bestückungsplan bezeichneten Pins der ICs Z25 (74LS245), sowie Z20 und Z16 (74LS244) abgenommen.



Fluch der Automation: »Weg da! Bei der letzten Störung vor vier Wochen hast du gesagt, daß ich den nächsten Handgriff tun darf!«

15) IORQ* wird von Pin3 von Z19 (74LS367), WR* von Pin13 und RFSH* von Pin5 von Z19 (74LS367) abgenommen.

16) +5V und GND können von beliebigen ICs in der Nähe der Stelle, wo das Banking-Board montiert wird, abgenommen werden.

+5V: Z16, 20, 25 und 30 jeweils Pin 20
Z17, 19, 23 und 26 jeweils Pin 16
Z13, 14, 18, 21, 22 und 24 jeweils Pin 14
GND: Z16, 20, 25 und 30 jeweils Pin 10
Z17, 19, 23 und 26 jeweils Pin 8
Z13, 14, 18, 21, 22 und 24 jeweils Pin 7

17) Die Signale CASEN1* bis CASEN4* werden (in dieser Reihenfolge) an die Pins 2, 10, 5 und 13 von Z21 (74LS32) gelegt. Wenn nur 256K RAM realisiert werden sollen, brauchen diese Signale nicht benutzt zu werden. Dann wird nur die linke Reihe RAM-Sockel mit 41256-RAMs bestückt und eine Verbindung zwischen Pin2 von Z21 (74LS32) und Pin1 von Z17 (74LS139) hergestellt.
Wenn 512K RAM gewünscht sind, werden die beiden linken RAM-Reihen bestückt und CASEN1* an Pin2 und CASEN2* an Pin10 von Z21 (74LS32) geführt.

18) Wenn das Banking in den oberen 32K des Z80-Adreßraumes erfolgen soll, muß ein invertiertes A15 an das Banking-Board geführt werden. Dafür kann ein freies NAND-Gatter von Z8 (74LS00) verwendet werden. A15 (von Pin9 von Z20, 74LS244) wird an die Pins 12 und 13 von Z8 gelegt und an Pin11 von Z8 kann das invertierte A15 abgenommen werden.

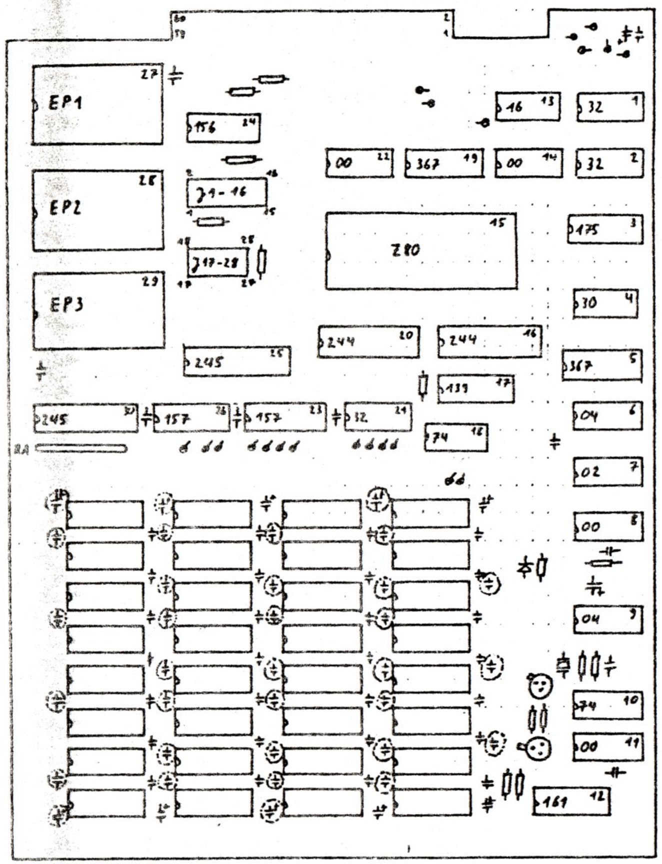
Um das Banking sowohl unter GDOS als auch unter CP/M (und wenn es das mal geben sollte: CP/M+) nutzen zu können, ist es zweckmäßig, zwischen A15 selbst und dem invertierten A15 mit einem Umschalter wählen zu können.

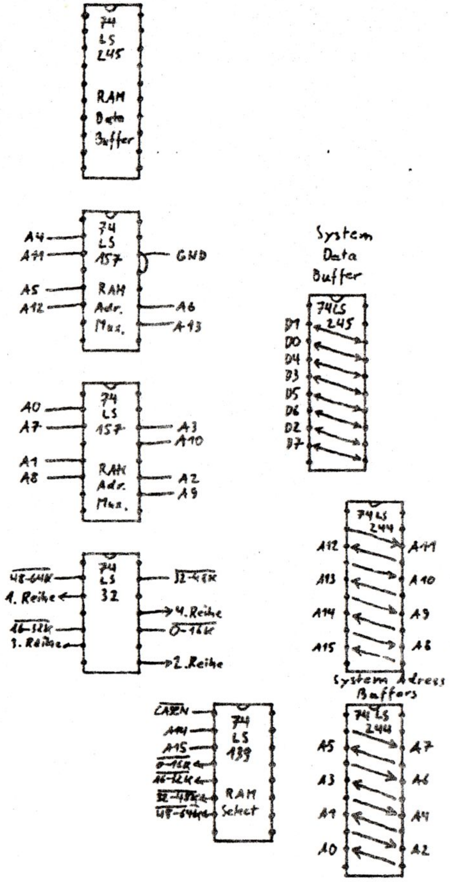
Da im GENIE III die Ports E0H bis FFH belegt sind, muß für das Banking-Board eine andere Portadresse gewählt werden. Geeignet ist die Portadresse DFH; dafür sind dann die Adreßleitungen an folgende Punkte anzuschließen:

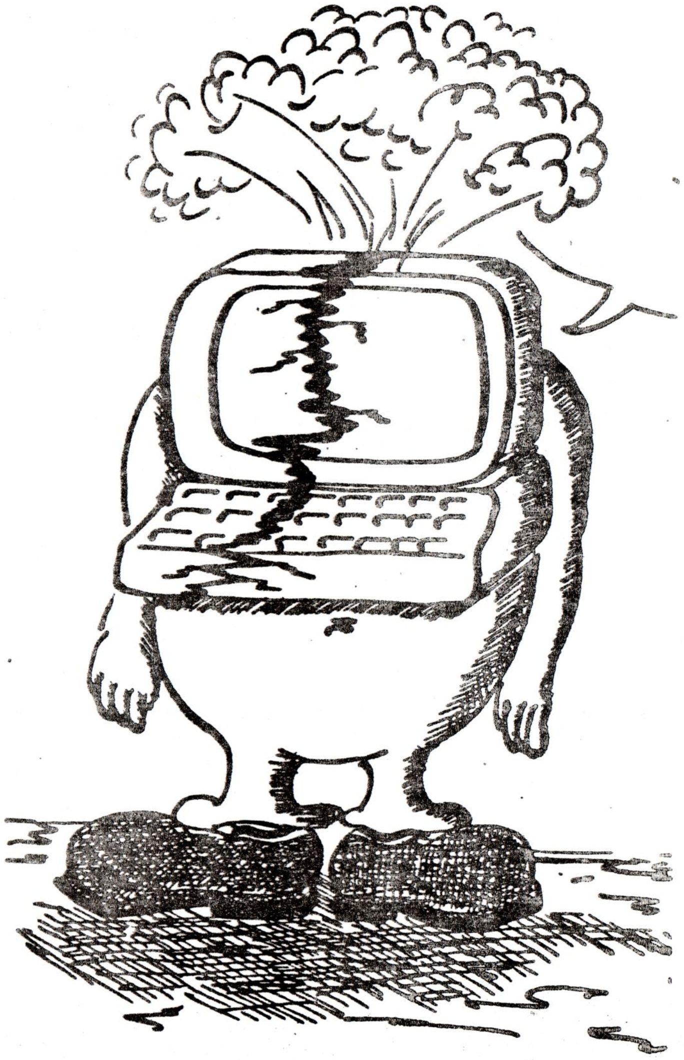
A7	an	7 und n
A6	an	6 und m
A5	an	8
A4	an	5
A3	an	4
A2	an	3
A1	an	2
A0	an	1

Der Punkt 11 wird mit keinem der Punkte 1-7 verbunden. Die Punkte 9 und 10 werden nicht angeschlossen.

Anmerkung
Die Einbauanleitung wurde aus einer Untersuchung des CPU-Boards eines GENIE III abgeleitet und noch nicht durchgeführt. Wegen der Analogie zum GENIE I ist aber nicht zu erwarten, daß die Erweiterung versagt.







Bitte beachtet dazu auch die
Seiten 69 - 75
dieses INFO's

CLUB 80 - ECB - BUS - SYSTEM

Im letzten INFO wurde die Idee geboren den TRS-80 als Selbstbau in ECB-Bus-Technik aufzubauen. Daraufhin meldeten sich bis jetzt schon sieben Mitglieder, die an diesem Projekt interessiert sind.

Wir möchten nun in dieser INFO ein Konzept vorstellen, daß mit seinen Möglichkeiten sicher viele von Euch anspricht.

ECB-Bus ---- Bus Erweiterung

In Verbindung mit dem aus der INFO schon länger bekannten ECB-Bus-Adapter und dem Aufbau des Grundbausteins können alle vorhandenen, sowie die für später geplanten, Zusatzplatinen an den schon vorhandenen Computer angeschlossen werden.
Der Grundbaustein besteht aus folgenden Teilgruppen:

--- 19 Zoll-Gehäuse
--- ECB-Bus-Platine, 96-polig mit 10 Karten-Steckplätzen
--- Stromversorgung, +12V/2A...-12V/0.4A...+5V/4A

ECB-Bus ---- Selbstbau TRS 80

Als weiteres besteht die Möglichkeit einen kompletten Rechner in ECB-Bus-Technik aufzubauen. Dazu ist auch der Aufbau des oben beschriebenen Grundbausteins erforderlich. Hinzu kommen noch 4 Platinen (der Computer).

SB1T80 ... CPU-Karte mit 64K-RAM, CP/M-fähig, doppelter Takt Bankinglogig
SB2T80 ... 80-Zeichenkarte mit 6845 und HRG 384 * 192 (RB-Kompatibel) (SoftwareKompatibel zum NEWDOS und ähnliche)
SB3T80 ... Tastatur-Karte mit Druckerausgabeadresse für Tandy
SB4T80 ... DD - Floppykontrolerkarte mit Datenseperator

Die Karten SB3T80 und SB4T80 sind schon lauffähig. Die beiden anderen Karten sind noch in Entwicklung.

ECB-Bus ---- Zusatzplatinen

Die Zusatzplatinen passen für beide Bautypen unseres Projektes. Sie werden im einzelnen in den kommenden INFO's vorgestellt. Als lauffähige Karten sind in der Sammlung folgende Zusatzplatinen zum Nachbauen:

ZK01 ... Druckerkarte für Druckausgabe Video-Genie
ZK02 ... Universelle I/O-Karte, auf jede der 256 möglichen Adressen einstellbar
ZK03 ... Epromprogrammierkarte
ZK04 ... RS 232
ZK05 ... 2k zusätzliche RAM/Eprom

EK01 ... Meßwertaufnahme (Erweiterungskarte zu ZK02)
EK02 ... Laststeuerung (Erweiterungskarte zu ZK02)

Für später sind:
... Soundkarte mit 6581 Soundchip, ... Farbgrafikkarte 256 * 192,
... Digitalisierer, ... Plotter, ... Messen/Steuern/Regeln
als Zusatzkarten geplant.

Auch die Karten aus der c't sind verwendbar. Sicher habt Ihr selbst auch Vorschläge für eine Zusatzkarte. Bitte meldet Euch dazu!!!

ECB-Bus ---- Allgemeines

Unser Ziel ist es, erst einmal für alle Interessierten den Grundbaustein aufzubauen. Damit haben schon einmal alle die Möglichkeit die Zusatzkarten zu nutzen (den Bus auszuprobieren). Wie oben schon angeführt benötigt man dazu noch die ECB-Bus-Adapterkarte und den schon vorhandenen Computer.
Anschliesend wird dann der ECB-Bus-Computer aufgebaut. Dies ist der 2. Bauabschnitt unserer Aktion. So kann sich jeder selbst entscheiden, wie weit er mitbauen möchte.
Wir bieten in jedem Fall beim Aufbau und der eventuellen Fehlersuche unsere Unterstützung an.
Desweiteren haben wir uns überlegt, Euch mehrere "Aktionsstufen" anzubieten. Dadurch hat jeder von Euch die Möglichkeit an dieses System zu kommen.
Wir bieten also an:

--- Kompletter Selbstaufbau in eigener Regie
--- Kompletter Selbstaufbau in einer Gruppe (zu zweit oder zu dritt) bzw. mit teilweiser Unterstützung.
--- nur mechanischer Aufbau. Die Platinen werden von uns mitaufgebaut und geprüft es bleibt nur noch der Zusammenbau.
--- übernahme eines fertigen Gerätes (Ausstattung nach Wunsch)

ECB-Bus ---- Der Bus

Das wichtigste an diesem System ist der Bus. Wir haben den ECB-Bus aus der c't übernommen und auf unsere Belange angepasst. Da schon einige Bus-Typen vorhanden sind haben wir mit Rücksicht auf die Kompatiblität folgende Belegung gewählt.

	a		b		c
+5V	1	+5V	1	+5V	1
D5	2	MRD -	2	D0	2
D6	3	MWR -	3	D7	3
D3	4	IN -	4	D2	4
D4	5	OUT -	5	A0	5
A2	6	reserviert CP/M	6	A3	6
A4	7	" "	7	A1	7
A5	8	" "	8	A8	8
A6	9	" "	9	A7	9
WAIT -	10		10		10
BUSRQ -	11		11	IEI	11
A18	12		12	A19	12
+12V	13		13	-12V	13
	14		14	D1	14
-5V	15		15	-15V	15
2 PHI	16		16	IEO	16
A17	17		17	A11	17
A14	18		18	A10	18
+15V	19		19	A16	19
M1 -	20		20	NMI -	20
	21		21	INT -	21
	22		22	WR -	22
BAI	23		23	GDT	23
VCMOS	24		24	RD -	24
BAO	25		25	HALT -	25
	26		26	PWRCL -	26
IORQ -	27		27	A12	27
RFSH -	28		28	A15	28
A13	29		29	PHI	29
A9	30		30	MREQ -	30
BUSAK -	31		31	RESET -	31
GND	32	GND	32	GND	32

Die noch nicht bezeichneten Stifte sind frei bzw. aus kompatiblitäts Gründen freigehalten.
Die Signale CAS und RAS fehlen wegen bevorzugtem CMOS-RAM Einsatz bzw. Refresch-Kontroler auf der RAM-Karte.

ECB-Bus ---- Kosten

Wie schon angeführt ist der erste Schritt, das Aufbauen des Grundbausteins, für alle gleich, egal ob Ihr Euch dann für die Bus-Erweiterung oder den Selbstbau-TRS-80 entscheidet.
Aus diesem Grund empfiehlt sich eine Sammelbestellung. Ein weiterer Vorteil der Sammelbestellung wäre der kostengünstigere Einkauf durch den Mengenrabatt, der bei entsprechenden Stückzahlen gewährt wird. Wir hoffen mindestens die magische Zahl 10 zu erreichen. Dazu sollten wir allerdings ein paar Leute mehr sein. Vielleicht bekommen wir noch weitere Interessierte für unser Projekt, wenn sich auch Nichtmitglieder daran beteiligen. Fragt bitte in Eurem Freundes- und Bekanntenkreis nach weiteren Mitbauern. Es lohnt sich für Euch auf jeden Fall, denn dadurch könnt Ihr Euren Geldbeutel schonen.
Nun endlich zu dem Preis unseres Projektes. Für die Grundeinheit haben wir einen Betrag von DM 250,-- als oberste Grenze errechnet. Für diesen Preis bekommt Ihr dann das 19 Zoll-Gehäuse, die Bus-Platine, 10 Stück 96-polige Federleisten und die Stromversorgung. Des weiteren haben wir für die vier Computerplatinen (SB1T80 - SB2T80) eine Preisvorstellung von ca. DM 550,--. Da die Halbleiterpreise ständigen Schwankungen unterlegen sind, möchten wir uns hier noch nicht festlegen. Wir haben aber die Hoffnung, daß der Preisverfall weiterhin anhält und wir unter die Preisvorstellung kommen werden. Wir werden den Computerteil sowieso kartenweise aufbauen, sodaß der Geldbetrag nicht aufeinmal fällig wird sondern nach und nach mit Vorstellung der einzelnen Karte in der INFO.
Noch eine Anmerkung zum Preis - liegen wir unter unserem Preisvorschlag, bekommt Ihr Euer Restgeld wieder ausbezahlt oder Ihr könnt es stehen lassen für die nächste Sammelbestellung. Natürlich erwarte ich im Gegenzug von Euch, daß Ihr mir bei eventueller überschreitung unserer Preisvorstellungen den dann noch fehlenden Betrag auf meinem Konto wieder gutschreibt. Ist ja Ehrensache. Wir werden aber die Preise so wählen, daß dies nicht der Fall sein wird.

ECB-Bus ---- Bestellung

Um am kostengünstigsten zu arbeiten, bitten wir alle Interessenten sich mit Hilfe des im INFO beiliegenden Antwortbogens anzumelden.
Anmeldeschluß ist der 20. August 1986.
Bis zu diesem Termin möchte ich alle Anmeldungen zusammenhaben.
Anmeldungen sind nur gültig, wenn bis zum Anmeldeschluß auch der entsprechende Betrag auf meinem Postgirokonto eingegangen ist.
Ich werde dann die Sammelbestellung der Bauteile in die Wege leiten, sodaß bis zum nächsten INFO -dort erscheint dann die Aufbauanleitung- die Bauteile bei Euch ankommen.

Das Anmeldeformular findet Ihr am Ende des INFO's.

ECB-Bus ---- Kontaktpersonen

Zum Abschluß möchte ich Euch nun noch die Kontaktadressen bekanntgeben. Wenn Ihr noch weitere Informationen benötigt, meldet Euch bitte bei uns. Die vollständige Adresse findet Ihr in der Adressenliste.
Ulrich Böckling, Bernd Drohwälder, Eckehard Kuhn und Jens Neueder.

Jens Neueder

ARD und ZDF im Spiegel der Zuschauer

Drei Buchstaben, die jeden Abend pünktlich um 20 Uhr Millionen Bundesbürger in die „gute Stube" flimmern, reizen offensichtlich ungemein die Phantasie der Betrachter: „Aktion Ruhendes Deutschland" oder „Alpträume Rühren Daher" deuteten Fernsehzuschauer das Kürzel ARD in einem

ARD: Anschalten, Reingucken, Draufhauen

Ideenwettbewerb, zu dem das Ulk-Magazin „mad" aufgerufen hatte. „Allgemeines Ratloses Durcheinander", „Anbieter Reizloser Dumpfkost", „Armutszeugnis Resignierter Dünnbrettbohrer" und „Alles Reichlich Daneben", hieß es da in den Zuschriften.

Noch drastischer antworteten die Leser auf die Frage, was für ein Programm die ARD biete: „Abführmittel, Rezeptfrei Dargeboten" textete ein Teilnehmer der Aktion bissig. „Abfall, Ramsch, Dutzendware", „Altmüll, Richtig Durchgequirlt", „Alter, Ranziger Durchschnittskäse" oder „Achtung, Reine Dosenkost", höhnten andere Leser.
Die Frage, was ARD bei den Zuschauern bewirke, reizte eine Teilnehmerin zur Aufforderung: „Anschalten, Reingucken, Draufhauen!" Weniger gewalttätig lauteten andere Zuschriften: „Alles Rennt Davon", „Anschauen Reduziert Denkvermögen" oder „Alle Rufen Dallas!"
Aber auch den Kollegen vom ZDF sollte es nicht besser ergehen. Ihr Sender wurde mit nicht weniger kernigen Deutungen bedacht. Darunter waren „Zankapfel Der Familie", „Zerstörung Deines Feierabends", „Zwangsdroge Der Frührentner", „Zweites Deutsches Fiasko", „Zimmerfang Durch Flimmerzwang", „Zimmermanns Detektiv-Fortbildung" und „Zensur Durch Flachmänner". Bei dem Ideenwettbewerb gingen natürlich auch einige positiv klingende Deutungen ein. Die waren aber nicht so originell.

BöRSE . . . BöRSE . . . BöRSE . . . BöRSE

ANGEBOTE GENIE16

Genie 16 C statt 3495,-DM nur 3195,-DM
Genie 16 TC statt 4495,-DM nur 4095,-DM
Genie 16 XC statt 5995,-DM nur 5495,-DM
Genie 16 TXC statt 6495,-DM nur 5895,-DM

Info anfordern bei:

Peter Spieß
Offsetdruck + EDV-Zubehör
Trugenhofenerstr. 27
8859 Rennertshofen 1
Tel: 08434/454

Floppy-Laufwerk

EPSON SMD-100

zu verkaufen. 40 Tks., SS, DD, 3,5" Shugart-Bus, direkt an Controller Genie 1, 2, TRS80 M.1, M.3 ansteckbar; incl. 4 Disketten (NEWDOS auf Disk) und Manual. NAGELNEU !
Preis: einhundertsechzig Mark (VB)

256K RAM, gebanktes DOS

RAM-Speicher satt, alle DOS-Overlays im Speicher, kein Nachladen von Diskette mehr, dazu noch zusätzliche 128K RAM für Anwenderdaten

fertig bestückte Platine zum Selbsteinbau ohne 41256-RAMs mit Software zum Selbstkostenpreis von 50,-DM; unbestückte, ungebohrte Platine mit Software für 15,-DM

Bitte Diskette mit Lieblings-DOS und 3,-DM in Briefmarken für Rückporto beilegen.

Helmut Bernhardt, Hafenstr.7, 2305 Heikendorf

Wer kann mir weiterhelfen !

So richtig angefangen hat es mit meinen Schwierigkeiten, als ich versucht habe, einen groesseren File (34K) mit dem Editor des FORTRAN-Packetes von Tandy weiterzubearbeiten. Zuerst einmal fiel mir auf, dass der Editor von sich aus Page-Marken in den File eingebaut hatte. Grund: Da er mit der Zeilen-Nummer 100 beginnt und in 100ter-Schritten weiterzaehlt, war es zu einem Zeilennummernueberlauf gekommen. In diesem Fall beginnt er eine neue Seite, wiederrum mit 100 beginnend, bis zum naechsten Ueberlauf. So weit so gut, waere da nicht im Verlauf meines weiteren tuns geaendert, umnummeriert, geloescht und nach irgendwelchen Strings gesucht worden. Da jeder Abend auch einmal ein Ende hat, so kommt auch das Retten des eigenen Werkes. Zu guter letzt, mal sehen ob soweit alles stimmt, schnell noch eine Probeuebersetzung mit M80. Die Welt waere um einiges schoener, gaebe es Murphy nicht. Jede Menge Fehlermeldungen ueber fehlende Labeldefinitionen. Das aber konnte nicht sein, denn die waren nachweislich zuvor noch gesichtet worden. Also alles zurueck, E80 wieder aufrufen und den File nochmals geladen. Und tatsaechlich, die Label PEEKW, PEEKB, POKEW und POKEB - um diese handelte es sich naemlich - waren verschwunden, samt dem zugehoerigen Assemblertext !?!. Um es noch genauer zu sagen, eine Seite Text = 58 Zeilen fehlten.

Um dem Problem naeher zu kommen, habe ich dann folgendes getan: Den gesammten Speicher mittels CLEAR geloescht und danach E80 und den File erneut geladen. Dammit die Sache in Schwung kommt anschliessend neu durchnummeriert und ein wenig nach einem String suchen lassen. Zwischen jedem dieser Schritte erfolgte jeweils ueber DEBUG eine Speicherkontrolle (deswegen das CLEAR). Dabei ist mir dann aufgefallen, dass mit jeder Aktion E80 den freien RAM-Speicher weiter vollgeschrieben hat (???). Da der Stack am oberen Speicherende liegt,musten sich Text und Stack dann unweigerlich einmal begegnen. Sofern man es nicht zu sehr uebertreibt wird scheinbar nur der Text bzw. die angelegten Vermerke ueber Aenderungen und der Gleichen veraendert. Man kann es auch so weit treiben, dass der Editer regelgerecht abstuert (so dann im weiteren geschehen). Warum der Indexfile nicht angelegt wird, laesst sich hiermit noch nicht erklaeren, denn dieser fehlt auch dann, wenn man den File sofort nach dem Laden wieder abspeichert. An dieser Stelle muss ich noch dazu sagen, dass dieser Fehler nicht generell auftritt, was die Sache nur Unverstaendlicher macht, leider.

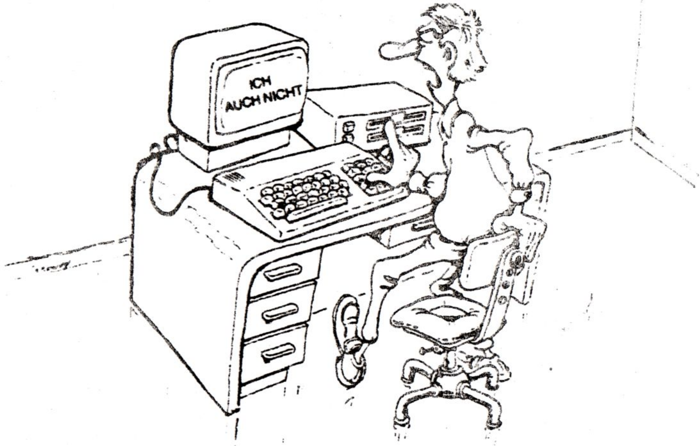
Und nun meiner Frage an die Mitglieder : Wem ist dieser Fehler bekannt und kennt den Zap oder andere Massnahmen ?. In diesem Zusammenhang noch einen Zusatz. Ich habe ausserdem einen E85 (= E80), bei dem allerdings der verfuegbare Textspeicher kleiner ist (um ca. 5K). Da ich nur eine defekte und nicht lauffaehige Version besitze, konnte ich dem nicht weiter nachgehen. Eventuell ist bei dieser Variante der Fehler beseitigt. Kann hierzu ebenfalls jemand etwas sagen ?.

Mein zweites Problem betrifft den M80-Assembler. Als ich versuchte, mit der INCLUDE-Funktion zu arbeiten weigerte er sich, diese zu kennen. Und dass, obwohl er das laut Handbuch muesste. Hiervon ebenfalls betroffen sind die Alias-Namen $INCLUDE und MACLIB. Hat Microsoft bei der Anpassung fuer das Model-I diese moeglicherweise heimlich unterschlagen ?. Es ist mir jedenfalls nicht gelungen, den M80 zu ueberreden. Moeglicherweise gibt es einen anderen Befehl, der das Gleiche bewerkstelligt. Wem ist hierzu etwas bekannt. Gibt es womoeglich auch Versionen, denen diese Befehle noch nicht abhanden gekommen sind oder gilt dies generell ?. Wenn dem so ist, waere das schade, aber nicht zu aendern.

Wer mich zu diesen Themen anrufen will, die Nummer steht ganz hinten im Info unter Mueller, K., ich bin aber erst nach 19.00h zu erreichen. Schriftlich ist's womoeglich billiger und man kann sich dann hinterher alles nochmals in Ruhe durchlesen (Schrecken oder Euphorie und so !).

Das duerfte ersteinmal genug sein. Bis zum naechsten Info habe ich vielleicht das naechste Packet an 'Dahlschlaegen' 'an Land' gezogen. Bis dahin Tschuess

Kurt Müller



Paß auf! Ich laß mir heute morgen von dir keine Frechheiten bieten.

Buchbesprechung

Immer wieder habe ich die traurige Erfahrung gemacht, daß der Preis eines Buches leider nur sehr wenig über seine Qualität aussagt. Dies ist vor allem auch und nicht zuletzt bei der Spezies der Computerbücher so! So stehen in meinem Bücherschrank eine ganze Menge sehr teurer aber wertloser Schinken herum und sicher geht es anderen Computerfreaks nicht viel anders! Informationsaustausch tut hier Not! Schließlich brauchen nicht mehrere Clubmitglieder mit teurem Geld die gleiche Erfahrung zu bezahlen, nämlich daß dieses oder jenes Buch nichts taugt. Aus diesem Grunde habe ich mir vorgenommen, in lockerer Folge einige neuere Publikationen (die alten hat ja sowieso fast jeder!) vorzustellen und meine Meinung zu ihnen zu äußern. Sehr freuen würde ich mich, wenn noch weitere Mitglieder sich zu Veröffentlichungen, egal ob positiv oder negativ, auslassen würden. Sicher kann so manche müde Mark gespart und/oder besser angelegt werden!

CP/M Bücher

Immer mehr unserer Mitlieder können auf ihren Maschinen (Model 1/3 mit entspr. Hardwareänderung und Model 4/4P sowieso) das Betriebssystem CP/M fahren. CP/M ist aber nicht zu vergleichen mit den sonstigen Diskettenbetriebssystemen, die es für den TRS-80 gibt und so tun sich die Benutzer von TRS-, NewDOS usw. relativ schwer damit. Natürlich gibt es auch über diesen Bereich der Computerei genügend Literatur, aber welches Buch soll man sich anschaffen?

CP/M kompakt (Plate, Franzis-Verlag)
Noch nicht mal 10,00 (in Worten "zehn"!) Mark kostet ein Kurzhandbuch für CP/M, das Büchlein "CP/M kompakt"! Trotzdem enthält es alle für den CP/M-Benutzer wichtigen Informationen. Angefangen bei den Control-Sequenzen (z.B.: CTRL P, CTRL S usw.) über die resistenten (DIR, USER, usw.) bis zu den transienten (PIP, STAT usw.) Befehlen ist hier alles fein säuberlich und sehr übersichtlich aufgelistet. Im zweiten Teil des nur 87 Seiten starken, im Taschenbuchformat gehaltenen Büchleins werden alle BIOS- und BDOS-Aufrufe mit genauer Angabe der zu übergebenden Parameter angegeben, sodaß auch Assembler-Programmierer von der Publikation profitieren können. Zu empfehlen ist "CP/M kompakt" allen CP/M-Usern, die sich einen schnellen überblick über die verfügbaren CP/M-Befehle verschaffen wollen, ohne gleich einen ganzen Berg von Handbüchern durchzuwälzen. Zum gründlichen Einarbeiten in das "meistverbreitete Betriebssystem für Microcomputer" ist das Buch aber nicht geeignet!

Dazu gibt es andere Literatur. Z.B.:

Betriebssystem CP/M (Plate, Franzis-Verlag),
dieses Buch kostet immerhin schon fast 50,-- DM. Es behandelt CP/M von Grund auf (es werden erst einmal die Begriffe "Monitor" und "Betriebssystem" erklärt) und enthält auch Musterprogramme (BIOS, BDOS). Es wendet sich mehr an den Programmierer (Assembler) als an den Anwender, auch wenn im letzten Drittel des Buches die CP/M-Befehle kurz erklärt sind. Leider ist das Buch zu sehr auf den mc-CP/M- Computer zugeschnitten und bringt dem Tandy-Benutzer kaum etwas, ist also kaum empfehlenswert!

CP/M Handbuch (R. Zaks, SYBEX),
auch ein Buch in der Preislage eines halben Hunderters, ist da schon besser geeignet einen CP/M-Neuling in die Bedienung desselben einzuführen. Nach einer allgemeinen Einführung in die Grundlagen (von "was ist eine Floppy und wie wird sie in den Drive eingelegt" bis "Organisation von Dateien" ist da alles zu finden) werden die CP/M-Befehle ausführlich erklärt. Dieser Teil umfaßt ca. 60% des Buches, die verbleibenden Seiten bringen alle Befehle nochmals in einer Kurzbeschreibung, sodaß das Buch auch als Nachschlagewerk dienen kann! Im großen und ganzen ist das Buch sehr brauchbar und empfehlenswert, was bei diesem renommierten Autor und dem Verlag auch kaum anders zu erwarten war! Aber auch hier müssen Abstriche gemacht werden, da sich das Buch praktisch nur an den Anwender wendet, den Programmierer aber praktisch ganz vergißt.

CP/M - Anatomie eines Betriebssystems (J.D. Dennon, M u. T), ist zwar die teuerste, dieses Buch kostet immerhin beinahe 70,-- DM, aber auch die beste der hier vorgestellen Publikationen über CP/M. Es ist praktisch als Lehrgang aufgebaut und führt den Leser systematisch in den Umgang mit CP/M. Dabei werden im ersten Teil nicht nur die Befehle angesprochen, sondern auch die Programmiersprachen M- und C-BASIC kurz gestreift. Danach wird ausführlich auf den Assembler eingegangen und dabei CP/M systematisch auseinandergenommen. Das geht so weit, daß am Schluß ein Programm zur Rettung gelöschter Dateien geschrieben werden kann. Das Buch ist sehr unterhaltsam und kurzweilig geschrieben und immer wieder mit vielen Beispielen und Fragen zur Selbstkontrolle gespickt. Leider fehlt dafür ein Zusammenfassungsteil, so daß man das Buch praktisch nicht als Nachschlagewerk benutzen kann. Ein "Einstieg in der Mitte" ist jedoch durchaus möglich, so daß der nicht an BASIC und sonstigem interessierte Assemblerprogrammierer ohne weiteres ein paar Kapitel auslassen kann.

Meiner Meinung nach ist eine Kombination der Bücher "CP/M kompakt" und "CP/M - Anatomie eines Betriebssystems" genau die richtige Mischung, sowohl für CP/M-Neulinge als auch für Leute, die tiefer ins System einsteigen wollen! Euer

Hartmut Obermann

===> Unsere Programm-Diskothek <===

HALLO BOYS !

(Girls habe ich im Mitgliederverzeichnis nicht entdecken können. Sollte sich dennoch eins unter einem männlichen Pseudonym darin verstecken, so sei sie hiermit extra gegrüßt!)

Um 14 Programme wurde unsere Diskothek in diesem Monat bereichert! Den fleißigen Abtippern und Einsendern sei auch hier noch einmal herzlich gedankt! Und selbstverständlich sollen sie auch extra aufgeführt werden.
Es waren dies:
Hans-Martin Stephan (8 Programme)
Josef Konrad (4 Programme)
Klaus Hermann (1 Programmpaket)

Welche Programme das sind, seht Ihr in nachstehender Liste.
Ich habe deren Aufmachung wieder etwas verbessert. Wenn Ihr nichts einwendet, wird auch der nächste Programm-Katalog in dieser Form erscheinen (und, wie ich hoffe, bald!)
Allerdings dann nach Programm-Gattungen (Utilities, Spiele usw.) geordnet und innerhalb derselben alphabetisch nach Programm-Namen.
Etwas Schwierigkeiten bereitet mir immer die sog. "Benotung" der Programme. Es ist fast unmöglich, hierbei gerecht zu sein. Da ich der Gerechtigkeit sowieso nicht zum Siege verhelfen kann, werde ich ihre Truppen wohl in Zukunft in der Kaserne lassen und damit eine Pioniertat für die allg. Abrüstung tun. Sagt mir, ob Ihr damit einverstanden seid. (Schweigen gilt als Zustimmung!!)

Wenn Ihr gestattet, noch eine Anmerkung in eigener Sache: Meine Programme "LIESDIR" und "LIESKOMM", die ich im letzten Heft veröffentlichte, enthielten einige Bugs. Wer es bemerkt und mir trotzdem nicht mitgeteilt hat, den muß ich unangemessener Zurückhaltung "zeihen"!
Ich habe beide Programme inzwischen nicht nur korrigiert, sondern auch erweitert, so daß man File-Kommentare jetzt bequem speichern und auch wieder ändern kann. So habt Ihr hiermit ein Werkzeug zur Kommentierung eines Disketten-Inhaltes auf einem Blatt Papier, das man mit in die Diskettenhülle stecken kann, so daß man nicht erst eine "Datenbank" umständlich befragen muß. Ich mache es seit langem so und finde es sehr praktisch...

MIT "SUUM GEQUIEKE" (LAT.) UND "GREETINGS A LOT" (ENGL.) VERBLEIB' ICH WIE IMMER "DER DISKO-KA-JOT" !


```
CREF/CMD      UTL ProgrNr 0335 Teile: 06 Note 0 Eing. 6 /7 /86
========          f.Comp                    Quelle:
Verfasser KONRAD, Josef            eingetippt von K., J.
Cross-Reference-Programm fuer 64- sowie 80-Zeichen-Ausgabe
mit ALCOR-PASCAL-Source: auch mit Ausblenden von PASCAL-
Kommentaren
                                                Disk-Nr. 9

EXT/CMD       UTL ProgrNr 0336 Teile: 02 Note 1 Eing. 6 /7 /86
=======           f.Comp alle               Quelle:
Verfasser KONRAD, Josef            eingetippt von K., J.
zum Aendern der Extension von Files.
Aufruf: EXT,#,ALT,NEU      (# = Laufwerk-Nr.)
                                                Disk-Nr. 9

FORMDRUK/CMD UTL ProgrNr 0337 Teile: 02 Note 2 Eing. 6 /7 /86
============      f.Comp                    Quelle:
Verfasser KONRAD, Josef            eingetippt von K., J.
Formatiertes Drucken von ASCII-Files (mit ALCOR-Source).

                                                Disk-Nr. 9

PASFORMK/CMD UTL ProgrNr 0338 Teile: 04 Note 1 Eing. 6 /7 /86
============      f.Comp                    Quelle:
Verfasser KONRAD, Josef            eingetippt von K., J.
Zur Umwandlung in Kleinbuchstaben (mit Source-Code).

                                                Disk-Nr. 10

CASLABEL/BAS UTL ProgrNr 0339 Teile: 01 Note 1 Eing. 6 /16/86
============      f.Comp                    Quelle: 80 MICRO
Verfasser                          eingetippt von
Druckt Etiketten fuer Kassetten
(Einsender: Hans-Martin STEPHAN)
                                                Disk-Nr. 10

FRAME/CMD     UTL ProgrNr 0344 Teile: 05 Note 2 Eing. 6 /21/86
============      f.Comp                    Quelle: 80MICRO 9/83
Verfasser BOWKER & KaJot Mb        eingetippt von KaJot Mueh-tein
BASIC-Erweiterung durch einen neuen Befehl, z.B. "FRAME".
Dieser zeichnet einen Rahmen (FRAME) um den Bildschirm.
Source-Code, DO-File u. zwei BASIC-Beispiele    Disk-Nr. 10


SVC/BAS       UTL ProgrNr 0320 Teile: 03 Note 3 Eing. 6 /23/86
============      f.Comp Model 4            Quelle: 80MICRO 4/86
Verfasser H.Brothers               eingetippt von Klaus Hermann
Das Programm ruft SVC (SuperVisorCall) aus einem BASIC-
Program auf. Hierzu liegt ein DEMO-Programm vor.
Informationen hierzu s.INFO Juli 86 (& File)    Disk-Nr. 10
```

```
biorhyth/bas SPL ProgrNr 0340 Teile: 01 Note 1 Eing. 6 /16/86
============       f.Comp                   Quelle: SIEHE UNTEN
Verfasser Stuebs                 eingetippt von Hans-M.Stephan
Die ziemlich bekannte (Pseudo-)Weissagung mit "Stimmungskur-
ven" fuer Geist, Gemuet und Koerper...
Aus: Anwenderprogr.f.TRS80 u.GENIE.-Hofacker    Disk-Nr. 6

HALLO/BAS    SPL ProgrNr 0341 Teile: 01 Note 1 Eing. 6 /16/86
=========          f.Comp                   Quelle: S.UNTEN
Verfasser                        eingetippt von Hans-M. Stephan
Der Computer stellt dumme Fragen und gibt seine Kommentare
zu den schlauen Antworten...
(aus: BASIC-Computer-Spiele, Bd.1 (SYBEX)       Disk-Nr. 10

REDBARON/BAS SPL ProgrNr 0345 Teile: 01 Note 2 Eing. 6 /16/86
============       f.Comp                   Quelle: S.UNTEN
Verfasser STUEBS                 eingetippt von Hans-M.Stephan
Bei diesem Spiel muss man aufpassen, dass dem Kameraden
neben einem nichts passiert.
Aus: Anwender-Programme f.TRS80 und GENIE       Disk-Nr. 10

MONDLNDG/BAS SPL ProgrNr 0346 Teile: 01 Note 1 Eing. 6 /16/86
============       f.Comp                   Quelle: MC 8/83
Verfasser                        eingetippt von Hans-M.Stephan
Man muss den Treibstoff-Vorrat richtig einteilen, um bei der
Landung nicht zu zerschellen!
                                                Disk-Nr. 10

AUTOKOST/BAS KFM ProgrNr 0347 Teile: 01 Note 3 Eing. 6 /16/86
============       f.Comp                   Quelle: C'T 3/84
Verfasser                        eingetippt von Hans-M.Stephan
Laufende detaillierte Erfassung und Ueberwachung der
Kfz-Kosten (mit Datenbank, versteht sich).
                                                Disk-Nr. 10

ENTSCHDG/BAS MTH ProgrNr 0343 Teile: 01 Note 4 Eing. 6 /16/86
============       f.Comp                   Quelle: SIEHE UNTEN
Verfasser STUEBS                 eingetippt von Hans-M.Stephan
Ein gutes Optimierungs-Programm im Frage-und-Antwort-Stil,
so dass man nicht merkt, dass Mathe (LIN.OPTIMIERUNG) dahin-
ter steckt.-Aus "Anw.Progr.f.TRS80 u.GENIE"     Disk-Nr. 10

BOOKINV/BAS  DAT ProgrNr 0342 Teile: 01 Note 3 Eing. 6 /16/86
===========        f.Comp                   Quelle: SIEHE UNTEN
Verfasser                        eingetippt von
Der Name sagt es:  Buechnerverzeichnis.
Aus: 80MICRO Magazine, Nov.1982: eingesandt von H.M.Stephan
                                                Disk-Nr. 10
```


- I M P R E S S U M -

1. Vorsitzende: **Peter STEVENS**
Postfach 56
4600 Dortmund 1
☎ 0231 /593883

2. Vorsitzende: **Hartmut OBERMANN**
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

Hardwarekoordinator: **Eckehard KUHN**
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar: **Klaus-Jürgen MÜHLENBEIN**
Am Mönchgarten 28
6940 Weinheim -Lü.
☎ 06201 /55052

Redaktion: **Jens NEUEDER**
Panoramastraße 21
7170 Michelbach /Bilz
☎ 0791 /42877

Redakteure dieser Ausgabe:
Helmut Bernhardt ¥ Klaus Hermann
Klaus-J. Mühlenbein ¥ Kurt Müller
Jens Neueder ¥ Hartmut Obermann
Gerald Schröder ¥ Arnulf Sopp
Günther Wagner ¥

Bankverbindung des CLUB 80
Postgirokonto: Peter STEVENS, Sonderkonto CLUB 80
Konto-Nummer: 285 491 - 465
Postgiroamt: Dortmund
BLZ: 440 100 46

HEFT 14
Juli 1986

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

wieder ist ein neues INFO fertiggestellt. Ganz besonders freut es mich, daß wir in diesem INFO keine Zeitungsartikel haben, und trotzdem unser bisher dickstes INFO ausgeben. Hierfür meinen Dank an unsere fleißigen Autoren.

Auch möchte ich mich für die guten Kritiken bedanken, die ich für meine Redaktionsarbeit bekam. Wie ich mit einem kurzen Blick in die Fragebögen feststellte. Leider war es mir nicht möglich, eine ausreichende Auswertung vorzunehmen, da die Mehrzahl der Fragebögen noch fehlt. Ich hoffe, daß sie bis zur nächsten INFO noch eintrudeln. Bemängelt wurde die unleserliche Kontaktadresse. Dies werde ich mit Erscheinen der nächsten INFO abgestellt haben. Wenn das mit der e.V. -Eintragung klappt, möchte ich sowieso ein neues Deckblatt entwerfen.
Auch die manchmal schlechten Listingabdrucke wurden wieder angeführt. Dies liegt aber an den schlechten Vorlagen, die ich von Euch bekommen habe. Ich verweise nochmals auf die "redaktionellen Richtlinien".

Da wir nach Möglichkeit wieder ein SONDERHEFT für Euch herausbringen wollen, möchte ich Euch um Eure Mitarbeit bitten. Gesucht werden Artikel zu folgenden Themen:
Modem (Hard und Soft), Grafik M4/4p, ROM
Sicher habt Ihr schon einiges dazu in Euren "Archiven".

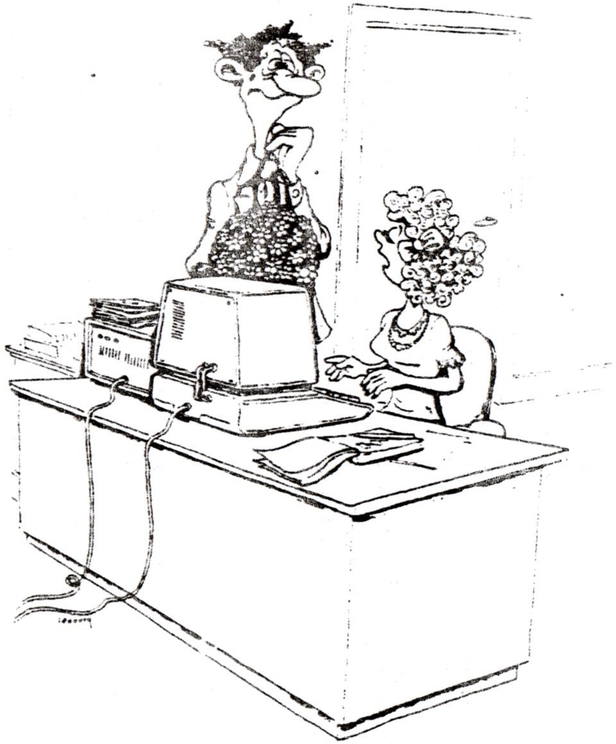
Des weiteren ist Mithilfe bei den Kurzbefehlslisten für BASIC und DOS erwünscht. In der Regel sind es Übersetzungen schon vorhandener Kurzanleitungen, sowie das Schreiben eines "Erfahrungsberichtes" über einen Befehl. Wie zum Beispiel in diesem INFO auf Seite 6 dargestellt. Bitte meldet Euch zur Koordination der Arbeit bei der Redaktion.

Ich wünsche Euch viel Freude an der neuesten INFO.

Bis zum nächsten INFO verabschiedet sich

Euer

J. Neueder



Der Computer der Partnervermittlung hat Sie aber etwas – nun ja – anders beschrieben . . .

CLUB 80 - ECB - BUS - SYSTEM

ANMELDUNG

Wie schon in der Systembeschreibung angeführt habt Ihr nun hiermit die Möglichkeit Euch an diesem Club-Projekt zu beteiligen. Diejenigen, die mitbauen wollen bitte ich, die Anmeldung auszufüllen. Auch der "Nichttechniker" findet durch unsere "Aktionsstufen" die Möglichkeit, kostengünstig an einen ECB-Bus-Rechner zu gelangen bzw. seinen Rechner um einen ECB-Bus zu erweitern.

Allgemeine Information an die Organisatoren

Um Sammelbestellungen zu koordinieren brauchen wir von den Interessierten natürlich einige Informationen. Auch wollen wir wissen, für welche Geräteversion Ihr Euch entscheidet. Wem im Moment nur die ECB-Bus-Erweiterung anspricht, hat natürlich auch später noch die Möglichkeit zum Computer aufzurüsten. Die entsprechenden Unterlagen werden ja im INFO veröffentlicht. Sicher fährt man aber günstiger, wenn man gleich an den geplanten Sammelbestellungen teil nimmt.

Ich interessiere mich für:

____ ECB-Bus ---- Bus Erweiterung
____ ECB-Bus ---- Selbstbau TRS 80

Ich wähle folgende "Aktionsstufe":

____ --- Kompletter Selbstaufbau in eigener Regie
____ --- Kompletter Selbstaufbau in einer Gruppe (2-er oder 3-er) bzw. mit teilweiser Unterstützung.
____ --- nur mechanischer Zusammenbau der fertigen Baugruppen. Die Platinen werden von uns mitaufgebaut und geprüft.
____ --- übernahme eines fertigen Gerätes (mit Wunschausstattung)
____ --- Ich bin nur theoretisch an dem Projekt interessiert und würde gegebenenfalls bei Problemlösungen helfen.

Vorschläge zu den Zusatzplatinen

Ich würde mich für folgende weitere, außer den schon vorhandenen oder geplanten, Zusatzplatinen interessieren:

____
____
____
____

ECB-Bus ---- Bestellung

Ich bestelle ______ Grundbausteine für je DM 250,--
und habe DM ______ auf angeführtem PostscheckKonto überwiesen.

....
Ort Datum Unterschrift

Achtung:

Die Bestellung hat nur Gültigkeit, wenn bis zum 20. August 1986 der entsprechende Betrag Eurer Bestellung auf dem

PostgiroKonto	17 63 33 -700
beim Postgiroamt	Stuttgart
BLZ	600 100 70
Kontoinhaber	Jens NEUEDER
	Panoramastr. 21, 7178 Michelbach/Bilz

unter der Bezeichnung Grundbaustein eingegangen ist !!

Ich werde dann die Sammelbestellung in die Wege leiten.

SOFTWARE-WETTBEWERB

Die Ausschreibung erfolgt im Genie/TRS80 User-Club Bremerhaven und im Club-80 gleichzeitig.

Aufgabenstellung:

Zu erstellen ist ein BASIC-Programm mit maximal 33 Zeilen. Die Thematik ist freigestellt. Das Programm muß auf den Rechnern Genie I, II, IIs, (III), oder TRS80 Mod. 1, Mod. 3 lauffähig sein und sollte unter der Herrschaft von NEWDOS80 oder G-DOS stehen. Je mehr "Spezialitäten" eingebaut werden, umso höher fällt die Bewertung aus (z.B. Diskettenroutinen, ansprechen der HRG, Druckerausgabe, Maschinenroutinen, etc.). Das Programm muß natürlich mit einer entsprechenden Betriebsanleitung ausführlich dokumentiert werden. Abtippen aus Zeitschriften (auch aus Clubinfos) oder einsenden von kommerziellen Programmen ist verboten. Im Namen aller Teilnehmer bitte ich um Fairness !!!

Zu gewinnen gibt's:

1. Preis: elektronische Schreibmaschine BROTHER EP-20

2. Preis: Ein 10er Pack Disketten

3. Preis: Buch "Start in die Computergrafik"

4.-10. Preis: je eine Leerdiskette

Die Preise wurden gestiftet von der Firma:

Peter Spieß
Offsetdruck + EDV-Zubehör
Trugenhofenerstr. 27
8859 Rennertshofen 1
Tel: 08434/454

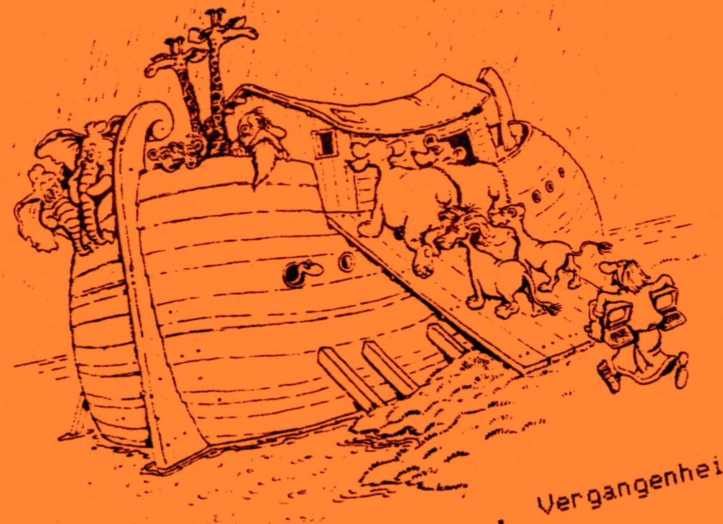
Einsendeschluß ist der 30.09.86
(Datum des Poststempels)

Der Rechtsweg ist ausgeschlossen ! Die Gewinner werden schriftlich benachrichtigt. Die Bekanntgabe erfolgt dann im jeweiligen Clubinfo. Bitte die Diskettenformatierung angeben ! Die Disk's werden nach der Auswertung an den Einsender zurückgeschickt. Einsendeadresse:

Genie/TRS80
User-Club
Peter Spieß
Trugenhofenerstr. 27
8859 Rennertshofen 1

CLUB 80

Clubinfo
der
TANDY -
GENIE -
und KOMTEK -
ANWENDER



.... Vergangenheit ....

... eine Schöpfungsgeschichte?

.... Zukunft ....

... wahrscheinlich ein Kultgegenstand einer untergegangenen Zivilisation!

15. AUSGABE

KONTAKTADRESSE : CLUB 80 / PETER STEVENS / POSTFACH 56 / 4600 DORTMUND 1
TEL.: 0231 / 593883

Inhaltsverzeichnis

Seite:

Clubinternes

Fragebogenauswertung . . . . 01 - 03
Termine / Gemischtes 04
ECB-Bus-TRS-80 allgemein 05
Vorstellung eines Mitgliedes 06

Software

Programmierwettbewerb 07
Kreis ohne Trigometrie 08
Logische Operationen in BASIC . . . 09 - 10
HRG-Bildübertragung ohne HRG . . . . 11 - 12
Berichtigung: Mein erstes 14
Mein 2. ASSEMBLER-Programm . . . . . 15 - 20
DIR-/ INHALT- SYS ? . . . . 21 - 24
The Disappearing DOS . . . . 25 - 26
Lib-Befehl: GO(,adresse. . . . . . . 27 - 28
Tscrips-Verschlüsselung . . . . . . 29 - 31
Ach du liebe Zeit! . . . . 32 - 36
Nachtrag: Ach du liebe Zeit . . . . 37 - 38
M4/4p Grafikshorty 39
M4/4p Grafikprogramm 40
M4/4p TANDY-Logo . . . . 41 - 42
M4/4p ROM-Image-Loader . . . . 43 - 46
M4/4p UHRUNTEN . . . . 47 - 48
CALL um die Ecke . . . . 49 - 50
Korrektur: CALL um die Ecke. . . . . 51 - 52
Real Time Black Box? . . . . 53 - 56
RPNL- Programmiersprache . . . . . . 57 - 64

Hardware

ECB-Adapterplatine . . . . 65 - 70
ECB-Bus-Projekt: Grundeinheitsaufbau 71 - 72
ECB-Bus-Projekt: Eingangsplatine . . 73 - 76
ECB-Bus-Projekt: ZK01/02 . . . . . . 77 - 83
ECB-Bus für M 3/4?? . . . . 84 - 86

Seite:

Börse

Wer hat was ???--Wer sucht was ??? 87 - 88
Wer kann weiterhelfen . . . . 87 - 88

Sonstiges

Normung und Rechtschreibung . . . . 89 - 90
Sprach-Entwicklung 92

Programmbibliothek

Einleitung 93
Public-Domain-Software . . . . 94 -106

Die letzten Seiten

Impressum 107
Schluß 108
Clubmitgliederadressen . . . . am INFO-Ende
Eingangskarten-Layout . . . . am INFO-Ende
ZK01/02-Layout . . . . am INFO-Ende

Nachdem ich Euch diesmal an dieser Stelle, wie sonst üblich, kein Vorwort von unserem 1. Vorsitzenden bieten kann, möcht ich den Platz nutzen und die schon lange fällige Fragbogenauswertung bringen. Ich hoffe, daß genügend Informationen, für Euch, darin enthalten sind. Leider hatte ich keine Zeit mehr, die Antworten zu jedem Thema, zu sortieren.

FRAGEBOGEN MITGLIEDERDATEN UND -INTERESSEN

AUSWERTUNG

Es wurden durch das Club-INFO 53 Fragebogen an die Mitglieder versandt.
Davon kamen bis zum Auswertungstag, den 31.08.1986, 29 Stück zurück.

2. Hardwaredaten :

Computer : TRS-80 MI 7x, TRS-80 MIII 1x, TRS-80 M4/M4p 5x
VG EG3003 2x, VG I 10x, VG II 6x, VG IIs 1x, VG III 2x, VG IIIs 1x
Komtek 1 1x, MC CP/M, ECB-TRS-80 1x

Drucker : EPSON MX 80 FT 3x davon mit Graftrax 1x, RX 80 FT 2x, TA DRH 80 2x
NEC 8023B-C 2x, ITOH 8510A 3x, DP 510 3x, STAR RADIX 10 1x, SG-10 1x
Quen Data DMP-1180 1x, HEATH H14 1x, CP80 1x, Gemini 10X 5x
Tandy LP VI 1x, LP VIII 1x, SD 4035 1x, KX-P1091, FX 80 1x

Laufwerke : 40 Trk,DS,DD 14x, 40 Trk,SS,SD 7x, 40 Trk,SS,DD 11x
80 Trk,DS,DD 39x, 35 Trk,SS 1x
Teac 55A+55B, SA 405 1x, SD 5211x, ohne Angabe 2x

HRG : 640 X 240 1x, 384 X 192 13x, 256 X 512 1x, 480 X 192 1x, IIIs-HRG 1x

CP/M : 12x davon Montezuma 1x, 2.2 5x, 3.0 1x, diverse 2x

Telefon-Modem : Tandy AC3 2x, Dataphon s21d 1x, RB 300 1x, ohne Angabe 1x

Modemerfahrung: JA 3x, gering 5x

sonstiges Zubehör :RS 232 2x, Prommer 4x, Spooler 64K 2x
Exatron Stringy-Floppy 2x, Gripp II 1x
Z 80-Zeichenkarte 2x, V24 1x, 3.54 MHz 3x
Banker: 256K RAM, ROM/RAM 2x
I/O /RAM 1x, Grafik 512 X512 1x, Basiccode 2x
Analog/Digital -Wandler 1x, Supertape 2x
Schnittstellen 1x, Joystick 2x, Lightpen 1x
Video-Snovel 1x, ECB-Bus 1x, Inverse Zeichend. 1x
Meßwertaufnahme 1x
Nixdorf Kugelkopfdrucker nicht anschlußfertig 1x
Telefax-Gerät 1x, Typenradschreibmaschine 4x

3. Deine Interessen :

Mein Hauptinteresse liegt bei: Software 17x Hardware 4x Beidem 8x

Ich "computere" seit:
2 J. 2x, 3 J. 7x, 4 J. 6x, 5 J. 5x, 6 J. 5x, 7 J. 1x, 8 J. 2x, 9 J. 1x

Im Moment beschäftige ich mich mit folgendem:

Software : HRG-Software 4x davon M4 1x, HRG1B 2x, VG IIs 1x
alles 2x, Z80 Assembler 2x, RPNL-Compiler 1x, mathematisches 2x
Analyse + Änderung VG IIIs 1x, NEWDOS 2x, C 1x, CP/M 1x
TURBO-PASCAL 2x, Programmanpassung MI auf M4/4p/3 1x
Spiel / Postspiele 1x, 80-Zeichenanwendung 1x, Textprogramme 3x
DBASE 1x, Fortran 1x, Bau-Programme 1x, Vereinsverwaltung 1x
Datenverwaltung(Sport) 1x, Programmiersprachen 1x
Datenaustausch C64/TRS-80 1x, Terminalprogramme 1x

Hardware : ECB-Bus 5x, Telefax Signale auf HRG 1x, Reperaturen 1x
Erweiterung eines Z80 Selbstbaues 1x, HRG1B-Selbstbau 2x
VG III aufwärtskompatibler Selbstbau 1x, EPROM brennen 1x
Analog/Digital 1x, Steuern/Messen 2x, Echtzeit-System 1x, RS232 1x
Spooler 2x, Direktgekoppeltes Modem 1x, Uhr im 4p 1x

Allgemein : Programmanwendungen 3x, CP/M Software 2x, Computer + Recht 1x
Maschinensprache 1x, BASIC 1x, Elektronik 1x, Modellfliegen 1x
Altbier 1x, alles 1x, wenig 1x

Ich würde mich für folgendes interessieren:

Software : HRG-Software 7x davon M4 1x, HRG1B 4x, VG IIs 1x, Drucker 1x
Utilitys 3x davon M4 1x, CP/M 3x, Amateurfunk allg. 1x
Softwareschutz/-Knack 1x, Funktionsbelegung 1x
dt. Anleitung Mumath 1x, Fortran 80-Anwendungen 1x, alles 2x
gebanktes CP/M 3.0 f. VG I und a. BANKER 2x, Videodat 300 1x
Mathematische Probleme 2x, 80-Zeichen 1x, Lisp 1x, Ram-Disk 1x
Simulationen 1x, Lehrprogramme 1x, ständiger Hintergrundmonitor 1x

Hardware : ECB-Bus-TRS-80 selbstbau 5x, ECB-Bus 3x, Modemselbstbau 1x
ECB-Adapter 1x, Meßwerterfassung 2x, Maus 1x, HRG 2x, Drucker-HRG 1x
prellfreie Tastatur 1x, Videodat 300 1x, VG-Ausbau 1x, alles 2x

Allgemein : PASCAL 1x, Funktionen f. Grafik 1x, Visicalc-Tastenbelegung GIII 1x
68020 CPU 1x, Pils 1x, alles 2x

Ich fände es gut, wenn es in unserem Club eine
Assembler-Gruppe 2x, CP/M-User-Gruppe 2x, HRG-Ecke 4x, Lisp-Ecke 1x
BASIC-Gruppe 1x, Modem-Ecke 2x, (Norddeutsche) Hardware-Gruppe 2x
Anwender) VG I 1x, Textverarbeitungs-Ecke 1x, Schutz/Knack-Ecke 1x
Mathe/Naturwissenschaft-Ecke 1x, Anfänger-Ecke 1x geben würde.

sonstiges : disassemb. Dos-Listing ab SYS1 1x
öftere Clubtreffen 1x, regionale Treffen 1x

4. Clubleben :
Wie siehst Du Dich als Clubmitglied ?
mehr passiv 10x mehr aktiv 13x unentschieden 6x
Welche Themengruppen würden Dich in der Info interessieren ?
Bitte prozentual angeben (Wichtigkeitswertung) !!

a) Grundlagen 556,95%: Allgemeines 15x, Basic 12x, Maschinensprache 21x
b) sonstiges: Hardware 2x, Programmstrukturierung 1x
DOS-Vergleich 1x, Hardwareelektronik 1x
Ram-Erweiterung 1x, BANK-Switching 1x
c) Allgemeines 282,53%: Neuigkeiten 21x, Unterhaltendes 12x
d) Clubinternes 221,07%: was besonders: Computer + Recht 1x
prakt. Erlebnisse 3x (Kauf, Test)
Börse 1x, Gruppenarbeit 1x
Bereichstreffen 1x, wer macht was 2x
Hardware der Mitglieder 3x
e) Programmiersprachen 526,01%: welche: FORTH 3x, BASIC 7x, PASCAL 13x, C 5x, COBOL 2x
FORTRAN 5x, RPNL 1x, Z80-Assembler 8x
alles 1x
f) Software 689,77%: Thematik: Grafik 6x, Utilitys 4x, Mathemathik 3x
Programmiertechniken 2x, Hardware-Treiber 1x
Dos-Patches 1x, Datenbanken 1x, Softwarelösungen 1x
Erklärungen 1x, Verbesserungen 2x, Assembler 1x
Betriebssysteme 1x, Modem 1x, Banker 1x, allgemein 1x
Programmvorstellungen 2x, Anwendungen 1x, Physik 1x
Dateienverwaltung /Datenbanken 1x, Terminalsoftware 1x
Simulationen 1x
g) Hardware 523.67%: Thematik: ECB-Bus (alles) 6x, DFü 1x, RAM-Disk 2x
Interface 1x, Verbesserungen 3x, Modem 2x
kleinere/mittlere Projekte 2x, Kompatible 1x
Amateurfunkerweiterungen 1x, alles 4x
Messen/Regeln/Steuern 1x
Computerkommunikation 1x
h) Dein Vorschlag dazu : TIPS/TRICKS 1x, Fragebogenbeantwortung 1x
mehr eigene Beschreibungen/Kritiken d. Mitglieder zu
Hard und Soft 1x, Lehrg. Künstl. Intelligenz 1x

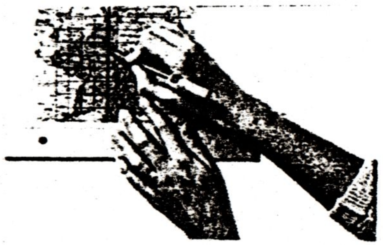
--- einmal keine Angaben deshalb nur 2800% von 29 Antworten (100% freibleibend)
Daraus ergibt sich ein effektiver prozentualer Anteil (umgerechnet/gerundet):
a) 19,89% -- c) 10,09% -- d) 7,9% -- e) 18,79% -- f) 24,63% -- g) 18,7%

Hättest Du Lust dich in irgendeiner Art und Weise
für den Club zu engagieren ? Ja 22x Nein 4x Keine Angabe 3x
Wenn ja, wobei: Papierbeiträge 5x, kleine Hardware-Probleme 1x
Anfängerkurs Hardware 1x, Programmiersprachen 1x
Hardware 5x, je nach Thematik 1x, Anleitungen 1x
Software 5x, übersetzungen 1x, laufende Clubarbeit 5x
Was gefällt Dir an der INFO nicht ? keine Angaben 9x
Lesbarkeit der Listings 10x
" " Tel.-Nr. auf der Titelseite 1x
teilweise falsche Rechtschreibung 1x
zu viele Zeitschriftenartikel 2x
Klebebindung 1x, zu wenig Autoren 1x
nicht computerbezogene Witze 1x
teilweise zu Systemspezifisch 2x
zu wenig Hardware 1x
uninformative überschriften 1x
Was gefällt Dir an der INFO ? bis auf ... in Ordnung 16x
gutes Layout 9x, hohes Nivea 2x
gute Autoren 1x
computerbezogene Witze 2x

Hobbytronik in Stuttgart vom ....15. - 19. Oktober 1986
Wir treffen uns am Sonntag den 19.10.86
jeweils 2-stündig an der Information am
Eingang. Erster Treff gleichzeitig mit
Beginn der Eröffnung um 9.00 Uhr.
Platinenbestellschluß für ECB-TRS-80 ..12. Oktober 1986
Nächster Redaktionsschluß ....31. Oktober 1986
Norddeutsches Regionaltreffen ....25. + 26. April 1987

Aus beruflichen Gründen ist Hartmut Obermann
die Woche über nicht zu erreichen!
Dies gilt von jetzt bis zum 27.11.86
Wer noch Post von Ihm zu bekommen hat muß sich
leider etwas gedulden. Telefonisch kann der
Hartmut samstags erreicht werden.

HRG-Pack-Tip !!
Möchte man zu HRG-Pack ein spezielles
BASIC geladen haben (z.B. Basic/IV),
wird RUNTIME mit Extension IV eingegeben.
Ulrich Böckling



Mit einem Elektronikstift gibt man eine Landkarte auf einem »Tablett« in den Computer ein.

Digitizer

Um dem Computer schnell und zuverlässig zu sagen, was er auf dem Bildschirm zeichnen soll, gibt es »Digitizer«, Digitalisierungsgeräte, die eine analoge Zeichnung in Bruchteilen von Sekunden in digitale Signale umwandeln. Diesen Code des Zeichentabletts versteht der Computer und läßt die entsprechenden Bildpunkte auf dem Monitor aufleuchten. Doch woher weiß das Zeichenbrett, wo ich gerade meinen Stift ansetze? Der Stift ist mit einem Radiosender vergleichbar. Eine Spule sendet elektromagnetische Wellen aus. Das Tablett ist die Antenne. Unter der Oberfläche liegt ein dichtes Netz feinster Antennendrähte, die von Chips abgefragt werden. Wo der Empfang am stärksten ist, da muß der Griffel gerade zeichnen. ✶

HARDWAREPROJEKT ECB-BUS-TRS-80

Nach dem Bestellaufruf -für die Grundeinheit- im INFO 14 haben sich zehn Clubmitglieder gemeldet. Acht von Ihnen sind aktiv in die Sache eingestiegen. Für diese acht "Mutigen" habe ich die entsprechenden Bauteile Anfang September bestellt und bekommen bzw. selbst beim Hersteller besorgt. Die Bauteileauslieferung erfolgt nach dem 28. September, da ich erst zu diesem Zeitpunkt die Netzteile vom Bernd bekommen kann.

Zwischenzeitlich sind auch die ersten Anleitungen für unser Projekt fertig. Einmal ist dies der Grundeinheitsaufbau und zum Anderen die ersten beiden Platinen:
Eingangsplatine TRS-80 und ZK01/02
Beide Karten dienen jetzt im Moment in erster Linie zur überprüfung des Bus, sind aber später auch weiterhin verwendbar.
Die Eingangsplatine stellt die Verbindung zwischen dem schon vorhandenen Computer und dem ECB-Bus her. Dadurch hat man die Möglichkeit, die restlichen neun Steckplätze der Busplatine, auf die anliegenden ECB-Bus-Signale zu überprüfen oder später diese Steckplätze mit den in den folgenden INFO's vorgestellten Zusatzkarten zu belegen. Eine davon wäre zum Beispiel die in diesem INFO vorgestellte ZK01/02. Auch "Selbstgestricktes" ist durch Bus und Eingangsplatine an den "alten" Computer dann anschließbar.
Zur überprüfung des Bus ist es notwendig, jeden freien Kartenplatz einmal auszutesten, dies geschieht am einfachsten mit den Testprogrammen von Seite 78 in Verbindung mit ZK01/02.

Der weitere Aufbau unseres ECB-Computers geht wie folgt von statten:
Nachdem nun die Grundeinheit aufgebaut und getestet ist, kann jeder nach Belieben seine "ECB-Träume" verwirklichen. Wir werden dazu beitragen, indem wir in den folgenden INFO's ECB-Zusatzkarten und natürlich die vier ECB-Computerkarten vorstellen. Die Platinen dazu können -von dem der gerne alles machen möchte- selbst erstellt werden. Die entsprechenden Vorlagen im Maßstab 1:1 liegen jeder INFO bei. Ansonsten sind die Platinen für ca. 12,-- bis 15,-- DM (gegen Vorauskasse) fertig geätzt aber ungebohrt vom Bernd erhältlich. Die Bauteile, die in den Schaltungen verwendet werden, sind in jedem Elektronikladen erhältlich. Da es sich in der Regel um herkömmliche Bauteile handelt, möchten wir darum bitten, sie Euch -entsprechend der jeweiligen Stückliste- selbst zu besorgen. Porto- und Verpackungskosten würden hier den eventuellen günstigeren Einkauf wieder verteuern. Auch wäre unser Zeitaufwand für die Sache nicht mehr vertretbar. Wo es sich aber um spezielle oder auch kostenaufwendige Bauteile handelt, werden wir auf jeden Fall eine Sammelbestellung rechtzeitig anbieten.

Falls sich doch einmal Versorgungsprobleme ergeben sollten, sind wir natürlich gerne bereit diese zu beseitigen. Bitte meldet Euch rechtzeitig bei einem von uns wenn es für Euch irgendwelche Probleme beim Nachbauen gibt.

Viel Spaß beim Basteln und bis zur nächsten "ECB-Bus-Vorstellung" Euer
Jens

Rich. Rensch, Orgelbaumeister
Bahnhofstr. 100, 7128 Lauffen
Postfach 226 W.Germany Tel. 07133-8415

EIN NEUES MITGLIED STELLT SICH VOR: Juli 1986

RENSCH über RENSCH

Vor etwa 10 Jahren fing es an: mit programmierbaren Taschenrechnern. Ab April 81 erste Erfahrungen mit BASIC an einem SHARP PC 1211. Nach etwa 1 Jahr waren aber doch die mageren 1024 Bytes ausgereizt, und es kam ein gebrauchtes Video Genie I (EG 3003) mit „unglaublichen" 16 KB her. Es lebt heute noch, natürlich längst mit Expander, Doubler, einem 80- und zwei 40-Track-Laufwerken, alles DS/DD und zum Umschalten zwischen LW 0 (=40 Tr.) und LW. 2 (=80 Tr.) eingerichtet (ähnlich Vorschlag in Info 14, S. 87). Aber schon 1982 brachte der Weihnachtsmann den neuen GENIE III. Im CHIP-Preisausschreiben gewann ich auch noch den HAUPTGEWINN, ein „Textsystem" (GENIETEXT 2.0, der erste und wohl einzige Hauptgewinn in meinem Leben!). Seitdem verstauben die verschiedensten Scripsit-Versionen (mit Kurs in Deutsch auf CC), Zorloff und Newscript (trotz Korrektur-Lexikon und automatischem Glossarschreiber), weil es alles nicht an den GTXT herankommt!

Der jüngste Filius (das letzte unserer 9 Kinder) strickte ein recht brauchbares Nachkalkulationsprogramm (in Basic) für Orgelbauten, mit dem wir nun schon seit 1983 recht erfolgreich arbeiten. Weitere professionelle Programme kamen bald dazu (GP 4, Visicalc, Lohnabrechnung usw.). Nur für die Buchführung hatten wir seit 1979 schon einen TA 20 NP (mit erstaunlichen 8 KB für 25.000,- DM und Magnetbandstreifen auf den Kontokarten als „Massen-"Speicher!), der bei der nächsten Steuergesetzänderung prompt nicht mehr mitkam. Jetzt machen wir die Buchführung mit einem Kompatiblen (Commodore PC 20 mit Festplatte).

Vor etwa 1 Jahr streikte der GENIE III plötzlich total, gerade als Trommeschläger pleite machte. „Phönix" ließ kopfschüttelnd mitteilen, daß sie „solche alten Modelle" nicht mehr unterstützen könnten. Der Monat ging dem Ende entgegen, und die Lohnabrechnung war fällig, für die wir früher immer ca. 4 Tage gebraucht hatten (mit dem Computer nur noch 1/2 Tag!). Was tun? Keiner wollte so einen Uraltcomputer noch reparieren. So schaute ich in die Kleinanzeigen von CHIP und fand tatsächlich einen zu verkaufenden GENIE III („wegen Umstellung") für „lumpige" zweieinhalb Tausender (mit Software - nochmal ein originaler GENIETEXT, sogar mit Kalkulation!). Der Zahltag war gerettet, und schließlich fand sich der (manchmal m.E. zu Unrecht gescholtene) Schmidtke in Aachen bereit, den anderen GENIE III für lumpige 600,- DM wieder flott zu machen: er lötete sämtliche Lötstellen auf den Platinen nach! Das hätte ich schließlich auch noch hingebracht. Löten 20,- DM, „Gewußt wo" 580,- DM!

Immerhin habe ich nun ganz allein für mich einen „Seniorchef-Computer", und ich diktiere kaum mehr, sondern schreibe fast alle Briefe selber.

Dem CLUB 80 gehöre ich nun seit März 86 an. Es ist der dritte Computerclub, die beiden Vorgänger sind sanft entschlafen. Es scheint auch der erste Club zu sein, dessen Club-Info funktioniert. Leute, seid Euch dessen bewußt und pflegt Eure fleißigen Officials. Auch ich will gerne etwas dazu helfen, soweit es meine schwachen Kenntnisse erlauben. Ich hoffe auch, daß der KaJot mit seinem Baby- und Opa- Assemblerkurs (das meine ich durchaus positiv, denn sowas fehlt meist!) weiter macht (bitte nicht zu schnell!). Vielleicht können damit einige von uns doch noch die höheren Computerweihen der Assemblerkunst erlangen. Also auf ans Werk!

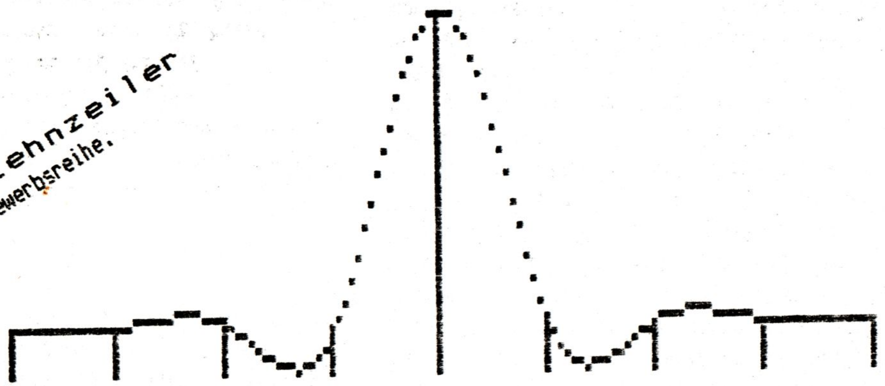
Richard Rensch

SPLINE-INTERPOLATION

Dieter Kaspar

```
1 READ N:N=N-1:DIM X(N),Y(N),D(N),R(N),B(N),C(N),Z(N),XX(128),YY(128): FOR I=0 T
O N: READ X(I) : NEXT                                       : FOR I=0 TO N: READ Y(I) :
 NEXT
2 D(0)=X(1)-X(0):FORI=1TON-1:D(I)=X(I+1)-X(I):C(I)=2*(D(I-1)+D(I)):NEXT:Z(2)=D(1
)/C(1):FORI=3TON-1:Z(I)=D(I-1)/(C(I-1)-D(I-2)*Z(I-1)):NEXT:FORI=1TON-1:R(I)=(Y(I
+1)-Y(I))/D(I)-(Y(I)-Y(I-1))/D(I-1):R(I)=3*R(I):NEXT:'Tridiag GLS
3 B(1)=R(1)/C(1):FORI=2TON-1:B(I)=(R(I)-D(I-1)*B(I-1))/(C(I)-D(I-1)*Z(I)):NEXT:C
(N-1)=B(N-1):FORI=N-2TO1STEP-1:C(I)=(B(I)-Z(I+1)*C(I+1)):NEXT:C(0)=0:C(N)=0
4 FOR I=0 TO N-1: B(I)=(Y(I+1)-Y(I))/D(I):B(I)=B(I)-((2*C(I)+C(I+1))*D(I)/3):D(I
)=(C(I+1)-C(I))/(3*D(I)):NEXT:W=(X(N)-X(0))/127:XX(0)=X(0):FORI=1TO127:XX(I)=XX(
I-1)+W:NEXT
5 FOR J=0 TO 127: XX=XX(J):FOR I=0 TO N-1: IF (X(I+1)-XX)>=0 THEN XX=XX-X(I):YY(
J)=Y(I)+XX*(B(I)+XX*(C(I)+XX*D(I))): GOTO 7
6 NEXT I
7 NEXT J:IN=YY(0):SU=IN:FOR I=1 TO 127:Y=YY(I): IF Y>SU THEN SU=Y :GOTO 8 ELSEIF
 Y<IN THEN IN=Y
8 NEXT: FOR I=1 TO N:Y=Y(I):IF Y>SU THEN SU=Y:NEXT:ELSEIF Y<IN THEN IN=Y :NEXT E
LSE NEXT: CLS:FORI=0TON:Y=Y(I):GOSUB10:K=(X(I)-X(0))/W:FORJ=47TOZSTEP-1:SET(K,J)
:NEXTJ,I:FORI=0TO127:Y=YY(I):GOSUB10:SET(I,Z):NEXT
9 FOR I=0 TO N: Y=Y(I): GOSUB 10: K = (X(I)-X(0))/W: FOR J=47 TO Z STEP -1: SET
(K,J): NEXT J,I : FOR I=0 TO 127 : Y=YY(I): GOSUB 10 :        SET (I,Z): NEXT:GOT
09
10 Z=INT(47*(SU-Y)/(SU-IN)):RETURN:                                        DATA 9,
                                                             1,2,3,4,5,6,7,8,9,
                                            0,0,0,0.5,0,0,0,0
```

... und wieder ein Zehnzeiler in unser Wettbewerbsreihe.



»Unsere Zeit ist so aufregend, daß man die Menschen eigentlich nur noch mit Langeweile schockieren kann.«

Kreis ohne Trigonometrie

In der mc 8/86 steht ein kleines Plot-Programm, das Kreise ohne die Verwendung von trigonometrischen Funktionen zeichnet. Das Programm ist für diejenigen interessant, die sich wie ich ohne CIRCLE-Befehl durch das Leben zeichnen müssen.

Ich habe das Programm nur in einem Punkt abgeändert: der Divisor IN war im Original mehr oder weniger zufällig gewählt, so daß entweder kleine Kreise stundenlang gezeichnet wurden oder große Kreise ebensogroße Lücken aufwiesen. Die folgende Version sollte bei jeder Kreisgröße einen lückenlosen Kreis in maximaler (Basic-)Geschwindigkeit ergeben.

```
10 MX=Mittelpunkt, X-Koordinate
20 MY=     "     , Y-    "
30 R=Radius
40 X=R : Y=0 : IN=1.1*R
50 X=X-Y/IN
60 Y=Y+X/IN
70 PLOT MX+X,MY+Y : PLOT MX+X,MY-Y : PLOT MX-X,MY+Y : PLOT MX-X,MY-Y
80 IF X>0 THEN 50
```

Gerald Schröder

Vorlage von Jürgen Zahn in mc 8/86

»Die Frau wird erst an dem Tag mit dem Mann wirklich gleichberechtigt sein, an dem man auf einen bedeutenden Posten eine inkompetente Frau beruft.«

Logische Operationen in BASIC

OR, AND und NOT, die sog. logischen Operationen, sind für viele Basic-Programmierer ein Buch mit sieben Siegeln. Dabei sind diese drei Operationen nur die Grundfunktionen, von denen alle weiteren Logikoperationen, wie NAND, NOR, XOR abgeleitet werden können. Um wenigstens ein wenig Verständnis für diese Funktionen zu wecken, hier ein paar Grundlagen und weiterführende Gedanken!

Am besten veranschaulicht man sich die logischen Operationen anhand sog. Funktionstabellen. Sie enthalten, je nach Funktion, ein oder zwei Eingangszustände und den Ausgangszustand. Die Funktionstabelle für die AND-Verknüpfung verdeutlicht dies:

A	B	C
0	0	0
0	1	0
1	0	0
0	1	0

Wie man hier sehen kann, sind auf der linken Seite die beiden Eingänge in den jeweiligen Zuständen gezeigt, die sie annehmen können. Die rechte Seite zeigt den daraus resultierenden Ausgang. Die Funktionstabelle verdeutlicht sehr gut, was eigentlich schon der Name der Funktion aussagt:
Am Ausgang C erscheint nur dann eine "1", wenn Eingang A und Eingang B eine "1" führen!

Auch bei der OR-bzw. ODER-Verknüpfung macht die Funktionstabelle sehr schnell deutlich, wie die Sache funktioniert:

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Der Ausgang C wird immer dann eine "1" sein, wenn mindestens einer der Eingänge eine "1" führt. Ob dies Eingang A oder B ist, spielt keine Rolle.

In Basic sieht die Sache so aus:

```
10 FOR A = 0 TO 1
20     FOR B = 0 TO 1
30         C= A OR B
40         PRINT A,B,C
50     NEXT B
60 NEXT A
```

Was jedoch passiert, wenn wir von den 0'en und 1'sen weggehen? Nehmen wir einmal an, A soll 7 sein, B hat den Wert 1! Wie sieht das Ergebnis in C aus? Eigentlich ist die Sache ganz einfach, auch wenn man sich etwas unter die Oberfläche des dezimalen Zahlsystems begeben muß. Die BASIC-Operationen OR, NOT und AND behandeln die ihnen übergebenen Argumente immer in ihrer binären Darstellungsweise. Es wird also innerhalb der Maschine die Operation 111 b AND 101 b durchgeführt. Setzen wir diese beiden Binärzahlen in eine etwas abgewandelte Funktionstabelle um, sehen wir schnell, warum die BASIC - Operation 7 AND 5 als Ergebnis 5 bringt!

```
      1 1 1 b            7 d
AND   1 0 1 b     AND    5 d
-------------     ----------
=     1 0 1 b     =      5 d
```

A	C
0	1
1	0

Sehen wir uns jetzt die Operation NOT an. Sie hat nur einen Eingang und bewirkt am Ausgang die genaue Umkehrung des Eingangswertes! Aus einer 0 am Eingang wird eine 1 am Ausgang und umgekehrt!

Was bringt die BASIC-Zeile "PRINT NOT A", wenn A zuvor mit dem Wert 9 geladen wurde???

Ganz einfach:

```
        1 0 0 1 b         9 d
   NOT  ---------    NOT  ---
        0 1 1 0           6 d
```

Soweit also die Grundfunktionen. Was hat es nun mit den weiterhin genannten Operationen NAND (NOT AND), NOR (NOT OR) und XOR (eXclusiv ODER) auf sich? Diese sind im BASIC nicht realisiert, können jedoch aus den Grundfunktionen zusammengesetzt werden! Die Funktionstabellen zeigen wie:

NOR

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

NAND

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Als BASIC-Zeile wäre:

```
NOR  -> C = NOT (A OR B)
NAND -> C = NOT (A AND B)
```

Bei beiden Funktionen wird also jeweils das Ergebnis der Grundfunktion negiert (NOT-verknüpft!).

Bei der XOR-Operation wird's etwas komplizierter. Hier erst mal die Funktionstabelle:

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Der Ausgang ist immer nur dann '1', wenn die beiden Eingänge ungleich sind! Haben beide Eingänge gleiche Wertigkeit, ist der Ausgang '0'!

Wie hat nun aber die BASIC-Zeile auszusehen, in der wir mit Hilfe der implementierten Logikoperationen OR, AND und NOT die XOR-Funktion simulieren?

Schamhaft muß ich gestehen, daß ich auch nicht ohne fremde Hilfe auf die Lösung des Rätels kam (manchmal habe ich wirklich ein gewaltig dickes Brett vorm Kopf!). Ich mußte mir von Ulrich Böckling mit Rat und Tat zur Seite stehen lassen. Trotzdem, oder gerade deshalb, möchte ich einen kleinen Wettbewerb starten und biete demjenigen, der mir als erster die Lösung des Problems als BASIC-Zeile schickt, eine Flasche Wein aus eigenem Weinberg (garantiert glycolfrei!!!). Disqualifiziert werden all die Teilnehmer, die jetzt schnell Ulrich Böckling anrufen, um sich dort, genau wie ich, Hilfe zu holen. Als Datum gilt das Datum des Poststempels und nicht etwa die Ankunft der Lösung bei mir, damit diejenigen, die weiter weg wohnen, nicht benachteiligt sind. Telefonanrufe zählen nicht!
Hier nochmals das Problem bzw. die Frage:

C = A XOR B

Wie muß die BASIC-Zeile aussehen, die mit Hilfe der Funktionen OR, AND und NOT diese Verknüpfung realisiert!

Im nächsten Heft folgt dann die Auflösung, die Nennung der Teilnehmer und des Gewinners und noch ein paar Tips und Tricks für den Umgang und die Verwendung der Logikoperationen! Viel Glück und Erfolg wünscht Euch

Hartmut Obermann

HRG - Bildübertragung ohne HRG!!!

Mit dem unten abgedruckten Programm können im WDR-Format abgespeicherte, hochauflösende Bilder direkt auf einen grafikfähigen Drucker ausgegeben werden. Dadurch kommen auch Grafikfreaks ohne HRG in den Genuß hochaufgelöster Bilder! Das Programm ist auf einem Model 4P und für den EPSON RX-80 geschrieben. Für andere Drucker und das Model 1 müssen natürlich Änderungen vorgenommen werden.
Das Programm ist ab sofort bei Kajott in der Clubdiskothek zu haben. Dort bekommt ihr auch Bilder im WDR-Format!

Hartmut Obermann

```
110 REM DRUCKBILD - fuer TRS-80 Model 4 mit Level II - BASIC
140 REM W.BACK MAI 85 geaendert 6.86 Hartmut Obermann
160 CLEAR 1000 : CLS
161 IF INP(248)<>61 THEN PRINT@0,"Drucker nicht bereit!";:GOTO161
162 'Zeilenvorschub auf 'eine Pixelhoehe' einstellen
163 LPRINTCHR$(27);"3";CHR$(3)
165 ON ERROR GOTO 390
170 DEFINT A-Z
180 CLS:PRINT:PRINTTAB(10)"* * * B I L D - D R U C K E R * * *":PRINT
185 PRINT"Dieses Programm druckt im WDR-Format gespeicherte Bilder auf
    grafikfaehigen EPSON-Druckern aus!":PRINT
190 PRINT "Dateiname : ";:LINE INPUT NAM$
200 PRINT"Laufwerk : ";:LINE INPUT LAUF$
210 OPEN"I" , 1 , NAM$ + ":" + LAUF$
220 LINE INPUT #1,N$ : X = INSTR(N$,"BILD") : IF X=0 THEN 220
230 LINE INPUT #1,R$ : PRINT R$
240 LINE INPUT #1,V$ : VE% = VAL(MID$(V$,2,3)) : PRINT VE%
250 LINE INPUT #1,H$ : HO% = VAL(MID$(H$,2,3)) : PRINT HO%
260 LINE INPUT #1,D$ : LINE INPUT #1,D$
265 PRINT"Werte OK!? ";
266 J$=INKEY$ : IF J$="" THEN 266
267 PRINTJ$ : IF J$<>"J" AND J$<>"j" THEN 220
268 'N$ = kennbuchstabe fuer normale (K) oder doppelte (L) Dichte
     N1 und N2 repraesentieren die Anzahl der zu druckenden Pixel
269 IF HO%>480 THEN N$="K" ELSE N$="L":
    N2=CINT(HO%/256) : N1=CINT(HO%-N2*256)
270 FOR V = 0 TO VE%
280 IF EOF(1) THEN GOTO 382
290 LINE INPUT #1 , A$ : X=0 : IF LEN(A$)=0 THEN NEXT V
291 IF INSTR(A$,"Uebertragung beendet")<>0 THEN GOTO 381
294 'EPSON-Drucker auf den benoetigten Grafikmodus einstellen
295 LPRINTCHR$(27);N$; : Z=N1 : GOSUB 10000 : Z=N2 : GOSUB 10000
300 FOR H = 0 TO HO% STEP 6 : X = X + 1
310 A = ASC(MID$(A$,X,1)) - 32
320 IF A AND 1 THEN Z=128 ELSE Z=0 : GOSUB 10000
330 IF A AND 2 THEN Z=128 ELSE Z=0 : GOSUB 10000
340 IF A AND 4 THEN Z=128 ELSE Z=0 : GOSUB 10000
350 IF A AND 8 THEN Z=128 ELSE Z=0 : GOSUB 10000
360 IF A AND 16 THEN Z=128 ELSE Z=0 : GOSUB 10000
370 IF A AND 32 THEN Z=128 ELSE Z=0 : GOSUB 10000
380 NEXT H : Z=13 : GOSUB 10000 : NEXT V
381 PRINTA$
382 CLOSE: PRINT"Bild gedruckt!"
389 LPRINTCHR$(27);"@":END
390 PRINT@0,"*":PRINT@0," ":RESUME 380
9999 'Ausgabe der Grafik an den Drucker unter Umgehung des normalen
      Druckertreibers, da dieser bestimmte Zeichen 'verschluckt'!
10000 I=INP(248):IFI<>61THEN10000
10010 OUT 248,Z : RETURN
```

Es stimmt! Die Inventur ist wirklich viel einfacher mit dem Computer.

Man erzählt sich, daß alles mit einem Fehler im Programm begann . . .

Selbst Gott braucht die Werbung. Er hat Glokken.

Die Freiheit des Menschen liegt nicht darin, daß er tun kann, was er will, sondern daß er nicht tun muß, was er nicht will.

CTRL SHIFT RETURN BREAK INSERT CLEAR GOTO
SELECT START < DELETE BAKS ← SHIFT LOCK
SYSTEM RESET ↑ RETURN LINE FEED ESC RPT
SHIFT ↓ RETURN BS → SELECT INSERT +
ENTER SHIFT 1 = CTRL START= RETURN + > , 0
CLEAR BACKSPACE TAB PRINT REM . . . .

"Mein erstes Assembler-Programm" –
eine Berichtigung zur Beschwichtigung!
=======================================

"Na so ein Heini! Der will uns was erzählen?" – das hat so mancher von euch, der bereits in den ersten Kinderschuhen der Fa.Assembler steckt, gedacht, sofern er meinen Beitrag im letzten Heft gelesen hat. "Wie kann man nur 'REPEAT' mit 'ER' abkürzen und uns glauben machen, es hieße 'LDIER'!" (14.Ausgabe, Juli 86, S.12 Mitte)!

Recht so! Nur daß dieser Mist-Ake nicht auf meinem Assembler-Mist gewachsen war, sondern an meinem allzu flüchtigen Umgang mit TSCRIFS lag! Hatte ich doch versehentlich statt des Steuerzeichens für Fettdruck ($E) ein # getippt – und so blieb hier ein unbeabsichtigtes "E" stehen.

Doch dies ist beileibe keine Entschuldigung!
Wer was schreibt, sollte es mindestens ein-, besser dreimal (3x) auf Fehler durchlesen, bevor er es veröffentlicht!. Hier tat ich (und noch jemand...!) es nur je einmal; und wir erwiesen uns als "betriebsblind".

Eine größere (fast nicht mehr "läßliche") Sünde war es, euch falsch über die Plazierung von Definitionen und Gleichsetzungen zu belehren! Ich danke an dieser Stelle nochmals *Arnulf SOPP*, der sich als einziger zu meinem Beitrag meldete und auf diesen Kardinalfehler hinwies!

Ich hatte bereits hervorgehoben, daß man beim Assembler zwischen "Anweisungen" und "Befehlen" zu unterscheiden hat. In meiner Rekrutenzeit (nicht beim "Bund", sondern im "3E8h-jährigen Reich") gab es da keinen Unterschied – aber man lernt ja nicht aus. *

Der Unterschied ist, daß Definitionen wie 'DEFM' (DEFine Message), 'DEFB' (DEFine Byte), 'DEFS' (DEFine Storage) usw. keine Befehle an die CPU, sondern nur Anweisungen an den Assembler sind, die er beim Assembliervorgang befolgen soll und die nachher im fertigen Assemblat (die Leute nennen es mal den "OPCODE" .= OPeration Code und mal den OBJECT-CODE – warum soll ich dann nicht einen dritten wohlklingenden Namen, der eindeutiger ist, dafür erfinden??) gar nicht mehr in Erscheinung treten; sie haben nämlich heimlich bewirkt, daß der Code bereits alle Zutaten enthält, die ihn der CPU schmackhaft machen. * * Das gleiche (doch! Heute schreibt man's klein: ich hab nachgesehen) gilt für die Gleichsetzungen mit 'EQU'.

Weil diese "Anweisungen" (die Bezeichnung "Definitionen" wäre m.E. logischer!) die CPU nichts angehen, ja für sie nach der Assemblierung überflüssig geworden sind, haben sie mitten im Programm, das sie durchlaufen soll, nichts zu suchen!! Also bitte:
Gleich an den Anfang des Source-Codes damit! Und zwar vor 'ORG'!
(Oder notfalls auch an das Ende, oder teils vor, teils hinter – aber nie mitten hinein!)

Wer bis hierher mitgedacht hat, sollte fragen: "Wieso hat denn das Programm überhaupt funktioniert?" (Denn alle haben es ja ausprobiert: keiner hat sich beklagt...)

Um das herauszufinden, habe ich die Bytes, die in den DEF... und EQU-Zeilen stehen, mal so betrachtet, wie die CPU sie sieht: nämlich als "Befehle", indem ich diese einer Disassemblertabelle entnahm (eine "reziproke Mnemonics-Sammlung"). Siehe da: dabei stellte sich heraus, daß es sich im wesentlichen um bedingte Sprungbefehle handelte, wobei deren Bedingung nie erfüllt war, so daß der PC (ihr wißt ja: PC = le Postal Coureur oder Pfadfinder, der der CPU jeden nächsten Schritt zeigt!) fröhlich mir-nichts--dir-nichts drüber hinwegging! * G.S.C. = Glücks-Sache CeJot?? Nein: Genialer Source Code!! – Zur Nachahmung nicht empfohlen!

Diese Aufmerksamkeitstests werden nicht die letzten gewesen sein...

Bis auf ein neues! Euer xatot

Mein 2.Assemblerprogramm !

MA + MA-LA = MU

Wer kennt "Das Große LaLuLa" ?
Lest Christian Morgenstern!
Er assemblierte schon vor 80 Jahren unsere Hochsprache ("GINGGANZ") und stellte im LaLuLa die größte Mnemonics-Sammlung seiner Zeit zusammen ("BIFZI BAFZI ...")
Im Gegensatz zum Hochdeutsch von C.M.'s LaLuLa kann K.M.'s MaMaLaMu nur Englisch interpretiert werden:
MAthematics + MAchine-LAnguage = MUsic

Diesmal also nach dem Augenschmaus in der letzten Ausgabe einen Ohrenschmaus!
* ZILOG = "Zauberei In LOGik" macht's möglich. *

Zuvor die unvermeidliche Theorie:

Wird der Verstärker mit schwankenden Spannungen beschickt, so gerät die Membran ins Wackeln. Sind die Schwankungen regelmäßig, so wackelt auch die Membran gleichmäßig, d.h. die Abstände Höchstpunkt-Ruhepunkt-Tiefstpunkt sind zeitlich konstant (je nach Flächen-Trägheitsmoment ihres Querschnittes auch geometrisch, d.h. die Schwingungen sind symmetrisch). Dann spricht man nicht mehr von Wackeln (= diffuse Geräusche), sondern von Schwingungen. Die sind mal "schön" (konsonant), mal "häßlich" (dissonant). Aber das ist bereits Geschmackssache. PENDERECKI (polnischer Komponist, *1933) ist selten "schön", aber stets hinreißend! (Dies trifft übrigens besser als das Allerweltswort "interessant".)

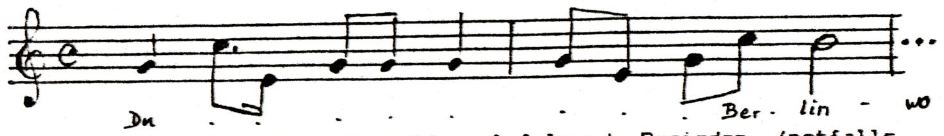
Schwingungen in Sinusform, die sich in ganzzahligen Frequenz-Verhältnissen überlagern - teils harmonisch, teils hin- oder zerreißend - können wir unserem Tasteninstrument zwar nicht entlocken. Aber so scharf hören wir nicht hin: die schiefen Rechtecksignale, die wir dem Verstärker aufzwingen, genügen uns voll und ganz, um uns zu bestätigen, daß wir "erfolgreiche Assemblierer" sind, und seien wir noch so (un)mu-, dafür aber um so mehr phy-sikalisch!

Als "Eichwert" gilt der Kammerton 'a', heute mit 440 Schwingungen pro Sekunde (440Hz). Jeden Halbtonschritt darüber bzw. darunter erhalten wir durch Multiplikation mit bzw. Division durch $\sqrt[12]{2}$ (zwölfte Wurzel aus zwei). Nach zwölf Halbtönen erhalten wir wegen $(\sqrt[12]{2})^{12}$ den Faktor 2, das ist wieder ein 'a', diesmal genau eine Oktave höher.

Mit diesem Wissen können wir Das Große LaLuLa leicht starten!
Einen Taschenrechner, der beliebige Wurzeln ziehen kann, hat wohl jeder; andernfalls hilft die gute alte Logarithmentafel (notfalls ein Dentist). Auch hat jeder, der ins Assemblerreich aufbrechen will, ein gutes Z80-MaLa-Book (inzwischen ist wohl klar, wie man Machine Language abkürzt...), in dem die Taktzyklen und -zeiten stehen, die jeder Z80-Befehl bei der Ausführung benötigt.

Wenn wir die Taktzeiten aller Einzelbefehle, die die Membran zum Hin- und Herschwingen veranlassen, zusammenzählen, hängt das Ergebnis natürlich von der Struktur ab, die wir dem Programm gaben. Hier kommt es nicht auf elegante Kürze an, sondern nur darauf, daß die Summe der einzelnen Taktzeiten möglichst genau mit der "Periode" der gewünschten Tonhöhe übereinstimmt (absolut genaue Übereinstimmung ist übrigens physikalisch unmöglich, weil HEX keine Dezimalzahlen kennt (Widerspruch erbeten!)) Die Periode ist (Mus-, Elektr-, Stat- und andere -iker wissen das) der Reziprokwert der Frequenz:
Periode T in µs = 1/(Frequenz in MHz)
(1MHz = 1 Mega-Hertz = eine Million Schwingungen in der Sekunde)

Da der Kammerton 'a' ---> 0,00044 Schwingungen/µs macht, ergeben sich für die Töne, die wir beispielsweise für diese Melodie



brauchen, gemäß der Wurzelformel folgende Perioden (notfalls sagt uns das auch ein gutes Physikbuch, z.B. der GRIMSEHL, Band 1 Mechanik, Kapitel Akustik (nicht mit dem Akku-Register A der CPU zu verwechseln)):

Ton	Frequenz f (Hz)	Periode T (µs)	C (Hex)
e	329.63	3033.70	B3
g	392.00	2551.02	96
h	493.88	2024.78	76
c	523.25	1911.13	6F

Die letzte Spalte nenne ich "Frequenz-Index" C; sie ergibt sich aus der einfachen Formel $C = (T - 69)/16.5$, die ich aus meinen Programmzeilen 170 bis 230 plus 350 bis 380 errechnet habe (s.Anhang). Diesen (vom Programm abhängigen) Frequenz-Index laden wir später ins C-Register.

Nun charakterisiert jeden Ton nicht nur eine Schwingungsdauer (die wir ihrer Kürze wegen gar nicht hören können), sondern zwecks Hörbarkeit auch eine Verweildauer B, während welcher wir ihn hören. Deshalb bedienen wir uns einer Schleife (Zeilen 170 bis 230), in der das Frequenz-Unterprogramm (Zeilen 350 bis 380) immer wieder abgearbeitet wird. Diese Verweildauer laden wir vor dem Schwingungsteil, der von C umfaßt wird, ins Register B (Zeile 150).

Warum verwende ich B und C? Ich hatte einen Hintergedanken, der mir zunächst frech und gewagt erschien (typisch Verfasser). Ich war dann selbst verwundert, daß es klappte. Ja: Zauberer-80 lehrt uns noch wundern...

Die Idee: Wenn ich B und C verwende, kann ich sie vielleicht als Doppelregister (16Bit-Register) behandeln und mit der "Anweisung" (Mathematiker sagen stattdessen "Erklärung" oder "Definition"; "Anweisung" ist für den Anfänger irreführend, weil er dies leicht mit "auszuführendem Befehl" verwechselt. So schlecht sind MaLa-"Lehrbücher"!) - also mit der Erklärung
DEFW (=DEFine Word)
als 2-Byte-Kombination in den Speicher laden, aber jedes Byte einzeln wieder herausholen, siehe Zeilen 20 bis 120 (im Gänsemarsch in Speicherzellen schreiben). In Zeile 150 wird dann zuerst die Tondauer (Reg. B) und in Zeile 350 danach der Frequenz-Index (Reg. C) wieder aus dem Gedächtnis (memory) geholt.

Diese Tonhöhen und -längen sind in den Zeilen 20 bis 120 als 16 Bit lange "Wörter" abgelegt, b e v o r das eigentliche Musikprogramm in Zeile 140 (hier in Speicheradresse 8017h) beginnt. Das niederwertige Byte, das stets <u>zuerst</u> abgelegt und auch wieder herausgeholt wird (siehe 2.Spalte = Operation Code, kurz "OPCODE" genannt), entspricht der Tondauer, das höherwertige der Tonhöhe (Frequ.)

In Zeile 260 wird das A-Register jedesmal, wenn es sich das dritte, fünfte, siebente usw. Byte für die Tondauer geholt hat, gefragt, ob die "Melodie zu Ende" ist. Dies ist der Fall, wenn das Nullbyte gemäß der Definiton "DEFB" von Zeile 130 erreicht, d.h. wenn A=0 ist. Das Mnemonic, das diese Frage stellt, lautet: CP 00h, was soviel bedeutet wie: Vergleiche den Inhalt von A mit Null! Faktisch wird nicht "gefragt", sondern die Differenz zwischen dem Akku-Inhalt und dem Argument von CP gebildet; bei Gleichheit ist sie natürlich Null. Sobald dies eingetreten ist, springt das Programm in der nächsten Zeile ins DOS zurück: "JP Z,-->dorthin", das ist ein <u>bedingter</u> Sprung, der nur dann ausgeführt wird, wenn das Z-Bit im "Flag-Register" "gesetzt", d.h. gleich Eins ist und somit anzeigt, daß das letzte Ergebnis im Akku eine Null war! Ist das nicht der Fall (weil die Melodie noch weiter geht), so wird der Sprungbefehl übersprungen und der Programmzähler PC macht gehorsam im laufenden Programm weiter.

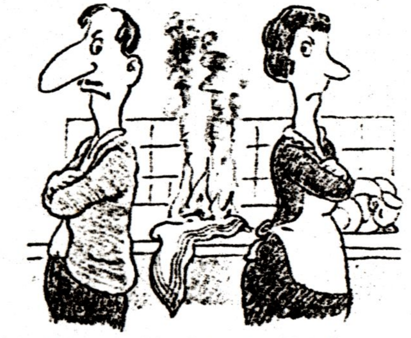
Die Übergabe der so definierten Parameter an den Verstärker erfolgt über Port 255d = 0FFh durch einen "OUT"-Befehl (Zeilen 160 bzw. 210), der abwechselnd eine Eins oder eine Null an ihn schickt, wobei die Eins ihn auf ein Potential von 0.85V (hoher Wert) und die Null ihn auf ein solches von 0.46V (mittlerer Wert) legt. Über die beiden Bits 0 und 1 des Port(s) kann natürlich auch eine Zwei (10h) gesandt werden, die ihn an Masse legt und damit die Schwingungsamplitude (Lautstärke) mehr als verdoppelt; doch wir wollen nichts übertreiben, schließlich sitzt vor dem Lautsprecher normalerweise noch ein Poti.

Nachdem wir mit diesen Behauptungen uns aber bereits verdammt nahe an Hardware-Belange herangetastet haben, mit denen ich mit Rücksicht auf die Umwelt wegen überhöhter Zinn-Verschmutzung absolut nichts im Sinn habe, wollen wir uns edleren Dingen zuwenden und uns die Musik anhören, die wir da erzapft haben! (Nein, das "v" wurde nicht vergessen. Wer will, kann apf gegen eug eintauschen.)

Ich sagte schon, daß ich selbst über den halbwegs gelungenen Effekt überrascht war; stand es doch wieder mal in keinem "Leerbuch", daß man ein 16-Bit-Wort (DEFW) wieder zerreißen und seine beiden Hälften einzeln und unabhängig voneinander als vollgültige Steuercodes, hier als kusimalische Komponenten (Frequenz&Dauer), einsetzen könne! Im allgemeinen entsprechen zwei halbe Fliegen eher einer toten Fliege; hier aber haben wir zwei vollwertige Brummer nach der Teilung! * Jedoch:

Wie man sieht, enthält die Melodie (für Nichtschwimmer, die sich nicht ins tiefe ASS-Becken trauen, habe ich sie in klassischer Mensural-Notation bereits verraten - er summe sie vor sich hin; aber bitte con sordino, um niemand zu kränken...) viermal das g hintereinander. In meiner ersten Version erwies sich dies als ein <u>einziger</u> Ton mit der Länge von zweieinhalb Taktschlägen (ein Viertel im Viervierteltakt - durch c gekennzeichnet - entspricht einem Taktschlag; 3 Achtel + 1 Viertel sind also 2,5 Taktschläge); d.h. der gewünschte Rhythmus kam nicht zustande, weil die einzelnen frequenzgleichen Töne ineinander

Französisch: Le torchon brûle entre eux.
Das Spültuch brennt zwischen ihnen.



Der Haussegen hängt schief.

Amerikanisch: That's the way the cookie crumbles.
Wie ein Keks halt so krümelt.
Wie das Leben halt so spielt.
Markus Hubers, Gescher

verschwammen. Also mußten sowas wie "Zwischenräume" her. Zu diesem Behufe fädelte ich ab Zeile 280 bis 330 eine kleine Trennschleife ein. Die Werte für die Halteregister D und E wurden einfach ausprobiert.

A propos:

Das Salz beim ASSEMBLIEREN
ist grad' das AUSPROBIEREN!

Wenn's dann mal klappt, freut man sich, der Frust schwindet, das Erfolgserlebnis ist da und man kann's vergessen und neuen Ufern (= Untaten) zustreben... Hierin weiß ich mich eins mit größ'ren Geistern.

Übrigens: Wenn man glaubt, man könne statt D=0FFh plus E=0Ah gleich das Doppelregister DE mit 0FF0Ah laden, "so irrt sich der" und fällt beim Versuch, die Sache dann mit "DEC DE" in Gang zu bringen, wie Busch's Frosch vom Baum - er stürzt ab. Warum, weiß ich bis heute nicht. Welch ein Widerspruch zu der freudigen Entdeckung der Leistungsfähigkeit des Doppelregisters BC, siehe oben!! Vielleicht verrät es mal ein Experte, der an den Fortschritten der Halbassemblierer interessiert ist!(???)

Der Frequenz-Index C bestimmt übrigens, genauer als oben gesagt, nicht die ganze, sondern nur die halbe Periode (Zeile 360, Label 'HLBPER'), weil ja genau nach jeder halben Periode die Portbelegung, sprich das Ausgangspotential von Hoch nach Tief vice versa geändert werden muß.

Damit wären wir am Ende des Programmchens. Entstanden ist weder ein Beethoven-Plagiat noch eine Tanz-Orgie, sondern nur der Anfang eines Gassenhauers. <u>Wer kennt seinen Text?</u> Einstmals kleene Balina ha'k'n ooch ümma jefiff'n!
(Aba det 'J' in KaJott is keen verstümmeltet 'G', gloobt man blos nich!!)

Nun komponiert die bekannte Melodie mal zu Ende!

KaJott

MA + MA − LA = MU

Dauer Frequ.

```
8000            00010          ORG     8000H
8000 7F96       00020 ANF      DEFW    967FH    ; VIERTELNOTE G
8002 5D6F       00030          DEFW    6F5DH    ; PUNKTIERTE ACHTEL C
8004 1FB3       00040          DEFW    0B31FH   ; SECHZEHNTEL E
8006 3F96       00050          DEFW    963FH    ; ACHTEL G
8008 3F96       00060          DEFW    963FH    ;   DTO.
800A 7F96       00070          DEFW    967FH    ; VIERTELNOTE G
800C 3F96       00080          DEFW    963FH    ; ACHTELNOTE G
800E 3FB3       00090          DEFW    0B33FH   ; ACHTELNOTE E
8010 3F96       00100          DEFW    963FH    ; ACHTELNOTE G
8012 3F6F       00110          DEFW    6F3FH    ; ACHTELNOTE C
8014 FF76       00120          DEFW    76FFH    ; HALBE NOTE H
8016 00         00130          DEFB    00H      ; SCHLUSS DER MELODIE.
8017 210080     00140 START    LD      HL,ANF   ; ZEIGER AUF ERSTE NOTE.
801A 46         00150 TON      LD      B,(HL)   ; TONDAUER.
801B 23         00160          INC     HL       ; ZEIGER AUF FREQUENZ-INDEX.
801C 3E01       00170 DAUER    LD      A,01H    ; ERSTE HALBPERIODE.
801E D3FF       00180          OUT     (0FFH),A; 0,85V AUF VERSTAERKERPORT.
8020 CD4080     00190          CALL    FREQU    ; ZUM UP FREQUENZGEBER.
8023 3E00       00200          LD      A,00H    ; ZWEITE HALBPERIODE.
8025 D3FF       00210          OUT     (0FFH),A; 0,46V AUF VERSTAERKERPORT.
8027 CD4080     00220          CALL    FREQU    ; ZUM UP FREQUENZGEBER.
802A 10F0       00230          DJNZ    DAUER    ; WIEDERHOLUNG DER PERIODE,
                00235                           ; SOLANGE TON ANDAUERT.
802C 23         00240          INC     HL       ; ZUM NAECHSTEN TONLAENGEN-
                00245                           ; BYTE.
802D 7E         00250          LD      A,(HL)   ; PRUEFUNG, OB MELODIE
802E FE00       00260          CP      00H      ; ZU ENDE IST;
8030 CA2D40     00270          JP      Z,402DH  ; WENN JA ===> DOS,
8033 1E0A       00280          LD      E,0AH    ; SONST TRENNUNG (ZAESUR)
8035 16FF       00290 TRENN2   LD      D,0FFH   ; ZWISCHEN DEN TOENEN,
8037 15         00300 TRENN1   DEC     D        ; DA GLEICHE TOENE SONST
8038 20FD       00310          JR      NZ,TRENN1;       NICHT
803A 1D         00320          DEC     E        ; ZU UNTERSCHEIDEN
803B 20F8       00330          JR      NZ,TRENN2;       SIND.
803D C31A80     00340          JP      TON      ; ZUM NAECHSTEN TON.
8040 4E         00350 FREQU    LD      C,(HL)   ; UNTERPROGRAMM
8041 0D         00360 HLBPER   DEC     C        ; ZUR FESTLEGUNG DER
8042 C24180     00370          JP      NZ,HLBPER; FREQUENZ (TONHOEHE)
                00375                           ; DURCH DEN BERECHNETEN
                00376                           ; 'FREQUENZ-INDEX'.
8045 C9         00380          RET              ; ZUR NAECHSTEN HALBPERIODE.
8017            00390          END     START
00000 TOTAL ERRORS
33635  TEXT AREA BYTES LEFT

ANF     8000 00020    00140
DAUER   801C 00170    00230
FREQU   8040 00350    00190 00220
HLBPER  8041 00360    00370
START   8017 00140    00390
TON     801A 00150    00340
TRENN1  8037 00300    00310
TRENN2  8035 00290    00330
```

Anm. Leider sendet EDTASM [mein] keine Minuskeln; es sähe besser aus!

A N H A N G
===========

Berechnung der Frequenz-Indices

Die Dauer einer Schwingung – das ist die reziproke Tonfrequenz – entspricht der Zeit, die die CPU benötigt, um je eine positive und eine negative Halbschwingung zu erzeugen, indem sie abwechselnd ein höheres und ein niederes Potential über den Port 255 zum Verstärker schickt. Dies macht sie, indem sie der Befehlsfolge ab Label "DAUER"*) in Zeile 170 bis zur Dekrementierung in Zeile 230 gehorcht, wobei <u>zweimal</u> ein CALL zum Unterprogramm für den Frequenzindex C in Zeile 350 bis 380 durchgeführt wird. Wir müssen also die Taktzeiten aller Befehle, die hierbei anfallen, zusammenzählen.

Hierzu stellen wir uns zunächst die Anzahl Taktzyklen zusammen, die die Befehle machen, die in den vorgenannten Zeilen vorkommen:

Mnemonic	Anz.Taktzyklen
LD A,n	7
OUT n,A	11
CALL nn	17
DJNZ e (Sprungweite)	13 (solange B<>0)
DEC r (Register)	4
JP NZ,nn	10
RET	10

Zählt man die für jede der oben genannten Zeilen jeweils gültigen Zahlen zusammen – wobei die Zeilen 350 und 380 doppelt und die beiden Zeilen 360 und 370 sogar 2Cmal**) zu zählen sind, s.o. – so erhält man für die Anzahl Taktzyklen die Gleichung: TZ = 117 + 28C.**) Da (bei meinem alten Z80) ein Taktzyklus ziemlich genau 0.59 Mikrosekunden dauert, (was von mir am 1.4.0815 mit einer Stoppuhr exakt gemessen wurde,) dauert die Abarbeitung der Befehlsfolge für die Tonperiode (indirekt also für die Frequenz) T = 0.59(117+28C) µs, woraus die im Beitrag erwähnte Formel für den "Frequenz-Index" C folgt. Da in Tabellenwerken i.d.R. nicht die Periodendauern, sondern die Frequenzen der Tonhöhen in Hz aufgeführt sind, erhält man die richtige Größenordnung der Periode, indem man den Kehrwert der in MHz verwandelten Tabellenwerte in die Formel einsetzt.

KB

*) „DAUER" = Schwingungsdauer (nicht Tondauer!)

**) Wie (Mathe-) üblich, wurde der Mult.-Operator fortgelassen: „2Cmal" bedeutet also „2·C-mal"; „28C" bedeutet „28·C".

DIR/SYS oder INHALT/SYS?

WEGE ZUR GDOS/NEWDOS-KOMPATIBILITÄT

Ein zunächst kaum ins Auge fallender Unterschied zwischen NEWDOS-80V.2 und GDOS besteht in der Benennung des Directory-Files. Formatiert man eine Diskette mit NEWDOS, so heißen die dabei kreierten SYS-Files BOOT/SYS und DIR/SYS. Tut man dasselbe mit GDOS, so heißen diese beiden Files GDOS/SYS und INHALT/SYS. Da man auch in GDOS das Directory mit DIR (oder mit I) aufrufen kann, fällt das wie gesagt auch dem Umsteiger zunächst kaum auf. Im Gegenteil, man empfindet es als Umsteiger nach kurzer Gewöhnung als großen Vorteil, das Directory statt mit drei Tastenanschlägen (DIR) mit nur einem (I) aufrufen zu können.

Trotzdem stellen sich nach einiger Zeit gewisse Nachteile heraus. Es gibt eine große Anzahl Programme, die auf den DIR/SYS-File zugreifen, z.B. Programme, die die Directory-Einträge (ohne die SYS-Files) alphabetisch sortieren und sortiert auf die Diskette zurückschreiben, eine besonders bei DD und/oder 80-Track-Laufwerken mit bis zu ca. 200 Einträgen fast unentbehrliche (max. 15 x 4 = 60 Einträge auf einer Bildschirmseite) Utility. Auch die Katalogisierungsprogramme greifen auf DIR/SYS als File zu. Heißt nun aber der Directory-File INHALT/SYS, während das Programm auf DIR/SYS zugreifen will (und umgekehrt), so gibt es eine Fehlermeldung „Datei nicht im Inhaltsverzeichnis" (bzw. "FILE NOT IN DIRECTORY"): Inkompatibilität zwischen NEWDOS und GDOS.

Natürlich ist eine Abhilfe relativ einfach möglich, indem man den File je nachdem in DIR/SYS bzw. INHALT/SYS (mit RENAME bzw. N - auch eine der so angenehmen Befehlsabkürzungen im GDOS, diesmal gleich fünf Tastenanschläge weniger!) - umbenennt.

Weit eleganter wäre es natürlich, die GDOS-Systemdiskette durch einen Patch zu veranlassen, von vornherein beim Formatieren den Directory-File gleich DIR/SYS statt INHALT/SYS zu benennen. Der GDOS-„Erfinder" J.H.L.Heicke gibt als Grund für die Benennung des Inhalt-Files mit INHALT/SYS statt DIR/SYS an: „Diese Namensänderung wurde wie bei GDOS/SYS beschlossen, um eine Zerstörung der abweichenden internen Aufteilung durch direkt zugreifende Programme zu erschweren" (JHL Heicke, GDOS-Systembeschreibung, 1984). Der Zugriff wird, wie gezeigt, tatsächlich erschwert. Eine Zerstörungsgefahr ist mir hingegen noch nicht begegnet, wie z.B. das Programm DIRSORT von Heinfried Cznottka, Braunschweig, beweist: es greift offensichtlich direkt auf den Directory-File zu und schreibt die Einträge alphabetisch sortiert auf die Diskette zurück, aber (im Gegensatz zu anderen Programmen gleichen Namens und gleicher Funktion) unabhängig vom Namen des Directory-Files!

So schaute ich mir den in NEWDOS und GDOS für das Formatieren und Kopieren zuständigen Systemfile SYS6/SYS genauer an. Auch als Nicht-Assemblergewandter findet man leicht im relativen File-Sektor (FRS) 24d/18h beim relativen Sektor-Byte (SRB) 84h die Einträge „GDOS/SYS" und (SRB A4h) „INHALT/SYS". Also gleich mal umgezappt in BOOT/SYS und DIR/SYS (die restlichen drei Bytes werden mit Blanks - 20h - aufgefüllt) und probiert!

Tatsächlich heißen nach dem Formatieren die beiden SYS-Files einer neu formatierten Diskette nun BOOT/SYS und DIR/SYS, wie ein Auflisten mit I,S (bzw. DIR,S) beweist. Alles geschafft - denkt man. Aber leider hat man sich getäuscht! Beim Ausprobieren von Programmen mit einem Zugriff auf DIR/SYS gehen diese weiterhin mit „Datei nicht im Inhaltsverzeichnis" auf Fehler.

Noch schlimmer: auch wenn man im Programm den Namen DIR/SYS in INHALT/SYS umwandelt (was vorher zum Erfolg führte), bleibt das Programm mit der Fehlermeldung „Datei nicht im Inhaltsverzeichnis" hängen. Und selbst wenn man nun mit DDE oder SUPERZAP den File DIR/SYS zu listen versucht, erscheint die gleiche Fehlermeldung.

Der relative Laufwerkssektor (DRS), in welchem das Directory einer Diskette beginnt, ist leicht aus den PDRIVE-Werten dieser Diskette zu errechnen:

NEWDOS GDOS	GPL EIB	x 5 x	DDSL SBIV	=	DRS des Directory-Beginnes
Beispiele:	6	x 5 x	48	=	1440 (für GENIE III bzw. 80-Tr.-Laufwerke DS/DD)
	2	x 5 x	17	=	170 (40 od. 35-Tr.-Laufwerke SS/SD)

GPL = Grans Per Lump = EIB = Einheiten Im Block; 1 Einheit (Gran) = 5 Sektoren in NEWDOS und GDOS.

DDSL = Disk Directory Start Lump = SBIV = StartBlock des InhaltsVerzeichnisses.

In der normalen PDRIVE-Einstellung beginnt also bei 80-Track Laufwerken Double Sided mit Double Density der Directory-File im relativen Drive-Sektor (DRS) 1400d/5A0h. Also mit SUPERZAP mal nachgesehen, was auf DRS 1440d steht: der Directory-Beginn! Im DRS 1443d und 1444d stehen die File-Einträge für BOOT/SYS und DIR/SYS. Sie scheinen zu stimmen.

SUPERZAP erlaubt mit dem Kommando DNTH (Display Name/Type Hashcode) die Berechnung des Hashcodes für jeden Filenamen. Dabei ergibt sich

A2 für BOOT/SYS
A1 für GDOS/SYS
C4 für DIR/SYS
CE für INHALT/SYS.

Überprüft man nun die Hashcodes im DRS 1441d/5A1h, SRB 00 für BOOT/SYS und SRB 01 für DIR/SYS, so stellt man fest, daß hier weiterhin die Codes A1 und CE für GDOS/SYS und INHALT/SYS stehen, obwohl die die Filenamen im DRS 1403d/5A3h und 1404d/5A4h entsprechend der gemachten Änderung in SYS6/SYS in BOOT/SYS und DIR/SYS umgeändert sind.

Das kann nichts anderes bedeuten, als daß durch diesen Systemzap zwar die Filenamen geändert sind, nicht aber die zugehörigen Hashcodes, so daß das System beim Aufruf der Filenamen die entsprechenden Files BOOT/SYS und DIR/SYS nicht finden kann, weil der entsprechende Hashcode in DRS 1441d nicht zu den geänderten Namen paßt. Im Gegensatz zur Benennung eines Files mittels CREATE oder RENAME (bzw. N) aus dem DOS oder OPEN aus dem BASIC wird also der Hashcode beim Formatieren nicht nach dem im FRS 24d/18h von SYS6/SYS festgelegten Namen errechnet, sondern er muß an einer anderen Stelle festgelegt sein und danach eingetragen werden. ABER WO ???

Mit SUPERZAP habe ich den GDOS-File SYS6/SYS nach dem Byte CE (= Hashcode für INHALT/SYS) durchsucht und dabei 19maliges Vorkommen festgestellt und notiert. Beim Durchsuchen des SYS6/SYS-Files von NEWDOS nach CE fehlte CE gegenüber dem entsprechenden GDOS-File im FRS 25d/19h bei SRB 98h; es hieß hier C4, das ist der Hashcode für DIR/SYS! Auf die gleiche Weise fand sich im gleichen FRS 25d/19h dicht daneben, bei SRB 95h, A1 (Hashcode für GDOS/SYS) im GDOS-File und A2 (Hashcode für BOOT/SYS) im NEWDOS-File.

Nach entsprechendem Umzappen ergab die Probe endlich das gewünschte Ergebnis: die Formatierung zeigt die umgeänderten

HEFT 15 September 1986

Filenamen BOOT/SYS und DIR/SYS mit den richtigen Hashcodes A2 und C4.

Hier noch einmal die erforderlichen Zaps im GDOS-File SYS6/SYS. Ich hielt es für angebracht, den Namen des BOOT/SYS-Files weiterhin bei GDOS/SYS zu belassen, da das nicht stört und doch einen gewissen Beleg auf jeder mit GDOS formatierten Diskette darstellt.

> FRS 25d/19h, RSB 95h von 36 A1 23 nach 36 A2 23 gezappt ergibt den Hashcode für BOOT/SYS beim Formatieren (bei mir nicht geändert).
>
> FRS 25d/19h, RSB 98h von 36 CE 3A nach 36 C4 3A gezappt ergibt den Hashcode für DIR/SYS beim Formatieren.

Die File-Namen müssen entsprechend dem eingesetzten Hashcode in FRS 25d/19h im Sektor davor, also FRS 24d/18h beginnend bei SRB 84h von GDOS in BOOT (wenn man will) und bei SRB A4h von INHALT in DIR geändert werden, wobei die SRB A7h bis A9h mit Blanks (20h) aufgefüllt werden. Die beiden Sektoren sehen also so aus:

SYS6/SYS, FRS 18h
GDOS original:

```
DRV 00  3701 0051 CDD7 427E E683 E281 421A 0203 7..Q..Bß....B...
1   10  CB4E C287 42CB 4EC2 8742 CB4E 20EF CB46 .N..B.N..B.N...F
1H  20  2808 CB4E 20E7 CB7E 28E6 7E36 D0C1 D1E6 (..N...ß(.ß6....
    30  FC20 0C1C 7BD6 0020 0314 1E00 D97E C9CD ....ä.........ß..
DRS 40  D742 360B 1098 21DD 427E FE03 28FB 23CD .B6...!.Bß..(.#.
979 50  3300 18F5 CDD7 42CB 4620 FC7E C93E 063D 3.....B.F..ß.>.=
3D3H60  20FD C91C 1F52 4553 4554 031C 1F01 09A1 .....RESET......
    70  6547 444F 5320 3F03 0100 B765 0000 005E eGDOS.?....e...^
    80  0000 0000 4744 4F53 2020 2020 5359 5360 ....GDOS....SYS`
    90  7F1F B205 0000 00FF FFFF FFFF FFFF FF5D...............ü
    A0  0000 0000 494E 4841 4C54 2020 5359 53A7 ....INHALT..SYS.
    B0  1DF9 E51E 0030 05FF FFFF FFFF FFFF FFCD .....0..........
FRS C0  9263 D218 52CD A063 E521 9759 CBD6 E1CD .c..R..c.!.Y....
24  D0  B56E DC95 4FCD B86F CDB5 6E30 07CD 2550 .n..O..o..nO..%P
18H E0  ED53 8159 0640 CDA7 4E21 9459 7EE6 F920 .S.Y.§..N!.Yß...
    F0  047E F680 77CD B363 21F3 6DCD 6744 CD10 .ß..w..c!.m.gD..
```

GDOS gezappt:

```
DRV 00  3701 0051 CDD7 427E E683 E281 421A 0203 7..Q..Bß....B...
0   10  CB4E C287 42CB 4EC2 8742 CB4E 20EF CB46 .N..B.N..B.N...F
0H  20  2808 CB4E 20E7 CB7E 28E6 7E36 D0C1 D1E6 (..N...ß(.ß6....
    30  FC20 0C1C 7BD6 0020 0314 1E00 D97E C9CD ....ä.........ß..
DRS 40  D742 360B 1098 21DD 427E FE03 28FB 23CD .B6...!.Bß..(.#.
151950  3300 18F5 CDD7 42CB 4620 FC7E C93E 063D 3.....B.F..ß.>.=
5EFH60  20FD C91C 1F52 4553 4554 031C 1F01 09A1 .....RESET......
    70  6547 444F 5320 3F03 0100 B765 0000 005E eGDOS.?....e...^
    80  0000 0000 4744 4F53 2020 2020 5359 5360 ....GDOS....SYS`
    90  7F1F B205 0000 00FF FFFF FFFF FFFF FF5D...............ü
    A0  0000 0000 4449 5220 2020 2020 5359 53A7 ....DIR.....SYS.
    B0  1DF9 E51E 0030 05FF FFFF FFFF FFFF FFCD .....0..........
FRS C0  9263 D218 52CD A063 E521 9759 CBD6 E1CD .c..R..c.!.Y....
24  D0  B56E DC95 4FCD B86F CDB5 6E30 07CD 2550 .n..O..o..nO..%P
18H E0  ED53 8159 0640 CDA7 4E21 9459 7EE6 F920 .S.Y.§..N!.Yß...
    F0  047E F680 77CD B363 21F3 6DCD 6744 CD10 .ß..w..c!.m.gD..
```


FRS 19h
GDOS original:

```
DRV 00  67CD F34D 212D 55E5 CDC3 673A 9759 CB5F g..M!-U...g:.Y._
1   10  C03A 9659 CB7F C011 0000 CD84 573A 9459 .:.Y......W:.Y
1H  20  CB4F C021 DA6D CD67 44CD AA67 11B5 6E01 .O.!.m.gD..g..n.
    30  0200 CCB4 67C2 1A52 3EFF CDB0 64ED 5BD1 ....g..R>...d.Ä.
DRS 40  5921 0042 221A 5B4D E5CD 6257 E1CB C63A Y!.B".ÄM..bW...:
980 50  CD59 32C5 592A 9943 7706 005F 3ACE 594F .Y2.Y*.Cw.._:.YO
3D4H60  D521 0042 CD1C 6221 7E59 11CB 4201 1600 .!.B..b!ßY..B...
    70  EDB0 E13A CA59 6707 0784 0100 B566 CD92 ...:.Yg......f..
    80  4C22 215B 0E00 183C C5AF CDB0 6479 3D20 L"!Ä...<....dy=.
    90  1E21 0042 36A1 2336 CE3A CE59 D602 4F07 .!.B6.#6.:.Y..O.
    A0  0781 321F 42C6 0A32 EE65 320C 6718 1421 ..2.B..2.e2.g..!
    B0  BA65 FE02 3805 200B 21DA 6511 0042 0120 .e..8...!.e..B..
FRS C0  00ED B0C1 DDCB 00C6 CDD4 57DD CB00 86C2 ...........W.....
25  D0  1A52 0C79 FE0A 38B0 C921 C559 DDE5 E5DD .R.y..8..!.Y....
19H E0  E1DD 7E04 DD6E 03CD 924C DD75 0ADD 740B ..ß..n...L.u..t.
    F0  CDF2 5CDD 750C DD74 0DDD 7E05 CDB4 4CB7 ..ö.u..t..ß...L.
```

GDOS gezappt:

```
DRV 00  67CD F34D 212D 55E5 CDC3 673A 9759 CB5F g..M!-U...g:.Y._
0   10  C03A 9659 CB7F C011 0000 CD84 573A 9459 .:.Y......W:.Y
0H  20  CB4F C021 DA6D CD67 44CD AA67 11B5 6E01 .O.!.m.gD..g..n.
    30  0200 CCB4 67C2 1A52 3EFF CDB0 64ED 5BD1 ....g..R>...d.Ä.
DRS 40  5921 0042 221A 5B4D E5CD 6257 E1CB C63A Y!.B".ÄM..bW...:
152050  CD59 32C5 592A 9943 7706 005F 3ACE 594F .Y2.Y*.Cw.._:.YO
5F0H60  D521 0042 CD1C 6221 7E59 11CB 4201 1600 .!.B..b!ßY..B...
    70  EDB0 E13A CA59 6707 0784 0100 B566 CD92 ...:.Yg......f..
    80  4C22 215B 0E00 183C C5AF CDB0 6479 3D20 L"!Ä...<....dy=.
    90  1E21 0042 36A1 2336 C43A CE59 D602 4F07 .!.B6.#6.:.Y..O.
    A0  0781 321F 42C6 0A32 EE65 320C 6718 1421 ..2.B..2.e2.g..!
    B0  BA65 FE02 3805 200B 21DA 6511 0042 0120 .e..8...!.e..B..
FRS C0  00ED B0C1 DDCB 00C6 CDD4 57DD CB00 86C2 ...........W.....
25  D0  1A52 0C79 FE0A 38B0 C921 C559 DDE5 E5DD .R.y..8..!.Y....
19H E0  E1DD 7E04 DD6E 03CD 924C DD75 0ADD 740B ..ß..n...L.u..t.
    F0  CDF2 5CDD 750C DD74 0DDD 7E05 CDB4 4CB7 ..ö.u..t..ß...L.
```

Es sei noch bemerkt, daß bei mir auch noch andere GDOS- und NEWDOS-Fassungen existieren, bei denen aus irgendwelchen Gründen die betreffenden Stellen etwas verschoben sind. Anhand der in den Sektorenlistings aufgeführten Zusammenhänge wird jeder in seiner Fassung die entsprechenden Stellen leicht auffinden.

Und nun viel Spaß - noch mehr Spaß mit kompatibleren GDOS- und NEWDOS-Systemen!

Richard Rensch

The Disappearing DOS

Five utilities let you access Model I/III system functions from DOS Ready.

While you can't deny that a disk operating system is a necessity, in some cases it's also a hindrance. For example, if you want to execute a low-level function like modifying memory, initializing a printer, sending data out a particular port, or calling a ROM subroutine, you have to leave DOS, load either Basic or Debug, execute the function, and then return to DOS.

I've written five utilities that break this DOS barrier and let you use almost all your Model I/III capabilities directly from DOS Ready. This is especially useful in writing Build files, where a transparent DOS gives you direct access to system functions.

Typing in the Utilities

The five utilities presented below all use the same shell, which appears in the listing for POKE/SRC (see the Program Listing). Type in and assemble POKE, then write the file to disk. Load POKE/SRC and, for each of the other four utilities, make the changes indicated in the Table. Then assemble and save those four programs to disk with the appropriate file name.

The Five

POKE/SRC modifies up to 20 contiguous bytes of memory with a single command so you can change the cursor character, write to the screen, change case, and so on from DOS Ready.

POKE's format is POKE nnnn,nn, where the first parameter is a four-digit hexadecimal (hex) address and the second is a two- to 40-digit hex data stream. For example, POKE 4023,nn changes the cursor character, with nn the hex code for the new character. POKE 3C00,54455354 displays the word "TEST" in the upper left-hand corner of the display. POKE 4019,00 sets the keyboard for lowercase characters; 01 instead of 00 sets it for uppercase.

LPRINT/CMD, a line printer utility, sends up to 20 control bytes to your line printer. This lets you execute a form feed or change modes on a programmable printer such as an Epson MX-80 without leaving DOS. The utility's format is LPRINT nn, where nn is a two- to 40-digit hex data stream. For example, LPRINT 0C sends a top-of-form control character to your line printer. LPRINT 1B451B47 sends an escape sequence; it sets the Epson MX-80 to emphasized, double-strike print mode.

Out/CMD, a port output utility, sends a data stream of up to 20 bytes to any Z80 port. You can initialize any port-mapped peripheral, such as a universal asynchronous receiver/transmitter (UART) or a Z80 speed-up modification, from DOS Ready. The command's format is OUT nn,nn, where the first parameter is a two-digit hex port address and the second is a two- to 40-digit hex data stream. For example, OUT EC,04 sets your screen to 32-character mode; CLS returns you to 64-character mode. OUT FF,01020102010201020102 generates a beep from the cassette port.

Call/CMD, a machine-language subroutine call utility, executes any machine-language subroutine and then returns to DOS. You can use this to reenter resident programs after a hang-up, to test subroutines, or to execute ROM subroutines. The command's format is CALL nnnn, where nnnn is a four-digit hex address. For example, CALL 0049 stops execution until you press a key, useful as part of a Build file. CALL 01D9 prints the screen contents on a line printer, also useful as part of a Build file.

Execute/CMD, a machine-code execution utility, lets DOS serve as a machine-language interpreter. You pass hex digits as a parameter, and Execute converts them to binary and puts them in a buffer beginning at 5300H. The buffer is padded with No Operations (NOPs) and terminated with a jump to DOS (JP 402DH). Thus, unless the machine-language routine contains a jump external to the buffer, or contains an infinite loop, DOS will regain control after execution. You use this utility to test short routines, move blocks of memory, or pass parameters to subroutines. Execute's format is EXECUTE nn, where nn is a two- to four-digit hex data stream. For example:

EXECUTE 21003C1100F0010004EDB0

moves screen data to the buffer at 0F000H. Following is the Assembly-language code for the above statement:

```
LD   HL,3C00H
LD   DE,0F000H
LD   BC,0400H
LDIR
```

The sequence below turns your TRS-80 into a typewriter:

EXECUTE CD4900CD3300CD3B00C30053

Following is the Assembly-language code for the above statement:

```
CALL  0049H ;ROM INKEY routine
CALL  0033H ;Display character
CALL  003BH ;Line print character
JP    5300H ;Jump to beginning
```

DOS Ready

The five utilities described above let you access almost all the Model I/III features from DOS Ready. If a needed DOS command doesn't exist, you can simulate it with one of these utilities. And if the excessive typing starts to bother you, you can type in Commando (see "Macro Economics," February 1986, p. 66), which lets you rename long, hard-to-remember commands to short, meaningful, logical names. ■

You can reach Craig Chaiken at 32 Beverly Drive, Avon, CT 06001.

LOAD 80

System Requirements

Models I and III
32K RAM
DOS
Editor/assembler
Printer

```
LPRINT/SRC  -  LPRINT/CMD
Add:        412          EXX
Change:     260          JR      NZ,PARAM2
            400          EXX
            410          CALL    3BH
Delete lines 300-380

OUT/SRC     -   OUT/CMD
Change:     400          LD      C,D
            410          OUT     (C),A
Delete:     lines 320, 330

CALL/SRC  -  CALL/CMD
Add:        292          LD      IX,EXIT
            294          PUSH    IX
Change:     340          EX      DE,HL
            350          JP      (HL)
Delete lines 360-420

EXECUTE/SRC  -  EXECUTE/CMD
Add:        930          DEFW  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
            940          DEFW  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
            950          DEFB  0C3H
            960          DEFW  EXIT
            970          END   START
Change:     300 PARAM1   LD     DE,START+100H
            460          JP     Z,START+100H
            570          JP     Z,START+100H
            920          ORG    START+100H
Delete lines 310-370
```

Table. Changes to POKE/SRC for the four other utilities.

Program Listing. POKE/SRC.

```
5200            00160 START   EQU     5200H
402D            00170 EXIT    EQU     402DH
021B            00180 PRINT   EQU     021BH
                00190 ;
5200            00200         ORG     START
                00210 ;
5200 7E         00220 SPACE   LD      A,(HL)          ;FIND FIRST PARAMETER
5201 FE0D       00230         CP      0DH
5203 C8         00240         RET     Z
5204 FE20       00250         CP      ' '
5206 2003       00260         JR      NZ,PARAM1
5208 23         00270         INC     HL
5209 18F5       00280         JR      SPACE
                00290 ;
520B CD2152     00300 PARAM1  CALL    GETBYTE         ;get address
520E 57         00310         LD      D,A
520F CD2152     00320         CALL    GETBYTE
5212 5F         00330         LD      E,A
5213 7E         00340         LD      A,(HL)
5214 FE2C       00350         CP      ','
5216 C26452     00360         JP      NZ,ERROR
5219 23         00370         INC     HL
                00380 ;
521A CD2152     00390 PARAM2  CALL    GETBYTE         ;get data string
521D 12         00400         LD      (DE),A
521E 13         00410         INC     DE
521F 18F9       00420         JR      PARAM2
                00430 ;
5221 7E         00440 GETBYTE LD      A,(HL)
5222 FE0D       00450         CP      0DH
5224 CA2D40     00460         JP      Z,EXIT
5227 CD4252     00470         CALL    LEGAL
522A DA6452     00480         JP      C,ERROR
522D 87         00490         ADD     A,A
522E 87         00500         ADD     A,A
522F 87         00510         ADD     A,A
5230 87         00520         ADD     A,A
5231 4F         00530         LD      C,A
5232 23         00540         INC     HL
5233 7E         00550         LD      A,(HL)
5234 FE0D       00560         CP      0DH
5236 CA2D40     00570         JP      Z,EXIT
5239 CD4252     00580         CALL    LEGAL
523C DA6452     00590         JP      C,ERROR
523F 81         00600         ADD     A,C
5240 23         00610         INC     HL
5241 C9         00620         RET
                00630 ;
5242 FE20       00640 LEGAL   CP      20H
5244 CA2D40     00650         JP      Z,EXIT
5247 D630       00660         SUB     30H
5249 FE00       00670         CP      0
524B 3F         00680         CCF
524C FA6252     00690         JP      M,ILLEGAL
524F FE0A       00700         CP      0AH
5251 3F         00710         CCF
5252 F8         00720         RET     M
5253 D607       00730         SUB     7
5255 FE0A       00740         CP      10
5257 3F         00750         CCF
5258 FA6252     00760         JP      M,ILLEGAL
525B FE10       00770         CP      16
525D 3F         00780         CCF
525E F26252     00790         JP      P,ILLEGAL
5261 C9         00800         RET
                00810 ;
5262 37         00820 ILLEGAL SCF
5263 C9         00830         RET
                00840 ;
5264 216D52     00850 ERROR   LD      HL,MERROR
5267 CD1B02     00860         CALL    PRINT
526A C32D40     00870         JP      EXIT
                00880 ;
526D 0A         00890 MERROR  DEFM    0AH,'*** Command Parameter Error ***'
     2A 2A 2A 20 43 6F 6D 6D
     61 6E 64 20 50 61 72 61
     6D 65 74 65 72 20 45 72
     72 6F 72 20 2A 2A 2A
528D 0A         00900         DEFB    0AH,0DH
     0D
                00910 ;
5200            00920         END     START
```

End

HEFT 15
September
1986

Ein neuer Lib-Befehl: GO<,adresse>

Es ist nicht selten sinnvoll, an eine beliebige Stelle im Speicher
springen zu können. Alle halbwegs ernstzunehmenden Monitoren bieten diese
Möglichkeit an, auch unser Debugger. Dieser ist jedoch in der Anwendung
sehr eingeschränkt: Er läuft nicht ohne Risiko unter Mini-DOS und kann
nur sehr umständlich in JOB-Dateien eingebaut oder auf F-Tasten program-
miert werden. Selbst wenn man es tut, ist jemand an der Tastatur kaum
entbehrlich. Schließlich findet er die Einsprungsadresse einer CMD-Datei
nicht selbsttätig, was das hier vorgestellte Programm kann. Es ist oben-
drein phosphatfrei und grundwasserneutral.

Auf jeden Fall sollte es in eine SYS-Datei gepatcht werden, um alle
Möglichkeiten ausnutzen zu können. In diesem Fall wurde SYS9/SYS von
G-DOS 2.4 benutzt, wo noch genügend Platz ist. Die neue Entry-Adresse ist
im Hex-Dump auf der nächsten Seite doppelt unterstrichen. Wer in seinem
SYS9 diesen Platz nicht mehr hat oder im schraffierten Teil, besonders an
der einfach unterstrichenen Stelle (Ladecodes) Unterschiede feststellt,
müßte u. U. auf ein anderes SYS-File ausweichen. Wie dann der Requestcode
für diesen Befehl anzupassen ist, wurde im Info schon verschiedentlich
beschrieben. Ebenfalls ist bekannt, wie das neue Befehlswort GO in die
Library zu implementieren ist.

Die Befehlssyntax geht aus der Überschrift hervor. Wenn nur GO ohne
nachfolgende Adresse eingegeben wird, holt das Programm das Sprungziel
aus den Speicherstellen 4403/04h. Dort wird die Entry eines CMD-Files
beim Laden abgelegt, auch dann, wenn nur LOAD,filename eingegeben wurde.
Ein wenig Vorsicht ist hier geboten. Hartmut Grosser stellt in seinem
"DOS-Buch" fest, daß dies der einzige Verwendungszweck für 4403/04h ist
und schließt daraus messerscharf, daß man dort im DOS-CALL-Status Para-
meter an ein Benutzerprogramm übergeben darf. Nach dieser ziemlich über-
flüssigen Anregung (es gibt noch mehr solche Freistellen) könnte es nun
sein, daß dort in der Tat zu irgendeinem Zeitpunkt nicht mehr die Ein-
sprungsadresse eines Programms steht, weil ein nicht selbst geschriebenes
Programm dort herumwühlt. Zumindest unmittelbar nach dem Laden kann man
sich jedoch darauf verlassen, daß GO ohne Angabe einer Adresse das Pro-
gramm zuverlässig anspringt.

Wird eine Adresse durch Komma oder Blank getrennt mit angegeben,
wird an diese Adresse gesprungen. Hat man also mehrere Programme im
Speicher, so kann jedes von ihnen aufgerufen werden. Ohne Angabe der
Adresse wird jedoch nur das zuletzt geladene gefunden. Auf diese Weise
ist es z. B. auch möglich, ROM- oder DOS-Routinen aufzurufen, die norma-
lerweise aus der Kommandoebene nicht zugänglich sind. Sie werden dabei
wie Unterprogramme behandelt, müssen also mit RET abschließen. Tun sie
das nicht, dann kann man u. U. in Level 2, im Debugger oder im Nirwana
landen, je nach dem.

Das Programm selbst (Listing umseitig) ist mit den Kommentaren wohl
hinreichend erläutert. Es sei nur hinzugefügt, daß die optionale Adresse
immer vierstellig in Hex eingegeben werden muß. Der Hexkenner "h" o.
dergl. erübrigt sich, denn es werden bei einer Adresse nur die ersten
vier Stellen gelesen. Was nachfolgt, wird einfach ignoriert.

Arnulf Sopp


```
5147            00001           ORG     5147h           ;ab hier Platz in SYS9
                00002
5147  FE6B      00003 sys9      CP      6bh             ;Lib-Befehl GO?
5149  C2004D    00004           JP      NZ,4d00h        ;sonst dort weiter
514C  CDD54C    00005           CALL    4cd5h           ;folgt eine Sprungadr.?
514F  2004      00006           JR      NZ,param        ;falls ja
5151  2A0344    00007           LD      HL,(4403h)      ;nein, aktuelle Startadr.
5154  E9        00008           JP      (HL)            ;Programm anspringen
                00009
5155  2B        00010 param     DEC     HL              ;wegen RST 10 Zeiger zur.
5156  CD6C51    00011           CALL    read            ;Ziffer einlesen
5159  57        00012           LD      D,A             ;1. Hälfte des MSB
515A  CD6C51    00013           CALL    read            ;nächste Ziffer
515D  79        00014           LD      A,C             ;unteres Nibble
515E  B2        00015           OR      D               ;komplettes MSB
515F  57        00016           LD      D,A             ;laden
5160  CD6C51    00017           CALL    read            ;nächste Ziffer
5163  5F        00018           LD      E,A             ;1. Hälfte des LSB
5164  CD6C51    00019           CALL    read            ;nächste Ziffer
5167  79        00020           LD      A,C             ;unteres Nibble
5168  B3        00021           OR      E               ;komplettes LSB
5169  6F        00022           LD      L,A             ;L <- Adreß-LSB
516A  62        00023           LD      H,D             ;HL <- komplette Adresse
516B  E9        00024           JP      (HL)            ;Adresse anspringen
                00025
516C  D7        00026 read      RST     10h             ;Ziffer aus dem String
516D  380A      00027           JR      C,ziphok        ;falls gültige Dezziffer
516F  FE47      00028           CP      'G'             ;ungültige Ziffer?
5171  300E      00029           JR      NC,error        ;falls ja
5173  FE41      00030           CP      'A'             ;gültige Ziffer?
5175  380A      00031           JR      C,error         ;falls nein
5177  D607      00032           SUB     07h             ;an Dezziffern angleichen
5179  E60F      00033 ziphok    AND     0fh             ;ASCII -> binär
517B  4F        00034           LD      C,A             ;Akku retten
517C  07        00035           RLCA                    ;unteres -> oberes Nibble
517D  07        00036           RLCA
517E  07        00037           RLCA
517F  07        00038           RLCA
5180  C9        00039           RET
                00040
5181  3E2F      00041 error     LD      A,2fh           ;Code "falsche Parameter"
5183  B7        00042           OR      A               ;NZ-Bedingung
5184  C30944    00043           JP      4409h           ;raus mit Fehler
                00044
5147            00045           END     sys9

00000 Fehler
```

```
DRV 00  0608 1A96 1323 20DA 01EA FB50 10FB D1E1  .....#.....F....
0   10  C1F6 35C9 27E2 5E4E 204F 4E4C 5927 2041  ..5.'RUN.ONLY'.A
OH  20  4242 5255 4348 0344 6F73 6665 656C 6572  BBRUCH.Dosfehler
    30  2066 6174 616C 0321 2120 2052 4553 4554  .fatal.!!..RESET
DRS 40  206D 6974 2054 6173 7465 2052 210D E7C2  .mit.Taste.R!...
148950  994E 7EE6 50FE 4DCA 2040 C3FE 6FC2 0042  .Nx.P.E.-§..k..M
5D1H60  CDD5 4C20 042A 0344 E92B CD6C 5157 CD6C  ..L..*.D.+.1QW.1
    70  5179 B257 CD6C 515F CD6C 5179 B36F 62E9  Qy.W.1Q_.1Qv.ob.
TRK 80  D738 0AFE 4730 0EFE 4138 0AD6 07E6 0F4F  .8..G0..A8.....O
41  90  0707 0707 C93E 2FB7 C309 4400 0000 0000  .....>/...D.....
29H A0  0000 0000 0000 0000 0000 0000 0000 0000  ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000  ................
TRS C0  0000 0000 0000 0000 0000 0000 0000 0000  ................
13  D0  0000 0000 0000 0000 0000 0000 0000 0000  ................
DH  E0  0000 0000 0000 0002 2000 0300 00F5 D3FA  ................
    F0  AFC3 0000 0202 4751 0000 0000 0000 0000  ......GQ........
```

Tscrips-Verschlüsselung

Im letzten Clubinfo habe ich darüber berichtet, wie man durch Abschaffen der Kassetten-E/A viel Platz im Tscrips erhält, den man nur nutzen muß. Eine Möglichkeit will ich Euch heute vorführen.
Beim Clubtreffen zeigte mir Hartmut eine Verschlüsselungsroutine in Pascal und fragte mich, ob man diese nicht in das Tscrips einbauen könnte. Diese Routine hat zwei Vorteile: die von ihr verschlüsselten Texte werden beim zweiten Durchlauf wieder entschlüsselt und das Codewort (bzw. Paßwort) existiert nur im Kopf des Benutzers. Da die Länge des Codeworts in dieser Version 1-53 Buchstaben beträgt, dürfte eine Entschlüsselung äußerst kompliziert sein.
Der Aufruf erfolgt beim Laden / Speichern mit dem Parameter "V": "S,V filename" beim Speichern und "L,V<,C> filename" beim Laden. Es wird jeweils nach dem Codewort gefragt. Da die Eingabe über die <BREAK>-Routine erfolgt, sind Kleinbuchstaben nur möglich, wenn der erste Buchstabe "F", "D" oder "R" lautet (Find, Delete, Replace).
Die Erweiterung wurde für Tscrips 5.4 geschrieben. Mit einer Änderung kann sie auch für Tscrips 4.0 übernommen werden. In Zeile 101 muß es dann heißen:

```
101        LD    DE,8E4Bh        ;anderer Anfang des Textspeichers
```

Vielen Dank an Hartmut!

Gerald Schröder

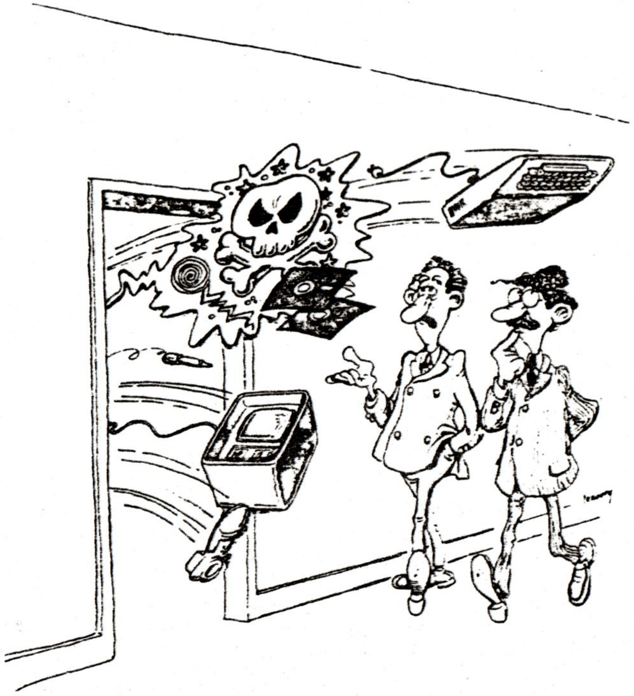
```
00001 ;         PATCH #2
00002 ;         07/86 BY GERALD SCHRÖDER
00003 ;
00004 ;         VERSCHLUSSELUNGS-ROUTINE FÜR TSCRIPS 5.4 / 4.0
00005 ;         IDEE UND LITERATUR VON HARTMUT OBERMANN
00006 ;
00007           ORG     5D42H
00008           CALL    CODLOA              ;CODEWORD EINGEBEN (LOAD)
00009 ;
00010           ORG     5DB8H
00011           CALL    CODSAV              ;CODEWORD EINGEBEN (SAVE)
00012 ;
00013           ORG     5DE5H
00014           CALL    VERLOA              ;ENTSCHLUSSELN BEI LOAD
00015 ;
00016           ORG     5E0AH
00017           CALL    VERSAV              ;VERSCHLUSSELN BEI SAVE
00018 ;
00019           ORG     626AH               ;ROUTINEN IN TAPE-E/A
00020 ;
00021 CODLOA    CALL    CODEEIN             ;CODEWORD EINGEBEN
00022           LD      A,(7CBDH)           ;ALTES KOMMANDO IM LOAD
00023           RET
00024 ;
00025 CODSAV    CALL    CODEEIN             ;CODEWORD EINGEBEN
00026           CALL    72A9H               ;ALTES KOMMANDO IM SAVE
00027           RET
00028 ;
00029 CODEEIN   BIT     2,(IY+0EH)          ;FLAG "V" GESETZT?
00030           RET     Z                   ;NEIN, KEINE VERSCHLUSSELUNG
00031           PUSH    HL
00032           PUSH    DE
00033           PUSH    BC
00034           LD      HL,7A25H            ;GRUNDWERTE FUR RND-ROUTINE
00035           LD      DE,7C74H            ;UEBERTRAGEN
00036           LD      BC,0004
00037           LDIR
00038           LD      (IY+07H),8FH        ;CURSORZEICHEN
00039 NOCHMAL   LD      HL,CODEW            ;ZEIGER AUF ABFRAGE-TEXT
00040           CALL    6BC8H               ;IN DIE LETZTE ZEILE
00041           LD      B,53                ;BIS 53 ZEICHEN EINGEBEN
00042           CALL    5E72H               ;EINGABE
00043           CALL    6BE9H               ;LETZTE ZEILE LöSCHEN
00044 ;                                     1. ZEICHEN AUS DEM BUFFER
00045           CALL    53FFH               ;=1E? (ENTER)
00046           JR      Z,NOCHMAL           ;JA, NOCHMAL EINGEBEN
00047           POP     BC
00048           POP     DE
00049           POP     HL
00050           RET
00051 ;
00052 CODEW     DB      CODEWE-CODEW        ;LÄNGE DES TEXTES
00053           DM      'Codeword? '
00054 CODEWE    EQU     $-1
00055 ;
00056 ;
00057 VERLOA    CALL    VERSCHL             ;ENTSCHLÜSSELN
00058           LD      HL,7CD1H            ;ALTES KOMMANDO
00059           RET
00060 ;
00061 VERSAV    CALL    VERSCHL             ;VERSCHLUSSELN
00062           LD      DE,7CB1H            ;ALTES KOMMANDO
00063           RET
00064 ;
00065 VERSCHL   BIT     2,(IY+0EH)          ;FLAG "V" GES.?
00066           RET     Z                   ;NEIN, KEIN ENT-/VERSCHLUSSELN
00067           PUSH    HL
00068           PUSH    DE
00069           LD      DE,7DD2H            ;ANFANG CODEWORD
00070           LD      B,0                 ;LÄNGE DES SEKTOREN (=256)
00071           LD      HL,7CD1H            ;ZEIGER AUF SEKTOR-BUFFER
00072 LOOP      LD      A,(DE)              ;CODEWORD-BUCHSTABE LADEN
00073           CP      1EH                 ;ENTER? (ENDE DES WORTES?)
00074           JR      NZ,SO1              ;NEIN
00075           LD      DE,7DD2H            ;ZEIGER AUF CODEWORD-ANFANG
00076           LD      A,(DE)              ;1. BUCHSTABEN LADEN
00077 SO1       LD      C,(HL)              ;BUCHSTABE AUS DEM SEKTOR-BUFFER
00078           BIT     0,L                 ;ZEIGER GERADE?
00079           JR      Z,SO2               ;JA
00080           RLCA                        ;CODEWORD-BUCHSTABE LINKS ROTIEREN
00081           JR      SO3
00082 SO2       RRC     A                   ;CODEWORD-BUCHSTABE RECHTS ROTIEREN
00083 SO3       XOR     C                   ;XOR MIT TEXT-BUCHSTABE
00084           LD      C,A                 ;ERGEBNIS NACH C
00085           PUSH    HL
00086           CALL    72B2H               ;RND-ZAHL IN A ERZEUGEN
00087           POP     HL
00088           XOR     C                   ;XOR MIT BUCHSTABEN
00089           LD      (HL),A              ;IN DEN SEKTOR-BUFFER
00090           INC     DE
00091           INC     HL
00092           DJNZ    LOOP                ;BIS SEKTOR ZUENDE
00093           POP     DE
00094           POP     HL
00095           RET
00096 ;
00097 ;
00098           ORG     66D8H               ;ANDERUNGEN IM SAVE/LOAD-AUFRUF
00099           JP      Z,5D97H             ;SPRUNG ZU SAVE
00100           BIT     0,(IY+0EH)          ;FLAG "C" = VERKETTEN?
00101           LD      DE,9177H            ;TEXTSPEICHER-ANFANG (8E4BH IN 4.0)
00102           JR      Z,SO4               ;NEIN, NICHT VERKETTEN
00103           LD      DE,(7D2BH)          ;ENDE DES TEXTES
00104 SO4       JP      5D23H               ;SPRUNG ZU LOAD
```

```
;
:
        ORG     6ACFH
        DB      'V'                 ;PARAMETER "V" STATT "T"
        END
```

Kleiner Tip für Tscrips

Bei allen Tscrips-Versionen kann man sich ein böses Fingerleiden zuziehen, wenn man zu oft TABt (mit Klammeraffe Rechtspfeil). ZEUS hat da eine bessere Lösung: SHIFT SPACE. Wer das auch gerne in Tscrips hätte, muß nur in die Speicherstelle 7992h das Byte 1Fh statt 20h schreiben.

Gerald Schröder



Und das ist das Büro unseres Programmierers.

Ach du liebe Zeit!

Die in unserem Club vertretenen Computer, die eine eingebaute Echtzeituhr haben, werden vom DOS leider teilweise im Stich gelassen. Es ist ein Relikt aus den alten Tagen des TRS-80, daß nach dem Wochentag nie gefragt wird. Ihn durch ein Programm aus dem Datum zu errechnen, wäre zu speicherplatzaufwendig geworden. Der zweite Grund, weshalb ich mich erneut mit der Uhr beschäftige, ist, daß das Boot-EPROM des Genie 3 s (und wohl auch anderer Rechner) die Uhr überhaupt nicht zur Kenntnis nimmt.

In meinem Artikel "Real Time Black Box?" ist bereits ausführlich beschrieben, wie die Uhr funktioniert. Deshalb baue ich heute darauf auf und verzichte auf allzu weitschweifende wiederholende Erklärungen. In jenem Artikel ist mir übrigens ein Fehler unterlaufen, der hiermit richtiggestellt werden soll: In dem kleinen Programm, das den Sonntag programmiert, muß anstelle der 7 eine 6 auf den Port 5A ausgegeben werden. Die 7 Wochentage werden nämlich ab 0 gezählt, so daß der höchste die Nr. 6 ist. Es war wirklich nicht meine Absicht, eine arbeitnehmerfeindliche Woche einzuführen.

Überhaupt ist dieser Punkt etwas heikel. Mein CP/M behauptet am Wochentag "0" steif und fest, er heiße Montag. Der im letzten Artikel erwähnte Auszug einer anderen Publikation findet, daß der Sonntag der erste Tag der Woche sei, auch für diesen Chip. Bibel beiseite, es ist egal! Wer nach dem vorgestellten Strickmuster einen Wochentag programmieren will, kann ihm jede beliebige laufende Nummer geben. Hauptsache, alle Programme oder Betriebssysteme benennen ihn gleich - und gemäß einer für alle gleichen Vereinbarung.

Um das Auslesen der Uhr sowohl im EPROM als auch unter DOS zu ermöglichen, muß die Ausgabe auf eine möglichst neutrale Art erfolgen. Beide Systeme verwalten nämlich den Bildschirm sehr unterschiedlich, so daß ein allgemeingültiger Algorithmus zur Anzeige nicht möglich ist. Daher wird hier beim Einsprung davon ausgegangen, daß HL als Zeiger auf einen Puffer geladen ist, in den der ASCII-String mit der Zeitangabe geladen werden soll. Jedes beliebige System kann anschließend diesen String auf seine Weise in den Bildschirm übertragen.

Dieser String hat folgende Gestalt: Fr, 22.08.86, 13:48:43. Es ist die komplette Zeit, wobei die einzelnen Angaben durch Komma und ein folgendes Leerzeichen getrennt sind. An der Stelle, wo hier der Satzende-Punkt (nicht Bestandteil der Zeitansage) steht, wird der String durch das Programm mit einem 0D-Byte (ENTER) abgeschlossen. Nach dem Anzeigen der Zeit wird deshalb immer in die nächste Zeile gesprungen.

Das Listing zeigt außer dem Programm selbst eine kleine Routine (ab Label start), die die Adresse eines Puffers nach HL lädt und dann das Ausleseprogramm aufruft. Zuletzt wird über die DOS-Routine an 4467 der Pufferinhalt auf den Bildschirm übertragen. Wenn das Programm immer parat ist (SYS-File, EPROM, Helmuts "sicheres Plätzchen"), dann genügen zum Abruf einer Zeitansage ein paar Bytes, wie man sieht.

Die nun folgenden Erläuterungen zum Verständnis des Programms sind grundsätzlich für alle Uhrenchips, also auch für alle Echtzeituhren im Club gültig. Nur das Auslesen des Chips selbst, das von dem Unterprogramm rdclk erledigt wird, dürfte von Wecker zu Wecker variieren.

Es beginnt damit, daß der Pufferzeiger HL auf den Stack gerettet wird. Damit steht er beim abschließenden POP (Z. 33) wieder auf dem Pufferanfang, so daß das aufrufende Programm einen mundgerechten Zeiger zurückbekommt. Das Y-Indexregister wird sodann als Zeiger auf eine Maskentabelle geladen. Die verschiedenen Register des RTC-Bausteins enthalten nämlich nicht nur die jeweiligen Ziffern einer Zeitangabe, sondern daneben gibt es noch Bits, die etwas von Schaltjahren, Nachmittagen und

12/24-Stunden-Anzeige erzählen. Sie müssen ausmaskiert werden, weil nur die Ziffern selbst interessieren.

Das erste auszulesende Register ist Nr. 6, wo die Wochentage verraten werden. Die chipinterne Adresse 6 kommt beim 63s in das obere Nibble eines Bytes, das gleichzeitig im unteren den Befehl "lesen" (xC) enthält. Deshalb wird das Register A mit dem Adreßbyte 6C geladen. Mit dieser teuren Fracht geht's ab ins UP rdreg, wo das gewünschte RTC-Datenregister über den Port 5B adressiert wird, um über den Port 5A ausgelesen werden zu können. Danach wird diese BCD-Ziffer gleich mit OR '0' in ihre ASCII-Entsprechung verwandelt, weil dies wegen der übrigen Zeiteinheiten ökonomischer erschien. Nachteil: Nach dem CALL in Z. 12 muß mit AND OF gleich wieder der Binärwert gebildet werden. Er ist nun die laufende Nummer eines Wochentags in der Tabelle daytab.

Es wird hier übrigens nicht rdreg ab Z. 49, sondern rdreg+1 ab Z. 50 aufgerufen. Der Umweg über das Register C, dessen Bedeutung später erläutert wird, ist hier nicht nötig.

Die Tagesnamen haben zwei Zeichen. Um einen Zeiger auf den korrekten Tag setzen zu können, muß der Akkuinhalt deshalb verdoppelt werden. Das geht am schnellsten mit RLCA. Als Zeiger wird DE geladen. Die Addition des LSB E und der Tagesnummer führt zum richtigen Byte in der Tabelle. Wer diese Routine in ein eigenes Programm (z. B. ein SYS-File) einbauen möchte, muß dabei allerdings höllisch aufpassen, daß die Tagestabelle komplett in einer Page liegt, d. h., das Adreß-MSB muß für alle Namen gleich sein. Z. B. die Grenze zwischen 61FF und 6200 darf nicht mitten in der Tabelle liegen.

Alternativvorschlag, falls das nicht geht:

```
        ADD     A,E
        LD      E,A
        LD      A,0
        ADC     A,D
        LD      D,A
```

Dann geht es ab Z. 18 normal weiter.

Nach dem Einstellen des Wochentagszeigers muß der Pufferzeiger um eine Stelle zurückgesetzt werden, denn das UP rdreg erhöhte ihn. Dann werden die beiden zutreffenden Zeichen in den Puffer geschrieben. Im UP delim kommen noch ein Komma und ein Blank dahinter, um die Zeiteinheiten optisch voneinander zu trennen.

In Z. 24 wird BC mit 034C geladen. Dabei wird B als Zähler für drei Datumseinheiten (Tag, Monat, Jahr) aufgesetzt, C beinhaltet das Adreßregister 4 der Uhr. Im oberen Nibble, wohlgemerkt (s. o.). Das untere ist wieder für das Lesen konfiguriert. Ähnlich verhält es sich mit dem Befehl LD DE,402E. E enthält dabei einen Punkt als Trennzeichen der Datumseinheiten, dessen ASCII-Wert 2E beträgt. Was es mit D auf sich hat, ist komplizierter:

Wir brauchen zur Anzeige zuerst die Zehnerstelle des Kalendertages, dann die Einerstelle, dann dasselbe für den Monat, dann für das Jahr. Im UP rdreg wird aber das Adreßregister (in C gespeichert) immer um 10 (weil im oberen Nibble; sonst nur 1) vermindert. Angesichts dieser Tatsache und eingedenk der Reihenfolge der Register, die in der Tabelle des letzten Artikels steht, ist also C nach einem Lesevorgang immer um 4 Register zu niedrig. Also muß 4 addiert werden. Wegen der Nibble-Akrobatik lautet der Wert tatsächlich 40. Und dieser Wert kommt in Z. 25 ins Register D. Nach der Addition der RTC-Adresse in Register C und D am Beginn des UP rdloop beträgt sie also nicht mehr 4, sondern 8. Das ist das Register für die Monatszehnerstelle, die wir nun brauchen. Uff!

Das Unterprogramm rdloop wird nun aufgerufen. Dort passiert zunächst das soeben Erwähnte. Dann geht es gleich in rdclk weiter, einem UP, das

Technik ist wie ein Messer. Man kann damit morden oder damit Brot schneiden.

Es ist anatomisch schwierig und immer ein wenig lächerlich, sich gratulierend auf die eigene Schulter zu schlagen.

zwei RTC-Register ausliest. Es ruft nämlich zunächst rdreg auf, das wir vom Auslesen des Wochentags her bereits kennen und geht anschließend in rdreg über. Dort wird das zuständige Datenregister adressiert und ausgelesen. Die uninteressanten Bits werden ausmaskiert. Der Binärwert wird gleich ins ASCII-Format umgewandelt und in den Puffer geschrieben.

Nach der Rückkehr von rdclk kommt das Trennzeichen, beim Datum ein Punkt, aus E in den Puffer (Z. 39). Wenn alle drei Einheiten des Datums geschrieben sind, stört natürlich der Punkt hinter der Jahreszahl. In Z. 42 wird deshalb der Pufferzeiger auf ihn zurückgestellt, damit er mit einem Komma und einem anschließenden Blank überschrieben werden kann.

Nun ist die Uhrzeit dran. In Z. 27 wird wieder BC mit der Anzahl von Schleifen (in B) und der RTC-Adresse (in C) geladen. Diese Adresse lautet jetzt 5, weil dort die jetzt fällige Zehnerstelle der Stunden steht. Wie aus der erwähnten Tabelle hervorgeht, können wir jetzt immer ein Register tiefer rutschen, denn die Reihenfolge der Register deckt sich mit der Reihenfolge der Ziffern in einer Zeitangabe. Die Subtraktion in Z. 51 im UP rdreg trifft also immer genau das jeweils gewünschte Register. So kann auch DE mit ':', dem Trennzeichen für die Zeiteinheiten geladen werden. Auf diese Weise kommt 00 nach D, also der Summand für die Addition in Z. 36 ist 0.

Der Rest ist Handwerk. Wieder wird rdloop aufgerufen, um diesmal die Uhrzeit in den Puffer zu schreiben. Danach stehen dahinter wieder ein Komma und ein Blank, was ganz am Ende des Strings jedoch stört. Der Pufferzeiger HL wird deshalb in Z. 30/40 auf das Komma zurückgestellt, wo dann stattdessen OD, also ENTER gepatcht wird. Nachdem der Zeiger mit POP wieder auf den Pufferanfang gesetzt ist, hat Reserve Ruh'.

Solche Selbstkorrekturen und verschiedenes Andere machen das Programm ziemlich kompliziert, nahezu unübersichtlich. Das Problem ist jedoch, daß es im EPROM residieren soll. Es kann sich also nicht selbst modifizieren wie die Uhrenroutine des 63s im RAM. Bei dem gleichzeitigen Anspruch, es sehr kurz zu halten, sollte eben ein UP möglichst alles können.

Das Listing und der darunterstehende Sektordump geben eine Demoversion wieder, die bestenfalls als Spielkram zu bezeichnen wäre. Da jede Armbanduhr dasselbe leistet, lohnt ein solches Programm im Anwenderspeicher nicht. Interessant wird es erst, wenn es in einem SYS-File, im (EP-)ROM oder in einem Speicherbereich ständig bereitliegt, der Systemroutinen vorbehalten ist.

Arnulf Sopp

```
5200                   ORG     5200h

        ;Rahmenprogramm, je nach dem anzupassen
5200 217F52    start   LD      HL,buffer       ;einen Puffer festlegen
5203 CD0952            CALL    prog            ;Uhr dorthin auslesen
5206 C36744            JP      4467h           ;im Bildschirm anzeigen

        ;Programm zum Auslesen der Uhr
5209 E5        prog    PUSH    HL              ;Pufferzeiger retten
520A FD216452          LD      IY,clktab       ;Tabelle d. Ziffernmasken
520E 3E6C              LD      A,6ch           ;ab Reg. 6 (Wochentage)
5210 CD5352            CALL    rdreg+1         ;Wochentag auslesen
5213 E60F              AND     0fh             ;ASCII -> binär
5215 07                RLCA                    ;* 2, weil 2 Zeichen/Tag
5216 117152            LD      DE,daytab       ;Tabelle der Tagesnamen
5219 83                ADD     A,E             ;auf richtigen N. stellen
521A 5F                LD      E,A             ;Zeiger korrigieren
521B 2B                DEC     HL              ;auf Pufferanfang zurück
521C EB                EX      DE,HL           ;Quelle <-> Ziel tauschen
521D EDA0              LDI                     ;2 Buchstaben des Tages-
521F EDA0              LDI                     ;namens puffern
5221 EB                EX      DE,HL           ;rücktauschen
5222 CD4852            CALL    delim           ;", " hinter Tagesnamen
5225 014C03            LD      BC,034ch        ;3 Durchl. ab Reg. 4+4=8
5228 112E40            LD      DE,402eh        ;Summand 40, Trennz. "."
522B CD3D52            CALL    rdloop          ;3 Datumseinheiten ausl.
522E 015C03            LD      BC,035ch        ;3 Durchl. ab Reg. 5
5231 113A00            LD      DE,':'          ;Summand 00, Trennz. ":"
5234 CD3D52            CALL    rdloop          ;3 Uhrzeiteinh. auslesen
5237 2B                DEC     HL              ;zurück auf letztes ","
5238 2B                DEC     HL
5239 360D              LD      (HL),0dh        ;stattdessan CR puffern
523B E1                POP     HL              ;Zeiger auf Pufferanfang
523C C9                RET
523D 79        rdloop  LD      A,C             ;RTC-Register
523E 82                ADD     A,D             ;je nach dem korrigieren
523F 4F                LD      C,A             ;neu schreiben
5240 CD4F52            CALL    rdclk           ;1 Zeiteinheit auslesen
5243 73                LD      (HL),E          ;Trennzeichen puffern
5244 23                INC     HL              ;Zeiger nachstellen
5245 10F6              DJNZ    rdloop          ;bis 3 Einheiten ausgel.
5247 2B                DEC     HL              ;auf letztes Trennzeichen
5248 362C      delim   LD      (HL),','        ;mit "," überschreiben
524A 23                INC     HL              ;Zeiger nachstellen
524B 3620              LD      (HL),' '        ;Blank dahinter
524D 23                INC     HL              ;Zeiger nachstellen
524E C9                RET
524F CD5252    rdclk   CALL    rdreg           ;Einerstelle einer Einh.
5252 79        rdreg   LD      A,C             ;RTC-Adreßregister
5253 D35B              OUT     (5bh),A         ;Datenreg. adressieren
5255 D610              SUB     10h             ;auf nächstes Register
5257 4F                LD      C,A             ;neu laden
5258 DB5A              IN      A,(5ah)         ;Zeit-Ziffer auslesen
525A FDA600            AND     (IY)            ;relevante Bits maskieren
525D F630              OR      '0'             ;binär -> ASCII
525F 77                LD      (HL),A
5260 23                INC     HL              ;Pufferzeiger nachstellen
5261 FD23              INC     IY              ;dto. Maskenzeiger
5263 C9                RET

        ;Tabelle der Ziffernmasken für die Zeiteinheiten
5264 07        clktab  DB      07h     ;Wochentag,     0-6
5265 03                DB      03h     ;Zehner Datum,  0-3
5266 0F                DB      0fh     ;Einer  Datum,  0-9
```



```
5267 01                DB      01h     ;Zehner Monat,  0-1
5268 0F                DB      0fh     ;Einer  Monat,  0-9
5269 0F                DB      0fh     ;Zehner Jahr,   0-9
526A 0F                DB      0fh     ;Einer  Jahr,   0-9
526B 03                DB      03h     ;Zehner Stunde, 0-2
526C 0F                DB      0fh     ;Einer  Stunde, 0-9
526D 07                DB      07h     ;Zehner Minute, 0-5
526E 0F                DB      0fh     ;Einer  Minute, 0-9
526F 07                DB      07h     ;Zehner Sek.,   0-5
5270 0F                DB      0fh     ;Einer  Sek.,   0-9

        ;Tabelle der Wochentagsnamen
5271 4D        daytab  DM      'Mo'
5273 44                DM      'Di'
5275 4D                DM      'Mi'
5277 44                DM      'Do'
5279 46                DM      'Fr'
527B 53                DM      'Sa'
527D 53                DM      'So'
               buffer
5200                   END     start

00000 Fehler

buffer 527F    clktab 5264    daytab 5271    delim  5248    prog   5209
rdclk  524F    rdloop 523D    rdreg  5252    start  5200
```

```
DRV 00    0181 0052 217F 52CD 0952 C367 44E5 FD21 ...R! R..R.gD..!
0   10    6452 3E6C CD53 52E6 0F07 1171 5283 5F2B dR>l.SR....qR._+
0H  20    EBED A0ED A0EB CD48 5201 4C03 112E 40CD .......HR.L...§.
    30    3D52 015C 0311 3A00 CD3D 522B 2B36 0DE1 =R.ö..:..=R++6..
DRS 40    C979 824F CD4F 5273 2310 F62B 362C 2336 .y.O.ORs#..+6,#6
252050    2023 C9CD 5252 79D3 5BD6 104F DB5A FDA6 .#..RRy.Ä..O.Z..
9D8H60    00F6 3077 23FD 23C9 0703 0F01 0F0F 0F03 ..Ow#.#.........
    70    0F07 0F07 0F4D 6F44 694D 6944 6F46 7253 .....MoDiMiDoFrS
    80    6153 6F02 0200 5200 0000 0000 0000 0000 aSo...R.........
    90    0000 0000 0000 0000 0000 0000 0000 0000 ................
    A0    0000 0000 0000 0000 0000 0000 0000 0000 ................
    B0    0000 0000 0000 0000 0000 0000 0000 0000 ................
FRS C0    0000 0000 0000 0000 0000 0000 0000 0000 ................
0   D0    0000 0000 0000 0000 0000 0000 0000 0000 ................
0H  E0    0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0    0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Arnulf Sopp

Nachtrag zu "Ach du liebe Zeit!"

In dem oben genannten Beitrag war die Rede davon, daß die Routine zum kompletten Auslesen der Uhr das Boot-EPROM des Genie III s zieren sollte. Sie ziert mittlerweile. Und noch etliches mehr. Alle Programme, die sich sinnvoll callen lassen, sind im Anschluß aufgeführt. Es sind teilweise solche, die schon immer drin waren, solche, die zwar drin waren, sich aber aus verschiedenen Gründen nicht als Unterprogramme eigneten (aber inzwischen eignen) und schließlich solche, die erst neuerdings drin sind.

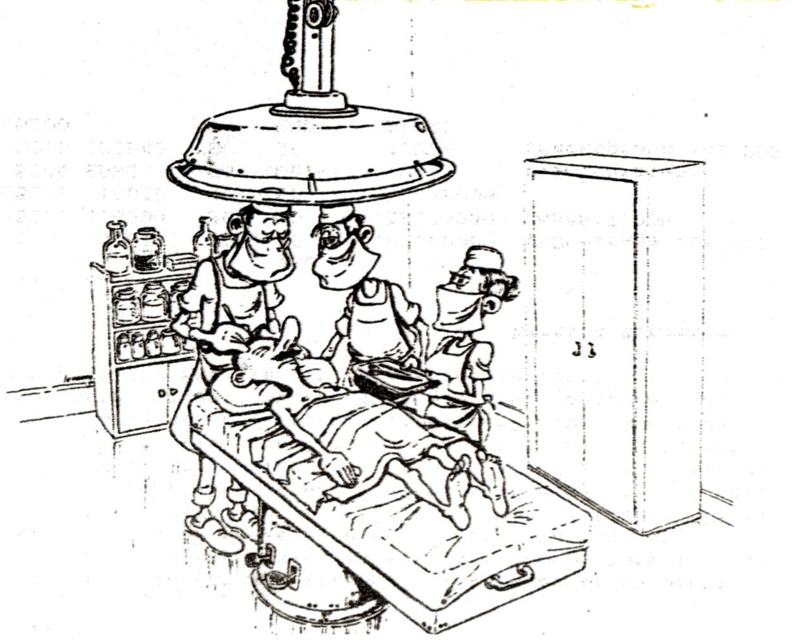
Nicht erwähnt ist dort, daß der Zeichensatz jetzt gefälliger aussieht. Das ist zwar Geschmackssache, aber Unterlängen auf Kosten der Bäuche des p und des q und ähnlich faule Kompromisse kommen nicht mehr vor.

Neu sind auch zwei Befehle, E (für EXECUTE) und U (für UHR). Letzterer tut natürlich das oben Beschriebene. E ruft eine Routine als Unterprogramm auf. Bisher konnte nur mit dem G-Befehl die Kontrolle endgültig an ein Programm im Speicher übergeben werden, das nicht in den Monitor zurückkehrt. EXECUTE hätte ich lieber CALL genannt und mit C aufgerufen, aber dieses Kommando ist bereits für COPY reserviert. Schließlich soll das EPROM mit der Originalversion voll kompatibel bleiben. Immerhin ist hier eine Analogie zu ZETBUG gegeben.

Da ich momentan mit ziemlicher Ausschließlichkeit am EPROM arbeite, ist es sehr wahrscheinlich, daß es zum Zeitpunkt des Erscheinens unseres Infos bereits noch mehr weiß und kann. Diese Aufstellung ist daher vermutlich bereits übermorgen sehr unvollständig.

Wer daran interessiert ist, mag sich bei mir melden. Für die BASIC-Experten unter uns dürfte es allerdings ziemlich uninteressant sein. Der veränderte Zeichensatz, der unter DOS ohnehin überschrieben wird und das gefälligere Logo lohnen nicht einmal das Porto. Wer aber in Maschinensprache arbeitet, bekommt interessante Ware, wie die nachfolgende Aufstellung zeigt. Komplexe Routinen, die sonst jeweils neu programmiert werden müßten, können nun einfach mit einem CALL abgerufen werden.

Interessenten möchten mir bitte die Registriernummer ihres EPROMs nennen. Man erfährt sie, indem man bei Reset oder beim Einschalten die F1-Taste gedrückt hält. Ich verlasse mich auf diese Angabe und wahre damit das Urheberrecht von Uwe Böker. Da immer noch 90% des Programms von ihm stammen, möchte ich nur 15 Mark in gängigen Briefmarken dafür. Ich habe nicht etwa das Gelübde des Reichtums abgelegt sondern kalkuliere folgendermaßen: Ein EPROM kostet z. Zt. ca. 8 Mark. Dazu gibt es Fotokopien der Bedienungs- und Einbauanleitung. Das Porto entspricht dem Gewicht. Den Rest von etwa 5 Mark werde ich auf das Wohl des Erwerbers indirekt der Witwen- und Waisenkasse der hiesigen Gastwirtsinnung zufließen lassen - im wahrsten Sinne des Wortes. Prost!



Das ist alles!? Keine Gehäuseschrauben? Keine Platinen?

```
          *** neue Befehlskürzel: ***
          E adresse  Unterprg. ab adresse aufrufen
          U          Uhr vollständig auslesen

          *** verwendbare Subroutinen: ***
frstscr   oder
019A      Kleinbildschirm löschen, Hello anzeigen
chrset    oder
027B      Zeichensatz (DE) laden (256 Zeichen ab ASCII 00)
sngchr    oder
027D      B Zeichen ab ASCII A aus (DE) laden
scr3c00   oder
08DF      Scroll des Kleinbildschirms

scroll    oder
08E8      Scroll (DE Anf. 1. Z., HL Anf. 2. Z., BC Rest)
delay15   oder
0A63      ca. 15 ms Verzögerung
cphlde    oder
0CA3      HL und DE vergleichen (wie RST 18h)
gethex    oder
0CD4      DE bin. <- Hex-ASCII-String ab (HL) bis Blank
          Cy: falsches Zeichen, Z: 4 Zeichen oder Blank
bincphr   oder
0E89      A bin. <- gültige Hex- und Dezziffer
calend    oder
0F36      vollst. Uhr-Anzeige als ASCII-String nach (HL)
getdec    oder
0FAC      ASCII-Dezimalstring binär nach HL laden
          (Ende bei der ersten Nicht-Ziffer)
putdec    oder
0FCD      Binärzahl in DE als ASCII-Dez-String nach (HL)
```

Arnulf Sopp

In letzter Minute eingetroffen!!

E I N L A D U N G

Liebe Leute,

in einem der vorherigen INFO's wurden Regionaltreffen angeregt. Im südlichen Teilen der Bundesrepublik finden solche Treffen bereits mehr oder weniger häufiger statt. Das sowas im nördlichen Teil nicht klappt, hat glaube ich weniger mit den sog. Nodlichtern zu tun, als mehr mit der geringen Anzahl der Mitglieder! Aber trotzdem will ich einen Versuch starten und Euch nach Osnabrück einladen. Nicht, weil ich hier zufällig wohne, sondern weil Osnabrück recht günstig für ein solchten Vorhaben liegt. Denn von Kiel bis Koblenz ist Osnabrück bei Richtgeschwindigkeit in gut drei Stunden zu erreichen!
Als Termin habe ich das nächste Frühjahr ins Auge gefasst, und zwar den

Samstag und Sonntag 25.+ 26. April 1987

Falls Ihr Interesse an einem solchen Treffen habt, lasst es mich bitte bald wissen, damit ich die notwendigen Vorbereitungen einleiten kann. Ein Pensionshaus in der Umgebung von Osnabrück wird sich zu einem ähnlichen Preis finden lassen wie bei unserer Jahreshauptversammlung. Und falls jemand mit der Bahn anzureisen wünscht, kann der Transport zur Versammlungsstätte auch organisiert werden.

Bleibt mir nur noch, auf eine große Beteiligung zu hoffen! Ich würde mich jedenfalls sehr freuen, Euch im nächsten Jahr in Osnabrück begrüssen zu können.

Hans-Martin Stephan

M4/4P ecke

GRAFIK - SHORTY FUER MODELL 4/4P
================================

```
1 CLEAR.-4993:CLS:V=2944:PRINT CHR$(15):POKE 120,134:WHILE 1:D=D-5*(D=0):Y=Y+(Y<
71)*(D<4)-(Y>0)*(D>6):D=D MOD 3:X=X+(X<159)*(D=0)-(X>0)*(D=1):P=80*(Yö3)+Xö2-204
8:C=PEEK(P):B=2^(2*Y MOD 6+X MOD 2):C=C*(C>127)*(C<192)AND-1+B*(M=2)OR 128-B*(M=
1)XOR-B*(M=3)
2 POKE P.C XOR B:A$=INKEY$:D=VAL(A$):N=INSTR("OXXX."+CHR$(13),A$)+(A$=""):POKE P
,C:WHILE N>4:U=-V*(N=6):V=V-U:FOR I=-2048 TO-129:POKE I-U,PEEK(I-V):NEXT:IF U TH
EN POKE 120,135:PRINT CHR$(14)ELSE N=1:WEND:IF D+N THEN M=N+(N>0)-M*(N=0):WEND E
LSE 2
```

****
Kleines Grafikprogramm fuer Model 4/4P aus 80Micro 07/86/160

Mit diesem Zweizeiler lassen sich Grafiken mittels des Ziffernblocks erstellen. Die Ziffern 1-9 (ausser 5) werden fuer die entsprechende Richtung benutzt. Die Ziffer 0 benutzt man, um aus dem momentanen Modus zu kommen.

Den Modus waehlt man mit den Funktionstasten:
F1 - Zeichen   F2 - Loeschen   F3 - Invertieren

Mit ENTER wird die erstellte Grafik gespeichert: mit
SYSTEM"DUMP Filename (START=60544,end=62463)"
wird die Grafik auf die Disk gespeichert. Mit
SYSTEM"LOAD Filename"
kann man die Grafik wieder von der Disk laden und innerhalb des Programms mit < . > aufrufen.

Hinweis zur Programmeingabe:
die drei "XXX" in Zeile 2 werden durch das jeweilige Zeichen der Funktionstasten F1,F2 und F3 ersetzt (es handelt sich dabei um Pixel)
So und nun viel Spass beim Grafikeln

Klaus Hermann

m4/4p ecke

TANDY - LOGO (TRSDOS 6.x)
=========================

Nicht jedem gefällt wohl das Tandy-Logo, das beim booten auf der Bildfläche erscheint. Außerdem sieht man sich mit der Zeit satt daran. Der eine oder andere hätte vielleicht auch sein eigenes Erkennungszeichen oder ähnliches auf dem Schirm. Deshalb eine kurze Anleitung für eine Änderung.
Das Logo ist in den letzten 3 Records von SYS0/SYS abgespeichert. Mit einem entsprechenden Programm oder auch mit DEBUG lassen sich die Grafikzeichen leicht in Leerzeichen oder andere Grafikzeichen umwandeln. Es ist jedoch unbedingt erforderlich, daß die Ladeadressen nicht überschrieben werden, ansonsten hängt sich das System beim booten auf. Die Zeichen für das Logo sind in den Listings durchgestrichen.
Die Copyright Meldung steht ebenfalls in SYS0/SYS (Record 7). Die Verfahrensweise für eine Änderung ist dieselbe wie oben bereits beschrieben.
So und zuletzt noch die Zeile mit der Meldung "LS-DOS62Level-AN". Diese Meldung steht in BOOT/SYS, Record 2 ab Byte 10H.
Nach einer Änderung in Leerzeichen hat man nun nur noch einen leeren Bildschirm vor sich. Datum- und Zeiteingabe lassen sich mit dem SYSTEM-Befehl abschalten.

Klaus Hermann

SYS0/SYS Drive 0 Record X'0007'

0123456789ABCDEF	BYTE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
.&.G.....<.:..D.	<00>	CD	26	1A	47	07	07	07	E6	07	3C	CD	3B	19	C5	4F	06
...#..8.....DäY.	<10>	08	AF	CB	23	17	B9	38	02	91	1C	10	F6	4F	7B	59	C1
.W>..+..oz(....7	<20>	C9	57	3E	04	CD	2B	1A	CE	6F	7A	28	01	87	C9	01	37
..LS-DOS 06.02.0	<30>	0D	F8	~~4C~~	~~53~~	~~2D~~	~~44~~	~~4F~~	~~53~~	~~20~~	~~30~~	~~36~~	~~2E~~	~~30~~	~~32~~	~~2E~~	~~30~~
OD- Copyright 19	<40>	~~30~~	~~44~~	~~2D~~	~~20~~	~~43~~	~~6F~~	~~70~~	~~79~~	~~72~~	~~69~~	~~67~~	~~68~~	~~74~~	~~20~~	~~31~~	~~39~~
84 Logical Syste	<50>	~~38~~	~~34~~	~~20~~	~~4C~~	~~6F~~	~~67~~	~~69~~	~~63~~	~~61~~	~~6C~~	~~20~~	~~53~~	~~79~~	~~73~~	~~74~~	~~65~~
ms Inc..5^.All R	<60>	~~6D~~	~~73~~	~~20~~	~~49~~	~~6E~~	~~63~~	~~2E~~	01	35	5E	F8	~~41~~	~~6C~~	~~6C~~	~~20~~	~~52~~
ights Reserved.	<70>	~~69~~	~~67~~	~~68~~	~~74~~	~~73~~	~~20~~	~~52~~	~~65~~	~~73~~	~~65~~	~~72~~	~~76~~	~~65~~	~~64~~	~~2E~~	~~20~~
Licensed to Tand	<80>	~~4C~~	~~69~~	~~63~~	~~65~~	~~6E~~	~~73~~	~~65~~	~~64~~	~~20~~	~~74~~	~~6F~~	~~20~~	~~54~~	~~61~~	~~6E~~	~~64~~
y Corporation...	<90>	~~79~~	~~20~~	~~43~~	~~6F~~	~~72~~	~~70~~	~~6F~~	~~72~~	~~61~~	~~74~~	~~69~~	~~6F~~	~~6E~~	~~2E~~	01	02
....(.B.6.E.#.=.	<A0>	00	01	F2	1B	28	06	42	06	38	06	45	06	23	06	3D	06
v.5...-.....0.(.	<B0>	79	19	35	06	85	05	2D	05	03	05	00	05	30	05	28	05
....	<C0>	D0	0B	82	03	87	19	A8	07	8D	07	89	06	08	1B	0B	1B
..B.a...........	<D0>	F4	1A	7E	19	7B	19	0F	1B	A0	19	F5	1C	DA	1C	D7	1C
....	<E0>	EB	1C	D0	1C	93	19	AF	19	AC	19	F4	1A	F4	1A	F4	1A
....	<F0>	F4	1A	B5	19	BC	19	C0	19	C4	19	C8	19	CC	19	D0	19

SYS0/SYS Drive 0 Record X'0010'

0123456789ABCDEF	BYTE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
.........:ö.....	<00>	00	01	82	80	03	00	00	C5	3A	7C	00	CB	5F	C4	8C	03
..x. ....x...!..	<10>	C1	0B	78	B1	20	FB	C9	C5	E5	78	E6	07	07	21	D1	03
Ox...N#n.....<g:	<20>	4F	78	06	00	09	4E	23	6E	0F	0F	0F	E6	1F	3C	67	3A
ö...(..$.%..!..A>	<30>	7C	00	E6	0B	28	06	CB	24	CB	25	CB	21	F3	E5	41	3E
.....A<....- ..%	<40>	01	D3	90	10	FE	41	3C	D3	90	10	FE	2D	20	F0	E1	25
.....FLVHöDb§K<	<50>	20	EB	FB	E1	C1	C9	50	4C	56	48	5C	44	62	40	6B	3C
r8v4.0...B.0....	<60>	72	38	79	34	80	30	01	00	00	7E	D6	30	D8	FE	0A	D0
..)).)..D.DM.#..	<70>	C5	E3	29	29	09	29	06	00	4F	09	44	4D	E1	23	18	E9
......&W........	<80>	F3	AF	D3	84	C7	01	26	57	~~F9~~	~~82~~	~~AF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~
....	<90>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~
...........tm.#.	<A0>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~9F~~	~~A1~~	~~84~~	~~90~~	~~80~~	~~74~~	~~6D~~	01	23	A9
....	<B0>	F9	~~8B~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~8F~~	~~BF~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~
....	<C0>	~~8F~~	~~8F~~	~~CF~~	~~8F~~	~~AF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~97~~	~~90~~	~~A1~~	~~86~~	~~98~~
...!............	<D0>	~~A1~~	~~84~~	01	21	FA	F9	~~82~~	~~AF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BC~~	~~BC~~	~~BC~~
....	<E0>	~~BC~~	~~BC~~	~~50~~	~~A8~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~	~~BE~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~9F~~	~~A1~~	~~86~~
.......L........	<F0>	~~98~~	~~A1~~	~~86~~	~~98~~	~~81~~	01	1D	4C	FA	~~8D~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~

SYS0/SYS Drive 0 Record X'0011'

0123456789ABCDEF	BYTE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
....	<00>	~~BF~~	~~BF~~	~~BF~~	~~B0~~	~~AA~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~87~~	~~98~~	~~A1~~
....	<10>	~~86~~	~~98~~	~~A1~~	~~86~~	01	1B	9D	FA	~~82~~	~~AF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~
....	<20>	~~BF~~	~~B0~~	~~AA~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~9F~~	~~A1~~	~~86~~	~~98~~	~~A1~~	~~86~~	~~98~~
....	<30>	~~81~~	01	17	EF	FA	~~8D~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~B0~~	~~BA~~	~~BF~~	~~BF~~
............§...	<40>	~~BF~~	~~BF~~	~~BF~~	~~87~~	~~98~~	~~A1~~	~~86~~	~~92~~	~~A1~~	~~86~~	01	15	40	FB	~~82~~	~~AF~~
....	<50>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~9F~~	~~A1~~	~~86~~	~~98~~	~~A1~~	~~86~~	~~98~~
........:.......	<60>	~~81~~	01	11	92	FB	~~AB~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~	~~CF~~	~~97~~	~~A8~~	~~81~~
....	<70>	~~96~~	~~A8~~	~~81~~	~~96~~	01	11	E2	FB	~~BA~~	~~BC~~	~~B0~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~
.........0......	<80>	~~BF~~	~~8A~~	~~90~~	~~AE~~	~~8A~~	~~90~~	~~A5~~	01	15	30	FC	~~A0~~	~~BE~~	~~BF~~	~~BF~~	~~BF~~
....	<90>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~B8~~	~~92~~	~~A4~~	~~89~~	~~92~~	~~A4~~	~~89~~	~~90~~	01	17
....	<A0>	7F	FC	~~B8~~	~~BF~~	~~BF~~	~~87~~	~~83~~	~~83~~	~~83~~	~~83~~	~~83~~	~~83~~	~~83~~	~~8B~~	~~BF~~	~~BF~~
....	<B0>	~~B4~~	~~89~~	~~92~~	~~A4~~	~~89~~	~~92~~	~~A4~~	01	1B	CD	FC	~~A0~~	~~BE~~	~~BF~~	~~BF~~	~~81~~
....	<C0>	~~A0~~	~~BE~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BD~~	~~B0~~	~~B2~~	~~BF~~	~~BF~~	~~BD~~	~~92~~	~~A4~~	~~89~~
....	<D0>	~~92~~	~~A4~~	~~89~~	~~90~~	01	1D	1C	FD	~~B8~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~80~~	~~AA~~	~~BF~~
....	<E0>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~B4~~	~~89~~	~~92~~	~~A4~~
....!3..........	<F0>	~~89~~	~~92~~	~~A4~~	01	21	6A	FD	~~A0~~	~~BE~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~B4~~	~~89~~

SYS0/SYS Drive 0 Record X'0012'

0123456789ABCDEF	BYTE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
....	<00>	~~BE~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~	~~8F~~	~~87~~	~~B0~~	~~B8~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BD~~	~~92~~
........#.......	<10>	~~A4~~	~~89~~	~~92~~	~~A4~~	~~89~~	~~90~~	01	23	B9	FD	~~B8~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~
....	<20>	~~BF~~	~~BF~~	~~BF~~	~~BD~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~	~~BC~~	~~BE~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~
............#...	<30>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~B4~~	~~89~~	~~92~~	~~A4~~	~~89~~	~~92~~	~~84~~	01	23	07	FE	~~A0~~
....	<40>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~
....	<50>	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BF~~	~~BD~~	~~92~~	~~84~~	~~81~~
.:..............	<60>	02	02	00	1E	00	00	00	00	00	00	00	00	00	00	00	00
....	<70>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<80>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<90>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<A0>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<B0>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<C0>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<D0>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<E0>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
....	<F0>	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Model 4P ROM-Image-Loader

Wie schon im letzten Info erwähnt, muß man das Model 4P, wenn es als Model 3 laufen soll, mit einem "ROM" füttern. Dazu liefert Tandy das File MODELD/III auf einer TRSDOS 1.3-Diskette mit. Diese legt man in den Bootdrive und drückt RESET (ev. Optionen könnt ihr im letzten Info nachlesen!).Die Procedur des Ladens dauert ziemlich lange, was angesichts des recht kurzen Laders im Boot-ROM nicht weiter verwunderlich ist.

Dieser Umstand allein hätte mich aber bestimmt nicht dazu gebracht, einen anderen Weg zum Laden des ROM's zu suchen. Den eigentlichen Antrieb erhielt ich erst, als ich feststellen mußte, daß Tandy aus mir unerfindlichen Gründen nicht etwa das Original Model 3-, sondern ein geändertes ROM mitliefert! Wäre dieses "ROM" eine Verbesserung des M. 3-ROM, könnte man die Sache ja noch verstehen, aber das Gegenteil scheint der Fall zu sein!

Während Tandy im Original Model 3-ROM z.B. eine Abfrage der Funktionstasten vorgesehen (aber nicht aktiviert) hat, ist diese Möglichkeit für's Model 4 nicht mehr vorgesehen. Noch unsinniger erscheint dieser Umstand wenn man weiß, daß das Model 3 keine Funktionstasten eingebaut hatte, die bekam erst das Model 4 spendiert. Ähnlich verhält es sich mit der Druckroutine des Model 3. Hier haben die ROM-Macher eine Tabelle zum Umsetzen der internen Codes in die entsprechenden Druckercodes eingebaut. Diese Maßnahme erscheint relativ unsinnig, da man die Tabelle im ROM ja nicht ändern kann. Beim Model 4 ist diese Tabelle nicht mehr zu finden, obwohl sie hier wohl angebrachter (weil im "Pseudo-ROM änderbar!) gewesen wäre.

Es gibt noch einige solcher Beispiele, die aus Platzgründen nicht alle hier aufgezählt werden sollen. Erwähnen will ich nur noch, daß auch manche Programme nur mit den Original-ROM's laufen. So verweigert z.B. TASMON (für Model 3) seine Arbeit sowohl mit dem MODELA- als auch mit dem MODELD-ROM und selbst mit einem Original Model 3-ROM für die deutsche Tastatur gibt es erhebliche Probleme. Richtig funktioniert das Programm nur mit dem amerikanischen Original-ROM des Model 3. Grund genug finde ich, die verschiedenen ROM's auf einer Diskette bereitzuhalten und je nach den gegebenen Bedürfnissen zu wählen, welches man benutzen will!

Leider sieht das Boot-ROM des 4p diese Möglichkeit nicht vor. Es erkennt immer nur das File mit dem Namen "MODELD/III" als das zu ladende und ignoriert alle anderen ROM-Images. Man müsste also, um immer auf alles gefasst zu sein, vier "ROM"-Disketten bereithalten. Diese Platzverschwendung veranlasste mich, eine Lösung für das Problem zu suchen (und zu finden).

Wie oben schon erwähnt, ist der ROM-Lader nicht gerade der Schnellste. So lädt z.B. TRSDOS 6.2 oder CP/M ein 12k langes Programm (und nichts anderes ist das ROM-Image) in weniger als einem Drittel der Zeit, die der Lader dazu braucht. Aus dieser Erkenntnis heraus entstand folgendes Konzept:

1. Alle verfügbaren ROM-Images werden auf einer TRSDOS 6.2 - Systemdiskette gespeichert und
2. von hier in den Speicher geladen und gestartet!

Probleme bei der Realisierung des Konzeptes:

1. Die ROM-Images können nicht direkt in ihren tatsächlichen Speicherbereich (0000h - 3000h) geladen werden, da hier TRSDOS residiert und überschrieben werden würde!
2. TRSDOS benutzt eine andere Speicheraufteilung als die, die für den Model 3 - Modus notwendig ist!
3. Der Stackpointer des TRSDOS liegt mitten im späteren ROM-Bereich und kann dort ganz schön für Durcheinander sorgen!

Lösungen:

zu 1.:

Die ROM's wurden mittels eines kleinen Maschinenprogramms (siehe Listing 1) nach oben (7000h) verschoben. Dazu wird das entsprechende ROM auf die alte Weise geladen, NewDOS gebootet und dann ROMSHIFT/CMD gestartet. Danach wird TRSDOS 6.2 gebootet und das ROM unter dem entsprechenden Namen auf die Diskette gedumpt.

Beispiel: DUMP ROM1/CIM (START=X'7000', END=X'A000h').

Die Original Model 3-ROM-Images muß man sich natürlich zuvor auf ähnliche Weise aus einem Model 3 beschaffen!

Die ROM's werden später von dem Programm "ROM-Lader" (siehe Listing 2) auf Befehl geladen und, nachdem die zu bootende Model 3-Diskette eingelegt wurde, wieder auf ihren richtigen Platz verschoben und gestartet (Z. 76 - 82)!

zu 2.:

Das Model 4p (und natürlich auch das Tischmodel) wurde, um mit dem Model 3 kompatibel zu bleiben, mit einer raffinierten Bankinglogik ausgestattet. über Steuerports kann man die RAM-Konfigurationen so verändern, daß entweder die Memory-Map des M. 3 (ROM bis 3000h, danach Keyboard, Videospeicher und dann RAM bis FFFFh) eingestellt ist oder aber die Tastatur und der Video-RAM nach oben "unters Dach" (2k Video-RAM von F7FFh - FFFFh, Tastatur von F3FF-F7FE) verschoben wird. Das ist die Konfiguration für TRSDOS und CP/M.

Um das Model 4p aus dem TRSDOS-Modus in den M. 3-Modus zu versetzen, muß nach dem Verschieben des ROM-Images auf den Port 84h der Wert 00 ausgegeben (Z. 80 und 81) werden (06 im TRSDOS-Mode).

zu 3.:

Um dem Stack keine Möglichkeit zu geben im "ROM" Unheil anzurichten (wie einigemale bei Vorversuchen passiert!), wird der Stackpointer (CPU-Register SP), kurz vor dem Verschieben des ROM-Images auf seinen alten Platz, nach oben (E000h) verlegt.

Zu dem Programm in Listing 2 ist eigentlich nicht viel zu sagen, es erklärt sich mittels der Kommentare praktisch von selbst. Eine kurze Anmerkung noch zu den sog. Supervisor-Calls. Das sind für den Programmierer nutzbare DOS-Routinen, ähnlich denen in NewDOS oder im BASIC-ROM, die allerdings nicht wie dort üblich mit "CALL ADDR" aufgerufen werden. Um einen Supervisor-Call aufzurufen wird dessen Nummer in den Accu geladen und dann ein ReStarT 28h durchgeführt. Sind Parameter an den SVC zu übergeben, so geschieht das über die anderen CPU-Register (zum Beispiel Zeilen 13 u. 61 - 64).

Hartmut Obermann

Listing 1:

```
                10   ORG    5200h
5200 210070     20   LD     HL,0000H    ;ROM-Image von 0000h
5203 110000     30   LD     DE,7000H    ;nach 7000h
5206 010038     40   LD     BC,3800H
5209 EDB0       50   LDIR               ;verschieben!
520B C32D40     80   JP     402DH       ;Rücksprung ins DOS
5200            90   END    START
```

m4/4p ecke m4/4p ecke

Listing 2

```
             ;*********************************************************************
             ;*                    Model 4p - ROM-Lader                           *
             ;*   Lädt verschiedene ROM-Versionen von einer TRSDOS 6.x - Diskette  *
             ;*********************************************************************
             ;* Hartmut Obermann, Schwalbacher Str. 6, 6209 Heidenrod, 06124 / 3913 *
             ;*********************************************************************
             ;
5000                 ORG     5000H              ;Hier (oder anderswo) solls sein!
             ;
             ;************** Supervisor-Calls *************************************
             ;
000A         §DSPLY: EQU     10                 ;Text auf Screen
004E         §FSPEC: EQU     78                 ;FCB erstellen
0001         §KEY:   EQU     1                  ;Tastatur abfragen
004C         §LOAD:  EQU     76                 ;File laden
0069         §CLS:   EQU     105                ;Bildschirm löschen
             ;
             ;************** Definitionen *****************************************
             ;
5000         FCB:    DEFS    32                 ;Platz für FileControlBlock
5020 52      ROM:    DEFM    'ROM'              ;Name des einzulesenden Files
5023 20      NUM:    DEFM    ' '                ;Nummer des einzulesenden Files
5024 2F              DEFM    '/CIM'             ;und dessen Extension
5028 0D              DEFB    0DH                ;Ende des Filenames
5029 4D      TEXT1:  DEFM    'Model 4P - ROM-Lader'  ;Hier steht was,
503D 0A              DEFB    0AH
503E 2D              DEFM    '--------------------'
5052 0A              DEFB    0AH
5053 43              DEFM    'C by Hartmut Obermann'  ;und von wem es ist!
5068 0A              DEFB    0AH,0AH
506A 4D              DEFM    'MODELA/III        = 1'  ;Und was man machen kann!
507E 0A              DEFB    0AH
507F 4D              DEFM    'MODELD/III        = 2'
5093 0A              DEFB    0AH
5094 4D              DEFM    'MODEL3US/ROM      = 3'
50A8 0A              DEFB    0AH
50A9 4D              DEFM    'MODEL3GE/ROM      = 4'
50BD 0A              DEFB    0AH,0DH            ;Ende des Textes
50BF 52      TEXT2:  DEFM    'ROM-Image ist geladen!
                              Diskette wechseln und <ENTER> drücken!!!'
5107 0D              DEFB    0DH
             ;
             ;************** Hauptprogramm ****************************************
```

```
             ;
5108 3E69    START:  LD      A,§CLS             ;Bildschirm löschen
510A EF              RST     28H
510B 212950          LD      HL,TEXT1           ;und Text ausgeben!
510E 3E0A            LD      A,§DSPLY
5110 EF              RST     28H
5111 3E01    ASK:    LD      A,§KEY             ;Tastatur abfragen,
5113 EF              RST     28H
5114 FE31            CP      '1'
5116 280C            JR      Z,LOAD
5118 FE32            CP      '2'                ;bis eine Taste zwischen 1 und 4
511A 2808            JR      Z,LOAD
511C FE33            CP      '3'
511E 2804            JR      Z,LOAD
5120 FE34            CP      '4'
5122 20ED            JR      NZ,ASK             ;gedrückt wurde!
5124 212350  LOAD:   LD      HL,NUM             ;Nummer des zu ladenden Files
5127 77              LD      (HL),A             ;in den Filenamen einfügen!
5128 212050          LD      HL,ROM             ;FileControlBlock initialisieren
512B 110050          LD      DE,FCB
512E 3E4E            LD      A,§FSPEC
5130 EF              RST     28H
5131 210050          LD      HL,FCB             ;und File laden!
5134 3E4C            LD      A,§LOAD
5136 EF              RST     28H
5137 21BF50          LD      HL,TEXT2           ;Aufforderung "<ENTER> drücken!"
513A 3E0A            LD      A,§DSPLY           ;anzeigen und
513C EF              RST     28H
513D 3E01    LOOP:   LD      A,§KEY             ;warten bis <ENTER>
513F EF              RST     28H
5140 FE0D            CP      0DH
5142 20F9            JR      NZ,LOOP            ;gedrückt wurde!
5144 3100E0          LD      SP,0E000H          ;Stack nach "oben" verlegen!
5147 210070          LD      HL,7000H           ;ROM-Image von 7000h
514A 110000          LD      DE,0000H           ;nach 0000h
514D 010038          LD      BC,3800H
5150 EDB0            LDIR                       ;verschieben!
5152 3E00            LD      A,00H              ;Operation-Port richtig
5154 D384            OUT     (84H),A            ;initialisieren
5156 C30000          JP      0000H              ;und "ROM" starten!!!
5108                 END     START
00000 Total errors
```

m4/4p ecke

UHRUNTEN für Model 4p!

Nach der Lektüre des Artikels von Arnulf Sopp "Wenn die Uhr mal stört" (Info Nr. 14 Seite 39ff) , war ich natürlich höchst wissbegierig, wie die Sache sich auch beim Model 4p im Model 3-Mode realisieren läßt. Also nichts wie ran an den Speck!

Aber schon beim Ausprobieren der Funktion CLOCK,Y bekam ich erstmal große Augen; es tat sich (fast) nichts! Die Floppy lief zwar an und auch die eigentliche Aufgabe des Befehls, nämlich ein Flagbit im Byte 4210h zu setzen, wurde (wie ich später nach längerer Spurverfolgung herausfand) einwandfrei erfüllt, aber die Uhrzeit erschien trotz dem nicht auf dem Bildschirm. Ein Fehler im DOS!? Kann nicht sein, wenn ich das MODELA/III-ROM (die Model 4p-ROM-Version für die amerikanische Tastatur) benutze, läuft die Sache einwandfrei. Also ein Bug in der deutschen Version des ROM-Image! Aber wo?

Der Fehler ist schnell (alles relativ!) gefunden. Zum Glück stimmt das Model 4p-"ROM" gerade an dieser Stelle einigermaßen mit dem Original Model 3-ROM überein, so daß das ROM-Listing (Grosser/Röckrath) weiterhelfen kann. Ein einziges Byte ist, durch welchen unglücklichen Umstand auch immer, um den Wert 5 zu niedrig geraten. Um die Uhr wieder zum Laufen zu bekommen, muß also nur an der Speicherstelle 355Dh aus dem Wert 19h ein 1Eh gemacht werden. Das macht man aber nicht im Speicher (da geht's zwar auch, aber nur mit einem kleinen Trick), sondern direkt im ROM-Image-File, damit nicht nach jedem Neuladen der Ärger wieder anfängt!

Und schon geht der Ärger wieder los! Das MODELD/III-File befindet sich auf einer Diskette im TRSDOS 1.3 - Format. Dort kann es nicht ohne weiteres bearbeitet werden, da weder TRSDOS 6.2 noch NewDOS direkten Zugriff auf dieses Format haben und unter TRSDOS 1.3 gibt es keinen Diskeditor (bzw. ich kenne keinen). Also muß das File erst von seiner Originaldiskette auf NewDOS, und nach der Änderung wieder zurück kopiert werden. Dazu muß man den PDRIVE folgendermaßen einstellen:
TI=AM,TD=E,TC=40,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA=2.
Vorsicht! Auch nach der Einstellung dieser Parameter kann man mit "DIR" nicht auf die Diskette schauen (dazu gibt es in der Tandy-Public-Domain das Programm TRSDIR) oder etwa mit SUPERZAP ans Ändern denken. Man kann sich aber mit "COPY MODELD/III:1 :0" das ROM-Image auf seine Floppy holen und nach der Änderung wieder zurückschreiben. Die eigentliche Änderung findet im relativen Sector 36h (54d), relatives Byte 34h (52d) statt. Dort wird, wie oben schon gesagt, statt 19h der Wert 1Eh eingetragen!

Nun aber endlich doch zu "UHRUNTEN"! Ich finde nämlich Arnulfs Idee gar nicht so schlecht, noch dazu da das Problem der Verlegung für 4p-Besitzer sehr einfach zu lösen ist. Man muß nur der ROM-Routine, die die Uhrzeit in die rechte obere Bildschirmecke schreibt, mitteilen, wo sie sie statt dessen hinpacken soll. Das könnte man natürlich auch mittels einer festen Änderung im ROM-Image machen, es gibt aber eine elegantere Möglichkeit.

Wie im Beitrag "ROM-Image-Loader" schon kurz erwähnt, ist das 4p mit einer raffinierten Bankinglogik ausgestattet. Die Logik kann über die Ports 84h (132d) und 9Ch (156d) gesteuert werden.

Mit Hilfe des Port 9Ch kann man das BOOT- ROM ein- bzw. ausblenden (siehe auch "BOOT-ROM des 4p", Info Nr. 14 Seite 63-65). Der Port 84h dient zur Steuerung der RAM-Bänke. An diesem Port interessiert uns zur Zeit nur das Bit 0, mit dem der Bereich zwischen 0000h und 37FFh schreibgeschützt (Bit 0 = 0) oder beschreibbar (Bit 0 = 1) gemacht werden kann. Normalerweise wird für den 3'er-Modus der Port mit dem Wert 00h geladen, also alle Bits auf 0 gestellt und damit das RAM zum Pseudo-ROM, sprich schreibgeschützt. Das kann man durchaus mal mit DEBUG oder vom BASIC aus mit POKE und PEEK probieren, alle Änderungsversuche werden fehlschlagen!

Will man, wie wir in unserem Fall, Änderungen am "ROM" vornehmen, setzt man das erste Bit von Port 84h, indem man ihn mit 01h lädt. Danach lassen sich ohne weiteres Speicherstellen, sowohl mit DEGUG als auch vom BASIC oder von Anwenderprogrammen heraus ändern. Aber Vorsicht! Allzuleicht zerstört man sich dabei die eigene Arbeitsgrundlage und der Rechner verschwindet "im Keller" und es hilft nur erneutes Laden des ROM-Image. Aus diesem Grunde sollte man auch nach kontrollierten Änderungen, wie wir sie jetzt vorhaben, wieder 00h in den Port 84h schreiben, um versehentliche ROM-Änderungen zu vermeiden.

Nun aber endlich zur Sache! Um die Uhr "nach unten" zu verlegen, müssen wir in Adresse 359Eh den gewünschten Wert, sagen wir z.B. 3FF8h, eintragen. Dann landet die Uhr ganz unten rechts, aber natürlich kann man jeden Wert innerhalb (und sogar ausserhalb) des Bildschirmspeichers eintragen, ganz nach eigenem Geschmack. Aber Vorsicht, auch hier lauert wieder einmal eine Gefahr. Wie immer in der Maschinensprache wird die Adresse, und nichts weiter ist der von uns eingetragene Wert, im Format LSB/MSB eingetragen. Hier zur Verdeutlichung zwei kurze Listings in BASIC und Assembler.

```
10 REM UHRUNTEN in BASIC für Model 4p
20 OUT 132,1 : REM Aus Pseudo-ROM mach RAM!
30 POKE 13726,248 : REM LSB der neuen Adresse setzen
40 POKE 13727,63 :  REM MSB der neuen Adresse setzen
50 OUT 132,0 : REM Aus Sicherheitsgründen wieder ROM!
60 CMD"CLOCK,Y" : REM Effekt gleich mal ausprobieren!
```

```
5200         10          ORG     5200h
5200 3E01    20 START    LD      A,00h        ;Aus ROM
5202 D384    30          OUT     (84h),A      ;mach RAM!
5204 21F83F  40          LD      HL,3FF8h     ;die neue Adresse
5207 229E35  50          LD      (359Eh),HL   ;eintragen und
520A 3D      60          DEC     A            ;aus RAM wieder
520B D384    70          OUT     (84h),A      ;ROM machen. Dann
520D C32D40  80          JP      402Dh        ;zurück zum DOS!
             90          END     START
0000 TOTAL ERRORS
```

Natürlich muß man nicht zurück ins DOS springen, sondern kann gleich mit irgendeinem Programm (z.B. SCRIPSIT) weitermachen; aber da fallen euch bestimmt selbst genug Anwendungen ein.

Damit Schluß für diesmal! Viel Spaß beim Zapen und Zapfen, euer

Hartmut Obermann

CALL um die Ecke

Was ich hier vorstellen möchte, ist in dieser Form nur auf dem Genie III s oder vergleichbaren Computern lauffähig. Das Strickmuster, nach dem es aufgebaut ist, eignet sich jedoch für alle in unserem Club vertretenen Maschinen, überhaupt für alle Z80-Rechner. Es geht darum, Speicherbereiche zu erreichen, die ohne Umwege nicht verfügbar sind. Dazu gehören beispielsweise die Boot-EPROMs der neueren Genies, aber auch Programme, die bei einfacheren Computern etwa mit Hilfe eines Bankers im Adreßbereich des ROMs, der Tastatur oder des Bildschirms laufen. Die hierfür notwendigen Umschaltsequenzen kann man sich schenken.

Vorbild waren dafür das Level-4-ROM von TCS (Genie I/II) und CP/M. Im Genie-Sonder-ROM kann ein String auf den Bildschirm ausgegeben werden, der sich unmittelbar hinter dem Ausgabebefehl befindet. So braucht man keinen Zeiger zu laden. CP/M ruft alles Mögliche mit einem CALL nach 0005h auf, je nach dem, was in den Registern übergeben wird. Ähnlich ist es hier:

```
CALL    .call0    ;s. Listing
DEFW    .4080h    ;Unterprogramm-Adresse
```

Mit dieser Befehlssequenz wird ein Unterprogramm an der Adresse 4080h in der normalerweise nicht zugänglichen Bank 0 des G3s aufgerufen. Um das tun zu können, müßte man sonst zuerst die Bank 0 selektieren, dann die Unterprogrammadresse anspringen, zuletzt wieder in die Arbeitsbank (normalerweise 1) zurückkehren. Aber bereits beim Umschalten auf Bank 0 würde sich ein solches Programm selber in die ewigen Jagdgründe banken, es wäre bei RET nicht mehr auffindbar.

Die hier vorgestellte Methode arbeitet im Common-Bereich, der unabhängig von der eingestellten Bank immer selektiert ist. Von hier aus kann die Umschaltung risikolos vorgenommen werden. Die häufiger gebrauchten Umschaltroutinen liegen bereits als Unterprogramme vor, andere wurden (ab Label bank1) hier eigens angefügt. So ist es mit dem G3s nunmehr möglich, Unterprogramme in allen Speicherbereichen aufzurufen. Folgende Routinen besorgen die Selektion:

```
RAM "neben" den memory-mapped Devices:  06A0h
dieses RAM wieder ausblenden:           06ABh
RAM in Bank 0:                          35D4h (Label bank0)
wieder Bank 1 selektieren:              35CDh (Label bank1)
Boot-EPROM selektieren:                 35E5h (Label rom)
Boot-EPROM wieder ausknipsen:           35DEh (Label ram)
ein Unterprogramm in Bank 0 aufrufen:   35BBh (Label call0)
(Restauration des alten Zustandes nicht erforderlich)
```

Das EPROM oder der Speicher im memory-mapped-Adreßbereich sind mit einem gewöhnlichen CALL an die genannten Adressen erreichbar. Nur bei Bank 0 geht das aus den oben skizzierten Gründen nicht, daher ist die Routine call0 implementiert. Wenn sie mit einem CALL aufgerufen wird, steht die RET-Adresse im Stack. Es ist die Adresse unmittelbar hinter dem CALL. Nichts spricht dagegen, dort sonstwas unterzubringen, in unserem Fall die Adresse des gewünschten Unterprogramms. Hiervon geht die Routine call0 aus. Deshalb erhöht sie die RET-Adresse auf dem Stack um zwei Stellen, um die CALL-Adresse zu überspringen. Bei der Rückkehr aus dem Unterprogramm wird nur wieder sinnvoller Code vorgefunden.

Alle genannten Routinen, auch call0 und die zitierten System-Unterprogramme können gefahrlos aufgerufen werden, auch unter BASIC mit USR. Der User sollte jedoch ziemlich genau wissen, was er da tut. So knipst er sich beispielsweise mit dem Programm an 06A0h (s. o.) die Tastatur und den Bildschirm aus. Da hilft nur noch abschalten. Daher muß der Anwender in seinen Programmen immer für eine Restauration des alten Zustands sorgen. Die zuständigen Routinen sind oben benannt. Auch der Interrupt-Status muß bedacht werden. Alle aufgezählten Routinen, die einen besonderen Speicherbereich selektieren, schalten die Interrupts ab. Unter denen, die den Normalzustand restaurieren, läßt nur 06ABh (s. o.) sie anschließend wieder zu. Der User muß deshalb selbst für den gewünschten INT-Status sorgen (unter BASIC mit CMD"T" und CMD"R").

Für andere Computer sind diese Adressen ungültig. Der Leser muß sich jeweils darüber informieren, wie es bei seiner eigenen Maschine aussieht. Dazu gehört natürlich auch die Suche nach einem freien Speicherplatz, an dem neu hinzukommende Routinen untergebracht werden können. Für die weitaus meisten von euch dürfte es der Adreßraum ab 7000h sein, der beim Genie I/II mit dem ES 64 MBA nutzbar gemacht werden kann.

Arnulf Sopp

```
35BB                   ORG     35bbh          ;hier freier Speicher

              ;ein Unterprogramm in Bank 0 anspringen
35BB  E3      call0    EX      (SP),HL        ;HL retten,Stack laden
35BC  D5               PUSH    DE             ;retten
35BD  5E               LD      E,(HL)         ;E <- (Stack)
35BE  23               INC     HL             ;auf Stack + 1
35BF  56               LD      D,(HL)         ;D <- (Stack + 1)
35C0  23               INC     HL             ;auf RET-Adresse
35C1  ED53CB35         LD      (calladr),DE   ;CALL-Adresse patchen
35C5  D1               POP     DE             ;DE restaurieren
35C6  E3               EX      (SP),HL        ;dto. HL. RET-Adr. -> St.
35C7  CDD435           CALL    bank0          ;Bank 0 selektieren
35CA  CD0000           CALL    0000h          ;dort Unterprg. aufrufen
35CB          calladr  EQU     $-2            ;(Adresse variabel)

              ;verschiedene Speicherbereiche zugänglich machen
35CD  F5      bank1    PUSH    AF             ;retten
35CE  DBF9             IN      A,(0f9h)       ;Systembyte 0
35D0  CBF7             SET     6,A            ;Bank 1 selektieren
35D2  1806             JR      exit9          ;dort weiter

35D4  F5      bank0    PUSH    AF             ;retten
35D5  DBF9             IN      A,(0f9h)       ;Systembyte 0
35D7  E63F             AND     3fh            ;Bank 0 selektieren
35D9  F3               DI                     ;INTs sperren
35DA  D3F9    exit9    OUT     (0f9h),A       ;Systembyte neu schreiben
35DC  F1               POP     AF             ;restaurieren
35DD  C9               RET

35DE  F5      ram      PUSH    AF             ;retten
35DF  DBFA             IN      A,(0fah)       ;Systembyte 1
35E1  CBD7             SET     2,A            ;Boot-EPROM ausschalten
35E3  1806             JR      exita          ;dort weiter

35E5  F5      rom      PUSH    AF             ;retten
35E6  DBFA             IN      A,(0fah)       ;Systembyte 1
35E8  CB97             RES     2,A            ;Boot-EPROM selektieren
35EA  F3               DI                     ;INTs sperren
35EB  D3FA    exita    OUT     (0fah),A       ;Systembyte neu schreiben
35ED  F1               POP     AF             ;restaurieren
35EE  C9               RET

0000                   END

00000 Fehler
```

Korrektur zu "CALL um die Ecke"

Man sollte wirklich auch den selbstverständlichsten Kleinkram dreimal checken und ausprobieren, bevor man unschuldige Leser damit frustet. In meinem Programm, das in dem oben zitierten Artikel vorgestellt wird, steckt ein fataler Fehler. Wenn ein Unterprogramm in einer benachbarten Bank aufgerufen wird, muß unbedingt dafür Sorge getragen werden, daß sich der Stack im Common-Bereich befindet. Andernfalls würde nämlich das RET am Ende der Routine die Rückkehradresse in der eigenen Bank suchen – und fatalerweise sogar finden! Es fragt sich bloß, was für eine absonderliche Adresse das ist.

In der hier korrigierten Form läuft das Programm aber einwandfrei, denn der Stack wird nach Common verlagert. Der erste Teil, der unverändert geblieben ist, ist weggelassen. Da sich nun aber die Adressen hinter der Korrektur verschieben, ist der ganze Rest gelistet. Die veränderten Adressen für die in der Tabelle genannten Umschaltroutinen lauten jetzt so:

bank0 (die Bank 0 selektieren) 35D7h
ram (das Boot-EPROM ausschalten) 35EBh
rom (das Boot-EPROM selektieren) 35EFh


```
;verschiedene Speicherbereiche zugänglich machen
bank1   PUSH    AF          ;retten
        IN      A,(0f9h)    ;Systembyte 0
        SET     6,A         ;Bank 1 selektieren
        LD      SP,0000h    ;Stack restaurieren
spbuf   EQU     $-2         ;Adreß-Operand variabel
        JR      exit9       ;dort weiter

bank0   PUSH    AF          ;retten
        LD      (spbuf),SP  ;Stackpointer retten
        LD      SP,37d0h    ;eigener Stack im Common
        IN      A,(0f9h)    ;Systembyte 0
        AND     3fh         ;Bank 0 selektieren
        DI                  ;INTs sperren
exit9   OUT     (0f9h),A    ;Systembyte neu schreiben
        POP     AF          ;restaurieren
        RET

ram     PUSH    AF          ;retten
        IN      A,(0fah)    ;Systembyte 1
        SET     2,A         ;Boot-EPROM ausschalten
        JR      exita       ;dort weiter

rom     PUSH    AF          ;retten
        IN      A,(0fah)    ;Systembyte 1
        RES     2,A         ;Boot-EPROM selektieren
        DI                  ;INTs sperren
exita   OUT     (0fah),A    ;Systembyte neu schreiben
        POP     AF          ;restaurieren
        RET
```

Arnulf Sopp

Real Time Black Box?

Für das Genie III s gibt es eine Echtzeituhr (real time clock, RTC), auf der der Uhren- und Kalenderchip MSM 5832 steckt. Da TCS bzw. Phönix aus Gründen der Wirtschaftlichkeit vermutlich dieselbe Modellpolitik betreibt wie VW (Audi und VW unterscheiden sich hauptsächlich am Firmensignet), ist anzunehmen, daß noch weitere Genies diese Uhr haben. Es macht mich nicht besonders stolz, den UHR- oder DATUM-Befehl korrekt eintippen zu können. Vielmehr muß es möglich sein, ohne Unterstützung des DOS die Uhr in eigenen Programmen auszulesen.

Zu diesem Zweck fragte ich bei Phönix an, erhielt aber außer dem Versprechen, man wolle sich erkundigen, keine Reaktion. So blieb mal wieder nichts anderes übrig, als die Uhr-Routinen des Betriebssystems zu analysieren. Willkommene Unterstützung bot ein Auszug eines Artikels über eine andere Uhr mit dem MSM 5832, woraus ich die Bedeutung der internen Register (s. Tabelle am Ende dieses Textes) ersehen konnte. Die Angabe der Quelle muß ich leider schuldig bleiben, weil ich sie selbst nicht kenne.

Der MSM 5832 hat 13 Register, in denen die in der Tabelle genannten Zeitinformationen im BCD-Format vorliegen. Das bedeutet, daß alle Zeiteinheiten (außer dem einstelligen Wochentag) als zweistellige Dezimalzahlen schreib- bzw. lesbar sind. Für eine Digitaluhr mit LCD-Anzeige ist das zweifellos von Vorteil. In binär "denkenden" Computern wäre allerdings eine duale Codierung sinnvoller. Nun ja, ist nun mal so.

Um zum Stellen oder Auslesen der Uhr die entsprechenden Register zu erreichen, müssen dem Chip ihre 4 Bit breiten Adressen mitgeteilt werden. Zugleich muß eines von zwei weiteren Bits auf 1 stehen, um die Uhr entweder zum Lesen oder zum Schreiben vorzubereiten. Offenbar wird sie völlig deaktiviert (läuft aber intern weiter), wenn als Abschluß einer Schreib-/Leseoperation beide Bits rückgesetzt werden. Über diesen Punkt bin ich mir nicht ganz sicher. Jedenfalls wird in der Genie-Routine so verfahren.

Im Genie III s ist diese Ansprache der Zielregister folgendermaßen gelöst: Der Port 5B erhält einen Output mit dem Bitmuster xxxx1yy0. Die vier x-Bits beinhalten die Adresse des gewünschten Registers. Die Bits y unterscheiden zwischen Schreiben und Lesen. Beim Stellen (Schreiben) der Uhr lauten sie 01, beim Auslesen 10. So wird z. B. das Register 9 (Einerstelle der Monate) zum Schreiben mit dem Output 10011010 (=9A) auf den Port 5B adressiert. 9C würde das Lesen des Monats-Einers ermöglichen.

Nachdem das Zielregister angesprochen ist, kann die Zeiteinheit eingegeben oder ausgelesen werden. Hierfür ist beim Genie III s der Port 5A zuständig. Hierbei ist zu beachten, daß nur die unteren 4 Bit von Belang sind. Außerdem beinhalten die Register 5 und 8 zusätzliche Informationen über Schaltjahre, 12h- bzw. 24h-Anzeige sowie die Tageshälfte für die AM/PM-Anzeige. Die letztlich relevanten Bits müssen deshalb vor der Anzeige maskiert werden. Umgekehrt müssen beim Stellen der Uhr diese Zusatzbits je nach dem gesetzt oder rückgesetzt sein.

Im Anschluß an diesen Text sind die beiden Routinen gelistet, die im Genie III s die Uhr stellen bzw. lesen (Labels wrclock und rdclock). Dabei handelt es sich um einen Auszug aus einem Listing des kompletten residenten Betriebssystems, das ich mir anfertigte, um die leider sehr bescheidene Dokumentation zu ergänzen. Daher fehlen einige Informationen, die mit LIST OFF ausgespart sind. Der Leser möge deshalb die jeweilige Ladeadresse der linken Spalte entnehmen. Labels, die nicht innerhalb des Listings definiert sind, ergeben sich aus der zweiten Spalte von links. Die Programmlogik dürfte sich aus den reichlichen Kommentaren ergeben.

Das Programm liegt "hinter" dem Bildschirm, ist also nicht ohne weiteres für Änderungen zugänglich. Zuvor muß auf den Port FA ein Byte mit gesetztem Bit 0 ausgegeben werden (also eine ungerade Zahl). Mit CALL 06A0 erledigt das auch das Betriebssystem. Nach der Manipulation wird mit CALL 06AB oder einem geradzahligen Output auf Port FA der Screen wieder zugeschaltet. Vorsicht mit diesem Port! Ein falscher Output kann die Arbeit von Stunden zunichte machen. Zuerst den Input lesen, dann das Bit 0 ändern, dann neu ausgeben!

Sagte ich weiter oben "Logik"? Der Algorithmus ist zwar wirklich ziemlich elegant und läßt sich kaum nennenswert verkürzen oder beschleunigen. Daß aber in Zeile 745 die Sekunden immer auf 0 gestellt werden, obgleich das DOS ohne weiteres die Eingabe von beispielsweise 13:54:32 zuläßt, ist nicht einzusehen. Es ist auch im Betriebssystem nicht vorgesehen, Schaltjahre zu berücksichtigen oder auch den Wochentag zu programmieren. Das ist peinlich, wenn CP/M ihn anzeigt - natürlich mit siebenfacher Chance den falschen. Mit dieser kleinen Routine läßt er sich ein- für allemal einstellen:

```
        ORG     5200h
start   CALL    06b5h           ;auf 1,78 MHz schalten
        LD      A,6ah           ;Register 6 schreiben
        OUT     (5bh),A         ;Befehl ausgeben
        LD      A,7?            ;Beispiel hier: Sonntag
        OUT     (5ah),A         ;Ausgabe an die Uhr
        XOR     A               ;A <- 00, Uhr "stillegen"
        OUT     (5bh),A         ;Befehl ausgeben
        JP      06beh           ;7,2 MHz und Ende
        END     start           ;dort Einsprung
```

Die Triebfeder für mich, den Wecker auf die Couch zu legen, war zunächst reiner Wissensdurst. Der Leser möchte jedoch vielleicht etwas damit anfangen. Bei zeitkritischen Messungen kann beispielsweise das Register 0 via Polling einen Sekundentakt liefern, der von den Systeminterrupts unabhängig ist. Wenn für kaufmännische Anwendungen die Wochentage interessieren, die vom DOS aus nicht abgerufen werden können, so sind sie, wie ich hoffe, nach meinen Erläuterungen nun ebenfalls programmgesteuert verfügbar. Offen gestanden fällt mir im Moment nicht mehr dazu ein. Vielleicht lesen wir im nächsten Info einen interessanten Vorschlag von einem anderen Mitglied?

Tabelle zur Adressierung des MSM 5832:

Register	Datenbits D3	D2	D1	D0	Bedeutung
0	x	x	x	x	Sekunden Einer
1		x	x	x	Sekunden Zehner
2	x	x	x	x	Minuten Einer
3		x	x	x	Minuten Zehner
4	x	x	x	x	Stunden Einer
5	a	b	x	x	Stunden Zehner *)
6		x	x	x	Wochentag (Montag = 0)
7	x	x	x	x	Datum Einer
8		c	x	x	Datum Zehner *)
9	x	x	x	x	Monat Einer
A				x	Monat Zehner
B	x	x	x	x	Jahr Einer
C	x	x	x	x	Jahr Zehner

*) a: Register 5, Bit 3 gesetzt im 24h-Format
b: Register 5, Bit 2 gesetzt ab Mittag (PM-Flag)
c: Register 8, Bit 2 gesetzt in einem Schaltjahr

Arnulf Sopp

```
                ; Routinen zum Stellen und Lesen der Hardware-Uhr des 63s
                ; (C) by TCS Computer GmbH
3E97  CDB506    wrclock CALL  lsp           ;auf 1,78 MHz umschalten
3E9A  214140            LD    HL,time       ;Uhrzeit, Sekunden (RAM)
3E9D  0E02              LD    C,02h         ;2 Durchl. (Zeit, Datum)
3E9F  3E23              LD    A,23h         ;= INC HL
3EA1  32BC3E            LD    (incdec1),A   ;dort patchen
3EA4  160A              LD    D,0ah         ;WR ab RTC-Reg. 0 (Sek.)
3EA6  0603      wr3val  LD    B,03h         ;Zähler drei Werte
3EA8  7E        wr1val  LD    A,(HL)        ;Sekunden usw. laden
3EA9  D9                EXX                 ;Register retten
3EAA  6F                LD    L,A           ;HL <- Sekunden
3EAB  2600              LD    H,00h         ;als 16-Bit-Wort
3EAD  3E0A              LD    A,0ah         ;Divisor 10
3EAF  CDB44C            CALL  diva          ;Sekunden/10, Einer in A
3EB2  D9                EXX                 ;Register tauschen
3EB3  CD193F            CALL  wrrtc         ;RTC-Register schreiben
3EB6  D9                EXX
3EB7  7D                LD    A,L           ;Sekunden, Zehner
3EB8  D9                EXX
3EB9  CD193F            CALL  wrrtc         ;RTC-Register schreiben
3EBC  23        incdec1 INC   HL            ;Uhrzeit Minuten usw.
3EBD  10E9              DJNZ  wr1val        ;bis ss/mm/hh fertig
3EBF  167A              LD    D,7ah         ;ab RTC-Reg. 7 (Datum)
3EC1  3E2B              LD    A,2bh         ;= DEC HL
3EC3  32BC3E            LD    (incdec1),A   ;dort patchen
3EC6  2E46              LD    L,46h         ;Datum Tag (RAM 4046)
3EC8  0D                DEC   C             ;von Zeit auf Datum
3EC9  20DB              JR    NZ,wr3val     ;falls noch nicht geschr.
3ECB  AF                XOR   A             ;A <- 0. 0 Sek. (Quatsch)
3ECC  160A              LD    D,0ah         ;RTC-Reg. 0 schreiben
3ECE  CD193F            CALL  wrrtc         ;RTC-Schreibroutine
3ED1  CDBE06            CALL  hsc           ;auf 7,2 MHz umschalten
3ED4  DDE5      rdclock PUSH  IX            ;retten
3ED6  DD212C3F          LD    IX,clktab-1   ;Ziffernmasken-Tabelle
3EDA  2E44              LD    L,44h         ;Datum Jahr (RAM 4044)
3EDC  0E02              LD    C,02h         ;2 Durchl. wie oben, RD
3EDE  3E23              LD    A,23h         ;= INC HL
3EE0  32F53E            LD    (incdec2),A   ;dort patchen
3EE3  160C              LD    D,0cch        ;ab RTC-Reg. 12 (Jahr)
3EE5  0603      rd3val  LD    B,03h         ;Zähler 3 Werte
3EE7  CD0A3F    rd1val  CALL  rdrtc         ;RTC auslesen (Zehner)
3EEA  87                ADD   A,A           ;*2
3EEB  5F                LD    E,A           ;sichern
3EEC  87                ADD   A,A           ;*4
3EED  87                ADD   A,A           ;*8
3EEE  83                ADD   A,E           ;*10
3EEF  5F                LD    E,A           ;sichern
3EF0  CD0A3F            CALL  rdrtc         ;RTC auslesen (Einer)
3EF3  83                ADD   A,E           ;+ Zehnerstelle
3EF4  77                LD    (HL),A        ;in den Puffer (RAM 404x)
3EF5  2B        incdec2 DEC   HL            ;1 Stelle tiefer
3EF6  10EF              DJNZ  rd1val        ;bis 3 Werte fertig
3EF8  165C              LD    D,5ch         ;ab RTC-Reg. 5 (Stunden)
3EFA  3E2B              LD    A,2bh         ;= DEC HL
3EFC  32F53E            LD    (incdec2),A   ;alten Befehl restaur.
3EFF  2E43              LD    L,43h         ;Zeit Stunden (RAM 4043)
3F01  0D                DEC   C             ;von Datum auf Zeit
3F02  20E1              JR    NZ,rd3val     ;falls noch nicht gelesen
3F04  DDE1              POP   IX            ;restaurieren
3F06  AF                XOR   A             ;A <- 00, RTC rücksetzen
3F07  D35B              OUT   (5bh),A       ;auf RTC-Adreßport
3F09  C9                RET
3F0A  7A        rdrtc   LD    A,D           ;RTC-Adreßreg., RD/WR-Bit
```



```
3F0B  D35B              OUT   (5bh),A       ;Register adressieren
3F0D  1610              LD    D,10h         ;1 tiefer im ober. Nibble
3F0F  92                SUB   D
3F10  57                LD    D,A           ;D <- neues Register
3F11  DD23              INC   IX            ;nächste BCD-Ziffernmaske
3F13  DB5A              IN    A,(5ah)       ;RTC Register auslesen
3F15  DDA600            AND   (IX+00h)      ;relevante Bits maskieren
3F18  C9                RET
3F19  5F        wrrtc   LD    E,A           ;Zeit/Datum retten
3F1A  7A                LD    A,D           ;RTC-Adreßreg., RD/WR-Bit
3F1B  D35B              OUT   (5bh),A       ;RTC-Register adressieren
3F1D  1610              LD    D,10h         ;1 höher im oberen Nibble
3F1F  82                ADD   A,D
3F20  57                LD    D,A           ;D <- neues Adreßregister
3F21  E6F0              AND   0f0h          ;nur oberes Nibble
3F23  FE60              CP    60h           ;Std.-Zehner adressiert?
3F25  7B                LD    A,E           ;Zeit/Datum
3F26  2002              JR    NZ,setrtc     ;falls anderes Digit
3F28  CBDF              SET   3,A           ;24h-Anzeige-Bit setzen
3F2A  D35A      setrtc  OUT   (5ah),A       ;Zeit stellen
3F2C  C9                RET
                ; Tabelle von Bitmasken für die Zeiteinheiten:
3F2D  0F        clktab: DB    0fh,0fh       ;Jahr Zehner, Einer
3F2F  01                DB    01h,0fh       ;Monat Zehner, Einer
3F31  03                DB    03h,0fh       ;Tag Zehner, Einer
3F33  03                DB    03h,0fh       ;Stunde Zehner, Einer
3F35  07                DB    07h,0fh       ;Minute Zehner, Einer
3F37  07                DB    07h,0fh       ;Sekunde Zehner, Einer

00000 Fehler
```

Das ist also interaktive Software! . . .

RPNL - Die andere Art zu programmieren

RPNL - was ist das ?

Bevor ich zum Thema komme sei gesagt, daß RPNL nicht auf meinem 'Mist' gewachsen ist. Ich habe den Compiler nur an Newdos/80 angepaßt, bzw. nicht anpaßbare Teile (wegen CP/M) neugeschrieben (Disk-Befehle und die Ein-/Ausgabe etc.). Verbrochen wurde der Compiler von Herr Dipl.-Ing. G. Wostrack. Soviel zum Copy-Right und nun zur Sache.

RPNL bedeutet ausgeschrieben 'Reverse Polish Notation Language'. Als Eltern standen PASCAL und FORTH Pate, d.h. der Compiler-Befehlssatz ist an den von PASCAL angelehnt, die innere Struktur und die Art wie programmiert, wird stammt von FORTH. Herrausgekommen ist ein Sprachkonzept, das soweit möglich nur die guten Seiten von beiden Eltern in sich vereinigt. Hierzu gehört zum Beispiel, daß die Programme strukturiert aufgebaut werden können. Es gibt Unterprogramme, Funktionen und das Hauptprogramm, welches alle Einzelteile miteinander verbindet. Für diejenigen, die ihre Probleme besonders schnell erledigt wissen wollen, können auch in Assembler geschriebene Routinen eingebunden werden. Dies erfolgt natürlich alles nach RPNL-Konvention.

Als Schleifen-Strukturen stehen REPEAT ... UNTIL ... LOOP und die FOR ... LOOP zur Verfügung. Mit ihnen können praktisch alle vorkommenden 'Wiederholungs-Probleme' erledigt werden. Verzweigungen lassen sich einfach über IF ... ELSE ... ENDIF realisieren.

Wie FORTH, so kennt auch RPNL Dicenaries. Es gibt von ihnen insgesamt drei: Das Anwender-, das Compiler- und das Declarations-Wörterbuch. Der Programmierer hat jedoch nur auf jenes Zugriff, in das der Compiler die neu definierten Worte ablegt - das Anwender-Wörterbuch. Die verbleibenden beiden Wörterbücher sind nicht erweiterbar, sie können nur während der Compilierung vom Compiler aufgerufen werden.

Der hervorstechenste Unterschied zu FORTH ist jedoch die Tatsache, daß sich mit RPNL eigenständige Programme erzeugen lassen (mittels 'SAVE name'), die direkt vom DOS-Level aus gestartet werden können. Es muß also nicht jedesmal das gesamte 'System' geladen werden (wie z.B. zum Starten eines Basic-Programmes). Ermöglicht wird dies durch aufteilen des Systems in zwei Teile: dem Runtime-Package und dem Compiler. Das Runtime-Package beginnt am unteren Ende ab 5600H, der Compiler liegt am oberen Ende des Speichers ab F000H. Für eigene Treiber stehen z.Zt. ca. 500 Byte zur Verfügung (ab FE00H bis FFFFH). Das Anwenderwörterbuch wächst von F000H nach unten im Speicher.

Wird nun ein Programm compiliert, so wird der erzeugte Code an das Runtime-Package angehängt. Programmnamen etc. werden parallel in das Dicenary eingetragen, sodaß das System stets über alles informiert ist. Verdaulich sind für den

Compiler jedoch nur reine ASCII-Files ohne Zeilen-Nummern !. Es sollte also ein Texteditor, der ASCII-Files erzeugt, benutzt werden. Die max. Zeilenlänge ist auf 126 Zeichen begrenzt.

Wie wird programmiert ?

Die Programmerstellung erfolgt häppchenweise. Da in jeden Programm Variable, Konstante und dergleichen vorkommen, erfolgt die Definition dieser Dinge vor ihrem ersten Gebrauch. Hierfür wird die Anweisungsklammer DECLARE/DEND benutzt. Das sieht dann folgendermaßen aus:

```
DECLARE
   VAR TEMP     CONST TEST 100
   ARRAY BILD 220
   STRING FRAGE 'Wie Bitte ?'
DEND
```

Das gezeigte Beispiel enthält alle Datentypen die RPNL kennt. Ich werde sie der Reihe nach abhandeln. Zuvor aber noch einige Worte zur Schreibart. RPNL benötigt KEINE Zeilennummern, auch ist es egal wie der Programmtext gestalltet ist. Das obige Beispiel könnte auch ebensogut in einer Zeile geschrieben sein (sofern sie in den Input-Buffer paßt !). Ob eingerückt wird oder nicht ist gleichgültig, auf derelei Dinge legt der Compiler keinen Wert. Die Schreibweise der Anweisungen interessiert ihn jedoch schon. Der bestehende Befehlssatz ist in Großbuchstaben verfaßt und muß auch so geschrieben werden, andernfalls werden die Anweisungen nicht erkannt. Neue Definitionen können dafür aber durchaus in gemischter Groß/Klein-Schreibung erfolgen, müßen dann aber in der einmal festgelegten Form benutzt werden !.

Nun zum Beispiel. Zuerst die Anweisung DECLARE selbst. Hier handelt es sich um eine Compiler-Anweisung, die das Flag 'COMPILE' auf FALSE und somit das System in den 'Compile-Modus' versetzt. Dannach wird das Dicenary 'DECLST' eröffnet. In diesem befinden sich alle gültigen Deklarations-Worte. Die Anweisung DEND nimmt eine Sonderstellung ein. Es ist in der Liste ebenfalls enthalten und hat die Aufgabe, das Compile-Flag wieder auf TRUE zu setzen und so den Compile-Modus zu beenden. Erkennt der Compiler nun beim Durchsuchen seiner Liste einen der Befehle, so wird die zugehörige Routine aufgerufen. Diese führt nun alle Aktivitäten durch, die notwendig sind, um die nachfolgenden Parameter zu verarbeiten.

VAR TEMP
Diese Anweisung bewirkt, daß im Dicenary ein Eintrag mit dem Namen TEMP erfolgt, gefolgt von einer Adresse. Diese Adresse dient bei späteren Bezugnamen auf diese Variable dazu, um an deren Inhalt herranzukommen. Die exakten Verhältnisse sind noch etwas komplizierter, da es sich hier um ein System von 'Zeiger auf Zeiger auf ein Programm' handelt (Quasie von hinten rum durch die Küche ins Auge).

CONST TEST 100
Bei CONST liegen die Verhältnisse ähnlich, nur wird hier statt der Adresse auf den Speicherplatz der Variablen, der Wert der Konstanten auf den Stack gebracht. Während bei einer Variablen-Definition diese automatisch mit Null initialisiert wird, erhält die Konstante natürlich einen Wert. Dieser muß hinter dem Konstanten-Namen (hier TEST) angegeben werden.

ARRAY BILD 220
Vom Typ her ist die Anweisung ARRAY eine Variable. Lediglich bei Reservierung des Speicherplatzes wird anders vorgegangen. Es werden statt einem 220+1 16Bit Plätze vereinnahmt. Plus 1 deshalb, weil der Platz 0 mitzählt. Später im Programm wird bei Aufruf von BILD ein Zeiger auf den ersten (!) Speicherplatz des Arrays auf dem Stack übergeben. Dieser Umstand ist zu berücksichtigen, da es nämlich bedeutet, daß der Anwender seine eigenen Berechnungen für den jeweiligen Array-Platz durchführen muß. Man hätte auch anders verfahren können, dieser Weg bietet jedoch die größte Verwendungsvielfalt. Eine Überwachung der Grenzen (0 und 220) erfolgt nicht (!!!), hierrauf hat der Programmierer selbst zu achten.

STRING FRAGE ´Wie Bitte ?´
Und als letztes die Anweisung STRING. Mit ihr kann dem Wort FRAGE eine Zeichenkette zugewiesen werden. Diese wird wie folgt vom Compiler im Speicher abgelegt:

- Ein Byte als Längenzähler (0...255)
- Die angegebene Zeichenkette (ohne CR oder andere Stringendekennungen !!!)

Der bei Aufruf übergebene Zeiger zeigt auf den Längenzähler. Natürlich erzeugt der Compiler auch hier wieder einen Eintrag im Dicenary und hängt den erzeugten Programm-Code an das Runtime-Package an.

Das PROGRAM
Mit der DECLARation allein ist noch nicht viel anzufangen. Interessant wird die Sache erst mit der Anweisung PROGRAM. Mit ihr kann ´programmiert´ werden. PROGRAM startet den Compiler über das bereits genannte Flag und eröffnet das Anwender-Wörterbuch. Die nachfolgende Zeichenkette wird als Name des Programms in das Dicenary eingetragen. Alle nun folgenden Worte werden als Anweisungen aufgefaßt und compiliert. Ausgenommen hiervon sind Kontrollanweisungen wie IF...ELSE...ENDIF, FOR...LOOP etc. Sie veranlassen den Compiler zu besonderen Aktionen, die unter Umständen eine ganze Kette von Anweisungen dem Runtime-Package hinzufügen. Mit END wird der Compiler wieder verlassen (END ist in der Wirkung identisch mit DEND). Zum besseren Verständnis nun ein Beispiel:

```
PROGRAM BEISPIEL
  ;Gebe die Zahlen von 1 bis 100 aus
  100 1             ;Laufbereich der Schleife
  FOR               ;Anfang der FOR-Schleife
     I TEMP :=      ;Laufindex I nach TEMP
     TEMP PRINT     ;Inhalt von TEMP ausgeben
  LOOP              ;Ende der FOR-Schleife
END                 ;Programmende
```


Das Semikolon wird in RPNL dazu benutzt, Kommentare zu kennzeichnen, ein Blank zum nachfolgenden Text ist nicht erforderlich. Der Aktuelle Laufindex kann mit dem Wort ´I´ auf den Stack geholt werden. Das man ihn von dort nicht unmittelbar ausgeben kann, ist beabsichtigt. Es soll mit der Zwischenspeicherung in einer Variablen eine bessere Lesbarkeit der Programme erreicht werden. Im gewählten Beispiel kommt diese Absicht nicht so deutlich zum Tragen, der Vorteil zeigt sich erst bei umfangreicheren Programmen. Dort ist es dann schon von Interesse sein, stets zu wissen, wo die Zahlen geblieben sind.

Wer bereits einmal in FORTH programmiert hat, wird wissen, wie schnell der Überblick über den Datenbestand auf dem Stack verloren gehen kann. Nach meinen bisherigen Erfahrungen bedeutet dieser ´kleine Umweg´ keinen Nachteil, denn das Hin- und Herschaufel von Zahlen auf dem Stack beansprucht in der Regel die gleiche Zeit und ist unübersichtlicher.

Assembler wird mit CODE und CALL erst schön
Wer es besonders eilig hat, wird eine Lösung in Assembler vorziehen. Hierfür gibt es die Anweisungen CODE & CALL. Beide Befehle rufen wiederrum den Compiler auf, unterliegen dabei aber einer gewissen Rangordnung. Es gilt:

Ebene 1:	**PROGRAM**
Ebene 2:	**CODE**
Ebene 3:	**CALL**

Die Priorität verläuft vertikal, wobei eine Ebene nur jeweils die nächsttiefere oder eigene Ebene aufrufen kann. Das Überspringen einer Ebene ist nicht erlaubt und erzeugt nicht lauffähige Programme.

Da RPNL z.Zt. noch keinen Inline-Assembler enthält, muß der Maschinen-Code in Hexform eingegeben werden, was so aussieht:

```
CALL $BEISPIEL3
   ...
   C9              ;das notwendige RET !
CEND

CODE BEISPIEL2
   C1 D1 E1 C5 CD $BEISPIEL3
   5F E1 ...
CEND
```

Wie zu erkennen, dürfen die Hex-Zahlen nur zweistellig sein. Abweichungen hiervon werden mit einer Fehlermeldung geahndet.

Die Länge des Assemblerteils ist nicht festgelegt. Das liegt im Ermessen des Programmierers. Eine Besonderheit stellt der Assemblerbefehl CD $BEISPIEL3 dar. Hier wird ein Unterprogramm Namens $BEISPIEL3 aufgerufen. CODE sucht nun im Anwender-Dicenary nach diesem Namen und trägt dessen zugehrige Code-Adresse im erzeugten Programm ein. Das Dollarzeichen vor dem Namen ist nicht zwingend, ich benutzte es nur zum Kennzeichnen von Unterprogrammen, die mit CALL...CEND erzeugt wurden. Diese Vorgehensweise ist so unmittelbar nur bei Unterprogrammen gangbar. Soll der Inhalt einer Variablen direkt per Maschinenbefehl in eines der Register geladen werden, so ist noch einges zusätzlich zu tun. In diesem Fall muß die geladene Adresse um zwei erhöht werden. Dann erst kann unter Benutzung der 'neuen' Adresse der Variableninhalt in ein Register geholt werden. Oder anders:

```
Reg. <- Adr.
Reg. = Reg. + 2
Reg. <- (Reg.)
```

Das ist zugegeben sehr umständlich, aber in diesem Fall schlägt die innere Strucktur von RPNL voll auf den Programmierer zurück. RPNL ist eben nicht FORTRAN oder BASIC. Dafür hat man dann aber auch die Möglichkeit den ganzen Abfrage-Overhead dieser Sprachen nicht mitschleppen zu müßen.

Die Anweisung CEND schließlich schaltet den Compiler wieder aus. Die Anweisungsklammer CALL...CEND arbeitet nach dem gleichen Schema. Es muß aber unbedingt am Ende des Assemblerteils ein RET (= C9H) angefügt werden, sonst rennt der Z80 ins Leere (weil ja ein Unterprogramm !). Zu beachten ist hier ganz besonders, daß der Datenstack über den Stackpointer (SP) der CPU gebildet wird und somit die Rücksprung-Adresse zuoberst auf dem Stack liegt.

COMPILE name

Ist das Programm fertig, so kann es jetzt mit COMPILE übersetzt werden. Der Syntax für 'name' ist wie gewohnt. Lediglich ein Passwort darf nicht benutzt werden, dieses wird z.Zt. nicht akzeptiert. Das Programm wird an das Runtime-Modul ancompiliert und kann bei bedarf mit SAVE auf Diskette gesichert werden.

Treten bei der Compilierung Fehler auf, so wird abgebrochen und eine Fehlermeldung auf dem Bildschirm ausgegeben. Das fehlerhafte Programm wird in diesem Fall gelöscht und das System reinitialisiert. Eine Extension oder Laufwerk müßen nicht angegeben werden, hier werden vom Compiler optional '/RPN' und vom DOS gemäß der Eintragung unter SYSTEM das Laufwerk eingetragen.

SAVE name

Ist das Programm endlich compiliert, so möchte man es auch noch auf Diskette abspeichern. Hierfür ist die Anweisung SAVE mit nachfolgendem Programmnamen zuständig. Für den Namen gelten die Newdos-Konventionen mit Ausnahme der Passwörter. Diese können

z.Zt. nicht benutzt werden. Aber das läßt sich wohl verschmerzen. Die Extension und das Laufwerk müßen nicht unbedingt angegeben werden, diese werden automatisch angehängt (/Ext = CMD, das Laufwerk erledigt Newdos gemäß SYSTEM). Damit stände es nun auf der Diskette und kann von DOS-Ready aus aufgerufen werden.

Es ließe sich noch einiges mehr zu RPNL sagen, aber der Redaktionsschluß und mein Urlaubstermin nahen. Als Praxisteil nun noch zwei Programme in RPNL, in denen alles zusammengefaßt ist. Im ersten Programm werden darüberhinaus auch Diskbefehle verwendet. Wer es noch ausführlicher beschrieben lesen will, der besorge sich das Buch.

/1/ Vostrack, Gustav
RPNL : eine FORTH-ähnliche Sprache mit struktur-unterstützenden Sprachkonstrukten / Gustav Vostrack. - Sprendlingen : Luther, 1984.
ISBN 3-88707-022-4

Tschüß

Das Programm PRINTF gibt einen ASCII-File inklusive der Steuercodes auf dem Monitor oder Drucker aus.

```
DECLARE
        VARF FILE               VAR TEMP
        VAR TEMP1               VAR TEMP2
        STRING (D) 'D'          STRING (Y) 'Y'
DEND

PROGRAM PRINTF
        SYSFIELD ?
        CLS  WRITE '*********************************'
        CR   WRITE '**   File-Print Utility        **'
        CR   WRITE '*********************************'
        CR 14 OUTCHAR
        WRITE 'Drucker (D) oder CRT (C) ? : '
        TEMP1 READ(CH)
        WRITE 'Filename ? : '
        TEMP READ(CH)
        TEMP1 ? (D) =(S)
        IF
            WRITE 'Form Feed (Y/N) ?: '
            TEMP2 READ(CH)
            TEMP2 ? (Y) =(S)
            SCROFF DRON
            IF
              PAGE
            ENDIF
        ENDIF
```

```
        TEMP ? ?(B)
        0 > IF
                INCHAR
                TEMP ? FILE FILESPEC DROF
                FILE OPEN(1)
                REPEAT
                UNTIL
                  FILE READF
                  0= IF
                        127 AND OUTCHAR
                        INCHAR ZEICHEN ?(B)
                        0= IF
                              FALSE
                           ELSE
                              TRUE
                           ENDIF
                     ELSE
                        TRUE
                     ENDIF
                LOOP
                FILE CLOSE
            ENDIF
        DROFF SCRON
        SYSFIELD :=
END
```

Das Programm LINE zeichnet eine Linie in der HRG

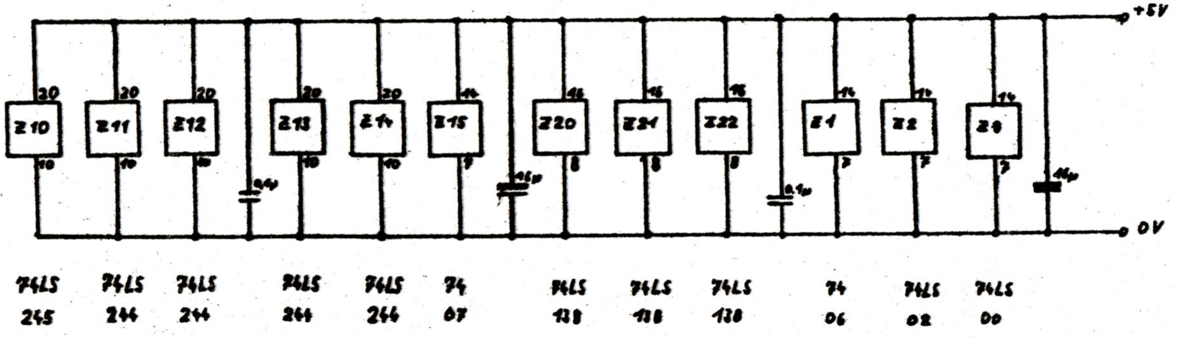
```
CODE +2DY 21 DY 23 23 5E 23 56 EB 29 EB E1 19 E5         CEND
CODE +2DX 21 DX 23 23 5E 23 56 EB 29 EB E1 19 E5         CEND
CODE -2DY 21 DY 23 23 5E 23 56 EB 29 EB E1 A7 ED 52 E5 CEND
CODE -2DX 21 DX 23 23 5E 23 56 EB 29 EB E1 A7 ED 52 E5 CEND

PROGRAM LINE     ;(X1 Y1 X2 Y2 FLAG ->    )
   ;Rette                     einige                      Parameter
MODE :=
   Y2 :=     X2 :=
   Y1 :=     X1 :=
   ;Errechne DX und bestimme aus dessen Vorzeichen, ob
   ;INCX = +1 oder -1 sein muß, anschließend wird der
   ;Absolutwert von DX gebildet
   X2 ? X1 ? - 32768 + DUP
   32768 >=
   IF
      32768 - DX :=
      1 INCX :=
   ELSE
      32768 SWAP - DX :=
      TRUE INCX :=
   ENDIF
   ;Hier wird das Gleiche mit DY und INCY durchgeführt
   Y2 ? Y1 ? - 32768 + DUP
   32768 >=
   IF
      32768 - DY :=
      1 INCY :=
   ELSE
      32768 SWAP - DY :=
```



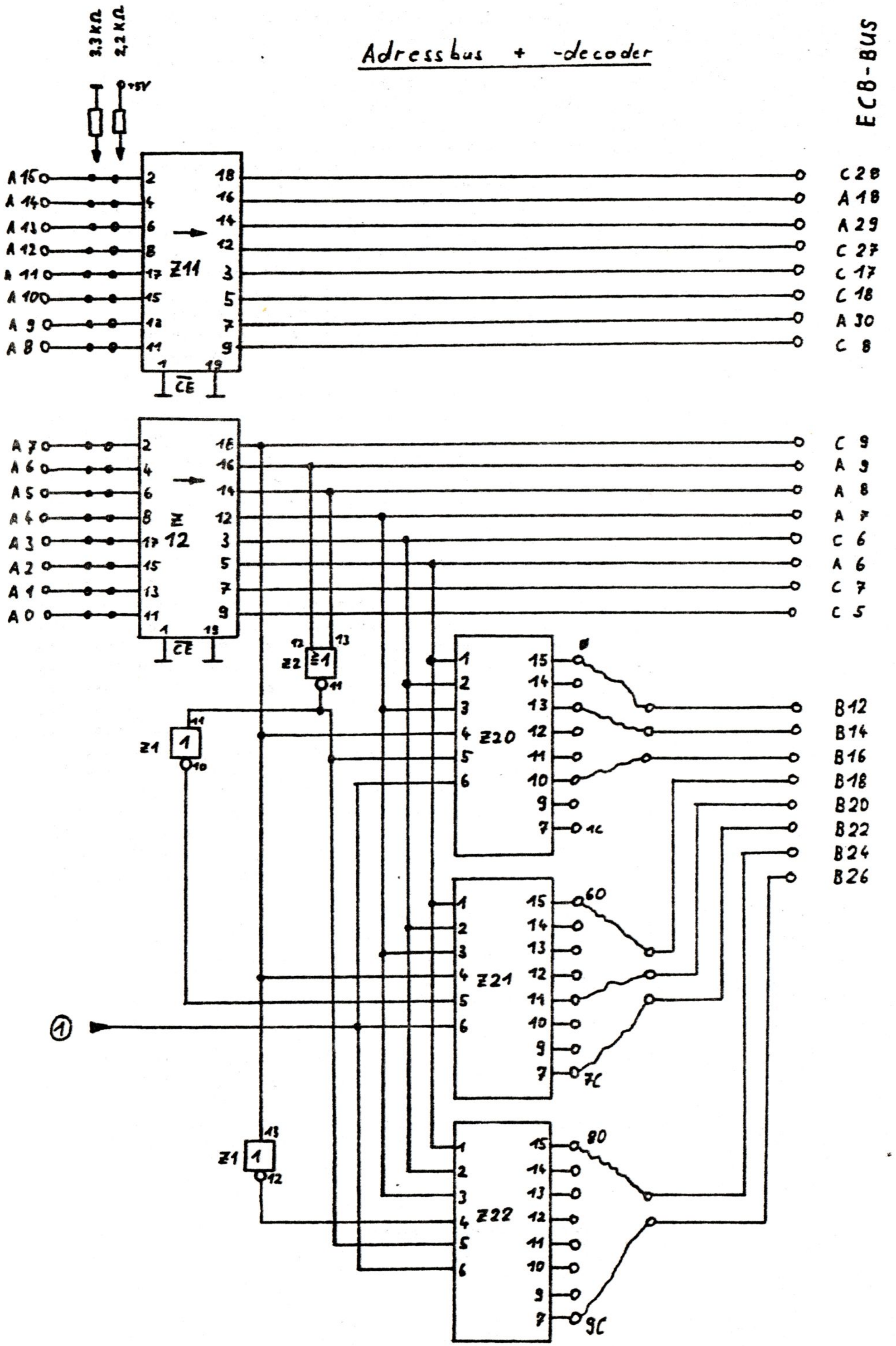
```
      TRUE INCY :=
   ENDIF
   DX ? DY ? >    ;Bestimme die Steigung m
   IF
      ;Schleife für Steigung < 45 Grad
      DY ? DUP + DX ? - 32768 + EE :=
      DX ? 0
      FOR
         X1 ? Y1 ? MODE ? DOT
         EE ? 32768 >=
         IF
            Y1 ? INCY ? + Y1 :=
            EE ? +2DY -2DX EE :=
         ELSE
            EE ? +2DY EE :=
         ENDIF
         X1 ? INCX ? + X1 :=
      LOOP
   ELSE
      ;Schleife für Steigung >= 45 Grad
      DX ? DUP + DY ? - 32768 + EE :=
      DY ? 0
      FOR
         X1 ? Y1 ? MODE ? DOT
         EE ? 32768 >=
         IF
            X1 ? INCX ? + X1 :=
            EE ? +2DX -2DY EE :=
         ELSE
            EE ? +2DX EE :=
         ENDIF
         Y1 ? INCY ? + Y1 :=
      LOOP
   ENDIF
END
```

Nun verstehen wir uns! Endlich habe ich den Durchbruch bei dem verdammten Ding geschafft.

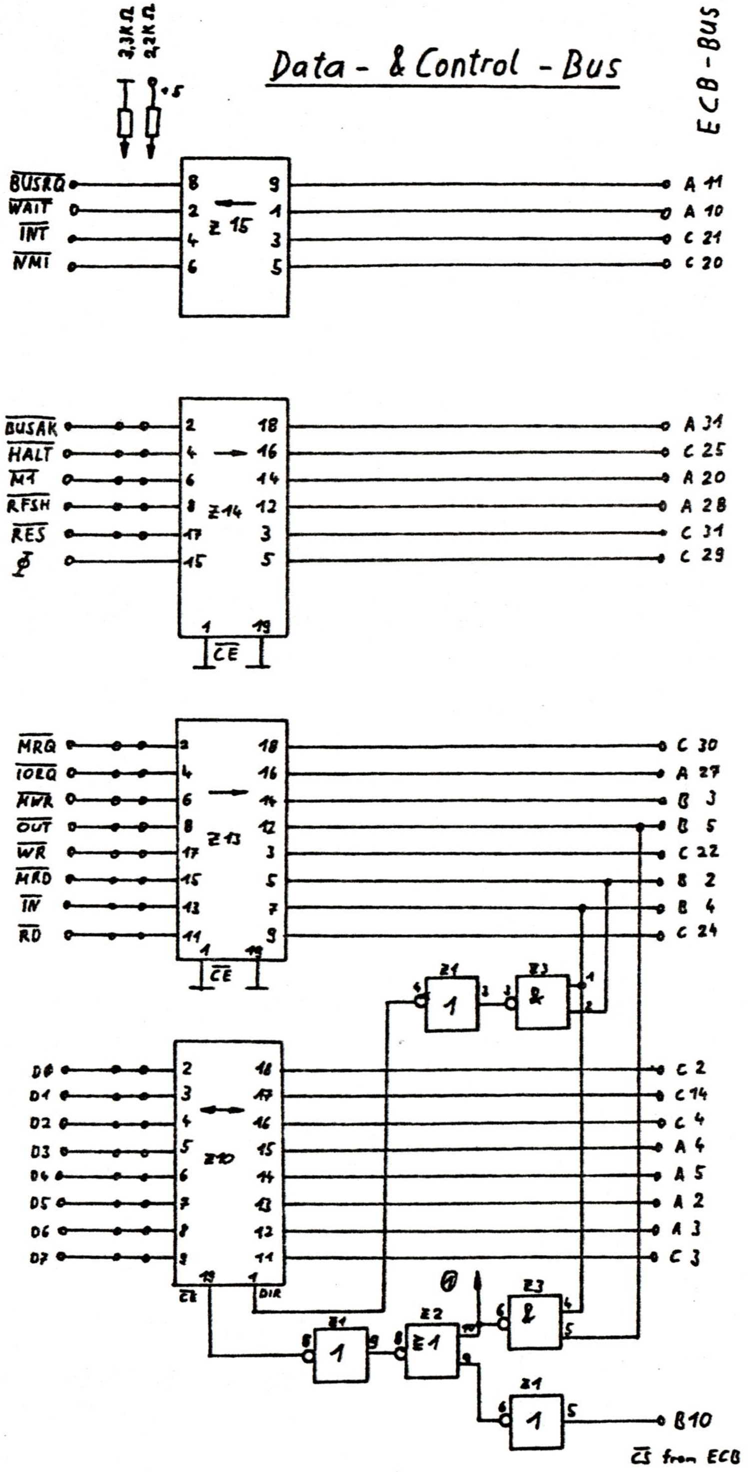
ECB-Adapterplatine
Ein erster Entwurf für eine ECB-Adapterplatine ist nun fertig.
Da die in Heft 9 vorgestellte Platine nicht funktionsfähig ist,muss die Adapterplatine vorläufig noch in Fädeltechnik aufgebaut werden.
Schaltungsbeschreibung:
Alle Signale gelangen über ein Flachbandkabel vom TRS/VG/Komtek zu einem 2 X 25 - poligen Pfostenstecker auf der Interfacekarte.Dieser Pfostenstecker wird auf der Lötseite je nach Gerätetyp individuell mit den Eingängen (terminier-Widerständen) verdrahtet. Dies ist notwendig wegen
der unterschiedlichen Beschaltung der Geräte und damit,das Verbindungskabel ein einfaches 1 zu 1 Kabel sein kann. (Anpressstecker)
Im Schaltplan sind an den meisten Eingängen 2 Kreise eingetragen.Diese deuten darauf hin,dass die Leitung, so wie ganz links oben gezeichnet, zu terminieren ist. Die 4 zum Computer führenden Leitungen, sowie der Takt sind ohne Terminierung .
Der Control-Bus und der Adressbus sind mit 74LS244 gepuffert, die fix das CS-Signal auf 0 haben. (immer aktiv)
Der Datenbus-Puffer ist ein 74LS245, der seine Richtungssteuerung von MRD oder IN erhält. Sein CS erfolgt entweder durch IN/OUT oder ein externes CS vom ECB-Bus auf B10.
(Wenn eine Karte IM Bus-System selbst seine Adresse decodiert,so muss das dabei entstandene CS-Signal auch auf Pin B10 gelegt werden , (open collector) damit der Datenbus-Puffer angesteuert wird.Das ist nur bei Memory-mapped mode notwendig)
Wenn man also für eine bestimmte Karte den Interrupt mode 2 verwenden will, so muss man diese Art der Ansteuerung wählen, da ja der Interrupt-Vektor auf den Bus gelegt werden muss.
Die 4 zum Computer führenden Control-lines sind mit einem 7407 gepuffert, da diese Leitungen voneinander unabhängig sein müssen.
Adressdecodierung:
Ziel war es soviele I/O Portadressen als möglich zu decodieren. Aus Platzgründen wurden nicht mehr als 3 Decoderbausteine 74LS138 samt den notwendigen Verknüpfungsbausteinen untergebracht.
Z20 decodiert in Viererschritten von Port 0 an bis 1C. z.B.:0,.4..8..C..10..14..18..1C. Bei den dazwischenliegenden Adressen erfolgt natürlich ebenfalls ein CS-Signal.z.B.:liefert der Pin 15 von Z20 ein CS-Signal für jede I/O Adresse von 00H bis 03H.
Das IC Z21 liefert eine Dreifachadressierung. Der Baustein bringt also ein CS-Signal auf Pin 15 für eine Adresse von 20H oder 40H oder 60H. Das hat den Vorteil, dass die Chance,irgendeine fertige Karte mit passender Adresse zu bekommen sehr hoch ist.
Z22 wiederum ist nach demselben Schema wie Z20 eingesetzt. Nur ist diesmal das MSB auf on, das heisst: die Adressen von 80H bis 9CH werden decodiert.
Alle Ausgänge der Decoderbausteine liegen auf Lötstiften. In ca. 1cm Abstand stehen die 8 Lötstifte,die zu den CS-Leitungen im ECB-System führen. So kann man mit kurzen Lötbrücken die gewünschten Adressen bestimmen und braucht auf einer ECB-Karte keine Adressdecodierung mehr vornehmen.



Adressbus + -decoder



Data- & Control-Bus



Verschiedene Anschlussleisten:

Leitung	TRS-80	ECB-Leiste	Komtek
A11	9	17C	6
A12	5	27C	11
A13	6	29A	14
A14	10	18A	13
A15	7	26C	16
Φ	16	29C	45
D4	18	5A	42
D3	26	4A	39
D5	28	2A	37
D6	24	3A	44
+5V		1ABC	
D2	32	4C	40
D7	20	3C	43
D0	30	2C	46
D1	22	14C	41
$\overline{INT}$	21	21C	10
$\overline{NMI}$	(47)	20C	5
$\overline{HALT}$	(48)	25C	9
$\overline{MREQ}$	1	30C	47
$\overline{IORQ}$	(20)	27A	35
$\overline{RD}$	(21)	24C	38
$\overline{WR}$	(22)	22C	36
$\overline{BUSAK}$	(23)	31A	12
$\overline{WAIT}$	33	10A	8
$\overline{BUSRQ}$	23	11A	7
$\overline{RESET}$	2	31C	33
$\overline{M1}$	(27)	20A	4
$\overline{RFSH}$	3	28A	3
GND	8 29 37	32ABC	
A0	25	7C	34
A1	27	7C	32
A2	40	6A	30
A3	34	6C	28
A4	31	7A	26
A5	35	8A	24
A6	38	9A	22
A7	36	9C	20
A8	11	8C	19
A9	17	30A	21
A10	4	18C	27
$\overline{MRD}$	13	2B	25
$\overline{MWR}$	15	3B	48
$\overline{IN}$	19	4B	
$\overline{OUT}$	12	5B	

Die ECB Norm verwendet nur die Stiftreihe A+C.
Die B-Reihe dient am Interface Für $\overline{CS}$-Adressdecodierung
26C und 31C sind auf der Platine verbunden
IEI IEO = Daisy-Chain

Die eingekreisten Zahlen beim TRS-80 Interface sind keine Bus-Pins sondern diese Leitungen müssen direkt von der CPU aus verlötet werden.

ECB-Bus Belegung

	A	B	C
1	+5V	+5V	+5V
2	D5	$\overline{MRD}$	D0
3	D6	$\overline{MWR}$	D7
4	D3	$\overline{IN}$	D2
5	D4	$\overline{OUT}$	A0
6	A2		A3
7	A4		A1
8	A5		A8
9	A6		A7
10	$\overline{WAIT}$	$\overline{CS}$ für ECB	
11	$\overline{BUSRQ}$		IEI
12	A18	$\overline{CS1}$	A19
13	+12V		-12V
14		$\overline{CS2}$	D1
15	-5V		-15V
16	½ Φ	$\overline{CS3}$	IEO
17	A17		A11
18	A14	$\overline{CS4}$	A10
19	+15V		A16
20	$\overline{M1}$	$\overline{CS5}$	$\overline{NMI}$
21			$\overline{INT}$
22		$\overline{CS6}$	$\overline{WR}$
23			GDT
24		$\overline{CS7}$	$\overline{RD}$
25			$\overline{HALT}$
26		$\overline{CS8}$	PWRCTL
27	$\overline{IORQ}$		A12
28	$\overline{RFSH}$		A15
29	A13		Φ (Takt)
30	A9		$\overline{MRQ}$
31	$\overline{BUSAK}$		$\overline{RESET}$
32	GND	GND	GND

ECB-BUS-GRUNDEINHEITSAUFBAU

Mit diesem Artikel möchte ich Euch für den Aufbau der Grundeinheit einige Tips geben. Diejenigen, die sie bestellt haben, kommen ja nun demnächst in den "Genuß" sie auch zusammenbauen zu dürfen.

An erster Stelle steht der mechanische Zusammenbau unseres Gehäuses. Einen groben überblick dazu gibt ja schon die dem Gehäuse beiliegende Anleitung, aber ein paar Anregungen zur Erleichterung der Arbeit sind schon noch nötig.
Zu beachten ist die richtige Anordnung der sechs Schienen. Die vorderen beiden Schienen sind diejenigen, welche den Deckelfalz und die Löcher für die Platinen-Führungsschienen haben. Diese werden im ersten Seitenteilbefestigungspunkt, zusammen mit der Winkelschine -an der vorher der Griff mit den selbstschneidenden Schrauben befestigt wurde-, verschraubt. Im dritten Befestigungspunkt des Seitenteiles werden dann die gelochten Schienen ohne Deckelfalz angebracht. Das zweite Loch von vorne bleibt frei. Die Lochbleche für den Einbau der Busplatine werden an der Schiene (auf Lochplatz drei des Seitenteiles) so befestigt, daß bis zur Vorderkante des Gehäuses der Abstand von 17,5 cm vorhanden ist. Später ist dann die Busplatine mit ihren Buchsenleisten von der Gehäuse-vorderseite einzulegen und daran anzuschrauben. Zur Entlastung der Platine werden alle zehn Buchsenleisten angeschraubt. Vorerst werden aber die restlichen zwei Winkelschienen über Loch vier des Seitenteiles durch Verschrauben, der Schienen mit Deckelfalz aber ohne Lochung, befestigt. Günstig ist es, in jede der "END-Schienen" drei "Einschiebemuttern" einzubringen. Sie dienen dann dem Anschrauben der Rückwand. Gleichzeitig sollten auch das Boden- und Deckelblech eingeschoben werden. Alles Weitere für den Gehäuse-Zusammenbau dürfte dann klar sein.

Zum Einlöten der zehn 96-poligen Busstecker auf die Busplatine ist zu beachten, daß die Buchsenleisten mit ihren Bezeichnungen übereinstimmend mit den Busbezeichnungen angebracht werden. Das heist, wenn die Stromversorgungsanschlüsse der Busplatine nach rechts zeigen und die Lötseite (beschriftete Platinenseite) nach unten zeigt, ist der Punkt 1A auf der Busplatine und auch auf der Buchsenleiste oben links zu finden. Dies gilt für jeden der zehn Steckplätze. Anhand der mitgelieferten Stecker kann dann überprüft werden, ob die Leitungen richtig durchverbunden sind. Besondere Beachtung ist der mittleren Reihe zu schenken, denn sie ist von der Stromversorgungsseite aus gesehen nach dem vierten Busplatz unterbrochen. Auf dieser Reihe haben wir unsere speziellen Signale, die wir von Fall zu Fall hier durchschalten bzw. auch auftrennen können. Weiterhin ist bei der linken Seite der jeweiligen Buchsenleiste die Verkettung von 23/25 und auf der rechten von 11/16 zu beachten. Hier wird jeweils ein Platztausch der Leitungen durchgeführt. Ist die Platine nun auf Durchgang und Isolation geprüft kann sie, wie links beschrieben, eingebaut werden. Danach ist das Netzteil anzuschließen. Die genaue Platzangabe für den Einbau kann noch nicht erfolgen, da ich die entsprechenden Netzteile noch nicht in Besitz habe und deshalb einen günstigen Einbauplatz noch nicht empfehlen kann. Dies wird aber bis zum entgültigen Zusammenbau noch nachgeholt.

Auf jeden Fall sollte der Anschluß der 220V nach VDE erfolgen. Die Mindestforderungen hier sind:
Berührungssicherheit der Spannungsführenden Leiter (Phase und Rückleiter)
Angeschlossenem Schutzleiter am Gehäuse.

Empfehlenswert ist auch die Verwendung eines zweipoligen Netz-Schalters mit beleuchtbarer Wippe zum Ein-/Ausschalten. Als weitere "Gestaltungsmöglichkeiten" wären Netzfilter zur Störabblockung und Steckdosen für externe Geräte denkbar. Dem jeweiligen Ausbau sind auch hier keinerlei Beschränkungen auferlegt.
Nun aber zur anderen Netzteilseite, der Niederspannung. Die Anschlußdrähte werden entsprechend ihrer Bezeichnung (zur Kontrolle vorher die Spannungen einmal messen) auf den gleichnamigen Punkt der Busplatine eingelötet. Danach sollten die entsprechenden Spannungen auf jeder Busbuchsenleiste greifbar sein. Bitte auch das überprüfen.

Das waren nun einige Tips für den Grundeinheitsaufbau. Sollten sich doch noch Probleme ergeben, bin ich gerne bereit sie mit Euch zu lösen.
Ist die Grundeinheit dann betriebsfertig geht es mit den hier nachfolgend vorgestellten Karten weiter.
Viel Erfolg beim Aufbau
Jens Neueder

Die Eingangsplatine. Bernd Drowälder

Die Eingangsplatine hat für eine Anpassung der Erweiterungen an den TRS-80 zu sorgen. Eine galvanische Trennung ist nicht erforderlich. Aus diesem Grunde kann auf Impulsübertrager oder optoelektronische Koppelelemente verzichtet werden. Die auf der Mutterplatine zur Verfügung stehenden Signale müssen mehrere TTL-Eingänge treiben können.

Diese Forderungen werden durch den TTL-Baustein 74LS243 erfüllt. In diesem Baustein sind 8 Leistungstreiber,von denen je 2 antiparallel geschaltet sind,integriert.
Um die 16 Adressleitungen,die 8 Datenleitungen,die Steuerleitungen RD*,WR*,IN*,OUT*,MUX,CAS* und RAS* in beiden Richtungen treiben zu können sind 8 dieser Bausteine nötig. Die ICs werden so beschaltet,daß die Leitungen jeweils immer in einer Richtung getrieben werden (GBA und GAB* miteinander verbunden).

Die Steuerungen für die Richtungsumschaltung der Adress- und Steuerleitungen übernimmt das Signal TEST. Das TEST* Signal wird innerhalb des TRS-80 über einen Pull-Up-Widerstand auf 5 Volt gelegt,wodurch die Treiber vom TRS-80 aus zur Mutterplatine durchgeschaltet werden und die CPU Z-80 im aktiven Zustand bleibt. Wird das TEST* Signal auf 0 Volt gelegt,werden die Adress- und Steuerleitungstreiber in Richtung TRS-80 geschaltet. Die CPU Z-80 schaltet ihre Adress-, Daten- und Steuerleitungen auf hochohmig. Dieses ist z.B. bei einem DMA (Direct Memory Acess) erforderlich.

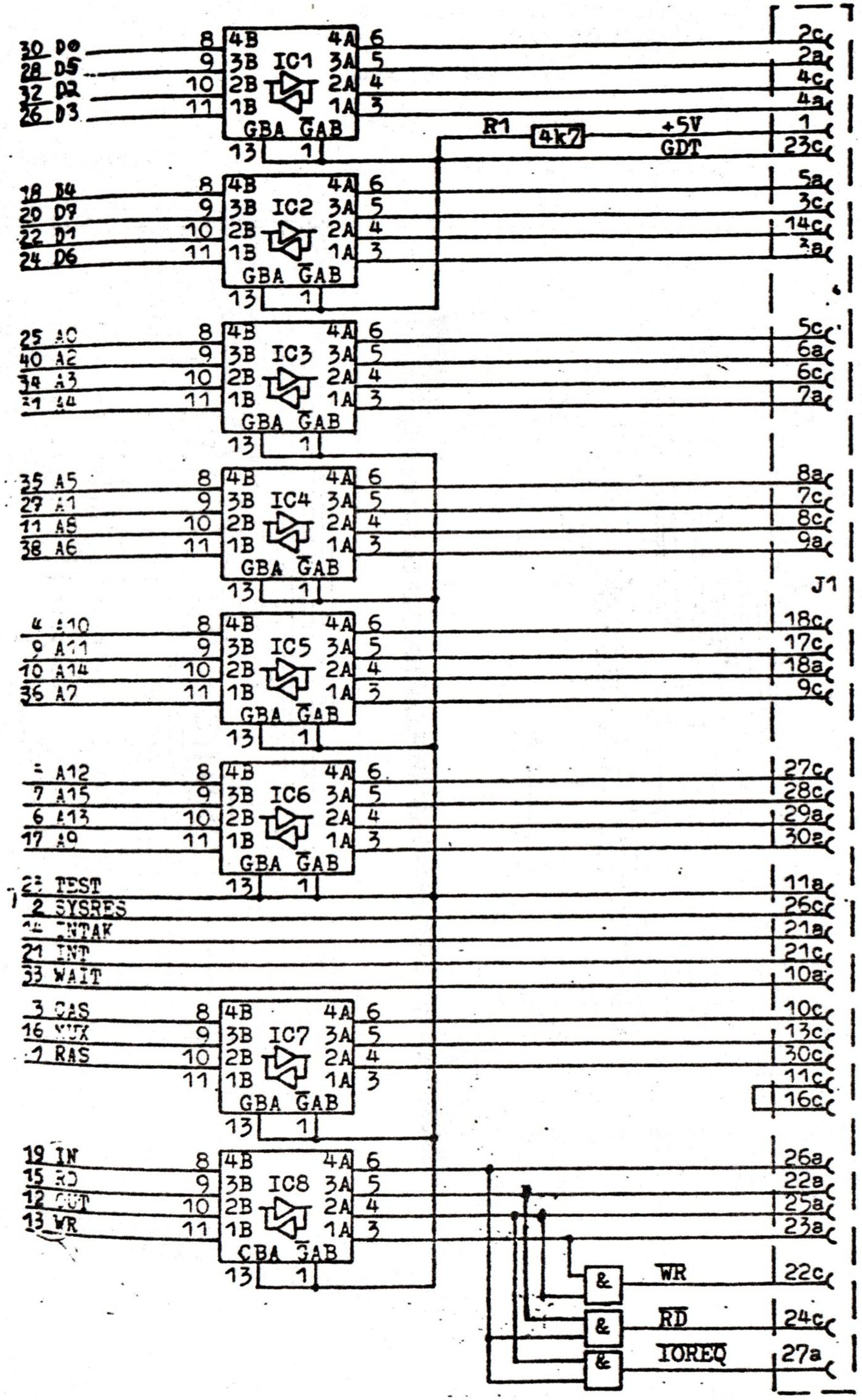
Bei der Datenrichtungsumschaltung wäre bei Berücksichtigung aller Möglichkeiten eine aufwendige Decodierung der Steuer- und Adressleitungen erforderlich. Außerdem wäre bei einem DMA zu beachten,daß die Datenrichtungsumschaltung genau invers zum normalen Betrieb geschieht.

Eine Lösung,die dieses Problem ohne großen Aufwand bewältigt,ist die Einführung der GDT (Gate Data Transfer). Diese Leitung,die jeder Erweiterung über die Mutterplatine zugänglich ist,wird mit den Anschlüssen GBA und GAB der Datentreiber IC1 und IC2 verbunden und über einen Pull-Up-Widerstand auf H-Signal gelegt. Die Daten werden so vom TRS-80 aus in Richtung Mutterplatine getrieben.
Soll die Richtung umgeschaltet werden,so muß die Leitung GDT auf L-Signal gelegt werden. Dies geschieht auf den Erweiterungsplatinen durch die entsprechenden Decodierungen,die mit einem Open-Kollector-Baustein oder mit einer einfachen Diode (siehe ZK01 diese Info-Ausgabe) abgeschlossen werden. Durch das Aufschalten dieser Open-Kolector und Tri-State-Ausgänge auf die Leitung GDT erhalten wir ein Wired AND, d.h.,daß sobald einer dieser Ausgänge auf L-Signal liegt,die Leitung GDT ebenfalls L-Signal führt. Diese Maßnahmen müssen bei der Auslegung von Erweiterungen berücksichtigt werden,da sonst Fehlfunktionen unvermeidbar sind! Käufliche Karten,können zu 90% mit einer einfachen Diode umgestellt werden.

Die Leitungen SYSRES*,INTAK*,INT* und Wait werden ohne Treiber auf die Eingangsplatine durchgeschleift,da sie nicht oft Verwendung finden und zum Betreiben von 2 TTL-Lasten ausreichend Leistung liefern.

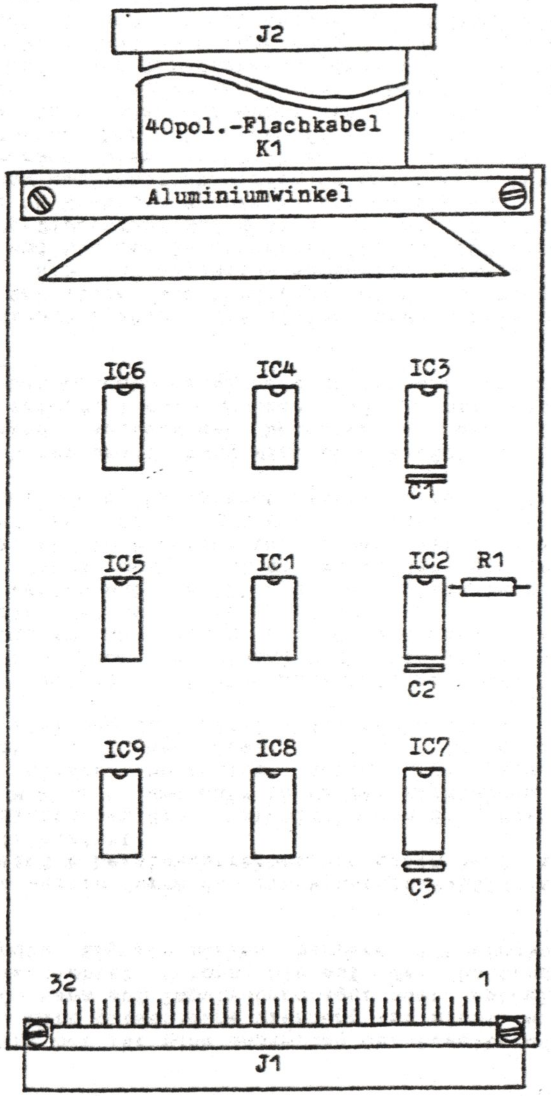
Die Signale WR*,RD* und IORQ* der CPU Z-80 stehen am Ausgang des TRS-80 nicht direkt,sondern nur als deren Oder-Verknüpfungen IN*,OUT*,RD* und WR* zur Verfügung.Da eine Kompatibilität mit dem ELZET-80-BUS erwünscht ist und dort mit den Originalsignalen gearbeitet wird,werden sie über Und-Verknüpfungen zurückgewonnen.
Das Signal RAS* ist identisch mit dem CPU-Signal MREQ*.

Schaltplan Eingangskarte



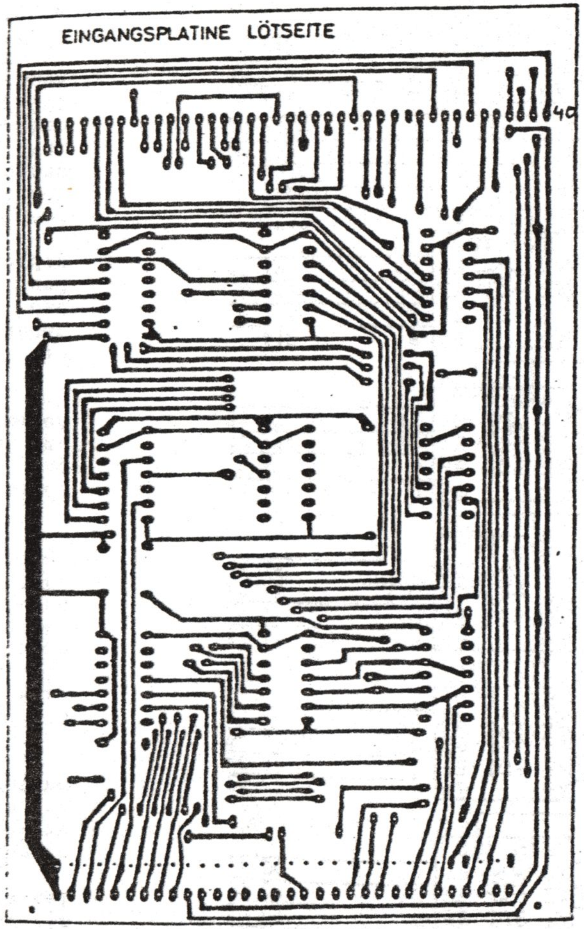
ECB-TRS-80: EINGANGSPLATINE

Bestückungsplan Eingangskarte



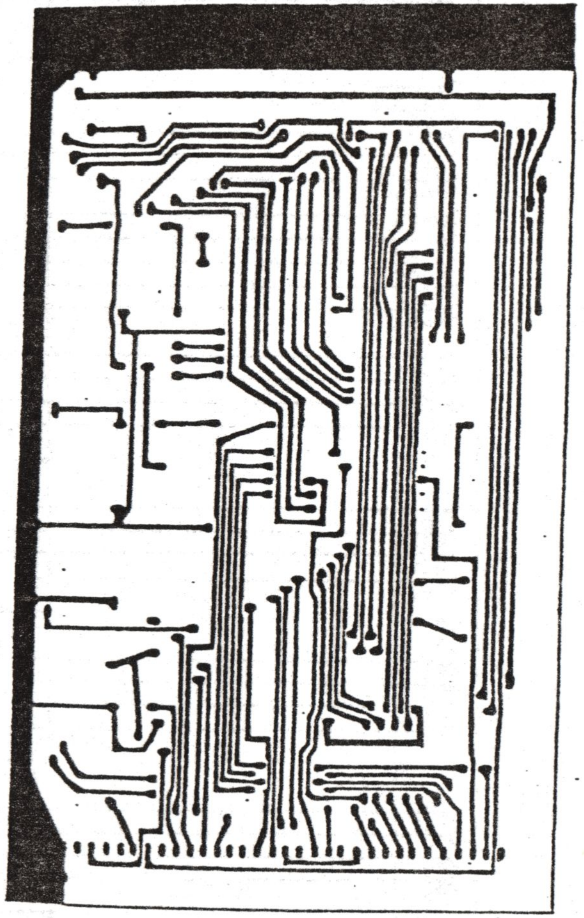
Platinenlayout Eingangskarte

Lötseite



Platinenlayout Eingangskarte

Bestückungsseite



Stückliste Eingangskarte

Halbleiter

IC 1-8	74LS243
IC 9	74LS08

Widerstände

R1	4,7k

Kondensatoren

C1-C3	100nF

Sonstige Bauteile

J1	64 pol. Steckerleiste
J2	40 pol. Steckverbinder (TRS-80)
K1	40 cm Flachbandkabel 40-adrig

ECB-Bus - Die erste Zusatzplatine !!!! (ZK01/02) Bernd Drowälder

Der folgende Artikel,beschreibt eine universelle 8-Bit-Parallelschnittstelle,für den ECB-Bus.
Die Karte bietet folgende Möglichkeiten.

1. 8-Bit Parallelausgang. (Wert wird gespeichert!)
2. 8-Bit Paralleleingang. (zusätzlich benutzbar!)
3. Einstellbar auf jede der 256 möglichen I/O Adressen!

Wer will,kann 256 Karten gleichzeitig betreiben.
Durch die freie Wahl der Adresse,dürfte wohl jeder,noch ein "freies Plätzchen",in seinem I-O Bereich finden.
Die Auswahl erfolgt über 8-Dilschalter im Binärsystem.

Tabelle zur Adresseinstellung.

```
S1=2^0=1    Stelle.
S2=2^1=2    Stelle.
S3=2^2=4    Stelle.
S4=2^3=8    Stelle.
S5=2^4=16   Stelle.
S6=2^5=32   Stelle.
S7=2^6=64   Stelle.
S8=2^7=128  Stelle.
```

Es folgen einige Beispiele.

gewünschte Adresse: Dezimal:	Hex:	Dilschalter: S1	S2	S3	S4	S5	S6	S7	S8
255	FF	Ein	Ein	Ein	Ein	Ein	Ein	Ein	Ein
254	FE	Aus	Ein	Ein	Ein	Ein	Ein	Ein	Ein
253	FD	Ein	Aus	Ein	Ein	Ein	Ein	Ein	Ein
252	FC	Aus	Aus	Ein	Ein	Ein	Ein	Ein	Ein
...	..	...	...	...	...	...	...	...	...
003	03	Ein	Ein	Aus	Aus	Aus	Aus	Aus	Aus
002	02	Aus	Ein	Aus	Aus	Aus	Aus	Aus	Aus
001	01	Ein	Aus	Aus	Aus	Aus	Aus	Aus	Aus
000	00	Aus	Aus	Aus	Aus	Aus	Aus	Aus	Aus

Mit Hilfe der Tabelle,müßte eigentlich jeder seine gewünschte Adresse einstellen können.Es folgen noch ein paar Tips zum Testen der Karte.Dazu ist es jedoch erforderlich,acht Leuchtdioden,gemäß Bild 1.1 an die Karte anzuschließen.

Testprogramm A in Basic. (Ausgeben)

```
10REM EINGESTELLTE ADRESSE 0
20REM ALLE DILSCHALTER AUS !
30INPUT "WELCHER WERT SOLL AUSGEGEBEN WERDEN";W
40OUT0,W
50goto30
```

Testprogramm B in Basic. (Ausgeben)

```
10REM EINGESTELLTE ADRESSE 0
20REM ALLE DILSCHALTER AUS !
30REM LAUFLICHT
40FOR I=0 TO 7
50OUT0,2^I
60GOSUB 100:REM GEWÜNSCHTE VERZÖGERUNG
70NEXT I
80GOTO 40
90REM VERZÖGERUNG
100FOR V=1 TO 200
110NEXT V
120RETURN
```

Testprogramm C in Basic. (Einlesen)

```
10Rem EINGESTELLTE ADRESSE 0
20REM ALLE DILSCHALTER AUS !
30REM DIE EINZELNEN DATENEINGÄNGE MÜSSEN AUF MASSE GELEGT WERDEN
40CLS
50W=INP(0):PRINT W:GOTO 50
```

Wird kein Eingang auf Masse gelegt,muß der angezeigte Wert 255 betragen !
Wird D0 auf Masse gelegt,dann 254. (D1=253,D2=251,D3=247 u.s.w.)

Weitere nützliche Beispiele folgen in den nächsten Infos.

HEFT 15 September 1986

Schaltplan (ZK01)

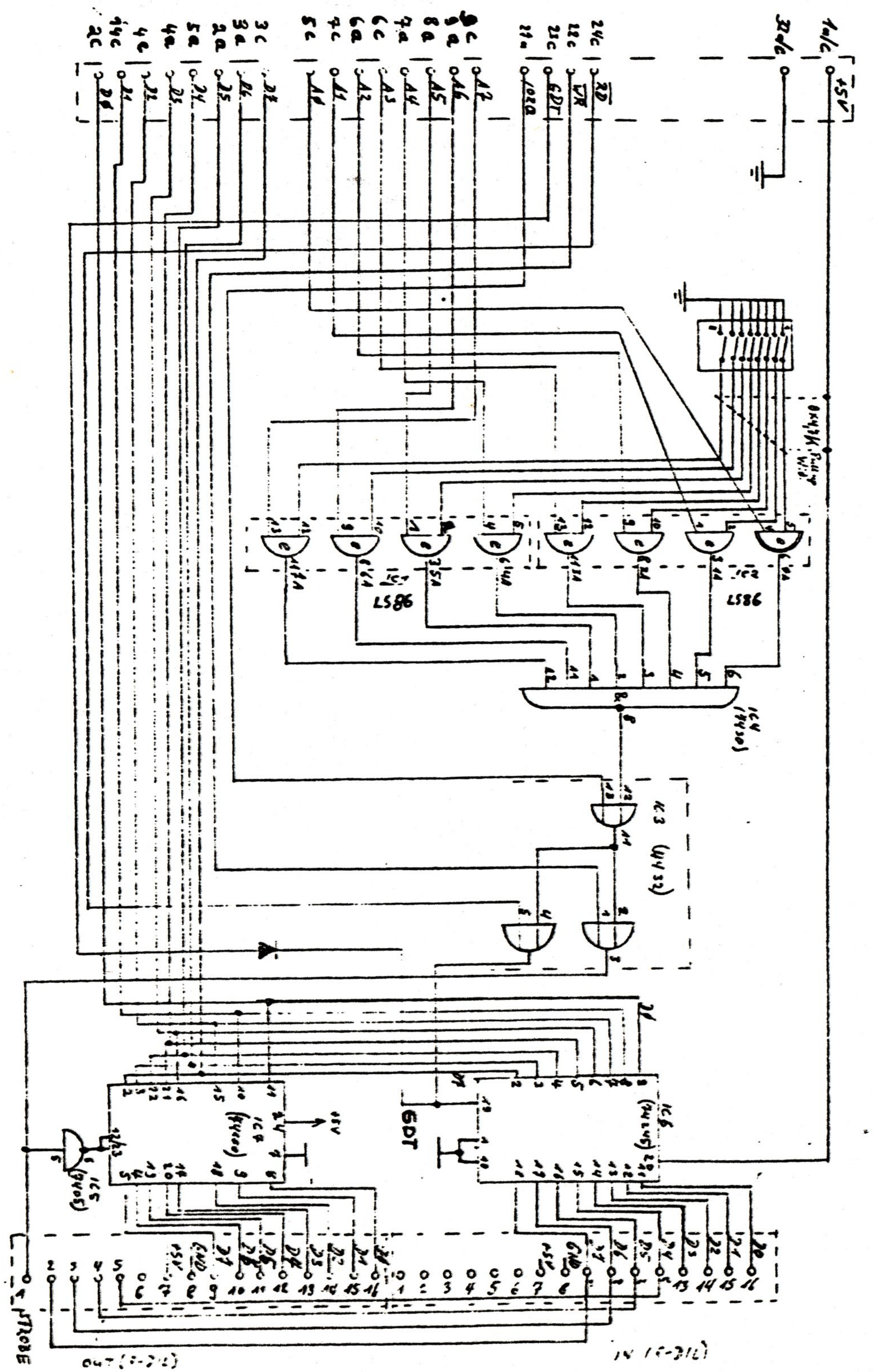
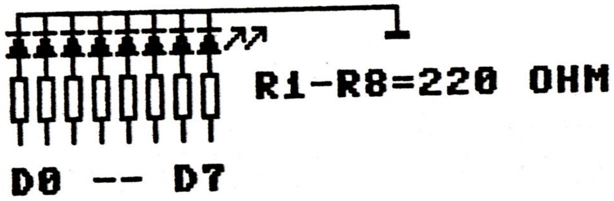


BILD 1.1



<u>Stückliste ZK01</u>

Halbleiter

IC 1,2	74LS86
IC 3	74LS32
IC 4	74LS30
IC 5	74LS04
IC 6	74LS245
IC 7	74LS100

Widerstände

R1-R8	4,7k

Kondensatoren

C1-C2	100nF

Sonstige Bauteile

J1	64 pol. Steckerleiste
D1	1N4148
Dilschalter	8*ON-OFF

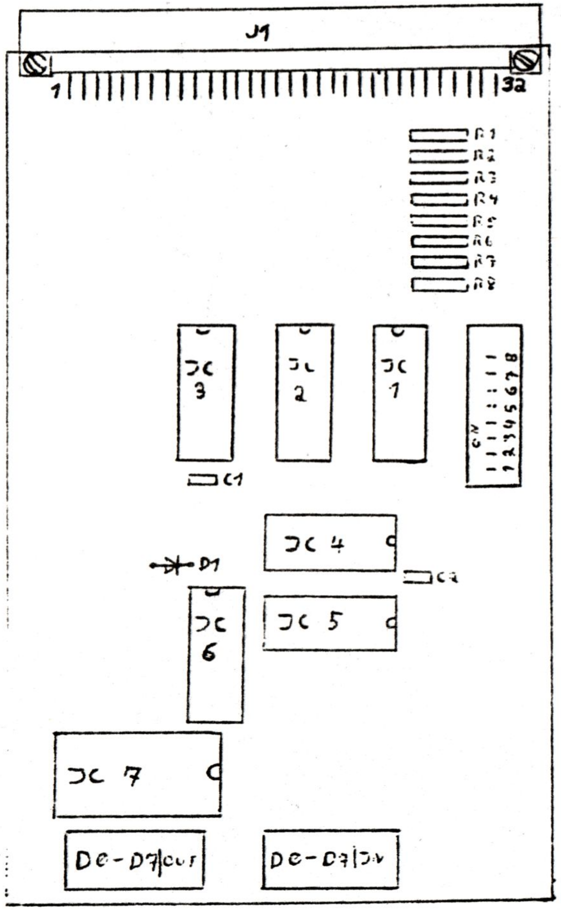
P.S. Am besten alle Ic,s auf Fassungen !!!

Um den Hartmut bei seinem <u>Hardware-Grundlagen-Kurs</u> zu unterstützen, folgt in der nächsten Info eine genaue Beschreibung der <u>Hardware</u>.

Viel Spaß beim Aufbau !!! Bernd Drowälder

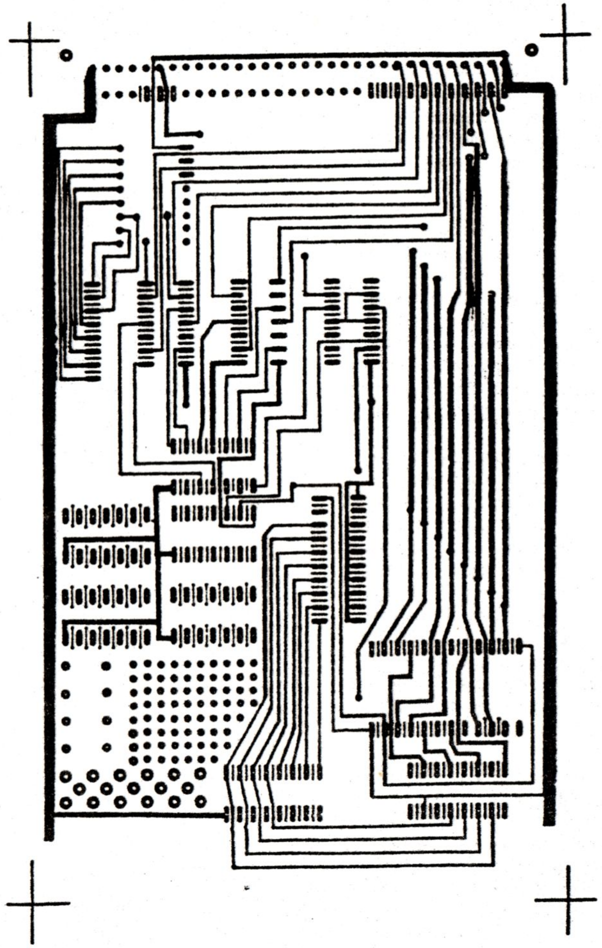
ECB-TRS-80: ZUSATZPLATINE ZK01/02

Bestückungsplan (ZK 01)



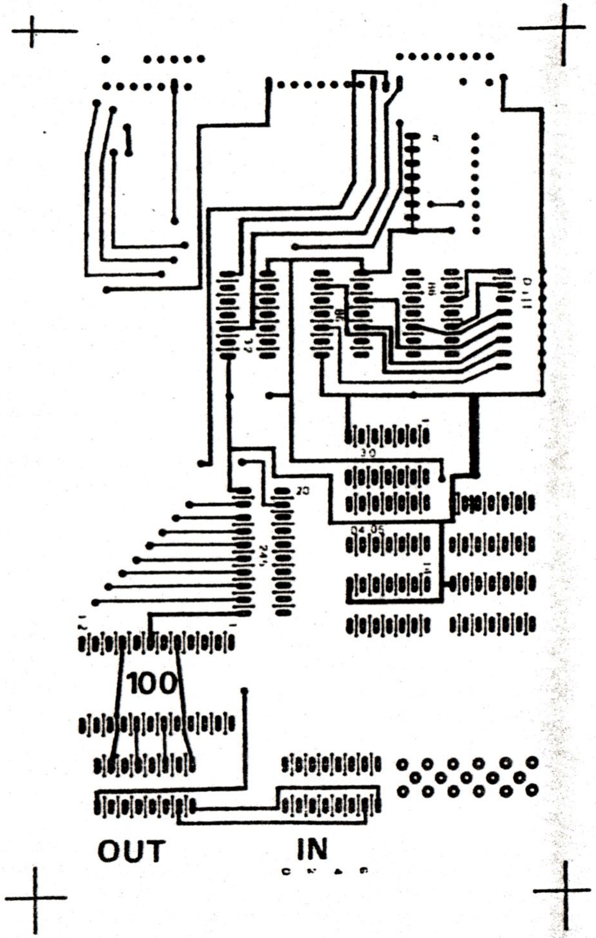
Platinenlayout (ZK01)

Lötseite



Platinenlayout (ZK01)

Bestückungsseite



ECB-TRS-80 Platinen

... für Selbermacher !!!
Eine Platinenvorlage im Maßstab 1:1
ist am INFO-Ende beigefügt.

... für die, die Vorgefertigtes mögen !!!
Die fertig geätzte Platine (aber ungebohrt) ist
von Bernd Drohwälder erhältlich. Der Preis liegt
für jede Platine so um die DM 15,--.
Bestellungen nur gegen Vorauskasse möglich!
Bestellschluß ist immer 14 Tage nach INFO-Erscheinen.

Ich glaube, er braucht die alte Schocktherapie!

ECB-Bus für Model 3/4!???

Erheblich älter als die Idee einem Model 1-kompatiblen ECB-Bus-Nachbau Leben einzuhauchen, ist der Plan den Tandy-Maschinen mittels einer Erweiterungsplatine die ECB-Welt zu eröffnen. Erstmalig propagiert wurde sie von unserem, inzwischen auf IBM umgestiegenen, Mitglied Walter Zwickel. Leider schleppte sich die Entwicklung und Fertigung der Platine so lange hin, bis fast alles Interesse zum Erliegen gekommen war. Jetzt aber, nach der Veröffentlichung des neuen (und hoffentlich erfolgreicheren) ECB-Projekts, wachsen in manchem Optimisten doch wieder die Hoffnung auf eine RAM/CMOS-Floppy oder eine Harddisk, auf eine Uhr oder einen EPROM-Brenner. Diese und ähnliche Hoffnungen mögen für Besitzer eines Model 1 oder des kompatiblen Nachbaus durchaus ihre Berechtigung haben, für Model 3/4-Eigner sind sie nicht uneingeschränkt angebracht!

Natürlich kann man auch dem Model 3/4 die volle Palette der ECB-Bus-Erweiterungen zukommen lassen, dies jedoch nur mit einem, meiner Meinung nach nicht vertretbar hohen Aufwand an Modifikationen im Gerät selbst. Grund dafür ist der leider etwas mickrig, aber für normale Zwecke durchaus ausreichend ausgefallene I/O-Bus der großen Brüder des Model 1. Dieser macht nämlich seinem Namen "I/O-Bus" alle Ehre und enthält nur Signale, die dafür auch von Nöten sind! Er enthält, neben 25 Masseleitungen und einem nicht benutzten Pin, praktisch nur Leitungen, die für die portorientierte Ein/Ausgabe benötigt werden. Für den vollständigen ECB-Anschluß fehlen also vor allem die "oberen" acht Adressleitungen!

Um den ECB-Willigen unter den Model 3/4-Besitzern ihre Hoffnung auf Erweiterung der Anwendungsmöglichkeit ihres Computers nicht ganz zu nehmen, möchte ich hier kurz aufzeigen, was ohne übertrieben großen Aufwand machbar ist. Dazu ist zu bemerken, daß bis jetzt (20.08.86) der ECB-Anschluß auch bei meinem 4p noch nicht realisiert ist und ich also rein theoretische Informationen, die jedoch nach bestem Wissen und mit Sorgfalt zusammengetragen sind, weitergebe. Ich hoffe schon im nächsten Info mehr zu diesem Thema bringen zu können!

Nun aber endlich ans Eingemachte! Die ECB-Busbelegung, wie sie beim Selbstbauprojekt benutzt werden soll, ist ja schon im letzten Info veröffentlicht worden. Hier eine Aufzählung der Anschlüsse, die man mit einem Model 3/4-I/O-Bus bedienen kann.

ECB-Bus Pin-Nr.	Signal	4p-I/O-Bus Pin-Nr.	Besonderheiten
a 1	+ 5V	---	Spannung für ext. Erweiterungen
a 2	D 5	11	Datenleitung 5
a 3	D 6	13	" " 6
a 4	D 3	7	" " 3
a 5	D 4	9	" " 4
a 6	A 2	21	Adressleitung 2
a 7	A 4	25	" " 4
a 8	A 5	27	" " 5
a 9	A 6	20	" " 6
a 10	/WAIT	41	beim 4p /IOBUSWAIT genannt
a 11	/BUSRQ	---	nicht im 4p-I/O-Bus vorhanden!
a 12	A 18	---	" " "!
a 13	+12V	---	Spannung für ext. Erweiterungen

ECB-Bus Pin-Nr.	Signal	4p-I/O-Bus Pin-Nr.	Besonderheiten
a 14	---	---	nicht belegt!
a 15	- 5V	---	Spannung für ext. Erweiterungen
a 16	2 Phi	---	nicht im 4p-I/O-Bus vorhanden!
a 17	A 17	---	" " "!
a 18	A 14	---	" " "!
a 19	+15V	---	Spannung für ext. Erweiterungen
a 20	/M1	47	
a 21	---	---	nicht belegt!
a 22	---	---	" "!
a 23	BAI	---	nicht im 4p-I/O-Bus vorhanden!
a 24	VCMOS	---	Spannung für Ext. Erweiterungen
a 25	BAO	---	nicht im 4p-I/O-Bus vorhanden!
a 26	---	---	nicht belegt!
a 27	/IORQ	49	beim 4p /XIOREQ genannt
a 28	/RFSH	---	nicht im 4p-I/O-Bus vorhanden!
a 29	A 13	---	" " "!
a 30	A 9	---	" " "!
a 31	/BUSAK	---	" " "!
a 32	GND	2-50	alle geraden Pins!
b 1	+ 5V	---	Spannung für ext. Erweiterungen
b 2	/MRD	---	nicht im 4p-I/O-Bus vorhanden!
b 3	/MWR	---	" " "!
b 4	/IN	33	beim 4p /XIN genannt!
b 5	/OUT	35	beim 4p /XOUT genannt!
b 6-9	CP/M	---	projektspez. verwendete Pins!
b10-31	---	---	nicht belegt!
b 32	GND	2-50	alle geraden Pins!
c 1	+ 5V	---	Spannung für ext. Erweiterungen
c 2	D 0	1	Datenleitung 0
c 3	D 7	15	" " 7
c 4	D 2	5	" " 2
c 5	A 0	17	Adressleitung 0
c 6	A 3	23	" " 3
c 7	A 1	19	" " 1
c 8	A 8	---	nicht im 4p-I/O-Bus vorhanden!
c 9	A 7	31	Adressleitung 7
c 10	---	---	nicht belegt!
c 11	IEI	---	Signal der Interruptkette!
c 12	A 19	---	nicht im 4p-I/O-Bus vorhanden!
c 13	-12V	---	Spannung für ext. Erweiterungen
c 14	D 1	3	Datenleitung 1
c 15	-15V	---	Spannung für ext. Erweiterungen
c 16	IEO	---	Signal der Interruptkette!
c 17	A 11	---	nicht im 4p-I/O-Bus vorhanden!
c 18	A 10	---	" " "!
c 19	A 16	---	" " "!
c 20	/NMI	---	4p-intern verwendet!
c 21	/INT	39	beim 4p /IOBUSINT genannt!
c 22	/WR	---	nicht im 4p-I/O-Bus vorhanden!
c 23	/GTD	---	nicht im 4p-I/O-Bus vorhanden!
c 24	/RD	---	nicht im 4p-I/O-Bus vorhanden!
c 25	/HALT	---	nur bei mehrfach-CPU-Systemen!
c 26	/PWRCL	37	Peripherie-RESET!
c 27	A 12	---	nicht im 4p-I/O-Bus vorhanden!
c 28	A 15	---	" " "!
c 29	Phi	---	" " "!
c 30	/MREQ	---	" " "!
c 31	/RESET	---	CPU-Reset! Nicht an /XRESET des 4p-I/O-Bus anschließen!!!
c 32	GND	2-50	alle geraden Pins!

Wie man sieht, bleiben die meisten Pins des ECB-Bus leer. Vor allem die für den Anschluß von RAM-Erweiterungen so wichtigen Adressleitungen A8-A18 und die Signale für den DMA (Direct Memory Excess) fehlen. Für die Ansteuerung von portorientierten Erweiterungskarten ist dagegen (fast) alles Nötige vorhanden.

Noch ein paar Worte zu einigen Signalen:

1. Die Spannungen auf dem ECB-Bus (+/- 5, 12 und 15 Volt) dienen der Stromversorgung der Erweiterungsplatinen und werden sowieso von einem externen Netzteil aufgebracht!

2. Die Taktfrequenz der CPU (Phi) sollte man nachträglich auf den nicht benutzten Pin 45 des 4p-I/O-Bus legen (also doch im Gerät löten, aber nur ein kleines bischen!), da sie von manchen Karten (vor allem solche, die mit der PIO arbeiten) benötigt wird! Das Doppel der Taktfrequenz (2 Phi) wird nur selten benötigt und kann, falls nötig, durch einen mit Phi syncronisierten Taktgenerator erzeugt werden.

3. Das ECB-Signal /RESET ist normalerweise ein Eingang für die CPU-Kartem um diese zurückzusetzen. Das 4p-I/O-Bus-signal /XRESET ist ein Ausgang und dient zum Zurücksetzen der Peripherie! Die beiden Signale sollte man nicht durcheinanderbringen bzw. koppeln!

4. Die zum Aufbau einer Interruptkette benötigten Signale IEO und IEI müssen nicht zur CPU und damit auch nicht zum 4p-I/O-Bus durchgeschleift werden.

5. Wie schon gesagt fehlen die Signale A8-A18, die Speicherzugriffssignale /WR, /WR, /MREQ, /MRD, /MRD und die DMA-Signale BAI und BAO, sowie weitere Leitungen, die aber nur für Speicherzugriffe notwendig sind!

6. Alle Signale des I/O-Bus sind intern gebuffert und können ohne weiteres direkt auf die oben angegebenen Leitungen des ECB-Bus gelegt werden. Zu beachten ist nur noch die Funktion der Leitung /EXTIOSEL. Diese Leitung muß, bevor die CPU auf externe Daten zugreifen kann, auf "low" gelegt werden. Das bewerkstelligt man am besten mittels einer NAND-Verknüpfung von /XIN und der angesprochenen Portadresse.

Was ist also machbar, was unmöglich? Unmöglich ist zunächst einmal nichts, vorausgesetzt man "besorgt" sich die fehlenden Signale direkt im Gerät, sprich man lötet wild darin herum! Aber auch wenn man das nicht will, bleibt der ECB-Bus interessant. Man kann dann aber "nur" portorientierte Erweiterungen wie Uhr, EPROMmer, Messwerterfassung, Ein/Ausgabe (Relaissteuerungen, Modeleisenbahnsteuerung) und sogar so umfangreiche Projekte wie den c't-Harddisk-Controller, realisieren! Ich hoffe aber, daß sich trotzdem ein paar Unentwegte finden werden, die sich für die ECB-Erweiterung ihres Model 3/4/4p interessieren und so die Gruppe der ECB-Nachbauer und -Begeisterten erweitern. In diesem Sinne viel Spaß und Glück beim Löten, euer

Hartmut Obermann

BÖRSE -- BÖRSE

Hilfe! Hilfe!

Ich bin ein Disketten-
verschwender!

Wer hat? Doubler
Wer weiß? gesucht
Wer weiß wo?
für GENIE I

Herbert Albers
Zum Düwelshöpen 14
2117 Wistedt
04182/8799

Hilfe! Hilfe!

Ich suche Leute, die mit FORTRAN programmieren!
Günther Wagner

SUCHE aus den Jahren 1985/86 versch. LOAD 80 - Disketten.
Klaus Hermann, Tel. 07127/70024

Verkaufe:
GRIP 2.1 / 2.6 / 3.1
780 X 280 Punkte Grafik-Auflösung,
96 Buchstaben je Zeile -- 33 Zeilen,
Hardware-Spooler, V24, Druckausgabe,
Tastatureingabe seriell oder parallel.
(Beschrieben in der c't)

VP: 350,-- DM
Manfred Held

Wer Interesse an einer
GENIE 3/3s - ECKE
in unserer Club-INFO hat,
sollte sich beim lieben Arnulf melden!!

Normung und Rechtschreibung

Nachdem in unserer letzten "CLUB-Info" wieder der Druckfehlerteufel grausam zugeschlagen hatte, möchte ich Euch heute eine Vereinfachung in der Rechtschreibung vorschlagen. Mit dieser Rechtschreibreform werden in Zukunft sicher keine Fehler mehr auftauchen.

Die neue Rechtschreibreform sollte "Schritt für srit" folgendermaßen eingeführt werden:

Erster Schritt:
Wegfall der Großschreibung!
einer sofortigen einführung steht nichts mehr im wege, zumal schon viele grafiker und werbeleute zur kleinschreibung übergegangen sind.

zweiter schrit:
wegfall der dehnungen und schartungen!
dise masname eliminirt schon di gröste felerursache in der grundschule, den sin oder unsin unserer konsonantenverdopelung hat onehin nimand kapirt.

driter schrit:
v und p werden ersetzt durch f,
sch, tz und z durch s!
das alfhabet wird um swei buchstaben redusirt, sreibmasinen und sesmasinen fereinfachen sich, wertfole arbeitskräfte könen der wirtsaft sugefürt werden.

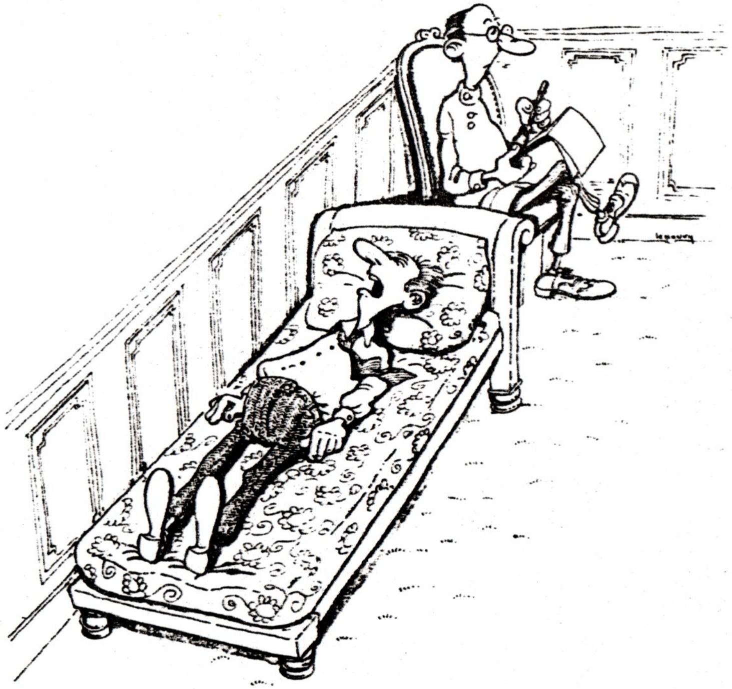
firter srit:
q und ch werden ersest durch k,
j und y werden ersest durk i,
fh (ph) wird ersest durk f!
iest sind son seks bukstaben drausen und di sfrake woltuend klar, di sulseit kan sofort fon neun auf swei iare ferkürst werden. anstat aksig frosent rektsreibunterikt könen nüslikere fäker, wi fisik, kemi und reknen mer geflegt werden.

fünfter srit:
wegfal fon ä-, ö-, ü-seiken!
ales uberflusige ist iest ausgemerst, di ortografi ist wider slikt und einfak, naturlik benotigt es einige Zeit, bis dise fereinfakung uberal riktig ferdaut wird, fileikt sasungsweise ein bis swei iare.

dem folgt ein sekster srit:
g ersest durk k, t ersest durk d,
und warum ersesd man nikd w durk f?
wen ale dise fordsride durkkessd sind, durfde als naksdes Zil di fereinfakunk der nok sfirken und unsinikeren deudsen kramadik anfisird ferden.

Di Redakdion

PS: Wer solche Texte nachmacht oder verfälscht oder nachgemachte oder verfälschte sich verschafft und in die Club-INFO bringt, wird mit Computerentzug nicht unter zwei Jahren bestraft.



Mein Computer versteht mich nicht!
Immer Syntax Error, Syntax Error!

* * * SPRACH-Entwicklung * * *

Assoziationen eines Assemblierers
"auf der Suche nach dem verlorenen Mnemonic"
(Text frei nach Marcel Proust; Ton nach Richard Strauß)

Ein Public Domain-Programm in JotCal, das auf allen Modellen lauffähig ist! Originaldiskette bei KalJot erhältlich.
Filename: "BABYLON"

```
1  ' Dunkler ASSembler                         ; A
2  'Finstere ASSoziationen
3  'Keller-ASSeln                              ; u
4  'KellergerASSel
5  'Untergrund-Geflüster                       ; f
6  'Grundtöne
7  'Grundsprache                               ; s
8  'Basis-Sprache
9  'BASIC-Sprache                              ; t
10 'BASF-Sprache
11 'BAYER-Sprache                              ; i
12 '(Bayrisch)
13 'Hochsprache                                ; e
14 'HOECHST-Sprache
15 'Diskant                                    ; g
16 'KANT's Sprache
17 'Kannit-verstan
18 'Kanni-balen-Sprache                        ; &
19 'Ver-bal-Sprache
20 'Schriftsprache
21 'Geheimsprache                              ; N
22 'Geheim-Code
23 'Hieroglyphen                               ; i
24 'Hiro Hito
25 'Heros-Hymnen                               ; e
26 'Heldengesänge
27 '4 Letzte Lieder  (Rich.Strauß)             ; d
28 'Schwanengesang
29 'Abgesang                                   ; e
30 'Absprache
31 'Abwärtssprache                             ; r
32 'Tiefsprache
33 'Kellersprache                              ; g
34 'Kellerasseln
35 'Rasseln                                    ; a
36 'Röcheln
37 'Röhren                                     ; n
38 'Retörnen
39 'RETURN                                     ; g
40 'RET  <<==--- heureka (ich hab's) !
```

END (lich)

JT

* Neues von der DISKO-Front ! *

Unsere DISKOTHEK ist gewaltig gewachsen!
Der Löwenanteil stammt diesmal aber nicht aus den eigenen Reihen.
Wie ist das zu erklären? Haben wir uns etwa doch entschlossen, allerlei Software zu kopieren, Copyrights zu umgehen, Protected Programs zu knacken???
Honny soit qui mal y pense!

Keineswegs!

Wir haben ein großes Paket sogenannte "Public Domain Software" an Land gezogen: gut 250 Programme aus USA, verteilt auf 20 Disketten!
Es ist eine Menge Nützliches und Interessantes darunter. Seht Euch den nachstehenden Katalog an!
Übrigens: Diese Programme wurden nicht zusätzlich im Katalog der Diskothek aufgenommen, weil ein erschöpfender Katalog mit Erläuterungen schon existiert, wie Ihr seht. Sie befinden sich auf 40Tr./SS/SD-Disketten. Wenn jemand etwas davon wünscht, so erhält er grundsätzlich eine Kopie der kompletten Diskette, auf der sich sein(e) Wu(e)nsch(e) befindet! Meistens gehören ohnehin mehrere Files zu einem Programm. Und 40/SS/SD kann ja jeder lesen und sich ggf. umformatieren. - Eine wichtige Bitte:

Bei_Bestellungen_bitte_die_Disk-Nummer_angeben!

Nun kann's losgehen! Ich rechne mit einer Sintflut von Programm-Anforderungen (möglichst auf eigenen, noch unformatierten Disketten). Wer sie gleich selbst formatieren will, der verwende für TRS80:
TI=A TD=A TC=39 (oder 40) SPT=10 TSR=bel. GPL=2 DDSL=17 DDGA=2
bzw. die entsprechenden deutschen Parameter für das GENIE.

Viel Erfolg + Spaß mit der "PD"!
wünscht:

Euer KaJot

Unsere 'Public Domain' - Software

DISK 01 - NEWDOS80 PROGRAMME

APD/ASM	- Quellcode fuer ein PDRIVE-Automatikprogramm
APD/DOC	- Dokumentation fuer APD/ASM, APD1/CMD, APD3/CMD
APD1/CMD	- PDRIVE-Automatik fuer das Modell I
APD3/CMD	- PDRIVE-Automatik fuer das Modell III
CATALOG1/BAS	- Diskettenkatalog-Programm fuer NEWDOS80-Benutzer
COMPDIR/BAS	- Vergleicht die DIRECTORY zweier Disketten und gibt das Ergebniss auf dem Drucker, Bildschirm bzw. als Diskfile aus.
COMPDIR/DOC	- Dokumentation fuer COMPDIR/BAS
DIALND/CMD	- Terminalprogramm (waehlt Telephonnummern autom.)
EDTASMN/TXT	- Kurzanleitung fuer EDTASMN/CMD
FASTMENU/CMD	- Wahlmenue, ermoeglicht Programmstart mit einem Tastendruck (nur Modell I)
HELP/CMD	- Erklaert alle NEWDOS-Befehle
NAMEIT/BAS	- Ermoeglicht Aenderung der NEWDOS80-Meldung beim booten der Systemdiskette
NAMEIT2/BAS	- Ermoeglicht Aenderung der NEWDOS80 READY-Meldung und des BASIC-Logos
PDRIVE/BAS	- Analysiert die PDRIVE-Daten einer Fremdiskette
REROUTE/ASM	- Quellcode fuer REROUTE/CMD, Dokumentation in Kommentarzeilen
REROUTE/CMD	- Leitet die Druckerausgabe auf ein Diskfile um
TRSDIR/ASM	- Quellcode fuer TRSDIR/CMD
TRSDIR/CMD	- Zeigt die Directory einer Modell III TRSDOS 1.3 Diskette unter NEWDOS80 (Modell I/III) an
TRSDIR/DOC	- Dokumentation fuer TRSDIR/ASM und TRSDIR/CMD

DISK 02 - VERSCHIEDENE MODELL I/III PROGRAMME

CALC/CMD	- Erlaubt Berechnungen auf Betriebssystemebene, Syntax : CALC Formel z.B.: CALC 2/4+5
CHECKER/BAS	- Vergleicht zwei im ASCII-Format abgespeicherte BASIC-Programme und druckt die Unterschiede aus
COMPARE/CMD	- Vergleicht zwei Files byteweise, Syntax: COMPARE FILE1 FILE2
DISKINDX/BAS	- Druckt DISKINDX/DAT (von DISKINDX/CMD)
DISKINDX/CMD	- Diskettenkatalogisierprogramm (max. 2500 Prg.)
DISKINDX/DOC	- Anleitung fuer DISKINDX/CMD
DOU/CMD	- Fuehrt eine Sequenz von DOS-Befehlen (aehnlich /JCL) aus
DOU/DOC	- Dokumentation zu DOU/CMD
LOADADDR/BAS	- Analysiert /CMD-Files und gibt die Ladeadresse im RAM an
NOTE/CMD	- Ermoeglicht das Anlegen von Notizen, Kommentaren und Instruktionen auf der Disk. Selbsterklaerend
ORGAN/CMD	- Gibt Musik ueber den Cassettenport wieder
PERUSE/CMD	- Erzeugt ein Programm, welches Textfiles in Lese-geschwindigkeit auflistet. Syntax: PERUSE TEXT/TXT
PILOT/BAS	- Dieses BASIC-Programm poked einen PILOT-Interpreter in den oberen RAM-Bereich
PILOT/INS	- Instruktionen fuer PILOT/BAS in PILOT. Programmablauf: RUN "PILOT/BAS" LOAD "PILOT/INS" NAME
REPLACER/BAS	- Benennt Variablennamen in BASIC-Programmen um
REPLACER/DOC	- Dokumentation zu REPLACER/BAS
SCHEDULE/BAS	- Terminkalender in BASIC
SETDATE/ASM	- Quellcode fuer SETDATE/CMD; Dokumentation in Kommentarzeilen
SETDATE/CMD	- Verbesserte Datumseingabe auf DOS-Ebene

HEFT 15 September 1986

SUPERLST/CMD - Listet BASIC-Programme uebersichtlicher
SUPERLST/DOC - Dokumentation zu SUPERLST/CMD

DISK 03 - TERMINAL- und KOMMUNIKATIONS-PROGRAMME

BINHEX/CMD	- Konvertiert ASCII Hex Files in Binaerfiles und umgekehrt. Gibt die Checksumme für jedes File an
BUSY/CMD	- Beantwortet ankommende Anrufe mit "Computer is busy"; Backgroundtask fuer autom. antwortende Modems
LHELP/TXT	- Help file fuer LTERM/CMD
LTERM/CMD	- LTERM Terminalprogramm Version 3.0a
LTERM/DOC	- Dokumentation yu LTERM/CMD
LYNXTERM/CMD	- STERM 1.6 an das LYNX-Modem angepasst
LYNXTERM/DOC	- Dokumentation zu LYNXTERM/CMD
MODEM/CMD	- Modem Terminalprogramm Version 1.1b
MODEM/DOC	- Dokumentation zu MODEM/CMD
ORCONV/CMD	- Konvertiert ORCHESTRA-80/85/90 Files in ASCII und umgekehrt
ORCONV/DOC	- Dokumentation zu ORCONV/CMD
STERM/CMD	- STERM Terminalprogramm Version 1.6b
STERM/DOC	- Dokumentation zu STERM/CMD
XMOD1200/CMD	- 1200 Baud Version von XMODEM, ein Filetransferprogramm. Autom. Fehlererkennung und -korrektur
MOD300/CMD	- 300 Baud Version von XMODEM
XMODEM/DOC	- Dokumentation zu XMODEM1200 und XMODEM300

DISK 04 - VERSCHIEDENE MODELL 4, MULTIDOS. DOSPLUS. CASS.-PRG.

CLRDIR/CMD	- Loescht unter MULTIDOS 1.x unbenutzte Directory Eintraege; Syntax CLRDIR :1
DCTDSP/CMD	- Zeigt das "DEVICE CONTROL TABLE" unter TRSDOS6.x an (Modell 4)
DISKCAT/BAS	- Diskettenkatlogisierprogramm unter DOSPLUS
DISKCAT/DOC	- Dokumentation zu DISKCAT/BAS
FREEMAP/CMD	- Zeigt die freien "GRANULES" unter MULTIDOS 1.x an Syntax: FREEMAP :1
KBMOD/ASM	- Quellcode fuer KBMOD/CMD; incl. Kommentarzeilen
KBMOD/CMD	- Modell 4 Programm; installiert einen "FULL-SCREEN-EDITOR" und die hohe Taktfrequenz im Model III-Modus
LABEL4/BAS	- Druckt auf dem Modell 4 Disklabels
LCOMM/TXT	- Benutzungsanleitung fuer LCOMM
MEMDISK4/CMD	- Installiert auf dem Modell 4 eine RAM-DISK
MOD4BASC/BAS	- Konvertiert Modell I/III BASIC-Programme in Modell 4 BASIC-Programme
MOD4BASC/DOC	- Dokumentation zu MOD4BASC/BAS
MOD4INFO/TXT	- Patches fuer TRSDOS 6.0 und weitere M.4 Infos
MODIII4F/BAS	- BASIC-Subroutines, um auf dem Modell 4 im Modell III-Modus die Taktfrequenz umzuschalten
MOVESYS4/JCL	- Modell 4 JCL-File um alle SYS-Files auf die RAM-DISK zu uebertragen
NEWLIST/CMD	- BASIC Programmlister Version 6.5; unter DOSPLUS, eventuell auch unter anderen Betriebssystemen
NEWLIST/DOC	- Dokumentation zu NEWLIST/CMD
ORGAN4/CMD	- Ermoeglicht Musikwiedergabe ueber das Modell 4 Sound Modul
SCANCAT/BAS	- Utility fuer DISKCAT/BAS; listet und sucht im Diskettenkatalog
SOUND134/BAS	- Demonstration von Sound-Routinen. für M.I/III/4
SPEED/CMD	- Utility fuer das Modell 4; SPEED 4 bzw. SPEED 2 stellen die gewuenschte Taktfrequenz der CPU ein
TEXEDIT4/BAS	- ASCII File-Editor Version 4.1 fuer das Modell 4
TTERM16K/BAS	- Terminal-Programm fuer 16 K Cassettensystem-User
TTERM32K/BAS	- Terminal-Programm fuer 32 K Cassettensysten-User

DISK 05 - PROGRAMME VON NORTHERN BYTES

ART6809/TXT	- Dokumentation zu ASM6809/BAS, PONG/ASM, SETRES/ASM und VARS/TXT
ASM6809/BAS	- Ein 6809 Assembler in MICROSOFT BASIC
CENTRJKL/ASM	- Eine NEWDOS80-JKL-Routine für CENTR. 739 Drucker
COCOPY/BAS	- Ein Modell 100 Assembler File Transferprogramm Dokumentation in den ersten Programmzeilen
EDPATCH/ASM	- Passt EDTASMN PLUS fuer das M.1 an das M.3 an
EDPATCH/TXT	- Dokumentation zu EDPATCH/ASM
EPSONJKL/ASM	- NEWDOS80-JKL-Routine fuer EPSON Drucker
FDCIII/BAS	- WESTERN DIGITAL 1793 Floppy Disk Controller-Test
FREEMAP/CMD	- Zeigt die freien Granules unter MULTIDOS 1.x Syntax: FREEMAP :1
JKL/TXT	- Dokumentation zu CENTRJKL/ASM und EPSONJKL/ASM
PONG/ASM	- Ein Beispiel 6809 - Assemblerpr. für ASM6809/BAS
SETDATE/CMD	- Verbesserte Datumseingabe beim Booten der Disk. Verbesserte Version des Programms auf Disk 02 Dokumentation in NORTHERN BYTES, Volume 5, No.2
SETRES/ASM	- Ein Beispiel 6809 - Assemblerpr. für ASM6809/BAS
V/CMD	- Filevergleichsprogramm, verbesserte Version von COMPARE/CMD auf Disk 02, Syntax: V File1 File2
VARS/TXT	- Liste aller benutzten Variablen von ASM6809/BAS
ZAPDEBUG/ASM	- QUELLCODE fuer ZAPDEBUG/CMD
ZAPDEBUG/CMD	- Zap Utility fuer TRSDOS 1.3B
ZAPDEBUG/TXT	- Dokumentation zu ZAPDEBUG/ASM und ZAPDEBUG/CMD

DISK 06 - VORWIEGEND SPIELPROGRAMME

COMPDIR/BAS	- NEWDOS80-Utility, vergleicht die Directories zweier Disketten und gibt das Ergebnis auf dem Schirm, Drucker und als Diskfile wieder. Verbesserte Version von COMPDIR/BAS auf DISK 01
CONCEN/BAS	- Konzentrationsspiel (PUZZLE)
EMPIRE/BAS	- Ein Wirtschaftssimultionsprogramme fuer mehrere Spieler, die Staaten regieren
FLIGHTDC/BAS	- Dokumentation zu FLIGHTSM/BAS
FLIGHTSM/BAS	- Flugsimulator
OTHELLO/BAS	- OTHELLO-Spiel fuer einen oder mehrere Spieler
REACTOR/BAS	- Simuliert einen Kernreaktor (mit Graphik)
TTT/ASM	- Quellcode fuer TTT/CMD
TTT/CMD	- Dreidim. TICTACTOE (in einer 4x4x4-Matrix)

DISK 07 - PROGRAMME AUS NORTHERN BYTES VOLUME 5 NO. 5

BASICOMP/BAS	- Ein BASIC Compiler fuer die Modelle I u. III
BASICOMP/DOC	- Dokumentation zu BASICOMP/BAS u. SAMPLE/BAS
BINCLOCK/ASM	- Quellcode zu BINCLOCK/CMD
BINCLOCK/CMD	- "BINAERUHR" fuer die Modelle I u. III
BINCLOCK/DOC	- Dokumentation zu BINCLOCK/CMD
CODE/ASM	- Quellcode für CODE/CMD, Dokumentation in Kommentarzeilen
CODE/CMD	- Ver- und entschluesselt ASCII-Files
SAMPLE/BAS	- Beispielprogramm zu BASICOMP/BAS
SD456/ASM	- Quellcode zu SD456/CMD (mit mehrfachen DEFB-Statements, siehe SD456/DOC)

R.t. 2../7

SD456/CMD	- NEWDOS80-SCREEN DUMP-Routine fuer die Modelle I u. III (SCREEN auf Disk)
SD456/DOC	- Dokumentation zu SD456/CMD
VERFILE/ASM	- Quellcode zu VERFILE/CMD
VERFILE/CMD	- Fileverifizierungsprogramm, prueft auf Disklesefehler
VERFILE/DOC	- Dokumentation zu VERFILE/CMD

DISK 08 - PROGR. AUS NORTHERN BYTES UND ZWEISPRACHIGER HANGMAN

ALARM/BAS	- NEWDOS80-UTILITY fuer das MODELL I, beim booten ertönt Alarm, wenn eine bestimmte Tastenkombi nation nicht gedrueckt wird
ALARM/DOC	- Dokumentation zu ALARM/BAS
DIALER/ASM	- Quellcode zu DIAL/CMD, Dok. in Kommentarzeilen
DIALER/CMD	- Waehlt automatisch Telephonnummern
ENGLANTO/DO1	- Datenfile fuer HANGMAN/BAS
ENGLFREN/DO1	- " " " / "
ENGLSPAN/DO1	- " " " / "
ENGLSYNO/DO1	- " " " / "
EXPRESS/BAS	- "Expression Input Routine" fuer M. 1 und 3 BASIC
EXPRESS/DOC	- Dokumentation zu EXPRESS/BAS
HANGMAN/BAS	- Zweisprachiger "HANGMAN", Wortspiel zum Erlernen von Fremdsprachen, Synonymen und Antonymen
HANGMAN/DOC	- Dok.zu HANGMAN/BAS und zugeh. Datenfiles
INOUT/FOR	- Ein- und Ausgaberoutinen unter MICROSOFT FORTRAN
LANGUAGE/DAT	- Datenfile fuer HANGMAN/BAS
LPDESC/ASM	- Quellcode um DATA-Statements in LPDESC/BAS zu erzeugen
LPDESC/BAS	- Ermoeglicht das Drucken von Kleinbuchstaben mit echten Unterlängen auf: LP VII und DMP-100
LPDESC/DOC	- Dokumentation zu LPDESC/ASM und LPDESC/BAS
MERGELN/ASM	- Quellcode zu dem Maschinenprogramm in den DATA-Zeilen von MERGELN/BAS
MERGELN/BAS	- Erlaubt schnelles und einfaches Aneinanderfuegen von BASIC-Programmzeilen
MERGELN/DOC	- Dokumentation zu MERGELN/ASM u. MERGELN/BAS
SUBINOUT/TXT	- Dokumentation zu INOUT/FOR u. TESTIO/FOR
SUPASS/ASM	- Quellcode fuer SUPASS/CMD
SUPASS/CMD	- Gibt die richtige "Checksumme" für die Seriennummer der SUPERUTILITY an (nützlich fuer die Umwandlung des Programmes in ein CMD-File)
SUPASS/DOC	- Dokumentation zu SUPASS/CMD
TESTIO/FOR	- Demonstrationsprogramm fuer INOUT/FOR
TRSPATCH/BAS	- Modifiziert TRSDOS1.3 fuer das Modell III
WORDIN/BAS	- Erzeugt neue Datenfiles fuer HANGMAN/BAS

DISK 09 - MINI-BBS UND PROGRAMME AUS NORTHERN BYTES

DIRSLOT/ASM	- Quellcode fuer DIRSLOT/CMD
DIRSLOT/CMD	- NEWDOS80-Utility, ermoeglicht das Einfuegen von Filenamen an beliebiger Stelle der DIRECTORY
DIRSLOT/DOC	- Dokumentation fuer DIRSLOT/CMD
DR/ASM	- Quellcode fuer DR/Cmd
DR/CMD	- "Disk Route Programm" fuer NEWDOS80V2
DR/DOC	- Dokumentation zu DR/CMD
INTERRPT/ASM	- Quellcode fuer INTERRPT/CMD
INTERRPT/CMD	- NEWDOS80 Modell I-Utility, ermoeglicht die Unterbrechung von laufenden Programmen, diese werden auf der Disk abgespeichert und koennen spaeter wieder fortgesetzt werden
INTERRPT/DOC	- Dokumentation fuer INTERRPT/CMD
MINIBBS2/BAS	- Teil des MINI-BBS-Systems von MIKE BERNSTEIN
MINIBBS2/DOC	- Dokumentation zu MINIBBS2/BAS, MINIBBS2/JCL, MINIHOST/ASM, MINIHOST/CMD und XMODEM30/CMD
MINIBBS2/JCL	- Teil des MINI-BBS-Systems von MIKE BERSTEIN
MINIHOST/ASM	- " " " - " - " " " "
MINIHOST/CMD	- " " " - " - " " " "
MOVE/SRC	- Kopiert das Modell 4-ROM ins RAM; im M.3-Modus
VCLIST/BAS	- Listet VISICALC /VC-Files auf Drucker, Screen u. als Diskfiles
VCLIST/DOC	- Dokumentation zu VCLIST/BAS
XMODEM30/CMD	- Teil des MINI-BBS-Systems von MIKE BERNSTEIN

DISK 10 - PROGRAMME AUS NORTHERN BYTES, VOL. 5, NO. 7

ALDATE/ASM	- Ergaenzt ALLWRITES ALK/CMD-Programm, übergibt automatisch das Systemdatum an Briefe
ALDATE/DOC	- Dokumentation zu ALDATE/ASM
BOOT/BAS	- Modifiziert die NewDOS-Bootmeldung (Model 1)
DIALER/BAS	- Waehlt Telephonnummern mit dem Cassetten-Relais, Modell III-Benutzer: zuerst das Programm listen!
FILTOMCI/BAS	- Wandelt alle Modell I/III-Files um, dass sie per Modem uebertragen werden koennen
LINE/BAS	- Zeichnet Linien zwischen zwei beliebigen Punkten auf dem Bildschirm
LINE/DOC	- Dokumentation zu LINE/BAS
MCIPGMS/DOC	- Dokumentation zu FILTOMCI/BAS und MCITOFIL/BAS
MCITOFIL/BAS	- Wandelt den "OUTPUT" von FILTOMCI/BAS ins Originalformat um
MENINSTL/ASM	- Quellcode fuer MENINSTL/CMD
MENINSTL/CMD	- Installationprogramm fuer MENU/CMD
MENU/ASM	- Quellcode zu MENU/CMD
MENU/CMD	- Installiert auf dem Modell III eine menuegesteuertes TRSDOS1.3
MENU/DOC	- Dokumentation zu MENU/CMD und MENINSTL/CMD
REVSTR/ASM	- Quellcode zu dem Maschinenprogramm in den DATA-Zeilen von REVSTR/BAS
REVSTR/BAS	- Dieses BASIC-Programm gibt das "REVERSE" von Strings aus. USR-Routine!
REVSTR/DOC	- Dokumentation zu REVSTR/ASM und REVSTR/BAS
WHERE/ASM	- Quellcode zu WHERE/CMD
WHERE/CMD	- Gibt die genaue Position von Fehlern in BASIC-Programmen an
WHERE/DOC	- Dokumentation zu WHERE/CMD

DISK 11 - NEWDOS80-PROGRAMME (VON J. VABULAS) UND VERMISCHTES)

BOOT43A/CIM	- Ermoeglicht das BOOTEN eines Modell 4 mit einer DOSPLUS3.5-Diskette (mit MODELA/III-File)
BOOT43A/DOC	- Dokumentation zu BOOT43A/CIM
BRO/CTL	- SUPER SCRIPSIT Druckertreiber für BROTHER Typenraddrucker
COMP/ASM	- Quellcode zu COMP/CMD
COMP/CMD	- Schneller Filevergleich unter NEWDOS80V2
COMP/DOC	- Dokumentation zu COMP/CMD
D/CMD	- Menuegesteuertes Filehandling unter NEWDOS80V2
D/DOC	- Dokumentation zu D/CMD
DISKCOMP/ASM	- Quellcode fuer DISKCOMP/CMD
DISKCOMP/CMD	- NEWDOS80V2-Utility, vergleicht den Inhalt zweier Disketten
DISKCOMP/DOC	- Dokumentation zu DISKCOMP/CMD

HEFT 15 September 1986

HDSTART/ASM	-	Quellcode fuer HDSTART/CMD
HDSTART/CMD	-	Patch fuer NEWDOS80V2.5, ermoeglicht das gleichzeitige Betreiben einer HARDDISK mit dem HOLMES ENGINEERING SPRINTERMODUL (SPEED UP)
HDSTART/DOC	-	Dokumentation zu HDSTART/CMD
SD/ASM	-	Quellcode fuer SD/CMD
SD/CMD	-	Sortierte Directory mit (mit wildcards!)
SD/DOC	-	Dokumentation zu SD/CMD

Tut mir leid, Chef! Wir spielten eines dieser Mikrospiele, und ich brauchte die ganze Nacht, um aus dem Labyrinth zu entkommen.

DISK 12 - POTPOURRI

DATASET/ASM	-	Quellcode fuer DATASET/CMD; mit Kommentarzeilen
DATASET/CMD	-	Ähnlich SETDATE (DISK 02); Datum im T/M/J-Format
DISTANCE/BAS	-	Berechnet die Distanz und den Kurs zwischen zwei Staedten
DS/ASM	-	Quellcode fuer DS/CMD
DS/CMD	-	Wie DATESET/CMD, schreibt das Datum als Diskfile (DATE/TXT) auf Disk, dadurch Uebernahme in die Textverarbeitung moeglich
GM/BAS	-	Attraktives Menue-Programm, Anpassung notwend.
MAP/CMD	-	Utility fuer MULTIDOS, gibt die Lage von Files auf der Diskette an. Syntax: MAP :drive
MSTRMIND/NB	-	MASTERMIND in NEWBASIC (MODULAR SOFTWARE)
P/CMD	-	Stellt die Druckmodi von ITOH-Druckern ein
SD/ASM	-	Quellcode zu SD/CMD, mit Kommentarzeilen
SD/CMD	-	Wie SETDATE/CMD (DISK 02), schreibt Datum als Textfile auf Disk; Uebernahme in Textverarbeitung moeglich
SETPRT/BAS	-	Quelltext zu P/CMD - fuer ZBASIC-Compiler
SUNRISE/BAS	-	Angabe von Sonnen Auf- und Untergangszeiten fuer alle geographischen Orte (Angabe in GMT)
WEEKDAYS/NB	-	Ein Spiel in NEWBASIC (MODULAR SOFTWARE)

DISK 13 - Programme aus NORTHERN BYTES VOLUME 6, No. 1-2

AUSDATE/ASM	-	Quellcode zu AUSDATE/CMD, mit Kommentarzeilen
AUSDATE/CMD	-	Revidierte Version von DATESET/CMD (DISK 12), speichert Datum im Tag/Monat/Jahr-Format
BLANK/ASM	-	Quellcode zu BLANK/CMD, mit Kommentarzeilen
BLANK/CMD	-	Modell III-Utility, blendet Bildschirm bei Nichtbenutzung des Rechners nach einer Wartezeit aus, nach beliebiger Tastatureingabe Restauration des Bildschirminhaltes (schont dem Monitor)
BLANK1/CMD	-	Modell I-Version von BLANK/CMD (Zeile 1810 im Quellcode ist von 447BH auf 4410H geaendert)
DEMO	-	BASIC-Programm, Demo zu BLANK/CMD, Start mit START/JCL
FIXHIT/ASM	-	Quellcode zu FIXHIT/CMD
FIXHIT/CMD	-	NEWDOS80V2-Utility, rekonstruiert bei defekten Disketten das HASH INDEX TABLE (alle Formate)
FIXHIT/DOC	-	Dokumentation fuer FIXHIT/CMD
FKEY/ASM	-	Quellcode zu FKEY/CMD
FKEY/CMD	-	M. 4-TRSDOS-Utility, weist den Funktionstasten neue Codes (vom Benutzer definierte) Codes zu
FKEY/DOC	-	Dokumentation zu FKEY/CMD
FNDEMO/BAS	-	Demonstriert den Gebrauch von Funktionen in BASIC-Programmen
FNDEMO/DOC	-	Dokumentation zu FNDEMO/BAS
FX80/BAS	-	Laedt die Standard-TRS80-Blockgraphik in den FX80, so das EPSON MX80-Graphik-Programme lauffaehig werden
FX80/DOC	-	Dokumentation zu FX80/BAS
PROMPT/BAS	-	Modell I-Utility, ermoeglicht Aenderung des "BASIC-Prompts: READY"
START/JCL	-	JCL-FILE, initialisiert BLANK/CMD und DEMO
TRACE/ASM	-	Quellcode zu TRACE/CMD
TRACE/CMD	-	Aehnlich der TRACE-Funktion unter TRSDOS2.3 aber fuer NEWDOS80V2
TRACE/DOC	-	Dokumentation zu TRACE/CMD

DISK 14 - PROGRAMME AUS NORTHERN BYTES VOLUME 6, NO. 3

DGRAPH/BAS - BASIC-Demo fuer VDCTL/BAS
FIXGAT/ASM - Quellcde zu FIXGAT/CMD
FIXGAT/CMD - NEWDOS80V2-Utility, repariert den GAT-Sektor der Directory (SS und DS-Disks)
FIXGAT/DOC - Dokumentation zu FIXGAT/CMD
RD/ASM - Quellcode zu RD/CMD, inclusive Kommentarzeilen
RD/CMD - Revidierte Version von DS/CMD (DISK 12), DOS-Patch speichert Datum im TT/MM/JJ-Format
RD/DOC - Dokumentation zu RD/CMD
TATUPNI/ASM - Quellcode zu TATUPNI/CMD
TATUPNI/BAS - BASIC-Demo zu TATUPNI/CMD
TATUPNI/CMD - NEWDOS80V2-Utility, BASIC-Erweiterung, Input an beliebiger Bildschirmposition. M. III-TRSDOS -Version : Zeilen 350-370 im Quellcode aendern!
TATUPNI/DOC - Dokumentation zu TATUPNI/CMD
VDCTL/ASM - Quellcode fuer VDCTL/OBJ
VDCTL/DOC - Dokumentation zu VDCTL/OBJ
VDCTL/OBJ - Modell 4-Utility, erlaubt direkten Zugriff auf das VIDEO-RAM

DISK 15 - "THE CHICAGO GREENE MACHINE BBS", UND WHERE/CMD

```
AD           - ********************************************
BBS/JCL      - * Folgende Programme gehoeren zu            *
BLDMENU/BAS  - * "THE CHICAGO GREENE MACHINE BULLETIN BOARD" *
CHATFLAG     - * Von George Matyaszek, Sysop               *
CHATFLAG/ORG - * Infos unter HOWTORUN/TXT                  *
COMMAND/BAS  - * Nur fuer TRS80 Modell I unter NEWDOS80V2  *
DFTBUTS/CMD  - *                                           *
DLMOD/BAS    - *     (EIN KOMPLETTES MAILBOX-SYSTEM)       *
DRIVER/CMD   - *                                           *
EMAIL        - *                                           *
ENTRY/BAS    - *                                           *
HOWTORUN/TXT - *                                           *
INFO FIL     - *                                           *
LOGRED/BAS   - *                                           *
MANAGER/BAS  - *                                           *
MODEM881/CMD - *                                           *
PACKER/BAS   - *                                           *
USE          - ********************************************
```

WHERE/ASM - Quellcode fuer WHERE/CMD, verbesserte Version des Programmes auf DISK 10.
WHERE/CMD - Genaue Angabe der Fehlerposition in BASIC-Programmzeilen
XMODEM/CMD - Gehoert zu "THE CHICAGO GREENE MACHINE BBS"

DISK 16 - VERSCHIEDENE MODELL I/III-PROGRAMME

CAT/CMD - NEWDOS80-UTILITY, gibt die Belegung der Diskette an. Nach Filenamen, relative Sektoren und Spur- und Sektornummer
CAT/DOC - Dokumentation zu CAT/CMD
CAT/SRC - Quellcode fuer CAT/CMD
CLAWDOS/CMD - Modell III "MINI OPERATING SYSTEM" im TRSDOS1.3-FORMAT
CLAWDOS/DOC - Dokumentation zu CLAWDOS/CMD
CLAWDOS/SRC - Quellcode zu CLAWDOS/CMD
CLONEI/CMD - NEWDOS80- und TRSDOS1.3-Utility, sehr schnelles "filecopy"
CLONEI/DOC - Dokumentation zu CLONEI/CMD
CLONEI/SRC - Quellcode fuer CLONEI/CMD
DIS/CMD - NEWDOS80-Utility, sortierte Directory, in fuenf Spalten (statt vier)
DIS/DOC - Dokumentation zu DIS/CMD
DIS/SRC - Quellcode fuer DIS/CMD
FORM/CMD - Schnelles Diskettenformatierungsprogramm mit vielen Optionen, erzeugt keine Directories
FORM/DOC - Dokumentation zu FORM/CMD
FORM/SRC - Quellcode fuer FORM/CMD
GRAPHPRO/BAS - Graphikeditor, gibt die Graphik in PRINT-Statements aus, die in BASIC-Programme eingefuegt werden koennen
GRAPHPRO/DOC - Dokumentation zu GRAPHPRO/BAS
PHONETXT/BAS - Untersucht amerikanische Telephonnummern auf enthaltene Buchstabenkombinationen
SUPCALC2/BAS - Taschenrechnerprogramm in doppelter Genauigkeit (inclusive Konstanten: PI, E,etc.)
SUPCALC2/DOC - Dokumentation zu SUPCALC2/BAS

DISK 17 - VIDEO STORE MANAGER UND DIV. PROGRAMME VON M. PATRICK

CONVBIN/ASM - Quellcode fuer CONVBIN/CMD
CONVBIN/CMD - Gibt fuer jede Tastatureingabe de Binaercode an (Modell I/III)
TRACK/CMD - Komplettes Verwaltungsprogramm fuer einen VIDEO-Laden (Unter TRSDOS1.3 oder DOSPLUS, ev. andere DOSSE). Der Quellcode benoetigt einen Assembler der Get bzw. Include Files unterstuezt
TRA/ASM - Teil A des Quellcodes fuer TRACK/CMD
TRB/ASM - Teil B des Quellcodes fuer TRACK/CMD
TRC/ASM - TEIL C des Quellcodes fuer TRACK/CMD

DISK 18 - HELP FILE GENERATOR, DOKUMENTATION FUER VIDEO STORE MANAGER UND DIVERSES (VON MAL PATRICK)

TRACK/DOC - Dokumentation fuer TRACK/CMD von Disk 17
UNIHELP/ASM - Quellcode zu UNIHELP/CMD
UNIHELP/CMD - Modell I/III-Utility fuer alle Betriebssysteme. Generiert und editiert HELP-FILES
UNIHELP/DOC - Dokumentation fuer UNIHELP/CMD
WSMAPPED/DOC - Text: Anleitung zum "patchen" von WordStar3.0 unter MONTEZUMA CPM2.2 (Modell 4). Die Bildschirmroutinen werden wesentlich beschleunigt

DISK 19 - KEYBOARD MACRO, FILE UTILITIES UND ANDERE PROGRAMME

KEYMAC/ASM - Quellcode zu KEYMAC/CMD
KEYMAC/CMD - Model 3-Utility, Tastaturerweiterung um Woerter bzw. um Wortfolgen. Abspeicherung der Tastaturdefinitionen auf der Disk. Die Definitionen duerfen auch "RETURNS" enthalten. UNMACRO/CMD beendet diese Utility
KEYMAC/DOC - Dokumentation zu KEYMAC/CMD
MASTER/ASM - Quellcode des Hauptmoduls von VFU/CMD
NOCAT/ASM - Quellcode fuer NOCAT/CMD
NOCAT/CMD - NEWDOS80-Utility, Modell I/III, druckt sortierten Filekatalog auf dem Bildschirm. Benutzt eine "Screen Sort "-Routine. Fuer DOSPLUS3.4 sind zwei Patches notwendig.
SECT1/ASM - Quellcode des 1. Untermoduls von VFU/CMD
SECT2/ASM - " " 2. " " " / "
SECT3/ASM - " " 3. " " " / "
UNMACRO/ASM - Quellcode fuer UNMACRO/CMD
UNMACRO/CMD - Entfernt KEYMAC/CMD aus dem RAM und setzt den HIGHMEM POINTER auf den gewuenschten Wert
VFU/CMD - Modell I-NEWDOS80V2-Utility (die Patches fuer DOSPLUS sind im Quellcode vermerkt), vielseitige File-Utility mit eingebauter HELP-Funktion. Keine modifizierte Version von VFU/CMD von MULTIDOS!
VFU/DOC - Dokumentation zu VFU/CMD

DISK 20 - WEITERE PROGRAMME VON MEL PATRICK

ARROWPRT/ASM - Quellcode fuer ARROWPRT/CMD
ARROWPRT/CMD - Umdefinition der Pfeiltasten. Die Pfeiltasten werden auf EPSON Druckern als Pfeile gedruckt. Alle Betriebssystem, Modell I/III
DISCAT/ASM - Quellcode fuer DISCAT/CMD
DISCAT/CMD - Creiert Diskettenkatalog (NEWDOS80V2 u. DOSPLUS)
DOTPRINT/DOC - Textfile: Anleitung um ältere DOTPRINT-Versionen ALLWRITE anzupassen.
HEXCAL/ASM - Quellcode fuer HEXCAL/CMD
HEXCAL/CMD - Hexadezimal Calculator, arithm. Funktionen, log. Verknuepfungen (AND, OR, XOR). Eingebaute HELP-Funktion. Ferner sind Anzeige von Speicherinhalten und deren Modifikation moeglich. Alle Betriebssysteme, Modell I/III
MARO/ASM - Quellcode fuer MACRO/CMD
MARO/CMD - Generiert einen Laufrahmen um den Bildschirm. Alle Betriebssysteme, Modell I/III
NEWBLD/ASM - Quellcode fuer NEWBLD/CMD
NEWBLD/CMD - Ersetzt das CHAINBLD/BAS der NEWDOS80-Diskette. Benoetigt weniger Speicherplatz
NEWBLD/DOC - Dokumentation zu NEWBLD/CMD

DISK 21 - DRUCKER PROGRAMME UND FILEVERGLEICHS-UTILITIES

COMPARE/ASM - Quellcode fuer COMPARE/CMD
COMPARE/CMD - Einfache Filevergleichs-Utility, Modell I/III, alle Betriebssysteme
EDTFMT/ASM - Quellcode fuer EDTFMT/CMD
EDTFMT/CMD - Formatierte Druckerausgabe (Titel, Seitennummer) von Standard EDTASM-Files. Optionen: Aneinanderfuegen mehrerer Files, entfernen von Kommentarzeilen. Modell I/III, alle Betriebssysteme
FONTDVR/ASM - Quellcode fuer FONTDVR/CMD

FONTDVR/CMD - Utility fuer ELECTRIC PENCIL. Druckt den mit dem von FONTWRTR/CMD creierten Zeichensatz. Arbeitet auch mit beliebigen BASIC-Programmen zusammen. Modell I/III, alle Betriebsysteme
FONTDVR/DOC - Dokumentation zu FONTDVR/CMD
FONTWRTR/ASM - Quellcode fuer FONTWRTR/CMD
FONTWRTR/CMD - Creiert Zeichensatz fuer EPSON Drucker. Modell I/III, alle Betriebssysteme. Benutzung zusammen mit FONTDVR/CMD
FONTWRTR/DOC - Dokumentation zu FONTWRTR/CMD

DISK 22 - KOMMUNIKATIONSPROGRAMME UND VERMISCHTES (MEL PATRICK)

ASCFMT/ASM - Quellcode fuer ASCFMT/CMD
ASCFMT/CMD - Formatierte Druckerausgabe von BASIC-Programmen (im ASCII-Format), Option: normal/single statement pro Druckzeile, M. 1/3, alle Betriebss.
BYE/ASM - Quellcode fuer BYE/CMD
BYE/CMD - Utility fuer SMARTDVR/CMD. Erlaubt dem Anrufer, sich durch die Eingabe von BYE (im DOS READY Mod.) aus dem System auszuloggen. Dokumentation in SMARTDVR/DOC.
DIALER/ASM - Quellcode fuer DIALER/CMD
DIALER/CMD - Ergänzung für Terminalprogramme. Erstellt Listen von Telefonnummern und wählt daraus Nummern automatisch an. Start durch gleichzeitiges Drücken 5,6,7-Tasten. Fuer Patches siehe den Quellcode. Modell I/III, alle Betriebssysteme
HAYES/ASM - Quellcode fuer HAYES/CMD
HAYES/CMD - Sendet "AT"-Code ans HAYES-Modem (zusammen mit HOST80/CMD), Modell I/III, alle Betriebssysteme
HOST80/ASM - Quellcode fuer HOST80/CMD
HOST80/CMD - "HOST DRIVER PROGRAM", arbeitet mit den meisten Direktmodems zusammen. Viele Optionen, Modell I und III, alle Betriebssysteme
MORSE/ASM - Quellcode fuer MORSE/CMD
MORSE/CMD - Morseuebungsprogramm, Modell I/III, alle Betriebssysteme
SMARTDVR/ASM - Quellcode fuer SMARTDVR/CMD
SMARTDVR/CMD - Bindet das HAYES SMARTMODEM ins System ein. Kann als Treiberprogramm für ein BBS-System verwendet werden
SMARTDVR/DOC - Dokumentation zu SMARTDVR/CMD und BYE/CMD

DISK 23 - FILE MANAGER PROGRAMM (VON DAN SHELBY) UND PROGRAMME

AUS NORTHERN BYTES VOLUME 6,NO. 4-5

BONDS/EES - Demonstrationsfile zu FILMGR48/BAS
BONDS/RPG - " " "
Siehe README/BAS und FILMGR48/DOC
FILMGR49/BAS - FILE MANAGER PROGRAMM von DAN SHELBY. Siehe README/BAS und FILMGR48/DOC
FILMGR48/DOC - Dokumentation zu DAN SHELBYS FILE MANAGER PROGR. Weitere Informationen in README/BAS
FILMGR48/FIX - FILE MANAGER Filekonvertierungsprogramm. Weitere Informationen in FILEGR48/DOC und README/BAS
FM48/JCL - FILE MANAGER JCL-File
MENU/DAT - Datenfile fuer das FILE MANAGER PROGRAMM
NDAMPRSD/TXT - Patch fuer das NEWDOS80-BASIC. wird als Voreinstellung fuer die "-Funktion angenommen

PLANNER/BAS	- Einfaches Finanzplanungsprogramm in BASIC
PLANNER/DOC	- Dokumentation zu PLANNER/BAS
PLATES/EBS	- Demonstrationsfile zu FILMGR48/BAS
PLATES/RPG	- " " " Siehe README/BAS und FILMGR48/DOC
README/BAS	- Ein BASIC-Programm, welches eine eine kurze Einfuehrung in FILMGR48 gibt. Beinhaltet kurze Beschreibungen von : BONDS/EBS, BONDS/RPG, FILMGR48/BAS, FILMGR48/DOC, FILMGR48/FIX, FM48/JCL, MENU/DAT, PLATES/RPG, README/BAS, SPEED/CMD und MIDGET/CRF
SCRPRT/BAS	- BASIC-Programm mit derselben Funktion wie SCRPRT/CMD. Nur M. 3 (arbeitet mit Cassette-Systemen)
SCRPRT/CMD	- Modell III-Utility, schnelle Druckausgabe des Bildschirms auf einigen TANDY Druckern
SCRPRT/DOC	- Dokumentation zu SCRPRT/CMD, SCRPRT1/CMD und SCRPRT/BAS
SCRPRT/SRC	- Quellcode fuer SCRPRT/CMD
SCRPRT1/CMD	- Modell I-Version von SCRPRT/CMD. Nur Disk
SPEED/CMD	- MODELL 4-Utility, ermöglicht im Modell III-Modus eine Taktfrequenz von 4 MHz. Arbeitet mit FILMGR48/BAS zusammen. Siehe README/BAS und FILMGR48/DOC
SZAP80/ASM	- Quellcode fuer SZAP80/CMD
SZAP80/CMD	- Modell III-Utility, ermoeglicht SUPERZAP im 80 x 24 Modus (VIDEO4)
SZAP80/DOC	- Dokumentation zu SZAP80/CMD
WIDGET/CRF	- Demonstrationsfile zu FILEGR48/BAS. Siehe README/BAS und FILEMGR48/DOC
WRDSRCH/BAS	- Wirklich funktionierender Kreuzworträtselgenerator
WRDSRCH/DOC	- Dokumentation zu WRDSRCH/DOC

DISK 24 - SETDATE + PROGRAMME AUS NORTHERN BYTES VOL. 6, NO.4-5

BINTNIB/CMD	- Verschlüsselt Files im umgekehrten NIBBLE FORMAT (mit NIBIBIN/CMD verschlüsselt)
GOSTMENU/ASM	- Quellcode fuer GOSTMENU/CMD
GOSTMENU/CMD	- Programmauswahlmenue in Maschinensprache
GOSTMENU/DOC	- Dokumentationn für GOSTMENU/CMD
HEAPSORT/BAS	- HEAPSORT-Demonstration in BASIC. Siehe SORTS/DOC
MAZE/ASM	- Quellcode für MAZE/CIM
MAZE/BAS	- BASIC-Programm mit einem Interface zu MAZE/CIM, zeichnet sehr schnell Labyrinthe
MAZE/CIM	- MASCHINENSPRACHE-ROUTINE, Sehr schneller Entwurf von LABYRINTHEN.
MAZE/DOC	- Dokumentation zu MAZE/ASM, MAZE/BAS, MAZE/CIM und MAZEGEN/BAS
MAZEGEN/BAS	- Generiert Lbabyrinthe, siehe MAZE/DOC
NIBTBIN/CMD	- Entschluesselt Files, die mit BINTNIB/CMD verschlüßelt worden sind. Syntax: NIBTBIN FILENAME
ROM/ASM	- Quellcode fuer ROM/CMD
ROM/CMD	- Speichert das Modell 3-ROM im oberen RAM-Bereich ab, und ermoeglicht damit ein schnelleres booten des Modell 4p
ROM/DOC	- Dokumentation zu ROM/CMD
SETDATE/CMD	- Modell 4-Version von SETDATE (DISK 02), beendet die laestige Datumsabfrage. beim booten des Betriebssystemes
SETDATE/FIX	- Patch fuer TRSDOS6.2. verhindert loeschen des Datums wenn SYSTEM (DATE=NO) gesetzt ist. Syntax: DO SETDATE/FIX
SETDATE6/ASM	- Quellcode der Modell 4-Version von SETDATE
SETDATE6/DOC	- Dokumentation zu SETDATE/CMD, SETDATE/FIX und SETDATE6/ASM
SHELSORT/BAS	- BASIC-Demo, demonstriert "SHELLSORT-Routine"
SORTS/DOC	- Dokumentation zu HEAPSORT/BAS und SHELSORT/BAS
SPS/JCL	- JCL-File zur Initialisierung von SPSMOD/CMD. Siehe SPSMOD/DOC fuer weitere Informationen
SPSMOD/CMD	- Modifiziert SCRIPSIT/LC, so dass einfache Berechnungen in der Textverarbeitung moeglich sind
SPSMOD/DOC	- Dokumentation zu SPS/JCL und SPSMOD/CMD
SPSMOD/SRC	- Quellcode fuer SPSMOD/CMD

Genau wie man mir gesagt hatte, mein Computer hat mich wirklich von allen Zwängen befreit.

- I M P R E S S U M -

1. Vorsitzende
Peter STEVENS
Postfach 56
4600 Dortmund 1
☎ 0231 /593883

2. Vorsitzende
Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

Hardwarekoordinator
Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar
Klaus-Jürgen MÜHLENBEIN
Am Mönchgarten 28
6940 Weinheim -Lü.
☎ 06201 /55852

Redaktion
Jens NEUEDER
Panoramastraße 21
7170 Michelbach /Bilz
☎ 0791 /42077

Redakteure dieser Ausgabe:

Bernd Drohwälder *	Klaus Hermann
Dieter Kasper *	Eckehard Kuhn
Klaus-J. Mühlenbein *	Kurt Müller
Jens Neueder *	Hartmut Obermann
Richard Rensch *	Gerald Schröder
Arnulf Sopp *	

sowie Artikel aus: 80 Micro

Bankverbindung des CLUB 80

Postgirokonto	Peter STEVENS
	Sonderkonto CLUB 80
Konto-Nummer	285 491 - 465
Postgiroamt	Dortmund
BLZ	440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Schluß

Hallo Club-80er,

wie in den vorherigen INFO's möchte ich auch hier wieder einige abschließende Worte schreiben.

Auch diesmal haben wir wieder ein über hundert Seiten starkes INFO zusammenbekommen, wofür ich mich bei den Mitwirkenden recht herzlich bedanken möchte. Gleichzeitig auch meinen Dank an diejenigen, die bei der Fragebogenaktion mitgemacht haben. Leider hat sich nur knapp über die Hälfte der Mitglieder daran beteiligt, ich hoffe aber doch, daß die Auswertung etwas Brauchbares für die Mitglieder enthält. Vielleicht sind doch einige neue Trents zu entdecken, über die in einer der nächsten INFO's zu berichten wäre. Vielleicht ist auch ein Thema dabei, daß den ein oder anderen einmal reizt, den Bleistift zu schwingen (in die Tasten zu hauen). Ich, als Redakteur, würde mich sehr darüber freuen.

Als weiteres liegt mir natürlich unser ECB-Bus-Projekt am Herzen. Nachdem nun die Grundeinheit fertig zum Ausliefern ist und sicher bald von den Beteiligten zusammengebaut sein wird, geht es nun an den "Harten Kern" der Sache -- den Computer --. Aus diesem Grunde haben wir in nächster Zeit wieder ein kleines "Hardwaretreffen" vereinbart. Dort werden wir dann die noch nötigen Einzelheiten absprechen. Auf jeden Fall geht es mit der "Bastelei" im nächsten INFO wieder weiter.

Einstweilen wünscht Euch viel Spaß an der neuesten INFO

Euer

J. Neueder

schon eine Zahl steht! Und eine Speicher-Adresse ist ein reiner Zahlenwert.
In diesem Fall wird das Register HL in Klammern gesetzt, dann weiß Ass bzw. die CPU: Nicht das Register HL, sondern die Speicherstelle, die in HL steht, soll mit dem Zahlenwert BFh gefüllt werden!
Steht nicht ein Register-Buchstabe, sondern eine Zahl in Klammern, so ist damit eine "Hausnummer", d.h. eine Speicher-Adresse gemeint.
Soll die Speicherstelle 3C00h mit BFh geladen werden (damit sie weiß wird), so müssen wir das also dem Register HL (in dem die Adresse 3C00h bereits steht) mitteilen, indem wir es einklammern.
Allgemein gilt immer, wenn eine Klammer auftritt:
"Lade die Speicherstelle, die du im Sinn hast, mit dem Wert sowieso!"
("Sinn" = Klammer.)

In unserem Falle ist der "Wert" ein Code, nämlich der ASCII-Code "BFh".
Und noch eine Falle hätte Herr Ass bei diesem Versuchs-Befehl vorgefunden: er hätte "BFh" für einen String, eben für die Buchstaben-Folge BFh gehalten. Wir meinten aber eine Hex-Zahl. Die akzeptiert er als solche nur, wenn sie mit einer Ziffer beginnt. Doch woher nehmen? Code 191 (Dez) fängt in HEX-Darstellung nunmal mit dem Buchstaben B an! "Was tun?" sprach ZEUS hier nicht; sondern: "Eine Null ist immer gut: sie schadet niemand - wenn was dahinter steckt!!" (Wie recht er hat...)
Jetzt ist also alles klar (??) :

LD (HL), 0BFh

ist richtig. Nicht HL wird mit BFh geladen, sondern (HL), das heißt:
die Speicherzelle, deren Adresse in HL steht,
und das ist ja (siehe 1.Befehl) die erste Bildschirm-Position!
Wir dürfen uns den Schweiß abwischen, den ZEUS vor unseren Erfolg gesetzt.
Denn nun arbeitet ASS für uns. Es bedarf nur noch eines Fußtrittes, unentschlossen wie ASS ist - aber dann rast er!
Der Tritt, der den Ball (Fußball 1986!) zum Sausen bringt:

*Lade den Inhalt der Quelle HL ins Ziel DE, kurz "**LD**";*
*Inkrementiere (erhöhe) beide um 1, kurz "**I**";*
vermindere gleichzeitig den Zähler BC um 1;
*Repeat (wiederhole) diesen Vorgang, kurz "**R**"*
und zwar solange, bis in BC eine Null steht.

Fassen wir diese Abkürzungen zusammen, so erhalten wir das schlaue Kürzel

LDIR

das dies alles allein zuwege bringt.
Jetzt ist der Bildschirm zwar nicht sauber - aber rein. So schnell schafft das kein Persil.
Was steht denn da in der 4.Spalte? Sieht aus wie Aushänge-Schilder. Sind es auch, wie sie vor wichtigen Häusern hängen (z.B. bei Friseuren, Bäckern und -Ähnlichem).
Gelegentlich muß man der CPU sagen, wo sie mit ihrer Arbeit anfangen und wo aufhören soll. Sie versteht "START" und "END". Nicht immer ist "START" identisch mit "ORG" (= origine = Ursprung, eigentlich also auch gleich Anfang; hier aber Anfang des Codes im Speicher, START hingegen ist das Startloch für die CPU, wie RUN für BASIC.
Übrigens, der Ordnung halber sei's gesagt, obschon der Club80 wohl namensgerecht nur den Mikroprozessor Z80 oder dessen Abkömmlinge betreibt: Alles hier Gesagte betrifft eben diesen; ich weiß nicht, ob alles z.B. für den "6502" (hochwohllöblicher COMMODORE) oder andere auch zutrifft. Wie gesagt: "OHNE GEWÄHR!"

Dies ist die „Ersatzseite"!
für S. 12 INFO 14

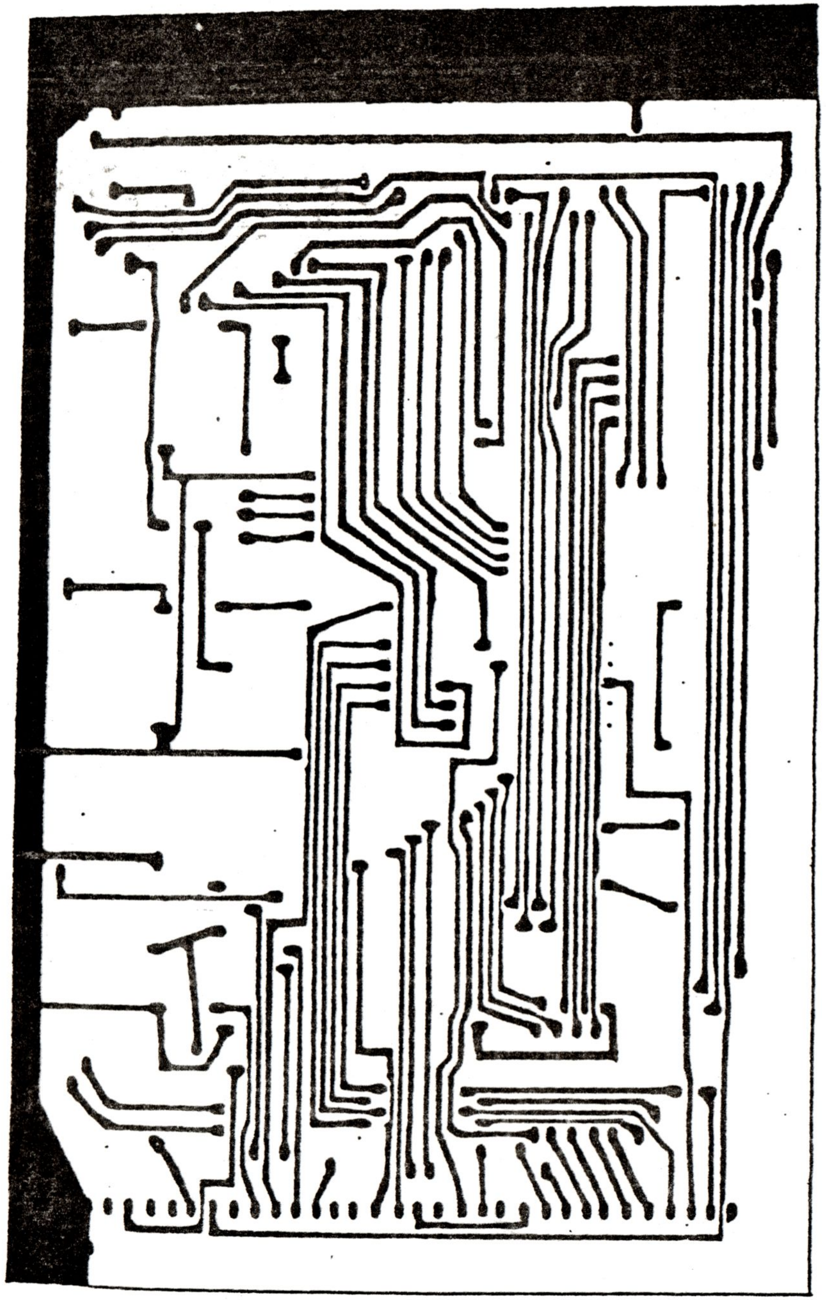
(Nach Erscheinen der nächsten Adressenliste kann diese Seite dort eingeklebt werden)

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat	// geschäftlich
Alber	Herbert	Alemannenstraße 20	7732	Niedereschach	07721 /7102	// -
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Wuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781	//
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907	// 0431 /74047
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621	//
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441	Wettringen	05233 /4320	// 02557 /1236
Emmrich	Helmut	Waldstraße 5	6682	Ottweiler	06824 /4114	// -
Frink	Thomas	An der Schleifmühle 2	5042	Erftstadt	02235 /76255	//
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573	//
Hartel	Eberhard	Neenstetter Straße 20	7901	Breitingen	07340 /7281	//
Heile	Heinz-Dieter	Blankensteiner Straße 13	4320	Hattingen	02324 /28458	//
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024	//
Hill	Peter	Eckstraße 36	6750	Kaiserslautern 31	0631 /54782	//
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289	//
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948 /542	// 02921 /70431
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630	// 089 /522071
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494	// -
Kopschina	Peter	Strandallee 138	2409	Scharbeutz	-	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417	// -
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /668151
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083	// 04103 /702662
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913	// 06124 /2051215
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /3791	//
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551	// 04103 /7025310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383	// 0203 /331383
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221 /707044	//
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631	// 08025 /41247
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284	//
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434 /454	// 08431 /7041604
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585	// 05401 /90036
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883	// 0231 /593883
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497	// 02362 /23127
Troesch	Eberhard	Altenessener Str. 414	4300	Essen 12	0201 /342324	//
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285	//
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361	// -
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194	//
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /29145	// -
Zwickel	Walter	Lengfelden 123	A- 5101	Bergheim	0043662/51130	// -

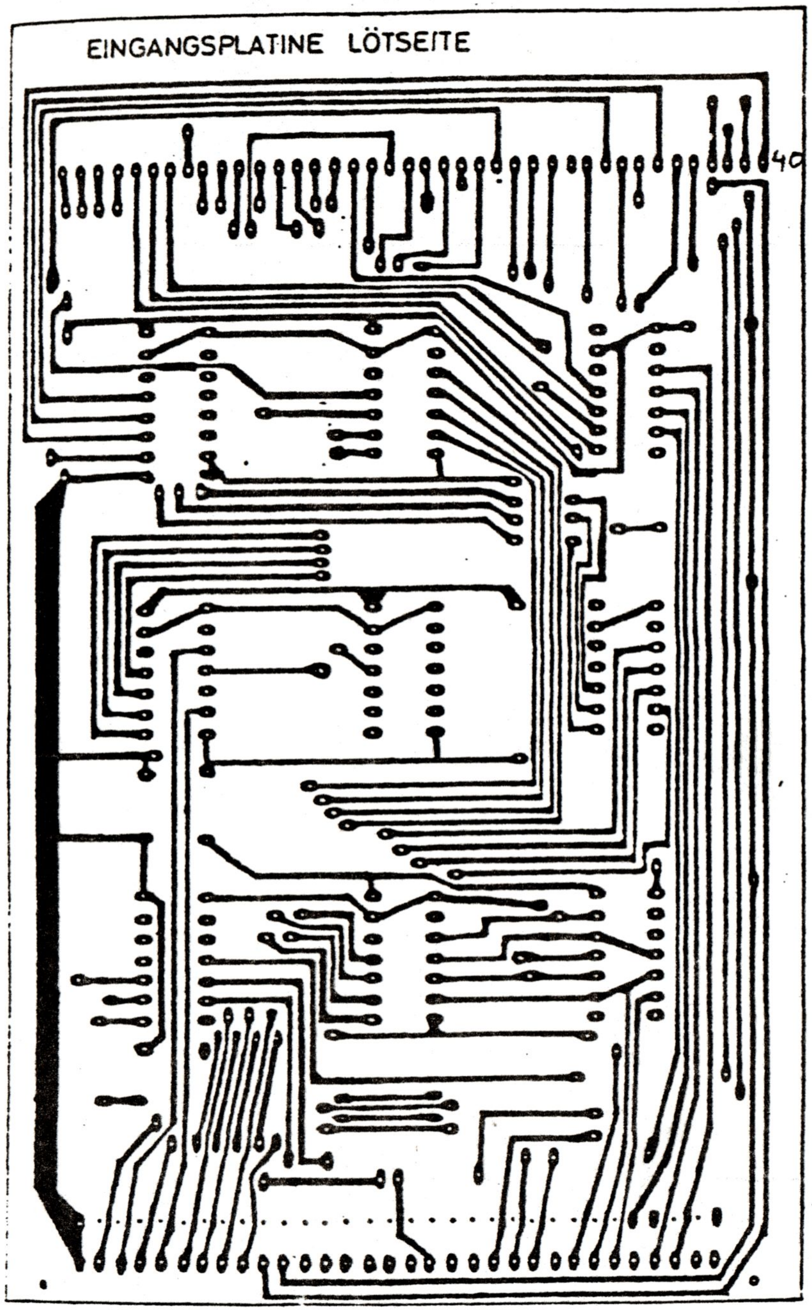
Stand: September 1986

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion



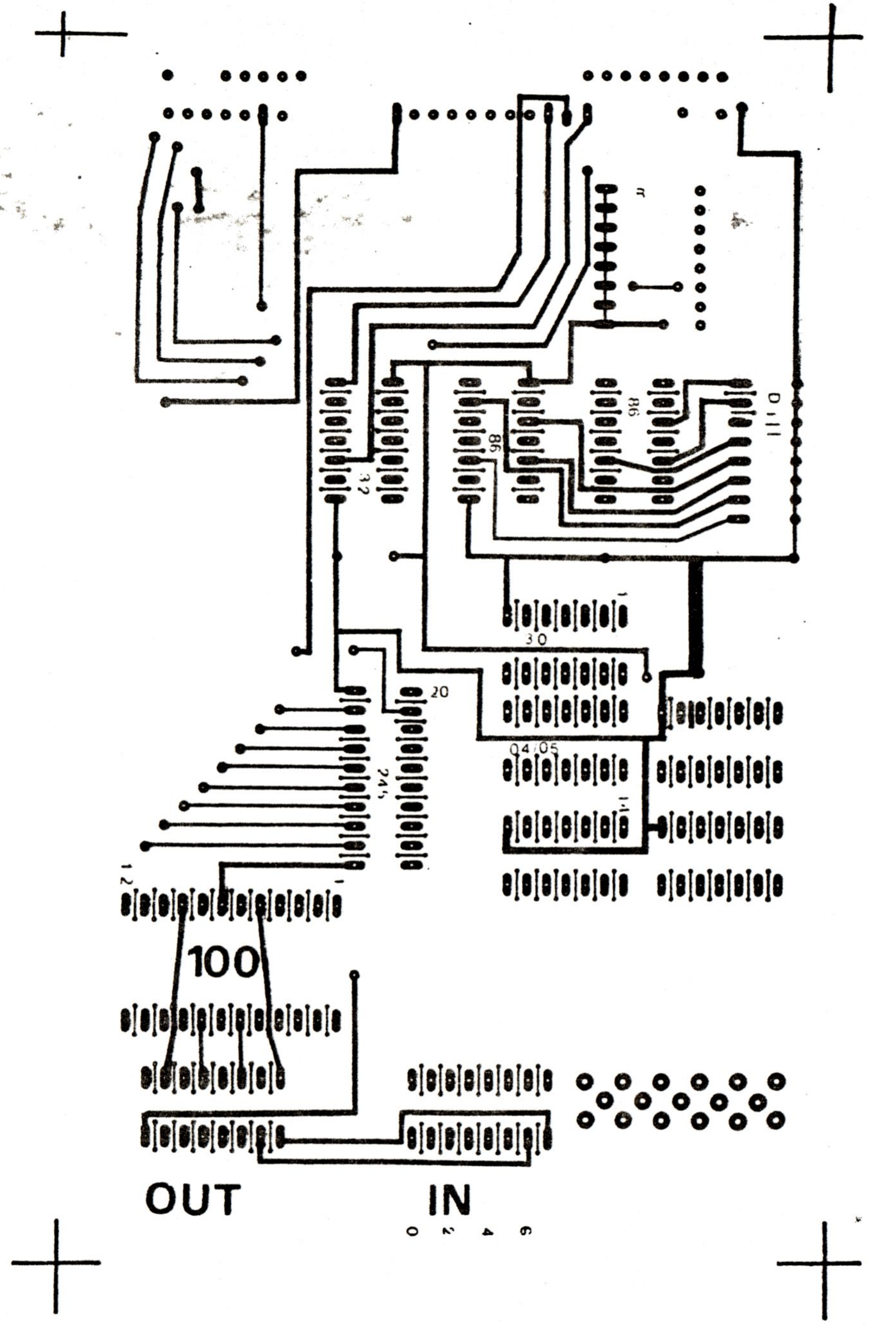
Platinenlayout Eingangskarte

Bestückungsseite



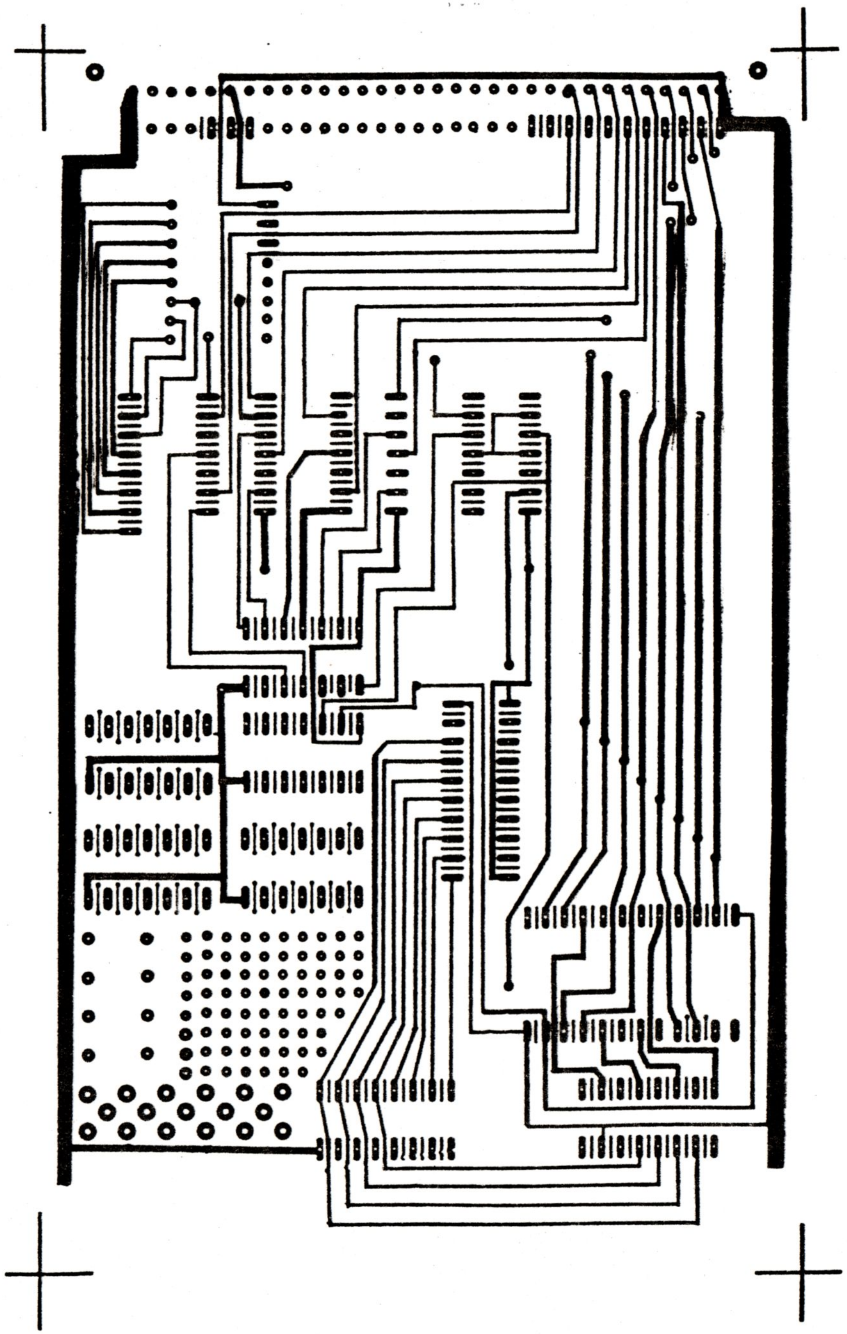
Platinenlayout Eingangskarte

Lötseite



Platinenlayout (ZK01)

Bestückungsseite

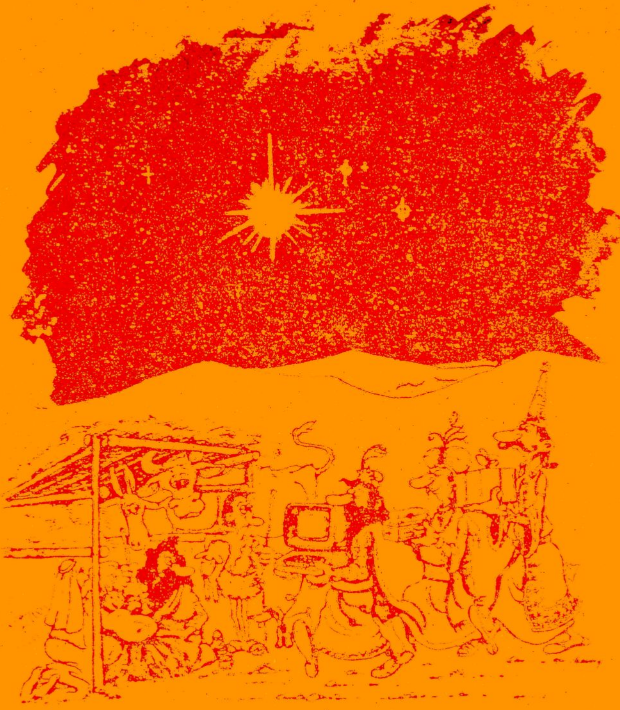


Platinenlayout (ZK01)

Lötseite

CLUB 80

CLUBINFO DER TANDY - GENIE - UND KOMTEK - ANWENDER



Frohes Fest
Frohes Fest 1986
Frohes Fest

16. AUSGABE

KONTAKTADRESSE : CLUB 80 / PETER STEVENS / POSTFACH 56 / 4600 DORTMUND 1
TEL.: 0231 / 593883

Inhaltsverzeichnis

Seite:

Clubinternes

Neues vom Vorstand	01 - 02
Mitgliedsbeitrag	03
Hobbytronik-Treff	03
In eigener Sache	04
Meinung: Wettbewerb	05
Termine / Messen	06

Software

Programmierwettbewerb	07
Logische Operationen 2	09 - 10
Logiksimulation mit Basic	11 - 13
Logisches zu Logik	14 - 17
BASIC- Laufende Programme erweitern	18
Spline Interpolation (kubisch)	19 - 20
Goldener FIBONACCI	21 - 23
Steno-Logik	24 - 26
Graphik-JKL, Tscrips und NEC P6	27 - 28
Telefon, ein UD-Zeichen	29 - 31
Syskopy1 und Ramdisk	32
Mein 3. Assemblerprogramm	33 - 36
RPNM-Programmiersprache	37 - 40

Hardware

Hardwaretip	41 - 44
Taktumschaltung (allerletzte Lösung)	45 - 46

Seite:

Börse

Wer hat was ???--Wer sucht was ???	47 - 48
Wer kann weiterhelfen	47 - 48

Sonstiges

Der Kniff mit der Logik	06
Am Anfang war ein Ungetüm	06
TANDY - better again	40
Drucker	44
Software zum Nulltarif	48
Bits übers Radio	48

Programmbibliothek

Neues von der DISKA-Front	49 - 53
Diskothekar gesucht	53 - 54

Die letzten Seiten

Impressum	55
Schluß	56
Clubmitgliederadressen	am INFO-Ende
Sonderteil DIGITAL	am INFO-Ende
Sonderteil Gills	am INFO-Ende

Neues vom Vorstand

Vorstandssitzung

Am Sonntag den 16.11.86 13:30 Uhr fand in einem Hotel in Höhr-Grenzhausen eine Sitzung des Vorstands statt. über diese Sitzung will ich euch so ausführlich wie möglich berichten.

Der Vorstand war, bis auf K.J. Mühlenbein, vollzählig. Es nahmen also Teil:

1. Vorsitzender — Peter Stevens
2. stellv. Vorsitzender — Hartmut Obermann
3. Hardwarekoordinator — Eckehard Kuhn
4. Clubinforedakteur — Jens Neueder

Die angesetzten Themen waren:

1. übergabe der Geschäftsführung des **CLUB 80** vom ersten an den zweiten Vorsitzenden.
2. Eintragung des **CLUB 80** als e.V.
3. Planung des jährlichen Clubtreffens
4. Weitere Clubaktionen

Zu 1.: Da unser 1. Vorsitzender Peter Stevens durch sein ständig expandierendes Geschäft (er ist selbstständig) keine Zeit mehr für die korrekte Abwicklung der Geschäftsführung des **CLUB 80** hat, habe ich diese übernommen. Hauptsächlich handelt es sich dabei um die Betreuung der Mitglieder, Aufnahme der Neuzugänge, Verwaltung der Clubbücherei und last not least das Führen der Clubkasse.
Die Besetzung des Vorstandes an sich bleibt von dieser Umverteilung der Arbeit unberührt. Unser 1. Vorsitzender bleibt also weiterhin Peter Stevens!

Zu 2.: Inzwischen (es hat ja lange genug gedauert) wurde unsere Satzung vom Amtsgericht als rechtlich einwandfrei anerkannt. Danach könnte nun also die eigentliche Eintragung als e.V. erfolgen. Der Vorstand hat sich jedoch dafür entschieden, mit der Einschreibung noch bis zum nächsten Clubtreffen zu warten, um dann den Club sofort mit dem neuen Vorstand eintragen zu lassen. Ansonsten würde schon Ende März (siehe auch 3.) ein weiterer kostenplichtiger Gang zum Amtsgericht notwendig!

Zu 3.: Das nächste Clubtreffen des **CLUB 80** findet (wenn nichts außergewöhnliches dazwischenkommt) Mitte März statt. Der genaue Termin lautet 13.-15.03.1987! Auf diesen relativ frühen Termin hat sich der Vorstand geeinigt, um nicht mit dem Regionaltreffen "nord" (angeregt und organisiert durch Hans Martin Stephan, Termin: 25./26.04.87) zu konkurieren. Schließlich soll die Jahreshauptversammlung nicht zu einem Regionaltreffen "süd/mitte" werden, sondern alle Mitglieder ansprechen! Das Treffen wird diesmal etwas weiter nördlich, nämlich im Bereich zwischen Fulda und Kassel (ABK Hattenbach) stattfinden. Wir hoffen damit allen Mitgliederwünschen bezüglich Anfahrtstrecke und Entfernung gerecht zu werden. Der genaue Ort des Treffens steht noch nicht fest, vielmehr habe ich Jürgen Wucherer, der uns schon beim ersten Treffen gut beraten hat, auf die Suche nach einem passenden Hotel geschickt. Im nächsten Info, das Ende Januar bei euch eintrudeln wird, wird dann der Austragungsort bekanntgegeben und ein Anmeldezettel beigelegt von denen hoffentlich recht viele mit positivem Bescheid wieder bei mir landen.
Merkt euch also das Datum 13.-15.03.1987 unbedingt vor!!

Zu 4.: Unter Punkt 4 wurden mehrere Unternehmungen besprochen, die der Club und hier vor allem der Vorstand in nächster Zeit durchführen wird.

I. mit dem nächsten Heft zusammen wird ein Gesamtinhaltsverzeichnis der bisher erschienenen Infos herauskommen.

II. Durch unser Mitglied Gerald Dreyer soll versucht werden, Kontakt zu einem TRS 80-Club in Holland aufzunehmen. Dabei wird vor allem an einen Austausch der Infos und Programmbibliotheken gedacht. Dabei tritt natürlich das Sprachproblem störend in Erscheinung. Sollte jemand die holländische Sprache verstehen, wäre ich dankbar, wenn er sich bei mir meldet.
Auch zu einem Club in den USA wollen wir in Verbindung treten.

III. Für unseren Zehnzeiler-Wettbewerb wurden die Preise festgesetzt. Dem Gewinner des ersten Preises wird für das Jahr 1987 der Jahresbeitrag erlassen! Immerhin nur den halben Jahresbeitrag braucht der Gewinner des zweiten Preises zu berappen. Und dem Dritten im Bunde winken als Preis zehn Disketten.
Vielleicht wird ja durch diese Preise das Interesse und die Teilnahmefreudigkeit noch etwas angeregt.
Einen Einsendeschluß gibt es noch nicht, jeder kann noch mitmachen und hat noch die Chance zu gewinnen!

Soweit also zur Sitzung des Vorstands. Ich hoffe, alle Mitglieder sind mit den Entscheidungen, welche auf dieser Sitzung gefällt wurden einverstanden. Natürlich ist der Vorstand immer offen für Kritik und Anregungen und so würde ich mich freuen, wenn aufgrund dieses Sitzungsberichts Reaktionen (negative oder positive) eurerseits bei mir eintrudeln!

Weihnachts- und Neujahrswünsche 1987!!!

Kaum zu glauben, daß das Jahr 1986 sich schon so rapide seinem Ende zuneigt! Mir kommt es immer noch so vor, als wäre das Clubtreffen von ein paar Wochen gewesen, da steht schon das nächste ins Haus. Sechs Rieseninfos hat der **CLUB 80** in diesem Jahr auf die Beine gestellt und dabei ist die Qualität nicht etwa gesunken sondern noch kräftig gestiegen. Das recht langsam angelaufende Hardwareprojekt hat in diesem Jahr nun doch erhebliche Fortschritte gemacht und auch die Softwarebibliothek kann sich sehen lassen!

Ihr seht, ich versuche recht verzweifelt ein Fazit, einen Schlußstrich zu ziehen hinter dieses Jahr. Warum es mir nur schlecht gelingt liegt wohl daran, daß unter eine noch laufende, sich ständig in Bewegung befindende Sache nur schwer ein Strich zu ziehen ist. Wie soll ich dieses Jahr abschließen, wenn ich schon an die Antivitäten im nächsten denke!?

Ich kann eigentlich nur hoffen, daß sich der Club weiter so entwickelt, wie er es bisher getan hat. Dann brauche ich keinen Nachruf auf 1986 zu schreiben, sondern kann getrost allen Mitgliedern ein frohes Weihnachtsfest und ein glückliches Neues Jahr 1987 wünschen, euer

Hartmut Obermann

P.S.: Besonders herzlich fällt dieser Gruß natürlich an all jene aus, die sich aktiv an der Gestaltung des Clublebens, egal in welcher Form auch immer, beteiligt haben und hoffentlich auch 1987 beteiligen werden!

Mitgliedsbeitrag 1987

Eine der Aufgaben des geschäftsführenden Vorstands ist das Führen der Clubkasse. Dazu gehört natürlich nicht nur das Ausgeben des Geldes innerhalb des Geschäftsjahres, sondern auch das Eintreiben der Mitgliedsbeiträge. Von meinem Vorgänger Günther Wagner weiß ich, daß dies sogar eine der zeitaufwendigsten Beschäftigungen ist, die einem Vorstand aufgehalst werden. Aus diesem Grunde wurde durch den Vorstand während seiner letzten Sitzung folgende Regelung getroffen:

Absolut letzter Termin
für die Einzahlung des Beitrages ist der
15. Januar 1987!

"Absolut letzter Termin" heißt, wer bis dahin seinen fälligen Beitrag nicht bezahlt hat, bekommt das Info Nr. 17, welches Ende Januar erscheint, nicht mehr! Durch diese Maßnahme hofft der Vorstand längeren Mahnaktionen und einem Verlust durch das Versenden von Infos an ohne Abmeldung ausgetretene Mitglieder vorzubeugen.

Ich hoffe ihr versteht und unterstützt diese Maßnahme, euer

Hartmut Obermann

Kontonummer weiterhin: Postgirokonto Peter Stevens
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

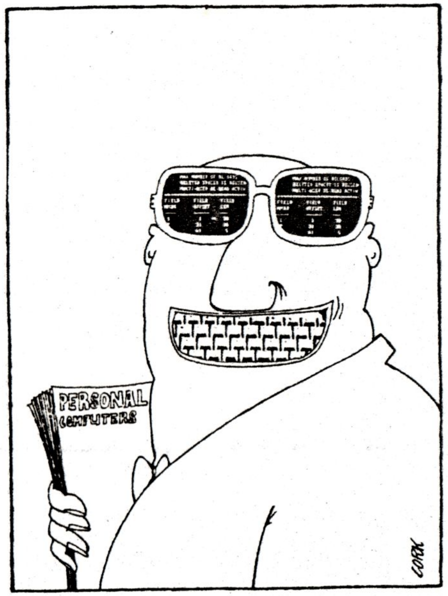
In eigener Sache

Fur das Kommende INFO zum Jahreswechsel möchte ich eine Kleine Themenvorschau geben:

1-- Jahresinhaltsverzeichnis aller bisherigen INFO's
2-- Komplette Liste der Mitglieder
mit deren HardwareKonfiguration
(wer dazu noch Änderungen bekanntzugeben hat
sollte dies bitte umgehend tun)
3-- Programm-Katalog unserer DisKothek

Punkt 2 hatte ich Euch ja schon länger versprochen. Leider bin ich zwegs Zeitmangel nicht mehr dazu gekommen. Auch sind die Daten noch nicht von allen Mitgliedern vorhanden. Deshalb kann ich auch nur eine unvollständige Liste bieten.

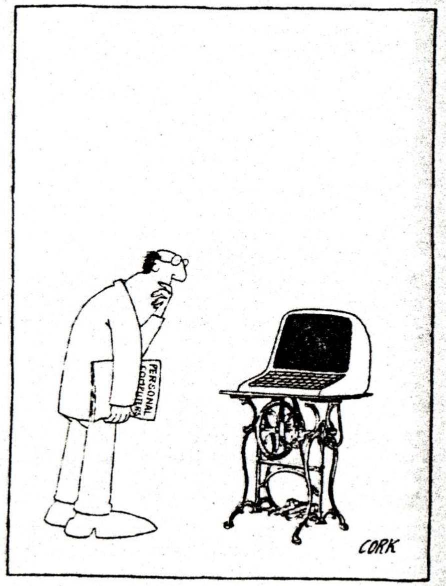
Die RedaKtion



Hobbytronik-Treff

Wie im letzten Info unter "TERMINE" angekündigt, trafen sich am 19.10. auf der Hobbytronik in Stuttgart Mitglieder des CLUB 80. Dieses Treffen war nicht weiter organisiert, sondern jeder der Lust, Zeit und Geld (letzteres war nicht unbedingt Bedingung, ich bin schließlich auch hingefahren und habe trotzdem kein Geld) hatte konnte sich einfinden. Tatsächlich zusammengefunden haben sich insgesamt 10 Leute (8 Mitglieder + 2 bessere Hälften). Es waren anwesend: Bernd Drohwälder, Ulrich Böckling, Klaus Hermann, Eckehart Kuhn, Jens Neueder mit Freundin Sabine, Richard Rensch mit Sohn und last not least meine Frau mit ihrem ständigen Begleiter.

Ich finde solche lockeren Treffs immer wieder schön und anregend. Einerseits lernt man dabei Mitglieder kennen, die man wohl sonst nie zu Gesicht bekommen hätte, andererseits trifft man immer wieder Freunde mit gleichen Interessen (z.B. war die Gruppe Böckling/Neueder/Obermann schon mehrfach zusammen auf der Hobbytronik, sowohl in Stuttgart als auch in Dortmund) und hat so Gelegenheit zum Meinungsaustausch. Ich würde mich freuen, wenn sich noch mehr Mitglieder bei solchen Kurztreffs sehen ließen und hoffe auch **Dich** beim nächstenmal zu treffen! Euer Hartmut Obermann



Gorbatschow fordert Reagan auf: »Fragen Sie mal Ihren Computer, was mit Amerika im Jahre 2000 sein wird!« Sekunden später kommt die Antwort: »Sozialistisch.« Als Gorbatschow seinen Computer nach der Zukunft der Sowjetunion befragt, sieht er verwirrt auf den Bildschirm und meint: »Kann ich nicht lesen – das ist Chinesisch!«

»Können Sie mir helfen? Ich suche ein Geschenk für meinen Onkel – er wird morgen 64 Jahre alt.« »Da haben Sie aber Glück! Ich habe hier zufällig einen Schachcomputer mit genau 64 Feldern.«

MEINUNG:

Zum Thema Programm - Wettbewerb...

Der Peter bezeichnete das Preisausschreiben einen 33-Zeiler zu programmieren als Reinfall und es war wohl auch einer. Doch woran mag es gelegen haben, da wir doch einige wahre Spitzenleute in unserem Club beherbergen? (ich gehöre nicht dazu)
Persönlich störte ich mich an der Rahmenbedingung. Denn die Zeilenanzahl ist wohl das letzte worauf beim Programmerstellen geachtet wird!!!! Ganz abgesehen davon, daß die Übersicht enorm darunter leidet. Als Beispiel habe ich nochmals mein Interpolationsprogramm ausgedruckt. Bei der 10-Zeilenform blicke ich jetzt nach einem Monat nämlich selber nicht mehr so ganz durch.

Aus diesem Grunde will ich vorschlagen, daß in Zukunft doch besser ein konkretes Problem vorgelegt wird, wobei sinnvolle Nebenbedingungen... hohe Geschindigkeit
geringer Speicherbedarf
anwenderfreundlich
(zum Beispiel) im Vordergrund stehen. Übersichtlichkeit und eine Minidokumentation im Programm sollten ja selbstverständlich sein. Die Sprache (Assembler, Basic, C, Cobol, Forth, Fortran, Pascal...) in der die Lösung geschrieben steht, möge lieber freigestellt sein. Es ist bestimmt aufschlußreich, wie umstandlich / listig die eine oder andere Aufgabe in den verschiedenen Sprachen aussieht.
Ich würde mich sehr freuen wenn das nächste Problem in dieser Form gestellt wäre!

Wie denkt ihr darüber ?

Dieter Kasper

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....28. Dezember 1986
Mitgliedsbeitrag 1987 (letzter Termin) 15.Januar 1987
Jahreshauptversammlung 1987 ....13. - 15. März 1987
Norddeutsches Regionaltreffen ....25. + 26. April 1987

-- Messen '86 --

Hobby-tronik ....Dortmund 18. - 22. Februar 1987
CeBIT ....Hannover 4. - 11. März 1987
Internationale Computer-Ausstellung ..Köln .... 11. - 14. Juni 1987
HOBBY ELEKTRONIK 87 ....Stuttgart 5. - 8. November 1987
PRODUCTRONICA ....München 10. - 14. November 1987

BASIC-Rätsel

Der Kniff mit der Logik

Wer möglichst kurze und möglichst schnelle Programme schreiben möchte, kommt um logische Verknüpfungen nicht herum. Vor allem, wenn es darum geht, mehrere Wenn-Dann-Bedingungen (IF – THEN) zusammenzufassen. sind entsprechende Verknüpfungen mit AND und OR ratsam. Dabei muß man allerdings darauf achten, wie unterschiedlich sich diese logischen Operanden verhalten. Denn bei AND wird die Anweisung hinter THEN nur dann ausgeführt, wenn alle mit AND verbundenen Bedingungen hinter IF erfüllt sind. Anders bei OR: Es reicht, wenn eine der Bedingungen hinter IF erfüllt ist, und schon wird die Anweisung hinter THEN ausgeführt. Im Beispiel geht es darum, ob ein Ergebnis ausgedruckt werden soll. Der Computer fragt endlos, egal, wie die Antwort lautet. Warum?

```
500 REM DRUCKER-
    ABFRAGE
510 PRINT "AUSGABE AUF
    DRUCKER (J/N)?"
520 GET TASTE$ : IF
    TASTE$="" THEN 520
530 IF TASTE$<>"J"
    OR TASTE$<>"N"
    THEN 510
540 IF TASTE$="J" THEN
    GOSUB 1000
550 END
1000 PRINT »DRUCK-
     ROUTINE«
```

Auflösung auf S. 111

Computergeschichte(n)

Am Anfang war ein Ungetüm

Die Zuhörer, die sich in einem kleinen Hörsaal der Technischen Universität Berlin 1938 versammelt hatten, glaubten ihren Ohren nicht zu trauen. Da trugen zwei ältere Semester, Konrad Zuse und sein Freund Helmut Schreyer, nach der Vorführung von Versuchsschaltungen den Gedanken einer elektronischen Rechenmaschine vor. Was die gesetzten Herren Professoren fast noch mehr verwirrte, war die Unbekümmertheit, mit der die zwei ihre Wünsche an Bauteilen auflisteten. Mit Kopfschütteln wurde der Bedarf an zweitausend Röhren und einigen tausend Glimmlampen quittiert. »Reine Phantasterei – selbst die modernsten elektronischen Sendeanlagen haben nicht mehr als einige hundert Röhren. Und dann allein der Stromverbrauch von so einem Ungetüm!«
Aber der eigenwillige Erfinder, der seine ersten Versuchsmodelle im ausgeräumten elterlichen Wohnzimmer baute, gab nicht auf. 1940 war Z 2 fertig: ein elektronischer Rechner mit einer Speicherkapazität von 64 Wörtern, programmgesteuert über eine Lochstreifeneingabe.
Das war so revolutionär, daß Zuse sogleich einen Auftrag bekam. Für die Deutsche Versuchsanstalt für Luftfahrt baute er seinen dritten Rechner, den Z 3, der heute als erster echter Computer im Deutschen Museum steht. Nur die Luftfahrttechniker mit ihren aufwendigen und komplizierten Berechnungen hatten das Zukunftsweisende an Zuses Computern erkannt. Hätte er damals mehr Unterstützung erhalten, würden wir heute vielleicht den Computer nicht als amerikanische Erfindung, sondern als deutsche empfinden.

Der deutsche Computerpionier Konrad Zuse (hinten sein Z3).

Ehret die alten Römer!

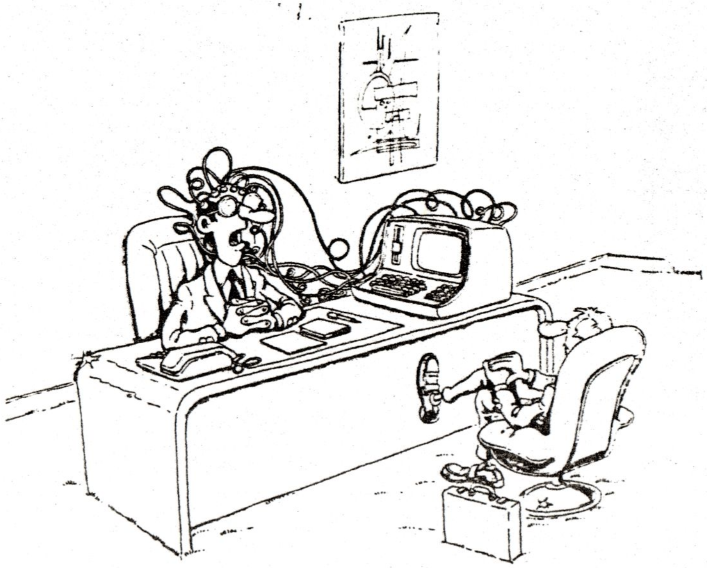
Keiner weiß heute mehr, wozu die römischen Zahlen gut sind. Ich auch nicht.
Vielleicht, um den Bildungsgrad von Touristen zu prüfen, die vor ehrwürdigen Bauwerken stehen und angesichts römischer Inschriften über deren Entstehungszeit nachgrübeln...
0123456789 - wer kennt die arabischen Ziffern nicht?
4711 - wer nicht die aus ihnen gebildeten Zahlen?
Aber wer kennt noch: MMMMDCCXI ?? (Unter uns: Es ist dasselbe!)
Willkommen wäre also ein Programm, das aus römischen Zahlen arabische zaubert.
Aber das wäre zu einfach! Außerdem wäre das so, als schenkte man uns ein Wörterbuch "Chinesisch/Deutsch", wenn wir nach China reisen - es würde uns nichts helfen, solange wir nicht Chinesisch können! Wir brauchen zuerst ein Wörterbuch "Deutsch/Chinesisch"!

Also fangen wir mit der Übersetzung der arabischen Zahlen in die römischen an. Wer Lust und Mut hat, kann danach das "Anti-Programm" selbst bauen: zur Übersetzung vom Römischen ins Arabische! (Was natürlich viel leichter ist...)

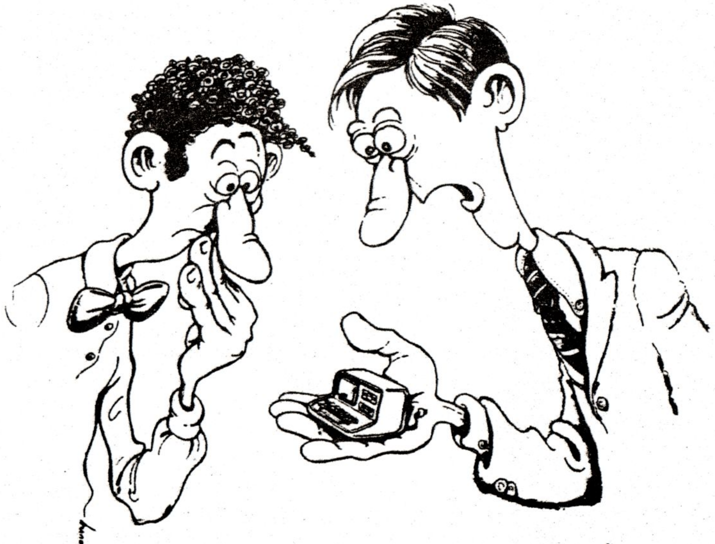
Tschüßchen! * KaJot *

Schon wieder ein "Zehnzeiler"!...

```
1 CLS:PRINT"   Umwandlung von arabischen in roemische Zahlen
   ==================================================
 "'              <C> K.J.Muehlenbein, Weinheim 1986
2 DEFINTE,H,N,T,Z:DEFSTRC,D,I,L,M,R,X,V:I="I":V="V":X="X":L=
"L":C="C":D="D":M="M":PRINT"Gib eine ganze Zahl <10000 ein!
 ":INPUTZO:T=INT(ZO/1000):IFT>0:FORN=1TOT:R=R+M:NEXT
3 Z1=ZO-1000*T:H=INT(Z1/100):IFH>0:IFH>4GOSUB7ELSEFORN=1TOH:
R=R+C:NEXT
4 Z2=Z1-100*H:Z=INT(Z2/10):IFZ>0:IFZ>3GOSUB8ELSEFORN=1TOZ:R=
R+X:NEXT
5 E=Z2-10*Z:IFE>0:IFE>3GOSUB9ELSEFORN=1TOE:R=R+I:NEXT
6 GOTO10
7 IFH=5R=R+D:RETURN:ELSEIFH<9:R=R+D:FORN=1TOH-5:R=R+C:NEXT:R
ETURN:ELSER=R+C+M:RETURN
8 IFZ=4R=R+X+L:RETURN:ELSER=R+L:IFZ>5FORN=1TOZ-5:R=R+X:NEXT:
RETURNELSERETURN
9 IFE=4R=R+I+V:RETURN:ELSEIFE<9R=R+V:IFE=5RETURNELSEFORN=1TO
E-5:R=R+I:NEXT:RETURN:ELSER=R+I+X:RETURN
10 PRINTR:PRINT"ALLES KLAR? * ALLES KLAR!":END
```

WIR HABEN EIN SEHR GUTES BETRIEBSKLIMA



Nun müssen wir uns nur noch selbst verkleinern!

Logische Operationen in BASIC 2

Liebe Clubfreunde,

leider mangelt es mir zur Zeit an der nötigen Muße um die im ersten Teil dieses Beitrages angekündigte Fortsetzung zu verfassen. Aber aufgeschoben ist ja nicht aufgehoben, wie schon das alte Sprichwort sagt und so verspreche ich euch die Fortsetzung für eines der nächsten Infos.

Aber wenigstens die sechs fleißigen und schnellen Einsender von Lösungen zu meinem kleinen Preisrätsel sollen nicht länger auf die Folter gespannt werden und endlich erfahren, wer denn nun die Flasche mit dem garantiert glycolfreien Wein gewonnen hat.

Eingegangen sind insgesamt sechs Lösungen. Teilgenommen haben: Klaus Hermann, Eckehard Kuhn, K.J. Mühlenbein, Kurt Müller, Bernd Retzlaff und Arnulf Sopp. Alle eingesandten Lösungen waren richtig.

Die vollständigsten (aber das war ja nicht gefordert) Lösungen stammen von Kajott und Bernd Retzlaff. Beide verschönten ihre richtigen Lösungen noch mit einem entsprechendem Demoprogramm, wobei Bernd immerhin einen drei Seiten langen Brief zusammenbrachte, während Kajott das Ganze auf eine normale Postkarte quetschte (allein das ist schon anerkennenswert!).
Hier die Lösungszeilen:

```
C=(A OR B) AND (NOT(A AND B))                  Kajott
IF (A OR B) AND NOT (A AND B) THEN C=1 ELSE C=0   Bernd
```

Wie ihr seht, kamen bei gleichem Lösungsweg zwei unterschiedliche BASIC-Zeilen heraus, die jedoch beide ihre Plicht erfüllen!

Auch Kurt Müller packte seine Lösungszeile in ein Programm und garnierte das Ganze noch zusätzlich mit einer Herleitung der Lösung mittels der Boolschen Algebra und einem Karnoaugh-Diagramm (leider habe ich heute keine Zeit euch in die Feinheiten und Geheimnisse dieses Algebraablegers einzuführen, aber vielleicht läßt sich Kurt ja dazu breitschlagen!?). Seine Lösungszeile lautet: C=(NOT(A) AND B) OR (A AND NOT (B)).

Zu genau der gleichen Lösung kam Eckehard Kuhn, der allerdings außer der Lösung (und natürlich dem Absender und Empfänger) keine weiteren Worte auf der von ihm benutzten Postkarte verlor (was ja auch nicht gefordert war!).

Gleich zwei Lösungen sandte Klaus Hermann ein und beide funktionierten. Seine Lösungszeilen lauten:

```
1. C=(A AND NOT B) OR (NOT A AND B)
2. C=NOT(NOT A OR B) OR NOT(A OR NOT B)
```

Was wieder mal beweist, nicht nur nach Rom führen viele Wege!

Die wohl kürzeste Lösung stammt von unserem Assemblerfreak Arnulf Sopp. Er verwandte neben den logischen Operationen auch eine mathematische und so lautet seine BASIC-Zeile:

```
C=(A OR B) - (A AND B)
```

Ich wollte zuerst auch nicht glauben, daß das funktionieren kann. Aber nachdem ich die Zeile flux in das Programm von Kajott eingesetzt hatte, stellte sich die Funktionsfähigkeit sofort heraus!

Sechs Einsendungen und sechs richtige Lösungen (es gibt übrigens noch mehr, so erfüllt die Zeile "C=(A+B) AND 1" auch ihren Zweck) machen mir natürlich die Entscheidung schwer, wer nun den edlen Tropfen bekommen soll. Noch dazu, da alle Lösungen fast gleichzeitig bei mir eintrafen! Aus diesem Grunde habe ich mich zu der, so hoffe ich gerechtesten Entscheidung durchgerungen, die als Preis ausgesetzte Flasche selbst zu leeren! Prost!!!

Damit sich die Teilnehmer aber nicht umsonst so viel Mühe gegeben haben, werde ich zusätzlich jedem der Sechs eine Flasche zukommen lassen und hoffe, daß sie sie auf das Wohl des CLUB 80 trinken werden. Also nochmal Prost!!!

Damit auch die restlichen Leser dieser kleinen Artikelserie, die sich vielleicht schon auf den zweiten Teil gefreut haben, auch nicht zu kurz kommen, folgt hier nun ein Artiel aus der letzten "mc", der sich mit einem ähnlichen Problem beschäftigt. Viel Spaß bei der Lektüre wünscht euch

Hartmut Obermann

Fast hätte ich was vergessen! Leider ist mir im ersten Teil ein kleiner Tipfehler unterlaufen, den ich trotz mehrfacher Kontrolle übersehen habe. Die Wahrheitstafel der AND-Verknüpfung enthält in Zeile vier einen Fehler! Die korrigierte Version sieht so aus:

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Meinung zur Meinung von Dieter Kasper Seite 5:

Die Rahmenbindung meines Programm-Wettbewerbes auf max. 33 Zeilen geschah aus der Sicht des Auswertenden. In der Hoffnung auf viele Teilnehmer wollte ich vermeiden, einen Stapel ellenlanger Listings auf sinnvolles und ideenreiches Programmieren zu durchsuchen und daraus eine Rangfolge zu bestimmen. BASIC war vorgegeben; was nützt mir das schönste Programm in C, Cobol oder Forth ? Ich bin dieser Sprachen nicht mächtig (solls ja auch noch geben). Vielleicht lag es an der (Un-)Lust der Mitglieder ? Die Preise waren doch wohl ok ! Ein bischen Rahmenbindung muß bei einem Wettbewerb wohl sein.
Sei's drum, es hat trotzdem Spass gemacht. (BitPit)

Kurt Peter

Logiksimulation mit Basic

Bei jeder Umsetzung logischer Verknüpfungen in eine digitale Schaltung stellt sich die Frage, ob das, was auf dem Papier steht, noch Ähnlichkeit mit der Zielsetzung hat. Wie man ohne großes Softwarepaket das Erstellen einer Wahrheitstafel einem beliebigen Basic-Rechner überträgt, zeigt der folgende Beitrag.

Das Problem ist jedem, der sich mit digitaler Hardware einläßt, sicher vertraut: Man hat sich aus den Karnaugh-Diagrammen oder den logischen Gleichungen eine Schaltung aufgezeichnet, eventuell noch optimiert und Laufzeiten angepaßt, so daß die Gatteranordnung nicht mehr viel mit dem ursprünglichen Entwurf gemeinsam hat. Auf den ersten und oft auch auf den zweiten, etwas gründlicheren Blick läßt sich nicht mehr sagen, ob sich bei all den (natürlich sorgfältig ausgeführten) Veränderungen nicht doch ein Fehler eingeschlichen hat. Besonders undurchsichtig wird das Ganze, wenn man zur Minimierung der Bausteinzahl noch sogenannte „Punktverschiebungen" durchgeführt hat, um alles einheitlich auf NANDs und NORs zu reduzieren – spätestens jetzt ist jede Übersicht verloren. Die Versuchung ist groß, die eigene Fehlerquote bei 0% anzusiedeln und sich an den Aufbau zu wagen. Ist etwas mehr Mut vorhanden und hat man die Tabelle doch aufgestellt, so kommt oft bei mehrmaligem Überprüfen jedesmal eine andere Tabelle heraus, denn auch das Erstellen von Wahrheitstabellen ist eine fehlerträchtige Beschäftigung – und damit eigentlich eine ideale Aufgabe für einen Computer.

Die Basic-Befehle

Schaut man sich den Befehlssatz der handelsüblichen Basic-Interpreter an, so fallen unter der Rubrik „Operatoren" die Befehle NOT, AND, OR und EXOR auf. Sie werden meist zur Verknüpfung mehrerer Bedingungen in IF-Ausdrücken verwendet, eignen sich jedoch auch vorzüglich zum logischen Rechnen. Dazu muß man nur wissen, daß die logische 1 als -1 dargestellt wird; die Null bleibt, wie sie ist.

Eine Wertetabelle stellt die Ausgabewerte für alle möglichen Kombinationen der Eingangssignale dar. Das Durchspielen aller Möglichkeiten läßt sich in jeder höheren Programmiersprache durch das Verschachteln von Schleifen erzielen. Und da Basic beliebige Schrittweiten – also auch negative – zuläßt, ist das Problem eigentlich schon gelöst.

Bild 1 zeigt, wie man sich die Wertetabelle eines NAND-Gatters (*Bild 2*) ausdrucken lassen kann. Die beiden FOR-Schleifen bewirken, daß die Eingänge A und B alle möglichen Kombinationen aus Null und Eins durchlaufen. In der inneren Schleife erfolgt die Verknüpfung und das Ausdrucken der Tabelle. Die Funktion ABS verhindert, daß vor eine logische Eins ein Minus geschrieben wird, denn wie bereits erwähnt rechnet Basic intern mit diesem negativen Wert.

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

Bild 2. Die Wahrheitstabelle des NAND-Gatters

Eine Sonderstellung nimmt unter Umständen die EXOR-Funktion ein. Ist sie ordnungsgemäß implementiert (wie z.B. bei MBasic), so ergibt sich die Wahrheitstabelle von *Bild 4*. Es kann aber auch anders kommen: So ist etwa beim Basic des AIM-65 diese Funktion nicht vorgesehen. Trotzdem meldet der Rechner beim vorliegenden Programm (*Bild 3*) keinen Fehler und gibt als Ergebnis für Z die Werte von B aus! Die Erklärung liegt in der Tatsache, daß bei diesem Basic Leerzeichen als Trenner nicht erforderlich sind. Aus

Z=A XOR B wird so für den Rechner

Z=AX OR B, und da die Variable AX nicht initialisiert wurde und daher auf Null gesetzt wird, ergibt sich konsequenterweise für Z der Wert von B. Es ist also nicht verkehrt, sich von der ordnungsgemäßen Funktion des XOR-Befehls zu überzeugen und sich gegebenenfalls mit der Umschreibung

Z=(NOT(A) AND B) OR (A AND NOT(B)) zu behelfen.

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Bild 4. So sieht die Tabelle bei richtig arbeitendem XOR-Befehl aus

Ein praktisches Beispiel

Bei den bisher aufgeführten Beispielen kommen natürlich die Fähigkeiten eines Basic-Rechners nicht sonderlich zum Tragen. Daher soll als nächstes ein Beispiel aus der Praxis herangezogen werden. Um aus dem Adreßraum von einem KByte, der von einem Chip-Select zur Verfügung gestellt wird, die ersten 64 Byte für einen zusätzlichen Peripheriebaustein auszublenden, entstand die Schaltung nach *Bild 5*. Sie ist umgeformt in Bausteine des Typs NAND und NOR, um die Anzahl der Chips zu reduzieren. Im Programm (*Bild 6*) sieht man, daß die äußerste Schleife die höchstwertigste, die innerste die niederwertigste Leitung enthält. Zusammen mit der Reihenfolge der PRINT-Befehle in der innersten Schleife ergibt sich das gewohnte Bild der Wahrheitstabelle (*Bild 7*). Um die Gleichungen überschaubar zu halten empfiehlt es sich, Zwischenvariable einzuführen und in der Reihe des Signalflusses hinzuschreiben. Fest auf Null oder Eins gelegte Eingänge kann man entweder in Gleichungen mit aufnehmen, oder man reduziert den entsprechenden Term vorher. Beides findet man in diesem Beispiel.

Bild 7 zeigt, daß die Schaltung trotz Transformation noch so arbeitet, wie es geplant war. Der erste 64-Byte-Raum wird ausgeblendet, C1 wird aktiv. Alle folgenden Räume sind für C2 reserviert, solange CS, das den 1 KByte-Raum selektiert, aktiv ist. Wenn CS inaktiv wird, bleiben auch C1 und C2 gesperrt. Auch dieses Beispiel ist natürlich nicht außergewöhnlich kompliziert, doch es bestehen schon einige Fehlermöglichkeiten.

Rückkopplung

Bislang wurden nur Schaltungen betrachtet, die Signale streng geradeaus verknüpfen. Ganz neue Perspektiven eröffnen sich, wenn man asynchrone Signale rückkoppelt. Obwohl die Mehrzahl solcher Fälle vermieden werden kann (und sollte!), läßt es sich nicht immer umgehen. Solche Verknüpfungen verlangen Aufmerksamkeit, denn sie können mehrere unangenehme Eigenschaften besitzen, wie die folgenden Beispiele zeigen. Interessant ist jedoch, daß sich die damit verbundenen Probleme mit dem bisher beschriebenen Simulationsverfahren ebenfalls handhaben und darstellen lassen.

Bild 8 zeigt eine rückgekoppelte Schaltung, *Bild 9* das Simulations-Programm und *Bild 10* die Wertetabelle. Solche Verknüpfungen lassen sich analysieren, indem man die Rückkopplung aufschneidet und dort, wo sie eingespeist wird, einen eigenen Signalnamen vergibt. Dann läßt man die Simulation in der oben beschriebenen Weise ablaufen, jedoch mit vier anstatt mit drei Eingangssignalen. Zur Betrachtung des Verhaltens der Schaltung sind jedoch nur die tatsächlichen Eingänge A, B und C interessant. Sie bilden quasi den Eingangszustand, dessen verschiedene Werte durch die waagrechten Striche getrennt sind.

Wie wird Bild 10 nun ausgewertet? Die Anschlüsse D und G sind natürlich nicht unabhängig, sondern über eine zeitliche Verzögerung verbunden. Besitzen sie in einer Zeile denselben Wert, so ist der Zustand stabil und der Wert des Ausganges G eindeutig. Dies ist zum Beispiel für den Eingangszustand ABC = 001 für beide Wertepaare von D und G gegeben. Allerdings sieht man hier schon die erste Besonderheit, die bei rückgekoppelten Schaltungen auftritt: War der Rückkoppeleingang D vorher auf Eins, so geht auch G auf Eins. War er jedoch auf Null, so nimmt auch G diesen Wert an. Beim Übergang nach ABC = 001 ergibt sich also der Wert von G aus dem vorherigen Wert von G, das Ergebnis ist also von der Vorgeschichte abhängig. Gleiches gilt auch für den Zustand ABC = 011. Alle übrigen Zustände zeigen ein Ver-


```
210 REM
220 LPRINT
230 LPRINT" A   B   Z"
240 LPRINT"---------"
250 : FOR A=0 TO -1 STEP -1
260 :       FOR B=0 TO -1 STEP -1
270 :             Z = NOT(A AND B)
280 :             LPRINT ABS(A); ABS(B);  ABS(Z)
290 :       NEXT B
300 : NEXT A
310 LPRINT
320 LPRINT"Wahrheitstafel von Beispiel 1"
330 LPRINT
340 END
```

Bild 1. Dieses Programm druckt die Wahrheitstabelle eines NAND-Gatters

```
210 LPRINT
220 LPRINT" A   B   Z"
230 LPRINT"--------"
240 : FOR A=0 TO -1 STEP -1
250 :       FOR B=0 TO -1 STEP -1
260 :             Z = A XOR B
270 :             LPRINT ABS(A); ABS(B); ABS(Z)
280 :       NEXT B
290 : NEXT A
300 LPRINT
310 LPRINT "Wahrheitstafel fuer Beispiel 2"
320 LPRINT
330 END
```

Bild 3. Von der richtigen Wirkung des XOR-Befehls sollte man sich mit diesem Programm überzeugen

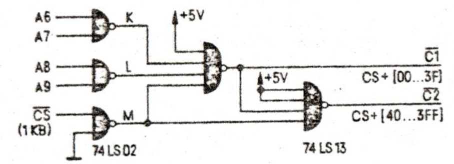


Bild 5. Ein Beispiel aus der Praxis

```
210 REM
220 LPRINT
230 LPRINT"CS A9 A8 A7 A6 C1 C2"
240 LPRINT"--------------------"
250 : FOR CS=0 TO -1 STEP -1
260 :    FOR A9=0 TO -1 STEP -1
270 :       FOR A8=0 TO -1 STEP -1
280 :          FOR A7=0 TO -1 STEP -1
290 :             FOR A6=0 TO -1 STEP -1
300 :                   K  = NOT (A6 OR A7)
310 :                   L  = NOT (A8 OR A9)
320 :                   M  = NOT (CS OR 0)
330 :                   C1 = NOT (K  AND L AND M )
340 :                   C2 = NOT (C1 AND M AND -1 AND -1)
350 :                   LPRINT ABS(CS); ABS(A9); ABS(A8); ABS(A7);
360 :                   LPRINT ABS(A6); ABS(C1); ABS(C2)
370 :             NEXT A6
380 :          NEXT A7
390 :       NEXT A8
400 :    NEXT A9
410 : NEXT CS
420 LPRINT
430 LPRINT"Wahrheitstafel fuer Beispiel 3"
440 LPRINT
450 END
```

Bild 6. Das Programm zur Simulation der Schaltung aus Bild 5

CS	A9	A8	A7	A6	C1	C2
0	0	0	0	0	0	1
0	0	0	0	1	1	0
0	0	0	1	0	1	0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	0	1	0	1	1	0
0	0	1	1	0	1	0
0	0	1	1	1	1	0
0	1	0	0	0	1	0
0	1	0	0	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	1	0
0	1	1	0	1	1	0
0	1	1	1	0	1	0
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	0	0	0	1	1	1
1	0	0	1	0	1	1
1	0	0	1	1	1	1
1	0	1	0	0	1	1
1	0	1	0	1	1	1
1	0	1	1	0	1	1
1	0	1	1	1	1	1
1	1	0	0	0	1	1
1	1	0	0	1	1	1
1	1	0	1	0	1	1
1	1	0	1	1	1	1
1	1	1	0	0	1	1
1	1	1	0	1	1	1
1	1	1	1	0	1	1
1	1	1	1	1	1	1

Bild 7. Die Tabelle zeigt das ordnungsgemäße Arbeiten der Schaltung aus Bild 5

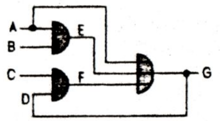


Bild 8. Rückgekoppelte Logikschaltungen wie diese sind auf den ersten Blick immer ein wenig undurchsichtig

```
220 LPRINT
230 LPRINT" A  B  C  D  E  F  G"
240 LPRINT"---------------------
250 : FOR A=0 TO -1 STEP -1
260 :   FOR B=0 TO -1 STEP -1
270 :     FOR C=0 TO -1 STEP -1
280 :       FOR D=0 TO -1 STEP -1
290 :           E = (A AND B)
300 :           F = (C AND D)
310 :           G = (A OR E OR F)
320 :           LPRINT ABS(A); ABS(B); ABS(C); ABS(D);
330 :           LPRINT ABS(E); ABS(F); ABS(G);
340 :           IF G <> D THEN LPRINT"*" : GOTO 360
350 :           IF G =  D THEN LPRINT
360 :       NEXT D
370 :       LPRINT"--------------------"
380 :     NEXT C
390 :   NEXT B
400 : NEXT A
410 LPRINT
420 LPRINT"Wahrheitstafel fuer Beispiel 4"
430 LPRINT
440 END
```

Bild 9. Mit dieser Simulation wird die Schaltung von Bild 8 durchschaubar

A	B	C	D	E	F	G	
0	0	0	0	0	0	0	
0	0	0	1	0	0	0	*
0	0	1	0	0	0	0	
0	0	1	1	0	1	1	
0	1	0	0	0	0	0	
0	1	0	1	0	0	0	*
0	1	1	0	0	0	0	
0	1	1	1	0	1	1	
1	0	0	0	0	0	1	*
1	0	0	1	0	0	1	
1	0	1	0	0	0	1	*
1	0	1	1	0	1	1	
1	1	0	0	1	0	1	*
1	1	0	1	1	0	1	
1	1	1	0	1	0	1	*
1	1	1	1	1	1	1	

Bild 10. Nach kurzem Einschwingen verhält sich die Schaltung von Bild 8 stabil

halten, das nach kurzem Einschwingen in einem eindeutigen Wert für G mündet. Dies sei am Beispiel ABC = 000 gezeigt: Wenn G vor dem Übergang auf diesen Eingangswert gleich Null war, dann gilt dies auch für D. In diesem Fall folgt sofort ein stabiler Zustand für den Ausgang G. War G jedoch vor dem Übergang auf Eins, dann wird G wie vorher auch Null, jedoch wird diese Null als Eingangswert D eingegeben, und nach einem zweiten Durchlauf befindet sich die Schaltung im Ruhezustand. Genauso verhalten sich auch alle anderen Zustände, wobei im Zustand ABC = 100 die Schaltung für G = 1 zur Ruhe kommt. Das Kennzeichnen der kurzzeitig instabilen Zustände wird in den Programmzeilen 340 und 350 vorgenommen.

Schwingungen

Noch kritischer verhält sich die Schaltung in *Bild 11,12 und 13*: Hier schwingt G zwar für alle Zustände nach kurzer Zeit ein, und eine Abhängigkeit vom vorangegangenen Wert ergibt sich in keinem Zustand, dafür zeigt aber ABC = 111 ein außerordentlich unschönes Verhalten: Die Schaltung schwingt! Man sieht das sofort daran, daß D und G immer verschiedene Werte haben. Daraus ergibt sich, daß G eine Eins rückkoppelt, wenn D gerade Null war und umgekehrt. Dieser Zustand ist demzufolge unbrauchbar, wenn nicht gerade ein Oszillator aufgebaut werden sollte. Im allgemeinen empfiehlt es sich, die Schaltung neu zu überarbeiten.

Die aufgeführten Beispiele decken in etwa den Problemkreis ab, der beim Entwurf von asynchronen Digitalschaltungen im Bereich der Logik auftreten kann.

Selbstverständlich müssen bei einem gründlichen Design noch weitere Anforderungen überprüft werden. Dazu zählen unter anderem Laufzeitprobleme, die nicht mit solch einfachen Hilfsmitteln erfaßt werden können. Nicht umsonst ist zum Thema Logik-Simulation eine schnell wachsende Industrie vornehmlich auf dem CAE/CAD-Sektor entstanden. In vielen Fällen wird die vorgestellte Lösung jedoch ausreichen; im Einzelfall können die Spezialprobleme immer noch einer separaten Betrachtung unterworfen werden.

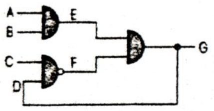


Bild 11. Nochmal Rückgekoppelt: Trotz der wenigen Elemente neigt diese Schaltung zum Schwingen

```
220 LPRINT
230 LPRINT" A  B  C  D  E  F  G"
240 LPRINT"---------------------
250 : FOR A=0 TO -1 STEP -1
260 :   FOR B=0 TO -1 STEP -1
270 :     FOR C=0 TO -1 STEP -1
280 :       FOR D=0 TO -1 STEP -1
290 :           E = (A AND B)
300 :           F = NOT (C AND D)
310 :           G = ( E AND F)
320 :           LPRINT ABS(A); ABS(B); ABS(C); ABS(D);
330 :           LPRINT ABS(E); ABS(F); ABS(G);
340 :           IF G <> D THEN LPRINT"*" : GOTO 360
350 :           IF G =  D THEN LPRINT
360 :       NEXT D
370 :       LPRINT"--------------------"
380 :     NEXT C
390 :   NEXT B
400 : NEXT A
410 LPRINT
420 LPRINT"Wahrheitstafel fuer Beispiel 5"
430 LPRINT
440 END
```

Bild 12. Das Simulationsprogramm zu Bild 11

A	B	C	D	E	F	G	
0	0	0	0	0	1	0	
0	0	0	1	0	1	0	*
0	0	1	0	0	1	0	
0	0	1	1	0	0	0	*
0	1	0	0	0	1	0	
0	1	0	1	0	1	0	*
0	1	1	0	0	1	0	
0	1	1	1	0	0	0	*
1	0	0	0	0	1	0	
1	0	0	1	0	1	0	*
1	0	1	0	0	1	0	
1	0	1	1	0	0	0	*
1	1	0	0	1	1	1	*
1	1	0	1	1	1	1	
1	1	1	0	1	1	1	*
1	1	1	1	1	0	0	*

Bild 13. Die Wahrheitstafel der Schaltung von Bild 11

The HACKTORY SOFTWARE

A. SOPP
Wakenitzstr. 8
2400 Lübeck 1
Telefon:
0451-79 19 26

Arnulf Sopp, Wakenitzstr. 8, D-2400 Lübeck 1

Herrn
Hartmut Obermann
Schwalbacher Str. 6

6209 Heidenrod-Kemel

Ihr Schreiben:	Ihr Zeichen:	Mein Schreiben:	Mein Zeichen:	Datum:
31. 10. 86				3. 11. 86

Auch Prost!

Lieber Hartmut,

man soll nicht glauben, daß so ein süffiger Tropfen noch ohne Glykol hergestellt werden kann. Wer packt das heute noch? Ganz herzlichen Dank dafür! Keine Frage, daß ich ihn auf das Wohl des Clubs und jeden dritten Schluck auf Dein spezielles einnehme.

Über den Auflösungsartikel für das nächste Info mußten wir reichlich lachen, mein Bauch und ich. Insb. daß Du Dir in Deiner Ratlosigkeit den Preis kurzerhand selber zuerkennst, zeugt sowohl von Humor als auch von einer vernaschten Zunge. Bei dem Stoff kein Wunder.

Bei den eingesandten Lösungen frage ich mich, wie ein Mensch mit einem Kopf durchschnittlicher Größe auf sowas kommen kann. Ohne Probelauf mit anschließend angeschmossenem Debugger zum Belauern der Registerinhalte oder was fühle ich mich sogar außerstande, sie zu verstehen. Ich begreife nur, daß Bernd und - äh ... - Hartmut falsche Lösungen anbieten. So trull dem Spender das Lala auch steigt, aber eine Lösung, die nur die Ergebnisse 0 und 1 zuläßt, kann nicht stimmen. Bei Bernd ist diese Alternative links und rechts vom ELSE eindeutig. Bei Dir nicht, aber:

Jede Zahl AND 1 kann je nach Bit 0 nur 0 oder 1 betragen, denn der Operand 1 hat nur das Bit 0 auf 1, der Rest ist 0. Nach der Wahrheitstabelle kann demzufolge kein anderes Ergebnis herauskommen, weil bei diesem Operanden alle anderen Partnerbits eine 0 antreffen. 0 AND ir-

genduas = 0, und das siebenmal. Nur beim allerersten mal (Bit 0) hat die 1 noch Chancen. Und 0000.0000.0000.0001b (BASIC-Integerzahl) ist nun mal = 1.

Beim (simulierten; oder sagt man emulierten?) XOR muß aber jeder Wert zwischen 0 und 255 bzw. 65535 möglich sein. Beispiel:

```
    0000.1111.0000.1111b
XOR 1100.1100.1100.1100b
=   1100.0011.1100.0011b = C3C3h = 50115d
```

aber (A sei meinetwegen = 5555h, B = 1C72h):

```
    0101.0101.0101.0101b (A)
+   0001.1100.0111.0010b (B)
=   0111.0001.1100.0111b (A+B)
AND 0000.0000.0000.0001b (1)
=   0000.0000.0000.0001b = 1d/h/o/dingsbums
```

Nimm es zwar ernst, aber nicht tragisch; es geht mir nur darum, mit aller Gewalt recht zu behalten. Aber Jux beiseite, falls ich nicht irre (Kajott wäre vielleicht der geeignete Prüfstein meiner Behauptung), solltest Du das unbedingt rechtzeitig vor der Veröffentlichung korrigieren, denn unter den Clubmitgliedern sind außer den Teilnehmern vielleicht noch dreieinhalb, die das merken. Du hast einen Ruf (nach Harvard?) zu verlieren! Bernds Buddel kannst Du ja als Anerkennungspreis für den fleißigen Teilnehmer deklarieren, der nur leider auf der falschen Fährte landete. Wenn in dem von Dir angekündigten mc-Artikel auch sowas steht, gäbe das Stoff für eine schöne Glosse über Redaktionen, die allenfalls Kommafehler bemerken.

Das (nach meiner Theorie) Mißverständnis ist möglicherweise darin begründet, daß solche Wahrheitstabellen nur ein Bit beschreiben. Dabei muß man aber im Auge haben, daß dieses eine Bit nur Beispiel für alle übrigen ist. Um nun nicht eine Tabelle mit 256² oder gar noch mehr verschiedenen Operanden-Paarungen auf ein lausiges Stück DIN A4 gnubbeln zu müssen, unterwirft man sich dieser Beschränkung und den Leser der seinigen. Pardon!

Welches Beispielprogramm Kajott abgeliefert hat, weiß ich nicht. Nach den Resultaten bin ich offen gestanden ein bißchen skeptisch. Mein Beispielprogramm ist anschließend gelistet. Der ML-Teil (da fühle ich mich sicherer als in BASIC) ist absolut wasserdicht. Die CALLs nach 0A7F und 0A9A werden im User's Manual zur ~~Übergabe und -nahme~~ von Parametern an/von ML-Programmen angegeben. Was da herauskommt, ist ein bombensicheres XOR. Es wird in Zeile 60 angezeigt. Nun kannst Du alle Lösungen als Zeile 70 mit anschließendem PRINT C eintippen. Dann kannst Du mit ein paar Testoperanden herumspielen und sehen, ob das USR-Ergebnis mit dem Lösungsvorschlag übereinstimmt. Wehe, wenn nicht!

Für das nächste Glas von Deinem köstlichen Dope brauche ich die nötige Andacht. Deshalb Schlunz jetzt und EOF.

Dein

Hallo Hartmut!

Nachträglich fiel mir noch auf, daß das Bit 15 bei Integerzahlen als Vorzeichen interpretiert wird. Daraus schloß ich messerscharf, daß der in der ML-Routine höherwertige Operand max. 127 betragen darf, sonst ist das Ergebnis zwar richtig, aber wegen des Minuszeichens schwierig zu interpretieren. Also probierte ich das vorsichtshalber mal eben durch und stellte fest, daß sogar beide Operanden höchstens 127 betragen dürfen, sonst wird Überlauf vermeldet. Das ist aber nur eine Besonderheit des BASIC, mit der diese Testroutine nicht klarkommt. Sonst wäre es wesentlich komplizierter geworden. Es ändert nichts an der Tauglichkeit der Routine, denn bereits 2 ist mehr als 1; es ging ja darum, zu zeigen, daß die beiden (oder mehr?) falschen Lösungen nur max. 1 hervorbringen.

Das mußte ich noch nachtragen, damit der Test bei der Wahl ungeeigneter Operanden nicht plötzlich "beweist", daß alle Lösungen falsch sind.

Ob der Wein damit zu tun hat, daß ich gestern abend nicht zuende dachte? Sei's drum, er war die Blamage wert!

Mit gelallten Grüßen, Dein

```
10 CMD"load,ätsch/cmd"'          USR-Routine laden
20 INPUT"A (max. 255):";A'       1. Operand
30 INPUT"B (max. 255):";B'       2. Operand
40 C=A*256+B'                    2 X 8 Bit für XOR
50 DEFUSR0=&HF000'               XOR-Operation durchführen
60 PRINTUSR0(C)'                 Ergebnis verraten
```

```
                00001 ;*******************************************************
                00002 ;                   ATSCH/CMD
                00003 ;          Routine zur Entlarvung von
                00004 ;          alkoholsüchtigen Wein-Erschleichern
                00005 ;          (C) 1986 by Don Promillo
                00006 ;*******************************************************
                00007
F000            00008          ORG     0f000h
F000  CD7F0A    00009          CALL    0a7fh          ;HL <- USR-Argument
F003  7C        00010          LD      A,H            ;1. Operand
F004  AD        00011          XOR     L              ;2. Operand, Erg. -> A
F005  6F        00012          LD      L,A            ;HL mit Ergebnis laden
F006  2600      00013          LD      H,00h          ;MSB = 0, weil nur 8 Bit
F008  C39A0A    00014          JP      0a9ah          ;Argument an BASIC überg.
0000            00015          END

00000 Fehler
```

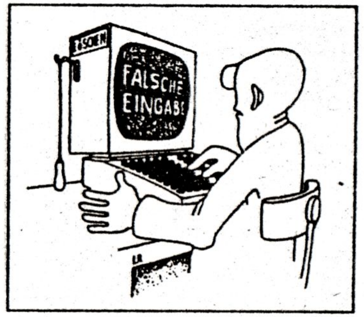
Liebe Freunde,

natürlich hat unser Bitfummelmeister Arnulf recht mit dem, was er über Bernd's und meine Lösung (es ist übrigens eine von mehreren die ich ausprobiert habe!) sagt. Geirrt hat er sich allerdings wenn er glaubt, mir wäre dieser "Fehler" nicht aufgefallen. In dem Programm für welches ich die XOR-Funktion eigentlich brauchte, ist mir die Unbrauchbarkeit dieser Lösung sofort aufgefallen. Trotzdem freue ich mich über die prompte Reaktion und scheue mich nicht, Arnulfs Brief (gekürzt!) und damit meinen Fehler sofort zu veröffentlichen, vor allem da dadurch doch noch eine Fortsetzung von "Logische Operationen in BASIC" erscheint!

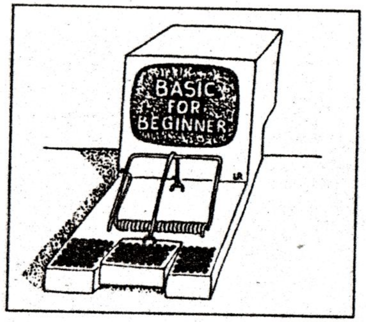
Vielen Dank dafür an Arnulf und alles Gute für alle Clubmitglieder, euer

Hartmut Abermann

Weitere Informationen über die Digitaltechnik in unserem Sonderheft am Ende des INFO's
Die Redaktion



Frage an den Computer: »Kann ein Computer auch Witze machen?« Antwort des Computers: »Selbstverständlich. Er kann nur nicht drüber lachen.«



BASIC – Laufende Programme beliebig erweitern

Mit der hier vorgestellten Routine läßt sich ein bereits laufendes BASIC-Programm um eine oder mehrere Programmzeilen erweitern. Ändert man normalerweise ein Programm, dann muß der BASIC-Interpreter angehalten werden. Mit einem RUN-Befehl läßt sich dann das modifizierte Programm wieder starten, allerdings mit einem Schönheitsfehler: Die Variablen des ursprünglichen Programmes sind gelöscht. Ganz anders bei der hier vorgestellten Routine: Sie ist selbst ein Teil des momentan laufenden BASIC-Programmes und kann somit dessen Text erweitern, ohne daß die Variablen in der vom BASIC-Interpreter angefertigten Tabelle überschrieben werden.

Vorteilhaft kann man auch das

```
10 CLS:DIMPZ$(500):CLEAR5000
20 INPUT"Wieviele Programmzeilen sollen angehaengt werden ";N
30 INPUT"Zeilennummer der ersten anzuhaengenden Zeile     ";Z
35 INPUT"Abstand zwischen den neuen Programmzeilen        ";I
40 FORA=1TON
50 PRINTA"- te Zeile eingeben ( Zeile"Z;")";:INPUTP$(A)
60 P$(A)=STR$(Z)+" "+P$(A)+CHR$(13)
70 Z=Z+I:NEXTA
80 OPEN"O",1,"INTER"
90 FORA=1TON
100 PRINT#1,P$(A);
110 NEXTA:CLOSE
120 MERGE"INTER"
130 INPUT"Inter-File loeschen (J/N) ";D$
140 IFD$="J"THENKILL"INTER"
150 END: REM  ***** ODER WEITERES PROGRAMM *****
```

kurze BASIC-Programm als Unterprogramm schreiben, das dann ein laufendes Programm beliebig oft aufruft. So lassen sich zum Beispiel recht leicht bei der Programmentwicklung neue mathematische Funktionen definieren, die dann als gewöhnliche Programmzeilen am Ende des BASIC-Programmes stehen. Oder eine andere interessante Anwendung: Im Speicher steht ein Maschinensprache-Programm. Es läßt sich über PEEK-Instruktionen lesen und anschließend als DATA-Zeile an das Programmende anfügen. Die Umrechnung der hexadezimalen Werte in dezimale entfällt, da diese Arbeit die hier vorgestellte Routine automatisch erledigt.

In den Zeilen 20 bis 30 fragt die Routine nach der Anzahl der anzuhängenden Zeilen sowie der Startzeile. Die zuvor eingegebenen Zeilennummern werden dann mitsamt dem Zeileninhalt in Zeichenketten umgewandelt. Anschließend erweitert das Programm jede Zeichenkette um ein Leer- und ein Return-Zeichen und erzeugt somit eine komplette BASIC-Zeile.

Sind alle neuen Zeilen eingegeben, dann werden sie im ASCII-Format auf die Diskette gerettet. Der MERGE-Befehl in der Zeile 120 fügt die zuvor gespeicherten Zeilen dem momentan laufenden BASIC-Programm an. Auf die so angehängten Zeilen kann dann BASIC über eine GOTO- oder GOSUB-Instruktion zugreifen.

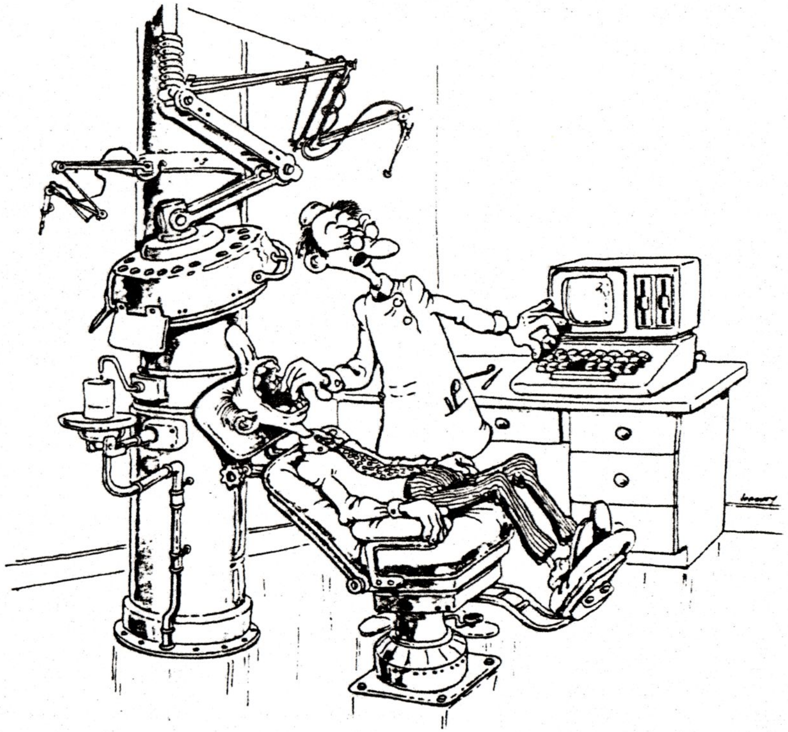
Beim Anfügen der Zeilen ist jedoch darauf zu achten, daß die erste Zeilennummer größer ist als die letzte vorhandene im laufenden BASIC-Programm. Nur so kann der MERGE-Befehl korrekt arbeiten.

Bei der Programm-Erweiterung entsteht auf der Diskette eine temporäre Datei mit dem Namen „INTER". In der Zeile 130 fragt deshalb die Routine, ob diese Datei wieder gelöscht werden soll.

Ursprünglich ist das Hilfsprogramm auf einem TRS80-Computer unter Newdos entworfen worden. Es läßt sich aber auch ohne Probleme an fast alle anderen Computer anpassen, deren BASIC-Interpreter über einen MERGE-Befehl verfügt.

Ralf Krenzke

```
100 ' Spline Interpolation (kubisch)             10.08.1986
110 ' BEDEUTUNG:    Zu gegebenen Punkten...
111 '                 >>    X: 1, 4, 6, 7           ... N = 4
112 '                 >>    Y: 0, 1, 6, 3
113 '                  ( X-Werte müssen aufsteigend sein !! )
114 '              ...eine möglichst glatte Kurve durch sie zu
115 '                 legen. Diese Methode ist wesentlich besser
116 '                 als die bekannte Polynominterpolation !
117 '
118 ' Algorithmus natürlich nicht von mir.
120 '
130 ' Dieter Kasper
140 ' Zeppelinstr. 9
150 ' 8952 Marktoberdorf
160 ' Tel.: 08342 / 1630
170 '
180 '
190 ' N+1: Anzahl der Stützpunkte
200 READ N:N=N-1
210 DIM X(N),Y(N)                      :'Stützpunkte
220 DIM D(N),R(N),B(N),C(N),Z(N)  :'Hilfsfelder
230 DIM XX(128),YY(128): Wert des SPLINES an der Stelle (XX,YY)
240 '
250 'Einlesen der Stützpunkte
260 FOR I=0 TO N: READ X(I) : NEXT
270 FOR I=0 TO N: READ Y(I) : NEXT
280 'FOR I=0 TO N: INPUT" X , Y";X(I),Y(I):NEXT
290 '
300 ' Aufstellen des Tridiagonalen Gleichungssystems
310 ' und Lösen desselben
320 D(0) = X(1)-X(0)
330 FOR I=1 TO N-1: D(I) = X(I+1)-X(I)
                  : C(I) = 2*(D(I-1)+D(I)): NEXT
340 Z(2)=D(1)/C(1)
350 FOR I=3 TO N-1: Z(I)=D(I-1)/(C(I-1)-D(I-2)*Z(I-1)):NEXT
360 FOR I=1 TO N-1: R(I)=(Y(I+1)-Y(I))/D(I)-(Y(I)-Y(I-1))/D(I-1):
                    R(I)=3*R(I):                    NEXT
370 B(1)=R(1)/C(1)
380 FOR I=2 TO N-1: B(I)=(R(I)-D(I-1)*B(I-1))/(C(I)-D(I-1)*Z(I)):
                    NEXT
390 C(N-1)=B(N-1)
400 FOR I=N-2 TO1STEP-1: C(I) = (B(I)-Z(I+1)*C(I+1)): NEXT
410 C(0)=0: C(N)=0
420 FOR I=0 TO N-1: B(I)=(Y(I+1)-Y(I))/D(I)
                  : B(I)=B(I)-((2*C(I)+C(I+1))*D(I)/3)
                  : D(I)=(C(I+1)-C(I))/(3*D(I)):          NEXT
430 ' An welchen Stellen soll die Splinefunktion ausgewertet
      werden ?
440 W = (X(N)-X(0))/127
450 XX(0)=X(0):FOR I=1 TO 127 : XX(I)=XX(I-1)+W : NEXT
460 '
470 'Auswertung des Splines ( X-Werte zwischen x(0) und x(n)!)
480 FOR J=0 TO 127: XX=XX(J)
490 FOR I=0 TO N-1
500 IF (X(I+1)-XX)>=0 THEN XX=XX-X(I):
                        YY(J)=Y(I)+XX*(B(I)+XX*(C(I)+XX*D(I)))
                      : GOTO 520
510 NEXT I
520 NEXT J
540 ' Zeichnen der Kurve
550 IN=YY(0):SU=IN
560 FOR I=1 TO 127:Y=YY(I): IF Y>SU THEN SU=Y :GOTO 580
570                          IF Y<IN THEN IN=Y
580           NEXT
590 FOR I=1 TO N:Y=Y(I): IF Y>SU THEN SU=Y :GOTO 610
600                       IF Y<IN THEN IN=Y
610 NEXT
620 CLS
630 FOR I=0 TO N: Y=Y(I): GOSUB 700
640                  K = (X(I)-X(0))/W
650                  FOR J=47 TO Z STEP -1: SET (K,J): NEXT J,I
660 FOR I=0 TO 127 : Y=YY(I): GOSUB 700
670                     SET (I,Z): NEXT
680 GOTO 680
690 END
700 Z=INT(47*(SU-Y)/(SU-IN)):RETURN
710 DATA  10
720 DATA  1,2,3,4,5,6,7,8,9,10
730 DATA  0,0,0,0,0,9,9,9,9,9
```

Endlich weiß ich, wozu dies verdammte Ding zu gebrauchen ist.

* Goldener FIBONACCI ! *

Her mit Papier, Bleistift und Lineal! Laßt uns eine Strecke zeichnen von beliebiger Länge.
Nennen wir diese Länge "1"; sie wird unser "persönlicher" Maßstab! Markieren wir ihre Enden.

Jetzt verlängern wir diese Strecke um ein Stück s. s soll kürzer sein als 1, und zwar nur so lang, daß es zur Strecke 1 im gleichen Verhältnis steht wie die Strecke 1 zur Gesamtstrecke 1+s; so daß gilt:

$$s:1 = 1:(1+s) \quad \text{oder kurz} \quad \underline{s = 1/(1+s)}$$

Umformung ergibt die quadratische Gleichung: $s^2 + s = 1$.
Daraus erhalten wir als (positive) Lösung:

$$s = (\sqrt{5} - 1)/2 = 0{,}618039...$$

also natürlich eine Irrationalzahl (wegen der Wurzel aus einer Zahl, die kein Integer-Quadrat ist); aber das stört uns nicht, wir brechen sie ab, wo es uns Spaß macht. Z.B. wie hier nach der 6.Dezimalstelle. Oder auch erst nach der 15.!
Wieso 15? Wer wird so lange an der Wurzel herumrechnen? Na wat'n, wozu ham'wa denn'n Kompjuta?? sacht da Berlina!

Also: laßt uns ein Programm "entwickeln!" - Aber nicht eins, das eine genauere Wurzel aus 5 zieht - das wäre uninteressant - sondern eins, das sich die Gesetzmäßigkeit zunutze macht, die in unserer Ableitung des Streckenverhältnisses 0,618039...:1 steckt!
Dies ist nämlich nichts anderes als der berühmte

"Goldene Schnitt"

den sich viele Maler, Bildhauer, Architekten und andere als Leitbild (Leitmaß!) genommen haben. Dieser liegt dann vor, wenn sich eine Strecke zu einer größeren genau so verhält wie die größere ihrerseits zur Gesamtlänge, also zur Summe beider Teilstrecken.

Setzen wir dies Spielchen fort!
Wir fügen zur Gesamtlänge (1+s) wieder ein Stückchen hinzu, das zu (1+s) im gleichen Verhältnis steht wie s:1. Und siehe da - dann steht auch (1+s) zu dieser neuen Gesamtlänge wieder in dem gleichen Verhältnis! Wie lang ist diese neue Gesamtlänge? Natürlich (1+s)*(1+s)! Die Formel für jede neue Länge - wenn wir so fortfahren - lautet: $s_n = (1+s)^n$.
Aber uns interessieren nicht die Längen, sondern nur deren Verhältnisse untereinander. * Vergegenwärigen wir uns noch einmal das Vorgehen. Zuerst galt doch:

s:1 = 1:(s+1), in Worten:

Der Zähler des neuen Bruches verhält sich zu seinem Nenner wie der Nenner des vorhergehenden Bruches zu dessen Summe Zähler+Nenner. Der alte Nenner wird also neuer Zähler und die Summe (alter Zähler+Nenner) wird neuer Nenner.
Ganz einfach*). Was kommt da heraus? (Kürzen wir ab: Zähler=Z, Nenner=N, alt=a, neu=n)

(1) nZ = aN =
nN = aZ (s) + aN (1) = $\frac{1}{s+1}$

(2) nZ = aN =
nN = aZ (1) + aN (s+1) = $\frac{s+1}{s+2}$

(3) nZ = aN =
nN = aZ (s+1) + aN (s+2) = $\frac{s+2}{2s+3}$

(4) nZ = aN =
nN = aZ (s+2) + aN (2s+3) = $\frac{2s+3}{3s+5}$

(5) nZ = aN =
nN = aZ (2s+3) + aN (3s+5) = $\frac{3s+5}{5s+8}$

(6) .... $\frac{5s+8}{8s+13}$

(7) .... $\frac{8s+13}{13s+21}$

(8) .... $\frac{13s+21}{21s+34}$

u.s.w.

Die Zahlenfolge, die sich hier deutlich abhebt, heißt die

FIBONACCI-Folge

nach dem italienischen Mathematiker *Leonardo Pisano Fibonacci* (ca. 1180-1240). Sie entsteht, wenn man mit 0 und 1 beginnt und zur letzten immer die vorletzte Zahl addiert (Rekursionsformel):
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 .....
Das 1+ n.Glied lautet: $0{,}4472136[1{,}618034^n-(-0{,}618034)^n]$.
Bildet man das Verhältnis zweier Nachbarglieder, so stimmt es um so besser mit dem Goldenen Schnitt überein, je weiter man in der Folge voranschreitet - der Fehler verschwindet aber nie ganz, weil in Zähler und Nenner immer das "Korrekturglied" s fehlt. Es ist dann so, als hätte man in obigen Brüchen s=0 oder s=1 gesetzt. Daß dem so ist, zeigt das nachstehende Programm recht deutlich. Die Qotientenfolge konvergiert erst im Unendlichen gegen s. [Wegen der "Genauigkeit" unseres Computers konvergiert sie ab 29.Glied nicht mehr.]

So ist also *FIBONACCI* das mathematische Pendant zu den zahlreichen Alchimisten des Mittelalters, jenen Chemikern, die sich unzählige Nächte vergeblich abmühten, aus dem "Stein der Weisen" GOLD zu machen: Er hat es fast geschafft! Wie diese war er nahe dran - hätte er nur schon an den "Goldenen Schnitt" gedacht! * Aber das wäre so gewesen, als hätte ein Alchimist Aurum in seine Brühe gegossen und dann gerufen: εὑρεκα!
Und sowas macht man nicht!

- Also sprach χαιοτ μß

*)
Der Mathematiker wird bemerkt haben, daß die Ableitung nicht ganz zwingend war. Der saubere Beweis geht einen anderen Weg, den ich mir hier aber verkniffen habe, weil wir uns ja nicht in einem Lehrgang über Zahlentheorie befinden. - Hugh!

FIBONACCI-Folge und 'Goldener Schnitt'

von KaJot Mb

```
10 CLS
   PRINT CHR$ (23)"       Die FIBONACCI-Reihe"
   FOR I = 1 TO 300
   NEXT
   PRINT CHR$ (28)
20 CLEAR 70
   DEFINT I
   DEFDBL V,N,Z
   DEFSTR F,S
   S = STRING$ (63,"-")
   N = 1
   F = ".6180339887"
   PRINT "Wieviele Rekursionsschritte ? ";
   INPUT "(Bitte nur gerade Anzahl eingeben)";IT
   ZZ = 2
   FOR IO = 1 TO IT STEP 20
30 CLS
   PRINT " V                N                Quotient V/N"
   PRINT S
40 PRINT "'Goldener Schnitt' = Grenzwert: "F
   PRINT S
50 IF IO = 1
   THEN PRINT V,N
60 FOR I = 1 TO 20
   ZZ = ZZ + 1
   IF ZZ < = IT
   THEN Z = N
        N = N + V
        ON (2 + (2 * INT (I / 2) = I)) GOSUB 80,70
        V = Z
        NEXT I
        INPUT "weiter <ENTER> ";E$
        NEXT IO
        END
   ELSE END
70 PRINT N,
   RETURN
80 PRINT N,Z / N
   RETURN
```

[... ein Achtzeiler !]

* * * STENO-LOGIK * * *

Lewis ROSENFELDER zeigt in "BASIC FASTER AND BETTER & other mystries", erschienen 1981 bei *IJG Inc.* (existiert m.W. nicht mehr), ISBN 0 936200 030, Möglichkeiten im Umgang mit BASIC (z.T. unter Verwendung von "Maschinensprache"), die hier und da auch den geübten und erfahrenen Programmierer überraschen und bereichern. Es ist daher sehr bedauerlich, daß die in unserem INFO begonnene recht gute Übersetzung des - immerhin fast 300 Seiten langen - Buches nicht vollendet werden konnte. (Ehrlich: Ich möchte diese Herkulesarbeit nicht machen!) Lest einmal seine Erläuterungen zum Aufbau von "Logik-Kürzeln", wie ich es einmal nennen möchte. Seit ich das las, verwende ich sie mit Vorliebe.

Das Prinzip besteht darin, algebraische Beziehungen (Gleichungen/Ungleichungen) in logische Wahrheitswerte

logisch "WAHR" = 1 (bzw. im BASIC: -1)
logisch "NICHT-WAHR" = 0

umzusetzen.

Zwischenbemerkung
"Logisch Nicht-Wahr" ist übrigens nicht etwa dasselbe wie "logisch falsch"! "Logisch falsch" ist schlichtweg unlogisch und nichts weiter wie ein Denkfehler; es stellt sich quer, d.h. ein Programm, das eine Unlogik enthält, läuft erst gar nicht. Man muß den Fehler suchen. Wenn ich schreibe "2 * 2 = 3.9", so ist das zwar "falsch", aber nicht "unlogisch": die Logik besteht hier darin, daß eine Antwort jedenfalls möglich ist. Sie lautet (ohne besondere Lautstärke): "Das ist nicht WAHR! Deine 'Gleichung' hat den 'Wahrheitswert Null'!" Das Programm antwortet allerdings kürzer: einfach mit "0". Hingegen wäre es unlogisch zu schreiben: "2 * 2 = VIER" oder "2 * 2 =WEISSNICH", denn numerische Variable können nicht einem String gleichgesetzt werden. Dieses Programm "läuft" nicht; es folgt ==>ERROR-Meldung. (Nun ja,das ist auch eine "Antwort"...)
- Ende der Zwischenbemerkung.

Wie geht das vor sich?
Lade BASIC und gib ein: PRINT 3=4
BASIC antwortet hierauf leider nicht: "Gehe nochmal in die Schule und gib (!) dir eine Chance!" - sondern sagt schlicht: "NULL!" (0).

Gib jetzt ein: A = (3=4)
PRINT A
Die Antwort lautet wieder: "NULL!" (A=0)

Auf A=3>4
PRINT A
folgt ebenfalls: A=0

Nun aber: A = 3<4
Print A
---> -1 erklärt uns der fröhliche Bildschirm! Dies aber nicht etwa deshalb, weil 3-4=-1 (was ja in der Tat richtig ist), sondern weil es WAHR ist, was da geschrieben steht.

Wir fragen jetzt mal: PRINT 3 < 100. Auch hierauf lesen wir -1, obwohl 3-100=-97 ist (glaube ich).

Jetzt schreiben wir: A = 3 * 12
PRINT A

und erhalten 36 (ich fragte meinen Enkel. Es stimmt!! Der kann vielleicht rechnen!...) und nicht -1, denn "3*12" ist ja keine Behauptung, sondern eine Frage. Fragen beantwortet der Computer, so gut er kann (jawohl, das Kleine und auch das Große Einmaleins – das kann er!)

Nun fragen wir nicht, sondern wissen, daß 3*12=36 ist. Mit diesem Wissen formulieren wir die Behauptung:

A = (3 * 12) = 36

Auf PRINT A antwortet BASIC: "Ja, stimmt!" – allerdings in seinem Jargon nur kurz: "-1".
Probiert's: A = (3*12=35)! Antwort: A=0 (in Worten: "Du lügst!"

Kurzum: Wenn wir eine Variable mit einer Gleichung oder auch einer Ungleichung laden, so nimmt sie nicht irgendeinen Zahlenwert daraus an, sondern "-1 XOR 0"; in Worten:
Entweder -1, wenn die (Un-)Gleichung WAHR,
oder 0, wenn sie NICHT-WAHR war.

Kleiner Alptraum:
Aus dem, was früher einmal wahr war,
ergibt sich später oftmals Wirrwar...

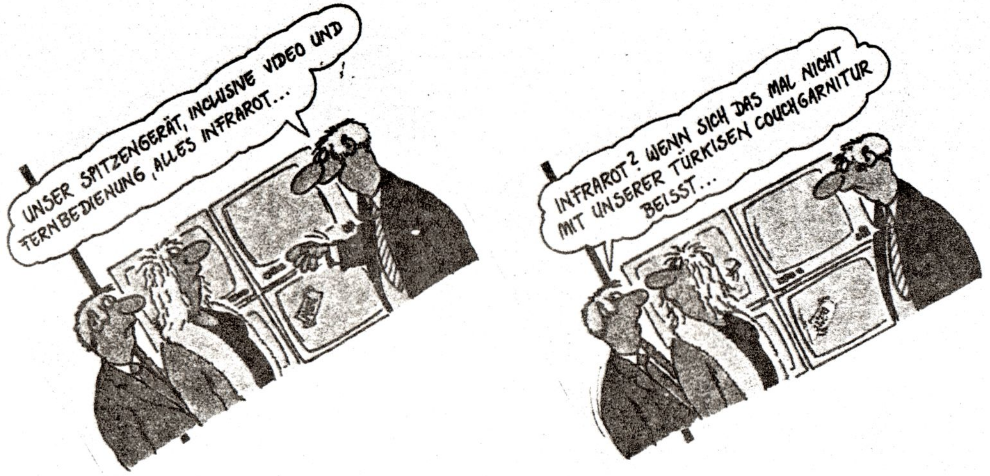
Auch eine falsche "Gleichung" ist nicht etwa "unlogisch", sondern sie ist die (oft absichtliche) "logische Darstellung von etwas Falschem"! – Alles klar?
(Nicht nur Diplomaten und Werbeleute argumentieren so spitzfindig! Mathematik dringt sehr spitz überall ein, und mathematische Logik hilft ihr beim Anspitzen...)

Das Tollste ist: Die gleichen Späßchen kann man nicht nur mit Zahlen (numeri), sondern auch mit Strings veranstalten. Und das macht die "Steno-Logik" so reiz- und besonders wertvoll!

Eine praktische Anwendung dieser Zusammenhänge wäre etwa:
Wir besitzen ein Lexikon mit drei Bänden:
Band 1: A – H
Band 2: I – Q
Band 3: R – Z
Wir suchen nun ein Wort: LINEINPUT"Was suchst du ?";W$
Die folgende Zeile sagt uns, in welchem Band dieses Wort zu finden ist:

```
B%=1-( (W$ > "HZ") + (W$ > "QZ") )
==================================
```
*) s.u.

Wenn W$ = "FRANZ", so sind beide Ungleichungen NICHTWAHR (F ist nicht größer als H und erst recht nicht größer als Q). Die Klammern werden Null und es bleibt die 1, also Band 1! Ebenso für "HERMANN".
Wenn W$ = "LIES'CHEN", ist die erste Ungleichung W$>"HZ" WAHR = -1, die zweite aber 0 (NULL), da L<QZ.
-1 von 1 abgezogen, ergibt 2 ⇒ Band 2!
Wenn W$ = "QZA", landen wir in Band 3 statt 2 (QZA > QZ) und werden dort vergeblich nach dem Wort "QZA" suchen – weil es das nicht gibt. Wir werden es deshalb wohl auch kaum suchen!
Wenn W$ = "RUDI", dann treffen beide Ungleichungen zu und jede liefert -1. Somit erhalten wir: B% = 1-(-1+-1) = 3! Im Band 3 unseres Nokixels finden wir garantiert "RUDI", falls er den Eintrag sich nicht verbat!



(Wer tät' das schon
und stünd' nicht gern im Klexion??)

Ich habe das Z hinter H bzw. Q angefügt, damit man bei Eingabe von Wörtern, die mit H bzw. Q anfangen, nicht im nächsten Band landet. (Wieso? – Aha! Gelle? – Man kann das natürlich auch noch narrensicherer formulieren. Übt euch!) – Das Z hinter dem H bzw. Q mußte also sein!

Nun mag der eine odere der andere meinen, das sei doch etwas "umständlich" und er würde "Le mots" auch so finden (nicht von *SARTRE*), denn die Anfangsbuchstaben stünden ja auf den Buchrücken! Bittschön! Er begebe sich der Chance, "STENO-LOGIK" zu üben!
Oder er finde geistreichere und nützlichere Anwendungen! Das Rüstzeug dazu hat er nun. Und wenn diese wenigen Sätze zur Sache eine Anregung waren, so freut sich auch der Verfasser.

Aber wesentlich mehr, wie gesagt, bei *ROSENFELDER!*
Ihn empfiehlt euch ohnehin: Xalot

---

*) statt :

IF ((W$ > "HZ") AND (W$ <= "QZ")) THEN B% = 2 ELSE
IF W$ > "QZ" THEN B% = 3 ELSE B% = 1

Das ist 2,5 mal so lang !

PS: Solche (in abhängigen Variablen implizierten) Gleichungen bzw. Ungleichungen dürfen auch als Faktoren auftreten !

Kj

Graphik-JKL, TSCRIPS und der NEC P6

Beim Kauf eines neuen Druckers ist fast immer eine Anpassung der alten Formulare auf die neuen Möglichkeiten und gegf. auf die geänderten Steuersequenzen notwendig. Glücklicherweise orientieren sich viele Anbieter an der Epson-"Norm", inzwischen endlich auch NEC. So war der Umstieg mit relativ wenig Arbeit verbunden. An dieser Stelle soll beschrieben werden, was da zu tun war. In meiner speziellen Situation war es der Weg von Gemini zu NEC, der Computer ist das G3s, das Textprogramm ist TSCRIPS, die JKL-Routine Bestandteil von H-DOS 2.4. Aber mit jeder anderen Hard/Soft-Konfiguration treten dieselben Probleme auf, so daß ich ausnahmsweise mal wieder hoffen kann, etwas für alle zu bieten.

Fangen wir mit JKL an. Eine Routine für die Graphik-Hardcopy braucht bei der Anpassung von Gemini oder Epson auf den NEC Pinwriter P6/P7 kaum mehr als den Zeilenabstand zu ändern. Die Einstellung von n/144" kennt er nämlich nicht. Von seinen anderen Möglichkeiten ist hier die Anwahl von n/180" zu empfehlen. Das ist natürlich nicht dasselbe, daher muß n geändert werden. Rechnerisch ergibt sich der eineinviertelfache Zähler. In Versuchen hat sich herausgestellt, daß wahrscheinlich wegen größerer Zeichenhöhe das Eineinhalbfache korrekt ist.

Im konkreten Beispiel von H-DOS auf dem G3s hatte das folgende Konsequenzen: Die Routine druckt zwei Halbzeilen, wovon die obere immer 8 Dots hoch ist. Das entspricht auf Gemini 16/144". Für den NEC werden es 24/180". Die untere Halbzeile hängt vom eingestellten Bildschirmformat ab und ist variabel. Der neue Algorithmus zur Berechnung soll hier nicht vorgestellt werden, denn das ist wirklich zu G3s-spezifisch. Für die Computer mit dem unveränderlichen Bildschirm von 16mal 64 Zeichen mit je 12 Dots Höhe werden hier für beide Halbzeilen 18/180" statt 12/144" eingestellt.

Die Randeinstellung geschieht beim Gemini mit ESC-M-n. Das 'M' wird beim NEC, wie bei den Epson-Typen, einfach gegen ein 'l' getauscht. Und schließlich ist noch die Initialisierung des Druckers zu ändern, damit nicht vorherige Einstellungen die Hardcopy stören. Außerdem wird zumindest in H-DOS der Drucker anschließend wieder resettet, um die krausen Einstellungen nicht in einem späteren Listing oder Brief wiederzufinden. Bei Epson und Gemini klappt das mit ESC-§. Diesen Code gibt es auch für den NEC; dabei werden aber viele Voreinstellungen nicht zurückgesetzt. Das Total-Reset hat den Code FS-§ (1Ch-40h).

Die geänderten Codes sind in dem Sektordump am Ende dieses Artikels unterstrichen. Dort sind die Werte selbstverständlich in Hex wiedergegeben. Der NEC-Graphikmodus mit 8 Nadeln geht mit dem Papier leider sehr zärtlich um, so daß die Kopie im Info vielleicht schwer zu lesen ist. Ich wollte jedoch das Resultat der Änderungen präsentieren und nahm daher nicht das gewöhnliche JKL. Eine Programmversion, die mit allen 24 Nadeln schön schwarze Flächen malt, habe ich noch nicht erstellt.

Auch mit TSCRIPS hat es Überraschungen gegeben. Dieses Programm möchte dem User möglichst viel Arbeit bei der Druckerformatierung abnehmen und stellt deshalb eine Reihe von Befehlen zur Verfügung. Wenn nun dummerweise der Drucker dieselben Befehle versteht, und man muß nun unbedingt einen solchen an ihn, nicht an TSCRIPS ausgeben, was dann?

Um die Überschrift mache ich für solche Artikel in aller Regel nicht viel Gedöns. Sie wird meist nur gesperrt gedruckt. Um zu zeigen, wie die eben beschriebenen Schwierigkeiten in den Griff zu kriegen sind, ist sie diesmal aber äußerst aufwendig geworden. Auf dem TSCRIPS-Bildschirm sah das so aus:


```
>PL=72 LM=0 RM=130 TM=4 BM=68 PF=2■
&1C40&$1&09&$p&011B4D1C56010E&$4Graphik-JKL, TSCRIPS und der NEC
P6&1C5600&$5$p&001B45&■
>RM=74■
```

Die obere Zeile steuert für TSCRIPS das Papierformat. Dort wird der rechte Rand auf 130 Zeichen eingestellt, damit die leider mitgezählten Steuerzeichen der zweiten und dritten Zeile nicht zu einem Zeilenvorschub mitten in der Zeile führen und der Randausgleich nicht zwischen alle Wörter riesige Blank-Lücken setzt. In der letzten Zeile vor dem eigentlichen Text wird wieder auf meine übliche Zeilenlänge von 74 Zeichen zurückgekehrt.

Es beginnt mit FS-§, dem Brutalo-Reset. Das '§' ist ein gewöhnliches ASCII-Zeichen, das von TSCRIPS unverändert auf den Drucker gegeben werden sollte. Wird es aber nicht. So wird es denn eben zwischen '&'-Zeichen eingeschlossen als Hexzahl codiert. Der nächste Kopfstand dieser Art ist einige Zeichen weiter die Elite-Einstellung mit ESC-M (bei Gemini ESC-B-2). Auch hier heißt es stattdessen in Hex 1B-4D. Mit dem Gemini war die Eingabe $B&02& noch möglich, weil TSCRIPS gegen das 'B' offenbar weniger einzuwenden hat als gegen das 'M'.

Dies gilt noch für eine ganze Reihe weiterer Codes. Ohne jetzt die komplette Überschrift erklären zu wollen (viel Spaß beim Tüfteln!), möchte ich einfach empfehlen, zu experimentieren. Wann immer ein Code offensichtlich nicht korrekt an den Drucker übergeben wird, sollte man es zunächst mit der '&'-Methode versuchen. Oder wie das je nach Textprogramm auch immer gelöst sein mag.

Arnulf Sopp

```
H-DOS-Diskdaten-Editor Vers. 2.4
(C) 1986 by TCS / The HACKTORY

Datei: SYS27/SYS

drv: 0          frs: 0003h     drs: 0611h
000300: 064F AF32 A150 79CB 7F20 053E 01E7 0050
000310: AF32 9A50 0600 2100 0011 0008 C5E5 0600
000320: C53E FF0E 18CD 6550 D919 D9C1 E1E5 1640
000330: 1916 083E 000E 00CD 6550 0E00 E109 C1F1
000340: 4FD3 FA3A 4038 CB57 2006 79F5 F618 D3FA
000350: 10CA F1D3 FA31 0000 FB21 DE50 CD51 50AF
000360: C97E 23FE FFC8 CD5B 5018 F6F5 CDD1 0520
000370: FBF1 D3FD C9E5 32BB 503A 8E50 AA32 8E50
000380: 21D8 5071 21D5 50CD 5150 E1C5 E5E5 D97E
000390: 23D9 2600 0604 07CB 1410 FBE6 F0C6 086F
0003A0: 2442 08CB C7D3 F9CB 8703 7E2C E308 D3F9
0003B0: 08AE C542 E521 CA4F 07CB 1623 10FA E1C1
0003C0: 19E3 10DE E121 D14F 427E E600 CD5B 5018
0003D0: 03CD 5B50 3AC0 50EE 0331 1050 1E10 EAE1
0003E0: C123 10A7 C90D 1B33 001E 4CC0 00FF 1C40
0003F0: FF1B 5C09 FF02 0100 4D00 0000 0000 0000
```

✆, ein UD-Zeichen auf dem Drucker

Die meisten von uns haben sehr intelligente Drucker, mit denen sich allerhand anstellen läßt. Dennoch wird gerne alles mögliche dem Textprogramm überlassen. So sind beispielsweise die Benutzer von TSCRIPS unfehlbar am Telefon zu erkennen, das wohl zum Lieferumfang des Programms für graphikfähige Drucker gehört. Einzelne von uns haben sich immerhin die Mühe gemacht, dem Telefon mit Hilfe der §P-Option des Programms eine eigene Gestalt zu geben. Es geht auch anders.

Mit den meisten im Club vertretenen Druckern kann man einen eigenen Zeichensatz oder auch nur einzelne Zeichen selbst definieren (user defined characters, hier kurz UD-Zeichen genannt). Dazu wird der gerade angewählte Zeichensatz in ein User-RAM kopiert, anschließend kann man einzelne oder alle Zeichen mit eigenem Code modifizieren. Schließlich wird dieses User-RAM anstelle des EPROMs im Drucker selektiert. Nun steht der frei programmierte Zeichensatz zum Druck zur Verfügung.

Die Handbücher der Drucker geben sich alle Mühe, das Verfahren verständlich zu machen. Besser erklären kann ich es auch nicht. Aber vielleicht kann ich dem Leser Mut machen, es doch auch einmal zu versuchen. Es wäre wirklich schade, den teuren Drucker nicht voll auszureizen. Es lockt z. B. die Aussicht, auch ein eigenes Telefon für TSCRIPS zu kreieren und dabei nicht einmal einen Graphik-Zeichensatz laden zu müssen. Genau das habe ich auch getan und werde es jetzt beschreiben. Das Resultat war ein kleiner ✆.

Alles beginnt natürlich mit dem Entwurf des Zeichens. Die Abbildung zeigt den Telefonhörer (durchgezogene Linie) und den Versuch seiner digitalen Umsetzung (Punkte). Dabei sind in waagerechter Richtung je zwei Punkte für eine Linie vorgesehen, weil die Striche sonst zu dünn würden. Diese Punkte können nun als Bits etlicher Bytes aufgefaßt und in Hex oder wie auch immer notiert werden. Das Weitere ist von Drucker zu Drucker unterschiedlich. Hier wird der Vorgang für den NEC P6 erläutert.

Der P6 hat 24 Nadeln, die bei welcher Schrift auch immer, also auch bei UD-Zeichen, alle aktiviert werden. Deshalb sind zur Programmierung eines Zeichens in senkrechter Richtung pro Punktspalte drei Bytes erforderlich. In der Zeichnung trennen waagerechte Striche diese Drittel. Drucker mit 8 oder 9 Nadeln benötigen dafür nur ein Byte. Die Wertigkeit der Bits (Punkte) nimmt von oben nach unten ab, d. h. das höchstwertige Bit des jeweiligen Bytes entspricht dem oberen Punkt. Die Reihenfolge der drei Bytes ist oben-mitten-unten, die der Punktspalten geht von links nach rechts.

Die eigentliche Programmierung sollte mit einem Reset des Druckers beginnen, um etwa zuvor eingestellte und evtl. ungünstige Steuerungen zu löschen. Anschließend wird ein Zeichensatz selektiert, in diesem Fall die Proportionalschrift. Bei anderen Druckern (z. B. Gemini-10X) erübrigt sich das. Da hier nur ein einziges Zeichen neu definiert werden soll, wird das User-RAM sodann mit dem gerade selektierten Zeichensatz angefüllt. Sonst wäre es nämlich am Ende bis auf das eine neue Zeichen vollkommen leer, jeder Buchstabe würde als Blank wiedergegeben.

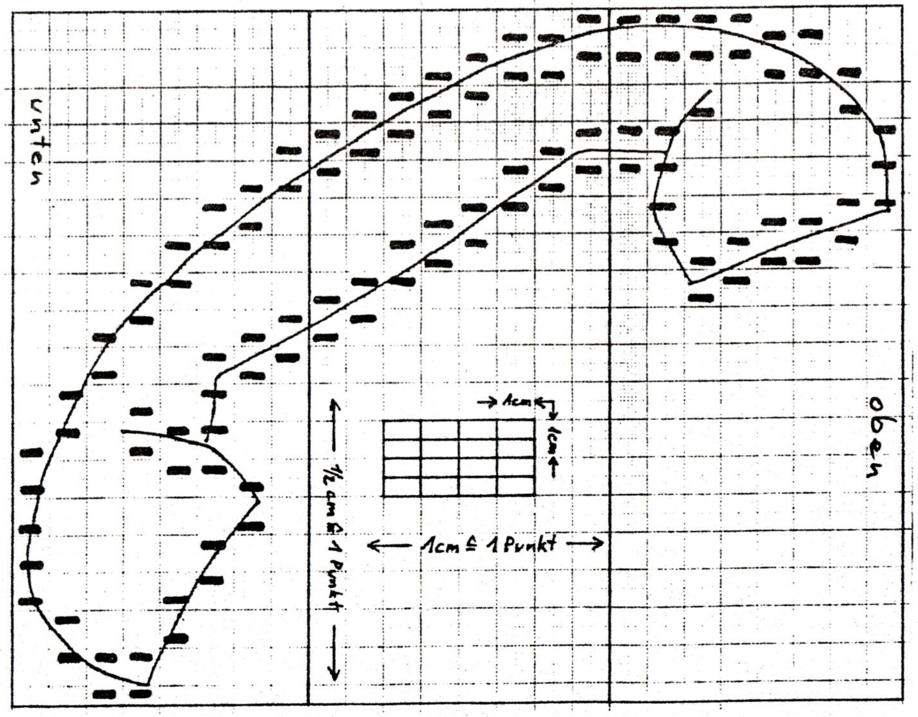
Schließlich kommt der Befehl zur Vorbereitung des User-RAMs. Er beinhaltet eine bestimmte Steuersequenz und den ASCII-Code des zu ändernden Zeichens. Beim NEC werden die Ober- und Untergrenze des ASCII-Bereiches eingegeben, der neu definiert werden soll. Da es hier nur um ein Zeichen geht, in diesem Falle 7Fh, sind Ober- und Untergrenze identisch. Zuletzt muß dem Drucker noch mitgeteilt werden, wie breit (in Punkten) das Zeichen werden soll. In diesem konkreten Beispiel beginnt es mit 2 leeren Dotspalten, fährt fort mit 37 Spalten, die Punkte enthalten werden und einer wieder leeren Spalte. Auch dies ist beim Gemini nicht erforderlich; lest euch das in Druckerhandbuch einfach durch.

Nun folgen in der langen Tabelle die Codes für die 37 Punktspalten, für jede drei Bytes. Wenn man sich diese Hexzahlen binär vorstellt, erkennt man in der Zeichnung die einzelnen Bits als gesetzte Punkte wieder. Weniger aufwendige Schriftarten als LQ PS erfordern weniger Code. Beim Gemini kostet es, soweit ich mich erinnere, sogar nur ca. 25-30 Bytes insgesamt. Die Auflösung der Skizze muß diesen Gegebenheiten selbstverständlich Rechnung tragen. Also weniger malen.

Bisher operierten wir noch immer vom druckerinternen Zeichensatz aus. Der neu gestaltete wird nun selektiert. Am Ende der Tabelle erscheint nun das Zeichen selbst, das wir natürlich sogleich auf dem Papier bewundern wollen. Ein nachfolgender Wagenrücklauf (0Dh) sorgt dafür, daß der Drucker sofort loslegt und nicht erst etliche Zeichen in seinem Puffer ansammelt.

Dieser ganze Klumpatsch von insgesamt 134 Bytes wird von einer kurzen Routine auf den Drucker ausgegeben, die beim Label start beginnt. Sie soll hier nicht näher erläutert werden. Wer einen Tandy hat, muß OUTI durch ein Kostrukt ersetzen, das anstelle des Ports FDh die Speicherstelle 37E8h bedient. Genauso einfach geht es in BASIC. Dazu verstaut man am besten alle erforderlichen Codes in DATA-Zeilen, die in einer READ-Schleife auf den Drucker ausgegeben werden.

Arnulf Sopp, ✆ 0451-791926



```
5200                ORG     5200h

5200 211052  start  LD      HL,telefon        ;Tabelle der Druckercodes
5203 01FD86         LD      BC,tabend-telefon*256+0fdh      ;B<-Länge
                                              ;C <- Druckerport FD
5206 CDD105  loop   CALL    05d1h             ;Druckerstatus lesen
5209 20FB           JR      NZ,loop           ;falls Dr. nicht bereit
520B EDA3           OUTI                      ;Byte ausgeb., Zeig. erh.
520D C8             RET     Z                 ;ins DOS, wenn fertig
520E 18F6           JR      loop              ;sonst weiter ausgeben

5210 1C      telefon DB     1ch,'$'           ;Reset für den Drucker
5212 1B             DB      1bh,'p',01h       ;Schriftart LQ PS
5215 1B             DB      1bh,':',00h,00h,00h   ;copy PS-font
521A 1B             DB      1bh,'&',00h       ;UD-Definition
521D 7F             DB      7fh,7fh           ;ASCII 7F programmieren
521F 02             DB      02h,25h,01h       ;2 + 37 + 1 Dots breit

5222 0F             DB      0fh,80h,00h       ;ab hier eigentl. Zeichen
5225 30             DB      30h,60h,00h
5228 0F             DB      0fh,90h,00h
522B 70             DB      70h,68h,00h
522E 00             DB      00h,14h,00h
5231 44             DB      44h,0ah,00h
5234 83             DB      83h,85h,00h
5237 00             DB      00h,42h,80h
523A 83             DB      83h,0a1h,00h
523D 00             DB      00h,40h,0c0h
5240 C2             DB      0c2h,30h,20h
5243 30             DB      30h,08h,40h
5246 4A             DB      4ah,14h,30h
5249 34             DB      34h,08h,00h
524C 08             DB      08h,06h,18h
524F 04             DB      04h,01h,00h
5252 00             DB      00h,02h,88h
5255 00             DB      00h,01h,44h
5258 00             DB      00h,00h,0a0h
525B 00             DB      00h,00h,44h
525E 00             DB      00h,00h,22h
5261 00             DB      00h,00h,08h
5264 00             DB      00h,00h,32h
5267 00             DB      00h,00h,09h
526A 00             DB      00h,00h,30h
526D 00             DB      00h,00h,41h
5270 00             DB      00h,00h,00h
5273 00             DB      00h,00h,41h
5276 00             DB      00h,00h,20h
5279 00             DB      00h,00h,01h
527C 00             DB      00h,00h,20h
527F 00             DB      00h,00h,11h
5282 00             DB      00h,00h,02h
5285 00             DB      00h,00h,10h
5288 00             DB      00h,00h,0eh
528B 00             DB      00h,00h,00h
528E 00             DB      00h,00h,0ch

5291 1B             DB      1bh,'%',01h       ;UD-Zeichensatz selekt.
5294 7F             DB      7fh,0dh           ;Zeichen drucken und Ende
             tabend

5200                END     start             ;dort Einsprung

00000 Fehler
```


SYSCOPY1 und RAMDISK für Helmut Bernhardts Banker

Ihr habt ja alle hoffentlich aufmerksam den Artikel über Helmuts Banker und das erste Programm dafür, RAMSYS von Arnulf, gelesen. Der Banker hat mich sofort fasziniert und so mußte er her. Leider stellte sich ein Problem: auf dem Genie IIIs gibt es die Programme SYSCOPY und MEMDISK, die aber mit dem Banker nicht laufen. SYSCOPY leistet Ähnliches wie RAMSYS, nur benötigt es absolut keine Systemdiskette mehr. Alle SYS-Files lassen sich aus dem Speicher laden. MEMDISK versorgt den Benutzer mit einer 128K-Disk als höchstes Laufwerk (vorhandene Laufwerke +1). Diese Ramdisk hat durch ihre Geschwindigkeit unheimliche Vorteile gegenüber den lahmen Diskettenlaufwerken.

Ich habe mir also erlaubt, auch ohne TCS oder Phoenix um Erlaubnis zu bitten, diese beiden Programme zu disassemblieren und auf Helmuts Banker umzustricken. Das ist nicht so problematisch, wenn man viel Zeit aufbringt und auf einige Disketten verzichten kann, die man beim Probieren zerschießt. Die Produkte heißen bei mir SYSCOPY1 und RAMDISK.

Im Gegensatz zu RAMSYS arbeitet SYSCOPY mit einem kleinen Directory, in das die SYS-Files beim Lesen eingetragen werden. Somit entfällt jegliche Änderung am Programm selber. Es dürfte mit allen Newdos/Gdos-Versionen laufen, solange SYS0 nicht stark verändert wurde, was kaum zu befürchten ist. Die SYS-Files liegen dann in Bank 1 und 2, wobei in Bank 2 je nach Länge/Anzahl der Files noch mehr oder minder viel Platz ist, den man später problemlos nutzen kann. In Bank 1 stehen auch die Service-Routinen. Nach dem Aufruf lädt SYSCOPY erst einige Sekunden seine SYS-Files und dann kann man die System-Diskette entfernen. Beim Kopieren wird zwar noch danach gefragt, aber es kann auch eine Pappscheibe in Drive 0 liegen, wenn die als Diskette anerkannt wird. Alle SYS-Files werden aus den Banks geladen.

RAMDISK benutzt einigen Platz in Bank 2, den SYSCOPY1 frei läßt. Als RAM-Disk dienen die Banks ab 3 aufwärts. Beim Aufruf kann man die RAM-Disk in 64K-Schritten (also jeweils 2 Banks) von 64K bis 896K einstellen (falls Euer Banker soweit bestückt ist). Wenn schon vorher einmal RAMDISK gestartet worden war (auch vor einem Reset), bleibt der alte RAM-Disk-Inhalt erhalten, wenn man nicht ausdrücklich eine Neu-Formatierung wünscht. Die RAM-Disk wird als Laufwerk 2 oder 3 angesprochen (beispielsweise läuft auch SUPERZAP!). Vor dem ersten Start muß man eine kleine Änderung am Source-Code vornehmen und das Programm neu assemblieren, weil jedes DOS (Newdos, Gdos, Hdos) einen anderen Initialisierungswert benötigt. Das dürfte aber nicht zu schwer sein.

SYSCOPY1 und RAMDISK sind über die Clubbibliothek des Club 80 erhältlich.

Gerald Schröder

* Mein 3. Assemblerprogramm *

Aller guten Dinge sind drei! - Also dürfte dies hier eigentlich nicht mehr von mir kommen! Aber es heißt ja:
Unter den Blinden ist der Einäugige König!
Womit ich nicht sagen will, daß ihr blind seid...
Vielmehr: daß ich nicht "zweiäugig" bin - was das auch immer heißen mag.
Kurzum: Für die "Sehenden" ist dies hier wiederum nicht geschrieben. Sondern für diejenigen, die gern mal "hinter die Kulissen" schielen möchten - falls sie nicht bereits zu den Drahtziehern gehören, die die Puppen tanzen lassen.

Dank schulde ich zween, die mir beim Drahtziehen halfen: den Herren *Schröder* und *Sopp* (in alphabetischer Folge) oder auch den hilfreichen Kollegen *Arnulf* und *Gerald* (ebenfalls in alphabetischer Reihenfolge!)
Nicht, daß sie das nachstehende Programm für mich oder mit mir gebastelt hätten - das wäre zu einfach für mich gewesen! Es ist schon eine eigene Creation (wie man unschwer an seinem hochgeistigen Inhalt erkennt...). Aber sie erklärten mir in gewohnt selbstloser und hingebender Weise einige Tricks/Drehs/Wendungen/Kniffe, die ich - wieder einmal! - in keinem Leerbuch fand!
Sicher ist auch dieses nun schon etwas längere Programm möglicherweise noch ungeschickt. Aber ich kann nur wieder sagen:
Es läuft!
Also ist es nicht falsch. Höchstens noch nicht rationell oder tricky.
Nach meinem ersten Programm (vorletztes INFO) mit fest vorgegebener MM (Bildschirm-Meldung oder auch "Monitor Message") und meinem zweiten (letztes INFO) mit MM (Mathematischer Musik oder auch Musikalischer Mathematik) widmet dieses dritte sich nun der vornehmsten Aufgabe eines Computers: mit seinem Herrchen - oft "USER" genannt, was wohl soviel heißt wie
"Unermüdliches Spielen Erzeugt Routine"
zu sprechen! Was auch als
Dialog
bezeichnet wird. Er ist zwar kurz und bündig. Aber ich hoffe, er zeigt denjenigen, die wie ich Anfänger/Einsteiger sind, wie man ihn grundsätzlich aufbaut. Lange, lange habe ich danach gesucht. Obgleich es nach meiner Meinung die wichtigste Aufgabe ist, die man einem Computer anvertrauen kann, steht dafür nirgends ein praktisches Beispiel - weder im berühmten *R.ZAKS* noch sonstwo (wer mir das Gegenteil mitteilt, sei gegrüßt und bedankt!)
Immerhin, eine gute Hilfe war mir
Craig A. Lindley, "TRS-80/Z80 Assembly Language Library"
Wayne Green Publications 1983, ISBN 0-88006-060-3. („Nur" rd. DM 130)

Das Programm greift auf die LDIR-Funktion sowie auf die Bedeutung des $-Zeichens zurück; beides habe ich schon im ersten Beitrag erklärt. Im übrigen habe ich diesmal versucht, es besser zu kommentieren, nachdem mich *Arnulf* mit Recht gerügt, daß ihr ja nicht lachen, sondern lernen wolltet! Wahrscheinlich hat bisher "keiner weder gelacht noch gelernt"! (Ich hoffe, ihr erkennt die doppelte Verneinung, sonst bringe ich das nächste Mal einen Kursus über den Umgang mit logischen Operatoren. Bitte melden!)

Zwei Fehler aus den früheren Beiträgen habe ich zu korrigieren:
"OPCODE" und *"OBJECT CODE"* ist natürlich nicht dasselbe:

OPCODE = Operation Code = Befehlscode = der Anweisungsteil, mit dem jeder Befehl an die CPU beginnt; ihm folgt der "Operandenteil", d.i. der "Befehlsempfänger" (sozusagen der "Thumbe Thor", heute auch "G.I." oder "Sad Sack" genannt), der dem Befehl gehorchen muß (manchmal ist da auch gar keiner, der zu gehorchen hat; dann kommandiert der UvD sich selbst!)
Wenn sämtliche OPCODES+OPER(AND)IERTEN, die den Source- (Quell)-Code darstellen, "assembliert" sind, haben wir endlich den *OBJECT CODE*, der nur noch aus Sedezimalzahlen besteht, die die verkürzte Form der eigentlichen BIT-Konfigurationen des Maschinen-Codes darstellen.

Und die zweite Korrektur:
Speicheradressen-Definitionen, die durch die Pseudo-Anweisungen (jetzt wissen wir ja, was das ist)
DEFB = Define Byte
DEFW = Define Word = 2 Bytes
DEFM = Define Message
DEFS = Define Storage (wie in vielen sogenannten Leerbüchern zu lesen steht; ich bin aber mit Arnulf der Meinung, daß die Erklärung "Define Space" logischer ist)
gesetzt werden, an denen das Definierte beginnt, dürfen nie vor, und solche, die einen konstanten Wert (z.B. eine ROM-Adresse) festlegen - wie die *EQU*-Gleichsetzungen - dürfen oft nicht hinter der *ORG*-Anweisung (auch eine Pseudo-Anweisung) stehen! Aus Fehlern wird man klug (ich müßte infolgedessn inzwischen schon sehr klug sein; stimmt aber nicht: folglich habe ich noch viele Fehler vor mir! Merkt euch das und seid vorsichtig!)

Das Wichtigste in diesem Musterprogramm sind die drei Zeilen 100, 110 und 130. Sie zeigen den Rückgriff auf ROM-Routinen!
Das englische Wort "idle" ist verwandt mit "eitel" und bedeutet "faul". Je fauler der Programmierer ist, um so mehr überläßt er die Arbeit den Routinen oder Unterprogrammen, die fertig im ROM vorliegen ("gewußt, wo!" Haha!-Siehe "ROM-Listing" von Röckrath -
kommt Röckrath,
kommt ROM-Rat!
- und um so "eitler" gefällt er sich in diesem (Nichts-)Tun!
Natürlich ist das nicht eitel, sondern "edel".

Mindestens genau so edel ist der Zugriff auf DOS-Routinen. Manche meinen, noch edler. Hier hilft das DOS-Buch von Grosser (bei Röckrath) weiter (das allerdings nur ohne den obligaten Bindestrich zu haben ist, genau wie das "Maschinen Sprache Buch" von Röckrath; Maschinensprache ist eben Maschinensprache, nicht Deutsch, ganz klar!)

Doch hierüber ein andermal.
Vielleicht.
Für heute schließe ich. Egal:
Es reicht.

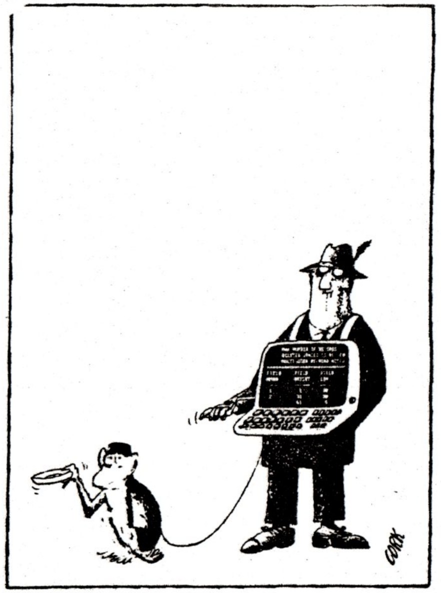
KaJot

„Mein 3. Assembler-Programm"

```
                 00010 ; DIALOG MIT DEM ORAKEL ZU WEINHEIM
                 00030 ; VON KAJOT, GESTARTET 28.10.86
                 00040 ; KEINE RECHTE ANGEMELDET!
                 00045 ; NACHAHMUNG EMPFOHLEN !
                 00050 ;
                 00060 ; ZUERST WERDEN 3 ROM-ROUTINEN DEFINIERT:
                 00070 ;
01C9             00100 CLS      EQU     01C9H    ;CLEAR SCREEN
0049             00110 INCHW    EQU     0049H    ; 1 ZEICHEN EINGEBEN!
05D9             00130 INBUFF   EQU     05D9H    ; 1 ZEILE EINGEBEN!
                 00133 ;
6000             00136          ORG     6000H    ; HIER SPEICHER-ANFANG
                 00140 ;
                 00150 ;DEFINITION VON CODE-WOERTERN = SPEICHERADRESSEN
                 00160 ;
6000 57          00170 FRAGE1   DEFM    'WILLST DU VERREISEN <J/N> ?'
601B 4F          00210 ANTW1    DEFM    'OK - BLEIBE HIER UND ASSEMBLIERE!'
603C 57          00230 FRAGE2   DEFM    'WOHIN?'
6042 42          00270 ANTW2    DEFM    'BLEIB LIEBER ZUHAUS UND LERNE WAS!'
                 00290 ;
                 00300 ; NACH DIESEN VORBEREITUNGEN BEGINNT
                 00305 ; DAS EIGENTLICHE PROGRAMM
                 00310 ;
6064 CDC901      00330 START    CALL    CLS      ; KLAR SCHIFF!
6067 011B00      00340          LD      BC,ANTW1-FRAGE1; LAENGE DER 1.FRAGE
606A 114A3C      00350          LD      DE,3C4AH; SIE SOLL IN DIE 2.ZEILE
606D 210060      00360          LD      HL,FRAGE1; ADRESSE, WO DIE FRAGE
                 00370                            ; BEGINNT, WIRD GELADEN
6070 EDB0        00380          LDIR              ; TRANSPORT DER FRAGE
                 00390                            ; ZUM BILDSCHIRM
                 00400 ;
                 00410 ; DIE ANTWORT <J ODER N> WIRD ERWARTET,
                 00420 ;BEI VERNEINUNG FOLGT VERABSCHIEDUNG,
                 00430 ; BEI BEJAHUNG ERFOLGT WEITERE FRAGE -
                 00440 ; NUR MIT GROSSBUCHSTABEN, ALSO
                 00450 ; "UPPER CASE" =APPEN(ZELLER) KAS ANTWÒRTEN!
                 00460 ;
6072 CD4900      00470 ANTWJN   CALL    INCHW    ;EQU SIEHE OBEN!
6075 CBAF        00480          RES     5,A      ;BIT 5 ZURUECKSETZEN, FALLS
                 00490                           ;MIT KLEINBUCHSTABEN (LC)
                 00500                           ;GEANTWORTET WURDE
6077 FE4A        00510          CP      'J'      ; LAUTETE DIE ANTWORT 'J'?
6079 2815        00520          JR      Z,FRAGE ;FALLS JA, ZUR 2.FRAGE!
607B FE4E        00530          CP      'N'      ; ODER LAUTETE SIE 'N'?
607D 20F3        00540          JR      NZ,ANTWJN; FALLS AUCH NICHT, DANN
                 00550                           ; WIEDERHOLE DIE ANTWORT!
607F 012100      00560          LD      BC,FRAGE2-ANTW1; FALLS NEIN, FOLGEN
                 00562                        ; ABSCHIEDSWORTE, DEREN LAENGE
                 00564                        ; ERGIBT SICH AUS DER DIFFERENZ
                 00566                        ; DER ZEILEN 'FRAGE2' - 'ANTW1'
6082 118A3C      00570          LD      DE,3C8AH; SIE STEHN GENAU DARUNTER
6085 211B60      00580          LD      HL,ANTW1; SIEHE DIE MESSAGE OBEN
6088 EDB0        00590          LDIR             ; LDIR SCHREIBT'S HIN
608A CD4900      00595          CALL    INCHW    ; AUSSTIEG MIT ENTER!
608D C32D40      00600          JP      402DH    ; ZURUECK INS DOS - ENDE!
                 00610 ;
                 00620 ; FALLS DU VERREISEN WILLST, WIRST DU GEFRAGT:
                 00625 ;          'WOHIN?'
                 00630 ;
6090 010600      00640 FRAGE    LD      BC,ANTW2-FRAGE2; LAENGE VON 'WOHIN?'
                 00645  ; ERKLAERUNG ANALOG ZEILE 564!
6093 118A3C      00650          LD      DE,3C8AH ; SIE SOLL IN DIE 3.ZEILE
6096 213C60      00660          LD      HL,FRAGE2; "WOHIN?"
6099 EDB0        00670          LDIR              ; LDIR SCHREIBT'S HIN!
```



```
                 00680 ;
                 00690 ; NUN ANTWORTE! ES WERDEN HIERFUER 48 STELLEN
                 00700 ; ZUR VERFUEGUNG GESTELLT (D.I.DER REST DER ZEILE)
                 00710 ;
609B 0630        00720          LD      B,30H    ; 30H = STELLENZAHL
609D 21913C      00723          LD      HL,3C91H; CURSOR-POSITION FUER
60A0 222040      00726          LD      (4020H),HL      ; DIE ANTWORT
60A3 210042      00730          LD      HL,4200H; PUFFER F.D.ANTWORT
60A6 CDD905      00740          CALL    INBUFF   ; EQU SIEHE OBEN
                 00750 ;
                 00760 ; DAS ORAKEL ZU WEINHEIM ANTWORTET:
                 00770 ;
60A9 012200      00780          LD      BC,START-ANTW2; LAENGE DER ANTWORT
60AC 110A3D      00790          LD      DE,3D0AH; ZEIG SIE 2 ZEILEN TIEFER
60AF 214260      00800          LD      HL,ANTW2; SIEHE MESSAGE OBEN
60B2 EDB0        00810          LDIR             ; ZUM BILDSCHIRM
60B4 CD4900      00815          CALL    INCHW    ; AUSSTIEG MIT ENTER!
60B7 C9          00820          RET              ; ZURUECK ZUM AUFRUFER!
6064             00850          END     START
00000 TOTAL ERRORS
32432  TEXT AREA BYTES LEFT

ANTW1  601B 00210   00340 00560 00580
ANTW2  6042 00270   00640 00780 00800
ANTWJN 6072 00470   00540
CLS    01C9 00100   00330
FRAGE  6090 00640   00520
FRAGE1 6000 00170   00340 00360
FRAGE2 603C 00230   00560 00640 00660
INBUFF 05D9 00130   00740
INCHW  0049 00110   00470 00595 00815
START  6064 00330   00780 00850
```



Liebe Clubfreunde,
hier folgt der Rest der RPNL-Beschreibung. Die nachstehende Liste enthält eine nach Funktionen sortierte Aufstellung aller in RPNL vorkommenden Anweisungen.

```
**************************
*  Summary  of  Commands  *
**************************
RPNL    Vers.  2.0     1986
```

In dieser Kurzanleitung werden für Daten folgende Abkürzungen benutzt:

n	für eine 16bit Zahl
d	für eine 8bit Zahl
adr	für eine 16bit Adresse
flag	für logisches Flag
name	für eine 16bit Buffer-Adresse

Logische Befehle:

XOR	(n1 n2	-> n3	) Logisches XOR
OR	(n1 n2	-> n3	) Logisches OR
AND	(n1 n2	-> n3	) Logisches AND
NOT	(n1	-> -n1	) Logische Negation
TRUE	(	-> flag	) Liefert 0FFFFH als Wahr-Flag
FALSE	(	-> flag	) Liefert 0000H als Falsch-Flag
=	(n1 n2	-> flag	) Vergleich zweier Zahlen
0=	(n1	-> flag	) Test auf n1 gleich 0
<	(n1 n2	-> flag	) Kleiner als
<=	(n1 n2	-> flag	) Kleiner gleich
>	(n1 n2	-> flag	) Größer
>=	(n1 n2	-> flag	) Größer gleich
<>	(n1 n2	-> flag	) Ungleich
=(S)	(adr1 adr2	-> flag	) String-Vergleich
>=(S)	(adr1 adr2	-> flag	) String-Vergleich

Arithmetische Befehle:

+	(n1 n2	-> n3	) Addition zweier Zahlen
-	(n1 n2	-> n3	) Subtraktion zweier Zahlen
/	(n1 n2	-> n3	) Division zweier Zahlen
*	(n1 n2	-> n3	) Multiplikation zweier Zahlen
MOD	(n1 n2	-> n3	) Liefert den Divsionsrest
/MOD	(n1 n2	-> n3 n4	) Liefert das Divisionsergebnis und den Rest (n4)
RND	(	-> n1	) Liefert eine Zufallszahl zwischen 0...65535

Stack-Befehle:

DROP	(n1	->	) Vernichtet den TOS
SWAP	(n1 n2	-> n2 n1	) Vertauscht TOS & NOS
OVER	(n1 n2	->n1 n2 n1	) Copiert NOS zum TOS
ROT	(n1 n2 n3	->n2 n3 n1	) Rotiert n1 zum TOS
INC	(n1	-> n1+1	) Erhöht TOS um 1
DEC	(n1	-> n1-1	) Erniedrigt TOS um 1

Zuweisungs-/Lade-Befehle:

:=	(n1 adr	->	) Legt n1 unter adr ab
:=(B)	(d1 adr	->	) Legt d1 unter adr ab
?	(adr	-> n1	) Holt den Wert von adr
?(B)	(adr	-> d1	) Holt Byte von adr

I/O-Bfehle:

INCHAR	(	-> d1	) Holt ein Zeichen von der Tastatur (ohne zu warten !)
READ(S)	(adr	->	) liest einen String nach adr von d. Tastatur
READ(CH)	(adr	->	) Liest einen String nach adr von d. Tastatur
OUTCHAR	(d1	->	) Gibt ein Zeichen aus
PRINT	(adr	->	) Gibt die unter adr befindliche Zahl aus
PRINT(S)	(adr	->	) Gibt den unter adr befindlichen String aus

I/O-Steuerbefehle:

SCROFF	(	->	) Schaltet die Bildschirmausgabe ab
SCRON	(	->	) Schaltet die Bildschirmausgabe wieder ein
DROFF	(	->	) Schaltet die Druckerausausgabe ab
DRON	(	->	) Schaltet die Druckerausausgabe ein
CLS	(	->	) Löscht den Bildschirm
PAGE	(	->	) erzeugt einen Seitenvorschub auf dem Drucker
CR	(	->	) Erzeugt einen Zeilenvorschub

System-Variable:

SYSFIELD	(	-> adr	) Liefert adr auf den ersten freien Speicherplatz am Programmende
ZEICHEN	(	->	) Enthält das Zeichen nach einem INCHAR-Befehl

Disk-Befehle:

READF	(name -> n1 flag)	Liest ein byte von d. Disk
WRITEF	(n1 name -> flag)	Schreibt ein byte auf Disk
OPEN(I)	(name -> flag)	Eröffnet File zum Lesen
OPEN(O)	(name -> flag)	Eröffnet File zum Schreiben
OPEN(R)	(name -> flag)	Eröffnet Schreib/Lese-File
CLOSE	(name ->)	Schließt einen File
DELETE	(name ->)	Löscht einen eröffneten File
EOF?	(name -> flag)	Prüft auf END-OF-FILE
FILESPEC	(adr name -> flag)	Weist einem geschlossenen File eine neue Filespec zu

System-Befehle:

NEW	(->)	Löscht das programm u. initialisiert den Compiler neu
SYSTEM	(->)	Zurück zum DOS
NOFILES	(->)	Löscht alle Diskbefehle aus dem Runtime-Modul
LIST	(->)	Listet alle bisher comp. Worte
MEMORY	(->)	Liefert den freien Speicherplatz
SAVE	(filespec->)	Speichert das Prgramm auf Disk
COMPILE	(filespec->)	Compiliert ein Programm
FILES	(->)	Listet das Directory

Compiler-Anweisungen:

CALL	Leitet Compilierung von Hex-Inlinecode ein für Unter-Programme
CEND	Beendet CALL-Anweisung
CODE	Leitet Compilierung von Hex-Inlinecode für in Assembler definierte RPNL-Worte ein
CEND	Beendet CODE-Anweisung
DECLARE	Eröffnet die Compilierung von Variablen, Konstanten, und dergleichen
DEND	Beendet DECLARE-Anweisung
PROGRAM	Leitet die Compilierung von High-Level Code ein
END	Beendet die PROGRAM-Anweisung

Schleifen-StrucktTuren:

FOR	(n2 n1 ->)	Initialisiert die FOR-Loop
I	(-> n1)	liefert den Laufindex der aktuellen. FOR-Loop auf den Stack
LEAVE	(->)	Bewirkt vorzeitigen Abruch der aktuellen FOR-Loop
LOOP	(->)	Schließt die FOR-Loop ab
REPEAT	(->)	Beginn der Unbedingten Wiederholungsschleife
UNTIL	(flag ->)	Prüft, ob Aussprungbedingung erfüllt ist
LOOP	(->)	Schließt die REPEAT-Loop

Kontroll-Strucktturen:

IF	(flag ->)	Beginn der Bedingungsanweisung
ELSE	(->)	Ende des TRUE-Pfades
ENDIF	(->)	Ende der Bedingungsanweisung und/oder des TRUE/FALSE-Pfades

***

TANDY . . . better again

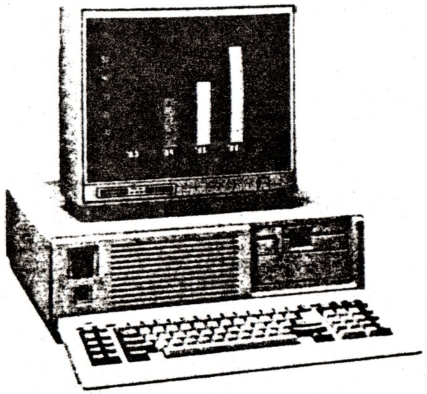
Unter diesem Motto stand eine Pressekonferenz am 5.9.86 in München, in der TANDY seine neuen Rechner vorstellte. Diese sind nun auch voll auf den Markt-Standard der Kompatibilität mit IBM eingestellt.

U. Vanden Bosche, TANDY Marketing-Direktor für Europa, stellte erhöhte Anstrengungen in Aussicht, auch in Europa den Verkauf von IBM kompatiblen Rechnern zu steigern, nachdem TANDY in den USA zum größten Hersteller von IBM kompatiblen Rechnern geworden ist. Um dieses Ziel zu erreichen, sollen die TANDY Computer Shops modernisiert werden und es soll eine umfassende Schulung der Mitarbeiter erfolgen.

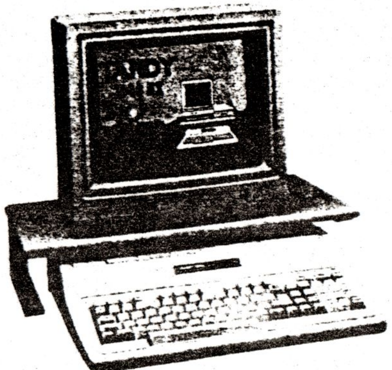
Das Flagschiff der neuen Serie ist der 3000 HD 40. Die technischen Daten:

- INTEL 80286 CPU
- 8 MHz Takt
- 10 Erweiterungsplätze
  - 3 für 8-bit XT
  - 7 für 16-bit AT
- MS DOS 3.2 oder auf Wunsch XENIX, System 5
- 1 5 1/4 Zoll Laufwerk mit 1.2 MB Kapazität
- 1 40 MB Harddisk.

Dieser Rechner ist für Mehrplatzsysteme geeignet und TANDY wird auch in Kürze ein eigenes LAN, genannt ViaNet auf den Markt bringen.



3000 HD 40



1000 EX

Ein weiteres Mitglied der Rechnerfamilie ist der 3000 HL. Dies ist ein XT mit AT Qualitäten. Auch er besitzt eine 80286 CPU mit 8 MHz. Der Speicher ist durch Zusatzplatinen auf 4 MB aufrüstbar. Die anderen Daten entsprechen denen eines XT.

Der 1000 SX ist XT kompatibel, seine Taktfrequenz kann aber zwischen 4.77 MHz und 7.1 MHz umgeschaltet werden. Damit wird eine höhere Rechenleistung erzielt.

Nach unten wird die Palette mit dem 1000 EX abgerundet. Dieser Rechner ist von der Software IBM kompatibel, es wird MS DOS 2.11 verwendet. Um aber eine kompaktere Bauweise, und somit ein kleineres, handlicheres Gerät zu erzielen, ist die Hardware nicht IBM kompatibel.

Die technischen Daten dieses Rechners sind:

- 8088 CPU
- 4.77 oder 7.1 MHz Takt
- 256k, ausbaubar bis 640k
- MS DOS 2.11

Dieser Rechner wird für 1.750,– DM incl. Mehrwertsteuer, einschließlich einem s/w-Monitor angeboten.

Natürlich besitzen alle Rechner eine deutsche Tastatur und einen parallelen Druckerausgang. Mit jedem Rechner wird das Programmpaket DeskMate II mitgeliefert, in das eine Textverarbeitung, eine Tabellenkalkulation, eine Datenbank, sowie eine Mailbox und ein Kalender integriert sind.

***

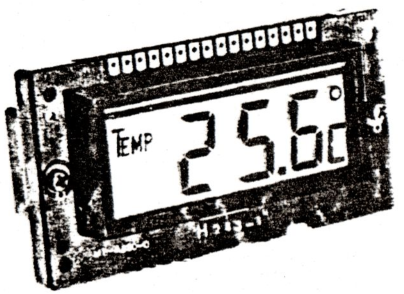
Hardwaretips: Digitales Temperatur- Zeitmodul

Nicht vorenthalten wollte ich euch das Datenblatt eines von mir auf der Hobbytronik erworbenen Uhrenmoduls, von dem ich glaube, daß es für alle Clubmitglieder interessant ist.

Das Modul ist sehr universell einsetzbar (Heizungsregelung, Innen/Außenthermometer incl. Uhr für Heim und Auto, Eiswarner usw.) und kann sogar die Temperatur, über eine serielle Schnittstelle, an Computer übergeben (ob das auch mit der Uhrzeit funktioniert habe ich noch nicht ausprobiert).

Zweiundzwanzig Märker kostete das Wunderdingchen auf dem CONRAD-Messestand, viel "teurer" kann es wohl auch im Versand nicht sein.

Hartmut Obermann



Digitales Temperatur-Zeitmodul
Best.-Nr. 19 55 88

CONRAD

Technische Daten:

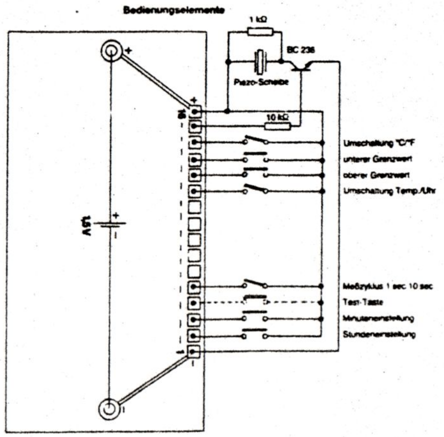
Meßbereich:	-20°C bis +70°C (0°F bis 160°F)
Auflösung:	0,1°C (F)
Genauigkeit:	±1°C (−20°C bis +70°C)
Meßrate:	10 Sekunden oder 1 Sekunde
Alarmausgang:	Intervalle 4 kHz - Dauer 6 Sekunden
Schaltausgang:	Oberer Grenzwert - Dauer 1 Minute Unterer Grenzwert - Dauer 1 Minute Oberer/unterer Grenzwert - Dauer 1 Sek.
Grenzwerteinstellung:	1°C-Schritte
Uhrenbetrieb:	12-Stunden-Anzeige
Betriebsspannung:	1,5 V (1,25 - 1,65 V)
Stromaufnahme:	ca. 3 µA

Externe Beschaltung:

Anschlußbelegung
1: Masse (Minuspol der Versorgungsspannung)
2: Stundeneinstellung
3: Minuteneinstellung
4: Testbetrieb (alle Segmente des Displays werden aktiviert)
5: Meßraten-Umschaltung - Bei geschlossenem Schalter wird die Temperatur im Abstand von 1 Sekunde und bei offenem Schalter alle 10 Sekunden ermittelt. Beachten Sie, daß die Grenzwerteinstellung nur im 10-Sekunden-Betrieb möglich ist.
6: Schaltausgang für den oberen Grenzwert. An diesem Ausgang liegt für die Dauer der Grenzwertüberschreitung - mindestens aber 1 Minute - „high"-Pegel (+1,5 V) an.
7: Schaltausgang für den unteren Grenzwert. An diesem Ausgang liegt für die Dauer der Grenzwertunterschreitung - mindestens aber für 1 Minute - „high"-Pegel (+1,5 V) an.
8: Steuerausgang für den oberen und unteren Grenzwert. An diesem Ausgang liegt bei Erreichen des oberen oder unteren Grenzwertes für die Dauer einer Sekunde „high"-Pegel an.
9/10: Serielles Interface - Pin 9 ist der Datenausgang, an Pin 10 liegen die Clock-Impulse an.

CONRAD

11: Temperatur-Uhrzeit-Umschaltung: Bei offenem Schalter ist das Modul im Temperaturmodus, während bei geschlossenem Schalter die Uhrzeit angezeigt wird.
12: Unterer Grenzwert
13: Oberer Grenzwert
14: Umschaltung °C oder °F: Bei offenem Schalter erfolgt die Temperaturanzeige in °Celsius, während bei geschlossenem Schalter °Fahrenheit angezeigt werden.
15: Alarmausgang: Bei Grenzwertüberschreitung liegt an diesem Ausgang für die Dauer von 6 Sekunden ein Intervallton mit einer Frequenz von 4 kHz an.
16: Betriebsspannungsanschluß (+1,5 V)



CONRAD

CLUB 80 HARD

CLUB 80 HARD

Bedienung:

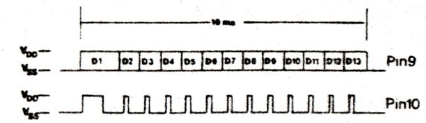
Uhrzeiteinstellung:
1. Schalten Sie das Modul auf Uhrzeitanzeige (Schalter an Pin 11 geschlossen).
2. Drücken Sie die Minutentaste (Pin 3) für ca. 2 Sekunden bis der Doppelpunkt nicht mehr blinkt. Bei jedem Tastendruck wird die Uhr um eine Minute vorgestellt. Andauerndes Drücken bewirkt ein schnelles Verstellen mit einer Frequenz von 8 Hz.
3. Das Einstellen der Stunden erfolgt in gleicher Weise durch Betätigung der an Pin 2 angeschlossenen Taste.

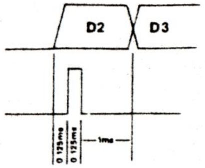
Temperaturbetrieb:

Temperaturbetrieb:
1. Schalten Sie das Modul auf Temperaturbetrieb (Schalter an Pin 11 offen).
2. Schalten Sie auf Meßrate 10 Sekunden (Schalter an Pin 4 offen).
3. Drücken Sie die an Pin 13 angeschlossene Taste. Im Display erscheint der untere Grenzwert. Drücken Sie gleichzeitig die Minutentaste, um den unteren Grenzwert einzustellen.
4. Die Einstellung des oberen Grenzwertes erfolgt in gleicher Weise durch Betätigung der an Pin 12 angeschlossenen Taste in Verbindung mit der Minuteneinstelltaste.
5. Bei aktiviertem Grenzwert erscheint Hi und (oder) Lo in der Anzeige. Der Grenzwertalarm kann unabhängig für Hi oder Lo stillgelegt werden, indem Sie gleichzeitig die entsprechende Grenzwerttaste und die Stundentaste drücken. Das Symbol (Hi oder Lo) wird dann im Display ausgeblendet.
Beachten Sie, daß die Grenzwerteinstellung nur möglich ist, wenn das Modul auf Temperaturanzeige mit 10-Sekunden-Abtastrate geschaltet ist. Die Alarmausgänge funktionieren bei beiden Abtastraten (1 Sek/10 Sek), sie funktionieren nicht, wenn das Modul auf Uhrenbetrieb geschaltet ist.

CONRAD

Serieller Datenausgang (Pin 9/10):





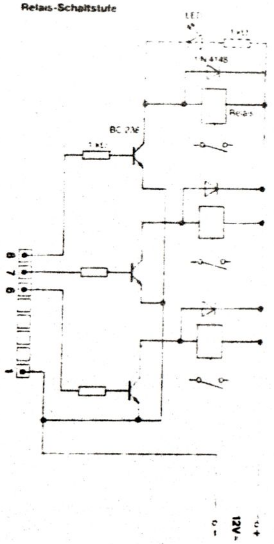
Bit-Tabelle

D_1	Hunderterstelle bei Fahrenheitanzeige. Vorzeichen bei Celsiusanzeige (Minus = high, Plus = low)	
D_2	8	Zehnerstelle
D_3	4	
D_4	2	
D_5	1	
D_6	8	Einerstelle
D_7	4	
D_8	2	
D_9	1	
D_{10}	8	1/10-Stelle
D_{11}	4	
D_{12}	2	
D_{13}	1	

Alarm- und Schaltausgänge:

Alarmausgang: An Pin 15 liegt ein Alarmsignal an, welches bei Erreichen des oberen oder unteren Grenzwertes aktiviert wird. Das Signal steht für die Dauer von 6 Sekunden an und besteht aus einem Intervallton mit einer Frequenz von 4 kHz. Über einen Schalttransistor kann sehr einfach ein piezokeramischer Schallwandler oder ein Miniaturlautsprecher (Kopfhörerkapsel 32 Ω) angesteuert werden.
Schaltausgang: An den Anschlüssen 6 und 7 liegt „high"-Pegel an, wenn die Temperatur den oberen oder unteren Grenzwert erreicht. Die Ausgänge sind solange „high", wie die Grenzwertüberschreitung andauert, mindestens aber für 1 Minute. Beachten Sie, daß diese Mindestschaltzeit auch erhalten bleibt, wenn die tatsächliche Temperatur bereits den entgegengesetzten Grenzwert erreicht hat. Die Schaltzustandsänderung erfolgt erst nach Ablauf dieser Periode. Am Modulanschluß 8 liegt „high"-Pegel an, wenn der untere oder obere Grenzwert erreicht wird. Im Gegensatz zu vorher beträgt die Dauer nur ca. 1 Sekunde.

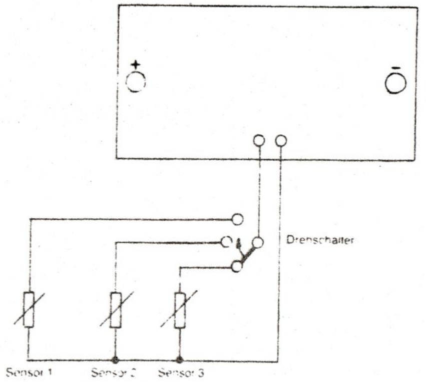
Relais-Schaltstufe



Meßstellenumschaltung:

Mit einem einpoligem Stufenschalter mit entsprechender Stufenzahl können beliebig viele Meßstellen angewählt werden. Pro Meßstelle ist ein eigener Sensor (Best.-Nr. 19 55 96) notwendig. Die Polarität der einzelnen Sensoren braucht nicht beachtet werden. Ebenso ist die Leitungslänge nicht kritisch. Der Leitungswiderstand darf jedoch 30 Ω nicht übersteigen. Bei ungünstiger Leitungsführung (parallel zum Lichtnetz) muß als Sensorleitung ein abgeschirmtes Kabel verwendet werden. Die Abschirmung wird dann mit Masse (Pin 1) verbunden.

Schaltbeispiel für drei Meßstellen



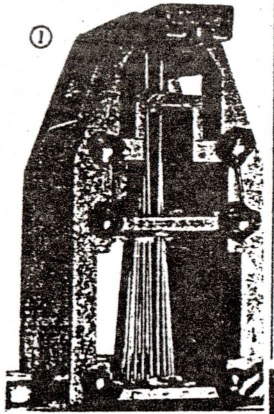
Technische Änderungen vorbehalten!
Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen gleich welcher Art, ob Fotokopie, Mikrofilme oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung durch CONRAD ELECTRONIC GmbH.
© Copyright 1985 by CONRAD ELECTRONIC GmbH, 8452 Hirschau

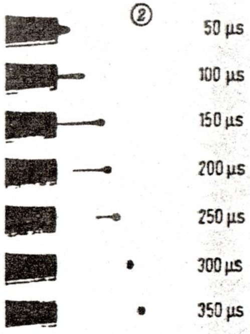
Computerlexikon

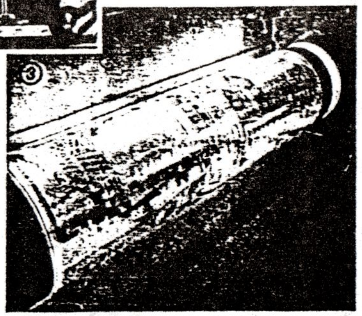
DRUCKER

Ob sie nun hämmern, nadeln, spucken oder mit einem Laserstrahl schreiben, das ist mehr eine Frage der Schönschreibansprüche an einen Drucker. Dagegen ist der Drucker selbst nahezu ein Muß für jeden Computerbesitzer. Will man den Inhalt eines Bildschirms oder gar eine ganze Datei zu Papier bringen, wird ein Drucker unerläßlich. Wenn die Daten schon im Computer stecken, wäre es ja ein Wahnsinn, nun vom Bildschirm alles mit einer Schreibmaschine abzutippen. Das kann der Computer besser und schneller, ohne daß sich neue Fehler einschleichen. Die Daten zwischen Computer und Drucker werden dann seriell oder parallel übertragen. Bei der seriellen Übertragung »gehen« die Bits vom Computer zum Drucker über eine Leitung wie im Gänsemarsch. Bei der schnelleren parallelen Übertragung stehen dagegen acht Leitungen zur Verfügung.

Gleichzeitig werden acht Bit nebeneinander übertragen, was einem Byte, also einem ASCII-Zeichen entspricht. Dann rattern die Typenräder oder Kugelköpfe wie bei einer Schreibmaschine los. Bei Nadeldruckern dagegen wird jeder Buchstabe aus Punkten zusammengesetzt, die das Nadelpaket des Druckkopfes blitzschnell zusammenstellt. Ebenfalls aus Punkten zusammengesetzt werden die Buchstaben bei einem Tintenstrahldrucker. Hier spuckt ein Satz Düsen winzige Farbtropfen gezielt aufs Papier. Neueste und schnellste Gruppe unter den Druckern sind die Laserprinter. Lautlos und fast mit Druckqualität spucken sie bis zu acht vollgeschriebene Seiten pro Minute aus.







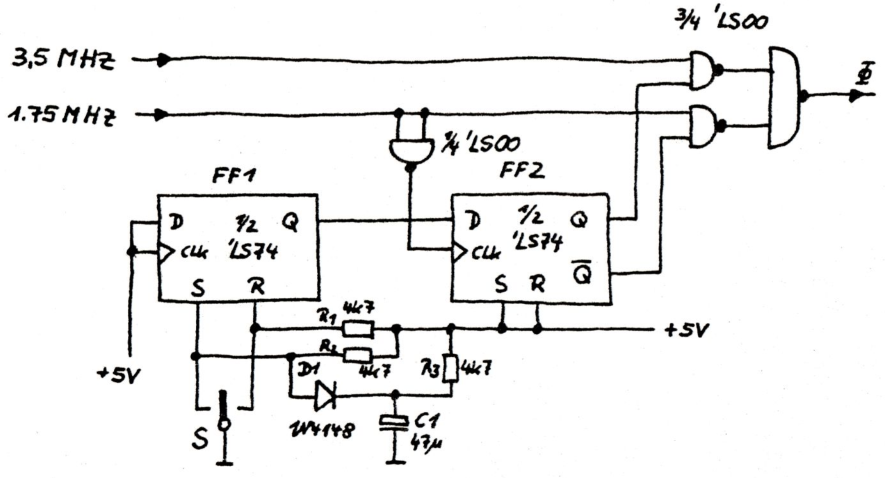
Nadeldrucker (1): Ein Bündel Stifte punktet die Buchstaben. Tintenstrahldrucker (2): Eine Düse schleudert Tintentropfen. Walze eines Laserdruckers (3): Sie ähnelt der eines Fotokopierers.

Taktumschaltung
Die allerletzte Lösung

Wer seinen Rechner bereits mit allerlei nützlichen Dingen vollgestopft hat, der wird es kaum begrüßen, den ohnehin recht ansehnlichen Kabelverhau unnütz vergrößern zu müßen. Was allso tun ?. Zuerst einmal ist festzustellen, welches ´Kabel-Minimum´ überhaupt notwendig ist. Da sind:

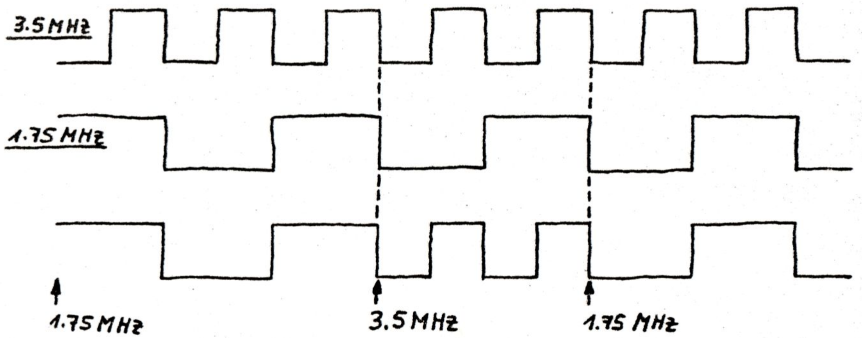
2x Stromversorgung
3x Taktsignal (2 Eing., 1 Ausg.)
<u>3x Umschaltung</u>
8x Anschlußleitungen

Vorrausgesetzt habe ich dabei, daß eine softwaremäßige Umschaltung nicht benötigt wird. Sonst ist ein Dekoder mit zusätzlichen 10 Leitungen nicht zu umgehen. Eben jenes aber galt es zu vermeiden. Damit ist der ´Kupferbedarf´ bestimmt. Bliebe noch der ´Siliziumaufwand´. Die ideale Lösung besteht aus einem IC, nur die habe ich nicht gefunden. Macht auch nichts, mit zwei IC´s gebt´s schließlich auch. Von Nöten sind also ein 74LS00 sowie ein 74LS74. Die NAND´s wandern in einen Multiplexer, die Flip-Flops bilden einen Pulssynchronisierer. Das alles inklusive einiger Zutaten wie Lötzinn, Draht, Widerständen etc. in einen Topf, kurz aufkochen und dann in etwa einer ³/₄ Stunde zusammenbauen. Für die allzeit bereiten Hobbyköche, die ihren Tauchsieder - sprich Lötkolben - stets parat haben, hier nun das Kochrezept (=Schaltbild):



Nun !, wie hat es gemundet ?. Zum Nachtisch gibt es Theorie, das schließt den Magen und rundet die Sache so richtig ab.

Die Schaltung baut darauf auf, daß die beiden CPU-Takte mittels Teilung durch 2 entstanden sind. Damit gibt es einen Zeitpunkt, zu dem beide Signale flankensynchron den Zustand wechseln. Oder anders ausgedrückt, beide Signale wechseln gemeinsam von HIGH nach LOW bzw. von LOW nach HIGH. Der ganze Witz der Schaltung basiert genau auf dieser Sache. FF1 wird nun mit einem Wechseltasters (Taster mit zwei möglichen Tastrichtungen) über S und R angesteuert (Gleichzeitig erfolgt so eine Entprellung des Tasters). Durch die Zwischenspeicherung des ´Umschaltbefehls´ in FF1 ist es nun möglich, den Taktwechsel zum richtigen Zeitpunkt vorzunehmen. Dieser kommt mit dem Auftreten der negativen Flanke des 1.75Mhz Taktes. FF2 übernimmt dann die Information von FF1 und schaltet mit seinen Ausgängen den Mutiplexer um. Nachfolgend hierzu ein Pulsdiagramm mit den beiden Takten und einer Umschaltsequenz SLOW -> HIGH -> SLOW. Der Inverter in der Taktleitung von FF2 ist notwendig, da sonst auf der positiven Flanke umgeschaltet wird.



Wie zu erkennen ist, verlängert/verkürzt sich jeweils eine Hälfte der betroffenen Periode. Das stört die CPU jedoch nicht weiter. Die Schaltung arbeitet so sicher, daß selbst schnelles hin- und herschaltet zwischen den beiden Takten zu keinem ´hängenbleiben´ des Z80 führt (nicht führen kann, da ja <u>immer</u> auf die negative Flanke zum Umschalten gewartet werden muß !).

Damit bei jedem Einschalten des Rechners stets der schnelle Takt angewählt ist, ist über die Diode D1, C1 und R3 ein Power-On-Reset eingebaut.

Somit steht nun eine weitere Variante der Taktumschaltung zur Wahl. Allen die derartiges noch nicht oder erst noch einbauen wollen, eine angenehme Wahl der Qual.

Tschüß

Kurt Müller

BÖRSE -- BÖRSE -- BÖRSE

* * V e r k a u f e * *

Model III (deutsche Tastatur, 2 Laufwerke)
HRG (640 * 240) von TANDY
absoluter Topzustand
Preis: Gegen Gebot
Thomas Buskowiak   Tel.: 069 - 56 01 621

WER HAT   GENIETEXT
für GENIE III mit zugehörigem MIDABAS
evtl. auch leihweise für einmalige Anwendung

UNIDAT
ebenso ??????????????????????

Richard Rensch, Bahnhofstr. 100, 7128 LAUFFEN AM NECKAR

STAR Gemini-10X zu verkaufen

Centronics-Interface (parallel; serielles Interface nachrüstbar), Übertragungsgeschwindigkeit 6000 cps, Druckgeschw. 120 cps
8 intern. Zeichensätze, selbstdefinierte Zeichen, HRG, 6 versch. Zeichenbreiten, Sub- und Superscript, kursiv, versch. Schriftstärken, horiz. und vert. Tabulation usw.
Bei Bedarf mit Druckerkabel (kost nix extra) und Interface (DM 50,-, passend für Expansion-Stecker des Genie I/II, überflüssig bei TCS-FDC-Laufwerken)
Software für Bildschirm-Hardcopy (Klötzchengraphik, inverse Zeichen und HRG für Genie I/II/IIIs; bitte angeben!)
Das Schriftbild und die Graphikmöglichkeiten sind euch aus meinen Artikeln sattsam bekannt.
Preisvorstellung: DM 500,- (VB)

Arnulf Sopp

Software zum Nulltarif und Programme auf Probe

Die Preise für Computer sinken, Programme hingegen sind noch verhältnismäßig teuer. Jeder Computer-Einsteiger macht diese Erfahrung. Kaum bekannt ist, daß Tausende von Originalprogrammen kostenlos zur Verfügung stehen.

Seit es Computer gibt, haben sich Benutzer in Klubs zusammengeschlossen, um ihre Erfahrungen auszutauschen. Im Laufe der Jahre haben unzählige „User", vor allem in den USA, Programme unterschiedlicher Qualität für alle nur denkbaren Anwendungsgebiete geschrieben. Diese Programme sind nicht nur copyrightfrei, die Programmierer fordern sogar zum Kopieren und zur Weitergabe auf. Allerdings dürfen diese Programme nicht kommerziell genutzt werden.

Die EcoSoft (Economy Software AG) in 7890 Waldshut-Tiengen, Kaiserstraße 21, bietet seit Jahresanfang Freiprogramme für die Systeme Commodore 64, Apple-Computer und IBM PC an.

So stehen für den populären C 64/128 etwa 160 Disketten mit rund 1500 Einzelprogrammen und für Apple-Computer sogar 250 Disketten mit insgesamt 6000 Einzelprogrammen zur Verfügung.

Eine fantastische Fundgrube. Da findet man neben Programmierhilfen Musik- und Lernprogramme, Rezepte, Flugsimulatoren, kurz, alles nur Wünschenswerte. Gegen Einsendung von 10 Mark erhält man einen kompletten Programmkatalog auf Diskette.

Ganz kostenlos ist die Software natürlich nicht: Je nach Bestellmenge wird pro Diskette eine Bearbeitungsgebühr berechnet. Und: Die Qualität der Programme ist unterschiedlich. Neben früheren Versionen heute kommerziell angebotener Programme sind da auch einfache BASIC-Spielchen im Angebot. Aber das ist eigentlich Nebensache, setzt man einmal die rund 15 Mark für eine Diskette mit 20 Programmen gegen die mindestens 40 bis 50 Mark, die man sonst für ein einziges Spielprogramm zahlen muß.

Eine weitere Alternative zur teuren kommerziellen Software ist unter der Bezeichnung „Shareware" zu haben, die ebenfalls von EcoSoft vertrieben wird. Die Autoren geben ihre Programme zunächst kostenlos ab und überlassen es dem Anwender, sie auf Brauchbarkeit zu testen. Sagt dem das Programm zu, läßt er sich gegen Übersendung eines entsprechenden Geldbetrages beim Autor registrieren. Dafür erhält er die jeweils neueste Version, oft auch ausführliche Handbücher, und kann den Autor direkt um Hilfe bitten.

Bits übers Radio

Unter diesem Titel bringt der Computer-Laden einmal im Monat Wissenswertes, Aktuelles und Kurioses aus der Welt der großen und kleinen Computer. Die Sendung verrät Tips und Tricks für Anfänger. Profis und Amateure berichten, wie sie den Zugang zum Computer gefunden haben. Und schließlich schicken die Macher der Sendung sogar Programme über den Äther, die sich die Zuhörer aufnehmen und in ihren Computer laden können. Die Sendung wird jeweils an Dienstagen von 8.30...9.00 Uhr und von 14.30...15.00 Uhr ausgestrahlt (Südfunk, Stuttgart).

Folgende Termine sind vorgesehen:

Beginn 8.30 Uhr	Beginn 14.30 Uhr
25. November	18. November
23. Dezember	16. Dezember
3. Februar	27. Januar
3. März	24. Februar
31. März	24. März
5. Mai	28. April
2. Juni	26. Mai
30. Juni	23. Juni

* Neues von der DISKO-Front *

Seit unserem vorletzten INFO-Heft (Nr.14) wurde unsere Programmbibliothek nicht nur durch ein umfangreiches "Public Domain"-Paket bereichert, desen Programme im letzten Heft (Nr.15) komplett aufgeführt wurden, sondern auch durch 20 Einsendungen aus unseren Reihen! Diese Programme mit den Nummern 348 bis 367 werden nachstehend kurz vorgestellt.
Hartmut Obermann sandte acht und *Gerald Schröder* elf Programme ein. Der "Diskothekar" dankt im Namen des Clubs!
Da ein Teil der Programme von *Gerald* sich auf "Banking" bezieht und ich auf keiner Bank sitze (und infolgedessen auch kein "Bankier" bin), habe ich ihn gebeten, ein paar erläuternde Worte mehr dazu zu veröffentlichen, als ich sie in den jeweils drei Kommentarzeilen, die für jedes Programm vorgesehen sind, bringen kann. Ich hoffe, daß er das noch bis zum Redaktionsschluß dieses INFOs hingekriegt hat.
Besonderen Dank verdient *Rainer Jablotschkin!* Er hatte das hervorragende HRG-Programm GRAPE2.1 bereits ehrlich erworben und wollte es vor etlichen Wochen in guter Absicht versehentlich dem Club anbieten. Dieses freundliche Ansinnen wurde vom Diskothekar unter Hinweis auf unsere strenge Satzung (keine Raubkopien!) bedauernd abgewiesen. Nun tat Rainer das einzig Richtige: Statt zu schmollen (=leichtere Form von Grollen), schrieb er den Urheber und Copyright-Inhaber, *Martin Winter, Stuttgart*, an und erhielt für dieses Programm die Freigabe für den CLUB80! * Vielen Dank, Rainer! * Das komplette Programm sowie die Anleitung dazu befinden sich nunmehr in unserer Diskothek! Kopie des Freigabe-Schreibens von Herrn *Winter* vom 5.Sept.1986 liegt dabei.
Es ist allerdings nicht sicher, daß dieses Beispiel Schule macht, denn Herr *Winter* schrieb auch, daß er am Verkauf dieses Programmes ohnehin nicht mehr interessiert sei. Trotzdem band er die Freigabe an die Bedingung:
GRAPE darf nur innerhalb des CLUB80 weitergegeben werden!
Doch das gilt ja, wie ihr wißt, ohnehin grundsätzlich für alle eingesandten Programme - wer zuwiderhandelt, "fliegt"! (Siehe Satzung. * übrigens: Wer sie noch nicht hat oder auch schon wieder "verlegt" hat, erhält sie beim Vorstand!)

Nun noch etwas in eigener Sache.
In den 10 Monaten, seit ich die PB verwalte, ist eine ganz ansehnliche Menge Programm-Anforderungen bei mir eingegangen. Da macht man schon mal lustige Erfahrungen! Z.B.: Irgendwann irgendwo muß ich mal im Zusammenhang mit der Beschreibung der persönlichen Computer-Konfiguration (zu Deutsch: Ausstattung des Computers nebst Peripherie) erwähnt haben, daß zu meinem alten *TANDY* (TRS80/M1) ein "EXPANSIONS-INTERFACE" gehört. Das müssen einige Kameraden total mixverstanden haben! (Kein Tippfehler.) Einige glaubten offenbar, das sei eine *"Dehnmaschine"*, die ich erfunden hätte ("to expand" heißt ja in der Tat "ausdehnen", "erweitern"; aber Expansion ist nicht dasselbe wie Expander!). Diese gutgläubigen Optimisten baten mich um Kopie von einigen Dutzend Programmen und schickten mir dazu ein oder auch zwei Disketten, die auf 40 Spuren mit SS/SD formatiert waren, wo (bekanntlich) etwa 78 freie Granules draufgehen, so daß ich davon ausgehen mußte, daß sie nur einseitige Laufwerke sowie keinen Doubler besitzen! Es ist mir trotz aller Bemühungen bis heute nicht gelungen, 78 Granules auf 478 Granules auszu*dehnen*!! (Mein "Expander" taugt eben nichts!)
Aus diesem Grunde "habe ich mich entschlossen", in der Datenbank der Diskothek das Feld "NOTE", dessen Beantwortung mir ohnehin schon immer gewisse Schwierigkeiten bereitete, in "Anzahl Granules" umzufunktionieren; kurz: Grans! * Ich habe mir die fast unendliche Mühe gemacht, die Anzahl Granules in sämtlichen bereits über 300 Programmen nachzuschauen und in allen Records der Datenbank nachzutragen! Whooff!
Nun seid bitte so nett und rechnet bei euren Bestellungen erst einmal die Anzahl Granules der gewünschten Programme (einschl. etwaiger Paketteile) zusammen. Es handelt sich übrigens dabei um eine reine Additionsaufgabe der 2.Grundschulklasse, für die es sehr preiswerte Taschenrechner gibt. Nicht etwa extra den Computer dafür anschmeißen! Der kann nicht computo (lat.rechnen). Ich wüßte mit einer Bestellung von 18.6666666666666 Programmen wieder nichts anzufangen...
Im übrigen bin ich stets sehr erfreut, wenn man mir - wie es die meisten bereits dankenswerterweise taten - bei Bestellungen nicht nur den Titel, sondern auch die Disknummer aufgibt; die zusätzliche Angabe der Programm-Nummer (wozu haben wir diese denn?) bietet Redundanz (=erhöhte Sicherheit durch "doppelt-gemoppelt") - und der Diskothekar kann auch weiterhin die Aufträge mit gewohnter Promptesse abwickeln!
Bei Bestellung von "Public Domain"-Software bitte ich, irgendwo einen Vermerk >PD< anzubringen. → (PD)
da diese Disketten ihre eigene Numerierung von 1-25 haben, damit keine Verwechslung mit der Disketten-Numerierung der laufenden Programmbibliothek (Source = CLUB80) eintritt.
Oder: Römische statt arabische Ziffern verwenden! Z.B. "XIX" statt "19". (Wer schreibt ein Programm dafür??) Danke sehr.
AaPopo: Am liebsten formatiere ich eure Disks selbt (gemäß Angabe natürlich)! Mit vorformatierten Disketten habe ich bereits Schiffbruch erlitten; besonders dann, wenn sie von den beneidenswerten 80Track-Leuten kamen, die mir aus liebenswerter Rücksicht auf meine bescheidene 40Dräck-Ausstattung 40 Tracks auf ihre Disks zaubern wollten! Geht nicht! (oder selten.) Irgendwo habe ich gelesen, daß man mit 80-Spur-Laufwerken zwar 40Spur-Disketten lesen könne, daß es aber beim Beschreiben derselben zu Komplikationen komme. (Stimmt auch nicht ganz. Ich habe es erlebt, daß große Könner und Besitzer modernster "Konfigurationen" meine bescheidene Ware nicht lesen konnten!
(Statt Genuß an meinem Stuß - hatten sie nur Ärgernus...)
Also gebt mir bitte nur eure Wünsche auf! I.a. ist es keine Hilfe, Vorgekautes zu verdauen. Selbst die geniale "ID"Funktion von *Heidenreich* hat schon oft kläglich versagt!
(Arnulf, was sagst Du dazu? Wahrscheinlich liest Du dies hier gar nicht...)
Formatieren in einem Aufwaschen mit dem Kopieren ist hingegen ein Klacks. Ich habe nur 40 Tracks, so leid mir's tut!

Viel Spaß an und Erfolg mit der Diskothek des CLUB80!
Dies wünscht

K.-P.

/WDR-Files GRF Progr 0348 Teile: 18 / Grans: 167 Disk 12
========== f.Comp. Mod1,z.T.4 Quelle WDR-COMPUTERCLUB
Verfasser versch. eingetippt von H. Obermann
Grafikprogramme: APFELMANN/DRAGON/DRIVEUM/HRGCHESS/KREISE/
SHUTTLE/SINXX/SURFACE/TRON/VERSCHLU/WDR2306/ZYKLOID/DRUCKBLD
LIESBILD/SCANBILD.-Ueberblick in: PROGR/TXT

CALL/CMD UTL Progr 0349 Teile: 02 / Grans: 02 Disk 12
======== f.Comp. alle Quelle
Verfasser H.Obermann eingetippt von H.Obermann
Maschinenroutinen von DOS aus "callen".
Z.B.: CALL 01C9 --> CLS

EXECUTE/CMD UTL Progr 0350 Teile: 02 / Grans: 03 Disk 12
=========== f.Comp. alle Quelle
Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
Das Progr. erlaubt es, kleine von Hand assemblierte Maschi=
nen-Routinen direkt einzugeben und auszufuehren. Z.B.:
210000110070 entspr.LD HL,0000h LD DE 7000h

LIESHRG/BAS GRF Progr 0351 Teile: 01 / Grans: 01 Disk 12
=========== f.Comp. Mod. 1/4 Quelle
Verfasser Hartmut Obermann eingetippt von Hartm.Obermann
Programm zum Einlesen von Model1-HRG-Files in die Model4-HRG

LPRINT/CMD UTL Progr 0352 Teile: 02 / Grans: 02 Disk 12
========== f.Comp. alle Quelle
Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
erlaubt die direkte Ausgabe von Steuersequenzen vom DOS aus
an einen Drucker.- Z.B.: LPRINT 1B40 ==> ESC §
(Initialisierungssequenz f. EPSON-Drucker)

OUT/CMD UTL Progr 0353 Teile: 02 / Grans: 02 Disk 12
======= f.Comp. alle Quelle
Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
erlaubt die Ausgabe von HEX-Werten an Ports direkt im DOS.
Z.B.: OUT 01,00 ==> "HRG einschalten!" (Model 1)

POKE/CMD UTL Progr 0354 Teile: 02 / Grans: 02 Disk 12
======== f.Comp. alle Quelle
Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
wie BASIC-Befehl 'POKE', jedoch unter DOS verwendbar. Z.Bsp.
POKE 4210,60 ==> "Double speed!" (Model 4)

GRAFLIX/BAS GRF Progr 0355 Teile: 07 / Grans: 15 Disk 12
=========== f.Comp. alle Quelle COMPPERS.3.10.84
Verfasser F.DEBATIN eingetippt von Hartm. Obermann
Gutes Grafikprogramm mit eigenem HRG-Treiber "HRGLIX/CMD".
Grafik kann mit Text aus DOTWRITER-Zeichen gemischt werden.
Start: 'HIMEM CFOOH HRGLIX RUN"GRAFLIX/BAS"'

GRAPE21 GRF Progr 0356 Teile: 18 / Grans: 65 Disk 10
======= f.Comp. alle Quelle M.WINTER,STUTTG.
Verfasser Martin Winter eingesandt von R. Jablotschkin.
Das bekannte Programm fuer die hochaufloesende Grafik wurde
von M.Winter fuer den CLUB80 freigegeben. Es darf nur inner=
halb des Clubs weitergegeben werden!

Programm-Eingaenge seit INFO 14 (ohne Publ.Dom.)-Stand: 22.10.86

TRSSYS/SRC UTL Progr 0357 Teile: 01 / Grans: 06 Disk 13
========== f.Comp. TRS 80 Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Laedt alle Systemfiles in zwei Banks. Man kann somit die
Systemdiskette aus Laufwerk 0 entfernen.
(Arbeitet mit Helmut Bernhardts Banker.)

SYSCOPY1/SRC UTL Progr 0358 Teile: 01 / Grans: 06 Disk 13
============ f.Comp. GENIE IIs Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Arbeitet aehnlich wie TRSSYS/SRC (Nr.357), jedoch
n u r a u f G E N I E II s !

TRSRAM/SRC UTL Progr 0359 Teile: 01 / Grans: 11 Disk 13
========== f.Comp. TRS 80 Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
initialisiert RAMDISK als Lw2 oder 3 mit 64-896KB Speicherpl
Parameter: F=RAMDISK formatieren; 1-E: RAMDISK mit 1-14*64K.
N=RAMDISK entfernen, alt.Zust.restaurieren.

RAMDISK/SRC UTL Progr 0360 Teile: 01 / Grans: 10 Disk 13
=========== f.Comp. GENIE IIs Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Wie "TRSRAM/SRC" (Progr.-Nr.359), jedoch
n u r f u e r G E N I E II s !

COMPARE/SRC UTL Progr 0361 Teile: 01 / Grans: 09 Disk 13
=========== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Vergleicht zwei CMD-Files und beruecksichtigt dabei unter=
schiedliche Laengen oder Spruenge bei den Ladeadressen.

SYS8/SRC UTL Progr 0362 Teile: 01 / Grans: 12 Disk 13
======== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Ein neues SYS8/SYS fuer G-DOS. Par.$ fehlt.Neu: xxxxxxxx/yyy
x/y=Buchst./Ziffer, "?"=Zeichen beliebig, "*"=alle folgenden
Zchn.belieb.-Nur entspr.Files werden angez.

TRSMAC/CMD UTL Progr 0363 Teile: 02 / Grans: 16 Disk 13
========== f.Comp. Quelle MIT SOURCE-CODE
Verfasser SCHROEDER, Gerald eingetippt von dto.
Hilfsprogr.f.Eingaben. Wird ueb.Parametersatz aufgerufen. Es
koennen dann Texteingaben gemacht oder Flags umgeschaltet
werden. Ein spezieller "Full-Screen-Editor"

TRSTST/CMD UTL Progr 0364 Teile: 02 / Grans: 18 Disk 13
========== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Fuehrt das Programm "TRSMAC/CMD" (Nr.363) vor. Mit Source-C.

TRSDEMO/BAS UTL Progr 0365 Teile: 01 / Grans: 02 Disk 13
=========== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Fuehrt ebenfalls das Programm "TRSMAC/CMD" (Nr.363) vor.

Programm-Eingaenge seit INFO 14 (ohne Publ.Dom.)-Stand: 22.10.86

TRSDEF/BAS UTL Progr 0366 Teile: 01 / Grans: 15 Disk 13
========== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Zum Erstellen von Eingabemasken f. BASIC-Programme.
Siehe hierzu auch das Hilfsprogramm "BILDUMW/BAS" (Nr.367).

BILDUMW/BAS UTL Progr 0367 Teile: 01 / Grans: 03 Disk 13
========== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Hilfsprogramm zu "TRSDEF/BAS" (Nr.366). Wandelt die Eingabe=
maske in BASIC-Anweisungen um (auch fuer andere Programme
verwendbar).

Wer möchte Diskothekar werden?

Wenn dieses Jahr zu Ende geht, werde ich die Programmbibliothek unseres CLUB80 genau ein Jahr lang verwaltet haben!
Es war eine interessante Arbeit. Ich hatte nicht nur das beruhigende Gefühl, außer dem pflichtgemäßen Clubbeitrag in DM auch einen physikalischen und physiologischen Beitrag (Tasten Tippen und Gehirnzellen Strippen) zu leisten, sondern ich habe auch viel dabei zugelernt. Von dem Programm-Reichtum, der mir auf diese Weise unverlangt (und unverdient) zugeflossen ist, ganz abgesehen.

Nun hat sich für mich aber eine neue Situation ergeben. Aus sehr persönlichen Gründen muß ich dieses schöne Hobby jetzt leider aufgeben.

Clubkollegen - vor allem diejenigen, von denen man sonst noch nie gehört hat - haben hier eine Chance, sich zu Hause in aller Gemütlichkeit um den Club verdient zu machen, wenn sie ansonsten keine Lust, Übung, Mut oder sonstwasweißich haben, selbst Programme zu schreiben und in unserem INFO zu veröffentlichen!

Wer also Lust hat, der melde sich bald bei mir! Ihm steht nicht nur ein Potential von nahezu 600 Programmen zur Verfügung (auf nur 40 Disketten) nebst Anleitungen, sondern es eröffnet sich auch eine Fülle interessanter Kontakte, auf die man sonst gar nicht gekommen wäre.

Die Einarbeitung ist einfach. Ich habe eine für kluge Köpfe (ihre Stückzahl geht aus der Mitgliederliste hervor) völlig hinreichende Anleitung - nur 1 Seite! - für die Datenbank SUPER ausgearbeitet (die ausführliche steht natürlich auch zur Verfügung). Viele von euch kennen sie. Sie ist zwar etwas langsam (BASIC), hat aber den Vorteil, sehr sicher im Umgang und flexibel genug zu sein, um sich vielen Bedürfnissen anzupassen; z.B. Ausdrucken mit beliebiger Formatierung und Felder-Auswahl, selektive Suchfunktion mit mehreren Prioritätsstufen, wobei binäres Suchen sehr schnell geht, usw. Gerade weil sie in BASIC vorliegt, läßt sie sich leicht nach eigenem Wunsch und Geschmack verändern. So habe ich die Form, wie die "Diskothek" (Programmbibliothek; siehe in diesem Heft) ausgedruckt wird, durch einfache Eingriffe in das Programm selbst gestaltet und natürlich an meinen Drucker (EPSON) angepaßt - was jederzeit leicht geändert werden kann.

Fantastische Hilfe beim Kopieren leistet die Utility (wirklich eine "Nützlichkeit"!) DIS/CMD! Wird ein Programm gewünscht, lege ich nur die Diskette mit der vom Besteller genannten Nummer ein, sage "S Filename" (=Suche Programmnamen); schon springt der Cursor dorthin - falls das Programm drauf ist; andernfalls hat der Besteller eine falsche Disk-Nummer genannt. Genau dann hilft die Datei SUPER weiter. Sonst, wenn die Disknummer stimmt, wird SUPER überhaupt nicht gebraucht. Ich habe es in den 10 Monaten bisher ein einziges Mal gebraucht! (Am 22.10.86. Der Gute, der die WDR-Files bestellte, hatte mir überhaupt keine Disknummer genannt!) SUPER ist dann mein NOKIXEL <bitte rückwärts lesen!> Aber auch nur dann!
Der Cursor springt also sofort auf das gewünschte File. Dann tippe ich nur noch "C n" - fertig; und DIS/CMD kopiert mir das Programm auf Laufwerk n mit der Diskette des "Kunden"! Keine Eingabe von "COPY" und Filenamen usw.usf.

Für diejenigen, die die Mitgliederliste mit den Anschriften sorgfältig und gewissenhaft abgetrennt haben, damit sie unser INFO auch einmal einem TRS80iger, GENIEler oder KOMTEKer, der noch "außen vor" steht, zeigen und für uns werben können, wiederhole ich meine Anschrift hier nochmal:

K.-J.Mühlenbein, Am Mönchgarten 28, 6940 Weinheim

Ich rechne mit zahlreichen Bewerbern! Dann entscheiden: Ver- und Vorstand. Bis bald also! Euer

K.-J.

- IMPRESSUM -

1. Vorsitzende — **Peter STEVENS**
Postfach 56
4600 Dortmund 1
☎ 0231 / 593883

2. Vorsitzende — **Hartmut OBERMANN**
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 / 3913

HardwareKoordinator — **Eckehard KUHN**
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 / 45417

DisKothekar — **Klaus-Jürgen MÜHLENBEIN**
Am Monchgarten 28
6940 Weinheim -Lü.
☎ 06201 / 55052

Redaktion — **Jens NEUEDER**
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 / 42877

Autoren dieser Ausgabe:

Helmut Bernhardt	Dieter Kasper
Eckehard Kuhn	Klaus-J. Mühlenbein
Kurt Müller	Jens Neueder
Hartmut Obermann	Gerald Schröder
Arnulf Sopp	

sowie Artikel aus: 80 Micro

Bankverbindung des CLUB 80

Postgirokonto	Peter STEVENS
	SonderKonto CLUB 80
Konto-Nummer	285 491 - 465
Postgiroamt	Dortmund
BLZ	440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

SCHLUSS

Hallo Club-80er,

leider kann ich Euch jetzt erst das neueste INFO bieten. Dies hatte private sowohl auch berufliche Gründe. Weiterhin hat es im Club eine Umorganisation gegeben, die Euch natürlich auch bekanntgegeben werden musste.
Darum leider erst jetzt das 16. Clubinfo.

Da dies auch das letzte INFO sein wird, daß Ihr in diesem Jahr bekommt (das Nächste erscheint Anfang/Mitte Januar 87), möchte ich mich bei allen Mitwirkenden, an unserer Clubinfo, recht herzlich für die geleistete Arbeit bedanken.
Gleichzeitig geht aber auch meine Bitte an alle, auch im nächsten Jahr wieder so produktiv zu sein. Auch Autoren-Neulinge sind im Kreis der INFO-Schreiber herzlichst willkommen.
Durch die INFO wird doch eine Menge an Informationen an die Clubmitglieder gebracht. Sicherlich war für jeden von Euch etwas Brauchbares dabei. Ich denke auch, daß wir mit dem Konzept unserer Clubinfo recht gut liegen (Zumindest gab es keine größeren Klagen). Bedenkt man dabei noch, daß wir mehr und mehr eine RARITÄT -mit unserem 8-Bit'er- besitzen und auf uns alleine gestellt sind, ist die Aktivität in unserem Club doch recht gut. Ich wünsch Euch und mir, daß es auch weiterhin so bleibt.

Am INFO-Ende sind diesmal zwei SONDERHEFTE beigefügt. Ich habe mich für diesen Weg entschieden, da die behandelten Themen recht umfangreich sind. Gleichzeitig möchte ich damit auch die Arbeit der Autoren würdigen, die sich wieder einmal sehr viel Mühe gemacht haben.

Als Vorstandsmitglied möchte ich Euch ZUM SCHLUSS MEINES SCHLUSSES noch an die anstehenden Termine erinnern. Bitte merkt Euch die wichtigsten Termine vor.

Einstweilen wünscht Euch viel Spaß an der neuesten INFO, ein FROHES FEST und einen GUTEN RUTSCH IN'S NEUE und Allzeit HAPPY HACKING.

Euer

Jens Neueder

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117 Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060 Bergisch-Gladbach 1	02204 /62781	//
Bernhardt	Helmut	Hafenstraße 7	2305 Heikendorf	0431 /241907	// 0431 /74047
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000 Frankfurt 1	069 /5601621	//
Böcker	Dieter	Lehmweg 4	2930 Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414 Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200 Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441 Wettringen	05233 /4320	// 02557 /1236
Emmrich	Helmut	Waldstraße 5	6682 Ottweiler	06824 /4114	// -
Frink	Thomas	An der Schleifmühle 2	5042 Erftstadt	02235 /76255	//
Grajewski	Werner	Zedernweg 29	4220 Dinslaken	02134 /54573	//
Hartel	Eberhard	Neenstetter Straße 20	7901 Breitingen	07340 /7281	//
Heile	Heinz-Dieter	Blankensteiner Straße 13	4320 Hattingen	02324 /28458	//
Held	Manfred	Stirnerstraße 22	8835 Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401 Pliezhausen	07127 /70024	//
Hill	Peter	Eckstraße 36	6750 Kaiserslautern 31	0631 /54782	//
Hummel	Anton	Schubertstr. 2	7612 Haslach	07832 /8289	//
Jablotschkin	Rainer	Thiekamp 29	4780 Lippstadt 8	02948 /542	// 02921 /70431
Kasper	Dieter	Zeppelinstr. 9	8952 Marktoberdorf	08342 /1630	// 089 /522071
Konrad	Josef	Anzengruberstraße 35	8038 Gröbenzell	08142 /8494	// -
Kopschina	Peter	Strandallee 138	2409 Scharbeutz	-	// -
Krüger	Karl-Herbert	Bruchweg 65	4920 Lemgo	-	// -
Kuhn	Eckehard	Im Dorf 14	7443 Frickenhausen 1	07022 /45417	// 07022 /77442
May	Holger	Marienstr. 9	5768 Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505 Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /668151
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940 Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050 Hamburg 80	040 /7246083	// 04103 /702662
Neueder	Jens	Panoramastraße 21	7178 Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209 Heidenrod 1	06124 /3913	// -
Perschbach	Patrick	Waldstr. 52	5000 Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835 Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146 Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080 Fürstenfeldbruck	08141 /3791	//
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128 Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082 Uetersen	04122 /43551	// 04103 /7025310
Rychlik	Andreas	Königsberger Allee 120	4100 Duisburg 1	0203 /331383	// 0203 /331383
Schmitz	Paul-Jürgen	Lübesckerstraße 6	6236 Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000 Köln 71	0221 /707044	//
Schrewe	Christian	Fliederweg 32	4000 Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105 Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160 Miesbach	08025 /1631	// 08025 /41247
Smerling	Frank	Tangstedter Str. 5	2080 Pinneberg	04101 /207204	//
Sopp	Arnulf	Wakenitzstr. 8	2400 Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859 Rennertshofen 1	08434 /454	// 08431 /7041684
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506 Hagen a.TW.	05401 /99585	// 05401 /90037
Stevens	Peter	Postfach 56	4600 Dortmund 1	0231 /593883	// 0231 /593883
Trapp	Harald	Kranichstr. 46	4270 Dorsten 1	02362 /42497	// 02362 /23127
Troesch	Eberhard	Altenessener Str. 414	4300 Essen 12	0201 /342324	//
Volz	Oliver	Dusestraße 13	7000 Stuttgart 80	0711 /731285	//
Wagner	Günther	Gartenstraße 4	8201 Neubeuern	08035 /3361	// -
Weiß	Dieter	Bürglestraße 3	7209 Wehingen	07426 /7194	//
Wucherer	Jürgen	Menzelstraße 1	7750 Konstanz	07531 /54686	// -
Zwickel	Walter	Lengfelden 123	A- 5101 Bergheim	0043662/51130	// -

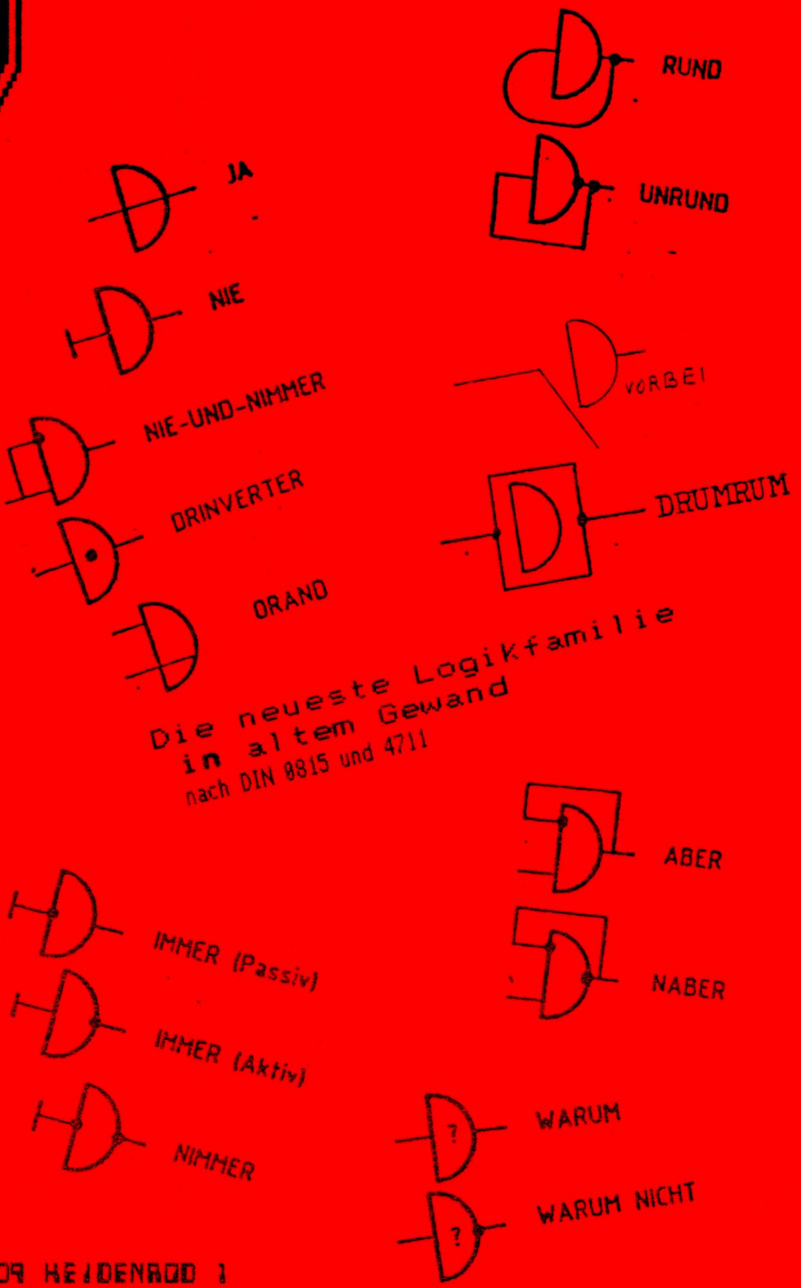
Stand: November 1986

CLUB 80

Clubinfo

der

TANDY - GENIE - und KOMTEK - ANWENDER



17. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
TEL.: 06124 / 3913

Inhaltsverzeichnis

Seite:

Clubinternes

Neues vom Vorstand 01
Clubtreffen '87 02
Clubtreffpunkt . . . . .03 - 04
Eine Antwort 05
Gesucht: ..Grafik-Standart 06
Termine / Hessen 07

Software

Wie Phoenix aus der Asche ... . . .09 - 15
BOOLE XOR ABEL . . . . 16 - 18
Assembler kontra LISP . . . . 19 - 25
Z80, HD64180 und Illegals . . . . . . 27 - 32
TAV's Z80 Tuning . . . . 33 - 35
TAV's Z80 Tuning II . . . . 36 - 37

Hardware

Hardwaretip ... Neue IC's 38
Hardware - Liste CLUB 80 . . . . 39 - 44
Selbstbau Doubler für EXP1 . . . . . 45 - 48
Tandy (-) Schneider . . . . 49 - 53
Nachtrag: Dein GIIIs, ... 54

Sonstiges

directory 26

Programmbibliothek

liegt als Sonder-Heft bei

Die letzten Seiten

Impressum 55
Schluß 56
Clubmitgliederadressen . . . . am INFO-Ende
Termin-Kalender 87 . . . . . . am INFO-Ende
Clubtreffenanmeldung 87. . . . am INFO-Ende

Neues vom Vorstand

Diesmal habe ich zwei Nachrichten für euch, eine gute und eine schlechte!

Zunächst die gute Nachricht:
Dank der intensiven Bemühungen von Jürgen Wucherer (bei dem ich mich herzlich bedanken möchte) haben wir wieder ein preisgünstiges und schönes Hotel für unser Clubtreffen gefunden. Näheres dazu siehe bei "Clubtreffen"!

Die weniger gute Nachricht lautet:
nachdem ich erst vor kurzem die Geschäftsführung des Clubs von Peter Stevens übernehmen musste, werde ich sie wohl genauso schnell wieder abgeben müssen. Der Grund dafür liegt bei meinem Dienstherren, dem BMVg (für Leute denen das nichts sagt: Manfred Wörner). Man wird mich (ich hoffe es kommt nichts dazwischen) auf eigenen Wunsch ab 05.01.1986 auf einen neuen Dienstposten versetzen. Dieser Dienstposten zwingt mich leider dazu, für ein paar Monate eine sog. Wochenendehe zu führen. Ich werde also die Woche über unterwegs und nur an Wochenenden, und längst nicht an allen, daheim sein. Ich hoffe trotzdem, mit Hilfe meiner besseren Hälfte, die Clubarbeit mindestens bis zum Treffen im März korrekt und ohne übergroße Verzögerungen erledigen zu können.
Damit sind wir wieder genau da angelangt, wo wir auch im letzten Jahr standen:

Der Club 80 sucht einen Vorsitzenden!

Hier einmal eine Aufzählung der Arbeiten, die ein Vorsitzender im Laufe eines Jahres zu erledigen hat:
- Betreuung der Clubmitglieder und vor allem der Neuzugänge
- Anregen und Koordinierung von Aktivitäten (z.B. Clubtreffen)
- Verwaltung der Clubkasse
- Vorworte zu den Infos
- Verbindung zu anderen Clubs knüpfen/halten

Wieviel Zeit für diese Arbeiten benötigt wird, läßt sich nur schwer sagen, denn je nach Engagement schwankt der Arbeitsaufwand stark.

Sollte sich irgend jemand mehr für die Arbeit des Vorsitzenden interessieren und mehr Informationen benötigen, bin ich gerne bereit, Auskunft zu erteilen. Ich befürchte aber, daß sich wohl auch diesmal (wie schon in den letzten Jahren) niemand für diesen so wichtigen Posten bewerben wird und der Club wieder einmal nur mit Mühen an einer Auflösung vorbeikommt. Die gesamte Suche nach Vorständen bei den beiden Clubtreffen erinnert mich immer mehr an einen Witz über meinen Berufsstand.

Frage: Was tut ein Soldat, wenn es heißt "Freiwillige einen Schritt vortreten!"?
Antwort: Er tritt einen Schritt zurück, um die Freiwilligen vorbeizulassen!!!

Damit genug für heute über dieses traurige Thema und hin zu erfreulicheren Dingen.

Clubtreffen '87

Wie schon im letzten Info erwähnt, findet das Clubtreffen 1987 in der Zeit von Freitag dem 13.03.87 (hoffentlich ist niemand abergläubisch) und Sonntag dem 15.03.87 statt. Wie weiter oben schon erwähnt, haben wir auch diesmal, dank der schnellen und aktiven Hilfe unseres Freundes Jürgen Wucherer, ein schönes, gut gelegenes und preisgünstiges Hotel für das Treffen gefunden.

Der Ort des Treffens ist diesmal Alsfeld, ein direkt an der Autobahn A5/E4 (zwischen Giessen und Bad Hersfeld) gelegenes, romantisches Städtchen mit historischem Stadtkern. Alsfeld ist sowohl für die "Nordlichter" als auch für die "Südländer" gleich gut zu erreichen und erspart durch seine Autobahnnähe mühsehlige Fahrten auf unbekannten Landstraßen (ist also besser gelegen als die letzten beiden Tagungsorte).

Das Hotel Klingelhöffer, welches uns diesmal Unterkunft und Tagungsräume zur Verfügung stellt, liegt mitten in der Stadt und fällt vor allem durch die Fachwerkfassade und die rustikal-gemütliche Innenausstattung auf. Der Preis für übernachtung mit Frühstück beträgt pro Person in einem Einzelzimmer 35, in einem Doppelzimmer 30 Mark. Der Tagungsraum wird uns kostenlos zur Verfügung gestellt. Natürlich kann man bei Klingelhöffers auch preisgünstig und sehr gut Essen. Hier die genaue Adresse:

Hotel Klingelhöffer
Hersfelder Straße 47/48
6320 Alsfeld (Hessen)
Tel.: 06631 / 2073.

Eine Tagesordnung für das Treffen steht noch nicht fest, ich verspreche euch aber, daß es weniger Administratives als bei den letzten Treffen geben wird. Der Freitag wird sowieso ohne festes Programm ablaufen, der offizielle Teil beginnt erst Samstags nachmittags und der Sonntag gehört wieder ganz dem Erfahrungstausch der Mitglieder. Die genaue Themenaufstellung für den offiziellen Teil des Treffens geht den Teilnehmern noch rechtzeitig zu.

Mit dem Hotel wurde schriftlich eine vorläufige Zimmerreservierung für 25 Einzel- und 3 Doppelzimmer vereinbart. Es ist dringend nötig, bis spätestens Ende Februar eine genaue Reservierung vorzunehmen. Aus diesem Grunde muß das dem Info beiliegende Anmeldeformular bis spätestens 20. Februar bei mir eingegangen sein. Wer sich bis zu diesem Termin nicht darüber im klaren ist, ob er kommen kann oder nicht, muß für seine Zimmerreservierung selbst sorgen.

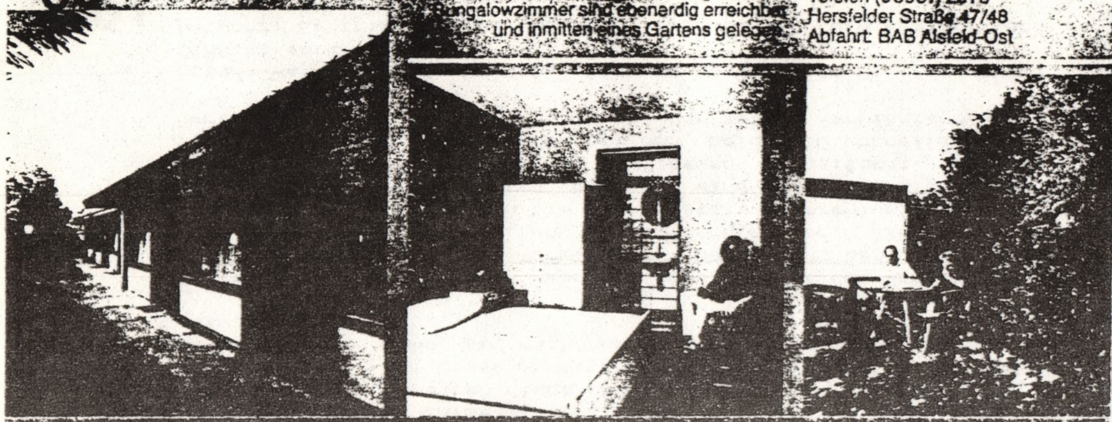
Ich hoffe, daß auch diesmal wieder viele Mitglieder die Möglichkeit nutzen, sich mit Gleichgesinnten zu treffen, Erfahrungen und Informationen auszutauschen und gute alte Bekannte und Freunde wiederzusehen. Vielleicht kann man schon einige Ergebnisse unserer ECB-Bastler sehen und sicher bekommt man neueste Informationen über die ständig wachsende Club-Programmbibliothek. Eventuell kann ich auch den einen oder anderen Spezialisten dazu überreden, eine kleine, verständliche Einführung in sein Gebiet zu geben. Alles in allem weiß ich jetzt schon, daß sich der Besuch des Treffens unter Garantie lohnt und freue mich auf jeden der kommt!

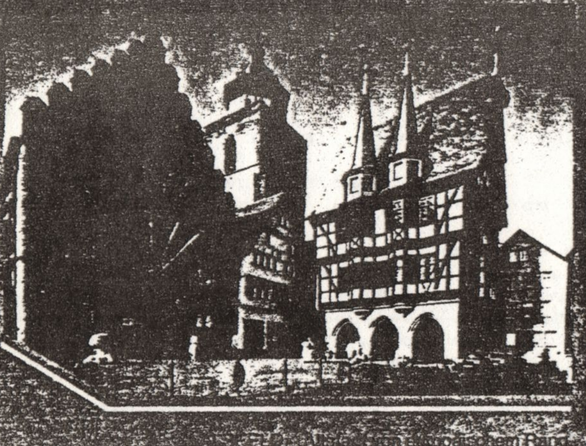
Damit Schluß für heute. Viele Systemcrashs wünscht euch Euer

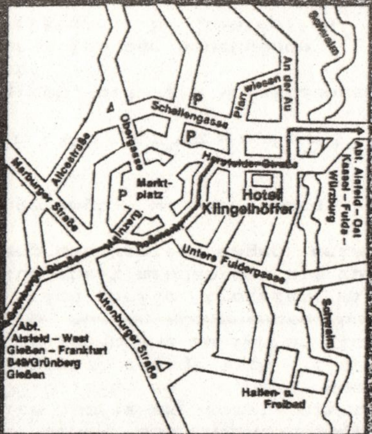
Hartmut Obermann

Die hier abgebildeten Bungalowzimmer sind ebenerdig erreichbar und inmitten eines Gartens gelegen.

6320 Alsfeld (Hessen)
Telefon (06631) 2073
Hersfelder Straße 47/48
Abfahrt: BAB Alsfeld-Ost



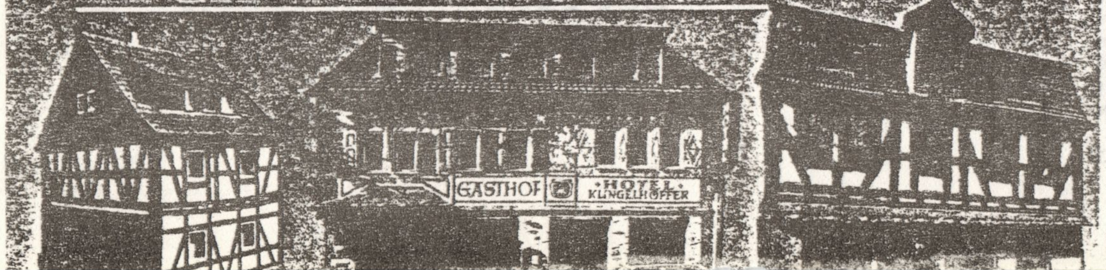




... Altstadt von europäischem Rang ... Deutschen ... straße gelegen.

Anfahrt: Aus Richtung Frankfurt – Gießen
Abfahrt Alsfeld-West
Aus Richtung Kassel – Fulda – Würzburg
Abfahrt Alsfeld-Ost

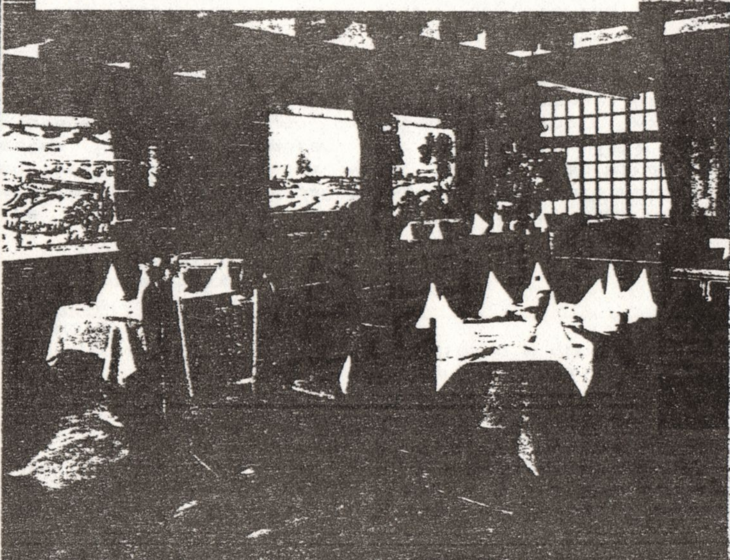
WIR STELLEN VOR...
Hotel Klingelhöffer
MIT SEINEN GÄSTEHÄUSERN



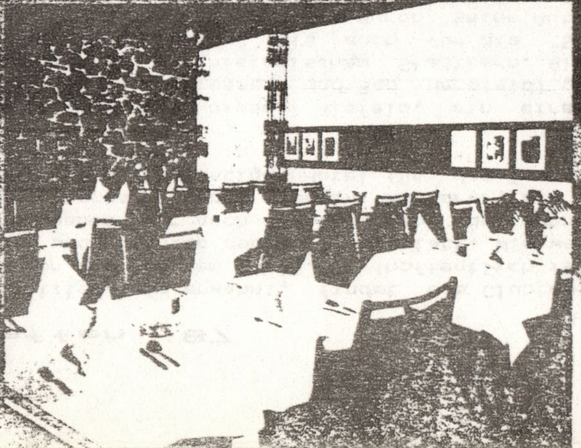
Herzlich Willkommen

Leider ist der Prospekt
über unseren nächsten Tagungsort
nicht besser kopierbar.
Ich wollte es Euch aber dennoch nicht vorenthalten.

Die Redaktion



Das Hotel-Gasthof „Klingelhöffer" in Alsfeld, [illegible] traditionsreicher Familienbetrieb besteht seit 1765. Wo einst Milchkutsche und Postbusse hielten, wo die Holzfuhrleute ihre Vesper einnahmen, erwartet Sie heute ein Hotel mit zeitgemäßem Komfort und kultivierter Gastlichkeit. In verkehrsgünstiger, jedoch ruhiger Lage auf der Nord-Süd-Reiseroute bieten wir [illegible] komfortable Gästezimmer mit 70 Betten. Alle Zimmer sind mit Dusche/WC oder Bad/WC ausgestattet. Gästebungalows, Appartements und Ferienwohnungen garantieren Ihnen einen erholsamen Aufenthalt. In der gemütlichen [illegible] Gaststube [illegible] Sie unter landesüblichen [illegible] Spezialitäten aus [illegible] hessischen Küche [illegible]



* * Eine Antwort * *
an Dieter Kasper

(Zu Peter Spieß' Programmier-Wettbewerb)

Jeder Beitrag ist willkommen; ein kritischer ganz besonders. Also auch *Dieter's* in Heft 16, Seite 5. Ich will auch keine Gegenkritik zu seiner Stellungnahme zu *Peters* Wettspiel-Angebot üben. Das wäre *Peters* Sache.

> Apropos: Auch ich könnte Peter z.B. vorwerfen, daß seine Aufgabe meine Fachkenntnisse überstieg, so daß ich aus dem Rennen von vornherein ausschied! (Offenbar liefen aber alle unter "ferner liefen")... Doch das verkneife ich mir.

Ich möchte *DIETER* nur auf einen - wie ich glaube - Irrtum hinweisen, wenn er sagt, "daß die übersicht enorm darunter leidet", wenn man beim Programmieren möglichst rationell (sparsam) auf die Zeilenzahl achtet, diese also nach Möglichkeit einschränkt. Nach einer IF-Bedingung geht das in der Tat meistens nicht, wenn man nicht mit lauter "ELSEs" um sich schlagen will (eine Else am Halse genügt im allgemeinen...)
Aber Zeilennummern kosten Speicher. Das ist zwar bei BASIC i.a. irrelevant, aber es stellt ein ökonomisches Prinzip dar.
Ich strebe möglichste Verdichtung (nicht nur Dichtung) an. Eine Zeile kann 240 Bytes aufnehmen und mehrere, durch Doppelpunkte getrennte Anweisungen enthalten. Nun, das weiß jeder Programmierer. Auch, daß das sehr "unübersichtlich" ist.

ABER:
1) Man kann erst mal gemütlich ("übersichtlich") programmieren und dann, wenn alles läuft, die einzelnen Zeilen mit einem Bildschirm-Editor (z.B. SEDIT o.a.) maximal zusammenfügen; und
2) kann man bei der Veröffentlichung (oder späteren eigenen Betrachtung) das Listing mit NAME ausdrucken.
"NAME" trennt die statements wieder und ersetzt die Doppelpunkte durch Line Feeds - s. mein kleines Listing in Heft 16, Seite 23, zur FIBONACCI-Folge! "NAME" spaltet nicht nur die langen Zeilen wieder auf, sondern macht auch FOR...NEXT--Schleifen sowie die häßliche IF...THEN...ELSE-Formel (die man sich mit logischen Operationen ersparen kann, siehe Zeile 60 ab "ON") sehr übersichtlich! - Findest Du nicht auch, Dieter? (Zugegeben: Mein Beispiel ist zu schlicht - aber nicht ganz schlecht...)

Last not least: Auch ich würde mich sehr "freuen, wenn das nächste Problem in dieser (von Dir gewünschten) Form gestellt würde" - wie Du am Schluß schreibst. Nur: ich frage mich, an wen sich diese Aufforderung richtet? Nicht auch an Dich selbst? Ich hätte mich (und sicher viele Kollegen auch) also noch mehr gefreut, wenn Du bei dieser Gelegenheit mit Deinem eigenen Vorschlag einer Problemstellung auch gleich den Anfang gemacht hättest! * Denn auch von uns ist niemand dazu "verpflichtet".

* * * Grüß' Dich, *Dieter!* * * * *KaJot M-bein*

PS: Auf *Gerald Schröders* "Flugreichweiten-Problem" hat sich m.W. seinerzeit auch niemand gerührt! Hierüber siehe a.a.O.

Gesucht: Grafik-Standard

Der gute alte TRS80 hatte eine niedliche Klötzchengrafik, mit der sich kaum ein Anwender nach Ablauf der Garantiezeit zufrieden gab. Nur leider gibt es ein Problem dabei: die eingebauten Grafikkarten, sind nicht genormt, werden also unterschiedlich angesteuert und besitzen keine einheitliche Auflösung. Der heimliche Standard dürfte die 384x192-Grafik sein, aber genauso gibt es 640x400 (Mod. IV), 512x512 (IIIs) und 480x192 (IIs). Dazu kommen noch Eigenkonstruktionen oder Anbauten über den ECB-Bus.
Nun gibt es sicherlich für jede Karte mehr oder weniger gute Grafik-Programme, aber nur die Bilder lassen sich (mit einiger Umformung) untereinander austauschen. Warum soll das bei den Programmen nicht auch so sein? Bestes Beispiel ist der IBM-PC, bei dem es keine wirkliche Standard-Karte gibt (Color, Hercules, EGA). Trotzdem bedient jedes bessere Grafik-Programm alle Karten!
Es ist wohl unmöglich, die Hersteller bzw. Software-Firmen im TRS80-Bereich auf eine Norm einzuschwören, aber club-intern sollten wir dies meiner Meinung nach versuchen.
Deshalb folgender Vorschlag: ein paar Grafik-/Assembler-Freaks tun sich zusammen und knobeln einen CP/M-ähnlichen Standard aus. Das soll heißen, daß wir uns auf einen Einsprungpunkt einigen, von dem jedes Programm Gebrauch machen kann. In den Registern sind beim Einsprung genormte Werte zu finden, nach denen ein individueller Grafik-Treiber die Grafik bedient. Als Anfang wären für jede Grafik-Karte folgende Funktionen zu definieren:
1) Abfrage max. X-Koord., max. Y-Koord.
2) Punkt X/Y setzen (SET)
3) Punkt X/Y löschen (RESET)
4) Punkt X/Y abfragen (POINT)

Aus diesen Funktionen lassen sich m.E. alle anderen Grafik-Funktionen zusammenbauen. Alle höheren Funktionen wie Linien und Kreise sind dann allgemeingültig. Natürlich wären auch Basic-Treiber möglich, aber diese brauchten nur auf einem Computer gebaut zu werden und würden dann mit jeder anderen Karte laufen.
Der Treiber, der die obengenannten Funktionen realisiert, müßte natürlich auch seinen Platz haben. Wahrscheinlich ist ein Mindest-Raum anzusetzen, in dem sich für jede Karte diese Funktionen realisieren lassen.

Dies ist als ein Vorschlag zu verstehen, der sich an alle Leser richtet. Es bringt nichts, wenn ich mich allein hinsetze und einen Standard für Newdos und evtl. CP/M ausarbeite, den dann niemand annimmt. Ich möchte deshalb die Interessierten bitten, mir zumindest ihre Meinung zu schreiben. Außerdem habe ich allein natürlich keine Chance, etwas Vernünftiges auf die Beine zu stellen und brauche deshalb Unterstützung von Programmierer-Seite. Da ich sowieso nur eine Grafik-Karte kenne und die nur einmal im Club vertreten ist, wäre meine Arbeit sinnlos. Ich hoffe, daß einige Freaks sich mit mir zusammenraufen und diesen Standard etablieren, damit unsere Rechner auch eine Grafik-Zukunft haben. Ich habe auch keinerlei Ambitionen, bei diesem Projekt die Führung zu übernehmen, sondern wäre froh, wenn sich berufenere Hände als meine dazu finden würden.

Ich warte auf Eure Reaktion!

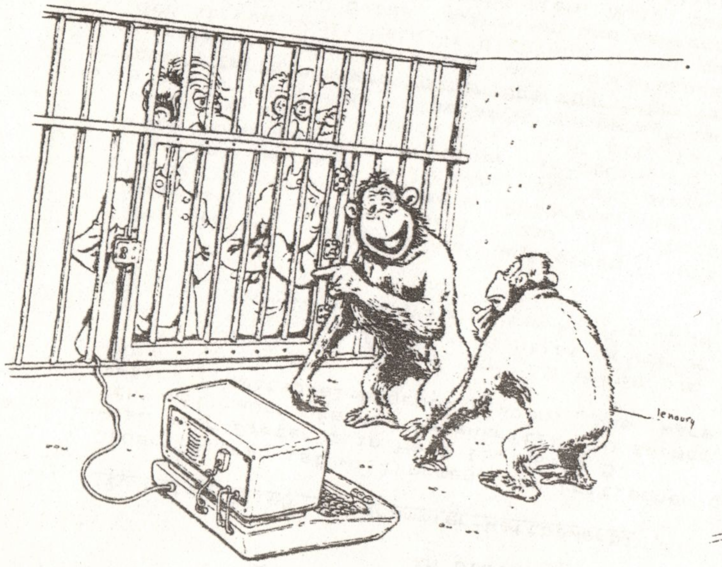
Gerald Schröder

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....28. Februar 1987
Jahreshauptversammlung 1987 ....13. - 15. März 1987
Norddeutsches Regionaltreffen ....25. + 26. April 1987

-- Messen '86 --

Hobby-tronik	Dortmund	18. - 22. Februar 1987
CeBIT	Hannover	4. - 11. März 1987
Internationale Computer-Ausstellung ..	Köln	11. - 14. Juni 1987
HOBBY ELEKTRONIK 87	Stuttgart	5. - 8. November 1987
PRODUCTRONICA	München	10. - 14. November 1987

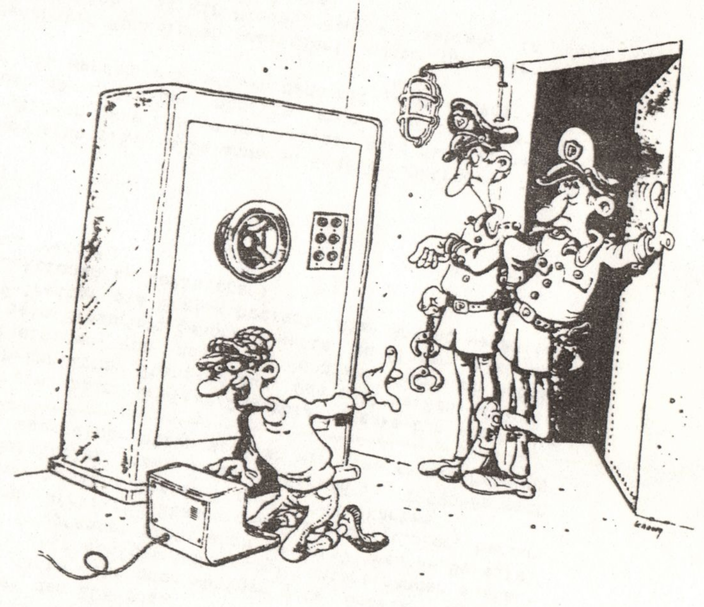


Laß die nicht sehen, was du damit machen kannst, sonst sitzt du nachher auch acht Stunden im Büro.

WEISST DU, WIE SPÄT ES IST? WO WARST DU? HAST DU DIE BATTERIEN FÜR MICH MITGEBRACHT? ICH BIN DIE WARTEREI JETZT WIRKLICH LEID! ALSO NUN HÖR MIR MAL ZU.....

Amerikanisch: Having a ball.
Einen Ball haben.
Sich köstlich amüsieren.

Go fly a kite.
Geh, laß einen Drachen fliegen.
Mach, daß du wegkommst.
Erwin Knop, Big Bear City, California



Mach weiter, Kalle... wir ham noch genau sieben Minuten bis die Bullen hier sind.

Persisch: Dare baq-e ssabs be kasi nišān dadan.
Jemandem die Tür zum grünen Garten zeigen.
Jemanden anlocken, um ihn zu übervorteilen.

Wie Phoenix aus der Asche <u>oder</u>
Vier Computer in einem Schrotthaufen

Die Beilage von Arnulf und Helmut zum letzten Info hat Euch sicherlich geschockt, denn schließlich verfügt nicht jedermann über einen IIIs, den er aufmotzen kann. Mir fehlt zwar das Kleingeld für einen IIIs, aber deshalb gönne ich den IIIs-Besitzern diesen Triumpf noch lange nicht. Deshalb habe ich kurzerhand Arnulfs Idee abgekupfert und auf den schon in der Überschrift erwähnten TRS80 angewandt. Wer sich jetzt extra einen IIIs gekauft hat, um mit Arnulf gleich ziehen zu können, ist selbst schuld.

Wie sieht das in der Praxis aus? Ganz einfach, Ihr belagert einfach den schon mehrmals erwähnten Helmut, bis er Euch einen Banker einbaut. Dann tippt Ihr das folgende Programm ab oder laßt es Euch vom nächsten Diskothekar schicken. Wenn Ihr dann <Shift><Down-Arrow><Ziffer> betätigt, wobei <Ziffer>:= 0 bis 3, könnt Ihr zwischen vier Computern hin- und herschalten. In jedem Computer kann ein Programm laufen, solange dieses Programm
1. die Interrupts angeschaltet läßt und
2. keine fiesen Nebeneinbauten benutzt.

Punkt 2 bezieht sich auf Grafikkarten o.ä. Damit wir uns richtig verstehen: benutzen darf das Programm sie schon, solange beim Sprung zu einem anderen Computer die Sache wieder ausgeschaltet ist. Der weiß nämlich dann überhaupt nichts davon.

Gerettet wird der Speicher ab 3000h aufwärts. Dadrunter sollte ein festes ROM sein, das gefälligst nicht geändert wird.

Einige interne Notierungen:
- die Umschaltung ist in die normale Interrupt-Routine des Gdos/Newdos-SYS0 eingebaut
- alle Routinen liegen in Bank 1
- die Banks 1,2 und 3 enthalten die "unteren" Hälften der Computer 1-3, jeweils von B000-FFFF
- die Banks 0 und 4-6 enthalten die "oberen" Hälften der Computer 0-3
- die "untere" Hälfte von Computer 0 wird jeweils mit der des momentan aktiven Computers getauscht
- es werden jeweils nur die "unteren" Hälften vertauscht (weil das Common nun mal eine "untere" Bank ist), während die "oberen" Hälften nur umgeschaltet werden müssen
- d.h.: wenn von Computer 1 auf 3 umgeschaltet wird, wird zuerst Computer 0 selektiert, bevor Computer 3 endgültig ans Rohr kommt
- deshalb spielt sich auf dem Bildschirm bei einer solchen Umschaltung kurz ein Wirrwarr ab, das aber nur mühsam zu umgehen wäre
- das Vertauschen läuft folgendermaßen ab:

```
  (von Bank 1 aus) 7E00-7FFF nach Bank 1 retten
                   Tausch-Routinen nach 7E00 (ins Common) übertragen
                   Tausch-Routinen starten
                          gerettetes 7E00 ff. mit dem aus sel. Bank tauschen
                          Common 3000-7DFF mit sel. Bank tauschen
                           (dabei Buffer 7F00-7FFF benutzt)
                          zurück nach Bank 1
                   geretteten Bereich 7E00 ff. ins Common
                   starten des Computers
```

- um unerwünschte Nebeneffekte (wie Druckerausgabe) auszuschließen, wird der I/O-Bereich 3700-37FF nicht gerettet
- das Programm ist lange nicht so elegant wie Arnulfs Lösung, aber es tut seinen Dienst

So das war's schon. besorgt Euch 256Kb und los geht's mit SIDEKIC

Gerald Schröder

Hardware: Helmut
Literatur: Arnulfs Listings

```
;        SIDEKICK für TRS80-Kompatible mit Helmut's Banker
;        im Dez. 1986 von Gerald Schröder
;        nach einer Idee von Arnulf Sopp


         ORG     5300h             ;Stack ab hier abwärts!

start    LD      (spbuf),SP        ;SP retten

         LD      HL,retpro         ;Rückkehr-Adresse (bei 1.
         PUSH    HL                ; Aufruf Comp. 1-3)
         LD      B,4               ;Stack vorbereiten
loop1    PUSH    AF
         DJNZ    loop1
         LD      (rettsp),SP       ;retten für Comp. 1-3

         LD      A,1               ;Bank 1
         DI
         OUT     (0ech),A          ;einschalten
         LD      HL,schalt         ;Umschaltroutine
         LD      DE,schalt-diff    ;in Bank 1
         LD      BC,ende-schalt
         LDIR
         XOR     A
         OUT     (0ech),A          ;wieder Bank 0

         LD      HL,umleit         ;Umleitung in Interrupt-
         LD      DE,umleit-diff1   ;Routine anlegen
         LD      BC,ende1-umleit
         LDIR

         LD      HL,zurück         ;Rücksprung
         LD      DE,zurück-diff2   ;in SYS0 anlegen
         LD      BC,ende2-zurück
         LDIR

         LD      SP,start          ;SP neu setzen

         LD      B,3               ;drei Computer
loop2    CALL    copy              ;in die Banks
         DJNZ    loop2

         EI
         LD      SP,0000           ;SP zurück
spbuf    EQU     $-2

retpro   RET                       ;ab


;        kopiert oberen und unteren Teil in Banks

copy     PUSH    BC                ;Zähler retten
```

```
        LD      A,B             ;als Bank-Nr.
        OUT     (0ech),A        ;für unterer Teil
        LD      DE,8000h+3000h  ;Ziel=0b000h
        LD      B,50h           ;Länge = 5000h
        LD      C,E
        LD      H,30h           ;Anfang = 3000h
        LD      L,E
        LDIR                    ;(3000-7fff)=>(b000-ffff)

        ADD     A,3             ;für oberer Teil
        EX      AF,AF'          ;retten
        LD      D,60h           ;Buffer: 6000h-6100h

loop3   INC     B               ;ein Sektor
        XOR     A               ;Bank 0
        OUT     (0ech),A        ;ein
        LDIR                    ;Sektor aus 0 in Buffer
        DEC     D               ;Sektor-Zeiger auf 6000h
        LD      A,H             ;Quelle- als Zielzeiger
        LD      H,D
        LD      D,A
        DEC     D
        INC     B
        EX      AF,AF'          ;Bank-Nr. zurück
        OUT     (0ech),A        ;anwählen
        EX      AF,AF'          ;wieder retten
        LDIR                    ;aus Buffer in Bank
        DEC     H               ;Buffer auf 6000h
        LD      A,D             ;und wieder tauschen
        LD      D,H
        LD      H,A
        OR      L               ;schon über ffffh hinaus?
        JR      NZ,loop3        ;nein
        OUT     (0ech),A        ;ja, Bank 0 ein
        POP     BC              ;Zähler zurück
        RET



;       Umleitung in der Interrupt-Routine

diff1   EQU     $-45f6h
umleit  LD      A,1             ;Bank 1
        OUT     (0ech),A        ;ein
        JP      schalt-diff     ;und anspringen
ende1   EQU     $


;       Rückkehr-Routine

diff2   EQU     $-4053h
zurück  OUT     (0ech),A        ;Bank x für oben an
        JP      45fdh           ;weiter
ende2   EQU     $



;       Schaltroutine in Bank 1
;       Ansprung von Interruot-Routine aus

```



```
diff    EQU     $-8000h
schalt  LD      HL,3880h        ;Tast.-zeile
        BIT     0,(HL)          ;SHIFT?
        JR      Z,goon3         ;nein, ab
        SRL     L               ;Tast.zeile 3840h
        BIT     4,(HL)          ;Down-Arrow?
        JR      Z,goon3         ;nein, ab
        SRL     L
        SRL     L               ;Tast.zeile 3810h
        LD      A,(HL)
        AND     0fh             ;Taste 0 bis 3?
        JR      Z,goon3         ;nein, ab
        LD      L,A             ;retten
        XOR     A               ;Zähler=0
goon1   INC     A               ;+1
        SRL     L               ;solange
        JR      NC,goon1        ;bis Taste erreicht
        DEC     A               ;Zähler korrigieren

        LD      (rettsp),SP     ;SP retten

        LD      DE,goon4-diff   ;Rückkehradresse
        JR      spsuch          ;neuen SP berechnen

goon4   PUSH    IX              ;Register dieses Comp.
        PUSH    IY              ;retten
        EX      AF,AF'
        EXX
        PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      SP,ende-diff+150        ;zeitweilig neu

        EX      AF,AF'          ;Comp.-Nr. zurück
        CP      0               ;schon eingeschaltet?
momba   EQU     $-1
        JR      Z,abgang        ;ja
        LD      (banku-diff),A  ;dort unterer Teil
        OR      A               ;Comp. 0?
        JR      Z,goon2         ;ja, oben = Bank 0
        ADD     A,3             ;sonst oben = u +3
goon2   LD      (banko-diff),A  ;Bank oberer Teil
        LD      A,(momba-diff)  ;Comp. 0 eingeschaltet?
        OR      A
        CALL    NZ,tausch-diff  ;nein, Comp. 0 holen
        LD      A,0             ;Bank unterer Teil
banku   EQU     $-1
        LD      (momba-diff),A  ;sichern: mom. Comp.
        OR      A               ;falls nicht 0:
        CALL    NZ,tausch-diff  ;Computer holen

abgang  LD      DE,goon5-diff
        JR      spsuch
goon5   LD      B,12            ;SP korrigieren
ab1     DEC     SP
        DJNZ    ab1
;                                Register des
        POP     HL              ;Computers zurück
        POP     DE
```

```
        POP     BC
        POP     AF
        EXX
        EX      AF,AF'
        POP     IY
        POP     IX
        LD      SP,(rettsp)     ;SP restaurieren

goon3   LD      A,(37ech)       ;urspr. Int.-rout.
        LD      A,(37e0h)
        RLCA
        LD      A,0             ;Bank-Nr. für oberer Teil
bankc   EQU     $-1
        JP      zurück-diff2    ;nach SYSO für Abgang


;       Unterprogramm: Stack-Pos. für geg. Comp. A
;       berechnen

spsuch  LD      L,A             ;Nr. nach L
        LD      H,0
        LD      SP,ende-diff+20 ;Stack-Offset
        LD      B,5             ;+32*Comp.-Nr.
sps1    SLA     L
        DJNZ    sps1
        ADD     HL,SP
        LD      SP,HL           ;SP in Bank 1 setzen
        EX      DE,HL           ;Rückspr.adr. nach HL
        JP      (HL)            ;und ab



;       Unterprogramm Tausch:
;       vertauscht unteren Teil von Computer 0 mit dem
;       eines der drei anderen Computer

tausch  LD      (bankx1-diff),A ;gew. Bank-Nr. sichern
        LD      (bankx2-diff),A
        LD      (bankx3-diff),A

        LD      HL,anf-diff3    ;Bereich Tausch-Rout.
        PUSH    HL              ;retten
        LD      DE,0a000h       ;Buffer für Original
        LD      BC,0200h
        LDIR                    ;Orig.-Code retten

        POP     DE              ;7e00h
        PUSH    DE
        LD      HL,anf-diff     ;Routinen zum Tauschen
        LD      BC,ende-anf
        LDIR                    ;in die untere Hälfte
;                                bei 7e00h
        LD      (stack-diff),SP ;SP retten
        LD      SP,7f00h        ;auch nach unten
        LD      HL,0fe00h       ;7e00h aus Bank x
        LD      DE,7f00h        ;Buffer
        LD      BC,9f00h        ;MSB (Buffer in Bank) -1
        CALL    anf-diff3       ;Aufruf Tausch-Rout.
        LD      SP,0000         ;SP zurück
stack   EQU     $-2
```

```
        POP     DE              ;7e00 zruück
        LD      H,9fh           ;von 9f00 aus
        INC     B               ;BC=0200h
        LDIR                    ;7e00-7fff neu füllen
        RET


diff3   EQU     $-7e00h
anf     PUSH    BC              ;9f00 (x-1) auf Stack
        PUSH    DE              ;7e00 (z)     "    "
        LD      B,1
        LD      A,2             ;2 Durchläufe

;7e00-7fff tauschen
loopp   EX      AF,AF'          ;Zähler retten
        LD      A,0
bankx1  EQU     $-1
        OUT     (0ech),A
        POP     DE              ;z zurück
        POP     AF              ;(x-1) zurück
        PUSH    HL              ;y retten
        PUSH    AF              ;(x-1) retten
        PUSH    DE              ;z retten
        LDIR                    ;y -> z, Bank x -> Buffer
        INC     B
        LD      A,1
        OUT     (0ech),A
        POP     HL              ;z zurück
        POP     DE              ;(x-1) zurück
        PUSH    HL              ;z retten
        LDIR                    ;z -> (x-1)
        INC     B               ; Buffer -> Bank 1
        LD      H,D             ;x
        POP     DE              ;z zurück
        PUSH    HL              ;x retten
        PUSH    DE              ;z retten
        LDIR                    ;x -> z, Bank 1 -> Buffer
        INC     B               ; (Original-Code)
        LD      A,0
bankx2  EQU     $-1
        OUT     (0ech),A
        POP     HL              ;z zurück
        POP     AF              ;x zurück
        POP     DE              ;y zurück
        PUSH    AF              ;x retten
        PUSH    HL              ;z retten
        LDIR                    ;z -> y, Buffer -> Bank x
        INC     B               ; (Orig. in Bank x)
        LD      H,D             ;(y+1)
        EX      AF,AF'
        DEC     A
        JR      NZ,loopp

;Stack: (x+1),z

        POP     DE              ;Buffer z
        POP     AF              ;(x+1) weg
        LD      H,30h           ;Anfang mit unten (3000h)
        LD      A,0
bankx3  EQU     $-1
```

```
        OUT     (0ech),A        ;Bank x ein

loop    PUSH    DE              ;"Buffer z" retten
        PUSH    HL              ;"von y" retten
        LDIR                    ;1.: Sektor in Buffer

        POP     DE              ;"nach y" zurück
        LD      H,D             ;"von x" (aus Bank)
        SET     7,H             ;deshalb: immer >=8000h
        PUSH    HL              ;retten
        INC     B               ;ein Sektor
        LDIR                    ;2.: aus Bank nach unten

        POP     DE              ;"nach x" (Bank)
        POP     HL              ;"von z" (Buffer)
        INC     B               ;ein Sektor
        LDIR                    ;aus Buffer in Bank

        INC     B               ;ein Sektor
        DEC     H               ;Buffer bleibt
        LD      A,D             ;"von y":
        AND     7fh             ;immer unter 8000h
        LD      D,H             ;Buffer in DE
        LD      H,A             ;"von y" in HL
        CP      37h             ;I/O-Bereich?
        JR      NZ,lop          ;nein, weiter
        INC     H               ;ja, überspringen
lop     CP      7eh             ;=fertig?
        JR      NZ,loop         ;weiter, wenn <> 7e00

        LD      A,1             ;Bank 1
        OUT     (0ech),A        ;wieder ein
        RET                     ;fertig
ende    EQU     $



rettsp  EQU     4058h


        END     start
```

Türkisch: Iki karpuzu bir koltugu sigderamazsin.
Du kannst keine zwei Melonen unter einen Arm klemmen.
Man kann nicht auf zwei Hochzeiten gleichzeitig tanzen.

* * * BOOLE XOR ABEL * * *

(oder: "Ent- oder -weder: d a s war hier die Frage!")

Im vorletzten INFO verkündete Hartmut (unser Obermann) ein Preisausschreiben, für dessen Gewinner er eine Flasche Wein aussetzte.
Damit hat er in dankenswerter Weise einen "frischen Wind" in unser Clubleben gebracht, und ich möchte sagen, so ein Wettbewerb sollte eigentlich in jeder unserer INFO-Nummern stehen! Doch wer bezahlt den vielen Wein bei so vielen Könnern? Unsere Clubkasse wäre sicher bald in den berüchtigten "Roten Zahlen". Es kann also immer nur der Initiative des Einzelnen überlassen bleiben.

Wie Hartmut inzwischen mitteilte, gingen mehr als eine richtige Lösung ein. Da das ja zu erwarten war - denn die sog. "logischen Funktionen" (auch Logic Operands genannt) kann eigentlich jeder begreifen, der logisch denken kann; und eben das ist wohl ohnehin die Voraussetzung für die halbwegs seriöse Beschäftigung mit unserem Hobby! -

also: Da das zu erwarten war, hatte Hartmut den Preis zunächst demjenigen ausgesetzt, dessen Lösung als erste einging!
Um so nobler war es von ihm, letztenendes doch j e d e m Einsender einer richtigen Lösung je eine Flasche aus "seinem" Weinberg zu verpassen...
Ein **PROST!** auf den edlen Spender und meinen besonderen Dank auch noch einmal an dieser Stelle, Hartmut!

Nun aber kommt, wie so oft, hier "das dicke Ende"! Sehen wir uns doch noch einmal Hartmuts Aufgabenstellung im INFO-Heft 15 vom September 1986, Seite 10, an!

Dort heißt es wörtlich:

"Wie muß die BASIC-Zeile aussehen,
die mit Hilfe der Funktionen OR, AND und NOT
diese (XOR-)Verknüpfung realisiert?" !!!

Logisch, daß die Lösung sich auch auf diese Operatoren beschränken muß!

Die Einsender haben dies - sine ira et studio (Tacitus) sei es gesagt: mit einer Ausnahme - sorgfältig beachtet und mehr oder weniger geschickte Vorschläge angeboten. Soweit OK.

Die "eine Ausnahme" (na: wer wohl?) hat ein bißchen anders herum gedacht. Doch davon nachher.

Wenn man schon außer Logik-Altmeister *BOOLE* auch den norwegischen Algebra-Meister *ABEL* (1824 Traktat über die algebraische Gleichung 5.Grades) heranzieht und eine **SUBTRAKTION** zuhilfe nimmt (und auch dies akzeptiert wird), so müßte es auch erlaubt sein, eine noch wesentlich kürzere und somit "elegantere" Lösung ins Feld zu führen, die auf die Boole'sche Algebra ganz verzichtet:

Was wird denn verlangt? C soll immer dann Null sein, wenn A und B gleich sind; und es soll Eins sein, wenn sie verschieden sind. Daraus folgt sofort (nicht nur algebraisch, sondern auch "logisch"):

C = - (A<>B) !

Das Minuszeichen deshalb, weil für "Logisch Wahr" in der "BASIC-Algebra" '-1' statt '+1' erzeugt wird.

Schauen wir uns die entsprechende "WAHRHEITSTAFEL" (dies ist übrigens die korrekte Bezeichnung in der BOOLE'schen Algebra) an:

A	B	A<>B	C=-(A<>B)
0	0	0	0
0	1	-1	1
1	0	-1	1
1	1	0	0

Es kommt also genau das heraus, was "XOR" tut: eine Eins nur dann, wenn ENTWEDER A ODER B gleich Eins ist - aber Null, wenn BEIDE gleich, also auch gleich Eins sind: im Gegensatz zum "OR"-Operanden, der im letzteren Fall gleich Eins ist! * ("Logisch", ja?) *[Wer entdeckt den Trugschluß?] *)*

Das komplette Programmchen lautet somit:

```
10 CLS: FOR A = 0 TO 1 : FOR B = 0 TO 1      ← Hier liegt der Hund...
20 C = - (A <> B)
30 <L>PRINT A, B, C
40 NEXT B, A : END
```

Übrigens ein prima "Einzeiler"! Denn das geht natürlich alles auch in eine einzige Zeile. Ich habe es nur der Deutlichkeit halber getrennt.

Unser Arnulf (die "Ausnahme") hat mir im letzten INFO die undankbare Aufgabe übertragen, soz. den "Schiedsrichter" in der "Kontroverse" (klingt zu hart; gemeint ist der Strategie-Unterschied) zwischen ihm und Hartmut zu spielen. Ich wage es:

Beide sind aus ihrer jeweiligen Perspektive im Recht. Zwar beinten - ich meine: fußten beide auf *BOOLEs* schöner Algebra. Doch Boole (1815-1864) wußte gewiß noch nichts von Bits und schon gar nicht von Bytes. Seine Lehre befaßte sich mit der *Aussagenlogik*, und zwar so allgemein, daß aus ihr in unserem Jahrhundert die euch sattsam bekannte Mengenlehre entstand (mit welcher unsere Erstklässler einige Jahre spielen durften).

Um nun die von *BOOLE* definierten Logik-Operatoren auch schlichteren Menschen - z.B. Informatikern - verständlich zu machen, werden sie immer am bitteren BIT-Beispiel exerziert, und wer nur oder vorzugsweise BASIC macht, denkt hieran zuerst und zuletzt (so myself!)

Arnulf hingegen, der selbst zugibt, kein Jünger des BASIC zu sein, hat schon als GENIE-Baby (oder noch früher?) den süßen BYTE-Honig geschleckt! Diese Byte-Biene reproduziert in ihrem Geiste grundsätzlich wieder Bytes, denn er steht der CPU wesentlich näher als wir BASICisten - und die futtert bekanntlich keine Bi(scui)ts, sondern nur Bytes in ihrer Gänze, oft sogar zwei auf einmal (ich versuchte das bereits in meinem "1.Assembler-Programm" anzudeuten).

Hier lag also der tiefere Unterschied zwischen beiden Herren: Dem Byte-Beflissenen offenbart sich natürlich 256^2 mal soviel wie dem billigen BIT-Steller! Deshalb vermißte Arnulf eine integrale (alles umfassende) Lösung.

Allerdings: Hartmut hatte danach ja auch gar nicht gefragt! (s.o.)

Was heißt überhaupt: "BIT"?

Ich glaube: "BASIC Ist Trumpf".

Aber Glaube gehört woandershin. Z.B. in die Politik. Oder Religion. Bei BASIC nützt er nichts. Und schon gar nichts im Umgang mit Bytes. Diese bedeuten nämlich:

"Bißchen Ybung Tut Es Selten!"

Aber bei dieser Gelegenheit fällt mir etwas anderes ein, das nur indirekt mit logischen Operatoren, aber auch sehr viel mit Logik zu tun hat.

Rosenfelder beschreibt in seinem hervorragenden Buch "BASIC FASTER & BETTER" eine Kurzschreibweise für logische Bedingungen (oder Entscheidungen), die die oft umständliche Formulierung "IF...THEN...ELSE" überflüssig machen. Ich habe das Gefühl - na so'n Quatsch; ein Programmierer und Software-Freak "fühlt" nicht: er denkt (oder denkt nicht *=XOR!)*- also ich habe den Gedanken, daß sich hierin noch weniger Kollegen auskennen als mit den "logischen Operaturen"! Und deshalb habe ich im letzten INFO ein kleines Privatissimum darüber begonnen - allerdings ohne meine Weinvorräte anzupreisen, denn sie sind so alt, daß ich nicht für deren $(CH_2)_2(OH)_2$-Freiheit garantieren kann...

So läßt uns BOOLE
gar oft recht cool.
Denn schon der junge Recke ABEL
ergriff die Algebra beim Nabel!
Ich sage das ganz ohne Spott
mit Hochachtung für beid': KaJott.

*) Der Trugschluß:
Diese "Wahrheitstafel" gilt nur für A,B = 0 oder 1, nicht für beliebig große Zahlen! Schon bei 1-Byte-Zahlen (0 bis FFh=255d) gibt es unter 256*256 = 65536 Möglichkeiten nur 256 Fälle, in denen A=B, also nach meiner "Kurzgleichung" C=0 ist (die Wahrscheinlichkeit hierfür beträgt somit rd. 4 Promille, was ausnahmsweise mal zu wenig ist...), während in rd. 996 Promille der Fälle mein C=1 wird. Dies widerspricht der "Gleichverteilung", die obige Wahrheitstafel vorspiegelt.
XOR hingegen wird bei der Kombination zweier 1-Byte-Zahlen zwar auch in 256 Fällen gleich Null, nämlich immer dann, wenn A=B; aber in den übrigen 65280 Fällen wird es nicht 1, sondern kann alle Werte von 1 bis 255 annehmen!
- Meine schöne kurze Gleichung für C ist daher zwar "elegant" - aber ffaallsch, weil sie "XOR" nicht abdeckt!
(wie manches Weib...)

[L. UHLAND: "Nun muß sich alles, alles wenden...!" - aus: Frühlingslieder <!>] *KJ*

PS: Ich danke Arnulf für seinen Hinweis!

HEFT 17 Dezember 1986

Assembler kontra LISP: Die Türme von Hanoi

Das Problem: Gegeben sind drei "Bauplätze" A, B und C. Auf einem Platz befindet sich ein Stapel (Turm) von Scheiben, nach Größe aufgeschichtet (die größte zuunterst). Bewege den Turm von Platz A nach B unter Benutzung des Hilfsplatzes C und unter Einhaltung der Regeln:
- Es darf nur eine Scheibe zur Zeit bewegt werden.
- Es darf keine (größere) Scheibe auf eine kleinere gelegt werden.

Lösungshinweis: Benenne die Scheiben von 1 bis n (die größte heißt n). Das Ergebnis soll eine Liste der Züge sein, z.B. bei n=2: (1 A C)(2 A B)(1 C B). (soll bedeuten: Scheibe 1 von Turm A nach Turm C, dann Scheibe 2 ...)

(aus: Peter Schefe: "Informatik - Eine konstruktive Einführung")

So, Leute, mit diesem Problem wurde ich vor einigen Wochen konfrontiert, und zwar im Rahmen einer Erstsemster-Vorlesung Informatik bei dem Prof, der auch das oben genannte Buch verbrochen und bei uns armen Studenten dafür ganz schön kassiert hat. Nun bin ich zwar leidenschaftlicher Assembler-Freak, aber der Herr wollte den Kram gern in LISP. Falls Euch das nichts sagt: macht nichts. Das ist angeblich die Sprache der "Künstlichen Intelligenz", was immer das auch sein mag. Sollte sich jemand künstlich dafür interessieren, schreibe ich im nächsten Info gern mehr darüber - dann ist nämlich das Semester zuende und ich sollte es dann können. Aber dann müßt Ihr mich erst brieflich oder sonstwie dazu auffordern. Wenn Ihr Pech habt, schreib' ich auch so was dazu, aber verlaßt Euch nicht drauf.

Weiter im Text. Auf die Lösung kann man ungefähr so kommen (hat mir jedenfalls mein Übungsgruppen-Leiter gesagt): Der erste Arbeiter hat die Aufgabe bekommen, den Turm von A nach B zu versetzen, wobei er C benutzen darf. Nun sagt er sich: "Verdammt, es wäre besser, wenn alle Scheiben bis auf die letzte (die größte) auf C wären, dann könnte ich die unterste (größte) Scheibe nach B packen und müßte dann nur noch den Rest-Turm von C nach B schaffen." Also stellt er dem nächsten untergebenen Arbeiter die unangenehme Aufgabe: "Schaffe alle Scheiben bis auf die unterste von A nach C. Du kannst dabei B benutzen." Dieser Arbeiter denkt nun ähnlich wie sein Chef und delegiert die Aufgabe weiter, bis ein Arbeiter erreicht wird, der die kleinste Scheibe bewegen muß, was er sofort tun kann. Man könnte auch erst dann aufhören, wenn es keine zu bewegenden Scheiben mehr gibt, denn kann dieser Arbeiter sagen: "Danke, Chef. Aufgabe ist schon erledigt."
Die jeweiligen "Rest-Türme" werden nach dem gleichen Prinzip bewegt.

Die Lösung in LISP sieht ungefähr so aus:

```
(DE HANOI (Scheiben)
    (HAN1 Scheiben 'A 'B 'C NIL))

(DE HAN1 (Scheibe von nach Hilfsturm Ergebnis)
    (COND ((ZEROP Scheibe) Ergebnis)
          (T (HAN1 (SUB1 Scheibe)
                   von
                   Hilfsturm
                   nach
                   (CONS (LIST Scheibe von nach)
                         (HAN1 (SUB1 Scheibe)
                               Hilfsturm
                               nach
                               von
                               Ergebnis))))))
```

Dazu wäre folgendes zu sagen:
Es handelt sich bei dem verwendeten LISP um UCI-LISP, das auf einer DEC-10 läuft (noch, bald kommt das Ding auf den Schrotthaufen). Die (hier benutzten) eingebauten Funktionen:
(DE name (Lambda-Ausdruck)): definiert Funktion, die unter "name" aufgerufen werden kann.
'x=(QUOTE x): gibt an, daß "x" eine Konstante ist (keine Variable)
NIL: leere Liste oder Null oder Wahrheitswert FALSE
(COND ...): Konditional, eine Art IF...THEN...ELSE...
(ZEROP x): wird TRUE, wenn x=0 und FALSE, wenn x<>0
T: ist der Wahrheitswert TRUE, also wie (1=1)
(SUB1 x): vermindert x um 1 bzw. hat als Ergebnis (x-1)
(CONS x l): setzt das Element x vorne auf die Liste l auf bzw. hat als Ergebnis die Liste aus dem Element x und der Liste l
(LIST x y z ...): setzt aus den Elementen x... eine Liste zusammen bzw. hat als Ergebnis die Liste mit den Elementen x, y, z ...
HANOI, HAN1: zwei Funktionen bzw. deren Namen
Scheiben, Scheibe, von, nach, Hilfsturm, Ergebnis: Variablen

Wie Ihr seht, ist alles auf Listen und Funktionen aufgebaut. Die Arbeitsweise des Programms:
Durch Eingabe von "(HANOI x)", wobei x eine Zahl ist, wird das Programm gestartet. Die einzige Ausgabe ist das Ergebnis dieses Funktionsaufrufs (Funktion HANOI mit dem Argument x, also genauso wie f(x) oder cos(x)).
Als erstes erhält die Variable "Scheiben" den Wert x zugewiesen. Dann wird eine zweite Funktion (HAN1) aufgerufen, diesmal mit folgenden Zuweisungen:
- Scheibe := Anzahl der Scheiben
- von := "Von-Turm" = A
- nach := "Ziel-Turm" = B
- Hilfsturm := C
- Ergebnis := NIL (leere Liste, noch kein Zug)

Nun wird abgefragt, ob die zu bewegende Scheibe "0" ist (ZEROP Scheibe). Wenn dies zutrifft, braucht keine Scheibe bewegt zu werden ("Danke, Chef ...").
Ansonsten wird immer (durch das "T") ein etwas komplizierter, doppelt rekursiver Ausdruck ausgeführt: Es wird wieder die Funktion HAN1 aufgerufen, wobei die Anzahl der zu bewegenden Scheiben um eins vermindert wird, also ist das neue "scheibe":=(altes "scheibe" -1). "Nach"- und Hilfsturm werden vertauscht, denn danach sollen die Scheiben bis auf die größte auf dem Hilfsturm liegen. Also neues "von":=altes "von"; neues "nach":=altes "Hilfsturm" und neues "Hilfsturm":=altes "nach". Außerdem muß noch ein "Ergebnis" übergeben werden. Dieses Ergebnis besteht aus einer Liste (die mit "CONS" gebaut wird). Diese Liste hat als erstes Element den Zug, den der Arbeiter ausführt, nämlich die Scheibe "Scheibe" von Turm "von" nach Turm "nach" zu packen (diese Liste wird durch "LIST" erzeugt). Dahinter folgen noch alle Züge, die gemacht werden müssen, nachdem dieser Zug gemacht wurde, d.h. hier wird der "Restturm" von dem "Hilfsturm" endgültig zum "nach"-Turm bewegt. Also muß hier wieder die Funktion HAN1 aufgerufen werden, nur mit entsprechend vertauschten Parametern.
Mit anderen Worten: eigentlich drehen wir die Arbeit um. Zuerst wird der letzte Aufruf von HAN1 abgehandelt, es wird also der Restturm zum Zielturm gebracht, wobei das Ergebnis der Funktion die Liste der dazu nötigen Züge ist. Dann fügen wir vorne an die Liste den Zug an, den der Arbeiter machen sollen. Nun erst kann der zweite Aufruf von HAN1 erfolgen, bei dem als "Ergebnis" die Liste der danach folgenden Züge übergeben wird (die Anführungsstriche bedeuten: die Variable mit dem Namen "Ergebnis" ist gemeint). Diese Übergabe des Ergebnisses nennt man Akkumulieren.
Das hört sich alles sehr schwierig an, aber nach dem fünften Lesen müßte es langsam dämmern. Schließlich habe ich auch einige Minuten (Rechenzeit) und einige Stunden (Arbeitszeit) gebraucht, bis diese Lösung stand, so kurz sie auch ist. Und vorher habe ich etliche Stunden in der Vorlesung mein Sitzfleisch geschult.

Aber Ihr fragt Euch: was soll das Ganze? Will der mir etwa was beibringen? Weit gefehlt! Denn nun folgt der interessante Teil: die Lösung in anderen Programmiersprachen. In Pascal soll es ähnlich aussehen wie in LISP, aber den Artikel überlassen ich anderen, denn Pascal kommt erst im nächsten Semester dran. Aber die gute alte Maschinensprache kann das auch! Wer glaubt, daß Rekursionen nicht möglich sind, lasse sich mit nachfolgendem Programm eines besseren belehren. Es macht für die Rekursion intensiv vom Stack Gebrauch, der deswegen auch überwacht werden muß, sonst überschreibt er plötzlich das Programm.

Interessant ist nun natürlich der Vergleich: von der Länge her gewinnt LISP klar, aber sicherlich wäre das Teil compiliert ziemlich lang. Auf der DEC 10 haben wir aber nur einen Interpreter, also kann ich nichts darüber sagen. Aus dem gleichen Grunde fallen Geschwindigkeitsvergleiche flach, denn je nach Tageszeit sind bis zu 12 Leute gleichzeitig am Werkeln, was natürlich die Reaktionszeit herabsetzt und außerdem müssen die Daten erst mühsam von/zu den Terminals übertragen werden, was unheimlich verlangsamt. Aber eine Kapazitätsprüfung ist angebracht. Ohne besondere Platz-Reservierung schafft es die DEC 10 gerade bis zu einem Turm mit 8 Scheiben (in Worten: acht, immerhin ein Großrechner, auf dem die Rechenstunde 2500 DM kosten soll, wenn er auch nur 256 Kb Hauptspeicher (in 36-Bit-Worten) hat!). Mein kleiner IIs mit seinen 64Kb Hauptspeicher, von denen in diesem Programm nur 32Kb für den Stack benutzt werden dürfen, schafft 16 in ca. 2 Minuten. Da sich mit jeder neuen Scheibe die Rechenzeit verdoppelt, habe ich dort die Tests abgebrochen. Aber bei 64 Scheiben hat er ca. 20 min gerechnet (bis mir der Geduldsfaden riß), ohne einen Stacküberlauf zu produzieren. Ich würde sagen, daß die gesamte Rechenzeit bei 64 Scheiben ca. 800 Mio Jahre beträgt, kann das aber selbst nicht glauben und bin über jeden Gegenbeweis dankbar.

Außerdem warte ich auf weitere Vergleiche und eventuelle Anfragen wegen LISP.

Gerald Schröder

(Folgende Unterprogramme des nachfolgenden Listings stammen nicht von mir, sondern von Anton Hummel: HEXJ, PRTHX, ASC.)

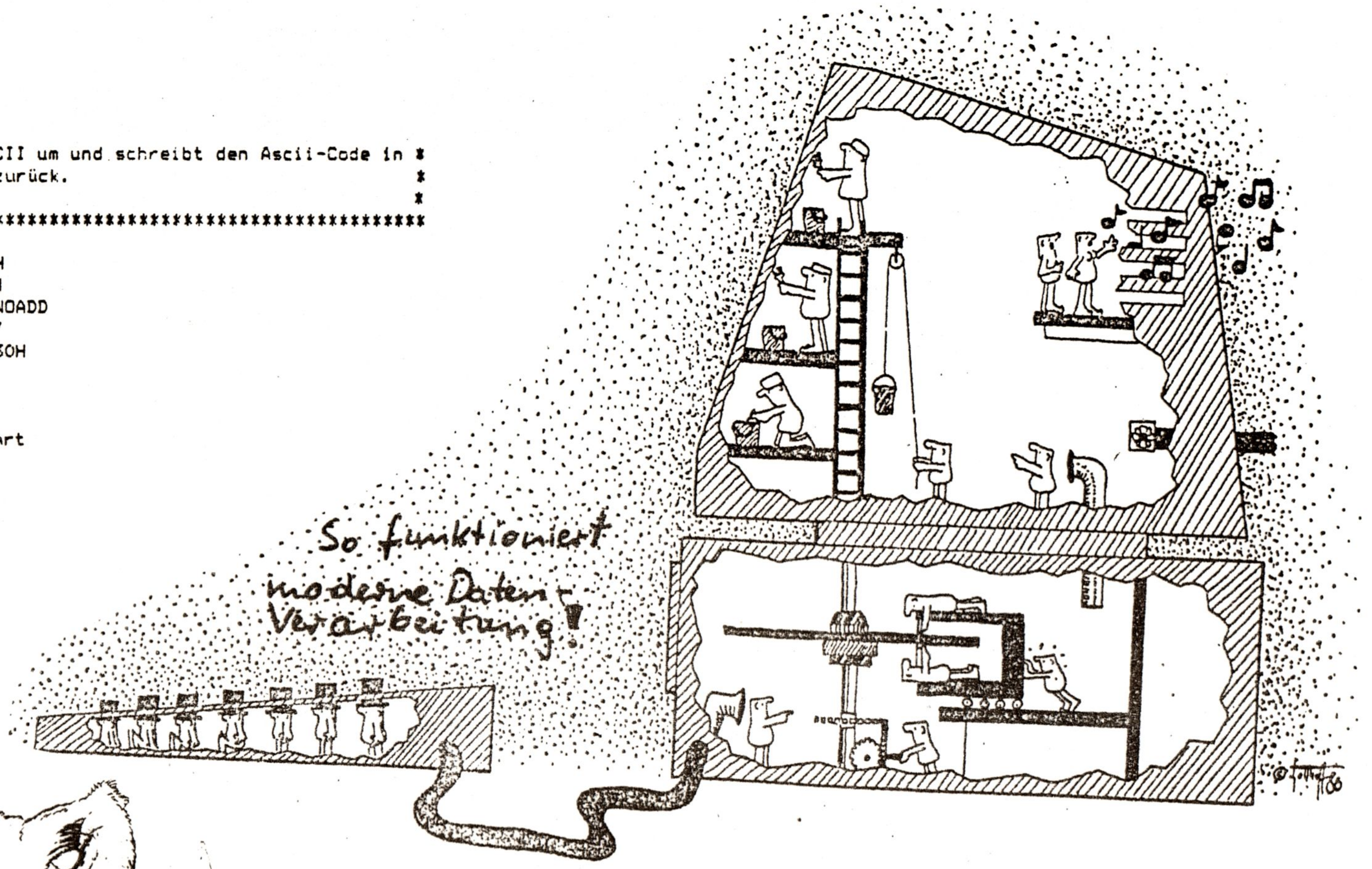
```
;         Hanoi
;         rekursive Lösung des Problems "Türme von Hanoi"
;         von Gerald Schröder

;         im Dez. '86 nach eigener LISP-Vorlage

;         Aufruf durch "HANOI/CMD xx", wobei xx die
;         Höhe des Turms in Hex angibt

out       EQU     33h            ;Ausgabe-Routine (o. 3Bh für Drucker)


          ORG     5200h
start     CALL    holarg         ;Zahl holen
          LD      B,A            ;Anzahl Scheiben nach B
          CALL    PRTHX          ;Höhe ausgeben
          LD      A,0dh          ;mit CR
          CALL    out
          LD      (stack),SP     ;SP retten
          LD      SP,0000        ;neu setzen

          LD      H,'a'          ;"von"-Turm
          LD      L,'b'          ;"nach"-Turm
          LD      C,'c'          ;"Hilfsturm"
          LD      IY,3880h       ;für BREAK-Abfrage

          CALL    hanoi          ;Tausch aufrufen

          LD      A,0dh          ;CR
          CALL    out            ;ausgeben
ab        LD      SP,0000        ;SP zurück
stack     EQU     $-2
          RET                    ;ab ins DOS


;         ***************************************
;         * Haupt-Unterprogramm: Packe B Scheiben *
;         * von Turm H nach L mithilfe von C.     *
;         * dieser Teil ist rekursiv!!!           *
;         ***************************************

hanoi     INC     B              ;0 Scheiben?
          DEC     B
          RET     Z              ;ja fertig
          BIT     0,(IY+0)       ;Shift?
          JP      NZ,402dh       ;ja, BREAK
          PUSH    HL             ;rette von/nach
          PUSH    BC             ;rette Hilfst., Scheiben

          CALL    sphigh         ;ab, wenn Stack zu groß
          DEC     B              ;eine Scheibe weniger
          LD      A,C
          LD      C,L            ;"nach" als "Hilfsturm"
          LD      L,A            ;"Hilfsturm" als "nach"
          CALL    hanoi          ;rekursiver Aufruf

          POP     BC             ;Hilfsturm, Scheiben
          POP     HL             ;von/nach
          CALL    ausgabe        ;Bewegung ausgeben

          DEC     B              ;eine Scheibe weniger
          LD      A,C
          LD      C,H            ;"von" als "Hilfsturm"
          LD      H,A            ;"Hilfsturm" als "von"
          JP      hanoi          ;Abgang mit Tauschen


;         ***************************************
;         * gibt Tauschaktion aus im Format:    *
;         *        (xx v n)                     *
;         * wobei xx = Nr. der Scheibe (in hex) *
;         *        v = von Turm (a, b oder c)   *
;         *        n = nach Turm (a, b oder c)  *
;         ***************************************


ausgabe   LD      A,'('          ;Klammer-auf
          CALL    out            ;ausgeben
          LD      A,B            ;Scheiben-Nr. nach A
          CALL    PRTHX          ;in Hex ausgeben
          LD      A,20h          ;Leerzeichen
          CALL    out            ;ausgeben
          LD      A,H            ;von-Turm
          CALL    out            ;ausgeben
          LD      A,20h          ;Leerzeichen
          CALL    out            ;ausgeben
          LD      A,L            ;nach-Turm
```

```
          CALL     out                  ;ausgeben
          LD       A,')'                ;Klammer-zu
          CALL     out                  ;ausgeben
          LD       A,20h                ;Leerzeichen
          CALL     out                  ;ausgeben
          RET


;         ******************************************
;         * überprüft, ab SP unterhalb von 8000h *
;         * wenn ja: zu groß, ab                 *
;         ******************************************

sphigh    PUSH     HL
          LD       H,0
          LD       L,H
          ADD      HL,SP
          BIT      7,H                  ;>=8000h?
          POP      HL
          RET      NZ                   ;ja, Bit 7 = 1

          LD       HL,spmes             ;nein, Meldung
          CALL     4467h                ;ausgeben
          JR       ab                   ;und Abgang

spmes     DM       0ah,'Stack zu groß! Abbruch.',0dh



;         **********************************
;         * holt Argument (Höhe des Turms *
;         * in Scheiben, Hex-Zahl)        *
;         **********************************

holarg    CALL     4cd5h                ;nächstes Zeichen
          JR       Z,error              ;=CR: Fehler
          LD       A,(HL)               ;holen
          CALL     wertaus              ;Zahl auswerten
          LD       A,B                  ;Ergebnis nach A
          RET      NC                   ;kein Fehler: fertig

error     LD       HL,errmes            ;bei Fehler:
          CALL     4467h                ;Meldung ausgeben
          JP       402dh                ;ab ins DOS

errmes    DM       'Fehler: kein oder falsches Argument!',0dh


;         ***********************************************
;         * wertet ein- oder zweistellige Hex-Zahl aus *
;         ***********************************************


wertaus   CALL     HEXJ                 ;ASCII in binär
          RET      C                    ;Fehler: zurück
          LD       B,A                  ;nach B retten (einst.)
          RRCA                          ;falls zweistellig:
          RRCA                          ;höherwertiges Nibble
          RRCA                          ;erzeugen
          RRCA
          LD       C,A                  ;und nach C
```



```
          INC      HL
          LD       A,(HL)               ;nächstes Zeichen
          CP       0dh
          RET      Z                    ;CR: fertig
          CALL     HEXJ                 ;in binär
          RET      C                    ;falsch: Fehler
          OR       C                    ;sonst höheres N. dazu
          LD       B,A                  ;für Ergebnis-Ausgabe
          RET


;*********************************************************
;*                                                       *
;* CALL HEXJ                        geschützte Register: *
;*                                  alle außer AF        *
;* Wandelt Ascii-Zeichen aus dem Register A in Binär-    *
;* zahl um und speichert Ergebnis in A.                  *
;* Ausgabe-Flags: C=0 = alles Ok                         *
;*                C=1 = Fehler                           *
;*********************************************************


HEXJ      CP       30H
          RET      C
          CP       3AH
          JR       C,HEX1
          SUB      7
          CP       40H
          JR       NC,HEX2
HEX1      AND      0FH
          RET
HEX2      SCF
          RET

;*********************************************************
;*                                                       *
;* CALL PRTHX                       GESCHÜTZTE REGISTER: *
;*                                  ALLE AUßER AF        *
;* Zeigt den Inhalt des Registers A als 2-stellige       *
;* Hexadezimalzahl auf dem Bildschirm an.                *
;*                                                       *
;*********************************************************


PRTHX     PUSH     AF
          RRCA
          RRCA
          RRCA
          RRCA
          CALL     ASC
          CALL     out      ;(AUSGABE)
          POP      AF
          CALL     ASC
          JP       out


;*********************************************************
;*                                                       *
;* CALL ASC                         geschützte Register: *
;*                                  alle außer AF        *
;* Wandelt den Inhalt der unteren 4 Bit des Registers A *
```

```
;* von Binär in ASCII um und schreibt den Ascii-Code in *
;* das Register A zurück.                               *
;*                                                      *
;********************************************************

ASC     AND     0FH
        CP      0AH
        JR      C,NOADD
        ADD     A,7
NOADD   ADD     A,30H
        RET


        END     start
```



*Erklär **du** ihm doch, daß wir Pac-Man nicht haben!...*

Computerlexikon:

directory

In gewöhnlichem Englisch bedeutet das Wort Adreßbuch oder auch Telefonbuch, im Computer-Englisch dagegen ist die Bedeutung eng begrenzt. Das directory ist das Inhaltsverzeichnis aller auf einem Massenspeicher abgespeicherten Dateien, sei es nun ein Magnetband, eine Floppy oder eine Festplatte. Bei einem Personal Computer bewirkt der Befehl DIR, wenn man im Betriebssystem arbeitet, daß sämtliche Dateinamen aufgelistet werden, meist mit dem Umfang des belegten Speicherplatzes; also der Anzahl der Bytes, die sie einnehmen. Das erleichtert nicht nur das Suchen, sondern zeigt auch gleich, wieviel Speicherplatz noch frei ist. Natürlich kennen auch Heimcomputer so einen Befehl, doch ist er hier von Gerätetyp zu Gerätetyp verschieden. Will man bei dem weit verbreiteten C 64 einen Blick ins »Inhaltsverzeichnis« einer Diskette tun, genügt der Befehl LOAD "$",8 schon hat man alles auf dem Bildschirm. ★

Z80, HD64180 und Illegals im ROM-Modus

Das Herz unseres kleinen Haufens Silikonmasse besteht bekanntermaßen aus einem Z80. Dieser Z80 kann über 1100 Befehle verarbeiten, von denen aber nur 694 von den Erbauern vorgesehen waren. Der Rest ist also Abfallprodukt.
Nun gibt es einen neuen Prozessor, den HD64180, der Z80-kompatibel ist, aber einge Sachen mehr kann (512Kb RAM verwalten, DMA, Schnittstelle...). Nur haben die Erbauer dieses Dings leider vergessen, die Befehle des Z80 zu berücksichtigen, die nicht offiziell dokumentiert sind. Das kann böse Probleme geben, denn wenn der neue Proz auf einen ihm unbekannten Befehl trifft, macht er einen Brutalo-Reset nach 0000h, was in unserem Fall jedes Programm ins Jenseits befördert.
Es wäre dies kein Problem, wenn niemand die "Illegals" bzw. die nicht dokumentierten Befehle benutzen würde, aber viele Programmierer tun es doch. Prominentes Beispiel: der Editor/Assembler ZEUS (3 Illegals).

Ihr könnt nun
1. den neuen Prozessor vergessen (sehr schwer!),
2. alle Programme mit Illegals vergessen (nicht ganz so schwer),
3. alle Illegals durch normale Befehle ersetzen (schwer) oder
4. dem HD64180 die unbekannten Befehle beibringen (fast leicht).

Punkt 3 erschien mir brauchbar, bis sich ZEUS nach den Änderungen so merkwürdig benahm. Leider sind die Ersatz-Befehle für die Illegals nämlich einen kleinen Tick länger als die Illegals und das gibt big problems.
Also auf zu Punkt 4! Vorraussetzung: Veränderliches RAM/ROM bei 0000h: bei jedem CP/M-fähigen Rechner der Fall.

Außerdem habe ich in meinem kleinen Programm noch einiges ausgenutzt:

Arnulf hett seggt, daß RST 00h nie gebraucht wird. Also benutze man RST 00h für etwas anderes. In diesem Fall bot sich ein Banking an, das Helmuts alten Banker simuliert. Verfahrensweise: Der Speicher von 8000h-ffffh wird als Common 1 definiert und je nach gewünschter Bank wird der Offset von Common 1 neu gesetzt. Bank 0 ist die normale Bank, also sind dann Interrupts erlaubt. Ansonsten werden die Interrupts gesperrt. Die Bank-Nr. wird vor dem RST 00h in A geladen.
Mein kleiner IIs kennt ein Parallel-RAM. D.h. Neben Tastatur, Bildschirm etc., die ja memory-mapped-I/O sind, gibt es einen RAM-Bereich von 3400h-3fffh. Je nach Wunsch kann ich nun entweder Ein-/Ausgaben vornehmen oder in diesem Extra-RAM einige zusätzliche Programme laufen lassen. Wiederum müßte jeder TRS80-CP/M-Rechner irgendetwas in der Richtung haben.
Nun versteht Ihr auch, warum ich in der Überschrift von "ROM-Modus" sprach. Der ganze Kram klappt bei CP/M in dieser Form nicht. Dafür andere Lösungen zu suchen, überlasse ich den CP/M-Freaks.
Zu guter Letzt kann ich auch das ROM/RAM überschreiben (s. wiederum CP/M-fähige Rechner). Da sich in dem ROM allerhand Müll angesammelt hat, kann man gewisse Teile ohne Skrupel killen, was ich auch getan habe.

Mit diesen ganzen Einschränkungen wird das Programm zwar sehr auf eine Maschine festgelegt, aber wer wird sich schon einen HD64180 anschaffen? Wohl doch nur Leute, die sowieso am Basteln sind und bei deren Drahtansammlungen nicht viel vom TRS80 übrig geblieben ist.

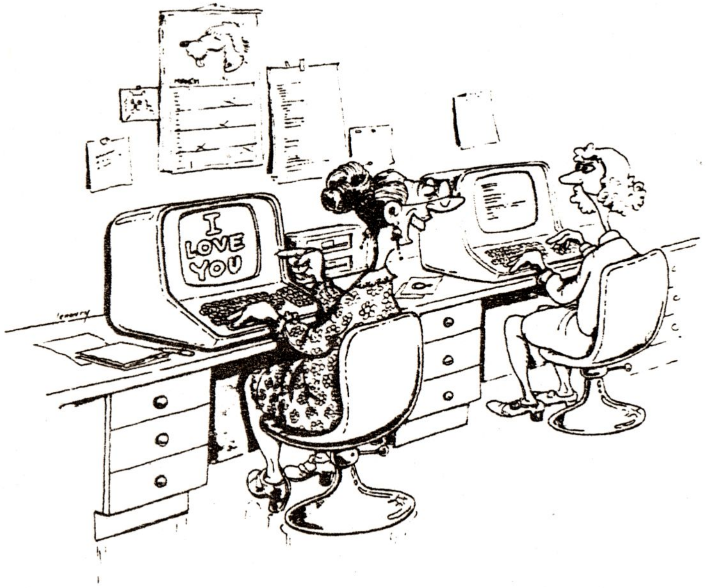
Noch kurz zum eigentlichen Programm: Der erste Teil ist nur der Ansprung der Routine von 0000h aus. Der zweite Teil ist ein Ausgang aus der Routine, bei dem die Interrupts ausgeschaltet bleiben. Die anderen Ein/Aus-Routinen stellt OVL3/SYS aus dem Gdos 2.4 zur Verfügung. Der dritte Teil ist derjenige, der beim Aufruf die Routine ins Parallel-RAM bringt. Danach folgt die eigentliche Routine, die im gegenwärtigen Zustand noch nicht mal 100h Bytes lang ist. Sie berücksichtigt alle Illegals, die in einem Artikel der "data welt" im Mai/Juni 1986 erwähnt wurden.

Die HD64180-spezifischen Befehle INO/OUTO wurden über DBs realisiert, weil mein alter ZEUS sie natürlich nicht kennt.

Der Programmierstil ist leider sehr "unschön", wie mein Prof sagen würde, vor allem aufgrund der vielen Patches direkt im Programmablauf, aber es tut seinen Dienst.

Gerald Schröder

Hardware: Helmut Bernhardt
Literatur: "TAV's Z80 Tuning" aus "data welt" 5 und 6 '86
Z80-Reference-Manual



Du siehst, ich bin immer noch attraktiv!

```
                ;       Error-Trap für Genie IIs mit HD64180
                ;       Weihnachten '86 by Gerald Schröder


06A0            pramon  EQU     06a0h           ;schaltet Par.-RAM ein
06AB            pramoff EQU     06abh           ;schaltet Par.-RAM aus


                ;-----------------------------------------------------------

                ;       0000: Ansprung Error-Trap und RST 00h

0000                    ORG     0
0000  CDA006            CALL    pramon          ;Par.-RAM ein
0003  C30034            JP      routin-offset           ;anspringen


                ;-----------------------------------------------------------

                ;       neuer Ausgang: Par.-RAM aus und Interrupts
                ;       bleiben disabled

06D2                    ORG     06d2h
06D2  DBFE      prama   IN      A,(0feh)
06D4  CB87              RES     0,A
06D6  D3FE              OUT     (0feh),A
06D8  F1                POP     AF
06D9  C9                RET


                ;-----------------------------------------------------------

                ;       Initialisierungsroutine

5200                    ORG     5200H
5200  3E80      start   LD      A,80h           ;Common 1 ab 8000h
5202  ED                DB      0edh,39h,3ah    ;setzen (OUTO (3ah),A)

5205  CDA006            CALL    pramon          ;Par.-RAM ein
5208  211652            LD      HL,routin       ;Routine
520B  110034            LD      DE,routin-offset        ;dorthin
520E  01C900            LD      BC,ende-routin          ;übertragen
5211  EDB0              LDIR
5213  C3AB06            JP      pramoff         ;Par.-RAM aus und ab


                ;-----------------------------------------------------------

                ;       Routine im Par.-RAM für Error-Trap und RST 00h
                ;       bei Error: fehlende Codes simulieren
                ;       bei RST 00h: Bank A einschalten

1E16            offset  EQU     $-3400h
5216  F5        routin  PUSH    AF
5217  ED                DB      0edh,38h,34h    ;INO A,(34h)
521A  CB7F              BIT     7,A             ;TRAP gesetzt?
521C  2010              JR      NZ,illopc       ;ja, illegaler Opcode

                ;       *******************************
                ;       * Banking: Bank A einschalten *
                ;       *******************************

521E  F1                POP     AF
521F  F5                PUSH    AF                      ;retten
5220  07                RLCA
5221  07                RLCA                            ;korr. Offset erzeugen
5222  07                RLCA                            ;(Basis Common 1 neu)
5223  ED                DB      0edh,39h,38h            ;OUTO (38h),A
5226  B7                OR      A                       ;Bank 0?
5227  C2D206            JP      NZ,prama                ;nein, keine Interrupts
522A  F1                POP     AF
522B  C3AB06            JP      pramoff                 ;Interrupts ein



                ;       *******************************
                ;       * illegaler Opcode: simulieren *
                ;       *******************************

522E  CBBF      illopc  RES     7,A                     ;TRAP löschen
5230  ED                DB      0edh,39h,34h            ;OUTO (34h),A
5233  322534            LD      (ufo-offset),A          ;UFO retten
5236  F1                POP     AF                      ;A vom Stack
5237  E3                EX      (SP),HL                 ;HL mit Err.-Adr. tausch.
5238  F5                PUSH    AF                      ;A wieder retten
5239  C5                PUSH    BC                      ;BC als Hilfsregister
523A  3E00              LD      A,0                     ;UFO
523B            ufo     EQU     $-1
523C  CB77              BIT     6,A                     ;gesetzt?
523E  2801              JR      Z,nosub                 ;nein, Opc. direkt davor
5240  2B                DEC     HL
5241  2B        nosub   DEC     HL                      ;Zeiger auf ill. Opc.
5242  7E                LD      A,(HL)                  ;laden
5243  47                LD      B,A                     ;und retten
5244  FECB              CP      0cbh                    ;CBxx? (SLIA)
5246  23                INC     HL                      ;Zeiger weiter
5247  7E                LD      A,(HL)                  ;2. Byte laden
5248  200F              JR      NZ,ddfd                 ;nein, Oper. mit IX/Y

                ;       ***********************************
                ;       * illegaler Opcode: CBxx = SLIA x *
                ;       ***********************************

524A  D620              SUB     20h                     ;korrigieren
524C  323F34            LD      (CBxx-offset),A         ;in den Programm-Text

524F  C1                POP     BC                      ;Register zurück
5250  F1                POP     AF
5251  23                INC     HL                      ;PC neu setzen
5252  E3                EX      (SP),HL                 ;und HL zurück
5253  37                SCF                             ;für invertiert!!!
5254  CB10              RL      B                       ;und rotieren
5255            CBxx    EQU     $-1                     ;10-17 für Register B-A
5256  C3AB06    ab      JP      pramoff                 ;weiter im Programm


                ;       ***************************************
                ;       * illegaler Opcode: erstes Byte DD/FD *
                ;       ***************************************

```

```
5259  FECB            ddfd    CP      0cbh                 ;danach CBh?
525B  2828                    JR      Z,ddfdcb             ;ja
525D  326834                  LD      (ersatz-offset),A         ;in den Text
5260  FE26                    CP      26h                  ;"LD hX/Y,konstante"?
5262  2807                    JR      Z,ldkon              ;ja
5264  FE2E                    CP      2eh                  ;"LD lX/Y,konstante"?
5266  2803                    JR      Z,ldkon              ;ja

5268  AF                      XOR     A                    ;keine Konstante: NOP
5269  1802                    JR      weiter

                      ;       Konstante nach MSB/LSB von IX/Y

526B  23              ldkon   INC     HL                   ;Zeiger auf Konstante
526C  7E                      LD      A,(HL)               ;diese laden

526D  326934          weiter  LD      (konst-offset),A          ;u. in den Text
5270  78                      LD      A,B                  ;DD/FD zurück
5271  326534                  LD      (ddfd1-offset),A          ;setzen
5274  326B34                  LD      (ddfd2-offset),A

5277  C1                      POP     BC                   ;alle Register zurück
5278  F1                      POP     AF
5279  23                      INC     HL                   ;PC neu
527A  E3                      EX      (SP),HL              ;HL zurück

527B  DDE5            ddfd1   PUSH    IX                   ;bzw. IY
527D  E3                      EX      (SP),HL              ;nach HL
527E  7F              ersatz  LD      A,A                  ;Operation ausführen
527F  00              konst   NOP                          ;evtl. Konstante
5280  E3                      EX      (SP),HL              ;neues IX/Y zurück
5281  DDE1            ddfd2   POP     IX                   ;bzw. IY
5283  18D1                    JR      ab


                      ;       ************************************
                      ;       * DD/FD gefolgt von CB             *
                      ;       * RLC/RRC/RL/RR/SLA/SRA/(SLIA)/SRL, *
                      ;       *  (IX/Y+d),B/C/D/E/H/L/A          *
                      ;       ************************************

5285  78              ddfdcb  LD      A,B                  ;DD/FD zurück
5286  32C034                  LD      (ddfd3-offset),A          ;setzen
5289  32C534                  LD      (ddfd4-offset),A
528C  23                      INC     HL                   ;Zeiger auf Offset
528D  7E                      LD      A,(HL)               ;"d" aus (IX/Y+d) laden
528E  32C234                  LD      (offs1-offset),A          ;setzen
5291  32C734                  LD      (offs2-offset),A
5294  23                      INC     HL                   ;Zeiger auf Operation

                      ;       Lade-Befehl errechnen und in den Text

5295  7E                      LD      A,(HL)               ;Befehl laden
5296  E607                    AND     7                    ;nur die Bits 0-2
5298  3C                      INC     A                    ;+1
5299  47                      LD      B,A                  ;als Zähler
529A  3E3E                    LD      A,3eh                ;Opcode 46h-7eh erzeugen

529C  C608            loop1   ADD     A,B                  ;(46/4e/56/5e/66/6e/7e)
529E  10FC                    DJNZ    loop1                ;=LD B/C/D/E/H/L/A,(IX..)
```



```
52A0  32C634                  LD      (lade-offset),A ;setzen

                      ;       Rotations-/Schiebe-Befehl erzeugen

52A3  7E                      LD      A,(HL)
52A4  E6F8                    AND     0f8h                 ;die Bits 0-2 löschen
52A6  0F                      RRCA                         ;und raus rotieren
52A7  0F                      RRCA
52A8  0F                      RRCA
52A9  3C                      INC     A                    ;+1
52AA  47                      LD      B,A                  ;als Zähler
52AB  3EFE                    LD      A,0feh               ;6/e/16/1e/26/2e/3e=RLC,

52AD  C608            loop2   ADD     A,B                  ;RRC,RL,RR,SLA,SRA,SRL
52AF  10FC                    DJNZ    loop2                ;(u. RES/SET x)

52B1  48                      LD      C,B                  ;NOP erzeugen (für reta)

52B2  FE36                    CP      36h                  ;SLIA?
52B4  2008                    JR      NZ,noSLIA            ;nein

                      ;       SLIA (IX/Y+d) oder SLIA (IX/Y+d),r

52B6  0637                    LD      B,37h                ;SCF erzeugen
52B8  7E                      LD      A,(HL)               ;SLIA (IX/Y+d)?
52B9  FE36                    CP      36h
52BB  2002                    JR      NZ,nopro             ;nein, kein Problem
52BD  0EC9                    LD      C,0c9h               ;RET für reta erzeugen

52BF  3E16            nopro   LD      A,16h                ;RL erzeugen (statt SLIA)

52C1  32C334          noSLIA  LD      (opera-offset),A          ;setzen
52C4  78                      LD      A,B                  ;SCF oder NOP
52C5  32BF34                  LD      (cflag-offset),A          ;setzen
52C8  79                      LD      A,C                  ;RET oder NOP
52C9  32C434                  LD      (reta-offset),A ;setzen

52CC  C1                      POP     BC                   ;Register zurück
52CD  F1                      POP     AF
52CE  23                      INC     HL                   ;PC korrigieren
52CF  E3                      EX      (SP),HL              ;und HL zurück

52D0  CDBF34                  CALL    cflag-offset         ;Operation ausführen
52D3  1881                    JR      ab

                      ;       Unterprogramm: Operation simulieren

52D5  00              cflag   NOP                          ;oder SCF

52D6  00              ddfd3   NOP                          ;DD o. FD für IX/Y
52D7  CB                      DB      0cbh                 ;immer CBh
52D8  00              offs1   NOP                          ;Offset d (IX/Y+d)
52D9  00              opera   NOP                          ;Operationsanweisung

52DA  00              reta    NOP                          ;RETurn bei SLIA (IX/Y+d)

52DB  00              ddfd4   NOP                          ;DD/FD für IX/Y
52DC  7F              lade    LD      A,A                  ;Ladebefehl
52DD  00              offs2   NOP                          ;Offset (s. offs1)
52DE  C9                      RET

52DF                  ende    EQU     $



5200                          END     start

00000 Fehler
```

TAV's Z80 Tuning

Illegals

Die Z80-CPU ist mit ihren 694 Befehlen ein überaus leistungsfähiger Mikroprozessor. Dieser riesige Befehlsschatz läßt sich durch sogenannte "Illegals" auf über 1100 Op-Codes erweitern. Ein Illegal ist ein vom Hersteller nicht definierter Op-Code, der normalerweise eine Fehlfunktion des Prozessors verursacht. Viele dieser Fehlfunktionen repräsentieren jedoch eine sinnvolle Befehlserweiterung.

Im Bereich der Realtime-Programmierung kann es vorkommen, daß eine Routine einige tausendmal innerhalb einer Programmschleife aufgerufen wird. In solchen Fällen macht sich das Einsparen jedes Taktzyklus bemerkbar. Ein gezielter Einsatz der folgenden Operationen kann insbesondere dazu beitragen, kritische Bereiche zu unterschreiten und somit ungewollte Fehlfunktionen eines Programmes zu vermeiden. Weiterhin können in Grenzfällen Speicherplatzprobleme gelöst werden, da ein Illegal mindestens durch zwei konventionelle Befehle ersetzt werden muß.

Illegals lassen sich darüberhinaus hervorragend in einen Programmschutz einbauen, da sie kein mir bekannter Disassembler in sinnvoller Form interpretieren kann.

Neue Shift- und Rotations-Befehle

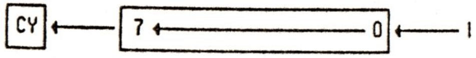
Als erstes möchte ich Ihnen eine gänzlich neue Shift-Operation vorstellen - die **SLIA**-Funktion. (Shift Left Inverted Arithmetic). Sie entspricht der **SLA**-Operation mit dem Unterschied, daß hierbei das relative Bit 0 mit logisch 1 (High) aufgefüllt wird.

Bitte beachten Sie, daß ihr Assembler die oben vorgestellten symbolischen Befehlscodes nicht direkt verstehen kann. Bei der Programmierung müssen die ihnen zugeordneten Op-Codes mittels den DEFB- und DEFW-Anweisungen folgendermaßen in den Source-Code implementiert werden.

Beispiel:

```
LD       A,(MEM1)
;Hohe Testzahl
DEFW     •37CB
;SLIA A (Testzahl=Testzahl•2+1)
LD       (MEM1),A
;Ergebnis zurückschreiben
RET ;Test beendet
```

In diesem Beispiel wird ein beliebiger Testwert aus der Speicherzelle MEM1 in den Akkumulator geladen, mittels SLIA A Operation manipuliert und in die Speicherzelle MEM1 zurückgeschrieben. Die Implementierung des SLIA-Befehles erfolgte in unserem Beispiel mit Hilfe der **DEFW**-Anweisung. Es ist in diesem Zusammenhang unbedingt zu beachten, daß High- und Low-Byte des benötigten 2-Byte Op-Codes vertauscht werden müssen (siehe Tabelle). Falls die **DEFB**-Anweisung verwendet werden sollte, müssen die Op-Codes in normaler Reihenfolge eingegeben werden.
An dieser Stelle endet der 1. Teil über Z80-Operationen, die in keinem Handbuch stehen. Im nächsten Monat werden wir uns unter anderem mit neuen Befehlen zur gezielten High/Low-Byte Manipulation der 16-Bit Index-Register beschäftigen. Bis dahin wünsche ich Ihnen viel Spaß beim Experimentieren mit der Z80-CPU und immer ein Bit zuviel.(tav) ●



Name	Op-Code	Name	Op-Code
SLIA B	CB 30	SLIA C	CB 31
SLIA D	CB 32	SLIA E	CB 33
SLIA H	CB 34	SLIA L	CB 35
SLIA (HL)	CB 36	SLIA A	CB 37

Links orientiertes indiziert-indirektes invertiert-arithmetisches "Shiften" mit Transfer des Endergebnisses in ein beliebiges 8-Bit Register.

Name:	Op-Code	Name	Op-Code
SLIA (IX+d),B	DD CB d 30	SLIA (IY+d),B	FD CB d 30
SLIA (IX+d),C	DD CB d 31	SLIA (IY+d),C	FD CB d 31
SLIA (IX+d),D	DD CB d 32	SLIA (IY+d),D	FD CB d 32
SLIA (IX+d),E	DD CB d 33	SLIA (IY+d),E	FD CB d 33
SLIA (IX+d),H	DD CB d 34	SLIA (IY+d),H	FD CB d 34
SLIA (IX+d),L	DD CB d 35	SLIA (IY+d),L	FD CB d 35
SLIA (IX+d)	DD CB d 36	SLIA (IY+d)	FD CB d 36
SLIA (IX+d),A	DD CB d 37	SLIA (IY+d),A	FD CB d 37

d = Distanzadresse (00 - FF)

RLC (xr+d),r

Name:	Op-Code	Name	Op-Code
RLC (IX+d),B	DD CB d 00	RLC (IY+d),B	FD CB d 00
RLC (IX+d),C	DD CB d 01	RLC (IY+d),C	FD CB d 01
RLC (IX+d),D	DD CB d 02	RLC (IY+d),D	FD CB d 02
RLC (IX+d),E	DD CB d 03	RLC (IY+d),E	FD CB d 03
RLC (IX+d),H	DD CB d 04	RLC (IY+d),H	FD CB d 04
RLC (IX+d),L	DD CB d 05	RLC (IY+d),L	FD CB d 05
RLC (IX+d),A	DD CB d 07	RLC (IY+d),A	FD CB d 07

Indiziert-indirekte Verschiebeoperationen mit Transfer des Endergebnisses in ein Register.

RR (xr+d),r

Name:	Op-Code	Name	Op-Code
RR (IX+d),B	DD CB d 18	RR (IY+d),B	FD CB d 18
RR (IX+d),C	DD CB d 19	RR (IY+d),C	FD CB d 19
RR (IX+d),D	DD CB d 1A	RR (IY+d),D	FD CB d 1A
RR (IX+d),E	DD CB d 1B	RR (IY+d),E	FD CB d 1B
RR (IX+d),H	DD CB d 1C	RR (IY+d),H	FD CB d 1C
RR (IX+d),L	DD CB d 1D	RR (IY+d),L	FD CB d 1D
RR (IX+d),A	DD CB d 1F	RR (IY+d),A	FD CB d 1F

RL (xr+d),r

Name:	Op-Code	Name	Op-Code
RL (IX+d),B	DD CB d 10	RL (IY+d),B	FD CB d 10
RL (IX+d),C	DD CB d 11	RL (IY+d),C	FD CB d 11
RL (IX+d),D	DD CB d 12	RL (IY+d),D	FD CB d 12
RL (IX+d),E	DD CB d 13	RL (IY+d),E	FD CB d 13
RL (IX+d),H	DD CB d 14	RL (IY+d),H	FD CB d 14
RL (IX+d),L	DD CB d 15	RL (IY+d),L	FD CB d 15
RL (IX+d),A	DD CB d 17	RL (IY+d),A	FD CB d 17

RRC (xr+d),r

Name:	Op-Code	Name	Op-Code
RRC (IX+d),B	DD CB d 08	RRC (IY+d),B	FD CB d 08
RRC (IX+d),C	DD CB d 09	RRC (IY+d),C	FD CB d 09
RRC (IX+d),D	DD CB d 0A	RRC (IY+d),D	FD CB d 0A
RRC (IX+d),E	DD CB d 0B	RRC (IY+d),E	FD CB d 0B
RRC (IX+d),H	DD CB d 0C	RRC (IY+d),H	FD CB d 0C
RRC (IX+d),L	DD CB d 0D	RRC (IY+d),L	FD CB d 0D
RRC (IX+d),A	DD CB d 0F	RRC (IY+d),A	FD CB d 0F

SRA (xr+d),r

Name:	Op-Code	Name	Op-Code
SRA (IX+d),B	DD CB d 28	SRA (IY+d),B	FD CB d 28
SRA (IX+d),C	DD CB d 29	SRA (IY+d),C	FD CB d 29
SRA (IX+d),D	DD CB d 2A	SRA (IY+d),D	FD CB d 2A
SRA (IX+d),E	DD CB d 2B	SRA (IY+d),E	FD CB d 2B
SRA (IX+d),H	DD CB d 2C	SRA (IY+d),H	FD CB d 2C
SRA (IX+d),L	DD CB d 2D	SRA (IY+d),L	FD CB d 2D
SRA (IX+d),A	DD CB d 2F	SRA (IY+d),A	FD CB d 2F

SLA (xr+d),r

Name:	Op-Code	Name	Op-Code
SLA (IX+d),B	DD CB d 20	SLA (IY+d),B	FD CB d 20
SLA (IX+d),C	DD CB d 21	SLA (IY+d),C	FD CB d 21
SLA (IX+d),D	DD CB d 22	SLA (IY+d),D	FD CB d 22
SLA (IX+d),E	DD CB d 23	SLA (IY+d),E	FD CB d 23
SLA (IX+d),H	DD CB d 24	SLA (IY+d),H	FD CB d 24
SLA (IX+d),L	DD CB d 25	SLA (IY+d),L	FD CB d 25
SLA (IX+d),A	DD CB d 27	SLA (IY+d),A	FD CB d 27

SRL (xr+d),r

Name:	Op-Code	Name	Op-Code
SRL (IX+d),B	DD CB d 38	SRL (IY+d),B	FD CB d 38
SRL (IX+d),C	DD CB d 39	SRL (IY+d),C	FD CB d 39
SRL (IX+d),D	DD CB d 3A	SRL (IY+d),D	FD CB d 3A
SRL (IX+d),E	DD CB d 3B	SRL (IY+d),E	FD CB d 3B
SRL (IX+d),H	DD CB d 3C	SRL (IY+d),H	FD CB d 3C
SRL (IX+d),L	DD CB d 3D	SRL (IY+d),L	FD CB d 3D
SRL (IX+d),A	DD CB d 3F	SRL (IY+d),A	FD CB d 3F

TAV's Z 80-Tuning

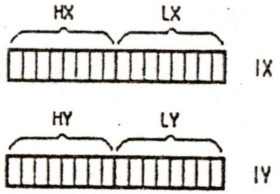
Noch mehr Illegals

In der letzten Ausgabe haben wir erfahren, was "Illegals" sind und wie durch sie die Leistungsfähigkeit der Z80-CPU gesteigert werden kann. Heute soll dieses Wissen ergänzt und neue, bisher unbekannte Befehle vorgestellt werden.

Jedem Z80-Programmierer ist sicherlich bekannt, daß er zwei 8 Bit-Register zu einem 16 Bit-Register zusammenfassen kann, wobei die darin enthaltenen 8 Bit-Register weiterhin getrennt behandelt werden können. In diese sinnvolle Registerorganisation konnten die beiden Indexregister IX und IY bisher leider nicht einbezogen werden.

Vier neue 8 Bit-Register

Durch die Illegals ist es jedoch möglich, die Indexregister aufzusplitten d.h. in jeweils zwei 8 Bit-Register zu zerlegen. Die so gewonnenen 8 Bit-Register bezeichnen wir am besten mit HX bzw. HY und LX bzw. LY.



Wie wir an diesem Schaubild erkennen können, beziehen sich die Low-Byte-Register (L) auf die relativen Bits 0 bis 7 und die High-Byte-Register auf die relativen Bits 8-15. Die acht folgenden Befehlstabellen enthalten alle Op-Codes, die Sie benötigen, um die Indexregister so effektiv wie möglich einzusetzen. Bitte beachten Sie, daß kein Assembler die vor den Op-Codes stehenden symbolischen Befehlsformen verstehen kann. Sie dienen nur zu Ihrer Orientierung.

Lade 8-Bit-Indexregister mit Konstanten

Name:	Op-Code	Name	Op-Code
LD HX,dd	DD 26	LD HY,dd	FD 26
LD LX,dd	DD 2E	LD LY,dd	FD 2E

Lade 8-Bit-Register mit 8-Bit-Indexregister

Name:	Op-Code	Name	Op-Code
LD A,HX	DD 7C	LD A,HY	FD 7C
LD A,LX	DD 7D	LD A,LY	FD 7D
LD B,HX	DD 44	LD B,HY	FD 44
LD B,LX	DD 45	LD B,LY	FD 45
LD C,HX	DD 4C	LD C,HY	FD 4C
LD C,LX	DD 4D	LD C,LY	FD 4D
LD D,HX	DD 54	LD D,HY	FD 54
LD D,LX	DD 55	LD D,LY	FD 55
LD E,HX	DD 5C	LD E,HY	FD 5C
LD E,LX	DD 5D	LD E,LY	FD 5D

Lade 8-Bit-Indexregister mit 8-Bit-Register

Name:	Op-Code	Name	Op-Code
LD HX,A	DD 67	LD HY,A	FD 67
LD LX,A	DD 6F	LD LY,A	FD 6F
LD HX,B	DD 60	LD HY,B	FD 60
LD LX,B	DD 68	LD LY,B	FD 68
LD HX,C	DD 61	LD HY,C	FD 61
LD LX,C	DD 69	LD LY,C	FD 69
LD HX,D	DD 62	LD HY,D	FD 62
LD LX,D	DD 6A	LD LY,D	FD 6A
LD HX,E	DD 63	LD HY,E	FD 63
LD LX,E	DD 6B	LD LY,E	FD 6B

Lade 8-Bit-Indexregister mit 8-Bit-Indexregister

Name:	Op-Code	Name	Op-Code
LD HX,LX	DD 65	LD HY,LY	FD 65
LD LX,HX	DD 6C	LD LY,HY	FD 6C

Additionsbefehle (8-Bit-Indexregister)

Name:	Op-Code	Name	Op-Code
INC HX	DD 24	INC HY	FD 24
INC LX	DD 2C	INC LY	FD 2C
ADD A,HX	DD 84	ADD A,HY	FD 84
ADD A,LX	DD 85	ADD A,LY	FD 85
ADC A,HX	DD 8C	ADC A,HY	FD 8C
ADC A,LX	DD 8D	ADC A,LY	FD 8D

Subtraktionsbefehle (8-Bit-Indexregister)

Name:	Op-Code	Name	Op-Code
DEC HX	DD 25	DEC HY	FD 25
DEC LX	DD 2D	DEC LY	FD 2D
SUB HX	DD 94	SUB HY	FD 94
SUB LX	DD 95	SUB LY	FD 95
SBC HX	DD 9C	SBC HY	FD 9C
SBC LX	DD 9D	SBC LY	FD 9D

Logische Operationen (8-Bit-Indexregister)

Name:	Op-Code	Name	Op-Code
AND HX	DD A4	AND HY	FD A4
AND LX	DD A5	AND LY	FD A5
XOR HX	DD AC	XOR HY	FD AC
XOR LX	DD AD	XOR LY	FD AD

Vergleichs-Operationen (8-Bit-Indexregister)

Name:	Op-Code	Name	Op-Code
CP HX	DD BC	CP HY	FD BC
CP LX	DD BD	CP LY	FD BD

Wie Sie sehen, eröffnet uns die 16 Bit High/Low Byte-Manipulation der Inexregister völlig neue Möglichkeiten. In der Praxis erspart uns gerade ihr gezielter Einsatz jede Menge Bytes und Taktzyklen, da aufwendige Shift- und Maskenoperationen in vielen Fällen durch einen einzigen Befehl ersetzt werden können. Außer der direkten High/Low-Byte Manipulation gibt es noch weitere Möglichkeiten, die Indexregister in bisher nicht bekannter Weise einzusetzen. Die zwei folgenden Tabellen beziehen sich auf eine indiziert-indirekte Einzelbitmanipulation mit anschließendem Transfer des Resultats in ein von uns definiertes 8 Bit-Register.

Eine äquivalente Sicherung des Ergebnisses haben wir übrigens im letzten Monat in Bezug auf einige neue Shift- und Rotationsbefehle kennengelernt. Die neuen Einzelbit-Befehle sind sowohl in Verbindung mit dem IX- als auch dem IY-Register anwendbar. Nachfolgend finden Sie zwei entsprechende Op-Code-Gerüste, die sich auf alle in den Tabellen vorhandenen Befehlen beziehen. In diesem Gerüst muß **XX** durch einen Tabellenwert und **of** durch eine 8 Bit-Distanzadresse (Offset) ersetzt werden.

Op-Code für IX: DD CB of XX
Op-Code für IY: FD CB of XX

Indeziert-indirektes Rücksetzen eines Bits mit anschließendem Trasfer des Ergebnisses in ein Register

Name:	A	B	C	D	E	H	L
RES 0,(XR+of)	87	80	81	82	83	84	85
RES 1,(XR+of)	8F	88	89	8A	8B	8C	8D
RES 2,(XR+of)	97	90	91	92	93	94	95
RES 3,(XR+of)	9F	98	99	9A	9B	9C	9D
RES 4,(XR+of)	A7	A0	A1	A2	A3	A4	A5
RES 5,(XR+of)	AF	A8	A9	AA	AB	AC	AD
RES 6,(XR+of)	B7	B0	B1	B2	B3	B4	B5
RES 7,(XR+of)	BF	B8	B9	BA	BB	BC	BD

Indeziert-indirektes Setzen eines Bits mit anschliessendem Trasfer des Ergebnisses in ein Register

Name:	A	B	C	D	E	H	L
SET 0,(XR+of)	C7	C0	C1	C2	C3	C4	C5
SET 1,(XR+of)	CF	C8	C9	CA	CB	CC	CD
SET 2,(XR+of)	D7	D0	D1	D2	D3	D4	D5
SET 3,(XR+of)	DF	D8	D9	DA	DB	DC	DD
SET 4,(XR+of)	E7	E0	E1	E2	E3	E4	E5
SET 5,(XR+of)	EF	E8	E9	EA	EB	EC	ED
SET 6,(XR+of)	F7	F0	F1	F2	F3	F4	F5
SET 7,(XR+of)	FF	F8	F9	FA	FB	FC	FD

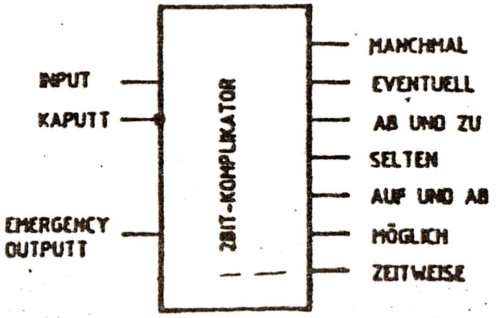
Mit dem Thema "Illegals" sind wir in ein Gebiet eingedrungen, das bis jetzt nur teilweise erforscht worden ist. Es ist also durchaus denkbar, daß die Z80-CPU weitere sinnvolle Befehle beinhaltet und Sie während Ihrer eigenen Programmentwicklungen auf weitere Illegals stoßen. Diesbezügliche Anregungen nehmen wir jederzeit dankend entgegen.

Zum Schluß noch ein kleiner Tip, der nichts mit Illegals zu tun hat: Ein 16 Bit-Register kann in Null Taktzyklen durch 256 geteilt werden! Wie? Rechnen Sie doch einfach mit dem High-Byte des verwendeten Doppelregisters weiter.

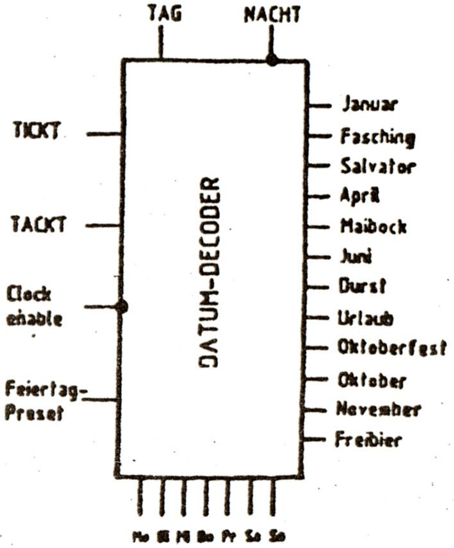
(tav)

Weitere neue IC's:
Sicher habt Ihr auch gleich die entsprechende Beschaltung dazu gefunden. Viel Spaß beim Löten und Ausprobieren.

Der 2Bit-Komplikator



Datum-Decoder



Jens Neueder

Nun folgend erscheint die schon lange angekündigte Hardware - Liste unserer Mitglieder.

Wie schon mehrmals angedeutet ist sie, wegen mangelnder Beteiligung von Eurer Seite aus, unvollständig. Der Stand dieser Datenauflistung ist der Dezember 1986. Ich möchte Euch nun herzlich bitten, änderungen, die Eure Geräte-Konfiguration betreffen, an mich weiterzuleiten. Ich hoffe, daß ich doch eines Tages einmal eine vollständige und aktuelle Liste bieten kann. Einstweilen soll und muß Euch dieses Werk genügen.

Die Liste wurde in folgende Punkte aufgeteilt:

Punkt 1 ... Computer -- hierzu sind sicher keine weiteren Erklärungen nötig.
2 ... Drucker -- Die STAR- und die Epson-Drucker (FX,MX,RX) dürften im Normalbetreib kompatibel sein. Gleiches gilt auch für die ITOH und NEC'ler. Zu den weiteren Typen kann ich keine Aussage treffen.
3 ... Laufwerke -- Die Anzahl der Laufwerke wurde in diese Aufstellung vernachlässigt. Auch wurde nur das jeweils vielseitigere Laufwerk angegeben.
4 ... CP/M -fähig
5 ... HRG -- 384 X 192 entspricht der HRG 1B, 640 X 240 läuft auf dem M4
6 ... Sonstiges -- hier werden weiter zusätzliche Geräte aufgezählt

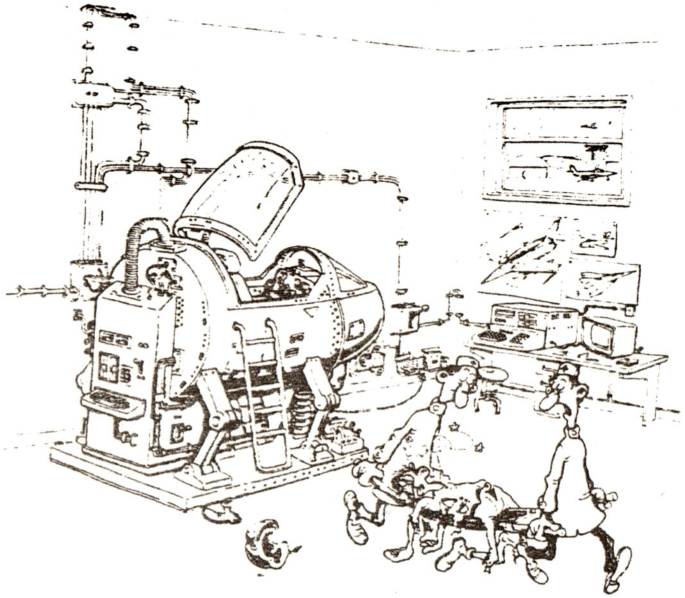
1. Computer

VG I : Herbert Albers, Helmut Bernhardt, Dieter Kasper, Holger May, Waldemar Misioch, Kurt Müller, Richard Rensch Bernd Retzlaff, Paul-Jürgen Schmitz, Walter Schäfer, Hans-Martin Stephan
VG II : Manfred Held, Rainer Jablotschkin, Waldemar Misioch, Peter Spieß, Jürgen Wucherer
VG IIs : Gerald Schröder
VG III : Richard Rensch, Helmut Bernhardt
VG IIIs : Arnulf Sopp
TRS 80 MI L2 : Ulrich Böckling, Werner Föster, Josef Konrad, Karl-Herbert Krüger, Eckehard Kuhn, Klaus-Jürgen Mühlenbein, Jens Neueder, Hans Raggan, Andreas Rychlik
Tandy M III : Helmut Emmerich
TRS 80 M4 : Rüdiger Sörensen
TRS 80 M4p : Gerald Dreyer, Klaus Hermann, Hartmut Obermann, Dieter Reichelt, Peter Stevens, Günther Wagner
ECB-Bus-TRS 80: Bernd Drowälder
Komtek 1 : Helmut Bernhardt
mc CP/M : Helmut Bernhardt
C 64 : Peter Stevens

Hardware - Liste CLUB 80

2. Drucker

STAR DP-510 : Herbert Albers, Rainer Jablotschkin, Jürgen Wucherer
STAR SG-10 : Dieter Kasper
STAR Gemini 10x : Holger May, Gerald Schröder, Walter Schäfer, Hans-Martin Stephan
FX 80 : Peter Stevens
MX 80 : Waldemar Misioch
MX 80 FT : Gerald Dreyer, Karl-Herbert Krüger
RX 80 : Werner Förster
RX 80 F/T : Klaus-Jürgen Mühlenbein, Rüdiger Sörensen
RX 80 F/T + : Hartmut Obermann
ITOH 8510 A: Helmut Bernhardt, Ulrich Böckling, Richard Rensch
NEC 8023B-C : Jens Neueder, Peter Spieß
NEC P 6 : Paul-Jürgen Schmitz, Arnulf Sopp
HEATH H14 : Kurt Müller
KX-P1091 : Bernd Drowälder
LP VI : Günther Wagner
LP VIII : Helmut Emmerich
TA DRH 80 : Manfred Held, Josef Konrad
QD DMP-1180 : Eckehard Kuhn
CP 80 : Hans Raggan
TCS Printstar 10 : Dieter Reichelt



Diese computergesteuerten Flugsimulatoren sind ein bißchen zu realistisch – besonders bei Bruchlandungen.

3. Laufwerke

Typ 5 1/4 Zoll
40 SS SD : Holger May, Hans Raggan
SS DD : Werner Förster, Bernd Retzlaff
DS SD : -
DS DD : Gerald Dreyer, Klaus-Jürgen Mühlenbein, Hans-Martin Stephan
80 SS SD : -
SS DD : -
DS SD : -
DS DD : Helmut Bernhardt, Ulrich Böckling, Bernd Drowälder, Manfred Held, Dieter Kasper, Josef Konrad, Karl-Herbert Krüger, Waldemar Misioch, Jens Neueder, Hartmut Obermann, Dieter Reichelt, Richard Rensch, Andreas Rychlik, Paul-Jürgen Schmitz, Gerald Schröder, Walter Schäfer, Arnulf Sopp, Peter Spieß, Günther Wagner

Stringy-Floppy : Jens Neueder
Typ 3 Zoll : Bernd Drowälder
Typ 3 1/2 Zoll : Jens Neueder

SA 405 : Herbert Albers
SD 521 : Herbert Albers
TEAC-55FV-13 : Rainer Jablotschkin
TEAK 55 A+B : Kurt Müller

4. CP/M -fähig

Böckling	Ulrich
Dreyer	Gerald
Drowälder	Bernd
Förster	Werner
Held	Manfred
Kasper	Dieter
Krüger	Karl-Herbert
Reichelt	Dieter
Retzlaff	Bernd
Rychlik	Andreas
Schmitz	Paul-Jürgen
Sopp	Arnulf
Stevens	Peter
Sörensen	Rüdiger
Wagner	Günther

5. Grafik

HRG - Pixelauflösung

384 × 192

Bernhardt	Helmut
Böckling	Ulrich
Drowälder	Bernd
Förster	Werner
Jablotschkin	Rainer
Krüger	Karl-Herbert
Kuhn	Eckehard
Misioch	Waldemar
Mühlenbein	Klaus-Jürgen
Müller	Kurt
Neueder	Jens
Rychlik	Andreas
Spieß	Peter

640 × 240

Dreyer	Gerald

480 × 192

Schröder	Gerald

Die neueste Version vom „Krieg der Sterne"!

6. Weiteres Zubehör

Bernhardt	Helmut	... Banker (256K RAM /ROM/RAM) I/O/RAM /A/D-Wandl	256 × 512 HRG v. Garf/Kempten
Böckling	Ulrich	... 3.4 MHz /Inverse Zeichendarstellung	Basiccode /Supertape
Dreyer	Gerald	... RS 232 /Telefax-Gerät	
Drowälder	Bernd	... Eprommer /Meßwertaufnahme /usw.	
Förster	Werner	... EM 2005 /80 Zeichenkarte	
Held	Manfred	... Grip II /Prommer 80 /Spooler /V24	
Kasper	Dieter	... 3.54 MHz	
Krüger	Karl-Herbert	... Funk mit RTTY	
Kuhn	Eckehard	... Stringy-Floppy /80-Zeichenkarte /RS 232	/Nixdorfkugelkopfdr.
May	Holger	... Joystick /Siemens-Fernschreiber T100S	
Mühlenbein	Klaus-Jürgen	... Recorder /Verstärker	
Neueder	Jens	... Stringy-Floppy /Joystick /	
Rensch	Richard	... Doubler /Expander /STAR RADIX 10	
Rychlik	Andreas	... Eprommer /Lightpen /Supertape /Basiccode	3.54 MHz /Video-Snovel /ECB-Bus /RS232
Schröder	Gerald	... 80-Zeichenkarte	
Sopp	Arnulf	... div. Schnittstellen /Banker /Prommer-80	VG I
Spieß	Peter	... Prommer /Spooler 64K /Grafikkarte 512 × 512	EG 64 /Typenradschreibmaschine
Stevens	Peter	... Olivetti ETIII Typenradschreibmaschine	
Wucherer	Jürgen	... Olivetty Typenradschreibmashine .	

Selbstbau Doubler für EXP1

Helmut Bernhardt

Nachdem sich bei mir nun schon der zweite WD1771 Single Density Floppycontroller verabschiedet hat und der Chip wohl auch bald nicht mehr zu haben ist, entschloß ich mich, mir selbst einen Doubler zu bauen, der ohne diesen Käfer auskommt. Es ist auch nicht einzusehen, warum auf einem Doubler zwei Floppycontroller sitzen müssen, wo doch der WD1791 alles das, was der WD1771 kann (nämlich beim Booten die erste Spur der Diskette in Single Density und gelegentlich auch mal ganze Disketten in SD zu lesen oder zu schreiben, wenn ein unterausgerüsteter Zeitgenosse den digitalen Informationsaustausch nur in diesem Format leisten kann) nebenbei auch erledigen könnte.

Letzteres Argument soll nun aber auch gegenstandslos werden, denn den hier beschriebenen Doubler kann sich leicht jeder selbst bauen und auch finanzieren. Grundlage ist der WD2793 Floppycontroller, der schon aufgrund seiner Bescheidenheit bei der Spannungsversorgung (ausschließlich +5V und GND) mein Wohlwollen verdient hat. Außerdem enthält er auf dem Chip schon einen PLL-Datenseparator und eine Schreibvorkompensation. Dadurch können die meisten ICs auf den Doublern von TCS und dem davon abgekupferten von RB entfallen, die nur diesen Job leisten.

Der WD2793 kann intern den an Pin 24 angelegten Takt von 2MHz (für den Betrieb von 8" Floppies) auf 1MHz (5" Floppies) runterteilen, wenn an den Pins 17 und 25 Low-Pegel angelegt wird. über Pin 37 läßt sich vorgeben, ob in Single Density (high) oder Double Density (low) gelesen und geschrieben werden soll.

Durch die 5 zusätzlichen ICs links neben dem WD2793 auf dem Doubler Board lassen sich diese Features programmieren. Zwei Flip Flops geben über ihre Q-Ausgänge die Pegel für die Pins 17, 25 und 37 vor. Nach einem RESET ist der Floppycontroller auf Minifloppy und Single Density eingestellt. Durch Ausgabe folgender Werte an das Contollregister mit der memory mapped Adresse 37ECH bzw. das Sektorregister 37EEH lassen sich die Betriebszustände folgendermaßen einstellen:

37ECH:	1111 1xx0 bin (FFH)	Single Density
	1111 1xx1 bin (FEH)	Double Density
37EEH	1000 0xx0 bin (80H)	Mini Floppy (5 1/4")
	1100 0xx0 bin (C0H)	Maxi Floppy (8")

Diese Daten haben im Contoll- bzw. Sektorregister keine Wirkung. Aus den Daten und der Schreibadresse werden nur extern die Triggersignale für die Flip Flops decodiert, wobei für die Density D0 und für die Drive-Größe D6 den einzustellenden Pegel vorgibt.

Diese Schaltung ist in Abb.1 gezeigt. Das einseitige Layout und die Anordnung der Drahtbrücken und Bauteile ist aus den folgenden Abbildungen zu ersehen. Die zur Position des WD2793 leicht versetzte Dil40-Anordnung von Lötpunkten wird von der Lötseite mit einem entsprechenden Stecker bestückt, der in den Sockel des WD1771 auf dem EXP1 gesteckt wird. Zur Zugentlastung dieses Steckers sollten an ca. 10 der durchragenden Pins auf der Bestückungsseite kurze Drahtstücke angelötet werden. Das verhindert ein eventuelles Abreißen der Leiterbahnen beim Herausziehen des Doublers aus dem 1771-Sockel.

Beim Einsatz dieses Doublers muß der invertierende Datentreiber 74LS640 neben dem 1771-Sockel durch den nichtinvertierenden pinkompatiblen Typ 74LS245 ersetzt werden, da der WD2793 keine invertierenden Daten-Ein-/Ausgänge (wie WD1771 und WD1791) hat. Die Entscheidung fiel für den WD2793, der sich nur in seinen nichtinvertierenden Dateneingängen vom WD2791 unterscheidet, wegen unüberwindlicher Beschaffungsschwierigkeiten beim WD2791. Das bedeutet aber leider auch, daß dieser Dobler bei anderen Floppycontroller-Boards nicht verwendet werden kann, da dort nicht einfach ein Treiber ausgewechselt werden kann.

Auch ein weiterer Nachteil dieses Doublers soll noch erwähnt werden. Für die einwandfreie Funktion von Datenseparator und Schreibvorkompensation sind einige Trimmungen an den Potentiometern P1 und P2 und dem Trimmkondensator C1 vorzunehmen. Dazu ist ein Oszilloskop nötig. Wenn das nicht verfügbar ist, kann im Notfall auch mit Probieren getrimmt werden. Dann werden P1, P2 und C1 auf Mittelstellung gedreht und durch wiederholte Bootversuche und ständiges Verändern der Einstellung von P2 und C1 die günstigste Einstellung für das Lesen einer Diskette gefunden. Mit P1 muß dann noch die Schreibvorkompensation geregelt werden, was durch Schreiben auf Diskette und anschließendes Lesen geprüft werden kann. Die Enstellungen sind nicht sehr kritisch, so daß auch dieses Vorgehen möglich ist.

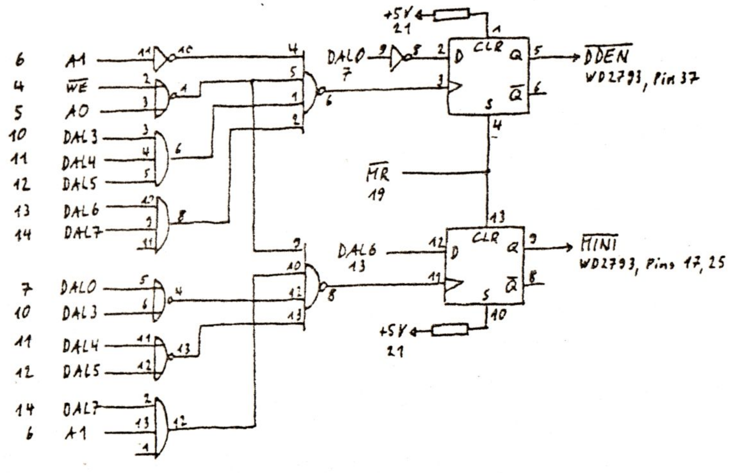
Die im Datenblatt vorgegebene Einstellung mit Hilfe eines Oszilloscops läuft folgendermaßen ab:

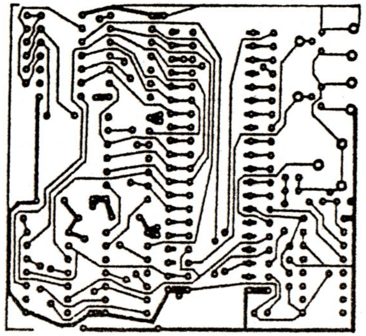
Der Computer wird mit gedrückter <BREAK> Taste eingeschaltet, um im Level II BASIC zu landen. Es wird der Befehl POKE 14316,255 (auf DD schalten) ausgeführt. Dann werden mit einem Jumper die beiden Stifte "TEST" kurzgeschlossen (Pin 21 des WD2793 an GND). Mit P1 kann dann die Breite der Pulse an Pin 31 (WD) unter Oscarkontrolle auf einen Wert zwischen 100 und 300ns eingestellt werden.

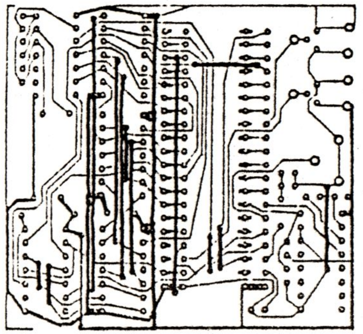
Mit P2 wird dann die Pulsbreite an Pin 29 (TG43) auf 60-65ns eingestellt. Mit C1 wird an Pin 16 (DIRC) eine Frequenz von 250kHz eingestellt.

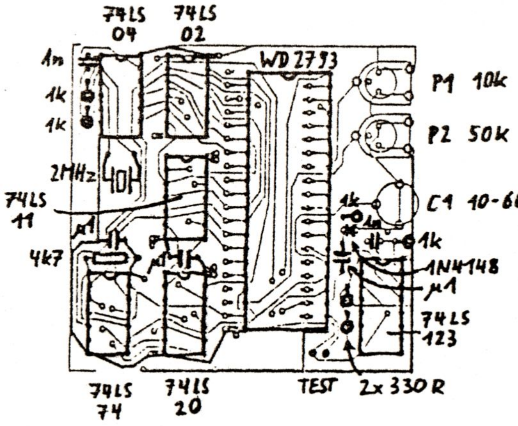
Um die Hardware-Voraussetzungen, auch 8"-Floppies betreiben zu können, zu nutzen, muß die Anordnung der Signale auf dem 34poligen Shugart-Bus für Minifloppies auf die für den 50poligen Bus für 8"-Floppies umsortiert werden. Bei älteren 8"-Drives muß diesen zusätzlich noch das Signal TG43 (WD2793, Pin29) über einen Open Collector Inverter zugeführt werden. Neuere 8"-Drives benötigen dieses Signal nicht mehr.

Wer sich diesen Doubler aufbauen will, kann für 10,-DM (incl. Rückporto) von mir ein geätztes, ungebohrtes Board bekommen und darf sich mit der Beschaffung der Bauteile, dem Bohren, Löten und Trimmen selbst abmühen.









N3: Minifloppy-Stecker

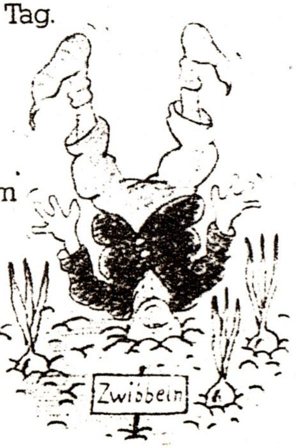
Belegung:			:	Funktion:	
			:		
01,02:	GND	/READY,/MINI	:	GND:	Signalmasse
03,04:	GND	/HOLD	:	/DS0-3:	Laufwerkauswahl
05,06:	GND	/DS3	:	/HOLD:	Kopf laden
07,08:	GND	/INDEX	:	/MOTOR ON:	Motor ein
09,10:	GND	/DS0	:	/DIR:	Spurwechsel-Richtung
11,12:	GND	/DS1	:	/STEP:	Spurwechsel-Impuls
13,14:	GND	/DS2	:	/WR DATA:	Schreibdaten
15,16:	GND	/MOTOR ON	:	/WR GATE:	Schreiben ein
17,18:	GND	/DIR	:	/SIDE:	Kopfauswahl
19,20:	GND	/STEP	:	/MINI:	Mini/Maxi-Umschaltung
21,22:	GND	/WR DATA	:		
23,24:	GND	/WR GATE	:	/READY:	Laufwerk bereit
25,26:	GND	/TRACK 0	:	/INDEX:	Indexloch-Impuls
27,28:	GND	/WR PT	:	/TRACK 0:	Kopf auf Spur 0
29,30:	GND	/RD DATA	:	/WR PT:	Schreibschutz ein
31,32:	GND	/SIDE	:	/RD DATA:	Lesedaten
33,34:	GND	/READY,/DCHG	:	/DCHG:	Diskettenwechsel

N4: Maxifloppy-Stecker

Belegung:			:	Funktion:	
			:		
01,02:	GND	/LOW CUR	:	GND:	Signalmasse
03,04:	GND	/MOTOR OFF	:		
05,06:	GND	--	:	/DS0-3:	Laufwerkauswahl
07,08:	GND	--	:	/HOLD:	Kopf laden
09,10:	GND	/TWO SIDED	:	/MOTOR OFF:	Motor aus
11,12:	GND	/DCHG	:	/DIR:	Spurwechsel-Richtung
13,14:	GND	/SIDE	:	/STEP:	Spurwechsel-Impuls
15,16:	GND	--	:	/WR DATA:	Schreibdaten
17,18:	GND	/HOLD	:	/WR GATE:	Schreiben ein
19,20:	GND	/INDEX	:	/SIDE:	Kopfauswahl
21,22:	GND	/READY	:	/LOW CUR:	Schreibstrom niedrig
23,24:	GND	--	:		
25,26:	GND	/DS0	:		
27,28:	GND	/DS1	:	/READY:	Laufwerk bereit
29,30:	GND	/DS2	:	/INDEX:	Indexloch-Impuls
31,32:	GND	/DS3	:	/TRACK 0:	Kopf auf Spur 0
33,34:	GND	/DIR	:	/WR PT:	Schreibschutz ein
35,36:	GND	/STEP	:	/RD DATA:	Lesedaten
37,38:	GND	/WR DATA	:	/DCHG:	Diskettenwechsel
39,40:	GND	/WR GATE	:	/TWO SIDED:	Zweiseitige Diskette
41,42:	GND	/TRACK 0	:		
43,44:	GND	/WR PT	:		
45,46:	GND	/RD DATA	:		
47,48:	GND	--	:		
49,50:	GND	--	:	-- = frei	

Jiddisch: A nechtige tug.
Ein nächtiger Tag.
Das wird nie passieren.

Di solst wachsen wi a zibbele mitm kop in drerd.
Du sollst wachsen wie eine Zwiebel mit dem Kopf in der Erde.
Hol dich der Teufel.



Rumänisch: La pastele cailor.
Wenn die Pferde Ostern feiern.
Wenn Ostern und Pfingsten auf einen Tag fallen.

Max Frankl, Hofolding

Spanisch: Hizo la vista gorda.
Er machte die Augen dick.
Er drückte ein Auge zu.

Echó la soga tras el calder.
Er warf den Strick hinter dem Schöpfeimer her.
Er warf die Flinte ins Korn.

Edelgard Moers, Darsten

Tandy <-> Schneider

Vielen mag die oben angedeutete Kombination zwischen Tandy- und Schneider-Computern auf den ersten Blick etwas seltsam erscheinen, aber wenn man die Sache einmal genauer besieht, bekommt sie auch Sinn.

Gemeinsames

- Sowohl der gute "alte" Tandy (Model 1 - 4) als auch der Schneider CPC 464/664/6128/Joyce arbeitet mit einer Z80-CPU.
- Die Schneider-CPC-Modelle sind, sobald sie mit einer Diskstation ausgerüstet sind, CP/M-fähig und auch die Tandys können, nach gewissen Umbauten (4/4p direkt) dieses Betriebssystem benutzen.

Diese beiden Umstände machen den Tausch von Programmen und Daten zwischen den beiden Computertypen interessant. Behindert wird die Sache dabei dadurch, daß die CPC-Modelle normalerweise mit 3"-Diskettenlaufwerken ausgerüstet sind, die bei Tandy-Besitzern (und nicht nur dort) nicht gerade weit verbreitet sind.

Möglichkeiten

Dieses Hindernis wäre z.B. dadurch zu umgehen, daß der Daten und Programmaustausch über eine RS232-Verbindung abgewickelt wird. Aber auch diese ist beiweitem nicht bei allen CPC- und Tandygeräten vorhanden!
Die zweite Möglichkeit besteht darin, ein 3"-Laufwerk an einen Tandy anzuschließen. Diese Laufwerke sind z.Z. schon für knapp 100 DM (40/SS/SD) mit shugartkompatiblem Bus zu haben. Natürlich kann man auch das original Schneiderlaufwerk anschließen, muß dabei aber einiges beachten (siehe weiter unten)!
Als letzte Möglichkeit bietet sich ein Anschluß "normaler" Drives (5 1/4 Zoll) an den Schneider an. Für diese Möglichkeit, spricht vor allem, daß man dann nicht die doch recht teuren 3-zölligen Scheibchen verwenden muß.

5 1/4" Drive mit Shugart-Standardbus am Schneider CPC 464

Billige 5 1/4"-Laufwerke bekommt man schon zu Preisen um 150,- DM (z.B. Laufwerke aus IBM-kompatiblen, die gegen ein Festplattenlaufwerk oder einen 80-Track-Drive getauscht wurden). Diese Laufwerke kann man, nach einer kleinen Modifizierung, ohne Probleme am Schneider betreiben (alle folgenden Angaben beziehen sich auf einen CPC 464 mit Controller und 3"-Drive von Schneider). Hier zunächst einmal die Belegung des Schneider-Diskinterface im Gegensatz zum Shugart-Bus:

Pin-Nr.	Shugart-Bus	CPC-Diskinterface
1	GND	/Drive Ready
2	NC (bei neueren Laufwerken /Drive 3 select)	GND
3	GND	/Backside select
4	NC	GND
5	GND	/Read Data
6	NC	GND
7	GND	/Write Protect
8	/Index Pulse	GND
9	GND	/Track zero
10	/Drive 0 select	GND
11	GND	/Write Gate
12	/Drive 1 select	GND
13	GND	/Write Data
14	/Drive 2 select	GND
15	GND	/Step
16	/Motor On	GND
17	GND	/Direction select
18	/Direction select	GND
19	GND	/Motor On
20	/Step	GND
21	GND	+ 5 Volt !!!
22	/Write Data	GND
23	GND	/Drive 1 select
24	/Write Gate	GND
25	GND	/Drive 0 select
26	/Track zero	GND
27	GND	/Index Pulse
28	/Write Protect	GND
29	GND	+ 5 Volt !!!
30	/Read Data	GND
31	GND	+ 5 Volt !!!
32	/Backside select	GND
33	GND	+ 5 Volt !!!
34	NC (bei neueren Laufwerken /Drive Ready)	GND

Erklärungen: NC = Not Connected (nicht angeschlossen)
GND = Ground (Masse)
/ = Signal ist low-aktiv (Beispiel: /Drive 0 select ist low-aktiv d.h., das Laufwerk ist angewählt, wenn das Signal auf null Volt liegt.)

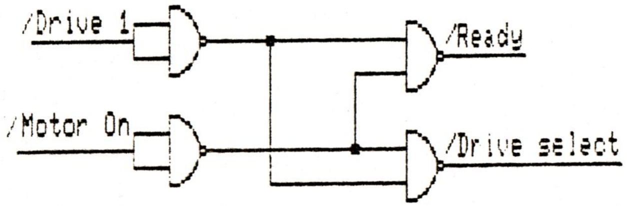
Wenn man die Signale der beiden Bussysteme einmal vergleicht, stellt man fest, daß sie beide die gleichen Signale verwenden. Was man aus dieser Tabelle nur relativ schwer ableiten kann, ist der Umstand, daß der Schneider-Floppybus nicht weiter als ein um 180 Grad gedrehter Shugart-Bus ist! Will man ein "normales" Laufwerk an diesen Bus anschließen, muß man einfach einen entsprechenden Edgecardconnektor (34-poliger Platinenstecker) zusätzlich auf das Schneider-Floppykabel aufpressen und diesen dann um 180 Grad verdreht (Pin 1 des Steckers auf Pin 34 des Laufwerks) aufstecken. So einfach ist das!

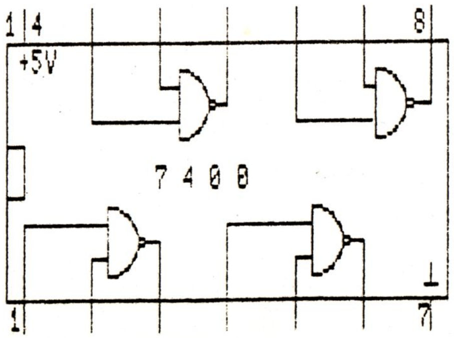
Ganz so einfach ist es aber doch nicht. Nachdem man der Floppy noch eine Stromversorgung verpaßt hat und das ganze ausprobieren will, stellt man zuerst einmal erschreckt fest, daß das zusätzliche Laufwerk dauernd selektiert zu sein scheint. Auf alle Fälle leuchtet die rote LED an der Frontblende ständig und dies, obwohl noch gar kein Floppyzugriff (weder auf 0 noch auf 1) erfolgt ist. Der Grund dafür findet sich in der etwas sparsam angelegten Schaltung des Schneidercontrollers. Das Signal /Drive 1 select wird nämlich durch eine einfache Negierung von /Drive 0 select erzeugt und damit ist Laufwerk 1 immer dann ausgewählt, wenn Drive 0 nicht selektiert ist! D.h., solange Drive 0 selektiert ist (z.B. bei DIR oder beim Laden eines Programms) erlischt die LED am Laufwerk 1 und geht sofort wieder an, nachdem die Schreib- oder Leseoperation beendet ist. Sehr schnell kommt man dahinter, daß dies nur ein unbedeutender Schönheitsfehler ist, da ja nur die LED leuchtet, der Motor sich aber nur dreht, nachdem auch das "/Motor On"-Signal gesendet wird.

Also ohne Bedenken weiterprobiert und im CP/M einfach mal "B:" eingeben, sprich Laufwerk 1 ansprechen. Denkste sagt der CPC, behauptet "Drive B not ready!" und er meint das ganz ernst! Dabei stimmt mit dem Laufwerk doch alles (Betriebsspannungen, Umdrehungsgeschwindigkeit usw.) und am Tandy läuft es einwandfrei. Woran kann es nur liegen, daß der CPC das Laufwerk auch nach viel Zureden nicht akzeptiert!? Das Problem stellt sich (wie so oft) als recht simpel heraus. Der Schneidercontroller möchte vom Laufwerk, nachdem dieses angesprochen wurde, auf dem Pin 1 (/READY) ein LOW-Signal geliefert bekommen, welches ihm sagt: "Laufwerk vorhanden und alles in Ordnung!". Genau dieses Signal liefern (abgesehen von neueren Laufwerken) die 5 1/4"-Drives nicht! Man könnte natürlich die Leitung ständig auf Masse legen und den Controller damit täuschen. Da aber das 3"-Laufwerk dieses Signal ordnungsgemäß liefert, käme diese Maßnahme einem Abbau der Betriebssicherheit gleich.

Abhilfe für beide Probleme schafft die unten gezeigte, kleine Schaltung. Die Signale /Drive 1 und /Motor On werden invertiert und dann NAND-verknüpft. Das erzeugte Signal ist immer nur dann low, wenn sowohl /Drive 1 und /Motor On aktiv (low) sind. Die NAND-Verknüpfung ist zweimal ausgeführt, um Rückwirkungen zwischen /READY- und /Drive select-Signal zu vermeiden. Das erzeugte /READY-Signal wird mit Pin 34 am Floppyanschluß verbunden; /Drive select wird dort angeschlossen, wo normalerweise der Jumper/DIP-Schalter sitzt, mit dem festgelegt wird, mit welcher Nummer das Laufwerk angesprochen wird (der Jumper muß dann natürlich entfallen bzw. der DIP-Schalter auf OFF gestellt werden!).

Mit dieser kleinen Zusatzschaltung kann man ein "normales" Laufwerk am Schneider CPC 464/664/6128 - Disk-Interface betreiben!





Nach dieser kleinen Modifikation kanns losgehen. Zunächst einmal muß man eine Diskette in Laufwerk B formatieren. Nichts einfacher als das, denkt sich der (vermeintlich) CP/M-Erfahrene und ruft "FORMAT" auf. Seltsamerweise fehlt die gewohnte Frage, in welchem Laufwerk formatiert werden soll. Also Programm abbrechen, und "FORMAT B:" eingeben; ein Parameter-Error ist der Erfolg. Im Handbuch kann man dann nachlesen, daß man mit dem mitgelieferten Formatierungsprogramm nur Disketten in Drive A in die gewünschte Form bringen kann! Schei.....!!! Zum Glück mißfiel diese Einschränkung schon einem kleveren Assemblerprogrammierer, der fix eine Erweiterung zu FORMAT verfasste. Diese braucht man nur einzutippen (etwa eineinhalb Seiten Hexdump, wer das Programm für sich oder einen Bekannten benötigt, kann es bei mir gegen Einsendung einer 3"-Disk und Rückporto bekommen) und nun kanns wirklich losgehen.

Formate

Der CPC beherrscht mehrere verschiedene Formate, darunter das CP/M 86-Format des IBM-PC. Dieses bietet zwar nur 156k freien Diskettenplatz, ist aber als Tauschformat gut geeignet, da es von vielen anderen CP/M-Computern (u.a. den Tandy's) ohne Probleme bearbeitet werden kann. Wer aber auch das eigentliche CPC-Format benutzen möchte, kann es mit folgenden Informationen leicht auf seiner Maschine implementieren:

Schneider CPC 464 Systemdiskette	
Tracks	40
Sides	1
Density	Double
Capacity	171k
Records per Track	36
Block Shift Count	3
Block Mask	7
Extent Mask	0
Disk Storage Maximum (größte Block-Nr. -1)	170
Directory Maximum (Anzahl der Einträge -1)	63
Allocation Vector 0	192
Allocation Vector 1	0
Directory Check Size	16
Track Offset (Anzahl der Systemspuren)	2
Sectors per Track (Anzahl der phys. Spuren)	9
Sector Size (Byte)	512

Reihenfolge, in der die (9) Sektoren gelesen werden (sog. Interleaving-Tabelle): 65,66,67,68,69,70,71,72,73

Für das Model 4 und Montezuma-CP/M muß mit einem Editor folgender Datensatz erstellt und in das ASCII-File DISK.FDF eingebunden werden: *Schneider CPC 464 (40T, SS, DD, 171K)
36,3,7,0,170,63,192,0,16,2,9,2,40,128
65,66,67,68,69,70,71,72,73

Danach kann man direkt und ohne weitere Probleme Daten mit dem CPC austauschen. Auch Programme lassen sich ohne weiteres transferieren, ob sie später auf der Zielmaschine auch laufen ist eine andere Frage, die ich hier nicht weiter ansprechen will.

3"-Schneider-Drive am Shugartbus

Wie wir schon gesehen haben, bereitet das Betreiben eines 5 1/4 Zoll-Laufwerkes am Schneider, von ein paar anfänglichen Schwierigkeiten abgesehen, keine weiteren Probleme. Wenn man aber kein zusätzliches Laufwerk für den CPC anschaffen möchte, kommt nur noch der Anschluß des "Schmalspurlaufwerks" an den Shugart-Bus in Frage.

Dieser bereitet noch weniger Probleme als das umgekehrte Verfahren. Man muß nur einen kleinen Adapter basteln, da das 3"-Laufwerk einen Pfostenfeldstecker als Anschluß benötigt und das direkte Aufpressen eines solchen nicht ohne weiteres möglich ist. Beim Erstellen des Adapters ist auf folgende Punkte zu achten:

1. die Leitungen 1-33 des Shugartbus können direkt mit 2-34 der 3-Zoll-Floppy verbunden werden.

2. Die Anschlüsse 21,29,31 und 33 am Floppy führen 5 Volt und dürfen nicht angeschlossen werden!

3. Das /READY-Signal (Pin 1 bzw. 34) sollte nicht angeschlossen werden.

4. Am Schneiderlaufwerk besteht nicht die Möglichkeit, durch Jumper oder DIP-Schalter die Nummer des Drives zu ändern. Aus diesem Grund muß der Anschluß 25 (Drive 0 select) des Laufwerks mit dem gewünschten Selectsignal des Shugartbus (10, 12, 14 oder 2) direkt verbunden werden. Die nicht benötigten Selectsignale sollten nicht angeschlossen werden.

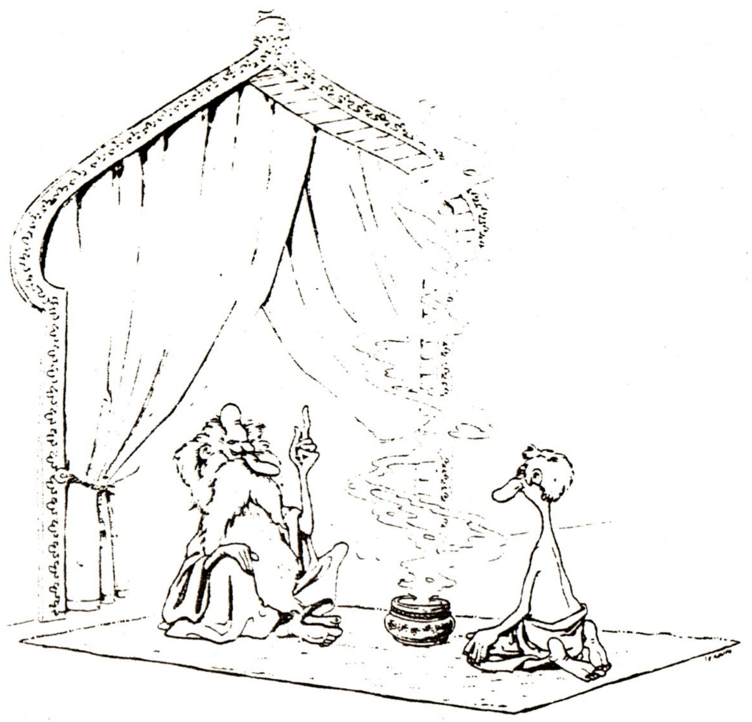
5. Die restlichen Signale können laut der oben abgedruckten Tabelle direkt verdrahtet werden.

Probleme mit dem 3"-Drive am 4p gibt es nicht. Eventuell kann es notwendig sein, im Betriebssystem die Steprate auf 12 ms heraufzusetzen, falls es öfters zu Schreib/Lesefehlern kommen sollte. Natürlich kann man das Laufwerk mit allen Betriebssystemen benutzen und auch alle möglichen Formate definieren (z.B. NewDOS 40/SS/DD). Tauschen kann man aber nur in den beiden oben schon erwähnten CP/M-Formaten.

Quellenhinweise:
De Snider un sin Disk (Detlef Grell in c't 6/85)
CP/M-Übungbuch für CPC 464/664/618 (Data Becker)
Betriebssystem CP/M (Plate / Franzis Verlag)

Sollte jemand Fragen zum Thema "Tandy <-> Schneider" haben, bin ich gerne bereit sie zu beantworten und, wo Probleme auftreten, zu helfen wo ich kann. Viel Spaß beim Löten und "schneidern"

Hartmut Obermann



Vernichte Dein Ego! Zerstöre die Illusionen der Wirklichkeit! Verbanne den Traum irdischen Vergnügens! Eine halbe Stunde mit unserem EDV-Rechnungsprogramm sollte dafür ausreichen.

Nachtrag zu "Dein G3s, die 4 unbekannten Wesen"

Wir mußten in diesem Artikel ratlos zugeben, daß die Vervierfachung des Graphikspeichers nicht geklappt hatte. *Hatte!* Sie klappt jetzt. Schuld war ein unbemerkter Lötzinntropfen, der den Pin 1 der RAMs (gemultiplexte Adressen A16 und A17) per Kurzschluß auch noch mit Refreshes beglückte.

Die Schaltung ist also vollkommen in Ordnung und kann nachgebaut werden. Man sollte aber etwas vorsichtiger löten als wir. Es lohnt sich: Vielleicht sind 8 Graphikseiten für alberne Bildchen wirklich überflüssig. Aber da es sich um gewöhnlichen RAM-Speicher handelt, ist der Zugewinn von 256 kB durchaus attraktiv. Welcher "Kompatible" kann schon mit 1 1/4 MB RAM, gleichzeitig 4 bzw. 8 Bildschirmen und 4 Zeichensätzen aufwarten?

Helmut Bernhardt
Arnulf Sopp

- I M P R E S S U M -

1. Vorsitzende — **Peter STEVENS**
Postfach 56
4600 Dortmund 1
☎ 0231 /593883

2. Vorsitzende — **Hartmut OBERMANN**
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

Hardwarekoordinator — **Eckehard KUHN**
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar — **Klaus-Jürgen MÜHLENBEIN**
Am Mönchgarten 28
6940 Weinheim -Lü.
☎ 06201 /55052

Redaktion — **Jens NEUEDER**
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren dieser Ausgabe: Helmut Bernhardt ✗ Klaus-J. Mühlenbein
Jens Neueder ✗ Hartmut Obermann
Gerald Schröder ✗ Arnulf Sopp

sowie Artikel aus: Datawelt

Bankverbindung des CLUB 80

Postgirokonto	Peter STEVENS
	Sonderkonto CLUB 80
Konto-Nummer	285 491 - 465
Postgiroamt	Dortmund
BLZ	440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

printed by S

S C H L U S S

Hallo Club-80er,

zum Abschluß des Jahres 1986 ist das 17. Clubinfo fertiggestellt.
Wie im letzten INFO versprochen, werden mit diesem INFO einige redaktionelle Aufräumarbeiten erledigt. Leider habe ich aber doch nicht alles erledigen können, wie ich es Euch angekündigt hatte. Zu guter Letzt gab auch noch der Kopierer, den ich zum Verkleinern verwende, den Geist auf. Es wurde also Zeit aufzuhören für dieses Jahr.

Die Publikation unseres Jahresinhaltsverzeichnises hat sich zu einem umfangreichen Werk gemausert. Aus diesem Grund möchte ich es als ein Sonderheft herausbringen. Dieses Sonderheft wird, weil es mir zeitlich nicht früher möglich ist, nun erst mit dem 18. INFO erscheinen. Ich hoffe, Ihr habt dafür Verständnis.

Beiliegend zu dem 17. INFO erhaltet Ihr den neuesten Programm-Katalog, einen Club-Kalender mit unseren wichtigsten Terminen und das Anmeldeformular für unser Clubtreffen im März. Bitte nehmt den Anmeldetermin rechtzeitig war. Am 20. Februar ist Anmeldeschluß. Mit den anderen beiden Beilagen wünsche ich Euch viel Spaß und erfolgreiches Schaffen.

Wieder ist ein Jahr Clubinfo's vorüber. Es ist das letzte Heft mit dem orangefarbenem Deckblatt. Ich hoffe, Ihr seid mit meiner Arbeit für das Jahr 1986 zufrieden gewesen. Gleichzeitig wünsche ich Euch, daß Ihr Weihnachten und den Jahreswechsel gut überstanden habt.
Im neuen Jahr geht es mit frischem Mut und neuer Deckblattfarbe weiter. Ich würde mich freuen, wenn Ihr mich im "Neuen Jahr" weiterhin mit Euren Beiträgen unterstützt. Bis dahin alles Gute.

Euer Jens

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907	// 0431 /74047
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621	// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441	Wettringen	05233 /4320	// 02557 /1236
Emmrich	Helmut	Waldstraße 5	6682	Ottweiler	06824 /4114	// -
Frink	Thomas	An der Schleifmühle 2	5042	Erftstadt	02235 /76255	//
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721 /21841	// 09721 /51256
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573	//
Hartel	Eberhard	Neenstetter Straße 20	7901	Breitingen	07340 /7281	//
Heile	Heinz-Dieter	Blankensteiner Straße 13	4320	Hattingen	02324 /28458	//
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024	//
Hill	Peter	Eckstraße 36	6750	Kaiserslautern 31	0631 /54782	//
Hummel	Anton	Schubertstr. 2	7612	Haslach	07832 /8289	//
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948 /542	// 02921 /70431
Kasper	Dieter	Zeppelinstr. 9	8952	Marktoberdorf	08342 /1630	// 089 /522071
Konrad	Josef	Anzengruberstraße 35	8038	Gröbenzell	08142 /8494	// -
Kopschina	Peter	Strandallee 138	2409	Scharbeutz	-	// -
Krüger	Karl-Herbert	Bruchweg 65	4920	Lemgo	05261 /13686	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417	// 07022 /77442
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /668151
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083	// 04103 /702662
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913	// -
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /3791	//
Reichelt	Dieter	Philipp-Schmitt-Straße 30	6902	Sandhausen	06224 /52906	// -
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551	// 04103 /7025310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383	// 0203 /331383
Schmitz	Paul-Jürgen	Lübecker Straße 6	6236	Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221 /707044	//
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631	// 08025 /41247
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284	//
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434 /454	// 08431 /7041684
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585	// 05401 /90037
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883	// 0231 /593883
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131 /32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285	//
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361	// -
Weiß	Dieter	Bürglestraße 3	7209	Wehingen	07426 /7194	//
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686	// -
Zwickel	Walter	Lengfelden 123	A- 5101	Bergheim	0043662/51130	// -

Stand: Dezember 1986

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

CLUB80 TERMIN - KALENDER 1986

JANUAR

Mo	Di	Mi	Do	Fr	Sa	So
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

FEBRUAR

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	◆	

MÄRZ

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	‡	‡	‡
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

APRIL

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
◆	28	29	30			

MAI

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

JUNI

Mo	Di	Mi	Do	Fr	Sa	So
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	◆					

◆ --) Redaktionsschluß des jeweiligen INFO's

‡ --) 3. Clubtreffen in Alsfeld (siehe INFO 16/17)

Messe-Daten sind unterstrichen. Bitte schaut dazu in die Terminliste.

Bitte denkt an die Termine und nehmt sie rechtzeitig war !!!

JULI

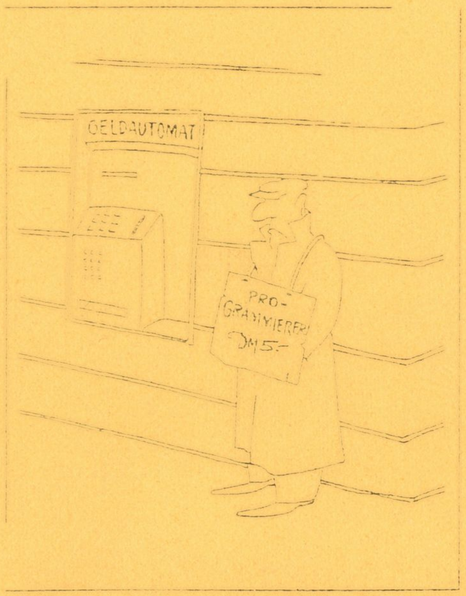
Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

AUGUST

Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
◆	25	26	27	28	29	30
31						

SEPTEMBER

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				



OKTOBER

Mo	Di	Mi	Do	Fr	Sa	So
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
◆	27	28	29	30	31	

NOVEMBER

Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

DEZEMBER

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
◆ 29	30	31				

CLUB 80

Clubinfo der TANDY - GENIE - und KOMTEK - ANWENDER

Tarantula auf dem Mikrochip? Weit gefehlt: Größenvergleich eines Halbleiterlasers mit einer Waldameise, aufgenommen unter dem Raster-Elektronen-Mikroskop: Auf dem quaderförmigen Plättchen (links unten) ist ein Draht von 25tausendsel Millimeter – dünner als ein Menschenhaar – als Stromzuführung gelötet. Eine Milliarde Lichtimpulse pro Sekunde – das entspricht einigen Dutzend TV-Programmen oder 15 000 Telefon-Gesprächen – können so in die Glasfaser eingespeist werden.

18. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
TEL.: 06124 / 3913

INHALTSVERZEICHNIS

Seite:
und Autor:

Clubinternes

Neues vom Vorstand .... 1 - 2
Hartmut Obermann
Liebe Clubfreunde 3
Helmut Obscherningkat
Termine / Messen 4
Redaktion

Software

Das CARTESIsche Blatt .... 5 - 10
Klaus-Jürgen Mühlenbein
40 Tracks? Warum nicht 42? 11
Helmut Obscherningkat
Die Druckerabfrage 12
Paul-Jürgen Schmitz
Keine Angst vor Proportionalschrift 13 - 14
Neues zur Proportionalschrift ...... 15 - 17
Lprint mit modernen Genies .... 19 - 20
Arnulf Sopp
Pagelist/CMD .... 21 - 26
Artikel aus ...
Filter unter TRSDOS 6.x .... 29 - 35
Rüdiger Sörensen
Zeus mit 64 und 80 Zeichen .... 36 - 41
Schöner listen mit ZEUS .... 41 - 43
Arnulf Sopp
CP/Mac - Eine Brücke zwischen ... 44 - 46
Richard Rensch
Trau - schau, wem! .... 47 - 48
Klaus-Jürgen Mühlenbein

Seite:
und Autor:

Hardware

Richtigstellung: 256k-Banker 48
Helmut Bernhardt
Floppydisk-Controller-Karte .... 49 - 56
Bernd Drowälder
Videodat-System .... 57 - 60
....
Das Projekt (Fernseher -> Monitor) 61 - 66
Kurt Müller
512K-RAM-Adress. m. flexibler MMU 67 - 72
Helmut Bernhardt

Börse

Wer hat was -- wer will was .... 73 - 74

Sonstiges

Erster Megabit-Chip in Serie 18
Die richtige Dimension 26
Artikel aus PM
Trace oder die Geschichte einer ... 75
Helmut Obscherningkat
Tandy schließt Computer-Center 75
"Mailbox" ist der Renner 75
Bei Siemens heißt es Chip, Chip, Hurra! 76
aus verschiedenen Zeitschriften

Programmbibliothek

Neue Eingänge in der Diskettothek .. 77 - 79
Klaus-Jürgen Mühlenbein

Die letzten Seiten

Impressum 81
Schluß 82
Clubmitgliederadressen .... am INFO-Ende
Gesamt-Inhalts-Verzeichnis .... am INFO-Ende
Redaktion

Neues vom Vorstand

Liebe Mitglieder,

diesmal gibt es eigentlich recht viel zu berichten, trotzdem möchte ich mich kurz fassen um euere Aufmerksamkeit nicht allzu sehr zu beanspruchen.

Clubtreffen '87

Zuerst einmal zum Thema Clubtreffen. Inzwischen sind schon recht viele Anmeldungen bei mir eingegangen und ich kann euch versprechen, daß es wieder eine sehr interessante Veranstaltung werden wird. Sollte sich jemand noch kurzfristig dazu entscheiden zum Clubtreffen zu kommen, bitte ich ihn sich persönlich um seine Unterbringung zu kümmern. Ich werde am Sonntag den 22.02. persönlich die bis dahin eingegangenen Anmeldungen in feste Zimmerreservierungen umsetzen. Nach diesem Datum eintreffende Anmeldungen werden von mir nur noch registriert, für die Unterbringung hat der "Spätzünder" dann selbst zu sorgen!

Unter den Teilnehmern am Clubtreffen haben sich auch drei Freiwillige gefunden, die zu folgenden Themen Vorträge halten werden:

Arnulf Sopp:	Unterschiede und Vorteile des Assemblers ZEUS gegenüber EDTASM!
Helmut Bernhard:	Hardwareunterschiede und Vorteile des HD 64180 gegenüber der "alten" Z80-CPU und deren Ersatz durch diesen Superchip!
Gerald Schröder:	Softwareprobleme, welche beim Einsatz der CPU HD 64180 auftreten können und deren Lösung!

Hier die Tagesordnung des Treffens:

Fr.13.03.87:	Tag zur freien Verfügung, kein offzielles Programm	
Sa.14.02.87:	13:00-16:00	Vorträge (genauer Zeitansatz noch nicht bekannt)
	16:00-17:00	Kaffeepause
	17:00-18:30	Vorträge
	18:30-19:30	Abendessen
	19:30-21:00	Mitgliederversammlung (Kassenprüfung, Entlastung des alten Vorstands, Wahl eines neuen Vorstands, Diskussion über weitere Vorhaben des Clubs und sonstige Themen)
So.15.02.87:	keine offiziellen Punkte mehr geplant, ev. noch Zeit für weitere Vorträge	

Wie ihr seht, ist der offizielle Teil relativ kurz gehalten. Da diesmal keine neue Satzung besprochen und verabschiedet werden muß, dürfte die Zeit trotzdem durchaus ausreichen. Zudem bildet die Tagesordung nur einen sehr flexiblen Rahmen, in dem die einzelnen Tagesordnungspunkte durchaus verschoben oder im Zeitansatz verändert werden können!

Hier noch einmal Adresse und Telefonnummer des Tagungshotels:

Hotel Klingelhöffer
Hersfelder Straße 47/48
6320 Alsfeld (Hessen)
Tel.: 06631 / 2073.

Fusion

Keine Angst! Der Club 80 steigt nicht etwa in die Diskussion um die Kernenergie ein, sondern es geht um die "Vereinigung" (noch so ein zweideutiges Wort) zweier Computerclubs.

Sicher kennen einige den Computer-Club Bremerhaven der sich, genau wie wir, mit den Tandy-Computern und ihren Nachfolgern befasst. Einige von uns (u.a. meine Wenigkeit) sind sogar Mitglied im BCC. In letzter Zeit geht es mit den BCC'lern leider stark bergab. Ihr Info bestand im letzten halben Jahr praktisch nur noch aus Beiträgen von Arnulf Sopp und Helmut Bernhard und wird in nächster Zeit wohl noch dünner werden, da Arnulf seinen Austritt erklärt hat. Der Betreuer des BCC, Ralf Folkerts hat nun im letzten Clubinfo an seine Mitglieder die Frage gestellt, was sie zu einem Vorschlag von Helmut Bernhard sagen, mehrere Clubs zusammenzuschließen.

Diese Frage, nämlich ob es sinnvoll und wünschenswert ist mit dem BCC eine Vernunftehe einzugehen, stelle ich hiermit auch euch, den Club 80-Mitgliedern und hoffe auf Reaktionen. Natürlich steht dieses Thema auch auf der Tagesordnung des Treffens (als weitere Vorhaben) und sollte ausgiebig diskutiert werden!

Pflichtbeitrag

Noch ein kontroverses Thema, welches ich euch ans Herz legen möchte. Auch unserem Club kann irgendwann der Tot durch Passivität drohen (wenn er nicht zuvor Mangels Vorstand stirbt!). Eine mögliche Lösung dieses Problems könnte die Einführung eines Pflichtbeitrages sein. Unter Pflichtbeitrag verstehe ich, daß nur Mitglied werden oder bleiben kann, der mindestens einmal im Jahr einen Beitrag (und sei er auch noch so klein) zum Clubinfo leistet. Es wäre auch die Möglichkeit gegeben, für Mitglieder, die keinen Beitrag zum Info leisten den Mitgliedsbeitrag herauf-, bzw. für Infoschreiber herabzusetzen.

All dies soll nur als Anregung und Denkanstoß dienen und eine Diskussion in Gang setzen. Ich hoffe, ihr greift diese Gedanken auf und sie werden zu einem Thema während des Clubtreffens!

Bibliothek

Der Club 80 hat, ich bezweifle ob das alle Mitglieder wissen, eine recht umfangreiche und ansehnliche Büchersammlung aufgebaut. Leider wird diese Bibliothek nur sehr wenig genutzt. Hiermit möchte ich alle Mitglieder aufrufen, diese Einrichtung intensiver zu gebrauchen und sich ein Buch auszuleihen, statt es zu kaufen. Natürlich haben wir nicht alle Bücher nach denen ein Computerfreund fragen kann, aber die wichtigsten Werke gerade zum TRS 80 besitzen wir schon!

Leider befinden sich immer noch einige Bücher im Umlauf, die sich beim letzten Clubtreffen 1986 in Holzhausen Mitglieder ausgeliehen haben (teilweise ohne sich in die ausgelegte Entleiherliste einzutragen). Ich bitte alle Mitglieder, die sich auf dem Treffen Bücher ausgeliehen haben festzustellen, ob sie diese schon zurückgegeben haben. Sollte dies nicht der Fall sein, bitte ich um schnellstmögliche Rücksendung!

So, genug für diesmal! Ich wünsche euch allen einige angeregte Stunden bei der Lektüre des Clubinfos und viel Spaß beim Computern, euer

Hartmut Obermann

Liebe Clubfreunde,

Nachdem ich nun gleich zu Anfang mit einem grossen Haufen Papier, in Form von allen zurückliegenden Info's empfangen habe, darf ich mich sicher als Mitglied betrachten.
Um auch den scheinbar üblichen Gepflogenheiten zu entsprechen, will ich mich im folgenden der übrigen Familie vorstellen.
Nachdem ich die Mitgliederliste gründlich studiert habe, kann ich zu meiner Freude feststellen, dass mit meinem Beitritt der Club80 "europäisch" geworden ist: Ein Schweizer, ein Oesterreicher und nun ein 1/2 Franzose.
Das 1/2 kommt daher, dass ich ein gebürtiger Berliner bin, aber nach dem Krieg in Frankreich hängen geblieben bin und hier meine Existenz aufgebaut habe. Daraus könnt Ihr auch errechnen, dass ich nicht mehr ganz taufrisch bin; ich glaube sogar, dass ich der Senior des Vereins sein werde. Ich bin am 28.Mai 1922 geboren, wer bietet mehr ?
Nun zu meinem Vorleben: In meiner aktiven Zeit habe ich mich als Ingenieur mit der Konstruktion von Werkzeugmaschinen befasst, natürlich seit etwa 1975 mit numerischer Steuerung. Ursprünglich war ich reiner Maschinenbauer aber mein Hobby, die Elektronik, hat mir die Anpassung an die neuen Technologien verhältnismässig leicht gemacht.
Wie jeder weiss, hat die Technik seither Riesensprünge gemacht hat: von der Röhrentechnik über Transistoren und den integierten Schaltkreisen bis zum Mikroprozessor. Aber diese Zeit war wahnsinnig interessant und hat an die Lernfähigkeit schon einige Anforderungen gestellt.
Was bleibt mir heute ? Zuerst war es ein Junior-Computer von ELEKTOR, dann kam gleich ein Tandy TRS 80 mod.3 ins Haus, der dann Weihnachten 1985 durch das mod.4 ersetzt wurde, ausgerüstet mit 2 Drive 40 ss dd und der integrierten RS 232 Schnittstelle. Dann kam die HGR-Grafikkarte mit 640x240 Punkten von Graphyx Solution. Vorher hatte ich mir aber schon einen A3 Flachbrett-Plotter gebaut. Die Intelligenz mit eingenem Z 80 stammt von Winter, Stuttgart. Dazu gesellt sich dann noch ein Typenraddrucker DM2 (aus heutiger Sicht eine Dummheit !).
Auf jeden Fall kann man damit keine Langeweile bekommen. Im Wesentlichen beschäftige ich mit Basic-Programmen, vor allen Dingen mit Grafik-Anwendungen (deshalb HGR und der Plotter). Ich möchte gerne endlich meine Kenntnisse in Assembler vertiefen, aber EDTASM von Tandy läuft nur noch im III Mode und ist dann nur noch eine halbe Sache. Dazu möchte ich mein mod.4 recht bald zum Empfangen von RTTY benutzen, dazu fehlt mir noch das Filter-programm Baudot --> ASCII.
Ach ja da wäre ja noch meine Software. Also erst einmal das TRSDOS 6.1 und 6.2, dann superSCRIPSIT (franz.), HYPERCROSS zum lesen von CP/M und IBM PC-Files (nur interressant mit 80-traks-Laufwerk), HGR Tournament chess (sehr stark). Dann für mod 3 TRSDOS 1.3, VISICALC, PROFILE, EDTASM, und sonst noch einiges.
Natürlich besitze ich einen Haufen von französischen Zeitschriften und auch Büchern, wovon ich selbstverständlich auch einiges in die Programmbibliothek einbringen könnte. Aber da bleibt der Haken mit der Uebersetzung, welche bekanntlich sehr zeitaufwendig sein kann. Aber darüber lässt sich eventuell noch reden.
Das wäre vorerst das Wichtigste. Bei allen, welche bis hierher gelesen haben, bedanke ich mich für ihre Aufmerksamkeit.

Helmut Obscherningkat

-- Termine -- Termine -- Termine --

Jahreshauptversammlung 1987 ....13. - 15. März 1987

Große TRS-80 Clubtage ....11. April 1987
Hauptversammlung bei dem holländischen Club
(Auskunft dazu bei G. Dreyer)

Nächster Redaktionsschluß ....27. April 1987

Norddeutsches Regionaltreffen ....25. + 26. April 1987
Achtung:
Wer Interesse hat melde sich bis zum 5. April beim Hans-Martin an !!!

Micro-Treff Ludwigshafen ....23. + 24. Mai 1987
Willi-Graf-Haus (am Ruthenplatz,Leuschnerstraße 151)
Sa 10-18Uhr, So 10-16Uhr

-- Messen '87 --

CeBIT	Hannover	4. - 11. März 1987
Internationale Computer-Ausstellung ..	Köln	11. - 14. Juni 1987
HOBBY ELEKTRONIK 87	Stuttgart	5. - 8. November 1987
PRODUCTRONICA	München	10. - 14. November 1987

Das CARTESIsche Blatt
Eine algebraische Kurve dritter Ordnung

(K.-J.Mühlenbein, Weinheim)

Unter einer algebraischen Kurve versteht man die Menge aller Punkte (x,y), deren Koordinaten einer algebraischen Gleichung der Form

$$\sum_{\nu,\mu=0}^{n} a_{\nu\mu}\, x^{\nu} y^{\mu} = 0$$

genügen. Man bildet also die Summe aller Produkte der mit Exponenten von 0 bis n versehenen Koordinaten x und y, nachdem diese Produkte mit vorgegebenen Koeffizienten $a_{\nu\mu}$ multipliziert wurden.

Ein Beispiel möge diesen Satz verdeutlichen.
Gegeben sei folgendes Koeffizientenschema der $a_{\nu\mu}$ für n=3:

Koeffizienten $a_{\nu\mu}$

ν	μ=0	1	2	3
0	1	0	-2	1
1	0	-3	0	3
2	2	0	-2	1
3	1	1	-4	2

Es ist also z.B.:
$a_{00} = 1$, $a_{01} = 0$, $a_{11} = -3$, $a_{31} = 1$.

Dann sieht die Summe wie folgt aus:

$$\begin{aligned} &1 &+ 0 &- 2y^2 &+ y^3 \\ &+ 0 &- 3y\cdot x + 0 & &+ 3y^3 x \\ &+ 2x^2 &+ 0 &- 2y^2 x^2 &+ y^3 x^2 \\ &+ x^3 &+ y &- 4y^2 x^3 &+ 2y^3 x^3 = 0 \end{aligned}$$

Die Menge aller Punkte der x,y-Ebene, deren Koordinaten diese Gleichung erfüllen, bildet eine algebraische Kurve.

Das Beispiel wird übersichtlicher, wenn man alle Koeffizienten streicht, für die die Summe "Spalten- + Zeilenindex" größer als 3 ist. Es verbleibt:

ν	μ=0	1	2	3
0	1	0	-2	1
1	0	-3	0	
2	2	0		
3	1			

Die Gleichung lautet dann:

$$1 + 2(x^2 - y^2) - 3xy + x^3 + y^3 = 0$$

Eine solche Kurve, für die die Summe der Exponenten in jedem Glied höchstens gleich 3 ist, nennt man "algebraische Kurve dritter Ordnung" oder kurz "Kubik".

Wie sieht sie aus?
Sicher ist sie symmetrisch (da x und y in dieser Gleichung offensichtlich vertauschbar sind) und somit schön! Ihre grafische Darstellung geschieht am besten auf dem Umweg über einen Parameter, von dem sowohl x als auch y abhängt, da die Auflösung nach y zu einer komplizierten irrationalen (weil Wurzeln enthaltenden)

Form führen würde.

Um es für diese Demonstration grafischer Programmierung noch einfacher zu gestalten, soll nur die rechte Hälfte der letzten Gleichung verwendet werden:

$$x^3 + y^3 - 3xy = 0$$

Dies ist natürlich auch eine Kubik. Sie stellt das sog. *"CARTESIsche Blatt"* dar (s.Fig.1).

Mit den Substitutionen p=-3x und $q=x^3$ geht die Gleichung in die sog. "reduzierte" Form einer kubischen Gleichung über:

$$y^3 + py + q = 0$$

Unter der Voraussetzung, daß das Kriterium

$$K=(q/2)^2 + (p/3)^3$$

nicht negativ ist, kann diese Gleichung mithilfe der *CARDANIschen* Formel (die gar nicht von *Cardano*, sondern von seinem Zeitgenossen *Tartaglia* stammt!) gelöst werden. Man sieht, daß diese Voraussetzung nur für negative x erfüllt ist, denn nur dann ist sowohl q^2 als auch p^3, also auch K positiv.

Für positive x ist die Voraussetzung K>=0 erst ab x>=1.5874 erfüllt, denn bei diesem Wert wird K=0 und geht von negativen zu positiven Werten über:

$$K=0 \longrightarrow (q/2)^2 = -(p/3)^3$$
$$x^6/4 = x^3$$
$$x=\sqrt[3]{4}=+1.5874$$

Das Gebiet 0...x...1.5874 muß also anders behandelt werden. Hier liegt der sog. "casus irreducibilis" vor, den die Mathematiker bis zum Ende des 16.Jahrhunderts nicht lösen konnten, bis *VIETA* um 1600 – wie einst *Columbus* ein Jahrhundert vor ihm – das Problem "mit einem Schlag" löste, indem er "einfach" zu Polarkoordinaten in der komplexen Ebene überging; und siehe da: Es erwies sich, daß alle drei Wurzeln in diesem "nicht-reduzierbaren Fall" besonders anschaulich (nämlich reell) sind, im Gegensatz zum *"Cardanischen Fall"*, in dem oft zwei imaginäre Wurzeln auftreten! – Der rechnerische Ablauf soll hier nicht nachvollzogen, sondern gleich die Parameterdarstellung angeführt werden, die sich für x bzw. y als Funktion des Winkels φ ergibt:

$$x = \sqrt[3]{4\cos^2\varphi}$$
$$y = 2\sqrt{x}\,\cos(\varphi/3)$$

Das praktische Vorgehen besteht darin, cosφ für viele, möglichst dicht beieinander liegende feste x-Werte zu berechnen, um hieraus für jedes x die jeweils drei Werte von y zu gewinnen, die sich aus der Periodizität des Kosinus ergeben – drei Werte deshalb, weil der Kosinus für die Berechnung von y von φ/3 zu bilden ist; y hat daher für jedes x zwischen 0 und 1.5874 drei verschiedene Lösungen, nämlich für drei Winkelwerte, die sich um je 120° (entsprechend 2π/3) unterscheiden.

<u>Listing 1:</u> Zunächst wird die ab -2 stetig um 0.01 wachsende Laufvariable N fortlaufend der Variablen X zugeteilt. Dann wird y für diesen negativen x-Bereich berechnet (Zeilen 250-270). Danach erfolgt ab Zeile 90 die Zuteilung der positiven Werte an das X-Feld und die Berechnung von C=cosφ (Zeile 220), und zwar für 159 x-Werte. In Zeile 230 werden über die Weiche W% (=2, 0, 1) die drei Werte für den Parameter φ=arccosC berechnet (der

arccos muß in BASIC bekanntlich durch den arctan dargestellt werden), die den von φ – und damit indirekt von X – abhängigen Verlauf der drei Kurvenäste bestimmen. Den Werten 2 bzw. 0 bzw. 1 der Weiche W entspricht der mittlere bzw. obere bzw. untere Ast der Kurvenschleife. In dieser Reihenfolge wird das "CARTESIsche Blatt" (das man auch als "Cartesische Schleife" bezeichnen könnte) beim Programmlauf auf dem Bildschirm aufgezeichnet, also in stetiger, "eleganter" Weise.

Für $x > \sqrt[3]{4}$ ist diese algebraische Funktion übrigens wieder eindeutig. Auf den positiven Halbachsen sind die Grenzen des drei-deutigen Bereichs bei 1.59 markiert. Die Kurve spiegelt sich an der x,y-Diagonalen.

Die Berechnung von Y(1) bis Y(678) für 678 Kurvenpunkte erfordert in BASIC etwas Zeit und je nach Prozessortakt also etwas Geduld. (Bei einer Taktfrequenz von 1.75 MHz ca. 4 Minuten.) Zur Information über den momentanen Stand der Berechnung läuft deshalb oben rechts im Bildschirm der Zähler I mit. Wenn dieser den Wert 678 erreicht hat, wird der Grafiktreiber geladen und das CARTESIsche Blatt gezeichnet. Es ist recht hübsch, dem imaginären Zeichenstift beim schwungvollen Zeichnen auf der tabula rasa des Bildschirms zuzusehen!

Wenn man das CARTESIsche Blatt in horizontaler Stellung mit der y-Achse als Symmetrieachse zeichnen möchte, muß das Koordinatensystem um 45° nach rechts gedreht werden. Wegen $\sin(-45°) = -\cos(-45°) = -\frac{1}{2}\sqrt{2}$ entsteht aus obiger Gleichung nach der Transformation

$$x = (x'-y')/\sqrt{2}$$
$$y = (x'+y')/\sqrt{2}$$

die Gleichung:

$$y^3 - 3y^2 + yx^2 + 3x^2 = 0,$$

die sich leicht nach x auflösen läßt:

$$x = y\sqrt{(3-y)/(3+y)},$$

worin x und y jetzt die transformierten Koordinaten bedeuten.

Hier kann nun der Umweg über einen Parameter entfallen. Für die grafische Programmierung spielt es keine Rolle, ob y als Funktion von x oder x als Funktion von y programmiert wird. Das HRG-Programm verlangt lediglich, daß definierte Wertepaare (x,y) in kontinuierlicher Reihenfolge vorliegen – zweckmäßigerweise also in Gestalt zweier eindimensionaler Felder mit gleichem Index. *Das Ergebnis zeigt Fig. 2.*

Listing 2, Zeile 90: Das Feld Y wird mithilfe von Z in Schrittweiten um 0.05 von -2 bis +3 definiert. Das Feld der dazu gehörenden X-Werte wird in Zeile 100 berechnet. Da diese algebraische Kurve symmetrisch zur y-Achse ist, wurde X nur einmal berechnet, und zwar für die links unten beginnende, bis zum Scheitelpunkt reichende Hälfte der Kurve. Sodann wird in Zeile 150 das Vorzeichen aller X-Werte umgekehrt und dieses Feld dem gleichen Y-Feld zugeteilt. Diese Kurve wird daher nicht wie die erste in einem Zug, sondern in zwei Durchgängen gezeichnet. Durch diesen Trick wurde erheblich an Rechenzeit eingespart.

Alles weitere sind Grafikbefehle, die – wie schon in Listing 1 – der Syntax des hier verwendeten HRG-Plotsystems entsprechen. Wird ein anderes HRG-System verwendet, so brauchen nur diese Befehle durch dessen Syntax ersetzt und an seine Besonderheiten angepaßt zu werden.

Zur Syntax des hier verwendeten Plotsystems:

Für die beiden Darstellungen wurde das Plotsystem *"GRAPE"* von *M.WINTER* mit dem EPSON-Treiber "MXBORB/DRV" verwendet. Diese Befehle sind durch ein vorangestelltes Minuszeichen gekennzeichnet. Es bedeuten:

-LOADRV("MXBORB ")	Aufruf des Treibers (hier für EPSON MX80 und RX80)
-DEFPIC(...)	Größe der Bildfläche innerh. der gesamten Zeichenfläche
-WINDOW(...)	Minimum- und Maximumwerte für x und y
-AXIS(...)	Festlegung der Achsen bzw.Gitter einschl. Skalierung.
-MOVETO(X,Y)	Definition der Anfangskoordinaten f.d. "Zeichenstift".
-MOVEON(X,Y)	zeichnet Verbindung vom Anfangspunkt zum Punkt X,Y
-POLYGO(X,Y,N)	verbindet N Punkte, die durch die Felder X(I), Y(I) definiert sind, in der Reihenfolge, die durch den Feldindex I vorgegeben ist
-MARKPT(...)	markiert Punkte, die ebenfalls durch X/Y-Felder definiert sind.
-TEXT(...)	zeichnet Beschriftungen
-NWRITE(...)	zeichnet numerische Beschriftungen

12 Monate: Das Kind entdeckt, wieviel Leben in die Bude kommt, wenn man mitten in der „Schwarzwaldklinik" den Stecker des Fernsehers rauszieht.

18 Monate: Das Kind lernt: Die Fernbedienung zu verstecken macht noch viel mehr Freude als die alte Geschichte mit dem Stecker.

24 Monate: Das Kind ruft die Eltern mit dem Lockruf *„Oooooo — ooooo — iiiii!"* zum Fernsehgerät, wobei man allerdings nicht weiß, ob es „Otto ist hier", „Nowottny" oder „Lottoziehung" heißen soll.

30 Monate: Nun beginnt das Kind, sich eine eigene Meinung zu bilden und spricht zur Freude der Eltern das Wort *„Mistprogramm"* fehlerfrei aus.

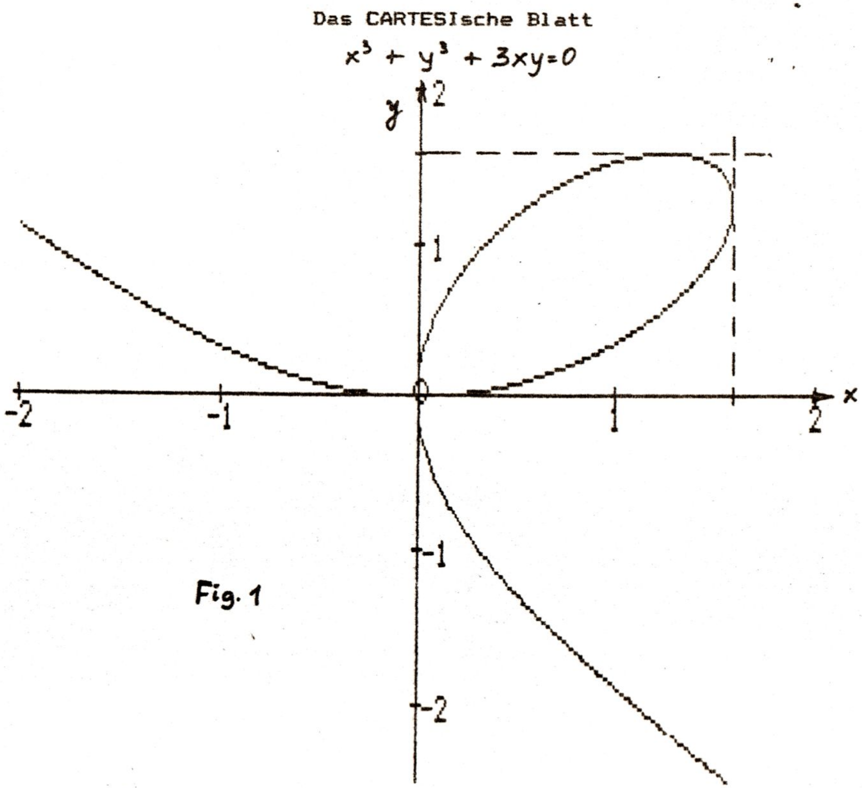


Fig. 1

Listing 1 zu Fig.1

```
10 '          ########################################
20 '          #                                      #
30 '          #      Das CARTESIsche Blatt (1)       #
40 '          # K.J.Muehlenbein, Weinheim, Jan.1987 #
50 '          #                                      #
60 '          ########################################
70 '
80 CLS:DEFINTI-K:DIMX(678),Y(678):W%=2:GOSUB250
90 FORN=0TO1.58STEP.01:GOSUB220:NEXT:IFW%=1THEN110
100 W%=0:FORN=1.58TO0STEP-.01:GOSUB220:NEXT:W%=1:GOTO90
110 CLS:-LOADRV("MX80RB "):-WINDOW(-2.2,2.2,-2.5,2.2):-POLYGO(
X,Y,678)
120 -MOVETO(-2,0):-MOVEON(2,0):-MOVETO(0,2):-MOVEON(0,-2.5)
130 SX(1)=-2:SY(1)=0:SX(2)=-1:SY(2)=0:SX(3)=1:SY(3)=0:SX(4)=2:
SY(4)=0
140 SX(5)=0:SY(5)=2:SX(6)=0:SY(6)=1:SX(7)=0:SY(7)=-1:SX(8)=0:S
Y(8)=-2
150 SX(9)=1.59:SY(9)=0:SX(10)=0:SY(10)=1.59
160 -STSIZE(2):-MARKPT(6,SX,SY,10):-STSIZE(1)
170 FORJ=0TO3STEP3:FORK=J-2TOJ-1
180 -NWRITE(K,K,-.2,0,2,0,"M"):-NWRITE(K,.1,K-.1,0,2,0,"M")
190 NEXTK,J
200 U(1)=0:V(1)=0:-STSIZE(2):-MARKPT(10,U,V,1)
210 GOTO210
220 I=I+1:X(I)=N:C=-SQR(N*N*N/4):PRINT§55,I
230 PHI=(-ATN(C/SQR(-C*C+1))+1.5708)+W%*6.2832
240 Y(I)=2*SQR(N)*COS(PHI/3):RETURN
250 FORN=-2TO0STEP.01:I=I+1:PRINT§55,I:X(I)=N
260 A1=-N*N*N/2:WU=SQR(N*N*N*N*N*N/4-N*N*N)
270 A2=A1+WU:A3=WU-A1:A4=A2Ä(1/3):A5=-A3Ä(1/3)
280 Y(I)=A4+A5:NEXT:RETURN
```

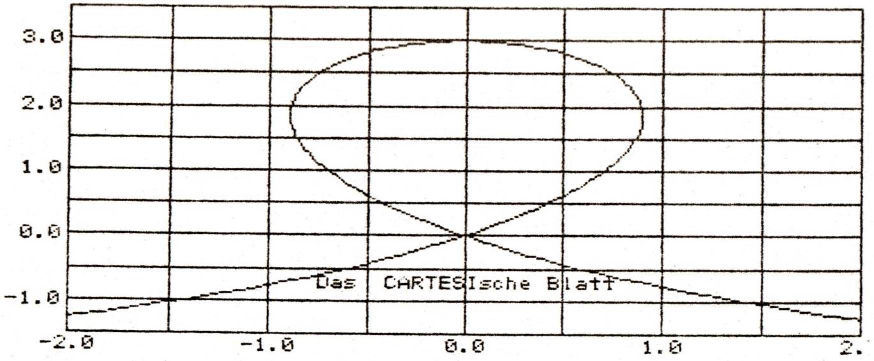


Fig. 2

Listing 2 zu Fig.2

```
10 '          ########################################
20 '          #                                      #
30 '          #      Das CARTESIsche Blatt (2)       #
40 '          # K.J.Muehlenbein, Weinheim, Jan.1987 #
50 '          #                                      #
60 '          ########################################
70 '
80 CLS:DIMX(101),Y(101):DEFINTI
90 FORZ=-2TO3STEP.05:I=I+1:PRINT§59,I:Y(I)=Z
100 X(I)=Z*SQR(ABS((3-Z)/(3+Z))):NEXT
110 -NFRAME(0,0):-DEFPIC(-.9,-.9,-.1,-.1)
120 -WINDOW(-2,2,-1.5,3.5)
130 -AXIS("X",-1,0.5,"G",2,1,2)
140 -AXIS("Y",-1,0.5,"G",2,1,2):GOSUB180
150 FORI=1TO101:X(I)=-X(I):PRINT§59,I:NEXT:GOSUB180:CLS
160 -TEXT("Das  CARTESIsche Blatt",22,0,-.8,0,"M")
170 GOTO170
180 -POLYGO(X,Y,101):RETURN
```

40 Tracks ?   Warum nicht 42 ?

Es kommt natürlich auf die mechanischen Toleranzen des Laufwerks an. Bei mir, (TRS-80 mod.4) kann ich auf die Antwort "Number of cylinders" ruhig die Zahl 42 eingeben, wie nachstehender Ausdruck beweist. Das Directory steht dann auf Zylinder 21.
Für Alle, welche 9k zusätzlich auf den Datendisketten gebrauchen können; d.h. statt 174k sind es dann 183k.
Aber Vorsicht beim Austausch von Disketten, es ist möglich, dass es nicht bei allen Computern funktioniert !
(Aus der Zeitschrift TRACE 1/82)

Helmut Obscherningkat

```
TRSDOS Ready
DIR :1

Drive :1  BASIC2     40 Cyl, DDEN, Free =  174.00K /   180.00K,  Date 04-Feb-87

TRSDOS Ready


TRSDOS Ready
FORMAT

FORMAT - 6.2.0 - Copyright 1982/83/84 by Logical Systems, Inc.
All Rights Reserved. Licensed 1982/83/84 to Tandy Corporation.

Which drive is to be used ? 1
Diskette name ? BASIC2
Master password ? PASSWORD
Single or Double density <S,D> ? D
Number of cylinders ? 42
Disk contains data -- Name=BASIC2    Date=02/04/87
Are you sure you want to format it ? Y
Formatting cylinder 41
Verifying  cylinder 41
Directory will be placed on cylinder 21

Initializing DIRECTORY information: ......................

Formatting complete

TRSDOS Ready


TRSDOS Ready
DIR :1

Drive :1  BASIC2     42 Cyl, DDEN, Free =  183.00K /   189.00K,  Date 04-Feb-87

TRSDOS Ready
```


Die Druckerabfrage

Wer hat sich nicht schon geärgert, wenn sich unsere Maschine aufhängt, nur weil gerade kein Drucker angeschlossen ist oder der angeschlossene nicht ON LINE geschaltet war.
Dabei ist die Lösung dieses Problems recht einfach: An der Druckeradresse (14312) können nämlich alle drei Zustände abgefragt werden. Mit nur 3 Zeilen kann somit jedes Programm "lernen" den jeweiligen Druckerstatus einzubeziehen, bevor es zu einer lästigen Schleife kommt. Dazu ein kleines Beispielprogramm:

```
1 '***************** DRUCKRDY/BAS *****************************
2 '* Dieses Programm zeigt, wie der aktuelle Status des
3 '* angeschlossenen Parallel-Druckers festgestellt werden
4 '* kann. Diese Zeilen können insbesondere dann abgefragt
5 '* werden, wenn das "Aufhängen" vermieden werden soll!
6 '* 18.1.1987, Paul-Jürgen Schmitz
7 '************************************************************
8 'Nachrichten an den Benutzer -----------------------------
9      A$="Drucker hat keinen Saft!"
10      B$="Drucker ist ON LINE "
11      C$="Drucker OFF LINE "
12 'Abfrageschleife ---------------------------------------
13    DR=PEEK(14312)
14    IF DR<16 THEN PRINTA$:GOTO 17
15    IF DR<65 THEN PRINTB$:GOTO 17
16    PRINT C$
17 GOTO 13
```

Bei mir nutze ich die hier dargestellte Möglichkeit im Rahmen des Systemstarts. Denn mit AUTO wird ein JOB gestartet, der eine Reihe lästiger Arbeiten übernimmt und insbesondere dem Drucker verschiedene Steuerzeichen übermittelt. Wenn ich den Drucker nicht benötige, kann ich ihn beim Start gefahrlos auslassen. Das System überspringt dann den LPRINT-Befehl.

Brasilianisch: Macaco velho nao mete a mao em cumbuca.
Ein alter Affe steckt seine Hand nicht in ein Kürbisgefäß.
Ein gebranntes Kind scheut das Feuer.

Französisch: Manger la grenouille.
Den Frosch fressen.
Mit der Kasse durchbrennen.
Viktor Kunzemüller, Freiburg

Il est noir comme un téléphone.
Er ist schwarz wie ein Telefon.
Er ist total blau.
Astrid Krauß, Allersberg

Keine Angst vor Proportionalschrift!

Viele bei uns im Club vertretene Drucker beherrschen die Proportionalschrift, bei der z. B. ein i nicht gewaltsam mit breiten Serifen beglückt wird, damit es optisch ungefähr die Breite eines M bekommt. Alle Buchstaben werden stattdessen so breit gedruckt, wie man sie auch mit der Hand schreiben würde.

Die Proportionalschrift ist in der Anwendung leider sehr problematisch. Alle Textprogramme, die einen Randausgleich bieten, gehen von gleicher Breite aller Buchstaben aus. Wenn bis zum rechten Rand noch wenige Zeichen fehlen, wird in der Zeile mit verteilten zusätzlichen Leerzeichen aufgefüllt. Das erklärt die unterschiedlich breiten Abstände zwischen den Wörtern bei fast allen Artikeln in diesem Heft.

Wie das mit Proportionalschrift aussähe, ist hier demonstriert. Die Zeichen sind natürlich von unterschiedlicher Breite, was das Textprogramm nicht weiß. Es zählt weiterhin brav die Zeichen und gibt sich mit dem rechten Rand alle Mühe – umsonst. Bei diesem Flattersatz könnte man gleich auf den Randausgleich verzichten. Dann wären wenigstens die Abstände zwischen den Wörtern gleich breit.

Dennoch gibt es einen Weg. Der offensichtlich untaugliche Randausgleich muß ausgeschaltet werden. Bei TSCRIPS geht das mit >J=N■. Der Randausgleich wird nun dem Drucker überlassen. Dazu muß ihm unabhängig von den Randangaben für das Textprogramm mitgeteilt werden, wo er die Ränder setzen soll. Die Codefolge sieht am Beispiel des NEC P6 folgendermaßen aus:

ESC 'a' 03h	Randausgleich rechts und links
ESC 'l' 09h	linken Rand auf die 9. Stelle setzen
ESC 'Q' 47h	rechten Rand auf die 71. Stelle setzen

Da TSCRIPS leider das Q nicht einwandfrei bearbeitet, wird diese Codesequenz etwas umständlich eingegeben:

$a&03&$1&091B5147&

ESC und Q werden in Hex als 1B und 51 codiert. Die hier gewählten Werte für die Ränder sind selbstverständlich dem persönlichen Geschmack überlassen.

Einen Nachteil hat diese Methode, wie man an diesem Text erkennt. Der NEC P6 versucht den Ausgleich des rechten Randes auf jeden Fall, auch in der letzten Zeile eines Absatzes. Er kann ja nicht wissen, daß wir in diesem Falle darauf keinen Wert legen. Erst wenn eine Verbreiterung der Zeichen-Zwischenräume um höchstens 50% nicht mehr hilft, gibt er auf und druckt linksbündig. Die Abhilfe wäre kompliziert: Mit dem W-Kommando des TSCRIPS wird die Druckzeilenlänge auf dem Bildschirm eingestellt. Vor die jeweils letzten Zeilen eines Absatzes kommt dann der Druckerbefehl zu Ausschalten des Randausgleichs, dahinter der Befehl zum Wiedereinschalten. Und da diese zusätzlichen Codes von TSCRIPS leider bei der Zeilenlänge mitgezählt werden, muß obendrein mit >RM=n■ ein möglichst weiter rechter Rand vorgegeben werden, der danach auch wieder rückgängig zu machen ist.

Ich habe mir anders geholfen: Das hier gelistete Programm zeichnet einen senkrechten HRG-Strich am linken Rand der 46. Bildschirmstelle in allen Zeilen. Was links davon liegt, wird sowieso linksbündig gedruckt. Rechts davon muß die Zeile mindestens die 65. Stelle erreichen, also soeben den Bildschirmrand überschreiten, um optisch erträglich gedehnt zu werden. Was zwischen dem Strich und dem Bildschirmrand liegt, wird dann einfach kürzer oder länger formuliert. Das ist in diesem Artikel aber nicht geschehen, um den Effekt der Dehnung zu zeigen. Das Zusatzprogramm ist nun Bestandteil von TSCRIPS (s. Sektordump). Für andere HRG-Systeme als das des G3s muß es entsprechend umgeschrieben werden.

Arnulf Sopp


```
                00001 ;Patch in TSCRIPS, um die Proportionalschrift in den Griff zu kriegen
                00002
4200            00003        ORG     4200h        ;im Sektorpuffer, wo es nicht stört
4200  210080    00004 start  LD      HL,8000h     ;Anfangsadresse des HRG-Speichers
4203  110180    00005        LD      DE,8001h     ;die zweite Stelle der HRG
4206  013F00    00006        LD      BC,0031h     ;Länge einer Bildschirmzeile -1
4209  DBFA      00007        IN      A,(0fah)     ;Systemport, der u. a. die HRG steuert
420B  F5        00008        PUSH    AF           ;Zustand des Ports retten
420C  F608      00009        OR      08h          ;die HRG f. Lesen und Schreiben freigeben
420E  F3        00010        DI                   ;lieber keine Interrupts
420F  D3FA      00011        OUT     (0fah),A     ;Systembyte neu ausgeben
4211  75        00012        LD      (HL),L       ;erste HRG-Stelle löschen (L=00)
4212  EDB0      00013        LDIR                 ;erste HRG-Zeile löschen
4214  2E2E      00014        LD      L,2eh        ;die 46. Bildschirmstelle adressieren
4216  74        00015        LD      (HL),H       ;dort den linken Punkt setzen (H=80)
4217  69        00016        LD      L,C          ;den HRG-Anfang adressieren (C=00)
4218  01C07F    00017        LD      BC,7fc0h     ;Länge der gesamten HRG minus 1 Zeile
421B  EDB0      00018        LDIR                 ;den Punkt als Strich in die HRG schr.
421D  F1        00019        POP     AF           ;alten Zustand des Systemports ...
421E  D3FA      00020        OUT     (0fah),A     ;... wiederherstellen
4220  FB        00021        EI                   ;jetzt stören die Interrupts nicht mehr
4221  C30091    00022        JP      9100h        ;zum Entrypoint von TSCRIPS springen
4200            00023        END     start        ;hier neue Entry

00000 Fehler
```

```
DRV 00  5F18 2630 02CB F906 263E 23B9 2815 3CB9  _.&0....&)#.(.<.
0   10  2811 78B9 280D 3E5E B928 083C B928 043C  (.x.(.>^.(.<.(.<
0H  20  B920 0670 23CD 6840 4871 2322 2B7C C901  ...p#.h§Hq#"+ö..
    30  04B6 82DE 7A01 0442 84DE 7A01 2600 4221  ....z..B..z.&.B!
DRS 40  0080 1101 8001 3F00 DBFA F5F6 08F3 D3FA  ......?.........
929 50  75ED B02E 2E74 6901 C07F EDB0 F1D3 FAFB  u....ti.........
3A1H60  C300 9102 0200 4200 0000 0000 0000 0000  ......B.........
    70  0000 0000 0000 0000 0000 0000 0000 0000  ................
    80  0000 0000 0000 0000 0000 0000 0000 0000  ................
    90  0000 0000 0000 0000 0000 0000 0000 0000  ................
    A0  0000 0000 0000 0000 0000 0000 0000 0000  ................
    B0  0000 0000 0000 0000 0000 0000 0000 0000  ................
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000  ................
69  D0  0000 0000 0000 0000 0000 0000 0000 0000  ................
45H E0  0000 0000 0000 0000 0000 0000 0000 0000  ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Neues zur Proportionalschrift

Das im letzten Artikel zu diesem Thema beschriebene Problem ist gelöst. Der Graphikstrich ist nicht mehr erforderlich, weil ein anderes Programm selbsttätig herausfindet, ob die letzte Zeile eines Absatzes erreicht ist, wo man keinen rechten Randausgleich mehr wünscht.

Ein Problem besteht darin, daß vom Drucker nicht abgefragt werden kann, in welchem Betriebszustand er sich gerade befindet (Schriftart, Randausgleich, Graphik usw.). Um sichergehen zu können, muß ihm der Befehl zum Links-/Rechts-Randausgleich deshalb in einer Form gegeben werden, die das Textprogramm jederzeit feststellen kann. Deshalb wurde ein neuer Steuercode für TSCRIPS eingeführt: Der Code 7F bedeutet normalerweise für den Drucker Delete, er ist also frei.

An der Stelle 86E3h in TSCRIPS 5.4 werden alle Textzeichen an den Drucker ausgegeben. Hier steht nun ein Patch, der die Ausgabe abfängt (CALL check7f, s. Listing). Im UP check7f wird geprüft, ob ~~ob~~ der neue Steuercode gerade im Akku steht. Falls nein, wird das andere Zeichen über die Druckerroutine 05B4h des Interpreters ausgegeben. War es aber 7F, ist noch lange nicht alles klar. Es könnte sich ja um ein 7F-Byte für ein HRG-Zeichen handeln, weil gerade ein Buchstabe eines IMAGEx/DAT-Files geschrieben wird. In diesem Falle geht das Byte ebenfalls unverändert an den Drucker weiter. Ist es aber unser Steuerzeichen, wird es spannend:

Jetzt soll nicht mehr auf 7F geprüft werden, denn der Stein rollt bereits. Stattdessen ist es nun CR (carriage return, 0Dh), worauf geprüft werden soll. Die Adresse von check0d wird deshalb in TSCRIPS für das neue UP gepatcht. Zuvor wird die Codefolge für den Randausgleich an den Drucker ausgegeben. Sie steht am Label fulljst. Die Ausgabe übernimmt das spezielle UP prtstr zum Drucken von Zeichenketten, das mehrmals im Programm aufgerufen wird. Die zu druckenden Zeichenketten müssen entweder mit CR oder FFh aufhören. Es arbeitet ganz ähnlich wie 446A im DOS. Der Ende-Code 03h wäre hier allerdings fatal, weil er bei fulljst vorkommt und mit auf den Drucker gelangen soll. FFh kommt praktisch nie vor und war deshalb hier zu bevorzugen.

Nun laufen alle Druckerausgaben über das UP check0d. Dort muß zunächst geprüft werden, ob ein einlaufendes 0Dh-Byte vielleicht auch zu einem IMAGE-Zeichen gehört. In diesem Falle wäre der Inhalt der Speicherstelle 8941h >00h. Dann wird das Byte eben ausgedruckt. Ist es aber ein gewöhnliches CR, wird überprüft, ob es sich um eine Kurzzeile handelt (letzte Zeile eines Absatzes oder eine kurze Zeile wie diese.
Um das feststellen zu können, war ein Umweg nötig. Alle einlaufenden Codes müssen dazu in einem Puffer gesammelt werden, der erst dann ausgedruckt wird, wenn ein CR erscheint. Das UP buffchr übernimmt die Pufferung.

Nun ist also endlich ein CR angekommen. Ist es eine Kurzzeile? In diesem Falle wäre die Anzahl der gepufferten Zeichen kleiner als die mögliche Zeilenlänge. Die ergibt sich aus der Differenz von RM (Einstellung des rechten Randes) und LM (linker Rand). Diese beiden Werte sind in 7C64h und 7C65h gespeichert und werden dort abgefragt. Vorsichtshalber werden vom Resultat noch 10 Zeichen subtrahiert, denn bis zum Ende eines vollen Wortes oder bis zu einem Trennungsstrich muß ja nicht die höchstmögliche Zeichenzahl erreicht sein. Wenn der so ermittelte Wert mindestens soeben erreicht ist, wird die Zeile ausgedruckt.

Ist sie nicht erreicht, wollen wir den Ausgleich des rechten Randes abschalten. Die zuständige Codefolge (Label leftjst) wird über prtstr ausgegeben. Dann folgt der Inhalt der aktuellen Zeile. Schließlich wird der Randausgleich wieder eingeschaltet. Die Zeile ist nun gedruckt, der Zeiger auf die aktuelle Pufferstelle wird auf 0 gesetzt. Das war's.

Das ist natürlich Partisanen-Heimarbeit, die nicht ohne eine gewisse Vorsicht zu genießen ist. So sind z. B. Druckersteuerungen verboten, die das

Byte FFh benötigen. FFh würde nicht mit ausgegeben. Es wird auch jedesmal der rechte Randausgleich geschaltet, nachdem eine Kurzzeile gedruckt ist. Wer weiß, vielleicht möchte der User weiter hinten im Text lieber Flattersatz haben. In diesem Falle müßte man nach jedem Absatz den Befehl dafür wiederholen, denn die Initialisierungen nach dem Auftauchen von 7Fh sind nicht rückgängig zu machen. Es erschien mir aber übertrieben, das Programm für Eventualitäten vorzubereiten, die so gut wie nie auftauchen. Wer mit dergleichen rechnet, muß ja nicht den Randausgleich über das Steuerzeichen 7Fh ausgeben.

Am Ende des Listings beim Label start wird in der Druckroutine des Interpreters eine Änderung vorgenommen, die verhindern soll, daß im Printer-DCB die gedruckte Zeilenzahl vermerkt wird. Dafür wird nämlich das Indexregister IX benutzt, das aber in TSCRIPS gar nicht auf den DCB, sondern ganz woanders hinweist. Die Folgen könnten verheerend sein. Ein schlichtes RET an der betreffenden Stelle bricht rechtzeitig den Vorgang ab. Wenn das RET an Ort und Stelle ist, wird TSCRIPS angesprungen.

Arnulf Sopp

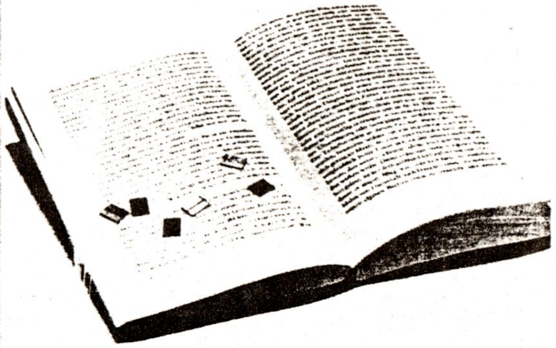
```
86E3          00001          ORG    86e3h       ;dort OUT (FD),A
86E3  CD1A31  00002          CALL   check7f     ;auf Signalzeichen 7F prüfen
86E4          00003 calladr  EQU    $-2         ;wird beim Auftauchen von 7F verändert
              00004
3000          00005 buffer   EQU    3000h       ;Zeilenpuffer von 3000-30BF
              00006
30C0          00007          ORG    30c0h       ;hier Druckerumleitung
30C0  00      00008 buffptr  DB     00h         ;zeigt zunächst auf den Pufferanfang
              00009
30C1  F5      00010 check0d  PUSH   AF          ;Zeichen retten
30C2  3A4189  00011          LD     A,(8941h)   ;Flag für Drucker-HRG
30C5  B7      00012          OR     A           ;Graphik aktiv (alle Codes erlaubt)?
30C6  28D4    00013          JR     Z,buffchr   ;Zeichen puffern, falls nicht aktiv
30C8  F1      00014          POP    AF          ;sonst Zeichen restaurieren
30C9  C3B405  00015          JP     05b4h       ;und ausdrucken
              00016
30CC  F1      00017 buffchr  POP    AF          ;Zeichen restaurieren
30CD  E5      00018          PUSH   HL          ;Textzeiger retten
30CE  21C030  00019          LD     HL,buffptr  ;Zeiger auf Puffer
30D1  E5      00020          PUSH   HL          ;seine Adresse retten
30D2  6E      00021          LD     L,(HL)      ;HL ist jetzt Pufferzeiger
30D3  77      00022          LD     (HL),A      ;das Zeichen in den Puffer einschreiben
30D4  E1      00023          POP    HL          ;HL <- buffptr
30D5  34      00024          INC    (HL)        ;Zähler um 1 Zeichen erhöhen
30D6  FE0D    00025          CP     0dh         ;CR?
30D8  2029    00026          JR     NZ,exit2    ;anderes Zeichen ausdrucken, falls nein
30DA  E5      00027          PUSH   HL          ;retten
30DB  21647C  00028          LD     HL,7c64h    ;dort RM gespeichert (rechter Rand)
30DE  7E      00029          LD     A,(HL)      ;in den Akku laden
30DF  23      00030          INC    HL          ;auf 7C65, dort LM gespeichert (l. Rand)
30E0  96      00031          SUB    (HL)        ;die Differenz errechnen
30E1  E1      00032          POP    HL          ;restaurieren
30E2  D60A    00033          SUB    0ah         ;vorsichtshalber 10 Zeichen weniger
30E4  32EB30  00034          LD     (linlen),A  ;ungefähre Zeilenlänge zum Vergleich
30E7  7E      00035          LD     A,(HL)      ;Zeiger auf Drucktext
30E8  2E00    00036          LD     L,00h       ;HL <- Anfang der Druckzeile
30EA  FE00    00037          CP     00h         ;schon wenigstens x-10 Zeichen bis CR?
30EB          00038 linlen   EQU    $-1         ;(je nach Diff. von RM - LM veränderlich)
30EC  300E    00039          JR     NC,exit1    ;ausdrucken, falls volle Zeile
              00040
```

```
30EE  E5                      PUSH  HL              ;nein, Kurzzeile: Pufferzeiger retten
30EF  211231                  LD    HL,leftjst      ;String für linksbündigen Druck
30F2  CD0531                  CALL  prtstr          ;ausgeben
30F5  E1                      POP   HL              ;Zeiger auf den Pufferanfang
30F6  CD0531                  CALL  prtstr          ;Zeile jetzt linksbündig ausdrucken
30F9  211631                  LD    HL,fulljst      ;String für rechten und linken Randausgl.
30FC  CD0531        exit1     CALL  prtstr          ;ausgeben
30FF  AF                      XOR   A               ;A <- 00h, Zeiger auf Pufferanfang
3100  32C030                  LD    (buffptr),A     ;der Puffer ist jetzt wieder frisch
3103  E1            exit2     POP   HL              ;Textzeiger restaurieren
3104  C9                      RET                   ;erledigt

3105  7E            prtstr    LD    A,(HL)          ;Zeichen des Strings laden
3106  FEFF                    CP    0ffh            ;Ende erreicht?
3108  C8                      RET   Z               ;falls ja
3109  CDB405                  CALL  05b4h           ;Zeichen drucken
310C  FE0D                    CP    0dh             ;CR erreicht?
310E  C8                      RET   Z               ;falls ja
310F  23                      INC   HL              ;nein, auf nächstes Zeichen zeigen
3110  18F3                    JR    prtstr          ;und weiter im String

3112  1B            leftjst   DB    1bh,'a',00h,0ffh     ;Sequenz für linken Randausgl.
3116  1B            fulljst   DB    1bh,'a',03h,0ffh     ;  "      "  vollen     "

311A  FE7F          check7f   CP    7fh             ;Signalzeichen für Randausgleich?
311C  C2B405        print7F   JP    NZ,05b4h        ;Zeichen audrucken, falls nein
311F  3A4189                  LD    A,(8941h)       ;Flag für Drucker-HRG
3122  B7                      OR    A               ;HRG aktiv (auch 7F erlaubt) ?
3123  3E7F                    LD    A,7fh           ;Code restaurieren
3125  20F5                    JR    NZ,print7F      ;falls Graphik aktiv
3127  E5                      PUSH  HL              ;sonst Textzeiger retten
3128  211631                  LD    HL,fulljst      ;Codesequenz für Rechts-/Linksausgleich
312B  CD0531                  CALL  prtstr          ;an den Drucker ausgeben
312E  21C130                  LD    HL,check0d      ;Prüfung auf CR
3131  22E486                  LD    (calladr),HL    ;in TSCRIPS patchen
3134  E1                      POP   HL              ;Textzeiger restaurieren
3135  C9                      RET                   ;zurück, ohne 7F auszudrucken

4200                          ORG   4200h           ;im Sektorpuffer, wo's nicht stört
4200  DBFA          start     IN    A,(0fah)        ;Systemport 1
4202  F5                      PUSH  AF              ;Zustand retten
4203  E6DF                    AND   0dfh            ;Schreibschutz 0000-2FFF aufheben
4205  D3FA                    OUT   (0fah),A        ;Systembyte neu schreiben
4207  3EC9                    LD    A,0c9h          ;Befehl RET
4209  32BE05                  LD    (05beh),A       ;dort patchen (DCB nicht aktualisieren)
420C  F1                      POP   AF              ;Systemport 1
420D  D3FA                    OUT   (0fah),A        ;alten Zustand restaurieren
420F  C30091                  JP    9100h           ;an alten Entrypoint von TSCRIPS springen

4200                          END   start           ;dort neuer Einsprung für TSCRIPS

00000 Fehler

buffchr 30CC   buffer  3000   buffptr 30C0   calladr 86E4   check0d 30C1   check7f 311A
exit1   30FC   exit2   3103   fulljst 3116   leftjst 3112   linlen  30EB   print7F 311C
prtstr  3105   start   4200
```


News:

Erster Megabit-Chip in Serie

Sindelfingen/Bundesrepublik: Noch scheinen die Grenzen der Miniaturisierung in der Siliciumtechnik lange nicht erreicht zu sein. Gerade ist der erste Megabit-Speicherchip in Serie gegangen. Dieses elektronische »Gedächtnis« ist in der Lage, rund 1 048 000 Einsen und Nullen zu speichern. Um sich diese Kapazität zu verdeutlichen: Auf sechs fingergroßen Chips kann ein 250 Seiten starkes Buch gespeichert werden! Damit auch wirklich kein Buchstabe verlorengeht, unterliegt jede Speicherzelle insgesamt 300 Kontrollen, die dann eine Lebensdauer des Speichers von 100 000 Betriebsstunden garantieren. ★



Sechs Megabit-Chips, die auf einem Buch liegen. Das Unvorstellbare: Auf den sechs Winzlingen läßt sich das Buch speichern.

LPRINT mit modernen Genies

Der Microsoft-Interpreter unserer Computerfamilie geht von ziemlich antiquierten Dampfdruckern aus, die außer dem Alphabet und Carriage Return nicht viel kapieren. Deshalb wird bei ASCII 0 nur die Druckerbereitschaft überprüft (von BASIC aus eh' nicht abfragbar), ASCII 10 wird als 13 ausgegeben und immer wird im Drucker-DCB ein Zeilenzähler unterhalten mit dem Verdacht, daß der Printer nicht einmal weiß, was ein Form Feed ist. Zu Gutenbergs Zeiten mag das noch einen Sinn gehabt haben; heute kostet es nur noch Speicherplatz und Rechenzeit.

Das leidige Problem von LPRINT CHR$(0) und (10) ist schon auf mehrfache Weise gelöst worden. Jedesmal aber war dazu ein Programm erforderlich, das die nötigen Manipulationen vornahm. TRS-80-kompatible Computer, die ein Microsoft-ROM haben, werden wohl auch weiterhin nicht um die Softlösung herumkommen. Diejenigen aber, die nur ein Boot-EPROM besitzen und sich den Interpreter von der Systemdiskette holen (Genie 2s, 3, 3s, vermutlich auch die höheren TRS-80), können diesen Ärger für alle Zeiten loswerden.

Dabei fallen so nebenbei weitere Vorteile ab. Wie aus den Sektordumps ersichtlich, wird in den betroffenen Ladebereichen RAM für Wichtigeres frei. Vom Drucker-DCB werden nur noch die drei Bytes von 4025-4027h gebraucht. Auch hier ist der Rest frei. Der größte Vorteil versteht sich von selbst: Alle Codes von 0-255 können unterschiedslos mit LPRINT auf den Drucker ausgegeben werden.

Die Patches werden im BASIC-Interpreter vorgenommen, der bei den Genies Bestandteil von SYS0/SYS ist. Beim G3s ist zusätzlich eine Änderung in OVL4/SYS nötig. Ob das auch bei den anderen Computern dieser Art sein muß, ist leicht herauszufinden: Man kann nach der Modifikation in SYS0/SYS neu booten und dann mit dem Debugger die betroffenen Speicherbereiche untersuchen. Die vielen Nullen müßten dann dort erscheinen. Ist das nicht der Fall, muß für die anderen Genies in den zugehörigen OVLx/SYS-Files die entsprechende Codefolge gesucht werden. Das sind OVL2/SYS für das G3 und OVL3/SYS für das G2s.

Die zu ändernden Ladebereiche sind 03A0-03B9h, 058E-05B3h und 05BE-05D0h. In den mir unbekannten Sektoren der Overlays 2 und 3 können diese Ladeadressen aus den Sektor-Headern errechnet werden, sofern man nicht FED/CMD hat, der eine Adresse selbsttätig suchen kann. Besagter FED hat übrigens eine fatale Methode, die Druckerbereitschaft abzufragen. Er geht mit gelöschtem Akku (ehemals Bereitschaftsabfrage!) über das UP 0023h, das nach den geschilderten Änderungen in dieser Weise nicht mehr benutzt werden kann. Da FED aber das einzige mir bekannte Programm ist, das den Drucker so umständlich belauert, ist es schon sinnvoller, FED zu ändern als auf die Vorteile der beschriebenen Patches zu verzichten:

rel. Skt. 18h, rel. Byte DA/DBh: 2300 -> D105

Grundsätzlich sollte man selbstverständlich vor jedem Zap im DOS ein Backup anlegen, um arbeitsfähig zu bleiben, falls wirklich einmal irgendein Programm mit der neuen Version nicht zusammenarbeiten will. Sollte dieser Fall mit den hier beschriebenen Patches in G-DOS auftreten, wäre ich für einen Hinweis dankbar. Es könnte u. U. sinnvoll sein, einen Teil der Patches rückgängig zu machen.

Arnulf Sopp

SYS0/SYS   Drive 0   Sector 23   X'0017'

```
0123456789ABCDEF  BYTE  00 01 02 03 04 05 06 07   08 09 0A 0B 0C 0D 0E 0F
=========================================================================
....#U.?......:=  (00)  01 FE EC 04 23 7D E6 3F   C0 11 C0 FF 19 C9 3A 3D
§..2=§..#U..o...  (10)  40 F6 08 32 3D 40 D3 FF   23 7D E6 FE 6F C9 11 80
....(......80(..  (20)  04 D5 FE 08 28 C0 FE 0A   D8 FE 0E 38 4F 28 A1 FE
.(...(...(...(..  (30)  0F 28 A2 FE 17 28 D7 FE   18 28 B7 FE 19 28 C5 FE
.(...(...(......  (40)  1A 28 BC FE 1B 28 C2 FE   1C 28 8D FE 1D CA A1 04
..(7..((.w#:=§..  (50)  FE 1E 28 37 FE 1F 28 3C   C9 77 23 3A 3D 40 E6 08
(.#ŏ.§........(!  (60)  28 01 23 7C FE 40 C0 11   C0 FF 19 E5 11 00 3C 21
§(..........U..o  (70)  40 3C C5 01 C0 03 ED B0   C1 EB 18 19 7D E6 C0 6F
..§..ŏ.§(...TU.?  (80)  E5 11 40 00 19 7C FE 40   28 E2 D1 E5 56 7D F6 3F
_......§6 #ŏ. .U  (90)  5F 13 18 04 E5 11 00 40   36 20 23 7C BA 20 F9 7D
. ...y.$........  (A0)  BB 20 F5 E1 C9 79 18 24   00 00 00 00 00 00 00 00
................  (B0)  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
................  (C0)  00 00 00 00 00 00 00 00   00 00 00 00 F5 CD D1 05
 ..2.7..........  (D0)  20 FB F1 32 E8 37 C9 00   00 00 00 00 00 00 00 00
..........:.7...0 (E0)  00 00 00 00 00 00 00 00   00 3A E8 37 E6 F0 FE 30
..)..3.H.I.. 0%.  (F0)  C9 E5 3E 0E CD 33 00 48   CD 49 00 FE 20 30 25 FE
```

OVL4/SYS • Drive 0   Sector 0   X'0000'

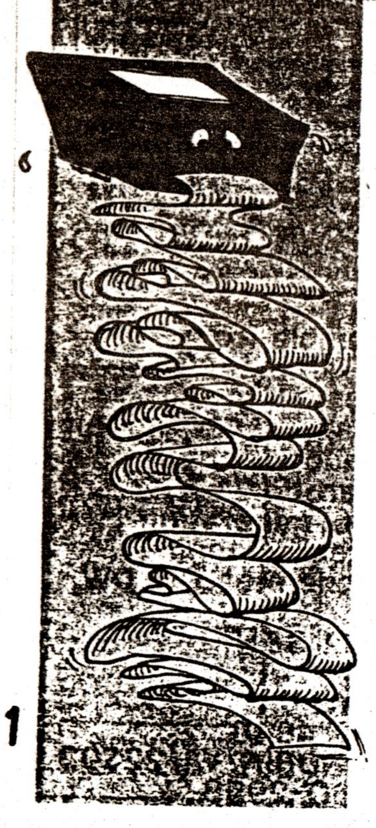
```
0123456789ABCDEF  BYTE  00 01 02 03 04 05 06 07   08 09 0A 0B 0C 0D 0E 0F
=========================================================================
..V.....2....).M  (00)  01 03 56 00 0B 01 05 C0   32 10 00 00 01 3E 00 4D
^ßß..^..........  (10)  5E 7E 7E 7F 7F 5E 00 00   00 00 00 00 00 00 00 00
................  (20)  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
................  (30)  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
..............B`7 (40)  00 00 00 00 00 00 00 00   00 00 00 00 01 42 60 37
nPV.......).....  (50)  6E 50 56 0C 1F 02 19 1C   02 09 29 09 00 00 00 00
n§P.......).....  (60)  6E 40 50 0A 14 06 10 12   02 0E 29 09 04 00 00 00
n§P.......).....  (70)  6E 40 50 0A 1B 09 18 1A   02 0A 29 09 02 00 00 00
n§P......./.....  (80)  6E 40 50 0A 13 02 10 11   03 0F 2F 0F 00 00 00 00
...2STD        .  (90)  01 0E C3 32 53 56 44 20   20 20 20 20 20 20 20 0D
...D0...D0... ..  (A0)  01 03 A2 44 30 01 03 CB   44 30 01 03 00 20 8B 01
)...............  (B0)  29 A0 06 F5 00 DB FA CB   C7 F3 D3 FA F1 C9 F5 DB
................  (C0)  FA CB 87 D3 FA F1 FB C9   F5 DB FA CB B7 D3 FA F1
...........'....  (D0)  C9 F5 DB FA CB F7 D3 FA   F1 C9 05 22 00 00 00 00
................  (E0)  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
................  (F0)  00 00 00 00 00 00 00 00   00 00 00 00 00 00 01 09
```

Pagelist/CMD

Produce paginated line listings and right justify your text

Model I

Rik Karlsson, Oakton, VA



I hate BASIC program line listings! Especially the way I write program lines: long, complex and without a memory-grabbing space in sight. They are just plain hard to read. That is, until now. Oh, yes, I still write the same way, but now I'm LLISTing using PAGELIST/CMD and it makes all the difference.

The Nov/Dec 1980 *80-U.S. Journal* featured an article on LLIST/CMD, a 'filter' for hardcopy BASIC line listings that gives the user a way to limit the length of any one printed line and also provides subsequent line indentations. This right-justified LLIST is good, but it was obvious that more needed to be done to satisfy me. So, I grabbed my modified-for-disk Editor-Assembler, Z-80 programming book, wish list of expected results, and went to work.

First, I decided to use LLIST/CMD as the basis for my new program. Okay, what was at the top of my need list? Pagination.

Pagination is the automatic halting of a hardcopy printout at the bottom of a page of material. This is important for a couple of reasons. One, paginated line listings are easier to store as a flat file than continuous roll ones. Second, if you are like me, you run off periodic 'proofs' of work in progress to check all of that good logic. I know, some of you folks sit down and flowchart, program in BASIC away from the keyboard, and don't have to see what you've done at the machine. But, I'm one who loves ideas more than all that workup work—go with the flow at the keyboard, I say, and see what happens! That's what interpreted BASIC's all about, right?

Periodic proofs of line listings use up valuable roll paper. I've found a source of 'used' paper (one side clear) for this purpose, and since it isn't in rolls, I'm stuck if I don't have pagination. If you aren't into recycling, here's one area where the savings can come in a hurry. Anyway, PAGELIST/CMD paginates or not, as you select.

After programming pagination into PAGELIST/ CMD, it was time to 'open up' the listings for readability. The answer to this need is called linefeed-on-colon. Even the longest, most complex BASIC lines have colons to separate statements. Here is where money can be made in readability.

Let's look at the anatomy of the assembly language program and see what's going on here (Listing 1).

Since PAGELIST/CMD is meant to be used under BASIC, I added a step in the initialization sequence of the former LLIST/CMD. Now, the PAGELIST/CMD execution module is automatically protected in high RAM by resetting the DOS top of memory pointer at 4049H. Lines 810 and 820 do this.

Lines 830-890 prompt the user for a maximum line length. The default value is 64 characters. An <ENTER>, instead of a number, in response to this query, is an acceptance of the default value and advances the program to a prompt for number of indentation spaces desired.

Lines 900-990 ask for and get the space response. Note that lines 950-970 insure that line length exceeds spaces requested. These two requests and queries are almost the same as programmed in LLIST/ CMD.

Lines 1000-1060 are new code, and request a yes or no response to whether pagination is desired. If it is, the indicator byte at 'PMODE' is set to one. An <ENTER> or 'N' answer advances the program to the next request, leaving pagination off.

Lines 1070-1140 ask for a yes or no response to activating the colon-on-linefeed option. A 'Y' causes the colon comparison code to be completed in the execution module.

Line 1150 is the end of initialization and jumps the program to DOS READY.

The INPUT routine, lines 1160-1320, is all new code as compared to LLIST/CMD. First, it uses the Level 2 input routine at address 0040H to wait for, and get, a user response. An RST 16 is executed to analyze the input. If a number is found, the carry flag is set and a RETurn is executed. Notice in the code after CALL INPUT (in the initialization area), how the carry flag is used to direct traffic.

If no decimal number is present, a check for an <ENTER> is made. If found, the Z flag is set and a RETurn is executed. The Z flag also is set for an 'N' response. Notice again, how the Z flag directs program flow in the initialization module.

If neither number, <ENTER> nor 'N' is found to be input, a 'Y' is looked for. No 'Y', no RETurn—an 'Input Error!' message is displayed.

The messages in PAGELIST/ CMD are printed by a CALL to 4467H. This is a generalized DOS print-to-screen routine which recognizes a 03H or 0DH stop byte. I added a few touches to the request messages for clarity. Lines 1360-1500 are the messages.

PAGELIST/CMD is really two separate programs, if you will. The initialization module beginning at label 'START' is actually a program that configures the execution module according to the user's wishes. Once used, the initialization module becomes irrelevant to PAGELIST operation. Notice that it is ORGed low in memory and really could be almost anywhere. The execution module starting at label 'LLIST', however, is the workhorse routine of PAGELIST.

PAGELIST's execution module is a 'filter'. Its job is to sit between the printer and the ROM LLIST/ LPRINT code and insure that hardcopy printing is carried out according to the user's desires. Inserting PAGELIST into the line of events is accomplished by lines 750-780 of the initialization module.

Okay, now that our gatekeeper to the printer is on the job, how does he do what we have asked him to do?

First, the C register contains the byte enroute to the printer. Lines 1520-1590 compare that byte with three codes of interest; carriage return, top of form and quote mark. The first two cause execution to be routed to a few bytes of code that reset the line character counter back to zero, then prints the code. Remember, we've specified a maximum line length acceptable, so that every time a linefeed is executed, the character counter must be reset to a starting value.

The recognition of a quote mark causes a swapping of bytes at FF50H. This is Mister Toggle. He gets the present value of the comparison byte at location 'CKQTE+1' (address FF4AH), and reverses it. If it is a zero, he makes it a one, a one becomes a zero. Why? Because that byte controls the linefeed on colon function later on, as we will see. If none of the three comparisons are found to be true, the character counter is incremented by one and another comparison is attempted.

The byte at 'CLCPR' plus one, is initialized to the ASCII code for a colon upon user selection of the linefeed-on-colon option. If no colon is detected, execution falls through to check for maximum line length. If the line is at that length, a carriage return is effected. If not, the byte is printed normally, and we're back to line 1520.

If a colon is detected, a jump is made to 'CKQTE'. Okay, if Mister Toggle is a one, execution is routed back upstream for normal handling. Why? Because a one will appear only if just one quote mark has been encountered, meaning that a colon has been detected within a quoted string. Since the only colons of interest are between BASIC program statements, no action is taken. If Mister Toggle is a zero, an even number of quotes (or none at all) has been detected, and the colon

Listing 1 — PAGELIST Assembly Language Program

```
00100 ;PAGELIST/CMD
00110 ;
00120 ;                    BY RIK KARLSSON
00130 ;                 3103 FOX MILL ROAD
00140 ;               OAKTON,VIRGINIA 22124
00150 ;
00160 ;THIS IS A GREATLY MODIFIED VERSION OF LLIST/CMD
00170 ;WHICH APPEARED IN 80-U.S. MAGAZINE OF NOV/DEC 1980.
00180 ;THIS PROGRAM ONLY RUNS UNDER A DISK OPERATING SYSTEM.
00190 ;
00200 ;THE KEY DIFFERENCES BETWEEN THIS PROGRAM AND LLIST/CMD
00210 ;INCLUDE THE PAGINATION OPTION, LINEFEED ON COLON OPTION,
00220 ;AND THE USE OF SOME DOS ROUTINES TO PRINT THE MESSAGES.
00230 ;THE INPUT ROUTINE CALLS LEVEL-II ROM AS WELL. WHEN
00240 ;INITIALIZING, <ENTER> ALWAYS ADVANCES THE USER TO THE
00250 ;NEXT INPUT PARAMETER THUS FORCING A REVIEW AND
00260 ;ACCEPTANCE OF EACH ONE.
00270 ;
00280 ;SUBROUTINES TO HANDLE THE PAGINATION OPTION ARE ADDED
00290 ;AS 'PGREQ','MODE' AND 'HOLD.' LINEFEED ON COLON IS
00300 ;ACTIVATED BY 'COLON' AND RECOGNIZED AT 'CLCPR.'
00302 ;THE BYTE AT 'CKQTE'+1 IS TOGGLED SO THAT NO LINEFEED
00304 ;WILL OCCUR ON A COLON ENCLOSED IN QUOTES.
```

is acted upon.

Now, execution falls through to carriage return, reset character counter, and check-page-length-in-lines subroutines. Line 1840, 'CALL MODE', executes a subroutine that increments the counter for number of lines printed, then checks for end of page. If 57 lines is detected, a message is displayed, and a wait for <ENTER> loop begun. <ENTER> causes execution to continue by entering spaces at the head of the next line and, finally, the next character being printed. Whew!

And, it happens that way for each character printed, folks. "But, what if I want to change the parameters? Do I have to go back to DOS and start all over again?" No, Virginia, there is a Santa Claus in two forms, PAGELIST/BAS, a companion program, or direct command input of parameters. The direct command input mode is fully covered in the lead in (Listing 1, lines 430-690) documentation to the assembly language program listing. In most cases, only the line counter reset is actually needed in practice. To reset the line counter to zero following completion of a full program line listing, 'POKE &HFF94,0.' For those few times when a more extensive reconfiguration is required, the PAGELIST/BAS BASIC program is more thorough, capable and automatic.

Here is what the PAGELIST/BAS program is all about.

Recall that PAGELIST/CMD was composed of two separate programs, an initialization/configuration module and an execution module. The execution module is the only part of the composite left operative in memory after initialization.

PAGELIST/BAS (Listing 2) is RUN under BASIC and is a BASIC language configuration module for the PAGELIST machine code execution module. When RUN, it first checks to see that the execution module is indeed active. This is accomplished by examining the lineprinter device control block (DCB). If the DCB printer driver address points to the start address of the execution module, we're in business. If not, the fact is reported, and the BASIC program is terminated. To use PAGELIST in this case, you will have to start over

```
                00310 ;
                00320 ;ALSO ADDED IS AN AUTOMATIC MEMORY PROTECT FEATURE
                00330 ;AT INITIALIZATION WHICH SETS THE DOS MEMORY SIZE
                00340 ;POINTER AT 4049H TO THE START OF THE EXECUTION MODULE
                00350 ;MINUS ONE BYTE. THE USER CAN NOW ENTER BASIC, LOAD AND
                00360 ;RUN A PROGRAM AND THEN BE ABLE TO LLIST WITH THE
                00370 ;PARAMETERS SET WITHOUT SETTING MEM SIZE. IT MUST BE
                00380 ;NOTED THAT SINCE THIS ROUTINE RUNS AS A FILTER
                00390 ;BETWEEN THE BASIC ROM ROUTINES AND THE PRINTER
                00400 ;OUTPUT ROUTINE, ANY PRINTING TO HARDCOPY BE AFFECTED
                00410 ;BY PARAMETERS SET FOR PAGELIST.
                00420 ;
                00430 ;PAGELIST/CMD PARAMETERS CAN BE RESET FROM BASIC. THIS
                00440 ;IS IMPORTANT IF THE USER LLISTS MORE THAN ONE PROGRAM.
                00450 ;THE PAGINATION COUNTER MUST BE RESET TO ZERO BEFORE
                00460 ;ATTEMPTING ANOTHER LLIST. TO RESET THE PAGE COUNTER
                00470 ;TO ZERO, IN BASIC COMMAND MODE, ENTER:
                00480 ;        POKE &HFF99,0
                00490 ;
                00500 ;TO RESET THE NUMBER OF SPACES FROM BASIC, ENTER:
                00510 ;        POKE &HFF5C, <NUMBER OF SPACES>
                00520 ;
                00530 ;NOTE: THE NUMBER OF SPACES MUST BE LESS THAN LINE LENGTH
                00540 ;
                00550 ;TO RESET THE LINE LENGTH FROM BASIC, ENTER:
                00560 ;        POKE &HFF39, <LENGTH DESIRED>
                00570 ;
                00580 ;TO TURN OFF THE PAGINATION FEATURE FROM BASIC, ENTER:
                00590 ;        POKE &HFF98,0
                00600 ;
                00610 ;TO RESET THE NUMBER OF LINES PER PAGE, ENTER:
                00620 ;        POKE &HFF76, <LINES DESIRED>
                00630 ;
                00640 ;TO NEGATE TO LINEFEED ON COLON FEATURE, ENTER:
                00650 ;        POKE &HFF34,0
                00660 ;
                00670 ;THE COMPANION PROGRAM PAGELIST/BAS WILL ALLOW AUTOMATIC
                00680 ;ACCESS TO AND CHANGE OF PAGELIST/CMD PARAMETERS UNDER
                00690 ;BASIC. THE PROGRAM IS PRESENTED SEPARATELY.
                00700 ;
                00710 ;        MARCH 28, 1981  ROLLINGCREST
                00720 ;
                00730 ;
8000            00740          ORG     8000H           ;ORIGIN INIT MODULE
8000 2A2640     00750 START    LD      HL,(4026H)      ;GET LP DRIVER ADDRESS
8003 2254FF     00760          LD      (DRIVER+1),HL   ;
8006 2260FF     00770          LD      (DRIV2+1),HL    ;
8009 2268FF     00780          LD      (ENDIT+1),HL    ;
800C 2120FF     00790          LD      HL,LLIST        ;EXECUTION MODULE START
800F 222640     00800          LD      (4026H),HL      ;
8012 2B         00810          DEC     HL              ;BACK ONE SPACE
8013 224940     00820          LD      (4049H),HL      ;PROTECT MEMORY
8016 219980     00830 LINE     LD      HL,MSG0         ;FIRST MESSAGE
8019 CD6744     00840          CALL    4467H           ;DISPLAY IT
801C CD6780     00850          CALL    INPUT           ;GET RESPONSE
801F 3007       00860          JR      NC,SPREQ        ;BYE
8021 CD5A1E     00870          CALL    1E5AH           ;CONVERT TO HEX
8024 7B         00880          LD      A,E             ;GET LSB
8025 3239FF     00890          LD      (MAXLEN+1),A    ;SET LINE LENGTH VALUE
8028 211781     00900 SPREQ    LD      HL,MSG1         ;NEXT MESSAGE
802B CD6744     00910          CALL    4467H           ;DISPLAY IT
802E CD6780     00920          CALL    INPUT           ;GET RESPONSE
8031 300D       00930          JR      NC,PGREQ        ;BYE
8033 CD5A1E     00940          CALL    1E5AH           ;CONVERT TO HEX
8036 3A39FF     00950          LD      A,(MAXLEN+1)    ;GET LINE LENGTH
8039 BB         00960          CP      E               ;COMPARE TO SPACES
803A 38EC       00970          JR      C,SPREQ         ;TOO MANY, TRY AGAIN
803C 7B         00980          LD      A,E             ;OKAY, GET LSB
803D 325CFF     00990          LD      (NUMSPC+1),A    ;SET SPACES NUMBER
8040 213A81     01000 PGREQ    LD      HL,MSG2         ;PAGINATION REQUEST
8043 CD6744     01010          CALL    4467H           ;DISPLAY IT
8046 CD6780     01020          CALL    INPUT           ;GET (Y/N) RESPONSE
8049 38F5       01030          JR      C,PGREQ         ;A NUMBER INPUT
804B 2804       01040          JR      Z,COLON         ;A 'N' INPUT
804D 2198FF     01050          LD      HL,PMODE        ;A 'Y' INPUT
8050 34         01060          INC     (HL)            ;SET MODE BYTE TO 'ON'
8051 215481     01070 COLON    LD      HL,MSG3         ;COLON ON REQUEST
```


reason. Just insure that the resident BASIC programs, with which PAGELIST/BAS will be merged, have lines beginning above number 10.

Not to belabor the obvious, but there may be occasions when a 'high' version of PAGELIST/BAS is desired—when a BASIC program to be LLISTed uses lines 1-10. You could make a PAGELIST/BAS with line numbers from, say 65520-65529. In this case, to operate it, MERGE, then 'RUN 65520' rather than RUN as the start command.

So, there you have it. Assemble PAGELIST/CMD, then key in and save PAGELIST/BAS (with ASCII option), and you are on your way to meeting your conviction. And, you know, it really is a better way!

Figure 1
PAGELIST Parameters
PAGINATION <OFF>
LINEFEED ON COLON <OFF>
LINE LENGTH = 64
SPACES = 2
CHANGE PARAMETERS (Y/N)?
N
READY
>

Figure 2
PAGELIST Parameters
PAGINATION <ON> — LINES PER PAGE = 57
LINE COUNTER = 0 (zero)
LINEFEED ON COLON <ON>
LINE LENGTH = 64
SPACES = 2
CHANGE PARAMETERS (Y/N)?
N
READY
>

Figure 3
PAGELIST Parameters
PAGINATION <ON> — LINES PER PAGE = 57
LINE COUNTER = 33
LINEFEED ON COLON <ON>
LINE LENGTH = 64
SPACES = 2
CHANGE PARAMETERS (Y/N)?
Y
DISCONNECT PAGELIST/CMD FROM SYSTEM (Y/N)? N
CHANGE PAGINATION STATUS (1=ON 0=OFF)? 1
RESET LINE COUNTER (Y/N)? Y
CHANGE LINES PER PAGE? 57

```
80F4 45         01430          DEFM    'ENTER MAX LINE LENGTH (DEFAULT=64)'
8116 03         01440          DEFB    03H             ;TERMINATOR
8117 45         01450 MSG1     DEFM    'ENTER NUMBER OF SPACES (DEFAULT=5)'
8139 03         01460          DEFB    03H             ;TERMINATOR
813A 50         01470 MSG2     DEFM    'PAGINATION DESIRED (Y/N)?'
8153 03         01480          DEFB    03H             ;TERMINATOR
8154 4C         01490 MSG3     DEFM    'LINEFEED ON COLON (Y/N)?'
816C 03         01500          DEFB    03H             ;TERMINATOR
FF20            01510          ORG     0FF20H          ;EXECUTION MODULE, 48K
FF20 79         01520 LLIST    LD      A,C             ;OUTPUT CHARACTER
FF21 FE0D       01530          CP      0DH             ;A <CR>?
FF23 2819       01540          JR      Z,OUTCR         ;YES, HANDLE IT
FF25 FE0C       01550          CP      0CH             ;TOP OF FORM CODE?
FF27 2815       01560          JR      Z,OUTCR         ;SAME AS <CR>
FF29 E5         01570          PUSH    HL              ;SAVE POINTER
FF2A FE22       01580          CP      '"'             ;A QUOTE?
FF2C CC8CFF     01590          CALL    Z,QUOTE         ;THEN TOGGLE BYTE
FF2F 2197FF     01600          LD      HL,TEMP         ;CHARACTER COUNT BYTE
FF32 34         01610          INC     (HL)            ;COUNT ONE MORE
FF33 FE00       01620 CLCPR    CP      0               ;COLON COMPARE OPTION
FF35 2812       01630          JR      Z,CKQTE         ;DO IT ON MATCH
FF37 7E         01640          LD      A,(HL)          ;GET VALUE
FF38 FE40       01650 MAXLEN   CP      64              ;MAX LINE LENGTH REACHED?
FF3A 2812       01660          JR      Z,CRLF          ;YES, SEND A <CR>
FF3C 1828       01670          JR      ENDIT-1         ;NO, OUTPUT CHARACTER
FF3E E5         01680 OUTCR    PUSH    HL              ;SAVE POINTER
FF3F 2197FF     01690          LD      HL,TEMP         ;CHARACTER COUNT BYTE
FF42 3600       01700          LD      (HL),0          ;RESET COUNT TO ZERO
FF44 CD6AFF     01710          CALL    MODE            ;SEE IF PAGINATION ON
FF47 181D       01720          JR      ENDIT-1         ;NOW GO ON
FF49 3E00       01750 CKQTE    LD      A,0             ;GET IT
FF4B B7         01770          OR      A               ;CHECK IT
FF4C 20E9       01780          JR      NZ,MAXLEN-1     ;A ONE, NO <CR>
FF4E C5         01790 CRLF     PUSH    BC              ;SAVE REGISTERS
FF4F 0E0D       01800          LD      C,0DH           ;<CR> CODE
FF51 3601       01810          LD      (HL),1          ;RESET CHAR COUNT TO ONE
EF53 CD0000     01820 DRIVER   CALL    $-$             ;OUTPUT TO PRINTER
FF56 E5         01830          PUSH    HL              ;SAVE TEMP LOCATION
FF57 CD6AFF     01840          CALL    MODE            ;PAGINATION ON?
FF5A E1         01850          POP     HL              ;RESTORE
FF5B 0605       01860 NUMSPC   LD      B,5             ;NUMBER SPACES TO PRINT
FF5D 0E20       01870 LOOP     LD      C,20H           ;SPACE ASCII CODE
FF5F CD0000     01880 DRIV2    CALL    $-$             ;PUT TO PRINTER
FF62 34         01890          INC     (HL)            ;CHAR COUNTER UP ONE
FF63 10F8       01900          DJNZ    LOOP            ;DO TIL DONE
FF65 C1         01910          POP     BC              ;RESTORE
FF66 E1         01920          POP     HL              ;RESTORE
FF67 C30000     01930 ENDIT    JP      $-$             ;BACK TO PRINTER
FF6A 2198FF     01940 MODE     LD      HL,PMODE        ;PAGINATION ON/OFF BYTE
FF6D 7E         01950          LD      A,(HL)          ;GET IT
FF6E B7         01960          OR      A               ;IS IT ON?
FF6F C8         01970          RET     Z               ;NO, GO BACK
FF70 2199FF     01980          LD      HL,COUNT        ;YES, POINT TO COUNTER
FF73 34         01990          INC     (HL)            ;UP IT ONE
FF74 7E         02000          LD      A,(HL)          ;GET VALUE
FF75 FE39       02010          CP      57              ;MAX LINES ALLOWED?
FF77 2801       02020          JR      Z,HOLD          ;YES, WAIT FOR PAGE
FF79 C9         02030          RET                     ;NO, KEEP GOING
FF7A 219AFF     02040 HOLD     LD      HL,MSG4         ;POINT TO WAIT MESSAGE
FF7D CD6744     02050          CALL    4467H           ;DISPLAY IT
FF80 CD4900     02060          CALL    0049H           ;WAIT TIL <CR>
FF83 CDC901     02070          CALL    01C9H           ;CLEAR SCREEN
FF86 2199FF     02080          LD      HL,COUNT        ;POINT TO COUNTER
FF89 3600       02090          LD      (HL),0          ;RESET TO ZERO
FF8B C9         02100          RET                     ;GO BACK NOW
FF8C 214AFF     02110 QUOTE    LD      HL,CKQTE+1      ;GET INDICATOR BYTE
FF8F 7E         02120          LD      A,(HL)          ;
FF90 B7         02130          OR      A               ;IS IT ZERO?
FF91 2802       02140          JR      Z,ONE           ;YES, THEN MAKE ONE
FF93 35         02150          DEC     (HL)            ;MAKE A ZERO
FF94 C9         02160          RET                     ;RETURN
FF95 34         02170 ONE      INC     (HL)            ;MAKE A ONE
FF96 C9         02180          RET                     ;RETURN
FF97 00         02200 TEMP     DEFB    0               ;CHARACTER COUNTER
FF98 00         02210 PMODE    DEFB    0               ;PAGE MODE INDICATOR
FF99 00         02220 COUNT    DEFB    0               ;LINE COUNTER BYTE
FF9A 1A1D       02230 MSG4     DEFW    1D1AH
```

from DOS.

When PAGELIST is active, the BASIC program reports the total present configuration of the execution module. Figures 1 and 2 show typical screen displays.

The user is offered a chance to change the parameters. Figure 3 shows a typical question and response sequence. An <ENTER> in response to any query is accepted as a bye, and that parameter is left unchanged. After the last query is answered, the program automatically cycles to the initial report mode for checking by the user. Answering an 'N' to the change parameters query ENDs the program by DELETEing PAGELIST/BAS from memory.

The way I use PAGELIST/BAS is to maintain it in ASCII form on disk (SAVE "PAGELIST/BAS: d", A"). In this way, it can be MERGEd with any BASIC program in memory, and RUN without harm to it. Of course, the line numbering system of one through 10 was chosen for this

```
8054 CD6744   01080         CALL  4467H          ;DISPLAY IT
8057 CD6780   01090         CALL  INPUT          ;GET (Y/N) RESPONSE
805A 38F5     01100         JR    C,COLON        ;A NUMBER INPUT
805C 2806     01110         JR    Z,QUIT         ;A 'N' INPUT
805E 2134FF   01120         LD    HL,CLCPR+1     ;WHERE COMPARE BYTE GOES
8061 3E3A     01130         LD    A,':'          ;COLON ASCII CODE
8063 77       01140         LD    (HL),A         ;PUT INTO PLACE
8064 C32D40   01150 QUIT    JP    402DH          ;INIT DONE, GO TO DOS
8067 219580   01160 INPUT   LD    HL,BUFF        ;POINT TO INPUT BUFFER
806A 0603     01170         LD    B,3            ;MAX INPUT LENGTH
806C CD4000   01180         CALL  0040H          ;WAIT FOR RESPONSE
806F 2B       01190         DEC   HL             ;BACK ONE
8070 D7       01200         RST   16             ;CHECK OUT INPUT
8071 D8       01210         RET   C              ;A NUMBER, GO BACK
8072 D60D     01220         SUB   0DH            ;A <CR>?
8074 C8       01230         RET   Z              ;YES, GO BACK
8075 D641     01240         SUB   41H            ;A 'N'?
8077 C8       01250         RET   Z              ;YES, GO BACK
8078 FE0B     01260         CP    0BH            ;A YES?
807A 2002     01270         JR    NZ,ERROR       ;NO, ANSWER UNACCEPTABLE
807C 3C       01280         INC   A              ;SET FLAGS
807D C9       01290         RET                  ;GO BACK
807E 218680   01300 ERROR   LD    HL,ERRMSG      ;POINT TO ERROR MESSAGE
8081 CD6744   01310         CALL  4467H          ;DISPLAY IT
8084 18E1     01320         JR    INPUT          ;GO BACK FOR INPUT
8086 49       01330 ERRMSG  DEFM  'INPUT ERROR! >'
8094 03       01340         DEFB  03H            ;TERMINATOR
0004          01350 BUFF    DEFS  4              ;MAX OF 4
8099 1C1F     01360 MSG0    DEFW  1F1CH          ;CLEAR SCREEN
809B 20       01370         DEFM  '          PAGELIST  PROGRAM'
80B4 1A1D     01380         DEFW  1D1AH
80B6 20       01390         DEFM  '           BY RIK KARLSSON'.
80CE 1A1D     01400         DEFW  1D1AH
80D0 2D       01410         DEFM  '________________________________'
80F2 1A1D     01420         DEFW  1D1AH
```

```
CHANGE SPACES? 2
CHANGE LINE LENGTH? 64
CHANGE LINEFEED ON COLON
  STATUS (1= ON 0=OFF)? 1<
```

```
FF9C 43       02240         DEFM  'CHANGE PAGE - PRESS <ENTER>'
FFB7 0D       02250 ZEND    DEFB  0DH            ;TERMINATOR
8000          02260         END   START          ;AUTOSTART AT INIT
00000 TOTAL ERRORS
27936  TEXT AREA BYTES LEFT
```

Listing 2 — PAGELIST BASIC Program Listing

```
1 IFPEEK(&H4027)<>&HFFANDPEEK(&H4026)<>
  &H20CLS:PRINT@512,"PAGELIST/CMD NOT A
  CTIVE:":PRINT:DELETE1-11
2 CLS:PRINT"PAGELIST PARAMETERS":PRINTS
  TRING$(33,45):IFPEEK(&HFF98)=1PRINT"P
  AGINATION <ON> - LINES PER PAGE = ";P
  EEK(&HFF76):PRINT"LINE COUNTER = ";PE
  EK(&HFF99):ELSEPRINT"PAGINATION <OFF>
  "
3 PRINT"LINEFEED ON COLON ";:IFPEEK(&HF
  F34)<>1THENPRINT"<ON>"ELSEPRINT"<OFF>
  "
4 PRINT"LINE LENGTH = ";PEEK(&HFF39):PR
  INT"SPACES = ";PEEK(&HFF5C):INPUT"CHA
  NGE PARAMETERS (Y/N)";CP$:IFCP$<>"Y"T
  HENDELETE1-11
5 CP$="":INPUT"DISCONNECT PAGELIST/CMD
  FROM SYSTEM (Y/N)";CP$:IFCP$="Y"THENP
  OKE&H4026,PEEK(&HFF54):POKE&H4027,PEE
  K(&HFF55):RUN
6 A=PEEK(&HFF98):INPUT"CHANGE PAGINATIO
  N STATUS (1=ON 0=OFF)";A:IFA<2POKE&HF
  F98,A:CP$="N":IFA=1INPUT"RESET LINE C
  OUNTER (Y/N)";CP$:IFLEFT$(CP$,1)="Y"A
  NDRIGHT$(CP$,1)="Y"THENPOKE&HFF99,0EL
  SEPOKE&HFF99,VAL(RIGHT$(CP$,LEN(CP$)-
  1))
7 A=PEEK(&HFF76):INPUT"CHANGE LINES PER
   PAGE";A:IFA>0POKE&HFF76,A
8 A=PEEK(&HFF5C):INPUT"CHANGE SPACES";A
  :IFA>0POKE&HFF5C,A
9 A=PEEK(&HFF39):INPUT"CHANGE LINE LENG
  TH";A:IFA<PEEK(&HFF5C)THEN9ELSEPOKE&H
  FF39,A
10 POKE&HFF34,&H3A:INPUT"CHANGE LINEFEE
  D ON COLON STATUS (1=ON 0=OFF)";A:IFA
  =0POKE&HFF34,1
11 GOTO2
```

Jiddisch: Gaj straschen die gens.
Geh lieber die Gänse erschrecken.
Du jagst mir keine
Angst ein.

Schwedisch: Det är svårt att lära
gamla hundar att sitta.
Es ist schwierig, alten Hunden das
Sitzen beizubringen.
Was Hänschen nicht lernt, lernt
Hans nimmermehr.

BASIC-Rätsel: **Die richtige Dimension**

Will man in BASIC eine Reihe von Variablen, die zusammengehören – zum Beispiel die monatlichen Durchschnittstemperaturen während eines Jahres – »unter einen Hut bringen«, so kann man sie zu einem Feld (Array) zusammenfassen. Innerhalb eines Feldes hat jede Variable den gleichen Namen, sie unterscheidet sich von den anderen nur durch ihren Index. Das ist eine Art »Hausnummer«, die in Klammern hinter dem Variablennamen steht. Man muß dem Rechner immer vorweg sagen, wie viele Elemente dieses Feld haben soll. Lediglich wenn es nicht mehr als zehn Elemente sind, kann bei den meisten Computern der entsprechende DIM-Befehl entfallen. Innerhalb eines Programms darf die gleiche Variable kein zweites Mal DIMensioniert werden.

Bleiben wir beim Temperaturbeispiel: Im folgenden Programm soll der Computer die zwölf Durchschnittswerte eines Jahres abfragen. Nach der ersten Eingabe steigt er mit einer Fehlermeldung für Zeile 100 aus. Wo liegt das Problem?

```
100 DIM TEM(12)
110 MONAT=MONAT+1
120     INPUT"TEMPERATUR-
";TEM(MONAT)
130 IF MONAT < 12 THEN
GOTO 100
```

Lösung auf Seite 103.

Ungarisch: Száz szónak is egy a vége.
Auch hundert Worte haben ein Ende.
Langer Rede kurzer Sinn.

Fragen Sie Frau Olga

In Alexanders Zimmer sammeln
sich die Computerspiele an,
bis daß der Junge nur noch stammeln
und nur noch Knöpfe drücken kann!
Ihr Rat? fragt Gerda H. aus Graben.

Frau Olga: Knöpfchen muß man haben!

●

Rat sucht auch Karin G. in Spitz:
Computerspiele sind ein Schmarren.
Ich frag' Sie, worin liegt der Witz,
allein auf ein Gerät zu starren
und nur gegen sich selbst zu spielen?

Frau Olga rät: Nach Partnern schielen.

●

Was wird uns, fragt Ruth M. aus Rethen,
noch an Geräten zugemutet?
Gleich welches Zimmer wir betreten,
überall piepst es oder tutet –
alles, weil Nick sagt, das sei schick.

Frau Olga: Piept's vielleicht bei Nick?

»Als Heiko zum Skatspielen noch in seine Stammkneipe ging, hat mir der Freitagabend besser gefallen!«

Playgirl der Woche

». . . und seinen Computer kriege ich auch noch klein!«

Italienisch: Tra dire e fare ce in mezzo il mare.
Zwischen Sagen und Tun ist das Meer.
Das ist leichter gesagt als getan.

Liebe Freunde,

Als neues Mitglied im Club möchte ich mich zunächst kurz vorstellen :

Ich heiße Rüdiger Sörensen, geb. am 30.3.56, Student der Meteorologie an der Uni Mainz. Mein Model IV habe ich seit nunmehr 3 Jahren, Schwerpunkte der Computerei liegen bei numerischen Modellen (Großrechner - geschädigt !) und im Ausprobieren von Programmiersprachen (z.Zt. 'C').

Hartmut Obermann bat mich, quasi als Einstand, ein paar Zeilen über Filter unter TRSDOS 6.x zu schreiben. Das will ich hier versuchen.

TRSDOS 6.x macht es dem Benutzer recht leicht, eigene I/O-Routinen zu schreiben und diese, auch resident, im DOS einzubinden. Als Beispiel dient hier ein Drucker-Filter, der die Übersetzung von TRSCII-Umlauten in Epson-Codierung vornimmt.

Die gesamte Ein/Ausgabe von TRSDOS ist byte-orientiert, das heißt, es werden Zeichen übergeben, und zwar an logische Devices (Geräte). Die Steuerung erfolgt über Supervisor Calls, das sind im Prinzip DOS-Calls, aber nicht auf feste Adressen, sondern als RST 28H codiert. TRSDOS unterscheidet :

@PUT, wenn an ein Device gesendet
@GET, wenn von einem Device angefordert
@CTL, wenn ein Kontrollbyte gesendet wird.

Der Device-Treiber/Filter muß die Behandlung der verschiedenen Requests selbst vornehmen. Damit er erkennen kann, welche Art (@PUT,@GET,@CTL) von Request vorliegt, wird das Flag - Register vom DOS voreingestellt :

C,NZ : @GET
Z,NC : @PUT
NZ,NC : @CTL

Beim Aufruf des Treibers/Filters zeigt das Register IX auf den Device Control Block (DCB) des Treibers/Filters. Das zu verarbeitende Byte wird im C-Register übergeben. Damit TRSDOS ein Filterprogramm im Speicher findet, muß es einen standartisierten 'Header' haben (s. Beispiel). Um das Filtermodul in den Speicher zu bekommen, ist eine Installation erforderlich. Das geschieht über ein Programm, welches der Benutzer zu schreiben hat, es hat folgende Aufgaben :

Das SET-Kommando, mit dem das Installationsprogramm gestartet wird, übergibt in DE an das Programm die Adresse eines freien DCB. Dieser wird an die Stelle MODDCB im Filter transferiert. Dann wird überprüft, ob das Programm mit SET gestartet wurde. Falls nicht, --> Abbruch. Als nächstes wird die höchste vom Benutzer noch verwendbare Speicheradresse ermittelt und der Filterteil des Programmes dahin verschoben, anschließend wird HIGH$ korrigiert. Dann wird TRSDOS noch über den Filter informiert und fertig.

Der Installationsteil des Programms ist im wesentlichen immer derselbe. Ich benutze ihn daher auch für alle Treiber/Filter, die ich schreibe.

Das Kopfteil des eigentlichen Filters sieht so aus :

Zunächst ein relativer Sprung an den Beginn des Codes. Dann folgt die Adresse des letzten Filter-Bytes, ein Byte, das die Länge des Modulnamens beinhaltet, der Name des Moduls (hier 6 Zeichen), zwei Byte für MODDCB, und zwei Nullbytes. Danach können Daten folgen, bis zum Codebeginn. Zunächst sollte der Filter anhand des Flagregisters prüfen, welche Art von Request vorliegt. Der vorgestellte Filter behandelt nur @PUT - Requests, alle anderen werden in der Device-Kette weitergereicht ohne Änderung. Das Weitergeben des Bytes erfolgt mit dem Supervisor Call (SVC) @CHNIO. Das DOS weiß dann, welches Device das nächste in der Kette ist.

Die Vorteile dieser Methode liegen auf der Hand. Es ist möglich, praktisch beliebig viele Filter vor ein Ein/Ausgabe - Gerät zu schalten und das DOS übernimmt die Kontrolle und Steuerung. Man kann (mit dem DOS - Kommando RESET) selektiv einzelne Filtermodule entfernen, allerdings i.A. ohne Rückgewinn des belegten Speicherplatzes.

Wer den Filter ausprobieren möchte, für den gibt's ein paar Tips :

1) das Programm soll mit einem Macro-Assembler (EDAS) übersetzt werden, der Output geht auf die Datei UMLAUT/FLT.

2) der Filter wird aktiviert durch :

SET *UM UMLAUT
FILTER *PR *UM

3) ev. Fehlermeldungen sollten sich selbst erklären.

4) Der Filter berücksichtigt selbstverständlich *nur* Zeichen, die über den @PRT SVC ablaufen. Das ist bei den meisten Programmen der Fall, aber nicht bei allen. Superscripsit ist eine Ausnahme. Übel dran ist man auch bei Programmen, die im zu druckenden Zeichen das Bit 7 löschen. Da hilft meist nur das Ändern des Druckertreibers.

5) Wenn man bei installiertem Filter ein File ausdrucken lassen will, geht das mit LIST FILESPEC (PRT,ASCII8).

Literatur :

Model 4 Technical Reference Manual, Section 2, Device Driver and Filter Templates.

Viel Spaß beim Ausprobieren. Wer Fragen hat :
Telefon 06131/32860.

Gruß, *Rüdiger.*

```
                   00003 ;
                   00004 ;          UMLAUT Version 1.1 1/86
                   00005 ;
                   00006 ;          Equates
                   00007 ;
000A               00008 LF         EQU     10
000D               00009 CR         EQU     13
00D1               00010 AE         EQU     209
00D2               00011 OE         EQU     AE+1
00D3               00012 UE         EQU     AE+2
00D6               00013 SAE        EQU     214
00D7               00014 SOE        EQU     SAE+1
00D8               00015 SUE        EQU     SAE+2
00D9               00016 SZ         EQU     SAE+3
000A               00017 @DSPLY     EQU     10
0014               00018 @CHNIO     EQU     20
0016               00019 @EXIT      EQU     22
0064               00020 @HIGH$     EQU     100
0065               00021 @FLAGS     EQU     101
                   00022 ;
                   00023 ;                  Macro Definitionen
                   00024 ;
0000               00025 SVC        MACRO   #NUMBER
0000               00026            LD      A,#NUMBER
0000               00027            RST     28H
0000               00028            ENDM
0000               00029 PSHAL      MACRO
0000               00030            PUSH    BC
0000               00031            PUSH    DE
0000               00032            PUSH    HL
0000               00033            ENDM
0000               00034 POPAL      MACRO
0000               00035            POP     HL
0000               00036            POP     DE
0000               00037            POP     BC
0000               00038            ENDM
                   00039 ;
3000               00040            ORG     3000H
3000 ED532331      00041 BEG        LD      (DCB),DE            ;rette DCB-Adresse
3004 ED533031      00042            LD      (MODDCB),DE         ;und übergebe sie an
ter
3008 E5            00043            PUSH    HL                  ;rette HL
3009 218D30        00044            LD      HL,SGNON            ;Installierundsmeldun
300C               00045            SVC     @DSPLY              ;           "
300C+3E0A          00046            LD      A,@DSPLY
300E+EF            00047            RST     28H
300F E1            00048            POP     HL                  ;hole HL
3010               00049            SVC     @FLAGS              ;und sieh nach, ob wi
SET
3010+3E65          00050            LD      A,@FLAGS
3012+EF            00051            RST     28H
3013 FDCB025E      00052            BIT     3,(IY+2)            ;gekommen sind
3017 200C          00053            JR      NZ,SETHI            ;OK, dann springe
3019 21B530        00054            LD      HL,SETMSG           ;Fehlermeldung
301C               00055            SVC     @DSPLY
301C+3E0A          00056            LD      A,@DSPLY
```



```
301E+EF            00057            RST     28H
301F 21FFFF        00058            LD      HL,-1               ;extended Error Message
3022               00059            SVC     @EXIT               ; --> TRSDOS
3022+3E16          00060            LD      A,@EXIT
3024+EF            00061            RST     28H
3025 210000        00062 SETHI      LD      HL,0                ;HIGH$ ermitteln
3028 45            00063            LD      B,L                 ;
3029               00064            SVC     @HIGH$              ;
3029+3E64          00065            LD      A,@HIGH$
302B+EF            00066            RST     28H
302C 222731        00067            LD      (OLDHI),HL          ;und retten
302F 280C          00068            JR      Z,RELOC             ;springe, wenn OK
3031 210431        00069            LD      HL,NOMEM            ;kein Speicherplatz frei
3034               00070            SVC     @DSPLY
3034+3E0A          00071            LD      A,@DSPLY
3036+EF            00072            RST     28H
3037 21FFFF        00073            LD      HL,-1
303A               00074            SVC     @EXIT
303A+3E16          00075            LD      A,@EXIT
303C+EF            00076            RST     28H
303D FD218A31      00077 RELOC      LD      IY,RELTAB           ;Neue Adressen des Filters b
erechnen
3041 118A31        00078            LD      DE,FLTEND           ;
3044 AF            00079            XOR     A                   ; 0 --> A
3045 ED52          00080            SBC     HL,DE               ;
3047 E5            00081            PUSH    HL                  ; HL --> BC
3048 C1            00082            POP     BC                  ;
3049 FD6E00        00083 RELOC1     LD      L,(IY+0)            ;trage die neuen Adressen im
 Filtermodul ein
304C FD6601        00084            LD      H,(IY+1)
304F 7C            00085            LD      A,H
3050 B5            00086            OR      L
3051 280F          00087            JR      Z,MOVE
3053 5E            00088            LD      E,(HL)
3054 23            00089            INC     HL
3055 56            00090            LD      D,(HL)
3056 EB            00091            EX      DE,HL
3057 09            00092            ADD     HL,BC
3058 EB            00093            EX      DE,HL
3059 72            00094            LD      (HL),D
305A 2B            00095            DEC     HL
305B 73            00096            LD      (HL),E
305C FD23          00097            INC     IY
305E FD23          00098            INC     IY
3060 18E7          00099            JR      RELOC1
                   00100 ;
3062 ED5B2731      00101 MOVE       LD      DE,(OLDHI)          ;verschiebe das Filtermodul
--> HIGH$
3066 218A31        00102            LD      HL,FLTEND
3069 016600        00103            LD      BC,FLTLEN
306C EDB8          00104            LDDR
                   00105 ;
306E EB            00106            EX      DE,HL               ;neuen HIGH$-Wert im TRSDOS
eintragen
306F 0600          00107            LD      B,0
3071               00108            SVC     @HIGH$
3071+3E64          00109            LD      A,@HIGH$
3073+EF            00110            RST     28H
```

```
3074 23          00111          INC     HL
3075 DD2A2331    00112          LD      IX,(DCB)          ;DCB im TRSDOS eintragen
3079 DD360041    00113          LD      (IX+0),01000001B          ;Filter für @PUT-Req
uests
307D DD7501      00114          LD      (IX+1),L
3080 DD7402      00115          LD      (IX+2),H
3083 21DF30      00116          LD      HL,DONMSG         ;OK-Meldung
3086             00117          SVC     @DSPLY
3086+3E0A        00118          LD      A,@DSPLY
3088+EF          00119          RST     28H
3089 210000      00120          LD      HL,0              ;HL = No Error
308C C9          00121          RET                       ;zurück zu TRSDOS
                 00122 ;
                 00123 ;                Messages
                 00124 ;
308D 55          00125 SGNON    DM      'UMLAUT Filter (c) Rüdiger Sörensen
     4D 4C 41 55 54 20 46 69
     6C 74 65 72 20 28 63 29
     20 52 DB 64 69 67 65 72
     20 53 D7 72 65 6E 73 65
     6E 20 31 39 38 36 0D
30B5 0A          00126 SETMSG   DM      LF,'Muß mit Kommando SET installiert
',CR
     4D 75 D9 20 6D 69 74 20
     4B 6F 6D 6D 61 6E 64 6F
     20 53 45 54 20 69 6E 73
     74 61 6C 6C 69 65 72 74
     20 77 65 72 64 65 6E 2E
     0D
30DF 0A          00127 DONMSG   DM      LF,'UMLAUT - Filter ist jetzt reside
     55 4D 4C 41 55 54 20 2D
     20 46 69 6C 74 65 72 20
     69 73 74 20 6A 65 74 7A
     74 20 72 65 73 69 64 65
     6E 74 2E 0D
3104 0A          00128 NOMEM    DM      LF,'Kein Speicherplatz verfügbar.',C
     4B 65 69 6E 20 53 70 65
     69 63 68 65 72 70 6C 61
     74 7A 20 76 65 72 66 DB
     67 62 61 72 2E 0D
3123 0000        00129 DCB      DW      0
                 00130 ;
                 00131 ;*******************************************************
*************
                 00132 ;
                 00133 ;              Hier beginnt der residente Filtertei
                 00134 ;
                 00135 ;*******************************************************
************
                 00136 ;
3125 181B        00137 BEGIN    JR      START
                 00138 ;
                 00139 ;              Speicherplatz fuer Daten
                 00140 ;
3127 0000        00141 OLDHI    DW      0
3129 06          00142          DB      6                 ;Länge des Modul - N
312A 55          00143          DM      'UMLAUT'          ;Name
     4D 4C 41 55 54
```



```
3130 0000        00144 MODDCB   DW      0                 ;für's System
3132 0000        00145          DW      0                 ;  "     "
3134 1B          00146 GERMAN   DB      27                ;Zeichenkette von 7 Byte, di
e von US auf
3135 52          00147          DB      'R'
3136 02          00148          DB      2                 ;deutschen Zeichensatz schal
tet. Zeichen
3137 00          00149 CHAR     DB      0                 ;übergibt und US wieder eins
chaltet.
3138 1B          00150          DB      27                ;
3139 52          00151          DB      'R'               ;CODES FUER EPSON RX-80 F/T
313A 00          00152          DB      0                 ;und TRS80 4
313B 5B          00153 TABLE    DB      91
313C 5C          00154          DB      92
313D 5D          00155          DB      93
313E 7B          00156          DB      123
313F 7C          00157          DB      124
3140 7D          00158          DB      125
3141 7E          00159          DB      126               ;Tabelle der Druckercodes
                 00160 ;
                 00161 ;        FILTER ROUTINE
                 00162 ;
3142 2810        00163 START    JR      Z,GOTPUT          ;springe, wenn @PUT - reques
t
3144             00164 FLTPUT   PSHAL
3144+C5          00165          PUSH    BC
3145+D5          00166          PUSH    DE
3146+E5          00167          PUSH    HL
3147 0602        00168          LD      B,2
3149 DD2A3031    00169          LD      IX,(MODDCB)       ;vorbereiten für SVC
314B             00170 RX01     EQU     $-2
314D             00171          SVC     @CHNIO            ;pass down the chain
314D+3E14        00172          LD      A,@CHNIO
314F+EF          00173          RST     28H
3150             00174          POPAL
3150+E1          00175          POP     HL
3151+D1          00176          POP     DE
3152+C1          00177          POP     BC
3153 C9          00178          RET                       ;zurück
3154 79          00179 GOTPUT   LD      A,C
3155 FED1        00180          CP      AE                ;kleiner als Ä ?
3157 380C        00181          JR      C,LEAVE           ;wenn ja, raus
3159 FEDA        00182          CP      SZ+1              ;größer als ß ?
315B 3008        00183          JR      NC,LEAVE          ;wenn ja, raus
315D FED4        00184          CP      UE+1              ;ist es Ä,Ö,Ü ?
315F 3807        00185          JR      C,CHNG            ;wenn ja, ändern
3161 FED6        00186          CP      SAE               ;ist es ä,ö,ü,ß ?
3163 3003        00187          JR      NC,CHNG           ;dann ändern
3165 BF          00188 LEAVE    CP      A                 ;setze Z-Flag
3166 18DC        00189          JR      FLTPUT            ;zurück
3168 D6D1        00190 CHNG     SUB     AE                ;0 für
316A FE04        00191          CP      4                 ;Kleinbuchstabe ?
316C 3802        00192          JR      C,CHNG1           ;wenn nein, springe
316E D602        00193          SUB     2
3170 6F          00194 CHNG1    LD      L,A
3171 2600        00195          LD      H,0
3173 113B31      00196          LD      DE,TABLE
3174             00197 RX02     EQU     $-2
```

```
MISOSYS EDAS-4.3 01/01/87 00:07:19 UMLAUT-Filter 1.1 1986        Page 00005

3176 19         00198         ADD   HL,DE
3177 7E         00199         LD    A,(HL)
3178 323731     00200         LD    (CHAR),A
3179            00201 RX03    EQU   $-2
317B 0606       00202         LD    B,6               ;einschalten.
317D 213431     00203         LD    HL,GERMAN         ;HL --> Anfang Zeichenkette;
317E            00204 RX04    EQU   $-2
3180 4E         00205 HUPS    LD    C,(HL)            ;Zeichen uebergeben
3181 CD4431     00206         CALL  FLTPUT            ;die Kette hinunter
3182            00207 RX05    EQU   $-2
3184 23         00208         INC   HL                ;und auf US - CHARSET zurück
3185 10F9       00209         DJNZ  HUPS              ;wenn fertig,
3187 4E         00210         LD    C,(HL)            ;letztes Zeichen
3188 18DB       00211         JR    LEAVE             ;dann zum CALLer zurück
318A            00212 FLTEND  EQU   $
0066            00213 FLTLEN  EQU   FLTEND-BEGIN+1
318A 4B31       00214 RELTAB  DW    RX01
318C 7431       00215         DW    RX02
318E 7931       00216         DW    RX03
3190 7E31       00217         DW    RX04
3192 8231       00218         DW    RX05
3194 0000       00219         DW    0
3000            00220         END   BEG
00000 Total errors
```

So ein Mist, diese computergesteuerte Benzinkontrolle!

ZEUS mit 64 *und* 80 Zeichen

1. Das Problem

ZEUS ist in den Kreisen unseres Clubs vielleicht nicht der am häufigsten angewendete Editor-Assembler, weil die Einsteiger eine gewisse Scheu haben, ihn zu benutzen. Nämlich EDTASM wird im DOS-Manual schön erklärt, und da kann man nicht viel falsch machen. Aber ZEUS ist mit Abstand das leistungsfähigste Werkzeug unter den kleinen, handlichen. Das Problem besteht nun aber darin, daß ZEUS zwar ohne weiteres mit 80 Zeichen betrieben werden kann, dann aber gewisse Merkwürdigkeiten zeigt: Das Scrolling auf- oder abwärts geht nur über 15 Zeilen, der Cursor im Editionsmodus steht dann immer ein Kilobyte hinter dem Zeichen, auf das er zeigen soll. Dagegen sind Kräutlein gewachsen, die im folgenden beschrieben werden. Bei der Gelegenheit sollte auch der Lästigkeit begegnet werden, daß die Anzeige der vorigen Zeile nur mit A bzw. mit CTRL-Hochpfeil klappt.

2. Die Anregung

Mir war bekannt, daß Gerald Schröder ZEUS schon einmal näher besehen hatte, und ich wollte von ihm wissen, wo der Edit-Cursor erzeugt wird, um ihm Manieren beibringen zu können (s. o.). Das wußte er auch nicht, aber er gab mir stattdessen ein paar Tips, die die Lawine schließlich ins Rollen brachten: An der Stelle 4108h ist die Anzahl der Bildschirmzeilen abgelegt. Auf dieses Wissen konnte bei der weiteren Untersuchung aufgebaut werden. Und dann lieferte er auch gleich die Lösung für das Hochpfeil-Problem: Zwei Bytes sind in ZEUS zu ändern, mehr nicht. Sie erscheinen im Listing.

3. Der Ansatz

In der Initialisierungsphase liest ZEUS die Stelle 4108h aus und schreibt diesen Wert nach 7751h. Die Anzahl der Scroll-Zeilen ist um 1 niedriger. Der verminderte Wert wird nach 7742h geladen. Jetzt weiß das Programm, wieviele Zeilen beim Scrolling zwischen zwei ENTER-Eingaben abgerollt werden sollen. Eine Umschaltung zwischen 64 und 80 Zeichen, also 15 und 24 Scroll-Zeilen, muß demnach diese beiden Speicherstellen mit den zutreffenden Zahlen beglücken.

Nun mußte ein Weg gefunden werden, ZEUS mitzuteilen, wann er umzuschalten hat. ZEUS erkennt die eingegebenen Kommandos auf eine interessante Weise: Der ASCII-Wert der gedrückten Taste wird zunächst um 41h (Buchstabe A) vermindert. So wird A zu 00h, B zu 01h usw.. Dieser Wert wird verdoppelt. Nach einer weiteren Operation ist er dann zum Zeiger in eine Tabelle geworden, wo die Adressen der zugehörigen Bearbeitungsroutinen stehen. Hier kann man eingreifen:

Im Normalbetrieb ist U das Kommando mit dem höchstzulässigen ASCII-Wert. Nach der Subtraktion von 41h und dem Verdoppeln ist aus V (ASCII 56h) der Wert 2Ah geworden. ZEUS prüft nun, ob ein V (nunmehr 2Ah) oder ein noch höheres Zeichen eingegeben wurde und verzweigt gegf. in eine Fehlerroutine. Zur Umschaltung auf das andere Bildschirmformat kann man nun prüfen, ob *genau* V gedrückt wurde. Wenn ja, soll umgeschaltet werden.

In der Folge erbrachten verschiedene Tests weitere Schwierigkeiten, die aber mehr oder weniger leicht lösbar waren: Wenn ZEUS mit dem Zusatz (,F) aufgerufen wird, unterhält er einen eigenen Druckertreiber. Normalerweise geht er von 80 Zeichen pro Druckzeile aus. Bei der A-Option (assemblieren) ergeben sich aber durch die zusätzliche Angabe der Speicherstelle und des Objektcodes bis zu 96 Zeichen. Es stellte sich heraus, daß der Normalwert in 4103h gespeichert ist, wo er leicht auf 96 (60h) zu ändern war. Ebenfalls bei A wurde im 80-Zeichen-Modus merkwürdigerweise auf 64 Zeichen zurückgeschaltet. Der Grund war ein offenbar überflüssiges PUSH HL an der Stelle 6DD3h. Wenn diese Stelle erreicht wird, steht in HL die Adresse der DOS-Umschaltroutine auf 64 Zeichen. So konnte einfach das PUSH HL in ein NOP verwandelt werden, so daß der spätere RET-Befehl ohne Umweg über das Betriebssystem zu ZEUS zurückkehrt.

ZEUS sollte durch die neuen Features nicht länger werden, um nicht seinen Programmspeicher zu verkürzen. Wohin mit den zusätzlichen Routi-

nen? Hier war eine Kurzsichtigkeit der DOS-Entwickler bei den höheren Genies hilfreich: Die BASIC-Initialisierungen stehen beim alten Genie 1 im ROM, von wo sie nach 4000h ff. verschoben werden. Das G3s macht das noch ganz genauso, obwohl der gesamte "ROM"-Inhalt (das G3s hat da RAM) von der Systemdiskette kommt. Er könnte also ohne weiteres gleich an seine endgültige Bestimmungsadresse geladen werden. Dieser Teil im Adreßraum ab 0674h ist demnach frei, weil er nur beim Booten ausgelesen wird.

Das Genie 3s hat leider mitten in diesem Bereich eine Routine, die nicht überschrieben werden darf. Daher war es hier notwendig, die neuen Unterprogramme in zwei Raten nach 0674h (resid1) und 06D2h (resid2) zu verladen. Für andere Modelle kann man sich das schenken. Wer ein G1/2 oder einen TRS-80 mit einem Banker hat, kann sich dort ebenfalls niederlassen. Ansonsten wird man wohl doch den Programmspeicher beschneiden müssen.

Wo ZEUS beim Umschalten die Informationen über das Bildschirmformat hernimmt, ist aus den Labeldeklarationen zu entnehmen. Das kann je nach Computermodell sehr unterschiedlich sein. Der geneigte Leser muß sich informieren, wie das bei seiner eigenen Maschine gelöst ist. Auch die Ausgabe der Codes 10h für 64 und 13h für 80 Zeichen genügt möglicherweise nicht. Hier kann ich aus naheliegenden Gründen aber nur auf das Genie 3s eingehen.

4. Die Lösung

Wenn Patches zur Modifikation in das Programm eingebracht werden, überschreibt man damit andere Codes. Es ist sinnvoll, immer Befehle zu ändern, die genau drei Bytes lang sind. So läßt sich dann ein CALL von dieser Länge zappen. Die überschriebenen Befehle müssen dann im neuen Unterprogramm selbstverständlich - und sei es in geänderter Form - nachgeholt werden. Alle hier eingebrachten Teile arbeiten mehr oder weniger stur nach diesem Strickmuster. Sie sehen im einzelnen folgendermaßen aus:

Zunächst müssen die neuen Programmteile nach dem Einsprung in ZEUS an ihre Bestimmungsadresse geladen werden. Das wäre normalerweise möglich, indem man ihnen kurzerhand ein ORG an der endgültigen Stelle verpaßt. Das G3s kann jedoch den Adreßbereich bis 3000h schreibschützen. Sollte das gerade der Fall sein, käme die Fehlermeldung "Ladeversuch auf ROM-Speicherplatz". Daher stehen die neuen Unterprogramme zunächst im Sektorpuffer (ORG 4200h), wo sie nichts anrichten und wohin sie beim Laden des Programms ohnehin zuerst geschrieben werden. Das Programmsegment init hebt nun den Schreibschutz auf und verlädt den neuen Code. Ganz zu Anfang aber holt er aus der Speicherstelle 37F6h die gerade eingestellte Zeilenzahl pro Bildschirmseite und patcht sie nach 4108h, von wo aus ZEUS dann das weitere selbst veranlaßt. Schließlich springt init nach 5312h, dem ursprünglichen Entrypoint.

Das UP toggvid holt zunächst die beiden Befehle ADD A,A (verdoppele den Akku) und CP 2Ah (prüfe auf V) nach. Wurde kein V eingegeben, macht ZEUS die weitere Bearbeitung (RET NZ). Ansonsten aber wird aus 37F6h die Anzahl der Zeilen pro Seite ausgelesen und in die jeweils andere Zeilenzahl umgerechnet. Sie landet schließlich in 7751h (s. o.) und, um 1 vermindert, in 7742h. Je nach dem sich neu ergebenden Format wird der Akku mit dem entsprechenden Steuercode geladen, der nun über 0033h die Umschaltung des Bildschirms vornimmt. ZEUS unterhält einen Eingabepuffer, in dem nun das Kommando V noch immer steht. HL zeigt darauf. Da V ja verboten ist, wird zur Vermeidung einer Fehlermeldung der Code 01h dort eingeschrieben. Das ist so, als wäre die BREAK-Taste gedrückt worden. Es erscheint deshalb nunmehr nur das Prompt-Zeichen. Die Umschaltung der Bildschirmformate war also recht einfach.

Komplizierter ist die Lösung des Problems, daß der Edit-Cursor bei 80 Zeichen verrückt spielte. Er wird nicht über den Bildschirm-DCB angezeigt, sondern direkt in den Screen geladen. Wo er wirklich hingehört, muß ZEUS zunächst mitgeteilt werden. In 4020h im Video-DCB ist die aktuelle Cursorposition abgelegt. Nur leider ist sie falsch, weil das G3s ein Kind des G3 ist, wo der Bildschirm von 3C00h-43FFh geht. Im G3s ist es der Raum 3800h-3FFFh. Die untere Bildschirmhälfte liegt demnach parallel zur Tastatur. Die Cursoradresse in 4020h gilt relativ zu 3C00h, unabhängig vom tatsächlichen Beginn des Bildschirms. Der Abstand der Cursoradresse von 3800h ist in 37FCh abgelegt. Aus den Informationen dieser Zeiger läßt sich nun die zutreffende Bildschirmstelle errechnen. Wenn das gelaufen ist, muß natürlich das Zeichen an der Cursorposition gerettet werden, denn der Cursor wird es gleich überschreiben. Das bedeutet, zunächst die Tastatur auszuknipsen, um die untere Bildschirmhälfte zu erreichen. Im Betriebssystem des G3s gibt es dafür eine fix und fertige Routine an 3643h (Label getvchr). Mit anderen Computern wird man sich die Mühe selbst machen müssen, das Zeichen unter dem Cursor zu laden.

Um nun im Edit-Modus wahlweise den Cursor oder das ASCII-Zeichen blinkenderweise in den Bildschirm zu schreiben, müssen die Unterprogramme dispchr und restore ähnlich umständlich vorgehen. Wie das im einzelnen geschieht, möchte ich hier nicht erklären, denn dazu wäre eine ausführliche Beschreibung der ursprünglichen ZEUS-Methode notwendig. Nur so viel: Für alle Computermodelle können die beiden Unterprogramme einfach übernommen werden. Nur der CALL nach putvchr (3649h) muß u. U. angepaßt werden. putvchr macht das Gegenteil von getvchr: Das Zeichen (Cursor oder ASCII) muß in den Bildschirm geladen werden. Dazu muß wieder die Tastatur verschwinden (oder beim G3 der RAM-Bereich 4000h-43FFh).

Letztenendes klappte alles. Ein Schönheitsfehler trat dabei aber auf. Wenn man nun Shift-Rechtspfeil drückte, um die letzte Programmzeile zu listen, erschien die vorletzte. Wie man das geradebiegt, habe ich nicht herausgefunden. Eine andere Lösung haute hin: Jetzt ist es Shift-Abwärtspfeil, und der funktioniert (s. Listing).

5. Der Nutzen

Das Listing enthält recht lange Kommentare. Klar, die Zeilenlänge hat sich ja um 16 Zeichen vergrößert. So kann man sich nun ausführlicher über den Zweck eines jeden Befehls ausquatschen, um auch für den Einsteiger verständlicher zu sein. Aber auch der Vollblut-Crack in seiner einsamen Klause hat etwas davon, wenn seine Steno-Kommentare locker auf 64 Zeichen passen: Beim Assemblieren kommen die Adressen und Hexcodes mit auf den Bildschirm, was zur Folge hat, daß nun doch wieder eine Programmzeile auf zwei Bildschirmzeilen geschrieben wird. Es wird dabei unübersichtlich. Wenn man nun mit 64 Zeichen programmiert und mit 80 Zeichen assembliert, sieht es auf dem Screen so aus wie auf dem abgedruckten Listing. Schnuckelig, oder?

6. Die Implementation

Wer seinen ZEUS in der beschriebenen Weise verändern möchte, kann folgendermaßen vorgehen: Vom DOS aus LOAD,ZEUS/CMD eingeben und dann das gelistete Programm aufrufen. Der bereits geladene ZEUS wird dadurch gepatcht und läuft in der neuen Version. Wenn alles zur Zufriedenheit klappt, sollen die Neuigkeiten natürlich fest in ZEUS/CMD implementiert werden. Wie der letzte Sektor (rel. Skt. 27h) dann auszusehen hat, ist dem Dump zu entnehmen. Die übrigen Patches folgen hier. Wer sie für einen anderen Computer als das Genie 3s einbringen will, muß selbstverständlich die entsprechenden Anpassungen vornehmen.

rel. Skt.	rel. Byte	alter Code	neuer Code	(alle Angaben in Hex)
00	87	40	60	Druckzeilenlänge
1D	3C	E5	00	PUSH HL gelöscht
1F	06 - 09	2A 20 40 5E	CD D2 06 5F	Cursoradr. errechnen
1F	0C - 0E	CB 0A 36	CD 74 06	Edit-Cursor anzeigen
1F	17 - 19	28 F1 73	CD 87 06	Video-Zeichen restaur.
25	32	10	40	für Aufwärtspfeil
25	72	5B	0B	für Aufwärtspfeil
25	82	19	1A	Shift-Abwärtspfeil
26	9B - 9D	87 FE 2A	CD E1 06	Video-Umschaltung

Arnulf Sopp

```
            ;*****************************************************************
            ;
            ;       Verschiedene Patches in ZEUS/CMD zur Anpassung an das Genie 3s
            ;       von The HACKTORY Arnulf Sopp & Gerald Schröder, (C) 1986
            ;
            ;       --)    Hochpfeil jetzt ohne CTRL-Taste wirksam        Schröder
            ;       --)    Umschaltung zwischen 80 und 64 Zeichen mit V   Sopp
            ;
            ;       Kopie und Anpassung an andere Computer ausdrücklich erwünscht!
            ;
            ;*****************************************************************

            ;Symbolvereinbarungen für Speicherstellen außerhalb ZEUS (evtl. anpassen)
0033        outchr  EQU   0033h   ;Zeichen aus dem Akku über den Video-DCB anzeigen
0674        resid1  EQU   0674h   ;Ladeadr. der Unterprg., dort urspr. Boot-Routine
06D2        resid2  EQU   06d2h   ;nächste Ladeadresse, dort urspr. BASIC-Initial.
3643        getvchr EQU   3643h   ;Zeichen aus dem Großbildschirm in den Akku laden
3649        putvchr EQU   3649h   ;Zeichen aus dem Akku in den Großbildschirm laden
37F6        linppag EQU   37f6h   ;hier ist die Anzahl d. Zeilen pro Seite abgelegt
37FC        vidoffs EQU   37fch   ;dort MSB Bildschirmoffset von 3800 an gerechnet
4D20        crsaddr EQU   4D20h   ;dort im Video-DCB die aktuelle Cursorposition
                                  ;(rel. Adresse im Bildschirm + 3C00, nicht 3800!)

            ;versch. Patches innerhalb des Programms zum Aufruf der neuen Features
4103                ORG   4103h             ;hier ist d. Drucker-Zeilenlänge abgelegt
4103 60             DB    60h               ;jetzt sollen 96 Zeichen Normalwert sein

6DD3                ORG   6dd3h             ;hier steht ursprünglich PUSH HL
6DD3 00             NOP                     ;jetzt nicht mehr, es war überflüssig

6F95                ORG   6f95h             ;und hier LD HL,(4D20) (Cursoradresse)
6F95 CDD206         CALL  curspos-offset2   ;stattdessen Adresse errechnen lassen
6F98 5F             LD    E,A               ;Zeichen an der Cursorposition sichern

6F9B                ORG   6f9bh             ;hier steht RRC D - LD (HL),n
6F9B CD7406         CALL  dispchr-offset1   ;durch eigene Anzeigeroutine ersetzen

6FA6                ORG   6fa6h             ;dort steht JR Z,6F99 - LD (HL),E
6FA6 CD8706         CALL  restore-offset1   ;eigene Routine zum Restaurieren des Z.

75A9                ORG   75a9h             ;dort steht der Operand von BIT 6,B
75A9 40             DB    40h               ;jetzt BIT 0,B (für Aufwärtspfeil)

75E9                ORG   75e9h             ;und hier der Operand des Befehls CP 5Bh
75E9 0B             DB    0bh               ;jetzt heißt er CP 0Bh (Aufwärtspfeil)

75F9                ORG   75f9h             ;hier 19 für CP 19 (Shift-Rechtspfeil)
75F9 1A             DB    1ah               ;nun aber Check auf Shift-Abwärtspfeil

770A                ORG   770ah             ;hier stand ADD A,A - CP 2Ah
770A CDE106         CALL  toggvid-offset2   ;daraus wird jetzt CALL Umschaltung

            ;Unterprogramme an Adressen, die nur beim Booten belegt werden
            ;und daher für diesen Zweck frei sind:

            ;im Edit-Modus einen Cursor bzw. das Zeichen unter dem Cursor anzeigen
4200                ORG   4200h             ;im Sektorpuffer, wo es nicht stört
3B8C        offset1 EQU   $-resid1          ;Offset für Unterprg. dispchr und restore
4200 E3     dispchr EX    (SP),HL           ;RET-Adresse, dort steht d. Cursorzeichen
4201 7E             LD    A,(HL)            ;dieses in den Akku laden
4202 21A26F         LD    HL,6fa2h          ;neue RET-Adresse ein paar Bytes später
4205 E3             EX    (SP),HL           ;auf den Stack zurücklegen
4206 CD4936         CALL  putvchr           ;Zeichen anzeigen, auch im Bereich 3800
4209 CB0A           RRC   D                 ;zwischen Cursor und Zeichen abwechseln
420B D8             RET   C                 ;falls der Cursor stehenbleiben soll
420C F5     onvid   PUSH  AF                ;sonst eventuelle Tastatureingabe retten
420D 7B             LD    A,E               ;und das Zeichen aus E in den Akku laden
420E CD4936         CALL  putvchr           ;dieses (notfalls ab 3800) anzeigen
4211 F1             POP   AF                ;Tastatureingabe restaurieren
4212 C9             RET                     ;und zurück

            ;beim Blinken und nach der Edition das Zeichen unter dem Cursor restaur.
4213 20F7   restore JR    NZ,onvid          ;ggf. altes Bildschirmzeichen restaur.
4215 F1             POP   AF                ;sonst CALL-Ebene vom Stack löschen
4216 C3996F         JP    6f99h             ;überschriebenen Sprungbefehl nachholen

            ;Position d. Zeichens unter dem Cursor auf dem 80-Zeichen-Video errechnen
3B47        offset2 EQU   $-resid2          ;Lade-/Arbeitsoffset d. übrigen Programme
4219 2A204D curspos LD    HL,(crsaddr)      ;Puffer im DCB für d. akt. Cursorposition
421C 3AFC37         LD    A,(vidoffs)       ;MSB des Offsets vom physik. Video-Anfang
421F EE04           XOR   04h               ;als Subtrahenden zu 3C00 anpassen
4221 94             SUB   H                 ;MSB der relativen Cursoradresse errech-
4222 ED44           NEG                     ;nen
4224 67             LD    H,A               ;und neu in den Zeiger laden
4225 C34336         JP    getvchr           ;A (- Byte an der Cursorposition, zurück

            ;das Kommando V erkennen und ggf. zwischen 64 und 80 Zeichen umschalten
4228 87     toggvid ADD   A,A               ;überschriebener Code, Akku verdoppeln
4229 FE2A           CP    2ah               ;wurde V eingegeben?
422B C0             RET   NZ                ;falls nein
422C 3AF637         LD    A,(linppag)       ;dort steht die Anzahl der Zeilen / Seite
422F EE09           XOR   09h               ;16 in 25 umwandeln und umgekehrt
4231 325177         LD    (7751h),A         ;ZEUS die neue Anzahl verraten
4234 3D             DEC   A                 ;tatsächliche Anzahl der Scroll-Zeilen
4235 324277         LD    (7742h),A         ;dort patchen (15 bzw. 24 Scroll-Zeilen)
4238 E6F0           AND   0f0h              ;Resultat 15 oder 24 Zeilen / Seite?
423A 3E10           LD    A,10h             ;Steuercode für 16*64 Zeichen (über DCB)
423C 2802           JR    Z,switch          ;falls 80 Zeichen (24 Scroll-Zeilen)
423E F603           OR    03h               ;13h = Steuercode für 25*80 Zeichen
4240 D5     switch  PUSH  DE                ;retten, wird bei CALL 0033 verändert
4241 CD3300         CALL  outchr            ;Code ausgeben (= Bildschirm umschalten)
4244 D1             POP   DE                ;Register wieder restaurieren
4245 3601           LD    (HL),01h          ;BREAK-Taste simulieren (V löschen)
4247 C9             RET                     ;das war alles

            ;ZEUS beim Einsprung auf das aktuelle Bildschirmformat einstellen,
            ;zusätzliche Unterprogramme an ihre Bestimmungsadresse übertragen
4248 3AF637 init    LD    A,(linppag)       ;hier steht die Anzahl der Zeilen / Seite
424B 320841         LD    (4108h),A         ;ZEUS die Anzahl kund und zu wissen geben
424E E5             PUSH  HL                ;Adresse des Aufrufbefehls retten
424F 210042         LD    HL,dispchr        ;Anfangsadresse von dispchr und restore
4252 117406         LD    DE,resid1         ;spätere Ladeadresse, an der sie arbeiten
4255 011900         LD    BC,curspos-dispchr      ;Länge beider Unterprogramme
4258 DBFA           IN    A,(0fah)          ;Zustand des Systembytes 1 des Genie 3s
425A F5             PUSH  AF                ;retten
425B E6DF           AND   0dfh              ;evtl. Schreibschutz 0000-2FFF aufheben
425D D3FA           OUT   (0fah),A          ;Systembyte ohne Schreibschutz ausgeben
425F EDB0           LDIR                    ;Unterprogramme an d. Ladeadresse übertr.
4261 1ED2           LD    E,resid2&0ffh     ;spätere Ladeadr. der übrigen Programme
4263 0E2F           LD    C,init-curspos    ;Länge der restlichen Unterprogramme
4265 EDB0           LDIR                    ;an die Ladeadresse übertragen
4267 F1             POP   AF                ;alter Zustand des Systembytes 1
4268 D3FA           OUT   (0fah),A          ;restaurieren
426A E1             POP   HL                ;Adresse des ZEUS-Aufrufbefehls
426B C31253         JP    5312h             ;zum ehemaligen Entrypoint von ZEUS

4248                END   init              ;hier neuer Einsprung in ZEUS
```

```
DRV 00   0508 ED52 38DE EB3E 0032 6E74 B728 06CD ...R8..).2nt.(..
0   10   3300 3E1E 113E 0DCD 6076 1809 32E0 59CD 3.)..)..`v..2.Y.
0H  20   D86D D942 4BC5 CD73 6908 22D9 77CD C86C .m.BK..si.".w..l
    30   C10B 78B1 C8CD F46B C008 3A6E 74B7 2808 ..x....k..:nt.(.
DRS 40   3A20 40FE 0032 B377 C5F4 DF6E F299 772A :.§..2.w...n..w*
569 50   3C77 2B22 3C77 11C0 FF2A 2040 1922 2040 <w+"<w...*.§.".§
239H60   193E 1ECD 3300 2220 4021 0000 18BF 0170 .)..3.".§!.....p
    70   0042 E37E 21A2 6FE3 CD49 36CB 0AD8 F57B .B.ß!.o..I6....ä
    80   CD49 36F1 C920 F7F1 C399 6F2A 2040 3AFC .I6.......o*.§:.
    90   37EE 0494 ED44 67C3 4336 87FE 2AC0 3AF6 7....Dg.C6..*.:.
    A0   37EE 0932 5177 3D32 4277 E6F0 3E10 2802 7..2Qw=2Bw..).(.
    B0   F603 D5CD 3300 D136 01C9 3AF6 3732 0841 ....3..6..:.72.A
FRS C0   E521 0042 1174 0601 1900 DBFA F5E6 DFD3 .!.B.t..........
39  D0   FAED B01E D20E 2FED B0F1 D3FA E1C3 1253 ......./........S
27H E0   0202 4842 0000 0000 0000 0000 0000 0000 ..HB............
    F0   0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Schöner listen mit ZEUS

Ein neuer Druckertreiber für Ästheten

Anläßlich der Arbeit an der 80-Zeichen-Darstellung für ZEUS fiel natürlich so manches an Erkenntnissen ab, was für die 80 Zeichen keine Rolle spielte. So kam z. B. ans Licht, daß im ganzen Programm nur an einer einzigen Stelle der Drucker angesprochen wird. Es ist die Adresse 59E4h, wo der Befehl CALL NZ,003Bh steht (die NZ-Bedingung ist dann erfüllt, wenn das H-Kommando eingegeben wurde oder das A-Kommando mit der H-Option). Wenn es ZEUS dem Zapper so leicht macht, kann er nicht widerstehen:

Es sollte ermöglicht werden, beliebige Druckersteuerungen zumindest in die Kommentare einzubeziehen, um das Schriftbild zu modifizieren. Möglich war das schon immer, wenn man auf der Diskette im Quelltext mit DDE die nötigen Codes zapte. Zu diesem Zweck mußten beim Programmieren Dummy-Zeichen im Text vorgesehen werden, die man dann mit den Druckersteuerungen überschreiben konnte. Aber das ist umständlich.

Die hier vorgestellte Alternative geht von folgender Philosophie aus: Im Alltag muß man die neuen Möglichkeiten als Spielkram ansehen, so daß es sich aus Gründen der Ernsthaftigkeit verbot, auch nur ein Byte in ZEUS/CMD dafür zu opfern. Stattdessen gibt es ein kleines Programm, das zunächst ZEUS lädt, ihn mit der Umleitung auf seine eigene Druckerroutine verändert und dann anspringt. Es ist so programmiert, daß auch der Aufruf des Assemblers mit dem Zusatz (,F) möglich ist (Aktivierung eines ZEUS-eigenen Druckertreibers, der aber an unserem hier nicht vorbeikommt).

Die Steuercodes für den Printer sollten in Hex einzugeben sein, denn andernfalls wäre man auf die Zeichen angewiesen, die die Tastatur bietet. Das äußerst wichtige ESC (escape, 1Bh, 27d) und viele andere wären sonst nicht über die Tastatur erreichbar. So kam es nun darauf an, ZEUS zu signalisieren, wo Hexcodes beginnen und aufhören. Es bot sich an, den Code 7Fh (127d) zu nehmen, denn der bedeutet für den Drucker DEL (delete, lösche das vorangegangene Zeichen). Niemand wird so verrückt sein, diesen Code in einem Listing zu verwenden, denn dann hätte er sich das vorangegangene Zeichen gleich sparen können. Er ist also frei verfügbar. Deshalb geht die Druckerumleitung folgendermaßen vor:

Am Label devia wird geprüft, ob das auszugebende Zeichen dieses DEL ist. Falls nein, wird einfach nach 003Bh weitergesprungen, wo ZEUS in der Originalversion ohnehin hinwollte. Ist es aber DEL, folgen zunächst ein paar Vorbereitungen. Da das nächste Auftauchen von 7Fh als Signal für das Ende der Hex-Sequenz dient, werden die Adressen der Umleitung und der System-Druckerroutine vertauscht. Jetzt springt jedes Zeichen ungleich DEL immer in das Unterprogramm prthex, wo es als eine Hexziffer behandelt wird. Wenn wieder DEL erscheint, wird alles in den alten Zustand versetzt.

prthex sorgt zunächst dafür, daß nur Großbuchstaben umzurechnen sind. jetzt wird in der üblichen Weise aus einem ASCII-Zeichen eine binäre Hexziffer gemacht. Anschließend muß herausgefunden werden, ob bereits zwei Ziffern, also ein komplettes Hex-Byte, eingelesen sind. Das geht gut mit einem ständigen XOR FFh des Zählbytes counter. Dabei wird es jedesmal entweder zu FFh oder zu 00h. Die Z-Bedingung ist also erfüllt oder nicht. Wenn nein, dann handelt es sich um die erste Ziffer, die als höherwertige vom unteren in das obere Halbbyte rotiert werden muß. Sie wird danach in dem Puffer prtcode abgelegt. Ist aber bereits die niederwertige Ziffer errechnet, dann geht es bei prtout weiter, wo beide Ziffern miteinander verknüpft werden.

Das Resultat ist das Byte, das als Steuercode auf den Drucker ausgegeben werden soll. Die Ausgabe erfolgt hier über die unübliche Adresse 05B4h. Dort sind alle Codes zwischen 0 und 255 erlaubt. Das Unterprogramm an 003Bh würde nur durchlassen, was ihm beliebt. 05B4h führt auch automatisch den Test auf die Druckerbereitschaft durch.

Die Ladeadresse des Programms (ORG) ist hier 3000h. Sie ist nicht auf allen Tandy-Kompatiblen zu realisieren. Gegf. muß sich der User eine andere Adresse suchen. Wer ein späteres Modell aus unserer Computerfamilie, einen fähigen Banker oder Helmut Bernhardts "sicheres Plätzchen" hat, ist mal wieder fein raus.

Um die Möglichkeiten des Programms zu demonstrieren, habe ich ein bißchen mit Proportional-, Kursiv- und gespreizter Schrift herumgespielt. Solch ein bunter Papagei von einem Listing läßt natürlich Zweifel aufkommen, ob der Programmierer noch alle Tassen im Schrank hat. Weniger verrückt kann es aber sehr sinnvoll sein. Wie die ersten 6 Zeilen auf dem Bildschirm aussehen, ist anschließend wiedergegeben. Es ist hier nicht von Interesse, was die einzelnen Codes bei meinem NEC P6 bewirken. Da muß jeder das Manual seines eigenen Druckers befragen.

Arnulf Sopp

HEFT
18
Februar
1987

```
;         ▒1b70010e1b▒4Schöner listen▒1b70001b▒5
;         ▒1b7001▒Ein neuer Druckertreiber für ZEUS/CMD▒1b7000▒

          ORG     3000h               ;▒0E▒hier ist Platz

prdevia   CP      7fh                 ;▒1b▒4DEL▒1b▒5, Einleitung Hex-Sequenz?
```

Schöner listen

Ein neuer Druckertreiber für ZEUS/CMD

```
                          ;
                          ;

3000                      ORG   3000h          ;hier ist Platz

3000  FE7F      prdevia   CP    7fh            ;DEL, Einleitung Hex-Sequenz?
3002  C23B00              JP    NZ,003bh       ;Zeichen drucken, falls nein
3003            prtrout   EQU   $-2            ;(Sprungadresse variabel)
3005  E5                  PUSH  HL             ;retten
3006  D5                  PUSH  DE
3007  2A0330              LD    HL,(prtrout)   ;Sprungadresse des Drucker-CALLs
300A  ED5B4130            LD    DE,(calladr)   ;und seiner Umleitung
300E  224130              LD    (calladr),HL   ;bei jedem Auftreten von DEL
3011  ED530330            LD    (prtrout),DE   ;beide Adressen vertauschen
3015  D1                  POP   DE             ;restaurieren
3016  E1                  POP   HL
3017  C9                  RET                  ;zurück, ohne 7F auszudrucken

3018  CDB545    prthex    CALL  45b5h          ;Klein- in Großbuchstaben umwandeln
301B  FE3A                CP    3ah            ;Dezimalziffer?
301D  3802                JR    C,ciphok       ;falls ja
301F  D607                SUB   07h            ;Hexziffern angleichen
3021  E60F      ciphok    AND   0fh            ;ASCII-Ziffer binär umwandeln
3023  F5                  PUSH  AF             ;Hex-Digit retten
3024  3A4330              LD    A,(counter)    ;Zähler 00 oder FF (2 Digits)
3027  EEFF                XOR   0ffh           ;Argument, um zw. 00 und FF zu wechseln
3029  324330              LD    (counter),A    ;neu patchen
302C  2809                JR    Z,prtout       ;falls schon zwei Digits gelesen
302E  F1                  POP   AF             ;Hex-Digit restaurieren
302F  07                  RLCA                 ;unteres in das obere Nibble rotieren
3030  07                  RLCA
3031  07                  RLCA
3032  07                  RLCA
3033  324430              LD    (prtcode),A    ;höherwertige Ziffer dort patchen
3036  C9                  RET                  ;zunächst zurück, ohne zu drucken

3037  F1        prtout    POP   AF             ;Hex-Digit restaurieren
3038  E5                  PUSH  HL             ;retten
3039  214430              LD    HL,prtcode     ;höheres Digit des zu druckenden Codes
303C  B6                  OR    (HL)           ;beide Hexziffern verknüpfen
303D  E1                  POP   HL
303E  C3B405              JP    05b4h          ;Code ausdrucken und zurück

3041  1830      calladr   DW    prthex         ;Drucker-Umleitung
3043  00        counter   DB    00h            ;Ziffernzähler bis 2
3044  00        prtcode   DB    00h            ;Puffer für MSB des Druckzeichens

3045  E5        start     PUSH  HL             ;Adresse des Aufrufbefehls retten
3046  115830              LD    DE,fcb         ;Zeiger auf FCB mit Filenamen
3049  CD3044              CALL  4430h          ;ZEUS laden
304C  C20944              JP    NZ,4409h       ;raus, falls Fehler aufgetreten
304F  110030              LD    DE,prdevia     ;Adresse der Druckerumleitung
3052  ED53E559            LD    (59e5h),DE     ;in ZEUS patchen
3056  E3                  EX    (SP),HL        ;HL (- Aufrufadr., Stack (- Entry ZEUS
3057  C9                  RET                  ;ZEUS anspringen

3058  5A        fcb       DM    'ZEUS/CMD',0dh ;Dateiname

3045                      END   start          ;hier Einsprung

00000 Fehler

calladr 3041   ciphok 3021   counter 3043   fcb   3058   prdevia 3000   prtcode 3044
prthex  3018   prtout 3037   prtrout 3003   start 3045
```


CP/Mac – EINE BRÜCKE ZWISCHEN GDOS/NEWDOS UND CP/M = EIN NEUER COMPUTER!?

Unter den Kleinanzeigen in MC 12/86 konnte man, wenn man gut hinschaute und es nicht übersah, die Worte GDOS/NEWDOS aufblitzen sehen. Da wir Oldtimer mit unseren fast schon als Exoten geltenden GENIEs mit Angeboten für unsere Maschinen nicht mehr allzu sehr verwöhnt werden, ist der aufmerksame Leser sofort alarmiert und liest weiter: „Dank CP/Mac haben Sie mit GDOS/NEWDOS vollen Zugriff auf Ihre CP/M-Disketten!". Da denke ich an meine in der Schublade kaum benutzt herumliegende CP/M-Systemdiskette, die zwar mit einem imponierenden Graphik-Logo „CP/M 2.2a für GENIE III" startet, aber deren Tastenbelegung leider für die Umlaute und ß nicht stimmt (und außerdem erwartet sie, daß man in Englisch absolut perfekt ist!), so daß die Programme damit eigentlich keinen praktischen Gebrauchswert haben. Die läppischen 3,20 DM für das Vorab-Handbuch sollte man eigentlich doch mal springen lassen! Gedacht – getan. Wenig später trudelt ein über 40 Seiten starkes „Bedienungshandbuch zur Version 3.3 für CP/Mac, Hilfsprogramm zur Bearbeitung von CP/M-Disketten unter GDOS 2.2x und NEWDOS/80.2 auf den Geräten Tandy TRS-80 Modell I, EACA GENIE I, II und GENIE III" ein, das neugierig macht. Der sehr mäßige Preis von 75,- DM animiert zur Bestellung. Das Risiko kann allzu groß nicht sein.

Eine Woche später kommt die Diskette mit Programm und Hilfsprogrammen – und zwei Nächte am Computer sind unumgänglich, so faszinierend ist dieses phantastische „Hilfsprogramm"! Wenn ich auch vorher schon mit dem (bei einem damaligen Dollarkurs von DM 3,50/$ – Verzeihung! – sauteuren) HyperSuperCrossPlus-Programm aus USA PC- und CP/M-Disketten mit 170 verschiedenen Formaten formatieren und kopieren und das Inhaltsverzeichnis lesen konnte, aber so etwas hatte ich nicht erwartet. Dieses CP/Mac kann zwar nicht formatieren, aber sonst noch viel mehr!

Man bootet seine normale GDOS/NEWDOS-Systemdiskette und ruft CP/Mac auf. Es meldet sich mit folgendem „Kurzmenü" der Befehle, das man sich während der Arbeit mit einem einfachen Druck auf <ENTER> jederzeit wieder auf den Bildschirm holen kann:

```
CP/Mac                     CP34091207  Pdrive: CPM22X
                           Rel. 3.3 (C) 1986 dnh, Essen

LOG : Diskette einloggen (1)            CCG : Kopie CP/M -> GDOS-Datei
CGC : Kopie GDOS -> CP/M-Datei          CDB : Kopie CP/M-Diskblöcke
CDR : Kopie CP/M-Diskrecords            USR : Zuordnung Userbereich
DFB : CP/M-Datei editieren              DDB : CP/M-Diskblock editieren
DDR : CP/M-Diskrecord editieren         PDR : Pdrivesatz (CPM22X   )
FRE : Freien Disk-Raum anzeigen         FIX : CPMAC sichern
ERA : CP/M-Datei(en) löschen            DIR : CP/M-Inhaltsverzeichnis
END : CPMAC beenden.                    TYP : CP/M-Datei auflisten
DFE : CP/M-DIR-Eintrag editieren        REN : CP/M-Datei umbenennen
HLP : Hilfe anfordern                   SYS : System-Offset (       )
ILF : löschen/erweitern (0)             BAT : Blockbelegungstabelle

CP/Mac Ready:_
```

Bildschirmausdruck nach dem Aufruf von CP/Mac mit dem Kurzmenü der Befehle
Man kann es während der Arbeit mit <ENTER> jederzeit auf den Bildschirm bringen

Die „Betriebsebene" von CP/Mac ist eine Art Zwischenstation zwischen GDOS/NEWDOS und CP/M: Einerseits kann man die vertrauten GDOS/NEWDOS-Befehle wie I/DIR, LIB, LIST, LOAD usw. aufrufen und benutzen (alle MiniDOS-Befehle stehen weiter zur Verfügung), und andrerseits - es ist wie im Traum! - greift man wie selbstverständlich auf die CP/M-Diskette im Laufwerk 1 (das nun nach CP/M-Art B heißt) mit Drei-Tasten-Befehlen wie DIR (zur Unterscheidung vom GDOS/NEWDOS-DIR und gewissermaßen als Abschiedsgruß, noch einmal von dem Parameter 1 gefolgt), FRE, PDR, FIX, USR, ERA, REN usw. zu.

Nach kurzer Eingewöhnung bemerkt man, daß die CP/Mac-Befehle teilweise noch besser sind als die des Original-CP/M: So bleibt beim Listen von Dateien mit TYP (gegenüber TYPE im CP/M) das Bildschirmlisting jeweils am unteren Rand des Monitors stehen - funktioniert immer, ob man bei 64 x 16- oder 80 x 24-Zeilen-Einstellung arbeitet. TYPE in CP/M kann das nicht. Auch einen Disk-Daten-Editor (ähnlich dem DDE - „unserem" aufgemotzten sogar - oder Superzap) mit allem Änderungskomfort in ASCII und HEX besitzt Original-CP/M nicht, aber CP/Mac. Mit datei-, sektor-, CP/M-block- oder diskweisem Zugriff auf die ganze Diskette: man kann CP/M unter GDOS/NEWDOS direkt bearbeiten (das geht bei den sonst sehr komfortablen teuren USA-Programmen nicht).

Dazu kommen Kopierbefehle von GDOS/NEWDOS nach CP/M und von CP/M nach GDOS/NEWDOS mit fast allem von GDOS/NEWDOS gewohnten Komfort, dateiweise und nach IDL-Liste, und „Wildcards" nach CP/M-Manier (bei der Richtung CP/M —> GDOS/NEWDOS). Je nach Richtung wird die Extension dabei automatisch mit der richtigen Trennmarke („/" oder „.") versehen, wenn man den Dateinamen beibehalten will. In diesem Fall wird mit Hilfe des ILF-Befehles bei der Richtung CP/M —> GDOS/NEWDOS sogar für das Rückkopieren —> CP/M gleich die Dateiliste mit erstellt (oder, je nach eingegebenem Parameter, an eine vorhandene /ILF-Liste angehängt).

Noch ein besonderer Gag: Wenn man dem CP/Mac-Befehl einen „/" voransetzt, erhält man die Ausgabe auf dem Drucker. So kann man sich, wenn man will, mit „/HLP,<befehl>" eine Kurzfassung des Handbuches ausdrucken lassen, je mit einem Ausführungsbeispiel. Setzt man dem Befehl ein „?" voran, so erfolgt ggf. eine Abfrage, z. B. beim Kopieren (etwa entsprechend dem FRAG/CFWO-Parameter in GDOS/NEWDOS). Ein vorgesetztes „!" bewirkt, daß Benutzerrückfragen unterdrückt werden (analog KDWA/NDMW bei GDOS/NEWDOS). Auf diese Weise kann man Dateien und Programme (in BASIC bzw. soweit sie nicht auf das eine oder andere Betriebssystem zugreifen) beliebig zwischen den Betriebssystemen GDOS/NEWDOS und CP/M transferieren. Natürlich ist CP/Mac kein CP/M-Emulator, also kein Hilfsprogramm, um CP/M-Software unter GDOS/NEWDOS <u>ablaufen</u> zu lassen. Aber CP/Mac ist ein CP/M-„Access"Programm, welches uns GDOS/NEWDOS-Leuten von unserem gewohnten Betriebssystem her Zugriff auf CP/M-Dateien und Kopiermöglichkeiten eröffnet, um unseren Dateien die CP/M-Welt mit WORDSTAR, dBASE, Turbo Pascal zu öffnen.

Wenn man zwischendrin meint, man hätte einen völlig anderen Computer unter den Fingern (dieser Eindruck stellt sich wirklich bald ein!), kann man dank MiniDOS schnell mal mit <123> in den Speicher sehen und sich davon überzeugen, daß trotz CP/M-Diskette im Laufwerk 1 - Verzeihung: B! - GDOS/NEWDOS noch da ist. Mit END ist man dann plötzlich von der „Fahrt ins CP/M-Land" - wie in einem Science-Fiction-Film - wieder daheim im altvertrauten GDOS/NEWDOS, das man ja nie verlassen hat.

Erwähnenswert sind auch einige gekonnt gebaute Utilities: PDEDIT zum Anpassen des CP/Mac auf weitere, nicht in den zahlreichen fertigen Datensätzen schon definierte CP/M-Formate, sowie COMER/BAS, um z.B. aus einem unter GDOS/NEWDOS erzeugten

Assemblerprogramm die für GDOS/NEWDOS notwendigen Ladeinformationen (01 <Satzlänge> <Ladeadresse>) für den Betrieb unter CP/M zu entfernen. CMDER/BAS tut das umgekehrte: reine Datenfiles (als die auch .COM-Programme unter CP/M geschrieben werden), werden in das GDOS/NEWDOS-Format umgesetzt (einige Disassembler unter GDOS/NEWDOS verlangen für die Quelldatei zwingend das /CMD-Format). CRLF/BAS fügt den für GDOS/NEWDOS-Dateien (im ASCII-Format gespeicherten BASIC-Programmen zum Beispiel) ausreichenden CR-Symbol 0Dh das für CP/M notwendige LF-Symbol 0Ah hinzu. Die BASIC-Programme lohnen das Anschauen: sie sind prächtig geschrieben und vorbildlich dokumentiert. Man kann viel daraus lernen!

Auch für die Arbeit unter GDOS/NEWDOS einsetzbar und recht informativ ist für Besitzer von GENIE III die FDCINFO/CMD genannte Hilfsroutine, die dem 16-zeiligen 64-Zeichen-Bildschirmformat unten zwei Zeilen anfügt, die Informationen vom Laufwerkcontroler über die Arbeit der Schreib-Leseköpfe (in welchem Laufwerk gerade gearbeitet, ob geschrieben oder gelesen wird) bietet, die natürlich auch beim Löschen des übrigen Bildschirmes stehen bleiben. Die ca. 175 Byte dieser Routine stehen im RAM hinter F000h.

Schließlich bietet CP/Mac den unter CP/M üblichen „Type-Ahead"-Tastaturpuffer, der es erlaubt, während der Arbeit des Computers bereits bis zu 255 Zeichen auf der Tastatur einzugeben, die nach Abarbeiten der gegenwärtigen Aufgabe verwertet werden (lustig bei der Eingabe von <ENTER> beim „Durchblättern" der Sektoren mit den Editoren). Wenn man CP/Mac nicht nach Vorschrift mit Eingabe von <END>, sondern z. B. über MiniDos verläßt, bleibt dieses Feature auch unter GDOS/NEWDOS erhalten, solange ein unter GDOS/NEWDOS laufendes Programm den im RAM reservierten Puffer von A100h bis AFFFh und die in CP/Mac ca. bei 5600h stehende zugehörige Routine nicht überschreibt.

Der Autor des Programmes, Ulrich Heidenreich, hat mir auf meinen Wunsch für unser Club-Info folgende Produktbeschreibung von CP/Mac gegeben:

CP/Mac - Die Verbindung zwischen GDOS/NEWDOS und CP/M

CP/M ist das z. Z. bekannteste und weitverbreiteste Betriebssystem für Computer-Systeme mit CPUs der 80-er-Familie. Ebenso bekannt sind allerdings auch die Unzulänglichkeiten des CP/M, die teils auf die System-Portabilität, teils auf das Alter von CP/M zurückzuführen sind. Auf der anderen Seite steht dagegen GDOS/NEWDOS80 als eines der leistungsfähigsten Betriebssysteme, die je für 8-Bit-Rechnersysteme geschrieben wurden!

Nun ist ein Programmpaket „CP/Mac" erhältlich, das helfen soll, die Vorteile des CP/M-Systems, vornehmlich dessen Programm-Portabilität, mit den Eigenschaften von GDOS/NEWDOS80 zu verbinden. CP/Mac stellt ein Sub-DOS unter GDOS/NEWDOS80 dar, mit dessen Hilfe CP/M-Disketten bearbeitet werden können. So stehen mit CP/Mac - in an GDOS/NEWDOS angepaßter Form - die CP/M-Befehle DIR, STAT, ERA, REN, TYPE und USER, ein „Cross-PIP" für Dateikopien GDOS/NEWDOS80 <-> CP/M und ein CP/M-Disketten-Editor zur Verfügung.

Gegen einen DIN-A-4-Rückumschlag (DM 3,10 Porto) ist das CP/Mac-Handbuch vorab erhältlich von Ulrich Heidenreich, Werderstr. 35, 4300 Essen 1.

Ich möchte allen interessierten Clubmitgliedern wärmstens empfehlen, von diesem Angebot Gebrauch zu machen. Jeder, der meint, 8 Bit sei passé, wird seine Meinung mit CP/Mac revidieren müssen!

Richard Rensch

* Trau – schau, wem! *

– Ein BITchen Englisch –

"Vertrauen ist gut – Kontrolle ist besser!" soll ein führender Politiker gesagt haben. Um sein Copyright zu respektieren, sollte ich seinen Namen nennen; doch will ich mir das verkneifen.
Was er mit sechs Wörtern sagte, war zwar nicht neu; sie drücken das gleiche aus wie die kürzeren drei unseres alten Sprichwortes, nämlich: Erst hinschauen – dann (evtl.) vertrauen!
Genau das will das englische Verbum "CONTROL" besagen; jedenfalls in dem Zusammenhang, in dem wir Elektronikbeflissene (sei's HARD <X>OR SOFT) es immer wieder zu lesen bekommen: in Veröffentlichungen von Experten, in EDV-Artikeln, Bau- und Gebrauchsanleitungen u.ä.
Vor allem aber in übersetzungen. Und hier fast immer falsch übersetzt! Nämlich mit dem deutschen Lehnwort "Kontrolle". Das ist i.a.R. falsch. Auch namhafte Autoren machen diesen Fehler. Zwar sind die Wörter "control" und "Kontrolle" etymologich verwandt, aber semantisch durchaus verschieden. Wer ihren Unterschied mißachtet, erleidet leicht jenen Schiffbruch, vor dem die Überschrift warnen will!
Im großen "WILDHAGEN" steht bei *control:*
einschränken / im Zaum halten / beherrschen / leiten / beaufsichtigen / prüfen / (und schließlich auch) kontrollieren.
Wenn wir im Deutschen "kontrollieren" sagen, so meinen wir fast stets "Nachkontrolle". Ein Gegenstand wird gefertigt / eine Arbeit wird geleistet / eine Rede wird gehalten / (ein Programm wird abgefahren) usw. -- und dann wird "kontrolliert", ob es gut oder richtig war. War es das nicht, so wird der Gegenstand / die Arbeit / die Rede / (das Programm) usw. verworfen. Die Mühe war dann vergebens, Geld und Zeit sind verloren, evtl. ist Schaden schon angerichtet. Man hat erst "vertraut" und hinterher erst "hingeschaut"! Der Prozeß wurde kontrolliert, nachdem er abgelaufen – mit einem Wort: Es ist zu spät!
Um auf Hardware zu kommen: Das ist es nicht, was z.B. der *Drive Controller* leistet! Dieser ist kein "checker" (Aufseher), kein "tester" (Prüfer), kein "investigator" (Untersucher) oder wie man auch immer solche Leute nennen möge, die nachgucken! Er ist vielmehr aktiv vor und während des Ablaufs seines Schützlings = des zu überwachenden Prozesses! Ohne ihn läuft erst gar nichts! Was er tut, bezeichnet man im Deutschen mit "leiten" oder "steuern". To control heißt steuern, nicht (nur) kontrollieren!
In der industriellen Fertigung findet häufig nicht nur eine Endkontrolle, sondern auch ein 'production controlling' statt – eine Maßnahme während der Fertigung. Je nachdem, ob diese Maßnahme nur das Ergebnis des vorangegangenen Fertigungsabschnittes "kontrolliert", ist es eine "Produktionskontrolle", oder es ist eine "Fertigungssteuerung", wenn an diesem Punkt der Produktion Maßstäbe für den folgenden Fertigungsabschnitt gesetzt werden, m.a.W.: wenn der weitere Verlauf der Fertigung damit beeinflußt, gesteuert wird.
Dieser feine Unterschied ist wohl jedem klar, der einmal im Produktionsprozeß gestanden – oder hinreichend Fantasie hat. Diesen Unterschied sollte man bei übersetzungen auch in unserem Fach beachten. Sonst sind sie nicht mehr wert als die Auskünfte eines "Fachverkäufers", der die verlangte Ware aus dem jeweiligen "Fach" (seines Ladens) nimmt und dem wißbegierigen Kunden überreicht, ohne ihm die gewünschte fachliche Auskunft zu erteilen.
Etwas kontrollieren (check), also darauf "vertrauen", daß es schon richtig laufen wird, und erst hinterher nach"schauen" – das ist somit etwas wesentlich anderes, als etwas steuern (control), d.h. während des Ablaufs drauf schauen: "Trau – schau, wem!"
Der eingangs zitierte Politiker meinte mit "Kontrolle" natürlich genau das, was der Engländer (Amerikaner) i.a. unter "control" versteht. Hier irrte wieder mal der übersetzer. (Es sei denn, er ging davon aus, daß kontrollierte Tatbestände reversibel seien.)

* + * + * + * + *

Ich habe allerdings wenig Hoffnung, daß mein Geunke (dies sind drei, nicht zwei Silben!) von denen erhört wird, "to whom it may concern" – immerhin kenne ich Ausnahmen...
Denn: Ich kann es nur beobachten, aber nicht steuern!
Jedenfalls schaue ich mir stets z.B. erst den Verfasser an, bevor ich seinem Artikel, Brief oder seiner Rede traue...

KaFot

CLUB 80 HARD --- CLUB 80 HARD

Richtigstellung: 256K-Banker

Daß man beim Eintippen gelegentlich mal die SHIFT-Taste nicht erwischt, ist durchaus nichts besonderes; daß man beim Korrekturlesen aber nicht merkt, daß da anstelle der beabsichtigten Gänsefüßchen eine '2' steht, ist schon ein mittelprächtiges Verbrechen. Und genau das muß ich hier beichten.

In meiner Einbauanleitung für den 256K-Banker ist mir solches in der Tabelle zur Verdrahtung zwischen Banker und CPU-Board passiert. Beim GENIE I und -II liegt IORQ* natürlich nicht an Pin9 von IC2 sondern an Pin9 von IC16, für das dann auch die Bezeichnung 74LS367 zutrifft.

Für den zweiten Fehler fällt mir leider keine so elegante Ausrede ein. Beim TRS80 sucht man am IC Z55 vergeblich nach einem Pin 17, um sich da das Signal A0 zu holen. Ein 74LS367 hat nur 16 Beinchen und A0 liegt dort an Pin 11.

Erstaunlich, daß noch niemand protestiert hat, weil seine Aufrüstung auf 256K nicht funktioniert. Wahrscheinlich habt ihr alle diese Fehler entdeckt und selbständig beim Einbau korrigiert. Nur schade, daß mich niemand darauf hingewiesen hat.

Helmut Bernhardt

Bernd Drowälder

Floppydisk-Controller-Karte für den ECB-Bus

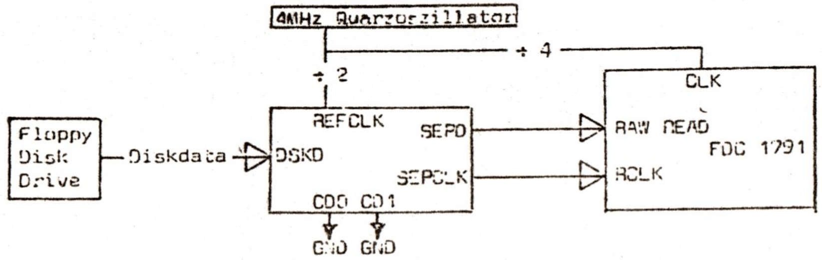
Die Karte zeichnet sich durch folgende Merkmale aus:

1.) software-mäßiges Umschalten zwischen Single- und Double-Density ist möglich

2.) einfacher Datenseperator ohne Abgleich

Der FDC 1771 arbeitet nur mit einfacher Dichte und hat bereits einen Datenseperator integriert.
Der FDC 1791, der in dieser Schaltung verwendet wird, arbeitet dagegen wahlweise mit einfacher oder mit doppelter Dichte. Er benötigt dagegen aber einen externen Datenseperator.

Beim Schreiben auf Diskette werden nicht nur die reinen Daten, sondern auch die Taktimpulse mit aufgezeichnet. Beim Lesen müssen nun die Daten innerhalb der kombinierten Daten- und Taktpulsfolgen erkannt werden. Die Bezeichnung Datenseperation für diesen Vorgang ist aber nicht ganz zutreffend. Die kombinierten Daten- und Taktimpulse werden zusammen auf den RAW READ-Eingang des 1791 gelegt. Am RCLK-Eingang erwartet der 1791 nun ein sogenanntes "Datawindow". Eine Trennung der Daten -was Datenseperation eigentlich bedeutet- findet also nicht statt. Das Datenfenster wird dem 1791 vom FDC 9216B bereitgestellt.



Blockschaltbild für 5 1/4-Zoll Single Density

Der 9216 benötigt je nach Laufwerk und Aufzeichnungsart eine bestimmte Referenzfrequenz. Um nun nicht andauernd den Takt ändern zu müssen, hat der FDC 9216B die Möglichkeit, intern eine Teilung des Taktes zu bestimmen.

Laufwerk	Dichte	Referenzfrequenz
8 Zoll	Double Density	8 MHz
8 Zoll	Single Density	4 MHz
5 1/4 Zoll	Double Density	4 MHz
5 1/4 Zoll	Single Density	2 MHz

Über die Eingänge CD0 und CD1 kann der interne Teiler eingestellt werden. Da die Schaltung für 5 1/4 Zoll, Single- und Double-Density, vorgesehen ist, benötigt man also einen Referenztakt von 4 MHz. Für Single-Density wird nur ein Referenztakt von 2 MHz benötigt; der Takt muß also intern dutch zwei geteilt werden. Die Teiler des 921B werden folgendemaßen eingestellt:

CD1	CD0	Teiler
0	0	1
0	1	2
1	0	4
1	1	8

Der erste Track einer Diskette muß beim TRS-80 in einfacher Dichte geladen werden. Erst danach kann auf doppelte Dichte umgeschaltet werden. Dieses Umschalten erfolgt beim TRS-80 über die Adresse des Command-Registers des FDC 1791. Damit keine unerlaubten Operationen dabei ausgeführt werden, mußte ein Befehlswort gefunden werden, welches keine Reaktion des Contollers auslößt.
Dieses Datenwort ist FEH zur Umschaltung auf Single-Density und FFH zur Umschaltung auf Double-Density. Wie aus dem Datenblatt des 1791 ersichtlich ist, kennt der Controller dieses Command-Wort nicht.

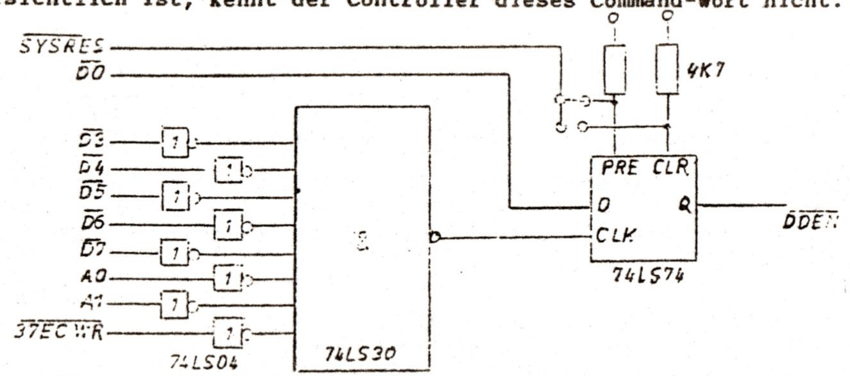


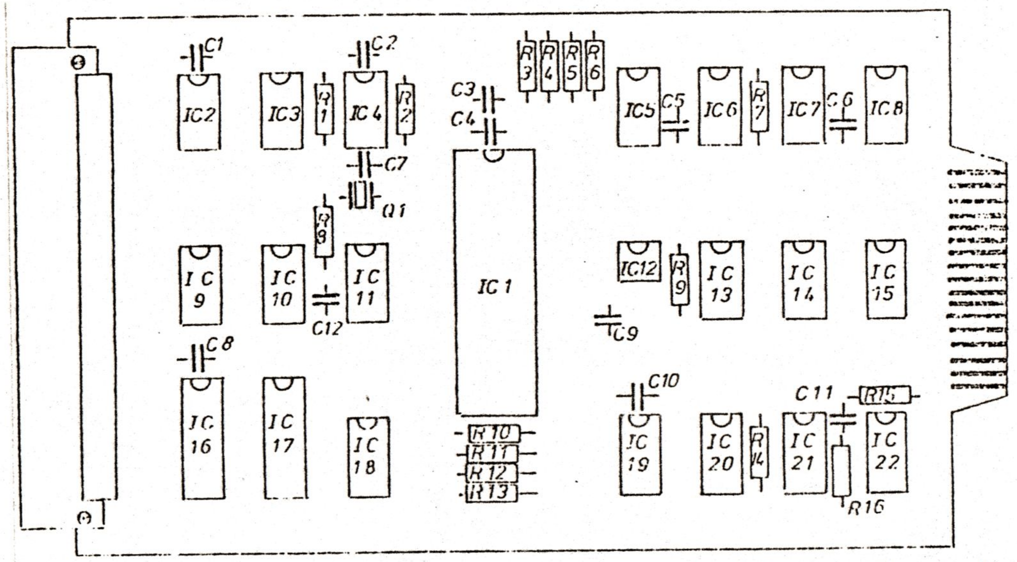
Bild 6.0 zeigt die Schaltung zur Umschaltung der Dichte

Mit Invertern wird die Invertierung der invertierenden Eingangstreiber wieder umgekehrt. Der Ausgang des 74LS30, einem NAND-Gatter mit 8 Eingängen, nimmt nur LOW-Pegel an, wenn alle 8 Eingänge HIGH-Pegel haben. Damit der Ausgang LOW wird, müssen D3 bis D7 HIGH und die Leitungen A0,A1 (werden nur einmal invertiert) und 37EC WR LOW sein. Dann fällt der Ausgang auf LOW und das D-Flip-Flop übernimmt die Information von D0, die am D-Eingang des 74LS74 anliegt, an den Ausgang Q.

Die Datenleitung D0 am D-Eingang des 74LS74 wird nur einmal durch die Eingangstreiber invertiert. Beim Datenword FEH liegt am D-Eingang dann ein HIGH an und -beim folgenden Clock- wird das HIGH an den Ausgang übernommmen.
Über den DDEN-Eingang des FDC 1791 wird dem Controller so mitgeteilt, daß mit einfacher Dichte gearbeitet werden soll.Gleichzeitig liegt auch an der Steuerleitung CD0 des Datenseperators HIGH-Pegel an und veranlaßt den internen Teiler , die Referenzfrequenz intern durch 2 zu teilen.Wie für Single-Density gefordert, beträgt die Referenzfrequenz dann 2 MHZ.Entsprechent ist der Ausgang des D-Flip Flops beim Datenwort FFH LOW, der Controller arbeitet in Double Density und der Datenseperator mit einer internen Referenzfrequenz von 4MHZ.

HEFT 18 Februar 1987

Die BOOT-Disketten im NEWDOS 80 V.2.0 Format enthalten deshalb sofort zu Anfang der BOOT-Routinen einen Befehl zum Umschalten auf Double oder Single Density.Im Double Density Format weisen die Disketten dabei eine Besonderheit auf.Das NEWDOS arbeitet dann mit zwei BOOT- Sektoren,einen auf Track 0 Sektor 0 zum laden in Single Density und einen auf Track 1 Sektor 0 zum Weiterladen in Double Density.Der Track 0 wird vom Betriebssystem dann bei der weitern Arbeit ignoriert.Track 1 wird als Track 0 angesehen und das Betriebssystem arbeitet insgesamt dann nur noch mit 39 statt mit 49 Tracks.
Dieses kann aber noch umgangen werden. über den Preset-Eingang des 74LS74 wird bei jedem Reset der Ausgang Q auf HIGH gelegt. Somit ist sichergestellt, daß bei jedem Reset automatisch auf Single-Density zum Laden des ersten Tracks umgeschaltet wird. Wird der Sys-Reset aber nicht auf den Preset-, sondern auf den Clear-Eingang des Flip-Flops gelegt, schaltet der Controller sofort auf Double-Density um und die ganzen 40 Tracks können in doppelter Dichte geschieben werden. Da man aber nicht mehr zu Disketten von anderen Rechnern kompatibel bleibt, wird diese Möglichkeit nur optional vorgesehen.



Bestückungsplan Floppykarte

.4.' Bestückungsliste Floppykarte

Halbleiter

IC1	1791	Floppy Disk Controller
IC2	74LS139	
IC3	74LS155	
IC4,10,18	74LS04	
IC5,13	74LS74	
IC6	74LS90	
IC7,8	4518	
IC9,11	74LS30	
IC12	FDC 92165	Dataseparator
IC14,15	7416	
IC16,17	74LS240	
IC19	74LS175	
IC20	74LS20	
IC21	74LS123	
IC22	74LS123	

Kondensatoren

C1 - C6,C8 - C10	100 nF
C7	100 pF
C11	47 uF
C12	nicht einsetzen

Widerstände

R1,R2	1 KOhm
R3-R6,R9,R13,R15,R16	4,7KOhm
R7,R10-R12	220 Ohm
R8	0 Ohm
R14	220KOhm

Sonstiges

J1	64 pol. Steckerleiste
Q	Quarz 4 MHz
4	Lötnagel

Umwandlung von Betriebssystemen einfacher Dichte in Betriebssysteme doppelter Dichte

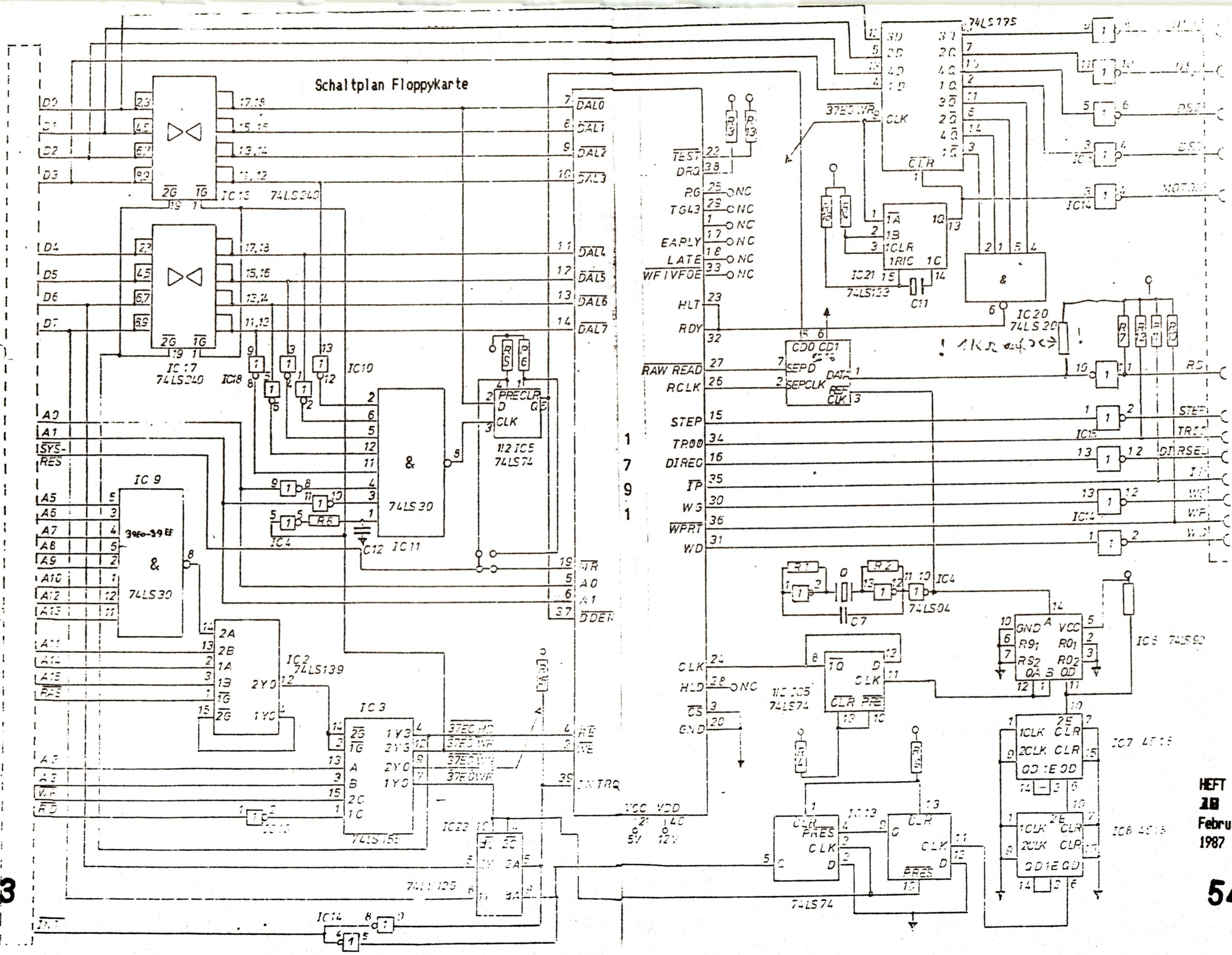
Disketten, die mit einfacher Dichte beschrieben sind, müssen leicht abgeändert werden, bevor sie mit dieser Schaltung auf doppelte Dichte umgewandelt werden können. Dieses ist bedingt durch den FDC 1791. Im Gegensatz zum 1771 kennt der 1791 nur ein sogenanntes Data-Adress-Mark. Dieses wird beim Betrieb mit dem 1771 vom NEWDOS dazu verwendet, die Directory zu markieren. Um diesen Fehler zu beseitigen, müssen Disketten, die mit dieser Schaltung betrieben werden, einmalig die Directory schreibgeschützt bekommen. Dieses erfolgt mit dem Programm DIRCHECK/CMD, welches auf jeder NEWDOS 80-Systemdiskette vorhanden ist. Diese Maßnahme hat keinen Einfluß auf das Arbeiten mit anderen Rechnern. Die Disketten können auch weiterhin auf allen Rechnern verwendet werden.
Ist die Diskette jetzt soweit vorbereitet, muß noch dem Betriebssystem mitgeteilt werden, daß mit doppelter Dichte gearbeitet werden soll. Dies erfolgt durch Abändern der Pdrive-Angaben einer Diskette (siehe NEWDOS 80-Handbuch).

Mit dieser Schaltung wird es auch ermöglicht, Programme vom TRS-80 Modell III auf den TRS-80 Modell I zu kopieren und umgekehrt. Dazu muß sich in Laufwerk 0 eine Boot-Diskette befinden und die Pdrive-Angaben für Laufwerk 1 auf 40 Track mit doppelter Dichte eingestellt sein. Mit einem einfachen Copy kann dann von der Modell III-Diskette, die sich im Laufwerk 1 befindet, ein Programm rüberkopiert werden. Steht nur ein LAufwerk zur Verfügung, erfolgt die Kopie in leicht abgewandelter Form. Soll zum Beispiel das Programm TEST/CMD kopiert werden, muß folgende Eingabe gemacht werden:
COPY,0,TEST/CMD /CMD SPDN=1
Bei der Kopie muß für die Source-Diskette die Modell I-Diskette und für die Destination-Diskette die Modell III-Diskette eingelegt werden. Durch den Zusatz SPDN=1 wird für die Zeit des Kopierens angenommen, daß das Laufwerk 0 die Charakteristik des Laufwerks 1 hat.
Abschließend muß noch erwähnt werden, daß diese Schaltung nur unter den NEWDOS 80 Versionen 1 und 2 (und kompatibele) läuft. Beim TRS-DOS wird die Umschaltung auf Double-Density nicht unterstützt. Beim Versuch, eine TRS-DOS-Diskette schreibgeschützt zu schreiben, wird diese unweigerlich zerstört.

Da viele Computer-Freunde bereits eine Floppy-Controller-Karte besitzen, befindet sich im Anschluß an die Floppyschaltung eine weitere Schaltung zum Umrüsten dieser Karte auf Double-Density mit Hilfe einer Aufsatzkarte.

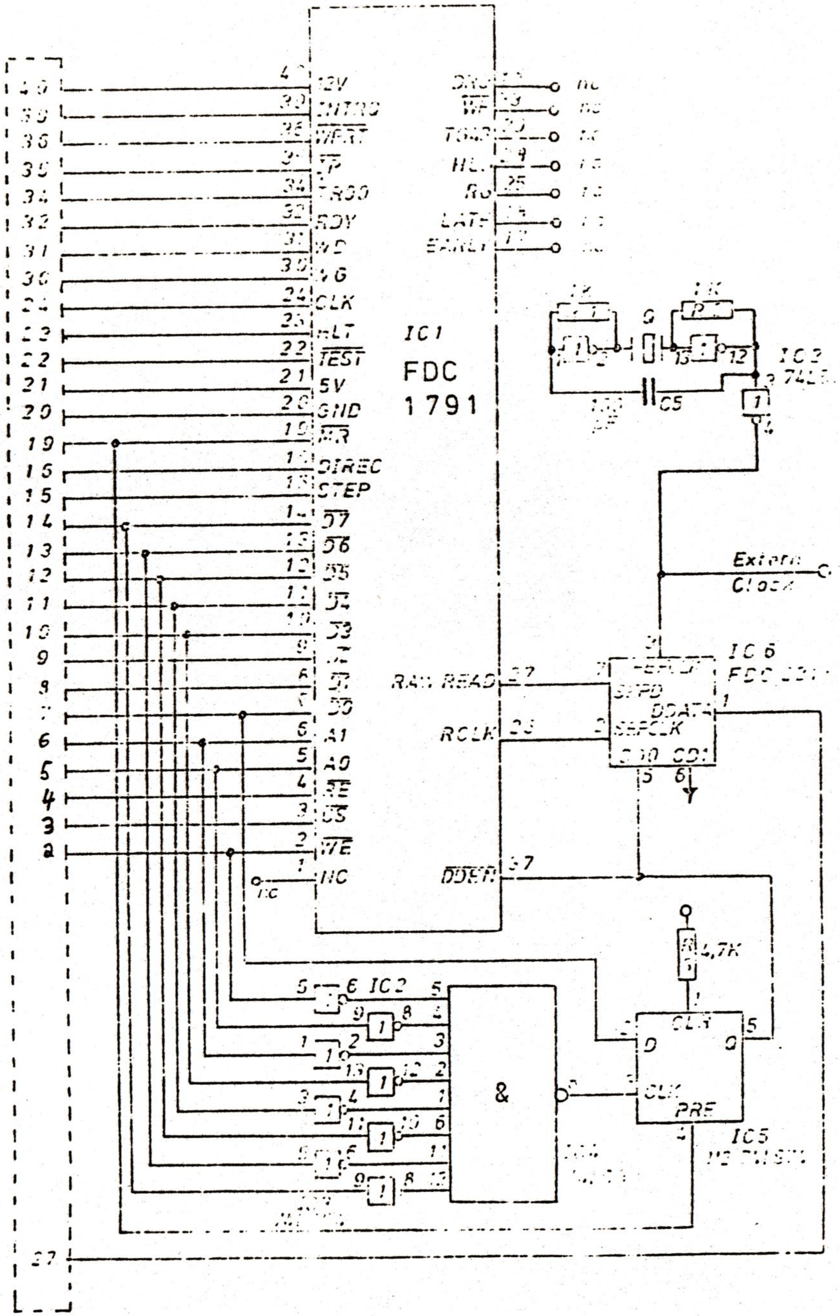
P.S.: Eine kleine Änderung, die für den CP/M-Betrieb notwendig ist, folgt später. Viel Spaß beim Nachbau.Bernd Drowälder

Schaltplan FloppyKarte

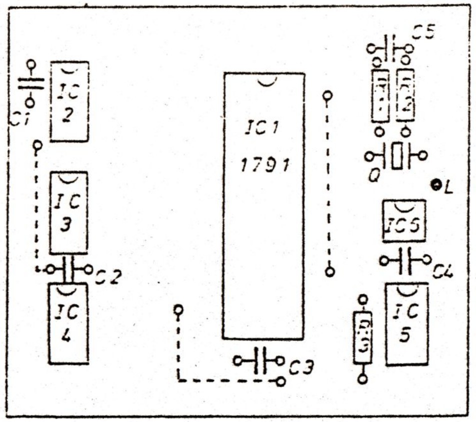


HEFT 18
Februar 1987

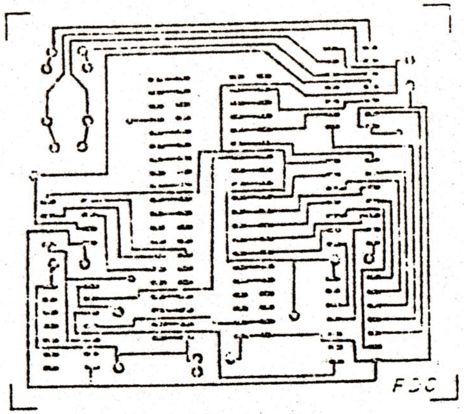
Schaltplan Floppyaufsatzkarte



Bestückungsplan Floppyaufsatzkarte



Platinenlayout Floppyaufsatzkarte



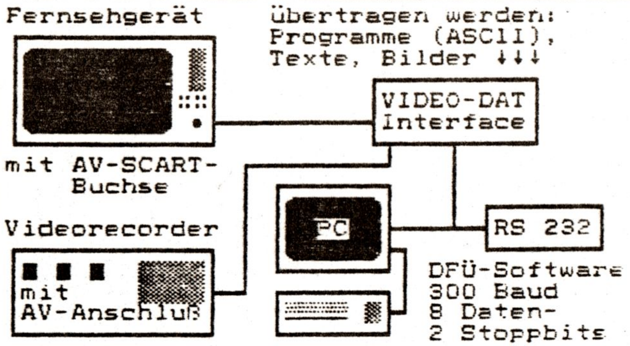
Bestükungsliste Floppyaufsatzkarte

Halbleiter	
IC1	FDC 1791
IC2, IC3	74LS04
IC4	74LS30
IC5	74LS74
IC6	FDC 9216 B
Widerstände	
R1, R2	1 kOhm
R3	4,7 kOhm
Kondensatoren	
C1 - C4	100 nF
C5	100 pF
Sonstiges	
Q	4.000 MHz Quarz
L	Lötnagel
Z	20 pol. Wrapleisten

HARDWARE
VIDEODAT - SYSTEM

Ein einfaches Verfahren zur Übermittlung von Daten während einer Fernsehsendung, ohne daß der Zuschauer beim Betrachten der Sendung gestört wird.

Redaktion: Wolfgang Back Januar 1986



Das VIDEODAT 300 Interface kann als

Bausatz	oder als	Fertigmodul
DM 68,00		DM 108,00
DM 7,50	Porto+Verpackung	DM 7,50
DM 75,50	««« Summe »»»	DM 115,50

gegen Vorkasse auf das Konto der Firma CONDATA Weilerstraße 20, 5040 Brühl bei der Kreissparkasse Köln Konto-Nr. 1182028051 BLZ 370 502 99 bestellt werden.

Das VIDEODAT-SYSTEM wurde von den Mitarbeitern der Fernsehsendung COMPUTER-CLUB erfunden und realisiert, um den Service einer zusätzlichen Information zur Sendung zeitgerecht gestalten zu können. Es ist uns nicht bekannt, daß diese Art der Datenübermittlung weltweit realisert oder probiert wurde.

Gegenwart: Seit ca. 3 Jahren unterhält die Sendung COMPUTERCLUB einen zusätzlichen Informationsservice, der aus dem Ausstrahlen von Computerprogrammen oder Texten über den Tonkanal des Fernsehers besteht.
An bestimmten Stellen in der Sendung wird die Übertragung angekündigt und das normale gesprochene Wort o.ä. durch ein moduliertes Computersignal ersetzt. Der Zuschauer zu Hause zeichnet diese Tonübertragung auf einen gewöhnlichen Kassettenrekorder auf, indem er eine Verbindung zur Kopfhörer- oder zur DIN-Buchse seines Fernsehers herstellt. Nach gelungener Aufzeichnung kann er dann dieses so empfangene Programm oder den so empfangenen Text in den Heimcomputer einspielen und mit dieser Information direkt weiterarbeiten.
Durch den sogenannten BASICODE wurde sichergestellt, daß etwa 30 verschiedene Heimcomputer an dieser Datenübertragung

Vorteile: ohne großen technischen Aufwand und ohne spezielle Zusatzgeräte können viele Zuschauer gleichzeitig und direkt in den Genuß eines zentral erstellten Computerprogrammes oder eines Informationstextes kommen. Das Computerprogramm wird lauffähig übermittelt, ebenso ist der empfangene Text auf dem Bildschirm der Heimcomputer lesbar oder gar- bei Vorhandensein eines Druckers- auf dem Papier darstellbar.
Das Verfahren wird seit drei Jahren erfolgreich in der Sendung COMPUTERCLUB eingesetzt.

Nachteile: während der Ausstrahlung des Computerprogrammes kann keine normale Sprache oder Musik dem Bild unterlegt werden. Dem Prinzip der Datenübermittlung liegt zugrunde, daß wir das Datensignal mit vollem Pegel ausstrahlen müssen. Der Frequenzbereich, in dem dies definitionsgemäß stattfinden muß, ist im hörbaren Bereich und das Signal entspricht einem schrillen Ton, der als "nicht angenehm" zu bezeichnen ist. Der Einsatz dieser Datenübermittlung kann daher nur dort zur Anwendung kommen, wo der Wunsch nach diesen Informationen über den Tonkanal größer ist, als der Konsum einer "normal" gestalteteten Sendung. Dieses beschränkt sich daher auf Sendungen, in denen der Computer inhaltlich im Mittelpunkt steht.
Der Fortschritt der Computertechnik in den letzten Jahren führte dazu, daß "gute" und aus der Norm herausragende Comuterprogramme immer aufwendiger und länger werden. Dies bedeutet natürlich gleichzeitig, daß die Ausstrahlung dieser Programme, nach dem oben beschriebenen Verfahren entsprechend länger dauert. Dieses aber findet seine natürliche Grenze in der Zumutbarkeit des schrillen Dauertones im Tonkanal der Fernsehsendung.
Dieser Sachzwang führte dazu, daß wir uns Gedanken machten, wie wir heute diese Datenübermittlung neben einer Fernsehsendung eleganter realisieren können. Das Ergebnis zeichnen wir als VIDEODAT - SYSTEM, da DATEN im VIDEO-signal übermittelt werden.
Das Prinzip des Fernsehens besteht zum einen aus der drahtlosen oder drahtgebundenen gleichzeitigen Ausstrahlung eines Bild- und eines Tonsignals.
Das man anstatt natürlicher Sprache oder Musik über den Tonkanal auch Daten anderer Art schicken kann, wurde oben beschrieben.
Natürlich läßt sich auch der Bildkanal zum Transport von Dateninformationen nutzen. Zusätzliche Informationen werden heute bereits in der sogenannten "Austastlücke" mitgesendet, so z.B. der VIDEOTEXT der Fernsehanstalten.
Der VIDEOTEXT könnte natürlich sehr gut zur gewünschten Übermittlung von Zusatzinformationen genutzt werden. Doch gibt es hier auch Einschränkungen. Zum einen ist der VIDEOTEXT so angelegt, daß hier allgemeininteressierende Informationen angeboten werden sollen. Zum anderen ist die Kapazität (80 Seiten) nicht allzu groß. Dies beschränkt man bewußt, um die Umblätterzeiten gering zu halten.
Daneben gibt es noch einen weiteren Nachteil, den viele vielleicht als Vorteil bezeichnen mögen: für viele Anwendungen ist der VIDEOTEXT zu schnell, so daß die notwendige Hardware aufwendig und damit teuer sein muß. (VIDEOTEXT-Decoder mit zusätzlicher Schnittstelle zum Computer)
Das VIDEODAT - SYSTEM legt deshalb die Zusatzinformation in das Nutzvideosignal und zwar: die Datenübertragung findet in der obersten Ecke des Fernsehers in den Zeilen 25 bis 35 statt.
Viele vorbereitende Tests ergaben, daß das von uns eingestanzte Schwarzfenster - in der linken oberen Ecke - bei den meisten Heimfernsehern außerhalb des Sichtfensters liegt. Auch neuere Rechteckröhren lassen das Datenfenster fast

völlig verschwinden. Sollte dennoch - durch Einstellfehler - oder bei Benutzung von Studiotechnik das Kästchen sichtbar sein, so ist dies als nicht störend empfunden worden.
In diesem kleinen Schwarzfenster findet die zusätzliche Datenübermittlung statt. Derzeit wird in jedes zweite Halbbild (es geht auch mit jedem Halbbild) ein Byte (sprich: ein Buchstabe oder eine Zahl) mit Start- und Stoppbit übertragen Wir haben die Geschwindigkeit so gewählt daß die Datenübertragung mit 300 Baud stattfindet, da viele Zuschauer bereits mit ihrem Heimcomputer dieses Verfahren (mit Akustikkoppler) kennen und eine entsprechende Software bereits haben.

300 Baud: das bedeutet etwa 30 Zeichen pro Sekunde, die mit diesem Verfahren derzeit übermittelt werden. Bezogen auf eine Halbstundensendung, wie es der COMPUTERCLUB oder der COMPUTERSHOP darstellt, ist die zusätzliche mitliefer- bare Datenmenge einfach zu errechnen:

30 Zeichen mal 60 Sekunden= 1800 Zeichen pro Minute.

30 Minuten mal 1800 Zeichen = 54000 Zeichen pro Sendung.
Umgerechnet auf DIN-A4 - Seiten bedeutet dies:
bei 60 Anschlägen auf 40 Zeilen ergibt eine DIN-A4-Seite: 2400 Zeichen.
54000 Zeichen pro Sendung / 2400 Zeichen pro Seite = <u>22</u> Seiten
Es kann somit, ohne daß eine Sendung gestört wird, zusätzlich eine erhebliche Datenmenge zur Auswertung mit dem Heimcomputer mitgeschickt werden. Diese Informationen können sowohl Text als auch Programme oder auch Bilddateien sein. Über weitere Möglichkeiten werden wir noch berichten.
Ein weiterer Vorteil besteht bei dem VIDEODAT - SYSTEM in der Videorecorderfestigkeit. Im Gegensatz zum VIDEOTEXT wird die Datenübermittlung zum einen relativ langsam und vor allem im Videonutzsignal vorgenommen. Da die Modulation des Signals aus weiß = "ja" und schwarz = "nein" oder vice versa besteht, ist dieses Verfahren äußerst unkritisch in puncto Übertragungssicherheit.
Einige Versuche ergaben, daß nach mehrmaligem Umspielen von einem VHS-Recorder zu einem Beta-Gerät und vice versa, der normale Bildinhalt zwar immer mehr verrauschte, die Datenübertragung jedoch sicher und fehlerfrei funktionierte.
Obgleich diese Übertragungssicherheit von der Theorie her prinzipbedingt ist, so verblüfften die Tests in der Praxis doch sehr.
Das VIDEODAT - SYSTEM bietet in puncto Datenübermittlung zum Endverbraucher hin viele Vorteile. Auch eine Würdigung gegenüber dem VIDEOTEXT der Fernsehanstalten versuchten wir aufzuzeigen.
Es soll natürlich auch über den Nachteil gesprochen werden, den man bei VIDEODAT gegenüber der beschriebenen Tonmethode zu sehen hat. Bei der Tonmethode kommt man mit einem handelsüblichen Kassettenrekorder und einer Verbindungsschnur aus Bei dem VIDEODAT - SYSTEM muß man sich ein sogenanntes INTERFACE bauen oder besorgen, um an dem Datenverkehr teilnehmen zu können. Ein Computer zum Empfang der Signale benötigt man bei beiden Systemen.
Dieses Interface, das man sich derzeit aus "Bauteilen zum Ladenpreis" für etwa DM 40,00 zusammenbauen kann, holt aus dem Videosignal das Datensignal heraus und bereitet es für den Computer auf. Die Auswertung der Information übernimmt der Computer; ein Verfahren, das die meisten Heimcomputerbesitzer bereits heute beherrschen.
In der Sendung COMPUTERCLUB vom 12. Januar 1986, stellten wir das technische Verfahren von VIDEODAT vor und fragten nach der Bereitschaft der Zuschauer, sich ein solches Interface zu besorgen. Die Reaktion war überwältigend positiv.

Ausblicke und Nutzungsmöglichkeiten

Man kann die Möglichkeiten dieses neuen VIDEODAT - SYSTEMS zuerst einmal ganz pauschal beschreiben:
Jede Sendung könnte für alle, die ein Interesse an zusätzlicher Information besitzen, diese Information auf einfache Art mitliefern.
Die Prozedur ist von uns bewußt einfach gehalten worden. Jede Sendung, auch bereits fertige Sendungen bis zuruck ins Jahr 1952, können mit der Information versehen werden. Das Aufbringen der Information ist studiounabhängig. In Kurzform: MAZ 1 liefert das Muttersignal das über den Computer geschleift wird und automatisch zeichnet MAZ 2 Bild und Daten auf.
Nun wird man sicherlich nicht jede Sendung mit dem zusätzlichen Datendienst ausrüsten wollen. Von Vorteil wird es für alle Sendungen sein, die einen hohen Lerninhalt besitzen, so im Bereich Schulfernsehen, Telekolleg, Wissenschaft usw. Ein Beispiel sei hier aufgeführt:

Schulfernsehsendung über die Dampfmaschine.

Mitgeliefert werden über VIDEODAT 22 DIN-A4-Seiten über die Geschichte der Dampfmaschine, die Funktion usw.
Der Lehrer zeichnet diese Sendung vielleicht auf. Er spielt außerhalb des Unterrichts die Daten über einen Videorecorder in einen der vorhandenen Schulcomputer, druckt das Ergebnis aus. Gemeinsam mit den Schülern schaut er sich bei passender Gelegenheit die Sendung an.
Man kann an viele weitere Möglichkeiten denken:
Zusätzliche Informationen an Hörgeschädigte (Vielfaches der Untertitelung).
Zusätzliche Informationen für Nichtsehende im Braillecode zum Nachlesen.
Übermittlung von Computerprogrammen, von Begleitschriften.
Übermittlung der Hobbytips in der Hobbythek, Rezeptvorschläge o.ä. bei eff-eff usw.

Wir planen im COMPUTERCLUB für unsere derzeit 25.000 Mitglieder die Clubzeitschrift mit Programmen, Text und Bildern auf diese Weise dem Zuschauer monatlich zu übermitteln.

Es sind bei dieser Beschreibung eines neuen Zusatzdienstes nicht alle möglichen Formen der Nutzung niedergeschrieben. Bei der Einfachheit der Handhabung und bei der gleichzeitig sicheren Übermittlung und bei der Möglichkeit der Aufzeichnung auf andere Magnetträger lassen sich in der Zukunft noch viele weitere Einsatzmöglichkeiten finden. Außerdem arbeiten wir daran, für die Nutzer des Dienstes einen kostengünstigen Customerchip zu entwickeln, der alle notwendigen Funktionen zur Teilnahme am Datendienst bis hin zum Computer enthält. Es kann dann in Bälde mit einem Preis für das notwendige VIDEODAT - INTERFACE von etwa DM 10.00 gerechnet werden. Diese Investition dürfte für jeden Interessierten tragbar sein.

Das Projekt

Wie aus einem Fernsehgerät ein Monitor wird

Was tun, wenn ein billiger Monitor gesucht wird, die käuflichen aber für den eigenen Geldbeutel zu teuer sind ? Nichts einfacher als das. Man sucht sich einen TV-Portable, welches dem eigenen Designwunsch am nächsten kommt und hilft der Natur ein wenig auf die Beine. Hier sind lediglich einige Veränderungen an der Videoendstufe vorzunehemen, sowie für einen geeigneten Videoeingang zu sorgen. In der Regel kann dazu der Antennenanschluß benutzt werden. Fernsehempfang ist ja eh nicht gefragt. Also wozu den Luxus ?. Genau betrachtet ist das eigentlich bereits alles. Konkret gerechnet sind neben dem Geld ca. ½ Tag Lötarbeit und so um die DM180.- zu investieren. Wer´s nicht glaubt, dem sei gesagt, daß mein ´Monitor´ auf genau diesem Wege das Licht der Welt erblickt hat. Die Videobandbreite liegt im Bereich 10...12Mhz, also für 64 Zeichen/Zeile ein gestochen scharfes Bild (übrigens läßt sich die Videobandbreite auf simple Weise testen. Hierzu werden Helligkeit und Kontrast normal eingestellt und dann die Helligkeit langsam verringert. Bei dieser Prozedur muß das zuvor mit Buchstaben gefüllte Bild gleichmäßig dunkler werden = ausreichende Bandbreite. Bei Geräten mit zu geringer Bandbreite verschwinden die Punkte in den Buchstaben und Zahlen als erstes, sodaß nur noch die horizontalen Linien zu erkennen sind. Hier ist das Klassenziel nicht erreicht und wohl noch ein wenig ´Arbeit´ von Nöten. Mit welchen Methoden ich mein ´Opfer´ nun ´zurechtgebogen´ habe, folgt als nächstes im Detail.

Vorüberlegungen
Bevor ans Werk gegangen werden kann, muß zuerst die elektrische Situation im Fernsehgerät erkundet werden:

- Wo kommt das Videosignal her ?
- Wie wird der Arbeitspunkt der ersten Stufe am Ausgang des Videodemodulators eingestellt ? (Wichtig für die Einspeisung unseres eigenen Videosignals aus dem Rechner)
- Wo liegen die Ton-ZF-Sperren im Videokanal ? (Müssen entfernt werden, da sie nur stören)
- Wie ist die Beschaltung im Kollektor bzw. Emitterkreis des Videoendtransistors ? (Wichtig, da später an diesen Stellen Teile eventuell ausgebaut bzw. eingefügt werden müssen)
- Auf welche Pins an der Röhre wird die Heizung geführt ? (Hier muß mit Spulen ´verdrosselt´ werden, um die Kapazität zwischen Kathode <-> Heizfaden nach Masse zu reduzieren)

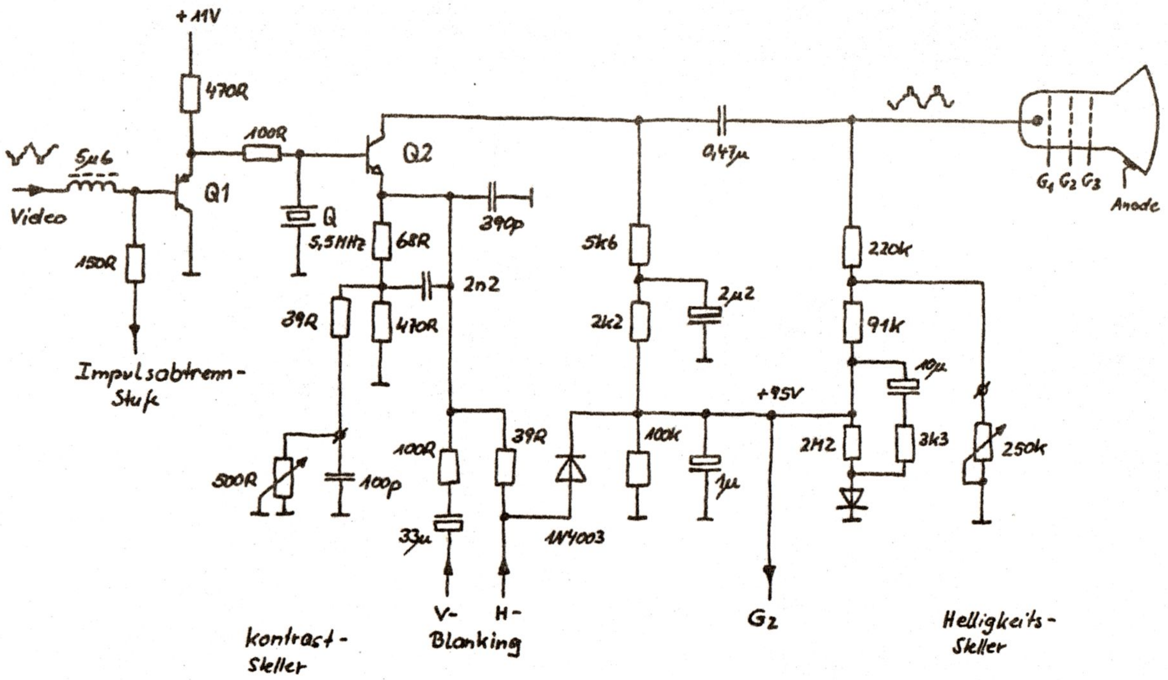


Bild 1: Gesamtschaltung des Videokanals

Die Videovorstufe

Als erstes ist die Verbindung der Videovorstufe (Q1) zum Videodemodulator aufzutrennen (Bild 1). Die Basisspannung ist vorher mit einem DVM zu messen, damit nach Abschluß der ersten ´Teilstufe´ die richtige Einstellung wiedergefunden werden kann. Eine einfache Lösung bietet sich über entfernen der 5.6 uH Spule in der Basisleitung von Q1 an. Man erspart sich so das Auftrennen von Leiterbahnen. Stattdessen sind nun eine Reihe von neuen Bauteilen einzufügen (s. Bild 2):

R1	100R		Trimmpoti
R3	1K	1/8 W	Kohleschichtwiderst.
R4	10K		Trimmpoti
C1	220uF	16V	Elko, stehend
C2	100nF	50V	Keramikkondensator
C3	47uF	16V	Elko, stehend

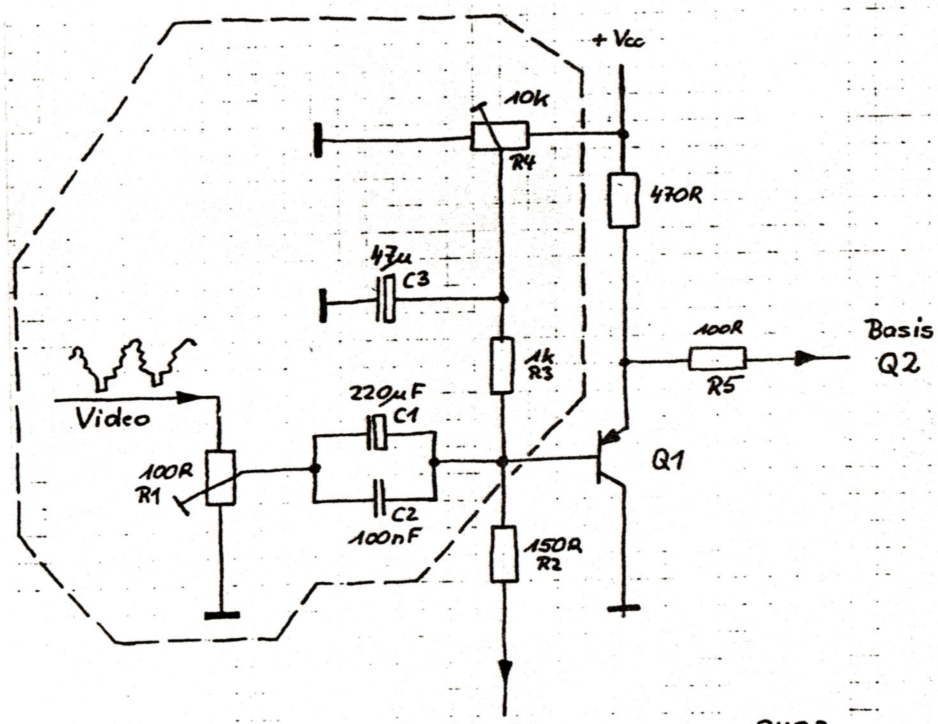


BILD2

Der Clou an dieser Methode ist folgender: L1, R15 und die Röhrenkapazität C_K bilden einen Parallelschwingkreis dessen Resonanzfrequenz in geeigneter Weise zur -3dB Grenzfrequenz der Videoendstufe gelegt ist. Bei richtiger Wahl wird dann der Amplitudenabfall der Videoendstufe im Bereich der Grenzfrequenz kompensiert und es ergibt sich eine neue, höherliegende Grenzfrequenz (s. Bild 5). Der Problempunkt an dieser Lösung ist C_K, sie ist meistens nicht exakt bekannt und unterliegt in der Regel größeren Streungen. Hier hilft nur probieren.

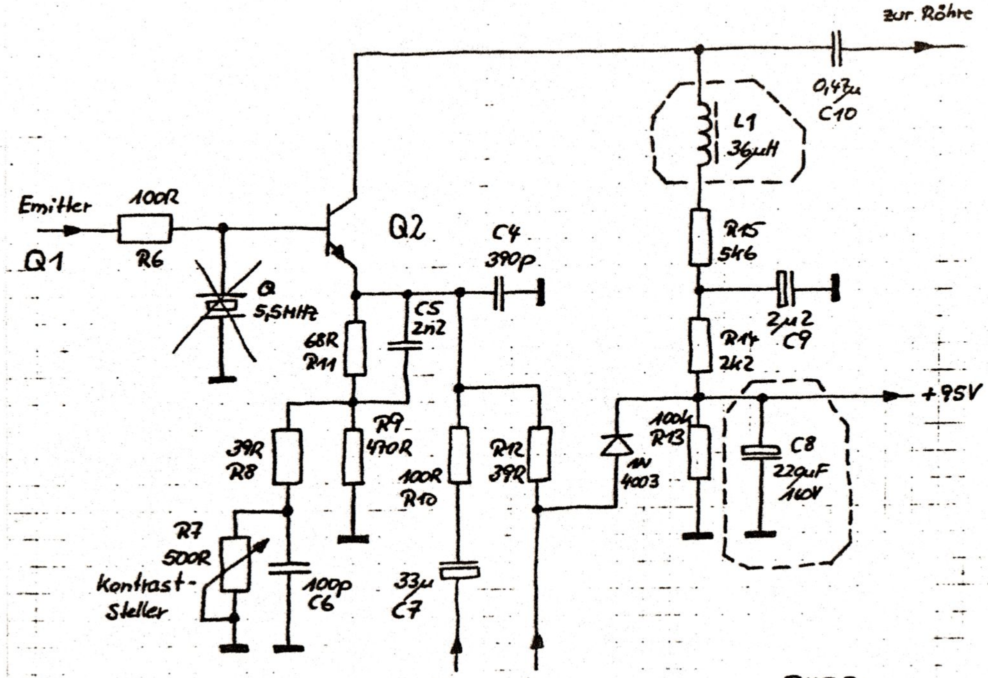


BILD3:

Nicht übersehen werden darf bei allem die Beschaltung im Emitterkreis von Q2. Hier hat C4 eine ´tragende´ Rolle. Er reduziert für hohe Frequenzen zunehmend die gegenkoppelnde Wirkung des Netzwerkes (R11//C5)+(R9//(R8+(R7//C6))). Kurz gesagt, beides zusammen ergibt erst die richtige Mischung.

Es fehlt eigendlich nur noch die Bildröhre. Auch sie soll nicht ungeschoren bleiben. Damit die Videoendstufe nicht allzuviel ´Blindleistung´ leisten muß, wird die Heizung mit zwei 10uH Spulen HF-mäßig gegen Masse entkoppelt (s. Bild 6). Diese Spulen sollten Breitbanddrosseln sein und einen Strom von ca. 300mA ohne Probleme führen können. Ich habe hier Spulen der Fa. Dale verwendet (standen gerade zu Verfügung).

Der Trimmer R1 ist mit 100 Ohm sicherlich nicht der ideale Abschlußwiderstand, 75 Ohm wären richtiger. Jedoch sind bei den hier anzutreffenden Verhältnissen keine so strengen Maßstäbe zu setzen Reflektionen sind bei meinem Gerät nicht zu erkennen, dafür sind die zu erwartenden Flanken (t_R = 0.36 / 10Mhz = 0.036E-06 = 36nsec) viel zu langsam und die Leitungen auch zu kurz. Alles andere was so aussieht, dürfte eher durch Überschwingen der Flanken am Ausgang der Videoendstufe anzusehen sein - als Folge einer falschen Frequenzgangkorrektur. R1 ist aus Befestigungsgründen am günstigsten an der Antennenbuchse zu befestigen. Diese gibt gleichzeitig nach Entfernen der Leitung zum Tuner eine gute ´BNC´-Buchse ab. C1 muß groß bemessen sein, da sonst während des Bildrücklaufs die Umladung so erheblich ist, daß sie als Graukeil im Bild erkennbar. C2 fällt die Aufgabe zu, die Eigeninduktivität von C1 im Zaum zu halten. Es würden sonst die höherfrequenten Signalanteile unerwünschterweise benachteiligt (Tiefpass aus L_{ELKO} und dem r_E der Transistorstufe Q1). C3 besorgt die Siebung der Basicvorspannung (Netzbrumm, Rauschen, Spikes etc.). Soviel zum Thema Videovorstufe. Der größere Brocken kommt jetzt: die Videoendstufe. Wer will, kann ja erst einmal Kaffeekanne zücken und sich ein Päuschen einlegen.

Die Videoendstufe

Das erste Teil, welches weichen muß, ist der Quarz für die Ton-ZF-Sperre (Q, 5.5Mhz). Er ist hier fehl am Platze. Der 100 Ohm Widerstand in der Basisleitung kann bleiben, er bildetete in Verbindung mit dem Quarz einen frequenzabhängigen Spannungsteiler mit sehr selektiver Wirkung. Die Bauteile im Emitterkreis werden so belassen, da für das angestrebte Endziel passend dimensioniert. Der ´Knalleffekt´ mit der großen Wirkung kommt in den Kollektor: die 36uH Spule L1.

Wenn die Lastkapazität am Kollektor bekannt ist (meist jedoch nicht), kann L1 auch berechnet werden (ungefähr). Ich hatte nur diese eine, die dann überraschenderweise auch noch den richtigen Wert besaß (Der dümmste Bauer hat die dicksten Kartoffeln und so...).

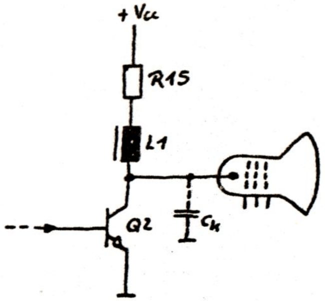


BILD4:

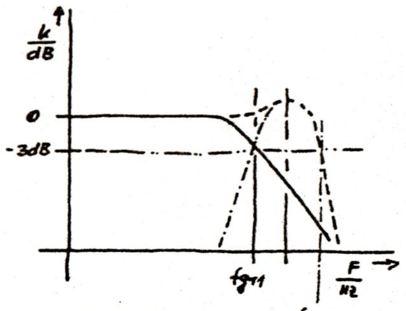


BILD5:

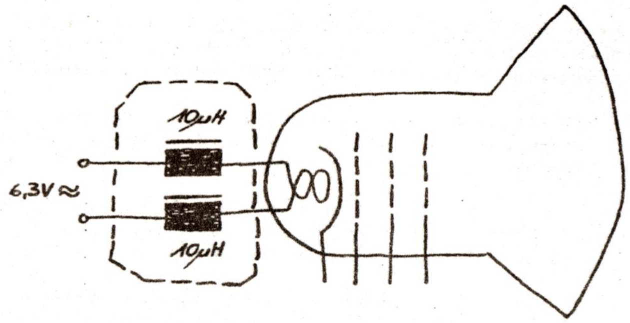


BILD6:

Nebeneffekte und Randbemerkungen

Bevor das unvermeidbare Ende naht, noch einige Anmerkungen. Aufgrund der extremen Bandbreitenerweiterung ist es nicht zu vermeiden, daß sich die Funktion des Kontrasteinstellers verändert hat. Er dient jetzt nur noch dazu, die Verstärkung für die höherfrequenten Signalanteile richtig einzustellen, damit sich ein möglichst linearer Frquenzgang ergibt. Die passende Einstellung ist daran zu erkennen, daß eine Linie gleichmäßig hell ist. Als Kontraststeller fungiert jetzt R1, wobei allerdings die Amplitude nicht beliebig klein werden darf, da sonst die Bildsynchronisierung aussetzt.

Nachdem dann soweit alles fertig war, zeigte sich auf dem Bildschirm ein von oben nach unten verlaufender Graukeil (dunkel oben, heller unten). Nach einigen Suchen und Messen stellte sich C8 mit 1uF als zu klein geraten herraus. Ursache für diese zusätzliche ´Bildinformation´ war der Vertikalaustastimpuls. Er steuerte während des Bildrücklaufs die Videoendstufe soweit zu, daß kein Kollektorstrom mehr fließen konnte. Folge: die Spannung über C8 stieg wegen der fehlenden Belastung auf die Spitzenspannung des Horizontalrückschlagpulses an und verhunzte so den Arbeitspunkt der Bildröhre. Einfachste Abhilfe bestand darin, dafür zu sorgen, daß auch im Bildrücklauf ein Kollektorstrom fließen konnte. Hierfür mußte R10 entsprechend erhöht werden. C8 habe ich durch einen Elko mit 220uF ersetzt, um möglichst dicht an die Spitzenspannung des Rückschlagimpulses herranzukommen, den je höher U_{CE} ist (in Grenzen versteht sich !), umso besser wird die Strahlfocusierung (diese Spannung führt gleichzeitig auf G_2 der Bildröhre = Fokus-Gitter).

Insgesammt kann sich sich das Ergebnis sehen lassen. Ein ´richtiger´ Monitor hätte keine besseren Ergebnisse liefern können. Vielleicht fühlt sich der Ein oder Andere ermutigt, es auch einmal auf diesem Weg zu versuchen. Nur Mut, es kann nicht mehr wie ca. DM 600.- kosten: DM 180.- für das verhunzte TV-Gerät und ca. DM 398.- für den neuen Monitor !

Tschüß

512K-RAM-Adressierung mit flexibler MMU
2 serielle Schnittstellen
2 DMA-Kanäle
2 16-Bit-Timer
Erweiterter Befehlssatz
Höhere Geschwindigkeit

durch Ersetzen des Z80 durch den HD64180

Der Hersteller der HD64180-CPU benötigte zur Darstellung aller Leistungen dieses Z80-aufwärtskompatiblen Prozessors in einem 'Datenblatt' ein 206 Seiten starkes Buch. Deshalb soll hier garnicht erst der Versuch gemacht werden, die Vorteile dieses Chips gegenüber seinem Vorgänger Z80 zusammenzufassen. Hier soll nur dem Bastler eine Möglichkeit gezeigt werden, wie dieser Prozessor im alten Z80-Microcomputer die Regie übernehmen soll.

Für die Unterbringung des HD64180 anstelle des Z80 im Computer wird eine Kleine Adapterplatine benötigt, an die von der Lötseite ein 40poliger Dilstecker angelötet wird und auf der die neue CPU und 5 weitere ICs untergebracht werden. Diese Platine wird in den Sockel des Z80 gesteckt und liefert über den Dilstecker alle Z80-äquivalenten Signale des HD64180 an den Computer.

Auf der Platine sind schon zwei Treiber-ICs enthalten, die bei den Signalen der CPU-internen seriellen Schnittstellen die Pegelwandlungen +12V/-12V => GND/+5V und umgekehrt vornehmen. Der Sendetreiber muß über zwei zusätzliche Leitungen +12V und -12V zugeführt bekommen. Die Signale der seriellen Schnittstellen Können vom Stecker CNA direkt abgenommen werden und z.B. an ein Modem geführt werden.

Auf der Adapterplatine wird der Takt für den Prozessor aus einem internen Oszillator und einem externen Quartz hergeleitet. Am Pin 64 des HD64180 wird die durch 2 geteilte Quartzfrequenz, die dann dem Systemtakt entspricht, bereitgestellt und kann bei Bedarf mit der Drahtbrücke J1 über den Pin 6 des Z80-Sockels an den Computer geliefert werden. Dann muß aber die bisherige Z80-Systemtaktquelle davon abgetrennt werden. Es muß geprüft werden, ob der Takt für den Z80 anderweitig als feste Frequenz verwendet wird. Dann muß dort weiterhin die Taktfrequenz für den Z80 hinggeführt werden. Wenn aber der Takt für die Synchronisation mit der CPU benötigt wird, muß dieser vom Pin 64 des HD64180 entnommen werden.

Außerdem werden mit 3 weiteren ICs aus den zusätzlichen Adreßleitungen A16 bis A18 Freigabesignale für die Ansteuerung von 2 x 8 Stück 41256-RAMs und die an die Pins 1 dieser RAMs zu legende gemultiplexte RAM-Adresse MA8 erzeugt. Je nach Funktionsweise des bisherigen Adreßmultiplexers für vorher 4164-RAMs wird für die Steuerung des Adreßmultiplexers nun MUX oder MUX* vom Stecker CNB verwendet. Für die Wahl zwischen diesen Signalen muß geprüft werden, ob das bisherige MUX-Signal low oder high aktiv war, entsprechend ist dann das low aktive MUX* oder das high aktive MUX zu verwenden. Das ist deshalb von Bedeutung, weil sonst beim Refresh die oberen Adressen an die RAMs durchgeschaltet werden, und dann das Refreshing nicht stattfinden kann.

Das Signal RAS* von CNB kann direkt anstelle des bisherigen RAS*-Signals an die RAMs gelegt werden (eventuell über 33 Ohm). Bei Bestückung des Computers mit nur 8 Stück 41256-RAMs wird nur das Signal CAS0* von CNB benötigt. Dieses Signal wird nicht direkt an die RAMs geführt sondern an den Eingang des Gatters gelegt, mit dem die Verknüpfung mit dem RAM-Freigabesignal erfolgt. Diese Stelle in der Schaltung herauszufinden, ist die einzige Anforderung an den Bastler, der sich die Vorteile des HD64180 für seinen Computer zunutze machen will. Hier muß auch geprüft werden, ob das bisherige CAS-Signal high oder low aktiv an diese Stelle gelangte. Wenn hier ein high aktives CAS nötig ist, muß CAS0* noch über einen Inverter geführt werden.

Wenn die bisherige Herleitung dieser Signale aus dem Systemtakt für den Z80 oder über Gatterlaufzeiten aus dem MERQ*-Signal erfolgte, kann diese weiterbenutzt werden. Bei einer Änderung des Systemtaktes beim Umrüsten kann bei größeren Unterschieden zwischen alten und neuen Takt eine Herleitung von RAS*, MUX bzw. MUX* und CAS* aus MERQ* über Gatterlaufzeiten eventuell nicht mehr funktionieren. Hier muß dann probiert werden, wieviele Verzögerungsgatter zwischen den einzelnen Signalen günstig sind.

Wenn das RAS*-, MUX- und CAS*-Timing bisher aus dem Systemtakt für den Z80 (nicht aus einem dazu vielfachen Muttertakt) hergeleitet wurde, kann diese Schaltung weiterbenutzt werden, indem die bisherige Taktquelle unschädlich gemacht wird und stattdessen der Takt vom Pin 64 des HD64180 über Pin 6 des Z80-Sockels ins System

geführt wird. Eventuelle Treiber des ursprünglichen Systemtaktes zwischen ursprünglicher Taktquelle und Z80 müssen überbrückt werden, um auch Bauteile zu erreichen, die mit dem Takt vor diesen Treiber versorgt werden.

Dann werden nicht die Signale RAS*, MUX bzw. MUX* und CAS* des Adapterboards benutzt. Hier wird dann stattdessen das Signal MUX bzw. MUX* (was immer es dort ist) der bisherigen Schaltung an den Freigabe-Eingang des 74LS125-Treibers für A16 auf dem Adapterboard gelegt. der bisher mit diesen Pin verbundene Pin des 74LS04 muß dann aus der Fassung gebogen werden.

Wenn eine weitere Reihe von 8 Sockeln für RAMs vorhanden ist, oder wenn man das Risiko auf sich nehmen will, 8 weitere 41256-RAMs auf die anderen 8 Stück huckepack zu löten, dann wird der Pin 15 dieser zusätzlichen 8 RAMs mit CAS1* von CNB direkt beschaltet. Beim Huckepack-Verfahren dürfen die Pins 15 der beiden übereinanderesitzenden ICs also nicht miteinander verbunden werden.

Es sei noch angemerkt, daß die meisten 4164-RAMs mit einer 7-Bit-Refreshadresse auskommen, während 41256-RAMs grundsätzlich eine 8-Bit-Refreshadresse benötigen. Es muß also gewährleistet sein, daß A0-A7 beim Multiplexen zusammen durchgeschaltet werden. Bei älteren Geräten ist es wahrscheinlich, daß anstelle von A7 eine höhere Adresse zusammen mit A0-A6 durchgeschaltet wird. In diesem Fall sind A7 und die falsch angeschlossene höhere Adresse an den Multiplexer-Eingängen zu vertauschen.

Ein praktisches Beispiel

Nach diesem allgemeinen philosophischen Prolog, der eigentlich nur dem versierten Bastler prinzipiell zu berücksichtigende Punkte aufzeigt und wenig zum Wie sagt, soll nun an einem praktischen Beispiel gezeigt werden, was zu tun ist.

Das Beispiel ist ein GENIE IIs, dessen Besonderheit darin besteht, daß dort eine Umschaltung des Taktes zwischen 1,77 und 5,3 (8) MHz möglich ist. Wenn die 'LSP'-Taste gedrückt ist, das EPROM eingeschaltet ist und/oder, wenn durch die Software ein Bit in einem Steuerlatch auf 0 gesetzt wird (nach RESET ist dieses Bit immer 0), läuft der Computer nur mit 1,77 MHz. Ein Lesen des EPROMs ist bei höheren Taktraten nicht möglich. Diese Takterzeugung und -umschaltung kann für die HD64180-CPU nicht verwendet werden, mit diesem Prozessor muß man auf eine Taktumschaltung verzichten. Wegen der nach einem RESET selbstauferlegten 4 WAITs dieser CPU ist es aber durchaus möglich, ein EPROM auch bei sehr hoher Taktrate noch zu lesen. Nach Abschalten des EPROMs können auch die WAITs zurückgenommen werden. Der GENIE IIs läuft so mit ausschließlich 9,216 MHz ohne Probleme. Es muß allerdings darauf geachtet werden, daß vor dem Reaktivieren des EPROMs die WAITs wieder eingefügt werden.

Dafür wurde einfach das Adapterboard für die Takterzeugung und -umschaltung entfernt und das 74S74-IC dieses Boards in den Sockel (U18) gesteckt, aus dem das Board entnommen wurde. Der Pin 6 dieses ICs ist aus der Fassung zu biegen. Bei U19 ,74LS125 wurden die Pins 8 und 9 aus der Fassung gebogen und diese Pins des Sockels miteinander verbunden. Damit ist der Teiber zwischen alter (abgehängter) Taktquelle und Z80, Pin6 überbrückt. Auf dem HD64180-Adapterboard werden an den dafür vorgesehenen Stellen ein 18,432MHz-Quartz und die beiden 22pF-Kondensatoren bestückt. Der Pin 64 des HD64180 und der Pin 6 des Steckers für den Z80-Sockel werden durch einen Draht miteinander verbunden.

Beim GENIE IIs ist außerdem zu berücksichtigen, daß durch entsprechende Bits im Systemsteuerlatch zwischen RAM und memory-mapped I/O-Baugruppen innerhalb der unteren 16K umgeschaltet werden kann. Die Decodierung der Freigabesignale dieser Baugruppen berücksichtigt natürlich nur die Z80-Adressen bis A15 und nicht die HD64180-Adressen A16 bis A18. Entsprechend würden diese Baugruppen auch in den entsprechenden Adreßbereichen anderer Banks (physikalische Adressen 10000H-7FFFFH) angesprochen werden.

Um dies zu verhindern, muß in der Adreßdecoder-Baugruppe auf der I/O-Karte anstelle von A15 ein durch OR-Verknüpfung aus 0-64K* und A15 hergeleitetes 0-32K* Signal verwendet werden, das an den aus der Fassung gebogenen Pin 5 von U13, 74LS138 gelegt wird (siehe Abbildung).

Aus geometrischen Gründen läßt sich die Standardversion des Layouts des Adapterboards im GENIE IIs nicht verwenden. Ein für diesen Computer geeignetes Layout ist ebenfalls gezeigt, bei dem aber wegen ungünstigerer Anordnung der Bauteile zueinander sehr viel mehr Drahtbrücken und freie Verdrahtungen nötig sind.

Beim GENIE IIs wird das RAS*-, MUX- und CAS*-Timing aus dem Systemtakt hergeleitet und kann deshalb beibehalten werden. MUX von Pin 1 von U6, 74S157 kann für die Umschaltung von A16 und A17 auf dem HD64180-Adapterboard verwendet werden. Dafür wird durch freie Verdrahtung von Pin 1 von U6 eine Verbindung mit MUX* von CNB des

Adapterboards hergestellt und der Pin 5 des 74LS04 auf dem Board aus der Fassung gebogen.

Auf den speziell für den GENIE IIs gebauten Adapterboard muß die Erzeugung des 0-64K*-Signals noch in freier Verdrahtung nachgearbeitet werden. Dazu stehen die OR-Gatter des 74LS32 (Pins 1,2,3 und 13,12,11) noch zur Verfügung. Das Signal kann über den unbenutzten Pin des Steckers CNB dem I/O-Board zugeführt werden.

Die Sonderausführung des Adapterboards für den GENIE IIs entspricht prinzipiell auch der angegebenen Schaltung. Jedoch wurden teilweise andere Gatter innerhalb der ICs benutzt, so daß die Pin-Nummern in der Schaltung nicht überall zutreffen.

Wieviele WAITs der einzelne Computer beim HD64180 und höherer Taktrate beibehalten muß, ist im Einzelfall zu untersuchen. Es ist durchaus nicht in jedem Computer ein Takt von 9,216MHz möglich, auch wenn alle 4 WAITs bei Memory-Zugriffen beibehalten werden.

Und wie geht es in meinem Computer?

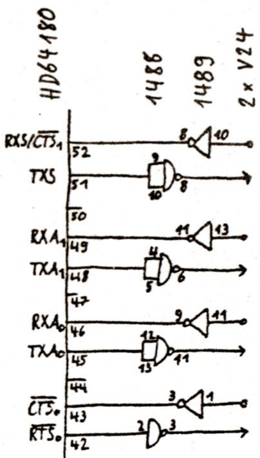
Auf diese Frage hin muß ich nun doch wieder auf die allgemeingültigen Bemerkungen zurückkommen. Ich habe weder das Geld, noch die Zeit, noch überhaupt die Lust, diesen Hardware-Patch in allen mir zugänglichen Computern auszuprobieren, um dann für alle Geräte einfach nachzuvollziehende Lötanleitungen vorzulegen.

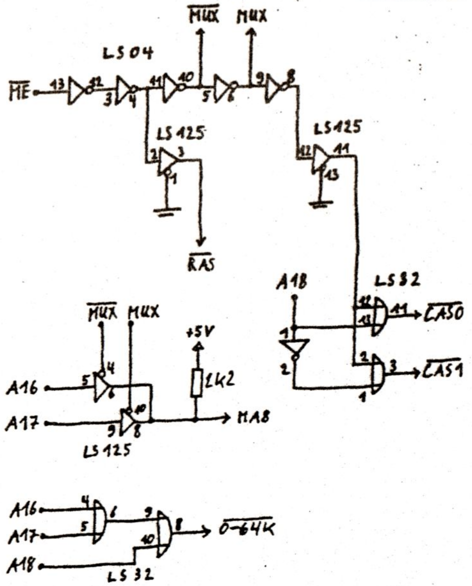
Ein bißchen Abenteuer und Eigenleistung und dadurch auch mehr Spaß an der Sache soll denen bleiben, die sich an dieses Projekt heranwagen. Als Unterstützung kann ich lediglich eine ungebohrte geätzte Platine (einer der beiden vorgestellten Varianten) für 10,-DM zur Verfügung stellen und eventuell fernmündlich oder -schriftlich helfen, soweit ich vom Stand meiner zwangsläufig begrenzten Hobbyistenkenntnisse und der Versorgung mit Schaltplänen der verschiedenen Geräte dazu in der Lage bin.

Über Einbauanleitungen des HD64180-Adapters in spezielle Computer (und dadurch indirekt über Erfolgsmeldungen) im Info würde ich mich sehr freuen.

Helmut Bernhardt

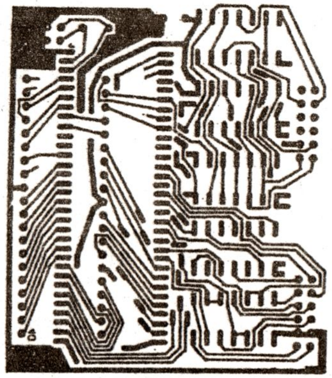
Beschaltung der Pins der seriellen Schnittstellen

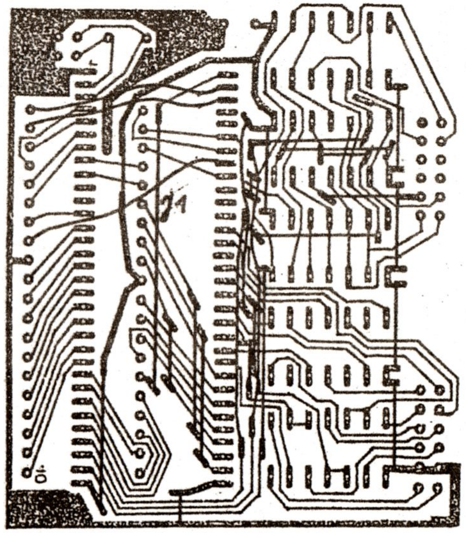


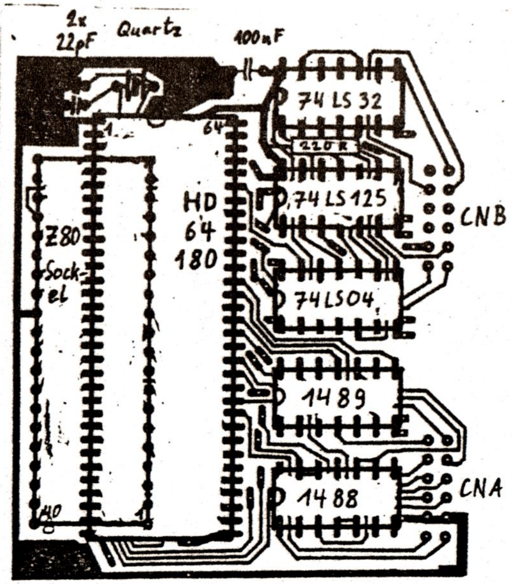


Schaltung zur Ansteuerung von 512K RAM auf dem Board

Standardversion des Adapterboards
Layout, Drahtbrücken und freie Verdrahtung, Bestückung,
Pinbelegung der Stecker CNA und CNB





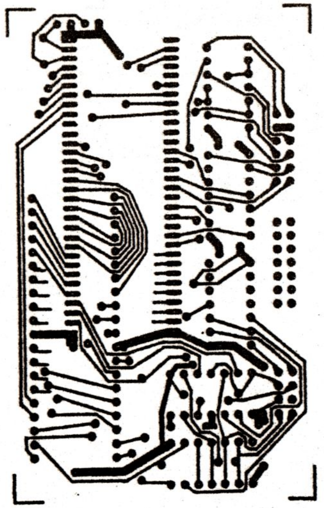


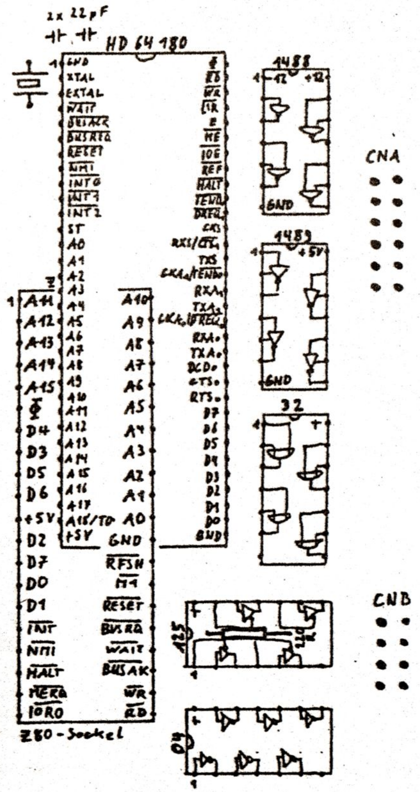
NC	0-64K
CAS1	CASØ
NC	NC
NC	NC
RAS	MA8
MUX	MUX

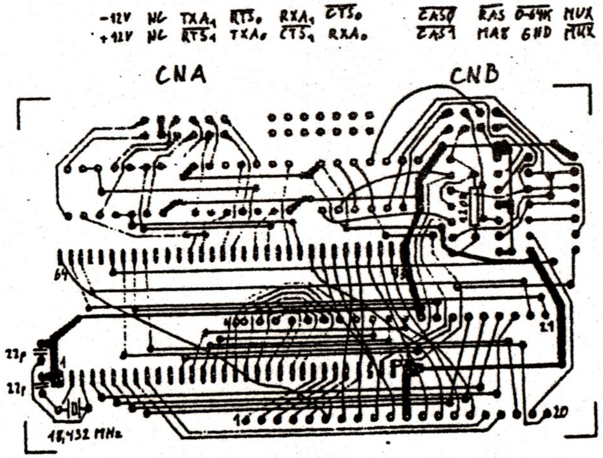
RXA0	RXA1
TXS	CTS1/RXS
CTS0	TXA0
RTS0	TXA1
GND	GND
+12V	-12V

HEFT 18
Februar
1987

Sonderversion für GENIE IIs





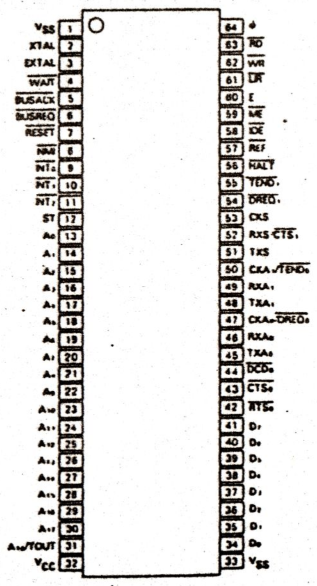


Direkt zu verbindende Pins

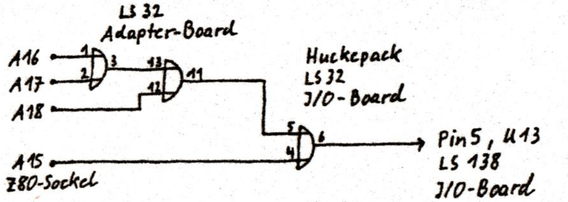
HD64180 Pin	Signal	Z80 Pin
13	A0	30
14	A1	31
15	A2	32
16	A3	33
17	A4	34
18	A5	35
19	A6	36
20	A7	37
21	A8	38
22	A9	39
23	A10	40
24	A11	1
25	A12	2
26	A13	3
27	A14	4
28	A15	5
34	D0	14
35	D1	15
36	D2	12
37	D3	8
38	D4	7
39	D5	9
40	D6	10
41	D7	13
6	BUSREQ*	25
5	BUSACK*	23
7	RESET*	26
8	NMI*	17
9	INT0* = INT*	16
59	ME* = MERQ*	19
58	IOE* = IORQ*	20
63	RD*	21
62	WR*	22
57	REF* = RFSH*	28
12	ST* = M1*	27
56	HALT*	18
4	WAIT*	24
64	PHI	6 #
32	+5V	11
1,33	GND	29

CLOCK ist beim Z80 ein Eingang und beim HD64180 ein Ausgang

Pinout der HD64180-CPU



(Top View)



Herleiten von 0-64K* auf dem Adapterboard der Sonderversion und Verknüpfen mit A15 zur Erzeugung des Signals 0-32K* für den Pin 5 von U13 auf dem I/O-Board

G III s:
ein Jahr alt, 256k-Byte RAM,
mit grünem Monitor und CP/M
2500,--DM

Akustikkoppler
Preis auf Anfrage

Gerald Dreyer

Wer suchte einen ITOH-Treiber für GRAPE ?

Ich habe 4 Stück - weiß aber nicht mehr, wer danach fragte...

K.-J. Mb
Ø62Ø1-55Ø52

BÖRSE -- BÖRSE -- BÖRSE

Ich suche ein Genie, einen TRS-80, einen Komtek o. ä.. Floppies (möglichst 2) und einen Drucker (möglichst Epson-kompatibel). Es braucht natürlich nicht alles aus einer Hand zu stammen, besonders nicht aus einer allzu geldgierigen.

Arnulf Sopp, Tel. 0451-791926

Suche Stecknadel im Heuhaufen!!

Druckernadel vom Epson RX80 abgebrochen. Neuer Druckkopf kostet ca. 250,- DM (ist mir zu teuer). Wer weiß wo man Druckernadeln bekommt oder anderen defekten Druckkopf.

Werner Förster
09721/21841

W E R kann mir bei der Installation eines Laufwerkes helfen??? Ich habe ein QUME 5-Zoll-Laufwerk 542, 40 Track DS, welches ich als LW 3 an mein GENIE III anschließen möchte. Lt. Manual ist es interfacekompatibel mit dem SHUGART SA450-Laufwerk. Neben den zwei im GENIE III fest eingebauten Laufwerken ist seit langer Zeit schon extern ein BASF-Laufwerk Model 6106 als Lw. 2 angeschlossen. Welche Jumper müssen überbrückt werden? Die Beschriftung: HS, DS0-3, MX, HM. Lt. Manual "the Qume Trak 542 is configured from the factory to operate in a single drive system. However, it can be modified by the customer to operate in a multiple drive system". Das Lw. wurde jedoch ohne irgendwelche Jumper in diesem Bereich geliefert, und eine überbrückung von DS3 (+ Umsetzen des terminator resistor von Lw. 2 auf 3) zeitigte keinen Erfolg. Wer kann helfen nach Zusendung des ganzen Laufwerkes, oder nur des Manuals oder einfach durch Rat?
Richard Rensch, Bahnhofstr. 100, 7128 Lauffen, T. 07128-8415

Zum Tandy TRS-80 mod.4 hätte ich einige Fragen :

Beim Auflisten von Basic-Programmen in mod.3 kann ich mit den Pfeiltasten wunderschön Zeile für Zeile das gesamte Programm durchblättern. Leider funktioniert das nicht mehr in mod. 4. Warum ? und wer kann Abhilfe schaffen ?

Betr.Info 12/S.76
Die Auflistung der Befehlsworte gilt wahrscheinlich für TRS-DOS 6.1. Für mein TRS-DOS 6.2 musste ich den Tabellenzeiger auf 26870 setzen, dann lief das Programm. Allerdings fehlte der Befehl CAT :n (Auflistung aller Filenamen).

Betr.Info 15/S.40
das Grafik-Shorty für mod.4/4P läuft bei mir nicht. Durch welches Zeichen muss ich die zwei <ö> ersetzen ?. Ich habe eine französische Tastatur :

```
        1  2  3  4  5  6  7  8  9  0  °  _  #  BREAK
        &  é  "  '  (  6  è  !  ç  à  )  -  $
        -----------------------------------------
        A  Z  E  R  T  Y  U  I  O  P     *
        a  z  e  r  t  y  u  i  o  p     @
        -----------------------------------------
         Q  S  D  F  G  H  J  K  L  M  %  ENTER
         q  s  d  f  g  h  j  k  l  m  ù
        -----------------------------------------
SHIFT    W  X  C  V  B  N  ?  .  /  +  SHIFT  CLEAR
          w  x  c  v  b  n  ,  ;  :  =
        -----------------------------------------
CTRL    <         S P A C E            CAPS
        >
        -----------------------------------------
```

Wie und mit welcher Hilfe kann ich mod.4 von 64k auf 128k aufstocken, um MEMDISK vernünftig anwenden zu können.

Was für einen Assembler/Disassembler gibt es für mod.4 und wo kann man ihn kaufen ?

Helmut Obscherningkat

Tandy schließt Computer-Center

RATINGEN (PW) — Die Tandy Corp. hat drei der insgesamt sechs Computer-Center in Deutschland geschlossen. Die Begründung lautet: PC-Kunden sind zu 95 Prozent Großkunden, die am Arbeitsplatz aufgesucht werden wollen und nicht mehr in die Geschäfte kommen.

Laut Tandy-Geschäftsführer Georg Michielin erwartet die deutsche Tochter „trotz oder wegen der Schließung" einen Anstieg des Umsatzes um 20 Prozent im laufenden Geschäftsjahr.

Wie in den USA seien auch in Deutschland die MS-DOS-PCs Hauptumsatzträger. Ihr Anteil betrage 30 Prozent. Dagegen sei der Anteil von Maschinen mit Tandy-Betriebssystem auf 10 Prozent gesunken. Der Umsatz im Bereich Zubehör trage 60 Prozent zum Gesamtergebnis bei.

Ein Teil der Mitarbeiter der in München, Stuttgart und Düsseldorf geschlossenen Computer-Center werde nun im Außendienst zur Betreuung von Großkunden eingesetzt. Die rund 50 unabhängigen Händler seien von diesen Maßnahmen nicht betroffen.

Branchenbeobachter rechnen damit, daß die texanische Tandy in Kürze grundlegende Änderungen in der Struktur des Konzerns bekanntgeben wird, die sich auch in Deutschland auswirken könnten.

Es wird erwartet, daß Tandy ihre einzelnen Divisions und Geschäftsbereiche klarer trennt und den Schwerpunkt bei den Computern setzt. Mitte vergangenen Jahres gab es im Unternehmen bereits Überlegungen, das internationale Geschäft der Tandy-Tochter „Intertan" anzugliedern, die dann die rund 2200 eigenen Niederlassungen und Franchise-Läden außerhalb der USA übernehmen sollte.

TRACE oder die Geschichte einer Zeitschrift

Hier bei mir in Frankreich gab es eine Menge von TRS-Anwendern. Sie nannten Ihre Maschine, nach französischer Art, zärtlich Therese. Das Interesse war so gross, dass unabhängig von bestehenden Clubs, ein Verleger eine wunderbare Zeitschrift herausgab. Sie hiess T R A C E und sollte alle Tandy- und Genie-Anwender ansprechen. Und das tat sie auch. Es gab eine Menge von Programmlistings und viele praktische Ratschläge. Das Interesse war so gross, dass man sich bald entschloss, jedem Heft eine Diskette mit allen im Heft veröffentlichten Programmen beizulegen.

Wie auch in anderen Ländern, kam dann die böse Konkurrenz, Tandy zog nur sehr schwerfällig mit und war mit einmal aus dem Rennen. Der ganze Aufwand war umsonst und die schöne Zeitschrift TRACE musste nach nur acht Exemplaren 1984 ihr Erscheinen einstellen.

Warum ich das schreibe ? Nun bei mir liegen noch die acht Hefte, die sich allerdings nur auf die Tandy-mod. 1,2 und 3 beziehen, die aber vielleicht doch noch maches enthalten, was im Club80 nicht bekannt ist. Deshalb stelle ich der Redaktion gerne Hefte und die zwei Disketten mit allen Programmen zur Verfügung.

Nach Durchsicht bin ich dann gerne bereit, den einen oder anderen Artikel zu übersetzen oder auch mal ein Progamm umzuschreiben. Sagt mir nur, an wen ich das Paketchen senden soll.

Helmut Obscherningkat

„Mailbox" ist der Renner

Für Computer-Freaks ein heißer Draht

Bonn. Seit gut sieben Monaten ist „Mailbox" am Netz, seit gut sieben Monaten rufen Tag für Tag rund einhundert Computerfreunde im Bonner Streitkräfteamt an, um per „Mailbox", einem digitalisierten Informationssystem, das Allerneueste über die Streitkräfte zu erfahren. Wer einen Heimcomputer mit Akustikkoppler und Telefonanschluß hat, der erreicht „Mailbox" unter der Rufnummer (0228) 628516. Die Parameter sind: 300 Baud, 8 Datenbits, 1 Stopbit, keine Parität, Code IBM ASCII. „Mailbox" bietet Tag und Nacht eine Menge Service: So erfährt man die neuesten Termine von Tagen der offenen Tür, Ausstellungen der Streitkräfte und Konzerten der Bundeswehr-Big-Band. Diese Veranstaltungstermine werden wöchentlich aktualisiert.

Milliarden-Aufwand für das Mega-Projekt

Bei Siemens heißt es Chip, Chip, hurra!

Von unserem Redaktionsmitglied Dieter Keller

Klein, aber oho – so läßt sich der Mega-Chip charakterisieren, dessen Massenfertigung jetzt im neuen Siemens-Werk in Regensburg angelaufen ist. Mit dem bloßen Auge betrachtet ist er äußerst unscheinbar: ein nicht einmal fingernagelgroßes goldglänzendes Fleckchen, für den praktischen Einsatz mit Anschlußkontakten versehen und in ein schwarzes Plastikgehäuse verpackt. Genauso sehen auch Chips aus, die viel weniger können. 1 Million Informationseinheiten – oder 1 Megabit – können auf dem 1-M-Chip gespeichert werden. Das ist der Inhalt von rund 64 Schreibmaschinen-Seiten.

Etwa alle 3 Jahre hat sich in der Vergangenheit die Leistungsfähigkeit der Chips vervierfacht – und das zum gleichen Preis. Schafften die modernsten 1971 erst 1000 Bit oder 1 Kilobit, ist derzeit der 256-K-Chip der leistungsfähigste mit 256 000 Bit. Bei der Technik war Siemens 2 Jahre im Rückstand gegenüber den Japanern und Amerikanern. Um diese Lücke aufzuholen, startete der deutsche Konzern Anfang 1984 das „Mega-Projekt". 1,7 Mrd. DM an Investitionen und 800 Mio. DM für Forschung und Entwicklung läßt sich der Elektrokonzern die Aufholjagd kosten. Der Forschungsminister schießt 240 Mio. DM zu. „Wir sind kosten- und zeitmäßig im Plan", meint Siemens-Chef Dr. Karlheinz Kaske. „Wir haben das Gröbste hinter uns", fügt Megaprojekt-Leiter Friedrich hinzu.

Auf dem Halbleiter-Markt hat es allerdings im letzten Jahr weltweit einen Einbruch gegeben. Folge: Die Konkurrenz macht mehr Tempo bei der Entwicklung neuer Produkte. Siemens befürchtete, der 1-M-Chip könnte 1 Jahr früher als gedacht auf den Markt kommen. Daher gingen die Münchener eine Kooperation mit Toshiba ein, besorgten sich bei den Japanern die erforderliche Technologie. Beim 4-M-Chip, viermal so leistungsfähig, wollen dann die Europäer aber alleine mit vorn dabei sein. Dazu forscht Siemens gemeinsam mit Philips.

Noch schwieriger als der Entwurf des Chip sind die Fertigungsmethoden. Denn der 1-M-Chip erfordert eine Strukturfeinheit von 1,2 Tausendstel Millimetern, kaum vorstellbar klein, und beim 4-M-Chip wird's noch winziger. Entsprechend schwierig ist es, die Massenfertigung in Gang zu bringen. 300 Arbeitsschritte sind erforderlich, und rund 2 Monate dauert es, bis ein 1-M-Chip fertig ist – trotz Schichtbetrieb rund um die Uhr. Noch ist die Ausschußquote hoch. Von 80 v. H. muß sie auf unter 50 v. H. gebracht werden, bis die Auslieferung im Sommer beginnt. Dann sollen im Monat 1 Million 1-M-Chips die Fabrik verlassen.

Gefertigt wird in Reinsträumen, die so sauber sind, daß ein Operationssaal dagegen ein Hort des Schmutzes zu sein scheint. Das kleinste Staubpartikel könnte einen Chip unbrauchbar machen. Die Eheringe müssen abgelegt werden, an Make-up ist nicht zu denken.

Und wozu das Ganze? Würden es die alten Chips nicht auch noch tun? Und könnte man die nicht auch von anderen Firmen beziehen? Auf die letzte Frage heißt es bei Siemens, integrierte Schaltungen – kurz Chips genannt – seien die Schlüsseltechnik für moderne Geräte und Systeme, mit denen der Elektrokonzern sein Geld verdient. Wichtig ist es, die Herstellverfahren zu beherrschen, um neben Standardspeichern anwendungsorientierte Chips höchster Komplexität und Leistungsfähigkeit herstellen zu können, die ganz speziell auf eine bestimmte Aufgabe abgestellt sind, etwa für die Daten- und Kommunikationstechnik, die Fertigungs- und Prozeßautomatisierung.

Viele Dinge werden zumindest zu vertretbaren Kosten erst mit sehr leistungsfähigen Chips möglich. Ein 256-K-Chip kostet derzeit 4 bis 4,50 DM, beim 1-M-Chip wird zunächst mit 40 bis 50 DM gerechnet. Beispielsweise läßt sich das Flimmern des Fernsehbildschirms wegbekommen, indem es laufend digital gespeichert wird. Das darf aber nur 20 bis 30 DM kosten. Zu schaffen ist das erst mit dem 4-M-Speicher, der in den Labors schon gefertigt wird. Hierfür hat Siemens gerade in München-Perlach ein neues Gebäude mit 4000 qm Reinsträumen gebaut. Alleine das Gebäude hat 200 Mio. DM gekostet, 400 Mio. DM die Geräte. Die ersten Muster des 4-M-Chip sollen Ende 1988 ausgeliefert werden, die Massenfertigung soll 1989 laufen.

Auch danach ist die Grenze des Möglichen noch nicht erreicht. Siemens ist gerüstet, daß der 3-Jahres-Rhythmus weitergeht. Am 16- und 64-M-Chip wird schon geforscht. Als eine damit mögliche Funktion führt Projektleiter Friedrich ein direkt übersetzendes Telefon an: Auf der einen Seite wird etwa deutsch hineingesprochen, auf der anderen Seite kommt es japanisch heraus.

* * * Neue Eingänge in der "Diskettothek" * * *

9 ganze Eingänge sind seit November 1986 zu verzeichnen, die nicht mehr im neuen Programmkatalog berücksichtigt werden konnten, weil dessen Manuskript bereits bei Peter SPIESS zum Druck vorlag. Sie sind nebenstehend aufgeführt.
Vier stammen von Richard RENSCH, drei von Gerald SCHRÖDER. Ich hoffe, daß die dreizeiligen Kurzkommentare, die ich den Programmen verpaßte (s.d.) "nicht nur notwendig, sondern auch hinreichend" sind (wie sich der ordentliche Mathematiker ausdrückt).

Der achte Titel (LOAD801984) ist ein Paket von 75 Programmen aus der 80 MICRO 1984, das Andreas RYCHLIK freundlicherweise einsandte - allerdings ohne jeglichen Kommentar. Es war mir nicht möglich, diese Programme zu studieren, um sie zu kommentieren. Nur wenige davon konnte ich mit meinem MODEL 1 lesen. Damit ihr eine annähernde Vorstellung bekommt, was sie be-inhalten, hier das Inhaltsverzeichnis (sog. "Directory") dieser Spezialdiskette, die auch nur als Ganzes kopiert wird (bitte merken!):

BEN/CMD	BSP3D/BAS	CROSS/BAS *)	DRIVER/MAC
EQUAPLOT/BAS	EUROPA/GMF	EXEC/CMD	FNDX/BAS
FORMGN/BAS	FRIEND/BAS	FX80RB/DRV	GERMAN/FNT *)
GRAPE/CMD *)	GRAPE21/CMD *)	HIRES/DRV	ITOH8510/DRV *)
ITOHPL/DRV *)	KWORD/BAS	LOAD0384/BAS	LOAD0384/JCL
LOAD0484/BAS	LOAD0584/BAS	LOAD0584/JCL	MAIN1/CMD
MAIN1/SRC	MARQUEE/CMD	META/DRV	MFI/REL *)
MICROTAB/L2X	MPH/CMD	MSG	MSG/BAS
MUSIKON/BAS	NUKE/L2X	PARI01/BAS	PARI02/L2X
PARI03/L2X	PARI04/CMD	PARI04/SRC	PIE/BAS
PLAYBYTE/CMD	PLOT/BAS	PLOT/DRV	PLOTTER/MAC
PROJECT2/CMD	PROJECT4/CMD	SCREEN/BAS	SINSTEP/CMD
SKAL/BAS	SPLAN/BAS	SPOOL/CMD	SPOOL/SRC
SPOOL1/CMD	STNDRD/BAS	TEXT1/CMD	TEXT1/SRC
TEXT2A/SRC	TEXT2B/SRC	TEXT3A/L2X	TEXT3A/SRC
TEXT3B/CMD	TEXT3B/SRC	TEXT3C/CMD	TEXT3C/SRC
THREAT/BAS	TTSPRG/CMD	TTSPRG1/CMD	UTIL/BAS
VCSORTER/BAS	VOTRAX/BAS	WORD	WORD/BAS
WORDCHKR/BAS			

*) nicht aus 80 MICRO [gehört zu GRAPE]

(Anm.: Wie man beim Schreiben mit TSCRIPS einen solchen Directory-Ausdruck direkt von der Diskette in den Text übernimmt, hat euch ARNULF Als Rechter Nachdenker Unter Lieben Fellows ja vor einiger Zeit in einem INFO verraten, wißt ihr noch?)

Wer Lust hat und womöglich sogar über den Jahrgang 1984 der 80 MICRO verfügt, möge sich der sozialen Mühe unterziehen, diese Programme anzuschauen und zu kommentieren. Er wird dann auch in der Diskettothek namentlich verewigt!

Der neunte Titel schließlich stellt zwei Grafikprogramme vor, mit denen ich erstmals Anwendung von M.WINTERs HRG--Programm "GRAPE" gemacht habe. Man kann den Zeichenstift beim Zeichnen dieser nicht alltäglichen algebraischen Kurven dritter Ordnung beobachten. Wenn man sie mit '-SAVE("CARTBLAT")' gespeichert hat, bekommt man sie mit dem Ladebefehl '-LOAD("CARTBLAT")' auch sofort als Ganzes auf den Bildschirm; die Beobachtung ihrer Entstehung finde ich aber interessanter. Ein bißchen Geduld braucht man eh'...

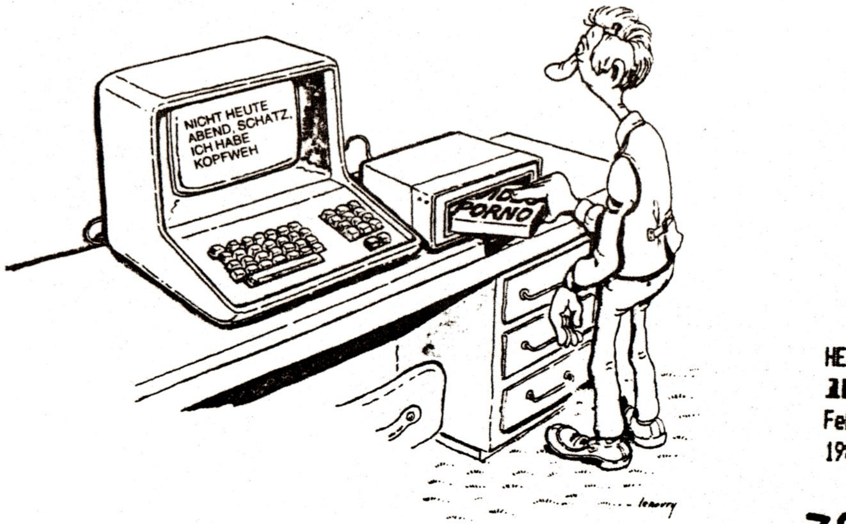
Das "Erlebnis" mit dem LOAD80-Programmpaket veranlaßt mich zur Wiederholung einer dringenden Bitte:

1) Wer Programme einsendet, versehe sie bitte auch mit einem Kommentar zu deren Inhalt und Zweck. Der "Diskettothekar" - wer auch immer es ist - kann sie nicht immer kommentieren - sei es, daß er sie gar nicht abfahren oder nicht einmal lesen kann (z.B. wenn für ein nicht voll kompatibles Modell gedacht) oder aus anderen Gründen!
2) Daß die Nennung von Quelle und Verfasser eine Selbstverständlichkeit ist für Leute, die sich (satzungsgemäß!) verpflichtet haben, niemals ein Copyright zu verletzen, sei hier auch nochmals vermerkt! Nichtbeachtung dieses Grundsatzes versperrt den Eingang in die Diskettothek.

Zu meinem Erstaunen ist von unserer umfangreichen "PUBLIC DOMAIN"-Software mit rund 250 Programmen noch kaum Gebrauch gemacht worden. Ich möchte daher alle, die es wohl noch nicht mitbekommen haben, über welchen Schatz der Club da verfügt, darauf hinweisen, daß sich das ausführliche Inhaltsverzeichnis dieser 24 Disketten im INFO Nr.15, September 1986, befindet. Ich empfehle, es herauszutrennen und extra bereit zu legen.

Erfolg mit und Freude an unserer "Diskettothek" (diese treffende Bezeichnung prägte Mitglied Richard RENSCH)

wünscht allen nah und fern,
den Hackern groß und klein
mit dem "Z80-Stern":
Klaus-Jürgen Mühlenbein



Programm-Eingaenge seit INFO Nr.16 (November 1986)

```
ADDLINE/BAS   UTL Progr 0368  Teile: 01   / Grans: 03  Disk 10
===========        f.Comp. alle           Quelle CHIP 10/86,S.239
Verfasser R.Krenzke & R.Rensch  eingetippt von Richard Rensch
Hiermit kann an ein laufendes BASIC-Programm eine Erweiterg.
angehaengt oder nicht belegte Zeilennummern eingeschoben
werden, ohne die Variablen zu loeschen.

DIRDRUCK/BAS UTL Progr 0369  Teile: 01   / Grans: 04  Disk 10
============       f.Comp. alle           Quelle
Verfasser Richard Rensch        eingetippt von Richard Rensch
Das Directory in einem beliebigen Laufw. wird einschl. Kopf=
zeile, die naehere Angaben enthaelt, unter Beruecksichtigung
opt.Parameter (I, /ext (ausser A)) gedruckt

FILEMOD/BAS   UTL Progr 0370  Teile: 03   / Grans: 18  Disk 10
===========        f.Comp. alle           Quelle
Verfasser Richard Rensch        eingetippt von Richard Rensch
ASCII-Files koennen auf einen String beliebiger Laenge durch
sucht werden; dieser kann gegen einen anderen String
ausgetauscht werden.

NOSCROL3/BAS UTL Progr 0371  Teile: 01   / Grans: 01  Disk 10
============       f.Comp. Mod.III        Quelle
Verfasser Richard Rensch        eingetippt von R.Rensch
Eine waehlbare Anzahl Zeilen kann am oberen Rand des
Bildschirms vor dem Scrollen (Aufwaertsrollen) des Bild=
schirms geschuetzt werden.

HANOITRM/BSR SPL Progr 0372  Teile: 03   / Grans: 20  Disk 10
============       f.Comp. GENIE/TRS      Quelle
Verfasser Gerald Schroeder      eingetippt von dto.
Loesung in Maschinensprache.- Aufruf mit "HANOI xx" (xx=Anz.
d.Scheiben in Hex). - Naeheres auf Textfile. (Siehe auch
INFO Nr.17 od.18 <z.Z.noch nicht bekannt> )

SIDEKICK/BSR UTL Progr 0373  Teile: 03   / Grans: 20  Disk 10
============       f.Comp. TRS 80         Quelle
Verfasser Gerald Schroeder      eingetippt von dto.
Sidekick fuer TRS80 mit Helmut Bernhardt's Banker. Man kann
4 Computer nebeneinander benutzen. Umschaltung: SHIFT DOWN
ARROW Ziffer 0-3. Sh.Art.INFO17od.18(zZ ? )

HD64180/SRC  DOS Progr 0374  Teile: 02   / Grans: 09  Disk 12
===========        f.Comp.                Quelle
Verfasser Gerald Schroeder      eingetippt von dto.
Befehlserweiterungen fuer den neuen Prozessor HD64180
mit Textfile "ILLEGAL/TXT", das die "Illegals" behandelt.


LOAD801984   DIV Progr 0375  Teile: 56   / Grans: 00  Disk 14
==========         f.Comp. TRS80          Quelle 80MICRO "LOAD80"
Verfasser verschiedene          eingetippt von  (216 Granules)
Verschiedene LOAD80-Programme aus 80MICRO 1984, eingesandt
v.A.Rychlik Dez.86.- Inhaltsbeschreibungen liegen nicht vor.
Disk also kompl.anfordern (enth.auch GRAPE)

CARTBLAT/GRP GRF Progr 0376  Teile: 04   / Grans: 22  Disk 13
============       f.Comp. GENIE/TRS1     Quelle
Verfasser K.-J.Muehlenbein      eingetippt von KJ. Muehlenbein
Zwei HR-Grafikprogramme, die eine algebr.Kurve 3.Ordnung in
zwei versch.Winkelpositionen zeichnen. Ausfuehrlicher Kommen-
tar in INFO 18/87 (etwas fuer Mathematiker)
```

Ja, ja, ich weiß! Er kann kalkulieren, analysiert den seelischen Zustand eines jeden Menschen nach den Regeln des Karma, sagt das Wetter für nächstes Wochenende auf Sylt voraus, komponiert eine Symphonie, übersetzt 783 Sprachen von Arabisch bis Zulu, spricht Recht nach den Gesetzen des Fürstentums Thurn & Taxis aus dem Jahr 1735, kontrolliert die Klimaanlage einer Mine, untersucht alle Informationen, die zur Verfügung stehen und kontrolliert jedes Projekt dieses Planeten – aber was kann er für **mich** tun?!

- I M P R E S S U M -

1. Vorsitzende

Peter STEVENS
Postfach 56
4600 Dortmund 1
☎ 0231 /593883

2. Vorsitzende

Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

Hardwarekoordinator

Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar

Klaus-Jürgen MÜHLENBEIN
Am Mönchgarten 28
6940 Weinheim -Lü.
☎ 06201 /55052

Redaktion

Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren

Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck

Peter Spieß
Trugenhoferstraße 27
8859 Rennertshofen 1
☎ 08434 /454

Bankverbindung des CLUB 80

Postgirokonto	Peter STEVENS
	Sonderkonto CLUB 80
Konto-Nummer	285 491 - 465
Postgiroamt	Dortmund
BLZ	440 100 46

Das INFO erscheint zweimonatlich.
Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

SCHLUSS

Hallo Club-80er,

wie Ihr seht habe ich es gerade noch geschafft das 18. Club-INFO vor der Hauptversammlung fertigzustellen. Dadurch haltet Ihr auch gleichzeitig die Tagesordnung für die Zeit unseres Treffens in den Händen.
Angemeldet haben sich bis jetzt:
Arnulf Sopp, Bernd Retzlaff, Eckehard Kuhn, Gerald Schröder, Hans Raggan mit Frau, Hans-Martin Stephan, Helmut Bernhardt, Herbert Albers, Jens Neueder mit Freundin, Jutta Obermann, Jürgen Wucherer, Klaus Hermann, Manfred Held, Thomas Buskowiak, Ulrich Böckling mit Frau und Walter Piller mit Frau

Besonders gut finde ich, daß dieses Jahr Clubmitglieder über verschiedene Themen referieren werden. Das ist mit Sicherheit ein Grund mehr sich für eine Teilnahme am Clubtreffen zu entscheiden. Ich hoffe, ich treffe recht viele von Euch in Alsfeld. Platz, für alle die noch kommen mögen, ist vorhanden.

Als nächstes muß ich zu unserem Clubkalender kommen. Hier ist mir leider ein Fehler unterlaufen. Die Tages- und Wochenzuordnung für das Jahr 87 stimmt, aber bei der überschrift des Kalenders passierte das Malheur -es sollte doch als Jahresangabe 1987 erscheinen und nicht 1986.
Macht der Gewohnheit. Man hat halt am Anfang des neuen Jahres die Zahl des Vergangenen erst so richtig drauf. -Sorry.

Abschließend möchte ich noch einige Erkärungen zu unserer Inhalts-übersicht der INFO's kommen. Im folgenden Text kurz "IüI" genannt. Sie liegt, wie im letzten INFO versprochen, dieser Ausgabe bei und beinhaltet alle Titel der erschinenen Artikel der Clubinfo's 1 bis 18. Der Stand dieser "IüI" ist somit Februar 87.
Leider sind von INFO 1 bis 5 keine Seitennumerierungen angebracht. Aus diesem Grund habe ich mir erlaubt, die fehlende Seitennumerierung für diese INFO's, nach meinen INFO's orientiert, zu verteilen. Ich bitte Euch nun, die Seitennumerierung zu den Artikeln, wie in der "IüI" dargestellt, nachzutragen. Erst dann hat unsere "IüI" für alle Club-INFO's seine Gültigkeit.
Weiterhin kann es sein, daß ich den einen oder anderen Artikel, da er nicht bei den INFO's abgeheftet ist, vergessen habe aufzunemmen. Für Tips über fehlende oder falsche Einträge wäre ich Euch dankbar.
Die Aufmachung und Handhabung der "IüI" entspricht der unseres Programmkataloges. Das "IüI" wird jeweils zum Jahreswechsel aktualisiert.

So, nun viel Spaß an der neuen INFO, der "IüI" und ungestörte Stunden am Computer wünscht Euch Euer
Jens

P.S.: Ich hoffe man trifft sich zum Treffen !!

CLUB 80 Mitgliederadressenliste

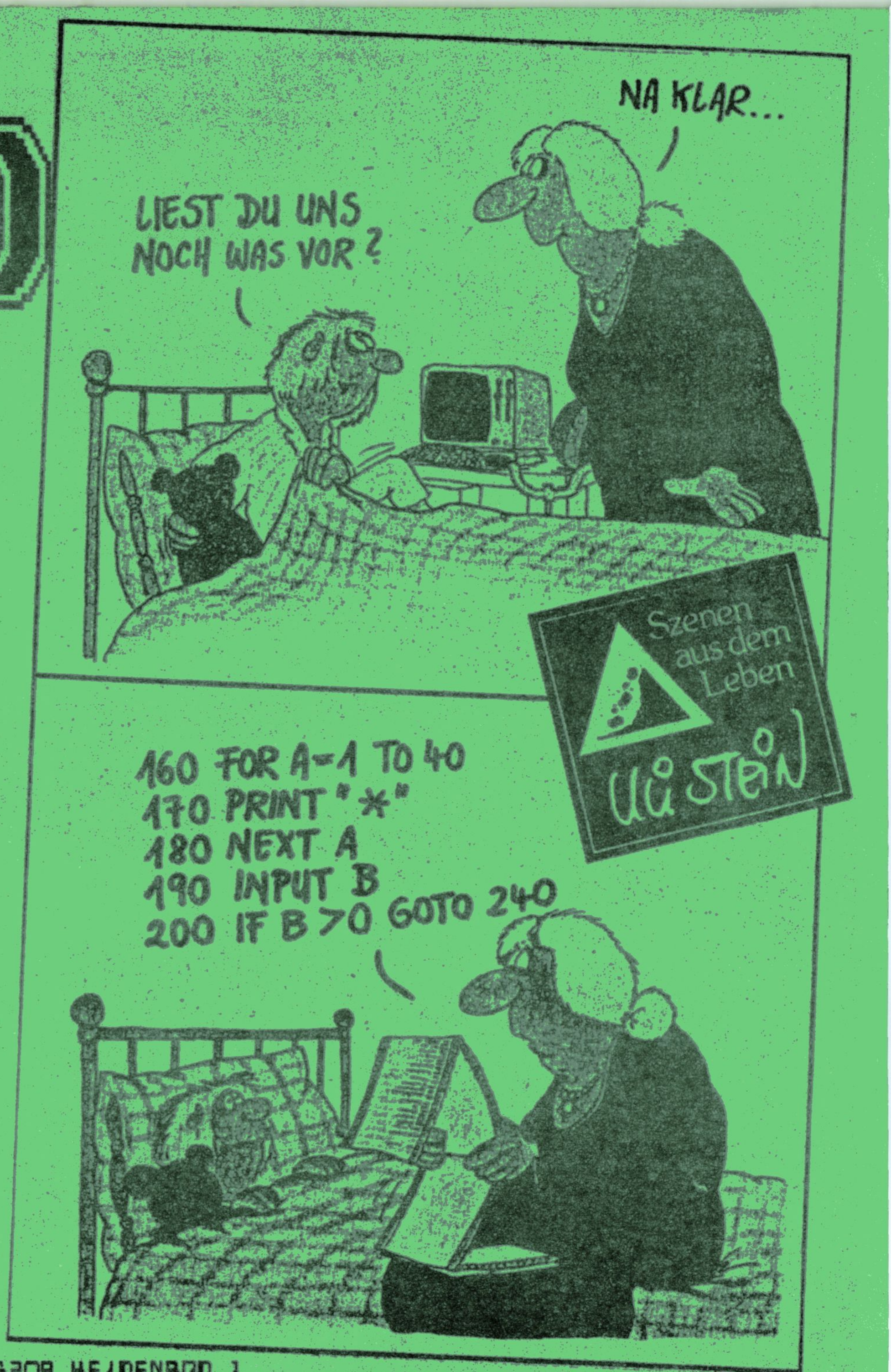
Name	Vorname	Straße	PLZ	Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907	// 0431 /74047
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621	// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441	Mettringen	05233 /4320	// 02557 /1236
Emmerich	Helmut	Waldstraße 5	6682	Ottweiler	06824 /4114	// -
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721 /21841	// 09721 /51256
Grajewski	Werner	Zedernweg 29	4220	Dinslaken	02134 /54573	//
Gromottka	Uwe	Lange Reihe 40	2803	Weyhe	0421 / 88496	// 0421 /300-6870
Heile	Heinz-Dieter	Marx Straße 70c	4320	Hattingen	02324 /67495	// -
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024	//
Hill	Peter	Bergstraße 65	6754	Otterberg	-	// -
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948 /542	// 02921 /70431
Kopschina	Peter	Strandallee 138	2409	Scharbeutz	-	// -
Krüger	Karl-Herbert	Bruchweg 65	4920	Lemgo	05261 /13686	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417	// 07022 /77442
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668	// -
M[illegible]sioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /107945
M[illegible]hlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083	// 04103 /702662
N[illegible]ueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913	// -
O[illegible]scherningkat	Helmut	1 RUE DES BRUYERES	F-68360	Soultz	0033089/762690	// -
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418	//
[illegible]aggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611	// 0711 /2630473
[illegible]ank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /3791	//
Reichelt	Dieter	Philipp-Schmitt-Straße 30	6902	Sandhausen	06224 /52906	// -
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551	// 04103 /7025310
R[illegible]chlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383	// 0203 /331383
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221 /707044	// -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631	// 08025 /41247
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207204	//
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434 /454	// 08431 /7041684
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585	// 05401 /30096
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883	// 0231 /593883
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131 /32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285	//
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361	// -
[illegible]acherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686	// -

Stand: Februar 1987

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

CLUB 80

CLUBINFO DER TANDY - GENIE - UND KOMTEK - ANWENDER



19. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
TEL.: 06124 / 3913

- INHALTSVERZEICHNIS -

Seite:
und Autor:

Clubinternes

Neues vom Vorstand .... 1 - 2
Hartmut Obermann

Protokoll der Mitgliederversammlung 3 - 4
Helmut Bernhardt

Der neue Zweite 5
Gerald Schröder

Adressenänderung 5
Hartmut Obermann

Zehnzeiler-Wettbewerb 6
Redaktion

Clubwappen gesucht 7
Gerald Schröder

An alle Clubmitglieder 8
Eckehard Kuhn

Termine / Messen 9
Redaktion

Diskussion 10
Hans-Martin Stephan

Software

Grafik-Standart: allein geht's auch 11 - 12
Gerald Schröder

CRAMER und die Determinanten .... 13 - 15
Klaus-Jürgen Mühlenbein

INDEX-Dateien 15
Klaus Hermann

TSCRIPS & die &-Codes .... 16 - 18
ASCII-Tabelle immer zur Hand .... 19 - 20
Arnulf Sopp

BIVARANA .... 21 - 25
Klaus-Jürgen Mühlenbein

BASIC - programmierbare Funktionen 26
Klaus Hermann

contra .... 27 - 28
Arnulf Sopp

Visicalc-Sortieren .... 29 - 31
Klaus-Jürgen Mühlenbein

Ergänzung zu Grafik-Shorty 32
Klaus Hermann

Von der Wiege bis zur Bahre .... 33 - 37
Nachtrag zu "Von der Wiege ..." ... 38
Noch mehr Fernsehen für's Geld 39
Futter für den CRTC .... 40 - 41
Arnulf Sopp

Nachtrag zu Programm DISASSEM M4 .. 42
Boot-ROM Version und Datum M4 42
Klaus Hermann

Password-Bypass bei TRSDOS 6.2 42
RESET-feste MEMDISK M4-CP/M .... 43 - 44
Hartmut Obermann

Seite:
und Autor:

Hardware

Indikator-Karte für ECB-Bus .... 45 - 46
Peter Spieß

Floppycontroler Kompatibilität ..... 47 - 48
Bernd Retzlaff

80-40 Track Umschaltung 49
Eckehard Kuhn

Doubler mit WD2793 49
Helmut Bernhardt

Joystick am Model 4/4p 50
Hartmut Obermann

Z80-Bus-Anschluß .... 51 - 56
Universalgenie .... 57 - 60
Kleines Logik-Lexikon .... 61 - 62
HCT kontra LS in der Praxis .... 63 - 65
CMOS kontra TTL .... 66 - 69
Der "Leidensweg" der CMOS-Chips .... 67 - 68
Nanohenry blockieren den Fortschritt 70
Artikel aus c't

Börse

Wer hat was -- wer will was .... 71 - 74

Sonstiges

DER - DIE - DAS macht mir Spaß ..... 75 - 77
Klaus-Jürgen Mühlenbein

Ein neuer 80-Zeichen-Treiber........ 77
Gerald Schröder

CP/Mac - na und ? 78
Arnulf Sopp

Tandy Vertragshändler 79
....

Datenverluste 80
Artikel aus c't

Computerkriminalität 81
Makler war faul 82
Kollege Blaumeise .... 83 - 86
BDOS-Tip 87
Comparex hat Premiere 87
BASF mit neuer Diskette 88
Comparex rechnet sich gute Chancen aus 89
Tag der offenen Tür 91
Pascal (Verschiedenes) 92
aus verschiedenen Zeitschriften

Programmbibliothek

Der neue Diskettothekar .... 93 - 94
Werner Förster

Die letzten Seiten

Impressum 95
Schluß 96
Clubmitgliederadressen .... am INFO-Ende
Redaktion

Neues vom Vorstand

Aus alt mach neu!

Wie ihr sicher schon im Protokoll der Mitgliederversammlung gelesen habt oder noch lesen werdet, hat man mich, trotz heftiger Gegenwehr und in Abwesenheit wieder zum Vorsitzenden des CLUB80 gemacht. Ich habe die Wahl, nach der Benachrichtigung durch meine bessere Hälfte, die mich auf dem Clubtreffen vertreten hat, mit einem lachenden und einem weinenden Auge angenommen. Ersteres weil ich froh bin, daß der Club nicht mangels Vorstand "eingegangen" ist (und zum Vorstand gehören ja immerhin noch mehr Leute!), letzteres, weil ich doch lieber für eine gewisse Zeit ohne offiziellen Posten geblieben wäre. Die Freude überwiegt aber deutlich, und so stütze ich mich mit viel Schwung in die alte, neue Aufgabe!

Zunächst möchte ich einmal all denen danken, die zum guten Gelingen des Clubtreffens beigetragen haben. Besonders zu nennen sind dabei Arnulf Sopp, Helmut Bernhard und Gerald Schröder, die ihre Vortragsthemen (so wurde mir aus verschiedenen Quellen berichtet) exzellent an den Mann brachten! Ich hoffe, daß sich auch für das nächste Treffen wieder Leute finden, die über ähnlich interessante Themen referieren.

Weiterhin möchte ich an dieser Stelle meiner Frau Jutta ganz offiziell dafür danken, daß sie mich "besser vertreten hat, als wenn ich selbst anwesend gewesen wäre!", wie mir unser Freund Kajott schrieb. Danken möchte ich auch Klaus Jürgen Mühlenbein und Peter Stevens, den "Aussteigern" des Vorstandes, für ihre Arbeit als Diskothekar bzw. 1. Vorsitzender.

Last not least begrüße ich die beiden Vorstandsneulinge Gerald Schröder (2. Vorsitzender) und Werner Förster (Softwareverwalter), die sich haben breitschlagen lassen, einen Teil der regelmäßig anfallenden Arbeiten im Club zu übernehmen.

Regionaltreffen!

Während des Clubtreffens wurde u.a. beschlossen mindestens noch zwei Regionaltreffen, eins für "Nordlichter" und eins für "Südländer" zu veranstalten. In diesem Zusammenhang hier gleich eine Information zum Regionaltreffen "nord", welches für den 25./26.04.87 geplant war. Wie ihr schon an der Formulierung "geplant war" erkennt, fällt es leider aus!!! Ein neuer Termin wird so bald als möglich bekanntgegeben.

Schau - trau keinem?

In seinem, im letzten Info erschienen Beitrag "Trau - schau, wem! (Ein BITchen Englisch)", hat unser Freund "Kajott" Mühlenbein ein Thema angeschnitten, welches wohl mehr Mitglieder interessiert, als man dies auf den ersten Blick glauben möchte. Nicht etwa, daß es hartnäckige Hardwarebastler groß interessiert, wenn Bernd Drowälder aus seinem Floppy-Controler einen Floppy-Controller (man achte auf die zwei l!) macht, die wissen sowieso was gemeint ist und werden ihm kaum seine Kompetenz absprechen, bloß weil ihm ein Tippfehler unterlaufen ist (oder?). Vielmehr hat dieser an sich gut gemeinte Beitrag zur Verbesserung unserer (Allgem)Einbildung wohl eher einige Leute, die gerade Ihre Passivität abstreifen, sich hinter die Tastatur klemmen und mal einen Beitrag für das Info schreiben wollten, in ihrem Glauben bestärkt, dies doch lieber bleiben zu lassen!

Leute laßt euch nicht ins Boxhorn jagen! Zumindest ich lese lieber einen fachlich versierten Beitrag mit Tippfehlern, als mich in Bälde mit einem Clubinfo zu begnügen, welches außer Deckblatt und Impressum nichts mehr zu bieten hat. Also schreibt, ohne Angst vor (Tipp-)Fehlern (ich tue es auch), überwindet die Scheu und liefert den Zweiflern den Beweis: "Zwei statt einem 'l' führen noch lange nicht zur fachlichen Disqualifikation eines Beitrags!!!", oder was meint ihr???

CLUB 80-Bibliothek

Noch immer fehlen einige der Bücher, die beim Clubtreffen '86 in Holzhausen ohne Eintragung in die ausgelegte Liste ausgeliehen wurden. Ich glaube inzwischen nicht mehr daran, daß sich die Bücher noch einmal wiederfinden. Leider gehören auch so wertvolle (fachlich) und teure Publikationen wie die "Microsoft BASIC decoded" und "51 BASIC-Dialekte im Vergleich" zu den Verlusten!
Ich werde für das nächste Info eine aktuelle Liste der Bücher erstellen, die man sich bei mir ausleihen kann. Von diesem Angebot wird übrigens viel zu wenig Gebrauch gemacht, was sich hoffentlich nach Erscheinen der neuen Liste ändern wird!

Damit genug für dieses Mal! Alles Gute und viel Spaß beim Hacken und Löten wünscht allen

Hartmut Obermann

„Er hat einen Magneten verschluckt!"

Protokoll der Mitgliederversammlung des Club 80 am 14.03.87

Haupttagesordnungspunkt war die Bildung eines neuen Vorstandes. Der derzeitige erste Vorsitzende Hartmut Obermann, der aus beruflichen Gründen an der Mitgliederversammlung selbst nicht teilnehmen konnte, ließ über seine ihn bestens vertretende Frau ausrichten, daß er für diesen Job nur dann weiter bereitstehen würde, wenn er von den Clubmitgliedern die Zusage erhielte, daß er an ihn gerichtete Fachfragen an Mitglieder weiterreichen dürfe und die sich dieser Probleme annähmen.

Der Antrag von Arnulf dazu, daß sich die anwesenden Mitglieder dazu verpflichten, vom Vorstand übertragenen Fachfragen zu beantworten oder an entsprechende für kompetenter gehaltene Mitglieder weiterzuleiten, wurde einstimmig angenommen. Und schon war Hartmut Obermann einstimmig zum neuen ersten Vorsitzenden gewählt worden.

Die Wahl des Clubdiskothekars verlief etwas schleppender, weil in diesem Zusammenhang auch das Problem der Diskformate erörtert wurde. Diese Aufgabe sollte von einem Mitglied versehen werden, dem möglichst alle Formate zugänglich sind. Für den Austausch zwischen Mitgliedern und Software-Sammlung des Clubs sollen aber trotzdem zwei Formate als Standard verbindlich vorgegeben werden (40 Track, SS, SD und 80 Track, DS, DD). Der neue Diskothekar wird die PDRIVEs im nächsten Info bekanntgeben und dann auch sonstige Modalitäten erläutern.

Werner Förster, der sowohl 40- als auch 80-Track Drives besitzt, erklärte sich für diese Aufgabe bereit und wurde einstimmig gewählt.

Die Wahl des Vorstandes wurde dann kurz unterbrochen, weil Walter Piller inzwischen die Kassenprüfung abgeschlossen hatte.

Den Einnahmen (incl. übernahme aus dem Vorjahr) von 3.333,89DM standen Ausgaben von 2.741,30DM gegenüber. Es blieb ein überschuß von 592,59DM. Mit den üblichen Dankesworten und freundlichen Redewendungen wurde der alte Vorstand einstimmig entlastet.

Für den Posten des zweiten Vorsitzenden wurden Bernd Retzlaf und Gerald Schröder vorgeschlagen. Bei 3 Enthaltungen und 3 Gegenstimmen wurde Gerald Schröder neuer zweiter Vorsitzender.

Und schließlich wurden Jens Neueder als Redakteur unseres Infos und Eckehard Kuhn als Hardwarekoordinator in ihren Ämtern bestätigt.

Die Sorge, der Club könnte eventuell keinen neuen Vorstand mehr zusammenstellen, war damit aus der Welt, so daß nun ohne diesen Druck die weiteren Punkte diskutiert werden konnten.

Zum Thema Regionaltreffen oder häufiger ein allgemeingültiges Clubtreffen wurde ein eleganter Kompromiß gefunden: Außer der Jahreshauptversammlung, die in geographisch zentraler Lage durchgeführt wird, sollen nun auch künftig je ein offizielles Clubtreffen im Süden und im Norden der BRD stattfinden, an denen aber jeder teilnehmen kann, auch wenn er nicht in dem Teil der BRD ansässig ist. Diese zusätzlichen Clubtreffen sollen aber keinen beschlußfassenden Charakter haben.

Von Jutta Obermann wurde stellvertretend für Hartmut der Antrag gestellt, daß die Vorstandsmitglieder auch nicht nachgewiesene Kosten (z.B. Telefongespräche) abrechnen dürfen. Wenn bei der sich anschließenden Diskussion nicht der dann auch einstimmig angenommene Antrag, daß solche Kosten prinzipiell ersetzt

werden und der Vorstand sich intern über die Modalitäten einigen soll, gestellt worden wäre, würde die Hauptversammlung wahrscheinlich heute noch tagen.

So blieb aber noch Zeit, auch noch das Thema "Fusion mit dem Bremerhavener Club" anzusprechen. Die dabei abzusehenden Schwierigkeiten bestehen darin, daß wir ein Club mit einer Satzung und beabsichtigter Eintragung in das Vereinsregister sind, während der Bremerhavener Club ein lockerer Zusammenschluß von TRS80- und GENIE-Usern ist. Da der Namen und die Satzung unseres Clubs beibehalten werden sollen, hängt der Zusammenschluß davon ab, ob der Bremerhavener Club sich dem anschließen will. Entsprechend ließ sich darüber auch kein Beschluß fassen, weil der Vorsitzende des Bremerhavener Clubs, der dafür eingeladen worden war, nicht gekommen ist.

Zum Schluß wurde noch darauf hingewiesen, daß beim letzten Clubtreffen aus der Bibliothek Bücher ausgeliehen und nicht zurückgegben wurden, daß das 10-Zeiler-Preisausschreiben nun beendet wird und daß ab jetzt solche Leute anonym unterstützt werden sollen, die für Beiträge im Clubinfo ihrer Rechtschreibung nicht ganz trauen, oder glauben, daß jemand anderes den Sachverhalt vielleicht besser formulieren könnte.

Das Thema "gestaffelte Clubbeiträge je nach Aktivität im Vorstand oder im Info" wurde noch kurz angesprochen, fiel dann aber dem stärkeren Interesse für den Computer und für das Bier zum Opfer.

Helmut Bernhardt

Anwesenheitsliste:

Walter Piller, Manfred Held, Eckehard Kuhn, Thomas Buskowiak,
Klaus Hermann, Helmut Bernhardt, Klaus-Jürgen Mühlenbein,
Bernd Schneider, Werner Förster, Ulrich Böckling, Bernd Retzlaff,
Gerald Schröder, Herbert Albers, Jutta Obermann (i.V.), Arnulf Sopp,
Jens Neueder

Wir hätten uns alle gefreut, wenn wir noch weitere Clubkameraden beim Clubtreffen angetroffen hätten. Vielleicht klappt es ja das nächste Mal.

Die Redaktion

Der neue Zweite

Moin, Dschungs! Meinereiner wurde ohne Vorwarnung auf dem Clubtreffen zum 2. Vorsitzenden gewählt. Da Ihr sicherlich davon genauso überrascht seit wie ich, werde ich mich wohl vorstellen müssen.

Mein Name ist Hase und mein Motto lautet: Traue keinem über 30. Daraus könnt Ihr messerscharf schließen, daß ich gerade 20 geworden bin und damit zu den Jüngsten im Club gehöre. Meine digitale Laufbahn begann in der Schule mit den alten CBMs und einem Pet. Mein erstes eigenes Gerät war ein Genie I, das mich auch diesem Club zuführte, als dieser gerade gegründet worden war.

Inzwischen residiert bei mir ein Genie IIs mit diversen Veränderungen von Helmut und beruflich strebe ich die einzig logische Laufbahn an, indem ich in Hamburg Informatik mit Nebenfach Soziologie im 2. Semester studiere. Zu meiner Schande muß ich gestehen, daß inzwischen ein Atari ST den Hauptteil meiner Korrespondenz übernimmt, aber ich gelobe feierlich, den alten Z80 nicht ganz zu vergessen. Beweise werden im Clubinfo vorgelegt.

Große Wahlversprechen habe ich mir verkniffen (Spenden an der richtigen Stelle tun's auch) und somit brauche ich auch nichts zu halten. Der Big Boss ist Hartmut und ich werde schön artig machen, was er mir aufträgt. Da, wie Ihr sicher wißt, Hartmut beruflich wie clubintern stark engagiert ist, könntet Ihr ihm einen großen Gefallen tun, wenn Ihr mir Eure Fragen stellt (falls Ihr mir deren Beantwortung zutraut).

So, nun hab' ich alles gesagt. Meine Adresse steht in der Mitgliederliste.

Ollns klor?

Gerald Schröder

Adressenänderung!!!

In der letzten Juni...
Birkenfeld an d...
neue Adr...
laut...

Hartmut's Adressenänderung ist
hinfällig --- Es bleibt wie's ist !!!

Telefonabstimmtermin
Der 22.06.87 wird nicht zu halten sein.
Darum ein neuer Termin für diesen Tag
Do 25.06.87
Sicher ist das INFO bis zu diesem Termin bei Euch.

...en. ...ier ...ag bei Ein ...eilen, daß der ...ungen, die an mich ...erichtet sind, meine alte ...nen. ohne Verzug

Hartmut Obermann

Zehnzeiler-Wettbewerb

Es darf abgestimmt werden!
Nachdem unser Zehnzeiler-Wettbewerb nun schon über ein Jahr läuft, wird jetzt zur Abstimmung geschritten. Ich hoffe, Ihr beteiligt Euch recht rege daran.
Bevor ich Euch die Abstimmodalitäten erkläre, möchte ich zuvor die Zehnzeiler, in der Reihenfolge ihres Erscheinens, hier kurz auflisten:

a) DEZ/HEX-Umwandlung	von Klaus-J. Mühlenbein
b) Grafik-Hardcopy-Programm	von Herbert Albers
c) BASIC-pur-Treiber für HRG1b	von Arnulf Sopp
d) Spline-Interpolation	von Dieter Kasper
e) arabisch/römische-Zahlenumwandlung	von Klaus-J. Mühlenbein

Nun zur Abstimmung.
Da eine Ausarbeitung eines Abstimmzettels mir im Moment zu viel Arbeit bereiten und bei Euch gleichzeitig die Portokasse belasten würde, habe ich mich einmal zu einer Telefonaktion entschlossen. So kommt Ihr mit ca. 2-einhalb Groschen günstig weg (solange Euch nicht zu einem längeren Gespräch zumute sein sollte) und ich habe dann an zwei Abenden die Abstimmung zusammen.

Abgestimmt wir dann wie folgt:
Anrufen bei der Redaktion
Nennung Eures Namens und Abgabe der + / - -Stimme
Ein + für das Programm, daß Jhr als das Beste anerkennt
Ein - für das Programm, daß Eurer Meinung nach weniger gut ist

Alsdann ist die Abstimmung beendet
Wer wem welche Stimme gegeben hat wird nicht bekanntgegeben.
Die Namensangabe dient nur zu meinem überblick.
Die Bekanntgabe der Gewinner folgt unter Angabe der + / - -Punktestände für das jeweilige Programm in dem kommenden INFO.

HEFT 19
Juni
1987

__Telefonabstimmtermine:__
Montag der 22. Juni 1987 und
Samstag der 27. Juni 1987 jeweils von 19.00 bis 22.00 Uhr

Club-Wappen gesucht

Als ich mein Einstandsschreiben als 2. Vorsitzender an den 1. Vorsitzenden schrieb, fragte ich ihn, ob ich nun nur noch mit Tandy-Krawatte auftreten dürfe. Diese etwas respektlose und nicht im geringsten ernst gemeinte Bemkerkung führte zu folgender Idee:

Warum nicht ein **Club-Emblem** einführen?

Jede noch so unbedeutende Gruppe aus Schlamm-Robben (sprich: Bundeswehrangehörigen) hat ein Emblem, jedes Kuhdorf einen Löwen im Wappen, warum nicht auch unsere Mini-Lobby der Elektronik-Industrie?

Also suchen wir (***alle* Clubmitglieder und zwangsweise zur Mitarbeit verpflichteten Familienangehörigen**) ein Emblem, das uns repräsentiert, kennzeichnet, markiert und auch angemessen veräppelt. Der letzte Punkt sollte meines Erachtens nicht unterschätzt werden, aber Ihr dürft darüber anderer Meinung sein.

Also laßt Eure Phantasie mal schweifen. Alles ist erlaubt, was von unserem Clubredakteur nachgemacht und gefälscht werden kann. Aus drucktechnischen Gründen wäre eine schwarz-weiße Version wohl am günstigsten. Das Emblem sollte klar und treffend uns bzw. unsere Gedächtniskrücken charakterisieren. Hier einige Beispiele aus unserem Leben: der abgemagerte Rumpfadler, auf jedem Markstück in Lebensgröße zu bewundern; das Edelweiß der Gärtner der Nation, auch fälschlich Gebirgsjäger genannt; das gelbe Horn aus der glorreichen Vergangenheit der Bundespest; das niedliche Strichmuster auf der Stirn aller Statistiker (s. auch: Barcode); das Zeichen der Volkszähler: der Beamten-Abakus (für alle, die nur bis drei zählen können).

Wenn Ihr einen oder auch mehrere Vorschläge habt, schickt sie an:

Gerald Schröder
Am Schützenplatz 14
2105 Seevetal 1

Es wäre schön, wenn ich im nächsten Clubinfo schon über erste Erfolge berichten könnte. Natürlich werde ich **nicht** über Eure Vorschläge richten, sondern den gesamten Vorstand anrufen. Im Zweifelsfall werden die besten (oder alle) Embleme veröffentlicht und wir stimmen ganz demokratisch ab.

Also: ran an den Bleistift.

Gerald Schröder

An alle Clubmitglieder

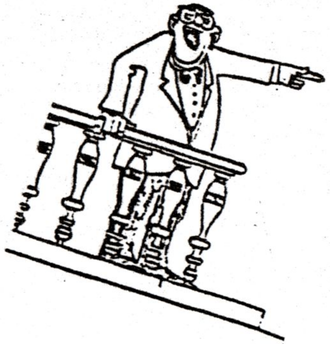
Da unser ECB-Bus Projekt wie Ihr in diesem Info lesen könnt, nun schon beachtliche Fortschritte gemacht hat, möchte ich hier eine Umfrage starten, die sofort beantwortet werden sollte.
Um Erweiterungen Koordinieren zu können brauche ich von euren Computern die Adressen der belegten Ports.
Als Beispiel gebe ich hier die belegten Ports des Computers von Manfred Held an.

Port	Art der Belegung
0-5	HRG von RB
9-F	Prommer
A0-A1	Grip2
A2-A3	Tastatur Port
A4-AF	Uhr
B0-B2	Pal-Brenner
E8	Drucker
EC-EF	Floppy-Controller
80-87	RS232 von RB
40-7F	HD 64180 Intern
E0-E3	Floppy Select
C0	CP/M

Nun schaut in euren Unterlagen nach und antwortet mir bald, da ich im nächsten Info das Ergebnis veröffentlichen will. Ich hoffe nicht nur, daß Ihr mir schreibt sondern ich erwarte es.

STICHTAG 22.JUNI 87

Eckehard Kuhn



-- Termine -- Termine -- Termine --

Anruftage für die Abstimmung des Zehnzeiler-Wettbewerbs:
Montag 22. Juni 1987
Samstag 27. Juni 1987 jeweils von 19.00 bis 22.00 Uhr

Nächster Redaktionsschluß ....30. Juni 1987

Norddeutsches Regionaltreffen .... ???????? 1987

A c h t u n g :
Das Regionaltreffen der "Nordlichter" verschiebt sich!
Ein neuer Termin wird nach Absprache bekanntgegeben!!!

-- Messen '87 --

HOBBY ELEKTRONIK 87 ....Stuttgart 5. - 8. November 1987
PRODUCTRONICA ....München 10. - 14. November 1987

Diskussion

Diskussion

Diskussion

Diskussion

Schreibt Eure Meinung zu diesen Themen !!

Lieber Jens,

zu der aufgeflammten Diskussion über den Zusammenschluß der verschiedenen TRS 80-User-Clubs in der Bundesrepublik möchte ich auch noch meinen Senf geben.

Es ist den Ausführungen von Helmut Bernhardt im INFO des Bremerhavener Clubs eigentlich nichts mehr hinzuzufügen. Trotzdem möchte ich noch eine Idee in eine andere Richtung beisteuer.
Im September '86 belegte ich einen Informatik-Kurs an der Volkshochschule in Osnabrück. Der verwendete Rechner, ein IBM-PC, und das MS-DOS konnten mir allerdings keine besondere Achtung abnötigen, außgenommen vielleicht die grafische Darstellung am Bildschirm, die um eine vielfaches feiner und auch kontrastreicher ist als bei unserem Rechner. Das änderte sich ziemlich schlagartig, als sich ein Bekannter einen Schneider-PC zulegte. (Am Schneider finde ich eigentlich außer dem Preis nicht viel Gutes!) Aber Programmpakete wie Lotus 1-2-3 oder Symphonie machten mir schmerzlich die Grenzen unseres System deutlich!!! (Obwohl der Bekannte erst im Versuchsstadium war und nur an der Oberfläche der Software herumkratzte.)
Diese integrierten Programmpakete sind allem, was ich bislang kennengelernt habe um Klassen überlegen. Von so einer Kapazität und Bedienerführung können wir bei SUPER oder AIDS III nur träumen, um nur einige Beispiele zu nennen! Seitdem ringe ich hart mit mir, aber es ist schon abzusehen, daß ich mir über kurz oder lang einen IBM-kompatiblen Rechner anschaffe! Mein Genie werde ich deshalb nicht verschrotten, aber ich würde es besonders schade finden, wenn der Background eines Clubs wie dieses Club 80 fehlen würde. Mein Vorschlag ist deshalb:
Den Club 80 sozusagen aufwärtskompatibel machen und eine MS-Dos Ecke (oder Unterabteilung) eröffnen. Das würde meiner überzeugung nach auch die leidige Fluktuation eindämmen!
Ich wäre Dir dankbar, wenn Du meinen Gedanken im nächsten INFO zur allgemeinen Diskussion stellen würdest. Bestimmt kommen noch viele Meinungen zusammen, die ein dauerhaftes überleben des Clubs zu ermöglichen.

In Punkto Selbsbaurechner müßte ich dann noch mal mit Dir reden, ob eventuel ein Ausstieg noch möglich ist, vielleicht rufst Du mich mal an? (Am besten bei mir zu Hause, weil ich nämlich ab dem 1.VI. Urlaub habe.

So, das wars

Grafik-Standard: allein geht's auch

Als ich im letzten Info zum großen Aufraffen bei der Gestaltung eines Grafik-Standards aufrief, war das Ergebnis schon klar: mäßiges Interesse. Die wenigen Zuschriften waren sowohl herzlich mitfühlend wie auch skeptisch. Also habe ich den nächsten Schritt allein gemacht: ein Grafiktreiber für die HRG 1b steht. Wer sich dafür interessiert, kann ihn bei unserem Bibliothekar oder bei mir bestellen. Noch ist er nur für Assembler-Programmierer (oder solche, die es werden wollen) interessant. Hier eine kurze Beschreibung.

Grundsätzlich ist es egal, wo sich der Treiber im Speicher befindet. Ich halte es für die beste Lösung, ihn in das HIMEM zu legen und dieses dann neu setzen zu lassen. Da für verschiedene Geräte auch verschieden lange Treiber geschrieben werden müssen, wird dann der verfügbare Hauptspeicher kleiner oder größer. Das sollten die allgemeingültigen Programme aber berücksichtigen können.

Einsprung ist bei ***jedem*** Treiber FFF0h. Hier steht dann ein JP xxxxh, der zu dem jeweiligen Grafik-Treiber verzweigt. In FFF3/4h steht die maximal mögliche X-Koordinate, also die horizontale Auflösung minus 1. In FFF5/6h steht die maximale Y-Koordinate. Bei der HRG 1b sind dies 383 (dezimal) und 191 (dezimal), die von jedem Standard-Programm über z.B. LD HL,(FFF3h) für max-x abgefragt werden können. Der Ursprung (0;0) des Koordinatensystems soll übrigens immer unten links auf dem Bildschirm liegen, so daß der Punkt oben rechts die Koordinaten (max-x;max-y) hat.

Der Treiber selbst verändert kein Register und macht keine eigenen Fehlermeldungen. Tritt ein Fehler auf (falls z.B. die X- oder Y-Koordinate zu groß ist), wird das Carry-Flag als Fehler-Kennung gesetzt.

Die Parameter für CALL FFF0h müssen in den Registern A, HL und DE abgelegt sein. Dabei gilt:

```
A = 0: Punkt x/y abfragen (POINT)
    1:     "      setzen    (SET)
    2:     "      löschen   (RESET)
    3:     "      invertieren (INVERT)
    4: ganzen Bildschirm setzen (SET ALL)
    5:    "        "     löschen (RESET ALL)
    6:    "        "     invertieren (INVERT ALL)
    7: wenn L <> 0: Grafik-Bildschirm einblenden
            L  = 0:     "        "     ausblenden
```

Bei den Aktionen 0-3 müssen in HL die X- und in DE die Y-Koordinate des angesprochenen Punktes stehen. Bei Aktion 0 gilt der Punkt als gelöscht, wenn nach dem CALL das Zero-Flag gesetzt ist. Bei NZ ist der Punkt gesetzt.

Wie gesagt: dieser Treiber liefert die Grundfunktionen für die HRG 1b. Treiber für andere Geräte müssen ganz anders programmiert werden, aber die gleichen Resultate bei CALL FFF0h bringen. Ich habe einige Standard-Routinen geschrieben, die auf jedem (TRS-80-kompatiblen) Rechner mit jeder Grafik laufen, soweit ein Treiber zur Verfügung steht, der die oben beschriebenen Funktionen erfüllt. Diese Standard-Routinen können als Bibliothek betrachtet und beliebig in andere Programme eingebaut werden.

Die Standard-Routinen umfassen bis jetzt: Linien zeichnen, Kreise zeichnen, Spiegeln (einer Dot-Reihe oder eines Rechtecks; vertikal oder horizontal), Schrift Files von Diskette laden und diese Schriften halbiert, normal groß oder beliebig vergrößert per Grafik ausgeben. Bis jetzt stehen zwei Schrift-Files zur Verfügung (BLOCK und SCRIPT), aber es dürfte keine Schwierigkeit sein, auch alle DOTWRITER- oder TSCRIPS-Image-Files umzustricken.

Diese Sachen sind bis jetzt nur für den Assembler-Programmierer oder -Anfänger interessant. Ein Einbau in das Basic ist auf vielfältige Art (USR oder per neuen Befehlen) möglich, aber unsinnig, solange noch nicht genug Standard-Routinen existieren.

Wer sich nach diesem Anfang für die Weiterarbeit an dem Projekt interessiert, kann sich bei mir melden.

Gerald Schröder

50% von 100. Moment mal, das haben wir gleich . . .

CRAMER und die Determinanten

Eine Determinante ist ein geordnetes quadratisches Zahlenschema, das einen numerischen Wert darstellt. Die Anzahl Zeilen (Z) = Anzahl Spalten (S) nennt man die "Ordnung" der Determinante.

Ihren Wert erhält man, wenn man das Schema mit einem bestimmten Algorithmus behandelt. Diese Behandlung kann allerdings langwierig und umständlich werden und deshalb auch noch mit Fehlern behaftet sein, so daß es sich anbietet, diesen Algorithmus zu programmieren.

Hier ist ein solches Programm zur Berechnung von Determinanten dritter oder vierter Ordnung. (Ein Programm für höhere bzw. beliebige Ordnung wäre recht kompliziert. Eine spannende Aufgabe für Hobby-<BASIC>-Programmierer; dem, der sie löst, würde ich einen Preis aussetzen, wenn sich die Selbständigkeit der Lösung beweisen ließe...!)

Der Nutzen der Determinantenrechnung ist vielfach. Eine der beliebtesten und einfachsten Anwendungen ist die Lösung linearer Gleichungssysteme nach der *CRAMERschen Regel*. Hierbei wird das System zunächst nach seinen Unbekannten (X) so geordnet, daß diese mit ihren Koeffizienten auf der linken Seite stehen und die reinen Konstanten auf der rechten Seite der Gleichungen. Dann bilden die Koeffizienten eine Matrix, und zwar eine quadratische, weil die Anzahl der Gleichungen gleich der Anzahl Unbekannter sein muß; diese Gleichheit ist ja die Voraussetzung für die Lösbarkeit des Systems. Falls weniger Gleichungen als Unbekannte vorliegen bzw. ermittelt werden können, gibt es "unendlich viele Lösungen" (i.a. keine brauchbare); wenn es mehr Gleichungen sind, ist das Problem "überbestimmt" und i.a. ebenfalls nicht lösbar - es sei denn, es sind nicht alle Gleichungen voneinander "linear unabhängig"; dies ist eine weitere Voraussetzung.

Eine solche quadratische Matrix kann als Determinante (durch senkrechte Striche markiert) gedeutet und dementsprechend berechnet werden (wie - das würde hier zu weit führen. Wen es interessiert, der schreibe mir; ich erkläre es gern ausführlich.)

Sonderfall: Wenn die Koeffizienten-Determinante den Wert Null hat, kann die *CRAMERsche Regel* allerdings nicht angewandt werden (das Programm meldet das!); es muß dann mit dem üblichen (meistens umständlicheren) *GAUSZ'schen Algorithmus* gearbeitet werden. Dieser Sonderfall ist aber nicht allzu häufig.

Das Programm ist sehr einfach zu handhaben. Außer der Lösung von linearen Gleichungssytemen mit 3 oder 4 Unbeklannten bietet es die Möglichkeit, eine Determinante 3. oder 4.Ordnung zu berechnen, da diese Algorithmen ja für die Anwendung der *CRAMER'schen Regel* eingearbeitet sind. Sie sind als Unterprogramme für sich ansprechbar.

Überhaupt lohnt es sich m.E., sich den Aufbau meines Programmes einmal genauer anzusehen: Es folgt streng den Regeln der sog. *strukturierten Programmierung* - obgleich es "nur" ein BASIC-Programm ist! Wer da behauptet, man könne in BASIC nicht strukturiert programmieren, hat nur eine Ahnung von BASIC - nämlich die, daß BASIC sehr leicht dazu verführt, chaotisch zu programmieren, weil es den Programmierer eben nicht zwingt, ein Flußdiagramm mit klarer Struktur zu entwerfen. *Man muß sich eben s e l b e r zwingen!* Insoweit behaupte ich:

Nur "Hobby" ist die Programmier-Bemühung?
Nein, sie ist rationale Selbsterziehung!
(OK - noch mehr gilt dies für's Assemblieren, brauch' drüber drum kein Wörtchen zu verlieren...)

Viel Erfolg wünscht beim "Bemühtsein"
immer euer: *Jalot Mühlbein!!*

Lösung eines linearen Gleichungssystems mit 3 oder 4 Unbekannten
einschl. Berechnung von Determinanten 3. oder 4. Ordnung

```
10 CLS:CLEAR100:PRINT"          Lineares Gleichungssystem mit 3 oder 4
Unbekannten
20 PRINT"      einschl. Berechnung von Determinanten 3. oder 4. Ordnun
g
30 PRINT"             <C>  Klaus-Jürgen Mühlenbein, Weinheim, Mai 1987":P
RINT"     "STRING$(58,"#"):PRINT
40 DEFINTI,J,K,N,S,W,Z
50 PRINT"       . <1>       Determinante 3.Ordnung
      <2>                 "CHR$(34)"       4.   "CHR$(34)"

      <3>       Lin. Gleichungssystem mit 3 Unbekannten
60 PRINT"         <4>             "CHR$(34)"            "CHR$(34)"           "CHR
$(34)"  4       "CHR$(34):PRINT
70 INPUT"Welche Berechnung <1 - 4> ";W:N=W-(W<3)*2:IFW<3THEN140
80 CLS:PRINT"Das Gleichungssystem ist zuerst in die Normalform zu brin
gen:

            A11*X1 + A12*X2 + ... + A1n*Xn = B1
90 PRINT"              A21*X1 + A22*X2 + ... + A2n*Xn = B2
              .    +   .    + ... +   .    =  .
100 PRINT"               .    +   .    + ... +   .    =  .
            Am1*X1 + Am2*X2 + ... + Amn*Xn = Bn":PRINT:PRINT
110 '
120 ' Eingabe der Koeffizienten-Matrix:
130 '
140 FORZ=1TON:PRINT"Elemente der "USING"#. ";Z;:PRINT"Zeile:
150 FORS=1TON:INPUTL(W,Z,S):NEXTS,Z
160 IFW>2:PRINT"Konstanten der rechten Seite:
";:FORZ=1TON:INPUTR(Z):NEXTZ
170 ONERRORGOTO570
190 ONWGOSUB350,430,350,430
200 IFW<3GOTO520
210 IFW=3GOTO220
220 DN=D:FORK=1TON:GOSUB480:W=W+1:GOSUB390
230 IFW=4,GOSUB350
240 IFW=5,GOSUB430
250 DZ(K)=D:W=W-1:NEXTK:GOTO560
260 '
270 '    Unterdeterminante 3.Ordnung konstruieren:
280 '
290 FORZ=1TO3:FORS=1TO4:IFS=JTHEN310
300 SO=SO+1:L(W+1,Z,SO)=L(W,Z+1,S)
310 NEXTS:SO=O:NEXTZ:RETURN
320 '
330 '    Determinante 3.Ordnung berechnen:
340 '
350 FORZ=1TO3:FORS=4TO5:L(W,Z,S)=L(W,Z,S-3):NEXTS,Z:FORI=1TO6:P(I)=1:N
EXT:FORS=0TO2:FORZ=1TO3:P(S+1)=P(S+1)*L(W,Z,S+Z):NEXTZ,S:FORS=6TO4STEP
-1:FORZ=1TO3:P(10-S)=-P(10-S)*L(W,Z,S-Z):NEXTZ,S:D=0:FORI=1TO6:D=D+P(I
):NEXT:RETURN
360 '
```

```
370 '  Übergabe der Konstanten der rechten Seite in die K.Spalte:
380 '
390 FORZ=1TON:L(W,Z,K)=R(Z):NEXT:RETURN
400 '
410 '  Adjunkte der Determinante 4.Ordnung bilden:
420 '
430 FORJ=1TO4:GOSUB290:W=W+1:GOSUB350:W=W-1:U(J)=D:IFINT(J/2)=J/2,U(J)
=-U(J)
440 NEXTJ:D=0:FORS=1TO4:D=D+L(W,1,S)*U(S):NEXT:RETURN
450 '
460 '  Zwischendeterminante aufstellen, um Originalwerte zu retten:
470 '
480 FORZ=1TON:FORS=1TON:L(W+1,Z,S)=L(W,Z,S):NEXTS,Z:RETURN
490 '
500 '   Ergebnis der Determinante anzeigen:
510 '
520 CLS:PRINT"Determinante =  "D:END
530 '
540 '  Lösungsmenge der Unbekannten anzeigen:
550 '
560 FORI=1TON:X(I)=DZ(I)/DN:PRINT:PRINT"X"USING"# = ";I;:PRINTX(I):NEX
T:END
570 IFERR=20,PRINT:PRINT"       Das System ist nicht mit der CRAMERsche
n Regel lösbar,
        weil die Koeffizienten-Determinante verschwindet!":END
```

Ein paar Tips zum Arbeiten mit großen Datenfeldern durch Verwendung von INDEX-Dateien

von Klaus Hermann

Dies würde etwa folgendermaßen aussehen:

```
--------------------------------------------------------------------
--------------- EINLESEN DER INDEXDATEI IN DATENFELD ---------------
DIM DISKNAME(Feldgröße), SATZZEIGER(Feldgröße)
OPEN"I",1,"DISKNAME/IND:1"
FOR X = 1 TO FELDGRÖßE
   INPUT 1, DISKNAME(X), SATZZEIGER(X)
--------------------------------------------------------------------
NEXT X
CLOSE 1
--------------------------------------------------------------------

------------- DISKNAME SUCHEN --------------------------------------
INPUT"Diskettenname";EINGABE$
FOR X = 1 TO FELDGRÖßE
   IF EINGABE$ = DISKNAME(X) THEN =a=
NEXT X
PRINT"Diskettenname nicht vorhanden"
END

=a=                          REM Die Hauptdatei wird bereits vorher
GET  2,  SATZZEIGER(X)       geöffnet.
--------------------------------------------------------------------
```

Das Abspeichern von neuen Datensätzen erfolgt, indem man den neuen Datensatz entweder in einen gelöschten Satz schreibt oder an das Ende der Hauptdatei hängt. Danach wird die Indexdatei ergänzt und sortiert.

Zum Sortieren läßt sich noch sagen, daß es sich eigentlicht nur bei sehr großen Dateien lohnt. Kleinere Indexfiles sind auch unsortiert recht schnell nach Wörtern durchsucht.

TSCRIPS & die &-Codes

Meist bei Druckersteuerungen, oft aber auch, um Sonderzeichen des Druckers in den Text aufzunehmen ("♥" oder so), müssen die entsprechenden Hex-Codes in den Text eingeschrieben werden. Sie werden zwischen &-Zeichen eingeschlossen.

Bei den Sonderzeichen des Druckers ist es mühsam, jedesmal im dicken Manual die ASCII-Tabelle zu finden und nach dem Zeichen zu fahnden. Das geht sehr viel einfacher, wenn der komplette Zeichensatz auf Kommando im Bildschirm erscheint. Dazu muß er punktweise erzeugt werden. In "normalen" Genies oder TRS-80 geht das z. B. mit der HRG. Auf meinem G3s, vermutlich auch mit dem G3 und evtl. weiteren kompatiblen Computern kann der normale Zeichensatz soft entsprechend geändert werden. Dies ist hier geschehen.

Das G3s bietet auch die Bequemlichkeit, das Bildschirmformat nahezu beliebig zu variieren. Da hier 16 X 16 Codes angezeigt werden sollen, von denen jeder in der Anzeige 5 Stellen verbraucht, ist die Einstellung auf 80 Zeichen pro Zeile günstig. Wir brauchen 16 Zeilen zur Anzeige und eine weitere, um Tastatureingaben aufzunehmen. Der Video-Controller wird also insgesamt auf 80 X 17 Zeichen programmiert.

Wie auch immer erzeugt, die ASCII-Tabelle steht nun auf dem Bildschirm. Sinn der Sache ist natürlich, Hex-Codes in den TSCRIPS-Text einzubauen. Das wird von dem hier vorgestellten Programm unterstützt. Jede Eingabe einer Hex-Ziffer wird angenommen. Mit BREAK kann die Operation abgebrochen werden. Mit dem Linkspfeil kann wie gewohnt korrigiert werden. Nach ENTER wird der eingegebene String in den Text übernommen. Es werden jedoch nur Strings von geradzahliger Länge akzeptiert, da jede Hexzahl zwei Ziffern hat.

Das Programm ist nicht von so elementarer Wichtigkeit wie z. B. die Trennungsroutine des TSCRIPS, deshalb wird es extern als CMD-File nach dem TSCRIPS-Kommando <@O> aufgerufen. Daher muß auch nach getaner Arbeit mit der CLEAR-Taste von der Kommandozeile in den TSCRIPS-Text zurückgekehrt werden.

Im einzelnen funktioniert das Programm folgendermaßen: Zunächst wird über den Systemport FAh auf den Großbildschirm ab 3800h eingestellt. Benutzt wird er allerdings nur ab 3A00h, weil für 80 X 17 Zeichen nicht der ganze Videospeicher erforderlich ist. Danach wird der benutzte Teil des Großbildschirms gelöscht. Das kann nicht über die Routine an 01C9h geschehen, weil die "glaubt", der TSCRIPS-Bildschirm beginne erst bei 3C00h.

Nach dem Löschen wird das neue Bildschirmformat eingestellt. Das G3s bietet dazu eine Routine an (3551h), die als Vorgaben nur in HL den Start einer Tabelle, in A das erste zu programmierende CRTC-Register und in B die Anzahl der zu programmierenden Register braucht. Die Tabelle vidpar enthält alle CRTC-Parameter einschließlich der Blinkgeschwindigkeit und der Position des Cursors.

Im Segment dsploop werden nun alle ASCII-Zeichen angezeigt. Am Anfang steht der Code in Hex, dann ein Doppelpunkt, dahinter schließlich das Zeichen selbst, gefolgt von einem trennenden Blank. Nach der Anzeige kann wieder auf den Kleinbildschirm zurückgeschaltet werden, um die Tastatur ab 3800h freizugeben.

Die gemachten Eingaben werden in der üblichen Weise überprüft: Nur Hexziffern sowie die Steuerzeichen BREAK, Linkspfeil und ENTER sind erlaubt. Bei jeder anderen Eingabe passiert überhaupt nichts. Nach einem gültigen Zeichen wird dann die Cursorposition neu programmiert. Sie kann sich nur in den Grenzen der letzten Zeile bewegen.

Die Übernahme des Hex-Strings geht folgendermaßen: In 7C2Bh hat TSCRIPS einen Zeiger auf das derzeitige Ende des Textes. An dieses Ende kommt nun zunächst das &-Zeichen. Es folgen dann die Hexziffern. Ein weiteres

&-Zeichen schließt den String ab. Die Stelle dahinter ist nun das neue Ende des Textes. Es wird mit einer Null für TSCRIPS markiert, der Zeiger auf diese Stelle wird in 7C2Bh neu gepatcht.

Mit Hilfe der HRG kann das auch auf Computern mit nicht variablem Bildschirmformat gemacht werden. Lesen wir im nächsten Info, wie? Vielleicht reizt es jemanden.

Arnulf Sopp

```
4200          00001          ORG   4200h          ;im Sekorpuffer ist Platz genug
              00002
4200  DBFA    00003 start    IN    A,(0fah)       ;Systembyte 1
4202  F5      00004          PUSH  AF             ;retten
4203  F610    00005          OR    10h            ;Großbildschirm selektieren
4205  F3      00006          DI                   ;keine Interrupts zulassen
4206  D3FA    00007          OUT   (0fah),A       ;Systembyte neu ausgeben
              00008
4208  219F3F  00009          LD    HL,3f9fh       ;Bildschirmende
420B  3620    00010          LD    (HL),' '       ;letzte Stelle löschen
420D  119E3F  00011          LD    DE,3f9eh       ;vorletzte Bildschirmstelle
4210  019F05  00012          LD    BC,059fh       ;Bildschirmlänge - 1
4213  EDB8    00013          LDDR                 ;Bildschirm löschen
              00014
4215  219442  00015          LD    HL,vidpar      ;Tabelle der CRTC-Parameter
4218  0610    00016          LD    B,10h          ;Länge 16 Bytes
421A  AF      00017          XOR   A              ;A <- 00, ab CRTC-Register 0
421B  CD5135  00018          CALL  3551h          ;CRTC programmieren
              00019
421E  EB      00020          EX    DE,HL          ;HL <- Bildschirmanfang - 1
421F  23      00021          INC   HL             ;HL <- Bildschirmanfang
4220  AF      00022          XOR   A              ;A <- 00, Zeichen ab ASCII 00
4221  F5      00023 dsploop  PUSH  AF             ;retten, wird verändert in 4068
4222  CD6840  00024          CALL  4068h          ;ASCII-Code in Hex anzeigen
4225  363A    00025          LD    (HL),':'       ;':' dahinter
4227  23      00026          INC   HL             ;nächste Bildschirmstelle
4228  F1      00027          POP   AF             ;ASCII-Zeichen
4229  77      00028          LD    (HL),A         ;anzeigen
422A  23      00029          INC   HL             ;übernächste Stelle
422B  23      00030          INC   HL
422C  3C      00031          INC   A              ;nächstes ASCII-Zeichen
422D  20F2    00032          JR    NZ,dsploop     ;falls FF noch nicht überschritten
              00033
422F  F1      00034          POP   AF             ;sonst Systembyte 1 restaurieren
4230  D3FA    00035          OUT   (0fah),A
4232  FB      00036          EI                   ;INTs wieder zulassen
              00037
4233  CD4900  00038 keyloop  CALL  0049h          ;auf Zeichen von der Tastatur warten
4236  FE01    00039          CP    01h            ;BREAK?
4238  284C    00040          JR    Z,exit         ;raus, falls ja
423A  FE0D    00041          CP    0dh            ;ENTER zum Beenden?
423C  282B    00042          JR    Z,patch        ;falls ja
423E  FE08    00043          CP    08h            ;Rückwärtspfeil zur Korrektur?
4240  281A    00044          JR    Z,leftarr      ;falls ja
4242  FE30    00045          CP    '0'            ;gültige Ziffer?
4244  38ED    00046          JR    C,keyloop      ;falls nein
4246  FE3A    00047          CP    ':'            ;höher als gültige Dezziffer?
4248  380B    00048          JR    C,ciphok       ;falls nein
424A  CDB545  00049          CALL  45b5h          ;ggf. in Großbuchstaben umwandeln
424D  FE47    00050          CP    'G'            ;höher als gültige Hexziffer?
424F  30E2    00051          JR    NC,keyloop     ;falls ja
4251  FE41    00052          CP    'A'            ;zwischen 9 und A?
4253  38DE    00053          JR    C,keyloop      ;falls ja
4255  77      00054 ciphok   LD    (HL),A         ;anzeigen
4256  23      00055          INC   HL             ;nächste Bildschirmstelle
4257  CD8942  00056          CALL  setpar         ;Cursor weiterstellen
425A  38D7    00057          JR    C,keyloop      ;falls Zeilenende nicht überschritten
              00058
425C  2B      00059 leftarr  DEC   HL             ;eine Stelle zurück
425D  3620    00060          LD    (HL),' '       ;diese löschen
425F  CD8942  00061          CALL  setpar         ;den Cursor zurückstellen
4262  3002    00062          JR    NC,error1      ;falls Zeilenanfang unterschritten
4264  18CD    00063          JR    keyloop        ;und weiter Tastatur abfragen
4266  23      00064 error1   INC   HL             ;Zeiger wieder vorstellen
4267  18CA    00065          JR    keyloop        ;weiter in der Schleife
              00066
4269  7D      00067 patch    LD    A,L            ;LSB der Cursorposition
426A  B7      00068          OR    A              ;blieb die Eingabezeile leer?
426B  2819    00069          JR    Z,exit         ;raus, falls ja
426D  0F      00070          RRCA                 ;Cy <- Bit 0, muß 0 sein (nur gerade Z.)
426E  38C3    00071          JR    C,keyloop      ;sonst noch eine Ziffer oder Linkspfeil
4270  4D      00072          LD    C,L            ;Zeichenzähler anpassen
4271  2A2B7C  00073          LD    HL,(7c2bh)     ;derzeitiges Ende des TSCRIPS-Textes
4274  3626    00074          LD    (HL),'&'       ;leitet Hex-Sequenzen ein
4276  EB      00075          EX    DE,HL          ;DE <- Textspeicher
4277  13      00076          INC   DE             ;nächste Stelle nach '&'
4278  21003F  00077          LD    HL,3f00h       ;Beginn des Strings
427B  EDB0    00078          LDIR                 ;String in den Text übertragen
427D  EB      00079          EX    DE,HL          ;HL <- Ende des Textes
427E  3626    00080          LD    (HL),'&'       ;Hex-String mit '&' abschließen
4280  23      00081          INC   HL             ;nächste Stelle
4281  3600    00082          LD    (HL),00h       ;Textende mit 0 kennzeichnen
4283  222B7C  00083          LD    (7c2bh),HL     ;und in TSCRIPS vermerken
              00084
4286  C3C901  00085 exit     JP    01c9h          ;Bildschirm restaurieren und raus
              00086
4289  3E0F    00087 setpar   LD    A,0fh          ;CRTC-Register 15, Cursor-Lobyte
428B  D3F6    00088          OUT   (0f6h),A
428D  7D      00089          LD    A,L            ;LSB der Cursorposition
428E  FE50    00090          CP    50h            ;Zeilengrenzen über- o. unterschritten?
4290  D0      00091          RET   NC             ;auch nicht erlaubt
4291  D3F7    00092          OUT   (0f7h),A       ;sonst Cursor weiterstellen
4293  C9      00093          RET
              00094
4294  6E      00095 vidpar   DB    6eh,50h,58h,0ah,14h,06h,11h,12h ;Videoparam. für
429C  02      00096          DB    02h,0eh,49h,09h,02h,00h,07h,00h ;17 X 80 Zeichen
              00097
4200          00098          END   start

00000 Fehler

ciphok  4255   dsploop 4221   error1  4266   exit    4286   keyloop 4233   leftarr 425C
patch   4269   setpar  4289   start   4200   vidpar  4294
```

ASCII-Tabelle immer zur Hand

In dem Beitrag "TSCRIPS & die &-Codes" wird eine ASCII-Tabelle vorgestellt, die zur Verwendung in TSCRIPS allein eigentlich viel zu schade ist. Hier ist nun eine Variante für das DOS allgemein.

Sie wird in ein SYS-File eingepflanzt. Wie das geht, wurde hier schon verschiedentlich gezeigt. Welche SYS-Datei dafür noch Platz bietet, muß der Leser mit SUPERZAP oder DDE prüfen. 94 Bytes müssen frei sein.

Diese Variante ist der für TSCRIPS so ähnlich, daß nur noch die sich unterscheidenden Teile im Listing vorgestellt werden. Was mit LIST OFF ausgespart ist, kann man an den springenden Zeilennummern erkennen. Alles, was mit TSCRIPS zusammenhängt, entfällt hier natürlich. Es sind auch nur noch 16 X 80 Zeichen notwendig. Der Cursor ist ausgeschaltet.

Wie sich das auf dem Bildschirm macht, ist aus der Hardcopy zu entnehmen. Sie zeigt den Zeichensatz des Druckers, wie er für TSCRIPS natürlich besonders interessant ist. Da der Drucker bei HRG einfacher Dichte nur jede dritte Nadel abschießt, sieht sie vielleicht etwas dünn aus. Ich hoffe, daß man alles Wichtige erkennen kann.

Arnulf Sopp

```
5179            00001         ORG     5179h             ;am Ende von SYS28/SYS
                00002
5179  FE3E      00003 sys28   CP      3eh               ;Requestcode für SYS,AT?
517B  C2004D    00004         JP      NZ,4d00h          ;an den Anfang von SYS28/SYS, falls nein
                00005
517E  DBFA      00006         IN      A,(0fah)          ;Systembyte 1
5180  F5        00007         PUSH    AF                ;retten
5181  F610      00008         OR      10h               ;Großbildschirm selektieren
5183  F3        00009         DI                        ;keine Interrupts zulassen
5184  D3FA      00010         OUT     (0fah),A          ;Systembyte neu ausgeben
                00011
5186  21FF3E    00012         LD      HL,3effh          ;Bildschirmende
5189  3620      00013         LD      (HL),' '          ;letzte Stelle löschen
518B  11FE3E    00014         LD      DE,3efeh          ;vorletzte Bildschirmstelle
518E  01FF04    00015         LD      BC,04ffh          ;Bildschirmlänge - 1
5191  EDB8      00016         LDDR                      ;Bildschirm löschen
                00017
                00038
51AD  F1        00039         POP     AF                ;sonst Systembyte 1 restaurieren
51AE  D3FA      00040         OUT     (0fah),A
51B0  FB        00041         EI                        ;INTs wieder zulassen
51B1  CD4900    00042         CALL    0049h             ;auf Tastendruck warten
51B4  CDC901    00043         CALL    01c9h             ;Videoformat restaurieren
51B7  AF        00044         XOR     A                 ;Z-Bedingung ('kein Fehler')
51B8  C9        00045         RET
                00046
51B9  6E        00047 vidpar  DB      6eh,50h,58h,0ah,14h,06h,10h,12h ;Videoparam. für
51C1  D2        00048         DB      02h,0eh,29h,09h,02h,00h         ;16 X 80 Zeichen
                00049
5179            00050         END     sys28

00000 Fehler

dsploop 519F    sys28  5179    vidpar  51B9
```

ÜBRIGENS...
„EINEN GUTEN RITTERFILM ERKENNT MAN AM PRÄDIKAT: BESONDERS SCHWERTVOLL!

Gut gesagt!
„Wir schalten zum Sportstudio um? Da hab' ich ja wohl auch noch ein Mördchen mitzureden!" (Ehefrau, auf ihren Krimi pochend)

* * * B I V A R A N A * * *

– kein Zauberwort, sondern meine Abkürzung für "Zweifache Varianzanalyse". – Was ist das?

Kein Ding im Makrokosmos ist ohne Ursache. (Im Mikrokosmos ist das etwas anders; man denke an *Plancks* Quantensprünge, an *Heisenbergs* "Unschärferelation" in der Atomphysik oder an die Mutation in der Biologie, diese Initiatorin jeden Fortschritts, besser gesagt: jeder Weiterentwicklung).

Ja – keine Wirkung hat nur e i n e Ursache! Stets mehrere – meist unbekannt viele – Einflußgrößen erzeugen eine einzige Wirkung. Die meisten Ursachen bilden jedoch nur die "Rahmenbedingungen"; Auslöser ist oft nur eine einzige Ursache. So kann man zwischen Haupt- und Nebenursachen unterscheiden. Die Rangeinstufung – auch "Gewichtung" genannt – läßt sich mathematisch bestimmen.

Berücksicht werden muß hierbei, daß Ereignisse gleicher Art (z.B. Messungen des gleichen Objekts oder Vorganges) untereinander nie völlig gleich ~~sind~~ ausfallen. Auch zwei Eier nicht. Das Sprichwort "Sie gleichen sich wie ein Ei dem anderen" ist falsch; es muß heißen "Sie ähneln sich...", denn es gilt:
EI ≠ EI !

Man führt deshalb vernünftigerweise stets eine Anzahl Wiederholungen durch (z.B. 10 Messungen oder 100 Befragungen usw.) und bildet davon den Mittelwert. Die Einzelwerte leiden unter einer Eigenschaft, die man "Streuung" nennt.

Befassen wir uns hier lediglich mit dem Fall, daß zwei Einflußgrößen vorliegen: Ein Einfluß X und ein Einfluß Y; außerdem natürlich der Einfluß der "Streuung", der stets vorliegt. – Beispiel:

Ein Stück Land wird in 5 Parzellen (X) unterteilt. In jeder Parzelle werden 6 verschiedene Sorten Rüben (Y) angepflanzt: Mohrrüben, Rote Rüben, Zuckerrüben, Rettich, Kohlrabi und Radieschen, von jeder Sorte je 10 Pflänzchen (Biologen bitte ich um Nachsicht, falls ich hier abiologisch vorgehe; lest trotzdem weiter!) * Der *Wachstumserfolg* Z soll in Gramm Eßbarem gemessen werden (also ohne Blattwerk).

Wir haben somit 5 Variable X(1)...X(5) für die Bodenart, 6 Variable Y(1)...Y(6) für die Rübenart und 10 Gewichte für jede der 30 Kombinationen "Parzelle-Nr./Rübentyp", die wir als Zielgröße Z(X,Y,1)...Z(X,Y,10) bezeichnen, also 300 Gewichte Z(X,Y,N).

Dieser sogenannte "Feldversuch" dient dazu, herauszufinden, ob die Pflanzenart oder die Bodenbeschaffenheit einen stärkeren Einfluß auf das Wachstum hat, denn auch Nährboden, Beregnung und Düngung sind nicht überall gleich. (Man könnte den gleichen Versuch auch mit der gleichen Pflanze, aber verschiedener Düngung anstellen.)

Der statistische Algorithmus, der diese Frage für zwei Einflußgrößen beantwortet, heißt "Zweifache Varianzanalyse". Bei dieser Auswertung erübrigt sich die Anordnung im sogenannten "Lateinischen Quadrat".

Die Zweifache Varianzanalyse ist mathematisch gerade noch gut überschaubar, wäre aber von Hand schon sehr rechenaufwendig. Wenn wir es jedoch mit einem Problem mit N>2 Einflußfaktoren zu tun haben, so ist die Berechnung zwar ebenfalls möglich, aber sie wird immer unübersichtlicher und unbequemer, je größer N ist. Aber dafür haben wir ja heute den Computer!

Mein Programm "BIVARANA" (selbstverständlich auch Bestandteil unserer Programmbibliothek) berechnet für die Z.V.A. die sogenannten "F-Testwerte" sowie die dazu gehörigen "Freiheitsgrade" (der Statistiker – von der mathematischen, nicht von der demografischen Fakultät – weiß, was das ist. Für den Gebrauch meines Programmes ist dieses Wissen jedoch nicht erforderlich).

Es fällt noch eine dritte Testgröße F(WW) für die sogenannte "Wechselwirkung" an. Diese sagt aus, ob auch die Reihenfolge, in der sich (in unserem Beispiel) die Pflanzen aufgrund ihres Wachstums anordnen lassen, von der Bodenqualität abhängig oder unabhängig ist. Dieser Aspekt läßt sich auch umkehren: Die Wechselwirkung zeigt ebenfalls, ob bzw. wie stark die Reihenfolge, in der sich die Parzellen aufgrund ihrer Bodenqualität (nach Maßgabe des hervorgebrachten Rübenwuchses je Rübenart) anordnen lassen, auch von der Rübenart abhängt.

Die Werte, die das Programm für F(X), F(Y) und F(WW) ausgibt, werden nun mit den Werten einer sogenannten "F-Tabelle" verglichen, die (im Falle der Z.V.A.) zwei Eingänge hat, nämlich für die beiden Freiheitsgrade f1 und f2, die vom Programm für jeden der Parameter X, Y und WW ebenfalls ausgegeben werden.

Anstelle einer umfangreichen Tabelle habe ich einen Auszug daraus für die häufigsten Wertebereiche von f1 und f2 (die von der Anzahl der Einzelwerte abhängen) mit Hilfe des HRG-Systems *GRAPE* als Diagramm dargestellt. Die analytische Buerteilung erfolgt nun so:

<u>Wenn der berechnete F-Wert g r ö ß e r ist als der, den man im nachstehendem Diagramm für die jeweils berechnete f1,f2-Kombination abliest (wobei f1 der Parameter der Kurvenschar ist), so ist der betreffende Einflußfaktor mit mindestens 95% Wahrscheinlichkeit *"signifikant"*.</u> Ist das berechnete F kleiner als das F aus dem Diagramm, so hat der betreffende Faktor praktisch keinen signifikanten Einfluß auf das Ergebnis gehabt; er ist *irrelevant*. Die Unterschiede in der Zielgröße sind dann "rein zufällig"!

– Das Verfahren läßt sich auf alle Streuprobleme mit zwei wesentlichen Einflüssen anwenden; es ist nahezu verteilungsunabhängig, wenn N, die Anzahl der Einzelwerte je Einflußgrößenkombination, genügend groß ist.

* * * * *

Ich wünsche euch von nun an tiefere Erkenntnisse in allen Alltagsfragen! Kalot Mb

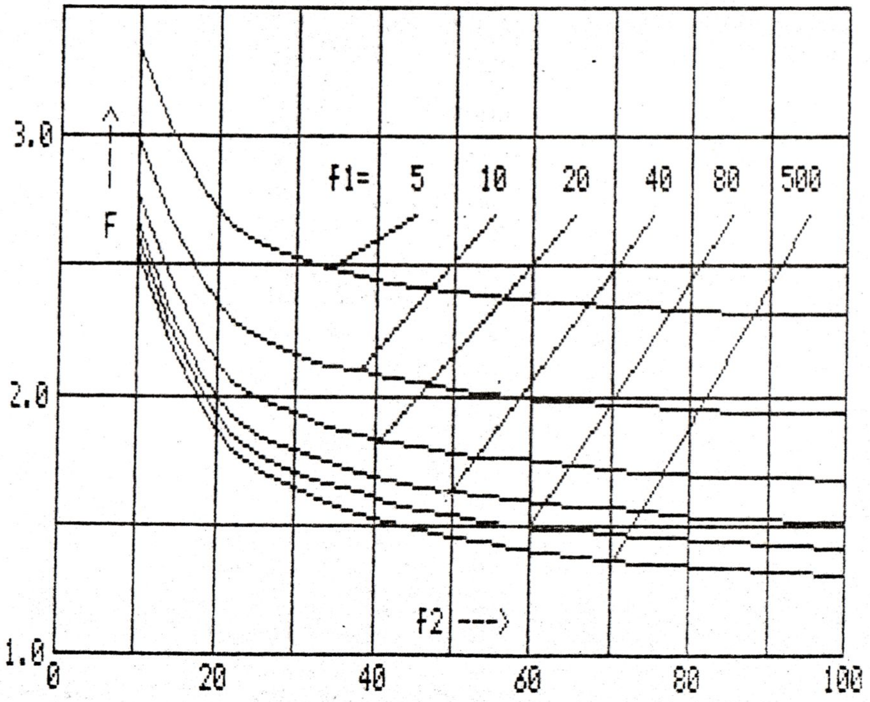
```
5 CLS:CLEAR1000:PRINT" ";STRING$(62,43)
10 PRINT"+    ZWEIFACHE VARIANZANALYSE fuer gleiche Besetzungszahlen
 +
20 PRINT"+                *** B I V A R A N A ***

 +
30 PRINT"+ Verfasser:  K.-J.Muehlenbein,  Weinheim a.d.B.,  Januar 1
983 +
40 PRINT"+ Lit.Graf/Henning/Stange, Formeln u.Tabellen d.math.Statis
tik +
45 PRINT"              Verbesserte Version Maerz 1987
50 PRINT" "STRING$(62,"=")
52 DEFINTF-Q:DEFDBLA-D,R-T,Z:DEFSTRE,U-W:DD$="* * * * * DRUCKER EINS
CHALTEN !! * * * * *"
55 IFPEEK(14312)<>63THENFORI=765TO705STEP-1:PRINT§I,LEFT$(DD$,766-I)
;:FORII=1TO35:NEXTII:PRINTCHR$(958-I):IFPEEK(14312)<>63THENNEXTIELSE
62:FORI=1TO200:NEXTI:PRINT§705,CHR$(255);:GOTO55
57 IFPEEK(14312)<>63:FORI=1TO200:PRINT§705,CHR$(255);:GOTO55
61 Y$="":Y$=INKEY$:IFY$=""THEN61
62 CLS:PRINT"Sollen bereits gespeicherte Werte von Diskette abgerufe
n werden <J,j oder ENTER> ?
63 Y$="":Y$=INKEY$:IFY$=""THEN63ELSEIFY$=CHR$(13)THEN70
65 AD=1:INPUT"Dateiname ";W
67 OPEN"I",1,W+"/Bez":INPUT#1,P,Q,N,E3,E4,E5,E1:FORI=1TOP:INPUT#1,VX
(I):NEXTI
68 INPUT#1,E2:FORJ=1TOQ:INPUT#1,VY(J):NEXTJ:CLOSE:NN=8*P*Q*N:GOSUB97
0:GOTO445
70 CLS:PRINT"Eingabe von Untersuchungsergebnissen :":PRINT:INPUT"Tit
el der Untersuchung ";E4
75 INPUT"Heutiges Datum ";UM
80 INPUT"Zielgroesse ";E3:INPUT"Maszeinheit ";E5
90 PRINT"          1. Einflussgroesse ";:INPUT"(Spalteneinfluss)
 >Maximal 20 Buchstaben ! <  ";E1:IFLEN(E1)>20THEN90ELSEINPUT"System
atische oder Zufallskomponente < S / Z > ";U1
100 INPUT"Anzahl ihrer Variablen ";P:DIMVX(P)
110 PRINT"Welche Bezeichnung hat die
120 FORI=1TOP:PRINTI;:INPUT". Variable (max. 20 Buchstaben) ";VX(I):
NEXTI
130 PRINT:PRINT"          2. Einflussgroesse ";:INPUT"(Zeileneinfluss
)
 >Maximal 20 Buchstaben ! <    ";E2:IFLEN(E2)>20THEN130ELSEINPUT"Sys
tematische oder Zufallskomponente < S / Z ) ";U2
140 INPUT"Anzahl ihrer Variablen ";Q:DIMVY(Q)
150 PRINT"Welche Bezeichnung hat die
160 FORJ=1TOQ:PRINTJ;:INPUT". Variable (max. 20 Buchstaben) ";VY(J):
NEXTJ:PRINT
170 INPUT"Die fuer jede Variablenkombination  g l e i c h  g r o s
s e
Anzahl der Einzelwerte  ";N:DIMZ(P,Q,N),S(P,Q),SI(P),SJ(Q),C(P)
180 CLS:PRINT"       E i n g a b e  der  E i n z e l w e r t e
190 PRINT:PRINT"( Zur Fehlerkorrektur:
 ' - 9999 '  in die  n a e c h s t e  Variable eingeben ! )
200 PRINTSTRING$(63,"=")
210 FORI=1TOP:PRINTVX(I)
220        FORJ=1TOQ:PRINT,,VY(J)
230 FORK=1TON
240 PRINT"Z(";STR$(I);",";STR$(J);",";STR$(K);") =",,K". Wert = ";
250 INPUTZ(I,J,K):IFZ(I,J,K)=-9999GOSUB1100
270 NEXTK,J:PRINT:NEXTI
275 Z=0:PRINT:INPUT"Alle Eingaben O.K.? (Falls nein,  '-9999' eingebe
n ! )    ";Z:IFZ=-9999THENGOSUB1110ELSEGOTO340
280 GOTO275
340 INPUT"Datei-Name (ohne Extension und Drive-Nr.) ";W:NN=8*N*P*Q
```



```
345 INPUT"In welchem Drive sollen die Daten gespeichert werden ";DN$
350 OPEN"O",1,W+"/Bez:"+DN$
360 PRINT#1,P;Q;N;E3;",";E4;",";E5;",";E1
370 FORI=1TOP:PRINT#1,VX(I):NEXTI
380 PRINT#1,E2
390 FORJ=1TOQ:PRINT#1,VY(J):NEXTJ
400 CLOSE
410 OPEN"O",1,W+":"+DN$,"FF",NN
420 FORI=1TOP:FORJ=1TOQ:FORK=1TON
430 PUT1,*,,Z(I,J,K);
440 NEXTK,J,I:CLOSE
445 CLS:E9="":INPUT"Sollen die Messwerte ausgedruckt werden  < J > -
 sonst ENTER !  ";E9:IFE9<"J"THEN570
450 LPRINTCHR$(27)"§",E4:LPRINT,STRING$(LEN(E4),"="):LPRINT" "
475 LPRINTCHR$(27)"l"CHR$(8)
480 LPRINT"Zielgroesse :  ";E3:LPRINT" "
490 LPRINT"Einflussgroessen :";TAB(39)"M e s s w e r t e
500 LPRINTLEFT$(E1,14);TAB(15)"/";" ";LEFT$(E2,14);TAB(47)E5
510 LPRINTSTRING$(64,"-"):LPRINT"
520 FORI=1TOP
530 LPRINTVX(I):FORJ=1TOQ
535 LPRINT,VY(J)
540 FORK=1TONSTEP4:LPRINT,,:FORKK=0TO3:IFK+KK<=NLPRINTUSING"###.##";
Z(I,J,K+KK);:LPRINT"  ";
550 NEXTKK:LPRINT"":NEXTK:LPRINT" ":NEXTJ,I
560 LPRINTSTRING$(63,"-"):LPRINTCHR$(13);CHR$(13)
570 CLS:PRINT"           B i t t e   w a r t e n  !!
(Ich rechne ...............................)":TRON
580 FORI=1TOP:FORJ=1TOQ:FORK=1TON
590 S(I,J)=S(I,J)+Z(I,J,K):S=S+Z(I,J,K):D=D+Z(I,J,K)Ä2:NEXTK
600 SI(I)=SI(I)+S(I,J):C(I)=C(I)+S(I,J)Ä2:NEXTJ
610 A=A+SI(I)Ä2:C=C+C(I):NEXTI
620 FORJ=1TOQ:FORI=1TOP
630 SJ(J)=SJ(J)+S(I,J):NEXTI
640 B=B+SJ(J)Ä2:NEXTJ
650 SP=(A-SÄ2/P)/Q/N:SQ=(B-SÄ2/Q)/P/N
660 SB=(C-A/Q-B/P+SÄ2/P/Q)/N:SN=D-C/N
670 SS=D-SÄ2/P/Q/N
680 FS=P-1:FZ=Q-1:FW=FS*FZ:GW=P*Q*(N-1)
690 S2=SB/FW:S3=SQ/FZ:S4=SP/FS:IFN>1THENS1=SN/GW:ELSES1=S2
700 TW=S2/S1
710 IFU1="S"ORU1="s"THENTZ=S3/S1:GZ=GW:ELSETZ=S3/S2:GZ=FW
720 IFU2="S"ORU2="s"THENTS=S4/S1:GS=GW:ELSETS=S4/S2:GS=FW
730 TROFF:CLS:FORB%=1TO5:LPRINTCHR$(07):NEXTB%:IFN=1GZ=FW:GS=FW
740 PRINT"MITTELWERTE der Zielgroesse ";E3
750 PRINT:PRINTE1,,E2
760 FORI=1TOP:PRINTVX(I);
770 FORJ=1TOQ:PRINT,,LEFT$(VY(J),15),USING"####.##";S(I,J)/N:NEXTJ:N
EXTI
780 PRINTSTRING$(64,"-"):LINEINPUT"Anzeige der F - TESTWERTE  < ENT
ER ! > :";E
790 PRINT:PRINT"Einflussgroesse "E1" :";TAB(48)"F = ";USING"####.##"
;TS:PRINT,"mit den Freiheitsgraden f1 ="FS" und f2 ="GS:PRINT
800 PRINT"Einflussgroesse "E2" :";TAB(48)"F = ";USING"####.##";TZ:PR
INT,"mit den Freiheitsgraden f1 ="FZ" und f2 ="GZ:PRINT
810 IFN>1PRINT"Wechselwirkung";TAB(48)"F = ";USING"####.##";TW:PRINT
,"mit den Freiheitsgraden f1 ="FW" und f2 ="GW:PRINT
820 PRINT"1 Mittelwerte ausdrucken
2 F-Testwerte ausdrucken
3 ENDE":INPUTII:ONIIGOTO830,880,825
825 END
830 LPRINT" ":LPRINTTAB(50)"MITTELWERTE
840 LPRINTE1;TAB(25)E2;TAB(50):E3;" ("E5")
850 FORI=1TOP:LPRINTVX(I)
```

```
860 FORJ=1TOQ:LPRINTTAB(25);LEFT$(VY(J),15);TAB(50)USING"####.##";S(
I,J)/N:NEXTJ:NEXTI
870 LPRINTSTRING$(64,"-"):IFAD=1J$="":INPUT"F-Testwerte ausdrucken <
J,j oder ENTER> ";J$:IFJ$<"J"END
875 PRINT"Ausdruck der F - TESTWERTE - - -
880 LPRINT" ":LPRINT,"Zweifache Varianzanalyse
890 LPRINT" ":LPRINT"Einflussgroesse "E1" :";TAB(41)"F = ";USING"###
#.##";TS
900 LPRINT,"mit den Freiheitsgraden  f1 ="FS" und  f2 ="GS:LPRINT" "
910 LPRINT"Einflussgroesse "E2" :";TAB(41)"F = ";USING"####.##";TZ
920 LPRINT,"mit den Freiheitsgraden  f1 ="FZ" und  f2 ="GZ:LPRINT" "
930 IFN>1LPRINT"Wechselwirkung :";TAB(41)"F = ";USING"####.##";TW
940 IFN>1LPRINT,"mit den Freiheitsgraden  f1 ="FW" und f2 ="GW
950 LPRINTSTRING$(63,"="):LPRINTCHR$(15)"(Datei-Name: ";W;")",,UM:LP
RINTCHR$(18):END
970 OPEN"I",1,W,"FF",NN
980 FORI=1TOP:FORJ=1TOQ:FORK=1TON
990 GET1,*,,Z(I,J,K);
1010 NEXTK,J,I:CLOSE:RETURN
1100 I1=I:J1=J:K1=K
1110 INPUT"Erster Index des falschen Wertes ";I
1120 INPUT"Zweiter    ''   ''      ''       ''    ";J
1130 INPUT"Dritter    ''   ''      ''       ''    ";K
1140 PRINT"Richtiger Wert von  Z (";I;",";J;",";K;") = ";:INPUTZ(I,J
,K)
1150 I=I1:J=J1:K=K1-1:PRINT:RETURN
```

Integralgrenzen der F-Verteilung



BASIC - programmierbare Funktionen
==================================

Mit Hilfe von Funktionen, die man sich als kleine Bibliothek abspeichern und bei Bedarf in Basicprogramme einfuegen kann, lassen sich oft umstaendliche Programmierarbeit und wiederholtes Eintippen der jeweiligen Routinen ersparen. Dadurch dass man innerhalb der Funktionen lokale Variable benutzen kann, kann man sie in jedem Programm benutzen, ohne auf die Variablennamen schauen zu muessen. Hier nun zwei Beispiele von Funktionen:

Mit der ersten Funktion werden an Zahlen fuehrende Nullen vorangestellt, wobei man die Anzahl der Stellen frei waehlen kann. Natuerlich kann man anstatt der Null auch ein anderes Zeichen waehlen. Nuetzlich ist diese Funktion z.B. dann, wenn man einen formatierten Ausdruck moechte und der PRINT USING Befehl nicht angewendet werden kann.
Hier nun die Funktion:

```
DEF FNZAHL$(ZAHL%,ANZST%) = STRING$((ANZST%-LEN(STR$(ZAHL%))+1),"0")
+RIGHT$(STR$(ZAHL%),LEN(STR$(ZAHL%))-1)
```

ZAHL% = eingegebene Zahl aus INPUT-Routine
ANZST% = Anzahl der gewXnschten Stellen incl. fuehrenden Nullen

Bsp.:

```
INPUT"ZAHL --";ZAHL%
PRINT FNZAHL$(ZAHL%,4)
```

****

Die naechste Funktion prueft, ob eine Datumseingabe innerhalb der zulaessigen Grenzen liegt. Nicht geprueft werden der 30. und 31. eines Monats und Schaltjahre. Diese Tests lassen sich aber ebenfalls in entsprechende Funktionen einbauen und mit der untenstehenden verknuepfen.

```
DEF FN CHKDATE(A$) = (MID$(A$,3,1)=".") AND (MID$(A$,6,1)=".")
AND (VAL(MID$(A$,1,2))=>1) AND (VAL(MID$(A$,1,2))<=31)
AND (VAL(MID$(A$,4,2))=>1) AND (VAL(MID$(A$,4,2))<=12)
AND (VAL(MID$(A$,7,2))<=99) AND ( LEN(A$)=8)
```

Bsp.:

```
INPUT"Datum :"; DATUM$
IF FN CHKDATE(DATUM$) THEN PRINT"Datum richtig"
                       ELSE PRINT"Datum falsch"
```

(KH)

§ contra @

Wenn wir unsere Drucker auf den deutschen Zeichensatz umschalten, erscheint als ASCII 40h das Paragraphenzeichen. Die Katasteranwälte und Scharfrichter unter uns mögen sich darüber freuen, i. a. meinen wir mit 40h aber den Klammeraffen. Mit einem Patch in TSCRIPS läßt sich das zumindest für dieses Textprogramm beheben. Die im Listing angegebenen Adressen gelten für die Version 5.4.

Zu erklären gibt es dabei nicht viel. Es wird eben überprüft, ob gerade ein ASCII 40h auf den Drucker gehen soll. Wenn ja, könnte es sich um einen Graphikcode bei einer IMAGE-Schrift handeln. Der soll natürlich nicht verändert werden. Es kann auch sein, daß 40h zu einer Steuersequenz gehört (z. B. ESC @ oder FS @ zum Resetten des Druckers). Das wird durch Belauern des zuvor gedruckten Zeichens abgecheckt. Wenn es kleiner als ASCII 20h war, passiert auch nichts. Ansonsten aber wird auf den amerikanischen Zeichensatz mit seinem richtigen Klammeraffen umgeschaltet, dann 40h geprintet und schließlich wieder auf den deutschen Zeichensatz gegangen. Der Codestring für diese Druckerausgabe steht beim Label §string. Er ist gegf. an andere Drucker anzupassen.

Wieso heißt das Label nicht @string? Weil ZEUS mit diesem TSCRIPS-Patch natürlich nichts anfangen kann. Und warum konnte oben § gedruckt werden, wo doch TSCRIPS nun verändert ist? Weil mein Drucker das Paragraphenzeichen noch einmal bei ASCII 15h hat. Wo ihr das im Zeichensatz eures Druckers findet, falls ihr doch einmal ein § sehen wollt, müßt ihr im Manual nachschlagen.

Das Listing ist eine Fortschreibung des Patches, der den vollen Randausgleich bei Proportionalschrift mit einem NEC P6 erlaubt. Da dieses Programm seit seiner Veröffentlichung in dem Beitrag "Neues zur Proportionalschrift" wesentlich verbessert wurde, gebe ich hier noch einmal das (fast) komplette Listing in seiner neuen Form wieder. "Fast" deshalb, weil da noch mehr drinsteht, was ihr in dem Artikel "Von der Wiege bis zur Bahre ..." und seinem Nachtrag lesen könnt. Die mit LIST OFF ausgesparten Teile sind an den springenden Zeilennummern zu erkennen.

Arnulf Sopp

```
86E3            00027          ORG     86e3h
86E3  CD0030    00028          CALL    check7f         ;auf § und Steuerzeichen 7F prüfen
86E4            00029 calladr  EQU     $-2
                00030
                00036 ;prüfen, ob Steuerzeichen 7F (voller Randausgleich, Proportionalschrift)
3000            00037          ORG     3000h           ;Mimen auch möglich, dann aber 6049 änd.!
3000  FE7F      00038 check7f  CP      7fh             ;Steuerzeichen 7F?
3002  E5        00039          PUSH    HL              ;Textzeiger retten
3003  2007      00040          JR      NZ,check40      ;auf Klammeraffen prüfen, falls nein
3005  CD3E30    00041          CALL    chkgrph         ;feststellen, ob Druckergraphik aktiv ist
3008  2822      00042          JR      Z,toggful       ;Randausgleich schalten, falls Steuer-7F
300A  1819      00043          JR      prntout         ;ausdrucken, falls HRG-7F
300C  CD3E30    00044 check40  CALL    chkgrph         ;gehört 40h vielleicht zu Druckergraphik?
300F  2014      00045          JR      NZ,prntout      ;raus, falls ja
3011  FE40      00046 gochk40  CP      '§'             ;Klammeraffe?
3013  2010      00047          JR      NZ,prntout      ;raus, falls nein
3015  3A8E30    00048          LD      A,(previos)     ;gehört es vielleicht zu ESC § oder FS §?
3018  FE20      00049          CP      ' '             ;dann wäre es < Blank
301A  3E40      00050          LD      A,'§'           ;§ restaurieren
301C  3807      00051          JR      C,prntout       ;falls zu einem Steuercode gehörig
301E  218530    00052          LD      HL,§string      ;Codestring für den Klammeraffen
3021  CD6A44    00053          CALL    446ah           ;auf den Drucker ausgeben
3024  AF        00054          XOR     A               ;A <- 00, auf dem Drucker wirkungslos
```



```
3025  E1        00055 prntout  POP     HL              ;Textzeiger restaurieren
3026  328E30    00056          LD      (previos),A     ;zuletzt gedrucktes Zeichen merken
3029  C3B405    00057          JP      05b4h           ;Zeichen ausdrucken
                00058
                00059 ;7F angetroffen, Drucker auf vollen Randausgleich bis CR angetroffen
302C  214530    00060 toggful  LD      HL,check0d      ;Routine zum Überprüfen auf CR
302F  22E486    00061          LD      (calladr),HL    ;als neue Umleitungsadresse patchen
3032  AF        00062          XOR     A               ;A <- 00
3033  328D30    00063          LD      (count),A       ;Zeichenzähler auf 0 setzen
3036  218230    00064 makeful  LD      HL,fulljst      ;Codestring für vollen Randausgleich
3039  CD6A44    00065          CALL    446ah           ;auf den Drucker ausgeben
303C  18E7      00066          JR      prntout         ;und die fehlende 03 auch ausdrucken
                00067
                00068 ;prüfen, ob das Zeichen 7F zu einem Drucker-HRG-String gehört
303E  6F        00069 chkgrph  LD      L,A             ;Zeichen retten
303F  3A4189    00070          LD      A,(8941h)       ;z. Zt. Druckergraphik aktiv?
3042  B7        00071          OR      A               ;(dann >00)
3043  7D        00072          LD      A,L             ;Zeichen restaurieren
3044  C9        00073          RET                     ;zurück mit Z-/NZ-Flag
                00074
                00075 ;prüfen, ob CR ankommt: Bei Kurzzeilen Randausgleich ausschalten
3045  E5        00076 check0d  PUSH    HL              ;Textzeiger retten
3046  CD3E30    00077          CALL    chkgrph         ;feststellen, ob Druckergraphik aktiv ist
3049  20DA      00078          JR      NZ,prntout      ;Graphikzeichen ausdrucken, falls ja
304B  FE7F      00079          CP      7fh             ;mit 7F Randausgleich ausschalten?
304D  2821      00080          JR      Z,togglft       ;falls ja
304F  218D30    00081          LD      HL,count        ;Zeichenzähler
3052  34        00082          INC     (HL)            ;aufaddieren
3053  FE0D      00083          CP      0dh             ;war es ein CR?
3055  20BA      00084          JR      NZ,gochk40      ;auf Klammeraffen prüfen, falls nein
3057  E5        00085          PUSH    HL              ;Adresse count retten
3058  21647C    00086          LD      HL,7c64h        ;hier rechter Rand gespeichert
305B  7E        00087          LD      A,(HL)          ;laden
305C  23        00088          INC     HL              ;hier (7C65) linker Rand gespeichert
305D  96        00089          SUB     (HL)            ;A <- Zeichen/Zeile (Differenz v. beiden)
305E  D60A      00090          SUB     0ah             ;vorsichtshalber noch 10 weniger
3060  E1        00091          POP     HL              ;Zeiger auf den Zähler count
3061  96        00092          SUB     (HL)            ;bisher aufgelaufene Zeichen
3062  3600      00093          LD      (HL),00h        ;Zähler rücksetzen
3064  3E0D      00094          LD      A,0dh           ;CR restaurieren
3066  38BD      00095          JR      C,prntout       ;falls Langzeile
3068  217E30    00096          LD      HL,leftjst      ;Codestring für linken Randausgleich
306B  CD6A44    00097          CALL    446ah           ;auf den Drucker ausgeben
306E  18C6      00098          JR      makeful         ;vollen Randausgleich restaurieren
                00099
                00100 ;nach zweitem 7F auf linken Randausgleich zurückschalten
3070  210030    00101 togglft  LD      HL,check7f      ;Umleitung, um auf 7F zu prüfen
3073  22E486    00102          LD      (calladr),HL    ;in TSCRIPS patchen
3076  217E30    00103          LD      HL,leftjst      ;Codestring für linken Randausgleich
3079  CD6A44    00104          CALL    446ah           ;auf den Drucker ausgeben
307C  E1        00105          POP     HL              ;Textzeiger restaurieren
307D  C9        00106          RET
                00107
                00108 ;Druckercodes für linken und vollen Randausgleich, Zählbyte
307E  1B        00109 leftjst  DB      1bh,'a',00h,0dh ;Codestring für linken Randausgleich
3082  1B        00110 fulljst  DB      1bh,'a',03h     ;Codestring für rechten Randausgleich
3085  1B        00111 §string  DB      1bh,'R',00h,'§',1bh,'R',02h,03h ;String für Klammeraffen
308D  00        00112 count    DB      00h             ;Zähler der Zeichen pro Zeile
308E  00        00113 previos  DB      00h             ;Puffer für letztes Zeichen
                00265
9100            00266          END     9100h           ;Einsprung in der normalen TSCRIPS-Entry

00000 Fehler
```

```
10 '        *********************************************************
20 '        *                V I S I C A L C - DATEI                *
30 '        *  im DIF-Format nach Spalten oder Zeilen sortieren *
40 '        *     <C> Klaus-Jürgen Mühlenbein, Weinheim 5/87        *
50 '        *********************************************************
60 CLS:CLEAR25000:PRINTSTRING$(59,"*")
70 PRINT"*                          V I S I C A L C -                   *
*       Datei im Data Interchange Format sortieren          *
80 PRINT"*                                                              *
90 PRINT"*          <C> K.-J.Mühlenbein, Weinheim, Mai 1987            *
100 PRINTSTRING$(59,"*");PRINT
110 DEFINTI,J,L,S,T,V,Z:DEFSTRA,D,F,K,L,P,U,W,Y:DIMK(16)
120 INPUT"Welche DIF-Datei soll angezeigt bzw. umgeordnet werden
 (ohne /DIF !) ";F:F=F+"/DIF"
130 INPUT"          <1> nur anzeigen
      <2> nur ausdrucken
      <3> anzeigen und ausdrucken
      <4> umordnen und neu speichern
      <5>  =  <1> + <4>
      <6>  =  <2> + <4>
      <7>  =  <3> + <4>
";J
140 OPEN"I",1,F
150 FORI=1TO16:INPUT#1,K(I):NEXT:Z=VAL(K(7)):S=VAL(K(11))
160 CLS:PRINT"Die Tabelle hat"S"Spalten und"Z"Zeilen":DIMD(S,Z),DD(S,Z)
170 '
180 ' ******* Anzeige der Zeilennummern während der Wartezeit *****
190 '
200 TRON
210 FORIS=1TOS:INPUT#1,Y1,Y2,Y3
220 FORIZ=1TOZ:INPUT#1,T,V,W(1)
230 W(0)=STR$(V):D(IS,IZ)=W(T)
240 NEXTIZ,IS:CLOSE
250 CLS:TROFF
260 ONJGOSUB320,390,320,440,320,390,320
270 ONJGOSUB280,280,390,710,440,440,390:IFJ=7THEN440
280 END
290 '
300 ' ********* Anzeige der VC-Datei auf dem Bildschirm *******
310 '
320 PRINT"                                   Zum Anhalten und Fortsetzen: <SHI
FT §> !
      (bitte etwas warten...)
330 FORI=1TO1500:NEXT:FORIZ=1TOZ:FORIS=1TOS:PRINT"Zeile"IZ", Spalte"IS"
: ";D(IS,IZ):NEXTIS:PRINT:NEXTIZ:PRINT§936,"ENTER!"
340 A="":Y="":Y=INKEY$:IFY=""THEN340ELSEPRINT:PRINT"Noch einmal anzeige
n <jJ> ";:INPUTA:IFA="j"ORA="J",GOTO320
350 RETURN
360 '
370 ' **************** Ausdrucken der VC-Datei ***************
380 '
390 FORIZ=1TOZ:FORIS=1TOS:LPRINT"Zeile"USING"##";IZ;:LPRINT", Spalte"US
ING"##";IS;:LPRINT": "D(IS,IZ):NEXTIS:LPRINT:NEXTIZ:RETURN
```



```
420 ' ***************** VC-Datei umordnen ***************
430 '
440 CLS:INPUT"
                 eine Spalte umordnen  ==> 1
                 eine Zeile  umordnen  ==> 2
";J1:IFJ1<1ORJ1>2THEN440
450 U=MID$("SpalteZeile",7+(J1=1)*6,6)
460 PRINT:PRINT"Welche "U" ?",:LINEINPUTNR$:IFVAL(NR$)THENNR=VAL(NR$)EL
SEN1$=LEFT$(NR$,1):N2$=MID$(NR$,2,1):NR=ASC(N1$)-64:IFN2$<>""THENNR=(AS
C(N1$)-64)*26+ASC(N2$)-64
470 INPUT"Wörter oder Zahlen ordnen <W/Z> ";A:IFA="Z"ORA="z",T2=6:DEFDB
LP:ZAHL=1
480 INPUT"<s>teigend oder <f>allend ordnen ";A:IFINSTR("Ss", A)THENT1=1
ELSET1=3
490 V=J1*T1+T2:ONERRORGOTO1110:IFV>6,V=V-6
500 ONVGOSUB510,520,530,10,10,540
510 N=Z:GOSUB550:GOSUB610:GOTO630
520 N=S:GOSUB550:GOSUB610:GOTO640
530 N=Z:GOSUB550:GOSUB620:GOTO630
540 N=S:GOSUB550:GOSUB620:GOTO640
550 DIMP(N):FORI=1TON:IFZA<>1:IFJ1=1,P(I)=D(NR,I):ELSEP(I)=D(I,NR)
560 IFZA=1:IFJ1=1,P(I)=VAL(D(NR,I)):ELSEP(I)=VAL(D(I,NR))
570 NEXT:RETURN
580 '
590 ' ***** Daten nach ASCII ordnen und den Variablen zuteilen ***
600 '
610 CMD"O",N,*IX(0),P(1):RETURN
620 CMD"O",N,*IX(0),-P(1):RETURN
630 FORI=1TOS:FORJ=1TOZ:DD(I,J)=D(I,IX(J-1)):NEXTJ,I:GOTO650
640 FORI=1TOS:FORJ=1TOZ:DD(I,J)=D(IX(I-1),J):NEXTJ,I
650 FORI=1TOS:FORJ=1TOZ:D(I,J)=DD(I,J):NEXTJ,I
680 '
690 ' ************** Abspeichern der geänderten Datei ***********
700 '
710 INPUT"Neuer Filename (ohne Extension und Laufwerk)";FL:INPUT"In wel
chem Laufwerk speichern )";LW:FL=FL+"/DIF:"+LW
720 WW(0)="0":WW(1)="1":WW(3)="-1"
730 OPEN"O",1,FL
740 '
750 ' **************** KENNSATZ *****************
760 '
770 PRINT#1,"TABLE"
780 PRINT#1,WW(0);",";WW(1)
790 PRINT#1,CHR$(34);CHR$(34)
800 PRINT#1,"VECTORS"
810 PRINT#1,WW(0);",";STR$(Z)
820 PRINT#1,CHR$(34);CHR$(34)
830 PRINT#1,"TUPLES"
840 PRINT#1,WW(0);",";STR$(S)
850 PRINT#1,CHR$(34);CHR$(34)
860 PRINT#1,"DATA"
870 PRINT#1,WW(0);",";WW(0)
880 PRINT#1,CHR$(34);CHR$(34)
```

```
900 ' ***** Aufbau des VISICALC-Files ***
910 '
920 FORJ=1TOS
930 PRINT#1,WW(3);",";WW(0)
940 PRINT#1,"BOT"
950 FORI=1TOZ
960 IFVAL(D(J,I))THENGOSUB1040ELSEGOSUB1090
970 NEXTI,J
980 PRINT#1,WW(3);",";WW(0)
990 PRINT#1,"EOD"
1000 CLOSE:END
1010 '
1020 ' *****  Speichern numerischer Eingaben (VALUEs) *****
1030 '
1040 PRINT#1,WW(0);",";D(J,I)
1050 PRINT#1,"V":RETURN
1060 '
1070 ' *****   Speichern von Worteingaben (STRINGs)   *****
1080 '
1090 PRINT#1,WW(1);",";WW(0)
1100 PRINT#1,CHR$(34);D(J,I);CHR$(34):RETURN
1103 '
1106 ' ************ FEHLER-ROUTINE ****************
1109 '
1110 IFERR=8,CLS:PRINT:PRINT:PRINT"        Diese Reihe besteht aus Wört
ern  u n d  Zahlen
  und kann deshalb  nicht sinnvoll geordnet werden!"
1120 END
```

Programm bei mir erhältlich!

(Bitte Disk einsenden – wird ggf. neu formatiert – plus Porto)
< PDRIVEs angeben >
– nur 40 Tracks –

Klaus-Jürgen Mühlenbein
Am Mönchgarten 22, 6940 Weinheim
Ruf 08201 - 5 50 52

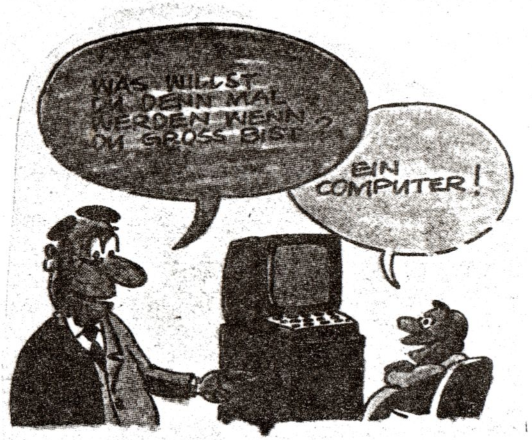
Schwyzertütsch: Dem chalberet na de Mälchstuel.
Dem kalbt sogar noch der Melkschemel.
Der hat mehr Glück als Verstand.

Tschechisch: Aby se vlk nažral a koza zůstala celá.
Auf daß der Wolf satt wird und die Ziege heil bleibt.
Einen Kompromiß aushandeln.

Japanisch: Nō aru taka wa tsume wo kakusu.
Ein erfolgreicher Falke verbirgt seine Klauen.
Stille Wasser gründen tief.

Indonesisch: Bunga uang.
Blume des Geldes.
Zinsen.

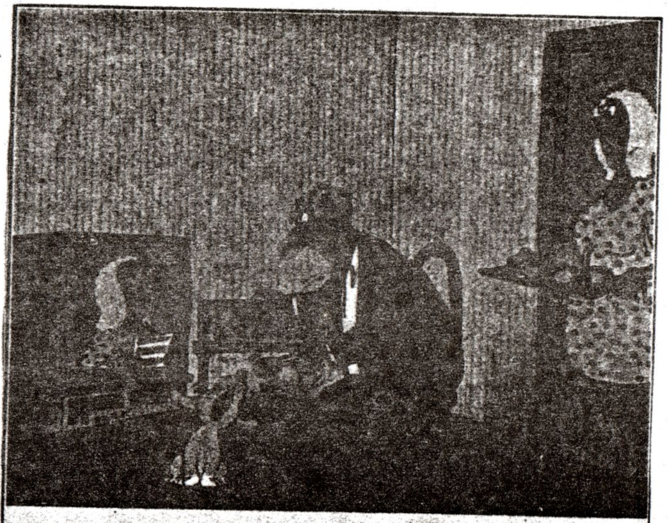
Rudolf Haselberger, Gröbenzell



Ergänzung zum Grafik-Shorty (Info 15, S.40)

Im Listing tauchen in der Zeile zwei "ö" auf, die natürlich eine andere Bedeutung haben. Um das kleine Programm lauffähig zu machen, müssen die zwei "ö" durch den Backslash (Integer Division) "\" ersetzt werden. Das Backslashzeichen erhält man durch die Tastenkombination CLEAR / (= CLEAR SHIFT 7 bei der deutschen Tastatur).

Klaus Hermann


Karl Sch. bei der Nachvertonung eines Videofilms, den er von seiner musikliebenden Gattin gedreht hat

Von der Wiege bis zur Bahre ...
... *bequemere Formulare!*

Es hat mich schon immer geärgert, daß ich manchmal vier- oder sechsmal die Woche die Anschrift eines häufig angeschriebenen Computerpartners tippen muß, wegen der übrigen Partner nicht selten zwanzigmal die Woche das Datum. Wenn ich dann nicht auswendig weiß, wo einer wohnt, darf ich auch noch in den Infos nach seiner Adresse suchen - ach so, dieses Info war mal wieder vorsichtshalber ohne Namensliste - weitersuchen! Das macht in der Summe zwar nur vielleicht eine Viertelstunde in der Woche aus, aber es ist überaus lästig. Die Zeiten sind vorbei: Das Programm, das hier vorgestellt wird, reduziert die notwendigen Tastenbedienungen auf zwei Zeichen plus seinen eigenen Namen, mit dem man es aufruft.

Es sucht dann selbsttätig nach dem Platz oder den Plätzen im Briefkopf, wo das Datum erscheinen soll und setzt es dort ein. Wenn nirgends eines vorgesehen ist, geht es gleich zum nächsten Schritt über: Es schaut im Formular nach, wo die Adresse hingehört. Auch hier passiert nichts, wenn es keine gibt. Findet es aber ein Adreßfeld, liest es dort zwei Zeichen aus, z. B. die Anfangsbuchstaben eines Namens. In einem Extrafile ADRESSEN/LIB sucht es nach der so gekennzeichneten Anschrift und setzt sie in den Briefkopf ein. Wenn keine Anschrift mit diesem Kürzel gefunden wird, kommt in der unteren Statuszeile eine entsprechende Mitteilung. Nur dann muß man sich bequemen, den Empfänger doch noch zu Fuß einzutippen.

Das Programm wird von TSCRIPS aus mit der Klammeraffe-O-Option aufgerufen. Nach diesem Tastendruck erscheint bei der Version 5.4 die Meldung 'DOS-COMMAND?'. Man tippt den Namen des Programms ein, das nun loslegt und alles erledigt. Zunächst sieht es aus, als sei überhaupt nichts passiert, denn nur im Text, nicht im Bildschirm werden die Updates durchgeführt. Kehrt man nun mit CLEAR aus der DOS-Command-Ebene zurück, steht plötzlich in einem bisher leeren Adreßfeld eine komplette Anschrift, das Datum erscheint da, wo dafür bisher nur ein paar Leerstellen reserviert waren.

Dieser im Briefformular reservierte Platz muß eine genau definierte Länge haben. Es wäre durchaus variabel gegangen, hätte dann aber den Programmieraufwand und den damit verbundenen Speicherplatzbedarf, auch die Rechenzeit empfindlich erhöht. Der Platz für das Datum muß genau 10 Zeichen betragen. Die Blockmarkierung umfaßt am Anfang 3 Zeichen (Blockmarke, Blockname, Rechtspfeil), am Ende eines (Blockende-Marke). Es müssen also innerhalb des Blocks weitere 6 Zeichen erscheinen (irgendwelche Punkte oder so), um auf die Länge 10 zu kommen.

Dasselbe gilt für das Adreßfeld. Es muß genau 254 Zeichen groß sein (4 Zeichen zur Blockmarkierung, die beiden Zeichen des Namenskürzels und 248 weitere Platzhalter, z. B. wieder Punkte). Ein fast kompletter Sektor für eine Adresse erscheint reichlich verschwenderisch. Diese Größe erleichtert beim Suchen in ADRESSEN/LIB jedoch ungemein die Arbeit. Schließlich dürfte man es kaum je schaffen, mit einem gewöhnlichen Brief den Speicher voll auszureizen, so daß es einfach egal ist. Die beiden zu einer Page fehlenden Zeichen erklären sich daraus, daß das Namenskürzel aus ADRESSEN/LIB natürlich nicht mit in den Text übertragen wird.

Die Datei ADRESSEN/LIB wird mit TSCRIPS erzeugt. Man schreibt ein Adreßfeld, wie man es damals bei der Erstellung seines Briefkopf-Formulars machte. Es sollte etwa so aussehen:

```
xx$W&01&An                    ■
Name                          ■
Straße                        ■
                              ■
PLZ Stadt                     ■
                        $W&00&■
```


Wie man sieht, können ohne weiteres bereits Druckersteuercodes darin enthalten sein. Hier wird mit ESC-W-01 auf Breitschrift geschaltet, die am Ende wieder rückgängig gemacht wird. Wie man weiterhin sieht, sind 256 Zeichen gar nicht einmal so wahnsinnig viel; mit noch mehr Druckercodes und einem ziemlich langen Straßennamen kann es bereits eng werden.

Um in ADRESSEN/LIB nun nicht nur diese eine, sondern eine ganze Latte solcher Blanko-Anschriften zu haben, wird die erste einfach kopiert. Mit der üblichen Vorgehensweise wird die Adresse als Block markiert, der sich nun beliebig oft kopieren läßt. Danach können alle Blankos mit echten Anschriften ausgefüllt werden. Eine sollte leer bleiben, denn wenn man eine neue braucht, sollte noch etwas zum Kopieren da sein, am besten ganz am Ende des Files.

Die ausgefüllten Anschriften müssen nun ihre Duftmarke aus zwei Zeichen erhalten. Die erste Zeile des obigen Musters sieht dann z. B. für Jens Neueders Adresse folgendermaßen aus:

```
JN$W&01&Herrn                 ■
```

Aus dem Dummy 'xx' wurde JN, aus dem Wegweiser 'An' die Anrede 'Herrn'. Das Programm findet nun in ADRESSEN/LIB die Anschrift von Jens-80 und kopiert sie ab 'Herrn' oder, genauer gesagt, vom Druckercode an in den Text.

Bei der Beschreibung der Programmlogik möchte ich mich kurz fassen. Das Datum steht binär in den Speicherstellen 4044-4046h. Von dort wird es ausgelesen und in drei Schritten mit Hilfe der ROM-Routine an 0FEFh dezimal umgewandelt. Tag, Monat und Jahr werden sodann noch ein wenig nachgearbeitet, damit führende Nullen verschwinden und Leerzeichen hinter den Punkten erscheinen.

Nun wird ein Block namens D im Text gesucht. Wo der Text beginnt, kann aus 5277h in TSCRIPS ausgelesen werden. Das Textende, woraus sich die Länge der Suchstrecke errechnen läßt, steht in 7C2Bh. In dem Unterprogramm seekblk wird der Block D gesucht. Dort kommt das Datum hinein. Anschließend wird nach weiteren Blöcken mit dem Namen D gesucht. Es könnte ja sein, daß es an mehreren Stellen erscheinen soll.

Mit der Anschrift geht es ebenso. Der Unterschied ist nur, daß nur nach *einem* Block A gesucht wird. Daß ein Brief mehrere Adreßfelder hat, kann ausgeschlossen werden. Wenn ein Block A existiert, wird ADRESSEN/LIB eingelesen. Das geschieht mit den üblichen DOS-Routinen, die ich nicht weiter beschreiben will. Der Einfachheit und Faulheit halber wird sektorweise eingelesen, bis ein Fehler auftritt. In 99,9% der Fälle kann es sich nur um den Fehler "Ende der Datei angetroffen" handeln. Deshalb wird nicht extra geprüft, ob es vielleicht "Pizza im Laufwerk" war.

Nach Auftreten dieses "Fehlers" wird in der eingelesenen Adreß-Bibliothek nach der Anschrift gesucht, deren Kürzel im Formular erschien. Bei erfolgreicher Suche wird die Adresse an die Stelle des Blocks A kopiert, wie es bereits mit dem Datum im Block D geschehen war. Wenn der betreffende Empfänger noch nicht in der Bibliothek steht, kommt in der Statuszeile der Hinweis. Das war es dann auch schon.

Wer dieses Programm nutzen möchte, hat zunächst die Mühe, in seinem alten Formular die reservierten Plätze für Datum und Adresse auf genau 10 bzw. 254 Stellen zu ändern. Das mag eine Viertelstunde dauern. Von da ab spart er aber einen Haufen Zeit.

Arnulf Sopp

```
                 ;                       A D R E S S E N
                 ;
                 ;        ein Hilfsprogramm zur Verwendung mit TSCRIPS 5.4
                 ;
                 ;                    (C) 1987 Arnulf Sopp
                 ;
                 ; In einen markierten Block mit dem Namen D wird das Datum eingetragen,
                 ; in einen Block mit dem Namen A eine fertige Anschrift von Disk geladen.
                 ; Das Programm wird nach Klammeraffe-O als DOS-Befehl aufgerufen.


3067                     ORG     3067h           ;ab hier Platz (Himem geht aber auch)

                 ;das aktuelle Datum auslesen und einen ASCII-String daraus bilden
3067  E5         start   PUSH    HL              ;alles retten
3068  D5                 PUSH    DE
3069  C5                 PUSH    BC
306A  214640             LD      HL,4046h        ;Tag im Datum
306D  116531             LD      DE,strbuff      ;Puffer für fertigen Datumsstring
3070  010003             LD      BC,0300h        ;drei Durchläufe für Tag, Monat, Jahr
3073  C5         datloop PUSH    BC              ;retten
3074  7E                 LD      A,(HL)          ;Datum laden
3075  2B                 DEC     HL              ;auf nächstes Datum stellen
3076  E5                 PUSH    HL
3077  D5                 PUSH    DE
3078  61                 LD      H,C             ;HL <- 00xx
3079  6F                 LD      L,A             ;HL <- Datum
307A  222141             LD      (4121h),HL      ;Integer-X-Register des Interpreters
307D  215F31             LD      HL,numbuff      ;fünfstelliger Ziffernpuffer
3080  CDEF0F             CALL    0fefh           ;Binärzahl in Dez-ASCII in den Puffer
3083  216131             LD      HL,numbuff+2    ;3 führende Nullen übergehen
3086  D7                 RST     10h             ;A <- Zehnerstelle
3087  D1                 POP     DE              ;Stringpuffer
3088  FE30               CP      '0'             ;führende 0?
308A  2802               JR      Z,nextcph       ;falls ja
308C  12                 LD      (DE),A          ;Ziffer in den Puffer
308D  13                 INC     DE              ;nächste Pufferstelle
308E  D7         nextcph RST     10h             ;nächste Ziffer einlesen
308F  12                 LD      (DE),A          ;Ziffer puffern
3090  13                 INC     DE              ;nächste Pufferstelle
3091  EB                 EX      DE,HL           ;HL <- Pufferadresse
3092  362E               LD      (HL),'.'        ;Punkt hinter das Datum
3094  23                 INC     HL              ;nächste Stelle
3095  3E20               LD      A,' '           ;Blank
3097  77                 LD      (HL),A          ;Blank hinter den Punkt setzen
3098  23                 INC     HL              ;nächste Pufferstelle
3099  EB                 EX      DE,HL           ;Register zurücktauschen
309A  E1                 POP     HL              ;Datumszeiger
309B  C1                 POP     BC              ;Schleifenzähler
309C  10D5               DJNZ    datloop         ;nächste Datumsstelle bearbeiten
309E  1B                 DEC     DE              ;auf Punkt hinter der Jahreszahl stellen
309F  1B                 DEC     DE
30A0  12                 LD      (DE),A          ;Punkt mit Blank überschreiben

                 ;Blöcke namens D suchen und dort das Datum ablegen
30A1  2A7752             LD      HL,(5277h)      ;dort Anfang des Textpuffers abgelegt
30A4  ED5B2B7C           LD      DE,(7c2bh)      ;hier Ende des Textpuffers
30A8  E5                 PUSH    HL
30A9  B7                 OR      A               ;Cy löschen
30AA  ED52               SBC     HL,DE           ;Textlänge ermitteln
30AC  44                 LD      B,H             ;als Zähler nach BC
30AD  4D                 LD      C,L
30AE  E1                 POP     HL              ;Textanfang
30AF  E5                 PUSH    HL              ;wird noch gebraucht
30B0  C5                 PUSH    BC
30B1  1EC4               LD      E,'D'+128       ;Blockname D mit gesetztem Bit 7
30B3  CD3031     seekD   CALL    seekblk         ;einen Datumsblock suchen
30B6  2012               JR      NZ,seekA        ;Adreßblock suchen, wenn kein Datumsblock
30B8  E5                 PUSH    HL              ;sonst Suchadresse retten
30B9  D5                 PUSH    DE              ;dto. Blocknamen in E
30BA  C5                 PUSH    BC              ;dto. Bytezähler
30BB  2B                 DEC     HL              ;zurück auf die Blockmarke
30BC  EB                 EX      DE,HL           ;DE <- Zieladresse des Datums
30BD  216531             LD      HL,strbuff      ;Datumsstring
30C0  010A00             LD      BC,000ah        ;10 Stellen des Strings
30C3  EDB0               LDIR                    ;Datum in den Text übertragen
30C5  C1                 POP     BC
30C6  D1                 POP     DE
30C7  E1                 POP     HL
30C8  18E9               JR      seekD           ;die Suche fortsetzen

                 ;nach einem Anschriftenblock namens A suchen
30CA  C1         seekA   POP     BC              ;Textlänge als Bytezähler
30CB  E1                 POP     HL              ;Textanfang
30CC  1EC1               LD      E,'A'+128       ;Blockname A mit gesetztem Bit 7
30CE  CD3031             CALL    seekblk         ;einen Adreßblock suchen
30D1  2058               JR      NZ,exit         ;Ende, falls keiner vorhanden
30D3  E5                 PUSH    HL              ;Blockadresse retten

                 ;der Block existiert; das Anschriftenfile ADRESSEN/LIB laden
30D4  215231             LD      HL,filname      ;Name der Datei ADRESSEN/LIB
30D7  11B17C             LD      DE,7cb1h        ;FCB von TSCRIPS
30DA  CD1C44             CALL    441ch           ;Dateinamen in den FCB übertragen
30DD  2A2B7C             LD      HL,(7c2bh)      ;Dateipuffer gleich hinter dem Text
30E0  23                 INC     HL              ;aber die Ende-Null freilassen
30E1  0600               LD      B,00h           ;logische Recordlänge =256
30E3  CD2444             CALL    4424h           ;Datei eröffnen
30E6  CD3644     readsec CALL    4436h           ;einen Sektor lesen
30E9  2006               JR      NZ,readone      ;falls Ende der Datei erreicht
30EB  24                 INC     H               ;Puffer 256 Bytes weiterstellen
30EC  22B47C             LD      (7cb4h),HL      ;neuen Puffer im FCB vermerken
30EF  18F5               JR      readsec         ;weitere Sektoren laden

                 ;ein Namenskürzel mit zwei Zeichen als Suchwort laden
30F1  7C         readone LD      A,H             ;MSB der Pufferadresse
30F2  320531             LD      (addrend),A     ;zur Ende-Überprüfung sichern
30F5  E1                 POP     HL              ;Blockadresse
30F6  23                 INC     HL              ;auf Namenskürzel stellen
30F7  23                 INC     HL
30F8  5E                 LD      E,(HL)          ;DE <- Namenskürzel
30F9  23                 INC     HL
30FA  56                 LD      D,(HL)
30FB  2B                 DEC     HL              ;auf Blockmarke zurückstellen
30FC  2B                 DEC     HL
30FD  2B                 DEC     HL
30FE  2B                 DEC     HL
30FF  E5                 PUSH    HL              ;Blockadresse retten

                 ;im Adressenfile nach demselben Kürzel suchen
3100  2A2B7C             LD      HL,(7c2bh)      ;Puffer der Anschriften
3103  23                 INC     HL              ;hinter dem Text, s. o.
3104  3E00               LD      A,00h           ;MSB der höchsten Pufferadresse
3105             addrend EQU     $-1             ;wird weiter oben gepatcht
3106  4E         seekadr LD      C,(HL)          ;BC <- Namenskürzel in ADRESSEN/LIB
3107  23                 INC     HL
3108  46                 LD      B,(HL)
3109  2B                 DEC     HL              ;auf Anfang einer Anschrift zurückstellen
310A  EB                 EX      DE,HL           ;HL <- Namenskürzel im Briefkopf
```

```
310B  B7                OR      A               ;Cy rücksetzen
310C  E5                PUSH    HL              ;Suchkürzel retten
310D  ED42              SBC     HL,BC           ;bei Gleichheit muß 0000 herauskommen
310F  E1                POP     HL
3110  EB                EX      DE,HL           ;Register zurücktauschen
3111  2810              JR      Z,found         ;falls die richtige Anschrift gefunden
3113  24                INC     H               ;auf die nächste Anschrift stellen
3114  BC                CP      H               ;schon Ende der ADRESSEN erreicht?
3115  20EF              JR      NZ,seekadr      ;weitersuchen, falls noch nicht
                  ;der Bereich ist überschritten, diese Anschrift gibt es nicht
3117  E1                POP     HL              ;Stack bereinigen
3118  213A31            LD      HL,errtext      ;Text 'Anschrift nicht gefunden'
311B  11C03F            LD      DE,3fc0h        ;Anfang letzte Bildschirmzeile
311E  011800            LD      BC,filname-errtext      ;Länge des Textes
3121  1806              JR      wupp            ;und raus
                  ;Anschrift gefunden; in den Text patchen (dient auch zur Fehleranzeige)
3123  D1          found POP     DE              ;Blockadresse im Text
3124  23                INC     HL              ;Namenskürzel überspringen
3125  23                INC     HL
3126  01FE00            LD      BC,00feh        ;Länge des Blocks
3129  EDB0        wupp  LDIR                    ;Adresse in den Text oder Fehler anzeigen
                  ;alles erledigt; Register restaurieren und zurück zu TSCRIPS
312B  C1          exit  POP     BC              ;Register restaurieren
312C  D1                POP     DE
312D  E1                POP     HL
312E  AF                XOR     A               ;A <- 00, 'keine Taste'
312F  C9                RET                     ;in TSCRIPS weitermachen
                  ;Unterprogramm zum Aufsuchen der Blocks (Name in E)
3130  3E97        seekblk LD    A,97h           ;Blockmarke des TSCRIPS
3132  EDB1              CPIR                    ;Blockmarke suchen
3134  C0                RET     NZ              ;Ende, wenn Zähler abgelaufen
3135  7E                LD      A,(HL)          ;A <- Zeichen hinter der Blockmarke
3136  BB                CP      E               ;Blockname gefunden?
3137  20F7              JR      NZ,seekblk      ;anderer Block, weitersuchen
3139  C9                RET
                  ;verschiedene Texte und Puffer
313A  41          errtext DM    'Anschrift nicht gefunden'
3152  41          filname DM    'ADRESSEN/LIB',0dh      ;Filename für den FCB
315F  31          numbuff DM    '12345'         ;Platz für 5 Ziffern
3164  00                DB      00h             ;wird von der Routine an 0FEF verändert
3165  33          strbuff DM    '31. 12. 99  '  ;Platz für endgültigen Datumsstring
3067                    END     start           ;dort Einsprung

00000 Fehler

addrend 3105   datloop 3073   errtext 313A   exit    312B   filname 3152   found   3123
nextcph 308E   numbuff 315F   readone 30F1   readsec 30E6   seekA   30CA   seekD   30B3
seekadr 3106   seekblk 3130   start   3067   strbuff 3165   wupp    3129
```


Nachtrag zu "Von der Wiege ..."

Von dem Programm ADRESSEN/CMD, das automatisch das Datum und eine Anschrift in TSCRIPS-Texte einliest, ist eine verbesserte Version fertiggeworden. Sie hat folgende Vorzüge gegenüber der älteren Version:

1. Das Programm braucht nicht mehr unter DOS-COMMAND (nach @-O) mit seinem Namen aufgerufen zu werden. Jetzt ist es Bestandteil von TSCRIPS und wird mit der Tastenkombination @-K aktiviert. Die alte Funktion von @-K, das Einstellen der Cursorgeschwindigkeit, entfällt nun.

2. Für die Auswahladressen aus dem Textfile ADRESSEN/LIB wird an Speicherplatz nur noch der TSCRIPS-Sektorpuffer benötigt. Das bedeutet, daß der Textspeicher nicht mehr beschnitten wird.

3. Die Blockmarken des Adreßfelds werden nicht mehr überschrieben. So kann durch einfaches Einsetzen eines neuen Namenskürzels der gleichlautende Brief an weitere Empfänger gerichtet werden. Daher muß das Adreßfeld nun *zwischen* den Markierungen 254 Zeichen enthalten, nicht mehr *einschließlich* der Blockmarken.

4. Jetzt wird ein anderer FCB in TSCRIPS benutzt. Es ist derjenige, aus dem mit dem Kommando ?N der aktuelle Filename ausgelesen und angezeigt wird. Nach OPEN enthält der FCB keinen ASCII-Namen mehr, sondern die binären Informationen über das File ADRESSEN/LIB. Auf diese Weise wird verhindert, daß man seinen mühsam erstellten Briefkopf mit einem fertigen Brief überschreibt, falls man den gespeicherten alten Namen benutzen möchte. Dann würde die DOS-Fehlermeldung "kein Dateiname" ausgegeben.

Gleich ist geblieben, daß das Datum und die Anschrift erst nach CLEAR, einem Scroll o. dergl. sichtbar werden, denn die Änderungen werden nur im Textpuffer durchgeführt.

Das Programm zeigt noch eine Merkwürdigkeit, mit der man aber gut leben kann: Wenn eine Anschrift in ADRESSEN/LIB nicht gefunden wurde und anschließend CLEAR eingegeben wird, kann der Cursor auch über die ENTER-Marken hinausbewegt werden. Das gibt sich, wenn man mit Shift-Auf- oder Abwärtspfeil an das obere oder untere Textende geht. Diese Prozedur bereinigt offenbar, was durch die Routine in Unordnung gebracht wurde. Da ich in meinem Programm aber keinen Fehler entdecken konnte, gehe ich davon aus, daß der Teil von TSCRIPS, in den ich eingreife, ein Bug enthält. Soll er halt. Es haben sich keine ernstzunehmenden Nachteile daraus gezeigt.

Wer das Programm haben möchte, kann mir eine formatierte Diskette schicken (mindestens 40/SS/DD, höchstens 80/DS/DD; bitte Zettel mit PDRIVE-Angaben nicht vergessen!). Das Porto kostet DM 1,90, eine passende Versandtasche wird auch nicht verschenkt. Interessenten legen also bitte DM 2,50 in gängigen Briefmarken bei. Das File kommt dann als ZEUS-Quellcode. Wegen der unterschiedlichen Hardware-Voraussetzungen kann es nämlich erforderlich werden, das ORG ins Himem zu verlegen, wenn ab 3000h kein freies RAM vorhanden ist.

Da TSCRIPS zu kurz ist, um die Erweiterung noch aufnehmen zu können, muß es zunächst mit dem DOS-Befehl APPEND,dummyfile,TSCRIPS/CMD verlängert werden. Anschließend wird mit der CDD-Funktion von SUPERZAP der neu hinzukommende Objektcode angehängt.

Arnulf Sopp

Noch mehr Fernsehen für's Geld

Der CRTC (cathode ray tube controller, Bildschirmcontroller) 6854, der in einigen Genies eingebaut ist, schert sich nicht um das Tandy- oder sonst ein Bildschirmformat. Alle Eigenschaften der Anzeige (Zeichen pro Zeile, Höhe eines Zeichens, Zeilen pro Seite usw.) lassen sich einzeln programmieren. Neulich sah ich einen Schneider Joyce. Das wäre nicht unbedingt meine Maschine, aber seine 90 Zeichen pro Zeile sind besonders in der Textverarbeitung günstig.

Alle Computer in unserem Club, die einen CRTC haben, können das auch. Dazu brauchen nur seine ersten vier Register entsprechend geladen zu werden. Die Inhalte der übrigen Register werden einfach vom Format 80 X 25 Zeichen übernommen.

Register 0	(Bildbreite in Zeichen):	70h	(112d)
1	(angezeigte Zeichen pro Zeile):	5Ah	(90d)
2	(horizontale Sync-Position):	5Dh	(93d)
3	(Breite des Sync-Impulses):	0Eh	(14d)

Die Bildbreite im Register 0 läßt neben den angezeigten Zeichen noch einen Rand, damit in den Bildecken kein Zeichen verschwindet. Die Werte der Register 2 und 3 wurden experimentell ermittelt, um ein möglichst zentriertes Bild zu erhalten. Je nach Monitor können hier Änderungen oder auch Justagen an den Reglern erforderlich sein.

Das Register 6 (angezeigte Zeilen) kann vernachlässigt werden, solange sein Inhalt mindestens 16h (22d) beträgt. Der Bildwiederholspeicher ist nämlich 2 kB = 2048 Bytes groß. 2048÷90≈22,76. Da der Controller keine halben Zeilen bearbeitet, ist am Ende der 22. Zeile eben Schluß, gleichgültig, ob das Register 6 schon Alarm geschlagen hat oder nicht.

Von hier an kann ich nur noch für das G3s sprechen, weil mir andere Maschinen zum Ausprobieren nicht zur Verfügung stehen.

Die Werte für das Format 80 X 25 stehen ab 3760h. So müssen zum Umschalten auf 90 X 22 Zeichen in die Speicherstellen 3760h-3763h die oben genannten Werte geschrieben, anschließend der DOS-Befehl 80 eingegeben werden. Natürlich kann man auch den CRTC direkt programmieren, indem man über Port F6h das interne Register selektiert und danach über Port F7h den Inhalt ausgibt. Es ist auch möglich, die neuen Werte in die Stellen 37F0h-37F3h zu schreiben und danach CLS auszugeben. Viele Wege führen nach Read-Only Memory. Und alles das ist problemlos in BASIC machbar.

Arnulf Sopp

Futter für den CRTC

Die in unserem Club vertretenen Computer, die den Videocontroller (cathode ray tube controller, CRTC) 6845 haben, können das Bildschirmformat sehr flexibel einstellen. In meinen beiden Beiträgen, in denen es um eine ASCII-Tabelle im Bildschirm geht (mal speziell für TSCRIPS, mal allgemein für's DOS) wurde der CRTC jeweils für einen Sonderzweck speziell programmiert. Im Alltag sind viele Fälle denkbar, wo ein bestimmtes Bildschirmformat wünschenswert ist, das nicht vom DOS angeboten wird. Dann jedesmal den Controller zu programmieren, ist mühsam.

Der Library-Befehl ## springt in SYS22/SYS ein. Dort checkt er den Befehlsstring auf ein paar Argumente. Wenn er nicht solche findet, deren Bearbeitungsroutine in SYS22 liegt, verzweigt er weiter nach SYS26. Diese SYS-Datei hat an ihrem Ende glücklicherweise noch sehr viel Platz, so daß dort Routinen für zusätzliche Argumente untergebracht werden können. Der niedrigste Buchstabe, der als ##-Zugabe zulässig ist, ist H (für "halbes" Format, also 16 X 64 Zeichen). Es liegt nahe, die sieben Buchstaben A-G für zusätzliche Bildschirmformate zu nutzen. Der Platz in SYS26 ist so reichlich, daß auch noch die CRTC-Parametertabelle für sieben zusätzliche Formate hineinpaßt.

Das hier gelistete Programm fängt die ##-Routine dort ab, wo auf den Parameter T geprüft wird. Der Befehl JP Z,4D9F wird überschrieben mit einem Jump zur weiteren Prüfung. Dort wird zunächst der überschriebene Befehl nachgeholt, denn ##,T muß weiterhin möglich bleiben. Nun erfolgt die Prüfung auf die neuen Parameter A-G. Wenn einer davon eingegeben wurde, wird er nun in die laufende Nummer eines CRTC-Parametersatzes aus der Tabelle umgerechnet. HL wird als Zeiger auf diesen Satz geladen. Die 16 Parameter werden sodann nach 37F0-37FF transferiert, wo bei G-DOS 2.4 der gerade aktive Parametersatz abgelegt ist. Es folgt ein CALL nach 01C9, womit der Bildschirm gelöscht wird. Gleichzeitig mit CLS (genauer: mit HOME, das in CLS enthalten ist) werden die CRTC-Parameter aus 37F0 ff. ausgelesen und an den Controller geschickt. Das neue Bildschirmformat ist damit eingestellt.

Es kann durchaus sein, daß andere Tandy-kompatible Computer mit dem 6845 einen neuen Parametersatz auf andere Weise aktivieren. Hier muß der Leser sein Manual zu Rate ziehen oder die entsprechende Routine des Betriebssystems belauern. Es wäre interessant, die Ergebnisse im nächsten Info nachlesen zu können.

Zurück zum Programm. Es erschien sinnvoll, die Änderung eines Bildschirmformats ebenso einfach zu machen wie in OVL4/SYS (Genie 3s). Dort finden sich die CRTC-Parameter im ersten Sektor so angeordnet, daß sie mit DDE oder SUPERZAP genau je eine Bildschirmzeile beanspruchen. Aus diesem Grunde folgen dem RET-Befehl 6 NOPs, um die Tabelle für DDE linksbündig zu haben. Das verhilft zu besserer Übersicht. Die Tabelle selbst ist im Programm mit Nullen gefüllt. Im Hexdump sieht man echte Parameter. Die ersten drei Sätze sind Eigenschöpfungen, auf die ich gleich eingehen möchte, die nächsten vier sind vorerst noch die DOS-eigenen. Sobald Bedarf für ein weiteres Format besteht, ist das mit DDE schnell erzeugt.

Das Format, das sich mit ##,A einstellen läßt, hat 16 X 80 Zeichen. Es eignet sich besonders gut für Zwecke wie die oben erwähnte ASCII-Tabelle. Das Format B hat 16 X 64 Zeichen, aber mit 16 Scan-Zeilen pro Zeichen. Der Vorteil dieses Formats liegt darin, daß die Zeichen dichter aussehen, man sieht nicht mehr jeden einzelnen Matrixpunkt. Noch wichtiger ist, daß die Programmierung der HRG dabei wegen der lückenlosen Abfolge der HRG-Speicherabschnitte sehr vereinfacht wird. Das Format C schließlich ist 16 X 80 mit 16 Scan-Zeilen, also eine Kombination aus beidem. Je nach Computer müssen keineswegs alle Parameter übertragbar sein. Es kann erforderlich werden, die richtigen Werte auszuprobieren.

Im Hex-Dump (SYS26/SYS, relativer Sektor 04) ist die Routine für ##,A-G unterstrichen. Die Tabelle mit den 7 CRTC-Parametersätzen ist eingerahmt. Der Patch im ersten Sektor, der auf diese Routine umleitet, lohnt keinen eigenen Sektordump. Beginnend beim relativen Byte 83h ist die Folge CA-9F-4D in C3-33-51 zu ändern.

Arnulf Sopp

```
                ;Zap in SYS26/SYS, um 7 weitere CRTC-Parametersätze zugänglich zu machen
                ;
                ;                         (C) 1987 Arnulf Sopp

4D7F                       ORG     4d7fh          ;bisher JP Z,4D9Fh
4D7F  C33351               JP      extensn        ;Umleitung auf die Erweiterung

5133                       ORG     5133h          ;ab hier wieder Platz in SYS26
5133  CA9F4D      extensn  JP      Z,4d9fh        ;Überschriebenen Befehl nachholen
5136  FE41                 CP      'A'            ;CRTC-Parametersatz A?
5138  381A                 JR      C,exit         ;falls nein
513A  FE48                 CP      'H'            ;bis max. Parametersatz 6
513C  3016                 JR      NC,exit        ;falls kein gültiges Zeichen
513E  D641                 SUB     'A'            ;aus Zeichen Binärzahl machen
5140  07                   RLCA                   ;*16 (je 16 Parameter pro Satz)
5141  07                   RLCA
5142  07                   RLCA
5143  07                   RLCA
5144  215C51               LD      HL,param       ;Beginn der Parametertabelle
5147  85                   ADD     A,L            ;Start-LSB + laufende Nummer
5148  6F                   LD      L,A            ;HL <- Beginn des Parametersatzes
5149  11F037               LD      DE,37f0h       ;dort aktiver Parametersatz abgelegt
514C  011000               LD      BC,0010h       ;16 Codes pro Parametersatz
514F  EDB0                 LDIR                   ;Satz übertragen
5151  CDC901               CALL    01c9h          ;CLS, neues Videoformat aktivieren
5154  AF          exit     XOR     A              ;Z-Bedingung: kein Fehler
5155  C9                   RET                    ;Ente

                ;Füllbytes, damit die Tabelle im Sektor am Byte x0 beginnt (linsbündig)
5156  00                   DB      00h,00h,00h,00h,00h,00h

                ;Tabelle von 7 Parametersätzen, je nach Geschmack auszufüllen
515C  0000        param    DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;A
516C  0000                 DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;B
517C  0000                 DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;C
518C  0000                 DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;D
519C  0000                 DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;E
51AC  0000                 DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;F
51BC  0000                 DW      0000h,0000h,0000h,0000h,0000h,0000h,0000h,0000h ;6

0000                       END

00000 Fehler
```

```
DRV    00  020A 121A 030B 131B 040C 141C 050D 151D
0      10  01EA 0051 ED73 4E37 F331 FE3B DBF9 F5E6     Q sN7 1 ;
0H     20  3ED3 F9CD 804D F1D3 F931 0000 C9E5 2600 >     §   1    &
       30  CD38 377C 3C32 5337 AFFB E1C9 3A80 38CB  878<2S7     : 8
DRS    40  47CA E14D 3E5D EFCA 9F4D FE41 381A FE48 G  M>Ü   M A8  H
155950     3016 D641 0707 0707 215C 5185 6F11 F037 0  A     !öQ o  7
617H60     0110 00ED B0CD C901 AFC9 0000 0000 0000
       70  6E50 580A 1406 1012 020E 2909 0300 0000 nPX        )
       80  6E40 500A 1302 1011 020F 2909 0400 0000 n§P        )
       90  6E50 580A 1302 1012 020F 2909 0300 0000 nPX        )
       A0  6E50 560C 1F02 191C 0209 2909 0000 0000 nPV        )
       B0  6E40 500A 1406 1012 020E 2909 0400 0000 n§P        )
FRS    C0  6E40 500A 1B09 181A 020A 2909 0200 0000 n§P        )
4      D0  6E40 500A 1302 1011 030F 2F0F 0000 0000 n§P        /
4H     E0  0000 0000 0000 0000 0000 0000 0000 0000
       F0  0000 0000 0000 0000 0000 0000 0202 044D                  M
```


m4/4p ecke

Nachtrag zum Programm DISASSEM/BAS für Model 4

Wie so üblich beim Abtippen eines Listings, haben sich auch in diesem Programm einige Fehler eingeschlichen, auf die mich ein aufmerksamer Anwender hingewiesen hat:

Zeile	falsch	richtig
220	B(I)=","	B(I)="A,"
280	Z>-327691!	Z>-32769!
360	LEN(G(K))>2	LEN(G(K))<2
450	STR$(J1 AND 7),1)	STR$((J1 AND 56)/8),1)
720	J1057	J1=57
730	(IX,IND)	(IX+IND)

Klaus Hermann

Password-Bypass bei TRSDOS 6.2

TRSDOS 6.x nervt mich, der ich auf diesem Betriebssystem nicht besonders sattelfest bin, immer wieder mit seiner Passwordfunktion. Ständig meldet es irgendwelche Files, die ich selbst und ohne Password erstellt habe, als durch ein solches geschützt. Dies läßt sich mit folgendem Patch für immer ausschalten!

PATCH SYS2/SYS.LSIDOS (D02,33=18:F02,33=28)
PATCH SYS0/SYS.LSIDOS (D00,83=80:F00,83=00)

Quelle: 80 Micro, March 1987 Seite 25

Mehr Spaß mit TRSDOS 6.2 wünscht

Hartmut Obermann

Model 4 Boot-ROM Version und Datum

Durch Drücken der Taste "V" und anschließendes Drücken der RESET-Taste wird die ROM-Version und das Datum angezeigt.

Klaus Hermann

Model 4 - CP/M: RESET-feste MEMDISK!

Für das Model 4/4p gibt es von Montezuma mehrere Versionen des bekannten CP/M 2.2. Die Unterschiede zwischen den Versionen liegen praktisch nur im Bereich des BIOS (Schnittstelle zwischen dem CP/M von Digital Research Inc. und der Hardware). Die zur Zeit auf dem deutschen Markt aktuelle BIOS-Version 2.22 (in USA soll es schon eine neuere geben!?) stellt eine Routine zur Verfügung, die automatisch erkennt, ob der Rechner mit der 128k-Speichererweiterung ausgerüstet ist, und das zusätzliche RAM sogleich als MEMDISK initialisiert. Dies ist gegenüber den älteren BIOS-Versionen, bei denen diese Funktion durch Zusatzprogramme gelöst wurde, schon ein erheblicher Fortschritt!

Einen erheblichen Nachteil haben beide Arten, die MEMDISK zu initialisieren. Dieser besteht darin, daß nach einem Kaltstart (einmal RESET genügt) der Inhalt der MEMDISK unwiderruflich gelöscht ist. Diesen Makel kann man mit dem folgenden Patch beseitigen.

Das BIOS 2.22 stellt als ersten Schritt der MEMDISK-Initialisierung fest, ob überhaupt eine zweite Speicherbank vorhanden ist. Ist dieser Test positiv, wird zuerst die (Speicher-)Bank 1 und dann die Bank 2 angewählt und "formatiert". Erst dann wird, in einer weiteren eigenständigen Routine, dem CP/M mitgeteilt, daß ein Laufwerk M: vorhanden ist! Um die Kompatibilität des Systems zu wahren, durfte an diesem Prinzip nichts geändet werden! Aus diesem Grunde wurde folgender Weg eingeschlagen:

1. noch vor der überprüfung, ob überhaupt zusätzliches RAM vorhanden ist, wird überprüft, ob die Taste "N" während des Bootvorganges gedrückt wurde
2. wurde die "N" nicht gedrückt, so wird der Initialisierungvorgang ganz normal fortgesetzt
3. wurde "N" betätigt, so wird sowohl der Test, ob Zusatz-RAM vorhanden ist, als auch die "Formatierung" übersprungen und direkt bei der Eintragung des zusätzlichen Speichermediums in das CP/M-System weiter gemacht!

Das Assemblerprogramm ist sehr einfach zu realisieren:

```
EB53 CD6FED    CALL   KBSCAN     ;Tastatur abfragen
EB56 FE4E      CP     'N'        ;wurde "N" gedrückt?
EB58 CA9CEA    JP     Z,NFMT     ;nicht Formatieren!
EB5B 210000    LD     HL,0000H   ;HL auf 0000 setzen (dies ist
                                 ;die Anweisung, die durch den
                                 ;Sprung zu dieser Routine
                                 ;überschrieben wurde!)
EB5E C37DEA    JP     BACK       ;zurück zur ursprünglichen
                                 ;Routine!
```

Um die Abfrageroutine unterbringen zu können, wurde die Systemmeldung im Track 1, Sector 13 etwas gekürzt (siehe Sectordump). Die Änderungen kann man mit einem Diskettenmonitor (DU, DISKMON usw.) vornehmen. Um die Routine auch in den normalen Ablauf einzubinden, muß noch im Sector 11, Track 1, ab relativ Byte xxxx aus 21 00 00 ein C3 53 EB gemacht werden. Die Track- und Sectornummern beziehen sich auf eine Systemdiskette 80/DS/DD; die zu ändernden Stellen sind aber auch in anderen Systemen schnell zu finden!.

Nach durchgeführter Änderung kann man getrost RESET'ten, ohne Angst um seine Daten und Programme auf der MEMDISK haben zu müssen. Allerdings darf man nicht vergessen "N" während des BOOT-Vorganges "N" zu drücken!!!

Viel Spaß mit CP/M und der MEMDISK wünscht


```
1000  7263 6820 496E 632E 150D 0A42 494F 5320  rch Inc....BIOS
1010  7665 7273 2032 2E32 3220 2863 2920 2870  vers 2.22 (c) (p
1020  2920 3139 3834 204D 6F6E 7465 7A75 6D61  ) 1984 Montezuma
1030  204D 6963 726F 2F4A 424F 1516 0D0A 0A0D   Micro/JBO......
1040  3E3E 3E20 4D65 6D6F 7279 2044 7269 7665  >>> Memory Drive
1050  204D 3A20 1645 4E41 424C 4544 160D 0A0A   M: .ENABLED....
1060  0031 0001 CD07 F3CD E0EC 0E00 CD78 F101  .1...........x..
1070  0A00 097E 2366 6F7E 2366 6F22 34F7 2100  ...B#foB#fo"4.!.
```

m4/4p ecke

oben: der Originalsector
unten: die Änderungen sind unterstrichen!

```
1000  7263 6820 496E 632E 150D 0A42 494F 5320  rch Inc....BIOS
1010  7665 7273 2032 2E32 322B 2028 6329 2870  vers 2.22+ (c)(p
1020  2920 3139 3834 204D 6F6E 7465 7A75 6D61  ) 1984 Montezuma
1030  204D 6963 726F 2F4A 424F 1516 0D0A 0A00   Micro/JBO......
1040  4D45 4D44 4953 4B20 4D3A 2016 4F4B 2116  MEMDISK M: .OK!.
1050  0D0A 00CD 6FED FE4E CA9C EA21 0000 C37D  ....o..N...!...ü
1060  EA31 0001 CD07 F3CD E0EC 0E00 CD78 F101  .1...........x..
1070  0A00 097E 2366 6F7E 2366 6F22 34F7 2100  ...B#foB#fo"4.!.
```

```
0F00  C34B EAC3 61EB C3D0 EBC3 F2EB C302 ECC3  .K..a...........
0F10  1AEC C33E ECC3 52EC C3E5 F1C3 78F1 C393  ...>..R.....x...
0F20  F1C3 9CF1 C3DD F1C3 EDF1 C324 F2C3 2CEC  ...........$..,.
0F30  C3E2 F181 0B03 20FD F655 F661 F66D F679  ...... ..U.a.m.y
0F40  F673 EC9A EE8D F024 F172 F131 0000 CDD3  .s.....$.r.1....
0F50  ECAF 3204 0032 3FEB 3D32 5EF6 326A F632  ..2..2?.=2^.2j.2
0F60  76F6 3282 F621 9CF5 0110 0022 FDF6 0922  v.2..!....."..."
0F70  FFF6 0922 01F7 0922 03F7 2100 003E EFD3  ..."..."..!..>..
```

In diesem Sector ist nur aus 21 00 00 (LD HL,0000) ein Sprung zur neuen Routine (JP EB53 = C3 53 EB) zu machen!

```
0F00  C34B EAC3 61EB C3D0 EBC3 F2EB C302 ECC3  .K..a...........
0F10  1AEC C33E ECC3 52EC C3E5 F1C3 78F1 C393  ...>..R.....x...
0F20  F1C3 9CF1 C3DD F1C3 EDF1 C324 F2C3 2CEC  ...........$..,.
0F30  C3E2 F181 0B03 20FD F655 F661 F66D F679  ...... ..U.a.m.y
0F40  F673 EC9A EE8D F024 F172 F131 0000 CDD3  .s.....$.r.1....
0F50  ECAF 3204 0032 3FEB 3D32 5EF6 326A F632  ..2..2?.=2^.2j.2
0F60  76F6 3282 F621 9CF5 0110 0022 FDF6 0922  v.2..!....."..."
0F70  FFF6 0922 01F7 0922 03F7 C353 EB3E EFD3  ..."..."...S.>..
```

Indikator-Karte für den ECB-Bus

Das ECB-Bus-System nimmt seit den ersten Auseinandersetzungen im Club ja schon beachtliche Umfänge an. So möchte ich als Clubmitglied und Info-Drucker auch einmal einen Beitrag zum ECB-System beisteuern.

Alle bisher erschienenen Karten sind mit hochkomplizierter Elektronik bestückt und bewältigen fast unvorstellbare Aufgaben. Aber --- eine der für den User wichtigsten ECB-Karten wurde offensichtlich bis jetzt übersehen: Die INDIKATOR-KARTE. Sie funktioniert gänzlich ohne Softwareunterstützung, d.h. sie arbeitet sozusagen still im Hintergrund (Background working). Sie gibt dabei dem Benutzer eine eindeutige Kontrolle über die Energieversorgung des ganzen Systems. Dabei dient die rote LED (D1) zur Anzeige von vorhandenen 5V und die grüne LED (D2) für die 12V Versorgung.

Schaltungsbeschreibung

LED D1 wird über den Widerstand R1 betrieben. Dieser dient zur Herabsetzung der für die LED zu hohen Spannung von 5V. Vor der Anwendung von den Spannungen entsprechenden Glühlampen wird von meiner Seite dringend abgeraten. Diese würden durch ihre Altertümlichkeit das moderne Halbleiter-Konzept des Computers zerstören. Die Anzeige für die 12V-Versorgung funktioniert schaltungstechnisch genauso, wie schon am Anfang des Absatzes bei der 5V-Versorgung beschrieben. Als Unterschiede sind die LED-Farbe (grün) und der höhere Widerstandswert von R2 zu beachten. Dieser höhere Wert ist durch die um 7V höhere Spannung zu erklären. Auf Grund der nicht durchgeschleiften Interrupt-Daisy-Chain sollte die Karte als letzte in das System gesteckt werden.

Aufbau

Zum Aufbau der Karte ist nicht viel zu sagen; es sollten aber schon gewisse Elektronik-Kenntnisse vorhanden sein. Als erstes werden die beiden Widerstände eingesetzt und verlötet. Als nächstes folgt die 64-pol. Steckerleiste und zu guter letzt die beiden LED's; auf die richtige Polung ist zu achten.

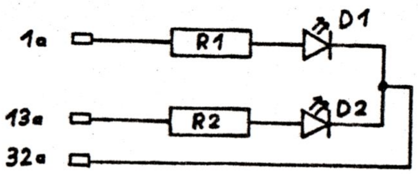
Inbetriebnahme

Die Karte ist, wie schon früher erwähnt wurde, als letzte in das System zu stecken. Abgleicharbeiten sind nicht notwendig. Nach dem Einschalten des Computers müssen beide LED's aufleuchten. Auf evtl. Rauchentwicklung ist zu achten. Aufsteigender Rauch zeigt mit 99% Sicherheit einen Kurzschluss an.

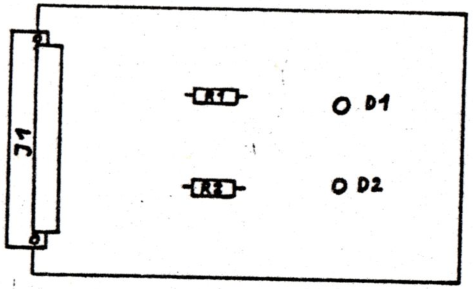
Eine fertig gebohrte und mit Bestückungsaufdruck versehene Platine kann zum Preis von 84,72 DM + MwSt. von mir bezogen werden. Ich wünsche allen Nachbauern viel Erfolg.

****
* Euer BitPit *
****

Schaltplan Indikator-Karte



Bestückungsplan Indikator-Karte



Bestückungsliste Indikator-Karte

Halbleiter

D1 LED 5mm rot
D2 LED 5mm grün

Widerstände

R1 220 Ohm 0,5 Watt
R2 470 Ohm 0,5 Watt

Sonstiges

J1 64 pol.- Steckerleiste
1 Platine

HEFT 19
Juni
1987

Floppycontroller Kompatibilitaet

Es war einmal - so koennte man die Geschichte der Tandy Floppy Controller (und Kompatible) beginnen lassen.
Aber ich will hier keine Maerchen erzaehlen, sondern auf einige Eigenschaften dieser Floppycontroller hinweisen.
In grauer Vorzeit wurde von der Fa. Tandy fuer das Modell I ein Floppycontroller entwickelt. Damals wurde der Controllerchip WD-1771B von Western Digital ausgewaehlt, ein zu damaliger Zeit toller Chip, der viel aber nicht alles konnte (welcher Chip kann das schon). Wie den meissten von euch bekannt sein duerfte, ist dieser WD-1771B nicht der Lage *"Double Density"* zu fahren.
Als nun die Technik weiter fortgeschritten war, und Tandy das Modell III auf den Markt brachte, hatte diese Maschine einen Double Density Contoller mit dem WD-1793 Chip. Es kamen nun auch sogenannte *"Doubler"* fuer das Modell I auf den Markt, bei denen eine Huckepackplatine in die Fassung des WD-1771B gesteckt wurde und auf dieser Platine befanden sich dann zwei Controller, naemlich der WD-1771B und der WD-1791. Der WD-1791 leistet das gleiche wie der WD-1793, er besitzt allerdings einen invertierten Datenbus, waehrend der WD-1793 einen normalen (nicht invertierten) Datenbus hat. Somit war jetzt fuer Singel Density und fuer Double Density je ein eigener Controller vorhanden.
Da aber der WD-1791 auch Single Density verarbeiten kann, kam die Fa. TCS (der Importeur des Video Genie I) auf die Idee einen Disk-Controller nur mit dem WD-1791 zu bauen. Damit waren dann aber die Probleme der Kompatibilitaet geschaffen.
Solange man auf diesen Floppy-Stationen ein Double-Density-DOS (z.B. GDOS, NEWDOS-80 etc.) faehrt, gibt es keine Probleme. Auch ein Single Density NEWDOS-80 laeuft bei mir problemlos. Schwierigkeiten gibt es aber wenn man versucht z.B. TRSDOS 2.3 zu laden. Das geht nicht. Weiterhin gibt es Probleme Disketten in Single Density zu lesen, die auf einem Rechner mit WD-1771B oder *"Doubler"* formatiert und beschrieben wurden. Obwohl die PDRIVE-Parameter richtig eingestellt wurden, ergab der Directory Befehl die Fehlermeldung *"Lesefehler Inhaltsverzeichnis"*.
Nachdem ich nun das "DOS - Buch" von L. Roeckrath und die Datenblaetter des WD-1771B und des WD-1791 durchgesehen hatte, wurde mir auch klar warum es diese Probleme gibt. Beim Formatieren und Schreiben von Daten auf die Diskette wird in jeden Sektor eine sogenannte 'DATA ADRESS MARK' (DAM) geschrieben. Mit Hilfe dieser DAM's unterscheidet das DOS dann die normalen Datensektoren von den Directory - Sektoren. Als DAM kommen beim WD-1771B die Werte 'F8h', 'F9h', 'FAh' und 'FBh' in Frage, beim WD-1791 nur die Werte 'F8h' und 'FBh'. Die Werte 'F9h' und 'FAh' kann der WD-1791 nicht verarbeiten. Und hier liegt nun auch der Grund fuer den "Lesefehler Inhaltsverzeichnis".
Normale Datensektoren werden von beiden Controllerchips mit der DAM 'FBh' gekennzeichnet. Die Directory-Sektoren werden vom WD-1771B mit der DAM 'FAh', und vom WD-1791 mit der DAM 'F8h' gekennzeichet. Da aber der WD-1791 mit der DAM 'FAh' nichts anfangen kann, kommt es zu diesem Fehler.
Eine Moeglichkeit dies Problem zu loesen, ist das sogenannte "Lesschuetzen" der Directory. "Lesschuetzen" ist zwar ein grauenhafter Ausdruck, aber so nennt sich dieser Vorgang nun einmal.

1.) Im GDOS oder NEWDOS-80 wird im System - Befehl der Parameter 'BN' auf 'Y' gesetzt. Daraufhin wird das DOS veranlasst eine Single-Density-Directory mit der DAM 'F8h' zu schreiben.
2.) Man benutzt das Programm "DIRCHECK/CMD" von GDOS oder NEWDOS-80. Dies Programm liest das Directory und gibt gegebenenfalls den Hinweis auf falsche Adressmarkierungen im Inhaltsverzeichnis. Mit der Option 'W' wird die Directory dann mit korrigierten DAM's wieder auf die Disk geschrieben
3.) Mit "SUPER UTILITY'. In der Option 'Repair' gibt es auch eine Unteroption 'Write Protect Directory". Hier werden auch die DAM's korrigiert und auf Disk geschrieben.

Diese drei Punkte funktionieren allerdings nur mit Datendisketten. Mit DOS - Disketten geht es zwar auch, aber das DOS laesst sich dann nicht mehr booten. Hierzu verweise ich auch auf den Beitrag von Bernd Drohwaelder ueber die ECB-Bus Floppy Controllerkarte mit dem WD-1791. Da das TRSDOS 2.3 z.B. die DAM 'FAh' als Directorykennzeichnung erwartet, kann es nun die Directory nicht mehr finden. Als einziger Ausweg bleibt dann nur noch auf die Mitarbeit eines solchen DOS's zu verzichten, was beim TRSDOS wahrscheinlich zu verschmerzen ist.
So, ich hoffe nun alle Klarheiten ueber die Unterschiede des WD-1771B und des WD-1791 in Bezug auf die "DATA ADRESS MARK's" beseitigt zu haben.
Wer trotzdem etwas von diesem Thema begriffen hat, dem kann dieser Artikel vielleicht etwas helfen. Um es noch eimal ganz klar zu sagen, die beschriebenen Probleme treten nur bei Disk-Controller-Karten auf, welche nur den WD-1791 benutzen.

Bernd Retzlaff

Tut mir leid mit Ihrer Gehaltserhöhung, Schmidt, aber der Computer will sie nicht genehmigen.

80-40 Track Umschaltung

Dieses Thema wurde im Info schon öfters behandelt. Jedoch ist es bei jedem Laufwerk anders. Hier möchte ich die Umschaltung von 80 auf 40 Track, für das Teac Laufwerk FD 55 FV-13 vorstellen.

Um die Laufwerke auf 40 Track umzustellen muß der Widerstand R19 (10 Ohm) entfernt werden,damit eine Unterbrechung entsteht. Der Widerstand lässt sich am besten entfernen wenn die Platine ausgebaut wird.Dazu müssen zuerst die fünf Stecker entfernt werden. Zwei befinden sich auf der Bestückungsseite und drei auf der Lötseite. Ein Stecker auf der Bestückungsseite lässt sich nicht ganz abziehen. Nun werden die drei Befestigungsschrauben entfernt, die Platine kann nun vorsichtig angehoben werden und der Stecker vollens abgezogen. Zur Umschaltung von 80 auf 40 Track wird ein einpoliger Schließer verwendet. In die Zuleitung ist der 10 Ohm Widerstand wieder einzubauen.Der Schalter ist so anzuschließen, daß einmal die Brücke offen ist (40 Track) und einmal die Brücke über den 10 Ohm Widerstand geschlossen. das 80 Track Format funktioniert auch ohne den Widerstand jedoch wird durch einen zu hohen Strom die ICs mit der Zeit in mitleidenschaft gezogen.

Eckehard Kuhn

Joystick am Model 4/4p

Ein schon mehrfach angeschnittenes Thema ist der Anschluß eines Joystick an die verschiedenen Tandy- und Genie-Modelle. Hier nun eine Kürzestanleitung zum Anschluß an ein Model 4/4p.

Sowohl beim 4 als auch beim 4p ist die Tastatur mit der CPU-Platine über ein 20-adriges Kabel mit Stecker verbunden. Um die Joystickfunktionen den Cursortasten zuzuordnen sind folgende Verbindungen herzustellen: Joystickbuchse / Tastaturstecker

(1)	<oben>	(12)
(2)	<unten>	(13)
(3)	<links>	(14)
(4)	<rechts>	(15)
(8)	<gemeinsam>	(7)

Der "Feuer"-Knopf kann sowohl der ENTER- als auch der SPACE-Taste zugeordnet werden.

(6)	<ENTER>	(9) oder
(6)	<SPACE>	(16).

Durch die Verwendung eines Umschalters für diese Funktion ist man erheblich flexibler!

Hartmut Obermann

Doubler mit WD2793
Verbessertes Datenlesen

Ein zuverlässigeres Lesen der Daten von der Diskette mit dem vor kurzen beschriebenen Selbstbau-Doubler auf Basis des WD2793 läßt sich durch folgende kleine Änderung erreichen:

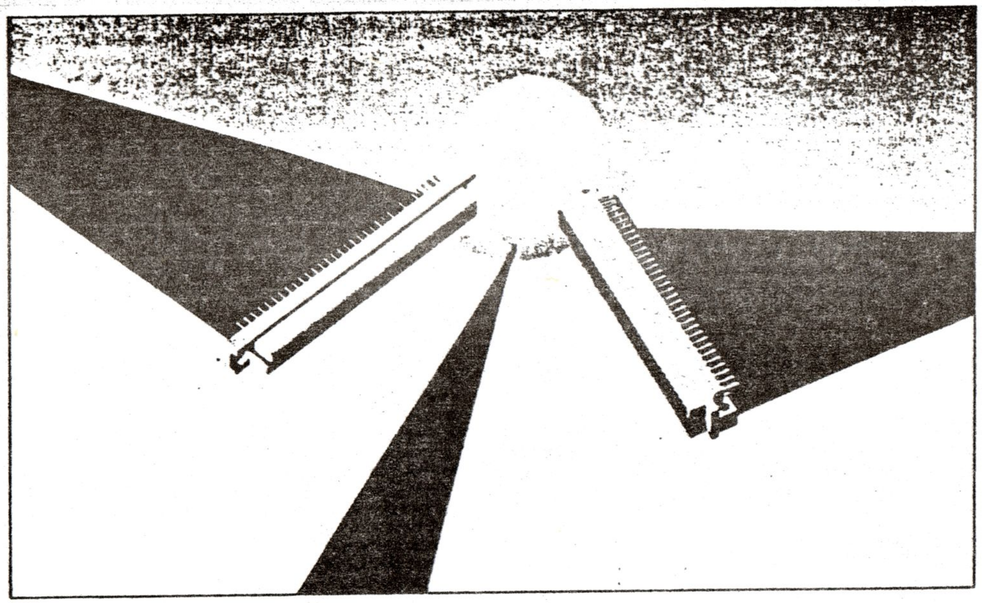
1) Auf dem Doubler-Board wird die Verbindung zwischen Pin 13 des 74LS123 und Pin 27 des WD2793 durchtrennt. Der Pin 27 des WD2793 wird stattdessen mit Pin 4 des 74LS123 verbunden.

2) Auf dem EXP1-Board werden die Pins 13 und 12 des 74LS04 aus der Fassung gebogen bzw. durchtrennt (wenn das IC direkt eingelötet ist). Auf der Lötseite werden die Pins 13 und 12 des 74LS04 miteinander verbunden.

Durch diese Änderung wird die Gatterlaufzeit des überbrückten 74LS04-Inverters eingespart. Die Fehlerrate beim Lesen von der Diskette wird damit drastisch gesenkt.

Das Layout des Doublers ist dem unter Punkt 1) beschriebenen Patch beereits angepaßt. Das Baord wird in Zukunft nur noch in dieser Form geätzt. Der Patch gemäß Punkt 2) ist dadurch zwingend.

Helmut Bernhardt



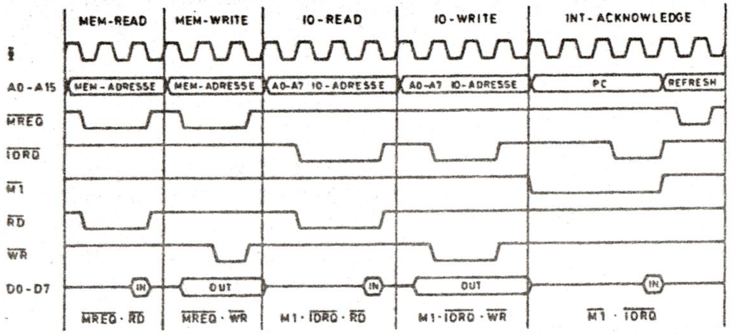


Bild 1: Timing Z80, stark vereinfacht. A0-A15 – Adressen, $\overline{MREQ}$ – Memory Request, $\overline{IORQ}$ – Input/Output Request, $\overline{M1}$ – Machine Cycle 1, $\overline{RD}$ – Read, $\overline{WR}$ – Write, D0-D7 – Daten.

Ankopplung am ECB-Bus

Z80-Bus-Anschluß

Georg Umbach

Es soll sie noch geben – jene Freaks, die tatsächlich noch zum Lötkolben greifen und sich spezielle Erweiterungen für ihre Computer selber 'stricken'. Sei es, daß es die Erweiterung in der gewünschten Art (noch) nicht gibt oder daß es sich gar um die Entwicklung eines Prototyps handelt. Zumindest sind zum Eigenentwurf genaue Kenntnisse des verwendeten Bus-Systems erforderlich.

Alle Mikrocomputer herkömmlicher Art haben drei Bus-Systeme: einen Datenbus, einen Adreßbus und einen Steuerbus. Auf dem Datenbus werden die Nachrichten als Bitmuster zwischen den einzelnen Komponenten des Computers übertragen. Auf dem Adreßbus liegt, als Bitmuster kodiert, die 'Anschrift' der Komponente, die außer der CPU an der jeweiligen Operation beteiligt ist. Auf dem Steuerbus wird angezeigt, welcher Art die Informationsübertragung ist, also ob ein Lese- oder Schreibzugriff auf den Speicher oder auf I/O-Ports erfolgen soll.

Timing

Die CPU (oder vorübergehend ein anderer Sonderbaustein) bestimmt das Geschehen auf den Bus-Systemen. Soll zum Beispiel ein Zugriff auf den Speicher erfolgen, legt die CPU zuerst die Adresse auf den Adreßbus. Anschließend wird das Signal $\overline{MREQ}$ gesetzt, wodurch der Speicherzugriff gekennzeichnet wird. Mit den Signalen $\overline{RD}$ und $\overline{WR}$ zeigt die CPU an, ob der Speicher gelesen oder beschrieben werden soll.

Die I/O-Zugriffe geschehen ähnlich wie Speicherzugriffe, jedoch mit dem Unterschied, daß das Signal $\overline{IORQ}$ aktiv wird und nicht das Signal $\overline{MREQ}$. Ein weiterer Unterschied zum Speicherzugriff ist, daß eine I/O-Adresse lediglich acht Bit breit ist. Diese Signalzusammenhänge sind im Timing-Diagramm dargestellt, das allerdings stark vereinfacht wurde.

Zwischen der CPU und dem Adreßbus sind, zumindest bei größeren Systemen, Pufferbausteine geschaltet, da die Ausgänge der CPU nur wenige andere Bausteine treiben können. Bei kleineren Einplatinensystemen kann diese Pufferung meist entfallen, da die Anzahl der an die CPU angeschlossenen Bausteine gering ist. Diese Puffer müssen durch die CPU zum System-Bus hin hochohmig geschaltet werden können, wenn das System DMA-fähig sein soll und der DMA-Baustein nicht auf der CPU-Platine sitzt (DMA – Direct Memory Access, direkter Zugriff auf den Speicher des Rechners). In solchem Falle wird die Adressierung des Systems vorübergehend nicht von der CPU, sondern von einem DMA-Contoller vorgenommen.

I/O-Adressierung

Alle Leitungen, die vom CPU-Bus an einer I/O-Karte (Baugruppe) anliegen, sollten gepuffert sein oder nicht mehr als einen TTL-Eingang versorgen. Dies reduziert die Störanfälligkeit, und der Bus wird zumindest kapazitiv weniger belastet. Allerdings werden durch die Pufferung die Laufzeiten etwas vergrößert, was aber meistens unbedeutend ist.

Entsprechend Bild 2 nun sollen vier Z80-I/O-Bausteine auf einer Karte betrieben werden. Jeder Baustein 'verbraucht' vier I/O-Adressen. Damit werden insgesamt 16 Adressen benötigt; die Z80-CPU kann über die Adreßleitungen A0 bis A7 aber 256 Ports adressieren. Also muß eine Dekodierung der in Frage kommenden Adressen erfolgen. Die Adreßleitungen A0 und A1 wählen den jeweiligen Port innerhalb des Bausteins aus, A2 und A3 selektieren den jeweiligen I/O-Chip über den Dekoder 74LS139. Weiterhin müssen A4 bis A7 zur Auswahl der Karte herangezogen werden. Hierfür ist das IC 74LS85 (4-Bit-Vergleicher) zuständig. Damit werden die Chip-Select-Signale für vier I/O-Bausteine nur aktiv, wenn die mit S1 bis S4 eingestellte Adresse angesprochen wird.

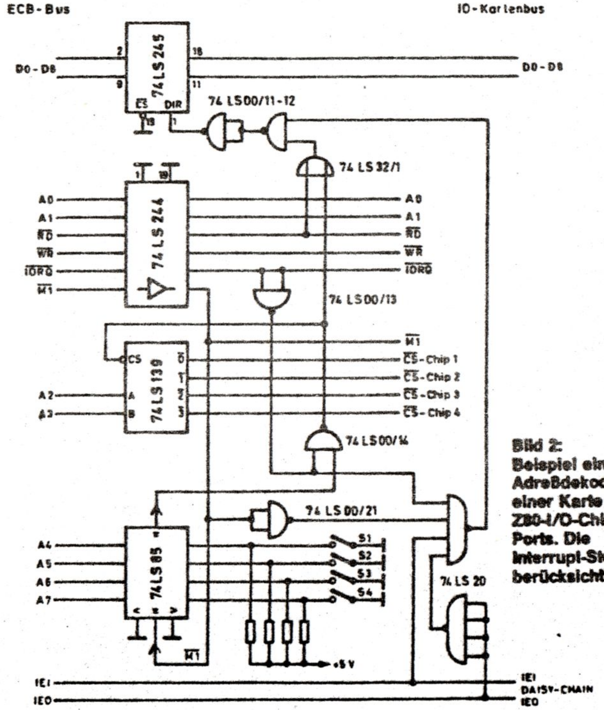


Bild 2: Beispiel einer Adreßdekodierung auf einer Karte mit vier Z80-I/O-Chips mit je vier Ports. Die Interrupt-Steuerung wurde berücksichtigt.

Zur Adressierung gehört aber auch die Steuerung des Datenbuspuffers 74LS245. Wird ein Baustein auf der Karte mit I/O-Lesen angesprochen, so muß die Datenrichtung umgeschaltet werden (74LS245 Pin1 – Direction). Durch das ODER-Gatter 74LS32/1 und die NAND-Gatter 74LS00/11-12-14 wird bei Kartenauswahl (74LS85 '='-Ausgang logisch 1) und dem Signal $\overline{RD}=0$ das Steuersignal für die Richtungsumschaltung des 74LS245 (Pin 1 = 0) gebildet.

Eine Besonderheit liegt in den Interrupt-Möglichkeiten des Z80-Prozessors. Im Interrupt-Modus 2 senden Z80-I/O-Bausteine während der Interrupt-Bestätigung ein Byte (den sogenannten Interrupt-Vektor) über den Datenbus zur CPU. Dieser wird zur Bestimmung einer Tabellenadresse herangezogen, wo die Adresse der Interrupt-Routine steht. Die Richtungsumschaltung des Datenbuspuffers erfolgt in diesem Betriebsfall aus der Verknüpfung der Signale $\overline{M1}=0$, $\overline{IORQ}=0$, IEI = 1 und IEO = 0. Die Signale IEI (Interrupt Enable Input) und IEO (Interrupt Enable Output) verbinden mehrere mögliche Interrupt-Quellen derart miteinander, daß bei korrekter Auswertung dieser Signale nicht mehrere Karten einen Interrupt-Vektor auf den Bus legen können.

Die Berücksichtigung des $\overline{M1}$-Signals bei der I/O-Dekodierung ist sehr wichtig, da während der Interrupt-Bestätigung und dem Einlesen des Vektors aus dem I/O-Chip eine Adresse auf dem Adreßbus liegt und ohne Berücksichtigung des $\overline{M1}$-Signals ein Datenpuffer auf einer anderen I/O-Karte, wenn diese Adresse zufällig 'paßt', falsch angesteuert werden könnte und so den Vektor auf dem Bus stören würde.

Dieses Beispiel zeigt, daß in Z80-Systemen die I/O-Dekodierung mit der zugehörigen Datenpuffersteuerung sorgfältig ausgeführt werden muß, da sonst sporadisch Störungen auftreten können.

Spiegelung

Man kann den Vergleicher 74LS85, die Adreß-Pullup-

Widerstände und die Schalter S1 bis S4 einsparen, wenn man die Karte zum Beispiel den festen Adressen 70h bis 7Fh zuordnet (Bild 3). Die genannten Bauteile könnte man zum Beispiel durch drei UND-Gatter 74LS08 ersetzen und die Adreßleitung A7 unberücksichtigt lassen. Die Karte ist damit unter den I/O-Adressen 7xh und Fxh zu erreichen. Man nennt diese Mehrdeutigkeit Spiegelung von Adreßbereichen. Diese Spiegelung ist nicht weiter von Bedeutung, solange das System nicht erweitert wird.

Fehler sind aber vorprogrammiert, wenn das System im Spiegelungsbereich Fxh mit neuen Ports erweitert wird. Die 'alte' Karte wird dann zusammen mit der neuen angesprochen, Fehlfunktionen sind die Folge. Unvollständige Dekodierung sollte man daher nur bei Systemen anwenden, die nicht erweitert werden, wie zum Beispiel Einplatinencomputer. Ein Beispiel hierzu zeigt Bild 4. Bei dieser Schaltung können ohne Adreßdekoderaufwand nur mit den Adreßleitungen A2 bis A7 sechs Bausteine mit je vier Ports angesprochen werden.

Chip 1:	111110xx	F8H - FBH
Chip 2:	111101xx	F4H - F7H
Chip 3:	111011xx	ECH - EFH
Chip 4:	110111xx	CCH - CFH
Chip 5:	101111xx	BCH - BFH
Chip 6:	011111xx	7CH - 7FH

Dieses Beispiel zeigt, daß man nicht unbedingt Dekoder-ICs benötigt. Die Adreßleitungen können direkt als Select-Signal dienen, allerdings nur bei solchen I/O-Bausteinen, die zusätzliche Eingänge für Lesen, Schreiben und gegebenenfalls I/O-Zugriff haben. Bei der Schaltung nach Bild 4 überlappen sich die Spiegelungsbereiche. Werden durch Software I/O-Adressen erzeugt, die mehr als eine '0' in den höchsten sechs Bit enthalten, so kommt es zur gleichzeitigen Aktivierung mehrerer I/O-Chips. Man sieht hier sehr deutlich, daß bei vermindertem Hardware-Aufwand besondere Forderungen an die Software gestellt werden, die nicht durch Programmlogik zu begründen sind.

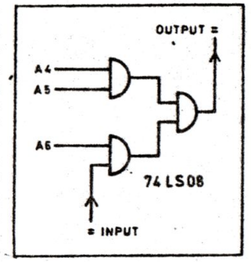


Bild 3: Die variable Adreßdekodierung durch ein IC 74LS85 und den DIL-Schalter wird durch diese fest verdrahtete Logik ersetzt. Es entstehen Spiegelungen, da die Adreßleitung A7 nicht berücksichtigt wurde.

Speicheradressierung

Mit ihren 16 Adreßleitungen kann die Z80-CPU 64 KByte (64 x 1024 = 65536) adressieren.

Bei EPROMs sind zur Zeit jedoch Speichergrößen von 4 KByte bis 32 KByte mit jeweils 8 Bit Datenbreite üblich, bei statischen RAMs sind es 2 KByte bis 32 KByte bei 8 Bit Datenbreite. Meist sind mehrere Speicherbausteine zusammenzuschalten, um den erwünschten oder den vollständigen Adreßraum von 64 KByte abzudecken. Wird von einem System beispielsweise 48 KByte EPROM-Bereich und 16 KByte RAM-Bereich gefordert, so kann man zum Beispiel sechs EPROMs und zwei RAMs mit je 8 KByte Speicherplatz verwenden. Die zugehörige Adreßdekodierung zeigt Bild 5. Aus den oberen drei Adreßleitungen A13 bis A15 werden mit dem Dekoder 74LS138 Signale gebildet, mit denen je einer von acht Speicher-Bausteinen ausgewählt wird, indem das IC mit dem Signal $\overline{CS}=0$ aktiviert wird.

In dem Beispiel 'Einplatinencomputer' (Bild 4) sind nur ein EPROM- und ein RAM-Baustein zu adressieren. Für die Unterscheidung von nur zwei Bausteinen reicht aber eine Adreßleitung, hier A15. Ist sie logisch 0, wird das EPROM angewählt, ist sie '1', wird das RAM aktiviert, sofern kein I/O-Zugriff ansteht. Auch hier sind wieder Spiegelungen zu erwarten, wenn EPROM und RAM je kleiner sind als 32 KByte. Ein EPROM vom Typ 2732 (4 KByte) kann vollständig mit den Adressen A0 bis A11 angesprochen werden, A12 bis A14 bleiben unberücksichtigt. Damit erscheint solch ein EPROM achtfach gespiegelt, da drei Bit nicht ausgewertet werden – so ist zum Beispiel die Adresse 0xxxh ebenso gültig wie 7xxxh.

Signal		Pin
Stromversorgung +5V	+5V	1a
Datenleitung 5 E/A	D5	2a
Datenleitung 6 E/A	D6	3a
Datenleitung 3 E/A	D3	4a
Datenleitung 4 E/A	D4	5a
Adreßleitung 2 A	A2	6a
Adreßleitung 4 A	A4	7a
Adreßleitung 5 A	A5	8a
Adreßleitung 6 A	A6	9a
CPU-Steuersignal E	/Wait	10a
DMA-Steuersignal E	/BUSRQ	11a
DMA-Daisy-Chain E	(BAI)	12a
Stromversorgung +12V	+12V	13a
Adreßleitung 19 A	(A19)	14a
Stromversorgung -12V	(-12V)	15a
2 x Taktsignal A	2<phi>	16a
DMA-Daisy-Chain A	(BAO)	17a
Adreßleitung 14 A	A14	18a
Stromversorgung +15V	+15V	19a
CPU-Steuersignal A	/M1	20a
Adreßleitung 22 A	(A22)	21a
Adreßleitung 23 A	(A23)	22a
	(RETIRQ)	23a
Backup-Akku	(VCMOS/UBAT)	24a
Taktsignal A	<phi>	25a
	(/WRITEN)	26a
CPU-Steuersignal A	/IORQ	27a
CPU-Steuersignal A	/RFSH	28a
Adreßleitung 13 A	A13	29a
Adreßleitung 9 A	A9	30a
DMA-Steuersignal A	/BUSAK	31a
Masse	GND	32a

Signal		Pin
Stromversorgung +5V	+5V	1c
Datenleitung 0 E/A	D0	2c
Datenleitung 7 E/A	D7	3c
Datenleitung 2 E/A	D2	4c
Adreßleitung 0 A	A0	5c
Adreßleitung 3 A	A3	6c
Adreßleitung 1 A	A1	7c
Adreßleitung 8 A	A8	8c
Adreßleitung 7 A	A7	9c
Adreßleitung 16 A	(A16)	10c
Interrupt-Daisy-Chain E	IEI	11c
Adreßleitung 17 A	(A17)	12c
Adreßleitung 18 A	(A18)	13c
Datenleitung 1 E/A	D1	14c
Stromversorgung -15V	-15V	15c
Interrupt-Daisy-Chain A	IEO	16c
Adreßleitung 11 A	A11	17c
Adreßleitung 10 A	A10	18c
Adreßleitung 21 A	(A21)	19c
CPU-Interrupt E	/NMI	20c
CPU-Steuersignal E	/INT	21c
CPU-Steuersignal A	/WR	22c
Adreßleitung 20 A	(A20)	23c
CPU-Steuersignal A	/RD	24c
CPU-Steuersignal A	/HALT	25c
Peripherie-Reset A	/PWRCL	26c
Adreßleitung 12 A	A12	27c
Adreßleitung 15 A	A15	28c
Taktsignal A	<phi>	29c
CPU-Steuersignal A	/MREQ	30c
CPU-Steuersignal E	/RESET	31c
Masse	GND	32c

Der am häufigsten verwendete Bus für Z80-Steckkartenrechner ist der ECB-Bus. Alle CPU-Signale sowie die Stromversorgungsleitungen liegen auf diesem Bus. 'Erfunden' wurde der ECB-Bus von Kontron-Mitarbeitern, die ihn wie hier abgebildet konzipiert haben. Für verschiedene Anwendungen wählten andere Hersteller eine modifizierte Belegung, die die in Klammern gesetzten Signale betreffen kann.

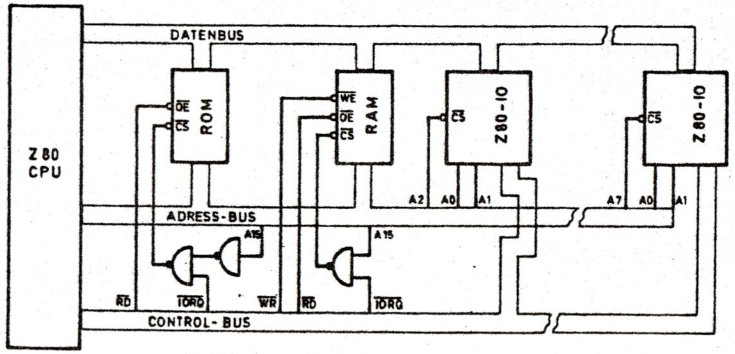


Bild 4: Ein Einplatinencomputer mit minimalem Adreßdekodieraufwand für I/O- und Speicherzugriffe. Es sind bis zu sechs I/O-Bausteine mit je vier Ports angeschlossen.

A15————A0
0000 xxxx xxxx xxxx
0111 xxxx xxxx xxxx

Dynamische Speicher

Warum einzelne Speicherbereiche ausdekodieren, wenn doch die CPU genau 64 KByte adressieren kann – dann schaltet man doch einfach acht dynamische Speicher mit je 64 KByte und je einem Datenbit parallel, und alle Probleme mit Dekodierung und Spiegelungen sind gelöst. Das ist leider nur zum Teil richtig, denn neue Probleme entstehen.

Ein dynamischer Speicher besteht aus einer quadratischen Matrix, die bei den erwähnten Bausteinen aus 256 × 256 Speicherkondensatoren gebildet wird. Diese Zellen müssen immer gelesen und wieder zurückgeschrieben (refreshed) werden, um den Speicherinhalt zu erhalten, was eine besondere Adressierung notwendig macht.

Zuerst werden acht Adreßbits an die Adreßpins der ICs gelegt. Dieser Teil der Adresse wird im RAM-IC durch die negative Flanke des RAS-Signals (Row Address Strobe) gespeichert. Entsprechend dieser acht Adreßbits wird eine ganze Zeile von 256 Datenbits ausgewählt und in ein Datenzwischenregister gebracht. Zeitparallel werden an die Adreßpins weitere acht Adreßbits angelegt. Diese werden durch die negative Flanke des CAS-Signals (Column Address Strobe) in einem speicherinternen Register zwischengespeichert. Mit dieser zweiten Adreßhälfte wird aus dem Datenzwischenregister das entsprechende Datenbit (Spalte) ausgewählt, das zum Datenausgang gelangt.

Erfolgt statt eines Lesezugriffs auf den Speicher eine Schreiboperation, wird hingegen das entsprechende Bit im Datenzwischenregister, adressiert durch die Spaltenadresse, gesetzt beziehungsweise gelöscht.

Der Inhalt des Datenzwischenregisters wird nach dem Speicherzugriff in die alte Zeile zurückgeschrieben und dort wie-

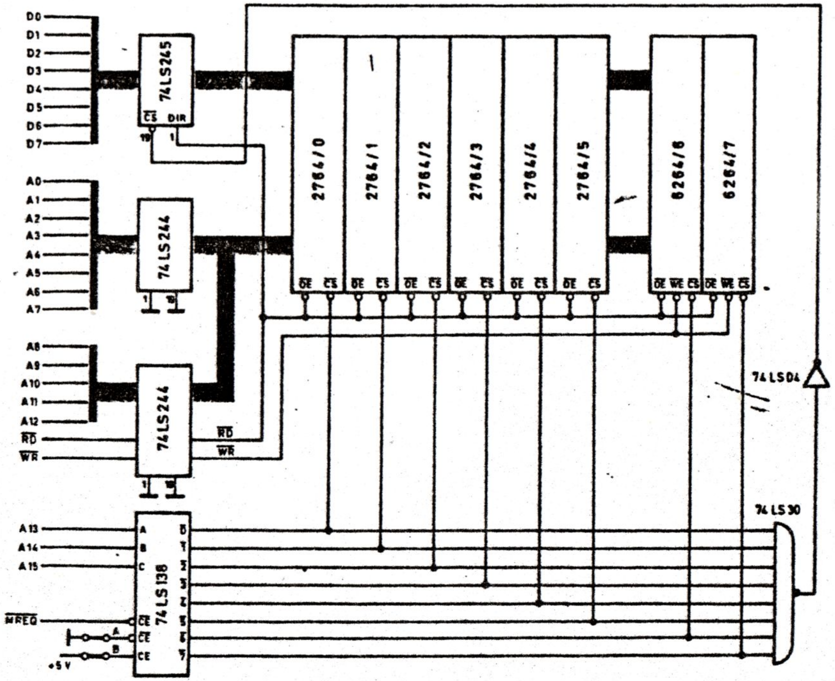


Bild 5: Eine Speicherkarte mit Adreßdekodierung für acht Speicherbausteine.

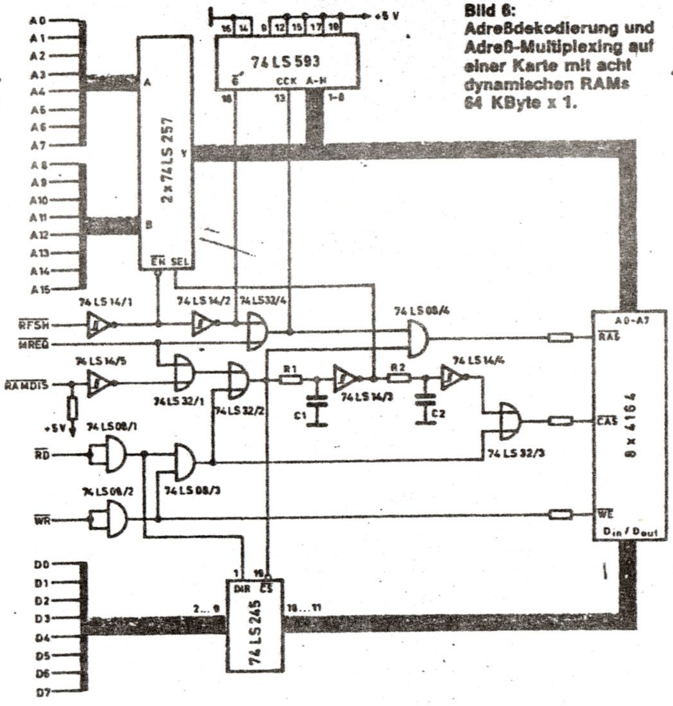


Bild 6: Adreßdekodierung und Adreß-Multiplexing auf einer Karte mit acht dynamischen RAMs 64 KByte x 1.

der in den 256 kapazitiven Speicherzellen gespeichert. Diese Zeile ist also 'refreshed' und eventuell durch 'Schreiben' aktualisiert.

Durch diese Funktionsweise des dynamischen Speichers ist hier eine besondere hardwaremäßige Adressierung erforderlich. Da beide Adreßhälften erst nacheinander gebraucht werden, haben die Speicherhersteller die beiden Adreßregister im Speicher-IC auf dieselben Anschluß-Pins gelegt. Die Einsparung von Anschlüssen und damit kleinere IC-Gehäuse haben allerdings einen erhöhten Aufwand an Adreßlogik zur Folge. Die Adressen müssen in zwei Hälften zerlegt und zeitmultiplex eingegeben werden.

Es tritt aber noch ein weiteres Problem auf: Auch wenn aus bestimmten Datenzeilen des dynamischen Speichers keine Daten benötigt werden, muß diese Zeile doch innerhalb bestimmter Maximalzeiten einmal gelesen und der Inhalt wieder zurückgeschrieben werden. Hierzu wird ein 'Refresh-Zyklus' benutzt, bei dem nur eine Adreßhälfte angegeben wird, was zum Lesen und Zurückschreiben einer kompletten 256-Bit-Datenzeile führt. Dies kann durch eine spezielle Hardware erfolgen, wenn die CPU nicht auf den Speicher zugreift. Die Z80-CPU zeigt dies mit einem speziellen Signal an ($\overline{RFSH}$). Gleichzeitig mit Erscheinen dieses Signals legt sie eine Refresh-Adresse auf die unteren sieben Bit des Adreßbusses – der Refresh-Vorgang geschieht also automatisch. Bei Speicher-ICs bis zu einer Größe von 64 KBit reichen die sieben Adreßbits für einen vollständigen Refresh aus. Bei den heute üblicherweise verwendeten 256-KBit-Chips benötigt man jedoch eine 8 Bit breite Refresh-Adresse.

Die vorgestellte Schaltung zum Adressieren eines dynamischen Speicherblocks ist ausgelegt für acht Speicher-ICs mit der Organisation 64 KByte × 1, zum Beispiel NEC-ICs 4164. Die Schaltung sieht keine spezielle Refresh-Methode vor, wie sie einige Speicherbausteine verlangen.

Die Signale $\overline{MREQ}$=0, $\overline{RAMDIS}$=1, $\overline{RD}$ oder $\overline{WR}$=0 leiten einen Speicherzugriff dadurch ein, daß ein $\overline{RAS}$-Signal erzeugt wird. Diese Verknüpfung erfolgt über das ODER-Gatter 74LS32/1-2, den Schmitt-Trigger 74LS14/5 und das UND-Gatter 74LS08/1-4. Der Eingang SEL der beiden Multiplexer 74LS257 liegt noch auf logisch 0. Damit sind die Adreßbits A0 bis A7 auf den Speicherblock geschaltet, und die negative Flanke von $\overline{RAS}$ verursacht die Speicherung von A0 bis A7 im RAM.

Nach Ablauf der Zeitkonstante R1∗C1 geht das Signal SEL auf logisch 1, es werden A8 bis A15 an die Speicher geschaltet. Nach einer weiteren Verzögerung, die Dauer wird bestimmt durch R2∗C2, wird das $\overline{CAS}$-Signal erzeugt. Mit seiner negativen Flanke werden A8 bis A15 im RAM gespeichert. Durch $\overline{RAS}$=0 und $\overline{CAS}$=0 werden die Datenleitungen des Speichers durchgeschaltet. Das ODER-Gatter 74LS32/3 sorgt später für eine schnelle Beendigung des Speicherzugriffs ohne Verzögerungen durch die Zeitkonstanten aus R1∗C1 und R2∗C2, die im Bereich einiger zehn Nanosekunden liegen.

Wird nun das Signal $\overline{RFSH}$ 'low', werden die Muliplexer 74LS257 hochohmig, und die acht Ausgänge des 8-Bit-Zählers 74LS593 werden als Refresh-Adresse an die Speicherchips geschaltet. Ist das Signal $\overline{MREQ}$ ebenfalls '0', wird ein $\overline{RAS}$-Signal gebildet, diesmal durch die Gatter 74LS14/1-2, 74LS32/4 und 74LS08/4.

Aus der Ansteuerlogik wird weiterhin noch das Freigabesignal für die Datenpuffersteuerung gewonnen. Gesperrt werden kann der Zugriff auf diesen Speicher durch RAMDIS=0, wodurch man zum Beispiel die RAMs während eines Boot-Vorgangs sperren kann.

Erzeugt zum Beispiel eine andere Speicherkarte mit EPROM-Bestückung ein Signal mit dem Pegel logisch 0, wenn sie aktiviert wird, so kann man mit diesem Signal den dynamischen Speicher für den Zugriff über den Eingang RAMDIS ausblenden.

Banking

Zur Zeit sind dynamishe RAM-Chips mit Kapazitäten von 256 KByte × 1 aktuell. Aber sie sind bald überholt, und es werden sich die 1 MByte-Chips etablieren. Durch Erweitern der Multiplexer in der Schaltung nach Bild 6 ist das Problem des größeren Adreßraums zu lösen. Aber 8-Bit-CPUs bleiben üblicherweise bei 16 Adressen und 64 KByte Adreßraum. Hier müssen nun weitere Adressen zu den vorhandenen hinzugefügt werden.

Diese zusätzliche Adreßerzeugung kann zum Beispiel durch das Einschreiben von weiteren Adreßbits in ein Ausgaberegister mittels eines I/O-Zugriffs erfolgen. Allerdings ist darauf zu achten, daß die CPU in allen 'Banks' ein Stück allgemeinen Adreßraum behält, damit sie sich beim Wechseln der Bank nicht den Speicher 'unter den Füßen' wegschaltet, in dem auch das gerade aktive Programm zur Bankumschaltung steht. Ein Schaltungsvorschlag dazu zeigt Bild 7.

Spricht die CPU die Adressen xFxxxH an, werden die Registerausgänge des 74LS374 hochohmig, und die Bankadressen A16 bis A23 werden durch Pulldown-Widerstände erzeugt. Diese Adressen sind also immer 00Fxxxh. In diesem Bereich müssen dann die Routinen zur Bankumschaltung, sämtliche Interrupt-Vektoren und Interrupt-Programme stehen, da nur dieser Adreßbereich per Software nicht umgeschaltet werden kann. Andernfalls besteht die Gefahr, daß die Interrupt-Routinen gerade zu dieser Zeit in einer anderen Bank stehen und das Programm nun 'ins Leere' läuft. Leider geht bei dieser Banking-Methode der 'blinde' Speicherplatz für das gemeinsame Segment in jeder weiteren Bank verloren.

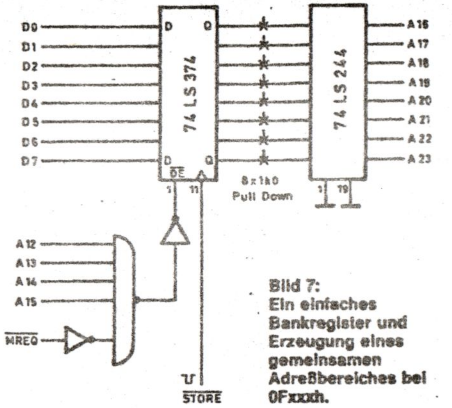


Bild 7: Ein einfaches Bankregister und Erzeugung eines gemeinsamen Adreßbereiches bei 0Fxxxh.

Segmente

Eine elegantere Art, mehr als 64 KByte zu adressieren, ist die Aufteilung des Speichers in Segmente von zum Beispiel je 4 KByte. Zur Adressierung solch eines Blocks werden die Adressen A0 bis A11 benötigt. Die höheren Adressen, also A12 bis A19, werden durch Auswahl von sechzehn 4-KByte-Segmenten aus 256 möglichen Segmenten erzeugt. Das Zauberwort heißt variable Umkodierung – eine Schaltung hierfür zeigt Bild 8.

Zwei schnelle Schottky-TTL-RAMs 74S189 sind mit ihren Dateneingängen an den Datenbus geschaltet. Die Adreßeingänge und Steuerleitungen der 74S189 liegen dazu parallel. Die Datenausgänge bilden A12 bis A19. Wird eine I/O-Adresse Fxh angesprochen, dann schaltet der Multiplexer 74LS157 die Adreßeingänge des 74S189 auf die Adressen A0 bis A3.

Die 16 × 8 Speicherplätze der RAMs wirken jetzt wie 16 I/O-Register, die einzeln setzbar sind. Zu berücksichtigen ist aber, daß die eingeschriebenen Daten invertiert auf den Adressen A12 bis A19 erscheinen. Daher ist ein invertierender Treiber 74LS240 erforderlich.

Nimmt man an, daß die folgenden Dateninhalte in den 16 Adreßplätzen stehen, so ist die Segmentlogik scheinbar nicht aktiv, da keine Umkodierung der Adressen A12 bis A15 stattfindet.

Adressen/ A15 - A12	Daten/ A19x - A12x
0000	0000 0000
0001	0000 0001
.	.
1110	0000 1110
1111	0000 1111

Für 16 Eingangskombinationen erscheinen auch nur 16 Ausgangskombinationen. Aber die Datentabelle kann per Programm teilweise neu gesetzt werden – andere Speichersegmente zu 4 KByte stehen jetzt bereit. Beispiel:

Adressen/ A15 - A12	Daten/ A19x - A12x
0000.	0001 0000
0001	0001 0001
.	.
1110	0001 1110
1111	0000 1111

Auffällig ist, daß das letzte Segment nicht umbenannt ist – es enthält die Software für die Segmentsteuerung, sämtliche Interrupt-Vektoren und Interrupt-Routinen. Auch hier gilt, wie schon bei 'Banking' erläutert, daß auf keinen Fall der CPU das Programm 'weggeschaltet' werden darf.

Bei diesem Umschalten müssen natürlich nicht alle möglichen Segmente neu definiert werden, es können auch weniger sein. Und wenn sie aus besonderen Gründen im Speicher nicht hintereinander, sondern ungeordnet liegen – bitteschön, dieses System ist flexibel. Ein weiterer Vorteil ist, daß nicht wie beim normalen Banking gemeinsame Speicherbereiche als Speicherplatz verlorengehen.

Beim Booten ist zu beachten, daß die beiden RAMs 74S189 noch nicht definiert sind. Nach einem Hardware-Reset wird das RS-Flipflop aus zwei NAND-Gattern 74LS00 zurückgesetzt, und der RAM-Eingang CE liegt auf logisch '1' – die Ausgänge sind hochohmig. Die acht Pull-up-Widerstände und die Inverter erzeugen dann die Adresse 00xxxh. Als erstes werden dann per Software die RAMs 74S189 wie im ersten Beispiel gesetzt. Dann wird eine der betreffenden I/O-Adressen gelesen. Das RS-Flipflop (74LS00) wird gesetzt, und es werden ganz normal 64 KByte adressiert. c't

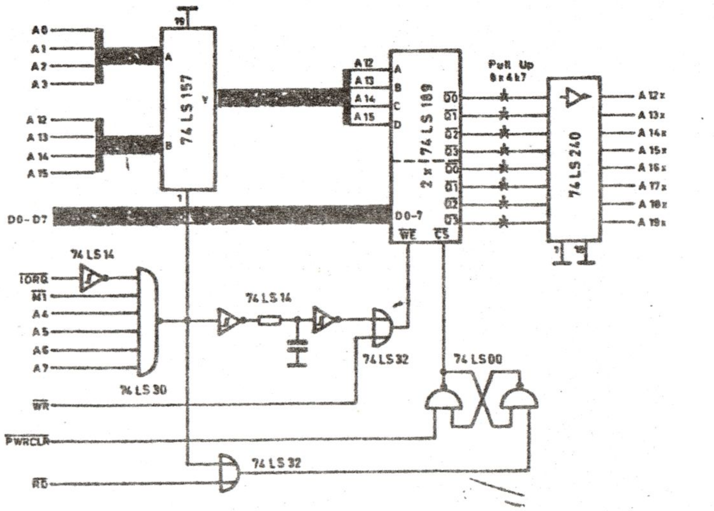
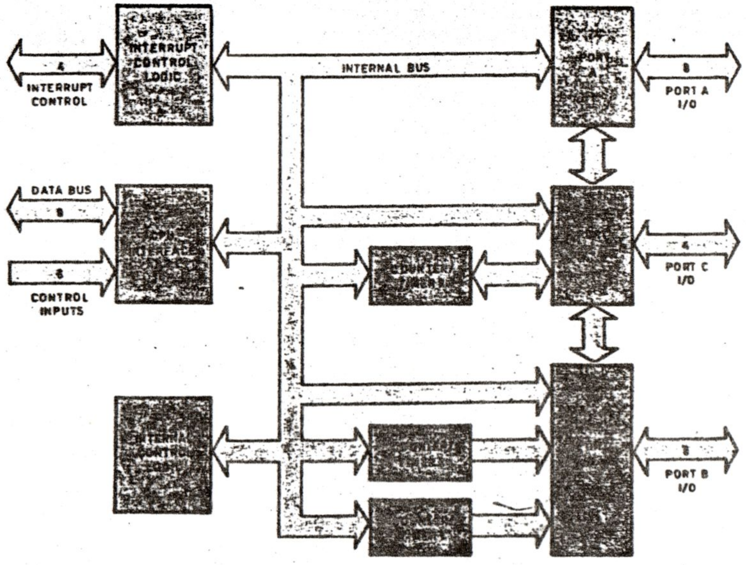


Bild 8: Erzeugung von Segmentadressen durch variable Umkodierung.



Universalgenie

Der Peripheriebaustein Z8536 CIO

Stefan Wimmer

Schnittstellen sind für ein Computersystem das Tor zur Außenwelt. Mit seinen drei Parallelports und drei Zählern bietet der hier vorgestellte Baustein schon fast alles, was in kleinen und oft auch größeren Rechnern an Ein-/Ausgabe erforderlich ist. Die zahlreichen Kombinationsmöglichkeiten der 'Innereien' sowie die Unterstützung von Vektor-Interrupts unterstreichen die Vielseitigkeit dieses Bausteins.

CIO steht für 'Counter/Timer and Parallel I/O'. Hinter dieser kurzen Charakterisierung verbirgt sich einer der universellsten Peripheriechips, die es derzeit am Markt gibt. Ein Blick auf die 'Highlights' aus der Bausteinbeschreibung genügt, um einen ersten Eindruck von der Vielseitigkeit zu bekommen:

- zwei unabhängige, doppelt gepufferte, bidirektionale 8-Bit-Universal-E/A-Ports
- ein bidirektionaler 4-Bit-Spezial-E/A-Port
- alle 3 Ports programmierbar in Polarität, Richtung und weiteren speziellen Parametern (1's catcher, Open-Drain-Treiber)
- vier Handshakemodes (einschließlich eines IEC-Bus-gleichen 3-Draht-Handshaking)
- Request/Wait-Signal für Hochgeschwindigkeits-Datenübertragung (DMA)
- flexible Mustererkennungslogik (programmierbar als 16-Vektor-Interrupt-Controller)
- drei unabhängige 16-Bit-Zähler/Zeitgeber mit mehreren Betriebsarten, wahlweise nachtriggerbar/nicht nachtriggerbar
- bis zu vier externe Zugriffsleitungen je Zähler/Zeitgeber
- fast alle internen Register beschreib- und lesbar
- alle Parallelport-Register direkt im (E/A-)Adreßbereich zugänglich
- durch universelles Bus-Interface an viele verbreitete CPUs anschließbar

Das Blockschaltbild des Z8536 CIO zeigt die verschiedenen 'Abteilungen': CPU-Interface, die drei Parallelports, die drei 16-Bit-Zähler/Zeitgeber, Interrupt-Kontrollblock und interner Kontrollblock. Eine große Anzahl Eingriffsmöglichkeiten erlaubt dem Anwender, sich den Baustein seinen Ansprüchen gerecht zu konfigurieren.

Rein und raus

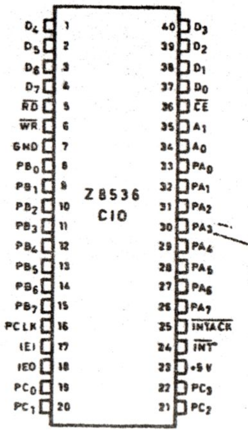
Die beiden 8-Bit-Universalports A und B sind identisch aufgebaut, mit der Ausnahme, daß Port B auch für die externen Zugriffsleitungen der beiden Zähler/Zeitgeber 1 und 2 herangezogen wird. Wie bei der guten alten Z80-PIO lassen sich beide Ports als Eingabe-, Ausgabe- oder als Bitport betreiben. Ebenso ist bidirektionaler Betrieb mit Handshaking möglich, hier hat man dann die Wahl zwischen vier verschiedenen Handshake-Modi. Schließlich können beide Ports auch zu einem 16-Bit-Port zusammengekoppelt werden. In allen Betriebsarten ist es möglich, eine Mustererkennung zu aktivieren, die beim Auftreten eines vorgebbaren Bitmusters einen Interrupt auslöst.

Port C unterscheidet sich in seinem Aufbau und seinen Möglichkeiten von den beiden anderen. Er stellt, falls erforderlich, die Handshake-Leitungen für Port A und B, wozu auch eine Request/Wait-Leitung zur Synchronisation der CIO mit DMA-Bausteinen beziehungsweise CPUs zählt. Auf diese Weise nicht 'verbratene' Portpins können dann noch entweder als E/A oder aber als externe Anschlüsse des dritten Zählers/Zeitgebers verwendet werden.

Zwar fehlen Port C die Mustererkennungslogik und ein paar andere Besonderheiten der Ports A und B, dafür bietet er aber ein anderes 'Special': Er kann bitadressiert beschrieben werden, was soviel heißt, daß man gezielt den Pegel einzelner E/A-Pins verändern kann, ohne die anderen zu beeinflussen. Hierdurch kann man sich das Einlesen des Ports vor dem Setzen oder Löschen des gewünschten Bits sparen.

Heute so, morgen so,...

Die Betriebsart 'Bit Port' funktioniert wie bei der 'alten' PIO: Eine '1' im Datenrichtungsregister definiert den entsprechenden Pin als Eingang, eine '0' als Ausgang. Falls einzelne Pins für Zähler/Zeitgeber benötigt werden, sind sie in der vorgeschriebenen Richtung zu programmieren. Außer der Datenrichtung kann man mit einem weiteren Register (Data Path Polarity Register) festlegen, welche Pins invertiert werden und welche nicht. (Alle im folgenden getroffenen Aussagen beziehen sich auf nichtinvertierte E/A-Pins.)



Wird ein Eingabepin gelesen, so erhält man gewöhnlich den Wert, der gerade ansteht. Man kann aber auch einen sogenannten 'Ones Catcher' einfügen: Erkennt dieser 'Einsenfänger' an seinem Eingang eine '1', dann setzt er seinen Ausgang auf '1' und behält diesen Zustand bei, bis er wieder gelöscht wird. Dies erfolgt durch Schreiben einer '0' auf den Eingang. (In allen anderen Fällen wird das Schreiben auf Eingabepins ignoriert.)

Enthalten die Ports A und B Ausgabepins, so erhält man beim Lesen das entsprechende Bit des Ausgaberegisters. Lesen von Port C liefert dagegen immer den Zustand, der an den Pins ansteht. Dieser Unterschied wird dann interessant, wenn über das 'Special I/O Control Register' der Ausgang nicht als normaler Gegentaktausgang, sondern als 'open Drain' (entspricht 'open Collector') spezifiziert wurde und damit von weiteren, externen 'Open'-Ausgängen übersteuert werden kann ('wired AND').

Die schon erwähnte Bitadressierung beim Schreiben von Port C wird dadurch erreicht, daß man mit den oberen vier Bits des C-Datenregisters festlegen kann, welche der unteren vier Bits beim Schreiben verändert werden dürfen. Man kann sich so eine Art Writeprotect-Maske zusammenstellen.

Für die Ports A und B stehen vier Handshake-Modi zur Verfügung: Interlocked, Strobed, Pulsed und 3-Wire. Dabei stellt Port C die Handshake-Leitungen, und die Ports werden doppelt gepuffert. Das heißt, daß ein zweites Datum ein- beziehungsweise ausgegeben werden kann, bevor das erste abgeholt worden ist. Falls erforderlich, kann man den Doppelpuffermodus auch abschalten.

Im unidirektionalen Betrieb wird das 'Interrupt Pending'-Statusbit (IP) gesetzt, wenn der Ein- oder Ausgabeport von der Peripherie bedient wird, und durch einen CPU-Zugriff auf das Datenregister wieder gelöscht. Bei einem bidirektionalen Port dagegen muß die CPU dieses Bit explizit zurücksetzen. Die doppelte Pufferung läßt sich dahingehend ausnutzen, daß das IP-Bit erst dann gesetzt beziehungsweise gelöscht wird, wenn jeweils zwei Datenbytes übertragen worden sind. Die Port-Bedienroutine transferiert dann pro Aufruf 16 Bit, was besonders in Verbindung mit einer 16-Bit-CPU von Vorteil sein kann.

Wie bereits angeführt, kann man die Ports A und B zu einem 16-Bit-Port zusammenschalten. Der kontrollierende Port, was die Steuer-/Statusregister, Handshaking und Mustererkennung anbelangt, ist in diesem Fall Port A; Port B muß als 'Bit Port' ohne Mustererkennung programmiert werden.

Beim Einsatz von Handshaking sind die zugehörigen Port-C-Pins als Eingang zu programmieren, ihre tatsächliche Richtung wird automatisch passend eingestellt. Der Inhalt des Polaritätsregisters von Port C bleibt dagegen voll gültig. Somit dürfte das Problem von 'falschen', weil invertierten Handshake-Signalen der Vergangenheit angehören.

... und übermorgen...

Zu den Handshake-Modi im einzelnen: Beim 'Strobed Handshake' markiert die negative Flanke des ACK-Signals den Zeitpunkt, zu dem die Daten in die CIO eingelesen werden (Eingabe) beziehungsweise vom Peripheriegerät abgeholt worden sind (Ausgabe). Die CIO versetzt daraufhin ihren RFD-/DAV-Ausgang (Eingabe: RFD = Ready for Data; Ausgabe: DAV = Data available) in den inaktiven Zustand, bis das Portregister wieder frei/gefüllt ist. Die positive ACK-Flanke hat keine Bedeutung.

Anders beim 'Interlocked Handshake'. Hier ist der RFD-/DAV-Ausgang so lange blockiert, bis das angeschlossene Gerät das ACK-Signal zurückgenommen hat (siehe Timing-Diagramm).

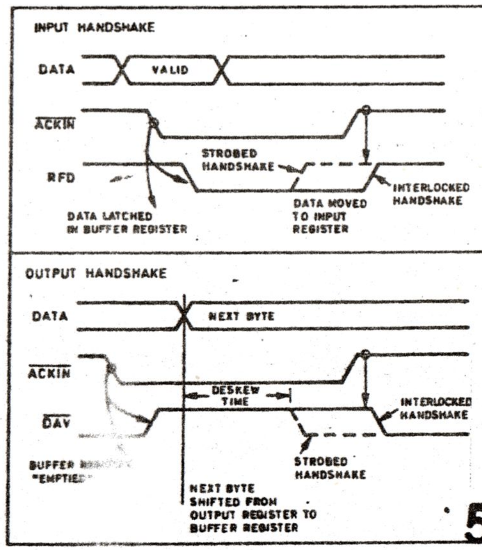
Das '3-Wire Handshake' ist eines der besten Bonbons, die die CIO zu bieten hat: Es ist nämlich identisch mit dem der IEEE-488-Schnittstelle (IEC-Bus). Im Prinzip funktioniert es wie das 'Interlocked Handshake', mit dem Unterschied, daß nun die positiven Flanken von zwei Signalen (RFD und DAC) angeben, ob Daten bereitstehen oder von allen angeschlossenen Empfängern übernommen sind. Da, wie der Name unschwer vermuten läßt, für das 3-Draht-Handshaking drei Leitungen gebraucht werden, kann immer nur ein Port damit arbeiten. Außerdem ist es im Bidirektional-Modus nicht verfügbar, was jedoch nicht allzu sehr ins Gewicht fällt, da man die Richtung eines Ports jederzeit per Software umdrehen kann.

Dem aufmerksamen Leser wird aufgefallen sein, daß ein Feature, das in den Timing-Diagrammen zu erkennen ist, noch gar nicht erwähnt wurde: die 'deskew time'. Hinter diesem Begriff verbirgt sich die Möglichkeit, bei einem Ausgabeport eine Verzögerung zwischen dem Anlegen neuer Daten an die Ausgabepins und dem Zeitpunkt einzufügen, zu dem sie mittels Handshake-Signal (DAV) für gültig erklärt werden. Hierzu steht pro Port ein separater 4-Bit-Zähler zur Verfügung, so daß bis zu 16 Taktzyklen eingeschoben werden können.

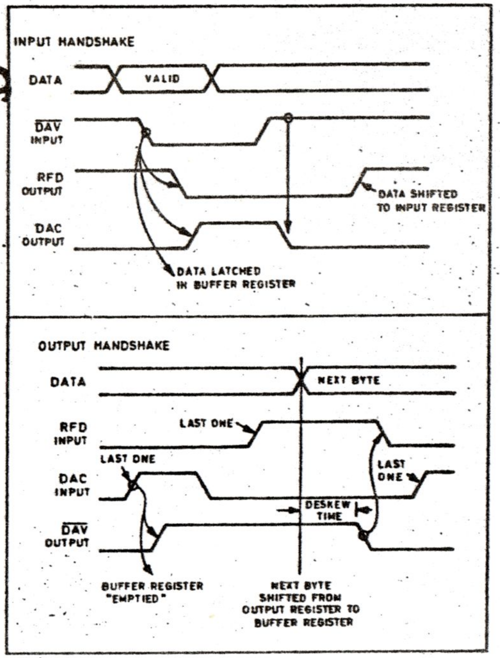
Dieses Verlängern der sogenannten 'Data Setup Time' wird mit der Betriebsart 'Pulsed Handshake' in noch viel größerem Rahmen sowie auch für Eingabeports möglich. Hier erfolgt die Verzögerung durch einen der drei Zähler/Zeitgeber, ansonsten funktioniert dieses Handshaking wie 'Interlocked'.

... ausgemustert?

Ähnlich vielfältig sind die Möglichkeiten der Mustererkennungslogik. Zu der von der PIO bekannten Variante, bei der ein Interrupt dann erzeugt wird, wenn die AND- beziehungsweise OR-Verknüpfung invertierter oder nicht invertierter



Die Impulsdiagramme für die Handshake-Arten 'Strobed' und 'Interlocked'.



Das 3-Leitungs-Handshaking der CIO ist IEC-Bus-kompatibel.

Eingänge '1' ergibt, sind hinzugekommen:

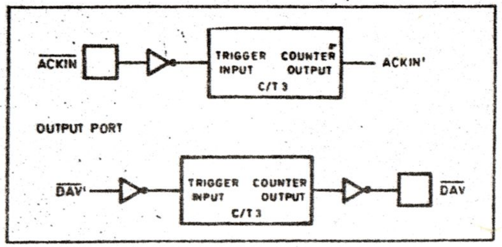
- 'Latch on Pattern Match'; die CIO speichert den Zustand der Portpins bei Auftreten der Interrupt-Bedingung, bis das IP-Statusbit gelöscht wird (etwa durch Lesen des Datenregisters). So können auch kurze Impulse 'eingefangen' werden, ohne den tatsächlichen Zustand des Ports zu verhüllen, wie es der '1's Catcher' tun würde (der natürlich auch zur Verfügung steht).
- Flankentriggerung; der Interrupt wird durch die steigende, die fallende oder jede Flanke eines Signals ausgelöst. Dadurch läßt sich verhindern, daß ein länger dauerndes Signal die Mustererkennung blockiert.
- 'OR-Priority Encoded Vector'; damit arbeitet die CIO als Interrupt-Controller, wobei jedem Portpin ein eigener Interrupt-Vektor zugeordnet wird. Da Port A und B ohnehin getrennte Vektorregister haben, sind bis zu 16 verschiedene Interrupts für wichtige Ereignisse oder 'dumme' (nicht vektorinterruptfähige) Peripheriebausteine möglich.

Die Mustererkennung arbeitet übrigens nicht nur bei Eingängen, sondern auch bei Ausgängen. Dies ist insofern interessant, als man dadurch beispielsweise bei Einsatz eines DMA-Controllers für den Datentransfer auf Steuerzeichen (für Blockende und ähnliches) reagieren kann.

Er-Zähler

Die drei Zähler/Zeitgeber sind identisch aufgebaut. Jeder besitzt einen 16-Bit-Rückwärtszähler, ein 16-Bit-Zeitkonstantenregister, ein 16-Bit-'Current Count Register' und zwei 8-Bit-Register zur Kontrolle und für den Status. Auf die 16-Bit-Register wird in zwei 'Portionen' zu je 8 Bit zugegriffen.

Auch die Zähler/Zeitgeber besitzen mehrere Möglichkeiten, nach außen zu wirken. Bis zu je vier E/A-Leitungen (Port B und C) können für externe Zugriffe



Beim Pulsed-Handshaking ist der Zähler 3 mit von der Partie.

programmiert werden: als Zählereingang, als Gate-Eingang, als Trigger-Eingang und als Zähler/Zeitgeberausgang. Der Ausgang des Zählers/Zeitgebers 1 kann intern auf einen der Eingänge von Zähler Nummer 2 geschaltet werden, die Möglichkeiten der Programmierung beider Zähler bleiben davon unberührt (außer daß der betreffende Eingang von Zähler 2 dann nicht mehr 'extern' sein darf).

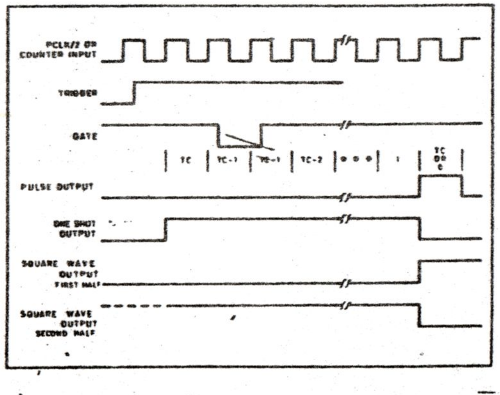
Die Funktionen der Ein- und Ausgänge sowie der Trigger-Eingänge bedürfen wohl keiner weiteren Erörterung; mit dem Gate kann zu jedem beliebigen Zeitpunkt die Taktquelle (extern oder Clock/2) gesperrt werden. Um bei laufendem Zähler den Zählerstand abzufragen, kann dieser mit einer '1' im 'Read Counter Control'-Bit (RCC) in das Current-Count-Register 'gelatcht' und dann in aller Gemütsruhe von der CPU abgeholt und verarbeitet werden. (Ohne dieses Kommando folgt der Inhalt des Current-Count-Registers unmittelbar dem Zählerstand.)

Jeder Zähler/Zeitgeber läßt sich für verschiedene Ausgangssignale programmieren: 'Pulse', 'Single Shot' oder 'Square-wave'. Natürlich kann man per Programm auch festlegen, ob ein Zähler kontinuierlich durchläuft oder nach einem Zyklus anhält, ob er nachtriggerbar ist oder nicht und ob er Interrupts auslösen darf.

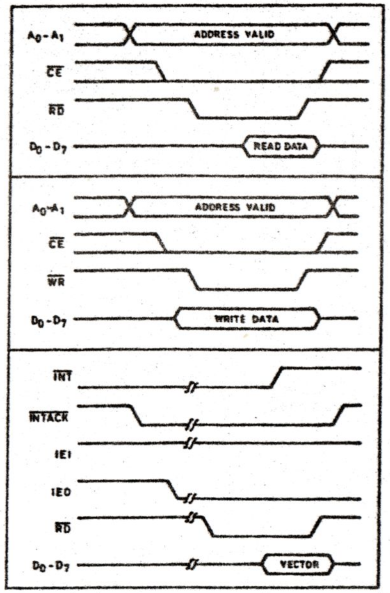
Programmpausen

Die CIO besitzt insgesamt fünf potentielle Interrupt-Quellen: die drei Zähler und die zwei Ports A und B. Die Priorisierung des Bausteins erfolgt – wie bei Zilog üblich – über die Daisy-Chain-Ein-/-Ausgänge IEI und IEO.

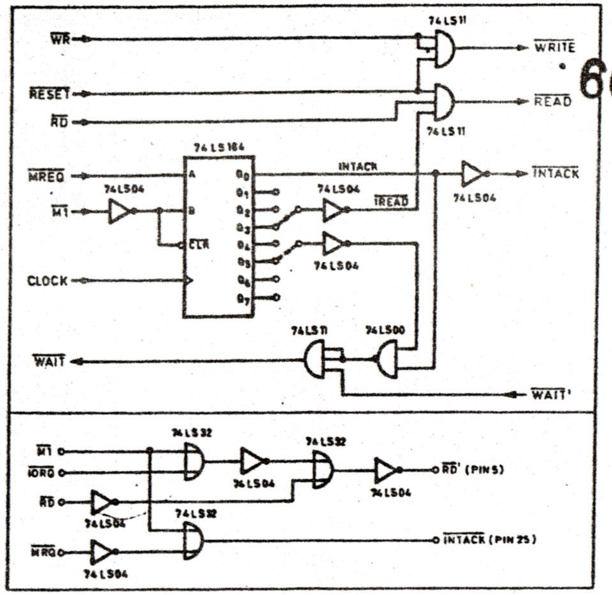
Jede Interrupt-Quelle hat in ihrem Kontroll- und Statusregister drei Bits zur Steuerung der Interrupt-Logik: das 'Interrupt Pending'-Bit (IP), das 'Interupt Under Service'-Bit (IUS) und das 'Interrupt Enable'-Bit (IE).



So verhalten sich die Zählerausgänge in den verschiedenen Betriebsarten.



Das Bus-Timing beim Interrupt-Acknowlodge weicht vom Z80-Timing ab, um die CIO auch an anderen CPUs einsetzen zu können.



Zwei Interfaces für den Anschluß an eine Z80-CPU: Das obere (nach einer Zilog-Applikation) ist auch für Systeme mit vielen Port-ICs geeignet, für kleinere Systeme reicht das untere.

IP wird gesetzt, wenn eine Interrupt-Bedingung vorliegt. Ist auch das IE-Bit gesetzt, darf die Quelle die INT-Leitung 'betätigen'. Quittiert die CPU die Anforderung (Interrupt-Acknowledge), so wird das IUS-Bit der höchstpriorisierten Interrupt-Quelle gesetzt, um die in der Daisy-Chain folgenden Quellen zu sperren. (Aus eben diesem Grund kann das IUS-Bit auch direkt von der CPU beschrieben werden.)

Zur Steuerung der Interrupts des ganzen Bausteins gibt es noch das 'Master Interrupt Control Register' (MIC). Mit diesem können die CIO-Interrupts global gesperrt oder erlaubt werden; ebenso besteht hier die Möglichkeit, den IEO-Ausgang unmittelbar zu beeinflussen und somit die gesamte restliche Interrupt-Kette 'abzuhängen'.

Für die 'Aufbewahrung' der Interrupt-Vektoren besitzt die CIO drei Vektorregister (zwei für die Ports A und B und eins für die Zähler). Auch hier läßt sich wieder einiges steuern: Es kann festgelegt werden, ob ein Basisvektor mit Statusinformationen der jeweiligen Interrupt-Quelle beladen oder ob überhaupt ein Vektor ausgegeben werden soll. Letzteres ermöglicht es, die CIO auch in Systemen einzusetzen, in denen kein Vektor-Interrupt zur Verfügung steht. Damit dann aber nicht erst alle Statusregister 'abgeklappert' werden müssen, um schließlich festzustellen, daß der Interrupt gar nicht von der CIO kam, hat Zilog noch das 'Current Vector Register' spendiert, wo die CPU auf der Suche nach dem Störer nachschauen kann, wer da etwas von ihr will. (War es nicht die CIO, enthält das Register den Wert 0FFh.)

Von außen

Von der CIO existieren zwei Versionen: Zum einen die Version Z8036 für Prozessoren mit gemultiplextem Daten/Adreßbus (Z-BUS-Version; auf diese möchte ich hier nicht näher eingehen) und zum anderen die Version Z8536 für nicht gemultiplexte Busse. Zilog hat dem zweiten Baustein nicht das gewohnte Z80-Bus-Interface mitgegeben, sondern will mittels einer universelleren Variante auch andere Prozessoren von der CIO profitieren lassen.

Sieht man sich einmal das Bus-Timing an, so ist beim Lese- und Schreibzugriff noch nichts Außergewöhnliches zu bemerken. Aber dann: Beim Interrupt-Acknowledge verlangt der Baustein ein aufbereitetes INTACK-Signal und danach das Lesesignal RD.

Um diese Signale aus den 'normalen' Z80-Signalen zu erzeugen, schlägt Zilog eine Schaltung mit drei Gatter-ICs und einem Schieberegister vor. Sicher eine saubere Lösung, vor allem, wenn viele Portbausteine im System sind, weil die Daisy-Chain dann viel Zeit braucht. Aber meistens geht es auch ohne so viel Aufwand (eventuell gar mit PROMs oder PALs). Gerade beim Entwurf von Einplatinenrechnern wird man die Einfachheit der zweiten abgebildeten Variante zu schätzen wissen.

In Anbetracht des tollen Komforts, den die Z8536-CIO bietet, muß noch die 'Gretchenfrage' des Systementwicklers geklärt werden: Wieviele Portadressen 'verbrät' dieser Superchip? Doch keine Angst – zwar beinhaltet der CIO-Baustein alles in allem 48 Register, diese werden aber durch einen kleinen Kunstgriff in nur vier I/O-Adressen untergebracht:

Unmittelbar erreichbar sind lediglich die Datenregister der drei Parallelports. Die anderen Register, die (hoffentlich) nicht so oft bemüht werden müssen, sind anzusprechen, indem man zuerst an die vierte Adresse die Nummer des gewünschten Registers ausgibt, um dann im darauffolgenden Schreib- oder Lesezugriff auf dieselbe Adresse die Daten zu transferieren. Für aufeinanderfolgende Lesezugriffe auf dasselbe Register (Polling eines Statusregisters) braucht man die Registernummer nur beim ersten Mal zu nennen.

Literatur

Z8036 Z-CIO, Z8536 CIO, Technical Manual, Zilog Inc., 1982

Application Notes:
Initializing the CIO (1982); Interfacing the Z-BUS Peripherals to the 8086/8088 (1982); Interfacing the Z8500 Peripherals to the 68000 (1982); Interfacing Z80 CPUs to the Z8500 Peripheral Family (1983).

c't

Weise die Vorteile der CMOS- und TTL-Technologie vereint wurden. Bereits bei 3 V Versorgungsspannung erreicht man die Geschwindigkeit von TTL-Chips (je geringer die Betriebsspannung, desto langsamer sind CMOS-Chips).

Zum anderen kommt Texas Instruments mit seiner ACL-Reihe (Advanced CMOS Logic) auf den Markt, jedoch ist die Produktpalette noch klein und nicht alle Daten endgültige Werte. ACLs sind aber im wesentlichen mit FACT vergleichbar. Die Markteinführung der AC-/ACT-Reihe verzögerte sich auch dadurch, daß TI (nach längerer Bedenkzeit) ein neues Pinout anstrebt (siehe Kasten).

Durch ihre in allen Belangen hervorragenden technischen Daten ist die (F)AC-/(F)ACT-Serie zur Zeit der härteste Konkurrent der ALS- und LS-Serie. Im Moment steht nur noch der höhere Preis für die CMOS-Chips ihrem Siegeszug im Wege. Wenn bei den bipolaren ICs nicht noch einmal ein gravierender Fortschritt erzielt werden kann, dürften sie in absehbarer Zeit ausgespielt haben; denn fast alle modernen hochintegrierten Chips (Gate-Arrays, 32-Bit-Prozessoren) werden in CMOS gefertigt. Und da ist es schon aus Kompatibilitätsgründen günstig, neue Systeme in einheitlicher Technologie aufzubauen.

Literatur

[1] The TTL Data Book for Design Engineers Volume 1 & 2, Texas Instruments, Ausgabe 1985

[2] Pocket Guide, Übersicht für Entwickler, Band 1, Texas Instruments, 1979, 1983 und 1985

[3] Eberhard Kühn, Handbuch TTL- und CMOS-Schaltkreise, Hüthig-Verlag, Berlin 1985

[4] TTL-Taschenbuch, Teil 1, IWT-Verlag, Vaterstetten 1983

[5] FAST – Fairchild Advanced Schottky TTL, Fairchild, 1982

[6] Advanced CMOS-Technology, Logik Data Book, Fairchild, 1985

[7] zu ACL (AC/ACT) von Texas Instruments: vorläufige Datenblätter und Applikations-Seminar 1986

[8] Handbuch der Elektronik, Digitaltechnik, Herausgeber: Institut zur Entwicklung moderner Unterrichtsmethoden e.V., Bahnhofstr. 10, 2800 Bremen 1, 1984

[9] Data Book High Speed CMOS, SGS, 1984

Kleines Logik-Lexikon

Bipolar und unipolar

Bipolar nennt man die herkömmlichen Dioden und Transistoren. Grob vereinfacht fließt hier Strom über Siliziumabschnitte mit positiver und negativer Dotierung, wodurch am eigentlichen Ladungstransport immer sowohl Elektronen als auch Elektronenlöcher beteiligt sind. Diese zwei (Vorsilbe 'bi-') Ladungsträger mit unterschiedlichen Polaritäten bezeichnet das Wort 'bipolar'.

Bei Bauelementen in Feldeffekt-Technologie fließt der zu steuernde Strom nur über Silizium mit einer Dotierung, wodurch nur eine ('uni-') Art von Ladungsträgern (Löcher oder Elektronen) vorhanden ist.

Durchlaufzeit

Bei der Angabe von Durchlaufverzögerungen muß man zum einen berücksichtigen, daß Flankenwechsel von low nach high und von high nach low unterschiedlich lang sein können, zum andern spielt natürlich die kapazitive Ausgangsbelastung eine Rolle. Wenn in Datenblättern nichts weiter angegeben ist, wird im allgemeinen eine kapazitive Belastung von 15 oder 50 pF bei einem für die jeweilige Technologie typischen Lastwiderstand zugrundegelegt. Die Verzögerungszeiten für HL- und LH-Wechsel werden üblicherweise arithmetisch gemittelt.

Manchen Leser wundert es vielleicht, daß bei den Logik-Chips um jede Nanosekunde gerungen wird, wo sich in Mikroprozessor-Systemen mit 4 MHz Taktfrequenz (1/f = 250 ns) doch alles in Größenordnungen von mehreren hundert Nanosekunden abspielt. Die Notwendigkeit kurzer Laufzeiten wird aber umgehend klar, wenn man bedenkt, daß sich die angegebenen Verzögerungswerte immer nur auf einzelne Gatter beziehen, nicht auf komplexere Schaltungen.

Schwärmen die Hersteller von AS-TTLs zum Beispiel davon, daß nunmehr Taktfrequenzen im Bereich von 200 MHz erreichbar sind, sinkt dieser Wert bei einem D-Flipflop (AS74) schon auf 105 MHz, bei einem synchchronen 4-Bit-Zähler wie dem AS169 auf 75 MHz.

Aber auch niedrige Laufzeiten auf Gatter-Ebene sind gelegentlich von Bedeutung, etwa um ein bestimmtes Timing bei der Ansteuerung von dynamischen RAMs einzuhalten. 'Richtig schnell' kann es aber auch in ganz normalen PCs zugehen, denn auf höchstauflösenden Grafik-Adaptern kommt man schon mal auf Bildpunkt-Raten, die Taktfrequenzen oberhalb 50 MHz erforderlich machen.

Schaltfrequenzen und Leistungsaufnahme

Die Leistungsaufnahme hängt ganz entscheidend von der Betriebsfrequenz ab. Bei jedem Schaltvorgang muß einerseits kurzzeitig ein erhöhter Strombedarf zum Umladen der parasitären Kapazitäten gedeckt werden. Zum andern werden (besonders bei CMOS-, aber auch bei TTL-Chips) die beiden in Serie liegenden Ausgangstransistoren (Gegentakt-Endstufe) gleichzeitig leitend, was eine kurzzeitig (bei CMOS sehr) hohe Stromaufnahme zur Folge hat. Daraus ist ersichtlich, daß der mittlere Leistungsbedarf mit der Schalthäufigkeit zunimmt.

Bei einem LS-Chip mit 15 pF kapazitiver Last am Ausgang beträgt die Leistungsaufnahme zum Beispiel erst bei etwa 10 MHz das Doppelte gegenüber dem Wert bei 100 kHz und steigt dann weiter recht kräftig (halbwegs linear) an. Bei 50 pF jedoch verdoppelt sich die Leistungsaufnahme bereits bei 4 MHz.

Es zeigt sich also, daß auch die Leistungsaufnahme von TTL-Chips recht stark von der kapazitiven Belastung abhängt. Das ist deshalb wichtig, weil diese Abhängigkeit vielfach ausschließlich CMOS-ICs angelastet wird, bei denen diese allerdings auch extrem deutlich wird.

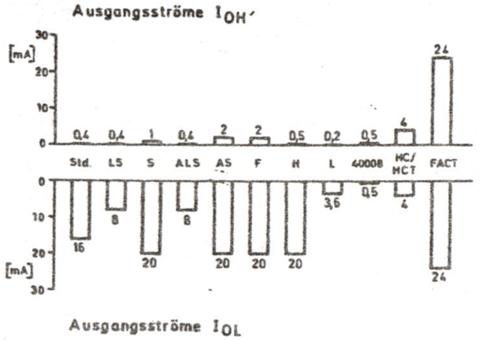
Das liegt im wesentlichen daran, daß CMOS-ICs so gut wie keine statische Leistungsaufnahme aufweisen (solange sie nicht an einem ohmschen Verbraucher arbeiten müssen, versteht sich). Wegen ihrer fantastisch geringen Leistungsaufnahme im statischen Betrieb wirken sich hier natürlich schon frühzeitig Schaltverluste aus, die linear mit der Frequenz steigen. Je nach Beschaltung (hauptsächlich kapazitive Last) gibt es allerdings eine Frequenz, ab der zum Beispiel LS- und HCT-Gatter den gleichen Leistungsbedarf haben.

Treiberleistung

Wieviel Strom ein Vertreter einer TTL- oder CMOS-Baureihe liefern kann, wird oft in Form von Lastfaktoren (Fan-in und Fan-out) angegeben. Die Berechnung der Lastfaktoren wird jeweils getrennt für den High- und für den Low-Pegel-Ausgangsstrom durchgeführt. Ohne weitere Angaben besagt ein Fan-out von 10, daß an einen IC-Ausgang 10 Eingänge von Vertretern der eigenen Gattung mit einem Fan-in von 1 angeschlossen werden können.

Da beide Lastfaktoren für jede IC-Familie andere absolute Werte haben, gibt man bei Vergleichen zwischen IC-Familien normalerweise die Zahl der LS-TTL-Eingangslasten an, die getrieben werden können. Weil TTL-ICs sowohl deutlich unterschiedliche Treiberströme gegen High- und Low-Pegel treiben als auch (darauf abgestimmt) unterschiedliche Eingangslasten darstellen, muß bei einem Vergleich mit CMOS-ICs (symmetrische Ausgangsströme, Eingänge rein kapazitiv) der Fan-out für H- und L-Pegel gesondert angegeben werden. Wegen dieser vielen Umrechnereien werden seit neuestem beispielsweise im Texas Instruments Pocket Guide nur noch die absoluten Ströme angegeben.

Interessant ist, daß inzwischen CMOS-Bausteine (wie die aus der FACT-Reihe) höhere Treiberströme liefern können als Mitglieder der 74-AS-Familie. Typischerweise können TTL-ICs innerhalb ihrer Baureihe 10 bis 60 Eingänge treiben, ein AS-Chip kann auch durchaus 100 ALS-Eingänge verkraften. Ganz andere Dimensionen eröffnen sich aber bei HCT, wo ein Ausgang bis zu 4000 HCT-Eingänge treiben kann – allerdings stark abhängig von der Frequenz, denn diese vielen Eingänge stellen natürlich eine üppige Kapazität dar.



In puncto Treiberleistung (Ausgangsströme) hinkten CMOS-ICs lange hinter ihren TTL-Kollegen her. Seit den FACT-Chips hat sich da einiges geändert.

Statischer Störabstand

Die Störfestigkeit einer mit Logik-ICs aufgebauten Schaltung ist nicht nur vom Schaltungslayout (gutes Netzteil, wenig Übersprechen zwischen Leiterbahnen und ähnliches) abhängig, sondern auch von den ICs selbst. Je größer der Pegelbereich, der ausgangsseitig zur Verfügung steht und durchfahren werden kann, ehe die Eingangsschaltung reagiert, desto besser.

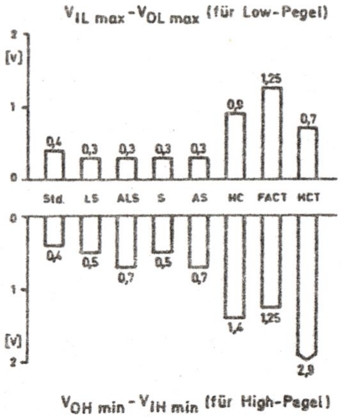
Ein Maß für die Störfestigkeit ist der statische Störabstand, der noch einmal in einen garantierten (worst-case) und einen typischen unterteilt wird. Der Worst-case-Störabstand gibt an, wie groß die maximale Amplitude einer dem Signal überlagerten Störspannung sein darf. Er wird aus den absoluten Grenzwerten ermittelt:

– Für High-Pegel wird die Differenz aus der niedrigstmöglichen High-Spannung am Ausgang des einen und der minimalen High-Spannung zum Umschalten auf H-Pegel am Eingang des anderen ICs bestimmt.

– Für Low-Pegel bildet man die Differenz aus der größtmöglichen Low-Spannung am Ausgang des einen IC und der maximalen Eingangsspannung, ab der das andere auf Low-Pegel schaltet.

Für LS-TTL liegen gemäß [2] die Werte bei nur 0,3–0,5 V, was bedeutet, daß eine Störspannung in dieser geringen Höhe zu fehlerhaften Pegelwechseln führen kann. HC-Chips brillieren hier mit Werten von mindestens 0,7 V bei Low-Pegel und mindestens 1,4 V bei High-Pegel.

Auch die FACT-Technologie schneidet hier gut ab. Aber da bei dieser neuen CMOS-Technologie auch die Umschaltimpulse – und damit die Störspannungen – viel höher als bei den TTL-Baureihen sind, fällt der positive Effekt hier nicht so stark ins Gewicht.



Auch der Störabstand ist eine wichtige Kenngröße von Logik-ICs. Hier wiesen die CMOS-ICs schon seit jeher bessere Werte als die TTL-Chips auf.

Normalerweise entwirft man aber Schaltungen gemäß der typischen Störspannungsabstände, zu deren Berechnung man wie oben gezeigt vorgeht, jedoch die typischen Ein- und Ausgangspegel heranzieht. HCT schneidet auch hier (H: 3,4 V, L: 1,3 V) deutlich besser ab als etwa LS (H: 2,3 V, L: 0,85 V).

Dynamischer Störabstand

Der dynamische Störabstand gibt Auskunft darüber, ab wann Impuls-Amplituden, die länger als die Verzögerungszeiten andauern, den statischen Störabstand überschreiten und damit fehlerhafte Zustandsumschaltungen bewirken. Dynamische Störungen in einem TTL- oder CMOS-System sind zum Beispiel kurzzeitige Spannungsschwankungen des Netzteils, Übersprechen beim Umschalten von ICs oder Reflexionsspannungen. Bei einer Störimpuls-Amplitudenlänge von 15 ns kann die 74-FACT-00 einen Störimpuls von 3 V auf der Versorgungsleitung verkraften, ein 74-LS-00 hingegen nur etwa 2,5 V.

Umgang mit CMOS

CMOS-Schaltkreise sind im Vergleich zu bipolaren Logik-ICs weitaus empfindlicher gegen elektrostatische Spannungen. Aufgrund des hohen Eingangswiderstandes kann sich die Gate-Kanal-Kapazität durch statische Aufladung auf Spannungen von einigen tausend Volt aufladen. Bereits ab rund 100 V muß man damit rechnen, daß die Gate-Kanal-Isolierung durchschlägt.

Da solche Spannungen in der Praxis recht oft auftreten, beispielsweise durch Schlurfen über Teppichboden, sorgen bei CMOS-Chips Ableitdioden an allen Anschlüssen für sicheren Schutz bis etwa 2000 V. So ist es sicher empfehlenswert, die ICs zum Beispiel in Leitgummi-Matten zu lagern, erst recht, wenn man sie in Kunststoffschubladen aufbewahrt.

Wesentlich übler als beispielsweise das vielzitierte 'Begrapschen mit ungeerdeten Fingern' nehmen CMOS-Chips aber die Ansteuerung ohne Betriebsspannung und negative Spitzen der Betriebsspannung (manche Netzteile produzieren solche 'Unterschwinger' beim Abschalten). Beides erklärt sich dadurch, daß die Isolation der n-Kanal- von den p-Kanal-FETs auf dem Chip erst durch Anlegen der Betriebsspannung als p-n-Übergang realisiert wird; auch sind die Schutzdioden bei solchen Betriebsarten gefährdet.

HEFT 19 Juni 1987

HCT kontra LS in der Praxis

Von Kompatibilität und Stromverbrauch

Detlef Grell

'Ab 'ner bestimmten Frequenz saugen die HCTs genausoviel wie LS-Chips, und der Ersatz von Bustreibern, die ohmsche Lasten treiben, ist sowieso witzlos. Außerdem sind die HCTs noch viel zu teuer, schwer zu kriegen, und laufen tun sie auch nicht immer.' Solcherlei Äußerung kann man nicht als völlig falsch abtun, aber mittlerweile sind die HCT-Chips zumindest im Versandhandel ohne Probleme erhältlich und kosten 'nur noch' das Doppelte von LS-Chips. Und so kann man ganz konkret nachprüfen, was es mit den restlichen Behauptungen auf sich hat.

Für die vorliegende 'Untersuchung' wurde der c't86 herangezogen, der in einer weitgehend CMOSisierten und in einer bipolaren Version zur Verfügung stand. Bei der Umrüstung auf CMOS wurden aber nicht nur LS-Chips (soweit erhältlich) durch HCTs ersetzt, sondern darüber hinaus alle komplexeren Bausteine (8255, 6845, 8086 und so weiter), derer ich habhaft werden konnte. Zu Tabelle 1 an dieser Stelle nur so viel, daß man sie nicht ohne genaue Aufschlüsselung der gewechselten und verbliebenen ICs bewerten kann.

HC-/HCT – alles kompatibel?

Nein. Zunächst die drei wichtigsten Faktoren, die einen Ersatz von LS- durch HCT-Chips betreffen:

1. Die HC-Reihe ist eingangsseitig vollkommen CMOS-konform. Das bedeutet, ein sicheres Umschalten von HC-Gattern nach low erfolgt unterhalb von 1,0 V Eingangsspannung, ein sicheres Umschalten nach high erst oberhalb 3,5 V. TTL-Ausgänge, die auf HC-Eingänge arbeiten, unterschreiten mit 0,4 V zwar in jedem Fall die Low-Schaltschwelle, aber mit im ungünstigsten Fall 2,4 V High-Pegel unter Last nicht die High-Schaltschwelle (siehe Bild).

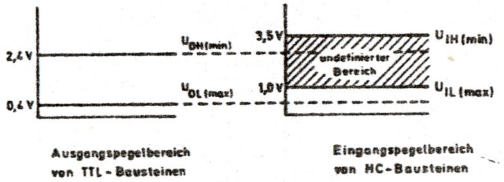
Abhilfe kann man schaffen, indem man dem TTL-Ausgang mit einem sorgsam berechneten Pull-up-Widerstand etwas hochhilft oder aber die extra dafür entwickelten HCT-Chips mit TTL-konformen Schaltpegeln am Eingang einsetzt. Der umgekehrte Fall, also wenn HC-Ausgänge auf TTLs arbeiten, ist dagegen wegen des großen Ausgangshubs von CMOS-Chips unkritisch.

2. Unbenutzte Eingänge sollen auf definierten Pegel gelegt werden; denn sowohl bei HC als auch bei HCT kann nicht ausgeschlossen werden, daß diese extrem hochohmigen Eingänge unbeschaltet 'floaten' und dabei undefinierte Zustände annehmen (auch Oszillation möglich).

3. Die Anzahl von LS-Eingängen, die ein HC-/HCT-IC treiben kann, liegt mit 10 um Faktor zwei geringer als bei LS-TTL mit 20. Bei Hochstromtreibern (Bustreiber wie 244) ist der Unterschied sogar Faktor vier: die entsprechenden LS-TTLs können 60 einfache LS-Lasten treiben, die HC-Versionen nur 15. Konkret kann ein LS244 einen High-Strom von 15 mA und einen Low-Strom von 24 mA treiben, ein HC(T)244 in beiden Fällen nur 6 mA.

Andere Inkompatibilitäten, vermutlich aufgrund abweichender Schaltpegel und anderer innerer Verstärkung, zeigen sich in der Praxis zum Beispiel daran, daß die meisten einfachen Quarz-Oszillator-Schaltungen für LS mit HCT nicht mehr laufen.

Bei nachträglicher Umrüstung einer Schaltung, die vollständig für LS optimiert ist, können aber auch zu wenige Blockkondensatoren oder falsches Leiterplatten-Layout zu Problemen führen; denn die CMOS-Chips verursachen zum Beispiel höhere und kürzere Stromspitzen beim Umschalten.



Anders als bei den speziell darauf eingerichteten HCT-Chips kann es mit HC-Typen Probleme geben, wenn ein stark belasteter LS-Ausgang auf einen HC-Eingang arbeitet.

Komplexe CMOS-Chips

Neuerdings sind auch viele komplexe ICs wie Prozessoren (8086/88, Z80) und Peripherie-Bausteine (82xx, Z80-Familie), die sich als NMOS-Versionen etablierten, in CMOS zu haben (sogar die kritische, in NMOS nicht erhältliche Z80-DMA für 6 MHz Takt soll verfügbar sein!).

Die Probleme beim Ersatz dieser ICs gegen die NMOS-Vorgänger sind ähnlich wie bei LS und HC-/HCT. Auf NMOS optimiertes Schaltungs-Layout und offene Eingänge (hier verhält sich NMOS wegen interner Pull-up-Widerstände oft wie TTL und zieht definiert gegen high, wenn auch längst nicht immer) dürften die Hauptschuldigen sein, wenn ein Austausch nicht klappt.

Oft genug sind die (neueren!) CMOS-Chips aber besonders wohlmeinend gleich für die doppelte Taktfrequenz gegenüber ihren NMOS-Vorbildern ausgelegt worden, oder sie weisen aus anderen Gründen ein 'verbessertes' – aber in jedem Fall leider anderes – Timing auf.

Und dann gibt es noch ein grundsätzliches Problem, das man auch bei NMOS-ICs beobachten kann, die von Zweitherstellern stammen: die Chips sind nicht immer identisch. So sind uns etwa Z80-CPUs bekannt, die das I-Register nach einem Reset unterschiedlich vorbesetzen, und 8255-Bausteine mit abweichendem Timing für das Chip-select-Signal. Die Übertragung von ICs in NMOS nach CMOS bedeutet eine weitgehende Neuentwicklung (völlig andere Herstellungsmasken), und man darf sich keinesfalls darauf verlassen, daß etwa undokumentierte Befehle (wie beim Z80) oder ähnliches übernommen wurden.

Verwunderliches

Die Aufzählung der Inkompatibilitäten ist vermutlich nicht vollständig, denn einigen wahrhaft mysteriösen Effekten ist auch bei näherer Betrachtung nicht mit Logik (wahrscheinlich jedoch mit Logik-Analysator) beizukommen. Hier ein paar typische Rätsel, mit denen man konfrontiert werden kann.

Zwei Z80-CPUs von NEC unterschiedlichen Herstellungsdatums verhielten sich wie folgt:

– Keine funktionierte in einer intelligenten Terminalkarte.

– Beide liefen auf der intelligenten Floppy-Karte (IFC-Karte, c't 8/85) sowohl im Floppy-Modus als auch in der Betriebsart als eigenständiger CP/M-Rechner.

– Beide liefen scheinbar korrekt in einem anderen CP/M-Rechner, aber nur einer war in der Lage, korrekt mit den Z80-Restart-Befehlen umzugehen – der andere schmierte ab.

Ein CMOS-6845 (CRT-Controller), der MB 89321A von Fujitsu, tut klaglos Dienst in besagtem Terminal, verlegt aber in der Farbgrafikkarte für den c't86 die unterste Zeile außerhalb des sichtbaren Bildfensters. Das aber nicht immer, und meistens ließ sich der Effekt durch Ein-/Ausschalten des Rechners (arme Festplatte) nach kurzer Zeit abstellen.

Na ja, und wer unsere Berichterstattung über V30/V20 mitverfolgt hat, wurde auch hier mit den merkwürdigsten, selten reproduzierbaren Phänomenen konfrontiert, die sich nie richtig befriedigend erklären ließen.

Zur Ehrenrettung aller genannten Chips sei aber gesagt, daß keine Schaltplan-Recherche und Suche nach offensichtlichen Inkompatibilitäten stattfand.

	Bipolar/ NMOS (5 MHz)	CMOS (5 MHz)	CMOS (7,5 MHz)
IFC	0,87	0,53	0,53
Unicard*)	0,69	0,17	0,17
Farbgrafik	1,37	0,93	0,93
CPU II**)	0,73/0,76	0,07/0,22	0,09/0,22
RAM***)	0,85	0,44	0,55
I/O II	0,42	0,02	0,02
Summe	4,93/4,96	2,16/2,31	2,29/2,42

Alle Angaben in Ampere

*) Unterschiedliche Terminierung
**) Werte ohne/mit Busterminierung
***) RAMs unterschiedlicher Hersteller auf den Karten

Tabelle 1. Wenn eine vollständige Umrüstung in CMOS, wie etwa bei der CPU- und I/O-Karte, möglich ist, kann die Stromaufnahme bis auf ein Zwanzigstel gesenkt werden.

Umrüstung

Die Praxis zeigte, daß über 90 Prozent der Chips ohne Probleme austauschbar waren. Da ich bei Problemkandidaten wiederum keine Ursachenforschung betrieben habe, müßte die Erfolgsquote mit Schaltplanstudium (offene Eingänge, Fan-out- und Timing-Prüfung) noch zu steigern sein.

Wenn im folgenden nicht alle ICs getauscht wurden, so auch deshalb, weil noch nicht alle verfügbaren Chips angeschafft waren und es für einige LS-Typen noch keine HCT-Varianten gibt. Die Anschaffung dynamischer CMOS-RAMs unterblieb schlicht aus Kostengründen, zumal die Datenblätter kaum mehr als 50 Prozent Stromersparnis verheißen. Und bei PALs und PROMs gibt es unseres Wissens leider noch keine adäquaten CMOS-Versionen. Da vor allem die PALs furchtbar viel Strom saugen, ist das sehr bedauerlich.

Meßbedingungen

Es wurde ein einfaches Digital-Meßinstrument im nicht übermäßig genauen 10-A-Meßbereich verwendet, von dessen glaubwürdiger Funktion wir uns mittels Drehspulmeßgerät überzeugt hatten.

Der Spannungsabfall am Meßwiderstand (etwa 0,2 Ohm) reduziert die Betriebsspannung am Prüfling. Da die nicht mit CMOS bestückten Karten mehr Strom verbrauchen als die anderen, werden erstere mit geringerer Betriebsspannung versorgt und dadurch etwas bevorzugt. Denn gerade bei TTL-ICs machen Schwankungen von 0,2 bis 0,3 V schon einige Milliampere bei der Stromaufnahme aus.

Alle Karten wurden im dynamischen Betrieb am Bus gemessen, also stets bei ordnungsgemäß funktionierendem Gesamtsystem.

IFC-Karte		**Unicard**	
ersetzt:	CPU und CTC (Z80A), 12 x LS	ersetzt:	8253, 17 x LS
gleich:	WD 2797, Z80A-DMA, 1 EPROM, 8 dyn. RAMs, 3 PALs, 1 x LS04, 1 x HC393, 2 x 7406	gleich:	Adreß-PROM, Uhrenchip (CMOS), 1 x 75452, 1 x LS 92
Farbgrafikkarte		**CPU-Karte**	
ersetzt:	6845, 29 x LS	ersetzt:	2 EPROMs, 8086, 8259A, 8288, 8284, 15 x LS
gleich:	2 stat. RAMs, 1 EPROM, 6 PALs, 1 x 75451, 1 x 75454, 1 x S04, 1 x LS76A	gleich:	–
RAM-Karte		**I/O-Karte**	
ersetzt:	15 x LS	ersetzt:	8253/4, 8255, 8251A, 12 x LS
gleich:	32 dyn. RAMs, Adreß-PROM, 1 x S00	gleich:	–
			Die Karten waren nicht vollständig, aber untereinander gleich bestückt.

Tabelle 2. Nicht alle NMOS- und Bipolar-ICs konnten durch CMOS-Chips ersetzt werden. Hier die genaue Aufstellung zur Interpretation von Tabelle 1.

Interpretationen

Zur Auswertung der Ergebnisse in Tabelle 1 muß stets berücksichtigt werden, wie viele und welche ICs getauscht wurden (Tabelle 2). In den Tabellen ließen sich nicht alle Sachverhalte unterbringen.

IFC-Karte

Obwohl nur zwei der komplexen ICs und zwölf LS-Chips getauscht wurden, ließ sich die Stromaufnahme fast auf die Hälfte reduzieren. Daß hier bei CMOS-Betrieb kein Unterschied zwischen 5 und 7,5 MHz zu verzeichnen ist, liegt daran, daß diese Karte nur als Port vom Host angesprochen wird und intern mit eigenem Takt (4 MHz) arbeitet.

CMOS-DMA und -EPROM konnten nicht rechtzeitig beschafft werden, hier bietet sich noch die Chance, den Stromverbrauch weiter zu senken. 7406 und WD2797 gibt es nicht in CMOS, das LS04 (Quarzoszillator) wurde sicherheitshalber nicht getauscht. PALs und dynamische RAMs wurden generell nicht getauscht (siehe oben).

Unicard

Die Unicard stellt weitgehend I/O-Ports bereit (Tastatur, Timer, Uhr und so weiter). Die angegebenen Werte sind allerdings nicht unter 'regulären' Bedingungen zustandegekommen, denn nur die CMOS-Version ist mit Terminierungswiderständen bestückt, nur bei ihr läuft also das Netzteil für die

Terminierung unter Last. Obwohl man also bei der bipolar bestückten Karte noch ein paar Milliampere drauflegen müßte, verbraucht die CMOS-Version deutlich weniger Strom als die bipolare.

Für den LS92 gibt es noch keinen HC-Ersatz, ebensowenig (unseres Wissens) wie für 75xx-Typen.

Farbgrafikkarte

Hier gab es (außer EPROM) nur einen komplexen Chip zum Austauschen, die statischen RAMs waren ohnehin schon CMOS-Low-power-Versionen. Zwar funktionierte der 6845 (jedenfalls in dieser Karte) nur unzureichend, aber sein Einsatz allein schlägt schon mit 0,1 A zu Buche. Abgesehen davon, daß die sechs PALs eine beträchtliche 'Grundlast' bilden, zeigt sich vor allem, daß der Ersatz von 29 LS-ICs nur müde 0,2 A Einsparung bringt.

Auf dieser Karte geht es nämlich recht lebhaft zu (immerhin werden hier 60mal in der Sekunde 128.000 Pixels über den Schirm gehetzt), und es wurde ja bereits erklärt, daß CMOS-ICs bei hoher dynamischer Beanspruchung kaum Vorteile bieten. Da die Karte hauptsächlich mit sich selbst beschäftigt ist, fallen unterschiedlich schnelle Zugriffe der CPU (5 und 7,5 MHz) gar nicht ins Gewicht.

Der 74LS76A wird nicht in HC-MOS gefertigt. Der Quarzoszillator lief weder mit dem eigentlich vorgesehenem LS04 noch mit HC-Typen sonderlich gut, sondern am besten mit einem S04.

CPU-Karte

Die CPU-Karte konnte vollständig in CMOS realisiert werden. Dabei wurden für alle komplexen Chips (8086, 8288, 8284 und 8259A) die Versionen von NEC (V30, 71088, 71011 und 71059) verwendet. Bei dieser reinen CMOS-Version sieht man neben der faMOS geringen Stromaufnahme endlich auch mal deren Frequenzabhängigkeit, wenngleich der Stromanstieg geringer ausfällt als die Erhöhung der Frequenz.

Die Tabellenwerte hinter dem Schrägstrich ergeben sich, wenn die aktive Busterminierung im System ist. Diese wirkt sich bei anderen Karten kaum meßbar aus, denn die CPU-Karte hat mit Abstand die meisten Ausgänge am Bus. Daß die CMOS-Karte dadurch stärker belastet wird, dürfte an dem anderen 'natürlichen' Ruhepotential von CMOS-Ausgängen liegen.

Hochinteressant aber dennoch die Tatsache, daß die dynamische Beanspruchung der meisten ICs auf dieser Karte hier sehr gering ist, so daß ein beträchtlicher Einspareffekt zu registrieren ist; denn eigentlich sollte man ja annehmen, daß gerade im Umfeld der CPU, der zentralen Schaltstelle, 'das Leben pulst'.

RAM-Karte

Die Werte für diese Karte sind nicht direkt vergleichbar, da die 32 dynamischen RAM-Chips auf beiden Karten von unterschiedlichen Herstellern stammen und dadurch Verfälschungen von 20 Prozent durchaus denkbar sind. Andererseits paßt die Stromersparnis durch den Tausch von 15 LS-Chips ganz gut ins Gesamtbild. Für das S-IC ließ sich kein funktionsfähiger HCT-Ersatz finden, ein FACT-Chip hätte möglicherweise Chancen.

I/O-Karte

Auch diese Karte konnte (in der verwendeten Teilbestückung) völlig in CMOS realisiert werden. Hier habe ich zunächst mal Meß- und andere Fehler gesucht und schließlich den Drucker angeschlossen, um zu sehen, ob die Karte überhaupt funktionierte. Und wie sie funktionierte! Auch der Anschluß des Druckers erhöhte den Strombedarf nur um wenige Milliampere. Die Meßergebnisse dieser Karte demonstrieren in bester Weise, wie langsam Port-I/O stattfindet und wie massiv hier die minimale statische Leistungsaufnahme von CMOS-Chips zur Geltung kommt.

Die komplexeren Chips (8255, 8254, 8251) wurden wiederum durch die 710xx-Versionen von NEC ersetzt. Beide Karten waren zwar nicht maximal, aber gleichwertig bestückt. So fehlten beiden die V.24-Empfänger und je ein Satz 8254/51.

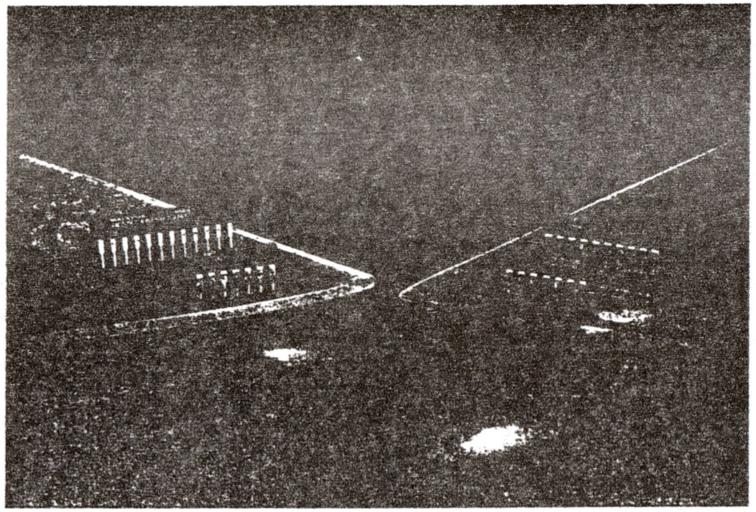
Resümee

Der in CMOS aufgebaute c't86 benötigt in ungünstigster Konstellation, also bei 1,5facher Taktfrequenz und mit Busterminierung, weniger als die Hälfte Strom gegenüber der günstigsten bipolaren Ausführung – und das, obwohl noch nicht alle möglichen ICs (zum Beispiel 40 dynamische RAMs) getauscht wurden.

Geht man mal davon aus, daß sich durch CMOS-PALs die Leistungsaufnahme der Farbgrafikkarte nur auf die Hälfte der bipolar bestückten Karte senken ließe, auf der anderen Seite bei den I/O-orientierten Systemeinheiten aber Einsparungen bis zum 20fachen möglich sind, sollte man ein reines CMOS-System auf mindestens ein Drittel, wenn nicht gar ein Viertel der Stromaufnahme einer bipolaren Version bringen können.

Der geringere Strombedarf zieht mehrere Vorteile nach sich. Die Wärmeentwicklung im Systemgehäuse sinkt deutlich, und zwar sowohl auf den Systemplatinen als auch am Netzteil. Denn jedes Watt Nutzleistung zieht – je nach Wirkungsgrad des Netzteils – 0,2 bis 1 Watt Verlustleistung (sprich Wärme) nach sich. So könnte man sich entweder an einer erheblich gesteigerten Lebensdauer kühlerer Bauteile ergötzen oder auf einen Lüfter verzichten.

ct



CMOS kontra TTL

Die Vertreter der bedeutendsten Logik-Familien im Vergleich

Jörg Koch
Frank Kremser

Die groß herausgestellten Erfolge bei RAMs, wo schon 4-MBit-Chips als Labor-Muster existieren, und bei Mikroprozessoren, die es – wie etwa der V70 von NEC – schon auf 375 000 Transistorfunktionen bringen, überstrahlen den Fortschritt bei den 'Arbeitstieren' der Digital-Elektronik, den einfachen Logik-ICs. Die lange Zeit wegen ihrer Langsamkeit in der Mikro-Computerei 'unbrauchbare' CMOS-Technik hält nun auch in diesem Bereich Einzug und ist auf dem besten Weg, die 'alten TTL-Heizungen' zu verdrängen.

Kein anderer Bereich wird so geprägt von spektakulären Erfindungen und neuen Technologien wie die Entwicklung von Halbleiter-Chips. Der eine Trend ist klar: Durch eine permanent gesteigerte Anzahl von Bauelement-Funktionen bei möglichst verringerter Chip-Fläche lassen sich immer komplexere und leistungsfähigere Chips herstellen.

Der andere Trend geht zu immer noch schnelleren ICs, und oft genug gehen beide Richtungen Hand in Hand. Denn bei beiden Zielsetzungen hat man mit dem gleichen Problem zu kämpfen, das sich auf eine sehr simple Formel bringen läßt: Je mehr Bauteil-Funktionen auf gleichem Raum und je größer die Geschwindigkeit, desto wärmer wird's auf dem Chip.

Im folgenden wird auf den aktuellen Stand der Technik eingegangen, und es werden die charakteristischen Merkmale der meistverbreiteten IC-Gattungen beleuchtet. Um der Frage nachzuspüren, warum CMOS-Chips lange Zeit so langsam waren, bedarf es einer kleinen IC-Historie.

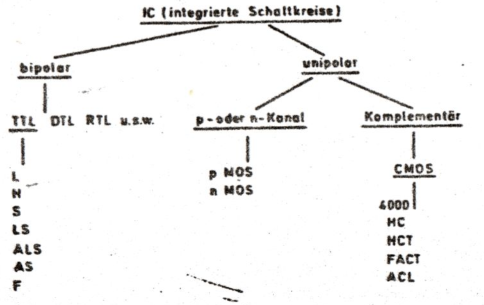
'Klassifizierung'

Die heute am meisten verbreiteten Logik-Familien, die im Mittelpunkt dieses Beitrags stehen, werden in zwei große Klassen unterteilt. Dabei orientiert man sich an den verwendeten Halbleiterbauelementen, die in bipolare und unipolare unterschieden werden (siehe auch dritten Kasten).

Mittlerweile können die IC-Produzenten allein bei den bipolaren Logik-ICs auf eine reichhaltige Palette zurückblicken. Wir beschränken uns hier aber auf einen kleinen Ausschnitt. Wir lassen beispielsweise durchaus bedeutende Vertreter wie etwa die sehr schnellen (und energieverschwenderischen) ECL-Typen ebenso außer acht wie DTL- oder RTL-Chips, wobei letztere heute ohnehin keine praktische Bedeutung mehr haben.

Zur Herstellung unipolarer Schaltkreise wird fast ausschließlich eine besondere Art von Feldeffekt-Transistoren (FETs), die MOS-FETs, benutzt. Die Abkürzung MOS steht für Metall-Oxid-Semiconductor und soll zum Ausdruck bringen, daß bei diesen FETs die Steuerelektrode (Gate) vom leitenden Kanal durch eine Si-Oxid-Schicht extrem gut isoliert ist (sehr hoher Eingangswiderstand).

Bei den MOS-Schaltungen wird noch eine weitere Unterteilung in P- beziehungsweise NMOS (nur p- oder nur n-Kanal-FETs) und CMOS (p- und n-Kanal-FETs in komplementärer Schal-



Die Hauptunterscheidung bei integrierten Schaltkreisen nimmt man anhand der eingesetzten Halbleiter-Bauelemente in 'bipolar' und 'unipolar' vor.

tungstechnik) vorgenommen. Viele Mikroprozessoren (zum Beispiel 8080, 8086, 68000) werden in NMOS-Technologie gefertigt. Die bedeutendsten Vertreter der CMOS-Technologie waren bis vor kurzem die Logik-ICs der 4000er Reihe.

TTL-Chronik

Im Laufe der Zeit sind insgesamt acht verschiedene TTL-Baureihen entstanden. Hauptsächlich unterscheiden sie sich durch ihre Geschwindigkeit und Leistungsaufnahme. Die Baureihen sind alle untereinander anschluß- und funktionskompatibel, was bedeutet, daß sie (weitgehend) gegeneinander ausgetauscht werden können.

Standard-TTL

Die Produktion der 74-Standard-TTL-Reihe begann 1963, und diese IC-Urahnen werden (wenn auch verbessert) heute noch hergestellt und eingesetzt. Diese Logik-Familie war lange Zeit die preiswerteste, ist aber heute weitgehend von der moderneren LS-TTL-Serie (siehe unten) verdrängt worden.

Low-power und High-speed

Vier Jahre nach der Standard-Serie kamen die 74-H-Typen (High-speed) auf den Markt. Sie waren zwar schneller als die Vorgänger, konnten sich wegen ihrer großen Leistungsaufnahme jedoch nicht auf breiter Ebene durchsetzen. Gleichzeitig begann die Serienproduktion der 74-L-TTLs (Low-power). Die Leistungsaufnahme dieser Serie war zwar gering, aber dafür waren sie im Vergleich zur Standard-Serie fünf- bis zehnmal langsamer.

Bei der L- und H-Serie hat man eigentlich keine technischen Neuerungen eingebracht, sondern nur die Schaltungsauslegung (schneller und größere Leistungsaufnahme beziehungsweise umgekehrt) von Standard-TTL variiert. Beide Serien sind inzwischen technologisch veraltet und werden nicht mehr produziert.

Schottky und Low-power-Schottky

Die Möglichkeit, Schottky-Dioden in TTL-Chips zu integrieren, bedeutete einen gewaltigen Fortschritt in der TTL-Technologie. Die Schottky-Diode besitzt (durch Halbleiter-Metall- statt p-n-Übergang) eine geringere Schwellenspannung als einfache Silizium-Dioden. Schaltet man sie parallel zur Basis-Kollektor-Diode herkömmlicher Transistoren, so geraten diese nicht mehr vollständig in die Sättigung. Die so entstandenen Schottky-Transistoren sind ohne größere Leistungsaufnahme deutlich schneller.

Die ersten Schottky-TTLs (74-S-xx) wurden ab 1969 produziert. Vertreter dieser Gattung sind etwa doppelt so schnell und haben eine etwas geringere Leistungsaufnahme als die der 74-H-Serie.

Seit 1971 wird die mittlerweile am meisten eingesetzte 74-LS-Familie (Low-power-Schottky) produziert, die mehrere hundert Mitglieder umfaßt. Bei diesen setzte man den Geschwindigkeitsvorteil allerdings durch hochohmigere Schaltungsauslegung in einen deutlich verringerten Leistungsbedarf um. LS-Chips sind daher genauso schnell wie die ICs der Standard-Reihe, nehmen aber nur etwa ein Fünftel an Leistung auf.

Advanced Schottky und Advanced Low-power-Schottky

1980 gesellte sich dann die 74-ALS-Reihe (Advanced Low-power-Schottky) und 1982 die 74-AS-Familie (Advanced Schottky) zu den bisherigen hinzu. Diese repräsentieren im wesentlichen Fortschritte (advanced heißt fortgeschritten) durch Verfeinerungen der S- und LS-Typen.

Schaltungstechnisch hat man beispielsweise die Eingangswiderstände um den Faktor vier erhöhen können und durch einen Schaltungstrick (einen sogenannten 'Miller-Killer') die Einflüsse der Sperrschichtkapazität drastisch gesenkt.

Technologisch gelangen weitere Kapazitätssenkungen, indem die Transistoren auf dem Chip statt mittels p-n-Übergängen durch Oxid-Schichten isoliert und die Chip-Strukturen insgesamt etwa um Faktor zwei gegenüber LS-TTL verkleinert wurden.

AS-/ALS-Chips sind so ausgelegt, daß sie etwa doppelt so schnell sind wie die jeweils zugrundeliegenden S-/LS-Typen und dennoch nur halb soviel Leistungsaufnahme aufweisen. Auch bieten die A-Typen eine größere statische und dynamische Störsicherheit, höhere zulässige Impulsfrequenz, geringere Temperaturabhängigkeit und einen größeren Eingangs- und Betriebsspannungsbereich.

FAST-TTL

Nicht vergessen darf man die 74-F-Reihe, die seit 1979 von Fairchild produziert wird. Das 'F' steht für FAST, was natürlich 'schnell' assoziiert, aber laut Hersteller 'Fairchild Advanced Schottky TTL' bedeutet. Ein typisches F-Gatter ist doppelt so schnell wie ein ALS-Gatter und damit nur geringfügig langsamer als ein AS-Chip. Der Leistungsbedarf liegt etwa beim Vierfachen von ALS und etwa bei der Hälfte von AS.

Obwohl diese Chips ein günstigeres Verhältnis von Leistungsbedarf zu Geschwindigkeit aufweisen als die AS-/ALS-Konkurrenz und auch eine große Typenvielfalt verfügbar ist, konnten sie bislang nicht die Bedeutung letzterer erringen.

Langsamer Start

Bei den integrierten Schaltungen mit unipolaren Halbleitern hat man vergleichsweise spät damit begonnen, auch die 'kleinen' Logik-ICs (also mit geringem Integrationsgrad) zu fertigen. Von speziell für Mikroprozessor-Familien maßgeschneiderten Puffern oder Multiplexern abgesehen gibt es keine Logik-Chip-Familie in P- oder NMOS-Technik, sondern der Einstieg erfolgte gleich in CMOS.

4000er Reihe

Ende der 60er Jahre kam die erste CMOS-Reihe (CD 4000A) ohne Ausgangspuffer auf den Markt. Diese Reihe wurde 1976 von Typen der Familien CD beziehungsweise HEF 4000B/4500B mit gepufferten Ausgängen abgelöst und ist bis heute unverändert im Handel erhältlich.

Diese ICs begründeten sozusagen den schlechten Ruf der CMOS-Technologie, nämlich

Der 'Leidensweg' der CMOS-Chips

Detlef Grell

Als man 1959 bei Texas Instruments mit der Entwicklung integrierter Schaltungen begann, war der Feldeffekt-Transistor erst zwei Jahre alt, seine Technologie steckte also noch völlig in den Kinderschuhen. Kein Wunder also, daß der IC-Start mit dem 1947 erfundenen bipolaren Transistor stattfand, zumal die nachher tatsächlich zur Integration verwendeten MOS-FETs sogar erst 1962 in Erscheinung traten.

Zu den bipolaren integrierten Schaltungen gab es somit jahrelang keine MOS-Konkurrenz, so daß die TTL-Logik den Markt völlig beherrschte. Auch besaßen deren Hersteller bald einen Know-how-Vorsprung, der sich in der Produktion von ICs mit Leistungsdaten niederschlug, mit denen es die ersten MOS-ICs nicht aufnehmen konnten. Der Makel, daß bipolare Transistoren etwa die 15fache Chip-Fläche von MOS-Transistoren benötigen, kam erst ziemlich spät zum Tragen.

Denn in den Anfängen der Integration begnügte man sich (lange Zeit fertigungstechnisch bedingt) mit weniger als 1000 Transistoren pro Chip. Die damals entstandenen Digital-ICs enthielten vorwiegend Inverter, Gatter, Flipflops und ähnliches, waren also allesamt ICs mit geringem Bedarf an Chip-Fläche. Man konnte die Schaltungen auch recht niederohmig und damit schnell auslegen, da bei den wenigen Bauelementen die Wärmeabführung relativ unkompliziert war. Diese ICs gibt es heute noch, wenn auch inzwischen um ein Vielfaches verbessert, in Gestalt von TTL-Chips.

Der Erfolg der 'langsamen' MOS-ICs begann mit dem Wunsch nach höheren Integrationsdichten durch das Aufkommen von Mikroprozessoren und Speicher-Chips. Der geringere Bedarf an Chip-Fläche bei MOS beruht unter anderem darauf, daß bei Reihenschaltung von FETs gleicher Kanal-Polarität der Drain des ersten als Source des folgenden verwendet werden kann. MOS bietet aber noch mehr Vorteile:

- Die höhere Integrationsdichte bedeutet eine kleinere Chip-Fläche und damit mehr ICs pro Wafer (das sind diese Silizium-Scheiben, auf denen man die ICs 'züchtet'). Das führt zu einer besseren Ausbeute und dadurch zu geringeren Kosten.
- MOS-FETs sind sehr hochohmig. Dadurch kann die gesamte Schaltungsauslegung ebenfalls hochohmiger ausfallen, wodurch die Leistungsaufnahme (Wärmeentwicklung) sinkt. Letztlich ist auch die Begrenzung der Wärmeentwicklung eine wichtige Voraussetzung für eine hohe Integrationsdichte.
- Der Herstellungsprozeß war zumindest in den Anfängen einfacher und damit billiger als bei bipolaren Schaltungen.

Erkaufen mußte man sich diese Vorteile allerdings auch mit Nachteilen:

- Die hochohmige Schaltungsauslegung hat geringere Ausgangsströme (Treiberleistung), verglichen mit bipolaren ICs, zur Folge.
- Die kleinen Ausgangsströme begrenzen auch chipintern ganz erheblich die Schaltgeschwindigkeit, weil parasitäre (ungewollte, aber unvermeidliche und bei CMOS recht hohe) Kapazitäten nur sehr langsam umgeladen werden können.
- Bei einigen MOS-Technologien sind wegen abweichender Betriebsspannungen Konverter für Schaltschwellen und Schaltpegel erforderlich, um etwa die Zusammenarbeit mit TTL-ICs zu ermöglichen.

Warum machte man die herstellungstechnisch inzwischen günstigeren MOS-ICs nicht niederohmiger und damit schneller? Und warum machte man bipolare ICs nicht hochohmiger und der Höchstintegration zugänglich?

Nun, man machte beides, aber hier gibt es prinzipbedingte Barrieren. Bipolare Transistoren brauchen zu ihrer Funktion einen Steuerstrom, FETs kommen mit einer Spannung aus. MOS-FETs sind daher von 'Natur aus' an den Eingängen so hochohmig, wie es bipolare Transistoren nie sein können.

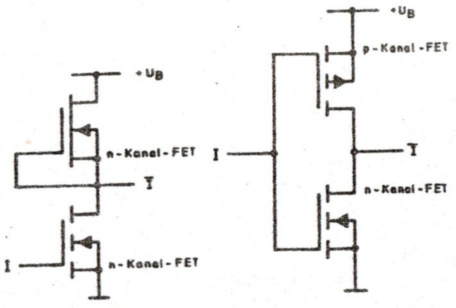
Ein hochohmiger Eingang bedeutet nun allerdings nicht zwangsläufig auch einen hochohmigen Ausgang (kleiner Ausgangsstrom), und so hätte die MOS-Technologie schnell alle Trümpfe in der Hand gehabt, wenn es frühzeitig gelungen wäre, die Kanal-Widerstände zu senken. Aber wenn man die Kanäle vergrößerte (anders konnte man zuerst nicht vorgehen), erhöhte sich proportional die Eingangskapazität, und die Schaltgeschwindigkeit blieb weitgehend konstant.

Eingangskapazitäten waren bei bipolaren Transistoren nie das Hauptproblem, schon sehr früh hatte man deren Auswirkungen auf kaum noch störende Größenordnungen vermindert. Man hatte vor allem Probleme, leitende Transistoren schnell wieder abzuschalten (langsame Ladungsträger-Rekombination in der Basis). Hier erzielte man Fortschritte durch immer dünnere Basis-Schichten und gezieltes Verunreinigen mit Gold-Atomen.

Bei den FETs jedoch kämpfte man verbissen um die Verringerung der Kapazitäten (durch Gestaltung von Gate und Kanal, vor allem aber auch durch Verkleinerung der Gesamtstruktur) und Senkung der Kanal-Widerstände durch immer raffiniertere Herstellungsverfahren.

Der 'Leidensweg' der MOS-Technik begann mit PMOS, weil man den Herstellungsprozeß am besten beherrschte. Die ersten PMOS-ICs benötigten allerdings ungünstig hohe (bis 29 V) und mehrere Betriebsspannungen. Nach diversen Verbesserungen der PMOS-Technologie bekam man endlich die NMOS-Technologie in den Griff. Die dreifach höhere Ladungsträgerbeweglichkeit von Elektronen in n-Kanal-FETs gegenüber den Löchern in PMOS-FETs machten diese FETs wesentlich niederohmiger und damit schneller.

Aber wie bei bipolaren Schaltungen fließt in N- und PMOS-Gattern (schaltungsbedingt) immer ein Betriebsstrom, der zu einer permanenten Leistungsaufnahme führt. Wenn man hingegen jede Schalterstufe sowohl aus einem n- als auch einem p-Kanal-FET wie bei einer komplementären Verstärker-Stufe kombiniert, fließt ein nennenswerter Strom nur noch in den Umschaltmomenten, weil dann beide Transistoren kurzzeitig leiten.



NMOS - Inverter   CMOS - Inverter

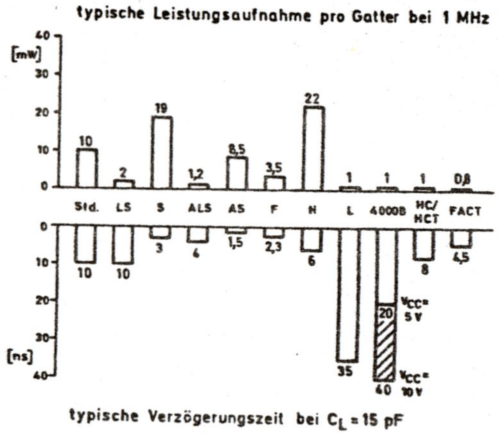
Bei einfachen MOS-Schaltungen fließt in jeder Stufe ein Betriebsstrom, während bei CMOS immer nur einer der komplementären Transistoren leitet, über die Verstärkerstufe selbst also kein Stromfluß erfolgt.

Die Leistungseinsparung bei CMOS (etwa 10 nW) gegenüber NMOS (etwa 0,1 mW pro Gatter) ist gewaltig, auch wenn dieser Wert nur für den statischen Betrieb gilt, also wenn keine Pegelwechsel erfolgen. Deshalb zieht man für Vergleiche oft das Geschwindigkeits-Leistungs-Produkt heran.

An diesem Wert kann man direkt ablesen, ob ein Chip nur auf Kosten höherer Leistungsaufnahme schneller gemacht wurde oder ob er bei Berücksichtigung beider Aspekte Vorbeziehungsweise Nachteile bietet. Texas Instruments gibt zum Beispiel für HC einen Wert von 1,2, für ALS 4,0 Picojoule bei 100 kHz an.

Man brauchte in CMOS nun allerdings doppelt so viele FETs wie bisher, und diese mußten technologiebedingt gut voneinander isoliert werden. Auch wurden sehr bald kräftigere Ausgangstreiber in die Schaltung aufgenommen. Führen alle drei Punkte zu einer Verringerung der Integrationsdichte, erhöhen die zusätzlichen Verstärker auch noch die Signaldurchlaufzeit.

Alles in allem war die Herstellung von ICs in CMOS um ein Mehrfaches aufwendiger als in der bis dato favorisierten NMOS-Technologie. Und waren die ersten CMOS-Chips der 4000er Reihe auch ziemlich 'lahme Krücken', so waren sie beispielsweise für Geräte mit Batteriebetrieb die einzig sinnvolle Lösung. Und die neuesten Vertreter der HCT- und (F)ACT-Serie zeigen deutlich, daß sich der 'Fortschrittskampf' gelohnt hat.



Die wichtigsten Kennwerte von gängigen CMOS- und TTL-Logik-ICs, Leistungsaufnahme und Schaltgeschwindigkeit, im Vergleich. Angegeben sind die Durchnittswerte eines NAND-Gatters.

furchtbar langsam zu sein. Mit einer typischen Verzögerungszeit von 50 Nanosekunden bei 5 V und etwa 25 bei einer TTL-unverträglichen Betriebsspannung von 10 V konnten sie nur durch ihren geringen Leistungsbedarf bei langsamen Anwendungen und ihre hohe Störsicherheit einige Marktbedeutung erringen. Nachteilig bei diesen ICs ist weiterhin, daß sie im Regelfall nur wenig Ausgangsstrom liefern und damit meist nur einen LS-Eingang treiben können.

74-C-Reihe

Zusammen mit den gepufferten 4000ern wurde die 74-C-Familie eingeführt. Die Mitglieder dieser Familie weisen die Leistungsdaten der 4000-Reihe auf, verfügen jedoch über die Anschlußbelegung der Standard-TTLs und sind weitgehend funktionskompatibel. Sie konnten aber weder die 4000er Reihe verdrängen, noch weckten sie sonderliche Begeisterung im Lager der TTL-Anwender. Allenfalls fanden sie als leistungssparender Ersatz für LSL-ICs (langsame störsichere Logik) Eingang in extrem langsame Schaltungen.

High-speed-CMOS

Mit der neuen Silicon-Gate-Technologie mit Oxid-Isolation führte man Anfang der 80er Jahre die 74-HC-Familie (High-speed-CMOS) und die 74-HCT-Reihe (High-speed-CMOS mit TTL-kompatiblen Eingängen) ein. Beide orientieren sich, was Pinout, Betriebsspannungsbereich (5 V) und Logik-Funktionen betrifft, an den gängigsten TTL-Chips. Damit treten sie in direkte Konkurrenz zu den LS-Typen, mit denen sie weitgehend austauschbar sind.

Es werden aber wohl nicht alle (zum Teil ja auch schon veralteten) Typen des TTL-Spektrums in High-speed-CMOS hergestellt werden, auf der anderen Seite gibt es statt dessen einige höherintegrierte Typen aus der 4000er Reihe und völlige Neuentwicklungen.

Die meisten HC(T)-Chips können es in puncto Geschwindigkeit und Ausgangsstrom mit den LS-Typen aufnehmen. Wenn man aber LS-TTL-Chips zwecks Stromersparnis durch High-speed-CMOS ersetzen will, so sollte man die darauf abgestimmte HCT-Serie verwenden und beachten, daß die HC(T)-Chips nur die halbe Treiberleistung von LS-TTLs haben (mehr dazu im Beitrag 'HCT kontra LS in der Praxis' in diesem Heft).

Advanced CMOS

Schon die HC-/HCT-Landschaft ist nicht ganz einheitlich, was technische Daten betrifft, und bei den Advanced CMOS-Chips (AC-/ACT) gilt das noch viel mehr. So besteht im Moment anscheinend nicht das Verlangen, einen verbindlichen CMOS-Standard zu schaffen, sondern die Hersteller versuchen derzeit, sich mit 'Traumdaten' gegenseitig zu übertreffen. Diesen Sachverhalt sollten Sie auch bei der Bewertung unserer Tabellen berücksichtigen, denn fast jeder Hersteller kitzelt hier und da noch eine Nanosekunde weniger Gatterlaufzeit und ein Milliampere Treiberstrom mehr heraus.

Die FACT-Serie von Fairchild ist ein solches Musterbeispiel für Advanced CMOS-ICs, bei deren Schaffung alle herstellungstechnischen Register gezogen und in beeindruckender

Kennwert		HCT	FACT	ALS	AS	Einheit
Betriebsspannungsbereich	min. max.	2,0 6,0	2,0 6,0	4,5 5,5	4,5 5,5	V V
Temperaturbereich	min. max.	−40 +85	−40 +85	0 +70	0 +70	°C °C
Störspannungsabstand (worst case, statisch)	low high	0,7 2,9	1,25 1,25	0,3 0,7	0,3 0,7	V V
Treiberstrom	I_{OH} I_{OL}	4,0 4,0	24,0 24,0	0,4 8,0	2,0 20,0	mA mA
DC-Fan-out 74xx00, jeweils ungünstigster Wert aus high/low		10	60	20	50	LS-Lasten
Leistungsaufnahme	statisch bei 1 MHz	(unterhalb 10 µW) 0,17	 0,8	1,0 1,2	8,5 8,5	mW/Gatter mW/Gatter
Geschwindigkeits-Leistungs-Produkt (bei 100 kHz, C_L = 15pF)		1,4	0,4	4,0	13	pJ
typische Verzögerungszeit eines 74xx138 bei C_L = 50pF		20*)	8,0	13,0	5,8	ns
Mindestwert der maximalen Taktfrequenz eines D-FF 74xx374		24	100	35	125	MHz

*) vermutlich bei C_L = 15 pF

Das gern angeführte Geschwindigkeits-Leistungs-Produkt (Speed Power Product) wird in keinem unserer zahlreichen Datenbücher definiert. Etwas Probieren mit den Werten im TI-Pocket-Guide ergab, daß dort das Produkt aus typischer Verzögerungszeit (siehe Bild links oben) und Leistungsaufnahme bei 100 kHz angegeben wird.
Alle Daten der Nicht-FACT-ICs wurden dem TI-Pocket-Guide Vol.1, 1985 entnommen, da diese in sich schlüssig waren. Eine ähnliche Tabelle von Fairchild stellt zum Beispiel das Geschwindigkeits-Leistungs-Produkt für statischen Betrieb (0,01) einem ALS-Wert von 6,0 pJ gegenüber, und bei der Verzögerungszeit des xx138 gibt man für ALS 8,5 ns an. Überhaupt zeichnet sich das FACT-Buch von Fairchild durch dauernde Widersprüche zwischen Textwerten und Grafiken und völlig unzulänglichen Wertangaben in den Datenblättern aus. Wir haben daher die glaubwürdigsten Werte aus dem FACT-Buch genommen und teilweise Grafiken extrapoliert. Dennoch darf man diese Tabelle nicht überbewerten.

Die charakteristischen Werte eines ALS- und eines FACT-Chips, zweier der zur Zeit modernsten Vertreter aus dem TTL- und CMOS-Bereich, im direkten Vergleich.

Nanohenry blockieren den Fortschritt

Neue IC-Anschlußbelegung für schnellere Chips

Eberhard Meyer

Seit Einführung der Logik-ICs ist ihr Produkt aus Geschwindigkeit und Leistungsverbrauch so weit verringert worden, daß die AS-Familie beginnt, der superschnellen ECL-Logik Konkurrenz zu machen. Es zeichnet sich auch immer deutlicher ab, daß die zukünftige Entwicklung von bipolar zu CMOS tendiert, denn schon schickt sich die neueste Chip-Generation in Advanced CMOS an, der Advanced Schottky-Logik den Rang abzulaufen.

Im Moment steht der zügigen Einführung der Chips in Advanced CMOS aber Uneinigkeit der Hersteller darüber im Wege, wie ein ganz bestimmter physikalischer Effekt am besten zu behandeln ist: Selbst in den Entwicklungslabors der Halbleiterindustrie hat man offensichtlich die massiven Störungen unterschätzt, die die Induktivität der feinen Drähte vom eigentlichen Chip zu den Gehäuseanschlüssen verursachen können.

Und dabei sind die Vorgänge (hinterher ist man immer schlauer) leicht erklärbar: Stellen Sie sich vor, Sie schalten in die GND- und 5-V-Leitung eines 74HC245 je eine kleine Spule. Im statischen Betrieb wird dies keine Auswirkungen haben, denn der Gleichspannungsabfall über den Spulen ist vernachlässigbar klein.

Wenn jedoch durch Änderung der Eingangspegel ein Ausgang umschaltet, zieht das IC eine kleine Stromspitze über die Zuleitungen der Stromversorgung. Dort jedoch verhindern die Induktivitäten einen schnellen Stromanstieg. Die Folge ist sowohl ein Einbruch der positiven Versorgungsspannung um vielleicht 0,5 V auf 4,5 V als auch ein 'Abheben' des GND-Pin auf zum Beispiel 0,5 V.

Je mehr Ausgänge dieses 74HC245 gleichzeitig schalten, desto stärker tritt dieser Effekt in Erscheinung. Und wenn sieben der acht möglichen Treiber gleichzeitig ihren Ausgangszustand wechseln (was gar nicht so selten passiert), wird am Ausgang des achten Gatters eine erhebliche Störamplitude zu messen sein.

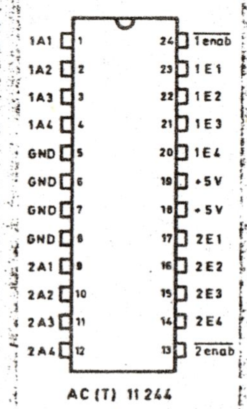
Kurze Zeit nach Auslieferung der ersten Muster von ACL-Bausteinen beklagten die Entwickler seltsame Effekte. Obwohl man den ICs im einzelnen nichts Fehlerhaftes 'nachmessen' konnte, funktionierten die damit aufgebauten Schaltungen nicht einwandfrei. In den Labors von Texas Instruments stellte man dann tatsächlich fest, daß die Induktivität von 14nH, die die Versorgungsleitungen im Innern eines 20poligen DIL-ICs aufweisen, beim Umschalten mehrer Ausgänge tatsächlich zum Problem werden.

Da die Störspannung direkt proportional zur Induktivität und zur Stromanstiegsgeschwindigkeit ist ($u = -L * di/dt$), gibt es genau zwei Wege, des Problems Herr zu werden:

Zum einen könnte man die Stromspitzen verkleinern und die Flanken flacher machen. Dies kollidiert jedoch mit dem Bemühen um die größtmögliche Arbeitsgeschwindigkeit der ICs. Zum anderen kann man versuchen, die Induktivität der Zuleitungen zu verringern.

Den ersten Weg beschreitet eine Gruppe mit der Firma Fairchild an der Spitze, die mit ihrer FACT-Logik nicht das Allerletzte an Geschwindigkeit herausholt, dafür aber die ICs mit der herkömmlichen TTL-Anschlußbelegung anbietet. Unter Führung von Texas Instruments versucht die zweite Gruppe den radikalen Neubeginn der anderen Lösung.

Die Stromversorgungsleitungen der neuen ACL-Bausteine liegen nicht mehr an den Ecken der IC-Gehäuse, sondern in der Mitte. Dieses neue Layout wird als 'Center-Pinning' im Gegensatz zum traditionellen 'End-Pinning' bezeichnet. Um die Zuleitungsinduktivität weiter zu verringern, hat man sogar mehrere GND- und 5-V-Pins vorgesehen.



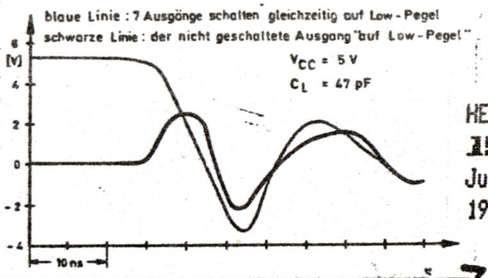
Der gute alte 244 im Center-Pinning-Gehäuse hat vier Beine mehr. Die Induktivität beträgt dadurch an GND nur noch 1,2nH, an +5 V 1,7nH.

Immerhin gelang es dadurch, die störenden Induktivitäten zum Beispiel beim oben genannten 245 auf etwa ein Zehntel des ursprünglichen Wertes zu senken. Im verkleinerten SO-Gehäuse (für Oberflächenmontage) sinkt die Induktivität noch einmal um den Faktor zwei. Es sind also Werte von deutlich unter 1 nH erreichbar.

Allerdings erfordern die neuen Gehäuse zunächst ein Umdenken bei den Schaltungsentwicklern, und wegen der erhöhten Anzahl Pins kommt ihr Einsatz auch etwas teurer. Am unangenehmsten aber dürfte sein, daß das neue Pinout die Umrüstung bestehender Produkte auf ACL verhindert.

Für den Markterfolg des neuen Konzepts ist aber auch die internationale Normung entscheidend. Nur diese stellt dem Anwender eine störungsfreie Belieferung durch mehr als einen Anbieter über lange Zeiträume hinaus sicher. Deshalb versucht TI zur Zeit, für das neue Pinout der ACL-Serie eine Normung durchzusetzen, woran der Konkurrenz natürlich gar nichts liegt. Ein anderes Pinout sei nicht erforderlich, da man die Stromspitzen auch technologisch in den Griff bekomme, heißt es beim Kontrahenten.

Doch selbst wenn Fairchilds Auffassung zum gegenwärtigen Zeitpunkt vertretbar ist, werden noch schnellere Logik-Familien in Zukunft wieder den Ruf nach dem Center-Pinning laut werden lassen. So hat sich Texas Instruments bei ACL, der derzeit schnellsten CMOS-Familie, jetzt schon auf das Center-Pinning festgelegt. Damit dürfte ein heftiger Kampf um den Markt der 90er Jahre eingeleitet worden sein.



Im alten Corner-Pinning-Gehäuse können beim Umschalten vieler Ausgänge so starke Einbrüche der Betriebsspannung entstehen, daß auch statische Ausgänge 'in Bewegung' geraten können.

Walter Piller
Rohnenstrasse 8
CH-8835 Feusisberg
Tel. 01/784 74 18

BÖRSE -- BÖRSE -- BÖRSE

Software Modell I TRS 80 (aus Liquidation)

E = englisch / D = deutsch

Accounts Receivable	E	DM. 15.--
Business Mailing List	E	"
Disk Mailing List	E	"
Inventory Control	E	"
Manufacturing Inventory Contr.	E	"
Disk Operating Syst. Disk Basic	E	"
Inventory Control System	D	"
FIBU Hauptbuchhaltung D-System	D	"
Adressenerfassung (Vereine)	D	"
Geschäftsadress Programm	D	"
Versafile (Info-System Aufbau)	D	"
Termin überwachung	D	"
Wärmebedarf Berechnung	D	"
Sortier-Routine	D	DM. 5.--
5 Spiele auf Kasetten	E	"
FORTRAN (ausverkauft event. Copy)	E	DM. 80.--
Editor/Assembler (Macro 80)	E	80.--
Bedienungshandbuch mit Basic	D	gratis

Lieferung solange Vorrat

Versandkosten Anteil DM. 5.--

Angebot!!!

Als Reaktion auf eine Vorstellung des CLUB 80 in der PASCAL international 2/87 hat mir Herr Waldherr folgendes Angebot zukommen lassen (Auszug aus dem Brief):

"Das Peutinger-Gymnasium hat für seine Informatikabteilung, die ich betreue, neue Maschinen gekauft. Von 12 TRS 80, die ich im Auftrag der Schule verkaufe, sind noch 4 vorhanden."

Nach telefonischer Rücksprache sieht das Angebot so aus:

2 * TRS 80 M.1 16k ohne Erweiterungen für VB 150,-- DM
2 * TRS 80 M.1 48k inkl. Floppy-controler für 250,-- DM
Zu den Maschinen gehören natürlich Netzteil und Monitor sowie die Originalhandbücher.
1 * Druckerumschalter 4 in 1 (vier Computer an einen Drucker) für VB 100,-- DM

Nähere Auskünfte können Interessenten erhalten bei:

Michael Waldherr
Schöner Graben 31
7090 Ellwangen/Jagst
Tel.: 07961 / 6994

Hartmut Obermann

Gesucht: BSORT M4, BSORT 51
Wer hat für diese Programme die Anleitung?
H. Obermann

Verkaufe neues JVC-Disk-Laufwerk
40 DS DD Slimline
Preis incl. weißem Slimline-Stahlblech-gehäuse 220,-DM VB
oder incl. Netzteil und Eigenbaugehäuse 250,-DM VB
H. Obermann

Preiswert, nicht billig!!!

Schon seit ca. einem Jahr habe ich nun mein Model 4p. Genau so lange steht ein, kurz vor dem Kauf des 4p angeschaffter Monitor ungenutzt bei mir im Keller. Es handelt sich dabei um einen Zenith Computermonitor mit grüner Bildröhre. Der Monitor war praktisch nie in Gebrauch und ist so gut wie neu! Um meinen Geldbeutel etwas voller und den Keller etwas leerer zu machen, würde ich mich für 120,-- DM (Festpreis) von dem Gerät trennen. Wer also einen neuen Monitor braucht, gleich bei mir melden!

Hartmut Obermann

INFOLGE ÜBERFUSSES

ZWEI LAUFWERKE (fast) ZU VERSCHENKEN!

1 SHUGART 400, 35 Track, einseitig, SD/DD — DM 50,--
1 BASF 6106, 40 Track, einseitig, SD/DD — DM 75,--

Beide Laufwerke sind als LW 0-2 konfigurierbar (ggf. bitte Gewünschtes angeben!) und befinden sich in gutem Zustand. Die genannten Preise gelten bei Abholung, andernfalls + Porto.

Richard Rensch, Bahnhofstraße 100, 7128 LAUFFEN am Neckar

Hallo Jens,
falls vom Club jemand
Interesse hat, soll er sich
so bald wie möglich bei
mir melden.

Bitpit

7550/3

Genie-TRS80-User-Club
Peter Spiess

Trugenhofener Str. 27

D-8859 Rennertshofen 1

ist leider gestorben
Bitpit

. 13.3.1987

Lieber Computer-Freak!
Lieber Computer-Fachmann!

Auch Computer-Spezialisten müssen einmal ausspannen und die Gehirngänge einmal in unserer Bergluft ausruhen lassen.

Wir möchten daher speziell für Computer-Clubmitglieder und Betreiber einen erholsamen Wanderurlaub in Filzmoos anbieten.
Filzmoos ist ein Ort in 1000 m Seehöhe am Südhang des Hohen Dachsteins und bietet über 200 km Wanderwege und wunderbaren Skilauf auf den nahegelegenen Gletscher.

Unser spezielles Angebot für Computerclubs in der Zeit vom:

3. Oktober bis 17. Oktober

1 Woche Halbpension (Frühstücksbuffet - Menüwahl)
1 Begrüßungscocktail
1 Kaffee- und Kuchenparty
1 Geführte Wanderung mit dem Chef
mit Wanderjause

ÖS. 2.400,-

Wir würden uns freuen, Sie in unserem Hause verwöhnen zu dürfen. Wir verbleiben inzwischen bis zu einer etwaigen Rückantwort

mit freundlichen Grüßen
aus dem tiefverschneiten Filzmoos
Familie Steinbacher

PS: Auf Wunsch senden wir Ihnen gerne Prospekte zu!

Hallo Kollegen,

irgend jemand hat sich im letzten Clubinfo beschwert, daß einige nur passive Mitglieder sind, da ich mich auch damit angesprochen fühle und es mir so langsam reicht mich mit einem Betriebssystem herum zu ärgern werde ich jetzt aktiv.
Von KJ habe ich mir ein HDOS bestellt, da dieser aber nur 40 Track Laufwerke hat war nichts mit HDOS. Nach ein paar Telefonanrufen hatte ich den Peter Spieß an der Strippe. Er sagte mir, daß er ein HDOS auf 80 Track habe und ich sandte ihm eine Disk.
Mit folgenden PD-Parametern
TI=EHK,TO=6,SP=80,SEK=36,SWZ=0,EIB=6,SBIV=48,AEIV=6
Die Antwort ließ nicht lange auf sich warten. Leider trat kein Erfolg ein bei einem anbooten, es erschien nur "HDOS" sonst nichts wie bei KJ's Disk. Da ich anderweitig beschäftigt war kam ich erst jetzt dazu mir die Sache etwas genauer anzusehen. Ich schaute einmal in die "technische Beschreibung" des Genies und siehe da, da gab es auf der Floppykarte Jumper. Ich erinnerte mich auch dunkel daran, daß als ich mein "Spielzeug" bekam ein ähnlicher Fehler bestand und ein Herr NOLDEN von TCS mir sagte ich solle den obersten Jumper nach ganz oben stecken, danach lief alles. Also kurz diesen wieder nach unten und siehe da es meldete sich HDOS. So erstes Hindernis überwunden.
Aber schon taten sich neue Probleme auf.
1.Die Hochpfeiltaste ist mit Ä belegt
2.KEY/CMD läuft nicht mehr Shorthand von BASIC sind weg und die Tastaturentprellroutine
3.Entweder bin ich zu dumm das Handbuch zu lesen oder kann mir jemand mitteilen wie man z.B. die schönen Copybefehle eingibt wie z.B. mit IDL oder FRAG etc.

Vielleicht findet sich auch jemand der mir ANLEIT/DOS und REFCARD/DOS ausdruckt, da ich keine Beschreibung von TSCRIPS habe und keinen passenden Drucker.

Apropos Drucker und Hardware etc.
Ich habe einen Genie IIS (einen der Ersten),
2 780k Laufwerke 80 Track DS/DD,
1 HP-ThinkJet Drucker,
ärgere mich mit GDOS und HDOS herum,
als Textverarbeitung habe ich NAME/CMD von TCS (Asembler)
geschäftlich arbeite ich mit größeren Maschienen wie:
PDP11/23,11/23+ mit RT11 ,11/73 mit RSX und seit neuestem mit einem Siemens PC16-20 (mit 80286) und MSDOS bis jetzt aber vorwiegend als Benutzer von fertiger Software.

Oliver Voß

* DER – DIE – DAS *
* macht mir Spaß! *

(Ein Leserbrief)

Liebe Freunde (und Freundinnen?) der letzten GENI-, TRS-I-, PERSI-, INDI- und MOHIK-ANER !

Seid ihr zum letzten Gerangel gerüstet?
Schilde, auch Klingen und Nasen geputzt?
Fingernägel und Bärte gestutzt?
Ob's euch nach Widerspruch dann noch gelüstet??

Hinreichend unbeliebt bin ich inzwischen wegen meiner ständigen Nörgelei über Orthographie und Interpunktion. Genau das wollte ich! Nicht, um die Schreiber zu belehren oder gar zu "vergratzen". Das könnte dem Zustrom an Beiträgen schaden (? Es würde ohnehin nichts nützen). Nein: Weil es egoistisch ist und unwahrhaftig, sich "beliebt zu machen".

Ein junger Franzose adligen Geblüts, direkter Nachfahre von Montesquieu, der einige Wochen mein Gast war, schrieb das erbärmlichste Französisch, das ich je gelesen. Daraufhin von mir (wie immer!) angesprochen, antwortete er nur: "Comme ci, comme ça – c'est la vie! C'est ne pas importante!" Dieses "Laissez faire" ist die wahre Freiheit von ungeliebten Zwängen...

Schließlich ist es ja auch unsere Sache, *was wir mit und aus unserer Sprache machen – mit unserem Eigentum dürfen wir ja machen, was uns beliebt (??). Aber nicht mit dem anderer Leute. Auch nicht mit dem Englischen. Oder den lediglich "eingedeutschten" Wörtern.*

Was soll man mit den englischen Begriffen machen, die nun wohl unausrottbar in die "Fachsprache" eingegangen sind?

Mit der Artikelsuche fängt es bereits an! Heißt es nun "der File" – "die File" – oder "das File" ?
Alle drei Artikel sind in Gebrauch. Müssen zwei nicht falsch sein?

Was heißt hier "falsch" oder "richtig"? Ich habe lange hierüber nachgedacht. (Lacht nicht! Ihr habt Besseres zu tun!)

Der Engländer unterscheidet die Geschlechter nicht. (Der Franzose kennt wenigstens zwei!) Sachlich und nüchtern, wie er ist, kennt er nur "the" – für ihn offenbar Ausdruck der Sachlichkeit, also Sächlichkeit: *Als kühler Rechner denkt er neutral und weder männlich noch weiblich (seine Insel wäre längst übervölkert – stellt euch das vor! * Yet – man verwechsle Neutralität nicht mit Objektivität, you know: "Right or wrong – my country!")*

Sollten wir uns somit ständig des Artikels "das" befleißigen? Etwa so:
Das Computer – das Diskette – das Chip – das Pin – das Interpreter – das Compiler – das CPU – das Record usw.usf.? *Klingt komisch, nicht? * Nicht komisch klingt jedoch:*
Das Keyboard – das Manual – das Memory – das Byte!

Woran liegt das?
Offenbar doch daran, daß wir bei Anhörung und Benutzung der meisten Begriffe sofort unbewußt an ihr deutsches Analogon denken und deshalb dafür den entsprechenden deutschen Artikel benutzen. Denn es heißt ja:
Das Tastenfeld – das Handbuch – das Gedächtnis – das Wort usw.

Hingegen sagen wir nicht:
Das Rechner, sondern der Rechner (deshalb "der" Computer);
Das Scheibe, sondern die Scheibe (deshalb "die" Diskette);
Das Baustein, sondern der ----- (deshalb "der" Chip);
Das Stift *), sondern der Stift (deshalb "der" Pin);
wir sagen: Der Übersetzer (deshalb "der" Interpreter);
wir sagen: Der Sammler (deshalb "der" Compiler);
wir sagen: Der Bericht (deshalb "der" Record <auch der Rekord>);
wir sagen: Die Zentraleinheit (deshalb "die" CPU);
**) (dies ist etwas für Damen)*

So einfach ist also das Rezept: Man denke Deutsch – und schon "hat man's"! – Soo?? Und wie sollen wir zu "BUS" *sagen? "Der* BUS"? (Omnibus?) – *Und zu* "FILE"? *Wie lautet denn die richtige Übersetzung von* "FILE"?
Ich habe im WILDHAGEN *nachgeschaut, dem besten und umfangreichsten Wörterbuch (2 Großbände Englisch/Deutsch/Englisch mit sämtlichen Idioms/Redewendungen; entspr. dem* LAROUSSE *für Französisch!). – Dort lesen wir:*
FILE = "der Faden" (das "Filament"!); "der Draht"; "die Feile"; "das Dokument"; "der Ordner"; "die Akte"; "das Aktenbündel"; "das Aktenheft"; "der Stoß Papiere"; "die Sammelmappe"; "der Gänsemarsch"; "die Reihe"; "in Reihe marschieren" (etymologischer Zusammenhang mit 'defilieren'!); "die Rotte" *– alle 14 Übersetzungen haben – das empfinden wir – etwas mit unserem Objekt – dem/der/dem File – zu tun! Alle drei Artikel kommen aber auch vor! Für welchen sollen wir uns entscheiden?→Also muß eine Konvention her.*

Als ich 1945 beim Ami als Tellerwäscher arbeitete und bald – nach gründlicher Prüfung durch den CIC – wegen meines IQ und angeblicher „confidability zum Aktenverwalter befördert wurde, vernahm ich oft die Anweisung (instruction oder statement?): "File it in!" – wobei man gleichzeitig auf einen der riesigen Ordner in einem der Fächer verwies (der Papierkrieg war nicht geringer als bei uns – im Gegentum...). Da sich der Intelligence Service seinerzeit geirrt hatte und mein IQ damals schon nicht hoch genug war, identifizierte ich "File" seitdem stets mit "Fach" (schließlich war ich ja ein "Fachmann" und kein "Ordner", vielmehr hatte ich den Ordnern ordentlich zu gehorchen, sonst wäre meine Familie verhungert, ohne daß ich dafür von irgendwem einen ordentlichen Orden erhalten hätte...) – kurzum: Für mich war "File" seitdem "d a s Fach", also auch: das File!*

* trustiness

Vielleicht hatte Uncle Sam aber auch "die Akte" gemeint und nicht "das Fach". Also "die File".
Oder "den Vorgang". Also "den File".
Ich weiß es bis heute nicht. Der IQ ist auch nicht nachgewachsen.

Seit der ersten Bekanntschaft mit der Elenden Droge Vexiergeist *("EDV") hielt ich viele Jahre lang ein "File" für das Programm und blieb deshalb beim sächlichen File. Später lernte ich, daß auch eine Datei, eine Datensammlung also, ein "File" ist: also eine File! Das kommt mir aber vor wie ein Transvestit. Ich bleibe – Männlein und Weiblein beiseiteschiebend – beim neutral übergeordneten, "rein Sächlichen": Das File.*
Wieder mit "die File"! – Ach nein, "mit" regiert den Dativ. Also: Wieder mit "der File"! – Zum Teufel, das klingt männlich. Sollte weiblich sein. Also am besten:

Es lebe d a s File!

Jetzt möchte ich euch Neudeutsche aber noch mehr verunsichern. Es geht um den/die/das Compiler! Nein, es geht um seine/ihre/seine Aussprache.

Meinst du es englisch, spricht es sich "Kompailer" (Schluß-r verschlucken). Das Verb "compile" spricht sich "kompail".

Meinst du es deutsch, so spricht's sich nicht "kompailieren" (oder gar "kompalieren" - auch schon gehört), sondern schlichtweg "kompilieren". Durch die Endung "...ieren" ist es nämlich zu einem deutschen (Lehn-)Wort geworden, und das sprechen wir so aus, wie es geschrieben wird.

Aber müßten wir dann nicht eigentlich auch "Komputer" sagen? Statt "Kompjuter"? - In der Tat, lieber Leser. Du sprächest das Wort dann so, wie es seiner vornehmen Herkunft geziemt: Lateinisch.

Aber du mußt nicht. Du sagst ja auch nicht Tatscher, wenn du die derzeitige Chefin von England meinst. Und zwar deshalb, weil du genau den richtigen Tatsch hast. Und weil du weder betatscht noch (wie ich vermute) be-touch-t bist, lieber Leser. Und welcher Schreiber ist schon betucht??
(OK - ich auch nicht!)

KaJot

Ein neuer 80-Zeichen-Treiber samt neuem DEBUG

Alle Clubmitglieder, die mit der 80-Zeichen-Karte von Schmidtke gestraft sind, mußten bis jetzt mit dem das HIMEM versauenden X80/CMD leben. Damit das anders wird (und aus diversen anderen Gründen), habe ich mir das ROM-Listing geschnappt, das DOS-Buch daneben gelegt und mir einen vollkommen neuen 80-Zeichen-Treiber überlegt. Er belegt nur den Platz des alten 64-Zeichen-Treibers im ROM und im SYS0. Der neue Treiber ist also nur lauffähig, wenn das das ROM überschrieben werden kann. Bei meinem IIs ist das kein Problem, denn dort ist 0000-3400 normales RAM, in das beim Booten ein Pseudo-ROM geladen wird.

Der 80-Zeichen-Treiber arbeitet mit allen Programmen, die ihre Ausgaben über 0033h abwickeln. Springt ein Programm direkt in die Bildschirmroutinen, stürzt es wahrscheinlich ab. Ein Feature habe ich allerdings nicht aus den ROM-Routinen übernommen: die 32 CPL bzw. bei 80 Zeichen 40 CPL. Die Vergrößerung über Port FF klappt bei 80 Zeichen sowieso nicht und wozu braucht der normale Mensch schon die halbe Bildschirmbreite?

Es gab bis jetzt noch immer ein Problem, das gegen die 80-Zeichen-Darstellung sprach: DEBUG schreibt direkt in den Bildschirmspeicher und ist somit bei Einschalten der Schmidtke-Karte disabled. Dieses Problem mußte auch beseitigt werden und somit habe ich kurzerhand die kritischen Stellen in DEBUG umgeschrieben, so daß jetzt alle Ausgaben über den Bildschirm-DCB gehen. Dadurch ist DEBUG sowohl auf 64 als auch auf 80 Zeichen lauffähig. Allerdings benötigte ich etwas Platz, um die neuen Ausgabe-Routinen in SYS5 unterzubringen und warf deshalb die Kommandos "Lesen" und "Schreiben" eines Sektors heraus.

Der neue 80-Zeichen-Treiber und das neue SYS5 sind bei unserem neuen Clubbibliothekar erhältlich.

Gerald Schröder

CP/Mac - na und? ...

... wird sich mancher Leser von Richard Renschs begeistertem Artikel gesagt haben. Wer nämlich G-DOS/NEWDOS-80 hat, pfeift nur zu gern auf CP/M. Ich bin zwar ebenfalls ein geradezu fanatischer CP/M-Ignorant (auch i. S. der wörtlichen Übersetzung: ignorare = nicht kennen, nicht wissen), mußte aber erkennen, daß es ohne nun mal nicht geht. Dabei denke ich weniger an WordStar und die anderen von Richard erwähnten Programme. Die laufen unter CP/M natürlich auch ohne CP/Mac; sie auf eine G-DOS-Disk zu kopieren, macht wenig Sinn. Vielmehr ist es die Möglichkeit, eben gerade nicht *unter* CP/M arbeiten zu müssen (igitt!) und dennoch *für* CP/M arbeiten zu können.

Beispiel: Mein gelegentlich hart geforderter Prommer soll angeblich soft kaputtzukriegen sein. So habe ich es bisher nicht gewagt, ein eigenes G-DOS-fähiges Programm für ihn zu schreiben. In der c't wurde leider nur PROM.COM für CP/M angeboten. Aber ich denke natürlich nicht im Traum daran, EPROM-Inhalte mühsam unter CP/M zu entwerfen und zu testen. Sie entstehen lustbetont mit ZEUS unter G-DOS. Wenn alles zum besten steht, brauche ich nur noch den Käfer zu brennen. Das geht nun aber leider nur mit PROM.COM. CP/Mac zieht das File für mich auf die CP/M-Disk.

Weiteres Beispiel: Mein kleines selbstgestricktes Programm für den Akustikkoppler war ja gut und schön. Die Aussicht, viele Stunden mit Programmierarbeit zu verbringen, um die Leistungen von LYNC.COM zu erreichen, war aber nicht allzu verlockend. Dasselbe Problem: Sollte ich zu übertragende Texte oder Listings mit einem Monster wie WordStar erstellen? Ich doch nicht! Was TSCRIPS oder ZEUS erledigt hat, kann LYNC sodann bequem mit CP/Mac hingeblättert werden.

Die Aufzählung ließe sich nahezu beliebig fortsetzen. Wer CP/M häufiger als ich benutzen muß, wird davon ein Lied zu singen wissen. Denken wir nur an die bequeme Editionsmöglichkeit von CP/M-Files, ohne tränenüberströmt von G-DOS Abschied nehmen zu müssen. Diese großen Vorteile für die Hack-*Praxis* auch und gerade des *DOS*-Fans sind es, die ich Richards Ausführungen noch nachtragen wollte.

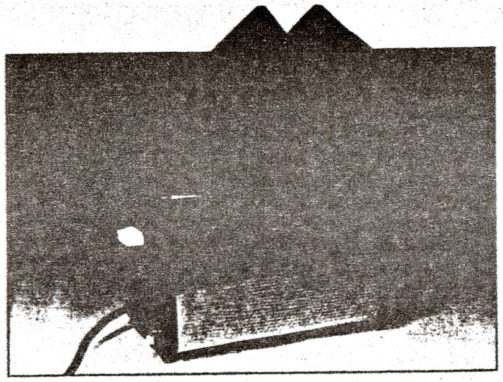
Den Autor Ulrich Heidenreich kenne ich seit Jahren. Er war am Anfang sozusagen mein Assembler-Mentor. Was aus dieser Quelle kommt, sind Nägel mit Köpfen.

Arnulf Sopp

TANDY VERTRAGSHÄNDLER

Computer Schorradt	Hahnerberger Str. 19	5600 Wuppertal 12	0202/401306
Bürosysteme H. Jans	Am Strampel 28	4460 Nordhorn	05921/76102
CD Computerdienst	Untergasse 1	6973 Boxberg	07930/2071
STA-Data Control	Rohrbacherstr. 27	6900 Heidelberg	06221/13093
Hohendorf Hard & Software	Freudenbergstr. 53	6200 Wiesbaden	06121/24727
Doris Koepke	Bitzenstr. 11	5464 Asbach	02683/4547
Klick Hard & Soft	Unterkohlfurt 20	5600 Wuppertal	0202/471929
CC Computer GmbH	Elisabethstr. 5	4600 Dortmund 1	0231/528184
Wilhelm Muscheid	Kampenstr. 82	5900 Siegen	0271/4881
Reinhard Noedler	Auf dem Klengenberg 10	3422 Grossalmerode	05604/7672
Digital Computer	Knesebeckstr. 76	1000 Berlin 12	030/8827791
Grunert Datentechnik	Kapuzienerstr. 13	7888 Rheinfelden	07623/63535
Thieme & Wirtz	Mechernicherweg 57	5353 Mechernich	02443/3809
hard & soft GmbH	Herrenkellergasse 16	7900 Ulm	0731/62699
Galster EDV	Erlanger Str. 19	8510 Fürth	0911/797624
Lauer & Schreitmiller	Blumenstr. 2	8900 Augsburg	0821/312071
Elektro Burger	Leimitzer Str. 11-13	8670 Hof	09281/40075
Klöpper & Wiege	Pöstenweg 76	4920 Lemgo	05261/14770
Computertechnik Joerss	Wuppertaler Str. 58	5650 Solingen	0212/51637
Richter Elektronik	von-Ketteler-Str. 32	5090 Leverkusen	0214/65478
Ing. S. Piringer	Weningstr. 1	8380 Landau	09951/8601
Pfotenhauer Microcomputer	Neulandstr. 16	7590 Achern 2	07841/5056-7-8
Füssner - Computersysteme	Am Bauhof 19	4440 Rheine	05971/12539
Kirchner Elektronik	Finkenstr. 5	4100 Duisburg	0203/376165
Fischer Computer	Gottlieb-Daimler-Str. 7	7730 Villingen	07721/70046
Computerstube BARBER	Fuchskamp 4	3032 Fallingbostel	05162/3818
FMS - Computer	Lindesmühl Promenade 10	8730 Bad Kissingen	0971/4044
Wagner GmbH	Fürther Str. 338	8500 Nürnberg	0911/329060
Digitronik	Am Kamp 17	2082 Holm	04103/88672
MCT Loster	Hochfeldring 24	8446 Mitterfels	09961/7122
Haller Computer GmbH	Büttnerstr. 29	8700 Würzburg	0931/16705

Einer unter Euch wird mich verraten.



Datenverluste

Was Disketten übelnehmen

Jochen Blumenthal

Da gibt es kein Vertun – Disketten sind die am häufigsten eingesetzten Datenträger. Und was tut man nicht alles, um die wertvollen Daten auf der kleinen Kunststoffscheibe vor allen denkbaren Widrigkeiten zu schützen. Wie man weiß, können Magnetfelder Schlimmes anrichten. Doch wie sensibel sind Disketten nun wirklich gegenüber Magnetfeldern, und was können ihnen elektrostatische Felder oder gar Röntgenstrahlen anhaben?

Um dies ein für allemal festzustellen, führte ich eine Reihe von Experimenten durch. Alle Versuche wurden mit derselben 5 1/4"-Diskette durchgeführt, um die Bedingungen möglichst konstant zu halten. Lediglich bei den Röntgen-Versuchen benutzte ich verschiedene Disketten, jedoch vom selben Fabrikat. Das Testobjekt wurde vor dem Versuch frisch formatiert und anschließend auf fehlerhafte Sektoren untersucht. Als Analyse-Verfahren diente die Scan-Funktion des Kopierprogramms 'Double Image' im C64-Modus eines C128.

Um das Wesentliche gleich vorwegzunehmen: Weder Röntgenstrahlen noch starke elektrostatische Ladungen konnten den Testdisketten billigster Qualität etwas anhaben. Einzig magnetische Felder sind der Freßfeind des Datenbits.

Der weitverbreitete Brauch, Disketten in Alufolie einzuwickeln und sich davon Schutz für dieselben zu erhoffen, hat lediglich zur Folge, daß sie später für das Frühstücksbrot fehlt. Magnetfelder lassen sich lediglich durch Materialien bremsen, die selbst magnetisch sind, also zum Beispiel Stahlblech oder Weißblech. Als 'magnetische Leiter' können sie Magnetfelder nach der Art eines Faradayschen Käfigs wirkungsvoll abschirmen und verhindern, daß die magnetischen Feldlinien ihren todbringenden Weg durch die Diskette nehmen.

Alufolie ist jedoch als elektrischer Leiter in der Lage, elektrische Feldlinien umzuleiten, und eignet sich weiterhin zum Schutz von Bauteilen, die gegen statische Aufladung empfindlich sind. Das Deponieren von Disketten auf Floppy-Laufwerke oder Monitore brachte keine Veränderung des Disketteninhalts. Es sei hier jedoch vor schlichter Wärmeentwicklung besonders gewisser Commodore-Laufwerke gewarnt. Sie läßt Disketten, die auf den Lüftungsschlitzen abgelegt sind, bis zur Unbrauchbarkeit verbiegen.

Selbst ein einstündiges Ankleben der Testdiskette auf die Bildschirmmitte eines laufenden 70-cm-Fernsehers brachte außer hörbarem statischem Knistern nichts Erwähnenswertes. Eine pulsierende, 2 cm lange Funkenstrecke, in 3 cm Entfernung mehrfach über die Diskette geführt, ließ dieselbe völlig ungerührt.

Beim Röntgenexperiment wurden Disketten – ungeschützt und in Stahlblechschachteln verpackt – von einem zahnärztlichen Röntgengerät aus 50 cm Entfernung beschossen. Überraschenderweise erlitten die Daten dabei in keinem Fall einen Schaden.

Um die Anfälligkeit gegen magnetische Felder zu untersuchen, wurden Versuche mit einem Permanentmagneten von 80 mm Durchmesser und einer Höhe von 22 mm durchgeführt. Seine Anziehungskraft reichte aus, um eine Stahlplatte von etwa 10 kg zu tragen.

Beim Angriff auf die Daten ließ ich den Magneten jeweils fünfmal im Uhrzeigersinn über der Diskette kreisen. Verblieb der Magnet für einen längeren Zeitraum über der Diskette, so brachte dies keine höhere Fehlerrate. Die Diskette mußte diese Behandlung bei verschiedenen Abständen des Magneten erdulden, wobei sie entweder ungeschützt oder in Schachteln aus V2A-Stahl beziehungsweise Weißblech verpackt war.

Es sei angemerkt, daß V2A-Stahl zwar nicht magnetisierbar ist, dafür aber keine abschirmende Wirkung besitzt. Boxen aus diesem Material eignen sich also nicht zum Schutz von Disketten. Besser geeignete Materialien sind Stahlblech oder Weißblech, wobei Weißblech noch den Vorzug der leichteren 'Lötbarkeit' besitzt. Eine selbstgefertigte Weißblechschachtel sollte aber auf keinen Fall mit der Weller-Lötstation verarbeitet werden, da die Temperaturregelung (Magnastat) mit einer permanentmagnetischen Lötspitze arbeitet und die Blechschachtel magnetisiert. Ein Programm ging mir aus diesem Grund verloren.

Als Resümee kann man sagen, daß dann absoluter Schutz gegeben ist, wenn sich die Diskette in einer Weißblech- oder Stahlschachtel befindet und durch Schaumstoffklötzchen an Deckel und Boden eine Mindestdistanz von 10 mm zu den Wänden ihres Behältnisses einhält.

Ungeschützte Diskette:

Abstand zum Magneten:	Defekte Blöcke:
9 mm	keine Fehler in fünf Versuchen
8 mm	125 von 664 Blöcken defekt
7 mm	189 von 664 Blöcken defekt

Durch Weißblechschachtel (Stärke 0,2 mm) geschützte Diskette:

Abstand zum Magneten:	Defekte Blöcke:
4 mm	keine Fehler in fünf Versuchen
3 mm	31 von 664 Blöcken defekt
2 mm	254 von 664 Blöcken defekt

Durch V2A-Stahlschachtel (Stärke 0,1 mm) geschützte Diskette:

Abstand zum Magneten:	Defekte Blöcke:
9 mm	keine Fehler in fünf Versuchen
8 mm	65 von 664 Blöcken defekt
7 mm	154 von 664 Blöcken defekt

Die Versuchsergebnisse bei verschieden starker Annäherung eines starken Permanentmagneten.

ct

Computerkriminalität

Eine Übersicht von Rechtsanwalt Mathes, Eschwege

Nach langjähriger Entwicklungs- und Beratungsphase ist am 15.05.1986 das »zweite Gesetz zur Bekämpfung der Wirtschaftskriminalität (zweites WiKG)« verkündet worden, welches am 01.08.1986 in Kraft getreten ist. Dieses Gesetz knüpft an das erste Gesetz zur Bekämpfung der Wirtschaftskriminalität vom 29.07.1978 an und ergänzt dieses in erster Linie in den Bereichen sozialschädlichen Verhaltens unter Ausnutzung und Verwendung moderner Technologien, insbesondere der Datentechnik.

So wurden teilweise völlig neue Straftatbestände geschaffen und im übrigen bereits bestehende Strafvorschriften ergänzt.

Im folgenden soll ein Überblick über die wesentlichen Neuerungen gegeben werden.

1. Ausspähen von Daten
(§§ 202 a, 205 Strafgesetzbuch)
»Wer unbefugt nicht unmittelbar wahrnehmbar gespeicherte oder übermittelte Daten, die nicht für ihn bestimmt und gegen unberechtigten Zugang besonders gesichert sind, sich oder einem anderen verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.«
Die Tat wird nur auf Antrag verfolgt (§ 205 Strafgesetzbuch).

Diese Vorschrift erweitert den bisherigen Bereich des Geheimnisschutzes. Die Tathandlung besteht darin, daß der Täter sich Daten, der zum Beispiel durch Paßworte, Benutzerkenn-Nummer, Magnetkarten oder in andere Weise zugangsgesichert sind, sich oder einem anderen verschafft.

Eine eigene Kenntnisnahme ist nicht erforderlich (z.B. bei Verschaffung verschlüsselter Daten).

Das reine »Hacking«, welches bedeutet, daß sich der Täter lediglich den Zugang zu den Daten eröffnet, ohne diese jedoch anzuzapfen, bleibt straflos. Ebenso wie der Versuch der Tat.

2. Computerbetrug
(§ 263 a Strafgesetzbuch)
»Wer in der Absicht, sich oder einem Dritten einen rechtswidrigen Vermögensvorteil zu verschaffen, das Vermögen eines anderen dadurch beschädigt, daß er das Ergebnis eines Datenverarbeitungsvorganges durch unrichtige Gestaltung des Programmes, durch Verwendung unrichtiger oder unvollständiger Daten, durch unbefugte Verwendung von Daten oder sonst durch unbefugte Einwirkung auf den Ablauf beeinflußt, wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft. Der Versuch ist strafbar.«

Tathandlung ist hier das Verschaffen eines rechtswidrigen Vermögensvorteiles durch Schädigung eines anderen, indem durch unrichtige Programmgestaltung eines Datenverarbeitungsvorganges oder durch Inputmanipulationen z.B. Leistungen erschlichen werden (Kindergeld für nicht vorhandene Kinder, Renten für längst Verstorbene etc.) oder aber ein solcher Erfolg durch die unbefugte Verwendung von an sich richtigen Daten (Codes, Identitätsnachweise, etc.) herbeigeführt wird.

Letzterer Fall soll insbesondere die Mißbrauchsmöglichkeiten, die sich aus der Automation des modernen Zahlungsverkehrs ergeben haben, unterbinden (bargeldloser Zahlungsverkehr über Selbstbedienungsterminals oder »Home-banking« über das BTX-System).

3. Fälschung beweiserheblicher Daten
(§ 269 Strafgesetzbuch)
»Wer zur Täuschung im Rechtsverkehr beweiserhebliche Daten so speichert oder verändert, daß bei ihrer Wahrnehmung eine unechte oder verfälschte Urkunde vorliegen würde, oder derartig gespeicherte oder veränderte Daten gebraucht, wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft. Der Versuch ist strafbar.

4. Datenveränderung
(§ 303, 303 a Strafgesetzbuch)
»Wer rechtswidrig Daten löscht, unterdrückt, unbrauchbar macht oder verändert, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.«

Diese Neuregelung schützt vor »Sachbeschädigung im Datenverkehr«.

5. Computersabotage
(§ 303 b Strafgesetzbuch)
»Wer eine Datenverarbeitung, die für einen fremden Betrieb, ein fremdes Unternehmen oder eine Behörde von wesentlicher Bedeutung ist, stört (Anmerkung des Verfassers: Und zwar durch bestimmte näher bezeichnete Sabotagehandlungen an einer Datenverarbeitungsanlage oder einem Datenträger oder aber auch durch bloße Datenveränderung), wird mit Freiheitsstrafe bis zu fünf Jahren oder mit Geldstrafe bestraft.«

Dieser neue Tatbestand soll einem störungsfreien Funktionieren datenverarbeitender Systeme in Wirtschaft und Verwaltung Rechnung tragen und durch die relativ hohe Strafandrohung verhindern, daß Täter durch derartige Betriebs- oder Wirtschaftssabotage Schäden anrichten, die unermeßlich sein können.

6. Softwarepiraterie
(§§ 108 a, 109 Urheberrechtsgesetz)
Hiernach wird mit Freiheitsstrafe bis zu fünf Jahren oder Geldstrafe belegt, wer gewerbsmäßig und unerlaubt ein urheberrechtlich geschütztes Werk vervielfältigt oder verbreitet.

Gerade kürzlich wurde ein »fanatischer Computerfreak« zu einer Freiheitsstrafe von sechs Monaten auf Bewährung und Zahlung einer Geldbuße in Höhe von DM 3.600,– von einem bayrischen Amtsgericht verurteilt, weil er im großen Stil Raubkopien hochwertiger Computerprogramme für PC's getauscht und verkauft hatte.

7. Computerspionage
(§ 17 Gesetz gegen den unlauteren Wettbewerb, UWG)
Dieser Tatbestand bedroht das unbefugte Verschaffen oder Sichern eines Geschäfts- oder Betriebsgeheimnisses unter Einsatz von bestimmten technischen Mitteln sowie die unbefugte Verwertung eines so erlangten Geheimnisses mit Freiheitsstrafe bis zu drei Jahren, in besonders schweren Fällen bis zu fünf Jahren.
Der Versuch ist ebenfalls strafbar.

Hierdurch soll insbesondere verhindert werden, daß z.B. ein Betriebsangehöriger kurz vor seinem Ausscheiden aus dem Betrieb durch Übertragung oder Sicherung von Daten sich spezielle Kenntnisse, Verfahren oder Geheimnisse aneignet, um sie später selbst oder in einem anderen Betriebe zu verwerten.

Schlußbemerkung
Das zweite Gesetz zur Bekämpfung der Wirtschaftskriminalität hat die in vielen Bereichen der Computerwelt seit Jahren klaffenden Strafrechtslücken weitgehend geschlossen und die straffreien Mißbrauchsmöglichkeiten drastisch reduziert.

Ein jeglicher »Computerspezialist« tut gut daran, sich diese Vorschriften einzuprägen und sein Verhalten dementsprechend auszurichten.

COMPUTER

Makler war faul

Wer ein billiges Computerprogramm kauft, bekommt schnell die Kripo ins Haus.

Kristian Glaser ist – wie viele seiner Alters- und Klassenkameraden – ein Computerfreak. Deshalb bestellte der Fünfzehnjährige sofort, als er in einer Fachzeitschrift die Kleinanzeige las: „Makler ZX 81 zu verkaufen." Das elektronische Monopoly-Spiel sollte nur 15,50 Mark kosten. Daß es sich dabei um eine Raubkopie handeln könnte, müßte der Düsseldorfer Gymnasiast geahnt haben. Denn Computer-Programme kosten im Fachladen im Schnitt mindestens 50 Mark.

Aber jeder Pädagoge, jeder Jugendpsychologe weiß: Das Bewußtsein, Unrecht zu erkennen und zu meiden, ist bei Jugendlichen längst nicht so ausgeprägt wie bei erwachsenen Menschen. Und das Wort „Urheberrecht" sagt einem Jugendlichen kaum etwas. Im Gegenteil: Die heranwachsende Chip-Generation macht sogar einen Sport daraus, Codes zu knacken. Und Profis fertigen aus geknackten Programmen zahllose Raubkopien, die sie zum eigenen Nutzen zum Billigpreis verscherbeln. Fachleute schätzen, daß sie jährlich einen Schaden von mindestens 500 Millionen Mark anrichten.

Kristian Glaser: Hausdurchsuchung, weil ein Computer-Programm von einem Raubkopierer stammte

Zwei Jahre später, als der „Makler ZX 81" längst vergessen in des Schülers Schubladen lag, klingelte es bei seiner Mutter. Drei Männer begehrten barsch Einlaß und schwenkten einen Durchsuchungsbeschluß für „Wohn- und Geschäftsräume, einschließlich aller Nebengelasse."

Beate Glaser erfuhr: Ihr Sohn stehe im Verdacht „des Vergehens des Urhebergesetzes". Da werde doch wohl mit Kanonen auf Spatzen geschossen, meinte die Mutter.

Eben keine Lappalie. Denn: Der Software-Verkäufer war offenbar ein Raubkopierer. Als er aufflog, fand die Kripo in der Kunden-Kartei neben vielen anderen auch die Adresse des Düsseldorfer Schülers. Da solche Händler heute Abertausende von Raubkopien über Haupt- und Unterverteiler in Schülerkreisen vertreiben, packen die Fahnder ohne Vorwarnung zu.

44 Cassetten beschlagnahmten die Fahnder im Hause Glaser. Ein Strafverfahren wurde allerdings eingestellt. Der Schüler wurde lediglich verwarnt. Anwaltskosten für die Familie: 1165 Mark. Die Rechtsschutzversicherung übernahm keinen Pfennig. Denn ein Verstoß gegen Urheberrechte ist nicht versichert.

Kollege Blaumeise

Abenteuer in Business-Land

WIR SPAREN, koste es, was es wolle;
von Michael Amon

Wieder war einer jener launigen Montagmorgen angebrochen, die Kollege Blaumeise so haßte. Im Nebenzimmer rumorte bereits Kollege Buntspecht. Er benutzte die Tristesse einer eben ausgebrochenen, neuen Arbeitswoche immer, um die Mitarbeiter mit überfallsartigen Fragen in die Enge zu treiben und mit gezielten Organisationsanweisungen die Verwirrung zusätzlich zu erhöhen. Unter dem Motto »Wir erzeugen das Chaos nicht – wir sind es« schüttete er bereits früh am Morgen den Bürokaffee samt dem Kinde aus und trug so unwillentlich das Seine dazu bei, daß die Morgenmüdigkeit nur langsam und nicht etwa abrupt der Arbeitswut wich. Viel Bürokaffee gab's ja nicht mehr zum Ausleeren, seit Buntspecht die Putzfrauen eingespart hatte. Seit diesem unglücklichen Moment kochte niemand mehr Kaffee. Und da in der Zwischenzeit Bedenken aufgetreten waren, ob es unter Umständen nicht doch billiger wäre, die eben Gekündigten wieder einzustellen, als daß jeder Mitarbeiter unter Bergen von Staub sein eigenes Süppchen, pardon: Kaffeechen kochte, war die Morgenmüdigkeit auf unbestimmte Zeit prolongiert. Da außerdem die EDV-Abteilung das entsprechende Rentabilitätsprogramm erst zu schreiben begonnen hatte, war Buntspecht natürlich nicht in der Lage, konkrete Entscheidungen zu treffen. Darüberhinaus hatte der Chefprogrammierer bei der letzten Abteilungsleiter-Sitzung die Grundsatzfrage gestellt, ob es nicht überhaupt billiger sei, das gefragte Programm außer Haus erstellen zu lassen, als die eigenen Experten mit dieser Frage zu quälen. So brütete die Arbeitsgruppe »Morgenkaffee« darüber, ob man nicht die EDV-Abteilung auflösen und den fertigen Kaffee aus einem nahen Restaurant zuliefern lassen sollte.

Diese Infragestellung der eigenen Abteilung brachte die EDV auf Trab. Alle Schleusen wurde geöffnet, und wahre Datenströme ergossen sich über den nun ziemlich wehrlosen Buntspecht. Darüber hinaus hatte man dafür gesorgt, daß der neue Farbschirm von Buntspecht alle Zahlen, die die EDV betrafen, nur in schönstem Rosa zeigte, die Zahlen aller anderen Abteilungen jedoch in unheilverkündendem Rot. Ein besonderer Scherzvogel hatte das neue CD-Lesegerät mit einer Platte von Falco gefüttert und die Daten auf den Drucker geschickt. Der rapte jetzt seit Tagen und war ebensowenig zu beruhigen wie Buntspecht.

Im Moment aber war Kollege Buntspecht dem Papierverbrauch auf der Spur. Neueste Anweisung: die Papierstreifen der Elektronenrechner müssen auf beiden Seiten benutzt werden. Und was bei den Rechenstreifen möglich ist, muß auch beim Klopapier funktionieren! Außerdem ist das ja nur der erste Schritt! Danach werden die Toiletten samt Kanalgebühren gänzlich eingespart. Dezentralisation ist die Devise. Jedem Dienstnehmer sein eigener Nachttopf; Dies erfüllt kühn die Forderung der Gewerkschaften nach mehr Mitbestimmung am Arbeitsplatz und senkt gleichzeitig Kosten. Man könnte fast sagen: das Ei des Kolumbus, würde es in den Büroräumen nicht so faulig riechen.

Aber was wollen die Leute eigentlich? Regen sich auf über Autoabgase, und dann bekommen sie kostengünstig Landluft aus eigener, überschaubarer Erzeugung ins Zimmer, und es ist ihnen wieder nicht recht. Das Wichtigste bei einer radikalen Kostensenkung ist ein klares Feindbild. Das wußte Buntspecht, und er hatte eines: die EDV. Erstens verstand er von Computern nichts, und wenn er von etwas nichts verstand, dann konnte es auch nichts taugen. Und zweitens mochte er diese Kerle nicht, die immer in einer ihn seltsam anmutenden Fremdsprache miteinander sprachen. Wer versteht schon Sätze wie: »Zuerst habe ich das File gespoolt, es war nicht in der Cue, dann hab' ich es gecancelt, und danach kam der Head-Crash. Und zuletzt hat sich das ganze System aufgehängt.« Mögen sie sich alle aufhängen, dachte Buntspecht verzweifelt. Keiner verstand ihn, nicht einmal sein Terminal.

Am liebsten hätte Buntspecht die ganze EDV auf Sparflamme geschaltet, aber der Plunder brannte ja nicht. Was blieb ihm also anderes über, als böse Miene zum ebenso bösen Spiel zu machen. Alleine die Kosten, die die Herrschaften für die verschiedenen Seminare verschwendeten! Da waren in den letzten drei Monaten alle auf Kursen für eine Programmiersprache namens C gewesen. Wochenlang war die Abteilung leergefegt. Und als alle endlich wieder da waren und mit der Arbeit hätten beginnen können, was geschah da? Der Herr Abteilungsleiter kam und sagte: »Tut mir leid, lieber Buntspecht, aber der Markt hat sich geändert. Vergessen Sie C, Modula-2 ist jetzt gefragt. Aber machen Sie sich nichts draus, in längstens drei Monaten wird unser eingespieltes Team das beherrschen. Wir müßten nur das eine oder andere Seminar besuchen. Im Anschluß vielleicht ein paar Auslandsaufenthalte zum Erfahrungsaustausch mit anderen Firmen, und die Sache läuft."

Bescheiden stellte Buntspecht die Frage, wie das käme, habe man ihm doch vor einem Vierteljahr noch C als für die Zukunft des Unternehmens unverzichtbar dargestellt.

»Tja, da sehen Sie, daß die Zukunft auch nicht mehr das ist, was sie einmal war!« lautete die philosophisch unwiderlegbare Antwort.

Was Wunder, daß Buntspecht dauernd in laute Resignation und die Räume der EDV stetig in stille Leere verfielen.

Da also das Unmögliche nicht zu schaffen war, beschränkte Buntspecht sich auf das Mögliche, und machte sich damit unmöglich. Während Buntspecht deshalb schon zeitig am

Morgen mit dem Maßband von Toilette zu Toilette eilte, um den Morgendurchmesser der Klopapierrollen zu ermitteln (abzüglich des Abenddurchmessers läßt sich so der durchschnittliche Tagesverbrauch pro, äh, also, pro Kopf ermitteln), währenddessen saß Kollege Kater, ein echter Kater, vor dem Wasserglas und versuchte, den Goldfisch zu fangen. Was blieb ihm auch anderes übrig, seit Buntspecht ihm die tägliche Futterration gekürzt hatte. Dabei war er der Kater des Chefs, aber es gibt Menschen, dachte Kater, denen ist nichts heilig.

Blaumeise dagegen fürchtete um den Goldfisch, der Buntspecht wegen der hohen Fütterungskosten schon so lange im Magen lag, daß Buntspecht wünschte, er (der Goldfisch natürlich) würde in Katers Magen liegen. Wenn der Kater den Goldfisch erwischte, hatte Buntspecht den Goldfisch ebenfalls erwischt, denn ein neuer Goldfisch würde nie und nimmer genehmigt werden, sah doch das Budget so eine Ausgabe für heuer nicht vor. Andererseits würde eine eventuelle Zertrümmerung des Glases durch Kater einen erhöhten Aufwand für Reinigungskosten verursachen. Das waren die eigentlichen wirtschaftlichen Zusammenhänge. Buntspecht kämpfte mit sich selbst, und der Kostenrechner obsiegte über die menschlichen Triebe und verjagte Kater.

So kam Buntspecht in den durch nichts gerechtfertigten Ruf, ein Freund der Goldfische zu sein, fast ein Humanist also. Blaumeise aber zweifelte an sich selbst: Buntspecht, diese Mensch gewordene Einsparungsmaßnahme, ein Menschen- und Goldfischfreund? Undenkbar;

Weil er das nicht verstand, war Blaumeise halt nur ein kleiner Kollege und Buntspecht der grandiose Organisator. Blaumeise hatte eben keine Ahnung von den großen und komplizierten Zusammenhängen der Betriebswirtschaft.

Kollege Blaumeise

Abenteuer im Business-Land

von Michael Amon

»Die Zeit wird meßbar«

Natürlich war es nicht verwunderlich, daß Buntspecht - nun, da das ganze Haus endlich elektronisch geworden war - danach sann, wie man aus diesem an sich überflüssigen Tatbestand etwas Sinnvolles machen konnte.

Wie wäre es zum Beispiel mit einem ordentlichen Personal-Informations-System? Das ist nicht etwa das, was der naive, von Vorbildung ungetrübte Laie vielleicht glauben könnte. So ein System dient keineswegs dazu, das Personal besser über die Vorgänge in der Firma zu informieren. Weit gefehlt. Das genaue Gegenteil geschieht: die Firma informiert sich selbst über ihr Personal - zumindest im theoretischen Idealfall. Daß sich die Praxis von diesem Idealfall asymptotisch entfernt, sollte Buntspecht erst später erahnen.

(Eingeweihte wissen selbstverständlich längst, daß ein solches System nur dazu dient, alle Vorurteile, die über einen Mitarbeiter bestehen, statistisch und somit wissenschaftlich unwiderlegbar zu untermauern!)

Die Gleitzeit hatten sie ja schon vor einiger Zeit eingeführt. Seit damals hatten sie mechanische Stempeluhren und zwei getrennte Gehspuren beim Eingang. Letztere dienten dazu sicherzustellen, daß jene Mitarbeiter, die zu spät kamen, nicht mit jenen zusammenstießen, die zu früh gingen. Die mechanische Stempeluhr war natürlich so eine Sache. Jeden Monat waren drei Mitarbeiter der Personalabteilung damit beschäftigt, die gestempelten Zeiten zu addieren und vorher noch das meist unleserliche händische Zusatz-Gekritzel zu entziffern. Das alles würde man einsparen können, wenn man eine elektronische Zeiterfassung hatte, die direkt mit dem Computer gekoppelt war. Ein Knopfdruck am Monatsschluß, und die Anwesenheitszeiten wurden ausgedruckt und die drei Kartenaddierer eingespart. Buntspecht frohlockte, er würde heuer endlich sein Ziel erreichen und vom Chef zum Jahreswechsel das Goldene Sparschwein für die beste Kostensenkungsinnovation erhalten.

Das System wurde natürlich perfekt gehandhabt. Zugangskontrolle mit dem elektronisch lesbaren Personalausweis, am Monatsschluß automatische Übernahme aller Daten in die Gehaltsverrechnung. Und schon wieder zwei Dienstposten überflüssig gemacht. Das Goldene Sparschwein nahte.

Gewiß, es gab da gewisse Einführungsprobleme. Aber revolutionäre Neuerungen haben eben ihren Preis. Am ersten Tag sah dieser Preis so aus, daß die Mitarbeiter sich beim Eingang stauten, weil der Computer keinen hereinlassen wollte. Wer auch immer seine Karte ins Lesegerät schob, die Eingangstür öffnete sich nicht. Der Betriebsrat nützte diese spontane Massenbewegung natürlich gleich aus und erklärte die Ansammlung zum Streik, der Leiter der EDV erklärte sich für unzuständig, man möge sich bitte an Buntspecht wenden. Der aber hatte glücklicherweise verschlafen. Zuerst hatte er nämlich am Vorabend vor lauter Aufregung (es nahte schließlich sein großer Tag) nicht einschlafen können, und in der Nacht hatte er dann so schön vom Goldenen Sparschwein geträumt, daß er seinen rostigen Wecker glatt überhört hat. So kam es, daß Blaumeise sich ein Herz nahm, ein Fenster im Erdgeschoß ein- und sich selbst zur EDV durchschlug und zum Entsetzen aller Fachleute den Stecker zog. Der Computer stellte mangels Nahrung augenblicklich seine Tätigkeit ein, die Tür ging auf und die Leute hinein. So einfach war es, die moderne Technik und eine spontane Streikbewegung zu überlisten. Blaumeise war der Held des Tages.

Als Buntspecht endlich auftauchte, war der Spuk schon vorbei. (Erwähnt sei noch, daß am Abend keiner hinaus konnte, und Blaumeise wieder den Stecker zog.)

Der Monatsschluß nahte, und Buntspecht drückte erwartungsvoll das ominöse Knöpfchen (jenes, das die Daten in die EDV einspielt, wir erinnern uns). Das Lämpchen leuchtete, und alles war in Ordnung. Was nicht in Ordnung war, war der Umstand, daß die Personalabteilung inzwischen bereits soweit reduziert worden war, daß die Gehaltsüberweisung ohne weitere Kontrolle durchgeführt werden mußte. Es war nicht zum Schaden der Leute. Irgendwer hatte vergessen, eine Monats-Soll-Zeit einzugeben, welche daraufhin vom Programm mit Null angenommen wurde. Die Folge war, daß alle gearbeiteten Stunden als Überstunden honoriert worden sind. Seit damals leidet das Unternehmen unter gewissen Liquiditätsproblemen. Ganz im Gegensatz zu seinen Mitarbeitern: die sind solvent wie schon lange nicht.

Doch Buntspecht ließ nicht locker. Jetzt sollte auch noch die Kantine an das System angeschlossen werden. Jeder konnte mit seinem Ausweis einkaufen, die Zeche wurde einem am Monatsschluß automatisch vom Gehalt abgezogen. Die Mitarbeiter - aus Schaden wird man klug - zogen es am Einführungstag vor, ihre Nahrung selbst mitzubringen. Im ganzen Haus brutzelten die köstlichsten Gerichte und eine Hungersnot war gerade noch abgewendet worden. Nicht ganz: einer hungerte tapfer. Es war Buntspecht, der unverbesserliche Utopist, der voller Optimismus nur seinen Ausweis und viel Appetit mitgebracht hatte. Beides vergeblich.

Da jedoch im Verlauf der Zeit auch die größte Panne zur Gewohnheit wird, spielte sich alles ein, und eines Tages war es soweit, daß auch die Kantine klappte. Zwar durften die Daten nach einem Einspruch des Betriebsrates nicht gespeichert und damit auch nicht automatisch in der Gehaltsverrechnung verwendet werden, aber immerhin, die Listen gab es, und Blaumeise schaute hin und wieder hinein - meist bevor er zum Chef mußte. Den Listen konnte Blaumeise nämlich entnehmen, ob ER gerade auf Diät war oder nicht. War Ersteres der Fall, empfahl es sich, SEIN Zimmer nur in gebückter Stellung zu betreten, damit ER einen nicht sah, und einen SEIN Zorn nicht treffen konnte. Denn wenn ER auf Diät war, war ER immer zornig, dazu bedurfte es keines besonderen Grundes, die Diät war Grund genug. Nichteinmal Kater konnte dann auf Milde hoffen, denn er mußte jede Diät mitmachen. Und mit dem Selberfangen war das auch so eine Sache, denn seit der Einführung der Zutrittskontrolle kam nicht einmal eine Maus mehr ohne Ausweis ins Gebäude herein. So hatte das Leben als Chefkater auch seine Schattenseiten.

Besonders stolz war Buntspecht aber auf die vollelektronische Orientierungstafel, die er in seinem Büro aufhängen hat lassen. Auf dieser Tafel, die direkt mit dem Computer verbunden war, hatte jeder Mitarbeiter sein Lämpchen. Wenn ein Mitarbeiter kam und dem Computer als anwesend gemeldet wurde, leuchtete das Lämpchen auf. Ging er, verlosch es wieder. Die meisten Besucher von Buntspecht lobten dessen Kunstverstand, denn sie hielten das unruhig flackernde Objekt, das da an der Wand hing, für die Skulptur eines besonders avantgardistischen Künstlers, wer wagt da schon Kritik!

Wiedereinmal zeigte die Erfahrung, daß vorhergesagte Rationalisierungen nicht oft stattfinden. Längst schon waren die gefeuerten Mitarbeiter wieder eingestellt und noch einige dazu. Sie alle mußten jetzt das ziemlich störanfällige Personalsystem kontrollieren, das heißt, daß jetzt alles doppelt gemacht wurde. Die alten Stempeluhren hingen wieder an ihrem alten Platz, es wurde wieder addiert und gerechnet und manuell eingegeben. Und dann wurden die Ergebnisse verglichen, die Übereinstimmungen genau festgehalten, die Unterschiede zur EDV-erstellten Abrechnung untersucht und begründet (auf dem computerlesbaren Formular Nummer 144). Die zwei Gehspuren gibt es immer noch, sie werden jetzt aber auch vom Sportverein genutzt, der die Zeit, die jeder vom Aufzug bis zur Tür hinaus braucht, endlich elektronisch auf die hundertstel Sekunde genau messen kann. Am Ende des Jahres wirft der Computer eine Liste der Besten aus, und man kann den Mitarbeiter, der im abgelaufenen Jahr das Gebäude nach Büroschluß immer am schnellsten verlassen hat, mit dem Seidenen Zielband ehren.

Aus dem Goldenen Sparschwein für Buntspecht ist diesmal wieder nichts geworden. Er hat zu Silvester den Einarmigen Zirkel in Blech erhalten. Diese Auszeichnung bekommt jener Mitarbeiter, der mit dem größten Aufwand die geringste Wirkung erzielt. Dafür wurde Buntspecht neuerdings schon um sechs Uhr in der Früh gesehen, wie er vom Aufzug zur Eingangstür rennt, und wieder rennt, und wieder rennt. Ob er wohl ganz heimlich für das Seidene Zielband trainiert?

87

BDOS-Tip

Das CP/M 3.0 Betriebssystem sieht mehrere Möglichkeiten zu Reaktionen auf Fehler im Zusammenhang mit den Diskettenstationen vor.

Normalerweise wird das laufende Programm, z.B. ein Turbo-Pascal-Programm, bei einem Diskettenfehler unterbrochen, selbst wenn es sich nur um das Fehlen einer gesuchten Datei handelt. Ein leeres oder nicht existierendes Laufwerk anzusprechen, hat da schon zu manchem Datenverlust oder sogar zum Turbo-Systemabsturz geführt.

Schade um das aktuelle »Workfile« oder das geöffnete Datenfile, welches dann nur noch aus Bruchstücken besteht.

Über die BDOS-Funktion 45 («Set BDOS Error-Mode«) läßt sich dies, auch von einem Pascal-Programm heraus, geschickt vermeiden.

Assembler-Programme laden das C-Register mit dem Wert 45 (dez.), das E-Register mit 255 (dez.) und rufen die BDOS-Bearbeitungsroutine auf. Turbo-Pascal-Programme begnügen sich mit dem Prozedur-Aufruf **BDOS(45,255)** – und schon wird der BDOS-Error-Mode von »default« (Programm stoppen und Fehler anzeigen) in »return« (gebe Fehler-Nr. im Register H zurück, schreibe in Register A eine 255 und setze Programm fort) umgeschaltet.

Eine Zurückschaltung in den »default«-Mode ist mit **BDOS(45,0)** möglich. Ein Aufruf von BDOS (45,254) führt zum »display and return«-Modus: Es wird eine Fehlermeldung anzeigt und das Programm fortgesetzt.

Leider kann ein Turbo-Pascal-Programm diese CP/M-Fehlermeldung nicht über IORESULT anzeigen, da IORESULT nur die Fehlernummer aus dem A-Register zurückgibt. Trotzdem ist aber die Fehlererkennung durch den Wert 255 gegeben.

Eine Beschreibung dieser BDOS-Funktion, seiner Fehlermeldung und weitere BDOS-Funktionen finden sich im »Digital Research CP/M Plus Programmer's Guide«, welches zum CP/M 3.0 System mitgeliefert wird.

(Günther Prühs)

COMPAREX hat Premiere

Erster Messeauftritt der BASF/Siemens-Gesellschaft

(Dö) Die COMPAREX Informationssysteme GmbH – das zum Jahresbeginn 1987 in Mannheim gegründete Unternehmen von BASF und Siemens – präsentierte auf der CeBIT '87 neue Großcomputermodelle. Für den Kunden liegt der Nutzen der neuen Typen vor allem in der hochentwickelten Technologie.

Der technologische Vorsprung wird bei einem direkten Vergleich mit leistungsgleichen führenden Wettbewerbsmodellen deutlich: Die Leistungsaufnahme der COMPAREX-Maschinen ist wesentlich geringer. Der Bedarf an Stellfläche beträgt nur einen Bruchteil. Noch drastischer werden für den Anwender die Vorteile beim Vergleich des auf dem Doppelboden des Rechenzentrums lastenden Gewichtes deutlich. Die nur 840 kg leichte PCM-Maschine ist problemlos zu installieren.

Als zweite Produktneuheit zeigte die COMPAREX Informationssysteme GmbH auf der CeBIT ihr neues 3480-kompatibles Bandsystem im Betrieb. Das System arbeitet mit 1/2"-Chromdioxid-Computerband-Kassetten und hat besonders hohe Datenübertragungsraten.

Bereits im Januar hatte die COMPAREX ihre Angebotspalette an Großrechnern um das neue Modell 7/90-3 erweitert. Sie deckt somit mit nur einer Rechnerfamilie die Leistungsspanne von der 17-Mips- bis zur 70-Mips-Klasse ab. (Mips bedeutet Millionen Rechenoperationen pro Sekunde.)

BASF mit neuer Diskette

88

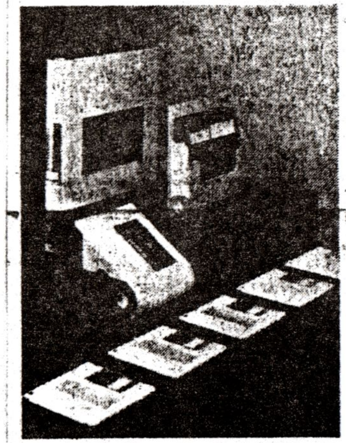
CeBIT-Messe in Hannover: Verbraucher profitiert von technischer Entwicklung

(Dö) »Der Markt wächst – das Qualitätsniveau steigt – der Wettbewerb wird härter.« So charakterisierte der Leiter der BASF-Datentechnik, Dr. Karl Uhl, die Situation auf dem Markt für Disketten. Gleichzeitig presche die BASF auf dem Weg zu höheren Speicherkapazitäten erfolgreich voran. Als Neuigkeit auf dem CeBIT 1987 (Centrum Büro-Informations-Technik) kündigte er eine 3.5"-Diskette mit einer auf 2 Megabytes verdoppelten Speicherkapazität an. CeBIT, die größte europäische EDV-Fachmesse, fand vom 4. bis 11. März in Hannover statt.

Beachtliche Speicherkapazität hat eine handliche Datenkassette mit Chromdioxid-Band. Sie ist etwa so groß wie zwei nebeneinanderliegende Zigarettenschachteln und vermag im Lauf ihrer Weiterentwicklung die kaum vorstellbare Datenmenge von 2 Gigabytes zu speichern. Die heute genutzte Kapazität liegt bei 200 Megabytes. Dabei handelt es sich um eine Weiterentwicklung der von BASF produzierten Datenkassette für die Groß-EDV.

2 GB Daten entsprechen dem Inhalt von 5 600 herkömmlichen Disketten von je 360 KB Kapazität oder umgerechnet von zwei Millionen einseitiger Geschäftsbriefe. Diese Korrespondenz würde immerhin 3 500 Aktenordner mit rund 10 Tonnen Gewicht füllen. Die Magnetbandkassette dagegen ist 220 Gramm leicht.

Mangels entsprechender Laufwerke wird es die 2-GB-Kassette zwar in naher Zukunft noch nicht zu kaufen geben, aber die BASF will an diesem Beispiel zeigen, daß die Entwicklungsmöglichkeiten der magnetischen Datenaufzeichnung noch lange nicht erschöpft sind.



Zur CeBIT '87 ergänzt die BASF-Datentechnik ihr Diskettensortiment um die FlexyDisk® 3.5"-2HD mit 2 Megabytes unformatierter Speicherkapazität und um eine Archivbox für 3.5"-Disketten.
Foto: BASF

Optische Platten langsam im Kommen

Die optische Speicherplatte wird nach Ansicht der BASF-Datentechnik keine Ablösung der magnetischen Speichertechnologie bringen. Sie ist vielmehr eine Ergänzung in der Vielfalt der Speichertechnologien. Die BASF entwickelt in ihren Ludwigshafener Forschungslabors sowohl einmal beschreibbare optische Platten mit organischen Farbstoffschichten als Datenträger als auch beliebig oft beschreibbare magneto-optische Platten, die mit Metallegierungen arbeiten.

Disketten bleiben das preiswerteste Speichermedium für Klein- und Personal-Computer. Die optischen Speicherplatten und die dazugehörigen Laufwerke sind noch viel zu teuer, um Disketten als Ablagespeicher abzulösen.

Geschäftsverlauf BASF-Datentechnik

Nach Einbringung des Geschäftes mit kompatiblen Rechnersystemen in die COMPAREX konzentriert sich das Geschäft mit EDV-Produkten der BASF auf Forschung/Entwicklung, Produktion und Vertrieb von Speichermedien. Im Geschäftsjahr 1986 betrug der Weltumsatz mit Speichermedien an Dritte ca. 410 Millionen DM. Darin sind ca. 220 Millionen DM in Europa getätigt worden und ca. 78 Millionen US-Dollar in Nordamerika. Während der Umsatzzuwachs in Europa ca. 9% beträgt, ergibt sich währungsbedingt in Nordamerika ein Rückgang um 7%, der in Landeswährung einer Steigerung von 25% gleichkommt.

Am stärksten wuchs ihr Umsatz bei den Computerband-Kassetten für 3480-kompatible Bandlaufwerke, die sie vorwiegend an Rechenzentren verkauft.

Mit einem Investitionsaufwand von 70 Millionen DM hatte die BASF in 1985 und '86 in ihrem Willstätter Werk eine hochautomatisierte Fertigungsstraße mit einer Jahreskapazität von über 10 Millionen Kassetten aufgebaut.

Insgesamt investiert die BASF in den Jahren 1986 und 1987 rund 100 Millionen DM in die weitere Automatisierung der Diskettenfertigung und der Qualitätskontrolle einzelner Herstellungsschritte.

Auf der CeBIT in Hannover:

BASF-Siemens-Gemeinschaftsunternehmen greift IBM an

Comparex rechnet sich gute Chancen aus

(eingesandt von K.J. Mühlenbein, BASF)

Von unserem Redaktionsmitglied Dieter Keller

Weniger Verkäufe als Kontakte erwartet die Comparex Informationssysteme GmbH mit Sitz in Mannheim von der bevorstehenden Cebit-Messe in Hannover. Hier präsentiert sich das Gemeinschaftsunternehmen des Ludwigshafener Chemiekonzerns BASF und des Münchner Elektronikunternehmens Siemens erstmals der Öffentlichkeit, nachdem es zum Jahresanfang offiziell die Arbeit aufgenommen hat. Neu vorgestellt werden sollen ein neuer Großrechner sowie Peripheriegeräte.

Auf der menschlichen Seite hat der Start der Comparex weniger Probleme als erwartet gebracht, berichtet BASF-Sprecher Horst Dönicke. Von den rund 1000 Beschäftigten kamen etwa 200 von Siemens, der Rest von der BASF. Zusammengefaßt haben beide unter dem neuen Dach ihr Geschäft mit kompatiblen Computer-Systemen. Die BASF hatte sich seit 17 Jahren auf diesem Markt getummelt und Geräte der japanischen Firma Hitachi unter eigenem Namen vertrieben. Siemens hatte kompatible Rechner verkauft, die in Japan von Fujitsu bezogen wurden; dies als Abrundung zum Geschäft mit Geräten, die mit dem Siemens-eigenen Betriebssystem BS 2000 arbeiten, das die Münchner behalten haben.

In der Computerbranche ist am US-Riesen IBM nicht vorbeizukommen. Die Zahlen zeigen es deutlich: 1985 wurden in Europa Großrechner für 30 Mrd. DM verkauft. In diese Kategorie fallen Rechnersysteme, die mindestens eine halbe Million DM kosten, also weder Personal Computer noch mittlere Datenanlagen, wie sie als Schwerpunkt etwa Nixdorf verkauft. Mit Großcomputern bewältigen Unternehmen ihre Buchhaltung, schreiben Versicherungen ihre Rechnungen, erfassen Zeitungen Text und Anzeigen für den Fotosatz. IBM erreichte 1985 alleine 16 Mrd. Umsatz DM.

COMPAREX
Ein Unternehmen von BASF und Siemens

IBM-kompatible Rechner wurden 1985 für rund 2,5 Mrd. DM verkauft; Tendenz: steigend. Auf diesem Markt tummelt sich die Comparex. Ihre beiden Mütter als Vorgänger hatten es 1986 mit kompatiblen Rechnern auf knapp 1 Mrd. DM Umsatz gebracht, davon alleine die BASF auf 720 Mio. DM, was einen Umsatzsprung um 30 v. H. bedeutete. 1987 peilt die Comparex gut 1 Mrd. DM an. Sie will schwarze Zahlen schreiben, und das trotz gewaltigen Preisverfalls.

Wer sich zum ersten Mal einen Großrechner anschafft, greift meist zu IBM. Schließlich will man auf Nummer Sicher gehen. Die Comparex kommt dann häufig mit zusätzlichen Peripheriegeräten ins Geschäft und schließlich auch beim Zentralrechner, wenn der zu klein wird. Gilt doch als Faustformel, daß der Leistungsbedarf eines Rechenzentrums jährlich um 40 v. H. steigt. Wettbewerbsargument ist dann der Preis, der rund 15 v. H. unter dem von IBM liegt, oder die größere Leistungsfähigkeit. Dabei kommt zugute, daß die Japaner in der technologischen Entwicklung um 2 Chip-Generationen oder 2 bis 3 Jahre der IBM voraus sind. Ihre Rechner bieten häufig die gleiche Leistung auf erheblich weniger Platz – wichtiges Argument bei zentnerschweren Geräten, für die Fußböden verstärkt und aufwendige Klimaanlagen eingebaut werden müssen.

Die neu verkauften Geräte mit dem Comparex-Schild kommen praktisch alle von Hitachi, nachdem Fujitsu Lizenzstreitigkeiten mit IBM hatte. Im Jahr kaufen die Mannheimer für rund 500 Mio. DM in Japan ein. Dabei vertreibt die Hitachi-Rechner in Europa auch NAS. Vorteil von Comparex: das viel dichtere Servicenetz, ein sehr wichtiges Argument beim Kauf. Steht der Computer, muß binnen kürzester Zeit ein Servicetechniker her, sonst wird es teuer. Dabei haben sich Siemens und BASF ideal ergänzt: Die Münchner waren beim Service im Inland besonders stark, die Ludwigshafener im Ausland.

Die Zentrale der Comparex mit rund 250 Beschäftigten sitzt in Mannheim, wo vorher schon die BASF-Datentechnik residierte. In Ludwigshafen stehen Ersatzteil- und Auslieferungslager. Hier gibt es auch einen Notdienst rund um die Uhr, eilige Ersatzteile fliegt in dringenden Fällen Air Pegasus von Mannheim in alle Teile Europas.

Die Kunden haben, so berichtet Dönicke, sehr positiv auf den Zusammenschluß wie den neuen Namen reagiert. Ein starker Konkurrent zur IBM ist gut für den Wettbewerb. Bei der Wahl des Namens wurde bedacht, für die Kooperation mit weiteren Unternehmen offen zu sein, auch wenn es hier noch keinerlei Pläne gibt. Außerdem mußte er in allen europäischen Ländern gut klingen; hat die Comparex doch in 9 Ländern eigene Töchter und in 5 weiteren Vertreter. Im Namen versteckt ist der Computer, aber auch das lateinische Wörtchen „compar", zu deutsch gleichwertig. Eine ganz bewußte Anspielung auf IBM.

Das Stichwort:

Kompatible Computer

Computer bestehen aus vielen Einzelgeräten, dem Zentralrechner und sogenannten Peripheriegeräten wie Bandlaufwerken zum Speichern von Daten, Druckern oder Bildschirmen. Damit sich alle Geräte eines Systems untereinander vertragen, müssen sie eine einheitliche Sprache verstehen. Dies ist die Aufgabe des Betriebssystems. Was der Computer konkret machen soll, legt dann das Anwendungsprogramm fest, etwa Buchhaltung, Textverarbeitung oder Produktionssteuerung. Peripheriegeräte, die für ein bestimmtes Betriebssystem konzipiert sind, sind stecker-kompatibel, es können also per Stecker die Geräte ganz verschiedener Hersteller verbunden werden. IBM hat mit seinem Betriebssystem weltweit Normen gesetzt. Andere Hersteller haben nur die Wahl: Entweder sie richten sich hiernach und bauen kompatible Geräte, oder sie setzen bewußt auf ein eigenes Betriebssystem, wie etwa Siemens mit BS 2000. Entscheidet sich ein Anwender für ein bestimmtes Betriebssystem, so hat er sich für nicht absehbare Zeit festgelegt. Denn seine Anwenderprogramme, deren Entwicklung häufig Millionen kostet, laufen nur mit diesem Betriebssystem. dik

Être au violon.
In der Geige sein.
Hinter schwedischen Gardinen sitzen.

Rainer Rübsteck, Bad Neuenahr

Der Tag der offenen Tür

Ein Telefongespräch abgehört von Paul Antar

Guten Tag! Firma COMCOM, Terminal achthundertneunundzwanzigeinhalb. Sie rufen bestimmt wegen dem »Tag der Offenen Tür« an. Morgen geht's los, pünktlich um neun Uhr, Halle zweihundertfünfundvierzig dreitausendfünfhundert Computer warten nur darauf gekauft zu werden. —Was? Eine Unverschämtheit? — Mann, jetzt hören Sie einmal ganz ruhig zu. Sie haben morgen die Möglichkeit, Computeranlagen zu besichtigen, Sie müssen keinen Computer kaufen! Beileibe nicht, wir tun Ihnen gar nichts. Aber warum nutzen Sie nicht die Chance, am Computer kommt man heutzutage sowieso nicht mehr vorbei, die halbe Menschheit lebt schon irgendwie von dem Zeugs. Wenn die Dinger mal streiken sollten, ist's zappenduster. Dann geht nichts mehr, dann sitzen wir alle auf der Straße. —Sie können das Wort Computer nicht mehr hören?— Nun hören Sie mal gut zu und auf den Rat eines alten Hasen, der fast alles geknackt hat, was es zu knacken gab: Das mit dem Computer kommt sowieso, das ist schlimmer als AIDS. Die Leute stecken sich an irgendeinem Monitor an und dann haben sie es weg. Die alten Atombomben konnte man wenigstens noch abschaffen, aber Computer? Unmöglich, die kann man nicht mehr abschaffen! Der Vorschulunterricht beginnt ja jetzt Gott sei Dank mit Informatik, aber an manchen Grundschulen wird ja leider immer noch MS-DOS 12.0 gelernt. Ein echtes Relikt. So was sollte man abschaffen. Die Pimpfe installieren heutzutage schon im Kindergarten Festplattenrechner, und da wollen Sie sich noch nicht mal einen PC anschauen? — So, so, Sie können die Dinger nicht mehr sehen! —Soll's ja auch geben, die Verweigerer. Grundgesetz – Meinungsfreiheit – freie Entfaltung – Individualität – nicht so wie bei den alten Sozies. Ich will Ihnen mal etwas sagen: Da geht echt was ab mit den Computern. Im Guiness-Buch der Rekorde steht Dauercomputing mit achtzehn Jahren, drei Monaten und ein paar Zerquetschten und den Rekord im Kopieren unbekannter Datenträger hält der ALLOS III super mit achtundzwanzig Millisekunden! Und Sie riskieren noch nicht einmal einen Blick auf einen Handheld, vierzig Mega, alles drin, Kugelschreibergehäuse, achtundvierzig Tools gratis, zum Einführungspreis von neun Monetas. Hören Sie noch zu? Was machen Sie denn beruflich, vielleicht ließe sich da so ein Ding unterbringen. Oder schreiben Sie doch Bücher, wenn Sie keine Computer sehen können. Wir haben da einen Minitext im Angebot, den legen Sie sich ganz einfach unter die Zunge und geben ihn dann beim Verlag ab. — Manager sind Sie von Beruf, und da haben Sie keine Kopfhörer-Workstation? Sie wollen mich doch nicht auf den Arm nehmen? — Ach so! Pleite gegangen! Passiert jedem mal, ist halb so schlimm. Bald werden Sie sich bestimmt wieder einen kleinen YT leisten können! Haben Sie schon gehört, gestern haben sie endlich den Flugverkehr in Frankfurt eingestellt. Es läuft ja jetzt alles sowieso per DFÜ im interkontinentalen Netzwerk. Eine tolle Sache! Damit können Sie Zuhause den Computer jahrelang alles erledigen lassen und Sie liegen in Rio am Pool, oder computern halt. — Ob ich weiß, von woher Sie anrufen? – Keine Ahnung, da muß ich raten. Vom Mond aus vielleicht, oder aus einer Telefonzelle am dritten Marsmeer. — Alles falsch? Aus Köln! — Ei, da sind wir doch auch! Hier steht unser zentraler Computer, da schlägt das Herz unserer Firma! Hier begraben wir auch unsere überalterten Prozessoren und vergessenen Betriebssysteme. Also neulich, da haben sie hier ganz in der Nähe, in Hamburg, einen irren archäologischen Fund gemacht. Eine echte IBM 68000 mit zwei kleinen Laserdruckern und sehr gut erhalten. Jetzt müssen im Europäischen Museum die anderen Kisten etwas zusammenrücken, oder die bauen gleich neu an. Und jede Menge Disketten, ganz alte Dinger, mit so einem großen Loch in der Mitte, hoffentlich lassen die sich auch entziffern. Müßte aber gehen, das ist ja damals ganz simpel gemacht worden – so eine Art magnetischer Keilschrift oder so. Sind Sie noch dran? Weshalb rufen Sie eigentlich an? Museum – Computer – Pleite – Manager – Computer – Schule – Tag der Offenen Tür – jetzt haben wir's! — Ob ich da eingeteilt bin? — Natürlich! Ab acht Uhr hänge ich am Keybord und bin online! — Zu Hause bleiben? Alles Essig? Tut Ihnen leid, von wegen dem Rausschmiß. So eine Pleite kann eben jedem mal passieren, habe ich selber gesagt. Wird schon nicht so schlimm werden. In spätestens einem halben Jahr kann ich mir wieder einen Kompatiblen leisten. Der Finanz-Host hat uns einen Kuckuck ins ROM geblasen. Feierabend! Da kann man nichts machen. Aber da läuft noch eine alte Sache mit der Super-Soft aus Honolulu. Die wird die Rechner zum Glühen bringen und dann kommen wir wieder ganz groß raus. Vertriebsrechte im indischen Meer in Aussicht genommen. Verkauf über Intersoft-International, Kasse über CARD. Acht Tage installieren und dann können wir die Füße wieder hochlegen. Alles klar Boß, bin wieder dabei! — Was ich jetzt mache, ja weiter studieren natürlich! Die angebrochene Doktorarbeit in Informatik über den »Zyklus des Pipelineing agressiver Bits im Controller-Bus« fertigmachen und dann mal weitersehen. Vielleicht reizt mich ein Abstecher zum In-Ohr-Publishing. Alte Adresse. Sie melden sich wieder. Alles klar! Tschüüüüüüss. — — Zentrale! – Ich hau gleich ab, alles klar für offline? Stopbit ist auf total gesetzt. CARD auf pleite. Parameter auf END. Ich installiere jetzt Go Home for ever und eröffne das Protokoll. Alles rüber? — Was fehlt noch? — Klamotten ausziehen, Schuhe dalassen, Paßwort vergessen und Hände waschen – wegen der Chips unter den Fingernägeln. O.K. wird gemacht. — Arrivederci schöne neue Welt!

Programming with Turbo Pascal

David W.Carroll

Micro Text Productions, Inc., McGraw Hill Inc., New York
Paperback mit 310 Seiten und Übungsdiskette.
ISBN 0-07-852909-3

Seit über einem Jahr besitzt dieses Buch, ein Werk aus dem Verlag der US-amerikanischen Zeitschrift »Byte«, seinen Stammplatz auf meinem Schreibtisch. Obwohl es sich hierbei nicht um eine umfassende Abhandlung à la Wirth, Duntemann oder Koffmann handelt, greife ich zu diesem Buch und nicht zu einem »Klassiker«, wenn ich schnell etwas prüfen will.

Der Autor David Carroll ist seit zwanzig Jahren im Computer-Bereich tätig und in den USA für seine Beiträge in Dr. Dobbs Journal, Micro/Systems Journal (Turbo-Pascal Corner) und Business Software News bekannt.

Das Buch an sich besteht aus achtzehn stramm mit Information gefüllten Kapiteln. Die ersten drei beschäftigen sich mit allgemeinen Fragen der Installation, des Editors und einer Introduction to Computer Programming, wobei die völlige Ignorierung von Diagrammen zu bedauern ist. In Kapitel IV bekundet der Autor seinen Glauben an die BNF Notation, einer Technik, die konsequent bis zur letzten Buchseite durchgehalten wird.

Durch eingeschobene Kapitel mit vier einfachen Programmen (u.a. ein Mittelwertprogramm und ein Programm, das Quadrat und Quadratwurzel berechnet) teilt sich der gesamte Stoff in zwei etwa gleich große Hälften. Die erste behandelt Turbo-Pascal bis zu den Procedures, Functions und einfachen I/O-Anweisungen, während die zweite Hälfte gleich mit Entscheidungs-Anweisungen (Decision Statements) beginnt, Pointer und dynamische Strukturen folgen läßt und mit einer relativ kurzen »Reference Information« über das Machine Level Interface abschließt. Als interessante Beispiele zeigt er Programme, die Directoryangaben zum Ziel haben und die Systemuhr als Event-Timer benutzen.

Die Betonung des Buches liegt insgesamt auf nichttechnischem Gebiet. Die mathematischen Anwendungen beschränken sich auf zwei Programme zur Matrix-Manipulation. Dem Autor ist nach seinen eigenen Worten daran gelegen, dem Leser die wesentlichen Elemente des Standard Pascals beizubringen. und nicht einen allumfassenden Text zu schreiben. Dies ist ihm nach meinem Dafürhalten auch sehr gut gelungen.

Auf der beigefügten Diskette sind an die achtzig Programme abgelegt, so daß der »Turbo Neuling« ein ungewöhnlich breites Spektrum an Übungsprogrammen genießen kann. Ein besonderer Leckerbissen sind drei Utility-Programme TURBOHLP, TURBOXRF und TURBOLST, wobei ersterer ein speicherresidenter Online Help ist. Leider gibt es kein Inhalts-verzeichnis der Programme, so daß der Leser erst über Programmnamen wie MILEAGE.PAS rätseln muß.

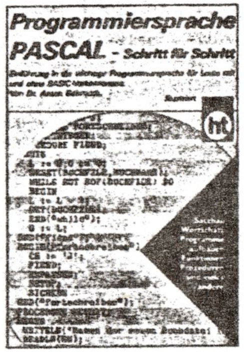
Das Inhaltsverzeichnis des Buches ist mit ganzen zweieinhalb Seiten Umfang beiweitem nicht ausreichend. Dafür sind die Appendices 12 Stück an der Zahl sehr zu begrüßen. Eine Liste der Keyboard Return Codes für den IBM-PC ist hier zu finden, sowie die vollständige Auflistung der Turbo-Pascal Syntax in Backus-Naur-Form. Wer die Übungdiskette nicht mehr braucht. hat mit diesem Buch einen sehr nützlichen ja nahezu unentlichen Begleiter als Quick Reference zu Turbo-Pascal. Der Sprachgebrauch des Buches ist ohne jeglichen Slang, so daß auch Leser mit »nur« Computer-Englisch keine Schwierigkeiten haben werden.

(Isabel Saplac)

Programmiersprache PASCAL - Schritt für Schritt

von Anton Schmuck

Humboldt-Taschenbuchverlag, München, 1986
Reihe ht-ratgeber Nr. 551
159 Seiten
ISBN 3-581-66551-4
Preis: 8,80 DM



Ein Text für den kleinen Geldbeutel

Der Autor wendet sich einerseits an Computerneulinge, und andererseits schlägt er auch die Brücke von BASIC zu Pascal. Im ersten Drittel des Buches werden die Grundlagen des Programmierens besprochen. Danach werden die grundlegenden Sprachelemente von Pascal vorgestellt und an vielen kleinen Beispielen erläutert: Der Aufbau einfacher Programme, die einfachen Datentypen und Strings, Operatoren und Standardfunktionen, Unterprogramme einschließlich knapper Hinweise zu Rekursion, Blockstrukturierung, strukturierte Datentypen, Array, Menge und Verbund. Auch der Datentyp Datei (File) wird am Ende des Buches anhand eines Beispielprogramms erwähnt, jedoch nicht ausführlich im Textteil berücksichtigt.

Die Darstellung ist an UCSD-Pascal orientiert. Turbo-Pascal wird nur einmal kurz erwähnt, und einige der Programme laufen unter Turbo-Pascal auch nicht wie gewünscht. Man merkt dem Buch an, daß sich der Autor selbst von BASIC ausgehend in die Thematik von Pascal, genauer UCSD-Pascal, eingearbeitet hat, was gar kein Nachteil ist, sondern ein Vorteil für alle diejenigen Leser, die diesen Weg von BASIC zu Pascal ebenfalls gehen. In den Text eingeschoben sind zahlreiche Übungen und Anregungen zu eigenen Lernschritten, was für den Anfänger sehr hilfreich ist. Die Sprache des Textes ist zumeist sehr gut verständlich, nur finden sich (für meinen Geschmack) gelegentlich zu viele Hinweise auf das Englische. Beispiel: »Die CASE-Anweisung ist das zweite bedingte Statement.« Aber diese Stilfrage ist eine Nebensache. Hauptsache ist, daß hier solide Information für den Übergang von BASIC zum Programmieren in Pascal zu finden sind. Natürlich weist jedes Buch in der 1. Auflage den einen oder anderen (Druck-)Fehler auf, doch halten sich die Fehler in Dr. Schmucks Buch in Grenzen.

Ich kann dieses Buch jedem Anfänger in Pascal empfehlen, denn es ist auch wegen des erfreulich niedrigen Preises eine gute »Einstiegslektüre« für Interessenten, die sich erst noch für eine Hochsprache entscheiden wollen. Apropos niedriger Preis: es gibt zahlreiche Bücher, die für den dreifachen bis fünffachen Preis dieses Bandes

Sehr geehrte CLUB80-Mitglieder,

wie am 14.03.87 beim Clubtreffen in Alsfeld vereinbart, übernehme ich ab sofort die umfangreiche CLUB80 Programmbibliothek. Ganz kurz zu meiner Person, bin 29 Jahre alt, Diplomverwaltungswirt und seit 8 Jahren als Organisationsprogrammierer bei der Stadtverwaltung Schweinfurt tätig. Den TRS-80 M I (64k, RS232, Akkustikkoppler, 2 x 80 Track- und 1 x 40 Track-Laufwerke, Epson RX80, 80-Zeichenkarte) benutzte ich in den letzten Jahren überwiegend für die berufliche Weiterbildung (Schwerpunkt Programmiersprachen, Dateiverwaltung, Datenbanken, gute Dienstprogramme).

Nun ein paar Punkte zur Programmweitergabe:

- Kajot hat mir versichert, daß alle CLUB80-Programme frei von Rechten Dritter sind (also kein Copyright vorhanden ist). Urheberrechtlich geschützte Programme werden von mir nicht angenommen (der Einsender muss also mitteilen, ob er selbst der Autor ist, oder woher er das Programm hat). Ich hoffe Ihr habt dafür Verständnis, denn ich muß schließlich dafür gerade stehen.

- Der Programmautor sollte sein Meisterwerk nur dokumentiert einsenden, d.h. dem Programm sollte eine Beschreibung in Form einer Textdatei (mit einem gängigen Textverarbeitungsprogramm erstellt) beiliegen. Mein Vorschlag wäre die Datei mit der File-Extension /DOC zu benennen. Bei der Textdatei sollte sich der Autor auf den normalen Zeichensatz beschränken, damit kein spezielles Textverarbeitungsprogramm zum Ausdrucken benötigt wird. Gleichzeitig ist es sinnvoll einen Beschreibungsvorschlag für die CLUB80-Bibliothek zu machen.

- Da es sich in der Regel um ganze Programmpakete handeln wird, wäre es sinnvoll ebenso eine Inclusion List File (kurz ILF) zu erstellen. Das hat den Sinn, daß man solche Pakete mit einem Copy-Befehl (z.B. COPY,0,1,,CBF,ILF=dateiname/ILF) kopieren kann oder anhand dieses Files genau weiß welche Programme zu einem Verfahren gehören. ILF-Files erstellt man mit einem Textverarbeitungsprogramm. Die zusammengehörenden Programme sind einzugeben und mit X'OD' = <RETURN> zu trennen. Die ILF-Datei wird als ASCII-File gespeichert (z.B. in SCRIPSIT mit S,A dateiname/ILF)

- Kopieraufträge sollten nur schriftlich bei mir eingereicht werden, denn telefonisch bin ich nicht immer erreichbar. Laut Kajot übernimmt der CLUB80 traditionell die Portokosten für die Software-Anforderungen, ich bin jedoch der Meinung, daß das Rückporto beigelegt werden sollte, Ihr erspart mir damit sicherlich eine Menge Zeit und der Clubkasse tut dies bestimmt auch ganz gut. Verzögerungen bei der Bearbeitung können entstehen, denn ich muß mich zuerst in die umfangreiche Bibliothek einarbeiten - habt also etwas Geduld.

- Da das Kopieren ganzer Disketten oft zeitsparender ist als das Zusammensuchen auf verschiedenen Datenträgern, wäre es mir ganz recht, wenn Ihr genügend Disketten beilegt.

- Jetzt kommt eigentlich das Wichtigste für Programmeinsender und Anfragende - das Diskettenformat. Wie eingangs erwähnt kann ich 40 und 80 Track-Disketten erstellen. Hier mein Vorschlag für's Format:

80-Track, DS, DD
Pdrive-Angaben:
TI=CK,TD=G,TC=79,SPT=36,TSR=0,GPL=8,DDSL=17,DDGA=6

40-Track, SS, DD
Pdrive-Angaben:
TI=CK,TD=E,TC=39,SPT=18,TSR=0,GPL=2,DDSL=17,DDGA=2

Falls es damit Probleme geben sollte, so macht mir in einem INFO bessere Vorschläge, eine Einigung ist in jedem Fall möglich. Mein ganz persönlicher Wunsch wäre es, wenn die Disketten bereits in einem dieser Formate formatiert sind und ein System (eventuell auch Minimalsystem) enthalten. Als Basis für die Weitergabe von CLUB80-Programmen wäre NEWDOS80 geeignet (das kennt jeder).

Das war's zum Wechsel der Programmbibliothek. Sicherlich sind einige Anregungen und Wünsche bereits in älteren Infos bereits angesprochen worden. Ich wollte hier nicht bekannte Informationen nochmals wiederholen, sondern hatte einfach noch keine Zeit sämtliche Infos zu lesen - vielen Dank für Eure Aufmerksamkeit

Werner Förster

– IMPRESSUM –

1. Vorsitzende

Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

2. Vorsitzende

Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

Hardwarekoordinator

Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar

Werner FÖRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion

Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren

Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck

Peter Spieß
Trugenhoferstraße 27
8859 Rennertshofen 1
☎ 08434 /454

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Bankverbindung des CLUB 80

Postgirokonto Peter STEVENS
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

SCHLUSS

Hallo Club-80er,

mit einem Monat Verspätung ist es mir nun endlich doch gelungen, das neueste Clubinfo auf die Beine zu stellen. Dies lag diesmal nicht an den fehlenden Beiträgen, obwohl auch jetzt noch einige "versprochene Werke" fehlen, sondern mehr daran, daß ich im Moment privat und beruflich im Dauerstreß lebe.
Für die verspätete INFO-Ausgabe möchte ich mich bei Euch entschuldigen. Ich hoffe, daß sich das nicht wiederholen wird. Leider ist aber ein Wiederholungsfall nicht ganz ausgeschlossen, da es in diesem Jahr bei mir noch weiterhin hektisch zugehen wird. Aber hoffen wir das Beste -- für Euch und für mich.

Die von mir für das Jahr 1987 gesetzten Redaktions-Termine möchte ich beibehalten. So hoffe ich mit dem nächsten INFO schon in einem Monat aufwarten zu können (Der Termin ist ja Ende diesen Monat). Sicher wird das INFI dann etwas dünner ausfallen als das jetzige INFO.

Als weiteres möchte ich Euch in meinem Schluß nochmals an den Zehnzeiler-Wettbewerb erinnern. Ich rechne mit reger "Telefonitis" an den zwei Telefon-Abstimm-Tagen. Auch bin ich gerne bereit gleichzeitig kurze Mitteilungen (z.b. für die Börse) zu notieren, so daß sie dann noch kurzfristig im 20. INFO erscheinen können.
Sicher wird es für Euch genau so spannend, wie für mich, wer nun das "Programm des Jahres" in seiner Programmierkiste hat. Ich wünsche auf alle Fälle jedem das Glück Erster zu werden.

Zum Abschluß dieses INFO's verbleibt mir nur noch, mich für das Vertrauen, daß Ihr mir für ein weiteres INFO-Jahre ausgesprochen habt, zu bedanken und Euch viel Freude an der neuesten INFO zu wünschen.
In der Hoffnung, daß man sich telefonisch trifft und/oder bis zum nächsten INFO verbleibe ich Euer

Jens Neueder

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907	// 0431 /74047
Betz	Heinrich	St. Wolfgangsstraße 13	8551	Hausen	09191 /31698	// 09191 /611108
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621	// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441	Wettringen	05233 /4320	// 02557 /1236
Emmerich	Helmut	Waldstraße 5	6682	Ottweiler	06824 /4114	// -
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721 /21841	// 09721 /51256
Gromottka	Uwe	Lange Reihe 40	2803	Weyhe	0421 / 80496	// 0421 /300-6870
Heile	Heinz-Dieter	Marx Straße 70c	4320	Hattingen	02324 /67495	// -
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024	//
Hill	Peter	Bergstraße 65	6754	Otterberg	-	// -
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948 /542	// 02921 /70431
Kopschina	Peter	Strandallee 138	2409	Scharbeutz	-	// -
Krispin	Michael	Schwanstraße 8	4130	Moers 3	02841 /73690	// -
Krüger	Karl-Herbert	Bruchweg 65	4920	Lemgo	05261 /13686	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417	// 07022 /77442
Mand	Harald	Kleinflintbeker Straße 7	2302	Flintbek bei Kiel	04347 /3629	// 0431 /3013580
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083	// 04103 /702662
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913	// -
ObscherningKat	Helmut	1 RUE DES BRUYERES	F-68360	Soultz	0033089/762690	// -
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /3791	//
Reichelt	Dieter	Philipp-Schmitt-Straße 30	6902	Sandhausen	06224 /52906	// -
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551	// 04103 /605310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383	// 0203 /331383
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221 /707044	// -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631	// 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228 /643853	// -
Smerling	Frank	Tangstedter Str. 5	2080	Pinneberg	04101 /207284	//
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434 /454	// 08431/46423
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585	// 05401 /30096
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883	// 0231 /593883
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153 /1564	// -
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131 /32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285	//
Wacker	Fred	Postfach 2246	7550	Rasstatt	07222 /52574	// -
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361	// -
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686	// -

Stand: Juni 1987

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

CLUB 80

Clubinfo der TANDY - GENIE - und KOMTEK-ANWENDER

Lass't den Nachwuchs an die Kiste
— oder: wie der Vater so der Sohn
(Heinrich Betz Junior)

20. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
POSTFACH 27 / TEL.: 06124 / 3913

- I N H A L T S V E R Z E I C H N I S -

Seite:
und Autor:

Clubinternes

Neues vom Vorstand .... 1 - 3
Hartmut Obermann
Termine / Messen 4
Redaktion
Vorstellung neuer Clubmitglieder ... 5 - 8
Reiner Stober
Heinrich Betz
.. und noch ein Leserbrief.......... 9 - 10
Klaus-Jürgen Mühlenbein

Software

MSDOS im INFO 11
Pascal für Interessierte .... 12 - 14
Rüdiger Sörensen
Hardkopie auch ohne Interrupt ...... 15 - 16
Andreas Rychlik
Mathematischer überfluß .... 17 - 19
Klaus-Jürgen Mühlenbein
Ein Schoner für die Bildröhre ...... 20 - 21
Arnulf Sopp
Futter für Genie III's CRTC .... 22 - 26
Ulrich Heidenreich
Callerlei .... 27 - 34
.. weil Kürze so würzig ist .... 35 - 36
Klaus-Jürgen Mühlenbein
Kleinschrift für den EDTASM/PLUS ... 37 - 43
R-Befehl MSDOS-Like .... 44 - 56
Ulrich Heidenreich
Tape-Games auf dem Model III/4/4p .. 57
Umkehr ! .... 58 - 60
Hartmut Obermann
Keine Angst vor Haifischen! 61
Trennen in TSCRIPS 62
Klaus-Jürgen Mühlenbein

Seite:
und Autor:

Hardware

Der Wachhund 63
Heinrich Betz
Auslesen des Grafikspeichers .... 65 - 70
Helmut Bernhardt, Jörg Seelmann-Eggebert
Tastatur- und Druckerplatine .... 71 - 80
Bernd Drowälder
Anschluß gesucht .... 81 - 89
Artikel aus c't

Börse

Wer hat was -- wer will was .... 91 - 92

Sonstiges

Nachtrag: In VISICALC sortieren 93
Klaus-Jürgen Mühlenbein
Fehler: Extended NEWDOS 94
Nachtrag: RESET-feste MEMDISK 94
Hartmut Obermann
Im Schatten des Dinosauriers ...... 95 - 96
....
Auf zu neuen Ufern .... 97 - 98
Hartmut Obermann
PC statt GENIE ? .... 99 - 100
Paul-Jürgen Schmitz
Mathematik für Feinschmecker .... 101 - 107
Artikel aus PASCAL
Bin ich strahlengefärdet .... 109 - 110
....

Club Bibliothek

Club-80 - Bücherei .... 111 - 116
Hartmut Obermann

Die letzten Seiten

Impressum 117
Schluß 118
Clubmitgliederadressen .... am INFO-Ende
Redaktion

Neues vom Vorstand

Entschuldigung!

Durch meinen Beitrag "Schau - trau keinem?", erschienen im letzten Info Nr. 19 unter der Rubrik "Neues vom Vorstand", habe ich mich selbst mit beiden Beinen in den berühmten Eimer mit der stinkenden braunen Masse gestellt (und ich kann euch sagen, es ist keine Schokolade!)!

Mit dem besagten Beitrag wollte ich den Mitgliedern, die sich nicht trauen aktiv am Clubinfo mitzuarbeiten, weil sie fürchten, wegen orthographischer Fehler belächelt zu werden, diese Angst nehmen. Als Aufhänger dazu diente mir ein Beitrag von Kajott. Dabei habe ich in meinem Übereifer einige Regeln vergessen, die ich ab sofort zu beachten verspreche!

1. Ich habe den Beitrag direkt im Anschluß an die Lektüre der beiden Artikel von Kajott Mühlenbein und Bernd Drowälder geschrieben, statt erst mal eine Nacht darüber zu schlafen und damit gewisse Distanz zu dem Problem zu gewinnen.
2. Ich habe angenommen, daß der Beitrag von Kajott in einem direkten Zusammenhang mit dem von Bernd steht, es aber unterlassen nachzuprüfen, ob dieser Umstand wirklich zutrifft!
3. Ich habe meinen Beitrag direkt an die Redaktion geschickt, es aber unterlassen Kajott einen Abzug davon zukommen zu lassen und ihn dadurch der Möglichkeit beraubt, die Sache noch vor der Veröffentlichung aufzuklären!

Hiermit möchte ich mich in aller Form bei Kajott entschuldigen und verspreche nochmals, die drei von mir verletzten Regeln ab sofort einzuhalten!

Um das Ganze nocheinmal aus der Sicht von Kajott darzustellen, ist es mir ein Bedürfnis, seinen Brief an mich im folgenden zu veröffentlichen! (siehe Seite 2)

Trotz dieses bedauerlichen Vorgangs stehe ich aber zu meiner Meinung, daß Veröffentlichungen wie "Trau - schau, wem!" und "DER - DIE - DAS" (Info 19, Seite 75/76) zwar sicher der Allgemeinbildung und vielleicht auch der Hebung des nach Kajott "stellenweise grausamen Clubniveaus" dienen, bezweifle aber, daß sich dadurch neue Artikelschreiber für's Info finden lassen! Und ich bleibe auch dabei, daß für mich die fachliche weit vor der orthographischen Richtigkeit rangiert und deshalb nicht ständig auf Tipp- und/oder Schreibfehlern herumgeritten werden sollte (egal, ob nun jemand direkt angesprochen wird oder die Belehrung "der Allgemeinheit" gilt)!

Damit soll (zumindest für diesmal) das Thema abgeschlossen sein. Was es noch so an Neuigkeiten gibt, erfahrt ihr nach den beiden Briefen von Kajott, wobei ich noch darauf hinweisen möchte, daß der Brief an mich gekürzt wiedergegeben wird!

Hartmut Obermann

Klaus-Jürgen Mühlenbein
Diplom-Physiker

Im Mönchgarten 28
6940 Weinheim
☎06201/55052

K.-J.MÜHLENBEIN, Am Mönchgarten 28, 6940 Weinheim

Herrn
Hartmut Obermann

Schwalbacherstr.6

6209 Heidenrod/Kemel

Ihr Schreiben	Ihr Zeichen	Mein Schreiben	Datum
18.06.87	*	*	24.06.87

Lieber Hartmut!

Ich bin recht traurig darüber, daß Du die Atmosphäre zwischen uns vergiftet hast. Ich habe noch nie jemand persönlich öffentlich angegriffen, und meine Wertschätzung für den sachlichen Gehalt dessen, was unsere Kollegen veröffentlichen, solltest Du längst kennen. Wenn ich schon sonst nicht viel verstehe, kann man mir wenigstens die Versuche konzidieren, das "Clubniveau" zu heben, denn das ist stellenweise grausam (was keine "Einbildung" ist) - aber das stets ohne persönlichen Angriff. Bernds Artikel habe ich vorher gar nicht gekannt. Wie hätte ich mich dazu überhaupt äußern können?

Natürlich muß ich mich jetzt gegenüber Bernd rechtfertigen. Da ich nicht hinterhältig bin, füge ich Dir einen Durchschlag bei, denn es gehört sich, daß man jemand informiert, ~~wenn~~ bevor man über ihn spricht/schreibt x). Eine gewisse Schärfe darin mußt Du aus meiner großen Enttäuschung und Verärgerung heraus verstehen. Denn gegenüber allen übrigen Mitgliedern bleibt ja nun einmal ein falscher Eindruck haften...

Falls Du überhaupt an meiner Mitgliedschaft noch interessiert bist (auch wenn ich im Moment keine organisatorische Aufgabe übernehmen konnte), solltest Du im nächsten INFO eine Richtigstellung bringen - denn Irren ist ja menschlich, eine Begradigung ist keine Schande und ich bin keineswegs nachtragend.

Anders täte mir die Konsequenz sehr leid, denn ich schätze den CLUB 80 nicht nur wegen der sachlichen Hilfe, die mir schon zuteil wurde, sondern auch wegen meiner Zuneigung zu einer jungen Gemeinschaft, die von einer (unpolitischen) Sache begeistert ist, und weil sich seine Aktivität und Dynamik - angefangen von der wieder bestens gestalteten Clubzeitung über die Bibliotheken bis zu den Treffen usf. - sehen lassen können, so daß ein Ralf Folkerts mit seinem müden "Bremerhavener" (aus dem ich z.Z. gerade den Gerhard Loose für uns werbe) und dem hingequälten INFO sich nur alle 10 Finger danach lecken kann!

Denk mal drüber nach.
Ich würde mich wieder freuen.
Gruß an Deine Frau!

Kajott

CP/M - Public Domain

Unsere CLUB 80 Programmbibliothek hat, dank eines günstigen Angebotes, eine erhebliche Erweiterung erfahren. Durch Vermittlung von Gerald Dreyer konnte der Club ca. 80 Disketten der CP/M UserGroup übernehmen. Die Disketten enthalten nur Public Domain-Software und kann daher ohne Bedenken weitergegeben werden.

Wer die Verwaltung dieses riesigen Softwarepools übernehmen wird, ist zur Zeit noch nicht ganz geklärt. Im Gespräch ist Andreas Rychlik; Werner Schäfer möchte ich die zusätzliche Arbeit eigentlich nicht aufbürden, er hat mit der Diskothek bestimmt schon genug zu tun. Sollte sich noch jemand in der Lage sehen, diese Aufgabe zu übernehmen (er sollte wenn möglich über 80-Spurlaufwerke verfügen und möglichst viele CP/M-Formate lesen/schreiben/formatieren können) so soll er sich bitte bei mir melden.

Ich hoffe euch schon im nächsten Info näheres mitteilen zu können!

Umzug!

Vielleicht ist es euch schon beim Durchblättern des letzten Infos aufgefallen: Ich ziehe nicht um!
Mein Dienstherr hat es sich, vierzehn Tage bevor ich an meinen "neuen" Dienstort umziehen konnte, anders überlegt und mich an den "alten" zurück versetzt. Damit bleibt also (zumindestens vorerst) alles beim alten. Erfreulich ist, daß ich nun auch wieder über die Woche zu abendlicher Stunde erreichbar bin und mich wieder mehr um den Club und die Mitglieder kümmern kann!

Tandy in Deutschland tot!

Schon im Info Nr. 18 veröffentlichten wir einen Zeitungsausschnitt, dem zufolge Tandy in Deutschland seine Pforten schließt. Im Info 19 wurde auf Seite 79 eine Liste von Vertragshändlern abgedruckt, die die Aufgaben (Verkauf, Wartung) der Tandy-Läden alter Art übernehmen. Soweit mir bekannt, wird auch die Tandy-Zentrale in Ratinen geschlossen bzw. hat schon dicht gemacht.

Durch diese Maßnahmen wird das Arbeiten mit unseren, zum Teil schon recht betagten, Maschinen sicher nicht leichter und wenn mal irgend etwas kaputt geht, steht man ganz schön im Regen! Tandy scheint jedenfalls nun auch noch den letzten Rest Interesse an den "alten" Maschinen der TRS 80-Serie und ihren Benutzern verloren zu haben und so wird diese Spezies der Computeranwender wohl immer kleiner und kleiner werden!

Dadurch wird es noch wichtiger, sich gegenseitig zu helfen wo es nur irgend geht. Dies ist auch die Chance für Zusammenschlüsse, wie den CLUB 80, noch eine relativ lange Zeit weiter zu bestehen, obwohl die betreuten Computer eigentlich schon zu den Dinosauriern der Homecomputerszene gehören und Liebhaberstücke sind.

Damit genug für diesmal. Ich wünsche euch viel Spaß bei der Lektüre des Infos und noch einen schönen Urlaub und Sommer,
euer

Hartmut Obermann

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....20. September 1987

Norddeutsches Regionaltreffen .... ???????? 1987

A c h t u n g :
Das Regionaltreffen der "Nordlichter" verschiebt sich!
Ein neuer Termin wird nach Absprache bekanntgegeben!!!

-- Messen '87 --

HOBBY ELEKTRONIK 87 ....Stuttgart 5. - 8. November 1987
PRODUCTRONICA ....München 10. - 14. November 1987

Regionaltreffen im Großraum Stuttgart

Als Termin schlagen wir vor:
Mitte Oktober bis Mitte November

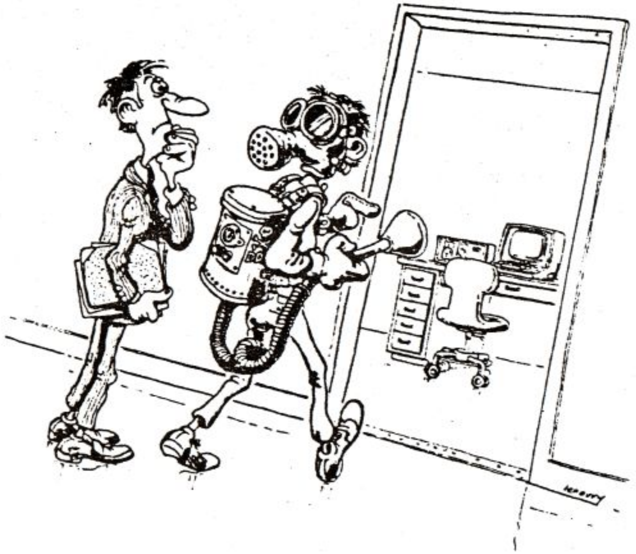
Wer Interesse hat, an dem Treffen teilzunehmen, melde sich bitte beim Klaus Hermann.

Liebe Freunde!

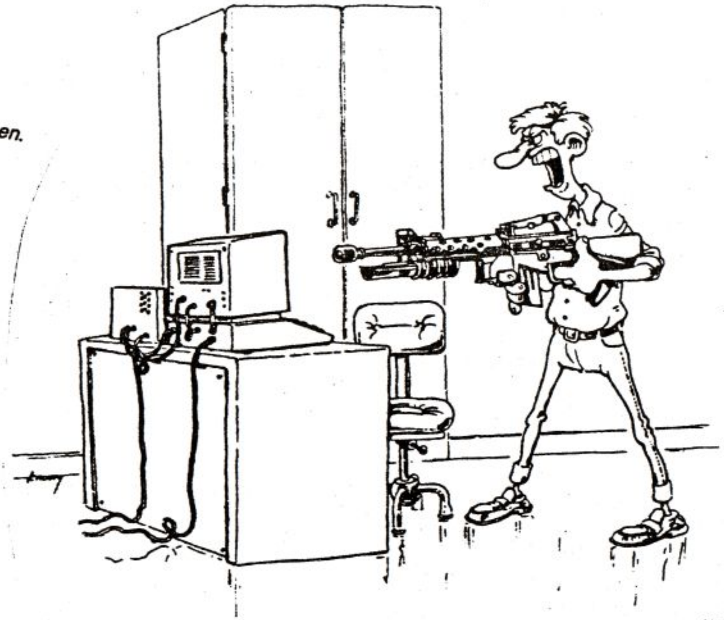
Als neues Mitglied im Club möchte ich mich kurz vorstellen.

Ich heiße Reiner Stober, geb.I5.I0.56, komme aus der Gegend von Hannover und bin dort auch zur Zeit in einer Umschulung als Meß und Regelmechaniker.Aus beruflichen Gründen arbeite ich viel am Computer und glaube, das ich von Euch noch einiges dazulernen kann.Meine Anlagen sind ein Genie mit Centronicstreiber Modell I mit I6 k,sowie ein Drucker von General Elektrik, ein ZX-8I mit I6 k, und seit kurzem auch ein TRS-80 Modell I mit Controller. Leider habe ich für den TRS-80 kein Handbuch und brauche dringend die Belegungen für den Drucker.Es handelt sich um zwei Steckplätze mit je 34 Belegungen.Ferner suche ich ein 5 I/4 Zoll Laufwerk möglichst unter I00.-DM. Denn bei Völkner Elektronik bekommt man Epson Laufwerke schon für ca.I50.-DM.Als erstes suche ich einige Programme auf Cassette wie z.b.s.Assembler.Auch mit der Textverarbeitung ist nicht viel los.Es ist ein Programm aus der MC für den TRS-80.
Wer mich anrufen möchte, meine Telefon Nr.05I53/I564 aber nur am Wochenende ab Freitags I8.00 Uhr.

Gruß, Reiner.



Diesmal werde ich ihm die Mucken für immer austreiben.



Was soll das heißen, Bedienerfehler!?

HEFT
20
August
1987

Safety First

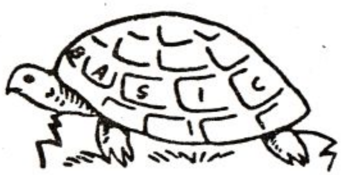
Als neues Mitglied habe ich mit Bewunderung in den Infos gelesen, in welche Tiefen der Hard- und Software des TRS-80 manche aktiven Clubfreunde eingestiegen sind. Offen gesagt, habe ich bis jetzt bei meinem 4P-Computer nur wenig unter der Oberfläche geschürft. Er dient mir als nützliches Werkzeug, um Assemblerprogramme für Single-Board-Computer zu erstellen.

Das Werkzeug

Etwas verlegen wurde ich, als mir Hartmut Obermann empfahl, als Clubneuling einen Beitrag in einem der Infos zu veröffentlichen. Was sollte ich von meinen bescheidenen Kenntnissen an andere Clubmitglieder weitergeben? Doch im Gespräch stellte sich heraus, daß nicht nur die Anatomie des Computers interessiert, sondern auch seine Anwendungen. Da bin ich dann schon eher wieder dabei, denn seit einigen Jahren betreue ich die Meß- und Prozeßtechnik in einer mittelständischen Schokoladenfabrik.

Wie viele von Euch habe ich die erste Bekanntschaft mit Computern über die Programmiersprache BASIC gemacht. Was lag näher, als zu versuchen, kleine Prozeßsteuerungen auch in dieser Sprache zu programmieren. Vor einigen Jahren lieferte National Semiconductor den Prozessor INS 8073, der in seinem internen ROM einen Basic-Interpreter mitführte. Der Aufbau war so geschickt gemacht, daß man mit einem RS-232-Terminal und zwei externen Spannungen völlig autark Programme entwickeln und in ROMs brennen konnte. Mehrere Steckkarten mit diesem Prozessor sind heute noch an wichtigen Stellen der Schokoladenherstellung im Einsatz und haben über Jahre hinweg ihre Leistungsfähigkeit und Zuverlässigkeit bewiesen.

Zwei Nachteile haben jedoch dazu geführt, daß ich den National Prozessor durch den Z-80 ersetzte. Einmal wird der INS 8073 seit etwa drei Jahren nicht mehr geliefert, zum anderen habe ich feststellen müssen, daß BASIC zwar bequem zu programmieren, aber bei der Ausführung des Programms recht langsam ist. Bei einer Anwendung ging es darum, die von einer Waage mit 300 bd seriell abgegebenen Gewichtswerte zu empfangen und zu verarbeiten. Schon bei diesem relativ langsamen Datenstrom war der INS 8073 mit seinem Basicinterpreter völlig überlastet. Es kam zu Overrun-Errors im UART und zum Datenverlust.



Erste Versuche mit einem Single Board Computer "EMUF" der Firma Kanis in Pöcking brachten die gewünschte Steigerung der Geschwindigkeit. Erheblich mehr Aufwand war jedoch für die Programmierung erforderlich, nachdem sich kein freundliches "READY" mehr nach dem Einschalten meldete, sondern der Z-80 ab seiner Adresse 0 mit Programmcode versorgt werden wollte. Mit den Steckkarten, die eine Z80-CPU, UART, CTC, ROM und RAM enthalten, habe ich mehrere Steuerungen aufgebaut. Je nach Anforderung an die Genauigkeit wurden verschiedene Analog/Digital-Wandler eingesetzt: CA3162, ICL 7109, ICL 7135.

"Safety First" habe ich als Überschrift gewählt und tatsächlich ist die Zuverlässigkeit oberstes Gebot im industriellen Bereich. Die Schäden, die beim Ausfall oder bei Fehlfunktion der Elektronik entstehen können, sind enorm. Als Beispiel sei der Defekt eines kleinen Kondensators genannt, der den A/D-Konverter dazu brachte, zu niedrige Werte anzugeben. Die Folge war, daß ein großer Behälter mit Kakaobutter beim Füllen überlief und eine beachtliche Menge des teuren Rohstoffs im Abfluß verschwand.

Nur erstklassige Bauelemente zu verwenden und diese möglichst noch überzudimensionieren ist eine der goldenen Regeln um Ausfällen vorzubeugen. Bei Seriengeräten, die in großen Stückzahlen hergestellt werden, muß man mit dem Pfennig rechnen, bei Sonderanfertigungen für Einsätze an wichtigen Stellen im Prozeß lohnt es sich nicht, am Material zu sparen. Ein einziger Produktionsausfall oder Ausschußproduktion kosten ein Vielfaches der ersparten Kosten beim Einkauf der Bauelemente.

Eine gefährliche Fehlerquelle sind Störungen, die über das Starkstromnetz oder über die I/O-Leitungen eingefangen werden. Über dieses Thema und die Maßnahmen, um die Einstreuung von Spikes zu verhindern, ließe sich ein eigener Bericht schreiben. Als recht störungsfest haben sich Schaltkreise in CMOS-Technik erwiesen. Alle aktiven Bauelemente auf der SBC-Platine habe ich in dieser Technik bestückt. Bei den Peripherieschaltungen (Decoder, Latches, Monostables) hat sich am besten die Metal Gate CMOS-Serie 4000 bewährt, die wegen iher hohen Signalpegel einen guten statischen Störabstand hat und wegen ihrer relativ langsamen Schaltgeschwindigkeit (2 MHz , bei 5 Volt) unempfindlich gegen kurze Spikes ist

Besondere Anforderungen an die Zuverlässigkeit gelten auch bei der Software. Sämtliche nur denkbaren äußeren Bedingungen müssen so abgefangen werden, daß es nie zum Programmabsturz kommt. Zum Test muß der Bediener auf alle Tasten in beliebiger Folge und Kombination drücken können, um anschließend immer wieder in die Grundstellung zuückzufinden.

Meinen TRS-80 verwende ich, um die Programme für die Single Board Computer zu entwickeln. Per Download über die serielle Schnittstelle werden sie zunächst ins RAM des SBC geladen und dort getestet. Sind die Ergebnisse des Versuchslaufs zufriedenstellend, wird der Objectcode über die serielle Schnittstelle an einen EPROM-Programmer (Minato, Japan) übertragen und das dort erstellte EPROM dem SBC eingesetzt.

Heinrich Betz

...und noch ein Leserbrief...

an die stumme Gemeinde

==> mit Preisausschreiben ! <==

Liebe Kollegen,
die ihr bisher stumm wart! Sicher habt ihr entweder-
- schon einmal ein kleines BASIC-Programm geschrieben; oder -
- euch an einem Maschinenprogramm (in Assembler) versucht oder-
- bei einem gekauften, geschenkten oder... also bei einem nicht selbst verfaßten Programm irgendeine Entdeckung gemacht oder-
- einen "Hänger" oder sonstige Schwierigkeit und somit eine Frage dazu gehabt; oder -
- irgendetwas sogar schon selbst an eurem Computer oder seiner Peripherie herumgebastelt; oder -
- zumindestens hierzu ein paar Fragen gehabt; oder -
- ein Programm, einen INFO-Beitrag oder eine Publikation in irgendeiner der zahlreichen Zeitschriften entdeckt, das/der/die euch besonders ge- oder mißfallen hat; oder -
- oder - oder - - -

Wenn nichts davon zuträfe, wäret ihr auf keinen Fall in einem Club. Und schon gar nicht in unserem! Denn dann läge euer Interesse ja irgendwo anders. Ihr seid doch bestimmt nicht daran interessiert, ständig über Sachen im INFO zu lesen, mit denen ihr nicht im geringsten zu tun habt bzw. in irgendeiner Form, auch nur auf einem Teilgebiet, auch selbst umgeht!

Also darf ich davon ausgehen, daß Ihr alle irgendwie und irgendwo Praktiker seid. Ja, auch als sogenannte "Einsteiger" oder Anfänger!

Wißt ihr übrigens, wer der größte Praktiker ist?

Der Anfänger!! * Der "probiert" nämlich herum und herum und herum - und eines Tages entscheidet es sich: Entweder er gibt auf ==> dann tritt er auch in keinen Computerclub ein; oder er hat die "Zitterschwelle" überwunden und ist gefesselt; dann geht er vielleicht in einen Club. Falls er das tut, läßt er sich von den Kollegen, die schon weiter sind, anregen. Und auf einmal ist er "mittendrin" im Kreis der "Hobbyisten" (auf Deutsch: "Freaks") und merkt gar nicht, daß hinter ihm Neulinge - nicht nur Jüngere! - stehen, die, wie einst er selbst, noch am Anfang sind und die ersten Fragen stellen!

Die ersten "dummen Fragen"? - Ihr wißt doch: Dumme Fragen gibt es nicht - nur dumme Antworten! Wer hier fragt, bekommt immer eine gute Antwort von irgendeinem unseres Häufleins von 55 Mann (Stand: Juni 87); jawohl, dafür verbürge ich mich! Eine Frage, ein Bericht, ein Programm o.a. kann gar nicht "so dumm" sein, daß es nicht abgedruckt und von allen Lesern beachtet und gewürdigt würde! Diese Erfahrung habe ich an mir selbst gemacht. Es gibt nämlich garantiert immer noch eine Handvoll Leute, die zur Zeit (!) noch weniger wissen und beherrschen als man selbst! Und für die ist jeder Beitrag äußerst wertvoll, selbst der kleinste. Das ist meine persönliche Erfahrung in drei Clubjahren. Und es ist wirklich schön, hin und wieder ein Echo zu bekommen!!

Vor kritischen Lesern braucht sich niemand zu fürchten. Für etwaige Tippfehler oder Fehler in der Rechtschreibung (wer hat schon eine Sekretärin!) braucht man sich nicht zu schämen, wenn man darin nicht so sicher ist! Zuallererst kommt es darauf an, was einer mitteilen will, und wenn es, wie gesagt, noch so "bescheiden" ist. Wir sind keine Politiker, die mit schönen Worten hohle Phrasen - sogenannte "Worthülsen" - von sich geben. (Auch die machen oft genug sprachliche Schnitzer!)
Und schon gar keine lauten, aber nicht lauteren Werbeleute!

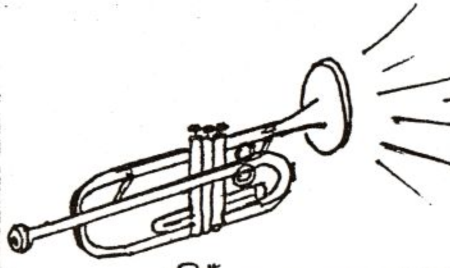
Und am allerwenigsten braucht sich einer zu schämen, dem eine Hilfe angeboten wird, sei es beim Programmieren, Basteln, Gehen, Fliegen, Sprechen oder Schreiben...!

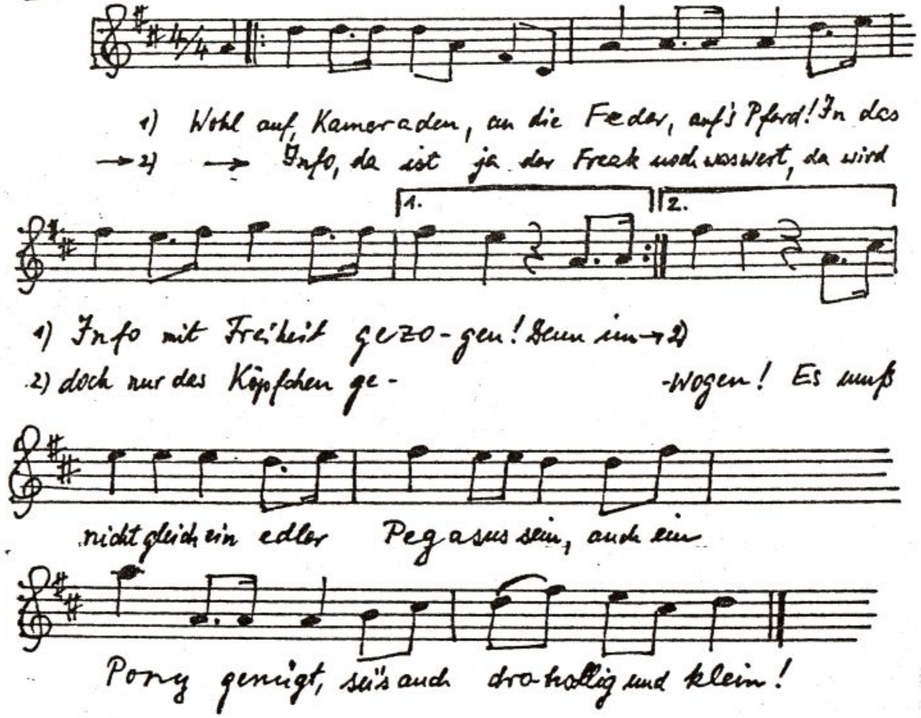
← Wer die Trompete bläst? Die CPU natürlich!

Wer kennt das alte Lied der Kavallerie? Bei den Reitern und Rittersleuten des CLUB 80 lautet es:

Wohlauf, Kameraden, an die Feder, aufs Pferd!
Im das INFO mit Freiheit gezogen!
Denn im INFO, da ist ja der Freak noch was wert,
da wird doch nur das Köpfchen gewogen!
Es muß nicht gleich ein edler Pegasus sein,
auch ein Pony genügt, sei's auch drollig und klein!

Jens und Peter sollen mal zeigen, was für einen Zustrom an Beiträgen aus eurem Kreis sie "verkraften" können -
das walte Gott! - KaJot





[Melodie und halber Text: Altes Marschlied der preußischen Kavallerie —
(aber auch bei Bayern beliebt) → Textvarianten von ↑ - Jott]

⇒ Wer diese Melodie auf Disk (40 Tr., sonst beliebig) einsendet [Maschinensprache], ⇐
⇒ erhält von mir eine Flasche Reitersekt (Flaschengärung)
⇒ Marke „DERBY"! ⇐

KJ

MSDOS im INFO ???????

Im letzten Info wurde die Frage aufgeworfen (von wem, konnte ich nicht lesen. Verzeihung !!), ob es sinnvoll sei, sich als Z80-Klein-Unternehmer mit der großen weiten Welt der IBM-Kompatiblen zu befassen. Dazu hätte ich auch den einen oder anderen Gedanken zu Papier zu bringen. Im Rahmen meines Studiums bringe ich viel Zeit vor der Tastatur eines IBM-AT zu, von dessen Eigenarten und Betriebssystem man auf diese Weise ja auch etwas ergattert, sofern man sich nicht nur mit Textverarbeitung beschäftigt, oder gar damit, für teures Geld erworbenen Programmen bei der Arbeit zuzusehen. Das Eigentümliche an diesem Rechner/Betriebssystem ist bekanntlich die weite Verbreitung, von der wir nur träumen können (CP/M Benutzer natürlich ausgenommen). Es hat eigentlich schon immer Leute gegeben, welche die Leistungsfähigkeit jener Maschinen mit derjenigen eines Apparates vergleichen, der mal gerade ohne Tricks 64 Kbyte adressieren kann. Deswegen habe ich keine Lust,das zum wiederholten Male zu tun.
MSDOS ist ein Betriebssystem, das wesentlich mehr kann als alles, was uns beim Computern das Leben leichter macht. Und die Prozessoren sind schneller. Und der Speicher ist mehr. Und die Festplatten sind billiger. Aber leider, MSDOS läuft nicht auf dem Z80. (Ach, übrigens, schon von Zilogs neuem Z280 gehört ?)
Also, warum eine MSDOS-Ecke ? Sollen Leute, die sich doch einer anderen Hardware verschrieben haben, noch Beiträge für den MSDOS-Blätterwald beisteuern ? Schließlich, man braucht ja nur eine der gängigen Computerzeitschriften aufzuschlagen und bekommt Tips, Programme, Literaturhinweise, Händlernachweise, Hardwareangebote, u.s.w.u.s.f. in Hülle und Fülle zu sehen (es kommen mir manchmal die Tränen). Selbst 80-Micro publiziert seit einiger Zeit mehr für das Model 1000 und Co. (Hardware Requirements : 128K, GWBasic) als für das Model 4.
Dennoch gibt es eine Reihe von Gemeinsamkeiten und Anregungen, die für uns von Nutzen sein können. Integrierte Software gehört meiner Meinung nach gerade nicht dazu. In erster Linie sind wohl Softwarelösungen zu nennen, insoweit sie in für uns zugänglichen Hochsprachen formuliert sind und nicht gerade die Neuorganisation von Harddisks behandeln. Ein gängiges Werkzeug in diesem Zusammenhang ist TURBO-Pascal, da es in nur wenig abweichenden Versionen für MSDOS und CP/M existiert. Es gibt eine ganze Reihe von glänzenden Programmen, die in dieser Sprache formuliert werden/wurden. Auch C und Fortran, COBOL und wie sie alle heißen machen Ideen portabel. Insofern lasse ich mir ein Eckchen MSDOS gefallen, wenn ich es auf meinem Model 4 gebrauchen kann. Die Moral von der Geschichte :

1) Das Bessere (?) muß nicht immer des Guten Feind sein.

2) lernt Pascal !!!

In diesem Sinne, c.u., Rüdiger.

Pascal für Interessierte

In einem anderen Beitrag habe ich zum Erlernen der Sprache Pascal aufgefordert. Obwohl es eine Unzahl von Literatur zu diesem Thema gibt, will ich versuchen, in einigen kleinen Aufsätzen meine Erfahrungen mit dieser Sprache zu schildern. Meine erste Begegnung fand, glaube ich, so um 1982 statt, als ich über die Sprache las. Bis dahin beschränkten sich meine Programmiererfahrungen auf FORTRAN 66, eine aus heutiger Sicht unakzeptable Tortur. Dann mußte ich im Rahmen eines Numerik-Praktikums ALGOL 60 studieren, meine erste Bekanntschaft mit sogenannten strukturierenden oder auch strukturierten Sprachen. Um keine Mißverständnisse aufkommen zu lassen : Strukturiert programmieren kann man und sollte man in jeder Sprache, selbst in Assembler. Eine strukturierte Sprache ist aber ein außerordentlich nützliches Hilfsmittel. Der Unterschied zwischen Pascal und BASIC soll nachfolgend etwas verdeutlicht werden, denn man kann wohl davon ausgehen, daa BASIC verfügbar und benutzbar für jeden ist.
Pascal wurde als Hilfsmittel zum Lehren der Programmierkunst entwickelt. Darunter hat der Anwender ein wenig zu leiden. Viele Implementierungen an Großrechenanlagen sind für Naturwissenschaftler ungeeignet, weil numerische Funktionen schlicht fehlen und nur schlecht ergänzt werden können (Beispiel komplexe Zahlen) oder weil die strenge Typüberprüfung Pascals, für Lehrzwecke ideal, die Modularisierung von Programmen schwierig macht. All das gilt für Micros nicht (mehr) uneingeschränkt. Turbo-Pascal ist DIE Sprache für CP/M. Keine andere mir bekannte Hochsprachenimplementierung ist nur annähernd so leistungsfähig, jedenfalls nicht für CP/M.

Aber nun ans Eingemachte! Beschäftigen wir uns zunächst mit den Datentypen von Pascal. Was ein Typ ist, weiß ein jeder, der schon mal ein DEFINT in BASIC gesehen hat. Pascal unterscheidet eine Reihe von vordefinierten Typen, von denen die meisten Bestandteil aller Sprachen sind.

INTEGER :
das sind ganze Zahlen im Bereich von -MAXINT bis +MAXINT, wobei diese vordefinierte Konstante von der jeweiligen Implementierung abhängt, bei 8-Bit-Rechnern ist MAXINT typisch 7FFFH.

REAL :
das sind rationale (nicht etwa reelle) Zahlen, die äußerlich durch Dezimalbrüche dargestellt werden, also zum Beispiel 3.141592. Der gültige Bereich der REAL-Zahlen variiert stark je nach System, bei TURBO ist m.E. die größte darstellbare REAL-Zahl so um 1.0E+37

CHAR :
das ist schon eine Neuerung. Eine Variable vom Typ CHAR hat den Wert eines Zeichens aus dem verfügbaren Zeichensatz, und das ist i.allg. ASCII.

Auf den Typen INTEGER und CHAR ist eine Ordnungsfunktion definiert, d.h. man kann die Werte eines Types abzählen, z.B. folgt die INTEGER-Zahl 423 auf 422 und das Zeichen "2" ist (in der Regel) der Vorgänger des Zeichens "3". Typen mit Ordnugsfunktion werden von Pascal bevorzugt behandelt, es gibt einige vordefinierte Funktionen, die auf diese Typen Anwendung finden. Der Typ REAL hat keine Ordnungsfunktion (Preisfrage : welche Zahl folgt auf 1.234 ?).

Mit einem Typ allein kann man noch kein Programm entwickeln. Wie man weiß, benötigt man dazu mehr, vor allem Kontrollstrukturen und Variablen. Auch in BASIC sind Variablen typbehaftet, man kann einer Stringvariablen A$ keinen Ausdruck zuweisen, der als Ergebnis eine INTEGER-Größe hat.
Der Ausdruck :

A$ = 2*J

ist illegal, unabhängig vom Typ der Größe J. Im Gegensatz zu BASIC muß man in Pascal eine Variable vereinbaren, das heißt, der Übersetzer meckert, wenn ein Name im Programm auftaucht, den er nicht kennt. Wenn BASIC dasselbe passiert, nimmt es einfach an, daß es eine SINGLE-Größe ist. Also nun zur formalen Deklaration (Erklärung, Bekanntmachung) von Variablen. Im Programm muß zuerst das Schlüsselwort VAR stehen, gefolgt von einem oder mehreren Namen und deren Typen. Beispiel :

```
VAR x,y : REAL; i,j : INTEGER; c : char;
```

Durch diese Zeile werden Speicherplätze für 5 Variablen reserviert und deren Namen und Adressen dem Übersetzer bekannt gemacht. In Pascal muß ein Name mit einem Alphazeichen anfangen, dann dürfen Alphazeichen oder Ziffern folgen. Die erlaubte Gesamtlänge von Namen ist bei TURBO (glaube ich) 32 Zeichen, wobei auch der Unterstrich erlaubt ist. Zurück zu den Typen : REAL, CHAR und INTEGER sind unverträglich. Man kann (bis auf eine Ausnahme) Werte nicht zwischen Variablen dieser Typen zuweisen, die Anweisung
j := x;
mit den oben definierten Variablen ist syntaktisch falsch. Dagegen ist
x := j;
richtig (die erwähnte Ausnahme), Pascal läßt dies zu, weil keine versehentlichen Rundungsfehler auftreten können (Es gibt keine andere Möglichkeit, INTEGER-Größen in REAL umzuwandeln). Darüber hinaus gibt es eine Reihe von vordefinierten Funktionen, mit denen Typtransfer legalisiert wird :
ROUND(x) liefert den gerundeten Wert der REAL-Variablen x zurück.
TRUNC(x) liefert den abgerundeten Wert.
CHR(i) liefert das Zeichen mit der Ordnungszahl i.
ORD(c) liefert die Ordnungszahl des Zeichens c;
Den Zuweisungsoperator haben wir auch schon kenngelernt (das "Gleichheitszeichen") :

" := "

Für das allererste Pascal-Programm fehlen nur noch ein paar Kleinigkeiten : der Programmkopf und die Blockklammern. Ein Pascal-Programm sieht immer so aus :

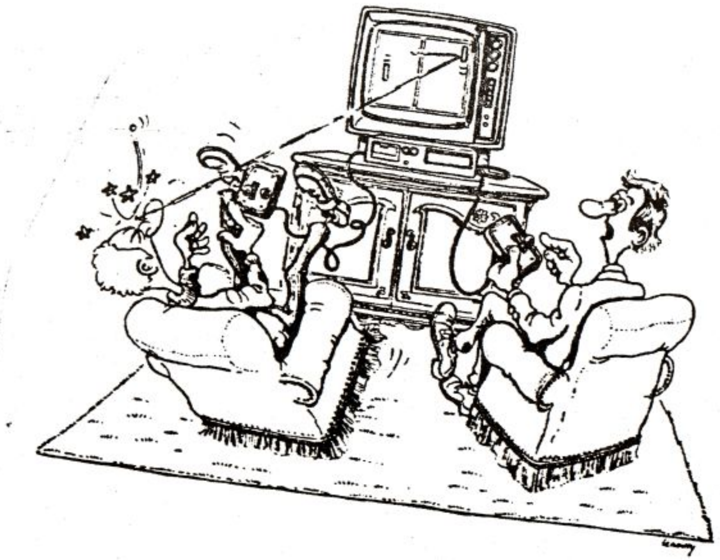
```
PROGRAM name (extfiles);
   var section;
   begin
   (* Programmrumpf *)
end.
```

Die VAR - Sektion haben wir behandelt, den Programmrumpf basteln wir noch :


```
PROGRAM erstes (INPUT,OUTPUT);
VAR
   x,y : REAL;
   i,j : INTEGER;
   c : CHAR;
BEGIN
   x := y*(i/j);
END.
```

Natürlich ist dieses Programm, obwohl syntaktisch richtig und vollständig, völlig blödsinnig. Es würde zwar arbeiten, aber seine Ergebnisse wären zufällig und blieben in Ermangelung von Ein/Ausgabe auch noch geheim. Beim nächsten Mal gibt's dann ein Richtiges zum Ausschneiden !

Viel Spaß mit der Syntax wünscht Rüdiger.



Oh, Verzeihung!

Hardkopie auch ohne Interrupt

Liebe Clubkameraden,

ich wollte schon immer Hardkopien von Spielen mit Blockgrafik machen, doch bei den meisten Spielen werden die Interrupts ausgeschaltet und somit ist keine Hardkopie möglich.

Auch ist es nicht möglich die Grafik solcher Spiele in die HRG 1B zu kopieren.

Ferner sind bei CP/M keine Bildschirm-Hardkopien möglich und der Befehl Ctrl-P, der die Bildschirmausgabe auf den Drucker umlenkt, klappt nicht immer.

Doch ist mir da eine Idee gekommen, die mir schon oft geholfen hat:

Beim booten einer Single-Density Diskette wird der erste Sektor in den Speicher zwischen 4200H und 42FFH geladen und anschließend springt der Rechner nach 4200H. Also habe ich ein Programm auf den ersten Sektor einer Diskette geschrieben, welches den Bildschirmspeicher von 3C00H bis 3FFFH in die oberen Bereiche des RAMs verschiebt. (In meinem Beispiel zwischen F000H und F3FFH.) Nachdem ich nun diese Diskette boote, hängt sich zwar der Rechner auf, aber der "alte" Bildschirminhalt ist im Bereich von F000H bis F3FFH verschoben worden. Wenn ich nun die System-Diskette wieder boote, wird der "alte" Bildschirmspeicher nicht zerstört, da ja das NEWDOS nur bis 5200H geht. Auch werden die Interrupts durch das booten wieder eingeschaltet. Anschließend wird der "alte" Bildschirminhalt duch ein kleines Programm wieder in den aktuellen Bildschirmspeicher verschoben und von dort aus kann er mit JKL als Hardkopie ausgedruckt werden.

Ich weiß, daß diese Lösung zwar recht umständlich und nicht optimal ist, doch ist mir keine bessere Möglichkeit eingefallen.

Das Programm MOVEUP:

Das Programm MOVEUP verschiebt den Bildschirmspeicher nach F000H:

```
4200  21003C   LD    HL,3C00H    ;von Speicherbereich
4203  1100F0   LD    DE,F000H    ;nach Speicherbereich
4206  010004   LD    BC,0400H    ;Anzahl der Bytes
4209  EDB0     LDIR              ;verschiebe
420B  C30000   JP    0000H       ;springe nach MEM SIZE
```

Man formatiert nun eine Diskette in Single-Density und schreibt das Hexlisting mit Superzap am Anfang des ersten Sektors. Die Befehlsfolge lautet z.B.:

```
PDRIVE 0,1 TI=A,TD=A,TC=40,SPT=10,GPL=2,DDSL=17,DDGA=2
FORMAT 1
SUPERZAP
DD   1,0
```

mit MOD00 hackt man das Hexlisting ein: 21003C1100F0010004EDB0C30000
nach ENTER und Y wird die Disk modifiziert.

Wenn man nun eine Hardcopy von einem Bild machen will, legt man die modifizierte Disk in das Laufwerk 0 und bootet. Der Rechner meldet sich dann zwar mit MEM SIZE, aber er hat vorher das Bild verschoben.

Die modifiziere Disk ist weiterhin noch als Datendiskette geeignet.

Programm MOVEDOWN

Nach booten der Systemdiskette ruft man folgendes Programm auf, welches man schnell mit einem Monitor einhacken kann:

```
F400  2100F0   LD    HL,F000H    ;von Speicherbereich
F403  11003C   LD    DE,3C00H    ;nach Speicherbereich
F406  010004   LD    BC,0400H    ;Anzahl der Bytes
F409  EDB0     LDIR              ;Verschiebe
F40B  214038   LD    HL,3840H    ;HL als Zeiger auf Tastatur
F40E  7E       LD    A,(HL)      ;hole Tastaturzeile
F40F  CB3F     SRL   A           ;Bit 0 ins Carry Flag schieben
F411  30FB     JR    NC,F40EH    ;springe, wenn ENTER nicht gedrückt
F413  C32D40   JP    402DH       ;springe ins DOS
```

Dieses Programm holt den "alten" Bildschirm wieder zurück und man kann mit JKL eine Hardkopie mit Blockgrafik machen. Mit ENTER kommt man wieder ins DOS.

Auch kann man den Bildschirm mit DUMP,NAMEN/BIL,F000H,F3FFH,402DH abspeichern.

Andreas Rychlik

* * * Mathematischer Überfluß * * *

Eckehard KUHN stellte mir eine mathematische Aufgabe:

"Schreibe ein Programm, das den SINUS (und andere trigonometrische Funktionen) sowie die quadratische Wurzel auf beliebig viele Stellen genau berechnet. Die Rechengeschwindigkeit spiele dabei keine Rolle!"

Ich dachte: Nichts leichter als das!
Trotzdem habe ich mich sehr darüber gefreut. Denn auch "leichte" Aufgaben können interessant sein. * Und diese hatte es in sich!

Die trigonometrischen Funktionen SINUS, COSINUS und TANGENS (sowie deren Umkehrungen) und die Wurzel sind Bestandteile unseres BASIC-Interpreters (oder ROM's - ich bitte um Nachsicht, wenn ich mich falsch artikuliere...)
Wozu also noch extra ein Programm dafür?
Ganz klar: Wenn ich z.B. Sternpositionen berechnen will, brauche ich viel mehr Dezimalstellen, als diese Maschine bietet. Wenn ich SIN(X) aufrufe, bekomme ich bestenfalls 8 Stellen (davon 7 genaue). Auch wenn ich Variable mit "doppelter Genauigkeit" wähle (X#), bringt das nicht mehr; es erscheinen dann zwar 16 Stellen, aber mindestens die letzten 9 Ziffern sind falsch!! (Probiert's und vergleicht mit eurem Taschenrechner!) Der 8-BITter schafft einfach nicht mehr.

Es sei hier dahingestellt, warum Eckehard mehr als 8 Stellen braucht (ich habe ihn absichtlich nicht gefragt, um mich bei dieser Aufgabenstellung nicht beeinflussen zu lassen). Es ist auch ganz egal; denn was aus dieser Bemühung herauskam, war für mich eine neue Erkenntnis (!): Mehr als 16 Stellen können gar nicht dargestellt werden! Auch wenn man zur "wissenschaftlichen" Schreibweise mit Gleitkomma, also zur Exponentendarstellung übergeht, gelingt es nicht, am Ende ein Ergebnis mit mehr als 16 Stellen niederzuschreiben! (Von deren "Gültigkeit" ganz zu schweigen!)

Immerhin: 16 richtige Stellen wären doppelt soviel wie 8. Mein Programm bringt 16 Stellen, davon mindestens 14 richtige. ("6 Richtige" am Samstag wären mir allerdings etwas lieber!)
Es bedient sich hierzu der bekannten Näherungsberechnungen durch Reihendarstellung der Funktionen. Die quadratische Wurzel wird mithilfe der binomischen Reihe in Taylor'scher Darstellung berechnet. Sie konvergiert zwar schlechter und ist daher etwas ungenauer. Aber, was soll's: 14 genaue Stellen genügen, um z.B. den Erdumfang auf einen Tausendstel Millimeter genau aus dem Erdradius zu berechnen! Die transzendente Zahl PI wurde von einem Großcomputer in viereinhalb Stunden mit 100265 Dezimalstellen berechnet, das ergab 22 Druckseiten... nochmal: Was soll's? *
Die Reihen werden abgebrochen, sobald zwei aufeinander folgende Glieder 16 gleiche Ziffern haben. Wie es dahinter ausschaut, "geht niemand was an"...

Demjenigen, dem es gelingt, mein Programm zu überbieten, möchte ich eigentlich den *NOBEL*-Preis des Monats August 1987 für binäre Mathematik verpassen.
Aber leider bin ich weder *NOBEL* noch nobel.
Sondern - sapperlot
nur der Xalot

Berechnung trigonometrischer Grundfunktionen

```
1 CLS:PRINT"                TRIGONOMETRISCHE BERECHNUNGEN
         (C) K.-J.MÜHLENBEIN, WEINHEIM 1987":PRINT:PRINT
2 DEFINTI,K:DEFDBLC,G,P,S,W,X:DIMG(35):PI=3.141592653589793:INPU
T"WINKEL <IN GRAD> ";W:X=W*PI/180:G(1)=X:SI=G(1):II=3:K=1:PRINT:
PRINT"           SINUS-BERECHNUNG":PRINT
3 G(II)=X:FORI=2TOII:PRINT§408+I+II,"*":G(II)=G(II)*X/I:NEXT:K=-
K:SI=SI+K*G(II)
4 IFSZ=SI,PRINT"SIN("W")="SI:PRINT:GOTO10
5 SZ=SI:II=II+2:GOTO3
10 PRINT"             COSINUS-BERECHNUNG":PRINT
11 G(0)=1:CO=G(0):K=1:II=2
12 G(II)=X:FORI=2TOII:PRINT§601+I+II,"*":G(II)=G(II)*X/I:NEXT:K=
-K:CO=CO+K*G(II)
13 IFCZ=CO,THENPRINT"COS("W")="CO:GOTO15
14 CZ=CO:II=II+2:GOTO12
15 PRINT:PRINT"              TANGENS-BERECHNUNG":PRINT"TAN("W")="SI/
CO:END
```

* * * * *

HARDCOPY eines Ausführungsbeispiels:

```
           TRIGONOMETRISCHE BERECHNUNGEN
         (C) K.-J.MÜHLENBEIN, WEINHEIM 1987


WINKEL <IN GRAD> ? 23.7

          SINUS-BERECHNUNG    *************************
SIN( 23.7 )= .4019477766559601

          COSINUS-BERECHNUNG * ***************************
COS( 23.7 )= .9156625933395612

          TANGENS-BERECHNUNG
TAN( 23.7 )= .4389693098524373
READY
>
```

(Anm.: Die Sterne bewahren den User vor Wartefrust...***)

WURZELBERECHNUNG

```
 5 CLS
   PRINT "                WURZELBERECHNUNG"
   PRINT
   *10*
   '***    <C> K.J.MÜHLENBEIN, WEINHEIM 1987 ***
10 DEFINT I,N
   DEFDBL G,W,X,Y,Z
20 INPUT "RADIKAND ";Y
   N = LEN ( STR$ ( INT (Y))) - 1
   IF N / 2 < > INT (N / 2),N = N + 1
25 IF 10^N < 32768,
   THEN DEFINT F
30 F = 10^N
   X = Y / F - 1
   G = X / 2
   W = 1 + G
   I = 1
   K = 1
40 GN = G * X * I / (I + 3)
   K = - K
   W = W + K * GN
   IF W = WZ,
   THEN GOTO 60
50 WZ = W
   I = I + 2
   G = GN
   PRINT "*";
   GOTO 40
60 Z = W
   FOR I = 1 TO N / 2
   Z = Z * 10
   LPRINT CHR$ (07)
   NEXT
   PRINT
   PRINT "WURZEL AUS "Y"="Z
   END
```


Ein Schoner für die Bildröhre

In letzter Zeit wurden verschiedentlich Programme veröffentlicht, die nach einer Weile den Bildschirm abschalten sollen, wenn keine Taste gedrückt wurde. Kocht sich der User nämlich gerade einen Kaffee, braucht die Beschichtung der Bildröhre ja nicht unbedingt mit Elektronen bombardiert zu werden. Eine nützliche Sache also.

Bei den in unserem Club vertretenen Computern geht das ohne weiteres auch. Sinnvoll ist es aber nur bei solchen, die soft den Bildschirm abschalten können, ohne ihn zu löschen. Die TRS-80-ähnlichen Maschinen müßten dafür einen Aufwand treiben, der das Unternehmen wieder fragwürdig erscheinen läßt: Bildschirminhalt in einen Puffer retten, Bildschirm löschen, bei gedrückter Taste Bildschirminhalt wieder restaurieren. Und das alles müßte im eh' schon arg gebeutelten Himem passieren.

Hier wird vorgestellt, wie es beim Genie IIIs gemacht werden kann. Als Zeitmaß dienen die Systeminterrupts, die alle 25 ms auftreten. Die hier benutzte Methode ist die Überprüfung des Bits 12 eines 16-Bit-Zählers (Bit 4 seines MSB). Wenn der Zähler 1000h erreicht hat, sind ca. 102 Sekunden verstrichen. Jetzt unterstellt das Programm, daß der User im Moment länger nachdenken muß oder überhaupt muß; der Bildschirm wird dunkel, ohne daß sein Inhalt verändert würde. Sobald eine Taste gedrückt wird, erscheint der alte Bildschirm wieder. In der Zeit dazwischen sah man lediglich den Cursor blinken, denn der wird selbständig vom Video-Controller erzeugt. Ihn auch noch auszuschalten, erschien mir vom Aufwand her übertrieben. Es kann auch nicht schaden, auf diese Weise zu bemerken, daß der Computer überhaupt noch läuft.

Programme, die die Interrupts maskieren, machen dieser Routine natürlich einen Strich durch die Rechnung. In einem zweiten Fall ist die Routine sinnlos, sogar fatal: Programme, die die Tastatur nicht über den DCB abfragen (z. B. alle SCIPSIT-Enkel), müssen nach 102 Sekunden ohne Monitor bedient werden. Aber da gibt es einen Ausweg, der lediglich ein bißchen umständlich ist: Wenn es dunkel wird, kann man mit <DFG> ins Mini-DOS oder mit <123> in den Debugger gehen und sofort anschließend mit <;> bzw. <G> und <ENTER> dieser Unbill begegnen. Das ist Geschmacksache. Ich ziehe es vor, in diesen Fällen einfach auf den Bildschirmknipser zu verzichten.

Das Programm erklärt sich wohl hinreichend aus den Kommentaren. Modifikationen für andere Genies dürften nach ihrer Lektüre kein Problem mehr sein. Eine Besonderheit muß jedoch erwähnt werden: Die Routine läuft im Adreßbereich des Druckertreibers. Der wurde von allem überflüssigen Quatsch befreit. Als Nebeneffekt sprang dabei heraus, daß alle Codes von 0-255 mit LPRINT CHR$(x) ausgegeben werden können. Im Memory-Dump ist der geänderte Druckertreiber unterstrichen. Die Nullen dazwischen enthalten später das Programm.

Ebenfalls erwähnenswert ist ein Schönheitsfehler: Bei Betätigung einer der F-Tasten wird der Bildschirm auch ausgeschaltet. Warum das so ist, weiß ich nicht. Aber damit kann man gut leben; beim nächsten Tastendruck ist alles wieder beim alten.

```
0580  CD4F 3613 0B78 B1C8 CB78 C339 3579 1824   O6  x   x 95y $
0590  0000 0000 0000 0000 0000 0000 0000 0000
05A0  0000 0000 0000 0000 0000 0000 0000 0000
05B0  0000 0000 F5CD D105 20FB F132 E837 C900            2 7
05C0  0000 0000 0000 0000 0000 0000 0000 0000
05D0  003A E837 E6F0 FE30 C9E5 3E0E CD33 0048   : 7   0 >  3 H
```

Arnulf Sopp

```
                ;*************************************************************
                ;       Bildschirman- und -ausknipser zur Schonung der Bildröhre
                ;       Version für das Genie IIIs
                ;       Arnulf Sopp
                ;*************************************************************

4200                    ORG     4200h           ;Platz genug im Sektorpuffer
4200    2A1640  start   LD      HL,(4016h)      ;Treiberadresse im Testatur-DCB
4203    22A605          LD      (kbdrver),HL    ;in das Programm patchen
4206    21A505          LD      HL,restvid      ;Adresse der Umleitung
4209    221640          LD      (4016h),HL      ;in den DCB patchen
420C    C9              RET                     ;zurück ins DOS oder zu BASIC

4601                    ORG     4601h           ;Adr.-Operand eines CALLs in d. INT-Rout.
4601    9005            DW      cntvid          ;stattdessen die Umleitung CALLen

0590                    ORG     0590h           ;Platz in der gekürzten Druckerroutine
0590    21BE45  cntvid  LD      HL,45beh        ;Fortsetzung der INT-Routine
0593    E5              PUSH    HL              ;als RET-Adresse auf den Stack
0594    2AC305          LD      HL,(counter)    ;bisherigen Zählerstand laden
0597    23              INC     HL              ;Zähler erhöhen
0598    CB64            BIT     4,H             ;schon Zählerstand 1000h erreicht?
059A    CBA4            RES     4,H             ;nur bis 0FFFh zulassen
059C    F5              PUSH    AF              ;retten
059D    DBFA            IN      A,(0fah)        ;Systembyte 1 einlesen
059F    280E            JR      Z,exit1         ;weitermachen, falls noch nicht 1000h
05A1    E67F            AND     7fh             ;Port FAh, Bit 7 löschen (Bildschirm aus)
05A3    180A            JR      exit1           ;dort weiter

05A5    CD0000  restvid CALL    0000h           ;Tastaturtreiber aufrufen (Dummy-Adresse)
05A6            kbdrver EQU     $-2             ;Adreß-Operand
05A8    B7              OR      A               ;wurde eine Taste gedrückt?
05A9    C8              RET     Z               ;zurück, falls nein
05AA    F5              PUSH    AF              ;ASCII-Code retten
05AB    DBFA            IN      A,(0fah)        ;Systembyte 1 einlesen
05AD    F680            OR      80h             ;Bit 7 setzen (Bildschirm an)
05AF    22C305  exit1   LD      (counter),HL    ;Zähler neu laden
05B2    180B            JR      exit2           ;Teil des Druckertreibers überspringen

05BF                    ORG     05bfh           ;ab hier wieder Platz in der Druckerrout.
05BF    D3FA    exit2   OUT     (0fah),A        ;Systembyte 1 neu schreiben
05C1    F1              POP     AF              ;restaurieren
05C2    C9              RET

05C3    0000    counter DW      0000h           ;Platz für den Zähler

4200                    END     start

00000 Fehler

cntvid  0590    counter 05C3    exit1   05AF    exit2   05BF    kbdrver 05A6    restvid 05A5
start   4200
```


Futter für Genie III's CRTC

Im letzten Info beschrieb Arnulf Sopp eine Möglichkeit, den ##-Befehl des GDOS für die weiteren Bildformate A bis G aufzubohren. Darüber hinaus hoffte er auf ein Leser-Echo, wie dies wohl bei anderen TRS-kompatiblen als seinem Genie IIIs möglich wäre.

Hier ist ein Echo: Wer n u r ein Genie III an Stelle des 'S' besitzt und auch n u r ein GDOS 2.1 sein Eigen nennt, wird mit Arnulfs 3S-Patches wenig anfangen können, da hier die Sache ein wenig anders gelagert ist.

So residiert im GDOS 2.1 der #-Befehl zusammen mit dem LWT (Laufwerkstest) in SYS23/SYS; SYS26/SYS dagegen enthält die Treiberroutinen für das 64x16-Bildformat zur Restaurierung desselben mit dem Befehl "64". Um auch hier die zusätzlichen Formate zu implementieren, werden folgende Patches in SYS23/SYS notwendig:

```
         ;...
M5085:   LD   BC,16         ; Länge eines Parametersatzes
         LD   HL,M50B0      ; Adresse des ersten Parametersatzes
         CP   'X'           ; ##,X ?
         JR   Z,M50A7       ; Ja!
         ADD  HL,BC         ; Nein, Zeiger auf nächstes Format
         CP   'V'           ; ##,V ?
         JR   Z,M50A7       ; Ja!
         ADD  HL,BC         ; Nein, Zeiger auf nächstes Format
         CP   'H'           ; ##,H ?
         JR   Z,M50A7       ; Ja!
         ADD  HL,BC         ; Nein, Zeiger auf nächstes Format
         CP   'S'           ; ##,S ?
         JR   Z,M50A7       ; Ja!
         CP   'T'           ; ##,T ?
         JR   Z,M50F0       ; Ja, Bildseiten tauschen!

;Alt:

         LD   A,34H         ; Nein, mit Fehlercode "Schlechter Parameter"
         RET                ; zurück

;Neu:

         JP   M513D         ; Ansprung zur Erweiterung
```

```
M50A7:  LD   DE,M37F0          ; Ziel: CRTC-Kontrollblock
        LDIR                   ; Parametersatz transferieren
        XOR A
        JP   M37C0H            ; Daten an CRTC senden.

        ORG 513DH

M513D:  LD   HL,NEWTAB         ; HL: Neue Parametertabelle
        SUB 'A'                ; aus 'A' wird 0 ... aus 'G' wird 6
        JR   C,PARERR          ; kleiner als 'A' => Fehler
        CP   7                 ; größer als - korrigiertes! - 'G' ?
        JR   C,PAROK           ; Nein, Parameter O.K.

PARERR: LD   A,34H             ; Ursprüngliche Fehler-Ausgangs-
        OR   A                 ; Routine
        RET

PAROK:  OR   A                 ; Format "A", Zeiger stimmt bereits =>
PARLOP: JP   Z,M50A7           ; Daten an CRTC senden.
        DEC A                  ; Formatbezeichnung weiter-"scannen"
        ADD HL,BC              ; Formatzeiger weiterstellen.
        JR   PARLOP            ; Irgendwann zwischen A und G funkticklappts!

        DM   'XXXXXXXX'        ; Füller-Bytes für Beginn der Parameter-
                               ; Tabellen auf einer 16-Byte-Grenze

NEWTAB  EQU $
```

Leider ist SYS23/SYS derart ungünstig aufgebaut, daß ein Patch via Superzap, DDE oder ähnlichen Hilfsprogrammen nicht so einfach durchführbar ist:SYS23/SYS in der Version vom 1.12.1985 scheint nämlich häufiger mit einem Monitorprogramm überarbeitet worden sein, da hier und dort etliche NOPs eingefügt sind. So zeigt der Sektor-Dump (Superzap-Format) des SYS23/SYS neben anderen solcher ausge-NOPter Bereiche einen Freibereich von 101 Bytes ab 511BH gefolgt von einer Unterroutine für den Laufwerkstest, dem weitere 58 freie Bytesfolgen. Um nun einen - für den Patch nötigen - zusammenhängenden Freibereich zuerzeugen, erscheint es am günstigsten, SYS23/SYS erneut zu assemblieren. Da Arnulf Sopp und ich es einmal planten, ein "GDOS decoded and other mysteries"zu erstellen, hatte ich es als glücklicher Besitzer einer reassemblierfähigen SYS23/SYS-Quellversion recht einfach, diesen Patch durchzuführen, wie das gezeigte Assembler-Listing demonstriert. Nichtsdestotrotz möchte ich hier versuchen, die dieser reassemblierten Version zugrunde liegenden Patches im Superzap-Format zu präsentieren:

Ursprüngliche Version des SYS23/SYS im GDOS 2.1:

```
DRV 00  0102 004D FEF9 2030 0D28 710D CA63 500D ...M...0.(q..cP.
0   10  3EFD 2006 EF3E 5818 0800 0D20 183D EF3E >....>X......=.>
0H  20  48CD 8550 CDC9 01AF C900 0000 0000 0000 H..P............
    30  0000 0000 0000 0000 3E2A C900 0000 0000 ........>*......
DRS 40  0000 0000 3E01 D3F6 3E40 D3F7 3E00 D3F5 ....>...>§..>...
137050  C900 0000 3E01 D3F6 3E50 D3F7 AFC9 0000 ....>...>P......
55AH60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  0000 0000 ED7B 6E4D AFC3 C037 CD40 4D00 .....änM...7.§M.
    80  ED73 6E4D F316 0031 5F50 D5CD C901 210D .snM...1_P....!.
    90  4E11 803E 0180 00ED B021 8D4E 1140 3C01 N..>.....!.N.§<.
    A0  8001 EDB0 D121 EC37 3A40 38FE 0420 06CD .....!.7:§8.....
    B0  C901 FB18 BFCD E94D 0100 00CD E94D E52A .......M.....M.* CALL des
FRS C0  0D50 3E20 7721 586F AFED 42CD 8051 3E3F .P>.w!Xo..B..Q>? LWT-UPs
0   D0  BC38 0E20 07CB 3DCB 3D00 1808 2100 3F18 .8....=.=...!.?.
0H  E0  0321 3F3F 3E2A 7722 0D50 E118 BB3A 1038 .!??>*w".P...:.8
    F0  B720 1503 CB4E 28F5 3A10 38B7 200A 03CB .....N(.:.8.....


DRV 00  B0B0 B0B0 B0B0 B0B0 B0B0 B0B0 0102 0050 ...............P
0   10  B0B0 B0B0 B0B0 B0B0 B0B0 B0B0 BFC0 3E00 ..............>.
0H  20  0000 0000 0000 0000 0000 0000 0000 0000 ................
    30  0000 0000 0000 0000 0000 0000 0000 0000 ................
DRS 40  0000 0000 0000 0000 0000 0000 0000 0000 ................
137350  0000 0000 0000 0000 0000 0000 0000 0000 ................
55DH60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  CAC8 CCF3 CDD5 4CCA 0035 7EFE 24CA 3435 ......L..5ß.$.45
    80  FE4E CAC0 3721 FC37 FE30 3809 FE38 3005 .N..7!.7.08..80.
    90  D630 7718 2701 1000 21B0 50FE 5828 1809 .0w.'...!.P.X(..
    A0  FE56 2813 09FE 4828 0E09 FE53 2809 09FE .V(...H(...S(...
    B0  5428 4D3E 34C9 0011 F037 EDB0 AFC3 C037 T(M>4....7.....7 Fehler-
FRS C0  6350 522E 1800 1818 020B 2B0B 0000 0000 cPR.......+..... Ausgang
3   D0  4E40 3D23 2302 2023 020B 2B0B 0400 0000 N§=##..#..+.....
3H  E0  5F40 4A4E 190C 1014 020B 2B0B 0000 0000 _§JK......+.....
    F0  6840 4E22 1900 1818 020B 2B0B 0700 0000 h§N"......+.....
```

```
DRV 00  ED73 6E4D 315F 503E 09D3 FA01 0004 2100 .snM1_P>......!.
0   10  01EA 0051 3C11 0040 7EF5 1A77 F112 0B13 ...Q<..§ß..w....
0H  20  2378 B120 F33E D5D3 FAED 7B6E 4DAF C900 #x...>....änM... Der Teil
    30  0000 0000 0000 0000 0000 0000 0000 0000 ................ des Frei-
DRS 40  0000 0000 0000 0000 0000 0000 0000 0000 ................ raums, in
137450  0000 0000 0000 0000 0000 0000 0000 0000 ................ den das
55EH60  0000 0000 0000 0000 0000 0000 0000 0000 ................ LWT-UP ver-
    70  0000 0000 0000 0000 0000 0000 0000 0000 ................ schoben wird!
    80  0000 0000 0000 0000 0000 0000 0000 0000 .......,........
    90  0000 0000 C5F5 E5D5 E5ED 5BC2 4D21 943C ..........Ä.M!.<  LWT-UP
    A0  CD63 40E1 017D 3FED 42E5 D121 AE3C CD63 .c§..ü?.B..!.<.c  (alte
    B0  40D1 E1F1 C1C9 0000 0000 0000 0000 0000 §...............   Lage)
FRS C0  0000 0000 0000 0000 0000 0000 0000 0000 ................
4   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
4H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  6A68 6C20 312E 3132 2E38 3200 0202 004D jhl.1.12.82....M
```

Änderungen in SYS23/SYS:

```
DRV 00  0102 004D FEF9 2030 0D28 710D CA63 500D ...M...0.(q..cP.
0   10  3EFD 2006 EF3E 5818 0800 0D20 1B3D EF3E >....>X......=.>
0H  20  48CD B550 CDC9 01AF C900 0000 0000 0000 H..P............
    30  0000 0000 0000 0000 3E2A C900 0000 0000 ........>*......
DRS 40  0000 0000 3E01 D3F6 3E40 D3F7 3E00 D3F5 ....>...>§..>...
137050  C900 0000 3E01 D3F6 3E50 D3F7 AFC9 0000 ....>...>P......
55AH60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  0000 0000 ED7B 6E4D AFC3 C037 CD40 4D00 .....änM...7.§M.
    80  ED73 6E4D F316 0031 5F50 D5CD C901 210D .snM...1_P....!.
    90  4E11 803E 0180 00ED B021 8D4E 1140 3C01 N..>.....!.N.§<.
    A0  8001 EDB0 D121 EC37 3A40 38FE 0420 06CD .....!.7:§8.....  Veränderter
    B0 ¬C901 FB18 BFCD E94D 0100 00CD E94D E52A .......M.....M.*  CALL des
FRS C0  0D50 3E20 7721 586F AFED 42CD 1B51 3E3F .P>.w!Xo..B..Q>?  LWT-UP's
0   D0  BC38 0E20 07CB 3DCB 3D00 1808 2100 3F18 .8....=.=...!.?.
0H  E0  0321 3F3F 3E2A 7722 0D50 E118 BB3A 1038 .!??>*w".P...:.8
    F0  B720 1503 CB4E 28F5 3A10 38B7 200A 03CB .....N(.:.8.....
```



```
DRV 00  B0B0 B0B0 B0B0 B0B0 B0B0 B0B0 0102 0050 ...............P
0   10  B0B0 B0B0 B0B0 B0B0 B0B0 B0B0 BFC0 3E00 ..............>.
0H  20  0000 0000 0000 0000 0000 0000 0000 0000 ................
    30  0000 0000 0000 0000 0000 0000 0000 0000 ................
DRS 40  0000 0000 0000 0000 0000 0000 0000 0000 ................
137350  0000 0000 0000 0000 0000 0000 0000 0000 ................
55DH60  0000 0000 0000 0000 0000 0000 0000 0000 ................
    70  CAC8 CCF3 CDD5 4CCA 0035 7EFE 24CA 3435 ......L..5ß.$.45
    80  FE4E CAC0 3721 FC37 FE30 3809 FE38 3005 .N..7!.7.08..80.
    90  D630 7718 2701 1000 21B0 50FE 5828 1809 .0w.'...!.P.X(..
    A0  FE56 2813 09FE 4828 0E09 FE53 2809 09FE .V(...H(...S(...
    B0  5428 4DC3 3D51 0011 F037 EDB0 AFC3 C037 T(M.=Q...7.....7 Ansprung
FRS C0  6350 522E 1800 1818 020B 2B0B 0000 0000 cPR.......+..... der Erweiter-
3   D0  4E40 3D23 2302 2023 020B 2B0B 0400 0000 N§=##..#..+..... ung statt
3H  E0  5F40 4A4B 190C 1014 020B 2B0B 0000 0000 _§JK......+..... Fehlerausgang
    F0  6840 4E22 1900 1818 020B 2B0B 0700 0000 h§N"......+.....

DRV 00  ED73 6E4D 315F 503E 09D3 FA01 0004 2100 .snM1_P>......!.
0   10  01EA 0051 3C11 0040 7EF5 1A77 F112 0B13 ...Q<..§ß..w....
0H  20  2378 B120 F33E 05D3 FAED 7B6E 4DAF C9C5 #x...>....änM... LWT-UP
    30  F5E5 D5E5 ED5B C24D 2194 3CCD 6340 E101 .....Ä.M!.<.c§.. jetzt hier
DRS 40  7D3F ED42 E5D1 21AE 3CCD 6340 D1E1 F1C1 ü?.B..!.<.c§.... im urspr.
137450  C9 ...                                                   Freiraum
```

Erweiterung für die Formate A bis G:

```
        ... 21 5C51 D641 3804 FE07 3804 3E34 B7C9 .!öQ.A8...8.>4..
55EH60  B7CA A750 093D 18F9 5858 5858 5858 5858 ...P.=..XXXXXXXX <- Füller
    70  4141 4141 4141 4141 4141 4141 4141 4141 AAAAAAAAAAAAAAAA <- Formate
    80  4242 4242 4242 4242 4242 4242 4242 4242 BBBBBBBBBBBBBBBB <- für
    90  4343 4343 4343 4343 4343 4343 4343 4343 CCCCCCCCCCCCCCCC <- ##,A
    A0  4444 4444 4444 4444 4444 4444 4444 4444 DDDDDDDDDDDDDDDD <- bis
    B0  4545 4545 4545 4545 4545 4545 4545 4545 EEEEEEEEEEEEEEEE <- ##,G
FRS C0  4646 4646 4646 4646 4646 4646 4646 4646 FFFFFFFFFFFFFFFF
4   D0  4747 4747 4747 4747 4747 4747 4747 4747 GGGGGGGGGGGGGGGG
4H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  6A68 6C20 312E 3132 2E38 3200 0202 004D jhl.1.12.82....M
```

Ulrich Heidenreich
Werderstr. 25
4300 Essen 1

* * C A L L E R L E I * *

Liebe Freunde aus dem Lande, in dem man die Maschinensprache spricht!
Und all ihr übrigen, die sich nicht mit den Umständlichkeiten des BASIC begnügen wollen! Die ihr bereits ahnt, daß im "Nur-lesespeicher", auch kurz ROM genannt, so manches Handwerkzeug bereitliegt, auf das kein "USER MANUAL" hinweist, weshalb letzteres auch Handbuch genannt wird, weil man beim Lesen oft die Hände ringt und weil der Kopf dabei häufig leer bleibt (treffenderweise deshalb meist "Lehrbuch" genannt) –
ihr unverdrossenen fröhlichen Programmierer, die ihr einen *RÖCKRATH* zu schätzen wißt, weil er euch den unentbehrlichen *ROM-Rath* erteilt – die ihr seine profunden Publikationen nun schon zum dritten Mal gekauft habt, weil das kostbare Werk vom pausenlosen Hin- und Herblättern schon so zerfleddert ist, daß es auseinanderfällt, als hätte einer darin verzweifelt nach seinem letzten 5-DM-Schein gesucht (den es gar nicht mehr gibt) –
statt nützlicher Routinen
sieht man nur noch Ruinen...

ihr könnt aufatmen!
Denn nun sind die wichtigsten, meistgebrauchten Routinen, besser: Arbeitshilfen, herausgezogen und in knappster Form zusammengestellt! Ihr braucht nur noch Allerlei zu *"CALLen!"*
* SUPER machte es möglich! * SUPER, die gemächlichste, aber flexibelste aller Datenbänke – aua! – Datenbanken, macht's auch möglich, daß jede Lücke, jeder Fehler, die/den ihr entdeckt (und das sind sicher viele!), auf schnellstem Wege nachgefüllt, nachgebessert werden kann. Sofern ihr sie mir meldet. Denn ich bin kein ROM-Experte, und ich überlasse euch auch die gesamte Verantwortung bei der Benutzung dieser Liste, die ich aus *RÖCKRATH*'s ROM-Listing nur ganz knapp extrahiert habe, um mir die ständige Such- und Nachschlagearbeit zu ersparen.

<u>Cum grano salis:</u>
Wer *R-R's* ausführliches ROM-Listing nicht vorher gründlich studiert hat, wird wohl kaum in allen Fällen mit diesem Extrakt zurechtkommen! Drum:

LABORA CUM IRA ET STUDIO! JaJot

Korrekturmöglichkeiten


```
Name   : (ohne)                         Einsprungadr./HEX 1936
         FOR/NEXT und GOSUB/RET. (Holt Daten v.Stack zur.)
INPUT  :
OUTPUT:
Veraend.Reg.

Name   : ASC                            Einsprungadr./HEX 2A0F
         Ergibt ASCII-Code des 1. Zeichens eines Strings
INPUT  :
OUTPUT:
Veraend.Reg.

Name   : BASIC                          Einsprungadr./HEX 06CC
         BASIC-Warmstart
INPUT  :
OUTPUT:
Veraend.Reg.

Name   : BLINK                          Einsprungadr./HEX 022C
         schaltet Stern in Bildschirmecke um
INPUT  :
OUTPUT:
Veraend.Reg. AF

Name   : CALL                           Einsprungadr./HEX 03C2
         ruft I/O-Routinen ueber DCB auf
INPUT  : BC muss auf Stack sein (!) - B = DCB-Typ
OUTPUT: (s.jeweilige Parameter u. Registerbeeinflussungen)
Veraend.Reg.  -

Name   : CHECK                          Einsprungadr./HEX 1C96
         Syntax-Check (RST 08)
INPUT  : HL=Pointer; Byte nach RST 08
OUTPUT: (HL)=Byte->RST08; HL->naechst.Zeich.,sonst SN-Err.
Veraend.Reg. AF, HL=P

Name   : CHR$                           Einsprungadr./HEX 2A1F
         erzeugt 1-Byte-String m.Zeichen; ASCII = Argument
INPUT  :
OUTPUT:
Veraend.Reg.

Name   : CHRGET                         Einsprungadr./HEX 1D78
         sucht nchst.sign.Zchn.u.gibt Info darueber (RST10)
INPUT  :   -
OUTPUT: A = Zeichen; Ziffer: CY=1; ':' od. 00: Z=1
Veraend.Reg. AF, HL=P

Name   : CLS                            Einsprungadr./HEX 01C9
         Bildschirm loeschen (CLS-Anweisung)
INPUT  :
OUTPUT:
Veraend.Reg. A=1F, F
```

```
Name   : CPHLDE                     Einsprungadr./HEX 1C90
         vergleicht HL und DE
INPUT  : HL; DE: 16BIT Integer ohne Vorzeichen
OUTPUT: HL>DE: Z=0,CY=0; =DE: Z=1,CY=0,A=0; <DE: Z=0,CY=1
Veraend.Reg. AF


Name   : CR                         Einsprungadr./HEX 20FE
         CR ausgeben
INPUT  : (409C)=Ausgabe-Flag
OUTPUT:    -
Veraend.Reg. AF=0044


Name   : CURPOS                     Einsprungadr./HEX 0348
         ermittelt Cursor-Position in der Zeile
INPUT  : (4020) = Cursor-Adresse, (4030) = Port-Status
OUTPUT: A = Cursor-Position
Veraend.Reg. AF


Name   : DELAY                      Einsprungadr./HEX 0060
         Zeitschleife oder Warteschleife
INPUT  : BC bestimmt Zeitdauer (14.66 microsec x BC)
OUTPUT:
Veraend.Reg. AF=0044, BC=0000


Name   : DVAL                       Einsprungadr./HEX 0E65
         wandelt String in Zahl (Double) um
INPUT  : HL = Zeiger auf String
OUTPUT: X = Zahl
Veraend.Reg. AF, BC, DE, HL=P


Name   : FILLO                      Einsprungadr./HEX 1269
         Nullen in Puffer schreiben
INPUT  : A = Anzahl Nullen; HL = Pufferzeiger
OUTPUT: HL=Pufferzeiger, um Anzahl der Nullen erhoeht
Veraend.Reg. AF=0044, HL


Name   : HERE                       Einsprungadr./HEX 000B
         ermittelt Stand des PC
INPUT  :
OUTPUT: HL= Zeiger auf dem Aufruf folgendes Byte
Veraend.Reg. HL


Name   : ILDXA                      Einsprungadr./HEX 27F8
         Zahl als Integer (ohne Sign) nach X
INPUT  : A=Zahl
OUTPUT: HL, X=Zahl
Veraend.Reg. AF=0044, HL


Name   : INBUFF                     Einsprungadr./HEX 05D9
         Eingabe einer Zeile in den Puffer
INPUT  : B=max.Anzahl einzugebender Zeichen, HL=Puff.Anfang
OUTPUT: B=Anz.d.tatsaechl.eingegebenen, C=max.Anz. Zeichen
Veraend.Reg. AF, BC, DE=401D
```


ROM-Routinen des LEVEL 2 BASIC-Interpreters

```
Name   : INCH                       Einsprungadr./HEX 002B
         Tastaturabfrage
INPUT  :
OUTPUT: A=ASCII-Code neu gedrueckter Taste - oder 0
Veraend.Reg. AF, DE=4015


Name   : INCH2                      Einsprungadr./HEX 0358
         Tastaturabfrage (s. INCH)
INPUT  :
OUTPUT: ASCII-Code gedrueckter Taste oder 0
Veraend.Reg. AF


Name   : INCHW                      Einsprungadr./HEX 0049
         Tastaturabfrage (wartet, bis eine Taste gedrueckt)
INPUT  :
OUTPUT: A=ASCI-Code neu gedrueckter Taste
Veraend.Reg. AF, DE=4015


Name   : INCHW2                     Einsprungadr./HEX 0384
         Tastaturabfrage (wartet, bis Taste gedrueckt)
INPUT  :
OUTPUT: A=ASCII-Code gedrueckter Taste
Veraend.Reg. AF


Name   : INKEY                      Einsprungadr./HEX 019D
         Tastaturabfrage ohne Programmstop
INPUT  :
OUTPUT:
Veraend.Reg.
= RST 10


Name   : INLINA                     Einsprungadr./HEX 1BB3
         gibt '?' aus und erwartet Eingabe einer Zeile
INPUT  :
OUTPUT: HL = I/O-Pufferanfang-1; F: CY=1, falls BREAK
Veraend.Reg. AF, DE=4010, HL


Name   : INLINE                     Einsprungadr./HEX 0361
         liest Zeile von Tastatur in I/O-Puffer
INPUT  :
OUTPUT: HL=I/O-Pufferanfang - 1
Veraend.Reg. AF, DE=401D, HL


Name   : INP                        Einsprungadr./HEX 2AEF
         Eingabe-Port lesen
INPUT  :
OUTPUT:
Veraend.Reg.


Name   : LEFT$                      Einsprungadr./HEX 2A61
         linken Teil eines Strings abtrennen
INPUT  :
OUTPUT:
Veraend.Reg.
```

Name : LEN Einsprungadr./HEX 2A03
ergibt Laenge eines Strings
INPUT :
OUTPUT:
Veraend.Reg.

Name : LETTER Einsprungadr./HEX 1E3D
testet, ob Zeichen ein Buchstabe
INPUT : HL = Zeiger auf zu untersuchendes Zeichen
OUTPUT: wenn Buchstabe: CY=0, sonst CY=1
Veraend.Reg. AF

Name : LIST Einsprungadr./HEX 2B2E
Programm auf Bildschirm auslisten (listen)
INPUT :
OUTPUT:
Veraend.Reg.

Name : LLIST Einsprungadr./HEX 2B29
Programm auf Drucker auslisten (llisten)
INPUT :
OUTPUT:
Veraend.Reg.

Name : MEM Einsprungadr./HEX 27C9
ergibt Groesse des freien Speichers
INPUT :
OUTPUT:
Veraend.Reg. A

Name : MID$ Einsprungadr./HEX 2A9A
mittleren Teil eines Strings abtrennen
INPUT :
OUTPUT:
Veraend.Reg.

Name : MROOM Einsprungadr./HEX 28BF
Platz fuer String in Stringspace machen
INPUT : A=Stringlaenge
OUTPUT: DE = Stringspace-Adresse
Veraend.Reg. F, BC, DE, HL

Name : NEXTL Einsprungadr./HEX 20F9
CR ausgeben, wenn Cursor nicht in Pos.0
INPUT : (409C)=Ausgabe-Flag; (40A6)=Cursor-Pos.
OUTPUT:
Veraend.Reg. AF=0044

Name : OUT Einsprungadr./HEX 2AFB
Wert auf Ausgabe-Port ausgeben
INPUT :
OUTPUT:
Veraend.Reg.

Name : OUTCH Einsprungadr./HEX 0033
Ausgabe eines Zeichens auf den Bildschirm
INPUT : A=ASCII-Code des auszugebenden Zeichens
OUTPUT:
Veraend.Reg. F, DE=401D

Name : OUTCH2 Einsprungadr./HEX 033A
gibt Zeichen auf den Bildschirm aus
INPUT : A=auszugebendes Zeichen
OUTPUT: (40A6) = Cursor-Position nachher
Veraend.Reg. F

Name : OUTCHF Einsprungadr./HEX 032A
gibt Zeichen auf Bildschirm, Drucker od.Kass. aus
INPUT : A=auszugebendes Zeichen. (409C)=Ausgabeflag, s.u.
OUTPUT:
Veraend.Reg. F
Ausgabeflag=0-->Bildschirm, 1-->Drucker, 80->Kass.

Name : POS Einsprungadr./HEX 27F5
ergibt Cursor-Position
INPUT :
OUTPUT:
Veraend.Reg. A

Name : PRCR Einsprungadr./HEX 0394
CR auf Drucker ausgeben
INPUT : A=auszugebendes Zeichen; (409B)=Druckkopfposition
OUTPUT: (409B) = Druckkopfposition nachher
Veraend.Reg. A=0D

Name : PRINT Einsprungadr./HEX 003B
Ausgabe eines Zeichens auf den Drucker
INPUT : A=ASCII-Code des auszugebenden Zeichens
OUTPUT: -
Veraend.Reg. AF, DE=4025

Name : PRINT1 Einsprungadr./HEX 039C
Zeichen auf Drucker ausgeben
INPUT : A=auszugebendes Zeichen; (409B)=Druckkopfposition
OUTPUT: (409B)=Druckkopfposition nacher
Veraend.Reg.

Name : PRINTS Einsprungadr./HEX 28A6
druckt einen String
INPUT : (HL+1)...=String (durch '"' oder 0 bendet)
OUTPUT:
Veraend.Reg. AF, BC, D=0, E, HL=P

Name : PRRDY Einsprungadr./HEX 05D1
testet, ob Drucker bereit
INPUT : -
OUTPUT: Z-1, wenn Drucker bereit
Veraend.Reg. AF

Name : RESET Einsprungadr./HEX 0000
System-Initialisierung
INPUT : -
OUTPUT: -
Veraend.Reg.

Name : RIGHT$ Einsprungadr./HEX 2A91
rechten Teil eines Strings abtrennen
INPUT :
OUTPUT:
Veraend.Reg.

Name : RND Einsprungadr./HEX 14C9
Zufallszahl erzeugen
INPUT : X=Dummy-Argument (X=0) oder Intervall-Ende (X > 0)
OUTPUT: X = Zufallszahl
Veraend.Reg. AF, BC, DE, HL

Name : SADDA Einsprungadr./HEX 0F89
Summe bilden (summieren)
INPUT : X=Summand; A=Summand (signed 8-Bit-Integer)
OUTPUT: X = Summe
Veraend.Reg. AF, BC, DE, HL

Name : SEOL Einsprungadr./HEX 1F07
ELSE-Anweisung: sucht Ende der Zeile
INPUT :
OUTPUT:
Veraend.Reg. AF, BC, D, HL=P
Einsprung 1F09, wenn Trennzeichen2 = 0

Name : SEOS Einsprungadr./HEX 1F05
DATA-Anweisung: sucht Ende der Anweisung
INPUT :
OUTPUT:
Veraend.Reg. AF, BC, D, HL=P

Name : STRING$ Einsprungadr./HEX 2A2F
erzeugt String aus gleichen Zeichen
INPUT :
OUTPUT:
Veraend.Reg.

Name : TYPCHK Einsprungadr./HEX 25D9
testet Typ des X-Registerinhaltes (RST 20)
INPUT : (40AF)=Typ X
OUTPUT: A=TypX-3; Int.:S=P=C=1; Single:C=1; Double: P=1
Veraend.Reg. AF

Name : USR Einsprungadr./HEX 40BE
User-Startadresse
INPUT :
OUTPUT:
Veraend.Reg.
FC-ERROR

Name : VAL Einsprungadr./HEX 2AC5
String in Zahl umwandeln
INPUT :
OUTPUT:
Veraend.Reg. keine

Und Sie behaupten, Ihr Computer kann meine Sekretärin ersetzen . . .

* ...weil Kürze so würzig ist... *

(Mein längstes Assembler-Programm)

– es hat ganze 6 (sechs!) Zeilen – und ich bin besonders stolz darauf! Aus zwei Gründen:

1) Meine bisherigen Programme besaßen 0 (Zero oder Null) Zeilen; es ist also das längste. (Die früher veröffentlichten 3 Programme zählen nicht, denn sie waren nur Lehr- oder, wenn ihr wollt, 'Leer'-stücke, da sie zu nichts weiter nütze waren;

2) Es füllt das erstemal eine erhebliche Lücke in unserem DOS aus und ist – mir jedenfalls – sehr viel nütze, wie ihr gleich erfahren werdet.

Jedesmal wenn ich vor dem Ausdrucken eines Directory (übrigens, wie mein dickes Dickschonäri mich neulich belärte, spricht sich dies nicht "Dairektorri", sondern zufällig so wie man's schreibt: "Dihrecktori" aus!), oder vor der Anwendung von JKL oder sonstwelchen Aktionen *im DOS* eine überschrift, einen Hinweis oder sonstwas eben schnell mal ausdrucken wollte, ärgerte es mich, daß ich erst BASIC aufrufen, dann LPRINT schreiben, dahinter meinen Text setzen und anschließend mit CMD"S" wieder ins DOS zurückgehen mußte, bevor ich meine eigentliche Aktion durchführen konnte! Wie umständlich!!

Das ist nun endgültig vorbei. Dank meines Sechszeilers.

Zwar hat unser (Clubbe-)treuer Hartmut selbst eine DOS-Druckroutine ins SYS28/SYS eingebaut (s. seinen Beitrag im INFO unter der überschrift "EXTENDED NEWDOS 80" sowie sein Betriebssystem "NEWDOS80+10"), die auf den Namen DR hört und als Library-Erweiterung eben diesem Befehl "DR" ziemlich schnell gehorcht; jedoch ist mir der freie Platz in den SYS-Files zu wertvoll für diese selten benötigte (s.o.) Routine! Aus der Not, in die ich beim Versuch, sie ins SYS-File ← seines anderen DOS einzubauen, geriet, wurde die Tugend, ein eigenes Maschinenprogramm zu "entwickeln"! Wie gesagt: Es ist mein erstes nützliches, und daher sowohl mein bestes als auch mein längstes...

Meine kleine Ruthine (ja, so heißt meine neue Freundin) hat noch einen Vorteil: Wenn man sich in dem unentbehrlichen Modus "DIS" befindet, muß man diesen erst verlassen, wenn man Library-Befehle aufrufen will wie "PDRIVE n", "SYSTEM n", "DIRSORT n" o.a. Das ist bei einem SYS-freien .../CMD-File nicht nötig. Ich gehe also mit dem Cursor über die "Ruth" ('der darf dat') und schlage zu (ENTER); danach gebe ich gleich den Text ein, den ich ausdrucken möchte. Es dürfen alle Zeichen sein, auch Blanks und sogenannte Steuerzeichen. Und das in beliebiger Länge! (Nicht wie bei ähnlichen Routinen begrenzt auf 80 oder weniger Zeichen.) * Anschließend bleibt das System im DOS.

Wer Angst hat vor Maschinensprache spr. vor dem Programmieren in Assembler, der sei mit nachstehender kurzer Erläuterung des ebenso kurzen Prográmmchens getröstet:

"Faulheit" ist das oberste Gebot beim Programmieren!

Machen wir's uns bequem und überlassen das Denken den Herren in ROM! Hier im ROM liegt alles griffbereit:
a) eine Routine (auch UP = Unterprogramm genannt), die begierig

darauf wartet, Zeichen von der Tastatur entgegenzunehmen (Speicherplatz 49H = 73d);
b) eine Routine, die das so Empfangene auf den Bildschirm gibt (Speicherplatz 33H = 51d);
c) eine Routine, die es in den Druckerpuffer gibt (Speicherplatz 3BH = 59d);

Das ist bereits das Kernstück des "Programms". Jetzt muß man nur noch dafür sorgen, daß der Zeichenpuffer auch ausgegeben wird, denn das tut er erst, wenn er entweder eine Druckzeilenlänge enthält oder der Befehl "NUN ABER SCHLUSS!" erteilt wird (ODH = 13d). Das be- sorgen die beiden nächsten Zeilen. Solange dieser Befehl nicht erfolgt, wartet das Prográmmle immer wieder auf ein weiteres Zeichen von der Tastatur (s.Rücksprung in der letzten, der 6.Zeile).

Die allerletzte sowie die allererste Zeile erwähne und zähle ich nicht, weil sie Eigentümlichkeiten eines j e d e n Assemblerprogramms sind, die ich den Experten abgeguckt habe...

Nun zum Schluß noch eine wichtige Anmerkung über eine Feststellung, die mich in allergrößtes Staunen versetzt hat:
Das assemblierte Programm (siehe Listing) meldet ganz zum Schluß, daß diese 6 resp. 8 Zeilen "00000 TOTAL ERRORS" enthalten! Was für phänomenale Möglichkeiten ich da verpaßt habe! Es ist mir trotz größter Dummheit nicht gelungen, auch nur einen (00001) "ERROR" hineinzubringen – geschweige denn eine fünfstellige Menge...

HALLO Leute vom CLUB 80!
dieser Programmierer macht sich!
Gießet ruhig euren Spott
über'n Anfänger: Kalott!

* * * "D O S D R U C K" * * *

```
5200              00100          ORG     5200H
5200 CD4900       00110 START    CALL    49H       ;ZEICHEN-EINGABE
5203 CD3300       00120          CALL    33H       ;BILDSCHIRM-AUSGABE
5206 CD3B00       00130          CALL    3BH       ;ZEICHEN ZUM DRUCKERPUFFER
5209 FEOD         00140          CP      ODH       ;WAR ES ENTER?
520B CA2D40       00150          JP      Z,402DH   ;DANN DRUCKEN UND INS DOS
520E 18F0         00160          JR      START     ;HOLE NAECHSTES ZEICHEN
5200              00170          END     START
00000 TOTAL ERRORS (kaum zu glauben)
34831 TEXT AREA BYTES LEFT

START  5200 00110    00160 00170
```

[diese zusätzliche Information ist ein völlig überflüssiges Geschenk von EDTASM!]

K-8

Kleinschrift für den EDTASM/PLUS

Eigentlich hat ja Arnulf Sopp den Stein ins Rollen gebracht, als er neulich seinen Zeus-Assembler so in den Himmel lobte, um gleichzeitig den guten alten EDTASM/PLUS schlechtzumachen: Ja, wenn er wenigstens Kleinschrift beherrsche, der EdPlus, dann wäre er gar nicht mal so übel! Aber so nerve das ständige Hin- und Herschalten zwischen Groß- (für Mnemonics) und Kleinschrift (für Kommentare) doch.

Wer nun ein alter Hacker ist und zufälligerweise sowieso im EdPlus herumstochert, um ihm das 80x24-Format schmackhaft zu machen, läßt natürlich einen solchen Vorwurf nicht unbeantwortet, zumal Arnulf mit dem Kleinschriftproblem gar nicht mal so unrecht hat. Und schwerer als der 80er Patch kann die Operation wohl auch nicht werden: Ein paar "AND 0DFH"s an die richtigen Stellen geschummelt und schon versteht der EdPlus auch das kleine Alfabet.

Gesagt, getan: Man nehme den 1-2-3-Debugger, einen Disassembler und n' Flasch' Bier dazu, suche die neuralgischen Punkte im EdPlus und erlebe dann seine erste "einfach-ein-AND-0DFH-Patchen"-Ernüchterung:

```
M75E1:  LD      A,(DE)      ; Ein Zeichen aus der Befehlstabelle holen
        RLA                 ; Bit-7-Endemarkierung ins Carry schieben
        PUSH    AF          ; das Carry retten
        OR      A           ; das Carry löschen
        RRA                 ; RLA rückgängig machen
        CP      (HL)        ; Mit dem Zeichen im Quelltext vergleichen
        INC     DE          ; Befehlstabellenzeigerweiterstellung
        JR      NZ,M75F7    ; ungleich, nächsten Befehl vergleichen
        DEC     C           ; Befehlslängenzähler (wo kommt der her?)
                            ; herunterzählen.
        JR      Z,M75FF     ; Zero => Befehl komplett verglichen!
        POP     AF          ; Befehls-Ende=Carry-Flag herholen
        INC     HL          ; Quelltextzeigerweiterstellung
        JR      NC,M75E1    ; Weiter vergleichen
```

Maskieren Sie, geneigter Leser, hier einmal mit einem einfachen Patch die Kleinschrift heraus! Auch an weiteren Adressen schlägt Murphy mit solchen - eine einfache Maskierung so erschwerenden - Vergleichsbefehlen zu. Hinzu kommt zudem, daß ein und dasselbe "CP (HL)" teils in (HL), teils im Akku das Zeichen aus dem Quelltext übergeben bekommt. Wo maskiert man dann das Kleinschrift-Bit-5 heraus: aus A, aus (HL), gar nicht, immer, ... ?

Nach vielen unzähligen mißlungenen Versuchen, diesen In-welcher-Richtung-auch-immer-Vergleich immun gegen Kleinschriftunterschiede zu machen (unter anderem auch mit der zunächst genial erscheinenden Lösung "XOR (HL)" und "AND 0DFH") blieb nichts anderes übrig, als doch etwas mehr Aufwand zu treiben:

Statt eines einfachen Patches entschied ich mich schweren Herzens, sowohl A als auch (HL) getrennt auf Kleinschrift zu untersuchen, um dann je nach Bedarf nur (HL), nur A, keins von beiden oder auch beide - letzeres nötig bei der Verwaltung der Symbol-Tabelle! - der 0DFH-Maske zu unterziehen: Das Ergebnis findet sich im EDPATCH-Listing als Unterprogramm LOCOMP.

Nun warf sich aber die Frage auf, wie diese 1-Byte-Befehle sinnvoll durch entsprechende Aufrufe zum Unterprogramm LOCOMP zu ersetzen wären. Das Standard-Patch-Vorgehen, einfach drei passende Bytes mit einem CALL LOCOMP zu überschreiben und diese drei Bytes dann im Unterprogramm "nachzuholen", verbot sich aus einfachem Grund: Zunächst schien eine unüberschaubare Anzahl CP (HL)s dem Patch zu harren, was dann auch eine ebensolche Anzahl LOCOMP-Restaurations-Byte-Folgen notwendig gemacht hätte.

Erst eine uralte Erkenntnis über den EdPlus - damals entdeckt vom c't-Autor Heinz-Peter Heidinger - brachte d i e Lösung: Der EdPlus "verbiegt" einige RST-Vektoren. Warum also nicht dem guten Beispiel folgen und den weder von ihm noch außerhalb von BASIC benötigten RST 8 zum LOCOMP-Patch verwenden? Daß im Endeffekt dann doch nur drei 0BEHs durch 0CFHs zu ersetzen waren, rechtfertigt meiner Meinung nach aber nicht, im Nachhinein auf diese elegante Lösung zu verzichten: Als Schulbeispiel für den Fall, daß wirklich mal nur je ein Byte an -zig Patch-Stellen zur Verfügung steht, kann diese Methode nämlich allemal dienen!

Neben diesen verbleibenden drei Vergleichsbefehlen, nämlich an 62DDH zur Erkennung der Assembler-Direktbefehle, an 6E69H für "all purpose" und an 75E6H in der oben beschriebenen Mnemonic-Erkennung waren dann weitere Eingriffe nötig, die ich mangels Phantasie (?) dann doch mit der Drei-Byte-Ersatz-Methode erschlug. Im einzelnen bergen diese letzten Patches nicht viel Interessantes, so daß ich zu deren Erläuterung auf die Kommentare im EDPATCH-Listing verweisen möchte.

Ulrich Heidenreich
Werderstr. 35
4300 Essen 1

```
                TITLE   EdPlus-Patches (C) 1987 by
            ;=================================================
            ;
            ;       Patches in:
            ;       "EDTASM+ Version 1.01 vom 4. September 1981"
            ;       zur Verarbeitung von Kleinbuchstaben.
            ;       Zusätzlich eventuelle Anpassung an das
            ;       80-Zeichen-Format des Genie III mit:
            ;
0001        GENIE3  EQU     1
            ;
            ;       Durchführung des Patches:
            ;
            ;       1. Diese Quelle als EDPATCH/CMD assemblieren.
            ;       2. Zurück ins DOS
            ;       3. LOAD,EDPLUS/CMD
            ;          (oder wie Sie auch immer Ihren
            ;          EDTASM+ genannt haben!)
            ;       4. EDPATCH
            ;       5. Fertig!
            ;
            ;
            ;WICHTIGER HINWEIS:
            ;
            ;       Diese zusätzlichen Patches residieren am Ende
            ;       des EDTASM+ im ursprünglichen Textpuffer, der
            ;       seinerseits dann wieder hinter diesen Patches
            ;       beginnt.
            ;
            ;       Nun erlaubt der EDTASM+ aber, mit "QZ" seinen
            ;       Debugger Z-BUG aus dem Speicher zu entfernen,
            ;       um den Textpuffer entsprechend zu vergrößern.
            ;       Dies würde aber gleichzeitig den Speicher für
            ;       diese Patches für den Quelltext freigeben.
            ;
            ;       Um dieses zu verhindern, darf der Befehl "QZ"
            ;       nicht mehr ausführbar sein:

64B2                ORG     64B2H
64B2 0C63           DEFW    630CH

            ;       Stattdessen ist es aber hier möglich, durch
            ;       Setzen des Flags:
            ;
            ;ZBUG   EQU     1
            ;
            ;       eine EDTASM+-Version mit aktivem Z-BUG, bzw.
            ;       durch Löschen des Flags:
            ;
0000        ZBUG    EQU     0
            ;
            ;       eine Version ohne Z-BUG zu erzeugen!
            ;
            ;=================================================

8245                ORG     8245H   ; Hinter EDTASM+ ohne Z-BUG
                    COND    ZBUG
                    ORG     8FB9H   ; Hinter EDTASM+ mit Z-BUG
                    ENDC

            ;-------------------------------------------------
            ;GETTYP          Modifikation zur Annahme der Zeichen
            ;                d,t,o,g und h als Typkennzeichner
            ;-------------------------------------------------
```



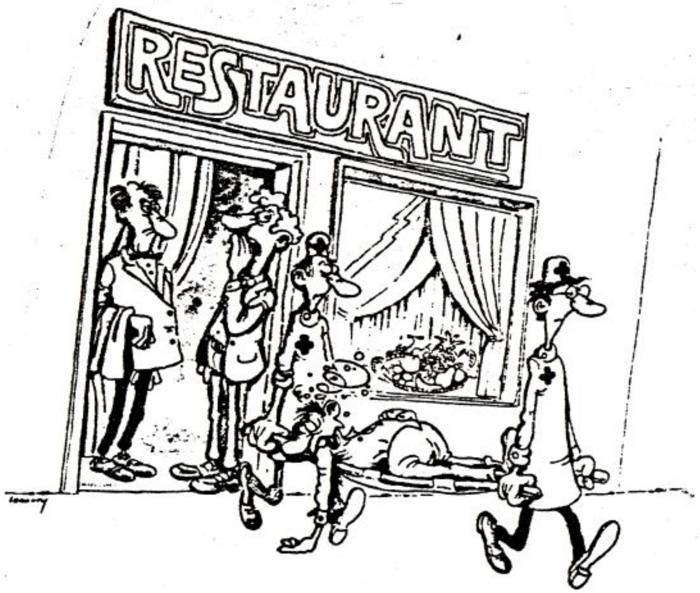
```
8245 CD5A65 GETTYP: CALL    655AH   ; Original-CALL "nachholen" !

            ;-------------------------------------------------------
            ;LCMASK          Maskiert Kleinschrift aus A,
            ;                falls Kleinschrift in A
            ;-------------------------------------------------------

8248 C5     LCMASK: PUSH    BC      ; BC retten
8249 F5             PUSH    AF      ; Flags retten
824A 47             LD      B,A     ; Akku-Inhalt retten
824B FE61           CP      'a'     ; unter Klein-"A" ?
824D 3807           JR      C,NO    ; Ja, kein Kleinbuchstabe!
824F FE7F           CP      'ß'+1   ; über "ß" ?
8251 3003           JR      NC,NO   ; Ja, kein Kleinbuchstabe!
8253 E6DF           AND     0DFH    ; Sonst: Großschrift maskieren.
8255 47             LD      B,A     ; Korrigierten Akku-Inhalt retten.
8256 F1     NO:     POP     AF      ; Flags restaurieren
8257 78             LD      A,B     ; Akku-Inhalt restaurieren.
8258 C1             POP     BC      ; BC auch wieder her!
8259 C9             RET             ; Fertig.

            ;-------------------------------------------------------
            ;LOCOMP          Erweiterung der "CP (HL)"-Befehle zur
            ;                Erkennung von Kleinbuchstaben.
            ;                (Wird via RST 8H ausgeführt!)
            ;-------------------------------------------------------

825A C5     LOCOMP: PUSH    BC      ; BC retten
825B 47             LD      B,A     ; Vergleichsoperator A in B retten
825C 4E             LD      C,(HL)  ; Vergleichsoperand => C
825D 79             LD      A,C     ; Vergleichsoperand auf
825E CD4882         CALL    LCMASK  ; Großschrift maskieren!
8261 77             LD      (HL),A  ; Neuen Operanden schreiben.
8262 78             LD      A,B     ; Vergleichsoperator restaurieren.
8263 CD4882         CALL    LCMASK  ; dto. wie Vergleichsoperand
8266 BE             CP      (HL)    ; Vergleich durchführen!
8267 71             LD      (HL),C  ; Vergleichsoperanden  restaurieren
8268 C1             POP     BC      ; BC restaurieren.
8269 C9             RET             ; fertig!

            ;-------------------------------------------------------
            ;GETHEX          Weitere Modifikation zur Annahme von
            ;                a ... f als gültige Sedezimal-Codes
            ;-------------------------------------------------------

826A CD8065 GETHEX: CALL    6580H
826D C34882         JP      LCMASK

            ;-------------------------------------------------------
            ;EDIT            Modifikation im Zeileneditor zur Annahme
            ;                von Kleinbuchstaben als Edit-Befehle
            ;-------------------------------------------------------

8270 216D6B EDIT:   LD      HL,6B6DH        ; Originalsequenz nachholen
8273 C34882         JP      LCMASK

            ;-------------------------------------------------------
            ;LANDS           Modifikation in Load und Save zur
            ;                Annahme auch der Kürzel d und c.
            ;-------------------------------------------------------

8276 CD5465 LANDS:  CALL    655AH           ; Originalsequenz nachholen
8279 C34882         JP      LCMASK
```

```
                 01300 ;----------------------------------------------------
                 01310 ;STARS          Modifikation in der Auswertung
                 01320 ;               des *LIST OFF/ON
                 01330 ;----------------------------------------------------
                 01340
827C 1A          01350 STARS:  LD      A,(DE)
827D CD4882      01360         CALL    LCMASK
8280 FE4C        01370         CP      'L'
8282 C9          01380         RET
                 01390
                 01400 ;Neuer Text-Puffer-Beginn:
                 01410
8283 FFFF        01420 EOP:    DEFW    -1
                 01430
8285 218B82      01440 INIT:   LD      HL,DMPCMD
8288 C31944      01450         JP      4419H
                 01460
                 01470 *LEBERKÄS OFF
                 01500 *LEBERKÄS ON
                 01510
                 01520 ;Zwischen den Leberkäsen steht:
                 01530
                 01540 ;DMPCMD:DEFM    'DUMP,EDNEU/CMD,5200H,8FF9H,5780H'
                 01550 ;       DEFB    0DH
                 01560
                 01570 ;----------------------------------------------------
                 01580 ;Es folgen nun direkte Patches im EDTASM+.
                 01590
                 01600 ;Obwohl es durchaus genügen würde, lediglich die
                 01610 ;Befehlsargumente anzupassen, habe ich hier der
                 01620 ;Verständlichkeit wegen die vollständigen
                 01630 ;betroffenen Befehls-Sequenzen eingesetzt!
                 01640 ;----------------------------------------------------
                 01650
                 01660 ;Umlegen des Text-Beginn-Zeigers hinter EdPatch:
                 01670
6231             01680         ORG     6231H
6231 218382      01690         LD      HL,EOP
                 01700
                 01710 ;RST 8-Zeiger für CP(HL)-Sequenzen verbiegen:
                 01720
6237             01730         ORG     6237H
6237 215A82      01740         LD      HL,LOCOMP
623A 220140      01750         LD      (4001H),HL
623D 214940      01760         LD      HL,4049H
6240 0000        01770         DEFW    0
6242 0000        01780         DEFW    0
                 01790
                 01800 ;Direkte Patches in den zugehörigen Teilroutinen:
                 01810
713D             01820         ORG     713DH
713D CD6A82      01830         CALL    GETHEX
                 01840
7113             01850         ORG     7113H
7113 CD4582      01860         CALL    GETTYP
                 01870
6B8F             01880         ORG     6B8FH
6B8F CD7082      01890         CALL    EDIT
                 01900
5A29             01910         ORG     5A29H
5A29 CD7682      01920         CALL    LANDS
                 01930
7567             01940         ORG     7567H
7567 CD7C82      01950         CALL    STARS
                 01960
```



```
                 01970 ;Ändern der Höchstgrenze für gültige Zeichen
                 01980 ;in Op-Codes von ursprünglich 5BH in 7FH:
                 01990
658C             02000         ORG     658CH
658C FE7F        02010         CP      7FH
                 02020
                 02030 ;Ersetzen der betreffenden CP (HL)-Befehle durch den RST 8:
                 02040
62DD             02050         ORG     62DDH
62DD CF          02060         RST     8
                 02070
6E69             02080         ORG     6E69H
6E69 CF          02090         RST     8
                 02100
75E6             02110         ORG     75E6H
75E6 CF          02120         RST     8
                 02130
                 02140 ;Änderungen für's Genie-III:
                 02150
                 02160         COND    GENIE3
                 02170
                 02180 ;Hochtasten mit Hochpfeil an Stelle des "Ä":
                 02190
57DA             02200         ORG     57DAH
57DA FE0B        02210         CP      11
                 02220
57DE             02230         ORG     57DEH
57DE FE7F        02240         CP      7FH
                 02250
6490             02260         ORG     6490H
6490 0B          02270         DEFB    11
                 02280
64FC             02290         ORG     64FCH
64FC FE0B        02300         CP      11
                 02310
                 02320 ;Erhöhen des Zeilenzählers für den P-Befehl
                 02330 ;von 16 auf 24 Zeilen:
                 02340
69AC             02350         ORG     69ACH
69AC 011800      02360         LD      BC,24
                 02370
                 02380 ;Bei Auf-/Abwärtstasten setzt der Original-EDTASM+ den
                 02390 ;Cursor durch "AND 0C0H" an den Zeilen-Anfang. Mit der
                 02400 ;80x24-Darstellung muß dies durch die Folge ...
                 02410
650D             02420         ORG     650DH
650D F5          02430         PUSH    AF
650E 3E1D        02440         LD      A,1DH
6510 CD3300      02450         CALL    0033H
6513 F1          02460         POP     AF
6514 00          02470         NOP
                 02480
                 02490 ; ... ersetzt werden!
                 02500
                 02510         ENDC
                 02520
                 02530 ;----------------------------------------------------
                 02540 ;Zum Schluß - der Vollständigkeit halber - noch einige
                 02550 ;Uralt-Patches, die vielleicht doch einigen Lesern
                 02560 ;noch nicht bekannt sind:
                 02570 ;----------------------------------------------------
                 02580
                 02590 ;1. Anpassung der Datumskonversion an das deutsche Format
                 02600 ;   "TT/MM/JJ" für die Kopfzeile des Assembler-Listings:
                 02610
```

```
5EA0                ORG   5EA0H
5EA0 214640         LD    HL,4046H      ; statt LD HL,4045H

5EB2                ORG   5EB2H
5EB2 2B             DEC   HL            ; statt INC HL

5EC5                ORG   5EC5H
5EC5 00             NOP                 ; statt DEC HL

                    ;2. Erweiterung der Fehlermeldung TRSDOS ERROR xx durch
                    ;   Ausgabe des zugehörigen Fehlertextes:

595B                ORG   595BH
595B CD1D62         CALL  621DH

621D                ORG   621DH
621D E680           AND   80H
621F CD0944         CALL  4409H
6222 212662         LD    HL,NEWTXT
6225 C9             RET

6226 45     NEWTXT: DEFM  'Err.'
6227 72
6228 72
6229 2E

8285                END   INIT
00000 TOTAL ERRORS

DMPCMD  828B
EDIT    8270
EDP     8263
GENIE3  0001
GETHEX  826A
GETTYP  8245
INIT    8285
LANDS   8276
LCMASK  8248
LOCOMP  825A
NEWTXT  6226
NO      8256
STARS   827C
ZBUG    0000
```



Vielleicht sollten wir doch lieber den Computer rausschmeißen und den Chef zurückholen.

R-Befehl MSDOS-Like

Um was geht's eigentlich?

MSDOS-Verwöhnte Können dem R-Befehl unseres NEWDOS-/ GDOS-Betriebssystems nur ein müdes Lächeln abringen, haben sie doch mit den Funktionstasten F1 bis F4 nicht nur die Möglichkeit, DOS-Befehle zu wiederholen, sondern vorher noch beliebig zu ändern. Andererseits besitzt die BASIC-Version unserer Betriebssysteme mit dem EDIT-Befehl ein Werkzeug, das die Möglichkeiten des MSDOS weit in den Schatten stellen würde, wäre es auch unter DOS anwendbar!

Ein Editor fürs DOS!

Einer einfachen Implementation in die "Befehlseingabe" - also auf DOS-Ebene - der Interpreter-ROM-Routinen steht nun aber im Wege, daß diese Routinen doch zu sehr ihrer Verwendung als BASIC-Editor nahestehen: So werden zum Beispiel die BASIC-I/O-Routinen 0384H und 032AH an Stelle von 0049H bzw. 0033H hierin verwendet, die bekanntermaßen ohne Initialisierung der BASIC-Vektoren im Niemandsland verschwinden. Darüber hinaus nutzt zum Beispiel der I(nsert)-Unterbefehl die Interpreter-Routine an 1958H, die zwar brav Zeichen einfügt, jedoch aber anhand der BASIC-Programmzeiger durchaus auch einen ?OM Error zum Besten geben Kann.

Aus diesen und weiteren Gründen ist schon etwas mehr Aufwand nötig, um MSDOS ein Schnippchen zu schlagen: Hier bietet sich an, den an sich schon vorhandenen R-Befehl des NEWDOS entsprechend aufzubohren und einem diesem Verwendungszweck angepaßten Zeileneditor eine /SYS-Datei zu spendieren. Wie das Einrichten einer /SYS-Datei und der zugehörige Patch in SYS1/SYS genau aussieht, mag der an Einzelheiten interessierte Leser im

"DOS-Buch",
Verlag Röckrath-Microcomputer,
Anhang "ID - Eine Idee zum NEWDOS/80"

nachlesen! Sollte das dort beschriebene Hilfsprogramm in SYS22/SYS bzw. SYS15/SYS bereits installiert sein, so beachten Sie bitte die diesbezüglichen Hinweise im Assemblerlisting. (Wer aber SYS22/SYS für seine eigenen DOS-Erweiterungen bereits "verschwendet" hat, sei auf SYS24/SYS aufmerksam gemacht: Das V24-Testprogramm wird wohl Kaum einmal als System-Overlay benötigt werden!)

SYS1/SYS patchen

SYS1/SYS besitzt zwar in seiner Sprungleiste einen Eintrag für den Befehl R, der im Normalfall jedoch nie benötigt wird. Zwei Besonderheiten machen diesen Eintrag aber notwendig: Zum Einen braucht der Befehl LIB (GDOS auch:?) diesen Eintrag, um auch den R-Befehl anzuzeigen, zum Anderen wird bei gewählter System-Option BE=N - also: "R-Befehl verboten" - über diesen Eintrag die Fehlermeldung "unzulässige DOS-Funktion" des SYS1/SYS angesprungen. Der tatsächliche R-Befehl besteht dagegen lediglich darin, die beiden ersten Zeichen des DOS-Befehlspuffers - hier steht ja jetzt R,<cr> - zu restaurieren und mit dem so wieder hergestellten vorherigen Befehl normal fortzufahren.

Diese Tatsache, daß zur Beachtung der System-Option BE=N bereits ein an sich unnötiger Eintrag in der SYS1-Sprungleiste zur Verfügung steht, nutzen wir aus, um mit dem R-Befehl den Editor in SYS22/SYS zu aktivieren:

Alte BE-Option-Behandlung:

```
4E5F 3A6D43     SEARCH:  LD    HL,(FLAG4)   ; System-Option BE:
4E62 CB6F                BIT   5,A          ; Bit 5 gelöscht =>
4E64 280F                JR    Z,NOCMDR     ; Befehl "R" nicht erlaubt
```

Neue BE-Option-Behandlung:

```
4E5F 3A6D43     SEARCH:  LD    A,(FLAG4)    ; System-Option BE wird erst
4E62 CB6F                BIT   5,A          ; in SYS22/SYS geprüft =>
4E64 180F                JR    NOCMDR       ; Befehl "R" hier nicht beachten!
```

Alter Eintrag in der Sprungleiste:

```
5066 52                  DEFM  'R'          ; Befehlskürzel
5067 80                  DEFB  80H          ; Entry-Counter 0 in
5068 23                  DEFB  23H          ; SYS1/SYS => Fehlermeldung
```

Neuer Eintrag in der Sprungleiste:

```
5066 52                  DEFM  'R'          ; Befehlskürzel
5067 80                  DEFB  81H          ; Entry-Counter 1 in
5068 23                  DEFB  0F8H         ; SYS22/SYS => R-Editor
                                            ; (bzw. DEFB 0FAH für SYS24/SYS)
```


Nötige Patches in SYS1/SYS:

```
DRV 00  E311 D649 0102 004E C5D5 2A18 4322 A743 ...I...N..*.C".C
0   10  064F 211B 43C3 4000 D1ED 7B9D 43CB 6928 .O!.C.§...ä.C.i(
0H  20  04CB 6020 BFC1 78B1 2003 23CB B6CB A6ED ..`...x...#.....
    30  439D 43D5 F1C3 3351 31E0 41F5 F101 2D40 C.C...3Q1.A...-§
DRS 40  1108 44C5 D5EB 216A 43CB AE21 1843 0650 ..D...!jC..!.C.P
143150  7EFE 0DC8 E51A 13CD B545 FE0D 7723 2807 ß........E..w#(.
597H60  10F3 F13E 36B7 C93A 6D43 CB6F 180F 2A18 ...>6..:mC.o..*. statt 28
    70  4311 520D DF20 062A A743 2218 43E1 1158 C.R....*.C".C..X
    80  4FE5 1ABE 1323 28FA 2B1B 0730 05CD D54C O....#(.+..0...L
    90  301D E11A 0713 30FB 1313 1AB7 20E3 0143 0.....0........C
    A0  E416 417E FE2A 2010 3EEB 0E07 EFC1 C9C1 ..Aß.*..>.......
    B0  1A4F 131A 4713 1A57 CB71 280F 3A6A 4307 .O..G..W.q(.:jC.
FRS C0  3009 E680 3E38 C28B 4DB7 C979 E61F 4FC5 0...>8..M..y..O.
1   D0  4A79 E6C0 C465 5120 20CB 6928 20CD D94C Jy...eQ...i(...L
1H  E0  38CB C501 C551 CDC5 4CC1 2005 CDD9 4C38 8....Q..L.....L8
    F0  BCD5 11E0 51CD 6851 D13E 3020 B0CB 61C4 ....Q.hQ.>0...a.
    F0  BCD5 11E0 51CD 6851 D13E 3020 B0CB 61C4 ....Q.hQ.>0...a.
```

```
DRV 00  0049 B02A 0049 4E46 4F81 FF00 0102 0050 .I.*.INFO......P
0   10  4A4B 4C80 A510 4B49 4C4C 8045 904C 4388 JKL...KILL.E.LC.
0H  20  E500 4C46 81FE 004C 4942 82E3 004C 4953 ...LF...LIB...LIS
    30  5485 F08B 4C4F 4144 80A4 504C 5754 81F9 T...LOAD..PLWT..
DRS 40  004E 81E4 B04E 4446 C028 0050 4155 5345 .N...NDF.(.PAUSE
143350  88EB 0050 4483 E900 504F 5254 82FF 0050 ...PD...PORT...P
599H60  5249 4E54 86F0 8850 524F 5486 E900 5055 RINT...PROT...PU  statt:
    70  5247 4589 E900 5281 F800 5381 E900 5354 RGE...R...S...ST  80 23
    80  4D54 89EB 0055 4852 82E5 0056 2B84 E500 MT...UHR...V+...  (SYS24/SYS:
    90  5632 3480 FA00 5A86 FF00 5A45 4954 8AE9 V24...Z...ZEIT..  81 FA)
    A0  0026 83E5 0021 83EB 8A3B 86E3 002F 85E3 .&...!...;.../..
    B0  003F 82E3 003E C048 004D 3E82 EBB0 2323 .?...>.H.M>...##
FRS C0  82F9 0046 2380 FB00 3830 83F9 0036 3484 ...F#...80...64.
3   D0  F900 4444 4581 F100 4944 80F8 0000 0021 ..DDE...ID.....!
3H  E0  584F 0E40 0608 7ECB 7F23 2005 CDB7 5110 XO.§..ß.#....Q.
    F0  F523 237E B7CA B551 0DCC B551 28E4 CDAD .##ß...Q...Q(...
```

Der "neue" Editor

In SYS22/SYS muß dann natürlich genau diese Restauration der beiden ersten Zeichen nachgeholt werden, bevor die so vorbereitete Zeile dem EDIT übergeben wird. Da - wie oben beschrieben - der BASIC-Zeileneditor für diesen speziellen Zweck sowieso neu geschrieben werden muß, spricht nichts dagegen, bei dieser Gelegenheit gleich einige Verbesserungen einzubauen:

- Die Edit-Unterbefehle K(ill) und D(elete) klammern nicht wie im Original die gelöschten Zeichen mit Ausrufungszeichen; diese werden vielmehr sofort gelöscht und der Rest der Zeile entsprechend eingerückt.

- Ebenso geschieht I(nsert) nicht mehr "blind", sondern es wird der Rest der Zeile auf dem Bildschirm sichtbar ausgerückt.

- Bei Aufruf des R-Befehls gibt Edit selbstätig einmal den L(ist)-Befehl.

- Der nicht-zerstörende-Backspace zur Positionierung des Cursors rückwärts löscht nun auch im Bildschirm keine Zeichen: Die zu ändernde Zeile bleibt somit immer vollständig mit sämtlichen aktuellen Änderungen sichtbar!

- Der Befehl C(hange) verbietet nun die Annahme von Steuerzeichen; im "alten" Editor fehlte diese Notbremse mit dem Mißerfolg, daß unsichtbare Steuerzeichen häufig unerklärliche Fehlermeldungen erzeugten!

Neben diesen "Schönheitsreparaturen" am Zeileneditor fand ich es sinnvoll, einen weiteren Befehl R(estore) den Edit-Unterbefehlen hinzuzufügen, da der Original-DOS-Befehl "R" die unschöne Eigenart besitzt, nach einem zufälligen ungewollten Zeilenvorschub auf DOS-Ebene den - dann vorletzten - Befehl zu "vergessen". Dies mag folgendes Beispiel verdeutlichen:

So soll's klappen:

```
Gegebener Befehl: Das steht im DOS-Eingabepuffer: Das sind die 2 ersten Zeichen:
--------------------------------------------------------------------------------
DIR,0,I,S,A<cr>   DIR,0,I,S,A<cr>                 DI
R<cr>             vorher:  R<cr>R,0,I,S,A         DI
                  nachher: DIR,0,I,S,A
```

So kann's aber passieren:

```
Gegebener Befehl: Das steht im DOS-Eingabepuffer: Das sind die 2 ersten Zeichen:
--------------------------------------------------------------------------------
DIR,0,I,S,A<cr>   DIR,0,I,S,A<cr>                 DI
versehentlich:
<cr>              <cr>IR,0,I,S,A                  <cr>I
dann:
R<cr>             vorher:  R<cr>R,0,I,S,A         <cr>I
                  nachher: <cr>IR,0,I,S,A
```

Auf diese Weise hat DOS dann trotz Restauration der ersten beiden Zeichen den Befehl "vergessen", da das versehentlich gegebene <cr> (NEW LINE, ENTER) zu Beginn im DOS-Eingabepuffer verbleibt. Der oben erwähnte zusätzliche Edit-Unterbefehl R(estore) ermöglicht nun, eben dieses <cr> zu löschen, um anschliessend das verlorengegangene "D" des Ursprungsbefehls wieder "davorzuhängen".

Alle weiteren Einzelheiten zum komfortablen R-Befehl entnehme der interessierte Leser bitte den Kommentarzeilen des Assemblerlistings!

Ulrich Heidenreich
Werderstr. 35
4300 Essen 1

```
        title   R-Editor in SYS22/SYS

;*****************************************************************
;*                                                               *
;*      SYS22/SYS        Erweiterung des NEWDOS/GDOS-Befehls     *
;*                       "R" um den bekannten BASIC-Zeileneditor *
;*                                                               *
;*****************************************************************

;----------------------------------------------------------------
;
; Soll diese Befehlserweiterung zusätzlich zum
;
; "ID - Eine Idee zum NEWDOS/80"
;
; in SYS22/SYS untergebracht werden, so sind im Hilfsprogramm ID
; folgende Änderungen vorzunehmen:
;
; Alle Marken "EXIT" sind wegen des gleichlautenden Labels im
; Editor durch "IDEXIT" zu ersetzen!
;
; Die Zuweisung
;
; INCHAR        EQU     0049H
;
; ist aus dem ID-Kopf zu entfernen, da dieses Label auch
; im Editor Verwendung findet!
;
; Ab der Marke "ENDE" sind folgende Zeilen hinzuzufügen:
;
;ende:  xor     a               ; System-Overlay als nicht aktiv
;       ld      (4317h),a       ; markieren, da Sektorpuffer den
;       pop     af              ; Editor überschreibt
;       ret
;
; An Stelle von
;
;buffer:defs    256+16
;
; Tritt nun
;
;buffer equ     $
;
; Die Marke "START" ist durch "S22ID" zu ersetzen;
; die Direktive END START entfällt selbstverständlich; ab hier
; ist stattdessen der Editor anzufügen!
;
;----------------------------------------------------------------

        page
```

```
4D00                    org     4d00h

4D00 0C         start:  inc     c               ; Entry-Counter prüfen:
4D01 0D                 dec     c

                ; Diese Zeile nur einfügen, falls ID ebenfalls in SYS22/SYS:
                ;
                ;       jr      z,s22id         ; 0 = ID !
                ;
4D02 0D                 dec     c               ; 1 = R  !
4D03 CA204D             jp      z,s22r
4D06 3E2A       ildos:  ld      a,2ah           ; Weder noch:
4D08 B7                 or      a               ; "Illegal DOS Function"
4D09 C9                 ret                     ; melden!

4467            outext  equ     4467h           ; Text (HL) bis 0dh,3 ausgeben
4318            cmdbuf  equ     4318h           ; DOS-Befehlspuffer
43A7            twocha  equ     43a7h           ; 1. beiden Zeichen statt R,(cr)

                ;------------------------------------------------------------
                ;       Edit-Unterbefehle D(elete) und R(estore)
                ;------------------------------------------------------------

4D0A 7E         delete: ld      a,(hl)          ; Zeilen-Ende
4D0B FE0D               cp      0dh             ; erreicht?
4D0D C8                 ret     z               ; ja => das darf nicht gelöscht werden!

4D0E 54         restor: ld      d,h             ; Bei Restore jedoch!
4D0F 5D                 ld      e,l             ; DE:Lösch-Stelle
4D10 13                 inc     de              ; plus einem Zeichen!
4D11 1A         delop:  ld      a,(de)          ; Zeichen von eins hinter Lösch-Stelle
4D12 1B                 dec     de              ; an Lösch-Stelle
4D13 12                 ld      (de),a          ; umschaufeln!
4D14 13                 inc     de
4D15 13                 inc     de
4D16 FE0D               cp      0dh             ; Zeilen-Ende umgeschaufelt ?
4D18 20F7               jr      nz,delop        ; Nein, weiter schaufeln!
4D1A CD924E             call    dispt           ; Ja, Rest der Zeile anzeigen.
4D1D 10EB               djnz    delete          ; Replikator beachten!
4D1F C9                 ret                     ; Fertig!

                ;------------------------------------------------------------
                ;       Start-Ansprung des Editors. D(elete)
                ;       und R(estore) beginnen davor, um
                ;       für Relativ-Sprünge günstige Sprung-
                ;       weiten zu erzielen!
                ;------------------------------------------------------------

4D20 3A6D43     s22r:   ld      a,(436dh)       ; System-Option BE ?
4D23 CB6F               bit     5,a             ; Ja, dann R-Befehl
4D25 CA064D             jp      z,ildos         ; nicht erlaubt!

4D28 3E3E               ld      a,3eh           ; Xtend/Hack-Flag auf nicht
4D2A 323E4E             ld      (hxflag),a      ; zerstörenden backspace setzen.
4D2D 2AA743             ld      hl,(twocha)     ; 1. beide Zeichen des DOS-Befehls
4D30 221843             ld      (cmdbuf),hl     ; restaurieren, da durch R,(cr)

                ;------------------------------
                ;       Neu-Start für Edit-
                ;       Unterbefehl A(gain)
                ;------------------------------

4D33 211843     again:  ld      hl,cmdbuf       ; überschrieben. DOS-Puffer in
4D36 11C14E             ld      de,edibuf       ; Edit-Puffer
```



```
4D39 D5                 push    de              ; übertragen.
4D3A 015000             ld      bc,80
4D3D EDB0               ldir
4D3F E1                 pop     hl              ; Edit-Puffer-Inhalt

                        page
```

```
                        title   R-Editor in SYS22/SYS

;***********************************************************************
;*                                                                     *
;*      SYS22/SYS       Erweiterung des NEWDOS/GDOS-Befehls            *
;*                      "R" um den bekannten BASIC-Zeileneditor        *
;*                                                                     *
;***********************************************************************

;-----------------------------------------------------------------------
;
; Soll diese Befehlserweiterung zusätzlich zum
;
; "ID - Eine Idee zum NEWDOS/80"
;
; in SYS22/SYS untergebracht werden, so sind im Hilfsprogramm ID
; folgende Änderungen vorzunehmen:
;
; Alle Marken "EXIT" sind wegen des gleichlautenden Labels im
; Editor durch "IDEXIT" zu ersetzen!
;
; Die Zuweisung
;
; INCHAR        EQU     0049H
;
; ist aus dem ID-Kopf zu entfernen, da dieses Label auch
; im Editor Verwendung findet!
;
; Ab der Marke "ENDE" sind folgende Zeilen hinzuzufügen:
;
;ende:  xor     a               ; System-Overlay als nicht aktiv
;       ld      (4317h),a       ; markieren, da Sektorpuffer den
;       pop     af              ; Editor überschreibt
;       ret
;
; An Stelle von
;
;buffer:defs    256+16
;
; Tritt nun
;
;buffer equ     $
;
; Die Marke "START" ist durch "S22ID" zu ersetzen;
; die Direktive END START entfällt selbstverständlich; ab hier
; ist stattdessen der Editor anzufügen!
;
;-----------------------------------------------------------------------

                page
```

```
4D00                    org     4d00h

4D00 0C         start:  inc     c               ; Entry-Counter prüfen:
4D01 0D                 dec     c

                ; Diese Zeile nur einfügen, falls ID ebenfalls in SYS22/SYS:
                ;
                ;       jr      z,s22id         ; 0 = ID !
                ;
4D02 0D                 dec     c               ; 1 = R  !
4D03 CA204D             jp      z,s22r
4D06 3E2A       ildos:  ld      a,2ah           ; Weder noch:
4D08 B7                 or      a               ; "Illegal DOS Function"
4D09 C9                 ret                     ; melden!

4467            outext  equ     4467h           ; Text (HL) bis 0dh,3 ausgeben
4318            cmdbuf  equ     4318h           ; DOS-Befehlspuffer
43A7            twocha  equ     43a7h           ; 1. beiden Zeichen statt R,<cr>

                ;--------------------------------------------------------
                ;       Edit-Unterbefehle D(elete) und R(estore)
                ;--------------------------------------------------------

4D0A 7E         delete: ld      a,(hl)          ; Zeilen-Ende
4D0B FE0D               cp      0dh             ; erreicht?
4D0D C8                 ret     z               ; ja => das darf nicht gelöscht werden!

4D0E 54         restor: ld      d,h             ; Bei Restore jedoch!
4D0F 5D                 ld      e,l             ; DE:Lösch-Stelle
4D10 13                 inc     de              ; plus einem Zeichen!
4D11 1A         delop:  ld      a,(de)          ; Zeichen von eins hinter Lösch-Stelle
4D12 1B                 dec     de              ; an Lösch-Stelle
4D13 12                 ld      (de),a          ; umschaufeln!
4D14 13                 inc     de
4D15 13                 inc     de
4D16 FE0D               cp      0dh             ; Zeilen-Ende umgeschaufelt ?
4D18 20F7               jr      nz,delop        ; Nein, weiter schaufeln!
4D1A CD924E             call    dispt           ; Ja, Rest der Zeile anzeigen.
4D1D 10EB               djnz    delete          ; Replikator beachten!
4D1F C9                 ret                     ; Fertig!

                ;--------------------------------------------------------
                ;       Start-Ansprung des Editors. D(elete)
                ;       und R(estore) beginnen davor, um
                ;       für Relativ-Sprünge günstige Sprung-
                ;       weiten zu erzielen!
                ;--------------------------------------------------------

4D20 3A6D43     s22r:   ld      a,(436dh)       ; System-Option BE ?
4D23 CB6F               bit     5,a             ; Ja, dann R-Befehl
4D25 CA064D             jp      z,ildos         ; nicht erlaubt!

4D28 3E3E               ld      a,3eh           ; Xtend/Hack-Flag auf nicht
4D2A 323E4E             ld      (hxflag),a      ; zerstörenden backspace setzen.
4D2D 2AA743             ld      hl,(twocha)     ; 1. beide Zeichen des DOS-Befehls
4D30 221843             ld      (cmdbuf),hl     ; restaurieren, da durch R,<cr>

                        page
```



```
                ;--------------------------------------------------------
                ;       Neu-Start für Edit-
                ;       Unterbefehl A(gain)
                ;--------------------------------------------------------

4D33 211843     again:  ld      hl,cmdbuf       ; überschrieben. DOS-Puffer in
4D36 11C14E             ld      de,edibuf       ; Edit-Puffer
4D39 D5                 push    de              ; übertragen.
4D3A 015000             ld      bc,80
4D3D EDB0               ldir
4D3F E1                 pop     hl              ; Edit-Puffer-Inhalt

                ;--------------------------------------------------------
                ;       Neu-Start für Edit-
                ;       Unterbefehl L(ist)
                ;--------------------------------------------------------

4D40 CD6744     redo:   call    outext          ; ausgeben
4D43 3E0E               ld      a,0eh           ; Cursor einschalten
4D45 CDB34E             call    outcha
4D48 3E1B               ld      a,27            ; Cursor auf Zeilenanfang
4D4A CDB34E             call    outcha          ; setzen.

                ;--------------------------------------------------------
                ;       Edit-Hauptschleife
                ;--------------------------------------------------------

4D4D 114D4D     edit:   ld      de,edit         ; Rückkehradresse
4D50 D5                 push    de              ; => Stack: d.h. RET immer hierhin!
4D51 0600               ld      b,0             ; Replikator löschen
4D53 CDBB4E     inloop: call    inchar          ; Befehl/Replikator lesen
4D56 D630               sub     '0'             ; ASCII-Korrektur
4D58 380E               jr      c,edcmd         ; Kleiner 0 => Befehl
4D5A FE0A               cp      10              ; Größer 9
4D5C 300A               jr      nc,edcmd        ; ebenfalls Befehl
4D5E 4F                 ld      c,a             ; Replikator-Ziffer => C
4D5F 78                 ld      a,b             ; A:alter Replikator
4D60 07                 rlca                    ; mal 2
4D61 07                 rlca                    ; mal 4
4D62 80                 add     a,b             ; plus 1 => mal 5
4D63 07                 rlca                    ; mal 10
4D64 81                 add     a,c             ; plus Replikator-Ziffer
4D65 47                 ld      b,a             ; => Replikator in Register B
4D66 18EB               jr      inloop          ; weiter einlesen.

                ;--------------------------------------------------------
                ;       Ansprung der Edit-Unterbefehle
                ;--------------------------------------------------------

4D68 05         edcmd:  dec     b               ; Replikator = 0
4D69 04                 inc     b               ; ?
4D6A 2001               jr      nz,rgz          ; Nein, OK
4D6C 04                 inc     b               ; sonst: Replikator = 1
4D6D C630       rgz:    add     a,'0'           ; ASCII-Korrektur rückgängig machen
4D6F FE0D               cp      0dh             ; NEW LINE ?
4D71 284D               jr      z,enter
4D73 FE08               cp      08h             ; BACKSPACE ?
4D75 2868               jr      z,bsp
4D77 FE20               cp      20h             ; SPACE ?
4D79 2859               jr      z,space
4D7B CBAF               res     5,a             ; Kleinschrift herausmaskieren.
4D7D FE44               cp      'D'             ; D(elete) ?
4D7F 2889               jr      z,delete
4D81 FE52               cp      'R'             ; R(estore) ?
4D83 2889               jr      z,restor
```

```
4D85 FE51          cp      'Q'          ; Q(uit) ?
4D87 2864          jr      z,quit
4D89 FE4C          cp      'L'          ; L(ist) ?
4D8B 2825          jr      z,list
4D8D FE53          cp      'S'          ; S(earch) ?
4D8F 2860          jr      z,search
4D91 FE49          cp      'I'          ; I(nsert) ?
4D93 CA1D4E        jp      z,insert
4D96 FE43          cp      'C'          ; C(hange) ?
4D98 CA804E        jp      z,change
4D9B FE45          cp      'E'          ; E(xit) ?
4D9D 2824          jr      z,exit
4D9F FE58          cp      'X'          ; X(tend) ?
4DA1 2861          jr      z,xtend
4DA3 FE4B          cp      'K'          ; K(ill) ?
4DA5 CA6B4E        jp      z,kill
4DA8 FE48          cp      'H'          ; H(ack) ?
4DAA 2863          jr      z,hack
4DAC FE41          cp      'A'          ; A(lter) ?
4DAE C0            ret     nz           ; nein, zurück nach "edit"
4DAF D1            pop     de           ; Rückkehradresse vom Stack
4DB0 1881          jr      again        ; "Same procedure as last year!"

                   ;-------------------------------------------
                   ;       Edit-Unterbefehl L(ist)
                   ;-------------------------------------------

4DB2 D1            list:   pop     de           ; siehe Kommentar zu quit!
4DB3 CD6744                call    outext       ; (rest der) Zeile ausgeben
4DB6 3E0D                  ld      a,0dh        ; Zeilenvorschub
4DB8 CDB34E                call    outcha       ; ausgeben
4DBB 21C14E                ld      hl,edibuf    ; Mit Zeilenbeginn Editor
4DBE 1880                  jr      redo         ; erneut starten.

                   ;-------------------------------------------
                   ;       Edit-Unterbefehle ENTER / NEW LINE
                   ;       und E(xit)
                   ;-------------------------------------------

4DC0 CD6744        enter:  call    outext       ; (Rest der) Zeile ausgeben
4DC3 21C14E        exit:   ld      hl,edibuf    ; Inhalt des Edit-Puffers
4DC6 111843                ld      de,cmdbuf    ; in den DOS-Befehlspuffer
4DC9 D5                    push    de           ; (Adresse retten)
4DCA 015000                ld      bc,80        ; kopieren.
4DCD EDB0                  ldir
4DCF E1                    pop     hl           ; Befehl aus diesem
4DD0 D1                    pop     de           ; - nach Stackkorrektur! -
4DD1 C31944                jp      4419h        ; ans DOS übermitteln!

                   ;-------------------------------------------
                   ;       Edit-Unterbefehl "Leertaste"
                   ;-------------------------------------------

4DD4 7E            space:  ld      a,(hl)       ; Ende der Zeile
4DD5 FE0D                  cp      0dh          ; erreicht?
4DD7 C8                    ret     z            ; ja, "rechter" geht's nicht!
4DD8 23                    inc     hl           ; sonst: Textzeiger weiter
4DD9 CDB34E                call    outcha       ; Zeichen ausgeben
4DDC 10F6                  djnz    space        ; evtl. Replikator beachten.
4DDE C9                    ret                  ; Fertig!

                           page

                   ;-------------------------------------------
                   ;       Edit-Unterbefehl "Rückpfeil"
                   ;-------------------------------------------

4DDF 11C14E        bsp:    ld      de,edibuf    ; Anfang der Zeile
4DE2 DF            bsplop: rst     18h          ; erreicht?
4DE3 C8                    ret     z            ; ja "backer" gehts nicht!
4DE4 2B                    dec     hl           ; sonst: Textzeiger zurück
4DE5 3E18                  ld      a,24         ; dto. Cursor
4DE7 CDB34E                call    outcha
4DEA 10F6                  djnz    bsplop       ; Replikator ?
4DEC C9                    ret

                   ;-------------------------------------------
                   ;       Edit-Unterbefehl Q(uit)
                   ;-------------------------------------------

4DED D1            quit:   pop     de           ; Rückkehradresse nach edit vom Stack
4DEE C30044                jp      4400h        ; alles ignorieren!

                   ;-------------------------------------------
                   ;       Edit Unterbefehl S(earch)
                   ;-------------------------------------------

4DF1 CDBB4E        search: call    inchar       ; Was sollen wir suchen?
4DF4 5F                    ld      e,a          ; Such-Zeichen => E
4DF5 7E            sloop:  ld      a,(hl)       ; Zeilen-Ende bei Suche
4DF6 FE0D                  cp      0dh          ; erreicht?
4DF8 C8                    ret     z            ; Ja, nicht gefunden, zurück nach edit!
4DF9 CDB34E                call    outcha       ; sonst: Zeichen ausgeben
4DFC 23                    inc     hl           ; Textzeiger weiterstellen
4DFD 7E                    ld      a,(hl)       ; Suchzeichen
4DFE BB                    cp      e            ; erreicht?
4DFF 20F4                  jr      nz,sloop     ; Nein, weiter suchen!
4E01 10F2                  djnz    sloop        ; Wenn gefunden, Replikator beachten!
4E03 C9                    ret                  ; Dann aber fertig!

                   ;-------------------------------------------
                   ;       Edit-Unterbefehle X(tend) und H(ack)
                   ;-------------------------------------------

4E04 7E            xtend:  ld      a,(hl)       ; Zeilen-Ende
4E05 FE0D                  cp      0dh          ; erreicht?
4E07 280D                  jr      z,bspdel     ; Ja, dann hier wie I(nsert) weiter!
4E09 CDB34E                call    outcha       ; Sonst: Zeichen ausgeben
4E0C 23                    inc     hl           ; Textzeiger weiter
4E0D 18F5                  jr      xtend        ; Zeilen-Ende suchen!

4E0F 360D          hack:   ld      (hl),0dh     ; Genau hier Zeilenende einsetzen
4E11 3E1E                  ld      a,30         ; Im Bildschirm ab hier löschen
4E13 CDB34E                call    outcha       ; weiter wie I(nsert), jedoch

4E16 3E36          bspdel: ld      a,36h        ; bei Hack/Xtend zerstörenden
4E18 323E4E                ld      (hxflag),a   ; Backspace aktivieren.
4E1B 1805                  jr      bcont

                           page
```

```
                02910 ;-------------------------------------
                02920 ;        Edit-Unterbefehl I(nsert)
                02930 ;-------------------------------------
                02940
4E1D 3E3E       02950 insert: ld    a,3eh          ; Bei Insert: Statt Backspace nur
4E1F 323E4E     02960         ld    (hxflag),a     ; Cursor-Rückstellung erlauben.
                02970
4E22 CDBB4E     02980 bcont:  call  inchar         ; Was wollen wir I(nserten) ?
4E25 FE0D       02990         cp    0dh            ; NEW LINE bitteschön nicht einfügen,
4E27 2897       03000         jr    z,enter        ; sondern wie Edit-Befehl behandeln!
4E29 FE1B       03010         cp    1bh            ; ESC = Ende Einfügen
4E2B 2006       03020         jr    nz,noesc       ; Nein!
4E2D 3E3E       03030         ld    a,3eh          ; Ja, auf nicht zerstörenden Backspace
4E2F 323E4E     03040         ld    (hxflag),a     ; zurückschalten
4E32 C9         03050         ret                  ; und zurück zu edit!
                03060
4E33 FE08       03070 noesc:  cp    8              ; Backspace?
4E35 2019       03080         jr    nz,nobsp       ; Nein!
4E37 11C14E     03090         ld    de,edibuf      ; Geht's "backer" ?
4E3A DF         03100         rst   18h            ; Na?
4E3B 28E5       03110         jr    z,bcont        ; Nein, war wohl nichts!
                03120
4E3D 2B         03130         dec   hl             ; Textzeiger zurückstellen
4E3E 3E0D       03140 hxflag: ld    a,0dh          ; Bei Hack & Xtend hier: LD (HL),0dh =
4E40 3A3E4E     03150         ld    a,(hxflag)     ; Zeilenende einsetzen
4E43 FE36       03160         cp    36h            ; und Zeichen
4E45 3E08       03170         ld    a,8            ; löschen!
4E47 2802       03180         jr    z,delbsp       ; Sonst nur:
4E49 3E18       03190         ld    a,24           ; Cursor zurückstellen
4E4B CDB34E     03200 delbsp: call  outcha
4E4E 18D2       03210         jr    bcont          ; weiter "inserten"
                03220
4E50 FE20       03230 nobsp:  cp    20h            ; Steuerzeichen?
4E52 38CE       03240         jr    c,bcont        ; Bloß nicht einfügen!
4E54 47         03250         ld    b,a            ; Zeichen retten
4E55 11104F     03260         ld    de,edibuf+79   ; Edit-Puffer-Ende
                03270
4E58 1B         03280 inslop: dec   de             ; Ein Zeichen davor
4E59 1A         03290         ld    a,(de)         ; um
4E5A 13         03300         inc   de             ; eine Stelle nach
4E5B 12         03310         ld    (de),a         ; hinten schieben
4E5C 1B         03320         dec   de
4E5D DF         03330         rst   18h            ; Einfügstelle erreicht?
4E5E 20F8       03340         jr    nz,inslop      ; Nein, immer weiter so!
4E60 70         03350         ld    (hl),b         ; Ja, Zeichen einfügen.
4E61 78         03360         ld    a,b            ; Auf'm Bildschirm wollen
4E62 CDB34E     03370         call  outcha         ; wir's aber auch sehen!
4E65 23         03380         inc   hl             ; Textzeiger weiterstellen.
4E66 CD924E     03390         call  dispt          ; Rest der Zeile anzeigen.
4E69 18B7       03400         jr    bcont          ; weiter "inserten"
                03410
                03420 ;-------------------------------------
                03430 ;        Edit-Unterbefehl K(ill)
                03440 ;-------------------------------------
                03450
4E6B CDBB4E     03460 kill:   call  inchar         ; Was soll's denn bitteschön sein?
4E6E 4F         03470         ld    c,a            ; Das retten wir mal in C
4E6F C5         03480 kiloop: push  bc             ; Replikator, Zeichen retten
4E70 0601       03490         ld    b,1            ; Replikator auf 1 setzen
4E72 CD0A4D     03500         call  delete         ; 1 Zeichen löschen
4E75 C1         03510         pop   bc             ; Kill-Replikator, -Zeichen her!
4E76 7E         03520         ld    a,(hl)         ; welches Zeichen hat "delete"
4E77 FE0D       03530         cp    0dh            ; herangeschaufelt? Etwa Ende?
4E79 C8         03540         ret   z              ; ja, weitermachen = Unsinn!
4E7A B9         03550         cp    c              ; Ist Zeichen erreicht?
```



```
4E7B 20F2       03560         jr    nz,kiloop      ; Nein, weiter K(ill)en!
4E7D 10F0       03570         djnz  kiloop         ; Replikator ?
4E7F C9         03580         ret                  ; Ende!
                03590
                03600 ;-------------------------------------
                03610 ;        Edit-Unterbefehl C(hange)
                03620 ;-------------------------------------
                03630
4E80 7E         03640 change: ld    a,(hl)         ; Zeilen-Ende erreicht?
4E81 FE0D       03650         cp    0dh            ; Na?
4E83 C8         03660         ret   z              ; Ja, bloß nicht ändern!
4E84 CDBB4E     03670         call  inchar         ; Neues Zeichen anfordern,
4E87 FE20       03680         cp    ' '            ; aber Steuerzeichen
4E89 D8         03690         ret   c              ; ignorieren!
4E8A 77         03700         ld    (hl),a         ; sonst: Zeichen einsetzen
4E8B 23         03710         inc   hl             ; Textzeiger weiterstellen
4E8C CDB34E     03720         call  outcha         ; und Zeichen ausgeben.
4E8F 10EF       03730         djnz  change         ; Wie oft denn noch?
4E91 C9         03740         ret                  ; Na, also: Fertig!
                03750
                03760 ;-------------------------------------
                03770 ;        UP für I(nsert) und D(elete):
                03780 ;        Rest der Zeile ausgeben und Cursor
                03790 ;        an aktuelle Änderungsstelle
                03800 ;        zurückstellen.
                03810 ;-------------------------------------
                03820
4E92 E5         03830 dispt:  push  hl             ; Textzeiger retten
4E93 0E00       03840         ld    c,0            ; Zeichenzähler nullsetzen.
4E95 7E         03850 l1:     ld    a,(hl)         ; Textende erreicht?
4E96 FE0D       03860         cp    0dh            ; Na?
4E98 2807       03870         jr    z,e1           ; Ja!
4E9A CDB34E     03880         call  outcha         ; Sonst: ein Zeichen der restlichen
4E9D 0C         03890         inc   c              ; Zeile ausgeben und mitzählen.
4E9E 23         03900         inc   hl             ; Textzeiger weiterstellen.
4E9F 18F4       03910         jr    l1             ; Weiter auf Zeilen-Ende hoffen.
4EA1 3E1E       03920 e1:     ld    a,30           ; Zeilen-Ende erreicht: Für alle
4EA3 CDB34E     03930         call  outcha         ; Fälle: Bildschirm säubern.
4EA6 E1         03940         pop   hl             ; Textzeiger restaurieren.
4EA7 0C         03950         inc   c              ; Waren überhaupt
4EA8 0D         03960         dec   c              ; Zeichen übrig?
4EA9 C8         03970         ret   z              ; Nein, alles so OK!
4EAA 3E18       03980         ld    a,24           ; sonst:
4EAC CDB34E     03990 e2:     call  outcha         ; Cursor auf Edit-Position
4EAF 0D         04000         dec   c              ; zurück-
4EB0 20FA       04010         jr    nz,e2          ; transferieren!
4EB2 C9         04020         ret
                04030
                04040 ;-------------------------------------
                04050 ;        I/O-Unterprogramme
                04060 ;-------------------------------------
                04070
4EB3 D9         04080 outcha: exx
4EB4 F5         04090         push  af
4EB5 CD3300     04100         call  033h
4EB8 F1         04110         pop   af;
4EB9 D9         04120         exx
4EBA C9         04130         ret
                04140
4EBB D9         04150 inchar: exx
4EBC CD4900     04160         call  49h
4EBF D9         04170         exx
4EC0 C9         04180         ret
                04190
                04195         page

                04200 ;-------------------------------------
                04210 ;        Edit-Zeilenpuffer
                04220 ;        (Der Editor arbeitet
                04230 ;         nicht direkt im DOS-
                04240 ;         Befehlspuffer, da sonst
                04250 ;         der A-Befehl nicht
                04260 ;         möglich wäre!)
                04270 ;-------------------------------------
                04280
4EC1            04290 edibuf: defs  80
                04300
4D00            04310         end   start
00000 TOTAL ERRORS
```

m4/4p ecke

Tape-Games auf dem Model III/4/4p

Viele der alten Model 1-Spiele sind auch auf dem TRS 80 Model 3 und damit auf dem 4/4p im M.3-Modus lauffähig. Ein großes Problem gab und gibt es aber mit den Spielen, die mit LMOFFSET auf einem Model 1 von Tape auf Disk übertragen wurden. Wenn man die vorgeschriebene Procedur einhält, stürzt die Maschine ab bzw. bootet neu. Eine sehr einfache Methode, diese Spiele doch zum Laufen zu bringen, soll hier erklärt werden.

Nachdem man auf dem Model 1 das Programm gestartet hat, muß man RESET drücken und gleichzeitig BREAK halten um ins ROM-BASIC zu kommen. Dieser Vorgang bleibt beim Model 3/4/4p natürlich der gleiche. Auch SYSTEM <ENTER> muß man wie gehabt eingeben. Danach kann man aber nicht einfach durch Eingabe von "/" das Programm starten, sondern man muß zusätzlich die Startadresse mit angeben.

Das ist leichter gesagt als getan. Woher soll man die Startadresse nehmen!? Ganz einfach, man sucht sie sich mit SUPERZAP, DDE, FED oder einem ähnlichen Diskettenmonitor aus dem /CMD-File. Die Startadresse findet man direkt vor der Meldung "Do the following:..." (siehe dazu auch Info Nr. 9, Seite 9/10 "Do it yourself - LMOFFSET with Disable DOS"). Dort steht die Bytefolge C3 xx xx, wobei C3 der Hex-Code ist, der die Z80-CPU dazu veranlasst, ihre Arbeit bei der Stelle fortzuführen auf die die beiden folgenden Bytes zeigen. Die eigentliche Startadresse wird also durch die beiden Bytes nach C3 repräsentiert. Sie sind in der, für die Maschinensprache typischen Reihenfolge LowByte/HighByte geschrieben.

Ein Beispiel verdeutlicht das Ganze. In einem Programm findet man die Bytefolge: C3 90 7A. C3 repräsentiert wie wir wissen den Sprung, die Adresse zu der gesprungen wird lautet 7A90h. Nun braucht man nur noch die Adresse ins dezimale Zahlensystem umzurechnen (wozu hat man seinen Computer!?) und, wie oben beschrieben, nach dem "/" als Startadresse anzugeben. Schon läuft (hoffentlich) das Spiel!!!

Damit man sich nicht jedesmal die Adresse wieder neu suchen muß, kann man sie mit dem Zapper in den "Do the ..."-Text mit einbauen. Das ganze kann dann so aussehen!

```
Tue folgendes:
<BREAK> halten und RESET drücken um L2-BASIC zu starten
<BREAK> loslassen und zwei mal <ENTER> drücken
SYSTEM <ENTER> und dann
/53248 <ENTER> eingeben
```

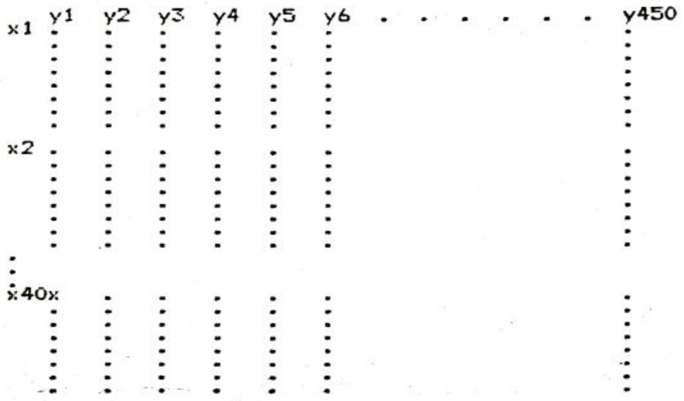

Viel Spaß beim Spiel wünscht

Hartmut Obermann

Umkehr !

Unser Clubkamerad Gerald Dreyer stellte mir vor kurzem eine Aufgabe, bei deren Lösung ein Problem auftrat, welches mich ein paar Stunden stark beschäftigte. Da ich glaube, daß dieses Problem und die/meine Lösung zumindest für die an der Maschinensprache Interessierten und die Tüftler unter uns von Interesse ist, will ich beides hier kurz erläutern.

Die mir gestellte Aufgabe bestand darin Datenfiles, wie sie von dem Grafikprogramm DGRAF (erstellt Grafiken aus DBase-Dateien) erzeugt werden, in die HRG des Model 4p einzulesen. Diese Files sind folgendermaßen organisiert:



Wie man deutlich sieht, handelt es sich um ein Datenformat, welches relativ leicht auf Drucker ausgegeben werden kann. 450 Bytes (=Bildpunkte) pro Zeile mit logischerweise jeweils 8 Bit (=Bildpunkte) in der Höhe. Insgesamt ergibt sich also eine Auflösung von 450 * 320 Bildpunkten. Leider ist das Format für die HRG nur sehr bedingt geeignet. Das hat folgende Gründe:

1. **Die HRG ist byteweise in y-Richtung organisiert, das Bild liegt also, wenn man es direkt einliest, auf der Seite.**
2. **Die HRG hat zwar eine Auflösung von 640 Punkten in x-Richtung aber nur 240 Punkte in y-Richtung.**

Es sind also zwei Dinge zu tun, um die DGRAF-Files einlesen zu können:

1. **das Bild um 90 Grad zu drehen und**
2. **die Bildhöhe um die Hälfte verringern (auf 160 Punkte).**

Dabei trat nun das eigentliche Problem auf, daß die Datenbytes einzeln "umgekehrt" werden mussten. Beispiel:
Folgendes Byte (1) muß nach der Umkehrung wie (2) aussehen!

```
        Bit 7 6 5 4 3 2 1 0
        (1) 1 0 0 1 0 1 1 0
        (2) 0 1 1 0 1 0 0 1
```

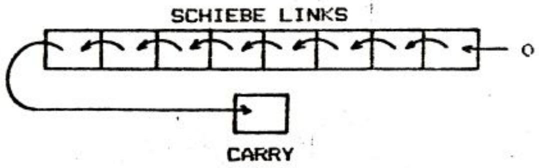
Bit 7 muß also auf Bit 0 verschoben werden, 6 auf 1, usw. usw. 0 auf 7. Nochmal ein Beispiel:

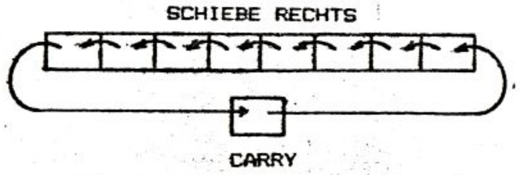
```
        Bit 7 6 5 4 3 2 1 0
        (1) 0 1 0 1 0 1 0 1
        (2) 1 0 1 0 1 0 1 0
```

vor der Umkehr Bit 0 1 2 3 4 5 6 7!!!

Zunächst gedachte ich, dem Problem mathematisch auf den Grund zu gehen und versuchte allerhand Operationen von der Addition bis zum Negieren, einschließlich aller Kombinationen zusammen mit Schiebebefehlen links' und rechts (die Problemlösung wurde in TurboPascal gesucht!). Leider brauchte ich sehr lange bis ich zu der Erkenntnis kam, daß es so nicht zu schaffen ist(?). Hätte ich die Lösung gleich in einer Maschinenspracheroutine gesucht, wären mir warscheinlich etliche Zweifel an meinem (so hoffe ich) klaren Menschenverstand erspart geblieben. Auf alle Fälle kam ich, nachdem ich eine Hochsprachenlösung verworfen hatte, recht schnell zu folgendem Ergebnis.

Die Z80-CPU (und natürlich auch andere Prozessoren) stellt dem Programmierer verschiedenartige Schiebe- und Rotationsoperationen zur Verfügung. Dabei wird der Inhalt eines Registers oder einer Speicherstelle verändert, indem man den Inhalt nach links oder rechts um jeweils ein Bit verschiebt. Den prinzipiellen Unterschied zwischen Schieben und Rotieren könnt ihr am besten folgender Grafik entnehmen:





Wie ihr seht, wird beim Schieben Bit 0 (7) mit einer 0 aufgefüllt, alle anderen Bits wandern eine Stelle nach links (rechts) wobei Bit 7 (0) aus dem/der Register/Speicherst. in das Carry-Flag wandert. Wird ein zweitesmal geschoben, geschieht das gleiche, dabei geht das ursprüngliche Bit 7, welches beim ersten Schieben nach CARRY gewandert war, verloren!
Das Rotieren ist im Prinzip nichts weiter als eine Schiebeoperation, bei der allerdings nicht mit einer 0 aufgefüllt wird, sondern mit dem Wert, der in CARRY stand. Wurde diese Operation neunmal durchgeführt, stellt sich das/die Register/Speicherstelle so dar, als wäre nichts weiter passiert!

Die Schiebe- und Rotieroperationen gibt es in einer großen Vielfalt und in unterschiedlichsten Variationen. Über den ganzen Umfang dieser CPU-Befehle und auch über die Anwendungsgebiete möchte ich mich hier nicht auseinandersetzen und verweise daher auf einschlägige Literatur (wie z.B. Programmierung des Z80 von Rodnay Zaks, in der Bibiothek des CLUB 80 enthalten).

Zwei der Rotierbefehle des Z80 lauten RL s (Rotiere links durch das Carryflag) und RR s (Rotiere rechts durch das Carryflag). "s" steht dabei für eines der Register (A,B,C, D,E,H,L) oder eine durch HL, IX+d oder IY+d adressierte Speicherstelle. Mit diesen Befehlen und ein wenig drumherum kann man ein Byte sehr schnell "umkehren"! Dazu erst einmal folgendes Listing:

m4/4p ecke

```
1   LD    A,(Byte)
2   LD    B,A
    -----------------
3   RL    B        diese beiden Anweisungen müs-
4   RR    A        sen achtmal wiederholt werden!
    -----------------
5   LD    (Byte),A
```

Die folgende Darstellung macht die S[illegible] wahrscheinlich schon erheblich deutlicher! In Byte sol[illegible] Beginn die Zahl 202 (Binär 11001010) stehen.

Inhalt von	Register A Bit 7	6	5	4	3	2	1	0	Carry Bit	Register B 7	6	5	4	3	2	1	0
Nach Anweisung Nr. 1	1	1	0	0	1	0	1	0	?	?	?	?	?	?	?	?	?
2	1	1	0	0	1	0	1	0	?	1	1	0	0	1	0	1	0
3	1	1	0	0	1	0	1	0	1	1	0	0	1	0	1	0	?
4	1	1	1	0	0	1	0	1	0	1	0	0	1	0	1	0	?
8 mal wiederholen! 3	1	1	1	0	0	1	0	1	1	0	0	1	0	1	0	?	0
4	1	1	1	1	0	0	1	0	1	0	0	1	0	1	0	?	0
3	1	1	1	1	0	0	1	0	0	0	1	0	1	0	?	0	1
4	0	1	1	1	1	0	0	1	0	0	1	0	1	0	?	0	1
3	0	1	1	1	1	0	0	1	0	1	0	1	0	?	0	1	0
4	0	0	1	1	1	1	0	0	1	1	0	1	0	?	0	1	0
3	0	0	1	1	1	1	0	0	1	0	1	0	?	0	1	0	1
4	1	0	0	1	1	1	1	0	0	0	1	0	?	0	1	0	1
3	1	0	0	1	1	1	1	0	0	1	0	?	0	1	0	1	0
4	0	1	0	0	1	1	1	1	0	1	0	?	0	1	0	1	0
3	0	1	0	0	1	1	1	1	1	0	?	0	1	0	1	0	0
4	1	0	1	0	0	1	1	1	1	0	?	0	1	0	1	0	0
3	1	0	1	0	0	1	1	1	0	?	0	1	0	1	0	0	0
4	0	1	0	1	0	0	1	1	1	?	0	1	0	1	0	0	0

Nach der letzten Anweisung (5) steht in Byte 83, was, schaut man sich die Zahl einmal binär an (01010011), genau die Umkehrung des Anfangswertes ist! Das Wunder vollbringen die jeweils acht RL- und RR-Operationen, bei denen die Bits, jeweils im Carryflag zwischengespeichert, aus dem Register B in das Register A wandern. Dadurch, daß einmal links und einmal rechts geschoben wird, steht der Wert dann im Zielregister in umgekehrter Reihenfolge zur Verfügung. Im Prinzip ist es übrigens egal, welche Register benutzt werden und ob zuerst links und dann rechts geschoben wird oder umgekehrt. Das Ergebnis bleibt das gleiche!

Natürlich kann man das ganze auch recht elegant in einer Schleife programmieren und muß so nicht acht mal die RL/RR-Operationen hintereinander schreiben. Dabei ist aber zu beachten, daß man bei Verwendung einer Schleife die Ausführungszeit erheblich erhöht. Die Rotierbefehle müssen nämlich auf alle Fälle achtmal durchlaufen werden und zusätzlich kommt dann noch das decrementieren eines Zählers und die Schleifenabfrage dazu. Wo also mehr auf die Ausführungszeit und weniger auf den Platzbedarf geachtet werden muß (was bei der Einbindung in ein TurboPascal-Programm der Fall ist), sollte man die einfache Form einer Schleife vorziehen!

Nun noch ein praktisches Beispiel für die Anwendung dieser Routine. EPSON- und ITOH-Drucker unterscheiden sich in ihrer Grafikfähigkeit besonders darin, daß EPSON die oberste Nadel des Druckkopfes als Bit 7, ITOH jedoch als Bit 0 eines Grafikbytes ansieht. Mit der vorgestellten Routine kann man diese Inkompatibilität ausschalten!

Zuletzt zwei Fragen. Liebe Assembler-Freaks, gibt es eine Möglichkeit das beschriebene Problem schneller zu lösen? Liebe BASIC-, Pascal- und sonstigen Programmierer, gibt es für das vorgestellte Problem eine Hochsprachenlösung, welche in akzeptabler Zeit das gleiche Ergebnis bringt?
Viel Spaß beim Bit-Tüfteln, Euer

Hartmut Overmann

* Keine Angst vor Haifischen! *

Auch von Leuten, die des Lesens & Schreibens durchaus kundig, liest man oft lange Texte, deren Wörter offenbar starr und "unzerreißbar" sind...
Gemeint ist die Scheu vor Trennungen.

Ist es etwa eine besondere Schwäche von *TSCRIPS*?

Jahrelang schrieb ich hiermit und ärgerte mich stets über mein Unvermögen, die Zeilen optimaler zu füllen, weil das herrische, aber doch wohlgemeinte TSCRIPS-Kommando "J=Y" (Augen rechts! Rrrricht't euch!!) lieber Super-Vacua zwischen die Wörter bläst als zuzulassen, daß ein einziger Schütze A...h-aha - wollte sagen: Sad Sack, aus der Reihe tanzt.
Zugegeben: Mein Unvermögen wurde durch eine höchst mystische sogenannte "Benutzer-Anleitung" legitimiert, die mehr verdunkel- als erhellte. Zu tiefergehendem Studium aber war ich zu faul -

bis *Arnulf SOPP*,
ganz salopp,
selber ob
dieser Crux mit Recht empört,
mich das Haifischen gelehrt!

Wieso "Hai"?
Lautschrift erleichtert das Leben!
(Stabreim versehentlich...)

"HYPHEN" = engl. "Trennstrich", gesprochen "HAIFN", war es, was da fehlte und den ernsten Schreiber quälte!
"HYPHENATION" ist jener Vorgang, der in einem aesthetischen Schreibwerk geübt - ja, ich möchte sagen: gepflegt wird. Und für den *TSCRIPS* nach "BREAK" auf das Befehlskürzel "H" in der Kommando- (oder Status-)zeile wartet!.
Verständlich: Schlechte Erklärungen können einem Angst machen, können einen frustrieren; zumal die Anwendung ein wenig Übung erfordert.
Aber dann, erst gekonnt, ist es ein Genuß!
Ich kann gar nicht genug lange Wörter bilden, Wortschlangenausdrückeherstellungsgier ist seitdem mein neuestes Hobby, um möglichst viele Haifische zu fangen! Wie die gegen ihren Willen getrennten Wörter an den Zeilenenden ihre Zungen rausstrecken, als wollten sie mir zurufen: "Bähh - ...und wir gehören doch zusammen!" -- Jawohl, das sollt ihr auch! Aber die Zeilen sind schöner, sind voller geworden. Kein (?) gäähhnender gap mehr!

Nun gut: Auch TSCRIPS ist nicht vollkommen. Es gibt nach wie vor Abstände, die mir unverständlich sind; wenn ich z.B. die überschüssig-überflüssigen Abstände zusammenzähle und feststellen muß, daß da das kurze Wort, mit dem die nächste Zeile beginnt, doch noch hineingepaßt hätte... Dies zu erklären, ist mir zu hoch. Sicher kann das einer unserer ML-Experten.

Hier folgt nun eine kurz gefaßte
Gebrauchsanleitung zum Trennen in TSCRIPS,
da *Arnulf* seine Hilfe sicher nicht auf mich beschränkt wissen wollte:

Trennen in TSCRIPS

1) Setze Bildschirmbreite "Width" = W = RM - LM.

2) Setze den Cursor an den Anfang des zu trennenden Textes.

3) Drücke den Klammeraffen gleichzeitig mit Q und danach den Bindestrich '-'. Hierauf wird der Text vorübergehend ein wenig nach rechts gerückt und es erscheint ein Pfeil nach rechts.

4) Jetzt gehe ans Textende und drücke wieder das Klammeräffchen gleichzeitig mit Q, anschließend - bei festgehaltenem Äffchen! - den Abwärtspfeil. (Das geht schneller, als es sich anhört, weil Q und Abwärtspfeil fast beieinander liegen.)

5) Drücke die BREAK-Taste, danach das 'H' (für 'Hyphenation' = Trennung). Wenn jetzt ENTER (RETURN oder NEW LINE) gedrückt wird, erscheint die bange Frage nach der gewünschten "Heißen Zone". Das bedeutet:
Wähle eine Zahl zwischen 2 und 9 - am zweckmäßigsten die 2 (das heißt, daß bereits nach zwei Buchstaben getrennt werden kann; dann geht am meisten in die Zeile hinein).

6) Nach ENTER wird jetzt ein erster Ort zum Trennen angeboten (und zwar liegt er links vom Cursor). Falls er orthographisch falsch ist, kann der Cursor mit dem Linkspfeil solange nach links gerückt werden, bis eine trennbare Stelle erreicht ist. Dann taste man den "Bindestrich" '-' (der in diesem Fall zu einem Trennstrich wird).
Falls es hier keine trennbare Stelle gibt, gehe man mit ENTER weiter.
Merke: Nach rechts Rücken ist nicht möglich. Das würde den Abzähl-Algorithmus des Systems durcheinander bringen.

7) Wenn es nichts mehr zu trennen gibt - d.h. wenn der Text durchgelaufen ist - meldet sich das System von selbst mit "HYPHENATION COMPLETE".

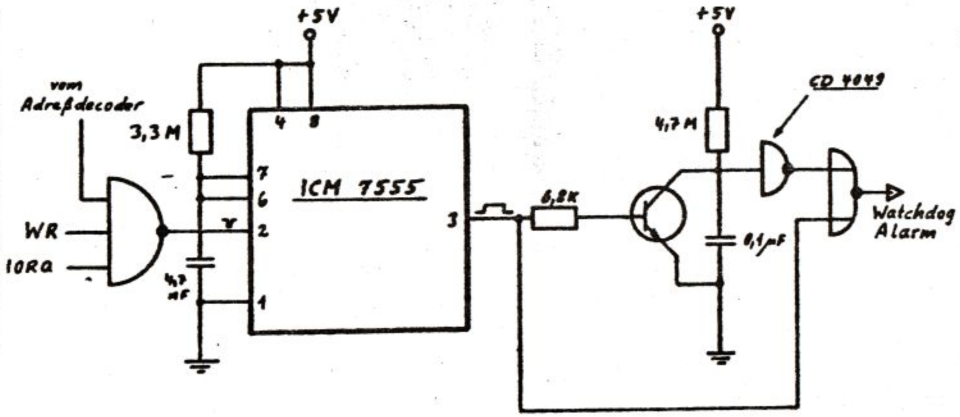
XaJot

Der Wachhund

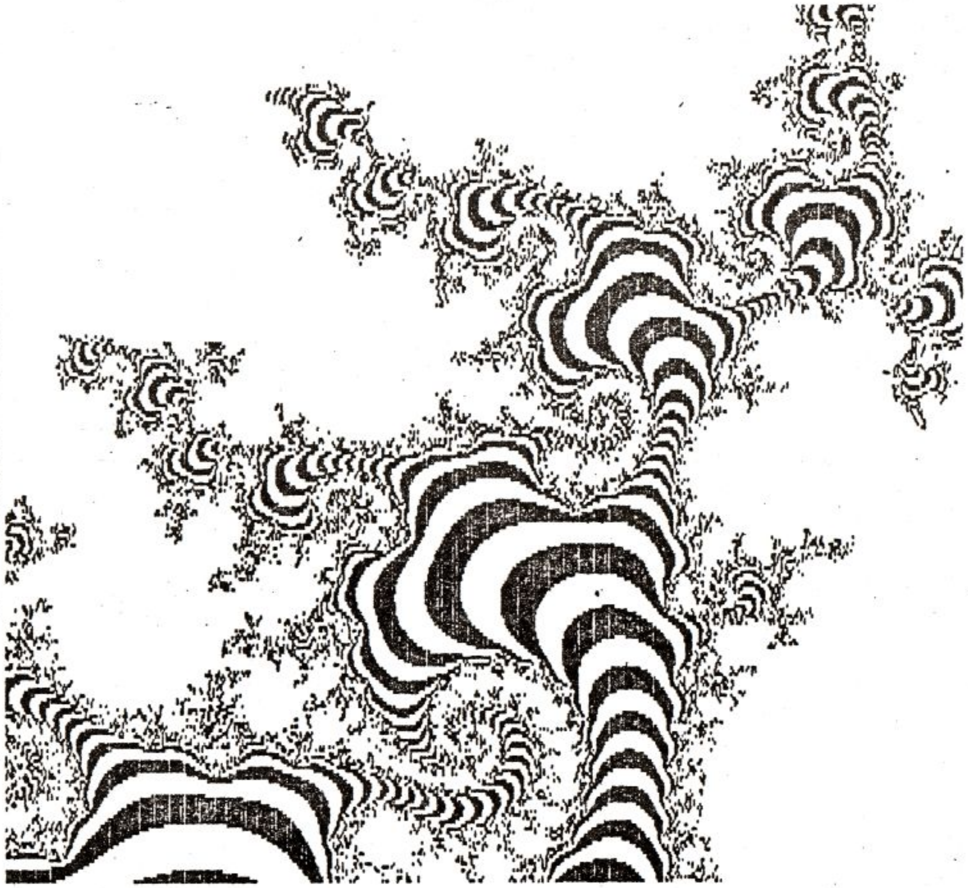
Eine Schaltung, die man aufbaut in der Hoffnung, daß man sie nie braucht, ist der Watchdog. Für den Fall, daß Hardware oder Software außer Tritt kommen, muß verhindert werden, daß unkontrollierte Schaltzustände mit schlimmen Folgen für den Prozeß entstehen. Die Schaltung, die als "Notbremse" funktioniert, erwartet in regelmäßiger Folge (einige Millisekunden Abstand) Impulse vom Hauptprogramm. Bleiben die Impulse aus, wechselt ein Signalpegel, mit dem man z.B. einen NMI-Interrupt auslösen kann, der die CPU in eine definierte Fehlerroutine zwingt. Zusätzlich kann man über einen Leistungstransistor die Stromversorgung der Ausgangsrelais abschalten und damit sämtliche Aktionen im Prozeß stoppen. Gleichzeitig erscheint ein optisches oder akustisches Alarmsignal, das dem Bedienungspersonal anzeigt, daß etwas schief gelaufen ist.

Zum Schluß ein Schaltungsvorschlag für einen Watchdog, der mit wenigen gängigen Bauteilen auskommt, bereits auf einen einzigen ca. 100 ns breiten Eingangsimpuls ragiert und dessen Zeitkonstante sich in einem sehr weiten Bereich durch die Wahl eines Kondensators einstellen läßt.

Heinrich Betz



CLUB 80 HARD - - - CLUB 80 HARD



Auslesen des Graphikspeichers

GDP64-Karte des NDR-Klein-Computers im TRS80 und GENIE

Helmut Bernhardt, Jörg Seelmann-Eggebert

Zwischen der in den Computern TRS80 und GENIE wohl am weitesten verbreiteten HRG 1B und der GDP64 des NDR-Klein-Computers bestehen folgende gravierenden Unterschiede:

Feature	HRG 1B	GDP64
Bildformat	192x384	256x512
Mischbarkeit des Videosignals mit dem des Textbildschirms	ja	nein
Geschwindigkeit der Graphikausgabe	langsam	sehr schnell
Auslesbarkeit des Graphikspeichers	ja	nein
Anzahl Bildschirmseiten	1	4
Programmierbarkeit	umständlich	komfortabel
Anschluß an den Computer	beide frei	verdrahtet
BASIC-Treiber vorhanden	ja	ja
Verschleiß an Portadressen	128	32

Die wesentlichsten Nachteile der GDP64 bestehen in der nicht vorhandenen Auslesbarkeit des Graphik-Speichers, wodurch ein Abspeichern von Bildern auf Diskette, ein Ausdrucken der Graphik, ein Spiegeln und Invertieren des Bildes und die Implementation eines Graphik-Cursors nicht möglich sind.

Die nicht vorhandene Mischbarkeit des Videosignals mit dem des normalen Textbildschirms kann dadurch umgangen werden, daß entweder ein zweiter Monitor benutzt wird oder eine einfache softgesteuerte (und/oder handgesteuerte) Umschaltung des Videosignals für den Monitor dazugestrickt wird. Ein möglicher Schaltungsvorschlag wird zum Schluß noch vorgestellt.

Zunächst aber soll das Hauptproblem, das Auslesen des Graphikspeichers, gelöst werden. Dazu sei vorweg bemerkt, daß wegen des kritischen Timings des EF9366-Graphikprozessors nicht garantiert werden kann, daß jeder Chip dabei funktioniert. Mit einem Pulldown-Widerstand kann der abfallenden Flanke des entsprechenden Signals zwar etwas Dampf gemacht werden, ob das bei dem jeweiligen Exemplar des 9366 dann ausreicht, muß probiert werden. Durch Verringern des Widerstandswertes läßt sich wahrscheinlich auch noch was rausholen.

Für das Auslesen des Graphik-Speichers der GDP64 benötigt man 2 zusätzliche ICs und einen 2k2-Widerstand. Hauptbestandteil ist ein 74LS374-Latch, in das der EF9366 den Inhalt der ihm benannten 8 Bits des Bildwiderholspeichers einträgt und aus dem sich die CPU dieses Datum abholen kann. Außerdem sind noch zwei OR-Gatter zur Ansteuerung des Latch nötig.

Und was muß gemacht werden ?

Nicht viel. Bei dem zusätzlichen 74LS374 werden außer den Pins 10 und 20 alle Pins um 45 Grad hochgebogen. Dann wird IC 20, 74LS245 der GDP64-Karte aus der Fassung gezogen und das 74LS374 mit den Pins 10 und 20 auf den 74LS245 an dessen Pins 10 und 20 angelötet.

Dann wird IC 19, 74LS273 aus der Fassung genommen und auf dem Tisch so neben den eben gebauten Doppeldecker gelegt, wie die ICs 19 und 20 nebeneinander auf der Karte gesteckt haben. Mit möglichst kurzen isolierten Drahtstücken werden dann alle in der folgenden Tabelle aufgeführten Verbindungen zwischen Huckepack-74LS374 und 74LS273 hergestellt. Beim 74LS273 sollten die Drahtenden möglichst weit oben am Pin angelötet werden, damit die Pins nachher auch wieder in die Fassung passen. Danach kann das Verbundsystem wieder in die Fassungen gesteckt werden.

Nun werden die Verbindungen des Doppeldecker-74LS374 mit den Pins des 74LS166 entsprechend der rechten Hälfte der Tabelle hergestellt, wobei dann der 74LS166 aus der Fassung zu ziehen und locker auf diese zu stellen ist (damit sich die Drahtlängen richtig bemessen lassen und andererseits beim Anlöten keine dauerhafte Verbindung zwischen IC und Fassung hergestellt wird; das entspräche dann auch nicht mehr dem Sinn einer Fasung).

Verbindungen zwischen dem Huckepack-LS374 und IC 19 sowie IC4

Signal !	IC19, LS273 Pin	Huckepack-LS374 Pin	Pin	IC4,LS166 Pin
D1 !	3 -------	2	3 -------	12
D3 !	4 -------	5	4 -------	10
D5 !	7 -------	6	7 -------	4
D7 !	8 -------	9	8 -------	2
D6 !	13 -------	12	13 -------	3
D4 !	14 -------	15	14 -------	5
D2 !	17 -------	16	17 -------	11
D0 !	18 -------	19	18 -------	14

Nun wird ein zusätzliches 74LS32 genommen und dessen Pins 1-4, 6 und 8-13 ebenfalls um 45 Grad hochgebogen. Mit den Pins 5, 7 und 14 wird es auf dem aus der Fassung genommenen IC8, 74LS32 an dessen Pins 5, 7 und 14 angelötet. Nun kann auch dieser Doppeldecker wieder in seine Fassung gesteckt werden. Eventuell müssen die abespreitzten Pins der beiden Huckepack-ICs noch etwas zurechtgebogen werden, damit sie sich nicht gegenseitig berühren und auch keinen Kontakt mit ihren Täger-ICs haben. Dann werden folgende Verbindungen hergestellt:

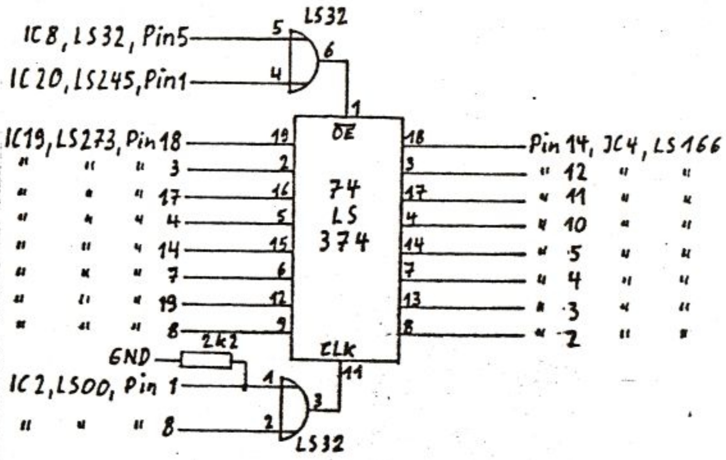
Huckepack-LS32 Pin	Verbinden mit
1	IC2,74LS00,Pin1
2	IC2,74LS00,Pin8
3	Huckepack-LS374,Pin11
4	IC20,74LS245,Pin1
6	Huckepack-LS374,Pin1

Und schließlich werden noch die Pins 1 und 7 des IC2, 74LS00 über einen 2k2-Widerstand verbunden (Pulldown des MFREE-Signals des EF9366).

Damit ist alles getan, um die dem EF9366 vorgegebene Speicherstelle auslesen zu können.

Wenn danach das Auslesen des HRG-Speichers nicht funktioniert, ist das nicht verwunderlich, weil man zunächst etwas falsch gemacht hat; zumindest ich (Helmut) bin das so gewöhnt. Dieser Fehler kann aber durch Vergleich des Strippengewirrs mit der Einbauanleitung schnell gefunden werden.

Möglicherweise treten dann beim Auslesen immer noch Fehler auf. Die Ursache liegt dann beim oben beschriebenen Timing des EF9366 mit seinem MFREE-Signal. Man kann dann noch mit einem kleineren Widerstand als 2k2 Versuche machen oder eventuell einen anderen EF9366 ausprobieren. Letztere Variante kann natürlich recht kostspielig werden, bis man ein funktionierendes Exemplar gefunden hat. Alternativ kann man sich damit abfinden, daß es nicht funktioniert und alle Umbauten wieder abreißen (weniger erstrebenswert).



Und was bringt die GDP64 gegenüber der HRG 1B mehr ?

An Software-Unterstützung gibt es dank der Bemühungen von Christof Ueberschaar ein 4K-Byte großes BASIC-Treiberprogramm, das ins HIMEM geladen wird und nach Einbinden in das Disk-BASIC sehr komfortable Graphik-Befehle bereitstellt. Gegenüber der Variante des Treibers für die HRG 1B, wo die Graphik-Befehle durch ein # eingeleitet werden und daher der BASIC-Interpreter bei jedem Befehl erst prüfen muß, ob ein # vorliegt, ist hier der CMD-Vektor verbogen worden. Es wird dann nur noch bei Auftauchen des CMD-Tokens im Programmtext eine zusätzliche Abfrage gemacht, ob das nach CMD obligatorische (") folgt und nur dann, wenn das nicht der Fall ist, weiter untersucht, ob das Byte ein gültiger Graphik-Befehl ist, der dann abgearbeitet wird.

Diese Philosophie hat zwar den Nachteil, daß man anstelle des einfachen # jedesmal 3 Zeichen eintippen muß, bevor der eigentliche Graphik-Befehl kommt, andererseits wird dadurch nicht die Abarbeitung der normalen BASIC-Befehle verlangsamt, womit ein Teil des Geschwindigkeitsvorteils der GDP64 verschenkt wäre.

Außerdem ist die Punktadressierung der GDP64 bedeutend einfacher als die der HRG 1B. Es werden dem EF9366 einfach die X- und Y-Koordinaten des Punktes als 16-Bit-Wörter in entsprechende Register geschrieben und dann einer der Befehle "Punkt setzen", "Punkt löschen" oder "Punkt abfragen" ausgeführt. Bei der HRG 1B bedeutet das ein umständliches Berechnen der Speicheradresse und des entsprechenden Bits im Byte.

Noch gravierender ist die Eleganz beim Zeichnen von Linien über Vektorbefehle. Hier werden die Punktkoordinaten (X1/Y1) für den Anfangspunkt einer Linie in die entsprechenden Register eingetragen und in zwei weitere Register werden die Werte Delta X und Delta Y eigetragen. Es sind dies die Differenzen X2-X1 und Y2-Y1 (Koordinaten von Anfangs- und Endpunkt einer Linie). Diese Differenzen müssen stets positive Zahlen sein. Das Vorzeichen (Vektorrichtung) wird durch entsprechende Bits in den Vektorbefehlen vorgegeben. Es läßt sich auch noch festlegen, ob die Linie durchgezogen, gepunktet oder punktstrichliert ausgegeben werden soll, oder ob die Linie überhaupt gezeichnet werden soll oder eine vorher hier gezeichnete Linie wieder gelöscht werden soll.

Wenn die Line dann mit einer Geschwindigkeit von 1,5 Mio Punkten pro Sekunde gezeichnet ist, stehen automatisch in den X- und Y-Koordinatenregistern die Koordinaten X2 und Y2 der eben gezeichneten Linie, so daß von deren Endpunkt durch Vorgabe neuer Werte für Delta X und Delta Y gleich ein weiterer Vektor gezeichnet werden kann, ohne daß die Endpunktskoordinaten der vorherigen Linie noch als Anfangskoordinaten der neuen Linie ausgegeben werden müssen.

Dann hat der EF9366 noch einen eingebauten Zeichengeneratort (der aber leider nur die amerikanischen Zeichen kennt), mit dem ein komfortables Einbinden von Texten in die Grapik möglich ist. Außer Aufrecht- und Schrägschrift sowie in Breite und Höhe voneinander unabhängig in 16 Stufen in der Größe varierbaren Zeichen läßt sich vorgeben, ob normal von links nach rechts, auf dem Kopf stehend von rechts nach links oder um 90 Grad gedreht von unten nach oben oder von oben nach unten geschrieben werden soll. Für die Textausgabe ist vor der Ausgabe der Textzeichen neben dem Festlegen dieser Modi auch noch der Fußpunkt des ersten Zeichens in die Koordinatenregister einzutragen. Nach der Ausgabe eines Zeichens erhöht sich der Pointer auf den Fußpunkt des dahinter auszugebenden Zeichens, wobei die oben beschriebenen Modi natürlich berücksichtigt werden.

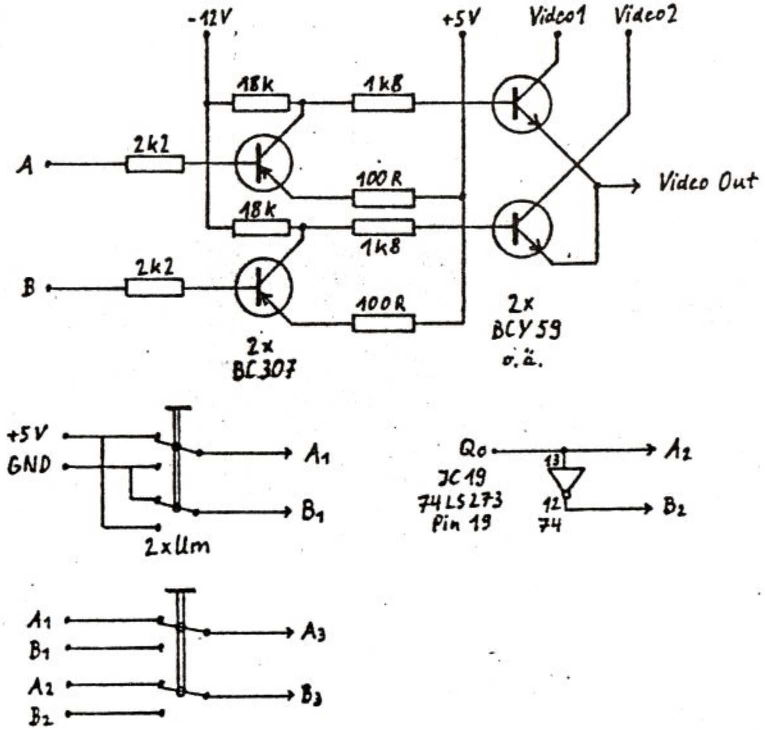
Diese Bemerkungen sollen keinesfalls als erschöpfende Beschreibung der Leistung der GDP64 zu werten sein. Für die Programmierung des EF9366 kommt man ohne Datenblatt nicht aus. Es sollte hier nur ein ungefährer Eindruck vermittelt werden, wofür der Aufwand überhaupt getrieben wurde, bei der unsicheren Erfolgsausicht, den Hardware-Patch überhaupt anzufangen.

Das Umschalten der Video-Signale

Da die Video-Signale des Text/Bauklötzchen-Graphik-Interface des Computers und der GDP64-Karte nicht direkt miteinander mischbar sind, muß mit einer Umschaltung zwischen den Signalen gearbeitet werden. Für Experimentierzwecke reicht dazu ein einfacher Umschalter, mit dem abwechselnd eines der beiden Video-Signale durchgeschaltet werden kann. Die Bezugsmassen der beiden Signale sind miteinander und mit der Abschirmung des Kabels zum Monitor zu verbinden.

Als Dauerlösung ist aber eine Softumschaltung wesentlich eleganter. Dafür kann das Bit 0 des unter Port 60H decodierten Latch (IC19,LS273), mit dem auch die HRG-Speicherbanks angewählt werden, mißbraucht werden. Es ist dann

noch ein Inverter nötig, den IC9, 7404, Pins 13 und 12 liefert. Dabei ist zu beachten, daß mit einem OUT-Befehl an Port 60H dann sowohl die für Auslesen und Schreiben einzustellenden Banks als auch die Selektion des Video-Signals gesteuert werden. Da dieses Latch sich nicht auslesen läßt, muß über dessen Inhalt im RAM Buch geführt werden, damit beim Ändern der Bankanwahl nicht das Video-Signal umgeschaltet wird und umgekehrt. Außerdem sollte der Pin 1 des IC 19, 74LS273 nicht mit +5V sondern mit RESET* (oder einem aus RESET* und NMI* erzeugten SYSRES*) verbunden werden, damit durch Löschen des Latch beim Einschalten (bzw. Drücken des RESET-Knopfes, wodurch ein NMI* erzeugt wird) ein definierter Zustand vorliegt.



In Abb.2 ist oben eine prinzipielle Schaltung zur Umschaltung der Video-Signale (Video1 und Video2) gezeigt, deren Steuereingänge A und B entgegengesetzte logische Pegel haben müssen. Wenn anstelle von -12V nur -5V zur Verfügung stehen, kann auch -5V verwendet werden, wenn anstelle der beiden 18K-Widerstände solche von 7k5 genommen werden.

Unten in Abb.2 sind zwei Möglichkeiten für die Beschaltung der Steuereingänge A und B gezeigt, womit sich entweder durch Hand- oder durch Softumschaltung das gewünschte Video-Signal selektieren läßt. Und schließlich ist noch die Umschaltung der Umschaltungsvarianten per Hand gezeigt, wofür dann noch ein zweiter 2fach-Umschalter nötig ist.

GRAPHIC-TREIBER FÜR EF9366 © 1984 UE-SOFT

Start: im NEWDOS/80 mit 'DO GRAPHIC'

Befehle:

Linie:	CMDLINE(x1,y1)-(x2,y2),S/R[P][D] x1,y1=von-Koordinaten, x2,y2=bis-Koordinaten S=set, R=reset, P=punktierte Linie, D=gestrichelte Linie
Box:	CMDLINE(x1,y1)-(x2,y2),S/R[B/BF] x1,y1=Eckpunkt, x2,y2=gegenüberliegender Eckpunkt B=Box, BF=gefüllte Box
Kreis:	CMDSQR(x,y),r,S/R[F] x,y=Mittelpunkt, r=Radius F=gefüllter Kreis
Set:	CMDSET(x,y) x,y=Koordinaten des zu setzenden Punktes
Reset:	CMDRESET(x,y)
Position:	CMDPOS(x,y)
Print:	CMDPRINT[(x,y)]"Text";[(x,y)]Variable x,y=Print at-Koordinaten
Löschen:	CMDCLEAR[(x,y)]"Text";[(x,y)]Variable
Schriftart:	CMDSIZE(b,h),r b=Breite (1-16), h=Höhe (1-16), r=Richtung (A-D)
CLS:	CMDCLS
Invers:	CMDNOT
Teilinvers:	CMDFIELD(x1,y1)-(x2,y2)
Hardcopy:	CMDLPRINT
Sichern:	CMDSAVE"Filename"
Laden:	CMDLOAD"Filename"
GDP-Regs:	CMDRSET
Spiegeln:	CMDFLEX
Linie bis:	CMDDRAW(x,y),S/R[P][D]

Tastatur- und Druckerplatine (ECB) **Bernd Drowälder**

Aufbau des Tastaturteiles:

Die Tastatur des TRS80 ist in Form einer 8x8 Matrix aufgebaut. Dabei werden die Adressleitungen A0-A7 über Inverter mit Open-Kollektorausgängen auf die Matrixzeilen geführt und die Matrixspalten werden über 8 invertierende Treiber auf den Datenbus des TRS80 gelegt. Die Tastatur wird vom Rechner als Speicherbereich gesehen. Sie belegt den Speicher von 3800H bis 38FFH. Mit Hilfe des Keyboard-Signales werden die Tastaturspalten über die Treiber auf den Datenbus geschaltet. Das KYBD-Signal decodiert den Bereich 38XXH. Dieses geschieht mit Hilfe eines 74LS138, einem 3-Bit Binärdecoder.

Erzeugung des KYBD-Signales:

Der Ausgang Y7 des Binärdecoders nimmt nur L-Pegel an, wenn A11,A12 und A13 H-Pegel und die Adressleitungen A8-A10,A14 und A15 auf L-Pegel liegen. Da aus der Tastaturmatrix nur ausgelesen werden kann, wird außerdem das Signal RD* zur Decodierung benutzt.

Die Adressleitungen A0-A7 wurden bisher nicht berücksichtigt. Diese Adressen werden benötigt, um die Tastaturreihen auszuwählen, die ausgelesen werden sollen. Über eine der Adressen A0-A7 wird eine der Matrixreihen ausgewählt. Über den Inverter mit Open Kollektor-Ausgang wird auf die Matrixreihe Low-Potential gelegt. Wird jetzt eine Taste gedrückt, zieht das Low-Potential am O.K.-Inverterausgang die Spalte, die über einen Pull-Up-Widerstand auf 5V liegt, auf L-Potential herab. Über den invertierenden Treiber gelangt ein H-Pegel auf die der Spalte entsprechenden Datenleitung. Über eine Tabelle erkennt der Rechner dann, welche Taste vom Benutzer gedrückt wurde.

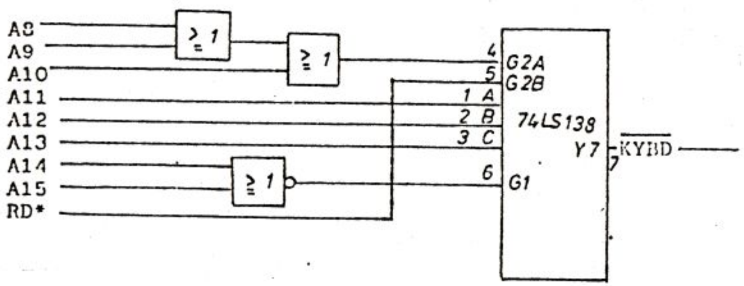
Aufbau des Druckerteiles:

Der TRS80 hat die Möglichkeit, eine Centronics 8-Bit Parallel-Schnittstelle anzuschließen. Die Steuerung dieser Schnittstelle erfolgt Memory-Mappes über die Adresse 37E8H.
Die Adresse 37E8H wird mit einem 74LS133, drei NOR-Gattern und einem Inverter ausgewählt.
Der 74LS133 ist ein NAND-Gatter mit 13-Eingängen. Der Ausgang eines NAND wird nur LOW, wenn alle Eingänge auf H-Pegel liegen. Die Adressen, die schon auf HIGH liegen werden direkt mit dem NAND-Gatter verbunden, die anderen werden über die NOR-Gatter und dem Inverter auf H-Pegel gebracht. Der Ausgang wird also nur LOW, wenn die Adresse 37E8H anliegt. Um den Drucker steuern zu können wird das Ausgangssignal mit den Leitungen RD und WR verknüpft, um sowohl über die Adresse 37E8H Schreiben als auch Lesen zu können.
Das Signal 37E8H WR gibt einen Puls an das 8-Bit D-Register, welches daraufhin die anliegenden Daten übernimmt und an den Ausgang des D-Registers weiterleitet. Außerdem wird mit diesem Signal über ein Monoflop ein 1 us langer STROBE Puls erzeugt, der dem Drucker mitteilt, daß gültige Daten zum Drucken anliegen.

Mit dem 37E8H RD Signal werden 4 Treiber auf den Datenbus des Rechners durchgeschaltet. Der Rechner kann über diese Leitungen den momentanen Status des Druckers abfragen.

Um die Druckerkarte auch alleine mit dem normalen TRS80 nutzen zu können, werden 2 Dioden eingefügt, die über die GDT-Leitung die Treiber zum Rechner durchschalten.

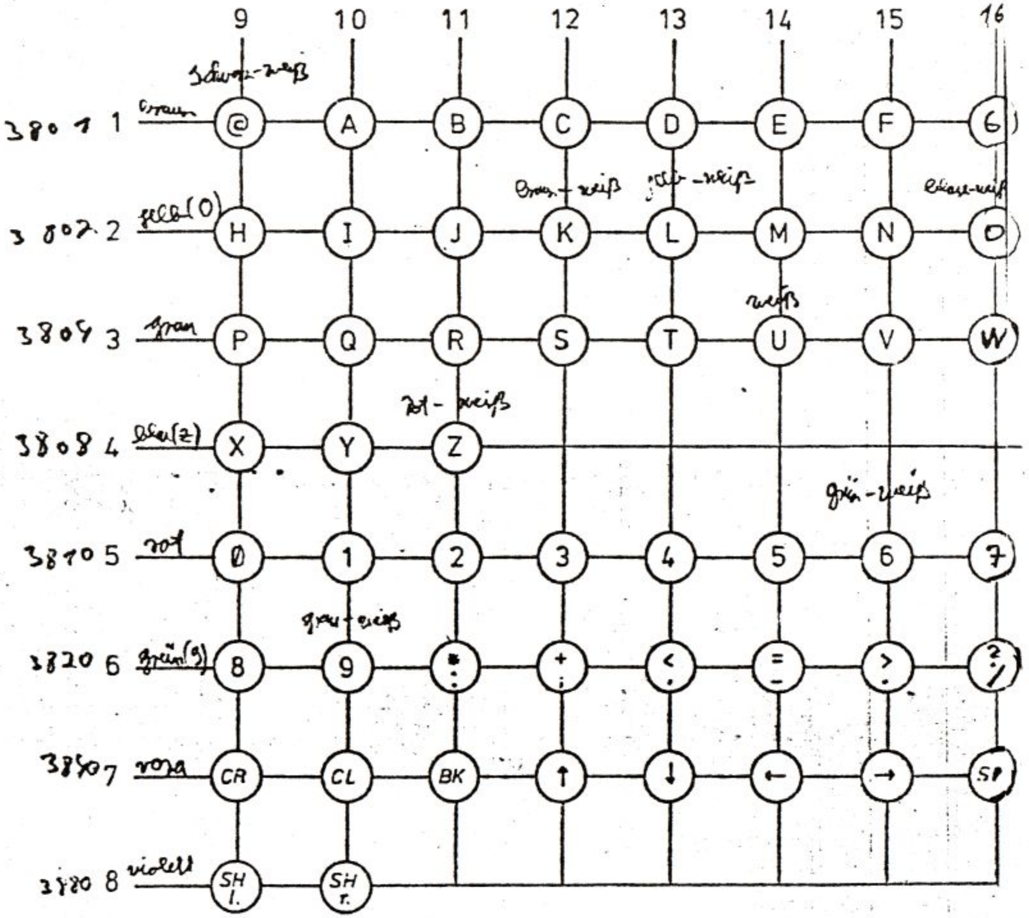
Erzeugung des KYBD-Signales:

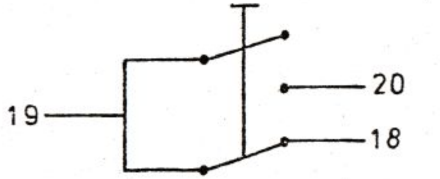


4.1 Anschlußplan des Druckersteckers

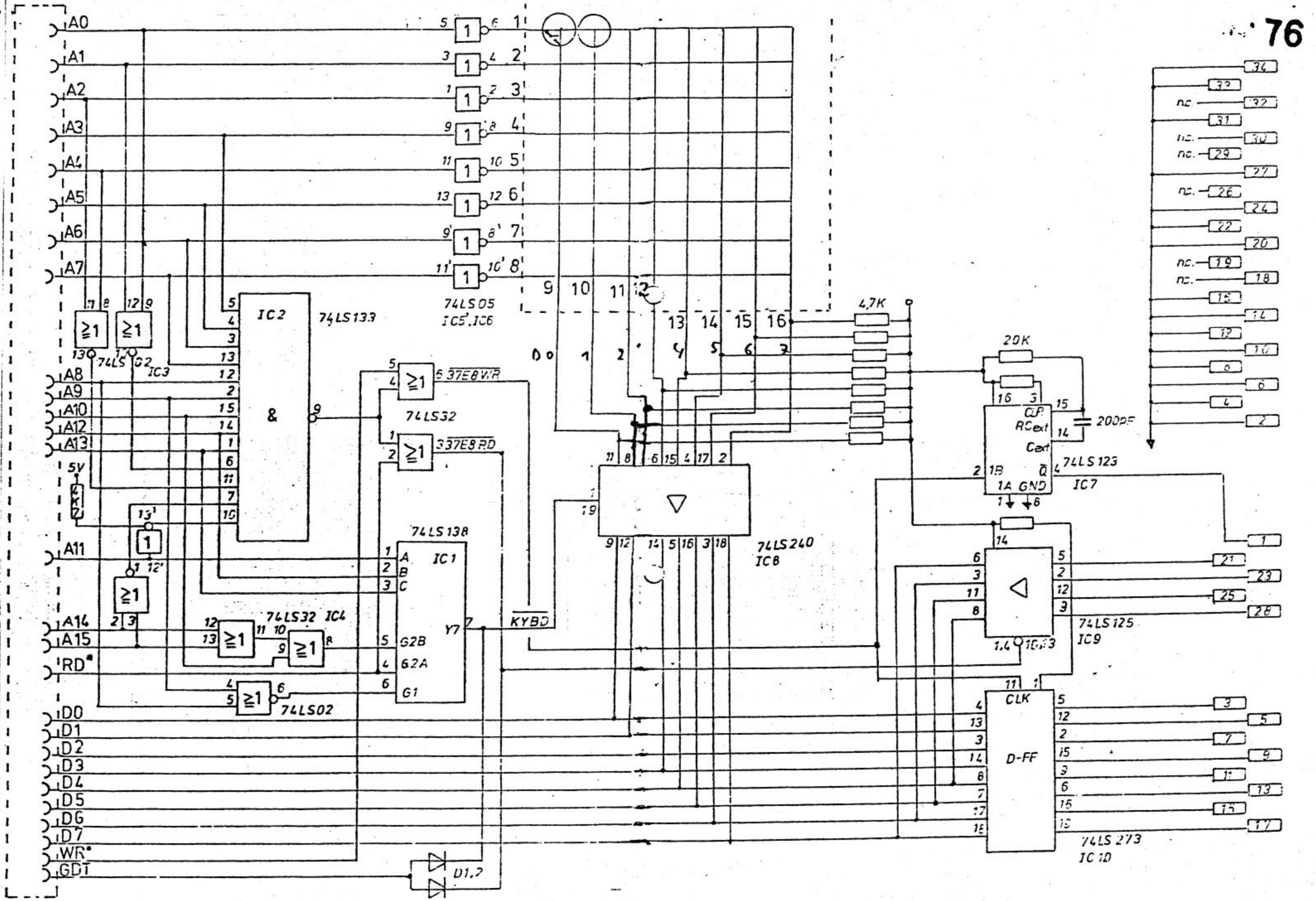
Leitung	Signalname	Beschreibung des Signals
1	DATA STROBE	Gültige Daten liegen an
2	GND	
3	D1	
4	GND	
5	D2	
6	GND	
7	D3	DATENLEITUNGEN
8	GND	
9	D4	D1 - D8, die zum Drucker
10	GND	
11	D5	geführt werden
12	GND	
13	D6	
14	GND	
15	D7	
16	GND	
17	D8	
18	GND	
19	NC	Nicht verbunden
20	GND	
21	BUSY	Es werden noch Daten verarbeitet
22	GND	
23	OUT OF PAPER	Kein Papier mehr
24	GND	
25	UNIT SELECT SLCT	Ein Drucker ist ausgewählt
26	PRIME*	Löscht den Drucker-Buffer
27	GND	
28	FAULT*	Fehlermeldung
29	NC	Nicht verbunden
30	NC	Nicht verbunden
31	GND	
32	NC	Nicht verbunden
33	GND	
34	GND	

4.1.1 Tastaturmatrix

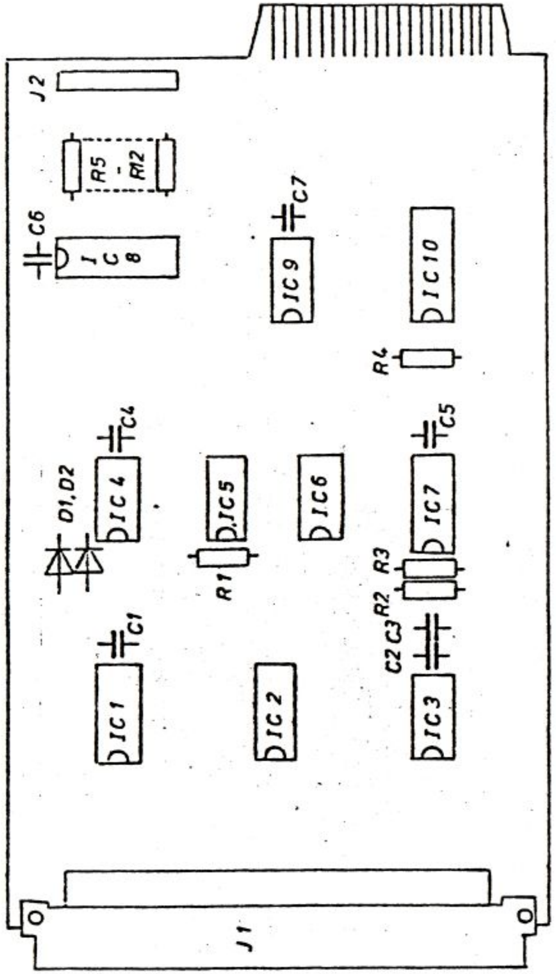




4.2 Schaltplan Drucker, Tastatur



4.3 Bestückungsplan Tastatur- und Druckerkarte



4.4 Bestückungsliste Tastatur- und Druckerkarte

Halbleiter:

IC1	74LS138
IC2	74LS133
IC3	74LS02
IC4	74LS32
IC5	74LS05
IC6	74LS05
IC7	74LS123
IC8	74LS240
IC9	74LS125
IC10	74LS273

Widerstände:

R1,R3,R4,R5-12	4,7 kOhm
R2	20 kOhm

Kondensatoren:

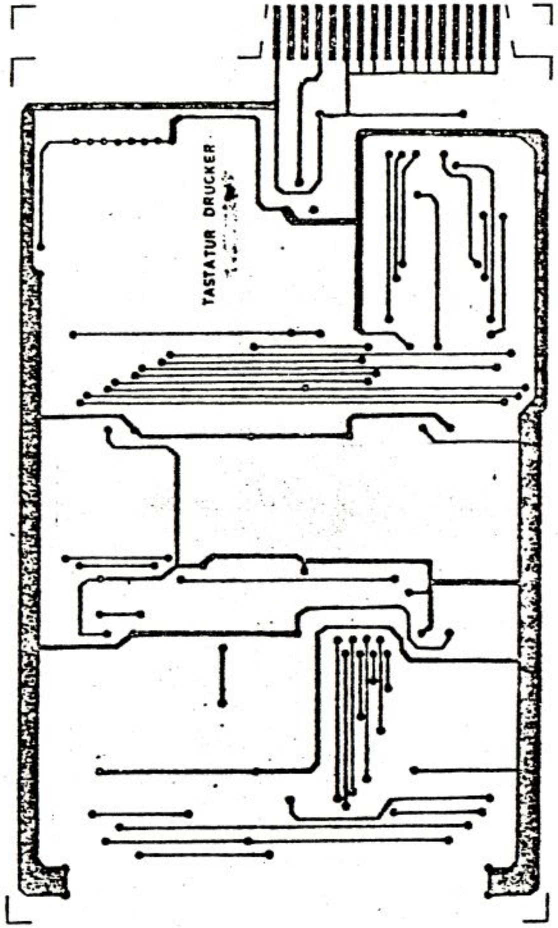
C1,C2	1oo nF
C3	200 pF
C3-C7	100 nF

Sonstige Bauelemente:

D1,D2	1N4148
J1	64 pol. Steckerleiste
J2	20 pol. Steckleiste, abgewinkelt

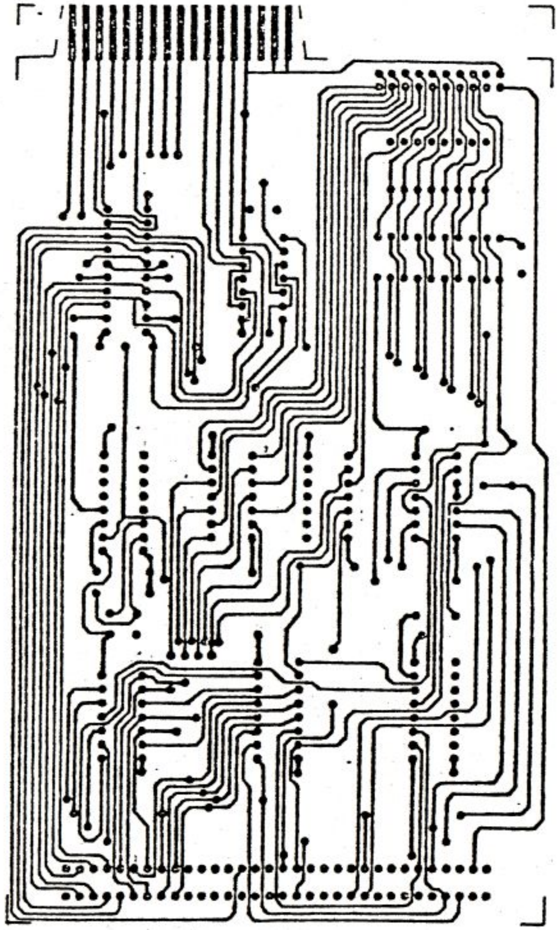
4.5 Platinenlayout Tastatur- und Druckerkarte

Bestückungsseite



Platinenlayout

Lötseite



Unser Clubfreund Richard Rensch hat vor einiger Zeit, ich glaube es war im Info Nr. 18, um Hilfe beim Anschluß eines Laufwerks an sein Genie III gebeten. Nach einiger Zeit und einem regen Brief- und Schaltplanwechsel konnten wir das Laufwerk, das eigentlich für den Anschluß an einen PC-Kompatiblen gedacht war, doch noch dazu überreden am Genie III zu funktionieren. Seit dieser Zeit wollte ich möglichst viele der Informationen, die ich in den verschiedensten Quellen entnommen habe, zu einem Beitrag über die Shugart-Schnittstelle zusammenfassen. Dies hat sich nun erledigt! Der folgende Artikel aus der c't läßt wohl keine Fragen mehr offen und dürfte für die Allgemeinheit interessant sein. Viel Spaß bei der Lektüre, euer

Hartmut Obermann

Anschluß gesucht

Floppy-Laufwerke zum Laufen gebracht – vom sogenannten Shugart-Bus und ähnlichen 'Standards'

Willi Wagemuth, Detlef Grell

'Shugart-Bus', neuerdings auch 'ANSI-Bus', und 'Standard' sind die Schlagworte, die fast immer als erste fallen, wenn die Rede auf den Anschluß von Floppy-Laufwerken an Rechner kommt. Folglich sollte es doch ein leichtes sein, ein Floppy-Laufwerk erfolgreich an einem Floppy-Controller zu betreiben, wenn beide mit eben diesem standardisierten Bus-Anschluß versehen sind. Im Prinzip ja, aber mit dem armseligen Rüstzeug Standard ist es bei weitem nicht getan.

Wir haben nichts gegen Standards, ganz im Gegenteil! Nur wenn sich keiner dran hält, kann man gut drauf verzichten – weshalb wir im folgenden auch genau das tun und die Realität lediglich anhand des Begriffes 'Floppy-Bus' beschreiben werden.

Nicht ohne Papiere!

Wichtigste Vorbedingung, die es in fast allen Fällen ermöglicht, auch mit unterschiedlichsten Laufwerken und Controllern zum Zuge zu kommen, ist eine ausreichende Dokumentation von beiden. Dabei reicht beim Controller im allgemeinen die Beschreibung der wirklich angeschlossenen und bedienten Bus-Signale aus.

Beim Laufwerk braucht man zunächst dasselbe, es sind aber vor allem die Steckbrücken, die einen entscheidenden Einfluß auf die Funktion des Laufwerks und zum Teil auch auf die Belegung der Bus-Schnittstelle haben. Wenn das Laufwerk nicht auf Anhieb 'spielt' und man vor dem Problem steht, zehn oder manchmal bis zu dreißig Jumper auf gut Glück zu variieren, sind Frust oder gar Beschädigungen der Hardware vorprogrammiert.

Zwar werden wir Ihnen im folgenden eine Fülle von Tips geben, die einen teilweisen Dokumentationsmangel kompensieren können, aber geben Sie nicht uns die Schuld, wenn sich Ihre 50-Mark-Glückskauf-Floppy als böser Flop entpuppt.

Alles Abweichler

Die Inkompatibilitäten zwischen Floppy-Drives (und in ähnlicher Weise zwischen Controllern) sind aus dem langsamen Wachsen der Anforderungen an die Laufwerke und den steigenden Möglichkeiten derselben zu erklären. Der Urvater des Floppy-Bus ist der 50polige Bus für 8-Zoll-Laufwerke, wobei man gleich anfügen muß, daß nur die Hälfte als Signalleitungen benutzt wird, da die anderen Leitungen Massepotential führen.

Bereits damals wurde die Firma Shugart mit ihrem recht sinnvollen Bus-System bekannt, das den meisten Erfordernissen genügte. Allerdings hat dieser Bus nicht sofort in der Branche Nachahmer gefunden; Control Data zum Beispiel hatte recht abweichende Vorstellungen davon, wie 8-Zoll-Laufwerke bedient werden müssen – Besitzer gebrauchter 8-Zoll-Laufwerke wissen das zu beklagen.

Aber auch wenn man tatsächlich ein Laufwerk mit angeblichem Shugart-Bus findet, so passen zwar oft die Stecker, seltener jedoch alle Signale zusammen. Und wenn man Pech hat, paßt zwar all das Genannte, aber die zeitliche beziehungsweise logische Ansteuerung durch den Controller erweist sich als abweichend.

Die Vielzahl von Steuerleitungen (die zum Teil bei den neueren 5.25-Zoll-Drives entfallen sind) und eine Fülle von Optionen, die per Steckbrücken einstellbar sind, erfordern bei 8-Zoll-Laufwerken genaue und ausführliche Unterlagen, will man sie in 'endlicher Zeit' in Betrieb nehmen. Um diesen Beitrag nicht unnötig aufzublähen, werden wir aber nur noch mit einigen Randbemerkungen auf die inzwischen doch weitgehend veralteten 8-Zöller eingehen.

Etwas günstiger ist die Situation auf dem Markt für 5.25-Zoll-Laufwerke. Hier konnten sich die Hersteller tatsächlich auf so etwas wie einen Standard einigen. Durch Wegfall einiger Optionen, die sich schon bei den 8-Zoll-Drives als entbehrlich erwiesen hatten, beziehungsweise durch Erhöhung der Laufwerks-Intelligenz konnte man den eh nie ganz ausgenutzten 50poligen Bus auf 34 Leitungen abspecken.

Die ersten 'kleinen' Laufwerke boten noch nicht so viele Betriebsvarianten wie heutige Drives (auf der Laufwerksplatine war gerade Platz für die grundlegende Schreib-/Lese- und Steuer-Logik). So wurden anfangs nur die Signale an den Pins 8 bis 30 (siehe Kasten) und bald auch 32 belegt, allerdings von so vielen Laufwerksherstellern einvernehmlich, daß Abweichungen von der hier beschriebenen Form quasi nicht mehr zu finden sind. Kann man sich bei den heutigen Laufwerken auf diese Grundfunktionen beschränken, so hat man auch wenig Probleme: Fast alle Laufwerke und Controller arbeiten auf dieser Ebene zusammen.

Dennoch soll nicht vergessen werden, daß die Firma Philips anfangs einen recht stark abweichenden Bus für 5.25-Zoll-Systeme propagiert hat. Die meisten Signale auf diesem Bus entsprechen zwar den bisher genannten, liegen aber auf anderen Pins (der Stecker ist nämlich 50polig, wenn auch nicht etwa zum 8-Zoll-Shugart-Bus kompatibel). Die wichtigste Abweichung ist, daß für jedes Laufwerk eine eigene 'Ready'-Leitung zur Verfügung steht (Tabelle 3).

Mehrfachbelegung

Der 'Ärger' begann, als statt ursprünglich maximal drei nun auch vier Laufwerke selektiert werden sollten und die Laufwerke ihren Status (Ready-Signal) übermitteln konnten. (Bis dahin hat man die Steuersoftware im Rechner gemäß den Angaben der Laufwerkshersteller halt so lange warten lassen, bis man sicher sein konnte, daß das Laufwerk bereit war.)

Hier kochten nun verschiedene Hersteller wieder ihr eigenes Süppchen. So findet man die Ready-Leitung mal auf Pin 34, mal auf Pin 6 oder aber auch gar nicht. Vielleicht liegt auf Pin 6 aber auch das Select-Signal für das vierte Laufwerk (siehe Kasten). Ebenso ungewiß ist, ob das Signal 'Head Load' auf Pin 2 oder Pin 4 oder überhaupt zu finden ist (manche einfachen Laufwerke lassen den Kopf nach Einlegen der Diskette dauernd auf der magnetischen Oberfläche 'schleifen').

Dann gibt es noch das Signal 'In Use' (in Gebrauch), mit dem das Laufwerk in Zugriffspausen, wenn der Rechner anderweitig beschäftigt ist, bei Laune gehalten werden kann, damit der nächste Zugriff schneller geht (Kopf bleibt geladen). Neuere Laufwerke kennen oft noch eine Formatumschaltung von normalen Disketten auf solche mit hoher Schreibdichte, und manche haben Signale, mit denen ein Diskettenwechsel signalisiert wird.

Für all diese Funktionen stehen aber nur vier Leitungen (2, 4, 6, 34) zur Verfügung. Diese vier genannten Leitungen werden also je nach Leistungsvermögen des Laufwerks und Laune des Herstellers belegt und führen typischerweise zu Anschlußproblemen – aber wie soll ein Hersteller auch die nicht standardisierten Zusatzfunktionen anders unterbringen?

Ist es schon schwierig genug herauszufinden, wie man nur ein solches Laufwerk an einem Floppy-Controller zum Laufen bringt, so beginnt beim Anschluß eines weiteren Drives mit anderen Bus-Varianten das Spiel aufs neue. In beiden Fällen sind bei arglosem Anschluß ('Ist doch alles Shugart-Bus, oder?') durchaus Beschädigungen der Elektronik möglich, wenn nämlich Controller-Ausgänge (Select, Head Load) mit Floppy-Ausgängen (Ready, Disk Change) zusammenprallen.

Problematisch kann es auch werden, wenn ein Laufwerk kein Ready-Signal liefert, dieses auf dem Controller aber benötigt wird. Da hilft nur 'Tricksen', indem man etwa das Select-Signal des betreffenden Laufwerks dem Controller selbst (über ein Open-collector-Gatter, nichtinvertierend, oder eine Germaniumdiode) wieder als Ready anbietet.

Oder, wenn das Laufwerk sehr lange zum Start braucht, bastelt man eine etwas aufwendigere Logik, die die Indexpulse auswertet und Ready meldet, wenn diese in einer Geschwindgkeit folgen, die man bei Nenndrehzahl erwarten muß. Wichtig ist, daß jedes Laufwerk ein unabhängiges Ready meldet, es also mit dem Select verknüpft wird.

Qual der Wahl

Damit aber noch nicht genug. Je moderner die Laufwerke, desto flexibler sind sie meistens, was die Belegung dieser 'neuralgischen Pins' betrifft: Diverse Steckbrücken (Jumper) auf den Laufwerken ermöglichen – in Grenzen – die Anpassung von Laufwerken unterschiedlicher Hersteller bezüglich eben dieser Signale.

Da bei der Bezeichnung dieser Steckbrücken allerdings absolut keine Eintracht herrscht, nicht mal unbedingt bei verschiedenen Laufwerken des gleichen Herstellers, kommt man hier nicht mehr ohne Laufwerkshandbuch weiter. Zwar gibt es inzwischen bei einigen Herstellern gleiche Jumper-Bezeichnungen (HL, HM oder ähnliches), aber man kann nie sicher sein, in welchen Kombinationen mit anderen Jumpern sie was bewirken. Vor allem ist nie ersichtlich, welche Kombinationen verboten sind, weil sie unter Umständen zu Beschädigungen führen.

Eine dieser Optionen ist besonders hervorzuheben: Die Funktion 'Head Load' sollte man stets mit 'Motor on' verknüpfen, so daß der Kopf beim Einschalten des Motors immer geladen wird. IBMs PCs zum Beispiel verlangen diese Einstellung definitiv.

Kühlen Kopf bewahren

Moderne 3.5-Zoll-Laufwerke zum Beispiel bieten sehr kurze Kopf-Ladezeiten beziehungsweise Beruhigungszeiten. Neuere Rechner, etwa Atari oder Amiga, sind dann im allgemeinen vollständig in ihrer Controller-Software auf diese Zeiten abgestimmt. Probleme gibt es, wenn man an solche Rechner zu Konvertierungszwecken ältere 5.25-Zoll-Drives mit längeren Kopf-Ladezeiten anschließt. Da das Ready-Signal keine Aussage über den Lade-Zustand des Kopfes macht, kann also ein noch nicht eingeschwungener Kopf zum Schreiben oder Lesen veranlaßt werden – was typischerweise zu Fehlern führt.

Außerdem trennt man damit bei den meisten Laufwerken die Leitung 'Head Load' vom Bus ab. Das kann notwendig sein, wenn möglicherweise ein anderes Laufwerk im System ist, das über diese Leitung zum Beispiel zwischen normaler und doppelter Schreibdichte umgeschaltet wird. Zum andern erspart es vielfach das 'Kopfgeklapper', das einige Floppy-Controller beim Positionieren veranstalten. Auch zum Beispiel Assembler mit intensiven Diskettenzugriffen (meist unter CP/M) vertrödeln oft so viel Zeit zwischen einzelnen Disk-Zugriffen, daß der Controller schon wieder den Kopf abfallen läßt.

Manchmal kann es sogar hilfreich sein, das floppy-interne Head-Load-Signal per Lötkolben fest an Motor on zu ketten: Zum Beispiel ließ sich ein 8-Zoll-Drive (Siemens) trotz zig Optionen nicht an einem Controller mit µPD 765 zum Steppen überreden. Der 765 (wie auch einige Controller von Western Digital) läßt – Disketten schonend – nämlich auch ohne 'Head Load' den Kopf positionieren. Das bewußte Laufwerk steppt aber partout nicht ohne aktives Head-Load-Signal.

Und da wir schon mal dabei sind: Dieses Laufwerk gibt sich auch nicht mit den kurzen Select-Impulsen zufrieden, die der 765 an die Drives eins bis drei ausgibt. Die Sache funktioniert ohne Zusatzhardware nur, wenn das Laufwerk als viertes im Bunde angesprochen wird, weil der Controller-Chip hier nämlich länger selektiert.

Die Lampe des Laufwerks koppelt man am besten mit dem Signal 'Drive Select', denn dann sieht man sicher, welches Laufwerk gerade angesprochen wird. Auf andere Einstellmöglichkeiten kommen wir bei speziellen Problemen noch zurück.

Breite Spuren

Manche 80-Spur-Laufwerke lassen eine Umstellung für 40-Spur-Betrieb zu. In c't wurde auch eine kleine Schaltung veröffentlicht, mit der man jedes 80-Spur-Laufwerk zu Doppel-Stepping überreden kann (c't 5/86, S. 46, MSDOS-Disketten auf dem Atari ST). In beiden Fällen wird aber nur die Positionierung des Kopfes an die 40-Spur-Diskette angepaßt, die Spurbreite bleibt erhalten.

Das kann unter folgenden Umständen zu Problemen führen: wenn eine Diskette, die in einem echten 40-Spur-Laufwerk formatiert und beschrieben wurde, auf einem 'umgebauten' 80-Spur-Drive beschrieben wurde und diese Diskette später wieder auf einem echten 40-Spur-Laufwerk weiterverwendet werden soll.

Dabei werden nämlich in die 'breiten' 40er Spuren der Diskette schmalere vom 80-Spur-Laufwerk eingebettet, und es können sehr wohl Reste der ursprünglichen 'Breitspur' übrigbleiben. Wird diese Diskette jetzt wieder dem 40-Spur-Laufwerk angeboten, so kann dieses möglicherweise das 'Gemisch' nicht mehr lesen. Resultat: Die Diskette wird als physikalisch defekt gemeldet, ist also auch mit Tricks nicht mehr auf dem 40-Spur-Drive lesbar (wohl aber noch auf dem 80er).

Normalerweise gibt es aber keine Probleme, wenn eine 'jungfräuliche', also unformatierte Diskette auf dem 80-Spur-Laufwerk formatiert wird, so daß keine 'Breitspurreste' vorhanden sind.

Es gibt allerdings noch etwas zu beachten, wenn man 80-Spur-Drives doppelt so große Schritte nehmen läßt, als sie üblicherweise sollen: Sie brauchen auch die doppelte Zeit dafür! Wenn

Ihr Laufwerk also die ersten Spuren im allgemeinen recht zuverlässig lesen kann, danach aber sporadisch Fehler auftauchen, so könnte die Steprate, also die verfügbare Zeit, um die nächste Spur aufzusuchen, zu knapp sein. Die Steprate läßt sich üblicherweise per Software (Betriebssystem) über den Controller-Chip einstellen (Rechnerhandbuch!).

Higher and Higher

Etwa 1983 kamen die ersten 5,25-Zoll-Laufwerke mit 80 Spuren und – für damalige Zeiten – sehr hoher Schreibdichte (High Density) auf den Markt. Ihre Kapazität ließ sich bis etwa 1,4 MByte hochtreiben, abhängig von der Formatierung. Zur Erzielung der hohen Schreibdichte wurden ein paar Anleihen bei den 8-Zoll-Laufwerken aufgenommen. So sind diese Drives zum Beispiel aus Controller-Sicht, also steuerungstechnisch, kompatibel, obwohl sie den 34poligen Anschlußstecker der normalen 5,25-Zoll-Drives aufweisen.

Die wichtigsten Unterschiede zu normalen 5,25-Zoll-Drives liegen in der Datenübertragungsrate und der Motordrehzahl. Die herkömmlichen 5,25-Zoll-Laufwerke arbeiten bei einfacher Schreibdichte (single Density, FM-Aufzeichnung) mit 125 KBit/s, ein Format, das nur bei ganz wenigen, restlos veralteten Rechnern zu finden ist, denn auf einer einseitigen 40-Spur-Scheibe lassen sich keine 100 KByte unterbringen. Die heute fast immer anzutreffende doppelte Schreibdichte (double Density, MFM-Verfahren) bietet bei doppelter Transferrate (250 KBit/s) auch die doppelte Speicherkapazität. Die Diskette macht dabei 300 Umdrehungen pro Minute.

8-Zoll-Laufwerke hingegen arbeiten seit jeher schon bei einfacher Schreibdichte mit 250 KBit/s Übertragungsrate und mit 500 KBit/s bei doppelter Dichte. Längere Spuren (größerer Durchmesser), eine daraus und aus der Motordrehzahl von 360 U/min resultierende etwa verdoppelte Geschwindigkeit zwischen Schreib-/Lesekopf und Disk stellen dabei etwa die gleichen Anforderungen an das Diskettenmaterial wie bei den einfachen 5,25-Zoll-Drives.

Die 5,25-Zoll-Drives mit High-Density-Modus arbeiten ebenfalls mit 360 U/min und einer Transferrate von 500 KBit/s (bei double Density). Daß diese damit de facto vorgenommene 'Verkleinerung' der 8-Zoll-Scheibe auf 5,25 Zoll überhaupt möglich war, lag an den Fortschritten bei der Datenträgerherstellung.

Diese ließ sich der anfangs einzige Anbieter (Maxell) auch fürstlich bezahlen, und normale 80-Spur-Disketten waren auf diesen Laufwerken nicht einsetzbar. Aber umgekehrt liefen übrigens auch diese ersten High-Density-Disketten nicht in herkömmlichen Laufwerken (andere magnetische Daten).

Damit fristeten diese Laufwerke zunächst ein sehr tristes Schattendasein. Sie konnten quasi nur in Rechner eingebaut werden, die in der Lage waren, Laufwerke mit 8-Zoll-Double-Density zu bedienen. Den Einbau konnten nur technisch versierte Leute vornehmen, da hierzu ja auch eine Konvertierung des 50poligen 8-Zoll-Bus an den 34poligen für 5,25-Zoll-Drives vorgenommen werden mußte; und die teuren Disketten waren zu nichts in der Welt kompatibel.

High and low

Etwa ein Jahr später erkannten die Hersteller die Nachteile solch einer 'Insellösung'. Sie schufen eine neue Generation von High-Density-Laufwerken, die dem Anwender auch Zugang zur alten 5,25-Zoll-Welt gewährt. Diese neuen HD-Laufwerke avancierten zu sogenannten Multifunktionslaufwerken (abgekürzt MF), die zwischen 'normalem' 80-Spur-Betrieb (300 U/min, 250 KBit/s, herkömmliche Magnetisierung) und High-Density-Betrieb (360 U/min, 500 KBit/s, hohe Magnetisierung) umschaltbar sind.

Die Umschaltung zwischen hoher und normaler Schreibdichte ist nicht einheitlich gelöst. Einige Laufwerke haben einen Jumper, mit dem man eine der beiden Betriebsarten fest wählen kann. Andere lassen sich nur über eine Bus-Leitung umschalten, wobei man aber die Polarität des Umschaltsignals wählen kann.

Auch bezüglich der Umschaltung der Motordrehzahl findet man Unterschiede. So gibt es Laufwerke, etwa für den IBM AT, die grundsätzlich nur mit 360 U/min arbeiten. Um damit herkömmliche Disketten bearbeiten zu können, braucht man – wie der AT – einen Floppy-Controller, der mit der recht ungewöhnlichen Datentransferrate von 300 KBit/s aufwarten kann. Zum Thema AT kommen wir aber noch ausführlich.

Andere Laufwerke hingegen schalten zusammen mit der Schreibdichte auch stets die Drehzahl um. Noch andere, und neuerdings wohl die meisten, überlassen dem Anwender per Jumper die Wahl, ob er Betrieb in beiden Schreibdichten nur mit 360 U/min (für ATs also) oder mit Drehzahlumschaltung möchte.

Es ist wohl einzusehen, daß die Jumper-Stellungen sehr sorgfältig anhand der Dokumentation den jeweiligen Bedürfnissen anzupassen sind. Eine mit 360 U/min, einfacher Dichte und 250 KBit/s beschriebene 80-Spur-Diskette ist von keinem Nicht-Multifunktionslaufwerk lesbar, und umgekehrt sind 'regulär' beschriebene 80-Spur-Scheiben in einem schneller rotierenden MF-Laufwerk nur mit einer Datenrate von 300 KBit/s zu verarbeiten. Die meisten fabrikneuen MF-Laufwerke werden übrigens heute standardmäßig mit einer Jumperung für den IBM AT ausgeliefert.

Generationsprobleme

Wie erwähnt war die erste Generation HD-Laufwerke und -Disketten völlig unverträglich mit den herkömmlichen. Mit Aufkommen der MF-Drives ist eine Annäherung technischer Daten von HD-Laufwerken und -Disketten an die alte 5,25-Zoll-Welt zu beobachten. Es gibt also HD-Scheiben, die es in MF- und in Standardlaufwerken tun, aber es klappt nicht unbedingt bei jeder Disk-Sorte mit jedem Laufwerks-Fabrikat. Wichtig ist, daß sich solche Unverträglichkeiten nicht von Leseproblemen unterscheiden lassen, die etwa durch falsche Formate (Disk-Parameter) oder fehlerhaften Anschluß des Laufwerkes hervorgerufen werden.

Wenn Ihnen also ein Bekannter eine auf seinem Rechner Laufwerk erprobte High-Density-Scheibe zum Testen mitgibt, diese aber in Ihrem Laufwerk nur Schreib-/Lesefehler produziert, so besorgen Sie sich sicherheitshalber zunächst einmal eine HD-Scheibe eines Fabrikats, das der Laufwerkshersteller empfiehlt, ehe Sie wochenlang Bus-Signale oszilloskopieren.

Ein weiterer 'kleiner Unterschied' zwischen Nur-HD-Drives und Multifunktionslaufwerken liegt darin, daß erstere eine sogenannte Schreibvorkompensation benötigen (unter anderem auch, damit sie gleichzeitig mit 8-Zoll-Drives an einem Controller laufen können). Diese Write Precompensation wird vom Controller ab Spur 43 (über das Signal 'Track greater 43') vorgenommen. Die inneren (kürzeren) Spuren ließen sich dadurch später besser lesen. Das Problem ist: Reine HD-Laufwerke müssen mit Vorkompensation betrieben werden. MF-Laufwerke hingegen verkraften diese in der Regel nicht!

Wenn das Formatierprogramm gleich ein Verify (Kontrolle der Formatier-Information) durchführt, merkt man schon beim Formatieren, wie es das eigene Laufwerk ab Spur 43 lieber hat. Hat man einen 'dummen' Formatter, erlebt man möglicherweise erst bei mehr als zur Hälfte mit Daten gefüllter Scheibe 'merkwürdige' Schreib/-Lesefehler mit herben Datenverlusten.

Extrawurst – Industriestandard PC

Wenn Sie glauben, daß der 'Industriestandard PC' jetzt eitel Kompatibilität beschert, müssen wir Sie bitter enttäuschen: Jetzt geht's nämlich erst richtig los.

PCs und ATs beziehungsweise die Schar der damit Kompatiblen bieten zwei besondere Probleme. Zum einen haben einige (!) dieser Geräte ein teilweise manipuliertes Anschlußkabel. Man erkennt es meistens schon daran, daß das Flachbandkabel nicht völlig unversehrt, also 'in einem Stück' von Stecker zu Stecker geführt wird, sondern einige Adern 'umgelegt' wurden.

Aus nicht ganz überzeugenden Gründen bewirkt man damit, daß alle angeschlossenen Laufwerke (übrigens auch die Festplatten) stets über die Leitung 'Drive Select 1' aktiviert werden, also deren Jumper für die physikalische Laufwerksauswahl auf 'DS1' (zweites Laufwerk, da meist von 0 gezählt wird) gesteckt werden müssen.

Weiterhin hat man festgelegt, daß über das interne Kabel nur maximal zwei Laufwerke bedient werden. Die beiden verbliebenen Select-Leitungen werden – höchst ungewöhnlich – als separates Motor-on-Signal für jedes der beiden Laufwerke ausgeführt. Im Gegensatz zu anderen Rechnern, die mit einer Motor-on-Leitung also die Motoren aller angeschlossenen Laufwerke gemeinsam einschalten, bedienen PCs und Abkömmlinge ihre Drives einzeln.

Allerdings läßt der PC nach außen (über eine Cannon-Steckverbindung) den Anschluß weiterer zwei Laufwerke zu, liefert also Select 3 und 4 nebst den Motor-on-Signalen. In einfachen PCs und XTs findet man gewöhnlich nur Controller, die Standardlaufwerke mit 250 KBit/s Transferrate bedienen können. Controller, die auch Multifunktionslaufwerke im High-Density-Modus unterstützen und 500 KBit/s Datenrate erlauben, sind nur als Sonderzubehör erhältlich.

Das zweite Problem sind die meisten ATs, die mit Multifunktionslaufwerken und einer Festplatte ausgerüstet sind. In diesen findet sich fast immer ein sogenannter Kombi-Controller für zwei Harddisks und zwei umschaltbare MF-Drives.

Alleskönner AT

Die meisten Exemplare dieser Gattung (zum Beispiel das Original oder der Kaypro 286i) schalten – wie bereits angedeutet – die Geschwindigkeit der MF-Laufwerke bei Betrieb mit Normaldisketten (40-Spur-Betrieb, 360 KByte) nicht auf 300 U/min zurück, sondern lassen diese weiterhin mit 360 U/min rotieren.

Damit man kompatibel zu herkömmlichen 40-Spur-Laufwerken schreiben und lesen kann, muß dazu nur die Datentransferrate nicht auf die üblichen 250 KBit/s, sondern auf 300 KBit/s heruntergeschaltet werden, was der Controller auch tut. Dieser durchaus cleveren Maßnahme verdankt der AT einen überdurchschnittlich schnellen Zugriff auf diese Disketten.

Der 34polige Floppy-Bus

Die derzeit am weitesten verbreiteten Floppy-Laufwerke sind (noch) die 5,25-Zoll-Laufwerke mit ihrem 34poligen Bus. Erfreulicherweise verfügen auch die neueren 3,5-Laufwerke, die die 5,25-Zöller langsam ablösen, und sogar die 3,0-Zoll-Laufwerke (die sich aber wohl nicht auf breiter Ebene durchsetzen werden) über den gleichen Bus mit 34 Leitungen.

Bei diesem Bus führen alle Leitungen Masse (0 V), die mit Anschluß-Pins mit ungeraden Nummern verbunden sind. Das heißt, alle benachbarten Signalleitungen (an Pins mit geraden Nummern) werden bei Verwendung der üblichen Flachbandkabel durch Masseleitungen voneinander getrennt. Die Verbindung zum Floppy-Laufwerk erfolgt bei 5,25-und 3,0-Zöllern mit einem Platinen-Direktstecker, bei 3,5-Zoll-Drives auch vielfach über einen Pfostenstecker (weiblich). Der aktive Zustand sämtlicher Leitungen ist 'low', also etwa 0 Volt.

Pin 2

Dieser Anschluß ist bei verschiedenen Laufwerken unterschiedlich belegt:

1. Head Load
Laufwerkseingang. Damit wird der Schreib-/Lesekopf des Laufwerks geladen (an die Diskette gedrückt und vormagnetisiert). Nach diesem Vorgang sind Wartezeiten ähnlich wie nach einem Step-Impuls einzuhalten (siehe Pin 20).

2. In Use
Laufwerkseingang. Damit kann dem Laufwerk mitgeteilt werden, daß weitere Zugriffe erfolgen werden (oft kann man damit aber auch nur die Lampe an der Frontblende schalten).

3. High-/Normal-Density-Umschaltung
Laufwerkseingang. Mit diesem Signal wird bei Multifunktionslaufwerken die Betriebsart zwischen normaler Schreibdichte (300 U/min, 250 KBit/s Datenrate) und hoher Schreibdichte (360 U/min, 500 KBit/s) umgeschaltet. Die Polarität dieses Signals ist im allgemeinen auf dem Laufwerk per Jumper einstellbar.

4. Disk Change
Laufwerksausgang. Er signalisiert bei selektiertem Laufwerk, daß die Diskette gewechselt wurde. Meistens wird dafür ein Schalter am Diskettenschacht herangezogen beziehungsweise das 'Index'- oder 'Write-Protect'-Signal wird ausgewertet.

Pin 4

Auch dieser Anschluß ist bei verschiedenen Laufwerken unterschiedlich belegt.

1. Head Load (siehe Pin 2, Punkt 1)
2. In Use (siehe Pin 2, Punkt 2)
3. Disk Change (siehe Pin 2, Punkt 4)

Pin 6

Auch dieser Anschluß hat bei verschiedenen Laufwerken unterschiedliche Belegung.

1. Drive Select 3 (siehe Pin 10, 12, 14)
2. Ready (siehe Pin 34)

Pin 8: Index/Sector

Dieser Laufwerksausgang geht auf Low-Pegel, wenn das Indexloch der Diskette (bei 3,5-Zoll-Drives ist es eine Kerbe im Laufwerks-Teller) den Sensor passiert. Bei Standardlaufwerken (300 U/min) erscheinen daher fünf Impulse pro Sekunde, bei Laufwerken mit hoher Schreibdichte (360 U/min) sechs. Der Impuls kennzeichnet den Beginn einer Spur bei Soft-Sektorierung und dient zur Synchronisierung des Floppy-Controllers. (Nur sehr wenige ältere Computer verwenden noch Disketten mit Hard-Sektorierung, bei der für jeden Sektorbeginn ein Loch vorhanden ist.)

Diese Impulse können auch zur Drehzahlbestimmung vom Rechner ausgewertet werden. Die meisten Laufwerke generieren daraus intern das Ready-Signal, das anzeigt, daß der Motor die zum Betrieb nötige Drehzahl erreicht hat. Dieses Signal wird nur aktiv, wenn die dem Laufwerk zugeordnete Select-Leitung aktiviert wird.

Pin 10: Drive Select 0
Pin 12: Drive Select 1
Pin 14: Drive Select 2

Laufwerkseingang. Das folgende gilt auch für die Leitung 'Drive Select 3', falls diese vorhanden ist. Laufwerksintern kann über drei (oder vier) Steckbrücken entschieden werden, über welche Select-Leitung das Laufwerk angesprochen und aktiviert werden soll. Der Controller darf nur dann mehr als eine Select-Leitung aktivieren, wenn sichergestellt ist, daß dabei nicht zwei Laufwerke gleichzeitig über dieselben Ausgangsleitungen antworten (Bus-Crash). Ebenso dürfen auf einem Laufwerk nur dann mehrere Select-Jumper gesteckt sein, wenn ausgeschlossen werden kann, daß dadurch mehr als ein Laufwerk aktiviert wird.

Einige Laufwerke können auch in einen All-Select-Modus versetzt werden (wirkt, als wären alle Brücken gesteckt). Das kann sinnvoll sein, wenn nur ein Laufwerk angeschlossen ist. Bei einigen älteren Laufwerken (vorwiegend allerdings achtzölligen) ist auch eine Multiplex-Betriebsart möglich. Dabei aktiviert der Controller mehrere Select-Leitungen, die dann aber als binäre Adresse ausgewertet werden. Dadurch lassen sich über drei Leitungen acht, über vier 16 Laufwerke auswählen.

Pin 16: Motor on

Laufwerkseingang, über den der Motor des Laufwerks eingeschaltet wird. Hier kann bei vielen Laufwerken durch Jumper bestimmt werden, ob der Motor unabhängig von den Select-Leitungen eingeschaltet wird oder nicht. Bei einigen Disk-Controllern sollte man die Stellung für unabhängiges Einschalten des Laufwerks wählen, da die Select-Signale nicht kontinuierlich anstehen (etwa beim µPD 765).

Außerdem kann es oft vorteilhaft sein, die Option einzustellen, bei der zusammen mit 'Motor on' der Kopf des Laufwerks geladen wird (siehe Text).

Pin 18: Direction Select

Über diesen Eingang des Floppy-Laufwerkes bestimmt der Controller, in welche Richtung der Schreib-/Lesekopf des Laufwerkes bei einem Spurwechsel bewegt wird. Dabei bewirkt Low-Pegel die Bewegung zur Mitte der Diskette (höhere Spurnummern), High-Pegel die Bewegung nach außen (zur Spur 0). Dieses Signal wird nur dann abgefragt, wenn gleichzeitig ein Step-Impuls vorliegt.

Pin 20: Step

Laufwerkseingang. Jeder Impuls (Low-Pegel) an diesem Anschluß bewirkt bei selektiertem Laufwerk eine Bewegung des Schreib-/Lesekopfes in die durch das Signal 'Direction Select' vorgegebene Richtung. Die Bewegungsgeschwindigkeit wird durch die Wiederholrate (Step Rate) des Step-Impulses bestimmt. Die Zeiten liegen bei modernen Laufwerken zwischen drei und sechs Millisekunden, bei älteren und besonders bei Billiglaufwerken können sie bis zu 30 Millisekunden betragen. Die aktuellen Zeiten sind dem jeweiligen Datenblatt des Laufwerkes zu entnehmen und gegebenenfalls in der Steuersoftware des Floppy-Controllers (meist im Betriebssystem) an die Laufwerke anzupassen.

Durch die Abhängigkeit vom Select-Signal können die Step-Impulse für mehrere Laufwerke verschachtelt werden, so daß diese gleichzeitig ihren Kopf positionieren können. Dies wird zwar von manchen Disk-Controllern unterstützt, doch die meisten preiswerten Betriebssysteme nutzen solche Spezialitäten nicht.

Nach einer Positionierung des Kopfes sind beim Lesen und Schreiben Wartezeiten einzuhalten, da der Kopf sich erst in seiner neuen Position stabilisieren muß. Bei Lesevorgängen ist dies nicht so wichtig, da hier Lesefehler durch den Disk-Controller oder das Betriebssystem des Rechners (erneutes Lesen) ausgeglichen werden, aber beim Schreiben ist Vorsicht geboten, da durch Positionsfehler der beschriebene Sektor unleserlich werden kann. Die Wartezeiten sind dem Datenblatt des Laufwerkes zu entnehmen (notfalls kann man mit 15 bis 25 Millisekunden experimentieren).

Pin 22: Write Data

Laufwerkseingang. Über diese Leitung werden die zu speichernden Daten seriell ans Laufwerk übertragen. Voraussetzung ist, daß das Signal 'Write Gate' aktiv und das Laufwerk 'ready' ist.

Es gibt unterschiedliche Übertragungsverfahren. Allen gemeinsam ist, daß immer ein kompletter Sektor übertragen werden muß. Die Datentransferrate hingegen variiert je nach Verfahren und Taktrate zwischen 125 und 500 KBit/s (siehe Text).

Pin 24: Write Gate

Laufwerkseingang. Mit diesem Signal wird das zum Schreiben der Daten nötige Löschen der alten Daten eingeschaltet und die Schreibdaten (Write Data) werden freigegeben. Voraussetzung ist wieder, daß das Laufwerk selektiert und 'ready' ist.

Pin 26: Track 0

Dieser Laufwerksausgang signalisiert bei selektiertem Drive, daß der Kopf auf Spur 0 positioniert ist.

Pin 28: Write Protect

Laufwerksausgang. Signalisiert bei selektiertem Drive, daß der Schreibschutz der Diskette aktiviert ist und das Laufwerk keine Daten auf Diskette schreiben (und damit ändern) kann.

Pin 30: Read Data

Laufwerksausgang. Wenn das Drive selektiert und 'ready' ist, 'Write Gate' hingegen inaktiv, werden die Daten von der Diskette gelesen und zum Floppy-Controller im Rechner übertragen.

Pin 32: Side 1 Select

Laufwerkseingang. Über ihn wird bei zweiseitigen Laufwerken (Schreib-/Lesekopf für jede Diskettenseite) bestimmt, welcher Kopf benutzt wird. Mit High-Pegel wird die Seite 0 (die auch von einseitigen Laufwerken benutzt wird), mit Low-Pegel die Seite 1 angesprochen.

Nach einem Umschalten des Kopfes sollte mit einem Schreibvorgang einige Zeit gewartet werden, damit die Schreiblogik des Laufwerkes Zeit hat, auf den anderen Kopf umzuschalten (im Datenblatt des Laufwerks nachsehen oder 1 bis 2 Millisekunden warten). Wird diese Zeit nicht eingehalten, so kann der Schreibvorgang einen Sektor zerstören.

Pin 34:

Auch dieser Pin wird im allgemeinen unterschiedlich belegt:

1.Ready
Laufwerksausgang. Diese Leitung übermittelt bei selektiertem Laufwerk die Einsatzbereitschaft: eine Diskette ist eingelegt, der Motor läuft auf Nenndrehzahl. Das Laufwerk ist also (bis auf 'Headload') bereit, eine Funktion wie Lesen, Schreiben oder Positionieren des Kopfes auszuführen. Dabei ist bei manchen Laufwerken das Ready-Signal für die Positionierung des Kopfes nicht unbedingt erforderlich, und es gibt Controller, die den Kopf positionieren, ohne daß er geladen sein muß. Diese Leitung ist bei Laufwerken, die Anschluß 6 als Ready-Signal benutzen, meistens frei.

Erzeugt ein Laufwerk kein Ready-Signal, muß man es für einige Controller künstlich erzeugen (Zusatzhardware, siehe Text). Ist es dem Controller 'egal', ob er ein Ready-Signal bekommt oder nicht, muß die Steuersoftware nach dem Ansprechen des Laufwerkes ('Motor on' und 'Select' aktiv) die Hochlaufzeit des Motors und die Kopfladezeit durch geeignete Zeitschleifen abwarten (Laufwerksdaten dem Handbuch entnehmen).

2. In Use
(siehe Pin 2, Punkt 2)

3. Disk Change
(siehe Pin 2, Punkt 4)

Das ist aber noch nicht alles. Multifunktionslaufwerke sind 'von Natur aus' 80-Spur-Laufwerke, und da es hier vielfach Mißverständnisse gibt, sei es besonders betont: auch die im AT. Daß der AT damit in der Betriebsart 'einfache Schreibdichte' grundsätzlich ein 40-Spur-Format unterstützt, liegt daran, daß sein ROM-BIOS dafür sorgt, daß stets eine Spur übersprungen, also ein sogenanntes 'Double Stepping' durchgeführt wird.

Manchen Besitzer eines AT mit nur einem Laufwerk wird der leere Laufwerksschacht schon bewogen haben, hier ein 'übriggebliebenes' oder günstig erworbenes 40-Spur-Laufwerk unterzubringen. Nach dem zunächst Gesagten könnte man ja annehmen, daß Double-Stepping (wem nützt ein 20-Spur-Laufwerk?) und 300 KBit/s Datentransferrate (verkraftet kein einfaches 40-Spur-Drive) alle Hoffnungen zunichte machen.

Stepping Stones

Und genau diese Effekte erlebt man, wenn man ein Laufwerk einsetzt, das auf Pin 34 ein Ready- oder Disk-Change-Signal liefert. Alle Probiererei mit SETUP oder Treibern erlöst einen nicht: Solange der AT hier Signale empfängt, behandelt er das neue Drive als MF-Laufwerk. Erst wenn diese Leitung gekappt wurde, enthüllt der AT neue Controller-Fähigkeiten: kein Double-Stepping mehr, 250 KBit/s Datenrate.

Bei PCDOS 3.1 läßt sich das neue Laufwerk formatieren, indem man nicht mehr die Option '/4' angibt. Ab DOS 3.2 wird's komplizierter. Hier stellt sich FORMAT.COM ohne '/4' auch bei dem einfachen Laufwerk auf das High-Density-Format ein, und mit der Option auf 300 KBit/s und Double-Stepping. Abhilfe: Man muß den mitgelieferten Treiber DRIVER.SYS mit dem Aufruf

```
DEVICE=DRIVER.SYS
                D:1/F:0
```

in CONFIG.SYS eintragen. D:1 bezeichnet das zweite physikalische Laufwerk, und F:0, daß es sich um ein echtes 360-KB-Laufwerk handelt. Achten Sie unbedingt auf das Leerzeichen vor dem ersten Schrägstrich, sonst gibt es eine Fehlermeldung. Der Treiber gibt als Meldung aus, unter welchem Laufwerksbuchstaben er zu finden ist.

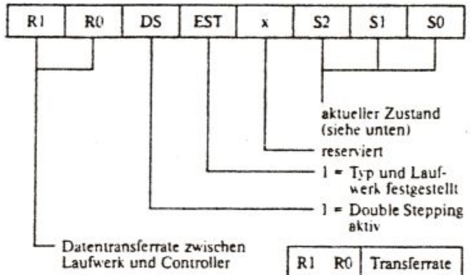
Jetzt wird FORMAT.COM mit Angabe des entsprechenden Laufwerksbuchstaben aufgerufen, aber ohne Zuhilfenahme weiterer (verlockender) Parameter, die das DOS-Handbuch bereithält. Schreiben und lesen kann man anschließend wieder über Laufwerk B:.

80 Spuren / 720 KB

Und da wir nun mal dabei sind: Natürlich kann man auf diese Art (aber nur ab DOS-Version 3.2!) auch einfache 80-Spur-Laufwerke beziehungsweise 3,5-Zoll-Drives anschließen. Zu letzteren kommen wir noch, aber hier sei noch mal darauf hingewiesen, daß auch hier wieder Pin 34 (Ready oder Disk Change) stillgelegt werden muß! Das Formatieren klappt ebenfalls wieder nur über Treiber, die bei Standard-80-Spur-Drives folgenden Aufruf enthalten müssen (aus Platzgründen schreiben wir in zwei Zeilen):

```
DEVICE=DRIVER.SYS
                D:1 F:2 T:80
```

Auch bei diesem Format kann man nach dem Formatieren weiter unter dem Buchstaben B: arbeiten, da sich die neuen DOS-Versionen die vollständige Formatinformation beim ersten Zugriff aus dem Bootsektor holen und den Rechner korrekt darauf einstellen.



R1	R0	Transferrate
0	0	500 KBit/s
0	1	300 KBit/s
1	0	250 KBit/s
1	1	reserviert

S2	S1	S0	aktueller Zustand
0	0	0	360-KB-Disk in 40-Spur-Laufwerk, Status ungeprüft
0	0	1	360-KB-Disk in MF-Laufwerk, Status ungeprüft
0	1	0	1,2-MB-Disk in MF-Laufwerk, Status ungeprüft
0	1	1	360-KB-Disk in 40-Spur-Laufwerk, Status geprüft
1	0	0	360-KB-Disk in MF-Laufwerk, Status geprüft
1	0	1	1,2-MB-Disk in AT-Laufwerk, Status geprüft

Tabelle 1. Die Bedeutung der einzelnen Bits in den Disk-Status-Zellen des AT. Die Speicherstellen liegen im RAM an Adresse 0040:0091 (1. Laufwerk) und 0040:0091 (2. Laufwerk). 'Status geprüft' (established) heißt, daß dieser Status durch einen Zugriff verifiziert wurde.

Sollte der AT sich gelegentlich vertun, so kann ein Blick mit dem Debugger in seine Disk-Status-Zellen und eine eventuell erforderliche Korrektur Abhilfe schaffen. Ab Adresse 90h bei Datensegment 40, also 40:90, liegen für die beiden physikalischen Laufwerke zwei Speicherzellen, die Auskunft über die eingestellte Laufwerksbetriebsart geben.

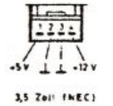
Wenn man diese manipuliert, so kann man den AT zu allem überreden, was Tabelle 1 und das jeweilige Laufwerk hergeben – und damit natürlich auch das 80-Spur-Format mit 720 KB Speicherplatz mittels Multifunktionslaufwerk erzwingen. Letzteres allerdings nur bis zum nächsten Öffnen der Laufwerksklappe (Disk Change); denn dann datiert der AT diese Zellen wieder mit den ihm aus dem ROM-BIOS geläufigen Werten auf.

Nach dem Kaltstart steht bei 40:90 (und 91, wenn zwei Laufwerke im System sind) üblicherweise 61h. Nach einem Zugriff auf 360-KB-Scheiben im Multifunktionslaufwerk wird der Wert auf 74h geändert, also sind Double-Stepping und 300 KBit/s aktiv. Nach dem Zugriff auf eine 1,2-MB-Floppy findet man folgerichtig 15h. Nachdem eine Diskette in einem 360- oder 720-KB-Laufwerk gelesen wurde, wird dieser Wert auf 93h gesetzt.

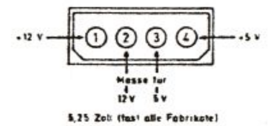
Interessant dabei ist, daß auch der Startwert 61h eine Rate von 300 KBit pro Sekunde einstellt, aber dem Controller scheint es nichts auszumachen, wenn das Laufwerk den Bootsektor mit 250 KBit/s anliefert. Das Double-Stepping stört beim Lesen von Spur 0 ohnehin nicht.

Verändert man aber diesen Wert versehentlich auf 15h, etwa durch Aufruf von FORMAT ohne Parameter auf Drive B:, dann ist es aus. Das ROM-BIOS des AT probiert in Multifunktionslaufwerken nach Erkennen eines Disk-Wechsels zunächst aus, ob eine 360-KB-Disk einliegt, dann wechselt es zu High Density, aber nicht mehr zurück. Da ein Laufwerksstatus 15h durch das fehlende Disk-Change-Signal von einem einfachen Laufwerk nicht aufgehoben werden kann, bleibt nur die Korrektur im RAM beziehungsweise ein Warmstart.

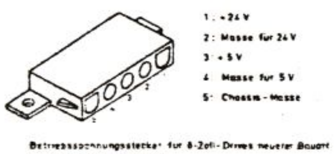
Um nun mit MF-Laufwerken 80 Spuren schreiben und lesen zu können, braucht man nur die bewußte Diskette einzulegen und den Status auf 54h zu ändern, denn man sollte das mit 360 U/min laufende Drive natürlich mit 300 KBit/s bedie-



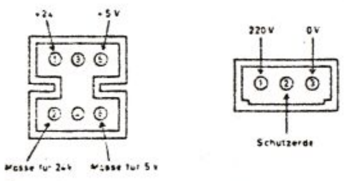
3,5 Zoll (NEC)



5,25 Zoll (fast alle Fabrikate)
3,0 Zoll (z.B. Hitachi)



Betriebsspannungsstecker für 8-Zoll-Drives neuerer Bauart (z.B. Shugart SA 810/850)



Ältere 8-Zoll-Laufwerke betreiben den Laufwerksmotor mit Netzspannung (220 V). So ist zum Beispiel das Siemens FDD 100-8D mit dieser Art der Stecker ausgerüstet.

Vielfalt statt Standard, auch bei den Steckverbindungen für die Betriebsspannungszuführung. Man kann die Anschlußstecker für 5,25- und 3,5-Zoll-Laufwerke zwar nicht an die jeweiligen anderen Drives anschließen, aber die vertauschte Zuordnung von Pin-Nummern zu Betriebsspannungen kann tödlich in die Irre führen. Die dargestellten Steckverbindungen sind die Versionen am Floppy-Laufwerk, sozusagen 'aus der Sicht des Steckers'.

Bus-Pin	BASF 6106/08	Tandon TM 50–1/2	Shugart SA 400 L	Philips X 3111 bis 3114	BASF 6128/38	NEC 1035	Hitachi HFD 305 SX	Mitsubishi M 4854	TEAC 55 GFV
2	Head Load	Connector Clamp	-	Ready/ Head Load	Head Load	-	-	Two Sided/ Drive Selected	High/ Normal Density
4	-	Spare	-	In Use/ Select 4	In Use (opt.)	Head Load/ In Use	In Use	In Use/ Head Load	In Use/Head Load/Open
6	Ready	Drive Select 3	Drive Select 4	Ready/Motor on	Ready/Select 4	Drive Select 3	Drive Select 3	Drive Select 3	Drive Select 3
8	Index	Index/Sector	Index/Sector	Index	Index	Index	Index	Index	Index
10	Select 1	Drive Select 0	Drive Select 1	Select 1	Select 1	Drive Select 0	Drive Select 0	Drive Select 0	Drive Select 0
12	Select 2	Drive Select 1	Drive Select 2	Select 2	Select 2	Drive Select 1	Drive Select 1	Drive Select 1	Drive Select 1
14	Select 3	Drive Select 2	Drive Select 3	Select 3/In Use	Select 3	Drive Select 2	Drive Select 2	Drive Select 2	Drive Select 2
16	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on	Motor on
18	Direction in	Direction Select	Direction Select	Direction	Dir Sel	Direction Select	Direction Select	Direction Select	Direction Select
20	Step	Step	Step	Step	Step	Step	Step	Step	Step
22	Write Data	Composite Write Data	Write Data	Write Data	Write Data	Write Data	Write Data	Write Data	Write Data
24	Write Gate	Write Enable	Write Gate	Write Gate	Write Gate	Write Gate	Write Gate	Write Gate	Write Gate
26	Track 00	Track 0	Track 00	Track 00	Track 00	Track 0	Track 00	Track 00	Track 00
28	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect	Write Protect
30	Read Data	Composite Read Data	Read Data	Read Data	Read Data	Read Data	Read Data	Read Data	Read Data
32	Side Select	Side Select	-	Head Select	Side Select	Side Select	Side 1 Select	Side One Select	Side One Select
34	In Use/ Change Disk	Connector Clamp	-	In Use/Disk Change/Ready	In Use/Disk Change/Ready	Ready	Ready	Ready	Ready/ Disk Change
Anzahl Steck-brücken	10	8	8	29	27	bis zu 25	4	11	17
Laufwerks-Typen	5,25 Zoll, 40 Spuren, ein-/zweiseitig	5,25 Zoll, 40 Spuren, ein-/zweiseitig	5,25 Zoll, 40 Spuren, einseitig	Vier verschie-dene 5,25-Zoll-Drives, 40 und 80 Spuren, ein- und zweiseitig	5,25 Zoll, zweiseitig, 40/80 Spuren	3,5 Zoll, div. 80-Spur-Drives	3,0 Zoll, 40 Spuren, einseitig	5,25 Zoll, 80 Spuren, zweiseitig, nur High Density	5,25 Zoll, 80 Spuren, zweiseitig High/Normal Density

Tabelle 2. So verschieden können Floppy-Bus-Anschlüsse sein. Hier ein bunter Querschnitt recht weit verbreiteter 5,25- und 3,5-Zoll-Floppy-Laufwerke verschiedensten Typs und Alters. Wir haben auch bewußt die teilweise recht unterschiedlichen Originalschreibweisen für die verschiedenen Signale übernommen.

nen. Auch Formatieren geht, wenn auch nur wieder über den Treiber. Aufruf: wie oben /D:0 (erstes Drive) oder /D:1 (zweites) und /F:2/T:80. Wie man das Ganze so bastelt, daß der Status nicht verlorengeht, beschreiben wir in einem der nächsten Hefte.

Die Kleinen

Nun zu den 3- und 3,5-Zöllern. Der offensichtlichste Unterschied des Anschlusses liegt darin, daß hier eine Pfostenleiste statt eines Stücks Platine zur Kontaktierung benutzt wird. Ein Pfostenstecker statt des üblichen Platinensteckers angepreßt, und das war's. Die Signalbezeichnungen stimmen (mit allen Anmerkungen) mit denen bei 5.25-Zoll-Laufwerken überein.

Lebenswichtig aber ein Unterschied bei den Betriebsspannungssteckern! Bei 5,25- und bei 3,5-Zoll-Laufwerken kommen vierpolige Stecker zum Einsatz. Diese Stecker sind zwar völlig inkompatibel, aber ihre Pins sind bei beiden zur Kennzeichnung der anliegenden Betriebsspannungen jeweils von 1 bis 4 durchnumeriert – allerdings entgegengesetzt (siehe Bild)!

Die wirklichen Probleme beginnen damit, daß die meisten neuen Laufwerke mit stromsparenden und etwas leistungsschwachen CMOS-ICs ausgerüstet sind. Dadurch sind sie nicht mehr in der Lage, die bei den anderen Drives bislang üblichen Abschlußwiderstände (zur Bus-Terminierung) von 150 Ohm zu treiben. Sie brauchen zunächst leichtere Lasten auf dem Disk-Controller (Werte von 1,5 kOhm sind angebracht).

Hat man ein gemischtes System aus 3,5- und 5,25-Zöllern, so müssen auch die Widerstands-Arrays auf letzteren vergrößert werden. Können die meisten 5,25-Zoll-Laufwerke und Controller sogar zwei bis drei Laufwerke mit 150-Ohm-Arrays verkraften (obwohl nach Vorschrift immer nur das letzte am Kabel eines haben soll), so ist das für 3,5-Zöller und übrigens auch für ganz neue 5,25-Zoll-Drives zuviel des Guten. Ein gemeinsames Array von 470 Ohm wird im allgemeinen von alles Drives als vertretbarer Kompromiß hingenommen.

Als nächstes tauchen die Laufwerke mit kodierten Drive-Select-Leitungen auf: Hier lassen sich meist nur zwei Select-Leitungen verwenden, nämlich bei Laufwerkseinstellung 1 und 2.

Bei einer weiteren Sorte von Laufwerken wird man mit einer Motordrehzahl von 600 U/min überrascht! Diese Exoten (Sony) sind aber höchst selten. Sie bieten nicht etwa eine erhöhte Kapazität, sondern nur einen doppelt schnellen Datentransfer, vorausgesetzt, man hat einen darauf eingerichteten Controller (500 KBit/s).

Die neuen High-Density-Drives, die beispielsweise in IBMs neuer Personal-System/2-Familie Dienst tun und 1,44 MByte fassen, begnügen sich wieder mit den altbekannten 360 U/min.

Den krönenden Abschluß bildet die Firma Schneider beziehungsweise Amstrad. Obwohl die 3,0-Zoll-Drives in den CPCs etwa so kompatibel zu 5,25-Zöllern sind wie letztere untereinander (bis auf die gleichfalls schwächeren Treiber), hat man die Anschlußstecker einfach spiegelbildlich belegt. Um an einem CPC an den externen Stecker ein 'Fremdlaufwerk' anzuschließen, ist man gezwungen, sich einen Stecker mit vertauschten Anschlüssen zu löten. (Kleiner Tip für Ungläubige: man kann die Preßsteckverbindungen beliebig aufsetzen, die Spiegelung bleibt!)

Grenzenlose Vielfalt

Die aufgezeigten Probleme machen es wohl überdeutlich: unbekannte Laufwerke ohne detaillierte Unterlagen zum Laufen bringen zu wollen kann einigermaßen spannend sein. Dies gilt besonders für gebrauchte 8-Zoll-Laufwerke, da bei diesen derartig viele Varianten (auch mit verschiedenen Bus-Anschlüssen) im Umlauf sind und bei einigen Typen so viele Jumper-Optionen möglich und nötig sind, daß man sich selbst mit passender 'Landkarte' kaum zurechtfindet.

Aber auch bei 5,25-Zöllern sollte man nicht wahllos zugreifen. Für den Apple II und dessen TTL-Controller umgebaute Laufwerke laufen meist nicht an normalen Floppy-Controllern, umgekehrt geht's fast nie. Während die Aufzeichnungsverfahren mit Standard-Controllern unabhängig von den Daten eine relativ konstante Frequenz aufweisen, auf die die meisten Laufwerke mit Filtern abgestimmt sind, variiert dies beim Apple-Aufzeichnungsverfahren sehr stark. Ohne Umbauanleitung ist man hier meist aufgeschmissen.

Des weiteren gibt es sehr alte 5,25-Zoll-Laufwerke, die nur 35 statt 40 Spuren haben. Der Unterschied ist quasi nicht zu sehen, aber natürlich im Betrieb deutlich zu merken. Haben diese Laufwerke schon einen Spindelantrieb zur Kopfpositionierung, so kann man unter Umständen den Anschlag entfernen und noch ein paar Spuren rausschinden. Wird aber über eine Plastikschnecke positioniert, so hat man Pech gehabt.

Auch 3,5-Zöller mit nur 40 Spuren sind im Umlauf. Diese lassen sich dann zwar gut dort einsetzen, wo bisher etwa 5,25-Zöller mit 40 Spuren verwendet wurden. Mit Datenaustausch Richtung Atari oder Amiga ist's dann aber erst mal nichts.

Bus-Pin	Siemens FDD 100–8 D	Shugart SA 810/860	Controller-Belegung Janich & Klass, FDC 8/5	
			8"-Shugart-Bus	5,25"-Philips-Bus
2	-	Ext. Wrt. Current Switching (opt.)	Low Current	RD Data
4	-	-	-	Head Load
6	-	-	-	Track
8	-	True Ready (opt.)	-	Index
10	-	Two Sided (opt.)	Two Sided	Low Current/ Terminator
12	Disk Change	Disk Change (opt.)	Fault	Step
14	-	Side Select (opt.)	Side Select	Direction
16	In Use	In Use (opt.)	Fault Reset	Write Gate
18	HDLD	Motor on/Head Load	Head Load	WR Data
20	Index	Index	Index	Select 1
22	Ready	Ready	Ready	Select 2
24	Sector	Sector	-	Select 3
26	Select 0	Drive Select 1/Side Select	Select 1	Motor on
28	Select 1	Drive Select 2/Side Select	Select 2	Rdy 1
30	Select 2	Drive Select 3/Side Select	Select 3	Rdy 2
32	Select 3	Drive Select 4/Side Select	Select 4	Rdy 3
34	Step in	Direction Select/Side Select	Direction	Rdy 4
36	Step	Step	Step	Write Protect
38	Wrt Data	Write Data	WR Data	Terminator
40	Write	Write Gate	Write Gate	Side Select
42	Track 00	Track 00	Track 0	-
44	Wrt Protect	Write Protect	Write Protect	-
46	Raw Data	Read Data	RD Data	Terminator
48	Sep Data (opt.)	Sep Data	-	-
50	Sep Clk (opt.)	Sep Clock	-	-
Laufwerks-Typ	8 Zoll, 77 Spuren, einseitig	8 Zoll, 77 Spuren, ein-/zweiseitig	-	-

Tabelle 3. Ohne tiefere Erklärung auch ein paar typische Bus-Belegungen für 8-Zoll-Laufwerke. Nicht zu vergessen der 50polige 5,25-Zoll-Anschluß von Philips.

ct

Laufen, was heißt laufen? Wenn es laufen würde, brauchte ich den Fehler ja nicht zu suchen.

BÖRSE -- BÖRSE -- BÖRSE

==

Ich kann folgende Sachen günstig besorgen :

1 x HRG 1B Grafik mit 384x192 Punkten von RB-Elektronik
föher kostete der Fertigbaustein 369,- DM
jetzt nur 100,- DM (incl. Software)

1 x Computer Netzteil mit +/- 5V und +/- 12V
60 Watt, komplett fertig:
50,- DM

2 x TEAC A Laufwerke 40 Track SS/DD, slime line
hervorragend für TRS-80
je Laufwerk 90,- DM

1 x 80 Zeichen-Karte von Schmitke-Elektronik für den
VIDEO GENIE I mit Banking-Logig
und Software
200,- DM

nagelneue Drucker, orginalverpackt, 6 Monate Garantie:

Star DP 510 100 Zeichen/s 198,- DM
Star Gemini 10 xi 120 Zeichen/s, IBM-Zeichensatz
298,- DM
Star Delta 10 x 160 Zeichen/s, 8K-Buffer, parallel +
seriell: 398,- DM

Andreas Rychlik; Königsberger Allee 120; 4100 Duisburg 1
==

Angebot für den Bastler:
2 Cassettendrive's von ADAM
1 Tastatur (ADAM)
sowie 3 Programm-Cassetten mit Programmen
für ADAM-Rechner zu verschenken. (gestifet von Peter Spies)
Gesamt oder einzeln bei der Redaktion erhältlich. -p-

Suche MS-DOS Programm
zum lesen von NEWDOS80-
Disketten unter MS-DOS.
P. Spieß
Trugenhofenerstr. 27
8859 Rennertshofen 1

Verkaufe:
TRS 80 M I
MDX 2 Expansoion Interface (defekt) incl. Doubler
Floppystation mit 1 TEAC FD55FV (2X80,DS,DD)
ohne Netzteil 2 BASF 6106 (40,DD)
Drucker DMR 80
gegen Gebot gesamt oder einzeln

Weiterhin abzugeben:
Manuals: TRSDOS, NEWDOS 80, PASCAL 80, Mumath, Musimp,
SUPER DBMS, MZAL-Assembler, Scripsit, ZBASIC,
VISICALC, Alcor Pascal, Fortran 80
gesamt oder einzeln gegen Versandkosten

Josef Konrad — Anzengruber Straße 35 — 8038 Gröbenzell

Suche RS232 für Modell 4
Rüdiger Sörensen, ☎ 06131/32860

Nachträge und andere Mißgeschicke

— = * In VISICALC sortieren * = —

Zu INFO 19 S.29-31

VISICALC bietet ein recht nützliches Instrumentarium: Man kann damit bequem sehr umfangreiche Tabellen schreiben; man kann Formeln eingeben - sogar trigonometrische -, deren Ergebnisse von VISICALC bei jeder beliebigen Änderung automatisch neu berechnet werden - eben weil hierfür nicht Zahlenwerte, sondern Formeln eingegeben werden - man kann (Version 3) Relationen zu Bezugsskalen herstellen lassen, man erhält auf einfache und schnelle Weise Summen oder Durchschnitte (Mittelwerte) von Zeilen oder Spalten, man kann beliebig lange Kommentare eingeben und hierfür die Spaltenbreite anpassen - eine bestimmte Version arbeitet sogar auf Wunsch mit unterschiedlichen Spaltenbreiten im gleichen Rechenblatt! - man kann die Daten in Kombination mit dem Programm *"VISIGRAF"* grafisch darstellen (Diagramme zeichnen); man kann *VISICALC* bis zu einem gewissen Grade sogar als kleine übersichtliche Datenbank benutzen (ich verwalte hiermit meine Eingaben und Ausnahmen): man kann... man kann vieles, aber eben nicht alles, was man mit *BASIC* machen könnte - zum Beispiel: *sortieren!*
Für unseren Bremerhavener Schwesterclub - Verzeihung: Bruderclub ("Club" ist maskulin...) - hatte ich vor Jahren ein Programm geschrieben, mit dem man eine *VISICALC*-Datei in BASIC-Variablen einlesen, diese manipulieren und danach wieder in eine VC-Datei zurückschreiben kann. Voraussetzung hierfür ist, daß die VC-Datei im sogenannten *"DIF-Format"* vorliegt; das ist das system- und sprachunabhängige "DATA INTERCHANGE FORMAT". Wer es näher studieren möchte, lese

Hergert, "Erfolg mit VISICALC" (SYBEX), oder
"DIF: A Format for Data Exchange between Applications Programs"
in BYTE, Nov.81, S.174-206, oder
D.Haan, "What's the DIF?", 80 MICRO Sept.83, p.108-114

BASIC besitzt die sehr vielseitige Sortier-Routine CMD "O". Hiermit können sowohl Numeri als auch Strings sortiert werden, sowohl auf- wie abwärts, nach einer oder gleichzeitig mehreren Feldvariablen, sowohl direkt wie auch indirekt (indiziert). Das vermißte ich bei *VISICALC*.
Ich habe deshalb meine damals veröffentlichte "Utility" dazu verwendet, ein Sortierprogramm für VC-Dateien zu bauen. (Danke für den Hinweis, lieber Leser, daß es dafür den *"VCSORTER/BAS"* gibt; aber der funktioniert nicht! Er setzt außerdem voraus, daß man die zu sortierende VC-Datei vor seiner Anwendung vergewaltigt, da man die zu sortierende Zeile oder Spalte zuerst an den Blattrand "moven" muß; das habe ich nicht so gern...!)
Hat man die VC-Datei erst einmal mit "/S#S,Filename, untere rechte Ecke" im DIF-Format gespeichert, so besorgen Menu und Dialog in meinem Programm alles weitere. Es können unabhängig voneinander Zeilen oder Spalten sortiert werden, auf- oder abwärts, Numeri oder Strings: das Programm erkennt, worum es sich handelt. Da die VISICALC-Spalten keine Nummern, sondern eine alphabetische Kennzeichnung haben (wobei das Alphabet zweieinhalbmal durchlaufen wird, d.h. nach "Z" folgt "AA, AB, ..." usw.), kann man statt einer Nummer auch diese Alpha-Kennzeichnung eingeben; das Programm rechnet sie in den entsprechenden Zahlenwert um.
Das Programm kann auch dazu dienen, eine VISICALC-DIF-Datei lediglich auf dem Bildschirm anzusehen oder auszudrucken, ohne *VISICALC* zu laden (oder zu besitzen). Dabei werden gleichzeitig Zeilen- und Spaltennummer jeder Eintragung angegeben.
Selbstverständlich steht das Programm in unserer Programm-Diskettothek zur Verfügung (jetzt bei Clubkamerad *Werner Förster*).

K.-J. Mühlenbein

Fehler!

In seinem vorletzten Brief (den letzten könnt ihr auf Seite 2 lesen) hat mich Kajott auf ein paar Fehler in meinem Artikel "Extended NewDOS" aufmerksam gemacht. Natürlich darf und will ich diese Fehler nicht unter den Tisch kehren, da eventuell Maschinensprache-Anfänger diese Texte studieren und dann durch falsche Informationen verwirrt werden können! Aus diesem Grund hier ein Ausschnitt aus dem Brief von Kajott:

Ach so, noch das PS wegen der Fehler im vorgenannten Beitrag:
2.Seite unten: Eine Granule hat immer 5 Sektoren, egal bei welcher Disk-Type: zehn Sektoren sind also stets gleich zwei Granules, nicht nur bei SS/SD.
4.Seite (hinter der Directory-Tabelle), im unteren Drittel: die BIT-Folge muß 10110111 lauten und entspricht B7H, nicht B5H.
"Platz im SYS 1", unten: In den aufgeführten 36 Bytes sind nicht "nur relative" Sprünge (JR); in Zeile 50CD steht ein absoluter Sprung (JP Z, M519D) ! Dieser führt allerdings sowieso aus diesem Bytebereich hinaus und ist deshalb für die Verschiebung der 36-Byte-Folge irrelevant.
Nächste Seite, 2.Absatz: Du schreibst, "Der Platz dort ist nur 32 Bytes lang". Das stimmt nicht, er ist sogar nur 23 Bytes lang. (Das kann natürlich auch ein Vertipper sein, ist aber für den interessierten Leser, der daraus lernen will, leider verwirrend, weil er ja (wie ich als Anfänger z.B.) zunächst alles glauben muß, was in einer Info steht! (So geht es mir aber dauernd mit den berühmten Lehrbüchern, denen ich jedes Wort glaube, weil ich, solange ich noch daraus lerne, die vielen Fehler darin ja noch nicht erkenne. Inzwischen habe ich sogar im "GROSSER" einige gravierende Fehler gefunden! Wie soll da einer weiterkommen...
Gleiche Seite, 5.Absatz: Da steht: "...muß man im Sektor 0 rel. die Bytes 25/26 von B750 in BB51 ändern." M.E.muß es BF51 (statt BB51) heißen! Denn die Verschiebung beträgt 264 Bytes, nicht 260!

Mit schlechtem Gewissen gelobe ich Besserung

Hartmut Obermann

Ergänzung/Berichtigung zu "RESET-feste MEMDISK!"

Bei meinem MEMDISK-Patch am Montezuma-CP/M (Info 19 Seite 43/44) ist mir ein Fehler(chen) unterlaufen. Der Fehler resultierte aus dem Umstand, daß ich so kurze Patchs normalerweise nicht mit dem Assembler, sondern per Hand und Liste "assembliere". Um die Sache im Info veröffentlichen zu können, mache ich dann nachträglich einen Assemblerlauf und von Fall zu Fall einen Sektordump.

Hier ist mir der Fehler unterlaufen, daß ich im Assemblerlisting CP "N" statt CP "n" eingetragen habe. Danach habe ich auf einer Musterdiskette das Ergebnis in die Sektoren eingetragen und ausgedruckt. Ausprobiert habe ich diese Version des Patchs nicht mehr, da ja meine per Hand (und mit CP "n") übersetzte Idee funktionierte!

Natürlich funktioniert, rein vom Programm her gesehen, auch das abgedruckte Programm, aber man muß jedesmal SHIFT drücken und das kann man sich wirklich ersparen! Wie Ihr seht, entstehen Fehler manchmal auf sehr seltsame Weise und können nur durch konsequente Dokumentation dessen, was man wirlich gemacht hat, vermieden werden!

Hartmut Obermann

Der Markt für Textverarbeitungssysteme

Im Schatten des Dinosauriers

München (sg) — Trotz integrierter Programme und, seit einiger Zeit auch Desktop Publishing, sagen Marktforscher für den Bereich Textverarbeitungs-Software auch zukünftig stabile Verhältnisse voraus. Die Zeiten des stürmischen Wachstums wie zu Beginn der 80er Jahre sind jedoch vorbei — nur noch wenige PC-Anwender setzen ihr System ausschließlich als Ersatz für die Schreibmaschine ein.

Auf jedem vierten Personal Computer läuft hierzulande unter anderem auch ein Textverarbeitungssystem. Nach dem Marktforschungsinstitut IDC, das zuletzt 1985 eine Analyse dieses Software-Bereichs vornahm, existierten zu dieser Zeit entsprechende Programme im Gesamtwert von 204 Millionen Mark. Jedes Jahr kommen rund 40000 Textverarbeitungspakete hinzu.

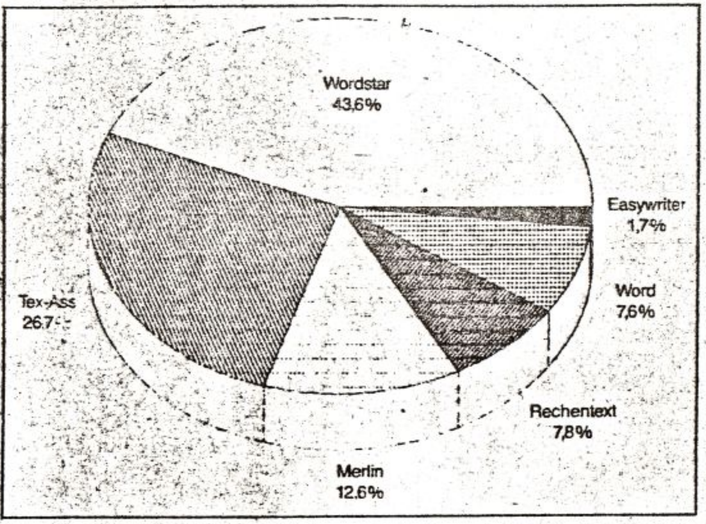
Die Marktanteile und Umsätze gelten als stabil, was nicht immer ein gutes Zeichen sein muß. Im Bereich Textverarbeitung bedeutet dies, daß von über 130000 installierten Programmen in der Bundesrepublik knapp 70 Prozent auf ein Textverarbeitungssystems namens Wordstar entfallen. Böse Zungen behaupten, besonders die noch weit verbreiteten älteren Versionen des Programms besäßen echte Dinosaurier-Qualitäten: groß, schwerfällig, unflexibel und dabei alles beherrschend.

Ein paar Eigenschaften stehen tatsächlich in keinem Zusammenhang mit der enormen Marktdurchdringung des Systems. Nobody is perfect: Unbestreitbar haben alle auf dem Markt erhältlichen Programme ihre »Problemstellen«, doch deren Anteil an der Installationsbasis ist vergleichsweise gering — nach Wordstar kommt eine Weile nichts, dann tauchen Konkurrenzprodukte wie TexAss, Word und Trend mit deutlich unter 10 Prozent auf.

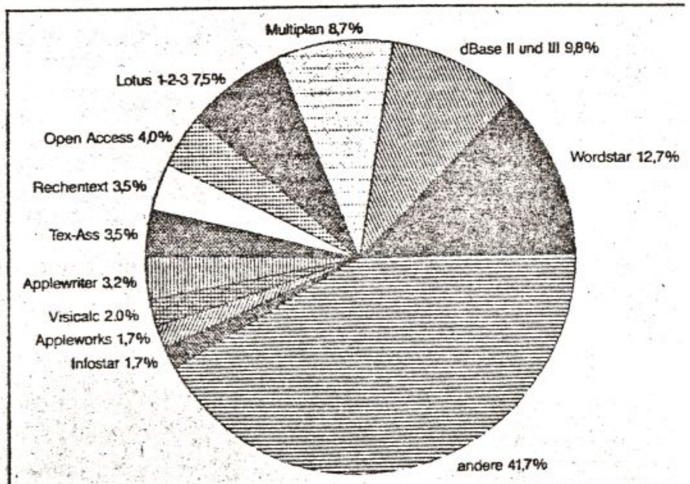
Hersteller Micropro räumt denn auch freimütig ein, bei der Weiterentwicklung seines Paradeprodukts geschlafen zu haben. Mit der Vorstellung von Wordstar-Erweiterungen und neuen Releases sei man aber nun aufgewacht. Zeit wurde es offenbar, denn die Nummer eins im Markt für Textverarbeitung steckte bis vor kurzem in den roten Zahlen. Dies hat sich laut Leon Williams, President der Micropro Corporation, im vergangenen Quartal geändert — bei einem Umsatz von 9,3 Millionen Dollar habe man einen Gewinn von 550000 Dollar erwirtschaftet.

Die deutsche Niederlassung behauptete laut Marktforscher IDC 1985 mit Wordstar seine Spitzenposition. Von den insgesamt 38346 ausgelieferten Textverarbeitungspaketen entfiel ein Anteil von 43,6 Prozent auf das GmbH-Produkt. An zweiter Stelle lag das Programm Tex-Ass der Firma Bongartz & Schmidt (26,7 Prozent), gefolgt von Brains Merlin, MIS Rechentext (7,8 Prozent), Microsofts Word (7,6 Prozent) sowie Easywriter von Computer Associates (1,7 Prozent).

Wie häufig der Personal Computer als Ersatz für die Schreibmaschine eingesetzt wird, verdeutlich eine Studie der Intelligenz Electronics Europe.



Marktanteile für Textverarbeitungspakete nach Stückzahlen — 1985 wurden in Deutschland 38346 Programme (inklusive Lizenzen und OEM-Auslieferungen) für PCs installiert Quelle: IDC



Textverarbeitung hält die Spitze — Anteil am deutschen Softwar Gesamtmarkt nach Stückzahlen Quelle: Intelligent Electronic Euro

Der bundesdeutsche Gesamtmarkt für Software wurde Mitte '85 von einem Textprogramm angeführt — Wordstar hielt nach Stückzahlen einen Anteil von 12,7 Prozent. Erst auf den weiteren Plätzen folgten integrierte Programme wie dBase II und III (9,8 Prozent), Multiplan (8,7 Prozent), Lotus 1-2-3 (7,5 Prozent) sowie das Programm Open Access (4,0 Prozent).

Allerdings, und dies könn den Anbietern von reinen Tex verarbeitungspaketen mittelfi stig Probleme bereiten, setz immer weniger Anwender ihr PC ausschließlich als Textsyste ein. Nach dem stürmisch Wachstum zu Anfang der 80 Jahre wird sich dieser Berei laut IDC als eigenständiges Se ment für Basis-Applikation lediglich behaupten.

Auf zu neuen Ufern!

Liebe Clubfreunde!

Im letzten Info hat unser Feund Hans Martin Stephan einen Vorschlag zur Diskussion gestellt, der mir schon öfter von ausscheidenden Mitgliedern (die meisten "Aussteiger" sind "Aufsteiger" in die PC/Atari-Welt) angetragen wurde. Ich möchte hier kurz meinen Standpunkt dazu erläutern und hoffe, daß sich auch andere zu Wort melden!

Ich habe das Problem der Abwanderung von Tandy TRS80-Besitzern in die "Kompatibilität" schon lange beobachtet und mir immer wieder die Frage gestellt, wie man dieser Entwicklung entgegentreten könnte. Eine Möglichkeit die Attraktivität der TRS 80-Computer zu steigern ist meiner Meinung nach die vermehrte Unterstützung von CP/M im CLUB 80 und im Info.

Früher bildeten die TRS 80-Besitzer für Softwareanbieter einen interessanten Kundenkreis. Das sah man daran (und sieht man noch heute), daß sehr viele Sprachen auf den Tandy-Maschinen implementiert wurden. Nicht nur Tandy hat sich da, zusammen mit Microsoft, sehr engagiert, auch und vor allem kleinere Entwickler versuchten sich den TRS80-Markt zu erobern. Wäre z.B. der heute sehr verbreitete Turbo-Pascal-Compiler vor fünf oder sechs Jahren auf den Markt gekommen, er wäre ohne Zweifel auch unter TRS-/NewDOS zu haben, vielleicht wäre er sogar für dieses Betriebssystem entwickelt worden. Heute ist das anders geworden! Kaum eine(s) der neuen Sprachen(Programme) wird noch auf unsere Maschinen übertragen; wir stehen ziemlich im Abseits!!!

Durch die vermehrte Verbreitung von CP/M-Maschinen (Schneider CPC/Joyce, Commodore C128 usw.) ist dieses Betriebssystem, das sich schon in Gefahr befand, durch MS-DOS u.ä. verdrängt zu werden, wieder zum Leben erweckt worden. Wie groß die Aktivitäten in diesem Bereich sind, kann man sich kaum vorstellen! Nicht nur die sehr umfangreiche Public-Domain-Software, die erhältlich ist, auch die neuesten Sprachen (u.a. das schon erwähnte Turbo-Pascal, Modula, Prolog, Lisp) und viele kommerzielle Programme (WordStar, DBase, Multiplan usw. usw.) verleihen dem eher schlechten Betriebssystem CP/M große Attraktivität. Viele unserer TRS80-Maschinen können (Model 1 und III sowie Genie I und II nach Umbauten; Model 4/4p, Genie III, IIs, IIIs sofort) auch unter CP/M betrieben werden und damit am Aufschwung dieses DOS teilhaben.

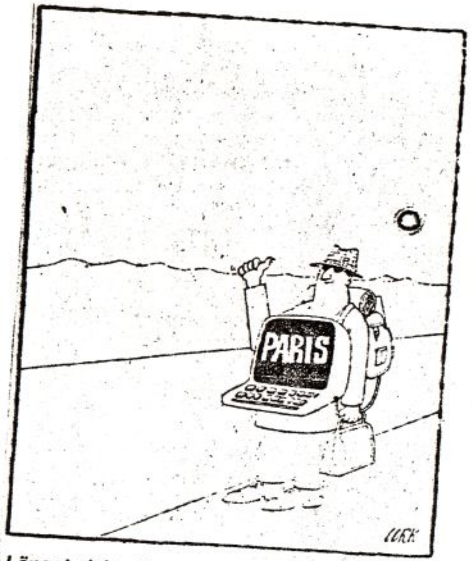
Darin sehe ich auch die Chance eines Brückenschlags zu denen, die sich trotz dieser neuen Aussichten nicht davon abhalten lassen, auf die Kompatiblenwelle aufzuspringen. Turbo-Pascal, "C", Prolog, DBase und vieles andere mehr läuft auf beiden Systemen, sowohl auf CP/M als auch auf MSDOS. Es ließen sich also durchaus Gemeinsamkeiten finden, trotz der, auf den ersten Blick unüberwindlichen Kluft!

Ich (und viele weitere Clubmitglieder) arbeite schon einige Zeit mit CP/M und werde dieses Thema in nächster Zeit durch entsprechende Veröffentlichungen im Info mehr unterstützen. Natürlich hoffe ich dabei auf Hilfe der anderen CP/M-User. Auch bin ich jederzeit dafür, im Info eine MS-DOS-Ecke einzurichten und zu unterstützen. Diese Aktivitäten dürfen aber nicht dazu führen, daß sich der CLUB 80 weg von einem "Club der Tandy-Freunde" und hin zu einer CP/M-UserGroup oder einem MS-DOS-Verein entwickelt. Vielmehr muß es so sein, daß die neuen Zweige zu zusätzlichen Standbeinen des CLUB 80 werden!

Zu guter Letzt möchte ich noch ein rein organisatorisches Problem in Punkto MS-DOS-Ecke ansprechen. Selbstverständlich wird schon jetzt kein Artikel abgewiesen, weil er sich mit diesem Thema beschäftigt. Wenn also jemand über MS-DOS, Turbo-Pascal oder ähnliches schreiben will, kann er das jederzeit tun! Die Frage ist nur die, ob es für eine solche Einrichtung auch Leute gibt, die bereit sind etwas zu schreiben. Gelesen wird es auf alle Fälle!

Damit genug zu diesem Thema! Ich hoffe auf eine angeregte Diskussion, Euer

Hartmut Obermann



Längst nicht alle Einsatzbereiche für Personalcomputer sind hinreichend bekannt

PC statt GENIE ?

Seit ich mein GENIE I 1983 kaufte, ist der Umgang mit dem Computer zu meinem gewohnten Arbeitsmittel geworden. Aus meinem GENIE habe ich zuletzt auch wohl so viel wie möglich herausgeholt: 3,5 MHz, verbessertes DOS etc.
Dennoch schreibe ich diesen Text jetzt auf meinem neuen PLANTRON PC XT mit 20 MByte Festplatte, den ich vor kurzem für 3.500,- DM kaufte. Wie ich aus verschiedenen Texten und Gesprächen entnehme, denkt auch von Euch der ein oder andere in dieser Richtung nach. Deshalb möchte ich etwas zum "Informationshaushalt" beitragen:
Insgesamt ist ein IBM-Kompatibler schon als Verbesserung zum GENIE I und II anzusehen; dennoch sind die Unterschiede gerade im Vergleich zum IIs und IIIs beim PC nicht so wesentlich. Wer hier zuschlagen will, sollte also nicht überzogene Erwartungen hegen. Die folgenden Erfahrungen habe ich bei mir gemacht:

Geschwindigkeit
Mein PC arbeitet ursprünglich mit dem 8088-2 bei wahlweise 4,7 und 8 MHz. Diese "Geschwindigkeit" wird allerdings mehr oder weniger durch ein ineffizientes MS-DOS mühelos vernichtet. Ich habe deshalb sogleich den Prozessor durch den NEC V20, der Z80 kompatibel ist (!), ersetzt. Dies läßt sich leicht durchführen und damit läuft meine Mühle ca. 3 mal schneller als der Ursprungs-PC. Dies reicht für die meisten Anwendungen aus, allerdings ist dies wenig im Vergleich zu den 5,4 (7,2) des AT mit 6 (8) MHz oder ca. 18 fachen Geschwindigkeit eines 80386-Rechners.
Ein Vergleich: Eine BASIC-Anwendung auf dem GENIE benötigte für 100.000 Durchläufe rund 3 Minuten: der PC brauchte dafür 8 in BASIC und sogar - wer hätte das gedacht - 11 Minuten in "C".

Erweiterungen
Im Hinblick auf den Systemaufbau mit vielen Erweiterungssteckplätzen wäre diese Mühle allerdings ideal für einen Freak. Wenn man was basteln will, geht dies hier wesentlich einfacher, als beim GENIE. Ich habe beim GENIE z.B. nach dem Einbau der seriellen Schnittstelle, Speed up etc. eine ungeheuere Menge von Drähten legen müssen. Viel mehr ist kaum mehr zu überschauen.
Der PC kann dagegen leicht mit Steckkarten ausgebaut werden, z.B. existieren 2 MB RAM Extensionskarten, Grafikkarten und Schnittstellen. Der RAM kann leicht intern bis auf 1 MByte ausgebaut werden; diese werden vom Prozessor noch direkt adressiert z.B. als RAM-Disk.
Für mathematische Berechnungen habe ich einen 8087 Co-Prozessor installiert. Leider benutzen nur wenige Standardprogramme die 20-30fache Rechengeschwindigkeit und höhere Genauigkeit wirklich aus.

Zuverlässigkeit, Handhabung
In dieser Hinsicht ist gerade der PLANTRON vorbildlich. Die AT-Tastatur ist sehr Bedienungsfreundlich, getrennte Cursor-Tasten, ergonomische Tastenform, 12 Funktionstasten, Leuchtanzeigen für GROSSSCHRIFT, Num Lock, Power.
Die Disketten sind nahezu unverwüstlich. Selbst die billigen Marken laufen ohne Schwierigkeiten.
Der hochauflösende bernsteinfarbene Monitor ist absolut flimmerfrei und endlich können auch 80 Zeichen pro Zeile dargestellt werden, was bei uns ja nur bei den modernen GENIE's funktioniert.
Die eingebaute Festplatte gehört zwar nur zu den Mittelschnellen, dennoch ist es ein großer Geschwindigkeitsunterschied gegenüber den Diskettenlaufwerken. Mehr Leistung ist auch erforderlich, da schon dieses Textprogramm über 200 kB benötigt; Overlays noch nicht mitgerechnet.
Probleme, wie Abstürze durch Hardware-Fehler, wie sie mein GENIE im Sommer manchmal hatte, kommen - wohl auch Dank eingebautem Gebläse - nicht mehr vor.

Betriebssystem
MS-DOS ist mehr oder minder ein "aufgemotztes" CP/M-80 und läßt in vielen Eigenschaften selbst gegenüber dem recht guten NewDOS 80 Lücken. Z.B. gibt es kein LIB: kein BOOT, kein CLOCK (UHR). Wenn man beim formatieren eine Taste festhält (was möglich ist) meckert das DOS und verabschiedet sich schließlich durch einen HALT, d.h. Ausschalten wird ohne RESET-Knopf erforderlich.
Positiv an der bei mir installierten Version 3.2 ist allerdings, daß viele Ergänzungen bereits implementiert sind und manches besser läuft. Z.B. gehört die RAM-Disk zum Standard. Auf der Festplatte können viele Unterverzeichnisse verwaltet werden.
Zum Betriebssystem werden auch eine Reihe von ergänzenden Programmen angeboten, z.B. zum sortieren des Inhaltsverzeichnisses, zum Zurückholen von versehentlich gelöschten Dateien usw. Diese Programme müssen aber zusätzlich teuer erworben werden.
Neben MS-DOS können auch andere Betriebssysteme verwendet werden, z.B. CP/M-86 und Concurrent PC-DOS.

Anwendungs-Software
Aufgrund des größeren Arbeitsspeichers und des vorwiegend kommerziellen Einsatzes bieten Softwarehäuser komplexere Programme an, als bei uns üblich und möglich sind. Allerdings sind die Unterschiede nicht überall groß und finden sich für viele Eigenschaften Vorläufer aus der CP/M und NewDOS-Welt.
Wegen WordStar kann man dabei ruhig bei CP/M bleiben, es läuft nur unwesentlich benutzerfreundlicher. Ich kann mir vorstellen, das im GENIE IIIs sogar mehr Power dafür bereit steht.
WordPerfect ist ein neues amerikanisches Textprogramm, mit einer Reihe von Eigenschaften, die ich schon länger suchte: erweiterbares Wörterbuch, über 200 Druckertreiber, gleichzeitige Bearbeitung von 2 Texten, beliebige Textlänge, also auch z.B. 200 Seiten im Arbeitsspeicher, automatische Sicherungskopien z.B. nach 5 Minuten, ohne daß der Benutzer eine Taste drücken muß (!) (welcher GENIEText-User würde sich dies nicht wünschen).
Weitere Eigenschaften: Rechnen im Text, Spaltenverarbeitung, Macros, PS-Schrift mit Randausgleich.
Wie aber z.B. Arnulf mit seinen vielen Verbesserungen gezeigt hat, kann vieles davon auch auf dem GENIE entwickelt werden.
Auf dem Markt werden neben Grafik-Programmen auch sogenannte integrierte Programme angeboten (OPEN ACCESS II, Framework II, SYMPHONY). Regelmäßig werden dabei Tabellenkalkulation, Textverarbeitung und Datenbank unter einer Benutzeroberfläche zusammengeführt, was insbesondere jeden freuen wird, der z.B. Serienbriefe drucken möchte oder Tabellen und Grafiken in einen Text integrieren möchte. Die einzelnen Programmfunktionen sind dabei mittlerweile recht brauchbar fortentwickelt, dennoch sind spezielle Programme für jede Einzelaufgabe i.d.R. mächtiger.

Transfer
Dateien können mit speziellen Programmen von NewDOS, G-DOS und CP/M-80 nach MS-DOS transferiert werden. Insofern spart man die nochmalig Erfassung.

Programmiersprachen
Unter MS-DOS werden inzwischen alle gängigen Programmiersprachen angeboten und auch noch laufend fortentwickelt.

Zusammenfassung
Mit der Maschine bin ich sehr zufrieden. Dabei ist sicherlich die berufliche Nutzung von wesentlicher Bedeutung: Ich schreibe viele Stunden wöchentlich daran. Im Vergleich zu allen übrigen Maschinen auf dem Markt hat mir insbesondere die sehr solide Verarbeitung und der günstige Preis gefallen. Die Basisversion mit 1 Laufwerk kostet rd. 1300,- DM wesentlich weniger, als ich für mein GENIE damals bezahlt habe! Wer allerdings mit dem Geld nicht so knapp ist und eher mit kommerziellen Programmen umgehen möchte, sollte unbedingt den AT vorziehen oder besser noch den immer billiger werdenden 386.
Dagegen kann ich nicht zu Schneider PC und verschiedenen Billigst-PC raten: diese Geräte sind entweder nicht für Dauerbelastung ausgelegt oder haben sonstige Nachteile, z.B. schlechte Tastatur, geringe Erweiterungsmöglichkeiten usw.
Wer sich für meine sehr preisgünstige "Quelle" interessiert, dem stehe ich gerne zur Verfügung.

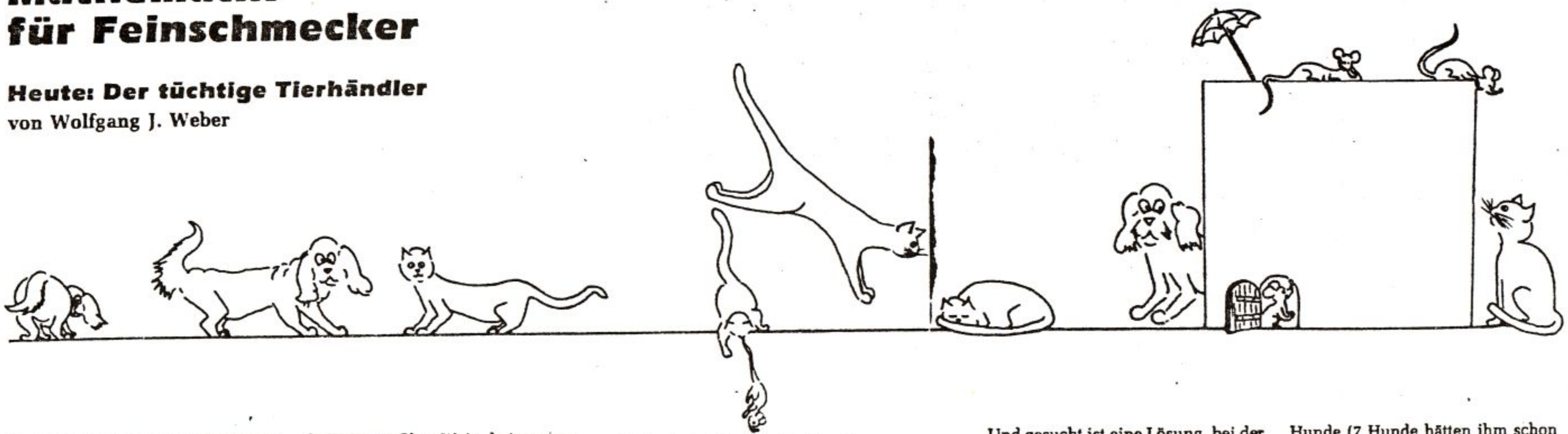
Zum Club
Ich bin an einem Erfahrungsaustausch zum PC interessiert, meine aber auch, daß in unserem Tollen Info hier nicht kostbarer Platz verschwendet werden sollte. Über Kontakte zu diesem Thema freue ich mich allerdings.

Paul-Jürgen Schmitz
5. Juli 1987

Mathematatik für Feinschmecker

Heute: Der tüchtige Tierhändler

von Wolfgang J. Weber



Im folgenden werden wir drei Hauptpersonen kennenlernen. Nomen est omen – manchmal jedenfalls. Da ist zunächst Herr Witzig, ein schneller Kopf, der allerdings gelegentlich gehörig neben das Ziel schießt. Herr Witzig programmiert gerne, auch in solchen Fällen, in denen es andere Wege zum Erfolg gäbe. Zweitens ist da Herr Denkmann, der ziemlich genau das Gegenteil von Herrn Witzig verkörpert: bedächtig in Wort und Tat, möchte er immer erst einen Vorgang vollständig verstanden haben, bevor er etwas unternimmt. Nur böswillige Beobachter interpretieren dies als Mangel an Initiative. Als Dritten treffen wir auf Herrn Rechenberg. Ihm macht die Mathematik großen Spaß, und er stellt seinen Mitmenschen gerne Denksportaufgaben mathematischer Natur. Am liebsten sind ihm übrigens solche Aufgaben, für die ihm schon gleich zu Beginn die Lösung mitgeteilt wurde, daher ist es auch nicht verwunderlich, daß Herr Rechenberg sich auf diese Weise nicht immer Freunde schafft. Hingegen dürfte vielen Lesern von PASCAL-International sein Hobby nicht ganz fremd sein, denn das Programmieren und das Problemlösen sind ja eng miteinander verwandt. Hören wir, was er auf Lager hat.

An diesem Abend sitzt Herr Rechenberg als Gastgeber gemütlich mit seinen Freunden Denkmann und Witzig bei einem Glas Wein beisammen. Man hat gerade seine jeweiligen Meinungen über die Politik, das Wetter, Virusprogramme und andere Landplagen dargestellt und zwanglos weitgehende Einigkeit erzielt. Das Gespräch droht zu verstummen, doch da hätte man Rechenberg unterschätzt.

»Stellen Sie sich mal einen Tierhändler vor, der an einem Tag 100 Tiere verkauft;«

»Aber lieber Rechenberg, es wurde gerade gemütlich, Ihr Wein ist auch ganz hervorragend, und jetzt wollen Sie uns vermutlich wieder eine erbauliche Geschichte über die unerschöpflichen Geheimnisse der Arithmetik erzählen«, protestiert Witzig.

»Ja, und zwar wird es Sie vielleicht sogar interessieren. Also, unser Tierhändler verkauft 100 Tiere, und er verdient dabei genau 100 Mark.«

»Und jetzt sollen wir sagen, wie alt er ist?« blödelt Denkmann.

Rechenberg: »Meine Herren, wo bleibt das Niveau? Nein, er verkauft 100...«

»Gut, gut, das wissen wir«, unterbricht ihn Witzig, der sich damit abgefunden hat, daß er wieder einen Beweis seiner stets wachen Intelligenz erbringen muß, um sein Glas Wein zu verdienen. Es gibt eben nichts umsonst im Leben. »Was hat er verkauft und was hat er im einzelnen eingenommen?«

»Verkauft hat der Tierhändler Hunde, Katzen und Mäuse. Er hat an jedem Hund 15 DM, an jeder Katze 1 DM und an jeder Maus 25 Pfennige verdient.«

»Lausige Gewinnspannen, wenn Sie mich fragen. Der wäre besser Apotheker geworden. Aber was sollen wir jetzt rauskriegen?« fragt Witzig.

»Vermutlich, wieviele Tiere er von jeder Sorte verkauft hat«, meldet sich Denkmann.

»Stimmt, meine Herren, das ist die Aufgabe, der Sie sich doch sicher gewachsen zeigen werden?« Der Tonfall Rechenbergs beschwört böse Ahnungen in den beiden anderen herauf, daß sich dieser Abend noch etwas in die Länge ziehen könnte. Vielleicht ist die Aufgabe sogar unlösbar?

»Drei unbekannte Größen, aber nur zwei Gleichungen, das ist nicht fair«, protestiert Witzig.

»Doch. Bedenken Sie, daß wir nicht irgendeine Lösung finden sollen, sondern eine ganzzahlige«, belehrt ihn Denkmann. »Also wir haben, wenn ich zusammenfassen darf, zwei Gleichungen.

Gleichung 1 lautet:

Hunde + Katzen + Mäuse = 100

Gleichung 2 lautet:

15 × Hunde + 1 × Katzen + 0,25 × Mäuse = 100

Und gesucht ist eine Lösung, bei der keine Drittelhunde oder Viertelkatzen auftreten, und natürlich auch keine negativen Zahlen.«

Alle schweigen; Rechenberg blickt gespannt auf seine Freunde, die grübelnd mit ihren Kugelschreibern hantieren und Zettel bekrakeln. »Darf ich Ihnen noch etwas Schreibpapier anbieten? Bitte bedienen Sie sich doch«, will Rechenberg gerade sagen, doch da schaut Witzig triumphierend in die Runde.

»Ich hab's. Er hat Null Hunde, Null Mäuse und 100 Katzen verkauft;«

»Ach, da enttäuschen Sie mich aber, lieber Witzig.« Rechenberg ist echt überrascht, denn an diese triviale Lösung hatte er selbst noch nicht gedacht. »Streng genommen ist auch das eine Lösung, doch wurde ja gesagt, daß der Tierhändler tatsächlich Hunde, Katzen und Mäuse verkauft hat, also von jeder Sorte mindestens ein Exemplar.«

Witzig gibt sich geschlagen, er hatte zu früh triumphiert. »Ja, dann darf ich Sie, lieber Rechenberg, bitten, mir Ihren Computer zur Verfügung zu stellen. Ich werde jetzt alle möglichen Fälle durchgehen lassen und testen, bei welchen die gegebenen Bedingungen zutreffen.«

»Wird das nicht sehr lange dauern?« gibt Denkmann zu denken.

»Das glaube ich nicht, denn der Händler kann nur zwischen 1 und 6 Hunde (7 Hunde hätten ihm schon mehr als 100 DM eingebracht), zwischen 1 und 100 Katzen und zwischen 1 und 400 Mäuse verkauft haben. Das macht 6 × 100 × 400, also 240000 Fälle, für einen Computer ist das keine problematische Sache. Also wo ist der Hauptschalter von diesem Kasten?« Denkmann: »Viel Spaß Herr Kollege, ich finde, daß unser Prozessor im Kopf zur Lösung der gestellten Aufgabe ausreichen sollte. Ich brauche nur Papier und Bleistift.«

An dieser Stelle wollen wir das gelehrte Kollegium verlassen und uns unsere eigenen Gedanken machen. Wie wird Witzig das Problem programmieren und wie wird Denkmann an die Sache herangehen?

Als Tip zum Programmieren: Die Anzahl der verkauften Mäuse muß durch 4 teilbar sein, und sie darf natürlich 100 nicht übersteigen (weil nur 100 Tiere insgesamt verkauft werden). Damit sind wesentlich weniger Fälle zu untersuchen als Witzig oben vermutete. Aber auch Denkmanns Ansatz ist sinnvoll: die Aufgabe kann durch schlichtes Nachdenken gelöst werden, bei dem als Hardware Papier und Bleistift ausreichen.

Lösung auf Seite 89

Lösung von Seite 86

Die Aufgabenstellung lautete: Ein Tierhändler verkauft **100 Tiere und realisiert dabei einen Gewinn von DM 100**. Bei den verkauften Tieren handelt es sich um Hunde (Gewinn je 15 DM), Katzen (Gewinn je 1 DM) und Mäuse (Gewinn je 1/4 DM). Wieviele Tiere von jeder Art wurden verkauft?

Zur Lösung sind zwei Zugänge möglich. Erstens Witzigs Ansatz: Man kann alle möglichen Kombinationen aufzählen und jeweils überprüfen, ob die beiden Bedingungen erfüllt sind. Das wird im folgenden Pascal-Programm getan (s. Listing).

Zweitens Denkmanns Lösung:

Man kann die Gleichungen genauer betrachten und noch mehr Schlüsse aus ihnen ziehen. Die Gleichungen sind:

Gleichung (1):

Hunde + Katzen + Mäuse = 100

Gleichung (2):

15 × Hunde + 1 × Katzen + 0,25 × Mäuse = 100

Es wurde bereits erwähnt, daß die Anzahl der verkauften Mäuse durch 4 teilbar sein muß, denn in Gleichung (2) treten ja überall ganze Zahlen auf, und deshalb muß auch der Summand 0,25 × Mäuse ganzzahlig sein. Die Gleichung (2) kann mit der Gleichung (1) kombiniert werden:

Hunde + 1 × Katzen + 0,25 × Mäuse = Hunde + Katzen + Mäuse

Die Katzen fallen algebraisch nicht mehr ins Gewicht, damit erhalten wir eine Gleichung mit nur noch zwei Unbekannten

15 × Hunde + 0,25 × Mäuse = Hunde + Mäuse

oder nochmals umgeformt

14 × Hunde = 3/4 × Mäuse

was besser in der folgenden Form geschrieben wird:

56 × Hunde = 3 × Mäuse

Nun betrachten wir die Primteiler der linken und der rechten Seite: Die Anzahl der Hunde muß durch 3 teilbar sein, weil 56 nicht durch 3 teilbar ist. Also kommt als Wert für die Unbekannte Hunde nur noch 3 oder 6 in Frage. (0 scheidet aus, weil ja mindestens ein Hund verkauft wurde.) 6 kommt als Lösung nicht in Frage, weil dies mit 112 verkauften Mäusen verknüpft wäre, was laut Aufgabenstellung nicht der Fall sein kann. Aus Hunde = 3 folgt Mäuse = 56. Weil insgesamt 100 Tiere im Spiel sind, wurden 100 − 56 − 3 = 41 Katzen verkauft. Wir machen die Probe:

Gleichung 1:

3 Hunde + 41 Katzen + 56 Mäuse = 100 Tiere

Gleichung 2:

15 × 3 + 1 × 41 + 0,25 × 56 = 45 + 41 + 14 = 100

Damit ist die Aufgabe gelöst.

Wir wollen uns der Bewertung enthalten, welcher der zwei vorgestellten Zugänge besser ist: wenn man einen Computer zur Verfügung hat, dann darf man ihn auch benutzen, wie Herr Witzig das getan hat. Andererseits ist es aber auch nicht verboten, den eigenen biologischen Computer zu benutzen, sofern man ihn bedienen kann, wie Herr Denkmann es uns vormacht. Zu einem sollte uns unser Kopf aber ganz bestimmt zu schade sein: alle möglichen Fälle einfach stumpf und blöde der Reihe nach durchzugehen und die Aufgabe durch direktes Probieren zu lösen. Wozu gibt es schließlich Computer?

(Wolfgang J. Weber)

```
PROGRAM tierhandel;

{ 100 Tiere werden verkauft, der Gewinn betraegt insgesamt 100 DM.
  Gewinnspannen: pro Hund: 15 DM, pro Katze 1 DM, pro Maus 0,25 DM.
  Wie viele Tiere von jeder Art wurden verkauft?
  Programmiersprache: Turbo-Pascal, Autor: W.J.Weber, Dezember 1986 }

VAR hunde, katzen, maeuse: INTEGER;

BEGIN
  ClrScr;
  GotoXY(10,10); WriteLn('Der tuechtige Tierhaendler');
  GotoXY(10,11); WriteLn('--------------------------');
  GotoXY(1,15);
  FOR hunde := 1 TO 6 DO
    FOR katzen := 1 TO 100 - hunde DO
    BEGIN
      maeuse := 4;
      REPEAT
        IF (hunde + katzen + maeuse = 100) AND
           (15 * hunde + katzen + (maeuse DIV 4) = 100) THEN
        BEGIN
          Write('Loesung gefunden: Es wurden');
          Write(hunde:3,' Hunde,',katzen:3,' Katzen und');
          WriteLn(maeuse:3,' Maeuse verkauft.');
        END;
        maeuse := maeuse + 4;
      UNTIL katzen+maeuse >= 100;
    END; { FOR katzen }
END.
```

Mathematik für Feinschmecker

Heute:
Wie man durch Würfeln theoretisch vielleicht reich, praktisch aber meistens arm wird

von Wolfgang J. Weber

Wieder treffen wir auf die Herren Rechenberg, Denkmann und Witzig, die regelmäßigen Lesern von PASCAL keine Unbekannten mehr sind. Heute diskutieren sie über ein Problem der Wahrscheinlichkeitsrechnung, bei der ja bekanntlich selbst Fachleute gelegentlich Trugschlüssen erliegen.

Rechenberg hat seinen letzten Urlaub in Südamerika verbracht. Dort lernte er in einem Spielcasino ein verblüffend einfaches Gewinnspiel kennen, bei dem nur gewürfelt wird. Spielregel Nr 1 ist: der Spieler darf viermal würfeln. Spielregel Nr 2 ist: wenn dabei (mindestens) eine Sechs erwürfelt wird, dann hat der Spieler gewonnen. Die Rate der Gewinnauszahlung ist 5:3, das heißt, wenn er z.B. umgerechnet DM 3 einsetzt und gewinnt, erhält er DM 5 = DM 3*(5/3) ausgezahlt. Bei einem Einsatz von DM 60 ergibt sich im Gewinnfall eine Auszahlung von DM 100, usw.

Unser Held Rechenberg hatte hier sofort seine Chance gesehen. »Mag sein,« dachte er sich, »daß diese Latinos noch nichts von Wahrscheinlichkeitstheorie gehört haben, jedenfalls haben sie sich hier offensichtlich verkalkuliert. Jedes Kind weiß, daß die Wahrscheinlichkeit, bei einem Wurf eine Sechs zu werfen 1/6 ist. Bei vier Versuchen steigt die Wahrscheinlichkeit logischerweise auf den vierfachen Wert, also auf 4/6 oder 2/3. Das heißt«, folgert Rechenberg nun messerscharf, »daß man durchschnittlich in zwei von drei Spielen einen Gewinn erwarten darf.« Auf lange Sicht könne er daher nur gewinnen, glaubte Rechenberg. Zu Beginn wollte er mit einem festen Einsatz von umgerechnet jeweils DM 60 spielen. Bei drei Spielen würde er durchschnittlich zwei Gewinne zu je DM 100 einstecken und dabei drei Einsätze zu je DM 60 verlieren: das macht einen erwarteten Gewinn von DM 200 − DM 180 = DM 20. Pro Spiel würde er dabei also im Durchschnitt DM 6,66 gewinnen, was zwar nicht dramatisch viel aber immerhin mehr als 10% des Einsatzes wäre.

So weit die Theorie. Leider sah die Praxis nicht ganz so golden aus. Eher trübe, um ganz offen zu sprechen. Denn Rechenbergs Rechnung ging nicht auf – er verlor mehr und mehr Geld, und die Wende zum stetigen Gewinn wollte sich, entgegen der Theorie, nicht einstellen. Anfangs dachte er noch, es handele sich um zufallsbedingte Schwankungen, die sich über kurz oder lang mehr als ausgleichen würden. Er war sich doch so sicher, daß er das Spiel richtig analysiert hatte! Nachdem sich seine Verluste aber auf weit über hundert DM beliefen, verlor er dann auch noch etwas die Nerven und erhöhte seinen Einsatz unkontrolliert. Das Ergebnis des Casinobesuchs war schließlich: er hatte einiges Geld verloren, seine Nerven waren ruiniert... und seine Freunde hatten etwas zu lachen. Rechenberg konnte sich nämlich nicht enthalten, nach seiner Rückkehr, bei geeigneter Gelegenheit, seinen Freunden Witzig und Denkmann von diesem Mißgeschick zu berichten.

»Das überrascht mich gar nicht, lieber Rechenberg. Diese Casinoleute sind doch Profis. Und Sie, seien Sie mir nicht böse, sind ein Dilettant, jedenfalls, was das Glücksspiel angeht. Es ist doch klar, wie ein solcher Zweikampf ausgehen muß!« Vor Witzigs guter Laune und seinen klugen Bemerkungen gibt es kein Entrinnen.

»Und wo, bitte schön, liegt der Fehler in meiner Argumentation?« Rechenberg findet seinen Freund heute gar nicht witzig.

»Ein kleiner Rechenfehler, mehr nicht«, murmelt Denkmann. »Sie sind da aber in guter Gesellschaft, in bester adeliger Gesellschaft sogar. Haben Sie schon einmal vom Chevalier de Méré gehört?«

»Franzose, aha. Baut er vielleicht Wein an? Chateau de Méré«

»Nicht ganz. Der Herr ist schon länger tot, es sind ziemlich genau 300 Jahre. Dieser Mr. ging in die Geschichte der Mathematik dadurch ein, daß er Blaise Pascal eine interessante Frage stellte.« Auch Denkmann befindet sich offenbar in guter Laune. »Mir wäre Rechenbergs Mißgeschick sicher erspart geblieben,« denkt er sich. »Manchmal ist selbst Bildung etwas wert.«

»Nun spannen Sie uns doch nicht derart auf die Folter«, drängt ihn Witzig. »Um ehrlich zu sein, mir leuchtet die Argumentation unseres Freundes Denkmann ganz und gar ein. Nur die allgemeine Lebenserfahrung hätte mich eventuell von einem gleichartigen Verlust bewahrt.«

»Meine Herren, die Lösung sollten Sie selbst finden.« Denkmann ist unerbittlich. »Es geht sogar auf zwei Arten. Als einen Weg könnten Sie einfach am Computer simulieren, daß jemand über viele Spiele hinweg je viermal würfelt. Dabei zählen Sie, wie oft mindestens eine Sechs auftritt. Wenn Sie ihr Ergebnis haben, dann erzähle ich Ihnen, wo der Pferdefuß in Rechenbergs Argumentation liegt. Denn auch auf theoretischem Weg, ganz ohne Computer, kann man die richtige Lösung herleiten.«

Wir verlassen an dieser Stelle die Gesprächsrunde, denn der Leser wird vermutlich selbst gerne über die aufgeworfenen Fragen nachdenken wollen. Bei der Computersimulation braucht man einen Zufallszahlengenerator, der Werte zwischen 1 und 6 liefert. Hat man z.B. in Pascal eine Funktion RANDOM, die zufällige Real-Zahlen zwischen 0 und 1 erzeugt, dann ergibt der Ausdruck (TRUNC(6*RANDOM)) + 1 die gewünschten Werte zwischen 1 und 6. In Turbo Pascal können derartige Werte einfach durch den Funktionsaufruf RANDOM (6) + 1 gewonnen werden. Für die theoretische Behandlung findet man Hinweise in Lehrbüchern der Wahrscheinlichkeitslehre unter dem Stichwort »bedingte Wahrscheinlichkeiten«, und häufig wird dort auch die Frage des Chevalier de Méré explizit behandelt.

Die Lösungen zu allen angeschnittenen Fragen finden sich im Anschluß.

Lösung:

Die Aufgabenstellung lautete: Jemand würfelt viermal. Wie groß ist die Wahrscheinlichkeit, daß dabei mindestens eine Sechs auftritt? Es wurde vermutet, daß diese Wahrscheinlichkeit 2/3 betrage, das ist aber nicht richtig. Im folgenden Pascal-Programm wird eine variable Anzahl von Würfelspielen simuliert und über die Zahl der gewonnenen und verlorenen Spiele genau Buch geführt. Am Ende werden die simulierten Werte ausgegeben und zusätzlich zur Illustration der Gewinnbetrag, der sich bei einem konstanten Einsatz von DM 60, ergeben hätte.

PROGRAM würfeln;

Wiederholtes Ausführen des Programms zeigt, daß zwar im Durchschnitt mehr Spiele gewonnen als verloren werden, daß die Wahrscheinlichkeit zu gewinnen aber nur geringfügig größer ist als 1/2. Ein typisches Ergebnis eines Programmlaufs bei z.B. 10.000 Spielen ist:

5221 Spiele gewonnen,
4779 Spiele verloren.

Die Annahme, daß 2/3 aller Spiele im Durchschnitt gewonnen werden, ist somit unhaltbar geworden.
Neben der praktischen Simulation dieses Problems durch ein Computerprogramm gibt es auch einen theoretischen Zugang. In der Argumentation des Herrn Rechenberg steckt ein Denkfehler. Die Wahrscheinlichkeit, in vier Würfen eine Sechs zu werfen, ist nicht 1/6 + 1/6 + 1/6 + 1/6. Eine solch einfache Addition von Wahrscheinlichkeiten ist hier nicht erlaubt, es widerspricht der Logik der Wahrscheinlichkeitslehre. (Das ist einfach einzusehen: angenommen, man darf siebenmal werfen. Wäre dann die Wahrscheinlichkeit 7/6? Nein, denn Wahrscheinlichkeiten, die größer als 1 sind, gibt es nicht!)
Rechenbergs Analyse hätte hier das Kalkül für sogenannte bedingte Wahrscheinlichkeiten erfordert. Wir gehen die korrekte Argumentation Schritt für Schritt durch.
Erstens: Wie groß ist die Wahrscheinlichkeit für Erfolg sofort im 1. Wurf? Antwort 1/6, klar.

Weiter: Wie groß ist die Wahrscheinlichkeit für einen Erfolg im 2. Wurf?
Antwort 1/6 * 5/6, denn Erfolg beim zweiten Anlauf setzt Mißerfolg im ersten Wurf voraus. Dessen Wahrscheinlichkeit ist 5/6, und beide Werte sind zu multiplizieren. Analog ergibt sich die Wahrscheinlichkeit für einen Erfolg im 3. Wurf zu 1/6 * 5/6 * 5/6.

Die Wahrscheinlichkeit für einen Erfolg im 4. Wurf ist 1/6 * 5/6 * 5/6 * 5/6.

Zusammen ergeben die errechneten Zahlen den Wert 671/1296 = 0,51775.

Dies ist die Gewinnwahrscheinlichkeit. Sie liegt näher bei 1/2 als bei 2/3, wie es ja auch die Computersimulation gezeigt hat. Es gibt übrigens auch einen vergleichsweise einfacheren Weg zur richtigen Lösung. Man fragt: »Wie wahrscheinlich ist es, in vier Würfen KEINE Sechs zu werfen?« (Mathematiker sagen, man berechnet das Gegenereignis.) Die Antwort ist, daß man viermal Pech haben muß, was jeweils mit der Wahrscheinlichkeit 5/6 der Fall ist. Die gefragte Wahrscheinlichkeit für das Gegenereignis ist also 5/6 * 5/6 * 5/6 * 5/6 = 625/1296.
Die eigentlich interessierende Wahrscheinlichkeit ist dann die Differenz zu 1. Man erhält in diesem Fall natürlich dasselbe Ergebnis wie oben: 0,51775.

Abschließend noch ein Ausflug ins Historische: Der erwähnte Chevalier de Méré stellte Blaise Pascal folgende Frage: Ist es wahrscheinlicher bei vier Würfen mit einem fairen Würfel (mindestens) eine Sechs zu werfen oder bei 24 Würfen mit je zwei Würfeln (mindestens) eine »Doppelsechs«, d.h. einen Sechserpasch.

Den ersten Teil der Frage haben wir schon beantwortet. Pascal fand (natürlich!) die richtige Antwort: die erste Wahrscheinlichkeit ist größer als die zweite für einen Sechserpasch, die etwas weniger als 1/2 beträgt.

Wer will, kann heutzutage etwas tun, was Blaise Pascal nicht konnte: auch das zweite Problem am Computer simulieren.

```
PROGRAM wuerfeln;

                .
( Es wird je viermal gewuerfelt, und geprueft, ob dabei eine Sechs
  auftritt. Wenn ja, wird der um den Faktor 5/3 vermehrte Einsatz
  ausgezahlt, bei DM 60 Einsatz wird z.B. DM 100 ausgezahlt. Wenn
  nein, ist der Einsatz verfallen.
  Wie oft gewinnt (oder verliert) man dabei auf lange Sicht?

  Programmiersprache: Turbo Pascal, Autor: W.J.Weber, Januar 1987   )

CONST gewinnzahl = 6;

VAR spielezahl,i,gewonnen,verloren,nettogewinn: REAL;

PROCEDURE Anfang;
  BEGIN
    CLRSCR;
    WRITELN('4 x Wuerfeln');
    WRITELN('-------------');
 END;

FUNCTION Wurf(n:INTEGER): INTEGER;
  BEGIN                              ( n ist nur Dummy-Parameter  )
    wurf := RANDOM(6) + 1;
                 ( Die Augenzahl muss zwischen 1 und 6 liegen )
  END;

BEGIN (* Hauptprogramm *)
  Anfang;
  WRITE('Wie oft soll gespielt werden? ');
  READ(spielezahl);
  WRITELN(' mal.');
  gewonnen := 0;
  verloren := 0;
  i := 0;                     ( Schleife wird durch Real-Variable )
  WHILE i < spielezahl DO     ( kontrolliert, um eine grosse Zahl  )
    BEGIN                     ( von Wiederholungen zu gestatten.  )
      i := i+1;
      IF gewinnzahl IN [wurf(1),wurf(2),wurf(3),wurf(4)]
        THEN gewonnen := gewonnen+1
        ELSE verloren := verloren+1
    END;
  nettogewinn := 100*gewonnen-60*spielezahl;
  GOTOXY(1,10);
  WRITE  ('Es gab ',gewonnen:6:1,' Gewinne');
  WRITELN(' und ',verloren:6:1,' Verluste.');
  WRITELN;
  WRITE('Bei konstantem Einsatz von DM 60 fuehrt das zu einem ');
  IF nettogewinn < 0
    THEN WRITELN('Gesamtverlust')
    ELSE WRITELN('Gesamtgewinn');
  WRITELN('von DM',ABS(nettogewinn):8:1,' .');
  WRITELN;
  IF nettogewinn <= 0
    THEN WRITELN('Traurig, traurig !')
    ELSE WRITELN('Hurra !')
END.
```


Mathematik für Feinschmecker

Heute: Telefonieren mit Hindernissen

Von Wolfgang J. Weber

Herr Rechenberg hat das Wagnis einer Auslandsreise auf sich genommen und befindet sich zur Zeit im Vereinigten Königreich Ihrer Majestät. England besitzt ja einen eigenartigen Ruf: vieles geht dort gemütlicher vonstatten als in Deutschland, dafür funktioniert aber auch nicht alles unbedingt auf Anhieb.
Dies erfährt auch gerade Rechenberg, der von seinem Telefonapparat im Hotelzimmer nach Hause telefonieren will. Die internationale Vorwahl der BRD ist 0049. Das klappt. Vorwahl Eschwege 5651 klappt auch noch. Doch die Teilnehmernummer 8702 geht nicht mehr.
Denkmann gibt nicht auf und ruft seinen Freund Witzig in Frankfurt an, der die Nummer 69-81932 hat. Das funktioniert. Er versucht es noch mit einer anderen Nummer 22627 und hat wieder kein Glück. Was ist hier los?
Einige Minuten Nachdenken bringen ihn auf den Gedanken, daß der Telefonapparat defekt sein könnte. Bei den funktionierenden Ziffernkombinationen treten alle Ziffern auf, außer der 7. Die Fehler treten bei Ziffernkombinationen auf, die eine 7 enthalten. Da es sich um ein Tastentelefon handelt, vermutet Denkmann messerscharf, daß die Taste 7 defekt sein könnte. Was tun? Beschwerde beim Manager, schließlich müßte der froh sein, daß ihm einer seiner Gäste Hinweise auf Mängel in seinem Hause gibt.
Der ist aber gar nicht froh, denn der bestellte Service lässt auf sich warten und Denkmann könnte schließlich noch 90% der ihn interessierenden Anschlüsse erreichen, da es zehn Tasten seien und doch nur eine davon defekt. Die Tatsache, 90% der Nummern wählen zu können, ist kein Grund ein größeres Theater zu veranstalten und so räumt Denkmann das Feld. Er legt sich schlafen und denkt noch ein wenig über die 90% nach. Da kommt ihm ein Zweifel ...
Auch den Lesern von PASCAL International sollten Zweifel gekommen sein. Wie viele sechsstellige, siebenstellige und achtstellige Ziffernkombinationen kann man mit dem defekten Telefon noch erzeugen? Gefragt ist wieder nach einem Computerprogramm einerseits und einer theoretischen Analyse andererseits.

Lösung:
Die Aufgabenstellung lautete: Zur Vefügung stehen alle Ziffern außer der "7". Wie viele sechsstellige, siebenstellige und achtstellige Ziffernkombinationen kann man mit diesen neun Ziffern erzeugen? Es wurde vermutet, daß noch jeweils 90% aller Kombinationen möglich sind, das ist aber nicht richtig.
Im folgenden Pascal-Programm werden alle gültigen siebenziffrigen Kombinationen durchgezählt, indem eine passende Zahl von FOR...TO-Schleifen ineinandergeschachtelt wird. Die Logik des Programms ist sehr einfach. Eine Anpassung an die Frage nach sechsstelligen bzw. achtstelligen Ziffernfolgen kann leicht vorgenommen werden.
Das Ausführen des Programms erfordert unangemessen viel Zeit. Hier liegen programmtechnisch noch einige Möglichkeiten zur Verbesserung. Das Programm zeigt aber deutlich, daß nicht etwa 90% der siebenziffrigen Folgen mit den funktionierenden 9 Zifferntasten erzeugt werden können, sondern noch nicht einmal die Hälfte, genau 47,82969%.
Von allen sechstelligen Ziffernfolgen können noch 53,1441% und von den achstelligen nur magere 43,046721% erzeugt werden.
Neben dem direkten Abzählen durch ein Computerprogramm gibt es auch eine theoretische Lösung. Rechenberg sollte hier das mathematische Teilgebiet der Kombinatorik benutzen. Wir wollen uns schrittweise das Prinzip verdeutlichen. Mit den zur Verfügung stehenden neun Ziffern können erzeugt werden:

9 einstellige Folgen (90% aller Folgen),

9 x 9 = 81 zweistellige Folgen (81% aller Folgen),

9 x 9 x 9 = 729 dreistellige Folgen (72.9% aller Folgen),

9 x 9 x 9 x 9 x 9 x 9 x 9 siebenstellige Folgen.

Insgesamt gibt es 10000000 (10^7 oder 1E7) verschiedene siebenstellige Ziffernfolgen. Das Produkt 9999999 hat den Wert 4782969, was etwa 47,8% der insgesamt möglichen Ziffernfolgen entspricht.
Ganz entsprechend kann der Fall der achtstelligen Ziffernfolgen erledigt werden:
$9^8 . 10^8$ = 43046721/10000000 = 43,0%.
Alles in allem kann Rechenberg nur ungefähr die Hälfte seiner Telefonate führen, nicht etwa 90%. (Dies ist natürlich eine statistische Aussage, die für große Zahlen recht genau, nicht aber den Einzelfall gilt.)
Auch an diesem Beispiel sieht man wieder, daß Computerpower nicht immer schneller zum Erfolg führt als Brainpower. Im sinnvollen Verhältnis der beiden zueinander liegt die Würze und die Kürze!

```
PROGRAM Ziffernkombinationen;

(Es werden alle siebenstelligen Ziffernfolgen darauf untersucht,
ob sie keine Ziffer "7" enthalten.
Der Prozentsatz der so gefundenen Ziffernfolgen wird ausgegeben.

Programmiersprache: Turbo Pascal, Autor: W.J.Weber, Marz 1987)

VAR funktioniert: Real;     ( ist erforderlich weil u.U.
                              MaxInt überschritten wird )
    L1,L2,L3,L4,L5,L6,L7: integer;     ( Laufvariablen )

BEGIN
  funktioniert:=0;
  FOR L1:=0 TO 9 DO
    IF L1<>7 THEN
    BEGIN
      Write('.');                     ( Programm lebt noch )
      FOR L2:=0 TO 9 DO
        IF L2<>7 THEN
        FOR L3:=0 TO 9 DO
          IF L3<>7 THEN
          FOR L4:=0 TO 9 DO
            IF L4<>7 THEN
            FOR L5:=0 TO 9 DO
              IF L5<>7 THEN
              FOR L6:=0 TO 9 DO
                IF L6<>7 THEN
                  FOR L7:=0 TO 9 DO
                    IF L7<>7 THEN
                    funktioniert:=funktioniert+1
      END;
    WriteLn;
    Write  (100*funktioniert/1E7:6:1);
    Write  ('% aller 7-stelligen Ziffernfolgen ');
    WriteLn('konnen erzeugt werden.')
 END.
```

Nun können im Dualsystem a_n, b_n und u_{n-1} in 8 verschiedenen Kombinationen auftreten und beeinflussen mit diesen auch u_n und z_n in verschiedener Weise. Diese 8 Kombinationen und deren Einfluß sozusagen als Funktion für u_n und z_n sind in der nachfolgenden sogenannten **Boole'schen Wahrheitstafel** dargestellt:

a_n:	O	O	O	O	L	L	L	L
b_n:	O	O	L	L	O	O	L	L
u_{n-1}:	O	L	O	L	O	L	O	L
u_n:	O	O	O	L	O	L	L	L
z_n:	O	L	L	O	L	O	O	L

u_n wird also unter folgenden Bedingungen gesetzt:

(NOT a_n & b_n & u_{n-1}) |
(a_n & NOT b_n & u_{n-1}) |
(a_n & b_n & NOT u_{n-1}) |
(a_n & b_n & u_{n-1}).

Der erste Ausdruck in Klammern stellt die vierte Spalte der Wahrheitstafel dar, der zweite Ausdruck in Klammern die sechste Spalte, der dritte Ausdruck die siebente Spalte und der vierte Ausdruck die letzte Spalte. Diesen langen logischen Ausdruck kann man zusammenfassen:

u_n = a_n & b_n |
(u_{n-1} & ((NOT a_n & b_n) |
(a_n & NOT b_n)).

Desgleichen erhält man für z_n:

(NOT a_n & NOT b_n & u_{n-1}) |
(NOT a_n & b_n & NOT u_{n-1}) |
(a_n & NOT b_n & NOT u_{n-1}) |
(a_n & b_n & u_{n-1})
= (NOT a_n & ((NOT b_n & u_{n-1}) |
(b_n & NOT u_{n-1})) |
(a_n & (NOT (b_n | u_{n-1}) |
(b_n & u_{n-1})).

Wenn Sie nun bedenken, daß in den beiden oben berechneten Ausdrücken die einzelnen Buchstaben nichts weiter bedeuten als elektrische Signale, und die Operationszeichen nichts weiter als Schaltungen, die die Eingangssignale steuern und verarbeiten, so daß die endgültigen Ausgangssignale als Ergebnis anzusehen sind, dann wissen Sie jetzt, wie Rechner rechnen – im Prinzip wenigstens. Abbildung 4 zeigt nochmals das Schema des Addiervorgangs im Rechner anhand der in Abbildung 3 dargestellten logischen Grundbausteine. Es liegt nun an Ihnen, sich vorzustellen (angenommen, Sie lassen von ihrem Computer eine Multiplikation durchführen), welch komplizierte Wege die Signale gehen, die man über die Tastatur eingibt und die dann Sekundenbruchteile später bearbeitet auf dem Bildschirm flimmern.

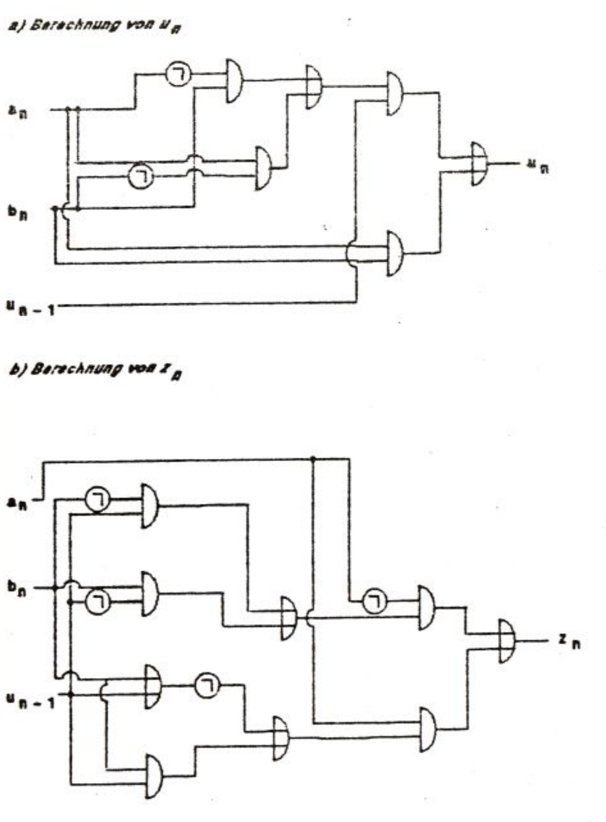
Leider bin ich in dieser Folge nicht mehr dazu gekommen, die Fehler der Rechner-Rechnerei wie angekündigt zu behandeln, doch hielt ich es für wichtiger, einmal einen etwas längeren Exkurs in die Bereiche der Mathematik zu bringen, da doch die Funktionsweise eines Computers wie gesehen (bzw. gelesen) auf der wohl logischsten der Naturwissenschaften basiert.

Freuen Sie sich auf ein weiteres Kapitel aus der Mathematik im nächsten Heft von PASCAL. Dann nämlich kommen wirklich die Fehler dran und die Art und Weise, wie ein Computer schwierigere Berechnungen durchführt. Wir verlassen dann die Boole'sche Algebra und gehen über zur Analysis mit ihren Funktionen, Differentialen, Folgen und Reihen.

Die nächste Folge wird dann der Abschluß dieses kleinen Kurses sein und ich hoffe, Sie können dann denen erklären, die Sie fragen, wie Rechner rechnen, wie Rechner rechnen (logisch, nicht wahr?).

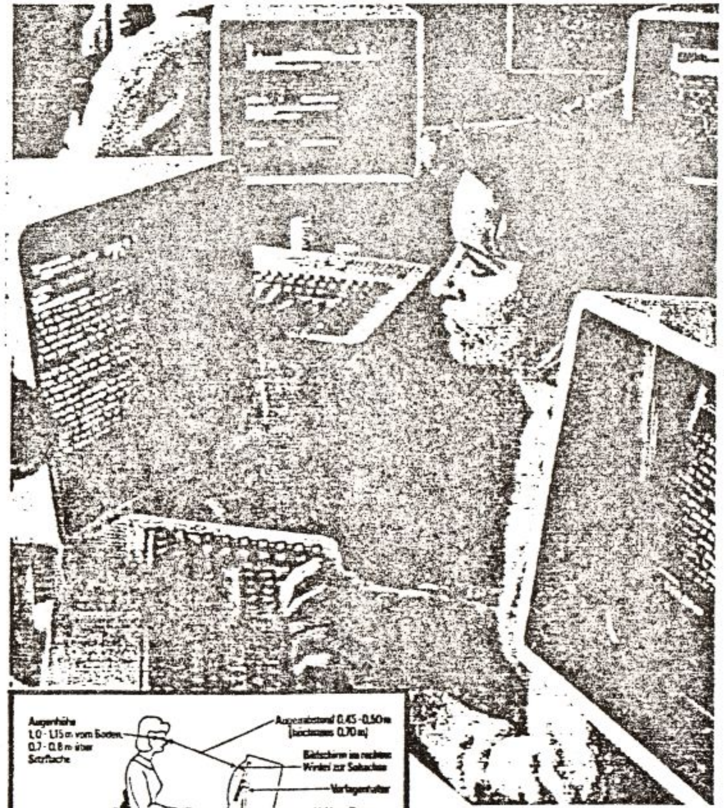
(Jörg Gondermann)

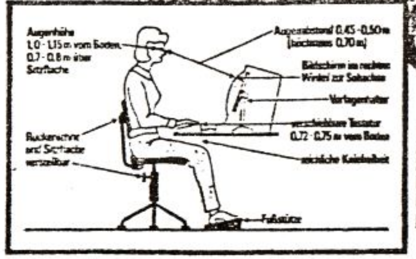
Abbildung 4: Additionsschaltung



Ich arbeite am Computer Bin ich strahlengefährdet?

Wissenschaftler haben jetzt Computer getestet. Ergebnis: Die Arbeit vor dem Bildschirm soll mit ganz bestimmten Risiken verbunden sein





Eine Datentypistin muß täglich bis zu 30 000mal mit den Augen zwischen Schreibvorlage, Tastatur und Bildschirm hin- und herwechseln. Die Zeichnung links zeigt Ihnen, wie der optimale Computer-Arbeitsplatz aussehen sollte

Klagen darüber, daß die Arbeit am Computer krank macht, gibt es schon, seit die ersten Geräte 1968 bei uns eingeführt wurden. Jetzt soll es auch Beweise dafür geben! Drei Bremer Physiker haben 16 der meistverkauften Personal- und Homecomputer untersucht und herausgefunden, daß keiner der Apparate strahlungsfrei ist. Alle Geräte sind von sogenannten elektrischen Feldern umgeben.

Unser zentrales Nervensystem reagiert darauf mit Augenflimmern und Kopfschmerzen. Langfristig kann es zu Herzbeschwerden, Kurzsichtigkeit und Atemnot kommen, bei schwangeren Frauen können Fehlgeburt oder Mißbildungen nicht ausgeschlossen werden.

HÖRZU sprach mit Dr. Ute Boikat vom Bremer Forschungs-Team.

Die meisten sitzen zu nahe vor dem Bildschirm

Dr. Ute Boikat (35), Physikerin bei der Gesundheitsbehörde

HÖRZU: Wodurch wird die Strahlung verursacht?

Dr. Boikat: Eigentlich sind es keine Strahlen, sondern elektrische Felder. Sie werden durch die Bildröhre ausgelöst, die unter Hochspannung (etwa 15 000 Volt) arbeitet. Die Felder beeinflussen unser zentrales Nervensystem. Hinzu kommen Nervenreizungen durch Magnetfelder, die von den Spulen in den Geräten erzeugt werden und durch statische Felder, die dadurch entstehen, daß ständig Elektronen von innen auf die Computerscheibe geschleudert werden. Diese Felder laden den Computer oft so stark auf, daß wir leichte Stromschläge kriegen, wenn wir sie berühren.

HÖRZU: Welche Felder sind am gefährlichsten?

Dr. Boikat: Unter Fachleuten ist umstritten, ob elektrische Felder mehr auf den Menschen einwirken als magnetische oder statische. Schwedische Wissenschaftler halten die Magnetfelder für besonders gefährlich, die Kanadier die elektrischen. Konkrete Beweise hat keiner.

HÖRZU: Eigentlich müßten die artverwandten Fernsehstrahlen dann genauso gefährlich sein wie Computerstrahlen?

Dr. Boikat: Sind sie auch. Wir haben in unseren Untersuchungen einen Farbfernseher einbezogen. Er hat die stärksten Magnetfelder erzeugt. Die Gefahr wird allerdings dadurch stark reduziert, daß man viel weiter vom Fernseher entfernt sitzt als vom Computer. Wer etwa drei Meter weg ist, bekommt kaum etwas ab. Die typische Arbeitsentfernung zum Computer-Bildschirm beträgt jedoch nur 30 Zentimeter.

HÖRZU: Sind die gesundheitlichen Schäden nur Vermutungen, oder gibt es auch wissenschaftliche Beweise?

Dr. Boikat: Die gibt es. Es ist allerdings nicht erwiesen, daß die von uns untersuchten Felder allein krank machen. Oft kommt Streß hinzu, ein falsch angelegter Computer-Arbeitsplatz – also die richtige Anordnung des Lichteinfalls auf den Bildschirm –, flimmernde Bilder...

HÖRZU: Sind Menschen am Computer auch krebsgefährdet?

Dr. Boikat: Mediziner behaupten das. Eindeutige Beweise dafür fehlen allerdings noch.

HÖRZU: Wer ist häufiger von Computer-Krankheiten betroffen, Frauen oder Männer?

Dr. Boikat: Eindeutig die Frauen. Weil mehr von ihnen an den Geräten arbeiten, aber auch, weil ihr Organismus, insbesondere während einer Schwangerschaft, feinfühliger auf elektrische Felder reagiert.

HÖRZU: Und die Kinder? Wie gesundheitsgefährdend sind ihre Spielcomputer?

Dr. Boikat: Genauso gefährlich wie Arbeitscomputer. Die Billig-Geräte mit viel Plastik schirmen die Felder am wenigsten ab. Kinder sollten auf keinen Fall länger als eine Stunde täglich vor dem Bildschirm sitzen.

HÖRZU: Gibt es keine ungefährlichen Computer?

Dr. Boikat: Doch, meist sind Geräte mit Metallgehäuse besser. Aber sie werden weniger benutzt, weil sie bis zu 2500 Mark teurer sind.

Professor J. Bernhardt (48) vom Bundesgesundheitsamt

„Nicht die Bildschirme sind das Problem, sondern die Bildschirm-Arbeitsplätze, das beweisen über 400 Untersuchungen auf der ganzen Welt. Ein Beispiel von vielen: In Skandinavien erkrankten vier Frauen an Gesichtsausschlägen. Ihre Computer hatten sich durch extreme Trockenheit in den Räumen statisch aufgeladen und verschleuderten Staubpartikel. Durch mehr Luftfeuchtigkeit und Erdung der Computer war das Problem sofort beseitigt.

Trotzdem bin ich dafür, daß bei uns endlich gesetzliche Grenzwerte für Computereinwirkungen festgelegt werden."

Computer-Arbeitsplatz: Was Sie dazu wissen sollten

Wichtig vorm Bildschirm: korrekte Haltung, richtiger Abstand

Man sollte nicht länger als vier Stunden täglich vor dem Bildschirm sitzen, pro Stunde zehn Minuten Pause einlegen. Gegen die statischen Felder schützt ein Abschirmgitter aus feinem Metallgewebe. Es wird an einem Kabel geerdet. Die feinen Staubpartikel lagern sich nicht mehr auf Gesicht oder Schirm ab, sondern bleiben in der Luft hinter dem Gitter. Die Gaze-Filter kosten 160 bis 300 Mark. Vor elektrischen und magnetischen Feldern schützen Metallgehäuse. Ersatzweise kann man den Monitor auch mit Haushalts-Alufolie einwickeln.

HEFT 20 August 1987

CLUB 80 — Bücherei

Im Heft 12, erschienen im März 1986, also vor mehr als einem Jahr, erschien zum letztenmal eine Bücherliste des CLUB 80. Seither ist im Club einiges geschehen, aber bei den Büchern hat sich nicht viel getan!

Leider sind einige der Publikationen, wie schon mehrfach im Info beklagt, bei einer unkontrollierten Entnahmeaktion während des 86'er Clubtreffens in Holzhausen abhanden gekommen. Dafür wurde aber die Bibliothek wieder von anderer Seite (Peter Spieß, Bernd Retzlaff, Gerald Dreyer) mit neuen Büchern aufgefüllt.

Viel zu wünschen übrig, ließ bis jetzt nur das Interesse der Mitglieder an dem Angebot, sich die Bücher beim Club auszuleihen, statt sich die manchmal doch recht teuren Veröffentlichungen selbst zu kaufen. So liegen die Bücher größtenteils bei mir im Keller herum und warten auf Leiher und Leser!

Ein Grund für die nur sehr schwache Nachfrage ist ohne Zweifel der oben angesprochene Umstand, daß seit über einem Jahr keine aktuelle Liste mehr erschienen ist. Dies soll sich nun ändern! Die folgend abgedruckte Liste ist auf dem neuesten Stand und enthält alle zur Zeit verfügbaren Bücher. Bei Anforderungen, die sich in nächster Zeit hoffentlich häufen, ist natürlich Paragraph 8 Absatz 2 der Clubsatzung zu beachten, der kurz gesagt folgendes festlegt:

1. Der Besteller hat die Kosten für den Versand in beide Richtungen zu tragen (bei Bestellungen 3,-- DM in Briefmarken beilegen)!

2. Die Ausleihzeit beträgt vier Wochen (auf Antrag auch länger, wenn das Buch nicht von jemand anderem angefordert wurde)! Wird die Ausleihfrist überschritten erfolgt eine kostenpflichtige Mahnung (5,- DM, für jede weitere Mahnung ebenfalls 5,- DM).

3. Verlorengegangene Bücher müssen ersetzt werden!

4. Weitergabe an Dritte (insbesondere Nichtmitglieder) ist untersagt und kann zum Ausschluß aus dem Club führen!

Ich hoffe, daß das doch recht gute Angebot bald mehr genutzt wird und damit ein zur Zeit brachliegender Teil der Serviceleistungen des CLUB 80 wieder reaktiviert wird!

Hartmut Obermann

Nr. 0002: Computerwissen

Michael Scharfenberger --- Markt & Technik

Tips fuer die Auswahl und Beschreibung von Anwendungsmoeglichkeiten von Hard- und Software sowie Erklaerung von mehr als 500 Begriffen.

Nr. 0003: Computerspiele und Knobeleien programm. in BASIC

Ruedeger Baumann --- Vogel-Buchverlag (CHIP-Wissen)

Von der Spielidee ueber die Spielstrategie kommt es zum Programm selbst. Keine Sammlung von Spielkonserven - keine Programmierkenntn. erford.

Nr. 0004: Mein Home-Computer - Eine Verbraucherfibel

- --- Vogel-Verlag (HC-Leserservice)

Die besten Tips fuer Kauf und Anwendung von Home-Computern.

Nr. 0005: Programmieren mit dem ZX81 in Basic u. Masch.-code

E. Floegel --- Hofacker, Holzkirchen

Sammlung von Spiel-, Schul- und anderen Programmen sowie einem Kapitel ueber die Programmierung des Prozessors Z80 (gute Programme dabei)

Nr. 0006: Games For Your TRS-80

Chris Palmer --- Virgin Books (Great Britain)

Sammlung von 20 Basic-Spiel-Programmen und einer Anleitung, wie man bessere Programme schreibt.

Nr. 0008: More Basic Computer Games

David H. Ahl --- Creative Computing Press, USA

84 Spiele fuer den TRS-80, wobei einige sehr interessante dabei sind. Das Buch ist fuer Freunde von Basic-Computer-Spielen nur zu empfehlen.

Nr. 0009: BASIC: Dateien, Listen und Verzeichnisse

Busch Rudolf --- Franzis-Verlag GmbH, München

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0010: BASIC: Matrix-Operationen

Busch Rudolf --- Franzis-Verlag GmbH, München

Eine Software-Sammlung mit vielen nuetzlichen Programmen in Kursform (also mit Lern-Effekt).

Nr. 0011: TRS-80 PROGRAMS

Tom Rugg und Phil Feldman --- Dilithium Press, Beaverton,USA

32 BASIC-Programme (Erziehung, Anwendung, Spiele, Graphic, Mathematik und Verschiedenes) fuer Level II.

Nr. 0012:   Programme und Tricks fuer Genie I und Genie II

Clemens Becher, Franz Seger   --- ?

Viele Programme, Tips und Tricks fuer den Genie.

Nr. 0013:   BASIC: Alles ueber PEEK und POKE

Heiko Requardt   --- Franzis-Verlag

Eine Software-Sammlung in BASIC (mit vielen guten Tips und Tricks fuer de n 'Amateur').

Nr. 0014:   TRS-80 und Video Genie ROM-Listing fuer Level II

Luidger Roeckrath   --- ?

ROM-Listing, RAM-Adressen, I/O-Adressen, Unterprogramme, Basic-Anweisunge n und Funktionen, Aufzeichnungsformate auf Cassette, ...

Nr. 0016:   Das Modem-Sonderheft (Datenübertr. mit Mikrocomp.)

verschiedene   --- MC, Franzis-Verlag, München

Grundlagen, Schaltungstechnik, DFü-Programme

Nr. 0017:   GENIE DATA (Heft 1/83 bis Heft 3/85)

--- Heinz Hueben Verlag

Eine Sammlung von 13 Computerheften.

Nr. 0020:   The Custom TRS-80 & Other Mysteries

Dennis Bathory Kitsz   --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das sich mit Soft- und Hardware für uns eren Computer beschäftigt.

Nr. 0022:   TRSDOS 2.3 Decoded & Other Mysteries

James Farvour   --- IJG Inc., 1953 West, USA

Ein kommentiertes TRSDOS 2.3 in englischer Sprache.

Nr. 0023:   TRS-80 Disk & andere Geheimnisse

H.C. Pennington (G. Daubach)   --- IJG Inc., 1953 West, USA

Das ins Deutsche übersetzte Buch beschreibt die Diskettenorg., die Zuordn ung freier Speicherbereiche, die Behebung v. Diskettenfehlern etc.

Nr. 0024:   TRS-80 Data File Programming (Model I/III)

Finkel & Brown   ---

Ein Lehrbuch, wie man in BASIC Datenfiles programiert und benutzt. Mit se hr vielen Beispielen (englisch-sprachig).

Nr. 0025:   Programming Techniques for Level II BASIC

William Barden   ---

Ein gutes Buch über schnelle Graphiken, Suchroutinen und Sortierroutinen im Level II BASIC (englisch-sprachig).

Nr. 0026:   Advanced Level II BASIC

Don Imman, Albrecht, Zamora   ---

Ein gutes Lehrbuch mit vielen Beispielen zu allen Themen der Computerei (englisch-sprachig).

Nr. 0027:   Level II BASIC

Don Imman, Albrecht, Zamora   ---

Ein gutes Lehrbuch für den Anfänger in BASIC (englisch-sprachig).

Nr. 0028:   Programming In Style

Th. Dwyer, Margot Critchfield   ---

Einführung in BASIC, strukturiertes Programmieren, Tips zur besseren Aufm achung der eigenen Programme, Tips für den Profi (englisch-sprachig)

Nr. 0029:   The First Book Of 80-US

verschiedene   --- Hofacker Verlag, Muenchen

Sammelband aus dem TRS-80 Users Journal.

Nr. 0030:   The Rest Of 80

verschiedene   --- Wayne Green Books, USA

Viele Beiträge aus der 80-Micro (Allgemeines, Graphik, Utilities).

Nr. 0031:   Z-80-Applikationsbuch

Michael Klein   --- Franzis Verlag, München

Eine Einfuehrung in die Programmier- und Interfacetechniken des Mikroproz essors Z 80.

Nr. 0032:   Z 80 Assembler Handbuch

--- Hofacker, München

Die Z 80 Befehle mit Kommentar und je einem Beispiel.

Nr. 0033:   Programmierung des Z 80

Rodnay Zaks (Bernd Ploss)   --- Sybex-Verlag, Duesseldorf

Als Lehr- und Lernmittel konzipiert ist dieses ins Deutsche übersetzte Bu ch ein umfassendes Nachschlagewerk und eine gruendliche Einfuehrung.

HEFT
20
August
1987

Nr. 0034: Daten - Disketten - Dateien

Hans-Joachim Sacht --- CHIP, Vogel-Verlag, Wuerzburg

Anregungen und Hinweise, wie Programme mit komfortabler Dateiverw. geschrieben werden koennen und was bei den versch. DOS zu beachten ist.

Nr. 0035: Anwenderprogramme fuer TRS-80 und Video Genie

M. Stuebs --- Hofacker, Muenchen

Viele (meist einfache) BASIC-Programme fuer den TRS-80.

Nr. 0036: Programmieren mit TRS-80

M. Stuebs --- Hofacker, Muenchen

Eine Einfuehrung in den TRS-80 und viele (meist einfache) BASIC-Programme.

Nr. 0038: Textsystem in BASIC und Z80-Assembler

Gert Scheller --- Luther-Verlag, Sprendlingen

Das Buch beschreibt ein fertiges Textsystem fuer den TRS-80, das in BASIC und Z80 modular aufgebaut ist, und leicht erweiterbar ist.

Nr. 0039: Textsystem in BASIC und Z80-Assembler - Band 2

Wolfgang Weber --- Luther-Verlag, Sprendlingen

Der Band 2 beschreibt Erweiterungen zum Programm aus Band 1.

Nr. 40 : Einfache Perpheriegeräte im Selbstbau

Owen Bishop --- Birkhäuser

Verbinden Sie Ihren Microcomputer mit seiner Umwelt! Einfache und nachbausichere Schaltungen zur Erweiterung des Computerhorizonts!

Nr. 41 : Start in die Computergrafik

Fred Wagenknecht --- Vogel-Verlag

Grundlagen und Programme der Computergrafik für TRS 80, VideoGenie und ColorGenie.

Nr. 42 : Level II BASIC Reference Manual

--- Radio Shack

Dieses Buch wird normalerweise von Tandy zum Model I dazugegeben. Für Genie-Besitzer ev. interessant!

Nr. 43 : BASIC-Code

Wiegand/Fillinger --- Ravensburger

Die kompatible Computersprache für über 25 Homecomputer. Mit 14 Programmlistings und Kassette. Natürlich auch für TRS 80 Model 1!!!

Nr. 44 : Kleiner BASIC-Wortschatz

Busch --- Franzis-Verlag

Die wichtigsten BASIC-Begriffe einfach erklärt und gelistet!

Nr. 45 : BASIC Computer Games

David Ahl --- Creative Computing Press

101 Computerspiele für Homecomputer!

- IMPRESSUM -

1. Vorsitzende

Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

2. Vorsitzende

Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

Hardwarekoordinator

Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar

Werner FÖRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion

Jens NEUEDER
Panoramastraße 21
7170 Michelbach /Bilz
☎ 0791 /42877

Autoren

Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck

Peter Spieß
Trugenhoferstraße 27
8859 Rennertshofen 1
☎ 08434 /454

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Bankverbindung des CLUB 80

Postgirokonto Peter STEVENS
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

SCHLUSS

Hallo Club-80er,

trotz Sommerloch, Dauernebenbeschäftigung als "Häuselsbauer" und Dank Eurer Mitarbeit konnte ich Euch ein dickes INFO fabrizieren.
In der Ferienzeit habt Ihr nun sicher auch einmal die Gelegenheit es ausgibig und in Ruhe zu lesen. Es ist ja auch diesmal wieder recht umfangreich und interessant geworden. Nicht zuletzt dadurch, daß wir diesmal einen "clubfremden" Schreiber dabei haben. Ihm möchte ich auf diesem Wege meinen besonderen Dank für seine Mitarbeit an unserem Clubinfo aussprechen.

Als nächstes möchte ich nun einen Punkt in unserem Clubleben ansprechen, bei dem Ihr mich sehr enttäuscht habt.

<u>Die Entscheidung des Zehnzeilerwettbewerbes !!</u>

Ganze 2 (in Worten: zwei) Clubmitglieder hatten sich zur telefonischen Abstimmung gemeldet. -- Waren die anderen alle im Urlaub ??
Ich finde, das daß kein gutes Zeichen für eine Interessengemeinschaft, wie wir es eine sind, ist, sich am Clubleben nicht zu beteiligen. Immerhin haben sich die Programmschreiber -auch für Euch- bemüht etwas auf die Beine zu stellen. über eine Reaktion von Eurer Seite aus hätten sie sich sicher gefreut!
Da wir in nächster Zeit ein Vorstadstreffen haben, werden wir Vorstandsintern eine Entscheidung über eine Belohnung der Wettbewerbs-Programmierer treffen.
<u>Ich finde Eure Art der aktiven clubinternen Mitarbeit echt mangelhaft.</u> Leichter und einfacher hättet Ihr eine Abstimmung (Höhrer abheben, wählen und Eure Stimme abgeben) wirklich nicht mehr haben können.

Aber nichts desto Troz wünsche ich Euch viel Freude mit Eurer neuesten INFO.

Jens

Alle waren voll!

-zählig erschienen! Wo!? Natürlich beim Vorstandstreffen vom 14.-16.08. beim 1. Vorsitzenden Hartmut Obermann. Dies war bemerkenswerterweise das erste Vorstandstreffen, bei dem wirklich der ganze Vorstand beisammen war. Außerdem konnten wir uns noch über mehrere Gäste freuen (Ulrich Böckling, Gerald Dreyer und Rüdiger Sörensen).

Während des Treffens konnten wir viele brennendheiße Themen abhandeln. Hier nun eine kurze Zusammenfassung (die Ergebnisse werden in einzelnen Artikeln erläutert):

- Schaffung einer MS-DOS-Gruppe bzw. -Ecke
- Anschaffung von
  - Reinigungsset für Diskettenlaufwerke
  - Büchern über CP/M, Sprachen usw.
  - neuer PD-Software aus den USA
- Einrichtung einer CP/M-Programmbibliothek
- Aktualisierung des Grafik-Sonderheftes
- Auflage weiterer Sonderhefte
- Auflage eines Sammelbandes der Clubinfos für Neuzugänge
- mehr Werbung für den Club durch
  - Beilagen zu Firmen-Werbesendungen
  - Plakate auf Messen, Ausstellungen und Flohmärkten (Neugestaltung eines Werbe-Plakates)
  - Mundpropaganda der Club-Mitglieder
- Termin- und Ortsplanung für das Clubtreffen 1988 (vom 12.5. bis 15.5., Raum Frankfurt-Darmstadt)
- allgemein Administratives.

Nach diesem strapaziösen Wochenende mit vielen ungesunden Sachen wie schwarzem Kaffee und wenig Schlaf wird der Vorstand eine Kur antreten müssen. Da im Moment die Clubfinanzen dies zulassen, wird die Vorstandsadresse auf unbestimmte Zeit nach Florida verlegt.

Bis zu unserer Rückkehr fröhliches Hacken wünscht

Der gesamte Vorstand.

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Mistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907	// 0431 /74047
Betz	Heinrich	St. Wolfgangsstraße 13	8551	Hausen	09191 /31698	// 09191 /611108
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000	Frankfurt 1	069 /5601621	// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441	Mettringen	05233 /4320	// 02557 /1236
Emmerich	Helmut	Waldstraße 5	6682	Ottweiler	06824 /4114	// -
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721 /21841	// 09721 /51256
Gromottka	Uwe	Lange Reihe 40	2803	Weyhe	0421 / 80496	// 0421 /300-6870
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024	//
Hill	Peter	Bergstraße 65	6754	Otterberg	-	// -
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948 /542	// 02921 /70431
Krispin	Michael	Schwanstraße 8	4130	Moers 3	02841 /73690	// -
Krüger	Karl-Herbert	Bruchweg 65	4920	Lemgo	05261 /13686	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417	// 07022 /77474
Lose	Gerhard	Viefhaushof 42	4306	Essen 13	0201 /212608	//
Mand	Harald	Kleinflintbeker Straße 7	2302	Flintbek bei Kiel	04347 /3629	// 0431 /3013580
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083	// 04151 /8891-37
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913	// -
Obscherningkat	Helmut	1 RUE DES BRUYERES	F-68360	Soultz	0033089/762690	// -
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /43791	//
Reichelt	Dieter	Philipp-Schmitt-Straße 30	6902	Sandhausen	06224 /52906	// -
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551	// 04103 /605310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383	// 0203 /331383
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221 /[illegible]	// -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631	// 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228 /643853	// -
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434 /454	// 08431 /7041604
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585	// 05401 /30096
Stevens	Peter	Postfach 56	4600	Dortmund 1	0231 /593883	// 0231 /593883
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153 /1564	// -
Sörensen	Rüdiger	Thomas-Mann-Straße 34	6500	Mainz 1	06131 /32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285	//
Wacker	Fred	Postfach 2246	7550	Rasstatt	07222 /52574	// -
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361	// -
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686	// -

Stand: Juli 1987

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

------ S U P E R D O S -----

Ein Erfahrungsbericht von Hartmut Obermann:

Vor kurzem habe ich unserem Freund Günther Wagner mitgeteilt, daß ich in meiner Programmbibliothek ein SUPERDOS genanntes Diskettenbetriebssystem habe. Daraufhin bat er mich, einen kurzen Erfahrungsbericht für unser Clubinfo zu verfassen. Dieser Bitte komme ich hiermit nach.

SUPERDOS ist ein Diskettenbetriebssystem von J. Wulff. Bei der ersten Inbetriebnahme fällt sofort die Ähnlichkeit mit NEWDOS80 auf. Im Endeffekt ist das eigentliche DOS auch nichts anderes als das bekannte NEWDOS, jedoch mit deutschen Meldungen. Steht bei NEWDOS z.B. "Device not available", wenn man einen nicht vorhandenen Diskdrive anspricht, so bringt SUPERDOS die Fehlermeldung "Bauteil nicht erreichbar". Die deutschen Fehlermeldungen erstrecken sich auch auf das Diskbasic von SUPERDOS. Diese Art der Fehlermeldung erscheinem einem geübten NEWDOS-Benutzer im ersten Augenblick sehr ungewohnt,(z.B.:Prüfzahlfehler beim Lesen) erweisen sich aber für einen Computerlaien (z.B. meine Frau) als durchaus nützlich.

Dies vor allem, da auch die sog. START-File, welche durch das Kommando AUTO LIST START bei jedem Neuinitialisieren auf dem Bildschirm erscheinen, in Deutsch geschrieben sind. Diese Files sind nichts anderes als eine Auflistung der im SUPERDOS untergebrachten Utilitys. Und hilft selbst dem unerfahrensten Operator beim Aufruf der benötigten Hilfsprogramme.

Als solche sind im SUPERDOS eingebaut:

1. Ein "Adapter Tape/Disk Disk/Tape" der sehr dem bekannten LMOFFSET ähnelt, jedoch zusätzlich die Möglichkeit enthält, die Adressen der zu übertragenden Programme auf dem Drucker auszugeben. Ein weiterer Vorteil besteht darin, daß nicht nur Start-, End- und Einsprungadresse sondern zusätzlich die Blockladeadressen angezeigt werden.

2. Ein Bildschirm-Editor, der es erlaubt, ähnlich wie z.B. auf Commodore-Computern Basicprogramme bildschirmorientiert zu editieren. Der Umgang mit diesem Modul will jedoch gelernt sein. Allzu schnell kann man, an den Zeileneditor gewohnt, sein komplettes Programm verhunzen. Der normale Zeileneditor bleibt übrigens erhalten und kann weiterhin benutzt werden.

3. Eine Chain-File-Erstellungshilfe, wie es sie ähnlich schon mehrfach gibt (z.B.:JOB/CMD oder CHAINBLD/CMD). Eine Besonderheit enthält dieses Programm allerdings. Aber leider keine positive. Wenn der Chainfile gespeichert werden soll, ist die Angabe eines Ziel-Drives nicht möglich und das Programm wird irgendwo (ich arbeite mit 3 Drives) auf den nächsten freien Speicherplatz geschrieben.

4. Ein File-Retter, der, bis auf die deutschen Meldungen natürlich, dem Programm RESCUE/CMD entspricht das vor kurzem in der CHIP abgedruckt war. Mir ist übrigens die Funktion dieses Utility-Programms immer noch nicht geheuer. Bei Versuchen damit ist mir aufgefallen, daß sich bei einem Aufruf des Programms direkt nach dem Löschen der Files diese nicht mehr wiederherstellen liesen, obwohl noch kein Zugriff auf die entsprechende Diskette erfolgt war.

5. Ein Dircheck-Programm, welches auch unter NEWDOS80 Verwendung findet.

6. Ein Disketten-Katalog, bekannt unter dem Namen INDEX/CMD. Dieses Programm ist sehr nützlich, da man sich in kürzester Zeit einen guten Überblick über seine Disketten verschaffen kann. Auch dieses Programm ist natürlich mit deutscher Anleitung und entsprechenden Fehlermeldungen ausgerüstet.

7. Ein MINI-SUPERZAP genanntes Programm, welches in Anlehnung an die großen ZAP-Programme die Anzeige des Inhalts eines Files und deren Änderung ermoeglicht.

8. Schon erheblich nützlicher ist der ebenfalls eingebaute Printer-Buffer. Dieses Programm reserviert einen festlegbaren Teil des RAM als Buffer für die Printerausgabe. Programme mit häufiger Druckerausgabe können so erhebich schneller ablaufen.

9. Ein sog. Supercopierer, mit dem man (fast) alle Disketten kopieren kann. Ich selbst habe damit schon TI 99/4A- und Commdore-Disketten erfolgreich (d.h. lauffaehig) kopiert. Die Sache hat jedoch den Haken, daß man zwei Drives benötigt. Im übrigen ist auch dieses Programm bekannt (COPYCAT/CMD).

10. Ein weiteres, sehr empfehlenswertes Hilfsmittel ist das eingebaute Tasten-Shorthand-Programm. Es belegt alle Tasten mit BASIC-Befehlen, welche mit der Tastenfolge "CLEAR" + Buchstabe aufgerufen werden. Auch dieses Programm ist nicht neu. Das Programm KEY48/CMD hat die gleiche Funktion. Einen bedauerlichen Umstand möchte ich im Bezug auf dieses sonst so nützliche Utility noch anführen. Es läuft leider nicht mit dem schon angesprochenen Bildschirmeditor, obwohl dies wünschenswert wäre.

11. Ein Utility-Pack mit folgenden Unterprogrammen:
- -Auflisten (auch auf den Drucker) der Systemparameter.
- -Aendern der Systemparameter (komfortabel).
- -Auflisten der PDRIVE-Parameter (Drucker).
- -Aenderungs-Routine für die PDRIVE-Parameter (weniger komfortabel).
- -Eine kurze Inforoutine, die über den druckbaren Zeichensatz, die angeschlossenen Drives und anderes informiert.
- -Eine komfortable Routine zum Einstellen der Druckparameter von Itoh-Druckern (wohl dem der einen solchen sein eigen nennt).

12. Last not least ein Wahlprogramm mit dem auf einen Tastendruck entweder Programme aufgerufen oder DOS-Kommandos ausgeführt werden können. Man erspart sich damit viel Tipperei, vor allem bei den teilsweise ja sehr langen COPY- Befehlen.

Alle angesprochenen Utilitys sind natürlich nicht neu. Jedes dieser Programme gibt es einzeln für die verschiedenen Betriebssysteme und SUPERDOS ist im Endeffekt nur (?) eine Zusammenziehung nützlicher Unterroutinen zu einem eine Einheit bildenden Betriebssystem. Trotzdem, oder vielleicht gerade deswegen lohnt sich die Anschaffung dieses DOS. Alle hier enthaltenen Utilitys einzeln zu erwerben kostet entweder viel Zeit (Tausch) oder eine Menge Geld (Kauf). So bietet z.B. der bekannte (berüchtigte) Softwareverkäufer Hanke (dies soll keine Schleichwerbung sein) das SUPERDOS für 50.00 DM an. Gleichzeitig kann man über ihn auch die Utilitys einzeln beziehen. Man kommt dann jedoch auf einen Preis von über 100.00 DM (in Worten: einhundert). Ob sich die Anschaffung lohnt, hängt also davon ab, wieviele der angesprochenen Utilitys man schon besitzt.Ganz kurz möchte ich noch das im SUPERDOS enthaltene BASIC ansprechen. Es hat zusätzlich zum normalen Basic, wie es z.B. NEWDOS80 enthält, zwei weitere Befehle. Und zwar den Line- und den Name-Befehl. Mit dem Line-Befehl können sehr schnell Linien, Rahmen, ausgefüllte Rechtecke usw. erzeugt werden. Der Befehl Name bewirkt ein formatiertes Listing von Basicprogrammen, welches die Übersichtlichkeit erheblich erhöht. Mit einem ":" als Zusatz kann auch auf dem Drucker formatiert gelistet werden.

Ich hoffe, dieser kleine Bericht hat Euch interessiert und so informiert, daß Ihr in der Lage seid, zu entscheiden, ob sich die Anschaffung von SUPERDOS lohnt.
(PS. Die Adresse von Hartmut Obermann könnt Ihr dem Clubinfo entnehmen.)

CLUB 80

Clubinfo

der

TANDY – GENIE – und KOMTEK – ANWENDER

21. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
POSTFACH 27 / TEL.: 06124 / 3913

INHALTSVERZEICHNIS

Seite:
und Autor:

Clubinternes

Neues vom Vorstand .... 1 - 3
Hartmut Obermann

Gebruikers Vereinigung 4
Joop Groenendijk

Auf zu neuen Ufern 5
Paul-Jürgen Schmitz

Termine / Messen 6
Redaktion

Club-Emblem .... 7 - 8
Gerald Schröder

Software

Werkzeuge eines Assembler-Prog. ... 9 - 10
Gerald Schröder

BASIC ruft Assembler .... 11 - 13
Artikel aus ELCOMP

Flopcopy auch ohne Interrupt 14

Dreispaltig drucken .... 15 - 21
Arnulf Sopp

CMD --) COM ohne CPMac .... 23 - 24

Wie baue ich mir ein CP/M? .... 25 - 29
Gerald Schröder

MS-DOS Ecke 30

Model 4 - Info 30
Klaus Hermann

Turbo Tip 31

Turbo Patch 32
Hartmut Obermann

dBase II .... 33 - 35
Harald Mand

Apfelmännchen auf der GDP64 .... 37 - 42
Helmut Bernhardt

Hardware

Genie, CP/M und 80 Zeichen und HRG 43 - 50
Manfred Held

Portbelegung .... 44 - 50
Eckehard Kuhn

Seite:
und Autor:

Börse

Wer hat was -- wer will was 51

Sonstiges

Arbeitstier mit gesundem Herz 52
Heinrich Betz

Was habt Ihr gegen CP/M .... 53 - 54
Helmut Bernhardt

LS-DOS 6.3 - Kopierschutz 55
Klaus Hermann

Das TSCRIPS-Drama .... 56 - 61
Klaus-Jürgen Mühlenbein

Programmbibliothek

Neue Eingänge in der Diskettothek .. 63 - 66
Werner Förster, Andreas Rychlik

Club Bibliothek

Club-80 - Bücherei .... 65 - 66
Hartmut Obermann

Die Bücherliste / Buchvorstellungen 67 - 81
Gerald Schröder

Die letzten Seiten

Impressum 82

Schluß 83
Redaktion

Hofacker-Verlag (Brief) 84
Hofacker-Verlag

Clubmitgliederadressen .... am INFO-Ende
Redaktion

Sonderausgaben .... am INFO-Ende
Gerald Schröder, Klaus-Jürgen Mühlenbein

Neues vom Vorstand

Diesmal könnte ich mit dem Material, welches sich in letzter Zeit bei mir angesammelt hat ein ganzes Info nur mit Neuigkeiten vom Vorstand füllen. Da ich sicher bin, daß dies nicht in euerem Interesse ist, werde ich mich möglichst kurz fassen und trotzdem versuchen nichts wesentliches auszulassen!

ELCOMP
Zumindest den TRS 80-Freunden, die sich schon länger mit ihrem Tandy-Computer herumschlagen, wird der Name ELCOMP ein Begriff sein. Diese vom Hofacker-Verlag herausgegebene Computer-Zeitschrift (die übrigens eine der ersten ihrer Art war!) beschäftigte sich in ihren Anfangsjahren recht häufig mit dem TRS 80-Computer. Da auch die Firma Hofacker mit der Zeit gehen muß, ist natürlich aus der heutigen ELCOMP die Tandylinie praktisch ganz verschwunden.

Unser Mitglied Ulrich Böckling hat sich nun einmal die Mühe gemacht und alle seine ELCOMP-Ausgaben (und er besitzt praktisch alle!) nach TRS 80-spezifischen Artikeln zu durchforsten. Die reiche Ausbeute, die er dabei gemacht hat, wird nun von Zeit zu Zeit bei uns im Info erscheinen und so auch den TRS 80-Usern zur Verfügung gestellt, denen die ELCOMP und die darin erschienenen Veröffentlichungen bisher kein Begriff waren!

In diesem Zusammenhang möchte ich dem Hofacker-Verlag für die freundliche Genehmigung danken, die uns erst ermöglicht, die ELCOMP-Artikel abzudrucken. Bitte beachtet dazu den Brief des Hofacker-Verlags, der aus postalischen Gründen leider auf der letzten Umschlagseite abgedruckt werden muß!

Mein ganz besonderer Dank gilt unserem Freund Ulrich Böckling, der uns durch seine Arbeit ein riesiges Reservoir interessanter Beiträge für das Clubinfo erschlossen hat!

Anschaffungen
Während des Vorstandstreffens wurde beschlossen ein Reinigungsset für Diskettenlaufwerke zu beschaffen. Zuständig dafür ist Werner Förster, bei dem das Set auch ausgeliehen werden kann. Genaueres dazu wird er euch sicher bald mitteilen! Weiterhin wird die Büchersammlung des CLUB 80 aufgestockt. Dazu wird Gerald Schröder eine Auswahl von Büchern zusammenstellen, über deren Ankauf der Vorstand dann entscheiden wird. Die Bücher sollen vor allem aktuell sein und sich mit CP/M sowie mit Hochsprachen beschäftigen. Weiterhin wird der Ankauf von Public-Domain-Software aus USA, sowohl für TRS/NewDOS als auch für CP/M erwogen!

CP/M Programmbibliothek
Wie schon im letzten Info berichtet, hat der CLUB 80 nun auch eine CP/M-Programmsammlung! Als Verwalter hat sich Andreas Rychlik zur Verfügung gestellt, dessen Vorstellung und Konzept ihr in diesem Info lesen könnt. Ich wünsche ihm mit der übernommenen Arbeit viel Glück und Spaß und hoffe, daß er sich bald vor Bestellungen nicht mehr retten kann!

Rückporto
Im Zusammenhang mit der CP/M-Programmbibliothek wurde auf dem Vorstandstreffen entschieden, daß ab sofort Besteller von Programmen aus einer der Sammlungen, genau wie Buchentleiher, die Portokosten tragen müssen! Es werden also nur noch dann TRS/NewDOS- und CP/M-Programme verschickt, wenn der Bestellung entsprechendes Rückporto beigelegt wurde. Ich bitte dies bei Bestellungen zu beachten!

Werbung
Es steht zwar nicht zu befürchten, daß unser Club demnächst an Mitgliedermangel zu Grunde geht, trotzdem soll mehr Werbung gemacht werden! Ein großer Teil dieser Werbung wird vom Vorstand durch eine Beilage zu einer Werbesendung der Firma Hermstedt realisiert. Weiterhin wird ein neues Werbeplakat erstellt, daß dann bei Messen, Ausstellungen und Flohmärkten sowie in Computerläden ausgehängt werden soll. Hier sind nun die Mitglieder gefordert!

Jedes Mitglied sollte sich nach guten Plätzen für die Plakate umsehen. Wenn eine Austellung oder ähnliches in eurem Bereich läuft, solltet ihr dort ein Plakat aufhängen und auch eurem Computerhändler könnt Ihr sicher eines unterjubeln. Mancher ahnt gar nicht, daß ganz in seiner Nähe noch ein Tandy-Besitzer einen recht einsamen Kampf mit seinem TRS 80 ausficht. Durch solche Aktionen können Verbindungen geschaffen werden, die sehr nützlich sein können!

Weiterhin sollten die Mitglieder mehr Mundpropaganda für den CLUB 80 machen! Sicher kennen viele unserer Mitglieder Tandy-Benutzer, die noch nicht dem Club angehören aber am Austausch von Informationen interessiert sind. Solche Leute sollten verstärkt angesprochen und für den Club geworben werden!

Material, wie Probeinfos und Werbeplakate, könnt ihr bei mir bekommen!

MS-DOS - Ecke
Auch dieses Problem wurde auf dem Vorstandstreffen diskutiert. Als Ergebis kam heraus, daß vorerst keine spezielle MS-DOS-Ecke oder -Gruppe geschaffen wird. Es werden aber selbstverständlich Berichte und Artikel unserer Mitglieder, die sich mit MS-DOS beschäftigen veröffentlicht (siehe z.B. Info 20 Seite 99/100). Die Entscheidung, ob sich solche Artikel häufen, wird damit praktisch den Schreibern und Lesern des Infos, also allen Mitgliedern des CLUB 80 überlassen!

Ich hoffe auch weiterhin auf rege Diskussionsbeiträge zu diesem Thema und möchte in diesem Zusammenhang auf einen Brief von Paul-Jürgen Schmitz zu diesem Thema hinweisen, der in diesem Info abgedruckt ist!

HEFT
21
September
1987

TRS 80 Gebruikers Vereniging

Schon seit einiger Zeit hat der CLUB 80 durch Gerald Dreyer Kontakt mit einem holländischen TRS 80-Club. Nun hat der Vorstand der "TRS 80 Gebruikers Vereniging" eine kurze Vorstellung für unser Info geschrieben, die im Anschuß an diese Rubrik abgedruckt wird. Leider haben wir immer noch das Problem, daß wir niemand haben, der uns die interessanten Artikel aus der Zeitschrift des holländischen Clubs übersetzt. Dadurch gehen uns einige wirklich interessante Veröffentlichungen verloren! Sollte jemand im Club der holländischen Sprache mächtig sein, sollte er sich auf alle Fälle bei mir melden, damit wir die Sprachbarriere durchbrechen können!

Clubtreffen 88

Es erscheint zwar noch nicht sehr dringend, trotzdem hat der Vorstand schon das Datum für das Clubtreffen 1988 festgelegt. Das Treffen findet im Zeitraum zwischen dem 12. und 15. Mai statt. Der 12. ist ein Donners- und Feiertag, so daß wir diesmal für das Treffen etwas mehr Zeit ansetzen konnten. Ihr solltet euch diesen Termin schon jetzt in eurem Kalender vormerken und freihalten! Genaueres über Ort und Ablauf folgt in den nächsten Infos.

Zehnzeilerwettbewerb

Jens hat ja schon in seinem Schlußwort zum Info 20 von der Pleite bei der Abstimmung über die fünf eingesandten Zehnzeiler berichtet. Mangels Abstimmungsergebnis mußte der Vorstand über die Vergabe der Preise entscheiden und hat dies wie folgt getan. Jeder der vier Einsender bekommt als Anerkennung für seine Teilnahme eine Zehnerpackung Disketten. Es erfolgt damit keine Bewertung der Beiträge, da sich der Vorstand nicht in der Lage sah, ohne Hilfe der Mehrzahl der Mitglieder eine solche zu erstellen. Ich bedanke mich recht herzlich bei denen die mitgemacht haben und die nichts dafür können, daß die ganze Veranstaltung mehr oder weniger ins Wasser gefallen ist!

Damit genug für diesmal! Ich wünsche euch ein paar schöne Stunden bei der Lektüre des Infos und viel Spaß beim Computern, Euer

Hartmut Obermann

GEBRUIKERS VERENIGING
POSTBUS 551
2070 AN SANTPOORT-NOORD
TELEFOON 023-384135

Liebe Freunde,

Es ist mir eine grosse Ehre einen Beitrage zu Ihrem Blatt leisten zu dürfen.
Ueber Herrn Van de Sar, unsere Kontaktperson, habe ich schon einige Exemplare von Ihrem Blatt empfangen und jedes Mal habe ich darüber gestaunt was Sie mit einer so kleinen Gruppe leisten.
Herr Van de Sar hat mich, als Sekretär des holländisches TRS-80 Gebraucher Vereins gebeten, einen kurzen Bericht über unseren Verein zu schreiben.

Unser Verein wurde im Oktober 1978 errichtet und richtete sich ausschliesslig auf dem Gebrauch von TRS-80 Computern, wie die Modelle I, III und 4, sowie auch auf Computer die damit Compatibel waren.
Selbstverständlich hat das Aufkommen des PC auch für unseren Verein Verlust von Mitgliedern bedeutet, aber es hat sich gezeigt, dass viele Leute, obwohl sie einen PC gekauft hatten, doch eine so starke Vereinstreue hatten, dass sie Mitglied blieben.
Aus diesem Grund haben wir letzten Mai auf einer Generalversammlung entschieden, dass wir uns demnächst auch auf den Gebrauch von PC richten werden.

Im Augenblick hat unser Verein noch circa 2000 Mitglieder, diese Zahl war leider grösser gewesen, ist aber wieder am steigen seit wir den eben gemeldete Entschluss gefasst haben.

Diese Mitglieder können sich mindestens einmal monatlich in Versammlungen der neun Abteilungen treffen aus denen unser Verein besteht.
Bei diesen Versammlungen gibt es die Möglichkeit Vorträgen bei zu wohnen, oder Löt- und Programmierhilfe zu bekommen.

Weiter hat unser Verein eine Softwarebibliothek, die ungefähr 60 Disketten mit eigener Software umfasst, daneben verbreiten wir auch noch Public-Domain Software so wie Sie die auch kennen.

Auch hat unser Verein eine sehr aktieve Hardwaregruppe, die schon sehr viel geleistet hat, wie zum Beispiel : ein Modem, ein Druckerschalter, einen Ramdisk usw.

Wie auch Ihren Verein, bringen wir jede zwei Monate ein Heft, Remarks, in dem wir versuchen unsere Mitglieder von den neuesten Entwicklungen auf dem laufenden zu halten Vereinsnachrichten veröffentlichen und Programme und Hardware-Verbesserungen verbreiten.

Ich hoffe, Sie können sich mit diesen Informationen ein Bild von unser Verein machen und ich wünsche Ihnen noch viele schöne Stunden am Pult Ihren Maschinen.

Joop Groenendijk,
Sekretär TRS-80 Gebruikers Vereniging

Paul-Jürgen Schmitz
Dipl.-Ökonom

Bremer Str. 9
6236 Eschborn
T.06196/46832

P.J.Schmitz · Bremer Str.9 · 6236 Eschborn

Herrn
Hartmut Obermann
Schwalbacher Str.6

6209 Heidenrod 1

25. August 1987

Auf zu neuen Ufern

Lieber Hartmut,

Deinen Artikel habe ich mit großem Interesse gelesen. Es freut mich sehr, daß nun auch mal CP/M angesprochen wird. Daran bin ich selbst interessiert, denn in der Tat findet dieses Betriebssystem immer mehr neue Anwender, da bekannte Hardware-Anbieter, wie Schneider auch in neuerer Zeit noch Rechner mit diesem Betriebssystem neu auf den Markt bringen.

Ich bin an einem Erfahrungsaustausch dazu sehr interessiert, da ich auf meinem GENIE I mit Banker CP/M 2.2 fahren kann. Ich habe mich damit beschäftigt in der Vorstellung die vielen professionell nutzbaren Programme unter CP/M verwenden zu können, z.B. um Serienbriefe zu schreiben. Leider waren meine Bemühungen noch wenig erfolgreich. Wenn Du mir dabei weiterhelfen könntest, würde ich mich sehr freuen.

MS-DOS: Wie auch schon mein Artikel dazu ausdrückt, arbeite ich beruflich mit einem PC und suche Erfahrungsaustausch zu dieser Maschine. Dabei muß es doch kein Widerspruch sein, wenn sich ein Anhang dazu im Info fände. Natürlich finden sich zu diesem System ohnehin mehr Veröffentlichungen auf dem Markt, daß man mit Informationen eher überschwemmt zu werden droht; der Wert des Info's wäre es demnach, auf die Unterschiede, Besonderheiten, Möglichkeiten und Probleme sowie deren Lösungen aus Sicht des Aufsteigers hinzuweisen. Ich finde es deshalb unterstützenswert dieses Thema separat aufzunehmen. Es wird praktisch ja nur zu fragen sein, ob dies im Rahmen des Info-Umfanges noch zu leisten sein wird. Wir müßten mal fragen, wieviele Leser daran interessiert sind. Übrigens schreibt mir selbst Arnulf Sopp, daß er sich evtl. einen PC hinstellen wird, also warum dann nicht auch darüber "hacken".

Herzliche Grüße

Jürgen

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....25. November 1987

Süddeutsches Regionaltreffen .... 8. November 1987
Wir sehen uns auf der Hobbytronik in Stuttgart.
Treffpunkt ist der INFO-Stand der Hobbytronik im Eingangsbereich.
Zeitpunkt: ab Eröffnung zu jeder vollen Stunde bis 5 Minuten danach

Norddeutsches Regionaltreffen ....20. - 22. November 1987

A c h t u n g :

Anmeldungen zwecks Zimmerreservierungen:
Hotel Zum Lindenhof
Bremer Straße 55 2107 Rosengarten-Neundorf
Tel.: 04108/ 7148
Was sonst noch auf dem Programm steht erfahrt Ihr beim Hartmut Obermann oder beim Gerald Schröder

-- Messen '87 --

HOBBY ELEKTRONIK 87 ....Stuttgart 5. - 8. November 1987
PRODUCTRONICA ....München 10. - 14. November 1987

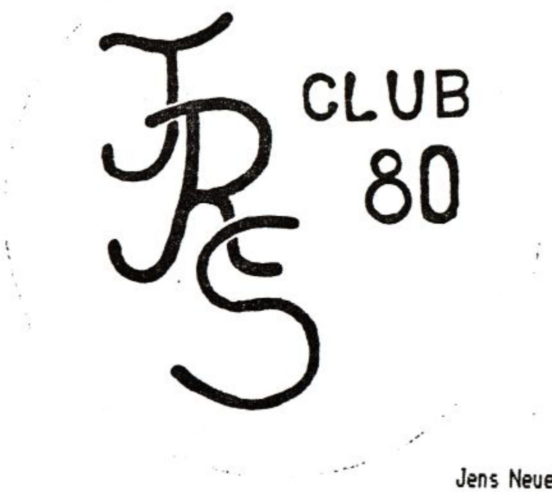
Das Club-Emblem - die Resonanz

Leider ist der Titel falsch, denn die Resonanz hat eigentlich gefehlt. Zwei Clubmitglieder (Hans-Martin und Jens, bei denen ich mich hier ganz herzlich bedanken möchte) haben mir ihre Vorschläge geschickt. Die Aktion ist trotzdem noch nicht beendet! Ich hoffe immer noch, daß sich unter Euch ein paar verkannte Künstler befinden, die sich beteiligen möchten. Schickt also ruhig Eure Vorschläge ein!

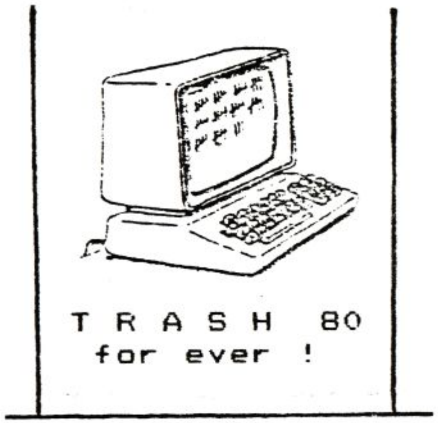
Als kleine Anregung werden nach diesem Artikel die bisher eingegangenen Vorschläge veröffentlicht. Es sind zwei von Jens und einer von Hans-Martin. Wenn Euch irgendeiner besonders gut gefällt, teilt es mir irgendwie mit (Telefon, Postkarte, Brief, Flaschenpost, persönliches Erscheinen,...).

Wie gesagt, erwarte ich noch Vorschläge und Stellungnahmen von Euch. Irgendwann wird dann der Vorstand endgültig darüber entscheiden, welches Emblem wir nehmen. Mehrheitsentscheidungen der Club-Mitglieder sind nach Jens' Erfahrungen nicht möglich, also ist es wohl so am besten.

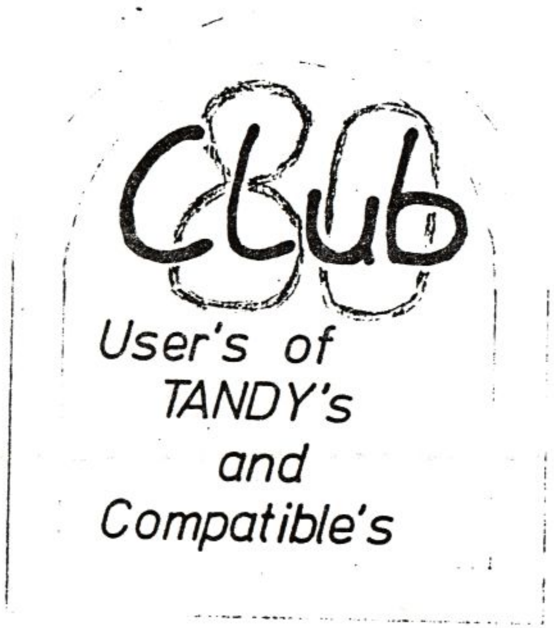
Gerald Schröder



Jens Neueder (1)



Hans-Martin Stephan



Jens Neueder (2)

Eckehard Kuhn

Die Werkzeuge eines Assembler-Programmierers

Vor kurzem sollte ich "außer Haus" unerwartet ein Problem in Assembler lösen. Dabei merkte ich erst, wie hilflos ich ohne die elementaren Hilfsmittel (zum Assemblern) bin. Ich möchte sie Euch hier kurz vorstellen, damit Ihr 1. als Anfänger ungefähr wißt, was Ihr Euch besorgen solltet und 2. als Fortgeschrittener mal seht, wie andere Leute arbeiten. Natürlich habe ich nicht der Wahrheit letzten Schluß für mich gepachtet, und es kann sein, daß Ihr ganz andere Werkzeuge für besser geeignet haltet. Dann schreibt gefälligst einen Artikel darüber!

1. Die *Paperware*

1.1. Ein Quick-Reference-Guide
Ich benutze ein selbstgemachtes, denn ich habe noch kein geeignetes käufliches gefunden. Es sollte enthalten:

- alle Z80-Opcodes, alphabetisch geordnet mit den zugehörigen Hex-Codes (Binär muß immer erst umgerechnet werden); sinnvoll beim schnellenCodierenvonwenigenBefehlenperDEBUG
- zu allen Codes die Art, wie sie die Flags beeinflussen und bei einigen auch, was sie genau machen (Rotier-/Schiebe-Befehle); dauerndbeimProgrammierenzugebrauchen
- alle Z80-Opcodes, aufsteigend nach Hex-Code sortiert; zum Disassemblieren (Decodieren) von wenigen Befehlen per DEBUG

1.2 Eine Umwandlungstabelle
Die Tabelle sollte 255 Einträge enthalten, jeweils mit ASCII-Zeichen und der dazugehörigen Zahl in dezimaler, hexadezimaler und binärer Schreibweise. S. auch ASCII-TAB aus Info 14. Diese Tabelle hilft bei kleinen Umrechnungen enorm.

1.3 Ein Buch mit einigen Assembler-Routinen
Dies ist sinnvoll, wenn man einige Algorithmen oft benötigt. Ein Beispiel ist die Umwandlung von ASCII-Eingaben in Zahlen oder umgekehrt von Zahlen (Registerinhalten) in darstellbare ASCII-Codes (Ziffern). Eine gute Sammlung von Standard-Routinen ist auch gut. Das Z80-Buch von Zaks bietet einige Mathe-Algorithmen; ansonsten kann ich es leider nicht empfehlen und kenne auch kein anderes Buch, das sich dafür eignet. Also: selbstmachen.

1.4 Das DOS-Buch von Grosser
Unverzichtbar beim Programmieren des Floppy-Controllers und beim intensiven Arbeiten mit den Innereien des DOS'.

1.5 Die Seiten 53-64 aus dem Newdos-Handbuch
Hier werden alle zum Diskettenzugriff wichtigen Routinen hervorragend beschrieben. Im Gegensatz zum DOS-Buch allerdings nicht die Routine selbst, sondern wie man sie benutzt. So sind schnell und effektiv Programme mit Diskettenzugriff schreibbar.

1.6 Das ROM-Listing von Röckrath
Das ROM selber ist nicht so interessant, obwohl man einige Routinen (bei mir vor allem 33h, 49h, 5d9h, RST 18h) gut gebrauchen und hier etwas über sie erfahren kann. Vor allem beim Disassemblieren von Programmen, die das ROM nutzen, ist es aber unverzichtbar. Schön ist auch die Belegung der Tastatur-Adressen dargestellt, so daß man leicht direkt darauf zugreifen (obwohl nicht unbedingt empfehlenswert).

2. Die *Software*

2.1 ZEUS
Zweifellos **das** Werkzeug für den Assembler-Programmierer. Allein der automatische Syntax-Check ist goldwert. Wenn Ihr Arnulf und mir nicht glauben wollt, sondern lieber den Edtasm oder sonstwas benutzt, ist das auch OK.

2.2 DSMBLR
Der Misosys-Disassembler kann eine verdammte Menge, vor allem, wenn man ihm sagt, wo DBs usw. stehen. Dies kann man mit FED feststellen (s. 2.3). Die Outputs können mit ZEUS bearbeitet werden. Durch diese Reihenfolge (FED, DSMBLR, ZEUS) habe ich schon viele Programme erforscht, kommentiert und schließlich geändert.

2.3 FED
Der File-Editor bietet mehrere Funktionen, wegen derer er sich hervorragend zur Erkundung von Programmen eignet. Um nur einige zu nennen: Suchen nach einer Adresse (L); Anzeigen, an welcher Adresse man sich befindet (M); Blättern durch die Ladecodes (Z). Super ist das Ding, um sich anzusehen, wo DMs/DWs/DBs in einem File auftauchen könnten (ASCII-Texte einfach zu lokalisieren, Tabellen oft durch stetig wiederkehrendes Zeichen (jedes zweite, nämlich den MSB) gekennzeichnet), um dies dann DSMBLR mitzuteilen. Dadurch kann man sich oft den Ärger ersparen, die Funktion von vollkommen unsinnig scheinenden Programmteilen begreifen zu wollen, die in Wirklichkeit nur die Copyright-Meldung darstellen.

2.4 DEBUG
Dürfte jeder kennen (sonst "123" drücken und das Newdos-Manual nochmal ganz genau durchlesen). Der Vorteil ist offensichtlich: Das Teil verbraucht nur DOS-Speicher und kein freies RAM. Damit (vor allen den Stepper-Funktionen I und C) lassen sich hervorragend Fehler in eigenen Programmen aufspüren oder das Verhalten fremder Programme nachvollziehen.

2.5 Routinen-Sammlung
Oft benötigte Routinen (wie oben angesprochen) sollte man sich dort abspeichern, wo sie schnell zu finden sind. Bei mir ist (neben den angesprochenen Ein-Ausgabe-Umwandlungsroutinen) dauernd ein Programmteil in Benutzung, der nach einem File-Namen fragt, diesen File dann byteweise einliest und wieder auf Diskette schreibt (evtl. unter einem anderen Namen). Zwischen dem Lesen und Schreiben füge ich dann verschiedene Routinen ein, die den File umwandeln (z.B. die Ladecodes ändern o. analysieren oder bestimmte Bytes entfernen).

3. Die *Hardware*

Hier ist nur ein Teil zu nennen: ein Taschenrechner, der auch mit Hexadezimal- und Binärzahlen rechnen bzw. diese in dezimale und umgekehrt umwandeln kann. Bei leistet dies der Casio fx-115.

Das war es schon, ohne jeden Anspruch auf Vollständigkeit. Ich hoffe, daß wenigstens die Assembler-Anfänger nun einen ungefähren Überblick haben, welche Werkzeuge in Ihren Erste-Hilfe-Kasten gehören, damit sie nicht so hilflos sind wie ich, wenn sie unerwartet um die Vorführung Ihrer Fähigkeiten gebeten werden.
Übrigens werden unter CP/M bestimmt ganz andere Werkzeuge benutzt. Ich hoffe, daß bald mal jemand darüber berichtet.

Gerald Schröder

BASIC ruft Assembler

Laden und Aufrufen von Routinen in Maschinensprache
6 Verfahren für TRS-80 und Video Genie

PET-Besitzer haben es gut. Sie können kleine Maschinenprogramme (ab jetzt nur noch MP's genannt) in den 2. Kassettenpuffer "poken", wo sie (die MP's) vor BASIC geschützt sind.

TRS-80-Besitzer haben es besser. Sie dürfen nämlich selbst bestimmen, was sie mit ihrem RAM anfangen, allerdings auch selbst darüber nachdenken, wie sie MP's und BASIC auseinander halten.

Nun, das Nachdenken des Autors erbrachte 6 Varianten. Sicherlich gibt es noch mehr, aber dieser Artikel soll ja kein Buch werden.

```
1 ' MP-CALL. VERSION 1: MP IM RESERVIERTEN RAM
2 '
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201
20 A=32750:FORX=A TO A+16:READB:POKEX,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO 50
```

Version 1

Version 1 im Listing ist die klassische Standardlösung. Das MP wird in den reservierten RAM "gepokt". Für nicht TRS-Fans: Der Rechner fragt nach dem Einschalten bzw. Booten im DOS "Memory Size ?". Wenn man die Frage nicht mit ⟨ENTER⟩ ignoriert, sondern eine Adresse eingibt, "denkt" BASIC, daß sein RAM bei dieser Adresse - 1 endet.

Das kleine MP, das hier zur Demonstration benutzt wird, füllt eine Bildschirmseite mit einem vom User wählbaren ASCII- oder Grafikzeichen. Geben Sie z. B. 65 ein, sehen Sie schlagartig 1024 A's oder bei 191 wird der Schirm weiß.

DEFUSR= ist ein Disk-BASIC-Befehl und eine praktische Abkürzung von POKE 16526, . . ., POKE 16527, . . . in Level 2. Mehr sollte man über Version 1 nicht schreiben, gehen wir zur 2.

Das MP muß lediglich als eine sinnige Folge von Bytes irgendwo im RAM stehen. Ein String darf fast den gesamten Zeichensatz also CHR$(0) bis CHR$(255) enthalten. Fast !

Steht eine 34 in dem String, weil sie da hinein "gepokt" wurde, wird das vom Computer als Anführungszeichen und somit als Stringende erkannt. LD HL,(nn), was 34 bedeutet, sollte man also im MP vermeiden. OK, lädt man also eine String mit den Bytes des MP's, wie in Zeile 20 gezeigt, und sagt dann via VARPTR, daß das MP da beginnt, wo die String im RAM startet, ist alles klar.
Zeile 21 lädt den USR-Vektor. Hier wird ein System mit ⟩16k unterstellt, was in L3 wohl der Normalfall sein dürfte.
Wenn Sie in Level 2 arbeiten und ⟨=16k haben, können Sie die beiden Bytes PEEK(X+1)/(X+2) direkt nach 16526/27 poken.

```
1 ' MP-CALL. VERSION 2: MP IN STRING. STRING IN HIGH RAM
2 '
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201
20 FOR X=1 TO 17:READ B:A$=A$+CHR$(B):NEXT
21 X=VARPTR(A$):DEFUSR=PEEK(X+1)+256*PEEK(X+2)-65536
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO 50
```

Version 2

In Version 2 steht die String mit dem MP im mit CLEAR reservierten Stringbereich, also im high RAM. Das hat den Nachteil, daß es nach jedem Programmstart erst dahin gebracht werden muß.

In Version 2.1 wird zuerst eine Stringkonstante definiert (Zeile 15). Die Zeichen darin sind beliebig. Der Autor hat die Ziffern geschrieben, um leichter 17 Zeichen (MP-Länge) abzählen zu können. Eine im Programm definierte String steht bekanntlich im Programmtext. Wenn man also in diese das MP schreibt, kann man anschließend die DATA-Zeilen und die POKE-Schleife löschen. Das Programm selbst hält dann das MP. Den Nachteil der Version 2 haben wir somit beseitigt und prompt einen neuen eingefangen.

```
1 ' MP-CALL. VERSION 2.1: MP IN STRING. STRING IM PROGRAMMTEXT
2 '
10 DATA 205,127,10,125,33,1,60,119,17,2,60,1,254,3,237,176,201
15 A$="12345678901234567"
16 A=A:X=VARPTR(A$):A=PEEK(X+1)+256*PEEK(X+2)
20 FORX=A TO A+16:READB:POKEX,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO30
```

Version 2.1

Vergleichen Sie einmal die DATA-Zeilen. In Version 2.1 sind einige Bytes geändert worden. Es gibt keine Null-Bytes mehr, und da man in einem MP nicht einfach alle Nullen in Einsen ändern kann, noch einige Changes. Warum ?

Nun, in einer BASIC-Zeile darf ein Nullbyte nur an einer Stelle stehen und zwar am Ende. Es ist das amtliche Schlußzeichen, das wir nicht einfach mitten in eine Zeile poken können.

Der seltsame Befehl A=A in Zeile 16 hat die Aufgabe A zu definierten. Man könnte auch A=4711 schreiben. A wird nach der VARPTR-Funktion benutzt. Würde man nach VARPTR A als neue Variable definieren, würde die Variablenliste länger, A$ verschoben werden, und die Routine mit einer falschen Adresse arbeiten.

Also Version 2 hat Nachteile, 2.1 auch, beide gemeinsam haben noch den Nachteil, daß das MP nicht länger als 255 Bytes (Version 2) bzw. 233 (Version 2.1) sein darf, ergo doch die Mem-Size-Lösung ?

Version 3 ist die Array-Lösung. Durch DIM MP(4) werden 5 Elemente dimensioniert, ergibt in single precision 20 Bytes, genug unsere 17 aufzunehmen.

Falls Ihr Rechner nicht mehr als 16k hat, können Sie die Routine verwenden. Darüber wird es mit dem POKE schwierig, denn bekanntlich muß man ja sobald 32767 überschritten werden, das MSB setzen, sprich ein negatives Vorzeichen einführen.

```
1 ' MP-CALL. VERSION 3: MP IN ARRAY (SINGLE PREC.)
2 '
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201
20 DIM MP(4)
21 B=B:X=X:A=VARPTR(MP(0))
22 FOR X=A TO A+16:READ B:POKE X,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 DEFUSR=VARPTR(MP(0)):C=USR(C)
50 GOTO 50
```

Version 3

Die schlechteste Lösung dafür steht im Level-2-Handbuch. Die einfachste ist 65536 zu subtrahieren.

Wie dem auch sei, der Array kann hier oder dort im RAM stehen, mal muß man positive, mal negative Zahlen poken, warum soll man den

Array nicht gleich mit den Zahlen laden (so als A(X)=B) ?

Wir sind leider in BASIC und nicht in APL. D. h. ein Array-Element belegt im Integer-Format immer noch 2 Bytes. Würden wir da eine Zahl <256 hineinladen, bliebe ein Byte auf 0, sprich jedes 2. Byte des "MP's".

Trotzdem ist Version 3.1 machbar. Man muß halt jeweils 2 Bytes in ein Element packen. Wie die Zeilen 21 – 25 zeigen, ist dies ganz einfach. Bei seinem ersten Versuch bekam der Autor promt eine Error-Meldung, die mit Zeile 23 behoben wurde. Diese prüft, ob die gepackte Zahl außerhalb der Integer-Grenze liegt (>32767) und verfährt dann wie bei POKE.

```
1 ' MP-CALL. VERSION 3.1: MP IN INTEGER-ARRAY
2 '
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201,0
20 DEFINT M:DIM MP(8)
21 FOR X=0 TO 8
22 READ L,H:B=L+256*H
23 IF B>32767 THEN B=B-65536
24 MP(X)=B
25 NEXT
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 DEFUSR=VARPTR(MP(0)):C=USR(C)
50 GOTO 50
```

Version 3.1

Die DATA-Zeile ist ein Byte länger geworden. Das MP an sich hat 17 Bytes. Da aber immer 2 Bytes gelesen werden, brauchen wir 18, weshalb ein Nullbyte angehängt wurde.
Dieser NOP wird nie ausgeführt, weil davor 201, im Klartext RET steht.

Sie möchten gerne, daß jemand, der Ihr Programm mit BREAK anhält und es dann listet oder wieder mit RUN startet, eine böse Überraschung erlebt ?

Dann nehmen Sie doch die Version 4 mit der DATA-Zeile von Version 1. In Version 4 ist die erste Zeile (0) dafür vorgesehen, anstelle der Punkte des MP aufzunehmen.
Wo Zeile 0 im RAM steht, wird in Zeile 15 festgestellt. Ansonsten ist im Programm nichts

Neues. Mit der DATA-Zeile wie im Listing, d. h. ohne Nullbytes, läuft das Programm beliebig oft. Mit Nullbytes (DATA-Zeile lt. Version 1) läuft es auch in der Schleife 30–50–30. Wenn Sie allerdings mit BREAK die Schleife verlassen und dann wieder RUN geben, ernten Sie nur Syntax-Error in Line 273.

Ein LIST zeigt dann nur wilde Zeichen in 0 und keine Zeile 273. Ursache sind wieder die Nullbytes mitten in der Zeile. Den gleichen Effekt können Sie übrigens auch in der Version 2.1 erzielen.

Wenn es Ihnen nur auf den Effekt als solchen ankommt, brauchen Sie kein MP. Poken Sie einfach im Laufe des Programms in Zeilen, die nicht mehr durchlaufen werden, eine Null.

```
0 REM......................
1 ' MP-CALL. VERSION 4: MP IN ERSTER BASIC-ZEILE
2 '
10 DATA 205,127,10,125,33,1,60,119,17,2,60,1,254,3,237,176,201
15 A=PEEK(16548)+256*PEEK(16549)+5
20 FORX=A TO A+16:READB:POKEX,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO 30
```

Version 4

P. Wollschlaeger, Hildesheim

ELCOMP 10/82

Flopcopy auch ohne Interrupt

Im Info 20/87 stellt Andreas Rychlik unter der Überschrift "Hardcopy auch ohne Interrupt" eine Methode vor, mit der Boot-Falle einen Ausdruck des Bildschirms anzufertigen. Vor Jahren wurde einmal in einer Computerzeitschrift eine ähnliche Idee veröffentlicht: Das Programm im Boot-Sektor fertigte eine Hardcopy gleich selber an.

Bei den TRS-80 und den älteren Genies ohne Banker fällt mir dazu auch nicht mehr ein. Hat der User jedoch die Möglichkeit, den unteren Speicherbereich direkt oder als Kopie (mit dem EG 64 MBA oder so) zu modifizieren, dann eröffnet sich eine einfachere, zudem flexiblere Lösung:

Programme, die die maskierbaren Interrupts (INT) ausknipsen, werden sich damit abfinden müssen, daß der nicht maskierbare Interrupt (NMI) nach wie vor ausgegeben werden kann. Bei den älteren Modellen unserer Computerfamilie geht das mit dem "Reset"-Knopf (der gar keinen Reset, sondern eben jenen NMI triggert). Beim Genie 3s können die beiden unbelegten Tasten leicht für diesen Zweck benutzt werden. Bei sonstigen Typen tut's ein Klingelknopf, der den NMI-Pin der CPU auf GND legt.

Der NMI springt bei 0066h ein. Dort steht in unserem Microsoft-Interpreter eine Kaltstart-Routine. Sie wird, falls der Interpreter überschrieben werden kann (s. o.), heimtückisch für unseren sinistren Zweck verbogen:

Der Library-Befehl DUMP soll den Bildschirminhalt als CMD-File auf Diskette schreiben, von wo er jederzeit wieder abgerufen werden kann. Die DUMP-Parameter enthalten außer dem Programmnamen (hier BILD/CMD, aber das ist Geschmacksache) natürlich auch die Bildschirmanfangs- und -endadresse sowie als Einsprungsadresse 0071h. An 0071h (Label wait) steht nämlich der Befehl JP 0049h. Er bewirkt, daß die Tastatur so lange interviewt wird, bis der User irgendeine Taste drückt. Am Ende der Tastatur-Routine steht RET, so daß es anschließend nahtlos im DOS, im BASIC oder sonstwo weitergeht.

Das hier vorgestellte Programm selbst bietet gegenüber der (bei ROM-Computern vermutlich unausweichlichen) Unbequemlichkeit einen weiteren Vorteil: Es endet nämlich ganz schlicht mit RET, so daß es in PACMAN gleich weitergeht, sobald der traumhafte High Score für die neidische Nachwelt auf Disk dokumentiert ist. RETN (2 Bytes lang) nach dem NMI ist nicht nötig, denn es restauriert lediglich den Zustand der INT-Akzeptanz (EI oder DI). Da bei Verwendung dieser Methode die INTs schon maskiert waren, können sie es nach dem NMI auch bleiben.

Arnulf Sopp

HEFT 21 September 1987

```
0066            00001         ORG     0066h         ;Entry des NMI
                00002
0066  E5        00003         PUSH    HL            ;Register retten
0067  F5        00004         PUSH    AF
0068  217400    00005         LD      HL,dump       ;dort steht der DUMP-Befehl
006B  CD1944    00006         CALL    4419h         ;DUMP-Befehl unter DOS-CALL ausführen
006E  F1        00007         POP     AF            ;Register restaurieren
006F  E1        00008         POP     HL
0070  C9        00009 start   RET                   ;weiter im Programm
                00010
0071  C34900    00011 wait    JP      0049h         ;auf Tastendruck warten
                00012
0074  44        00013 dump    DM      'DUMP,BILD/CMD,3C00H,3FFFH,0071H',0dh   ;DOS-Befehl DUMP
                00014
0070            00015         END     start         ;dort steht RET (gleich wieder ins DOS)

00000 Fehler

dump  0074    start 0070    wait  0071
```

Dreispaltig drucken

Arnulf Sopp

Das Programm, mit dem dieser Text ausgedruckt wurde, ist so lang, daß ich es nur in sehr groben Zügen erläutern möchte. Es interessiert wohl eh' keinen, wie es arbeitet. Deshalb will ich außer einigen Hinweisen zum Ablauf nur erklären, wie es an andere Hardware angepaßt werden kann, und welche Mängel es hat (mit denen man aber gut leben kann).

Fangen wir mit letzterem an. Jedes 0Ah-Byte (LF) wird brutal in 0Dh (CR) umgewandelt. Und jedes 0Dh wird als Zeilenende interpretiert. Deshalb sind beispielsweise Steuercodes unzulässig oder zumindest sehr schwierig zu handhaben, die 0Ah (10d) oder 0Dh (13d) enthalten (z. B. linken Rand auf die 10. oder 13. Stelle setzen).

Leere Zeilen (in TSCRIPS z. B. bei PF=2) am Kopf oder Fuß einer Seite werden nicht unterdrückt. So kann dort optisch eine Achterbahn entstehen. Es empfiehlt sich daher, PF=1 zu wählen (Absatz ohne Leerzeile).

TSCRIPS zählt die Seiten anders als dieses Hilfsprogramm. Daher muß im Text TM=1 und BM=72 (bei Endlospapier) gesetzt werden. Der Seitenvorschub wird vom Drucker erledigt.

Es ist den meisten von uns auch aus schmerzlicher Erfahrung bekannt, daß mehrere Drucker-Steuercodes im TSCRIPS-Text das Zeilen-Layout verhunzen können. Bei den sehr kurzen Drittelzeilen macht sich das umso auffälliger bemerkbar.

Es ist auch ziemlich schwierig, Steuercodes zu programmieren, die nur zu Beginn einer Zeile vom Drucker akzeptiert werden. In der Überschrift hätte ich z. B. gern auf den linken und rechten Randausgleich zugunsten einer Zentrierung verzichtet. TSCRIPS warf das Handtuch angesichts des Chaos von $- und &-Sequenzen.

Schon der Fettdruck im Titel dieses Artikels war ein echtes Problem. Die Ausgabe dieser Codes gilt nicht etwa nur für eine Textzeile – was weiß der Drucker von meiner Muttersprache und meinen Versuchen, sie zu schreiben? Er kennt nur Zeilen von ganz links bis ganz nach rechts. Das Fett-Kommando mußte daher nach jedem Wort wieder aufgelöst werden, sonst wäre die ganze Druckzeile fett gedruckt worden, also auch die mittlere und rechte Spalte.

Noch so ein Beispiel ist der Vorschlag für das Banking mit dem G3s, der später folgt. Wegen des Wechsels von Proportionalschrift auf LQ mußte auch nach jeder Zeile auf PS zurückgeschaltet werden, sonst wäre alles Folgende in diesen Zeilen, auch in den beiden anderen Spalten, unangenehm aufgefallen.

Und hätte ich jetzt, als ich dieses schreibe, gewußt, in welcher Spalte der Programmvorschlag erscheinen wird, hätte ich mich präziser ausdrücken können.

Es ist also so richtig etwas für Leute, die einfach nur einen Text tippen wollen und auf die äußere Form keinen gesteigerten Wert legen. Oder für totale Freaks, die den noch immer, wenn auch erschwert, möglichen TSCRIPS-Fähigkeiten ein Bonbon hinzufügen möchten.

Die Anpassung an andere Drucker als den NEC P6/7 ist einfach: Die Steuercodes müssen entsprechend umgeschrieben werden. In den Kommentaren des Listings sind sie erklärt. Codes für Features, die der eigene Drucker nicht hat, können schlicht gelöscht werden. Unabdingbar sind nur die linke (und mögl. rechte) Randeinstellung und die Möglichkeit, einen CR ohne Papiertransport zu programmieren.

Notfalls geht es sogar ohne das. Wenn bei der Grundeinstellung des Druckers (Label lp-init) Tabulatormarken gesetzt werden (die sollte der Drucker aber nun wirklich beherrschen), dann kann man zur nächsten Drittelzeile mit einem TAB (09h) gelangen. Für den beidseitigen Randausgleich sorgt notfalls das Textprogramm (bei TSCRIPS: J=Y). Falls man auf ihn überhaupt Wert legt.

Die Anpassung an einen anderen Computer als das G3s ist schon problematischer. Das Programm lebt davon, daß ein Text nicht sofort ausgedruckt, sondern zunächst gepuffert wird. Notfalls kann der Puffer in der oberen Hälfte des Textspeichers liegen, so daß es auch mit einem TRS-80 geht.

Banking ist eleganter und spart Speicherplatz. Hier wird der Megabyte-Banker von Helmut Bernhardt angewendet. Es geht auch mit seinem 256-kB-Banker. Einige Computertypen in unserem Club haben von Hause aus mehr als 64 kB. Mit denen funktioniert es auch. Gegf. muß dann auf die RAM-Floppy verzichtet werden.

Beim G3s ohne Umbauten sieht das etwa so aus: Alle Bankerei mit irgendwelchen Megabytes (hier über den Port F3h) wird ersetzt durch das Banking, das dem G3s angeboren ist: Mit der Codefolge

```
IN    A,(0F9h)
XOR   0C0h
OUT   (F9h),A
```

wird zwischen dem User-RAM in Bank 1 und einem möglichen Textpuffer in Bank 2 gewechselt.

Und schließlich das Problem von immer: Das Programm selbst, der Stack und der Textpuffer müssen irgendwo Platz haben. Wenn alle Stricke reißen, paßt alles ins Himem (das dann geschützt werden sollte). Wer aber mit Banking arbeitet, muß unbedingt darauf achten, daß das Programm und der eigene Stack in einem Common-Bereich liegen.

In der hier vorgestellten Version liegt das Programm hinter einigen TSCRIPS-Umleitungen ab 3260h. Da es einigermaßen lang ist und beim G3s bereits ab 3400h wieder Lebenswichtiges steht, mußte der Stack in die Cassetten-Routine des Interpreters verlegt werden. Das kann sich jeder Leser nach seinen eigenen Gegebenheiten einrichten.

Diese Routine arbeitet mit jedem Textprogramm zusammen, das über die Druckroutine des Interpreters printet. TSCRIPS tut das auch, wenn man eine kleine Änderung durchführt: Bei meiner Version (Rel. 5.4, aber vielleicht auch nicht mehr ganz sauber, als ich es bekam) wird nur an der Stelle 86E3h etwas auf den Drucker ausgegeben. Hier kann CALL 05BBh (CD-BB-05) gepatcht werden.

Bei der Anwendung zusammen mit TSCRIPS muß der linke Rand mit LM=0 ganz nach links gesetzt werden. Die RM-Einstellung ist vom Pitch abhängig, also etwa 20 – 40. Außer bei Proportionalschrift muß dieser Wert unbedingt derselbe sein wie der Druckercode für den rechten Rand.

Wer Lust hat, kann das Programm zum Bestandteil von TSCRIPS machen und für dieses Feature z. B. die @G-Option (DIR anzeigen) opfern. In der hier vorgestellten Form wird es als CMD-File mit @O oder auch schon vor TSCRIPS aufgerufen.

Es bleibt auch nach dem Ausdrucken aktiv, d. h. die Umleitungen des Druckers und der Interrupts werden nicht rückgängig gemacht. Es könnte ja sein, daß der Benutzer mit einem Text noch nicht zufrieden ist und er ihn verändert noch einmal ausdrucken will. Der Nachteil: Alle folgenden Ausgaben auf den Drucker, auch nach der Arbeit mit dem Textprogramm (z. B. DIR,P) werden dreispaltig gedruckt. Man sollte also zuvor neu booten.

Oder man ruft das Programm erneut auf, diesmal mit dem Zusatz ,N. Gemäß der G-DOS-Gewohnheit wird dadurch der normale Zustand der Drucker- und der INT-Routine restauriert. Dieser Teil des Programms läuft im Sektorpuffer des DOS, wo er nicht stört und keinen anderweitig benötigten Speicherplatz verbraucht.

Jetzt in aller gebotenen Kürze die Arbeitsweise des Programms. Durch die Drucker-Umleitung wird der ASCII-Klartext nicht sogleich ausgedruckt, sondern in einem Puffer zwischengespeichert (UP buff). Dabei werden ein Zeiger auf das bisherige Textende und ein Zeilenzähler unterhalten.

Inzwischen wartet eine Umleitung der Interrupt-Serviceroutine darauf, daß die PRINT-Taste des G3s gedrückt wird. Wer die nicht hat, kann stattdessen jede andere Taste zu diesem Zweck nehmen.

Wenn PRINT gedrückt wurde, beginnt der wirkliche Ausdruck mit einigen Berechnungen: Zunächst wird der erste CR nach der letzten Zeile aufgesucht. TSCRIPS und andere Programme haben nämlich die Untugend, mit einer Folge von Zeilenvorschüben das Blatt auszustoßen. Da dabei die Perforation niemals an der Abreißkante landet, ist das überflüssiger Schnickschnack.

Nicht nur das; in unserem Falle würden die abschließenden Zeilenvorschübe dem Programm einreden, es gäbe noch allerhand Zeilen. Sie würden – Fluch der Logik – die rechte Spalte mit allerhand leerem Papier anfüllen. Das muß nicht sein.

Die Stelle des stattdessen ermittelten wirklich letzten CR wird nun als Zeiger auf das tatsächliche Textende geladen. Bis dorthin werden nun die CRs, also die Zeilen des Textes abgezählt. Da bei dem hier sichtbaren Format 195 Textzeilen auf eine Seite passen, wird die Zeilenzahl zunächst durch diesen Wert dividiert, um die Anzahl der Seiten zu berechnen.

Für jede neue Seite muß das Zeilenlayout neu ermittelt werden. Das geht so: Je nach dem Ergebnis einer Division durch 3 (drei Drittelzeilen) wird der Beginn der mittleren und der rechten Textspalte festgestellt. Beim Ausgeben einer Druckzeile, die ja drei auseinanderliegende Textzeilen enthält, wird nun abwechselnd eine Textzeile aus dem linken, mittleren und rechten Bereich ausgegeben. Dazwischen liegen jeweils CRs, damit der Randausgleich des Druckers hinhaut. Durch die Einstellung von 0/180' Zeilenabstand (= überhaupt kein Papiervorschub) bleibt der Drucker in der Zeile. Erst vor der nächsten linken Spalte wird ein normaler Zeilenvorschub ausgegeben.

Irgendwann später ist schließlich der ganze Text ausgedruckt. Die Drucker-Routine, die ja nun nicht mehr drucken, sondern puffern soll, wird wieder verbogen. Mit RET wird am Ende nicht etwa gleich zum Textprogramm, sondern in

die Fortsetzung der INT-Routine verzweigt. Diese kehrt zurück, wenn sie die Register wieder restauriert hat, die sie zuvor rettete

Stutzt da jemand? Wenn ein INT auftritt, werden hardwarmäßig die Interrupts disabled. Das Ausdrucken geschah zwar während des Interrupt-Service, da aber weitere INTs einfach ausgeschlossen waren, ist das vollkommen gleichgültig. Ein Interrupt-Service von 10 Minuten Dauer (bei vielen Textseiten) schadet weder dem Computer noch dem Charakter.

Und nun viel Spaß mit der Heiligen Dreispaltigkeit (und dem Abtippen!)

```
                ;***************************************************************
                ;
                ;               Programm zum dreispaltigen Ausdruck von Texten
                ;               Wird vor dem Ausdruck über DOS-CALL aufgerufen
                ;
                ;               Arbeitet mit einem modifizierten TSCRIPS, wahrschein-
                ;               lich jedoch mit jedem ähnlichen Textprogramm zusammen
                ;
                ;       Modifikationen des Interpreters können, falls er in einem ROM
                ;       steht, durch Abfangen der Routinen in RAM-CALLs ersetzt werden.
                ;       Die ORGs können beliebig verlegt werden, dürfen jedoch nicht mit
                ;       dem Text des Editors in Konflikt geraten (gegf. HIMEM ändern).
                ;       Das Banking über Port F3 kann je nach Hardware auf eine andere
                ;       Technik der Pufferung des ASCII-Druckertextes geändert werden.
                ;       Die Steuercodes für den NEC P6/7 müssen je nach Drucker angepaßt,
                ;       gelöscht oder ergänzt werden.
                ;
                ;                       (C) 1987 by The HACKTORY Arnulf Sopp
                ;
                ;***************************************************************

4000            txt_buf EQU     4000h           ;in Bank 1 oberhalb Comm. (gegf. verleg.)

4601                    ORG     4601h           ;in der Interrupt-Serviceroutine
4601    8B32            DW      int             ;Umleitung auf PRINT-Tastentest

                ;---------------------------------------------------------------

3260                    ORG     3260h           ;hinter ein paar TSCRIPS-Umleitungen

                ;die Druckdaten abfangen und stattdessen im RAM ablegen
3260    E5      buff    PUSH    HL              ;Register retten
3261    F5              PUSH    AF
3262    F3              DI                      ;INTs aus, sonst passiert ein Unglück
3263    67              LD      H,A             ;Zeichen retten
3264    3E01            LD      A,01h           ;auf Bank 1
3266    D3F3            OUT     (0f3h),A        ;umschalten
3268    7C              LD      A,H             ;Drucker-Byte
3269    FE0A            CP      0ah             ;war es ein LF?
326B    2002            JR      NZ,byte_ok      ;falls nein
326D    3E0D            LD      A,0dh           ;evtl. LF in CR umwandeln
326F    210040  byte_ok LD      HL,txt_buf      ;Zeiger auf das aktuelle Drucker-Byte
3270            point   EQU     $-2             ;verändert sich
3272    77              LD      (HL),A          ;puffern anstatt zu drucken
3273    23              INC     HL              ;Zeiger erhöhen
3274    3600            LD      (HL),00h        ;nächste Pufferstelle vorsichtsh. löschen
3276    227032          LD      (point),HL      ;und wieder merken
3279    FE0D            CP      0dh             ;war oder wurde es ein CR?
327B    2007            JR      NZ,cnt_ok       ;falls nein
327D    2A5433          LD      HL,(lin_cnt)    ;Zeilenzähler laden
3280    23              INC     HL              ;erhöhen
3281    225433          LD      (lin_cnt),HL    ;wieder merken
3284    AF      cnt_ok  XOR     A               ;A <- 00: auf Bank 0
3285    D3F3            OUT     (0f3h),A        ;zurückschalten
3287    F1              POP     AF              ;Register restaurieren
3288    E1              POP     HL
3289    FB              EI                      ;Interrupts wieder zulassen
328A    C9              RET                     ;weiter in TSCRIPS

                ;---------------------------------------------------------------

                ;Umleitung der INT-Serviceroutine: prüfen, ob die PRINT-Taste gedr. ist
328B    3AE038  int     LD      A,(38e0h)       ;Tastaturzeile mit PRINT-Taste
328E    E680            AND     80h             ;wurde sie gedrückt?
3290    CABE45          JP      Z,45beh         ;INT-Routine fortsetzen, falls nein

                ;---------------------------------------------------------------

                ;die PRINT-Taste wurde gedrückt, es geht los:
                ;neuen Stack einrichten, Druckerumleitung rückgängig machen
3293    ED73EB32        LD      (sp_buff),SP    ;Stackpointer wegen des Bankings retten
3297    316002          LD      SP,0260h        ;Stack in der Überfl. Cassetten-Routine
329A    3E32            LD      A,32h           ;Opcode LD (nn),A
329C    21E837          LD      HL,37e8h        ;dort ist der Drucker memory mapped
329F    CDF832          CALL    restore         ;Befehl LD (37E8),A in die Druckroutine
32A2    21C233          LD      HL,lp_init      ;Initialisierungsdaten des Druckers
32A5    CDA233          CALL    print           ;Drucker soft resetten usw.

                ;den Zeilenzähler auf die tatsächliche Textlänge einstellen
32A8    3E01            LD      A,01h           ;auf Bank 1
32AA    D3F3            OUT     (0f3h),A        ;umschalten
32AC    2A7032          LD      HL,(point)      ;Zeiger hinter Textende
32AF    3E0D            LD      A,0dh           ;CR als Suchbyte
32B1    47              LD      B,A             ;BC <- mind. 0D00, langt als Suchstrecke
32B2    EDB9            CPDR                    ;HL vor letzten CR des Textes stellen
32B4    23              INC     HL              ;auf letzten CR vorstellen
32B5    ED5B5433        LD      DE,(lin_cnt)    ;dort Anzahl Textzeilen
32B9    13              INC     DE              ;weil die Suche mit DEC DE beginnt
32BA    010000          LD      BC,0000h        ;Suchstrecke für Schluß-CR vorbereiten
32BD    1B      end_lop DEC     DE              ;Zeilenzähler erniedrigen
32BE    0C              INC     C               ;BC <- 0001 als Suchstrecke
32BF    EDB9            CPDR                    ;CR 1 Byte zuvor aufsuchen
32C1    28FA            JR      Z,end_lop       ;so lange, bis anderes Byte gefunden
32C3    23              INC     HL              ;auf ersten CR des CR-Strings stellen
32C4    13              INC     DE              ;letzten CR noch mitzählen
32C5    EB              EX      DE,HL           ;HL <- Anzahl Textzeilen
32C6    AF              XOR     A               ;A <- 00: auf Bank 0
32C7    D3F3            OUT     (0f3h),A        ;umschalten

                ;Anzahl der Seiten berechnen, vollständige Seiten ausdrucken
32C9    3EC3            LD      A,195           ;= 65 Druckzeilen * 3 Textzeilen
32CB    CDB44C          CALL    4cb4h           ;HL <- volle Seiten, A <- Restzeilen
32CE    4F              LD      C,A             ;retten
32CF    7C              LD      A,H             ;prüfen, ob nur eine angebr. Seite
32D0    B5              OR      L               ;dann HL = 0000
32D1    79              LD      A,C             ;Akku restaurieren
32D2    F5              PUSH    AF              ;Zeilenzähler des letzten Blattes retten
32D3    44              LD      B,H             ;BC als Seitenzähler
32D4    280E            JR      Z,lst_pag       ;falls nur eine angebrochene Seite
32D6    4D              LD      C,L             ;Seitenzähler vervollständigen
32D7    C5      pag_lop PUSH    BC              ;retten
32D8    21C300          LD      HL,195          ;Zeilenzahl einer Seite
32DB    CDFF32          CALL    one_pag         ;eine Seite ausdrucken
32DE    C1              POP     BC              ;Seitenzähler
32DF    0B              DEC     BC              ;prüfen, ob alle ganzen Seiten ausgedr.
32E0    78              LD      A,B
32E1    B1              OR      C
```

```
32E2 20F3            JR      NZ,pag_lop       ;falls noch nicht

     ;letzte angebrochene Seite ausdrucken
32E4 F1      lst_pag POP     AF               ;Zeilenzahl des letzten Blattes
32E5 60              LD      H,B              ;H <- 00
32E6 6F              LD      L,A              ;HL <- Zeilenzahl des letzten Blattes
32E7 CDFF32          CALL    one_pag          ;letzte Seite ausdrucken

     ;Ausgang: alten Zustand restaurieren, über INT-Forts. nach TSCRIPS zurück
32EA 310000          LD      SP,0000h         ;alten Stack restaurieren
32EB         sp_buff EQU     $-2              ;(SP bis jetzt hier gepuffert)
32ED 210040  start   LD      HL,txt_buf       ;Anfangsadresse des Textpuffers
32F0 227032          LD      (point),HL       ;den Zeiger wieder auf 0 stellen
32F3 3EC3            LD      A,0c3h           ;Opcode JP nn
32F5 216C32          LD      HL,buff          ;Sprungziel
32F8 32BB05  restore LD      (05bbh),A        ;JP buff dort patchen
32FB 22BC05          LD      (05bch),HL
32FE C9              RET                      ;weiter im TSCRIPS-Editor (nach INT-Ende)

     ;Anzahl und Layout der Druckzeilen einer Seite berechnen
32FF 23      one_pag INC     HL               ;Zähler ab 1, nicht ab 0
3300 228433          LD      (lin_cnt),HL     ;neu vermerken
3303 2B              DEC     HL               ;alter Wert
3304 3E03            LD      A,3              ;Divisor für 3 Textzeilen pro Druckzeile
3306 CDB44C          CALL    4cb4h            ;HL <- HL/A, A <- Rest (Textzeilen /3)
3309 45              LD      B,L              ;Anzahl der mittleren Drittelzeilen
330A 2805            JR      Z,div_don        ;falls ja
330C 23              INC     HL               ;sonst eine linke u./o. mittl. Zeile mehr
330D 3D              DEC     A                ;Divisionsrest 1? (10/3, 13/3, 16/3 usw.)
330E 2801            JR      Z,div_don        ;falls Rest 1 (nur linke Spalte länger)
3310 45              LD      B,L              ;falls Rest 2 (linke u. mittl. Sp. läng.)
3311 7D      div_don LD      A,L              ;Anzahl der linken Drittelzeilen
3312 323E33          LD      (ln_cnt1),A      ;als Suchzähler patchen
3315 78              LD      A,B              ;dto. Anzahl mittlere Drittelzeilen
3316 324633          LD      (ln_cnt2),A

     ;eventuellen Drucker-Reset im Text löschen
3319 3E01            LD      A,01h            ;auf Bank 1
331B D3F3            OUT     (0f3h),A         ;umschalten
331D 010040          LD      BC,txt_buf       ;Startadresse des ASCII-Textes in Bank 1
331E         tx_beg1 EQU     $-2              ;wird ab 2. Seite geändert
3320 2A0040          LD      HL,(txt_buf)     ;erste 2 Drucker-ASCII-Bytes im Puffer
3321         tx_beg2 EQU     $-2              ;verändert ab 2. Seite
3323 11C046          LD      DE,1bh+256*'$'   ;DE <- 'ESC-$' (Halb-Reset des Druckers)
3326 E5              PUSH    HL               ;retten
3327 ED52            SBC     HL,DE            ;sind die ersten beiden Bytes ein Reset?
3329 E1              POP     HL
332A 2808            JR      Z,ign_res        ;löschen, falls ja
332C 1C              INC     E                ;DE <- 'FS-$' (auch Reset, aber brutaler)
332D B7              OR      A                ;Cy löschen
332E E5              PUSH    HL
332F ED52            SBC     HL,DE            ;war es der Totalreset des Druckers?
3331 E1              POP     HL
3332 2002            JR      NZ,strt_ok       ;falls kein Reset als Textanfang
3334 03      ign_res INC     BC               ;sonst Textzeiger hinter 'Reset' stellen
3335 03              INC     BC               ;(2 Bytes für 'FS-$' bzw. 'ESC-$')

     ;Zeiger auf die drei Drittelzeilen errechnen
3336 60      strt_ok LD      H,B              ;HL <- BC (Zeiger auf Textanfang,
3337 69              LD      L,C              ;evtl. hinter Reset-Sequenz)
3338 225733          LD      (lft_col),HL     ;als neuen Textanfang laden
333B 3E0D            LD      A,0dh            ;CR als Signal für Zeilenende
333D 0600            LD      B,00h            ;Drittelzeilen-Zähler links
333E         ln_cnt1 EQU     $-1              ;je nach dem
```



```
333F CDBA33          CALL    seek_ln          ;1. mittlere Drittelzeile suchen
3342 226333          LD      (mid_col),HL     ;ihren Anfang dort patchen
3345 0600            LD      B,00h            ;Drittelzeilenzähler Mitte
3346         ln_cnt2 EQU     $-1              ;je nach dem
3347 CDBA33          CALL    seek_ln          ;1. rechte Drittelzeile suchen
334A 226F33          LD      (rgt_col),HL     ;dort patchen

     ;Text dreispaltig ausdrucken
334D 21CD33  prt_lop LD      HL,no_lf         ;Codestring für CR ohne LF
3350 CDA233          CALL    print            ;ausgeben
3353 11DE33          LD      DE,margins       ;Zeiger auf Randeinstellungen
3356 210000          LD      HL,0000h         ;Anfang der 1. Drittelzeile
3357         lft_col EQU     $-2              ;evtl. hinter der Reset-Sequenz
3359 CD8233          CALL    prt_lin          ;eine Drittelzeile ausdrucken
335C 225733          LD      (lft_col),HL     ;Anfang der nächsten 1. Drittelzeile
335F 228133          LD      (sto_end),HL     ;= evtl. Ende der Seite
3362 210000          LD      HL,0000h         ;Anfang der 2. Drittelzeile
3363         mid_col EQU     $-2              ;je nach dem
3365 CD8233          CALL    prt_lin          ;nächste Drittelzeile ausdrucken
3368 226333          LD      (mid_col),HL     ;Zeiger nachstellen
336B 228133          LD      (sto_end),HL     ;s. o.
336E 210000          LD      HL,0000h         ;Anfang der 3. Drittelzeile
336F         rgt_col EQU     $-2              ;je nach dem
3371 CD8233          CALL    prt_lin          ;ausdrucken
3374 226F33          LD      (rgt_col),HL     ;Zeiger nachstellen
3377 228133          LD      (sto_end),HL     ;s. o.
337A 21D133          LD      HL,lf            ;Codestring für CR mit LF
337D CDA233          CALL    print            ;ausgeben (auf nächste Druckzeile)
3380 18CB            JR      prt_lop          ;weiterdrucken

     ;eine Drittelzeile mit Steuercodes ausdrucken
3382 E5      prt_lin PUSH    HL               ;Textzeiger retten
3383 010000          LD      BC,0000h         ;Zeilenzähler
3384         lin_cnt EQU     $-2              ;je nach dem
3386 0B              DEC     BC               ;herunterzählen
3387 ED438433        LD      (lin_cnt),BC     ;neu laden
338B 78              LD      A,B              ;ist das Ende erreicht?
338C B1              OR      C
338D E1              POP     HL
338E 281C            JR      Z,end_lin        ;falls ja
3390 1A              LD      A,(DE)           ;Stelle des linken Randes
3391 32DC33          LD      (l_marg),A       ;in den Codestring der linken Randeinst.
3394 13              INC     DE               ;Zeiger weiterstellen
3395 1A              LD      A,(DE)           ;Stelle des rechten Randes
3396 32D933          LD      (r_marg),A       ;in den Codestring der rechten Randeinst.
3399 13              INC     DE               ;Zeiger weiterstellen
339A E5              PUSH    HL               ;Textzeiger retten
339B 21D733          LD      HL,mrg_str       ;Codestring für beide Ränder
339E CDA233          CALL    print            ;ausgeben
33A1 E1              POP     HL               ;Textzeiger

     ;einen String bis einschl. zum nächsten CR ausdrucken
33A2 7E      print   LD      A,(HL)           ;Druck-Byte laden
33A3 23              INC     HL               ;Zeiger nachstellen
33A4 CDB405          CALL    05b4h            ;Zeichen auf den Drucker ausgeben
33A7 FE0D            CP      0dh              ;war es ein CR?
33A9 C8              RET     Z                ;fertig, falls ja
33AA 18F6            JR      print            ;sonst String weiter ausdrucken

     ;Rückkehr aus der Druckschleife, wenn der Zeilenzähler abgelaufen ist:
33AC AF      end_lin XOR     A                ;A <- 00: auf Bank 0
33AD D3F3            OUT     (0f3h),A         ;umschalten
33AF F1              POP     AF               ;Stack bereinigen (CALL-Ebene löschen
33B0 210000          LD      HL,0000h         ;Textzeiger auf Seitenende
```

```
33B1                sto_end EQU   $-2              ;wird jeweils gepatcht
33B3  221E33                LD    (tx_beg1),HL     ;moment. Textzeiger als neuen Start laden
33B6  222133                LD    (tx_beg2),HL
33B9  C9                    RET

                    ;der Beginn einer Drittelzeile aufsuchen
33BA  C5            seek_ln PUSH  BC               ;Zeilenzähler retten
33BB  47                    LD    B,A              ;BC (- Suchstrecke min. 0D00 (lang genug)
33BC  EDB1                  CPIR                   ;CR aufsuchen
33BE  C1                    POP   BC               ;Zeilenzähler
33BF  10F9                  DJNZ  seek_ln          ;bis der Zeiger B Zeilen weiter steht
33C1  C9                    RET

                    ;-----------------------------------------------------------------

                    ;verschiedene Drucker-Steuersequenzen und Randeinstellungen

                    ;Druckerinitialisierung mit Reset, vollem Randausgleich, skip over perf.,
                    ;Proportionalschrift, CR ohne LF (beim Label no_lf)
33C2  1C            lp_init DB    1ch,'$',1bh,'a',03h,1bh,'N',07h,1bh,'p',01h
                    ;Anm. für NEC P7: Proportionalschrift wird mit  ' , ' ,00h angewählt

                    ;weitere Drucker-Steuersequenzen und linke und rechte Randspalten
33CD  1B            no_lf   DB    1bh,'3',00h,0dh ;     "    "     CR ohne LF
33D1  1B            lf      DB    1bh,'1',00h,1bh,'2',0dh ;Codestr. für l. Rand = 0 & 6 lpi
33D7  1B            mrg_str DB    1bh,'Q',00h      ;     "    "     rechte Randeinst.
33D9                r_marg  EQU   $-1              ;hier Zeichenspalte des rechten Randes
33DA  1B                    DB    1bh,'l',00h,0dh ;Codestring für linke Randeinstellung
33DC                l_marg  EQU   $-2              ;hier Zeichenspalte des linken Randes
33DE  07            margins DB    7,29,30,52,53,75 ;li. und re. Ränder der Drittelzeilen
                    ;                              ;(je nach geschmacklichen Krit. ändern)

                    ;-----------------------------------------------------------------

10B8                        ORG   428C             ;im Sektorpuffer, wo es nicht stört

                    ;beim Einsprung: entweder Kuckucksei legen oder Programm deaktivieren
10B8  CDD54C        entry   CALL  4cd5h            ;prüfen, ob dem Programmnamen etwas folgt
10BB  CAED32                JP    Z,start          ;ab die Post, falls nein
10BE  21BE45                LD    HL,45beh         ;sonst Fortsetzungsadr. der INT-Routine
10C1  220146                LD    (4601h),HL       ;diese restaurieren
10C4  3E32                  LD    A,32h            ;Opcode LD (nn),A
10C6  21E837                LD    HL,37e8h         ;memory mapped Adresse des Druckers
10C9  C3F832                JP    restore          ;Dr.-Umleitung rückgängig machen, zurück

                    ;-----------------------------------------------------------------

10B8                        END   entry            ;dort Einsprung

00000 Fehler

buff     3260   byte_ok 326F   cnt_ok  3284   div_don 3311   end_lin 33AC   end_lop 328D
entry    10B8   ign_res 3334   int     328B   l_marg  33DC   lf      33D1   lft_col 3357
lin_cnt  3384   ln_cnt1 333E   ln_cnt2 3346   lp_init 33C2   lst_pag 32E4   margins 33DE
mid_col  3363   mrg_str 33D7   no_lf   33CD   one_pag 32FF   pag_lop 32D7   point   3270
print    33A2   prt_lin 3382   prt_lop 334D   r_marg  33D9   restore 32F8   rgt_col 336F
seek_ln  33BA   sr_buff 32EB   start   32ED   sto_end 33B1   strt_ok 3336   tx_beg1 331E
tx_beg2  3321   txt_buf 4C00
```

„Ich glaube, du hast recht: Es ist ein Kuckuck!"

HEFT **21** September 1987

CMD ==> COM ohne CPMac

Nachdem mein eigenes CP/M fertig war, wollte ich auch drei Programme, die ich mit ZEUS erstellt hatte, ohne große Änderungen ins CP/M übernehmen. Dazu schrieb ich mir ein Programm, das ich hier vorstellen möchte. Es ist in der Diskothek erhältlich.

Eines muß ich noch vorweg sagen: ich habe nichts gegen CPMac. Das Programm scheint echt top zu sein, aber mir ist alles suspekt, was ich bezahlen soll. Wenn ich mein Geld sparen kann, tue ich das auch.

Bei der Übertragung tun sich mehrere Probleme auf. Das erste lautet, daß ein für CP/M erstelltes Programm bei 100h beginnen muß, während im Newdos erst ab 5200h freies RAM zu finden ist. Da das Programm vor der Übertragung in CP/M ins RAM per "LOAD" geladen werden muß, darf es nicht unterhalb 5200h stehen. Ich habe mir 6000h als Basisadresse ausgesucht.

Nun kann man im Source die ersten zwei Zeilen so schreiben:

```
        ORG     6000h
offset  equ     $-100h
```

und dann bei jedem Auftauchen einer absoluten (nicht relativen wie bei JR) Adresse, dieses Label "offset" wieder davon abziehen. Das sieht dann zum Beispiel so aus:

```
...
        JP      start-offset
...
start   LD      HL,meldung-offset
...
```

Allerdings vergesse ich leicht mal ein Offset, was der Assembler nicht als Fehler meldet, aber meist zum Programmabsturz führt.

Also ziehe ich eine andere Möglichkeit vor: die erste Zeile meines Programmslautet:

```
        ORG     100h
```

also ganz normal für ein CP/M-Programm. Nur kann dieses Programm nach dem Assemblieren nicht geladen werden, sonst kommt irgendeine Fehlermeldung oder ein Super-Absturz.

Mein Programm "shift/cmd" arbeitet nun in mehreren Phasen. Phase 1 lädt mein CP/M (das als GDOS-CMD-File vorliegt) in das obere RAM. Eigentlich brauche ich nur das BIOS, wie Ihr gleich sehen werdet, aber das hatte ich gerade nicht einzeln rumliegen.

Am Anfang von Phase 2 fragt "shift" nach dem zu übertragenden Programm. Dieses wird nun in einen Buffer bei 6000h geladen. Dabei tut "shift" einfach so, als wäre 6000h gleichbedeutend mit 100h, denn so lautet die Ladeadresse des Programms (sonst ist es nicht unter CP/M lauffähig).

In Phase 3 wartet "shift" nun darauf, daß eine CP/M-Systemdiskette in Laufwerk 0 gepackt wird. Dann schaltet es um auf durchgehendes RAM und schiebt das bei 6000h stehende Programm in seinen endgültigen Speicherbereich bei 100h (wo es auch eigentlich stehen/laufen sollte). Dann springt "shift" das BIOS an und führt somit einen Cold Start durch. Dabei wird der Speicherbereich von 100h bis zum Anfang des CCP nicht gelöscht und somit steht nach dem Cold Start das Programm vollständig in der TPA. Es kann nun mit "SAVE xx test.com" gesichert werden. xx ist dabei die Länge des Programms in 256-Byte-Sektoren, die "shift" auch nochmal meldet, damit man möglichst nichts falsch machen kann.

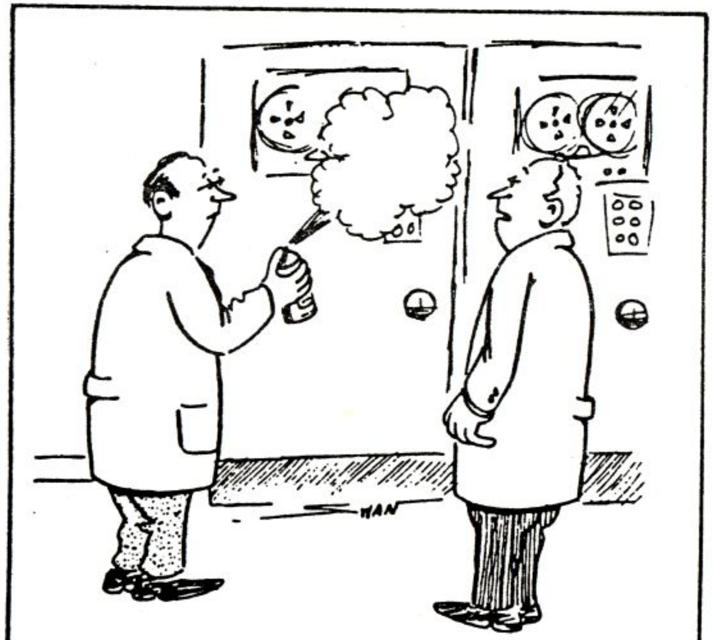
Natürlich hat dieses System gegenüber CPMac mehrere Schwächen:

1. Ihr habt kein cpm/cmd. Das braucht Ihr auch nicht, wenn Ihr nur ein BIOS habt (reicht, wenn dieses beim Cold Start sowohl CCP als auch BDOS lädt). Wenn das zu übertragende Programm nicht zu lang ist, kann man "shift" auch in einer Endlosschleife enden lassen und dann per RESET booten. Allerdings wird dann zumindest der Speicherbereich ab 4200h verändert (das Programm darf nicht länger als 65 Sektoren bzw. 16,25 KByte sein).

2. Das zu übertragende Programm ist zu lang. Da kann ich auch nicht helfen. Wendet Euch an CPMac. Allerdings ist es sowieso nicht sinnvoll, große Programme für CP/M unter ZEUS zu erstellen. Das lohnt sich nur bei kleinen Sachen, bei denen es zu aufwendig ist, erst Wordstar und den ganzen Kram zu bemühen.

Für mich hat sich das Programm gelohnt, denn so brauche ich mir nicht CPMac anzuschaffen (um es dann dreimal im Jahr zu benutzen) und kann trotzdem schnell mal was mit ZEUS programmieren und unter CP/M laufen lassen.

Gerald Schröder



Sagen Sie mal Herr Haferstroh,wo haben Sie denn gelern wie man ein Programm 'debuged'?

Für den Heimwerker: Wie baue ich mir ein CP/M?

Dieser Artikel ist nicht ...
... eine Einführung in CP/M
... eine Beschreibung von CP/M
... eine konkrete Bauanleitung für ein CP/M

sondern eine allgemeine Beschreibung, wie man sich ein CP/M aus Newdos heraus bauen kann bzw. wie ich dieses gemacht habe. Wenn ich Euch genau schildere, wie es aussieht, bringt Euch das fast nichts, denn mein Genie IIs läuft anders als Eure Rechner. Interessanter ist m.E. viel mehr der Weg zu einem eigenen CP/M. Der Artikel richtet sich an Leute, die kein CP/M haben und solche, denen ihr CP/M nicht ausreicht. Assembler-Kenntnisse sind mitzubringen.

1. Die Werkzeuge

Sie sind in einem anderen Artikel beschrieben worden. Alle Sources liegen unter ZEUS vor. Einige Hilfsprogramme, die ich selbst geschrieben habe, werde ich gesondert vorstellen.

2. Die Theorie des CP/M

Hinter CP/M steht eine ziemlich ausgefeilte Philosophie, die es ermöglicht, daß es auf möglichst vielen Rechnern läuft. Bevor man sich mit dem Bau (der Praxis) beschäftigt, muß erst die Theorie bekannt sein. Dazu gibt es zwei Wege: a) den Lehrer, b) das Lehrbuch.

Ich empfehle die erste Möglichkeit. Sie sieht folgendermaßen aus: Ihr sucht Euch einen Typen, der möglichst in der Nähe wohnt und außerdem etwas Ahnung von CP/M hat. Solche Leute wie Hartmut sind ein gutes Ziel. Nun beginnt die "Löcherphase". Ihr bohrt in (bzw. an) ihm rum, bis er Euch eine kostenlose Einführung in CP/M angedeihen läßt. Oft sind solche Leute auch froh, mal ihr Wissen präsentieren zu können, so daß die Löcher flach sein können. Aber den Bohrer noch nicht absetzen! Nun muß nämlich auch noch das letzte an Substanz aus dem Loch geholt werden. Entweder kann der Typ so gut erklären, daß Du danach besser Bescheid weißt als die Herren von Digital Research oder Du stehst so lange vor seiner Haustür, bis er Dich mit allen Materialien zu CP/M (Büchern, Fotokopien, Zeitschriften, Sources) bewirft, die er sowieso schon lange loswerden wollte. Die sammelst Du schön säuberlich auf und studierst im kleinen Kämmerchen weiter, bis Du oberschlau bist.

Die zweite Möglichkeit ist der zweite Teil der ersten: Du studierst von vornherein nur alles, was Du Dir zusammenschnorren oder Dein Geldbeutel sich leisten kann. Sobald Du weißt, was CCP, BDOS, BIOS, Sprungleiste (JP-Vektoren, BIOS-Aufrufe), DPH, DPB, XLT (und was es sonst noch an tollen Abkürzungen gibt) sind, bist Du für den nächsten Schritt gerüstet.

3. Die Ausgangslage

Beim Sichten der verfügbaren (zusammengeschnorrten, hinterhergeworfenen) Materialien wirst Du zweierlei feststellen: erstens geht jedermann davon aus, daß Du ein CP/M benutzt, wenn Du eines bauen willst; zweitens sind die meisten Beispiele entweder in 8080-Assembler oder unvollständig oder für ein 8-Zoll-SS/SD-Diskettensystem oder gar nicht vorhanden. Aber was hast Du? Ein Newdos, eine zusammengewürfelte (hochwertige) Hardware, einen Z80-Assembler, kein lauffähiges CP/M (willst Du ja erst bauen), keine Ahnung, aber große Pläne.

Was brauchst Du? Ein CP/M bestehend aus CCP und BDOS unter Newdos (als Source-Code); ein möglichst gut dokumentiertes Beispiel-BIOS (Dein eigenes mußt Du Dir bauen); ein Format-Programm; ein Programm, das Dein CP/M als System auf eine Diskette packt.

4. Wie bekommst Du was?

Fangen wir ganz vorne an. Die Paperware sollten Dir Club-Kameraden zur Verfügung stellen können. Eine gute Adresse wird hoffentlich bald unsere Club-Bibliothek sein.

Das CP/M ist ein Problem, denn Digital Research hat das Copyright. Allerdings gibt es schon Public-Domain-CP/Ms (z.B. für Atari) und ein komplettes Listing (Röckrath), so daß diese Sache wohl nicht mehr so heiß gegessen wird (wenn unser aller Freund Schmidt nicht die Rechte erwirbt). Am besten holst Du es Dir von jemanden, der es Dir unter Newdos geben kann. Bei der Source-Anpassung, damit es relokatibel wird (also überall laufen kann), kann ich helfen.

Dann wäre ein kommentiertes BIOS für eine möglichst ähnliche Maschine zu besorgen. Empfehlen kann ich das Montezuma-BIOS für das Modell IV bzw. die Dokumentation dazu, die ein fast vollständiges Listing des BIOS enthält. Die fehlenden Teile kann ich liefern (auch mit Kommentar). Außerdem wird das von mir erstellte BIOS in der Diskothek verfügbar sein.

Das Formatierungs- und das Systemgenerierungs-Programm müssen neu programmiert werden. Beispiele werden in der Diskothek zu finden sein (format/cmd und putsys/cmd).

Natürlich solltet Ihr die Hardware Eures Rechners gut kennen und gute Unterlagen darüber besitzen. Geeignet für das Schreiben der Disketten-Ein-/Ausgabe ist zum Beispiel das DOS-Buch von Grosser.

5. Die Grundgedanken

Das CP/M-System soll vollkommen vom Newdos aus hergestellt werden, denn das ist unsere "normale" Systemoberfläche, deren Werkzeuge wir auch am besten kennen.

Die Reihenfolge der Programmierung sieht folgendermaßen aus: zuerst das BIOS erstellen, dann eine Diskette mit dem eigenen Format-Programm formatieren und zuletzt das ganze CP/M-System (Bootsektor, CCP, BDOS, BIOS) auf diese Diskette packen, die dann eine bootbare CP/M-Systemdiskette sein wird.

Das BIOS ist einigermaßen umfangreich und sollte möglichst sicher funktionieren, denn es ist als ganzes nur schwer zu testen. Deshalb halte ich die Programmier-Methodik der Montezuma-Hersteller für unerläßlich: das Programmieren in Moduln. Das heißt, daß das BIOS in mehrere Teile (z.B. Bildschirmausgabe) aufgespalten wird, die Ihr nun getrennt programmiert und testet.

Erst wenn alle (oder möglichst viele) Moduln richtig funktionieren, macht Ihr Euch an die Programmierung des Format-Programms, denn Ihr könnt viel von dem Wissen, daß Ihr vorher gesammelt habt, und auch mindestens eines der Moduln (Disk-Ein-/Ausgabe) bei diesem Programm verwenden. Ebenso steht es mit dem Systemgenerierungsprogramm.

6. Die Moduln

Von Montezuma und mir sind die Namen vieler Moduln und auch eine Reihe von Labels vorgegeben, die Ihr weiter verwenden solltet, denn damit spart Ihr Euch eine Menge Arbeit. Ein großer Vorteil der Modul-Programmierung ist, daß Ihr langsam vorgehen und die Anforderungen dadurch in Grenzen halten könnt. Es steigert unheimlich die Motivation, wenn das erste Modul funktioniert. Das Druckerausgabe-Modul dürfte das einfachste sein, also fangt damit an. Dann geht Ihr zur Bildschirmausgabe über usw. Als letztes nehmt Ihr Euch die Disketten-Ein-/Ausgabe vor. Sie dürfte am schwersten sein. Allerdings solltet Ihr Euch jetzt überlegen, was Euer CP/M können soll, d.h. welche exotischen Diskettenformate von ihm gelesen werden sollen. Je nachdem ist mehr oder weniger Aufwand an dieser Stelle erforderlich. Mein CP/M-BIOS kann erheblich weniger als das von Montezuma, aber der größte Teil davon ist für mich einfach überflüssig. Zum Beispiel hat mein System nur zwei Diskettenlaufwerke und wird wohl nie eine Festplatte bekommen, warum also mehr als zwei Laufwerke berücksichtigen? An anderer Stelle habe ich ebenso auf die Implementierung des IO-Byte verzichtet, das zwar sehr schön sein mag, mir aber nichts bringt. Viele Sachen sind am Anfang einfach auch unsinnig und können bequem von Euch nachgerüstet werden, wenn die Version 1.0 Eures CP/M einmal läuft.

Bei der Programmierung und vor allem dem Test der einzelnen Moduln müßt Ihr darauf achten, welche Variablen und Routinen von zwei Teilen gemeinsam benutzt werden und diese Labels dann irgendwie dem gerade in Arbeit befindlichen Modul bekanntgeben (z.B. eine Routine, die nur aus einem RET bsteht, beim Test einbauen). Jedes Modul wird einzeln getestet. Dazu schreibt Ihr kurze Test-Programme, in die Ihr die Routinen einbindet und dann prüft, ob das richtige herauskommt, wenn Ihr etwas bestimmtes eingebt. So läßt sich zum Beispiel hervorragend testen, ob die Tastatur-Abfrage (conin) die richtigen Werte ergibt oder die Bildschirmsteuercodes auf dem Bildschirm die richtigen Reaktionen hervorrufen (conout).

Und gleich noch ein Vorteil: wenn Ihr schon irgendwo bestimmte Programmteile habt, die sich verwenden lassen, baut Ihr sie einfach als Modul ein! Wiederum könnt Ihr viele der BIOS-Moduln in anderen Programmen verwenden, z.B. die Disketten-Ein-/Ausgabe für format/cmd und putsys/cmd.

7. Das Linken der Moduln

Diese Sache sollte nicht zu leicht genommen werden! ZEUS hat hervorragende Fähigkeiten zum Linken. Ihr schreibt einfach einen Hauptfile (bei mir heißt er master/src), der alle anderen Bestandteile per GET-Kommando holt. Naturgemäß werdet Ihr dabei feststellen, daß irgendwelche Labels nicht definiert wurden und deshalb geändert, eingefügt oder sonstwas werden müssen. Dann müßt Ihr erkunden, wie groß Euer BIOS im Endeffekt ist und je nachdem die Basisadresse des CP/M nach unten verschieben. Auch kein größeres Problem: in master/cmd fügt Ihr einfach eine EQU $-Variable direkt vor dem END ein, deren Wert FF00h nicht übersteigen darf (der Rest ist Stack). Den Wert der Variablen könnt Ihr Euch per Kommando "Blabel" ansehen.

Nachdem das System nun vollständig in den Speicher paßt und keine Assemblierungsfehler mehr auftreten, könnt Ihr den File cpm/cmd erzeugen. Das dauert zwar etwas, aber jetzt habt Ihr Euer CP/M!

8. Das Programm format/cmd

Dieses Programm soll eine Diskette für Euch formatieren, und zwar in dem Format, das Ihr für Euer System bevorzugt. Die Format-Routine des Newdos ist nicht geeignet, denn wir brauchen ja kein Directory und außerdem sind 256-Byte-Sektoren nicht unbedingt das Optimum. Und Ihr solltet noch etwas weiter denken: Euer CP/M soll später auch mal Disketten formatieren können, also schreibt Ihr hier ein Programm, das sich später leicht auf CP/M übertragen lassen sollte, um für Euch dort die Disketten zu formatieren.

Beim Formatieren solltet Ihr eins beachten: das Füllbyte sollte (muß?) E5h sein. Ich habe es mit dem DD-Füllbyte 6DB6h versucht, was zur Folge hatte, daß mein CP/M später meldete "No file" (im Directory, ist OK), aber: "Diskette full". Eine volle Diskette ohne einen einzigen File drauf gibt es nicht. Es liegt allein am Füllbyte (schätzungsweise Bit 7 oder so, fragt jemand, der sich mit CP/M auskennt).

9. Das Programm putsys/cmd

Wenn Ihr eine formatierte Diskette und cpm/cmd habt, sollte irgendwie cpm/cmd bootfähig auf die Diskette gepackt werden. Dies erledigt putsys/cmd. Allerdings müßt Ihr dazu noch ein Modul erstellen, und zwar den Bootsektor. Das Ding sitzt auf Track 0, Sektor 0 (manchmal auch 1) und wird von Eurem ROM nach 4200h gezogen, dort gestartet und soll dann Euer CP/M irgendwie laden und zum Laufen bewegen. Es kann sein, daß Ihr dazu Track 0 nochmal neu formatieren müßt, wenn Euer ROM sonst Mist baut. Mein IIs liest jedenfalls alles, was man ihm anbietet, nur muß ein bestimmtes Byte den Wert 00 haben. Bootsektoren gibt's wieder bei Grosser zur Ansicht. Ansonsten wäre eine Newdos-Systemdiskette auch eine gute Informationsquelle.

Bei meinem CP/M lädt der Bootsektor das BIOS an den richtigen Platz und startet dieses. Das BIOS wiederum lädt erst CCP und BDOS und springt dann den CCP an. Das könnt Ihr auch anders lösen, aber bei mir bedeutet das, daß auf der Systemdiskette stehen müssen:

Track 0, Seite 0, Sektor 0: Bootsektor
Track 0, Seite 0, Sektor 1-3: BIOS
Track 0, Seite 0, Sektor 4, und Seite 1, Sektor 0-5: CCP und BDOS.

Dabei solltet Ihr bedenken, daß jeder Sektor 1024 (!) Byte lang ist.

Bei Euch kann das ganz anders aussehen, aber ich habe gute Erfahrungen mit dieser Anordnung gemacht.

putsys/cmd funktioniert bei mir folgendermaßen: zuerst nulle ich den Speicher von 5200h aufwärts (Befehl "0" im Gdos). Dann lade ich das CP/M: LOAD CPM/CMD. Nun wird putsys/cmd gestartet: PUTSYS/CMD. putsys wartet darauf, daß ich die formatierte CP/M-Diskette in Lw. 0 einlege, schreibt dann Bootsektor, BIOS, CCP und BDOS auf die Diskette und startet zuletzt das BIOS (oder wartet auf die Newdos-Diskette oder auf RESET).

Eine abgeänderte Form des putsys/cmd braucht Ihr übrigens wieder unter CP/M, denn Ihr wollt ja nicht alle Systemdisketten vom Newdos aus erzeugen.

10. Die letzten Tests

Die letzten Tests sind eigentlich nie zu Ende, denn irgendwo taucht immer was auf, das Euch nicht gefällt. Bei mir dauerte es drei Tage, bevor ich die erste CP/M-Diskette booten konnte, nachdem das System fertiggestellt war, denn ein Befehl der Cold-Start-Routine war falsch. Also nicht verzagen, wenn nicht alles gleich funktioniert.

Weitere Feinarbeiten werden sich enlos hinziehen, aber wenigstens könnt Ihr sie durchführen, während Ihr Euch bei anderen (gekauften) Systemen machtlos mit den Fehlern rumärgern müßt!

11. Was unter CP/M als erstes zu tun ist

Drei Programme hängen unter CP/M allein von Eurer Hardware ab und müssen deshalb möglichst bald erstellt werden. a) format.com, b) sysgen.com, c) pdrive.com.

format.com leistet dasselbe wie format/cmd, nur unter CP/M. Es formatiert eine Diskette; wenn Ihr Lust habt, auch in den blödesten Formaten (35 Tracks, SS/SD, 128 Bytes/Sektor).

sysgen.com ähnelt stark dem putsys/cmd, sollte aber etwas bearbeitet werden. Im Grunde könnt Ihr zwar das "laufende" CP/M-System auf die Diskette dumpen (natürlich mit Bootsektor), aber ich halte es für besser, das System frisch von einer Systemdiskette zu laden und dann auf eine neu formatierte Diskette zu feuern. TCS formatiert übrigens von putsys.com aus nochmal die ersten beiden (System-) Tracks in einem Standard-Format. Dann können die exotischsten Disketten mit einem System ausgerüstet werden, aber wer will das? Mir reicht ein Format.

pdrive.com ist vollkommen neu zu erstellen. Es soll Euch helfen, Disketten in anderen Formaten zu lesen. Theoretisch könnt Ihr die Werte auch per DDT oder so ändern, aber viel schöner ist es doch, wenn Ihr nur gefragt werdet, wieviele Tracks usw. die zu lesende Diskette hat und den Rest Euer Programm erledigt. Eine Rumpf-Version findet Ihr in der Diskothek. Sie ist nicht sehr komfortabel, aber wenn ich mal gut drauf bin, werde ich sie verbessern.

Wie Ihr Programme für CP/M mit ZEUS erstellt und dann unter CP/M laufen laßt, erzähle ich Euch in einem anderen Artikel.

12. Schlußwort

Das war ein verdammt langer Artikel, wenn er auch nicht an Arnulfs und Helmuts Produktionen heranreicht (vor allem in der Tiefe). Wenn Ihr Euch bis hierhin durchgekämpft habt und irgendwann mal ein CP/M bauen wollt, stellen sich Euch vielleicht Probleme, die ich nicht vorausgesehen und nicht angesprochen habe. Schreibt mir dann einfach und ich werde versuchen, Euch zu helfen.

Gerald Schröder

MS-DOS Ecke

Nachdem eine rege Diskussion offensichtlich gewünscht wird und im Gange ist, möchte ich auch noch einige Worte hierzu sagen. Über MS-DOS selbst braucht man nicht viel sagen, Höhen und Tiefen sind hinreichend bekannt.
Zur Frage, ob eine MS-DOS Ecke im CLUB80-Info erwünscht ist, sollte man einerseits sehen, daß es bereits zahlreiche Zeitschriften mit MS-DOS Veröffentlichungen gibt, von den Sachbüchern ganz zu schweigen. Ist dann der Platz hierfür in unserem Info nicht zu schade, zumal sicherlich viele der Clubmitglieder auch andere Zeitschriften lesen, in denen MS-DOS Probleme angesprochen werden ? Wenn nun eine MS-DOS Ecke eingerichtet wird, kommt dann auch bald eine UNIX, OS/2 oder sonstige Ecke, die eigentlich mit TANDY/GENIE nichts gemein hat ? Ich finde diesbezüglich eben, daß wir darauf achten sollten, die "80" in unserem Clubnamen nicht zu vergessen. Das Ziel sollte meiner Meinung nach sein, daß es im CLUB80 auch in einigen Jahren noch Anregungen und Tips für TANDY-Anwender geben sollte.
Auf der anderen Seite gibt es zwischenzeitlich doch einige Verbindungen zwischen MS-DOS und TANDY/GENIE-Computer, z.B. das im letzten Info bereits angesprochene TurboPascal, Datentransferprogramme , Model 4-Emulator für MS-DOS etc. Und sicher gibt es auch einige Clubmitglieder die mit beiden Systemen arbeiten.
Eine MS-DOS-Ecke könnte deshalb schon ein Plätzchen im Info finden. Die Beiträge sollten sich dann aber auf Gemeinsamkeiten beziehen und nicht etwa Einführungen oder Tips für MS-DOS beinhalten.
Wir sollten auf jeden Fall darauf achten, daß wir nicht dem Beispiel von Computerzeitschriften folgen, die 80er Anwendungen so nach und nach verdrängten.

Klaus Hermann

Model 4 - Info

Nachstehend ein kleiner Patch, der bewirkt, daß bereits vom Systemstart an der Großbuchstaben-Modus eingeschaltet ist. Eine manuelle Umschaltung per CAPS-Taste enfällt somit.

PATCH SYS0/SYS.LSIDOS (D00,80=20:F00,80=00)

Klaus Hermann

Turbo Tip

Viele Grafikkarten (so auch die HRG des TRS 80 Model 4) erlauben es, den Textbildschirm ganz abzuschalten bzw. auszublenden. Tritt bei der Erstellung von Grafikprogrammen unter Turbo Pascal ein Runtime-Error auf nachdem die HRG ein- und der Textbildschirm ausgeschaltet wurde, steht man im wahrsten Sinne des Wortes im Dunkeln, da Turbo den Textbildschirm nicht automatisch wieder einblendet. Mit folgendem Patch kann man Turbo Pascal dazu bewegen, dies doch zu tun!

Ab Adresse 016Bh* steht der Initialisierungsstring, den man bei der Installation von Turbo Pascal angeben kann. Dieser String wird unter anderem jedesmal dann an das Terminal geschickt, wenn Turbo nach einem Laufzeitfehler aus dem laufenden Programm zurückkehrt. Die Routine, die das besorgt, steht ab 030Ah und beginnt mit LD HL,016Bh. Diese Anweisung ersetzt man durch einen CALL xxxx, wobei xxxx die Adresse der eigenen Routine zur Umschaltung auf den Textbildschirm ist. Platz für die Routine findet man z.B. bei 2174h (Copyrighteintrag von Borland), wo auch Jochen Tucht schon seine Patchs untergebracht hat (siehe PASCAL international Heft 7/87). Beim Model 4p sieht die Routine wie folgt aus:

```
LD    A,0FCh
OUT   (83h),A
LD    HL,016Bh
RET
```

Wer aufgepaßt hat, dem ist sicher aufgefallen, daß die vorletzte Instruktion genau der entspricht, die man bei 030Ah durch den CALL auf die eigene Routine überschrieben hat. Während das Unterprogramm von Rechner zu Rechner unterschiedlich sein kann, muß die letzte Aktion vor der Rückkehr (RET) ins Hauptprogramm immer die gleiche sein, sonst passieren (je nach dem Wert von HL) unvorhersehbare Dinge!

Daß die Änderung am besten mit einem Debugger durchgeführt wird, mag eingefleischte Hochsprachenprogrammierer vielleicht abschrecken, trotzdem wünsche ich viel Spaß und Erfolg beim Erstellen von Grafiken mit Turbo Pascal!

Hartmut Obermann

*Alle gemachten Angaben beziehen sich auf Turbo Pascal 3.0 unter CP/M!

Hier ein kleines Programm, mit dem sich Turbo-PASCAL selbst patcht!

```
const
  a: array [1..8] of byte =
   ($3E,$FC,$D3,$83,$21,$6B,$01,$C9);
   (LD    A,252
    OUT   (131),A
    LD    HL,016Bh
    RET)

begin
  move(a,mem[$21AB],8);
  mem[$030A]:=$CD; (CALL   21ABh)
  mem[$030B]:=$AB;
  mem[$030C]:=$21;
end.
```


Turbo Patch

In der Zeitschrift PASCAL international Heft 5 und 7/87 stellte ein Herr Tucht zwei Patchs vor, die dem Turbo-PASCAL-Compiler zu noch mehr Benutzerfreundlichkeit verhelfen sollen. Leider sind Herrn Tucht ein paar kleine aber fatale Fehler unterlaufen, die ich in dem folgenden Patchprogramm beseitigt habe.

```
(Patch's fuer Turbo-Pascal 2.0 und 3.0 aus PASCAL international 5 und 7/87}
(Patch 1 dient dazu, Turbo-Pascal durch R.COM wieder aufrufbar zu machen, }
(ohne den gerade editierten Text zu verlieren. Patch 2 stellt zusaetzlich }
(noch die Moeglichkeit zur Verfuegung, den Namen des zu editierenden Files}
(schon beim Aufruf von Turbo mit anzugeben (z.B. TURBO TEST.PAS)          }
(Vorgehensweise Patch 1: TURBO mit oder ohne Fehlermeldungen starten      }
{                        TP-PATCH.PAS laden                               }
{                        nicht benoetigte Zeilen loeschen oder einklammern}
{                        TP-PATCH mit R-Option starten                    }
{                        W-Option aufrufen und auf die Frage nach dem neu-}
{                        en Workfile mit ENTER antworten                  }
{                        TURBO mit Q verlassen und wie folgt abspeichern: }
{                        TURBO 2.0 ohne Fehlermeldungen - SAVE 122 T2.COM }
{                        TURBO 2.0 mit Fehlermeldungen  - SAVE 127 T2.COM }
{                        TURBO 3.0 ohne Fehlermeldungen - SAVE 123 T3.COM }
{                        TURBO 3.0 mit Fehlermeldungen  - SAVE 129 T3.COM }
(Vorgehensweise Patch 2: genau wie bei Patch 1! Es ist zu beachten, dass  }
{                        Patch 2 an einem noch nicht mit Patch 1 behandel-}
{                        ten TURBO vorgenommen werden muss. Wurde TURBO   }
{                        schon mit Patch 2 behandelt, so ist die Funktion }
{                        von Patch 1 automatisch schon enthalten!!!       }
(Patch 1 fuer TURBO 2.0)
BEGIN
  MEM [$205D]:=$C3;
  MEM [$205E]:=$38;
  MEM [$205F]:=$21;
END.
(Patch 1 fuer TURBO 3.0)
BEGIN
  MEM [$2174]:=$C3;
  MEM [$2175]:=$38;
  MEM [$2176]:=$22;
END.
(Patch 2 fuer TURBO 2.0)
const
  a : array [1..45] of byte =
   ($06,$0B,$21,$6B,$00,$2B,$7E,$FE,$3F,
    $28,$1F,$10,$FB,$FE,$20,$28,$19,$21,
    $65,$00,$7E,$FE,$20,$20,$0B,$3E,$50,
    $77,$23,$3E,$41,$77,$23,$3E,$53,$77,
    $CD,$5B,$2C,$CD,$01,$24,$C3,$35,$21);
begin
  move(a,mem[$205D],45)
end.
(Patch 2 fuer TURBO 3.0)
(Geaendert am 16.07.87 von Hartmut Obermann)
(Der Fehler bei der Option X "(.OVR not found" und das Nachladen)
(der schon in TURBO.COM enthaltenen Fehlermeldungen wurde beseitigt!)
const
  a : array [1..53] of byte =
   ($06,$0B,$21,$6B,$00,$2B,$7E,$FE,$3F,
    $28,$1F,$10,$FB,$FE,$20,$28,$19,$21,
    $65,$00,$7E,$FE,$20,$20,$0B,$3E,$50,
    $77,$23,$3E,$41,$77,$23,$3E,$53,$77,
    $CD,$3A,$2D,$CD,$F9,$24,$C3,$38,$22,
    $54,$55,$52,$42,$4F,$20,$56,$33);
begin
  move(a,mem[$2174],53);
  mem[$2637]:=$a1;
  mem[$2dab]:=$a1;
  mem[$2da4]:=$c9;
end.
```

dBase II

Eine kurze Einleitung in das Datenbanksystem

v.Harald Hand

Bereits vor 2 Jahren machte ich Bekanntschaft mit diesem Datenbanksystem. In unserer Firma sollte ein Verwaltungssystem für Meßmittel auf einem PC erstellt werden. Ein Informatiker erstellte das entsprechende Programm mit dBase II. Als ich mich näher damit befaßte, stellte ich fest, dieses Datenbanksystem ist äußerst leistungsfähig und eignet sich für viele Verwaltungsaufgaben. So habe ich selbst angefangen Programme in dBase II zu schreiben und im Laufe der Zeit entstanden eine Menge Programme. Das letzte Programm wurde auf meinem Homecomputer (Tandy Modell 4P) geschrieben und ist eine Vereinsverwaltung für einen Sportverein.

Was ist nun dBase II ? . Es ist ein rationales Datenbanksystem wo der Benutzer schon mit relativ wenig Anweisungen in der Lage ist, Daten zu erstellen und zu bearbeiten. Zur Erstellung von kpl. dBase Programmen ist es nur ein kleiner Schritt weiter.

1. Erstellung einer Datenbank :

Eine dBase II Datenbank ist leicht durch den Befehl :
" CREATE < Dateiname > "
zu erstellen. Es müßen dann jedoch noch folgende Dinge eingegeben werden : Name, Typ, Größe und Dezimalstellen.

Zur Bearbeitung wird eine dBase II Datenbank mit dem Befehl
" USE < Dateiname > " geöffnet.
Sehr oft arbeitet man bei dBase mit Indexdateien, da die Datenfelder hier sortiert gespeichert sind z.B. bei Adressendateien nach Nachname, Wohnort, Alter etc. In dBase können bis zu 7 Indexdateien verarbeitet werden, wodurch sich jedoch die Verarbeitungsgeschwindigkeit verkürzt. Wichtig bei Indexdateien ist, daß sie vor dem Aufruf reorganisiert sein muß, da es sonst zu Fehlmeldungen (" RECORD OUT OF RANGE ") kommt.

Dateistrukturen können leicht durch den Befehl :
" MODIFY STRUCTURE "
geändert werden. Jedoch Vorsicht ! Dieser Befehl löscht alle bereits gespeicherten Daten. Vor einer Strukturänderung sollten die Daten in eine Hilfsdatei gespeichert werden.

```
z.B. " USE < Dateiname > "
     " COPY TO HILF "        ===> Hilfsdatei
     " MODIFY STRUCTUR "    ====> Ändern der Struktur
     " APPEND FROM HILF "   ====> Rücktransfer
```

Eine Einsicht in die Datenbank bekommt man nach dem Aufruf des Dateinamens durch die Befehle :
" LIST ALL " (jedoch läuft hier der Bildschirm durch) oder
" DISPLAY ALL " (es wird nur eine Bildschirmseite eingeblendet)
Selbstverständlich können auch Daten in einer Datei sehr einfach geändert werden. Dieses erfolgt nach dem Aufruf des Programmnamens durch den Befehl :
" BROWSE "
Der Cursor steht jeweils auf dem Datensatz und alle Daten in der ganzen Datenbank können so leicht geändert werden. Der Befehl

" EDIT 100 "
erlaubt dagegen nur eine Änderung des entsprechenden Datensatzen (hier Satz 100).

Datensätze können mit dem Befehl :
" DELETE RECORD < Nr. > "
gelöscht werden. Dabei wird der angesprochene Datensatz jedoch noch nicht endgültig gelöscht, sondern nur zum Löschen markiert. Die tatsächliche Löschung erfolgt erst mit dem Befehl :
" PACK "
Dabei sollten allerdings mit der Datei auch die entsprechenden Indexdateien eröffnet sein, da diese aktualisiert werden müssen. Bei unterschiedlichem Aktualisierungsgrad von Datei und Indexdatei erfolgt sonst die Fehlmeldung :
" RECORD OUT OF RANGE "
Sollte diese Fehlermeldung auftauchen muß die Datei mit folgenden Befehl neu indiziert werden.
" USE < Dateiname > "
" INDEX ON NACHNAME TO NACHNAME " (Indexdatei ist nach Nachname geordnet)
Zum Löschen markierte Datensaätze können leicht mit dem Befehl :
" RECALL NEXT 1 "
wieder aktiviert werden.
Man sollte immer wieder prüfen, ob die Datensätze endgültig gelöscht werden müssen, oder noch an anderer Stelle gebraucht werden (z.Beispiel bei der Vereinsverwaltung für eine Kündigungsliste).
Komplette Dateien können mit dem Befehl :
" DELETE < dateiname > "
gelöscht werden.

2. Sortieren und Indizieren von dBase II - Dateien :

Der Befehl :
" INDEX ON NACHNAME TO NACHNAME " indiziert eine Datei nach dem angegebenen Datenfeld in die Datei Nachname und stellt die Datensätze nach nach diesem Feld sortiert zur Verarbeitung bereit.
Mit den Befehlen :
" GO BOTTOM und GO TOP
kann zu dem ersten und letzten Datensatz einer Datei gesprungen werden, anschließend können mit dem Befehl :
" SKIP-1 "
die DAtensätze ab dem letzten Datensatz solange angezeigt werden, bis mit der Recordnummer 0 der Anfang der Datei erreicht ist.

3. Beenden von dBase II :

DBase II wird verlassen durch den Befehl :
" QUIT "
und kehrt zum Betriebssystem zurück.

Das die erste kurze Einleitung in dBase II. Wir können jetzt eine Datenbank erstellen, verändern, Indexdateien eröffnen und Daten löschen. Für eine erste Einleitung soll es genug sein. Im nächsten Info geht es dann weiter mit der Erstellung von Programmen, Editierbefehlen, Programmschleifen, Bildschrim- und Druckersteuerungen usw.

Falls jemand sich näher damit auseinandersetzen will stehen ich gerne mit Rat und Tat zur Verfügung. Im Moment bin ich dabei mich mit dBase III plus auseinanderzusetzen, da ich in der Firma einen neuen Rechner (Siemens PC 16-20 = AT Kompatibel) habe.

Als Fachliteratur für dBase II kann ich folgende Bücher empfehlen :

1. Das Datenbanksystem dBase II v.Dr.Peter Albrecht
   (Verlag Markt und Technik)

und

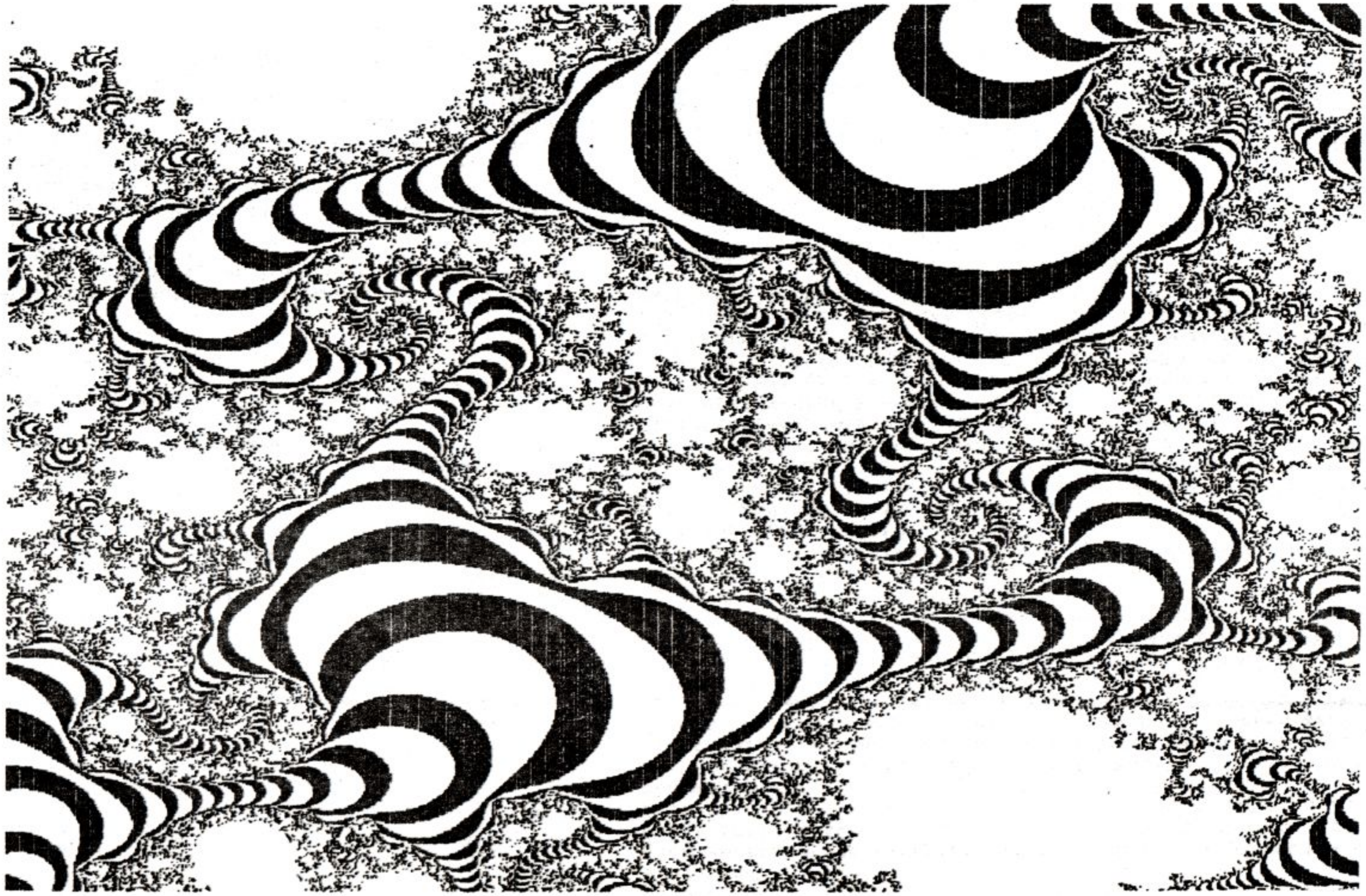
2. dBase II Anwendungen v. M.Apfelbeck
   (IWT-Verlag Vaterstetten b. München)

Die Preise liegen bei ca. 58,- DM.

So das war es fürs erste. Also bis zum nächsten Info.

Harald

Harald



Apfelmännchen auf der GDP64

Helmut Bernhardt

Da der Computer für mich zu 80% ein Spielzeug ist, weil ich keine elektrische Eisenbahn habe, an der ich rumbasteln kann, kommt es recht selten vor, daß ich damit ernsthafte Arbeiten erledige. Eine dieser Spielereien war das Aufbauen und auf Auslesbarkeit des Grafikspeichers Erweitern der GDP64-VideoKarte des NDR-Klein-Computers. Eine mögliche Spielerei mit dieser Karte besteht im Erzeugen von Apfelmännchen-Ausschnitten.

Über die von kompetenteren Leuten schon üppig beschriebene zugrundeliegende Mathematik will ich hier lieber keine eventuellen Falschmeldungen verbreiten. Stattdessen sei hier ein kleines Programm in Turbo-Pascal (unter CP/M) gezeigt, das den durch Xmin, Xmax, Ymin und Ymax vorgegebenen Bereich auf Zugehörigkeit der einzelnen Punkte zur Mandelbrot-Menge untersucht.

Ein Punkt gehört dann zur Mandelbrot-Menge, wenn die Iteration in der REPEAT-UNTIL-Schleife des Hauptprogramms Rmax-mal durchlaufen wird, ohne daß die Abbruchbedingung X*X +Y*Y > Kmax erfüllt wurde. In diesem Fall wird der durch Xi und Yi vorgegebene Punkt nicht gesetzt.

Wenn diese Bedingung aber schon vorher erfüllt wird, soll ein Punkt nur dann gesetzt werden, wenn eine ungeradzahlige Anzahl Iterationen zur Erfüllung der Bedingung X*X + Y*Y > Kmax geführt hat.

Da solche Berechnungen enorme Zeiten beanspruchen und man gewährleistet haben will, daß ein fertiggestelltes Bild zum Schluß auch gerettet wird, bevor der Kühlschrank den Computer in den Wald schickt und alles umsonst war, wird das Bild automatisch ausgelesen und auf Diskette geschrieben (zweiter Teil des Hauptprogramms).

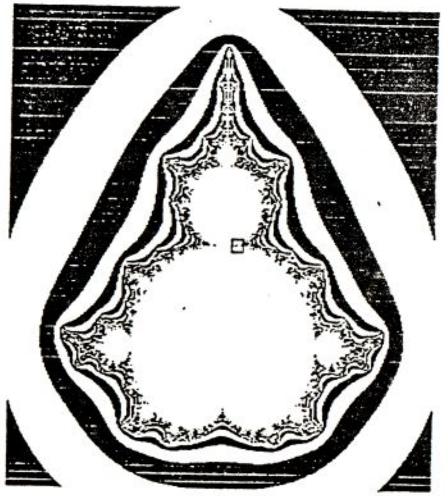
Natürlich läßt sich die relativ banale Rechnerei auch in BASIC unter NEWDOS/80 erledigen. Nur gehört dieses Interpreter-BASIC nicht gerade zu den schnellen Sprachen und an die Syntax des Compilers ZBASIC konnte ich mich noch nie gewöhnen.

Die für den Umgang mit Reals auf Z80-Rechnern wohl schnellste Sprache Turbo-Pascal benötigte mit einem bei 9,216 MHz laufenden HD64180 dann auch noch volle 4 Tage zur Erzeugung des größer wiedergegebenen Bildes, das mit einem entsprechend abgewandelten Programm auf der GRIP-5 mit 768*560 Bildpunkten errechnet wurde. Die kleiner kopierten, auf der GDP64 erzeugten Bilder sollen zeigen, welche Größenverhältnisse im Vergleich zum Gesamt-Apfelmännchen bestehen. Jedes Bild ist dabei ein verkleinerter Ausschnitt des vorherigen Bildes.

Die Parameter der einzelnen Bilder sind:

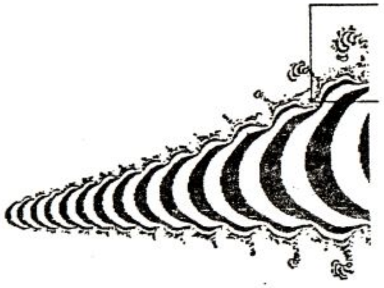
Xmin	Xmax	Ymin	Ymax	Kmax (Iterationen)
-2,0	0,5	-1,25	1,25	50
-0,8	-0,7	0,05	0,015	50
-0,795	-0,775	0,135	0,152	200
-0,795	-0,794	0,1434	0,1444	200
-0,79452	-0,79440	0,14432	0,14444	250
-0,79445	-0,79441	0,14434	0,14438	300
-0,79445	-0,79443	0,14435	0,14437	300

Listing of MANDEL.PAS

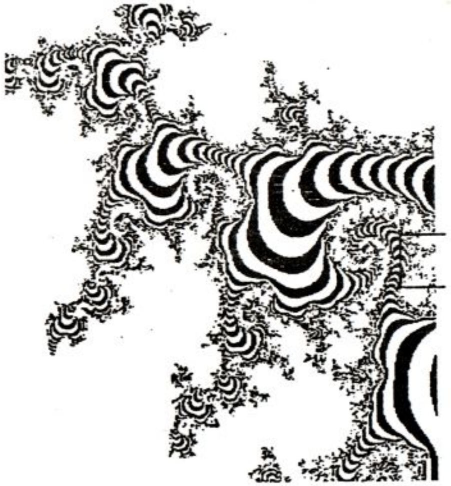


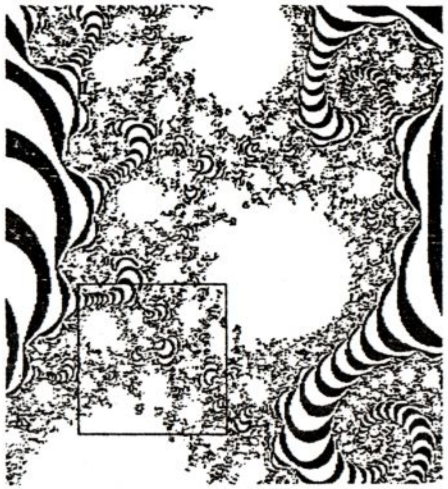
```
PROGRAM mandel;
CONST
   (* fuer die Apfelmaenner-Berechnung *)
   Xmin = -0.79445;
   Xmax = -0.79441;
   Ymin = 0.14434;
   Ymax = 0.14438;
   Kmax = 300;
   fname = 'aman18.gra';

   ak = 2;
   Rmax = 50;

   (* fuer die HRG GDP64 *)
   Xhrg   = 512;
   Yhrg   = 256;
   cmd    = $70;
   ctrl1  = $71;
   xmsb   = $78;
   xlsb   = $79;
   ymsb   = $7A;
   ylsb   = $7b;
   hrgrd  = $60;

   speicher = 256;

VAR
   puffer : ARRAY Ä1..speicherÜ OF byte;
   ib, A, B, I, J  : byte;
   f   : FILE;
   Xn, Yn, Xr, K : integer;
   Xi, Yi, dX, dY, p, q, X, Xalt, Y : real;

PROCEDURE ready;
VAR
    status : byte;
BEGIN
    REPEAT
       status := portÄcmdÜ;
    UNTIL ( status AND 4) =4
END;

PROCEDURE setx ( Xp : integer);
BEGIN
    ready;
    portÄxmsbÜ := Xp DIV 256;
    portÄxlsbÜ := Xp MOD 256;
END;

PROCEDURE sety ( Yp : integer);
BEGIN
    ready;
    portÄymsbÜ := Yp DIV 256;
    portÄylsbÜ := Yp MOD 256;
END;

PROCEDURE dot ( V : integer );
BEGIN
    ready;
    IF V = 0 THEN portÄcmdÜ := 1 ELSE portÄcmdÜ :=0;
```

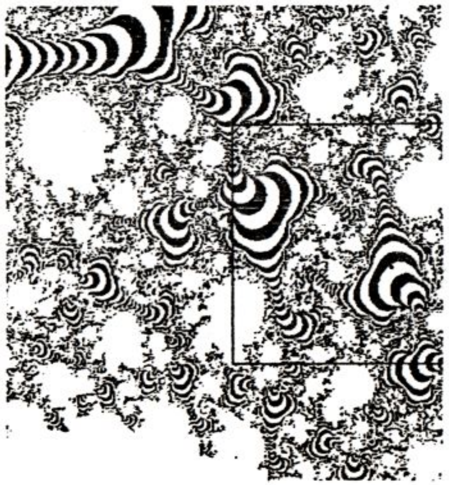
HEFT
21
September
1987



```
       ready;
       portÄcmdÜ := 128;
   END;

   PROCEDURE clh;
   BEGIN
       ready;
       portÄcmdÜ := 7;
       ready;
       portÄctrl1Ü := 33;
   END;


   BEGIN    (* Hauptprogramm *)
       clh;
       dX := ( Xmax - Xmin ) / Xhrg;
       dY := ( Ymax - Ymin ) / Yhrg;
       FOR Xn := 0 TO Xhrg - 1 DO
       BEGIN    (* Apfelmaenner-Berechnung *)
          Xi := Xmin + dX * Xn;
          setx ( Xn );
          FOR Yn := 0 TO Yhrg - 1 DO
          BEGIN
             Yi := Ymin + dY * Yn;
             K := 0;
             X := 0;
             Y := 0;
             REPEAT
                Xalt := X;
                X := X * X - Y * Y + Xi;
                Y := 2 * Xalt * Y + Yi;
                K := K + 1;
             UNTIL ( X * X + Y * Y > Rmax ) OR ( K = Kmax);
             IF K = Kmax THEN K := 0;
             sety ( Yn );
             dot ( K MOD ak );
          END;
       END;
       BEGIN    (* Abspeichern der HRG auf Disk *)
          assign ( f , fname );
          rewrite ( f );
          FOR A := 0 TO 63 DO
          BEGIN
             Xr := A * 8;
             setx ( Xr );
             FOR B := 0 TO 255 DO
             BEGIN
                sety (B );
                REPEAT
                      ready;
                      portÄcmdÜ := 15;
                      ready;
                      I := portÄhrgrdÜ;
                      ready;
                      portÄcmdÜ := 15;
                      J := portÄhrgrdÜ;
                UNTIL I = J;
                puffer Ä B+1 Ü := i;
             END;
             blockwrite (f,puffer,2);
```







Listing of MANDEL.PAS

```
        END;
        close ( f );
        writeln ( 'Bild fertig und abgespeichert !' );
    END;
END.
```

```
;Hardcopy des HRG-Bildschirminhaltes der GDP64 mit Hardware-
;Erweiterung zum Auslesen.
;Ausdrucken über EPSON MX80 mit Typ 3 ROMs
;
CMD     EQU     70H             ;Ports der GDP64
CTRL1   EQU     CMD+1
CTRL2   EQU     CMD+2
CSIZE   EQU     CMD+3
DELX    EQU     CMD+5
DELY    EQU     CMD+7
XMSB    EQU     CMD+8
XLSB    EQU     CMD+9
YMSB    EQU     CMD+10
YLSB    EQU     CMD+11
MFREE   EQU     60H             ;Auslesen nur mit Hardware-Patch
;
        LD      A,1BH           ;ESC, A, 08H   ( 8/72" Linespacing)
        CALL    PRINT           ;  Zeichen an Drucker
        LD      A,41H
        CALL    PRINT
        LD      A,8
        CALL    PRINT
        XOR     A
        OUT     (YMSB),A        ;YMSB immer 0
        LD      B,1             ;Zähler für Bildschirmhälften
B35     LD      A,0FFH          ;zweite Hälfte fertig?
        CP      B               ;
        JR      NZ,WEITER       ;nein: dann nächste 8-Dot-Spalte
        LD      A,0CH           ;sonst <CR> an Drucker
        CALL    PRINT           ;und
        JP      0000H           ;WBOOT, wenn fertig
;
WEITER  LD      A,B             ;1 oder 0 in B an XMSB
        OUT     (XMSB),A        ;für zweite oder erste Hälfte
        LD      C,0F8H          ;rechter Rand minus 8 Dots
B40     LD      A,1BH           ;ESC, L, 00H, 03H an Drucker
        CALL    PRINT           ; 768 Graphik-Bytes folgen
        LD      A,'L'
        CALL    PRINT
        XOR     A
        CALL    PRINT
        LD      A,3
        CALL    PRINT
        LD      A,C             ;X-Koordinate innerhalb einer
        OUT     (XLSB),A        ;Bildschirmhälfte
        LD      E,0FFH          ;Abwärtszähler für Y-Koordinate
B90     LD      A,E             ;nächstes Byte lesen
        OUT     (YLSB),A        ;Y-Wert vorgeben
TEST1   LD      A,0FH           ;Ausgabe anfordern
        OUT     (CMD),A
        CALL    WAIT
        IN      A,(MFREE)       ;HRG-Byte
        LD      D,A             ;zweimal lesen,
        LD      A,0FH           ;um eventuelle
        OUT     (CMD),A         ;Lesefehler
        CALL    WAIT            ;zu erkennen
        IN      A,(MFREE)
        CP      D
        JR      NZ,TEST1        ;wenn keine Übereinstimmung
        CPL                     ;Byte invertiert drucken
        CALL    PRINT           ;dreimal drucken,
        CALL    PRINT           ;um Bildschirmformat
        CALL    PRINT           ;ungefähr zu treffen
        DEC     E               ;nächster Y-Wert
        LD      A,0FFH          ;waren das schon alle ?
        CP      E               ;
        JR      NZ,B90          ;wenn nicht, nächstes Byte
        LD      A,C             ;X-Koordinate
        SUB     8               ;um ein Byte zurücksetzen
        LD      C,A
        LD      A,0DH           ;Wagenrücklauf
        CALL    PRINT
        LD      A,0AH           ;Zeilenvorschub
        CALL    PRINT
        LD      A,0F8H
        CP      C               ;Bildschirmhälfte fertig ?
        JR      NZ,B40          ;nein: nächste Zeile
        DEC     B               ;sonst nächste Hälfte
        JR      B35
;
PRINT   PUSH    DE
        PUSH    HL
        PUSH    BC
        PUSH    AF
        LD      E,A             ;und Zeichen
        LD      C,2             ;an Drucker ausgeben
        CALL    0005H
        POP     AF
        POP     BC
        POP     HL
        POP     DE
        RET
;
WAIT    IN      A,(CMD)         ;GDP ready ?
        AND     04H
        JR      Z,WAIT
        RET

        END
```

HEFT
21
September
1987

GENIE, CP/M und 80 Zeichen und HRG ???

Unser Computer ist schon ein gutes Geraet. So mit NewDOS, G-DOS, H-DOS ect. laesst sich sehr viel machen. Vor allem, haben wir sehr gute Programme und Programmiersprachen.

ABER, die Betriebssysteme (Newdos, G-DOS, ect.) sind zu keinen anderen Systemen (Osborne, Prof80, NDR, usw.) kompatibel.
Die Programme und Disketten koennen nur zwischen den Computern TRS 80,GENIE und KOMTEC getauscht werden. Und diese Computer sterben (leider) langsam aus.

Anders bei dem Betriebssystem CP/M. CP/M ist zwar auch schon auf dem absteigenden Ast, aber durch die grosse Zahl der CP/M User wird es noch eine ganze Weile unser System ueberleben. Bei den CP/M Computer ist nur ein kleiner Teil des Betriebssystems, das BIOS, an den Rechner angepasst, der Rest des Betriebssystems ist fuer alle Rechner gleich.
Durch diese Software-Schnittstellen (BDOS, CCP) ist CP/M weltweit zu einem Standart-Betriebssystem geworden.
CP/M ist bei weitem nicht so komfortabel als Newdos, aber durch diesen Standart sind alle kommerziellen Programme mit 8 Bit CPU nur unter CP/M lauffaehig.

Es gibt nun einige Schaltungen, mit denen man den GENIE oder den TRS 80 auf CP/M aufbohren kann. Viele von Euch haben das auch schon gemacht. Aber genau hier fangen die Probleme an. Nicht mit dem CP/M, das laeuft einwandfrei, sondern mit der Darstellung am Bildschirm.
Bekanntlich hat unser Bildschirm 64 Zeichen und 16 Zeilen. Um vernuenftig mit CP/M arbeiten zu koennen, braucht man aber 80 Zeichen und 24 Zeilen. Alle CP/M Programme (zb. WordStar, Dbase, Multiplan, Turbo Pascal, Mbasic, ect.) sind auf dieses Format angewiesen.

Schmitke Elektronik hat zwar vor laengerer Zeit eine 80 Zeichen Karte herausgebracht, aber diese ist inzwischen nicht mehr lieferbar. Ausserdem ist die Karte nicht unbedingt das, was man sich unter einer guten Loesung fuer dieses Problem vorstellt.

Hier kommt nun GRIP ins Spiel.
Zu einem Zeitpunkt, als ich CP/M eingebaut hatte, aber (Gott sei dank) noch keine Schmitke 80 Zeichenkarte vorhanden war, wurde GRIP(Grafik-Interface-Prozessor) in der CT beschrieben.
Nach der anfaenglichen Begeisterung und den (unvermeidlichen) Schwierigkeiten (CP/M Anpassung Software maessig) ist die GRIP bei mir bis heute voll im Einsatz.

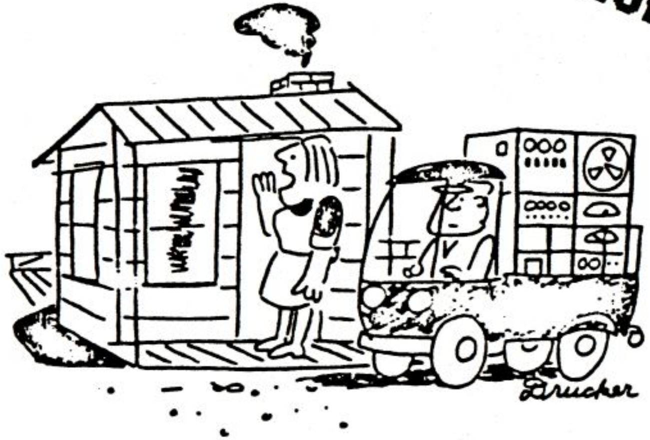
Nun zur GRIP selbst.

GRIP ist nicht nur eine 80 Zeichen Karte, sondern ein ganzes Computer System auf einer Europakarte. Sie laest sich wie ein Terminal ansteuern (TVI 950) und enthaelt ein Videointerface fuer Text und Hochaufloesende Grafik, mehrere serielle und parallele Schnittstellen zur Verbindung mit dem Hostrechner der und Peripherie sowie Anschluesse fuer einen Lichtgriffel und einen Lautsprecher.

Ein eigener Z80A-Prozessor uebernimmt den Datentransfer und die Grafik berechnungen. Die Karte laesst sich auf zwei verschiedene Arten von Host ansteuern: ueber eine V24 Schnittstelle oder ueber den ECB-Bus. Bei meinem System wird von letzterer Moeglichkeit Gebrauch gemacht.

Da GRIP 66kByte RAM enthaelt, konnen alle Schnittstellen gepuffert werden, d.h. die eingehenden Daten werden vor der Verarbeitung oder Ausgabe zwischengespeichert.

CLUB 80 HARD — — — CLUB 80 HARD



„Charlie! Wo soll der neue Computer hin?"

Portbelegung

Um das Umschreiben von Programmen zu erleichtern habe ich hier von einigen Computerkonfigurationen aus unserem Club die Portadressen zusammengetragen. Jedoch haben sich nur wenige (4) daran beteiligt. So konnte ich nicht feststellen ob sich nicht manche Ports bei den selbstgebastelten Zusatzkarten überschneiden.

Alle Portadressen sind in sedezimaler Schreibweise dargestellt.

Gesteuert wird GRIP mit doppel Escape (1bh 1bh xx yy), Escape (1bh xx) und Controll Zeichen (00 bis 1fh).

Die Karte laesst sich wie ein normales Text-Terminal einsetzen. Sie enthaelt im Text-Modus 9 verschiedene Zeichensaetze, darunter alle griechischen und sonstigen europaeischen Sonderzeichen. Auch ein eigener Zeichensatz laesst sich laden, d.h. alle Zeichen koennen vom Benutzer selbst umdefiniert werden.

Zusaetzlich sind in der Software noch Sonderfunktionen vorgesehen:

- Zwei Statuszeilen wovon nur immer eine sichtbar ist
  -erste Zeile mit Logo, Status, Uhr
  -letzte Zeile kann selbst definiert werden
- eine abfragbare Echtzeituhr
- Monitortestbild
- einstellbares Textformat (max. 96 Zeichen x 33 Zeilen)
- einstellbares Video-Signal (Anpassung des Video-Signals an den Monitor)
- Tasten-Umcodetabelle (fuer jede Taste kann statt des Orginal-Tastencode ein beliebiger Zeichenstring zum Host gesendet werden.) diese Moeglichkeit ist fuer uns leider unbrauchbar, da unsere Tastatur eine Matrix ist, und keinen direkten ASCII-Code abgibt.
- Tastenzwischenspeicher (max. 64 Zeichen)
- einen Melodiegenerator (Tastenklick)
- Abfrage des Lichtgriffels
- einschaltbaren Drucker Spooler
- und sonst noch einige Sonderfunktionen.

An Schnittstellen stehen zur Verfuegung:

- ECB-Bus
- V24/RS323
- 8 Bit Apsgang (Drucker)
- 8 Bit Eingang (Tastatur)
- BAS,Hsync,Vsync (mussen zusammengeschlossen werden bei normalen Monitoren)
- Sound

Die Kommunikation mit der GRIP erfolgt entweder ueber die V24 Schnittstelle oder ueber den ECB-Bus. Ueber den ECB-Bus werden zwei Portadressen (A0,A1 oder C0,C1 ueber Jumper auswaehlbar) benoetigt. Vom ECB-Bus sieht der Datentransfer so aus:

Auf der Adresse X1 werden die Daten mit I/O-Befehlen eingeschrieben oder ausgelesen; auf X0 der Status abgefragt. Ist Bit 7 des Statusbytes auf '1' gesetzt, so steht am Daten-Port ein Byte von GRIP zum lesen bereit. Statusbit 6 signalisiert mit '1', dass das naechste Datenbyte zum Daten Port geschrieben werden kann.

Die Reaktion auf ein Befehls-oder Datenbyte richtet sich mach dem Modus, in dem sich die Karte gerade befindet.

Im TVI-Modus verhaelt sich GRIP wie ein TVI950-Terminal. Es koennen zusaetzlich zu den Steuercodes (01h - 1fh) noch einige Sonderfunktionen gewaehlt werden:

- Cursorattribute (unsichtbar, blinkender/stehender Block, blinkende/stehende Linie)
- Cursorposition setzen/abfragen
- Zeichenattribute (Index tief/hoch, Kleinschrift, Breitschrift, Unterstreichen, Invertieren, Durchstreichen, unsichtbare Schrift)
- Invertieren ein/aus
- Bildschirm ein (hell)/dunkel
- Strichgrafik ein/aus (zb. fuer Multiplan)

__TRS-80 M1 und Sonderausführungen von Manfred Held (X).__

Port	
0-5	HRG von RB
0	ausschalten der HRG
1	einschalten der HRG
2	untere Ansteueradresse (LOB)
3	obere Ansteueradresse (HOB)
4	einlesen eines Grafikpunktes
5	setzen eines Grafikpunktes
80-87	RS232 - Schnittstelle von RB
80	Datenport für Kanal A
81	Datenport für Kanal B
82	Controlport für Kanal A
83	Controlport für Kanal B
84-87	CTC - Timer auf RS232
84	Kanal 1 der DART
85	Kanal 2 der DART
86-87	zur freien Verfügung
D0-D7	80 - Zeichenkarte von Schmidtke
D0	Adressregister CRTC
D1	Datenregister CRTC
D2	Steuerregister
D3-D7	doppelbelegt
9-F	X Prommer
B0-B2	X Palbrenner
A0-A2	X Grip 2
A2-A3	X Tastatur Port
A4-AF	X Uhr
E8	X Drucker
EC-EF	X Floppycontroller
40-7F	X HD 64180 Intern
E0-E3	X Floppyselect
C0	X CP/M

HEFT
21
September
1987

Im Tektronix-Modus verhaelt sich GRIP wie ein Tektronix-Terminal mit Speicherbildroehre.

Im Tektronix-Modus gibt es wieder verschiedene Modi:
- Alphamodus: ist wie ein "abgemagertes" Text-Terminal
- Punktmodus: es wird ein Punkt an der angegebenen Koordinade gezeichnet
- Vektormodus: es wird eine Linie zwischen zwei Koordinaden gezeichnet
- Inkrementalmodus: es wird von der letzten Koordinade 1-Punkt-weise weitergezeichnet (wie ein Plotter)

Auch im Tektromix-Modus gibt es Zeichenattribute:
- wie im TVI-Modus
- Zeichne durchgehende Linie
- Zeichne gepunktete Linie
- Zeichne strichpunkt Linie
- Zeichne kurz gestrichelte Linie
- Zeichne lang gestrichelte Linie

Soweit zu GRIP 2.6. Es gibt aber auch noch GRIP 3.1. Die Unterschiede betreffen aber nur die hoehere Aufloesung (768 x 560 Punkte), die man nur mit einem Monitor mit hoher Nachleuchtdauer einsetzen kann.
Dafuer fallen aber der Spooler, die Statuszeile und die Tastenumdecodierung weg.
Um mit GRIP 3.1 zu arbeiten muss man nur den EProm und einen Jumper tauschen (Kleinigkeit, kann jeder).

GRIP 4.1 ist wie GRIP 2, Softumschaltbar auf GRIP 3, hat einen erweiterten Befehlssatz, einen groesseren RAM Speicher und ist schneller im Bildaufbau.

Die letzte Entwicklung von CONITEC ist GRIP 5.43. Sie ist wie GRIP 4, ist jedoch noch schneller, hat 256kByte RAM, mehr Befehle und die Uhr ist Batterie gepuffert.

Zu den Preisen:
Meine GRIP war noch sehr teuer. Die Platine mit EProm und sehr gut beschriebenem Handbuch kostete damals wie heute ca. 250 deutsche Maerker (vor 3 Jahren waren die IC's noch teuerer).

Mittlerweile gibt es fertigaufgebaute GRIP's schon zu sehr guenstigen Preisen (z.b. Inserate von CT, MC, usw.)

Fuer die Platine mit EProm und Handbuch von GRIP 5.43 legt man ca. 350 DM's hin. (rentiert sich aber !)

Ausserdem gibt es noch GRIPS (z.Zeit V 1.15 und 2.10). GRIPS ist eine Erweiterung fuer GRIP 2.x. Durch diese Zusatzplatine und einen anderen EProm wird GRIP 2.x schneller und hat einen groesseren Befehlssatz.
Zusaetzlich kann man an GRIPS 2.10 eine Maus anschliessen.

Soweit zu GRIP selbst. Nun zum Anschluss an unser System.

In meinem System wird GRIP nur fuer CP/M benutzt (eigentlich eine Verschwendung). Die Karte wird ueber den ECB-Bus angesprochen. Man muss nur aufpassen, wenn ein 96 poliger Bus verwendet, das die B Leiste zum restlichem System unterbrochen ist. Denn GRIP kann auch Farbe. JA, GRIP kann wirklich Farbe, und das sogar in HRG. Man braucht dazu allerdings eine Zusatzplatine (ca.200.- DM). Dann hat man 16 von 4096 Farben gleichzeitig am Bildschirm zur Verfuegung. Mit einer 2 Farbkarte hat man 128 von 4096 Farben zur Verfuegung.

Fuer die Farbkarte(n) braucht GRIP eben diese B Leiste vom ECB Bus.

TRS80 M4/4p

Port	
F8-FB	Druckerport Datenausgang
	Druckerstatus Eingang
F4-F7	Ausgang FDC Configuration
F0-F3	Ausgang FDC Controlregister
	Eingang FDC Controlregister
E8-EB	RS232 Ausgänge
E8	UART Master Reset
E9	BAUD Rate Register
EA	UART Control Register
EB	UART Transmit Holding Register
E8-EB	RS232 Eingänge
E8	MODEM Status
E9	
EA	UART Status Register
EB	UART Receive Holding Register
FC-FF	Cassettenausgang
	Cassetteneingang
E4-E7	NMI Mask Reg. Ausgang
	NMI Mask Status Eingang
E0-E3	INT Mask Reg. Ausgang
	INT Mask Reg. Eingang
9C-9F	ein- und ausblenden des Boot-ROM
90-93	Tonausgang
84-87	Ausgang zur Umschalten des Video Speicher und Zeichendarstellung

Fuer den Monitoranschluss gibt es mehrere Moeglichkeiten:

1. zwei Monitore - aufwendigste Loesung - brutalo methode
2. zwei Adapterleitungen und jedesmal umstecken - Holzhacker methode
3. einen Umschalter - hauRuck methode
4. elektr. Umschalter - gute Loesung - kein Kontaktprellen / Verschleiss
5. eletkronischen Umschalter ueber Software Umschaltbar - elegante methode

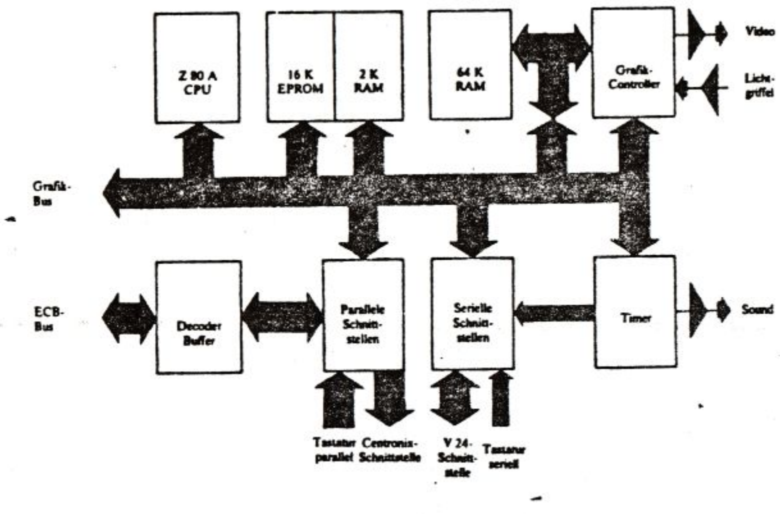
Bei meinem System wird (selbstvertaendlich) Punkt 5 verwendet. Dabei liegt der Umschalter auf Portnr. fbh Bit 4. Der Sound-Anschluss von GRIP und vom Genie sind ueber einen Addierer und einen kleinen Verstaerker zusammen auf den Lautsprecher gelegt. Ansonsten wird nur eine Stromversorgung benoetigt. GRIP braucht ne menge saft (5V ca. 1A), und das schafft das Netzteil des Systems nicht mehr. Will man noch die V24 Schnittstelle benutzen, muessen auch noch +-12V zur Verfuegung gestellt werden (ca. 50 mA).

Software seitig, habe ich in den Bootsector von CP/M eine kleine Routine zum umschalten auf GRIP eingepflanzt (ganze 4 Bytes). Im BIOS von CP/M ist die Displayausgabe auf Adresse EA92 (die ich nach langem suchen endlich gefunden hatte). Man suche sich im File SYS/SYS die Routine, und ersetze sie durch die im Handbuch abgedruckte, und beim naechsten Booten erscheind GRIP mit Logo.

Jetzt muessen nur noch die Programme (WordStar, dBase, Turbo, usw) an die GRIP angepasst werden. Das ist aber ein anderes Thema. Vielleicht schreibt irgend Jemand einmal einen Artikel darueber, wenn genuegend GRIP bei uns im Club eingesetzt sind.

Ich, von meiner Seite aus, kann GRIP nur weiterempfehlen. Zur Zeit duerfte es nichts besseres auf diesem Gebiet geben, das dieses Preis/Leistungs-verhaeltniss hat. Falls es noch Fragen zu GRIP gibt koennt Ihr mich ja anrufen oder Euch gleich an CONITEC wenden.

Viel spass beim GRIPeln wuenscht, Manfred Held

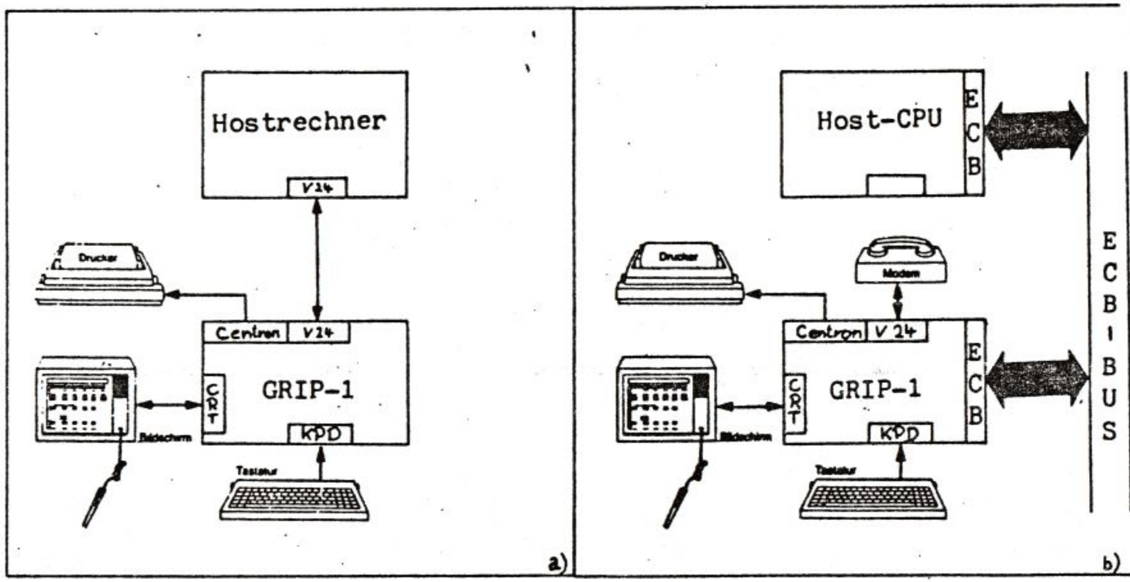


Genie IIs und Zusatzkarten von Gerald Schröder

Port	
0-3F	HD 64180 Intern (überschneidung mit HRG 1b und Host-Karte des IIs
7E	Zusatz-RAM im IIs
7F	Zusatz-ROM im IIs
80-87	RB-SIO
C0	EG 64 (Erweiterun auf 64 bzw. 96 K RAM)
D0-D7	80-Zeichenkarte von Schmidtke
EC-ED	Banker von Helmut Bernhardt
F9	Interrupt-Steuerung im IIs. Diverses im IIIs
FA	Systembyte im IIIs
FD	Druckerport bei alten Genies
FE	Systembyte im Genie IIs
FF	Kasetten E/A

(D0-D7,F1,F2,F9,: SIO/PIO bei IIs (TCS-Zusatz-Karte)

Eckehard Kuhn



BÖRSE -- BÖRSE -- BÖRSE

Suche Programm-Disk für VISICALC, TRS 80 Modell 3 Tastatur -QWERTZU-
40 oder 80 Tracks DD/SS. Deutsches Begleitbuch vorhanden.
Angebote mit Preisangabe (Copy für die Schweiz kein Problem)
an W. Piller Tel. 01/784 74 18
Rohnenstr. 8
CH-8835 Feusisberg

Ich habe auch noch folgendes Problem:
Für meinen TRS 80 Modell 3, habe ich noch verschiedene Handbücher mit Programmdisketten die auf dem Computer mit TRS DOS laufen. Da es sich jedoch um original amerikanische Programme handelt, sind sie auch für die amerikanische Tastatur ausgelegt. Mein TRS 80 hat jedoch die QWERTZU Tastatur, desshalb habe ich grosse Schwierigkeiten bei der Dateneingabe. Und zwar nicht nur weil Z und Y vertauscht sind, sondern auch die Zusatzzeichen sind unterschiedlich angeordnet. (werden für "commands" gebraucht)
Wer weiss mir eine Lösung ?
Gibt es event. ein Convert-Programm mit dem die Programmdisk umgeschrieben werden kann ?

Herzliche Grüsse
Walti

PS. Komme am 8. Nov. 87 zur Hobby Elektronic (Samstag nicht möglich)

<u>Angebot für den Bastler:</u>
2 Cassettendrive's von ADAM
1 Tastatur (ADAM)
sowie 3 Programm-Cassetten mit Programmen
für ADAM-Rechner zu verschenken. (gestifet von Peter Spies)
Gesamt oder einzeln bei der Redaktion erhältlich.

Eigentlich sind das keine guten Nachrichten für unsere Clubmitglieder, was im letzten Info unter der Überschrift "Tandy in Deutschland tot" stand. Wenn Hartmut Obermann unsere Computer als Dinosaurier der Homecomputerszene bezeichnet, dann hat er in gewisser Weise recht, obwohl es schmerzlich ist, einen erst vor drei Jahren auf den deutschen Markt gebrachten 4p-Computer schon heute zum alten Eisen zu zählen.

Doch überlegen wir mal: Wer von uns fährt schon immer das neueste Automodell? Ist nicht der mehrere Jahre alte Wagen für die meisten von uns ein nützlicher und unentbehrlicher Helfer? Ähnlich beim Computer: Entscheidend ist, daß er seinen Zweck erfüllt, unabhängig vom Baujahr. Wer sich mit Datenbanken, integrierten Systemen oder komplizierten mathematischen Problemen beschäftigt, braucht schnelle Mikroprozessoren mit 16 oder gar 32 Bit Wortbreite und Speichern im Megabytebereich. Wer jedoch bescheidenere Aufgaben mit dem Computer lösen will, kommt mit den "Dinosauriern" noch ganz gut zurecht.

In der industriellen Steuerungstechnik liegen die 8 Bit-Mikroprozessoren nach wie vor an führender Stelle. Wo Platzbedarf, vielfältiges Angebot an Hard- und Software und geringer Leistungsverbrauch eine Rolle spielen, sind die 8 Bit-Typen immer noch die erste Wahl. Gerade das Herz unserer TRS 80 - Computer, der Z80, ist - was die verkauften Stückzahlen betrifft - ein ganz großer Renner. Aus der Zeitschrift "Design & Elektronik" habe ich einen kleinen Abschnitt herauskopiert, der eine Bestätigung dafür ist, daß wir mit Zilogs Erstling nicht aufs falsche Pferd gesetzt haben.

> Obwohl die CPU-Architektur des Z80 bereits vor mehr als 10 Jahren entstand, ist diese Einheit immer noch der am weitesten verbreitete 8-Bit-Mikroprozessor, von dem — bei steigender Tendenz — jährlich etwa 50 Millionen Stück verkauft werden. Die Langlebigkeit des Z80 auf dem Mikroprozessormarkt ist sicherlich auf die Leistungsfähigkeit und Flexibilität seiner Architektur zurückzuführen.
>
> 14 Design&Elektronik

Was jährlich 50 Millionen mal verkauft wird, muß ein erstklassiges Produkt sein und wird in absehbarer Zeit nicht aussterben. Für Aufgaben im unteren oder mittleren Leistungsbereich ist das Leistungs/Preisverhältnis der neueren TRS-80 und des Z80 ganz hervorragend.

Heinrich Betz

Was habt Ihr gegen CP/M ?

Jedesmal, wenn sich einer der TRS80-/GENIE-Päpste zu CP/M äußert, sträuben sich bei mir die Nackenhaare. Da kommen dann Äußerungen zu Papier, die dem unvoreingenommenen Info-Leser gleich die Vorstellung aufzwingen, daß man von CP/M lieber die Finger läßt, wenn schon ein Herr H. aus E. oder ein Herr S. aus L. u.s.w. damit nicht arbeiten wollen oder vorgeben, damit Schwierigkeiten zu haben. Die gleichen Herren bedienen sich aber gerne der CP/M-Programme, die unter NEWDOS oder GDOS nicht verfügbar sind. Und gerade die Vielzahl guter Programme, die unter CP/M laufen, ist eines der Hauptargumente für dieses Betriebssystem.

Ein weiterer Pluspunkt für CP/M ist seine Hardware-Unabhängikeit. Voraussetzung auf der Hardwareseite ist lediglich, daß als CPU ein 8080, Z80, HD64180 oder Z280 vorliegt und daß ein Umschalten zwischen ROM und RAM (ab 0000H) möglich ist. Nach dem Abschalten des Boot-ROMs muß ein zusammenhängender RAM-Speicher von 0000H an vorliegen. Wie Peripherie angeschlossen ist, spielt keine Rolle. Diese Hardware-Komponenten werden entsprechend im BIOS berücksichtigt. Das BIOS ist der Teil des Betriebssystems, der elementare I/O-Operationen mit der Hardware erledigt (z.B. Zeichen von der Tastatur holen, Zeichen auf den Bildschirm ausgeben, Sektor von Diskette lesen oder auf Diskette schreiben u.s.w.). Da durch Anpassung im BIOS CP/M auf jeder Hardware-Kombination zum Laufen zu kriegen ist, ist es völlig egal, ob der Bildschirm memory mapped CPU-Adreßraum verschlingt oder als seriell angeschlossenes Terminal bereitsteht.

Diese Vielfalt an möglichen Computerarchitekturen ließ in der Blütezeit der 8-Bit-Rechner (und auch heute noch) hervorragende Maschinen entstehen, die sich in der Hardware zwar alle unterschieden, auf denen aber alle CP/M-Programme liefen.

Die neuere Entwicklung durch Eistieg eines führenden Großrechner-Herstellers in den Microcomputer-Markt mit einem Möchtegern-16Bit-Prozessor und einem dem CP/M ähnlichen Betriebssystem führte aber ungewollt in eine Sackgasse. Das DOS dieser Geräte sieht zwar auch die Abwicklung der I/O-Operationen über Betriebsystem-Aufrufe vor, was aber wegen der erbärmlichen Geschwindigkeit der Bildschirmausgabe nicht eingehalten worden ist. Die meisten Programme bedienen die Peripherie direkt (ähnlich wie wir das von allen besseren Programmen unter NEWDOS/GDOS kennen).

Dieses Verhalten der Software-Anbieter führte dazu, daß nur der Hardware-Hersteller sein Gerät verkaufen konnte, der es 100% kompatibel zum Vorbild baute. Wenn diese Voraussetzung nicht erfüllt war, liefen viele Programme nicht einwandfrei, derentwegen der Anwender sich das Gerät gekauft hatte. Mit solchen Geräten ist nicht viel zu verdienen. Der Begriff "Industrie-Standard" ist nur eine positive Bezeichnung dieser Misere. Dahinter verbirgt sich ein in der Hardware völlig festgefahrenes System, das mit dem TRS80 in dieser Hinsicht durchaus vergleichbar ist.

Aber zurück zum TRS80 und seinen Nachfolgern. Die fest vorgegebene und recht verbaute Hardware und die direkt darauf zugreifenden Programme ließen wenig Spielraum für Verbesserungen, ohne daß das Gerät die TRS80-Kompatibilität einbüßt. Um CP/M fahren zu können, war es nötig, daß Floppy, Drucker, Tastatur und Bildwiederholspeicher nicht mitten im Adreßraum der CPU liegen. Für NEWDOS/80 und GDOS war das aber zwingend. Es wurden Banker entwickelt, die zwischen ROM und memory mapped I/O und RAM umschalten oder nur zwischen ROM und RAM umschalten und memory mapped I/O an das obere Speicherende verschieben. Mit teilweise riesigem Aufwand wurde die gleichzeitige TRS80-Kompatibilität und CP/M-Fähigkeit erreicht. In einem Testbericht der CHIP zum GIIIs liest sich das so: "Mußte es denn ausgerechnet eine 8Bit-Maschine sein, in die soviel Arbeit und Ideenreichtum gesteckt wurde ?"

Dabei kann die Architektur eines CP/M-Rechners sehr einfach ausfallen, wenn man auf die TRS80-Kompatibilität verzichtet (siehe mc-CP/M-Computer). Es ist also nicht verwunderlich, daß bei dem dafür nötigen Aufwand und den damit verbundenen Einschränkungen heute niemand mehr TRS80-kompatible Computer anbietet.

Aber nicht nur die geringen Anforderungen an die Hardware und die danmit verbundene Gerätevielfalt und das riesige Software-Angebot sprechen für CP/M. Auch das Programmieren in Maschinensprache ist unter CP/M sehr viel einfacher. Alle Betriebssystem-Aufrufe und Hardware-Zugriffe erfolgen über CALL 0005H, wobei im Register C immer die gewünschte Funktion ausgegeben wird und in anderen Registern je nach Funktion die Parameter übergeben / übernommen werden. Der Programmierer braucht sich über die Hardware-Architektur überhaupt keine Gedanken zu machen.

Daß nun ausgerechnet unsere Software-Päpste um CP/M einen großen Bogen machen, liegt wahrscheinlich daran, daß sie die enorme Arbeit, die investiert werden muß, um mit NEWDOS/GDOS umgehen zu können, nicht umsonst geleistet haben wollen und auch nicht bei CP/M noch einmal von vorne anfangen wollen. Es kostet tatsächlich einige Überwindung, anstelle von ZEUS nun ein Maschinenprogramm mit Wordstar zu schreiben und mit einem der zahlreichen Assembler zu übersetzen. Man muß gleich mehrere Handbücher wälzen.

Etwas fadenscheinig kommen mir die Argumente wie miese Leistungsfähigkeit und mangelnder Bedienungskomfort vor. Zugegeben, es ist schon etwas lästig, wenn man sich immer erst erinnern muß, in welchem Drive nun die Diskette steckt, auf der die gewünschte Datei vorliegt. NEWDOS sucht mir das selbst heraus. Andererseits ärgert mich bei NEWDOS das dort übliche DIR-Chaos. Ein neu auf die Diskette geschriebener File verschwindet irgendwo mitten zwischen den schon vorher auf der Scheibe vorhandenen Files. Dagegen wird von CP/M das Directory so ausgegeben, wie die Einträge nacheinander erfolgt sind. Und dann gibt es da noch diese herrlichen Wildcards, mit denen eine selektive Anzeige des Directory möglich ist.

Die vorgeworfene geringe Leistungsfähigkeit von CP/M ist eher als Vorteil zu sehen. Dadurch, daß CP/M nur auf reine Betriebsystem-Funktionen beschränkt ist, nimmt es wenig Speicherplatz weg und braucht auch nicht mit Overlay-Technik zu arbeiten. Alle über die Betriebsystem-Funktionen hinausgehenden Leistungen müssen entsprechende Programme liefern, und dafür gibt es sehr gute Utilities. Es ist mir völlig gleichgültig, ob ein Disketten-Kopieren mit einem Betriebsystem-Befehl oder mit einem Programm durchgeführt wird, ausschlaggebend ist der Komfort bei der ganzen Aktion.

Und schließlich habe ich als CP/M-User Anschluß an all die vielen Z80-Rechner, die in den letzten 10 Jahren auf den Markt gebracht wurden. Was bei mir läuft, ist auch dort verwendbar und umgekehrt.

Andererseits ist es bedauerlich, daß NEWDOS/80 nicht einfach auf andere Geräte angepaßt werden kann. Selbst wenn sich irgendjemand diese Mühe machen sollte, alle hardware-abhängigen Routinen herauszusuchen und umzuschreiben, wäre das von zweifelhaftem Wert, weil halt die besseren Programme unter NEWDOS/80 die Treiber-Routinen für Bildschirm und Tastatur umgehen und selbst versuchen, Tasteneingaben in den Speicherzellen ab 3800H zu erkennen.

Erfreulich ist im Zusammenhang mit CP/M, daß sich unser Vorstand über die Unkenrufe der Gurus hinweg für eine Öffnung in Richtung CP/M ausgesprochen hat. Mit zusätzlich CP/M sind wir nicht mehr ganz so weit vom heutigen Geschehen entfernt wie ausschließlich mit NEWDOS/80 und GDOS. Oder weiß sich jemand zu erinnern, in den letzten 2 Jahren in einer Fachzeitschrift etwas über TRS80, GENIE, NEWDOS/80 oder GDOSs gelesen zu haben? Von CP/M liest man inmitten des ganzen IBM-, Amiga- und Atari-Gdröhns dann doch gelegentlich noch was.

Helmut Bernhardt

LS-DOS 6.3 - KOPIERSCHUTZ

Mit seinem neuen LS-DOS 6.3 folgt Logical Systems Inc. dem Beispiel anderer Softwarefirmen und verpasste dem DOS einen Kopierschutz. Allerdings auf eine besondere Art und Weise. Beim erstmaligen Booten einer neuen Systemdiskette werden vom DOS hardwarebezogene Daten ermittelt und innerhalb des Systems gespeichert. Wie dies vor sich geht und um welche Daten es sich handelt, ist bislang noch nicht bekannt. Auf jeden Fall werden diese Daten bei jedem Backup mitkopiert. Bei jedem Booten vergleicht nun LS-DOS 6.3 die innerhalb des DOS gespeicherten Daten mit denen aus dem momentan benutzen Computer. Wird ein Unterschied festgestellt, also ein anderer Computer verwendet, setzt das DOS ein Flag. Wann und wo dies geschieht, ist nicht bekannt. Auch wird dieser Test sehr schnell und nur bei einigen DOS-Befehlen durchgeführt. Ab und zu wird auch bei der Rückkehr von Anwenderprogrammen ins DOS nach diesem Flag geschaut. Wird nun tatsächlich ein anderes LS-DOS 6.3/Computer benutzt, wird man vom System aufgefordert, das richtige LS-DOS zu booten.
Es ist nun durchaus möglich bzw. fast sicher, daß diese Routine innerhalb des Systems aufgespürt wird. Der ermittelte Code wird jedoch sicherlich an der einen und anderen Stelle des Systems verschlüsselt gespeichert sein. Es soll auch nicht möglich sein, diesen Code zu entschlüsseln. Teile davon dürften aber in der Seriennummer eines jeden LS-DOS zu finden sein.
Versuche, die bislang unternommen wurden, den Schutzcode zu umgehen oder auszuschalten, führten in den meisten Fällen zum Absturz oder Zerstörung des Systems.

Das oben Erwähnte ist eine kurze Zusammenfassung aus einem Artikel der 80Micro. Über weitere Erfahrungen mit diesem Kopierschutz wurde bislang nicht berichtet.

Übrigens: Eine deutsche bzw. eingedeutschte Version des LS-DOS 6.3 soll ab Spätherbst 1987 auch bei uns erhältlich sein.

Klaus Hermann

Das TSCRIPS-Drama
(Ein Lehrstück in 35 Aufzügen)

Personen der Handlung:
  Alchemist Schelm
  Kopfjäger
  Hoher Offizier
  Stimme (Faust)
Orte der Handlung:
  Hackerplätze in Lück, Wim und Hiro
Zeit der Handlung:
  1987 (das genaue Datum ist bei jedem Akt angeführt)
Irgendwelche Ähnlichkeiten mit lebenden Personen sind nicht zufällig, sondern voll beabsichtigt.
Anmerkung: Um die Druckkopfnadeln zu schonen, wurden die Namen der handelnden Personen durch leicht verständliche Kürzel ersetzt. (Die Überschriften der Aufzüge sind den "Betreffs" der entsprechenden Briefe entnommen.)

20.4. AS: "Beitrag zur Scheibothek"
Hallo, Genosse von der anderen Fakultät! Ich habe da in meiner Giftküche ein Ding für TSCRIPS (ab jetzt: "TS") gebraut, das dem Vielschreiber das Leben erleichtern soll. Es setzt automatisch das Datum und die Anschrift in Deinen Briefkopf ein. Glaubst Du, daß ich das dem Fußvolk anbieten kann, ohne daß ich anschließend mit Rückfragen totgetrampelt werde? Als ehemaliger Diskothekar kennst Du Dich doch mit den Qualitäten unserer Leser aus. Hier: Nimm und lies!

23.4. KJ: "Aus der Stille des Urlaubs"
Nun, was kümmert Dich die Menge?
Wirf es mitten ins Gedränge,
was Dein siebter Sinn ersann!
Glücklich sollt sich jeder schätzen,
wenn er auch nur einen Fetzen
Deiner Gab' ergattern kann!
Fürchte nicht zuviele Fragen!
Alchemiste, laß Dir sagen:
"Rückmeldungen" sind sehr knapp!
Doch gesetzt den Fall, sie kämen,
sollte Dich das nicht sehr grämen:
Freude hängt vom Echo ab!
Glücklich preise sich die Meute,
daß sie dieses Mal und heute
nicht "IIIs" von Dir bekommt,
sondern (was wir selten haben)
für den <u>Alltag</u> gute Gaben,
wie es armen Leuten frommt.
Ich erkenne Dich nicht wieder -
"leise flehen Deine Lieder" -
singst mir einen Grabgesang,
Derweil Du mit TSCRIPS' kecken
A- und D-Utily-Blöcken
treibst den *Trappschuh* in die Ecken,
weil ihm sowas nicht gelang!
Die Programme sind 'ne Ware,
die durch gute Kommentare
erst den wahren Wert erhält.
Folglich mußt Du es ertragen,
falls man Dir mit hundert Fragen
später auf den Wecker fällt.
Hach, ich kann es selbst nicht lass'n-

Frage; kann auch "EDTASM"
Deine Source assembiliern?
Dann will ich am grünen Holze
(und sogar mit eitlem Stolze)
es auch einmal selbst probiern.
Bin in TSCRIPS gut zu Hause
wie in meiner eignen Klause;
habe die Version "Fünf-Fünf".
Kann ich's hiermit auch probieren?
Oder braucht man die "Fünf-Vieren"?
(Hältst mich jetzt für unvernünf-
ja, für uneinsichtig gar!?
Das ist traurig - aber wahr...)

25.4. AS: "Vom Dichten und Trachten"
Habe nun, ach, die Source auch noch für EDTASM umgeschrieben. Geb zu: Mein Anhängsel in TS einzubauen, ist etwas kompliziert. Man kann davon leicht einen Tea-Schwips kriegen...

28.4. AS:
Aha, Du bist also ein 5.5er. Her damit! Mit "Sicherheit" lassen sich die Patches auch dort einbauen (...!!!...)

4.5. KJ:
Ich leide an Breitspurigkeit. Deshalb kann ich Deine schmalen 80Spuren nicht lesen. Daß Du aber Dein Spurwerk nicht zu einer sauberen überspringprozession veranlassen kannst, ist ein Alptraum. Außerdem: Wieso schaffst Du noch mit "5.4"?

7.5. AS:
Ich bewältige die PD's jetzt programmatisch. Zur Hölle mit dem PD-Befehl!

19.5. KJ: "Bohrt der schon wieder?..."
Da von WF (WeinFurt) noch immer nichts kommt, schicke mir doch bitte, bitte --- mische einfach mal 80 und 40 durcheinander (auf Vorder- und Rückseite). *ZEUS* vergällt's Dir!

21.5. AS: "TSCRIPS und überhaupt"
Muß zugeben, daß mein Unding noch Mängel hat, denn es hängt sich ab + an = auf. Wenn der Fehler ausgemorzen ist, kriegst Du die Platte. Vorausgesetzt,ich bewältige die verdammten PD's...

2.6. KJ: "TRIED TO LOAD ROM"
Genau das ist die Antwort meines TRS80 jedesmal, wenn ich meine Version 5.5 mit Deiner Erweiterung (die ich inzwischen endlich aus WF erhalten habe) boote. Paßt sie doch nur zu 5.4? Jetzt kriegst Du doch zwei Disks von mir m.d.B., deren Rillen mal mit feilen Files zu füllen. "Auf Los geht's los! 80+40!"

8.6. AS:
Hallo Alter! Es steht geschrieben (Anmerkung des Kulissenschiebers: Wo?), daß bei der Implementation die ORGs ein waches Auge verdienen. (Aha!)
Du mußt mit ORG GANZOBEN das Teil neu assemblieren und dann anhängen. Nimm auf gut Glück HIMEM =FD00h. Mit ORG 4049h DEFW GANZOBEN-1 begrenze dann den Textspeicher.- Gute Nacht!

10.6. KJ: "Sehr geehrter Herr..."
Aus lauter Ehrerbietung vor so vielen ORGs in Ihren works müßte ich Sie eigentlich siezen. Und vor sovielen Ratschlägen, die Du mir erteilst und die ich nicht verstehe! Wohin denn nun mit dem DEFW 0FD00h? Und welche "Zeile 4049h" soll ich ändern? Die gibt's ja gar nicht in Deinem opus! Und wieso strampelst Du von einem ORG zum andern? 6mal! Das ist ja ORGiastisch!

(Anm.d.Regie: Die weiteren tollen statements von KJ seien dem Zuschauer hier vorenthalten.)
Kurz und ungut: Hier ist Dein Listing; jetzt male mir mal hinein, wie Du Dir das denkst, bitte! Schreib doch mal ein richtiges Ass-Buch, damit alle mal das Assemblieren begreifen!

12.6. AS: "Wes T ich scrips, des Löt ich kolb..."
Dummer! Ich taumle nicht "von einem ORG zum andern". Das sind Patches in TSCRIPS. Das heißt: Ab der Adresse, auf die "ORG" verweist, wird der Speicher mit den auf "ORG" folgenden Bytes belegt. Und 4049h ist natürlich nicht in meinem Programm. Es ist einfach der Speicherplatz, wo der Zahlenwert des HIMEM steht. Willst Du HIMEM ändern, z.B. in FE00h, so mußt Du diese neue untere (!) USER-Grenze dort wie folgt eintragen:

ORG 4049h DEFW 0FE00h

Das kann an einer beliebigen Stelle im Programm stehen (natürlich nicht mitten in einem Programmablaufteil). 9100h ist übrigens der Startpunkt von TS 5.4.
Lehrbuch Schreiben? Nee, ein dummer Leser (wie Du) genügt mir.Tschüss!

13.6. KJ:
So, jetzt hab' ich Dich erwischt: HIMEM ist nicht die untere, sondern die obere Grenze des ungeschützten User-Speichers!
(Auch der treueste Hund kann mal seinen Herrn beißen...)

17.6. AS: "TS-Erheiterungen"
Also, mein Lieber: In die Versionen 5.5 und 5.6 geht es nicht rein, ohne gewisse verwanzte Adressen zu ändern.
(Anm.des Kulissenschiebers: Vgl. Aufzug vom 28.4.!)
Jetzt besorge Dir also endlich mal die Version 5.4, mit der alle anderen anständigen Menschen arbeiten! Außerdem solltest Du Dir endlich ein 80er-Laufwerk anschaffen.

Denn leicht ist's wirklich nicht, zu schreiben
mit 80er Draiw auf 40er Scheiben!

5.7. KJ:
Gesagt, getan. Version 5.4 ist jetzt da. Sie tut alles, was ich will (was ich nicht von jeder sagen kann...):

Sie lädt, schreibt, speichert, schweigt und spricht -
nur drucken tut sie leider nicht!

Dauernd behauptet sie: "LINE PRINTER NOT READY!"
Das ist natürlich gelogen. Denn mein Liniendrucker ist mehr als READY. Er ist ganz scharf wie ein RADI aufs Drucken! Ich, nicht faul, disassembliere das Teil. Que lac je? Lauter E's und 83H's! Seitenweise! Na sowas! Wer multipliziert da wen? Wieder mal bin ich vom Stamme Kannitverstan.

7.7. AS: "TS und kein Ende"
TS besteht zur Hälfte aus Bildschirm. Das hättest Du am Auftauchen der Adresse 3C00 erkennen müssen. Der eigentliche Einsprung ins Textsystem ist bei 9100h. Der Code endet deshalb mit 02-02-00-91. Schau mal nach. - Mein nächster Rat (welche Nummer hat dieser jetzt eigentlich?) lautet: Forsche mal nach D3-FD und DB-FD <= OUT(FD) bzw. IN(FD)>. Da stehen 3 Bytes, aus denen Du 3A-E8-37 (Druckerstatus abfragen) bzw. 38-E8-37 (Zeichen ausgeben) machen kannst.
(Der Souffleur: "32-E8-37, nicht 38-...!"
Die Regie zum Souffleur: "Lauauauter!")

10.7. KJ: "Die (doch) endliche TS-Geschichte"
Guten Morgen, lieber $A_{lchemi}S_t$!
Habe gemacht. Wie Du sagtest. Oje, muß ich dumm sein. Immer noch sehe/lese ich: LINE PRINTER NOT READY! Sollte ich die 3 Bytes vielleicht doch woanders hinschreiben?

Nein, ich geb's auf..........
auf...

17.7. AS: "DEM WAHREN, GUTEN, SCHÖNEN"
Die 3 Bytes 3A-E8-37 bzw. 32-E8-37 (Anm.des Souffleurs: "Merkt ihr was? Am 7.7. hieß es 38-...! - O diese Alchimisten, geben Plumbum für Aurum aus! Da muß ich doch mal Rum-Bum machen!!) mußt Du anstelle von D3-FD-00 usw.usw. Der Nuller ist ein NOP (No OPeration). Prüfe mit dem Debugger, was an der Stelle 37E8 steht, wenn der Drucker Ready ist. Oder ersetze den ganzen Quark durch ein CALL 05B4. Suche in Deiner Version nach der Abfrage. Checke, ob gleich darauf die Ausgabe folgt..... (hier hat der Darsteller sein Rollenbuch verlegt - o nein, hier hat der Dichter dieses Dramas den Rosa Faden verloren. Die Geister, die er rief, versteht er selbst nicht mehr.)
Notfalls schicke mir einen Kontoauszug des Sektors, wo DB-FD-00 steht. Hier tut sich offenbar eine Unvereinbarkeit zwischen TRS80 und GENIE auf, die infowürdig ist.
(Der Produktionsleiter: In der Tat. über diese Unvereinbarkeit wurde im INFO schon berichtet, und zwar von einem Hohen Offizier. An den sollte sich der Kopfjäger mal halten.)

19.7. HO: "Hallo..."
Nimm und lies INFO 14, S.43! Die Version, die ich Dir für Deinen TRS80 schickte, 'war für's GENIE (!!) und kann bei Dir nicht laufen (!). Beim GENIE gehen Druckerein- und -ausgabe über den Druckerport, während sie beim TRS80 "memory-mapped" sind. Du mußt also alle (alle!) Port- in Memory-Schreib/Lese-Operationen umsetzen.- Das wär's.

22.7. KJ
Jetzt druckt's. Seit ich mit der Außenwelt nicht mehr per Schiff über den Hafen (port) verkehre, sondern meine Gedanken direkt aus meinem Gedächtnis (memory) durch die Luft zu ihr fliegen lasse - es druckt zwar, aber nur normale Texte; nicht Deinen Adressenautomaten. Hier weigert sich meine TreueRoSinante80mal und wiehert: "Formfehler beim Laden!"

23.7. AS: "Von den letzten Dingen"
Das Problem geht seiner Klärung entgegen. (!)
Wo OUT(FD),A steht, kannst Du D3-FD-00 gegen CD-B4-05 eintauschen. Ersetze mal den ganzen Schmus in FRS17 von RB B3 bis B7 durch CD-D1-05-00-00 - das geht sicher! Mehr fällt mir jetzt nicht ein - -
(An dieser Stelle fordert der Theaterdirektor zu einem unangesagten Posaunenstoß auf, da der größere Teil des Publikums soeben eingeschlafen ist!)

24.7. KJ: "Ach du heiliger A..."
Fortsetzung vom 22.7.: "Formfehler beim Laden!" Setzte ich zur Abwechslung mal ORG=0FD00h und ORG 4049h auf 0FCFFh (denn es muß doch um 1 niedriger sein!?), so erfuhr ich:
"LADEVERSUCH AUF ROM-SPEICHERPLATZ"
Ich habe Deine komplette Anleitung nochmal Schritt für Schritt durchexerziert.
(Hier folgt die Beschreibung der Schritte. Auch sie sei dem ungeduldigen Zuschauer erspart.)

27.7. AS: "Das übliche"
Die ORG's innerhalb TS müssen selbverständlich unverändert bleiben. Stelle sicher, daß alle CALL-Adressen, die innerhalb von TS liegen, ein Label haben, bloß keine Zahl. Du kannst auch 02-02-00-91 in 05-xx-egal-egal verwandeln. 05 heißt Kommentar. xx ist die Anzahl der jetzt noch freien Bytes bis rechts unten (im Sektor). Wenn Du dann meine Erweiterung mit APPEND anhängst, müßte auf Anhieb alles das richtige Format haben.
übrigens: Wenn es Dir doch noch gelingen sollte, das Ding auf Deiner Rosinante zum Laufen zu bringen, so wisse, daß unser Hoher Offizier auch daran interessiert ist. Einer von Euch wird's schon schaffen.

28.7. KJ: "Ente jut - (fast) alles jut"
Danke für die Hinweise! Jetzt klappen: Die automatische Datumeingabe, der Klammeraffe und die durchgehende Unterstreichung - bloß die automatische Adresseneingabe funkt nicht. Was soll das heißen in Deinem Erweiterungscode: "Kann bis vor das Label 'address' gelöscht werden? Ein Label 'address' gibt es in Deinem Listing überhaupt nicht! Ich habe jetzt mal TS disassembliert. Du lieber Gott! Bei Blatt 111 schaltete ich den Drucker ab (das sind 33,5m Papier)! Und in meinen Speicher ging das alles schon gar nicht hinein. So konnte ich auch nicht nach sämtlichen CALL-Labels suchen.

31.7. AS: "Von kalten Spalten und kessen Adressen"
Da-Da-Da-Do- - d.h.: Da das Datum doch korrekt bearbeitet wird, kann ich mir
De-Da-Da-Do- - d.h.: denken, daß das DOS Unterschiede aufweist.
Sieh mal im *GROSSER* nach und vergleiche mittels Debug.

5.8. KJ: "Die Große Aufgabe"
Ich geb's auf - - -.
Neuerdings druckt mein Drucker mit Deinem TS nur noch Chaos.
Obwohl auf dem Bildschirm Sinnvolles steht...

7.8. AS: "-nverstanden!"
Chaos ist die Quelle der Schöpfung. Du mußt halt lernen, Hieroglyphen zu lesen. Dechiffriere sie auf Millimeterpapier. Bei Deiner Patcherei scheint einiges in die Grütze gegangen zu sein. Oder die ORGs in unseren TS's stimmen nicht überein. Ich habe meins ja vielfach vergewohltätigt! Schick mir doch mal 'nen Hex-Ausdruck von so einem mißratenen Brief.

10.8. KJ: "Schreiben Ohne PinkelPause..."
Hier ist der HEX-Ausdruck des Chaos! Schaut wirklich verHEXt aus.

12.8. AS: "Von den vorletzten Dingen"
Dieser HEX-Ausdruck ist die Grafik für Deinen Briefkopf. Nur gibt Dein Drucker sie nicht als solche aus, weil das Programm die Anzahl Grafikbytes gleich Null setzt. Also ist beim Patchen was in die Grütze gegangen. Wahrscheinlich sind die Anzapfadressen bei Deiner und meiner Version nicht dieselben. Du mußt am besten jemand mit möglichst ähnlicher Hardware finden und darauf ansetzen.

13.8. KJ: "Nun sind die Ferien bald zu Ende,
da muß ich nochmal ganz behende..."
("Halt!" ruft die Regie, "Wir machen hier keine Dichterlesung! Da dieser Akt wieder voll gereimt ist, geht jetzt der Vorhang nieder!")

15.8. AS: "Von den hinterletzten Dingen..."
Mit Sicherheit ist Dein TS nicht OK. TS bedient den Drucker nicht über den DCB (CHROUT), sondern gibt selber den Output auf 37E8 bzw. Port FD aus. An der Stelle setzt der Check auf das Byte 7F (Randausgleich) an.
Letzter Versuch: Ersetze den CALL 05B4h durch LD(37E8),A.
übrigens: Die Zeichensätze von TS erzeugt der EPSON mithilfe seiner Drucker-HRG. Jeder Deiner Briefe enthält daher eine Menge Grafik. Durch Deine Zapperei ist aber die Angabe der An-

zahl Grafikbytes in die Binsen gegangen.

16.8. KJ: "Er- und andere -weiterungen"
Es sei Dir gewährt Deine Bitte: Verbannt sei aus unserer Mitte mit unbarmherzigem Tritte Herr TSCRIPS, weil ohne Fort--Schritte!

21.8. KJ: "Haha- hihi..."
Röckraths DOS-Buch sowie *WAGNERs* "Machine language Disk I/O & other mysteries" verhalfen dazu, den Adressenautomaten nun endlich dazu zu bringen, das Gewünschte auszuspucken! Wie das? Ich habe mich mal im Debuggen geübt und mit der Einzelschrittausführung ('I') die Veränderungen in den Registern und im Stack beobachtet. Dabei stellte ich fest, daß der Filename "ADRESSEN/LIB" nicht mit dem Schlußbyte 03 oder 0D in den FCB übernommen wurde. Flugs setzte ich dieses dort ein- und plötzlich ging's: Auf das Kommando 'G' des DEBUG wurde Deine Anschrift eingegeben! Du hast in der Source für EDTASM die Zeile DEFM 'ADRESSEN/LIB', 0Dh geschrieben, was bei EDTASM nicht erlaubt ist. Ich habe daraufhin das 0Dh gelöscht und als Extrazeile DB '0Dh' eingefügt. Mußte dann aber immer noch bei jedem Brief den Debugger bemühen...

24.8. AS: "Uff!"
Wir kommen der Sache näher. Du mußt nun nur noch das Zählbyte für diesen Record erhöhen!

29.8. KJ: "Fin de Partie!"
Erdal, der Frosch! *HEUREKA!* Ich hab's!
Wir haben beide am gleichen Fehlerstrang gezogen. Ich hatte Deine fehlerhafte Zeile nur halb richtig korrigiert, indem ich das 0Dh in Apoströphchen kleidete! Es muß natürlich DEFB 0Dh heißen! Hätte ich beim Assemblieren die Option /WE (Wait upon Error) gewählt, wäre ich früher gewarnt worden.
Nun aber kommt das Tollste: Die Entdeckung des Fehlers in Deinem Source-Code für EDTASM weckte mein Mißtrauen gegen diesen und motivierte mich, mich nun endlich einmal näher mit *ZEUS* zu befassen! So fing ich von vorne an und assemblierte Deine ZEUS-Source mit ZEUS -: Und nun klappt's wirklich!

31.8. AS: "**Betr.** ifft"
Alternativ zum Druckbefehl "BREAK P..." kann man meine Automatik auch mit @K abrufen. Dazu mußt Du nur das zum "Umleitung-bei-P"-ORG gehörige 01-Byte in 05 (=Kommentar) umzappen, dann wird dieser Teil nicht gelesen........

*   *   *   *   *

DEUS ex Machina (mit FAUSTens Stimme):

O glücklich, wer noch hoffen kann,
aus diesem Meer des Irrtums aufzutauchen!
Was man nicht weiß, das eben brauchte man,
und was man weiß, kann man nicht (immer) brauchen.
(Der Tragödie erster Teil)

— - — - — - — - — - —

Die letzten Zeilen sind (ihr wißt's) von Wolfgang Goethe.
Der ganze Rest: er stammt von Arnulf & KaJoete.
Ward diese Niederkunft der hundertdreißig Tage
oft den Akteuren und dem Publikum zur Plage
und tat so mancher sich bereits beim Lunch entfernen -
- so konnten, die da blieben, doch so manches lernen!
*   *   *   *
V O R H A N G

Sie war ja teuer — aber keine Steuern, keine Sozialleistungen und wenn es sein muß, arbeitet sie 24 Stunden.

Sie kostet nicht viel — aber man muß immer darauf achten, daß sie auch etwas tut.

HEFT
21
September
1987

Neues von der Diskothek

- Wie Hartmut im letzten Info bereits angekündigt hat, besitzt der Club nun ein Reinigungsset für Diskettenlaufwerke. Wenn Ihr also damit Probleme haben solltet, so poliert doch mal Eu're heißgelaufenen Schreib- Leseköpfe und laßt Euch Reinigungsdiskette und Reinigungsflüssigkeit zuschicken. Ihr müßt mir jedoch mitteilen, für welchen Laufwerkstyp (Single Side oder Double Side) es benötigt wird (da gibt's nämlich einen Unterschied).

- In den letzten Wochen sind bei mir die nachstehenden Programme eingegangen.

Nr.	Name	Autor	Kurzbeschreibung
367	DCSDR/CMD	K.J. Mühlenbein	Direkte Ausgabe von Texten auf den Drucker (siehe Info 20, Seite 35/36)
368	TRIGONOM/BAS	K.J. Mühlenbein	Berechnung der trigonomischen Grundfunktionen (siehe Info 20, Seite 17/18)
369	WURZEL/BAS	K.J. Mühlenbein	Wurzelberechnung (siehe Info 20, Seite 19)
370	CP/M Packet	Gerald Schröder	Alle Source-Files für CP/M-BIOS (nur Genie II) einschließlich Begleittext als ASCII und Textfile (Die Programme wurden von Gerald Schöder selbst programmiert, sind also nicht mit einem Copyright vorbelastet)
371	GDOS-Hilfpgm	Gerald Schröder	tformat: Programm zum Formatieren einer CP/M Diskette putsys, bootsec: Programm zum Generieren einer CP/M Systemdiskette shift: Programm zum übertragen von GDOS/CMD-files nach CP/M
372	CP/M-Hilfpgm	Gerald Schröder	format: Formatieren von CP/M Disketten sysgen: Erzeugen einer Systemdiskette pdrive: Einstellung der Laufwerksparameter
359	TRSRAM	Gerald Schröder	neuste Fassung des Programms RAMDISK
372	TRSRAMLW	Gerald Schröder	ähnlich TRSRAM, man kann jedoch wählen, auf welchem Laufwerk die RAMDISK installiert werden soll
357	TRSSYS	Gerald Schröder	neuste Fassung von TRSSYS Läd alle Systemfiles in zwei Banks. Man kann somit die Systemdiskette aus Laufwerk 0 entfernen.
373	HDRAMD	Gerald Schröder	wie TRSRAM, jedoch für den Prozessor HD64180
374	HDSYS	Gerald Schröder	wie TRSSYS, jedoch für den Prozessor HD64180
375	HD64256	Gerald Schröder	Utility-Program für HD64180; liefert TRAP-Abfang für Genie IIs
376	Z80TRE	Gerald Schröder	80-Zeichen-Treiber für Schmidtke-Karte im Genie IIs
377	SYS5	Gerald Schröder	neues SYS5 (Debug); nun auch unter 80-Zeichen benutzbar (nur mit Z80TRE)
378	TREIBER	Gerald Schröder	Grafik-Treiber für Genie IIs mit Graphic 1.5
379	HRGTRE	Gerald Schröder	Grafik-Treiber für HRG 1b
380	GRATEST1	Gerald Schröder	Testprogramme für Grafik-Treiber
381	GRASIB	Gerald Schröder	Assembler-Grafikroutinen-Bibliothek + Demo

Andreas Rychlik
Königsberger Allee 120
4100 Duisburg 1

Duisburg, den 2.8.1987

Hallo liebe Clubkameraden !

Wir haben jetzt neu auch eine CP/M 2.2-Software Bibliothek in unserem Club 80.

Ich habe mich bereit erklärt, diese umfangreiche CP/M 2.2-Software Bibliothek zu verwalten und bin auch nur für diese zuständig. Die NEW/TRSDOS-Software Bibliothek ist natürlich die Sache vom Werner Förster.

Wir haben:

ca. 82 Public-Domain Disketten von der CP/M-UserGroup
ca. 32 Public-Domain Disketten von der SIG/M-UserGoup
und noch einige eigene CP/M 2.2-Programme (z.B. für die HRG 1B)

Die Softwareliste gibt es nur auf Diskette, weil die Liste so lang ist, und nicht im Info abgedruckt werden kann !!!
Als Beilage zu der Katalog-Diskette gibt es eine Aufstellung der Diskettennummern, die wir alle haben.

Trotz der Katalog-Diskette werde ich in jedem Info ein Programm herausstellen, um die CP/M-Software schmackhaft zu machen.

Bezüglich der Diskettenformate ist folgendes zu erwähnen:

Da das Anpassen der CP/M-Diskettenformate meißt nur mit viel Problemen vonstatten geht, habe ich mir kein einheitliches Format ausgesucht.

Ich werde die Software in einem gewünschtem Format kopieren !
Dieses Format muß natürlich bei der Bestellung angegeben werden.

Mögliche Formate sind:

Montezuma Sys.	80 DS/DD	Montezuma Sys.	40 SS/DD
Schmidtke S80	DS/DD	Schmidtke S40	DS/DD
Schmidtke S40	SS/DD	Genie III s	80/DS/DD
Genie III	80/DS/DD	P 2000	80/DS/DD
Epson QX-10	40/SS/DD	IBM CP/M 86	40/SS/DD
Osborne	40/SS/SD	Rainbow	80/SS/DD

Ich glaube, daß jeder mit einem dieser Formate zurechtkommt.
Falls nicht, bitte nach meiner PDRIVE-Liste nachfragen! Dort sind ca. 100 Formate angegeben.

382	SCRIPT1 BLOCK1	Gerald Schröder	zwei Schriftsätze für GRABIB
383	VCMOD	K.J. Mühlenbein	hiermit können Zeilen oder Spalten eines im DIF-Format gespeicherten VISICALC-Files alphabetisch oder numerisch sortiert werden (siehe Info 19, Seite 29-31)
384	ZAPTS	Arnulf Sopp	Adressenmanager und andere neue Features für TSCRIPS 5.4

Das war's von der Club-Bibliothek,

Werner Förster

Kopieraufträge werden nur schriftlich angenommen und man sollte die Katalog-Nummer und die Software-Group angeben.

Am Besten wäre es natürlich, wenn alle Disketten in dem gewünschtem Format vorformatiert eingeschickt werden.

Auch muß der Einsender die vollen Porto- und Verpackungskosten in Briefmarken einsenden ! (Ab sofort trägt der CLUB 80 nicht mehr die Portokosten.)

-

Wer seine Programme einschicken will, sollte das Meisterwerk nur mit einer Dokumentation in ASCII als File abschicken. (*/DOC) Ferner muß kein Copyright vorhanden sein. Urheberrechtlich geschützte Programme werden nicht angenommen.

Natürlich können auch ganze PD-Disketten eingeschickt werden. Bestimmt ist einigen nicht bekannt, daß sie die Disketten, selbst wenn sie sie bei irgend jemand gekauft haben, jederzeit weitergeben dürfen.

Viel Spass mit der CP/M 2.2 Software !!!

CLUB 80 — Bücherei

Langsam kommt wieder etwas Leben in die beinahe schon tot geglaubte Clubbücherei, wenn auch das Leben mehr aus Zugängen als aus Ausleihungen besteht!

So kann ich erfreulicherweise berichten, daß sich eines der vermissten Bücher wieder eingefunden hat. Die Nr. 7 in unserer Liste, das Buch "Introduction to TRS 80 Graphics" hat sich wieder in die Reihe der auf die Ausleihung wartenden Publikationen eingeordnet!

Zusätzlich hat die Bücherei Zuwachs von einem TRS 80-Aussteiger, nämlich von Josef Konrad bekommen. Josef hat uns neben einer ganzen Sammlung von 80 US-Zeitschriften drei Bücher vermacht (siehe die folgende Liste). Für diese freundliche Spende möchte ich mich an dieser Stelle recht herzlich bedanken!

Noch mehr Bewegung wird es in der Clubbücherei geben, wenn die bei der Vorstandssitzung beschlossenen Neuankäufe von wirklich aktuellen Büchern getätigt werden. Ich hoffe und glaube, daß dann auch die Anzahl der Ausleihungen wieder rapide steigt!

Hartmut Obermann

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment

---

Nr. 46 : PACAL

David L. Heiserman --- TAB Books Inc.

Einführung in die Programmierung von PASCAL, Tiny PASCAL und Supersoft Tiny PASCAL

Nr. 47 : TRS 80 Assembly Language Programming

William Barden --- Radio Shack

Eine Einführung in die Programmierung des Z80-CPU des TRS 80.

Nr. 48 : 80 US (Zeitschift)

Amerikanische TRS 80-Zeitschrift Ausgaben Feb. 81 bis Juli 83 (teilweise fehlen Ausgaben)

Die Bücherliste - Fragen zur Einführung

Ich werde versuchen, Euch dieses Thema anhand von Fragen und kurzen Antworten nahe zu bringen. Wie Ihr feststellen werdet, hängen alle Fragen eng zusammen. Ob die Antworten jeweils stimmen, dürft Ihr getrost anzweifeln. Mir paßten sie jedenfalls gut in den Kram.

1. Frage: Warum benutzen so wenige Leute im Club CP/M?
Antwort: Unser Rechner war nicht für CP/M gedacht, also hat jeder mit Newdos angefangen und kann/will jetzt nicht umsteigen, vor allem, weil es nicht genug Informationen darüber gibt.

2. Frage: Warum programmiert Ihr alle in Basic, wo Pascal doch viel besser ist?
Antwort: Ihr wißt gar nicht, daß Pascal besser ist. Außerdem gibt es nur unter CP/M ordentliche Pascals (Turbo-Pascal) und das müßtet Ihr auch noch neu lernen. Wieder fehlen Informationen.

3. Frage: Warum leiht sich niemand etwas aus der Clubbibliothek aus?
Antwort: Die Bücher sind für Euch nicht interessant oder Ihr wißt nicht, was alles Interessantes darin steht.

4. Frage: Welche interessanten Bücher schaffen wir neu an?
Antwort: Das hängt ganz von Euch ab, aber vorerst hat der Vorstand zwei Themenkomplexe vorgesehen: Einführungen in CP/M und Pascal. Weitere Vorschläge (Themen oder konkrete Bücher) sind von Euch herzlich willkommen.

5. Frage: Wer sucht die Bücher aus?
Antwort: Klein-Gerald war dumm genug, sich die Vorarbeiten dazu aufbürden zu lassen. Im Endeffekt sollt Ihr uns sagen, welche Bücher wir gefälligst kaufen sollten. Also sagt mal was! (Auch negative Kritik wie z.B. "Bloß nicht den Zaks; der schreibt total unverständlich!" können wir gut gebrauchen.)

6. Frage: Wie geben wir den Leuten Tips, die sich unbedingt selbst ein Buch kaufen wollen?
Antwort: Wir machen einige Bücher schlecht und loben andere in den Himmel. Dann könnt Ihr Euch die schlechten kaufen und selbst darüber ärgern, wie mies sie sind. Wiederum seid Ihr gefragt: Jeder, der lesen gelernt und mehr als den Titel eines Buches entziffert hat, schreibt sofort, was er über das Buch denkt. Dann können andere sich dann nach demBuch die Hacken abwetzen oder mit einem abfälligen Blick bedenken, wenn sie es im Buchladen sehen.

7. Frage: Wie geben wir Euch die Tips konkret weiter?
Antwort: Seht Ihr vor Euch (das Info). Kurzkritiken (von mir aus auch lange) werden hier veröffentlicht und die wichtigsten Daten in eine Liste übernommen, die ich führen werden. In der Liste stehen (nach Sachgebieten geordnet) die Bücher und die Nummer des Infos, in dem die entsprechende Kurzkritik (bzw. auch mehrere Nummern, wenn mehr als einer das Buch gelesen haben sollte) abgedruckt ist. Wenn Ihr Euch dann dazu durchgerungen habt, mal ein Buch zu kaufen, seht Ihr in die Liste (gibt's bei mir oder von Zeit zu Zeit im Info) und sucht Euch nach den Kritiken das billigste und hoffentlich beste Werk aus. Da so ein Buch meist weit über 50 DM kostet, dürften sich diese kurzen Nachforschungen lohnen. Natürlich könnt Ihr auch erst 10 Bücher kaufen, bis Ihr das richtige findet.

8. Frage: Kritiken über welche Bücher sind interessant?
Antwort: Einfach alle. übrigens reicht es auch, wenn Ihr in einer Buchhandlung herumgestöbert und dabei ein Buch gefunden habt, auf das Ihr andere hinweisen möchtet oder das Euch überhaupt nicht gefiel, auch wenn Ihr es nicht gekauft habt! Ein Profi wie Arnulf braucht sich ein DOS-Buch nur anzusehen und weiß schon, ob es was taugt. Dann kann er das auch ruhig sagen, damit andere, die nicht soviel darüber wissen, nicht einen Fehler begehen (indem sie kaufen oder auch nicht).

9. Frage: Wer macht den Anfang?
Antwort: Ist schon passiert. Auch wenn Ihr es vielleicht nicht bemerkt haben solltet, hat Hartmut schon einiges zum Thema "Bücher über CP/M" gesagt. In diesem Info werdet Ihr außerdem einige Kurzkritiken von meinereiner finden, wobei ich zugeben muß, daß ich nur eins der Bücher wirklich gelesen habe und mich bei den anderen auf den ersten Eindruck nach dem Durchblättern verließ. Ich halte das für legitim. Dazu aber an anderer Stelle mehr.

10. Frage: Wer darf Kritiken schreiben und wie sollen die aussehen?
Antwort: Jeder darf das und was Ihr schreibt, ist vollkommen egal. Wir sind froh, wenn überhaupt jemand sich bequemt, ein paar Zeilen aufs Papier zu werfen. Auch Bücher, die schon besprochen wurden oder die schon in der Clubbibliothek vorhanden sind, sollten von Euch ruhig nochmal kritisch beschrieben werden (wobei kritisch auch heißen kann, daß Ihr total begeistert von einem Buch seit und ohne es nicht mehr leben könnt).

11. Frage: An wen sollen die Kritiken geschickt werden?
Antwort: An Jens, Hartmut oder mich. Denn Jens veröffentlicht die Sachen, Hartmut ist für die Bibiothek zuständig und meine Wenigkeit wird die Listen mit den besprochenen Büchern führen.

12. Frage: Woher könnt Ihr die Listen bekommen?
Antwort: Die jeweils neueste Liste gibt's bei mir (man bemühe die Bundespest).

13. Frage: Wann macht der endlich Schluß mit seinen blöden Fragen?
Antwort: Jetzt.

Gerald Schröder

Bücherliste

besprochende Bücher

(keine Garantie für Preis und ISBN)

Pascal

Verlag	Autor, Titel	Preis	ISBN	Info
Franzis	Plate/Wittstock: Pascal - Einführung - Programmentwurf - Strukturen	58,-	3-7723-6903-0	21
Sybex	Tiberghien: Das Pascal-Handbuch	64,-	3-88745-614-9	21
Vieweg	Krekel/Trier: Die Programmiersprache Pascal	68,-	3-528-03337-1	21
	Kaier/Rudolfs: T.-P. Wegweiser für Microcomputer: Aufbaukurs	58,-	3-528-04443-8	21
	Hering/Scheurer: Fortgeschrittene Programmiertechniken in Turbo-P.	52,-	3-528-04467-5	21
	Fedtke: Pascal	??	3-528-04488-8	21
Heise	Wostrack: Turbo-Tools u. -Utilities unter CP/M	49,80	0123-5	21
	Rauch: Modell der Wirklichkeit - Simulation dynamischer Systeme mit dem Mikrocomputer	29,80	0524-3	21
Markt & Technik	Lüke: Turbo-Pascal	49,-	3-89090-150-6	21
Teubner	Wirth: Systematisches Programmieren	25,80	3-519-42327-8	21
	Wirth: Algorithmen und Datenstrukturen	38,-	3-519-02330-X	21
	" in Modula-2	38,-	3-519-02250-8	21
	Erbs/Stolz: Einführung in die Programmierung mit Pascal	25,80	3-519-12506-4	21
	Dal Cin/Lutz/Risse: Programmierung in Modula-2	19,80	3-519-00100-4	21
	Ottmann/Widmayer: Programmierung mit Pascal	18,80	3-519-10084-3	21
	Mittelbach/Wermuth: Turbo-Pascal in der Praxis	24,80	3-519-02544-2	21
Springer	Wirth: Programmieren in Modula-2	42,-	3-540-13301-1	21
Springer	Marthy: Methodik der Programmierung in Pascal	42,-	3-540-16611-4	21
	Däßler/Sommer: Pascal	??	3-540-12835-2	21
Springer	Remmele: Pascal - systematisch - strukturiert	??	3-540-12250-8	21
	Brosell/Zhou: Modula-2	69,-	3-540-15159-1	21
Addison-Wesley	Amstel: Programmieren: Die Entwicklung von Algorithmen in Pascal	48,-	3-925118-19-5	21
Hanser	Wilson/Addyman: Pascal	28,-	3-446-13953-2	21
	Pomberger: Softwaretechnik in Modula-2	??	3-446-14122-7	21
Vogel	Renner: Turbo-Pascal	43,-	3-8023-0758-5	21
Oldenbourg	Herschel/Pieper: Pascal und Pascal-Systeme	34,80	3-486-23685-7	21
	Herschel: Turbo-Pascal	29,80	3-486-29064-9	21
Akademische Verlagsgesellschaft	Doberkat/Rath/Rupietta: Programmieren in Pascal	??	3-400-00474-X	21
IDEA	Price: Strukturiertes Programmieren	??	3-88793-055-X	21

CP/M

Verlag	Autor, Titel	Preis	ISBN	Info
Franzis	Plate: Betriebssystem CP/M	58,-	3-7723-7522-7	14
	RPB1: CP/M kompakt	9,80	3-7723-7522-7	14
Sybex	Zaks: CP/M- Handbuch	48,-	3-88745-053-1	14
Hüthig	Weiß: CP/M - Ein Betriebssystem stellt sich vor	39,-	3-7785-1266-8	21
Markt & Technik	Dennon: CP/M - Anatomie eines Betriebssystems	68,-		14
	Digital Research: CP/M Operating System Manual	38,-	3-89090-369-X	21
IWT	Pol: Vom Umgang mit CP/M	43,-	3-88322-004-3	21

HEFT 21
September
1987

Die ersten Kurzkritiken

In diesem Info findet Ihr irgendwo einen Artikel über die Bücherliste. Hier möchte ich nun einen Teil davon vorstellen, und zwar die ersten Kurzkritiken, wobei ich hoffe, daß diese Rubrik mit Eurer Hilfe (sprich: schreibt Kurz- oder Langkritiken!) zur regelmäßigen Institution werden kann.

über welche Themen sollen die Bücher sein, die Ihr besprechen könnt? Das ist vollkommen egal. Sicherlich ist die neueste Ausgabe von Müller's Kochbuch weniger interessant für die meisten Mitglieder (alles Männer + Computerfreaks <=> jemand anders steht in der Küche). Allerdings wird das Buch schon interessant, wenn zum Kochen ein Lötkolben und zum Umrühren ein Schraubenzieher benutzt werden oder die Rezepte in einer dbase-Datei erfaßt sind. Also schreibt einfach über alles.

Gleichermaßen ist die Form schnurzpiepe. Ob ein Text, Stichpunkte, etwas Subjektives ("Zu heiß!") oder Objektives ("Bei 100 Grad heißem Wasser zeigte meine Hand Rötungen."), kurz, mittel oder lang, in hochdeutsch, plattdeutsch oder bayrisch: die Leser sollen froh sein, daß sie überhaupt was zu lesen bekommen!

Um gleich ein negatives Beispiel zu geben, habe ich ein paar Kurzkritiken verfaßt. Dabei war mir der Themenkomplex "alles" zu groß und ich habe mich auf die gezielte Suche nach Büchern gemacht, die sich mit Pascal bzw. mit CP/M beschäftigen. Selbstverständlich sollt Ihr nicht auch verzweifelt nach Büchern suchern, über die Ihr Kritiken schreiben könnt, aber ich hatte hochoffiziellen Auftrag von unserem Oberguru, äh, Pardon: Ober*mann*, Bücher aus diesen Bereichen anzuschauen.

Meine Methode war zugegeben dilettantisch: ich habe mir einfach überlegt, welche Anforderungen ich an die Bücher stellen würde und versucht, diese zu überprüfen. Meine Kriterien waren bei Pascal darauf abgestimmt, für Anfänger geeignete Bücher zu finden. Wie gesagt, sind die Kriterien stark subjektiv. Ein Buch sollte also (nach meiner unmaßgeblichen Meinung) leicht verständlich formuliert sein, damit auch Hobby-Programmierer es verstehen können. Wichtig ist, daß das Buch gut gegliedert ist, zu jedem Kapitel Übungsaufgaben zur Überprüfung des Gelernten bietet und im Anhang über eine vollständige Liste der Pascal-Syntax in Form von Syntaxdiagrammen verfügt. Außerdem sollte die Aufmachung einigermaßen professionell sein (keine Fotokopien oder schlecht lesbaren Listings). Letztes Kriterium war der besprochene Pascal-Dialekt: vorzuziehen wäre eine allgemeine Besprechung (also über Standard-Pascal) mit mindestens einem Kapitel über die speziellen Eigenschaften von Turbo-Pascal, weil dies der für unseren Rechner wohl die am ehesten verfügbare (vollständige) Pascal-Version ist. Eine Einführung in Turbo-Editor und -Compiler wäre auch nicht schlecht. Ganz raus fallen auf jeden Fall Bücher, die sich mit UCSD-Pascal beschäftigen.

Wie habe ich diese Kriterien angewandt? Da ich kaum alle Bücher durchlesen konnte, habe ich mir jedes nur kurz angeschaut und die folgenden Punkte überprüft:

- Inhaltsverzeichnis übersichtlich?
- Einleitung gut lesbar und verständlich?
- übungsaufgaben vorhanden?
- Syntaxdiagramme vorhanden?
- Turbo-Pascal erwähnt?
- Druck in Ordnung?
- wie sehen die Programm-Beispiele (Listings) im Buch aus?

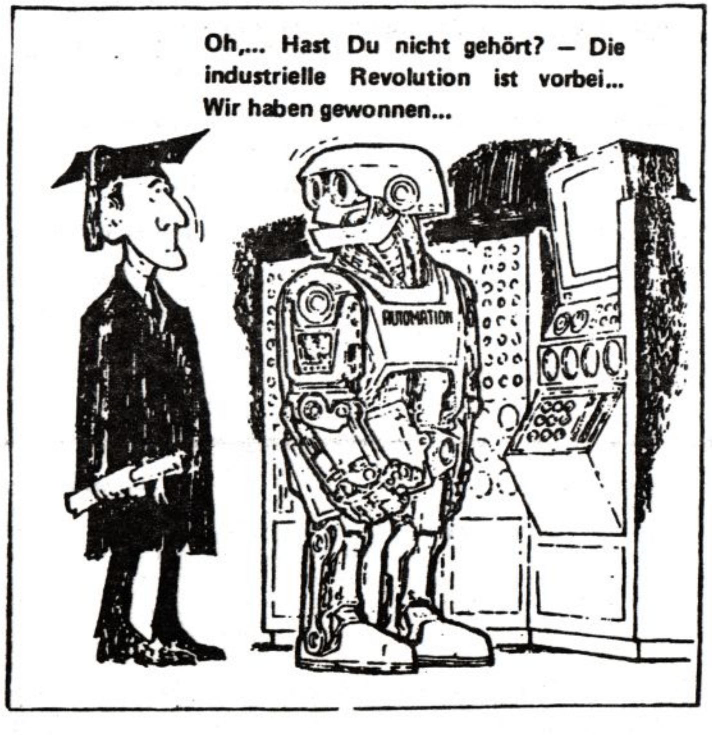
Vielleicht sind für Euch andere Dinge wichtiger, aber der absolute Anfänger bekommt mit dieser Liste vielleicht eine Vorstellung, wie er sein erstes Pascal-Buch auswählen könnte. Genug Auswahl gibt es jedenfalls. Selbst ohne UCSD-Pascal-Bücher habe ich noch ca. 50 Exemplare allein aus Katalogen herausgefischt, wobei man nach den Titeln meist schlecht sagen kann, welche Bücher für Anfänger geeignet sind. Aber vielleicht finden auch fortgeschrittene Programmierer bei den besprochenen Büchern welche, die sie interessieren.

Bei CP/M war es mir dagegen unmöglich, solche Kriterien aufzustellen. Vielleicht ist es wichtig zu wissen, wie die mitgelieferten Programme von Digital Research (wie ED) funktionieren. Aber wer wird sie schon benutzen, wenn er sofort bessere bekommen kann? Die Befehle der DOS-Oberfläche sind natürlich wichtig, aber die sind schnell erklärt. Anders sieht das schon mit der Programmierung aus, aber gehört das wiederum schon in ein Anfänger-Buch?

Hier überlasse ich gern kompetenteren Leuten das Feld (hoffentlich finden sich welche). Mit der Literatur zu CP/M sieht es eher spärlich aus. Meine Kritiken zu CP/M-Büchern sind auf jeden Fall mit noch mehr Vorbehalt als die zu den Pascal-Büchern zu lesen.

Im Anschluß an die Kritiken möchte ich Euch noch eine Liste mit Büchern vorlegen, die ich zwar in einem Katalog gefunden habe, mir aber weder in der Bibliothek noch in einer Buchhandlung anschauen konnte (kaufen wollte ich sie verständlicherweise nicht). Es wäre toll, wenn jemand sich auf das ein oder andere Buch stürzen würde. Vielleicht habt Ihr es ja gelesen oder Euch intensiv irgendwo angeschaut. Schreibt dann doch mal was darüber!

Gerald Schröder



Kurzkritik

Als ich unserem Freund Gerald Schröder während des Vorstandstreffens die Arbeit auf's Auge drückte, eine Liste der Bücher zu erstellen, die der CLUB 80 für die Clubbibliothek anschaffen sollte, habe ich nicht damit gerechnet, daß er so gründlich vorgehen würde. Ich möchte mich an dieser Stelle für sein großes Engagement dadurch herzlich bedanken, daß ich eine Kurzkritik über eines der Bücher verfasste, derer er nicht habhaft werden konnte!

Kaier/Rudolfs: Turbo-PASCAL Wegweiser für Mikrocomputer
Grundkurs (Vieweg, 48,--DM)

Das Buch ist für den geschrieben, der sich ohne große fremde Hilfe in die Sprache Turbo PASCAL einarbeiten will. Daher ist auch das erste Kapitel den allgemeinen Grundlagen der Datenverarbeitung und Programmentwicklung vorbehalten. Auf den folgenden 15 Seiten wird dann die Bedienung des Turbo PASCAL-Compilers recht anschaulich beschrieben, wobei aber immer wieder ausdrücklich auf das Handbuch verwiesen wird. In diesem Kapitel ist auch eine Auflistung der im Grundkurs behandelten Befehle zu finden, die allerdings nicht sehr übersichtlich gestaltet ist! In den folgenden Kapiteln wird, beginnend bei "Lineare Programme" bis zu "Array als strukturierter Datentyp", der Leser in gut gegliederten Lernabschnitten langsam in die Programmierung mit PASCAL eingeführt. Dabei werden die einzelnen Schritte jeweils an Beispielen erklärt und durchgearbeitet.

Fazit: Das Buch ersetzt keineswegs das Originalhandbuch, führt aber in sehr anschaulicher und leicht verständlicher Weise in die Programmierung mit Turbo PASCAL ein und ist gerade für Autodidakten genau die richtige Einstiegslektüre. Weiterführend ist der Aufbaukurs aus der gleichen Buchserie sehr zu empfehlen!

Hartmut Obermann

Kurzkritiken

Kaier/Rudolfs: Turbo-Pascal-Wegweiser für Mikrocomputer
(Vieweg, DM 58,-)

Dieses Buch ist auf Turbo-Pascal zugeschnitten; allerdings konnte ich mir den 1. Band (Grundkurs) nicht anschauen. Es ist sehr ausführlich und scheint mir gut verständlich. Der Druck der mir vorliegenden Ausgabe ließ allerdings zu wünschen übrig. Zum Inhalt: Gezeigt wird alles erstes die Bedienung des Turbo-Pascal-Editors, was sehr wichtig ist. Behandelt werden dann (fortgeschrittene) Datenstrukturen (Sets, Records, Files, Pointer) und verschiedene Algorithmen (Rekursion, Sortieren, Baumu. Datei-Behandlung). Ohne den ersten oder einen anderen einführenden Band ist dieses Buch nicht zu empfehlen.

Wirth: Systematisches Programmieren
(Teubner, DM 25,80)

Wie alle Werke von Wirth, dem Pascal-"Erfinder", ist dies ein Standardwerk, zumindest für jeden Informatik-Studenten. Die Sprache ist relativ trocken (eben wissenschaftlich), aber Wirth liefert hier eine gute, grundlegende Einführung, die für jeden verständlich ist. Er arbeitet viel mit Flußdiagrammen, was seine Programmbeispiele sehr schön veranschaulicht. Dieses Buch gehört zu den ältesten (vielleicht ist es das älteste) Pascal-Büchern. Deshalb finden sich auch keinerlei Dialekt-Einflüsse. Es handelt sich um "reines" Standard-Pascal.

Wirth: Algorithmen und Datenstrukturen
(Teubner, DM 38,-)

Dieses ist das zweite Standardwerk von Wirth. Er geht hier anhand von Pascal auf verschiedene Programm-Teile und Daten-Arten ein, die oft im normalen Programmierbetrieb benötigt werden und in anderen Sprachen oft schlecht zu realisieren sind. Die großen Themenkomplexe lassen sich unter "Bäume", "Sortieren" und "Arrays" zusammenfassen. Das Buch richtet sich an Leute, die schon mit Pascal programmiert haben. Leider wird im Anhang die Pascal-Syntax (welche Befehle sehen wie aus?) nur in der Backus-Naur-Form angegeben, die ich für schlecht lesbar halte. Positiv aber ist zu vermerken, daß es Übungsaufgaben in dem Buch gibt.

Wirth: Algorithmen und Datenstrukturen in Modula-2
(Teubner, DM 38,-)

Dieses Buch habe ich aufgenommen, weil es die Zukunft bedeutet. Wirth hat nach Pascal eine Pascal-ähnliche Programmiersprache definiert, die besonders den modularen Aufbau eines Programms betont, eben Modula-2 (die "2" steht für zweite Version). Die Themen entsprechen dem oben besprochenen Pascal-Band, die Programme sind fast identisch, nur eben in Modula formuliert, was sie in meinen Augen eleganter erscheinen läßt. Leider wird auch die Syntax von Modula nur in BNF angegeben.

Erbs/Stolz: Einführung in die Programmierung mit Pascal
(Teubner, DM 25,80)

Dieses Buch ist eine echte Einführung, die sich auch an den Computer-Neuling richtet. Es ist locker formuliert und mit vielen Zeichnungen verziert. Leider entstand es auch noch vor der Turbo-Zeit, aber dafür lehrt es eben das "reine" Pascal. Ich würde das Buch für den echten Anfänger sehr empfehlen, da es auch Übungsaufgaben zur eigenen Überprüfung bietet.

Dal Cin/Lutz/Risse: Programmierung in Modula-2
(Teubner, DM 19,80)
Wer schon in Pascal programmiert hat, wird sich vielleicht über Modula-2 gefreut haben, denn es behebt die alten Pascal-Schwächen und bietet neue Möglichkeiten, die in der Pascal-Definition nicht vorgesehen waren und erst von Pascal-Dialekten eingeführt wurden. Dieses Buch will nun dem Umstieg von Pascal auf Modula-2 erleichtern. Es stellt alle Unterschiede klar heraus und arbeitet sich von elementaren Programmstrukturen zu den Feinheiten vor, die systemnahe Programmierung erlauben. Alle neuen Elemente werden anhand von Listings gezeigt. Die Wirth'schen Systemmodul-Definitionen wind ebenso abgedruckt wie die Syntaxdiagramme (!!!) und die BNF zur Modula-2-Syntax. Ein Buch für Umund Aufsteiger.

Wirth: Programmieren in Modula-2
(Springer, DM 42,-)
Mir lag von diesem Buch nur die englische Ausgabe vor, die ich hiermit beschreiben will. Das Buch scheint gut geschrieben, eben ein echter Wirth, wieder nur mit der BNF zur Syntax-Definition. Das Buch richtet sich nicht nur an den fortgeschrittenen Pascal-Programmierer und wird wohl auch bald als Standardwerk gelten.

Marthy: Methodik der Programmierung in Pascal
(Springer, DM 42,-)
Dieses Buch ist als Einführung gedacht, wenn der Titel auch nicht so klingt. Es ist kurz gehalten, spricht aber alle wichtigen Themen an. Es scheint verständlich, allerdings fehlen übungsaufgaben.

Herschel/Pieper: Pascal und Pascal-Systeme
(Oldenbourg, DM 34,80)
Dieses Buch ist gute Einführung in Pascal und bietet einen Gesamtüberblick. Erklärt wird das Standard-Pascal, aber am Ende des Buches werden die meisten gebräuchlichen Pascal-Dialekte noch angesprochen. Leider sind keine übungsaufgaben vorhanden, die das Buch zu einer hervorragenden Einführung werden ließen. Es werden die Bestandteile eines Pascal-Systems (Editor usw.) allgemein und einige Pascal-Implementierungen beschrieben. Sogar die Hauptmerkmale von Modula werden erwähnt.

Krekel/Trier: Die Programmiersprache Pascal
(Vieweg, 68,-)
Das Buch rechtfertigt seinen Preis allein durch den Umfang, der aber für mich eher abschreckend wirkte. Der Druck dagegen ist billig. Für den normalen Menschen ist das Buch viel zu ausführlich. Es enthält unter anderem 400 Querverweise auf andere Literatur, die kaum jemanden interessieren dürften.

Däßler/Sommer: Pascal
(Springer, DM ??)
Dieses Buch besteht aus zwei Teilen. Der erste Teil ist eine kurze, sehr trockene Einführung in Pascal, die aber durchaus für Anfänger geeignet ist. Der zweite Teil enthält einen vollständigen DIN-Entwurf zu Pascal. Dieser Teil ist bestimmt für jeden interessant, der Beamter ist oder werden will.

Remmele: Pascal - systematisch - strukturiert
(Springer, DM ??)
Diese Einführung ist zwar trocken geschrieben, aber sehr ausführlich. Gleichzeitig mit Pascal können noch Struktogramme und ihre Anwendung gelernt werden. Damit erleichtert man sich später den Weg von der Idee zum Programm und ist gleichzeitig up to date, denn Flußdiagramme sind out. Es fehlen aber übungen.

Doberkat/Rath/Rupietta: Programmieren in Pascal
(Akademische Verlagsgesellschaft, DM ??)
Diese Buch hat nur eine Zielgruppe: Lehrer. Es ist billig aufgemacht. Die eingestreuten Syntaxdiagramme werden nicht im Anhang zusammengefaßt und es gibt keine übungsaufgaben.

Price: Strukturiertes Programmieren
(IDEA, DM ??)
Hier haben wir eine einfach zu lesende, sehr gut strukturierte und übersichtliche Einführung. Sie ist ausführlich und gut für Anfänger geeignet. Leider fehlen Syntax-Diagramme, dafür gibt es aber reichlich übungsaufgaben. Das Buch geht nicht tief und sollte später durch ein fortführendes Buch (Wirth's Algorithmen u. Datenstrukturen o.ä.) ergänzt werden.

Plate/Wittstock: Pascal: Einführung - Programmentwurf - Strukturen
(Franzis, DM 58,-)
Dieses äußerst dicke Buch ist für den Anfänger gedacht. Es liefert sowohl Syntaxdiagramme als auch übungsaufgaben. Es sind viele Listings abgedruckt, deren Druckqualität aber nicht so gut ist, weil sie nur fotokopiert und nicht neu gesetzt wurden. Auf Turbo-Pascal wird auch kurz eingegangen.

Tiberghien: Das Pascal-Handbuch
(Sybex, DM 64,-)
Dieses Buch sollte statt "Handbuch" eher "Lexikon" heißen, denn es liefert eine sortierte Stcihwortsammlung zu allem, was zu Pascal irgendwie dazugehört. Natürlich fehlen dabei auch nicht dialektspezifische Sachen. Vieles dürfte zwar geeignet sein, aber einiges ist wohl so weit von unserem Programmieralltag entfernt, daß es für immer ungelesen bleiben würde. Wer nicht intensiv in Pascal einsteigen will (alles lesen, alles eintippen, alles verstehen, für jeden Rechner), kann sich dieses Buch schenken.

Hering/Scheurer: Fortgeschrittene Programmiertechniken in Turbo-Pascal
(Vieweg, DM 52,-)
Dieses Buch hat zwei Anliegen: 1. Den Aufwand beim Entwickeln von Programmen verringern, indem allgemein oft gebrauchte Routinen vorgestellt werden. 2. Den Entwicklungsweg, das Programmieren, einzustudieren und allgemeine Hilfen zu geben. Dadurch soll wiederum der Aufwand verringert werden, aber hierbei liegt die Betonung auf der überlegung vor dem Eintippen (was kann ich von anderen Programmen verwenden usw.). Es werden auch Übungen angeboten. Viele Sachen beziehen sich allerdings auf MS-DOS. Dieses Buch ist nur für fortgeschrittene Programmierer gedacht!

Fedtke: Pascal: Algebra - Numerik - Computergrafik
(Vieweg, DM ??)
Dieses Buch zeigt nur Anwendungen von Pascal, und zwar in den drei im Titel genannten Gebieten. Es sind viele Zeichnungen vorhanden, aber trotzdem sollte niemand das Buch in die Hand nehmen, der nicht wirklich am Thema interessiert ist, denn die Sachen sind nicht so einfach zu verstehen (von der Theorie her).

Wostrack: Turbo-Tools und -Utilities unter CP/M
(Heise, DM 49,80)
Hier werden kleinere Hilfsprogramme als Listings vorgestellt, die das Programmieren erleichtern sollen. Ich glaube nicht, daß es sich lohnt, dieses Buch zu kaufen, denn die Programme gibt es bestimmt auf Diskette und da gehören sie auch hin (wer tippt das schon alles ab). Die Erläuterungen hätten als Doku-File dazugehört, sollten hier wohl aber als Käuferfang dienen (oder was weiß ich).

Rauch: Modell der Wirklichkeit - Simulation dynamischer Systeme mit dem Mikrocomputer
(Heise, DM 29,80)
Hier haben wir ein sehr interessantes Buch über Modelle und Modellbildung mit vielen Listings. Zwar wurde Turbo-Pascal verwendet, aber auf einem umgebauten Apple und mit vielen Grafikund sonstigen Erweiterungen in der Sprache. Diese Anpassungen (der Software) sind sorgfältig dokumentiert und müßten nachvollzogen werden, um sie auf einer anderen Maschine lauffähig zu machen. Das bedeutet viel Arbeit und ist nur für den absolut Interessierten zu empfehlen.

Lüke: Turbo-Pascal
(Markt & Technik, DM 49,-)
Dieses Buch bezieht sich vor allem auf die MS-DOS-Version des Turbo-Pascal, könnte aber trotzdem für uns geeignet sein. Es liefert keine Syntaxdiagramme, scheint aber vollständig und ausführlich zu sein. Zwar fehlt (zumindest im Inhaltsverzeichnis) etwas die Strukturierung, aber es werden auch einige Turbo-Tools bzw. Dienstprogramme beschrieben.

Ottmann/Widmayer: Programmierung mit Pascal
(Teubner, DM 18,80)
Dieses Buch ist für den absoluten Anfänger gedacht. Es liefert Syntaxdiagramme, aber keine Übungen. Der Stoff wird nicht sehr vertieft und es werden nur Programmteile als Beispiele angeführt, was die Sache übersichtlicher, aber nicht gerade leicht nachvollziehbar macht (eintippen und laufen lassen geht nicht!). Der Druck ist vorbildlich und macht das Lesen zur Freude.

Brosell/Zhou: Modula-2
(Springer, DM 69,-)
Dieses Buch ist eine dicke, vollständige Einführung in Modula, die wohl auch von einem Anfänger verstanden werden kann. Es sind viele Listings vorhanden.

Amstel: Programmieren: Die Entwicklung von Algorithmen in Pascal
(Addison-Wesley, DM 48,-)
Eigentlich stimmt der Titel dieses Buches nicht. Das "in Pascal" dürfte bestenfalls in Klammern stehen, denn Amstel geht nur um den Weg der Entwicklung von Algorithmen, wobei Pascal nur zufällig die Sprache ist, in die das Endergebnis der Überlegung übertragen wird. Dementsprechend werden viele Pascal-Eigenschaften wie Dateien, Mengen und Pointer überhaupt nicht angesprochen. Ein irreführender Titel.

Wilson/Addyman: Pascal
(Hanser, DM 28,-)
Dieses Buch bietet meiner Meinung nach äußerlich alles, was ein gutes Pascal-Lehrbuch ausmacht: Übungen, Syntaxdiagramme, verständliche Sprache, Ausführlichkeit, guten Druck. Ob der Inhalt das hält, kannich nicht sagen, aber die 28,- DM sind ein gutes Argument, es mal zu probieren.

Mittelbach/Wermuth: Turbo-Pascal in der Praxis
(Teubner, DM 24,80)
Dieses Buch ist nicht für den Anfänger geeignet, es ist für Umsteiger gedacht, die sich nach einem anderen Pascal nun in Turbo versuchen wollen. Evtl. könnte man es als Ergänzung zu einem allgemeinen Pascal-Anfängerbuch nehmen. Es wird eine Einführung in den Turbo-Editor angeboten. Das Buch bezieht sich sowohl auf CP/M als auch auf MS-DOS.

Renner: Turbo-Pascal
(Vogel, DM 43)
Dieses Buch ist nicht unbedingt für Anfänger geeignet; der Leser sollte schon vorher programmiert haben, wenn auch nicht unbedingt in Pascal. Der Inhalt bezieht sich auf MS-DOS, aber Editor/Compiler (Optionen) sind gut und ausführlich dokumentiert.

Herschel: Turbo-Pascal
(Oldenbourg, DM 29,80)
Dies ist eine elementare Einführung für Anfänger. Leider fehlen Übungsaufgaben, aber Syntaxdiagramme sind vorhanden. Der Editor wird auch beschrieben, allerdings beziehen sich viele Dinge auf MS-DOS.

Pomberger: Softwaretechnik in Modula-2
(Hanser, DM ??)
Mit diesem Buch sollen Leute angesprochen werden, die viel programmieren und dabei große Programmsysteme herstellen. Methoden und Arbeitsweise werden dabei direkt mit der dafür bestens geeigneten Sprache Modula-2 beschrieben. Ohne zumindest elemntare Kenntnisse von Pascal wird es schwer sein, dieses Buch zu verstehen. Dafür wird Modula so gut erklärt, daß es auch als Modula-Einführung für Umsteiger geeignet ist. Das Buch ist für berufliche u. Vollblut-Programmierer.

Digital Research: CP/M-Operating System Manual
(Markt & Technik, DM 38,-)
Bei diesem Buch handelt es sich um einen Nachdruck (bzw. mehr eine Kopie) der englischen Originalausgabe. Die Druckqualität ist schlecht und die Druckfehler wurden nicht korrigiert. Dafür hat man aber die Original-Dokumentation der CP/M-Vertreiber in der Hand, die auch die Struktur des Betriebssystems einigermaßen gut vermitteln. Leider sind die Programmbeispiele unbrauchbar, denn 1. sind sie in 8080-Maschinensprache geschrieben, 2. sind es nur Programmrümpfe (an den wichtigen Stellen: setzen sie hier Ihre Routine ein) und 3. sind sie, falls ausführlich, auf ein vollkommen unbrauchbares (jedenfalls für unsere Zwecke) System zugeschneidert. Empfehlen kann ich es nicht.

Weiß: CP/M - ein Betriebssystem stellt sich vor
(Hüthig, DM 39,-)
Dieses Buch stellt nur die CCP-Befehle und die normalerweise vorhandenen Dienstprogramme vor. Zum Programmieren mit CP/M ungeeignet.

Pol: Vom Umgang mit CP/M
(IWT, DM 43,-)
Dieses Buch ist sehr dick und ausführlich. Es erklärt alle Befehle, Hilfsprogramme und was man sonst noch braucht. Aber der Anhang ist nicht ausreichend, um mal schnell etwas nachschlagen zu können. Außerdem beziehen sich alle Programmbeispiele auf den 8080. Aber das Buch ist für den absoluten Computer-Neuling bestens geeignet, denn es führt elementar in die Benutzung des Computers ein (wie mache ich ein Backup usw.).

Pascal

(Bücher, die evtl. interessant wären)
(keine Garantie für Preis und ISBN)

Verlag	Autor, Titel	Preis	ISBN
Franzis	Busch: Der sichere Einstieg in P.	48,-	3-7723-7861-7
	Klein: Was ist Pascal?	32,-	3-7723-7001-2
Sybex	Miller: Pascal-Programmme (60 Algorithmen)	58,-	3-88745-007-8
	Hunter: 50 Pascal-Programme	48,-	3-88745-065-5
	" " (Turbo)	48,-	3-88745-645-9
	Bromm: Das Arbeitsbuch zu T.-P.	48,-	3-88745-629-7
	Rollke: Das Turbo-Pascal-Buch	42,-	3-88745-608-4
Hüthig	Fischer: Datenverarbeitung mit Turbo-Pascal	45,-	3-7785-1476-8
	Lange: Turbo-Pascal - Ständig im Griff (Referenz)	21,80	3-7785-1313-3
Vieweg	Harbeck: Einführung in T.-P. unter CP/M 80	32,-	3-528-04440-3
	Kaier/Rudolfs: T.-P. Wegweiser für Microc.: Grundkurs	48,-	3-528-04415-2
	Schneider: Einführung in Pascal	26,-	3-528-04200-1
	Pinke: Strukturiertes Programmieren in Pascal	35,-	3-528-04439-X
Heise	Feller: Pascal-Datenstrukturen u. prakt. Anwendungen (fortgesch.)	44,-	7020-8
	Gilder/Barrus: 112 T.-P.-Programme für math. und elektrotechn. Anwendungen (Anfänger)	49,80	3-88229-102-8
	Claßen: Mathematikprogramme in P.	36,80	3-620-00114-6
	Stede: T.-P.-Programme zur künstl. Intelligenz	?	9126-5
	Eisenecker: Matcalc - Matrizenrechnung in UCSD-/T.-P.	34,80	3-88229-143-5
Markt & Technik	Ciric/Thies: Turbo-P. Kursbuch 1.: Einführung	49,-	3-921803-34-9
	Kassera: Programmieren mit T.-P. (Ver. 3.0, MS-DOS)	59,-	3-89090-159-X

Verlag	Autor, Titel	Preis	ISBN
Teubner	Bielig/Schulz: 3-D-Graphik in P.	25,-	
	Erbs: 33 Spiele mit Pascal	32,-	
	Weber: Pascal in Übungsaufgaben	24,80	3-519-02539-6
Humboldt	Schmuck: Programmiersprache Pascal Schritt für Schritt	9,80	ht 551
Springer	Blaschek/Pomberger: Einführung in die Programmierung mit Modula-2	45,-	3-540-15158-3
	Gleaves: Modula-2 für Pascal-Programmierer	39,-	3-540-15157-5
Girardet	Weber/Mrowka: Grundkenntnisse P.	5.95	22910
Taylorix	Hamann/Lewerenz: Datenverarbeitung mit Turbo-Pascal	46,-	3-88640-037-9
Addison-Wesley	Angeli: A-W Kompaktführer T.-P.	22,80	3-925118-15-2
Hanser	Joepgen: Turbo-Pascal	58,-	3-446-14431-5
Vogel	Baumann: Programmieren mit Pascal (Anfänger)	25,-	3-8023-0667-8
Oldenbourg	Blaha/Haselberger/Schärf: Programmiertraining Pascal (Bd. 1)	26,80	3-486-20003-8
	Schärf/Vertneg: P. für Kaufleute	29,80	3-486-20000-3
	Vahrenkamp: P. auf Mikrocomputern	28,80	3-486-28921-7
Dümmlers	Heidemann: Algorithmen u. Datenstrukturen	8,80	3-427-42771-4
Bibl. Institut	Kaucher/Klatte/Ullrich: Pascal	26,80	3-441-05796-3
rororo	Ledgard/Singer: Pascal auf der Spur	12,80	8109
IWT	Rompel: Turbo-Pascal-Anwendungen	48,-	3-88322-167-8
	Bürgel: Modula-2	68,-	3-88322-169-4
McGraw Hill	Sand: Pascal-Programmiertechnik für Fortgeschrittene	35,-	3-89028-039-0
Expert	Scheibl: Pascal-Einführung	48,-	3-8169-0093-3
Vahle	Schönemeier: Turbo-P. Softregister	13,80	3-9800252-2-5

Verlag	Autor, Titel	Preis	ISBN
???	Kwiatowski/Dierig: Pascal-Computerspiele	20,80	3-7724-5374-0
	Welsh/Elder: Einführung in P.	42,-	3-437-40109-2
	Reid-Green: Computerspiele u. Grafik	56,-	3-528-04357-1

CP/M

Verlag	Autor, Titel	Preis	ISBN
Sybex	Miller: Programmieren mit CP/M	52,-	3-88745-077-9
	Steinhoff: SYBEX-Ratgeber CP/M	55,-	3-88745-301-8
Hüthig	Weiß: CP/M - Ständig im Griff (Referenz)	21,80	3-7785-1267-6
Vieweg	Schneider: Einführung in die Anwendung des Betriebssystems CP/M	29,80	3-528-04252-4
Markt & Technik	Digital Research: CP/M Plus (3.0)	38,-	3-89090-371-1
	D. R.: Programmer's Utilities and SID for CP/M 2.2/3.0	49,-	3-89090-372-X
Oldenbourg	Blackburn/Taylor: Terminal-Buch CP/M	24,80	3-486-29331-1
Addison-Wesley	Boon/Pelsmaeker: A-W Kompaktführer CP/M	22,80	3-925118-13-6
Vahle	Bürenkämper: CP/M-Softregister	13,80	3-9800252-6-8
Girardet	Pape: Das Mikrocomputer-Betriebssystem CP/M	5,95	3-7736-2359-3
Röckrath	Kämpf: CP/M-Assembler-Listing	59,-	3-925074-11-2

- IMPRESSUM -

1. Vorsitzende — **Hartmut OBERMANN**
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

2. Vorsitzende — **Gerald SCHRÖDER**
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

Hardwarekoordinator — **Eckehard KUHN**
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar — **Werner FÖRSTER**
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion — **Jens NEUEDER**
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck — **Peter Spieß**
Trugenhoferstraße 27
8859 Rennertshofen 1
☎ 08434 /454

Bankverbindung des CLUB 80
Postgirokonto: Peter STEVENS, Sonderkonto CLUB 80
Konto-Nummer: 285 491 - 465
Postgiroamt: Dortmund
BLZ: 440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

HEFT 21
September 1987

SCHLUSS

Hallo Club-80er,

entgegen aller Versprechungen kann ich leider erst jetzt das neueste INFO herausgeben, ich hoffe, Ihr habt trotzdem Eure Freude daran.

Es ist ja nun schon fast (unabsichtlich!!) zur Gewohnheit geworden, daß das INFO immer etwas später bei Euch eintrudelt. Ich kann aber im Moment die von mir selbst gesetzten Wunschtermine für das Erscheinen der INFO's zu meinem Bedauern nicht halten, zuviele private und berufliche Termine stehen ins Haus und müssen bewältigt werden. Ein Ende meines persönlichen "Terminboom's" ist vor dem Frühjahr/Sommer 88 nicht in Sicht, deshalb werde ich Euch also noch ein paar INFO's mit "unregelmäßigem" Erscheinungsdatum zumuten müssen.
Ein Informationsverlust entsteht bekanntlich dadurch nicht, unser INFO wird höchtens dicker, je länger ich "sammle". Ich hoffe, Ihr habt Verständnis dafür.

Nun möchte ich mich noch etwas erfreulicheren Dingen zuwenden. Diesem INFO liegen zwei "Extrawerke" bei:
- Zum Einen ist es eine Kurzanleitung die ich auf kartoniertes Papier kopiert habe. Etwas stabilisiert -damit Ihr sie auch richtig nutzen könnt-!
- Zum Anderen gibt es eine "Stichwort-Infothek", die als Unterstützung beim Suchen nach INFO-Artikeln dient.

Das war's auch schon wieder.
Ich wünsch Euch viel Spaß mit Dieser -
und bis zur neuesten INFO grüßt Euch

Jens

HOFACKER

ING. W. HOFACKER GMBH

Ing. W. Hofacker GmbH · Tegernseer Straße 18 · 8150 Holzkirchen

Herrn
U. Böckling
Gartenstr. 31

5410 Höhr-Grenzhausen

Der große Fachverlag für
- Elektronik
- Neue Technologien und Medien
- Home- and Personalcomputer
- Software für Microcomputer

Seit 1968

Postanschrift:
HOFACKER-VERLAG
Ing. W. Hofacker GmbH
Tegernseer Straße 18
D-8150 Holzkirchen/Obb.
Telefon: 0 80 24/73 31
Telex: 52 69 73
Sitz der Gesellschaft: Holzkirchen

Ihr Zeichen	Ihre Nachricht	Unsere Nachricht	Unser Zeichen	Datum
			JM	21. 7. 1987

Sehr geehrter Herr Böckling,

vielen Dank für Ihren Brief vom 20. 7. 1987 bezüglich Beiträgen zu TRS-80.

Ich bin damit einverstanden, daß Sie Beiträge bezüglich TRS-80 aus der ELCOMP in Ihrer Clubzeitschrift veröffentlichen. Ich bitte Sie aber, zu jedem Beitrag eine Quellenangabe zu machen (ELCOMP Heft Nr... Hofacker Verlag, Holzkirchen).
Bitte schicken Sie mir doch gelegentlich ein Belegheft.

Vielleicht können Sie auch noch auf die folgenden Spiele (Cassette) hinweisen, die bei uns noch für TRS-80 lieferbar sind:

- TRS-80 OPERA
- The Great Race
- Owl Tree

Jede Cassette ist für DM 5,-- erhältlich. Lieferung gegen Vorkasse auf unser Postscheckkonto München, Kto.-Nr. 15 994-807, BLZ 700 100 80 oder per Nachnahme (+ DM 6,50 NN-Gebühr).

Mit freundlichen Grüßen
Ing. W. Hofacker GmbH Verlag

J. Maier

PS.: Ich gehe davon aus, daß Ihre Club-Infos nicht kommerziell verbreitet werden, sondern nur an Ihre Clubmitglieder abgegeben werden.

CLUB 80

CLUBINFO
DER
TANDY -
GENIE -
UND KOMTEK -
ANWENDER

22. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
POSTFACH 27 / TEL.: 06124 / 3913

- INHALTSVERZEICHNIS -

Seite:
und Autor:

Clubinternes

Neues vom Vorstand .... 1 - 2
Hartmut Obermann
Geburtstagskinder 3
Jutta Obermann
Moin, Moin 4
Hartmut, Kurt, Gerald
Club-Emblem .... 5 - 6
Arnulf, Heinrich, Klaus
Vorstellungen .... 7 - 9
Hartmut, Alexander, Uli
Termine / Messen 10
Redaktion

Software

Der schnellste Zahnarzt .... 11 - 12
DA CAPO .... 13 - 16
Schwere Kavallerie .... 17 - 19
Klaus-Jürgen Mühlenbein
Von 8 auf 24 Pins .... 20 - 22
TSCRIPS und die 24 Nadeln .... 23 - 25
Arnulf Sopp
Noch ein Bildschirmschoner .... 26 - 27
Alexander Schmid
Ein weiterer Bildschirmschoner ..... 28 - 30
Arnulf Sopp
Etwas für Leute, die nie Zeit haben 31
Es geht noch schneller .... 32 - 33
Alexander Schmid
Ein Fehler in SYS28/SYS .... 33 - 34
The Times, They Are A-Changing ..... 35 - 36
Arnulf Sopp
BOBS - Basic Datenbanksystem .... 37 - 39
Klaus Hermann
PASCAL - Teil 2 .... 40 - 43
Rüdiger Sörensen
RPNL .... 44 - 52
Kurt Müller
Vergleich zweier Disk-Dateien ...... 53 - 56
Vergleich zweier Programme .... 57 - 59
Arnulf Sopp

Seite:
und Autor:

Hardware

Die Maus am TRS 80 .... 60 - 62
Hartmut Obermann
Das 80-Zeichen-Karten-Problem ... 63 - 66
Helmut Bernhardt
Schwimmende Bilder 66
Alexander Schmid

Börse

Wer hat was -- wer will was .... 67 - 70

Sonstiges

Prüfzahlfehler beim Lesen 71
Helmut Bernhardt
Fraktur 72
Klaus-Jürgen Mühlenbein
Sicherer Platz für Daten .... 73 - 77
Artikel aus TEST
Weihnachtsbummel (u. a.) 78
Spatzel vom Mausebär
32 Bit zum Niedrigpreis .... 79 - 80
Artikel aus ...

Programmbibliothek

Neues von der Diskothek .... 81 - 82
Werner Förster

Seite:
und Autor:

Club Bibliothek

Club-80 - Bücherei 83
Hartmut Obermann
Buchvorstellungen .... 84 - 86
.. Verschiedene

Die letzten Seiten

Impressum 87
Schluß 88
Redaktion
Clubmitgliederadressen .... am INFO-Ende
Redaktion
Sonderausgaben .... am INFO-Ende
Hans-Martin Stephan, Redaktion

Neues vom Vorstand

Hallo Freunde,

ich hoffe, daß euch dieses Info noch vor der Jahreswende 1987 -> 1988 erreicht und wünsche euch deshalb einen guten Rutsch und ein ebensolches neues Jahr! Sollte sich das Info doch verspäten, gelten diese Wünsche natürlich genau so nachträglich, wie die Weihnachtgrüße, die ich hiermit aussprechen möchte.

Zum Jahreswechsel habe ich eine gute und eine schlechte Nachricht für euch.
Zunächst die schlechte: bis spätestens zum 15. Februar muß jeder seinen Jahresbeitrag 1988 bezahlt haben. Diejenigen, die diesen Termin versäumen, bekommen das nächste Info nicht mehr und werden, wenn sie auch die dann folgende Mahnung außer acht lassen, aus dem Club ausgeschlossen.
Und nun die gute: da der **CLUB 80** im letzten Jahr sehr sparsam gewirtschaftet hat, bekommen alle Mitglieder, die vor dem 30.06.1987 dem Club beigetreten sind, eine Beitragsrückerstattung von 10,-- DM! Bei den Mitgliedern, die zwischen dem 01.07 und dem 31.12.1987 beigetreten sind, reduziert sich die Rückzahlung auf 5,-- DM. Um den Verwaltungsaufwand möglichst gering zu halten, wird der jeweilige Betrag vom zu zahlenden Jahresbeitrag 1988 abgezogen (beachtet dazu die den Infos beigelegten überweisungsformulare). Mitglieder, die den Club zum Jahreswechsel verlassen, kommen nicht mehr in den Genuß der Rückzahlung!

Ich habe noch eine kleine Bitte an diejenigen Mitglieder, die den **CLUB 80** zum Jahreswechsel, aus welchen Gründen auch immer, verlassen wollen. Bitte verabschiedet euch nicht sang- und klanglos durch Nichtbezahlen des Jahresbeitrages, sondern schickt mir eine Abmeldung. Eine Postkarte mit ein paar Zeilen reicht schon und erspart mir sehr viel Arbeit. Ich bedanke mich im voraus und wünsche denen, die uns wegen eines Systemwechsels verlassen viel Spaß und Erfolg mit der neuen Maschine!

Regionaltreffen
Wie ihr an anderer Stelle in diesem Info lesen könnt, fand am 21./22.11 ein Regionaltreffen "Nord" statt. Bezüglich dieses Treffens sprachen mich einige Mitglieder mit dem Wunsch an, auch für ihren Bereich Regionaltreffs zu veranstalten. Dazu möchte ich bemerken, daß nicht ich Veranstalter des "Nordlichtertreffens" war und ich mich auch nicht dazu bereitfinden werde, für andere Bereiche Regionaltreffen zu organisieren.

Meiner Meinung nach ist die Organisation und Durchführung Sache der Leute, die sich eine Zusammenkunft in ihrem Bereich wünschen. Dabei sollten sie wie folgt vorgehen:
1. eine kleine (3-4) Gruppe Interessierter finden und einen Termin absprechen (telefonisch);
2. den gefundenen Termin mittels eines Rundschreibens incl. Anmeldeformular etc. an die Mitglieder des CLUB 80, die in angemessener Entfernung zum Treffpunkt wohnen, verschicken (Portokosten können vom Club getragen werden!);
3. wenn zeitlich machbar, Information aller Mitglieder durchBekanntgabe des Termins im Info;
4. last but not least Information des Vorstandes!

Ein ähnliches Schema hat sich bei dem durch Gerald Schröder organisierten Treffen in der Nähe von Hamburg bewährt.

Zu diesem Thema möchte ich abschließend noch bemerken, daß ich sehr gerne bei den Vorbereitungen zu Regionaltreffen behilflich bin, mir aber nicht die Arbeit aufhalsen will und kann (schließlich gehe ich noch einer kleinen Nebenbeschäftigung nach), mehrere Regionaltreffen pro Jahr zu veranstalten!

Abwesenheit
Aus beruflichen Gründen werde ich mich (wieder einmal) für längere Zeit auf Reisen befinden. So werde ich mich für die Monate Jan., Feb. und März in Rentsburg (Schleswig-Holstein) aufhalten. Dabei wird es mir sicher nicht möglich sein, jedes Wochenende "in die Heimat" zu fahren. Unter diesem Umstand wird natürlich auch die Erledigung meiner Korrespondenz leiden; konkret: die Beantwortungszeit für Briefe wird wohl auf zwei bis drei Wochen steigen!

Ich möchte euch bitten, mir diese Verzögerungen schon im voraus nachzusehen und dringende Anfragen an die entsprechenden Vorstandsmitglieder (Gerald Schröder, Eckehard Kuhn usw.) zu richten. Die meisten administrativen Angelegenheiten (u.a. Verwaltung der Kasse, Versenden von Probeinfos usw.) wird meine bessere Hälte erledigen, so daß der Club in keiner Beziehung führungslos ist!

Bücherei
Wie schon im letzten Info ausführlich verkündet, erweitert der Club demnächst seine Bücherei. Auf dem Nordlichtertreff haben Gerald und ich folgende Bücher ausgewählt, die angeschafft werden:
- Turbo Pascal Wegweiser für Microcomputer (Grund- und Aufbaukurs);
- Pascal und Pascal-Systeme
- Programmierung in Modula 2
- CP/M - Anatomie eines Betriebssystems
- Das Z80 - Buch

Damit ist die Anschaffungswelle aber bei weitem noch nicht vorbei; im Frühjahr 1988 sollen weitere Bücher angeschafft werden. Wer noch Vorschläge/Buchbesprechungen hat, möge diese bitte an Gerald richten. Buchbesprechungen sollten zusätzlich direkt Jens Neueder zur Veröffentlichung zugesandt werden!

Sonderhefte
Eine weniger gute Nachricht muß ich euch leider noch eröffnen. Wie inzwischen sicher jeder mitbekommen hat, ist Jens Neueder durch den Bau seines eigenen Hauses bis über beide Ohren mit Arbeit eingedeckt. Dadurch wird sich der Neudruck vergriffener und die Erstellung neuer Sonderhefte bis in die Mitte des Jahres 1988 verzögern. Sollte sich jemand finden, der Jens diese Arbeit abnehmen kann, soll er sich bitte bei ihm melden.

Damit bin ich mal wieder am Ende angelangt. Ich wünsche euch viel Spaß bei der Lektüre des Infos, euer

Hartmut Obermann

Geburtstagskinder

Hallo, ihr lieben Clubfreunde,

heute wende ich mich einmal direkt an euch (normalerweise ist ja mein Mann mein Sprachrohr zu euch).
Ich möchte eine Idee des Bremerhavener Computerclubs aufgreifen und den Mitgliedern, die in den Monaten November, Dezember und Januar Geburtstag haben oder hatten, recht herzlich gratulieren.
Im voraus möchte ich noch bemerken, daß ich nicht von allen Mitgliedern das Geburtsdatum vorliegen habe. All diejenigen, die in der genannten Zeitspanne Geburtstag haben und nicht genannt sind, bitte ich, mir das Datum mitzuteilen. DANKE.

Hier also nun unsere "Geburtstagskinder":

Im Monat November hatten Geburtstag:

Krispin Michael am 2.
Hentz Werner am 3.
Mühlenbein Klaus-Jürgen am 7.
Schäfer Walter am 8.
Neueder Jens am 28.

Im Monat Dezember hat Geburtstag:

Böcker Dieter am 4.

Im Monat Januar haben Geburtstag:

Seelmann-Eggebert Jörg am 9.
Betz Heinrich am 17.
Drowäler Bernd am 29.

Noch einmal herzlichen Glückwunsch an alle genannten und ungenannten vom Vorstand und von mir.

Jutta Obermann

Moin, Moin!

Während die anderen noch beim Kaffeetrinken sind, dürfen wir (Hartmut und Gerald) einen kurzen Bericht über das Nordländertreffen verfassen.

Eins läßt sich jetzt schon sagen: die Sache war ein voller Erfolg! Schon am Freitag, den 20.11.87, war am Abend viel los, nämlich ein großes Trinkgelage der standfesten "Ältesten" des Clubs, Helmut Bernhardt, Arnulf Sopp, 1. und 2. Vorsitzender. Erst gegen 2.00 Uhr am Sonnabend war das Bett in Sichtweite.

Ein paar Stunden später fing der aktive Teil des Treffens sehr erfolgreich an, indem unser Assembler-Papst Arnulf sein IIIs durch einen 220 V-Reset ins Nirwana schickte. Dadurch war sein Geist aller Sorgen entledigt und er konnte uns so seine Geistesblitze unbehindert zuwerfen (noch ist nicht klar, ob er damit auch jemanden getroffen hat).

Nachdem die Stromversorgung wiederhergestellt war, konnte die Arbeit mit den Rechnern (zwei Genie I, ein Genie IIs, drei Model IVp) endlich beginnen. Es wurde kopiert, gehackt und eine Menge Informationen ausgetauscht. Notizblöcke wurden mit Ideen gefüllt, die später aufgegriffen und im Info veröffentlicht werden sollen. Als Beispiel wäre hier der von Hartmut angeleierte Vergleich zwischen drei Pascal-Dialekten, Z80-Assembler und RPNL zu nennen.

Eine der Hauptinteressen aller Anwesenden (ausgenommen Arnulf) war die Implementation und Anwendung von CP/M. Ein Problem bei der Anwendung war die Bedienung der Hilfsprogramme (Tastendefinition etc.). Wer zu diesen Thema und anderen Benutzungsproblemen etwas zu sagen hat: bitte mit kleineren Kapiteln im Clubinfo an die öffentlichkeit treten!

Wie es sich für ein Treffen gehört, kam auch die unvermeidliche Hardware auf ihre Kosten. Der Versuch, den Drucker vom Hans-Joachim Eilers an den Rechner von Harald Mand anzuschließen, endete mit einem generellen 'Lichtaus' für den Drucker. Eine Sicherung hatte sich zur Befehlsverweigerung entschlossen und war auf Nimmerwiedersehen durchgebrannt. Unser Beileid an den stolzen Besitzer!!!

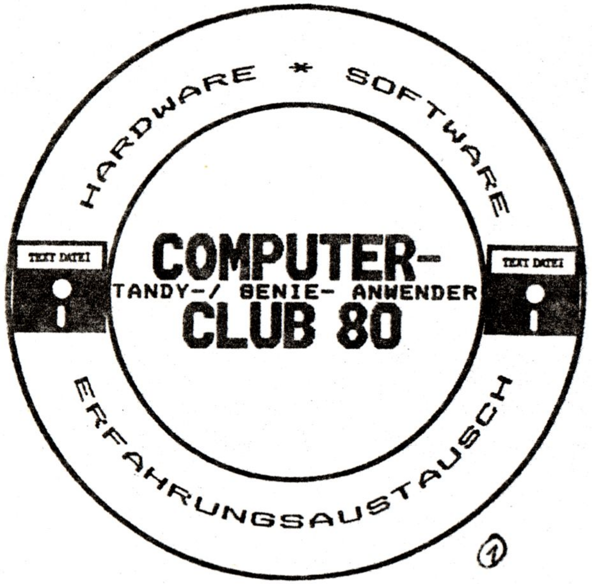
Neben den im Text schon genannten Software-Desperados und Hardware-Rambos nahmen am Treffen noch teil:

Kurt Müller mit Frau .....
Michael Krispin auch mit Frau .....
Matthias Homann (Gast mit Tendenz zum Clubeintritt)

In der Hoffnung, daß auch das nächste Regionaltreffen, egal ob Nord, Süd oder Mitte, ein solcher Erfolg wird, verbleiben wir (die Verfasser)

Hartmut Kurt und Gerald

Das Clubemblem - unser Aushängeschild



①

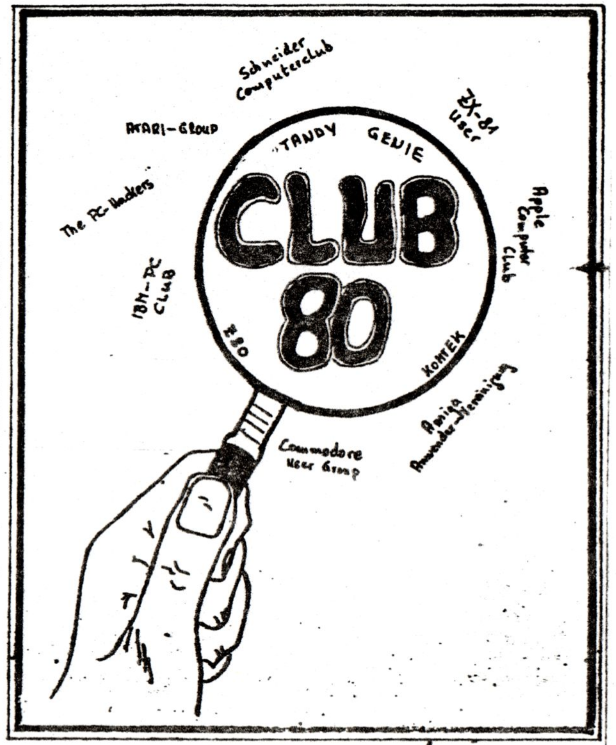
Angeregt durch Gerald Schröders Bericht im letzten Info über die bis jetzt noch magere Ausbeute bei den Entwürfen für ein Club-Emblem habe ich den Computer, Drucker, Zirkel, Schere und Leim zur Hand genommen und mit eigenen Versuchen begonnen. Im Laufe von zwei Stunden war der Papierkorb mit vielen Schnipseln gefüllt, schließlich lag aber doch eine brauchbare Vorlage auf dem Tisch.

Man kann die Qualität des Entwurfes mit etwas mehr Aufwand noch verbessern:

Mit einer schwungvollen, im Lichtsatz hergestellten Schrift, statt des Nadeldrucks zwischen den beiden Kreisbögen, würde das Emblem lockerer und lebhafter aussehen. Aber mit der jetzt laufenden Aktion soll ja erst einmal Material gesammelt werden für ein "Aushängeschild" unseres Clubs. Unsere Vorstandschaft kann dann entscheiden, ob einer der Entwürfe noch verbessert werden soll, oder - was auch denkbar wäre - daß aus Teilen mehrerer Entwürfe die endgültige Form unseres Emblems entsteht.

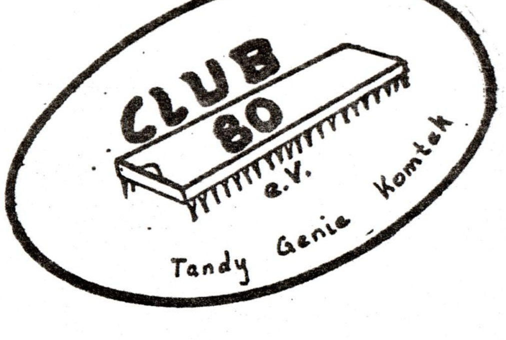
②

Heinrich Betz



①

Klaus Hermann ②



①

Arnulf Sopp

②

Vorstellung - die zweite!

Es ist schon einige Zeit her, daß eine kurze Vorstellung meiner Person im Info erschienen ist. Inzwischen hat sich einiges geändert, und das nicht nur bei mir. Dadurch, daß der CLUB 80 sich ständig über neue Mitglieder freuen kann, werden sehr oft gleichlautende Fragen zu meiner Person, meinen Interessen und vor allem zu meiner Gerätekonfiguration gestellt. All Dies möchte ich hier nochmals für alle Mitglieder offenlegen!

Zur Person:
Ich bin Jahrgang 1958 (gehe also stark auf die 30 zu) und am 24. August geboren. Nach der Hauptschule und anschließendem Abschluß der Mittleren Reife im Fachbereich Elektrotechnik erlernte ich den Beruf des Informationselektronikers. Kurz nach Abschluß der Lehre bemächtigte sich zuerst die Bundeswehr und dann meine heutige Frau meiner. Von beidem konnte und wollte ich mich bis heute nicht mehr trennen und so gebe ich nun in Fragebogen an, Beruf: Soldat, Familienstand: verheiratet, Kinder: eine Tochter (8 Jahre alt)!

Zu den Interessen:
Natürlich liegen meine Interessen, neben den Hauptinteressen Familie und Beruf, im Bereich der Computerei. Auf dieses Hobby kam ich durch die Notwendigkeit, mich während der Zeit, in der ich nur "Soldat auf Zeit" war, auf einen späteren Zivilberuf vorbereiten zu müssen. Da ich die größten Chancen im Bereich der EDV sah und sehe, habe ich versucht, mich intensiv in dieses Gebiet einzuarbeiten. Dabei erfasste mich das Computerfieber, das mich auch bis heute nicht mehr losgelassen hat!

Zur Zeit liegen meine Computerinteressen vor allem im Softwarebereich wobei allerdings auch die Hardwareseite nie ganz in den Hintergrund tritt (siehe "Maus am TRS 80"). Wenn mir die Arbeit als Clubvorstand Zeit dazu läßt, beschäftige ich mich unter anderem mit Turbo-Pascal (Grafiktreiber, Umschreiben von PC-Programmen), Assemblerprogrammierung (Patchs in Betriebssystemen) und dem Ausprobieren von Standartanwenderprogrammen (WordStar, dBASE, Multiplan usw.) und Sprachen. Eventuell werde ich mich demnächst auch mit MODULA beschäftigen. Ein Hobby im Hobby ist das Knacken von CP/M-Fremdformaten, was mir immer sehr viel Spaß macht.

Zum System:
Angefangen habe ich einmal mit einem 16k Model 1, welches ich langsam auf 48k und drei Laufwerke ausgebaut habe. Dabei haben vor allem die Diskdrives ständig gewechselt, so daß ich glaube, mit Anschluß und Konfigurierung dieser Massenspeicher einige Erfahrung zu haben.

Vor einiger Zeit bin ich dann auf ein Model 4p umgestiegen, welches natürlich sofort zwei 80/DS/DD-Laufwerke und einen Anschluß für externe Drives (da hängt eine 40/DS/DD-Floppy dran) erhielt. Kurz darauf folgten die 128k-Erweiterung und die Installation der micro-Lab-HRG (640*240 Punkte). Als weitere Zusatzgeräte gibt es eine Maus und einen Epson RX80F/T-Drucker. Demnächst soll der Speicher auf 320k aufgebohrt werden, und in der Schublade schlummert auch noch der Plan für einen ECB-Busanschluß (der ruht dort aber schon fast ein Jahr lang!). Auch der Anschluß eines 8"-Laufwerks ist schon lange geplant.

Das 4p ist zwar, Zitat:"Der schönste Portable der Welt!", tragbar ist er mit ca. 14kg Gewicht aber nur bedingt. Aus diesem Grund habe ich mir zusätzlich noch ein Model 100 zugelegt. Das Ding ist nicht größer als ein DIN A4-Blatt, 5 cm hoch, ca. 1.5 kg schwer und damit tatsächlich tragbar! Den 100'ter benutze ich hauptsächlich als Speicherschreibmaschine für unterwegs, da sich Dank der RS 232-Schnittstelle die übertragung der Texte auf das 4p sehr einfach gestaltet.

Damit wäre wohl alles gesagt, was von Interesse sein könnte und mehr wird außerdem nicht verraten!

Hartmut Obermann

Ich würde mich freuen, wenn noch mehr Mitglieder, die sich bisher noch nicht im Info vorgestellt haben, dies nachholen würden. Ich glaube, daß es sicher für alle Leser interessant ist, mehr über die anderen Clubmitglieder zu erfahren und so bestimmt noch mehr Kontakte zwischen den Mitgliedern entstehen!

Hallo Leute,

auch wenn es schwerfällt, aber Ihr habt wieder ein neues Clubmitglied zu beklagen. Ich habe bis jetzt zwei Jahre mein Unwesen im Bremerhavener und im Münchner Club getrieben. Der dritte Beitrag für den CLUB 80 war mir immer etwas zuviel des Guten, aber die sinkende Aktivität läßt einem fast keine andere Wahl.

Zur Person: Ich bin 22 Jahre alt und wurde durch eine Reihe grausamer Schicksalsschläge nach München verschlagen (ich bin ein Hamburger ...nein, keiner von Mc Donalds!). Momentan bin ich mitten im Studium (E-Technik) und hoffe einigermaßen heil durchzukommen. Zu meinen Hobbies zählt, wie man sieht, auch die Computerei. Mein Fuhrpark besteht aus einem Genie IIs das mit 8MHz läuft und mit einer HiRes (480x192), einer 80-Zeichen-Karte und 512k-RAM (mit Helmut Bernhardts Banker) ausgerüstet ist und einem Epson RX-80 F/T. Am liebsten beschäftige ich mich mit den kleinen Hintertürchen und Stolpersteinen im DOS und wie man die Library sinnvoll erweitert (hat jemand mehr als 65 Befehle untergebracht ?)

Auf gute Zusammenarbeit

Alexander Schmid

Guten Tag, Freunde!

Der guten Sitte des Club 80 folgend, sich als neues Mitglied kurz vorzustellen, möchte ich einige Zeilen zum Besten geben.

Einige von Euch mögen hier und dort schon den einen oder anderen Artikel von mir gelesen haben, sodaß diese sich schon ein Bild davon machen können, was sich der Club 80 da mit dem Herrn Heidenreich angelacht hat. Mein "EDV-Lebenslauf" dürfte diese Bild weiter abrunden:

So Ende 1979 wurde ich erstmals mit der Datenverarbeitung konfrontiert, als im Rahmen eines E-Technik-Studiums auch zwei Semester Datenverarbeitung nebst Praktikum angesagt waren. Nach anfänglich wenig Begeisterung ob dieser zusätzlichen Belastung neben "wichtigeren" Studienfächern schlug dieselbe bald in Richtung EDV um. Mit dem Erfolg, daß die ehemals als wichtiger angesehenen anderen Studienfächer bald zur Nebensache wurden und Ihr jetzt keinen Dipl. Ing. FH im Club begrüßen dürft. Als zudem noch mich störte, daß man den Rechner eigentlich nur via Lochkarten bedienen durfte, den Rechenzentrums-Assistenten dagegen, daß ich den Rechner nicht nur via Lochkarten bediente, folgte einem Hausverbot im Rechenzentrum die Anschaffung eines eigenen Computers.

Hätte ich damals das preiswerteste Modell erworben - manch einer mag sich noch an das "Superboard" erinnern - wäre ich wohl der Prozessorfamilie weiter gefolgt und hätte heute einen Commodore auf dem Schreibtisch stehen. Schon damals der Hardwareseite weniger zugeneigt entschied ich mich, - zum Glück? - gegen den Superboard-Bausatz für einen kompletten EG 3003 (später hieß er auch Genie I) und gesellte mich damit zu den 80er Freunden.

Diesem EG 3003 mit sagenhaften 16 KByte Hauptspeicher, keiner Groß/Kleinschrift und ohne deutsche Umlaute folgte, sobald es meine Brieftasche zuließ, das Genie III günstig zum Händler-Einkaufspreis für nur (!) 5182,- DM. Irgendwann dazwischen lernte ich auch Arnulf kennen, über den ich letztendlich auf den Club 80 aufmerksam wurde.

Nachdem mich neulich Vater Staat für 15 Monate aus dem bürgerlichen Leben entführte, opferte ich gesparten Wehrsold, Entlassungsgeld und ein wenig vom Notgroschen für einen AT-Kompatiblen. Aber keine Angst, Freunde: dem 80er werde ich schon allein aus dem Grunde treu bleiben, weil man mit NEWDOS/80 ein Entwicklungssystem und mit MS-DOS eine Anwenderprogrammabfahrkrücke unter den Fingern hat!

Für den Club 80 habe ich mich letztendlich auch deshalb entschlossen, weil in unserem 80er Hobby noch soviel Interessantes steckt, daß es einfach zu schade wäre, seine Werke mangels Interesse "echter" Fachzeitschriften in den Mülleimer wandern zu lassen.

Ach, so! Wen's neben der technischen Hobbyseite interessiert: Ich bin Jahrgang '58, 183 cm groß, rund 100 kg schwer und derzeit arbeitslos.

Euer Uli

„Frau Müller, Sie haben ein äußerst ungewöhnliches Baby!!"

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....28. Februar 1988
Jahresbeitragstermin ....15. Februar 1988 (spätestens !!)

-- Messen '88 --

Hobby-tronic & COMPUTERSCHAU ....Dortmund 3. - 7. Februar 1988
IMPRINTA ....Düsseldorf 18. - 24. Februar 1988

* * Der schnellste Zahnarzt * *

(Unmed.-techn.Assistent: KaJot Mb)

Seit es bei den Zahnärzten die amerikanischen Drehstühle gibt, auf denen der erwartungsvolle Patient mit 26400 UpM (= Kammerton!) gedreht wird, während der Bohrer stillsteht, macht das Wurzelziehen richtigen Spaß! Denn das mühsam unterdrückte "Aaaahhh..." des geehrten Gedrehten hat jetzt wenigstens die offizielle, genormte Tonhöhe, die die Anwesenden - Arzt, Assistentin und ihr Opfer - im musikalischen Genusse vereint und somit harmonische Stimmung aufkommen läßt. Kein Unwohlsein kann diese mehr beeinträchtigen. Und es geht flott von der Hand...

Warum erst in diesem Jahrhundert?

Schon vor 2250 Jahren lehrte uns *ARCHIMEDES*, wie man Wurzeln angenehm schnell und schmerzlos zieht. Und das mit beliebig vielen Stellen!

Als ich im letzten INFO u.a. ein Programm zum Wurzelziehen mitteilte, habe ich euch diese Tatsache unterschlagen, (um eure Geduld zu prüfen). Diejenigen, die es ausprobierten, haben das Programm (oder sich selbst) in der Luft zerrissen. Mit Recht!

Ein Fünfzeiler tut's!

In Sekundenschnelle, je nach Güte des Schätzwertes, den jeder sofort ansetzen kann, der die Quadrate der Zahlen von 1 bis 10 kennt; also die Zahlen: 1, 4, 9, 16, 25, 36, 49, 64, 81, 100. Als Schätzwert nimmt man diejenige Zahl von 1 bis 10, deren Quadrat den ersten beiden Stellen des Radikanden (das ist der "Patient") am nächsten kommt, wenn er eine gerade Anzahl Stellen hat bzw. der ersten Ziffer am nächsten kommt, wenn seine Stellenzahl ungerade ist; an diese vorläufige Zahl werden noch halb soviel Nullen angehängt, wie der Patient Stellen außerdem noch hat. Beispiel:

Um die Wurzel aus 4711 zu ziehen, beginne ich mit dem groben Schätzwert 70, denn 7 zum Quadrat (=49) kommt den beiden ersten Ziffern (47) am nächsten; dazu kommt eine Null, denn außer 47 hat dieser "Patient noch zwei Ziffern. Oder:

Um die Wurzel aus 8123456 zu ziehen, beginne ich mit dem Provisorium 3000: Die Stellenzahl ist ungerade; ich schaue mir also nur die erste Ziffer (8) an; das Quadrat von 3 kommt ihr am nächsten; dahinter gehören noch 3 Nullen, von wegen der 6 Ziffern nach der 8. - Capez? - Capé!

Mehr bedarf's nicht. Das Programm errechnet hieraus in Sekundenschnelle die - na was schon? Und das mit einer Genauigkeit, die kein Taschenrechner bietet - nämlich mit soviel Stellen hinter dem Komma, wie übrig bleiben, wenn man die Anzahl Stellen vor dem Komma von 16 abzieht. Mehr kann die "Variable mit doppelter Genauigkeit" bekanntlich nicht leisten.

Wer gibt einen Trick an, die mit 16 Stellen begrenzte "Genauigkeit" unseres 8-Bitters zu überbieten? Ich meine, nicht nur beim Zahnarzt, sondern bei jeder beliebigen Berechnung? (Man führe die Rechnung mit String-Transformationen durch!)

Hier nun *ARCHIMEDES'* Meisterleistung:

Wurzelberechnung nach ARCHIMEDES

```
1  ' *** Wurzelberechnung nach ARCHIMEDES ***
2  ' *** (C) K.-J.Mühlenbein, Weinheim 1987 ***
3  '
10 CLS
   DEFINT L
   DEFDBL R,S,W
   PRINT "                 Wurzelberechnung
20 PRINT
   INPUT "Radikand ";R
30 INPUT "Schätzwert für die Wurzel ";S
   L = LEN ( STR$ (S)) - 1
40 W = R / S
   IF ABS (S - W) < 1 / 10^(15 - L)
   THEN PRINT "Wurzel aus"R"="W
       END
50 S = (S + W) / 2      <- Archi's Trickzange!
   GOTO 40
```

* * DA CAPO * * DA CAPO * * DA CAPO * *

(oder: Wie man Töne erzeugt)

Haste Töne!

Nein? - Vielleicht möchtest du welche von dir geben?
Aber wie??
Ich meine natürlich: Wie mit dem Computer?
Nun, ein paar physikalische Grundlagen der Tonerzeugung mögen dazu verhelfen; die übertragung auf den Computer ist dann - wie ich inzwischen erlebte - vom jeweiligen Computer abhängig; selbst bei gleichem Prozessor. Also meist keine "kompatible Angelegenheit". Aber wenn du deinen Computer kennst - das sei vorausgesetzt - wird es dir leicht fallen, das hier Beschriebene auf dein Gerät zu übertragen. (Hierzu gehört nicht einmal Musikalität.) Und zwar natürlich in Maschinensprache! Denn wenn sich erst noch der Interpreter zwischen Tonanforderung und -ausführung einschalten muß, hörst du vor lauter "Denkpausen" - na, was wohl? Eben *Pausen* und kaum Töne...
Machen wir's so kurz wie bündig. Denn es soll keine Oper mit drei Akten werden. Ja nicht einmal eine Ouverture.

Schlimmstenfalls eine Tonleiter.
Und die bringt dich dann schon heiter
um mehr als einen Ton weiter...

Jeder reine Ton ist im wesentlichen eine Folge von Sinuswellen. Die zugehörigen "Obertöne" wollen wir verschweigen. Und was eine Sinuswelle ist, weiß jeder Techniker. Eben eine schöne Welle. Die hat einen Berg und ein Tal. Beide haben eine bestimmte Höhe bzw. Tiefe, die wir als absolut gleich groß annehmen wollen. Glaube nur nicht, dies sei die "Tonhöhe". Diese Höhe (Tiefe) bestimmt vielmehr die Intensität, mit der die Welle dich umschmeißt, wenn du darin badest. Man nennt sie die "Amplitude": (Physikalisch ist die Schallintensität zum Amplitudenquadrat proportional.)
Die *akustische* Tonhöhe jedoch - die nicht unsere Muskeln, sondern unsere Ohren etwas angeht - wird durch die *Häufigkeit* bestimmt, mit der die einzelnen Wellen dich anstoßen. Sie ist um so höher, je schneller die einzelnen Wellenstöße aufeinander folgen. Brummst Du noch wohlig "Ahhh", wenn dich alle zwei Sekunden ein Wellengang hin- und herschaukelt, so kreischst du bald in hellen (hohen) Tönen, wenn dich ein wilder Wellengang mit kurzen schnellen Stößen durcheinander schüttelt! Oder nicht?
Lautstärke ist also eine Sache der Amplitude -
Tonhöhe eine Sache des "Wellenganges", der langsam oder schnell aufeinander folgt; sagen wir: der "Schrittlänge" der Wellen, kurz: Wellenlänge. In der Akustik bevorzugt man zur Kennzeichnung der Tonhöhe das Gegenstück der Wellenlänge: die schon erwähnte Häufigkeit, mit der die Wellen in jeder Sekunde eintreffen; oder vornehmer: die *Frequenz* (proportional zur reziproken Wellenlänge), gemessen in Hertz (Hz) = Schwingungen pro Sekunde.
Frequenz mal Wellenlänge ergibt die Wellengeschwindigkeit. Im hier behandelten Fall: die Schallgeschwindigkeit (i.d.R. in Luft so um die 330 m/s).
Trifft die so erschütterte Luft auf deines Ohres Trommelfell, so *hörst* du was (hoffentlich). Und umgekehrt: Schlägst du auf ein Trommelfell (bitte, nicht des Ohres, nimm eine Pauke), so wird die Luft von ihm, nein: von den von ihm ausgehenden "Schwingungen" (Stoßwellen in der Luft) erschüttert. Wem sag' ich das?

Und genau so ein tonerzeugendes "Trommelfell" ist die Membran des Lautsprechers in oder an deinem Computer (so du einen Verstärker hast), den du, wie du weißt, durch "Stromstöße" - oder vielmehr Spannungs*schwankungen* in Schwingungen versetzen kannst.
So, damit haben wir schon alles, was an Voraussetzungen gebraucht wird: Wir wissen nun, daß die Membran des Lautsprechers in mechanische Schwingungen versetzt werden muß; dies vollbringt der Verstärker, in welchem elektrische Spannungen erzeugt werden müssen, die mit der gewünschten Frequenz zwischen Ausschlägen nach der einen Seite ("Wellenbergen") und solchen nach der anderen Seite ("Wellentälern") gleichmäßig abwechseln!
Die Aufgabe des Computers ist es nun, diese elektrischen Signale im "richtigen", d.h. melodiegerechten Rhythmus zu erzeugen. Da die Taktfrequenz der CPU aber festliegt und 200 bis 300 mal höher ist als alle hörbaren Töne, muß die CPU stets während der Dauer einer Halbwelle vor sich "hintrödeln", bevor der nächste Anstoß nach der anderen Seite (Halbwelle) erfolgt. Dies macht die Programmierung zugegebenermaßen etwas komplex und ein Musikprogramm entsprechend unübersichtlich (finde ich).
Wir können auch die Stärke der Ausschläge (ihre Amplitude) beeinflussen, um dadurch verschiedene Lautstärken hervorzurufen; also das, was man in der Musik "Dynamik" nennt. Aber das soll uns jetzt nicht beschäftigen, denn mit unseren "billigen" Computern haben wir hierfür nur sehr begrenzte Möglichkeiten. Daher klingt einfache Computer-"Musik" auch meist so "eintönig" (=monoton? - Ein untreffendes Wort! Denn es werden ja <u>viele</u> Töne erzeugt, nur jedoch mit gleicher Lautstärke; also eher: "mono-phon".)
Außer der *Schwingungsdauer*, die der reziproken Frequenz und damit Tonhöhe entspricht, müssen wir bei einem Lied natürlich auch noch die Dauer der einzelnen Töne selbst, eben die *Tondauer* berücksichtigen. Das bedeutet, daß der Wechsel zwischen höherer und niederer Spannung im Verstärker nicht nur mit einer der gewünschten Frequenz (= *Tonhöhe*) entsprechenden Geschwindigkeit erfolgen muß, sondern auch eine gewünschte Zeit lang *andauern* muß (= *Tondauer*). Bei der Realisierung im Programmablauf bedeutet das also, wie jedermann sofort erkennt, die Ineinanderschachtelung zweier Programmschleifen.
Damit ist nun, meine ich, wirklich alles gesagt, was zu den Grundlagen Computer-unterstützter Tonerzeugung (CAT = Computer Aided Tuning <Copyright 1987 KaJot, zur Weitergabe wie immer freigegeben>) zu sagen ist.
Bleibt nur noch die Frage: Wie kommt der Wille zum Ton von der programmgesteuerten CPU bzw. vom Speicher zum spannungsgesteuerten Verstärker?!
Dies ist offenbar von Computer zu ebendiesem verschieden. Beim *GENIE IIIs* erzeugt die Speicherstelle 3860Hex einen "Knacks", wenn sie angesprochen wird (weiß ich von Arnulf); beim *TRS-80* wird die Spannung über den Kassettenport 255dez = FFhex ausgeliefert, und zwar mit höherem Potential ("positiver Wellenberg"), wenn Bit 0, und mit niedrigerem Potential ("negativer Wellenberg" = "Wellental"), wenn Bit 1 gesetzt ist; dazwischen liegt in der Mitte die Ruhelage, um die die Welle schwingt, mit rückgesetzten Bits 0 und 1:

a) Wellenberg { ==> Bit 0 gesetzt / ==> Bit 1 nicht gesetzt } = 01 *binär* = 1dez

b) Ruhe ==> Bit 0 und 1 rückges. = 00 binär = 0dez

c) Wellental { ==> Bit 0 nicht gesetzt / ==> Bit 1 gesetzt } = 10 binär = 2dez

HEFT 22
Dezember 1987

Lassen wir die Spannung zwischen a) und b) schwanken, so erhalten wir also einen Ton. Das wollen wir gleich mal "in BASIC" ausprobieren:

```
FOR I = 0 TO 100 : OUT 255, 1 : OUT 255, 0 : NEXT
```

Wir erhalten einen Brummton, dessen Tiefe durch die CPU-Frequenz und die Interpretationsgeschwindigkeit bedingt ist.
Wenn wir statt 1 und 0 im Wechsel 2 und 0 auf den Port 255 ausgeben (also Zeilen b und c abwechseln), erhalten wir das Gleiche (gleicher Spannungsunterschied).
Lassen wir 1 und 2 miteinander wechseln (Zeilen a und c), erhalten wir (etwa) den gleichen Ton, aber viermal so stark (doppelter Spannungsunterschied = doppelte Amplitude = vierfache Intensität; die Hör-Intensität wird allerdings wegen des *WEBER-FECHNER*-schen Gesetzes nicht viermal so laut empfunden.) Mehr nicht zum Thema "Lautstärke" (s.o.)
Die *Tonhöhe* jedoch erzeugen wir *in Maschinensprache*, indem wir den ungleich schnelleren CPU-Takt *ohne dazwischen geschalteten Interpreter* so oft walten lassen, wie die Frequenz des jeweils gewünschten Tons es vorschreibt. Wie bereits gesagt.

Um dies alles etwas zu erfassen, schauen wir uns einmal das Listing des "CLUB80-Liedes" in diesem Heft an! Es handelt sich um eine revidierte Fassung, nachdem mich Arnulf auf einige Umständlichkeiten in meiner ersten Fassung hingewiesen und mir Tips gegeben hat, wie man den Code eleganter gestalten kann. (Die Tonwiedergabe selbst ist dadurch allerdings nicht schöner geworden - "Mann bleibt Mann" und Computer dto.)
Zeilen 52 bis 100 enthalten die Melodie "Wohlauf, Kameraden,...". Hierzu später.
In Zeile 10 wird die Adresse des Anfangs der Melodie ins Doppelregister HL geladen.
Zeilen 11 bis 15 sorgen dafür, daß die Töne nicht ineinander verschwimmen, sondern schön rhythmisch aufeinander folgen.
In Zeile 16 wird der Akku mit der Adresse des "LSB" (niederer Teil des 2-Byte-Wertes der Tonhöhe) geladen, der dann in Zeile 21 zum Register C weitergeschickt wird, falls das Wiederholungszeichen "DACAPO" noch nicht erreicht wurde. Wurde dieses erreicht (Zeile 76), wird das DACAPO-Flag zuerst unkenntlich gemacht (sonst gäbe es eine "Schallplatte mit Sprung") und dann dieser erste Teil der Melodie wiederholt. (Anmerkung: Nur die mit dem Asteriken * gekennzeichneten Zeilen stammen von mir; alles übrige von Arnulf.)
Zeilen 22-23 schicken das MSB, das höherwertige Byte der Tonhöhe, zum Register B, so daß jetzt das Doppelregister BC die die Frequenz bestimmende "Wellenlänge" 150 des ersten Tones enthält.
Zeilen 24-27 machen das Gleiche mit der Tondauer (hier: die "90"), die in den nächsten beiden inzwischen erreichten Adressen (hier 5249h und 524Ah) steht.
Nachdem der "Notenzeiger" HL in Zeile 28 auf die Adresse der nächsten Note gesprungen ist, wird in Zeile 29 die erste Schleife zur Kontrolle der Tondauer betreten. Solange sie läuft (Kontrolle in Zeilen 30-32), wird nun ab Zeile 33 bis Zeile 39 der Spannungspegel am Port 255 abwechselnd hoch und niedrig gesetzt und dazwischen das Unterprogramm "Knack" (Zeilen 41 bis 46) gefahren, welches die Membran schubst bzw. zerrt (push/pull). Nach jeder Vollschwingung = 2 Halbschwingungen = 1 Periode ...(Forts. nächste Seite) →

> Anm.: Der Techniker weiß, daß so keine reinen Sinuswellen erzeugt werden können; auf das hierfür notwendige Superpositionsprinzip mehrerer - theoretisch unendlich vieler - Frequenzen soll hier nicht eingegangen werden. Es genügt der Hinweis, daß dieser Mangel für den armseligen Ton dieser Computer"musik" mitverantwortlich ist. Die von Arnulf stammende Label-Bezeichnung "Knack" ist deshalb durchaus treffend.

...fragt das Programm in Zeilen 43/44, ob noch Schwingungen gewünscht werden; denn pro Sekunde sind ja je nach Tonhöhe viele Schwingungen auszuführen. Ist dies der Fall, geht's zurück nach Zeile 42 zum nächsten Schwingungs"knack", andernfalls zurück zu Zeile 39, die sich zunächst die soeben durch die dauernde Dekrementierung abgebaute Tonhöhe aus dem Stack zurückholt, um sie wieder zur Verfügung zu haben, falls sich nach dem Rücksprung nach Label "loop2" (Zeile 40) ergibt, daß - entsprechend dem jeweiligen Stand des Tondauerzählers, welcher nunmehr in Zeilen 30 bis 32 geprüft wird - der Ton noch länger andauern soll. Ist der Ton noch nicht zünde - Verzeihung: zuende - geht das Spielchen mit Spannungszuteilung usw. ab Zeile 33 von vorne los. Ist aber DE=0000000000000000bin *), so beschäftigt sich unser fleißiger Tonerzeuger auf eben die gleiche Weise mit dem nächsten Ton, der ihm in Zeile 16 vorgelegt wird - nicht ohne vorher das eingangs bereits gelobte (unmerkliche!) "Päuschen" (Zeilen 11-15) eingehalten zu haben...

> *) Anm.: Ich weiß zwar, daß die Gleichung DE=0 genügt hätte; doch weiß ich auch um die didaktische Unvollkommenheit von Computer-Lehrbüchern; 16 Nullen prägen dem Anfängerhirn optisch ein, daß es sich um ein Doppelregister mit 16 Bits handelt. [2 * 8 Bit = 2 Byte)

War dieses bisher der "programmatische" Teil des Programms, sozusagen das Instrument oder Handwerkzeug, so folgt jetzt der nicht weniger wichtige, aber wesentlich flexiblere Teil, der das zu bearbeitende Material enthält: die Melodie; die Folge der Töne. Dieser Teil ist flexibel, weil hierfür jederzeit andere Melodien eingesetzt werden können.
Statt einer ausführlichen Ableitung der Formeln für Frequenz- und Tondauervorgabe, die zu weit führen würde (das INFO ist kein Mathebuch), sei es gestattet, nur diese Formeln selbst anzugeben. Anzumerken ist lediglich, daß die Basiszahl, auf der die Berechnungen für die Tonhöhen beruhen, vom CPU-Takt abhängig ist. Was ich auch erst bei der Übertragung vom GIII auf meinen TRS-80 herausfand.
Drum:
Probieren ergänzt das Studieren!

Kafot

ANHANG: Datenformeln.
Kammerton a' --> 440 Hz
übrige Frequenzen: $f = 440 * (\sqrt[12]{2})^n$ /Hz
mit n>0 Anzahl der Halbtonabstände vom Kammerton nach oben
bzw. n<0 " " " " " " unten.
Höhenzähler (=jeweils 1.Zahl in der Tönetabelle):

$$H = k/f$$

worin die "Zeitkonstante" k experimentell zu ermitteln ist. Sie ist vom CPU-Takt sowie der Programmstruktur abhängig und beträgt bei meinem Programm k=66000. [„Basiszahl" für TRS80/1]
Jede Tonhöhe hat ferner einen eigenen "Zähler" für die Notenlängen. Als Basiswert für die Viertelnote mit dem Kammerton a' (=440Hz) wurde L=90 gewählt. Es gilt:

$$L = c * f$$ (2. Zahl i.d. Tönetabelle)

Der Längenzähler ist also eine lineare Funktion der Frequenz.
Die Proportionalitätskonstante c hat in unserem Musikbeispiel den Wert $c = 0{,}20\overline{45}\ldots/\mathrm{Hz}^{-1}$
Die anderen Zähler für 1/16-, 1/8- und 3/16-Noten sind entsprechende Bruchteile von L.

E N D E

* * * Schwere Kavallerie * * *

(von Sopp, nicht Suppé)

Es wurde geritten! Der *DERBY-Sekt* ist vergeben! Mein Preisausschreiben aus INFO 20, Seite 9 ("Noch ein Leserbrief") ist gelaufen. Kaum war das INFO raus, kam das Roß aus Lübeck angesprengt – sein Reiter nahm die Trophäe unter hellem, klarem Trompetenklang entgegen: *Arnulf* war der Erste (und Letzte). Er dokumentierte mir seine Lösung nicht nur mit einem sauberen Listing, ndern demonstrierte es auch gleich ad auditum von Tonband-Kassette:

"Wohlauf, Kameraden, auf's Pferd, auf's Pferd..."

erklang's aus meiner Stereoanlage.

(Drum: "Auf's Wohl, auf's Wohl, AS-semblihierer!")[Gewieher..] So mußt ich's wohl glauben, daß es auf dem GIIIs gut gelungen war – denn von Diskette hätte ich sein Programm nicht abspielen können: Das GENIE IIIs erzeugt die "Musik" direkt aus der Speicherzelle 3860h mittels eines Knacklautes, während der TRS-80 sie über seinen Kassettenport 255 (OFFh) hergibt (siehe mein "3.Assemblerprogramm" im INFO 15, Seite 15).

Daß diese beiden als "Musikinstrumente" nicht kompatibel sind, merkte ich aber erst, als ich unbekümmert *Arnulfs* Listing eingegeben und (mit *ZEUS*) assembliert hatte und dann dem Verstärker meines TRS-80 die zackige Melodie abverlangte – es kam zu einem Hänger... Erneuter Briefwechsel (siehe "Das TSCRIPS-Drama"; aber dies- nur einmal); darauf erst erfolgte besagte Aufklärung (man muß die Esoteriker beim Gewissen nehmen).

Suppés – pardon: *Sopps* "Leichte" wurde bei der Anpassung an den TRS-80 unter meinen Händen zur "Schweren Kavallerie". Der gelungene Effekt (auch aus meinem Verstärker ertönt jetzt das "Reiterlied") beweist mir zwar, daß mein Tun nicht falsch war. Selbst *ZEUS* hat mir "00000 Fehler" bescheinigt. Doch glaube ich, daß es eine elegantere Lösung für meine Anpassung geben kann. Ebenso für meine *"da capo"*-Erfindung: [redacted] Arnulf hatte die Noten für die Wiederholung des ersten Teils explizit hingeschrieben. Musikanten setzen stattdessen nur die beiden Wiederholungssymbole (senkrechte Balken durch die Notenlinien). So wollte ich es auch haben. Ich baute also ein "Da capo" ein (16 Bytes) und sparte mir dadurch die nochmalige Eingabe der gleichen 48 Integers = 96 Bytes. Es klappt zwar; aber vielleicht ginge auch das eleganter zu lösen und der Originalcode um mehr als diese 80 Bytes zu kürzen. Die kleine Verlängerung der Pause durch diese Subroutine ist für scharfe Ohren unüberhörbar. (Das nachfolgende Listing gibt *Arnulfs* Lösung wieder. Die von mir eingebrachten Teile sind mit einem '*' versehen. Für Verbesserungsvorschläge bin ich stets dankbar.)

Unser ältestes Clubmitglied, *TACITUS, der Große Schweiger*, hatte die Preisaufgabe natürlich auch gelöst. Und die Lösung natürlich nicht eingesandt. Und deshalb natürlich auch keinen Sekt bekommen. * Wir Drei haben nun beschlossen, den Oberen des *CLUB 80* und seinen Mitgliedern einen naheliegenden Vorschlag zu unterbreiten. Dieser sei – Würde, wem sie gebührt – hier pathet-poet-isch vorgebracht:

Tacitus / Arnulf / KajoT – kurz: TAKT –
erklären als unumstößlichen Fakt:
Der oben beschriebene Reiter-Hit,
geschrieben und assembliert Bit für Bit,
sei künftig als zünftiges Hacker-Symbol
das Lied des CLUB 80 !

*** Wohlauf & Auf's Wohl! ***

KaJot


```
                           ;    ############################################
                           ;    #               Das CLUB 80-Lied           #
                           ;    #  für GIIIs in Toene gesetzt von Arnulf Sopp  #
                           ;    # *für TRS80 in Szene gesetzt von Ka-Jot-bein *#
                           ;    #           (Nachahmung empfohlen)         #
                           ;    #           = no  COPYRIGHT 1987 =         #
                           ;    ############################################
                           ;
5200                             ORG     5200h
5200  214752          start      LD      HL,tab    ;Tab.f.Tonhöhen & -längen
5203  010002          loop0      LD      BC,512    ;Zähler f.Päuschen zw.zwei Tönen
5206  0B              loop1      DEC     BC        ;Zähler erniedrigen
5207  78                         LD      A,B       ;prüfen, ob er
5208  B1                         OR      C         ;schon abgelaufen ist
5209  20FB                       JR      NZ,loop1  ;weiter warten auf Godot
520B  7E                         LD      A,(HL)    ;LSB der Tonhöhe laden
520C  FE32                       CP      50        ;DA CAPO-Flag erreicht? *
520E  2825                       JR      Z,DACAPO  ;falls ja, wiederhole! *
5210  B7                         OR      A         ;00 als Endekennung?
5211  C8                         RET     Z         ;wenn ja, zuruck ins DOS
5212  4F                         LD      C,A       ;sonst LSB nach C
5213  23                         INC     HL        ;nachstes Byte der Tabelle
5214  46                         LD      B,(HL)    ;als MSB der Tonhöhe nach B
5215  23                         INC     HL
5216  5E                         LD      E,(HL)  ;LSB der Tondauer
5217  23                         INC     HL
5218  56                         LD      D,(HL)  ;MSB  "       "
5219  23                         INC     HL        ;Tabellenstelle für nächsten Ton
521A  1B              loop2      DEC     DE        ;Tondauerzähler erniedrigen
521B  7A                         LD      A,D       ;prüfen, ob er
521C  B3                         OR      E         ;schon abgelaufen ist
521D  28E4                       JR      Z,loop0   ;falls ja, nächster Ton
521F  3E01                       LD      A,1       ;Spannungspegel für Port 255 *
5221  C5                         PUSH    BC        ;Tonhöhe retten              *
5222  CD2D52                     CALL    knack     ;zur Tonausgabe (1.Halbwelle)
5225  C1                         POP     BC        ;Tonhöhe zurückholen         *
5226  C5                         PUSH    BC        ;aber auch nochmal retten    *
5227  CD2D52                     CALL    knack     ;zur Tonausgabe (2.Halbwelle)
522A  C1                         POP     BC        ;Tonhöhe zurückholen         *
522B  18ED                       JR      loop2     ;weiter in der Tondauerschleife
522D  D3FF            knack      OUT     (OFFH),A; Halbwelle ausgeben
522F  0B              frequ      DEC     BC        ;Frequenzzähler erniedrigen
5230  78                         LD      A,B       ;checken, ob schon abgelaufen
5231  B1                         OR      C         ;dann ist A=00
5232  C8                         RET     Z         ;zurück, falls dem so ist
5233  18FA                       JR      frequ        ;sonst Frequenz weiter halten
5235  01E803          DACAPO     LD      BC,1000      ;Ausschaltung des         *
5238  DD21A352                   LD      IX,HALBZT  ; DA CAPO-Flags             *
523C  ED43A352                   LD      (HALBZT),BC; mittels eines            *
5240  DD360202                   LD      (IX+2),2    ; Index-Registers          *
5244  C30052                     JP      start       ; Wiederholung des 1.Teils *
5247  9600            tab        DW      150,90 ab hier: Töne: erst Höhe, dann Dauer
524B  7000                       DW      112,120
524F  7000                       DW      112,90
5253  7000                       DW      112,30
5257  7000                       DW      112,60
525B  9600                       DW      150,45
525F  B200                       DW      178,36
5263  E000                       DW      224,30
5267  9600                       DW      150,90
```

```
526B 9600      00061         DW      150,67
526F 9600      00062         DW      150,22
5273 9600      00063         DW      150,90
5277 7000      00064         DW      112,90
527B 6400      00065         DW      100,33
527F 5900      00066         DW       89,151
5283 6400      00067         DW      100,101
5287 5900      00068         DW       89,37
528B 5400      00069         DW       84,160
528F 5900      00070         DW       89,113
5293 5900      00071         DW       89,37
5297 5900      00072         DW       89,151
529B 6400      00073         DW      100,134
529F 983A      00074         DW      15000,2      ; kurze Verschnaufpause vor
               00075                              ;  der nächsten Paraphrase
52A3 3200      00076 HALBZT  DW       50,150      ; <-- DA CAPO-Flag
52A7 9600      00077         DW      150,67
52AB 7700      00078         DW      119,28
52AF 6400      00079         DW      100,134
52B3 6400      00080         DW      100,134
52B7 6400      00081         DW      100,134
52BB 7000      00082         DW      112,90
52BF 6400      00083         DW      100,33
52C3 5900      00084         DW       89,151
52C7 6400      00085         DW      100,67
52CB 6400      00086         DW      100,67
52CF 7000      00087         DW      112,120
52D3 7000      00088         DW      112,60
52D7 5900      00089         DW       89,75
52DB 4800      00090         DW       72,180
52DF 9600      00091         DW      150,67
52E3 9600      00092         DW      150,22
52E7 9600      00093         DW      150,90
52EB 8500      00094         DW      133,50
52EF 7700      00095         DW      119,56
52F3 7000      00096         DW      112,60
52F7 5900      00097         DW       89,75
52FB 6400      00098         DW      100,67
52FF 7700      00099         DW      119,56
5303 7000      00100         DW      112,120
5307 00        00101         DB      0            ; Endekennung des Liedes
5200           00102         END     start

00000 Fehler

00000 Fehler

DACAPO 5235      HALBZT 5A23      frequ  522F      knack  522D
loop0  5203      loop1  5206      loop2  521A      start  5200
tab    5247
```


Von 8 auf 24 Pins

Computerfreaks sind von der Seuche infiziert, immer mehr und bessere Geräte haben zu müssen. Gott sei Dank! Wer sich in diesen Zeiten einen neuen Drucker kauft, wird wohl je nach Geldbeutel auf einen mit 24 Nadeln umsteigen. Wohl getan! Bei Listings und Texten ist der Unterschied nur in den schöneren Zeichen zu erkennen, die eben von dreimal so vielen Pins gemalt werden. In der Graphik aber wird man seinen teuren Drucker gelegentlich verfluchen. Wenn man sie wie gewohnt mit einem Byte pro Punktspalte erzeugt, kommt ein Streifenmuster heraus, denn es wird dabei nur jeder dritte Pin abgefeuert. Hier ein Vergleich zwischen 8- und 24-Nadel-Modus:

Das ist 8-Pin-Graphik eines Phantasiemusters: $
und das ist 24-Pin-Graphik desselben Musters: $

Wie man sieht (falls man das obere Zeichen in der Reproduktion überhaupt sieht), machen Querstreifen nicht immer dick; hier machen sie reichlich dünn. Es kommt deshalb darauf an, den restlichen 16 Nadeln etwas zu tun zu geben. Dabei genügt es nicht, den einleitenden Codestring zur Definition von 24-Pin-Graphik auszugeben und dann das Datenbyte dreimal zum Drucker zu schicken. Das sähe nämlich so aus:

Dies ist 24-Pin-Graphik von Quark: $

Da stehen nur drei eng gequetschte Graphik-Dollars übereinander. Statt dessen muß jedes Bit des auszugebenden Bytes verdreifacht werden. Dazu genügt der Akku nicht mehr. Man braucht drei Register, die hilfsweise je eines der drei senkrecht übereinanderstehenden Graphikbytes beinhalten.

Das im Anschluß an diesen Text gelistete Programm rettet daher zunächst alles, was im Programmverlauf verändert wird, außer dem Akku (das wäre überflüssig). Die Hauptsache passiert in der inneren Schleife loop2. Dreimal wird das jeweilige Bit 7 des Akkus ins Carry-Flag rotiert. Jedesmal wird dann dieses Bit nach E geschoben, dessen Bit 7 ins Carry. Von dort gelangt es ins Bit 0 von D, wobei das Bit 7 von D ins Carry geht. Und das wiederum wird zum Bit 0 von H. Simpler, aber weniger genau ausgedrückt: Der Akku wird in dreifacher Spreizung in die drei Hilfsregister rotiert.

In der äußeren Schleife loop1 werden die 8 Bits des ursprünglichen Graphikbytes abgezählt und in der beschriebenen Weise behandelt. Am Ende steht in den drei Registern H, D und E das verdreifachte Graphikbyte. Die drei Register werden nacheinander ausgedruckt, die alten Registerinhalte sodann restauriert.

Dies ist nur eine Hilfsroutine, die in ein eigentliches Arbeitsprogramm eingefügt werden muß, z. B. in TSCRIPS, das seine Sonderzeichensätze über die 8-Nadel-Graphik doppelter Dichte erzeugt. Es versteht sich, daß der einleitende Codestring zur Definition der Graphikart (Dichte, Geschwindigkeit, Anzahl der Nadeln) auf die 24 Pins angepaßt werden muß. Je nach Drukker, je nach Rahmenprogramm kann das sehr unterschiedlich aussehen.

Um überhaupt auszuprobieren, wie eine Graphik von 8 auf 24 Nadeln umgerechnet werden kann, dient das am Ende gelistete Testprogramm. Beim Label gra2 stehen die Codes, die sich zu dem oben ausgedruckten Dollar zusammensetzen. Der Rest sind Steuercodes (abgesehen von den Texten), die je nach Drucker variieren können (hier für den NEC P6).

Man darf nun nicht erwarten, daß die 8-Pin-Graphik mit 24 Nadeln wie gemalt aussieht. Wenn die einzelnen Bits verdreifacht werden, verschwinden nur die Streifen. Das Graphikzeichen sieht ein wenig klötzchenartig aus, denn noch immer sind es nur 8 verschiedene, bloß eben verdreifachte Informationen, die eine Punktspalte ausmachen.

Arnulf Sopp

```
                  00001 ;         Kernprogramm für die Konversion von 8- auf 24-Pin-Graphik
                  00002
0000  E5          00003           PUSH   HL            ;alle Register retten
0001  D5          00004           PUSH   DE
0002  C5          00005           PUSH   BC
0003  4F          00006           LD     C,A           ;Ursprungsbyte (für 8 Pins) retten
0004  0608        00007           LD     B,8           ;Zähler für 8 Bits des Ursprungsbytes
0006  68          00008 loop1     LD     L,B           ;retten
0007  0603        00009           LD     B,3           ;Zähler für Verdreifachung der Bits
0009  79          00010 loop2     LD     A,C           ;Datenbyte laden
000A  07          00011           RLCA                 ;Bit 7 -) Cy
000B  CB13        00012           RL     E             ;Cy -) Bit 0 v. E (unteres 24-Pin-Byte)
000D  CB12        00013           RL     D             ;dto.          D (mittleres)
000F  CB14        00014           RL     H             ;dto.          H (oberes)
0011  10F6        00015           DJNZ   loop2         ;bis 1 Bit verdreifacht ist
0013  45          00016           LD     B,L           ;Bitzähler
0014  4F          00017           LD     C,A           ;rotiertes Byte als neuer Ausgangswert
0015  10EF        00018           DJNZ   loop1         ;bis 8 Bits verdreifacht sind
0017  7C          00019           LD     A,H           ;oberes Datenbyte (für 24 Pins)
0018  CDB405      00020           CALL   05b4h         ;ausdrucken
001B  7A          00021           LD     A,D           ;mittleres Datenbyte
001C  CDB405      00022           CALL   05b4h
001F  7B          00023           LD     A,E           ;unteres Datenbyte
0020  CDB405      00024           CALL   05b4h
0023  C1          00025           POP    BC            ;Register restaurieren
0024  D1          00026           POP    DE
0025  E1          00027           POP    HL
0026  C9          00028           RET                  ;falls es ein Unterprogramm war
                  00029
0000              00030           END

00000 Fehler

loop1 0006     loop2 0009
```

```
                  00001 ;         Testprogramm zur Konversion von 8- auf 24-Pin-Graphik
                  00002
5200              00003           ORG    5200h
                  00004
5200  213D52      00005 start     LD     HL,gra1       ;Codestring für ein Graphikmuster, 8 Pins
5203  CD6A44      00006           CALL   446ah         ;den String auf den Drucker ausgeben
5206  217852      00007           LD     HL,gra3       ;Einleit. Gr. doppelter Dichte, 24 Pins
5209  CD6A44      00008           CALL   446ah         ;ausgeben
520C  216F52      00009           LD     HL,gra2       ;Anfang der 8-Pin-Datenbytes
520F  0608        00010           LD     B,8           ;Zähler der Datenbytes für 8 Pins
5211  C5          00011 loop0     PUSH   BC            ;retten
5212  E5          00012           PUSH   HL            ;dto. Datenzeiger
5213  4E          00013           LD     C,(HL)        ;Ursprungsbyte (für 8 Pins) laden
5214  0608        00014           LD     B,8           ;Zähler für 8 Bits des Ursprungsbytes
5216  68          00015 loop1     LD     L,B           ;retten
5217  0603        00016           LD     B,3           ;Zähler für Verdreifachung der Bits
5219  79          00017 loop2     LD     A,C           ;Datenbyte laden
521A  07          00018           RLCA                 ;Bit 7 -) Cy
521B  CB13        00019           RL     E             ;Cy -) Bit 0 v. E (unteres 24-Pin-Byte)
521D  CB12        00020           RL     D             ;dto.          D (mittleres)
521F  CB14        00021           RL     H             ;dto.          H (oberes)
5221  10F6        00022           DJNZ   loop2         ;bis 1 Bit verdreifacht ist
5223  45          00023           LD     B,L           ;Bitzähler
5224  4F          00024           LD     C,A           ;rotiertes Byte als neuer Ausgangswert
5225  10EF        00025           DJNZ   loop1         ;bis 8 Bits verdreifacht sind
```



```
5227  7C          00026           LD     A,H           ;oberes Datenbyte (für 24 Pins)
5228  CDB405      00027           CALL   05b4h         ;ausdrucken
522B  7A          00028           LD     A,D           ;mittleres Datenbyte
522C  CDB405      00029           CALL   05b4h
522F  7B          00030           LD     A,E           ;unteres Datenbyte
5230  CDB405      00031           CALL   05b4h
5233  E1          00032           POP    HL            ;Datenzeiger der 8-Pin-Graphik
5234  23          00033           INC    HL            ;erhöhen
5235  C1          00034           POP    BC            ;Bytezähler
5236  10D9        00035           DJNZ   loop0         ;bis alle Datenbytes verdreifacht sind
5238  3E0D        00036           LD     A,0dh         ;String mit CR abschließen
523A  C3B405      00037           JP     05b4h         ;CR drucken und zurück ins DOS
                  00038
523D  44          00039 gra1      DM     'Das ist 8-Pin-Graphik eines Phantasiemusters:',09h
526B  1B          00040           DB     1bh,'L',08h,00h ;Definition für doppelte Dichte, 8 Pins
526F  32          00041 gra2      DB     32h,32h,49h,0ffh,0ffh,49h,26h,26h,0dh   ;Datenbytes
5278  75          00042 gra3      DM     'und das ist 24-Pin-Graphik desselben Musters:',09h
52A6  1B          00043           DB     1bh,'*',21h,08h,00h,03h ;Def. für dopp. Dichte, 24 Pins
                  00044
5200              00045           END    start         ;dort Einsprung

00000 Fehler

gra1  523D    gra2  526F    gra3  5278    loop0 5211    loop1 5216    loop2 5219    start 5200
```

Raumschiffe! ... Sie haben meine Kommandozentrale zerstört.

HEFT
22
Dezember
1987

TSCRIPS und die 24 Nadeln

Im letzten Beitrag zu diesem Thema wurde gezeigt, wie eine Graphik für den Drucker von 8 auf 24 Nadeln umgerechnet werden kann. Ein für den Vielschreiber alltäglicher Anwendungsfall ist TSCRIPS. Dieses Textprogramm erzeugt Sonderschriften über die Drucker-HRG. Es geht von einem Printer mit 8 oder 9 Nadeln aus. Wenn einer mit 24 Pins angeschlossen ist, entsteht ein streifiges Muster, weil nur 8 Nadeln benutzt werden. In der Folge soll gezeigt werden, wie alle 24 Pins eine selbst definierte (oder in einem IMAGEx/DAT-File ergatterte) Schrift drucken. Die Steuercodes gelten für den NEC P6. Da er Epson-kompatibel ist, kann das Programm wohl ohne Änderungen von den meisten übernommen werden, die einen 24-Nadel-Drucker haben.

Dennoch empfiehlt es sich, es zunächst nur mit einer Kopie von TSCRIPS auszuprobieren. In meinem Textprogramm habe ich nämlich schon dermaßen herumgewühlt, daß ich selbst nicht mehr weiß, ob die zu ändernden Adressen der Originalversion entsprechen. Wenn überhaupt, dann ist es Version 5.4. Bei Problemen will ich versuchen, dem verzweifelten Leser hilfreich in die Seite zu treten.

Den Kern des Programms will ich nicht mehr erläutern. Er ist in dem Artikel "Von 8 auf 24 Nadeln" beschrieben. Bevor er aber erreicht wird, setzen diese neuen Überlegungen an. Das Zeichen "#" leitet den Ausdruck einer Sonderschrift ein. Dies ist der einzige Fall von Graphik, den TSCRIPS unterstützt, wenn man den Drucker nicht direkt mit &-Codes programmieren möchte. So lag es nahe, die #-Routine zu ändern.

TSCRIPS kennt als einzigen Modus die doppelt dichte Graphik mit 8 Nadeln, die mit ESC-'L'-n-m eingeleitet wird. Es wäre möglich, aber recht aufwendig gewesen, diese vier Bytes zu erkennen, abzufangen und zu ändern. Statt dessen bedient sich dieses Programm der Möglichkeit des NEC P6, jeden Graphikmodus einem jeden anderen zuzuordnen. Deshalb wird der Codefolge ESC-'L' (doppelte Dichte, 8 Pins) dieselbe Dichte, jedoch mit 24 Pins angedichtet (ESC-'?'-'L'-33). So kann die einleitende Graphik-Definintion einfach weiterhin unbesehen auf den Drucker ausgegeben werden. Erst die nachfolgenden Graphikdaten müssen umgerechnet werden.

Bis das erste davon ankommt, muß vier Bytes lang gewartet werden: ESC-'L'-n-m. Ein Zähler wird dafür aufgesetzt. Wenn der abgelaufen ist, passiert das, was im letzten Artikel zu diesem Thema beschrieben wurde: Der Inhalt des Akkus wird um das Dreifache gespreizt ausgegeben. Das Resultat ist dieselbe Sonderschrift (oder was auch immer), jedoch ohne die Querstreifen.

Zur Demonstration hier mein (zugegebenermaßen nicht eben bescheidener) Briefkopf, oben mit 8 Pins nach alter Art, darunter mit 24 Nadeln gedruckt:

The HACKTORY
SOFTWARE

A. SOPP
Wakenitzstr. 8
2400 Lübeck 1
Telefon:
0451-79 19 26

The HACKTORY
SOFTWARE

A. SOPP
Wakenitzstr. 8
2400 Lübeck 1
Telefon:
0451-79 19 26

Angelpunkt ist das Erkennen des Zeichens "#" im Text. Es leitet einen Graphikstring sowohl ein als es ihn auch beendet. Daher kann das Ende, das Restaurieren der alten Programmlogik, nur in einem *Austausch* des Originals mit der Umleitung bestehen. Ein direktes Setzen oder Löschen verbietet sich. Gottlob, denn platzsparender geht es nicht.

Im hier wiedergegebenen Listing wird vorausgesetzt, daß an der Stelle 86E3 ein auszugebendes Byte an den Drucker geschickt wird (CALL 05B4h). Wer meine bisherigen TSCRIPS-Erweiterungen benutzt, muß hier CALL check7f oder eine andere CALL-Adresse einsetzen, je nach den Anpassungen auf die eigene Hardware. Und wer eine andere TSCRIPS-Version fährt, sollte sich über den passenden Eingriffspunkt vergewissern. Es ist die Stelle, wo mit OUT (FDh),A oder LD (37E8h),A ein Byte auf den Drucker ausgegeben wird.

So ganz nebenbei werden die erwähnten TSCRIPS-Erweiterungen dabei kürzer. Nur mit "#" wird Graphik eingeleitet. Da die #-Routine Umleitungen patcht, ist in allen anderen Fällen (7Fh für Randausgleich bei Proportionalschrift, 40h als @, nicht als §) eine Überprüfung überflüssig, ob das soeben an den Drucker gesandte Byte etwa zu einer Graphik gehört. Zukünftige Erweiterungen von TSCRIPS dürfen sich weiterhin gerne auf das Graphik-Flag 8941h beziehen. Es ist immer noch =1, wenn gerade Graphik aktiv ist, =0, wenn nicht.

Arnulf Sopp

```
                     ;Patch in TSCRIPS, um Graphik (Sonderzeichensätze) m. 24 Pins zu erzeugen
8942                         ORG     8942h           ;Beginn der #-Routine
8942  CD0030                 CALL    graph           ;in die neue Graphik-Routine verzweigen

                     ;hier wird normalerweise ein Byte an den Drucker gesandt.
86E3                         ORG     86e3h
86E3  CDB405                 CALL    05b4h           ;Byte auf den Drucker ausgeben
86E4                 calladr EQU     $-2             ;die Adresse ändert sich bei Graphik

                     ;ein #-Zeichen wurde erkannt: Änderungen vornehmen
3000                         ORG     3000h           ;oder im Mimem oder so
3000  E5             graph   PUSH    HL
3001  D5                     PUSH    DE
3002  3E05                   LD      A,5             ;4 Zeichen: ESC-L-n-m, dann (5.) Graphik
3004  326030                 LD      (gracnt),A      ;als Zähler setzen
3007  215E30                 LD      HL,gramode      ;Zuordnung von ESC-L zu einem Graphikmod.
300A  3E20                   LD      A,20h           ;XOR-Operand für 1 (-) 33 (8 (-) 24 Pins)
300C  AE                     XOR     (HL)            ;Modus umschalten
300D  77                     LD      (HL),A          ;neu setzen
300E  2AE486                 LD      HL,(calladr)    ;aktuelle Adresse der Druckroutine
3011  112730                 LD      DE,cntdown      ;mögliche Umleitung
3012                 grdevia EQU     $-2
3014  ED53E486               LD      (calladr),DE    ;beide vertauschen
3018  221230                 LD      (grdevia),HL
301B  215B30                 LD      HL,grastrg      ;Codefolge zur Graphik mit 24 Pins
301E  CD6A44                 CALL    446ah           ;auf den Drucker ausgeben
3021  D1                     POP     DE
3022  E1                     POP     HL
3023  3A4189                 LD      A,(8941h)       ;Überschriebenen Befehl nachholen
3026  C9                     RET

                     ;ESC-'L'-n-m wird ausgegeben; mitzählen, bis 1. Graphikbyte kommt
3027  E5             cntdown PUSH    HL
3028  216030                 LD      HL,gracnt       ;Zähler für 4 Codes: ESC-'L'-n-m
302B  35                     DEC     (HL)            ;herunterzählen
302C  E1                     POP     HL
302D  C2B405                 JP      NZ,05b4h        ;Byte ausdr., falls noch kein Gr.-Datum
```

```
                   ;ein Graphikdatum kam an; von 8 auf 24 Nadeln umrechnen
3030  E5                   PUSH  HL            ;alle Register retten
3031  D5                   PUSH  DE
3032  C5                   PUSH  BC
3033  216030               LD    HL,gracnt     ;Zeiger auf Zähler der Steuercodes
3036  34                   INC   (HL)          ;wurde dekrementiert: wieder auf 0 setzen
3037  4F                   LD    C,A           ;Ursprungsbyte (für 8 Pins) retten
3038  0608                 LD    B,8           ;Zähler für 8 Bits des Ursprungsbytes
303A  68           grabyte LD    L,B           ;retten
303B  0603                 LD    B,3           ;Zähler für Verdreifachung der Bits
303D  79           grabit  LD    A,C           ;Datenbyte laden
303E  07                   RLCA                ;Bit 7 -) Cy
303F  CB13                 RL    E             ;Cy -) Bit 0 v. E (unteres 24-Pin-Byte)
3041  CB12                 RL    D             ;dto.          D (mittleres)
3043  CB14                 RL    H             ;dto.          H (oberes)
3045  10F6                 DJNZ  grabit        ;bis 1 Bit verdreifacht ist
3047  45                   LD    B,L           ;Bitzähler
3048  4F                   LD    C,A           ;rotiertes Byte als neuer Ausgangswert
3049  10EF                 DJNZ  grabyte       ;bis 8 Bits verdreifacht sind
304B  7C                   LD    A,H           ;oberes Datenbyte (für 24 Pins)
304C  CDB405               CALL  05b4h         ;ausdrucken
304F  7A                   LD    A,D           ;mittleres Datenbyte
3050  CDB405               CALL  05b4h
3053  7B                   LD    A,E           ;unteres Datenbyte
3054  CDB405               CALL  05b4h
3057  C1                   POP   BC            ;Register restaurieren
3058  D1                   POP   DE
3059  E1                   POP   HL
305A  C9                   RET

                   ;Steuercodes für den Drucker, Zählbyte für die Steuercodes ..
305B  1B           grastrg DB    1bh,'?','L',01h,03h   ;definiert ESC-'L' für 8/24 Pins
305E               gramode EQU   $-2                   ;1 oder 33 -) 8 oder 24 Nadeln
3060  00           gracnt  DB    00h                   ;Zähler für Graphik-Steuercodes

0000                       END

00000 Fehler

calladr 86E4    cntdown 3027    grabit  303D    grabyte 303A    gracnt  3060    gramode 305E
graph   3000    grastrg 305B    grdevia 3012
```


Noch ein Bildschirmschoner

Nachdem Arnulf sein Süppchen für seine Maschine gekocht hat, möchte ich hier eine Möglichkeit für die Minderbemittelten unter uns zeigen, die sich zwar kein GIIIs, aber wenigstens eine 80-Zeichen-Karte geleistet haben. Da ist nämlich der von Arnulf vermißte Video-Schalter drauf. Wenn man an den Steuerport der Karte per OUT eine 4 ausgibt, werden die 80x24 Zeichen dargestellt. Mangels Masse (Inhalt) wird die Mattscheibe auch schön schwarz. Wer will, kann die zwei OUT's (ein- und ausschalten) in Arnulfs Listing einbauen, ich habe aber noch ein anderes, nämlich das wohl bekannte 'BLANK' etwas umgestrickt. Da jetzt kein Buffer mehr gebraucht wird, ist das Ding auch ziemlich kurz und es reagiert im Gegensatz zu Arnulfs Produkt auch, wenn ein Programm etwas ausgeben will und nicht nur auf Knopfdruck.

Alexander Schmid

```
;*****************************************************************
;*                                                               *
;*                         'BLANK/CMD'                           *
;*                                                               *
;*       ROUTINE TO BLANK SCREEN AFTER SET PERIOD                *
;*                                                               *
;* CARL FORD, 256 N VIRGINIA AVE, PENNS GROVE, NJ 08069          *
;*                  (609)-299-0200                               *
;*                                                               *
;*               LAST REVISED - 11/12/84                         *
;*                                                               *
;* Diese Version arbeitet zusammen mit der 80-Zeichen            *
;* Karte von Schmidtke (es darf sonst kein Treiber               *
;* dafür aktiv sein!)                                            *
;*****************************************************************
;
           ORG     2bf5h    ;CSAVE + CLOAD  oder sonstwo

enqint     EQU     4410h    ;INTERRUPT ENQUE ROUTINE
deqint     EQU     4413h    ;INTERRUPT DEQUE ROUTINE
time       EQU     10       ;# OF 6 SECOND PERIODS
keydcb     EQU     4015h    ;KEYBOARD DEVICE CONTROL BLOCK
viddcb     EQU     401dh    ;VIDEO DEVICE CONTROL BLOCK
keytst     EQU     38ffh    ;KEY-PRESSED TEST LOCATION


;          *** INITIALIZE ROUTINES

;*** INSERT KEYBOARD & PRINT MONITORS INTO KEYBOARD QUE
entry      LD      HL,(keydcb+1)     ;GET CURRENT KEYBD VECTOR
           LD      (keyjmp+1),HL     ;PUT AT END OF ROUTINE
           LD      HL,keys           ;KEYBD PROCESSING ADDRESS
           LD      (keydcb+1),HL     ;PUT INTO DCB VECTOR
           LD      HL,(viddcb+1)     ;GET CURRENT VIDEO VECTOR
           LD      (vidjmp+1),HL     ;PUT AT CONTINUATION
           LD      HL,print          ;RESTORE SCREEN VECTOR
           LD      (viddcb+1),HL     ;PUT INTO DCB VECTOR
;          *** PUT "BLANK" INTO INTERRUPT QUE
           LD      DE,intrtn         ;INTERRUPT RTN ADDRESS
           CALL    enqint            ;PUT INTO INTERRUPT QUE
;          *** INTERRUPT PROCESSING ROUTINE
```

```
;            *** DECREMENT TIMER & CHECK FOR 0
intrtn  DB      0               ;WORK AREA
        DB      0               ;WORK AREA
        DB      255             ;# OF 6 SEC INTERVALS
        DB      255             ;LENGTH OF 1ST INTERVAL
        PUSH    AF              ;SAVE REGISTER
        LD      A,(timer)       ;GET CURRENT TIMER COUNT
        DEC     A               ;SHORTEN IT BY 1
        LD      (timer),A       ;SAVE IT
        JR      NZ,exit         ;TIMER NOT 0 YET
;       *** Bildschirm abschalten
        LD      A,4             ;Umschaltbyte laden
        OUT     (0d2h),A        ;Steuerport der 80Z-Karte
;       *** RETURN TO INTERRUPTED PROGRAM
        CALL    deact           ;DISABLE 'BLANK'
exit    POP     AF              ;RESTORE REGISTER
        RET

;       *** REMOVE "BLANK" FROM INTERRUPT QUE
;       FOR USE BY PROGRAMS THAT NEED TO DISABLE "BLANK"
deact   LD      DE,intrtn       ;VECTOR TO ROM ROUTINE
        CALL    deqint          ;REMOVE FROM QUE
        RET                     ;BACK TO CALLING PROGRAM

;       *** ROUTINE FOR MONITORING KEYBOARD
keys    PUSH    AF              ;SAVE REGISTER
        LD      A,(keytst)      ;GET TEST LOCATION
        CP      0               ;KEY PRESSED
        CALL    NZ,restor       ;RESTORE SCREEN, ETC
        POP     AF              ;RESTORE REGISTER
keyjmp  JP      0               ;TO KEYBOARD DRIVER

;       *** ROUTINE FOR INTERCEPTING VIDEO
print   PUSH    AF              ;SAVE REGISTER
        CALL    restor          ;RESTORE SCREEN
        POP     AF              ;RESTORE REGISTER
vidjmp  JP      0               ;TO PRINT DRIVER

;       *** CHECK SCREEN & RESTORE IF BLANKED
restor  EXX                     ;SAVE REGISTERS
        LD      A,(timer)       ;CHECK TIME LEFT
        CP      0               ;SCREEN ALREADY BLANKED?
        JR      NZ,reset        ;IF NOT, GO
        LD      A,0             ;Umschaltbyte
        OUT     (0d2h),A        ;Steuerport der 80Z-Karte
;       *** PUT "BLANK" BACK INTO INTERRUPT ROUTINE
        LD      DE,intrtn       ;ROUTINES ADDRESS
        CALL    enqint          ;RESTORE TO INTERUPT QUE
;       *** RECYCLE TIMER & RETURN TO INTERRUPTED PROGRAM
reset   LD      A,time          ;GET DELAY
        LD      (timer),A       ;RESET DELAY COUNTER
        EXX                     ;RESTORE REGISTERS
        LD      A,0             ;CANCEL KEYPRESS
        RET                     ;BACK TO CALLING ROUTINE

;       ***SET STORAGE SPACE
timer   DEFB    time            ;TIMING COUNTER

        END     entry           ;AUTO EXECUTE FROM "CMD"
```

Ein weiterer Bildschirmschoner

Selten kommt Resonanz auf etwas, das man für den Club schreibt. In der 38. Ausgabe der Zeitung des Münchner Clubs war es der Fall. Alexander Schmid stellte ein Programm vor, das nach einer gewissen Zeit den Bildschirm ausschaltet. Er konterte damit meinen Vorschlag, der freilich unverändert nur auf dem Genie 3s läufig ist. Alexander ist auch bei uns Mitglied, deshalb vermute ich, daß sein Beitrag auch in dieser Nummer des Infos steht.

Seine Routine ist meiner insoweit überlegen, als nicht nur der User den Screen wieder einschalten kann. Wenn ein Programm meint, es müsse etwas anzeigen, sieht man den Bildschirm ebenfalls wieder. Das hat Sinn und Verstand.

Deshalb war ich mit meinem Programm überhaupt nicht mehr zufrieden und schrieb es einfach neu. Die Umleitung für die Interrupts ist dieselbe. Nach einer guten Minute knipsen sie den Bildschirm aus. Angemacht wird er in der neuen Version nicht mehr nur durch einen Tastendruck, sondern wann immer irgendwer irgendwas von der Konsole will. Dabei wird jetzt übrigens auch die Graphik berücksichtigt, die in der ersten Version nicht mit abgeschaltet wurde.

Schönheitsfehler: Es werden *nur* die I/O-Routinen beachtet. Wenn etwa eine Graphik eingeblendet wird, kriegt das Programm nichts davon mit. Noch einer: Der Fernseher geht auch dann an, wenn etwas auf den Drucker ausgegeben wird. Ihn auszuschließen, hätte mehr Aufwand mit entsprechendem RAM-Verschleiß bedeutet. Der Drucker läuft so selten, daß man das in Kauf nehmen kann.

Ohnhin bleibt die Abschaltung des Bildschirms aktiv, auch wenn eine Ein- oder Ausgabe (Tastatur, Bildschirm, Drucker) aktiviert wird. Rd. 100 Sekunden später ist der Bildschirm wieder dunkel.

Zusätzlich kann dieses Feature auch wieder ausgeschaltet werden. Wenn das Programm mit irgendeinem Argument nach Komma oder Blank aufgerufen wird, wird es deaktiviert. Es erschien mir übertrieben, extra zu checken, ob das Zeichen nach dem Programmnamen Y für Yes oder J für Ja oder N für Nein bzw. No lautet. Auch ein A für Abschalten schaltet ab, allerdings auch ein E für (erfolgloses) Einschalten. Der Programmname allein schaltet jedenfalls ein.

Die diversen Ladeadressen (jeweils nach ORG) mögen verwirren. Es sind Speicherplätze, die z. Zt. in meinem DOS gerade noch frei sind. Morgen vielleicht nicht mehr. Die ORG-Adressen können beliebig geändert werden, wenn man an den bezeichneten Stellen keinen freien Speicher hat.

Der Teil, der zunächst im Sektorpuffer liegt, um von dort in das RAM im Tastatur-Adreßbereich verladen zu werden, konnte nicht einfach mit ORG 3B80h für diese Adressen definiert werden. Er wäre an der beim Laden noch selektierten Tastatur zerschellt (Fehlermeldung "Ladeversuch auf ROM-Speicherplatz").

Nach Alexanders Beitrag und meiner früheren Version braucht zur Programmlogik nichts mehr erklärt zu werden. Erwähnenswert sind nur die beiden Kuckuckseier in 4601h und 03CAh. An 4600h (innerhalb der INT-Serviceroutine) steht ein CALL nach 45BEh, der umgeleitet wird (Adreß-Operand in 4601/2h). An dieser Stelle sind alle Register gerettet, so daß man frei darüber verfügen kann.

Ähnlich verhält es sich für 03CAh. An 03C9h wird HL mit der Adresse 03DDh geladen (Operand in 03CA/Bh). Sie kommt als gemeinsame RET-Adresse aller I/O-Routinen auf den Stack. Wenn dort nun die Adresse der Umleitung steht, wird eben statt dessen diese aufgerufen. Sie endet dann mit einem JP nach 03DDh. In beiden Fällen wird HL mit der Fortsetzungsadresse geladen. Dorthin gelangt die jeweilige Umleitung mit JP (HL) am Ende der Prozedur.

Arnulf Sopp

```
                ;*****************************************************************
                ;       Bildschirman- und -ausknipser zur Schonung der Bildröhre
                ;       erweiterte Version für das Genie IIIs
                ;
                ;       nach einer Anregung von Alexander Schmid
                ;
                ;       Arnulf Sopp
                ;*****************************************************************

                ;jegliche I/O soll den Bildschirm und gegf. die Graphik wieder anschalten
03CA                    ORG     03cah           ;Rückkehradresse aller I/O-Routinen
03CA  BF05              DW      switch          ;Umleitung setzen

                ;ein Interrupt soll gegf. den Bildschirm und die Graphik ausschalten
4601                    ORG     4601h           ;CALL-Adresse in der INT-Routine
4601  BF05              DW      switch          ;Umleitung setzen

                ;Switch-Routine, des das RAM 'hinter' der Tastatur zugänglich macht
05BF                    ORG     05bfh           ;in der geänderten Drucker-Routine
05BF  F5        switch  PUSH    AF              ;Akku und Flags retten
05C0  C1                POP     BC              ;aber den Stack sauberhalten
05C1  57                LD      D,A             ;und für die Umleitung puffern
05C2  DBFA              IN      A,(0fah)        ;Systembyte 1
05C4  5F                LD      E,A             ;retten
05C5  F601              OR      01h             ;RAM im mm-I/O-Adreßbereich selektieren
05C7  F3                DI                      ;Interrupts maskieren
05C8  D3FA              OUT     (0fah),A
05CA  C3803B            JP      int_io-offset   ;UP bei gemeinsamem Einsprung aufrufen
05CD  7B        exit    LD      A,E             ;evtl. geändertes Systembyte 1
05CE  D3FA              OUT     (0fah),A        ;neu ausgeben
05DD  C5                PUSH    BC              ;ehem. Akku und Flags
05D1  F1                POP     AF              ;restaurieren
05D2  FB                EI                      ;Interrupts wieder zulassen
05D3  E9                JP      (HL)            ;weiter im Programm

                ;Umleitung der Interrupt-Service-Routine zum Verwalten des Zählers
4200                    ORG     4200h           ;im Skt.puffer, wo es nicht dumm auffällt
0680            offset  EQU     $-3b80h         ;Abstand zur Arbeitsadresse des Programms
4200  E3        int_io  EX      (SP),HL         ;HL <- RET-Adresse (hinter dem Caller)
4201  7C                LD      A,H             ;MSB der RET-Adresse
4202  FE46              CP      46h             ;kommt der CALL aus der INT-Routine?
4204  E3                EX      (SP),HL         ;Stack restaurieren
4205  2021              JR      NZ,io           ;falls nein (dann aus der I/O-Routine)

                ;Interrupt-Umleitung: gegf. den Bildschirm und die Graphik ausschalten
4207  210000            LD      HL,0000h        ;INT-Zähler laden
4208            counter EQU     $-2             ;verändert sich
420A  23                INC     HL              ;erhöhen
420B  CB64              BIT     4,H             ;1000h schon erreicht?
420D  CBA4              RES     4,H             ;nur bis 0FFFh zulassen
420F  22883B            LD      (counter-offset),HL     ;Zähler neu laden
4212  21BE45            LD      HL,45beh        ;Fortsetzungsadresse der INT-Routine
4215  7B                LD      A,E             ;ja, Input von Systembyte 1 (Port FA)
4216  2809              JR      Z,ret_int       ;falls 1000h noch nicht erreicht
4218  E682              AND     82h             ;Bildschirm- und Graphik-Bit isolieren
421A  32AE3B            LD      (vid_bit-offset),A      ;zur Erinnerung dort patchen
421D  7B                LD      A,E             ;Systembyte 1
421E  E67D              AND     7dh             ;Bildschirm- und Graphikbit rücksetzen
4220  5F        return  LD      E,A             ;neues Systembyte 1
4221  F601      ret_int OR      01h             ;mm-I/O bleibt selektiert
4223  D3FA              OUT     (0fah),A        ;Systembyte neu ausgeben
4225  C3CD05            JP      exit            ;dort weitermachen

                ;Umleitung für alle I/O-Routinen: gegf. den Bildschirm wieder einschalten
4228  AF        io      XOR     A               ;A <- 00
4229  B2                OR      D               ;ist etwas eingelesen/auszugeben?
422A  7B                LD      A,E             ;Systembyte 1
422B  2808              JR      Z,ret_io        ;falls nein (I/O-Operation beenden)
422D  F600              OR      00h             ;evtl. Bildschirm und Graphik einschalten
422E            vid_bit EQU     $-1             ;Bits 1 und 7 hier gepuffert
422F  210000            LD      HL,0000h        ;Zähler fängt neu an
4232  22883B            LD      (counter-offset),HL     ;frischen Zähler aufsetzen
4235  21DD03    ret_io  LD      HL,03ddh        ;Fortsetzungsadr. aller I/O-Routinen
4238  18E6              JR      return          ;I/O Routine normal fortsetzen
                endprog

                ;Initialisierung: Feature en- oder disablen
423A  CDD54C    start   CALL    4cd5h           ;Befehlszeiger auf Trennzeichen stellen
423D  2812              JR      Z,initial       ;falls nur CR folgt
423F  7E                LD      A,(HL)          ;noch ein Zeichen: welches?
4240  FE4A              CP      'J'             ;J für JA?
4242  280D              JR      Z,initial       ;falls ja

4244  21BE45            LD      HL,45beh        ;alte Fortsetzungsadresse der INT-Routine
4247  220146            LD      (4601h),HL      ;restaurieren
424A  21DD03            LD      HL,03ddh        ;Fortsetzungsadresse aller I/O-Routinen
424D  22CA03            LD      (03cah),HL      ;restaurieren
4250  C9                RET                     ;weiter im DOS oder in BASIC

4251  210042    initial LD      HL,int_io       ;Beginn des Umleitungsprogramms
4254  11803B            LD      DE,3b80h        ;Zieladresse 'hinter' der Tastatur
4257  013A00            LD      BC,endprog-int_io       ;Länge des Programms
425A  CDA006            CALL    06a0h           ;RAM 'hinter' der Tastatur selektieren
425D  EDB0              LDIR                    ;Programm dorthin übertragen
425F  C3AB06            JP      06abh           ;wieder mm-I/O selektieren und raus

423A                    END     start           ;dort Einsprung

00000 Fehler

counter 4208   endprog 423A   exit    05CD   initial 4251   int_io  4200   io      4228
offset  0680   ret_int 4221   ret_io  4235   return  4220   start   423A   switch  05BF
vid_bit 422E
```

Etwas für die Leute, die nie Zeit haben

Wenn man seine BASIC-Programme schneller machen will, benutzt man meistens Integers. Wenn man 'logisch' vorgeht, kann man dabei noch etwas mehr rausholen. Der Ausdruck X=X+1 kann z.B. durch X=-NOT X und X=X-1 durch X=NOT-X ersetzt werden.

Die Sache ist ganz einfach:

```
        5 = 00000101b
       -5 = 11111011b (Zweierkomplement: NOT 5 plus 1)
    NOT-5 = 00000100b = 4
```

Wer jetzt noch (fast) ganz genau nachmessen will, um wieviel seine Programme jetzt schneller sind, kann das mit folgendem Trick tun:

In den Speichstellen 407CH - 407EH steht der maximale Wert für die Stunden, Minuten und Sekunden der Echtzeituhr, in 44CBH steht, nach wievielen 25ms-Interrups die Uhr weitergestellt werden soll. Wenn man nun in 407CH - 407EH jeweils 99 und in 44CBH eine 1 schreibt, erhält man eine Uhr, die im 40stel Sekunden-Takt bis 999999 zählt (wem die knapp 7 Stunden nicht reichen, der soll sich eine CRAY ins Wohnzimmer stellen).

```
100 ' Genaue Laufzeitmessung
110 '
120 ' 407CH - 407EH: Tabelle zum Erhöhen der Uhrzeit
130 '               : (59 Sekunden, 59 Minuten, 23 Stunden)
140 '
150 ' 44CBH         : Anzahl der Aufrufe
160 '               : (40 entspricht einer Erhöhung pro Sekunde)
163 '
165 ' 44CCH         : Interruptzähler für Uhr
170 '
180 ' 4041H - 4043H: Speicher für Uhrzeit
190 '
200 CMD"t"
210 N=16508:POKE N,99:POKE N+1,99:POKE N+2,99     ' Tabelle
220 N=17611:POKE N,1:POKE N+1,1                   ' Interrupts
230 N=16449:POKE N,0:POKE N+1,0:POKE N+2,0        ' Uhr stellen
240 '
250 CMD"r":FOR N=1 TO 10000:NEXT:CMD"t"           ' oder sonstwas
260 '
270 N=16449:A=PEEK(N+2)*10000+PEEK(N+1)*100+PEEK(N)
280 PRINT A/40"Sekunden"
290 CMD"r"
300 END
```

Wichtig: Wenn die Messung genau sein soll, müssen die Laufwerke stehen! (Solange sich die Floppys drehen sind die Interrupts blockiert und die Uhr läuft nicht).

Vor dem eigentlichen Programm also eine FOR-NEXT-Schleife o.ä. einbauen, oder wie oben die Zeit mit POKE stellen.

Für 'X=X+1' in der Schleife in Zeile 250 habe ich 18.3 Sekunden gemessen, für 'X=-NOTX' aber nur 17.375 Sekunden!

Das ist ungefähr 5% schneller (bei 1.78 MHz wird's wohl genauso aussehen, auch wenn es länger dauert).

Alexander Schmid

Es geht noch schneller

Als ich etwas im GDOS rumgestöbert habe, habe ich einen Trick entdeckt, mit dem man seinen lahmen (Basic-) Programmen nochmal ins Kreuz treten kann:

Beim GDOS 2.4 ist u.a. ein Programm namens NEW dabei, welches eine Basic-Erweiterung darstellt. Unter den zusätzlichen Befehlen findet sich auch &KEYOFF, mit dem man die Tastatur abschalten kann. Alter Hut dachte ich zuerst, mit POKE 16405,0 hat man doch dasselbe. Denkste ! &KEYOFF ist noch schneller und außerdem funktionieren INKEY$ und die BREAK-Taste !

Das tolle an diesem Nachbrenner ist, daß er nur 9 Bytes lang ist (Tatsache). Um das Ding einzubauen zu können braucht man allerdings einen Banker, da hierfür ein CALL im ROM umgeleitet wird. Bei 1D1Eh steht ein CALL 0358h zur Tastaturabfrageroutine, die ja die große Bremse ist. Hierhin wird nun die Adresse des kleinen Zusatzprogramms geschrieben (bei NEW 3352h, kann aber beliebig sein), welches folgendermaßen aussieht:

```
3352 3A4038 LD  A,(3840h)   ; ENTER, BREAK usw. abfragen
3355 E604   AND 04          ; BREAK ?
3357 C8     RET Z           ; nein, weiter im Programm
3358 C35803 JP  0358h       ; ja, BREAK auswerten
```

Einfach, aber genial, was sich die Leute bei Phoenix da ausgedacht haben. Wenn die BREAK-Taste nicht gedrückt wird, wird die übrige Tastatur erst gar nicht abgefragt; nur bei INKEY$ oder INPUT wird der Rest auch noch abgeklappert, da dafür andere Routinen zuständig sind.

In Basic könnte die Sache so aussehen:

```
100 ' &KEYOFF
110 '
120 PRINT TIME$:FOR X=1 TO 10000:NEXT:PRINT TIME$:PRINT
130 '
140 WOHIN=&H2BF5                     ' beliebig
150 FOR X=0 TO 8:READ A:POKE WOHIN+X,A:NEXT
160 DATA 58,64,56,230,4,200,195,88,3
170 US%(0)=8448:US%(2)=8704:US%(4)=201   ' INTEGER !!
180 US%(1)=WOHIN:US%(3)=&H1D1F       ' 7455 dez.
190 DEFUSR=VARPTR(US%(0))            ' Disk-Version
190 A=VARPTR(US%(0)):POKE 16526,A-INT(A/256)*256:POKE 16527,A/256
200 A=USR(0)
210 '
220 PRINT TIME$:FOR X=1 TO 10000:NEXT:PRINT TIME$
230 END
```

Die etwas umständliche Methode, die zwei Bytes nach 1D1Fh/1D20h zu bringen ist notwendig, weil die Bytes auf einmal dorthin müssen. Mit POKE &H1D1F,XX:POKE &H1D20,YY hängt sich der Rechner leider auf und zwar aus folgendem Grund: Ursprünglich steht dort 58 03, unsere Routine steht z.B. bei FFF0h. Nach dem ersten POKE steht dort F0 03. Wenn der Interpreter jetzt die Tastatur abfragt landet er im Acker bei 03F0h und rührt sich nicht mehr, statt auch den 2. Wert zu POKEn und nach FFF0h zu springen.

Wie schon gesagt kann WOHIN irgendwo im RAM liegen. Die neun Bytes dürfen aber nie irgendwie überschrieben werden, da man sich sonst nur noch an den Reset-Knopf halten kann. Für Disk-Besitzer bietet sich z.B. 2BF5h (CSAVE-Routine) herrvorragend an.

Noch was für die Statistiker:

FOR X=1 TO 50000:NEXT :	Disk-Basic	40.125s
(REAL-Zahlen)	POKE 16405,0	26.95s
	&KEYOFF	18.65s

Alexander Schmid

Ein Fehler in SYS28/SYS

Als ausgesprochener Fan von ruinösen Spontankäufen habe ich mir gestern einen bidirektionalen Traktor für den NEC P6 zugelegt. Nicht daß ich etwa allzu viel zu plotten hätte. Es gibt jedoch ein paar andere Features des P6, die ohne Rückwärtstransport des Papiers unschöne Folgen haben können.

Der unidirektionale Traktor ist jetzt nicht mehr nötig. Braucht ihn jemand? Wer sich den NEC P6 kaufen möchte, müßte ihn extra gegen Aufpreis bestellen. Meiner ist billiger. Aber zur Sache:

SYS28 von G-DOS bietet einige mehr oder weniger nützliche Routinen an, die sich auf den Drucker beziehen. Sie müssen je nach Modell gepatcht werden. Das wollte ich gleich gründlich tun und disassemblierte daher SYS28 zunächst. Auffällig waren einige unmotivierte NOPs, die wohl auf früherem Gewühle by Trommelschläger bzw. Phönix beruhen. Bei näherem Hinsehen (wofern das bei ob solch dilettantischer Programmiertechnik verweintem Auge noch drin war) fanden sich massenweise katastrophale Klimmzüge, die nach Kürzung schrieen. SYS28 ist jetzt ca. 300 Bytes kürzer, wobei zu bedenken ist, daß der reine Programmcode ohne Tabellen überhaupt nur ca. 600 Bytes betrug.

Am gravierendsten war ein Fehler, der ausgerechnet beim Library-Befehl DR (Ausgabe von Text aus der DOS-ready-Ebene) auftritt: ESC (CHR$(27)) und Codes kleiner als 20h (Leerzeichen), die nach "#" eingegeben werden können, werden nur als 1. Zeichen im Textstring erkannt. Sofort danach wird der Text ausgedruckt, auch wenn nochmals "$" (für ESC) oder Pound folgen sollte. Das werden dann eben gewöhnliche Dollars oder Pound-Zeichen.

Im Listing ist eine DR-Routine wiedergegeben, die den Vorzug hat, zu funktionieren (ab Label dr). Sie ruft zur Feststellung der Druckerbereitschaft das UP testprt auf. Auch dort habe ich einiges geändert. Jetzt werden hier alle denkbaren Drucker-Fehler berücksichtigt. Übrigens auch mit der zutreffenden Meldung, was vorher auch nicht der Fall war.

Da SYS28 jetzt total umgebaut ist, darf der geneigte Leser sich nicht durch die anderen Ladeadressen beirren lassen. Was nicht für den Befehl DR bzw. die Feststellung des Druckerstatus von Belang ist, ist mit LIST OFF ausgespart, was an den springenden Zeilennummern zu erkennen ist.

Der Rest von SYS28 wurde einfach nur kürzer und schneller, zu korrigieren war nichts. Deshalb möchte ich nicht das komplette neue File vorstellen. Wer Interesse daran hat, kann mich antriggern.

Arnulf Sopp


```
4D28  CDD105   00044 testprt CALL    a05d1          ;Drucker bereit?
4D2B  C8       00045         RET     Z              ;falls ja
4D2C  E5       00046         PUSH    HL             ;Befehlszeiger retten
4D2D  B7       00047         OR      A              ;Drucker ausgeschaltet?
4D2E  0EAF     00048         LD      C,texnopr&0ffh ;Text 'Drucker aus'
4D30  2811     00049         JR      Z,display      ;falls ja
4D32  07       00050         RLCA                   ;Drucker busy?
4D33  0E91     00051         LD      C,texbusy&0ffh ;Text dazu
4D35  380C     00052         JR      C,display      ;falls ja
4D37  07       00053         RLCA                   ;out of paper?
4D38  0E96     00054         LD      C,texopap&0ffh ;Text dazu
4D3A  3807     00055         JR      C,display      ;falls ja
4D3C  07       00056         RLCA                   ;ready?
4D3D  0EB3     00057         LD      C,texnrdy&0ffh ;Text dazu
4D3F  3002     00058         JR      NC,display     ;falls nein
4D41  0EA2     00059         LD      C,texdesl&0ffh ;sonst ist der Drucker deselektiert

4D43  21884D   00060 display LD      HL,texdrck     ;Text 'Drucker'
4D46  CD6744   00061         CALL    a4467          ;anzeigen
4D49  69       00062         LD      L,C            ;HL <- Text je nach dem
4D4A  CD6744   00063         CALL    a4467          ;Fehlertext anzeigen
4D4D  2EC0     00064         LD      L,beep&0ffh    ;abschließender Piepser, CR
4D4F  CD6744   00065         CALL    a4467          ;ausgeben
4D52  E1       00066         POP     HL             ;Befehlszeiger
4D53  CD4900   00067         CALL    a0049          ;auf Tastendruck warten
4D56  FE0D     00068         CP      0dh            ;ENTER gedrückt?
4D58  28CE     00069         JR      Z,testprt      ;falls ja
4D5A  3E39     00070         LD      A,39h          ;sonst erzwungene Beendigung der Funktion
4D5C  C9       00071         RET
               00072
4D5D  CD284D   00073 dr      CALL    testprt        ;Druckerbereitsch. testen, evtl. Meldung
4D60  C0       00074         RET     NZ             ;falls erzwungene Beendigung der Funktion
4D61  CDD54C   00075         CALL    a4cd5          ;auf nächsten Befehlsparameter stellen
4D64  7E       00076 drloop  LD      A,(HL)         ;nächstes Zeichen
4D65  FE24     00077         CP      '$'            ;$ für ESC?
4D67  2002     00078         JR      NZ,nodoll      ;falls nein
4D69  361B     00079         LD      (HL),1bh       ;$ durch ESC ersetzen
4D6B  FE23     00080 nodoll  CP      '#'            ;# stellvertretend für CTRL?
4D6D  2003     00081         JR      NZ,nopound     ;falls nein
4D6F  23       00082         INC     HL             ;ja, nächstes Zeichen
4D70  CBB6     00083         RES     6,(HL)         ;CTRL-Zeichen erzeugen
4D72  7E       00084 nopound LD      A,(HL)         ;evtl. geändertes Byte  laden
4D73  23       00085         INC     HL             ;nächste Stelle im Befehlsstring
4D74  CDB405   00086         CALL    a05b4          ;Byte ausdrucken
4D77  FE0D     00087         CP      0dh            ;CR?
4D79  C8       00088         RET     Z              ;Ende, falls ja
4D7A  18E8     00089         JR      drloop         ;oben weiter
               00090
4D88  44       00098 texdrck DM      'Drucker ',03h
4D91  62       00099 texbusy DM      'busy',03h
4D96  6F       00100 texopap DM      'ohne Papier',03h
4DA2  64       00101 texdesl DM      'deselektiert',03h
4DAF  61       00102 texnopr DM      'aus',03h
4DB3  6E       00103 texnrdy DM      'nicht bereit',03h
4DC0  1E       00104 beep    DB      1eh,07h,0dh    ;Rest der Zeile löschen, Beep, CR

00000 Fehler
```

The Times, They Are A-Changing

Wer keine Echtzeituhr eingebaut hat, kann diese Seiten überschlagen. Wer zwar eine hat, aber am Wochentag nicht interessiert ist, auch (CP/M und CALVA-DOS, das ehemalige H-DOS, berücksichtigen ihn aber). Wer jedoch seine eingebaute Kalenderuhr mit *einem* Befehl vollständig programmieren will, könnte es mit meinem Programm versuchen:

Es hat eine ähnliche Funktion wie die Library-Befehle ZEIT und DATUM (TIME und DATE in NewDOS-80) und eignet sich daher zum Einbau in ein SYS-File, beispielsweise als Erweiterung des ZEIT-Befehls. Die Echtzeituhr ist jedoch batteriegepuffert und muß recht selten nachgestellt werden. Für die wenigen Male sind mir die Systemprogramme zu wertvoll.

Die Eingabesyntax entspricht den Befehlen ZEIT und DATUM. Zusätzlich kann der Wochentag eingegeben werden, und zwar abgekürzt als MO, DI, MI ... Es müssen nicht alle Zeitarten programmiert werden, die Reihenfolge ihrer Eingabe ist gleichgültig. Kleinbuchstaben beim Wochentag sind zulässig. So könnte die Programmierung der RTC aussehen:

xyz,FR,23.10.87,17:59:00<ENTER>

Dabei bedeutet xyz den Namen, mit dem das Programm natürlich zunächst aufgerufen werden muß.

In einer Endlosschleife wird der Befehlsstring, der dem Programmnamen folgt, nach zulässigen Zeitangaben durchforstet. Sie werden zur Bearbeitung dem System (Uhrzeit und Datum) bzw. dem Segment für den Wochentag übergeben. Eine Überprüfung der Anzahl solcher Angaben findet nicht statt. Daher hat man die Möglichkeit, dem obigen Beispiel vor <ENTER> noch DI für Dienstag anzuhängen, falls man etwa den falschen Tag eintippte (heute ist aber Freitag). Es gilt jeweils die letzte Eingabe.

Je nach Computer und Betriebssystem können Änderungen erforderlich sein. Die Ports 5Ah (Datenport der RTC) und 5Bh (Adreßport) können sich unterscheiden. Auch die Requestcodes für die Befehle ZEIT und DATUM müssen nicht mit diesem Programm übereinstimmen. Hier sollte der Leser seine Unterlagen zu Rate ziehen.

Arnulf Sopp


```
                 ;        Utility zum vollständigen Programmieren der Echtzeituhr

5200                      ORG     5200h           ;oder wo auch immer Platz ist

5200  CDD54C     start    CALL    4cd5h           ;Trennzeichen lesen, Befehlszeiger weiter
5203  200C                JR      NZ,nextpar      ;falls dem Befehl etwas folgt

                 ;das Folgende ist eine Spezialität des CALVA-DOS
                 ;für andere DOSes: weglassen und statt dessen oben RET Z programmieren
5205  3EBB                LD      A,0bbh          ;Requestcode für DOS-Befehl SYS,CL
5207  EF                  RST     28h             ;Zeitanzeige durchführen und raus

                 ;Unterprogramme zur Ausführung der DOS-Befehle ZEIT und DATUM
5208  0E0A       zeit     LD      C,0ah           ;Code für ZEIT-Befehl in SYS7/SYS
520A  1802                JR      jp_sys7         ;dort weiter
520C  0E0B       datum    LD      C,0bh           ;Code für DATUM-Befehl in SYS7/SYS
520E  3EE9       jp_sys7  LD      A,0e9h          ;Requestcode für SYS7/SYS
5210  EF                  RST     28h             ;DOS-Befehl ZEIT oder DATUM durchführen

                 ;prüfen, ob Uhrzeit, Datum oder Wochentag eingegeben wurde
5211  E5         nextpar  PUSH    HL              ;Befehlszeiger retten
5212  7E                  LD      A,(HL)          ;nächstes Zeichen im Befehlsstring
5213  FE3A                CP      '9'+1           ;folgt im Befehl eine Ziffer?
5215  3829                JR      C,numeric       ;falls ja (Uhrzeit oder Datum)

                 ;Wochentag wurde eingegeben: den Tag in der Tabelle aufsuchen
5217  4F                  LD      C,A             ;BC <- die beiden nächsten Zeichen
5218  23                  INC     HL              ;nächstes Zeichen im Befehlsstring
5219  46                  LD      B,(HL)
521A  215E52              LD      HL,daytab       ;Tabelle der Tagesnamen
521D  3E07                LD      A,7             ;Zähler für 7 Tage
521F  5E         seekday  LD      E,(HL)          ;DE <- Name eines Tages
5220  23                  INC     HL              ;Zeiger nachstellen
5221  56                  LD      D,(HL)
5222  23                  INC     HL              ;Zeiger auf den nächsten Tagesnamen
5223  EB                  EX      DE,HL           ;HL <- Tagesname
5224  ED42                SBC     HL,BC           ;Tagesnamen mit dem in der Tabelle vergl.
5226  2807                JR      Z,founday       ;falls den richtigen gefunden
5228  EB                  EX      DE,HL           ;Tabellenzeiger restaurieren
5229  3D                  DEC     A               ;Zähler erniedrigen
522A  20F3                JR      NZ,seekday      ;falls noch ein paar Versuche frei

                 ;Aufruf von DATUM mit falschen Parametern -> Fehlermeldung und raus
522C  E1         error    POP     HL              ;Stack bereinigen
522D  18DD                JR      datum           ;DATUM mit falschen Params: Fehlermeldung

                 ;Wochentag existiert in der Tabelle: Tag auf die Echtzeituhr ausgeben
522F  015B6A     founday  LD      BC,6a5bh        ;RTC-Reg. 6, WR-Flag, Adreßport 5B
5232  ED41                OUT     (C),B           ;Reg. 6 (Wochentage) für WR adressieren
5234  3D                  DEC     A               ;Tage werden ab 00 gezählt
5235  D35A                OUT     (5ah),A         ;Wochentag auf RTC ausgeben
5237  ED69                OUT     (C),L           ;L=00, beendet die Programmierung
5239  010200              LD      BC,0002h        ;2 Zeichen bis zum nächsten Teil des Bef.

                 ;gemeinsame Fortsetzung: Befehlszeiger nachstellen und das Ganze von vorn
523C  E1         goread   POP     HL              ;Befehlszeiger restaurieren
523D  09                  ADD     HL,BC           ;Befehlszeiger auf den nächsten Teil
523E  18C0                JR      start           ;Befehlsstring weiter interpretieren

                 ;Uhrzeit oder Datum wurde eingegeben: entsprechenden DOS-Befehl aufrufen
5240  FE30       numeric  CP      '0'             ;noch eine erlaubte Ziffer?
5242  38E8                JR      C,error         ;falls unsinniges Zeichen: Fehlermeldung
5244  23                  INC     HL              ;Zeiger auf Trennzeichen v. Zeit o. Datum
5245  23                  INC     HL
5246  7E                  LD      A,(HL)          ;Trennzeichen der Zeiteinheiten laden
5247  2B                  DEC     HL              ;zurück auf 1. Zeichen
5248  2B                  DEC     HL
5249  FE2E                CP      '.'             ;Datum? (Trennzeichen '.')
524B  2809                JR      Z,setdate       ;falls ja
524D  FE3A                CP      ':'             ;Uhrzeit? (Trennzeichen ':')
524F  20DB                JR      NZ,error        ;Fehleranzeige, falls nein
5251  CD0852              CALL    zeit            ;DOS-Befehl ZEIT durchführen
5254  1803                JR      adjptr          ;nächsten Teil des Befehls bearbeiten
5256  CD0C52     setdate  CALL    datum           ;DOS-Befehl DATUM durchführen
5259  010800     adjptr   LD      BC,0008h        ;8 Zeichen bis zum nächsten Befehlsteil
525C  18DE                JR      goread          ;Befehlsstring weiter interpretieren

                 ;Tabelle der Wochentagsnamen
525E  53         daytab   DM      'SO','SA','FR','DO','MI','DI','MO'

5200                      END     start           ;hier Einsprung

00000 Fehler
```

BDBS - Basic Datenbanksystem

Seit kurzem befindet sich in unserer Programmbibliothek ein Datenbanksystem für das Model 4/4P. Das Programmpaket umfaßt insgesamt 7 Programme, 1 Hauptprogramm und 6 Overlay-Programme. Es wurde von Hardin Brothers geschrieben und im März 1986 in der 80 MICRO veröffentlicht. Die Programme sind in einer übersichtlichen und strukturierten Form geschrieben und lassen sich dadurch leicht an eigene Bedürfnisse anpassen bzw. abändern, was durch die Verwendung von verständlichen Variablennamen und Kommentaren noch zusätzlich unterstützt wird.

Nachfolgend nun eine kurze Beschreibung anhand des Programm-Menüs:

<u>1. Definition einer Datenbank</u>
Innerhalb einer Datenbank können bis zu 500 Datensätze bearbeitet werden. Es ist jedoch nicht weiters schwierig, diese Zahl zu erhöhen. Insgesamt kann man 40 Datenfelder mit je 30 Zeichen definieren.

<u>2. Eröffnung einer Datenbank</u>
Eingabe des Namens und des Laufwerks der Datenbank

<u>3. Neue Datensätze zur Datenbank hinzufügen</u>

<u>4. Datensätze lesen, suchen, ändern und löschen</u>
Hier hat man die Möglichkeit, einen Datensatz entweder direkt unter der lfd. Nummer aufzurufen oder ihn nach einem einzugebenden Suchkriterium innerhalb der definierten Datenfelder suchen zu lassen. Beim Suchen ist zu beachten, daß keine Buchstabenkonvertierung vorgenommen wird. Es bietet sich bereits bei der Eingabe an, alle Worte z.B. in Großbuchstaben zu schreiben. Oder man baut sich eine entsprechende Konvertierungsroutine ein.

<u>4. Auswahldateien bilden</u>
Es lassen sich insgesamt sechs Datenfelder miteinander vergleichen (Vergleichsmöglichkeiten eines Datenfelds: <; <=; =; >=; >; <> entweder mit einem anderen Datenfeld oder einer Konstanten). Zudem besteht die Möglichkeit diese sechs Datenfelder untereinander durch die logischen Operatoren AND und OR miteinander zu verknüpfen. Die Auswahl erfolgt entweder von der gesamten Datenbank, der sortierten Indexdatei oder der alten Auswahldatei. Die Auswahldatei wird auf der Diskette abgespeichert.

<u>5. Sortierte Indexdateien bilden</u>
Mit diesem Menüpunkt lassen sich sortierte Dateien entweder von der gesamten Datenbank oder der Auswahldatei erstellen. Nach Auswahl eines Datenfeldes wird die Datei sortiert und in einer Indexdatei abgespeichert.

<u>6. Ausdrucke erstellen</u>
Man kann hier entweder von allen Datensätzen, von der Auswahldatei oder der sortierten Indexdatei Ausdrucke erstellen. Diese kann man sich entweder über Bildschirm oder Drucker anschauen oder für späteren Bedarf in einer Datei abspeichern lassen.

Sicherlich kann das Programm nicht mit einer kommerziellen Datenbank konkurieren, zumal der Basic-Interpreter bekannterweise ja nicht das Schnellste ist was es gibt. Aber für den Hausgebrauch finde ich das Programm recht gut. Vor allem braucht man keine umfangreiche Befehlssyntax kennen, um mit dem Programm arbeiten zu können. Außerdem läßt es sich leicht ergänzen bzw. ändern.
Beim Einsatz auf einem Model 4 mit 128 kB Speicher läßt sich die RAM-Disk gut als schneller Datenspeicher oder (wenn man innerhalb des Menüs öfters wechselt) zum schnellen Laden der Programm-Module einsetzen.

Vor der erstmaligen Benutzung sollte man im Programm DEFINE/OVL am Schluß der Zeile 2140 folgende Ergänzung vornehmen:

```
: DIM FIELD.NUMBER(FIELDS), LABEL$(FIELDS), EOL(FIELDS)
```

Eine weitere Änderung ist erforderlich, wenn man das Datum im Format TT/MM/JJ eingeben möchte. Hierzu müssen in MAIN/BAS in der Zeile 180 die "/" durch "." ersetzt werden. In Zeile 190 ist dieses nochmals notwendig; außerdem müssen die Vergleichszahlen "12" und "31" miteinander vertauscht werden.

Wie bereits oben erwähnt kann man das Programm für eigene Bedürfnisse ergänzen. Mir gefiel z.B. nicht, daß normalerweise nur ein Ausdruckformat und eine Auswahldatei zur Verfügung steht. Möchte man einen anderen Ausdruck oder andere Auswahlkriterien, bleibt einem nichts anderes übrig, als die bereits erstellten Dateien zu überschreiben. Auf Dauer ist dies jedoch keine brauchbare Lösung.

<u>Ergänzung des Basic Datenbanksystems</u>
Zur Erweiterung der Benutzungsmöglichkeiten werden die Ausdruck(/RPT) und Auswahl(/SLC) - Dateien durch Anhängen einer Nummer an den Dateinamen erweitert. Somit lassen sich bis zu je 99 versch. Dateien benutzen.
Benötigt wird die Nummer zur Erstellung von

1. versch. Ausdruckformaten (.../RPT)
2. versch. Auswahldateien (.../SLC)

Nicht erforderlich ist sie dagegen für die .../SEL-Datei. In dieser Datei werden die durch die .../SLC-Datei ausgewählten Datensätze abgespeichert. Da sich die .../SEL-Datei bei einer Ergänzung der Datensätze sowieso ändern würde, ist es nicht erforderlich diese für längere Zeit abzuspeichern. Gleiches gilt für die .../IND-Datei.

Der Datenbanknamen wird auf max. 6 Stellen begrenzt. Die restlichen zwei Stellen werden für die Nummer benötigt.

Erforderlich Änderungen der Programme:

Eingabe der Nummer:
in MAIN/BAS:

```
1120 ' Eingabe der Dateinummer (2stellig)
1122 NUMMER$="":
     PRINT "Geben Sie die Nummer der Datei ein (2stellig) ":
     INPUT " ==> ",NUMMER$
1124 IF LEN(NUMMER$) <> 2 THEN 1122
1126 IF VAL(NUMMER$) < 1 OR VAL(NUMMER$) > 99 THEN 1122
1128 RETURN
```

Datenbanknamen auf 6 Stellen begrenzen:
in MAIN/BAS:
Z 504: .... 6 ....

in DEFINE/OVL:
Z 2002: .....6 .....

Änderung der OVL-Programme/Ergänzung der Dateinamen mit Nummer
in DEFINE/OVL
Z 2124: GOSUB 1122: TEST$ = FILE$ + NUMMER$ + ...
Z 2190: ....: OPEN "O",1,FILE$ + NUMMER$ + ...

in SELECT/OVL
Z 2014: GOSUB 1122: TEST$ = FILE$ + NUMMER$ + ...
Z 2032: ... THEN GOSUB 1122: OPEN "O",3,FILE$ + NUMMER$ + ...

in REPORT/OVL
Z 2002: GOSUB 1122: TEST$ = FILE$ + NUMMER$ + ...
Z 2030: .... OPEN "I",1,FILE$ + NUMMER$ + ...

So und nun wünsche ich viel Spaß beim Arbeiten mit BDBS. Eine kurze Bedienungsanleitung befindet sich beim Programmpaket. Eine ausführliche Beschreibung findet sich in 80 MICRO, März 1986.

Klaus Hermann

Noch Interesse ?

Teil zwei der Vorstellung von PASCAL.

Beim letzten Mal haben wir uns schon mit einigen Datentypen und Kontrollstrukturen beschäftigt. Heute wird das noch ein wenig vertieft. Zu den vordefinierten Funktionen, die bereits im Sprachumfang enthalten sind, gehören neben denen aus der letzten Folge auch noch :

SUCC(X)
liefert den nächst höheren Wert der Variable X zurück, die ein Typ mit Ordnungsfunktion sein muß.

PRED(X)
liefert entsprechend den nächst niedrigeren Wert.

Die Funktion SUCC() wird oft benutzt, um INTEGER-Variablen zu inkrementieren, d.h. um 1 zu erhöhen. Das ist etwas schneller als das übliche Verfahren : I := I + 1;

Das in der letzten Folge vorgestellte Programm litt am Mangel von Hilfsmitteln zur Ein/Ausgabe von Daten. Dieses Manko soll nun behoben werden. Prinzipiell geht jedes PASCAL I/O über Files vonstatten. Zwei davon sind bereits vordefiniert und bedürfen keinerlei Deklaration : die Files *INPUT* und *OUTPUT*. Das sind die Bezeichner für Terminal-I/O. Schreiben kann man nach OUTPUT, lesen von INPUT. Dabei sind diese Bezeichner physikalisch mit den Terminal I/O-Devices identisch. Weil I/O via Terminal so häufig vorkommt, kann man die Bezeichner INPUT und OUTPUT einfach weglassen. Die Prozeduren, die in PASCAL das I/O übernehmen, heißen *READ* und *WRITE*. Beide Prozeduren haben Parameterlisten, das ganze geht ähnlich über die Bühne wie man das von den BASIC-Anweisungen INPUT und PRINT kennt. Ein kleines Beispiel :

```
WRITE( Dies ist eine Textkonstante );
```

gibt eben diese Konstante am Bildschirm aus, völlig gleichwertig ist die Anweisung :

```
WRITE(OUTPUT, Dies ist eine Textkonstante );
```

Konstanten einlesen kann man natürlich nicht, da ein Einlesen ja einer Zuweisung entspricht. Als Beispiel dafür ein Programmfragment :

```
VAR x : REAL;
....
READ(X);
```

READ und WRITE sind sogenannte generische Prozeduren, das bedeutet, daß die Anzahl der Parameter nicht feststeht, sondern erst während der Übersetzung bestimmt wird. Man kann damit alle vordefinierten Typen behandeln, allerdings nicht alle selbst definierten (das geht nämlich auch in Pascal, Datentypen selbst basteln !). Aber zuerst wollen wir doch mal sehen, was es sonst noch an Datentypen gibt. Da wäre erst mal, unverzichtbar, das Feld oder der Vektor oder das *ARRAY*. Felder kann man von allem definieren, was einem Spaß macht, auch von Feldern oder von Files oder von Steckrüben...
Allerdings hat Pascal eine gravierende Einschränkung : die Grenzen eines Feldes müssen Konstanten sein. Dynamische Felder gibt es nicht, aber was viel feineres...

Kommen wir zur praktischen Anwendung, diesmal ein kleines Beispiel aus der Mathematik. Seien A und B Vektoren im dreidimensionalen Raum, also darstellbar durch jeweils ein Feld mit 3, sagen wir REAL, Elementen. Dann gibt es eine Verknüpfung dieser beiden Vektoren, das sogenannte Skalarprodukt, welches wie folgt definiert ist :

$A \circ B := SUMME(A_i * B_i)$, $i = 1..3$

(gerade jetzt wünsche ich mir ein Textsystem, das mathematische Formeln darstellen kann). Das kleine $_i$ ist als Index zu verstehen.
In Pascal sieht das Programm vielleicht so aus wie das kleine Beispiel. Dazu ist noch etwas zu bemerken :
KOMMENTARE werden in Klammern eingeschlossen, und zwar entweder in die Kombination (*....*) oder in die geschweiften Klammern. Die Prozeduren READLN und WRITELN sind sind identisch mit READ bzw. WRITE, erzeugen aber nach Eingabe bzw. Ausgabe einen Zeilenvorschub.
Im Programm ist außerdem noch eine *FOR*-Schleife zu sehen. Der Schleifenkörper ist dabei immer eine Anweisung. Das kann auch eine zusammengesetzte Anweisung sein. Eine zusammengesetzte Anweisung besteht aus einer Anzahl von einfachen Anweisungen, welche in die Schlüsselworte *BEGIN* und *END* eingeklammert sind. Hört sich komisch an... Vielleicht tippt ihr das Beispiel mal ab und experimentiert ein wenig damit. Man könnte die Anzahl der ARRAY-Elemente vergrößern und rausfinden, was ein Skalarprodukt im fünfdimensionalen Raum ist.

Felder stellen ein außerordentlich nützliches Hilfsmittel dar, aber wem erzähle ich das. Bis jetzt gab es, z.B. gegenüber BASIC, noch wenig neues zu vermelden. Betrachten wir nun eine der großen Pascal-Stärken ; Selbstdefinierte Typen. Formal dient dazu das Schlüsselwort *TYPE*. In strengen Implementationen muß dieses Wort vor VAR stehen, in TURBO nicht. Hinter TYPE folgt der Name des zu definierenden Typs, ein Zeichen '=' und dann die Beschreibung des Typs, beendet wird die Deklaration mit dem Semikolon.
Nehmen wir ein Beispiel. Der Typ INTEGER enthält bekanntlich sowohl positive wie auch negative Zahlen. Stellen wir uns eine Anwendung vor, bei der eine Variable nur ganzzahlige Werte zwischen 1 und 12 annehmen kann, sagen wir, sie repräsentiere einen Monat. Es ist nun sinnvoll, einen Typ für die Variable zu erfinden. Also :

```
...
TYPE MONAT = 1..12;
VAR M : MONAT;
...
```

Nun werdet ihr sagen, alles schön und gut, und wozu soll das gut sein ? Ganz einfach, falls im Programmlauf ein Fehler auftritt und die Variable M, sagen wir, einen Wert von 25 zugewiesen bekommt, hält das Programm mit einem Laufzeitfehler. Damit kann man Fehler entdecken, die einem sonst durch die Lappen gehen, und man wundert sich. Übrigens nennt man den oben definierten Typ Unterbereichstyp oder Subrangetyp. Darüber hinaus gibt es noch den sog. Aufzähltyp. Wieder ein Beispiel :

```
...
TYPE MONAT =
(JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC);
VAR M : MONAT;
...
```

Man kann dann tatsächlich schreiben : IF M = OCT THEN WRITE('OKTOBER');

An diesem Punkt kommen wir wieder zur Diskussion der Ordnungsrelationen zurück. Jeder Wert eines Aufzähltyps entspricht der Stelle, an der er steht, z.B. ist der Wert vom MAR intern 2. Deshalb ist MAR kleiner als MAY, der durch 4 repräsentiert wird (der erste Wert ist Null). Wenn man sich durch einen solchen Aufzähltyp bewegen will, kann man natürlich nicht schreiben

```
M := M + 1;
```

sondern man muß die Funktionen PRED und ORD benutzen. Hoffentlich wird der Zusammenhang klarer, wenn ihr euch das zweite Beispielprogramm anschaut.
Zum Experimentieren lädt euch herzlich ein,

Rüdiger

```
program Skalar(input, output) ;

(* Dieses Programm fordert den Benutzer auf, einige Zahlen
   einzugeben und berechnet daraus das Skalarprodukt.      *)

(* Dargestellt werden :

                FOR - Schleife
                ARRAY - Typ
                READ/WRITE
                Kommentare
*)

CONST
   upper_bound = 3;

TYPE
   Vektor = array [ 1..upper_bound ] of real;

VAR
   a,b : vektor;
   i : integer;
   Skalar_Produkt : Real;

begin (* Hauptprogramm *)
   writeln;                                         (* Leerzeile ausgeben *)
   writeln('Gib dreimal zwei Zahlen ein.');        (* Prompt             *)
   writeln('Mehrere Zahlen in einer Zeile werden');
   writeln('durch LEERZEICHEN getrennt !');

   for i := 1 to upper_bound do                     (* FOR - Konstrukt    *)
      readln(a[i], b[i]);                           (* 2 Parameter,       *)
                                                    (* dann CR/LF         *)
   Skalar_Produkt := 0;                             (* Initialisieren !   *)
   for i := 1 to upper_bound do
      skalar_produkt := skalar_produkt + a[i] * b[i]; (* Rechnung *)

   writeln;                                          (* Leerzeile          *)
   writeln(' Das Skalarprodukt ist : ',Skalar_Produkt);
                                       (* beachte die 2 Parameter ! *)
   readln;                             (* warte auf RETURN *)
end. (* Fertig *)
```

```
program monate;

(* Dieses Programm demonstriert Unterbereichs- und
   Aufzaehltypen.
*)

TYPE
   MONAT = (JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC);

VAR
   M : MONAT;
   I,Z : INTEGER;

begin (* Hauptprogramm *)
   repeat                                     (* primitive Eingabepruefung *)
      ClrScr;                                 (* Bildschirm loeschen *)
      Write('Gib eine ganze positive Zahl bis 100 ein : ');
      ReadLn(i);
   until (i > 0) and ( i<= 100);
   z := 1; m := JAN;                          (* Monat auf Januar setzen *)
   while z <= i do                            (* Abbruch wenn z > i      *)
   begin
      writeln('Monat : ' ,ord(m)+1);          (* Wert von M ausgeben     *)
      m := succ(m);                           (* naechster Monat         *)
      z := z + 1;                             (* Zaehler erhoehen        *)
      if ord(m) > 12 then m := JAN;           (* von vorne anfangen      *)
   end;                                       (* while                   *)
end.                                          (* Program                 *)
```

RPNL ! . . . RPNL ? . . . ???

"Wer kennt sie nicht, Die Vierte - FORTH. Eine feine Sache, aber mal ehrlich, irgendwie ist das doch wie eine große Kiste randvoll gefüllt mit Lego-Steinen. Solange man nichts sucht ist alles OK, aber wenn ..."

Tja, an dieser Stelle verlassen wir besser das Geschehen und blenden um zu RPNL. Als das letzte Mal etwas zu diesem Thema gesagt wurde, schrieb man das Jahr 1986. Mittlerweile sind mehr als 12 Monate vergangen. Ein Lebenszeichen ist also angebracht. Mir ist nicht bekannt ob irgend jemand auf die Idee gekommen ist, den Compiler von unserem Diskothekar abzufordern, dort liegt er nämlich für Interessierte bereit. Meine Absicht ist es, die von mir eingebauten Arithmetik-Routinen vorzustellen.

Weil es mit den vier Grundrechenarten allein nicht getan ist, sind zur Unterstützung noch einige zusätzliche Dinge von Nöten. Vorweg deshalb die Liste mit den Befehlen und deren Bedeutung:

$PEEK	(->)	Unterprogramm: DE := (HL)
$POKE	(->)	Unterprogramm: (HL) := DE
:=(D)	(d1 adr ->)	Lädt 32Bit Datum nach angegebene Adresse
?(D)	(adr -> d1)	Holt 32Bit Datum von angegebener Adresse
DOUBLE	(n1 -> d1)	Wandelt einfachgenaue Zahl in eine doppeltgenaue Zahl um (vorzeichenrichtig)
DSIGN	(d1 d2 -> sign(d2)*d1)	Überträgt das Vorzeichen von d2 auf d1.
0<	(n1 -> flag)	Prüft n1 ob kleiner Null
PRINTD	(adr ->)	Gibt den Inhalt einer doppeltgenauen Variablen auf dem Monitor aus
BINASC	(d1 -> 0 a1 .. an cnt)	Wandelt d1 in einen ASCII-String um. 'cnt' ist die Anzahl der erzeugten Ziffern ohne Minuszeichen !
D+	(d1 d2 -> d3)	Addiert zwei doppeltgenaue Zahlen
D-	(d1 d2 -> d3)	Subtrahiert zwei doppelt genaue Zahlen
M*	(d1 n1 -> d2)	Multipliziert eine doppelt genaue Zahl mit einer einfachgenauen Zahl
MDIV	(d1 n1 -> d2)	Dividiert eine doppelt genaue durch eine einfach genaue Zahl

Ein weiterer Bestandteil der Arithmetik-Routinen sind die Variablen 'BASE' und 'DPL'. BASE hält die aktuelle Basis in der alle Zahlenausgaben erfolgen sollen. DPL bestimmt, an welcher Stelle innerhalb des Ziffernstrings der Dezimalpunkt eingesetzt wird. Die Zählung erfolgt von der niederwertigsten Ziffer ab in Rich-

tung der höchstwertigsten. Ein Beispiel:

```
DPL := 3     N := 123456789
```

Die Ausgabe erfolgt in der Form

```
123456.789
```

Nun aber zur Sache. Beginnen wir mit den Wort-Definitionen '$PEEK' und '$POKE'. Das vorangestellte Dollarzeichen soll ein Unterprogramm kennzeichnen, das mit einem Z80 CALL-Befehl aufgerufen werden kann. Ich erinnere in diesem Zusammenhang an meine Ausführungen im Club-Info No. 15, S. 57...64 zu diesem Thema. Den Abschluß bildet deshalb auch ein ganz ordinärer RET-Befehl. Zweck dieser beiden Worte ist es, die Übernahme von Daten aus Variablen, die mit DECLARE vereinbart wurden, zu vereinfachen und zu vereinheitlichen. Auf High-Level-Ebene ist die vorgehensweise folgendermaßen:

```
              :
              :
    <variablen-name> ?
    <variablen-name> :=
              :
              :
```

Nicht so jedoch, wenn die gleich Aktion in Assembler erfolgen soll. RPNL unterstützt auf dieser Ebene eigentlich nur das Einbinden von Adressen auf brauchbare Weise. Alles Andere muß vom Anwender kommen. Soll nun aus einer Variablen etwas gelesen werden, muß man wissen wie diese angelegt ist. Die Definition mit 'VAR' erzeugt zunächst einmal eine Adresse, die auf das Wort <variable> im Runtime-Modul zeigt. Das ist notwendig, da ebend dieses Wort den Zeiger generiert und auf den Stack legt, mit dessen Hilfe dann '?' auf den Inhalt der Variablen zugreift. Zur Verdeutlichung ein Beispiel:

```
          :                         :
          :                         :
  <variable-name>        +--->  <variable>
  <variable-adr> -------+       <  data  >
          :                         :
          :                         :
```

Das dürfte verdeutlichen was gemeint ist, wenn ich in Assembler schreibe: LD HL,<variablen-name>. Der Compiler sucht den Nameneintrag in seiner Liste. Dort findet er die Adresse und setzt diese in die Assembleranweisung ein. Wie das Beispiel zeigt, ist das aber nicht der Zeiger auf die Daten. Den erhält man erst nach zweimaligem Inkrementieren von HL. Entsprechendes gilt fuer Konstanten, deren Aufbau identisch ist. Den Rest erledigt nun '$PEEK' oder '$POKE' für den Anwender. Beide Unterprogramme erwarten im HL-Register den Zeiger auf das Datenwort bzw. den zu beschreibenden Speicherplatz. Das Register DE enthält das geholte bzw. das abzuspeichernde Datenwort.


```
DECLARE
    VAR BASE                ;hält die aktuell Zahlenbasis
    VAR DPL                 ;hält die Lage des Dezimalpunktes
    ARRAY NUMBER 2          ;die 32Bit Variable
DEND

10 BASE :=

CALL $PEEK
    5E          ;$PEEK: LD      E,(HL)
    23          ;       INC     HL
    56          ;       LD      D,(HL)
    23          ;       INC     HL
    C9          ;       RET
CEND

CALL $POKE
    73          ;$POKE: LD      (HL),E
    23          ;       INC     HL
    72          ;       LD      (HL),D
    23          ;       INC     HL
    C9          ;       RET
CEND
```

Die Belegung der Register ist so getroffen, daß ein einheitliches Parameterhandling auf Assemblerebene zustande kommt. Zu den nächsten beiden Worten gibt es nicht viel zu sagen, außer daß sie sich ausgiebig der bereits erwähnten Worte bedienen und so gleichzeitig als Anwendungsbeispiel dienen können.

```
CODE :=(D)          ;(d1 adr --     )
    E1              ;PEEKD: POP     HL
    CD $PEEK        ;       CALL    $PEEK
    D5              ;       PUSH    DE
    CD $PEEK        ;       CALL    $PEEK
    D5              ;       PUSH    DE
CEND

CODE ?(D)           ;(adr  -> d1    )
    E1              ;PEEKD: POP     HL      ;ADR
    C1              ;       POP     BC      ;DATA-H
    D1              ;       POP     DE      ;DATA-L
    CD $POKE        ;       CALL    $POKE
    50              ;       LD      D,B
    59              ;       LD      E,C
    CD $POKE        ;       CALL    $POKE
CEND
```

Als nächstes folgen weitere nützliche Dinge. Hier gibt es ebenfalls nicht viel zu erklären, alle Einzelheiten sind leicht zu erkennen. Wer eine Funktion zur Absolutwertbildung vermist, dem kann geholfen werden:

```
PROGRAM DABS            ;(d1     -> abs(d1)  )
        DUP DUP
        DSIGN
END
```

Ein Mangel den RPNL bisher auszeichnete, war das Fehlen des Wortes '0<'. Die Abfrage einer Zahl auf < 0 war dadurch nur sehr umständlich möglich. Die Definition sei hier gleich mitgeliefert.

```
CODE DOUBLE        ;(n1     -> d1    )
    E1             ;DOUBLE:  POP      HL
    E5             ;         PUSH     HL
    11 00 00       ;         LD       DE,0000
    CB 7C          ;         BIT      7,H
    28 01          ;         JR       Z,DBL
    1B             ;         DEC      DE
    D5             ;DBL:     PUSH     DE
CEND

CODE DSIGN         ;(d1 d2 -> sign(d2) * d1 )
    C1             ;DSIGN:  POP      BC
    CB 78          ;        BIT      7,B
    C1             ;        POP      BC
    28 0F          ;        JR       Z,DSIG
    D1             ;        POP      DE
    C1             ;        POP      BC
    21 00 00       ;        LD       HL,0000
    AF             ;        XOR      A
    ED 42          ;        SBC      HL,BC
    E5             ;        PUSH     HL
    21 00 00       ;        LD       HL,0000
    ED 52          ;        SBC      HL,DE
    E5             ;        PUSH     HL
CEND

CODE 0<                ;(n1 -> flag)
    E1                 ;ZLT:    POP      HL
    11 00 00           ;        LD       DE,0000
    CB 7C              ;        BIT      7,H
    28 01              ;        JR       Z,ZLT1
    1B                 ;        DEC      DE
    D5                 ;ZLT1:   PUSH     DE
CEND
```

Wie schon weiter oben gesagt, ist eine Nutzung von Systemresourcen auf Assemblerebene bisher kaum möglich. Aus diesem Grund sind die Divisions- und Multiplikations-Routinen als Unterprogramme angelegt. Es kann so beim Entwurf anderer Programme ohne Probleme darauf zugegriffen werden. Ausgenommen hiervon sind Addition und Subtraktion, da diese unkompliziert sind und sich mit den bestehenden Z80-Befehlen schnell erstellen lassen.

Nun aber ein paar Worte zu '$DIV' und '$MULT'. Zuerst zu $DIV. Damit der Algorithmus bei gesetztem Bit 15 im Divisor nicht aussteigt und Müll abliefert, ist eine Erweiterung des Dividenden auf 33 Bit unumgänglich. Hierfür wird der zweite Akkumulator AF' benutzt. Durch diesen Trick kann der volle Wertebereich von 0...65535 für den Divisor zugänglich gamacht werden. Eine vorzeichenbehaftete Division ist aber trotzdem nur bei vorheriger Auswertung der Vorzeichen und entsprechender Korrektur des Ergebnisses möglich.

Die Rechnung erfolgt nach althergebrachter Schulmanier: Divisor subtrahieren, wenn Zwischenergebnis positiv dann weiter, sonst Rückaddition, den Dividenden eine Stelle nach Links schieben und wiederholen. Nach 32 Runden steht dann das Ergebnis fest. Während der Rechnung besteht folgende Registerkonfiguration:

```
        <A'HLDE(SP)F'>
    -     <BC>
        ------------
            <DE(SP)>       := Ergebnis
          <HL>             := Rest
```

Rückspeichern der Register:

```
        <BC> <-- <HL>
        <HL> <-- (SP)

Ergebnis:   <DEHL>
REST    :     <BC>
```

Der Divisor geht verloren.

Die Anordnung der Register ist so dargestellt, wie sie vom Algorithmus benutzt werden. Mangels genügend 16bit-Register werden die niederwertigen 16 Bit des Ergebnisses während der Rechnung auf dem Stack gehalten

```
CALL $DIV          ;<DEHL>,R<BC> := <DEHL> / <BC>
    E5             ;$DIV:   PUSH     HL
    AF             ;DIV1:   XOR      A
    08             ;        EX       AF',AF
    AF             ;        XOR      A
    67             ;        LD       H,A
    6F             ;        LD       L,A
    3E 20          ;        LD       A,32     ;32 Runden sollens sein
    E3             ;DLOOP:  EX       (SP),HL  ;4. Register holen und
    08             ;        EX       AF',AF
    ED 6A          ;        ADC      HL,HL    ;nach links schieben
    E3             ;        EX       (SP),HL
    CB 13          ;        RL       E
    CB 12          ;        RL       D
    ED 6A          ;        ADC      HL,HL
    CE 00          ;        ADC      A,00
    ED 42          ;        SBC      HL,BC
    DE 00          ;        SBC      A,00
    30 03          ;        JR       NC,DIFOK
```

```
    09          ;           ADD     HL,BC
    CE 00       ;           ADC     A,00
    3F          ;DIFOK:     CCF
    08          ;           EX      AF',AF
    3D          ;           DEC     A
    20 E5       ;           JR      NZ,DLOOP
    44          ;DDONE:     LD      B,H
    4D          ;           LD      C,L
    E1          ;           POP     HL
    08          ;           EX      AF',AF
    ED 6A       ;           ADC     HL,HL    ;Ergebnis nachbehandeln
    CB 13       ;           RL      E
    CB 12       ;           RL      D
    C9          ;           RET
CEND
```

Zwei Zahlen zu multiplizieren ist weit unkomplizierter als die Durchführung einer Division. Es werden einfach weniger arithmetische Funktionen benötigt: Addition mit und ohne Carry genügen. Dieser Umstand hilft die doch recht zeitfressende Operation mit dem 4. Register auf dem Stack, zu umgehen.

Die belegung der Register während der Rechnung:

```
      <DEHLIX>
+        <BC>
```

Das Endergebnis ergibt sich aus den Inhalt von IX (Low Part) und dem Inhalt von HL (High Part). Damit die Schnittstelle am Ende wieder stimmt, sind noch einige Registerinhalte umzuspeichern:

```
DE <-- HL
HL <-- IX
```

Der Multiplikator in <BC> bleibt unverändert erhalten. Das ist besonders bei Kettenrechnungen etc. von Vorteil. Der alte Inhalt von IX geht nicht verloren.

```
CODE $MULT          ;<DEHL> := <DEHL> * <BC>
    3E 20           ;MULT:      LD      A,32
    DD E5           ;           PUSH    IX
    DD 21 00 00     ;           LD      IX,0000
    DD 29           ;MLOOP:     ADD     IX,IX
    ED 6A           ;           ADC     HL,HL
    CB 13           ;           RL      E
    CB 12           ;           RL      D
    30 05           ;           JR      NC,NOSUM
    DD 09           ;           ADD     IX,BC
    30 01           ;           JR      NC,NOSUM
    23              ;           INC     HL
    3D              ;NOSUM:     DEC     A
    20 EE           ;           JR      NZ,MLOOP
    DD E5           ;MDONE:     PUSH    IX
```



```
    D1          ;           POP     DE
    EB          ;           EX      DE,HL
    DD E1       ;           POP     IX
CEND

CODE D+         ;(d1 d2 -> d3 )
    C1          ;DADD:      POP     BC
    D1          ;           POP     DE
    E1          ;           POP     HL
    E3          ;           EX      (SP),HL
    19          ;           ADD     HL,DE
    E3          ;           EX      (SP),HL
    ED 4A       ;           ADC     HL,BC
    E5          ;           PUSH    HL
CEND

CODE D-         ;(d1 d2 -> d3 )
    C1          ;DSUB:      POP     BC
    D1          ;           POP     DE
    E1          ;           POP     HL
    E3          ;           EX      (SP),HL
    AF          ;           XOR     A
    ED 52       ;           SBC     HL,DE
    E3          ;           EX      (SP),HL
    ED 42       ;           SBC     HL,BC
    E5          ;           PUSH    HL
CEND

CODE M*         ;(d1 n2 -> d3 )
    C1          ;MULTI:     POP     BC      ;Multiplikator
    D1          ;           POP     DE      ;Multiplikant HIGH
    E1          ;           POP     HL      ;Multiplikant LOW
    CD $MULT    ;           CALL    $MULT   ;Produziere große Zahl
    E5          ;           PUSH    HL      ;Produkt LOW
    D5          ;           PUSH    DE      ;Produkt HIGH
CEND

CODE MDIV       ;(d1 n1 -> d2 )
    C1          ;MDIV:      POP     BC      ;Devisor
    D1          ;           POP     DE      ;Dividend HIGH
    E1          ;           POP     HL      ;Dividend LOW
    CD $DIV     ;           CALL    $DIV
    E5          ;           PUSH    HL      ;Quotient LOW
    D5          ;           PUSH    DE      ;Quotient HIGH
CEND
```


Wenn schon mit 9-stelligen Zahlen gerechnet werden kann, wäre es schön, sie auch auf dem Bildschirm ausgeben zu können. Einen entsprechenden Ausgabebefehl gibt es als nächstes. Er ist zweiteilig, um eine höhere Flexibilität in der Anwendung zu erhalten. Die Binärzahl wird mit dem Wort 'BINASC' in einen ASCII-String umgewandelt und Ziffer für Ziffer auf dem Stack abgelegt. Die Wandlung erfolgt unter Berücksichtigung der Systemvariablen 'BASE' und 'DPL'. Das ist besonders angenehm bei

der Ausgabe von DMarks. Bei DPL = 0 fehlt der Dezimalpunkt ganz, bei DPL <> 0 erscheint er links von der angegebenen Stelle. Damit eventuelle Sonderformate (Uhrzeit etc.) einfacher erzeugt werden können, liegt zuoberst auf de Stack noch ein Ziffern-Count, der angiebt, wieviele Ziffern der gesamte String lang ist (inkl. eventuellem Dezimalpunkt !). Das Vorzeichen der Zahl wird an dieser Stelle noch nicht eingesetzt (BINASC kennt nur positive Zahlen !!!). In der Regel wird dieser Zähler nicht benötigt und deshalb vor einsetzen des Vorzeichens vom Stack entfernt. Das Stringende ist erreicht, wenn die '0' hinter dem letzten ASCII-Zeichen erkannt wird.

```
CODE BINASC         ;(d1 -> 0 a1 a2 ... an cnt)
    D1              ;CONV:   POP    DE
    E1              ;        POP    HL
    DD 21 BASE      ;CONV1:  LD     IX,BASE
    DD 36 03 00     ;        LD     (IX+03),00
    FD 21 DPL       ;        LD     IY,DPL
    FD 7E 02        ;        LD     A,(IY+02)
    FD 77 03        ;        LD     (IY+03),A
    01 00 00        ;        LD     BC,0000
    C5              ;        PUSH   BC
    06 00           ;ALOOP:  LD     B,00
    DD 4E 02        ;        LD     C,(IX+02)
    CD $DIV         ;        CALL   $DIV
    79              ;        LD     A,C
    FE 0A           ;        CP     10            ;A > 9 ?
    38 02           ;        JR     C,LT10
    C6 07           ;        ADD    A,'A'-'9'-1
    C6 30           ;LT10:   ADD    A,'0'
    4F              ;        LD     C,A
    06 00           ;        LD     B,00
    C5              ;        PUSH   BC
    FD 7E 03        ;QPNT:   LD     A,(IY+03)
    3D              ;        DEC    A
    FD 77 03        ;        LD     (IY+03),A
    B7              ;        OR     A
    20 03           ;        JR     NZ,NOPNT
    0E 2E           ;        LD     C,'.'
    C5              ;        PUSH   BC
    DD 34 03        ;NOPNT:  INC    (IX+03)       ;STELLEN INCR.
    7C              ;        LD     A,H
    B5              ;        OR     L
    B2              ;        SR     D
    B3              ;        OR     E
    20 D5           ;        JR     NZ,ALOOP
    FD 7E 03        ;        LD     A,(IY+03)
    CB 17           ;        RL     A
    30 CE           ;        JR     NC,ALOOP
    06 00           ;        LD     B,00
    DD 4E 03        ;        LD     C,(IX+03)
    C5              ;        PUSH   BC
    DD 36 03 00     ;        LD     (IX+03),00
    FD 36 03 00     ;        LD     (IY+03),00
CEND
```

Die vorzeichenrichtige Behandlung einer Zahl wird von 'PRINTD' erledigt. Auf dem Stack wird genau wie bei 'PRINT' die Adresse einer Variablen auf den Speicherplatz der Zahl erwartet, deren Inhalt auf den Stack geholt und das Vorzeichen festgehalten. Mit 'DSIGN' erfolgt dann eine Absolutwertbildung und anschließender Wandlung duch 'BINASC'. Als letztes wird geprüft, ob ein Minuszeichen ('-') eingefügt werden muß. Damit kann die Zahl endlich ausgegeben werden.


```
PROGRAM PRINTD                  ;(adr ->     )
    ?(D) DUP >R
    OVER OVER DSIGN
    BINASC DROP                 ;Stellenzähler 'cnt' vernichten
    R> 0<
    IF 45 ENDIF                 ;'-' := 45 !
    32                          ;in jedem Fall ein führendes Blank
    REPEAT
        DUP 0=
    UNTIL
        OUTCHAR
    LOOP
    BASE ? 16 =
    IF
        WRITE 'H'
    ENDIF
    DROP
END
```

Mit '>R' und 'R>' ist es ermöglichen, Daten vom Parameter-Stack auf den Return-Stack und wieder zurück zu transportieren. Werte, die momentan stören, können so aus dem Weg geschaft werden, ohne daß extra eine Variable dafür eingerichtet werden muß. Die Benutzung beider Worte unterliegt jedoch Regeln, deren Nichtbeachtung garantiert zum System- und Programmabsturz führen:

1. Daten, die auf den Return-Stack ausgelagert werden, müssen in jedem Fall innerhalb einer PROGRAM-Definition von dort wieder zurückgeholt werden.

2. Die FOR...LOOP Konstruction macht ebenfalls gebrauch vom Return-Stack und legt dort die Schleifenparameter ab. Eine Überschneidung von mit '>R', 'R>' ausgelagerten Daten und den Schleifenparametern ist deshalb zu vermeiden.

'R>' und '>R' können leider nicht auf User-Ebene definiert werden, da die Startadresse des Return-Stack im Dicenary nicht verzeichnet ist. Hier kommt nur eine Definition im Assembler-Programm des Compilers in Frage. Als Behelfslösung ist aber folgendes brauchbar :

```
PROGRAM >R
    ZEICHEN :=
END

PROGRAM R>
    ZEICHEN ?
END
```

'ZEICHEN' ist eine Systemvariable, in die mit 'INCHAR' eingelesene Zeichen abgespeichert werden ! Sollte ZEICHEN nicht mehr frei sein, hilft nur eine eigene Variable.

Damit bin ich mit der Beschreibung des Arithmetik-Packetes am Ende und es kann an's experimentieren gehen.

Tschüß

Vergleich zweier Disk-Dateien

Normalerweise hat man keine zwei gleichen Programme mit unterschiedlichen Namen auf einer Scheibe. Doppelt genäht kostet Diskettenplatz. Es gibt jedoch Fälle, in denen das durchaus so ist. Wenn man beispielsweise nicht mehr so recht weiß, ob TS/CMD das gute alte TSCRIPS/CMD ist oder eine inzwischen erweiterte Version. Vielleicht sind sie ja identisch, so daß ein Programm gelöscht werden kann.

Ein anderer Fall ist mir heute begegnet und veranlaßte mich, die hier vorgestellte Routine zu schreiben. Die Tüftler unter euch werden gelegentlich vor demselben Problem stehen. Ich war dabei, MEMDISK/CMD zu analysieren, um es gewissen Hardware-Änderungen an meinem Computer anpassen zu können. Da wurden weite Teile erst mal in irgendwelche Banks an verschwiegene Adressen geschmuggelt, um später dort abgerufen werden zu können. Der Übersichtlichkeit halber war es sinnvoll, den Offset der jeweiligen Lade- zur Arbeitsadresse eines Programmsegments in die Labels der Sprungziele einzubeziehen.

Wer schon mit Offsets gearbeitet hat, wird darob so manchen Fluch getan haben. Oft vergißt man, dem Label den Zusatz "+offset" anzuhängen. Die Folgen sind in der Regel katastrophal. Es mußte daher eine bequeme Möglichkeit gefunden werden, festzustellen, ob die neu erzeugte Source noch gewisse Ähnlichkeiten mit dem untersuchten Programm aufweist.

Dazu wird sie assembliert und der neue Objektcode mit Hilfe dieser Utility mit dem alten verglichen. Verschiedene Assembler erzeugen u. U. verschieden lange Records. Die Zählbytes im Header unterscheiden sich dann, was die Utility als Abweichung anzeigen würde. Deshalb ist es im Zweifel ratsam, die noch unveränderte Source, wie sie beispielsweise DSMBLR/CMD vom Target-Programm abliefert, sofort wieder zu assemblieren.

Wenn die Source gestreamlined ist, wenn also beispielsweise Offsets berücksichtigt sind, und wenn sie unter anderem Namen assembliert wurde, geht's nun ab:

Die Utility wird wie gewohnt mit ihrem Namen aufgerufen, gefolgt von den beiden Namen der zu vergleichenden Files. Sie können alle Zusätze enthalten, die das DOS so für Filespecifications zuläßt. Die drei Dateinamen (die beiden letzten gegf. mit Extension!) werden wie üblich durch Blank oder Komma voneinander getrennt.

Das Programm fragt beim Einsprung zunächst, ob die Ausgabe der Abweichungen auf den Drucker erfolgen soll. Wenn man ziemlich sicher ist, daß nur wenige Bytes unterschiedlich sein dürften, genügt der Bildschirm. Wer sich selber nicht über den Weg traut, sollte die Abweichungen lieber ausdrucken lassen; es könnte eine längere Latte werden.

Für beide Programme werden sodann ein FCB eröffnet und ein Sektorpuffer reserviert. Über die zuständigen DOS-Routinen wird von beiden Programmen ein Sektor nach dem anderen geladen und verglichen. Sobald eine Abweichung auftritt, wird sie angezeigt:
rel. Skt. / rel. Byte : Abweichung ====> 0004/B7:A9C6
Diese Zahlen sind willkürlich gewählt und dienen nur als Beispiel: Im filerelativen Sektor 4 unterscheidet sich das relative Byte B7h in beiden Dateien. Das Byte B8h wird ebenfalls angezeigt, denn oft – zumindest im oben skizzierten Anwendungsfall – sind es 16-Bit-Adressen, bei denen man nicht aufgepaßt hat.

Die abweichenden Bytes werden nur aus einem File angegeben. Je nach Reihenfolge der Eingabe der Programmnamen können es die richtigen oder die falschen Bytes sein. Zur Anzeige kommen diejenigen aus dem im Befehlsstring zuerst benannten Programm.

Diese Detektiv-Utility dient einem weiteren gemeinnützigen Zweck. DSMBLR zerstückelt ASCII-Strings, Folgen von DEFBs und DEFWs usw., weil seine Zeilenlänge begrenzt ist. Da das Auge mithackt, wird man die Zeilenaufteilung in aller Regel neu ordnen. Dabei geht leicht einmal ein Blank oder ein unscheinbares anderes Byte über'n Jordan. Danach verschiebt sich der ganze Rest entsprechend nach vorne. Fährt man nun das Vergleichsprogramm, ergibt sich eine verdächtig hohe Anzahl von Abweichungen an fortlaufenden Sektoradressen. Am Beginn dieser Katastrophe kann man nun nach dem verschwundenen Byte fahnden. Wenn es zufällig im letzten Sektor des Files der Fall war, wird auch das kleinere Zählbyte im Record-Header als Abweichung angezeigt.

Auf die Programmlogik möchte ich hier nicht weiter eingehen. Sie erklärt sich aus den üppigen Kommentaren. Was so an den aufgerufenen DOS-Routinen passiert, kann Hartmut Grosser in seinem DOS-Buch ohnehin besser erklären als ich.

Auf wackeres Debuggen!

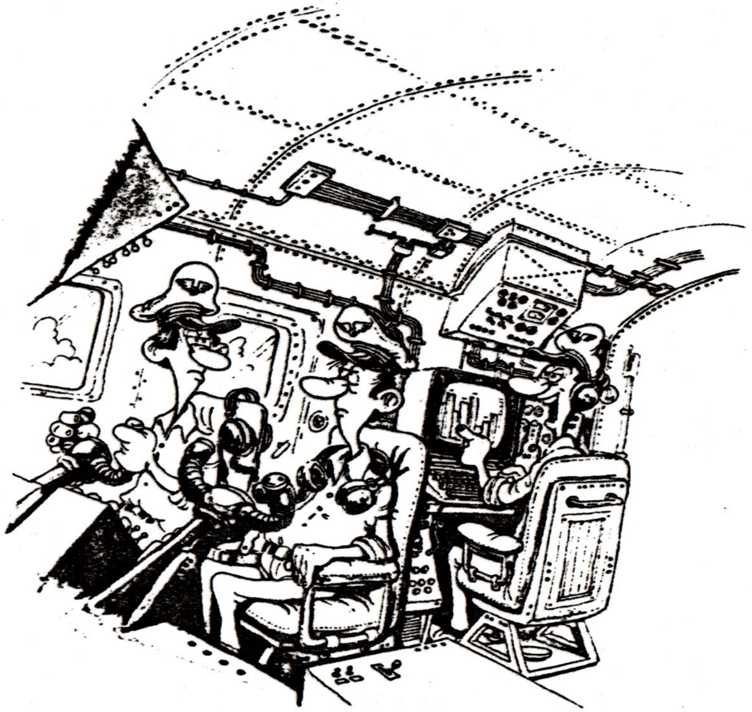
Arnulf Soppp

```
                ;        Utility zum Vergleich zweier Dateien
                ;
                ;        bei Abweichung werden zwei Bytes angezeigt,
                ;        weil es sich oft um eine fehlerhafte Adresse handelt
                ;
                ;        Copyright geschunken


5200                     ORG     5200h

                ;je nach Userwunsch Ausgabe der Abweichungen auf Bildschirm oder Drucker
5200  E5        start    PUSH    HL            ;Zeiger auf den Befehlsstring retten
5201  219C52             LD      HL,dvcreq     ;Text "Druckerausgabe? (J/N)"
5204  CD6744             CALL    4467h         ;anzeigen
5207  CD4900             CALL    0049h         ;Tastatur abfragen
520A  CD8545             CALL    45b5h         ;Eingabe in Großbuchstaben umwandeln
520D  FE4A               CP      'J'           ;Ausgabe auf den Drucker?
520F  2005               JR      NZ,open       ;falls nein
5211  3E6A               LD      A,6ah         ;Adreß-LSB der Druckerroutine
5213  328A52             LD      (device),A    ;Ausgabeadresse auf Drucker umleiten

                ;für beide Files einen FCB eröffnen -
                ;Fehler, wenn im Aufrufbefehl nicht die Namen
                ;zweier existierender Dateien enthalten sind
5216  E1        open     POP     HL            ;Zeiger auf den Befehlsstring
5217  CDD54C             CALL    4cd5h         ;Trennzeichen überspringen
521A  11EA52             LD      DE,fcb1       ;FCB für das erstgenannte File
521D  CD1C44             CALL    441ch         ;den Dateinamen in den FCB übertragen
5220  3E2F      error1   LD      A,2fh         ;Fehlercode "schlechte Parameter"
5222  C20944    error2   JP      NZ,4409h      ;falls kein gültiger Filename (Fehler)
5225  CDD54C             CALL    4cd5h         ;nächstes Trennzeichen überspringen
5228  110A53             LD      DE,fcb2       ;FCB für das zweite Programm
522B  CD1C44             CALL    441ch         ;Programmnamen in den FCB übertragen
522E  20F0               JR      NZ,error1     ;falls kein gültiger Filename
5230  210055             LD      HL,buffer2    ;Sektorpuffer für das zweite Programm
5233  45                 LD      B,L           ;logische Recordlänge = 256 Bytes (L=00)
5234  CD2444             CALL    4424h         ;FCB für das zweite Programm eröffnen
5237  20E9               JR      NZ,error2     ;falls ein Fehler aufgetreten ist
5239  11EA52             LD      DE,fcb1       ;FCB für das erste Programm
523C  210054             LD      HL,buffer1    ;dessen Sektorpuffer
523F  CD2444             CALL    4424h         ;diesen FCB ebenfalls eröffnen
```

```
5242  20DE     00042           JR      NZ,error2          ;falls ein Fehler aufgetreten ist
               00043
               00044 ;Sektoren beider Dateien einlesen
               00045 ;Programmende, falls die Dateien vollständig untersucht sind
5244  48       00046           LD      C,B                ;BC <- 0000 (ab Sektor 0 zählen)
5245  CD3644   00047 loadlop CALL    4436h              ;einen Sektor von Datei 1 einlesen
5248  F5       00048           PUSH    AF                 ;Fehlercode und Z/NZ-Flag retten
5249  FE1C     00049           CP      1ch                ;Fehlercode 'Ende der Datei angetroffen'?
524B  CA2D40   00050 dosexit JP      Z,402dh            ;zurück nach DOS-Ready, falls erledigt
524E  FE1D     00051           CP      1dh                ;Fehlercode 'hinter Ende der Datei'?
5250  28F9     00052           JR      Z,dosexit          ;dann auch fertig
5252  F1       00053           POP     AF                 ;anderer Fehlercode mit Fehlerflag
5253  20CD     00054           JR      NZ,error2          ;evtl. anderen Fehler anzeigen und raus
5255  110A53   00055           LD      DE,fcb2            ;Zeiger auf den FCB der Datei 2
5258  210055   00056           LD      HL,buffer2         ;Zeiger auf deren Sektorpuffer
525B  CD3644   00057           CALL    4436h              ;einen Sektor von Datei 2 einlesen
525E  20C2     00058           JR      NZ,error2          ;falls ein Fehler aufgetreten ist
               00059
               00060 ;je einen Sektor beider Dateien vergleichen
5260  110054   00061           LD      DE,buffer1         ;Sektorpuffer des 1. Programms
5263  1A       00062 complop LD      A,(DE)             ;1 Byte des Programms 1 laden
5264  BE       00063           CP      (HL)               ;dasselbe wie in Programm 2?
5265  2828     00064           JR      Z,match            ;falls Übereinstimmung
               00065
               00066 ;es wurde ein abweichendes Byte angetroffen: anzeigen
5267  E5       00067           PUSH    HL                 ;Zeiger retten
5268  210D52   00068           LD      HL,numbuff         ;Puffer für Hex-Daten
526B  D5       00069           PUSH    DE                 ;Zeiger retten
526C  50       00070           LD      D,B                ;DE <- Sektornummer
526D  59       00071           LD      E,C
526E  CD6340   00072           CALL    4063h              ;DE in Hex in den Puffer schreiben
5271  23       00073           INC     HL                 ;Trennzeichen '/' aussparen
5272  D1       00074           POP     DE                 ;Pufferzeiger restaurieren
5273  7B       00075           LD      A,E                ;relatives Byte im Sektor
5274  CD6840   00076           CALL    4068h              ;in den Hexpuffer schreiben
5277  23       00077           INC     HL                 ;Trennzeichen ':' aussparen
5278  1A       00078           LD      A,(DE)             ;abweichendes Byte
5279  CD6840   00079           CALL    4068h              ;in den Puffer
527C  1C       00080           INC     E                  ;Zeiger erhöhen - war es das Sektorende?
527D  2805     00081           JR      Z,onlyone          ;nur 1 Byte anzeigen, falls ja
527F  1A       00082           LD      A,(DE)             ;sonst nächstes Byte laden
5280  1C       00083           INC     E                  ;Zeiger weiterstellen
5281  CD6840   00084           CALL    4068h              ;Byte in Hex in den Puffer schreiben
5284  360D     00085 onlyone LD      (HL),0dh           ;Anzeigestring mit CR abschließen
5286  21B452   00086           LD      HL,texbuff         ;Anfang des kompletten Anzeigestrings
5289  CD6744   00087           CALL    4467h              ;Abweichung auf Video/Drucker ausgeben
528A           00088 device  EQU     $-2                ;Ausgabeadresse für Video oder Drucker
528C  E1       00089           POP     HL                 ;Zeiger restaurieren
528D  1D       00090           DEC     E                  ;wegen später INC E
528E  6B       00091           LD      L,E                ;beide Zeiger auf dieselbe Stelle setzen
               00092
               00093 ;gegf. nach Anzeige der Abweichung weiter prüfen
528F  1C       00094 match   INC     E                  ;beide Pufferzeiger erhöhen
5290  2C       00095           INC     L
5291  20D0     00096           JR      NZ,complop         ;falls noch nicht 256 Bytes bearbeitet
5293  11EA52   00097           LD      DE,fcb1            ;sonst FCB-Zeiger für Programm 1 laden
5296  210054   00098           LD      HL,buffer1         ;dto. Pufferzeiger
5299  03       00099           INC     BC                 ;Sektorzähler erhöhen
529A  18A9     00100           JR      loadlop            ;nächsten Sektor bearbeiten
               00101
               00102 ;versch. Texte, Puffer usw.
529C  44       00103 dvcreq  DM      'Druckerausgabe? (J/N) ',0eh,0dh         ;höfliche Frage
52B4  72       00104 texbuff DM      'rel. Skt. / rel. Byte : Abweichung ====> '      ;Text
52DD  78       00105 numbuff DM      'xxxx/yy:zzzz',0dh       ;Puffer für numerische Angaben
```



```
               00106
0020           00107 fcb1    DS      20h                ;32 Bytes Platz für den 1. FCB
0020           00108 fcb2    DS      20h                ;dto. für den 2. FCB
               00109
5400           00110 buffer1 EQU     $+0100h&0ff00h     ;beide Sektorpuffer an einer
5500           00111 buffer2 EQU     $+0200h&0ff00h     ;"glatten" Adresse hinter dem Programm
               00112
5200           00113         END     start

00000 Fehler

buffer1 5400   buffer2 5500   complop 5263   device  528A   dosexit 524B   dvcreq  529C
error1  5220   error2  5222   fcb1    52EA   fcb2    530A   loadlop 5245   match   528F
numbuff 520D   onlyone 5284   open    5216   start   5200   texbuff 52B4
```



Keine Ahnung, wie unsere Position ist, aber die Aktien werden demnächst steigen.

Vergleich zweier Programme: Noch'n Tool

In meinem Artikel "Vergleich zweier Disk-Dateien", vermutlich auch in dieser Nummer, wird eine Möglichkeit aufgezeigt, zu prüfen, ob man bei der Rekonstruktion eines Fremdprogramms einen Bock geschossen hat. Lt. Ed Murphy hat man auf jeden Fall. Bei der Weiterarbeit an meinem Problem (MEMDISK sollte geknackt werden) erwies es sich als sinnvoll, zunächst OVL4/SYS den Schneid abzukaufen. Das ist ein Overlay, das aus dem ziemlich universellen G-DOS 2.4 das Lieblings-DOS des Genie 3s macht. OVL4/SYS ist so chaotisch programmiert, daß einem die Tränen kommen. Also ein Einschub, bevor es konstruktiv weitergeht: OVL4/SYS erst mal auf Vordermann bringen.

In diesem File gibt es 47 (in Worten: siebenundvierzig!!!) ORG-Statements. Ihre Reihenfolge entspricht ungefähr der Ordnung auf meinem Schreibtisch. Eher schlimmer. Wer mich kennt, weiß, was das bedeutet! Also erst mal OVL4/SYS umbauen, damit ein mäßig begabter rothaariger Durchschnitts-Schleswig-Holsteiner halbwegs Durchblick bekommt. Gesagt, getan – geweint: Einen lausigeren Taschenrechner als mein G3s nach dieser Operation habe ich selten erlebt.

Irgendwo mußte ich mich bei den zahllosen Move-Anweisungen an ZEUS vertippt haben. Der Vergleich der Disk-Programme mußte versagen, denn OVL4/SYS war völlig umgebaut. Also konnte nur ein Vergleich beider *geladener* Objektfiles noch etwas bringen. Wie das? Die werkeln unten im Adreßbereich des Interpreters und des DOS. Die beiden würde ich mir durch meinen OVL4-Fehltritt ungern verhunzen lassen. Also mußten sie mit einem Versatz geladen werden, so daß das Betriebssystem nichts zu motzen hatte.

Die im Anschluß gelistete Utility bringt das. Sie nullt zunächst einen Puffer aus, von dem jedes der beiden zu vergleichenden Programme eine Hälfte belegen soll. Die Nullen sind wichtig, damit bei nicht zusammenhängenden ORG-Flicken in den Lücken gleiche Voraussetzungen herrschen. Anschließend wird für beide Programme in der gewohnten Weise ein FCB eingerichtet.

Nun muß das DOS zu wissen kriegen, daß es möglicherweise Harakiri begeht, wenn es beim Laden der Programme die im Record-Header vorgesehene Adresse glaubt. Sie muß um den Offset erhöht werden. Das zweite der beiden Programme muß doppelt so hoch gepuffert werden, denn es ist erforderlich, daß beide gleichzeitig im Speicher stehen. Ein Patch in LOADP (so nennt Hartmut Grosser in seinem DOS-Buch die Routine, die Disk-Programme im CMD-Format in den Speicher schlürft) sorgt für den Versatz. Der Rest ist Handwerk: Beide Puffer werden miteinander verglichen. Bei Nichtübereinstimmung eines Bytes werden die (tatsächliche, ohne Offset) Ladeadresse und dann das geheimnisvolle Byte angezeigt.

Gemessen am früher vorgestellten Vergleichsprogramm ist dieses hier sehr spartanisch. Der einzige Komfort sind die vier Labels am Beginn. Die Ladeadresse des Tools, der Offset zur Arbeitsadresse der zu untersuchenden Programme, ihre Pufferadresse und die Einsprungsadresse des Ausgabekanals (Bildschirm oder Drucker) können damit von Fall zu Fall den Erfordernissen angepaßt werden. Die beiden Namen der (hoffentlich – das walte Murphy!) eineiigen Programmzwillinge müssen ebenfalls jeweils geändert werden.

Da es sich um eine Utility für die Hand des Assembler-Cracks handelt, ist diese kleine Mühe weit weniger lästig als ein Programm für mich gewesen wäre, das jeder Schimpanse hätte bedienen können. Der Grund ist ebenso erfreulich wie simpel: Jedwedes Programm, wofern es nicht mehr als höchstens die Hälfte des ab 5200h verfügbaren Speichers belegt, kann damit überprüft werden.

Wo im Einzelfall resid liegt (die Arbeitsadresse des Tools), wo der Pufferanfang namens buffer steht, wie hoch der Offset ist (Label offset), das muß immer wieder neu ausgetüftelt werden. Aber die Kunde vom Addieren

und Subtrahieren soll sich einem Ondit zufolge in Clubkreisen schon herumgesprochen haben. Keine Hürde also.

Das Programm von neulich kann einfach abgefahren werden wie SUPERZAP/CMD. Dieses hier ist weniger gefügig, dafür aber in seinen Angaben aussagekräftiger. Man hat die Wahl.

Arnulf Sopp

```
5200                buffer  EQU   5200h            ;Anfang des Programmpuffers
5200                offset  EQU   5200h            ;Versatz zur tatsächlichen Ladeadresse
3000                resid   EQU   3000h            ;oder wo auch immer Platz sein mag
446A                display EQU   446ah            ;Anzeige durch den Drucker
                                                   ;(für den Bildschirm: 4467)

3000                        ORG   resid

3000  210052        start   LD    HL,buffer        ;Beginn des benutzten Puffers
3003  E5                    PUSH  HL               ;brauchen wir noch
3004  110152                LD    DE,buffer+1      ;nächste Stelle
3007  01FFAD                LD    BC,-buffer-1     ;Länge des Puffers
300A  -75                   LD    (HL),L           ;die 1. Stelle ausnullen (L = 00)
300B  EDB0                  LDIR                   ;den ganzen Puffer alle machen
300D  117830                LD    DE,fcb1          ;FCB für das 1. der zu vergleichenden Pr.
3010  D5                    PUSH  DE               ;brauchen wir auch noch
3011  210042                LD    HL,4200h         ;Sektorpuffer des DOS wird benutzt
3014  45                    LD    B,L              ;LRL = 256 (L = 00)
3015  CD2444                CALL  4424h            ;1. FCB eröffnen
3018  119F30                LD    DE,fcb2          ;FCB des 2. der zu vergleichenden Progr.
301B  D5                    PUSH  DE               ;wird auch noch gebraucht
301C  CD2444                CALL  4424h            ;den 2. FCB eröffnen
301F  216930                LD    HL,devia         ;Umleitung der LOADP-Routine des DOS
3022  224E4C                LD    (4c4eh),HL       ;Umleitung an geeigneter Stelle patchen
3025  E1                    POP   HL               ;Adresse des 2. FCB
3026  CD284C                CALL  4c28h            ;das 2. Programm adressenversetzt laden
3029  3EA4                  LD    A,offset/256*2   ;Offset für das 1. Programm (MSB genügt)
302B  326D30                LD    (base),A         ;in die Umleitung patchen
302E  E1                    POP   HL               ;Adresse des 1. FCB
302F  CD284C                CALL  4c28h            ;das 1. Programm adressenversetzt laden
3032  21654C                LD    HL,4c65h         ;ursprüngliche CALL-Adresse in LOADP
3035  224E4C                LD    (4c4eh),HL       ;restaurieren
3038  E1                    POP   HL               ;Anfang des Puffers
3039  110CA4                LD    DE,buffer*2      ;Anfang der oberen Hälfte des Puffers
303C  010052                LD    BC,buffer*2-offset     ;Länge der Prüfstrecke = 1 Hälfte
303F  1A            loop    LD    A,(DE)           ;ein Byte des 1. Programms laden
3040  BE                    CP    (HL)             ;mit dem passenden des 2. Pr. vergleichen
3041  C44C30                CALL  NZ,print         ;Fehler ausdrucken, falls ungleich
3044  23                    INC   HL               ;nächste Stelle unten
3045  13                    INC   DE               ;und oben im Puffer
3046  0B                    DEC   BC               ;Zähler erniedrigen
3047  78                    LD    A,B              ;prüfen, ob er schon
3048  B1                    OR    C                ;auf 0000 gesunken ist
3049  C8                    RET   Z                ;zurück ins DOS, falls ja
304A  18F3                  JR    loop             ;sonst weiter vergleichen

304C  E5            print   PUSH  HL               ;retten, was verändert wird
304D  D5                    PUSH  DE
304E  110052                LD    DE,offset        ;Offset zur tatsächlichen Ladeadresse
3051  B7                    OR    A                ;Cy löschen wegen SBC
3052  ED52                  SBC   HL,DE            ;tatsächliche Ladeadresse des Bytes
3054  EB                    EX    DE,HL            ;nach DE
```

```
3055 21eF30       LD     HL,numbuff      ;Puffer für numerische Werte in ASCII
305E E5           PUSH   HL              ;kühl und trocken aufbewahren
3059 F5           PUSH   AF              ;Akku wird auch verändert
305A CD6340       CALL   4063h           ;die Ladeadresse in den Puffer schreiben
305D 23           INC    HL              ;das Trennzeichen überspringen
305E F1           POP    AF              ;das nicht übereinstimmende Byte
305F CD6840       CALL   4068h           ;auch in den Puffer
3062 E1           POP    HL              ;Adresse des Pufferanfangs
3063 CD6A44       CALL   display         ;den Pufferinhalt ausdrucken
3066 D1           POP    DE              ;Register restaurieren
3067 E1           POP    HL
3068 C9           RET                    ;gelaufen

3069 CD654C devia CALL   4c65h           ;das 4. Byte des Record-Headers einlesen
                                         ;(= MSB der Ladeadresse)
306C C652         ADD    A,offset/256    ;um den Offset erhöhen (MSB langt)
306D        base  EQU    $-1             ;(der kann auch anders lauten)
306E C9           RET

306F 78   numbuff DM     'xxxx/yy',09h,03h       ;Puffer für die Anzeige

3075 4F   fcb1    DM     'OVL4/CMD',0dh  ;FCB für das Originalprogramm
001E              DS     30              ;reichlich Platz dafür
309F 4F   fcb2    DM     'OVL4/SYS',0dh  ;FCB für seinen Zwilling

3000              END    start           ;dort Entry

00000 Fehler

base    306D   buffer  5200   devia   3069   display 446A   fcb1   3078   fcb2   309F
loop    303F   numbuff 306F   offset  5200   print   304C   resid  3000   start  3000
```

Die Maus am TRS 80!

Kaum zu glauben aber wahr sind die Preise, die man heutzutage für eines dieser kleinen Rolltierchen auf den Tisch legen muß. Als ich mir vor knapp einem Jahr ein Mäuslein zulegte, habe ich 80,-- Märker gelöhnt und das, obwohl mir mein Finanzminister nur eine Gebrauchtmaus zugestand! Heute bekommt man die kleinen Grautiere neu schon für 69,-- DM (gesehen in der CHIP 10/87) und so wird die überlegung interessant, ob und wofür man eine Maus braucht und wie man sie an den TRS 80 anschließen kann.

Die erste Frage kann ich nur für mich persönlich beantworten, die Frage nach dem Anschluß glaube ich aber für alle Rechner der Tandyfamilie (und noch einige andere mehr) allgemeingültig gelöst zu haben. Ich benutze die Maus ausschließlich in Grafikprogrammen und dort ist sie sehr nützlich. Die Cursorpositionierung, die gerade beim 4p sehr problematisch ist, da die entsprechenden Tasten in der rechten unteren Ecke des Hackbretts kreuzartig zusammengefasst sind, wird erheblich vereinfacht und genauer. Die verschiedenen Zeichenprogramme sind mit Maus einfach leichter zu bedienen!

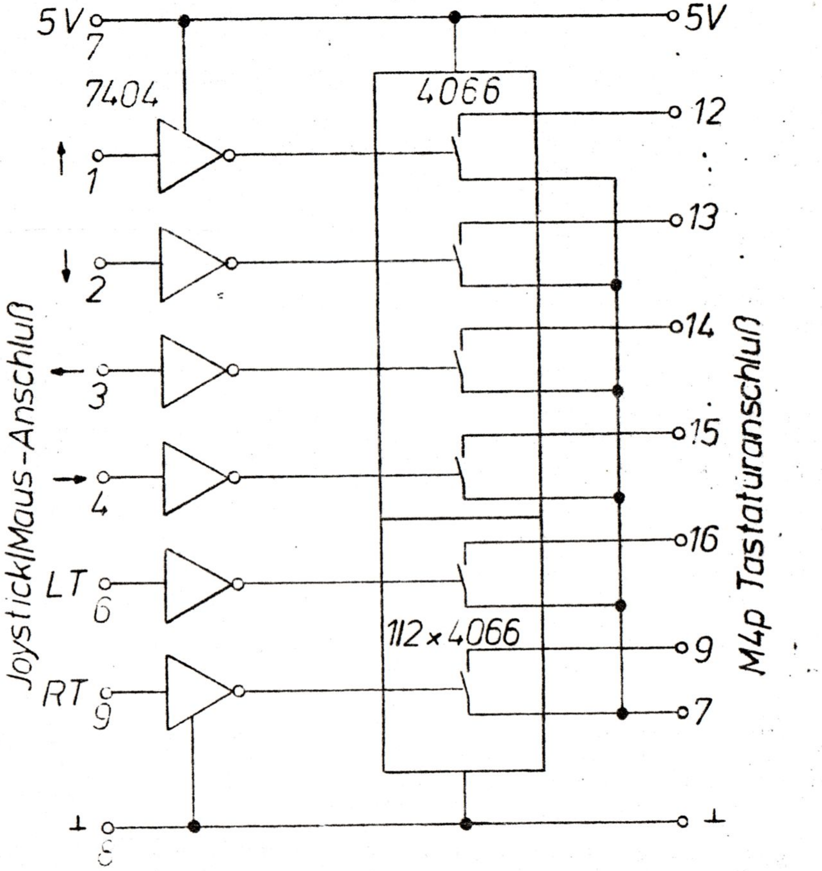
Was nun den Anschluß betrifft, müssen wir zunächst ein paar Sachen klären, da die Sache ansonsten mit Sicherheit in die Hose geht, denn "Maus ist nicht gleich Maus"! Wenn man sich einmal die Anzeigen in verschiedenen Zeitschriften genau ansieht, wird man feststellen, daß es mehrere verschiedene Maustypen gibt. Als Unterscheidungsmerkmal dient im allgemeinen die Art des Anschlusses an den Computer. Hier gibt es die sog. IBM-, RS 232- und last but not least die ATARI/Commodore-Mäuse. Obwohl mir alles, was den Vornamen Commodore trägt (ob es nun Computer, Floppys oder sonstwas sind) auf Anhieb erst einmal unsympatisch ist*, sind es die Mäuse dieser Firma die, die sich am leichtesten zur Zusammenarbeit mit den Tandys bewegen lassen.

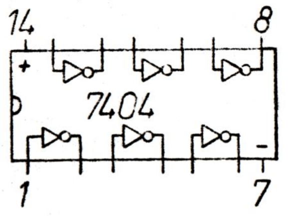
Selbstverständlich wird jetzt so mancher Besitzer einer RS 232-Schnittstelle sagen, daß diese Behauptung nicht stimmt. An seine serielle Schnittstelle kann er jederzeit eine RS 232-Maus anschließen und benötigt keinerlei zusätzliches Interface! Doch damit hat er nur teilweise recht. Zwar ist der hardwaremäßige Anschluß einer solchen Maus tatsächlich einfacher, die Einbindung in die Programme ist dagegen mit erheblichen Schwierigkeiten verbunden. Entweder müßte man für jedes Programm einen eigenen Maustreiber konstruieren oder einen solchen in das verwendete Betriebssystem einbauen.

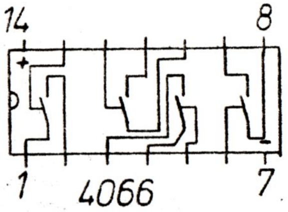
Die von mir entwickelte Schaltung enthebt den Anwender von der zusätzlichen Softwareeinbindung indem er die Maus praktisch parallel zur Tastatur schaltet. Sie ähnelt damit dem schon mehrfach gezeigten Anschluß eines Joystick an die TRS 80-Tastataur. Der Vergleich mit einem Joystick liegt schon deswegen nahe, da der Anschluß an den Computer über den gleichen Stecker erfolgt, und sich die Maus im Prinzip auch so verhält wie ein solcher Freudenstängel.

Drücke man den Hebel eines Joysticks in eine Richtung, wird ein Schalter geschlossen, der die Masseleitung mit dem entsprechenden Ausgang verbindet. Eine Maus gibt für jede Umdrehung in eine Richtung, am entsprechenden Ausgang eine Anzahl negativer Impulse ab. Diese Impulse kann man dazu nutzen, einen elektronischen Schalter die Pfeiltasten "drücken" zu lassen.

Und schicke uns Dein Manna vom Himmel, o Herr!







Der erwähnte elektronische Schalter ist in meiner Schaltung ein 4066. Dieses CMOS-IC enthält vier Analogschalter, die jeweils einzeln über einen Eingang betätigt werden können. Die prinzipielle Innenschaltung des IC's könnt ihr auf nebenstehender Zeichnung sehen. Einzig störender Umstand ist der, daß die Schalter mit positiver Spannung geschaltet werden müssen, während unser Mäuschen, wie oben schon erwähnt, negative Impulse abgibt. Aus diesem Grund ist noch ein Inverterbaustein 7404 zwischen Maus und 4066 zu schalten. Warum hier ein 7404 und nicht etwa ein CMOS-Inverter 4049 eingesetzt wurde, hat folgenden Grund. Während der Testphase, in der ich unter anderem mit einem solchen CMOS-IC arbeitete, traten immer wieder zunächst unerklärliche Cursorbewegungen auf, obwohl die Maus nicht bewegt wurde. Einige Messungen mit dem Oszilloskop zeigten sehr schnell, daß die CMOS-Inverter bei unbewegter Maus praktisch mit offenen Eingängen arbeiteten und dadurch keinen definierten Ausgangszustand annehmen konnten, was zu den erwähnten Cursorbewegungen führte. Der Einsatz der TTL-Inverter behob dieses Problem sehr schnell!

Wie man der Schaltskizze entnehmen kann, braucht man zum Anschluß der Maus also nur drei IC's (1 7404 und 1 1/2 4066), sowie eine Joystickbuchse. Der tastaturseitige Anschluß ist auf der Zeichnung für das Model 4p angegeben, kann aber ohne Probleme auf jede TRS 80- und im Prinzip auch auf jede andere Matrixtastatur umgesetzt werden.

Selbstverständlich kann an dem Mausanschluß auch jeder Joystick, auch die mit Dauerfeuereinrichtung, angeschlossen werden, worüber sich wohl vor allem die Spielfreaks freuen werden!

Damit genug gelötet! Viel Spaß beim (jetzt hätte ich doch beinahe "mausen" geschrieben) Basteln und Computern,

Hartmut Obermann

61 *Die Begründung für die ständige Verschlechterung des Wetters und die Verschiebung von Jahreszeiten und Klimazonen sehen Computerfachleute darin, daß Freund Petrus seinen Laden auf Computer umgestellt hat. Leider hat er dabei einen Fehler gemacht und sich einen Commodore angeschafft!

Helmut Bernhardt

Vor dem Untergang der offenen und entwicklungsfähigen Computertechnik durch Verdrängung durch den "Industrie-Standard" wurde von einer Aachener Firma für TRS-80 und GENIE eine 80-Zeichen-Karte angeboten. Wenn diese Karte auch in den meisten Fällen nur Probleme bereitete, war sie doch eine Option, die durch entsprechende Verbesserung durchaus eine Erweiterung für unsere Computer hätte werden können.

Diese Karte entsprach der TRS-80-Philosophie, daß der Bildwiederholspeicher in den Adreßraum der CPU eingeblendet ist und die CPU die Bildschirmverwaltung selbst erledigen muß, wie das bei unseren Computern für den 64x16-Bildschirm ohnehin gemacht wird und vom Level II-ROM und vom DOS auch unterstützt wird. Die 80-Zeichen-Karte wird allerdings nicht unterstützt. Nur bei neueren TRS-80- und GENIE-Rechnern ist bei der dort schon eingebauten 80-Zeichen-Option ein manchmal funktionierender Treiber im DOS enthalten. Aber all die Programme für den TRS-80, die vom 80x24-Bildschirm nichts wissen, und die Videoausgabe nicht über das DOS abwickeln, merken nichts davon, so daß bei Umschalten auf 80 Zeichen dann nur Zeichensalat entsteht.

Glücklicherweise gibt es immer Leute, die aus Begeisterung für ein bestimmtes Programm in mühevoller Arbeit dort die Unterstützung solcher Hardware-Erweiterungen einbauen. Dadurch wird ein 80x24-Zeichen-Bildschirm aber immer noch nicht TRS-80-Kompatibel.

Es sollte noch erwähnt werden, daß im GENIE III und IIIs ausschließlich mit dem CRTC6845 der Bildschirm verwaltet wird. Durch Programmierung des 6845 kann dieser im 64x16- und im 80x24-Format arbeiten. Im GIIIs ist außerdem der Zeichengenerator ein RAM-Speicher, in den beliebige Zeichensätze geladen werden können, wobei dann aber immer nur ein Zeichensatz zur Zeit benutzt werden kann. Außerdem verfügt der GIIIs noch über eine flexibel HRG, die mit dem Textbildschirm XOR-verknüpft angezeigt werden kann. Trotzdem bleibt es aber Aufgabe der CPU, im Textmodus die Bildschirmadressen zu berechnen und im HRG-Modus beim Zeichnen einer Linie alle Punkte der Linie zu errechnen.

Der eigentliche Sinn eines 80x24-Zeichen-Bildschirms ist auch nicht die Anpassung des NEWDOS/80 daran sondern die Nutzung unter CP/M. Für CP/M muß ohnehin das BIOS der Hardware angepaßt werden und da kommt es dann nicht mehr darauf an, ob ein 80x24-Bildschirm ähnlich unserem 64x16-Bildschirm bedient wird oder ob da völlig andere Verhältnisse vorliegen.

Wenn also eine noch so sehr an die TRS-80-Verhältnisse angenäherte 80-Zeichen-Karte dann doch nicht kompatibel ist, dann kann es auch eine beliebig angeschlossene Terminal-Karte sein. Solche Karten bedeuten zudem noch einen geringeren Programmieraufwand, weil das Anwenderprogramm bei einfacher Zeichenausgabe ohne absolute Cursor-Adressierung sich keine Gedanken um Bildschirmposition, Zeilen- und Seitenende und Scrolling machen muß. Diese Leistungen erbringt selbst das dümmste Terminal selbst.

Grundsätzlich ist ein Terminal ein Interface zwischen Computer und Anwender und umfaßt daher zur Ausgabe an den Anwender die Bildschirmausgabe (und manchmal zusätzlich auch die Druckerausgabe) und die Tastatureingabe. Die Eingabe erfolgt hauptsächlich über ASCII-Tastaturen, die mit einem Strobe-Signal dem Terminal ein ASCII-Zeichen übergeben. Das wird zum Teil parallel oder auch seriell gelöst.

Das Terminal gibt das Zeichen unbesehen an den Computer weiter und dieser echot es nach übernahme an das Terminal zurück. Erst dann wird das Zeichen vom Terminal auf den Bildschirm ausgegeben.

In einigen Fällen werden dabei Ausnahmen gemacht. So verstehen Terminals meistens ein bestimmtes Steuerzeichen als Befehl 'Local' und ein anderes (oder das gleiche ein zweites mal) als Befehl 'On Line'. Damit kann man umschalten zwischen Ausgabe nur der Zeichen vom Computer und sofortige Ausgabe der Zeichen von der Tastatur, ohne diese an den Computer weiterzureichen.

Manche Terminals können auch mit einem Steuerzeichen in einen Setup-Modus versetzt werden, von wo aus gewisse Parameter des Terminals über die Tastatur geändert werden können. Dazu kann gehören: die Umschaltung des Bildschirmformates, die Wahl nationaler Zeichensätze, die Änderung der Baudrate für die Kommunikation mit dem Computer u.s.w.

Die Verbindung mit dem Computer kann verschiedene Formen annehmen. Die verbreitetste Variante ist ein über eine serielle Schnittstelle angeschlossenes Terminal. Damit ist es möglich, daß das Terminal mit Bildschirm und Tastatur ein eigenständiges Gerät ist und der Computer weiter entfert stehen kann. Da bei den heutigen Tischgeräten dieser Gesichtspunkt nicht mehr ausschlaggebend ist, werden auch schon viele 'Video-Karten' für diverse Bussysteme als Steckkarten im Computer über den Systembus angeschlossen. Einige Varianten vereinigen beide Gesichtspunkte und verfügen über z.B. einen ECB-Bus-Stecker, der nur der Zuführung der Versorgungsspannungen dient, und eine RS232C-Verbindung mit dem Computer. Prinzipiell ist der Anschluß über den Systembus wegen des parallelen Datentransfers schneller. Bei Baudraten von 9600 oder 19200 ist aber auch der serielle Anschluß schnell genug. Auf alle Fälle kann er mit der Ausgabe eines memory mapped Bildschirms, der unter CP/M für die Auagabe jedes Zeichens ständig ein- und ausgeblendet werden muß, mithalten.

Das hat alles Vor- und Nachteile: Ein Terminal verlangt eine sehr viel leistungsfähigere Tastatur, die schon selbst die Umwandlung von ausgelesenen gedrückten Tasten in ASCII-Codes leisten muß. Bei unseren Rechnern muß das DOS bzw. der Z80 diese Umwandlung erledigen, was entsprechend umfangreiche Eingaberoutinen und auch CPU-Rechenzeit kostet. Da ein Programm aber bei Eingaben ohnehin nur auf die Tastatur wartet, ist das ohne Belang. Andererseits ist es bei einer ASCII-Tastatur auch nicht möglich, Mehrtasten-Befehle einzubauen (123, JKL, DFG). Ähnliches kann aber mit programmierbaren Funktionstasten oder einer Interrupt-getriebenen Terminaleingabe auch realisiert werden. Das wird aber in den meisten Fällen aus Angst vor Interrupt-Programmierung unterlassen.

Programmierbare Funktionstasten von ASCII-Tastaturen werden meistens durch Ausgabe von Codes > 7FH beim Drücken einer 'F'-Taste realisiert. Die Eingaberoutine des Betriebssystems lügt sich dann bei einem solchen eingegebenen Zeichen einen diesem Zeichen zugeordneten String als eingegeben vor.

Als reines Textausgabegerät kann also eine Terminal-Karte in TRS-80-Kompatiblen Computern insbesondere unter CP/M als leistungsfähiger Ersatz für eine von der CPU selbst verwaltete, memory mapped 80-Zeichen-Karte gute Dienste leisten. Der Anschluß einer ASCII-Tastatur ist möglich aber nicht zwingend. Gegenüber der Minimal-Tastatur des TRS-80 und des GENIE ist eine ASCII-Tastatur aber ein zusätzlicher Komfort, der die übliche Fingerakrobatik bei der Eingabe von Umlauten endlich durch dafür vorgesehene Tasten an der richtigen Stelle ersetzt.

Bei Terminal-Karten existiert ein eigener Prozessor, der den Bildwiederholspeicher verwaltet und die von der CPU übergebenen Zeichen entgegennimmt. Außer den darstellbaren Zeichen (20H-7FH) versteht er die meisten Steuerzeichen (00H-1FH) und setzt diese in entsprechendes Wohlverhalten um. Meistens reichen diese Befehle aber nicht für alle Wünsche der Bildschirmverwaltung aus, so daß

außerdem noch ESC-Sequenzen weitere Steuerfunktionen bereitstellen. In der Festlegung solcher ESC-Sequenzen und in deren Menge unterscheiden sich die verschiedenen Terminals voneinander. Die Bedienung eines Terminals ähnelt also mehr der eines Druckers als der des herkömmlichen TRS-80-Bildschirms. Und genauso wie bei den Druckern gibt es mehr oder weniger leistungsfähige Terminals.

So gibt es analog den Typenrad-Druckern auch reine Text-Terminals, die nur ASCII-Zeichen (natürlich auch deutsche Umlaute) darstellen können, und andere, die wie ein Matrixdrucker auch graphikfähig sind. Eine solche graphikfähige Terminalkarte, die GRIP, hat uns Manfred Held im letzten Info schon vorgestellt und auch gezeigt, daß sowas bei uns durchaus gute Dienste leisten kann. Die neueste Version der GRIP (Vers.5.53) ist in der Lage, HRG mit 768x567 Punkten darzustellen und Text in bis zu 96x35 Zeichen auszugeben. Durch den HD6345-CRTC, der aufwärtskompatibel zum MC6845 ist, wurde dort auch ein augenschonendes Smooth-Scrolling (Hochschieben des Bildschirminhaltes nicht textzeilenweise sondern dotzeilenweise) erreicht. Gegenüber der etwas lahmen Urform GRIP1 ist die GRIP5.53 eine der schnellsten Terminal-Karten für CP/M-Computer überhaupt.

Eine ähnlich leistungsfähige Karte ist die Video 1 von Janich & Claas, die ebenfalls HRG und Textdarstelleung beherrscht.

Weniger empfehlenswert ist die TERM 1 des mc-CP/M-Computers, die zwar wegen des EF9366-Video-Controllers in der Graphik einige interessante Aspekte bietet, die aber wegen der äußerst langsamen Textdarstellung überhaupt nicht zu gebrauchen ist.

Die GDP64 des NDR-Klein-Computers ist eine stark abgemagerte TERM 1 ohne eigenen Terminal-Prozessor. Hier muß die CPU selbst den EF9366 programmieren. Das bringt zwar Geschwindigkeitsvorteile, macht die normale Textausgabe aber recht umständlich. Diese Karte ist auch im eigentlichen Sinn kein Terminal, ihre Bedeutung ist auch einzig in der schnellen Vektorgraphik des EF9366 zu sehen.

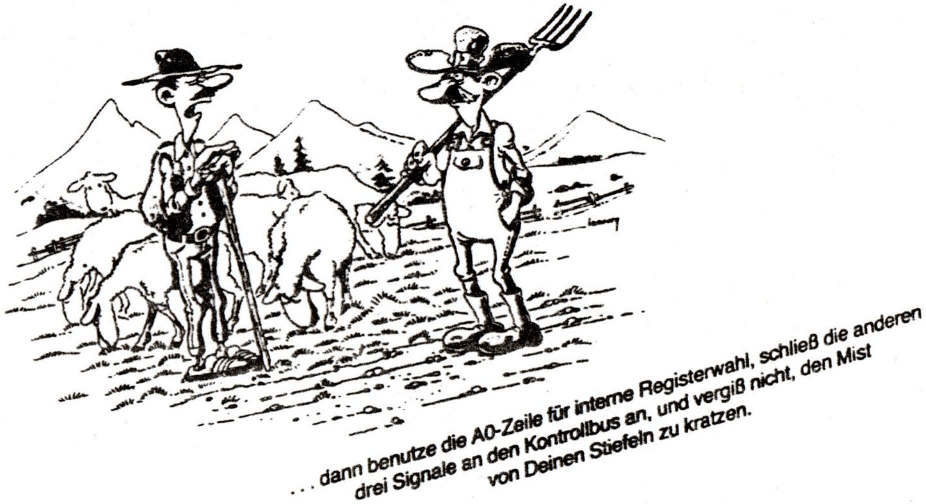
Beiden Karten mit dem EF9366 ist gemeinsam, daß sie die vereinahmten HRG-Inhalte nicht wieder hergeben (es sei denn, man rüstet diese Eigenschaft in Hardware nach).

In den meisten Fällen genügt es aber, wenn man für die Implementation von CP/M eine reine 80x24-Text-Terminal-Karte benutzt. CP/M fordert zwar keine 80x24-Zeichen-Darstellung und flexiblere Programme lassen sich auch für 64x16-Bildschirme installieren; die meisten unter CP/M laufenden Programme gehen aber von dieser Bildschirmgröße aus, so daß man beim 64x16-Zeichen-Format einen großen Teil der Informationen nur vorbeihuschen sieht und nicht lesen kann. Besonders ärgerlich ist dies bei den in vielen Utilities aus dem Programm heraus aufrufbaren HELP-Texten.

Eine sehr elegante Lösung besteht in der Verwendung des c't-Text-Terminals. Neben einer mit maximal 19200 Baud übertragungsrate sehr schnellen Textausgabe überzeugt diese Karte besonders durch ihr günstiges Preis/Leistungs-Verhältnis. Die Karte läßt sich für ca. 300 DM aufbauen und bietet dabei eine Fülle von Optionen. So lassen sich Zeichen mit bis zu 8 gleichzeitig wirksamen Attributen darstellen und es können gleichzeitig verschiedene Zeichen oder Zeichengruppen auf dem Bildschirm mit verschiedenen Attributen dargestellt werden. Die Attribute umfassen: Normaldarstellung, doppelt breit, doppelt hoch, halb hell, invers, blinkend und unsichtbar. Die Attribute werden durch entsprechende ESC-Sequenzen gesetzt und gelten dann für alle folgenden ausgegebenen Zeichen, bis andere Attribut-Befehle ausgegeben werden.

Literatur

Low-Cost: c't Text-Terminal; Tilman Reh; c't 9/86, 65
Das mc-Terminal; R.Jäger, H.-J. Regge, G. Tobergte, mc 9/83, 70
Das mc-Grafik-Terminal; R.-D. Klein; mc 8/83, 68; mc 9/83, 70
Grafik-Interface-Prozessor GRIP-1; J.C. Lotter, c't 6/84, 52; c't 7/84, 92



Schwimmende Bilder

Wer denkt, hier gibt's wieder was grafisches, hat leider Pech gehabt. Es geht vielmehr um einen kleinen, aber sehr störenden Fehler, der in der Hardware auftauchen kann.

Als ich für einen Freund mein altes Genie II wieder aus dem Schrank geholt habe und es auf Vordermann bringen wollte habe ich eine böse Überraschung erlebt: Booten ging noch, aber das Monitorbild war absolut unmöglich. Die Ränder wirkten, als ob Wasser die Mattscheibe herunterlaufen würde. Da bei mir kein Schlangenbeschwörer wohnt ging die große Sucherei los. Als Ursache stellte sich schließlich eine kaputte Diode im Netzteil heraus, die aus irgend einem Grund in Durchlaßrichtung hochohmig geworden war. Dadurch ist die Spannung für die TTLs von 5V auf ca. 4.3V gesunken. Warum das Ding überhaupt noch lief ist mir ein Rätsel. Nach dem Austausch war jedenfalls alles wieder in Ordnung und das Bild ruhig und stabil. Wenn Euer Kasten auch den Freischwimmer machen will, meßt erst mal die Spannungen nach. So schnell geht ein Rechner schließlich nicht kaputt. Meistens sind es nur solche lächerlichen Kleinigkeiten, die einem das Leben versauern.

Alexander Schmid

HEFT 22 Dezember 1987

BÖRSE -- BÖRSE -- BÖRSE

Es war mein schönstes!

GROSSER MARKTTAG !!!

VERSCHENKE (tatsächlich!)

EIN TOSHIBA-LAUFWERK

80 Track, doppelseitig

Es sei nicht verschwiegen, daß es reparaturbedürftig ist (infolge menschlichen Versagens versehentlich mit 220 V „bestromt" (aber nur ganz kurz, die Lichtsicherung flog sofort raus!). Für kündigen Hardwarefreak sicher eine Kleinigkeit. Abholung oder (als Bestellung) für Porto DM 3,— auf Postgirokonto München, Nr. 3847 37-804, Frau Ulrike Schneider; Kennwort: LAUFWERK.

Des weiteren habe ich mich entschlossen, mich von einem meiner beiden GENIE III zu trennen, wenn sich ein Interessent bei fairem Wertausgleich (ich dachte so an etwa 1270,— DM) findet.

Für VG I/II - TRS 80 Model I-Besitzer wären alle Probleme einer 80-Zeichenkarte, freies RAM für CP/M, Kabelsalat usw. komfortabel gelöst! Außerdem mit (selbstverständlich mitgeliefertem) NEWDOS/GDOS-System volle Kompatibilität zu diesen Maschinen (aber etwa doppelt so schnell!).

Bildschirm umschaltbar 64x16 oder 80x25 Z./Z. Freie Tastatur, deutsch mit Umlauten, Ziffernblock, 8 Funktionstasten frei belegbar (bis 16 mit SHIFT), zwei eingebaute 80-Track-Laufwerke DS (1,4 MByte formatiert), zwei weitere LW extern anschließbar. Aufrüstung bis 1 MByte RAM möglich (vgl. Clubinfo 14, Juli 86, S. 91 - 100, an meinem anderen GENIE III erfolgreich für lässige 199,- bis 249,- DM + Clubfreundhilfe praktiziert: RAM-Disk, SYS-RAM, vier nahezu unabhängige Computer mit verschiedenen Programmen gleichzeitig auf Knopfdruck usw., und nochmal 2 - 10 mal schneller...!)

Bei Interesse Unmassen von Software jeder Art (original), „Geniothek" mit GENIETEXT + Kalkulation, GenieCalc, Lohn + Gehalt, rel. Datenbank DIS (80x25 Z.-Bildschirm). Nachkalkulation (Orgelbau, aber leicht für jede Branche zu ändern, da komfortabel-professionelle BASIC-Programmierung meines Sohnes), Buchführungen (freiberuflich und andere), Zeichengenerator HEADLINE, ASEM(blerkurs) 4 und OPAL 68000 Crossassembler (alles von Wilke/IDA), SuperCross (ca. 180 Formate einschl. MS-DOS lesen) mit Basic-Programmkonvertierung etc. usw. und so fort (2 x die Hälfte vergessen!). Dazu CP/M mit Shell, mit Wordstar, Turbopascal u.a. Außerdem ca. 1/2 t einschlägige Literatur.
Darüber müßte gesondert verhandelt werden.

Alles befindet sich in gutem Zustand. Die genannten Preise gelten bei Abholung, andernfalls + Porto.

Richard Rensch Bahnhofstraße 100, 7128 LAUFFEN am Neckar

Unser ehemaliges Mitglied Uwe Gromottka hat sich, nachdem er sich auch beruflich mit MS-DOS beschäftigen muß, nun auch für private Zwecke einen solchen Rechner angeschafft. Dadurch wird bei ihm natürlich eine ganze Menge Hardware frei, die er nun veräußern möchte. Es steht demnach zum Verkauf:

- Video Genie II mit eingebautem Disk-Controler 1771/1791
- CMC Monitor grün 12 Zoll
- 1 Diskettenlaufwerk 40/DS/DD
- Schmidtke 80 Zeichen-Karte mit Banking für CP/M.

Auch Paperware gibt es eine ganze Menge:

- Trommelschläger's Service Manual
- TRS-80 Disk und andere Geheimnisse in deutsch
- Betriebssystem CP/M (Plate / Franzis-Verlag)
- Programmierung des Z80 (Zaks / Sybex)
- Kopieen von Programmanleitungen und anderen Schriften.

Die Preise sind Verhandlungssache und direkt zu erfragen bei: Uwe Gromottkat
Lange Reihe 40
2803 Weyhe
☎ 0421 / 88496

EPROM's (2716-27513) läßt man bei P. Spiels brennen! Auch C-MOS

DDR und Computer

Seit kurzem habe ich Kontakt mit einem jungen DDR-Bürger, der zur Zeit eine Lehre im Elektronikbereich durchmacht. Dieser junge Mann wünscht sich nichts sehnlicher (ist vielleicht etwas übertrieben!) als einen Computer. Da es so etwas in der "anderen Republik" zum Privatgebrauch nicht gibt, bat er mich um Hilfe.

Ich gebe diese Bitte an euch weiter. Wenn irgendwo, bei irgendwem ein Computer (vom programmierbaren Taschenrechner bis zum TRS 80/Genie) unbenutzt in der Ecke steht und sowieso auf den Müll soll, wäre dies eine sinnvolle Verwendung für das Gerät! Einzige Bedingung, die erfüllt sein sollte: das Gerät sollte funktionsfähig sein, da Ersatzteile in der DDR nicht zu beschaffen sind. Auch eine Anleitung wäre von vorteil, da begleitende Literatur mit Sicherheit fehlt!

Wer also seine Hardware vom TI 57 bis zum MS-DOS-Computer (kostenlos) los werden möchte, kann sich an mich wenden. Ich gebe auch gerne die Adresse zur weiteren Kontaktaufnahme weiter!

Hartmut Obermann

Unidirektionaler Traktor für den NEC P6 (im Lieferumfang des Neugeräts nicht enthalten), komplett mit höherem Geäusedeckel und Papierführung. Zustand wie neu. Neupreis Ende 1986: DM 130,-. Für DM 75,- von Arnulf Sopp, Tel. 0451-791926

BÖRSE -- BÖRSE -- BÖRSE

SOS

Suche für TRS-80/M I mit 48 k RAM
das NEWBAS-Programm auf Kassette.

W. Hentz
Am Tränkgarten 20
6457 Maintal 2
Tel. (0 61 09) 6 66 25 ab 19 Uhr

VERKAUFE:

- TRS 80, Model 1, L 2, 16 K
- 2 TANDON-Laufwerke, 40 Tr., SS, DD

beides gegen entsprechende Gebote an
Klaus Hermann

...---... HILFERUF von Ka-Jott in Not(t): ...---...

? Wer kennt das ?

TRS-80 MUSIKPROGRAMM

Disk Version von W. Müller, Ratingen

und kann mir die Anleitung zum Kopieren geben ?
Seit TANDY's Hochburg in Ratingen (b. Düsseldorf) aufgelöst ist, gibt's von dort keine Hilfe mehr – <wem sag' ich das?> [Jetzt sind wir Kundendussel doof geblieben]

SOS

Dieses Programm bringt die saubersten Töne hervor – aber es gelingt (mir) nicht, mehr als eine Bildschirmseite zu schreiben: THERE IS NO „SCROLLING"!

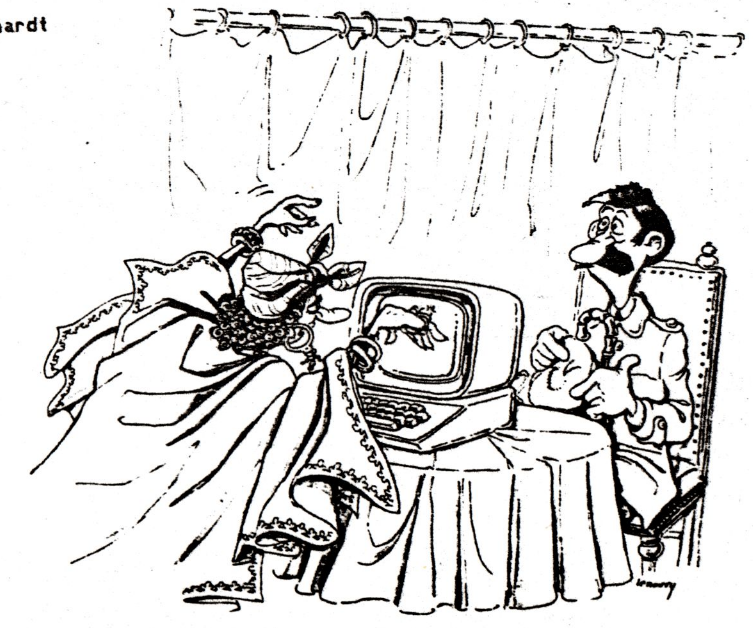
"Prüfzahlfehler beim Lesen"

Diese Horrormeldung stelle man sich vor, wenn man nach dem überarbeiten eines 3 Files umfassenden Assemblerprogramms, das beim letzten Assemblieren 7 K Byte Objektcode ergab, einen der Source-Files laden will. Wenn sich dieses Erlebnis dann auch noch bei den anderen beiden Files einstellt, reicht das dann erst mal für den Rest des Tages. Wer etwas härter im Nehmen ist, macht dann noch einige zaghafte Versuche mit SUPERZAP, das bei etwas Glück mit seiner Vermutung, daß einige oder alle Daten des defekten Sektor im Puffer enthalten sind, Recht hat. Leider weigert es sich, diese nach eventuell nötigen Korrekturen wieder auf die Diskette zurückzuschreiben. Der Umfang des Schadens läßt sich zwar feststellen aber nicht beheben.

So ganz wollte ich mich aber nicht mit der Vorstellung anfreunden, 32 DIN A4 Seiten gut kommentierten Programms neu einzutippen. Also erst mal versuchen, ob sich der Rest der Files auf eine andere Diskette kopieren läßt. Wenn dann da einige Sektoren mit E5H gefüllt sind, würde ich da mit SUPERZAP schon was reinbringen, das ich hinterher mit ZEUS wieder in sinnvollen Text verwandeln könnte. Hauptsache, es tritt kein CRC-Fehler mehr auf. Beim Kopieren mit COPY,0,1 traten dann auch die erwarteten Meldungen auf, die ich mit einem (F)ortfahren beantwortet habe. Insgesamt 5 Sektoren hat das DOS beanstandet. Ich sollte offensichtlich 1,25 KByte von E5H in von ZEUS verwertbares Material umzapen. Na gut, an die Arbeit, mit SUPERZAP den ersten beim Kopiermaterial beanstandeten diskrelativen Sektor geholt und - eitel Freude - es war keine einzige E5H zu sehen sondern mein Originaltext, der ohne CRC-Fehler geladen wurde.

Von den anderen 4 Sektoren war dann noch einer geringfügig zu korrigieren und nach einer halben Stunde waren meine Source-Files wieder heil. Das mag für manchen von Euch der Schnee von gestern sein, wenn man aber unter der Vorstellung lebt, bei solchen Katastrophen nur das Handtuch werfen zu können, dann mag diese Erfahrung recht hilfreich sein. Trotzdem hoffe ich, daß möglichst wenige von Euch drauf zurückgreifen müssen.

Helmut Bernhardt



* * * Fraktur - * * *

- mit jemand reden, heißt: "Eckig" mit ihm reden, ſozuſagen "gerade heraus"!
Fractura = lat. Bruch. - Frakturschrift (Frakturdruck) = "gebrochene Schrift".
"Mit jemand Fraktur reden" kann alſo zum Bruch führen! Darin kenne ich mich beſſer aus als in der Aſſemblerſprache. (Manch einer weiß das!)
Nicht nur, ſie zu reden - auch ſie zu leſen macht Spaß.
Denn dann iſt ſie viel ſchöner gedruckt, als meine hier, die ich mit TSCRIPS' Zeichenbau-Utility verbrochen ("fraktuiert"?) habe.
Recht eckig iſt ſie, im Gegenſatz zur runden lat. ANTIQUA, die man faſt nur noch vorgeſetzt bekommt.

Die Redensart ſtammt übrigens aus dem 17.Jahrhundert. (Das iſt das Jahrhundert, deſſen Jahreszahlen mit 16.. anfangen. Nicht 17.., wie neulich in einer "Fachzeitſchrift" zu leſen ſtand...)

Hier eine Koſtprobe meines Alphabetes, da bisher noch nicht alle Buchſtaben und Ziffern vorkamen:

ABCDEFGHIJKLMNOPQRS TUVWXYZ 1234567890
aäbcdefghijklmnoöpqrsſßtuüvwxyz

Manchem gefällt's nicht (u.a. meiner Frau.) "Geſchmacksſache". - Geſchmacksſache? Über den Geſchmack "läßt ſich (nicht) ſtreiten". (Dieſes unklare Sprichwort ſollte beſſer lauten: "Über Geſchmack läßt ſich diskutieren".)

Diejenigen, die weder mit mir über Geſchmacksfragen noch mit TSCRIPS über Fingerfertigkeit und HRG-Fähigkeit "ſtreiten" wollen, können gern meinen Zeichenſatz auf Diſk (oder Diskette) haben (formatiert zwischen 0 bis 80 Tracks, von undicht bis doppelt dicht). Anſchrift hinten.

Übrigens: Wer Angſt hat vor dem böſen "ſ", deſſen richtiger Einſatz heute weitgehend unbekannt iſt, braucht ſich darum keine Sorgen zu machen und kann getroſt ſtets "s" ſchreiben. * Wer aber die umfaſſende Regel wiſſen möchte - nun gut, ſie lautet:

Das "lange" (ſog. ſtumme oder ſcharfe) "ſ" ſteht am Silbenanfang und das "s" am Silbenende!

Wie ſchreibt ſich alſo: Das Beste?
Das Beste? - Mitnichten! Sondern: Das Beſte! - Denn beim Trennen iſt es das be=
ſte, man läßt das "ſt" beiſammen; dieſes wird nie zerriſſen!
Das "ſ" ſteht hier alſo am Silbenanfang.
Aber wieſo ſchreibt man dann das Wort "beſſer" beim Trennen: beſ=
ſer, alſo auch mit zwei langen "ſ", wo doch das ſ der er=
ſten Silbe ein "Schluß-Es" iſt???
Nun, ganz einfach:

<u>Keine Regel ohne</u> - ach so, ihr wißt schon.

Und die Ausnahmen ſind es doch, die die Regeln und das Leben ſo inter-
eſſant machen.

rXvss / - zaiot

Sicherer Platz für Daten?

Der größte Unsicherheitsfaktor beim EDV-Betrieb? Vermutlich der Mensch. Die Fehlerrate von Computerbedienern konnten und wollten wir nicht testen, dafür aber die Zuverlässigkeit des gebräuchlichsten Speichermediums für Personal Computer: 20 verkaufshäufige Disketten offenbarten uns ihr magnetisches Gedächtnis.

Was man schwarz auf weiß besitzt, kann man getrost nach Hause tragen.« Mit dieser Meinung steht der beflissene Schüler aus Goethes »Faust« auch im Computer-Zeitalter keineswegs alleine da. So ließe sich heute des Dichters Meisterwerk, Vers für Vers in Bits übersetzt, auf einem Dutzend Disketten unterbringen, aber man hätte den »Faust« damit gleichzeitig den Unwägbarkeiten der Elektronik auf Gedeih und Verderb ausgeliefert. Was zum Beispiel, wenn aufgrund eines Materialfehlers die Gretchenfrage von der Magnetschicht verschwinde oder der Osterspaziergang einem auf der Diskette haftenden Schmutzpartikel zum Opfer fiele? Sicher, auch Bücher, Zeitschriften, Aktenordner sind nicht unzerstörbar oder können verlorengehen.

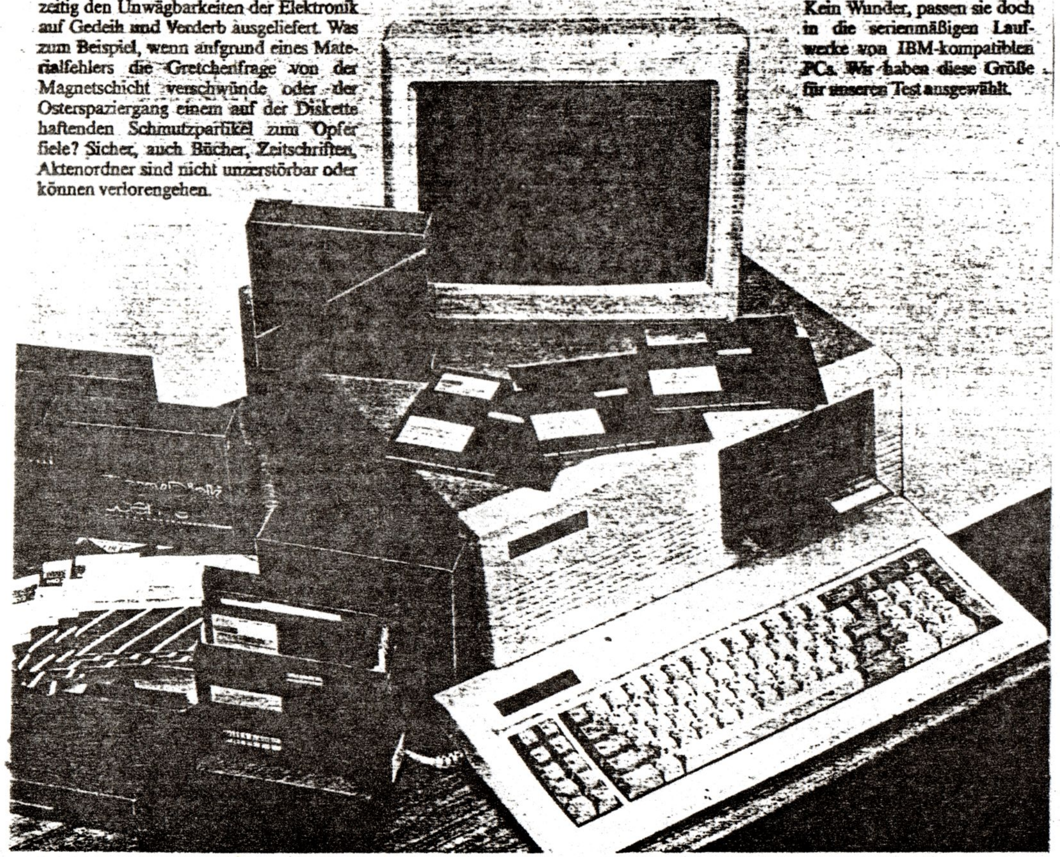
Aber der Gedanke, wichtige Daten einer unscheinbaren Plastikscheibe anzuvertrauen, widerstrebt zuweilen selbst aufgeklärten Zeitgenossen.

Wer mit Computern arbeitet, muß auch speichern. Bei Daten für den sofortigen Gebrauch genügt der Arbeitsspeicher, der in jedem Rechner eingebaut ist. Für die dauerhafte Aufbewahrung von Informationen braucht man dagegen sogenannte Massenspeicher. In der Regel sind das magnetische Datenträger. Nicht wenige Homecomputer-Besitzer benutzen dazu Audio-Kassetten – vor allem wegen der billigen Aufnahmegeräte. Diese Speicherung ist freilich ebenso umständlich wie zeitraubend, da man mitunter die ganze Kassette umspulen muß, um an die gewünschte Stelle heranzukommen. Bei Personal Computern (PC) verwendet man deshalb nur noch Disketten oder Festplatten. Beide gewährleisten den direkten Zugriff auf alle Daten. Festplatten haben darüber hinaus unbestreitbare Vorzüge (große Kapazität bei geringer Fehlerquote und sehr kurzen Zugriffszeiten), dafür bleiben die Daten auf Disketten mobil, geeignet also für Transport oder Versand.

Disketten gibt es in drei verschiedenen Größen, die nach der Kantenlänge der Hülle benannt sind:

■ 8-Zoll- oder Maxidisketten (Kantenlänge etwa 20 Zentimeter) haben für den PC-Bereich keine Bedeutung.

■ 5¼-Zoll- oder Minidisketten (Kantenlänge 13 Zentimeter) haben den Löwenanteil am Diskettenmarkt. Kein Wunder, passen sie doch in die serienmäßigen Laufwerke von IBM-kompatiblen PCs. Wir haben diese Größe für unseren Test ausgewählt.



■ 3½-Zoll- oder Mikrodisketten – sie messen neun Zentimeter an den Kanten – gehört nach Meinung mancher Experten die Zukunft. Die steil ansteigende Verkaufskurve spricht dafür. Neben der geringen Größe gilt auch ihr stabiles Hartplastikgehäuse als Pluspunkt. Besitzer von Apple- und Atari-PCs profitieren schon jetzt davon.

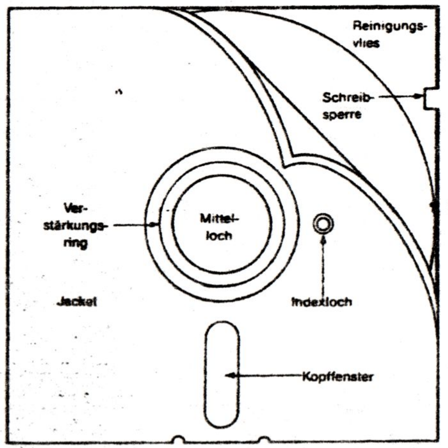
Obwohl wir uns bei unserem Test bereits auf eine Größe festlegten, konnten wir unmöglich das gesamte Angebot in diesem Segment berücksichtigen. So verzichteten wir zum Beispiel auf die No-name-Fabrikate. Auch mußten wir uns auf Disketten mit folgenden Kenndaten beschränken: zweiseitig beschreibbar, doppelte Speicherdichte, Spurdichte: 48 tpi, softsektoriert. Wenn Sie nun daraus folgern, daß es auch hardsektorierte oder einseitige Disketten mit einfacher Speicher- und anderer Spurdichte gibt, liegen Sie genau richtig. Weder das Äußere einer Diskette noch ihre Bezeichnung läßt indes erkennen, wie die Signale auf der Magnetschicht angeordnet sind.

Beschreibbarkeit: Da es Floppy-Laufwerke mit einem oder zwei Schreib-/Lese-Köpfen gibt, bekommt man auch Disketten, die sich von einer Seite und solche, die sich beidseitig beschreiben lassen. Der Unterschied? Im Prinzip keiner, nur sind zweiseitig beschreibbare Disketten vom Hersteller auch für diese Anwendung geprüft, während bei den anderen die Kehrseite durchaus fehlerhaft sein kann. Man könnte also probieren, eine Diskette, die für Einkopf-Laufwerke vorgesehen ist, beidseitig zu beschreiben – muß sich dann allerdings über einen Datenverlust infolge einer schadhaften Stelle nicht wundern. Deshalb unser dringender Rat: Verwenden Sie für Doppelkopf-Laufwerke nur die dafür vorgesehenen Fabrikate.

Die Signalaufzeichnung und -abtastung erfolgt bei Disketten spurweise, und zwar von außen nach innen in abgeschlossenen, konzentrischen Kreisen. Der Magnetkopf zieht also erst ganz außen seine Bahn, um dann nach jeweils einer Umdrehung in einen Kreis mit etwas geringerem Durchmesser zu »springen«. Je mehr Spuren der Magnetkopf auf der Diskette zieht, desto größer ist die Informationsfülle, die sich unterbringen läßt. Die Anzahl dieser Spuren wird in tracks per inch (tpi), also Spuren pro Zoll angegeben. Die gängigen Maßzahlen für die Mini-Diskette sind 48 und 96, wobei 48 tpi lediglich 40 Spuren und 96 tpi 80 Spuren auf der Diskette entspricht. Grundsätzlich könnte eine 48-tpi-Diskette auch in einem Laufwerk mit 96 tpi – wie es der IBM-AT-Computer besitzt – gelesen werden. Das Beschreiben klappt freilich nicht, denn wegen der höheren Signaldichte benötigen die IBM-AT-Laufwerke Disketten mit einer anderen Magnetfolie, sogenannte High-Density-Fabrikate.

5¼ Zoll – genau betrachtet

So unkompliziert, wie es von außen scheint, sind Disketten keineswegs. Jedes Detail hat seine Funktion: Das **Mittelloch** dient dazu, die Diskette zu zentrieren und zu transportieren. Am **Indexloch** erkennt das Laufwerk den Sektorenanfang. Das **Kopffenster** gewährt dem Schreib-/Lesekopf Zugang zur Magnetplatte. Die **Schreibsperre** funktioniert umgekehrt wie bei Kompaktkassetten: Bei »offener« Einkerbung kann die **Diskette** beschrieben werden, durch einen Klebestreifen lassen sich die Daten vor unbeabsichtigtem Löschen schützen. Das Äußere der Diskette nennt man übrigens **Jacket**. Es ist mit einem **Reinigungsvlies** verbunden, das die Magnetoberfläche sauberhält.



Freilich ist die Anzahl der Spuren nicht der einzige Anhaltspunkt für das Aufnahmevermögen der Magnetscheiben, denn auch die Speicherdichte längs der Spuren kann unterschiedlich sein. Da es aber kaum jemanden interessiert, wieviel Bit auf einer Diskettenspur unterzubringen sind, geben die Hersteller auf ihren Produkten meist nur an, ob diese einfache oder doppelte Speicherdichte haben. Der Normalfall ist heute die doppelte Dichte, abgekürzt zum Beispiel mit dem Buchstaben D. Disketten einfacher Dichte sind nur wenig gebräuchlich.

Eine Normung, wie sie sich zum Beispiel bei Audiokassetten bewährte, gibt es zwar auch für Disketten, aber durchgesetzt hat sie sich leider noch nicht. Der Käufer muß also bis auf weiteres die technischen Gründdaten mehr oder wenig mühsam aus der Artikelbezeichnung entschlüsseln. In einigen Fällen fehlen die Informationen auf der Verpackung oder Diskettenhülle ganz und gar. Disketten sind übrigens abwärtskompatibel, man kann also eine für 96 tpi geprüfte Diskette durchaus mit 48 tpi oder eine zweiseitige nur auf einer Seite bespielen.

Alle für Personal Computer gebräuchlichen Disketten sind softsektoriert. Das bedeutet: Außer dem Indexloch, das dem Schreib-Lesekopf anzeigt, wo er mit der Arbeit beginnen soll (siehe auch Schnittzeichnung oben), haben sie keine weiteren sichtbaren Markierungen in der Magnetplatte. Die Sektoreneinteilung übernehmen magnetische Informationen bei der Formatierung. Bei den ersten – hardsektorierten – Disketten besorgten hingegen richtige Löcher die Sektoreneinteilung – die somit auch sichtbar war.

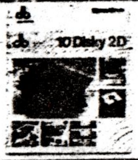
Funktionsprüfung

Die bei unseren PC-Systemen notwendige Formatierung war auch Gegenstand einer Prüfung, die wir an jeweils 90 Exemplaren jeder Diskettenmarke durchführten. Eigentlich dürfte es dabei keine Probleme geben, aber bei drei Fabrikaten widersetzten sich einzelne Disketten beharrlich dieser Sektoreneinteilung. Wenn aber auf einer Diskette einige tausend Bytes gesperrt sind oder sie sogar mechanische Fehler aufweist, kann sie als zuverlässiger Informationsträger kaum noch gelten. Dabei trägt die Boeder-Hi-Tec-Diskette, die in der Formatierprüfung versagte, das Prädikat 100 % error free. An diesem hohen Anspruch muß sie sich dann auch messen lassen.

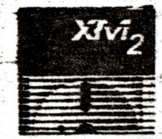
Im Gegensatz zu Video- und Audiobändern, bei denen ja auch magnetisch aufgezeichnet und wiedergegeben wird, dürften Disketten im Grunde genommen keine Fehler aufweisen. Während sich beispielsweise ein sogenannter Drop-out, also ein Signalausfall infolge Verschmutzung oder Beschichtungsfehler, dort lediglich als Tonstörung oder Blitzen im Bild bemerkbar macht, hat er hier mitunter verheerende Folgen. Wenn nämlich einige Daten auf der Diskette fehlen, geht das Programm nach dieser Fehlerstelle nicht einfach zur Tagesordnung über, sondern 74

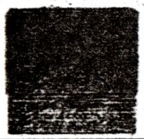
Fortsetzung auf Seite 28

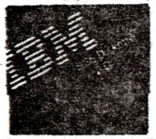
75 76



Fabrikat (Anbieter-Adressen siehe Seite 28)	[redacted]			BASF Flexy Disk Qualimetric 2D[1]	BASF Flexy Disk Science 2D[1]	Dysan Flexible Diskette 104/2D	Maxell MD 2-D	Nashua MD2D WP-R Art.-Nr. 08725-300	Quelle Privileg Art.-Nr. 993.511	RPS High Focus Floppydisk MN 2 DD-O-G
Preis für 10 Disketten nach Markterhebung in DM				55,– bis 76,–			40,– bis 53,–			49,– bis 54,90
Mittlerer Preis für 10 Disketten in DM				43,–	118,–	72,–	45,50	39,–	14,95	53,–
Verpackung				Stülpschachtel	Stülpschachtel	Stülpschachtel	Aufstellbox	Faltschachtel	Stülpschachtel	Faltschachtel
Material der Einstecktasche				Faservlies	Faservlies	Faservlies	Faservlies	Karton	Karton	Faservlies
test-Qualitätsurteil	[redacted]			GUT	GUT	GUT	GUT	GUT	GUT	GUT
TECHNISCHE PRÜFUNG	sehr gut	sehr gut	sehr gut	gut	sehr gut	gut	zufriedenstellend	zufriedenstellend	gut	gut
Prüfung der Abmessungen	gut +	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++
Dauerprüfung	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++
Temperaturprüfung bis 50 °C	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	zufriedenstellend O	sehr gut ++	mangelhaft*) – Hülle verformt	zufriedenstellend O	sehr gut ++
Verarbeitung der Hülle	sehr gut ++	sehr gut ++	gut +	gut +	gut +	sehr gut ++	gut +	sehr gut ++	gut +	gut +
Verarbeitung der Magnetplatte	sehr gut ++	sehr gut ++	sehr gut ++	zufriedenstellend*) O deutlicher Grat am äußeren Rand	sehr gut ++	sehr gut ++	mangelhaft*) – Aufzeichnungsbereich unvollständig poliert	gut +	gut +	zufriedenstellend*) O
FUNKTIONSPRÜFUNG	sehr gut	sehr gut	sehr gut	gut	gut	gut	sehr gut	sehr gut	gut	gut
Fehler beim Formatieren (90 Prüfmuster)	keine ++	keine ++	keine ++	keine ++	keine ++	keine ++	keine ++	keine ++	keine ++	keine ++
Elektromagnetische Eigenschaften	sehr gut ++	sehr gut ++	sehr gut ++	gut +	gut +	gut +	sehr gut ++	sehr gut ++	gut +	gut +







Fabrikat (Anbieter-Adressen siehe Seite 28)	Sentinel SS-22	Verbatim Verex Minidisk MD 250-01	Xidex XM, Flexible Disk 5022-1000	Boeder Hi-Tec 2D Art.-Nr. 37540	3M Sicherheits-Diskette DS, DD Art.-Nr. 745-0 RH (gleich)	Scotch Mini-Diskette 2S 2D 48 tpi (gleich)	IBM P/N 602 34 50	Kaufhof elite Minidiskette Art.-Nr. 693190	Memorex 2S/2D Flexible Disk Art.-Nr. 3203 5220	Verbatim Datalife Minidisk MD 550-01 Art.-Nr. 18188
Preis für 10 Disketten nach Markterhebung in DM		12,– bis 29,50			42,50 bis 63,–					35,– bis 45,–
Mittlerer Preis für 10 Disketten in DM	56,–	26,–	31,–	52,50	49,–	46,–	90,–	17,95	54,80	39,50
Verpackung	Aufstellbox	Faltschachtel	Stülpschachtel	Faltschachtel	Stülpschachtel		Stülpschachtel	Faltschachtel	Faltschachtel	Stülpschachtel
Material der Einstecktasche	Faservlies	Karton	Karton	Karton	Karton		Faservlies	Karton	Karton	Karton
test-Qualitätsurteil	GUT	GUT	GUT	ZUFRIEDEN-STELLEND	ZUFRIEDENSTELLEND		ZUFRIEDEN-STELLEND	ZUFRIEDEN-STELLEND	ZUFRIEDEN-STELLEND	ZUFRIEDEN-STELLEND
TECHNISCHE PRÜFUNG	sehr gut	gut	sehr gut	sehr gut	zufriedenstellend		zufriedenstellend	gut	zufriedenstellend	zufriedenstellend
Prüfung der Abmessungen	sehr gut ++	sehr gut ++	sehr gut ++	gut +	sehr gut ++		gut +	sehr gut ++	mangelhaft*) –	zufriedenstellend O
Dauerprüfung	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++		sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++
Temperaturprüfung bis 50 °C	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++	sehr gut ++		mangelhaft*) – Hülle verformt	zufriedenstellend O Hülle leicht verformt	sehr gut ++	mangelhaft*) – Hülle verformt
Verarbeitung der Hülle	gut +	sehr gut ++	gut +	sehr gut ++	gut +		gut +	gut +	sehr gut ++	sehr gut ++
Verarbeitung der Magnetplatte	gut +	zufriedenstellend*) O Aufzeichnungsbereich etwas unvollständig poliert	sehr gut ++	sehr gut ++	mangelhaft*) – Aufzeichnungsbereich unvollständig poliert		gut +	zufriedenstellend O Aufzeichnungsbereich etwas unvollständig poliert	gut +	gut +
FUNKTIONSPRÜFUNG	gut	gut	gut	zufriedenstellend*)	zufriedenstellend		zufriedenstellend	zufriedenstellend	zufriedenstellend	zufriedenstellend
Fehler beim Formatieren (90 Prüfmuster)	gut +	keine ++	keine ++	mangelhaft*) – bei einem Exemplar 5120 Bytes gesperrt	keine ++		keine ++	mangelhaft – eine Diskette mechan. beschädigt	mangelhaft – bei einem Exemplar 25600 Bytes gesperrt	keine ++
Elektromagnetische Eigenschaften	gut +	gut +	gut +	gut +	zufriedenstellend O aber bei zwei Exemplaren Signalausfälle möglich		zufriedenstellend O	zufriedenstellend O aber bei einem Exemplar Signalausfälle möglich	zufriedenstellend O aber bei einem Exemplar Signalausfälle möglich	zufriedenstellend*) O

*) Führt zur Abwertung (siehe „Ausgewählt ...“ auf Seite 28)

1) Lt. Anbieter ab August '87 Nachfolgeprodukte mit geänderter Bezeichnung im Handel

Test Disketten (5¼ Zoll)

Fortsetzung von Seite 25

kommt im ungünstigsten Fall völlig durcheinander. Ein Absturz mit Datenverlust ist dann leicht möglich.

Ähnlich können sich sogenannte Drop-ins auswirken. Das sind Störsignale, die das Laufwerk daran hindern können, die gespeicherten Daten fehlerfrei zu lesen. Daß es in diesem Prüfpunkt, der ebenso wie die Drop-outs, die Überschreibbarkeit, Modulation und mittlere Lesespannung unter unserer Überschrift »Elektromagnetische Eigenschaften« läuft, keine Ausfälle gab, spricht für die Qualität der Disketten.

Technische Prüfung

Kein Ruhmesblatt für die Endkontrolle sind hingegen jene von uns entdeckten Datenträger, bei denen schon die Verarbeitung zu wünschen übrig ließ. Ein Beispiel: Einige Memorex-Exemplare wichen in den Abmessungen so deutlich von der Norm ab, daß sie manchen Laufwerken mechanische Schwierigkeiten bereiten könnten. Ärgerlich ist es allemal, wenn eine Diskette klemmt. Unvollständig polierte Oberflächen auf der Magnetscheibe fielen uns gleich ein paarmal auf, besonders unangenehm bei Maxell und Scotch/3M. Weitgehend unempfindlich indes reagierten die meisten Test-Disketten auf hohe Temperaturen: 50 Grad Celsius überstanden fast alle ohne erkennbare Qualitätseinbußen. Bei einigen hatte sich allerdings nach dieser Prüfung die Hülle verformt.

Die Dauerprüfung meisterten alle Disketten ohne Probleme. Leider ist es dennoch nicht möglich, Aussagen über ihre Haltbarkeit auf Jahre hinaus zu machen. Unsere begrenzten Prüfzeiten lassen die dazu notwendigen Langzeitversuche einfach nicht zu. Wer allerdings von wichtigen Datenträgern Sicherheitskopien anfertigt, braucht erfahrungsgemäß nichts zu befürchten.

Trotz der beachtlichen prüftechnischen Unterschiede rühren wohl die meisten Diskettenfehler vom falschen Benutzerverhalten her. So darf man die Magnetschicht keinesfalls mit Flüssigkeiten oder Fremdkörpern in Berührung bringen oder anfassen. Verschmutzte Disketten gehören in den Abfall, Reinigungsversuche mit Verdünner oder speziellen Mittelchen sind nicht zu empfehlen. Die Magnetschicht wird beim Betrieb im Laufwerk durch ein Reinigungsvlies, das innen an der Hülle befestigt ist, ohnehin ständig gesäubert.

Disketten darf man niemals falten, knicken oder mit schweren Gegenständen belasten. Starke Hitze oder pralle Sonneneinstrahlung kann die Funktion ebenso beeinflussen wie die Einwirkung äußerer Magnetfelder. Vorsicht vor allem bei Büroschränken mit Magnetverschlüssen! Daß ein Bekleben oder Beschreiben der Informationsschicht »Gift« ist, bedarf eigentlich gar keiner besonderen Betonung. Und zur Aufbewahrung von Disketten eignen sich staubdichte Plastikboxen besonders gut. Einige Testfabrikate kann man bereits in ziemlich stabilen Aufstellboxen kaufen. Auch die oftmals mitgelieferten Stülpschachteln lassen sich recht gut wiederverwenden – und sei es nur für Transport- oder Versandzwecke.

Unser Rat

Datensicherheit ist (meist) keine Frage des Preises. Schon für knapp 15 Mark bekommt man eine Zehnerpackung »guter« Markendisketten ohne technische oder elektromagnetische Schwächen: Quelle Privileg macht's möglich.

Noch mehr Qualität für wenig Geld bietet die Boeder Disky, die ein »sehr gutes« Gesamturteil erreichte. Auch die Diskette von Fuji und die Verbatim Optima glänzten mit der Bestnote. Gewaltig sind mitunter die Preisspannen beim gleichen Produkt. Zehn Exemplare der BASF Flexy Disk zum Beispiel gibt es schon für 35 Mark, sie können aber auch mehr als das Doppelte (76 Mark) kosten.

Zwar bieten die meisten Disketten gute Qualität, aber vor einem Hard- oder Software-Defekt kann man nie sicher sein. Deshalb: Speichern Sie wichtige Daten auf verschiedenen Disketten und lagern Sie diese getrennt!

Ausgewählt, geprüft, bewertet

Im Test: 20 Disketten für Personal Computer (5¼ Zoll, zweiseitig beschreibbar, doppelte Speicherdichte, 48 tpi, softsektoriert), darunter eine Produktgleichheit. Einkauf der Prüfmuster: Februar 1987.

■ Preise
Überregionale Handelserhebung im Juni 1987. Beim mittleren Preis handelt es sich um den Median, bei weniger als fünf Verkaufsstellen um das arithmetische Mittel. Katalogpreise, einheitliche Preise bei Eigenmarken des Handels, Festpreise und Preise laut Anbieter werden ohne Preisspanne angegeben.

■ Bewertung

Technische Prüfung	40 %
Funktionsprüfung	60 %

■ Abwertung
Das test-Qualitätsurteil konnte nicht besser lauten als das Ergebnis der Funktionsprüfung. Das Urteil Funktionsprüfung konnte nicht besser sein als die Elektromagnetischen Eigenschaften. Bei einem »Mangelhaft« in der Formatierprüfung wurde die Funktionsprüfung um eine Stufe abgewertet. »Mangelhafte« Urteile in der Abmessungsprüfung, bei der Verarbeitung der Magnetplatte oder in der Temperaturprüfung hatten eine Abwertung der Technischen Prüfung um eine Stufe zur Folge. Bei einer »zufriedenstellenden« Verarbeitung der Magnetplatte konnte die Technische Prüfung bestenfalls »gut« sein.

■ Technische Prüfung
Die Außenabmessungen wurden mit Hilfe einer Fallehre nach ISO 7487/1-1985 entsprechend DIN 66 247 Teil 1/1.85 geprüft. Die Dauerprüfung umfaßte 3 Millionen Umläufe mit fest eingestelltem Schreib-Lesekopf auf Spur 10. Vergleich der Lesespannung vor und nach dem Test – auch auf den Nachbarspuren. Temperaturprüfung: Lagerung der Disketten 72 Stunden lang bei +50/–3 °C und 80 % relativer Feuchte, Akklimatisierung bei +23/–2 °C und ca. 50 % relativer Feuchte. Anschließend wurden die Lesespannung und die Außenabmessungen überprüft.
Die Verarbeitung der Hülle ließen wir nach folgenden Kriterien untersuchen: Oberflächenstruktur (Bildung von Fingerabdrücken), Ausformung der Falzkanten, Ausführung des Verschlusses, Befestigung der inneren Gleitschicht (Vieseinlage). In einer Sichtprüfung wurde festgestellt, ob sich am Schreib/Leseschlitz Flusen von der Vlieseinlage zeigen. Unter der Rubrik Verarbeitung der Magnetplatte haben wir folgende Teilprüfungen zusammengefaßt: Ausführung der Politur (Bearbeitungsspuren), Grate des Verstärkungsringes, Rakelstreifen (Streifen auf der Folie vom Herstellungsprozeß).

■ Funktionsprüfung
Zu Beginn des Tests wurden sämtliche zur Verfügung stehenden Prüfmuster (90 Stück pro Fabrikat) im MS-DOS Format (40 Spuren, 16 Sektoren) formatiert. Formatierfehler wurden protokolliert und im Testbericht festgehalten.
Die elektromagnetischen Eigenschaften wurden mit Hilfe eines speziellen Diskettentesters (Media Logic 2000) ermittelt und mit einer Bezugsdiskette der Physikalisch Technischen Bundesanstalt (PTB) verglichen. Die folgenden Eigenschaften wurden geprüft: Signalausfall (drop out), Störsignal (drop in), Überschreibbarkeit (overwrite), Modulation, mittlere Lesespannungsamplitude (average signal amplitude) auf den Spuren 00 und 39 sowie die Auflösung (resolution) auf der Innenspur. Die Meßwerte wurden mit den Mindestanforderungen der European Computer Manufacturers Association (ECMA) verglichen und darüber hinaus vergleichend beurteilt.

■ Anbieter-Adressen
BASF Aktiengesellschaft Datentechnik-VID/MM, Postfach 51 48, 6800 Mannheim 1. – Döbbelin & Boeder GmbH, Wikkerer Str. 50, 6093 Flörsheim am Main. – 3M Deutschland GmbH, Postfach 10 04 22, 4040 Neuss 1. – Dysan: siehe Xidex. – Fuji Photo Film (Europe) GmbH, Postfach 32 04, 4000 Düsseldorf. – IBM Deutschland GmbH, Pascalstr. 100, 7000 Stuttgart 80. – Kaufhof AG, Leonhard-Tietz-Str. 1, 5000 Köln 1. – Maxell Europe GmbH, Postfach 11 07 48, 4000 Düsseldorf 11. – Memorex GmbH, Postfach 71 06 53, 6000 Frankfurt/M. 71. – Nashua Copygraph GmbH, Postfach 43 40, 3000 Hannover 1. – Quelle International, 8510 Fürth 500. – RPS: Rhône-Poulenc Systeme GmbH, Emil-von-Behring-Str. 7–9, 6057 Dietzenbach 1. – Scotch: siehe 3M. – Sentinel Computer Products Europe n. V., Bodenstraat 12, B-3830 Wellen/Belgien. – Verbatim GmbH, Frankfurter Str. 63–69, 6236 Eschborn. – Xidex GmbH, Frankfurter Str. 100, 6236 Eschborn.

Zwischen Dienst und Feierabend- wie Wochenendhäuslesbauen ist es uns Provinzlern doch tatsächlich möglich,sich zu einen samstäglichen Einkaufstag in der Großstadt aufzuraffen.
Er unterscheidet sich m.E. von Üblichen, weil unsere Computerfreaks auch außer Haus nicht aufhören können Ihre Sprache zu sprechen, zum Bleistift:
Wir stehen vor einem Tisch mit irgendwelchen Skiunterhemden, mit Reizverschluß, die verschiedene Farben und Muster aufweisen. Nach begeisterten Gewühl meinerseits vernimmt man plötzlich "Du Spatzel vorne hängt ein Demo", sprich ein ausgepacktes Hemd liebevoll auf einen Bügel gehängt.
Nach diversen Einkäufen von weltlichen Dingen , wie emailierter Pfannen, Bratäpfelförmen usw. landet man unausweichlich in der Computerabteilung.
Man benötigt, wenn schon sonst Nichts, Disketten,sind solche nicht noname und günstig im Angebot greift man natürlich gleich doppelt zu,in Form von singel seidet dabel dänzidi (was nichts Anderes heißt als 2-seitig doppelte Dichte, aber daß dürfte den Wenigsten unter dem weiblen Geschlecht verborgen geblieben sein).
Super findets Mausebär dann noch wenn PD (Pablik Domein)zur Verfügung steht, wie in unserem Fall. Geduldig wühle ichin der Diskettenbox und suche für Ihn jene welche heraus, die er nach literarischen Übersicht für gut befunden hat. Nun denn- nichts wie ran an die Kiste und zupf-wird Diskette für Diskette kopiert.
Aber damit nicht genug, werden die gezupften wieder in die Box einsortiert so findet sich welch Jubel noch manches überaus interessante Diskettchen (3,5). Diese können ungehindert durch vorsorglichen Hamsterkauf kopiert werden.
Spätestensjetzt verabschiede ich mich.in der Hoffnung Abwechslung zu finden. Während ich das Schreibwarengeschäft durchkämme und all den Dingen mehr oder weniger Beachtung schenke, fällt meine Wahl auf zwei schnucklige Tüten (Verpackungsmaterial) die ich dezent wieder zurücklege, weil es die Menschentraube vor der Kasse es sicherlich nicht zuläßt meinen Geldbeutel vor einen halben Stunde zu zücken.
Irgendwann besinnt sich jeder seiner besseren Hälfte wieder, welche wohl in der Compterabteilung kleben bleibt. Nichts wie durchgewühlt zwischen Gesundheitsbewußten umhüllt von Knoblauchduft, glitzernden Kugeln u. baumelnden Engelchen,rauf zu Ihm.
Welch Wunder er stand nicht mehr an "seiner" Kiste, dafür ein bißchen weiter hinten. Mäusebär hat Gleichgesinnten entdeckt, welcher großzügiger Weise das Neueste zupfen läßt.
Ich beschränke mich erquickender Weise aufs Beobachten von Leuten, da er mit verklärtem Blick u. innerlicher Wonne, ja doch nicht gänzlich von meiner Anwesenheit beglückt ist.
Meine Augen fixieren einen Softie, wohl volljährig, begleitet von Mama. Er begnügt sich etwas unsicher mit der Begutachtung von elektrischen Schreibmaschinen. Mit gespreiztem Fingerchen wird später sogar ausprobiert, während seine Mama als Auswahlkriterium Nr. 1 die Anschlaglautstärke auserkoren hat.

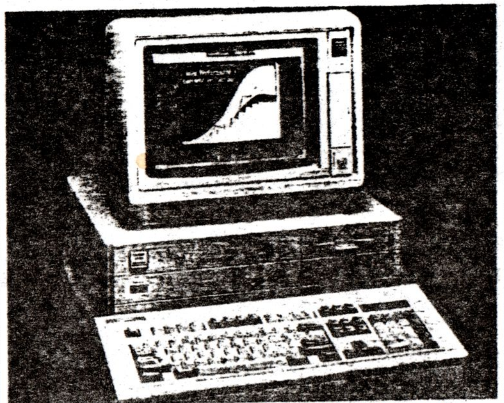
Später - viel später bekomme ich ein Küsschen auf die linke Wange, und Mausebär fragt an,was Spatzel noch kaufen will.
Immerhin ein schnuckliges Ende von einem von vielen Besuchen in ähnlichen Compterabteilungen.

Drucker ohne Laser - mit Adler Vorkriegsmodell - Spatzel vom Mausebär

Tandy 4000

32 BIT ZUM NIEDRIGPREIS

Einen Personal-Computer mit dem Intel 80386 hat jetzt auch Tandy vorgestellt. Das Gerät ist ein echter Preisknüller.



32-bit-Technik von Tandy: das Modell 4000; wenn die Zahlen stimmen – ein Preisknüller

So ganz vorn steht der amerikanische PC-Hersteller Tandy nicht in der öffentlichen Aufmerksamkeit. In seinem Heimatland USA behauptet er sich jedoch seit den Pioniertagen der Branche beständig auf einen der vorderen Plätze.

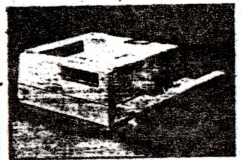
Dazu verhalfen ihm zunächst einmal seine eigene Ladenkette. Hinzu kommt noch, daß die Tandy-Produkte auch stets relativ kostengünstig, wenn auch nicht zu absoluten Tiefpreisen zu haben waren.

Jetzt hat sich das Unternehmen offenbar entschlossen, eine echte Preisbombe platzen zu lassen. Und das geschah zugleich mit dem Eintritt in den 32-bit-Club – mit dem Modell 4000 auf der Basis des Prozessores Intel 80386.

Bei Redaktionsschluß war ein deutscher Preis für die Maschine noch nicht bekannt. In den USA kostet das Gerät mit einer 20-MByte-Festplatte 3500 Dollar. Zum derzeitigen Umrechnungskurs entspricht das etwa 7500 Mark. Der Hersteller ist bekannt dafür, daß er seine amerikanischen Preise einigermaßen getreulich umsetzt, so daß hierzulande mit dem erwähnten Betrag zu rechnen ist.

Äußerlich sieht der neue Computer recht herkömmlich aus. Er hat die Tandy-typischen rundlichen Formen und ist auch nicht extrem kompakt – allerdings auch lange nicht so wuchtig wie der originale PC/AT.

Bei der Diskette hat sich der Hersteller von vornherein für das modernere



HP-kompatibel: Der Laserducker LP 1000

Erster MS-DOS-Handheld von Tandy: Der 1400 LT

$3\frac{1}{2}$-Zoll-Format entschieden. Es lassen sich aber auch andere Laufwerke installieren – etwa ein $5\frac{1}{4}$-Zoll-Laufwerk, aber auch ein Bandlaufwerk. Dafür gibt es praktischerweise drei Montage-Positionen.

Bei der internen Konstruktion verzichtete der Hersteller ebenfalls darauf, einem aktuellen technischen Trend zu folgen: Anders als bei manchen neuen Rechnern sind beim Tandy 4000 die unverzichtbaren Schnittstellen (je eine parallel und seriell) und auch die Videofunktionen nicht auf der Hauptplatine integriert.

Allerdings tun hier auch schon ASICs (application specific integrated Circuits, anwendungsspezifische integrierte Schaltkreise) ihren Dienst. Acht dieser maßgeschneiderten Chips ersetzen eine Vielzahl herkömmlicher Bauteile und verbessern gleichzeitig die Zuverlässigkeit der Maschine.

Der Arbeitsspeicher des Rechners läßt sich mit 1-Megabit-RAM-Chips auf der Hauptplatine bis acht Megabyte ausbauen. Mit einer Zusatzplatine in dem speziellen Steckplatz erreicht sie bis zu 16 MByte. Verwendet man die derzeit noch eher gebräuchlichen 256-Kbit-Chips, sind es zwei beziehungsweise vier MByte. Leider gab es vorerst noch keine Informationen darüber, wie der Arbeitsspeicher organisiert ist, so daß man über seine Leistungen im Vergleich mit anderen 80386-Maschinen noch nichts sagen kann.

Die übrigen Merkmale des Tandy 4000 sind eher konventionell zu nennen: Die Taktfrequenz beträgt 16 MHz, und es gibt vorerst wahlweise zwei Festplatten mit 20 und 40 MByte bei 65 beziehungsweise 28 Millisekunden mittlerer Zugriffszeit.

Bei dem optionalen Arithmetik-Prozessor für rechenintensive Anwendungen entschied man sich für den 80287. Er leistet zwar nicht so viel wie der 80387, ist aber eher verfügbar und nicht zuletzt auch billiger.

Überhaupt fällt der Tandy 4000 nicht durch außergewöhnliche Leistungen auf. Er rückt aber die Technik der leistungsstarken 32-bit-Personal-Computer für viele erstmals in erreichbare Nähe – zu seinem Preis gab es vor nicht allzulanger Zeit gerade einen normalen IBM PC/XT mit 8088-Prozessor.

Im übrigen hat Tandy zugleich mit der Vorstellung des starken PC-Modells seine Produktpalette auch an anderen Stellen kräftig erweitert. Neu bei Tandy ist ein zum IBM PC kompatibler Laptop-Computer. Das Modell 1400 LT bietet zwei oberhalb der Tastatur eingebaute Diskettenlaufwerke ($3\frac{1}{2}$ Zoll, 720 KByte) und einen beleuchteten Flüssigkristall-Bildschirm mit 640 mal 200 Bildpunkten und Graustufen.

Als Prozessor ist der (8088-kompatible) V20 von NEC eingebaut, der mit 7,16 oder 4,77 MHz läuft. Der Arbeitsspeicher ist 768 KByte groß. Davon dienen 128 KByte als RAM-Diskette oder als Drucker-Puffer. Das Gerät kostet in den USA rund 1600 Dollar, entsprechend knapp 3400 Mark. Es kann laut Hersteller mit einer Batterieladung rund vier Stunden lang arbeiten.

Zu den neuen Produkten zählt neben zwei Nadeldrucker-Modellen auch der Laserdrucker LP 1000. Dieses Gerät bietet die für die untere Leistungsklasse üblichen Merkmale: eine Auflösung von 300 mal 300 Punkten pro Zoll und ein (mechanisches) Drucktempo von sechs Seiten je Sekunde.

Das Gerät kann wie ein Laserjet Plus von Hewlett-Packard arbeiten. Der interne Arbeitsspeicher von 1,5 MByte erlaubt den Druck von Grafik.

Mit besseren Leistungsmerkmalen hat Tandy unter anderem sein PC/AT-kompatibles Modell 3000 ausgestattet. Die Maschine läuft jetzt mit 12 MHz (umschaltbar auf 6 MHz) und bietet als Grundausstattung 640 statt 512-KByte-RAM. Außerdem gibt es neuerdings ebenfalls drei Laufwerk-Plätze für Diskettenstationen verschiedenen Formats, Festplatte und Bandlaufwerk. Der Grundpreis der Maschine ist geringfügig gesunken (in den USA von 2200 auf 2000 Dollar). *-st*

Technische Daten

Tandy 4000

Prozessor: Intel 80386, 16 MHz Taktfrequenz, wahlweise Arithmetik-Prozessor 80287 (8 MHz)
Arbeitsspeicher: 1 MByte, ausbaubar bis 16 MByte
Massenspeicher: eine $3\frac{1}{2}$-Zoll-Diskette, 1,44 MByte, wahlweise 20- oder 40-MByte-Festplatte
Bildschirm: 12 Zoll (30,5 cm) monochrom oder 14 Zoll (35,5 cm) farbig
Schnittstellen: 1 × seriell, 1 × parallel (Steckplatine)
Steckplätze: 6 im PC/AT-Format, 2 im PC/XT-Format (davon 1 belegt), 1 32-bit-Steckplatz
Abmessungen (Länge × Breite × Höhe): 45,7 cm × 48,3 cm × 16,5 cm
Software: Betriebssysteme MS-DOS 3.2, XENIX System V, künftig auch OS/2

Neues von der Diskothek

Wie auf dem Vorstandstreffen vereinbart, hat Hartmut 3 Public Domain Disketten für das Model3 und 4 angeschafft. Da diese Disketten auf meiner Anlage nicht lesbar sind kann ich hier nur eine kurze übersicht und keine Programmbeschreibungen geben, vielleicht holt dies einer von Euch mal nach.

Nr.	Name	Titel
	UTILITY-Programme für das Model4	
Disk22	M3TOM4/CMD	Convert
	PIXEL/ASC PXLDEMO/BAS	Pixel
	SORT4/ASC SRT4DEMO/BAS	Sortierprogramm
	MAD/CMD	Memory Adresses
	PF/FLT	Programmable Filter Keys
	RESTORE/CMD	
	BASREF/CMD	Crosscheck
	RUN4/BAS	Autorun
	HELPER/BAS	
	TIMER4/CMD	Disk Timer
	FIND10/CMD FIND11/CMD	Finder
	ZAP4/CMD	Disk Zapper
	ECI/CMD ECI/JCL	ECI Command
	LOCATOR/BAS	
	SPACEMAP/CMD	

Nr.	Name	Titel
	The Best of Model3	
	SINSTEP/CMD	Stepping Through Basic
	WORDCHKR/BAS	Letter Perfect
	WORD/BAS	Hinrichs' Word Processor
	EXEC/CMD	Brick by Brick
	SPOOL/CMD	April Spool
	MPH/CMD	Smart Talk
	DRAW/CMD	Drawing Boards
	TIMER3/BAS	Keeping Time
	PROJMIND/BAS	Time Keeper
	EASYDATA/BAS	Little Wonder
	NOVACALC/BAS	Novacalc
	GRAPH/CMD	Grade-A Graphics
	PATCHER/BAS	Patch Maker
	BLANKER/CMD	Blank Expressions
	COMMANDO/CMD	Macro Economics

Nr.	Name	Titel
	The Best of Model4	
	TLOAD/CMD	A Happy Medium
	SETUP/JCL	Extra Strength DOS
	ENHANCE/OVL	ScripAid
	MAKEDO/BAS	Command Performance
	GRAPHICS/CMD	Restored Art
	M3TOM4/CMD	Convert
	RUN4/BAS	Autorun
	LIFE/CMD	Life
	WD/CMD	Windows
	MICROTAB/BAS	MicroTab
	PROJMIND/BAS	Project Minder
	MAIN/BAS	Basic Data Base System

Das war's von der Club-Bibliothek,

Werner Förster

Gutes ist zu berichten von der Clubbücherei! Zwar ist die Ausleihrate immer noch sehr niedrig, dafür wächst die Bibliothek aber munter weiter.

Das ist nicht zuletzt auf unser ehemaliges Mitglied Josef Konrad zurückzuführen, der dem Club nun auch noch das Buch "Einführung in 'C'" geschenkt hat. Ich möchte mich an dieser Stelle nochmals sehr herzlich für diese Spende bedanken!

Auch zwei sehr wichtige Bücher aus der Reihe "and other Mysteries", die uns beim Clubtreffen 86 abhanden gekommen waren, konnten zu einem recht günstigen Preis wiederbeschafft werden. Thomas Buskowiak, der sein Model 4p verkauft hat, war so freundlich uns hierbei zu helfen! Aus der gleichen Quelle stammt auch das Handbuch des CP/M 2.2-Betriebssystems, welches neu in unserer Sammlung ist.

Mit Gruß an alle Leseratten, Euer

Hartmut Obermann

Nr. 0018: Basic Disk I/O Faster And Better Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips und Tricks zum DOS, zur DISK, zu Files etc. zeigt (mit Programm-Diskette zum Buch).

Nr. 0019: Basic Faster And Better Other Mysteries

Lewis Rosenfelder --- IJG Inc., 1953 West, USA

Ein sehr gutes (englisches) Buch, das viele Tips, Tricks und Routinen in BASIC bietet.

Nr. 49 : Some Common BASIC Programs

Poole, Borchers, Koessel --- Osborne/McGraw-Hill

Mehr als 70 BASIC-Programme aus den verschiedensten Bereichen!

Nr. 50 : Das Handbuch des CP/M 2.2-Betriebssystems

--- Markt Technik

Originalhandbuch von Digital Research in englischer Sprache!

Nr. 51 : Der Einstieg in C

Paul Chirilan --- Markt Technik

Einfhrung in die Programmiersprache C anhand von ber 70 praxisnahen Beispielen!

Buchbesprechung

Dr. Eberhard Zehendner: *"Das Z80-Buch"*
Markt & Technik Verlag, DM 59.--
ISBN 3-89090-219-7

Das Werk präsentiert sich mit stolzen 682 Seiten und einem nicht weniger umfangreichen wie gewichtigen Inhalt. Das Kapitelverzeichnis am Anfang enthält satte 29 Punkte. Es beginnt mit einer kurzen Beschreibung des Z80 und geht dann zielstrebig ins Eingemachte wie Listenverarbeitung, Bitfelder, Integer- und Realarithmetik, Bearbeitung von Zeichenketten, Mengen und Verbunde etc. Alle Themen sind in Assembler ausgeführt. Hinzu kommen noch Übungsbeispiele, deren Lösungen am Ende aufgeführt und soweit notwendig, noch zusätzlich erläutert sind. Es wird endlich einmal nicht auf das Buch vom Herrn XYZ verwiesen, wo man näheres nachlesen kann und deshalb an dieser Stelle nur das Prinzip gezeigt wird. Die Assembler-Routinen sind keine 0815 Standardlösungen. Der Autor hat sich zu den beschriebenen Problemen durchaus seine Gedanken gemacht und recht raffinierte Lösungen gefunden. Aus meiner Sicht eine sehr gute Bereicherung der eigenen oder auch Club-Bibliothek, trotz des hohen Preises.

Tschüß Kurt

Joepgen, Turbo-Pascal, Das Kompendium für die Programmierpraxis
(Hanser Verlag, DM 58)

Ein ganz hervorragendes Werk für fortgeschrittene Turbo-Pascal Freunde. Grundbegriffe werden kurz angesprochen, trainiert werden aber hauptsächlich die Anwendungen von Turbo-Pascal, die sich mit anderen Programmiersprachen meines Wissens gar nicht oder nur schwer verwirklichen lassen wie z.B. Manipulationen mit Texten. Jede Anwendung ist mit einem Übungsprogramm versehen, sodaß das Gelernte sofort ausprobiert werden kann. Alle Programme sind auf drei Disketten zu haben, nur der Kostenpunkt von je DM 96 bzw. DM 258 für alle scheint mir persönlich viel zu hoch.

Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal, (Interest-Verlag DM 92)

Bei diesem Werk handelt es sich um eine Loseblattsammlung in einem DIN A 4 Ordner. Neben einer Einführung in das Turbo-Pascal System finden wir Hinweise zur Softwareerstellung, den ersten Teil eines Pascal-Kurses zum Selbststudium und eine Reihe von fertigen Programmen aus Wissenschaft, Technik, Finanzen etc.,sowie Module für Utilities. Leider ist alles auf MS-DOS angestimmt. Ca. alle drei Monate gibt's Ergänzugslieferungen. Allerdings scheint mir der Preis von DM 92 sowie der Seitenpreis von 35 Pfg. für die Nachträge jenseits der Schmerzgrenze.

Hans-Martin [illegible]

Ein Buch für den Handwerker

Gerald stellt in der 21. Ausgabe "Die Werkzeuge eines Assembler-Programmierers" vor. Wenn man Training als ein Werkzeug verstehen will, dann hat er eins vergessen. Der Z80 frißt dem Crack nur dann aus der Hand, wenn er am Ball bleibt, wann immer sich die Gelegenheit bietet.

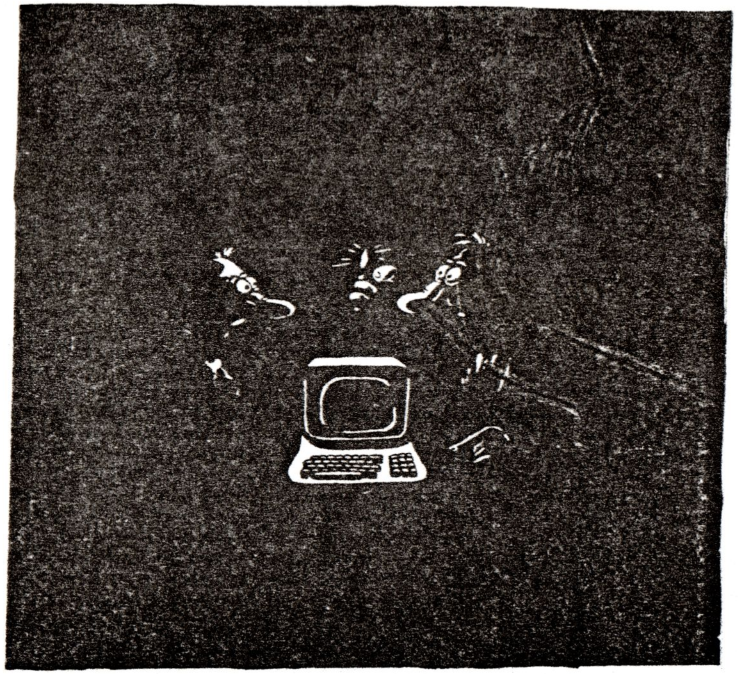
Aber aus einem anderen Grund beziehe ich mich auf Geralds Beitrag. Da gibt es nämlich durchaus ein Buch, das nahezu alle Forderungen erfüllt, die er an die Paperware stellt. Es ist "Z80 - Einführung und Programmierung" von Nichols, Nichols, Ray, erschienen im Eletor-Verlag. Es hat nur drei Nachteile:

1. Es kostet DM 35,- obwohl man es vom Umfang her nur als Taschenbuch bezeichnen kann.
2. Es enthält alle Angaben zu den Z80-Befehlen nur in Hex, obgleich für den fortgeschrittenen Programmierer die oktale Darstellung wünschenswert wäre.
3. Die wirklich sinnvolle ASCII-Tabelle enthält es auch nicht. Man kann sie sich selber erstellen, wenn mal eine Stunde frei ist. (Dem wirklichen Crack ist auch diese Stunde kaum je vergönnt.)

Das Buch ist als Anleitung für den Nanocomputer NBZ80 geschrieben, bezieht sich jedoch nur in wenigen Passagen auf dieses Gerät. Der ganze Rest ist allgemeine Z80-Knowledge vom Feinsten, angereichert mit Arbeitsaufgaben an den Lernenden. Im Anhang finden sich üppige Tabellen mit allem, was auch der Profi nicht auswendig weiß. Bloß eben nicht oktal.

Arnulf Sopp

Schließ doch den Computer an und frag ihn, wo der Lichtschalter ist . . .



Der sehr aufmerksame Leser wird vielleicht bemerken, daß das folgende Buch in PASCAL 1/87 bereits reviewt wurde. Dennoch und vor allem weil der Strom der Anfragen nach einem Buch für »Einsteiger« nicht abreißt, hier nochmals das wirklich für jeden geeignete Werk:

Sprachen für die Programmierung: Eine Übersicht
von Jochen Ludewig
Bibliographisches Institut, Mannheim, 1985,
[Reihe B.I. Hochschultaschenbücher, Band 622],
194 Seiten,
Preis: DM 22,80
ISBN 3-411-00622-6

Der Autor ist Software-Spezialist und hat es unternommen, für eine Unterrichtsreihe der Erwachsenenbildung, einen Überblick über die verschiedenen gängigen Programmiersprachen zusammenzustellen.

Ein weit reichender Blick von oben

Daraus ist das vorliegende Buch entstanden. Es sollte damit kein umfassendes Lehrwerk für Informatiker geschaffen werden, sondern ein populäres »Aufklärungsbuch« für alle, die mit Primitiv-Sprachen (Zitat) arbeiten, um diese zur Beschäftigung mit höheren Programmiersprachen anzuregen.

Behandelt werden - mehr oder weniger ausführlich - die Sprachen Assembler, FORTRAN, COBOL, PL/1, die ALGOL-Familie, Pascal, Modula, C, BASIC, LISP, LOGO, Smalltalk und Prolog, sowie am Ende eine Klassifikation von Programmiersprachen, die an eine Einteilung nach Chomsky angelehnt ist.

Betont wird durchgängig der Einfluß, den die Wahl der verwendeten Sprache auf die eingesetzten Programmiertechniken und Programmierfehler hat.

Der Stil der Darstellung ist unverkrampft und eingängig. Der Band enthält zudem zahlreiche Abbildungen, die sowohl informativ sind, als auch den Text auflockern.

Als besonderen Witz stellt der Autor jeder Sprache als Allegorie ein bestimmtes Transportmittel gegenüber: der Sprache C z.B. einen Land-Rover »viel Flexibilität speziell für bodennahe Aufgaben«, der Sprache LISP einen Helikopter »schwer zu handhaben, aber auf speziellen Gebieten unschlagbar«.

Das Erscheinen dieses Buches in der Reihe der Hochschultaschenbücher könnte den allgemein interessierten Leser vielleicht von der Lektüre abhalten, weil er ein akademisch ausgerichtetes Buch erwartet.

Dies ist aber ganz und gar nicht der Fall! Der Band sei allen Programmierern im Hobby und im Beruf warm empfohlen: der Lohn der Lektüre ist ein Überblick der wichtigsten eingesetzten Sprachen heute und eine Entscheidungshilfe für die Beurteilung neuer Sprachen morgen.

(Wolfgang J. Weber)

Das COMAL-80 Buch
von B. R. Christensen
Oldenbourg Verlag, München 1986,
269 Seiten,
Preis: DM 29,80
ISBN 3-486-20208-2

Ein Buch für Einsteiger

Das Buch enthält eine einführende Darstellung der Programmiersprache COMAL-80, wobei auf Seiten des Lesers keinerlei Vorkenntnisse über den Umgang mit Computern und das Programmieren vorausgesetzt werden.

Die Erläuterungen des Autors sind extrem verständlich, detailliert und genau: er führt Schritt für Schritt in den Gebrauch der Sprache auf dem Computer C-64 (Commodore) ein.

Es handelt sich um ein Lehrbuch für den Anfänger und nicht, wie es der Titel vielleicht vermuten ließe, um eine umfassende Sprachbeschreibung von COMAL-80.

Der Aufbau des Buches ist gut durchdacht.

Alle grundlegenden Merkmale und Befehle der Sprache werden behandelt, jedoch folgt die Darstellung keiner strengen technischen Systematik, sondern der Logik des Lernens. Das heißt: Informationen werden dort gegeben, wo sie gebraucht werden, und auch vor Wiederholungen scheut sich der Autor nicht.

Die Struktur der einzelnen Kapitel überzeugt durch die durchgängige Berücksichtigung der Bedürfnisse der Lernenden: am Anfang steht jeweils ein Überblick, in den Text sind zahlreiche Übungen eingebettet, und jedes Kapitel endet mit Aufgaben.

Die grafische Gestaltung des Typoskripts ist ansprechend und aufgelockert.

Da durchgängig auf das Gerät C-64 Bezug genommen wird, konnten auch Besonderheiten dieses Systems wie z.B. Sprites angesprochen werden.

Wer COMAL-80 auf anderen Geräten einsetzt, z.B. Schneider CPC-464 oder IBM PC), muß den Text an einigen Stellen (meist geringfügig) abändern.

COMAL ist eine ideale Sprache für Programmieranfänger, weil sie einen erstaunlich hohen Komfort bietet (Einfachheit, deutsche Systemmeldungen, hohe Geschwindigkeit bei der Programmausführung und vieles mehr) und dabei nicht den Weg zu höheren Sprachen wie Pascal oder C verbaut.

Das vorliegende liebevoll geschriebene Buch ist gerade für Programmieranfänger sehr gut geeignet; Leser mit Vorkenntnissen wären hingegen mit weniger breit angelegten Darstellungen besser bedient.

(Wolfgang J. Weber)

- IMPRESSUM -

1. Vorsitzende — Hartmut OBERMANN
Schwalbacher Straße 6
6209 Heidenrod 1
☎ 06124 /3913

2. Vorsitzende — Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2682

Hardwarekoordinator — Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar — Werner FÖRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion — Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck — Peter Spieß
Trugenhoferstraße 27
8859 Rennertshofen 1
☎ 08434 /454

Bankverbindung — des CLUB 80

Postgirokonto — Peter STEVENS, Sonderkonto CLUB 80
Konto-Nummer — 285 491 - 465
Postgiroamt — Dortmund
BLZ — 440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

SCHLUSS

Hallo Club-80er,

nun habe ich es doch noch geschafft, vor Weihnachten, mit dem neuesten INFO fertig zu werden, wenn es auch erst 1988 in Euren Briefkästen landen wird.

Somit kann ich mich schon jetzt recht herzlich für Eure rege Mitarbeit an den 87'er CLUBINFO's bedanken. Ohne Euren Fleiß beim Erstellen der Artikel, ausdenken und erproben der Hard- und Software, wären wir sicher nicht in der Lage so ein umfangreiches INFO zu bieten.
Also nochmals Dank, weiter so, und bitte nicht nachlassen.

Allen unseren CLUB-80-Mitgliedern wünsche ich (im nachhinein) ein frohes Weihnachtsfest (mit vielen brauchbaren Geschenken), und einen gelungenen Rutsch 1988 mit einer Wahnsinns-Silvester-Orgie.

Nun noch zu der aktuellen vor Euch liegenden INFO. Im Anhang daran findet Ihr eine Referenzkarte mir diversen Druckersteuerzeichen der gängigsten Drucker. Sicher ist sie eine Hilfe für Euch beim Umschreiben von Programmen mit Druckroutinen.
Weiterhin habe ich mal wieder eine Sonderpuplikation beigelegt. Unter dem Namen "Mikroelektronik" werden Euch wichtige und häufig gebrauchte Ausdrücke von A bis Z erklärt. Die Sonderpuplikation kann sicher einigen von Euch als kleines Nachschlagewerk dienen.

Ich wünsch Euch ein Frohes Fest und Guten Rutsch ins neue Jahr !
Bis zur neuesten INFO grüßt Euch

Jens

P.S.: Da ich aus den bekannten Zeitgründen nicht dazu komme mich persönlich, für die von Euch erhaltenen und noch kommenden Weihnachts- und Neujahrsgrüße, zu bedanken, möchte ich dies hiermit tun.

Stand: Dezember 1987

Bitte überprüft Eure Daten auf Richtigkeit und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907	// 0431 /74047
Betz	Heinrich	St. Wolfgangsstraße 13	8551	Hausen	09191 /31698	// 09191 /611108
Brandel	Hermann	Cäciliastraße 30	8439	Postbauer-Heng	09180/ 493	// 0911 /219-245
Braun	Harald	Postfach 8011	2300	Kiel 17	-	// -
Böcker	Dieter	Lehmweg 4	2930	Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441	Wettringen	05233 /4320	// 02557 /1236
Eilers	Hans-Joachim	Thomas-Dehler-Straße 6	2900	Oldenburg	0441 / 53239	// 0441 / 7983045
Emmerich	Helmut	Waldstraße 5	6682	Ottweiler	06824 /4114	// -
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721 /21841	// 09721 /51256
Gromottka	Uwe	Lange Reihe 40	2803	Weyhe	0421 / 80496	// 0421 /300-6870
Hartmann	Hans-Günter	Möwenstraße 9	2876	Berne 2	04406 / 1911	// 0421 / 2483405
Heidenreich	Ulrich	Werderstraße 35	4300	Essen 1	0201 / 282770	// -
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563	// 0911 /2195245
Hentz	Werner	Am Trankgarten 20	6457	Maintal 2	06109 /66625	// 0691 /583639
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024	//
Hill	Peter	Bergstraße 65	6754	Otterberg	-	// -
Jablotschkin	Rainer	Thiekamp 29	4780	Lippstadt 8	02948 /542	// 02921 /70431
Krispin	Michael	Schwanstraße 8	4130	Moers 3	02841 /73690	// -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417	// 09171 /832665
Lachmann	Wolfgang	Am Ringofen 11	6407	Schlitz	06642 /6940	// 069 /832051
Loose	Gerhard	Viefhaushof 42	4306	Essen 13	0201 /212608	// -
Mand	Harald	Kleinflintbeker Straße 7	2302	Flintbek bei Kiel	04347 /3629	// 0431 /3013580
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083	// 04151 /8891-37
Neueder	Jens	Panoramastraße 21	7178	Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913	// -
Obscherningkat	Helmut	1 RUE DES BRUYERES	F-68360	Soultz	0033089/762690	// -
Perschbach	Patrick	Waldstr. 52	5000	Koeln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /43791	//
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551	// 04103 /605310
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383	// 0203 /331383
Schmid	Alexander	St. Cajetan-Straße 38/VII	8000	München 80	089 /495326	// -
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000	Köln 71	0221 /707044	// -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631	// 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228 /643853	// -
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859	Rennertshofen 1	08434 /454	// 08431 /7041604
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585	// 05401 /30096
Stevens	Peter	Danziger Straße 11	5042	Erfstadt	02235 /42350	// -
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153 /1564	// -
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131 /32860	// 06131 /395268
Trapp	Harald	Kranichstr. 46	4270	Dorsten 1	02362 /42497	// 02362 /23127
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285	//
Wacker	Fred	Postfach 2246	7550	Rasstatt	07222 /52574	// -
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361	// -
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686	// -

Das DOSsier ist eine Extra-Leistung von DOS International. Hier finden Sie in übersichtlicher tabellarischer Form wissenswerte Informationen rund um Ihren PC und seine Peripherie. Die Seiten sind zum Ausschneiden oder Fotokopieren gedacht. So erhalten Sie mit der Zeit ein einzigartiges Archiv von geballten Informationen zu Ihrem Computersystem.

Druckersteuercodes auf einen Blick

Verschiedene Drucker benutzen unterschiedliche Steuercodes. Gleiche Funktionen funktionieren somit auf dem einen Drucker problemlos, während sie auf dem nächsten nicht zu finden sind. Das heutige DOSsier gibt Ihnen den Überblick in diesem Gewirr.

(Nils Teller/hg)

Druckermodus-Steuerung	ESC P-Norm	Dezimal	Epson FX 80	Fujitsu DX 2100	Citizen 120 D	Star SG 10	SG 10 IBM-Mode
Summer (Bell)	BEL	7	BEL	BEL	BEL	BEL	BEL
Rückschritt (Backspace)	BS	8	BS	BS	BS	BS	BS
Drucker initialisieren	ESC @	27, 64	ESC @	ESC @	ESC @	ESC @	ESC @
Papierendeerkennung aus	ESC 8	27, 56	ESC 8	ESC 8	ESC 8	ESC 8	ESC 8
Papierendeerkennung ein	ESC 9	27, 57	ESC 9	ESC 9	ESC 9	ESC 9	ESC 9
Druckkopf an den Zeilenanfang	ESC <	27, 60	ESC <	ESC <	ESC <	ESC <	ESC <
Druckwegoptimierung (Toggle)	ESC U	27, 85	ESC U n	ESC U n	ESC U n	ESC U n	ESC U n
Druckerpuffer löschen	CAN	24	CAN	CAN	CAN	CAN	CAN
letztes Zeichen im Puffer löschen	DEL	127	DEL	DEL	DEL	DEL	DEL
Drucker selektieren	DC 1	17	DC 1	DC 1	DC 1	DC 1	DC 1
Drucker inaktivieren	DC 3	19	DC 3	DC 3	DC 3	DC 3	DC 3
höchstwertige Bit (MSB) auf 1 setzen	ESC >	27, 62	ESC >	ESC >	ESC >	ESC >	ESC >
höchstwertige Bit (MSB) auf 0 setzen	ESC =	27, 61	ESC =	ESC =	ESC =	ESC =	ESC =
MSB-Modus löschen	ESC #	27, 35	ESC #	ESC #	ESC #	ESC #	ESC #
Druckqualität							
Draft und NLQ (Toggle)	ESC x	27, 120	n.v.	ESC % (2) (0)	ESC x n	ESC B 4	ESC 4 n
Schriftart							
Pica (Normal)	ESC P	27, 80	ESC P	ESC P	ESC P	ESC B 1	ESC P
Elite	ESC M	27, 77	ESC M	ESC M	ESC M	ESC B 2	ESC M
Proportional (Toggle)	ESC p	27, 112	ESC p	ESC p n	ESC p n	ESC p n	ESC p n
Schriftformen							
Breitschrift für die gegenwärtige Zeile	SO	14	SO	SO	SO	SO	SO
Breitschrift (SO) löschen	DC 4	20	DC 4	DC 4	DC 4	DC 4	DC 4
Breitschrift, unbegrenzt	ESC SO	27, 14	ESC SO	ESC SO	ESC SO	ESC SO	ESC SO
Breitschrift (Toggle)	ESC W	27, 87	ESC W n	ESC W n	ESC W n	ESC W n	ESC W n
Schmalschrift (Condensed) für eine Zeile	SI	15	SI	SI	SI	SI	SI
Schmalschrift, unbegrenzt	ESC SI	27, 15	ESC SI	ESC SI	ESC SI	ESC SI	ESC SI
Schmalschrift löschen	DC 2	18	DC 2	DC 2	DC 2	DC 2	DC 2
Fettschrift	ESC E	27, 69	ESC E	ESC E	ESC E	ESC E	ESC E
Fettschrift löschen	ESC F	27, 70	ESC F	ESC F	ESC F	ESC F	ESC F
doppelter Anschlag	ESC G	27, 71	ESC G	ESC G	ESC G	ESC G	ESC G
doppelter Anschlag löschen	ESC H	27, 72	ESC H	ESC H	ESC H	ESC H	ESC H
Kursivschrift	ESC 4	27, 52	ESC 4	ESC 4	ESC 4	ESC 4	ESC I 1
Kursivschrift löschen	ESC 5	27, 53	ESC 5	ESC 5	ESC 5	ESC 5	ESC I 0
Unterstreichen (Toggle)	ESC -	27, 45	ESC - n	ESC - n	ESC - n	ESC - n	ESC - n
Hochstellen	ESC S 0	27, 83, 0	ESC S 0	ESC S 0	ESC S 0	ESC S 0	ESC S 0
Tiefstellen	ESC S 1	27, 83, 1	ESC S 1	ESC S 1	ESC S 1	ESC S 1	ESC S 1
Hoch- / Tiefstellen löschen	ESC T	27, 84	ESC T	ESC T	ESC T	ESC T	ESC T
Wahl mehrerer Druckarten (Makro)	ESC !	27, 33	ESC ! n	ESC ! n	ESC ~ 3 n	ESC ? n	ESC ! n
Zeichensätze							
internationale Zeichensätze wählen	ESC R	27, 82	ESC R n	ESC R n	ESC R n	ESC 7 n	ESC R n
IBM-Zeichensatz 1	ESC 7	27, 55	ESC 7	ESC 7	ESC 7	n.v.	ESC 7
IBM-Zeichensatz 2	ESC 6	27, 54	ESC 6	ESC 6	ESC 6	n.v.	ESC 6
ladbarer Zeichensatz (Toggle)	ESC %	27, 37	ESC %	ESC %	ESC %	ESC %	ESC $ n
benutzerdefinierte Zeichen	ESC &	27, 38	ESC & n1..nm	ESC & n1..nm	ESC & n1..nm	ESC * n1..nm	ESC & n1..nm
ROM-Zeichensatz in Zeichengenerator übertragen	ESC : 0	27, 58, 48	ESC : 0	ESC : 0	ESC : 0	ESC * 0	ESC : 0
Grafik							
Standarddichte	ESC K	27, 75	ESC K	ESC K	ESC K	ESC K	ESC K
doppelte Dichte	ESC L	27, 76	ESC L	ESC L	ESC L	ESC L	ESC L
doppelte Dichte und doppelte Geschwindigkeit	ESC Y	27, 89	ESC Y	ESC Y	ESC Y	ESC Y	ESC Y
vierfache Dichte	ESC Z	27, 90	ESC Z	ESC Z	ESC Z	ESC Z	ESC Z
Makro-Modus	ESC *	27, 42	ESC *	ESC *	ESC *	ESC *	ESC g
9-Nadel-Modus	ESC ^	27, 94	ESC ^	ESC ^	ESC ^	n.v.	n.v.
Zeilenvorschub (einmalig)							
Wagenrücklauf (Carriage Return)	CR	13	CR	CR	CR	CR	CR
einfacher Zeilenvorschub (Line Feed)	LF	10	LF	LF	LF	LF	LF
Zeilenvorschub um n/216" (9 Nadeln)	ESC j	27, 106	ESC j n	ESC j n	n.v.	n.v.	n.v.
Zeilenvorschub um n/180" (24 Nadeln)	ESC J	27, 74	n.v.	ESC J n	n.v.	n.v.	ESC J n
absolute Steuerung (vertikal)							
1/8" Zeilenabstand	ESC 0	27, 48	ESC 0	ESC 0	ESC 0	ESC 0	ESC 0
7/72" Zeilenabstand	ESC 1	27, 49	ESC 1	ESC 1	ESC 1	ESC 1	ESC 1
1/6" Zeilenabstand	ESC 2	27, 50	ESC 2	ESC 2	ESC 2	ESC 2	ESC 2
n/216" Zeilenabstand	ESC 3	27, 51	ESC 3 n	ESC 3 n	ESC 3 n	n.v.	ESC 3 n
n/72" Zeilenabstand	ESC A	27, 65	ESC A n	ESC A n	ESC A n	ESC A n	ESC A n
Seitenvorschub (Form Feed)	FF	12	FF	FF	FF	FF	FF
Seitenlänge (Zeilen)	ESC C	27, 67	ESC C n	ESC C n	ESC C n	ESC C n	ESC C n
Seitenlänge (Zoll)	ESC C 0	27, 67, 0	ESC C 0	ESC C 0	ESC C 0	ESC C 0	ESC C 0
Überspringen der Perforation	ESC N	27, 78	ESC N n	ESC N n	ESC N n	ESC N n	ESC N n
Löschen von ESC N	ESC O	27, 79	ESC O	ESC O	ESC O	ESC O	ESC O
Vertikaler Tabulator	VT	11	VT	VT	VT	VT	VT
VT-Stops	ESC B	27, 66	ESC B n1..nm	ESC B n1..nm	ESC B n1..nm	ESC B O n1..nm	ESC B O n1..nm
VFU-Kanal wählen (Vertical Format Unit)	ESC /	27, 47	ESC / n	ESC / n	ESC / n	n.v.	n.v.
VFU-Stop setzen	ESC b	27, 98	ESC b n	ESC b n	ESC b n	ESC b n	ESC b n
absolute Steuerung (horizontal)							
rechten Rand setzen	ESC Q	27, 81	ESC Q n	ESC Q n	ESC Q n	ESC Q n	ESC Q n
linken Rand setzen	ESC 1	27, 49	ESC 1 n	ESC 1 n	ESC 1 n	ESC M n	ESC 1 n
Horizontaler Tabulator	HT	9	HT	HT	HT	HT	HT
HT-Stops	ESC D	27, 68	ESC D n1..nm	ESC D n1..nm	n.v.	ESC D n1..nm	ESC D n1..nm

Begriffe aus der

Mikroelektronik

Sonderheft

Redaktion CLUB 80

Dezember 1987

Begriffe aus der Mikroelektronik

Die Mikroelektronik ist heute zu einem Gegenstand des täglichen Lebens geworden. Begriffe wie *digital, Mikroprozessor* oder *Chip* (meist allerdings fälschlich für verkapselte integrierte Bauelemente gebraucht) gehören inzwischen zur Umgangssprache. In der folgenden Zusammenstellung werden die wichtigsten Begriffe aus Technologien* und Anwendungen der Mikroelektronik in alphabetischer Reihenfolge aufgelistet und erläutert. Der Leser hat damit die Möglichkeit, einen Begriff, dessen Bedeutung ihm unbekannt oder unklar ist, schnell zu finden und sich in wenigen Sätzen über ihn zu informieren. Tauchen innerhalb einer Erklärung Begriffe auf, die an anderer Stelle behandelt sind, so wird auf sie durch einen Pfeil (→) hingewiesen.

Die Zusammenstellung erhebt keinen Anspruch auf Vollständigkeit, umfaßt aber alle wichtigen und häufig gebrauchten Ausdrücke.

A

Abfallzeit
(fall time)

Maß für die Steilheit eines Impulsabfalls. Es ist die Zeit, in der ein Signal von 90 % auf 10 % seines Dachwertes abgefallen ist (Bild 1).

Äquivalenzschaltung, EXNOR-Schaltung
(exclusive NOR-gate)

Logische → Digitalschaltung, welche die Äquivalenz-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn beide Eingangssignale gleich sind, gleichgültig, ob sie beide 1 oder 0 sind (s. Bild 16). Mathematische Beschreibung: $X = (A \wedge B) \vee (\overline{A} \wedge \overline{B})$ (gelesen: X = A UND B ODER A NICHT UND B NICHT).

Addierer, Addierschaltung
(adder)

Schaltung zum Addieren zweier → Binärzahlen, im einfachsten Fall ein → Halbaddierer, der Summe und Übertrag der Addition zweier einstelliger Zahlen bildet. Ein → Volladdierer, der aus zwei Halbaddierern zusammengeschaltet werden kann, berücksichtigt bei der Addition zweier einstelliger Zahlen noch den Übertrag von der Addition der nächstniedrigen Stelle. Zum Addieren n-stelliger Zahler müssen entsprechend n Volladdierer verwendet werden.

Analog-Digital-Wandler, A/D-Wandler
(analog-to-digital-converter)

Schaltung zur Umformung eines analogen Eingangssignals in ein digitales Ausgangssignal. Die Zahl der Ausgangsbits bestimmt das Auflösungsvermögen, ein n-bit-Wandler kann 2^n unterschiedliche Signalwerte erfassen.

Anstiegsgeschwindigkeit
(slew rate)

Maß für die Ausgangsspannungsänderung eines Verstärkers, gemessen in Spannung pro Zeit (V/µs). Häufig als maximale (höchste) Anstiegsgeschwindigkeit angegeben, so daß aus diesem Wert berechnet werden kann, wieviel Zeit für eine bestimmte Spannungsänderung erforderlich ist oder welche Spannungsänderung höchstens in einer bestimmten Zeit erwartet werden kann.

Anstiegszeit
(rise time)

Maß für die Steilheit eines Impulsanstiegs. Es ist die Zeit, in der ein Signal von 10 % auf 90 % seines Dachwertes angestiegen ist (s. Bild 1).

Antivalenzschaltung, EXOR-Schaltung
(exclusive-OR gate)

Logische → Digitalschaltung, welche die Antivalenz-Funktion (Exklusiv-ODER) darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn beide Eingangssignale unterschiedliche Werte haben (s. Bild 13). Mathematische Beschreibung: $X = (A \wedge \overline{B}) \vee (\overline{A} \wedge B)$, (gelesen: X = A UND B NICHT ODER A NICHT UND B).

Astabile Schaltung, astabiler Multivibrator
(astable circuit)

Schaltung zum Erzeugen eines freischwingenden Rechtecksignals, dessen Frequenz durch RC-Glieder** bestimmt wird. Sie enthält im allgemeinen zwei kapazitiv gekoppelte → Inverter mit einer zusätzlichen kapazitiven Rückkopplung vom Ausgang des zweiten auf den Eingang des ersten Inverters.

Ausgangsfächer, Ausgangslastfaktor
(fan-out)

Innerhalb einer → logischen Digitalfamilie das Maß dafür, wie viele Eingänge von einem Ausgang parallel angesteuert werden können. Ein Ausgang mit dem Ausgangsfächer 15 kann demnach 15 Eingänge mit jeweils dem → Eingangsfächer 1 ansteuern.

---

* Technologie: Gesamtheit der technischen Arbeitsvorgänge in einem Fertigungsbereich.

** R: Abkürzung für Widerstand.
C: Abkürzung für Kondensator, Kapazität.

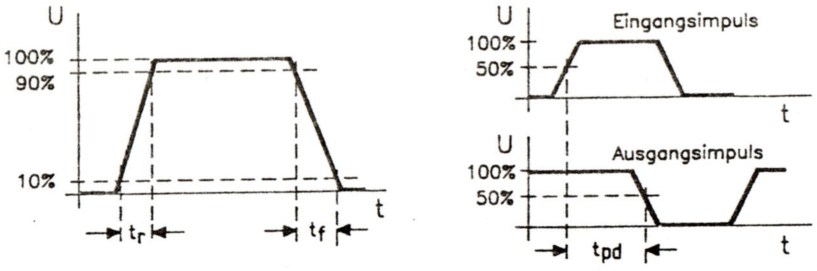


Bild 1: Definition von Anstiegszeit t_r, Abfallzeit t_f und Verzögerungszeit t_{pd}

B

BCD, binär codiertes Dezimalsystem
(BCD, binary coded decimal)

Ein → Binärcode, bei dem jede Ziffer einer Dezimalzahl in ein binäres 4-bit-Wort umgeformt wird. Z.B. wird die Dezimalzahl 30 zu 0011 0000. Der Code ist bewertet mit 1-2-4-8 (von rechts nach links) und wird deshalb auch als 1-2-4-8-Code bezeichnet.

BCD-Zähler
(BCD counter)

Ein → Binärzähler, der aus hintereinandergeschalteten Gruppen von jeweils vier Zählerstufen besteht. Jede dieser Gruppen liefert an vier Ausgängen eine Dekade des BCD-Codes und gibt nach dem zehnten Eingangsimpuls einen Übertragsimpuls an die nächste Gruppe ab.

BFL
(BFL, buffered FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → DFETs und → Schottkydioden besteht (Bild 2). Die Transistoren T_1 und T_2 bilden die logische Eingangsverknüpfung, die Transistoren T_3 und T_4 sind als ohmsche Widerstände geschaltet, der Transistor T_5 ist der Ausgangstransistor in Sourceschaltung, und die beiden Dioden D_1 und D_2 sind Pegelschiebedioden, die den Ausgangspegel an den Eingangspegel angleichen. Die BFL ist sehr schnell, hat eine hohe Ausgangsbelastbarkeit, aber auch eine relativ hohe Verlustleistung.

Binärcode
(binary code)

Ein Code, bei dem jedes Codeelement nur zwei verschiedene Werte, z.B. 0 und 1, annehmen kann.

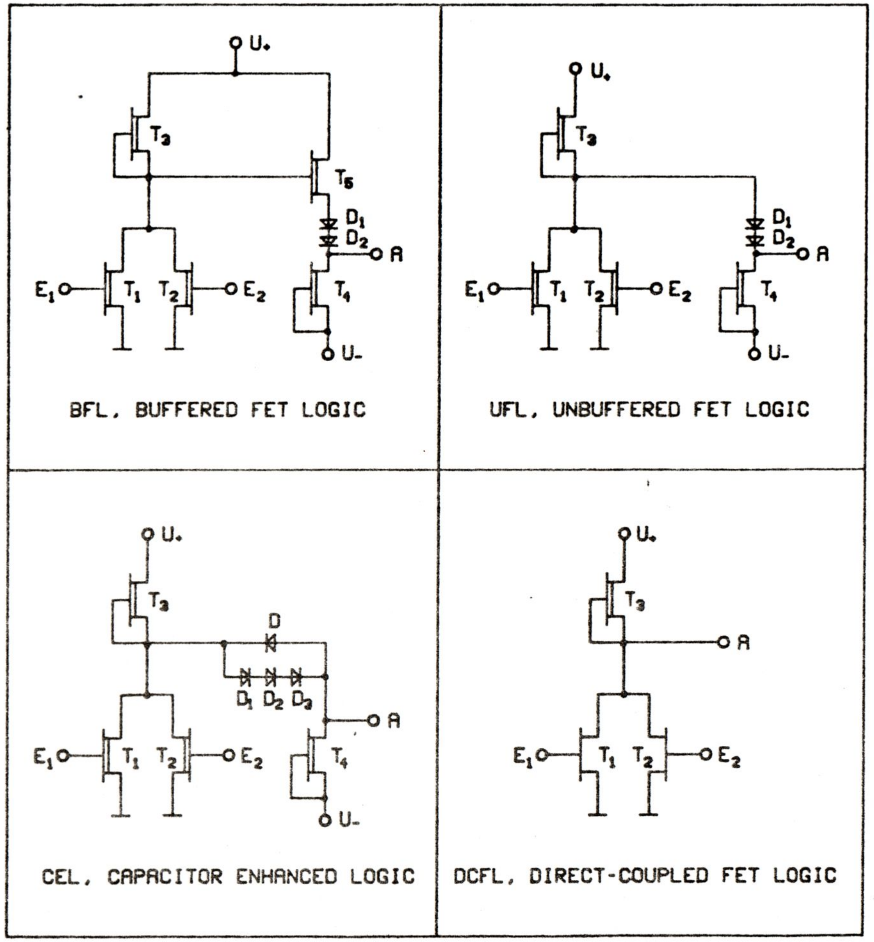


Bild 2: Grundschaltungen von GaAs-Digitalschaltungsfamilien

Binär-Dezimal-Umwandlung
(binary-to-decimal conversion)

Umwandlung des Binär- in den Dezimal-Code, z.B. der Binärzahl 1101 in die entsprechende Dezimalzahl 13. Dies kann elektrisch in einem Binär-Dezimal-Wandler geschehen, der üblicherweise 4 Eingänge und 10 Ausgänge hat (BCD-Dezimal-Wandler). Jeder der 10 Ausgänge entspricht einer der Ziffern von 0 bis 9 und führt dann Ausgangssignal, wenn die zugehörige binäre Signalkombination an den Eingängen anliegt.

Binäre Speicherzelle
(binary cell)

Die kleinste Einheit eines binären Speichers. Sie kann die logischen Werte 0 oder 1 annehmen und somit 1 bit speichern.

Binärzähler
(binary counter)

Schaltung, die nach jeweils zwei Eingangssignalen (dies können Impulse oder Impulsflanken sein) den Ausgangszustand ändert und somit durch 2 teilt. Der Zählbereich kann durch Hintereinanderschaltung gleichartiger Stufen beliebig erweitert werden. Bei n Stufen sind 2^n Zählerzustände gegeben, so daß die Ziffern von 0 bis 2^n-1 im → Binärcode dargestellt werden.

Bipolare Bauelemente
(bipolar devices)

Halbleiterbauelemente mit abwechselnd aufeinanderfolgenden p- und n-dotierten Schichten, z.B. npn-Transistor, Pin-Diode. Im Gegensatz dazu → Unipolare Bauelemente.

Bipolare Speicher
(bipolar memories)

Speicher (z.B. → RAMs oder → ROMs), die in Bipolartechnik ausgeführt sind. Im Vergleich mit → CMOS-Speichern sind höhere Arbeitsgeschwindigkeiten erreichbar, jedoch auf Kosten höherer Leistungsaufnahme und geringerer → Packungsdichte.

Bipolartechnologie
(bipolar fabrication)

Zur Herstellung von → bipolaren Bauelementen ist eine Reihe von → Maskierungs- und → Diffusionsschritten erforderlich. Ausgangspunkt ist ein einkristallines Halbleiter-→-Substrat, in das von der Oberfläche her Fremdatome (→ Dotiermittel) eindiffundiert werden, wodurch entweder n- oder p-leitende Schichten erzeugt werden. Die Dotierung geschieht örtlich begrenzt über Masken, die durch → Fotolithografie aufgebracht werden. Heute wird ausschließlich der → Planarprozeß in Verbindung mit → Epitaxie angewendet.

Bistabile Schaltung, Flip-Flop
(bistable circuit)

Logische Schaltung, die zwei stabile Ausgangszustände hat, zwischen denen sie umgeschaltet werden kann. Der jeweils eingeschaltete stabile Zustand bleibt dabei auch nach Wegnahme der Eingangssignale erhalten. Bistabile Schaltungen können deshalb zur Informationsspeicherung verwendet werden.

Bit
(bit)

Abkürzung für *binary digit* (binäres Zeichen). Es ist die kleinste Informationseinheit; das binäre Zeichen kann nur zwei unterschiedliche Werte, z.B. 0 oder 1, annehmen.

Boolesche Algebra, Schaltalgebra
(Boolean algebra)

Auf den englischen Mathematiker Boole zurückgehendes Verfahren, um logische Zusammenhänge mathematisch beschreiben zu können. Grundlage für die Berechnung von Systemen, die mit → logischen Schaltungen aufgebaut werden. Da eine logische Entscheidung nur entweder richtig oder falsch sein kann (1 oder 0), ist eine technische Darstellung mit binären digitalen Schaltungen, die ebenfalls nur zwei diskrete Signalzustände aufweisen, möglich.

Bus
(bus)

Im allgemeinen mehrpolige Verbindung (8polig, 16polig) zwischen Datensendern und Datenempfängern, häufig bidirektional, d.h. in beiden Richtungen verwendbar.

C

CFL
(CFL, capacitor enhanced FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T_1 und T_2 bilden die logische Eingangsverknüpfung, die Transistoren T_3 und T_4 sind als ohmsche Widerstände geschaltet. Die Dioden D_1-D_3 sind Pegelschiebedioden, die den Ausgangspegel an den Eingangspegel angleichen. D_4 ist in Sperrichtung gepolt und wirkt als Kondensator, der parallel zu D_1-D_3 geschaltet ist, wodurch die Schaltgeschwindigkeit gegenüber der → UFL erhöht wird.

Chip
(chip)

Unverkapseltes Halbleiterbauelement (Transistor, Integrierte Schaltung), das entweder zum Einbau in ein Gehäuse vorgesehen ist oder direkt in → Hybridschaltungen eingesetzt wird. Die Verbindungen mit der Schaltung werden durch gebondete* Drähtchen hergestellt.

CML, Logik mit Stromsteuerung
(CML, current mode logic)

Familie von Logikschaltungen, bei denen die Schalttransistoren im ungesättigten Bereich arbeiten, wodurch hohe Schaltgeschwindigkeiten, symmetrische Übertragungskennlinien und geringes Überschwingen erreicht werden, → ECL.

CMOS, Komplementäre MOS-Technik
(CMOS, complementary metal oxide semiconductor)

Eine → MOS-Technik, bei der im Gegensatz zur → NMOS- oder → PMOS-Technik beide Transistortypen gemeinsam verwendet werden. Der einfache → Inverter besteht aus der Reihenschaltung eines N-Kanal- und eines P-Kanal-Transistors, so daß im Ruhezustand und unabhängig vom logischen Eingangssignal immer einer der beiden Transistoren gesperrt ist. Strom fließt nur während des Umschaltvorganges (Bild 3). CMOS-Schaltungen haben deshalb geringe Verlustleistungen; sie haben hohe Störsicherheit und sind unempfindlich gegenüber Spannungs- und Temperaturschwankungen. CMOS-Schaltungen, die durch → Ionenimplantation hergestellt werden, arbeiten noch mit Betriebsspannungen von 1,5 V.

D

DCFL
(DCFL, direct-coupled FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → EFETs, → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T_1 und T_2 (EFETs) bilden die logische Eingangsverknüpfung, der Transistor T_3 (DFET) ist als ohmscher Widerstand geschaltet. Eine Pegelangleichung wie bei → BFL, → UFL oder → CFL ist hier nicht erforderlich. Die Leistungsaufnahme ist gering, die Ausgangsbelastbarkeit ist vergleichbar wie bei der UFL.

---

* **Bonden:** Verbinden von Metallen durch Druck, Temperaturerhöhung und/oder Ultraschallenergie.

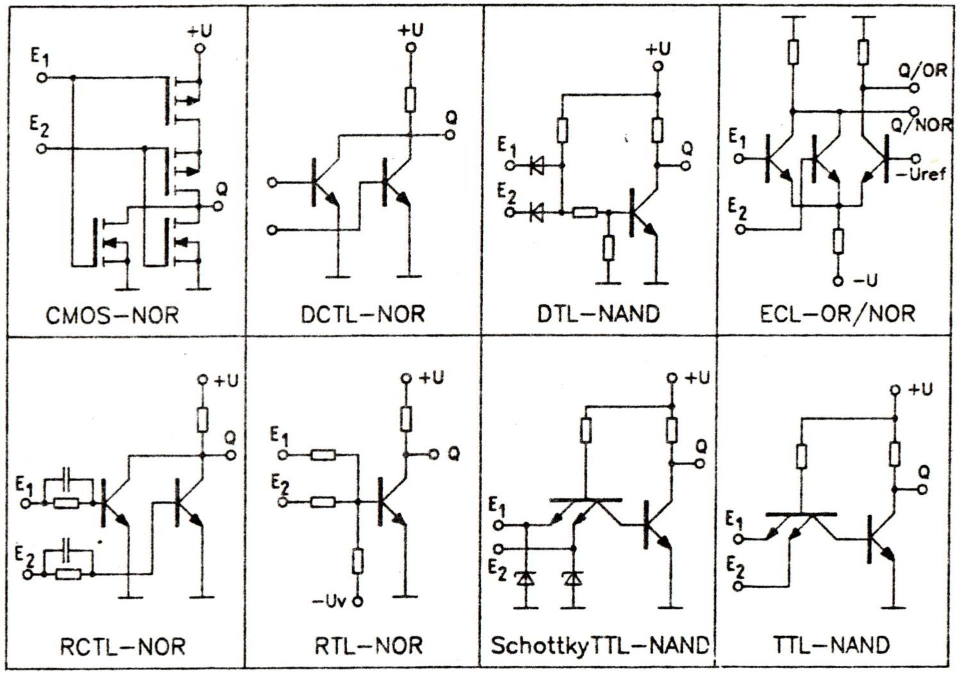


Bild 3: Grundschaltungen der wichtigsten Logikfamilien in vereinfachter Darstellung

U	Betriebsspannung
Uv	Vorspannung
Uref	Referenzspannung
E_1, E_2	Eingänge
Q	Ausgang

DCTL, Direkt gekoppelte Transistorlogik
(DCTL, direct-coupled transistor logic)

Nicht mehr gebräuchliche Logikfamilie, bei der parallel- oder hintereinandergeschaltete Transistoren die logische Verknüpfung bilden, wobei die Basisanschlüsse direkt von der vorhergehenden Schaltung angesteuert werden (s. Bild 3).

Decoder, Decodierschaltung
(decoder)

Schaltung, die einen Code in einen anderen Code umwandelt, z.B. BCD in dezimal, dezimal in binär, BCD in 7-Segment.

DFET, FET vom Verarmungstyp
(DFET, depletion mode FET)

Eine Art des → Feldeffekttransistors, bei dem schon dann ein leitfähiger Kanal zwischen Drain und Source besteht, wenn das Gate noch keine Steuerspannung aufweist. Je nach Polarität der Gatespannung nimmt die Leitfähigkeit weiter zu (Anreicherungsbetrieb) oder ab (Verarmungsbetrieb). Bei direkter Verbindung von Gate und Source (Gatespannung Null) wirkt der DFET wie ein ohmscher Widerstand.

D-Flipflop
(D-type flip-flop)

Digitale → Folgeschaltung mit mindestens einem D-(Daten-)Eingang und einem C-(Takt-, Clock-)Eingang und häufig zwei komplementären* Ausgängen. Der logische Wert des Datensignals wird mit dem Taktsignal auf den Ausgang übertragen und gespeichert (s. Bild 10). D-Flipflops werden als Zähler, Frequenzteiler, Speicher und Schieberegisterstufen verwendet.

Dickfilmschaltungen, Siebdruckschaltungen
(thick-film circuits)

Eine Art der → Hybridschaltungen, die üblicherweise im → Siebdruckverfahren hergestellt werden. Dabei können auf einem → Substrat, meist Keramik, alle passiven Bauelemente, wie Leiterbahnen, Streifenleitungen, Leitungskreuzungen, Widerstände und Kondensatoren, als verhältnismäßig dicke Schichten (5 bis 40 µm) aufgedruckt und in Brennprozessen mit dem Substrat und untereinander fest verbunden werden. Dickfilmschaltungen werden sowohl für die Konsumelektronik** als auch in hochwertigen Ausführungen für schnelle Digitalschaltungen (Bild 4) und für Mikrowellenschaltungen hergestellt. In Verbindung mit Halbleiter-→-chips lassen sich auch umfangreiche Systeme auf geringem Raum integrieren. Das abwechselnde Drucken von Leiterbahnebenen und dielektrischen Isolierebenen ergibt Vielschichtschaltungen (Multilayerschaltungen) für komplizierte Verdrahtungsstrukturen.

---

* **Komplementäre Ausgänge:** Ausgangspaar, welches das logische Signal und das invertierte Signal gleichzeitig abgibt.

** **Konsumelektronik:** Überbegriff für elektrische Bauelemente oder Geräte, die für den allgemeinen Gebrauch bestimmt sind.
Im Gegensatz dazu: *Kommerzielle Elektronik*, die zum Einsatz in Industriebetrieben, an Hochschulen oder an Forschungsinstituten bestimmt ist.

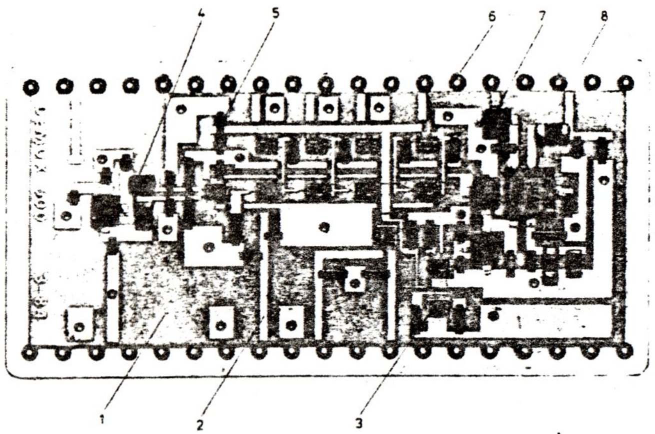


Bild 4: Hybrid integrierte Dickfilmschaltung. Beispiel eines schnellen digitalen Multiplexers (700 Mbit/s)
Die Siebdruckschaltung ist mit verkapselten Einzeltransistoren, Chipkondensatoren und monolithisch integrierten Schaltungen in Chipform bestückt

1 Keramiksubstrat
2 50-Ω-Leitung
3 Integrierte Schaltung
4 Chipkondensator
5 gedruckter Widerstand
6 Anschlußstift
7 Einzeltransistor
8 Gehäuseboden

Dielektrische Isolation
(dielectric isolation)

Üblicherweise sind die Bauelemente bipolarer Integrierter Schaltungen durch Sperrschichten voneinander isoliert, was zu störenden Sperrschichtkapazitäten und Leckströmen führt. Bestimmte Herstellungsverfahren benutzen deshalb Siliziumdioxid (SiO_2, Quarz), das durch Oxidation der Siliziumoberfläche entsteht, als isolierendes Dielektrikum. Dadurch werden die Streukapazitäten wesentlich verkleinert und die Leckströme beseitigt.

Differenzverstärker
(differential amplifier)

Symmetrische Schaltung aus mindestens zwei, möglichst identischen Transistoren mit Konstant-Stromspeisung (Bild 5). Die Differenz der Ausgangsspannungen ist dabei proportional (verhältnisgleich) der Differenz der Eingangsspannungen, während gleiche Eingangsspannungen, unabhängig von ihrer Höhe, keine Änderung der Ausgangsspannung zur Folge haben. Der Differenzverstärker ist wesentliches Bauteil aller → Operationsverstärker und der → Emitter-gekoppelten Logik.

Diffusion
(diffusion)

Verfahren zum → Dotieren eines Halbleiterkristalls mit Fremdatomen, um eine bestimmte Art (P oder N) und Höhe der Leitfähigkeit zu erreichen. Der Diffusionsvorgang geschieht bei hohen Temperaturen. Die Eindringtiefe des → Dotiermittels ist zeitabhängig.

digital
(digital)

Ziffernmäßig, von engl. digit Ziffer. Digitale Signale beschränken sich auf bestimmte diskrete Werte oder Wertebereiche eines Signals. Das einfachste und gebräuchlichste Digitalsystem ist das duale, zweiwertige, bei dem nur 2 Signalzustände möglich sind, die allgemein mit 0 und 1 bezeichnet werden, wobei z.B. 0 eine niedrige und 1 eine hohe Spannung sein kann. Daneben werden für besondere Anwendungen noch Digitalsysteme mit mehr als zwei Ziffern verwendet, z.B. das Ternärsystem mit drei Ziffern.

Digital-Analog-Wandler, D/A-Wandler
(digital-to-analog converter)

Schaltung zur Umformung eines digitalen Eingangssignals in ein analoges Ausgangssignal. Die Zahl der Eingangsbits bestimmt die Stufung des Ausgangssignals. Ein n-bit-Wandler erzeugt ein Ausgangssignal mit bis zu 2^n diskreten Signalwerten. Durch ein Tiefpaßfilter mit geeigneter Grenzfrequenz wird dieses treppenförmige Signal geglättet.

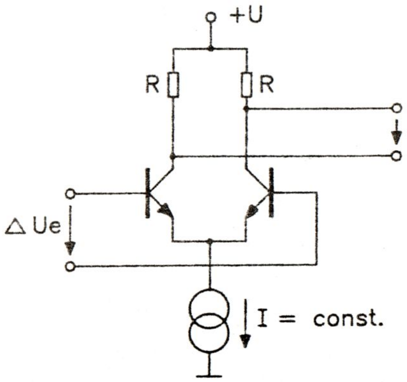


Bild 5: Prinzipschaltung des Differenzverstärkers. Die Differenz der Ausgangsspannungen ist proportional zur Differenz der Eingangsspannungen

Digitale Anzeigen
(digital displays, displays)

Sie werden zur Darstellung von Dezimalziffern oder auch als alphanumerische Anzeigen zur Darstellung von Buchstaben, Ziffern und Zeichen verwendet. Aufgebaut sind sie als Leuchtdioden (→ LED), Flüssigkristalle (→ LCD) oder Gasentladungsröhren. Am bekanntesten sind die 7-Segment-Anzeigen (Bild 6), die aus 7 einzeln ansteuerbaren Leuchtsegmenten die Ziffern von 0 bis 9 nachbilden und z.B. über einen BCD-7-Segment-→-Decoder an eine digitale Schaltung angeschlossen werden.

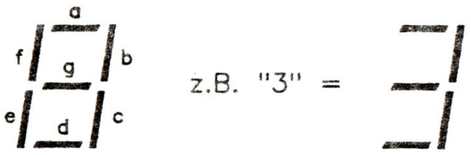


Bild 6: 7-Segmentanzeige
Die Segmente sind mit a bis g bezeichnet und werden einzeln angesteuert, bei „3" z.B. die Segmente a, b, c, d und g

Digitalschaltungen
(digital circuit)

Schaltungen, die mit → digitalen Signalen arbeiten, deren Eingangs- und Ausgangssignale demnach auf bestimmte Signalwerte oder Wertebereiche beschränkt sind. Im normalen Sprachgebrauch werden darunter ausschließlich binäre digitale Schaltungen verstanden, die nur zwei unterschiedliche Signalzustände, 0 und 1 oder → Low und → High, aufweisen.

DIP, Dual-in-Line-Gehäuse
(DIP, dual in-line package)

Gehäuse aus Kunststoff oder Keramik mit zwei parallelen Anschlußstiftreihen zur Aufnahme integrierter Schaltungen. Die Stifte sind im 1/10-Zoll-Raster angeordnet. Das Standard-DIP hat 14 Anschlüsse und ist 6,5 mm x 19 mm groß. Es gibt aber auch z.B. DIPs mit nur 8 und mit bis zu 60 Anschlüssen.

Disjunktion
(disjunction)

Logische Verknüpfung, die dann erfüllt ist, wenn mindestens eine Eingangsvariable den Wert 1 hat. Sie stellt die ODER-Funktion dar und kann mittels → ODER-Schaltungen realisiert werden. Mathematisch lautet die ODER-Funktion zweier Variabler $X = A \vee B$ (gesprochen: A ODER B).

Displays
(displays)

Anderer Ausdruck für → (digitale) Anzeigen.

Dotieren
(doping)

Einbau von Fremdatomen (→ Dotiermittel) in einen Halbleiterkristall zur Erhöhung der Leitfähigkeit durch Erzeugung von Überschußladungen (Elektronen oder Löcher).

Dotiermittel
(dopant)

Fremdatome, mit denen ein Halbleiterkristall gezielt verunreinigt wird zur Beeinflussung der Art und Höhe der Leitfähigkeit, z.B. durch → Diffusion oder → Ionenimplantation.

Dünnfilmschaltungen
(thin-film circuits)

Eine Art der → Hybridschaltungen, die durch Aufdampfen oder Aufsputtern* in Verbindung mit → Fotolithografie und Ätztechniken hergestellt werden. Dabei können auf einem → Substrat, meist Keramik oder Glas, alle passiven Bauelemente, wie Leiterbahnen, Streifenleitungen, Leitungskreuzungen, Widerstände und in Grenzen auch Kondensatoren, als sehr dünne (0,01 bis 1 µm) Schichten aufgebracht werden. Leiterbahnen werden häufig noch galvanisch verstärkt (bis auf etwa 10 µm). Dünnfilmschaltungen werden als Analogschaltungen für Frequenzen bis zum MHz-Bereich (Tantaltechnik), hauptsächlich aber als hybrid integrierte Mikrowellenschaltungen (→ MIC) bis zu höchsten Frequenzen eingesetzt. Wie bei den → Dickfilmschaltungen werden auch hier die Halbleiterbauelemente als → Chips nachträglich in die Schaltungen eingebaut (Bild 7).

Dynamische MOS-Schaltungen
(dynamic MOS circuits)

Im Gegensatz zu den üblichen Speichern, die statisch arbeiten, indem sie die Speicherwirkung von → bistabilen Schaltungen ausnutzen, wird in dynamischen MOS-Schaltungen die Information in Form einer Kondensatorladung gespeichert. Dieses Verfahren läßt sehr hohe → Packungsdichten zu, hat aber den Nachteil, daß die Kondensatorladungen über nicht zu vermeidende Leckströme abfließen. Dynamische Speicher müssen deshalb periodisch nachgeladen (aufgefrischt, refreshed) werden. Dies geschieht durch automatisches Auslesen und Wiedereinschreiben des Speicherinhalts.

Dynamisches RAM
(dynamic RAM)

Bestimmte → dynamische MOS-Schaltung (Speicher mit wahlfreiem Zugriff, → RAM), bei der die Information in Form von Kondensatorladungen gespeichert wird.

DTL, Dioden-Transistorlogik
(DTL, diode-transistor-logic)

Logikfamilie, bei der ein Transistor als → Inverter dient und die logische Verknüpfung durch Diodennetzwerke geschieht (s. Bild 2).

E

EAROM, Elektrisch veränderbarer Festwertspeicher
(EAROM, electrically alterable ROM)

Sonderform eines Festwertspeichers (→ ROM), der elektrisch programmiert werden kann und dessen Speicherinhalt nichtflüchtig ist, d.h. dessen Speicherinhalt auch bei Wegfall der Betriebsspannung erhalten bleibt. Grundsätzlich handelt es sich um einen Schreib-Lesespeicher (→ RAM), jedoch geht der Schreibvorgang sehr viel langsamer vor sich als der Lesevorgang, so daß diese Speicherart ausschließlich als ROM verwendet wird.

ECL, Emittergekoppelte Logik
(ECL, emitter-coupled logic)

Logikfamilie, deren Schalttransistoren im nichtgesättigten Bereich arbeiten (→ CML) und deren Grundprinzip auf den → Differenzverstärker zurückzuführen ist (s. Bild 2). Es ist die schnellste zur Zeit verwendete Logikfamilie auf Siliziumbasis mit Schaltzeiten von bis unter 1 ns. Die Schaltungen haben symmetrische Übertragungskennlinien, deshalb ist die Störsicherheit in beiden logischen Zuständen etwa gleich groß. Der → logische

* **Sputtern**: Dünnfilmprozeß, bei dem das aufzubringende Material durch Beschuß mit Edelgasionen aus der Materialoberfläche herausgelöst wird und sich als Film auf der Substratoberfläche niederschlägt.

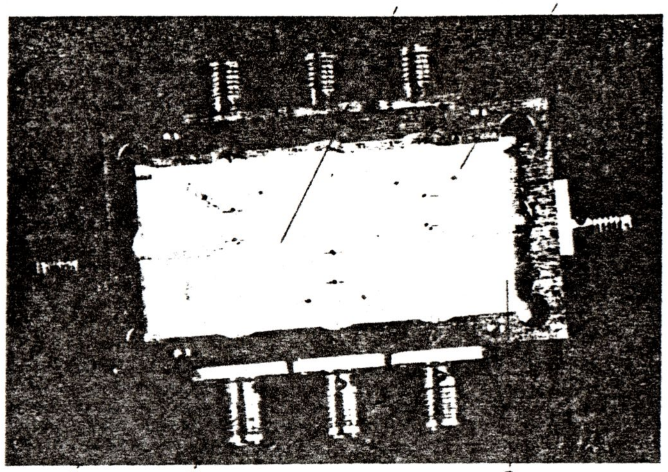


Bild 7: Hybrid integrierte Dünnfilmschaltung. Auf einem Keramiksubstrat (2,5 cm x 5 cm) sind Leitungsstrukturen aufgesputtert. Chipdioden sind leitfähig aufgeklebt und gebondet

1 Gehäuse
2 Koaxialbuchse
3 Keramiksubstrat
4 Diodenchip
5 Leiterbahn

Hub liegt bei etwa 1 V, der Ausgangsfächer ist wegen der Ausgangsemitterfolger hoch, es können 50-Ω-Leitungen direkt angeschlossen werden. Auf Grund des Differenzverstärkerprinzips kann die Referenzspannung gegen Temperatur- und Betriebsspannungsänderungen kompensiert werden. Außerdem steht stets ein komplementäres Ausgangspaar zur Verfügung.

EEROM, Elektrisch löschbarer Festwertspeicher
(EEROM, electrically erasable ROM)

Sonderform eines Festwertspeichers (→ ROM), der elektrisch programmiert werden kann und dessen Speicherinhalt nichtflüchtig ist. Im Gegensatz zum → EAROM muß dieser Speicher vor dem Neuprogrammieren durch eine angelegte Spannung vollständig gelöscht werden. Lösch- und Programmiervorgang können fast beliebig oft wiederholt werden.

EFET, FET vom Anreicherungstyp
(EFET, enhancement mode FET)

Eine Art des → Feldeffekttransistors, der bei Spannung Null zwischen Gate und Source nichtleitend ist. Erst ab einer bestimmten Gatespannung, deren Polarität umgekehrt zur Drain-Sourcespannung sein muß, setzt die Leitfähigkeit des Drain-Sourcekanals ein und nimmt mit wachsender Gatespannung zu.

Eingangsfächer, Fan-in
(fan-in)

Der Eingangsfächer stellt in Form einer dimensionslosen Zahl die Belastung dar, die ein bestimmter Eingang einer logischen Schaltung innerhalb einer logischen Familie für einen ihn ansteuernden Ausgang bildet. Normale Eingänge haben den Eingangsfächer 1, es gibt jedoch auch Eingänge (z. B. Takteingänge bei mehrstufigen Schieberegistern) mit höheren Werten. Die Summe der Eingangsfächer aller parallel an einen Ausgang geschalteten Eingänge darf den Ausgangsfächer dieses Ausgangs nicht überschreiten.

Elektronenstrahllithografie
(electron beam lithography)

→ Maskenherstellung im Elektronenstrahlverfahren.

Epitaxie
(epitaxy)

Epitaxie ist ein technologischer Vorgang im → Planarprozeß, bei dem man auf der Oberfläche eines Einkristalls aus der Gas- oder Flüssigphase eine Schicht aus dem gleichen Material aufwachsen läßt, wobei diese Schicht die Kristallstruktur fortsetzt. Durch gleichzeitiges Zusetzen von → Dotiermitteln wird die neue Schicht dotiert. Es entstehen dadurch beliebig dünne und gleichmäßiger dotierte Schichten, als dies bei der → Diffusion möglich ist.

EPROM, Elektrisch programmierbarer Festwertspeicher
(EPROM, electrically programmable ROM)

Sonderform eines Festwertspeichers (→ ROM), der elektrisch programmiert werden kann und dessen Speicherinhalt nichtflüchtig ist. Wie ein → EEROM muß auch dieser Speicher vor dem Neuprogrammieren vollständig gelöscht werden, jedoch geschieht dies hier durch Belichten der Halbleiteroberfläche mit UV-Licht*. Im Gehäuse dieser Speicher ist deshalb ein Quarzfenster eingebaut, durch welches das UV-Licht eindringen kann. Der Lösch- und Programmiervorgang kann nahezu beliebig oft wiederholt werden.

---

* UV-Licht: Licht im nichtsichtbaren Spektralbereich, oberhalb des Violett

Exklusions-Schaltung, Ausschließungs-Schaltung
(exclusion circuit)

Logische Schaltung mit zwei Eingängen, welche die → UND-NICHT-Verknüpfung bildet. Das Ausgangsergebnis ist nur dann wahr, wenn die erste Eingangsanweisung wahr, die zweite dagegen falsch ist. Mathematischer Ausdruck: $X = A \wedge \bar{B}$ (gelesen: A UND B NICHT).

Exklusiv-NICHT-ODER-Schaltung
(exclusive NOR)

→ Äquivalenzschaltung

Exklusiv-ODER-Schaltung
(exclusive OR)

→ Antivalenzschaltung

F

FAMOS
(FAMOS, floating-gate avalanche-injection MOS)

Eine Ausführungsform des → EPROMs. Als Speicherzelle dient ein → MOS-Transistor, dessen → Gate aus Silizium besteht und durch völlige Einbettung in hochisolierendes SiO_2 potentialmäßig schwimmend (floating) angeordnet ist (Bild 8). Bei der Programmierung wird durch Anlegen einer hohen Spannung ein Lawinendurchbruch (avalanche-injection) hervorgerufen, der Ladungsträger zum Gate transportiert. Diese Ladung kann nach Wegnahme der Programmierspannung von dem isolierten Gate nicht mehr abfließen und wirkt für den betreffenden Transistor wie eine dauernd anliegende Gatespannung. Die Abfrage des Speicherinhalts geschieht durch Messung des Drain-Source-Widerstandes jedes einzelnen Transistors. Gelöscht wird der gesamte Speicherinhalt mit UV-Licht.

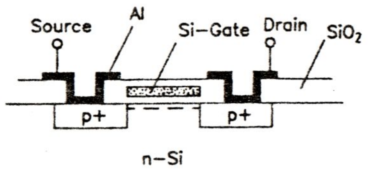


Bild 8: FAMOS-Transistor. Die Gateelektrode ist völlig in SiO_2 eingebettet: eingespeicherte Ladungen können deshalb nicht abfließen

FAST-Logik
(FAST logic)

Eine Weiterentwicklung der → TTL-Technik. Dabei werden die Bauelemente gegeneinander nicht durch Sperrschichten, sondern durch Oxid isoliert, wodurch die parasitären Kapazitäten verringert und die Bauelemente wesentlich kleiner gemacht werden können. Die Schaltschwelle liegt höher als bei der TTL und auch höher als bei der Schottky-TTL, die Signalverzögerungszeiten sind mit 2,7 ns um ca. 10 % niedriger, die Verlustleistung beträgt nur ein Fünftel (4 mW). FAST-Schaltungen sind zu den übrigen TTL-Familien kompatibel.

FET, Feldeffekttransistor
(FET, field effect transistor)

Feldeffekttransistoren sind → unipolare Halbleiterbauelemente mit Steuerung des Stromes in einem leitenden Kanal durch ein elektrisches Feld. Es wird nur **eine** Art von Ladungsträgern verwendet (n-Typ oder p-Typ). Im Gegensatz zum bipolaren Transistor geschieht die Steuerung strom- und damit leistungslos. Deshalb weisen FETs sehr hohe Eingangswiderstände auf (Gigaohm-Bereich). Zwei Grundformen sind zu unterscheiden: der Sperrschicht-FET (pn-FET) und der Isolierschicht-FET (IGFET, insulated gate FET). FETs haben meistens drei Elektroden: Source (Emitter), Drain (Kollektor) und Gate (Steuerelektrode), es gibt jedoch auch Ausführungen mit zwei Gates (Dual-gate FET). Zwischen Drain und Source wird die Betriebsspannung angelegt, zwischen Gate und Source die Steuerspannung.

Sperrschicht-FETs verwenden einen p- oder n-dotierten Kanal mit metallischen Anschlüssen für Drain und Source, in dem ein n- oder p-dotierter Bereich mit dem metallischen Gate-Anschluß eindiffundiert ist (Bild 9). Liegt keine Gatespannung an, so ist der Kanal leitend, und zwar in beiden Richtungen (ohmsches Verhalten). Durch eine an das Gate angelegte Sperrspannung wird der Stromfluß zwischen Source und Drain mit wachsender Sperrspannung laufend verringert (der Kanal wird eingeschnürt), bis schließlich der Strom Null wird (Abschnür-, pinch-off-Spannung).

Bei Isolierschicht-FETs, zu denen die → MOS-FETs (metal-oxide-semiconductor) und die MNS-FETs (metal-nitride-semiconductor) gehören, allgemein spricht man auch von MIS-FETs (metal-insulator-semiconductor), ist die Gateelektrode durch eine Oxid- oder Nitrid-Schicht vom Kanal isoliert (s. Bild 9). Da bei diesen Transistoren auch der geringe Sperrstrom, den pn-FETs aufweisen, entfällt, haben sie **sehr** hohe Eingangswiderstände. IG-FETs werden als p- oder n-Kanal-Typen hergestellt, jedoch muß noch zwischen solchen vom Anreicherungs- (enhancement type) und Verarmungstyp (depletion tpye) unterschieden werden. Anreicherungstypen sind ohne Gatespannung gesperrt und werden mit wachsender Gatespannung leitend. Verarmungstypen sind auch bei Gatespannung Null bereits leitend, die Leitfähigkeit des Kanals nimmt je nach Polarität der Gatespannung entweder zu oder bis zur völligen Sperrung ab.

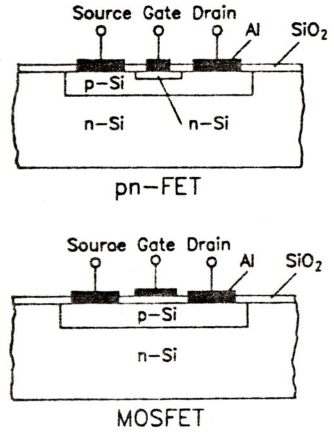


Bild 9 Aufbau eines planaren pn-FETs und eines planaren MOSFETs im Querschnitt

FIFO-Speicher
(FIFO memory, first in/first out memory)

Der FIFO-Speicher ist ein Schreib-Lesespeicher ohne Adressierung, bei dem die ankommenden Daten der Reihe nach, beginnend mit dem letzten freien Speicherplatz, eingeschrieben und in der gleichen Reihenfolge auch wieder ausgelesen werden, wobei nach jedem ausgelesenen Bit der gesamte Speicherinhalt nachrückt, so daß nur ein Lesevorgang möglich ist. Schreiben

und Lesen kann mit unterschiedlichen Taktfrequenzen geschehen (asynchroner Betrieb). FIFOs werden hauptsächlich als Zwischenspeicher zur Anpassung der Datenrate des Eingangssignals an die Datenrate des Verarbeitungssystems verwendet.

Flatpack, Flachgehäuse
(flatpack)

Quadratische oder rechteckige Gehäuseform für integrierte Schaltungen, meist aus Keramik, mit seitlich herausgeführten Anschlußfahnen. Kleiner, aber teurer, und schwieriger zu montieren als → DIP-Gehäuse.

Flipflop
(flip-flop)

Eine → bistabile Schaltung, die als Zähler, Frequenzteiler, Speicher oder Schieberegisterstufe verwendet werden kann. Die wichtigsten Flipflops sind: → D-, → JK-, → RS-, → RST- und → T-Flipflop (Bild 10).

Folgeschaltung, Sequentielle Schaltung
(sequential circuit)

Logische Schaltungen, deren Ausgangszustände nicht nur von den augenblicklich anliegenden Eingangssignalen abhängen, sondern auch von zeitlich vorhergehenden. Hierzu gehören z.B. → Flipflops, → Speicherschaltungen und → Schieberegister.

Fotolack
(photoresist)

Lichtempfindlicher Lack, der als dünne Schicht aufgetragen wird und zur → Maskierung bei der → Fotolithografie dient. Es gibt positive und negative Fotolacke; sie werden je nachdem verwendet, ob nach der Entwicklung die belichteten oder die unbelichteten Bereiche zurückbleiben sollen.

Fotolithografie
(photolithographic process)

Bei der Herstellung von Halbleiterbauelementen wird eine ganze Reihe von Masken benötigt, die bestimmte Bereiche der Oberfläche bei der → Diffusion oder beim Ätzen abdecken. Diese Masken werden fotografisch angefertigt, indem eine dünne (unter 1 μm) → Fotolackschicht auf der Oberfläche des Halbleiterkristalls angebracht ist, die über eine entsprechende Fotovorlage belichtet wird. Nach der Entwicklung bleibt das positive oder negative Muster der Vorlage zurück. Die dann noch verbliebenen und vom Entwicklungsvorgang gehärteten Lackbereiche schützen die Oberfläche gegen das Diffusions- oder Ätzmittel. Es können Linienbreiten bis etwa 1 μm erreicht werden. Feinere Auflösungen erfordern andere Verfahren, z.B. Belichtung mit Elektronenstrahlen.

Frequenzteiler
(frequency divider)

Digitale Schaltungen, welche die Frequenz einer Eingangsimpulsfolge auf einen ganzzahligen Bruchteil herabsetzen. Sie können beliebig hintereinandergeschaltet werden. Binäre Frequenzteiler teilen durch Potenzen von 2 (1:2, 1:4, 1:8 usw.), Dezimalteiler teilen durch Potenzen von 10 (1:10, 1:100 usw.). Außerdem gibt es programmierbare Teiler, bei denen durch Steuersignale das Teilungsverhältnis gewählt werden kann. Zur Frequenzteilung können → Flipflops verwendet werden.

FROM, Festwertspeicher mit Schmelzbrücken
(FROM, fusable ROM)

Eine spezielle Art des → PROMs, bei dem die Programmierung durch Abschmelzen von dünnen Metallbrücken geschieht. Das Programmieren kann nur einmal vorgenommen werden.

G

GaAs-Digitalschaltung
(GaAs digital circuit)

Ähnlich wie in der Si-Technik verschiedene Logikfamilien (→ TTL, → ECL) hergestellt werden, haben sich auch in der GaAs-Technik bestimmte Schaltungsfamilien ergeben. Verwendet werden in größerem Maßstab zur Zeit die Buffered-FET-Logik (→ BFL), die Unbuffered-FET-Logik (→ UFL) und die Capacitor-Enhanced-Logik (→ CEL), alle mit → DFETs, sowie die Direct-Coupled-FET-Logik (→ DCFL) mit → EFETs. GaAs-Digitalschaltungen haben Signalverarbeitungsgeschwindigkeiten von einigen Gbit/s und Signalanstiegs- und Abfallzeiten von teilweise unter 100 ps. Bild 11 zeigt als Beispiel eine OR/NOR-Schaltung mit fünf Eingängen (Harris HMD 11101).

Galliumarsenid, GaAs
(gallium arsenide, GaAs)

Ein Halbleiter aus der Gruppe der III-V-Verbindungen, bestehend aus Gallium (Gruppe III des Periodischen Systems) und Arsen (Gruppe V), der gegenüber Silizium zwei prinzipielle Vorteile hat; zum einen ist die Elektronenbeweglichkeit wesentlich höher, so daß Bauelemente mit sehr hoher Grenzfrequenz hergestellt werden können, zum anderen ist der undotierte Halbleiter semiisolierend, weswegen das GaAs-Substrat praktisch als Isolator betrachtet werden kann, in den alle zu integrierenden Bauelemente ohne weitere gegenseitige Isolationsvorkehrungen eingebaut werden können. Aus GaAs werden z.B. → FETs, → Bipolartransistoren, → HEMTs, → monolithisch integrierte Schaltungen, → MMICs und optoelektronische Bauelemente, wie → LEDs und Laserdioden hergestellt.

Gate
(gate)

Steuerelektrode eines → Feldeffekttransistors. Der englische Ausdruck Gate wird außerdem an Stelle von → Verknüpfungsschaltung oder → Gatter benutzt (z.B. UND-Gate, OR Gate).

Gate Array, Gatterfeld
(gate array)

Integrierte Digitalschaltung mit einer großen Anzahl von → Verknüpfungsschaltungen, die nach Kundenangaben (Kundenspezifikation) vom Hersteller zusammengeschaltet werden. Dadurch können kundenspezifizierte umfangreichere Schaltungen, für die sich wegen kleiner Stückzahlen die Entwicklung einer speziellen integrierten Schaltung nicht lohnt, preisgünstiger hergestellt werden.

Gatter, Gatterschaltung
(gate circuit)

Häufig benutzter Laborausdruck für → Verknüpfungsschaltung.

Geätzte Schaltung
(printed circuit)

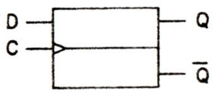
Wird häufig als gedruckte Schaltung bezeichnet. Es handelt sich um metallische Leiterbahnstrukturen, die mittels → Fotolithografie durch Ätzen oder elektrolytisches Abtragen auf ein- oder beidseitig metallbeschichteten Trägerplatten erzeugt werden. Die geätzten Schaltungen werden anschließend mit weiteren Bauelementen (Widerstände, Transistoren, integrierte Schaltungen, Stecker usw.) bestückt. Die Bauteile werden entweder von Hand oder maschinell verlötet.

Gleichstromverstärker
(direct current amplifier, DC-amplifier)

Verstärker, dessen einzelne Stufen direkt (gleichstrommäßig, galvanisch) miteinander verbunden sind, der demnach keine untere Grenzfrequenz besitzt. Fast alle → Operationsverstärker sind Gleichstromverstärker. Bei der Schaltungsauslegung müssen besondere Vorkehrungen gegen Betriebsspannungs-, Bauelementedaten- und Temperaturänderungen getroffen werden.

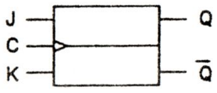
Golddotierte TTL-Schaltungen
(gold-doped TTL circuits)

Schaltungen aus der → TTL-Familie, bei denen durch → Dotieren mit Goldatomen die Erholzeit der Transistoren im Sättigungsbereich herabgesetzt wird. Sie sind etwas langsamer als die → Schottky-TTL-Schaltungen, dafür aber in einem größeren Temperaturbereich verwendbar.



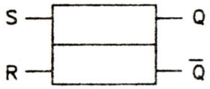
D	C	Q_{n+1}
X	0	Q_n
0	1	0
1	1	1

D–Flipflop



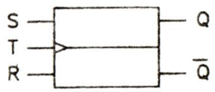
J	K	C	Q_{n+1}
X	X	0	Q_n
0	0	1	Q_n
0	1	1	0
1	0	1	1
1	1	1	$\overline{Q}_n$

JK–Flipflop



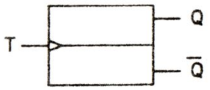
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	n.d.

RS–Flipflop



S	R	T	Q_{n+1}
X	X	0	Q_n
0	0	1	Q_n
0	1	1	0
1	0	1	1
1	1	1	n.d.

RST–Flipflop



T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

T–Flipflop

Bild 10
Die wichtigsten Flipflop-Schaltungen mit Wertetafeln
Q_{n+1} ist der augenblickliche Ausgangszustand
Q_n der vorhergehende
C = 1 kann auch eine Taktflanke sein
n.d. nicht definiert
X beliebig 0 oder 1

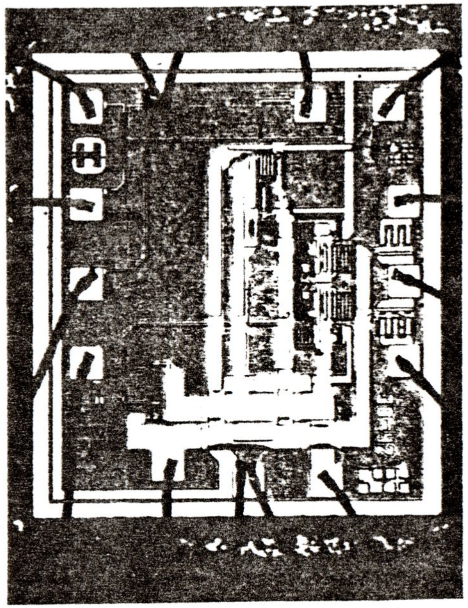
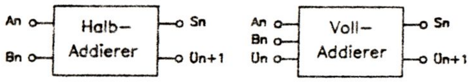


Bild 11: Monolithisch integrierte OR/NOR-Schaltung mit fünf Eingängen in GaAs-Technik (Harris HMD 11101)

H

Halbaddierer
(half adder)

Die einfachste Form eines binären → Addierers, der zwei einstellige Binärzahlen addiert und Summen- und Übertragssignal bildet. Er hat zwei Eingänge und zwei Ausgänge (Bild 12). Zwei Halbaddierer können zu einem → Volladdierer zusammengeschaltet werden.



An	Bn	Sn	Un+1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Un	An	Bn	Sn	Un+1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Bild 12: Schaltsymbole und Wertetabellen des Halb- und Volladdierers

Halbleiter
(semiconductor)

Material, dessen elektrische Leitfähigkeit zwischen derjenigen von Isolatoren und derjenigen von Metallen liegt. Die Leitfähigkeit ist stark von der Temperatur (sie steigt mit wachsender Temperatur) und von Verunreinigungen abhängig. Die zweite Abhängigkeit benutzt man, um durch gezielte Verunreinigungen (→ Dotieren) die Leitfähigkeit nach Typ und Stärke zu beeinflussen. Die bekanntesten Halbleiter sind Germanium (Ge), Silizium (Si) und Gallium-Arsenid (GaAs). Die Leitfähigkeit von Halbleitern kann teilweise auch durch Licht (Fotohalbleiter), elektrische Felder oder magnetische Felder (Halleffekt) beeinflußt werden.

Halbleiterspeicher
(semiconductor memories)

Schaltungen mit einer Vielzahl von → Speicherzellen, die in einer bestimmten Weise angeordnet (organisiert) sind und von außen adressiert werden können. Die Zahl der Speicherzellen ist meistens eine Potenz von 2, z.B. 256, 512, bei Speicherkapazitäten über 1000 bit wird nur das Vielfache des 1024-bit-Speichers mit einem angehängten k angegeben, z.B. 1 k = 1024, 2 k = 2048, 4 k = 4096, 64 k = 65536 usw. Außer den Speicherzellen enthält ein Halbleiterspeicher noch Hilfsschaltungen, wie Schreib- und Leseverstärker und Adressendecoder.

Die wichtigsten Halbleiterspeicher sind Speicher mit wahlfreiem Zugriff (→ RAM), Festwertspeicher (→ ROM) und programmierbare Festwertspeicher (→ PROM). Sie können als → bipolare Speicher oder in → MOS-Technik ausgeführt sein. Bipolare Speicher sind im allgemeinen schneller als MOS-Speicher, die dafür weniger Leistungsaufnahme und höhere → Packungsdichten haben. Halbleiter dringen von ihrer Speicherkapazität her immer mehr in den Bereich der Massenspeicher (Magnetband, Magnetblasenspeicher, Magnetplatte) vor.*

HEMT
(HEMT, high electron mobility transistor)

Der HEMT besteht aus einer Schichtenfolge von GaAs und GaAlAs (Gallium-Aluminium-Arsenid) und ist ein Transistor mit sehr hoher Grenzfrequenz, hoher Verstärkung und niedrigem Rauschen. Er ist sowohl für → monolithisch integrierte Mikrowellenschaltungen als auch für schnelle → Digitalschaltungen geeignet. Bild 13 zeigt den Aufbau: Auf einem undotierten und damit semiisolierendem GaAs-Substrat ist zunächst eine sehr dünne (ca. 70 Å) Schicht aus undotiertem GaAlAs aufgebracht. Darüber befindet sich eine Schicht aus n-dotiertem GaAlAs, die im Bereich der Gateelektrode teilweise abgeätzt wird. Unter den metallischen Source- und Drainelektroden ist eine n-dotierte GaAs-Schicht zur Verbesserung der Leitfähigkeit. Die Schichtenfolge wird üblicherweise durch → Molekularstrahlepitaxie erzeugt. Den Stromtransport übernehmen Elektronen, die in der Grenzschicht zwischen GaAlAs und GaAs in Form einer sehr dünnen („zweidimensionales Elektronengas") Schicht entstehen. Deshalb wird der HEMT auch als TEGFET oder 2DEGFET (two-dimensional electron gas FET) bezeichnet.

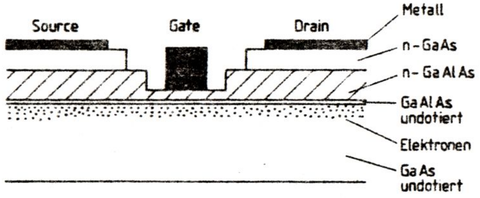


Bild 13. Aufbau eines HEMT im Querschnitt

Hexadezimalsystem
(hexadecimal system)

Zahlensystem, das in der Rechnertechnik häufig benutzt wird. Es baut sich aus Potenzen zur Basis 16 auf. Als Ziffern werden die Dezimalziffern 0 bis 9 und die Buchstaben A bis F verwendet (A ≙ 10, F ≙ 15). Die Hexadezimalzahl A3C z.B. bedeutet $10 \cdot 16^2 + 3 \cdot 16^1 + 12 \cdot 16^0 = 2560 + 48 + 12 = 2620$.

High, H, hoch
(high)

Bezeichnung für denjenigen der beiden logischen Werte oder Wertebereiche einer binären Digitalschaltung, der das höhere Potential aufweist. Im Gegensatz dazu → Low. High kann der logischen 1 (positive Logik) oder der logischen 0 (negative Logik) entsprechen. Bei → TTL-Schaltungen ist High z.B. + 5 V, (Low = 0 V), bei → ECL-Schaltungen −0,9 V (Low = −1,8 V).

HiNIL, HNIL, Störsichere Logik
(high noise immunity logic)

Eine → Logikfamilie, die sich durch eine besonders hohe Störsicherheit auszeichnet. Sie wird oft als Eingangsschaltung vor → TTL- und → CMOS-Schaltungen eingesetzt.

HTL
(high threshold logic)

Bezeichnung für Logikschaltungen mit hoher Störschwelle.

Hybridschaltung
(hybrid circuit)

Schaltung, die sich aus verschiedenen Arten von Bauteilen zusammensetzt, speziell als *hybrid integrierte Schaltung*, z.B. eine → monolithisch integrierte Schaltung mit aufgedampften Metallschichtwiderständen oder, und dies umfaßt den größten Teil aller Hybridschaltungen, die Kombination von passiven → Dickfilm- oder → Dünnfilmschaltungen mit Halbleiterbauelementen.

* Siehe auch den Beitrag „*Digitale Datenspeicher*", Unterrichtsblätter Jg. 37/1984, Nr. 12, S. 438–467 oder Nachdruck Ausgabe Januar 1986.

I

IC, integrierte Schaltung
(IC, integrated circuit)

Englische Abkürzung für → integrierte Schaltung.

Integrierte Schaltung, IC, IS
(integrated circuit, IC)

Verwirklichung einer elektrischen Schaltung durch Integration, d.h. gemeinsames, gleichzeitiges und untrennbares Herstellen und Zusammenfügen aller Bauelemente einschließlich aller Zwischenverbindungen auf einem gemeinsamen Träger (→ Substrat). Erst die Integration ermöglicht hochgradige Miniaturisierungen. Man unterscheidet → monolithisch und → hybrid integrierte Schaltungen, wobei zumindest die erstgenannten nachträglich weder veränderbar noch reparierbar sind. Im allgemeinen Sprachgebrauch werden integrierte Schaltungen abgekürzt auch als IC oder IS bezeichnet.

I^2L, Ionenimplantationslogik
(I^2L, ion implantation logic)
(Sprich: ..i-Quadrat el")

Eine Logikfamilie, hauptsächlich für großintegrierte Schaltungen (→ LSI). Bei der Ionenimplantationstechnologie wird der Halbleiterkristall nicht durch → Diffusion, sondern durch Beschuß mit hochenergetischen ionisierten Dotierungsatomen dotiert. Zur → Maskierung werden Aluminium- oder Siliziumdioxidmasken verwendet. Die Ionen haben nur eine sehr geringe Eindringtiefe (ca. 1 nm), sie kann über die Strahlenergie gesteuert werden. Der Dotierungsgrad des Halbleiters läßt sich sehr genau einstellen.

Inverter, Invertierschaltung, NICHT-Schaltung
(inverter, negation gate, NOT-gate)

Der Inverter ist ein Bauelement, das ein Eingangssignal umkehrt, z.B. aus einer Sinusschwingung die gegenphasige Sinusschwingung erzeugt, wobei im allgemeinen die absoluten Spannungswerte erhalten bleiben. Bei digitalen Schaltungen stellt er die NICHT-Funktion dar, wandelt also 1-Signale in 0-Signale um und umgekehrt. Mathematische Beschreibung: $X = \overline{A}$ (gelesen: X = A NICHT).

Ionenimplantation
(ion implantation)

Technologie zur Herstellung von Halbleiterbauelementen, bei der die → Dotierung durch Beschuß der Halbleiteroberfläche mit hochenergetischen ionisierten Fremdatomen geschieht, → I^2L.

IS, integrierte Schaltung
(IC, integrated circuit)

Abkürzung für integrierte Schaltung.

Isolierung integrierter Schaltungen
(IC isolation)

Da alle Bauelemente bei → integrierten Schaltungen in das gleiche → Substrat eingebettet sind, müssen sie gegeneinander isoliert werden. Dies geschieht entweder durch → pn-Übergänge, die in Sperrichtung gepolt sind, oder durch → dielektrische Isolierschichten, meist SiO_2. Sperrschichtisolierungen haben den Nachteil, daß sie benachbarte Bauelemente kapazitiv miteinander koppeln, die Erdkapazität erhöhen und Leckströme aufweisen. Dielektrische Isolierschichten verringern die Kapazitäten beträchtlich und sind sehr hochohmig, allerdings ist ihre Herstellung schwieriger und teurer.

J

JK-Flipflop
(J-K flip-flop)

Digitale → Folgeschaltung mit mindestens je einem J-, K- und C- (Takt-, Clock-)Eingang und häufig zwei komplementären Ausgängen. Das Taktsignal ruft je nach Kombination der beiden Datensignale an J und K eine Änderung des Ausgangszustandes hervor (s. Bild 5). Haben J und K 0-Stellung, so tritt keine Änderung am Ausgang auf. Haben J und K 1-Signal, so wird der Ausgangszustand invertiert (Frequenzteiler). 1 an J und 0 an K bewirkt 1 am Ausgang, 0 an J und 1 an K bewirkt 0 am Ausgang. JK-Flipflops werden als Frequenzteiler, Zähler, Speicher und als Schieberegisterstufen verwendet.

K

Kundenspezifizierte integrierte Schaltung
(custom integrated circuit)

Umfangreiche integrierte Schaltung, z.B. nur mit Verknüpfungsschaltungen (→ Gate array), aber auch mit → Folgeschaltungen, bei der die einzelnen Schaltungseinheiten noch nicht untereinander verbunden sind. Die letzte → Maskierung für die metallischen Zwischenverbindungen geschieht erst nach Angaben des Anwenders, der so auch kleine Serien einer speziellen Schaltung zu wirtschaftlichen Preisen erhält. Es kann und wird dabei oft ein Teil der auf dem → Chip vorhandenen Schaltungseinheiten ungenutzt bleiben.

L

LCD, Flüssigkristallanzeige
(LCD, liquid crystal display)

Eine Art der → digitalen Anzeigen, die Flüssigkristalle als passives Anzeigemedium verwenden. Der Flüssigkristall befindet sich zwischen den beiden Elektroden eines Plattenkondensators und ist transparent (durchsichtig), solange kein elektrisches Feld auf ihn einwirkt. Unter Feldeinwirkung wird er mehr oder weniger undurchsichtig. Flüssigkristallanzeigen sind passive Anzeigen, da sie selbst kein Licht aussenden. Der Kontrast hängt von den äußeren Beleuchtungsverhältnissen ab. Flüssigkristallanzeigen haben aber gegenüber → LED-Anzeigen den Vorteil eines äußerst geringen Leistungsbedarfs; deshalb werden sie hauptsächlich in batteriebetriebenen Geräten (Digitaluhren, Taschenrechnern) verwendet.

LED-Anzeigen
(light emitting diode displays)

Eine Art der → digitalen Anzeigen, die Leuchtdioden als aktives Anzeigemedium verwenden. Leucht- oder Lumineszenzdioden sind Gallium-Arsenid- oder Gallium-Phosphiddioden, die beim Durchfließen eines Stromes Licht bestimmter Wellenlänge aussenden. Sie haben deshalb stets farbiges Licht (rot, gelb, grün) und benötigen je nach Größe Stromstärken von einigen mA bis zu einigen 10 mA. Da die Flußspannung etwa 1 V beträgt, treten (im Gegensatz zu den → LCD-Anzeigen) ziemlich hohe Verlustleistungen auf. Ihr Vorteil liegt in dem hellen und gut erkennbaren Anzeigebild.

Logik
(logic)

Im Sprachgebrauch als Abkürzung für → logische Digitalfamilie gebraucht, z.B. TTL-Logik, ECL-Logik.

Logische Digitalfamilie
(digital-logic types)

Gesamtheit aller → logischen Schaltungen, die auf einer gemeinsamen Grundschaltung beruhen und miteinander beliebig kom-

binierbar sind, wobei gleiche Betriebsspannung, gleicher → logischer Hub und meist auch vergleichbare Schaltgeschwindigkeiten gegeben sind. Eine bestimmte Logikfamilie kann zu einer anderen kompatibel sein. Verbreitete Logikfamilien sind: 74xx-TTL, 4xxx-CMOS, 10xxx-ECL (die x stehen für fortlaufende Typenbezeichnungen).

Logische Schaltung
(logic circuit)

Elektrische Schaltung zur Darstellung einer logischen Operation, z.B. → UND-Schaltung. Kennzeichnend ist, daß der Zusammenhang zwischen Eingangssignalen und Ausgangssignal eindeutig festgelegt ist und in Form einer → Wertetabelle dargestellt werden kann.

Logische Variable
(logic variable)

Rechengröße der → Booleschen Algebra zur mathematischen Beschreibung logischer Zusammenhänge. Sie kann zwei feste Werte 1 und 0 annehmen, wobei 1 wahr und 0 falsch bedeutet. Logische Variable (Veränderliche) werden als Großbuchstaben geschrieben, z.B. $X = A \wedge B$ (gelesen: X gleich A UND B), dabei ist X die Ausgangsvariable, A und B sind die beiden Eingangsvariablen. Im Falle zweier Eingangsvariablen sind demnach 4 verschiedene Kombinationen möglich: A = 0 UND B = 0 (ergibt X = 0), A = 1 UND B = 0 (X = 0), A = 0 UND B = 1 (X = 0), A = 1 UND B = 1 (X = 1). Für jede dieser Kombinationen gibt es einen zugeordneten eindeutigen Wert für die Ausgangsvariable. Dieser Zusammenhang kann der besseren Übersichtlichkeit wegen in Form einer → Wertetabelle angegeben werden.

Logischer Hub
(logic swing)

Spannungsdifferenz zwischen den beiden logischen Signalzuständen → High und → Low einer Digitalschaltung. Da High und Low stets Spannungsbereiche sind, werden zur Berechnung des Hubs Mittelwerte angenommen.

Low, L, niedrig
(low)

Bezeichnung für denjenigen der beiden logischen Werte oder Wertebereiche einer binären Digitalschaltung, der das niedrigere Potential aufweist. Im Gegensatz dazu → High. Low kann der logischen 0 (positive Logik) oder der logischen 1 (negative Logik) entsprechen. Bei → TTL-Schaltungen ist Low z.B. 0 V (High = +5 V), bei → ECL-Schaltungen −1,8 V (High = −0,9 V).

Low-power-Schottky-TTL, LPSTTL
(low power Schottky TTL)

Sonderform der → Schottky-TTL-Schaltungen mit besonders niedriger Leistungsaufnahme (typisch 2 mW je Verknüpfungsschaltung) und trotzdem hoher Schaltgeschwindigkeit (typisch 8 ns).

LSI, Großintegration
(LSI, large scale integration)

Kennzeichnung für → monolithisch integrierte Schaltungen mit mehr als 1000 Grundschaltungen pro → Chip. Zur Unterscheidung → SSI, → MSI und → VLSI.

M

Maskenherstellung im Elektronenstrahlverfahren
(electron beam microfabrication)

Für extrem feine Strukturen (unter 1 μm) kann die herkömmliche → Fotolithografie nicht mehr verwendet werden, da selbst UV-Licht zu langwellig ist. Es wird deshalb an Stelle des Lichtstrahls ein Elektronenstrahl verwendet, der einen dünnen polymeren* Film auf der Halbleiteroberfläche belichtet. Die Strukturen werden durch rechnergesteuerte Bewegungen des Elektronenstrahls erzeugt.

Maskierung
(masking)

Herstellung von Masken, z.B. im → Fotolithografieverfahren für Diffusions- oder Ätzstrukturen. Als Maskenmaterial wird → Fotolack, Siliziumdioxid und teilweise auch ein aufgedampfter Metallfilm verwendet.

Master/Slave-Schaltung
(master slave circuit)
Logische → Folgeschaltung, die aus zwei hintereinandergeschalteten → Flipflops besteht. Eine Information wird z.B. mit der positiven Flanke des Taktimpulses in die erste Schaltung (Master) eingeschrieben, ohne daß sich das Ausgangssignal der zweiten Schaltung (Slave) ändert. Erst mit der negativen Flanke des Taktsignals erscheint die Information am Ausgang. Master/Slave-Schaltungen werden als → D- und → JK-Flipflops angeboten.

MBE
(MBE)

→ Molekularstrahlepitaxie

MESFET, Metall-Halbleiter-FET
(MESFET, metal-semiconductor FET)

Ein → Feldeffekttransistor, bei dem das Metall der Gateelektrode mit dem Halbleitermaterial des Drain-Source-Kanals einen Schottkyübergang bildet. Der MESFET arbeitet wie ein → PN-FET, ist aber einfacher herzustellen und wird sehr häufig in integrierten GaAs-Schaltungen verwendet.

MIC, integrierte Mikrowellenschaltung
(MIC, microwave integrated circuit)

Eine → Hybridschaltung, meist in → Dünnfilmtechnik für den Mikrowellenbereich. Sie besteht im wesentlichen aus Streifenleitungen und Halbleiter-→-Chips; Widerstände und Kondensatoren werden als Dünnfilmbauelemente, teilweise ebenfalls in Chipform verwendet (Bild 14).

Mikrocomputer
(microcomputer)

Allgemeine Bezeichnung für ein miniaturisiertes Computersystem aus integrierten Schaltungen, bestehend aus → Mikroprozessor, Speichern und Ein- und Ausgabeschaltungen, einschließlich der Betriebssoftware**.

Mikrominiaturisierung
(microminiaturization)

Verringerung der Bauteileabmessungen einer elektrischen Schaltung auf äußerst kleine Werte durch → monolithische Integration. Dadurch können mehrere tausend Bauelemente auf eine Fläche von 1 mm^2 konzentriert werden, → LSI, → VLSI.

---

* **polymer**: Aus großen Molekülen bestehend (Ausdruck aus der organischen Chemie).

** **Software**: Gesamtheit der *Programme* für eine Datenverarbeitungsanlage; im Gegensatz dazu **Hardware**: Gesamtheit der *Geräte* einer Datenverarbeitungsanlage.

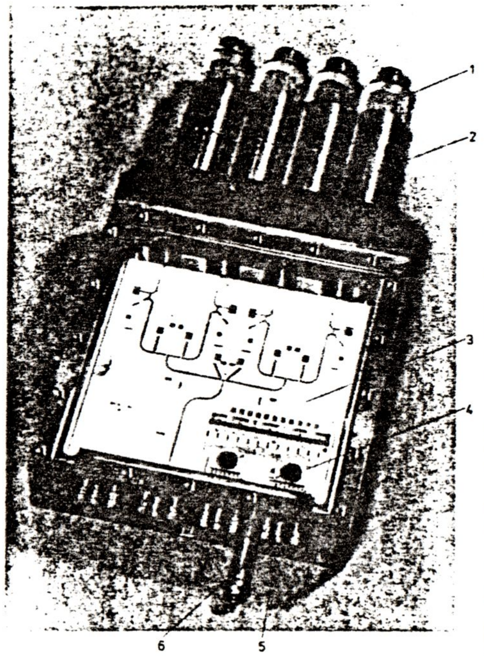


Bild 14: Integrierte Mikrowellenschaltung
Dünnfilmschaltung (Substratgröße 7,5 cm x 7,5 cm) für den 10-GHz-Bereich zur Antennensteuerung mit zusätzlich aufgebrachter Dickfilmschaltung
1 Strahler
2 Hohlleiter
3 Keramiksubstrat mit Dünnfilmschaltung
4 Keramiksubstrat mit Siebdruckschaltung
5 Koaxialbuchse
6 Durchführung für Steuerleitungen

Mikroprozessor
(microprocessor)

Zentrales Bauelement eines → Mikrocomputers, meist auf einem einzigen → Chip konzentriert, das aus der arithmetisch-logischen Einheit (zur Durchführung von Rechenoperationen und logischen Entscheidungen), dem Steuerblock und dem Registerfeld besteht. Mikroprozessoren können auch für sich betrieben werden und ersetzen dann fest verdrahtete Logikschaltungen.

MISFET
(MISFET, metal-insulator-semiconductor FET)

Allgemein ein → Feldeffekttransistor, bei dem die metallische Gateelektrode durch einen Isolator vom Halbleitermaterial des Drain-Source-Kanals getrennt ist. Hierzu gehören der → MOSFET und der → MNSFET.

MMIC
(MMIC, microwave monolithic integrated circuit)

→ monolithisch integrierte Mikrowellenschaltung.

MNOS, Metallnitrid-FET, MNS
(MNOS, metal-nitride silicon device)

Ein → Feldeffekttransistor, dessen → Gateelektrode zusätzlich zu einer dünnen SiO_2-Schicht noch eine Si_3N_4-Schicht (Siliziumnitrid) als Isolator zum Drain-Source-Kanal aufweist. Dadurch wird eine bessere Oberflächen-→-Passivierung und eine höhere Durchschlagsfestigkeit erreicht.

MNSFET
(MNSFET, metal-nitride-semiconductor FET)

Ein → Feldeffekttransistor, bei dem die metallische Gateelektrode durch eine Nitrid-Schicht vom Halbleitermaterial des Drain-Source-Kanals getrennt ist.

Molekularstrahlepitaxie, MBE
(molecular beam epitaxy, MBE)

Verfahren zum Herstellen epitaktischer Schichten, in dem ein gerichteter Strahl nichtionisierter Moleküle von einer Quelle auf Grund ihrer thermischen Energie auf ein Substrat (im allgemeinen ein Halbleitereinkristall) gerichtet wird. Mit der MBE lassen sich extrem dünne und sehr genau dotierte Schichten in beliebiger Reihenfolge erzeugen.

Monolithisch integrierte GaAs-Schaltung
(monolithic integrated GaAs-circuit)

Der Halbleiter → GaAs wird hauptsächlich für schnelle → GaAs-Digitalschaltungen, → MMICs und → monolithisch integrierte Millimeterwellenschaltungen verwendet. Da das undotierte GaAs Isolatoreigenschaften hat, können alle Bauelemente ohne Zwischenisolierungen nebeneinander angeordnet werden, und auch die leitenden Zwischenverbindungen können direkt auf die Substratoberfläche aufgebracht werden. Transistoren werden meistens als → MESFETs, Dioden als → Schottkydioden ausgeführt. Da GaAs für elektrooptische Bauelemente (Laser, LED, Fotodiode) geeignet ist, werden auch elektro-optische Integrationen, wie Treiberverstärker mit Laser oder Fotodiode mit Vorverstärker, realisiert.

Monolithisch integrierte Mikrowellenschaltung, MMIC
(microwave monolithic integrated circuit, MMIC)

Monolithisch integrierte Schaltung für den Mikrowellenbereich, größtenteils auf GaAs-, teilweise auch auf Siliziumbasis hergestellt, z.B. Schmal- und Breitbandverstärker, Mischer, Phasenschieber, Umschalter, Oszillatoren, Filterschaltungen oder Kombinationen solcher Baugruppen.

Monolithisch integrierte Millimeterwellenschaltung
(mm-wave monolithic integrated circuit)

Monolithisch integrierte Schaltung für den Millimeterwellenbereich, im allgemeinen auf GaAs-Basis.

Monolithisch integrierte Schaltung
(monolithic integrated circuit)

Hauptform der heute gebräuchlichen → integrierten Schaltungen, bei der alle Bauelemente in einen Halbleitereinkristall (Silizium, Gallium-Arsenid) durch → Diffusion oder → Ionenimplantation eingebaut sind. Alle aktiven (Dioden, Transistoren) und passiven (Widerstände, teilweise Leitungsverbindungen und Leitungskreuzungen, in geringem Maße auch Kondensatoren) Bauelemente werden in den Kristall eingebaut, die meisten metallischen Zwischenverbindungen und Anschlußflächen werden auf der Kristalloberfläche angebracht.

Monostabile Schaltung, monostabiler Multivibrator
(monostable circuit, single-shot circuit)

Eine → Digitalschaltung mit einem stabilen und einem nichtstabilen Ausgangszustand. Ein Eingangssignal bewirkt die Umschaltung des Ausgangs vom stabilen in den nichtstabilen Zustand.

Die Schaltung kehrt nach einer bestimmten Zeit, die durch die Größe eines RC-Gliedes vorgegeben ist, selbsttätig in den stabilen Zustand zurück. Monostabile Schaltungen werden z.B. benutzt, um aus einem kurzen Eingangs- (Trigger-)Impuls einen Impuls mit festgelegter Dauer zu machen.

MOS, Metalloxid-Halbleiter
(MOS, metal oxide semiconductor)

Ein → unipolares Halbleiterbauelement mit einem leitfähigen Kanal zwischen zwei metallischen Elektroden (Drain und Source), dessen Leitfähigkeit von einem elektrischen Feld beeinflußt wird. Dieses Feld entsteht durch Anlegen einer Spannung an die metallische dritte Elektrode (Gate), die vom Kanal durch eine Isolierschicht aus Siliziumdioxid (SiO_2) getrennt ist (s. Bild 9).

MOSFET, Metalloxid-Feldeffekttransistor
(MOSFET, metal oxide semiconductor field effect transistor)

Ein spezieller → Feldeffekttransistor mit → MOS-Struktur. MOSFETs werden mit p- oder n-leitendem Kanal hergestellt, und zwar jeweils als Anreicherungs- (enhancement type) oder Verarmungstyp (depletion type), so daß insgesamt vier unterschiedliche Transistorarten vorliegen.

MSI, mittlere Integrationsdichte
(MSI, medium scale integration)

Kennzeichnung für → monolithisch integrierte Schaltungen mit rund 100 bis 1000 Grundschaltungen pro → Chip. Zur Unterscheidung → SSI, → LSI, → VLSI.

Multichip-Schaltung
(multichip integrated circuit)

Zusammenschaltung mehrerer Einzel-→-Chips in einem gemeinsamen Gehäuse. Die Chips werden dabei entweder direkt miteinander und mit den Anschlußstiften des Gehäuses verbunden, oder sie befinden sich auf einer → Siebdruck- oder → Dünnfilmschaltung.

N

NICHT-Schaltung, Inverter
(NOT gate, negation gate)

Logische → Digitalschaltung, welche die NICHT-Funktion darstellt. Sie bildet als Ausgangssignal stets den *Komplementärwert* des Eingangssignals, wandelt also 1-Signale in 0-Signale um und umgekehrt. Mathematische Beschreibung: $X = \overline{A}$ (gelesen: X = A NICHT).

NICHT-ODER-Schaltung, NOR-Schaltung
(NOR gate, NOT OR gate)

Logische → Digitalschaltung, welche die NICHT-ODER-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 0, wenn mindestens eines, gleichgültig welches, der Eingangssignale den Wert 1 hat (s. Bild 13). Mathematische Beschreibung: $X = \overline{A \vee B}$ (gelesen: X = A ODER B, NICHT). Die NICHT-ODER-Schaltung ist eine Grundschaltung, mit deren Hilfe jede andere logische Schaltung aufgebaut werden kann.

NICHT-UND-Schaltung, NAND-Schaltung
(NAND gate, NOT AND gate)

Logische → Digitalschaltung, welche die NICHT-UND-Funktion darstellt. Ihr Ausgangssignal hat nur dann den Wert 0, wenn Eingangssignal A UND Eingangssignal B den Wert 1 haben (s. Bild 16). Mathematische Beschreibung: $X = \overline{A \wedge B}$ (gelesen: X = A UND B, NICHT). Die NICHT-UND-Schaltung ist eine Grundschaltung, mit deren Hilfe jede andere logische Schaltung aufgebaut werden kann.

NMOS, n-Kanal-MOS
(NMOS)

Bezeichnung für Schaltungen oder Schaltungsfamilien, die mit n-Kanal-→-MOS-Bauelementen aufgebaut sind. Dies sind Bauelemente, bei denen der Drain-Source-Kanal n-→-dotiert ist, also Elektronenüberschuß aufweist. NMOS ist geeignet für → VLSI-Schaltungen und hat günstigere elektrische Eigenschaften (Leistungsaufnahme, Schaltgeschwindigkeit) als die → PMOS-Technik.

O

ODER-Schaltung, OR-Schaltung
(OR gate)

Logische → Digitalschaltung, welche die ODER-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn mindestens eines, gleichgültig welches, der Eingangssignale den Wert 1 hat (s. Bild 16). Mathematische Beschreibung: $X = A \vee B$ (gelesen: X = A ODER B).

ODER-NICHT-Schaltung, Inklusionsschaltung
(OR NOT gate, inclusion gate)

Logische → Digitalschaltung, welche die ODER-NICHT-Funktion darstellt. Ihr Ausgangssignal hat dann den Wert 1, wenn entweder das Eingangssignal A den Wert 1 oder das Eingangssignal B den Wert 0 hat oder wenn beides erfüllt ist (s. Bild 16). Mathematische Beschreibung: $X = A \vee \overline{B}$ (gelesen: X = A ODER B-NICHT).

Operationsverstärker, Rechenverstärker
(operational amplifier)

Fast immer ein → Gleichstromverstärker, der ursprünglich in Analogrechnern zur Durchführung von Rechenvorgängen (Addition, Multiplikation, Integration, Differentiation) verwendet wurde, heute, besonders als monolithisch integrierter Operationsverstärker, das vielseitigste Analogbauelement überhaupt geworden ist. Operationsverstärker arbeiten nach dem Prinzip des → Differenzverstärkers, bestehen aber aus mehreren hintereinandergeschalteten Verstärkerstufen. Die Ausgangsspannung ist gleich der Differenz der Eingangsspannungen, multipliziert mit dem Verstärkerfaktor. Gleich große Eingangsspannungen haben demgegenüber praktisch keinen Einfluß auf die Ausgangsspannung. Operationsverstärker haben Verstärkungsfaktoren bis zu 100 000, hohe Eingangswiderstände (besonders, wenn als Eingangstransistoren → FETs verwendet werden) und einen niedrigen Ausgangswiderstand. Die Verstärkung ist frequenzabhängig, sie ist von 0 Hz bis zur oberen Grenzfrequenz (einige kHz bis einige MHz) konstant und fällt dann mit etwa 6 dB/Oktave* ab.

Operationsverstärker werden üblicherweise mit Gegenkopplung betrieben, wodurch die Leerlaufverstärkung (Verstärkung ohne Gegenkopplung) auf eine bestimmte Schleifenverstärkung (Verstärkung mit Gegenkopplung) herabgesetzt wird. Mit abnehmender Schleifenverstärkung nimmt die Bandbreite des Verstärkers zu (das Produkt aus Verstärkung und Bandbreite ist konstant). Wird die Gegenkopplung frequenzabhängig gemacht, erhält man Filterschaltungen. Bei positiver Gegenkopplung (Rückkopplung) können Operationsverstärker als freischwingende Generatoren (Sinus, Rechteck, Dreieck, Sägezahn) verwendet werden.

Optokoppler
(optical coupler, optical insulator)

Optokoppler bestehen aus einer → LED und einem Phototransistor (Transistor, der nicht über den Basisstrom aufgrund einer angelegten Spannung, sondern durch Licht angesteuert wird). Ein Steuerstrom durch die LED läßt diese Licht emittieren, das den Transistor beleuchtet und in diesem einen um den Faktor 100 bis 200 höheren Kollektor-Emitterstrom hervorruft. Ein- und Ausgangsstrom sind wegen der nichtleitenden Lichtstrecke galvanisch voneinander isoliert (bis zu 5 kV Durchschlagsfestigkeit bei Dual-in-Line-Gehäusen). Optokoppler können sowohl zur analogen, als auch zur digitalen Signalübertragung verwendet werden, im letzteren Fall sind Datenraten bis zu einigen 10 Mbit/s möglich.

* Oktave: Frequenzbereich oder Abstand zwischen einer unteren Frequenz und dem doppelten Wert dieser Frequenz.

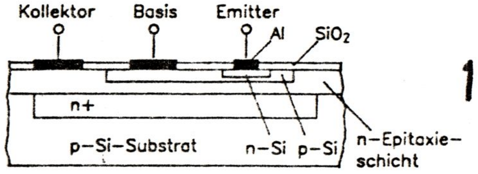


Bild 15 Aufbau eines planaren npn-Epitaxietransistors im Querschnitt

P

Packungsdichte
(packaging density)

Gibt die Zahl der Bauelemente pro Volumeneinheit (Raumeinheit) einer integrierten Schaltung an. Je höher die Packungsdichte, um so höher der Miniaturisierungsgrad. Heute werden Packungsdichten von vielen tausend Bauelementen pro mm^3 erreicht. Mit wachsender Packungsdichte muß jedoch gleichzeitig dafür gesorgt werden, daß die Verlustleistung pro Bauelement entsprechend verringert wird, damit die entstehende Verlustwärme nach außen abgeführt werden kann und nicht den Kristall unzulässig aufheizt.

Parasitäre Kapazität
(parasitic capacitance)

Die Bauelemente einer integrierten Schaltung sind zwar gegeneinander und gegenüber dem → Substrat isoliert (durch gesperrte PN-Übergänge oder → dielektrische Isolation), aber doch so dicht benachbart, daß zwischen ihnen und dem Substrat unerwünschte (parasitäre) Kapazitäten auftreten, so daß die elektrischen Eigenschaften (obere Grenzfrequenz, Schaltzeiten) negativ beeinflußt werden. Sie sind nicht zu vermeiden, können aber durch eine dielektrische Isolation stark herabgesetzt werden.

Passivierung, Oberflächenpassivierung
(surface passivation)

Letzter Schritt bei der Herstellung eines Transistors oder einer monolithisch integrierten Schaltung vor dem Aufdampfen der Metallisierung für die Anschlußflächen und Zwischenverbindungen. Auf die Halbleiteroberfläche wird dazu eine Siliziumdioxid- oder Siliziumnitridschicht aufgebracht; sie dient einerseits als Isolator zwischen Bauelementen und Metallisierung, andererseits als Schutz besonders der offenliegenden → PN-Übergänge gegen Feuchtigkeit und atmosphärische Verunreinigungen.

PLA, programmierbares logisches Feld
(PLA, programmable logic array)

Eine Anordnung von → UND- und → ODER-Schaltungen in Form einer Matrix*. Die einzelnen Bausteine können entweder vom Hersteller nach Kundenangaben durch eine entsprechende Metallisierung verdrahtet werden, oder der Kunde kann sie selbst programmieren, indem er bestimmte Verbindungen, die in Form von Schmelzbrücken angebracht sind, durchtrennt (ähnlich wie beim → FROM).

Planarprozeß
(planar process)

Heute allgemein verwendeter Herstellungsprozeß für Transistoren, Dioden und monolithisch integrierte Schaltungen, bei dem die Bauelemente nur in einer dünnen Schicht (einige µm) unter der Kristalloberfläche eingebaut werden und alle Anschlüsse auf der Oberfläche liegen (Bild 15). Der Rest des (einige 100 µm) dicken Kristalls dient als → Substrat.

PLL, phasengeregelte Schaltung
(PLL, phase-locked loop circuit)

Schaltung zur Erzeugung einer Sinus- oder Rechteckschwingung, deren Frequenz und Phase auf ein Referenzsignal synchronisiert wird, dessen Frequenz gleich, ein ganzzahliges Vielfaches oder ein ganzzahliger Bruchteil ist. Eine PLL-Schaltung besteht aus einer Phasenvergleichsschaltung, einem Verstärker, einem Tiefpaßfilter, einem spannungsgesteuerten Oszillator (VCO, voltage controlled oscillator) und oft auch aus → Frequenzteilerstufen. Sobald Frequenz oder Phase der erzeugten Schwingung mit dem Referenzsignal nicht mehr übereinstimmen, gibt die Phasenvergleichsschaltung eine Korrekturspannung ab, die den Oszillator in Frequenz und/oder Phase entsprechend nachregelt.

PMOS, p-Kanal-MOS
(PMOS)

Bezeichnung für Schaltungen oder Schaltungsfamilien, die mit p-Kanal → MOS-Bauelementen aufgebaut sind. Dies sind Bauelemente, bei denen der Drain-Source-Kanal p-→-dotiert ist, also Löcherüberschuß aufweist. Da Löcher eine geringere Beweglichkeit haben als Elektronen, sind PMOS-Bauelemente langsamer als solche vom → NMOS-Typ. Dies kann durch eine Verbreiterung des Kanals ausgeglichen werden, wodurch aber PMOS-Transistoren größer werden und PMOS-Schaltungen eine geringere → Packungsdichte aufweisen.

PN-Übergang
(p-n junction)

Die Grenze zwischen einem p- und einem n → dotierten Halbleiter, Grundlage für alle → bipolaren Halbleiterbauelemente (Dioden, Transistoren, integrierte Schaltungen). Der pn-Übergang hat Gleichrichterverhalten. In Sperrichtung gepolt ist er hochohmig (es fließt nur ein vernachlässigbar kleiner Reststrom), in Durchlaßrichtung gepolt ist er nach Überschreiten einer Schwellenspannung (Germanium etwa 0,3 V, Silizium etwa 0,7 V) niederohmig. In beiden Richtungen ist eine starke Temperaturabhängigkeit zu beobachten.

PROM, programmierbarer Festwertspeicher
(PROM, programmable read only memory)

Ein Festwertspeicher (→ ROM), der vom Anwender programmiert werden kann. Es gibt PROMs, bei denen nur eine einmalige Programmierung möglich ist, und solche, die beliebig oft neu programmiert werden können (→ EAROM, → EEROM, → EPROM).

Programmierbare Logic
(programmable logic)

Anordnungen von Logikschaltungen oder/und → sequentiellen Schaltungen, die vom Anwender nach Maßgabe seiner jeweiligen Systemforderungen programmiert werden können. Hierzu ge-

* Matrix: Zeilen- und spaltenweise Anordnung von Buchstaben, Zahlen oder Bauelementen.

hören → PLA, → RAM, → ROM, → PROM und → Mikroprozessor. Im Gegensatz hierzu die festverdrahtete Logik, bei der die Systemforderungen durch eine eigens entworfene und nachträglich nicht mehr wesentlich veränderbare Hardware verwirklicht werden.

R

RAM, Speicher mit wahlfreiem Zugriff, Schreib-/Lese-Speicher
(RAM, random access memory)

Speicher, bei dem auf jedes Speicherelement durch eine Adressierung jederzeit zugegriffen werden kann, und zwar sowohl zum Lesen als auch zum Schreiben. Es besteht aus einer bestimmten Zahl von Speicherzellen, die meistens wortweise organisiert sind, den Schreib- und Leseverstärkern und einem oder mehreren Adressendecodern. RAMs werden als Arbeitsspeicher in Datenverarbeitungssystemen verwendet.

RCTL, Widerstand-Kondensator-Transistor-Logik
(RCTL, resistor-capacitor-transistor-logic)

Veraltete → Logikfamilie, ähnlich der → RTL, mit zusätzlichen Koppelkondensatoren zur Erhöhung der Schaltgeschwindigkeit (s. Bild 2).

Register
(register)

Eine lineare Anordnung von Speicherbauelementen (z. B. → D-Flipflops, → JK-Flipflops) mit gemeinsamer Taktansteuerung, die entweder parallel oder seriell betrieben wird und z. B. ein Datenwort zwischenspeichert (Speicherregister) oder parallel-serien-wandelt (→ Schieberegister).

Ringzähler
(ring counter)

Ein vom Ausgang auf den Eingang zurückgekoppeltes → Schieberegister, bei dem eine feste Information immer im Kreis herumgeschoben wird. Mit jedem Taktimpuls ändert die Information ihre Position dabei innerhalb der Schaltung um eine Stelle. Der Ringzähler kann zur Freqenzteilung oder zur Impulszählung verwendet werden.

Ripple-Zähler
(ripple counter)

Hintereinanderschaltung von → Frequenzteilerstufen, wobei das Ausgangssignal der ersten Stufe die zweite Stufe ansteuert, das der zweiten Stufe die dritte usw. Wird zur Frequenzteilung und Impulszählung verwendet und auch als Asynchronteiler oder Asynchronzähler bezeichnet, weil Ausgangssignale der einzelnen Stufen nicht zeitgleich, sondern wegen der → Verzögerungszeiten der Stufen zeitlich gegeneinander versetzt erscheinen.

ROM, Festwertspeicher, Nur-Lese-Speicher
(ROM, read only memory)

Im Gegensatz zum → RAM kann beim ROM der Speicherinhalt nicht geändert werden. Der bei der Herstellung fest einprogrammierte Speicherinhalt kann nur beliebig oft ausgelesen werden.

RS-Flipflop
(R-S flip-flop)

Grundbaustein aller → Flipflops, bestehend z. B. aus zwei Transistoren in Emitterschaltung oder aus zwei → NICHT-UND-Schaltungen (s. Bild 10) mit zwei Eingängen S (*set:* setzen) und R (*reset:* zurücksetzen, löschen) und meist zwei komplementären Ausgängen. Ein 1-Signal am S-Eingang (bei gleichzeitig 0 an R) ruft ein 1-Signal am Ausgang hervor, ein 1-Signal am R-Eingang (bei gleichzeitig 0 an S) ruft ein 0-Signal am Ausgang hervor. Ist S = 0 und R = 0, so bleibt der Ausgangszustand unverändert (Speicherung der zuletzt eingegebenen Information). 1-Signal sowohl an S als auch an R stellt einen verbotenen Zustand dar, da in diesem Fall kein eindeutiges Ausgangssignal zugeordnet ist. Das RS-Flipflop ist eine → bistabile Schaltung, es wird auch als $\overline{RS}$-Flipflop (gelesen: R-NICHT-S-NICHT) hergestellt: hier sind die Eingänge invertiert, d. h. ein 0-Signal an S (bei gleichzeitig 1 an R) ergibt am Ausgang 1, ein 0-Signal an R (bei gleichzeitig 1 an S) ergibt 0. S = 0 und R = 0 ist der verbotene Zustand.

RST-Flipflop
(R-S-T flip-flop)

Ein → RS-Flipflop mit einem dritten Eingang T. Die Eingangssignale an R und S haben nur dann Einfluß auf den Ausgang, wenn ein Signal an T anliegt. R und S wirken als Vorbereitungseingänge, T als Takteingang (s. Bild 10). Nicht zu verwechseln mit dem → T-Flipflop.

RTL, Widerstand-Transistor-Logik
(RTL, resistor-transistor-logic)

Veraltete → Logikfamilie, die nur aus Transistoren und Widerständen besteht (s. Bild 3). Die logische Verknüpfung wird in Widerstandsnetzwerken durchgeführt, der Transistor dient zur Verstärkung und als → Inverter.

S

Saphir
(sapphire)

Ein Edelstein, der als → Substrat für hochwertige integrierte Schaltungen verwendet wird, → SOS-Technik.

Schaltungen mit diskreten Bauelementen
(discrete component circuit)

Schaltungen, die im Gegensatz zu → integrierten Schaltungen aus einzelnen (diskreten) Bauelementen (Widerstände, Transistoren, Kondensatoren usw.) aufgebaut werden.

Schieberegister
(shift register)

Schaltung aus z. B. → D-Flipflops oder → JK-Flipflops, die gemeinsam getaktet werden und bei der die Information einer Stufe an die benachbarte Stufe übergeben wird. Die Information kann parallel oder seriell ein- und ausgelesen werden, sie kann von links nach rechts oder von rechts nach links durch das Schieberegister geschoben werden. Schieberegister können zur Serien-Parallel- oder Parallel-Serien-Wandlung verwendet werden.

Schmitt-Trigger
(Schmitt trigger)

Schaltung aus zwei Transistoren mit interner Rückkopplung, die nur zwei bestimmte Ausgangsspannungen (z. B. 0 V und +5 V) hat und von einer Eingangsspannung angesteuert wird. Je nach Höhe der Eingangsspannung tritt entweder die eine oder die andere Ausgangsspannung auf, wobei die Umschaltung von der niedrigen zur hohen Ausgangsspannung bei einer anderen Eingangsspannung auftritt als die Umschaltung von der hohen zur niedrigen (Schalthysterese). Schmitt-Trigger werden zur Impulsformung oder Impulsregenerierung verwendet, z. B. um ein Sinussignal in ein Rechtecksignal oder einen verwaschenen Impuls in einen Impuls mit steilen Flanken umzuformen.

Schottky-Diode
(Schottky diode)

Halbleiterbauelement mit Diodencharakteristik, auch Hot carrier-Diode (Diode mit hoher Ladungsträgerbeweglichkeit) genannt, das nicht aus einem → PN-Übergang, sondern aus einem Metall-Halbleiterübergang besteht. Der Ladungstransport geschieht nur durch **einen** Ladungsträgertyp (Elektronen oder Löcher). Schottky-Dioden haben sehr kurze Schaltzeiten und wesentlich geringere Flußspannungen als Dioden mit PN-Übergang. Sie werden in zunehmendem Maße auch in integrierten Digitalschaltungen (→ Schottky-TTL, LPSTTL) verwendet, um deren Schaltgeschwindigkeit zu erhöhen.

Schottky-TTL, Transistor-Transistor-Logik mit Schottkydioden
(Schottky TTL)

Die Schaltzeit von logischen Schaltungen, bei denen die Schalttransistoren bis in den Sättigungsbereich durchgesteuert werden (z. B. → DTL, TTL), hängt ganz wesentlich davon ab, wie schnell die in der Basis gespeicherten Ladungsträger wieder abfließen können. Durch Schottky-Klemmdioden parallel zur Basis-Emitterstrecke der Schalttransistoren wird der Sättigungsvorgang vermieden und damit die Schaltgeschwindigkeit gesteigert, allerdings auf Kosten einer höheren Leistungsaufnahme (s. Bild 3). Schottky-geklemmte Schalttransistoren werden z. B. in → TTL-Schaltungen verwendet, die dann als Schottky-TTL bezeichnet werden, s. auch → LPSTTL.

Siebdruck
(screen printing)

Verfahren zur Herstellung von gedruckten Strukturen, bei dem ein durch → Fotolithografie maskiertes planes (ebenes) Sieb auf ein → Substrat gelegt und durch das ein Druckmittel (Siebdruckpaste) gestrichen wird. In einem Trocken- und Einbrennprozeß werden die gedruckten Strukturen fest mit dem Substrat verbunden. Siebdruck wird in der Elektrotechnik zur Herstellung von → Dickfilmschaltungen verwendet, indem Pasten mit bestimmten elektrischen Eigenschaften (Leiterpasten, Dielektrikumspasten, Widerstandspasten) in mehreren Ebenen übereinandergedruckt werden.

Siliziumdioxid, SiO_2
(silicon dioxide, SiO_2)

Oxid des Siliziums, das ein ausgezeichneter elektrischer Isolator und gegenüber den meisten Säuren und Ätzmitteln resistent* ist. Es wird zur → dielektrischen Isolierung und Oberflächen-→-passivierung von Halbleiterbauelementen verwendet.

Siliziumnitrid, Si_3N_4
(silicon nitride, Si_3N_4)

Nitrid des Siliziums, das zum → Passivieren von integrierten Schaltungen verwendet wird und die Halbleiteroberfläche schützt. Meist wird es als **erste** Passivierung verwendet, indem es direkt auf die Halbleiteroberfläche, noch unterhalb der Metallisierung für die Zwischenverbindungen und Anschlußflächen, gebracht wird. Die Metallisierung wird dann zusätzlich durch eine → Siliziumdioxidschicht geschützt.

SOS, Silizium auf Saphir
(SOS, silicon on sapphire)

Verfahren zur Herstellung monolithischer integrierter Schaltungen, bei dem als → Substrat nicht ein Siliziumeinkristall, sondern → Saphir verwendet wird, der sehr gute Isolationseigenschaften hat. Auf den Saphirkristall wird eine Silizium-→-epitaxieschicht aufgebracht, in die alle Bauelemente eingebaut werden. Danach wird alles überflüssige Silizium weggeätzt, so daß die Bauelemente als voneinander isolierte Inseln stehen bleiben. Man verringert auf diese Weise die → parasitären Kapazitäten beträchtlich und bekommt Schaltungen mit sehr kurzen Anstiegs- und Abfallszeiten. Die SOS-Technik wird hauptsächlich für → MOS-Schaltungen verwendet.

Speicher, Speicherschaltung
(memory)

Grundsätzlich eine Anordnung zur Speicherung von Information (z. B. Kernspeicher, Magnetplattenspeicher); hier: Halbleiterspeicher, eine → Digitalschaltung mit einer mehr oder weniger großen Zahl von → binären Speicherzellen, meist in Form von → Flipflops, und Hilfsschaltungen, wie Schreib- und Leseverstärkern, Adressendecodern. Man unterscheidet flüchtige und nichtflüchtige Speicher; bei den nichtflüchtigen bleibt die eingespeicherte Information auch nach Abschalten der Betriebsspannung im Speicher erhalten. Speicher können als → RAM, → ROM oder → PROM aufgebaut sein, sie können aus → bipolaren oder → unipolaren Bauelementen bestehen.

SSI, Kleinintegration
(SSI, small scale integration)

Kennzeichnung für → monolithische integrierte Schaltungen mit bis zu einigen 10 Grundschaltungen pro → Chip. Zur Unterscheidung → MSI, → LSI, → VLSI.

Subnanosekunden-ECL
(sub-nanosecond ECL)

Eine → ECL-Familie, deren Schaltungen Schaltzeiten von unter 1 ns aufweisen (→ Verzögerungszeit einer ODER-Schaltung z. B. 0,7 ns). Es sind dies die zur Zeit „schnellsten Schaltungen" auf Siliziumgrundlage.

Substrat, Trägermaterial
(substrate)

Allgemein die Unterlage, auf der oder in der sich eine integrierte Schaltung befindet. Bei monolithischen Schaltungen ist dies fast immer der Halbleitereinkristall (Ausnahme → SOS), bei → Hybridschaltungen Keramik oder Glas. Das Substrat sorgt in erster Linie für die mechanische Festigkeit und Handhabbarkeit einer Schaltung, aber auch für die Wärmeabfuhr und für die elektrische Isolation.

Superchip
(superchip)

Bezeichnung für → Chips mit besonders großer Fläche (einige cm^2), auf denen sich Schaltungen (zur Zeit Speicherschaltungen) mit mehreren 100000 Bauelementen befinden. Beispiel: ein RAM mit 64 k (65536 bit) Speicherkapazität auf einem etwa 6 cm^2 großen Chip.

T

TEGFET
(TEGFET, two-dimensional electron gas FET)

→ HEMT

T-Flipflop
(T flip-flop)

Eine → bistabile Schaltung mit einem Eingang und einem Ausgang oder einem komplementären Ausgangspaar (s. Bild 5). Ein Eingangssignal (Spannungswert oder Flanke) bewirkt ein Umschalten des Ausgangssignals. T-Flipflops werden als → Frequenzteiler verwendet.

Transistor
(transistor)

Allgemeine Bezeichnung für ein verstärkendes Halbleiterbauelement mit drei Elektroden. Zu unterscheiden sind → bipolare Transistoren (NPN-, PNP-Transistor) und → unipolare Transisto-

* **resistent:** widerstandsfähig, unangreifbar

ren (→ FET). Beim bipolaren Transistor wird der Strom zwischen Kollektor und Emitter über einen Steuerstrom beeinflußt, der zwischen Basis und Emitter fließt. Beim unipolaren Transistor wird der Widerstand zwischen Drain und Source durch ein elektrisches Feld verändert, das durch eine Gatespannung erzeugt wird.

TTL, T²L, Transistor-Transistor-Logik
(TTL, T²L, transistor-transistor logic)
(Sprich: „te-Quadrat el")

Eine digitale → Logikfamilie, bei der die logische Verknüpfung mit Transistoren (Multiemitter-Transistoren) durchgeführt wird (s. Bild 3). Es ist dies die heute am häufigsten benutzte Logikfamilie, auch in ihren Weiterentwicklungen als → Schottky-TTL und → LPSTTL, und wird in einer großen Zahl von Schaltungstypen angeboten, die teilweise sehr umfangreich sind. Sie hat jedoch den Nachteil eines hohen Leistungsbedarfs und einer geringen Störunempfindlichkeit, auch ist ihre Schaltgeschwindigkeit für viele Anwendungen zu gering.

T³L, Transistor-Transistor-Transistor-Logik
(T³L, transistor-transistor-transistor logic)
(Sprich: „te hoch drei el")

Eine spezielle Form der → TTL, bei der ein zusätzlicher Transistor verwendet wird, um die Ausgangsleistung zu erhöhen (höherer → Ausgangsfächer) und die Störempfindlichkeit zu verringern.

U

UFL
(UFL, unbuffered FET logic)

Eine Familie von → GaAs-Digitalschaltungen, die aus → DFETs und → Schottkydioden besteht (s. Bild 2). Die Transistoren T_1 und T_2 bilden die logische Eingangsverknüpfung, die Transistoren T_3 und T_4 sind als ohmsche Widerstände geschaltet, die Dioden D_1 und D_2 sind Pegelschiebedioden, die den Ausgangspegel an den Eingangspegel angleichen. Die UFL hat gegenüber der → BFL eine geringere Leistungsaufnahme, aber auch eine niedrigere Ausgangsbelastbarkeit.

UND-Schaltung, AND-Schaltung
(AND gate)

Logische → Digitalschaltung, welche die UND-Funktion darstellt. Ihr Ausgangssignal hat dann *und nur dann* den Wert 1, wenn beide Eingangssignale den Wert 1 haben (Bild 16). Mathematische Beschreibung: $X = A \wedge B$ (gelesen: X = A UND B).

Unipolare Bauelemente
(unipolar devices)

Halbleiterbauelemente, in denen für den Stromtransport nur ein einziger Typ von Ladungsträgern (Elektronen oder Löcher) verwendet wird (z.B. → Feldeffekttransistor). Im Gegensatz dazu → bipolare Bauelemente.

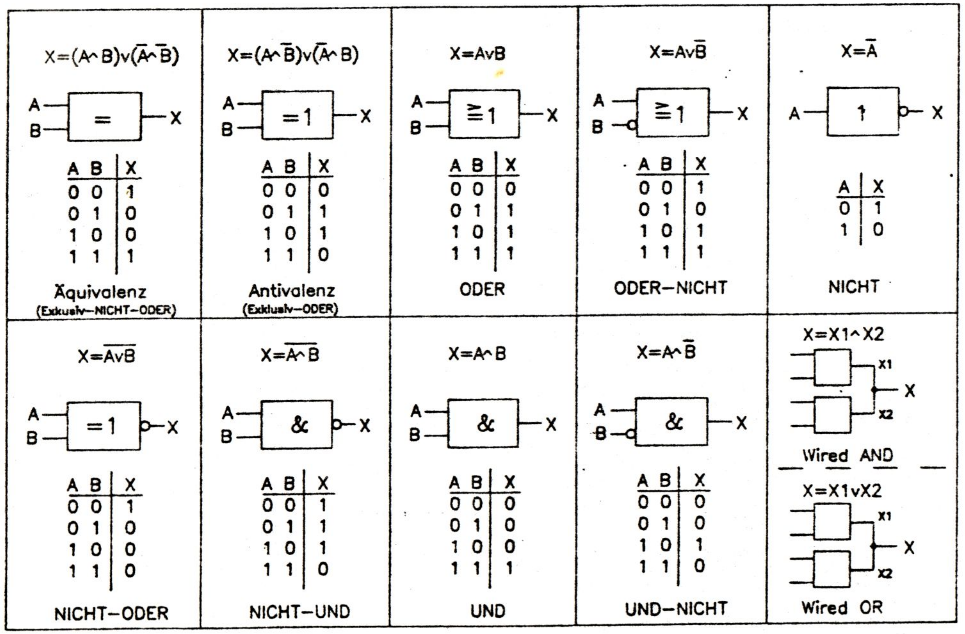


Bild 16. Die wichtigsten logischen Verknüpfungsschaltungen mit ihren Schaltsymbolen und mit Wertetabellen
A und B sind die Eingänge, X die Ausgänge

UND-NICHT-Schaltung, Exklusionsschaltung
(AND NOT gate, exclusion gate)

Logische → Digitalschaltung, welche die UND-NICHT-Funktion darstellt. Ihr Ausgangssignal hat nur dann den Wert 1, wenn das Eingangssignal A den Wert 1 und gleichzeitig das Eingangssignal B den Wert 0 hat (s. Bild 16). Mathematische Beschreibung: $X = A \wedge \overline{B}$ (gelesen: X = A UND B NICHT).

V

Vergrabene Schicht
(buried layer)

Ein Bereich mit sehr stark → dotiertem Material unter dem Kollektorgebiet von Planartransistoren zur Verringerung des Kollektorwiderstandes. Dabei werden die betreffenden Bereiche der Oberfläche z.B. durch → Diffusion dotiert, und anschließend wird das eigentliche Kollektormaterial durch → Epitaxie aufgebracht. Es folgen danach die Diffusion für Emitter und Basis

Verknüpfungsschaltung
(gate)

Logische → Digitalschaltung, die eine bestimmte logische Funktion oder Kombinationen von logischen Funktionen darstellt (z.B. UND-, ODER-NICHT-Schaltung). Je nach Zahl der Eingänge ergibt sich eine feste Zahl von Kombinationen der Eingangssignale (bei 2 Eingängen $2^2 = 4$, bei n Eingängen 2^n), für jede dieser Kombinationen gibt es *ein und nur ein* Ausgangssignal. Den Zusammenhang stellt man der besseren Übersichtlichkeit wegen in Form einer → Wertetabelle dar. Im Gegensatz zu → Folgeschaltungen ist das Ausgangssignal fest mit den Eingangssignalen verknüpft und ändert sich mit jeder Änderung eines Eingangssignals.

Verzögerungszeit, Laufzeit
(delay time, propagation delay time)

Diejenige Zeit, die ein Signal zum Durchlaufen einer digitalen Schaltung vom Eingang bis zum Ausgang benötigt. Sie wird gemessen zwischen dem Punkt mit 50% der Amplitude einer Flanke des Eingangssignals und dem Punkt mit 50% der Amplitude der zugehörigen Flanke des Ausgangssignals (s. Bild 1).

VLSI, Größtintegration
(VLSI, very large scale integration)

Kennzeichnung für → monolithische integrierte Schaltungen mit über 100000 Bauelementen pro → Chip. Zur Unterscheidung → SSI, → MSI, → LSI.

Volladdierer
(full adder)

Digitalschaltung, bestehend aus zwei → Halbaddierern zur Addition zweier einstelliger Binärzahlen unter Berücksichtigung des Übertrages der nächstniedrigeren Addition, wobei als Ausgangssignal das Summensignal und das Übertragssignal zur Verfügung stehen (s. Bild 12). Zur Addition mehrstelliger Binärzahlen müssen so viele Volladdierer parallel betrieben werden, wie die größte zu addierende Binärzahl Stellen hat. Ein 10-bit-Addierer z.B. kann zwei 10stellige Binärzahlen addieren, er hat 2 x 10 Eingänge und 11 Ausgänge (10 Summen- und einen Übertragsausgang).

Vorwärts-/Rückwärtszähler
(up/down counter)

Eine besondere Form des → Binärzählers, bei dem elektrisch zwischen Vorwärts- (Aufwärts-) und Rückwärts- (Abwärts-)Zählen umgeschaltet werden kann.

W

Wertetabelle, Wahrheitstafel
(truth table)

Auflistung aller möglichen Kombinationen von Eingangssignalen mit den zugehörigen Ausgangssignalen für eine logische Funktion oder Schaltung. Bei n Eingangssignalen ergeben sich 2^n Kombinationen (vgl. Wertetabellen in Bild 16). Bei → Folgeschaltungen kommt noch als weiteres Eingangssignal das Taktsignal hinzu, auf das man jedoch im allgemeinen verzichtet, weil das Taktsignal Voraussetzung für die Funktion einer Folgeschaltung ist und für Taktsignal 0 keinerlei Änderungen im Ausgangssignal auftreten.

Wired-AND, Verdrahtetes UND
(wired-AND)

Eine spezielle Form der UND-Schaltung, die durch einfaches Verbinden der Ausgänge mehrerer logischer → Digitalschaltungen entsteht (s. Bild 16). An dem Verbindungspunkt kann nur dann 1-Signal vorkommen, wenn alle Ausgänge 1-Signal haben. Das Verdrahtete UND ist z.B. bei manchen → TTL-Schaltungen möglich.

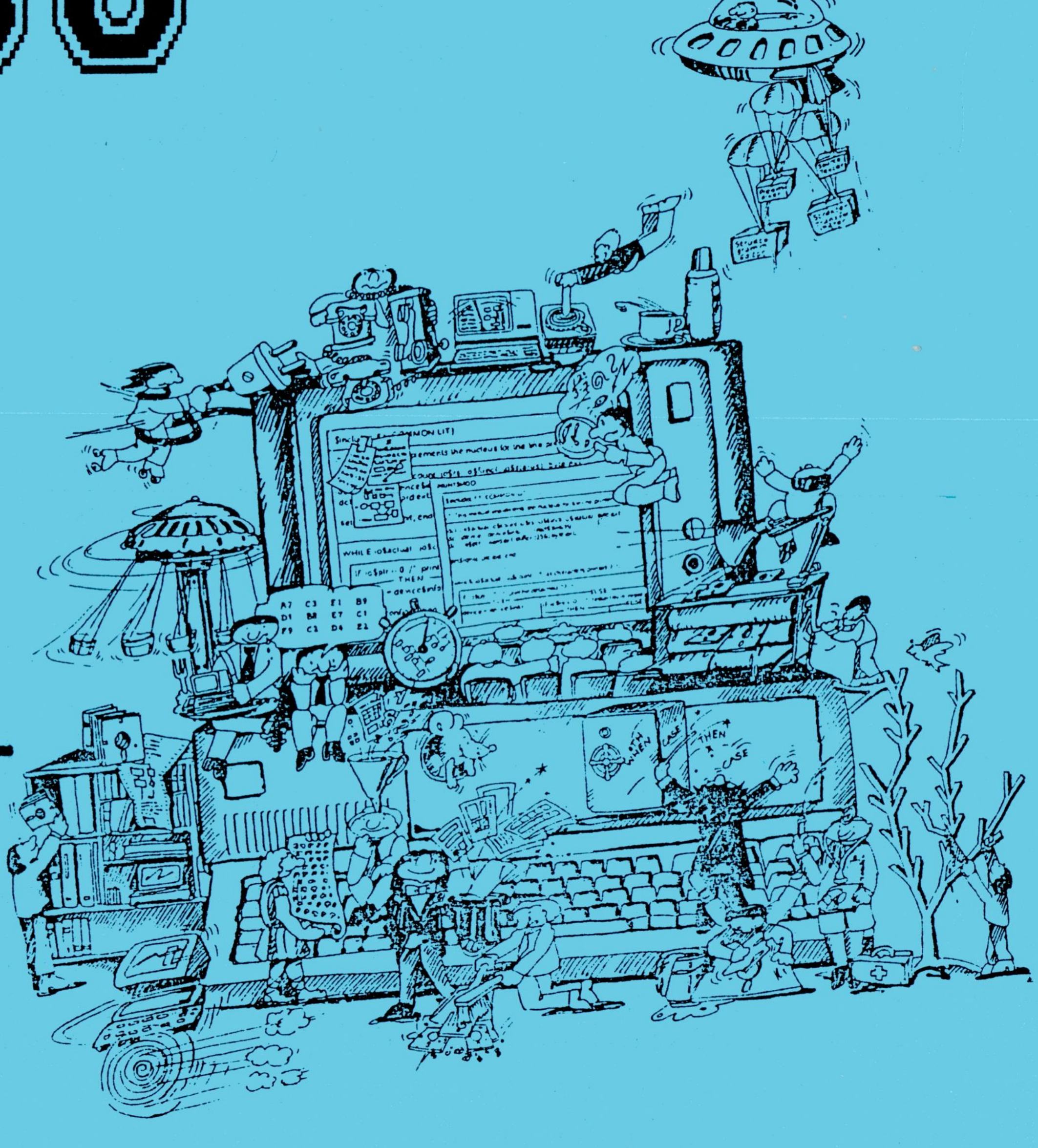
Wired-OR, Verdrahtetes ODER
(wired-OR)

Eine spezielle Form der ODER-Schaltung, die durch einfaches Verbinden der Ausgänge mehrerer logischer → Digitalschaltungen entsteht (s. Bild 16). An dem Verbindungspunkt kommt schon dann ein 1-Signal vor, wenn mindestens einer (gleichgültig welcher) der Ausgänge 1-Signal hat. Das Verdrahtete ODER ist bei fast allen → ECL-Schaltungen möglich.

Dipl.-Ing. Gerhard Hanke, WissMA
Forschungsinstitut der DBP beim FTZ Darmstadt
(B 1311.03.87 G)

CLUB 80

Clubinfo
der
TANDY -
GENIE -
und KOMTEK -
ANWENDER



23. AUSGABE

KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRAßE 6 / 6209 HEIDENROD 1
POSTFACH 27 / TEL.: 06124 / 3913

- I N H A L T S V E R Z E I C H N I S -

Seite: und Autor:

Clubinternes

Neues vom Vorstand .... 1 - 4
Hartmut Obermann

Geburtstagskinder 5
Jutta Obermann

Vorstellungen 6
Michel, Werner

Club-Emblem .... 7 - 8
Matthias, Gerald, KaJot

In eigener Sache 7
Redaktion

Termine / Messen 9
Redaktion

Software

Kleines Textprogramm .... 10 - 16
TRS 80: Sortieren 12
TRS 80 erstellt Hardcopy 17
Zeichengenerator für Nadeldrucker .. 18
Artikel aus MC

Wer sucht soll finden .... 19 - 21
Was TSCRIPS nicht kann .... 20 - 23
Korrigieren und Steuern innerhalb VC 24 - 26
Klaus-Jürgen Mühlenbein

Kleine Hexen bevorzugt .... 27 - 29
Arnulf Sopp

Datenübertragung / M4P als Terminal 30 - 32
Klaus Hermann

Druckausgabe für COBOL-Programme ... 32
Werner Förster

Vergleichstest .... 33 - 48
Hartmut, Kurt, Gerald

CP/M intern - kurz und Kompakt ..... 49 - 63
Gerald Schröder

dBase II .... 65 - 68
Harald Mand

Seite: und Autor:

Hardware

Maus mit sieben Beinen .... 69 - 70
Arnulf Sopp

Noch ein Druckerpatch .... 71 - 72
Manfred Held

Wieder mal: 40/80 Trackumschaltung . 72
Hartmut Obermann

Anmerkung zum Club-80-ECB-Bus-Projekt 73
Gepufferter DMA- und IM2-fähiger ECB 74 - 84
Helmut Bernhardt

Börse

Wer hat was -- wer will was .... 85 - 87

Sonstiges

Z80 Hochintegrations CPUs 88
Heinrich Betz

Testbericht KX 1083 .... 89 - 90
Harald Mand

Reingefallen mit Rheintec 90
Paul-Jürgen Schmitz

Echte Programmierer meiden PASCAL .. 91 - 97
DATAMON

Seite: und Autor:

Programmbibliothek

PD für M 4 /4P 97
Klaus Hermann

CP/M Soft .... 98 - 99
Andreas Rychlik

Club Bibliothek

Club-80 - Bücherei 100
Hartmut Obermann

Die letzten Seiten

Impressum 101
Schluß 102
Redaktion

Sonderausgaben .... am INFO-Ende
KaJot, Hartmut

Neues vom Vorstand

Das erste Mitglied

Falsch getippt, nicht das erste Neumitglied im Jahr 1988, sondern das erste Mitglied des CLUB 80 überhaupt, welches keinen TRS 80 oder auch nur einen kompatiblen sein eigen nennt! Matthias Homann heißt der Knabe, wohnhaft in Seevetal (bei Gerald Schröder gleich um die Ecke) und stolzer Besitzer eines modifizierten NDR-Klein-Computers mit 64180-CPU, GDP 644-HRG und ... und ... und ... Aber das soll ja weder eine komplette Vorstellung von Matthias, noch eine Werbeaktion für NDR-Klein-Computer werden. Vielmehr wollte ich euer Augenmerk auf den Umstand richten, daß man nicht unbedingt Besitzer eines TRS 80 oder kompatiblen sein muß, um Mitglied im CLUB 80 zu werden. Ich hoffe für Matthias (und auch für eventuell noch folgende Nicht-TRS 80-Besitzer) daß unser Info und die sonstigen Leistungen des CLUB 80 (Bücherei, Programmsammlung usw.) auch für ihn so interessant sind, daß er seinen Entschluß, bei uns einzusteigen nicht bereut und noch lange Zeit zu uns gehört!

Stringy-Floppy

Ebenfalls ein Unikum in unseren Reihen stellt Werner Hentz dar. Er ist zwar Besitzer eines TRS 80 Model 1, arbeitet aber, meinen Informationen nach als einziger im CLUB 80, nur mit Stringy-Floppys. Wer noch solche Dinger irgendwo rumliegen hat oder sogar damit arbeitet, kann sich ja mal mit Werner in Verbindung setzen!

MS-DOS

Die vor einiger Zeit recht heftige Diskussion über eine MS-DOS-Ecke im Info ist zwar etwas abgeflaut, zu einem Ende ist sie aber immer noch nicht gekommen. Nachdem ich jetzt selbst auch einen PC-kompatiblen mein Eigen nenne, würde ich gerne wissen, welche Mitglieder des Clubs ebenfalls zweigleisig fahren. Wer also neben dem TRS 80 auch noch einen anderen Computer (ob PC, Amiga, Atari oder sonst etwas) beackert, möge sich bei mir melden. Ein Anruf oder eine Postkarte genügen, wer mich gleich mit einem längeren Brief beehren möchte, kann das auch tun. Eine Antwort bekommt ihr auf alle Fälle!

Clubtreffen 88

Nach mehreren erfolglosen Versuchen habe ich (stimmt nicht, Jutta hat) endlich ein geeignetes Hotel für unser jährliches Club-Treffen gefunden. Es handelt sich um das Hotel Idsteiner Hof in Idstein. Ursprünglich wollten wir ja in den Raum Darmstadt/Bergstraße, hatten aber bei unserer Suche dort keinen Erfolg. Auf Idstein fiel die Wahl, weil es direkt an einer Nord-Süd-Autobahnverbindung (A 3/E 5), etwa 30 km nördlich von Wiesbaden und ca. 50 km nordwestlich von Frankfurt liegt. Damit ist wieder gewährleistet, daß die Anfahrstrecke weder für die Nordlichter (ich halte mich zu dieser Zeit übrigens in Rendsburg/Schleswig auf) noch für die Südländer zu lang ist.

Idstein ist ein sehr schönes kleines Städtchen, dessen Altstadtkern gerade neu renoviert wurde. Auch die nähere Umgebung (Wiesbaden, Limburg, Diez, Feldberg usw.) bieten einiges, so daß es auch für eventuell mit anreisende "bessere Hälften" nicht langweilig werden dürfte. Das Hotel liegt direkt an der Straße, die von der Autobahn in die Stadt führt und ist damit leicht zu finden. Die Zimmer sind zwar teilweise recht spartanisch, jedoch sehr ordentlich und sauber eingerichtet. Die Preise für Doppelzimmer liegen zwischen 20,-- und 30,-- DM pro Person incl. Frühstück. Die Tagungsräume, die sehr gemütlich ausgestattet und recht groß sind, werden uns wieder kostenfrei zur Verfügung gestellt.

Wo viel Licht ist, findet sich leider immer wieder auch der eine oder andere Schatten. Beim Idsteiner Hof ist es die leider recht beschränkte Unterbringungskapazität von 18 Betten, so daß eventuell einige Mitglieder anderseitig untergebracht werden müssen. Aber auch das ist kein Beinbruch, da die sehr hilfsbereite und freundliche Wirtin versprochen hat, bei der Suche nach Quartieren in Hotels in der Nähe tatkräftige Unterstützung zu leisten.

Da ich nicht weiß, wie lange es bei diesem Info vom Redaktionstermin bis zur Auslieferung dauert, werde ich die Einladung gesondert verschicken. Eigentlich müßte sie euch schon erreicht haben!? Wenn dies nicht der Fall ist (grande Malheur) und ihr am 12. bis 15. Mai am Clubtreffen teilnehmen wollt, ruft mich schleunigst an, damit ich euch noch eine Einladung zukommen lassen kann!

Eine Tagesordnung für das Clubtreffen besteht, wie jedes Jahr, noch nicht. Trotzdem bin ich sicher, daß auch dieses Treffen wieder ein voller Erfolg wird. Der Rahmen wird ungefähr wie folgt aussehen:

- Do. 12.05 - Anreise der Computersüchtigen, die das seltene Ereignis voll auskosten wollen. Zeit zum Meinungs-, Erfahrungs- und Programmaustauch
- Fr. 13.05 - Anreise für die nicht Abergläubischen, Gelegenheitshacker und diejenigen, die es nicht geschafft haben, für diesen Tag Urlaub (beim Chef oder der Frau) zu bekommen. Nach dem Mittagessen eventuell erste Vorträge (wenn sich jemand findet, der etwas wissenswertes und interessantes vorzutragen hat) und noch mehr Zeit zum Hacken
- Sa. 14.05 - Vor- und Nachmittags weitere Vorträge und viel Zeit zum Meinungs- und Erfahrungsaustausch. Nach dem Fünfuhrtee soll es dann etwas offizieller werden, sprich die Jahreshauptversammlung steigt.
- So. 15.05 - Da auch das schönste Clubtreffen mal ein Ende haben muß, wird dies wohl der Tag des Abschiednehmens werden. Die einen werden früher, die anderen später das Weite suchen und wie in jedem Jahr bleibt es jedem selbst überlassen, wann er den Tagungort verläßt.

Damit dieser Rahmen sich mit Leben füllt, sind natürlich alle Teilnehmer aufgerufen, sich möglichst aktiv am Treffen zu beteiligen. Gefragt sind vor allem Vorträge über die verschiedensten Aspekte der Hard- und Software (z.B. wird sich Rüdiger Sörensen über Turbo-Pascal auslassen). Aber auch der Gedankenaustausch über die Zukunft des CLUB 80 sollte nicht zu kurz kommen. Themen wie "MS-DOS-Ecke ja oder nein", "Verbesserung des Infos durch ansprechendere Gestaltung" oder "öffnung des CLUB 80 für nicht TRS 80-kompatible Computer" stehen bei der Jahreshauptversammlung mit Sicherheit auf der Tagesordnung. Auf alle Fälle gilt das Motto: je vielfälltiger das Angebot, desto Interessanter das Treffen!

Damit will ich für diesmal Schluß machen. Ich freue mich jetzt schon, euch möglichst vollzählig in Idstein begrüßen zu dürfen und wünsche euch in diesem Sinne alles Gute und viel Spaß am Computerhobby, euer

Hartmut Obermann

Hotel - Restaurant - Bierstube
IDSTEINER HOF
Inh. Maria Keil
Wiesbadener Straße 43
6270 IDSTEIN/TAUNUS
Telefon 06126/3227

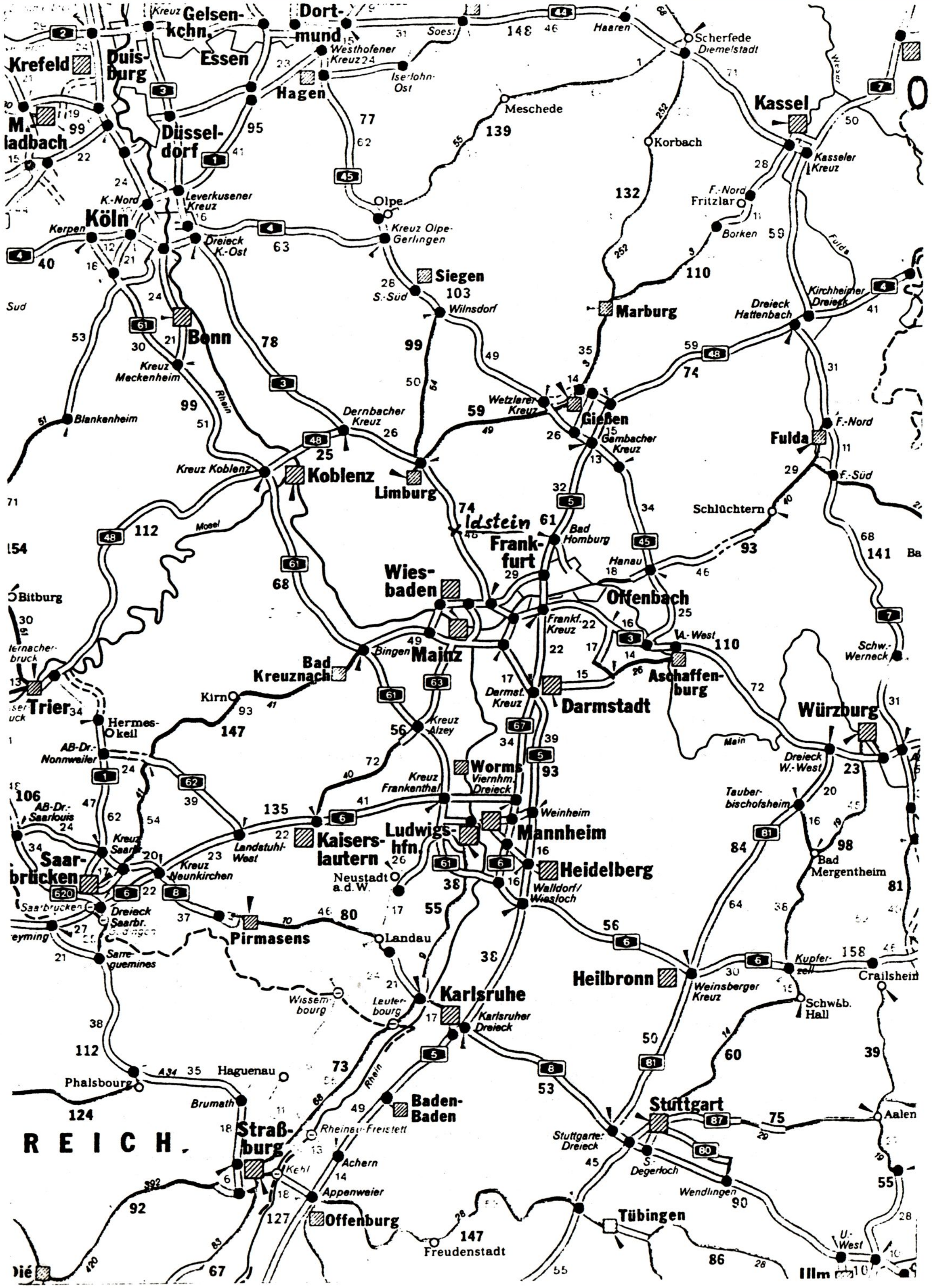
Wir empfehlen uns für Tagungen, Geschäfts- und Familienfeiern

Der Idsteiner Hof ist sehr leicht zu finden:

- Autobahn A3 Abfahrt Idstein abfahren
- Richtung Innenstadt fahren
- direkt an der Straße, die in die Stadt führt, kommt links eine Texaco-Tankstelle
- direkt nach der Tankstelle liegt der Idsteiner Hof
- Parkplätze gibt es auf der Rückseite des Gebäudes!

Gute Fahrt und sicheres Ankommen wünscht

Hartmut Obermann



G e b u r t s t a g s k i n d e r

Hallo Freunde,

heute ist es wieder soweit. Es weilen wieder viele Geburtstags"kinder" unter uns.
Im letzten Info kamen zwar einige Geburtstagswünsche reichlich spät, aber ihr könnt versichert sein, sie waren genauso herzlich, wie es diese sind.

Da sich bis jetzt noch kein im letzten Infobeitrag vergessenes Mitglied bei mir gemeldet hat, nehme ich an, daß sich niemand übergangen fühlte. Trotzdem möchte ich all diejenigen, von denen ich kein Geburtstagsdatum habe, bitten, mir dieses mitzuteilen.

So, nun endlich geht es los.

Im Monat **Februar** hatten Geburtstag:
am 2. : Spieß Peter
am 4. : Emmrich Helmut (der unseren Club leider vor kurzem verlassen hat)
am 9. : Schmitz Paul-Jürgen und Rensch Richard
am 17. : Schröder Gerald
am 26. : Wucherer Jürgen

Im Monat **März** gibt es nur einen (reichlich schwacher Monat):
am 30. : Sörensen Rüdiger

Im Monat **April** sind es wieder einige mehr:
am 7. : Trapp Harald und Schmid Alexander
am 26. : Loose Gerhard
am 28. : Piller Walter
am 30. : Böckling Ulrich

Weiterhin gratulieren wir zwei frischgebackenen Vätern und den dazugehörigen Müttern und wünschen beiden wenig durchwachte Nächte (wegen des Babys). Klaus Hermann (Monat Dezember '87) und Manfred Held (Januar '88) haben trotz der vielen Computerei noch Zeit für andere Dinge gefunden. Sie wurden beide Väter einer Tochter. Juhuu!!!

So, das war's mal wieder. Wie ihr seht, gab und gibt es wieder viel zu feiern.

Im Namen des Vorstands wünsche ich den Geburtstags"kindern" und allen anderen, die etwas zu feiern haben, viel Glück und hoffe, daß wir uns zum Clubtreffen in Idstein wiedersehen (12.05.-15.05.).

Eure Clubmaus

Jutta

Hallo Clubfreunde,

als neues Mitglied im Club möchte ich mich kurz vorstellen.

Ich heiße Werner Hentz, geb. 3.11.42, wohnhaft in Maintal - einer Stadt zwischen Frankfurt und Hanau. Beruflich bin ich in Sachen EDV schon seit gut 20 Jahren tätig und hoffe von Euch noch einiges dazulernen zu können. Es war reiner Zufall, daß über eine CHIP-Anzeige mich ein Frankfurter Arzt auf den CLUB 80 aufmerksam gemacht hatte.

Meine TRS-80/Modell I Anlage besitze ich seit Sommer 1979, die nach und nach in Stufen erweitert bzw. ausgebaut wurde. Als absolute Besonderheit werden meine Programme über STRINGY FLOPPY - einer Alternative zwischen Kassette und Diskette - geladen und gespeichert. Das Modell I arbeitet mit 48 kRAM, erhöhter Taktfrequenz und Kleinschreibung (incl. Zeichengenerator mit Unterlängen). Seit Herbst letzten Jahres kam der Thermo-Transferdrucker TPX-80 von C. ITOH dazu, der über das TANDY-Bufferkabel 26-1411 am Bus angeschlossen ist.

Von meinem Clubbeitritt im November letzten Jahres erhoffe ich mir in erster Linie gute Kontakte zu den übriggebliebenen Z80-Freaks.

Auf gute Zusammenarbeit
Werner Hentz

Mit freundlichen Grüßen Werner Hentz

Liebe Clubfreunde,

da ich nicht sehr stark in deutscher Sprache bin, möchte ich mich hier nur kurz vorstellen.

Eigentlich bin ich ein Franzose aus Bordeaux (Gironde) und 1935 geboren.

Vor ungefähr 6 Jahren habe ich mit einem Model 1 mit Kassetten begonnen. Dann habe ich mich an den TRS 80 Club von Yverdon angeschlossen. Um mich weiterzubilden, habe ich viele Bücher gekauft, um Software in Basic besser zu verstehen. Dafür habe ich in Hardware keine Erfahrung.

Um CP/M kennenzulernen, habe ich mir einen Commodore 128 und ein Genie III gekauft, sodaß ich die zahlreichen Unterlagen über Z80-Assembler und Pascal anwenden kann.

Soweit meine Vorstellung.

Auf Gute Zusammenarbeit mit euch freue ich mich

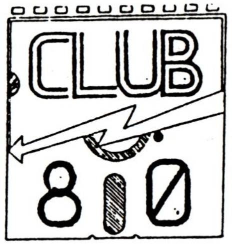
Michel

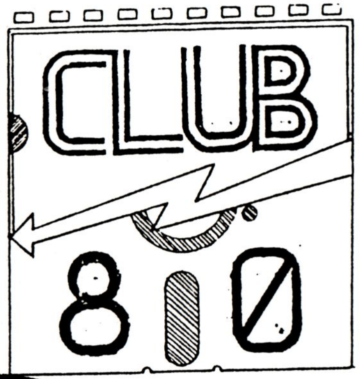
Thema „Club-Emblem“:
Zum Emblem selbst ist nicht viel zu sagen. Es soll ganz einfach 3 Punkte darstellen
1. den Clubnamen
2. Hardwarebezug
3. Softwarebezug

Das ganze sollte auch noch einigermaßen nett aussehen und auch noch ganz klein erkennbar bleiben. Was herauskam siehst Du hier.

Bis dann

Matthias





In eigener Sache:

Wir haben keine Druckerei für das Clubinfo mehr!!

Peter Spieß hört aus beruflichen Gründen auf.
Da er sich nun nur noch mit IBM-PC's beschäftigt,
ist er gleichzeitig auch aus unserem Club ausgetreten.

Aus diesem Grunde suchen wir einen neuen Vervielfältiger
für unsere kommenden Club-Info's. Vielleicht hat einer von
Euch
eine gute Adresse bei der der Club 80 weiterhin
kostengünstig
das Club-Info herstellen lassen kann?

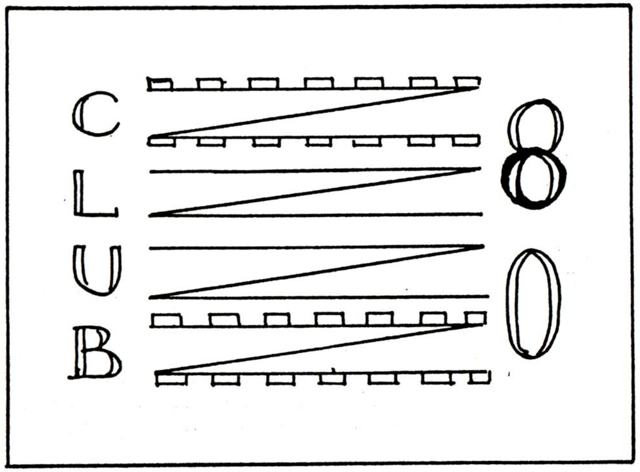
Meldungen dazu bitte an die Club-80-Redaktion.

Betr.: Club 80 - Emblem [umseitig]

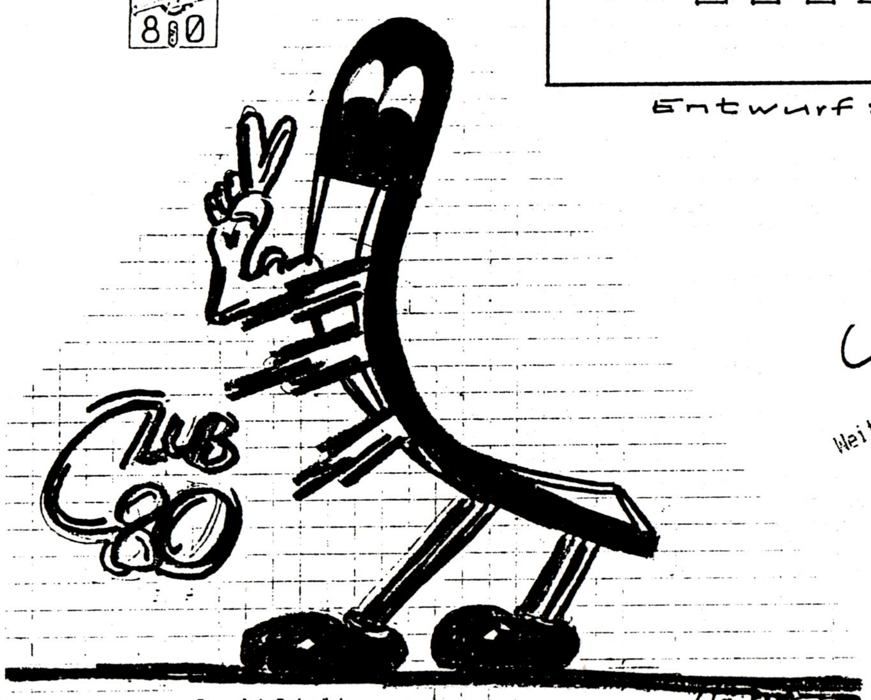
1) Die beiden Außenleisten links + rechts sind klar.

2) Der Mittelteil symbolisiert:
   a) das Z von Z 80
   b) je einen Chip oben und unten
   c) die „endlose“ Zickzack-Faltung des Druckpapierstapels

KaJot, Weinheim 18.1.88



Entwurf: KaJot



Gerald Schröder

Weitere Vorschläge zum Club-Emblem

-- Termine -- Termine -- Termine -- Termine --

Flohmarkt Nürnberg Treffen ....23. April 1988
Nächster Redaktionsschluß ....30. April 1988
Jahreshauptversammlung ....12. - 15 Mai 1988

-- Messen '88 --

CeBIT ....Hannover 16. - 23. März 1988
Büro + Computer ....München 4. - 7. Mai 1988
ORGATECHNIK ....Köln 20. - 25. Oktober 1988

Frank Sandlos

Kleines Textprogramm

Für TRS-80

Hier ein kleines Text-Verarbeitungsprogramm für das TRS-80-Modell 1. Mit wenig Aufwand kann man damit ganz schön Texte erstellen. Vielleicht auch als Beispiel geeignet, wie Besitzer anderer Computermodelle sich so ein Text-System anlegen könnten. Umlaute und Groß-Kleinschreibung sind möglich.

Das Programmlisting (*Bild 1*) enthält in den Zeilen 51000...51020 eine Basic-Routine, die ein Maschinenprogramm in den Speicher ab 32683 (dez.) schreibt (Speicher: 16 KByte), das Großbuchstaben mit einem Grafik-Block zur Unterscheidung von den Kleinbuchstaben versieht. Ferner sind die Kleinbuchstaben direkt wie bei einer normalen Schreibmaschine erreichbar, Großbuchstaben nur mit „Shift". Besitzer eines TRS-80 mit 16-KByte-Speicher und eingebauter Kleinschrift (und möglicherweise Umlauten) lassen die Zeilen 5,51000...51020 weg und benutzen Zeile 104 aus dem Listing in Bild 4. Ist ein Umlautsatz ebenfalls schon eingebaut, so kann auch die Sondertastenbelegung in Zeile 132 wegfallen. Man sollte dann auch die in den Zeilen 355...357 unterstrichenen Druckercodes experimentell überprüfen, die eventuell modifiziert werden müssen. Bei Benutzung der Kleinschreibroutine muß „MEM SIZE?" mit 32683 beantwortet werden. Selbstverständlich kann dieses Programm auch mit einem anderen Druckermodell als dem MX-80 von Epson benutzt werden, es müssen dann die Druckercodes in den Zeilen 132,355...357 geändert werden. Ist bei der Eingabe des Programmes das Programmzeilenende erreicht, d. h. der Computer nimmt keine Zeichen mehr für die betreffende Zeile an, obwohl im Listing noch Zeichen stehen, so gehe man in den Edit-Modus. Nur im Edit-Modus des TRS-80 läßt sich die im Handbuch angegebene Zeilenlänge von max. 255 Zeichen erreichen!

Dieses Programm addiert bei der Druckerausgabe zum ASCII-Wert des betreffenden Zeichens 32 hinzu, wenn er im Bereich von 64...90 liegt. Liegt dieser Wert zwischen 96 und 122, so wird 32 subtrahiert. Deshalb müssen die Druckeranweisungen in diesem Bereich angepaßt werden.

Die Bedienung des Programms

Nach Eingabe und Start des Programms wählt man die Hauptzeilenlänge, entweder 122 oder 72 Zeichen/Zeile. Diese Zeilenlänge wird dann beim Programmablauf und bei der Druckerausgabe beibehalten. Bei 122 Zeichen/Zeile wird der Drucker auf die Schriftart „SI" programmiert, ansonsten wird die Normalgröße beibehalten. Der Text wird mit einem genügend breiten Abstand vom linken Papierrand ausgedruckt (zum Abheften). Um eine optimale Ausnutzung des DIN-A4-Blattes zu erreichen, muß die obere Kante des jeweiligen Blattes mit dem anliegenden Rand der klappbaren Skaleneinteilung des MX-80 bündig abschließen. Nach der Wahl der Hauptzeilenlänge wählt man entweder die Schriftart der nächsten einzugebenden Zeile oder man unterbricht den Eingabemodus durch Drücken der Taste „Pfeil unten". Folgende Eingaben zur Schriftauswahl sind möglich:

- ‚ENTER' – Normalschrift (N)
- ‚E' – „Emphasized"-Schrift (E)
- ‚D' – „Double Printed"-Schrift (D)
- ‚S' – „Shift Out"-Schrift (S), dabei autom. Begrenzung der Zeilenlänge auf 36 Zeichen.

Nun erscheint eine Skala, unter der ein Leuchtbalken die Länge der Textzeile markiert. Dahinter befindet sich die Zeilennummer sowie nach einem Grafik-Block eine Abkürzung der Schriftart. Jetzt kann man den Text eingeben, mit Benutzung der in *Bild 2* beschriebenen Sondertasten. Will man die Eingabe beenden und die Textzeile speichern, einfach „Enter" drücken. Drückt man „Enter" ohne Zeicheneingabe, so wird die Zeile beim Ausdrucken des Textes ignoriert, es entsteht keine Leerzeile. Leerzeilen werden erst ausgedruckt, wenn man vor „Enter" mindestens ein Leerzeichen eingibt. Drückt man, anstatt die Schriftart der Folgezeile auszuwählen, die Taste „Pfeil unten", so erscheint eine Auswahltafel auf dem Monitor. Diese erklärt sich weitgehend von selbst, einige Punkte sollen jedoch näher erläutert werden.

Sichten der Textzeilen, „Pfeil unten"

Nach Drücken dieser Taste fragt das Programm nach der Anfangszeilennummer, ab der der Text gesichtet werden soll.
„A" + „Enter" bewirkt automatischen Durchlauf des ganzen Textes.
„Enter" bewirkt Anzeige der ersten Zeile. Danach manuelles Weiterschalten durch „Pfeil unten".

„xx" + „Enter", die Zeile Nr. xx wird angezeigt, manuelles Weiterschalten durch „Pfeil unten".

Zum automatischen Sichten: Nach Beendigung dieses Vorgangs entweder „Enter" zum Verlassen dieses Programmteiles oder „A" zur erneuten Durchsicht drücken.

Zum manuellen Sichten: „Enter" bewirkt Verlassen dieses Programmteiles, „K" bewirkt anschließende Korrektur der zuletzt angezeigten Zeile.

Einfügen einer Zeile, „Pfeil rechts"

Nach Tastendruck der „Pfeil rechts"-Taste muß die Zeilennummer der neu einzufügenden Zeile eingegeben werden. Diese muß genau in der Mitte zwischen den Nummern zweier schon eingegebener Zeilen liegen, also z. B. 1.5 oder 2.5, wenn die Zeilen 1 und 2 bzw. 2 und 3 schon eingegeben sind (möglich ist auch eine Überschreibung einer schon eingegebenen Zeile). Danach entweder „N", „E", „S" oder „D" zur Schrifttypauswahl drücken (Beantwortung der Frage „Typ?"). Nach Eingabe der Zeile „Enter" drücken, wie im Eingabemodus. Bitte beachten: Die vorgesehene Zeilenzahl pro DIN-A4-Seite beträgt 67 Zeilen. Deshalb nicht mehr als zwei Zeilen zusätzlich einfügen, wenn die volle Zeilenzahl in Anspruch genommen wird.

Korrektur einer Zeile, „K"

Hier gibt man entweder die Zeilennummer an oder drückt „Enter", wodurch die erste Zeile des Textes korrigiert werden kann. Auf dem Bildschirm wird nicht nur die Korrekturzeile, sondern zur besseren Übersicht auch die vorher-

gehende und nachfolgende Zeile angezeigt. Die besonderen Tastenfunktionen bei der Korrektur einer Zeile zeigt *Bild* 3.

Aufbau des Programms

Zeilen:

10...90	Titel, Eingabe der Hauptzeilenlänge
99	Beginn der Hauptschleife
103	Eingabe der Schriftart
131...143	Eingabemodus
150...155	Auswahltafel
160...169	Sichten der Textzeilen
175	Einfügen einer Textzeile
190...230	Korrektur
300...305	Speichern des Textes auf Kassette
310...315	Lesen des Textes von Kassette
350...359	Text Ausdruck
ab 51000	Basic-Loader

TIP: Zur besseren Orientierung kann man die Umlaute mit einem Prägegerät in Spezialfolie stanzen und vor die entsprechenden Tastenkappen kleben.
Bild 1 zeigt, wie gesagt, das komplette Programm für den TRS-80, Modell 1, 16 KByte, ohne Modifikationen. In *Bild 4* ist das Programm in einer Version für den TRS-80 mit 48-KByte-Speicher sowie Kleinschriftmodifikation und eventuell deutschem Zeichensatz zu sehen (verwendet wird der Kleinschrifttreiber aus mc 12/1982). In dieser Version sind einige Bedienungstasten geändert worden, näheres klärt sich beim Programmlauf von selbst. Besonders viel Ärger bereitete der Aufbau von Zeichenkettendateien in der Form PRINT#-1, A$,B$,... Die so aufgebaute Datei wird vom TRS-80 nur unvollständig eingelesen. Alle Versuche, diesen Fehler durch geschickte Programmierung zu umge-

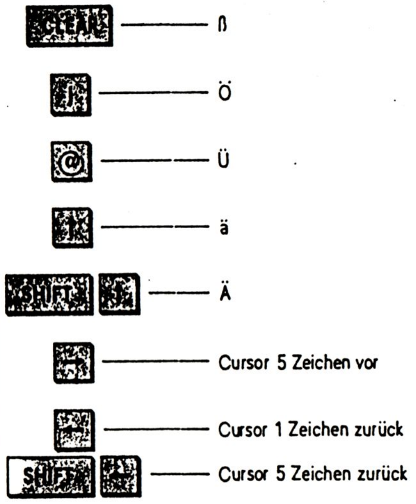


Bild 2. Die Sondertasten des Eingabemodus

TRS-80: Sortieren

In mc 1982, Heft 2, wurde eine Verbesserung der Stringsortierung für Verfahren wie z. B. Quicksort vorgestellt. Die Methode der Sortierung der Ordnungsnummer anstelle der eigentlichen Strings verbraucht aber zusätzlichen Speicherplatz, der dann an Aufnahmekapazität verlorengeht.
Jedem String eine Ordnungsnummer zuzuweisen (ein Integer-Array belegt 2n+m Speicherplätze, wobei n die Anzahl der Strings bzw. Ordnungsnummern ist und m die Anzahl der Bytes der internen Arraybeschreibung), verbraucht also bei großen Text-Arrays zusätzlich 2×n Bytes. Nun ist es aber beim TRS-80 oder beim Video-Genie so, daß sich Vertauschung von String-Inhalten ohne eigentliche Basic-Zuweisung ausführen läßt.
Wie schon in mc geschildert, wird bei einer Zuweisung an eine String-Variable freier Speicherplatz beschrieben und der alte Platz im Textspeicher einfach vergessen. Das kann dann bis zur völligen Belegung des Textspeichers und dem damit verbundenen Aufruf der Garbage-Kollektion führen.
Ist eine Textvariable eingerichtet, das ist bei Komponenten eines Arrays immer der Fall, so besteht diese Variable zunächst nur aus einem Deskriptor, der die Länge des Strings und die Startadresse enthält. Wird nun der Variablen ein (Text-)Wert zugewiesen, so wird dieser Wert im freien Textspeicher abgelegt und der Anfangszeiger des Deskriptors auf die entsprechende Stelle gesetzt. Des weiteren wird der Längenzähler auf den neuen Wert (Länge des zugewiesenen Wertes) gesetzt. Bei einer erneuten Zuweisung wird der alte Deskriptor auf einen neueingerichteten String im Textspeicher gesetzt usw. Die Belegung freien Speicherplatzes läßt sich vermeiden, wenn man nicht einfach eine Zuweisung wie etwa

```
C$=A$(1):A$(1)=A$(2):A$(2)=C$
```

vornimmt, sondern gleich die entsprechenden Deskriptoren verändert. Folgende Fragen tauchen dabei auf:

1. Wo findet man den Deskriptor?

2. An welcher Stelle steht der Längenzähler oder der Zeiger?

Beim TRS-80 liefert die Funktion VARPTR (‹Variable›), „Variablen-Pointer", die Stelle, an der im Speicher der Wert einer Variablen zu finden ist. Bei Textvariablen wird die Adresse des Deskriptors geliefert. Sei A$ eine Textvariable, so liefert

```
P=VARPTR(A$)
```

die Adresse des Deskriptors vom momentanen Stand von A$. Die Speicherstelle (P) enthält nun die Länge des Strings, (P+1) (LSB), (P+2) (MSB) enthalten den Zeiger auf den Textspeicher. Diese drei Parameter braucht man nur gegen andere auszutauschen, um der Textvariable A$ einen neuen Wert zuzuweisen, ohne dabei neuen Textspeicherplatz zu belegen. Folgendes Programm bietet sich daher zum „Swapping" (Austausch) von Texten an:

```
100 DIM A$(200)
...
200 A$=„Text1"
210 A$=„Text2"
...
1000 P=VARPTR(A$(1):Q
     =VARPTR(A$(2))
1010 P1=PEEK(P):P2=PEEK(P+1):
     P3=PEEK(P+2)
1020 Q1=PEEK(Q):Q2=PEEK(Q+1):
     Q3=PEEK(Q+2)
1030 POKE Q,P1:POKE Q+1,P2:POKE
     Q+2,P3
1040 POKE P,Q1:POKE P+1,Q2:POKE
     P+2,Q3
...
```

Nach Durchlauf der Zeilen 1000 bis 1040 sind die Inhalte der Variablen A$(1) und A$(2) ausgetauscht.
Bei anderen Systemen erfolgt die Verwaltung von Strings ähnlich.
Auf eine weitere Möglichkeit der Ausnutzung der Deskriptoren sei an dieser Stelle noch hingewiesen.
Will man etwa direkt auf den Video-Speicher des TRS-80 zugreifen, so kann dies „peek by peek" (Byte für Byte) oder abschnittweise erfolgen. Die erstere Methode, bei der jedes Byte einzeln „gepeekt" wird und in die entsprechenden Buchstaben umgewandelt werden muß, verbraucht viel Zeit und auch wieder viel Textspeicherplatz, da jeder Buchstabe im Textspeicher (unnützerweise) abgelegt wird.
Folgendes Programmstück bildet den Inhalt des Video-RAM auf einem Text-Array Z$(16) ab, wobei nach dieser Abbildung eine Zuweisung an andere Textvariable vorzunehmen ist, da beim Schreiben auf den Bildschirm unmittelbar in die Variableninhalte geschrieben wird.
Man kann natürlich auch zu Anfang des Programms diese Variablen entsprechend präparieren, und dann deren Inhalte abfragen.

```
100 DIM Z$(16)
...
1000 P=15360:FOR N=0 TO 15
1010 Q=VARPTR(Z$(N+1)):
     P1=N*64+P
1020 Q1=INT(P1/256):Q2=P1-256*Q1
1030 POKE Q,64:POKE Q+1,Q2:POKE
     Q+2,Q1
1040 NEXT N
```

Alle Variablen sollten bereits eingerichtet sein, um eine Verschiebung ihrer Adressen beim Einrichten neuer Variablen zu vermeiden.

Nach Abarbeitung des Programmstücks 1000...1040 zeigen die Pointer der Variablen Z$(1)...Z$(16) genau auf das Video-RAM. Man braucht also nicht mehr byteweise abzufragen, sondern nur noch zeilenweise. *Heinrich Seebauer*

```
5 GOSUB51020
10 CLS:CLEAR9400:DIMA$(133),Z(40):DEFINTA-Y:PRINT$86,"T E X T S Y S T E M":PRINT
STRING$(64,42):FORZ=0TO200:NEXT:G$=STRING$(4,176):H$=CHR$(143):I$=CHR$(140):K$=C
HR$(191):M$=CHR$(131)
20 N$="----"+H$+"----1----"+H$+"----2----"+H$+"----3----"+H$+"----4----"+H$+"---
-5----"+H$+"----6---4":P$="--- SCHRIFTSTAERKE(N;D;E;S) ------ ' "+CHR$(92)+" 'SC
HREIBMODUS BEENDEN ---"
25 ST$=">>>>>>>> START <<<<<<<<":KA$="** KASSETTENRECORDER BEREIT "
30 PRINT$211,G$" "G$" "G$" "CHR$(176)" "G$" "G$CHR$(176)$275,H$I$I$CHR$(188)" "K
$"     "K$I$I$CHR$(183)" "K$" "K$I$I$H$"     "K$$339,M$M$M$M$" "M$M$M$M$" "M$"  "M$
" "M$" "M$"        "M$
60 FORZ=0TO650:NEXT:PRINT$143,STRING$(34,176)$207,K$$240,K$$271,K$$304,K$$335,K$
$368,K$$399,STRING$(34,131)
70 FORZ=0TO500:NEXT:PRINT$194,"EIN"$260,"PROGRAMM"$331,"VON"$309,"F R A N K"$371
,"S A N D L O S":FORZ=0TO400:NEXT:PRINT:PRINT"HAUPTZEILENLAENGE:":PRINT:PRINT"
   72 ZEICHEN/ZEILE ----------- N(ORMAL)"
80 PRINT"    122 ZEICHEN/ZEILE ----------- E(NG)":PRINT:PRINT:PRINT"          BIT
TE ANFANGSBUCHSTABEN DRUECKEN !"
90 W$=INKEY$:IFW$="",90ELSEIFW$="N",ZH=72ELSEIFW$="E",ZH=122ELSE90
99 FORZ=0TO132STEP2
100 O$=STR$(-(Z/2+1)):O$=STRING$(4-LEN(O$),45)+O$+"-":P=832:ZL=ZH:CLS:PRINT$768,
P$;:E$=""
103 R$=INKEY$:IFR$="",103ELSEIFASC(R$)=13,R$="N":GOTO104ELSEIFR$="D",104ELSEIFR$
="E",104ELSEIFR$="S",ZL=36ELSEIFASC(R$)=10,150ELSE103
104 CLS:ZZ=0:FORZ1=Z-1TOZ-24STEP-1:IFZ1<0,110ELSEZ2=LEN(A$(Z1)):IFZ2>0,Z3=INT(Z2
/64)+1:ZZ=ZZ+Z3+1:IFZZ>12,110ELSEPRINT$(768-ZZ*64),A$(Z1);:NEXTZ1ELSENEXTZ1
110 PRINT$(P-64),N$E$STRING$(ZL-LEN(E$),143)O$R$;
131 A$=INKEY$:IFA$="",131ELSEA=ASC(A$):IFA=91,E$=E$+CHR$(123):GOTO133ELSEIF64<AA
NDA<123,E$=E$+A$:PRINT$P,E$;:IFLEN(E$)<ZL,131ELSEE=LEN(E$):GOTO140ELSEE=LEN(E$):
IFA=8ANDE>0,E$=LEFT$(E$,E-1):GOTO133ELSEIFA=24ANDE>4,E$=LEFT$(E$,E-5):GOTO133
132 IFA=32,E$=E$+" ":GOTO133ELSEIFA=31,E$=E$+CHR$(126):GOTO133ELSEIFA=59,E$=E$+C
HR$(92):GOTO133ELSEIFA<64ANDA>32,E$=E$+A$:GOTO133ELSEIFA=27,E$=E$+CHR$(91)ELSEIF
A=64,E$=E$+CHR$(93)ELSEIFA=9ANDE+5<ZL,E$=E$+"     "ELSEIFA=13ANDV=0,143ELSE140
133 E=LEN(E$):PRINT$P,E$STRING$(ZL-E,143)O$R$;:IFE<ZL,131
140 IF(V=0ANDE>=ZL)OR(V=2ANDE=>ZL),141ELSEIFV=1ORV=2,A$(ZA)=E$+O$+R$:GOTO205ELSE
131
141 W$=INKEY$:IFW$="",141ELSEW=ASC(W$):IFW=8,E$=LEFT$(E$,E-1):GOTO133ELSEIFW=24,
E$=LEFT$(E$,E-5):GOTO133ELSEIFW=13ANDV=0,143ELSEIFW=13ANDV=2,A$(ZA)=E$+O$+R$:GOT
O205ELSE141
143 A$(Z)=E$+O$+R$:NEXTZ:GOTO150
150 PRINTCHR$(15):CLS:PRINT$64,"KORREKTUR:          '"CHR$(92)" XX'  - SICHTEN DER
 ZEILEN AB XX"$146,"'"CHR$(94)" XX'  - EINFUEGEN EINER ZEILE"$210,"'K XX'  - KOR
REKTUR ZEILE XX"$384,"TEXT AUSDRUCKEN:             'A'"$576,"AUF CASSETTE SPEICHER
N:    'C'"
151 PRINT$768,"VON CASSETTE LESEN:           'L'"$913,"'ENTER'  - SCHREIBMODUS";$98
0,"E      - PROGRAMMENDE";
155 W$=INKEY$:IFW$="",155ELSEW=ASC(W$):IFW=10,160ELSEIFW=9,175ELSEIFW=13,CLS:GOT
O100ELSEIFW$="K",190ELSEIFW$="A",350ELSEIFW$="C",300ELSEIFW$="L",310ELSEIFW$="E"
,400ELSE155
160 ZA$="0":PRINT$114,"NR.";:INPUTZA$:IFZA$="A",168ELSEZA=VAL(ZA$)*2-2:IFZA>Z,16
0ELSEIFZA=-2,ZA=0:CLSELSECLS
```

Bild 1. Das Programmlisting für die 16-KByte-Version mit Kassettenspeicherung (MEM SIZE? 32683)

```
161 PRINTA$(ZA)
165 W$=INKEY$:IFW$="",165ELSEIFW$="K",191ELSEW=ASC(W$):IFW=13,150ELSEIFW=10,ZA=Z
A+1:IFZA=Z,ZA=0:GOTO161ELSEIFA$(ZA)="",165ELSEPRINTA$(ZA):GOTO165ELSE165
168 CLS:FORZV=0TOZ:IFA$(ZV)="",NEXTZVELSEPRINTA$(ZV):NEXTZV
169 W$=INKEY$:IFW$="",169ELSEIFW$="A",168ELSEIFASC(W$)=13,150ELSE169
175 PRINT$178,"NR.";:INPUTZA:PRINT$178,CHR$(201);:PRINT$178,"TYP";:INPUTR$:U$="E
 I N F U E G E N":Q=1:K$=STR$(-ZA):IFLEN(K$)>4,K$=STR$(ZA)ELSEK$=STRING$(4-LEN(K
$),45)+K$
176 K$=K$+"-"+R$:ZA=ZA*2-2:Z(T)=ZA:T=T+1:GOTO200
190 ZA=1:PRINT$236,"NR.";:INPUTZA:ZA=ZA*2-2
191 U$="K O R R E K T U R"
200 WW=0:IFZA=0,203ELSEIFA$(ZA-1)="",ZE=ZA-2ELSEZE=ZA-1
203 IFA$(ZA+1)="",ZF=ZA+2ELSEZF=ZA+1
205 V=0:CLS:PRINT$19,U$"  NR."ZA/2+1$64,A$(ZE);$256,A$(ZA);$448,N$LEFT$(K$,WW)CH
R$(95);$768,A$(ZF);:IFQ=1,230
210 W$=INKEY$:IFW$="",210ELSEW=ASC(W$):K$=A$(ZA):K=LEN(K$):IFW=13,150ELSEIFW=32A
NDK>WW+1,WW=WW+1ELSEIFW=8ANDWW>0,WW=WW-1ELSEIFW=9ANDK>WW+5,WW=WW+5ELSEIFW=27,A$(
ZA)=LEFT$(K$,WW)+RIGHT$(K$,K-WW-1):GOTO205
211 IFW=91,220ELSEIFW=31,230ELSEIFW=25,K=K+1:GOTO220ELSEPRINT$512,LEFT$(K$,WW)CH
R$(95)STRING$(K-WW,32);:GOTO210
220 ZL=WW+1:E$=LEFT$(K$,WW):O$=RIGHT$(K$,K-WW-1):R$="":P=512:V=1:GOTO131
230 ZL=ZH:E$=LEFT$(K$,WW):O$=RIGHT$(K$,6):R$="":P=512:V=2:Q=0:IFRIGHT$(K$,1)="S"
,ZL=36:GOTO110ELSE110
300 CLS:PRINT$279,"S P E I C H E R N":PRINT$390,KA$;:INPUTS:PRINT$788,ST$:PRINT#
-1,ZH,Z,T:TN=0:FORGV=0TOZ-2STEP2:GW=GV
301 W=LEN(A$(GW)):B$="A"+A$(GW):IFW=0,305
303 FORGG=1TOW+1:X=ASC(MID$(B$,GG,1)):Y=LEN(B$):T$=STR$(GG):IF(X>32ANDX<48)OR(X>
57ANDX<65),B$=LEFT$(B$,GG-1)+" "+RIGHT$(B$,Y-GG)+STRING$(4-LEN(T$),32)+T$+STR$(X
):NEXTGGELSENEXTGG
305 B=LEN(B$)-W-1:C=W:PRINT#-1,TN,B,C,B$:NEXTGV:IFT=0ORGV=T,150ELSEFORGV=0TOT-1:
TN=Z(GV):GW=TN:GOTO301
310 CLS:PRINT$282,"L E S E N":PRINT$390,KA$;:INPUTS:PRINT$788,ST$:INPUT#-1,ZH,Z,
T:FORGV=0TOZ-2STEP2:GW=GV:INPUT#-1,TN,B,C,B$
313 IFB=0,A$(GW)="":NEXTGV:GOTO150ELSEX$=RIGHT$(B$,B):FORGG=1TOBSTEP7:Q=VAL(MID$
(X$,GG,5)):IFNOTQ=0,B$=LEFT$(B$,Q-1)+CHR$(VAL(MID$(X$,GG+4,3)))+MID$(B$,Q+1,C+1-
Q):NEXTGG
315 A$(GW)=RIGHT$(B$,C):NEXTGV:IFT=0ORGV=T,150ELSEFORGV=0TOT-1:INPUT#-1,TN,B,C,B
$:GW=TN:GOTO313
350 CLS:PRINT$472,"DRUCKER ON-LINE ";:INPUTS:PRINT$788,ST$:IFZH=122,N=10:S=5:LPR
INTCHR$(15)CHR$(24)ELSEN=5:S=5:LPRINTCHR$(24)
355 C=20:FORGX=0TOZ-1:R$=RIGHT$(A$(GX),1):A=LEN(A$(GX)):IFA=0ORA=6,NEXTGXELSEA$=
LEFT$(A$(GX),A-6):IFR$="N",LPRINTTAB(N)CHR$(27)CHR$(C)A$:C=20:NEXTGXELSEIFR$="S"
,LPRINTTAB(S)CHR$(27)CHR$(C)CHR$(14)CHR$(27)CHR$(101)A$:C=102:NEXTGXELSE357
356 GOTO358
357 IFR$="E",LPRINTTAB(N)CHR$(27)CHR$(C)CHR$(27)CHR$(101)A$:C=102:NEXTGXELSEIFR$
="D",LPRINTTAB(N)CHR$(27)CHR$(C)CHR$(27)CHR$(103)A$:C=104:NEXTGX
358 LPRINTCHR$(27)CHR$(C);:IFZH=122,LPRINTCHR$(18);
359 GOTO150
400 CLS:PRINT" S - LOESCHEN/NEUSTART":PRINT" E - ENDE"
401 W$=INKEY$:IFW$="",401ELSEIFW$="S",RUN10ELSEIFW$="E",END
51000 DATA 42,30,64,34,213,127,33,198,127,34,30,64,42,38,64,34,242,127,33,233,12
7,34,38,64,195,25,26,245,121,205,244,127,56,5,62,130,205,51,0,241,197,205,0,0,19
3,79,254,8,192,42,32,64,43,126,254,130,192,62,8,195,51,0,245,121,246,32,205,244
51010 DATA 127,241,195,0,0,254,123,63,216,254,97,216,121,238,32,79,201
51020 IFPEEK(16526)=171ANDPEEK(16527)=127,RETURNELSEA=32683:FORX=AT032767:READB:
POKEX,B:NEXT:POKE16526,171:POKE16527,127:A=USR(0)
```

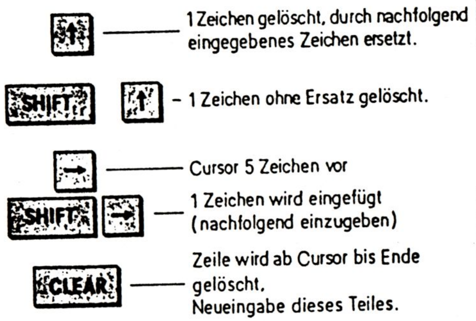


Bild 3. Die Sondertasten des Korrekturmodus


```
5 GOSUB51000
10 CLS:CLEAR15000:DIMA$(133),Z(40):DEFINTA-Y:PRINT§86,"T E X T S Y S T E M":PRIN
TSTRING$(64,45):FORZ=0TO200:NEXT:G$=STRING$(4,176):H$=CHR$(143):I$=CHR$(140):K$=
CHR$(191):M$=CHR$(131)
20 N$="-----"+H$+"-----1-----"+H$+"-----2-----"+H$+"-----3-----"+H$+"-"+CHR$(127)+"--4-
---"+H$+"-----5-----"+H$+"-----6---4":P$="--- Schriftstärke(n;d;e;s) ------- ' * 'Sc
hreibmodus beenden ---"
25 ST$=">>>>>>>> START <<<<<<<<":KA$="** Kassettenrekorder bereit "
30 PRINT§211,G$" "G$" "G$" "CHR$(176)" "G$" "G$CHR$(176)§275,H$I$I$CHR$(188)" "K
$"    "K$I$I$CHR$(183)"."K$" "K$I$I$H$"    "K$§339,M$M$M$M$" "M$M$M$M$" "M$" "M$
" "M$" "M$"        "M$
60 FORZ=0TO650:NEXT:PRINT§143,STRING$(34,176)§207,K$§240,K$§271,K$§304,K$§335,K$
§368,K$§399,STRING$(34,131)
70 FORZ=0TO500:NEXT:PRINT§194,"Ein"§260,"Programm"§331,"von"§309,"F R A N K"§371
,"S A N D L O S":FORZ=0TO400:NEXT:PRINT:PRINT"Hauptzeilenlänge:":PRINT:PRINT"
  72 Zeichen/Zeile ----------- n(ormal)"
80 PRINT"    122 Zeichen/Zeile ----------- e(ng)":PRINT:PRINT:PRINT"         Bit
te Anfangsbuchstaben drücken !"
90 W$=INKEY$:IFW$="",90ELSEIFW$="n",ZH=72ELSEIFW$="e",ZH=122ELSE90
99 FORZ=0TO132STEP2
100 O$=STR$(-(Z/2+1)):O$=STRING$(4-LEN(O$),45)+O$+"-":P=832:ZL=ZH:CLS:PRINT§768,
P$;:E$=""
103 R$=INKEY$:IFR$="",103ELSEIFASC(R$)=13,R$="n":GOTO104ELSEIFR$="d",104ELSEIFR$
="e",104ELSEIFR$="s",ZL=36ELSEIFR$="*",150ELSE103
104 CLS:ZZ=0:FORZ1=Z-1TOZ-24STEP-1:IFZ1<0,110ELSEZ2=LEN(A$(Z1)):IFZ2>0,Z3=INT(Z2
/64)+1:ZZ=ZZ+Z3:IFZZ>12,110ELSEPRINT§(768-ZZ*64),A$(Z1);:NEXTZ1ELSENEXTZ1
110 PRINT§(P-64),N$E$STRING$(ZL-LEN(E$),143)O$R$;
131 A$=INKEY$:IFA$="",131ELSEA=ASC(A$):IF64<AANDA<127,E$=E$+A$:PRINT§P,E$;:IFLEN
(E$)<ZL,131ELSEE=LEN(E$):GOTO140ELSEE=LEN(E$):IFA=8ANDE>0,E$=LEFT$(E$,E-1):GOTO1
33ELSEIFA=24ANDE>4,E$=LEFT$(E$,E-5):GOTO133
132 IFA=32,E$=E$+" ":GOTO133ELSEIFA<64ANDA>32,E$=E$+A$:GOTO133ELSEIFA=9ANDE+5<ZL
,E$=E$+"     "ELSEIFA=13ANDV=0,143ELSE140
133 E=LEN(E$):PRINT§P,E$STRING$(ZL-E,143)O$R$;:IFE<ZL,131
140 IF(V=0ANDE>=ZL)OR(V=2ANDE=>ZL),141ELSEIFV=1ORV=2,A$(ZA)=E$+O$+R$:GOTO205ELSE
131
141 W$=INKEY$:IFW$="",141ELSEW=ASC(W$):IFW=8,E$=LEFT$(E$,E-1):GOTO133ELSEIFW=24,
E$=LEFT$(E$,E-5):GOTO133ELSEIFW=13ANDV=0,143ELSEIFW=13ANDV=2,A$(ZA)=E$+O$+R$:GOT
O205ELSE141
143 A$(Z)=E$+O$+R$:NEXTZ:GOTO150
150 PRINTCHR$(15):CLS:PRINT§64,"Korrektur:          '# XX'  - Sichten der Zeilen a
b XX"§146,"'"CHR$(94)" XX'  - Einfügen einer Zeile"§210,"'k XX'  - Korrektur Zei
le XX"§384,"Text ausdrucken:             'a'"§576,"Auf Cassette speichern:   'c'"
151 PRINT§768,"Von Cassette lesen:           'l'"§913,"'ENTER'  - Schreibmodus";§98
0,"e     - Programmende";
155 W$=INKEY$:IFW$="",155ELSEW=ASC(W$):IFW=35,160ELSEIFW=123,175ELSEIFW=13,CLS:G
OTO100ELSEIFW$="k",190ELSEIFW$="a",350ELSEIFW$="c",300ELSEIFW$="l",310ELSEIFW$="
e",400ELSE155
160 ZA$="0":PRINT§114,"Nr.";:INPUTZA$:IFZA$="a",168ELSEZA=VAL(ZA$)*2-2:IFZA>Z,16
0ELSEIFZA=-2,ZA=0:CLSELSECLS
161 PRINTA$(ZA)
165 W$=INKEY$:IFW$="",165ELSEIFW$="K",191ELSEW=ASC(W$):IFW=13,150ELSEIFW=10,ZA=Z
A+1:IFZA=Z,ZA=0:GOTO161ELSEIFA$(ZA)="",165ELSEPRINTA$(ZA):GOTO165ELSE165
168 CLS:FORZV=0TOZ:IFA$(ZV)="",NEXTZVELSEPRINTA$(ZV):NEXTZV
169 W$=INKEY$:IFW$="",169ELSEIFW$="a",168ELSEIFASC(W$)=13,150ELSE169
175 PRINT§178,"Nr.";:INPUTZA:PRINT§178,CHR$(201);:PRINT§178,"Typ";:INPUTR$:U$="E
 I N F U E G E N":Q=1:K$=STR$(-ZA):IFLEN(K$)>4,K$=STR$(ZA)ELSEK$=STRING$(4-LEN(K
$),45)+K$
176 K$=K$+"-"+R$:ZA=ZA*2-2:Z(T)=ZA:T=T+1:GOTO200
190 ZA=1:PRINT§236,"Nr.";:INPUTZA:ZA=ZA*2-2
191 U$="K O R R E K T U R"
200 WW=0:IFZA=0,203ELSEIFA$(ZA-1)="",ZE=ZA-2ELSEZE=ZA-1
203 IFA$(ZA+1)="",ZF=ZA+2ELSEZF=ZA+1
205 V=0:CLS:PRINT§19,U$"  Nr."ZA/2+1§64,A$(ZE);§256,A$(ZA);§448,N$LEFT$(K$,WW)CH
R$(95);§768,A$(ZF);:IFQ=1,230
210 W$=INKEY$:IFW$="",210ELSEW=ASC(W$):K$=A$(ZA):K=LEN(K$):IFW=13,150ELSEIFW=32A
NDK>WW+1,WW=WW+1ELSEIFW=8ANDWW>0,WW=WW-1ELSEIFW=9ANDK>WW+5,WW=WW+5ELSEIFW=27,A$(
ZA)=LEFT$(K$,WW)+RIGHT$(K$,K-WW-1):GOTO205
212 IFW=123,220ELSEIFW=31,230ELSEIFW=25,K=K+1:GOTO220ELSEPRINT§512,LEFT$(K$,WW)C
HR$(95)STRING$(K-WW,32);:GOTO210
220 ZL=WW+1:E$=LEFT$(K$,WW):O$=RIGHT$(K$,K-WW-1):R$="":P=512:V=1:GOTO131
230 ZL=ZH:E$=LEFT$(K$,WW):O$=RIGHT$(K$,6):R$="":P=512:V=2:Q=0:IFRIGHT$(K$,1)="s"
,ZL=36:GOTO110ELSE110
300 CLS:PRINT§279,"S P E I C H E R N":PRINT§390,KA$;:INPUTS:PRINT§788,ST$:PRINT#
-1,ZH,Z,T:TN=0:FORGV=0TOZ-2STEP2:GW=GV
301 W=LEN(A$(GW)):B$="A"+A$(GW):IFW=0,305
303 FORGG=1TOW+1:X=ASC(MID$(B$,GG,1)):Y=LEN(B$):T$=STR$(GG):IF(X>32ANDX<48)OR(X>
57ANDX<65),B$=LEFT$(B$,GG-1)+" "+RIGHT$(B$,Y-GG)+STRING$(4-LEN(T$),32)+T$+STR$(X
):NEXTGGELSENEXTGG
305 B=LEN(B$)-W-1:C=W:PRINT#-1,TN,B,C,B$:NEXTGV:IFT=0ORGV=T,150ELSEFORGV=0TOT-1:
TN=Z(GV):GW=TN:GOTO301
310 CLS:PRINT§282,"L E S E N":PRINT§390,KA$;:INPUTS:PRINT§788,ST$:INPUT#-1,ZH,Z,
T:FORGV=0TOZ-2STEP2:GW=GV:INPUT#-1,TN,B,C,B$
313 IFB=0,A$(GW)="":NEXTGV:GOTO150ELSEX$=RIGHT$(B$,B):FORGG=1TOBSTEP7:Q=VAL(MID$
(X$,GG,5)):IFNOTQ=0,B$=LEFT$(B$,Q-1)+CHR$(VAL(MID$(X$,GG+4,3)))+MID$(B$,Q+1,C+1-
Q):NEXTGG
315 A$(GW)=RIGHT$(B$,C):NEXTGV:IFT=0ORGV=T,150ELSEFORGV=0TOT-1:INPUT#-1,TN,B,C,B
$:GW=TN:GOTO313
350 CLS:PRINT§472,"Drucker ON-LINE ";:INPUTS:PRINT§788,ST$:IFZH=122,N=10:S=5:LPR
INTCHR$(15)CHR$(24)ELSEN=5:S=5:LPRINTCHR$(24)
355 C=20:FORGX=0TOZ-1:R$=RIGHT$(A$(GX),1):A=LEN(A$(GX)):IFA=0ORA=6,NEXTGXELSEA$=
LEFT$(A$(GX),A-6):IFR$="n",LPRINTTAB(N)CHR$(27)CHR$(C)A$:C=20:NEXTGXELSEIFR$="s"
,LPRINTTAB(S)CHR$(27)CHR$(C)CHR$(14)CHR$(27)CHR$(69)A$:C=70:NEXTGXELSE357
356 GOTO358
357 IFR$="e",LPRINTTAB(N)CHR$(27)CHR$(C)CHR$(27)CHR$(69)A$:C=70:NEXTGXELSEIFR$="
d",LPRINTTAB(N)CHR$(27)CHR$(C)CHR$(27)CHR$(71)A$:C=72:NEXTGX
358 LPRINTCHR$(27)CHR$(C);:IFZH=122,LPRINTCHR$(18);
359 GOTO150
400 CLS:PRINT" s - Löschen/Neustart":PRINT" e - Ende"
401 W$=INKEY$:IFW$="",401ELSEIFW$="s",RUN10ELSEIFW$="e",END
51000 DATA33,226,255,34,30,64,33,132,255,34,22,64,205,227,3,197,213,87,254,32,32
,17,58,128,56,183,40,11,58,209,255,238,32,50,209,255,175,24,47,122,254,31,40,248
,58,64,56,254,2,32,16,122,33,210,255,1,8,0,237,177,32,5,1,7,0,9,86,122,230,223
51002 DATA254,65,56,11,254,94,48,7,122,33,209,255,174,24,1,122,209,193,201,0,65,
79,85,97,111,117,83,67,91,92,93,123,124,125,126,31,221,110,3,221,102,4,218,154,4
,221,126,5,183,40,1,119,121,254,32,218,6,5,254,128,210,166,4,195,125,4
51004 IFPEEK(16526)=119ANDPEEK(16527)=255RETURNELSEFORX=-136TO-1:READB:POKEX,B:N
EXTX:POKE16526,119:POKE16527,255:A=USR(0)
```

Bild 4. Hier die 48-KByte-Version für TRS-80 mit Kleinschrift und deutschem Zeichensatz (MEM SIZE? 65399)

TRS-80 erstellt Hardcopy

Eine Hardcopy vom Bildschirm erstellt die kleine Maschinenroutine von *Bild 1*.

```
1000 FOR J=15360 TO 16320 STEP 64:
     FOR I=J TO J+63: IF PEEK(I)>122
     THEN LPRINT CHR$(42);:IF PEEK(I)
     =<122 THEN LPRINT CHR$(PEEK(I));
1010 NEXT I:LPRINT" ":NEXT J
```

Bild 1. Eine Bildschirm-Hardcopy erstellt dieses TRS-80-Programm

Dazu ist der im Level-II und Level-III-Basic nicht benutzte Befehl „NAME" im Direktmodus oder von einem Programm aus aufzurufen. Dadurch bleibt der Befehl USR(0) für andere Zwecke frei. Je nach Speicherumfang muß die Startadresse geändert werden. Zu ändern ist eventuell auch die Druckertreiberadresse (037E8H).

Eine Hardcopy in Basic ermöglicht das Programm von *Bild 2*. Dabei kann in begrenztem Umfang auf nicht grafikfähigen Druckern Grafik ausgegeben werden. Alle Zeichen, deren Code größer als 122 ist, werden nämlich als Sternchen - CHR (42) - gedruckt. Natürlich kann man an seiner Stelle jedes andere Zeichen verwenden.

Wolfgang Bethmann

```
                    0000          ORG   0FFC0H
                    0001 ;HARDCOPY FUER TRS-80/VG/KOMTEK
                    0002 ;W.BETHMANN 11.1985
FFC0 21 C0 FF       0003          LD    HL,0FFC0H   ;Startadresse
FFC3 22 8E 40       0004          LD    (408EH),HL  ;Systemstartadresse
FFC6 22 B1 40       0005          LD    (40B1H),HL  ;MEM-SIZE setzen
FFC9 22 8F 41       0006          LD    (418FH),HL  ;Zeiger"Name"verbiegen
FFCC CD 94 03       0007          CALL  394H        ;UP: CR auf Drucker
FFCF 16 00          0008          LD    D,0         ;D auf 0 setzen
FFD1 DD 21 00 3C    0009          LD    IX,3C00H    ;1.Bildschirmplatz
FFD5 06 3F          0010 PRINT    LD    B,3FH       ;noch 63 Zeichen/Zeile
FFD7 DD 4E 00       0011 LOOP     LD    C,(IX)      ;Bildschirm nach C laden
FFDA DD E5          0012          PUSH  IX          ;Speicherinhalte retten
FFDC C5             0013          PUSH  BC          ;
FFDD CD 5F 30       0014          CALL  305FH       ;Druckertreiberadr.holen
FFE0 C1             0015          POP   BC          ;zurück vom Stack
FFE1 DD E1          0016          POP   IX          ;
FFE3 DD 23          0017          INC   IX          ;nächster Bildschirmplatz
FFE5 10 F0          0018          DJNZ  LOOP        ;Schleife noch 63 mal
                    0019 ;1.SCHLEIFE
FFE7 DD E5          0020          PUSH  IX          ;Speicherinhalte retten
FFE9 C5             0021          PUSH  BC          ;
FFEA D5             0022          PUSH  DE          ;
FFEB CD 94 03       0023          CALL  394H        ;UP: CR auf Drucker
FFEE D1             0024          POP   DE          ;zurück vom Stack
FFEF C1             0025          POP   BC          ;
FFF0 DD E1          0026          POP   IX          ;
FFF2 DD 23          0027          INC   IX          ;nächster Bildschirmplatz
FFF4 14             0028          INC   D           ;Zähler für CR
FFF5 3E 0F          0029          LD    A,0FH       ;noch 15 Zeilen
FFF7 BA             0030          CP    D           ;fertig ?
FFF8 20 DB          0031          JR    NZ,PRINT    ;sonst Schleife 15 mal
FFFA C3 CC 06       0032          JP    6CCH        ;zurück ins Basic
                    0033          END
```

Bild 2. Sogar begrenzte Grafikmöglichkeiten bietet dieses Basic-Programm

Zeichengenerator für Nadeldrucker

Selbst die einfachsten Nadeldrucker können Punktgrafiken drucken. Damit setzt nur die eigene Phantasie dem Zeichenvorrat des Druckers Grenzen. Braucht man ein mathematisches Sonderzeichen oder ein sonstiges Symbol, beginnt die Gestaltungsarbeit mit Papier und Bleistift. Vor jedem Versuch müssen die Punkte dann noch in ein Bitmuster umgesetzt werden. Diese Arbeit kann das abgedruckte Basic-Programm (*Bild 1*) erleichtern. Mit den Tasten des Zahlenfeldes steuert man die Punkte an, die gesetzt oder gelöscht werden sollen. Leider sind die Grafikpunkte des TRS-80 nicht quadratisch – die auf dem Bildschirm gemalte Figur wirkt also leicht verzerrt. Ein Tastendruck genügt, um das Resultat aufs Papier zu bringen. Im übrigen erklärt sich das Programm selber. Es wurde auf einem TRS-80, Model I geschrieben, dürfte aber nach Anpassung der Grafikbefehle (set (x, y)/reset (x,y)/point (x, y): Koordinate(x, y) setzen/löschen/prüfen) auch auf den meisten anderen Rechnern laufen. Bei einigen Druckern (Star DP-510 u. ä.) können nur sieben Bit direkt angesteuert werden. Der oberste Punkt muß dann über eine zusätzliche Abfrage (IF P(I)>127 THEN ...) angesteuert werden.

Von Drucker zu Drucker verschieden sind auch die Befehle zum Aufruf der Grafik. Sie sind in der DATA-Zeile abgelegt.

Als Beispiele für nützliche Sonderzeichen mögen Summenzeichen, Alpha, das Integralzeichen und das Zeichen für Unendlich genügen (*Bild 2*).

Henning Schulz-Rinne

```
90 CLEAR 100:CLS:DEFINT X,Y,I,J,M,P,C:DEFSTR E
100 MX=10:DIM P(MX),C(3):REM MX = Zahl der Spalten
110 FOR I=0 TO 3:READ C(I):NEXT
120 PRINT "Bedienung Über numerisches Tastenfeld:"
130 PRINT "+-----------+-----------+-----------+"
140 PRINT "I    (7)    I    auf-   I    (9)    I"
150 PRINT "I           I   wärts   I           I"
160 PRINT "+-----------+-----------+-----------+"
170 PRINT "I  links    I  Punkt    I  rechts   I"
180 PRINT "I           I  löschen  I           I"
190 PRINT "+-----------+-----------+-----------+"
200 PRINT "I  Spalte   I    ab-    I    (3)    I"
210 PRINT "I  löschen  I   wärts   I           I"
220 PRINT "+-----------+-----------+-----------+"
230 PRINT "I  aus-     I  Punkt    I  (ENTER)  I"
240 PRINT "I  drucken  I  setzen   I           I"
250 PRINT "+-----------+-----------+-----------+"
260 PRINT "Alles löschen: '/'-Taste: Ende: '-'""
270 PRINT "Zum Anfangen Taste drücken";
280 E=INKEY$:IF E="" THEN 280
290 CLS
300 FOR X=0 TO MX:SET (X,8):NEXT
310 FOR Y=0 TO 8:SET (MX+1,Y):NEXT
320 X=0:Y=0
330 SET(X,9):SET(MX+2,Y)
340 E=INKEY$:IF E="" THEN 340
350 IF (ASC(E)<44) OR (ASC(E)>56) THEN E="":GOTO 340
360 RESET(X,9):RESET(MX+2,Y)
370 ON ASC(E)-44 GOTO 640,380,520,530,510,400,330,420,440,470,330,490
380 IF POINT(X,Y)=0 THEN P(X)=P(X)+2X(7-Y):SET(X,Y)
390 GOTO 470
400 IF Y<7 THEN Y=Y+1 ELSE Y=0
410 GOTO 330
420 IF X>0 THEN X=X-1 ELSE X=MX
430 GOTO 330
440 IF X>0 THEN X=X-1 ELSE X=MX
450 IF POINT(X,Y) THEN P(X)=P(X)-2X(7-Y):RESET(X,Y)
460 GOTO 330
470 IF X<MX THEN X=X+1 ELSE X=0
480 GOTO 330
490 IF Y>0 THEN Y=Y-1 ELSE Y=7
500 GOTO 330
510 P(X)=0:FOR I=0 TO 7:RESET (X,I):NEXT:GOTO 330
520 FOR I=0 TO MX:P(I)=0:NEXT:CLS:GOTO 300
530 PRINT @512,;:FOR I=0 TO MX:LPRINT P%(I);:NEXT
540 FOR I=0 TO 3
550 IF PEEK(14312)=63 THEN POKE 14312,C(I) ELSE 550
560 NEXT I
570 FOR I=0 TO MX
580 IF PEEK(14312)=63 THEN POKE 14312,P(I) ELSE 580
590 NEXT I
600 LPRINT:LPRINT:GOTO 330
610 REM Es folgen die zur Initialisierung des Druckers nötigen
620 REM Daten (MX-80, Graphik doppelter Dichte, 11 pixels)
630 DATA 27,76,11,128
640 END
```

Bild 1. Das Programm hilft Sonderzeichen zu entwerfen

```
16   40   68   68   40   16   40   68   68   40   16 ∞
195  129  165  129  153  129  129  129  129  129  0 Σ
0    0    2    1    1    126  128  128  64   0    0 ∫
28   34   34   34   20   8    20   34   0    0    0 α
```

Bild 2. Eingaben und Ergebnisse

Wer suchet, der soll finden

(angeblich)

Zunächst einmal:
All denen, die dieses Bibelzitat lesen, wünsche ich noch ein

Glückliches Neues Jahr 1988,

(welches, wie ich hoffe, inzwischen noch nicht zu alt geworden ist).
Denjenigen, die dies hier nicht lesen, brauche ich das nicht mehr zu wünschen - denn sie *haben* bereits Glück!
Nur für die Unglücklichen, die bei Empfang fremder Disketten sich mit den sogenannten

PDRIVEs

herumquälen, dürfte das folgende vielleicht interessant sein (vorausgesetzt, sie besitzen noch so einen antiquierten Computer wie ich, der mit NEWDOS oder G- resp. H-DOS arbeitet!)

"Als ich noch - der 'Waldbauernbub' war..." hätte ich fast mit *Peter Rosegger* (kein Clubmitglied!) gesagt; also:
"Als ich noch ein Diskothekar war" (in Lauffener Originalton: ein Diskettothekar), habe ich mich oft genug damit herumgequält, eine Diskette, deren PDRIVEs nicht angegeben waren, zum Laufen (diesmal mit nur einem 'f') zu bringen.

(1.Anm.: Dies war aber nicht der Grund für die Aufgabe dieser Aufgabe; ich habe dabei manches gelernt, u.a. eben das, worüber ich hier zu berichten mich anschicke. * Ende der 1.Anm.)

Heute plagt mich das Problem (das übrigens auch von "ID/CMD" keineswegs immer gelöst wird!) nur noch selten. Zum Beispiel, wenn ich versuche, eine Diskette zu lesen, auf der überhaupt nichts drauf ist... O Gott, ist das ein Krampf! Wie erkennt man die Leere einer Diskette?? Ich bedanke mich für die Lehre, die mir hoffentlich jemand darüber erteilt! Es geht kaum etwas über eine solche Leerelehre!

Ein lieber Clubkamerad schickte mir 50 Disketten, die ich mir ansehen durfte, bevor er sie seiner neuen WAA übergab. Ich entschloß mich prompt zu einer Sitzblockade und saß etwa 3 Tage (ohne weggetragen zu werden! Nicht einmal von meiner Frau...)
Als sich die Leseproblematik bei diesem Unternehmen wieder einmal häufte, erinnerte ich mich des hervorragenden Buches von *GROSSER (im RÖCKRATH-Verlag)*, und studierte dort die Seite 4-3 (was soviel wie Seite 3 von Kapitel 4 heißen soll, nicht etwa Seite 4 bis 3! Anscheinend hatte *GROSSER* Nachträge vorgesehen (sind sie erschienen?). Dort ist nachzulesen, wo auf einer ("normalen") Diskette diese verdammten "PDRIVE-Parameter" stehen.

(2. Anm.: Ich halte diese Bezeichnung für unsinnig. Was soll "PDRIVE" anderes sein als die Abkürzung für "Parameters of Drive" ? Also ist 'PDRIVE-Parameter' ein schwarzer Rappe. Ich übersetze das grundsätzlich mit "Drive-Parameter". * Ende der 2.Anm.)

Mit diesem Erwerb unumstößlicher Wahrheiten (danke, großer GROSSER!) gelang es mir dann doch, sämtliche 50 Disks zu lesen - allerdings mit Ausnahme jener, auf denen vermutlich gar nichts stand. SUPERZAP zeigte dennoch einen Sektorinhalt an, den *GROSSER* offensichtlich nicht erfaßt hat...

Da mir z.Z. kein besserer INFO-Beitrag einfällt, möchte ich jenen, die ebenso hilflos vor ihrer Sphinx sitzen, wie einst ich, das Orakel verraten (ohne einen Anspruch auf DELPHI zu erheben) *(wo Fräulein SPHINX auch gar nicht sitzt):*

Die Drive-Parameter einer Diskette sind im 3.Sektor, d.i. der relative Sektor 2 (DRS 2) gespeichert, und zwar in jeder 16-Byte-Zeile eine Parameter-Gruppe, jeweils für die Laufwerke 0 bis 3 sowie die Pseudo-Laufwerke 4 bis 9.
Dieser Sektor läßt sich "mit Hülfe von SUPERZAP" (jawohl, mir wird ganz feyerlich zumute) bei einer normalen Diskette stets lesen.
Ich gebe hier immer vor, daß die Diskette "normal" sein muß. Es gibt andere, z.B. GENIETEXT, wo das nicht geht. (*ZENDER* hat sie weitgehend kopiergeschützt durch Umfunktionierung wichtiger Strukturen.)
Was die einzelnen Bytes besagen, interessiert in diesem Zusammenhang nur insoweit, als es Auskunft über die gesuchten verfl-, verzwickten Parameter gibt. Und das habe ich in der nachstehenden Tabelle in der Reihenfolge zusammengestellt, die für den forschen Forscher relevant ist.

Auf daß ein jeder schneller finde,
Denn Zeitvergeudung ist 'ne Sünde!

Ka-Jot

* Was TSCRIPS nicht kann - - - *

(Ein Vexierrätsel ?)

Wenn dieses INFO erscheint, steht Ostern vor der Haustür.
Das ist die Zeit, in der das Suchen allgemeine Fröhlichkeit verbreitet; nicht nur unter Kindern.
Drum laßt uns den Kindern gleich tun! (Denn "...der soll finden...")
Es geht um Zeichengestaltung. Aber diesmal heißt der Hase, der die Eier legt, nicht *TSCRIPS*, auch nicht *NEWSCRIPT* oder anderswie.

TSCRIPS kann viel. Die Tastenfolge "@P" erlaubt ein sofortiges Umschalten in den Zeichenmodus, in dem man Zeichen in einer 8x12-Matrix entwerfen oder vorhandene sehr schnell ändern kann - sogar während man einen Text schreibt - so daß auch für den in Arbeit befindlichen Text in aller Kürze beliebige Fantasiegebilde zur Verfügung stehen, die man, wenn sie nicht mehr benötigt werden, auch sofort wieder löschen kann.
Ich halte dies für einen besonderen Vorzug von *TSCRIPS*, weil ich für Mathe-Texte, manchmal aber auch für Briefe, viele Sonderzeichen (manchmal auch "Spaßvögel") brauche, die es im integrierten Zeichenvorrat natürlich nicht gibt.

(Anmerkung: *GENIETEXT* besitzt diesen Modus auch; der ist aber sehr viel umständlicher anzuwenden und zerstört meistens den laufenden Text. Andere Textsysteme mit dieser "facility" - deutsch: "Fazilität" - sind mir nicht bekannt.)

Identifizierung der Drive-Parameter

Die PD's *TI / TD / TC / SPT / TSR / GPL / DDSL / DDGA*

(bzw. für GENIE: *TI / TD / SP / SEK / SWZ / EIB / SBIV / AEIV*)

stehen in DRS 2 in der der jeweiligen rel. Laufwerk-Nr. entsprechenden Zeile; für das Systemlaufwerk 0 in Zeile 0 usw.
Es gilt:

<u>TI</u> ==> Byte 2 in binärer Darstellung interpretieren:

Bit 7 gesetzt entspricht: TI = H
" 6 " " TI = K
" 5 " " TI = M

Bittriade 4-3-2 =	001 entspricht:	TI = A
	011 "	TI = B
	100 "	TI = C
	010 "	TI = D
	101 "	TI = E
	000 "	ERROR
Bit 1 + 0 =		Track Stepping Rate

Byte E: Wenn Bit 3 gesetzt, dann TI = L

<u>TD</u> ==> Byte F = 00 01 02 03 04 05 06 07
TD = A B C D E F G H

<u>TC</u> ==> Byte 3 gibt die Anzahl Tracks (Spuren) an

<u>SPT</u> ==> Byte 4 gibt die Anzahl Sektoren Pro Track an

<u>TSR</u> ==> Byte C gibt die Track Stepping Rate (Spurwechselgeschw.) an

<u>GPL</u> ==> Byte 5 gibt die Anzahl Granules Pro Lump an

<u>DDSL</u> ==> Byte 0 gibt den "Lump" (Block) an, in dem das Directory beginnt

<u>DDGA</u> ==> Byte 9 gibt die Anzahl der dem Directory zugeteilten Granules an

<u>Disk-Typ</u>: Byte 7 in binärer Darstellung interpretieren:

	Bit:	7	6	4	2	1	0	
wenn								gesetzt,
dann Disktyp	:	8"	DS	¹)	²)	³)	DD	
sonst	:	5"	SS	¹)	²)	³)	SD	

¹) wenn gesetzt, dann TI=I, Sektorzählung ab 1, sonst ab 0
²) " " , " TI=L, " " 1, " " 0
³) " " , " TI=K, Spurzählung " 1, " " 0

<u>Byte 1</u> gibt die <u>Anzahl der Lumps auf der Diskette</u> an.

KaJott

In *TSCRIPS* kann man die selbst gebauten Zeichen auch schnell nach links, rechts, oben, unten verschieben. Rotieren, Dehnen und Spiegeln geht nicht. Wollte man Zeichen z.B. um einen Winkel drehen oder vergrößern/verkleinern, so muß man sie mit *DOTWRITER* entwerfen und mit *"MANIPULATE"* – einem Zusatzprogramm – manipulieren. *DOTWRITER* und *DOTPRINT* erfordern jedoch viel Zeitaufwand. Will man mehr produzieren, als die genannte Fertig-Software hervorbringt (pro-duziert = "vor-führt"), so besinne man sich auf das eigentliche Werkzeug, daß – o Gott, KaJott, jetzt erliegst Du auch schon dem Trend; aber ich lasse es trotz *TSCRIPS*'s wundervoller Editierkunst mal so stehen – also noch einmal: man besinne sich auf das Werkzeug, das solche Kunstwerke ermöglicht: den Heireß-fähigen Matrix-Nadeldrucker (so man hat!) Auch der Meißel des Bildhauers gehorcht nicht nur dem schöpferischen Künstler, sondern jedem Lausbuben, der ihn schwingt und den geduldigen Stein damit verunstaltet...

Verunstalten wir!
Bedienen wir uns also der Bitmuster-Druckroutine selbst, statt sie einem geklauften Textprogramm preiszugeben!
Ich habe den EPSON RX80. Wer einen anderen Drucker hat (natürlich nur einen HRG-fähigen, s.o.), kennt sicher sein entsprechendes Kommando und ändert mein nachstehendes Programm entsprechend (Zeile 360)
Die Bilder werden aus aufeinander folgenden Punktspalten (beim RX80 zu je 9 senkrecht übereinander liegenden Punkten) zusammengesetzt. Jeder Punkt repräsentiert ein Bit; die Bitzählung beginnt unten. Der unterste Punkt ist vom USER (Unsereiner Spricht Englisch, Right?) nicht ansprechbar; der nächste entspricht Bit 0, der oberste Bit 7. Jede Punktspalte wird als Byte codifiziert, daß das sich aus der Addition der gesetzten Bits ergibt. Ein "!" ist also eine Spalte, in der – von ganz unten gezählt – der dritte Punkt (Bit 1) sowie die Punkte 6 bis 9 (Bits 4 bis 7) gesetzt sind. Das ergibt das "Bitmuster" 1111 0010; und dies ist gleich F2Hex = 242dez. So erhält jede Spalte ihren Code, und da wir die Spaltencodes für unsere Bilder selbst konstruieren, können wir ihre Abfolge auch selbst manipulieren. Denn jede Spalte wird in unserem Programm durch eine hierdurch definierte Variable repräsentiert. Es empfiehlt sich, hierfür ein- oder noch besser zweidimensionale Feldvariable zu verwenden, denn diese lassen sich am elegantesten "manipulieren".

Mit nachstehendem Programm wird das Vorgehen verdeutlicht, und zwar an einem Beispiel, das euch etwas er-GOETZ-liches vorführt, wenn ihr es ausdruckt. Wer es nicht gleich erkennt, gehe ins Badezimmer. (Es braucht übrigens weder eine Badewanne noch eine Dusche darin zu sein. Aber die Hardware, die das Vexierbild von Sais entschleiert, ist bestimmt drin... *)

> Anmerkung: Da der EPSON die "Characters" 0, 10, 11 und 12 vom TRS80 nicht wunschgemäß akzeptiert, wurde ein Drucker-Hilfstreiber eingebaut. (*ARNULF* hat einen besseren. Aber dieser tut's auch.)

Ich bin sicher, ihr spiegelt euch in euren Texten wi(e)der!
* * * Ka – Jot * * *

*) Für das "i" genügte *TSCRIPS*! (p.H.: ï)

```
10 '              Ein "erGOETZlicher" Ausspruch
20 '              (nicht von KaJot Muehlenbein,
30 '        aber widergespiegelt im Februar 1988 in Weinheim)
40 '
50 '   Bitmuster-Codes (Punktspalten)
60 DATA 0, 255, 136, 136, 136, 136, 136, 255, 0      ' A
70 DATA 0, 255, 145, 145, 145, 145, 145, 110, 0      ' B
80 DATA 0, 255, 129, 129, 129, 129, 66, 60, 0        ' D
90 DATA 0, 255, 145, 145, 145, 145, 129, 129, 0      ' E
100 DATA 0, 0, 0, 129, 255, 129, 0, 0, 0             ' I
110 DATA 0, 255, 16, 40, 68, 130, 129, 1, 0          ' K
120 DATA 0, 255, 1, 1, 1, 1, 1, 1, 0                 ' L
130 DATA 0, 255, 64, 48, 8, 4, 2, 255, 0             ' N
140 DATA 0, 255, 132, 132, 132, 132, 72, 48, 0       ' P
150 DATA 0, 255, 136, 136, 140, 138, 81, 33, 0       ' R
160 DATA 0, 98, 145, 145, 145, 137, 137, 70, 0       ' S
170 DATA 0, 128, 128, 128, 255, 128, 128, 128, 0     ' T
180 DATA 0, 252, 2, 1, 1, 1, 2, 252, 0               ' U
190 DATA 0, 192, 48, 12, 3, 12, 48, 192, 0           ' V
200 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0                   ' Zwischenraum
210 '      T E X T :
220 DATA1,7,7,4,8,15,5,11,12,15,3,1,11,15
230 DATA3,4,8,6,4,8,15,4,10,7,1,13,2,12,15
240 DATA14,5,4,7,4,8,15,2,7,4,5,2,12,15
250 DATA4,11,15,4,10,11,9,1,10,12
260 GOSUB410                                   ' zum Drucker-Hilfstreiber
270 CLS:CLEAR400:DEFSTRA-Z:DIMX(15),XX(15),C%(53)
280                                     ' Konstruktion der Buchstaben
290 FORJ%=1TO15:FORI%=1TO9
300 READC%:X(J%)=X(J%)+CHR$(C%)
310 NEXTI%
320                                     ' Spiegelung
330 FORK%=9TO1STEP-1
340 XX(J%)=XX(J%)+MID$(X(J%),K%,1)
350 NEXTK%,J%
360 LPRINTCHR$(27);"K";CHR$(221);CHR$(1);' Bitmuster-Modus ein
365 '
370 ' Text aus DATA-Zeilen 220-250 einlesen und drucken :
375 '
380 FORI%=1TO53:READC%(I%):NEXT
390 FORI%=53TO1STEP-1:LPRINTXX(C%(I%));:NEXT
400 END
405 '        Drucker-Hilfstreiber fuer EPSON :
410 B$="21E837CB7E20FC211100397E32E837C9"
420 A=16571:FORP=1TO32STEP2
430 B=ASC(MID$(B$,P,1))-48:IFB>9THENB=B-7
440 T=ASC(MID$(B$,P+1,1))-48:IFT>9THENT=T-7
450 POKEA,B*16+T:A=A+1::NEXTP
460 POKE16422,187:POKE16423,64
470 RETURN
```


Die *"80 MICRO"* hat uns Tandianern ja nicht mehr viel zu bieten. Deshalb greift man gern auf ältere Ausgaben zurück und schaut nach, was dort noch auszuschöpfen ist.
Da ist z.B. in der April-Nummer von 1983 auf Seite 210 ein Artikel von *Arne Rohde*, welcher beschreibt, wie man innerhalb von VISICALC bequem sowohl Eingabekorrekturen vornehmen als auch Drucker-Steuersequenzen eingeben kann.
Während ersteres eine nicht allzu wichtige facility ist, über deren Wert man streiten kann - denn oft ist es nicht weniger bequem, die Eingabe in ein Feld schlicht zu wiederholen, als eine Korrektur an der fehlerhaften Eingabe mit dieser trickreichen Methode durchzuführen - bietet die zweite die häufig erwünschte Möglichkeit, den Ausdruck eines VISICALC-Rechenblattes individuell zu formatieren, z.B. durch Unterstreichen, Kursivschrift, Fettschrift etc. Besonders häufig benutze ich diese "facility", um den Drucker von VISICALC aus auf ELITE--Schrift oder Schmalschrift einzustellen, wenn es sich wegen unvorhergesehen größerer Spaltenanzahl, die mit PICA nicht in die Zeile paßt, als notwendig herausstellt oder wenn ich vor dem Laden von VISICALC vergessen hatte, den Drucker auf irgendeinen erwünschten besonderen Modus einzustellen.
Auch Grafik-Zeichen können mit dieser Utility ausgegeben werden (falls der Drucker dazu fähig ist).
Bei der Eingabe von Drucker-Steuersequenzen bedient das Programm sich eines Trickes: Da Zeichen wie *ESC*, @, *FF (Form Feed)*, *SI (Schmalschrift ein)*, *DC2 (Schmalschrift aus)*, *SO (Breitschrift ein)*, *DC4 (Breitschrift aus)*, *BEL* usw. nicht von der Tastatur aus direkt eingegeben werden können, wurde eine Umleitung programmiert: Auf ← (Linkspfeil) wird das eingegebene Zeichen in der ASCII-Tabelle um zwei Spalten nach links verschoben, auf → (Rechtspfeil) um acht Spalten nach rechts.

Ein Beispiel für den EPSON RX 80 und Verwandte:
Wenn einem erst unmittelbar vor dem Ausdruck des Rechenblattes einfällt, daß der Druck *fett* sein soll, kann man die entsprechende Steuersequenz wie folgt an den Drucker ausgeben:
Nachdem die bekannte Zeichenfolge / P P eingegeben wurde, erscheint in der Menue-Zeile:
LOWER RIGHT, "SETUP, &, -, +
Statt nun die "untere rechte Ecke" einzugeben, drückt man zunächst die Anführungsstriche ("). Dadurch gelangt man in den Modus der Druckersteuerung. Die Fettdruck-Einstellung lautet *ESC CHR$(69)* bzw. *CHR$(27);"E"*. Gibt man nun den Linkspfeil ← ein, so wird der ASCII-Wert des nächsten Zeichens um 32 vermindert. Um das Zeichen ESC = CHR$(27) zu erhalten, lautet also die Eingabe <← ;>. Denn der ASCII-Wert des Semikolons ist 59. Da 59-32=27, empfängt der Drucker durch diese Eingabe das richtige Signal, das ihn "hellhörig" macht, nämlich 'ESC' (ESCAPE = 'Fluchtsequenz'). Gibt man anschließend 'E' ein, so reagiert er dann entsprechend durch Fettdruck, sobald der Druckbefehl durch 'ENTER' und Eingabe der "unteren rechten Ecke" abgeschlossen ist. Stattdessen kann man aber beliebig oft wieder das Anführungszeichen eingeben, um weitere Steuersequenzen anzuschließen. Will man z.B. die drei Effekte :

ELITE (ESC M), Kursivschrift (ESC 4) und Unterstreichung, so lautet die Tastenfolge:

/ P P " ← ; M ENTER " ← ; 4 ENTER " ← ; - 1 ENTER

Danach weiter wie gehabt: "Untere rechte Ecke" usw. * Die Aufhebung dieser Befehle erfolgt entsprechend.
Mit dem Rechtspfeil (→) kann man zu anderen Zeichen gelangen, die der Drucker zwar bereithält, die von der Tastatur aus aber sonst nicht erreichbar sind. Er bewirkt die Addition von 128 zum ASCII-Wert des eingegebenen Zeichens.
(Ein sehr ähnliches "Umlenkungsverfahren" verwendet *ZENDER* in seinem *GENIETEXT*.)

Man lasse sich nicht beirren (be-irren; nicht beir-ren), wenn auf Drücken des Linkspfeils ein 'H' und statt des Rechtspfeils ein 'I' erscheint. Diese Buchstaben werden durch die anschließende Eingabe weggeschoben. Was sie bedeuten, habe ich nicht herausgefunden. Wahrscheinlich dienen sie nur als Bestätigung, daß man den richtigen Pfeil gedrückt hat. Oder als Gelächter über den User, wenn er zuviele ←X→ ohne weitere Eingaben hintereinander drückt. Denn dann antwortet sein Dialogpartner eben 'HIHI'!

Eine Kurzanleitung findet sich unten (oder nebenan).
Ich hätte auch das Source-Listing assembliert ausdrucken können. Doch da das sowie niemand abtippt *) und ich das Programm gern jedem Einsender einer Diskette mit Rückporto und Parameterangabe (möglich sind alle) zur Verfügung stelle, unterläßt dies:

Euer *K*a-J*o*t*

Ka-Jot

*) 3 (drei) Seiten

(Anlage!)

Kurzanleitung für VCMOD/CMD

ACHTUNG: VCMOD/CMD läuft nur, wenn auf VC/CMD zugegriffen werden kann!

1) EDIT-Mode

;	Replace-Mode einschalten. - Dann:
↓	Insert-Mode und zurück (durch Vorsetzen von " kann aus einer numerischen Eingabe ein Label gemacht werden)
CLEAR	Zeichen löschen (durch Löschen von " kann aus einem Label ein numerischer Eintrag gemacht werden)
BREAK	EDIT-Mode ohne Änderung verlassen
ENTER	Rückkehr in den normalen Eingabe-Modus. (Danach ENTER oder Pfeiltaste drücken)

2) Druckersteuerung

Tastenfolge: / P P " ←	;	ESC (Prefix)			
	/	Schmalschrift	ein	2	aus
	.	Breitschrift	ein	4	aus
	; - 1	Unterstreichg.	ein	- 0	aus
	; 4	Kursivschrift	ein	5	aus
	; E	Fettdruck	ein	F	aus
	; M	ELITE	ein	P	aus
	; @	Drucker RESET			
u.s.w.					

Anm.1: ← schiebt das eingegebene Zeichen in der ASCII-Tabelle zwei Spalten nach links;
→ schiebt das eingegebene Zeichen 8 Spalten nach rechts.

Anm.2: Mehrere Steuerbefehle können, durch ENTER getrennt, hintereinander eingegeben werden.
Sie werden durch Anführungszeichen <"> eingeleitet.

Kleine Hexen bevorzugt

Nur noch unverbesserliche Nostalgiker im Club benutzen in selbstquälerischer Mühwaltung DISASSEM, um für EDTASM eine Source zu erzeugen. Nun ja, ein Indianer kennt keinen Schmerz. Allgemein hat aber sich die Einsicht verbreitet, daß man das mit DSMBLR für ZEUS tun sollte.

DSMBLR erzeugt 16 Bit-Labels, die mit M beginnen, gefolgt vom Zahlenwert in Hex. Bei 8 Bit-Werten wird entweder die ASCII-Entsprechung zwischen Hochkommata gesetzt oder die Hexzahl mit einem H dahinter. Schönere Labels und Zahlenoperanden kann man sich nicht wünschen.

Der Haken: DSMBLR kennt nur Großbuchstaben. So kann eine Zeile z. B. so aussehen:

```
MFF00     LD     A,0A4H
```

Natürlich macht es wenig Spaß, mit ZEUS die entsprechenden Labels mit dauerhaft niedergehaltener Shift-Taste für die Majuskeln einzutippen. Deshalb ist die erste Aktion nach einer Disassembly immer, mit der C-Funktion von ZEUS Kleinbuchstaben zu erzeugen:

cA,a - cB,b - cC,c - ... - cH,h - cM,m

Wozu ist ein Computer da, wenn nicht zum Automatisieren solch stumpfsinniger Vorgänge?

Zuvor hatte ich in DSMBLR nach den Stellen gespürt, wo die Großbuchstaben erzeugt werden. M und H konnten bereits dort in m und h verwandelt werden (s. u.). Aber die Hex-Ziffern werden leider auf die gleiche undurchsichtige Weise erzeugt wie durch das DOS in der Routine an 4068h. Hier konnte ich nichts manipulieren, weil ich den äußerst krausen (aber kurzen und schnellen) Algorithmus nicht durchschaue.

Das hier vorzustellende Programm erledigt das. Es geht ziemlich simpel; da wird bei den Hexziffern einfach nur das Bit 5 gesetzt. Daher verweise ich zur nicht allzu komplizierten Programmlogik nur auf die reichlichen Kommentare. Etwas schwieriger war, das Programm herausfinden zu lassen, was eine Hex-Ziffer ist und was z. B. ein Zeichen aus einem Mnemonic.

Eine DSMBLR-Source hat als mögliche Zeichen oberhalb ASCII 30h nur die Ziffern (bis F), H, M und die Großbuchstaben, aus denen sich die Mnemonics zusammensetzen. Daher konnten die nunmehr kleinen Buchstaben m und h für die Suche nach einem Label oder einer Hex-Zahl herhalten. Eine Verwechslung mit dem M aus dem Befehl JP M,nn oder dem H von PUSH qq war nun nicht mehr möglich.

Um dies zu erreichen, müssen in DSMBLR/CMD folgende Änderungen durchgeführt werden: Im filerelativen Sektor 7 wird das sektorrelative Byte 01h von 4Dh (M) in 6Dh (m) geändert. Im Sektor 6 wird das Byte A4h und im Sektor 8 das Byte 6Ah von 48h (H) auf 68h (h) gesetzt. Möglicherweise sind durch fleißiges Patchen nicht bei allen in Umlauf befindlichen Versionen die Diskadressen dieselben. Dennoch lassen sich die zu ändernden Stellen leicht finden; es sind die einzigen, die LD A,'M' (3E-4Dh) und LD A,'H' (3E-48h) lauten. Daher sind sie leicht zu finden, z. B. mit FED.

Um die Konversion von Groß- in Kleinbuchstaben abzufahren, wird das Programm (bei mir heißt es CONVERT/CMD) mit seinem Namen, gefolgt vom Namen des zu modifizierenden Source-Files aufgerufen:

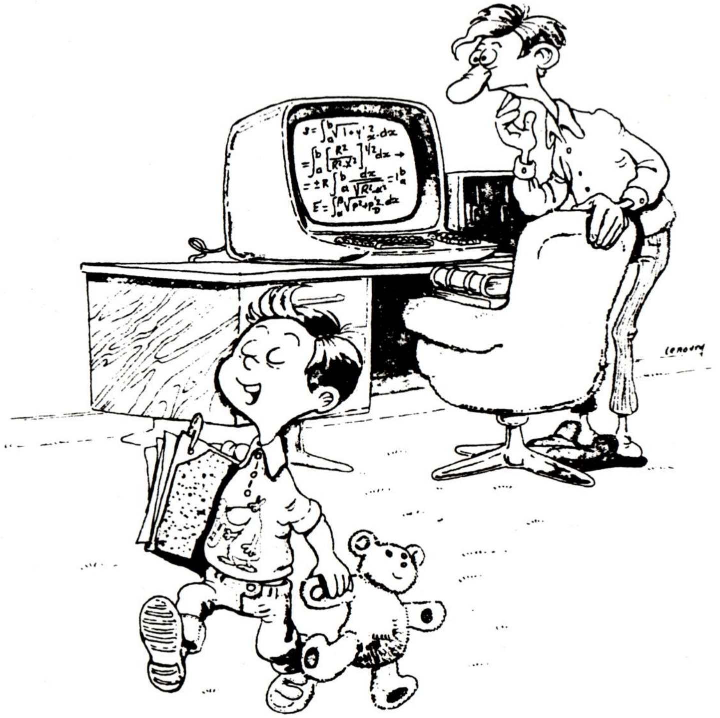
CONVERT,SOURCE<CR>

Wenn keine Typbezeichnung folgt, wird automatisch /SRC angenommen. Das ist die Extension, die DSMBLR generiert, und die ZEUS als gegeben unterstellt, wenn nichts anderes eingegeben wurde.

Arnulf Sopp


```
               ;       Programm zur Konversion von Groß- in Kleinbuchstaben
               ;       in Sources, die von DSMBLR erzeugt wurden
               ;       (notwendige Änderung in DSMBLR: M -> m und H -> h)
               ;       Arnulf Sopp 1985

5200                   ORG   5200h
5200  CDD54C   start   CALL  4cd5h          ;HL auf Dateinamen im Befehlsstring
5203  2844             JR    Z,synterr      ;falls keine Datei angegeben
5205  EB               EX    DE,HL          ;DE <- Dateiname
5206  215052           LD    HL,extensn     ;Default-Extension /SRC
5209  CD7344           CALL  4473h          ;anhängen, falls keine Ext. angegeben
520C  EB               EX    DE,HL          ;HL <- Dateiname
520D  118044           LD    DE,4480h       ;Adresse des DOS-FCB
5210  CD1C44           CALL  441ch          ;Namen in den FCB übertragen
5213  2036             JR    NZ,errexit     ;falls Fehler (z. B. kein gültiger Name)
5215  210053           LD    HL,5300h       ;Sektorpuffer
5218  CD2444           CALL  4424h          ;File eröffnen
521B  202E             JR    NZ,errexit     ;falls Fehler aufgetreten
521D  EB               EX    DE,HL          ;HL <- FCB-Adresse
521E  23               INC   HL             ;FCB +1. Bit 6: EOF-Wert nicht nach jedem
521F  CBF6             SET   6,(HL)         ;Schreibzugriff auf den NEXT-Wert setzen!
5221  2B               DEC   HL             ;zurück auf FCB +0
5222  EB               EX    DE,HL          ;Register zurücktauschen
5223  CD3644   loop0   CALL  4436h          ;einen Sektor einlesen
5226  2805             JR    Z,seekm        ;falls kein Fehler aufgetreten ist
5228  FE1C             CP    1ch            ;Fehlercode "Ende der Datei angetroffen"?
522A  C8               RET   Z              ;fertig, falls ja
522B  181E             JR    errexit        ;sonst anderer Fehler: anzeigen und raus
522D  E5       seekm   PUSH  HL             ;Pufferadresse retten
522E  010001           LD    BC,0100h       ;Zähler für CPIR
5231  C5               PUSH  BC             ;brauchen wir noch für CPDR
5232  3E6D     loop1   LD    A,'m'          ;Suchbyte für Labels
5234  EDB1             CPIR                 ;'m' finden
5236  201B             JR    NZ,seekh       ;falls Suche erfolglos beendet
5238  0604             LD    B,4            ;4 Stellen des Labels
523A  7E       loop2   LD    A,(HL)         ;Zeichen laden
523B  FE41             CP    'A'            ;Hex-Ziffer?
523D  3802             JR    C,cipherm      ;falls Dez-Ziffer
523F  CBEE             SET   5,(HL)         ;Klein- aus Großbuchstaben machen
5241  23       cipherm INC   HL             ;auf nächste Ziffer stellen
5242  0D               DEC   C              ;Bytezähler für CPIR nachstellen
5243  280E             JR    Z,seekh        ;dort weiter, falls Sektor überschritten
5245  10F3             DJNZ  loop2          ;bis gegf. 4 Zeichen umgewandelt
5247  18E9             JR    loop1          ;das nächste Label mit 'm' suchen
5249  3E34     synterr LD    A,34h          ;Fehlercode "Syntax o. Trennzeichen ..."
524B  F680     errexit OR    80h            ;Bit 7 setzen für Rücksprung aus 4409
524D  C30944           JP    4409h          ;Fehler anz., evtl. zum Aufrufer zurück
5250  53       extensn DM    'SRC'          ;Default-Extension für Source-Files
5253  2B       seekh   DEC   HL             ;Zeiger auf Ende des Sektorpuffers 53FF
5254  C1               POP   BC             ;Zähler 0100 für CPDR
5255  3E68     loop3   LD    A,'h'          ;Kenner von Hex-Zahlen
5257  EDB9             CPDR                 ;nach 'h' suchen
5259  2019             JR    NZ,wrsect      ;Skt. zurückschr., falls Suche erfolglos
525B  0604             LD    B,4            ;max. 4 Stellen der Zahl
525D  7E       loop4   LD    A,(HL)         ;Zeichen laden
525E  FE30             CP    '0'            ;eine Ziffer?
5260  3004             JR    NC,hexciph     ;weiter, falls nicht TAB, CR oder Komma
5262  0600             LD    B,00h          ;für restlichen CPDR-Zähler 00xx
5264  18EF             JR    loop3          ;nächstes 'h' suchen
5266  FE41     hexciph CP    'A'            ;Hex-Ziffer?
5268  38C2             JR    C,cipherh      ;falls Dez-Ziffer
526A  CBEE             SET   5,(HL)         ;Klein- aus Großbuchstaben machen
526C  2B       cipherh DEC   HL             ;eine Stelle zurück auf nächste Ziffer
526D  0D               DEC   C              ;Sektorpuffer schon unterschritten?
```

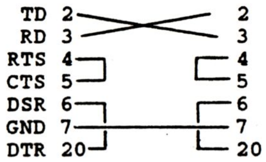
```
526E 2804      00065          JR     Z,wrsect      ;falls ja
5270 10EB      00066          DJNZ   loop4         ;bis ggf. 4 Zeichen umgewandelt
5272 18E1      00067          JR     loop3         ;nächstes 'h' suchen
5274 E1        00068 wrsect   POP    HL            ;Pufferadresse
5275 CD4544    00069          CALL   4445h         ;FCB auf alten Sektor zurückpositionieren
5278 20D1      00070          JR     NZ,errexit    ;falls Fehler aufgetreten
527A CD3944    00071          CALL   4439h         ;Sektor zurückschreiben
527D 20CC      00072          JR     NZ,errexit    ;falls Fehler
527F 18A2      00073          JR     loop0         ;sonst nächsten Sektor bearbeiten
5200           00074          END    start         ;dort Einsprung

00000 Fehler

cipherh 526C   cipherm 5241   errexit 524B   extensn 5250   hexciph 5266   loop0  5223
loop1   5232   loop2   523A   loop3   5255   loop4   525D   seekh   5253   seekm  522D
start   5200   synterr 5249   wrsect  5274
```



Tut mir leid, Papi – Dein Computer ist zu primitiv für meine Hausaufgaben.

Datenübertragung / Model 4P als Terminal

Durch die bereits serienmäßig in das Model 4P (Model 4 nachrüstbar) eingebaute RS 232 Schnittstelle und dem unter TRSDOS 6.x mitgelieferten Kommunikationsprogramm COMM ist das Model 4 gut gerüstet für Daten(fern)übertragung.
Datenfernübertragung mittels Akustikkoppler war mir bereits bekannt. Durch einen mir kürzlich zugelegten frei steckbaren RS 232-Stecker war es mir nun aber möglich, einen Null-Modem-Stecker herzustellen und so die Möglichkeiten einer Direktverbindung zwischen zwei Computern zu testen. Der Null-Modem-Stecker läßt sich auch mit wenig Aufwand selbst herstellen. Allgemein üblich ist nachstehende Verbindung:



Verbindet man nun die beiden Geräte, so lassen sich nicht nur Daten übertragen, was bei zwei verschiedenen Systemen oft die einzige Möglichkeit darstellt, einen Datenaustausch vorzunehmen. Eine weitere Möglichkeit ergibt sich auch durch die Benutzung des einen Geräts als Hauptcomputer und des anderen als Ein- oder Ausgabeterminal. Auf diese Art und Weise müßte man ein für uns nicht kompatibles System (z.B. MS-DOS) mit einem Model 4 bedienen können. Oder man setzt das Model 4 einfach für einen nicht vorhanden Monitor/Tastatur eines Fremdsystems ein.

Nachfolgend nun einige Verbindungsmöglichkeiten, die ich mal versucht habe. Als zweites Gerät hatte ich ebenfalls ein Model 4, was natürlich eine einfache Installierung zuließ. Große Schwierigkeiten dürften sich jedoch bei anderen Systemen ebenfalls nicht ergeben.

Zunächst muß bei beiden Geräten der Treiber für die RS 232 Schnittstelle installiert werden. Dies geschieht durch den Befehl

SET *CL TO COM/DVR

Anschließend werden die Parameter für die Datenübertragung eingegeben (Baud, Word, Parity). Gute Erfolge hatte ich mit folgender Einstellung:

SETCOM (BAUD=9600,WORD=8,PARITY=NO)

Bei einem Versuch, die Übertragung mit 19200 Baud vorzunehmen, stieg das System aus.
Nun noch kurz der Unterschied zwischen den zu benutzenden Befehlen LINK und ROUTE. LINK koppelt ein Gerät an das andere an, während ROUTE ein Gerät auf ein Ersatzgerät umleitet.
Der Monitor wird als "DO", die Tastatur als "KI" und die RS 232 Schnittstelle als "CL" bezeichnet.
Durch COMM *CL wird das Kommunikationsprogramm von TRSDOS aus aufgerufen.
Bei den nachstehenden Beispielen ist davon auszugehen, daß es sich bei Gerät 2 um das Model 4/4P handelt. Gerät 1 stellt das evtl. Fremdsystem dar.

Bsp. 1:

```
      Gerät 1:        LINK *DO *CL
      Gerät 2:        COMM *CL
```

Mit diesem Aufbau wird Gerät 2 im Prinzip nur als Zweit- oder Ersatzmonitor benutzt. Das Gerät 1 dient als Hauptcomputer, während Gerät 2 nur die Daten von Gerät 1 empfängt, jedoch keine Daten an Gerät 1 senden kann.

Bsp. 2

```
      Gerät 1:        LINK *KI *CL
      Gerät 2:        COMM *CL
```

Gerät 1 ist wieder der Hauptcomputer. Dieses Mal ist anstatt des Monitors jedoch die Tastatur angekoppelt. Die Befehle für Gerät 1 können nun noch zusätzlich/ersatzweise über Gerät 2 eingegeben werden. Sollen die eingegebenen Befehle auch auf Gerät 2 angezeigt werden, sind innerhalb COMM noch die Anweisungen DUPLX ON und ECHO ON einzugeben. Befehle an Gerät 1 können von beiden Geräten eingegeben werden.

Bsp. 3

```
      Gerät 1:        LINK *KI *CL
                      LINK *DO *CL
      Gerät 2:        COMM *CL
```

Wie Bsp. 2, nur daß jetzt das Ergebnis des an Gerät 1 gesandten Befehls auch auf Gerät 2 angezeigt wird.

Bsp. 4

```
      Gerät 1:        LINK *KI *CL
                      ROUTE *DO *CL
      Gerät 2:        COMM *CL (evtl. HNDSHK ON)
```

Hier wird die Bildschirmausgabe auf Gerät 1 unterdrückt und direkt an Gerät 2 geleitet. Gerät 1 wird nun direkt von Gerät 2 aus bedient. Dieser Aufbau läßt ein Arbeiten mit Gerät 1 zu, wenn dort z.B. Tastatur und Monitor nicht vorhanden sind. Voraussetzung wäre dann allerdings, daß beim Booten von Gerät 1 die o.a. Einstellung automatisch vorgenommen wird.

Bei der Bildschirmausgabe auf Gerät 2 ist zu beachten, daß die Daten seriell, also Zeile für Zeile, ausgegeben werden. Ursprünglich direkt adressierte Bildschirmpositionen werden also auf Gerät 2 direkt hintereinander ausgegeben.

Weiterhin ist noch zu beachten, daß die Datenausgabe von Gerät 1 über das Betriebssystem erfolgen muß. Teilweise werden die auszugebenden Daten unter Umgehung des Betriebssystem durch direkte Adressierung ausgegeben bzw. eingegeben. Öfters findet man dies bei MS-DOS unter 8088-Proz., um so das System etwas schneller zu machen. Auch bei der Grafikausgabe kann es zu Problemen führen.
Hauptsächlich einsetzen lassen sich diese Verbingungen, um Betriebssysteme zu bedienen. Für die Bedienung von Anwenderprogrammen ist diese Verfahrensweise nur bedingt geeignet.

Es wäre interessant zu wissen, bei welchen Systemen sich das Model 4 als Terminal einsetzen läßt. An einem Erfahrungsaustausch wäre ich interessiert.

(Den oben erwähnten RS-232-Stecker gibt es für DM 18,25 DM bei Westfalia Technica, 5800 Hagen. Die Signale sind durch Brücken beliebig zu vertauschen).

Klaus Hermann

Druckausgabe für COBOL-Programme

Der Schreibaufwand für COBOL-Druckausgaben ohne Report-Generator ist auf Grund der fehlenden expliziten Adressierung und Längenangaben sehr groß. Jede Druckzeile muss in der Form

```
   01  Druckzeile1.
       02  FILLER               PIC  X.
       02  LAUFENDE-NR          PIC  X(5).
       02  FILLER               PIC  X.
       02  NAME                 PIC  X(20).
       02  FILLER               PIC  X.
       02  VORNAME              PIC  X(20).
       02  FILLER               PIC  X.
           .
           .
```

definiert werden. Müssen [redacted] viele unterschiedliche Druckzeilen programmiert werden, so ist der Schreibaufwand nicht unerheblich.

Auf folgende Weise läßt sich hier Abhilfe schaffen. Man schreibt nachstehendes COPY-Element ab und speichert sich dieses unter einem beliebigen Namen (z.B. DRUCK).

```
   01  DRUBER                   PIC  X(80).
   01  D001 REDEFINES DRUBER    PIC  X(01) JUSTIFIED RIGHT.
   01  D002 REDEFINES DRUBER    PIC  X(02) JUSTIFIED RIGHT.
   01  D003 REDEFINES DRUBER    PIC  X(03) JUSTIFIED RIGHT.
           .
           .
   01  D080 REDEFINES DRUBER    PIC  X(80) JUSTIFIED RIGHT.
```

Dieses COPY-Element kann nun mit der COBOL-Anweisung 'COPY DRUCK' für alle zukünftigen Programme benutzt werden. Die langweilige Definition des oben stehenden Ausgabebereichs ist somit nur eine einmalige Sache.

Das Füllen des Druckbereichs erfolgt nun von RECHTS nach LINKS. Vor Bearbeitung des Ausgabebereichs wird dieser gelöscht. Wenn als ein auszugebender Wert auf die Schreibposition 60-63 plaziert werden soll, so erfolgt die Übertragung mit (es muß der Name der höchsten Druckstelle angegeben werden)

z.B.

```
        MOVE SPACE TO DRUBER.
        MOVE "TEXT" TO D063.
        MOVE "UEBERSCHRIFT" TO D037.
        WRITE ZEILE FROM DRUBER AFTER 1.
```

Viel Spaß

Werner Förster

Vergleichstest

Vorgeschichte

Wie ihr im letzten Info lesen konntet, habe ich ein Model 100 in Besitz, auf dem viele meiner Briefe und Texte (so z.B. dieser) entstehen. Zum Erstellen der Texte kann man den 100'ter gut verwenden, zum Drucken derselben taugt das Computerchen allerdings nicht. Daher schiebe ich die Geistesergüsse nach dem tippen über die RS232-Schnittstelle auf das Model 4p. Hier werden sie mittels TSCRIPS formatiert und gedruckt.

Störend wirkt sich dabei aus, daß die Umlaute des Model 100 (selbstverständlich) an einer anderen Stelle im Zeichensatz liegen, als die des 4p. Dieses Problemchen ist aber mit einem kleinen Konvertierungsprogramm sehr leicht zu lösen. Dieses Programm entstand zunächst in BASIC und glänzte vor allem durch seine Lahmheit. Aus diesem Grunde schrieb ich, in einer ruhigen Minute, ein Programm in Pascal.
Unter NewDOS laufen auf dem TRS 80 mehrere Pascal-Dialekte. Den ersten Versuch startete ich mit PASCAL 80. Nachdem das relativ gut klappte, kam ich auf die wahnwitzige Idee, das Programm auch in PASCAL 4.5 und Alcor-PASCAL zu realisieren und das ganze als eine Art Vergleichstest zu gestalten.

Als nach mehreren Stunden (teilweise hatte ich keine Dokumentation zu den PASCAL-Compilern) alle drei Programme liefen, wollte ich das Problem auch noch in RPNL umschreiben. Dazu hatte ich zwar eine wirklich gute Dokumentation, war aber trotz mehrstündiger verzweifelter Versuche nicht in der Lage, ein funktionsfähiges Programm zu erstellen. Als ich dann auf dem Nordlichtertreffen Kurt Müller traf, der ja RPNL unter NewDOS nutzbar machte, verdonnerte ich ihn sofort dazu, ein entsprechenden Programm in "seiner" Sprache zu erstellen. Last but not least wurde Gerald Schröder damit beauftragt, das Umlauteproblem in Assembler zu lösen.

Im Anschluß könnt ihr also zunächst einmal den Vergleichstest zwischen den drei PASCAL-Compilern und im Anschluß daran die Erläuterung der Problemlösung in RPNL und Assembler lesen. Ich wünsche euch viel Spaß bei der Lektüre und hoffe, ihr werdet dadurch dazu angeregt, einmal ein eigenes Problem in PASCAL, RPNL oder Assembler statt in BASIC zu lösen.

Pascal 80

Pascal 80 erinnerte mich bei der ersten Benutzung an Turbo-Pascal, weil es, genau wie dieser wohl bekanntest Pascal-Compiler unter CP/M, Editor, Compiler und Runtimelibary in einem Programm vereint. Nach dem Start erscheint ein Menü mit den Optionen:
- E - Editor
- Q - Quit to DOS
- K - Kill (Clear) Editor
- C - Compile Program in Editor
- R - Run Program in Editor
- S - Save Program in Editor
- L - Load Program to Editor
- A - Append Text to Editor
- W - Write Object Code to Disk
- X - Execute Program from Disk.

Der Editor ist zwar bei weitem nicht so komfortabel wie der, den man von Turbo gewohnt ist, aber im wesentlichen läßt sich gut damit arbeiten. Das Compilieren geht recht flott vonstatten. Dabei wird der Quelltext auf dem Bildschirm gelistet und, falls aufgetreten, ein Fehler angezeigt. Wenn man nach einem Fehler wieder in den Editor geht, befindet sich der Cursor in der Zeile, in der der Fehler aufgetreten ist!

Wurde ein Quelltext fehlerfrei übersetzt, kann man das entstandene Programm mit 'R' direkt vom Menü aus starten. Will man es als /CMD-File abspeichern, tritt eine deutliche Schwäche des Sprachsystems zutage. Um das zu tun, muß man nämlich folgende Schritte durchlaufen:
Quelltext compilieren und Objectcode abspeichern
PASCAL verlassen
AUTHCODE/CMD starten
PASCAL starten
AUTHOR/SRC laden, compilieren und starten
mit der Option 'X' wird dann das gewünschte Programm als /CMD-File auf Diskette geschrieben.
Diese Procedur ist recht umständlich und ziemlich zeitaufwendig. Die Programme AUTHCODE/CMD und AUTHOR/SRC dienen hier dazu, den Objectcode, zusammen mit der Runtimelibary (getrennt von Editor und Compiler) abzuspeichern. Einen richtigen Linker, wie er bei anderen PASCAL-Systemen üblich ist, gibt es hier nicht. Das erklärt auch die ungewöhnliche Größe des entstehenden Programms, in dem immer das gesamte Laufzeitsystem vorhanden ist, unabhängig davon, was gebraucht wird und was nicht!

Auffällig an Pascal 80 ist die Möglichkeit der Verwendung von INCLUDE- und CHAIN-Files, wobei auch diese beiden Funktionen an Turbo-Pascal erinnern. Mit INCLUDE kann man Quelltexte, die in einer Diskettendatei stehen, whärend des Compilierens in den im Editor befindlichen Sourcecode einbinden. CHAIN ähnelt den Overlays von T.-P. und erlaubt es, separat compilierte Objectdateien aus einem laufenden Programm heraus aufzurufen.

Abschließend wäre zu sagen, daß man Pascal 80 sein Alter (1981) nicht ansieht und nur manchmal, vor allem bei Diskoperationen, daran erinnert wird, daß es urspünglich für TRSDOS geschrieben war (aber unter NewDOS sowohl auf dem Model 1 als auch auf dem 3/4/4p lauffähig ist). Das mit 60 Seiten nicht allzu umfangreiche Handbuch erklärt die Möglichkeiten, Erweiterungen (gegenüber Standard-Pascal) und auch die Einschränkungen des Systems sehr gut.

Pascal 4.5

Pascal 4.5 besteht, im Gegensatz zu Pascal 80 oder Turbo-Pascal, aus drei Komponenten:
- Editor -> PASEDIT/CMD
- Compiler -> PASCT/CMD
- Laufzeitsystem -> PASCAL/CMD.

Das mag den BASIC-Freak zunächst einmal abschrecken, ist aber bei näherem hinsehen (und einigen Testläufen) gar nicht so schlimm.

PASEDIT ist ein sehr komfortabler Editor, der mich allerdings beim ersten Diskfehler (Schreibversuch auf schreibgeschützte Diskette) fürchterlich erschreckt hat. Nach Auftreten eines DOS-Fehlers landet man nämlich grundsätzlich im Betriebssystem. Welch ein Glück, daß dabei der soeben erstellte Quelltext nicht zerstört wird, sondern bei erneutem Aufruf von PASEDIT restauriert werden kann. Statt des systemeigenen Editors kann man übrigens jederzeit auch ein Textverarbeitungssystem benutzen, solange es in der Lage ist, ASCII-Files auf Diskette abzulegen. Im Prinzip ist also von SCRIPSIT bis LESCRIPT jeder Editor verwendbar!

Nachdem man den Quelltext erstellt hat, steht man vor dem Problem, den Compiler aufrufen zu müssen. Manchem, der sich noch nie mit einem Compiler herumgeschlagen hat, wird diese Formulierung seltsam vorkommen, aber der Compileraufruf ist tatsächlich ein nicht zu unterschätzendes Problem.

Schließlich muß man dem Compiler (ebenso wie manchen Linkern) zusätzlich zum Namen des zu übersetzenden Programms auch noch allerhand weitere Informationen geben (wer sich einmal mit BASCOM herumgeschlagen hat, wir ein Lied davon singen können). PASCT geht mit dem Programmierer da sehr viel sanfter um. Er fragt nicht nur nach dem Namen für Ein- und Ausgabedatei sondern auch nach den gewünschten Optionen, und davon gibt es immerhin sieben.

Weniger gut finde ich die Art, wie der Compiler Fehler anzeigt. Statt Klartextfehlermeldungen, wie es sich in der heutigen Zeit wohl gehört, bekommt man einen Fehlercode (z.B. Error 104) an den Kopf geworfen und muß sich aus einer Liste den Fehlergrund heraussuchen. Und dann beginnt das Spielchen, das einem die Freude an Pascal (und auch an anderen Compilersprachen) vermiesen kann:

1. Quelltext erstellt und abgespeichert
2. Compiler aufrufen
3. Fehlercode notieren (dabei vergißt man mit an Sicherheit grenzender Wahrscheinlichkeit, sich die Zeile zu notieren, in der der Fehler aufgetreten ist!)
4. Editor aufrufen und Fehler beheben und auf ein neues!

Auch Pascal 4.5 hat keinen Linker, vielmehr kann man mit dem Programm PASCAL/CMD die als Objectcode abgespeicherten Programme starten. Um eigenständige /CMD-Files zu erzeugen, muß man den compilierten Objectcode mittels des DOS-Befehls APPEND an die Datei PASCAL/OBJ anhängen. Zuvor sollte man sich natürlich von PASCAL/OBJ ein Backup angefertigt haben!!! Obwohl sich Pascal 4.5 in diesem Punkt nur in der Verfahrensweise, nicht aber im Prinzip von Pascal 80 unterscheidet, entstehen erstaunlich kleine /CMD-Programme. Weiterhin hilft einem das Programm PASJCL/CMD, welches auch im Quellcode vorliegt, beim Erstellen von lauffähigen Programmen, indem es nach verschiedenen Abfragen (Name des zu übersetzenden Quelltextes, Compileroptionen etc.) eine /JCL-Datei anlegt und mit ihr sowohl den Compiler startet als auch das "Binden" zum fertigen Programm erledigt!

Die Möglichkeiten von INCLUDE- und CHAIN-Dateien fehlen bei Pascal 4.5, dafür glänzt es aber durch eine sehr gute Zusammenarbeit mit NewDOS und ein paar sehr brauchbare Sonderbefehle. So kann man mit MOVEDN und MOVEUP Speicherbereiche verschieben (ähnlich dem Z80-Befehlen LDDR und LDIR), PEEKen und POKEn und, was sich bei meinem Problem als besonders nützlich erwiesen hat, mit TITLE(file) Dateinamen direkt von der Tastatur einlesen!

Die Anleitung zu Pascal 4.5 ist noch spärlicher ausgefallen, als die Pascal 80-Anleitung, durch die klarere Gliederung findet man sich aber sehr schnell zurecht. Etwas schwach finde ich die Erklärung der Befehle, die nicht zum Pascal-Standard gehören (z.B. TITLE usw.). Hier hilft nur das Studium der mitgelieferten Quelltexte (PASJCL/PAS) und viel probieren!

Alcor-Pascal

Eigentlich dürfte ich mich über dieses Sprachsystem gar nicht äußern, da ich zu meinen Versuchen damit keinerlei Anleitung vorliegen hatte. Daraus resultierten natürlich viele Fehlschläge und eine gewisse Antipathie gegen diesen Compiler. Trotzdem möchte ich wenigstens einige Worte darüber verlieren, jedoch an dieser Stelle schon bemerken, daß meine Aussagen rein subjektiv sind!

Im Gegensatz zu den Editoren von Pascal 80 und 4.5 kann man mit ED/CMD, dem Alcor-Editor, ohne Anleitung überhaupt nichts anfangen. Selbst heftigem und langwierigen Herumprobieren widersetzt er sich erfolgreich! Einziger Vorteil ist der Umstand, daß der Compiler Quellcodes im ASCII-Format verarbeitet und damit praktisch wieder jeder Texteditor zum Erstellen der Pascalprogramme nutzbar ist. Ich benutze auch hier PASEDIT/CMD, der mir unter Pascal 4.5 ans Herz gewachsen ist!

Der Compiler fragt nach dem Start nach Ein- und Ausgabe-, sowie nach der Listdatei, in die er das Compilerprotokoll ablegen soll. Auch hier werden leider nur Fehlernummern angegeben, den Klartext muß man sich aus einer Fehlerliste holen. Und natürlich kommt dann auch hier die, beim Pascal 4.5 schon beschriebene, umständliche und langwierige Entwanzung des Quelltextes. Über die Übersetzungsgeschwindigkeit an sich kann man sich nicht beschweren.

Nach dem Compilieren kann man sein Programm, inzwischen liegt es als /OBJ-File vor, mit Hilfe des Laufzeitsystems starten (RUN progname/obj). Um ein eigenständiges Programm zu erhalten muß man den Linker zu Hilfe rufen, der die benötigten Routinen aus den verschiedenen Dateien zusammensucht und bindet. Natürlich muß auch hier die komplette Runtimelibary mitgeschleppt werden und die /CMD-Files sind entsprechend groß, dafür kann man aber (so man kann!) von anderen Compilern (z.B. Alcor-C) erzeugte /REL-Dateien in die Pascalprogramme einbinden.

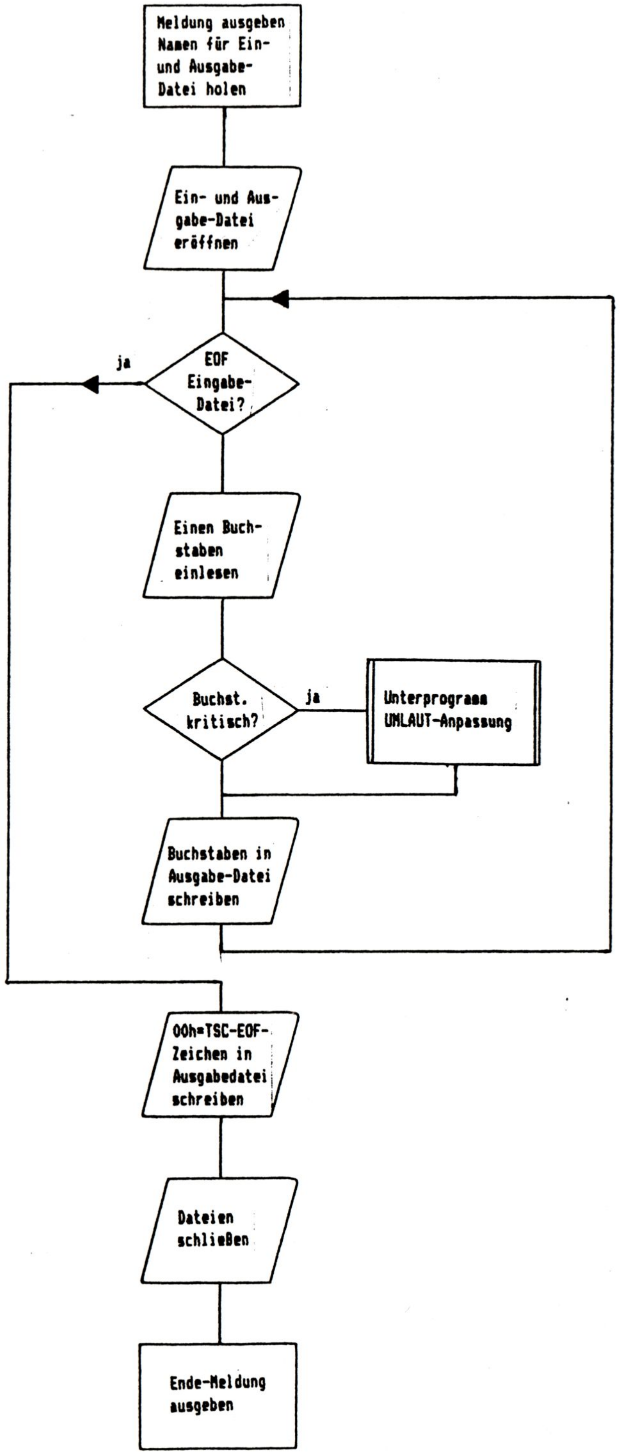
Etwas was mich an Alcor-Pascal sehr stört ist die Tatsache, daß jedes damit erstellte Programm nach seinem Start erst einmal fragt, welches die normalen Wege für INPUT und OUTPUT sein sollen. Sicher ist es manchmal sehr angenehm, an dieser Stelle z.B. den Printer als Ausgabemedium angeben zu können, trotzdem nervt nach mehreren Programmstarts die ständige Fragerei beträchtlich.

Leider kann ich hier nichts über das Handbuch zu diesem System aussagen, denn es wäre sicher ein Pluspunkt für Alcor! Auch scheint der Sprachumfang recht beachtlich zu sein, was ich aus den mir vorliegenden Quelltexten ersehen kann. Was mir weniger gefällt ist das Filehandling, das mir etwas umständlich erscheint. Zu bemerken wäre noch, daß es Alcor-Pascal sowohl für das Model 1 als auch für die Modelle 3/4/4p gibt. Im Gegensatz zu Pascal 80 und 4.5, von denen es nur eine Version für alle TRS 80-Modelle gibt, ist aber der Compiler für das M.1 nicht auf den anderen Maschinen lauffähig und umgekehrt.

Problem und Programmablaufplan

Schon im Vorwort habe ich kurz das Problem angesprochen, welches zu lösen gilt. Hier nun die genaue Problemdefinition: aus einer Datei unbestimmter Länge soll der gesamte Inhalt in eine zweite Datei übertragen werden, dabei sollen verschiedene Zeichen durch andere ersetzt werden.

Aus dieser Aufgabenstellung ergibt sich folgender Ablaufplan: (in den sich übrigens ein Fehler eingeschlichen hat! Die Abfrage "EOF Eingabe-Datei?" muß natürlich am Ende der Schleife "Buchst. einlesen / Buchst. kritisch? / Buchst. ausgeben" stehen. Da ich aber zu faul bin, das Flußdiagramm noch einmal zu zeichnen, lasse ich es bei dieser Anmerkung!)



Programm

Ohne größere Umschweife möchte ich euch jetzt auch gleich die Lösung des Problems in Pascal 4.5 präsentieren.

```
PROGRAM M100TSC (INPUT,OUTPUT,FILEA,FILEB);

VAR C            : CHAR;                  (*Variable deklarieren*)
    CIN          : INTEGER;

PROCEDURE UMLAUT(CIN:INTEGER);            (*Unterprogramm UMLAUT*)

BEGIN
  CASE CIN OF                             (*je nach Wert von CIN
    182: CIN:=123;                          eine der folgenden
    183: CIN:=124;                          Aktionen ausführen!*)
    184: CIN:=125;
    185: CIN:=126;
    177: CIN:=91;
    178: CIN:=92;
    179: CIN:=93;
  END;
  C:=CHR(CIN);                            (*Wieder in Buchstaben
END;                                        umwandeln*)

BEGIN
  WRITE(CHR(28),CHR(31));                 (*Bildschirm löschen*)
  WRITELN('UMLAUT');                      (*Startmeldung ausg.*)
  WRITELN;
  WRITELN('Model 100 -> TSCrips Umlaute-Umwandler');
  WRITELN('C by H. Obermann');
  WRITELN;
  WRITE('Eingabe-Datei? ');
  TITLE(FILEA);                          (*Eingabe-Datei abfr.*)
  WRITE('Ausgabe-Datei? ');
  TITLE(FILEB);                          (*Ausgabe-Datei abfr.*)
  RESET(FILEA);                          (*Eingabe-Datei  eröffnen*)
  WRITELN;
  REWRITE(FILEB);                        (*Ausgabe-Datei eröffnen
                                           sollte eine Datei
                                           gleichen Namens schon
                                           einmal bestehen, wird
                                           diese zuvor gelöscht!*)
  REPEAT                                 (*wiederhole folgendes -*)
      READ(FILEA,C);                     (*Buchstabe einlesen*)
      CIN:=ORD(C);                       (*in Integer umwandeln*)
      IF (CIN > 176) AND (CIN < 186) THEN UMLAUT(CIN);
                                         (*im kritischen Bereich?
                                         Wenn ja, dann Umwandeln!*)
      WRITE(FILEB,C);                    (*Buchstabe schreiben*)
  UNTIL EOF(FILEA);                      (*bis EOF erreicht!*)
  WRITE(FILEB,CHR(0));                   (*EOF für TSCRIPS*)
  CLOSE(FILEA);                          (*Dateien schließen*)
  CLOSE(FILEB);
  WRITELN('Umlaute-Umwandlung beendet!');
  END.
```

Selbstverständlich weiß ich, daß das Programm noch nicht ganz sauber programmiert ist und daß sich die eingefleischten Pascal-Programmierer die Haare raufen werden! Trotzdem zeigt das Programmchen meiner Meinung nach, daß der Umstieg von BASIC zu Pascal gar nicht so schwer ist! Vor allem die Lesbarkeit des Programms ist um Welten besser als die des, von mir an zunächst erstellten, BASIC-Programms. Ganz zu schweigen, von der erheblichen Geschwindigkeitssteigerung!

Vergleich

An dieser Stelle möchte ich einen kurzen Vergleich zwischen den verschiedenen Programmen machen, die, in verschiedenen Sprachen bzw. Dialekten, zur Lösung des Problems inzwischen erstellt wurden.

Sprache/Dialekt	Größe in Grans Quelltext	CMD-File	Geschwindigkeit in Sekunden
Alcor-Pascal	1	9	11.7 - 13.0
Pascal 80	1	10	11.0 - 15.0
Pascal 4.5	1	3	8.5 - 9.2
RPNL	3	3	10.5
Assembler	3	1	7.5 - 8.5

Die Größe der Dateien habe ich in Grans angegeben, weil ich es (zumindest bei einer solchen Anwendung) für uninteressant halte, ob ein Programm 196 oder 248 Byte lang ist. Beide Programme benötigen auf der Diskette auf alle Fälle 1 Granule!

Die Geschwindigkeit wurde mit einem Text gemessen, den ich gerade auf dem Model 100 erstellt hatte und der durch Zufall vorlag. Der Text war ca. 1.5 kByte lang und die Tests erfolgten unter NewDOS auf dem Model 4p mit einer Taktfrequenz von 4 MHz. Die Zeitendifferenzen bei den einzelnen Programmen ergeben sich daraus, daß einmal die Programmnamen mit und einmal ohne Drivenummer angegeben wurden.

Fazit

Eigentlich sollte sich jeder Leser das Fazit aus den obrigen Angaben selbst herauslesen können. Trotzdem hier noch ein paar zusammenfassende Sätze:

1. Wer Wert auf Geschwindigkeit legt und sich von BASIC (auch da gibt es ja Compiler) trennen möchte, dem stehen vor allem die Sprachen PASCAL, RPNL und Assembler zur Verfügung. Ich weiß natürlich, daß ich mir mit dieser Behauptung alle Anhänger anderer Sprachen wie COBOL, Forth usw. zum Feind mache, bin aber erst dann von dieser Behauptung abzubringen, wenn eine Problemlösung in der entsprechenden Sprache im Info erscheint!

2. Am schnellsten ist sicherlich die Assemblerlösung, aber nur Freaks wie Gerald Schröder und Arnulf Sopp schütteln ein solches Assemblerprogramm (so einfach es auch, gegenüber den Programmen, die wir von den beiden sonst gewohnt sind, auch sein mag) innerhalb von ein paar Minuten aus dem Ärmel.

3. Am wenigsten zu Schreiben hat man in Pascal (die Quelltexte in allen drei Dialekten umfassen nur ein Granual, die RPNL- und die Assemblerlösung bringen je 3 Grans auf die Waage!). Hat man die Qual der Wahl zwischen den Pascal-Compilern, weil man alle drei zur Verfügung hat, würde ich folgende Wahl treffen:
Pascal 80 - für Einsteiger und zum Lernen!
Pascal 4.5 - für normale Anwendungen, für Anfänger geeignet!
Alcor Pascal - für den Fortgeschrittenen (und nur mit Handbuch!)

4. Wer sich gerne einer sehr guten (schließlich vereinigt RPNL die Vorzüge von Forth mit denen von Pascal), leider aber sehr seltenen (besser vielleicht exotischen) Sprache zuwenden möchte, die noch dazu sehr maschinennahes Programmieren erlaubt und außergewöhnlich gut dokumentiert ist (sie liegt sogar im Quelltext vor!), kann sich auf RPNL stürzen!

Egal für welche Sprache und welchen Dialekt ihr euch entscheidet, ich wünsche euch auf jeden Fall viel Spaß und Erfolg beim Programmieren, euer

Hartmut Obermann

RPNL

Nachdem Hartmut sich nun ausgiebig zu den PASCAL-Versionen in seinem Test geäußert hat, folgt nun mein bescheidener Beitrag dazu. Da das zu lösende Problem bereits ausgiebig beleuchtet wurde, kann ich mich auf eine Beschreibung des Programmaufbaus beschränken.

Das Programm basiert auf einer Tabelle, in der die fraglichen ASCII-Codes und ihr Ersatz aufgelistet sind:

```
CALL ASCII.TAB  ;Konvertierungs-Tabelle
   B6 7B        ;Erster Eintrag altes Zeichen,
   B7 7C        ;...zweiter Eintag neues Zeichen
   B8 7D
   B9 7E
   B1 5B
   B2 5C
   B3 5D
   00 00        ;Tabellenende.
CEND


CODE TABELLE     ;( -> adr1) Tabellenzeiger auf den Stack
   21 ASCII.TAB ;LD      HL,ASCII.TAB
   E5           ;PUSH    HL
CEND
```

Da RPNL keine DATA-Anweisung kennt, muß, um derartiges zu erreichen, ein Trick angewandt werden. Die Daten-Tabelle wird einfach mit Hilfe der CALL-Definition erstellt (s. CALL ASCII.TAB). Auf dieser Definitionsstufe ist es dem Compile nämlich egal, was für Daten vom Anwender vorliegen. Es muß nicht notwendigerweise ein Unterprogramm sein ! Wenn wir schon freie Hand haben, dann ist es leider auch 'unser Bier', dafür zu sorgen, daß wir auch an die Daten kommen. Dazu muß eine Ebene höher ein weiteres Wort definiert werden, welches eben diesen Zeiger heranschafft (s. CODE TABELLE). Damit wäre der Zugriff auf das Wesentliche vorerst sichergestellt.

Da die Losung 'Austausch von Unerwünschtem gegen wichtigeres' lautet, ist natürlich noch ein Programm von Nöten, welches solches zu leisten im Stande ist. Als erstes der Gegenstand der Diskusion:

```
PROGRAM ?CONVERT         ;(char1 adr1 -> char2 adr1 flag)
    OVER OVER ?(B) =     ;TOS = Tabelleneintrag ?
    IF
       COUNT ? INC COUNT :=
       DUP INC ?(B)      ;Ersatzzeichen holen und...
       SWAP ROT DROP     ;...altes Zeichen ersetzen
       TRUE              ;Setze Erfolgs-Flag
    ELSE
       FALSE             ;Keine Konvertierung
    ENDIF
END
```

```
PROGRAM TEST.ASCII        ;(char1 -> char2)
    DUP DUP               ;prüfe ob Zeichen in der Tabelle
    177 < SWAP            ;Diese Bereichseingrenzung ist
    185 > OR NOT          ;Problemabhängig !!!
    IF
        TABELLE           ;Tabellen-Zeiger auf den Stack
        REPEAT
           ?CONVERT              ;Zeichen ersetzen ?
           OVER ?(B) 0= OR       ;Tabellenende ?
        UNTIL
           INC INC               ;Nein !, nächster Eintrag
        LOOP
        DROP              ;Nur das Zeichen selbst bleibt...
    ENDIF                 ;... auf dem Stack
    CHARACTERS ? INC CHARACTERS :=
END
```

Das Wort ?CONVERT prüft das Zeichen auf dem Stack, ob es in der Tabelle enthalten ist. Wenn ja, wird es gegen das Ersatzzeichen ausgetauscht. Über den Erfolg oder Mißerfolgt gibt ein Flag auskunft, welches mit auf dem Stack übergeben wird. Damit TEST.ASCII nicht an Informationen erstickt, ist es unbedingt notwendig, daß die Menge der Daten nicht zu groß wird. Mehr wie vier Zahlen sind auf dem Stack einfach nicht zu verwalten, ohne daß man in unvermutete Probleme gerät. Deshalb ist es wichtig, das ?CONVERT im Erfolgsfall das alte Zeichen vom Stack entfernt. Wer das nicht einsehen kann, versuche das Problem im Wort TEST.ASCII zu lösen.

Da nur einige wenige Zeichen zu Prüfen sind, wird durch eine Bereichsabfrage eine Vorauswahl durchgeführt. Liegt ein Zeichen im interessierenden Bereich, wird weiter geprüft und die Tabelle durchsucht. Am Ende der Routine wird noch etwas mitgezählt, damit am Schluß die Anzahl der durchgeleierten Zeichen bekannt gegeben werden kann. Das Gleiche passiert in der Routine ?CONVERT, hier wird die Anzahl der Ersetzungen mitgezählt.

Damit ist der Job für ein einzelnes Zeichen bereits erledigt. Fehlt nur noch das Prinzip auf eine beliebige Menge auszudehnen. Im Grunde nichts leichter als das: Man programmiert ein weiteres Wort ! Womit wir beim Hauptprogramm wären. Damit bei der Eingabe der Filenamen kein Murks passieren kann, sorgt das Wort GET.NAMES für saubere Verhältnisse. Hier wird geprüft, ob der Quellfile existiert und das Quelle und Ziel nicht identisch sind. Sind diese Dinge OK, kann die eigentliche Arbeit starten und die Hatz nach den falschen Bytes beginnen.

Ein weiterer Höhepunkt der Byteschaufelei findet sich in der Hauptschleife im Hauptprogramm. Als Ende-Kennung sind drei Zustände vorgesehen:

1. Ein Fehler wird bei READF oder WRITEF erkannt
2. Das Fileende wird erkannt,
3. Die NULL für SCRIPSIT File-Ende wird erkannt.

Damit beides richtig zur Geltung kommt, wird ein eifriges Bäumchen-Wechsle-Dich-Spiel auf dem Stack getrieben. Damit die Sache beim ersten gelesenen Byte nicht außer Tritt gerät muß die Dummy-'0' auf den Stack gelegt werden. Damit verständlich wird,

was sich da so abspielt, einige Momente im Leben eines Datenstacks zum mitschreiben: 0 = Dummy, d = Databyte, f = Flag

	READF	ROT	OR	OVER	0=	OR
	0	d	-	d	d	-
	d	f	d	f1	f1	d
TOS->	f	0	f1	d	f2	f3

Wie jeder unschwer erkennen kann, fehlt eine NULL !!!. Die wird von dem WRITEF im zweiten Teil der REPEAT/UNTIL/LOOP Schleife erzeugt (das ist deren Flag über die saubere Abwicklung des in den Zielfile geschriebenen Bytes). Tritt also beim Schreiben oder beim Lesen ein Fehler auf, so bleibt das nicht unbeachtet. Ebenso wie das SCRIPSIT- oder File-Ende nicht übersehen werden kann.

```
:
:
:
0                            ;Dummy '0'
REPEAT
   FILE1 READF               ;hole ein Byte
   ROT OR                    ;(Dummy -'0') OR (flag)
   OVER 0= OR                ;etwa Tscript Fileende ?
UNTIL                        ;nur weiter, kein Fehler
   TEST.ASCII
   FILE2 WRITEF              ;schreibe ein Byte
LOOP
DROP                         ;vernichte Stop-Flag
0 FILE2 WRITEF DROP          ;File-Ende markieren
:
:
:
```

Bei diesem Hürdenlauf der Flags war bis nach Hartmuts Test noch einen Fehler im Programm, den ich nicht bemerkt hatte: Nachdem der File gelesen war, blieb die SCRIPSIT-NULL auf dem Stack. Kein 'Fatal Error', aber unschön. Die hier abgelichtete Version ist jetzt korrekt. An den Testergebnissen ändert sich dadurch nichts ! Da für die Konvertierung eine Tabelle benutzt wird, wäre es natürlich auch denkbar, diese nicht fest im Programm zu verankern, sondern extern zu verwaren und je nach bedarf von Diskette nachzuladen. Dazu müßte das Wort ASCII.TAB das Einlesen der Daten übernehmen und im Wort TABELLE der Zeiger auf die Tabelle im RAM bereitgestellt werden. Im Wort GET.NAMES ist die Abfrage auf den Namen der externen ASCII-Tabelle einzubauen. Im Anschluß folgt nun das vollständige Programm.

HEFT
23
Februar
1988

```
;**======================================**
;**         ASCII Filter-Programm        **
;**                                      **
;**         Vers. 1.0        12.87       **
;**======================================**
;
;
DECLARE
   VARF    FILE1
   VARF    FILE2
   VAR     FILE.NAME1
   VAR     FILE.NAME2
   VAR     COUNT
   VAR     CHARACTERS
DEND


CALL ASCII.TAB              ;Konvertierungs-Tabelle
   B6 7B                    ;Erster Eintrag altes Zeichen,
   B7 7C                    ;...zweiter Eintag neues Zeichen
   B8 7D
   B9 7E
   B1 5B
   B2 5C
   B3 5D
   00 00                    ;Tabellenende. Doppel-Null für den Fall,
CEND                        ;daß jemand unbedingt die Null in eine
                            ;Null konvertieren moechte


CODE TABELLE                ;( -> adr1) Tabellenzeiger auf den Stack
   21 ASCII.TAB             ;LD      HL,ASCII.TAB
   E5                       ;PUSH    HL
CEND


PROGRAM GET.NAMES           ;( -> flag)           ;holt FILENAMEN
   SYSFIELD ?                                     ;RAM-Pointer retten
   ;
   ;Hole die Filenamen
   ;
   14 OUTCHAR                                     ;Cursor ON
   WRITE 'File1: ? ' FILE.NAME1 READ(CH)          ;erster FILE
   WRITE 'File2: ? ' FILE.NAME2 READ(CH)          ;zweiter FILE
   15 OUTCHAR                                     ;Cursor OFF
   CR
   ;
   ;Beide Namen müssen ungleich sein !
   ;
   FILE.NAME1 ? FILE.NAME2 ? =(S) NOT
   IF
      FILE.NAME1 ? FILE1 FILESPEC  ;neue Filenamen...
      FILE.NAME2 ? FILE2 FILESPEC  ;...einsetzen
      OR 0=                        ;ERROR-Flags zusammenfassen
   ELSE
      WRITE '*** ERROR: Input-File = Output-File !' CR
      FALSE                        ;Eingabefehler !
   ENDIF
   SWAP SYSFIELD :=                ;Restore RAM-Pointer
```



```
END


PROGRAM ?CONVERT           ;(char1 adr1 -> char2 adr1 flag)
   OVER OVER ?(B) =        ;TOS = Tabelleneintrag ?
   IF
      COUNT ? INC COUNT :=
      DUP INC ?(B)         ;Ersatzzeichen holen und...
      SWAP ROT DROP        ;...altes Zeichen ersetzen
      TRUE                 ;Setze Erfolgs-Flag
   ELSE
      FALSE                ;Keine Konvertierung
   ENDIF
END


PROGRAM TEST.ASCII                   ;(char1 -> char2)
   DUP DUP                           ;prüfe ob Zeichen in der Tabelle
   177 < SWAP                        ;Diese Bereichseingrenzung ist
   185 > OR NOT                      ;Problemabhängig !!!
   IF
      TABELLE                        ;Tabellen-Zeiger auf den Stack
      REPEAT
         ?CONVERT                    ;Zeichen ersetzen ?
         OVER ?(B) 0= OR             ;Tabellenende ?
      UNTIL
         INC  INC                    ;Nein !, nächster Eintrag
      LOOP
      DROP                           ;Nur das Zeichen bleibt...
   ENDIF                             ;... auf dem Stack
   CHARACTERS ? INC CHARACTERS :=
END


PROGRAM FILTER                       ;>>>>  Hauptprogramm  <<<<
   CLS
   0 COUNT :=
   0 CHARACTERS :=
   WRITE '**====================================**' CR
   WRITE '**             ASCII-Filter           **' CR
   WRITE '**====================================**' CR
   CR
   GET.NAMES                         ;Liefert bei OK TRUE-Flag
   IF                                ;Filespec's OK ?
     FILE1 OPEN(I)
     0=                              ;Quell-File vorhanden ?
     IF
        FILE2 OPEN(O)
        0=
        IF                           ;Abbruch bei FILE-ERROR
           WRITE 'Writing data to file'
           0                         ;Dummy '0'
           REPEAT
              FILE1 READF       ;hole ein Byte
              ROT OR            ;(Dummy -'0') OR (flag)
              OVER 0= OR        ;etwa Tscript Fileende ?
           UNTIL                ;nur weiter, kein Fehler
              TEST.ASCII
              FILE2 WRITEF      ;schreibe ein Byte
```

```
        LOOP
        DROP                  ;vernichte stop-Flag
        0 FILE2 WRITEF DROP   ;File-Ende markieren
        ENDIF
      FILE2 CLOSE
    ENDIF
    FILE1 CLOSE
  ENDIF
  COUNT ? 0 >
  IF
    CR WRITE 'Number of substitutions: ' COUNT PRINT
    CR WRITE 'Total # of characters  : ' CHARACTERS PRINT CR
    CR
  ELSE
    CR WRITE 'No substitutions !'
    CR WRITE 'File length is      :' CHARACTERS PRINT
       WRITE ' Byte' CR CR
  ENDIF
END
```

Die Stadt ist nicht groß genug für uns beide!

UMLAUT.M

Hartmuts Artikel über die Umlaut-Konvertierung in Pascal habt Ihr sicherlich schon gelesen (wenn nicht, tut es am besten jetzt!). Hartmut bat mich, ein paar Zeilen zu der Assembler-Lösung zu sagen, die Ihr hinter diesem Artikel abgedruckt findet. Eigentlich ist das Programm fast "selbstdokumentierend", wenn auch in ganz anderem Sinne als die Pascal-Lösung: in Assembler sind die Kommentare wichtiger als die Befehle.

Der Anfang mit den Abfragen der beiden Filenamen dürfte recht klar sein. Ich will nicht näher darauf eingehen. Nur der Ablauf sei kurz dargestellt: Zuerst wird ein Einleitungstext angezeigt, der hier sehr kurz ausfällt. Dann fragt das Programm nach dem Namen des Files, der die fehlerhaften Umlaute enthält. Falls dieser File nicht existiert, kommt eine Fehlermeldung und die Eingabe muß wiederholt werden, bis der Filename korrekt ist oder mit BREAK abgebrochen wird. Der zweite File soll die korrigierte Version des ersten aufnehmen. Es wird nur einmal gefragt; falls ein Fehler auftritt, bricht das Programm ab (mit einer DOS-Fehlermeldung).

Nun kommt der interessante Teil, den ich mit "***..." markiert habe. Ihn will ich näher erläutern: Zuerst wird ein Byte aus dem ersten File gelesen. Dabei muß DE auf den FCB des ersten Files zeigen! Zurückgegeben wird das gelesene Byte in A und ein Fehlerstatus. Bei Z (Zero-Flag gesetzt) war das Lesen erfolgreich, bei NZ (Zero-Flag gelöscht) ist ein Fehler aufgetreten, wobei jetzt die Nummer des Fehlers in A steht; in diesem Fall wird die Schleife abgebrochen, wobei noch nicht klar ist, ob dieses Ende erzwungen (Disk-Fehler) oder gewollt (Ende von File 1 = ich bin fertig) wurde. Falls das Lesen erfolgreich war, stellen die beiden ComPare-Befehle fest, ob es sich um einen Umlaut handelt. Falls nicht, kann das Byte sofort in den zweiten File geschrieben werden. Falls es sich um einen Umlaut handelt, wird 177 von seinem ASCII-Code abgezogen, so daß der erste Umlaut jetzt zur 0 wird, der zweite zur 1 usw. Diese Zahl gibt an, um welchen Umlaut es sich handelt. Nachdem H und L gesetzt wurden, hole ich durch LD A,(HL) den neuen Umlaut aus einer Tabelle. Dabei zeigt HL genau auf den richtigen Umlaut, weil der Anfang der Tabelle immer bei xx00h liegt. Also steht der erste Umlaut (der zu dem mit der Nummer 0 gehörende) in (xx00h), der zweite (der zu Nummer 1 gehört) in (xx01h) usw. Das Gleiche hätte ich erreicht, wenn ich die Nummer auf den Tabellenanfang tab addieren würde, aber das dauert länger (ein paar Micro-, Nano- oder sonstwas-Sekunden). Zuletzt wird der neue Umlaut (bzw. ein normaler Buchstabe, der kein Umlaut ist) in den zweiten File geschrieben. Dabei ist zu beachten, daß DE diesmal auf den zweiten FCB zeigen muß. Falls beim Schreiben ein Fehler auftritt, bricht das Programm ab und zeigt den Fehler an. Ansonsten (im Normalfall) wird die Schleife hier nie verlassen.

Bevor das Programm endet, werden alle beiden Dateien geschlossen (was bei der Lese-Datei evtl. nicht nötig ist, aber es sieht schöner aus). Je nachdem, ob ein "ordentlicher" Abbruch mit dem "richtigen" Fehler, nämlich "Dateiende erreicht", erfolgt ist oder nicht, wird eine "Alles roger"- oder eine Fehlermeldung (vom DOS) angezeigt.

Gerald Schröder

```
                 :        UMLAUT.M
                 :        letzte Änderung: 8.2.88 von Gerald Schröder

                 :        konvertiert File mit Umlauten

                 :        DOS- und ROM-Routinen:

0013               readbyt  EQU    0013h      ;ein Byte lesen
001B               writbyt  EQU    001bh      ;ein Byte schreiben
05D9               inbuf    EQU    05d9h      ;B Zeichen eingeben
402D               dos      EQU    402dh      ;Abgang ins DOS
4409               derror   EQU    4409h      ;DOS-Error ausgeben
441C               exfil    EQU    441ch      ;Filenamen in FCB
4420               init     EQU    4420h      ;File neu öffnen
4424               open     EQU    4424h      ;exist. File öffnen
4428               close    EQU    4428h      ;File schließen
4467               scrmes   EQU    4467h      ;Meldung ausgeben

                 :        Konstante:

001C               okerr    EQU    1ch        ;"Dateiende gefunden"


5200                        ORG    5200h

5200   210953      start    LD     HL,anfang  ;Anfangsmeldung
5203   CD6744               CALL   scrmes     ;anzeigen

5206   213A53      nochmal  LD     HL,frage1  ;Abfrage 1. Filename
5209   CD6744               CALL   scrmes     ;ausgeben
520C   218F53               LD     HL,buf1    ;Buffer für Eingabe
520F   060F                 LD     B,15       ;Länge 15
5211   CDD905               CALL   inbuf      ;B Zeichen nach HL
5214   DA2D40               JP     C,dos      ;ab, falls BREAK-Taste
5217   11AD53               LD     DE,fcb1    ;Zeiger auf FCB-Buffer
521A   CD1C44               CALL   exfil      ;Name übertragen in FCB
521D   21ED53               LD     HL,sekbuf1 ;Sektor-Buffer
5220   0600                 LD     B,0        ;Sektor-Länge 256
5222   CD2444               CALL   open       ;File öffnen
5225   2808                 JR     Z,zweiter  ;kein Fehler, weiter
5227   214A53               LD     HL,fehler  ;Zeiger auf Fehlermeldung
522A   CD6744               CALL   scrmes     ;ausgeben
522D   18D7                 JR     nochmal    ;nochmal eingeben

522F   216953      zweiter  LD     HL,frage2  ;ebenso für 2. File
5232   CD6744               CALL   scrmes
5235   219E53               LD     HL,buf2
5238   060F                 LD     B,15
523A   CDD905               CALL   inbuf
523D   DA2D40               JP     C,dos
5240   11CD53               LD     DE,fcb2
5243   CD1C44               CALL   exfil
5246   21ED54               LD     HL,sekbuf2
5249   0600                 LD     B,0
524B   CD2044               CALL   init       ;diesen neu schreiben
524E   2034                 JR     NZ,error   ;falls Disk-Fehler

                 : ****************************************************
5250   11AD53      loop     LD     DE,fcb1    ;Zeiger auf 1. FCB
5253   CD1300               CALL   readbyt    ;Byte aus File 1 lesen
5256   2016                 JR     NZ,ende    ;Ende oder Fehler
5258   FEB1                 CP     177        ;Buchstabe < 177?
525A   380A                 JR     C,ok       ;ja, ok, kein Umlaut
525C   FEBA                 CP     186        ;Buchstabe > 185?
525E   3006                 JR     NC,ok      ;ja, ok, kein Umlaut
5260   D6B1                 SUB    177        ;177=>0 usw.
```



```
5262   2653                 LD     H,tab/100h ;MSB der Umw.tabelle
5264   6F                   LD     L,A        ;Code als Tab-Offset
5265   7E                   LD     A,(HL)     ;geändertes Zeichen holen
5266   11CD53      ok       LD     DE,fcb2    ;Zeiger auf 2. FCB
5269   CD1B00               CALL   writbyt    ;Byte in File 2 schreiben
526C   28E2                 JR     Z,loop     ;kein Fehler: nächstes Z.
                 ;**********************************************************

                 :          Status NZ erreicht: irgendein Diskfehler trat auf
                 ;          Bei "Dateiende erreicht" ist das OK, ansonsten
                 :          hat was nicht geklappt!

526E   F5          ende     PUSH   AF         ;Fehlercode retten
526F   11AD53               LD     DE,fcb1    ;File 1
5272   CD2844               CALL   close      ;schließen
5275   11CD53               LD     DE,fcb2    ;ebenso File 2
5278   CD2844               CALL   close
527B   F1                   POP    AF         ;Fehlercode zurück
527C   FE1C                 CP     okerr      ;Dateiende gefunden?
527E   217C53               LD     HL,endmel  ;Zeiger auf OK-Meldung
5281   CA6744               JP     Z,scrmes   ;falls ja: ausgeben und
                                              ;Abgang ins DOS

5284   F6C0        error    OR     0c0h       ;Bit 7 für Fehler-Ausgabe
5286   C30944               JP     derror     ;setzen und Ausgabe u. ab

                 ;----------------------------------------------------------

                 :          Tabelle zur Umwandlung der kritischen Zeichen

0077                        DS     $+100h&0ff00h-$ ;Anfang auf "gerader"
                                              ;Adresse (LSB=00)
5300   5B          tab      DB     91,92,93,180,181,123,124,125,126


                 :          Texte

5309   1C          anfang   DM     1ch,1fh,'UMLAUT.M',0ah,0ah,'Alles klar?'
5320   0A                   DM     0ah,'Sonst ist es auch egal.',0ah,0dh
533A   0A          frage1   DM     0ah,'Welcher File? ',03
534A   0A          fehler   DM     0ah,'So nicht! Gleich noch einmal!',0dh
5369   0A          frage2   DM     0ah,'In welchen File? ',03
537C   0A          endmel   DM     0ah,'Alles roger, bye!',0dh



                 :          Buffer

000F               buf1     DS     15         ;Eingabe Filename 1
000F               buf2     DS     15         ;ebenso Filename 2
0020               fcb1     DS     32         ;FCB File 1
0020               fcb2     DS     32         ;ebenso File 2
0100               sekbuf1  DS     256        ;Sektorbuffer File 1
0100               sekbuf2  DS     256        ;ebenso File 2

5200                        END    start

00000 Fehler
```

CP/M intern - kurz und kompakt

0. Vorwort

1. CCP
   1.1 Wie sieht ein Filename aus?
   1.2 CCP-Kommandos
   1.3 CTRL-Codes

2. Programmierung
   2.1 Zero-Page
   2.2 Parameter-Übergabe
   2.3 Der FCB
   2.4 BDOS-Funktionen
   2.5 BIOS-Funktionen

3. Disk-Parameter-Tabellen
   3.1 DPH
   3.2 DPB

4. Genie IIs-CP/M
   4.1 DPB-Anhängsel
   4.2 Tastatur-Codes

5. Terminals
   5.1 ADM-3A
   5.2 VT-52
   5.3 TeleVideo TV950

Anhang:
   A) Ein Programmbeispiel für BDOS-Aufrufe
   B) Begriffserklärungen
   C) Die Speicherbelegung

0. Vorwort

Nachdem ich angefangen hatte, mich mit CP/M zu beschäftigen, wurmte mich gleich eine Sache ungemein: wie beim alten Newdos brauchte ich aus den dicken Anleitungsbüchern wieder nur ein paar Seiten, die wiederum nicht vollständig oder zu ausführlich waren. Also wollte ich mir eine kurze Zusammenfassung der Sachen schreiben, die ich oft brauchen würde. Daher auch der Titel "Kurz und kompakt". Leider mußte ich feststellen, daß die Kürze das Verständnis erschwerte und es wieder nötig machte, die dicken Wälzer heranzuziehen. Also habe ich einen Kompromiß beschlossen: die Kapitel sind kurz gehalten und mit massenhaft Abkürzungen versehen. Im Anhang dagegen werden einige Begriffe und Themen nochmal ausführlicher erläutert, damit ein Neuling damit besser zurechtkommt. Klassischerweise müßte die Aufteilung anders sein, aber das war nicht der Sinn der Sache.

Die Informationen habe ich mir aus dem "Handbuch zum CP/M" von Digital Research und dem CP/M-Buch von Plate zusammengesucht.

Leider bin ich mir bei vielen Sachen nicht sicher, ob sie stimmen, und wäre sehr froh, wenn das jemand korrigieren würde und Erweiterungen vorschlüge. Vor allem fehlen mir Infos über Terminal-Steuercodes.

Vielen Dank an Hartmut Obermann für das Korrekturlesen und viele Informationen und an Helmut Bernhardt für die CP/M-Fähigkeit meines Rechners.

Gerald Schröder

1. CCP

Der CCP wird direkt unter das BDOS in die TPA geladen. Er kann mit einem laufenden Programm überschrieben werden. Allerdings muß das Programm dann mit einem Warm Boot (JP 0000) beendet werden. Ansonsten kann man den CCP wieder direkt anspringen (wenn die Adresse bekannt ist). Er erwartet das Default-Laufwerk in Register C.

1.1 Wie sieht ein Filename aus?

a) bfile = bestimmter File, z.B. TURBO.COM
b) ufile = unbestimmter File, '?'=beliebiger Buchstabe, '*'=Rest
   beliebig, z.B. TURBO.??? oder TURBO.*

1.2 CCP-Kommandos

'ERA ufile'	: löscht Files
'DIR'	: Inhaltsverzeichnis akt. Laufwerk anzeigen
'DIR lw'	: Inhaltsverzeichnis eines bestimmten Laufwerks anzeigen
'DIR ufile'	: nur bestimmte Files anzeigen
'REN bfile=bfile'	: File umbennen (1. Namen in 2. umändern)
'SAVE n bfile'	: n Sektoren (n dez.) je 100h ab 100h sichern
'TYPE bfile'	: File anzeigen
'USER n'	: auf anderen Userbereich (für alle Laufwerke gültig) umschalten, n<16

1.3 CTRL-Codes (wichtige)

CTRL-C: Warm-Boot (nur am Zeilenanfang)
CTRL-P: Drucker-Listing an/aus
CTRL-R: Zeile nochmal anzeigen
CTRL-S: List-Pause, bis irgendeine Taste
CTRL-Z: Ende einer Aktion, auch: EOF (1ah)

2. Programmierung

2.1 Zero-Page

0000: Ansprung Warm-Boot
0001/2: Zeiger auf Warm-Boot-Routine (=BIOS+3)

0003: I/O-Byte:

Bits: 7/6 LIST, 5/4 PUNCH, 3/2 READER, 1/0 CONSOLE

Zuordnung:

	LIST (LST:)	PUNCH (PUN:)	READER (RDR:)	CONSOLE (CON:)
00	TTY:	TTY:	TTY:	TTY:
01	CRT:	PTP:	PTR:	CRT:
10	LPT:	UP1:	UR1:	BAT:
11	UL1:	UP2:	UR2:	UC1:

0004: Bits 7-4: USER-Nr.; Bits 3-0: Default-Lw.
0005: Ansprung BDOS (-Funktionen)
0006/0007: Zeiger auf BDOS (bis hierhin kann jedes Programm den Speicher benutzen, wenn es mit JP 0000 endet)
005ch-007fh: FCB für Benutzer
0080h-00ffh: Sektor-Buffer für Benutzer

2.2 Parameter-übergabe

übergabe von Parametern beim Aufruf von COM-Programmen
am Bëispiel von: 'turbo a:abc.pas xyz.com'

```
1. FCB: (005ch) ff.: 01,'ABC     PAS',0,0,0,0
2. FCB: (006ch) ff.: 00,'XYZ     COM',0,0,0,0 (muß gerettet werden!)
```

Rest der Zeile: (0080h) ff.: 18,' A:ABC.PAS XYZ.COM'
(vom CCP in Großbuchstaben umgewandelt, (0080h)=Länge)

das Anwender-Programm wird vom CCP geCALLt; es steht ein Stack mit 7 Ebenen zur Vefügung; also lieber eigenen Stack einrichten!

2.3 Der FCB

Format des FCB:

```
   DR F1 F2 .. F8 T1 T2 T3 EX S1 S2 RC D0 .. DN CR R0 R1 R2
 + 00 01 01    08 09 10 11 12 13 14 15 16    31 32 33 34 35
```

Erklärung FCB-Bytes (bzw. -Felder):

```
DR: Lw. für Zugriff: 00=Default-Lw.
                     01=Lw. A
                      ..
                     16=Lw. P
F1..F8: Filename in Großbuchstaben
T1..T3: Extension in Großbuchstaben
        Bit 7 von T1 gesetzt: File kann nur gelesen werden (R/O)
        Bit 7 von T2 gesetzt: SYS-File (bei DIR nicht angezeigt)
EX: akt. Extent-Nr., vom Benutzer auf 0 zu setzen, bei File E/A 0-31
S1: reserviert
S2: reserviert, bei OPEN/SEARCH/MAKE auf 0 setzen (oder gesetzt???)
RC: Record-Nr. im Extent , 0-127
D0..DN: reserviert
CR: akt. Record-Nr. bei seq. File, vom Benutzer auf 0 zu setzen
R0..R2: Record-Nr. bei Random-File, LSB=R0, MSB=R1, Overflow=R2
```

2.4 BDOS-Funktionen

BDOS-Funktionen:
(die Funktionsnr. ist jeweils nach C zu laden; es werden keine Register gerettet; s. auch Anhang A)

```
00: sysres
                System-Reset, wie Warm-Boot (zurück zum CCP, Lw. auf A)
01: bconin      OUT: A=Tasten-Code
02: bconout     IN:  E=auszugebendes Zeichen
03: breader     OUT: A=gelesenes Zeichen
                auf READER-Eingabe warten
04: bpunch      IN:  E=auszugebendes Zeichen
                auf PUNCH ausgeben
05: blist       IN:  E=auszugebendes Zeichen
06: condir      IN: E=0ffh: CONSOLE-Eingabe (Tastatur)
                   <0ffh: CONSOLE-Ausgabe (Bildschirm)
                OUT: (nur bei E=0ffh) A=00: keine Taste
                                       >00: Tastencode
07: getio       OUT: A=I/O-Byte
08: setio       IN:  E=I/O-Byte
09: prstr       IN:  DE=Zeiger auf String, der mit '$' endet
                String-Ausgabe auf den Bildschirm
```



```
10: getstr      IN:  DE=Zeiger auf Buffer
                     (DE)=max. einzugebende Zeichen (1-255)
                OUT: (buffer+1): Anzahl eingebene Zeichen
                     (buffer+2) ff.: eingebene Zeichen
11: bconst      OUT: A=0ffh: Taste gedrückt
                      =00h: keine Taste gedrückt
12: getver      OUT: H=00: CP/M
                      =01: MP/M
                     L=Versions-Nr. (z.B. 20h=2.0, 22h=2.2)
13: resdsk
                Disk zurücksetzen (bei Diskettenwechsel), DMA auf 0080h,
                Laufwerk A anwählen
14: bseldsk     IN:  E=Lw. (00h=A)
                das angewählte Lw. wird Default-Lw.
15: open        IN:  DE=Zeiger auf FCB
                OUT: A=0-3: alles OK
                       0ffh: Fehler
                im Filenamen auch '?' erlaubt
16: close       IN:  DE=Zeiger auf FCB
                OUT: A=Fehlerstatus (s. 15)
17: search1     IN:  DE=Zeiger auf FCB
                OUT: A=Fehlerstatus (s. 15)
                DMA mit entsprechenden DIR-Sektor geladen, File-Eintrag
                bei DMA + A*32
                wenn DR='?' wird das Default-Lw. auf allen USER-Ebenen
                durchsucht
18: searchn     IN:  DE=Zeiger auf FCB
                OUT: A=Fehlerstatus (s. 15)
                nächsten passenden Eintrag suchen, ansonsten siehe 17
19: delete      IN:  DE=Zeiger auf FCB
                OUT: A=Fehlerstatus (0ffh: nicht gefunden)
                Filename darf '?' enthalten, aber DR nicht, alle pas-
                senden Files werden gelöscht
20: readseq     IN:  DE=Zeiger auf FCB
                OUT: A=00: OK
                      >00: Fehler
                128-Byte-Sektor wird gelesen, CR und evtl. EX +1
21: writeseq    IN:  DE=Zeiger auf FCB
                OUT: A=Fehlerstatus (s. 20)
22: make        IN:  DE=Zeiger auf FCB
                OUT: A=0ffh: Disk ist voll
                      =0/1/2/3: OK
                neuen File öffnen
23: rename      IN: DE=Zeiger auf FCB 1 und 2
                OUT: A=0ffh: File nicht gefunden
                      <0ffh: OK
                erste  16 Bytes vom FCB: vorhandener Filename
                zweite 16 Bytes vom FCB: neuer Filename, DR muß 00 sein
24: getlog      OUT: HL=Login-Vektor
                welche Laufwerke wurden bisher angesprochen? (Bit 0 von
                L = Lw. A)
25: getdsk      OUT: A=akt. Default-Lw. (00h=Lw. A)
26: setdma      IN:  DE=DMA-Adresse
27: getalc      OUT: HL=Zeiger auf ALLOCation-Vektor
28: wpdsk
                akt. Lw. schreibschützen (R/O)
29: getro       OUT: HL=R/O-Vektor
                welche Laufwerke sind schreibgeschützt? (Bit 0 von
                L = Lw. A)
```

30: setatt IN: DE=Zeiger auf FCB
OUT: A=Fehler-Status (s. 15)
R/O bzw. SYS eines Files neu setzen im Directory
31: getdpb OUT: HL=Zeiger auf DPB
32: user IN: E=0ffh: hole akt. USER-Nr.
=0-15: USER neu setzen
OUT: (nur bei E=0ffh): A=akt. USER-Nr.
33: readrnd IN: DE=Zeiger auf FCB
OUT: A=00: OK
01: nicht geschriebene Daten gelesen
03: Extent kann nicht geschlossen werden
04: Versuch, ungeschriebenen Extent zu finden
06: hinter Ende der Diskette (R2>0)
bei Fehler 3: nochmal Extent 0 lesen
Record-Nr. wird nicht inkrementiert
34: writrnd IN: DE=Zeiger auf FCB
OUT: Fehler-Status, s. 33
A=5: neuer Extent kann nicht geöffnet werden
(Inhaltsverzeichnis voll)
Record-Nr. wird nicht inkrementiert
35: getsiz IN: DE=Zeiger auf FCB
OUT: R0/R1/R2: File-Größe
R0/R1/R2 werden auf letzte Record-Nr.+1 gesetzt
36: setrnd IN: DE=Zeiger auf FCB
OUT: R0/R1/R2
R0/R1/R2 bei einem File setzen, auf den bis jetzt seq. zugegriffen wurde; nun Random-Zugriff möglich
37: resdrv IN: DE=Lw.-Vektor
OUT: A=00
Laufwerke zurücksetzen, wenn Bits im Vektor gesetzt
40: write0 IN: DE=Zeiger auf FCB
OUT: Fehlerstatus wie bei 34
Block wird vor dem Schreiben mit Nullen gefüllt, wenn er vorher unbelegt war, ansonsten wie 34

2.5 BIOS-Funktionen

so erhält man die Startadresse: (0001/2)=Zeiger auf BIOS+3

folgende Tabelle: jeweils BIOS (Basisadresse) + Offset

+00: boot Kaltstart
+03: wboot Warmstart
+06: const Tastatur-Status, OUT: A=0ffh: Taste grdückt
+09: conin auf Tastendruck warten, OUT: A=Tastencode
+12: conout auf den Bildschirm ausgeben, IN: C=Zeichen
+15: list auf den Drucker ausgeben, IN: C=Zeichen
+18: punch auf den PUNCH ausgeben, IN: C=Zeichen
+21: reader auf Eingabe vom READER warten, OUT: A=Zeichen (1ah=EOF)
+24: home angewähltes Laufwerk auf Track 0
+27: seldsk Laufwerk anwählen, IN: C=Lw. (00h=A); OUT: HL=Adresse der Sektor-Übersetzungs-Tabelle, HL=0000: Fehler
+30: settrk Track anwählen, IN: BC=Track-Nr.
+33: setsec Sektor anwählen, IN: BC=Sektor-Nr.
+36: setdma DMA-Adresse setzen, IN: BC=DMA-Adresse
+39: read Sektor lesen, OUT: A=00: OK, A=01: Fehler
+42: write Sektor schreiben, IN: C=0: normales Schreiben; C=1: DIR-Sektor schreiben (sofort ausgeführt); C=2: 1. Sektor eines neuen Blocks; OUT: A=00: OK, A=01: Fehler
+45: listst Status des Druckers abfragen, OUT: A=0ffh: READY
+48: sectrn Übersetzung der Sektor-Nr., IN: BC=Sektor-Nr., DE=Adresse der Übersetzungs-Tabelle; OUT: HL=übersetzte Sektor-Nr.

3. Disk-Parameter-Tabellen

3.1 DPH (Disk Parameter Header)

Format:	XLT	0000	0000	0000	DIRBUF	DPB	CSV	ALV
Bytes: +	0	2	4	6	8	10	12	14

Bedeutung:
XLT: Zeiger auf Sektor-Übersetzungs-Tabelle
0000: für BDOS reserviert
DIRBUF: Zeiger auf einen 128-Byte-Buffer für Directory-Zugriffe kann für alle DPHs gleich sein
DPB: Zeiger auf den DPB für dieses Lw.
CSV: Zeiger auf einen Prüfsummenblock, mit dem festgestellt werden kann, ob die Diskette gewechselt wurde; muß für jeden DPH ein anderer sein; Länge: s. DPB
ALV: Zeiger auf einen Block, in dem BDOS die Diskettenbelegung festhält (Allocation-Vektor); muß für jeden DPH ein anderer sein; Länge: (DSM/8)+1 (DSM: s. DPB; evtl. auch: (DSM+1)/8)

3.2 DPB (Disk Paramater Block)

Format:	SPT	BSH	BLM	EXM	DSM	DRM	AL0	AL1	CKS	OFF
Bytes: +	0	+2	+3	+4	+5	+7	+9	+10	+11	+13

Achtung: indirekt festgelegt: BLS (Blockgröße in Bytes): 1024 * 2 hoch x, wobei x von 0 bis 4 gehen darf (BLS von 1024 bis 16384)

Bedeutung:
SPT: Anzahl 128-Byte-Sektoren per Track
BSH: x+3 (x aus BLS, s. oben), d.h.: wenn BSH=3, ist BLS=1024 usw.
BLM: 2 hoch BSH - 1, d.h. wenn BSH=3, ist BLM=7 usw.
EXM: hängt von DSM ab; wenn DSM<256: EXM=2 hoch x - 1, d.h.: x=0: EXM=0 usw.; wenn DSM>255: EXM=2 hoch (x-1) - 1; (x=1: EXM=0)
DSM: größte Blocknummer (in BLS gemessen); (DSM+1)*BLS ergibt die totale Speicherkapazität der Diskette in Bytes, ohne die System-tracks; also: DSM=(Tracks-OFF)*SPT*128/BLS - 1
DRM: Anzahl der Directory-Einträge-1 (d.h.: DRM=127: 128 Dir-Eintr.)
AL0/AL1:

Bytes	AL0								AL1							
Bits	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
belegt:	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15

es sind (DSM+1)*32/BLS Bits zu belegen, angefangen von AL0, Bit 7, über AL0, Bit 0, und AL1, Bit 7, bis AL1, Bit 0; bei BLS=1024 und DRM=128 sind AL0=0f0h und AL1=00h (4 Bits belegt)
CKS: Länge des Prüfblockes CSV (s. DPH); CKS=(DRM+1)/4; nur bei Dis-ketten, nicht bei festen Speichern (RAM-Disk oder Festplatte: CKS=0)
OFF: Anzahl der System-Tracks; werden am Disk-Anfang ignoriert

4. Genie IIs-CP/M: Besonderheiten

Beim Genie IIs-CP/M handelt es sich um eine CP/M-Version, die stark an das Montezuma-CP/M für das Model IV angelehnt wurde. Allerdings wurden nicht alle Features übernommen. Hier nur die für die Programmierung wichtigen Einzelheiten.

4.1 DPB-Anhängsel

DPB+15: Zeiger auf DCB für das Laufwerk (WORD)
DPB+17: Hardware-Byte

Der DCB enthält nur drei Byte-Einträge: Select-Bits für das Laufwerk, momentaner Track (Offh=keiner), momentane Select-Maske (mit Seitenanwahl).

Das Hardware-Byte sorgt für korrektes Lesen von Fremd-Formaten. Die Bedeutung der Bits:

- 7-5: noch unbelegt
- 4: Schreibdichte (0=einfache, 1=doppelte)
- 3: Doppelstep (0=nein, 1=ja, für 40er-Disks in 80er-Laufwerken)
- 2: Track-Zählung (0=gleiche Track-Nr. auf Vorder- und Rückseite, 1=ungerade Track-Nummern auf der Rückseite)
- 1/0: phys. Sektorgröße (00=128, 01=256, 10=512, 11=1024 Byte je Sektor)

4.2 Tastatur-Codes

Beim JIs übernimmt die Taste "P1" die Funktion der Control-Taste. Shift P1 stellt Groß-/Kleinschreibung um. Die anderen Funktionstasten liefern folgende Codes:

Taste	entspricht ohne Shift	mit Shift
Hochpfeil	CTRL-K (0bh)	CTRL-Ä (ESC, 1bh)
Senkr.pfeil	CTRL-J (LF, 0ah)	CTRL-Z (EOF, 1ah)
ENTER	CTRL-M (CR, 0dh)	CTRL-M (CR, 0dh)
EOF	7fh, DEL	5fh, " "
Linkspfeil	CTRL-H (BS, 08h)	CTRL-X (18h)
Rechtspfeil	CTRL-I (TAB, 09h)	CTRL-Y (TAB, 19h)
CLEAR	CTRL-Q (11h)	CTRL-R (12h)
BREAK	CTRL-C (03h)	CTRL-C (03h)
ß	7eh, "ß"	5eh, "^"
P2	-	-

5. Terminals

CP/M war ursprünglich ein Terminal-Betriebssystem, d.h. es lief auf einem Hauptrechner, an den ein oder mehrere Terminal(s) angeschlossen war(en). Je nach Terminal wurden verschiedene Tastatur-Codes geliefert und (was wichtiger ist) verschiedene Reaktionen auf Bildschirmsteuer-Codes bei der Ausgabe verursacht. Das macht es für Programmierer schwer, Programme zu schreiben, die auf jedem CP/M-Rechner lauffähig sind (z.B. mit welchem Code wird der Bildschirm gelöscht?). Daher bieten die meisten Programme eine Anpassung an verschiedene Terminals, z.B. durch Installations-Programme wie TINST bei Turbo-Pascal. Bei heutigen Rechnern ist die Bildschirmansteuerung und die Tastatur zwar meist direkt integriert, aber es werden immer noch verschiedene Terminals simuliert, um CP/M gerecht zu werden. Deshalb muß auch heute der Programmierer die Terminal-Codes kennen, wenigstens die seines Rechners. Deshalb hier eine Auflistung gängiger Terminal-Codes. Ergänzung und Korrektur dringend erwünscht.

5.1 ADM-3A

Name	Code	Funktion
BELL	7	Ton ausgeben
BS	8	Backspace
TAB	9	zur nächsten Tabulator-Pos.
LF	10	in die nächste Zeile
VT	11	in die vorherige Zeile
CRT	12	ein Zeichen nach rechts
CR	13	an den Zeilenanfang
INVOFF	14	Invers aus
INVON	15	Invers an
EOL	21	lösche bis Zeilenende
INVTOG	22	Invers umschalten
EOS	25	lösche bis Bildschirmende
CLS	26	lösche Bildschirm u. HOME
ESC	27	ESC-Sequenz einleiten (s.u.)
HOME	30	Cursor an Bildschirmanfang

Mit der Escape-Sequenz kann der Cursor beliebig auf dem Bildchirm positioniert werden. Nach dem ESC-Code (27) muß ein "=" (61) zum Bildschirm geschickt werden; dieses Zeichen wird nicht ausgegeben. Dann folgen zwei Angaben: zuerst die Zeile und dann die Spalte, in die der Cursor positioniert werden soll. Dabei bedeutet jeweils ein Blank (32) den Wert 0. Also wird durch die Sequenz "27,61,35,40" der Cursor in der 4. Zeile (Zeile 3 von Null an gezählt) an der 9. Stelle (Spalte 8 von Null an gezählt) ausgegeben.

5.2 VT-52

Alle Steuercodes werden durch ein ESC (27) eingeleitet.

Code	Funktion
'A'	Cursor eine Zeile hoch
'B'	Cursor eine Zeile runter
'C'	Cursor rechts
'D'	Cursor links
'E'	Bildschirm löschen & HOME
'H'	HOME
'I'	Cursor hoch; scrollen, wenn nötig
'J'	Bildschirm ab Cursor löschen
'K'	Zeile ab Cursor löschen
'L'	Zeile einfügen
'M'	Zeile löschen (Rest hochscrollen)
'Y' (y-32) (x-32)	Cursor an Position x/y setzen
'b' Farbe	Schriftfarbe neu wählen
'c' Farbe	Hintergrundfarbe neu wählen
'd'	Bildschirm bis Cursor löschen
'e'	Cursor einschalten
'f'	Cursor ausschalten
'j'	Cursorposition speichern
'k'	Cursor auf gespeicherte Position setzen
'l'	Zeile löschen
'o'	Zeile bis Cursor löschen
'p'	Invers ein
'q'	Invers aus
'v'	Überlauf an (Cursor geht über Zeilenende)
'w'	Überlauf aus (Cursor bleibt am Zeilenende stehen)

(aus: Data-Becker-Führer für Atari-ST)

5.3 TeleVideo TV950

ASCII	Hex	Funktion
^G	07	Ton ausgeben
^H	08	Backspace
^I	09	TAB
^J	0a	Cursor runter
^K	0b	Cursor hoch
^L	0c	Cursor rechts
^M	0d	CR
^V	16	wie ^J, aber ohne Scroll
^Z	1a	Bildschirm löschen & HOME
^^	1e	HOME
ESC '('	1b 28	normale Leuchtstärke
ESC ')'	1b 29	halbe "
ESC '*'	1b 2a	CLS & HOME (auch '+', ',', ':', ';')
ESC '.' n	1b 2e n	Cursor-Attribute setzen, n= '0' (30h): Cursor aus '1' (31h): Cursor als Block (blinkend) '2' (32h): " (nicht bl.) '3' (33h): Cursor als Unterstrich (blinkend) '4' (34h): " (nicht bl.)
ESC '=' s z	1b 3d s z	Cursor auf Spalte s-20h, Zeile z-20h
ESC '?'	1b 3f	Cursor-Pos. abfragen
ESC 'D' x	1b 44 x	Modus setzen: 'L'=lokal, 'H'=halb-duplex, alles andere: voll-duplex
ESC 'E'	1b 45	Zeile einfügen
ESC 'G' x	1b 47 x	Zeichen-Attribute setzen (x-30h), Bits: 6: Grafik-Modus (Zeichen als Grafik) 5: doppelt hoch 4: doppelt breit 3: unterstrichen 2: invers 1: blinkend 0: unsichtbar
ESC 'I'	1b 49	zurück zum letzten Tabulator
ESC 'Q'	1b 51	Zeichen bei Cursor einfügen
ESC 'R'	1b 52	Zeile löschen
ESC 'T'	1b 54	Zeile ab Cursor löschen
ESC 'U'	1b 55	Control-Modus ein (alle Steuerzeichen als Buchstaben mit halber Leuchtstärke)
ESC 'W'	1b 57	Zeichen löschen
ESC 'Y'	1b 59	bis Bildschirmende löschen
ESC 'b'	1b 62	Bildschirm invertieren
ESC 'd'	1b 64	Bildschirm normal
ESC 't'	1b 74	Zeile ab Cursor löschen
ESC 'y'	1b 79	bis Bildschirmende löschen
ESC 'z' n	1b 7a n	Zeichensatz wählen von '0' (30h): USA über Frankreich, Deutschland, England, Dänemark, Schweden, Italien bis '7' (37h): Spanien
ESC 'ä' ...	1b 7b ...	Schnittstelle initialisieren 1. Parameter: Baudrate ('?'=19200, '0' oder '>'= 9600, '<'=4800, ':'=2400, '8'=1200, '7'=600, '6'=300, '5'=150, '3'=110) 2. Parameter: '0'=ein, '1'= zwei Stopbits 3. Parameter: Parity ('0'=aus, '1'=ungerade, '3'=gerade) 4. Parameter: '0'=8-, '1'=7-Bit-Wortlänge

(aus: c't-Artikel über c't-Text-Terminal)

Anhänge

A) Beispielprogramm für BDOS-Aufrufe

```
wboot   EQU     0000h           ;Warm-Boot-Ansprung
bdos    EQU     0005h           ;BDOS-Aufruf-Ansprung
bconin  EQU     1               ;hole Zeichen
bconout EQU     2               ;gib Zeichen aus
prstr   EQU     9               ;gib String aus
getstr  EQU     10              ;hole String

start   LD      DE,string       ;Zeiger auf String
        LD      C,prstr         ;BDOS-Fkt.-Nr.
        CALL    bdos            ;String ausgeben
        LD      C,bconin        ;BDOS-Fkt.-Nr.
        CALL    bdos            ;Zeichen holen
        LD      E,A             ;Zeichen nach E
        LD      C,bconout       ;BDOS-Fkt.-Nr.
        CALL    bdos            ;Zeichen ausgeben
        LD      DE,buffer       ;Zeiger auf Buffer
        LD      C,getstr        ;BDOS-Fkt.-Nr.
        CALL    bdos
        LD      A,(buffer+1)    ;Länge holen
        OR      A               ;=0?
        JR      Z,ab            ;ja, ab
        LD      L,A             ;Länge nach HL
        LD      H,0
        LD      DE,buffer+2     ;Zeiger setzen
        ADD     HL,DE           ;Zeiger auf Buffer-Ende
        LD      (HL),'$'        ;Ende-Zeichen setzen
        LD      C,prstr         ;BDOS-Fkt.-Nr.
        CALL    bdos            ;Eingabe nochmal anzeigen
ab      JP      wboot           ;Abgang
;oder:  RET                     ;direkt zum CCP

string  DM      0dh,0ah,'Dies ist ein Test','$'
buffer  DB      5               ;Länge des Buffers
        DB      0               ;tatsächlich eingegeben
        DS      6               ;Buffer für Zeichen
```

B) Begriffserklärungen

In der Anleitung habe ich einige Begriffe und Abkürzungen verwendet, die nicht jedermann geläufig sein könnten. Deshalb hier einige Erläuterungen. Außerdem habe ich noch einige Begriffe aufgenommen, die im Zusammenhang mit CP/M (bzw. Computern allgemein) verwendet werden. Viele Informationen stammen aus der englischen Ausgabe des "Handbuch zum CP/M" von Digital Research.

HEFT 23 Februar 1988

BAT:, Batch — Batch nennt man das Hintereinanderablaufen von mehreren Programmen, ohne daß eine Person eingreift, wobei die Aktionen (Ein-/Ausgaben) auf einem Drucker protokolliert werden; BAT: bezeichnet im CP/M einen Zustand, in dem alle Eingaben von Reader kommen und die Ausgaben auf List gehen; ->I/O-Byte

BDOS — Basic Disk Operating System; das BDOS ist der Teil des Betriebssystems CP/M, der die Disketten-Ein-/Ausgabe für alle CP/M-Rechner standardisiert; es ist deshalb bei allen CP/M-Rechnern gleich und wird von Programmierern intensiv genutzt; das BDOS benutzt das BIOS; -> Anhang C

BIOS Basic Input/Output System; das BIOS ist der "unterste" Teil von CP/M; es regelt die Ein-/Ausgabe speziell für den Rechner, auf dem ein CP/M läuft; es muß für jeden Rechner neu programmiert werden; auf das BIOS sollte man möglichst nicht direkt zugreifen; -> Anhang C

Block eine Organisationsgröße auf der Diskette; die Größe wird im DPB festgelegt und kann 1K, 2K, 4K, 8K oder 16K betragen

Blocking ein Verfahren, das hilft, auf Disketten schneller zuzugreifen; da CP/M nur 128-Byte-Sektoren kennt, die Disketten-Systeme aber oft größere Sektoren (256, 512, 1024 Byte) unterstützen, werden mehrere logische Sektoren in einen physikalischen Sektor gepackt; dieses Packen wird Blocking genannt; -> Deblocking

CCP Console Command Processor; dieser Teil des Betriebssystems ist eigentlich ein Programm, das nur dazu dient, dem Benutzer eine Benutzeroberfläche zur Verfügung zu stellen, mit der er leicht die Disketten verwalten kann; es interpretiert die Benutzerkommandos (ERA, DIR usw.) und setzt sie in konkrete Aktionen um; der CCP ruft Anwenderprogramme auf und diese können, nachdem sie abgelaufen sind, in den CCP zurückkehren, falls sie ihn nicht überschrieben haben; falls sie ihn überschrieben haben, müssen sie mit einem Warm Boot enden, der den CCP neu lädt; es gibt verschiedene CCP-Versionen, die alle das Merkmal haben, daß sie direkt unter dem BDOS im Speicher stehen und laufen; -> Anhang C

Checksum -> Prüfsumme

CLS Clear Screen; den Bildschirm löschen

Cold Boot Kaltstart heißt, daß das Gerät noch nicht "warm" ist bzw. erst gerade eingeschaltet wurde und jetzt erstmal die grundsätzlichen Initialisierungen durchzuführen sind; normalerweise sollen danach nur noch Warmstarts (Warm Boot) durchgeführt werden; wenn Sie RESET geben, entspricht dies einem Kaltstart

COM Extension (z.B. TURBO.COM) von direkt aufrufbaren Programmen, d.h. man muß in der Benutzeroberfläche (CCP) nur den Namen des Programms eingeben, um es zu starten; COMmand-Programme werden immer nach 100h geladen und dort gestartet

Console, CON: Tastatur und Bildschirm in modernen CP/M-Systemen, Teletype (Fernschreiber) oder sonstwas bei älteren; über die Console werden Eingaben bzw. Befehle geholt und Ausgaben gemacht; über CON: kann die Ein-/Ausgabe auf andere Geräte umgeleitet werden; -> I/O-Byte

CP/M Control Program for Microcomputers; unser Thema

CR Carriage Return; an den Anfang der Zeile gehen; beim Newdos gleichzeitig auch "in die nächste Zeile", was totaler Quatsch ist (aber sehr effizient)

CRT: Cathode Ray Tube; ein Bildschirm, wie ihn heute jeder benutzt

DCB Device Control Block; ein Datenblock, der direkt die Eigenschaften eines physikalischen Geräts beschreibt bzw. beeinflußt; unter dem DPB anzusiedeln und oft auch für Tastatur etc. vorhanden; nur in bestimmten BIOS-Versionen vorkommend

Deblocking das Umgekehrte des Blocking: aus einem großen physikalischen Sektor den vom CP/M angeforderten 128-Byte-Sektor raussuchen; wird im BIOS erledigt

Default bezeichnet einen Wert, der angenommen wird, falls nichts anderes an-/eingegeben wurde; z.B. ist das Default-Laufwerk bei Diskettenzugriffen (wie bei "DIR") immer das eingeloggte Laufwerk, dessen Kennung als Prompt vom CCP ausgegeben wird ("A>")

DMA Direct Memory Access; im CP/M die Adresse, an der ein 128 Byte langer Buffer existiert, in den Daten von der Diskette eingelesen bzw. von dem Daten auf die Diskette geschrieben werden; im anderen Zusammenhang ein Baustein, der LDIR-ähnliche Befehle über den 64K-Speicher hinaus ausführt

DPB Disk Parameter Block; der DPB gibt die Charakteristika eines bestimmten Speichermediums (z.B. Diskettenlaufwerk) wieder; falls mehrere Diskettenlaufwerke gleichen Typs in einem System vorhanden sind, muß für alle zusammen nur ein DPB existieren, auf den dann mehrere DPHs zeigen; da die DPB-Einträge nicht für alle Formate ausreichende Informationen enthalten, werden sie je nach CP/M-BIOS-Version durch zusätzliche Einträge ergänzt, die leider keinerlei Normung unterliegen

DPH Disk Parameter Header; eine Datenstruktur, die über dem DPB angesiedelt ist (d.h. einen Zeiger auf einen DPB enthält); der DPH besteht u.a. aus Zeigern auf verschiedene Buffer, die das BDOS benutzt; für jedes logische Laufwerk im System (A:, B: usw.) muß ein eigener DPH existieren

EOF End Of File; ein reservierter Steuer-Code (1ah), der anzeigt, daß ein ASCII-File hier endet; oft auch die Tastenkombination Ctrl-Z oder Fehlermeldung

EOL End Of Line; Ende der Zeile; wie EOS verwendet

EOS End Of Screen; Ende des Bildschirms; oft im Zusammenhang "to EOS", z.B. bei "löschen bis Bildschirmende"

ESC Escape; besonder Taste bzw. besonderer Steuer-Code (1bh); wird benutzt, um eine Folge von Zeichen anzukündigen, die zur Steuerung eines Gerätes (Bildschirm, Drucker) dienen und nicht ausgegeben werden sollen

Extent Directory-Eintrag aus 32 Byte; Byte 0-15: Angaben zum File (Name, User-Bereich, Extension usw.); Byte 16-31: vom File belegte Blöcke (Nummern je 2 Byte); falls mehr als 8 Blöcke belegt werden, wird ein zweiter Directory-Eintrag für den File angelegt; nach Digital Research "16 K aufeinanderfolgende Byte in einem File"

FCB File Control Block; Daten-Block, über den der Programmierer auf einen File zugreift; im FCB stehen u.a. Name, Extension und Laufwerk, momentane Record-Nr. usw.

Flag Flagge; ein Bit, das etwas anzeigt, z.B. "der eben berechnete Wert war Null"; -> Overflow

HOME nach Hause (telefonieren, s. ET); Cursor in die linke obere Ecke des Bildschirms setzen bzw. Schreib-/Lesekopf des Diskettenlaufwerks über Track 0 positionieren

Interleave -> Skew-Faktor

I/O-Byte Input/Output-Byte; ein Byte in der Zero-Page, mit dem logische Geräte auf verschiedene physikalische umgeleitet werden können, falls dies Ihr CP/M-BIOS unterstützt; z.B. kann das logische Gerät CON: (Console) die Eingaben von unterschiedlichen physikalischen Geräten erhalten; läßt sich am besten per STAT ändern

K Kilobyte; 1024 Byte

Kaltstart -> Cold Boot

LF Linefeed; eine Zeile weiter gehen (Cursor oder Druckkopf); meist nach CR zu finden, um einen Zeilenvorschub zu gewährleisten

LIST, LST: das Gerät LIST bzw. LST: ist eines der logischen Geräte, die CP/M kennt und ist normalerweise ein Drucker

loggen meist: ein-/ausloggen; im CP/M: eine neu eingelegte Diskette zum ersten Mal ansprechen bzw. sie dem System bekanntmachen; bei Großrechnern: die Eingabe von Benutzername und -paßwort, bevor man anfangen darf, zu arbeiten

logisch alles, was nicht physikalisch ist; für das CP/M existieren nur logische Geräte, die je nachdem unterschiedlichen physikalischen Geräten zugeordnet sein können; eine Console-Ausgabe, die normalerweise auf einen Bildschirm gelangen würde, kann so auf einem Drucker erscheinen; außerdem: logische Sektoren sind Sektoren mit 128 Byte, die CP/M als einzige kennt; das Speichermedium "dahinter" (z.B. ein Laufwerk) kann ruhig mit größeren Sektoren arbeiten

LPT: Line Printer; das physikalische Gerät, das meist LST: zugeordnet wird

LSB Lowest Signifikant Byte; die unteren 8 Bit eines 16-Bit-Wortes; manchmal auch: Lowest Signifikant Bit, Bit 0 eines Bytes

MP/M Multi-Programming Monitor Control Program; der große Bruder des CP/M; für den Betrieb mehrerer Terminals an einem Haupt-Rechner

MSB Most Signifikant Byte; obere 8 Bit in einem 16-Bit-Wort; manchmal: Most Signifikant Bit, Bit 7 eines Bytes

Nibble obere oder untere 4 Bit in einem Byte

Offset fester Wert, der zu jeder Variable (z.B. Adresse oder Track-Nr.) addiert wird

Overflow überlauf; falls eine Rechnung oder ein Wert eine gesetzte Grenze überschreitet; meist durch ein Flag angezeigt

Page Seite; 256 Byte im Speicher, die allein durch Änderung eines Bytes, des LSB, adressiert werden können (00-ffh)

PTP:, PTR: irgendwelche physikalischen Geräte; P=Punch, R=Reader; -> I/O-Byte

Prüfsumme eine Quersumme oder ähnliches von mehreren Bytes; wird beim CP/M vom Directory gebildet, um festzustellen, ob die Diskette gewechselt wurde

Punch, PUN: Lochkartenstanzer; früher gebräuchliches Gerät aus der Steinzeit der Rechenanlagen; im CP/M ein logisches Gerät, an dem z.B. physikalisch eine serielle Schnittstelle (mit Modem oder so) hängen kann; -> I/O-Byte

Random Zufall; Random-Files sind Daten-Files, bei denen man die Datensätze (Records) direkt adressieren (lesen/schreiben) kann; -> sequentiell

Reader, RDR: Lochkartenleser; s. Steinzeit der Rechenanlagen; logisches Gerät des CP/M; zum Lesen einer seriellen Schnittstelle nicht schlecht; -> I/O-Byte

Record Diskettenorganisationseinheit; je nachdem mal (physikalisch) ein Sektor von 128 Byte oder (logisch) ein vom Benutzer definierter Datensatz anderer Länge

R/W, R/O Read/Write, Read/Only; Attribut eines Files oder einer Diskette; je nachdem darf nur gelesen oder auch geschrieben werden; -> SYS

Sector Translation Table Sektor-übersetzungs-Tabelle; oft XLT abgekürzt; eine Tabelle, die logische Sektornummern in phsyikalische übersetzt; wird benutzt, um die Drehung der Disketten auszunutzen beim Lesen von hintereinanderliegenden Sektoren; dabei heißt "logisch" hier manchmal auch, daß es sich schon um physikalische (256, 512 ... Byte-) Sektoren handelt; -> Skew-Faktor

Select hier: Anwahl(-Bits); eine Bit-Maske, in der 1 Bit gesetzt ist, um ein bestimmtes Laufwerk anzuwählen; manchmal wird noch ein zweites Bit gesetzt, um bei diesem Laufwerk die Rückseite anzuwählen; unterste Ebene des Disketten-Zugriffs; allein dem BIOS überlassen

sequentiell hintereinander; Zugriffsform auf Files, bei der nur der jeweils nächste Datensatz (Record) gelesen werden kann; -> Random

Skew-Faktor Konstante, die auf die zuletzt adressierte physikalische Sektor-Nummer addiert wird, um den nächsten logischen Sektor zu erhalten; mit diesem Faktor wird die Sector Translation Table aufgebaut, um den Diskettenzugriff zu beschleunigen

String Ansammlung von Zeichen; beim CP/M komischerweise immer mit einem "$" abgeschlossen (Zusammenhang mit Turing-Maschine?)

SYS Attribut von Files, die unsichtbar und nur für den system-internen Gebrauch bestimmt sind; SYS-Files auf User 0 können nach Digital-Research-Informationen von jeder User-Nummer aus aufgerufen werden; -> R/W, R/O

System-Tracks eine bestimmte Anzahl Tracks (oft 2) am Anfang einer Diskette, auf denen das CP/M (Boot-Sektor, BIOS, BDOS, CCP) steht, das beim Booten von dort geladen wird; diese Tracks werden nicht als Diskettenplatz gerechnet, sondern per Offset übersprungen; dieser Offset kann auch benutzt werden, um aus einer physikalischen Diskette zwei logische zu machen (eine aus den ersten 40 Tracks und eine mit Offset 40 aus den oberen 40 Tracks oder so)

TAB Tabulator; ein Steuercode (09), der Cursor oder Druckkopf veranlaßt, zur nächsten durch 8 teilbaren Position zu springen

TPA Transient Program Area; Bereich von 100h bis zum Anfang des BDOS, in den die Anwenderprogramme geladen und ausgeführt werden; der CCP steht auch in diesem Bereich; -> Anhang C

TTY: Teletype; Fernschreiber; uraltes Gerät, das als Console zur Ein-/Ausgabe benutzt wurde

UC:, UL:, UP:, UR: physikaliche Geräte, die "U"ser-definiert sind; nicht mit "USER-Bereich" zu verwechseln!; C=Console; L=List; P=Punch; R=Reader

User Benutzer; hier: es gibt im Directory verschiedene Bereiche (0-15), die jeweils nur zugänglich sind, wenn vorher der entsprechende Userbereich (z.B. per Kommando "USER 1") angewählt wurde; zur Benutzung bei Mehrplatzsystemen bzw. als Urform des Sub-Directory gedacht

Warm Boot soll heißen: jetzt ist das Gerät schon warm (länger am Laufen) und muß nicht mehr voll initialisiert werden; die meisten Programme enden mit einem Warm Boot, der dann CCP und evtl. auch BDOS lädt und den CCP startet

Warmstart -> Warm Boot

Wildcard ein Platzhalter in einem String (z.B. Filenamen) für einen beliebigen Buchstaben oder Teil; im CP/M "?" und "*"

Word, Wort 16-Bit-Datum bzw. "Informationseinheit aus 16 Bit"

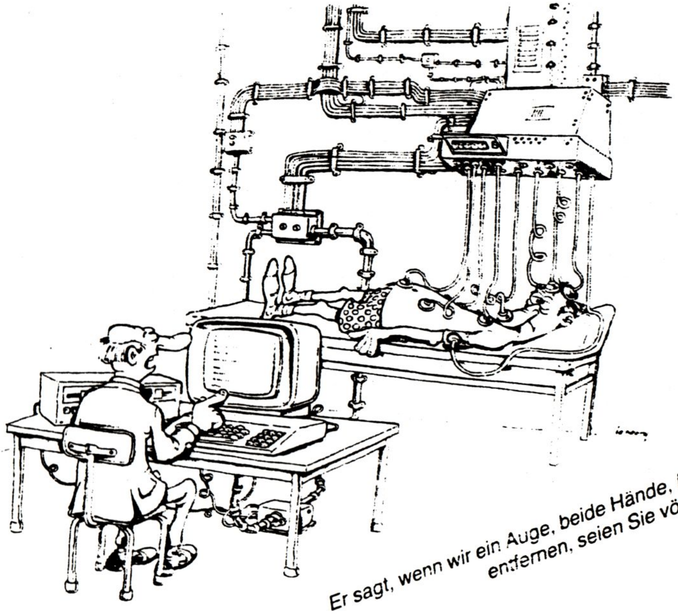
Zero-Page Seite 0; die ersten 100h im Speicherbereich; ursprünglich von Rechnern stammend, bei denen dieser Bereich leichter oder schneller adressieren läßt (6502); im CP/M als Schnittstelle von Anwenderprogramm zu BDOS benutzt; -> Anhang C

C) Der Speicheraufbau

Adresse	Bereich
0000h	Zero-Page
0100h	TPA
ccpbase	(CCP)
bdosbase	BDOS
biosbase	BIOS
ffffh	

Zero-Page

```
0000h          Warm Boot
0003h          I/O-Byte
0004h          User/Default-Lw.
0005h          BDOS-Ansprung
0008h-0027h RST-Vektoren 1-5 (unbenutzt)
0030h-0037h RST 6 (reserviert)
0038h-003ah RST 7 (bei Debug über DDT oder SID: Breakpoint-Entry)
003bh-003fh reserviert (unbenutzt)
0040h-004fh für BIOS reserviert (je nach BIOS-Version benutzt)
0050h-005bh reserviert (unbenutzt)
005ch-006bh 1. Filename bei Aufruf "*.COM file1 file2";
            bis 007fh: FCB zur freien Benutzung
006ch          2. Filename              - " -
0080h          Default-DMA-Buffer
```



Er sagt, wenn wir ein Auge, beide Hände, beide Füße und beide Ohren entfernen, seien Sie völlig normal.

Mein Programm sagt, wir werden hier nie Öl entdecken . . .

d B a s e I I

Eine kurze Einleitung in das Datenbanksystem

v. Harald Mand

Folge II :

4. Summieren von Feldern :

Mit dem Befehl ". S U M " lassen sich nummerische Datenfelder von Datensätzen über mehrere Sätze hinweg aufsummieren.

Beispiel : Berechnung der Summe aller Einkaufspreise

```
USE INVENTUR      =====> Eröffnung d. Datenbank
SUM EK:PREIS      =====> Aufsummierung aller
                         Datensätze
```

* Bis zu 5 Datenfelder bezw. Ausdrücke lassen sich jeweils Satz für Satz addieren und das Ergebnis getrennt in jeweils einer Speichervariablen ablegen.

  Beispiel : SUM MENGE, EK:Preis, VK:PREIS, DIFFERENZ, B:PREIS

  Speicherinhalte werden mit Hilfe des Befehls

  " D I S P L A Y   M E M O R Y " angezeigt.

5. Suchen eines Datensatzes :

Es lassen sich bestimmte Sätze mit Hilfe der folgenden Befehle auffinden :

" F I N D "

Mit diesem Befehl läßt sich ein bestimmter Satz in einer indizierten Datenbank auffinden.

* Der "FIND"-Befehl läßt sich nur anwenden, wenn der Schlüssel (z.B. Name), nach dem man suchen will, aktueller Hauptschlüssel der aktivierten Datenbank ist.

  Es genügt ggf. nur den Anfangsbuchstaben des Namens einzugeben.

  Der Satz, der durch einen "FIND"-Befehl lokalisert wurde, wird automatisch zum aktuellen Satz. Er läßt sich mit Hilfe des Befehls "DISPLAY" anzeigen.

  Beispiel : Der Name "Meier" soll aus einer indizierten Datenbankdatei "ADRESSE1" lokalisiert werden.

```
USE B: ADRESSEN INDEX ADRESSE1
FIND MEIER
DISPLAY
```


Mit dem Befehl " L O C A T E " läßt sich ein bestimmter Satz in einer Datenbank auffinden.

* Ist der "LOCATE"-Befehl erfolgreich, erscheint der Hinweis "Satz n" auf dem Bildschirm.

  Locate arbeitet am schnellsten auf nicht indizierten bezw. ohne Schlüsseldateien aktivierten Datenbanken.

  Beispiel : Aus der Datenbank ADRESSEN soll die Adresse von Frau Müller gefunden werden.

```
USE B : ADRESSEN
LOCATE FOR "MüLLER"$NAME
SATZ : XX
DISPLAY
```

  oder es sollen die Adressen herausgefunden werden, deren PLZ größer als 5000 ist.

```
LOCATE FOR PLZ>5000
SATZ : XX
DISPLAY
```

Der Befehl " C O N T I N U E " stellt die Fortsetzung des LOCATE-Befehls dar.

* Zwischen den Befehlen "LOCATE" und "CONTINUE" können ggf. auch andere Befehle zur Ausführung gebracht werden.

  Beispiel :

```
USE B : ADRESSEN
LOCATE FOR PLZ>5000
SATZ : XXXX
DISPLAY

CONTINUE
SATZ : XXXX
DISPLAY
CONTINUE
.
.
DATEIENDE ERREICHT
```

6. Positionieren auf einen Datensatz :

Mit dem Befehl " G O " oder " G O T O " läßt sich der dbase-interne Zeiger auf einen bestimmten Datensatz neu positionieren.

* Der Befehl "DISPLAY" zeigt den neuen Datensatz an.

  Beispiel :

```
USE B : ADRESSEN     od.  GO TO      ===> erster Daten-
GO TO RECORD 6                            satz
DISPLAY
                     od.  GO BOTTOM  ===> letzter Daten-
                                          satz
```

7. Einfügen von neuen Datensätzen :

Der Befehl " I N S E R T " ermöglicht das Einfügen eines Satzes an einer liebigen Position der aktivierten Datenbank.

* Es kann mit jedem Befehl nur ein Datensatz in die Datenbank eingeben werden.
Bei großen nicht indizierten Datenbanken ist der " I N S E R T " Befehl sehr zeitaufwendig.

Beispiel 1 : Einfügen hinter dem aktuellen Datensatz

Beispiel 2 : Einfügen vor dem aktuellen Datensatz

Beispiel 3 : Einfügen eines leeren Datensatzes

8. Zählen von Datensätzen :

Die Anzahl der Sätze in einer Datenbank werden mit dem Befehl " C O U N T " gezählt.

* dbase zeigt das Ergebnis in folgender Form an

" Anzahl der Sätze = xxxxx"

Beispiel 1 : Die Anzahl der Sätze in der Datenbank "ADRESSEN" soll gezählt werden.

```
. USE B:ADRESSEN
. COUNT
ANZAHL DER SATZE = 00008
```

Beispiel 2 : Es sollen in der Datenbank "ADRESSEN" nur die Sätze gezählt werden, deren Postleitzahlen größer als 2300 sind.

```
.USE B:ADRESSEN
.COUNT FOR POSTLZ >2300
ANZAHL DER SATZE = 00012
```

9. Überspringen von Datensätzen :

Mit Hilfe des " S K I P " Befehls kann der Zeiger, der in dbase auf den aktuellen Datensatz zeigt, auf vorhergehende bezw. nachfolgende Sätze gesetzt werden.

Beispiel 1 :

```
.USE B:ADRESSEN
.LIST
.5
.DISPLAY
.SKIP 3
.DISP
.SKIP-6
.DISP
.SKIP 2*2
.DISP
```


10. Sortieren der Datenbankdatei :

Der Befehl " S O R T " bewirkt die Sortierung der aktivierten Datenbank.

* Sortiert wird nach dem angegebenen Datenfeld
Ausgabedatei ist eine neue anzugebende Datei
Die aktivierte Datenbank wird nicht geändert und bleibt weiter im Zugriff.

```
.SORT ON feld TO dateiname
```

Beispiel :

```
.USE B:ADRESSEN
.SORT ON POSTLZ TO ADRESSE1
.USE B:ADRESSE1
.LIST
```

11. Umbenennung von Datenbankdateien :

Beliebige Dateien in dbase lassen sich mit dem Befehl " R E N A M E " umbenennen.

* Die Voreinstellung für den Dateityp ist ".DBF" .
Eine umzubenennende Datei darf nicht geöffnet (USE-Befehl) sein.

```
RENAME B:alter Name TO B:neuer Name
```

Beispiel : .RENAME B:ADRESSEN TO B:ANSCHRIFT

12. Anzeige von Datenbanken :

Mit Hilfe dieses Befehls " D I S P L A Y   F I L E S " lassen sich alle auf dem angegebenen Laufwerk vorhandenen Dateien vom Typ ".DBF" anzeigen.

Beispiel 1 : alle "DBF" - Dateien von Laufwerk B sollen angezeigt werden.

```
:DISPLAY FILES ON B:
```

Beispiel 2 : alle Dateien von Laufwerk B sollen angezeigt werden.

```
DISPLAY FILES ON B: LIKE *:*
```

weitere Möglichkeiten :

nur Indexdateien : ......LIKE *.NDX
nur Reportdateien : ..... LIKE *.FRM

Das war der 2. Abschnitt über dbase II . Er befaßte sich mit den notwendigen Befehlen zu Veränderungen und Sortierungen von Datenbanken. Beim nächsten Info geht es dann weiter mit der Erstellung von kpl. Programmen, sowie noch weiteren notwendigen Befehlen.

Ich wünsche Euch viel Erfolg bei der Anwendung.

Harald

Maus mit sieben Beinen

In der 22. Ausgabe des Clubinfos stellt Hartmut eine kleine Platine vor, mit deren Hilfe sich eine nicht seriell arbeitende Maus an unsere Computer anschließen läßt. Wer nur einen Funken Moral im Leibe und einen Hunni übrig hat, baut das Ding natürlich nach, so auch ich. Die hundert Mark gelten inklusive der Maus und aller Teile, versteht sich.

Der elektronische Schalter 4066 war bei meinem Dealer nicht vorrätig. Statt dessen gab er mir zwei Exemplare des 4016, der lt. Beschreibung praktisch identisch ist. Damit lief die Schaltung aber nicht. Daher möchte ich empfehlen, für die von Hartmut vorgeschlagenen Teile keine Ersatztypen zu nehmen.

Wir sehen im Info, daß Hartmut zwei Taster seiner Maus benutzt. Meine hat auch zwei, die jedoch einfach miteinander kurzgeschlossen waren. Hartmuts auch, aber er hat die Verbindung aufgetrennt und eine noch ungenutzte Leitung des Kabels damit bedröhnt. Nach dieser Anregung suchte und fand ich sogar noch zwei freie Leitungen, so daß ein weiterer Taster angeschlossen werden konnte.

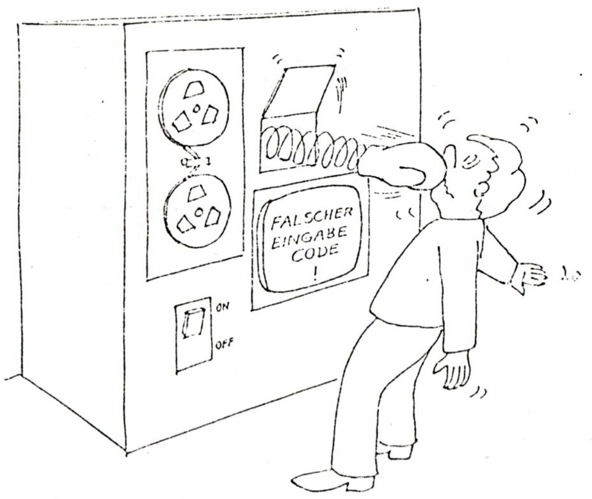
Hier war Bastelei in der Maus nötig: Zunächst mußte Hartmuts Operation durchgeführt werden, um die beiden bereits vorhandenen Tasten einzeln nutzbar zu machen. Für einen dritten Knopf fand sich Platz auf der kleinen Zusatzplatine, auf der die beiden Drucktasten sitzen. Die dafür notwendige Bohrung trennte eine Leitung auf, die jedoch mit einem Stück Draht wieder geschlossen werden konnte. Es handelt sich dabei um die Common-Leitung aller Tasten und der Rollenelektronik. Daher mußte sie auch an die neue Taste angeschlossen werden. Die letzte freie Leitung des Kabels kam an ihren zweiten Pin.

Elektronisch gab es also keine Schwierigkeiten. Mechanisch ist die Geschichte jedoch etwas heikel. Der einzubauende neue Taster braucht natürlich Raum. Glücklicherweise fand sich einer, dessen unten aus der Tastenplatine herausragender Teil mit knapper Not noch nicht die darunter liegenden Bauteile der Hauptplatine berührte. Es mußte dafür aber bereits ein wenig geschliffen werden. Sein Druckknopf steht so hoch über, daß er durch eine niederzubringende Bohrung im Gehäusedeckel gerade etwa 3 mm herausragt. Das ist eine optimale Arbeitshöhe.

Wer das nachbauen möchte, sollte sich unbedingt in seiner Maus sehr sorgfältig nach passenden Plätzen umsehen. Es ist auch nicht verkehrt, vorsichtshalber zwei oder drei verschiedene kleine Taster zur Auswahl bereitliegen zu haben. Wenn eine so glückliche Anordnung wie in meinem Exemplar nicht möglich ist, geht aber auch die einfachste Lösung: Die Taste wird über einem genügend großen Hohlraum, wovon es in allen Mäusen genug gibt, direkt in den Gehäusedeckel geschraubt. Um den Deckel ganz vom Boden trennen zu können, empfiehlt sich dann eine Steckverbindung zur übrigen Elektronik.

Leider mußte nun auch eine neue Obermann-Platine her, denn für sieben Anschlüsse hat der 7404 einen Inverter zuwenig. Ein zweiter Chip mußte sein. Außerdem mußte die Steckverbindung von dieser Platine zur Tastatur um zwei Pins verbreitert werden, die auch nicht mehr auf die sehr kleine alte Platine paßten. Im Prinzip hat sich aber nichts an dem von Hartmut vorgestellten Schaltungsvorschlag geändert, so daß die neue Schaltung hier nicht vorgestellt werden muß.

Freilich kann ich nur für meine eigene Maus (M1 von Noris Data) versichern, daß die Umbauten problemlos machbar sind. Hartmuts Maus sieht inwendig genauso aus, wahrscheinlich alle anderen auch. Man kann auf der Ladentheke schlecht dem Teil mit dem Schraubenzieher zuleibe gehen, um das zu überprüfen, aber die Chance, ein geeignetes Modell zu erwischen, dürfte bei 100% liegen.



Datenschutz automatisiert

Möchte jemand meine alte Platine haben? Sie treibt die Rollenelektronik und eine Taste. Auf dem 7404 ist noch ein Inverter, und auf dem zweiten 4066 sind noch drei Relais frei, so daß Hartmuts Trick mit der getrennten Verbindung zwischen den beiden vorhandenen Tasten ohne Anbauten noch durchführbar ist. Ich möchte die Platine aber nur unbestückt hergeben, denn in Lübeck kriege ich keinen 4066. Bitte melden!

In der Praxis stellte sich heraus, daß die Impulse von der Rolle sehr rasch aufeinander folgen. Die Maus darf daher nur sehr langsam bewegt werden. Vielleicht sollte man jedes zweite Loch in den beiden Blendenscheiben der Rollenmechanik verstopfen, so daß nur noch halb so oft ein Impuls erfolgt. Dann wäre die Garantie für die Maus endgültig verwirkt. Aber das hat den begeisterten Fummler noch selten abgeschreckt.

Es ist eine Wonne, mit der Maus zu arbeiten. Gerade eben beim Korrekturlesen dieses Textes karriökere ich damit im Bildschirm kreuz und quer, Maus in der Rechten, Bier in der Linken, Augen nur noch auf dem Text, nicht mehr auf der Tastatur. Das Teil arbeitet nämlich nicht nur mit Graphikprogrammen zusammen, sondern mit allem, was die Pfeiltasten und die anderen Tasten aus der Keyboard-Reihe 3840h bespitzelt. Also auch mit TSCRIPS. Hartmut, Deine Idee war Sahne!

Arnulf Sopp

Noch ein Druckerpatch

von Manfred Held

Nein,nein keine Angst, nicht noch ein X'der Patch im System um irgendwelche Steuerzeichen oder Grafikzeichen darzustellen.
Diesesmal ein Hardpatch ????!

Folgendes Problem:
Bei Newdos, Gdos ect. wird CR in LF umgewandelt. Es wird an den Drucker also nur CR (0Dh) ausgegeben. Der Drucker muss so eingestellt werden, dass nach einem CR automatisch ein Zeilenvorschub gemacht wird. Man kann dem Betriebssystem zwar beibringen, dass CR und LF getrennt zum Drucker gebracht wird, dies wurde auch schon oeffter getan, aber mit Programmen, die eigene Druckertreiber haben, steht man wieder vor demselben Problem.

Anders beim Betriebssystem CP/M. CP/M sendet CR und LF getrennt. Es wird keine Umwandlung vorgenommen. Auch die Programme behandeln CR und LF als zwei getrennte Zeichen.

Im Betriebssystem CP/M unseres Computers gibt es einen Softschalter, der wahlweise LF zu 00h umwandelt. Somit koennen wenigstens die Programme unter CP/M, in begrenztem Umfang genutzt werden.

Damit die Programme in CP/M die Faehigkeiten des Druckers voll nutzen koennen, benoetigt der Drucker ein getrenntes CR und LF. Das steht aber im Wiederspruch mit dem Drucker, der eben andes eingestellt werden muss.

Die Loesung:
Laut Druckerhandbuch gibt es einen Schalter oder Jumper (Autolinefeed oder CR(AUTO FEED XT) oder sonst irgendwie) damit der Drucker nach Empfang des CR einen Zeilenvorschub ausfuehrt.
Wichtiger ist aber der Anschluss am Drucker. Laut meinem Handbuch hat der Drucker am Pin 14 des Centronics-Steckers einen Eingang mit der Bezeichnung "AUTO FEED XT". In der Beschreibung zu den Anschlusspins des Steckers heisst es :

> Empfaengt der Drucker einen CR-Code, wenn dieses Signal L ist, schiebt er das Papier nach Druckende automatisch um eine Zeile vor.

Das heisst, wenn L-Pegel an diesem Pin anliegt, und CR empfangen wird, fuehrt der Drucker einen Zeilenvorschub aus. Liegt H-Pegel an, wird CR und LF getrennt behandelt. Der Drucker fuehrt keinen Zeilenvorschub aus, wenn CR alleine empfangen wird sondern wartet bis LF empfangen wird.
Wichtig dabei ist , dass der Schalter CR (AUTO FEED XT) auf "Drucken ohne LF" steht.

Fuer unser Problem heisst das, unter Newdos, Gdos ect. muss der Eingang von Pin 14 auf L-Pegel liegen, fuer CP/M auf H-Pegel.

Man koennte jetzt einen Schalter in den Anschlusstecker einbauen, mit dem Pin 14 einmal auf Masse geschaltet wird oder auf +5V. Das hat aber den Nachteil, vergisst man den Schalter bei Betriebssystemwechsel umzuschalten, gibt es Schrott am Drucker.

Die bessere Loesung ist, man sucht sich ein freies Port und ein freies Pin an dem Port und verbindet das Pin mit dem Eingang am Drucker, es muss nur beachtet werden, dass die Ruhelage des Ports nach einem Reset, L-Pegel ist.
Der Rest ist Software.
Ich habe im Bootsector des CP/M's ein freies Plaetzchen gefunden und vier Byte eingepflanzt, die jetzt bei Arbeiten mit CP/M den Drucker umschalten.
Somit stehen mir alle Moeglichkeiten des Druckers unter Newdos und CP/M voll zur Verfuegung.

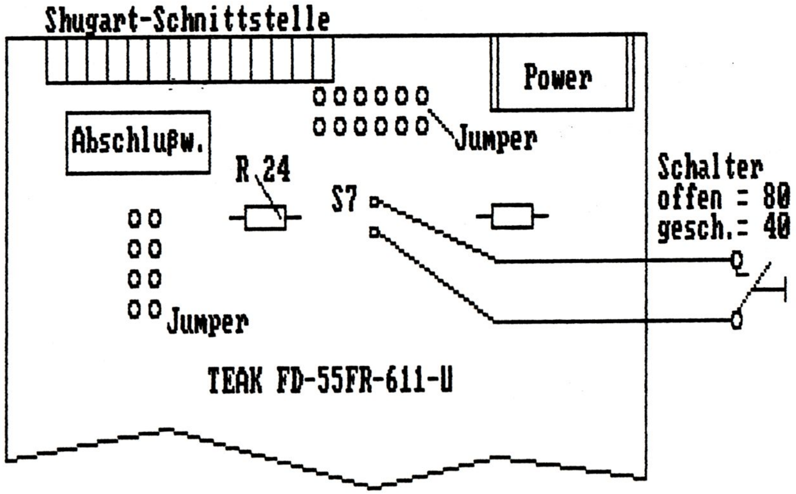
Wieder mal: 40 / 80 - Trackumschaltung diesmal: TEAK FD-55FR-611-U-Laufwerke

Die Firma TEAK scheint ständig bemüht ihre Laufwerksproduktion auf dem neuesten technischen Stand zu halten. Dadurch kommt aber mancher Käufer eines 80-Spur-TEAK-Laufwerks, von dem behauptet wird, daß es auf 40-Spurbetrieb umschaltbar ist, ganz schön in Schwierigkeiten. Immerhin erschienen allein im CLUB 80-Info schon zwei verschiedene Anleitungen für die Umschaltung von TEAK-Laufwerken.

Vor kurzem liefen mir zwei Laufwerke über den Weg, die man einem Bekannten als 40-Spurer verkauft hatte. Bei genauerer Betrachtung stellte sich heraus, daß es sich um Laufwerke mit der Bezeichnung FD-55FR-611-U handelte, die von einem anderen Anbieter als umschaltbare 80'er verkauft werden. Nach kurzer Suche war die Stelle gefunden, an der ein einfacher Ein/Aus-Schalter eingelötet werden muß, um sowohl 40 als auch 80 Tracks benutzen zu können.

Viel Spaß und Glück beim Löten, euer

Hartmut Obermann



Anmerkungen zum Club-80-ECB-Bus-Projekt

Liebe Club-Freunde, die Ihr den Mut gehabt habt, Euch an die Löterei des ECB-Bus-Projektes heranzuwagen, Euch speziell gilt der folgende Artikel zu dem Thema. Vorher möchte ich aber noch einige Bemerkungen loswerden, die verdeutlichen sollen, warum ich meine etwas abweichenden Vorstellungen eines ECB-Bus-Interfaces denen Eures Projektes entgegenstellen muß.

Die Grundvorstellung des ECB-Bus ist nicht mur das Umsortieren einer wilden Pinanordnung auf den Systembus-Steckern unserer Computer auf die noch unübersichtlichere des ECB-Bus sondern noch eine Reihe weiterer Srtrukturen, die speziell auf die besonderen Fähigkeiten der Z80-CPU abgestimmt sind und auch von einem vollständigen ECB-Bus unterstützt werden.

Dazu gehört zunächst der Interrupt Mode 2 (IM2), der voraussetzt, daß die Interrupt-auslösende Karte der CPU beim Interrupt-Acknowledge einen 8Bit-Vektor über den Datenbus mitteilt. Dieser Vektor muß vom ECB-Bus über den Datenbuffer auch zur CPU gelangen können, wenn IORQ* und M1* gleichzeitig low sind.

Außerdem müssen die Buffer aller Bussignale bidirektional sein, weil bei DMA-Zugriff auf Komponenten des CPU-Boards der DMA-Controller auf dem ECB-Bus auch die entsprechenden Baugruppen des CPU-Boards adressieren muß. Das bedeutet einen sehr viel größeren Aufwand bei der Steuerung der Treiberrichtung dieser Buffer.

Und schließlich soll der ECB-Bus nicht nur selbstgestrickte Karten aufnehmen sondern hauptsächlich den Einsatz der breiten Palette käuflicher Karten für ECB-Systeme ermöglichen. Daher verbietet sich auch die prinzipiell geniale Lösung, der Richtungssteuerung des Datenbuffers über Freigabesignale, die die ECB-Bus-Karten liefern. Kommerzielle Karten wissen nicht, daß Euer Bus solche Signale voraussetzt.

Das Konzept Eures ECB-Bus baut auf den Möglichkeiten des TRS-80-Systembus auf, die leider wirklich recht bescheiden sind. IM2- und DMA-Fähigkeit sind dort nicht vorhanden. Beim GENIE und beim Komtek 1 lassen sich aber alle Eigenschaften des ECB-Bus verwirklichen, ohne intern große Eingriffe machen zu müssen.

Um aber auch beim TRS-80 zumindest 90% der kommerziellen Karten, die ohne IM2 und DMA arbeiten, nutzen zu können, ist es unumgänglich, von der Richtungssteuerung des Datenbuffers durch die Karten selbst abzukommen. Da ist dann auch die bei mir eingeschlagene Ochsentour nötig, die Richtungssteuerung aus den Freigabesignalen der Baugruppen innerhalb des Computers herzuleiten.

Die im folgenden Artikel beschriebene ECB-Bus-Eingangkarte berücksichtigt nicht nur die o.a. Gesichtspunkte sondern bringt noch ein paar zusätzliche Leistungen, die nicht unbedingt an den ECB-Bus gebunden sind, sondern einige andere Hardware-Erweiterungen unterstützen. Diese Features sind aber nicht unbedingt wichtig und können auch ausgespart werden.

Es ist beim GENIE und Komtek ohne weiteres möglich, den ECB-Bus mit dieser Eingangskarte anstelle der alten zu betreiben. Beim TRS-80 sind noch zusätzliche Maßnahmen nötig, die einen voll funktionsfähigen ECB-Bus ermöglichen.

Helmut Bernhardt

Gepufferter DMA- und IM2-fähiger ECB-Bus für GENIE I und II

Helmut Bernhardt

Während bei Verzicht auf DMA- und IM2-Fähigkeit die Steuerung des Datenbuffers zwischen Computer und externem ECB-Bus relativ einfach zu realisieren ist (der Buffer darf nur in Richtung CPU treiben, wenn Daten von Baugruppen gelesen werden, die nicht auf der CPU-Seite des Buffers - also auf dem ECB-Bus - sitzen), sind bei der Einbindung dieser Features einige zusätzliche Aspekte zu berücksichtigen.

Der Interrupt-Mode2 der Z80-CPU setzt voraus, daß der Interrupt-Lieferant beim Interrupt-Acknowledge (M1* und IORQ* beide low) ein 8Bit-Wort auf den Datenbus legt, das die CPU als LSB einer Adresse in der Interrupt-Vektor-Tabelle interpretiert. Dieses Byte muß von einer Interrupt-Quelle vom ECB-Bus durch den Datenbuffer auch zur CPU gelangen. Der Buffer muß beim Interrupt-Acknowledge also auch in Richtung CPU treiben.

Bei übernahme der Daten-, Adreß- und Steuerbusse durch einen DMA-Controller (DMAC) auf dem ECB-Bus muß dieser auch Zugriff auf RAM, EPROM und alle CPU-seitig des Buffers liegenden Baugruppen haben. Für Adreß- und Steuerleitungen gestaltet sich diese Richtungssteuerung recht einfach. Hier genügt es, wenn durch das low aktive CPU-Signal BUSAK* (nur dann darf der DMAC arbeiten) die Buffer dieser Signale in Richtung zur CPU treiben, so daß der DMAC dort entsprechend adressieren kann. Für die Steuerung des Datenbuffers sind noch weitere Umstände zu berücksichtigen.

Wenn die CPU den Bus kontrolliert, kann beim Schreiben der Buffer immer in Richtung ECB-Bus treiben. Beim Lesen darf er nur dann nicht in Richtung CPU treiben, wenn interne Baugruppen (CPU-seitig des Buffers) adressiert werden. Sonst würde ein von außen gelesenes FFH gegen ein von der intern adressierten Baugruppe ausgegebenes Datenwort arbeiten. Abgesehen davon, daß die gegeneinander arbeitenden Komponenten sich irgendwann gegenseitig zerstören, würde die CPU Fehlinformationen lesen.

Wenn der DMAC (45) den Bus steuert, ist Lesen und Schreiben für die Treiberrichtung des Buffers das ganze Gegenteil. Wenn der DMAC Daten einer CPU-seitigen Systemkomponente liest, wobei er dann auch die Leitung RD* low zieht, muß der Buffer in Richtung ECB-Bus treiben. Das darf er aber nur dann, wenn er auf CPU-seitige Komponenten zugreift. Wenn er von einer Baugruppe auf dem ECB-Bus liest, muß der Buffer in Richtung CPU treiben, um dann nicht auf dem ECB-Bus zwei Datenwörter gegeneinander arbeiten zu lassen.

Für die Treiberrichtung des Buffers muß also berücksichtigt werden, ob der DMAC oder die CPU den Bus kontrolliert, ob die adressierte Baugruppe auf dem ECB-Bus oder CPU-seitig des Buffers liegt und ob gelesen oder geschrieben wird.

In Tabelle 1 sind die für verschiedene Betriebszustände aus obigen überlegungen abgeleiteten Treiberrichtungen (von der CPU aus gesehen) zusammengestellt. Dabei ist zu berücksichtigen, daß zusätzlich auch noch beim Interrupt-Acknowledge die Richtung "rein" sein muß, weil der GENIE keine eigenen vektorisierten INT-Quellen hat. Diese Tabelle ist Grundlage der Schaltung in Abb.1. Die Richtungssteuerung soll ausschließlich aus Signalen, die von CPU-seitigen Baugruppen zu beziehen sind, erfolgen. Nur dann ist auch gewährleistet, daß auf dem ECB-Bus jede handelsübliche Karte funktioniert. Kommerzielle ECB-Bus-Karten legen kein aktives Signal auf den Bus, das in die Buffersteuerung einbezogen werden kann.

Tabelle 1: Treiberrichtung des ECB-Bus Databuffers

Bus-Steuerung durch	interne Baugruppe adressiert Read	Write	externe Baugruppe adressiert Read	Write
CPU	raus	raus	rein	raus
DMAC	raus	rein	rein	rein

Der Aufwand dafür ist aber nicht unerheblich und der Anwender dieser Bussteuerung muß sein System genau kennen, um die nötigen Steuersignale an die Schaltung legen zu können. Einen Anhaltspunkt dafür, welche Freigabesignale vorkommen können, mag die Tabelle 2 geben. Darin sind die prinzipiell vorhandenen Baugruppen und einige häufig benutzte Hardware-Erweiterungen und die Punkte, wo deren Freigabesignale abzugreifen sind, zusammengestellt. Diese Punkte sind mit jeweils einem der Punkte A bis H auf der Eingangskarte zu verbinden.

Tabelle 2: Interne Freigabesignale

CPU-Board	Adresse	IC	Typ	Pin
RAM	4000-FFFF	37	367	15
	(3900-3BFF) (7)			
ROM1	0000-0FFF	10	3001	20
ROM2	1000-1FFF	11	3002	20
ROM3	2000-2FFF	12	3003	20
ROM4	3000-36FF(37DF) (7)	13	2716	20
Tastatur	3800-3BFF(38FF) (7)	7	368	1
Video-RAM	3C00-3FFF	35	32	3
Video-Board				
Cassette 1	Port FF	20	32	11
Cassette 2	Port FE	bleibt unberücksichtigt #1		
Expansion-Interface EG3014				
RAM1	8000-BFFF	37	20	10
RAM2	C000-FFFF	37	20	9
Floppy	37E0-37FF	29	139	12
Drucker	37E8 u.Port FD	31	156	9
RS 232 EG3020	Ports F8, F9	37	20	10
EXP1 von RB	37E0-37FF	18	155	2,14
	Port FD	14	32	3
RB.V24	Ports 80-88		Z80-CTC	16
			Z80-DART	35
HRG1B von RB	Ports 00-7F	?	155	2,14
Schmidtke 80Z.Karte	B000-BFFF	?	245	19
	Ports D0, D1	?	138	15
EG64MBA	Port DF	bleibt unberücksichtigt #1		
EG64MBA+ (6)	Ports DE, DF	"	"	"
256K-Banker (10)	Port EC	"	"	"
(u. ED mit dieser Karte)		"	"	"
CP/M-Banker (6)	Ports 50-5F	"	"	"

#1 diese Baugruppen werden nicht gelesen und brauchen deshalb nicht in die Buffersteuerung miteinbezogen zu werden, das IN A,(0DFH) des EG64MBA dient nicht der Datenaquisition, es setzt nur das 74LS259-Latch zurück, ein Datenwort wird nicht ausgegeben

Im Normalfall liegen alle memory-mapped (mm) Baugruppen CPU-seitig des Treibers, so daß dann einzig MERQ* (IC16, 74LS367, Pin5) an die Schaltung geführt werden muß. Wenn aber auch nur eine mm Baugruppe auf dem ECB-Bus steckt, müssen alle anderen mm Freigabesignale einzeln an die Schaltung geführt werden.

Durch Vorschalten weiterer AND-Gatter vor die Eingänge des 74LS30-ICs (z.B. 74LS08, 74LS11, 74LS21) kann die Anzahl der verwendbaren Eingänge für Freigabesignale beliebig erhöht werden (Abb.10).

Die DMA-Fähigkeit der ECB-Busbuffer alleine reicht nicht aus. Es muß durch Verbinden der Systembus-Leitung BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS* auch dafür gesorgt werden, daß beim DMA-Zugriff die Buffer der CPU-Signale in den hochohmigen Zustand versetzt werden. Das kann entweder auf der Eingangskarte durch Querverdrahtung der entsprechenden Pins von CN2 bzw. CN3 oder durch feste Verdrahtung auf dem CPU-Board gemäß Abb.11 erreicht werden.

Da für die Richtungssteuerung des Datenbuffers ohnehin einige ICs nötig waren, und der ECB-Bus auch IM2-fähig ist, ließ sich mit etwas Mehraufwand auch gleich eine Vektorisierung der Interrupts des 25ms-Timers und des Floppy-Controllers realisieren. Dafür ist Voraussetzung, daß auf dem ECB-Bus auf einer Karte ein freier Z80-CTC-Kanal (2) vorhanden ist, über den durch Software das gemeinsame INT*-Signal des Systems umgeleitet werden kann. Der CTC-Kanal ist dann im Zähler-Modus mit der Zählkonstanten 1 und Triggern durch eine negative Flanke zu programmieren.

Die direkte Verbindung des INT*-Signals (im Schaltplan INTEXP*) von der Floppy-Baugruppe zur CPU ist dafür aufzutrennen. Das Signal INTEXP* ist stattdessen an den entsprechend bezeichneten Punkt der Schaltung und an den freien Triggereingang des CTC-Kanals zu legen. Die INT*-Leitung des ECB-Bus ist an den Punkt INTECB* zu legen und der Ausgang INTCPU* ist mit dem INT-Eingang der CPU (Z80, Pin16) zu verbinden.

Damit ist gewährleistet, daß nach dem Einschalten oder Drücken des RESET-Knopfes immer der Interrupt vom FDC-Board direkt zur CPU durchgeschaltet wird und das Betriebssystem im IM1 arbeiten kann. Wenn allerdings andere Baugruppen auf dem ECB-Bus in Echtzeit im IM2 betrieben werden sollen, müssen auch die systemeigenen Interrupts dieses Spiel mitmachen können und das geht nur dann, wenn ihnen der CTC-Kanal beim INT-Acknowledge einen Vektor zur Verfügung stellt. Die Schaltung leistet es, daß ein LD A,xxxx xx1xB; OUT (0EDH),A das INT* Signal des FDC-Boards über den CTC umleitet.

Die Software muß dann aber Rücksicht auf diesen Umstand nehmen. In der INT-Vektor-Tabelle darf nicht einfach 4012H stehen (INT-Vektor des DOS), da die Serviceroutine des DOS nicht mit RETI sondern mit RET endet und der Z80-CTC dann nicht weiß, wann er seinen IEO-Ausgang wieder auf High zurücknehmen soll. Es muß folgender Umweg eingeschlagen werden:

```
INTAB  DEFW   SERV1   ;Anfang INT-Tabelle
       DEFW   SERV2   ;nächster Vektor
       DEFW   .....   ;u.s.w.
       ....   .....
       DEFW   CTCN    ;Vektor für Serviceroutine des INT vom CTC-Kanal
       ....           ;N, über den die System-Interrupts laufen
       DEFW   LETZT   ;Ende der Vektor-Tabelle
       ....   .....
CTCN   CALL   4012H   ;Serviceroutine des DOS
       EI
       RETI
```

Außerdem bleibt es dem Anwender offen, während des IM2-Betriebes anderer Baugruppen den 25ms INT verhungern zu lassen. Wenn beim Lable CTCN einfach ein EI, RETI steht, wird 37E0H nicht mehr gelesen und daraufhin erfolgt kein weiterer Interrupt des Timers mehr. Eine Reinitialisierung des Timer-INTs geschieht durch Umprogrammieren des Vektors auf ein anderes Lable, das wieder den CALL 4012H enthält, und einmaliges Lesen von 37E0H.

Durch Zurücksetzen von D1 in Port EDH wird wieder der INT des FDC-Boards direkt an die CPU durchgeschaltet und alle Interrupts des ECB-Bus werden unterbunden.

Mit der für die Interrupt-Steuerung benötigten Portdecodierung und mit dem Latch wurden noch einige zusätzliche Funktionen realisiert. Zunächst wurden auch die Portfreigabesignale OUTECH*, OUTEEH* und OUTEFH* hergeleitet. Davon kann OUTECH* zur Freigabe des 74LS273 Latch auf dem 256K-Banker (10) benutzt werden. Dieses Signal wird direkt an den Pin11 des 74LS273 geführt, wobei dann aber die bisher dahin führende Leitung zu durchtrennen ist. Damit ist dann das 74LS30 IC auf dem Banker überflüssig und auch die Leitungen A0 bis A5 sowie IORQ* und WR* brauchen nicht mehr dorthin geführt zu werden (A6 und A7 werden dort nicht nur zur Portdecodierung sondern auch noch zur Erzeugung der 8Bit-Refreshadresse benötigt und müssen deshalb weiterhin angeschlossen bleiben).

Mit Bit 0 in Port EDH wurde ein softgesteuertes Invertieren des zum 256K-Banker führenden Signals A15 realisiert. Damit läßt sich mit D0=0 an Port EDH (Einschaltzustand) vorgeben, daß die unteren 32K Adreßbereich des Z80 gebankt werden können, während D0=1 an Port EDH das Banking der oberen 32K festlegt.

Und schließlich kann mit D2=1 an Port EDH ein gemeinsames Invertieren der Adressen A14 und A15 erreicht werden. Die dabei gewonnenen Signale A14′ und A15′ können anstelle der Signale A14 und A15 an die Decoderschaltung für die 16K (ROMs und mm I/O), IC25, 74LS139, Pins 14 und 13 geführt werden und gestatten dadurch ein Verlegen dieser Komponenten an das obere Speicherende 48-64K. In diesem Bereich funktionieren dann zwar die ROMs nicht mehr, wenn man aber dann noch ein 74LS32 IC spendiert und mit dem Signal ED,3 (D3 von Port EDH) ein programmierbares Abschalten der ROMs einbaut (Abb.10), hat man auch noch die Funktionen des Omikron Mappers (Kompatibler Selbstbau in (9)) für den Betrieb von CP/M realisiert. Dieser leistet (wenn auch mit einem anderen OUT-Befehl) ebenfalls das Einstellen der Systemkonfiguration:

0000H-F7DFH	RAM (7) #1
F7E0H-F7FFH	Floppy, Drucker
F800H-FBFFH	Tastatur (7)
FC00H-FFFFH	Video-RAM

#1 der Omikron Mapper für den TRS 80 leistet selbst eine vollständige Decodierung des Bereichs des Sonder-ROMs des GENIE

Ein Patchen der CP/M Systemdiskette ist sehr einfach. Auf den Systemspuren läßt sich relativ einfach die Sprungleiste des Loader Bios finden, die aus einer Reihe von Jumps besteht (C9 nn nn C9 nn ...). Daran schließen sich die Disk Parameter Tabellen mit sehr vielen 00H an und unmittelbar danach stehen die Befehle LD A,40H; OUT (50H),A. Diese beiden Befehle sind durch die Befehle LD A,xxxx 11xxB; OUT (0EDH),A zu ersetzen. Damit läuft das CP/M des Omikron Mappers dann auch ohne denselben.

Wenn diese Eingriffe auf dem CPU-Board gemacht werden, ist der Computer aber von der ECB-Bus-Eingangskarte abhängig, weil von dieser dann lebenswichtige

Signale geliefert werden. Der ECB-Bus braucht selbst nicht angeschlossen zu sein.

Von dieser Abhängigkeit kann man sich durch einen Mehrfach- (oder mehrere Wenigfach-) Umschalter lösen. Abb.9 zeigt, wie man damit entweder die ursprünglichen Signale des CPU-Boards oder die steuerbaren Signale dieser Schaltung an die entsprechenden Stellen des CPU-Boards führt, und damit das Gerät entweder wie bislang ohne ECB-Eingangskarte oder mit derselben und allen ihren Mehrleistungen laufen lassen kann.

Um den Umfang der (zusätzlich zu denen des Systembus) an die Eingangskarte geführten Leitungen etwas zu reduzieren, bietet es sich an, IC7, 74LS30 auf der Eingangskarte nicht zu bestücken und dieses IC im Computer an geeigneter Stelle huckepack mit den Pins 7 und 14 auf ein anderes Dil14 IC zu löten und die internen Freigabesignale fest dorthin zu verlegen (Pins 1-6, 11-12). Dann braucht nur noch das Signal INTERN von Pin8 des 74LS30 an die Eingangskarte (CN2, Pin25c) geführt zu werden.

Wenn die zusätzlichen Features der Eingangskarte voll genutzt werden sollen, können alle benötigten Signale (SYSRES* A15* A14′ A15′ ED,3 OUTECH*) über freie Verdrahtung an unbelegte Pins von CN2 gelötet werden. Im Layout sind nur die Sinale INTCPU* INTEXP* INTECB* und INTERN über CN2 geführt. Dadurch können alle Verbindungen mit dem Computer (nicht nur der Systembus) über CN2 laufen.

Wer diese Variante des ECB-Bus bauen möchte, kann bei mir für 10 DM (incl. Rückporto) eine geätzte aber ungebohrte Karte im Europaformat bekommen. Sollte irgendjemand die Karte am Komtek 1 betreiben wollen, kann ich entsprechende Informationen zur Verfügung stellen, die eine Nutzung auch dort ermöglichen. Der Anschluß an den TRS 80 ist nicht ohne Sondermaßnahmen möglich, weil die internen Datentreiber IM2-Betrieb verhindern. Ein Hardware-Patch, der dagegen hilft, ist in (8) beschrieben. Eine Herleitung der Signale IORQ* MERQ* RD* und WR* aus den daraus gewonnenen Signalen IN* OUT* MWR* und MRD* des TRS 80 Systembus mit Gattern erzeugt Signale mit nicht ganz richtigem Timing, die eventuell Schwierigkeiten bereiten können; hier ist das Abgreifen der originalenSignale eine bessere Lösung.

Aufbau und Anschließen der Karte

Entgegen der üblichen Technik, solche Karten doppelseitig geätzt zu entwerfen, wurde hier ein einseitiges Layout mit einer unverschämten Menge Drahtbrücken bevorzugt, was das Durchkontaktieren an IC-Sockel-Pins vermeidet. Damit wird beim Aufbau aber viel Fummelei und beim Testen viel Ärger erspart.

Der Stecker CN1 (VG64, a,c, Stifte, gewinkelt) wird auf den ECB-Bus gesteckt. Für den Anschluß an den Systembus wurden 2 Varianten vorgesehen. Durch Einbau einer VG64 a,c-Buchsenleiste bei CN2 von der Lötseite (Abb.8) muß durch einen entsprechenden Drahtverhau zum Gegenstück für den 50poligen Platinenrandstecker des CPU-Boards die Anordnung der Signale umsortiert werden. Wer allerdings ein ästhetisches 50poliges Flachkabel bevorzugt, kann CN2 unbestückt lassen und stattdessen in die nicht angeschlossenen Lötaugen von CN3 von der Lötseite einen geeigneten 50poligen Stecker einlöten. Dann wird aber der zum Umsortieren der Signale nötige Drahtverhau nur auf die Karte zwischen CN2 und CN3 verlagert.

Außer den reinen Bus-Signalen des Computers müssen bei Bedarf auch noch die aktiven Steuersignale der Eingangskarte an den Computer zurückgeführt werden und die im Computer abzugreifenden Feigabesignale an die Eingangskarte geführt werden. Dafür sind am oberen Platinenrand entsprechende Lötpunkte vor-

-handen. Die wichtigeren Signale sind aber auch an Lötpunkte von CN2 geführt, womit dann alle Signale über CN2 (und bei entsprechender Querverdrahtung auch CN3) mit dem Computer verbunden werden können. Dafür reicht dann aber der 50polige Platinenrandstecker des CPU-Boards mit seinen 3 freien Pins nicht mehr aus. Da muß dann noch eine zusätzliche fliegende Steckverbindung herhalten, die diese zusätzlichen Signale übergibt.

Seckerbelegungen

CN1 (ECB-Bus) Reihe a	NR.	Reihe c	CN2 (vom Systembus) Reihe a	Nr.	Reihe c
GND	32	GND	GND	32	GND
BUSAK*	31	RESET*	BUSAK*	31	RESET*
A9	30	MERQ*	CLOCK	30	nc
A13	29	CLOCK	MERQ*	29	RFSH*
RFSH*	28	A15	IORQ*	28	RD*
IORQ*	27	A12	M1*	27	WR*
#1	26	#1	HALT*	26	nc
#1	25	HALT*	nc	25	INTERN
#1	24	RD*	nc	24	INTEXP*
#1	23	#1	NMI*	23	INTECB*
-12V #2	22	WR*	A11	22	A8
#1	21	INT*	A10	21	A14
M1*	20	NMI	A13	20	A9
#1	19	#1	A12	19	A15
A14	18	A10	nc	18	nc
#1	17	A11	nc	17	nc
#1	16	IEO	nc	16	nc
-5V	15	#1	nc	15	nc
#1	14	D1	A7	14	A6
+12V	13	#1	A5	13	A4
#1	12	A14	A3	12	A2
BUSRQ*	11	IEI	A1	11	A0
Wait*	10	#1	BUSRQ*	10	nc
A6	9	A7	nc	9	WAIT*
A5	8	A8	nc	8	nc
A4	7	A1	D5	7	D0
A2	6	A3	D6	6	D7
D4	5	A0	D3	5	D2
D3	4	D2	D4	4	D1
D6	3	D7	nc	3	INTCPU*
D5	2	D0	nc	2	nc
+5V	1	+5V	(+5V)	1	(+5V)

Stückliste

ICs:
4 x 74LS245
1 x 74LS174
1 x 74LS139
1 x 74LS125
1 x 74LS86
1 x 74LS32
1 x 74LS30
1 x 74LS11
1 x 74LS08
1 x 74LS00

Widerstände:
2 x 10k

Sonstiges:
1 x VG64 a,c Stifte gewinkelt
1 x VG64 a,c Buchsen gerade
1 x Leiterplatte
jede Menge Draht
viel Geduld

#1 diese Pins sind zum Teil nicht belegt oder führen Signale, die hier nicht benutzt werden
#2 die Lage von -12V wird nicht einheitlich gehandhabt, diese Spannung kann auch woanders liegen (muß bei jeder gekauften Karte geprüft werden)

Literatur

(1) Verbindend - Der ECB-Bus / Johannes Assenbaum / c't 12/85, 60-61

(2) Z80 CTC - Einsatz und Programmierung / Rolf Keller / c't 4/85, 92-94

(3) Interrupt-Priority-Encoder und Vektorgenerator für IM2 mit dem Z80 CTC / H. Bernhardt / Info des GENIE/TRS 80- User-Club Bremerhaven 12/85, 10-11

(4) Direkter Speicherzugriff; Schneller als die CPU - So arbeiten DMA-Controller-ICs / Rolf Keller / c't 8/85, 80-83

(5) Z80 DMA - Schnell, aber kompliziert / Rolf Keller / c't 12/85, 76-79

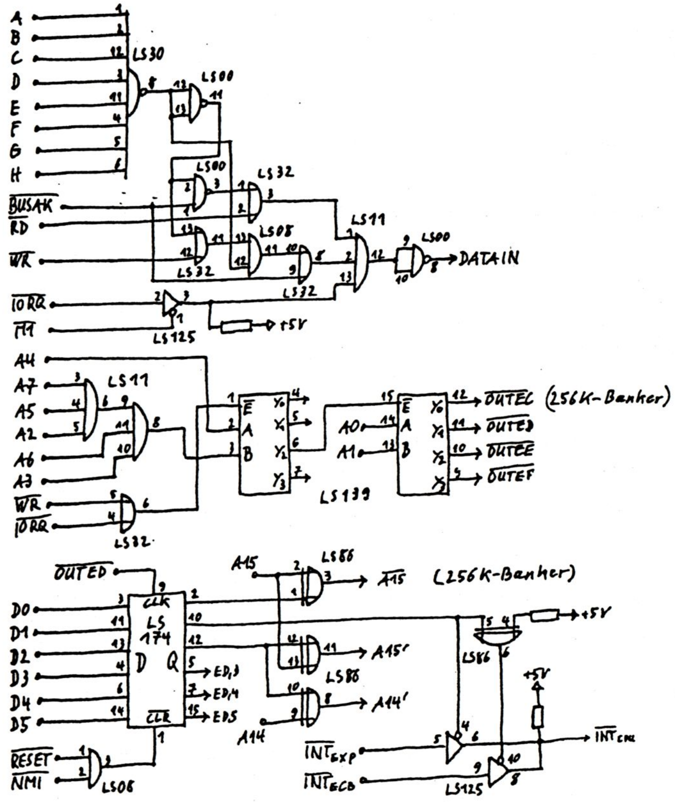
(6) Hardware-Umschaltung für den Betrieb von CP/M bei den Computern TRS 80, GENIE I und II und Komtek 1 / H. Bernhardt / Info des GENIE/TRS 80-User-Club Bremerhaven, Hardware-Sonderheft 11/85, 1-5

(7) GENIE I und II: Voll decodierter Sonder-ROM; RAM im Bereich 3900H bis 3BFFFH / H. Bernhardt / Info des GENIE/TRS 80-User-Club Bremerhaven 11/-85,....?

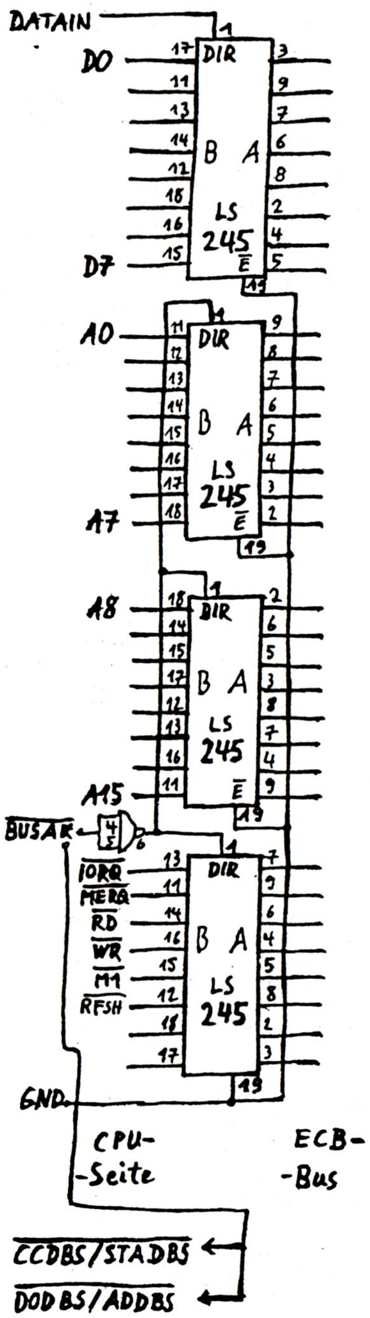
(8) Interrupt Your 80 - Without a hardware mod, your 80 is immune to rude interruptions / Douglas C. Fisher / 80 micro, January 1983, 258-266

(9) CP/M Hardware für TRS-80, GENIE I und II und Komtek 1 / H. Bernhardt, c.Ueberschaar, Info des TRS 80/GENIE-User-Clubs Bremerhaven

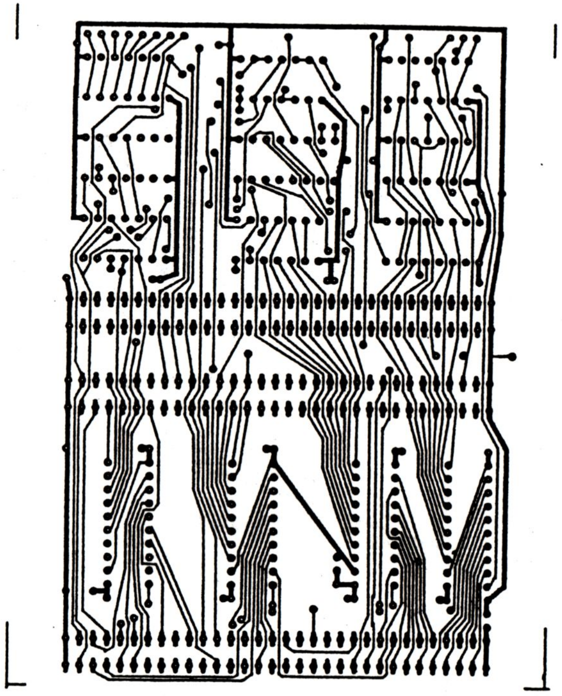
(10) 256K- (1M-) RAM für Z80-Systeme / H. Bernhardt / Info des Club 80 ,Nr.14,



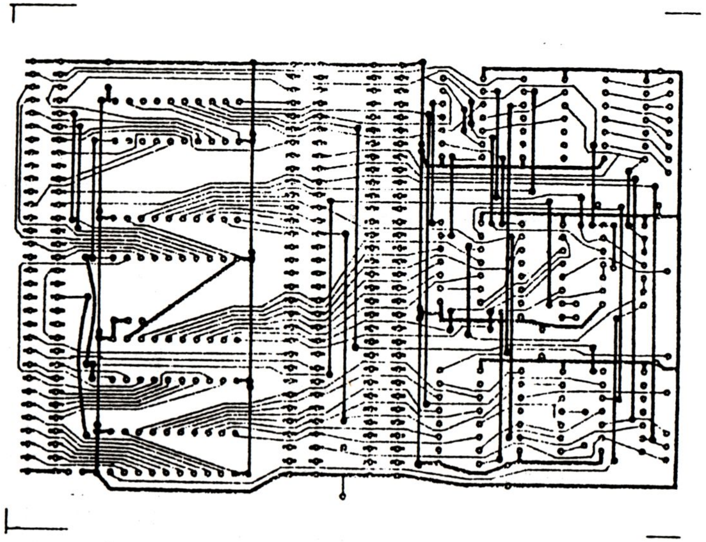
1) Steuerung des Databuffers, Portdecodierung, programmierbares Invertieren von A15 sowie A14 UND A15, Umschalten der Interrupts



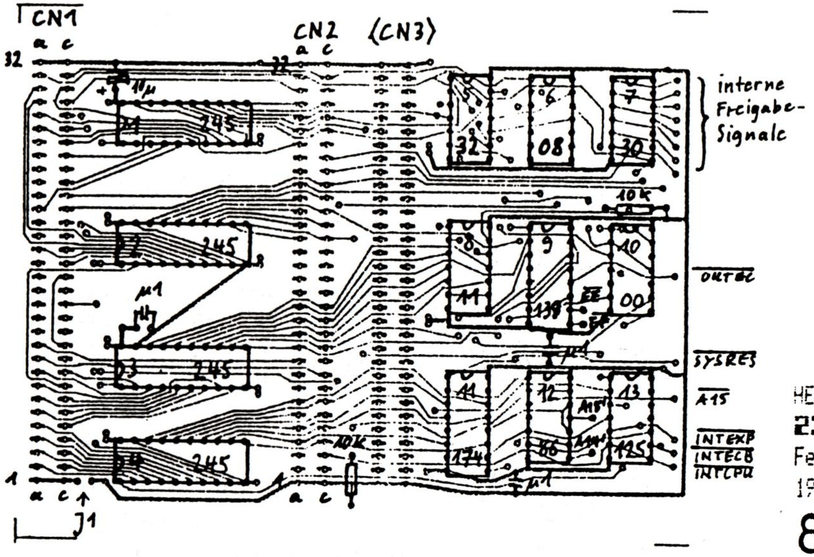
2) Pufferung des ECB-Bus



3) Layout der Eingangskarte

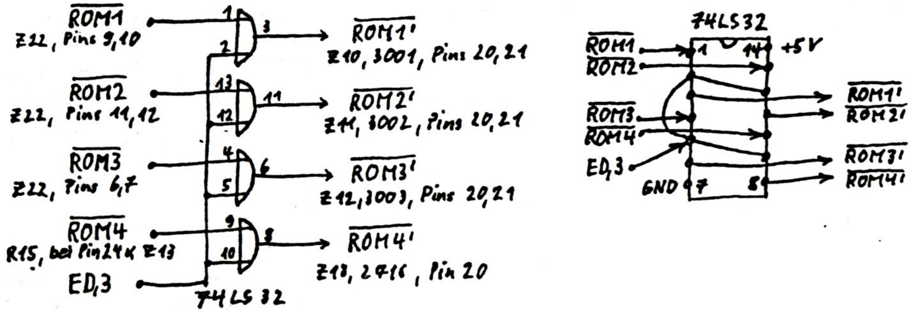


4) Drahtbrückenplan, die Drahtbrücken sind vor der Bestückung aller anderen Bauteile zu legen



5) Bestückungsplan, J1 ist nur dann zu legen, wenn im GENIE ein stärkeres Netzteil eingebaut wurde, das den ECB-Bus mitversorgen soll

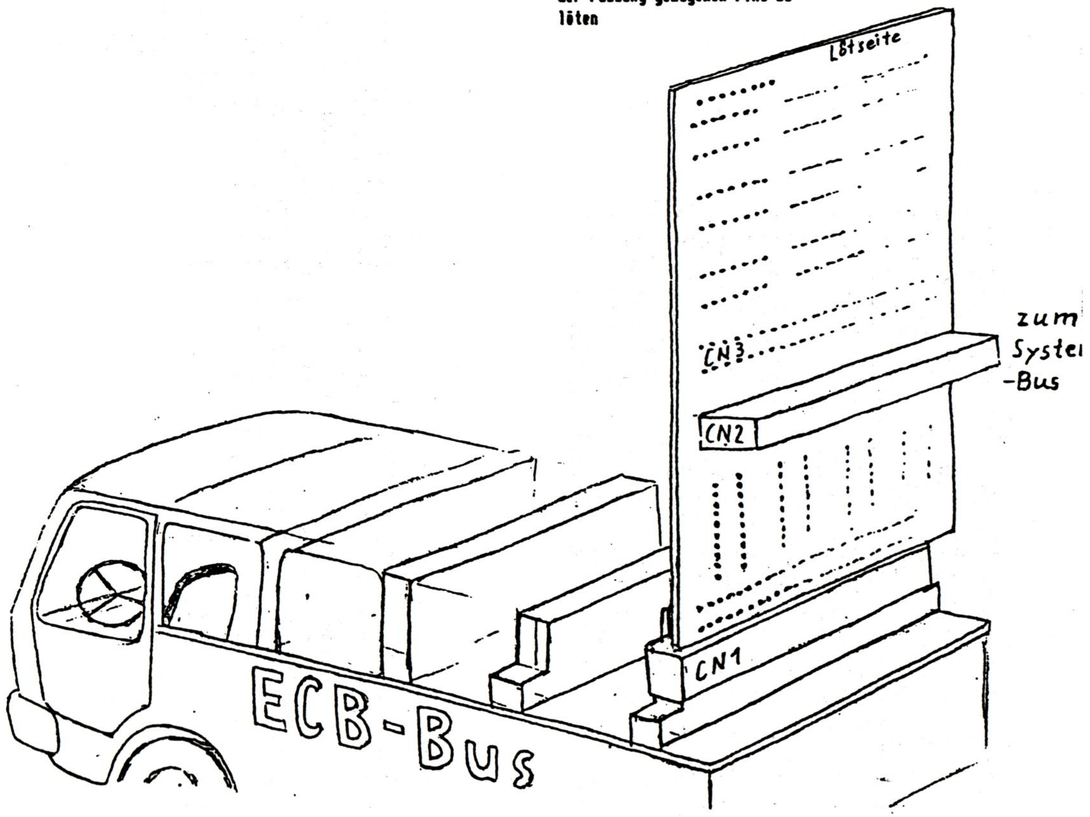
HEFT 23
Februar 1998



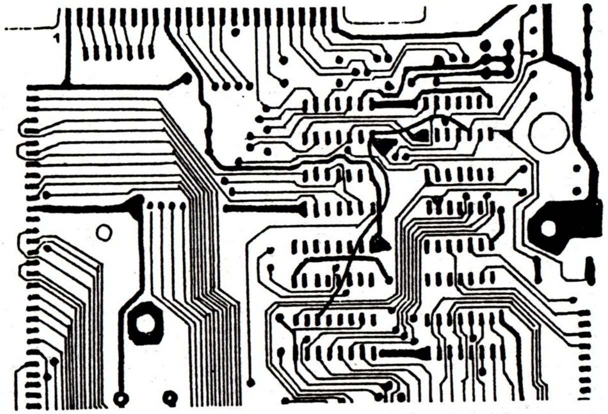
6) Beschaltung des Huckepack 74LS32 auf dem CPU-Board zum Ausblenden der ROMs

ROM1*	Z22, 74LS156, Pins 9,10	ROM1'*	Z10, 3001, Pins 20,21
ROM2*	" " " 11,12	ROM2'*	Z11, 3002, Pins 20,21
ROM3*	" " " 6,7	ROM3'*	Z12, 3003, Pins 20,21
ROM4*	R15, Seite zu Pin24 von Z13 (ROM4)	ROM4'*	Z13, 2716, Pin 20

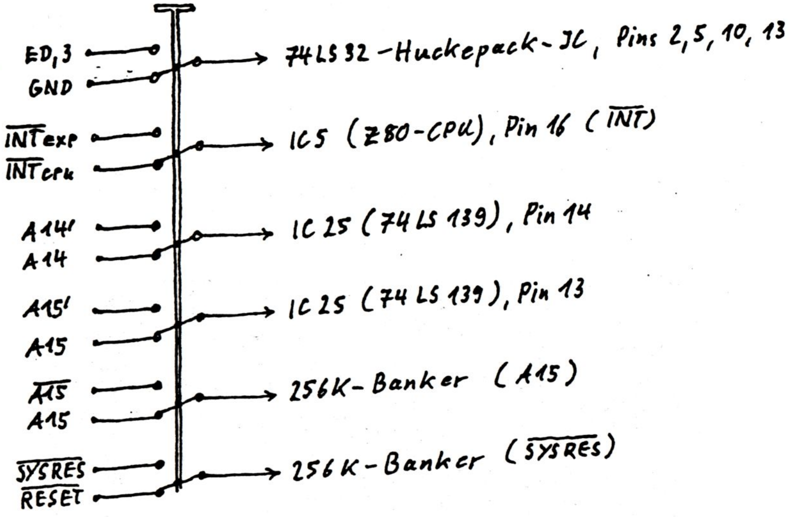
diese Signale sind an die aus der Fassung gebogenen Pins zu löten



7) Verbinden des ECB-Bus mit dem Systembus des Computers über die EingangsKarte



8) Verbinden von BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS*



9) Umschalten zwischen originalen GENIE-Signalen und steuerbaren Signalen der Eingangkarte

Suche:

Wer hat ein Basic- Programm (oder kann eines schreiben) das in der Lage ist, englische Texte ins deutsche zu übersetzen. Umgekehrt muß dies auch funktionieren. Nach Möglichkeit muß der Text auf Diskette abspeicherbar sein und auf einen Drucker ausgegeben werden können. Vokabeln gebe ich selber ein, da diese auf einen bestimmten Inhalt fixiert werden.

Zu verkaufen: Maus und TRS 80 M. 100

Im Zuge der Modernisierung meines Geräteparks habe ich mir vor Kurzem einen IBM-kompatiblen Portable (Toshiba T2100) zugelegt (keine Angst, ich bleibe weiterhin voll auf Tandy eingestellt). Leider arbeitet die Atari-Maus, die ich mir vor einiger Zeit zugelegt habe, nicht mit dem Toshiba zusammen. Aus diesem Grund habe ich folgende zwei Probleme zu lösen:

1. Ich benötige eine RS 232-Maus für den PC.
2. Meine Bessere Hälfte, zugleich Finanzminister im Hause Obermann, bewilligt die nötigen Mittel für eine solches Grautier nur, wenn ich zuvor die Atari-Maus verkaufe!

Aus diesem Grunde muß (will) ich mich von der im letzten Info beschriebenen Maus incl. Interface trennen. Die Schaltung funktioniert natürlich nicht nur am 4p, sondern auch mit allen anderen TRS 80- und Video Genie-Geräten. Wer also mal seine Grafikprogramme mit Maus fahren, oder auch nur mit einem solchen Peripheriegerät experimentieren will, sollte sich bei mir melden. Meine Preisvorstellung liegt so bei ca. 60 bis 70 Märkern für Maus incl. Interface, Verpackung und Porto.

Ebenfalls im letzten Info erwähnte ich, daß ich vor einiger Zeit ein Model 100 von Tandy erstanden habe. Da ich für diesen Computer, jetzt, nachdem ich den portablen PC habe, keine Verwendung mehr sehe, möchte ich auch dieses Gerät veräußern. Das Model 100 hat 24k RAM- und 32k ROM-Speicher, eingebaute Programme für Textverarbeitung, Adressen- und Terminverwaltung sowie Kommunikation über die eingebaute RS 232-Schnittstelle. Natürlich ist auch eine Kassetten- sowie eine Druckerschnittstelle vorhanden. Das 100'ter wird komplett mit 4 Akkus sowie einem englischen und einem deutschen Handbuch geliefert. Meine Preisvorstellung für dieses wirklich sehr schöne Gerät liegt bei ca. 450,-- DM (ev. 500,-- DM mit Kassettenrecorder).

Sowohl die Maus als auch das Model 100 sind 100 %'ig in Ordnung und ich gebe eine Übernahmegarantie von 14 Tagen. Wer sich für eines oder beide Geräte interessiert, kann mich entweder anrufen (ab Mitte April bis Ende Juni nur an Wochenenden) oder mir schreiben.

Hartmut Obermann
Schwalbacher Straße 6
Postfach 27
6209 Heidenrod / Kemel
Tel.: 06124 / 3913

BÖRSE -- BÖRSE -- BÖRSE

Drucker Gemini 10X

Wg. Umstieg auf 24 Nadeln stelle ich meinen 9-Nadel-Drucker einem Interessierten für schlappe DM 200,- (VB) zur Verfügung. Das Ding kann Grafik drucken und arbeitet bestens mit Tscrips und verwandten Programmen zusammen. Als Druckbeispiel mag mein CP/M-Artikel herhalten, der allerdings mit einem etwas älteren Farbband gedruckt wurde.

Schmidtke-80-Zeichen

Über die Karte habe ich ja schon viel gelästert, aber sie tut es noch. Aufgebaut ist sie für einsteckfertig für ein Genie IIs mit Z80-Prozessor. Bei mir arbeitet sie mit einem HD64180 (9 Mhz) nicht mehr zusammen. Sie müßte sich durch etwas Bastelei auch in eine für andere Genies taugliche Form bringen lassen, evtl. sogar für TRS-80. Für DM 50,- kann es jemand damit probieren. Ein angepaßtes Tscrips und die wichtigsten Routinen, um sich ein CP/M dafür selbst basteln zu können, sind vorhanden. Die Karte erledigt übrigens auch das Banking, wenn es sein muß, also kann ohne andere Zusätze CP/M gefahren werden.

Gerald Schröder, 04105 / 2602

VERKAUFE:
=========

- TANDON Floppy, 40 Tr, SS/DD, 80,-- DM
- TANDY Model 1, 150,-- DM
- Literatur:
  Personal Computer Lexikon, Markt und Technik Verlag
  Microcomputer Technik, Blomeyer
  Das DOS Buch, Hartmut Grosser
  Programmieren in Maschinensprache mit Z 80, C. Lorenz
  Programmieren mit TRS 80, M. Stubs
  Basic, G. Abeldt
  The First Book of 80 US
  Basic Disk I/O Faster and Better, L. Rosenfelder
  TRS-80 Disk and other Mysteries
  komplett 120,-- DM
- Texas Instruments, TI-58, mit Handbuch/Literatur, 60,-- DM
- TANDY Modell 4P, 900,-- DM

Klaus Hermann, Tel. 07127/70024

Betr. Computer für DDR
Ich habe kostenlos folgende Geräte abzugeben :

- Sharp PC 1211 mit Drucker und Kassettenspeicher, Betriebsanleitung und versch. Literatur.

- Junior-Computer in der Basisausführung mit Netzteil und Handbücher.

Sollte dafür Bedarf bestehen, so bin ich gerne bereit die Geräte dem Club zur weiteren Verwendung zuzusenden.

Das wäre alles für heute.
Mit freundlichen Grüßen,

Farbband für EPSON-Drucker

ab DM 10.95 in Weinheim erhältlich!

Bestellung (+Porto) bei „Ka-Jot"

(bis auf weiteres)

Achtung!!!

An alle Clubmitglieder, die am 23. April den Flohmarkt in Nürnberg besuchen, ich habe einen Tisch reservieren lassen und wenn jemand von euch ein Gerät zu veräußern hat, kann es mitbringen.

Ich habe außerdem ein Stringy-Floppy zu verkaufen mit I/O Program und vielen Kassetten. Preis Verhandlungssache.

Eckehard Kuhn

Z80 Hochintegrations CPUs in CMOS nutzen bestehende Entwicklungssysteme und Software

Unter den Bezeichnungen TMPZ84C011/013/015 bietet TOSHIBA **Hochintegrationsprodukte mit Z80-Kern** an. Die Produkte sind in 4 und 6 MHz-Version lieferbar und von −40 °C bis +85 °C spezifiziert. TMPZ84C011/015 haben ein 100-Pin Flat-Pack-Gehäuse, während TMPZ84C013 im 84-Pin PLCC gefertigt wird. Gemeinsam ist allen 3 Produkten neben der CPU ein On-Chip-Clock Generator und Controller.

TMPZ84C011 ist durch Ausstattung mit 5 Ports für I/O-intensive Aufgaben ausgelegt, während TMPZ84C013/014 mit 4 bzw. 2 USART-Kanälen für kommunikationsintensive Applikationen entwickelt wurden. Beide Produkte verfügen über einen integrierten Watch Dog Timer, TMPZ84C015 darüber hinaus über 2 Timer/Counter.

Bei der Entwicklung der Produkte wurde besonders Wert darauf gelegt, daß vorhandene Entwicklungssysteme für die Emulation verwendet werden können.

Für den Anwender bedeutet dies, daß der Einsatz dieser Produkte ohne größere Investitionen in Hard- und Software erfolgen kann.

Mit ihrem sicheren Gespür für Produkte, die der Markt in großen Stückzahlen braucht, haben die Japaner unseren bewährten Z80 um mehrere integrierte Zusatzfunktionen erweitert. Gedacht sind die neuen Schaltkreise für Controller-Anwendungen, bei denen der Mikroprozessor für fest vorgegebene Steuerungaufgaben mit Programmen aus dem ROM betrieben wird. Automaten, Hausgeräte und Verpackungsmaschinen sind Beispiele für den Controller-Einsatz. Werden größere Stückzahlen gefertigt, muß der Hersteller mit dem Pfennig rechnen. Sockel, Lötverbindungen und der Platz auf der Leiterplatte kosten Geld. Je mehr mechanische Verbindungen eine elektronische Schaltung hat, desto größer ist ihre Fehleranfälligkeit und um so schwieriger ist das Testen.

Unverständlich ist, daß Zilog die Entwicklung hochintegrierter Contoller-CPU's nicht schon längst betrieben hat. Andere Halbleiterhersteller, wie z.B. Intel mit seiner 8051-Serie, haben einen großen Marktanteil erobert, ohne daß Zilog die Chance mit seinem weltweit eingeführten Z80-uP und den dafür vorhandenen Entwicklungssystemen genutzt hätte. Der für einfachere Anwendungen vorgesehene Z8 ist nie so recht zum Zug gekommen und stand im Schatten seines erfolgreichen älteren Bruders. Nachdem der Z80 immer noch der in der größten Stückzahl verkaufte Mikroprozessor ist, kann man erwarten, daß Toshiba mit seinen hochintegrierten Schaltkreisen am Markt offene Türen einläuft. Hätte man diese Produkte schon vor drei oder vier Jahren (von Zilog) bekommen können, wäre heute wahrscheinlich der Z80 mit seinen Abkömmlingen der Industriestandard auch bei den Low Cost Controllern.

Heinrich Betz

Testbericht Drucker PEACOCK D 1018 (Baugleich mit Panasonic KX 1083) Nadeldrucker

Ende des vergangenen Jahres habe ich mich entschlossen meinen alten Drucker gegen ein besseres Exemplar einzutauschen.

Bei der Firma Computer-Service (W.Grundmann) konnte ich meinen alten Nadeldrucker in Zahlung geben und erstand ein Vorführmodell der Firma PEACOCK, den D 1018, der baugleich mit dem Panasonic KX 1083 ist. Mit diesem Gerät bin ich sehr zufrieden und die Testergebnisse der Zeitschrift "Warentest von Januar 1988" gaben dem Drucker die Note "GUT". Schwierigkeiten gab es zuerst mit den Farbandkassetten, aber mittlerweile habe ich ohne Schwierigkeiten vom Händler genügend Ersatz bekommen.

<u>Technische Daten des Nadeldruckers mit 9 Nadeln : (Epson kompatibel)</u>

Zeichenarten : Entwurf, Pica-Schönschrift, komprimierte Schrift, Elite Schönschrift, Proportional-Schönschrift.

Zeichensatz : 96 ASCII-Zeichen, 96 ASCII-Kursivzeichen, 32 Internationale Zeichen, 64 Block-Grafik-Zeichen, 132 IBM-PC-Sonderzeichen.

Punktmatrix : Punktdurchmesser 3/254 Zoll (0.3 mm)

Größe der Zeichen : Standardzeichen : 0.078(B)*0.095(H)Zoll
Hoch-/tiefgestellte Zeichen : 0.078(B)*0.053(H)Zoll

Anzahl der Zeichen pro Zeile :

Pica (Entwurf, Schönschrift)	80 Zeichen/Zeile	(10 Zeichen/Zoll)
Elite(Entwurf, Schönschrift)	96 Zeichen/Zeile	(12 Zeichen/Zoll)
Komprimiert	137 Zeichen/Zeile	(17 Zeichen/Zoll)
Pica gedehnt	40 Zeichen/Zeile	(5 Zeichen/Zoll)
Elite gedehnt	48 Zeichen/Zeile	(6 Zeichen/Zoll)
Kompimiert und gedehnt	68 Zeichen/Zeile	(8.5Zeichen/Zoll)

Druckgeschwindigkeit :

Entwurf Pica	180 Zeichen/s
Entwurf Elite	180 Zeichen/s
Schönschrift	33 Zeichen/s

Druckrichtung : Ausdruck von Text : bidirektional
Bitmuster : eine Richtung (links ==>rechts)

Zeit für einen Zeilenvorschub : ca 100 ms

Papierzufuhr : Traktorantrieb für Endlospapier
Friktionsantrieb für Enzelblattpapier

Kopf Lebensdauer : 100 Million Zeichen bei Entwurfs-Qualität

Farbband : Kassette mit nahtlosem Farbband

Wie geagt das war ein kurzer Steckbrief der technischen Daten. Öbrings lassen sich die einzelnen Schriftarten per Tastendruck von vorne bequem einstellen, ebenso die Länge des Blattformates.

Für meine Wordstar-Version habe ich mir eine eigene Druckeranpassung gemacht und so verfüge ich über zusätzlich weitere Schriftarten.

Hier eine kurze Öbersicht der Druckarten per Softwareansteuerung :

<u>PEACOCK-Druckeranpassung</u>

Worte und Leerzeichen unterstreichen
Schmalschrift
Kursiv
Riesenbreitschrift
fuer diverse Funktionen:
 Epson-Fettdruck
 Epson-Doppeldruck
 Epson-Elite; Epson-Pica-Schrift
 Proportionalschrift fuers Auge

schmale Breitschrift
kursiv Breitschrift
fette Breitschrift
kursiv schmale Breitschrift

Das war es wieder einmal in Kürze. Für interessierte, der Drucker kostet im Handes Zwischen 948,- und 1100,- DM.

Harald Mand

Reingefallen mit Rheintec

Liebe Clubfreunde,
sicher seit auch Ihr hier und dort darauf angewiesen, einmal Zubehör bei einem Händler zu kaufen. So war dies bei mir auch im letzten Jahr, als ich nach einer Speichererweiterung für meinen Rechner (PLANTRON XT/20 diesmal; mein GENIE steht im selben Zimmer) suchte und mich an eine Düsseldorfer Firma vertrauensvoll gewendet habe. Im Ergebnis ist jetzt mein Ärger groß, weil nichts klappte und die Firma am Schluß sich unfähig zeigte, wirklich zu helfen und unverschämter Weise auch die defekten Karten nicht zurück nehmen wollte. Ich muß für mich feststellen, daß ich bei dieser Firma nicht mehr kaufen werde.
Paul-Jürgen Schmitz
7. März 1988

Echte Programmierer meiden PASCAL

(Aus DATAMON 7/83, uebersetzt von Bernd Leeds)

Vor langer Zeit, in der goldenen Aera der Computer, war es noch einfach, die Maenner von den Memmen zu trennen (mitunter auch "Echte Maenner" und "Muesli-Fresser" genannt). Echte Maenner programmierten Computer und Muesli-Fresser liessen es bleiben. Ein echter Computerprogrammierer sagte Dinge wie "DO 10 I=1,10" oder "ABF1D", und der Rest der Welt quengelte "Computer sind mir zu kompliziert" oder "Ich kann zu Computern keine gefuehlsmaessige Bindung aufbauen - sie sind zu unpersoenlich". Dabei zeigt schon Remy Eyssen's Buch "Echte Maenner moegen kein Muesli" (Heyne TB 6290), dass Echte Maenner zu nichts und niemanden eine 'gefuehlsmassige Bindung' aufbauen, und dass sie auch keine Angst haben, unpersoenlich zu sein.

Aber die Zeiten aendern sich. Heute stehen wir einer Welt gegenueber, in der kleine alte Damen vollcomputerisierte Mikrowellenherde kaufen koennen, in der 12 Jahre alte Dreikaesehochs gestandene Maenner bei ASTROIDS und PACMAN sattmachen, und in der jeder seinen eigenen Homecomputer kaufen und sogar verstehen kann. Der Echte Programmierer ist gefaehrdet, von Studenten mit einem igITT 2020 (deutsche Version des ABFALL-II, Anm. d. Uebers.) im Gepaeck ersetzt zu werden!

Es gibt allerdings einige Unterschiede zwischen dem typischen, PAC MAN-spielenden Gymnasiasten und einem Echten Programmierer. Die Kenntnis dieser Unterschiede wird den Heranwachsenden ein Ziel geben nach dem sie streben koennen - ein Vorbild, eine Vaterfigur. Ausserdem schuetzt sie die Echten Programmierer vor Arbeitslosigkeit.

Der einfachste Weg, um einen Echten Programmierer zu erkennen, fuehrt ueber die von ihm benutzte Programmiersprache. Echte Programmierer benutzen FORTRAN. Muesli-Fresser benutzen PASCAL. Niklaus Wirth, der Schoepfer von PASCAL wurde einmal gefragt, wie man seinen Namen ausspreche. "You can either call me by name, pronouncing it 'Veert' or call me by value, 'worth'", antwortete er. Diese Bemerkung zeigt sofort, dass Wirth ein Muesli-Fresser ist. Der einzige Parameter-Uebergabe-Mechanismus, den Echte Programmierer akzeptieren, ist call-by-value-return (call-by-result, Anm. de. Uebers.), wie er in den IBM/370 FORTRAN-G- und -H-Compilern implementiert ist. Echte Programmierer brauchen schliesslich keine abstrakten Konzepte, um ihre Arbeit zu erledigen; sie sind vollkommen gluecklich mit einem Lochkarten-Stanzer, einem FORTRAN-IV-Compiler und einem Bier. Echte Programmierer erledigen Listenverarbeitung, Zeichenketten-Manipulation, Graphikdarstellungen (wenn ueberhaupt) und kuenstliche Intelligenz in FORTRAN. Was sie mit FORTRAN nicht machen koennen, machen sie in Assembler, was sie in Assembler nicht machen koennen, lassen sie veraechtlich liegen.

Akademische Computerspezialisten sind in den letzten Jahren auf's Abstellgleis der Strukturierten Programmierung geraten. Sie behaupten, dass Programme verstaendlicher werden, wenn bestimmte Sprachkonstrukte und Programmiertechniken benutzt werden. Sie koennen sich natuerlich nicht einigen, welche Konstrukte am besten geeignet sind, und die Beispiele, an denen sie ihren speziellen Standpunkt aufzeigen wollen, passen ausnahmenslos auf eine einzige Seite irgend eines obskuren Journals.

Als ich aus der Schule kam, dachte ich, ich sei der beste Programmierer der Welt. Ich konnte ein unschlagbares TIC-TAC-TOE-Spiel (Vier-in-einer-Reihe, Anm. d. Uebers.) schreiben, beherrschte 5 verschiedene Programmiersprachen und schrieb fehlerfreie 1000-Zeilen-Programme Dann kam ich in die Wirklichkeit. Meine erste Aufgabe bestand darin, ein 200.000-Zeilen-FORTRAN-Programm zu lesen, zu verstehen und um den Faktor 2 zu beschleunigen. Jeder Echte Programmierer wird einem versichern, dass die gesamte strukturierte Programmierung der Welt in einem solchen Fall nicht hilft - hier braucht man wirklich Talent.

Einige Beobachtungen zum Thema "Echte Programmierer und Strukturierte Programmierung":

- Echte Programmierer haben keine Angst vor GOTO's.
- Echte Programmierer schreiben 5 Seiten lange DO-Schleifen, ohne durcheinander zu geraten.
- Echte Programmierer lieben das arithmetische IF-Statement (das mit den 3 Ausgaengen, Anm. d. Uebers.), weil es den Code interessanter macht.
- Echte Programmierer schreiben selbstmodifizierende Programme, speziell, wenn sie damit in einer kleinen Schleife 20ns einsparen koennen.
- Echte Programmierer brauchen keine Kommentare; das Programm ist schliesslich selbstdokumentierend.
- Da FORTRAN strukturierte IF-, REPEAT UNTIL- oder CASE Anweisungen nicht kennt, braucht sich der Echte Programmierer nicht zu sorgen, dass er sie nicht benutzt. Ausserdem kann man sie noetigenfalls ueber "ASSIGNED GOTO's" simulieren.

Datenstrukturen sind in letzter Zeit in gewissen Kreisen populaer geworden. Wirth, der Muesli-Fresser, verfasste sogar ein ganzes Buch (Algorithmen und Datenstrukturen", Teubner 1975), in dem er behauptete, dass man Programme schreiben koenne, die auf Datenstrukturen aufbauen, statt es umgekehrt zu machen. Wie jeder Echte Programmierer weiss, gibt es nur eine wirklich nuetzliche Datenstruktur, das Array. Zeichenketten, Listen, Records und Mengen sind allesamt Sonderfaelle von Arrays und koennen auch so behandelt werden, ohne dadurch die Sprache zu verkomplizieren. Das Schlimmste an den ganzen schoenen Typen ist ausserdem, dass man sie deklarieren muss, waehrend Echte Programmiersprachen, wie man weiss, den Typ anhand des ersten Buchstabens eines maximal 6 Zeichen langen Bezeichners implizit festlegt.

Welches Betriebssystem der Echte Programmierer benutzt? CP/M? Gott bewahre! Das ist doch im Grunde ein Spielzeug-Betriebssystem. Selbst kleine alte Damen und Hauptschueler koennen CP/M benutzen und verstehen.

UNIX ist natuerlich schon viel komplizierter - der typische UNIX-Hacker weiss nie, wie das PRINT-Kommando diese Woche heisst - aber wenn man es genau nimmt, ist UNIX auch nur ein verherrlichtes Telespiel. Niemand arbeitet auf UNIX-Systemen an ernstzunehmenden Dingen - man schickt kleine Witzchen ueber USENET rund um die Welt, oder man schreibt ein neues Adventure-Spiel oder Forschungsberichte.

Nein, der Echte Programmierer benutzt OS/370. Ein guter Programmierer kann die Beschreibung des Fehlers IJK305I in seinem JCL-Manual (Job Control Language, Batch-Kommandosprache, Anm. d. Uebers.) finden und interpretieren. Ein sehr guter Programmierer erkennt die Fehler, ohne in sein Manual zu sehen. Ein wahrhaft ausserordentlicher Programmierer kann Fehler in einem 6-Megabyte-Hexdump finden, ohne hierfuer einen Taschenrechner zu benutzen.

OS/370 ist ein wirklich bemerkenswertes Betriebssystem. Mit einem einzigen falsch plazierten Leerzeichen kann man die gesamte Arbeit mehrer Tage zerstoeren, was die Wachsamkeit im Programmierteam ungemein foerdert. Der beste Weg zum System ist der Kartenstanzer. Zwar behaupten einige Leute, es gaebe ein Timesharing-System unter OS/370, aber nach sorgfaeltigen Nachforschungen bin ich zu dem Schluss gekommen, dass sie sich irren.

Welche Werkzeuge ein Echter Programmierer benutzt? Nun, theoretisch koennte er seine Programme ueber die Maschinenkonsole eingeben und laufenlassen. In den fuehen Tagen der Computerei, als Computer noch Maschinenkonsolen hatten, wurde dies auch gelegentlich getan. Der typische Programmierer wusste den System-Urlader Bit fuer Bit auswendig und tastete ihn ein, sobald er von seinem Programm zerstoert worden war. Damals war Speicher auch noch Speicher - der war nicht einfach leer, wenn der Strom ausfiel. Hauptspeicher von heute hingegen vergessen entweder Dinge, die sie behalten sollten, oder halten Informationen, die schon lange weg sein sollten. Aber zurueck zum Thema. Die Legende sagt, dass Seymour Cray, der Erfinder des Cray-I-Supercomputers und der meisten anderen Rechner von Control Data, selbst das erste Betriebssystem fuer die CDC 7600 an der Maschinenkonsole eingetastet hat, als sie das erste Mal eingeschaltet wurde. Cray ist selbstverstaendlich ein Echter Programmierer.

Einer der Echten Programmierer, die ich am meisten bewundere, arbeitete als Sytem-Programmierer fuer Texas Instruments. Eines Tages erhielt er ein Ferngespraech von einem Benutzer, dessen System mitten in einer wichtigen Arbeit abgestuertzt war. Der Typ reparierte dann den Schaden ueber's Telefon. Er brachte den Benutzer dazu, System-Tabellen in Hexadezimal zu reparieren und Registerinhalte ueber's Telefon durchzugeben. Die Moral von der Geschichte: obwohl ein Echter Programmierer normalerweise Kartenlocher und Schnelldrucker benutzt, kommt er im Notfall auch mit Maschinenkonsole und Telefon aus.

In einigen Firmen besteht die Programmeingabe allerdings nicht nur aus 10 schlangestehenden Ingenieuren, die auf einem 029-Locher warten. In meiner Firma z.B. steht kein einziger Kartenstanzer. Der Echte Programmierer muss in diesem Falle seine Arbeit mit einem Text Editor erledigen. Auf den meisten Rechnern stehen verschiedene Editoren zur Verfuegung, und der Echte Programmierer muss aufpassen, dass er einen erwischt, der seinen persoenlichen Stil wiedergibt. Viele Leute glauben, dass die besten Editoren der Welt am Xerox Pablo Alto Research Center geschrieben wurden und auf Alto- oder Dorado-Computern laufen. Ungluecklicherweise wuerde jedoch kein Echter Programmierer einen Computer mit einem Betriebssystem benutzen, das SmallTalk heisst, und sicherlich auch nicht ueber eine Maus mit einem Rechner kommunizieren.

Einige Konzepte der Xerox-Editoren sind mittlerweile in Editoren eingeflossen, die unter sinnvoller benannten Betriebssystemen arbeiten, so wie EMACS oder VI. Das Problem mit diesen Editoren ist, dass Echte Programmierer das Konzept des "Du kriegst, was Du siehst" fuer schlecht halten. Der Echte Programmierer will einen "Du hast es so gewollt, da hast Du's"-Editor, einen der kompliziert ist, skeptisch, leistungsfaehig, gnadenlos und gefaehrlich. TECO, um genau zu sein.

So wurde beobachtet, dass TECO-Kommandofolgen dem Leitungsrauschen aehnlicher sind als lesbarer Text. Eins der unterhaltsameren Spiele, die mit TECO moeglich sind, besteht darin, den eigenen Namen als Kommando einzugeben und zu raten, was dann passiert. So ungefaehr jeder moegliche Tippfehler kann dank TECO das gerade editierte Programm zerstoeren, oder schlimmer noch, kann kleine mysterioese Fehler in einstmals funktionierende Unterprogramme einbringen.

Aus diesem Grunde editieren Echte Programmierer nur sehr widerwillig Programme, die schon fast laufen. Sie finden es viel einfacher, den binaeren Objektcode direkt zu aendern, fuer gewoehnlich mit einem wundervollen Programm, das SUPERZAP heisst (auf Nicht-IBM-Rechnern entsprechend anders). Dies funktioniert so gut, dass viele laufende Programme auf IBM-Systemen keine Aehnlichkeit mit den ursprueglichen FORTRAN-Quellprogrammen haben. In einigen Fallen ist nicht einmal mehr das urspruengliche Quellprogramm vorhanden. Wenn dann der Zeitpunkt gekommen ist, so ein Programm zu aendern, wuerde kein Manager auch nur daran denken, einem geringeren als einem Echten Programmierer diese Arbeit zu uebertragen - kein Mueslifressender strukturierter Programmierer wuesste auch nur, wo er mit der Arbeit anfangen soll. Man nennt das Arbeitssicherungsmassnahme.

Hier eine Liste der wichtigsten Programmierhilfen, die der Echte Programmierer nicht benutzt:

- FORTRAN-Praeprozessoren wie MORTRAN oder RATFOR. Diese Haute Cuisine der Programmierung eignet sich hervorragend, um Muesli zu produzieren.
- Quellcode-orientierte Debugger. Echte Programmierer lesen Hexdumps.
- Compiler, die Code fuer Array-Indexpruefungen zur Laufzeit erzeugen. Sie ersticken jede Kreativitaet, zerstoeren die meisten der interessantesten Anwendungen der EQUIVALENCE-Vereinbarungen, und machen Aenderungen des Betriebssystems mit Hilfe negativer Indizes unmoeglich. Und schlimmer noch, solcher Code ist ineffizient.
- Programm-Pflege-Systeme. Ein Echter Programmierer haelt seine Software als Kartonstapel unter Verschluss, denn dies zeigt, dass der Besitzer seine wichtigsten Programme nicht unbewacht lassen kann.

Wo der typische Echte Programmierer arbeitet? Welche Art von Programmen derart talentierter Individuen wuerdig ist? Nun, man kann sicher sein, dass man nie einen Echten Programmierer beim Schreiben von Buchhaltungsprogrammen in COBOL erwischen wird, oder gar beim Sortieren der Abonnentenadressen des 'Spiegels'. Nein, ein Echter Programmierer braucht Aufgaben von weltbewegender Bedeutung.

Echte Programmierer arbeiten fuer das Los Alamos National Laboratory und schreiben dort Atomkriegs-Simulationen auf Cray-I-Supercomputern, oder sie arbeiten bei der National Security Agency und entschluesseln russische Funksprueche. Nur weil tausende Echte Programmierer fuer die NASA gearbeitet haben, waren 'unsere' Jungs eher auf dem Mond als die Kosmonauten. Die Computer im Space Shuttle wurden von Echten Programmierern programmiert, und auch die Betriebssysteme der Cruise Missiles der Firma BOEING wurden von diesen echten Professionals entworfen.

Einige der ehrfurchteinfloessendsten Echten Programmierer kennen das gesamte Betriebssystem der Pioneer- und Voyager-Sonden auswendig. Mit einer Kombination aus grossen, bodengebundenen FORTRAN-Programmen und kleinen, von Sonden mitgefuehrten Assemblerprogrammen vollbringen sie unglaubliche Kunststuecke der Navigation und Improvisation. So treffen sie nur 10 Kilometer grosse Fenster nahe Saturn nach 6 Jahren Flug durch den Weltraum, oder reparieren bzw. umgehen defekte Sensoren, Sender oder Batterien. Angeblich soll es einem Echten Programmierer sogar gelungen sein, in ein paar Hundert Byte ungenutzten Speichers innerhalb der Voyager-Sonde ein Mustererkennungsprogramm zu pressen, das einen neuen Mond des Jupiters suchte, fand und fotografierte.

Fuer die Galileo-Sonde ist vorgesehen, dass sie auf ihrem Weg zum Jupiter entlang einer schwerkraftgelenkten Bahn am Mars vorbeizieht. Diese Bahn fuehrt in einer Entfernung von 80 +/- 3 km an der Marsoberflaeche vorbei. Kein Mensch wuerde diese Art von Navigation einem PASCAL-Programm oder gar -Programmierer anvertrauen.

Viele Echte Programmierer dieser Welt arbeiten fuer die amerikanische Regierung, meist fuer das Verteidigungsministerium. So soll es sein. In letzter Zeit allerdings erscheinen dunkle Wolken am Horizont der Echten Programmierer. Es scheint, als haetten einige einflussreiche Muesli-Fresser im Verteidigungsministerium entschieden, dass in Zukunft

alle Verteidigungsprogramme in so einer Art von grosser vereinheitlichter Programmiersprache namens ADA geschrieben werden muessten. Lange Zeit schien es, als laege ADA's Bestimmung im Verstoss gegen alle Regeln der Echten Programmierung. Es ist eine Sprache mit Strukturen, Datentypen, strenger Typenbindung und Semikoli. Kurz, sie ist wie geschaffen um die Kreativitaet de typischen Echten Programmierers zu verkrueppeln.

Gluecklicherweise hat die jetzt vom DoD (Department of Defence) noch genuegend interessante Eigenschaften, um dem Echten Programmierer eine Annaeherung zu ermoeglichen: sie ist unglaublich komplex, sie enthaelt Moeglichkeiten, um mit dem Betriebssystem herumzumachen und Speicherbereiche neu zu verteilen, und Edgar Dijkstra mag sie nicht. Dijkstra ist wie man wissen sollte, der Autor von "GOTO's considered harmful", einem Meilenstein der Programmiermethodologie, der von PASCAL-Programmierern und Muesli-Fressern gleichermassen bewundert wird.

- Und ausserdem, ein zu allem entschlossener Echter Programmierer kann in jeder Sprache FORTRAN-Programme schreiben -

Der Echte Programmierer kann allerdings auch Kompromisse in Bezug auf seine Prinzipien eingehen und an etwas geringeren Aufgaben als der Vernichtung des Lebens arbeiten, sofern er dafuer entsprechend bezahlt wird. Viele Echte Programmierer schreiben z.B. Videospiele fuer ATARI, allerdings spielen sie nicht damit. Ein Echter Programmierer weiss, wie er die Maschine jedesmal schlagen kann, und damit ist es keine Herausforderung mehr. Jeder bei Lucas-Film ist ein Echter Programmierer, denn es waere doch verrueckt, das Geld von 50 Millionen STARS-WARS-Fans auszuschlagen.

Der Anteil der Echten Programmierern im Bereich der Computer-Graphics ist etwas niedriger als anderswo, was wahrscheinlich daran liegt, dass noch niemand irgendeinen Nutzen der Computer-Graphics entdeckt hat. Andererseits werden Computer-Graphics ueberwiegend in FORTRAN abgehandelt, daher gibt es einige Leute, die so das Schreiben von COBOL-Programmen vermeiden.

Im Allgemeinen spielt der Echte Programmierer wie er arbeitet - mit Computern. Er ist staendig darueber erheitert, dass sein Arbeitgeber ihn tatsaechlich fuer etwas bezahlt, was er nur so zum Spass sowieso tun wuerde - allerdings achtet er darauf, dieser Meinung nicht zu laut zu aeussern. Gelegentlich kommt der Echte Programmierer auch aus seinem Buero heraus, um sich ein wenig frische Luft und/oder zwei Bierchen zu genehmigen.

Hier daher einige Hinweise, wie man Echte Programmierer ausserhalb des Computerraumes erkennt:

- Auf Parties stehen Echte Programmierer in einer Ecke und diskutieren ueber Sicherheitsmassnahmen von Betriebssystemen und wie man um sie herumprogrammiert.
- Bei Fussballspielen vergleicht der Echte Programmierer die Ergebnisse mit seinem auf gruenliniertem Leporello-Papier gedruckten Computer-Simulations-Ergebnissen.
- Am Strand zeichnet der Echte Programmierer Flussdiagramme in den Sand.
- Ein Echter Programmierer geht in die Disco, um sich die Lichtorgel anzusehen.
- Bei Begraebnissen sagt der Echte Programmierer typischerweise: "Armer Hans-Helmut. Er war mit seinem Sortierprogramm schon fast fertig, als ihn der Herzinfarkt erwischt hat."
- Im Supermarkt besteht der Echte Programmierer darauf, seine Bierdosen selber ueber das Fenster des Barcodelesers zu schieben, weil er keinem Kassierer zutraut, dies beim ersten Versuch richtig zu machen.

In welcher Umgebung der Echte Programmierer am besten funktioniert? Nun, dies ist eine sehr wichtige Frage fuer die Manager von Echten Programmierern. Wenn man bedenkt, wie teuer es ist, einen von ihnen ihm Betrieb zu halten, dann sollte man ihn oder sie in eine optimale Arbeitsumgebung versetzen.

Der typische Echte Programmierer lebt vor einem Computerterminal. Rund um dieses Terminal liegen Ausdrucke von jedem Programm, an dem er je gearbeitet hat, sie stapeln sich grob chronologisch geordnet auf jeder ebenen Flaeche des Bueros. Im Zimmer verteilt finden sich ueber ein Dutzend mit kaltem Kaffee mehr oder weniger gefuellte Tassen. Gelegentlich schwimmen Zigarettenkippen darin herum, in einigen Faellen auch Reste von Orangenschalen. Irgendwo liegen Kopien des OS JCL-Manuals und der "Principles of Operation" herum. Ueber den Boden verteilt liegen Reste der Verpackungen von gefuellten Keksen (der Typ, der schon in der Fabrik so furztrocken gebacken wird, dass er auch bei laengerem Liegen im Automaten nicht schlechter wird).

Schliesslich, in der linken oberen Schublade des Schreibtisches, unter der Schachtel mit Muntermachern, liegt eine Schablone fuer Flussdiagramme, die sein Vorgaenger dort vergessen hat. Echte Programmierer schreiben Programme und keine Dokumentation; das ueberlaesst man den Typen von der Wartung.

Der Echte Programmierer ist in der Lage, 30, 40, ja sogar 50 Stunden in einem Rutsch zu arbeiten, und das unter hohem Zeitdruck. Genaugenommen mag er es so am liebsten. Schlechte Antwortzeiten regen den Echten Programmierer nicht auf - sie geben ihm die Chance, zwischen 2 Kommandos ein bisschen Schlaf zu ergattern. Wenn die Planung nicht genug Zeitdruck bereithaelt, dann tendiert der Echte Programmierer dazu, seine Arbeit herausfordernder zu machen, indem er sich die ersten neun Wochen mit einem kleinen, aber sehr interessanten Teil des Problems befasst, um dann in der letzten Woche seine Aufgabe in zwei oder drei 50-Stunden-Marathonsitzungen zu beenden. Dies beeindruckt nicht nur den Manager, sondern schafft gleichzeitig eine hervorragende Entschuldigung fuer das Fehlen einer Dokumentation.

Und ueberhaupt: kein Echter Programmierer arbeitet von 9 bis 5, ausser denen von der Nachtschicht. Echte Programmierer tragen keine Schlipse. Echte Programmierer tragen keine hochhackigen Schuhe. Echte Programmierer kommen zur Arbeit, wenn andere zum Mittagessen gehen. Ein Echter Programmierer vergisst vielleicht den Vornamen seiner Angetrauten, aber niemals den Inhalt der gesamten ASCII- (oder EBCDIC-) Tabelle. Echte Programmierer koennen nicht kochen. Da Supermaerkte um 3 Uhr morgens selten geoeffnet sind, muessen sie sowieso von Kaffee und Keksen leben.

Die Zukunft betrachtend machen sich eine ganze Reihe von Echten Programmierern Sorgen, dass die juengste Programmierergeneration nicht mehr mit der gleichen Lebensperspektive aufwaechst wie sie selbst. Viele der Juengeren haben noch nie eine Computer mit einer Maschinenkonsole gesehen. Kaum ein Schulabgaenger kann heute noch hexadezimal rechnen, ohne einen Taschenrechner zu benutzen. Die Programmierung durch symbolische Debugger oder Texteditoren, die Klammern zaehlen, und benutzerfreundliche Betriebssysteme. Und das Schlimmste ist, einige von ihnen werden auf die Menschheit losgelassen ohne je FORTRAN zu lernen! Sind wir dazu verdammt, eine Industrie von UNIX-Hackern und PASCAL-Programmierern zu werden?

Nun, aus meiner Erfahrung heraus glaube ich behaupten zu duerfen, dass das Schicksal den Echten Programmierern wohlgesonnen ist. Weder OS/370 noch FORTRAN zeigen irgendwelche Symptome des Aussterbens, trotz aller Anstrengungen der PASCAL-Programmierer. Selbst subtilere Tricks wie das Hinzufuegen strukturierter Schleifen zu FORTRAN sind fehlgeschlagen. Sicher, einige Computerhersteller liefern FORTRAN-

77-Compiler, aber jeder einzelne von ihnen laesst sich ueber eine einzige Compiler-Option in einen FORTRAN-66-Compiler verwandeln - mit DO-Schleifen wie von Gott geschaffen.

Selbst UNIX scheint fuer den Echten Programmierer nicht mehr so schlecht zu sein wie frueher. Die neueste UNIX-Version hat das Potential eines Betriebssystems, das eines Echten Programmierers wuerdig ist. Sie hat zwei verschiedene, leicht inkompatible Benutzer-Schnittstellen, einen geheimnisvollen und komplizierten Teletype-Treiber und virtuellen Speicher. Und wenn der Echte Programmierer die Strukturierung ignoriert, kann er sich sogar mit C anfreunden. Schliesslich gibt es keine Typenbindung, Bezeichner sind sieben (zehn? acht?) Zeichen lang und man hat Zeiger als Bonus. Das ist, als haette man die besten Teile von FORTRAN und Assembler vereint, von den kreativeren Moeglichkeiten des #define ganz zu schweigen.

Nein, die Zukunft ist nicht voellig schlecht. So hat sich in den vergangenen Jahren die populaere Presse sogar ueber clevere neue Brut von Computer-Schraten und -Hackern geaeussert, die Plaetze wie Stanford oder das MIT zugunsten der Wirklichkeit verlassen haben. Allen Anzeichen nach lebt der Geist der Echten Programmierer weiter in diesen jungen Maennern und Frauen. Und solange es schlecht beschriebene Ziele, bizarre Fehler und unrealistische Zeitplaene gibt, solange wird es Echte Programmierer geben, die bereit sind einzuspringen und das Problem zu loesen, und die sich die Dokumentation fuer spaeter aufheben.

LANG LEBE FORTRAN !

ED POST, WILSONVILLE

*** Brought to you by BIRNE for TORNADO BBS ***

PublicDomain für Modell 4/4P
============================

Im letzen INFO (Nr. 22) wurden mehrere PublicDomain Programme für das Modell 4/4P vorgestellt. Hierbei ist zu berücksichtigen, daß ein Teil der Programme mit doppelseitigen Laufwerken nicht arbeitet. Im einzelnen sind dies:
RESTORE/CMD - RUN4/BAS - ZAP4/CMD - SPACEMAP/CMD

Das Programm WD/CMD (Windows) arbeitet nur mit der US-TRSDOS Version.

Klaus Hermann

Hallo liebe Clubkameraden !

Wie im letztem Club-Info schon beschrieben, haben wir jetzt auch eine große CP/M 2.2-Software Bibliothek !

Ich habe bislang nur drei Zuschriften bekommen !!! Ist denn die CP/M 2.2-Software nicht gefragt ?

Damit Ihr einen kleinen Eindruck von der ganzen Biblothek bekommt, habe ich das Inhaltsverzeichnis der CP/M-USER-GROUP ausgedruckt:

Disk-Nr.:
----
01 - "Various CP/M Utilities - 0"
02 - --> see CPMUG Vol. 10 11
03 - "BASIC-E Games and Programs"
04 - "ACTOR, ML80 and Examples of FORTRAN-80 Code"
05 - "BASIC-E Programs and Microsoft BASIC programs"
06 - "Maillist Program and CP/M Utilities"
07 - "PILOT - Programmed Enquiry Learning Teaching"
08 - "Various CP/M Utilities - 1"
09 - "General Ledger from Interface Age Vol.2 No.10"
10 - "LLBASIC: Lawrence Livermore BASIC interfaced to CP/M"
11 - "TINIBASIC and Processor Technology BASIC/5"
12 - "PILOT Interpreters interfaced to CP/M"
13 - "BASIC-E, CBASIC Microsoft BASIC Programs"
14 - "Various CP/M Utilities - 2"
15 - "Utilities and non-BASIC Games"
16 - "Assemblers, Other Utilities and FOCAL"
17 - "Utilities, Denver Tiny BASIC non-BASIC Games"
18 - "Maths Routines, Monitors and CASUAL (no CP/M I/O yet)"
19 - "Various CP/M Utilities - 3"
20 - "BASIC-E, CBASIC Programs Pictures"
21 - "Microsoft BASIC Games Programs"
22 - "Monsterous StarTrek Games in BASIC"
23 - "STOIC - Stack Oriented Interactive Compiler"
23B- "STOIC - Overflow Programs From Vol 23" (in CPMUG 25)
24 - "CP/M Utilities, Macro Libraries RATFOR"
25 - "Various Assembler Utilites"
26 = "Microsoft BASIC FORTRAN Games and Utilities"
27 - "Microsoft BASIC Games"
28 - "BASIC-E Games Utilities, ALGOL-M"
29 - "Assembler Games Utilities, BASIC-E Source"
30 - "BASIC-E Ver 1.4 Source Code in PL/M"
31 - "Tarbell BASIC - 1"
32 - "Tarbell BASIC - 2"
33 - "Search and Rescue Programs"
34 - "SAM-76 Language"
35 - "FELIX - Graphics Animation System"
36 - "Assemblers, Editors and Utilities"

37 - "CBASIC2 Games, Utilities and CAI Programs"
38 - "Speed Up and Tarbell Disk Controller Utilities"
39 - "Music Programs for SOL CP/M"
40 - "Various Utilities, Disk Catalog System"
41 - "Ham Radio, Chess and FORTRAN programs"
42 - "Disassemblers, Diablo Clock Routines"
43 - "OSBORNE A/P,A/R Business Software I"
44 - "OSBORNE G.Ledger, Budget,ROBO, Business Software II"
45 - "OSBORNE Payroll Business Software III"
46 - "CP/M utilities, CPM LABEL, DU, DIR, REASM"
47 - "CP/M Utilities, MODEM, Copy, BMAP, PROM"
48 - "BDS C Sampler Disk, Comp. by Leor Zolman"
49 - "Rational FORTRAN (RATFOR), precompiler"
50 - "PASCAL PASCAL Compiler , Prgs. for UNIX"
51 - "STAGE2 Macro Processor by Dick Curtiss"
52 - "Copyfast vers. 3.5, BATCH/VARBATCH"
53 - "BDS-C Adventure disk"
54 - "BASIC Games, CAI Programs"
55 - "Adventure run-time disk"
56 - "Adventure source code implemented for CP/M"
57 - "Expanded Adventure supercedes CPMUG 56"
58 - "Miscellaneous CP/M Utilities"
59 - "8080 8085 memory and ICOM Controller diagnostics"
60 - "6502 Simulator, 6502 Zapple Monitor"
61 - "Bulletin Board, File Transfer CP/M Utilieties"
62 - "PASCAL Comunication programs"
63 - "Misc. CP/M Utilities"
64 - "Games, Disassembler, North Star Basic, CDOS simulator"
65 - "MITS-CP/M file conversion, PHELP, Utilities, FIG-Forth"
66 - "HELP File system for MBASIC, M80, CBASIC other"
67 - "CPMUG CATALOG disk vol. 01 - 42 with comments"
68 - "Miscellaneous CP/M Utilities"
69 - "Miscellaneous CP/M Utilities"
70 - "Miscellaneous CP/M Utilities"
71 - "Miscellaneous PASCAL-Z Programs - 1"
72 - "Miscellaneous PASCAL-Z Programs - 2"
73 - "Miscellaneous PASCAL-Z Utilities - 3"
74 - "Miscellaneous PASCAL-Z Utilities - 4"
75 - "PASCAL-Z Programs - 5, MBASIC Disassembler, Date routines"
76 - "Miscellaneous PASCAL-Z Utilities - 6"
77 - "PASCAL-Z Utilities - 7, Datebase System, Disk sort Program"
78 - "Miscellaneous CP/M Utilities"
79 - "MODEM Programs for MMI SMARTMODEM"
80 - "CROMENCO structured BASIC Programs, Mail list, Spelling"
81 - "Miscellaneous CP/M Utilities, Submit, Editor, Hard Disk"
82 - "North Star BIOS Routines"

Wie Ihr seht, gibt es eine ganz schöne Auswahl !

Wenn Ihr wollt, dann könnt Ihr noch mehr Informationen auf Disketten anfordern. Das Diskettenformat müßt Ihr allerdings mit angeben !

Bis Bald

Andreas

CLUB 80 — Bücherei

Hallo Freunde,

viel erfreuliches gibt es wieder von der Bücherfront zu berichten. Zunächst wäre da zu vermelden, daß es Jens endlich gelungen ist, zwei Sammelordner mit den "alten" Infos Nr. 1 bis 15 zusammenzustellen. Die Infos sind auf Karton vervielfältigt, so daß sie (hoffentlich) auch mehrere Einsätze auf dem Kopierer überleben.

Die Ordner werden wohl vornehmlich für Neuzugänge interessant sein, die sich noch nicht die älteren Ausgaben des CLUB 80-Info besorgt haben (oder besorgen konnten). Die Ausleihzeit beträgt, da eine rege Nachfrage zu erwarten ist, maximal drei Wochen. Die Besteller werden, wenn beide Ordner unterwegs sind, in eine Warteliste aufgenommen, so daß zuerst malt, wer zuerst kommt! Selbstverständlich geht das Porto zu Lasten des Ausleihers, so daß der Bestellung gleich drei DM (ev. auch in Briefmarken) beigelegt werden sollten.

Die Infos 16 bis 21 werden demnächst als Sammelband verfügbar sein, so daß Neuzugänge immer auch das schon dagewesene zur Verfügung haben. Ich möchte die Ausleiher an dieser Stelle bitten, die Ordner möglichst schonend zu behandeln, damit sie noch möglichst lange den Neuen zeigen können, wie aktiv ein Club mit nur ca. 60 Mitgliedern sein kann (und hoffentlich noch lange sein wird)!

Ebenfalls erfreulich ist die Tatsache, daß der Heise-Verlag wieder mal seine Lager räumt und der CLUB 80 dadurch sehr hochwertige Bücher zu äußerst günstigen Preisen erwerben kann. Welche der bestellten Bücher tatsächlich geliefert werden ("nur solange Vorrat reicht" stand auf dem Angebot), werde ich euch erst im nächsten Info mitteilen können. Wenn aber alles von einer solchen Qualität ist wie das Buch, das schon jetzt als Neuanschaffung in der Bibliothek auzuleihen ist, können wir sehr zufrieden sein.

Erst gestern habe ich, ganz kurzfristig und durch Zufall, eine Neuerwerbung für die CLUB 80-Bücherei einkaufen können. Es handelt sich dabei (Model- 1 und III- sowie Video Genie-Besitzer bitte nicht neidisch werden) um das Model 4/4p Technical Reference Manual. Dieses Buch vereint technische Informationen über das Gerät und über das Betriebssystem TRSDOS 6.x. Von "Model 4/4p Theory of Operation" über die genaue Beschreibung der verwendeten Spezial-IC's (CRT-Controller MC6835, Baud Rate Generator BR1941, Floppy Disk Controller FD 179x-02 usw.) bis zur detaillierten Erläuterung der Diskettenorganisation und den, für Assemblerprogrammierer sehr wichtigen, SVC's (Supervisor Calls) ist alles vorhanden, was das Herz begehrt.

Wieder einmal ans Tageslicht befördern, und euch in Erinnerung rufen, will ich die Sammelmappen, die Günter Wagner in seiner Zeit als Vorsitzender einmal zusammengestellt hat. Es handelt sich dabei um Kopieen von Artikeln aus verschiedenen Computerzeitschriften (von ELCOMP bis Computer persönlich), die sich alle mit dem TRS 80 bzw. Video Genie beschäftigen. Die Sammlung dürfte vor allem für die Mitglieder interessant sein, die sich zur Erscheinungszeit der Artikel (1980 bis 1983) noch nicht mit Computern beschäftigt haben.

Damit genug zu diesem Thema. Bestellen könnt ihr die Sammelmappen und Bücher direkt bei mir (Rückporto nicht vergessen)

Hartmut Obermann

- I M P R E S S U M -

1. Vorsitzende	Hartmut OBERMANN Schwalbacher Straße 6 6209 Heidenrod 1 ☎ 06124 /3913
2. Vorsitzende	Gerald SCHRöDER Am Schützenplatz 14 2105 Seevetal 1 ☎ 04105 /2602
Hardwarekoordinator	Eckehard KUHN Im Dorf 14 7443 Frickenhausen 1 ☎ 07022 /45417
Diskothekar	Werner FöRSTER Christoph-Krebs-Straße 9 8720 Schweinfurt ☎ 09721 /21841
Redaktion	Jens NEUEDER Panoramastraße 21 7170 Michelbach /Bilz ☎ 0791 /42877
Autoren	Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.
Druck	Peter Spieß Trugenhoferstraße 27 8859 Rennertshofen 1 ☎ 08434 /454
Bankverbindung	des CLUB 80 Postgirokonto: Peter STEVENS, Sonderkonto CLUB 80 Konto-Nummer: 285 491 - 465 Postgiroamt: Dortmund BLZ: 440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

S C H L U SS

Hallo Club-80er,

zum Abschluß des ersten 1988'er Info's darf ich mich wieder zu Wort melden. Diesmal bin ich mit dem INFO fast termingerecht fertig geworden. Leider habe ich auch eine negative Nachricht für Euch. Dazu gleich, zuerst aber meinen Dank an die fleisigen Autoren unserer INFO.

- Wieder sind wir um die hundert Seiten stark geworden.
- Macht weiter so!

Wie Ihr im Info schon habt lesen können, hat der Peter Spies seine Offsetdruckerei aus beruflichen Gründen aufgegeben. Ich hoffe, daß es mir gelingt rechtzeitig Ersatz für Ihn zu finden, so daß Ihr weiterhin Euer INFO bekommen könnt. Gleichzeitig bitte ich Euch um eure Mithilfe. Vielleicht hat jemand von Euch einen "Drucker" in der Hinterhand? -Offset natürlich!- Wer mir helfen kann melde sich bitte bei mir.

Wobei ich auch schon beim nächsten Thema bin.
Die Redaktionsadresse ändert sich !!
Ich bin zwar noch nicht "körperlich" umgezogen aber "postmäsig". Der Nachsendeantrag läuft für ein halbes Jahr. Trozdem bitte ich Euch alle, meine neue Adresse in Eure Listen aufzunehmen und ab sofort an den neuen (Redaktions-)Briefkasten zu schreiben:

Jens NEUEDER
Rudolf-Then-Straße 32
7178 Gschlachtenbretzingen

Telefonnummer bleibt gleich! Ihr könnt mich weiterhin unter 0791 / 42877 erreichen.

So, das war nun einmal wieder. Als Beilage findet Ihr in diesem INFO ein Infoinhaltsverzeichnis bis 1. 3. 1988 von KaJot.
Als weiterer Anhang ist noch ein Kartenauszug dabei. Ich bin sicher, daß Ihr damit die beste "Anflugschneise" nach Idstein finden werdet.
Eine neue Adressenliste habe ich diesmal nicht mit beigefügt. Es gibt noch einige säumige Zahler unter uns. Bis zum nächsten mal ist das sicher geklärt. Dann kommt auch eine neue Liste.

Bis zum Clubtreffen oder nächsten INFO grüßt Euch

Jens

PAGE 1 - INHALTSVERZEICHNIS DER INFOS BIS 1.3.88

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
ASCII	ASCII-TABELLE IMMER ZUR HAND	SOPP	19/19
ASSEMBLER	KLEINSCHRIFT fuer den EDTASM+	HEIDENREICH	20/37
	SCHUETZEN VON ASSEMBLERPROGRAMMEN	KONRAD J.	5
	HALBAUTOMATISCHE FESTLEGUNG VON ORG	KONRAD, J.	9/3
	MEIN DRITTES ASSEMBLERPROGRAMM	MUEHLENBEIN	16/33
	MEIN ERSTES ASSEMBLERPROGRAMM	MUEHLENBEIN	14/9
	"INPUT"	SCHROEDER	11/5
	ASSEMBLER CONTRA LISP	SCHROEDER	17/19
	DIE WERKZEUGE DES ASSEMBL.PROGRAMMIERERS	SCHROEDER	21/9
	EINIGE ASSEMBLER-ROUTINEN	SCHROEDER	14/17
	LMOFFSET with DISABLE DOS	SCHROEDER	9/9
	WERKZEUGE DES ASSEMBLER-PROGRAMMIERERS	SCHROEDER	21/9
	LPRINT ALLES einfacher! (Teil 3)	SOPP	9/11
	SCHOENER LISTEN MIT ZEUS	SOPP	18/41
	WER HAT ANGST VOR ASSEMBLER?	SOPP	5
	ZEUS MIT 64 UND 80 ZEICHEN	SOPP	18/36
	BASIC RUFT ASSEMBLER	WOLLSCHLAEGER	21/11
	TIP FUER SCHNELLERES ARBEITEN MIT EDTASM	ZWICKEL	6/67
BANKER	SYSCOPY1 und RAMDISK f.HELMUT's BANKER	SCHROEDER	16/32
BANKING	256K-RAM fuer Z80-SYSTEME	BERNHARDT	14/91
BASIC	EXTENDED BASIC(RESTOREn u. LPRINT alles)	OBERMANN	11/33
	EIN MYSTERIOESER BASIC-BEFEHL: IsA	SOPP	5
BERICHT	GENIE IIs - Erfahrungsbericht	SCHROEDER	9/67
BILDFORMAT	FUTTER fuer GENIE IIIs CRTC	HEIDENREICH	20/22
BILDSCHIRM	BILDSCHIRMAUSGABE mit INVERSEN ZEICHEN	?	12/79
	BIN ICH STRAHLENGEFAEHRDET?	BOIKAT, Ute	20/109
	DAS PROJEKT (FERNSEHER --> MONITOR)	MUELLER, K.	18/61
	NOCH EIN BILDSCHIRMSCHONER	SCHMID, A.	22/26
	SCHWIMMENDE BILDER	SCHMID, A.	22/66
	80-ZEICHENKARTE X80, SEDIT und TSCRIPS	SCHROEDER	13/17
	DIE 80-ZEICHEN-KARTE	SCHROEDER	12/63
	EIN SCHONER fuer die BILDROEHRE	SOPP	20/20
	EIN WEITERER BILDSCHIRMSCHONER	SOPP	22/28
	FUTTER fuer den CRTC	SOPP	19/40
	MEHR FERNSEHEN fuer's GELD (f.GENIE IIIs	SOPP	14/21
	NOCH MEHR FERNSEHN fuers GELD	SOPP	19/39
	VIDHEX - HEXANZEIGE d.BILDSCHIRMS m.HRG	SOPP	9/13
CHIP	BEI SIEMENS HEISST ES: CHIP CHIP HURRA!	ZEITSCHRIFT	18/76
	ERSTER MEGABIT-CHIP IN SERIE	aus Zeitschrift	18/18
	DER LEIDENSWEG DER CMOS-CHIPS	aus c't	19/67
	NANOHENRY BLOCKIEREN DEN FORTSCHRITT	aus c't	19/70
CODE	BASICODE-INTERFACE	?(aus ELEKTOR)	3
	BASICODE-2	?(aus Elektor)	3
COMP.-MUSIK	MEIN ZWEITES ASSEMBLERPROGRAMM	MUEHLENBEIN	15/15
COMP.WECHSEL	UNTERSCHIEDE (zw. MODEL 1 und 3 bzw. 4)	OBERMANN	14/55
COMPILER	BASCOM-KURZANLEITUNG	MUEHLENBEIN	13/38
	RPNL - DIE ANDERE ART ZU PROGRAMMIEREN	MUELLER, KURT	15/57
	RPNL-BESCHREIBUNG	MUELLER, KURT	16/37
	CALL mit BASCOM	OBERMANN	12/33
COMPUTER	MODEL 4p ROM-IMAGE-LOADER	OBERMANN	15/43
	TANDY<--> SCHNEIDER	OBERMANN	17/49
	CP/MAC-eine BRUECKE zw.G-/NEWDOS u.CP/M	RENSCH	18/44
	WIE PHOENIX AUS DER ASCHE (4 COMP.in 1)	SCHROEDER	17/9
	DEIN GIIIs... (NACHTRAG)	SOPP/BERNHARDT	17/54
	TRS-80 MODELL 4p, DER SCHOENSTE PORTABLE	WOLLSCHLAEGER	12/69
	TANDY - BETTER AGAIN	ZEITSCHRIFT	16/40
	32 BIT zum NIEDRIGPREIS (TANDY 4000)	aus - ? -	22/79
CONTROLLER	FLOPPYCONTROLLER-KOMPATIBILITAET	RETZLAFF	19/47
CP/M	WAS HABT IHR GEGEN CP/M ?	BERNHARDT	21/53
	GENIE, CP/M UND 80 ZEICHEN UND HRG???	HELD	21/43
	CMD ==> COM ohne CPMac	SCHROEDER	21/23
	F.HEIMWERKER: WIE BAUE ICH MIR EIN CP/M?	SCHROEDER	21/25

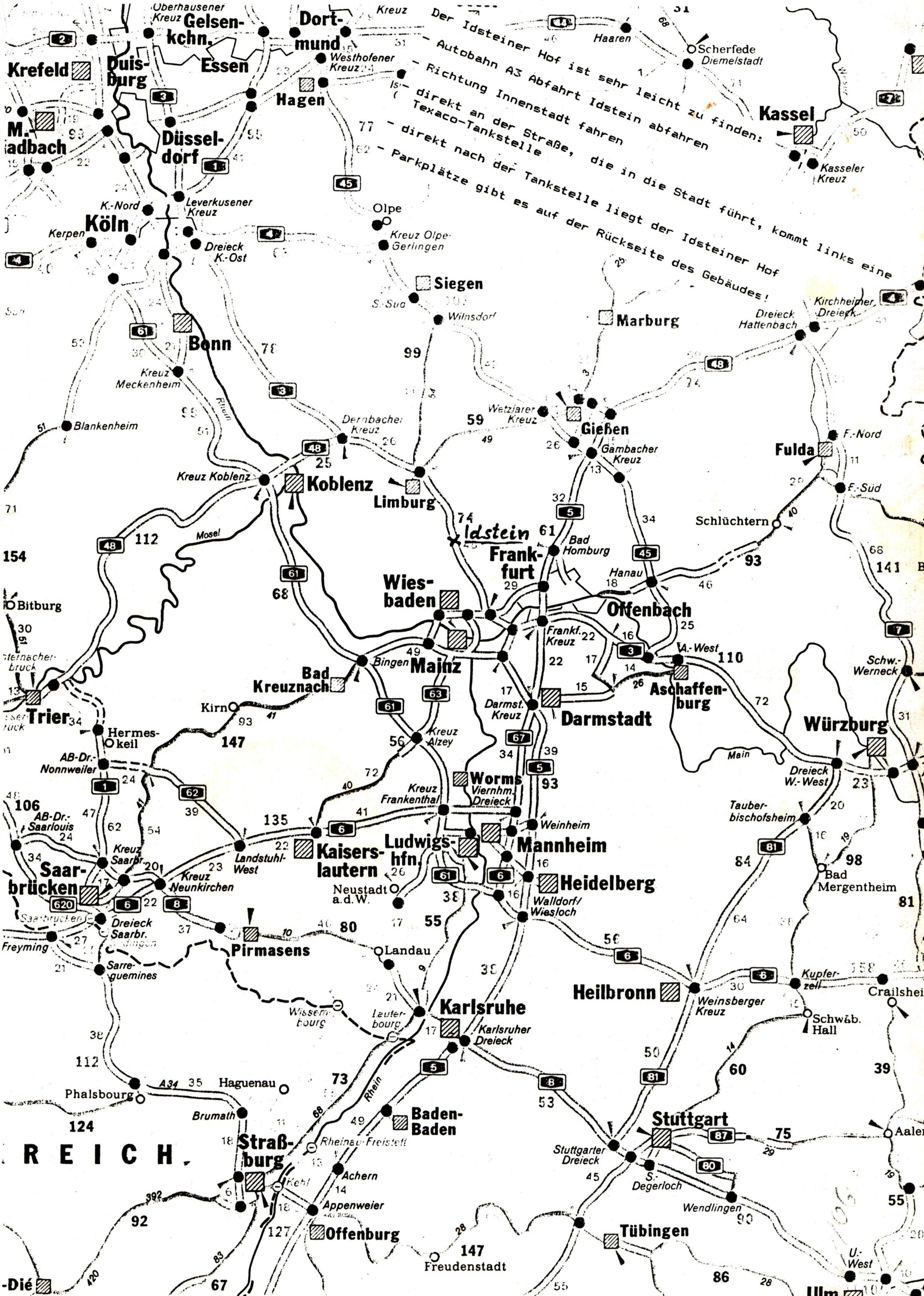
PAGE 2 - INHALTSVERZEICHNIS DER INFOS BIS 1.3.88

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
CP/M	Wie baue ich mir ein CP/M ?	SCHROEDER	21/25
DATEI	INDEX-DATEIEN	HERMANN	19/15
DATENBANK	BDBS - BASIC DATENBANK SYSTEM (MODEL 4)	HERMANN	22/37
	dBASE II - eine Einfuehrung	MAND	21/33
DATENUEBERTR	VIDEODAT 300	?	13/51
DDE	DDE fuer das GENIE IIIs	SOPP	12/17
DEF.FUNCTION	BASIC-PROGRAMMIERBARE FUNKTIONEN	HERMANN	19/26
DENKAUFGABEN	MATHEMATIK fuer FEINSCHMECKER	WEBER	20/101
DFUe WDR	DATENUEBERTRAGUNG p.FERNSEHBILD/WDR-CLUB	EDDE u.HERIBERT	12/62
DOS	R-BEFEHL MSDOS-LIKE (ein Editor f. DOS)	HEIDENREICH	20/44
	LS-DOS 6.3 - KOPIERSCHUTZ	HERMANN	21/55
	EXTENDED NEWDOS 80	OBERMANN	10/5
	NEWDOS/80+10 VERSION 2.4 (26.10.85)	OBERMANN	11/25
	SUPERDOS	OBERMANN	2
	DIR/SYS ODER INHALT/SYS?	RENSCH	15/21
	FILTER UNTER TRSDOS 6.x	SOERENSEN, R.	18/29
	MDOS im INFO???????	SOERENSEN, R.	20/11
	DOS - (FAST) OHNE FLOPPY	SOPP	14/69
DOS-FEHLER	FEHLER IM NEWDOS-BEFEHL 'BOOT'	RETZLAFF	11/21
DOUBLER	SELBSTBAU-DOUBLER fuer EXP1		17/45
DRUCKEN	DREISPALTIG DRUCKEN	SOPP	21/15
	EIN FEHLER IM SYS28/SYS	SOPP	22/33
	LPRINT MIT MODERNEN GENIES	SOPP	18/19
DRUCKER	SPEED WRITING (fuer EPSON MX-DRUCKER)	LINDSEY	12/65
	...WEIL KUERZE SO WUERZIG IST...	MUEHLENBEIN	20/35
	DIE DRUCKERABFRAGE	SCHMITZ	18/12
	VON 8 AUF 24 PINS (= DRUCKNADELN)	SOPP	22/20
DRUCKERGRAF.	GRAPHIK-JKL, TSCRIPS UND DER NEC P6	SOPP	16/27
DRUCKZEICHEN	AM BILDSCHIRM DEFINIERT/EDITOR f.DRUCKER	KLUGE	9/19
ECB-BUS	ECB-ADAPTER-PLATINE	?	15/65
	VERBINDEND	ASSENBAUM	11/39
	DIE EINGANGSPLATINE	DROWAELDER	15/73
	ECB-BUS-GRUNDEINHEITSAUFBAU	NEUEDER	15/71
	ECB-BUS-PROJEKT	NEUEDER	14/101
	ECB-BUS-TRS80-PROJEKT	NEUEDER	13/39
	ECB-BUS fuer MODEL 3/4!???	OBERMANN	15/84
ECHTZEITUHR	ACH DU LIEBE ZEIT!	SOPP	15/32
	REAL TIME BLACK BOX?	SOPP	15/53
ERFAHRUNG	PC STATT GENIE? (Ueber weitere PC's)	SCHMITZ	20/99
ERROR	"PRUEFZAHLFEHLER BEIM LESEN"	BERNHARDT	22/71
FILEVERGLCH	VERGLEICH ZWEIER DISK-DATEIEN	SOPP	22/53
	VERGLEICH ZWEIER PROGRAMME: noch'n TOOL	SOPP	22/57
FLOPPY	FLOPPY-TRICKS	OBERMANN	13/41
FORMATIERUNG	40 TRACKS? WARUM NICHT 42?	OBSCHERNINGKAT	18/11
GENAUIGKEIT	MATHEMATISCHER UEBERFLUSS	MUEHLENBEIN	20/17
GRAFIK	ERGAENZUNG ZU GRAFIK-SHORTY	HERMANN	19/32
	TRS80 M1/3 SCHNELLE GRAFIK IN BASIC	HORNUNG	2
	HRG-BILDUEBERTRAGUNG OHNE HRG (WDR-FILES	OBERMANN	15/11
	SCHNELL BEWEGTE GRAFIK AUF DEM TRS80	ROECKRATH	2
	GRAFIK-STANDARD: ALLEIN GEHT'S AUCH	SCHROEDER	19/11
	EINF.GRAFIKPROGRAMM: PLOT/BAS	SCHROEDER /ARPS	14/29
	LINIENMODELLE IN 3 DIMENSIONEN	SCHROEDER/ ARPS	14/35
	GRAFIK-CLS fuer das GENIE IIIs	SOPP	13/35
	STRINGANZEIGE MIT VOLLDAMPF	SOPP	6/41
GRAFIK-JKL	GRAFIK-JKL fuer das GENIE IIIs	SOPP	13/32
HAENDLER	TANDY-VERTAGSHAENDLER		19/79
	TANDY SCHLIESST COMPUTER-CENTER	ZEITSCHRIFT	18/75
HARDCOPY	HARDCOPY AUCH OHNE INTERRUPT	RYCHLIK	20/15
	FLOPCOPY AUCH OHNE INTERRUPT	SOPP	21/14
HARDWARE	DATENFERNUEBERTRAGUNG		6/19
	RELAIS-INTERFACE	?	9/47
	AUSLESEN DES GRAFIKSPEICHERS	BERNH./SEELMANN	20/65

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
HARDWARE	DAS 80-ZEICHENKARTE-PROBLEM u.d. LOESUNG	BERNHARDT	22/63
	SICHERES, EIN -, PLAETZCHEN	BERNHARDT	7/43
	WACHHUND, Der-, b.unkontr.Schaltzustand	BETZ	20/63
	BILLIGES CP/M	BREWER	9/53
	TAKTFREQUENZ GENIE I/II - VERDOPPLUNG	DREYER	5
	FLOPPYDISK-CONTROLLERKARTE	DROWAELDER	18/49
	TASTATUR- UND DRUCKERPLATINE	DROWAELDER	20/71
	TRS80, DER-, UND DIE AUSSENWELT	KLEIN, WILFRIED	4
	INTERFACE-ADAPTER 8255, PROGRAMMIERBARER	KUHN	13/47
	PORTBELEGUNG	KUHN	21/44
	MODERNISIERUNG STATT NEUKAUF	MATTHAEI/H.COMP	5
	HRG-1b-KARTE	MUELLER, KURT	14/89
	JOYSTICKS FUER DEN TRS 80 M1	NEUEDER	5
	DIE MAUS AM TRS-80!	OBERMANN	22/60
	ECB-BUS-PROJEKT	OBERMANN	9/42
	EXTRA RAM ("Ein sicheres Plaetzchen...")	OBERMANN	9/63
	FLOPPYTRICKS	OBERMANN	14/87
	TAKTUMSCHALTUNG FUER VIDEOGENIE	OBERMANN	8/25
	TESTADAPTER	REGGE	9/45
	SYNCHRONISIERTE TAKTUMSCHALTUNG	RETZLAFF	10/49
	UMBAU EINES TRS80 MOD.I 16K AUF 48K	RETZLAFF	10/45
	KLEINBUCHSTABEN FUER DEN TRS 80	ROECKRATH	5
	JOYSTICK-ANSCHLUSS	SOPP	6/29
	INDIKATORKARTE fuer ECB-BUS	SPIESS	19/45
	AKUSTIKKOPPLER	TRAPP	4
	FUNKTIONSWEISE DES DRUCKERPUFFERS	TRAPP	6/18
	EPROM-PROGRAMMIERGERAET	WOLF (aus MC)	5
	HARDWARETIP: NEUE IC's	ZEITSCHRIFT	17/38
	INTERFACE MIT 8251 UND 8255 FUER Z80	ZWICKEL	2
	ANSCHLUSS GESUCHT	aus c't	20/81
	Z80-BUSANSCHLUSS	aus c't	19/51
	SICHERER PLATZ fuer DATEN? (DISKETTEN)	aus test 9/87	22/73
	DIGITALTECHNIK, GRUNDLAGEN DER -	gesammelt: KUHN	SONDER
HRG	HRG vom BASIC aus (DEMO)	NEUEDER	10/31
	HRG-PIXELS aus dem BASIC!	NEUEDER	10/23
	HRG: 384x192 oder 192x192	NEUEDER	13/14
	HRG-BILDUEBERTRAGUNG per DFUe	OBERMANN	14/24
	UMKEHR!	OBERMANN	20/58
	GENIE IIIs, DIE HRG DES -	SOPP	12/21
	HRG 1b PROGRAMMIEREN, BESSER ERKLAERT	SOPP	10/25
	HRG UND BASIC NETTO	SOPP	10/27
	HRG-HARDCOPY fuer das G IIIs	SOPP	13/25
	HRG-SPEICHER LOESCHEN, DEN -	SOPP	10/29
JKL	JKL - ABER DRUCKERSCHONEND!	SOPP	12/13
KASSETTE	CASSETTENVERZEICHNIS ERSTELLEN	WAGNER, G.	10/33
KOPIERSCHUTZ	LS-DOS 6.3 - KOPIERSCHUTZ	HERMANN	21/55
LAUFWERK	LWT fuer GENIE I und TRS-80 MODELL I	RETZLAFF	13/40
LIBRARY	WAS TUN BEI VOLLER LIBRARY?	SOPP	12/15
LITERATUR	BUECHERLISTE: DIE ERSTEN KURZKRITIKEN	SCHROEDER	21/67
	WEITERE KURZKRITIKEN	VERSCH.	22/DIV
LOGIK	BOOLE XOR ABEL	MUEHLENBEIN	17/16
	STENO-LOGIK	MUEHLENBEIN	16/24
	LOGISCHE OPERATIONEN 2	OBERMANN	16/9
	LOGISCHE OPERATIONEN IN BASIC	OBERMANN	15/9
	LOGISCHES ZU LOGIK	SOPP	16/14
	LOGIKSIMULATION mit BASIC	ZEITSCHRIFT	16/11
M4/4p-ECKE	AUFRUF DER SUPERVISOR-CALLS in BASIC	HERMANN	14/62
	DAS BOOT-ROM des 4p	OBERMANN	14/63
	HRG fuer MODEL 3/4/4p	OBERMANN	14/67
	AUFGESTIEGEN?	WAGNER, G.	14/59
MATH.STATIST	BIVARANA (2fache Varianzanalyse)	MUEHLENBEIN	19/21
MATHE-SPIEL	WAS IST MIT DER PRIMZAHL '37' LOS?	MUEHLENBEIN	11/23

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
MATHE/GRAFIK	RECHNEN (ZEICHNEN) MIT DEM COMPUTER...	SCHROEDER	10/53
MATHEMATIK	WIE der NAT.LOG.die RECHENZEIT VERKUERZT	GUAGLIANO	14/77
	PRIMZAHLBERECHNUNGEN (3 PROGRAMME)	KASPER	5
	SPLINE-INTERPOLATION (KUBISCH)	KASPER	16/19
	CRAMER UND DIE DETERMINANTEN	MUEHLENBEIN	19/13
	DAS CARTESISCHE BLATT	MUEHLENBEIN	18/5
	DER SCHNELLSTE ZAHNARZT (WURZELZIEHEN)	MUEHLENBEIN	22/11
	GOLDENER FIBONACCI	MUEHLENBEIN	16/21
MEMDISK	RESET-FESTE MEMDISK M4-CP/M	OBERMANN	19/43
MODEL 4	BOOT-ROM-VERSION und DATUM M4	HERMANN	19/42
	TURBO TIP	OBERMANN	21/31
	KLEINE ZIPPERLEIN RASCH KURIERT	WOLLSCHLAEGER	12/57
MUSIK	DA CAPO... (WIE MAN TOENE ERZEUGT)	MUEHLENBEIN	22/13
	SCHWERE KAVALLERIE ("DAS CLUBLIED")	MUEHLENBEIN	22/17
PASCAL	APFELMAENNCHEN auf der GDP64	BERNHARDT	21/37
	GENIE, CP/M und 80 ZEICHEN und HRG ???	HELD	21/43
	PASCAL-PROBLEME	KONRAD, J.	9/73
	TURBO PATCH	OBERMANN	21/32
	TURBO PATCH (FEHLERKORREKTUR)	OBERMANN	21/32
	TURBO TIP	OBERMANN	21/31
	TURBO-TIP	OBERMANN	21/31
	KURZKRITIKEN und BUECHERLISTE	SCHROEDER	21/71
	PASCAL-VORSTELLUNG TEIL 2	SOERENSEN	22/40
	PASCAL fuer INTERESSIERTE	SOERENSEN, R.	20/12
PASSWORT	PASSWORT-BYPASS BEI TRSDOS 6.2	OBERMANN	19/42
PATCH	GROSZBUCHSTABENMODUS AB SYSTEMSTART (M4)	HERMANN	21/30
PDRIVE	PDRIVE-IDENTIFIKATION fuer NEWDOS 80	OBERMANN	10/17
PLOTTER	KREIS OHNE TRIGONOMETRIE	SCHROEDER	15/8
POKEPEEK	POKE+PEEK-ECKE	HANS -?-	4
PORT	PORTBELEGUNG von COMPUTERKONFIGURATIONEN	KUHN	21/44
	OUT,PORT#,xx,yy,...	SOPP	11/29
	VERBESSERT: ID,PORT#<,xx,yy,..>	SOPP	12/11
PROGRAMMIERG	STRUKTURIERTE PROGRAMMIERUNG	?	7/17
	KEINE MEINUNG ZUR STRUKTUR?	GRELL	7/28
	EDIT-BEFEHLE	HERMANN	14/6
	BASIC- LAUFENDE PROGRAMME ERWEITERN	KRENZKE	16/18
	LOKALE VARIABLE BEIM TRS 80	PAULO	7/7
	PLANVOLL PROGRAMMIEREN	SACHSE	7/23
	DIRECTORY LESEN	WAGNER, G.	9/4
	EDITOR	WAGNER, G.	9/5
	PROGRAMMIEREN HEISST m. DATEIEN ARBEITEN	WOLLSCHLAEGER	7/29
	TRICKS MIT STRINGS	WOLLSCHLAEGER	6/65
	DEM INTERPRETER aufs BIT GESCHAUT	ZEITSCHRIFT	12/37
	DIE RICHTIGE DIMENSION	Zeitschrift	18/26
PROZESSOR	Z80, HD64180 UND ILLEGALS	SCHROEDER	17/27
	TAV's Z80-TUNING	ZEITSCHRIFT	17/33
RAM	512K-RAM-ADRESSEN m. FLEXIBLER MMU	BERNHARDT	18/67
RAMFLOPPY	MINI-RAM-FLOPPY im "SICHEREN PLAETZCHEN"	SOPP	10/37
RECHT	COMPUTERKRIMINALITAET		19/81
	URHEBERRECHTSSCHUTZ fuer PROGRAMME	KAHLEN	10/55
RECORDORGAN.	ZU GERALDs ARTIKEL ueber LMOFFSET	SOPP	10/3
ROM-ROUTINEN	CALLERLEI	MUEHLENBEIN	20/27
SCHALTALGEBR	KLEINES LOGIK-LEXIKON	aus c't	19/61
SEMANTIK	TRAU - SCHAU, WEM! Ein Bitchen Englisch	MUEHLENBEIN	18/47
SICHERHEIT	RAM-DISK KOSTENLOS	WOLLSCHLAEGER	2
SOFTWARE	SUPERTAPE FUER TRS 80	BURGWITZ/STILLR	7/44
	422 NEUE Z80-BEFEHLE	HASSELBERG (MC)	4
	MONITOR IM GENIE	KASPER	6/23
	ERWEITERUNG FUER TASMON (NEWDOS-BEFEHLE)	KONRAD J.	6/25
	ALLERLEI NUETZLICHES FUER DEN TRS 80	LICHTERMANN /MC	4
	BASIC-PATCH FUER RESTORE N	OBERMANN	5
	DER GEKNACKTE TRS 80	ROECKRATH	6/33

STICHWORT	TITEL	VERFASSER	INFO-#/SEITE
SOFTWARE	EIN NEUER 80ZEICHEN-TREIBER	SCHROEDER	19/77
	COBOL - EINE EINFUEHRUNG	TRAPP	5
	DATENVERLUSTE	aus c't	19/80
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES	SOPP	13/23
	SONDERTASTEN DIVERSER GENIES (NACHTRAG)	SOPP	14/53
SPEICHER	CALL UM DIE ECKE (MEM-BEREICHE ERREICHEN	SOPP	15/49
SPIEL	JOYSTICK AN MODEL 4/4p	OBERMANN	19/50
	TAPE-GAMES AUF MODEL III/4/4p	OBERMANN	20/57
	ADVEN-80	SCARGILL	11/9
	DAS PIRATE ADVENTURE	SCHROEDER	11/15
	ADVENTURE GEHEIMAGENT XP-05	WAGNER, ALEX.	3
SPRACHE	COBOL-EINFUEHRUNG		8/5
	DER-DIE-DAS MACHT MIR SPASZ	MUEHLENBEIN	19/75
	CP/MAC - NA UND?	SOPP	19/78
SPURANZAHL	80/40-TRACKUMSCHALTUNG	KUHN	19/49
TAKT	TAKTUMSCHALTUNG - DIE ALLERLETZTE LOESG.	MUELLER, KURT	16/45
TASTATUR	UMLAUTE IM NEWDOS? ABER NATUERLICH!	NEUEDEER	10/41
	KNOEPFE	OBERMANN	14/57
	ES GEHT NOCH SCHNELLER (TAST.ABSCHALTEN)	SCHMID, A.	22/32
	UNIVERSALGENIE	aus c't	19/57
TELEFON	TELEFONBUCH MIT PFIFF	FRISCH	2
TEXT-GRAFIK	(TELEFON) - EIN UD-ZEICHEN auf d.DRUCKER	SOPP	16/29
TEXTSYSTEM	UMDEFINIERUNG v.SCRIPSIT-KOMMANDOTASTEN	KONRAD	8/14
	KEINE ANGST VOR... (TRENNEN IN TSCRIPS)	MUEHLENBEIN	20/61
	SCRIPSIT AUF DEM ALTEN GENIE	OBERMANN	14/43
	FREIRAEUME IN TSCRIPS	SCHROEDER	14/47
	SCRIPSIT-TABELLEN	SCHROEDER	14/45
	TEXTUMLEITUNG fuer TSCRIPS	SCHROEDER	14/49
	TSCRIPS-VERSCHLUESSELUNG	SCHROEDER	15/29
	§ CONTRA @	SOPP	19/27
	KEINE ANGST VOR PROPORTIONALSCHRIFT	SOPP	18/13
	NEUES ZUR PROPORTIONALSCHRIFT	SOPP	18/15
	TIPS fuer SCRIPS	SOPP	13/19
	TSCRIPS & DIE &-CODES	SOPP	19/16
	TSCRIPS UND DIE 24 NADELN	SOPP	22/23
	VON DER WIEGE BIS ZUR BAHRE (ADR.AUTOMAT)	SOPP	19/33
	DAS TSCRIPS-DRAMA - EIN LEHRSTUECK	SOPP/MÜHLENBEIN	21/56
TONERZEUGUNG	TONGENERATOR FUER TRS 80	PLUMHOFF	8/19
UHR	UHRUNTEN fuer MODEL 4p	OBERMANN	15/47
	THE TIMES, THEY ARE A-CHANGING	SOPP	22/35
	WENN DIE UHR MAL STOERT	SOPP	14/39
UTILITY	EINGABEROUTINE MIT KOMFORT	BULLING	7/13
	TRACE-ON	HOEH (COMP.PERS	4
	AUSGABEUMSCHALTUNG IN BASIC	KONRAD J.	6/24
	DIRECTORY LESEN UND KOMMENTARE SPEICHERN	MUEHLENBEIN	13/15
	FILES WIE SAND AM MEER...	MUEHLENBEIN	11/7
	IN VISICALC SORTIEREN	MUEHLENBEIN	20/93
	FILEVERGLEICH (s.auch Progr.-Bibliothek)	NEUEDER	10/43
	DDE & LWT	OBERMANN	10/19
	LPRINT ALLES einfacher	OBERMANN	8/3
	ZAP IM SUPERZAP/Zappen im Zeichenbereich	OBERMANN	8/13
	LPRINT ALLES	SOPP	6/31
	BILDSCHIRMAUSDRUCK BEIM TRS80+VIDEOGENIE	UEBERSCHAAR	5
	SMARTSCREEN, SMARTKEY und SMARTPRINT	ZEITSCHRIFT	12/45
	PAGELIST/CMD	aus Zeitschrift	18/21
VERTRETUNGEN	TRACE od.DIE GESCHICHTE EINER ZEITSCHR.	OBSCHERNINGKAT	18/75
VIDEO	VIDEODAT-SYSTEM		18/57
VERWANDELN	UMWANDLUNG: ROEMISCHE IN ARABISCHE ZAHL	MUEHLENBEIN	16/7
Z80	ARBEITSTIER MIT GESUNDEM HERZEN	BETZ	21/52
ZAPPEN	EIN BYTE UND SEINE FOLGEN	OBERMANN	10/21
ZEICHENSATZ	NEUE ZEICHENSAETZE fuer das GENIE IIIs	SOPP	12/28
ZEITMESSUNG	ETWAS fuer LEUTE, die NIE ZEIT haben	SCHMID, A.	22/31


Der Idsteiner Hof ist sehr leicht zu finden:
- Autobahn A3 Abfahrt Idstein abfahren
- Richtung Innenstadt fahren
- direkt an der Straße, die in die Stadt führt, kommt links eine Texaco-Tankstelle
- direkt nach der Tankstelle liegt der Idsteiner Hof
- Parkplätze gibt es auf der Rückseite des Gebäudes!
Idstein

CLUB 80

Clubinfo
der
TANDY –
GENIE –
und KOMTEK –
ANWENDER

24. AUSGABE

KONTAKTADRESSE : CLUB 80 / GERALD SCHRÖDER / AM SCHÜTZENPLATZ 14 / 2105 SEEVETAL 1
TEL.: 04105 / 2602

- I N H A L T S V E R Z E I C H N I S -

Clubinternes

Seite: und Autor:

Neues vom Vorstand .... 1 - 2
Gerald Schröder

Clubtreffen-Bilder .... 3 - 6
Heinrich Betz

Vorstellungen .... 7 - 8
Bernd, Matthias

Geburtstagskinder .... 7 - 8
Jutta Obermann

In eigener Sache 10
Redaktion

Termine / Messen 10
Redaktion

Software

Wie sag' ich es meinem WORDSTAR ? .. 11 - 12
Alexander Schmid

Druckausgabe mit PRBUF 12
Werner Förster

Polynomberechnung .... 13 - 14
Save Our Script ! .... 15 - 17
Klaus-Jürgen Mühlenbein

Ein Fehler in SYS14/SYS 19
Schon wieder SYS14/SYS .... 20 - 22
Binärumwandlung aus HEX und DEZ und ..23 - 24
Was tun bei zerschossener Source ? . 25 - 26
Quatsch in OVL4/SYS .... 27 - 28
Arnulf Sopp

Das neueste DOS für M4/4p .... 29 - 33
Helmut ObscherningKat

SHELL/CMD - ... . 34 - 36
Klaus Hermann

Hardware

Seite: und Autor:

Analog/Digital - Umsetzer .... 37 - 39
Reiner Stober

HRG, daß die Schwarte kracht ! 40 - 42
Arnulf Sopp

Börse

Wer hat was -- wer will was .... 43 - 44

Sonstiges

Disketten-Kundendienst 45
Artikel aus ...

Trends: .... 45 - 46
Artikel aus PM

Brehms-Tierleben erweitert von F.W. 47 - 48
Artikel aus ...

CLUB-80-Werbeschrift 49
Heinrich Betz

Basic-Quiz 50
Darf man Programme tauschen ? 50
Befehl aus lauter Löchern 52
Artikel aus ...

Programmbibliothek

Seite: und Autor:

Clubbibliothek 51
Werner Förster

Club-80 Bibliothek

Club-80 - Bücherei .... 53 - 55
Hartmut Obermann

Die letzten Seiten

Impressum 56
Schluß 57
Redaktion

INFOFORM .... am INFO-Ende
Mitgliederadressenliste .... am INFO-Ende
Redaktion

Moin!

Als erstes möchte ich den mehr oder weniger interessierten Leser auf die Unterschrift dieses Vorwortes hinweisen: meinereiner heißt nicht Hartmut, sondern Gerald (Schröder; der aus dem Norden). Wie kömmt's? Gute Frage; Antwort folgt, nachdem Ihr Euch durch tiefes Luftholen von dem Schreck erholt habt.

Das Clubtreffen 1988

Das Clubereignis des Jahres 1988 datierte vom 12. bis 15. Mai. In Idstein fanden sich 21 Clubmitglieder und 4 Frauen bzw. Freundinnen ein. An Computern konnten wir ein TRS-80 M.1, ein Genie I, zwei Genie IIs, ein Genie IIIs und vier Model 4p aus der Riege der TRS-80-Kompatiblen sowie als CP/M-Rechner einen NDR-Selbstbau und einen Prof 180X begrüßen. Als Gäste traten ein tragbarer Toshiba 2100 (MS-DOS-Rechner) sowie zwei Atari 1040 STF auf. Die Beteiligung war von allen Seiten also äußerst stark, nur die Damen der Schöpfung hielten sich zurück.

Neben dem Kennenlernen und dem Erfahrungsaustausch, was den größten Teil der Zeit in Anspruch nahm, stand der offizielle Teil auf dem Programm. Ich möchte hier die wichtigsten Punkte zusammenfassen. Ein ausführliches Protokoll findet Ihr irgendwann im Info.

Nach den Berichten vom letzten Jahr erfolgte die Wahl des neuen Vorstandes. Ekkehard Kuhn, Werner Förster und Jens Neueder nehmen ihre Positionen weiterhin wahr, wobei Werner jetzt auch die Clubbibliothek verwaltet. An der Spitze mußten wir leider eine Änderung vornehmen, denn Hartmut Obermann bat wegen beruflicher Verpflichtungen um vorübergehenden Urlaub vom Job des 1. Vorsitzenden, der nach langen Protesten gewährt wurde. Als Ersatz wurde der Jüngste des Clubs gewählt (siehe oben und unten). Den vakanten Job des 2. Vorsitzenden erhielt Bernd Retzlaff, während die Kasse weiterhin von Jutta Obermann verwaltet wird.

Ich möchte mich (sicherlich auch in Eurem Namen) bei Hartmut für die engagierte Arbeit bedanken, die er für den Club geleistet hat. Ich hoffe, daß er in einem Jahr wieder den Vorsitz übernehmen kann und werde versuchen, den Club in seinem Sinne weiterzuführen. Da ich weder die Energie noch die Stimmkraft von Hartmut habe, werde ich durch mehr oder minder geschicktes Delegieren die Aufgaben und Fragen auf kompetente Personen verteilen (wie schon mit Clubbibliothek und Kasse geschehen). Ihr könnt Euch darauf verlassen, daß ich die Aufgabe ernst nehme und alle Eure Fragen beantworten werde, soweit es mir möglich ist.

Außerdem stimmten wir über das Clubemblem ab. Aus den 13 im Info veröffentlichten Vorschlägen wurde das unten noch einmal abgedruckte von Heinrich Betz gewählt. Vielen Dank an Heinrich, der sich bereitgefunden hat, die Ausgestaltung für Werbeplakate u.ä. zu übernehmen.

Beim letzten Info hatte Jens Schwierigkeiten mit der Vervielfältigung. Für das Drucken gingen 500 DM drauf, weil Peter Spieß seine Druckerei aufgegeben hat. Glücklicherweise fand sich das Neumitglied Horst-Dieter Schroers bereit, den Druck zu übernehmen. Ich möchte ihm dafür danken, daß er das vor Euch liegende Ergebnis möglich machte und hoffentlich weiter möglich machen wird.

Eine schleichende Unterwanderung des Clubs durch CP/M, die u.a. durch die Mitgliedschaft von Matthias Homann dokumentiert wird, der keinen TRS-80-Kompatiblen besitzt, wurde offiziell anerkannt. Das soll heißen, daß wir nun neben der Pflege der TRS-80-Tradition auch das Motto CP/M auf unsere Fahnen

schreiben. Die anwesenden Mitglieder sprachen sich allerdings dagegen aus, auch Besitzer von MS-DOS-Rechnern oder Ataris anzusprechen. Artikel über diese Rechner werden toleriert, besonders in puncto Hochsprachen, aber die Akzente sollen weiterhin im Z80-Bereich liegen.

Zur Bücherei ist noch zu sagen, daß Werner gerne etwas mehr Arbeit als Hartmut haben möchte, der im letzten Jahr nur ca. 5 Ausleihungen verzeichnen konnte. Leute, was lest bzw. lernt Ihr denn??? Wir haben uns darauf verständigt, vorerst die Neuanschaffungen zu reduzieren, bis Ihr etwas lesefreudiger werdet. Es werden nur noch Bücher und Zeitschriften angeschafft, die sonst in Deutschland schwer zu bekommen sind, also vor allem amerikanische Sachen. Vorschläge dazu sind immer willkommen!

Im weiteren Verlauf des Clubtreffens hörten wir noch zwei Referate. Rüdiger Sörensen berichtete über seine Erfahrungen mit Turbo-Pascal und Turbo-Modula, während Harald Mand die Möglichkeiten von dBase II darstellte. Vielen Dank an die beiden Referenten für die interessanten Ausführungen!

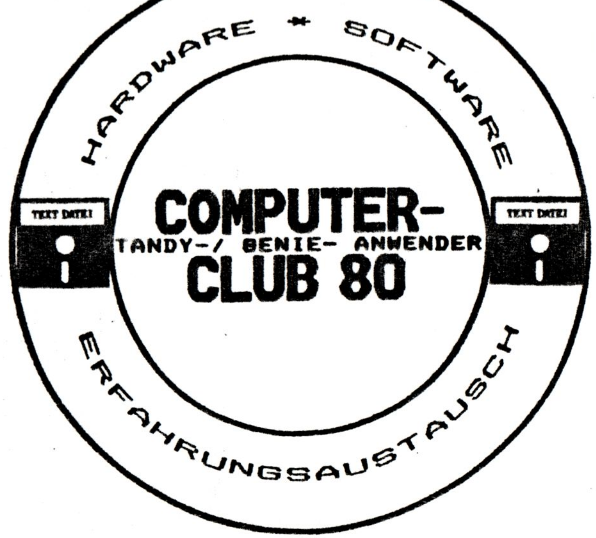
Der Tagungsort Idstein scheint ideal zu sein: Unterbringung und Essen waren gut, Räumlichkeiten sind prinzipiell ausreichend vorhanden, der Ort ist per Autobahn von allen Ecken gut zu erreichen. Also haben wir gleich wieder zugeschlagen und das Hotel vom 28.4.89 bis 1.5.89 gemietet (Freitag bis Montag; Montag ist Feiertag!). Streicht den Termin gleich in Eurem Kalender an!

So, das war's schon. Ich hoffe, daß Ihr Euch weiter rege am Clubleben beteiligt. Es wäre toll, wenn es in diesem Jahr wieder ein Nordländer- und endlich mal ein Südländertreffen geben würde. Weiterhin möchte ich Euch bitten, Hartmut erstmal mit Anfragen zu verschonen und mir Euer Vertrauen zu schenken. Ich werde versuchen, es zu rechtfertigen.

In diesem Sinne fröhliches Hacken wünscht

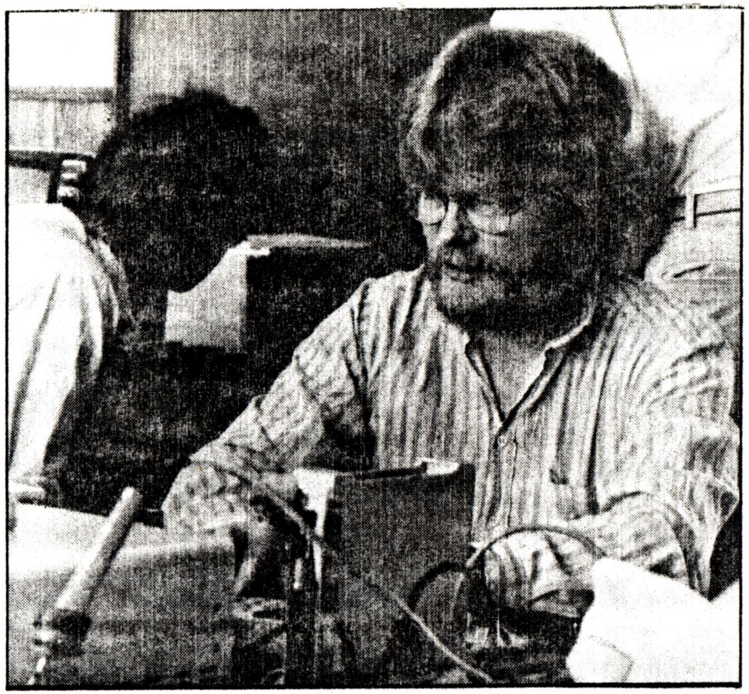
Gerald Schröder

Das neue Clubemblem (entworfen von Heinrich Betz):



HEFT 24
Juni 1988

über das Ausdrucken von DBASE-Graphiken unterhalten sich Gerald Dreyer und Harald Mand



Assembler-Spezialist Arnulf Sopp bei der Arbeit mit seinem Video-Genie

CLUBTREFFEN
Fotos von Heinrich Betz
14. MAI 1988

Beim Clubtreffen gab es nicht nur Hardware und Software, sondern auch "Fluidware". Gerald Dreyer sorgt für Nachschub mit der Kaffeekanne

HARDWARE * SOFTWARE
COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80
ERFAHRUNGSAUSTAUSCH
TEXT DATEI
TEXT DATEI

Unter mehreren Entwürfen wurde dieser als neues Emblem des Club 80 ausgewählt

Im leicht abgedunkelten Raum flimmern die Bildschirme. Man tauscht Erfahrungen aus

=---->

In dieser Runde wurden Vorschläge gemacht, man diskutierte und faßte Beschlüsse

Gerald Schröder freut sich, daß er einstimmig zum neuen Clubvorsitzenden gewählt wurde

Bernd Retzlaff

Kleiner Sand 98 2082 Uetersen Tel.04122/43551

Hallo liebe Leute!

Da man mich auf der Jahreshauptversammlung in Idstein zum 2. Vorsitzendes unseres Clubs gewählt hat will ich mich mal ganz kurz vorstellen, da mich ja nicht alle Mitglieder persönlich kennen.

Ich bin 33 Jahre alt, verheiratet und habe zwei Töchter im alter von 6 Jahren, woraus man schon ersehen kann, das meine Freizeit die für den Computer draufgeht etwas knapp bemessen ist da die Familie auch etwas Zeit beansprucht. So nun etwas zum Thema Computer. Ich besitze einen Genie I, Modifikationsstand unbekannt, da ich über die zahlreichen Basteleien nicht mehr Buch führe. Dazu gesellt sich dann noch ein eine Floppylaufwerkseinheit der Firma TCS mit zwei Laufwerken von 40 Tracks / Single Sided / Double Density und ein Drucker Panasonic 1081. Das Denkmöbel ist in der Zwischenzeit nicht mehr ganz serienmäßig sondern hat noch einige Modifikationen erhalten, die da währen, Speedup auf 3,5 Mhz, Hochauflösende Grafik (HRG1B) und eine CP/M-Modifikation. Eine 80-Zeichen Karte (Schmidke-Elektronik) befindet sich noch im Einbau .

Meine Hauptanwendungsgebiet des Genie-Rechners ist im Moment die Hardwarebastelei und Textverarbeitung. Beruflich bin ich als Technischer Angestellter im Bereich der Automatischen Meßtechnik beschäftigt. Hierbei ergibt sich zwangsläufig eine starke Verknüpfung von Hardware und Software, so das das Hobby Computer auch beruflich zum tragen kommt.

So nun aber genug der kurzen Vorstellung, sonst schlaft Ihr noch beim Lesen ein und das will ich als neues Vorstandsmitglied nicht verantworten müssen, da es sicher auch in diesem Info allerhand interessantere Artikel als diesen gibt.

Ich wünsche euch noch weiterhin viel Spaß am Computer!

Bernd

Hallo 80ger Fan's !

Hier ist sie nun, meine Vorstellung im Club. Seit Januar '88 bin ich da nun schon Mitglied und hab mich immer noch nicht Vorgestellt !

Also erstmal was zur Person. Ich bin 21 Jahre alt, davon ca. 6 als Computerfan. Gelernt hab ich Energieanlagenelektroniker (schön lange Berufsbezeichnung, sagt aber nicht viel aus) und tätig bin ich jetzt als Physikstudent in Hamburg (gerade angefangen). Der Computer ist in den letzten zwei Jahren als Hobby zwar etwas von Kino, Theater, Konzerten und sonstigen kulturellen Veranstaltungen zurückgedrängt worden (rein zeitlich, nicht aus Desinteresse), aber immer noch bedeutend.

Ich bin ja nun das erste Mitglied, das keinen Tandy, Genie oder Komtek Computer sein eigen nennen kann und mit diesen Geräten bisher auch wenig zu tun hatte. Aber es gibt ein gemeinsames Interesse ! Auch mein Computer hat einen Z80 (stimmt nicht: HD64180) und da gib es das schöne alte CP/M, was dann auch auf meinem läuft.

Mein Hauptinteresse an Computern liegt in der Hardware (da hat man was zum Anfassen!), doch was nützt das ganze Zeugs ohne (einigermaßen) passende Software ?

Nun zu meinem 'Equippment' :
Mein 'System' besteht aus einem NDR-Klein-Computer mit einer Floppy (5 1/4''). Die Systembezeichnung ist inzwischen eigentlich verkehrt, da ich praktisch an jeder Platine rumgebastelt hab, die Speicherkarte eine ganz eigene Creation ist und im Betriebssystem auch rumgepfuscht wurde (von wem bloß!). Das macht aber alles gar nichts, denn was soll ich im Club mit meinen Systemdetails ? Hauptsache, CP/M läuft und grundlegende Programmiertechniken stimmen überein.

Ich hoffe dies reicht für den Anfang !

Guten Datenfluss wünscht Euch

Matthias Homann

CLUB 80 — Geburtstagskinder!!!

Hallo liebe CLUB-Freunde,

das Clubtreffen ist nun vorbei und ich bin sicher, denen die dabei waren hat es gefallen. Jetzt können wir wieder mit Schwung in die nächste Inforunde gehen!

Auch diesmal gibt es wieder viele Feste zu feiern. Hier die Namen derjenigen, die in den Monaten April, Mai und Juni Geburtstag hatten bzw. haben.

April: 01. Mary Jo Kostya (unser erstes weibliches Mitglied)
07. Alexander Schmid
19. Wilhelm Tornow (unser neuestes Mitglied)
26. Gerhard Loose
28. Walter Piller
30. Ulrich Böckling

Mai: 03. Werner Förster
14. Wolfgang Lachmann
18. Hans Raggan
19. Rainer Jablotschkin
28. Helmut Obscherningkat und Gerald Dreyer

Juni: 01. Wolfgang Beckhausen
15. Harald Mand
20. Eckehard Kuhn
28. Matthias Homann

Meine herzlichsten Glückwünsche, auch im Namen des Vorstands, an alle genannten und ungenannten Jubilare, euere Clubmaus

Jutta

Ich weiß, daß meine bessere Hälfte es nicht gerne sieht, trotzdem muß ich hier noch ein paar Zeilen anfügen. Jutta hat einen sehr wichtigen Geburtstag vergessen, den ich hier noch anführen möchte, nämlich ihren eigenen. Unsere Clubmaus reiht sich in die Liste der im Juni geborenen genau zwischen Eckehard und Mattihas ein, hat also am 24. Grund zum Feiern. Im Namen des Vorstandes und ich glaube auch aller Clubmitglieder wünsche ich alles Gute!

Hartmut Obermann

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....30. Juni 1988

-- Messen '88 --

ORGATECHNIK ....Köln 20. - 25. Oktober 1988

In eigener Sache

Wie Ihr gesehen habt ist unsere "Messen & Termine" Spalte diesmal recht kurz.
Aus diesem Grund bitte ich um Eure Mithilfe, denn ich habe für dieses Jahr außer Redaktionsterminen keine mehr.
Interessant sind alle Termine oder Messen, die mit Computer und/oder Zubehör zu tun haben. Auch kleinere regionale "Veranstaltungen" zu diesem Thema sind von Bedeutung.
Ich hoffe, daß Ihr mir da weiterhelfen könnt.
Telefonanruf genügt.
☎ 0791 /42877

Die Redaktion

Wie sag' ich es meinem WordStar ?

Viele von Euch haben ja schon ein CP/M für Ihre Kiste (allein 16 von den 20 Anwesenden beim Clubtreffen) und auch schon mit dem Wordstar gekämpft. Als einer der größten Kritikpunkte wurde dabei immer wieder die lahme Textausgabe, besonders beim Scrollen, bemängelt. Wenn jedes Zeichen einzeln ausgegeben werden muß (Terminal-Betrieb) kann das ja nicht schneller gehen, aber es gibt auch einen direkten Weg, und den möchte ich hier am Beispiel der 80-Zeichen-Karte von Schmidtke demonstrieren. Die Anpassung z.B. für das Model IV/P und alle Rechner, die direkt auf den Video-Speicher zugreifen können, ist aber fast identisch und ohne Probleme zu machen.

Die folgenden Patches finden bei den Versionen 3.0 und 3.3 bei den gleichen Labeln statt. Nur für DDT o.ä. sind die absoluten Adressen natürlich anders, was hier aber niemanden zu stören braucht, der Installer reicht vollkommen.

Das Schwierigste ist, sich im Installer überhaupt bis zum Patcher durchzukämpfen. Bei der Version 3.0 kommt man, wenn man nach den ganzen Menüs auf die Frage 'Are the modifications now complete' mit 'N' antwortet, ans Ziel, beim 3.3 taucht diese Frage leider nie auf und man muß sich durch die Hintertür reinmogeln: gleich im ersten Menü darf man weder mit A,B,C,... oder mit X antworten, sondern mit +, obwohl das da gar nicht aufgeführt ist. Normalerweise sollte sich der dienstbare Geist jetzt melden.

Wenn diese Hürde genommen ist, kann's losgehen:

Achtung: beim 3.0 muß man die Adressen als 'LABEL:', 'LABEL:+1' usw. eingeben, beim 3.3 als ':LABEL' usw.

1.) bei MEMAPV wird FF eingetragen, um dem WS zu sagen, daß er memory-mapped arbeiten soll.

2.) bei MEMADR wird die Adresse des Video-RAMs eingetragen, z.B. '00 F0'
Achtung: wie üblich erst das Low-, dann das High-Byte!

3.) wer inverse Zeichen hat, sollte bei HIBIV ein FF schreiben

4.) um den Cursor nachher noch zu sehen, muß bei CRBLIV auch FF stehen

5.) um an den Bildschirmspeicher zu kommen, muß bei SWIN '3E 05 D3 D2 C9' und um ihn wieder auszuschalten bei SWOUT '3E 04 D3 D2 C9' rein

6.) damit der Cursor der 80-Z-Karte nicht stört, sollte beim
<u>WS 3.0</u>: 'C3 E0 02' bei INISUB und 'C3 F0 02' bei UNISUB stehen, beim
<u>WS 3.3</u>: 'C3 CB 02' bei INISUB und 'C3 DB 02' bei UNISUB.

7.) damit er jetzt nicht ins Leere springt muß bei MORPAT noch
'1E 1B 0E 02 CD 05 00 1E 0C 0E 02 CD 05 00 C9 00
1E 1B 0E 02 CD 05 00 1E 0D 0E 02 CD 05 00 C9 00' stehen.

Die Punkte 2.), 5.), 6.) und 7.) sind natürlich gerätespezifisch und müssen angepaßt werden. Beim Model IV/P z.B. sind außerdem 6.) und 7.) überflüssig. Da ich die Video-Adressen und die Schaltsequenzen leider nicht im Kopf habe, müßt Ihr halt ins Handbuch schauen, da steht es bestimmt irgendwo.

Wenn alles stimmt, sollte man erst mal einen Funktionstest machen, ob man nicht danebengepatcht hat. Nehmt also einen Text von mindestens einer Seite (ein DOC-File hat doch jeder irgendwo) und spielt damit etwas rum (editieren, abspeichern...). Falls es nicht kracht und die Bildschirmausgabe wie früher ist, bestehen gute Chancen, daß es dabei auch bleibt.

Damit es keinen unnötigen Frust gibt hier nochmal: die Ein- und Ausschaltsequenzen sind nur für die 80-Zeichen-Karte von Schmidtke und für nichts anderes zu gebrauchen, die Labels gelten aber allgemein.

Herzlich willkommen im Reich der schnellen Textverarbeitung

Alexander Schmid

Druckausgabe mit PRBUF

Bei der Bearbeitung der CLUB80-Bibliothek sind oftmals umfangreiche Textdateien oder Programme auszudrucken. Bearbeitet man die Druckausgaben mit dem NEWDOS-Commando PRINT, so muß man nach jedem File erst einen manuellen Papiervorschub machen.

Einfacher geht es mit dem Hilfsprogramm PRBUF. Mit einem Textverarbeitungsprogramm erstellt man sich eine Datei der zu druckenen Dateien und fügt zusätzlich die mit (☞) markierten Zeilen ein. Dieses Hilfs-File wird dann als ASCII-Datei gespeichert (in SCRIPSIT mit S,A name/JCL).

z.B.

```
☞   PRBUF 4096        (Printerbuffer)
☞   *PR P=72 L=66     (Seitenlänge, Anzahl
                        der Druckzeilen)
    PRINT AGENDA/ILF
☞   *PR T             (Seitenvorschub)
    PRINT DMS/ILF
☞   *PR T             (Seitenvorschub)
    PRINT ENHBAS/ILF
☞   *PR T             (Seitenvorschub)
    PRINT OMNITERM/ILF
```

Diesen JOB ruft man jetzt mit DO name auf und kann getrost in der Zwischenzeit einer anderen Tätigkeit nachgehen.

Viel Spaß

Werner Förster

POLYNOMBERECHNUNG

- aber vollständig!

Numerische Integration transzendenter Funktionen, Matrizenrechnung, Differentialgleichungen, Differentialgeometrie, Interpolationsverfahren - ja sogar Anwendungen in der Quantenmechanik, Variations- und Störungsrechnung und alle möglichen anderen Problemlösungen in BASIC und PASCAL habe ich in der Fachliteratur gefunden - nur kein Programm zur vollständigen Berechnung eines Polynoms beliebigen Grades, d.h. der Werte einer solchen ganzrationalen Funktion sowie ihrer sämtlichen Ableitungen!
Dabei sollte gerade dieses am häufigsten auftretende "Problem" leicht zu programmieren sein.
Ich lege deshalb hier eine eigene Lösung dafür vor und würde mich freuen, wenn jemand eine bessere - d.h. schnellere - gefunden hat.

Eine ganzrationale Funktion n^{ten} Grades hat die Gestalt:

$$f(x) = \sum_{i=0}^{n} a_i x^i$$

Ihre m^{te} Ableitung läßt sich mit den bekannten Kurzzeichen für Summe (Σ) und Produkt (Π) wie folgt zusammenfassen:

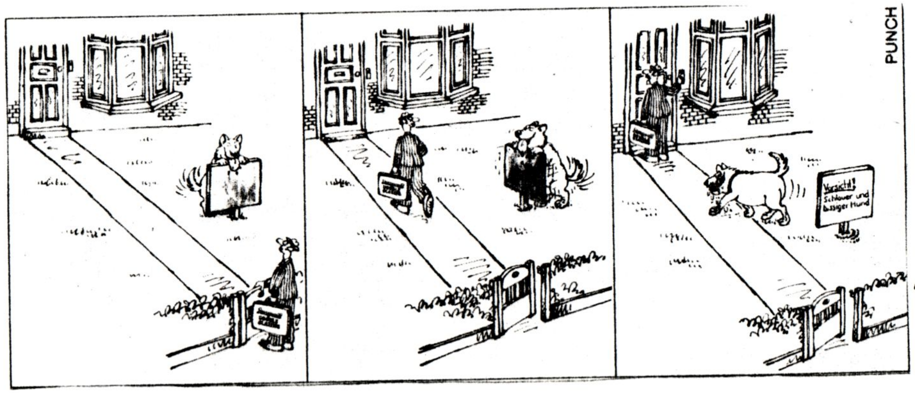
$$f^{(m)}(x) = \sum_{k=m}^{n} \{ a_k x^{k-m} \cdot \prod_{i=0}^{m-1} (k-i) \}$$

Diese Form stellt gleichzeitig die Anleitung für die Programmstruktur dar. Sie ist also mit zwei Schleifen - für k und i - aufzubauen. Es empfiehlt sich jedoch, auch die Potenz von x als Schleife zu programmieren, da die Potenzanweisung der CPU (a↑b) bei höheren Potenzen ungenau wird.

Wegen des Produktes in der zweiten Formel ist die erste in ihr nicht enthalten; setzte man in der Differentiationsformel m=0 (die "nullte Ableitung" ist die Funktion selbst), so erhielte man für das erste, das sogenannte lineare Glied des Polynoms den Wert Null. Mit der logischen IF-Bedingung kann man diesen Fehler jedoch ausschließen (s.Zeile 80) und erhält damit einen einheitlichen Rechengang sowohl für die Berechnung des Polynoms selbst als auch für sämtliche Ableitungen.

(Anmerkung: Das *HORNER*sche Schema funktioniert nur für die Funktion selbst sowie ihre erste Ableitung. Für höhere Ableitungen ist es nicht vorgesehen.)

Das Programm arbeitet auch mit dem primitivsten Computer (meiner hat z.Z. immer noch nur 1,77 MHz!) und "trotz BASIC" verblüffend schnell! Ich hoffe deshalb, daß es für die Mathematiker unter uns eine echte Hilfe ist,
...and IF NOT
contact KAJOT !



Berechnung eines Polynoms und seiner Ableitungen

<C> K.-J.Mühlenbein, Weinheim, 25.04.1988

```
5 CLS:PRINT"                P O L Y N O M B E R E C H N U N G
      Hiermit koennen sowohl das Polynom als auch die
  Differentialquotienten beliebiger Ordnung berechnet werden
7 CLEAR80:PRINTSTRING$(62,"="):PRINT:PRINT
10 DEFINTI-N:DEFDBLX
20 INPUT"Grad des Polynoms ";N:DIMA(N)
30 PRINT:PRINT"Gib die Koeffizienten ein.
Beginne mit der niedrigsten Potenz!
(Bei fehlender Potenz setze als Koeffizient Null ein!)":PRINT
40 FORI=0TON:PRINT"A ("I") = ";:INPUTA(I):NEXT
50 S=0:PRINT:INPUT"Die wievielte Ableitung soll berechnet wer
den ?
(zur Berechnung des gegebenen Polynoms selbst gib '0' ein) ";
M:IFM>NPRINT:LPRINTCHR$(07):PRINT"Die Ableitung kann nicht ho
eher als der Grad des Polynoms sein!":GOTO50
60 INPUT"Fuer welchen Wert von x ";X
70 FORK=MTON
75 PX=1:PN=1
80 IFM>0:FORI=0TOM-1:PN=PN*(K-I):NEXTI
90 IFK>M:FORJ=1TOK-M:PX=PX*X:NEXTJ
100 S=S+PN*PX*A(K)
110 NEXTK
120 PRINT:PRINT"Die"M".Ableitung hat an der Stelle x ="X"den
Wert"S
130 PRINT:INPUT"    Noch eine Berechnung <J / N> ";J$
135 IFJ$="n"ORJ$="N"THENEND
140 INPUT"    mit dem gleichen Polynom <J / N> ";N$
145 IFN$="n"ORN$="N"THENCMD"F=ERASE",A(0):PRINT:GOTO20
150 GOTO50
```

Save Our Script !

Wie man seinen TSCRIPS-Text rettet
oder: Aller guten Dinge sind -?-

Neulich ließ mich *TSCRIPS* im Stich.
Zweieinhalb DIN A4-Seiten hatte ich eingetippt, um die Mensch-
heit mit den größten Erkenntnissen zu beglücken, die seit
ARCHIMED in eines Menschen K(r)opf aufgegangen: Die Qua-
dratur des Kreises war mir geglückt!
Gerade wollte ich das zweite Kapitel meiner Enthüllungen er-
öffnen - da schlug mir *Murphey* den Schlüssel zum Nobel-
preis aus der Hand: War es ein Floh aus dem Netz? War es
ZEUS' Zorn über *Prometheus*? Jedenfalls: Mein Cursor war
weg, einige Dutzend Pseudo-Cursors (7FH) machten sich auf dem
Bildschirm mitten im Text breit, einige meiner wohlgesetzten
Wörter waren verstümmelt, als hätte ich sie gekränkt (so emp-
findlich sind wissenschaftliche Texte!) - zu Eis erstarrt war
das Ganze und blickte mich aus bösen, toten Augen an...
Was tun? Von vorn beginnen? Vor Schreck hatte ich prompt
vergessen, wie man einen Kreis quadriert!
Voller Hoffnung schlug ich *TRAPPSCHUHs* Kapitel "Rettung bei
Absturz" auf (Anleitung zu Version 5, Seite 12). Ich tat, was
er dort
1. als <u>ersten Versuch</u> empfiehlt:
   - a) Ich RESETtete ('Resetten' heißt noch nicht Retten!)
   - b) Ich lud *TSCRIPS* neu;
   - c) Im Logo (!) drückte ich "a" (für "alt"; meiner tut's auch mit Minuskeln!)
   - d) Ich drückte SHIFT-↑;
   - e) Ich drückte <CLEAR>;
   - f) Ich - nein, ich drückte nichts mehr. Jetzt drückte sich der Text. Er kam nicht! Er brachte mir was herfür...

Nicht CLEAR - nicht Bier
- außer einem Affenschwänzchen links oben; es äffte mich!

Also weiter in *TRAPPSCHUHs* Text: Wenn's so nicht gelingt,
empfiehlt er
2. als <u>zweiten Versuch</u>:

Mit <u>DUMP RETTUNG 9176H FFFFH</u>
den gesamten Textpuffer auf Diskette zu sichern und danach
TSCRIPS wieder zu starten; dann solle man mit @-O das
DOS-Kommando
<u>LOAD RETTUNG</u>
"absetzen". * Ich tat das alles genau - doch es brachte wieder
nichts! (Vielleicht hatte ich falsch "abgesetzt"? Ich hatte
dieses Kommando nur "eingegeben"!)
Dank DUMP hatte ich nun aber den gesamten Textpuffer (und mehr
als das) wenigstens auf Diskette und konnte ihn mir mit *SU-
PERZAP* ansehen. In der Tat: Mein (zwar verstümmelter) Text
befand sich dort in etlichen Sektoren! Ich war wieder ganz
froh, denn nun könnte ich meine Entdeckung notfalls von dort
mit JKL ausdrucken und nochmals abtippen. Nicht nur mein Text,
auch meine Erfindung und somit die Menschheit wäre gerettet!

Doch *TRAPPSCHUH*, der für meinen Mühlenbein-Fuß etwas zu
groß ist, rät weiter, man solle
3. als <u>dritten Versuch</u>

"das File RETTUNG nach versehentlichen OOH durchsuchen". Solche
waren in der Tat reichlich in dem vergewohltätigten Text vor-
handen. *TRAPPSCHUH* rät dem unkundigen Retter allerdings
leider nicht, <u>was</u> man aus den Doppelnullen machen solle. So
wird aus dem Retter zunächst ein Rater...
Aber auch die Beseitigung dieser 00s half nicht; als ich
wieder "LOAD RETTUNG" eingab, tat sich ebenfalls nichts - rein
gar nichts: NIRWANA!...

... --- ... ... --- ... ... --- ...
SOS!

* * * Wer hilft? * * *

Nicht TRAPP, nicht TROPP - sondern *SOPP*!
Arnulf, unser Columbus, knickte das Ei.

Wenn der Computer streikt und sich aufgehängt hat, während man
im Textsystem ist - mit Vorliebe dann, wenn man bereits 10
Seiten vollgeschrieben hat - bleibt einem in der Tat zunächst
nichts anderes als ein "RESET" übrig. Wenn der Text bereits aus
dem Speicher verschwunden war, hilft kein Betteln und Beten
mehr. In der Regel befindet er sich nach diesem "Warmstart"
jedoch noch im Speicher. Das läßt sich feststellen, indem man
TSCRIPS wieder lädt und die Anfangsfrage "ALT oder NEU?" mit
'A' bzw. 'a' beantwortet, damit der Textspeicher nicht gelöscht
wird. Wenn der Text dann nicht auf dem Bildschirm erscheint und
auch weder Rettungsaktion 1 noch 2 noch 3 (s.o.) ihn dahin
bringen, so kann das folgende Ursache haben:
TSCRIPS überträgt den Textpuffer Byte für Byte in das Video-
-RAM, bis es auf ein Nullbyte '00'H stößt. Dieses betrachtet es
als das Ende des Textes. Wenn nun durch den "Großen Unbekann-
ten", der das System zum Absturz brachte, schon das erste oder
eines der ersten Textbytes in 00 verwanzt wurde, wird eben so
gut wie nichts ins Video-RAM geladen. Geschah diese "Annullie-
rung" mitten im Text, so erscheint entsprechend nur der Text-
teil bis zu dieser Stelle.
Nun gibt es zwei Möglichkeiten, dagegen vorzugehen:
Entweder nach *TRAPPSCHUH* den Speicherinhalt mit DUMP si-
chern und anschließend entwanzen, oder gleich im Speicher ent-
wanzen, auf Neudeutsch: <u>DEBUG</u>!
Wir betrachten beides einmal genauer (A & B):
A) Das mit DUMP erzeugte File "RETTUNG" (s.o.) wird mit DDE
oder SUPERZAP begutachtet. Da das DUMPen ein File in Maschi-
nensprache erzeugt, enthält es sog. "Headers" und "Loader-
-Bytes", die dem System beim Ladebefehl LOAD (im DOS) anzeigen,
<u>wieviele Bytes wohin</u> geladen weren sollen. Dieser Header
lautet <u>01</u>. Also: überall wo "01" auftaucht, haben wir es mit
dem Beginn einer Folge von Loader-Bytes zu tun, die wie folgt
aufgebaut ist:

```
01-nn-LSB-MSB
 |  |   └──┴──► Ladeadresse
 |  └─────────► Anz. zu ladender Bytes + 2
 └────────────► Lade-"Header"
```

Ab Adresse MSB-LSB werden die nn-2 Bytes in den Speicher gela-
den, die auf das MSB folgen! Die vier Loader-Bytes werden na-
türlich nicht in den Speicher geschafft (wozu auch?). Der
Bytezähler nn schließt jedoch (aus Gründen Höherer ML-Mathema-
tik) die beiden Adreßbytes ein, so daß wir zwei Bytes abziehen
müssen, wenn wir wissen wollen, wieviele Datenbytes tatsächlich
geladen werden. (übrigens: nn=00 bedeutet hier nn=256! Der be-

treffende Sektor enthält dann 254 Datenbytes; wenn nn=01, enthält er 255, bei nn=02 sind es 256 Bytes - jeweils plus 4 Headerbytes!) Danach wird ein neuer Sektor oder "Block" angelegt.

Da ein Text meistens mehr als nur einen Block enthält, stehen also oft mehrere "Doppelnuller" auf der Diskette. Bei Bereinigung des RETTUNG-Files mit DDE oder SUPERZAP dürfen diese Doppelnuller, die auf ein Headerbyte "01" folgen, natürlich nicht beseitigt werden, sonst versagt der Ladevorgang. (Diesen Fehler hatte ich offensichtlich gemacht.) Alle übrigen "00" werden entweder in sinnvolle, zum Text (den man in SUPERZAP rechts in Klarschrift sieht) passende Zeichen oder in "20" (Blank) verwandelt. Danach sollte der Text, wie von *TRAPPSCHUH* beschrieben, nach 'LOAD RETTUNG' auf dem Bildschirm erscheinen.

⇒ 4.

B) Etwas weniger umständlich scheint mir das Verfahren mit DEBUG (und eben darauf hat mich *Arnulf* gebracht):
Nach RESET lud ich TSCRIPS ALT;
danach schritt ich 1, 2, 3
flugs zur Hex-DEBUGgerei! (123)

Jetzt mußte ich nur noch den Textanfang im Speicher finden. Ich fand ihn. Auch ohne *TRAPPSCHUH*.

Der Text steht ab Adresse 9176H im Speicher.

Er enthält natürlich keine Loader-Bytes, da im Speicher ja noch Originaltext steht! Nun machte ich aus den Doppelnullen Buchstaben, die in den Text paßten, oder verwandelte sie in 20H (selbstverständlich ohne "H"). Nur das Schlußbyte (s.o.) ließ ich stehen. Dann nichts wie "G" getippt, geENTERt und - siehe da: mein geschundener Text strahlte mir gesund und munter auf dem Bildschirm wieder entgegen!

Nun konnte ich meine Kreise wieder quadrieren und dachte dabei an die römischen Soldaten, die einst auf das Zauberwort "Noli turbare circulos meos!" *) vor *ARCHIMEDES* das Weite suchten (und fanden).

*) "Störet meine Greise nicht!"

Ergo: Das Memory ist (fast) unvergänglich. Solch ein Gedächtnis müßte man haben!

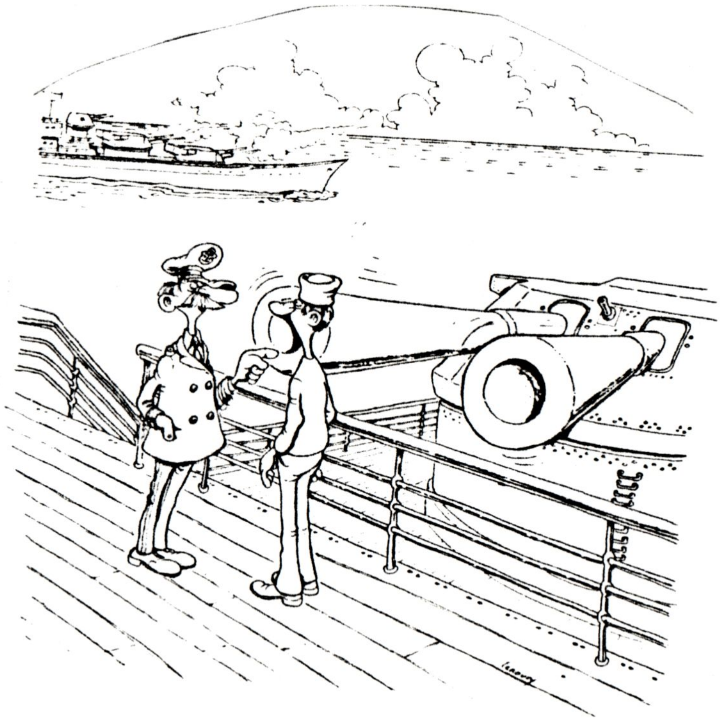
Mein nächtes Kapitel über das Thema "Wie ich den Stein der Weisen fand und Aurum aus SCH....IPs machte" ist in Arbeit. Nun habe ich keine Angst mehr vor den Römern aus der Steckdose!

Schreiben Ohne Schrecken !

HEURE-Ka-Jot

PS: NEWSCRIPT ist etwas narrensicherer und speichert lange Texte zwischendurch automatisch!

KJ



Sind Sie sicher, daß das neue Zielansteuerungsprogramm richtig funktioniert?

HEFT 24
Juni
1988

Ein Fehler in SYS14/SYS

In SYS14 befinden sich u. a. die Routinen für die Library-Befehle PRINT und LIST. Mit ihnen kann eine Datei auf den Bildschirm bzw. den Drucker ausgegeben werden. Dabei werden alle Zeichen durch einen Punkt ersetzt, die nicht anzeigbar oder druckbar sind. Jedenfalls unterhalb ASCII 20h (Leerzeichen) trifft das zu. Bei den Zeichen am oberen Ende des druckbaren Bereichs haben sich die Autoren des DOS offenbar verrechnet:

```
        AND   7fh        ;Bit 7 löschen
        PUSH  HL         ;retten
        LD    HL,m4370   ;dort höchstes druckbares Zeichen
        CP    (HL)       ;ist es das?
        POP   HL         ;restaurieren
        JR    C,m5145    ;nein, ausdrucken bzw. anzeigen
        JR    Z,m514b    ;falls genau dieses Zeichen
m5145   CP    ' '        ;Blank oder kleiner?
        RET   NC         ;zurück, falls anzeigbares Zeichen
        CP    0dh        ;CR?
        RET   Z          ;falls ja
m514b   LD    A,'.'      ;andere Zeichen durch Punkt ersetzen
        RET
```

Spielen wir das einmal durch. Das höchste druckbare Zeichen sei 7Eh, denn 7Fh ist für den Drucker DEL (löscht das zuvor ausgegebene Zeichen). Was dem Drucker soeben noch zugemutet werden kann, wird durch die SYSTEM-Option AX=n vorgegeben. Diese Routine druckt zwischen 20h und dem Zeichen gem. AX minus 1 alles. Das durch AX vorgegebene Zeichen (also ein noch eben druckbares!) wird zum Punkt, was darüber liegt, geht wieder unverändert durch zum Drucker. Also auch unser DEL, das nicht nur nicht auf dem Papier erscheint, sondern das obendrein seinen Vorgänger aus dem Druckerpuffer löscht. Nicht gerade logisch, das.

Mit einem einfachen Patch ist das zu beheben:

```
        ...              ;fängt wie oben an
        LD    HL,m4370   ;höchstes druckbares Zeichen
        INC   (HL)       ;auf kleinstes nicht mehr drb. Z.
        CP    (HL)       ;Zeichen für Ausdruck zu hoch?
        DEC   (HL)       ;Zeichen restaurieren
        POP   HL         ;dto. Register
        JR    NC,m514b   ;Punkt, falls kein druckbares Z.
m5145   CP    ' '        ;Blank oder kleiner?
        ...              ;weiter wie oben
```

Das AX-Zeichen wird hier zunächst um 1 erhöht, denn das *erste nicht mehr druckbare* Zeichen ist von Interesse. Nach dem Compare-Befehl wird es auf seinen alten Wert zurückgesetzt. Die beiden Bytes, die durch die Befehle INC (HL) und DEC (HL) verschlissen werden, werden anschließend durch eine weniger dilettantische Verzweigungslogik wieder reingeholt. Ein JR-Befehl genügt, wie man sieht.

So braucht man das geänderte SYS14/SYS nicht neu zu assemblieren. Die wenigen Bytes der Modifikation können mit SUPERZAP oder DDE eingegeben werden:

SYS14/SYS, rel. Sektor 0004, ab rel. Byte 53
BE E1 38 02 28 wird zu
34 BE 35 E1 30

Arnulf Sopp

Schon wieder SYS14/SYS

Nach meinem Beitrag "Ein Fehler in SYS14/SYS" schrieb mir Alexander Schmid, daß der Fehler in seiner PRINT- bzw. LIST-Routine überhaupt nicht auftrat. Mag sein, daß ich den fehlerträchtigen Sprungbefehl JR Z,m514b in grauer Vorzeit selbst einmal eingepatcht habe. Wenn bei euch dort JR NZ,m514b steht, ist alles in Butter.

Trotzdem arbeitet SYS14 auch dann noch unbefriedigend. Ist einmal der PRINT-Befehl eingegeben, gehen alle folgenden Ausgaben, *auch die von LIST*, auf den Drucker. Zum zweiten ist es albern, alle Zeichen durch einen Punkt zu ersetzen, die größer oder gleich dem Byte gemäß der SYSTEM-Option AX=n sind. Wirklich riskant ist nur der Code 7Fh, weil er das DEL-Zeichen des Druckers ist. Er würde das vorangegangene Zeichen löschen. Bei nicht IBM-kompatiblen Druckern kämen noch die Codes 80-9Fh hinzu, die dort i. a. eine Wiederholung der Steuerzeichen 00-1F sind. Alle restlichen Zeichen sollten getrost dargestellt werden dürfen.

Das gilt für den Drucker. Bei der Bildschirmausgabe ist 7Fh durchaus auch ein erlaubtes Zeichen. Aber die Codes ab C0h bedeuten für die Video-Routine eine unterschiedlich lange Kette von Blanks (space compression). Also soll der Fernseher alle Codes von 20-BFh anzeigen. Wegen dieses Unterschieds müssen daher LIST und PRINT beim Einsprung ihre eigenen Duftmarken im Programm setzen:

Der Akku wird mit dem LSB der Treiberadresse des Ausgabekanals geladen. Das ist für PRINT 3Bh für das Programm an 003Bh und für LIST 33h für die Routine an 0033h. Da dies bisher nur für PRINT geschah, war nach dem ersten Ansprung immer der Drucker selektiert.

Die Ausgrenzung der unerlaubten Zeichen ist ähnlich einfach. L wird mit dem DEL-Code 7Fh (PRINT) bzw. mit C0h (LIST) geladen. Wenn gedruckt werden soll, muß *genau bei 7Fh* statt dessen ein Punkt gesetzt werden. Das bedeutet JR Z,m514b (Opcode 28h). Bei der Bildschirmanzeige soll dies *ab C0h* geschehen. Der Sprungbefehl muß daher in diesem Falle JR NC,m514b lauten (Opcode 30h). Deshalb wird H beim Einsprung mit 28h (PRINT) oder 30h (LIST) geladen. H und L werden natürlich gemeinsam als HL beaufschlagt. Wenn nun HL an das Label (exclcod) geschrieben wird, sind das Exklusivbyte und der Sprungbefehl in einem Aufwaschen dem Ausgabekanal angepaßt.

Der alte Befehl AND 7Fh, der das Bit 7 löschte, ist in dieser Version nicht mehr enthalten. Es ist nicht einzusehen, weshalb z. B. aus einem Graphikblock oder einem inversen Zeichen ein gewöhnlicher Buchstabe werden soll. Mißverständnisse wären nicht auszuschließen. Wer jedoch einen nicht IBM-kompatiblen Drucker hat, sollte das AND 7Fh belassen. Auf diese Weise wird das Problem umgangen, daß die Codes 80-9Fh in ihrer Funktion identisch sind mit 00-1Fh. Eine Alternative dazu ist im kurzen zweiten Listing vorgeschlagen. Sie läßt alle Zeichen 20-7Eh und A0-FFh durch. Das hat den Vorteil, daß nun beispielsweise BFh eindeutig von 3Fh unterschieden werden kann.

Die hier vorgestellten Änderungen in SYS14/SYS sind nicht mehr durch Patchen von ein paar Bytes zu erledigen. Die Datei muß disassembliert, modifiziert und neu assembliert werden. Um dazu eine Hilfe zu geben, sind im großen Listing die reinen Datensequenzen aus SYS14 mit ausgedruckt, obgleich sie mit LIST oder PRINT nichts zu tun haben. Sie helfen dem Leser, Programm- von Datencode zu unterscheiden. Für den Disassembler DSMBLR/CMD kann mit diesen Informationen ein Screening-File geschrieben werden, um an passender Stelle DEFM, DEFB und DEFW zu erhalten.

Arnulf Sopp

```
4D17 0D              DEC   C             ;LIST
4D18 CACD50          JP    Z,m50d8
4D1B 0D              DEC   C             ;PRINT
4D1C CAC650          JP    Z,m50d3
```
mit LIST OFF ausgespart

```
4F3B 30     m4f48    DM    '0000H',03h   ;Puffer für Wert von 'ST=nn'
4F41 20     m4f4e    DM    ' AN ',03h    ;Bindewort für Routing
4F46 54     m4f53    DM    'TA',00h      ;Tastatur
4F49 C1              DB    0c1h
4F4A 1540            DW    m4015         ;Adresse des Tastatur-DCB
4F4C 4D              DM    'MO',00h      ;Monitor
4F4F C2              DB    0c2h
4F50 1D40            DW    m401d         ;Adresse des Bildschirm-DCB
4F52 44              DM    'DR',00h      ;Drucker
4F55 C2              DB    0c2h
4F56 2540            DW    m4025         ;Adresse des Drucker-DCB
4F58 4E              DM    'NL',00h      ;Niemandsland
4F5B 43              DB    43h
4F5C FD4C            DW    m4cfd         ;Treiberadresse für NL-Routing
4F5E 4B              DM    'KEINE',00h   ;Routing aufheben
4F64 90              DB    90h
4F65 0000            DW    0000h
4F67 53     m4f74    DM    'ST=',00h     ;Speicherstelle zur Ausgabepufferung
4F6B 60              DB    60h
4F6C 00              DB    00h           ;Flag für ST-Option
4F6D 0000            DW    0000h         ;Füllstoff auf 32 Bytes
```

```
5031 4C     m503e    DM    'LOG='
5035 0000            DW    0000h,m4fa8
5039 41              DM    'ANZ='
503D 0000            DW    0000h,m4fb6
5041 41              DM    'ADE='
5045 0080            DW    8000h,m4f92
5049 41              DM    'ADF='
504D 0040            DW    4000h,m4f92
5051 00              DB    00h
```

```
50AB 56     m50b8    DM    'VON='        ;Text für Untergrenze
50AF 0000            DW    0000h,m50a3   ;Wert wird in 50A3 abgelegt
50B3 42              DM    'BIS='        ;Text für Obergrenze
50B7 0000            DW    0000h,m509c   ;in 509C
50BB 48              DM    'HIMEM='      ;Text für Himem-Festlegung
50C1 0000            DW    0000h,m508c   ;in 508C
50C5 00              DM    00h           ;Ende-Kenner

            ;Lib-Befehl PRINT
50C6 3E3B   m50d3    LD    A,3bh         ;LSB von 003B, Druckroutine
50C8 217F28          LD    HL,287fh      ;ASCII-Zeichen 7fh (DEL), Opcode JR Z
50CB 1805            JR    listprt       ;dort weiter

            ;Lib-Befehl LIST (wird auch von PRINT benutzt)
50CD 3E33   m50d8    LD    A,33h         ;LSB von 0033h, Bildschirmroutine
50CF 21C030          LD    HL,30c0h      ;Zeichen C0h (Space-Compr.), Opc. JR NC
50D2 326851 listprt  LD    (m5170),A     ;LSB des Ausgabekanals dort patchen
50D5 223A51          LD    (exclcod),HL  ;dto. nicht druckbare/anzeigbare Zeichen
50D8 F1              POP   AF            ;? (nur Stack-Ebene löschen?)
50D9 E1              POP   HL            ;?
50DA CDD54C          CALL  m4cd5         ;den Befehlsstring weiter untersuchen
```

```
512F D5              PUSH  DE
5130 118044          LD    DE,m4480      ;FCB-Adresse
5133 CD1300          CALL  m0013         ;1 Byte aus der Datei lesen
5136 D1              POP   DE
5137 200D            JR    NZ,m514e      ;falls ein Fehler auftrat
```

```
5139 FEC0            CP    0c0h          ;erstes nicht mehr anzeigbares Zeichen
513A        exclcod  EQU   $-1           ;oder 7Fh (DEL-Code für den Drucker)
513B 3006            JR    NC,m514b      ;Punkt, falls kein anzeigbares Zeichen
513D FE20   m5145    CP    ' '           ;Blank oder kleiner?
513F D0              RET   NC            ;zurück, falls anzeig-/druckbares Zeichen
5140 FE0D            CP    0dh           ;CR?
5142 C8              RET   Z             ;so ausgeben, falls ja
5143 3E2E   m514b    LD    A,'.'         ;andere Zeichen durch Punkt ersetzen
5145 C9              RET                 ;zurück

00000 Fehler
```

```
0000 F5     m5145    PUSH  AF            ;Zeichen im Stack puffern
0001 08              EX    AF,AF'        ;alternativer Akku
0002 F1              POP   AF            ;diesen mit dem Original-Byte laden
0003 E67F            AND   7fh           ;Bit 7 löschen
0005 FE20            CP    ' '           ;Blank oder kleiner bzw. 80-9Fh?
0007 3802            JR    C,check0d     ;00-1Fh bzw. 80-9Fh durch Blank ersetzen
0009 08              EX    AF,AF'        ;Original-Byte restaurieren
000A C9              RET                 ;zurück, Byte ausgeben
000B FE0D   check0d  CP    0dh           ;CR?
000D C8              RET   Z             ;so ausgeben, falls ja
000E 3E2E   m514b    LD    A,'.'         ;andere Zeichen durch Punkt ersetzen
0010 C9              RET                 ;zurück

00000 Fehler
```

Binärumwandlung aus Hex und Dez

und kürzere E-Routine in SYS14/SYS

Um einen Fehler bei den Library-Befehlen LIST und PRINT zu beheben, untersuchte ich SYS14/SYS (s. dazu "Ein Fehler in SYS14/SYS"). Es versteht sich, daß ich das File ganz allgemein mal belauerte. Dabei fand sich eine interessante Routine, die einen ASCII-String, auf den HL zeigt, binär nach DE lädt. Dabei muß es sich um eine Hex- oder Dez-Eingabe aus gültigen Zeichen handeln. Hex-Zahlen müssen mit 'H' abgeschlossen sein. Bei einem ungültigen Zeichen oder fehlendem Hex-Kenner erfolgt die Fehlermeldung "schlechte Parameter".

Es besteht so oft der Bedarf dafür, daß sich die Vorstellung dieser Routine lohnt. Wer einen Computer mit mehreren Banks oder mit freiem Speicherplatz "hinter" dem Bildschirm oder der Tastatur hat, kann das Programm sogar sofort beim Booten laden, um es als Teil des Betriebssystems immer parat zu haben. Es verbraucht 80 Bytes (allerdings erst, nachdem ich es um einigen überflüssigen Quark erleichtert habe).

Den Kommentaren im Listing merkt man an, daß ich den Algorithmus selbst nicht 100%ig durchschaue. Mit aller gebotenen Vorsicht gehe ich davon aus, daß das Segment ab m4f26 nur dann zweimal durchlaufen wird, falls es sich um eine Hex-Eingabe handelt. Zumeist sage ich im Listing nicht "Hex", sondern "2. Durchlauf". Für die dennoch bei aller Unsicherheit durchgeführten Änderungen (das Original möchte ich aus Platzgründen nicht vorstellen) lege ich trotzdem meine Hand ins Feuer.

Beim Einsprung muß HL als Zeiger auf den ASCII-String geladen sein. Bis auf C werden alle Register benutzt (gegf. retten!). Beim Aussprung deutet BC auf den Anfang, HL auf das Ende der ASCII-Eingabe, DE enthält das binäre Äquivalent.

Es wäre interessant, im nächsten Info zu lesen, wie die Besitzer geeigneter Computer (s. o.) die Routine fest in ihr System eingebaut haben.

Oben im Listing erscheint die Service-Routine des Library-Befehls E (ERROR bei NewDOS). Sie wurde ein paar Bytes kürzer und damit schneller. Da wir gerade dabei sind, sei sie ebenfalls präsentiert. Die ausgelassenen Teile des SYS-Files sind an den springenden Zeilennummern zu erkennen. SYS14/SYS kann gefahrlos gem. dem Listing geändert und neu assembliert werden. Daß die Label-Namen zu den tatsächlichen Ladeadressen wegen der Kürzungen nur noch in einer ungefähren Beziehung stehen, ist bedeutungslos.

Arnulf Sopp

```
                00076 ;Lib-Befehl E
4D39  CDEC4E    00077 m4d39  CALL  m4eef     ;Error-Code einlesen
4D3C  CDD54C    00078        CALL  m4cd5     ;folgt etwas dem Code im Befehlsstring?
4D3F  C0        00079        RET   NZ        ;raus mit Fehler, falls ja (darf nicht)
4D40  EB        00080        EX    DE,HL     ;HL <- Error-Code
4D41  114000    00081        LD    DE,0040h  ;höchstmöglicher Fehlercode
4D44  DF        00082        RST   18H       ;Überschritten?
4D45  D20A4F    00083        JP    NC,m4f0d  ;"schlechte Par.", falls ungült. Code
4D48  7D        00084        LD    A,L       ;A <- Fehlercode (NZ-Bedingung erfüllt)
4D49  C9        00085        RET             ;Aussprung mit Fehleranzeige
                00086
                00375 ;UP liest Zahleneingaben aus (HL) nach DE ein
4EEC  E5        00376 m4eef  PUSH  HL        ;Befehlszeiger retten
4EED  0600      00377        LD    B,00h     ;Flag: 1. Durchlauf des UP (Dez und Hex)
4EEF  CD0F4F    00378        CALL  m4f12     ;Eingabe einlesen und binär umwandeln
```



```
4EF2  7E        00379        LD    A,(HL)    ;letztes Zeichen nochmals einlesen
4EF3  D641      00380        SUB   'A'       ;binär umwandeln
4EF5  FE08      00381        CP    08h       ;07 wäre Hexkenner 'H' (maximal erlaubt)
4EF7  300D      00382        JR    NC,m4f09  ;"schlechte Par.", falls unerlaubtes Z.
4EF9  E1        00383        POP   HL        ;Befehlszeiger
4EFA  0601      00384        LD    B,01h     ;Flag: 2. Durchlauf des UP m4f12
4EFC  E5        00385        PUSH  HL        ;Befehlszeiger wieder retten
4EFD  CD0F4F    00386        CALL  m4f12     ;weitere Eingabe einlesen und umwandeln
4F00  7E        00387        LD    A,(HL)    ;letztes Zeichen nochmals einlesen
4F01  FE48      00388        CP    'H'       ;Kenner für Hex-Eingabe?
4F03  23        00389        INC   HL        ;nächste Stelle
4F04  2004      00390        JR    NZ,m4f0d  ;ohne Hex-Kenner: "schlechte Parameter"
4F06  CB48      00391 m4f09  BIT   1,B       ;wurde das UP m4f12 zweimal durchlaufen?
4F08  C1        00392        POP   BC        ;Stack bereinigen
4F09  C0        00393        RET   NZ        ;falls 2. Durchlauf (Hex), andernfalls:
4F0A  3E2F      00394 m4f0d  LD    A,2fh     ;Fehlercode "schlechte Parameter"
4F0C  C30944    00395        JP    m4409     ;Fehler anzeigen, Rückkehr zum Aufrufer
4F0F  110000    00396 m4f12  LD    DE,0000h  ;Ausgangswert für Binärzahl
4F12  7E        00397 m4f15  LD    A,(HL)    ;Zeichen einlesen
4F13  D630      00398        SUB   '0'       ;ASCII -> binär
4F15  FE0A      00399        CP    0ah       ;höher als Dez-Ziffer? (evtl. Hex-Ziffer)
4F17  380A      00400        JR    C,m4f26   ;falls Dez-Ziffer (erlaubtes Zeichen)
4F19  CB40      00401        BIT   0,B       ;ist es der 2. Durchlauf des UP?
4F1B  C8        00402        RET   Z         ;fertig, falls nein (1. Durchlauf, Dez)
4F1C  D611      00403        SUB   11h       ;falls Hex-Ziffer: 'A' -> 00 usw.
4F1E  FE06      00404        CP    06h       ;noch erlaubte Hex-Ziffer? ('F' = 05)
4F20  D0        00405        RET   NC        ;Ende, falls nicht 'A' ... 'F'
4F21  C60A      00406        ADD   A,0ah     ;'A' ... 'F' -> 0A ... 0F
4F23  E5        00407 m4f26  PUSH  HL        ;Umwandlung ok., Befehlszeiger retten
4F24  62        00408        LD    H,D       ;HL <- Zwischenergebnis (MSB)
4F25  6B        00409 -      LD    L,E       ;(LSB)
4F26  CBC8      00410        SET   1,B       ;Flag: UP m4f12 wurde ganz durchlaufen
4F28  29        00411        ADD   HL,HL     ;*2 (Zwischergebnis links rotieren)
4F29  29        00412        ADD   HL,HL     ;*4 (um zwei Stellen)
4F2A  CB40      00413        BIT   0,B       ;2. Durchlauf des UP? (bei Hex-Eingabe)
4F2C  2803      00414        JR    Z,m4f38   ;falls nein (Dez-Eingabe)
4F2E  29        00415        ADD   HL,HL     ;*8 (Hex-Eingabe)
4F2F  1801      00416        JR    m4f39     ;dort weiter
4F31  19        00417 m4f38  ADD   HL,DE     ;plus Zwischenergebnis (Dez: *5)
4F32  29        00418 m4f39  ADD   HL,HL     ;Hex: *16, Dez: *10
4F33  5F        00419        LD    E,A       ;DE <- letzte Ziffer (Einerstelle)
4F34  1600      00420        LD    D,00h     ;als 16-Bit-Zahl (MSB = 00)
4F36  19        00421        ADD   HL,DE     ;Endergebnis
4F37  EB        00422        EX    DE,HL     ;DE <- Endergebnis
4F38  E1        00423        POP   HL        ;Befehlszeiger
4F39  23        00424        INC   HL        ;nächste Stelle im Befehlsstring
4F3A  18D6      00425        JR    m4f15     ;dort weiter

00000 Fehler
```

Was tun bei zerschossener Source?

Gestern passierte folgendes: Die (vermutlich in diesem Heft) vorgestellte Routine zur binären Umwandlung von Zahleneingaben in Hex oder Dez wollte ich in OVL4/SYS einbauen. OVL4 ist die Datei, die aus dem ziemlich universellen G-DOS 2.4 das spezielle DOS des Genie 3s macht. So wäre dann auch diese Routine ein fester Bestandteil meines Betriebssystems geworden.

Frisch ans Werk, ZEUS angeschmissen, den früher einmal unter großem Aufwand erstellten Quellcode OVL4/SRC geladen. Frust! Das File hörte mittendrin auf. 126 Sektoren Quellcode, das waren viele Tage Analyse, ebenso viel Modifikationsarbeit, eine Menge Gehirnschmalz, alles im Eimer! Mit DDE mußte ich feststellen, daß an zwei Stellen insgesamt 6 Sektoren durch ASCII-Text überschrieben waren. Da stand etwas, das TSCRIPS dort abgelegt hatte. Sobald ZEUS den ersten ihm nicht mehr plausiblen Quark erkannt hatte, hörte er mit dem Laden einfach auf und betrachtete den arg verstümmelten Quellcode als vollständig.

Die Reparatur war ein bißchen kompliziert, aber durchaus machbar. Zunächst mußte OVL4/SYS erneut disassembliert werden. Die überschriebenen Teile waren nämlich unwiederbringlich gelöscht; was dort gestanden hatte, mußte rekonstruiert werden. In dieser neuen Source konnten mit Delete-Befehlen die verlorenen Teile isoliert und erneut abgesavet werden. Die noch intakten hinteren Sektoren der zerstörten Source hatte ZEUS gar nicht mehr erreicht. Sie mußten ihm also zugänglich gemacht werden. Andernfalls hätte ich von dort an in der neuen Source alles neu analysieren und kommentieren müssen. Nein, danke!

Das ging so: Für die intakten Teile wurde ein Dummy-File mit DUMP erzeugt. Es hatte dieselbe Anzahl von Sektoren wie die zu rettenden Teile der alten Source. In diese Blinddatei wurde dann mit der CDS-Option von SUPERZAP einkopiert, was noch zu retten war. Damit hätte ZEUS aber noch nicht viel anfangen können. So ein Sektor fängt ja nicht ausgerechnet mit einer neuen Zeile an. ZEUS wäre daher kaum zu veranlassen, ihn zu laden. Die ersten 16 Bytes eines Beispielsektors sehen so aus:

```
001E00: 6869 726D 2067 6573 7072 6569 7A74 3F0D  hirm gespreizt?.
```

Es ist der rel. Skt. 1Eh des Files, wie die linke Zahl verrät. Offensichtlich steht da gerade der Rest des Kommentars ";Bildschirm gespreizt?" mit CR (0Dh) am Ende. Wenn es sich nicht um einen Fortsetzungs-, sondern um einen Anfangssektor handelt, den ZEUS seiner bisherigen Source anfügen soll, muß die Zeile mit einem Zähler für ihre Länge beginnen. Bis zum nächsten 0D (Zeilenende) sind es genau 16 (10h) Bytes. Sodann erwartet ZEUS im nächsten Byte ein Flag für die Art des in dieser Zeile stehenden Befehls. Es sollte unbedingt ein Kommentar sein, weil das völlig ungefährlich ist. Das erkennt ZEUS an einem Nullbyte. Also muß der Beginn des Sektors folgendermaßen geändert werden:

```
000000: 1000 3B6D 2067 6573 7072 6569 7A74 3F0D  ..;m gespreizt?.
```

Der gestreamlinete Sektor beginnt nun mit dem Zähler 10h (Zeilenlänge), dann folgt 00h (kein Befehl, sondern z. B. ein Kommentar), dann ein Semikolon (es *ist* ein Kommentar). Das restaurierte Segment der Source beginnt jetzt mit dem etwas ulkigen Kommentar ";m gespreizt?". Es ist übrigens der rel. Skt. 00h der Kopie ins Dummy-File, das zu diesem Zweck angelegt worden war (linke Zahl).

ZEUS lädt nun zuerst, was aus der kaputten Source noch zu laden ist. Dann wird das Segment der neu erzeugten Disassembly angehängt, das in die überschriebene Lücke gehört. Zuletzt folgt der Rest, der wie gezeigt restauriert wurde. Jedesmal wird beim Ladebefehl auf die Frage "append to buffer?" mit Y (yes) geantwortet. Schließlich steht die komplette Source im

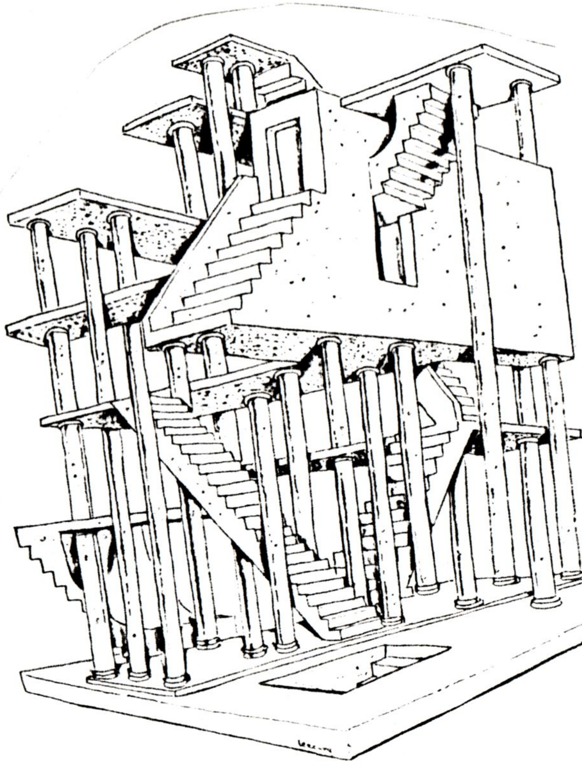
Buffer. Sie kann in dieser Form wieder gesavet werden. Zuvor sollte man freilich den Müll wie etwa den obigen dusseligen Kommentar löschen.

Diese Methode funktioniert auch, wenn der erste wieder brauchbare Sektor nicht einen Kommentar enthält. Was auch immer am Anfang steht, es wird mit dem Zähler bis zum nächsten 0D (CR), einer Null und einem Semikolon überschrieben. Nach dem Laden kann das Listing an dieser Stelle ziemlich witzig aussehen, aber das ist egal. Diese Zeile muß ohnehin als verloren gelten und wird gelöscht. Der nicht deletete Rest der neu erzeugten Disassembly muß natürlich enthalten, was auf die beschriebene Weise verunstaltet wurde.

Ganz zum Schluß mußte für die Lösung meines Problems eben leider doch wieder analysiert und kommentiert werden. Da aber in den verlorenen Sektoren die Kommentare sehr ausführlich gewesen waren, handelte es sich nur um ca. 100 Bytes reinen Programmcode, so daß sich die Arbeit in Grenzen hielt.

Und die Moral von der Geschicht': Wer meine TSCRIPS-Version benutzt, die offenbar überschreibt, was immer ihr beliebt, ist selber schuld. Ich hatte auch schon *billets doux* mitten im DIR gefunden, was den Wert der Diskette schlagartig auf die Materialkosten reduzierte.

Arnulf Sopp



Tja, aber ist die Welt überhaupt reif für computergesteuerte Architektur?

Quatsch in OVL4/SYS

G-DOS 2.4, das Betriebssystem der neueren Genie-Modelle, wird durch ein zusätzliches File, OVLx/SYS, an das jeweilige Computermodell angepaßt. Für das Genie 3s ist OVL4/SYS zuständig. Was hier gezeigt werden soll, gilt für die anderen Overlays und damit die anderen Genies entsprechend, sofern sie einen Video-Controller 6845 CRTC und einen ladbaren Zeichensatz haben.

Dieser Controller wird beim Booten erst programmiert, wenn das Overlay gerade arbeitet. Zuvor wäre es sinnlos, weil der Computer möglicherweise keinen hat (Allgemeingültigkeit von G-DOS 2.4). Dabei geht OVL4/SYS so vor, wie im oberen der beiden Listings auf der nächsten Seite gezeigt (Ansprung beim Label m32cf):

Zum Laden eines Zeichensatzes von der Diskette wird ein FCB eingerichtet. Anschließend wird vorübergehend nach SYS22/SYS verzweigt, wo das eigentliche Laden vonstatten geht. Dann wird Laufwerk 0 angeschmissen (Label m32ec; es soll sich während des Bootvorgangs immer weiter drehen, um für weitere Lesezugriffe nicht erneut gestartet werden zu müssen). Nun erst wird der CRTC programmiert. Dazu werden die drei Codes beim Label m32c0 auf den Bildschirm ausgegeben. Der Code 10h bewirkt die Einstellung auf 64X16 Zeichen, wobei der CRTC entsprechend mit Daten beschickt wird.

Wenn man sein DOS in Frieden läßt, funktioniert das prächtig. Ich habe aber den Parametersatz für 64X16 Zeichen geändert (es würde hier zu weit führen, die Gründe darzulegen). Die Folge war, daß der bereits geladene Zeichensatz beim Programmieren des CRTC plötzlich nicht mehr komplett sichtbar war. Die Oberlängen waren weg (z. B. Punkte auf den Umlauten). Es mußte daher die Reihenfolge in der Vorbereitung des Bildschirms umgekehrt werden: zuerst das Format programmieren, dann den Zeichensatz laden.

Das untere Listing zeigt, wie. Dabei konnten gleich noch ein paar Verbesserungen eingebaut werden, die den Ablauf beschleunigen und das Overlay um ein paar Bytes verkürzen. Zusätzlich wird nun gleich beim Booten das Format 80X25 Zeichen eingestellt, mit dem ich ohnehin fast ausschließlich arbeite.

Die Kürzungen, die zur Ablaufgeschwindigkeit und natürlich dem verringerten Platzbedarf beitragen, sind folgende: Der Zeiger HL braucht nur einmal für die Video-Steuercodes aufgesetzt zu werden. Anschließend steht er bereits automatisch auf dem Namen des Zeichensatzes. Der Aussprung mit XOR A (Z-Flag setzen, um dem System zu signalisieren: "kein Fehler aufgetreten") und RET erübrigt sich ebenfalls, weil die Routine zum Starten des Laufwerks das am Ende schon selber tut. Die restlichen Unterschiede (DE als Zeiger auf den FCB und B als Zähler der LRL laden) sparen zusammen auch noch zwei Bytes.

Daß es in diesem Teil des Overlays jetzt nur noch ein einziges ORG-Statement gibt, ist allerdings ein schon recht alter Hut und hat mit der neuen Änderung nichts zu tun. Schon früher habe ich das File völlig umgeordnet und solche ORGs gelöscht, die sich wegen der zusammenhängenden Programmsegmente von selbst verstanden.

Arnulf Sopp


```
32C0            00027        ORG   32c0h
32C0  10        00028 m32c0  DB    10h,00h,00h      ;Video-Codes: 64X16 Z., nichts, nichts
32C3            00099        ORG   32c3h
32C3  53        00100 m32c3  DM    'STD80:0',0dh,'   ',0dh ;Platz f. Namen des Zeichensatzes
32CF            00336        ORG   32cfh
32CF  21C332    00337 m32cf  LD    HL,m32c3         ;dort Filename des Zeichensatzes
32D2  118044    00338        LD    DE,m4480         ;anzulegender FCB
32D5  CD1C44    00339        CALL  m441c            ;Filenamen in den FCB übertragen
32D8  D9        00340        EXX                    ;Register tauschen
32D9  118044    00341        LD    DE,m4480         ;FCB
32DC  210042    00342        LD    HL,4200h         ;Pufferadresse
32DF  0600      00343        LD    B,00h            ;LRL = 00 = 256
32E1  CD2344    00344        CALL  m4423            ;FCB eröffnen
32E4  2006      00345        JR    NZ,m32ec         ;falls Fehler aufgetreten
32E6  D9        00346        EXX                    ;Register zurücktauschen
32E7  3E78      00347        LD    A,78h            ;SYS22: Zeichensatz laden
32E9  CD0244    00348        CALL  m4402            ;ausführen (dort steht RST 28)
32EC  AF        00349 m32ec  XOR   A                ;Drive 0
32ED  CD7647    00350        CALL  m4776            ;starten
32F0  0603      00351        LD    B,03h            ;3 Bildschirm-Steuercodes
32F2  21C032    00352        LD    HL,m32c0         ;dort stehen sie
32F5  7E        00353 m32f5  LD    A,(HL)           ;einen laden
32F6  CD3300    00354        CALL  0033h            ;ausgeben
32F9  23        00355        INC   HL               ;nächster Code
32FA  10F9      00356        DJNZ  m32f5            ;bis erledigt
32FC  AF        00357        XOR   A                ;Z-Flag: kein Fehler
32FD  C9        00358        RET                    ;zurück

00000 Fehler
```

```
32C0            00310        ORG   32c0h
32C0  13        00311 m32c0  DB    13h,00h,00h      ;Video-Codes: 80X25 Z., nichts, nichts
32C3  53        00312 m32c3  DM    'STD80:0',0dh,'   ',0dh ;Platz für Zeichensatz-Namen
32CF  0603      00313 m32cf  LD    B,03h            ;3 Codes für das Bilschirmformat
32D1  21C032    00314        LD    HL,m32c0         ;dort stehen sie
32D4  7E        00315 m32f5  LD    A,(HL)           ;einen laden
32D5  CD3300    00316        CALL  0033h            ;ausgeben
32D8  23        00317        INC   HL               ;nächster Code
32D9  10F9      00318        DJNZ  m32f5            ;bis erledigt
32DB  118044    00319        LD    DE,m4480         ;anzuleg. FCB (HL zeigt auf den ZS-Namen)
32DE  D5        00320        PUSH  DE               ;wird gleich noch gebraucht
32DF  CD1C44    00321        CALL  m441c            ;Filenamen in den FCB übertragen
32E2  D9        00322        EXX                    ;Register tauschen
32E3  D1        00323        POP   DE               ;FCB
32E4  210042    00324        LD    HL,4200h         ;Pufferadresse
32E7  45        00325        LD    B,L              ;LRL = 00 = 256
32E8  CD2344    00326        CALL  m4423            ;FCB eröffnen
32EB  2006      00327        JR    NZ,m32ec         ;falls Fehler aufgetreten
32ED  D9        00328        EXX                    ;Register zurücktauschen
32EE  3E78      00329        LD    A,78h            ;SYS22: Zeichensatz laden
32F0  CD0244    00330        CALL  m4402            ;ausführen (in 4402 steht RST 28)
32F3  AF        00331 m32ec  XOR   A                ;Drive 0
32F4  C37647    00332        JP    m4776            ;starten und zurück

00000 Fehler
```

m4/4p ecke m4/4p ecke

------------------------------------ 1 ------------------------------------

Das neueste DOS für den TRS 80 mod.4/4P

Wie ich aus diversen Bemerkungen in den Info's entnehmen konnte scheint Tandy in Deutschland nicht mehr sehr aktiv zu sein.
Nicht so in Frankreich. Es gibt in jeder größeren Stadt eine Tandy-Vertretung (etwa 200), welche außer Hi-Fi Geräten, Elektronikteilen auch noch Tandy Computer verkaufen. Natürlich werden da hauptsächlich die neuesten Geräte 1000, 2000,und 3000 angeboten. Aber es gibt in der Zentrale, in Paris, noch Leute, welche sich für die älteren Modelle verantwortlich fühlen.

Von dort bekam ich kostenlos (!) eine Diskette mit dem neuesten LS-DOS 6.3 F. Das "F" bezieht sich auf die Folge der Datumsangabe (TT/MM/JJ) und auf den französischen Tastaturtreiber. Deshalb habe ich vorsichtshalber eine Kopie der Diskette an Klaus Hermann geschickt. Er hat das entsprechende SYS-File ausgewechselt und nun, hoffe ich, daß er das neue DOS allen mod.4/4P-Besitzern zugänglich macht.

Die gesamte französische Beschreibung ist 29 Seiten lang. Ich habe deshalb vorerst die wichtigsten Teile in Kurzform übersetzt. Was noch übrig bleibt bezieht sich auf den Betrieb mit Festplatten und einige Aenderungen bei den SVC's. Sollte später dafür Bedarf bestehen, so bin ich gerne bereit auch noch den Rest zu übersetzen.

Helmut Obscherningkat

------------------------------------ 2 ------------------------------------

LS - DOS 6.3 F   Neue Ueberarbeitung vom TRSDOS 6.2 für TRS-80 mod 4/4P von TANDY - Frankreich.

Uebersetzung in Kurzform der wesentlichen Neuerungen

Erste Inbetriebnahme :

Es ist zu empfehlen zuerst einige Kopien der Systemdiskette anzufertigen.
Hierzu die Systemdiskette ins Laufwerk 0
eine leere Diskette ins Laufwerk 1
Es meldet sich LS-DOS 6.3
Datum (TT/MM/JJ) und Uhrzeit (HH:MM:SS) eingeben.
Uhrzeit kann mit <ENTER> übersprungen werden.

dann DISKCOPY :0 :1 <ENTER>

Die Leerdiskette wird formatiert und vollkommen kopiert.
Anschliessend wird gefragt, ob weitere Disketten zu kopieren sind.
Mit Y/N beantworten.

A C H T U N G ! ! !
Beim Uebergang vom TRS-DOS 6.2 zum LS-DOS 6.3 und umgekehrt unbedingt alle Disketten aus dem Gerät nehmen, RESET drücken und von vorn beginnen. Ich habe schon unangenehme Ueberraschungen erlebt und war froh noch Ersatzkopien zu haben. Am besten läßt man die jeweilige Systemdiskette immer bis zum Schluß im Laufwerk 0.

Aenderungen der DOS-Befehle bzw. neue DOS-Befehle

ATTRIB - Das Passwort des Anwenders (User) fällt weg.
- Das Passwort des Besitzers (OWNER) bleibt

DATE - Das System verarbeitet nun Kalenderdaten bis zum 1. Januar 1999

DIR - Hinter dem Datum wird noch die Uhrzeit angegben (12:00am)

TIME - Die Zeit kann mit folgender Syntax eingegeben werden :
TIME<HH:MM<:SS>><(CLOCK=ON/OFF)>

DATECONV/CMD
- Verändert Data-Disketten (keine Systemdisketten) älterer Version zur Verwendung des neuen Zeit/Datum-Formats.
Syntax : DATECONV :d

DISKCOPY/CMD
- Syntax wie oben beschrieben. s und d dürfen nicht die gleiche Laufwerknummer sein.

BACKUP - Verändert Systemdisketten 6.2 und älter in Datum und Zeit.
Syntax : BACKUP :s :d(S,I,OLD)

Um die neuen BASIC-Befehle zu überspielen :
- BACKUP BASIC:0 :1(I) <ENTER>
- DATECONV:1 <ENTER>

Es folgt dann noch eine Abhandlung um das neue LS-DOS auf Festplatten zu verwenden. Wenn gewünscht, könnte ich auch hier eine Uebersetzung liefern.

BASIC 01.01.02 Aenderungen und neue Befehle

RENUM hat jetzt 4 Positonen.
RENUM 1000,500,10,600 bedeutet, daß alle Zeilen zwischen 500 und 600 nach 1000 mit dem Inkrement 10 verschoben werden.

Beim Auslisten von BASIC-Programmen werden 6 Tasten zusätzlich verwendet :
- Pfeiltasten nach oben bzw. nach unten zum Anzeigen der vorhergehenden bzw. folgenden Zeile. (ENDLICH)
- Der Punkt "." zeigt die bearbeitete Zeile
- Das Komma "," zeigt die augenblickliche Zeile in mode EDITION.

Im neuen BASIC werden 4 Befehle durch nur einen Buchstaben ausgelöst :

A = AUTO
D = DELETE
E = EDIT
L = LIST

Dabei ist die Leerstelle überflüssig. Also L100-300 ist gleich L 100-300.

Die 4 folgenden Befehle sind im File BASIC/OV2 enthalten. Ist dies nicht geladen, wird mit File not found quittiert.

C Num1,Num2	= Copie einer Zeile Num1 ist die existierende Zeile Num2 ist die neue Zeile
F Object	= Suchen einer Zeile nach einer Variablen oder Schlüsselwort
M Num1,Num2,Num3	= Verschieben eines Blocks Num1 und Num2 sind existierende Zeilennummern und definieren Anfang und Ende des Blocks Num3 ist die Zeilennummer der neuen Position.
S Object oder S	= Suchen und Anzeigen einer Zeile, Variablen oder eines Schlüsselwortes

Die Befehle SAVE und LOAD werden 3 mal schneller ausgeführt sofern sie NICHT in ASCII geladen worden sind.

"Cross Référence" für BASIC - Programme

Aufzurufen mit BREF/CMD vom DOS.
Das zu untersuchende Programm darf nicht in ASCII geladen sein.
Syntax :

BREF filespec<(param,param,...)>

Die Parameter sind

VAR=ON/OFF	Standardwert : ON
LINE=ON/OFF	Standardwert : OFF
P=ON/OFF	Standardwert : OFF
W=n	Druckbreite Standardwert : 80

filespec ist der Name des zu untersuchenden BASIC-Programms. Parameter VAR listet die Variablen auf. Parameter VAR fügt die Zeilennummern hinzu. Parameter P druckt die Liste aus. Parameter W bestimmt Druckbreite des Druckers (Normal 80 bis 132), jedoch sind Werte zwischen 32 und 255 möglich.

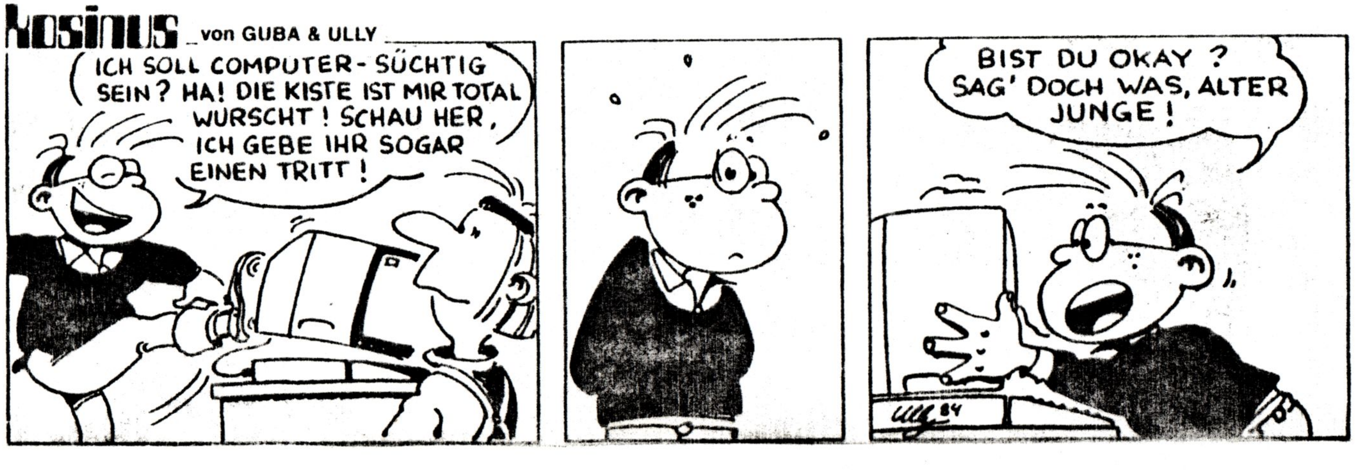
Die Variablennamen werden bis zu 14 Zeichen ausgegeben, der Rest fällt unter den Tisch.

BREF kann drei Fehlermeldungen zeigen :

Not a BASIC program
Das Programm existiert nicht, ist geschützt oder ist in ASCII geladen.

Out of memory - can't cross reference
Kann bei sehr langen Programmen vorkommen. Dann in zwei oder mehreren Teilen bearbeiten.

Line nnnn, Error in original program
Wenn das zu bearbeitende Programm einen Syntaxfehler enthält.



m4/4p ecke m4/4p ecke

---------------------------------- 5 ----------------------------------

Auf der gleichen Diskette befindet sich noch ein Textverarbeitungsprogramm

Es handelt sich um ein abgemagertes SCRIPSIT, ist aber doch besser als viele BASIC-Programme.

Aufruf vom DOS mit TED. Dann gelten folgende Befehle :

Cursor nach links	<-- oder <CTRL><H>
Cursor nach rechts	--> oder <CTRL><I>
Cursor nach unten	↓ oder <CTRL><J>
Cursor nach oben	↑ oder <CTRL><K>
Cursor an den Anfang der Zeile	<SHIFT> <--
Cursor ans Ende der Zeile	<SHIFT> -->
Cursor ans Ende des Textes	<SHIFT> ↓
Cursor an de Anfang des Textes	<SHIFT> ↑
Umschalten überschreiben/einfügen	<CTRL><A>
Blockmarkierung	<CTRL><B>
Löschen	<CTRL><D>
Text in einem File sichern	<CTRL><F>
Sucht die folgende Uebereinstimmung mit einer Zeichenkette	<CTRL><G>
Ladet einen File	<CTRL><L>
Nächste Zeile	<ENTER> oder <CTRL><M>
Cursor zur folgenden Seite	<CTRL><N>
Ersetzt eine Zeichenkette durch eine andere	<CTRL><R>
Sucht eine Zeichenkette	<CTRL><S>
Cursor an den Anfang der vorherigen Zeile	<CTRL><U>
Zurück zum LS-DOS	<CLEAR><SHIFT><=>

Die eigentliche Beschreibung ist zu lang. Bitte durch Ausprobieren herausfinden, oder bei mir anfragen.
Nachteile gegenüber SCRIPSIT : Verarbeitet nur ASCII-Zeichen also keine Umlaute, ebenso keine franz. accents.
Außerdem ist unverständlicherweise kein Befehl zum Ausdrucken vorgesehen. Also ausdrucken nur vom DOS mit LIST file.

SHELL/CMD - Eine Benutzeroberfläche unter TRSDOS / LSDOS

Wer unter NEWDOS mit MENUE oder unter CP/M mit CPMPOW arbeitet, wird sich sicher auch mit SHELL anfreunden.

SHELL ist ein Freeware Programm von Stephen Milliken. Es arbeitet unter TRSDOS 6.2 und kann eigentlich recht viel. Beim Umgang mit dem Betriebssystem nimmt es einge Menge Tipparbeit ab. Es erspart die Eingabe von einigen Betriebssystembefehlen, die hier durch Tastendruck aufgerufen werden. Außerdem bietet es noch einen einfachen Texteditor. Hier nun einige Möglichkeiten unter SHELL:

- sämtliche Files werden auf dem Bildschirm angezeigt und können mit Pfeiltasten angefahren und aufgerufen werden.
- Betriebssystembefehle werden per Tastendruck angewählt und ausgeführt.
- Der Bildschirm wird nach 2 Min. ausgeblendet und erscheint erst wieder auf Tastendruck
- Auswahl der anzuzeigenden Files mit Deskriptoren möglich
- per SYSGEN wird SHELL sofort nach dem Booten initialisiert.

SHELL wurde für TRSDOS 6.2 geschrieben. Ruft man es unter einer anderen Version auf, wird man aufgefordert, TRSDOS 6.2 zu laden.

Eine bzw. zwei Möglichkeiten mit SHELL auch unter LS-DOS 6.3 zu arbeiten, wurden in der 80Micro gezeigt.
1. Von Basic aus wird der Befehl *POKE 133,98* eingegeben. Dies bewirkt, daß im Speicher die Betriebssystem Versionsnummer von 6.2 auf 6.3 geändert wird (80Micro, Mai 1987)
2. Der SYS0-File wird gepatcht.
PATCH SYS0/SYS.LSIDOS (D00,91=62:F00,91=63)
Auch hier wird die aktuelle Version snummer (6.3) auf 6.2 abgeändert.

Der Nachteil beider Verfahren besteht nun aber darin, daß ein anderes Programm, das ebenfalls auf die Versionsnummer zugreifen will, nun eine falsche Nummer mitgeteilt bekommt.

Dies kann umgangen werden, wenn man SHELL selbst ändert. Auch hier gibt es zwei Möglichkeiten:

1. SHELL vergleicht die im Programm enthaltene Nummer (62) mit der im SYS0-File. Man muß nun die Versionsnummer in SHELL auf 63 abändern. Diese Nummer steht im Record \`0\` unter der Adresse \`F1\` (Ladeadresse 9EED). SHELL akzeptiert nun Betriebssysteme mit der Versionsnummer 6.3.

2. Die gesamte Routine, die die Versionsnummer abfragt, wird gelöscht und mit *NOP*-Befehlen ausgefüllt. Die Versionsnummer wird somit nicht mehr abgefragt und SHELL kann unter mehreren Versionen arbeiten. Die gesamte Routine steht im Record \`0\` unter der Adresse \`ED - EF\` (Ladeadresse 9EE9 - 9EFB).

Auf dem weiter unten stehenden Ausdruck des Records \`0\` werden beide Möglichkeiten nochmals aufgezeigt.
Welche der Möglichkeiten nun die Beste ist, soll nicht gefragt sein. Sicherlich hat jede Art ihre Vor- und Nachteile. Hauptsache ist jedoch, daß dieses Programm auch weiterhin benutzt werden kann.
Auf der nächsten Seite habe ich noch die Benutzeroberfläche und das Helpmenü ausgedruckt.

Sollte jemand Interesse an dem Programm haben, bin ich gerne bereit, die geänderten Versionen zuzusenden (Diskette und Rückporto bitte nicht vergessen).

Klaus Hermann

Dies ist die Benutzeroberflaeche von SHELL/CMD

```
SHELL/CMD ver.1.8a Copyright 1985 Stephen Milliken-Randolph, MA
*****************************************************************************


*****************************************************************************
DRIVE:2 - Page: 1 MEMDISK FREE SPACE: 56K Extension=/$$$ A ,S, 0
*****************************************************************************
BOOT/SYS       SI ! DIR/SYS       SI ! PRINTER/DAT     ! MP/CMD
PLOTTER/BAS       ! TEXT/CMD         ! START/JCL       !




*****************************************************************************
ENT,Drv 0-7, Atr, BASIC, Copy, Dir, Ext, Free, Goto, Help, Inv, Kill,
List, Mark, Purge, Quit, Rname, Sys, Transf,Invert, F1 FORMAT,
F2 BACKUP/EDIT, F3 FORMS/RS232,Pg+-,Desc: ?/:*=
```

Folgende Aufrufe sind unter SHELL/CMD moeglich

ENTER /CMD, /JCL, and /BAS files are executed.
DRIVE 0 - 7 The directory of drive 0,1,2,3,4,5,6, or 7 is displayed.
ATTRIBUTES Attributes displayed.
BASIC loaded. SHIFT B Blank screen toggle
COPY Copies file. SHIFT C allows password.
DIR Full DOS directory displayed. SHIFT D Directory to printer.
EXTENSION Changes wildcard extension. $$$=all. SHIFT E Executes DOS cmd.
FREE Free space map of the current drive. SHIFT F Free space ALL drives
GOTO Moves cursor to filename whose 1st letter is >= letter typed.
HELP This help display. SHIFT H TRSDOS help.
INVISIBLE Invisible file toggle. SHIFT I makes marked files (in)visible.
KILL Removes file from directory. SHIFT K allows password.
LIST Lists file. SHIFT L allows password. SHIFT S pauses display.
MARK Marks/unmarks the file. SHIFT M Modified file toggle switch.
PURGE Purges marked files. SHIFT P purges unused descriptions.
QUIT Exit to TRSDOS. Type * ENTER to reactivate SHELL.
RENAME Renames the file. SHIFT R allows password.
SYSTEM System file toggle switch. SHIFT S System status.
TRANSFER Copies marked files to specified drive. SHIFT T Top Of Form.
INVERT Marked files become unmarked and vice versa.
F1 FORMAT/CMD is run. SHIFT F1 Configuration file created.
F2 BACKUP/CMD is run. SHIFT F2 Simple text editor
F3 Set FORMS filter parameters. SHIFT F3 Set RS232 parameters.
? Read, / Write, : Display, * Print, = Sort description(s).
ARROW KEYS Up,down,left and right. + - Next/previous page.

SHELL/CMD version 1.8a was written by Stephen Milliken and is being distributed as FREEWARE.
NOTE: SHELL/CMD should be installed as an ECI
At TRSDOS Ready type: COPY SHELL18A/CMD SYS13/SYS.LSIDOS:0 (C=N)
SHIFT F1 while in SHELL/CMD will make it auto-execute on RESET.
See the SYSGEN command in the TRSDOS manual for more information.


```
SHELL63A/CMD Drive 0 Record X'0000'

0123456789ABCDEF BYTE 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
======================================================================
......* * * SHEL <00> 01 00 00 9E 96 01 2A 20 2A 20 2A 20 53 48 45 4C
L/CMD VER 1.8a-. <10> 4C 2F 43 4D 44 20 56 45 52 20 31 2E 38 61 2D 20
WRITTEN BY STEPH <20> 57 52 49 54 54 45 4E 20 42 59 20 53 54 45 50 48
EN MILLIKEN 10 C <30> 45 4E 20 4D 49 4C 4C 49 4B 45 4E 20 31 30 20 43
OCHATO PARK, RAN <40> 4F 43 48 41 54 4F 20 50 41 52 4B 2C 20 52 41 4E
DOLPH, MA 02368. <50> 44 4F 4C 50 48 2C 20 4D 41 20 30 32 33 36 38 20
- Copyright 1985 <60> 2D 20 43 6F 70 79 72 69 67 68 74 20 31 39 38 35
, All rights res <70> 2C 20 41 6C 6C 20 72 69 67 68 74 73 20 72 65 73
erved Distribute <80> 65 72 76 65 64 20 44 69 73 74 72 69 62 75 74 65
d as FREEWARE, N <90> 64 20 61 73 20 46 52 45 45 57 41 52 45 2C 20 4E
OT TO BE SOLD *. <A0> 4F 54 20 54 4F 20 42 45 20 53 4F 4C 44 20 2A 20
* *>.2....!..>d. <B0> 2A 20 2A 3E 01 32 A4 C8 06 00 21 00 00 3E 64 EF
_.0.!0.>..!..>. <C0> 7C FE D0 30 1B 21 7D CF 3E 0A EF 21 AD C0 3E 0A
.>..>i.>e...w... <D0> EF 3E 01 EF 3E 69 EF 3E 65 EF AF FD 77 04 18 1A
..._>..2M.>e..$. <E0> 06 08 0E 5F 3E 0F EF 32 4D CF 3E 65 EF FD 7E 1B
.c(.!..>..!..>.. <F0> FE 62 28 0C 21 DE C8 3E 0A EF 21 00 00 3E 16 EF
                         !!
                         !!
```

1. dieses Byte wird auf 63H abgeaendert

o d e r

2. die unterstrichenen Bytes werden alle auf 00H gesetzt.

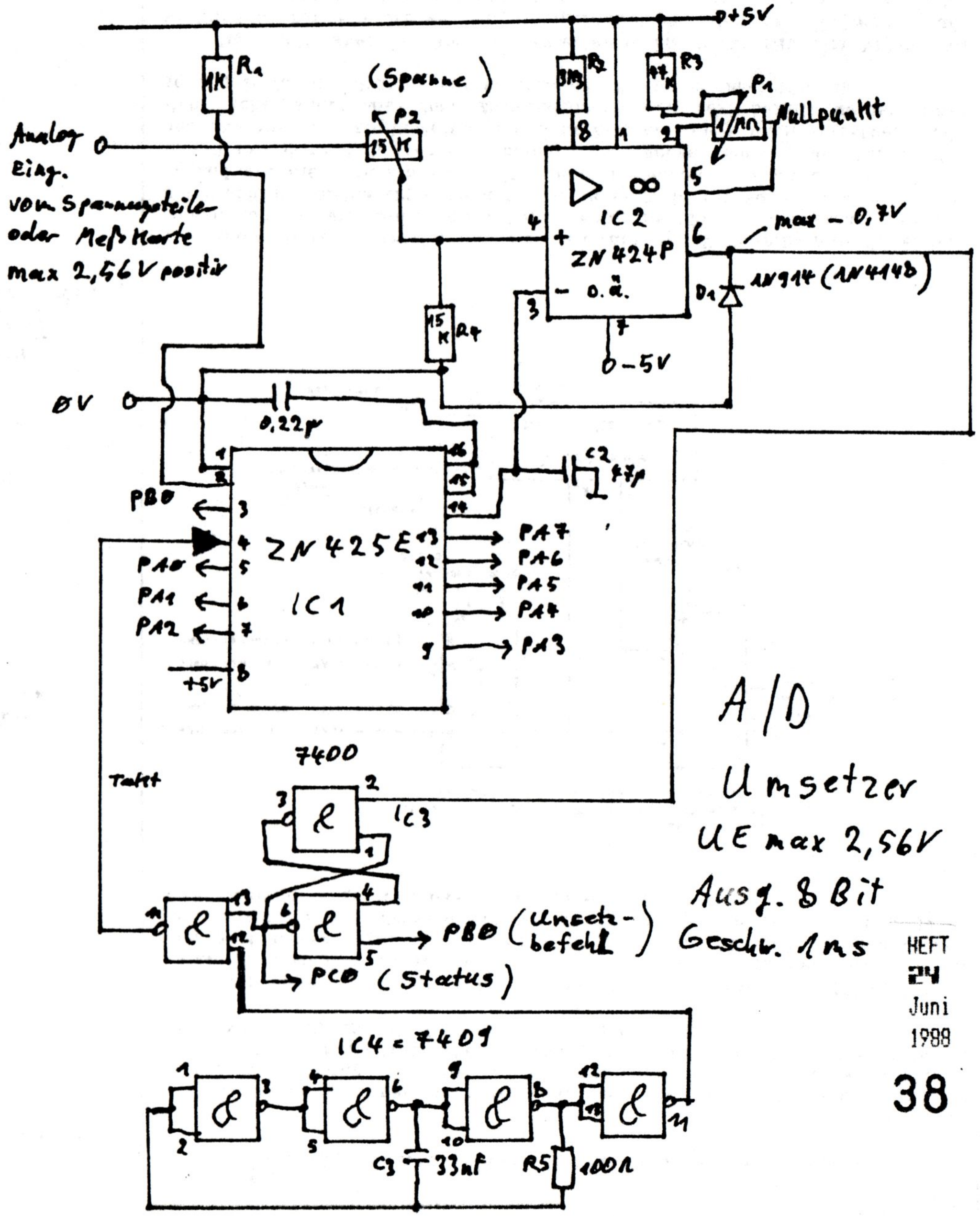
Analog/Digital - Umsetzer

Heute moechte ich Euch meine Schaltung fuer den ECB-Bus vorstellen. ECB-Bus muss hierbei in Klammern, weil es sich um ein 19"Gehaeuse von mir zur Ein-und-Ausgabe und zur Messdatenerfassung, sowie zu Steuerungszwecken handelt. Ich habe dieses Problem mit dem 8255-Interface geloest, wobei es etliche Versionen davon gibt. Meine Schaltung ist von Helmut Bernhardt.
In das 19"Gehaeuse gehe ich mit einen 25 pol. Sub-D-Stecker. Das Gehaeuse hat ein eigenes Netzteil mit +5V, -5V, +12V, -12V und ca. 5A. Fuer die A/D-Karte habe ich auch eine Folie angefertigt. Wer eine Platine haben moechte, kann sie bei mir fuer 5.-DM und Freiumschlag bekommen. Eine Zeichnung fuege ich bei, sowie eine FotoKopie der Bestueckungsseite.
Die Karte ist fuer den Einsatz von 0-20mA, sowie fuer die Messung von max. 2.56Volt vorgesehen. 0-20mA deshalb, weil in der Messtechnik mit diesem Strom gesteuert und geregelt wird. Mit Strom Kann man Grosse Entfernungen von z.b.s. 3Km ueberwinden ohne das es zu verlusten Kommt. An einer Karte fuer die Ausgabe von Spannung, sowie von 0-20mA arbeite ich noch. Es gibt da noch einige Probleme mit den Massen, wegen der Spannungsgesteuerten Konstantstromquelle.

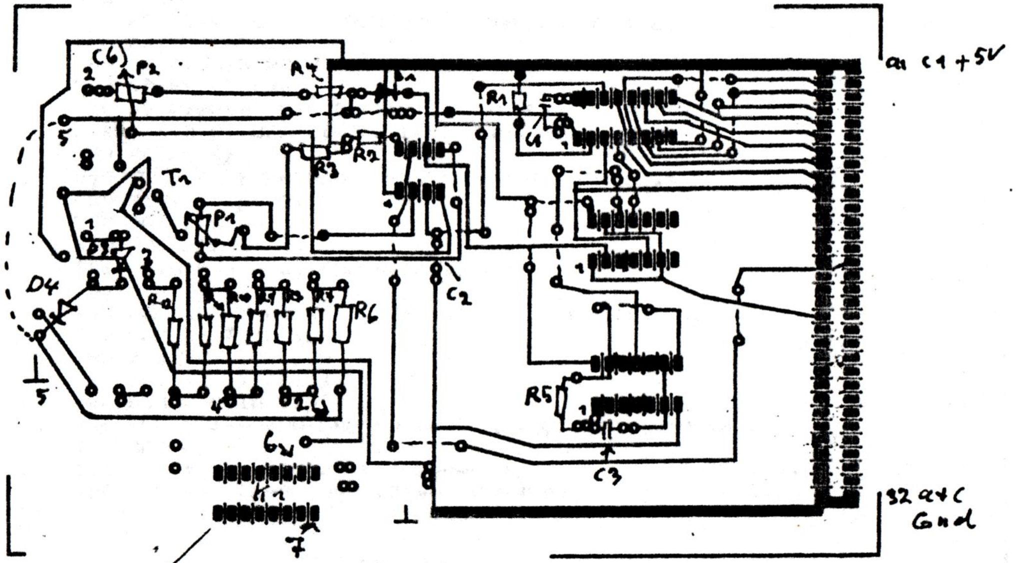
Belegung des 64pol. Stecker:
+5V=1a+c, PA0=2c, PA1=3c, PA2=4c, PA3=5c, PA4=6c, PA5=7c, PA6=8c, PA7=9c, PB0=10c, PB1=11c, PB2=12c, PB3=13c, PB4=14c, PB5=15c, PB6=17c, PB7=18c, PC0=19c, PC1=20c, PC2=21c, PC3=22c, PC4=23c, PC5=24c, PC6=25c, PC7=26c, -5V=15a, GnD=32a+c, -12V=22a, +12V=13a.

Viel Spass beim Basteln
Reiner Stober

```
100 REM **** Programm "Spannungsmessung"          ****
110 REM **** Prog.Name:  "Spamess"                ****
120 CLS
130 REM Z.130-190:Rahmen fuer die Bildschirmanzeige
140 FOR X=0 TO 127
150 SET (X,1):SET (X,47)
160 NEXT X
170 FOR Y= 1 TO 47
180 SET (0,Y):SET(1,Y):SET (126,Y):SET (127,Y)
190 NEXT Y
200 PRINT@ 328,"SPANNUNGSMESSUNG";
210 OUT 7,153
220 OUT 5,0:OUT 5,1
230 STAT = INP(6)
240 IF STAT>0 PRINT@ 584,"SPANNUNG ZU HOCH";:END
250 X=INP(4)
260 U=X*0.01:REM FAKTOR FUER 10mV/TREPPENSTUFE
270 PRINT@648,"DIE SPANNUNG BETRAEGT     :";
280 M$=#####.##"
290 PRINT@712,USING M$;U;
300 PRINT" VOLT";
310 FOR I=1 TO 100:NEXT I:REM WARTESCHLEIFE
320 GOTO 220
```

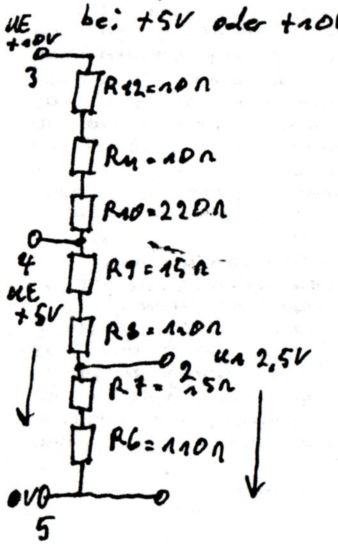
A/D Umsetzer
UE max 2,56V
Ausg. 8 Bit
Geschw. 1ms



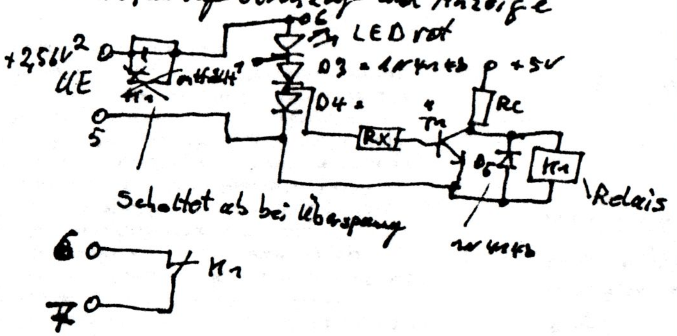
Bestückungsseite A/D Umsetzer

vorgesehen für Relais zum Abschalten der Eingänge bei Überspannung.

Erweiterung Eingang 0–20 mA bei +5V oder +10V

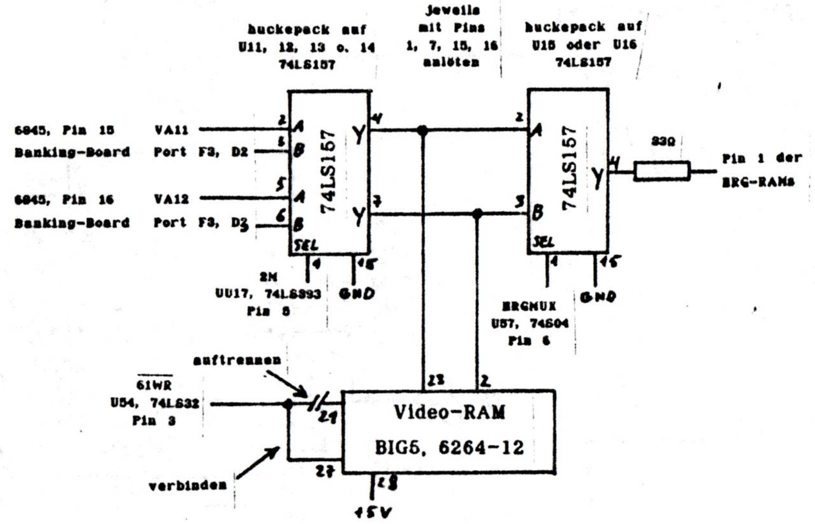


Überspannungsbegrenzung und Anzeige



Im Sonderheft über die Aufrüstung des Genie 3s stellten Helmut und ich unsere Erweiterung vor, mit der alle Speicherbaugruppen dieses Computers eine vierfache Kapazität erhalten. Dazu gehörten auch der Bildschirm mit 8 kB (zuvor 2 kB) und die Graphik mit 256 kB (vorher 64).

Viel später fiel mir auf, daß der Video-Controller (6845 CRTC) 14 Adreßleitungen hat. Damit könnte er sogar 16 kB Bildschirm verwalten. Er war aber noch immer wegen der ursprünglichen 2 kB nur mit 11 Leitungen verkabelt. So wurden die Adressen A11 und A12 des 6845 auch noch an den Bildwiederholspeicher angelötet. Das sieht nun so aus:



Nach dieser Operation ist die Hardware des Videoteils nun folgendermaßen konfiguriert: wie in alten Zeiten kann die CPU nur höchstens 2 kB von 3800–3FFFh gleichzeitig erreichen. Das langt z. B. für 80 X 25 Zeichen. Den Zugriff auf die vollen 8 kB erhält sie in Portionen zu 2 kB über die Bits 2 und 3 des Ports F3h (per OUT-Befehl). So ergeben sich aus der Perspektive des Z80 vier Einzelbildschirme zu 2048 Zeichen. Sie können selbstverständlich nacheinander mit *einem* zusammenhängenden Text befüllt werden. Dasselbe gilt für die Graphik mit achtmal 32 kB oder zweimal 128 kB.

Der CRTC aber ist über 13 Adreßleitungen direkt mit dem Video-RAM verbunden; für ihn ist es *ein* einziger Speicher, den er stückweise oder auch ganz anzeigen kann (wofern der Monitor mitspielt – meiner tut es nicht: s. u.). Daher besteht nun die Möglichkeit, die CPU in einem Teilbereich des Bildschirms von max. 2 kB Größe werkeln zu lassen, während auf Wunsch wesentlich mehr angezeigt wird.

Nun war es theoretisch möglich, den kompletten Videospeicher von 8192 Zeichen anzuzeigen. Da wäre aber in dem Gedränge nicht nur nichts mehr zu erkennen. Die Hardware des Monitors (nicht des Computers) setzt auch Grenzen in der vertikalen und horizontalen Punktdichte und damit in der Anzahl der anzeigbaren Zeichen senkrecht und waagerecht.

Helmut und ich wollten es wissen. Zum Ausprobieren der maximal möglichen Auflösung ging ich von dem Bildschirmformat 92 X 22 Zeichen bei einer Zeichenhöhe von 12 Scan-Zeilen aus. 92 Zeichen ist bei meinem Monitor die höchstmögliche Zeilenlänge, was ich früher schon einmal ausgetestet hatte. Bei 22 Zeilen ergibt sich eine Bildspeichergröße von knapp 2 kB, also das alte Maximum. 12 Zeilen pro Zeichen erwiesen sich als Obergrenze, denn bei 13 Scan-Zeilen begann mein Bildschirm zu schwimmen.

Um die größtmögliche Höhe des Bildschirms rauszukriegen, programmierte ich zunächst den Interlace-Modus. Dabei werden abwechselnd zwei Halbbilder angezeigt, um die Anzahl der möglichen Scan-Zeilen zu verdoppeln. Nach der Schimpansen-Methode (Rumprobieren) fand ich anschließend nach und nach die Maximaldaten für die Register 4-7 des CRTC heraus, um den Bildschirm auch vertikal bis zum Anschlag auszureizen.

Das Resultat waren 92 X 60 Zeichen, also fast fünfeinhalb kBytes Fernsehen. Für ASCII-Texte wird man dieses Format kaum je brauchen. Es ist witzlos, halbe Bibeln gleichzeitig auf dem Bildschirm zu haben. Für Graphik-Anwendungen kann jedoch eine extrem hohe Auflösung sehr interessant sein. Bei einer Zeichenbreite von 8 Punkten und einer Höhe von 12 Dots gibt dieses Format eine Auflösung von 736 X 720 Pixels her. Und das mal 2, weil das G3s zwei Graphikseiten speichern und anzeigen kann. Theoretisch sind sogar 1024 X 1024 Dots denkbar, was ein Monitor mit hinreichender Bandbreite dann auch anzeigen sollte. Daher das überlebensgroße "H" in der Überschrift.

Das im Anschluß gelistete Programm füllt den Bildschirm mit einem Graphikmuster, das erkennen läßt, wo welche 32-kB-Seite des Graphikspeichers beginnt. Zunächst werden die experimentell ermittelten Daten auf den CRTC ausgegeben. Es folgt ein wenig G3s-spezifisches Vorgeplänkel (ASCII-Bildschirm aus- und Graphik einblenden). Danach werden über die Bits 2 und 3 des Ports F3h nacheinander die 4 Bildschirme bzw. Graphik-Memories selektiert und mit dem Punktmuster des Bildschirmzählers beladen. Es entstehen senkrechte Streifen unterschiedlicher Lage und Dicke. Wo sich die Form eines Streifens ändert, beginnt ein neuer der 4 Bildschirme alter Größe.

Diese äußerst üppige Auflösung wird man wohl selten brauchen. Durchaus sinnvoll ist aber die Möglichkeit, in einem gesonderten Abschnitt des Bildschirms ständig ein Menü, eine oder mehrere Statuszeilen oder ein DIR in der Anzeige zu haben, während die CPU darunter oder darüber noch immer über volle 2 kB Videospeicher verfügen kann. Das Menü wird auch nicht durch einen Scroll oder CLS gelöscht, weil nur der CRTC, nicht aber die CPU gleichzeitig den vollen Bildwiederholspeicher von 8 kB beherrscht.

Arnulf Sopp

```
                 ;       Testroutine, mit der die höchstmögliche Graphikauflösung
                 ;       beim Genie 3s mit Helmut Bernhardts Mega-Banker
                 ;       durch Probieren ermittelt werden kann
                 ;
                 ;       Sopp

5200                     ORG     5200h           ;im User-RAM

5200  214152     start   LD      HL,data         ;Parametertabelle für den CRTC
5203  01F710             LD      BC,10f7h        ;Zähler 16 Register, CRTC-Datenport F7h
5206  AF                 XOR     A               ;A (- 00h, CRTC ab Reg. 0 programmieren
5207  D3F6       setcrtc OUT     (0f6h),A        ;internes CRTC-Register adressieren
5209  3C                 INC     A               ;auf nächstes von 16 Registern stellen
520A  EDA3               OUTI                    ;Datum auf CRTC ausgeben, Zeiger weiter
520C  20F9               JR      NZ,setcrtc      ;falls noch nicht 16 Daten ausgegeben
520E  DBFA               IN      A,(0fah)        ;Systemport 1 des G3s
5210  F5                 PUSH    AF              ;Inhalt retten
5211  F602               OR      02h             ;Graphik anzeigen
5213  E67F               AND     7fh             ;ASCII-Bildschirm ausschalten (nur Grph.)
5215  F5                 PUSH    AF              ;diesen Zustand auch retten
5216  F608               OR      08h             ;Graphik für die CPU für RD/WR freigeben
5218  F3                 DI                      ;vorsichtshalber
5219  D3FA               OUT     (0fah),A        ;Systembyte 1 neu schreiben
521B  0604               LD      B,04h           ;Zähler für 4 Bildsch. des Vierfach-G3s
521D  AF                 XOR     A               ;A (- 00h, ab Bildsch. 0 des Vierfach-G3s
521E  210080     fillscr LD      HL,8000h        ;Anfangsadresse des Graphikspeichers
5221  54                 LD      D,H             ;DE (- 8000h, = 32kB Graphik pro Bildsch.
5222  5D                 LD      E,L
5223  D3F3               OUT     (0f3h),A        ;einen von 4 Gr.-Bildschirmen selektieren
5225  4F                 LD      C,A             ;Inhalt von Port F3h retten
5226  70         setpix  LD      (HL),B          ;Pixelmuster 04, 03, 02, 01 anzeigen
5227  23                 INC     HL              ;nächste Stelle des Graphikspeichers
5228  1B                 DEC     DE              ;Bytezähler herunterzählen
5229  7A                 LD      A,D             ;prüfen, ob
522A  B3                 OR      E               ;der Zähler schon abgelaufen ist
522B  20F9               JR      NZ,setpix       ;falls nein: weiter Pixels anzeigen
522D  79                 LD      A,C             ;A (- Inhalt von Port F3h
522E  C604               ADD     A,04h           ;abgelaufen: nächsten Bildschirm selekt.
5230  10EC               DJNZ    fillscr         ;nächst. Bildsch. mit einem Muster füllen
5232  AF                 XOR     A               ;A (- 00h, Bildschirm 0 des Vierfach-G3s
5233  D3F3               OUT     (0f3h),A        ;zurückschalten
5235  F1                 POP     AF              ;Systembyte 1 (noch mit angez. Graphik)
5236  D3FA               OUT     (0fah),A        ;ab 8000h wieder RAM einblenden
5238  CD4900             CALL    0049h           ;auf irgendeinen Tastendruck warten
523B  F1                 POP     AF              ;ursprüngl. Inhalt von Port FAh (o. Gr.)
523C  D3FA               OUT     (0fah),A        ;restaurieren
523E  C3C901             JP      01c9h           ;CLS (Videoformat restaur.), EI und ENDE

                 ; Die Daten für den CRTC können nach Belieben zum Test geändert werden:

5241  70         data    DB      70h     ;Reg. 0: Bildschirmbreite 112 Zeichen
5242  5C                 DB      5ch     ;      1: angezeigt 92 Zeichen
5243  61                 DB      61h     ;      2: horizontale Sync-Position
5244  08                 DB      08h     ;      3: Breite des Sync-Impulses
5245  20                 DB      20h     ;      4: Bildhöhe 64 Z. (Hälfte wegen Interlace)
5246  10                 DB      10h     ;      5: Bildhöhenjustage in Zeilen
5247  1E                 DB      1eh     ;      6: angezeigt 60 Zeilen (Hälfte wg. Interl.)
5248  1F                 DB      1fh     ;      7: vertikale Sync-Position
5249  03                 DB      03h     ;      8: Interlace-Modus
524A  0B                 DB      0bh     ;      9: Zeilen pro Zeichen -1 (=12 lpc)
524B  29                 DB      29h     ;     10: Cursor: oberer Rand und Anzeigemodus
524C  09                 DB      09h     ;     11: dto. unterer Rand
524D  00                 DB      00h     ;     12: Bildschirmanfang (relativ zu 3800h), MSB
524E  00                 DB      00h     ;     13: dto. LSB
524F  00                 DB      00h     ;     14: Cursoradresse (relativ zu 3800h), MSB
5250  00                 DB      00h     ;     15: dto. LSB

5200                     END     start           ;dort Einsprung

00000 Fehler

data    5241    fillscr 521E    setcrtc 5207    setpix  5226    start   5200
```

BÖRSE -- BÖRSE

* ! Ringeltäubchen ! *

Verkaufe komplette TRS-80-Anlage!
Für nur DM 499.99 (VB) kann ein Clubkamerad oder Einsteiger erwerben:
TRS-80/1 mit 64K (48K RAM)/ Zehnerblock/ Kleinschrift/ Umlauttreiber (ä/ö/ü/ß/Ä/Ö/Ü)/ HRG 1b-Karte (RB-Elektronik) Resettaste oben / Verstärker (Selbstbau).
Dazu: Grüner Monitor (Original Radio Shack)
2 TEAC-Doppelkopf-Laufwerke, 40 Tracks, Slimline-Ausführung, mit DD-Controller; starkes Netzteil.
Selbstverständliches alles einwandfrei funktionsfähig.
Dazu die benötigte sowie erwünschte Software.

Anm.1: All das wird nur geschlossen abgegeben.
Anm.2: über die 99 Pfennig kann verhandelt werden!
Anm.3: Versand ausgeschlossen. Abholung Bedingung. Bei beruflicher Nutzung können Preis, Fahrtkosten und Spesen steuerlich abgesetzt werden! Gültige Quittung wird erteilt.
Anm.4: (Weinheimer!-)"Ehrenwort", daß alles mein legitimes Eigentum, das ich einmal für den ca. 10-fachen Preis erwarb und von dem ich mich nur sehr schweren Herzens trenne!

Anfragen an: K.-J. Mühlenbein, Am Mönchgarten 28, 6940 Weinheim. - Tel.: 06201-55052.

Für die Börse im Club-Info:

1. Verkaufe 1 Diskettenlaufwerk Siemens FDD 100-5
   Shugart-Bus, einseitig, 40 Spuren
   Abmessungen: 85 B x 150 H x 200 T (in mm)
   Preis: 100,- DM
   Hans-Günther Hartmann
   Adresse und Telefonnummer in der Mitgliederliste
2. Suche preiswertes, aber intaktes 19"-Gehäuse
   Hans-Günther Hartmann
   Adresse und Telefonnummer in der Mitgliederliste

M 4p
128 K, 1LW 40/SS, 1LW 80/DS
Daisywheel 2 mit deutschem Handbuch
diverse Software mit Handbüchern
VHB 1800 DM
Tel.: 04101/75422

BÖRSE -- BÖRSE -- BÖRSE

Zu Verkaufen: -- BÖRSE

Unser ehemaliges Clubmitglied Harald Trapp hat noch ein paar Computer-Baugruppen im Keller liegen, die es sicher nicht verdient haben, nie mehr von Strom durchflossen zu werden und ihren Besitzer zu erfreuen (oder zum fluchen zu bringen). Aus diesem Grund möchte er folgende Dinge verkaufen:

- 1 Stück 40 Spur-Floppy — DM 40,--
- 1 Stück RB-High Resolution Graphics Adapter mit Original Treiber (erweiterte Version) Neupreis ca. 350,-- DM — DM 70,--
- 1 Stück RB 2-fach V.24-Schnittstelle mit Original-Software für Mailbox usw. Neupreis ca. 250,-- DM — DM 60,--
- 1 Stück ZERO-Electronic EPROM-Brenner mit Texttool-Sockel — DM 50,--
- 1 Stück Genie I-Basisplatinen mit Tastatur und Speichererweiterung, Floppy-Controller (DS/DD) und Drucker-schnittstelle — DM 50,--
- 1 Stück Original Model 1/III COBOL (Diskette incl. Handbücher) — DM 50,--

Leider hat mir Harald nicht mitgeteilt, ob es sich um Fest- oder Verhandlungspreise handelt. Auf alle Fälle sind die Sachen so günstig (das ist meine Meinung!), daß er um einen Porto- und Verpackungsbeitrag von 10 DM bittet.

Harald Trapp
Kranichstraße 46
4270 Dorsten
Tel.: 02362/42497 oder 23127

Habe ich jemandem die Handbücher für das Programm OPAL 68000 Cross-Assembler zur Einsicht geliehen? Ich kann sie einfach nicht mehr finden! Wenn so, dann bitte ich freundlichst um

BALDIGE RÜCKGABE!

RICHARD RENSCH, D 7128 LAUFFEN/N.
Bahnhofstraße 100 • Postfach 226 • W. Germany • ☎ (07133)-8415

Disketten-Kundendienst

Unternehmen, die aus der Foto-/Chemie-Branche stammen — Beispiele hierfür sind 3M, Fuji und Kodak/Verbatim —, sind für die Herstellung von Disketten eigentlich geradezu prädestiniert. Den Schritt in den Disketten-Markt vollzog jetzt jüngst der Foto-Riese

Werkbild: Polaroid

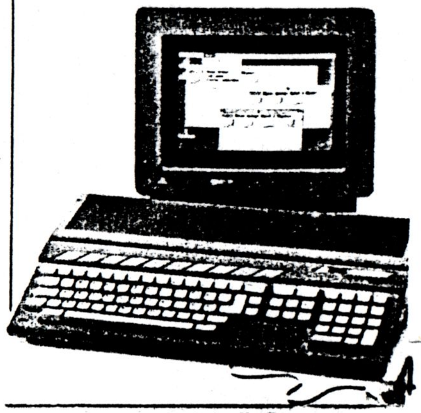
Polaroid. Um sich jedoch gegenüber der Vielzahl von Mitbewerbern abzuheben — 100prozentige Fehlerfreiheit wird sowieso garantiert —, hat man sich bei Polaroid einen speziellen Kundendienst einfallen lassen: Ein kostenloser »DataRecovery-Service« versucht, Daten, die durch Staub, Schmutz, oder durch sonstige »Bürounfälle« unlesbar geworden sind, zu retten. Man hat dafür eigens speziell ausgestattete Labore eingerichtet. Dieser Service gilt für alle Disketten im 3½-, 5¼- und 8-Zoll-Format von Polaroid. (eh)

Trends:

Der Heimcomputermarkt wird stabiler. Hatte man noch zur Jahreswende das Gefühl, sich kaum zwischen den unterschiedlichen Herstellern und Modellen entscheiden zu können, so hat sich zur Jahreshalbzeit die Situation aufgeklärt. Beherrscht wird der Markt nun von etwa drei Firmen, deren Angebote für jeden Geschmack das eine oder andere interessante Gerät vorweisen. Bei Commodore ist es der Dauerbrenner C64, der eine Art »Volkswageneffekt« für sich verbuchen kann. Obwohl eine veraltete Maschine, ist er immer noch von Interesse, da er auf Grund seines Umfelds an Peripheriegeräten und Software fast jeden Wunsch eines Fans erfüllen kann. Für den Anspruchsvolleren bietet Commodore mit seinen 128er Modellen eine interessante Alternative. Bei ATARI ist es vor allem das Einsteigermodell 130 XE, aus dem 800 XL hervorgegangen, das von Preis und Leistung her interessant ist. Dazu kommt die ST-Reihe mit vier Modellen, die mit ihrer 32-Bit-CPU und einem RAM von 512 bis einem Mega-Byte erfolgreich die Lücke zwischen Heim- und Personal Computer füllen. Fast 250 Programme stehen inzwischen zur Auswahl. Fest etabliert sind auch die Schneider-Computer. Sie sind vor allem durch das preiswerte Komplettangebot attraktiv, das dem Einsteiger keine zusätzlichen Kosten für Laufwerk und Monitor aufbürdet.

Fazit: die Exoten sind mehr oder weniger vom Markt verschwunden, entscheidende Preiseinbrüche sind in den nächsten Monaten nicht zu erwarten und auch keine maßgeblichen Neuerungen. Alles in allem eine ruhige, solide Situation, die eine Kaufentscheidung wesentlich vereinfacht, da man jetzt nicht mehr das Gefühl haben muß: Mein Computer, heute gekauft, ist morgen schon zum halben Preis zu haben und dazu hoffnungslos veraltet. ✶

Integriertes Laufwerk (rechts) und »Maus«: Heimcomputer werden immer komfortabler.



Brehms-Tierleben erweitert von

F. W.

Pech hat, wer seinen Computer im Versandhandel kaufte. Ihm ist eine Begegnung mit dem Computerhandel-Fachverkäufer entgangen. Ein Kontakt mit dieser Spezies verläuft erheiternd oder auch schockierend, ganz nach persönlichem Charakter, immer jedoch ist er interessant.

In Zeiten aussterbender Tierarten ist es wichtig unsere Kenntnisse von der Natur zu vervollständigen. Dieser Bericht ist all jenen zugedacht, denen es noch nicht gelungen ist, das scheue Wild aufzuspüren.

Der Computerhandel-Fachverkäufer (lat.: Lupus Basicus), im Volksmund auch Bitkäfer genannt, erweist sich als Stadtbewohner. Männliche Exemplare sind dabei häufiger anzutreffen als ihre Weibchen. Der engere Lebensraum begrenzt sich auf Reviere in Kaufhäusern und Einzelhandelsgeschäften. Die Einzelhandels-Bitkäfer machen häufig eine Mutation zum Inhaber durch, eine Charakterisierung, die ihn von seinen Artgenossen abhebt. Diese Art des Lupus Basicus soll hier zuerst beschrieben werden.

Ihr Verhalten ist nicht mehr typisch für die Gattung, ihr Jagdtrieb ist bis auf ein paar rudimentäre Reste verkümmert, dagegen ist ihr Wachinstinkt besonders ausgeprägt. Kein Wunder, daß sie, domestiziert wie sie nun einmal sind, gerne als Haustiere großer Firmen wie IBM und APPLE gehalten werden. Ein wenig Fürsorge ihrer Herrchen danken sie mit unerschütterlicher Produkttreue.

Die Beobachtung am lebenden Exemplar ist einfach. Betreten Sie den Laden und Sie werden das possierliche Geschöpf scheu und einsam vor einem Computer sitzen sehen. Es täuscht emsige Geschäftigkeit vor, um nicht mit dem Eindringling Kontakt aufnehmen zu müssen, weshalb wir es auch vorerst nicht ansprechen sollten.

Besser ist es, erst einmal das Revier abzugehen und die Schaustücke zu besichtigen. Dabei wird man von den heimlichen neugierigen Blicken des Inhaber-Bitkäfers verfolgt, der sicherste Beweis für seine schon sprichwörtliche Wachsamkeit. Strapazieren Sie bitte diese Wachsamkeit nicht über Gebühr, indem Sie irgendetwas anfassen. Das eben noch stille Wesen wird sich sofort in eine Bestie verwandeln, um das Eigentum seines Herrchens (IBM oder APPLE) zu schützen. Außerdem sollten wir als Liebhaber doch die Natur schützen und studieren, statt sie zu stören.

Irgendwann wird der Lupus Basicus Sie beschnuppern und mit einem Laut, der sich anhört wie "Wasgibtsdenn" den ersten Kontakt aufnehmen. Dieses Imponiergehabe des erwachsenen Männchens darf trotz des Tonfalls nicht als Drohung verstanden werden. Verhalten Sie sich also zurückhaltend, aber machen Sie jetzt nicht den Fehler und fragen Sie nach einem Homecomputer. Daraufhin würde der Bitkäfer sich nur majestätisch umdrehen und schweigend zu seinem Platz zurückkehren. Die soziale Hackordnung ist nun mal sehr ausgeprägt.

Möchten Sie Ihre Beobachtungen fortsetzen, dann stellen Sie sich als absoluten Laien vor. Ist sein Lieblingsgerät gerade greifbar, wird Ihnen Ihr Studienobjekt in den nächsten drei bis vier Stunden die Höhepunkte seiner Dressur vorführen. Er zeichnet Grafiken, tippt Texte und kennt allerlei lustige und kurzweilige Spielchen, in deren Verlauf er Ihnen auch mal seinen Platz an der Tastatur überläßt.

Ermüdet der Bitkäfer, läßt er sich mit "Istjatoll" oder "Habichnochniegesehen" zu neuen Leistungen anspornen.

Wie schon erwähnt, die im Einzelhandel auftretenden Individuen der Gattung Lupus Basicus sind nicht arttypisch. Interessanter, weil in der ursprünglichen ungezähmten Verkäuferform, präsentiert sich die Kaufhausvariante. Mit seinem nächsten Verwandten, der Familie der Versicherungsvertreter, hat er das herrische Revierverhalten gemein, ja es gehört zum Lebenskampf der Spezies. Die ungehemmte Jagd auf Kunden, die den Beobachter häufig abstößt, ist instinktiv verankert und entsteht nicht etwa aus purer Lust am Quälen.

Sie merken bereits, die Beobachtung des Bitkäfers kann auch gefährlich sein. Im Kaufhaus erfordert sie deshalb die strikte Einhaltung zweier Regeln:

Betreten Sie das Revier nie montags. Diesen Tag benötigt der Bitkäfer zur Regeneration und reagiert deshalb bei Störenfrieden sehr bissig. Die zweite Regel verletzen viele Naturfreunde, weil sie bisher nur den Inhaber-Bitkäfer kennengelernt haben. Das im Kaufhaus lebende Exemplar täuscht Tätigkeiten am Computer nicht vor. Es ist damit beschäftigt, Raubkopien für den Privat-Gebrauch anzufertigen und will dabei nicht gestört werden.

Aber auch wenn wir Montage und die Tage nach einer frischen Softwarelieferung vermeiden, der "Tiger des Kaufhausdschungels" bleibt ein gefährliches Abenteuer für Mutige.

Kaum haben Sie sein Revier betreten, gellt Ihnen aus irgendeinem Zubehörregal der Kampfschrei "Kannichihnenhelfen" entgegen. Ist Ihr Fluchtweg durch ein zweites Exemplar versperrt (der Lupus Basicus jagt auch in Rudeln) dann stellen Sie sich dem Kampf, geben Sie sich unentschlossen aber kaufbereit, als beste Waffe hat sich Geld erwiesen.

Zwar weiß der Bitkäfer schon lange, was er Ihnen andrehen will (den Ladenhüter oder seine Privatmarke), aber er wird mit Ihnen spielen wie die Katze mit der Maus. Auf seine Fragen nach Ihren Wünschen brauchen Sie nicht zu antworten. Diese sollen Ihnen ohnehin nur klar machen, wie wenig Sie eigentlich wissen und zum anderen sind die Ohren des Bitkäfers sowieso beim Kundenkampf verschlossen. Angeblich hören weibliche Exemplare noch, reagieren aber nur auf Komplimente über ihr Fachwissen.

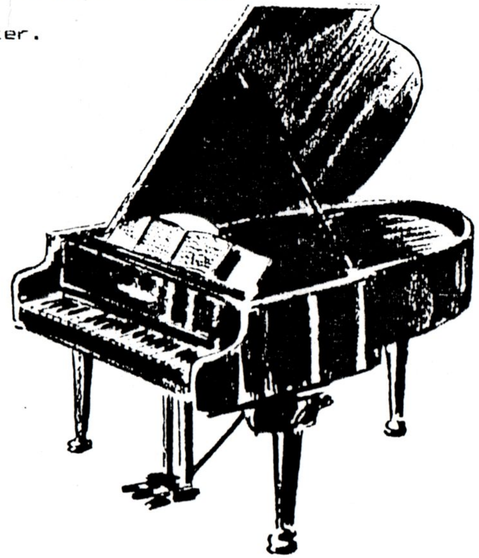
Dem Jagdritual entspricht es, daß der Lupus Basicus irgendwann einmal die Warnung "Begrenztessonderangebot" ausstößt. Auch für hartgesottene Abenteurer ist jetzt der Zeitpunkt gekommen, sich zurückzuziehen. Sagen Sie nicht, Sie müssen sich den Kauf noch einmal überlegen, denn wen der Bitkäfer erwischt, der überlegt nicht mehr. Geben Sie stattdessen scheinbar ganz offen zu, das sei alles doch viel komplizierter gewesen als Sie es sich vorgestellt hätten. De Lupus Basicus wertet dies als Demutsgeste. Ihre dargebotene Kehle (bildlich) läßt ihn zögern. Sie entziehen seinem Angriff die Wut und können entkommen.

Sollten Sie bei Ihren Streifzügen durch die Computerwildnis einmal auf einen Bitkafer stoßen, dessen Tätigkeit in unverbindlicher, freundlicher und sachkundiger Beratung verläuft, dann haben Sie einen ausgesprochenen Einzelgänger erwischt. Hegen und pflegen Sie diese Laune der Natur, es ist der Freund des Users.

Club-80-Werbeschrift

Ein vollendetes Instrument..

besitzen Sie mit Ihrem TRS-80 Computer.



Er ist eines der weltweit am meisten gebauten Geräte.

Die Hardware ist erprobt und leistungsfähig

Es gibt eine Fülle guter Programme für ihn.

Das Betriebssystem CP/M öffnet einen riesigen Schatz guter Software für den Tandy-Anwender.

Der im TRS-80 eingestzte Mikroprozessor Z-80 ist Weltmeister, wenn es um die Zahl der verkauften Chips geht. Jährlich wird er etwa 50 Millionen mal in elektronische Geräte eingebaut.

Wie kann man den vorhandenen Schatz an Hardware- und Softwareerfahrung am besten nutzen? Niemand kann das riesige Angebot überschauen. Aus diesem Grund haben sich Anwender des TRS-80 und kompatibler Geräte zu einer Gemeinschaft zusammengeschlossen, die sich gegenseitige Hilfe und Erfahrungsaustausch zum Ziel gesetzt hat.

HARDWARE * SOFTWARE
TEXT DATEI
COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80
TEXT DATEI
ERFAHRUNGSAUSTAUSCH

Der Club 80 bietet seinen Mitgliedern alle zwei Monate eine Zeitschrift mit Tips, Erfahrungsberichten, Literaturhinweisen und Kontaktadressen. Eine Büchersammlung steht allen Clubmitgliedern kostenlos zur Verfügung. Fordern Sie ganz unverbindlich ein Probeheft und weitere Informationen an bei:

Gerald Schröder
Am Schützenplatz 14
2105 Seevetal 1

Tel. (04105)2602

BASIC-QUIZ:

So fragt der Computer

Während eines Programms fragt der Computer nicht nur Werte oder Namen ab, die man dann eingeben und mit der Returntaste bestätigen muß, häufig kommen auch reine Ja-Nein-Entscheidungen vor. Für diese Eingaben reicht dann das Eintippen von J oder N, ohne daß dies noch mit »Return« bestätigt werden muß. Dies ermöglicht der GET-Befehl, der bei einigen BASIC-Versionen INKEY lautet. Bei diesem Befehl hält der Rechner nicht an, sondern »schaut nur nach«, ob ein Zeichen in einem speziellen Speicher, dem Tastaturpuffer, steht. Denn er fragt ständig mit einer geräteabhängigen Frequenz die Tastatur ab, ob vielleicht eine Taste gedrückt wurde. Man muß dem Computer also eigens sagen, daß er auf eine spezielle Eingabe warten soll. Eine der häufigsten Abfragen dieser Art steht oft am Ende eines Programms: die Frage, ob ein neuer Programmlauf gewünscht wird oder nicht.

```
100 CLS: REM Programmbeginn
860 PRINT "Neuer Programm-
    lauf (J/N)?"
870 GET TASTE $
880 IF TASTE $="" THEN GOTO
    870
890 IF TASTE $="J" THEN
    GOTO 100
900 ENDE
```

Warum ist Zeile 880 unbedingt notwendig?

Darf man Programme tauschen?

Im Kleinanzeigenteil von mancherlei Zeitschriften findet man oft Angebote wie „Tausche CP/M-Programme", „Suche Pascal, habe Spiele" und anderes mehr. Die Frage ist nur, ob ein solcher Tausch rechtlich überhaupt zulässig ist. Tausch bedeutet normalerweise, daß ein Tauschpartner dem anderen etwas gibt, was er selbst anschließend nicht mehr besitzt. Typisch dafür ist der Hardware-Tausch: „Tausche Sinclair-Spectrum gegen Commodore-64" ist so ein Beispiel. Anders der Software-Tausch: Wenn zwei Computerfans einen Pascal-Compiler und einen Basic-Interpreter tauschen, so besitzt (dank Duplizierung der Datenträger) hinterher jeder der beiden sowohl den Pascal-Compiler als auch den Basic-Interpreter. Das Resultat ist demzufolge in der Regel, daß illegalerweise eine Programmkopie für Dritte angefertigt wurde.

Ein Tausch von Programmen ist also normalerweise nur dann zulässig, wenn dabei keine Duplizierung stattfindet. Aber auch diese Art von Tausch kann durch eng formulierte Software-Kaufverträge unter Umständen verboten sein.

Fe.

Clubbibliothek

An alle die sich beruflich mit Siemens BS2000, MS-DOS und SINIX bzw. UNIX herumschlagen:

Zum schnellen Nachschlagen von Systemkommandos, Kommandos für Dienstprogramme und Systemverwalterkommandos habe ich umfangreiche TSCRIPS-Dateien erstellt die ich interessierten Leidensgenossen gerne zur Verfügung stelle.

BS2000	ca. 80 Seiten
SINIX	ca. 10 Seiten (wird noch erweitert)
MS-DOS	ca. 10 Seiten (wird noch erweitert)

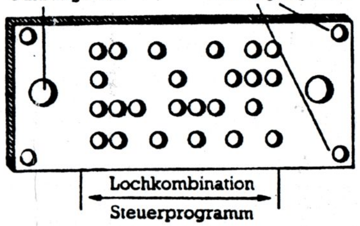
Die Beschreibungen habe ich in die CLUB80-Bibliothek aufgenommen. Falls es in unserem Club Siemens Mitstreiter gibt, so wäre ein Erfahrungsaustauch doch sicherlich von Nutzen.

Werner Förster

Ein Befehl aus lauter Löchern

Die Frage »Was war zuerst, das Huhn oder das Ei?« läßt sich zumindest bei der Entwicklung des Computers leicht beantworten. Lange bevor es den ersten Computer gab, existierten bereits Programme, die, auf Datenträgern abgespeichert, immer wieder verwendet werden konnten. Die Rede ist von Lochkarten.

Lochkarten-gesteuerter Webstuhl: der erste Schritt zur »denkenden« Maschine.



Im Jahr 1728 baute der französische Mechaniker Falcon als erster einen Webstuhl, der von einem mit Löchern versehenen Brettchen automatisch gesteuert wurde. Solche Brettchen bildeten die Vorform der Lochkarten, wie sie dann in der Datenverarbeitung verwendet wurden. Entscheidend verbessert wurde diese Erfindung von dem Franzosen Joseph-Marie Jacquard (1752–1834). Er stellte einen Webstuhl vor, der von richtigen Lochkartenprogrammen gesteuert wurde. Mehrere solcher Lochschablonen, inzwischen aus Pappe, bildeten ein Programm. Sie wurden nacheinander in die Abtastvorrichtung des Webstuhls eingeführt. So konnten automatisch die kompliziertesten Muster auch von ungelernten Hilfskräften gewebt werden. Bereits 1812 waren über 10000 solcher programmgesteuerten Webmaschinen in Betrieb.

Mit der Lochkarte als Programmträger für die Steuerung mechanischer Arbeitsabläufe war eine der wesentlichen Erfindungen für die Entwicklung von automatischen Rechenmaschinen und Computern gemacht. Noch heute werden Lochkarten in der Datenverarbeitung verwendet, z. B. bei der Auswertung großer Mengen von Zahlen. ✶

CLUB 80 — Bücherei

Hallo Bücher-Freunde!

Diesmal kann ich wieder über viele Neuzugänge in der Clubbücherei berichten. Wie schon im letzten Info erwähnt hat der CLUB 80 eine Sonderaktion des Heise-Verlags wahrgenommen und die Bücherei kräftig ergänzt. Vierzehn neue Bücher und die seit langem vermißte BASIC-Referenzliste findet ihr folgend aufgelistet und ich hoffe, daß das Interesse für die Bücher im gleichen Maß wächst wie die Bücherei selbst!

Hartmut Obermann

0037: Die grosse BASIC Referenztabelle der 51 Dialekte

Wolf-Detlev Luther --- Luther-Verlag, Sprendlingen

Nr. 0052: Pascal und Pascal Systeme

Herschel/Pieper --- Oldenbourg

Eine systematische Darstellung der Sprache Pascal, besonders geeignet zum Lernen im Selbststudium.

Nr. 0053: Microcomputer Grundkurs - Turbo Pascal Wegweiser

Kaier/Rudolfs --- Vieweg

Eine sehr gute Einführung in das Turbo Pascal-System. Das Buch ist zum Selbststudium für Anfänger und Umsteiger bestens geeignet!

Nr. 0054: Microcomputer Aufbaukurs - Turbo Pascal Wegweiser

Kaier/Rudolfs --- Vieweg

Fortsetzung der Einführung in das Turbo Pascal-System.

Nr. 0055: Programmierung in Modula-2

Dal Cin/Lutz/Risse --- Teubner

Eine Einführung in Modula, die vor allem für Pascal-Umsteiger sehr gut geeignet ist!

Nr. 0056: Das Z80-Buch

Eberhard Zehendner --- Markt und Technik

Eine gut geschriebene und leicht verständliche, zum Selbststudium geeignete Einführung in die Programmierung des Z80!

Nr. 0057: CP/M - Anatomie eines Betriebssystems

Jack D. Dennon --- Markt und Technik

Das Buch beschreibt den Einstieg in die Programmierung (vor allem Assembler) unter CP/M.

Nr. 0058 : Listingverwaltung in MBASIC

H.-J. Rothe --- Luther-Verlag

Individuelle Verwaltung zur Auffindung von Listings in Computerzeitschriften

---

Nr. 0059 : dBase II im überblick

Thomas Lauer --- Luther-Verlag

Band 1, die interaktiven Befehle

Nr. 60 : Programmieren von Microcomputern Band 10

D. Herrmann --- Vieweg-Verlag

Datenstrukturen in BASIC und PASCAL

Nr. 0061 : PASCAL - Datenstrukturen und praktische Anwendung

Hardy Feller --- Luther-Verlag

Nr. 0062 : BASIC Grundlagen und Beispiele

Günter Abeldt --- Frech-Verlag

Einführung in die Sprache BASIC

Nr. 0063 : MBASIC unter CP/M

Wilfried Plieth --- Luther-Verlag

Profi-Tips und Techniken für die Programmierung in MBASIC unter CP/M

Nr. 0064 : RPNL

Gustav Wostrack --- Luther-Verlag

Eine Forth-ähnliche Sprache mit strukturunterstützenden Sprachkonstrukten (siehe auch die Artikel von Kurt Müller zu diesem Thema!!!)

HEFT 24 Juni 1988

Nr. 0065 : Model 4/4p Technical Reference Manual

--- Radio Shack

Fast alles über Hard- und Software (TRSDOS 6.x) der Tandy-Modelle 4 und 4p

---

Nr. 0066: Personal Computer Lexicon

Günter Rolle --- Markt und Technik

Lexicon der Computerbegriffe

Nr. 0067: Mein Computer: 2 hoch 10 Fragen und Antworten

Richard Didday --- Vogel-Verlag

1 kByte Dialoge zur Computer- und Microcomputertechnik

Nr. 0068: Mikrocomputertechnik

H.P. Blomeyer-Bartenstein --- Hofacker

Z80, Z8 und Z8000 Hardwarearchitektur und Assemblerprogrammierung

---

Nr. 0069: Das DOS-Buch

Hartmut Grosser --- Röckrath

Die DOS-Bibel für New-und GDOS sowie Color Computer

Bankverbindung des CLUB 80
Postgirokonto Peter STEVENS
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

- I M P R E S S U M -

1. Vorsitzende
Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

2. Vorsitzende
Bernd RETZLAFF
Kleiner Sand 98
2082 Uetersen
☎ 04122 /43551

Hardwarekoordinator
Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar
Club-Bücherei
Werner FÖRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion
Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren

Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck
Horst-Dieter Schroers
Breslauer Straße 9
8016 Feldkirchen
☎ 089 / 9032615

SCHLUSS

Hallo Club-80er,

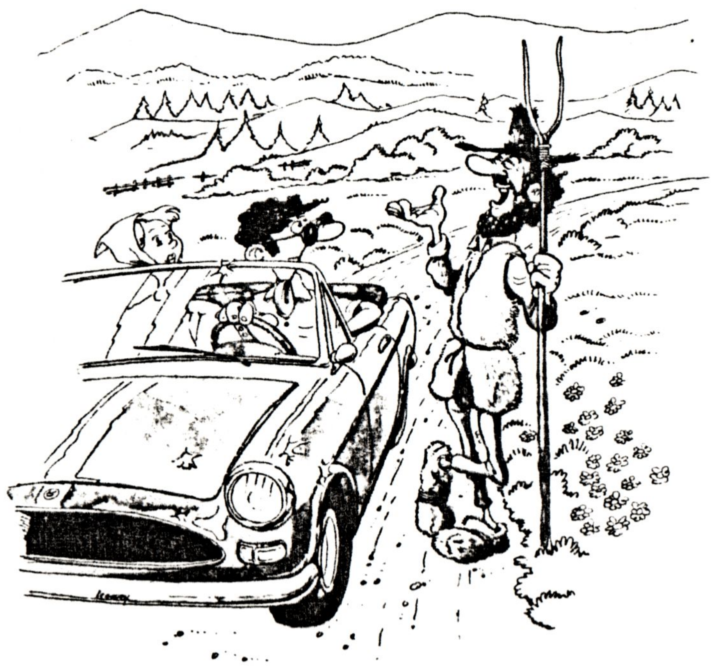
wieder ist es mir mit Erfolg gelungen die Herausgabe des INFO's zu verzögern. Nachdem das letzte INFO zwar pünktlich fertig war, Peter Spieß aber, trotz vorheriger Zusage den letzten Druck nicht mehr bearbeiten konnte - wars mit dem Terminhalten wieder nichts. Ihr habt die Geschichte sicher mitbekommen (zumindest die Auswirkungen).

Ich habe leider noch immer unter chronischem Zeitmangel zu leiden. Deshalb ist auch dieses INFO wieder einmal spät dran. Ich gelobe aber Besserung (wie immer)!
Um dies in die Tat umzusetzen möchte ich, gleich nach Erscheinen dieses INFO's mit der Erstellung des Neuen (Nr. 25) beginnen. Dann wäre ich auch wieder auf ein "Soll/Ist-Konto" mit den Erscheinungsterminen.
Um nun nicht ein "Sauregurkenzeit-INFO" mit wenigen Seiten zu produzieren - die "Artikelsammelzeit" fällt ja dann weg- möchte ich diesmal verstärkt um Eure Mithilfe bitten.
Sicher habt Ihr noch einen Artikel für unser Clubinfo. Sei es nun was Eigenes oder ein Zeitschriftenartikel --- Ihr wisst ja, was gebraucht wird.
Vielleich bekomme ich dazu noch eine "Sonderpuplikation" zusammen, dann wäre es wieder mal eine runde Sache. Ich hoffe nicht, daß ich damit auch noch in das berühmt-berüchtigte "Sommerloch" falle.

Nun zum Abschluß noch zu dem aktuellen INFO. Im Anhang findet Ihr aus aktuellem Anlaß wieder einmal die Anleitung für die Gestaltung der INFO-Beiträge. Wir hatten es am Clubtreffen so beschlossen.

Bis zum nächsten, vielleicht pünktlichen INFO
grüßt Euch Euer

Jens



Ich suchte Glück im Erfolg, im Geld, in der Liebe, der Religion – dann fand ich es in der Natur, im einfachen Leben. Ich baue mein eigenes Gemüse an und nähe meine Kleider selbst – den Rest erledigt mein Computer.

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat // geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117	Wistedt	04182 /8799 // -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060	Bergisch-Gladbach 1	02204 /62781 // -
Bernhardt	Helmut	Hafenstraße 7	2305	Heikendorf	0431 /241907 // 0431 /74047
Betz	Heinrich	St. Wolfgangsstraße 13	8551	Hausen	09191 /31698 // 09191 /611108
Brandl	Hermann	Cäciliastraße 30	8439	Postbauer-Heng	09180/ 493 // 0911 /219-245
Braun	Harald	Postfach 8011	2300	Kiel 17	- // -
Böckling	Ulrich	Am Sonnenhang 11	5414	Vallendar	0261 /69522 // 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200	Wiesbaden-Bierstadt	06121 /509218 // -
Eilers	Hans-Joachim	Thomas-Dehler-Straße 6	2900	Oldenburg	0441 / 53239 // 0441 / 7983845
Förster	Werner	Christoph-Krebs-Straße 9	8720	Schweinfurt	09721 /21841 // 09721 /51256
Hartmann	Hans-Günter	Möwenstraße 9	2876	Berne 2	04406 / 1911 // 0421 / 2483405
Heidenreich	Ulrich	Werderstraße 35	4300	Essen 1	0201 / 282770 // -
Held	Manfred	Stirnerstraße 22	8835	Pleinfeld	09144 /6563 // 0911 /2195245
Hentz	Werner	Am Tränkgarten 20	6457	Maintal 2	06109 /66625 // 0691 /583639
Hermann	Klaus	Gartenstr. 22	7401	Pliezhausen	07127 /70024 //
Homann	Matthias	Maschener-Straße 49 A	2105	Seevetal 1	04105 / 50222 // 040 / 7511-278
JablotschKin	Rainer	Südertor 10	4780	Lippstadt 8	02921 / 23245 // 02921 /70431
Kostya	Mary Jo	Balberstraße 68	CH 8038	Zürich	00411 / 482894 // 00411 / 258170
Krispin	Michael	Schwanstraße 8	4130	Moers 3	02841 /73690 // -
Kuhn	Eckehard	Im Dorf 14	7443	Frickenhausen 1	07022 /45417 // 09171 /832665
Lachmann	Wolfgang	Am Ringofen 11	6407	Schlitz	06642 /6940 // 069 /832051
Lohmann	Willi	Honnerskamp 2	4300	Essen 12	0201 / 356319 //
Loose	Gerhard	Viefhaushof 42	4306	Essen 13	0201 /212608 // -
Mand	Harald	Kleinflintbeker Straße 7	2302	Flintbek bei Kiel	04347 /3629 // 0431 /3013580
May	Holger	Marienstr. 9	5768	Sundern 2	02935 /1668 // -
Misioch	Waldemar	Adenauerring 25	8505	Röthenbach a. d. Pegnitz	0911 /506051 // 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940	Weinheim -Lützelsachsen	06201 /55052 // -
Müller	Kurt	Soltaustraße 24a	2050	Hamburg 80	040 /7246083 // 04151 /8891-37
Neueder	Jens	Rudolf-Then-Straße 32	7178	Gschlachtenbretzingen	0791 /42877 // 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209	Heidenrod 1	06124 /3913 // -
ObscherningKat	Helmut	1 rue des Bruyeres	F-68360	Soultz	0033/89-762690 // -
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418 //
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611 // 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /43791 //
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167 // 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551 // 04103 /605310
Rinio	Gerd	Rennbahnstraße	2000	Hamburg 74	- // -
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383 // 0203 /331383
Schmid	Alexander	St. Cajetan-Straße 38/VII	8000	München 80	089 /495326 // -
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	- // -
Schneider	Bernd	Wirtwiser-Straße 13	CH 8951	Fahrweid	0041/1/7480348 // -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897 //
Schroers	Horst-Dieter	Breslauer Str. 9	8016	Feldkirchen	089 / 9032615 // 089 / 9032615
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602 // -
Schut	Andre	Sanderstraße 26	1000	Berlin 44	030 / 6917861 //
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631 // 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228 /643853 // -
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926 // -
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585 // 05401 /30096
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153 /1564 // -
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131 /32860 // 06131 /395268
Tornow	Wilhelm	Ringweg 1	2301	Osdorf	04346/9922 // 0431/362055
Volz	Oliver	Dusestraße 13	7000	Stuttgart 80	0711 /731285 //
Waccus	Michel	Mühlhofweg 2a	CH 8266	Steckborn	054/612560 // -
Wacker	Fred	Weingärtenstraße 22	7554	Kuppenheim	07222 /48931 // -
Wagner	Günther	Gartenstraße 4	8201	Neubeuern	08035 /3361 // -
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686 // -

Stand: Juni 1988

Bitte überprüft Eure Daten auf Richtigkeit
und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

INFOFORM

An alle Autoren --- oder an die, die es noch werden wollen.

Aus gegebenem Anlaß möchte ich Euch einige Tips geben, die eine bessere übersicht und Form der INFO ermöglicht. Gleichzeitig hilft es mir, die Redaktionsarbeit zu vereinfachen. Ich bitte Euch daher, sich an diese **Richtlinie** für Redakteure zu halten.

-- DANKE --

1. ALLGEMEINES :

Durch das Verkleinern der Artikel bekommen wir viel Information auf kleinsten Raum. Dieses System wollen wir beibehalten, es ermöglicht uns nämlich das INFO mit geringem **Kosten und Material aufwand** zu erstellen. Um daher eine Verkleinerung Eures Beitrages in bester Qualität zu erhalten, ist folgendes nötig :

Textbeiträge oder Listings sollten ausgedruckt oder mit Schreibmaschine geschrieben sein. Als **Untergrund** für die Buchstaben und sonstige Zeichen empfiehlt sich weißes linienfreies Papier. Bitte sendet keine handschriftlichen Beiträge ein, da meist die persöhnliche **Handschrift** eines jeden nicht mehr an die in der Schule gelernte **Schönschrift** heranreicht und daher zu Entzifferungsproblemen führt.

Wer keinen Drucker/Schreibmaschine hat, kann mir ja sein Textfile auf Diskette zusenden. (Bitte Textverarbeitungssystem und Titel des File angeben)

Verwendet auch nicht gerade die kleinste Schriftart Eures Druckers oder ein altes Farbband, denn auch dies führt nach dem Verkleinern und Drucken zur Unleserlichkeit.

Bei Einsendung eines mehrseitigen Artikels (mehrerer Artikel) denkt bitte daran, die Reihenfolge der Blätter numerisch festzulegen (auf der Rückseite der Beiträge genügt). Ich hoffe, dadurch Verwechslungen innerhalb eines Beitrages oder ein Vertauschen der chronologischen Reihenfolge der Beiträge entgegenzutreten.

2. FORM :

Die Ausdruckform sollte eine Breite von 15 - 17cm haben, und die maximale Zeilenzahl pro Seite darf 62 Zeilen nicht überschreiten. Ich benötige die dadurch erstandenen freien Ränder für die Seitennumerierung /-markierung sowie als Kleberand zum Binden und Abheftlochrand für Euch.

3. ARTIKEL AUS ZEITSCHRIFTEN :

Beachtet bei Artikeln aus Zeitschriften, daß Text, der in farbigen oder gerasterten Flächen steht, durch das Kopieren meist schon schwer leserlich wird. Versucht beim Kopieren die Kopie mit einer helleren Stufe als normal zu erstellen. Ist die Textstelle trotzdem noch nicht **schwarz auf weiß**, solltet Ihr vielleicht eine Abschrift der entsprechenden Textstelle machen.

Ihr habt aber auch die Möglichkeit, mir das Original zukommen zu lassen. Ich habe die Gelegenheit, **heller** zu kopieren mit gleichzeitiger Verkleinerung. Dies ergibt meist eine brauchbare Veröffentlichung für das INFO. Das Original erhaltet Ihr auf jeden Fall mit der INFO zurück.

Hier sollte Euer Artikel (pro Seite) hineinpassen!

4. VERSAND :

Nachdem Ihr nun Euer **Schriftwerk** fertig habt, und Ihr es der Redaktion zukommen lassen wollt, solltet Ihr Eure Arbeit nicht zusammenfalten und in ein kleines Kuvert stecken. Diese Knicke geben dann beim Kopieren/Verkleinern graue Streifen, die die Buchstaben an dieser Stelle unkenntlich machen. Darum **ungeknickt** in ein der Größe entsprechendes Kuvert (max DIN A4) und ab an die Post. Portomäßig sind die Kosten für Euch ja gleich.

Wenn Ihr mir nicht unbedingt ein paar persönliche Mitteilungen zu machen habt, ist die Versendung als **Drucksache** der kostengünstigste Weg für Euch.

5. BEITRÄGE ZUR BÖRSE :

Beiträge zu diesem Thema schreibt bitte in folgender Art und Weise auf:

Sparte (Wer hat was, wer will was oder FrageKasten)
Euren Text (Angebot ,Wunsch oder Frage)
Euer Autogramm

Ich schneide die Teile aus Eurem Brief und Klebe sie dann in die entsprechende Sparte.

6. DISKETTEN :

Bitte gebt grundsätzlich Eure **DRIVE 1** an.
Lesen Kann ich bis zu 80 Track DS/DD jeweils in 5 1/4 und 3 1/2 Zoll-Format am TANDY.
Zurücksendung Eurer angekommenen Disketten ist Ehrensache.
Ihr denkt bei der Diskversendung natürlich auch an die richtige Versendungsform Eurer **Scheiben**. Am besten sind Kuverts mit **Luftpolsterung** --- daß wißt Ihr ja selbst.

Wenn ihr Euch an diese kleine **Autorenanleitung** haltet, dürfte das die Qualität unseres INFO's noch steigern. Ich hoffe, Ihr helft dabei mit.

Im voraus nochmals meinen DANK
Redaktion CLUB 80

CLUB 80

Clubinfo
der
TANDY - GENIE - und KOMTEK ANWENDER

25. AUSGABE

KONTAKTADRESSE : CLUB 80 / GERALD SCHRÖDER / AM SCHÜTZENPLATZ 14 / 2105 SEEVETAL 1
TEL.: 04105 / 2602

- I N H A L T S V E R Z E I C H N I S -

Seite:
und Autor:

Clubinternes

Neues vom Vorstand 1
Gerald Schröder
Happy Birthday 2
Jutta Obermann
Aktuell und wichtig 3
Redaktion
Vorstellungen 4
Günter, Wolfgang, Heinz-Dieter

Software

Wogegen selbst ZEUS vergeblich ... 05 - 06
Bangen um Banken .... 07 - 08
Ableitung der Negationsregel für Bin. 09 - 11
Klaus-Jürgen Mühlenbein
Binärumwandlung, letzter (?) Teil .. 12 - 13
Screening mit DSMBLR/CMD 14
Kurzanleitung für SCREENER/CMD ..... 15 - 16
Kleine Hexen ohne Aufwand 17
Die Pratze läßt das Mausen nicht ... 18 - 19
Vergleich zweier Disk-Dateien ...... 20 - 23
Bedingt assemblieren mit ZEUS ...... 24 - 27
Bedingte Assembly - notfalls ohne IF 28 - 29
Arnulf Sopp
Der Teufel steckt oft im Detail 30
HRG - Bildübertragung .... 31 - 34
Hartmut Obermann
Grafik unter CP/M .... 35 - 36
Alexander Schmid
Z-CP/M, eine deutliche Verbesserung 37 - 44
ZCPR33 - Installation auf M4 und ... 45 - 48
Rüdiger Sörensen
Assembler unter CP/M .... 49 - 54
Gerald Schröder

Seite:
und Autor:

Hardware

Freiheit für den Bus 55
Gerald Schröder
Die Ramdisk und das elfte Gebot 56
Alexander Schmid
Prof 180x und mc68000-ECB .... 57 - 59
TRS 80 Japan Modell aufbohren ...... 60 - 67
Helmut Bernhardt

Börse

Wer hat was -- wer will was 68

Sonstiges

Wo die Tomaten fliegen .... 27 / 44
Artikel aus ...

Die letzten Seiten

Impressum 69
Schluß 70
Redaktion
Edles aus ELCOMP .... am INFO-Ende
Klaus-Jürgen Mühlenbein
Mitgliederadressenliste .... am INFO-Ende
Redaktion

Tachchen!

Zuerst möchte ich mich dafür entschuldigen, daß Ihr erst jetzt das dritte Info dieses Jahres erhaltet. Wie Ihr gemerkt habt, war das Drucken ein größeres Problem als vorauszusehen war. Dafür ist die Druckqualität aber meiner Meinung nach sehr gut. Ich hoffe, daß ich in Euer aller Namen spreche, wenn ich mich bei Horst-Dieter dafür bedanke und ihn bitte, das Info weiterhin so zu drucken.

Über eins bin ich allerdings etwas enttäuscht: Eure schreibenden Tätigkeiten (bzw. die nicht stattfindenden). Obwohl zwischen dem ersten und zweiten Info dieses Jahres mehrere Monate lagen, kamen nicht viele Artikel zusammen. Ich gebe natürlich zu, daß Jens noch einige Artikel für dieses Info zurückgehalten hat, aber trotzdem könnten es mehr sein.

Also nochmal die Aufforderung: Schreibt Artikel! Ihr fragt Euch: "Worüber?" Die Antwort: "Alles!" Es ist keineswegs so, daß Arnulf und Helmut die Exklusivrechte haben oder für ihre Artikel bezahlt werden. Sie dokumentieren nur alles, was sie tun, und teilen es Euch mit, damit Ihr daraus lernt, Anregungen bekommt und auch etwas ähnliches versucht. Es wäre doch nun sehr schön, wenn Ihr den (wenigen) regelmäßigen Schreibern mal zeigt, was Ihr macht. Immerhin wäre es ja möglich, daß diese auch etwas davon interessiert.

Nun könnt Ihr zwar sagen: "Ich bin ja der einzige, der ... macht; das interessiert doch niemanden." Aber diese Ausrede gilt nicht. Seht Euch nur Arnulf an: er ist das einzige Mitglied mit einem Genie IIIs und dann schreibt er frecherweise gleich seitenlange Artikel dazu. Ich möchte diese aber nicht missen, denn allein die Assembler-Tricks und ungewöhnlichen Ideen, die er in seinen Listings offenbart, sind Gold wert. Warum sollten Eure Berichte weniger interessant sein? Ich würde gern mal etwas über Eure Erfahrungen mit verschiedenen Textverarbeitungsprogrammen hören; oder die Masse Leute, die jetzt mit einem Banker von Helmut arbeitet: was macht Ihr eigentlich damit? Hat denn niemand mal Programme dazu geschrieben oder die vorhandenen verbessert oder sie überhaupt benutzt? usw. Die Liste könnte ich endlos fortsetzen.

Meine Bitte lautet: Seht Euch an, was Ihr mit dem Rechner macht und schreibt dann darüber einen Artikel. Oder lest Euch mal den Rest dieses Infos durch und gebt den Schreiberlingen das, was die Psychologen "Feedback" nennen: Schreibt im Info, was Ihr dazu denkt, was Ihr nicht verstanden habt, was besser sein könnte.

Übrigens könntet Ihr auch eine Eure Meinung zu einem Buch äußern, sofern Ihr mal eins lest. Einen Anlauf hatte ich ja schon in Info 21 gemacht und es sind inzwischen einige wenige Besprechungen erfolgt, aber ich kann mir nicht denken, daß Ihr so wenig schmökert. Ein Steckbrief zur Suche in Euren Schränken: Bücher sind diese Dinger mit den vielen beschmutzten Seiten, die auch noch gründlich durchnumeriert sind. In unserer Clubbibliothek sollen auch einige Dutzend davon stehen (bzw. liegen und sich langweilen). Fragt mal Werner danach!

So, jetzt habe ich Euch genug aufgemuntert. Nehmt's Euch mal zu Herzen, damit das Info wieder ein paar Seiten mehr füllt.

Gerald Schröder

Happy Birthday

Servus und Grüß Gott an alle "Geburtstagskinder" des
CLUB 80

als erstes möchte ich mich für die verspäteten Geburtstagswünsche entschuldigen, die einige sicherlich treffen werden. Aus diesem Grunde habe ich diesmal nicht, wie üblich, drei Monate sondern vier Monate berücksichtigt.

Zwei unserer Mitglieder haben einen besonderen Geburtstag zu feiern. Im August und September werden Arnulf Sopp und Helmut Bernhardt, unsere beiden Nordlichter, 40 Jahre alt.
Meinen allerherzlichsten Glückwunsch an beide.

Ach Gott, beinahe hätte ich einen ganz wichtigen Geburtstag vergessen. Denn man glaubt es kaum, aber mein werter Göttergatte wird 30 Jahre alt (Wer ist es? Tip: siehe Geburtsjahr bei unter stehender Liste).

Die genauen Daten der drei und alle übrigen findet Ihr nachfolgend.

Juli	**: 18.**	**Hartmann**	**Hans-Günter**
	19.	**Brandl**	**Hermann**
	30.	**Retzlaff**	**Bernd**
August	**: 02.**	**Issig**	**Günter (Neumitglied)**
	05.	**Stephan**	**Hans-Martin**
		Heidenreich	**Ulrich**
	10.	**Bernhardt**	**Helmut (1948)**
	15.	**Misioch**	**Waldemar**
	16.	**Braun**	**Harald**
	17.	**Wagner**	**Günther (Clubgründer)**
	24.	**Obermann**	**Hartmut (1958)**
September	**: 08.**	**Albers**	**Herbert**
	11.	**Rychlik**	**Andreas**
	23.	**Sopp**	**Arnulf (1948)**
	24.	**May**	**Holger**
Oktober	**: 03.**	**Schneider**	**Bernd**
	15.	**Stober**	**Reiner**
	30.	**Held**	**Manfred**

25 August

Im Namen des Vorstandes wünsche ich Euch allen ein fröhliches Feiern und weiterhin alles Gute

Jutta

Aktuell und wichtig!

Jutta hat das alte Clubkonto aufgelöst und ein neues eröffnet. Die Bankverbindung lautet jetzt:

Hartmut Obermann Sonderkonto
Postfach 27
6209 Heidenrod

Kto-Nr.: 496 071-605
beim Postgiroamt Frankfurt am Main
BLZ: 500 100 60

Achtung!!

Wer eine Teilnahmebescheinigung für das Clubtreffen braucht, soll sich unter Beilage des Rückportos bei Gerald Schröder melden. Die Kosten für das Clubtreffen können dann von den Steuer abgesetzt werden.

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....29. Oktober 1988
ORGATECHNIK ....Köln 20. - 25. Oktober 1988

"Unsere Neuen!"

Hallo Clubfreunde,

hiermit stelle ich mich kurz vor,

Ich heiße Günter Issig, geb. 02.08.57, wohnhaft in Heilbronn. Beruflich habe ich nichts mit EDV zu tun und bin deshalb noch total unwissend.

Durch Zufall bin ich an einen TRS 80/II gekommen. Dies warf bis jetzt jedoch jede Menge Probleme auf.

Es wäre toll, wenn sich jemand findet, der mit mir in Kontakt treten und mir bzgl. Bedienung etc. weiterhelfen kann.

Bis "hoffentlich" bald

Hallo Clubfreunde !

Als neues Clubmitglied möchte ich mich kurz vorstellen.

Ich heiße Wolfgang Preußing, wohnhaft in Essen, bin 32 Jahre alt, verheiratet und habe eine vierjährige Tochter.
Über das Elektronikbasteln kam ich zur Computerei. 1981 erwarb ich einen TRS 80 Mod. 1. Im Laufe der Zeit kamen ein Expansions Interface der Fa. CE Computer Elektronik, ein Laufwerk BASF 6106 (SS,DD), ein Drucker CP 80 Type II und ein Akustikkoppler Tandy AC-3 hinzu. Selbst gelötet habe ich die I/O Karte aus 80-Micro 12/84 mit dem 8255. Z. Zt. versuche ich die RB V 24 - Karte zum Laufen zu bringen. Bisher noch ohne Erfolg.

Seit etwa zwei Jahren habe ich auch beruflich mit EDV zu tun (IBM AT unter Open Access II).

Auf gute Zusammenarbeit.

Hallo Gleichgesinnte!!

hiermit möchte auch ich mich als neues Mitglied im Club kurz vorstellen.

Mein Name ist Heinz-Dieter Meklenburg, geb.01.10.50 wohnhaft in Leck, ca. 17 Km von der Dänischen Grenze entfernt. Beruflich bin ich als Elektromaschinenbauer ausgebildet worden, und habe dann in Abendschule meinen Elektronic-Pass absolviert.

Mein erster Computer war der ZX 81 von Sinclair im Jahre 73, den ich immer noch besitze, und mit dem ich schon nach kurzer Zeit auf die Assemblersprache kam.
1979 bekam ich dann den GENI I,bei dem ich mit der gleichen Programiersprache weiter gemacht habe.

Im März 88 kaufte ich mir allerdings einen Atari 1040 ST dazu, um mit ihm speziell Schaltpläne und Layout`s zu zeichen.

Ich hoffe auf eine gute Zusammenarbeit und verbleibe hiermit Euer

Heinz-Dieter Meklenburg

* *Wogegen selbst **ZEUS** [einer unter vielen Göttern] * *
* * vergeblich kämpft ... * *

– „Links steht HEX, rechts steht ASCII". –

– „Hier können Sie leicht ASCII, also Klartext lesen". –

Diesen und ähnlichen Unsinn kann man immer wieder lesen, sogar in Veröffentlichungen großer Meister, deren Name Gewicht hat in der Runde der Software- und Maschinensprache-Experten! [Natürlich nicht bei Clubmitgliedern...]

Der erste Satz bezieht sich z.B. auf die Sektordarstellung in SUPERZAP. In der Tat: „Links steht HEX..." Aber was steht rechts? – Antwort: KLARTEXT – nicht „ASCII"! „Klartext" allerdings nur, wenn es sich um ein BASIC-Programm handelt; bei Maschinenprogrammen erscheinen nur die „messages" im Klartext. (Woraus messerscharf folgt, daß alles übrige in einem Maschinenprogramm „unklar" ist? --- Aber nur für die Ahnungslosen. Zu denen gehöre ich. Aber ich <u>vermute</u> einen Sinn dahinter...
[An ZEUS muß man eben glauben])

Rumpelstilzchen singt heute:

„Glaubst du wirklich, jeder weiß,
was A.-ES.-CE.-I.-I. heißt?" *)

ASCII = American Standard <u>CODE</u> for Information Interchange = Amerik. Code f. Ifo-Austausch.

*) Rumpelstilzchen war kein Dichter...

Der Ton liegt auf <u>CODE</u>, ihr Lieben! Buchstaben, Ziffern und Zeichen, kurz: „Characters" wurden codifiziert! Das „Gegenteil" des Codes, der <u>Klartext</u>, das sind die Zeichen u.s.w., die „Characters". Welcher BASIC-Freak kennt nicht die Codier-Anweisung für den Buchstaben Q:

<u>ASC</u> (Q) = 81_{dez} = 51H !

Dieser Freak weiß auch, daß die gegenteilige Anweisung

<u>CHR</u>$(81) = <u>CHR</u>$(&H51) = „Q"

lautet! <u>Das</u> ist <u>Klar</u>text – nicht ASCII! Klar?

Und eben das steht <u>rechts</u> bei einem Sektordump! – Gleichzeitig sehen wir eben, daß die ASCII-Darstellung sowohl durch HEX- als auch durch Dezimalwerte möglich ist. Im Sektordump (z.B. von SUPERZAP) steht der linke, d.h. der <u>ASCII-Teil</u>, in HEX. Er könnte hier – rein theoretisch – auch in DEZ stehen, wenn das technisch möglich wäre. Nur die Grundstruktur (2x4 Bit) unserer EDV erlaubt das nicht. – Aber deshalb ist der Teil, der <u>rechts</u> steht, noch lange <u>kein ASCII</u>, sondern die Rückübersetzung des Codes in die <u>Zeichensprache.</u> Daß diese bei Maschinenprogrammen keineswegs immer leserlicher „Klartext" ist, liegt einfach daran, daß auch die Maschinenbefehle (2^n-Bit-Codes) stur in „Zeichen" übersetzt werden [-von wem ??-], nicht in mnemonics. <u>Und schon gar nicht – in „ASCII"!</u>

Ich danke euch, liebe Leser, daß ich meinem „Ärger" einmal Luft machen durfte – Ärger über den Verstoß gegen unser „Grundgesetz": Die LOGIK. – KaJot

* * * * BANGEN um BANKEN * * * *

Bankeinbrüche schrecken mich nicht. Ich kann nichts dabei verlieren.
"DISAGIO" (fachmännische Aussprache: Diss-Aschi-Oh!) ist, glaube ich, ein unanständiges Wort.
"Diskont" klingt schon vertrauter für den Freak; es heißt wohl soviel wie "Diskette eingeschaltet" (? Das "t" am Schluß ist sicher wieder einer der üblichen Druckfehler.)
Aber "Rediskont"? Es erinnert mich an die Zeiten, als es noch keinen Kugelschreiber gab und man mit Stahlfedern des Typs "TO" oder "REDIS" schrieb. In der Tat: Mit der Rediskon(n)te ich viel besser schreiben als heute! Manchmal sogar "Schönschrift"!
Aber ob ich's heut' noch mit der Rediskönt...? -
"Diskönt' i' nimmer!"
Schließlich fiel mir in letzter Zeit ein Begriff auf, der sich auf "Wort" reimt: der "Port"! Also, ich muß schon sagen, <u>mein</u> "Portemonnaie" (schreibt man's so?) ist alles andere als ein "Hafen" - und schon gar kein sicherer!!

Kurzum: Ihr versteht jetzt schon klipp und klar,
daß ich wohl schon immer ein *GREENHÖRNER* war,
was Bankwesen, Ports und Verwandtes betrifft.
Ich hatt' diese Klippen "erfolgreich" umschifft...
Doch seit ich Besitzer - wie kam's in den Sinn? -
von 8 (acht) beachtlichen Banken nun bin (!),
kenn' ich nur noch heimlich ein einzig' Verlangen:
<u>Nie möchte ich mehr um die Banken mich bangen!!</u>

Was nützen die harden und soften Experten,
die freundschaftlich Tat, Rat und Hilfe gewährten,
die man mit Fragen und Klagen gehetzt - satt! -
wenn sich im Kopf ein "Hangup" festgesetzt hat!
(Ich faß mich an die Stirne
und fühl' 'ne weiche Birne!)

Nun ist ein Mondlicht aufgegangen,
die güldnen * * * * *, ach, sie prangen
am Monitor so klar!
Der Ce-Pe-Uhu nicht mehr schweiget
und aus des Keyboards Tiefen steiget
<u>der BANK-Direktor!</u> . . .

WUNNERBAR!

(Nachtrag:) Das Gedicht heißt: "Die **RAMDISK**".

Für die meisten von euch natürlich ein alter Hut.
Aber vielleicht nützt doch dem einen oder anderen (insonderheit jenen, die nicht in IDSTEIN waren) das kleine BASIC-Programm, das ich zum Testen eines Bankers (nicht Bankerts!) geschrieben habe.

Womit ich auf keinen Fall den Experten, die sowas in Maschinensprache vollziehen, Konkurrenz machen will - Gott behüte! Weiß ich doch nicht einmal, ob das Testprogramm auch für andere Bank-Unternehmen geeignet ist. Ich fürchte sogar, daß es nur meiner Bank-Filiale schmeckt (s.LOGO). Aber trotzdem: Laßt mich doch auch einmal auf meinen Käse stolz sein! Mir schmeckt er...

Und vielleicht ist dies eine Anregung für jene Clubkollegen, die noch nie etwas hier beigetragen haben, weil sie glaubten (oder fürchteten), ihr Geistesprodukt sei Käse... Merket:

Je schlimmer für die Neese,
je besser <u>schmeckt</u> der Keese!!

Dieses schwöre ich bei Gott!
KaJott.

```
1 CLS:PRINT"*****************************************************
***********
2 PRINT"*    Programm-Name :   BANKEN/TST          Datum : 24.07.88
        *
3 PRINT"*    Verfasser :       K.-J.Muehlenbein, Weinheim
        *
4 PRINT"*    Das Programm testet Helmut Bernhardts Banker
        *
5 PRINT"*              ACHTUNG! Vor dem Programmlauf muss im DOS
        *
6 PRINT"*                 HIMEM = 7FFFH      gesetzt werden!
        *
9 PRINT"*****************************************************
*******
10 CLEAR260:DEFINTA-K:DEFSTRL-Z
20 PRINT:PRINTCHR$(222)"MENUE
"CHR$(222)"-----"
30 PRINT"
          1  Bank anwählen
          2  Texteingabe in die Bank
          3  Textausgabe von der Bank
          4  Ende"
40 INPUT"Wahl ";C
50 ONCGOSUB70,80,120,160
60 GOTO165
70 PRINT:INPUT"Bank-Nummer ";B:OUT236,B:RETURN
80 PRINT:LINEINPUT"Texteingabe <nicht mehr als 3 Zeilen !>

";T
90 PRINT:INPUT"ab welcher Adresse speichern <zwischen 32768 und
65280!> ";S:A=VAL(S)-65536
110 FORI=OTOLEN(T)-1:POKEA+I,ASC(MID$(T,I+1,1)):NEXT:RETURN
120 PRINT"Ausgabe aus Bank"B"<J/N>";:INPUTX
130 IFX="N"ORX="n"GOSUB70
140 CLS:PRINTCHR$(255):PRINTCHR$(255)
145 FORJ=OTOLEN(T)-1
150 PRINTCHR$(PEEK(A+J));
155 NEXT:RETURN
160 END
165 PRINT§980,"Weiter mit <ENTER>
170 Y="":Y=INKEY$:IFY=""THEN170ELSECLS:GOTO20
```

Ableitung der Negationsregel für Binärzahlen

(von K.-J.Mühlenbein, Weinheim; 06201/55052)

Vorbemerkung

Die Addition von Binärzahlen ist einfach.
Die Subtraktion geschieht durch Addition des negativen Wertes.
Den negativen Wert einer Binärzahl erhält man durch Umkehr der Setzungen; d.h. Einser werden zu Nullen und Nullen zu Einsern gemacht; anschließend wird 1 addiert.
Wie kommt es zu dieser eigenartigen "Negationsregel"?
Dies soll nachstehend abgeleitet werden.

A. Grundlagen

1) Die allgemeine Form eines nach "Stellenwertigkeit" gebildeten Zahlensystems lautet:

$$a_n B^n + a_{n-1} B^{n-1} + \dots + a_i B^i + \dots + a_2 B^2 + a_1 B^1 + a_0 B^0$$

worin B "Basis" genannt wird, n ganzzahlig ist und die a_i ganzzahlige Koeffizienten sind, für welche gilt: $0 \leq a_i \leq B-1$.

Die Exponenten geben den "Stellenwert" an; n kennzeichnet die höchste Stelle.

2) Der Bereich eines beliebigen, nach dem "Gesetz der Stellenwertigkeit" gebildeten Zahlensystems Z(B;S) sei durch eine maximale Stellenzahl S=n+1 begrenzt.

Folgerung: Ein Dezimalsystem (B=10) mit höchstens S=4 Stellen hat dann die Zahl "9999" als höchsten Wert. Die Zahl "10000" (und größere) ist in diesem System nicht definiert ("nicht-existent"), denn für dieses System haben nur die 4 letzten Stellen Gültigkeit. Es ist dann also "10000" = (1)0000 = Null!
Der so begrenzte Wertebereich umfaßt einschließlich der Null B^S ganze Zahlen.
Diese Einschränkung berücksichtigt die begrenzte Anzahl Bits (8 bzw. 16) bei der Zahlendarstellung durch den Computer.

3) Die Anzahl m (gültiger) Stellen in einer solchen Zahlendarstellung ist gleich der Anzahl Summanden gemäß Definition 1), d.h. es gilt: m = n + 1.

Beispiel: Im System mit B=2 wird die Dezimalzahl '1' dargestellt durch $2^0 = 1$. Hier ist n = 0 und m = n + 1 = 1.
Die Dezimalzahl '5' wird dargestellt durch
$1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$; es ist n = 2; m = 2 + 1 = 3.

4) Im System Z(B;S) sind nach 2) alle Ausdrücke A = Null, für die $A \geq B^S$ gilt, denn bereits B^S hat gemäß 3) S+1 Stellen, ist also im System "nicht-existent".

5) Eine Zahl $\bar{X}$ heiße "komplementär zu X", wenn gilt: $\bar{X}+X = 0$.
$\bar{X} = -X$ entspricht somit der Negation. Wegen 4) kann gesetzt werden: $\bar{X} + X = B^S = 0$

6) Man findet demnach den zu X komplementären Wert $\bar{X}$ durch die Bildung:

$$\bar{X} = B^S - X$$

Da $\bar{X} = (-X)$, läßt sich als "Allgemeine Formel" schreiben:

$$\boxed{(-X) = B^S - X}$$

B. Anwendung auf das Dezimalsystem

Es sei B=10 und S=4. ein „eingeschränktes" Dezimalsystem Z(10;4).
Der Wertebereich reicht dann von 0 bis $10^4 - 1 = 9999$ und umfaßt 10^4 ganze Zahlen.
Der zu X komplementäre Wert $\bar{X}$ muß die Bedingung $\bar{X} + X = B^S = 10^4$ erfüllen (Satz A5). $\bar{X}$ ist daher die Größe, die den Wert X zu 10000 "komplementiert".
10^4 hat jedoch m = n+1 = 4+1 = 5 Stellen und besitzt somit für das Zahlensystem Z(10;4) den Wert "Null": $\bar{X} + X = (1)0000$. Um im System zu bleiben, wird 10000 durch 9999+1 ersetzt. Die Gleichung schreibt sich nun:

$$\bar{X} = 9999 + 1 - X = (9999-X) + 1$$

Die "Allgemeine Formel" ist einfacher:

$$\bar{X} = (-X) = 10^4 - X$$

Beispiel:
Die zu X=5703 im System Z(10;4) komplementäre Zahl $-\bar{X}$ ergibt sich wie folgt:

$$(-X) = (9999 - 5703) + 1 = „4297"$$

Bei der Subtraktion 9999-5703 wird jede Ziffer des Subtrahenden zur Ziffer 9 "komplementiert"; dies ergibt das "Neunerkomplement" 4296 als Zwischenwert. Nach Addition von 1 erhält man das "Zehnerkomplement" 4297. Im Sinne des hier definierten Dezimalsystems mit S=4 gilt daher die "Negation": -5703 = „4297"!

Man merke sich:
Jede Ziffer von X wird zur höchsten Ziffer im System komplementiert!
(Die höchste Ziffer ist stets B-1; im Dezimalsystem (B=10) also die "9").

C. Anwendung auf das Binärsystem

Im Binärsystem (Dualzahlensystem) des 8-Bit-Codes ist B=2 und S=8 ==> Z(2;8).
Sein Bereich umfaßt die Werte von 0 bis $B^8 - 1 = 2^8 - 1 = 255$, insgesamt die Anzahl $B^S = 2^8 = 256$ ganze Zahlen (einschl.Null). Ein Wert $B^S = 2^8$ ="256" ist in diesem System jedoch nicht definiert, denn diese Zahl hätte m=n+1=8+1=9 Stellen.
Statt "256" kann aber "255+1" geschrieben werden. Den zu X komplementären Wert $\bar{X}$ = (-X) erhält man daher wieder aus der Bildung:

(-X) = höchster Systemwert + 1 - X
bzw.
(-X<bin>) = 11111111<bin> - X<bin> +1<bin>
} (Gleichung C)

Die Rechenregel für die Negation von Binärzahlen besteht also wieder darin, daß jede "Ziffer" (d.i. jedes Bit!) von X zur höchsten Ziffer des Systems (das ist hier die "1") "komplementiert" - d.h. das "Einerkomplement" gebildet - und danach 1 addiert wird!
Das "Einerkomplement" der Null, also die Ergänzung zur Eins, ist natürlich die Eins. Das "Einerkomplement" der Eins, also die Ergänzung zur Eins, ist - nun, wer errät es? ....
Richtig: Die Null!

Die Addition einer Eins ergibt dann das sogenannte "Zweierkomplement". (Eine nicht streng konsequente Bezeichnung; denn das "Ergänzende" (das Komplement) ist hierbei nicht die Zwei, sondern die addierte "1"!)

Die schnellste Lösung erhält man jedoch wieder durch die "Allgemeine Formel":

$$(-X) = 2^8 - X$$

Beispiel:
Im Dezimalsystem (mit unbegrenzten Stellen) ist -100 der negative Wert (die "Negation") von 100.
Wie lautet -100 im 8-Bit-Binärsystem ?
Die "Allgemeine Formel" liefert:
(-X) = 2^8 <dez> - 100<dez> = 256<dez> - 100<dez> = "156"<dez>.
Versuchen wir, dies binär darzustellen, so müssen wir die umständlichere "Formel" (Gleichung C) zuhilfe nehmen, da 256<dez> sich nicht im [redacted] 8-Bit-System, also nicht mit einem Byte darstellen läßt:

	Höchster Wert:	1111 1111	<bin>	|-
	X = 100<dez> =	0110 0100	<bin>	
	Einer-Komplem. =	1001 1011	<bin>	
	+ 1<dez> =	0000 0001	<bin>	
-X =	- 100 <dez> =	1001 1100	<bin>	= "156"<dez>

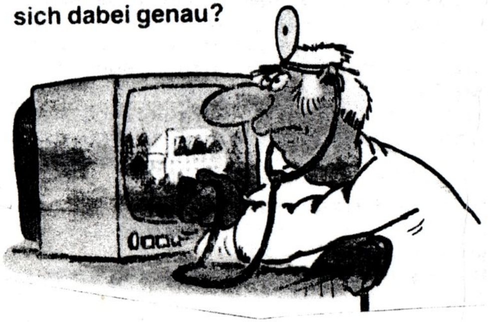
Etwaige "Overflow-Bits", die bei dieser Operation intern auftreten können, entfallen bei der "Allg. Formel" automatisch.

PS: Zahlen in Anführungszeichen bedeuten, daß nicht zwischen „Normaldarstellung" und Darstellung als „Komplementzahl" unterschieden wurde. (In der „normalen" Mathematik besorgt dies das Minuszeichen.)

KeJot

Unser Musikrätsel

Wer kennt sie nicht, die hübsche Titelmelodie aus der Fernseh-Serie „Schwarzwaldklinik"! Um was für ein Musikstück aber handelt es sich dabei genau?



A – Um die Fraktur in Zeh-Mull, Knöchelverzeichnis 256, von Ludwig van Bettpfosten?

B – Um die Akute Sinfonie in Tee-Dur, gespielt von den Wiener Vielhormonikern?

C – Um die Arie „Wie eiskalt ist dies Hehnchen", dargeboten von den Egerländer Simulanten unter der Leitung von Skalpellmeister Brinkmann?

Binärumwandlung, letzter (?) Teil

In meinem Beitrag "Binärumwandlung aus Hex und Dez und kürzere E-Routine in SYS14/SYS" ging es um ein interessantes und häufig brauchbares Programm, das Zahleneingaben in Hex oder Dez binär nach DE lädt. Aus gegebenem Anlaß habe ich mich dieser Routine noch ein paarmal gewidmet. Sie ist jetzt noch etwas kürzer geworden, hauptsächlich aber besser:

Wie gehabt zeigt HL beim Einsprung auf den Beginn des Eingabestrings. Bei der Rückkehr zeigt es auf das Zeichen nach der ASCII-Zahl. Bei DOS-Befehlen ist das z. B. 0Dh (ENTER) oder ein Trennzeichen zum nächsten Befehlsparameter (Komma oder Blank). DE enthält die Zahl in ihrer binären Entsprechung. Der Akku wird verändert. Das Flag-Register enthält Z=1, (Zero-Bedingung), wenn es nur gültige Hex- oder Dez-Ziffern oder eines der üblichen Trennzeichen gab (kein Fehler). Wenn ein Zeichen des Strings regelwidrig war, ist die NZ-Bedingung erfüllt (der Akku enthält in jedem Fall den Fehlercode "schlechte Parameter", 2Fh). BC, das nur innerhalb der Routine eine Funktion erfüllt, wird nicht verändert.

Ein Anwenderprogramm oder ein entsprechend angepaßtes SYS-File kann folgendermaßen von den Vorteilen profitieren: Zahleneingaben werden zwar auf gültige Zeichen überprüft, aber das Z-Flag kann ignoriert werden. So ist es möglich, in einer Eingabe das Trennzeichen zur nächsten Eingabe wegzulassen. Außerdem ist bei dieser Methode der Hex-Kenner "H" überflüssig, wenn es eindeutige Hex-Ziffern gibt (A-F). Wenn es aber nur Ziffern gibt, die auch das dezimale Zahlensystem kennt (0-9), ist "H" erforderlich. 27 ist nun einmal nicht dasselbe wie 27h. Aber für diese Routine ist 2A=2Ah, weil die Ziffer A nicht im dezimalen System vorkommt. Dann ist aber NZ erfüllt, damit es der Anwender, wenn er will, ganz genau nehmen kann.

Zuweilen sollte er das. Ist nämlich das dem Zahlenstring folgende Zeichen zufällig eine Hex-Ziffer (A-F), dann wird sie klaglos gelesen und ausgewertet. In solchen Fällen sollte man schon ENTER, ein Komma, ein Blank oder (bei Hex-Eingaben) H anfügen. Oder eben das Z-Flag beachten.

Es gibt einen Ausnahmefall, bei dem "kein Fehler" (Z=1) signalisiert wird, obwohl das Trennzeichen irgendwelcher Quatsch sein kann: Wenn HL auf Rubbish, bloß auf keine Dez- oder Hexzahl zeigt (gar keine Eingabe), dann ist nur scheinbar alles in Butter. DE enthält dann 0000h.

Eine ganze Reihe von SYS-Programmen und Anwender-Files akzeptieren dezimale oder sedezimale Eingaben. Ihre Umwandlungsroutinen können alle durch einen nur drei Bytes langen CALL an das irgendwo residente Programm ersetzt werden. Wo es residieren soll, muß sich der geneigte Leser je nach seiner Hardware aushecken. Im Listing ist die Standard-Adresse 5200h angegeben. In CALVA-DOS, dem Nachfolger von G-DOS 2.4 für das Genie 3s, lautet sie 3B68h.

Arnulf Sopp

```
5200                  ORG     5200h
     ;UP liest ASCII-Zahleneingaben in Hex und Dez aus (HL) binär nach DE ein
5200  C5      start   PUSH    BC              ;retten
5201  E5              PUSH    HL              ;dto. Befehlszeiger
5202  010001          LD      BC,0100h        ;B (- Flag: 1. Durchlauf, C (- Konst. 00
5205  CD2152          CALL    cnvrtup         ;Eingabe einlesen und binär umwandeln
5208  7E              LD      A,(HL)          ;letztes Zeichen nochmals einlesen
5209  D641            SUB     'A'             ;binär umwandeln
520B  FE08            CP      08h             ;07 wäre Hexkenner 'H' (maximal erlaubt)
520D  300B            JR      NC,exit1        ;raus m. Fehlerflag, falls unerlaubtes Z.
520F  E3              EX      (SP),HL         ;HL (- Stringzeiger
5210  41              LD      B,C             ;Flag: 2. Durchlauf des UP cnvrtup (Hex)
5211  CD2152          CALL    cnvrtup         ;Rest der Eingabe einlesen und umwandeln
5214  7E              LD      A,(HL)          ;letztes Zeichen nochmals einlesen
5215  FE48            CP      'H'             ;Kenner für Hex-Eingabe?
5217  2003            JR      NZ,exit2        ;ohne Hex-Kenner: 'schlechte Parameter'
5219  23              INC     HL              ;Befehlszeiger hinter letztes Zeichen
521A  CB48    exit1   BIT     1,B             ;wurde das Segment cnvrtup erreicht?
521C  3E2F    exit2   LD      A,2fh           ;Fehlercode 'falsche Parameter'
521E  C1              POP     BC              ;Stack bereinigen
521F  C1              POP     BC              ;Register restaurieren
5220  C9              RET

5221  51      cnvrtup LD      D,C             ;DE (- Ausgangswert 0000 für Binärzahl
5222  59              LD      E,C             ;(oben MSB, jetzt LSB)
5223  7E      cnvloop LD      A,(HL)          ;Zeichen einlesen
5224  D630            SUB     '0'             ;ASCII -) binär
5226  FE0A            CP      0ah             ;höher als Dez-Ziffer? (evtl. Hex-Ziffer)
5228  380A            JR      C,ciphok        ;falls Dez-Ziffer (erlaubtes Zeichen)
522A  CB40            BIT     0,B             ;ist es der 2. Durchlauf des UP? (Hex)
522C  C0              RET     NZ              ;fertig, falls nein (1. Durchlauf, Dez)
522D  D611            SUB     11h             ;falls Hex-Ziffer: 'A' -) 00 usw.
522F  FE06            CP      05h+1           ;noch erlaubte Hex-Ziffer? ('F' = 05)
5231  D0              RET     NC              ;Ende, falls nicht 'A' ... 'F'
5232  C60A            ADD     A,0ah           ;'A' ... 'F' -) 0A ... 0F
5234  E5      ciphok  PUSH    HL              ;Umwandlung ok., Befehlszeiger retten
5235  62              LD      H,D             ;HL (- Zwischenergebnis (MSB)
5236  6B              LD      L,E             ;(LSB)
5237  CB88            RES     1,B             ;Flag: UP cnvrtup wurde ganz durchlaufen
5239  29              ADD     HL,HL           ;*2 (Zwischergebnis links rotieren)
523A  29              ADD     HL,HL           ;*4 (um zwei Stellen)
523B  CB40            BIT     0,B             ;2. Durchlauf des UP? (bei Hex-Eingabe)
523D  2003            JR      NZ,convdez      ;falls nein (Dez-Eingabe)
523F  29              ADD     HL,HL           ;*8 (Hex-Eingabe)
5240  1801            JR      convhex         ;dort weiter
5242  19      convdez ADD     HL,DE           ;plus Zwischenergebnis (Dez: *5)
5243  29      convhex ADD     HL,HL           ;Hex: *16, Dez: *10
5244  5F              LD      E,A             ;DE (- letzte Ziffer (Einerstelle)
5245  51              LD      D,C             ;als 16-Bit-Zahl (MSB = C = 00)
5246  19              ADD     HL,DE           ;(bei Hex: vorläufiges) Endergebnis
5247  EB              EX      DE,HL           ;DE (- Enderg. (Hex: vorl., Dez: endg.)
5248  E1              POP     HL              ;Befehlszeiger
5249  23              INC     HL              ;nächste Stelle im Befehlsstring
524A  18D7            JR      cnvloop         ;dort weiter
5200                  END     start

00000 Fehler

ciphok  5234   cnvloop 5223   cnvrtup 5221   convdez 5242   convhex 5243   exit1   521A
exit2   521C   start   5200
```


Screening mit DSMBLR/CMD

Es gibt Disassembler, die selbsttätig DEFMs und DEFBs erzeugen. Sie haben jedoch alle zwei entscheidende Nachteile: DEFW kommt da nicht vor, weil der Disassembler nicht ahnen kann, was als Byte, und was als Word von Belang ist; DEFM und DEFB werden nur dann vermutet, wenn ansonsten Schwachsinn wie etwa LD A,A als Programmcode erzeugt werden müßte, oder wenn ein nicht von Zilog dokumentierter (aber u. U. durchaus ausführbarer und sinnvoller!) Befehl auftaucht. In allen anderen Fällen wird gnadenlos Programmcode erzeugt, auch wenn es sich in Wirklichkeit um eine Tabelle oder gewöhnlichen Text handelt.

Da lobe ich mir DSMBLR. Er bringt diese reichlich fragwürdige Grundintelligenz nicht mit, ist dafür aber zugunsten eines großen Source-Puffers schön kurz. Aber ihm muß dann eben in einer besonderen Disk-Datei mitgeteilt werden, wo Text (DEFM) oder Bytes (DEFB) bzw. Words (DEFW) zu definieren sind. Das ist ein ASCII-File, das man z. B. mit TSCRIPS schreiben kann. Dazu muß mit einer geeigneten Utility (z. B. FED) herausgefunden werden, wie solche Adressen lauten. Zu Fuß geht es auch: Am Record-Header kann man die Ladeadresse des ersten Bytes eines Records ablesen. Wenn man bis dorthin weiterzählt, wo der Programmcode aufhört, und Daten beginnen, kommt man auch zum Ziel.

Wie auch immer, es ist ziemlich aufwendig: FED anschmeißen, Adressen ermitteln, aufschreiben, TSCRIPS laden, Adressen eintippen. Das hat mich bei SYS4 (endlose Fehlermeldungen des DOS) fast eine Stunde gekostet. Danach tat ich den heiligen Schwur, solchen Stumpfsinn nur noch vom Computer erledigen zu lassen und schrieb ein Programm:

SCREENER lädt ähnlich wie SUPERZAP das zu disassemblierende Maschinenprogramm sektorweise in den Speicher. Ein Cursor erscheint, mit dem man beliebig im Sektor herumgurken kann. Wo DEFB, DEFW oder DEFM vermutet wird, genügt ein Tastendruck, um die jeweilige Ladeadresse automatisch in einen Puffer schreiben zu lassen. Nach getaner Arbeit kann man, wiederum mit einer Taste, die Abspeicherung des Screening-Textes auf Diskette veranlassen. Für die genaue Bedienung von SCREENER möchte ich auf die nachfolgende Kurzanleitung verweisen.

Wer das Programm haben möchte, kann sich an unseren Club-Diskothekar Werner Förster wenden. Er bekommt dann den Sourcecode, der mit ZEUS zu assemblieren ist. Wer ein G3 oder G3s besitzt, findet eine mit Sicherheit fehlerfreie Version vor. Beim Model 1 und dem Genie 1/2 bin ich ziemlich sicher, daß ebenfalls alles glattgehen müßte (ich besitze ja nur das G3s zum Austesten). Die Version für das Model 3/4 hat leider noch einen Fehler, an dem aber Hartmut Obermann z. Zt. arbeitet. Für Rückmeldungen über etwaige Fehler in der Version für das Model 1 (Genie 1/2) und für Anregungen zu einem weiteren Ausbau wäre ich dankbar.

Die Diskette von Werner enhält nur den Quellcode. Die Kurzanleitung erscheint nur hier im Info, weil mein wild gepatchtes TSCRIPS gewisse auf meinen Drucker zugeschnittene Zeichen erzeugt, die bei einer anderen TSCRIPS-Version und einem anderen Drucker möglicherweise Ärger verursachen können.

Arnulf Sopp

SCREENER ist ein Programm, mit dem bequem Screening-Files für DSMBLR erzeugt werden können. Es wird nur als Assembler-Quellcode für ZEUS weitergegeben, der für den eigenen Computer neu assembliert werden muß. Zur Anpassung an den Rechnertyp bietet der Sourcecode gleich am Anfang eine Hilfe an:

```
;Labels zur Anpassung des Programms an den eigenen Computer
m1      EQU     0       ;Model 1 usw.: 1    andere Rechner: 0
m3      EQU     0       ;Model 3:      1    andere Rechner: 0
g3      EQU     1       ;Genie 3/3s:   1    andere Rechner: 0

;Sicherung gegen falsche Label-Zuteilung (mehrfach 1 oder 3X 0)

model EQU       m1+m3+g3-1      ;das Resultat muß 0 ergeben
```

Das Label, das den eigenen Zielcomputer bezeichnet, muß zu 1 definiert werden, die beiden anderen müssen 0 lauten. Wird hierbei ein Fehler gemacht, kehrt das Programm automatisch beim Programmstart mit einer Fehlermeldung ins DOS zurück, noch bevor typspezifische Änderungen im DOS vorgenommen worden sind. Das Programm ist dann nach der Fehlerkorrektur neu zu assemblieren.

<u>Initialisierung des Screening-Textpuffers:</u>
Nach dem Programmaufruf wird der User gefragt, ob ein noch im Puffer stehender Text weiter bearbeitet werden, ein angefangener Text von Disk geladen oder der Speicher neu beschrieben werden soll.
<D> lädt einen Text von der Diskette,
<R> restauriert den alten Text,
<N> legt den Puffer neu an.
Wenn nach <A> kein Screening-Text im Puffer gefunden wird, wird dieser gelöscht (wie bei <N>).

<u>Laden des Zielprogramms:</u>
Danach wird der Benutzer nach dem Namen des zu disassemblierenden Maschinenprogramms gefragt. Die Antwort kann alle für Filespecifications üblichen Zusätze enthalten. Lautet die Extension "CMD", dann kann sie weggelassen werden (wird automatisch angehängt). Danach wird der erste Sektor des Zielprogramms angezeigt. Ein Cursor erscheint, der beim Genie 3/3s das jeweilige Zeichen invertiert, bei anderen Modellen einen Graphikblock erzeugt.

<u>Mögliche Eingaben bei der Sektoranzeige:</u>
Die Pfeiltasten steuern den Cursor. Mit Shift setzen sie ihn *ganz* nach oben, unten, links oder rechts.
Wenn der Cursor auf einem Byte steht, wo der Benutzer keinen Programmcode, sondern andere Daten vermutet, können Screening-Daten zur Generierung von Pseudo-Ops in DSMBLR erzeugt werden:
<M> erzeugt '$xxxx' für DEFM,
<W> " '#xxxx' für DEFW,
<B> " 'xxxx' für DEFB,
</> " '-xxxx' für "bis" (DEFB usw. "von - bis").
(xxxx ist in Hex-ASCII die Ladeadresse des markierten Bytes.)
Die errechneten Adressen werden unten links zur Kontrolle angezeigt. Die "bis"-Adresse wird nicht errechnet 1. am Textanfang (das wäre unsinnig), und 2. wenn das letzte Zeichen bereits ein "bis" war (Fehler).
Wenn der Cursor bei der Eingabe eines dieser Zeichen auf einem Record-Header oder in einem Kommentar-Record steht (kein ladbarer Code), ertönt beim Genie 3/3s ein BEEP, bei anderen Modellen unterbleibt lediglich die Anzeige der Adressen.

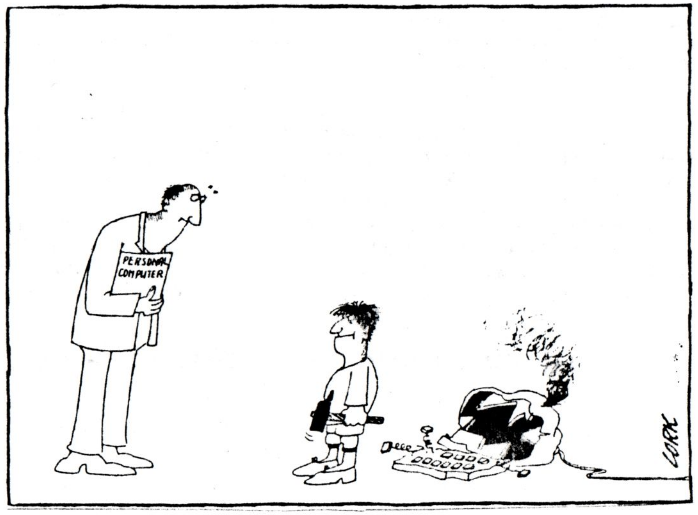
<u>Edition weiterer Sektoren:</u>
<;> (Plus ohne Shift) lädt den nächsten Sektor.
<+> (Plus mit Shift) " " letzten " .
<-> " " vorigen " .
<=> (Minus mit Shift) " " ersten " .
In allen Sektoren können in gleicher Weise Screening-Daten erzeugt werden.

<u>Edition des Screening-Textes:</u>
<A> löscht den Bildschirm und zeigt den bisher erzeugten ASCII-Text an. Korrekturen sind möglich:
<M> initialisiert etwaige Modifikationen und zeigt einen Cursor an (invertiertes Zeichen bzw. Graphikblock, s. o.).
Jedes eingegebene Zeichen ersetzt das Zeichen unter dem Cursor.
<CLEAR> löscht den Textpuffer von dem dem Cursor vorausgehenden Komma an (Trennzeichen der Hex-ASCII-Adressen). Der Cursor muß dabei nicht genau auf dem Komma stehen.

<BREAK> kehrt zur Anzeige des aktuellen Sektors zurück.

<u>Verlassen des Programms:</u>
<BREAK> kehrt jederzeit in die Befehlsebene vor dem Programmstart zurück (DOS oder BASIC).
<X> schreibt den ASCII-Text auf Diskette und kehrt in die vorige Befehlsebene zurück. Dabei wird der Screening-Text mit einem Punkt abgeschlossen, der Rest des aktuellen 256-Byte-Blocks wird ausgenullt.
Vor dem Saven des Textes wird der Benutzer nach dem Namen des Screening-Files gefragt. Alle üblichen Angaben sind möglich. Wenn die Extension "TXT" lauten soll, wird sie automatisch angehängt und kann bei der Eingabe weggelassen werden.

ZEUS, DSMBLR und SUPERZAP sind eingetragene Warenzeichen.
Für etwaige Schäden, die wider Erwarten beim Betrieb von SCREENER auftreten sollten, kann keine Haftung übernommen werden.
Die Weitergabe von SCREENER ist ausdrücklich erwünscht. Es wäre nett, wenn dabei der Name des Autors nicht verschwiegen würde.



Kleine Hexen ohne Aufwand

Anscheinend sehe ich gelegentlich den Wald vor lauter Bäumen nicht. In meinem Artikel "Kleine Hexen bevorzugt" rede ich noch davon, daß DSMBLR/CMD die Hexziffern auf dieselbe düstere Weise erzeugt wie das DOS - aber ich komme nicht auf die Idee, das DOS einfach zu CALLen. Wer keine Arbeit hat, beschafft sich eben welche. Erinnern wir uns: Es ist lästig, bei der Arbeit mit ZEUS für Labels, die von DSMBLR erzeugt wurden, immer die Shift-Taste zu drücken, weil die Hexziffern A-F als Großbuchstaben erscheinen.

Nun ist das damals vorgestellte Konversionsprogramm aber überflüssig geworden. Wenn man die Hex-Routine des DOS nämlich CALLt, erübrigen sich einige Bytes aus DSMBLR. Wie sich zeigte, sind es exakt so viele, wie erforderlich sind, um bei den Hexziffern A-F Kleinbuchstaben erzeugen zu lassen. Der dazu nötige kleine Zusatz paßt präzise in die frei werdende Lücke, so daß ein Extrafile nicht mehr nötig ist. Ein Detail davon bedarf der Erklärung: Das DOS wähnt HL als Zeiger auf den Hex-Puffer, DSMBLR unterhält jedoch den Puffer dort, wo DE hinzeigt. Daher die beiden Befehle EX DE,HL.

Listing 1 zeigt die Originalroutine in DSMBLR. Bei den Kommentaren habe ich boshafterweise einfach bei H. Grosser in seinem "DOS-Buch" abgeschrieben. Mir scheint nämlich, daß auch er den nahezu kabbalistischen Algorithmus nicht durchschaut hat, sonst hätte er sich ausführlicher gefaßt. In Listing 2 erscheint die Modifikation, um Kleinbuchstaben für die Hexziffern A-F zu erhalten. Darunter stehen schließlich noch die ersten drei Zeilen eines Hex-Dumps mit SUPERZAP aus dem gepatchten Sektor 7 von DSMBLR. Die Änderungen sind unterstrichen.

```
                ;Listing 1:     Originalversion in DSMBLR/CMD

5AFA                    ORG     5afah   ;Kommentare bei Grosser ("DOS-Buch", ab 4071h):
5AFA  E60F              AND     0fh     ;Bits 7...4 löschen
5AFC  C690              ADD     A,90h   ;in
5AFE  27                DAA             ;ASCII-
5AFF  CE40              ADC     A,40h   ;Code
5B01  27                DAA             ;umformen
5B02  12                LD      (DE),A  ;und ausgeben
5B03  13                INC     DE      ;Zeiger erhöhen
5B04  C9                RET

                ;Listing 2:     geänderte Version für kleine Hexziffern

5AFA                    ORG     5afah
                ;ab hier verändert:
5AFA  EB                EX      DE,HL   ;tauschen, weil die DOS-Routine nach (HL) lädt
5AFB  CD7140            CALL    4071h   ;das DOS tut dasselbe wie DSMBLR im Original
5AFE  EB                EX      DE,HL   ;Register (Zeiger) zurücktauschen
5AFF  1B                DEC     DE      ;Zeiger auf die Hexziffer zurückstellen
5B00  F620              OR      20h     ;Klein- aus Großbuchst. machen (0-9 unveränd.)
                ;ab hier der unveränderte Code:
5B02  12                LD      (DE),A
5B03  13                INC     DE
5B04  C9                RET

FRS 00  3E6D 1213 1803 CD05 5BCD F15A 2B7E 0100   >m......A..Z+ß..
7   10  F25A 0F0F 0F0F CDFA 5A7E EBCD 7140 EB1B   .Z......Zß..q@+.
7H  20  F620 1213 C97E FEA0 D83E 3012 13C9 CDF7   . ...ß...>0.....
```


Arnulf Sopp

Die Pratze läßt das Mausen nicht

Es mochte zunächst nach Spielkram gerochen haben, als Hartmut Obermaus die Murmeltastatur für Analphabeten (man nennt sie Maus) für die TRS-80-Kompatiblen nutzbar machte. Heute fiel mir jedoch eine weitere Anwendung für TSCRIPS ein, die das Teil richtig sinnvoll erscheinen läßt.

Beim Trennen von Texten hat man die Möglichkeit, mit "-" einen Trennungsstrich einzufügen, mit ENTER das Wort unter dem Cursor ungetrennt zu lassen, mit den waagerechten Pfeiltasten den Cursor auf eine trennbare Stelle zu setzen oder mit CLEAR den Vorgang abzubrechen. Andere Tasten werden nicht beachtet. Es fällt auf, daß bis auf den Minus- oder Trennstrich alle Zeichen in 3840h memory mapped sind. Das ist die Tastaturzeile, die die Maus bedient.

Wenn man nun außer dem Minuszeichen noch BREAK und/oder SPACE als Trennanweisung zuläßt, ist alles mit den Tasten in 3840h zu erledigen, also eine ehrenvolle Aufgabe für die Maus (oder auch den Joystick, der die gleichen Vorteile hat). Nun braucht man nicht mehr mit den Augen vom Bildschirm zur Tastatur zu wandern, aus Angst, blind könnte man sich vertippen.

Auch ohne Maus bietet diese Änderung in TSCRIPS einen Vorteil. Auf der Tastatur des alten Genie 1 liegt die BREAK-Taste direkt neben dem Minuszeichen. Wenn man es nun doch blind versucht, darf man getrost danebenhauen; man trifft dennoch eine Taste, mit der das Wort wunschgemäß getrennt wird.

Das klingt alles recht nett. Da der Joystick und die Maus von Haus aus (es lebe der Endreim!) aber nur eine Auslösetaste haben, muß auf jeden Fall gebastelt werden. Wie das geht, ohne das Teil kleinzukriegen, wurde in meinem letzten Beitrag zu diesem Thema beschrieben.

Zu der Modifikation in TSCRIPS: An der Adresse 6517h steht ein CALL-Befehl, mit dem die Tastatureingabe abgeholt wird. In dem aufgerufenen Unterprogramm wird der Tastencode gleichzeitig ggf. umgewandelt und an der Stelle (IY+1) abgelegt. Z. Zt. zeigt IY auf 7C2Fh, also lautet der Puffer für den Tastencode 7C30h.

Die CALL-Adresse kann problemlos auf eine Umleitung umgelenkt werden, wo ein Check auf BREAK und SPACE stattfindet (Label devia im Listing). Die Umleitung mit ihren 16 Bytes paßt locker in einen überflüssig gewordenen Text, der mit Cassettenoperationen zu tun hat (Adresse 7925h).

Dort wird nun zunächst auf 1Dh geprüft. Das ist der Code, den TSCRIPS aus der BREAK-Taste erzeugt. War es BREAK, dann geht es beim Label hyphan weiter: Die Minustaste wird simuliert. Dasselbe passiert, wenn Blank gedrückt wurde. Wenn es eine andere Taste war, geht es unverrichteter Dinge zurück.

Vor dem Listing des kleinen Programms möchte ich noch vorstellen, wie man an die hierzu notwendigen Informationen kommen kann, ohne eine kommentierte Source von TSCRIPS (oder irgendeines anderen Programms) zu besitzen. Der Leser wird es möglicherweise brauchen, denn es sind viele Versionen von TSCRIPS im Umlauf (meine heißt 5.4).

Wenn das Programm im Speicher steht, sucht man mit Debug nach einem signifikanten Code. In diesem Falle lautete er CP '-' (FE-2D, vergleiche den Akku-Inhalt mit dem Trennstrich). Es ist durchaus möglich, daß der Code mehrfach in einem Programm vorkommt. Deshalb ist es ratsam, alle diese Stellen aufzusuchen und in ihrer Umgebung nachzuforschen, ob der Rest für die gesuchte Routine plausible Befehle enthält.

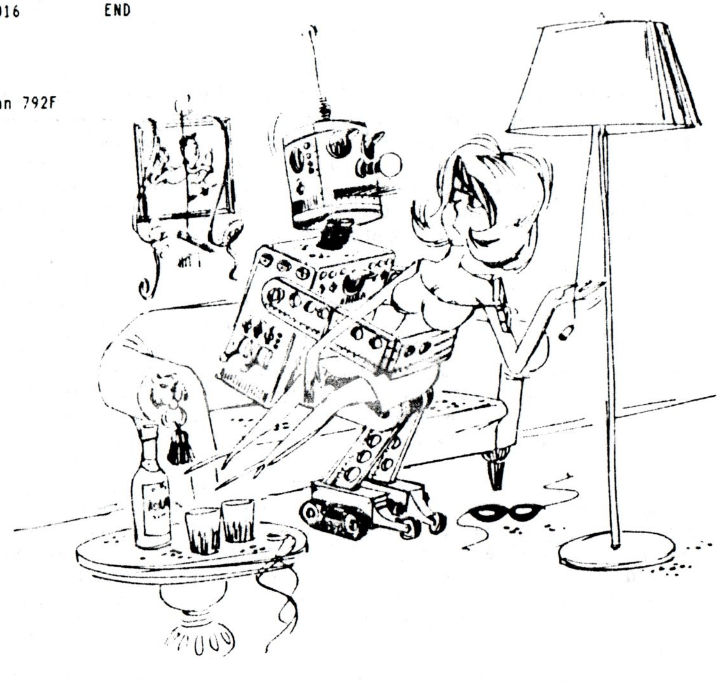
Wenn sich der Verdacht hinreichend erhärtet hat, daß man fündig wurde, wird sicherheitshalber weitergetestet: An einer hundsgemeinen Stelle, z. B.

beim Trennstrich-Vergleich, wird F7h gepatcht. Das ist der Befehl RST 30h (Aufruf von Debug). Wenn man nun das Zielprogramm abfährt, darf sich in unserem Falle nur dann Debug melden, wenn beim Trennen eine Taste gedrückt wurde. In der Anzeige des Monitors können wir dann ablesen, was gerade im Akku steht. So läßt sich auch herausfinden, was für einen Code TSCRIPS aus BREAK macht.

Der letzte Akt sind dann nur noch der Entwurf einer Modifikation und die Überlegung, an welcher Stelle im Programm man das Kuckucksei legen will.

Arnulf Sopp

```
                      00001 ;           TSCRIPS-Patch zur Silbentrennung mit einer umgebauten Maus
                      00002
6518                  00003        ORG    6518h      ;dort CALL-Adresse 8446h (Tastatur lesen)
6518  2579            00004        DW     devia      ;jetzt CALL auf die Umleitung
                      00005
7925                  00006        ORG    7925h      ;ehemaliger Cassetten-Text (überflüssig)
7925  CD4684          00007 devia  CALL   8446h      ;überschriebenen Befehl nachholen: Taste
7928  FE1D            00008        CP     1dh        ;wurde BREAK gedrückt?
792A  2803            00009        JR     Z,hyphan   ;Trennung erzeugen, falls ja
792C  FE20            00010        CP     ' '        ;Blank gedrückt?
792E  C0              00011        RET    NZ         ;zurück, falls andere Taste
792F  3E2D            00012 hyphan LD     A,'-'      ;      in Trennungsstrich umwandeln
7931  FD7701          00013        LD     (IY+1),A   ;in den Puffer
7934  C9              00014        RET               ;erledigt
                      00015
0000                  00016        END

00000 Fehler

devia  7925     hyphan 792F
```



»ICH HABE IHNEN DOCH GLEICH GESAGT, DASS ICH ECHT UND NICHT MASKIERT BIN!«

Nochmals: Vergleich zweier Disk-Dateien

In der 22. Ausgabe unseres Kirchenblättchens habe ich mich über ein Programm verbreitet, das zwei Files auf Disketten miteinander vergleicht und die Unterschiede ausgibt. Im harten Alltag hat sich dieses Programm als zuwenig komfortabel und nicht aussagekräftig genug erwiesen. Ich möchte euch nun nicht damit langweilen, wie die Änderungen gegenüber der alten Version arbeiten. Wer sich mit Assembler auskennt, ersieht das eh' aus den Kommentaren. Wer nicht, blättert sowieso weiter. Nur die Vorteile für die Praxis seien aufgezählt:

In der Anzeige erscheinen nun nicht mehr nur die Angaben über die dateirelative Position einer Abweichung sowie die Abweichung selbst (plus das nächste Byte) im zuerst genannten File. Jetzt werden die beiden Bytes des Unterschieds aus beiden Dateien angezeigt. Zusätzlich erscheint deren ASCII-Darstellung, denn es könnte sich um Text handeln. Dabei werden solche Bytes durch einen Punkt ersetzt, die nicht anzeigbar/ausdruckbar sind. Es sind nach unten die Steuerzeichen unterhalb ASCII 20h (Blank). Die Obergrenze ergibt sich aus der SYSTEM-Option AX=n. Dabei bezeichnet n den höchsten für den Drucker noch darstellbaren Wert.

Spaßeshalber habe ich in der Source die beiden ASCII-Puffer (s. Listing, Labels ascbuf1 und ascbuf2) anstatt mit 'AB' mit '@@' ausgefüllt und diese Version in die Datei COMP/CMD assembliert. Beim Aufruf des Programms mit der Syntax COMPARE,COMPARE/CMD,COMP/CMD kam folgende Anzeige:

```
rel. Skt. / rel. Byte : Abweich. ==> 0001/35: 4142 AB - 4040 @@
rel. Skt. / rel. Byte : Abweich. ==> 0001/3F: 4142 AB - 4040 @@
```

Ein weiterer Komfort wurde eingebaut, denn die Anzeige erfolgt bei diesem Maschinenspracheprogramm natürlich rasend schnell. Solange die SPACE-Taste niedergehalten wird, passiert weiter nichts. So kann man zwischendurch stoppen und sich schon einen ersten Reim auf die Anzeige machen. Das ist aber nur von Bedeutung, wenn wegen zahlreicher Abweichungen der Bildschirm das Scrollen anfängt. Im obigen Beispiel wäre dieses Feature unnötig gewesen. Es erschienen nur diese beiden Zeilen.

Und schließlich kann mittendrin mit der BREAK-Taste abgebrochen werden. Das ist z. B. dann sinnvoll, wenn sich ins Vergleichsprogramm nur ein einziges überzähliges Byte eingeschlichen hat. Dann verschiebt sich der ganze Rest nach hinten. Deshalb gibt es von dort an nur noch Abweichungen, die aber völlig uninteressant wären.

Auch die Option einer Druckerausgabe ist nun anders gelöst. Der Bildschirm wird immer beliefert, auch wenn *zusätzlich* die Anzeige auf dem Drucker gewünscht wurde. ML-Leckermäuler mögen sich an dem Trick delektieren, mit dem das passiert.

Arnulf Sopp

```
               ;       Utility zum Vergleich zweier Disk-Dateien

5200                   ORG     5200h

               ;Abfrage, ob die Druckerausgabe erwünscht ist, gegf. Anpassung
5200  E5       start   PUSH    HL              ;Befehlszeiger retten
5201  21E352           LD      HL,dvcrequ      ;Frage nach Ausgabekanal
5204  CD6744           CALL    4467h           ;anzeigen
5207  CD4900           CALL    0049h           ;Tastatur abfragen
520A  CDB545           CALL    45b5h           ;evtl. Klein- zu Großbuchstaben machen
520D  FE4A             CP      'J'             ;'J' für 'ja' eingegeben?
520F  2005             JR      NZ,open         ;falls anderes Zeichen ( = 'nein')
5211  3ECD             LD      A,0cdh          ;CALL-Opc.: LD HL,446A wird zu  CALL 446A
5213  32AB52           LD      (device),A      ;den CALL-Befehl patchen

               ;für beide zu vergleichenden Programme einen FCB einrichten
5216  E1       open    POP     HL              ;Befehlszeiger
5217  CDD54C           CALL    4cd5h           ;Trennzeichen prüfen
521A  113A53           LD      DE,fcb1         ;FCB für das erste der beiden Programme
521D  CD1C44           CALL    441ch           ;1. Dateinamen in den FCB übertragen
5220  3E2F     error1  LD      A,2fh           ;Fehlercode 'schlechte Parameter'
5222  C20944   error2  JP      NZ,4409h        ;raus mit Fehl., falls z. B. ungült. Name
5225  CDD54C           CALL    4cd5h           ;nächstes Trennzeichen prüfen
5228  115A53           LD      DE,fcb2         ;FCB für das zweite Programm
522B  CD1C44           CALL    441ch           ;2. Dateinamen in den FCB übertragen
522E  20F0             JR      NZ,error1       ;falls ein Fehler aufgetreten ist (s. o.)
5230  210055           LD      HL,buffer2      ;Sektorpuffer für das 2. Programm
5233  45               LD      B,L             ;B = 00, LRL = 256
5234  CD2444           CALL    4424h           ;den 2. FCB eröffnen
5237  20E9             JR      NZ,error2       ;falls ein Fehler auftrat
5239  113A53   -       LD      DE,fcb1         ;FCB für das 2. Programm
523C  210054           LD      HL,buffer1      ;dessen Sektorpuffer
523F  CD2444           CALL    4424h           ;den 1. FCB eröffnen
5242  20DE             JR      NZ,error2       ;falls ein Fehler auftrat
5244  48               LD      C,B             ;BC <- 0000, Sektorzähler ab Skt. 0000

               ;einen Sektor des 1. Prg. laden - Aussprung, falls Prg. zuende o. Fehler
5245  CD3644   loadlop CALL    4436h           ;einen Sektor des 1. Programms laden
5248  F5               PUSH    AF              ;Fehlercode retten
5249  FE1C             CP      1ch             ;'Ende der Datei angetroffen'?
524B  CA2D40   dosexit JP      Z,402dh         ;fertig, falls ja
524E  FE1D             CP      1dh             ;'hinter Ende der Datei'?
5250  28F9             JR      Z,dosexit       ;dto.
5252  F1               POP     AF              ;anderer Fehlercode, evtl. NZ-Flag
5253  20CD             JR      NZ,error2       ;raus, falls ein anderer Fehler auftrat

               ;einen Sektor des 2. Programms laden
5255  115A53           LD      DE,fcb2         ;FCB für das 2. Programm
5258  210055           LD      HL,buffer2      ;dessen Sektorpuffer
525B  CD3644           CALL    4436h           ;einen Sektor des 2. Programms laden
525E  20C2             JR      NZ,error2       ;falls ein Fehler auftrat

               ;ein Byte beider Sektoren vergleichen - weiter, falls gleich
5260  110054           LD      DE,buffer1      ;Sektorpuffer des 1. Programms
5263  1A       complop LD      A,(DE)          ;1 Byte daraus laden
5264  BE               CP      (HL)            ;ident. mit dem entspr. Byte des 2. Prg.?
5265  2855             JR      Z,match         ;falls ja

               ;ungleich - Abweichungen in den Puffer schreiben, Puffer anzeigen
5267  E5               PUSH    HL              ;nein, Sektorzeiger des 2. Prg. retten
5268  211F53           LD      HL,numbuf1      ;Puffer für ASCII-numerische Angaben
526B  D5               PUSH    DE              ;Sektorzeiger des 1. Programms retten
526C  50               LD      D,B             ;DE <- lfd. Nr. des aktuellen Sektors
526D  59               LD      E,C
526E  CD6340           CALL    4063h           ;in Hex-ASCII in den Puffer schreiben
5271  23               INC     HL              ;Pufferstelle dahinter
5272  D1               POP     DE              ;DE <- Sektorzeiger des 1. Programms
5273  7B               LD      A,E             ;dessen LSB als sektorrelatives Byte
5274  CD6840           CALL    4068h           ;in Hex-ASCII in den Puffer schreiben
5277  23               INC     HL              ;': ' im Zahlenpuffer überspringen
5278  23               INC     HL              ;(2 Stellen)
5279  C5               PUSH    BC              ;Sektorzähler retten
527A  012020           LD      BC,'  '         ;2 Blanks
527D  ED432A53         LD      (secbyt1),BC    ;Platz des 2. ungleichen Bytes löschen
5281  ED433453         LD      (secbyt2),BC    ;auch im Zahlenpuffer des 2. Programms
5285  012D53           LD      BC,ascbuf1      ;Pufferstelle für die ASCII-Entsprechung
5288  CDCA52           CALL    filbuff         ;abweichendes Byte in den Puffer schr.
528B  C4CA52           CALL    NZ,filbuff      ;auch nächst. B., falls noch nicht 256 B.
528E  C1               POP     BC              ;vorübergehend Sektorzähler restaurieren
528F  E1               POP     HL              ;Sektorzeiger des 2. Programms
5290  E5               PUSH    HL              ;wieder retten
5291  D5               PUSH    DE              ;den des 1. Programms retten
5292  EB               EX      DE,HL           ;DE <- Sektorzeiger des 2. Programms
5293  213253           LD      HL,numbuf2      ;Puffer für die Daten des 2. Programms
5296  C5               PUSH    BC              ;Sektorzähler retten
5297  013753           LD      BC,ascbuf2      ;ASCII-Puffer für das 2. Programm
529A  CDCA52           CALL    filbuff         ;Abweich. des 2. Programms in den Puffer
529D  C4CA52           CALL    NZ,filbuff      ;falls der Sektor noch nicht zuende ist
52A0  3E0D             LD      A,0dh           ;CR
52A2  02               LD      (BC),A          ;dort als EOS einschreiben (Textende)
52A3  C1               POP     BC              ;BC <- Sektorzähler
52A4  D1               POP     DE              ;Pufferzeiger des 1. Programms
52A5  21FA52           LD      HL,texbuff      ;Puffer des gesamten ASCII-Textes
52A8  CD6744           CALL    4467h           ;Pufferinhalt anzeigen
52AB  216A44   device  LD      HL,446ah        ;Dummy-Befehl, falls keine Druckerausg.
                                               ;sonst steht hier CALL 446A (drucken)

               ;Warteschleife, falls SPACE gedrückt, Abbruch, falls BREAK gedrückt
52AE  214038           LD      HL,3840h        ;Tastatur: Steuerzeichen
52B1  7E       wait    LD      A,(HL)          ;Tastaturstatus dieser Zeile laden
52B2  07               RLCA                    ;ist Bit 7 gesetzt? (SPACE)
52B3  38FC             JR      C,wait          ;anhalten, falls ja
52B5  FE08             CP      08h             ;ist BREAK gedrückt?
52B7  2892             JR      Z,dosexit       ;fertig, falls ja

               ;beide Sektorzeiger auf die Fortsetzung vorbereiten
52B9  E1               POP     HL              ;HL <- Sektorzeiger des 2. Programms
52BA  1D               DEC     E               ;weil später INC E folgt (Ausgleich)
52BB  6B               LD      L,E             ;beide Sektorzeiger auf dieselbe Stelle

               ;keine Abweichung gewesen oder Abw. angezeigt - Skt.-zeiger weiterstellen
52BC  1C       match   INC     E               ;eine Stelle weiter im Sektor des 1. Prg.
52BD  2C               INC     L               ;dto. 2. Programm

               ;Sektoren weiter vergleichen oder nächsten Sektor vorbereiten
52BE  20A3             JR      NZ,complop      ;falls der Sektor noch nicht zuende ist
52C0  113A53           LD      DE,fcb1         ;sonst DE <- FCB des 1. Programms
52C3  210054           LD      HL,buffer1      ;HL <- Sektorpuffer des 1. Programms
52C6  03               INC     BC              ;Sektorzähler erhöhen
52C7  C34552           JP      loadlop         ;dasselbe Spielchen mit dem nächsten Skt.

               ;abweichende Bytes eines Sektors und/oder Punkt in den Puffer schreiben
52CA  1A       filbuff LD      A,(DE)          ;aktuelles Byte eines Sektors des 1. Prg.
52CB  FE20             CP      ' '             ;Blank oder höher (ausdruckbar)
52CD  380A             JR      C,replpnt       ;falls kleiner: durch Punkt ersetzen
52CF  E5               PUSH    HL              ;retten
52D0  217043           LD      HL,4370h        ;dort höchster Druckercode abgelegt
52D3  34               INC     (HL)            ;auf 1. nicht mehr ausdruckbaren Code
```

```
52D4 BE              CP   (HL)          ;ist der Code [illegible]
52D5 35              DEC  (HL)          ;Code restaurieren
52D6 E1              POP  HL            ;dto. Zeiger
52D7 3802            JR   C,wrasc       ;unverändert ausgeben, falls kleiner
52D9 3E2E   replpnt  LD   A,'.'         ;nicht ausdruckbares Z. durch Punkt ers.
52DB 02     wrasc    LD   (BC),A        ;Zeichen oder Punkt in den ASCII-Puffer
52DC 03              INC  BC            ;nächste Pufferstelle
52DD 1A              LD   A,(DE)        ;Byte des Sektorpuffers
52DE CD6840          CALL 4068h         ;in Hex-ASCII in den Puffer schreiben
52E1 1C              INC  E             ;nächste Stelle im Sektorpuffer
52E2 C9              RET                ;zurück

52E3 44     dvcrequ  DM   'Druckerausgabe? (J/N)',0eh,0dh ;Abfrage, Cursor an, CR
52FA 72     texbuff  DM   'rel. Skt. / rel. Byte : Abweich. ==) ' ;Text
531F 78     numbuf1  DM   'xxxx/yy: zz'     ;Puffer für Angaben des 1. Bytes der Abw.
532A 7A     secbyt1  DM   'zz '         ;dto. des 2. Bytes der Abweichung
532D 41     ascbuf1  DM   'AB - '       ;dto. der ASCII-Entsprechung beider Bytes
5332 7A     numbuf2  DM   'zz'          ;dasselbe für die Abweichungen des
5334 7A     secbyt2  DM   'zz '         ;Vergleichsprogramms
5337 41     ascbuf2  DM   'AB',0dh

533A        fcb1     EQU  $             ;32 Bytes Platz für den FCB des 1. Prg.
535A        fcb2     EQU  $+20h         ;dto. FCB des 2. Programms

5400        buffer1  EQU  $+0140h&0ff00h ;Sektorpuffer für das 1. Programm
5500        buffer2  EQU  $+0240h&0ff00h ;dto. für das 2. Programm

5200                 END  start         ;dort Einsprung

00000 Fehler

ascbuf1 532D   ascbuf2 5337   buffer1 5400   buffer2 5500   comploop 5263   device  52AB
dosexit 524B   dvcrequ 52E3   error1  5220   error2  5222   fcb1     533A   fcb2    535A
filbuff 52CA   loadlop 5245   match   52BC   numbuf1 531F   numbuf2  5332   open    5216
replpnt 52D9   secbyt1 532A   secbyt2 5334   start   5200   texbuff  52FA   wait    52B1
wrasc   52DB
```

Bedingt assemblieren mit ZEUS

CP/M ist das kommende Betriebssystem für die Maschinen in unserem Club, denn die NewDOS/G-DOS-Rechner sterben irgendwann aus. Solange es aber noch ein paar letzte Mohikaner (sagte da gerade jemand "Ignoranten"?) gibt, die sich dem G-/NewDOS verschrieben haben, ist ZEUS wohl der Assembler der Wahl.

Er hat ein paar supergute Features, die mir gerade jetzt bei der Erstellung von SCREENER (evtl. auch in diesem Heft vorgestellt) sahnemäßige Dienste erwiesen haben. Eins davon steht in den nachfolgenden Listings in Zeile 857. Dort wird die ORG-Adresse dynamisch gehandhabt. Das Programm kann im Lauf der Zeit beliebig weiterwachsen. Immer wird sich der Sektorpuffer (Label buffcmd) am Anfang der nächsten freien Speicher-Page hinter dem Programm befinden, ohne daß der Programmierer dafür in irgendeiner Form noch Sorge zu tragen hätte.

Hier soll von einem anderen Vorteil die Rede sein. SCREENER soll möglichst auf allen im Club vertretenen Rechnern lauffähig sein. Nicht alle Systeme sind gleich. Deshalb muß mit Bedingungen gearbeitet werden, die sich jeweils auf ein bestimmtes Modell beziehen. Solche bedingten Befehle usw. werden bei ZEUS mit IF,cond eingeleitet und mit ENIF beendet.

Die Bedingung (condition) hat die Form eines Labels. Es wird irgendwo am Beginn des Programms deklariert. Dabei ist für ZEUS von Belang ob der Wert des Labels =0 (Bedingung nicht erfüllt) oder >0 (erfüllt) ist. In unserem Beispiel heißen die Labels m1, m3 und g3 (Zeilen 14-16). Sie erhalten gemäß den Kommentaren den Wert 0 oder 1, je nach Modell. Ein zusätzliches Label schützt den Programmierer vor Fehlern beim Umgang mit dieser Technik: Das Label model ist 0, falls kein Fehler gemacht wurde. Es ist >0, wenn man sich dabei vertat. Wir werden noch sehen, welchen Vorteil das hat.

Im Listing 1 sind Auszüge aus dem Programm wiedergegeben. Mehrere IF-Statements lassen alle Möglichkeiten zur Anpassung des Programms an den eigenen Computer offen. Der Programmierer kann das alles jedoch einfach vergessen, wenn er nur das Label m1 <u>oder</u> m3 <u>oder</u> g3 auf 1 und den Rest auf 0 gesetzt hat. Was dabei jeweils passiert, zeigen die anderen Listings. Dabei ist die Version für das Model 3/4/4p nicht wiedergegeben, weil sie sich in diesem Auszug nicht von der des Model 1 unterscheidet.

Aus Listing 2 und 3 geht hervor, was letztendlich von ZEUS als Bestandteil des CMD-Programms auf die Diskette geschrieben wird, je nach dem, für welches Modell das zutreffende Label zu 1 definiert wurde. Gegenüber dem Listing 1 fehlen einige Teile, die für das eigene Rechnermodell überflüssig sind. Es muß mit dieser Technik der bedingten Assembly also nicht unnötiger Ballast im Programm stehen, der von Fall zu Fall checkt, in welchem Computer es sich gerade befindet. Das hat aber auch einen Nachteil: Das so assemblierte Programm ist nicht mehr auf alle Modelle übertragbar. Es muß jeweils neu assembliert werden.

Schließlich zeigt Listing 4, daß mit dieser Methode auch ein wirksamer Schutz gegen Vertippen realisiert werden kann. Das Programm besteht bei einem Fehler bei der Zuordnung der Modell-Labels überhaupt nur noch aus einer Fehlermeldung (die natürlich auf allen Modellen lauffähig sein muß). Dabei half ein Trick, der wegen LIST OFF nicht in den Listings erscheint: Hinter dem Statement ORG 5200h (nicht gelistet) folgt zunächst das Bedingungs-Statement IF NOT,model. Das ganze Programm kommt also nur zustande, wenn model =0 ist (kein Fehler). Am Ende des Arbeitsprogramms, in den Listings also vor der Zeile 856, steht das abschließende ENIF.

Gewisse Labels sollen für alle Versionen gelten, auch dann, wenn die jeweilige Version mit einer Sequenz beginnt, die für andere unsinnig ist. Um die Allgemeingültigkeit zu erhalten, sollte daher dem Label eine eigene Zeile ohne Programmcode gegönnt werden. Das haben wir in diesem Beispiel bei den Labels buffcmd und start. Dabei spielt start noch eine Sonderrolle: Wenn die Bedingung model erfüllt ist (Fehler), gilt das Label dieses Namens aus der Zeile 863. Wenn nicht, dann dasjenige in Zeile 871. Ihm folgt sofort ein IF-Statement, so daß die eigene Zeile notwendig ist. Andernfalls gälte es nur für das Modell, das in IF definiert ist (hier Genie 3/3s).

Arnulf Sopp

```
;       Listing 1: Source-Text, noch nicht assembliert (alle Modelle)
;Labels zur Anpassung des Programms an den eigenen Computer:
m1      EQU     0       ;Model 1, Genie 1/2/2s: 1       andere Rechner: 0
m3      EQU     0       ;Model 3/4/4p:          1       andere Rechner: 0
g3      EQU     1       ;Genie 3/3s:            1       andere Rechner: 0
;Sicherung gegen falsche Label-Zuteilung (mehrfach 1 oder dreimal 0):
model   EQU     m1+m3+g3-1      ;das Resultat muß 0 ergeben
;Initialisierungen im Sektorpuffer, wo sie keinen Platz wegnehmen
        ORG     $+0100h&0ff00h  ;in der nächsten freien Page
buffcmd                         ;Sektorpuffer für das CMD-Programm
        IF      model           ;falls Modell-Label falsch gehandhabt
;Sicherung gegen falsche Handhabung der Labels m1, m3 und g3:
        DM      'Label für das Modell falsch zugeteilt!',0ah
        DM      'Source neu laden und assemblieren!',0dh
start   LD      HL,buffcmd      ;Alarm-Text
        JP      texttv          ;anzeigen und Ende
        ENIF
        IF      NOT,model       ;falls alles in Ordnung ist
        IF      NOT,g3          ;(das G3 hat soeben ein CLS durchgeführt)
        DB      1ch             ;HOME CURSOR
        ENIF
        DM      'SCREENER 2.1 by The HACKTORY',1fh,0dh  ;Hello-Text
start
        IF      g3              ;(nur beim Genie III/III s anwendbar)
;Einstellung des Videoformats
        LD      A,10h           ;Steuercode für 16*64 Zeichen
        CALL    outch           ;ausgeben
        LD      HL,zcmd         ;Lib-Befehl Z,R (inverse Zeichen ab 80h)
        CALL    doscal          ;Befehl ausführen
;Bei anderen TRS-80-Kompatiblen muß gegf. die Einstellung auf 16*64 Z.
;und inverse Zeichen ab ASCII 80h auf andere Weise gelöst werden, sofern
;sie nicht ohnehin überflüssig ist (TRS-80 (R) M1, Genie (R) I/II).
;Dieses Programm entstand auf einem Genie IIIs. Für das Genie III können
;noch Änderungen erforderlich sein.
        ENIF
;Einsprung für alle Modelle gemeinsam:
        LD      HL,buffcmd      ;Hello-Text
        CALL    texttv          ;anzeigen
        JP      restart         ;Hauptschleife anspringen
        ENIF
        END     start           ;dort Einsprung beim Programmaufruf
```



```
                ;       Listing 2: Source-Text, assembliert für Model 1, Genie 1/2
                ;Labels zur Anpassung des Programms an den eigenen Computer:
0001            m1      EQU     1       ;Model 1, Genie 1/2/2s: 1       andere Rechner: 0
0000            m3      EQU     0       ;Model 3/4/4p:          1       andere Rechner: 0
0000            g3      EQU     0       ;Genie 3/3s:            1       andere Rechner: 0
                ;Sicherung gegen falsche Label-Zuteilung (mehrfach 1 oder dreimal 0):
0000            model   EQU     m1+m3+g3-1      ;das Resultat muß 0 ergeben
                ;Initialisierungen im Sektorpuffer, wo sie keinen Platz wegnehmen
5800                    ORG     $+0100h&0ff00h  ;in der nächsten freien Page
                buffcmd                         ;Sektorpuffer für das CMD-Programm
5800  1C                DB      1ch             ;HOME CURSOR
5801  53                DM      'SCREENER 2.1 by The HACKTORY',1fh,0dh  ;Hello-Text
                start
                ;Einsprung für alle Modelle gemeinsam:
581F  210058            LD      HL,buffcmd      ;Hello-Text
5822  CD6744            CALL    texttv          ;anzeigen
5825  C30052            JP      restart         ;Hauptschleife anspringen
581F                    END     start           ;dort Einsprung beim Programmaufruf

00000 Fehler
```

```
                ;       Listing 3: Source-Text, assembliert für Genie 3/3s
                ;Labels zur Anpassung des Programms an den eigenen Computer:
0000            m1      EQU     0       ;Model 1, Genie 1/2/2s: 1       andere Rechner: 0
0000            m3      EQU     0       ;Model 3/4/4p:          1       andere Rechner: 0
0001            g3      EQU     1       ;Genie 3/3s:            1       andere Rechner: 0
                ;Sicherung gegen falsche Label-Zuteilung (mehrfach 1 oder dreimal 0):
0000            model   EQU     m1+m3+g3-1      ;das Resultat muß 0 ergeben
                ;Initialisierungen im Sektorpuffer, wo sie keinen Platz wegnehmen
5800                    ORG     $+0100h&0ff00h  ;in der nächsten freien Page
                buffcmd                         ;Sektorpuffer für das CMD-Programm
5800  53                DM      'SCREENER 2.1 by The HACKTORY',1fh,0dh  ;Hello-Text
                start
                ;Einstellung des Videoformats
581E  3E10              LD      A,10h           ;Steuercode für 16*64 Zeichen
5820  CD3300            CALL    outch           ;ausgeben
5823  21D256            LD      HL,zcmd         ;Lib-Befehl Z,R (inverse Zeichen ab 80h)
5826  CD1944            CALL    doscal          ;Befehl ausführen
                ;Bei anderen TRS-80-Kompatiblen muß gegf. die Einstellung auf 16*64 Z.
                ;und inverse Zeichen ab ASCII 80h auf andere Weise gelöst werden, sofern
                ;sie nicht ohnehin überflüssig ist (TRS-80 (R) M1, Genie (R) I/II).
                ;Dieses Programm entstand auf einem Genie IIIs. Für das Genie III können
                ;noch Änderungen erforderlich sein.
                ;Einsprung für alle Modelle gemeinsam:
5829  210058            LD      HL,buffcmd      ;Hello-Text
582C  CD6744            CALL    texttv          ;anzeigen
582F  C30052            JP      restart         ;Hauptschleife anspringen
581E                    END     start           ;dort Einsprung beim Programmaufruf

00000 Fehler
```

```
                00011 ;        Listing 4: Source-Text, assembl., alle Modell-Labels = 0 (Fehler)
                00012
                00013 ;Labels zur Anpassung des Programms an den eigenen Computer:
0000            00014 m1       EQU    0          ;Model 1, Genie 1/2/2s: 1      andere Rechner: 0
0000            00015 m3       EQU    0          ;Model 3/4/4p:          1      andere Rechner: 0
0000            00016 g3       EQU    0          ;Genie 3/3s:            1      andere Rechner: 0
                00017
                00018 ;Sicherung gegen falsche Label-Zuteilung (mehrfach 1 oder dreimal 0):
FFFF            00019 model    EQU    m1+m3+g3-1        ;das Resultat muß 0 ergeben
                00020
                00856 ;Initialisierungen im Sektorpuffer, wo sie keinen Platz wegnehmen
5300            00857          ORG    $+0100h&0ff00h    ;in der nächsten freien Page
                00858 buffcmd                           ;Sektorpuffer für das CMD-Programm
                00860 ;Sicherung gegen falsche Handhabung der Labels m1, m3 und g3:
5300  4C        00861          DM     'Label für das Modell falsch zugeteilt!',0ah
5327  53        00862          DM     'Source neu laden und assemblieren!',0dh
534A  210053    00863 start    LD     HL,buffcmd        ;Alarm-Text
534D  C36744    00864          JP     texttv            ;anzeigen und Ende
                00890
534A            00895          END    start             ;dort Einsprung beim Programmaufruf

00000 Fehler
```

Wo die Tomaten fliegen

Es war eine dieser typischen Parties, auf denen die Gäste sich gegenseitig mit den ewig gleichen Geschichten langweilen und man blitzschnell ein paar Gläser in sich hineinschüttet, um den Abend leichter zu überstehen.

Mit ein paar Drinks war sogar Barbara zu ertragen, die zum hundertsten Mal eine Anekdote von »Käpten Nemo« erzählte. Das war ihr »süüüßer Hund«, den sie in ihren Erzählungen immer als eine Kreuzung zwischen Dogge und Dobermann darstellte. In Wirklichkeit war das Biest ein giftiger, überfressener Scotch-Terrier, der seinen Bauch kaum noch über die Türschwelle wuchten konnte, aber in jeden Schuh biß, der ihm zu nahe kam.

Barbara redet im Durchschnitt ununterbrochen 30 Minuten, bis sie zum erstenmal wieder Luft holt. Da ihr danach meist nicht mehr einfällt, worüber sie eben noch gesprochen hat, hilft in solchen Momenten ihr treusorgender Ehemann Wolfgang, indem er sagt: »Erzähl doch mal die Geschichte, wie Käpten Nemo bei Richters auf den Perser gekotzt hat – also das war wirklich zum Totlachen.«

Doch diesmal hatte Wolfgang seiner Frau gar nicht zugehört. Er beugte sich zu Gastgeber Jürgen herüber und redete in einer Fremdsprache, die so klang: »Da sind mir doch glatt 50 Kilobyte ins ROM gerutscht. Und jetzt konnte ich sehen, wie ich die File wieder auf die Disc bekam.«

Jürgen machte Augen, als hätte er eben die heißeste Sex-Story seines Lebens gehört, und antwortete aufgeregt: »Warum hast du nicht versucht, sie mit dem Turbo-Pascal wieder übers Back-up zu retten.«

Ich war sprachlos. Hier unterhielten sich zwei völlig normale Leute, als seien sie Datenexperten des CIA.

Ausgerechnet Jürgen sprach über Computer. Dabei ist der eingeschworene Junggeselle technisch so unbegabt, daß er sich schon auf der Suche nach dem Werkzeugkasten den Daumen quetscht.

Doch jetzt war er wie weggetreten. Er schwärmte plötzlich von »seiner Kleinen« und deutete in Richtung Schlafzimmer. Dabei strahlte er, als warte dort Bo Derek unter der Decke.

Was war los mit ihm? Jürgen begegnete zwar mindestens einmal pro Woche der »Frau fürs Leben«, aber er hätte sie nie vor uns im Schlafzimmer versteckt. Im Gegenteil, er ließ sonst keine Gelegenheit aus, seine neuesten Eroberungen stolz im verheirateten Freundeskreis vorzuführen.

Ich wartete noch einige Minuten, nachdem die beiden in besagtem Zimmer verschwunden waren, und ging dann unauffällig hinterher. Ich war immerhin so höflich, vorher anzuklopfen, aber es tat sich nichts, und so öffnete ich einfach die Tür.

Der Anblick war wirklich erschütternd: Da saßen zwei erwachsene Männer auf dem Bett und starrten auf ein Fernsehgerät. In den Händen hielten sie kleine Plastikgriffe, auf denen sie wie wild rumdrückten. Dann jagten kleine Männchen über den Bildschirm und gingen vor irgendwelchen herabstürzenden Tomaten in Deckung. Dazu machte es ununterbrochen »Pfiuuuu« und »Donkdonkdonkdonk«.

»Das ist aber nicht die Sportschau«, war das einzige, was mir in meiner Fassungslosigkeit einfiel, und die beiden zuckten zusammen wie zwei Buben, die beim Marmeladenaschen erwischt wurden.

»Nein«, sagte Jürgen und starrte wieder auf den Schirm, auf dem die Tomaten inzwischen die Oberhand gewonnen hatten, »hier siehst du den neuen XP-75 in voller Aktion.« Er deutete auf eine telefonbuchgroße Tastatur, die vor ihm auf dem Nachttisch lag und mit dem Fernseher verbunden war.

Ich wußte zwar nicht, was der neue XP-75 ist, aber ich war

→ S. 44

Bedingte Assembly – notfalls ohne IF

Im letzten Beitrag zu diesem Thema ging es darum, mit IF-Anweisungen an ZEUS ein Programm sehr universell verwendbar zu machen. Ein und derselbe Sourcecode kann z. B. für alle möglichen Computer gültig sein, wenn nur ein Bedingungs-Label entsprechend angepaßt wird.

So leistungsfähig dieses Feature ist, so umständlich ist es aber, für jede denkbar auftretende Möglichkeit eigene Sequenzen zu programmieren, die zwischen IF und ENIF stehen. Diese Methode ist auch fehlerträchtig, wenn man nicht höllisch aufpaßt. Sollte einmal ein ENIF vergessen werden, wird der ganze Rest nicht assembliert, falls die mit IF formulierte Bedingung nicht zutrifft. Die Möglichkeit der Verschachtelung von IF-ENIF-Blocks (ganz ähnlich wie in BASIC) birgt zusätzliche Risiken bei unkonzentriertem Arbeiten.

Eine Alternative eröffnen die arithmetisch-logischen Fähigkeiten von ZEUS. Je nach dem Wert eines Bedingungs-Labels können Adressen, Werte usw. *errechnet* werden. In unserem Beispielprogramm (auf dessen Sinn und Zweck ich zum Schluß eingehen möchte) ist z. B. in Zeile 10 das Label drv (Laufwerksnummer) zu 0 definiert. Das bedeutet, daß sich die nachfolgenden Operationen auf das Laufwerk 0 beziehen. Der Block mit den PDRIVE-Parametern für Lw. 0 steht an 4371h. Jeder dieser Blocks ist 10 Bytes lang. Gem. der Label-Gleichung pdrive EQU 10*drv+4371h ergäbe sich daher für Lw. 1 (drv EQU 1) die Adresse 437Bh usw. Dasselbe gilt analog für alle Werte, die sich auf die drei zuerst definierten Bedingungs-Labels wr, ds und drv beziehen.

Die Syntax solcher Operanden ist genau festgelegt. Man möge sich dazu das ZEUS-Manual ansehen. Als Faustregel kann man sich merken, daß die Elemente wie bei einem sehr dummen Taschenrechner eingegeben werden müssen. Es gibt keine Klammern, der jeweils nächste Operator verwendet immer das bisherige Zwischenergebnis, das sofort errechnet wird. Z. B. die Definition des Labels spt in Z. 13: In der Form ds+1*5 kommt je nach Laufwerkstyp 5 oder 10 dabei heraus. Stünde dort aber 5*ds+1, so ergäben sich je nach dem ein oder sechs Sektoren pro Spur, beides völlig absurde Werte.

Verzichtet man nun konsequent auf IF-Zeilen, tritt ein Nachteil dieser Methode zutage. Was zum Teufel steht denn nun eigentlich in B, wenn dieses Register in Zeile 32 mit -wr*2+spt geladen wird? Oder wohin geht der darunter stehende CALL wr*10h+4630h? Diese Art der Programmierung kann sehr unübersichtlich sein, wie man sieht. Eine gesunde Mischung aus IF und Rechnerei wird wohl das Optimum sein.

Zum Nährwert des Programms: Bei allen Computern unserer Rechnerfamilie ist die erste Diskettenspur von einfacher Dichte. Die einzige Information, die sie enthält, ist der Code von BOOT/SYS (GDOS/SYS), davon aber nur die ersten beiden Sektoren. Der ganze Rest der Spur ist vollkommen leer und steht für jede Schandtat zur Verfügung. Bei einseitigen Laufwerken sind das immerhin noch drei Sektoren, bei doppelseitigen sogar noch acht, nach denen kein Hahn kräht. Da paßt allerhand hinein, das z. B. in einem SYS-File keinen Platz mehr fände.

Dieses Programm, das hier aber nur die Form eines allgemeingültigen Schemas hat, kann dort etwas einschreiben oder auch auslesen. Dabei sollten aber die ersten beiden Sektoren, die ja unbedingt zum Booten gebraucht werden, nicht verändert werden. Deshalb wird der Sektorzähler in Z. 32 zum Schreiben nur mit zwei Sektoren weniger beladen, die Sektoradresse erhält in Z. 30 beim Schreiben den Anfangswert 0002h.

Es wäre interessant, demnächst an dieser Stelle zu lesen, welche sinnvolle Verwendung sich jemand für diesen freien Diskettenplatz ausgedacht hat.

Arnulf Sopp

```
5200                    ORG     5200h

                 ;Label wr muß für RD (lesen) bzw. WR (schreiben) definiert werden
0000             wr      EQU     0        ;Sektoren schreiben:   1      lesen:    0

                 ;Label ds muß dem eigenen Laufwerktyp angepaßt werden
0001             ds      EQU     1        ;Laufwerk doppelseitig: 1      einseitig: 0

                 ;Label drv muß den Wert des gewünschten Laufwerks haben
0000             drv     EQU     0        ;Laufw. 0: 0   Laufw. 1: 1    usw.

                 ;sonstige Labels (brauchen nicht angepaßt zu werden)
000A             spt     EQU     ds+1*5   ;doppelseitig: SPT=10          einseitig: SPT=5
0040             pds     EQU     ds*40h   ;doppelseitig: PD-Wert 40h     einseitig: 00h
4371             pdrive  EQU     10*drv+4371h    ;Adresse des PD-Parametersatzes für Lw. x
8000             buffer  EQU     8000h    ;Puffer für zu lesende/schreibende Sektoren

5200  3E00       start   LD      A,drv              ;gewünschte Laufwerksnummer
5202  CD5B44             CALL    445bh              ;Laufwerk selektieren und Motor starten
5205  217543             LD      HL,pdrive+4        ;dort SPT im PDRIVE-Block
5208  7E                 LD      A,(HL)             ;Wert laden
5209  F5                 PUSH    AF                 ;retten
520A  360A               LD      (HL),spt           ;SPT auf 5 bzw. 10 Skt. pro Spur setzen
520C  E5                 PUSH    HL                 ;Adresse im PDRIVE-Block retten
520D  217843             LD      HL,pdrive+7        ;dort TI und TD im PDRIVE-Block
5210  7E                 LD      A,(HL)             ;s. o.
5211  F5                 PUSH    AF
5212  3640               LD      (HL),pds           ;5.25", SS bzw. DS, SD, Spur ab 0 zählen
5214  E5                 PUSH    HL                 ;s. o.
5215  110000             LD      DE,wr*2            ;ab Skt. 0 (RD) bzw. 2 (WR) (diskrelativ)
5218  210080             LD      HL,buffer          ;Pufferadresse
521B  060A               LD      B,-wr*2+spt        ;Zähler der zu lesenden/schreibenden Skt.
521D  CD3046     loop    CALL    wr*10h+4630h       ;diskrel. Lese- bzw. Schreibrout. des DOS
5220  2004               JR      NZ,exit            ;falls ein Fehler auftrat
5222  24                 INC     H                  ;sonst Pufferzeiger eine Page weiter
5223  1C                 INC     E                  ;DE auf den nächsten Sektor stellen
5224  10F7               DJNZ    loop               ;bis 5 bzw. 10 Sektoren bearbeitet sind
5226  E1         exit    POP     HL                 ;ursprüngliche Werte restaurieren
5227  C1                 POP     BC                 ;Flags würden sich bei POP AF ändern
5228  70                 LD      (HL),B
5229  E1                 POP     HL
522A  C1                 POP     BC
522B  70                 LD      (HL),B
                 ;vorläufiger Notausgang (im Ernstfall RET, JP 402D oder ähnlich)
522C  F7                 RST     30h                ;zum Begucken mit BEGUG
                 ;bei NZ-Bedingung: aufgetretener Fehler in A (Debug im X-Modus fahren)

5200                     END     start

00000 Fehler

buffer 8000   drv   0000   ds   0001   exit   5226   loop   521D   pdrive 4371   pds   0040
spt    000A   start 5200   wr   0000
```


Der Teufel steckt oft im Detail!!!

Dieses Sprichwort hat mich heute, an einem wirklich schönen und sonnigen Sonntag, der eigentlich zu allem andern taugt als dazu am Computer zu sitzen, mehr als drei Stunden wertvoller Freizeit gekostet. Um anderen solche Ungemach zu ersparen, hier kurz ein paar wertvolle Tips.

Wer wie ich mit mehr als einem Rechner arbeitet (TRS 80 Model 100, Model 4p und Toshiba T2100) kommt manchmal in die Verlegenheit, Daten (sprich Dateien) zwischen den einzelnen Systemen hin und her kopieren zu wollen/müssen. So ging es mir heute mit den Programmen zum Thema HRG-Bildübertragung (siehe an anderer Stelle in diesem Info), die ich unter MS-DOS gespeichert habe, die nun aber auf CP/M (für Andreas Rychlik) und auf NewDOS (für Werner Förster) kopiert werden sollten. Das hört sich schlimmer an, als es in Wirklichkeit ist.

Das Programm HyperCross, welches es sowohl unter NewDOS (für Model 1 und 3/4/4p) als auch unter TRSDOS 6.x (für TRS 80 Model 4/4p) gibt, leistet in einem solchen Fall gute Dienste und beherrscht zudem sowohl die MS-DOS-Formate als auch das von mir benutzte CP/M-Format (Montezuma 80/DS/DD). Ohne viel Zaudern wurde also das 4p angeworfen und zunächst alle Dateien von der MS-DOS-Disk auf eine CP/M-Diskette übertragen. Probleme? Keine!

Auch die Übertragung der Dateien von MS-DOS auf NewDOS hatte mir bisher, ich habe das ja nicht zum erstenmal gemacht, keine Probleme bereitet. Jetzt aber wollte HyperCross plötzlich nicht so, wie ich wollte und schon bei der ersten Datei, die ich übertragen wollte, erhielt ich einen ominösen TRSDOS Error 11. Hätte ich zu diesem Zeitpunkt in das TRSDOS-Handbuch geschaut, mir wäre wahrscheinlich einiges erspart geblieben. So aber versuchte ich eine andere Methode, die zunächst auch anstandslos funktionierte.

Nun benutzte ich DBLCROSS, ein Programm, welches nur auf dem TRS 80 Model 4/4p unter Montezuma-CP/M 2.2 lauffähig ist, und kopierte die Dateien von der CP/M-Diskette auf eine TRSDOS 1.3 Disk, die ich dann mit NewDOS zu lesen beabsichtigte. Und diesmal klappte auch alles, jedenfalls bis zu dem Zeitpunkt, wo ich mir die erste Datei zur Bearbeitung vornehmen wollte. NewDOS behauptete hartnäckig, die von mir angegebene Datei sei nicht vorhanden, führte sich aber selbst ad absurdum, denn bei DIR zeigte es die Datei an und zuvor hatte es ja auch bei "COPY,1,0,,nfmt,cbf" die Datei(en) von der TRSDOS 1.3-Disk ordnungsgemäß kopiert.

Die Sache hat mich wirlich sehr viel Nerven gekostet und so will ich euch nun nicht länger auf die Folter spannen und die Sache aufklären. Der Teufel, der ja (siehe Überschrift) im Detail steckt, verbarg sich in meinem Fall hinter einem simplen Bindestrich. Diesen hatte ich, zur Verbesserung der Lesbarkeit der Dateinamen unter MS-DOS eingeführt und unter CP/M damit natürlich keine Probleme. HyperCross warnte mich wohl mit einem TRSDOS-Error 11 schon, DBLCROSS kopierte aber den Bindestrich lustig in das NewDOS-Direktory und kümmerte sich nicht im mindesten darum, daß NewDOS damit überhaupt nichts anfangen kann.

Und die Moral von der Geschicht: traue Deinem Betriebssystem nicht (und erst recht nicht seinem Bediener)!!!

Hartmut Obermann

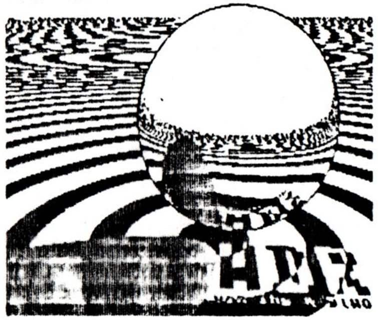
HRG – Bildübertragung

Vor mehr als zwei Jahren erschien im Clubinfo 14 auf den Seiten 24 bis 28 schon einmal ein Artikel mit dieser Überschrift. Im Grunde beschäftigte er sich auch mit genau dem gleichen Thema, nämlich der Übertragung von Bildern zwischen nichtkompatiblen Systemen bzw. über Modem/Akustikkoppler.

Das Thema haben Alexander Schmid und ich, während einer zwei Tage (die Nacht nicht zu vergessen) dauernden Session wieder aufgewärmt. Angefangen hat es damit, daß Alexander für sein Genie IIs einen Grafiktreiber unter CP/M geschrieben hat, der zu dem, den ich für das TRS 80 Model 4p verfasst habe, befehlskompatibel ist. Dadurch laufen Grafik-Programme, die ich unter CP/M in Turbo-Pascal geschrieben oder von PC's konvertiert habe, ohne weitere Änderungen auf dem IIs. Danach ging es darum Grafikdateien, die Alexander auf seinem IIs bzw. ich auf meinem 4p geschrieben haben, auf dem jeweils anderen Rechner zu laden und anzuzeigen. Auch dieses Problem wurde recht schnell gelöst. Ebenfalls ohne Probleme können wir inzwischen Grafikdateien der Atari ST-Serie (kopiert bei Gerald Schröder) lesen (und demnächst hoffentlich auch erzeugen. Darüber aber mehr in einem weiteren Artikel.

Zu guter Letzt probierten wir das von Alexander selbst gebaute Modem aus und verirrten uns dabei u.a. in die WDR-Mailbox, womit wir wieder beim Thema wären. In dieser Mailbox gibt es nämlich eine Rubrik "Bildübertragung", die nochmals unterschieden wir in Übertragung komprimierter und nicht komprimierter Bilder. Die nicht komprimierten Bilder behandelte der oben schon erwähnte Artikel im Info 14, hier soll auf die komprimierten Bilddateien eingegangen werden.

Wenn man in der WDR-Mailbox die Rubrik "Übertragung komprimierter Bilder" anwählt, erhält man folgende Auswahl. Unter (1) findet man einen sehr ausführlichen Erklärungstext, der im Anschluß zu meinen Ausführungen abgedruckt ist. Weiterhin gibt es eine recht große Anzahl von Bildern, von denen Alexander und ich, zur Freude des Herrn Postminister und zum Schrecken meiner(s) Frau (Finanzministers), sechs Stück in unseren Besitz brachten.



Hier das Menü der Rubrik "Übertragung komprimierter Bilder" aus der WDR-Mailbox (vom 12.08.1988)

1 Erlaeuterungen
2 Testbild
3 Verschlusszeitschaltung
4 Sesamstrasse
5 *Space-Shuttle
6 *WDR - Kugel
7 *3 - D Funktion
8 Eisenbahn-Interface 1
9 " 2
10 Interface-Stueckliste
11 Joystick-Interface 1
12 Joystick-Interface 2
13 Corvette
14 *Weltkarte
15 Triangle
16 Dragons
17 Schach
18 *Papagei
19 *Snoopy
20 Morse-Interface (CC 10/86)

Hier nun zunächst der unter "1 Erläuterungen" abrufbare Text.

Bildübertragung

Den meisten wird das bisher verwendete Prinzip der Bildübertragung bereits bekannt sein :

Ziel ist es ein Grafikbild in Zeichen umzuwandeln, die ohne Informationsverlust das komplette Bild repräsentieren, und ohne teure Konvertierungssoftware anderen Rechnern zugänglich gemacht werden können oder z.B. über Telefon übertragen werden können (wofür man dann einen Akustikkoppler und ein DFÜ-Programm haben muß).

Ein Grafikbild besteht natürlich aus Punkten, die in Zeilen organisiert sind. Um in einem möglichst unproblematischen Bereich der Ascii-Tabelle (Zahlenmäßige Verschlüsselung der Bildschirmzeichen incl. Steuerzeichen) zu bleiben werden immer nur 6 Punkte in ein Zeichen zusammengefaßt. Das ist deshalb nötig, weil nur in den seltensten Fällen Zeichen mit einem Ascii-Code, der größer als 126 oder kleiner als 32 ist, unverändert übertragen werden. Man hat also pro Zeichen einen Bereich von chr$(32) bis chr$(126), also 95 Werte. Da man allerdings immer nur ganze Punkte repräsentieren kann, interessiert nur die nächst kleinere 2-er Potenz also 64 = 2 hoch 6 ==> macht 6 Punkte.
Man faßt also immer 6 horizontal nebeneinander liegende Punkte zu einer Zahl zusammen; und zwar so, daß die 6 Punkte von links nach rechts die Wertigkeiten 1,2,4,8,16 & 32 erhalten und die Wertigkeiten all derjenigen Punkte, die nicht schwarz, sind addiert werden.
Addiert man zu dieser Zahl 32 hinzu, so ist die kleinste mögliche Zahl (alle Punkte Schwarz) 32 und die größte mögliche (kein Punkt schwarz) 95. Damit bewegt man sich im Bereich von 32 bis 95; die entsprechenden Ascii-Zeichen sind also alle unproblematisch.
Reiht man nun alle Ascii-Zeichen für je 6 Punkte von links nach rechts aneinander, so erhält man eine Zeile, die genau den gleichen Informationsgehalt, wie die durch sie repräsentierte Punktreihe (Zeile von Punkten im Grafikbild), besitzt.
Alle Zeichenzeilen aus den Zeilen des Grafikbildes, von oben nach unten zusammengefaßt in einem Textdatei entsprechen dem Grafikbild bis auf den letzten Punkt, lassen sich im Gegensatz dazu allerdings ohne weiteres übertragen.
Wenn man noch ein paar zusätzliche Konventionen, zur Erleichterung der Verarbeitung solcher Dateien einhält, ist das Grundgerüst der Bildübertragung komplett :

- *Jede Zeichenzeile wird mit Crlf (chr$(13),chr$(10)) abgeschlossen.*
- *Am Ende der Datei sind zwei Zeilen, in denen keine Informationen enthalten sind, also auch keine Leerzeichen. (d.h. die Zeilen sind komplett leer)*
- *Bevor die eigentliche Information beginnt, müssen einige Informationen in Klarschrift eingetragen werden :*
  1. *Zeile : Hier muß das Wort BILD stehen. (Anfangsmarke)*
  2. *Zeile : Hier muß der Name des Rechners, auf dem das Bild erstellt wurde, stehen.*
  3. *Zeile : Hier muß die Anzahl der horizontalen Bildpunkte, die eine Zeile tatsächlich enthält (Breite des Bildes in Bildpunkten) in der Form 'Hnnn' stehen. (H steht für horizontal, um Verwechslungen auszuschließen).*
  4. *Zeile : Hier muß die Anzahl der Zeilen, die das Bild hat, in der Form 'Vnnn' stehen.*
  5. *Zeile : Hier kann das Wort KOMPRESS stehen. (nähere Erläuterungen - siehe weiter unten)*
  6. *Zeile : Diese Zeile dient evtl. späteren Erweiterungen.*
- *Ab der 7. Zeile sind dann die eigentlichen Bildinformationen zu finden.*

Ziel der Bildkompression ist es, die Größe der Text-Bild-datei (speziell bei Schaltbildern) zu verringern, ohne daß dabei Informationen verloren gehen. Dies ist möglich, indem man Bereiche, die vollständig schwarz oder weiß sind, nicht mehr Punkt für Punkt repräsentiert, sondern als Bereichsbeschreibung zusammenfaßt.

Um eine Aufwärtskompatibilität mit früheren, nicht komprimierten, Dateien zu erhalten, gelten die gleichen Konventionen mit folgenden Erweiterungen :
In der 5. Zeile der Datei muß jetzt für komprimierte Dateien das Wort KOMPRESS stehen.
Die Zeichen chr$(96) bis chr$(126) bekommen jetzt eine besondere Bedeutung :
chr$(96) - chr$(99) : Hier sind mindestens 18 schwarze Punkte nebeneinander. Um die Kompression / Dekompression programmtechnisch zu vereinfachen, wird allerdings nicht die genaue Anzahl der nebeneinander liegenden schwarzen Punkte bestimmt, sondern die Anzahl der Zeichen mit dem Ascii-Code 32, die in einer nicht komprimierten Datei in einer Zeile nebeneinander zu liegen kommen würden.
Das bedeutet, daß man 6-er Gruppen von schwarzen Punkten zählt und die Anzahl dieser in zwei aufeinander folgenden Zeichen verschlüsselt. Um möglichst viele 6-er Gruppen in zwei Zeichen darstellen zu können, wird folgendes Verfahren benutzt :
Das erste Zeichen enthält sowohl die Information, daß es sich hier um 6-er Gruppen von schwarzen Punkten handelt, als auch einen Teil der Zahleninformation zur Darstellung der Anzahl dieser 6-er Gruppen. Dazu darf das erste Zeichen nur den Ascii-Code 96 bis 99 besitzen. Die verschiedenen Codes liefern verschiedene Offsets : 96 entspricht dem Offset 0

97	"	"	"	95
98	"	"	"	190
99	"	"	"	285

Das zweite Zeichen kann dann natürlich den gesammten gültigen Bereich von chr$(32) bis chr$(126) benutzen und stellt den zweiten Teil der Zahlenrepräsentation dar. Die genaue Anzahl der 6-er Gruppen errechnet sich dann nach
Anzahl = Offset + Ascii-Code(zweites Zeichen) - 32.

chr$(100) - chr$(103) : Hier sind mindestens 18 weiße Punkte nebeneinander. Die Repräsentation dieser entspricht genau der der schwarzen Punkte, mit einem kleinen Unterschied in der Offset-tabelle : 100 entspricht dem Offset 0

101	"	"	"	95
102	"	"	"	190
103	"	"	"	285.

chr$(104) - chr$(126) : Dieser Bereich dient zur Verschlüsselung größerer leerer Bereiche. Sollten in einem Bild eine oder mehrere Zeilen vollständig aus schwarzen Punkten bestehen, so läßt sich die Anzahl derer wieder in zwei Zeichen zusammenfassen.
Da es sich hierbei um Einzelzeilen und nicht um 6-er Gruppen handelt, ist der Bereich der Ascii-Codes von 104 bis 126 auch fast 6 mal so groß wie die anderen beiden Bereiche. Die Anzahl der leeren Zeilen berechnet man genau so, wie die der 6-er Gruppen (also auch mit Hilfe von zwei Zeichen), mit dem Unterschied, daß die Offset-tabelle wesentlich größer ist. Anstatt sie explizit anzugeben, kann man sie (genau wie die anderen beiden Offset-tabellen auch) berechnen. Dabei gilt :
Offset(Zeichen) = (Ascii-Code(Zeichen) - 104) * 95.
Entsprechend ist natürlich für die anderen beiden Tabellen die 104 gegen 96 bzw. 100 auszutauschen.

Je nach der Natur des zu komprimierenden Bildes ist die Ersparnis an Speicherplatz und Zeit (und bei Übertragung via Telefon auch an Geld) nicht unerheblich. Ein weitgehend leeres Bild z.B. kann im Extremfall mit zwei Zeichen dargestellt werden. Ein Bild, das keine durchgehend weiße oder schwarze Bereiche besitzt, kann dagegen nicht komprimiert werden, d.h. es ist nach der Kompression genau so groß wie vorher.

Die im Programm-Service der DFÜ enthaltenen Programme zeigen die Technik des Abtastens und Wiedergebens sowie der Kompression und Dekompression von Bildern in Basic und Turbo-Pascal für verschiedene Rechner wohl deutlich genug, damit man sie an den eigenen Typ anpassen kann. Außerdem wird gezeigt, wie man Bilder von einem Rechner beliebigen Typs auf einem Nadeldrucker vollständig wiedergeben kann.

Wie im letzten Absatz zu lesen, enthält die Mailbox nicht nur Bilder im WDR-Format, sondern auch Programme in verschiedenen Sprachen und für verschiedene Rechner, mit denen diese Bilder auf einem Grafikbildschirm angezeigt oder einem Drucker ausgedruckt werden können. Auch hier haben wir uns bedient und zunächst zwei Turbo-PASCAL- und danach noch ein BASIC-Programm herübergezogen. Die beiden Turbo-PASCAL-Programme sind ursprünglich für einen IBM-PC geschrieben worden, ließen sich aber in weniger als fünf Minuten an den oben schon erwähnten 4p/IIs-Grafiktreiber anpassen. Das Programm IBM-BILD.PAS dient dazu, sowohl komprimierte als auch unkomprimierte Bilder einzulesen und auf dem Bildschirm sichtbar zu machen. KOMPRESS.PAS macht, der Name verrät es schon, aus einem nicht komprimierten ein komprimiertes Bild im WDR-Format. Was das bringt, will ich an einem kleinen Beispiel verdeutlichen.

Das nebenstehende Bild stammt von einem Apple II+ und hat horizontal 280 und verikal 192 Punkte. Als unkomprimiertes WDR-File ist es genau 9600 Bytes lang. Komprimiert bringt es weniger als die Hälfte "auf die Waage" und schlägt nur mit 4268 Bytes zu Buche. Im Klartext: die Übertragung des komprimierten Bildes über die Telefonleitung kostet nur halb soviel, wie die des unkomprimierten!!! Ein weiterer Vorteil besteht darin, daß bei der Darstellung komprimierter Bilder Linie- statt Point-Befehle verwendet werden können und sich so die zur Darstellung benötigte Zeit auf ca. die Hälfte reduziert (gemessen auf dem IIs von Alexander).

Auch das BASIC-Programm, welches wir der Mailbox entnommen haben, dient dem Anzeigen von WDR-Bildern, und zwar sowohl von unkomprimierten als auch von komprimierten. Es ist für den TRS 80 Model 1 und seine Verwandschaft geschrieben, welche mit einer HRG 1b von RB-Elektronik ausgerüstet sind. Die drei Programme an dieser Stelle abzudrucken halte ich nicht für sonderlich sinnvoll, da sie (genau wie die in der oben angeführten Liste, der in der WDR-Mailbox vorhanden Bilder mit einem '*' gekennzeichneten Grafiken) ab sofort auch in den Programmsammlungen des CLUB 80 (bei Werner Förster für NewDOS und bei Andreas Rychlik für CP/M) zu haben sind. Wer sich für das Thema interessiert, kann sich an diese beiden, oder auch direkt an mich, wenden. Sollte jemand weitere Bilder aus der WDR-Mailbox herauskopieren, wäre ich dankbar, wenn er diese Dateien ebenfalls der Clubdiskothek zur Verfügung stellen würde.

Hartmut Obermann

Es wurden schon viele Vorschläge für einen einheitlichen Grafik-Standard im Club gemacht, aber die Resonanz war leider immer sehr gering. Ein kleiner Schritt in diese Richtung war auch der Artikel von Gerald Schröder, wo er sowas für das Genie II und IIs gebracht hat. Auf dem Clubtreffen konnte ich nun die Grafikfähigkeiten des Model IV bewundern, für das u.a. Hartmut Obermann einen Treiber unter Turbo-Pascal hatte. Um die bestehenden Programme nicht umschreiben zu müssen, lag es für mich nahe, die Prozeduren zu übernehmen und auf mein Genie IIs zu übertragen. Natürlich ist die Grafik hier ganz anders organisiert und so war ich erstmal auf der Suche nach einem Treiber, der die ganze Rechnerei übernimmt. Geralds fast vergessenes Werk war nun wie dafür geschaffen, diese Rolle zu übernehmen. Er erwartet nur die Koordinaten der Punkte in den Registern HL und DE und die Nummer der gwünschten Funktion in A, den Rest macht er von allein. Das ganze Problem war nur noch, den Treiber irgendwo unterzubringen und vom Turbo-Pascal anspringen zu lassen. Um z.B. einen Punkt zu setzen reicht folgendes (pseudo) Programmchen:

```
PROZEDUR set(X,Y)
  Lade HL mit X;
  Lade DE mit Y;
  Lade A mit 1;
  CALL Treiber
END
```

Am besten wird sowas als INLINE direkt mit ein paar Bytes Maschinencode erschlagen.

Wer jetzt traurig ist, daß er kein IIs hat, den kann ich trösten, es gibt nämlich genau den gleichen Treiber auch für die HRG 1b, womit wohl die meisten bedient sein dürften.

Um Euch einen kurzen überblick über die Möglichkeiten zu geben, habe ich hier das Definitions-Modul der Modula-Einbindung eingebaut:

```
(*************************************************************************)
(* GRAFMOD1 macht die HRG des Genie IIs auch unter Turbo-Modula nutzbar.  *)
(*                                                                        *)
(* Folgende Prozeduren werden zur Verfuegung gestellt:                    *)
(*                                                                        *)
(* HIRES                         -> schaltet die HRG ein und den          *)
(*                                  Textschirm aus                        *)
(* TEXTMODE                      -> schaltet die HRG aus und den          *)
(*                                  Textschirm ein                        *)
(* BEIDE                         -> wie HIRES (hier nicht anders          *)
(*                                  möglich)                              *)
(* CLSH                          -> löscht den Grafikschirm               *)
(*                                                                        *)
(* INVERS                        -> invertiert den Grafikschirm           *)
(*                                                                        *)
(* PLOT(X,Y,COLOR)               -> setzt, löscht oder invertiert         *)
(*                                  einen Punkt                           *)
(* DRAW(X1,Y1,X2,Y2,COLOR)       -> zieht eine Linie von (X1,Y1) nach     *)
(*                                  (X2,Y2)                               *)
(* DOTLINE(X1,Y1,X2,Y2,COLOR,MUSTER) -> wie DRAW, aber gestrichelt        *)
(*                                                                        *)
(* CIRCLE(X,Y,R,COLOR)           -> zieht einen Kreis mit Radius R        *)
(*                                  um (X,Y)                              *)
(* ELLIPSE(X,Y,ACHSE1,ACHSE2,COLOR) -> zeichnet eine Ellipse              *)
(*                                                                        *)
(* MOVETO(X,Y)                   -> setzt aktuelle Koordinaten auf (X,Y)*)
(*                                                                        *)
(* LINETO(X,Y)                   -> zieht Linie von den letzten Koordi-   *)
(*                                  naten nach (X,Y) mit der aktuellen    *)
(*                                  Farbe                                 *)
(*                                                                        *)
(* POINT(X,Y)                    -> Abfrage, ob (X,Y) gesetzt             *)
(*                                                                        *)
(* Bedeutung von COLOR:  0 - löschen                                      *)
(*                       1 - setzen                                       *)
(*                       2 - invertieren                                  *)
(*                                                                        *)
(*             MUSTER:  BITSET (beliebiger Typ, der mit 16 Bits dargestellt *)
(*                      werden kann, z.B. INTEGER-Zahl oder direkt als Bit- *)
(*                      muster z.B. mit {1,2,3,4,5,10,11,12,13} )         *)
(*                                                                        *)
(*************************************************************************)

DEFINITION MODULE GRAFMOD1;

  TYPE set15=BITSET;

  CONST XMAX=479;        (* maximale X-Koordinate *)
        YMAX=191;        (* maximale Y-Koordinate *)

  PROCEDURE HIRES;
  PROCEDURE TEXTMODE;
  PROCEDURE BEIDE;
  PROCEDURE CLSH;
  PROCEDURE INVERS;
  PROCEDURE PLOT(X,Y,Color:INTEGER);
  PROCEDURE DRAW(X1,Y1,X2,Y2,Color:INTEGER);
  PROCEDURE DOTLINE(X1,Y1,X2,Y2,Color:INTEGER; Muster:set15);
  PROCEDURE CIRCLE(X,Y,Radius,Color:INTEGER);
  PROCEDURE ELLIPSE(X,Y,A,B,Color:INTEGER);
  PROCEDURE MOVETO(X,Y:INTEGER);
  PROCEDURE LINETO(X,Y:INTEGER);
  PROCEDURE POINT(X,Y:INTEGER):INTEGER;

END GRAFMOD1.
```


Damit wären wir eine recht ansehnliche Gemeinde (Model IV, Genie I/II und Tandy mit HRG 1b und Genie IIs...) und es würde sich endlich lohnen, sogar größere Grafik-Programme entweder anzupassen oder selber zu schreiben. Von Hartmut Obermann habe ich z.B. etliche Programme aus den CHIP-Specials, die ich, von ihm schon für sein Model IV umgebaut, ohne Probleme laufen lassen konnte (Apfelmännchen (natürlich!), Funktionenplotter...).

Wer sich für nähere Einzelheiten interessiert, soll sich bei mir melden, für Fragen stehe ich gerne zur Verfügung. Den kompletten Treiber hier abzudrucken halte ich für ziemlich witzlos, da es eine elende Tipperei wäre, die sich mit Sicherheit keiner antun würde. Außerdem treten die Probleme normalerweise sowieso erst beim Einbauen auf und sowas läßt sich dann wesentlich besser individuell lösen und nicht global im Info.

Alexander Schmid

Z-CP/M, eine deutliche Verbesserung

von Rüdiger Sörensen.

1. Einführung.

Jeder kennt CP/M, das Control Program for Microcomputers. Der Name und das Programm stammen von der Firma Digital Research. CP/M ist eines der ältesten Betriebssysteme für Micros und bestimmt eines der schlechtesten. Ursprünglich wurde es für die Intel-8080 Familie von Prozessoren entwickelt, also I8080, I8085 usw. Als Zilog mit dem Z80 auf den Markt kam, wurden auch für diese Maschinen CP/M Systeme erstellt, denn der Z80 ist ja bekanntlich Software-verträglich mit dem I8080. Unter anderem auch durch diese Portierung auf den Z80 wurde CP/M eines der meist verwendeten Betriebssysteme überhaupt. Es wurde auf einer Unzahl der verschiedensten Rechner implementiert. Heutzutage gibt es noch eine Reihe von interessanten Rechnern, die voll auf CP/M aufbauen, die meisten davon sind in den USA zu finden (Northstar, Kaypro, Osborne und verschiedene Bus-Systeme wie S100, SB180 oder in Europa ECB-Bus). Die Software-Entwicklung unter CP/M war jahrelang geradezu unglaublich, praktisch jede Hochsprache wurde unter CP/M implementiert, die ersten Spreadsheets, Datenbanken, Textverarbeiter ebenfalls. Jeder kennt Programme wie DBase, WordStar, Multiplan und Turbo-Pascal. (Anm. Wenn ich von CP/M spreche, meine ich CP/M 2.2)

Ein Zeichen für die mangelhafte Benutzerunterstützung durch CP/M ist die Tatsache, daß praktisch alle Programme eigene Shells haben und die meistgebrauchten Aufgaben wie das Kopieren, Löschen von Files, Ausgabe eines Directories usw. selbst erledigen. Darüber hinaus werden vielfach diese Ausgaben in Utilities zusammengefaßt, man denke an CPMPOW, Sweep oder so was. CP/M selbst kennt nur 6 interne Befehle : DIR, ERA, REN, USER, TYPE und SAVE. Sie besorgen das Directory-Listen, Löschen, umbennenen von Files, Wechseln der USER-Ebene (eine Art Sub-Directory) und den Dump des Speichers auf die Floppy. das ist nicht gerade viel. Dazu gibt es die CP/M Standard-Utilities PIP, STAT, LOAD, ED, ASM, SUBMIT. Damit ist unter anderem das Kopieren von Dateien möglich, Ausgabe des freien Diskettenplatzes und der Länge von Files, Laden von Intel-HEX-Files, Erstellen und Ändern von Text-Dateien und Assemblieren von I8080-Quelldateien. Bedingt durch ihr Alter leiden die Utilities an übermäßiger Einfachheit. Deshalb gab es immer einen großen Markt für Verbesserungen jeder Form. Wegen der großen Verbreitung des Betriebssystems wurden neben kommerziellen auch eine Unzahl von public domain Programmen erstellt und der Öffentlichkeit zugänglich gemacht. Die letzte Zählung in einem der größten PD-Archive ergab über 8000(!) PD-Pakete. Dabei ist die Qualität der Programme zum Teil sehr hoch. Diesen Text schreibe ich auf einem PD-Wordprocessor.

Kurz gefaßt, zu CP/M kann man sagen : das Betriebssystem taugt nichts. Die Qualität und Anzahl der verfügbaren Programme übersteigt das Angebot von anderen Betriebssystemen. (Mir tut es leid, daß ein excellentes System wie TRSDOS 6.x so den Bach runtergeht, weil kein Mensch mehr Software entwickelt.)

2. Komponenten.

CP/M besteht aus vier Bestandteilen. Der erste ist das BIOS (Basic Input/ Output-System). Das BIOS ist ein Programm, welches den einzigen(!) Hardware-abhängigen Teil des Betriebssystems darstellt. Das BIOS enthält eine Sprungleiste, meist am Anfang, die in die einzelnen Routinen verzweigt. Die Reihenfolge der Sprünge und die Antwort des BIOS sind die einzigen standartisierten Teile des BIOS. Für eine genaue Beschreibung schaut in Geralds Artikel (wie man sich ein BIOS strickt) nach. Aufgaben des BIOS sind unter anderem Ein/Ausgabe Terminal, Drucker, RS232, Diskette. Der zweite Teil ist das BDOS, das eigentliche Disketten-Betriebssystem. Hier sind die Programmteile zum Lesen und Schreiben von Files gelagert. Das BDOS unterliegt dem Copyright von DRI. Der Code wurde meines Wissens niemals veröffentlicht. Der dritte Teil ist der CCP (Console Command Processor). Das ist das Ding, mit dem sich der Benutzer herumärgert. Der CCP hat die Aufgabe, Benutzereingaben zu interpretieren und auszuführen, falls kein Fehler auftritt. Der Funktionsablauf kann so skizziert werden :

```
wiederhole
Lies Benutzereingabe
trenne Kommando und optionale Parameter
wenn eingebautes Kommando
        dann führe Kommando aus
        sonst
                  wenn
                  Kommando = ausführbare Datei
                  dann lade Datei nach 100H und
                  starte Programm bei 100H
                  sonst gib Fehlermeldung aus.
für  immer.
```

Aufgrund der wenigen eingebauten Kommandos ist einzusehen, das CP/M reichlich auf den Scheiben rumkratzt. Hinzu kommt eine andere Eigenart: Jedes Programm darf den CCP überschreiben, sogar das BDOS (wenn z.B. kein I/O stattfindet). Nach Beendigung des Programms werden deshalb der CCP und das BDOS von der Diskette neu geladen und das System neu initialisiert. Das nennt sich Warmstart (warm boot). Wenn man nur kleine Programme hat, wird das ganz schön langweilig. Weiter oben habe ich die Utility SUBMIT erwähnt. Damit ist eine Batch-Verarbeitung möglich. Dabei ist es wirklich eine Kunst, keine grauen Haare zu bekommen. SUBMIT liest eine Textdatei, expandiert ev. Parameter und schreibt die einzelnen Kommandos in umgekehrter Reihenfolge in einen File namens $$$.SUB. Nach jedem Warmstart schaut nun der CCP nach, ob diese Datei existiert. Dann wird der letzte Datensatz gelesen, gelöscht und verarbeitet. Dann wieder Warmstart. Hhhhhh... Die Speicherorganisation unter CP/M ist genial einfach. Die ersten 256 Byte sind für Systemaddressen reserviert. Ab 0100H fängt der für Benutzerprogramme freie Bereich an (die TPA). dann der CCP, der auf eine Seitengrenze (Addresse XX00H) anfangen muß. Der CCP ist 800H Byte lang. Dann das BDOS, ebenfalls 800H Byte lang. Dann das BIOS mit unbeschränkter Länge. Die Addressen des BIOS und des BDOS stehen bei 0006H bzw. 0001H. Dadurch kann jedes Programm die notwendigen Addressen selbst ermitteln. Die letzte, vierte Komponente sind die System-Utilities.

3. Z-Systeme.

In unserem Verein hat jeder einen Rechner mit Z80 oder HD64180. Im Prinzip ist also CP/M auf jedem Rechner lauffähig, der über einen ungestörten RAM-Bereich von 0000H an verfügt. Spaß macht es erst ab 64K Speicher. Wie gesagt, ist aber das Betriebssystem für I8080-Maschinen gebaut. Warum also nicht die Zusatz-Power unserer Apparate nutzen? Im BIOS wird das ohnehin gemacht, aber BDOS und CCP könnten geändert werden (d.h. neu geschrieben) um leistungsfähigere Systeme zu entwickeln. Die Idee scheint gut, und glücklicherweise haben andere sie auch gehabt. Da das BIOS allein hardwareabhängig ist, sind Z80-Versionen der anderen Bestandteile prinzipiell auf jeder Z80-Kiste lauffähig (das gilt eingeschränkt für HD64180, wegen Illegals). Also, ans Werk! Z-Systeme sind in den USA Mailboxen, Server und Händler, die Software für Z80-basierende CP/M-Systeme anbieten. Über die Uni habe ich Zugang zu den Servern. Die dort angebotene Software ist in der Regel Public Domain, neuerdings meist Freeware. Public Domain heißt, jedermann zugänglich, änderbar und auch in geänderter Fassung beliebig weiterzugeben. Freeware macht eine Einschränkung : Programme sind in den meisten Fällen mit Copyright des Autors versehen, dürfen aber für nicht-kommerzielle Zwecke in ungeänderter Version benutzt und weitergegeben werden. Es gibt BDOS-Fassungen für Z80. Diese heißen allerdings nicht BDOS (DRI!) sondern P2DOS, TURBODOS oder Z80DOS. Meistens sind die Versionen schon älter und von vielen Leuten getestet, erkannte Fehler korrigiert. Es wäre kein wesentlicher Fortschritt, die DOS-Teile nur in Z80-Code umzusetzen. Das DOS wird zwar kompakter und unwesentlich schneller (wg. Disk-Zugriffen), leistet aber doch nicht mehr. Es ist deshalb Ziel der Programmierer, neue DOS-Funktionen einzubauen. Dabei geht aber die Kompatibilität von Programmen verloren, die diese neuen Features nutzen. Es gibt in den Z-Systemen deshalb Normierungsbestrebungen. Eine beliebte Erweiterung ist das Nutzen einer Hardware-Uhr, um Dateien mit Zeit/Datum Information zu versehen. Alle mir bekannten Z80-Systeme bieten das als Option an. Weitere Möglichkeiten sind beschleunigte Disketten-Zugriffe durch Disk-Caching oder public files, das heißt, Files, die in einer bestimmten USER-Ebene stehen, sind von jeder anderen USER-Ebene aus zugänglich. Ein CP/M-Eigenart wird ebenfalls meist abgeschafft : Wenn unter CP/M eine Diskette gewechselt wird, weigert sich BDOS, auf diese zu schreiben, bis sie durch einen Warmstart angemeldet wird. Neuere Z-Dos Versionen erkennen Disk-Wechsel und melden die Disketten automatisch an. Das spart Ärger. Ein typischer Fall unter CP/M : starte Turbo-Pascal und schreibe ein größeres Programm. Dann versuche, es zu sichern. Disk full. Diskette wechseln, Versuch zu saven, und CP/M sagt : "BDOS Err on x:, disk R/O". Feierabend. Wie man sieht, kann ein Z80-DOS nicht von Nachteil sein. Was ist mit dem CCP? Hier schlägt die Stunde des Programmierers. Läppische sechs residente Kommandos... Es gibt eigentlich nur einen Ersatz für den CCP, und der heißt ZCPR3x. Un der ist der Grund für diesen Artikel. An anderer Stelle (wahrscheinlich in diesem Info) habe ich die Installation beschrieben. Hier soll eine Beschreibung der Konzepte und Möglichkeiten folgen.

4. ZCPR3x.

Für ein Programm macht es keinen Unterschied, ob es unter ZCPR3 oder CCP gstartet wird. Alle (lt. Aussage des Autors, R. Conn) käuflichen Programme laufen unter ZCPR3 genau wie unter CCP. Das ist wichtig und beruhigend. ZCPR3 fügt eine Reihe von wesentlichen Verbesserungen ein. Teilweise sind sie von Großrechner-Systemen abgeguckt (UNIX, MULTICS, VMS, NOS). Anlehnungen an MSDOS finden sich ebenfalls. Fangen wir an :

Mich hat immer geärgert, das unter CP/M zwar Directories existieren in Form von 32 USER-Ebenen, daß es aber für den Normalbenutzer sinnlos ist, damit arbeiten zu wollen. Zum Beispiel ist es sehr schwierig, Dateien von einer Ebene in eine andere zu kopieren. Programme haben in der Regel keinen Zugriff auf andere Ebenen. Man kann sich kaum einen Überblick verschaffen über den Inhalt der einzelnen Ebenen, es sei denn, man loggt sich in die Ebene ein und startet DIR. Unter ZCPR3 sind alle Ebenen gleichberechtigt (manche mehr..). Im Prompt wird neben dem Laufwerk auch die Userebene ausgegeben. Mit den ZCPR3x Utilities ist Zugriff auf andere Ebenen problemlos möglich. Weiter verwaltet ZCPR3 sogenannte Named Directories. Man kann einer Ebene einen symbolischen Namen geben, unter dem sie ansprechbar ist. Das sieht aus wie MSDOS, nur gibt es keine hierarchische Struktur. Jede Ebene kann mit Password versehen werden. Beim Einloggen in eine Ebene kann ein Initialisierungsprogramm anlaufen. CP/M sucht eine ausführbare Datei nur in der aktuellen USER-Ebene. Optional kann die Laufwerksbezeichnung mit angegeben werden. ZCPR3 unterstützt einen sog. PATH, das heißt, einen Suchpfad, auf dem nach Dateien geguckt wird, wenn sie nicht im aktuellen DU zu finden sind (DU heißt DRIVE/USER). Genau wie die Named Directories kann auch der PATH jederzeit geändert werden. Entlang dem PATH werden nicht nur Programme gesucht, sondern auch Systemdateien. ZCPR3 hat eine Multiple Command Line Buffer (MCL). Damit kann eine Kommandozeile mehrere einzelne Kommandos enthalten, die durch ";" getrennt werden. Diese Fähigkeit ist eine der wichtigsten, auf sie bauen andere Nützlichkeiten auf. Z.B. die ALIASe. Ein ALIAS ist ein .COM-File, das als solches gestartet wird und nichts weiter macht, als eine Reihe von Kommandos in den MCL zu schieben. Dabei werden Parametervariablen expandiert, wie in SUBMIT. Beispiel für eine Kommandozeile:

```
A0:BASE>asm:;zas test;zlink test;test
```

Macht folgendes :
Wechselt in DU asm:
Assembliert TEST.Z80
Linkt TEST.REL
Startet TEST.COM

Als Alias würde man das so schreiben:

```
ASM:;ZAS $1;ZLINK $1;$1
```

Dabei wird der Alias mit einem Namen versehen, z.B. MAKE.COM, und dann so aufgerufen : A0:BASE>make test Was dann entsprechend expandiert wird, d.h. $1 wird durch den String "test" ersetzt. Um ein ALIAS anzulegen, wird eine Z3-Utility benötigt. Bis jetzt bin ich noch nicht auf die residenten Kommandos eingegangen. ZCPR3 hat, wie CCP, einige eingebaute Kommandos. Das sind in der Regel weniger als 6. Dafür existiert ein Speicherbereich, der RCP (resident command package), wo sich beliebige Pakete von residenten Kommandos hinladen lassen. Meine Lieblinkskonfiguration verfügt über

die Kommandos

GO, SAVE (CPR-resident)
H, DIR, ERA, REN, CP, TYPE, LIST, SP, NOTE, ECHO (RCP-resident)

Nicht schlecht, was? LIST gibt auf den Drucker aus, H gibt Liste der Kommandos, SP gibt Diskettenplatz aus, NOTE ist Kommentar, ECHO gibt sein Argument am Terminal aus. Die ist EINE Möglichkeit, einen RCP zu bauen. Es gibt viele andere, da noch andere Kommandos zur Verfügung stehen, zum Beispiel PEEK und POKE zum Ändern des Speichers, REG zum Ändern von ZCPR3-Registern u.a. Es ist durchaus möglich, während einer Sitzung dén RCP zu ändern, indem enfach ein neuer geladen wird. Neben den RCP's gibt es die FCP's. Das sind Flow Command Packages, eine Art Programmiersprache für den Command Processor. Auch sie sind resident und können jederzeit geladen werden. In Verbindung mit ALIAS und Batch-processing eine tolle Sache. Hier mein FCP :

IF, ELSE, FI, IFQ, ZIF

Zum FCP gehört ein transientes Programm namens IF.COM. Es ist einleuchtend, daß ein IF allein noch keine Steuerung des Kommandoflusses bewirken kann, dazu gehört auch noch eine Bedingung. Wenn der FCP eine Bedingung nicht kennt (meiner kennt gar keine), dann lädt er das transiente IF und sucht dort nach. Als Bedingungen kann man z.B. prüfen ob eine Datei existiert, ob ein Parameter gleich irgendwas oder nicht besetzt ist, ob ein Fehler aufgetreten ist, ob eine Shell aktiv ist, ob der Batch-Prozessor läuft und....

In Verbindung mit dem obigen ALIAS benutze ich folgende Konstruktion:

asm:zas $1;if input;zlink $1;$1;fi

Wenn der Assembler fertig ist, wartet der ALIAS auf eine Tastatureingabe. Falls der Assembler fehlerfrei gearbeitet hat, gebe ich <RETURN> oder <T> ein, um den Flow State auf TRUE zu setzen. Dann wird der ALIAS weiter fortgesetzt. Sollte ein Fehler aufgetreten sein, gebe ich <BREAK> oder <F> ein. Dann wird der Flow State FALSE, und bis zum "fi" (=endif) wird kein Kommando mehr ausgeführt.

ZCPR3 erlaubt seinen Utilities (und inzwischen auch vielen anderen Programmen) den Zugriff auf sein sog. Environment. Darin steht unter anderem, ob ein Fehler aufgetreten ist oder ob ZEX läuft (s.u.). 80 byte dieses Environment Blocks werden von TCAP benutzt, der ZCPR3 Terminal Capability. Darin ist codiert, wie das Terminal anzusprechen ist, also Sequenzen zum Bildschirm löschen, Clear to EOL, Cursor positionieren, Belegung der Cursortasten. Das heißt, daß jede Utility davon Gebrauch machen kann, ohne daß sie speziell installiert werden müßte. (Das stimmt nicht ganz, ZCPR3-Utilities müssen schon installiert werden. Das geschieht mit dem Programm Z3INS. Es erledigt die Installation aller ZCPR3-Utilities (COM-Files, über 80 Stück) in wenigen Minuten. Dabei wird in jedes Programm die Addresse des Environment-Descriptors eingetragen, also ein 2-Byte Wert.)

4.1 ZCPR3 Utilities.

Einige davon habe ich schon angesprochen, hier eine lose Liste (es gibt über 100 ZCPR3 Utilities) :

ALIAS, VALIAS erstellen ALIAS-Scripts (.COM-Files)
XDIR gibt directory aus
MCOPY kopiert Gruppen von Files
MLOAD macht aus HEX-Dateien .COM-Files.
UNERASE restauriert gelöschte Dateien.
RENAME benennt Gruppen von Files um, RENAME *.CMD=*.COM
VLU Video orientierte Library Utility
HELP gehört zum help-System. Das kann man nicht beschreiben, das muß man gesehen haben.
DU3 Disk Utility & Editor
MU3 Memory Utility & Editor
DIFF untersucht Dateien auf Identität
DPROG erlaubt mittels einer speziellen Sprache, I/O-devices zu programmieren.
CLEANDIR räumt directories auf, entfernt unbenutzte Einträge und sortiert nach alphabetischer Reihenfolge.
FINDF sucht auf dem angegebenen ode auf allen Laufwerk(en) nach den spezifizierten Files. Wildcards sind möglich. Die Suche erstreckt sich über alle 32 DU's.

4.2 Shells.

Eine Shell ist ein Kommando-Prozessor, der dem RCP vorgeschaltet ist. Es gibt verschiedene und sie lassen sich überlagern. Eine Beschreibung im Einzelnen :

o SH.COM

Die Abkürzungs-Shell. Mit den Utilities SHVAR oder SHDEFINE wird eine Datei namens SH.VAR angelegt. In ihr befinden sich Definitionen von Abkürzungen, z.B. W = TEST.Z80. Beim Starten von SH ändert sich der PROMPT:

A0:BASE>> TYPE %W

%W wird von SH in TEST.Z80 expandiert. Shell-variablen werden durch das "%" gekennzeichnet. Sie können verschachtelt werden.

o VFILER.COM

Die Directory-Maintenance-Shell. Full-Screen löschen, umbenennen, kopieren, unsqueezen von Files und Gruppen von Files. Information über Filesize. Macro-fähig. Aus der Shell können natürlich ZCPR3-Kommandos abgesetzt werden.

o MENU.COM, VMENU.COM

Damit lassen sich einfach Menus bauen. Zum Beispiel eine TURBO-Pascal Umgebung für den Assembler/Linker/Editor. VMENU erlaubt Full-Screen Möglichkeiten.

o HSH.COM,NHSH.COM

Die History-Shell. Sie ist für den Benutzer unsichtbar und verwaltet eine Liste der 20 letzten Kommandos. Auf Druck der CursorUp Taste erscheint das letzte Kommando und kann editiert und abgeschickt werden. (DOSEDIT läßt grüßen)

Alle Shells können gleichzeitig übereinander betrieben werden (4 Stück max.)

4.2 ZEX

ZEX ist der ZCPR3-Batch-Prozessor. Er kann alles, was SUBMIT kann, und viel mehr. Seine hervorstechende Eigenschaft ist, daß er im Speicher arbeitet, und nicht auf der Disk. Dadurch wird er viel schneller und flexibler. Natürlich werden FCP-Kommandos verarbeitet und mit GOTO besteht die Möglichkeit, Schleifen bedingt zu durchlaufen. Prinzipiell werden Dateien vom Typ .SUB und vom Typ .ZEX verarztet. Bei der Installation von ZCPR3 hat man die Möglichkeit, SUBMIT wahlweise zu aktivieren. Diese Eigenschaft habe ich zugunsten von ZEX abgeschafft. Er erledigt größere Aufgaben zuverlässig und weitaus schneller als SUBMIT.

4.3 CMDRUN

Das ist der gängige Name für einen ECP (External Command Processor)
Wenn der CPR mit einem Kommando aktiviert wird, passiert folgendes:

```
Wenn Kommando Bestandteil des CPR, führe es aus.
sonst
Wenn Kommando Bestandteil des RCP, führe es aus
sonst
Wenn Kommando = .COM-File im PATH, führe es aus
sonst (ECP starten)
Wenn Kommando = ALIAS definiert in ALIAS.CMD
expandiere Parameter
     führe ALIAS aus
sonst
                    starte ERROR-Handler
```

Diese Beschreibung betrifft meinen ECP. Im Prinzip kann jedes Programm zum ECP gemacht werden, z.B. ZEX. Falls der oben beschriebene Weg also fehlschlägt, würde ZEX den PATH absuchen, ob eine .SUB oder .ZEX Datei gefunden wird, die dann zur Ausführung gelangt.

4.4 ERRORx

Wenn auch der ECP nichts findet oder gar keiner installiert ist, gelangt der ERROR-Handler zum Zuge. Im einfachsten Fall gibt er nur ein Fragezeichen aus, wie CCP, er kann aber auch zum Beispiel die Kommandozeile zum Editieren und wiederverwenden bereitstellen.

5. Zusammenfassung.

Wer bis hierher gelesen hat, ist sicher ein potentieller ZCPR3x Anwender. Wie an anderer Stelle beschrieben, ist die Installation allerdings nicht ohne Tücken. Die meisten Leute werden mit der Auto Install Version Z-DOT-COM zufrieden sein. Andere mögen sich an die Installationshinweise halten. Von den PD-Servern habe ich inzwischen eine ganze Reihe von ZCPR33-Utilities bezogen, die zur Verfügung stehen. Man frage den Diskothekar. Falls Probleme auftreten sollten, stehe ich telefonisch oder schriftlich zur Verfügung.

6. Abkürzungen.

DRI	Digital Research
CP/M	Control Program for Micro Computers
ECP	External Command Processor
RCP	Resident Command Package
FCP	Flow Command Package
NDR	Named Directories
TCAP	Terminal Capability
CCP	Console Command Processor
ZCPR	Z80 Command Processor Replacement
CPR	Command Processor Replacement
BIOS	Basic Input Output System
BDOS	Basic Disk Operating System

Ich wünsche allen, die den Mut und die Zeit haben, allen denkbaren Erfolg.

Rüdiger.

mir ganz sicher, daß meine Freunde noch vor ein paar Tagen völlig normale Leute waren; liebenswerte Menschen, die auf Parties nicht in dunklen Schlafzimmern verschwanden, sondern sich ordentlich betranken, um anschließend ausfälliges Zeug zu reden und sich mit anderen Gästen zu streiten.
»Das hier ist die Zukunft«, sagte Wolfgang und schoß noch eine Tomate ab. »Eines Tages wird jeder so ein Ding haben.«
»Computer werden so selbstverständlich sein wie ein Telefon.« Jürgen sah mich mit verklärtem Blick an. »So ein Kasten kann dir jede Arbeit abnehmen, wirklich jede. Der macht alles für dich.«
Die Kiste hatte es tatsächlich geschafft. Sie hatte meine beiden Kumpel innerhalb kürzester Zeit um den Verstand gebracht. Denn wenn einer, der denkt, er würde impotent, wenn er auch nur eine Nacht ohne Frau im Bett verbringt, plötzlich nachts vor einer Maschine hockt und auf Tomaten schießt, dann muß irgend etwas schrecklich schiefgelaufen sein.
»Hattest du Probleme mit der Kleinen aus der Fotoagentur?« fragte ich vorsichtig, und dann versöhnlich: »Weißt du, bei mir klappt's manchmal auch nicht.«
»Du hast nichts begriffen«, sagte Jürgen, und seine Stimme klang richtig mitleidig. »Du versuchst, die Dinge zu ignorieren, aber das hier ist die Zukunft«, dabei drückte er ein paarmal auf den Plastikgriff in seiner Hand, und das Männchen auf dem Bildschirm gab ein lautes »Pfiuuuu« von sich.
»Schon in den nächsten Jahren«, ereiferte sich Wolfgang, »werden unsere Arbeitsplätze alle so aussehen. Und wenn du dich dagegen sperrst, verlierst du den Anschluß. Dann ziehen die jungen Leute vorbei, daß du kein Land mehr siehst.«
Zugegeben, mein Job als Steuerberater war nicht immer besonders spannend, aber ich konnte mir nicht vorstellen, daß ich eines Tages meine Bürozeit damit verbringen würde, auf vorbeifliegende Tomaten zu schießen.

Dann machte ich einen Fehler. Ich fragte ganz harmlos: »Kann der noch was anderes?«
Die beiden fielen über mich her wie zwei eifernde Sekten-Prediger bei der Jagd auf ein neues Opfer. Sie redeten von Bit und Chip, von RAM und ROM, Basic und Pascal und schoben zwischendurch immer neue kleine Plastikscheiben in den Computer, der darauf immer neue verwirrende Grafiken vorführte. Sie sprachen ununterbrochen und gleichzeitig. Meine Fragen interessierten sie nicht, sie waren glücklich, ihr Wissen loszuwerden.
Ich verstand nur Bahnhof, aber eines war sicher: Es hatte die beiden schwer erwischt.
Ich hatte schon viel von Computern gehört und von den schrecklichen Folgen, die diese Elektronengehirne für unsere Gesellschaft haben sollen. Aber ich konnte mir bisher nichts darunter vorstellen.
Für mich kam so eine Maschine sowieso nicht in Frage. Ich besaß nichts zu Hause oder auf dem Bankkonto, das sich lohnen würde zusammenzuzählen. Ich legte auch keine Fahndungskarteien über meine Freunde an, und schon beim Wort »Daten« zuckte ich angeekelt zusammen. Datenverarbeitung war etwas für Beamte mit Sockenhaltern und glänzenden Hosenböden.
Jetzt wurde mir die Computeritis zum erstenmal in ihrer entsetzlichen Tragweite vor Augen geführt. Sie hatte meine engsten Freunde befallen, und ich mußte hilflos zusehen, wie es mit ihnen bergab ging. Denn die Nacht auf der Party war nur der Beginn eines tragischen Leidensweges.
In dieser Zeit verloren sie nicht nur ihre Frauen, den Job und den Respekt der Nachbarn. Es wurde ihnen sogar der Überziehungskredit gestrichen und der Fernseher gepfändet. Es vergingen viele Monate, bis sie sich von diesem Schock wieder erholten. Ich mußte diese Tragödie in allen Einzelheiten miterleben. Da entschloß ich mich, etwas zu tun, um zu helfen, diese Landplage des 20. Jahrhunderts zu stoppen. Auch wenn es weltfremd klingen mag, ich bekenne mich zu meinen Gefühlen und sage auch jedem, der es nicht wissen will:

»Ich hasse Computer!«

Ich hoffe, daß ich mit diesem Buch die eine oder andere Seele retten kann. Und daß es denen Kraft und Argumente gibt, die sich auch in Zukunft ihre Software nicht im Computer-Shop, sondern in der Kneipe um die Ecke holen wollen . . .

→ F.F. Die Redaktion

ZCPR33 - Installation auf Model 4 (und anderen).

Von Rüdiger Sörensen.

Seit einiger Zeit läuft auf meinem Model 4 ZCPR3 als Kommando-Prozessor unter CP/M. Das Ding war die sogenannte Z-DOT-COM Version, die sich selbst installiert. Das hat gehörige Vorteile, vor allem im Aufwand, den man betreiben muß, um das System an's Laufen zu kriegen, aber auch, was die Speicheraufteilung angeht. Z.COM kann nämlich abgeschaltet werden, und dann ist nur noch der Standard - CP/M - CCP (Command Processor, ZCPR heißt Z80-Command-Processor-Replacement) vorhanden und man verfügt wieder über den vollen Speicher. Z.COM nimmt etwa 8K in Anspruch, was in meinen Augen eine ganze Menge ist.
Aus diesem Grunde, und weil das Z.COM System so langsam bootet, habe ich eine manuelle Installation von ZCPR in's Auge gefaßt und mich für die neue Version 3.3 entschieden. Leider läuft das System noch nicht hundertprozentig, aber es läuft, und deshalb will ich versuchen zu erklären, wie's geht. Zunächst eine Beschreibung der Fähigkeiten von ZCPR33. (Nicht aus dem Auge verlieren : für die meisten Anwenderprogramme sind die Fähigkeiten nicht sichtbar, aber wer in seiner Kiste selbst rumstrickt, wird begeistert sein.)

1) Alle USER-Ebenen (bis 31) können benutzt werden und auch mit Namen versehen werden. Man kann Directories (USER-Ebenen) verstecken und mit Passwordschutz versehen. Beim Einloggen in ein neues Directory kann automatisch eine Einlogsequenz ablaufen.

2) Man kann einen Suchpfad definieren, auf dem der ZCPR nach *.COM-Files sucht.

3) Man kann Abkürzungen (ALIAS) für Kommandofolgen definieren und Parameter (bis 10 Stück) an sie übergeben.

4) Die Länge der Kommandozeile ist benutzerdefinierbar. Mehrere Kommandos können in einer Zeile abgesetzt werden.

5) Die residenten Kommandos (RCP's, das sind die, die der ZCPR direkt ausführt ohne eine .COM datei zu laden) können vom Benutzer definiert werden. Es ist auch möglich, eigene Kommandos zu entwickeln und einzubinden oder während einer Sitzung andere residente Kommandos zu laden.

6) Es gibt bedingte Ausführung von Kommandos mit den FCP's, das sind Flow Command Packages. Wieder benutzerdefinierbar. Beispiele sind IF, ELSE, FI (endif), XIF (alle IF's beenden), die Bedingungen, die überprüft werden können, sind z.B. Existenz eines Files, Fehler bei einem Programm, leerer File. Negation ist möglich.

7) ZCPR unterhält einen Buffer, in dem Informationen über das benutzte Terminal vorhanden sind. Alle ZCPR-Utilities (über 70 Stück) können darauf zugreifen und man verfügt dann über Cursor-Addressierung, Schirm löschen, inverse Darstellung und dergleichen.

8) ZEX.
Das ist für mich fast das beste. ZEX ist ein Ersatz für SUBMIT, kann aber viel mehr und läuft resident im Speicher. Dadurch ist ZEX viel schneller als SUBMIT. ZEX kann aber auch *.SUB-Files ausführen.

9) Shells. Davon gibt es verschiedene. Eine Shell ist eine Art vorgeschalteter Kommando-Prozessor, der Eingaben des Benutzers filtert und eigene Aktionen starten kann. Beispiel : VFILER, eine Shell zur Diskettenverwaltung. Oder MENU, damit kann man Menus auf ZCPR - Ebene aufbauen, zum Beispiel um eine TURBO-PASCAL Umgebung für den Lieblingsassembler zu bauen.

10) Input/Output-Packages erlauben die Umleitung von I/O auf beliebige Geräte. Viel flexibler als IO-BYTE.

Soweit ein kleiner Lobgesang. Der Nachteil soll aber auch nicht verschwiegen werden : Ein voll ausgebautes ZCPR33 System kostet ca. 5K TPA. Bei mir habe ich immer noch 48K TPA, das langt bis jetzt für alle Anwendungen. Außerdem ist die Installation schwierig, unter Umständen sogar unmöglich. Ihr könnt prüfen, ob es bei euch geht, wenn ihr die folgende Liste der Notwendigkeiten anschaut.

Hardware: Z80 - System mit mindestens einem Laufwerk (ich empfehle 2-3 80T/DS oder Harddisk)

Software:

Sourcen zu den ZCPR3/33 Programmen und Utilities. ZCPR3/33 ist ein Freeware-System, das bedeutet, man kann es zu seinem persönlichen Gebrauch benutzen, ändern und weitergeben, aber nicht verkaufen.

Sourcecode des BIOS. Das dürfte das größte Problem sein. Das BIOS muß ziemlich gründlich geändert werden, zumindest die Kaltstartroutine.

SYSGEN, MOVCPM, DDT oder ähnliches, MAC. Wenn die Utilities geändert werden sollen, auch noch M80 und L80. Wenn ZCPR33 benutzt werden soll, ZAS (oder man editiert 128 KB Quellcode für M80).

nützlich, aber nicht unbedingt notwendig : MLOAD und ein Full-Screen Editor wie PMATE oder WORDSTAR.

Als erstes muß man sich hinsetzen und planen, was für Fähigkeiten der ZCPR bekommen soll. Meine Empfehlung ist, alles einzubauen, außer vielleicht den IO-Packages. Manche Sachen sind unbedingt notwendig, andere sehr zu empfehlen. Man benötigt folgende Speichergrößen (opt = optional, rec = empfohlen, req = muß sein):

2K	RCP	resident Command Package (rec)
1.5K	IOP	IO-Package (opt)
0.5K	FCP	flow command package (if, else..) (rec)
256 B		Environment descriptor (req)
128 B		Shell Stack (rec)
80 B		Message Buffers (req)
48 B		external FCB (rec)

128 B	NDR	named directory buffer (rec)
208 B		Command Line Buffer (req)
48 B		external Stack (req)
11 B		external path (req)
1 B		Wheel Byte (opt)

Das einfachste ist, oberhalb des BIOS Platz zu machen, indem man sein System mittels MOVCPM auf 59K verkleinert. Wenn das IOP eingebaut werden soll, dann muß man darauf achten, daß es auf einer Page Boundary beginnt, also bei einer HEX-Adresse, die mit 00 endet.
Wenn der Platz besorgt wurde, alle Addressen notieren.
Dann muß das BIOS geändert werden, und zwar die Kaltstartroutine. Folgendes wird gefordert :

> Alle ZCPR-Buffer (RCP,FCP,IOP etc.) müssen beim KALTSTART initialisiert werden, d.h. mit 00H gefüllt werden.
>
> Der Command Line Buffer wird initialisiert mit Information über die Länge des Buffers, dem String 'STARTUP' und der Länge dieses Strings. Dieser String wird nach dem Booten als Kommando interpretiert und ausgeführt. STARTUP sollte ein ZCPR-ALIAS sein, der mit dem Programm LDR alle System-Segmente (ENV, IOP, FCP, RCP, NDR, Z3T) in die Buffer lädt.

Wenn das passiert ist, kann das neue System getestet werden. Dazu einen Kaltstart ausführen (booten) und nachsehen, was im Speicher bei den zugehörigen Addressen steht. Wenn alles richtig initialisiert wurde, kann man daran gehen, ZCPR selbst zu entwickeln. Dabei müssen 2 Dateien editiert werden : Z3BASE.LIB und Z3HDR.LIB. In erstere sind die Addressen der Buffer einzutragen, nebst der Addresse des CCP, die sich ja durch MOVCPM geändert hat. In Z3HDR.LIB muß man sich die features aussuchen, die direkt in den ZCPR eingebaut sein sollen. Ich empfehle SAVE, GET, GO, JUMP. Alle anderen sind leicht in die RCP's zu stecken. Dann mit MAC assemblieren und auf die Systemspuren schreiben via DDT und SYSGEN. (Bei mir funktionierte SYSGEN nicht, wenn ZCPR33 mitspielte. Deshalb mußte ich ein kleines Programm schreiben, das einen COM-File liest und auf die Sektoren der Diskette schreibt, die vom CCP belegt sind. Der CCP ist 2K lang, macht 16 Sektoren zu 128 Byte.)
Sodann muß als erstes SYS.ENV gebaut werden. Dazu ist SYSENV.ASM zu assemblierten, in einen COM-File zu verwandeln und in SYS.ENV umzubenennen. Dasselbe gilt sinngemäß für SYSRCP, SYSFCP, SYSIOP, SYSNDR und SYS.Z3T. Dann müssen die Utilities installiert werden. Jede ZCPR3-Utility enthält einen Pointer auf den Environment Descriptor oder den Descriptor selbst. Das Eintragen macht Z3INS. In eine Datei, sagen wir SYS.INS, wird eine Liste aller Utilities eingetragen und dann, falls das gültige Environment SYS.ENV heißt, das Kommando

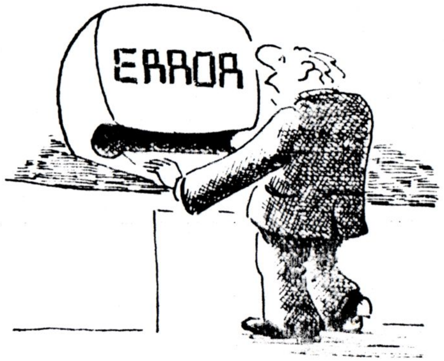
    Z3INS SYS.ENV SYS.INS

gegeben. Das Ganze dauert, bei 70 Utilities, nur ein paar Minuten. Dann kann mit LDR SYS.ENV,SYS.NDR,... das System hochgefahren werden und das ALIAS STARTUP mit der ALIAS oder VALIAS Utility erstellt werden. Das System ist fertig.
Beim Booten müssen natürlich STARTUP und die zu ladenden Packages jeweils auf der bootenden Scheibe sein. Das erste zu ladende Package ist notwendigerweise SYS.ENV. Ansonsten spielt die Reihenfolge keine Rolle.

Auf dem Model 4 gibt es leider (noch) ein Problem. Von den Utilities führen die meisten zum Absturz des Systems, obwohl sie richtig installiert worden sind. Die Ursache ist mir völlig unklar, zumal alle

Utilities einwandfrei arbeiten, wenn ich sie aus einem Debugger heraus lade und starte. Wenn einer von euch eine Idee hat, woran es liegen könnte, möge er sich bitte umgehend melden. Ich bin am Ende meiner Weisheit, habe aber die US-CP/M-Gemeinde um Rat gebeten und hoffe von dort auf Antwort.
Wenn sich einer von euch an die gewiß nicht einfache Aufgabe machen will, sein System auf ZCPR33 umzustellen, kann er ja bei mir mal fragen. Was das Model 4 angeht, kann ich ein vollständiges BIOS anbieten, inklusive deutschem Tastaturtreiber, modifizierter Kaltstartroutine und verbessertem Videotreiber (gibt Cursorposition zurück und schaltet Cursor an und aus). Ein Z80-BDOS ist ebenfalls vorhanden (Public Domain). Das Resultat ist ein System, das, neben der dramatisch erhöhten Leistungsfähigkeit; auch vollständig von Copyrights der Firma Digital Research frei ist. Außerdem liegt das BDOS und BIOS im Quellcode vor, und man kann sich seine Erweiterungen dazustricken. Bei mir, der ich noch 192K Speicher brach liegen habe, soll noch Track Buffering eingebaut werden. Vielleicht auch noch ein paar Grafikroutinen oder so was. Ich wünsche viel Spaß und Erfolg,

PS.
Das zuletzt geschilderte Problem (Utilities) ist inzwischen behoben. Es lag am Stack, der wohl entweder zu klein war, oder, weil ZCPR33 einen externen Stack im oberen Speicher erwartet, vom BIOS gelegentlich überschrieben wurde. Ich habe den Stack wieder nach ZCPR33 hinein gesetzt, denn dort war noch etwas Platz (ca. 140 Byte). Komischerweise wissen das die Utilities gar nicht, scheinbar greifen sie nicht direkt darauf zu. Für ZCPR33 gibt es eine Reihe von neuen, verbesserten Utilities. Seit ein paar Tagen "downloade" ich wie wild...



© Dietmar Große

Assembler unter CP/M

Seit CP/M auf meinem Rechner läuft, beschäftigen mich fast nur noch CP/M-Programme, und ich mußte feststellen, daß es sehr, sehr viele davon gibt. Erdrückend ist vor allem die Masse der Assembler, die für mich (und noch einige andere Assembler-Freaks im Club) besonders interessant sind. Bis jetzt konnte ich mir keinen Überblick verschaffen. Jeder schwört auf einen anderen, scheinbar denjenigen, den er als ersten zu sehen bekam oder zu dem er eine Anleitung besitzt.

Welche Assembler gibt es? Diese Frage kann ich namentlich nicht beantworten, aber ich möchte die Klassen nennen, in die sich Assembler einordnen lassen, wobei ein Assembler durchaus in mehrere Klassen fallen kann. Diese Klassen sollen Euch (und mir) helfen, einen Überblick zu bekommen und mögliche Bewertungskriterien für die Leistung eines Assemblers zu bestimmen. Ich würde mich freuen, wenn Ihr mal was darüber schreibt, welchen Ihr bevorzugt und warum (bzw. was der leistet und was nicht).

a. "Paketgröße"

Der erste Leistungsunterschied ist die Anzahl der Programme, die zu einem "Paket" gehören, mit dem dann möglichst wirkungsvoll Programme erstellt werden können.

Es sind zuerst die integrierten Editor-Assembler im Stile des ZEUS aus Newdos zu erwähnen. Der Editor zum (alleinigen) Editieren von Assembler-Programmen und der Assembler bilden eine unzertrennliche Einheit. Der Assembler erstellt gleich einen lauffähigen COM-File (entspricht den CMD-Files unter Newdos). Einen Editor-Assembler habe ich zwar unter CP/M noch nicht gefunden, aber sicherlich existiert er irgendwo. Vorteile: kurze Programme können schnell erstellt und geändert werden; die Syntax wird bereits beim Eintippen geprüft (Tippfehler EX HL,DE gleich erkannt); der Quelltext wird evtl. platzsparend mit Tokens gespeichert (wie bei BASIC oder ZEUS). Nachteile: größere Programme müssen aufgeteilt werden, wobei bei jeder Änderung das ganze Programm neu übersetzt wird (das dauert!); der Editor gefällt evtl. dem Benutzer nicht; andere Editoren oder Assembler können bei Verwendung von Tokens den Text nicht bearbeiten.

Als zweite Kategorie möchte ich die Stand-alone-Assembler nennen. Hier wird ein beliebiger ASCII-Editor (unter CP/M oft WORDSTAR) zum Schreiben des Quelltextes genommen. Dieser Text wird dann von dem Assembler in einen COM-File übersetzt. Vorteile: jeder kann seinen Lieblingseditor einsetzen, um die Quelltexte zu erzeugen; sowohl beim Editieren als auch beim Assemblieren ist mehr Speicher frei, weil das jeweils andere Programm fehlt; verschiedene Assembler verstehen evtl. die gleichen Quelltexte. Nachteile: beim Eintippen wird die Syntax nicht geprüft; die Abfolge Eintippen-Übersetzen-Fehlerkorrektur-Übersetzen dauert länger, weil zwei Programme beteiligt sind; der Quelltext belegt ohne Tokens mehr Platz.

Die dritte Form sind Assembler-Linker. Wieder wird kein Standard-Editor vorgesehen, sondern es kann ein beliebiger Editor genommen werden. Dann wird der erzeugte Text von dem Assembler in eine Zwischenform übersetzt (z.B. REL-File), die wiederum von dem Linker zu einem lauffähigen COM-File umgesetzt wird. Der Linker hat vor allem die Aufgabe, mehrere verschiedene REL-Files zu einem COM-File zusammenzufügen (linken). Dabei können die REL-Files auch von Pascal , Fortran oder sonstigen Compilern stammen. Vorteile wieder ist ein beliebiger Editor verwendbar; evtl. sind Assembler und Linker austauschbar;

größere Programme lassen sich gut aufteilen und dann schnell ändern, denn es muß meist nur ein Quellfile geändert und neu assembliert werden (das Linken geht dann relativ flott); Assembler-Routinen können mit Hochsprachen-Programmen gemischt werden. Nachteile: keine Syntaxprüfung beim Eintippen; es dauert lange, bis die Programme als COM-File vorliegen (editieren, assemblieren, Fehler korrigieren, assemblieren, linken); es wird noch ein File (der REL-File) mehr erzeugt, der natürlich Platz wegnimmt.

Es können natürlich auch andere Arten auftreten, aber das halte ich für unwahrscheinlich (nennt mir Gegenbeispiele!). Es kann z.B. sein, daß ein Editor-Assembler keine Syntax-Prüfung vornimmt, weil das bei Makros nicht geht, oder daß er keine Tokens erzeugt. Allerdings schwinden dann fast alle Vorteile dieser Art von Assemblern.

Zur Beurteilung sollte noch der Begriff "Turn-around-Zeit" erwähnt werden. Meines Wissens werden dabei immer zwei Zeiten betrachtet: Erstens die Zeit, die man braucht, um ein Programm zu erstellen (mit Fehlerfindung und -korrektur), und zweitens die Zeit, die bei einer Änderung des Programms gebraucht wird. Ich werde in der folgenden Zusammenfassung zwischen den Turn-around-Zeiten für kurze und lange Programme unterscheiden. Kurze Programme sind solche, deren Quelltext vollständig in den Hauptspeicher paßt. Hier ist vor allem die Erstellzeit wichtig. Längere Programme verteilen sich auf mehrere Quellfiles, wobei hier der Aufwand für Änderungen betrachtet werden muß.

Zusammenfassung "Paketgröße":

1. Editor-Assembler:
   Vorteile:
   - Syntaxprüfung beim Eintippen/Editieren
   - kurze Turn-around-Zeit bei kleinen Quelltexten
   - platzsparende Speicherung mit Tokens möglich
   - es wird nur ein Programm benötigt

   Nachteile:
   - der Editor ist fest
   - das Programm nimmt viel Speicherplatz weg
   - lange Turn-around-Zeiten bei großen Quelltexten
   - Tokens verhindern evtl. Austauschbarkeit von Quelltexten
2. Stand-alone-Assembler:
   Vorteile:
   - beliebiger Editor
   - mehr Speicher frei
   - Quelltexte oft austauschbar

   Nachteile:
   - keine Syntaxprüfung beim Editieren
   - Turn-around-Zeiten bei kurzen und langen Quelltexten hoch
   - mehr Speicherplatz verbraucht (keine Tokens)
   - immer zwei Programme (Editor und Assembler)
3. Assembler-Linker:
   Vorteile:
   - beliebiger Editor
   - mehr Speicher frei
   - evtl. austauschbare Assembler/Linker
   - kurze Turn-around-Zeiten bei langen Quelltexten
   - Mischung mit Hochsprache möglich

   Nachteile:
   - keine Syntaxprüfung beim Editieren
   - sehr hohe Turn-around-Zeiten bei kurzen Quelltexten
   - immer drei Programme (Editor, Assembler, Linker)
   - immer drei Files (Quelltext, Zwischenform, COM-File)

Welche Assembler sind zu empfehlen? Das kommt ganz auf den einzelnen an. Arnulf schwört auf das Editor-Assembler-Konzept des ZEUS. Ich würde es auch auf jeden Fall für Anfänger empfehlen, denn die Syntaxprüfung ist eine große Hilfe. Es kann mit BASIC verglichen werden, denn dort ist auch eine Syntaxprüfung beim Editieren möglich und Tokens werden zur Speicherung eingesetzt.

Allerdings hat ein Editor-Assembler seine Grenzen bei Makros (hier wird die Syntaxprüfung schwierig) und bei der Entwicklung größerer Programme. Dann wird eine Strukturierung und Aufteilung nötig, die sich am effektivsten mit der Assembler-Linker-Kombination zu lösen ist. Ich kenne bisher jedoch nur einen Assembler dieser Klasse (M80). Er hat ähnliche Sprachelemente wie Modula, das auch zur Entwicklung größerer Systeme hervorragend geeignet ist bzw. diese besonders unterstützt (Modul-Konzept, Import/Export-Listen, getrennte Assemblierung/Comilierung).

Allerdings gibt es heute weitaus am meisten Stand-alone-Assembler. Sie ähneln Pascal: Eine Strukturierung ist nur bei kleinen Programmen gegeben, größere Entwicklungen (mit mehreren Programmierern) sind schwierig. Ebenso wird bei Pascal-Compilern auch meist streng zwischen Editor und Compiler getrennt (selbst Turbo-Pascal kann keine Syntaxprüfung beim Eintippen liefern).

b. Fähigkeiten des Assemblers

Mit Fähigkeiten meine ich das, was der Assembler (und evtl. Linker) kann bzw. versteht. In den Anleitungen findet Ihr diese Fähigkeiten meist unter "Pseudo-Ops", also Befehlen, die für den Assembler gedacht sind und nicht (direkt) in Maschinensprache übersetzt werden. Ich habe hier nur einige aufgezählt, die mir besonders wichtig erscheinen bzw. die ich kenne. Euch fallen bestimmt noch mehr ein; schreibt was dazu!

1. Z80 <-> 8080: CP/M wurde für Systeme mit dem Prozessor Intel-8080 entwickelt. Erst später brachte Zilog den leistungsfähigeren Z80 heraus. Außer den zusätzlichen Befehlen entwickelte ZILOG auch eine neue Schreibweise, z.B. heißt der Befehl LD (Load) bei Intel MOV (Move). Alte Assembler kennen daher oft nur die Intel-Schreibweise, jüngere nur die von Zilog. Am besten ist eine Umschaltung wie ".Z80" und ".8080", so daß beide Schreibweisen benutzt werden können.

2. Labels: Bei den Labels halte ich es für wichtig, daß der Assembler zumindest alle Buchstaben akzeptiert (Groß- und Kleinbuchstaben) und möglichst viele Sonderzeichen zuläßt. Ob er Groß- und Kleinschreibung auch unterscheiden soll, ist eine andere Frage; ich bin dafür, allerdings sind leicht Tippfehler möglich ("Loop"<>"loop"). Auf jeden Fall sollte die Länge der Labels nicht begrenzt sein. Ein Abschneiden nach 6 Buchstaben ist nicht zeitgemäß. Einige Assembler benötigen unbedingt einen Doppelpunkt als Abschluß eines Labels. Auch hier sehe ich den Sinn nicht ein.

3. Bedingte Assemblierung (IF .. THEN .. ELSE .. ENDIF): Diese Anweisung ist vor allem bei der Entwicklung von Programmen wichtig, die auf verschiedenen Systemen oder mit unterschiedlicher Leistung assembliert werden sollen. Bei ZEUS lautet die mögliche Syntax nur "IF..ENDIF", was auch bei anderen Assemblern leider oft der Fall ist, so daß immer zwei Blöcke "IF x ... ENDIF" und "IF NOT,x ... ENDIF" erforderlich werden. Außerdem bieten die Assembler bei der Bedingung (für "x") durchaus unterschiedliche Leistungen. Komplexe Vergleiche und vom bisherigen Übersetzungsverlauf abhängige Entscheidungen sind vor allem bei Makros (s. Punkt 4) ...

4. Makros: Mit Makros kann man sich eigene Befehle schneidern. Bei der Assemblierung werden die Makro-Befehle, die wie normale Befehle aussehen, durch andere "echte" Befehle ersetzt. Dies ist z.B. nötig, wenn der neue Prozessor HD64180 effektiv eingesetzt werden soll. Die bei ihm vorhandenen Befehle gehen über den Intel- und Zilog-Standard hinaus und sind keinem Assembler bekannt. Sie müssen dem Assembler erst per Makro "beigebracht" werden. Aber auch als Abkürzung für bestimmte wiederkehrende Anweisungenfolgen (z.B. viermal RLCA) sind sie nützlich. Damit die Makros voll ausgenutzt werden können sind zwei andere Fähigkeiten des Assemblers nötig: Erstens müssen Argumente, die beim Aufruf übergeben werden, gut ausgewertet werden können, indem leistungsfähige Sprachelemente wie IFs zur Verfügung stehen (z.B. sollte ein Aufruf "RLCAx 4" zum Erzeugen von vier RLCAs möglich sein). Zweitens müssen lokale Labels unterstützt werden. Wenn also in einem Programm zweimal ein Makro aufgerufen wird, in dem das Label "Loop" vorkommt, darf dieses nicht zweimal im Programm auftauchen, sondern muß durch z.B. "Loop001" und "Loop002" ersetzt werden.

5. Import/Export von Labels: Wer die Hochsprache Modula kennt, weiß sofort, was ich meine. Es geht darum, daß ich mehrere getrennte Programmstücke habe, die getrennt entwickelt werden. Dem einen Programmstück muß nicht jedes Label aus dem anderen bekannt sein, sondern es sind bei guter Aufteilung nur sehr wenige nötig. In jedem Programmstück werden zwei Listen angegeben: Export = welche Labels den anderen Programmstücken sein dürfen (falls diese welche benötigen); Import = welche Labels aus einem anderen Programmstück in diesem benötigt werden. Damit existieren klare Schnittstellen zwischen den Programmstücken, welche nun Module genannt werden. In jedem Modul kann jetzt das Label "LOOP" auftreten und es gibt keine Probleme, wenn dieses nicht von beiden ex- und importiert wird. So wird das getrennte Entwickeln von Modulen und das spätere Ändern erleichtert.

6. Trennung von Daten und Befehlen: Meist lassen sich Programme in zwei Bereiche gliedern: die Befehle (das ablaufende Programm) und die Daten (Variablen, Konstanten, Buffer...). Die Trennung dieser Bereiche kann nützlich sein, wenn z.B. zwei Module getrennt entwickelt werden. Im Endprodukt, dem COM-File, sollen aber trotzdem die Befehle der beiden Module in einem Bereich und die Daten alle in einem anderen Bereich stehen. Wenn "sauber" programmiert wird, d.h. wenn sich das Programm während des Ablaufs nicht selbst verändert, kann der Befehlsteil später in ein EPROM gebrannt werden und dort ablaufen, während alle Daten in den RAM-Bereich gelegt werden. Allerdings ist es für Assembler-Freaks schwer, sich solche "unsauberen", aber extrem effektiven Tricks abzugewöhnen (das sind eben "echte" Programmierer").

7. Funktionen und Zahlensysteme: Der Assembler sollte möglichst viele Operatoren verstehen können: mathematische (+, -, *, /, MOD) wie logische (AND, OR). Dazu kommen noch solche Operationen wie das Shiften und andere Verknüpfungen (XOR). Bei Zahlensystemen sollte eine Umstellung des Defaults möglich sein und zumindest die binäre, dezimale und hexadezimale Darstellung unterstützt werden. Daß auch ASCII-Zeichen (in Anführungsstrichen o.ä.) angegebbar sein sollten, ist fast selbstverständlich. Alle diese Dinge erleichtern das Verständnis des Quelltextes doch sehr, denn es kann sonst später schlecht nachvollzogen werden, welche Rechnungen mit dem Taschenrechner oder einer Tabelle ausgeführt wurden, um nun gerade auf diesen Wert zu kommen.

8. Daten-Pseudo-Ops: Darunter fallen Anweisungen, die Platz für Bytes, Worte, Strings oder Buffer freihalten. Meist heißen sie DB (Define Byte), DW (Define Word), DS (Define Storage), wobei DB auch für Strings gelten sollte. Leider fehlt manchmal DC (Define Constant), welches einen Speicherbereich festgelegter Länge mit einem Wert füllt. Das ist statt DS dort nötig, wo beim Starten des Programms ein bestimmter Wert in einem Buffer erwartet wird; ansonsten muß dies umständlich mit DBs nachgebildet werden.

9. Meldungen: Während des Assemblierens ist es oft sinnvoll, wenn eigene Meldungen ausgegeben werden, z.B.: "Erzeuge TRS-80-Model-I-Version". Außerdem
ist es manchmal nötig, eigene Fehlermeldungen anzuzeigen, wie "Sie haben das Label xyz nicht richtig initialisiert", denn solche semantischen (inhaltlichen) Fehler kann der Assembler nicht erkennen (er findet nur syntaktische).

10. LORG: Es kann vorkommen, daß ein Programm zuerst geladen wird (z.B. nach 100h), sich dann selbst verschiebt (oder einen Teil), und dann dort weiter ausgeführt wird. Solche Situationen muß der Assembler oder Linker besonders berücksichtigen und den entsprechenden Teil des Programms so assemblieren, als würde er von Anfang an dort stehen. Eine Anweisung LORG (oder so ähnlich) würde dem Assembler/Linker sagen: "Paß auf, dieser Teil wird nicht an dieser Stelle, sondern woanders ausgeführt." Wenn diese Anweisung nicht existiert, wird die Sache unter CP/M schwierig, denn Programme werden nicht wie unter Newdos einfach beim Laden irgendwo in den Speicher plaziert, sondern immer ab 100h abgelegt.

11. Listing: Zu einem Quelltext möchte ich auch immer einen Ausdruck haben, der den Quelltext und das daraus erzeugte Maschinensprache-Programm zeigt. Schön ist es noch, wenn Drucker-Parameter eingestellt werden können und einige Anweisungen existieren, die das Listing übersichtlicher machen, indem Seitennummern, Titel und Untertitel erzeugt werden. Anweisungen zum Unterdrücken von Abschnitten des Listings (LIST ON/OFF) sind auch nötig. Eine Auflistung der Labels mit ihren Werten und dem Vorkommen (Cross-Reference) hilft ungemein bei der Fehlersuche.

c. Utilities

Unter Utilities verstehe ich Programme, die beim Erstellen von Assembler-Programmen nicht nötig, aber sehr nützlich sind. Sie helfen vor allem bei der Fehlersuche und der Erstellung größerer Systeme.

1. Debugger: Ein Debugger sieht nicht unbedingt so aus wie der des Newdos, obwohl der schon nicht schlecht ist. Unter CP/M gibt es symbolische Debugger, bei denen auch Labels eingegeben werden können. Dazu muß der Assembler oder ein anderes Hilfsprogramm einen File mit den verwendeten Labels (Symbolen) und ihren Werten erstellen, so daß diese dann beim Debuggen des COM-Files verfügbar sind. Außerdem ist eine Disassemblierung von Speicherstellen fast schon Standard (aber oft nur in 8080- oder Z80-Mnemonics). Einzelschritt-Abarbeitung und das Setzen von Breakpoints ("Unterbrechungspunkten" hört sich nicht so gut an) ist selbstverständlich. Einige Debugger bieten noch viel mehr, aber mir kommt es vor allem auf eine einfache und übersichtliche Darstellung des Speicherinhalts an, was leider vernachlässigt wird.

2. Disassembler: Mit denen habe ich unter CP/M die wenigste Erfahrung. Ich habe einen gesehen, der halbautomatisch Datenbereiche herausfand und mit dem das Programm gleich kommentiert und den Labels vernünftige Namen gegeben werden konnten. Allerdings war die Bedienung zu umständlich und komlex, so daß ich ihn mir nicht weiter angesehen habe. Eine Kennzeichnung bzw. halbautomatische Erkennung von Daten sollte aber immer möglich sein. Evtl. kann auch ein Symbol-File für einen Debugger erstellt werden.

3. Library-Manager: Vor allem bei der Entwicklung größerer Programme ist es sinnvoll, auf schon vorhandene Routinen zurückzugreifen. Wenn diese geeignet zusammengefaßt (d.h. in einem Library-File verzeichnet) sind, sollte ein Programm sich beim Assemblieren oder Linken per Import oder anderer Anweisungen die benötigten Routinen daraus hervorholen können. Dies ist natürlich besonders bei der Assembler/Linker-Kombination der Fall. Damit ist nicht gemeint, daß der Linker einfach alle Routinen (Module) aus der Library einbindet, sondern es sollten nur die benötigten genommen werden. Dieses Prinzip wird übrigens auch bei Modula verfolgt.

4. File/Disk-Editoren: Von diesen nützlichen Helfern gibt es zwar einige und die Erstellung sollte aufgrund der Normung von CP/M auch nicht schwer sein, aber an Superzap und FED reicht kein mir bekannter heran. Alle werden mit kryptischen Kommandos bedient, die Hilfsmenüs sind immer dann nicht sichtbar, wenn ich sie benötige, und das Ändern eines Sektoren scheint unmöglich zu sein. Glücklicherweise brauche ich die Editoren nur, wenn eine Diskette wirklich böse zerschossen ist, was aber sehr selten vorkommt.

d. Bedienung

Leider mußte ich feststellen, daß die Bedienung der Programme unter CP/M noch chaotischer als unter Newdos ist. Oft müssen mit dem Aufruf des Assemblers (oder anderen Programmen) gleich alle Parameter eingegeben werden. Wenn dies nicht geschieht, passiert auch nichts und es kommen selten ordentliche Fehlermeldungen. Einige Programme melden sich einfach mit einem Prompt (meist "*") und harren dann der Dinge, die da kommen (eingegeben werden) mögen. Leider erfährt man selten, was für eine Eingabe erwartet wird. Optionen müssen oft mit kryptischen Abkürzungen wie "/S" oder "-S" eingegeben werden und haben nie die erwartete Wirkung, falls die überhaupt bekannt ist.

Heutzutage (1988, der Begriff "Benutzerfreundlichkeit" ist schon länger in aller Munde) ist ein Hilfesystem oder eine Menüsteuerung nicht utopisch. Zwar bieten dies viele Programme, aber es beschränkt sich dann auf eine Auflisting aller Kommandos. Das ist ungefähr so effektiv wie ein Fahrkartenautomat mit Tasten ohne Aufschrift: Jeder darf ausprobieren, was wohl bei dieser oder jener Taste (bzw. diesem oder jenen Kommando) passiert. Anzahl und Form der Parameter werden sowieso nicht angegeben und abgefragt werden sie auch nicht. Die Meldungen "Syntax Error" oder "Command Error" scheinen als einzige im Wortschatz der Programmierer vorhanden zu sein. Na, dann gute Nacht!

Gerald Schröder

Freiheit für den Bus!
(des Genie IIs)

Das Genie IIs ist eigentlich nicht mehr als ein normaler TRS-80, aber mit einem Bus und mehreren Steckkarten. Dadurch ist die Erweiterung des Systems zwar leicht, aber die Signale auf dem Bus sind ganz auf die Nachbildung eines TRS-80 zugeschnitten und halten sich nicht im geringsten an den eigentlichen Standard ECB. Das kann ich zwar nicht ändern, aber mir stinkt es gewaltig, daß einige Signale (fast) nie benötigt werden. Die kann ich besser verwenden. Aber zuerst gilt es, diese Signale von ihren ursprünglichen Verbindungen abzuschneiden und lokal (auf den Karten) zu ersetzen.

Ich habe bei mir mit dem Signal $\overline{32C}$ angefangen. Es dient zur Umschaltung auf die 32-Zeichen-Darstellung, die wohl nie jemand benutzt. Eigentlich wäre 64C als Bezeichnung besser gewesen, denn High bedeutet, daß 64 Zeichen angezeigt werden. Aber irgendwie scheinen die Hardware-Freaks auf invertierte Logik zu stehen, weiß der Teufel (bzw. Uwe Böker) warum.

Dieses Signal tritt nur auf zwei Karten auf: Auf der I/O-Karte wird es erzeugt (durch Schreiben in Port FF) und kann abgefragt werden (durch Lesen von Port FF). Auf der Video-Karte wird eine Umschaltung damit vorgenommen.

Der Umbau:
1. I/O-Karte:
   a) U10, Pin 4 -> +5V (schätzungsweise mit Widerstand 1kΩ)
   b) U11, Pin 14 rausbiegen
      alternativ: Leiterbahn zum Bus-Signal A8 abtrennen
2. Video-Karte:
   a) U8 rausnehmen (LS157)
   b) an der Fassung von U8 (unten) folgende Verbindungen herstellen:
      Pin 12 <-> 13
      Pin 3 <-> 4
      Pin 6 <-> 7

Neben dem LS157 ist jetzt auch U7, Pin 2-7 frei. Das freie Bussignal wird bei mir dazu benutzt, um ein Signal $\overline{0\text{-}32K}$ von der CPU- zur I/O-Karte zu leiten. Dieses Signal wird benötigt, um bei Einsatz des Bankers zu erkennen, ob jetzt der Memory-mapped-I/O-Bereich oder eine höhere Bank angesprochen wird.

Was Ihr jetzt mit dem freien LS157 machen sollt? Jedenfalls nicht wegwerfen! Evtl. zeige ich Euch im nächsten Info, wie man damit die Memory-Map auf Portbetrieb umstellt (nützlich unter CP/M). Das nennt man dann Hardware-Recycling.

Gerald Schröder

Die Ramdisk und das elfte Gebot

Ich war so unvorsichtig, das elfte Gebot (Du sollst nicht schwärmen...)
nicht zu beachten und habe Gerald Schröder von meiner Solid-State-Floppy
erzählt. Zur Belohnung hat er mir natürlich sofort einen Artikel auf's Auge
gedruckt, damit sich jeder selbst ein Bild über den Sinn oder Unsinn von
sowas machen kann.

Die Karte ist im Grunde eine ganz gewöhnliche Speichererweiterung, aber sie
hat ein paar kleine Besonderheiten, die sie sehr interessant machen. Einmal
muß man im Rechner absolut nichts umbauen und sie ist keine Erweiterung auf,
sondern um max. 256kB, d.h. sie ist unabhängig von dem schon vorhandenen
Hauptspeicher. Die völlige Systemunabhängigkeit wird dadurch erreicht, daß
die Karte über Ports angesprochen wird und zwar fast wie ein richtiger
Floppy-Controller. Man gibt also über einen Port die 'Spurnummer' aus und
auf einem anderen die 'Sektornummer', dann werden die Daten einfach per IN
oder OUT hin und her geschoben, um die interne Adressierung auf der Karte
muß man sich also überhaupt nicht kümmern. Wenn man will, kann man sogar
seinen PC oder Atari damit beglücken. Bestückt wird sie wahlweise mit stati-
schen RAMs (akkugepuffert) oder EPROMs, wobei der Clou an der Sache der ist,
daß man die Karte auch gleich als EPROMMER verwenden kann. Leider braucht
man dazu aber ein spezielles Programm, von dem ich nur einmal unter CP/M was
gehört habe. Da bei mir der Inhalt aber sowieso nur ein paar Tage lang
konstant bleibt, habe ich mich lieber gleich für die RAM-Version entschie-
den. Der große Vorteil ist, daß ich gleich beim Einschalten alle meine
wichtigen Utilities griffbereit habe und das Laden völlig geräuschlos und
sehr schnell geht. Wenn man sich erst mal an sowas gewöhnt hat, möchte man
es nicht mehr hergeben. Stellt Euch vor, Ihr arbeitet viel mit Superzap,
dann werdet Ihr es wahrscheinlich auf fast allen Disketten irgendwo haben,
damit es immer zur Hand ist. Das braucht nicht nur viel Platz, auch wenn mal
ein Zap fällig ist oder es gegen ein anderes Utility ausgetauscht werden
soll, (bei Superzap wohl weniger, aber bei anderen Programmen) müßt ihr alle
Disketten durchackern, wo überall eine Kopie ist. Ich brauche das nur einmal
auf der RAM-Floppy zu tun und bin fertig. Am schönsten ist es aber, wenn man
dort Programme hat, die viel mit Overlays arbeiten. Früher habe ich immer
geflucht, wenn dauernd die Floppy angelaufen ist, heute merkt man das Nach-
laden überhaupt nicht mehr.

Erst habe ich Euch den Mund wässrig gemacht, jetzt kommt eine kleine Abkühl-
ung. Nichts gegen die c't, aber dieses Projekt teilt das Schicksal so viel-
er: es kann (wird) größere Probleme geben. Sowohl bei Helmut Bernhardt als
auch bei mir hat es längere Zeit gedauert, bis die Karte lief. Der Aufbau
selber ist nicht weiter schlimm, aber irgendwo muß sie sehr empfindlich
sein, wie es aussieht kann schon ein IC vom 'falschen' Hersteller das Aus
bedeuten. Wer sich eine längere Fehlersuche nicht zutraut, oder keinen
kennt, der einen kennt..., der sollte doch lieber vorsichtig sein.

Wem das noch immer nicht reicht, der soll sich mal ausrechnen, was das
kostet. Für die Leerkarte+Porto muß man DM 78 anlegen, für die CMOS-RAMs
(256kBit) zwischen 20 und 40 Mark, je nachdem wo man sie kauft (EPROMs ca.
10.-). Die Fassungen und das ganze Kleinzeug kommen so ungefähr auf 50.-,
d.h. für statische 256kB kommt man auf ca. 300 Eierchen.

Hat noch jemand bis hierher weitergelesen ? Gut, wenn noch Interesse besteht
stehe ich gerne zur Verfügung. Der Artikel in der c't war übrigens in der
Nummer 5/86, bei Bedarf schicke ich Euch gerne eine Kopie.

Bei mir läuft die SSD unter CP/M und Arnulf Sopp hat mir mal geschrieben,
daß er an einem Treiber für GDOS/NEWDOS bastelt; die Softwarefrage ist also
keine mehr.

Was tun die Mutigen, wenn es heißt 'Freiwillige vor' ? Richtig, sie machen
Platz, damit die Freiwilligen durchkommen. Also, wer meldet sich ?

Alexander Schmid

Prof 180x und mc68000-ECB

Helmut Bernhardt

Der mc68000-ECB aus mc 1-3/87 ist ein Coprozessor-Board mit einem Motorola 68000 Prozessor auf einer Europakarte für den ECB-Bus. Diese Karte kann in jedem DMA-fähigen ECB-Bus-Computer eingesetzt werden, wobei dann abwechselnd oder auch gleichzeitig Z80- und 68000-Software gefahren werden kann. Im einfachsten Fall kann das 68000-Board aber auch nur dem Z80 eine bis zu 2MB große RAM-Floppy zur Verfügung stellen.

Wenn der Computer im 68000-Modus läuft, spielt der Z80 (oder im Prof 180x der HD64180) I/O-Knecht für den 68000. Der 68000 bedient sich des BIOS des CP/M 80. Der 8Bit-Prozessor wartet nur auf Aufgaben, die ihm der 68000 stellt.

Die Kommunikation zwischen den beiden CPUs ist für beide unterschiedlich zu handhaben. Der 8Bit-Prozessor kann dem 68000 einen 1Byte-Befehl in ein Latch schreiben, während der 68000 per DMA in den Speicher des Wirts-Computers schreiben und auch daraus lesen kann. In den 16MB-Adreßraum des 68000 ist ein Fenster von 2MB Größe gelegt, in dem die vermeintlichen 64KByte des Z80-Systems erreichbar sind. Zur Adressierung im Z80-System werden dann die Adressen A1-A16 des 68000 auf A0-A15 des ECB-Bus geschaltet und BUSAK* low gezogen. Außerdem werden auch die Z80-kompatiblen Signale MERQ*, IORQ*, RD* und WR* erzeugt und der Datentreiber des 68000-Boards zum ECB-Bus wird freigegeben (für D8-D15 des 68000).

Der Prof 180x hat aber nicht nur 64KByte RAM; zur Adressierung innerhalb der 512K des Prof 180x müssen noch die Adressen A16-A18 mit festen Pegeln für den Bereich 40000H-4FFFFH erzeugt werden. In diesem physikalischen Adreßbereich liegt im Prof die TPA, in der ein Kommunikationsprogramm den Datenaustausch mit dem 68000 erledigt und von wo der 68000 sich Daten abholt und wohin er seine Daten an das 8Bit-System überträgt. Das läßt sich noch recht einfach mit einem 74LS367 Tristate-Treiber, der A16-A18 des ECB-Bus bedient, auf dem 68000-Board erledigen. Dieser Treiber wird mit den Pins 1, 8 und 16 auf IC13, 74LS367 aufgelötet. Die Pins 3, 5 und 7 werden mit A18, A17 und A16 des ECB-Bus verbunden und an Pin 2 wird ein Widerstand von 4K7 gegen +5V angeschlossen. Die Pins 4 und 6 werden an GND gelegt.

Das eigentliche Problem war weder vorher abzusehen noch durch einen einfachen Patch zu beheben - so schien es zumindest im ersten Augenblick. Die DMA-Fähigkeit des Prof 180x brücksichtigt zunächst die Steuerung der Treiber zum ECB-Bus. Durch ein PAL 14L8 ist die Bussteuerung ohne den sonst für die Berücksichtigung der DMA-Fähigkeit nötigen TTL-Aufwand mit folgender PAL-Gleichung für das Steuersignal des Datentreibers gelöst:

```
/DIR = /IOE * /RD * /ADR * A7 * /A5   ;Treiberrichtung beim Lesen interner Ports immer
                                      ;zum Bus hin
     + /ME * /RD * A19                ;Zum Bus beim Lesen der internen 512K RAM, A19 ist
                                      ;ein im gleichen PAL erzeugtes Signal zur Unter-
                                      ;scheidung zwischen internem und externem RAM
     + RD * LIR * BUSAK               ;zum Bus auch, wenn weder ein Interrupt-Acknowledge
                                      ;noch DMA-Zugriff noch eine Schreiboperation erfolgt
```


Die Treiber für den Adreßbus und Steuerbus werden einfach durch den Pegel von BUSAK* gesteuert. Wenn BUSAK* low aktiv ist, treiben diese die entsprechenden Signale vom Bus zur CPU-Karte.

Damit kann ein externer DMA-Controller zwar auf das CPU-Board zugreifen aber noch nicht den Speicher lesen und schreiben, denn das flexibel bei jeder CPU-Taktrate funktionierende Timing der RAM-Steuersignale RAS*, MUX und MUX* und CAS*, das aus HD64180-spezifischen Signalen abgeleitet wird, kann von einem DMAC nicht unterstützt werden. Die im Prof 180x verwendete Schaltung (Abb.2) zeigt, daß aus dem MERQ* mit der nächsten steigenden Flanke des Systemtaktes die Signale MUX und MUX* erzeugt werden und daß ein vom Systemtakt abhängiges HD64180-spezifisches Signal E für das Timing des CAS*-Signals herangezogen wird. Ein asynchron arbeitender DMAC kann diese Signale garnicht bedienen, E ist kein ECB-Bus-Signal und der CPU-Takt ist kein Tristate-Signal, das beim DMA-Zugriff vom DMAC bedient werden kann.

Es galt nun, möglichst unter Beibehaltung des eleganten und flexiblen Timings der RAM-Steuerung durch den HD64180, einem externen DMAC eine weitere Möglichkeit zu schaffen, ausschließlich mit MERQ* (=ME* beim HD64180) und Verzögerungsgliedern die getimeten RAM-Signale zu erzeugen. Wenn dies z.B. mit Gatterlaufzeiten gemacht wird, sind dann allerdings der Bandbreite möglicher Taktraten des DMAC engere Grenzen gesetzt. Die in Abb.3 dargestellte Schaltung erlaubt das durch BUSAK* gesteuerte Umschalten zwischen ursprünglichem Timing des Prof 180x und über Gatterlaufzeiten geregeltes Timing bei Kontrolle durch den externen DMA-Controller. Die Anzahl Gatter für die Verzögerung gilt für einen mit 8MHz getakteten 68000. Bei Verwendung entsprechend schneller RAMs und EPROMs auf dem 68000-Board macht diese Schaltung aber auch bei einem mit 16MHz getakteten 68000 mit.

Es sei noch erwähnt, daß im DMA-Betrieb MUX und CAS* nicht gleichzeitig kommen, wie die Schaltung beim ersten Hinsehen erwarten läßt. Das im PAL aus E erzeugte CAS* ist gegenüber E nochmals um die Gatterlaufzeit des PALs verzögert.

Die Schaltung sieht auf den ersten Blick nach sehr viel Flick- und Sägearbeit am CPU-Board aus. Das ist aber durch Verwenden eines kleinen Adapterplatinchens, das in den Sockel von Z9, 74S74 gesteckt wird und Z9 selbst mit aufnimmt, nicht der Fall. Außer dem 74S74 bringt das Platinchen noch ein 74LS367 (6 Tristate-Treiber) und ein 74LS368 (6 invertierende Tristate-Treiber) unter. über den Sockel von Z9 werden fast alle benötigten Signale (MERQ*, CLK, RFS*) dem Platinchen zugeführt und auch die jeweils erzeugten Signale MUX und MUX* abgeleitet. BUSAK* muß durch freie Verdrahtung von J9 und E von Pin60 des HD64180 bezogen werden. Das Ausgangssignal E wird an Pin7 des PALs gelegt. Damit die bisherige Verbindung zwischen Pin60 des HD64180 und Pin7 des PALs unterbrochen wird, muß Pin7 des PALs aus der Fassung gebogen werden.

Eine ähnliche Herleitung von MUX und/oder MUX* weisen einige moderne TRS80-kompatible Computer auf, bei denen eine Taktumschlatung zwischen 1,77MHz und einer anderen, deutlich höheren Geschwidigkeit möglich ist. Leider ist all diesen Geräten ein nicht DMA-fähiger Bus konstruiert worden. Ein externer DMAC kann hier nur direkt mit den entsprechenden Signalen der CPU verbunden werden. Unverständlicherweise werden dann aber die Treiber für Adreß- und Steuerbus doch durch ein invertiertes BUSAK* oder BUSREQ* freigegeben. Offensichtlich haben ganze Generationen von Kostrukteuren von einem Urtrottel abgeguckt. Deshalb muß man dann zusätzlich noch die Freigabe dieser Treiber mit UND anstelle mit BUSAK bzw. BUSRQ einrichten.

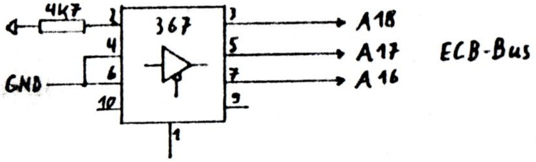


Abb.1: Erzeugen der richtigen Pegel von A16-A18 für die Adressierung der TPA beim Prof 180x bei physikalisch 40000H-4EFFFH

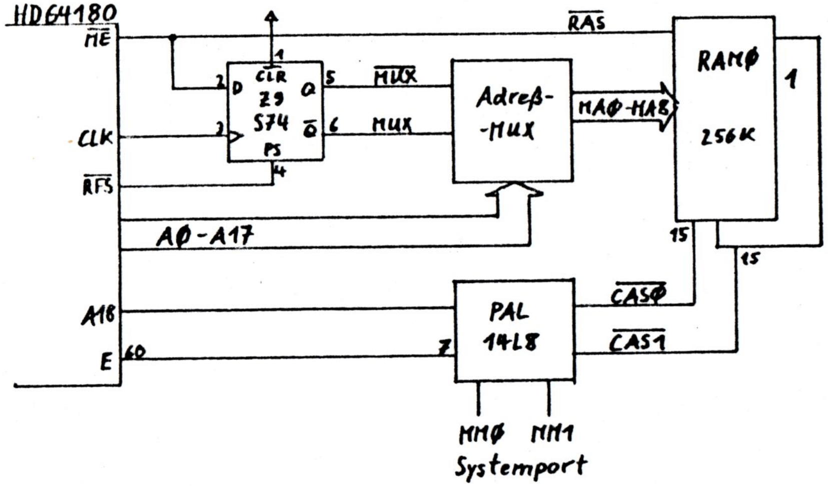


Abb.2: das für jede Taktrate funktionierende Timing der RAM-Steuersignale im Prof 180x

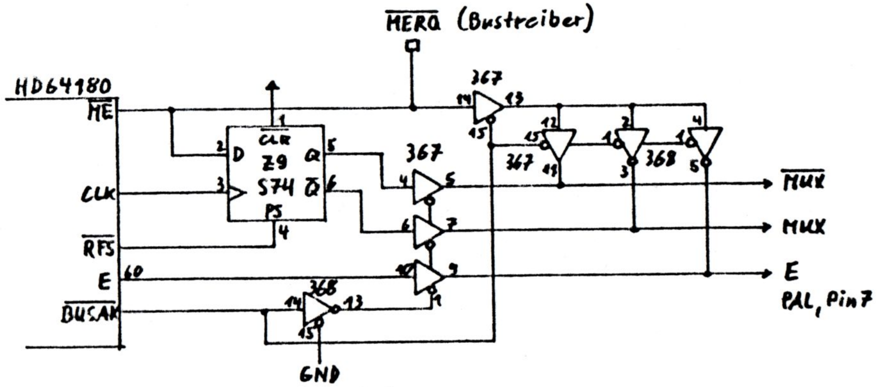


Abb.3: auf einer Adapterplatine im Sockel von Z9, 74S74 wird die Umschaltung zwischen den im Prof 180x erzeugten und den aus MERQ* abgeleiteten RAM-Steuersignalen untergebracht, BUSAK* regelt die Umschaltung

TRS 80 Japan Modell aufbohren

auf
1,77/3,55MHz Taktumschaltung
CP/M-Fähigkeit
256K gebanktes RAM

H. Bernhardt

Aufgrund eines mir zunächst üppig erschienenen, sich aber nachträglich als keinesfalls dem Aufwand adäquat erwiesenen Honorars ließ ich mich breitschlagen, in einem vermeintlichen TRS 80 Modell 1 einen 256K-Banker, einen CP/M-Banker und eine einfache Taktumschaltung einzubauen. Bei übernahme des Gerätes stellte sich aber heraus, daß es sich um ein Japan-Modell handelt, bei dem alle meine Unterlagen zum TRS 80 unbrauchbar sind. Es galt also zunächst, dessen Schaltung zu analysieren, um dann die nötigen Änderungen für die Erweiterungen vorzunehmen und zu sehen, an welchen Punkten die benötigten Signale abzugreifen sind. Solches hatte ich vor Jahren beim Komtek 1 schon mal durchgeführt, so daß mir der Mut nicht ganz verlorengegangen war.

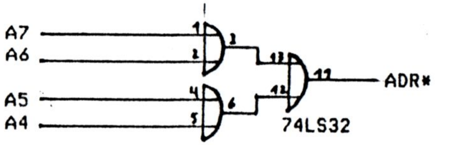
Eventuell besitzt hier im Club noch irgendjemand diese Variante des TRS 80 (vielleicht, ohne es selbst zu wissen, denn äußerlich sehen Original und Fälschung gleich aus), dem die von mir extrahierten Informationen etwas nützen. Deshalb will ich es riskieren, einige Seiten des kostbaren Platzes in diesem Info für diesen Zweck zu verbrauchen. Den 256K-Banker habe ich vor längerer Zeit schon einmal im Club 80 Info beschrieben, so daß hier nur noch dessen Einbau und Anschluß am CPU-Board breitgetreten werden soll.

Den CP/M-Banker (für den es ein angepaßtes CP/M 3.0 BIOS gibt, zu dem auch die Source-Codes vorliegen) habe ich vor noch längerer Zeit zusammen mit Christof Ueberschaar entwickelt und im Info des mittlerweile eingegengenen Bremerhavener TRS 80/GENIE-Clubs veröffentlicht. Um die zahlreichen Mitglieder des Club 80, die dort auch mal Mitlied waren, nicht noch einmal damit zu langweilen, soll nur die Schaltung dieses CP/M-Bankers und eine Skizze mit den Anschlußpunkten auf dem Platinchen vorgestellt werden. Wer darüber hinaus Interesse an dem Artikel hat, kann sich mit 1,90DM Rückporto im Brief bei mir melden.

Der CP/M-Banker setzt voraus, daß im Grundgerät 64K oder 256K RAM vorhanden sind. Durch Ausgabe des Bytes 00H an den Port 50H wird der ROM-Bereich abgeschaltet und im Bereich 0000H-37DFH wird RAM verfügbar gemacht. Diese Einstellung läßt sich durch Software nicht wieder rückgängig machen. Nur ein RESET stellt wieder die urspünglichen ROMs bereit. Durch ein kleines Maschinenprogramm kann der ROM-Inhalt nach oben kopiert werden, dann der Umschaltbefehl ausgeführt und schließlich der ROM-Inhalt an seine ursprüngliche Adresse zurückgeschoben werden. Damit ist das Level 2 ROM weich geworden und kann beliebig gepatcht werden.

Wenn an den Port 50H der Wert 40H (D6=1) ausgegeben wird, ist anschließend auch noch der memory mapped I/O-Bereich 37E0H-3FFFH nach F7E0H-FFFFH verlegt worden, so daß ein fast 62K großer RAM-Bereich ab 0000H vorliegt, womit CP/M leben kann.

Es sei noch gesagt, daß die Verwendung des Ports 50H (redundant decodiert auf 50H-5FH) bei gleichzeitig vorhandener HRG1B wegen deren Verschleiß an Portadressen (00H-7FH) eine feinere Decodierung der HRG1B erfordert. Prinzipiell genügt es, anstelle von A7 (an die HRG1B) ein aus A7 und A6 OR-verknüpftes Signal an die Karte zu führen. Damit reduziert sich deren Portadreßhunger auf den Bereich 00H-3FH. Da dann aber ohnehin ein zusätzliches IC 74LS32 nötig ist, läßt sich die HRG1B auch noch auf die Portadressen 00H-0FH beschränken.



anstelle von
A7 an die
HRG1B

Aber zurück zum Anschluß der Banker an das CPU-Board des Japan-Modells. Zunächst sollen die Änderungen besprochen werden, die den Einsatz von 64K bzw.256K RAM-Chips auf dem CPU-Board ermöglichen.

Auswechseln der 4116-RAMs

Dafür müssen die unsinnigerweise eingelöteten 4116-RAMs entfernt werden. Es werden alle Pins aller RAMs (Z15-Z22) möglichst hoch abgekniffen und die verbleibenden Stümpfe der Pins einzeln ausgelötet und schließlich die Lötlöcher mit einer Entlötpumpe freigemacht.

Vor dem Einlöten der DIL16-Fassungen für die neuen RAMs sind noch ein paar Änderungen nötig, um die +12V-Versorgung von den Pins 8 der RAMs und die -5V-Versorgung von den Pins 1 der RAMs zu entfernen und die +5V-Versorgung von den Pins 9 zu den Pins 8 der RAMs zu verlegen, und um an die Pins 9 ein neu zu erzeugendes gemultiplextes Signal MA7 (und bei 256K-Chips an die Pins 1 ein MA8) zu führen. Dafür sind folgende Arbeiten durchzuführen:

Alle Kondensatoren (100nF) oberhalb und unterhalb der RAMs, die mit einem Beinchen Kontakt mit den Pins 1 oder 9 der RAMs haben, werden ausgelötet.
Die bestückungsseitige, breite, die Pins 1 der RAMs verbindende Leiterbahn wird auf Höhe der 33Ohm-Widerstände (R101) durchtrennt (-5V an den Pins 1 der RAMs).
Die lötseitige, breite, um RAM Z15 und die 33-Ohm-Widerstände herum zu Pin 16 von Z14, 74LS157 führende Leiterbahn wird auf Höhe der 33-Ohm-Widerstände (R101) durchtrennt (+5V an den Pins 9 der RAMs).
Die vom Kondensator C15 auf der Lötseite zu Pin 8 von Z19 (RAM) verlaufende, breite Leiterbahn wird neben C15 durchtrennt (+12V an den Pins 8 der RAMs).
Eventuell müssen lötseitig noch einige zusätzliche Kondensatoren von 100nF über die Pins 8 und 16 der RAMs gelegt werden. Optimal ist ein 100nF-Kondensator je RAM. Diese Kondensatoren sollten nach dem Einlöten der DIL16-Sockel für die neuen 4164- bzw. 41256-RAMs lötseitig an die Pins 8 und 16 der Sockel gelötet werden. Es ist dafür zu sorgen, daß keine Kontakte zu anderen Pins entstehen (Isolierhülsen auf die Beinchen der Kondensatoren).
Für den Einsatz von 4164-RAMs mit 7Bit-Refresh reicht es, wenn A15 von Z48, Z80-CPU, Pin 5 an Pin 13 von Z13, 74LS157 und A14 von Z48, Z80-CPU, Pin 4 an Pin 14 von Z13, 74LS157 verdrahtet wird und der Pin 12 von Z13, 74LS157 mit den untereinander verbundenen Pins 9 der RAMs verdrahtet wird.

Für den Einsatz von 4164-RAMs mit 8Bit-Refresh oder 41256-RAMs (mit grundsätzlich 8Bit-Refresh) muß A7 von Pin 3 von Z14, 74LS157 abgetrennt werden. Die lötseitige Leiterbahn zu Pin 3 von Z14 wird direkt neben dem Pin 3 durchtrennt. An diesen Pin 3 wird dann A14 von Z48, Z80-CPU, Pin 4 verdrahtet. A15 von Z48, Z80-CPU, Pin 5 wird mit Pin 13 von Z13, 74LS157 und Pin 12 von Z13 wird mit den Pins 9 der RAMs verdrahtet. A7' von einem 8Bit-Refresh-Generator (z.B. auf dem 256K-Banker) wird an Pin 14 von Z13, 74LS157 gelegt.

Im Expansion-Interface des Japan-Modells (schwarzes Blechgehäuse) werden alle RAMs 4116 aus den Fassungen gezogen und verworfen.
Neben den RAMs befinden sich 2 Stück 74LS244-Treiber. Bei beiden ICs wird die

bestückungsseitig von den Pins 1 wegführende Leiterbahn direkt neben den Pins 1 durchtrennt.
Auf der Lötseite wird die die Pins 19 der 74LS244-ICs verbindende Leiterbahn neben der Durchkontaktierung beim 74LS32-IC durchtrennt. Bei beiden 74LS244-ICs wird eine Verbindung zwischen den Pins 1 und 19 hergestellt.
Auf der Bestückungsseite wird an einem der 74LS244-ICs ein Widerstand von 2k2 von Pin 1 nach Pin 20 gelegt.

Taktumschaltung

Nach dieser Änderung sollte bei Verwendung von 4164-RAMs mit 7Bit-Refresh der Computer normal mit 48K RAM arbeiten. Da die langsamen RAMs im Expansion-Interface nun nicht mehr benutzt werden, läßt sich eine Speed-Umschaltung zwischen 1,77MHz und 3,55MHz einbauen. Die hier mit einem einfachen Umschalter realisierte Variante schaltet einen der beiden Takte zur CPU durch. Das Umschalten ist also nicht während des Betriebs möglich (sollte es zumindest nicht, hier hat es in den meisten Fällen ohne Systemabsturz geklappt). Für diese Umschaltung sind folgende Operationen durchzuführen:

Auf der Bestückungsseite
wird die Leiterbahn, die zwischen Pins 13 und 14 von Z65, 74LS92 herauskommt und zwischen Pins 2 und 3 von Z66, 74LS367 verschwindet, durchtrennt.
Die Pins 12 und 13 von Z63, 74LS74 werden miteinander verbunden.
Die Pins 8 und 9 von Z65, 74LS92 werden an die Außenkontakte eines Umschalters gelötet. Der Mittelkontakt des Umschalters wird mit Pin 14 von Z66, 74LS367 verbunden (möglichst kurze Leitungen).

Auf der Lötseite
wird die Verbindung von Pin 12 von Z63, 74LS74 zur Durchkontaktierung unter dem IC durchtrennt.

CP/M-Banker einbauen

Für den Anschluß des CP/M-Bankers sind die in Abb.3 dargestellten Änderungen in der Adreßdecodierung der unteren 16K auf dem CPU-Board vorzunehmen. Im einzelnen bedeutet dies folgende Arbeiten:

Bestückungsseite:
Die Leiterbahn, die zwischen den Pins 2 und 3 von Z65, 74LS92 herauskommt und zwischen den Pins 11 und 12 von Z64, 74LS32 verschwindet, wird durchtrennt.
Der Pin 3 von Z66, 74LS367 wird mit Pin 13 von Z64, 74LS32 verbunden.
Der Pin 6 von Z64, 74LS32 wird mit Pin 12 von Z44, 74LS00 verbunden.
Die von Pin 12 von Z49, 74LS244 zur ca. 2 cm entfernten Durchkontaktierung führende Leiterbahn wird durchtrennt (A14).
Die daneben verlaufende (zwischen den Pins 11 und 12 von Z49, 74LS244 herauskommende) Leiterbahn wird durchtrennt (A15).
Die an Pin 10 von Z61, 74LS139 führende Leiterbahn wird neben dem Pin durchtrennt.

Lötseite:
Die Leiterbahn von Pin 11 von Z64, 74LS32 zur Durchkontaktierung bei Pin 8 von Z64, 74LS32 wird durchtrennt.
Die Leiterbahn von Pin 13 von Z44, 74LS00 zur Durchkontaktierung wird durchtrennt.
Die Verbindung zwischen den Pins 2 und 6 von Z64, 74LS32 wird durchtrennt.
Die Leiterbahn von der Durchkontaktierung auf Höhe der Pins 5 und 10 von Z64, 74LS32 zur Durchkontaktierung auf Höhe der Pins 3,4,11 und 12 von Z44, 74LS00 wird durchtrennt.

Auf dem CP/M-Banker:
Das IC Z10, 74LS32 wird nicht bestückt und der Jumper J5 bleibt offen. Die Jumper J1, J2, J3 und J4 werden gelegt.
Die Anschlüsse SYSRES*, ROM3*, ROM4*, ROM3'*, ROM4'* sowie 37EX* bleiben unberücksichtigt.

Der CP/M-Banker wird gemäß folgender Tabelle mit dem CPU-Board verbunden:

Signal auf dem CP/M-Banker	Anschluß auf dem CPU-Board		
A4	Z68	74LS244	Pin5
A5	"	"	16
A6	"	"	3
A7	"	"	18
A8	Z49	74LS244	Pin14
A9	"	"	16
A10	"	"	18
A11	"	"	3
A14	"	"	12
A15	"	"	9
12-14(16)K*	Z61	74LS139	Pin15
OUT*	Z30	74LS367	Pin7
RESET*	Z46	74LS04	Pin12
NMI*	Z45	74LS02	Pin10
ROM1*	Z60	74LS11	Pin8
ROM2*	Z61	74LS139	Pin6
RAM*	Z64	74LS32	Pin11
KB*	Z61	74LS139	Pin10
ROM1'*	Z42	ROM A	Pin20 — diese Pins werden aus
ROM2'*	Z43	ROM B	Pin20 — der Fassung gebogen
RAM'*	Z49	74LS367	Pin15
KB'*	Keyboard-Stecker Pin10		
A14'	an die Durchkontaktierung, deren Verbindung zu Pin 12 von Z49, 74LS244 durchtrennt wurde		
A15'	an die Durchkontaktierung direkt daneben		

256K-Banker einbauen

Um den Anschluß des 256K-Bankers vorzubereiten, wird auf Z14, 74LS157 noch ein weiteres IC 74LS157 mit den Pins 1, 8, 15 und 16 huckepack-gelötet; alle anderen Pins dieses ICs werden waagerecht abgebogen. An dieses IC Z14' werden die Signale A7', A15', A16 und A17 des 256K-Bankers angeschlossen.

Signal des Bankers	Pin von Z14'
A7'	Pin14
A15'	" 13
A16	" 11
A17	" 10

Die Pins 9 der RAMs werden mit Pin 12 von Z14' und die Pins 1 der RAMs mit Pin 9 von Z14' verbunden. Die eventuell beim Umbau auf 64K RAM (event. mit CP/M-Banker) hergestellte Verbindung zwischen Pin 12 von Z13 und den Pins 9 der RAMs wird wieder entfernt. Das anstelle der durchtrennten Leiterbahn (A7 der Z80-CPU) an Pin 3 von Z14 geführte Signal A14 (Pin 4 der Z80-CPU) bleibt angeschlossen.

Die Eingangssignale für den 256K-Banker werden an folgenden Punkten auf dem CPU-Board entnommen und an die entsprechend bezeichneten Punkte des 256K-Bankers geführt (freie Verdrahtung).

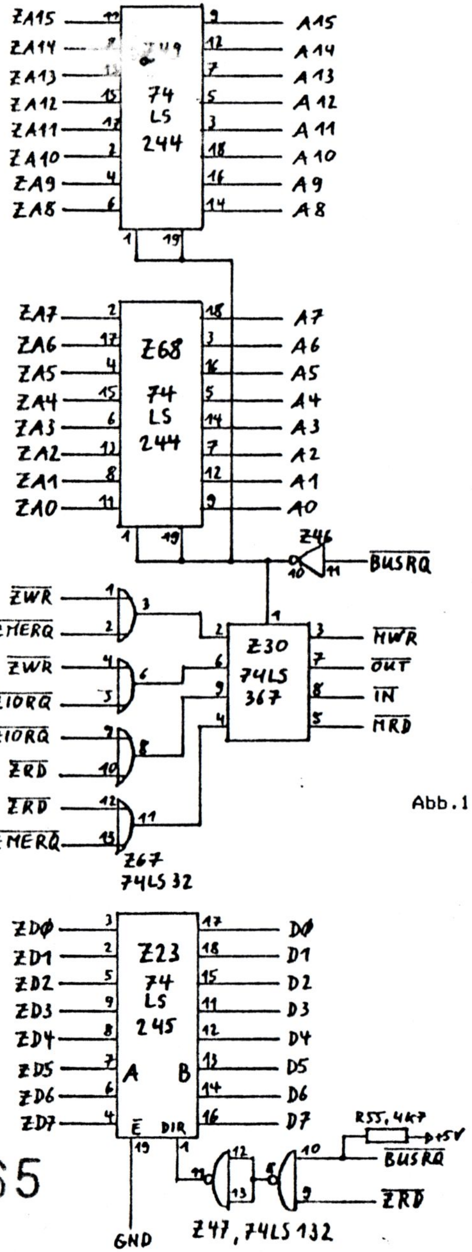
Signal	CPU-Board IC	Typ	Pin	256K-Banker Anschlußpunkt
D0	Z23	245	17	g
D1	"	"	18	f
D2	"	"	15	e
A0	Z68	244	9	8
A1	"	"	12	9
A2	"	"	7	1
A3	"	"	14	2
A4	"	"	5	10
A5	"	"	16	4 und 5
A6	"	"	3	6 und m
A7	"	"	18	7 und n
A15	Z49	244	9	h
SYSRES*	siehe unten			a
OUT*	Z30	367	7	1
RFSH*	Z49	Z80-CPU	28	o

Bei "nur" 256K RAM werden die Eingangssignale D3 und D4 sowie CASEN* und die Ausgangssignale CASEN1* bis CASEN4* nicht angeschlossen.
Das Signal SYSRES* wird auf dem CP/M-Banker aus RESET* und NMI* erzeugt und kann von dort bezogen werden. Wenn der CP/M-Banker nicht eingebaut wird, muß RESET* (von Z46, 74LS04, Pin 12) oder NMI* (von Z45, 74LS02, Pin 10) stattdessen verwendet werden.

Bei den spartanischen Platzverhältnissen im TRS 80 Modell 1 muß beim Einbau von Erweiterungen von vornherein geplant werden, wo die zusätzlichen Platinchen in der Schachtel Platz finden. Das CPU-Board liegt mit der Bestückungsseite nach unten im Gehäuse. Erweiterungen, die darunter liegen sollen, müssen also von der Bestückungsseite her angeschlossen werden. Platz ist unter dem CPU-Board nur in der hinteren Hälfte des Gehäuses vorhanden. Die Anschlußdrähte sollten also gleich so lang gewählt werden, daß diese Einbaulage möglich ist. Allerdings ist in digitalen Schaltungen prinzipiell das Einhalten kurzer Leitungslängen anzustreben. Man sollte also auch nicht zu großzügig mit der zu verwendenden isolierten Litze umgehen.

Obwohl so mancher rechtschaffener Standard-Z80 die 3,55MHz besser verträgt als mancher A-Typ, sollten doch die paar Mark und fünfzig für einen auf 4MHz ausgelegten Z80 investiert werden. Dagegen ist die Investition für die neuen RAM-Chips bei den heutigen Preisen durchaus die Überlegung wert, ob man das dem alten Möbel noch angedeihen lassen will.

Außerdem ist (auch wenn für einen potentiellen Nachbauer der Aufwand für die Analyse des Japan-Modells entfällt) der Arbeitsaufwand und die Wahrscheinlichkeit, einen Fehler zu machen, recht hoch. Der eigene Performance-Ehrgeiz sollte also ruhig überschlafen werden. Auch die CP/M-Fähigkeit ist bei einem nach wie vor auf 64*16 Zeichen beschränkten Bildschirm begrenzt tauglich und Tasten für deutsche Umlaute hatte ein TRS 80 Modell 1 noch nie.



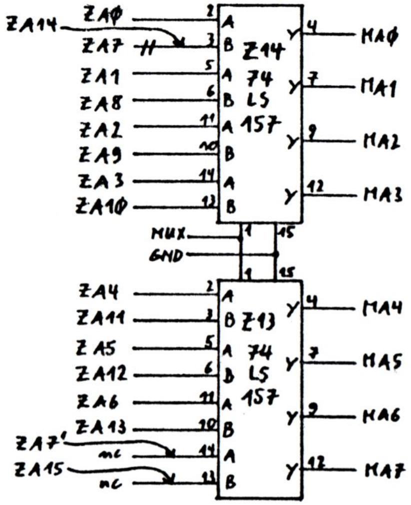


Abb.1: Treiber für Adressen und Daten des Z80 (die mit Z.. bezeichneten Signale sind die des Z80)
Erzeugen der Signale MWR*, MRD*, OUT* und IN*
Adreßmultiplexer für die RAMs (eingezeichnet sind die Änderungen für 4164--RAMs mit 7- oder 8-Bit-Refresh; für 8-Bit-Refresh-RAMs muß ZA7' von einem Generator für 8-Bit-Refresh-Adressen kommen)

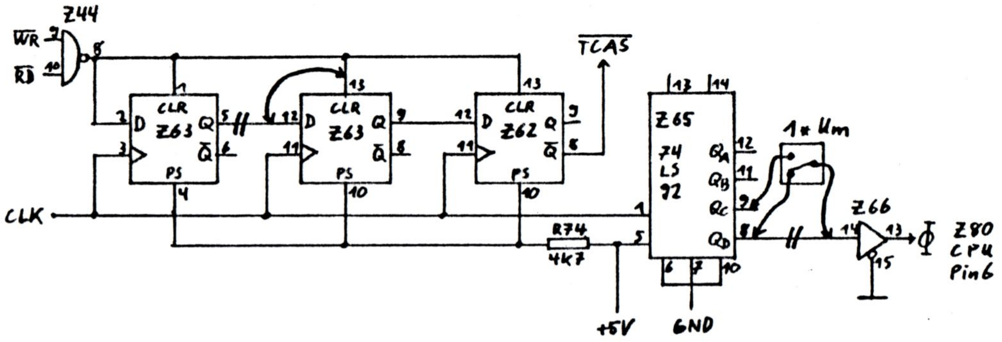


Abb.2: Änderungen für die Umschaltung des Systemtaktes zwischen 1,77 und 3,55 MHz

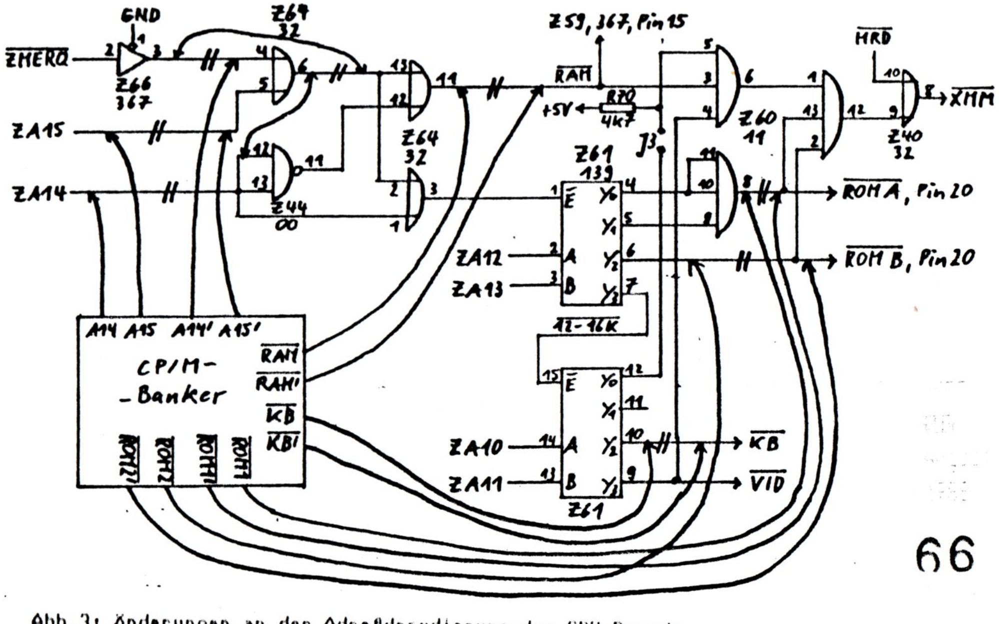


Abb.3: Änderungen an der Adreßdecodierung des CPU-Boards und Anschluß des CP/M-Bankers

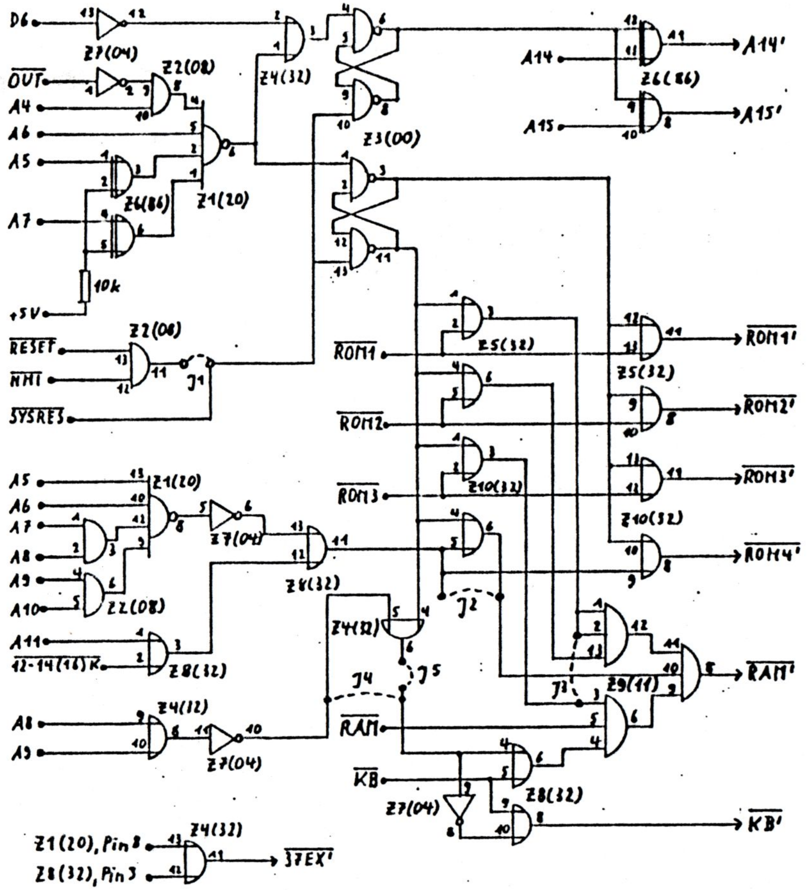


Abb. 4: Schaltung des CP/M-Boards

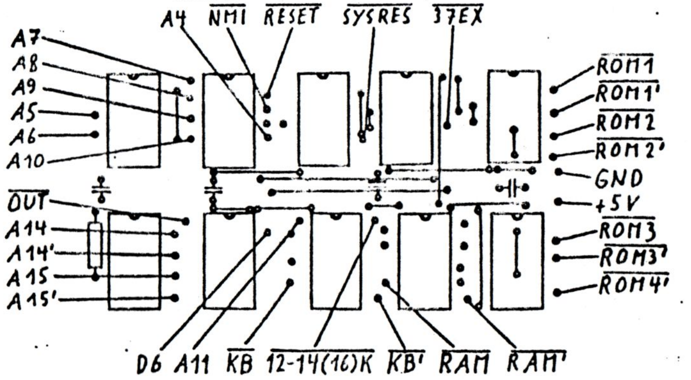


Abb. 5: Anschlüsse an das CPU-Board

BÖRSE -- BÖRSE -- BÖRSE

Eigentümer gesucht!

Auf dem Clubtreff in Idstein lieh ich mir eine **80 MICRO** Heft Juli 83 aus. - Der Eigentümer, dessen Name nicht im Heft ~~stand~~ steht, möge sich bitte melden, damit ich ihm das Heft zuschiden kann!

K.-J. Mühlenbein [06201/55052]

(traurig - aber wahr...)

ANZEIGE

Verkaufe
2x Model II
Heiko, 04109/1366

Folgende Dinge sind in meinem Inventar überflüssig und deshalb will ich sie losschlagen:

1 x **Diskettenlaufwerk**, einseitig, 40 Track, 2 Jahre alt, ohne Gehäuse und Stromversorgungsstecker. Preis: DM100.--

5 **Jahrgänge mc**: 1981...1985 jeweils als Bücher gebunden. Gute Verarbeitung, da von mir selbst gebunden Preis: DM25.--/Stück

1 x **Matrixdrucker**, Typ Heath H14. 8 Jahre alt aber sehr gut erhalten. Bisher keinerlei Ausfälle ! Serieller Anschluß 110...4800 Bd., mit Druckerkabel.

Wer interesse hat, rufe mich an oder schreibe mir. Ich bin abends ab 18.30 Uhr zu erreichen (am Arbeitsplatz nicht immer !)

Kurt Müller, Soltaustr. 24a, 2050 Hamburg 80, Tel: 040/7 24 60 83

- I M P R E S S U M -

1. Vorsitzende
Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

2. Vorsitzende
Bernd RETZLAFF
Kleiner Sand 98
2082 Uetersen
☎ 04122 /43551

HardwareKoordinator
Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

DiskotheKar
Club-Bücherei
Werner FÖRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion
Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren
Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck
Horst-Dieter Schroers
Breslauer Straße 9
8016 Feldkirchen
☎ 089 / 9032615

Das INFO erscheint zweimonatlich.

Es erfolgt Keine Zensur oder Kontrolle der jeweiligen eingeschicKten Infobeiträge durch die RedaKtion.

SCHLUSS

Hallo Club-80er,

Kurz bevor ich mich ein paar Wochen ins Ausland verdrücke, habe ich es doch noch geschafft, ein INFO zu erstellen. Ich hatte ja versprochen, gleich -nach Erscheinen des Letzten- mit der neuen INFO anzufangen.

Nach meinem Urlaub hoffe ich, daß wir endlich mal wieder die regulären Erscheinungstermine einhalten Können. Vielleicht Klappt es bis zum übernächsten INFO. Ich werde es auf alle Fälle versuchen.

Weiterhin Kamen Anregungen für ein neues Layout unserer Titelseite. Da wir ja nun ein gültiges Club-Emblem haben ist es richtig und sinnvoll, dieses auch auf der Titelseite zu verwenden. Leider ging es bei dieser Ausgabe noch nicht, da ich schon vorgearbeitet hatte. Ab Heft 26 haben wir dann ein neues "Outfit".

Bis dahin wünsche ich Euch noch eine schöne Zeit und unbegrenzte "Rechnerzeit".

Es grüßt Euch Euer

Jens

Edles aus ELCOMP

serviert von KJ

Müde, stets Gedrucktes zu lesen, lieber Leser? OK! Hier ist Geschriebenes aus erster !
[Sogar HRG: Himmlisch Rührende Grafik...]

Und ihr anderen, Enttäuschten! Habt Nachsicht – 7 (sieben) Jahre lang nahm mein "Model 1" einen wesentlichen Platz auf meinem Schreibtisch ein – viele Jahre lang verwandelten TSCRIPS + EPSON meine finsteren Gedanken in noch finstere Druckerschwärze, -- nun plötzlich ist der Platz leer, verwaist, traurig... und ich täusche euch mit einer Druckschrift, die doch nur eine -schrift ist, Marke kj.

„Ausgestiegen"? – Oh, no no no no no: Mein Chauvi - Peiniger - Unhold – kurz: CPU- befindet sich an einer ersten Adresse und wird z. Z. in einen Jungbronnen getaucht... Wenn er (sie) aus der Verjüngungskur wieder auftaucht: macht euch auf was gefaßt – (worauf, weiß ich selbst noch nicht!)

In der Zwischenzeit übe ich mich in der edlen Kunst des Schreibens – nachdem Ulrich so fleißig viele Artikel aus ELCOMP ab 1979 gesammelt und kopiert hat, war es meine verdammte Pflicht, diese ebenso fleißig auszumisten, zu ordnen und zu kommentieren. Nun schlummern sie in einer Mammelsappe bei Werner, dem Bibliothekar, und harren der Abrufe ...

Artikel aus ELCOMP [Stand: Mai 88]

Gruppe	Titel	Kurzkommentar/Verf.	Heft/Seite
CP/M	Das Standard-Interface für Microcomputer	← dto.	4/81/29-31
"	Erfahrung mit dBASE II	← "	6/84/56-57
EDITOR	XEDIT - Ext. Ed. f. TRS80/1	Leistet mehr als TANDY's Ed. [Wollschl.]	4/82/9-15
"	SCREDIT – ein Bildsch.-orientierter Ed. f. L.Z	[Heidenreich]	/83/42-46
FUNK	TRS80 u. Funkfernschreiben	Für Amateurfunker (m. Listing)	7/8/81/10-22
"	RTTY+CW mit dem TRS80	RTTY-Interface der Fa. Funk & Comp.	7/8/80/114
GRAFIK	Pixelgrafik in PASCAL	Grafik f. PASCAL m. TRS80 [Wollschl.]	9/82/30-33
"	Hardcopy m. TRS80/GENIE	Eine Bit Image Grafik m. allen Druckern	7/8/82/43-47
"	TRS80 – MX80 u. Grafik	Progr. f. d. Grafik auf dem APPLE	2/82/46-48
"	Hardcopy f. TRS-80-Grafik	Ein (langsamer!) Ersatz für JKL	11/12/80/103-105
HARDWARE	Motorsteuerung bei Minifloppies	(m. Schaltbild) [Keller]	-/83/486-487
"	Den Comp. richtig nutzen	Verwendung der RS232-Schnittstelle	4/5/84/88-92
"	Datenübertragung im Zeitmultiplexverfahren f. GENIE		7/8/82/76-81
"	EPROM für TRS80	mit BASIC u. M.-Programm → s. „Progr."	
MATHE	Berechng. generat. Funktion	Ein BASIC-Progr. hierfür	4/80/10-12
"	Suchlauf	Darstellg. von parametr. Kurven	3/79/6-12
"	Kurvenuntersuchg. m. PASCAL	Bestimmg. v. Extrema u. Nullstellen	../83/88-91
"	PRINTUSING als Rundungshilfe	Einfach „nur" ein Trick	1/82/46
"	Math. Funktionen f. PASCAL-Anfänger	(mit TRS-80) [Klaus]	-/83/104-108
"	TRS-80 als elektron. Kurvenlineal.	BASIC-Progr. zur Approximation	1/80/5-6
"	Iterative Verfahren	BASIC. – U. a. Lösung d. KEPLER-Glchg.	1/81/20-21
"	Ein Problem von Kopf um Kopf	-?→ eben etwas f. Köpfchen	7/8/82/119; 26-29
"	Die Mandelbrotmenge	Mathe + Kunst v. Comp. (Randstruktur)	3/4/86/9-14
"	Rundungsfunktion in BASIC	Minimiert Rundungsfehler	7/8/82/123
"	Kalenderrechnung	Ewiger Kal. v. 15.10.1582 – 22.1.3268 (!)	../../11-25

Gruppe	Titel	Kurzkommentar / Verf.	Heft/Seite
Masch.-Sprache	Z80-Assembler/Bsp.	- erklärt am nützl. Bsp. /Wollschl.	10/82/44-50
"	Programmieren in M.-Spr.	Kurs in 8 Teilen von Flögel	6/81-3/82
"	Relokatierprogr. f. Z80	Listing mit Flußdiagr. + HEX-Dump	11/12/81/42-44
"	OnePass 8080-Macroass. [MS]	für den TRS-80 / Matthaei	11/12/82/107-120
"	ACCEL3-„Comp. f. gehobene Ansprüche"	-	10/82/55-56
"	HEX-Binär-Wandler m. Z80	Ein bißchen Masch.-Spr. /Wollschl.	6/81/28-29
Mysteries	Vgl. CP/M u. MS-DOS	MS-DOS bei Dateien schneller; „Piping"	6/84/42
"	RAM + ROM im TRS-80	Das Programm „EXPLOR"/Wollschl.	11/12/82/38-48
"	BASIC hinter d. Kulissen geschaut	Was nie im Handbuch steht	4/82/41-42
"	DOS unter die Lupe genommen	" " " " "	4-8/82 [4 Teile]
"	Vgl. TRS-80 Monitore	Vgl. RSM-2 u. TBUG; Vorteile v. RSM	9/80/62-64
"	TRS-80 Tastaturspion	ASCII wird jedesmal in DEZ u. HEX gezeigt	7/8/80/4-5
"	Prüfsummen üb. ROM-Bereich	Zeigt Summen von je 256 B. ROM an	2/81/56
Programmierung	NEWDOS80: 3f. Dichte; 1,5x schnell	Erklärg. der Filetypen/Wollschl.	1/83/54-59
"	SWAP-virt. Vertauschen	numerischer Arrays/ Wollschl.	10/81/57-59
"	Chung/Yuen "tiny"-PASCAL	auf dem TRS-80, M.1	7/8/81/54-58
"	APL80-Interpreter f. TRS80	Progr.-Sprache f. TRS80 (3 Teile)	7-10/81
"	FLIST	Formatiertes Listing u.a. (Woll)	2/81/44-51
"	Kopplg. eines Masch.-Progr.	mit einem BASIC-Progr. (DATAs)	10/81/63-69
"	Sprungentf. b. rel. Adressg.	Ein BASIC(!)-Progr. z. Berechg.	3/79/15-16
"	Laufzeitreduzierung	beim BASIC-Interpreter	3/79/16-18
"	Neue BASIC-Befehle	5 neue Befehle (mit Listing)	4/5/84/46-49
"	Disk-BASIC in Level 2	Befehlserweiterungen f. L2 (Wollschl.)	7/8/81/45-48
"	Programmiertrick f. TRS-80	bei Kassettenbetrieb, - Vor- u. Nachteile	1/80/23-25
"	EPROM-Programmer f. TRS80	Schaltplan + Programme in BASIC u. M-Spr.	10/82/4-17
"	Programmieren in FORTH	Ein Kursus in 7 Teilen	6/82-6/84

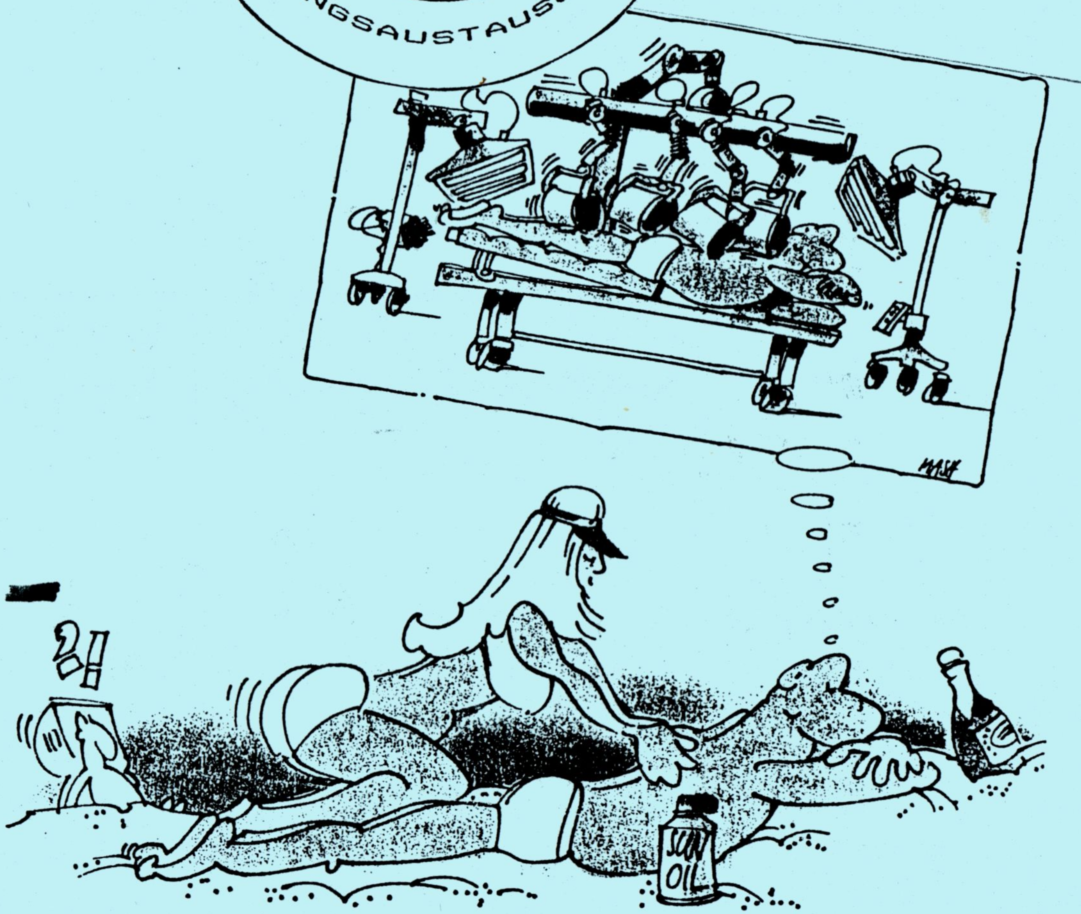
Gruppe	Titel	Kurzkommentar / Verfasser	Heft/Seite
Programmierung	BASIC-Monitor	System-Programme nach BASIC	3/81/34-37
"	BASIC-Monitor D	DATA ... für Masch.-Progr. in "	2/82/37-39
"	ABASIC	Rücksprung ins BASIC nach M.-Progr. [Wollschl.]	10/81/24
"	Schnelles Zählen in BASIC	z.B.: A+1 = -NOT A	10/80/14
"	Was ist FORTH?	Geschichte und Beschreibung	5/82/55-57
Utilities	Protokoll v. Bildschirmausg.	TRACE-Fkt. auch auf Drucker	5/80/21-22
"	Bildschirmausdruck o. Probl.	d.h. auch bei NMI! (z.B. Spiel)	6/84/53
"	Eingabe v. BASIC-Bef. m. SHIFT	Eine andere Art "Shorthand"	11/12/80/44-49
"	TRS80-READDISK/BAS	Lesen beliebiger, auch geschützter Sektoren	
"	Hardcopy Print f. TRS-80	Eine [notw.] Ergänzung f. Level II	11/12/79/112
"	Ein-/Ausgabeprogrammierg.	Hard- u. Softwarebeschreibung	10/79/1-5
"	TRS-80 als Timer/Wecker	M.-Progr. = 192 Bytes f. alle Zwecke! [Woll.]	7/8/82/32-34
"	Simul. v. 6502 auf TRS80/GENIE	Emulat. Progr. simuliert 6502-M.Proz.	7/8/82/10-15
"	Meldeton für TRS80	Schaltplan u. Listing (RS232 C-Schnittst.)	7/8/80/5-6

Platz für Nachträge:

CLUB 80

HARDWARE * SOFTWARE
COMPUTER-CLUB 80
TANDY-/ GENIE- ANWENDER
ERFAHRUNGSAUSTAUSCH

Clubinfo
der
TANDY - GENIE - und KOMTEK - ANWENDER



26. AUSGABE

KONTAKTADRESSE : CLUB 80 / GERALD SCHRÖDER / AM SCHÜTZENPLATZ 14 / 2105 SEEVETAL 1
TEL.: 04105 / 2602

- I N H A L T S V E R Z E I C H N I S -

Clubinternes

Neues vom Vorstand 01
Kurzbericht vom Nordlichtertreffen . 02
Gerald, Hartmut
Happy Birthday 03
Jutta Obermann
Aktuell und wichtig 04
Redaktion
Vorstellungen .... 05 - 09
Jörg, Christian, Sven, Peter, Detlev

Software

SORTICALC .... 11 - 14
Klaus-Jürgen Mühlenbein
Was man weis oder schon immer ...... 15 - 17
Dump 18
Lottozahlen .... 19 - 20
Kurt Müller
Atari-Bilder auf dem 4p .... 21 - 24
Hartmut Obermann
CP/M Warmstart ohne Nachladen ...... 25 - 27
Frank-Michael Schober
BDOS-Replcement, was'n das ? .... 27 - 28
Es war einmal ein Datum .... 29 - 30
Alexander Schmid
RPNL .... 31 - 35
Kurt Müller

Hardware

Besser erkennbare Darstellung von .. 37 - 38
EXP1-Floppy-Controller 38
Helmut Bernhardt
80/40 Trackumschaltung Mitsubishi .. 39
Hartmut Obermann
Entfernung der Kassettenschnittstell 40 - 41
Memory Mapped I/O (=) Port I/O GIIs. 42 - 45
Gerald Schröder
Darf's ein bisserl mehr sein 46
Alexander Schmid
Sharp 3541 .... 47 - 51
Artikel aus Computer Persönlich
Atari ST .... 52 - 58 (62)
Kurt, Gerald, Jens

Börse

Wer hat was -- wer will was .... 59 - 61
Werbung EDV-Verbrauchsartikel ...... 63 - 64

Sonstiges

Test: Brauche ich einen Computer ? . 10
Ein Segen für die Menschheit 36
aus "Ich hasse Computer"
Gebrauchtcomputermarkt TANDY & Genie 65
Hartmut Obermann
Wenn einer eine Reise tut 66
Alexander Schmid
dBase II, III und III Plus überblick 67 - 68
Artikel aus DOS 5'88
Auf gute Zusammenarbeit ! .... 69 - 70
Heinrich Betz
Entwicklungen ... und wie es weiterg 71 - 72
Paul-Jürgen Schmitz
Datenbanken richtig aufgebaut ...... 73 - 77
Artikel aus Window 9'88
Ein elektronisches Märchen 78
Dieter Kasper
Ein informatisches Märchen .... 79 - 80
Kostenlos telefonieren 80
Alexander Schmid
KI-Anwendung .... 81 - 82
Ein Wurm legte Tausende Computer lahm 83 - 84
Artikel aus Tageszeitungen

Club-80 Bibliothek

Neuzugänge 85
Abzugeben 85
Werner Förster

Die letzten Seiten

Impressum 86
Schluß 87
Mitgliederadressenliste .... am INFO-Ende
Redaktion

Extrablätter:
dBase II überblick und ALCOR-Pascal

Neues vom Vorstand

Hittfeld, 22.10.1988

Eigentlich wollte ich nur mal einen kurzen Besuch bei Gerald machen und den besonders guten Kuchen bei seiner Mutter genießen. Stattdessen sitze ich jetzt hier und bin dazu verdonnert, das Vorwort zum Info 26 zu schreiben (während sich Gerald hinter mir im Sessel räkelt und mir Stichworte zuruft).

Auch diesmal gibt es wieder Gutes und weniger Gutes zu berichten. Sehr gut ist, daß der Club in den letzten Wochen um 11 (in Worten: elf!) Mitglieder gewachsen ist. Besonders zu erwähnen sind in diesem Zusammenhang Gerald Dreyer und Heinrich Betz, die mit ihrer Werbeaktion erheblich an dieser Steigerung beteiligt waren.

Inzwischen ist Matthias Homann auch nicht mehr der einzige "nur-CP/M-User" im Club. Durch unseren während der Jahreshauptversammlung gefassten Beschluß, den Club reinen CP/M-Nutzern ohne Tandy-Ambitionen zu öffnen, gibt es inzwischen vier CP/M'er. Dabei sind natürlich die CP/M-nutzenden Tandy- und Genie-Besitzer, deren Zahl ständig steigt, nicht mitgezählt.

Jeder Neuzugang möchte natürlich so schnell als irgend möglich alle "alten" Infos haben. Dazu hat Jens Neueder vor einiger Zeit die Infos 1 bis 15 in Sammelordnern auf Karton zusammengefaßt. Diese Sammelordner kann jeder bei Werner Förster ausleihen. Die Sammelordner mit den Infos 16 bis 25 sind zur Zeit in Arbeit. In diesem Zusammenhang möchte ich einmal mehr auf die "Fundgruben"-Ordner hinweisen, die vor Jahren Günter Wagner zusammengestellt hat. In diesen hat er eine Unmenge Artikel gesammelt, die sich alle mit dem Thema Tandy TRS 80 bzw. Video Genie befassen. Gerade für Leute, die erst Mitte der 80'er Jahre zur Computerei gekommen sind, sind diese Ordner eine wahre Fundgrube. "Alte Hasen" finden darin z.B. viele Artikel aus der Zeit der ELCOMP und der guten Zeit der mc.

Da sich Gerald Schröder nicht mehr traut, Euch zu mehr Mitarbeit aufzufordern, bin wohl diesmal ich dran. Leute, wenn Ihr nicht mehr schreibt, wird unser Info bald nur noch aus der Titelseite, den Vorstellungen der Neuzugänge und dem Impressum bestehen. Dann wäre zwar die Vervielfältigung erheblich einfacher, das Lesen würde sich dann aber mit Sicherheit nicht mehr lohnen. Jens Neueder klagt schon, daß er trotz der geringen Auflage von nur vier Infos im Jahr 1988 fast keine Artikel mehr hat, um weitere Infos zu erstellen. Also schreibt, schreibt, schreibt!!! Schließlich wollen wir 1989, fünf Jahre nach der Gründung des CLUB 80, wieder sechs Infos auflegen, wie es sich gehört.

An dieser Stelle möchten wir uns noch bei Günter Wagner bedanken, der den CLUB 80 gegründet hat und uns Ende dieses Jahres verlassen wird. Ohne ihn wäre der Club nie ins Leben gerufen worden und hätte sich nicht zum heutigen Umfang entwickeln können. Vielen Dank an Dich, Günter!

Fröhliche Weihnachten und Frohes Neues Jahr wünschen

Hartmut und *Gerald*

Kurzbericht vom Nordlichtertreffen

Am Wochenende 19./20.11 fand in Heist (nördwestlich Hamburg) das zweite Nordlichtertreffen statt. Bitte fragt uns nicht, warum die Nordlichter schon das zweite Regionaltreffen durchführen, während die Südländer noch nicht mal ein einziges auf die Beine gestellt haben (ihr bekommt dann nämlich eine Antwort, die euch sicher nicht paßt)! Auf jeden Fall war es wieder mal ein Treff interessanter Computerfreaks und Hacker.

Es waren bei dem Treffen anwesend (zumindest körperlich):
Gerald Schröder, Bernd Retzlaff, Arnulf Sopp, Helmut Bernhardt, Kurt Müller, Jörg Brans, Gerd Rinio, Wilhelm Tornow, Christian Menk, Armin Ahlers. Als Vertreter aus dem Süden, last and least, Hartmut Obermann.

An Hardware waren zwei Model 4p, ein Genie IIIs, zwei Genie I, ein RTRS-80 und ein Toshiba T2100 (ratet mal, wem der gehört!?) vorhanden. Nicht vergessen sollte man natürlich die mechanischen und elektronischen Reparaturgeräte von Helmut, die auch ausgiebig gebraucht wurden. Zunächst mußte Arnulfs IIIs aus dem digitalen Nirwana zurückgeholt werden, dann hat sich das Bootlaufwerk von Jörgs 4p die Karten gelegt (es ist kaputt gegangen!!!) und mußte ausgetauscht werden. Für Bastelfreaks also genug Arbeit und für Hardwarelaien ein Anlaß zum Staunen.

Christian, unser z.Z. jüngstes Mitglied, hatte die Gelegenheit die Urahnen der heutigen PC's zu bestaunen. Ein Genie I ist für den Besitzer eines Sharp MZ 3541 ähnlich anzuschauen, wie eine JU-52 für einen Jetpiloten. Auch der Drahtverhau auf der Hauptplatine von Arnulfs IIIs ließ einige Ungläubige vermuten, daß die meisten Drähte nur Dekorationsfunktion hätten, was aber wirkungsvoll widerlegt werden konnte.

Am Sonntag konnten wir den Repräsentanten eines SVI-/MSX-Clubs (ca. 120 Mitglieder) begrüßen. Das Format ihrer CP/M-Disketten war schnell geknackt (Bericht in diesem oder dem nächsten Info) und somit steht einer weiteren fruchtbaren Zusammenarbeit nichts im Wege. Der Club gibt vier Infos im Jahr heraus, wobei wir einen Austausch der Artikel über CP/M anstreben.

Auch wenn das Regionaltreffen, im Gegensatz zur Jahreshauptversammlung recht kurz war, halten wir solche Treffen für sehr sinnvoll. Nur auf solchen Zusammenkünften können sich die Mitglieder persönlich kennenlernen, und der Informationsaustausch gestaltet sich mit Sicherheit im persönlichen Gespräch einfacher, als durch noch so umfangreichen Briefwechsel. Wir können den Clubmitgliedern aus dem Süden also nur raten, sich auch einmal aufzurappeln und ein Regionaltreffen zu organisieren!

In diesem Sinne alles Gute und ein bißchen mehr Initiative,

Der Vorstand*

*(Coproduktion Gerald & Hartmut)

HEFT 26 Januar 1989

Hallo meine lieben " Clubfreunde ",

Nach Erscheinen des letzten Infos habe ich mich gleich darangesetzt, um Euch wieder zu berichten, wer alles in der nächsten Zeit Geburtstag hat. In der Hoffnung, daß dieses Info rechtzeitig bei Euch ankommt, berücksichtige ich dieses Mal nur 3 Monate, anstatt wie letztesmal, 4.

Wie Ihr unten sehen könnt, sinkt die Geburtenrate in den Wintermonaten rapide ab. Woran das liegen mag ?

Wie üblich, appeliere ich an Euch, mir Bescheid zu geben, wenn ich jemanden vergessen habe. Ich entschuldige mich deshalb schon im voraus bei den Betroffenen.

Diesmal sind die Monate November, Dezember und Januar dran.

" Novembergeborene "

02.11.	:	Krispin Michael
03.11.	:	Hentz Wolfgang
07.11.	:	Mühlenbein Klaus-Jürgen
08.11.	:	Schäfer Walter
28.11.	:	Neueder Jens

" Dezembergeborener "

03.12.	:	Rinio Gerd

" Januargeborene "

09.01.	:	Seelmann-Eggebert Jörg
17.01.	:	Betz Heinrich

Im Namen des Vorstandes gratuliere ich Euch recht herzlich zum Geburtstag und wünsche Euch alles Gute und weiterhin viel Spaß am Computer.

Ein besonders herzlichen Glückwunsch geht an unseren Inforedakteur Jens, dem ich mehr Artikel für das Info wünsche. Deshalb meine Bitte an Euch, die Arbeit des Clubs und des Vorstands mehr zu unterstützen (z.B. mit Artikeln usw.).

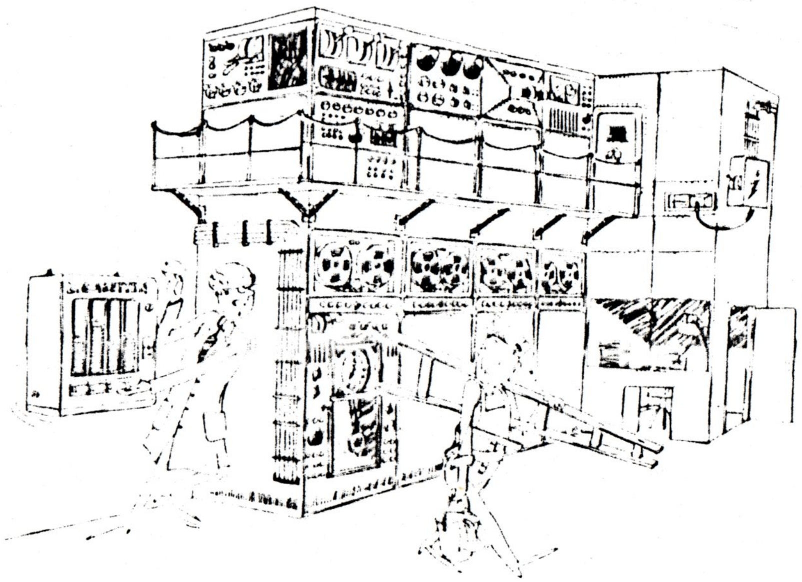
Jutta

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....ca. 3 Wochen nach Erscheinen dieses INFO's
Jahreshauptversammlung ....29. April - 1. Mai 1989

-- Messen '89 --

CeBIT ....Hannover 8. - 15. März 1989
Internationale Funkausstellung 1989 ....Berlin 25. August - 9. September
PRODUKTRONIA ....München 7. - 11. November 1989



»Hallo, Sie – wie funktioniert dieser komplizierte Apparat?«

Hallo liebe Clubmitglieder,

ich möchte mich als neues Clubmitglied kurz vorstellen. Ich bin 45 Jahre alt, von Beruf Steuerbeamter, und arbeite (dies ist kein Schreibfehler) als Prüfer seit Dezember 1984 mit einem TRS 80 Model 4P. Das Gerät ist mit einem Z80 H 8Mhz Microprozessor und 512 K ausgerüstet. Weiterhin besitze ich 4 Laufwerke, wovon 3 von 80 auf 40 Spuren umschaltbar sind. An Druckern stehen mir folgende Geräte zur Verfügung: DMP 430, HP Desk Jet, Silver Reed EX 32. Wie Ihr seht, habe ich eine ganze Menge in's Geschäft gesteckt. Die Hauptanwendung des Gerätes liegt bei Textverarbeitung (deutsche Version von Lescript / Vers. 1.70G und 1.80G) und Berechnung von Steuern (Lohn bzw. Einkommensteuerprogramme).

Durch meine Mitgliedschaft erhoffe ich mir Kontakte zu anderen Anwendern zu bekommen, und meine bescheidenen Programierkenntnisse in Basic zu erweitern. Ich selbst kann Euch unter Umständen mit umfangreicher Literatur aushelfen, da ich sehr viele Zeitschriften besitze. Dies sind u.a. C't (komplett), 80 Micro (ab 1983), Computer persönlich (ab 83) Byte usw. usw.. Desweiteren besitze ich recht gute Kenntnisse über den amerikanischen Markt, und habe selbst schon sehr viel in Amerika per schriftlicher Bestellung (mit Visa Card) gekauft: Bisher habe ich nur die allerbesten Erfahrungen gemacht. Mit der Hoffnung, daß ich bald einige von Euch kennen lerne, verbleibe ich

Mit freundlichen Grüßen

.... Jörg Brans

Hallo Clubmitglieder!

Hiermit möchte ich mich gerne vorstellen.

Ich heiße Christian Menk und wohne in Hanstedt (bei Hamburg). Ich bin 15 Jahre alt und besuche die 10. Klasse der Realschule. Ich werde voraussichtlich im Sommer 89 eine Lehre als Reiseverkehrskaufmann beginnen.

1982 bekam ich zu Weihnachten einen TI-99/4A von meinen Eltern und haben dann angefangen in Basic zu programmieren. Letztes Jahr habe ich mir dann von meinem Konfirmationsgeld einen SHARP dazugekauft. Seitdem stöbere ich ein wenig in CP/M herum. Meine Programmierfähigkeiten belaufen sich z. Zt. auf xxxxBasic(MBASIC 80) und ein bischen Pascal (TURBO Pascal). Im Moment arbeite ich an einem Programm mit dem man teilen kann. Allerdings soll die Teilaufgaben bis zu Ergebnisse errechnen können die über 100 Kommastellen haben.

Mein Interesse geht zusätzlich noch zu Assembler und PL/I.

Bis dann

Christian Menk

Hallo Clubfreunde,

als vierter User eines reinen CP/M-Rechners möchte ich mich kurz vorstellen.

Mein Name ist Sven Krahn, geb. am 05.04.1969 in Hamburg, wohnhaft in Villingen. Im Mai habe ich das Abitur gemacht und leiste seit Oktober meinen Wehrdienst ab. Danach habe ich vor, Wirtschaftsingenieurwesen zu studieren.

Mein Interesse für Computer begann vor etwa 10 Jahren. Nach verschiedenen Jobs und ständigem Hacken bei Freunden konnte ich mir 1984 endlich einen eigenen C64 leisten, mit dem ich Programme zuerst in Basic, dann in 6502-Maschinensprache schrieb. Kurz vor dem großen Preisrutsch konnte ich die ganze Anlage aber wieder losschlagen und mir statt dessen einen Olivetti M10 (Handheld) kaufen, der auch heute noch in meinem Besitz ist. (Der TRS80-M100 ist übrigens fast baugleich zum M10).

Zu CP/M kam ich dann durch Zufall. In einem Baseler Computershop konnte ich dem Verkäufer ein Gerät, das seit Wochen in der Reparatur stand, als nicht mehr reparierbar abschwatzen. Nach einiger Suche wechselte ich einen Kondensator aus und kam so für 50 Pfennig zu einem Computer. Nun brauchte ich noch ein Terminal dazu, denn nur mit einer Blechkiste ließ sich nicht allzuviel anfangen. Nach einigen Monaten fand ich dann ein gebrauchtes, das finanziell meinen Erwartungen entsprach, die ganze Anlage sieht somit ziemlich antik aus. Es handelt sich um einen Micro Decisions von Morrow Designs (2 Laufwerke SS/DD 48tpi) unter CP/M 2.2 mit einem Terminal TAB 132/15. Aufgrund der immer weniger werdenden Information über CP/M bin ich noch nicht sehr weit ins System eingedrungen, ich programmiere hauptsächlich in Basic, Pascal und C, versuche mich auch mal in Assembler, meistens aber auf meinem Handheld. Vielleicht ändert sich das ja jetzt...

Meine Interessen liegen bei allem, was sich mit einem Computer machen läßt, sei es hard- oder softwaremäßig, system-orientiert oder Hochsprache, Elektronik oder Anwendungsprogramm.

Ich hoffe, daß wir gemeinsam das aussterbende Betriebssystem CP/M noch eine Weile am Leben erhalten können.

Euer

Sven

Liebe Clubfreunde!

Seit 1980, also zu einer Zeit, als der TRS-80 mit 16 K noch ziemlich teuer war, besitze ich einen solchen Computer. Angefangen hat es damit, daß ich mich (teils berufsbedingt) für Textverarbeitung interessiere und mir ein Videoterminal mit der ELTEC-Karte baute. Die Tastatur hatte ich sehr preiswert im DARC (Deutscher Amateur Radio Club) bekommen. Aufgezeichnet wurde mit Kassette in Schreibgeschwindigkeit.

Da ich Übersetzungen aus dem Amerikanischen mache, war die fehlende Möglichkeit des Verschiebens von Wörtern, um bei schwierigen Texten einen einigermaßen lesbaren deutschen Satz zu erhalten, doch recht hinderlich. Das brachte mich zum TRS-80. Nicht aus Überzeugung, damals war der PET sehr verbreitet, sondern durch Zufall.

Nach Jahren der Kassettenaufzeichnung faßte ich mir ein Herz und kaufte eine leere Dr. Aumann-Floppykarte, nachdem ich vorher Kleinbuchstaben aufgerüstet und auf 48 K aufgebohrt hatte. Dann lag die Karte aber noch einige Monate herum, ehe ich sie mit den Bauteilen versah. Zuerst benutzte ich drei 40-Spur-Geräte, seit 1987 2 x 40 SS/SD und 1 x 80 DS/SD. Mit Platz für Erweiterungen in einem selbstgebauten ca. 4 Höheneinheiten großen 19-Zoll-Gehäuse. 1986 kam ein CITIZEN 120-D dazu.

Da ich den TRS-80 auch geschäftlich, in erster Linie jedoch als Hobby benutze, erhoffe ich mir vom Beitritt zum Club Unterstützung in Fällen, in denen ich mit meinem Latein am Ende bin (und das ist oft schon ziemlich schnell).

Durch den DARC habe ich Kontakt zur Arbeitsgemeinschaft Microcomputer und mehrmals hervorragende technische Hilfe bekommen. (Die AG Microcomputer hat den TRS-80 viele Male auf 6 Europakarten nachgebaut, eine Version kenne ich mit 7,5 M Festplatte).

Auf gute Zusammenarbeit!

Da(s) bin ich !(:)

48-jährig, verheiratet, zwei Kinder (m/12,w/3) und nun seit kurzer Zeit Mitglied im **CLUB 80**.
Mein "Maschinenpark" umfaßt zur Zeit:

- 1 **SANYO MBC-1150 (CP/M 2.2, 2x320k)**
- 1 ATARI 1040 STM (TOS, 1x720k(940k),nur Farbmonitor)
- 1 ATARI 600 XL (2x1050, 64k-Erweiterung, Centr.-Schnittst.)
- 1 ATARI 800 XL (2x1050, Centronics-Schnittstelle, Datasette)
- 1 TANDY TRS80 COLOR-COMPUTER II (ohne Peripherie)
- 1 SINCLAIR ZX81 (16k-Erweiterung, Centronics-Schnittstelle)
- 1 9-Nadel-Drucker PANASONIC KX-P1083
- 1 9-Nadel-Drucker PRÄSIDENT 6313 (z.Z. nur 8 Nadeln)
- 1 9-Nadel-Drucker CP80-SHINWA (Melchers)
- 1 9-Nadel-Drucker STAR 8240
- 1 Thermo-Transfer-Printer Brother HR5

An Programmiersprachen übe ich

- BASIC, Sanyo-BASIC, GFA-BASIC (V2.2 und V3.0), GW-BASIC
- Turbo-PASCAL (V2.0)
- LOGO
- PROLOG
- FORTH 83
- XLISP (V1.4, V1.5)

Fit bin ich in keiner !

Regelmäßig* arbeite (spiele) ich mit folgenden Programmen:

WORD-STAR	(SANYO)	1ST-Word-plus	(ATARI ST)
dbase II	(ATARI ST, SANYO)		
Print-Shop	(ATARI XL)	Print-Master	(ATARI ST)
FS2	(ATARI ST, ATARI XL)		
CALC-STAR	(SANYO)	Multiplan	(SANYO)
MS-DOS und CP/M	(ATARI ST)		

Für mich ist die Computerei kein Mittel des Broterwerbs sondern sie dient nur der (so vorhanden) Freizeitgestaltung.

Beruflich friste ich mein Dasein als Angestellter der Stadt. (Hier habe ich auch noch Gelegenheit, ein bißchen unter den SIEMENS-Betriebssystemen BS2000 und SINIX herumzufummeln.)
Bevor ich nun mit meiner "SELBSTDARSTELLUNG" das **CLUB 80**-Info in meine Biografie umfunktioniere, möchte ich an dieser Stelle (unter Hinweis auf meinen Briefkasten bei Interesse für weitere Details mich betreffend) diesen Beitrag beenden.

Mit abgebrochene Fingernägel vom Tastenquälen
rotgeränderten Augen vom Debuggen
leichten Magenschmerzen vom zu starken Bohnenkaffe und
freundlichen Grüßen an alle **CLUB 80**-Mitglieder sowie
aufrichtiger Anteilnahme am "Leid" deren Familienangehörigen

Peter Schröder

* = regelmäßig ist auch 1x im Vierteljahr !!

Hallo Clubfreunde,

als neues Mitglied möchte ich mich hiermit vorstellen.

Mein Name ist Detlev Bröcker, ich bin 41 Jahre alt und wohne in Düsseldorf. Von Beruf bin ich selbständiger Elektromeister.

Die Computerei betreibe ich seit 1982 mit einem TRS 80 M1. Mit der Zeit kamen das Expansions-Interface, 3 Floppy Laufwerke und ein Epson MX80 dazu. Zuletzt wurde noch ein TRS 80 M3 angeschafft, der aber nur geschäftlich genutzt wird.

Anfangs als Hobby gedacht, hat sich die ganze Sache auch auf den beruflichen Bereich ausgedehnt, wo mir das M3 gute Dienste leistet.

Ich erledige damit die Buchhaltung, die Adressverwaltung, das Rechnungschreiben und natürlich die Textverarbeitung.
Die Software ist zum größten Teil von mir selbstgeschrieben (in Basic), wenn Interesse besteht, stelle ich sie gerne der Club-Bibliothek zur Verfügung.

Hardwaremäßig bin ich auch stark interessiert und habe mich auch schon in der Theorie stark gemacht.
Doch nachdem ich eine Maschine in die ewigen Jagdgründe geschickt hatte und das dazugehörige Expansions-Interface gerade noch retten konnte, ist mein Selbstvertrauen in dieser Richtung stark erschüttert worden.

An einem Erfahrungsaustausch auf diesem Gebiet bin ich auf jeden Fall auch interessiert.

In diesem Sinne

Detlev

Test: Brauche ich einen Computer?

Wie finde ich eigentlich heraus, ob ich einen Computer brauche? Die Antwort auf diese Frage ist von vielen Faktoren abhängig.
Das Meinungsforschungsinstitut »Streng objektiv« hat folgenden Test zusammengestellt, mit dem jeder seine Computerbedürftigkeit überprüfen kann.

1. Ihr Beruf:

freier Künstler . . . . 0 Punkte
selbständiger Geschäftemacher . . . . . 5 Punkte
Politiker . . . . 100 Punkte

2. Ihre Ziele:

reich werden . . . . 100 Punkte
reich bleiben . . . . 5 Punkte
alles Scheiße . . . . 0 Punkte

3. Ihre Träume:

ein Morgen ohne Brummschädel . . . . . . 0 Punkte
eine Nacht mit Ornella Muti . . . . . . 50 Punkte
einen Tag mit von Brauchitsch . . . . . 100 Punkte

Auswertung:

Bis 0 Punkte: Gratuliere! Sie haben mit Ihrem ehrlichen und bescheidenen Leben bewiesen, daß man auch ohne Elektronik ein glücklicher Mensch werden kann. Gegen Kopfschmerzen ist kein Chip gewachsen.

Bis 200 Punkte: Eijeijei. Warum denn? War doch bisher alles paletti. Der Zaster stimmt und der Daimler steht in der Garage. Was machen da schon die Kopfschmerzen, solange sie von nichts Schlechterem als Mädels und Champagner kommen. Für Sie wäre ein Computer nur Zeitverschwendung. Träumen Sie lieber nicht von der tragbaren Olivetti, sondern lieber von Ornella Muti.

Über 200 Punkte: O je, Sie sind gefährdet. Als Politiker sollten Sie aber nicht so auf den Putz klopfen. Daß Sie reich werden wollen, nimmt Ihnen ja keiner übel, und mit einem geschickten Händchen sollte da was drin sein. Aber schlagen Sie sich das Treffen mit von Brauchitsch aus dem Kopf. Der hat doch seine Spendierhosen längst wieder ausgezogen. Sie sind maßlos und brauchen keinen Computer, sondern einen Psychiater.
Tip: Benutzen Sie mal den Behörden-Computer. Mit all den vielen schönen Daten müßte doch was zu machen sein, das einen weiterbringt.

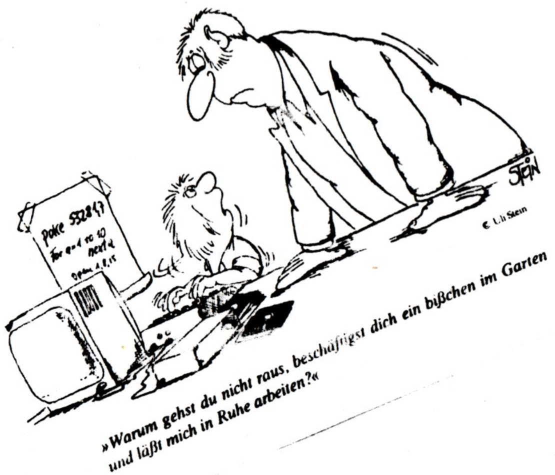
Dieser Test beweist ganz eindeutig: Es gibt im Grunde genommen keinen Menschen, der einen Computer wirklich braucht. Dafür gibt es unzählige Menschen, denen Computer ein Dorn im Auge sind. Denken wir nur an die Terroristen oder die Opfer der Steuerfahndung.
Natürlich gibt es Ausnahmesituationen, in denen ein Computer gewisse Vorteile bietet. Zum Beispiel beim Ordnen Ihres Telefonverzeichnisses, wenn die Zahl der Eintragung über 10000 liegt. Oder wenn Sie eine Raumfähre steuern oder auch nur einen kleinen Erstschlag mit taktischen Atomwaffen führen wollen. Für solche Fälle sind die elektronischen Helfer wirklich unentbehrlich.

Ein neuer Filename? Zält mal die Buchstaben!
Alle Neune?! Also kein Filename.
Nur eine "Utility". (Und immer noch nicht in Maschinensprache!)
Natürlich für Freunde von VISICALC, wie der Name vermuten lässt.
Hat VC (VISICALC) nicht schon alles, was man braucht? Besonders in der 3.Version (die sich leider durch einen stark eingeschränkten Speicher "auszeichnet"...) oder in der Modifikation "VCMOD/CMD", die das Herz jedes unverCALCten VISIonärs höher schlagen lässt...?
In der Tat: VISICALC kann viel und steht seinem grossen Bruder MULTIPLAN nur wenig nach. Aber:
Eine einzelne Zeile oder Spalte s o r t i e r e n - das geht nicht; oder gar das komplette Tabellenblatt in die alfabetische oder numerische Reihenfolge gemäss einer bestimmten Zeile bzw. Spalte bringen. Kann es nicht. Eine echte Lücke!
In die bin ich schon einmal hineingesprungen (s.INFO 19, Juni 1987, Seite 29)
Dabei hab' ich mir zwar nicht gerade das Genick gebrochen, aber einen Fuss verstaucht. Unter bestimmten Bedingungen oder unglücklichen Konstellationen klappte es nämlich nicht mit diesem Programm, das unser Bibliothekar *Werner* in seiner Bekanntgabe neuer Programme versehentlich mit "VCMOD/CMD" bezeichnet hat (INFO 21, September 1987, Seite 65). Das merkte ich neulich, als es darum ging, eine Tabelle, die im Laufe des Monats September stochastisch angefallen war, am Monatsende nach der Spalte zu ordnen, in der die Namen meiner Schüler standen. Also hiess es: Ran ans Werk und überarbeiten. Aber daraus wurde ein völlig neues Programm, das ich hiermit vorstelle und demjenigen, dem die Zeit des Abklopfens mit Recht zu schade ist, gern per Disk z.V. stelle. (Alle Formate.)
Voraussetzung für die Anwendung ist, das das (habt ihr's schon gemerkt? Ich befleissige mich bereits der neuen "Rechtschreibung", die z.Z. hier in Mannheim im "Institut für Deutsche Sprache" ausgebrütet wird, um uns das künftige Leben zu erleichtern... Wahrscheinlich habt auch ihr schon in der Zeitung davon gelesen! ss statt ß; das statt daß (wie logisch! Eigentlich müßte aus dem 'daß' ein 'dass' werden...) - also: Voraussetzung ist, das das zu sortierende File im sog. 'DIF-Format' (DATA INTERCHANGE FORMAT) vorliegt.
Um es nicht zu kompliziert zu machen, sortiert das Programm nur Spalten. Will man Zeilen sortieren, kann man die Tabelle im DIF-Format bekanntlich mit der C-Option leicht diagonal spiegeln!
Eine weitere Voraussetzung ist, das die Spalte, nach der sortiert werden soll, entweder nur numeri (Zahlen) oder nur strings (Wörter) enthält! Auch das ist eigentlich selbstredend - denn eine Datenfolge wie: "9640 Weinheim Am Mönchgarten 28" nach Zahlen sortieren zu wollen ("28 9640 Am Mönchgarten Weinheim") wäre ebenso sinnlos wie die Sortierung nach strings (die übrigens ganz genauso aussehen würde, da nach ASCII-Werten sortiert wird!)
Keiner hat das vor.
Wenn man nach einer Spalte sortieren will, in welcher Zahlen und Wörter gemischt sind - weil z.B. in ihrem Kopf eine Überschrift stet oder an ihrem Ende das Wort "Summe" u.ä. - so kann man den "astreinen" Teil der Tabelle, wie menniglich geleufig und gemeinhin bekannt, mit der #-Option aus der Tabelle herausschneiden, partiell speichern und dann mein Programm


```
10 '        ***********************************************************
20 '        *                  V I S I C A L C -                      *
30 '        *    Datei im Data Interchange Format umordnen            *
40 '        *    <C> Klaus-Jürgen Mühlenbein, Weinheim 10/88          *
50 '        ***********************************************************
60 CLS:CLEAR25000:PRINTSTRING$(59,"*")
70 PRINT"*                          V I S I C A L C -
  *
* Datei im Data Interchange Format umordnen und speichern *
80 PRINT"*
  *
90 PRINT"*          <C> K.-J.Mühlenbein, Weinheim, Oktober 1988
 *
100 PRINTSTRING$(59,"*"):PRINT
110 DEFINTI,S,T,X,Y,Z:DEFSTRC,F,K,V,W:DIMK(16)
120 PRINT:PRINT"Es koennen nur Spalten sortiert werden!"
130 PRINT"(Ggf. mit der C-Option umformen.)"
140 PRINT"Die Spalten der Tabelle duerfen jeweils
entweder n u r  aus Zahlen oder n u r aus Woertern bestehen!
(Andernfalls den gewuenschten Tabellenausschnitt als Hilfsdatei
im DIF-Format speichern!)"
150 PRINT"Bei Zahlenspalten darf die erste Zeile nicht leer sein
!
(Ggf. diese Felder mit '0' belegen)":PRINT," E N T E R !
160 Y$="":Y$=INKEY$:IFY$="",160
170 REM ****************** Kennsatz lesen ******************

180 PRINT"Welche DIF-Datei soll sortiert werden (Eingabe ohne Extension)":
[redacted] INPUTF:F=F+"/DIF"
190 OPEN"I",1,F
200 FORI=1TO16:INPUT#1,K(I):NEXT:Z=VAL(K(7)):S=VAL(K(11))
210 CLS:PRINT"Die Tabelle hat"S"Spalten und"Z"Zeilen":DIMN(Z,S),
W(Z,S),T(S),II(Z),N1(Z),W1(Z)

220 REM **************** Datei lesen *****************

230 FORIS=1TOS:INPUT#1,Y1,Y2,Y3,T(IS)
240 ONT(IS)+1GOSUB350,400
250 NEXTIS:CLOSE
260 CMD"lc,n":INPUT"Welche Spalte soll sortiert werden <Buchstab
en A...BK) ";C:CMD"LC"
270 CL=LEFT$(C,1):CR=MID$(C,2,1)
280 SO=-(CR="")*(ASC(C)-52)-12
290 IFCR<>"":SO=SO-(CL="A")*(ASC(CR)-26)-(CL="B")*ASC(CR)
300 INPUT"von Zeile X bis Zeile Y  <X,Y> ";X,Y
310 INPUT"Sollen die uebrigen Spalten mitsortiert werden <J/N>";
S$:SA=INSTR("JjNn",S$)

320 REM *********** Sortierschluessel definieren *************

330 ONT(SO)+1GOTO440,470
340 '
350 FORIZ=1TOZ
360 INPUT#1,N(IZ,IS):IFIZ<Z,GOSUB390ELSEINPUT#1,V
370 NEXTIZ
380 RETURN
390 INPUT#1,V,T:RETURN
400 FORIZ=1TOZ
410 INPUT#1,N,W(IZ,IS):IFIZ<Z,GOSUB430
420 NEXTIZ:RETURN
```

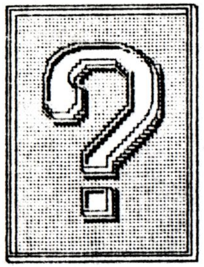
darauf anwenden. Anschliessend baut man es – wieder mit der #-Option – wieder in die ursprüngliche Tabelle ein. Das alles ist jedenfalls wesentlich weniger anstrengend als die müselige "MOVErei" unter Beachtung einer alfabetischen Reihenfolge!
Man kann mit dem Programm, wie gesagt, eine Spalte allein ordnen, wobei die Reihenfolgen in den übrigen Spalten erhalten bleiben, oder wahlweise die Zeilen aller übrigen Spalten mit der Spalte, nach welcher sortiert wird, "mitgehen" lassen. Der Trick dabei ist, das im ersten Fall die sog. "direkte Sortierung" <CMD"O",n,A(1)>, im anderen Fall die sog. "indirekte Sortierung" <CMD"O",n,*IX(1),A(1)> angewendet wird.
In der Tat: Ein mächtiger NEWDOS-Befehl.
Weshalb ich dieses Betriebssystem so ungern verlasse.
Oder weiss jemand ein besseres?
(Das das das kann, ist fenomenal...)

Rafol




```
430 INPUT#1,T:RETURN
440 FORIZ=XTOY:N1(IZ)=N(IZ,SO):NEXT
450 CMD"O",Y-X+1,*II(X),N1(X)
460 GOTO520
470 FORIZ=XTOY:W1(IZ)=W(IZ,SO):NEXT
480 CMD"O",Y-X+1,*II(X),W1(X)
490 '
500 REM ########### Abspeichern der geänderten Datei ##########
510 '
520 CLS:INPUT"Neuer Filename (ohne Extension und Laufwerk)";FF:I
NPUT"In welchem Laufwerk speichern ";FW:FF=FF+"/DIF:"+FW
530 WW(0)="0":WW(1)="1":WW(3)="-1"
540 OPEN"O",1,FF
550 '
560 REM ### KENNSATZ ###
570 '
580 PRINT#1,"TABLE"
590 PRINT#1,WW(0);",";WW(1)
600 PRINT#1,CHR$(34);CHR$(34)
610 PRINT#1,"VECTORS"
620 PRINT#1,WW(0);",";STR$(Z)
630 PRINT#1,CHR$(34);CHR$(34)
640 PRINT#1,"TUPLES"
650 PRINT#1,WW(0);",";STR$(S)
660 PRINT#1,CHR$(34);CHR$(34)
670 PRINT#1,"DATA"
680 PRINT#1,WW(0);",";WW(0)
690 PRINT#1,CHR$(34);CHR$(34)
700 '
710 REM ### Aufbau des VISICALC-Files ###
720 '
730 FORIS=1TOS
740 PRINT#1,WW(3);",";WW(0)
750 PRINT#1,"BOT"
760 GOSUB830
770 NEXTIS
780 PRINT#1,WW(3);",";WW(0)
790 PRINT#1,"EOD"
800 CLOSE
810 PRINT:PRINT:PRINT"DOS <0> oder BASIC <ENTER>
820 Y$=INKEY$:IFY$=""THEN820ELSEIFASC(Y$)=13THENENDELSECMD"S
830 IFX>1:FORIZ=1TOX-1:ONT(IS)+1GOSUB890,910:NEXTIZ
840 FORIZ=XTOY:IFT(IS)=1THEN930
850 PRINT#1,WW(0);",";:IFSA<3ORIS=SS,THENPRINT#1,N(II(IZ),IS)ELS
EPRINT#1,N(IZ,IS)
860 PRINT#1,"V"
870 NEXTIZ:IFY=Z,RETURN
880 FORIZ=Y+1TOZ:ONT(IS)+1GOSUB890,910:NEXTIZ:RETURN
890 PRINT#1,WW(0);",";N(IZ,IS)
900 PRINT#1,"V":RETURN
910 PRINT#1,WW(1);",";WW(0)
920 PRINT#1,W(IZ,IS):RETURN
930 PRINT#1,WW(1);",";WW(0)
940 IFSA<3ORIS=SS,THENPRINT#1,W(II(IZ),IS)ELSEPRINT[illegible],IS)
950 GOTO870
```

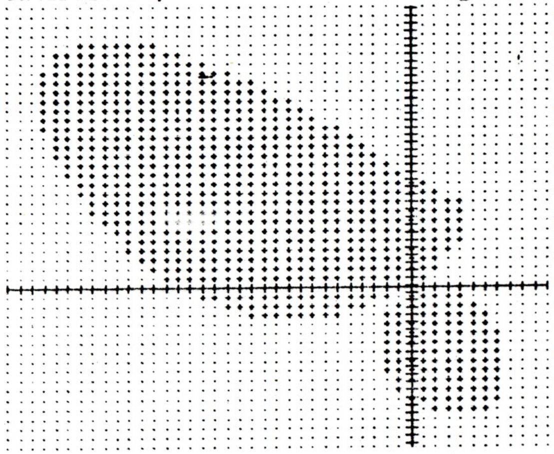
Was man weis oder schon immer wissen wollte



"Seit Anbeginn der Zeit wartet er nun, daß..." - Ähem..., nun denn. Wir haben keine Zeit zu warten. Für unsere Probleme werden andere Mittel benötigt als gerade "Warten". So etwas erzeugt nur Schwielen am Hin..., ja und außerdem gerät man schnell ins Hintertreffen. Deshalb empfehlen wir einen Computer als die ultimative Lösung Ihrer Probleme.

Obwohl dieses Programm ist nicht auf meinem Mist gewachsen ist, will ich es dennoch hier wiedergeben. Vorweg erst einmal die Quellenangabe: CHIP No. 1, 1989, Seite 235. Der Autor ist Thomas Wolf.

So, und nun zur Sache. Das Programm erscheint mir deshalb interessant, weil es recht elegant implizite Funktionen darstellt.



Dabei bedient es sich der Berechnung eines Punktrasters, welches die Umrisse des Funktionsgraphen abtastet und durch eine Fallunterscheidung (F(x,y) < 0) entweder einen Punkt oder ein Kleines Kreuz ausgibt. Auf diese Weise entstehen zwei Flächen, deren Berührungskanten den Funktionsgraphen abbilden. Zur Verdeutlichung ist das Rechenergebnis der im Programm per "DEF FN" eingebauten impliziten Funktion wiedergegeben. Die bei der Berechnung benutzten Koordinatengrenzen waren:

Xmin = -10 Ymin = -10 Pixelabstand = 5
Xmax = 10 Ymax = 10

Und die Funktion:

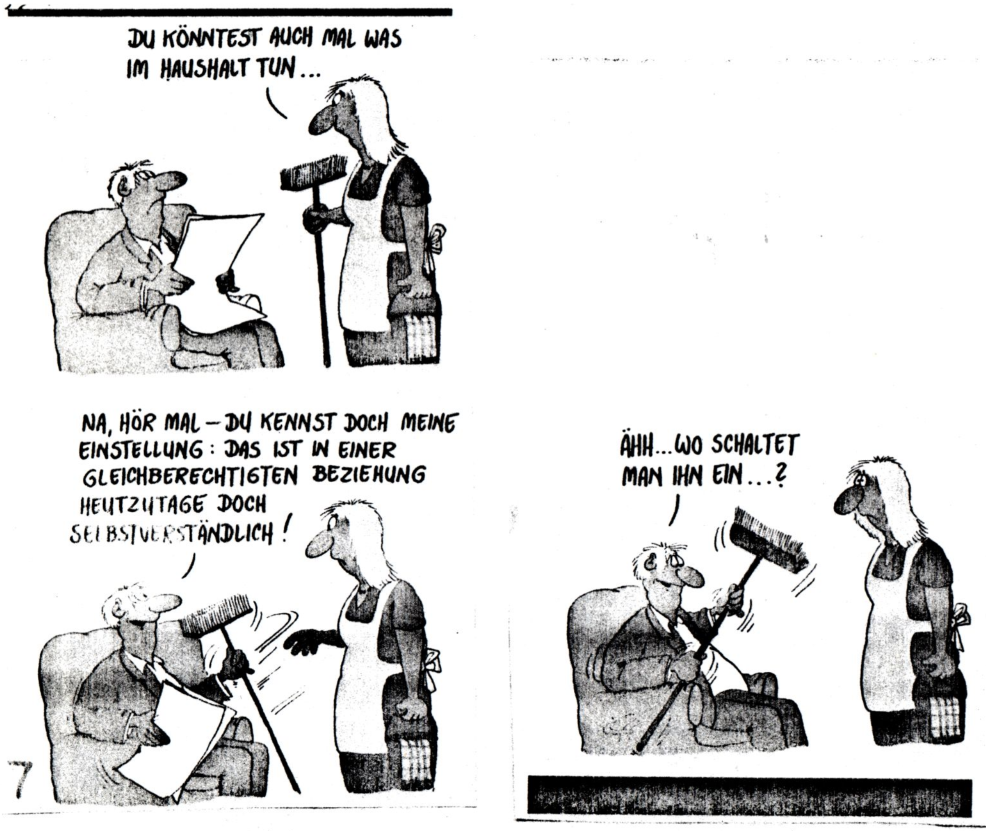
f(x,y)=9*(x-y)*(x-y-x*x)-3*x^4-y^4

Die Rechenzeit liegt bei etwa 30 Sekunden. Auf einem Genie I schätze ich so um die 5 Minuten.

Treiß Dirk


```
'
' Darzustellende Funktion
'
DEFFN f(x,y)=9*(x-y)*(x-y-x*x)-3*x^4-y^4
'
' Umrechnung der X- und Y-Koordinaten als Funktion
'
DEFFN i(x)=xn-(xmax-x)/dx
DEFFN j(y)=(ymax-y)/dy
'
' X- und Y-Ausmaße des Grafikbildschirms
'
yn=400
xn=640
'
' Lage des Koordinatenkreuzes eingeben
'
CLS
PRINT TAB(19);"Darstellung implizieter Funktionen"
PRINT TAB(19);"----------------------------------"
PRINT
INPUT "    x-Achse von ......: ",xmin
INPUT "               bis ......: ",xmax
INPUT "    y-Achse von ......: ",ymin
INPUT "               bis ......: ",ymax
PRINT
INPUT "Pixelabstand (z.B. 5): ",py
'
' Berechnung einiger Schrittweiten für die Grafik
'
CLS
px=xn/(yn*1.25)*py
dx=(xmax-xmin)/xn
dy=(ymax-ymin)/yn
'
' Berechnung der Funktion als Punktraster
'
HIDEM                              ! Mauszeiger ausschalten
GOSUB koordinatenkreuz
FOR i=0 TO xn STEP px
  FOR j=0 TO yn STEP py
    IF FN f(xmax-dx*(xn-i),ymax-j*dy)<0 THEN
      PLOT i,j
    ELSE
      PLOT i,j+1
      PLOT i,j-1
      PLOT i,j
      PLOT i+1,j
      PLOT i-1,j
    ENDIF
  NEXT j
NEXT i
```

```
'
' Auf eine Taste warten
'
REPEAT
UNTIL INKEY$<>""
'
SHOWM                                  !Mauszeiger wieder einschalten
END
'
PROCEDURE koordinatenkreuz
  LOCAL xnull,ynull,x,y                !Diese Variablen gelten nur hier
  xnull=FN i(0)
  ynull=FN j(0)
  LINE xnull,0,xnull,yn
  LINE 0,ynull,xn,ynull
  FOR x=INT(xmin) TO INT(xmax) STEP ABS((xmax-xmin)/100)
    LINE FN i(x),ynull-2,FN i(x),ynull+2
  NEXT x
  FOR y=INT(ymin) TO INT(ymax) STEP ABS((xmax-xmin)/100)
    LINE xnull-3,FN j(y),xnull+3,FN j(y)
  NEXT y
RETURN
```



Hallo Club-Freunde !,
wie dem vorletzten Club-Info zu entnehmen war, hatte Kajot so seine liebe Not mit einem Textrettungsversuch. Es scheint nicht ganz bekannt zu sein, daß der DUMP-Befehl außer Maschinensprache-Programme, auch einfache Daten auf Diskette bannen kann. Hierzu muß nur ein Parameter anders angeben werden. Die störenden Loader-Informationen werden dann unterdrückt. Nachfolgend die Beschreibung.

Tschüß Kurt

DUMP

DUMP,FNAME1,Startadresse,Endadresse <,Programmstart>

Das Kommando DUMP schreibt den Inhalt des Speichers auf die Diskette unter dem Namen FNAME1. Der Dump beginnt mit der eingegebenen Startadresse und hört mit der eingegebenen Endadresse auf. Start-, End- und Programmstartadresse werden in dezimal (Wert *kleiner als* 65536) oder hexadezimal (Wert *kleiner als* 10000H) eingegeben. Für die hexadezimale Eingabe muß das 'H' angehängt werden.

Das DUMP-Kommando hat zwei verschiedene Funktionen:

* Wird als Programmstartadresse 65535 (0FFFFH) eingegeben, dann wird ein exakter Speicherabzug auf die Floppy geschrieben. In den ersten 2 Byte steht dann lediglich die Startadresse. Diesen File kann man mit dem 'SUPERZAP' anzeigen oder ausdrucken (Funktion 'DMDB') oder auf einem anderen TRS-80 die Fehlersuche durchführen.

* Ist die eingegebene Programmstartadresse kleiner 65535 (0FFFFH) oder wird sie nicht angegeben, so wird der Speicherinhalt im Loader-Format auf die Floppy geschrieben. Dieser File kann dann später einfach über den Loader wieder ins Memory geladen werden.

ACHTUNG: Die Startadresse darf nicht kleiner als 5200H sein, da sonst das DOS zerstört wird.

Beispiel: DUMP,FILENAME/CMD:1,5200H,89ABH,5700H

Der Speicherinhalt zwischen der Adresse 5200H und 89ABH wird auf die Floppy 1 unter dem Namen FILENAME/CMD abgespeichert und als Startadresse 5700H eingetragen. Über das Kommando

LOAD,FILENAME/CMD

kann dieses Programm wieder in den Speicher geladen werden oder über das DOS-Kommando

FILENAME <,Parameter>

direkt ausgeführt werden. Im zweiten Beispiel ist dasgestellt, wie ein Programm bzw. der gesamte Speicherinhalt im Fehlerfall auf Diskette geschrieben wird, wenn man momentan keine Zeit hat, den Speicherinhalt zu analysieren. Vorrausgesetzung ist allerdings, daß man entweder im DOS ist oder über die Tasten 'DFG' ins Mini-DOS kommt.

DUMP,MEMORY/DMP:1,0,65535,65535

Mit diesem Befehl werden die kompletten 64K, einschließlich ROM und Display auf Floppy 1 in den File MEMORY/DMP geschrieben. Das erste und zwei Byte dieses Files sind dann 00 00 (Startadresse des Dumps). Im File sind sonst keine weiteren Kontrollzeichen. Zu einem späteren Zeitpunkt kann man dann über 'SUPERZAP' (Option 'DMDB') sich diesen Dump ansehen und auswerten. Zu beachten wäre noch, daß der DOS-Speicherbereich zwischen 4000H und 51FFH nicht mehr den gleichen Inhalt hat, da 'DUMP' aufgerufen wurde.

Hallo Clubfreunde,
Diesmal geht es um Geld oder wie vermeide ich die Qual mit den Lottozahlen. Das weiter unten wiedergegebene Programm stammt aus meiner Mottenkisten und erblickte vor nicht ganz 10 Jahren zum ersten mal das Licht dieser schnöden Welt. Damals mußte mein erster Komputer - ein 8-Bit Prozessor der auf den Namen SC/MP (National Semiconductor) hörte und in mehreren Folgen in der Bastelzeitschrift 'Elektor' erschien - für diese Aufgabe herhalten. Da das benutzte Integer-Basic ganz und gar nicht als schnell bezeichnet werden konnte, war diese eher 'mickerige' Aufgabe schon ein beachtlicher Brocken an Rechenzeit. Ich hatte deshalb an allen Ecken und Enden 'Geschwindigkeitsoptimiert' wo es nur ging. Was soviel wie Minimierung der benötigten Befehlszeilen bedeutete. Die Ur-Version enthielt sogar noch eine in Basic geschriebene Zahlenformatier-Routine, damit die Nümmerchen auch alle sauber untereinander standen. Alles in Allem war die Berechnung der 10 Ziehungen in unter fünf Minuten abgeschlossen !

Da ich auch heute - ***10 Jahre danach*** - immer noch keine große Lust verspüre mir die Lottozahlen selbst auszudenken, habe ich diesen Programmoldie nochmals überarbeitet und auf das GFABasic übertragen. Hier ist die ganze Aktion in ca. 2 sec. erledigt. Nur mehr gewonnen habe ich deshalb bisher nicht. All' jenen, die es genausowenig lassen können wie ich viel Spaß beim 'Ziehen der Zahlen' - und sei es auch nur auf dem verknitterten Papier aus dem Drucker.

Tschüß

Kurt

```
DIM z%(11)
h1$="         Die Lottozahlen der vergangenen Woche"
INPUT " Ausgabe auf den Drucker (Y/N): ";h3$
IF LEFT$(h3$,1)="Y" THEN
  f!=TRUE
ELSE
  f!=FALSE
ENDIF
CLS
DO
  '
  ' Entscheiden ob Drucker oder nicht
  '
  IF f!=FALSE THEN
    PRINT AT(1,1);h1$
  ELSE
    LPRINT h1$
    LPRINT
  ENDIF
  PRINT AT(1,3);
  '
  ' Die sechs Zahlen ziehen und ...
  '
  FOR i%=1 TO 10
    z%(0)=INT(RND*48+0.5)+1
    durchlaeufe%=1
    FOR j%=1 TO 5
      INC durchlaeufe%
      z%(j%)=INT(RND*48+0.5)+1
      FOR k%=0 TO j%-1
        IF z%(k%)=z%(j%) THEN
          DEC j%
          k%=j%
        ENDIF
      NEXT k%
    NEXT j%
    '
    ' ... aufsteigend sortieren ...
    '
    FOR k%=0 TO 4
      m%=k%
      FOR j%=k%+1 TO 5
        IF z%(m%)>z%(j%) THEN
          m%=j%
        ENDIF
      NEXT j%
      SWAP z%(k%),z%(m%)
    NEXT k%
    '
    ' ...und ausgeben.
    '
    IF f!=FALSE THEN
      PRINT TAB(5);USING "Spiel### => ",i%;
      FOR j%=0 TO 5
        PRINT USING "#####",z%(j%);
      NEXT j%
      PRINT "   [";durchlaeufe%;"] ";
      PRINT
    ELSE
      LPRINT TAB(25);USING "Spiel### => ",i%;
      FOR j%=0 TO 5
        LPRINT USING "#####",z%(j%);
      NEXT j%
      LPRINT
    ENDIF
  NEXT i%
  '
  ' Zehn Ziehungen sind ausgegeben, weiter ?
  '
  PRINT
  PRINT TAB(5);"Abbruch mit      'X'";
  PRINT TAB(5);"Neue Ziehung mit 'CR'";
  REPEAT
    key$=INKEY$
  UNTIL key$<>""
  EXIT IF key$="X"
LOOP
'
END
```

640 * 400 / 2 = ATARI-Bilder auf dem 4p (via T2100)

Während unseres Sommerurlaubs haben meine bessere Hälfte und ich unseren Vorsitzenden Gerald Schröder besucht und bei diesem Anlaß natürlich nicht nur Hamburg besichtigt. Wie kaum anders zu erwarten, wenn zwei Computerfreaks zusammentreffen, wurde auch ausführlich über das Hobby konferiert. Da Gerald sein IIs zu dieser Zeit gerade mittels Lötkolben und Zange außer Betrieb gesetzt hatte, blieb uns zum Probieren und Programmieren nur sein ATARI 1040 ST übrig.

Nicht verschweigen will ich, daß sich das eigentlich ganz gut traf, denn mein Toshiba T2100 [ein MS-DOS (Helmut Bernhardt würde sagen DomesDOS) kompatibler Laptop], der mich auf der ganzen Reise begleitete, kommt mit den 3½"-Disketten des ATARI sehr gut zurecht. Da mich die Grafikfähigkeiten des ATARI immer schon interessiert haben, lag kaum etwas näher, als sich ein paar Bilder vorzunehmen und diese zu übertragen.

Soweit zur Vorgeschichte; nun ans Eingemachte!

Gerald hat, wie schon erwähnt, einen ATARI 1040 ST mit einem monochromen Bildschirm. Die höchste Auflösung, die der ST hinbringt, beträgt 640 * 400 Punkte bei zwei "Farben". Letzteres heißt im Prinzip nichts anderes, als daß nur zwischen "Punkt gesetzt" und "Punkt nicht gesetzt" unterschieden werden kann. Die möglichen Auflösungen im Farbmodus und Farbbilder interessieren uns hier im weiteren nicht.

Zunächst mußten wir herausfinden, wie die Grafiken beim ATARI abgespeichert werden. Dazu wurde mit einem der am weitesten verbreiteten Grafikprogramme (STAD ST Aided Design) eine horizontale und eine vertikale Linie gezeichnet und dieses "Bild" im sogenannten DEGAS-Format abgespeichert. Mit einem Disketteneditor wurde die Datei dann analysiert und folgendes festgestellt:

1. zunächst enthält die Datei 34 Bytes als Header in dem u.a. festgelegt ist, ob es sich um ein Monochrom- oder ein Farb-Bild handelt. Der Header für Monochrombilder sieht wie folgt aus: 00 02 07 77 00 00 00 70 00 00 07 77 07 00 00 70
   07 70 00 07 07 07 00 77 05 55 03 33 07 33 03 73
   07 73.

Zur Unterscheidung der Farbbild-Header:
00 00 03 25 07 00 00 70 07 70 02 57 07 40 00 77
07 77 00 07 07 07 03 73 07 73 03 37 07 37 03 77
00 00.

Was die einzelnen Bytes und Bytefolgen bedeuten, kann ich euch leider nicht mitteilen (das wäre aber wahrscheinlich auch nur für ATARI-Besitzer interessant), aber man kann anhand des Headers immer unterscheiden, ob es sich überhaupt um ein Bild im DEGAS-Format (es gibt auch Grafikprogramme, die andere Abspeicherformate benutzen) handelt und ob ein Farb- oder ein s/w-Bild vorliegt.

2. Nach dem Header folgen 32000 Bytes Bildinformation, die folgendermaßen organisiert sind: je 80 Bytes repräsentieren eine Zeile; jedes Byte repräsentiert 8 Bildpunkte, wobei die Reihenfolge Bit 7 = erster, Bit 0 = letzter Bildpunkt ist. Ein einzelner waagerechter Strich über die ganze Bildbreite am oberen Bildrand ergibt also 80 Bytes mit dem Wert FFh. Ein senkrechter Strich am linken Bildrand ergibt im ersten Bildinformationsbyte den Wert 80h, dann 79 Bytes 00h, danach wieder ein Byte 80h usw. usw.. D.h., die Zeilen sind in der Reihenfolge abgespeichert, in der sie auch auf dem Bildschirm stehen.

Nachdem geklärt wäre, wie die Dateien aufgebaut sind, ist es eigentlich kein Problem mehr, die Bilder sichtbar zu machen. Zunächst habe ich, noch in Hamburg, die Bilder in den Grafikspeicher meines T2100 eingelesen. Im Gegensatz zum "normalen" CGA (Color Grafik Adapter), der nur 640 * 200 Pixel darstellen kann, beherrscht der Toshiba auch einen Grafikmodus mit 640 * 400 Punkten. Wenn nun auch noch die Zuordnung der Speicherstellen zu den Bildschirmzeilen so einfach gestaltet wäre wie beim ATARI, wäre die Sache sehr einfach; leider ist das aber nicht der Fall! Das soll aber hier nicht weiter interessieren, wir wollen uns ja dem 4p bzw. den TRS 80-Computern zuwenden.

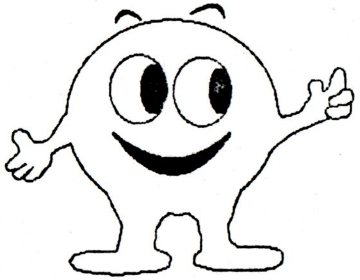
Die µLabs-Grafikkarte des Model 4p kann maximal 640 * 240 Punkte monochrom darstellen und ist damit um 40 vertikale Punkte besser als der CGA. Nichts desto trotz können die ATARI-Bilder nicht "am Stück" angezeigt werden. Trotzdem gibt es Möglichkeiten, die Grafiken doch zu nutzen. Entweder liest man nur den oberen oder den unteren Teil eines Bildes ein, oder man wählt einen Mittelteil, der möglichst viel Bildinformation enthält. Diese Methode ist relativ einfach zu realisieren, befriedigt aber nicht so ganz, da immer Bildinformationen verloren gehen.

Erheblich eleganter ist es, das Bild zu komprimieren und so auf 640 * 200 Punkte zu halbieren. Um das zu erreichen, könnte man natürlich einfach jede zweite (alle geraden) Zeile(n) weglassen, verliert dadurch aber erheblich an Bildinformation. Überträgt man z.B. ein Schaltbild, bei dem eine waagerechte Linie in der zehnten Zeile über den ganzen Bildschirm verläuft, wäre diese Linie (sie liegt ja in einer Zeile mit gerader Nummer) verschwunden, die Zeichnung damit unbrauchbar! Dasselbe könnte natürlich passieren, wenn man alle ungeraden Zeilen weglassen würde.

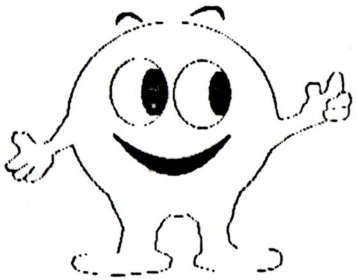
Eleganter und ohne Bildverlust ist die Methode, die Gerald Dreyer und ich schon einmal bei der Anzeige von BTP-Bilddateien (Grafikhilfsprogramm zu dBase II unter CP/M) angewendet haben. Dabei werden einfach jeweils zwei Zeilen bildpunktweise ODER-verknüpft. Hier ein kleines Beispiel:

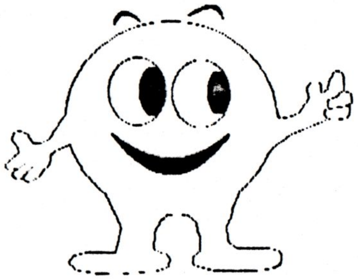
```
Bildpkt: 0 1 2 3 4 5 6 7           Bildpkt: 0 1 2 3 4 5 6 7

Zeile 1: * * * * * * * *  \
                           OR ->    Zeile 1: * * * * * * * *
Zeile 2: *                /

Zeile 3: * * * * * *      \
                           OR ->    Zeile 2: * * * * * *
Zeile 4: *                /

Zeile 5: *                \
                           OR ->    Zeile 3: *
Zeile 6: *                /
```

Das links dargestellte "F" ist rechts zwar etwas verstümmelt, zur Not aber noch als solches zu erkennen. Bei Bildern ist der Effekt erheblich besser, wie das Beispiel auf der folgenden Seite zeigt. Das erste Bild zeigt nur den oberen Teil der Grafik, das zweite die ganze, allerdings nach oben beschriebenem Verfahren komprimierte, Grafik.



Im linken Bild ist nur der obere Teil des Bitmännchens zu sehen, während rechts, der durch OR-Verknüpfung von jeweils zwei Zeilen komprimmierte Mister Bit ganz sichtbar wird.





Ganz egal, ob man alle geraden oder alle ungeraden Zeilen wegfallen läßt, die Bildverluste sind immer sichtbar!

Das folgende Listing in Turbo-Pascal 3.0 stellt nicht das ganze Programm zum Einlesen von ATARI-Bildern auf dem 4p, sondern nur die eigentliche Einlese/Anzeige-Routine dar. Das ganze Programm abzudrucken lohnt nicht, es ist in der CP/M-Programmsammlung bei Andreas Rychlik zu haben. Hinzuweisen ist auf die Art, in der die Grafikdatei eingelesen wird. Normalerweise funktioniert das Einlesen einzelner Bytes unter CP/M nämlich nicht. Ich habe dazu das Bibliothekmodul BYTEFILE.BIB aus der CHIP Spezial Turbo-PASCAL Nr. 4 benutzt, welches diese Möglichkeit zur Verfügung stellt. Auch das Modul gibt es bei Andreas. Die Dateioperationen enden daher auch immer mit "ByteFile" (z.B. "AssignByteFile") und sehen etwas gewöhnungsbedürftig aus!

Ich glaube, aus den Kommentaren im Listing geht eigentlich alles wesentliche hervor, so daß sich weitere Erklärungen erübrigen. Anzumerken wäre noch, daß auf anderen TRS 80-Computern (z.B. Genie IIs) das Programm durchaus etwas anders aussehen kann. Das liegt daran, daß hier nicht direkt in den Bildschirmspeicher geschrieben werden kann, sondern für jeden Bildpunkt die PLOT-Routine aufgerufen werden muß. Das verlangsamt die Sache zwar, ändert aber nichts am Prinzip und klappt ganauso gut, wie ich von Alexander Schmid weiß. Übrigens ähnelt auch die Einleseroutine für den 640*200-CGA-Modus der oben gezeigten Routine so stark, daß nur wenig Umbauarbeiten notwendig sind, um das Programm auf PC's umzustricken.


```
procedure restore(name:filename;var exist:boolean);

  var header : array [0..33] of byte;           {Buffer für den Header}
      zeile1, zeile2 : array [0..79] of byte;  {Buffer für je eine Zeile}
      i,j:integer;
      datei:ByteFile;

  begin
    clsh;                           {Löschen der HRG}
    {$i- }                          {Abschalten der Fehlerbehandlung durch Turbo-Pascal}
    AssignByteFile(datei,name);     {Datei öffnen}
    ResetByteFile(datei);
    {$i+ }
    if IOResultByteFile=0 then      {wenn dabei kein Fehler auftritt,}
      begin
        exist:=true;                    {"Alles OK"-Flag für die aufrufende Routine setzen}
        for j:=0 to 33 do
         ReadByteFile(datei,header[i]); {34 Byte Header Lesen}
        port[131]:=185;                 {Steuerport des µLabs-HRG programmieren}
        for j:=0 to 199 do              {für 200 Zeilen}
          begin
            port[128]:=0;               {x-Register auf 0}
            port[129]:=j;               {y-Register auf aktuelle Zeile}
            for i:=0 to 79 do           {eine Zeile = 80 Byte einlesen}
              ReadByteFile(datei,zeile1[i]);
            for i:=0 to 79 do           {noch eine Zeile einlesen}
              ReadByteFile(datei,zeile2[i]);
            for i:=0 to 79 do           {alle 80 Bytes beider Zeilen ODER-verknüpfen}
              begin                     {und in den Datenport schreiben. Es werden}
               port[130]:=(zeile1[i] or zeile2[i])
            end;                        {also immer 8 Bildpunkte gleichzeitig gesetzt!}
        end;                            {wenn 200 * 2 Zeilen gelesen und bearbeitet sind}
        CloseByteFile(datei)            {Datei schließen}
      end
    else                            {falls beim Öffnen der Datei ein Fehler aufgetreten}
      begin
        exist:=false;               {ist, "Alles OK"-Flag rücksetzen und zurück ins auf-}
        exit;                       {rufende Programm!}
      end;
  end;
```

Wem die Kompression des Bildes nicht behagt, der kann sie natürlich auch direkt auf den Drucker ausgeben. Alexander Schmid hat ein solches Programm geschrieben und wird es sicher auch in der CP/M-Bibliothek deponieren.

Bleibt nur noch die Frage zu klären, wo man ATARI-Bilder herbekommt. Ich habe es da ja relativ einfach mit meiner Methode ATARI - Toshiba T2100 - 4p, da der T2100 sowohl die ATARI- als auch die CP/M-Disketten des 4p direkt lesen/schreiben kann. Wer diese Möglichkeit nicht hat, muß auf Datenübertragung per serieller Schnittstelle zurückgreifen. Natürlich bin ich gerne bereit, Dateien (in fast jede gewünschte Format) zu übertragen. Außerdem sind auf der Programmdiskette in der CP/M-Bibliothek ein paar Bilder und ich habe auch noch ein paar, die ich, allerdings nur gegen Altersnachweis (da nicht ganz jugendfrei) weitergebe.

Viel Spaß mit den ATARI-Bildern,

CP/M-Warmstart ohne Nachladen ?

Ich kann den Ärger von Rüdiger Sörensen im Club-Info Nr. 25 nur teilen, wenn er sich über die "Dikettenraspelei" von CP/M bei ^C beschwert. Wenn man sich dazu noch vor Augen hält, daß ein Nachladen des CCP oder gar des BDOS eigentlich heute kaum noch notwendig ist, ist das Verfahren um so ärgerlicher. Die Nachladerei ist ein Relikt aus der Kinderstube von CP/M, als in seligen Urzeiten 16 kByte RAM in einem Rechner schon den Gedanken an CP/M aufkeimen ließen.
Die Generierungsanweisung für CP/M mit MOVECPM ist noch Heute Beweis für die Machbarkeit solcher "Mini-Systeme". Bei einem 16- oder 32-Kbyte-System muß natürlich alles, was nicht lebensnotwendig für das System ist, bei Start eines User-Programmes über Bord geworfen werden. Hier heißt die Devise TPA um jeden Preis!
Anders stehen die Dinge bei einem 64-KByte-System. Selbst umfangreiche Programme, wie DBASE II oder auch WordStar, selbst mit aktiviertem internem Drucker-Puffer, laufen ohne sich am CCP oder gar am BDOS vergreifen zu müssen.
Das Problem ist also nur, den Zugriff auf den Adress-Bereich des CCP zu verbieten. Wenn es gelingt den CCP generell vor überschreiben zu schützen, wird das Nachladen überflüssig und bei Warmstart wird lediglich noch das Rücksetzen des Systems notwendig. D.h. es wird in diesem Fall kein System auf der Diskette benötigt! Der Vorteil liegt auf der Hand. Das "Blankfeilen" der Systemspur entfällt, ^C wird zur Sekunden-Sache, fremde Disketten ohne eigenes System können problemlos auch in Laufverk A beschrieben werden, POWER wird zum TURBO-POWER...
Die Programme, die ein überschreiben des CCP bedarfsweise durchführen, orientieren sich am Inhalt der Adresse 06-07 (BDOS-BASE) und überschreiben alles was vor dieser Adresse liegt. Der Trick liegt jetzt darin, dem Programm eine Adresse zu übergeben, die vor dem CCP liegt. Damit ist der CCP tabu!
Dieses Verfahren wird z.B. auch von ZSID oder DU benutzt, um sich vor überschreiben zu schützen.
Nun könnte man sagen, nichts leichter als das, direkt vor dem CCP wird ein JUMP auf die eigentliche BDOS-BASE eingetragen und auf den Adressen 06-07 wird die Adresse zu eben diesem JUMP abgelegt.
Die CCP-BASE beginnt immer auf einer geraden Adresse, jedoch sind die ersten 6 Byte frei und werden nicht benutzt. Dort läßt sich also der Sprung zur BDOSBASE eintragen.

```
0005     : JUMP NEWBASE  (CCPBASE)

NEWBASE:   JUMP BDOSBASE
```

Leider ist die Angelegenheit so einfach nun auch wieder nicht. Bei Warmstart wird der Vereinbarungsbereich von Adr. 00 bis 07 neu initialisiert. D.h. nach einem Warmstart würde die alte BDOSBASE wieder eingetragen werden, damit ist der CCP wieder zum "Abschuß" freigegeben. Außerdem würde sich die in der Warmstart-Routine vorhandene Ladeprozedur für den CCP und das BDOS herzlich wenig um unsere Manipulation kümmern und nach wie vor den Ladevorgang durchführen.
Also muß auch hier geändert werden. Im allgemeinen beginnt die BIOS-Routine WBOOT mit dem Rücksetzen des Systems, wie Selektion DRIVE A, setzen Stackpointer usw. Dann folgt die Routine zum Nachladen einer entsprechenden Anzahl von Sektoren von den Systemspuren. Danach wird die Initialisierung des Vereinbarungsteiles vorgenommen. Abgeschlossen wird die Routine mit einem JUMP zum CCP.

Folgende Änderungen in der Warmstart-Routine müssen also vorgenommen werden:
- Änderung der Initialwerte für Adr. 06-07
- Nachladeroutine überspringen.

Mann kann die Sache natürlich auch komfortabel gestalten und das Überschreiben wahlweise zulassen oder nicht. Dazu kann ein System-Flag angelegt werden, in dem beispielsweise auch das Laufwerk festgelegt werden kann, von dem nachgeladen wird.
Weiterhin ist ein 2-Byte Merkflag für die jeweils aktuelle Sprungadresse auf Adr. 06-07 einzurichten, das irgendwo im BIOS installiert werden kann.
Im folgenden ein Vorschlag für ein solches Vorgehen:

```
Aufbau SYSFLAG  BIT 0-1 Laufwerks-Code des gewünschten aktuellen Lauf-
                        werkes nach ^C (A=00, B=01...), gleichzeitig
                        das Laufwerk, von dem evtl. CCP/BDOS nachgela-
                        den werden.
                BIT 7   Schalter für nachladen CCP/BDOS
                        0= kein nachladen
                        1= nachladen
```

Beispiel-Routine für WBOOT:

```
WBOOT: LD   SP,0080          ; Stackpointer auf Standard
       LD   A,(SYSFLAG)      ; BIT 0-1 Laufwerks-Code
                             ; BIT 7   0=kein nachladen, 1=nachladen
       PUSH AF               ; A-Reg. sichern
       AND  03,A             ; Nur BIT 0 und 1, Laufwerks-Code DRIVE A-D
       LD   C,A              ; Einstieg SELDSK vorbereiten
       CALL SELDSK           ; Aktuelles Laufwerk selektieren
       LD   HL,BDOSBASE      ; Wert für Adr. 06-07 bei ungeschützem CCP
       LD   (BASFLAG),HL     ; in Merkflag laden
       POP  AF               ; A-Reg. zurück
       BIT  7,A              ; Test auf nachladen
       JP   NZ,M1            ; CCP und BDOS nachladen, Vereinbarungsteil
                             ; aufbauen und Start
       LD   BC,NEWBASE       ; Wert für Adr.06-07 bei geschütztem CCP in
       LD   (BASFLAG),BC     ; Merkflag laden
       LD   A,C3             ; JP-Befehl auf NEWBASE vor CCP
       LD   (BC),A           ; eintragen
       LD   (NEWBASE+1),HL   ; Adr. BDOSBASE für JP-Befehl eintragen
       JP   START            ; Kein nachladen!

M1    :  .   Nachladeroutine CCP u. BDOS
         .
         .
START :  .   Initialisierung Vereinbarungsteil
         .
         .
       LD   HL,(BASFLAG)     ; BDOSBASE auf Adr.06-07
       LD 0006,HL            ; laden
         .
         .
       JP   CCP              ; Systemstart
```

Die Einstellung des SYSFLAG kann z.B. recht einfach mit Hilfe des [illegible] POWER über zwei der dort vorhandenen vier USER-Befehle realisiert werden.
Diese Befehle führen ein Unterprogramm aus, das in POWER auf folgenden Adressen eingetragen werden muß:

```
UP1  0140-0147H   ; Jeweils 8 Byte lang, müssen mit RET abgeschlossen
UP2  0148-014FH   ; werden!
UP3  0150-0157H   ;
UP4  0158-015FH   ;
```

Im folgenden Beispiel wird UP1 zur Einstellung Warmstart ohne nachladen CCP/BDOS und UP2 zur Einstellung mit nachladen verwendet.

```
UP1
---
0140H  LD   A,00          ; kein nachladen, DRIVE A ist aktuelles Laufw.
       LD   (SYSFLAG),A   ; nach ^C
       RET

UP2
---
0148H  LD   A,81          ;Nachladen CCP und BDOS von DRIVE B
       LD   (SYSFLAG),A
       RET
```

Das vorgestellte Verfahren ist seit einem Jahr in meinem BIOS integriert und hat sich bewährt.

Frank-Michael Schober

BDOS-Replacement, was'n das ?

Es dürfte sich ja allgemein rumgesprochen haben, daß CP/M inzwischen etwas betagt ist und auch etliche Kanten und Ecken hat. Einige fleißige Leute haben sich nun gedacht, selbst ist der Mann und haben für Abhilfe gesorgt. Das Original von Digital Research wurde natürlich in 8080-Assembler geschrieben und damit bringt man halt relativ wenig in den rund 3 1/2 KB unter, die dem BDOS zur Verfügung stehen.

Kurzer Einschub für die 'nur NEWDOSler': CP/M besteht aus drei Teilen. Ganz unten auf Hardwareebene ist das BIOS, das Basic In Out System. Dieses ist das direkte Interface zum jeweiligen Rechner und daher systemspezifisch. Es stellt die elementarsten Funktionen wie z.B. Zeichen ausgeben usw. Darüber kommt das BDOS, das Basic Disk Operating System. Dieses ist auf jeder Maschine gleich und führt seine Funktionen durch CALLs ins BIOS aus, stellt also eine standardisierte Schnittstelle für alle Ein-/Ausgaben dar. Als letztes kommt der CCP, der Command Console Processor, der die Schnittstelle zum Benutzer darstellt. Mit dem 'unterhält' man sich, wenn man auf der Betriebssystemebene Befehle eintippt.

Ein Weg, das BDOS zu verbessern war nun, einen leistungsfähigeren Befehlssatz, sprich den des Z80, zu nehmen. Damit gehört das Control-C nach jedem Diskettenwechsel ebenso der Vergangenheit an, wie die geheimnisvollen Fehlermeldungen. Außerdem wird eine Echtzeituhr unterstützt, damit man genau weiß, wann ein File erstellt oder zuletzt bearbeitet wurde. Weiterhin kann man Files 'public' machen, d.h. sie sind aus allen User-Bereichen zugänlich. Damit kann man z.B. den Compiler in einem Bereich und die Source-Files nach Anwendung geordnet in anderen Bereichen unterbringen.

Der Autor eines der vielen Replacements war so freundlich und hat einige gegenübergestellt:

Name	CP/M	ZRDOS+	Z80DOS	P2DOS21	DOS+25	SUPERDOS
Author	Digital Research, Inc.	Echelon, Inc.	Carson Wilson	H.A.J. Ten Brugge	C.B. Falconer	Benjamin Ho
Derivation	Unknown	Unknown	P2DOS, SUPERDOS	Unknown	P2DOS	P2DOS
Time stamps	No	No	(*)C,U,A	C,U	C,U,A	C,U
Disks auto-login	No	Yes	Yes	No	No	Yes
Archive	No	Yes	Yes	Yes	Yes	No
Public files	No	Public user areas	F2 attribute	F2 attribute	System files at A0:	F2 attribute
Get/Use stamps	No	No	Yes	No	No	No
Get/Set time	No	No	Yes	Yes	Yes	Yes
Error messgs.	Cryptic	Clear	Legible, give function & file	Legible, give function & file	Legible, give function & file	Legible, give function & file
Return current DMA	No	Yes	No	No	Yes	No
Wheel-protect files	No	Yes	No	No	No	No
Set/res warm boot	No	Yes	No	No	No	No
Source code	No	No	Yes	Yes	$50	Yes
Approx. price	$20	$60	free	free	free	free

(*) C = Create, U = Update, A = Last Access

Ich arbeite zur Zeit mit dem Z80DOS und bin sehr zufrieden. Die Sources habe ich von Gerald Schröder bekommen, der sie von Rüdiger Sörensen hat, der sie frisch über den großen Teich geholt hat. Der Aufwand ist nicht groß und lohnt sich allemal, selbst wenn es nur wegen dem Ctrl-C ist.

Alexander Schmid

Es war einmal ein Datum

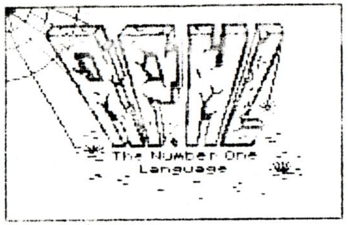
Die meisten BDOS-Replacements unterstützen (endlich?) auch eine Echtzeituhr. Damit kann man sich entweder ausgeben lassen, wielange man noch bis zum Kaffee Zeit hat, oder alle Einträge im Directory erhalten einen 'Zeitstempel', der einem sagt, wann das File erstellt wurde, oder wann zuletzt darauf zugegriffen wurde. Das ist vor allem dann nützlich, wenn man nicht mehr weiß, welches das allerletzte Update war, wo man den nobelpreisverdächtigen Geistesblitz hatte.

Schön und gut dachte ich, als ich mit Helmut Bernhardts Schaltplan so eine Zwiebel zusammengestrickt hatte, das schwierigste ist jetzt erledigt, aber es kommt ja erstens anders und zweitens als man denkt. Der RTC5832 gibt auf Anfrage seine Daten im BCD-Code her, was aber kein großes Hindernis ist. Wenn man die Register (siehe Arnulfs Artikel 'Real Time Black Box') nacheinander abklappert hat man schließlich das Datum in der Form 'Jahr, Monat, Tag, Stunde, Minute, Sekunde'. Leider scheint man in Amerika, wo die Autoren nunmal sitzen, nicht viel von dieser Notation zu halten. Das Z80DOS (und auch andere) wollen das Datum als 'Anzahl der Tage seit dem 1.1.78'. Fragt mich nicht, wie man auf sowas kommt, ich weiß es nicht. Zum Glück waren auf den Disketten auch Programme, mit denen man die Uhr stellen kann. Logischerweise mußten die eine Routine zum Umrechnen haben, damit man das Datum in der 'normalen' Form eingeben kann. Das unten abgedruckte Listing ist nun der eigentliche Teil, der die Umrechnung bewerkstelligt. Als Ergebnis bekommt man schließlich die Zahl im Register BC.

Arnulf möge mir verzeihen, daß ich das Listing nicht kommentiert habe, aber mich hatte im Moment nur interessiert, DASS es geht und nicht WIE. Wenn mich mal der Ehrgeiz packt, werde ich den Algorithmus vielleicht mal auseinander nehmen, aber auch nicht früher. Wer sich für die Datumsrechnung mehr interessiert, der sei herzlich eingeladen, sich dazu zu äußern.

```
;  Berechnung der Tage seit dem 1.1.1978
;
;  Eingabe : HL=Jahr   (1978..2157)
;            DE=Monat  (1..12)
;            BC=Tage   (1..31)
;  Ausgabe : BC=Tage   erster  Tag (00001) : Sonntag 01.01.1978
;                      letzter Tag (65535) : Sonntag 05.06.2157
;
DAYS:    LD      B,0
         PUSH    DE
         LD      DE,1978
DAYS0:   OR      A
         SBC     HL,DE
         ADD     HL,DE
         JR      Z,DAYS1
         PUSH    HL
         LD      HL,365
         ADD     HL,BC
         LD      B,H
         LD      C,L
         POP     HL
         EX      DE,HL
         CALL    LEAPYR
         EX      DE,HL
         INC     DE
         JR      NZ,DAYS0
         INC     BC
         JR      DAYS0
DAYS1:   POP     DE
         PUSH    HL
         LD      HL,DM
         LD      D,1
DAYS2:   LD      A,E
         CP      D
         JR      Z,DAYS4
         LD      A,C
         ADD     A,(HL)
         LD      C,A
         JR      NC,DAYS3
         INC     B
DAYS3:   INC     HL
         LD      A,D
         INC     D
         CP      2
         JR      NZ,DAYS2
         EX      (SP),HL
         CALL    LEAPYR
         EX      (SP),HL
         JR      NZ,DAYS2
         INC     BC
         JR      DAYS2
DAYS4:   POP     HL
         RET
;
; Korrektur für ein Schaltjahr
;
;  Eingabe : HL  = Jahr
;  Ausgabe : Z   = 1 Korrektur notwendig
;            Z   = 0 keine Korrektur nötig
;
LEAPYR:  LD      A,L
         AND     3
         RET     NZ
         PUSH    HL
         PUSH    DE
         LD      A,0FFH
         LD      DE,100
         OR      A
LEAPY0:  INC     A
         SBC     HL,DE
         JR      NC,LEAPY0
         ADD     HL,DE
         LD      H,A
         LD      A,L
         OR      A
         JR      NZ,LEAPY1
         LD      A,H
         AND     3
         JR      LEAPY2
LEAPY1:  XOR     A
LEAPY2:  POP     DE
         POP     HL
         RET

DM:      DEFB    31,28,31,30,31,30,31,31,30,31,30,31     ;Tage pro Monat
```

Alexander Schmid



Hallo RPNL-Freunde !,
Wie sicherlich noch keinem bekannt ist, gibt es von diesem tollen Hochsprachenkompiler inzwischen die Version 2.1. Einige Punkte wurden von mir verbessert oder abgeändert. So erscheint nun nicht mehr bei jedem 'NEW' oder bei einem Fehler diese lästige Copyright-Meldung. Wichtigeres hat sich jedoch im Bereich der Zahlen und im Deklarationsteil getan. Hier können jetzt 32bit Variablen definiert werden. Konstanten lassen sich ebenfalls in 32bit dezimal und ***hexadezimal*** (!) angeben. Hexadezimales wird wie bei einem Assembler durch ein angehängte 'H' gekennzeichnet. Das sieht dann so aus:

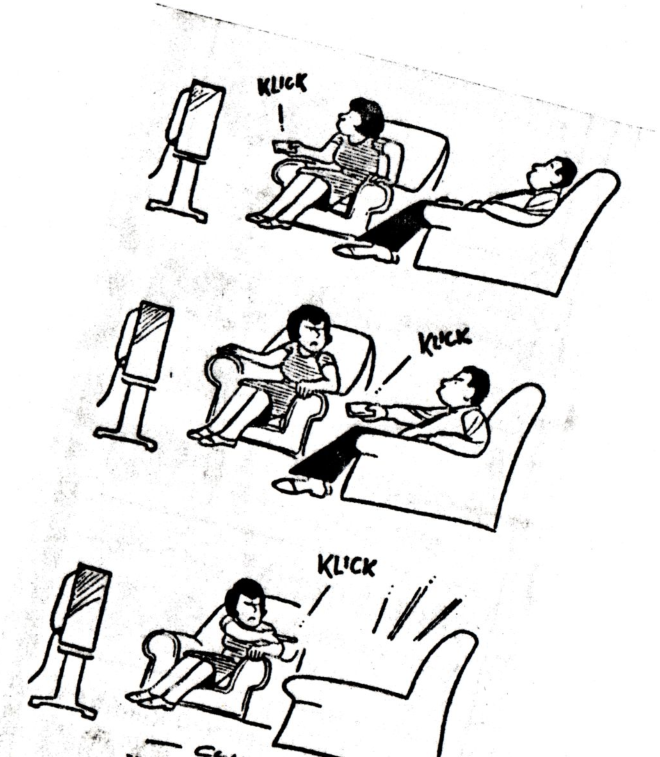
12345	einfache Genauigkeit (16bit)
12345333.	doppelte Genauigkeit (32bit)
12FFH	einfache Genauigkeit (16bit Hex)
1FF088DD.H	doppelte Genauigkeit (32bit Hex)

Eine Hex-Zahl in 32bit sieht etwas komisch aus, aber so ist es nun einmal. Dazu passend gibt es dann auch die entsprechenden Zuweisungsanweisungen :=(D) und ?(D). Array's mit 32bit gibt es nicht. Das wäre Platzverschwendung. Hier sind einfach die doppelte Menge an Bytes zu reservieren. Die Stringdeklaration wurde auch überarbeitet und funktioniert jetzt hoffentlich fehlerfrei. In der alten Version wurde bei fehlendem zweiten (') alles was sonst noch kam als String vereinnahmt. Das ist zwar simpel, aber kein guter Stil. Jetzt wird dem Programmierer seine Vergeßlichkeit um die Ohren geschlagen. Variablen und Konstanten für 32 Bit werden mit DVAR bzw. mit dem Punkt am ende der Zahl definiert. Bei der Konstanten erkennt der Compiler automatisch um was es sich handelt (DPL ist dann temporär ungleich Null. S. a. weiter unten). Die Arithmetik habe ich nur auf 32 * 16 Bit bei der Punktrechnung erweitert. Der Aufwand ist ganz beachtlich, außerdem ist das Runtimemodul mit den ganzen Erweiterungen und Umbauten jetzt gute 3.5KByte im Umfang. Für meinen Geschmack eigentlich zu Umfangreich. RPNL sollte eigentlich eine 'kleine Sprache' sein, zumal es mich bei den PASCAL-Systemen immer mächtig geärgert hat, daß auch bei kleinsten Programmen stets der 12K große Runtimekern vorhanden war. Einen Vermutstropfen will ich nicht verschweigen: Die Zahlenausgabe ist etwas langsamer geworden, als beim 'Original'. Ursache ist eben jene erweiterte Arithmetik. Die normalen Multiplikations- und Divisionsroutinen sind rausgeflogen und durch die neuen ersetzt. Was bedeutet, daß die 16 * 16 Bit Befehle genaugenommen auch mit 32 * 16 Bit Genauigkeit ablaufen.- Nur wird hier Vorzeichenrichtig erweitert und dann gerechnet. Eine entsprechende Umwandlungsfunktion von Single auf Double Precision ist dementsprechend eine Selbstverständlichkeit. Das sieht dann so aus: ...DOUBLE... und fertig ist die Laube. Das Vorzeichen kann sehr einfach mit folgender Sequenz übertragen werden: ... OVER SIGN oder mit ... ROT DUP >R -ROT R> DSIGN für doppelt genaue Werte. Übertragen wird stets das Vorzeichen der zweitobersten Zahl auf die zuoberste liegende Zahl auf dem Stack. Wer will kann auch einen Dezimalpunkt in die Ausgabe einbauen. Hierzu ist die Systemvariable DPL mit einem Wert größer Null zu laden. Der hier abgelegte Wert bezieht sich auf die Stellen ***nach*** dem Komma ! DPL wird nicht nach jeder Ausgabe zurückgesetzt. Dies muß explizit durch das Anwenderprogramm erfolgen. Eine Ausgabe in einem Printfeld ist über die Systemvariable FLD möglich. Dazu wird dort die Feldbreite eingeschrieben. Paßt die Zahl nicht in das Feld hinein, wird FLD automatisch auf 32 (!) gesetzt. Soetwas kann immer dann passieren, wenn die Systemvariable BASE auf den Wert 2 für Ausgabe im Binärsystem gesetzt ist. Hier sollte man sich also vorher gut überlegen, was man will. Die Änderung wird nicht automatisch zurückgenommen ! (Doppelt genaue Zahlen werden selbstverständlich mit PRINTD ausgegeben).

Zu guter letzt noch etwas aus Opa's Mottenkiste: Eine Zahlenkonvertierungs-Routine, die auch vor Hexadezimal- und 32bit-Werten nicht zurückschreckt. Das Stück stammt aus der RPNL-Version 2.1 Das Programm ist in der Lage, mit jeder Zahlenbasis zu arbeiten, aber wie schon beschrieben begnüge ich mich mit den beiden üblichsten Formen. Wem diese Beschränkung nicht gefällt, kann sie ja herausnehmen. Darüber hinaus kann sich das Programm den Ziffernstring selbständig aus einem Eingabestring herausfischen. Näheres sind aus dem DECLARE-Teil am Anfang des Listings zu ersehen. Dort sind einige Teststrings angegeben. Das 'DVAR' sollte nicht verwirren, da es sich hier um die Deklaration einer 32bit-Variablen handelt. Die Vers. 2.0 kennt diese Anweisung nicht. Hier muß stattdessen mit 'ARRAY name 2' der nötige Speicherplatz reserviert werden. Das Wandlungsergebnis befindet sich zusammen mit einigen Flags (siehe Kommentare im Listing) auf dem Stack und kann mit 'D.' und 'S.' ausgegeben werden. 'D.' bedeutet 'gebe eine doppelt genaue Zahl auf dem Bildschirm aus', 'S.' bewirkt das Gleiche für einfachgenaue Zahlen.

Tschüß

Kurt



Karikatur: Rohmann Pressebüro

```
;*********************************************
;**           N U M B E R / R P N          **
;*********************************************
;
DECLARE
   STRING $1 '11111'
   STRING $2 '22222 '
   STRING $3 ' 33333'
   STRING $4 ' 44444 '
   STRING $5 '55555   '
   STRING $6 '66666.'
   STRING $7 '1000H'
   STRING $8 '10X00'
   STRING $9 '100 XYZ'
   DVAR   NUM
DEND


CALL $DIGIT       ;( (AC)<(0) -> gesetztes carry)
 D6 30            ; $DIGIT:SUB      '0'
 D8               ;        RET      C
 FE 0A            ;        CP       9+1
 38 06            ;        JR       C,DIGI1
 D6 07            ;        SUB      7
 FE 0A            ;        CP       10
 D8               ;        RET      C
 B9               ;        CP       C            ;< BASE ?
 3F               ; DIGI1: CCF
 C9               ;        RET
CEND


CODE PARSE        ;(adr1 -> adr2 cnt flag)
 E1               ;PARSE:  POP     HL            ;(ADR1 -> ADR2 CNT FLAG)
 46               ;        LD      B,(HL)        ;LAENGE HOLEN
 04               ;        INC     B             ;FALLS LAENGE =0 !
 23               ;PAR1:   INC     HL
 05               ;        DEC     B
 28 07            ;        JR      Z,EPAR1
 7E               ;        LD      A,(HL)
 FE 21            ;        CP      ' '+1
 38 F7            ;        JR      C,PAR1
 18 07            ;        JR      PARSE2
 E5               ;EPAR1:  PUSH    HL            ;POINTER
 0E 00            ;        LD      C,00
 C5               ;        PUSH    BC            ;COUNT
 C5               ;        PUSH    BC            ;FLAG = TRUE BEI INHALT
 18 17            ;        JR      PARSE3        ;FLAG = FALSE KEIN INHALT
 0E 00            ;PARSE2:LD      C,00          ;LAENGENZAEHLER SETZEN
 E5               ;        PUSH    HL            ;POINTER AUF STRINGANFANG
 0C               ;PAR2:   INC     C
 78               ;        LD      A,B
 B7               ;        OR      A
 28 07            ;        JR      Z,EPAR2-1
 7E               ;        LD      A,(HL)
 23               ;        INC     HL
 05               ;        DEC     B
 FE 21            ;        CP      ' '+1         ;ZEICHEN <= BLANK ?
```



```
 30 F4            ;        JR      NC,PAR2
 0D               ;        DEC     C                    ;TRENNZEICH. NICHT MIT-
 06 00            ;EPAR2:  LD      B,00
 C5               ;        PUSH    BC
 01 FF FF         ;        LD      BC,0FFFFH
 C5               ;        PUSH    BC
CEND              ;PARSE3:JP      NEXT


CODE ?BASE        ;cnt (adr2 cnt -> adr2 cnt base)
 D1               ;QBASE:  POP     DE      ;cnt
 E1               ;        POP     HL      ;adr2
 E5               ;        PUSH    HL
 D5               ;        PUSH    DE
 19               ;        ADD     HL,DE          ;POINTER AUF LETZTES STRING-
 01 10 00         ;        LD      BC,0010H       ;...ZEICHEN ERRECHNEN
 2B               ;        DEC     HL             ;SONST ZU WEIT HINTEN !
 7E               ;        LD      A,(HL)
 FE 48            ;        CP      'H'
 28 02            ;        JR      Z,BEND
 0E 0A            ;        LD      C,0AH
 C5               ;BEND:   PUSH    BC
CEND


CODE (NUMBER)     ;                          ;(adr1 adr2 cnt base -> d1 dpl true)
 C1               ;NUMBER:POP     BC         ;(adr1 adr2 cnt base -> adr1 false)
 D9               ;        EXX
 C1               ;        POP     BC
 06 FF            ;        LD      B,0FFH          ;B'   = DPL
 D9               ;        EXX                     ;C'   = CNT
 DD E1            ;        POP     IX              ;DEHL = D1
 11 00 00         ;        LD      DE,0000         ;BC   = BASE
 21 00 00         ;        LD      HL,0000         ;IX   = ADR2
 DD 7E 00         ;NUM1:   LD      A,(IX)          ;HOLE EIN ZEICHEN
 CD $DIGIT        ;        CALL    $DIGIT
 38 2A            ;        JR      C,TSTP          ;KEINE ZIFFER
 F5               ;        PUSH    AF
 CD $MULT         ;        CALL    $MULT
 F1               ;        POP     AF
 C5               ;        PUSH    BC              ;NEUE ZIFFER EINFUEGEN
 4F               ;        LD      C,A
 09               ;        ADD     HL,BC
 EB               ;        EX      DE,HL
 0E 00            ;        LD      C,00
 ED 4A            ;        ADC     HL,BC
 EB               ;        EX      DE,HL
 C1               ;        POP     BC
 D9               ;NUM2:   EXX
 CB 78            ;NUM21:  BIT     7,B             ;ZAHL MIT DEZ.-PUNKT ?
 20 01            ;        JR      NZ,NOPNT
 04               ;NUM22:  INC     B               ;DPL = DPL + 1
 DD 23            ;NOPNT:  INC     IX              ;ADR2 = ADR2 + 1
 0D               ;        DEC     C               ;CNT = CNT -1
 D9               ;        EXX
 20 DD            ;        JR      NZ,NUM1
 33               ;NUM3:   INC     SP              ;ADR1 VERNICHTEN DA KON-
 33               ;        INC     SP              ;VERTIERT WERDEN KONNTE
 E5               ;        PUSH    HL              ;D1 AELEGEN
```

```
D5              ;         PUSH    DE
D9              ;         EXX
48              ;         LD      C,B
06 00           ;         LD      B,00
C5              ;         PUSH    BC            ;DPL AUF DEN STACK
01 FF FF        ;         LD      BC,0FFFFH     ;TRUE = ERFOLG GEHABT
C5              ;         PUSH    BC
18 1C           ;         JR      NUM4          ;FERTIG !
DD 7E 00        ;TSTP:    LD      A,(IX)
FE 2E           ;         CP      '.'           ;DEZIMALPUNKT ?
20 08           ;         JR      NZ,TSTH
D9              ;         EXX
00              ;         NOP
CB 78           ;         BIT     7,B           ;MEHR ALS EIN DEZI-
28 0B           ;         JR      Z,ERROR       ;  MALPUNKT
18 DB           ;         JR      NUM22
FE 48           ;TSTH:    CP      'H'           ;BASE = HEX ?
20 05           ;         JR      NZ,ERROR      ;WENN NICHT -> ERROR
D9              ;         EXX
0D              ;         DEC     C
D9              ;         EXX
28 D9           ;         JR      Z,NUM3        ;'H' STEHT NICHT HINTEN !
01 00 00        ;ERROR:   LD      BC,0000
C5              ;         PUSH    BC            ;FLAG = FALSE, KEIN ...
CEND            ;NUM4:    JP      NEXT          ;... ZIFFERNSTRING !

PROGRAM NUMBER  ;(adr1 -> d1    true) wenn Zahl
   DUP PARSE    ;(adr1 -> adr1 false) keine Zahl
   IF
      SWAP DUP
      ?(B) 45 =
      DUP >R
      IF
         INC SWAP DEC
      ELSE
         SWAP
      ENDIF
      ?BASE (NUMBER)
      IF
         DPL :=
         R> DUP DSIGN
         TRUE
      ELSE
         TRUE DPL :=
         R> DROP
         FALSE
      ENDIF
   ELSE
      FALSE
   ENDIF
END

PROGRAM D.
   NUM :=(D)
   NUM PRINTD
END

PROGRAM S.
   NUM :=
   NUM PRINT
END

;====================== E N D =========================
```

Ein Segen für die Menschheit

Der Computer gilt als neue industrielle Revolution. Als etwas, das in den nächsten Jahren unser Leben so total verändern wird wie noch keine Erfindung zuvor.
Tatsächlich haben die Minichips unsere Welt mehr durcheinandergebracht als Präservative, Trockenmilch und Hüfthalter zusammen. Täglich können wir die wunderbaren Erleichterungen spüren, die durch die neue Technik in allen Lebensbereichen eingeführt werden:

- Ist Ihr verstorbener Opa in diesem Jahr auch schon dreimal zur Musterung vorgeladen worden?
- Stehen auf Ihrer Telefonrechnung seit neuestem Gespräche nach Addis Abeba?
- Soll Ihr Hund sich als Schöffe zur Verfügung halten?
- Werden auf Ihrem Konto öfter mal die Versicherungsbeiträge doppelt abgebucht?
- Bekommen Sie immer häufiger mit der Post Angebote über Sprachkurse, Sexfilme oder Parteispenden?
- Haben Sie schon mal einen Flug nach Detmold gebucht und ein Ticket nach Dallas bekommen?

Wenn Sie solche oder ähnliche Erlebnisse hatten, können Sie sich glücklich schätzen, dann haben Sie etwas vom Hauch der »industriellen Revolution« gespürt. Denn schuld an der ganzen Misere ist kein anderer als Kollege Computer.
In diesem Punkt sind sich die Wissenschaftler ausnahmsweise einig: Die elektronischen Pfiffikusse sind nicht nur die größten Datenverwalter der Geschichte, sondern sie sind auch die Weltmeister, wenn es darum geht, irgendeine Sache gründlich zu vermasseln.
Nach dem alten Motto: »Wer viel weiß, kann auch viel vergessen« sind schon so einige Datenbanken in den düsteren Keller des elektronischen Vergessens abgestürzt.
Da hat die Maschine dem Menschen wirklich viel Ärger abgenommen. Wenn früher ein Mitarbeiter Fehler machte, konnte der Chef wenigstens noch den Zeigefinger heben und mit Gehaltskürzung drohen. Beim Computer nützt nicht einmal mehr die Versetzung in die Poststelle.
Dabei kann der Schnelldenker ungleich mehr Schaden anrichten als sein lebendiger Kollege. Wenn man nur an die täglichen Fehlalarme denkt, mit denen die Pentagon-Computer seit Jahren vergeblich versuchen, den dritten Weltkrieg auszulösen.
Kurz, diese geniale Erfindung aus Chips und Kabel hat sich inzwischen in der zivilisierten Welt unentbehrlich gemacht.
Es ist also völlig falsch, diesen Gipfel aller menschlichen Errungenschaften zu verteufeln: Betrachten wir doch nur einmal die wunderbaren Vorteile, die wir dem täglichen Umgang mit unseren elektronischen Helfern in Büro und Wohnstube verdanken: heftige, ununterbrochene Kopfschmerzen, Augenflimmern, verkrampfte Finger und nicht zu vergessen die reizenden Piepstöne, die seit neuestem unsere Umgebung erfüllen. Wer wollte da noch auf den Computer verzichten?

Besser erkennbare Darstellung von Text und HRG auf dem Bildschirm

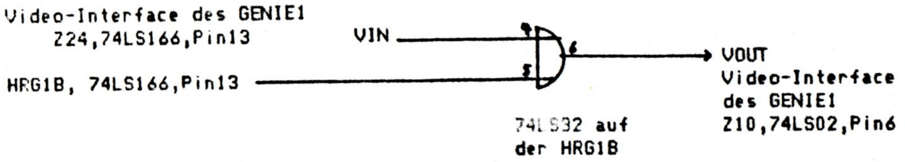
Helmut Bernhardt

Durch die Erzeugung synchroner Videosignale des normalen Text-Video-Interface und der HRG1B konnten diese beiden Signale gemischt werden, so daß beide Bildschirm-Informationen gleichzeitig dargestellt werden. Das ist einerseits ärgerlich, wenn z.B. BASICs READY> ständig nach der Erzeugung eines Bildes das Werk verunziert; es kann andererseits aber auch recht nützlich sein, wenn man zu Diagrammen erläuternde Texte anzeigen möchte. Weniger gelungen ist aber die Art, wie die Video-Signale gemischt werden. Das wird auf der HRG1B durch OR-Verknüpfung der beiden Signale gemacht, was zur Folge hat, daß in hellen Flächen der HRG-Anzeige kein Text zu erkennen ist. Die OR-Verknüpfung bewirkt, daß ein Bildpunkt immer hell gesteuert wird, wenn entweder der entsprechende Punkt in der HRG oder der entsprechende Punkt eines dargestellten Textzeichens oder auch beide angezeigt werden.

Eine XOR-Verknüpfung der beiden Video-Signale steuert einen Bildpunkt nur dann hell, wenn entweder der HRG-Punkt oder der Punkt eines Textzeichens angezeigt wird. Wenn allerdings beide Video-Signale gleichzeitig einen Bildpunkt hell darstellen wollen, wird dieser dunkel gesteuert. Das hat zur Folge, daß Texte vor hellem HRG-Hintergrund dunkel dargestellt werden. Selbst wenn ein Textzeichen zum/Teil vor einem hellen und zum Teil vor einem dunklen Hintergrund dargestellt wird, läßt es sich vollständig erkennen, weil sein Farbwert immer das Gegenteil der Hintergrundfläche ist.

Damit können dann nicht nur Text und HRG sehr viel besser gleichzeitig dargestellt werden; der Umstand läßt sich bei geschickter Programmierung der HRG auch dazu nutzen, inverse Zeichen darzustellen. Es müssen einfach immer die entsprechenden Felder, in denen im Text-Bildschirm ein Zeichen dargestellt wird, in der HRG hell gesteuert werden. Das bedeutet, daß pro auszugebendem Textzeichen auch 12 Bytes im HRG-Speicher auf FFH zu setzen sind. Die Zuordnung von Text-Video-RAM-Adressen zu HRG-Adressen läßt sich aus früheren Beiträgen von Arnulf Sopp zu diesem Thema entnehmen. Etwas aufwendiger wird es beim Scrollen von Texten, wenn Teilbereiche des Textes invers dargestellt sind. Dann genügt es nicht, nur mit LDIR den ganzen Text-Bildwiederholspeicher um 64 Adressen zu verschieben, dann müssen auch die hellen Felder der HRG um einen entsprechenden Offset verschoben werden, was nicht mehr mit einem einfachen LDIR zu erledigen ist.

So aufwendig das in Soft auch sein mag, in Hard ist die Änderung auf XOR-Verknüpfung der Video-Signale recht banal. Die bisherige OR-Verknüpfung sieht so aus:



Wenn anstelle des OR-Gatters des 74LS32 ein XOR-Gatter des pinkompatiblen 74LS86 genommen wird, ist die Sache erledigt. Leider werden von dem 74LS32 noch andere OR-Gatter für andere Zwecke verwendet, so daß man nicht einfach das IC gegen ein 74LS86 auswechseln kann. Man kann aber das 74LS86 mit den Pins 4, 5, 7 und 14 huckepack auf das 74LS32 löten, den Pin 6 des 74LS32 abkneifen und stattdessen den Pin6 des 74LS86 mit dem Pin15 des Sockels für den Anschluß des 24poligen Flachkabels verbinden.

So einfach ist das.

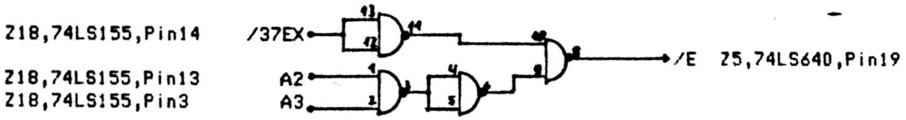
Endlich Schluß mit den Problemen beim EXP1-Floppy-Controller von RB-Elektronik

Helmut Bernhardt

Mehrmals hatte ich schon Probleme mit EXP1-Floppy-Controllern. Nach versuchsweisem Auswechlseln einiger ICs war der Spuk dann meistens vorbei - leider aber immer nur für begrenzte Zeit. Inzwischen habe ich einen grundsätzlichen Fehler in der Schaltung des EXP1 erkannt, der dafür verantwortlich ist:

Die Freigabe des invertierenden Datentreibers 74LS640 wird durch AND-Verknüpfung der Signale /37ECRD (Z18,74LS155,Pin4) und /37ECWR (Z18,74LS155,Pin12) erzeugt. Das hat zur Folge, daß bei Schreibzugriffen auf den Floppy-Controller WD1x91 der Treiber erst im gleichen Moment freigegeben wird, wie das /WR-Signal am FDC mit seiner negativen Flanke die Daten in den Controller pustet. Durch die Gatterlaufzeit bei der Freigabe des Treibers können hinter dem Treiber noch garkeine gültigen Daten anliegen. Es ist erstaunlich, daß das Board überhaupt funktioniert hat.

Abhilfe schafft ein Freigabesignal für den Treiber, das unabhängig von /RD und /WR ist. Dieses läßt sich mit einem 74LS00 folgendermaßen erzeugen:



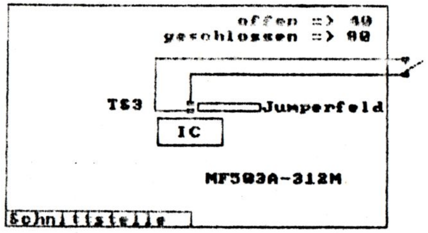
Das dafür nötige 74LS00 wird mit den Pins 7 und 14 auf irgendein anderes DIL14-IC (nur nicht das 74LS92) huckepack-gelötet und mit den entsprechenden Pinsvon Z18 und Z6 zu verdrahtet. Der Pin8 von Z15,74LS08 wird aus der Fassung gebogen.

Januar 1989

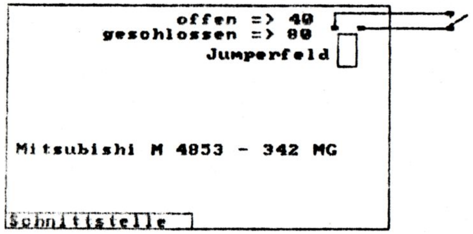
80/40-Trackumschaltung für Mitsubishi MF503A-312M

Wieder einmal ist mir ein auf 40 Spuren umschaltbares 80'er Laufwerk über den Weg gelaufen und wieder will ich kurz beschreiben, wie sich die Umschaltung bewerkstelligen läßt. Schließlich soll jeder, dem ein solches Laufwerk einmal unterkommt oder angeboten wird in der Lage sein, alle Fähigkeiten des Laufwerks für sich auszunutzen.

Wie sich schon der Überschrift entnehmen läßt, handelt es sich um ein 5¼"-Laufwerk der Firma Mitsubishi mit der Bezeichnung MF503A-312M. Es zeichnet sich durch Laufruhe und Betriebssicherheit aus (Jörg Brans ist schon länger im Besitz eines solchen Diskdrives). Nachteilig ist nur, daß das Laufwerk ein Auslaufmodell und damit wohl nicht mehr lange zu haben ist (außer eventuell Gebrauchtgeräte). Ein paar Drives gibt es allerdings noch, Bezugsadresse bei Jörg Brans!



Zunächst muß eine Leiterbahn zwischen den Lötpunkten "TS3" getrennt werden. Das bewerkstelligt man am besten mit einem sehr scharfen Messer. Wie der nebenstehenden Zeichnung zu entnehmen ist, wird dann ein einfacher Schließer an die beiden Lötpunkte angeschlossen. Dabei sollte man einen Lötkolben möglichst kleiner Leistung benutzen, da die Punkte sehr nahe bei einem in SMD-Technik ausgeführten Microcontroler liegen. Etwas Sorgfalt ist also angebracht.



Und weils so schön war, gleich noch eine Umschaltung. Dabei handelt es sich um ein Mitsubishi-Laufwerk mit der Typenbezeichnung M4853-342MG. Laut Jörg Brans ist es das Vorläufermodell des oben beschriebenen Drives. Die Umschaltung funktioniert im Prinzip genau wie oben beschrieben, nur die Lötpunkte sind an einer anderen Stelle. Wo, könnt Ihr nebenstehender Zeichnung entnehmen.

Hartmut Obermann

Entfernung der Kassettenschnittstelle von der I/O-Karte des Genie IIs

Vor einiger Zeit sah ich mir den Schaltplan der I/O-Karte meines Genie IIs an und mußte dabei feststellen, daß sich gut die Hälfte der Zeichnung auf die Kassettenschnittstelle bezog. Diese habe ich noch nie benutzt und werde es nie tun. Außerdem hängt an diesem Wust auch noch die Lautsprecher-Ansteuerung; da ich keine Lust habe, den Rechner zum piepsenden Nervtöter zu machen, ist dieser Teil auch überflüssig.

Welchen Sinn hat die Aktion? Ich hatte gerade eine Erweiterung der I/O-Karte geplant (s. Artikel "Memory mapped-I/O <=> Port-I/O") und mußte dazu ein 74LS157 auf der Karte unterbringen. Das wäre normalerweise nur im Huckepack-Verfahren möglich gewesen, wobei ich die neuen Strippen oben von IC-Pin zu IC-Pin hätte ziehen müssen. Das wäre ein undurchdringlicher Dschungel von verknüpften ICs geworden. Durch die Entfernung der Kassettenschnittstelle schaffte ich mir einen freien IC-Sockel, in den das 74LS157 paßte. Somit konnte ich alle Verbindungen auf der Lötseite herstellen, was auch einen Dschungel ergibt, wobei aber die ICs frei zugänglich bleiben. Außerdem lassen sich jetzt durch die Entfernung einer Menge Widerstände u.ä. die Leiterbahnen sehr viel besser verfolgen. Natürlich können Euch andere Gründe zu der Freisetzung ungenutzter Elektronikteile veranlassen.

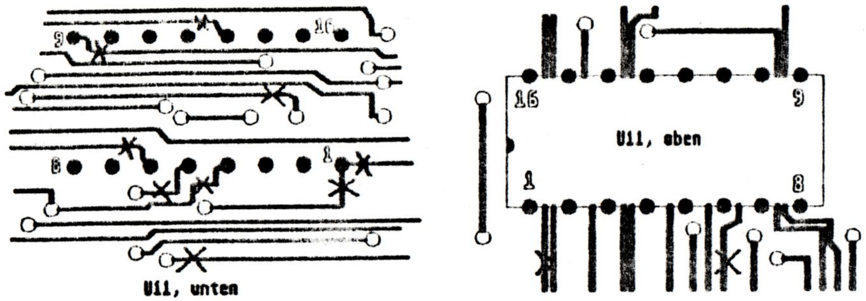
Im Einzelnen können ohne Verlust der anderen Leistungen des Systems folgende Elemente von der I/O-Karte entfernt werden:

ICs	: U11; U12; U2 Pins 6,7,9-12; U18 Pins 4-6,8-10
Widerstände	: R 4-19
Kondensatoren	: C 2-5,16
Dioden	: D 1-5
Transistoren	: T 1,2
Reed-Relais	: RE 1
Stecker	: S 1,4

Bis auf die Stecker habe ich wirklich alles entfernt bzw. die entsprechenden Pins von U2 und U18 herausgebogen. Übrigens würde ich sagen, daß sich mit den freien Pins von U2 z.B. eine zweite Centronics-Schnittstelle basteln ließe. Das wäre eine sinnvollere Anwendung als die Kassetten-Schnittstelle.

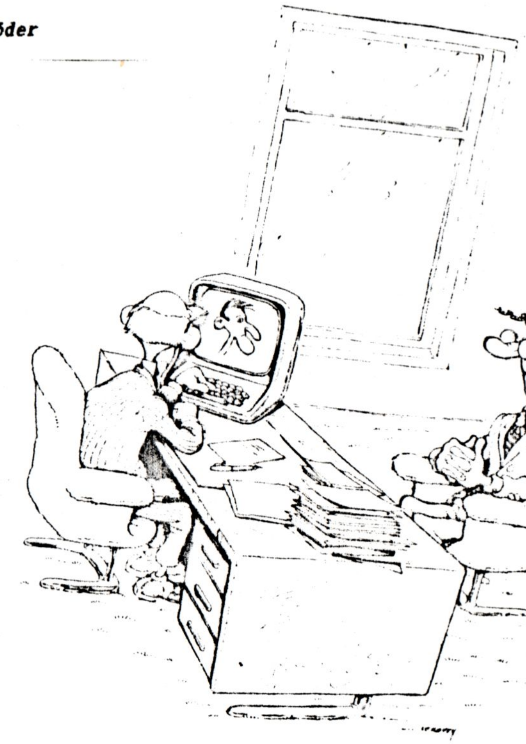
Da ich die Fassung von U11 für meine Zwecke nutzen wollte, mußten alle Verbindungen gekappt werden (bis auf GND und +5V). Seht Euch dazu die folgende Liste und die Bilder an:

Signal	Pin von U11	wo gekappt (oben=Bestückungsseite)
SYSRES*	1	unten, zweimal direkt am Pin
FFOUT*	9	unten, direkt am Pin
PH0	2	unten, an der Durchkontaktierung darüber
PH1	6	unten, direkt am Pin
MOTON	10	unten, an Durchkontaktierung Richtung U10
32C*	14	an der Steckerleiste, Anschluß A8
D0	4	unten, direkt am Pin
D1	5	unten, direkt am Pin
D2	12	oben, in Richtung U10
D3	13	oben, in Richtung U10, und unten, direkt am Pin



Leider mußten an zwei Stellen (SYSRES* und D3) zwei Leitungen unterbrochen werden, d.h. eine führte zu U11 und eine wieder weg. Diese unterbrochenen Verbindungen müssen überbrückt werden: Für SYSRES* von A14 (Steckerleiste) oder U8,1 (= Pin 1) oder U1,1 zu U2,3. Für D3 von C19 (Steckerleiste) zu U4,12.

Gerald Schröder



Tut mir leid – Sie entsprechen nicht unserem Anforderungsprofil.

Memory Mapped I/O <=> Port I/O beim Genie IIs

Beim TRS-80 und den meisten kompatiblen Geräten wird die Ein/Ausgabe, namentlich für Tastatur, Bildschirm, Drucker und Disk, über eine Memory Map gemacht, die bei 3700h bis 4000h liegt. Das ist mit Newdos zusammen ganz OK so, denn der Speicher wird von den Programmen erst ab 5200h benutzt. Wenn man den Rechner aber für CP/M umrüstet, gibt es Probleme, denn hier benutzen die Programme den Speicher ab 100h. Nun liegt die Memory Map plötzlich mitten im RAM rum und stört ziemlich. Beim TRS-80 gibt es eine Erweiterung, die diesen Bereich an das obere Speicherende verschiebt, wo er nicht mehr so sehr stört. Bei anderen Systemen, so auch beim Genie IIs, wird einfach die Memory Map gegen RAM ausgetauscht und nur eingeschaltet, wenn eine Ein- oder Ausgabe erfolgen soll.

Dies Umschaltung birgt einige Probleme: Das umschaltende Programm darf nicht in diesem Bereich laufen, sonst dreht es sich selbst den Hahn ab; der Stackpointer könnte gerade in diesen Bereich zeigen, da aber beim Umschalten das RAM dort verschwindet, muß er umgestellt und später restauriert werden. Außerdem dürfen nun keine Interrupts mehr stattfinden, denn auch die könnten in den gesperrten RAM-Bereich führen. Wenn nun ineinander verschachtelt mehrere Routinen auf die Memory Map zugreifen wollen, ist ein ganz schöner Verwaltungsaufwand erforderlich, um festzustellen, wo der SP gerade steht, ob er gerettet werden soll, ob die Memory Map eingeschaltet werden muß, ob die Memory Map wieder eingeschaltet und der SP wieder restauriert werden darf.

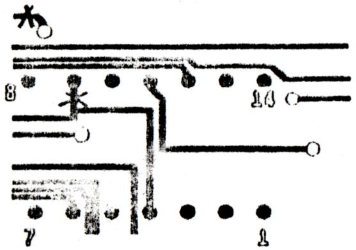
Was wird nun von der Memory Map eigentlich gebraucht? Der Bildschirm-Bereich ist oft überflüssig, denn meist wird für CP/M eine 80-Zeichen-Karte (oder eine Terminal-Karte) eingesetzt. Die Tastaturabfrage sollte ziemlich regelmäßig erfolgen, benutzt also sehr stark die Memory Map. Hier werden eigentlich nur 10 Adressen gebraucht (bei normaler Tastatur), aber eine miese Ausdekodierung bläht diese auf 1K verbratene Adressen auf. Die Drucker-Schnittstelle braucht nur eine Adresse und wurde bei manchen Genies schon in einen Port verlegt. Die Disketten-Ein/Ausgabe belegt 5 Adressen, aber durch die Ausdekodierung werden (mit Drucker) wieder 32 daraus. Diese Adressen werden bei Diskettenoperationen sehr intensiv genutzt.

Der Z-80 stellt 256 Ports zu Verfügung, die keinerlei Speicher belegen. Warum wurden die nicht benutzt? Gute Frage, vergeben ist beim TRS-80 nur einer (FFh), bei einigen Nachbauten noch FCh, FDh und FEh. Außerdem machen Erweiterungskarten (HRG 1b, Banker) Gebrauch von den Ports, aber die 256 werden trotzdem nur zum Bruchteil genutzt. Das Bildschirm-RAM brauche ich nicht und die Tastatur-Abfrage soll demnächst die Terminalkarte übernehmen, also bleiben nur Drucker und Diskette. Warum nicht diese in Ports verlegen?

Gemacht, getan. Nachdem die I/O-Karte gesäubert wurde (s. Artikel "Entfernung der Kassettenschnittstelle") hatte ich genug Platz für den Umbau. Es wäre einfach gewesen, wenn ich nur noch den Zugriff über die Ports haben wollte, aber leider sind BOOT-ROM, GDOS samt Programmen und momentan noch mein CP/M-BIOS auf die Memory Map eingestellt. Also muß eine Umschaltung vorgenommen werden, je nach Wunsch Ein/Ausgabe über Memory Map oder Ports.

Da ich nicht 32 Ports verbraten wollte, mußte der Kram erst soweit ausdekodiert werden, daß nur noch 16 Ports gebraucht werden (noch besser wäre natürlich eine Reduzierung auf 6, aber da wird's schwierig). Dazu braucht man nur ein OR-Gatter, das auch leicht zu bekommen ist: Entgegen dem Schaltplan ist das OR-Gatter von U19,8-10 überbrückt und sinnlos. Man kann stattdessen eine neue Verbindung von U16,8 zu C15 (Steckerleiste, WREQ*) ziehen. Alsdann wird die Brücke bei U19 entfernt und die entsprechenden Pins isoliert:

Signal	Pin an U19	wo getrennt?
WREQ*	8	unten, direkt daneben an Durchkontaktierung
PLUS	9	unten, direkt am Pin
noname	10	oben, links, neben +5V



U19, unten

U19, oben

Anmerkung:
Bei meinen Zeichnungen habe ich folgende Symbole benutzt:

- ● Lötpunkt (IC-Pin)
- ○ Durchkontaktierung
- ——— Leiterbahn
- X Unterbrechung einer Leiterbahn

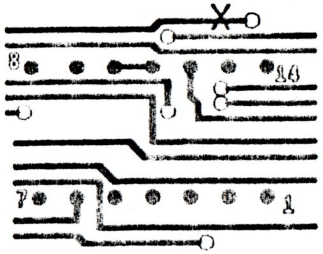
Nun kann die erste Änderung vollzogen werden: die Ausdekodierung auf 37Ex (vorher neben 37Ex auch 37Fx belegt = 32 Byte). Zuerst wird U22,13 getrennt (s. unten) und dann werden folgende Verbindungen hergestellt:

U20,9 -> U19,9 (AD4)
U20,6 -> U19,10 (FREE*)
U19,8 -> U22,13 (neu, nennen wir es xxEx*, im Schaltplan als PEX*)

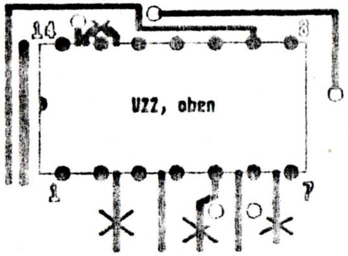
Nach dieser Änderung muß der Rechner noch genauso laufen wie vorher, denn der gespiegelte Bereich bei 37Fx wird meines Wissens nie benutzt.

Die zweite Änderung ist der eigentliche Umbau, aber der ist nach den Vorarbeiten nicht mehr schwierig. Zuerst werden folgende Verbindungen getrennt (die erste gehört noch zur ersten Änderung, s. oben):

Signal	Pin an U22	wo getrennt?
FREE*	13	oben, ganz links in Richtung U20
		oben, ganz rechts, in Richtung U20
37EX*	8	oben, an Pin 13
MWR*	1	unten, bei Durchkontaktierung über Pin 13/14
MRD*	5	oben, in der Mitte, Richtung U20



U22, unten



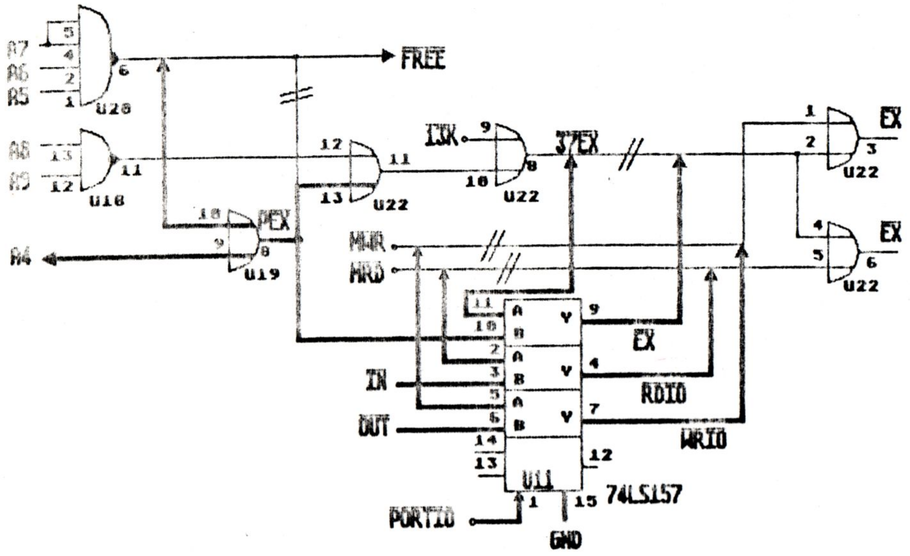
Nun wird das 74LS157 in die Fassung von U11 gesteckt und die Pins unten entsprechend der folgenden Liste verbunden. Es ist auch möglich, das IC auf ein anderes huckepack zu löten (nur Pin 8 und 16) und die Beinchen mit denen der entsprechenden ICs oben zu verbinden.

Signal	von...	nach...
FREE*	U17,12	U20,6
SPEAKER*	U11,1	U7,2
MRD*	2	A23 (Steckerleiste)
IN*	3	A21 "
RDIO*	4	U22,5
MWR*	5	A22 "
OUT*	6	A16 "
MMIO*	7	U22,1
EX*	9	U22,2
PEX*	10	U22,13
37EX*	11	U22,8
GND	15	U11,8

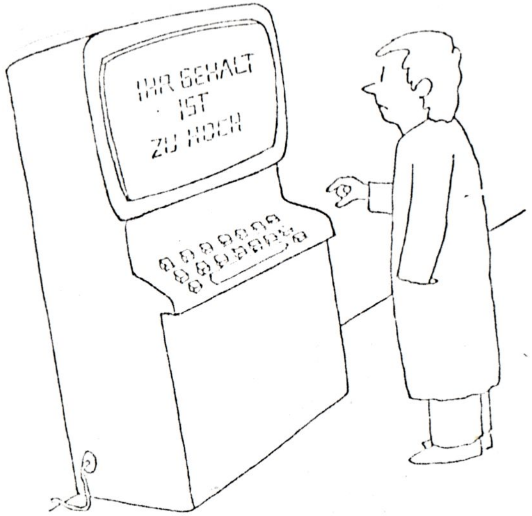
Wie Ihr vielleicht schon gemerkt habt, wird das Signal SPEAKER* zur Umschaltung zwischen Memory Map und Port I/O benutzt. Es hatte vorher die Aufgabe, einen evtl. angeschlossenen Lautsprecher an- und auszuschalten. Das Signal bot sich für meine Zwecke einfach an (im Schaltplan unten taucht es als PORTIO* auf).

Wie Ihr sicherlich wißt, habe ich mit diesem Artikel kein Publikum bzw. kann kaum Nachahmer finden (wer hat schon ein Genie IIs?). Aber ich möchte Euch anregen, Eure Kiste nicht einfach kritiklos hinzunehmen, sondern sie so zu verbessern, wie es Euch gefällt. Das brauchen nicht immer die großen Hardware-Freaks zu machen, jeder kann sich kleinere Sachen selbst überlegen. Dies war mein erster Versuch in Hardware und es hat mir immerhin Mut gemacht, es wieder zu versuchen.

Abschließend noch der neue Teil des Schaltplans, wie er von Helmut Bernhardt gezeichnet wurde:



Gerald Schröder



Darf's ein bisserl mehr sein ?
Ramdisk mit 512KB

Nachdem ich Euch kurz die c't-Solid State Disk vorgestellt hatte, nun etwas für die Feinschmecker. Platz ist dazu da, daß er ausgefüllt wird und so ist es mir dort langsam aber sicher zu eng geworden. Da auf einen vollen Bus keine zweite SSD paßt, habe ich mich als Alternative für die RAM- /EPROM-Floppy 'DISI-1' der Firma Conitec entschieden. Auf die passen mit RAMs immerhin 512KB und mit den neuen gebankten Eproms sogar bis zu 2MB (!). Die Ansteuerung ist fast identisch mit der der SSD, sodaß sie problemlos ausgetauscht werden kann.

An einem Donnerstag habe ich die Bestellung abgeschickt und pünktlich am Montag war das Päckchen per UPS da. Nach pfundweise Styroporflocken kam endlich ein in Antistatikfolie und dick Schaumstoff eingewickeltes zweites Paket zum Vorschein und darin war schließlich die Platine. Als begleitende Lektüre ist ein 5mm dickes, gebundenes Heft im A4-Format mit ausführlicher Anleitung und Schaltplan dabei und die Treiber-Software und einige Utilities kommen auf einer Floppy daher. Die Platine ist mit Präzisionssockeln und ICs fertig bestückt und macht einen sehr sauberen, wenn durch das Fehlen der RAMs (man kann sie auch mit RAMs bestellen) einen leeren Eindruck. Auf der Steckerleiste hat sogar jemand, ich nehme an, daß es der Entwickler ist, sein Autogramm hinterlassen. Die stehen wenigstens zu ihrem Werk. Daß ein IC (ein Treiber) gefehlt hat, darf einen bei dem fast geschenkten Preis von nur DM 412.- (incl. Mwst usw.) nicht weiter stören. Normalerweise hat man sowas ja sowieso irgendwo rumliegen und lieber sofort ohne IC, als mit und noch eine Woche warten.

Ein erster Test mit einem Eprom ging völlig in die Hose, beim Auslesen kam nur Käse raus. Ich weiß nicht, ob es an dem Eprom gelegen hat, oder ob ich die Jumper falsch gesteckt hatte, oder was auch immer, jedenfalls ging es nicht. Nach dem Einsetzen der CMOS-RAMs hat sie aber einwandfrei mitgespielt und bis jetzt sind keine weiteren Fehler aufgetreten.

An Signalen vom Rechner braucht die Karte lediglich acht Daten-, acht Adreßleitungen, WR, RD und IORQ, sodaß sich der Aufwand zum Anschließen an Fremdrechner sehr in Grenzen hält; gedacht ist sie natürlich für den ECB-Bus.

Wer viel Platz auf der Ramdisk braucht und wen der doch ziemlich hohe Preis nicht abschreckt, der ist mit der DISI-1 wirklich gut bedient. Im Gegensatz zur c't SSD muß man sich höchstens einen Adapterstecker bauen sie und funktioniert sehr sicher.

Alexander Schmid

Sharp 3541 — anwenderfreundlich und leicht erweiterbar

Mit dem jetzt auf den Markt gekommenen Business-Computer MZ-3500 ersetzt Sharp einen etwas glücklosen Abkömmling seiner Computerserie, den PC 3201. Schon auf den ersten Blick unterscheidet sich das neue Gerät vom Vorgänger: Zentraleinheit und die beiden Slimline-Floppylaufwerke sind in einem ansprechenden Gehäuse untergebracht, der dreh- und schwenkbare Bildschirm und eine abgesetzte, sehr flache ergonomische deutsche Norm-Tastatur zeigen, daß man den Weg vom klobigen Tastaturcomputer zum bequemen Arbeitsplatzsystem gefunden hat.

Kleider machen zwar Leute, aber wer die letzten Ausgaben von CP aufmerksam gelesen hat, weiß, daß zu einem modernen Computer mehr gehört als eine schöne Tastatur.

Sharp hat, und das war nicht anders zu erwarten, mit der Serie MZ-3500 ein grundsolides, auf bewährten Elementen aufgebautes Computersystem entwickelt. Im Gegensatz zur Mehrzahl der übrigen Arbeitsplatzcomputer, wie jetzt Rechner der Preisklasse zwischen 5000 und 15000 Mark vielfach genannt werden, ist der MZ-3500 nicht mit einem 16-Bit-, sondern mit zwei Z80A-8-Bit-Prozessoren ausgerüstet. Dazu gehören in der Grundversion dieser Reihe, dem MZ-3541, bereits 128 KB Hauptspeicher, zwei Slimline-Floppy-Disk-Laufwerke mit je 380 KB Kapazität, ein farbfähiger Bildschirmcontroller und eine gepufferte Tastatur mit eigener CPU.

Daß ein im Jahre 1983 auf den Markt gekommenes Computersystem nur mit 8-Bit-CPUs aufgebaut ist, verwundert im ersten Augenblick. Beschäftigt man sich mit den heute angebotenen 16-Bit-Systemen näher, fällt bald auf, daß 16 Bit nur im mathematischen Sinn das Doppelte von 8 Bit ist. Es fehlt noch an 16-Bit-Peripherie und ausgereifter Software. Auch wird es noch geraume Zeit dauern, bis die zwar auf 8-Bit-Systemen bewährten, aber im Schnellverfahren auf 16 Bit konvertierten Programme optimiert und den neuen CPUs angepaßt sind. Wen wundert da, wenn für viele 16-Bit-Computer nach kurzer Zeit Z80-Karten auf den Markt kommen, um, wie kürzlich in einer amerikanischen Computerzeitschrift zu lesen war, »aus diesen Sechzehnern endlich brauchbare Computer zu machen«. Dennoch hätte man dem MZ-3500 ein früheres Erscheinen gewünscht.

Viel Speicher

Beim MZ-3500 hat man sich augenscheinlich sehr viel Mühe mit der Ausgestaltung der Hardware gemacht. Um die Haupt-CPU nicht mit der Verwaltung des bis zu 256 KB umfassenden Speichers zu belasten, wurde ein besonderer Memory-Controller entwickelt; eine zweite Z80A-CPU übernimmt die meisten Ein-Ausgabe-Operationen und die Verbindung zu den beiden Grafikprozessoren. Auch die Tastatur mit einem 64 Zeichen umfassenden Eingabepuffer wird von einer eigenen CPU (80C49) kontrolliert.

Daß mit Speicher nicht kleinlich umgegangen wird, zeigt auch die Grafikerweiterung. Pro Seite werden 32 KB verwendet, was eine Auflösung von 640 x 400 Punkten erlaubt. Der Endausbau umfaßt drei Grafikseiten, die auch in Form von RGB-Speichern Farbausgabe erlauben.

Das Wichtigste ist serienmäßig eingebaut

Sharp scheint das bisher angewandte Verfahren, alle Schnittstellen getrennt vom Grundgerät zu verkaufen, aufgegeben zu haben und rüstet den MZ-3500 mit zwei gebräuchlichen Schnittstellen aus: einer seriellen RS232C-Schnittstelle und einem Druckeranschluß, der der Centronics-Norm entspricht. Beide Schnittstellen sind jedoch nur als Platinenanschlüsse auf der Rückseite der Zentraleinheit zugänglich, man benötigt also noch ein Spezialkabel, um ein Gerät mit Normsteckern anschließen zu können. Damit sind die Anschlußmöglichkeiten jedoch noch nicht erschöpft. Neben dem Hauptmonitor kann ein zweiter, getrennt ansteuerbarer Bildschirm angeschlossen werden, ebenso eine weitere Diskettenstation mit zwei Laufwerken. Auf der Vorderseite des Computers befindet sich die Anschlußbuchse für einen Lichtgriffel, das erforderliche Interface ist bereits eingebaut. Auch an der Tastatur befindet sich noch eine Buchse; leider war nicht in Erfahrung zu bringen, wozu sie einmal dienen soll. Leider waren die Verbindungskabel zwischen Zentraleinheit und Bildschirm, vor allem jedoch zum Drucker, sehr kurz bemessen, was die optimale Aufstellung des Systems etwas behindert.

Uhr mit Kalender

Gerade für den kommerziellen Anwender ist es wichtig, ständig über korrektes Datum und Uhrzeit zu verfügen. Dem wurde durch den Einbau einer batteriegepufferten Uhr mit Kalender Rechnung getragen. Auch ein Tongenerator für akustische Meldungen oder Bestätigungen ist eingebaut, er verfügt über einen Tonumfang von drei Oktaven.

Hochauflösende Farbgrafik

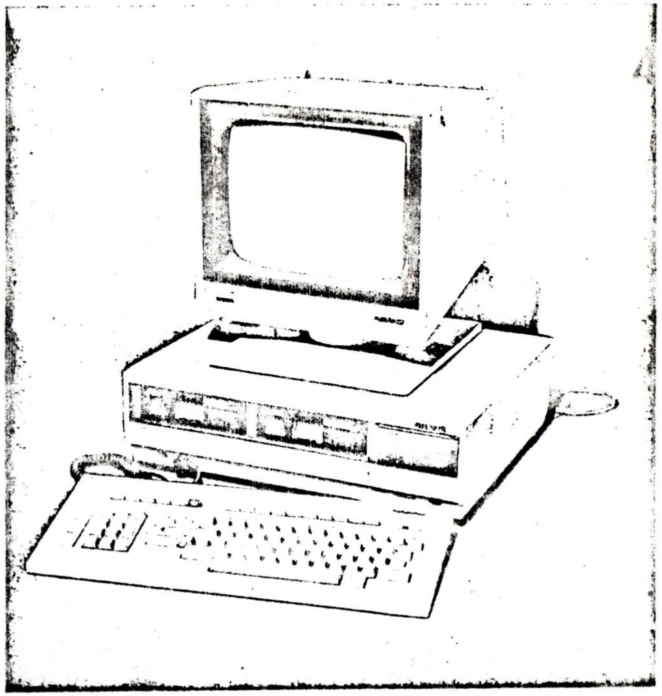
Ganz ohne Erweiterungen kommt jedoch auch der MZ-3500 nicht aus. Am wichtigsten sind wohl die Aufstockung des Hauptspeichers auf 192 beziehungsweise 256 KB und die Grafikerweiterung. Die Grafikplatine kann nach dem Entfernen einer Plastikabdeckung einfach unterhalb der beiden Floppylaufwerke eingesteckt werden. Sie enthält den Grafikcontroller und Steckplätze für insgesamt 96 KB dynamische RAM-Bausteine. In der Grundausstattung enthält diese Platine 32 KB RAM, also eine Grafikseite. Ausgebaut wird sie durch einfaches Einsetzen zusätzlicher Bausteine. Die Hauptspeichererweiterung findet leider im Rechnergehäuse keinen Platz mehr, man muß dazu einen Erweiterungsrahmen einbauen, der dann vier Zusatzkarten aufnehmen kann. Die RAM-Karte ist mit Sockeln für zwei Speicherbänke mit je 64 KB ausgestattet, die Erweiterung erfolgt auch hier durch einfaches Einstecken der Speicherbausteine.

Zur Erweiterung der Kommunikationsmöglichkeiten wird noch eine serielle Schnittstellenkarte mit zwei Kanälen angeboten; sie ist bei Verwendung des Computers als BTX-Terminal oder innerhalb eines Mehrplatzsystems erforderlich.

Servicefreundlicher Aufbau

Zu jedem Computertest gehört der Blick ins Innere; bei den Fähigkeiten des MZ-3500 ist dies natürlich besonders interessant. Die Abdeckhaube, wie das gesamte Gehäuse der Zentraleinheit aus Stahlblech, läßt sich nach dem Lösen einiger Schrauben leicht abnehmen. Darunter zeigt sich ein besonders übersichtlicher und damit servicefreundlicher Aufbau dieses Systems (Bild 1). Da der Rechner auf nur zwei Platinen aufgebaut ist, sind so gut wie keine Kabelverbindungen vorhanden; nur das getrennt aufgebaute und vollständig gekapselte Netzteil ist über Kabel mit der Hauptplatine und den Diskettenlaufwerken verbunden. Die Diskettenlaufwerke selbst können auf einem eigenen Chassis herausgeklappt werden und sind dann von allen Seiten zugänglich. Da sie auch in diesem Zustand betriebsbereit sind, können Einstell- und Reinigungsarbeiten leicht ausgeführt werden. Auf der rechten Seite ist nun auch der kleine Lüfter zu sehen, er ist bei geschlossenem Gehäuse kaum zu hören.

Die Systemerweiterung — das Testgerät wurde vom MZ-3530 (64 K, ein Laufwerk) zum MZ-3541 (256 K, zwei Laufwerke, Grafik) aufgerüstet — macht bei diesem Aufbau keine Schwierigkeiten; alle Steckhauben sind serienmäßig eingebaut, Kabelverbindungen sind nicht erforderlich, das Gehäuse muß nicht einmal zerlegt werden.



Zwei wichtige Teile der Hauptplatine sind auf der Unterseite des Geräts frei zugänglich: eine Schalterleiste zur Betriebsartenwahl und die Fassung des Zeichengenerators. Mit dem Betriebsartenschalter kann zum Beispiel die Polarität und Zahl der RS232-Quittungssignale eingestellt und die Bildfrequenz zur Ansteuerung eines hochauflösenden Grafikbildschirms erhöht werden. Auch kann hier von der Verwendung des Dezimalpunkts (bei numerischen Eingaben) auf Dezimalkomma umgeschaltet werden (!). Die Voreinstellungen werden von Basic und EOS übernommen, können jedoch softwaregesteuert verändert werden.

Alle selbständigen Baugruppen des Rechners (unter anderem RS232C-Schnittstelle, Uhr und Tastatur) erzeugen Unterbrechungssignale (Interrupts), die zum Beispiel von Basic in den Anweisungen ON TIME, ON KEY und POLLING verwendet werden. Damit wird der Aufbau von zeitabhängigen Programmteilen oder die Überwachung von Peripheriegeräten zum Kinderspiel.

Leider enthalten die Handbücher keinerlei Angaben über den Schaltungsaufbau oder die direkte Steuerung der Baugruppen. Der Entwurf eigener Systemsoftware, aber auch eigener Hardwareerweiterungen, dürfte damit so gut wie unmöglich sein.

Im ernsthaften kommerziellen Einsatz wird man wohl mit den beiden Minifloppy-Laufwerken trotz der recht hohen Kapazität bald nicht mehr auskommen. Zwei weitere 5¼-Zoll-Laufwerke sind für rund 3500 Mark erhältlich, für rund 7500 Mark gibt es ein 8-Zoll-Doppellaufwerk mit zusammen 2 MB Speicherplatz. Auch ein Festplattenlaufwerk ist lieferbar, es hat eine Kapazität von 10 MB und dürfte mit Controller zirka 10000 Mark kosten.

Viel Neues bei der Software

Der gute Eindruck setzt sich auch bei der mitgelieferten Software fort. Um dem Anwender ein breites

Spektrum an Möglichkeiten zu bieten, aber auch um Kompatibilität zu den Vorgängermodellen zu wahren, liegen dem MZ-3500 zwei Betriebssysteme und ein Basic-Interpreter bei. Es handelt sich dabei um das Sharp-eigene FDOS (Floppy-Disk Operating System) und das neu entwickelte CP/M-Plus-kompatible Betriebssystem EOS 3.0.

FDOS ist ein besonders anwenderfreundliches Betriebssystem. Von den acht FDOS-Befehlen befassen sich daher allein sechs mit dem Diskettenbetrieb, zum Beispiel mit der Initialisierung neuer Disketten und mit der Datensicherung. Mit dem FDOS-Befehl DEBUG kann ein kleiner Maschinenprogramm-Editor aufgerufen werden, dessen Anwendungsmöglichkeiten jedoch stark eingeschränkt sind, da sich der Zugriff auf den sogenannten Anwendermikrobereich (einem von Basic reservierten Maschinenprogrammbereich) beschränkt. Änderungen an Basic oder FDOS sind daher nicht möglich, der Endanwender wird daran jedoch auch kaum Interesse haben.

Unter FDOS muß auch der mitgelieferte Basic-Interpreter betrieben werden, er wird ganz einfach über den FDOS-Befehl Basic aufgerufen. Zu Basic gehören zwei deutschsprachige Handbücher mit über 400 Seiten Umfang. Beim Studium der implementierten Anweisungen und Funktionen zeigt sich bald, daß hier kein überflüssiger Aufwand betrieben wurde. Was hier aus dem Basic-Grundbefehlssatz, systembezogenen Anweisungen und einer großen Zahl aus anderen Hochsprachen entliehenen Sprachelementen zusammengefügt wurde, ist bemerkenswert.

Ebenfalls bemerkenswert, leider im negativen Sinn, ist die Tatsache, daß Basic nur über einen Zeileneditor verfügt, das heißt jede zu bearbeitende Zeile muß mit EDIT aufgerufen werden. Warum der sonst so komfortable Interpreter damit »gestraft« wurde, ist rätselhaft.

Neue Variablentypen

Indizierte Variablen (zum Beispiel A$(12)) und damit Arrays (definiert zum Beispiel mit DIM A$(100)) werden wohl in jedem Basic-Programm vorkommen. Um nun aus einem derartigen Array ein bestimmtes Element herauszusuchen oder alle Elemente auf Diskette abzuspeichern, war man bisher gezwungen, jedes einzelne Element mittels einer For-Next-Schleife zu bearbeiten. Sharp-Basic stellt dazu einen völlig neuen Variablentyp zur Verfügung, die »Kollektive Array-Variable« (zum Beispiel A$(*)). Diese neue Variable besteht aus dem mit DIM deklarierten Variablennamen und umfaßt alle Array-Elemente. Im folgenden Programmbeispiel wird der Unterschied deutlich:

```
Standard-Basic
OPEN "0",# 1,"PROBE"
FOR I=1 TO 100
PRINT# 1,A$,(I)
NEXT I
CLOSE

Sharp-Basic
OPEN "0",1,"PROBE"
BPRINT# 1,A$(*)
CLOSE
```

Der Inhalt der Datei PROBE ist in beiden Fällen derselbe.

Kollektive Array-Variablen können auch zum globalen Löschen (ERASE A$(*)) oder Suchen (SEARCH A(*),1,"NAME",A) verwendet werden. Die Anweisung SEARCH A$(*)",1,/"SCH-",A durchsucht das Array A$(x) nach Begriffen, die mit den Buchstabenkombination "SCH" beginnen. A enthält dann den Index des Array-Elements, das dieser Bedingung entspricht. Auch die Anweisungen MAXS (Suchen des größten Elements) dienen zum globalen Suchen.

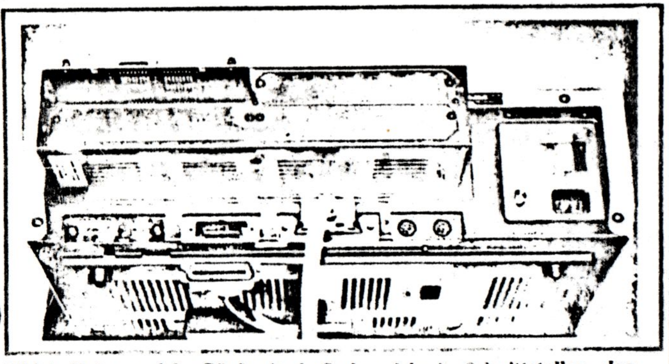


Bild 1. Unten auf der Rückseite befinden sich alle Schnittstellen, darüber der Erweiterungsrahmen mit der Speichererweiterung

Da Basic den gesamten Hauptspeicher (maximal 256 KB abzüglich Programm) als Variablenspeicher verwenden kann, müssen Länge und Typ der Variablen genau bestimmt werden. Dieser Umstand sorgt außerdem für sehr schnellen Zugriff und maximale Speicherausnützung. Basic sieht daher fünf Grundtypen vor: Integer- und Real-Variablen (12 Stellen, Exponent −99 bis +99), Stringvariablen des $-Typs (16 Stellen) und des §-Typs (16 Stellen, auch für Binärdaten mit Null-Elementen) und dimensionierte Stringvariablen mit beliebiger Stellenzahl (zum Beispiel X$(10)*200).

Lokale Variablen

Eine Besonderheit sind auch lokale Variablen innerhalb von Unterprogrammen, das heißt Variablennamen können innerhalb eines Programms mehrfach verwendet werden. Die Erstellung einer Modul-Bibliothek mit ständig verwendeten Routinen bietet sich deshalb natürlich an.

Erweiterter Diskettenzugriff

Neben den zum Beispiel aus Microsoft-Basic bekannten Dateitypen und Zugriffsarten wurden einige systemspezifische Sonderformen angefügt. So gibt es bei sequentiellen Dateien die Zugriffsform "A" (Append), um Daten an eine schon vorhandene Datei anzufügen, ohne die gesamte Datei vorher lesen zu müssen. Einmal erstellte Bildschirmmasken können mit den Anweisungen VSAVE und VLOAD gespeichert und wieder abgerufen werden. Ähnliche Befehle gibt es zum Speichern der Bildschirmgrafik. Sogar die Tastaturbelegung kann in speziellen Dateien gespeichert werden.

Password-Schutz von Programmen und Daten

Programme und Daten können auf Wunsch mit Schlüsselworten gesichert werden. Derart geschützte Dateien können weder kopiert noch aufgelistet werden.

Formatierte Ein- und Ausgabe

Neben den bekannten Anweisungen USING und IMAGE für die Datenausgabe enthält Basic sehr leistungsfähige Anweisungen zur Dateneingabe. Hier kann genau angegeben werden, welcher Datentyp eingegeben werden darf. Auch Länge und Format (zum Beispiel Anzahl der Dezimalstellen) der Eingabe kann bestimmt werden.

Leistungsfähige Schnittstellensteuerung

Sehr komfortabel ist auch die Steuerung der seriellen Schnittstelle ausgefallen. Zum einen können hier Drucker oder Plotter mit unterschiedlichen Übertragungsprotokollen angeschlossen werden, die Schnittstelle kann jedoch auch zur Kommunikation und zur Überwachung eingesetzt werden. Über einen Puffer und die Anweisung POLLING ist es zum Beispiel möglich, die Schnittstelle zu überwachen und erst bei Anliegen einer bestimmten Information zu reagieren. Zur Übertragung sind alle Variablentypen, auch kollektive Array-Variablen, zulässig. Als Besonderheit kann der MZ-3500 mit dem Befehl TERM in einen Terminalbetrieb versetzt werden und kann daher leicht in ein Mehrplatzsystem integriert werden.

Der Bildschirmkünstler

Schon die Grundgeräte (MZ-3530/MZ-3531/MZ-3541 ohne Grafikerweiterung) verfügen über vielfältige Möglichkeiten der Bildschirmgestaltung. Die Darstellung kann in vier Formaten erfolgen: 40 oder 80 Zeichen pro Zeile, die Zeilenzahl ist auf 20 oder 25 Zeilen einstellbar. Der 20-Zeilenbetrieb ist besonders übersichtlich und eignet sich vor allem zur Textdarstellung.

Jedes dargestellte Zeichen kann mit einer Reihe von Attributen versehen werden. Dazu gehört die Farbdarstellung mit getrennt wählbarer Vorder- und Hintergrundfarbe und die auch im Schwarzweiß-Betrieb mögliche Darstellungsart Blinken und Invertieren.

Jede Zeichenposition kann zusätzlich mit vertikalen und horizontalen Hilfslinien versehen werden. Mit Hilfe dieser Funktion können zum Beispiel leicht Eingabefelder oder Markierungen gezogen werden. Basic unterstützt dies mit dem Befehl TABLE. Zeichen und Attribute können getrennt voneinander gesetzt und gelöscht werden, was vor allem die Arbeit mit Bildschirmmasken erleichtert.

Bild 2. Schriften lassen sich in beliebigen Formen in die Grafiken übernehmen

Da noch eine einfache Blockgrafik (SET/RESET) und eine Vielzahl von Sonderzeichen vorhanden ist, lassen sich für den kommerziellen Anwender bereits ausreichende Grafiken (zum Beispiel Balkendiagramme) erstellen.

400 x 640-Punkt-Grafik mit eigenem Prozessor

Die als Option erhältliche Grafikkarte erweitert die Möglichkeiten nochmals beträchtlich. Die drei Grafikseiten mit einer Auflösung von je 400 x 640 Punkten werden im Farbmodus zum Speichern der drei Grundfarben Rot, Grün und Blau verwendet, im Schwarzweiß-Betrieb kann jede Seite getrennt angesprochen werden. Da ein Grafikprozessor zum Einsatz kommt, sind die in Basic implementierten zahlreichen Steueranweisungen sehr schnell und leistungsfähig.

Neben den Grundbefehlen zum Setzen und Löschen eines Punkts und dem Ziehen von Linien können Kreise und Ellipsen gezeichnet werden. Durch die Anweisung PAINT können definierte Bereiche in vier Helligkeitsstufen ausgemalt werden. Da die Bildschirmkoordinaten meist nicht mit den Werten der darzustellenden Funktionen harmonieren, kann mit der Anweisung SCALE ein beliebiger Koordinatenursprung und eine beliebige Skalierung eingestellt werden. Eine Umwandlung von Funktions- in Grafikkoordinaten ist dann nicht mehr nötig.

Textzeichen können in 90-Grad-Schritten gedreht in die Grafik übernommen werden, auch Vergrößerungen und Verkleinerungen sind möglich. Mit Hilfe eines über die Cursortasten beweglichen Grafikcursors lassen sich Gebiete eingrenzen, die dann mit Basic weiterbearbeitet oder an eine andere Stelle verschoben werden können. Der Grafikcursor kann aber auch zum Zeichnen mit verschiedenen Optionen verwendet werden.

Um die Inhalte der Grafikseiten zu Papier zu bringen, werden von Sharp verschiedene Matrixdrucker in der Preislage zwischen 1650 und 2900 Mark angeboten.

4-Farb-Tintenstrahl-Drucker als Zubehör

Als besonderen Leckerbissen gibt es noch einen Tintenstrahl-Drucker, der vier Farben darstellen kann. Mit ihm lassen sich auch Farbgrafiken originalgetreu wiedergeben. Die Betriebssysteme des MZ-3500 sind bereits mit den notwendigen Treiberroutinen versehen, so daß der Anschluß des Geräts keine Schwierigkeiten bereiten dürfte.

EOS 3.0 — das deutsche CP/M: keine Probleme mit Standard-Software

Als zweites Betriebssystem verfügt der MZ-3500 über das in Deutschland entwickelte EOS 3.0. Dieses System entspricht in der Grundstruktur CP/M-Plus, kann also auch über 64 KB hinausreichende Speicherbereiche verwalten. Dies zeigt sich schon daran, daß 63 KB als freier Benutzerspeicher ausgewiesen werden. EOS simuliert auf dem MZ-3500 zwei gebräuchliche Terminaltypen, das ADM-3A von Lear-Siegler und das TeleVideo TV950. Damit bestehen für die Anpassung von Standard-Software praktisch keine Probleme, da eines der beiden Terminals sicher in den Installationsprogrammen aufgeführt ist. Entsprechende Versuche mit Wordstar, Wordmaster, T/Maker, dBase und Multiplan verliefen erfolgreich.

EOS blendet im TV950-Modus ständig eine Statuszeile ein; sie enthält Uhrzeit, Datum und Informationen über den Systemzustand. Teile dieser Statuszeile können über das Anwenderprogramm definiert werden.

Besonders angenehm fiel die leichte Belegbarkeit der Tastatur auf; alle Tasten bis auf CTRL und ESC können ohne Zusatzprogramm direkt über die Tastatur neu definiert werden. Mit einem weiteren Tastendruck kann die Originalbelegung wiederhergestellt werden.

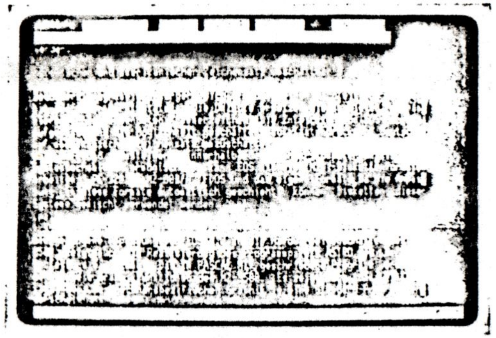


Bild 3. CP/M-Standardsoftware, hier WordStar, läßt sich einfach anpassen

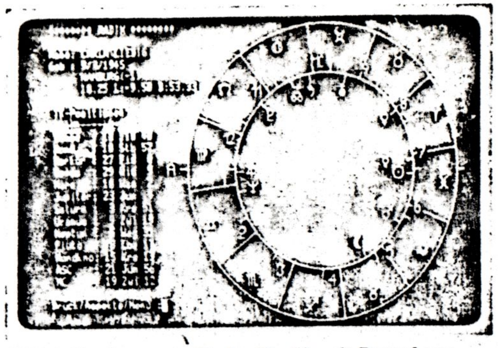


Bild 4. Rechnerspezifische Grafik mit Pascal am Beispiel eines astrologischen Programms

Bei der Arbeit mit Wordstar und Wordmaster stellten sich in Verbindung mit der sonst sehr angenehmen Tastatur zwei Probleme heraus: Da die ENTER-Taste recht klein ausgefallen und sehr dicht bei den Cursortasten plaziert ist, kommt es bei schnellem Schreiben oft zu unbeabsichtigtem Auslösen dieser Taste. Das zweite Problem zeigte sich bei der Arbeit mit Micropro-Produkten, also mit Wordstar und Wordmaster. Wird hier eine Steuerfunktion, zum Beispiel eine Cursortaste, zu lange betätigt, laufen die entsprechenden Funktionen nach, der Cursor fährt zum Beispiel weit über das Ziel hinaus. Im schlimmsten Fall kann das zum »Absturz« des gesamten Systems einschließlich Datenverlust führen.

Obwohl dieses Phänomen sicher mit der Tastaturpufferung zu tun hat, tritt es bei anderen Programmen, zum Beispiel TMAKER, nicht in Erscheinung. Vielleicht sollte hier eine abschaltbare Tastaturpufferung vorgesehen werden.

Deutscher und amerikanischer Zeichensatz umschaltbar

Besonders Programmierer ärgern sich oft über einen Nachteil von Rechnern mit deutscher Normtastatur: Die Umlaute ersetzen die in vielen Programmiersprachen verwendeten Klammerzeichen. Die Entwickler von EOS haben da aufgepaßt und die Möglichkeit geschaffen, auf Tastendruck von deutschem auf amerikanischen Zeichensatz umzuschalten.

Um Auf- und Umsteigern innerhalb der Sharp-Computerfamilie den Systemwechsel zu ermöglichen, kann EOS mit vier Mini-Diskettenformaten arbeiten: PC-3201, MZ-80B/A, FDOS und das eigene Format. Da die Anpassung an diese Formate automatisch geschieht, können Disketten dieser Systeme gemischt verwendet werden.

Die gelungene Anpassung von EOS 3.0 an die Hardwareeigenschaften des MZ-3500 zeigt auch die Realisierung des Programmpaketes Astro des Berliner Professors Gerd Klemm. Das in Pascal MT+ geschriebene Programm verwendet sehr viele Grafikfunktionen des MZ-3500. Es zeigt auch das einfache Einbinden der rechnerspezifischen Grafik in eine Standard-Programmiersprache. Astro erlaubt auf dem MZ-3500 die Berechnung und grafische Darstellung beliebiger astrologischer Daten auf der Grundlage einer systemeigenen Datenbank, die pro Diskette zirka 1800 Patienten aufnehmen kann.

EOS selbst stellt in vielem eine Verbesserung von CP/M dar. Besonders interessant ist die Tatsache, daß ein Diskettenwechsel nicht mehr angemeldet werden muß, ebenso gehört die stupide Meldung »BDOS-ERROR ON A:« der Vergangenheit an. Man erhält eine genauere Fehlerbeschreibung und kann im Dialog entscheiden, was weiter passieren soll. Ein weiterer großer Fortschritt ist der Bildschirmeditor von EOS. Bei Eingabefehlern muß also nicht mehr die ganze Zeile neu eingegeben werden, auch müssen fast gleiche Eingaben nicht ständig wiederholt werden. Auch in der Dateiverwaltung hat sich einiges getan. Durch ein sogenanntes Stamping werden Datum und Uhrzeit der Dateierzeugung und der letzte Zugriff im Inhaltsverzeichnis mitgeführt. Zusätzlich können Attribute angegeben werden; die Organisation von Diskettendateien wird dadurch erheblich erleichtert.

Dem lobenswerten Vorsatz, Befehle und Meldungen von EOS völlig in deutsch zu halten, ist man in der beim Test vorliegenden Version leider nur teilweise gefolgt. Es ist zu hoffen, daß in der Endversion dieser Schönheitsfehler behoben ist.

Fazit: uneingeschränkt empfehlenswert

Mit dem MZ-3500 ist ein Rechner auf dem Markt, der sich gleichermaßen für den Endanwender wie für den Programmierer eignet.

Mit FDOS und Basic kann sehr leistungsfähige systemnahe Software erstellt werden, EOS erlaubt die einfache Anpassung des umfangreichen CP/M-Programmangebots.

Damit kann zum sehr günstigen Systemgrundpreis von rund 7900 Mark (MZ-3541; zuzüglich MwSt.) eine sozusagen schlüsselfertige Problemlösung zusammengestellt werden, die vom Anwender keine Programmier- und kaum EDV-Kenntnisse fordert.

Dem Programmierer, aber auch dem anspruchsvollen Amateur, bietet sich vor allem in Verbindung mit der Grafikerweiterung und EOS 3.0 ein leistungsfähiges Instrument an, das wohl den meisten Ansprüchen gerecht werden wird.

Ansatzpunkte zu berechtigter Kritik ließen sich, von ein paar sicher zu behebenden Schönheitsfehlern, weder bei Hard- noch Software finden. Das sehr gute Preis-Leistungs-Verhältnis tut ein übriges, den MZ-3500 uneingeschränkt empfehlenswert zu machen. (Uwe Pansow)

Hallo Clubfreunde,
mein gutes altes Genie kommt nun doch so langsam in die Jahre und so stellte sich die unvermeidliche Frage: Neukauf, Ja oder Nein ? Nun versieht seit ca. 4 Monaten ein ATARI 1040 ST seinen Dienst bei mir. Dieser Zeitraum erscheint mir ausreichend, um einmal einen Erfahrungsbericht für das Club-Info zu verfassen. Um dem Verdacht der Einseitigkeit vorzubeugen, habe ich aufgrund eines Hinweises von Hartmut Obermann, noch einige andere Clubmitglieder mit einem ATARI-Rechner angeschrieben und zur Mitwirkung an einem Erfahrungsbericht aus der Sicht des jeweilig betroffenen aufgefordert. Vorgeschlagen hatte ich dabei folgende Punkte:

a) Der Anlaß zum Kauf

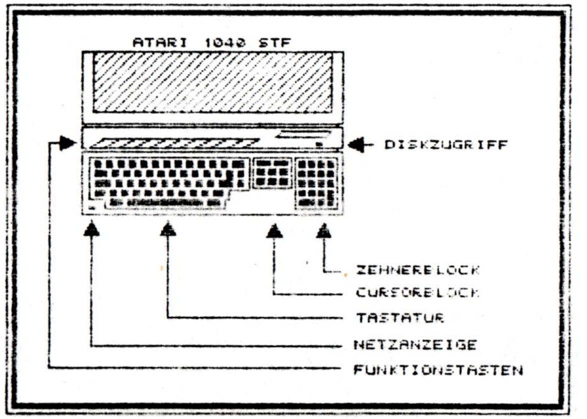
b) Erste Erfahrungen im Betrieb (Tastatur, Betriebssystem, Mausbenutzung etc.)

c) Softwareangebot und deren Qualität (Public Domain, käufliches)

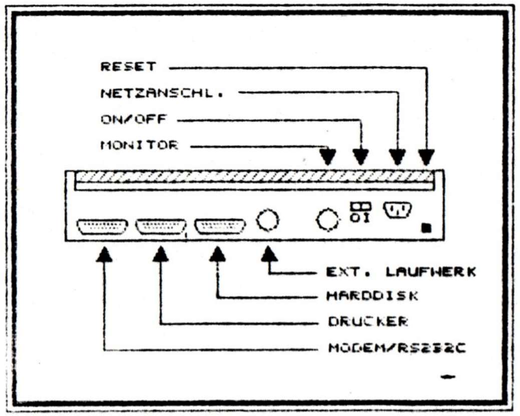
d) Ein Vergleich Tandy Model I (Genie I,II,III) (<->) ATARI) Hardware und Software

Was dabei herausgekommen ist, kann entweder in dieser, der nächsten oder auch schon in der vorherigen Info-Ausgabe nachgelesen werden. Hier nun mein Betrag:

Der erste Unterschied zu meinem alten Genie ist die deutsche Tastatur und Zeichensatz, nebst einer größeren Anzahl von zusätzlichen Tasten. An die Tastatur muß man sich gewöhnen, da die Federwirkung mit zunehmendem Druck auf die Taste zunimmt ! Im Grunde nicht weiter tragisch, führt jedoch bei schnellem schreiben dazu, daß Buchstaben verschluckt werden. Man darf also hinterher ausgiebig danach suchen. Ich habe mir deshalb angewöhnt, kräftig in die Tasten zu greifen. Ansonsten macht die Tastatur einen brauchbaren Eindruck. Am Gehäuse stören mich eigentlich mehr die reichlich großzügig ausgelegten Aussparungen für die Tastenfelder. Hier hätte man durch eine präzisere Herstellung des Gehäuseoberteils glänzen können. Zum Punkt Schnittstellen gibt es nicht viel zu sagen, es ist alles vorhanden, was so im normalen Privatbetrieb gefragt ist (siehe Zeichnung). Das der interne Bus außen nicht verfügbar ist, sollte man wissen. Wer gern zusätzliche Hardware benutzt, könnte Probleme bekommen. Wie mir aus der c't bekannt ist, läßt sich jedoch der ROM-Port hervorragend für



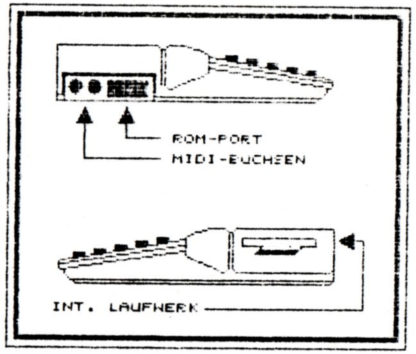
diese Aufgabe mißbrauchen. Bei mir steckt an dieser Stelle ein Uhrmodul (die gibt es aber mittlerweile auch als Einbauteil !). Da ich außer dem Centronics-Port keine der anderen Schnittstellen benutze, bleibe ich den Kommentar hierzu schuldig. Soweit mir bekannt ist, handelt es sich beim Centronics-Anschluß um einen ganz gewöhnlichen 8-Bit Port des internen Sound-Chips. Wer seinen Drucker über die serielle Schnittstelle ansteuern kann, hat so einen 8bit breiten I/O-Anschluß zur Verfügung (mit der Möglichkeit diesen auch als Eingang benutzen zu können).



Das eingebaute 3.5" Diskettenlaufwerk arbeitet recht ordentlich. Lediglich in letzter Zeit hatte ich beim Spiel STARGLIDER Probleme mit den innersten Spuren (80... 82). Hier kam es wiederholt zu Ladefehlern. Leider benutzt der Hersteller diese, um eine Art Kopierschutz zu erzeugen. Die normale Copy-Funktion des Betriebssystems erzeugt nur unbrauchbare Kopien, da Spuren ab >80 wohl nicht mehr korrekt Kopiert werden. Ansonsten muß ich sagen, das mir das 3.5" Format der Disketten mehr zusagt, als die 5.25" meines Genie's. Unglücklicherweise ist es jedoch mit der Speicherkapazität so eine Sache. Die erreichten 720K sind von der Wirkung her wie 360K bei den 8-Bittern. 16-Bit Rechner verbrauchen selbst bei "Ein-Byte Befehlen" grundsätzlich die doppelte Menge an Speicherplatz ! Der andere eigentlich schmerzlichere Punkt ist die unerwartete Speichergefräßigkeit der Anwendungsprogramme: Textverarbeitung ca. 160K Byte, Grafik-Programme ca. 150K Byte, in Assmebler geschriebene Programme (GFA-Basic und Compiler) ca. 60K Byte. Bei solchen Zahlen ist es dann kein Wunder, wenn die 720K dahinschmelzen und nach 4 bis 5 Programmen aufgebraucht sind ! Die "Speichergiganten" enstehen meist aus dem Umstand, daß die Programme in C oder anderen Hochsprachen geschrieben wurden. Die Effizienz solcher Wege ist ja ausreichend bekannt. Von dieser Tatsache einmal abgesehen, ist der verfügbare Platz für Texte und sonstigem "Spielkram" ausreichend. Es gilt sich ebend anzupassen: Basic auf eine Disk, Textverarbeitung auf eine Disk...

Über den Monitor kann ich nur positives sagen. Das Bild ist überzeugend ruhig und die Auflösung und Bildschärfe ist einfach super. Lediglich die Lösung, den Aus-Schalter an den Lautstärker-Regler zu ketten und diesen dann auch noch an die rechte Seite des Monitors zu verbannen, kann ich mir nur als Eigenwilligkeit des Designers erklären. Für den Alltagsbetrieb ist diese Lösung Käse. Ein Druckschalter im Monitorfuß und von Vorn zu bedienen, wäre der angenehmere Weg gewesen.

Dann wäre da noch das Betriebssystem. Mit stattlichen 192K ein echter Elephant (NewDOS80 ist, soweit ich mich erinnere, mit allen Overlays auch so um die 100K groß !). Dafür kann sich das innenleben allerdings sehen lassen. Während der eigentliche Betriebssystem-Teil (TOS genannt) recht simpel gehalten ist, sind die vorhandenen grafischen Mittel spitze. Es gibt so gut wie alles, was das Herz begehrt: Linien, Kreise, Rechtecke, Polygone und diesen ganzen Klimbim um Sprites beliebige Zeichensätze und Fensterverwaltung. Demgegenüber stehen allerdings immer noch vorhandene Betriebssystem-Fehler, die sich in gelegentlichen Ungereimtheiten äußern. Datenverluste hatte ich dadurch jedoch bisher noch nicht. Das äußere Erscheinungbild des GEM-Desktop ist recht ansprechend und gefällt mir eindeutig besser als die karge Kommando-Oberfläche des NewDOS. Die Funktionsvielfallt ist allerdings recht schmal, gemessen am Befehlsumfang von NewDOS80. Auch könnte das manchmal leidige 'runter- und wieder 'raufhangeln in der Directory-Hierachie etwas schneller gehen. Ebenso ist die Dastellung der Dateien als Icons bzw. als Text nicht ausreichend, da eine selective Anzeige nicht möglich ist. Dafür lassen sich die Dateien aber nach vier verschiedenen Kriterien sortiert anzeigen. Die unabdingbare Maus hat sich bei mir inzwischen einen festen Platz erkämpft. Dieses Tierchen ist wirklich eine feine Sache. Dank ihrer Anwesenheit entfällt bei der Textverarbeitung das lästige arbeiten mit den Cursortasten (um von A nach B zu gelangen: Die Maus auf den Fehler, einmal anklicken und schon ist der Cursor zur Stelle).



Zum Thema Arbeits-Tempo kann ich mich kurz fassen, das Spektrum reicht bei der Textverarbeitung von einem 'uuaaaah' bis zu einem entzückten 'ooohh'. Insgesamt hebt sich die Geschwindigkeit aber deutlich von der meines Genie ab: Eine Grafikfläche mit einem berechneten Muster füllen (384*192 Punkte groß), ist auf dem ATARI über 10 mal schneller als auf meinem Genie (mit 3,54Mhz Taktfrequenz). Bei solchen Zahlen kommt schon Freude auf. Mit dem Speicherplatz ist es - wen wunderts - nicht viel anders. 1 Mbyte sind für Z80-Verhältnisse 'einfach umwerfend'. Es soll aber nicht verschwiegen werden, daß man die auch bitter nötig hat. Ein Text mit vielen Zeichnungen (ca. 27) und einer länge von ca. 120KByte verschlingt glatt 400KByte an RAM ! Rechnet man das Übrige noch dazu (Bedarf des TOS, die Textverarbeitung und einige Accessories etc.), dann bleiben nur magere ~300KByte RAM übrig. Das reicht gerade noch für eine kleine RAM-Disk und dann ist die Bude voll. Komfort hat so seine Dimensionen. Die Software. Ein nicht unwichtiger Punkt. Meine Sammlung besteht zu gut 80% aus PD-Software, deren Qualität zum Teil äußerst überzeugend ist. So finden sich neben makrofähigem Text-Editor, Hochsprachen und Grafikprogramme, deren Leistungen sich sehen lassen können. Ich bin auf diesem Wege preiswert zu einer guten Softwarebasis gekommen. Lediglich mit den Viren muß man aufpassen, hier ist der ATARI eine ausgesprochen heißgeliebte Spielwiese, wie ich am eigenen Leibe erfahren mußte (ich durfte meine gesamte 80 Disketten umfassende Sammlung von diesen lieben Tierchen befreien). Trotz aller Fallen und Tücken bin ich mit dem Rechner zufrieden. Das Nachsehen wird auf Dauer wohl mein Genie haben, - leider - den die Kiste hat sich all die Jahre - immerhin 8 an der Zahl - wacker geschlagen.

Tschüß Kurt

Mein Zweitrechner - der Atari ST

Vor ungefähr einem Jahr hatte ich diese ständige Ungewißheit endlich satt: läuft mein Genie IIs morgen oder läuft es nicht? In den Ausfallzeiten ruhte meine gesamte schriftliche Arbeit, was auf die Dauer nicht mehr zu ertragen war. Also mußte ein Zweitrechner her, damit wenigstens immer einer betriebsbereit ist.

Warum ein Atari ST?

Das Einsatzgebiet des Rechners ist für mich die Erledigung allen Schreibkrams. Alle Briefe, Referate, Mitschriften, Artikel usw. werden getippt, weil sie sonst niemand (auch ich nicht) lesen könnte. Nebenbei habe ich noch das Anliegen, etwas zu lernen, sprich: etwas zu programmieren.

Das Budget war ziemlich klar: mehr als 2000,- sollte der Zweitrechner nicht kosten. Damit fiel mein Spitzenkandidat, der Macintosh, gleich aus der Wertung. Der Amiga war die nächstbeste Lösung, aber wegen seines Farbmonitors kostete er über 2000 DM. Außerdem sah ich nicht, wozu ich zum Briefeschreiben Farbe brauchte, ganz abgesehen davon, daß ich niemand in meinem Umfeld auftreiben konnte, der den Rechner hatte. Aber ohne einen Erfahrungsaustausch schien mir die Sache ziemlich witzlos.

Nachdem ein Prof-180X als reiner CP/M-Rechner wegen zu hoher Kosten deklassiert wurde, blieben nur zwei Kontrahenten übrig: der Atari ST und die gesamte Klasse der IBM-Kompatiblen. Die IBMs kannte ich mehr oder weniger aus der Uni und diversen Zeitschriften. So war mir klar, daß die Technik nur wenig über dem Niveau eines CP/M-Rechners lag. Die 640 KB Speicher waren zwar eine Verzehnfachung, aber die Programme blähten sich immer mehr auf, weil sie auf verschiedene Grafikkarten und andere Inkompatibilitäten der angeblich Kompatiblen Rücksicht nehmen mußten. Somit war der Speicher klar zu klein gewählt. Da selbst das Betriebssystem MS-DOS nahezu vollständig vom CP/M abgekupfert wurde, war eine Steigerung kaum zu erwarten. Ein AT-Kompatibler wäre interessant gewesen, aber die Dinger kosten sogar heute noch mehr als ich bezahlen könnte. Ein PC war wegen seiner 360-KB-Laufwerke zu klein (die meisten Programme belegen mehrere Disketten), also blieb nur ein XT mit Festplatte. Bei meiner Preis-Vorgabe hätte ich wohl ein Fernost-Produkt mit 640 KB Speicher, 360-KB-Laufwerk, 20-MB-Festplatte, HerculesGrafik-Karte und (normalen) Monochrom-Monitor bekommen können. Dem stand ein Atari 1040 STF mit 1 MB Speicher (auf 4 MB ausbaubar), 720-KB-Laufwerk, 640x400-Grafik und einem absolut flimmerfreien Monochrom-Monitor gegenüber. Die Festplatte fehlte hier, dafür gaben das 192 KB große ROM mit dem kompletten Betriebssystem und das starre Konzept (nur 3 Grafik-Auflösungen etc.) zu der Hoffnung Anlaß, daß die Programme eher kurz sein würden. Somit wurde die Festplatte eher zum Luxus als zur Notwendigkeit.

Der Rest war eine reine Glaubensfrage. Mein Bekanntenkreis spaltete sich in zwei Hälften: Die eine wollte mit dem Rechner Geld verdienen und hielt sich deshalb an das, was in der Wirtschaft Standard war, also IBM. Die andere Hälfte suchte einen Rechner, von und mit dem man etwas lernen konnte und dessen Technik und Software nicht allzu rückständig war; somit herrschte hier der Atari vor (seltsamerweise nicht der Amiga). Mit dem echten 16-Bit-Prozessor Motorola-68000 und der grafischen Benutzeroberfläche GEM samt Maus bot der Atari was Neues gegenüber dem IBM, was mich reizte und letztlich den Ausschlag gab.

Welche Erfahrungen habe ich gemacht?

Vom Auspacken bis zum Formatieren und Kopieren der ersten Diskette verging ca. eine halbe Stunde. Ausschlaggebend war dabei die im wahrsten Sinne des Wortes "kinderleichte" Bedienung per Maus, schon im normalen Betriebssystem. Die Tastatur ist nur sekundär und in den ersten Tagen brauchte ich sie überhaupt nicht. Ärgerlich war von Anfang an, daß (beim 1040 STF) die Tastatur gleich noch den Rechner und seitlich das Laufwerk enthält (an das man so schlecht herankommt), womit die ganze Sache unbeweglich wird. Positiv dagegen fiel mir sofort der Monitor auf: das Teil ist wirklich absolut flimmerfrei und ich möchte mit keinem anderen mehr länger arbeiten. Das schwarz-weiße Bild gibt mir oft den Eindruck, daß ich ein Blatt Papier vor der Nase hätte.

Das Betriebssystem läßt sich problemlos bedienen, was für die "echten" Programmierer ein Problem darstellt. Aber glücklicherweise haben die Designer des Systems einige Schwächen eingebaut, die bis heute nicht ausgemerzt sind. So wird das Copy-File-By-File mit einem Laufwerk zur Tortur, weil für jeden File die Disks zweimal gewechselt werden müssen (obwohl das RAM ausreicht, um alle Files auf einmal zu kopieren). Da hilft dann nur eine Ramdisk, ein zweites Laufwerk oder eine Festplatte.

Zu den (in meinen Augen) Systemfehlern gehören auch das Betriebssystem im ROM, das bekanntermaßen extrem fehlerhaft ist und in einem statischen RAM weitaus besser aufgehoben wäre, sowie die Begrenzung des Hauptspeichers auf 4 MB (neben der schlecht möglichen Aufrüstbarkeit). Da der Prozessor 16 MB adressieren kann, ist es ziemlich dämlich, das nicht auszunutzen. Aber wenigstens ist es mehr, als die IBMs bieten. Die haben allerdings (meistens) eine ordentliche Tastatur, was beim Atari nicht der Fall ist. Von der des C64 trennt sie nur wenig. Leider läßt sich nicht mal problemlos eine andere anflanschen. Dafür hat der User alle Tasten, die er sich wünschen kann, sogar eine "HELP"-Taste, die aber kaum ein Programm benutzt.

Praktisch sind die 3.5"-Laufwerke, denn die Disketten kann ich endlich in der Jackentasche transportieren, was bei den 5.25"-Disks zu leichten Problemen führte (8" sind da besser, die können schon als Kopfbedeckung dienen). Außerdem läßt sich über 720 KB nicht meckern, wenn auch mehr möglich wäre (s. Amiga oder neuerdings auch IBM).

Die Dokumentation ist sowohl für den absoluten Anfänger als auch für den versierten Programmierer nicht ausreichend. Anfänger müssen sich für teures Geld irgendwelche zweifelhaften Lehrbücher von Data Becker o.ä. kaufen, für Programmierer sieht es trauriger aus. Denn alle verfügbaren Veröffentlichungen kranken an einem Problem: Atari hat das System schlecht dokumentiert. Das Betriebssystem ist fast das Original-GEM, aber eben nur fast. Die Änderungen liegen im Dunkeln. Zusammen mit den im BIOS etc. eingebauten Fehlern ist das Chaos komplett. Ich habe bis jetzt noch kein Programm gefunden, das fehlerfrei läuft, d.h. auch mit installierten Ramdisks, verschiedenen Accessories usw. zusammenarbeitet. Bezeichnend ist, daß noch nicht einmal die "Type File"-Funktion des GEM richtig arbeitet.

Die Software

Die Programme auf dem Atari sind fast alle ohne Anleitung bedienbar, denn die Maus ist allgegenwärtig. Leider meint jedes Programm, die Maus müßte anders eingesetzt werden, was schon zur Verwirrung beiträgt.

Leider verzichten die Programmierer darauf, die verfügbaren Tastatur-Kommandos irgendwo leicht erreichbar zu dokumentieren, obwohl sie meistens vorhanden sind. So wird man oft zu den zeitraubenden Pull-Down-Menüs gezwungen, obwohl ein Ctrl-Code kürzer und verfügbar wäre.

Außerdem scheinen die Programme auch nie den Test-Status zu verlassen. Von meinem favorisierten Textprogramm, STEVE, gibt es inzwischen Version 3.08 (nach 2.3, 2.5, 3.0), und trotzdem mußte ich schon beim ersten Einschalten einen Fehler feststellen. Ebenso sieht es in fast allen Bereichen aus:

Unfertige Programme werden ausgeliefert, wobei oft noch Grundfunktionen fehlen. Die Verbesserungen bestehen dann darin, irgendwelche exotischen Drucker oder Scanner zu unterstützen, während die Grundfunktionen weiter fehlen. Dafür habe ich aber eine Leistung zur Verfügung, von denen andere Rechner-Besitzer nur träumen können, während ich selber von der Peripherie träume, mit der diese Leistung erreichbar wäre.

Ebenso steigt der Umfang der Programme immer mehr an, obwohl die Ausmaße derer der IBM-Familie noch lange nicht erreicht sind. Der Grund ist mir noch unklar. Mitverursacher sind sicherlich die Einbindung in GEM, die Platz schluckt, und die Vernachlässigung der Assembler-Programmierung zugunsten von C oder Basic, was bei den miesen Assemblern, die es bis jetzt gibt, kein Wunder ist.

Interessant sind die verschiedenen Betriebssysteme und Emulatoren, die es für den Atari noch gibt: RTOS und OS/9 für Multitasking (der Benutzer als störender Rand-Prozeß), CP/M-80 für die alten Freaks, PC/MS-DOS für die Unbelehrbaren, Macintosh für die Neidischen, C64 für die Spieler.

Neben der Textverarbeitung (ca. 90%) benutze ich den Rechner höchstens mal, um eine Grafik zu zeichnen, aber da steht meine nicht vorhandene künstlerische Begabung dagegen. Oder ich ziehe aus der umfangreichen Spiele-Kiste ein Grafik-Adventure und und erinnere mich wehmütig an die Zeiten, als die Klötzchengrafik von "Forbidden City" mit dem unverständlichen Sound das allerhöchste der Gefühle war. Programmiert habe ich dagegen auf dem Atari bis jetzt nur, wenn es sich nicht verhindern ließ, und auch nur mit Pascal. Zwar bin ich inzwischen glücklicher Besitzer eines Modula, aber das Turbo-Modula von CP/M ist immer noch besser.

Warum verschrotte ich mein Genie IIs nicht?

Die Frage ist einfach zu beantworten: Ich benutze zwar den Atari (als reiner Anwender), aber programmieren tue ich nur am Genie IIs. Außerdem läßt es willig meine bescheidenen Umbauversuche über sich ergehen, die beim Atari unmöglich sind (gesockelte ICs? Veraltet!). Zwar sind die Anwenderprogramme auf dem Atari leicht zu bedienen und sehr leistungsfähig, aber zum Programmieren bietet die Kiste mir nicht genug. Auf meinem IIs kenne ich wenigstens noch jedes Byte persönlich und weiß genau, warum die Kiste abstürzt; beim Atari hängt das allein vom Wetter und sonstigen mystischen Dingen ab.

Fazit

Wer gezwungen ist, mit dem Rechner Geld zu verdienen, muß sich wohl einen IBM-Kompatiblen kaufen. Diese Leute tun mir leid, denn sie bekommen wenig für ihr Geld und müssen bei den Leistungsdaten ihres Rechners noch vor CP/M-Rechnern Angst haben.

Wer allerdings mal was ganz anderes möchte, wie eine echt bedienungsfreundliche Oberfläche (ist auch nicht jedermanns Sache), einen Super-Monitor, viele ausgeflippte Programme, ein zukunftsweisendes Betriebssystem und einen echten 16-Bitter, der sollte sich mal den Atari ansehen. Allerdings ist das Ding bestenfalls ein Prototyp für den Hausgebrauch und mehr in die Kategorie "Spiel-Kram" einzuordnen. Aber hier kann jeder sehen, wo der Weg in Zukunft langgehen wird, denn nicht umsonst zieht IBM jetzt mit "Windows" und ähnlichem Zeug hinterher. Aber die Leistung in Reinkultur gibt es wohl nur beim Amiga (Multitasking, Farbgrafik) oder dem Mac II (standardisierte Programme, durchdachtes Betriebssystem).

Gerald Schröder

ATARI

Auch ich möchte mich, da ich den 1040 ST nun auch schon eineinhalb Jahre in Betrieb habe, kurz zum Atari-Thema melden.

Grundsätzlich kann ich mich den gedruckten Worten meiner Vorschreiber anschließen. Das Kopieren mit einem Laufwerk ist echt Quälerei! Für ein einzelnes File benötigt der Atari ca 4-5 Diskettenwechsel, die er jeweils mit einer Meldebox anzeigt. Arbeitet man mit einem Zweitlaufwerk -wie ich- oder RAM-Disk, fällt das nicht mehr ins Gewicht. Hier sind die Bytes ruckzuck umgeschaufelt.

Als einen weiteren großen Mangel finde ich die Geschichte mit den drei verschiedenen Auflösungen. Man sollte schon einen Farb-Multisync (oder je einen Farb-/SW-Monitor) auf dem Tisch stehen haben um alle Programme fahren zu können. Hier hätte man noch mehr tun können. Mindestens mit dem mitgelieferten SW-Monitor sollten alle Auflösungen (bei Farbe dann in Grautönen) darstellbar sein. Platz genug für eine entsprechend gestaltete Platine wäre schon, es ist da ja eh nicht viel drin. Vielleicht ist auch sonst nichts mehr verdient. Ich jedenfalls hatte mich damals für einen Farbmonitor entschieden. So kann ich zumindest die "mittlere" (640X200) und die "geringe" (320X200) Auflösung betreiben.

Nachdem ich mich nun gleich über die zwei -von mir aus gesehen- negativen Seiten hergefallen bin, möchte ich natürlich auch die Pluspunkte erwähnen, die mich zu der Kaufentscheidung Atari 1040 ST brachten.

Ich wollte auf jeden Fall einen modernen Rechner, mit dem man farbige Grafiken und Animationen erstellen und bearbeiten und zum Relaxen ein schickes Game usen kann.

Dies fehlte mir bei dem guten TRS-80 ganz und gar. Der Tandy ist aber auch weiterhin für mich im Einsatz. Da ich auch noch nichts umgeschrieben oder angepasst habe, verwende auch noch alle Programme sei es nun für Text, Daten und den anderen Krimskrams der sonst noch läuft.

So standen für mich damals zwei Rechner zur Auswahl. Atari oder Amiga. Der Amiga ist zwar in Grafik und Sound noch besser wie der Atari, aber auch teurer. Weiterhin war der Amiga Sofwaremäßig schlecht unterstützt -und wie die Zeit zeigt ist er auch heute noch hauptsächlich ein guter Spieler-.

Auf Seite 62 gehts weiter ...

VERKAUFE

TANDY Model 4P mit 2 Laufwerken 80 Spuren, DS, DD und jeder Menge Software (Preis VB 800 DM)

Nadeldrucker TANDY, 9 Nadeln, Papierbreite DIN A3 (Preis VB 200 DM)

Wer Interesse hat, kann sich schriftlich oder telefonisch melden:

Günther Wagner, Heubergstraße 8, 8201 Neubeuern, Tel. 08035/3927 (ab 19 Uhr oder am Wochenende)

Halt, hier hat sich gerade noch was gefunden. Unser ehemaliges Mitglied Harald Trapp hat noch etwas zu verkaufen. Bitte setze folgende Anzeige ins Info: DIN A3-Plotter von DENSON
serielle und parallele Schnittstelle
4 Farben, Einzelblatteinzug, 1/2 Jahr alt
wegen Wechsel zu A1-Plotter zu verkaufen
Preis 1800,- DM (50% des Neupreises!)

Verkaufe aufgrund Räumungsaktion:

- TRS 80, Model 4 P, 2 x 64 kB, 2 Laufwerke
- ALCOR-Pascal (Original) für Model 1,3 u. 4 (M3-Modus)
- 80-Micro Jahrgänge 1985, 1986 u. 1987

Preis: VS

Klaus Hermann, Tel.: 07127/71945

Er prommt...

Da ich es endlich geschafft habe, meinen Prommer zur Zusammenarbeit zu überreden, stehe ich ab sofort auch für 'brennende Fragen' zur Verfügung. Er schafft alles vom 2716 bis 27256, wenn ich einen Adapter gebaut habe auch 27512 und größer.

Alexander Schmid

Verkaufe für fast nichts

- Model I, Expansion, grüner Monitor, Voice-Synthi, 3 Laufwerke, Expson-MX-80
- Genie (defekt), Expansion, 2 Laufwerke
- 3 Kartons Dokumentation, 80-US, Software & Anleitungen
- Rechnertisch

außerdem:

- dt. Model III, 2 Laufwerke
- Model I, Expansion, s/w-Monitor, 2 Laufwerke

bitte melden bei: Horst Schuller
An der Pinnau 17
2359 Henstedt-Ulzburg
Tel. 04193/4158

Projekt: Börsenkurse etc. per VideoText in den Rechner

Ihr könnt das Videotextangebot (Wetter, Verkehr, Nachrichten, Sport, Börsenkurse usw.) in den Rechner übernehmen. Nötig dazu ist nur ein Adapter einer Schweizer Firma (RTS 1598 von Nordex AG, Postfach 673, CH-3000 Bern 14, Tel. CH/031/250405), eine serielle Schnittstelle und (noch) ein IBM-Kompatibler.

Dieter Kasper (ehemaliges Clubmitglied) würde diese Sache nun auch gerne für unsere Rechner (unter CP/M oder Newdos) nutzbar machen. Wer Lust hat, bei dem Projekt mitzumachen, wende sich an

Dieter Kasper
Schillerstr. 30
8952 Marktoberdorf
Tel. 08342/1630

Verkaufe wegen Systemwechsel:
TRS 80 Mod III mit
RS 232 (org. Tandy) und
2 Laufwerken 40Track SS DS
+ umfangreicher Literatur
+ div Software
VB 550,-

Christian Schrenc
Duisburger Str. 99
4000 Düsseldorf 30
0211/443374

Für den neuesten Krimi-Happening-Sitting in & out:

Erfahrungs-Austausch

über

dBASE II

gesucht!

Vorzugsweise mit TRS-80-User m. RAMDISK!
Aber auch mit anderem Freak. - Bitte melden bei:
K.-J. Mühlenbein, Am Mönchgarten 28, 6940 WEINHEIM
→ Es wird spannend! ←

H. Pfisterer
Bietigheim-Bissingen

Hermann Pfisterer · Werkzeugfabrik · D-7120 Bietigheim-Bissingen

Zentrierbohrer
Stufenbohrer
Sonderwerkzeuge

Herrn
Gerald Schröder
Am Schützenplatz 14

2105 Seevetal 1

03.10.88 wpf

Teletex - Modem für CP/M-Computer

Sehr geehrter Herr Schröder,

dem Chip-Magazin habe ich entnommen, daß Sie und Ihr Computer-Club sich mit CP/M-Rechnern beschäftigen.

Wegen Umstellung auf ein neues Gerät habe ich abzugeben, und biete Ihnen oder Ihren Club-Mitgliedern an:

1 Stück Teletex-Modem von Triumpf Adler
für serielle Schnittstelle,
für TEX-ASS Textverarbeitungsprogramm
gebraucht, voll funktionsfähig, postzugelassen
bisher mit PC Alphatronic P3 von TA betrieben.

Verkaufspreis: Verhandlungsache

Bitte setzen Sie sich bei Interesse mit mir in Verbindung.

Mit freundlichen Grüßen
Hermann Pfisterer
Werkzeugfabrik

Wolfram Pfisterer

Anschrift: Schleifmühle 7
D-7120 Bietigheim-Bissingen
Telefon: (0 71 42) 3 00 50
Teletex: 714215=zebo
Telefax: (0 71 42) 3 28 80

Inhaber: Dipl. Wirtsch. Ing. (FH)
Wolfram Pfisterer

Registergericht Heilbronn
HRA 777-Bes.

Bankverbindungen: Volksbank Bietigheim (BLZ 604 911 60)
Konto 430 613 008
Dresdner Bank (BLZ 604 800 08)
Konto 520 679 000
Postgiro: Stuttgart (BLZ 600 100 70) 52 074-706

... Fortsetzung von Seite 58

Für den Atari hingegen gibt es super (und auch funktionierende !!) Software zur Bild-Gestaltung und -Animation. Als Beispiel möchte ich hier nur die Programme der "Cyper-Serie" von Antik oder "Imagic" aus Heidelberg anführen.
Besonders für mich als Video-Filmer wird die Sache durch einen Digitizer (welcher auf den ROM-Port gesteckt wird) erst richtig interessant:
Filmen, die Bilder digitalisieren, rein damit in den Atari, verändern...ergänzen...animieren, raus als Videosignal und fertig ist ein Vorspann für eigene oder aufgenommene Viedeofilme.
Dies war für mich die Entscheidung!
Wer die Preise für einen Schriftgenerator und weitere wichtige Zutaten der Videofilmerei kennt, wird wissen, daß mit ST und entsprechender Software mindestens die "halbe Miete" gespart ist -wie man so schön sagt-.

Abschließend muß ich sagen, daß ich mit dem Atari und Farbmonitor sehr zufrieden bin. Am Anfang war die Software (bis auf Spiele) sehr auf SW ausgerichtet, aber jetzt laufen die meisten Sachen auch in der mittleren Auflösung. Es ist also auch kein Nachteil mehr "nur Farbe" zu haben.
Wenn man dann seine eigenen Animationen laufen hat und dann zur Entspannung noch etwas "gamet" möchte man fast gar nicht wieder zurück zum Tandy, der sich bernsteinfarben auf dunkelbraunem Grund mit einem "NEWDOS 80 READY" meldet.

Auf ein fröhliches, farbiges Pixeln
Jens Neueder

Matthias Böhne
EDV-Verbrauchsartikel
Försterweg 4
3354 Dassel-Sievershausen
Tel. 05564/2150

Oktober 1988

Sehr geehrte Damen und Herren,

der Bedarf an EDV-Verbrauchsartikeln steigt ständig an. Speziell für Computer-Club's haben wir daher eine Preisliste gängiger Farbbänder unseres Vertriebsprogramms zusammengestellt. Unser Lieferumfang umfaßt ca. 1000 Schreib- und Korrekturbänder sowie etwa 100 verschiedene EDV-Etikettenformate und Abpackungen. Wir liefern sowohl Endlosetiketten für Tintenstrahlrucker und Matrixdrucker, wie auch Einzelblattware für Laserdrucker.

Hier zwei Preisbeispiele:

ENDLOSETIKETTEN für Matrixdrucker

Format	70,0 x 70,0 mm	z.B. für 3,5 Zoll-Disk.
Farbe	weiß	
Kartoninhalt	2.000 Etiketten	

Preis je Karton bei Abnahme von:

1 Karton	DM 54,72	inkl. Mwst.
ab 3 Karton	DM 51,98	inkl. Mwst.
ab 6 Karton	DM 49,25	inkl. Mwst.

Format	89 x 36 mm	z.B. als Adressetikett
Farbe	weiß	
Kartoninhalt	4.000 Etiketten	

Preis je Karton bei Abnahme von:

1 Karton	DM 43,32	inkl. Mwst.
ab 3 Karton	DM 41,15	inkl. Mwst.
ab 6 Karton	DM 38,99	inkl. Mwst.

Gern senden wir Ihnen auf Wunsch weitere Informationen zu.

Mit freundlichen Grüßen

Die Lieferung erfolgt ab DM 200,- Warenwert frei Haus, ansonsten DM 5,70 Versandkostenpauschale. Zahlbar innerhalb 10 Tagen - 2% Skonto oder 30 [illegible] netto Kasse. Alle gelieferten Waren bleiben bis zur vollständigen [illegible] unser Eigentum.

An alle Computerclub's und Vereine

im Bundesgebiet

Auszug aus unserem Angebot: Stand 10/88

Qualitätsfarbbänder

- made in germany
- Einzelverpackung
- Lieferzeit i.d.R. 7 Tage
- Zahlbar nach Rechnungserhalt 10 Tg. -2% ... 30 Tg. netto

Brother
M 1409 ======>DM 10,75 /// M 1509/1709 =>DM 12,70/// M 1724 L ==>DM 13,90

Citizen
LSP 120 D ===>DM 10,10 /// MSP 120 D ===>DM 10,10/// MSP 10 ====>DM 8,50
MSP 10 E ====>DM 8,40 /// ///

Commodore
MPS 801 =====>DM 8,50 /// MPS 802 =====>DM 9,20/// MPS 803 ===>DM 8,20
MPS 1000 ====>DM 7,40 /// MPS 1200 ====>DM 10,10/// MPS 2000 ==>DM 11,35
MPS 2010 ====>DM 13,15 /// MPP 1361 ====>DM 8,50/// VC 1515 ===>DM 12,20
VC 1525 =====>DM 12,20 /// ///

Epson
FX 80 =======>DM 8,50 /// FX 100 ======>DM 10,20/// GX 80 =====>DM 7,40
LQ 800/850 ==>DM 9,50 /// LQ 1000/1050 >DM 10,95/// LQ 1500 ===>DM 10,95
LQ 2500 =====>DM 10,40 /// EX 1000 =====>DM 9,85///

Fujitsu
DL 2300 =====>DM 27,20 /// DX 2100/2200 >DM 11,50/// DX 2300/2400 DM 11,65

IBM
4201 (6328829) DM 10,65 /// 4202-001(1040150) 14,15/// 4207 XL 24 DM 11,90
4208 XL 24 DM 14,85 /// ///

NEC
P 5 ========>DM 11,50 /// P 6 =========>DM 11,35/// P 7 =======>DM 13,15
P 2200 ======>DM 12,05 /// P5/9XL (Multist.) 13,00///

OKI
ML 182/192/194 DM 8,50 /// ML 292 schwarz DM 17,50/// ML 292 color DM 26,00
ML 293 schwarz DM 19,20 /// ML 293 color DM 29,40/// ML 390/391 >DM 13,00

Panasonic
KX-P 1080.... DM 11,15 /// ///

Schneider
DMP 2000/3000 DM 9,30 /// NLQ 401 =====>DM 8,20/// Joyce Synth. DM 10,95
DMP 4000 ====>DM 13,25 /// ///

Seikosha
SP 800/1000 A DM 10,95 /// GP 80 =======>DM 12,20/// GP 100 ====>DM 12,20

Star
NB/ND/NL 10 =>DM 10,40 /// LC 10 =======>DM 9,25/// NB 24-10 ==>DM 12,30
NB 24-15 ====>DM 13,55 /// ND/NR/NX 15 =>DM 13,70///

Lieferung ab Warenwert DM 200,- frei Haus, sonst DM 5,70 Versandkostenanteil. Unser Lieferprogramm umfaßt etwa 1000 verschiedene Schreib- und Korrekturbänder. [illegible]te fordern Sie weitere Unterlagen

Tandy- und Genie-Geräte auf dem Gebrauchtcomputermarkt

Zufällig fand ich in einer CHIP 8/88, die ihren Weg vom Vorzimmer eines Zahnarztes über Harald Mand zu mir gefunden hat (kaufen würde ich mir die CHIP zur Zeit nicht!), eine Zusammenstellung von Preisen für Gebrauchtcomputer. Mit Marktanalyse oder ähnlichem hat die Zusammenstellung sicher nichts zu tun. Vielmehr haben die CHIP-Leute die Anzeigenteile ihrer Zeitschrift durchsucht und jeweils den niedrigsten und den höchsten Preis für ein bestimmtes Gerät ermittelt. Also kein statistischer Schnitt und auch keine tatsächlich erreichten Verkaufspreise, sondern nur Forderungen. Trotzdem hier die Ergebnisse für ein paar Rechner:

Modell	Ausstattung/Zubehör	Gebrauchtpreislage im Markt			
		1985	1986	1987	1988
TRS 80 M.1	48k, LW, HRG, SW	550-1500	800-2250	650-1100	600-1600
TRS 80 M.3	48k, 2 LW, SW	1000-3300	1200-1800	490-1200	400- 900
Genie I	48k, LW, Mon., Dru.	1600	500-1700	350-1000	350- 700
Genie II	64k, 2 LW, Mon., SW	500-2600	1000-2000	960-1100	750-1000

Soweit die Tandy- und Genie-Maschinen. Nun ein paar vergleichbare Rechner.

CPC 464	64k, LW, SW	750- 800	500- 790	350-2000	370-1000
TA P2	48k, Drucker, SW	2800-3200	2000-2600	1200-2000	800-1700
TA PC	64k, 2 LW, Mon., SW	1200-2700	950-2800	700-1300	700-1500
Apple II+	64k, 2-3 LW, Monitor Drucker, SW	1500-3200	750-3300	450-2000	700-1350
C64	LW, Mon., Dru., SW	500-1400	600-1200	320-1300	-1200

(CPC 464 von Schneider, TA's von Alphatronic)

Natürlich sind die Preise bei allen Maschinen sehr von der Ausstattung, dem Zustand und der mitgelieferten Software und Dokumentation abhängig. Einen Reim auf die angegebenen Zahlen muß sich jeder Leser also selbst machen!

Diese Liste soll übrigens keineswegs eine Aufforderung sein, sich seines Rechners zu entledigen! Auch bei mir laufen zwei Rechner und haben durchaus zu tun. Zudem gibt es Dinge, die ein (man mit einem) TRS 80 oder Kompatiblen durchaus besser (machen) kann als so mancher AMIGA/ATARI ST oder PC-Clone. Und wenn es nur der Umstand ist, daß man an den "alten" Z80-Geräten auch einmal Hand (sprich "Lötkolben") anlegen kann, ohne gleich Angst haben zu müssen, daß der geliebte Computer nie mehr zum Leben erwacht.

In diesem Sinne viel Spaß beim Computern und Basteln,

Hartmut Obermann

Wenn einer eine Reise tut...

...kann er viel berichten. Wie Hartmut ja schon geschrieben hat, habe ich einen Akustikkoppler und bin öfters in den münchner Mailboxen unterwegs. Hin und wieder stößt man dabei auf ganz witzige Sachen, die es durchaus wert sind, auch woanders veröffentlicht zu werden. In loser Reihenfolge werden hier also ab und zu solche Fundsachen auftauchen, wenn sie halbwegs einen Bezug zur Computerei haben.

"humor": Software verhindert Maschinenfehler
--- Di, 18.10.88 / 09:18:16 von LORC

Wie aus den jüngsten Veröffentlichungen der Amerikanischen Gesellschaft für digitale Impulsunterdrückung (ASPIC) hervorgeht, ist es nach jahrelanger Forschungsarbeit gelungen, ein Software-Programm zur Ausschaltung von Maschinenfehlern zu entwickeln. Die Routine wird in Fachkreisen als Meilenstein in der Geschichte der Systemtechnik bezeichnet, da sie die immensen, durch Maschinenfehler bedingten Zeitverluste weitgehend ausschaltet.

Der Programmname "Orema" ist vom lateinischen oremus (lasset uns beten) abgeleitet. Die Routine ruft in bestimmten Zeitabständen Gebete ab, in denen himmlischer Beistand für die Bewahrung des Arbeitsspeicherinhalts, die Festigkeit von Magnetisierungsschichten und solcher Lötstellen erfleht wird, die besonderen mechanischen Anfechtungen ausgesetzt sind.

Die liturgische Struktur der Orema-Routine gestattet den wahlfreien Aufruf wirkungsvoller, extern gespeicherter Stoßgebete in Lateinisch, Hebräisch oder Cobol. Durch die Orema-Unterroutine "Misere" werden dreimal täglich sogenannte zyklische Messen zelebriert (Permanentanzeige auf dem Bildschirm). Die ministrierenden Benutzer beschränken sich während dieser Zeit lediglich auf das Einlegen der Diskette und das Eingeben sakraler Nachrichten über die Tastatur (zum Beispiel "... und mit Deinem Geiste" oder "Kyrie eleison").

Litaneien in Hebräisch oder Cobol können direkt über einen Freien Kommunikationskanal (FKK) in die Zentraleinheit eingegeben werden. Lateinische Psalmen (Kennsatz: "te.deum") sind jedoch peripher über Diskette einzuspeichern. Hebräische oder Cobol-Litaneien haben sich als besonders wirkungsvoll erwiesen. Obwohl von den Herstellerfirmen bereits fertige Gebetsdisketten für alle denkbaren Maschinenfehler geliefert werden, kann der versierte Programmierer von sich aus hinter jedem Amen-Block Prozeduren mit Eigenerflehung (Kennsatz: "jaul") eingeben. Spezielle Nothelfergebete stehen für Multitasking und wichtige Tests zur Verfügung. Nach letzten Testläufen mit Orema wird die durchschnittliche Maschinenausfallzeit um 98,2 Prozent reduziert. Wie aus gutunterrichteten Kreisen verlautet, wird Orema in absehbarer Zeit auch für die Ausmerzung von Hardwarefehlern eingesetzt werden können.

Vom Verteidigungsministerium wurde inzwischen ein kybernetisch-theologischer Arbeitskreis eingesetzt, der sich mit der Entwicklung eines Software-Systems zur Abschaffung des menschlichen Versagens befaßt. Die Fertigstellung dieser Programme (Arbeitstitel "Meaculpa") wird allerdings noch geraume Zeit in Anspruch nehmen.

(Hans Michael Engelke, DOS International 11/88, Seite 14)

Mancher geplagte Model-I-Besitzer wäre wohl froh um so ein Programm...

Alexander Schmid

dBase II, III und III Plus im Überblick

Auch diesen Monat befaßt sich das DOSsier mit dBase. Nachdem in der letzten Ausgabe der gesamte Befehlssatz von dBase abgedruckt war, folgen diesmal drei Tabellen, die sämtliche Funktionen von dBase sowie alle »set« und »set to«-Befehle mit Voreinstellung und Bedeutung aufführen.

Dieses DOSsier ist die Fortsetzung zum DOSsier aus der letzten Ausgabe von DOS International. In der ersten Tabelle sind sämtliche Funktionen des Datenbanksystems dBase übersichtlich mit Bedeutung aufgelistet. Zusätzlich ist angegeben, in welchen dBase-Versionen die jeweilige Funktion existiert. Tabelle 1 enthält in der letzten Spalte zu jeder Funktion auch die Angabe, zu welcher Gruppe diese Funktion gehört.

Es ist nicht einfach, die dBase-Funktionen in Gruppen zu unterteilen, weil sich die Bereiche oft überschneiden. Die Abkürzungen für die in Tabelle 1 verwendeten Funktionsgruppen sind:

D - Datums-/Zeitfunktion
E - Environment- beziehungsweise Umgebungsfunktion
F - Dateifunktion
N - numerische Funktion
S - Zeichenkettenfunktion

Auch die in der Tabelle verwendeten Bezeichnungen »A«, »D« und »N« haben eine bestimmte Bedeutung:

- »A« steht für einen Ausdruck oder eine Zeichenkette,
- »D« bezeichnet einen Datumsausdruck oder eine Datumsvariable, und
- »N« repräsentiert einen numerischen Ausdruck.

Bei Bedarf sind an diese Kennungen als Indizes Kleinbuchstaben angehängt.

Die Tabelle der »set«-Schalter (Tabelle 2) beinhaltet Angaben zu den Voreinstellungen der jeweiligen Schalter, das heißt, die Schalterpositionen, in denen sich die Schalter zu Beginn einer dBase-Sitzung befinden.

Tabelle 3 faßt die Bedeutung der »set to«-Befehle zusammen. Achten Sie darauf, daß Sie zur Vereinfachung Ihrer Eingaben sowohl im Direkt- als auch im Programmodus auf die Befehlsabkürzungen zurückgreifen können, die sich jeweils aus den ersten vier Buchstaben eines Befehlswortes zusammensetzen. Beispielsweise hat die Abkürzung

`disp`

die gleiche Wirkung wie

`display`

(Ulrich Spranger/ma)

Funktion	Bedeutung	dBase II	dBase III	dBase III Plus	Funktionsgruppe
*	prüft, ob Datensatz zum Löschen markiert ist	x	–	–	F
#	aktuelle Datensatznummer	x	–	–	F
!(A)	wandelt »A« in Großbuchstaben um	x	–	–	S
$(A,Na,Nb)	Zeichenkette aus »A«, beginnend bei Position »Na« und »Nb« Zeichen lang	x	–	–	S
abs(N)	absoluter Wert von »N«	–	–	x	N
asc(A)	gibt ASCII-Code des ersten Zeichens von »A« aus	–	x	x	S
bof()	prüft, ob Dateianfang vorliegt	–	x	x	F
cdow(D)	gibt Wochentag von »D« als Wort aus	–	x	x	D
chr(N)	wandelt Code von »N« in ASCII-Zeichen um	x	x	x	S
cmonth(D)	gibt Monat von »D« als Wort aus	–	x	x	D
col()	aktuelle Bildschirmspalte	–	x	x	E
ctod(A)	wandelt Zeichenvariable in Datumsvariable um	–	x	x	D
date()	Systemdatum	x	x	x	D
day(D)	gibt den Tag von »D« als Zahl aus	–	x	x	D
dbf()	Name der geöffneten Datenbank	–	–	x	F
deleted()	prüft, ob der aktuelle Datensatz zum Löschen markiert ist	–	x	x	F
dow(D)	gibt den Tag von »D« als Zahl bezüglich der Woche aus	–	x	x	D
dtoc(D)	wandelt Datumsvariable in Zeichenvariable um	–	x	x	D
diskspace()	freier Speicherplatz auf aktuellem Laufwerk	–	–	x	E
eof	prüft, ob Dateiende vorliegt	x	–	–	F
eof()	prüft, ob Dateiende vorliegt	–	x	x	F
error()	Nummer des letzten Fehlers	–	–	x	E
exp(N)	natürlicher Exponent von »N«	–	x	x	N
field(N)	Name des Feldes, das mit »N« korrespondiert	–	–	x	F
file("Dateiname")	prüft, ob Dateiname existiert	x	x	x	E
fklabel(N)	Funktionstastenname, der mit »N« korrespondiert	–	–	x	E
fkmax()	maximale Anzahl der Funktionstasten, die programmierbar sind	–	–	x	E
found()	prüft, ob der letzte Suchbefehl Erfolg hatte	–	–	x	F
getenv(A)	Systemumgebungsvariable	–	–	x	F
inkey()	letzer Tastendruck als ASCII-Zeichen	–	–	x	E
int(N)	gibt »N« ohne Nachkommastellen aus (Integerfunktion)	x	x	x	N
isalpha(A)	prüft, ob erstes Zeichen von »A« ein Buchstabe ist	–	–	x	S
iscolor()	prüft, ob ein Farbbildschirm angeschlossen ist	–	–	x	E
islower(A)	prüft, ob erstes Zeichen von »A« ein Kleinbuchstabe ist	–	–	x	S
isupper(A)	prüft, ob erstes Zeichen von »A« ein Großbuchstabe ist	–	–	x	S
left(A,N)	bildet aus den ersten »N« Zeichen von »A« eine neue Zeichenkette	–	–	x	S
len(A)	gibt Anzahl der Zeichen von »A« aus	x	x	x	S
log(N)	natürlicher Logarithmus von »N«	–	x	x	N
lower(A)	wandelt »A« in Kleinbuchstaben um	–	x	x	S
ltrim(A)	formt »A« in eine linksbündige Zeichenkette um	–	–	x	S
lupdate()	gibt an, wann die Datenbank zuletzt geändert wurde	–	–	x	F
max(Na,Nb)	stellt den größeren Wert von »Na« und »Nb« fest	–	–	x	N
message()	letzte Fehlermeldung	–	–	x	E
min(Na,Nb)	stellt den kleineren Wert von »Na« und »Nb« fest	–	–	x	N
mod(Na,Nb)	ergibt den Rest der Division von »Na« durch »Nb«	–	–	x	N
month(D)	gibt Monat von »D« als Zahl aus	–	x	x	D
ndx(N)	Indexdatei-Name, der mit »N« korrespondiert	–	–	x	F
os()	gibt aus, welches Betriebesystem benutzt wird	–	–	x	E
pcol()	aktuelle Druckerspalte	–	x	x	E
prow()	aktuelle Druckerzeile	–	x	x	E
rank(A)	gibt ASCII-Code des ersten Zeichens von »A« aus	x	–	–	S
readkey()	Tastendruck, durch den der bildschirmorientierte Befehl verlassen wurde als numerische Ausgabe	–	–	x	E
reccount()	Gesamtzahl aller Datensätze der aktuellen Datenbank	–	–	x	F
recsize()	Datensatzgröße	–	–	x	F
recno()	aktuelle Datensatznummer	–	x	x	F
replicate(A,N)	wiederholt »A« »N«-mal	–	–	x	S
right(A,N)	bildet aus den letzten »N« Zeichen von »A« eine neue Zeichenkette	–	–	x	S
round(N,Na)	rundet »N« auf »Na« Nachkommastellen	–	x	x	N
row()	aktuelle Bildschirmzeile	–	x	x	E
rtrim(A)	kürzt »A« um alle rechten Leerstellen	–	–	x	S
space(N)	gibt »N« Leerzeichen aus	–	x	x	S
sqrt	Quadratwurzelfunktion	–	–	x	N
str(Na,Nb,Nc)	Zeichenkette aus »Na« mit einer Länge von »Nb« Zeichen und »Nc« Dezimalstellen	x	x	x	N
stuff(Aa,Na,Nb,Ab)	»Ab« wird über »Aa« geblendet, beginnend bei Position »Na« für »Nb« Zeichen	–	–	x	S
substr(A,Na,Nb)	Zeichenkette aus »A«, beginnend bei Position »Na« mit einer Länge von »Nb« Zeichen	–	x	x	S
test(A)	testet Ausdruck	x	–	-	
time()	Systemzeit	–	x	x	E
transform(A,Aa)	Umformung von Ausdruck »A« in das Format von »Aa«	–	–	x	N
trim(A)	kürzt »A« um alle rechten Leerstellen	x	x	x	S
type(A)	prüft Ausdruck	–	x	x	E
upper(A)	wandelt »A« in Großbuchstaben um	–	x	x	S
val(A)	ergibt numerischen Wert von »A«	x	x	x	N
version()	Versionsnummer von dBase	–	–	x	E
year(D)	gibt Jahr von »D« als Zahl aus	–	x	x	F

Tabelle 1. Die Funktionen der Datenbanksysteme dBase II, dBase III und dBase III Plus

Schalter	Voreinstellung	Bedeutung	dBase II	dBase III	dBase III Plus
set alternate	off	Protokolldatei ein/aus	x	x	x
set bell	on	Tonsignal ein/aus	x	x	x
set carry	off	Übernahme der Daten des letzten Datensatzes ein/aus	x	x	x
set century	off	Datumsanzeigen vierstellig ein/aus	x	–	-
set confirm	off	[Return] vor Sprung ins nächste Feld ein/aus	x	x	x
set console	on	Bildschirmausgabe ein/aus	x	x	x
set debug	off	Fehlermeldungen an Drucker leiten ein/aus	x	x	x
set deleted	off	zum Löschen markierte Datensätze nicht berücksichtigen ein/aus	x	x	x
set delimiter	off	schaltet Feldbegrenzungen ein/aus	–	x	x
set dohistory	off	Programmbefehle in Historydatei protokollieren ein/aus	–	–	x
set echo	off	aktuelle Programmzeile anzeigen ein/aus	x	x	x
set escape	on	Funktion der Escape-Taste ein/aus	x	x	x
set exact	off	exakte Zeichenkettenvergleiche ein/aus	x	x	x
set fields	on	Benutzung der Feldliste ein/aus	–	–	x
set fixed	off	regelt Dezimalstellenausgabe	–	x	x
set heading	on	Feldüberschriften ein/aus	–	x	x
set help	on	Hilfsfunktionen ein/aus	–	x	x
set intensity	on	inverse Menüdarstellung ein/aus	x	x	x
set menues	on	Hilfsmenüs ein/aus	–	x	x
set print	off	Druckausgabe ein/aus	x	x	x
set safety	on	Sicherheitsabfragen ein/aus	–	x	x
set scoreboard	on	Statuszeile ein/aus	–	x	x
set status	on	Statuszeile unten ein/aus	–	–	x
set step	off	Einzelschrittmodus ein/aus	x	x	x
set talk	on	Dialog-Modus ein/aus	x	x	x
set title	on	Anzeige des Dateinamens ein/aus	–	–	x
set unique	off	identische Indexschlüssel ein/aus	–	x	x

Tabelle 2. Die »set«-Befehle dienen zum Umstellen der dBase-Schalter, ...

Befehl	Bedeutung	dBase II	dBase III	dBase III Plus
set alternate to	Protokolldatei definieren	x	x	x
set catalog to	Katalogdatei eröffnen	–	–	x
set color to	Farbeinstellungen	x	x	x
set decimals to	Zahl der Dezimalstellungen	–	x	x
set default to	voreingestelltes Laufwerk wählen	x	x	x
set delimiter to	Feldbegrenzungszeichen festlegen	x	x	x
set device to	Ausgabeeinheit festlegen	–	x	x
set fields to	Feldliste festlegen	–	–	x
set filter to	Datenfilter setzen	–	x	x
set format to	Auswahl einer Formatdatei	–	x	x
set format to	Auswahl der Ausgabeeinheit	x	–	-set
set function to	Funktionstastenbelegung	–	x	x
set history to	Anzahl der Befehle im History-Modus	–	–	x
set index to	Indexdateien eröffnen	x	x	x
set margin to	linker Rand für Druckausgabe	x	x	x
set memowidth to	Breite eines Memofeldes	–	–	x
set message to	Nachrichtenzeile festlegen	–	–	x
set order to	Reihenfolge der Indexdateien	–	–	x
set path to	Angabe von Suchverzeichnissen	–	x	x
set procedure to	Prozedurendatei öffnen	–	x	x
set relation to	Relation zweier Datenbanken festlegen	x	x	x
set typeahead to	Größe des Tastaturpuffers festlegen	–	–	x
set view to	Viewdatei öffnen	–	–	x

Tabelle 3. ... während die »set to«-Befehle dBase an Ihre individuellen Bedürfnisse anpassen

A U F   G U T E   Z U S A M M E N A R B E I T !

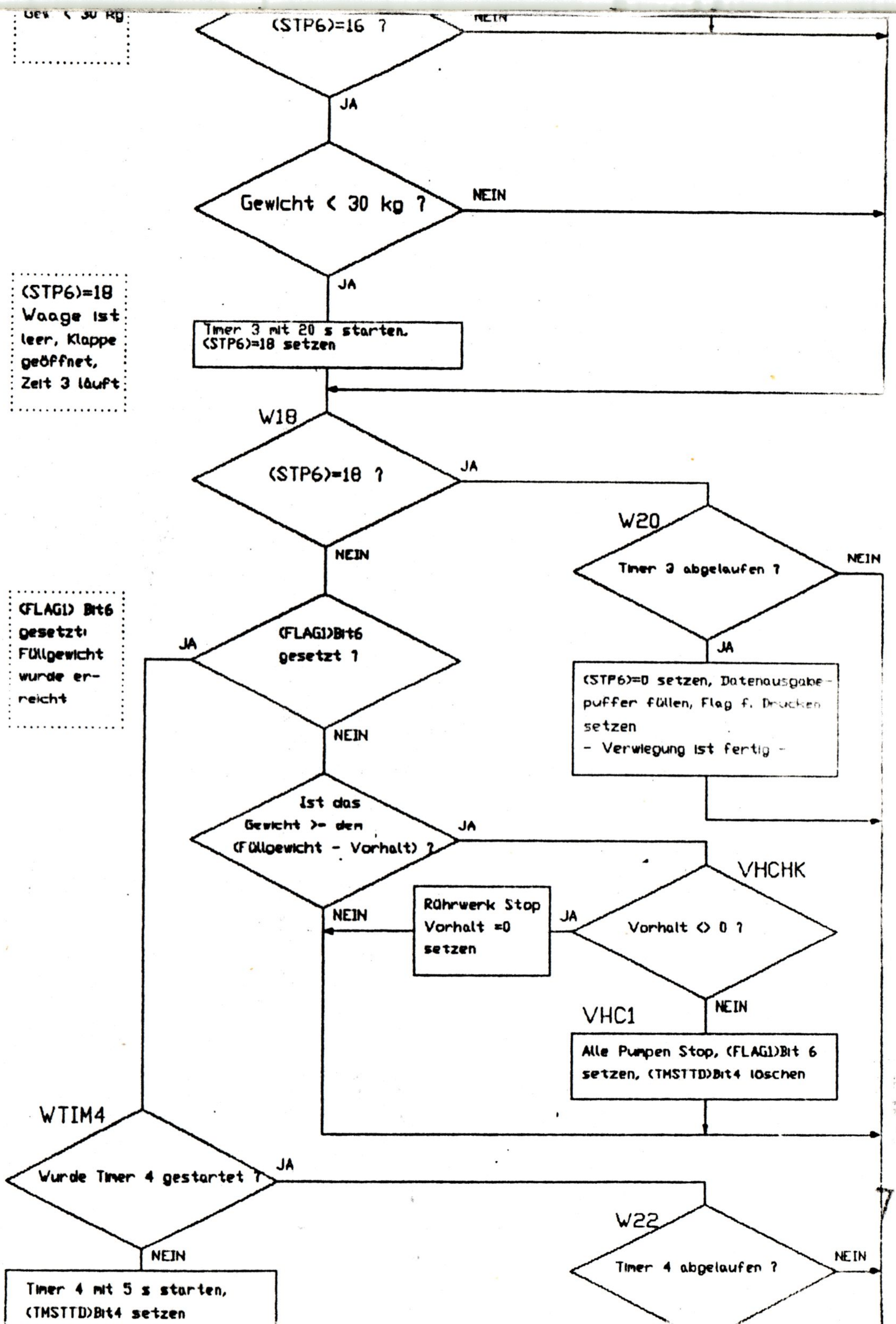
Tandy 4p erstellt das Programm, PC zeichnet das Flußdiagramm

Ein gut strukturiertes Programm braucht keine Hilfsmittel wie Flußdiagramme oder Struktogramme. So kann man es jedenfalls manchmal in der Literatur lesen. Aber, ehrlich gesagt: wenn ich mir eines meiner selbst geschriebenen Assemblerprogramme nach einem halben Jahr anschaue, brauche ich trotz der Kommentare im ZEUS-Sourcecode eine ganze Menge Zeit, um mich in meine früheren Gedankengänge wieder hineinzufinden. Diese Arbeit wird sehr erleichtert, wenn man aus einem Flußdiagramm den Ablauf und die Bedeutung der Timer, Schrittzähler und Flags ablesen kann.

Wer schon einmal Flußdiagramme gezeichnet hat, weiß, wie sehr man auf die räumliche Aufteilung auf dem Zeichenblatt achten muß, wie leicht Zeichenfehler vorkommen und wie schwierig nachträgliche Änderungen und Ergänzungen sind. Als Hilfe für die Dokumentation meiner auf Single Board Computern laufenden Prozeßsteuerungsprogramme verwende ich das Zeichenprogramm 'Autosketch' der Schweizer Firma Autodesk. Mit etwas Übung kann man damit Zeichnungen per Maus genau so schnell am Bildschirm eines MSDOS-Rechners erstellen, wie mit Bleistift und Lineal. Der Vorteil tritt erst richtig in Erscheinung, wenn man vorhandene Zeichnungen ändern muß. Mit wenigen Bewegungen der Maus lassen sich ganze Zeichnungsteile löschen, verschieben, kopieren oder neu erstellen. Das Ergebnis ist eine saubere und übersichtliche Dokumentation. Als Beispiel zeige ich einen Teil eines insgesamt 1,20 m langen Flußdiagramms.

Autosketch ist als Einstieg in das viel umfangreichere CAD-Programm AUTOCAD gedacht und trotz seines bescheidenen Preises schon recht leistungsfähig. Wen man vernünftig damit arbeiten will, braucht man einen PC mit mindestens 640 KByte RAM und einer Festplatte. Ein AT muß es nicht unbedingt sein. Auch auf einem 4,7 MHz-XT mit Coprozessor läuft das Programm ausreichend schnell. Mit einer Hercules-Karte ist die Auflösung auf dem monochromnen Bildschirm genügend fein, die Vorteile der Layer-Technik kommen jedoch erst mit der EGA-Farbgraphik zur Geltung. Zur Ausgabe der Zeichnungen eignet sich ein ganz gewöhnlicher 9-Nadel-Drucker (meine Zeichnung wurde auf einem Star LC-10 erstellt), oder - mit erheblich besserem Ergebnis - ein HP-kompatibler Plotter.

Heinrich Betz



Entwicklungen ..
.. und wie es weitergeht

Der erste elektronische Computer auf Basis von Tausenden von Vakuumröhren wurde 1946 gebaut und hieß ENIAC. Er füllte einen großen Raum, verbrauchte viel Strom, war häufig defekt aber rechnete immerhin viel schneller, als dies zur damaligen Zeit manuell oder elektro-mechanisch möglich war.
Im Jahre 1981, dem fünfunddreißigsten Jahrestag der Geburt des Computers, fuhren einige Studenten den alten ENIAC hoch, der sich nun im Smithsonian Institut befand und verglichen dessen Geschwindigkeit mit der eines damals modernen Mikrocomputers. Der Mikro - in der Größe einer kleinen Schachtel und mit IC's bestückt, die Hunderttausende von Transistorfunktionen enthalten, jeder das Gegenstück zu einer Vakuumröhre - war tatsächlich etwas schneller.

Heute steht auf dem Schreibtisch in vielen Büros ein Rechner mit einer Rechenleistung und Speicherkapazität die Anfang der 70er Jahre noch einem 500.000 DM teuren Minicomputer z.B. einer DEC PDP 11 Ehre gemacht hätte. Trotzdem, die Ansprüche wachsen und man beschwert sich, daß man zu lange warten müßte, alles so mühsehlig sei und es manchmal viele Minuten braucht, bis eine komplizierte Aufgabe wie die Sortierung einer umfangreichen Adressendatei bewältigt ist.

Ich selbst bin ja seit 1983 "EDV-Anwender", seit damals mit dem Genie ein preiswerter Rechner auf dem Markt angeboten wurde, der allerdings im kompletten Ausbau auch mehr als 5.000,- DM gekostet hat. Es war immer mein Interesse darin auch praktische Unterstützung z.B. beim Schreiben von Berichten zu erhalten, weshalb ich mich um Anwendungsprogramme und zuletzt auch um CP/M bemühte. Ich erinnere noch, mit welchem Stolz ich die ersten Serienbriefe druckte. Dies war nur durch mühselige Vorbereitung nach einer langsamen Verarbeitungsprozedur möglich. Dabei galt es immer zu hoffen, daß nicht zwischendurch ein Fehler auftrat und neu begonnen werden mußte.
Immerhin, war ich damit viel schneller, als bei der manuellen Methode und der Aufwand lohnte sich (zumindest für den erprobten wiederholten Durchlauf).

Mein Genie steht noch hier, aber kann sich für meine Aufgaben kaum noch erwärmen, denn seit ich meinen PLANTRON PC-XT mit MS-DOS ausgebaut habe und FRAMEWORK benutze, finde ich darin eine Umgebung vor, die nahezu alles abdeckt, was mir früher nur mit Einzelprogrammen möglich war. Mir ist dabei bewußt geworden, welche ungeheure Entwicklung sich hier vollzieht - schließlich hat es bei den IBM-PC Kompatiblen auch rund 5 Jahre gedauert, bis die Programme einigermaßen ausgereift sind. Mein PC verarbeitet mit rund 5 MB Arbeitsspeicher (704 kB unter DOS plus 4 MB Speichererweiterung) rund 100 mal größere Datenmengen, als es in den 48 kB des Genie möglich war und läuft dabei auch wesentlich flotter und zuverlässiger. Die Software ist - wie angedeutet, inzwischen auch so leistungsfähig geworden, daß sie die Hardware optimal auszulasten vermag.

Gleichwohl sind inzwischen Rechner mit 80386 CPU und 16 oder 20 MHz die "aktuellen" Maschinen, die vom Profi gekauft werden, noch rund 10 mal mehr Power entfalten. Neue Maschinen mit noch mehr Leistung stehen vor der Tür, so werden die ersten 80386er Rechner mit 33 MHz und die 80486 CPU (mit der Leistung von einigen MIPS, vergleichbar einer VAX 785) bereits getestet.

Als interessante Entwicklung dürften wohl die Transputer anzusehen sein. Inzwischen werden vergleichsweise preisgünstige Karten angeboten, die sich an einen PC anschließen bzw. einbauen lassen und diesen Rechner nur für die Ein- und Ausgabevorgänge benötigen, ansonsten aber die Apfelmännchen schon fast in Echtzeit produzieren könnten. Wie ich der aktuellen c't entnehme, lassen sich eine Reihe solcher einzelnen Transputer auch miteinander verknüpfen, so daß die Rechenaufgaben auf einzelnen CPU's parallel verarbeitet werden können, wobei sich die Rechenleistung potenziert! Es läuft wohl darauf hinaus, daß die Leistungsmerkmale der heutigen "ENIAC", nämlich eines GRAY Supercomputers in wenigen Jahren auf dem Schreibtisch verfügbar werden könnte. Damit können die Möglichkeiten der Software noch erweitert werden und Lösungen für komplexere Problemstellungen (Sprachein- und -ausgabe, automatische Übersetzung, Expertensysteme, Bilddatenerfassung etc.) in greifbare Nähe rücken. Übrigens ist es heute schon auf dem PC möglich, Texte aus Zeitschriften, Büchern oder von der Schreibmaschine per Scanner zu digitalisieren und dann per Software in ASCII-Zeichen zu übersetzen und dann mit der Textverarbeitung weiter zu verarbeiten. Hier wird es auch wohl in Zukunft noch weitere Entwicklungen geben.

In Anbetracht des wachsenden Marktes muß ich aber auch selbstkritisch feststellen, daß ich durchaus nicht immer ein Maximum an Leistung brauche, um z.B. "diesen Text zu schreiben".

Mir hat bislang die Erkenntnis aus dem Umgang mit dem Genie und den vielen dafür verfügbaren Programmen bis hin zu KaJot's und Arnulf's "Reiterlied" geholfen, ein grundlegendes Verständnis von diesen Abläufen und technischen Chancen und Grenzen zu erhalten. Ich würde es aber auch sehr begrüßen, wenn wir neue Themen, bis hin z.B. zur Frage der Programmierung eines Transputers vom Genie aus (?) aufgreifen könnten. Zumindest würden mich solche Themen interessieren und ich würde mich über solche Beiträge freuen.

Paul-Jürgen Schmitz
30. Oktober 1988

Datenbanken richtig aufgebaut

Relationale Datenbanksysteme gehören aufgrund der Verkaufszahlen zu den Spitzenreitern der PC-Software. Ihre meist leichte Bedienbarkeit verführt viele Anwender dazu, große umfangreiche Dateien aufzubauen. Zu spät überrascht manchen die bittere Erkenntnis, daß man nicht nur die Bedienung der Software beherrschen muß, sondern auch eine Reihe einfacher, aber wichtiger Entwurfsregeln beachten sollte, um nicht im »elektronischen Chaos« zu versinken.

Ein Datenbanksystem ist wie eine Schlagbohrmaschine: In der Hand des erfahrenen Benutzers ist es ein leistungsfähiges Werkzeug; in der Hand des ahnungslosen Laien läßt sich damit großer Schaden verursachen. Natürlich hinkt dieser Vergleich wie alle: Mit einem »relationalen Datenbanksystem« (zum Beispiel dBase III, K-Man, Rbase oder die entsprechenden Komponenten integrierter Pakete) können Sie nichts in Ihrem PC kaputtmachen! Aber Sie bereiten sich Ärger, wenn Ihre Datenbanken widersprüchliche Informationen enthalten, nicht erweiterungsfähig und flexibel sind oder bestimmte Auswertungen nicht erlauben. Dabei ist alles – wie immer – ganz einfach, wenn man die Regeln zum Aufbau relationaler Datenbanken kennt.

Zuerst sei jedoch definiert, was eine Datenbank ist. Einfach formuliert ist eine Datenbank eine Anzahl von zusammengehörigen, miteinander verknüpften Dateien. Eine Datei wiederum ist (aus Sicht des Anwenders) eine simple Tabelle mit Zeilen und Spalten, die mit einem Namen versehen und über diesen ansprechbar ist, wie Bild 4 zeigt. Die »Verknüpfung« besteht darin, daß Daten einer Datei (Tabelle) zu denen der anderen zuzuordnen sind – eine zweite Tabelle würde zum Beispiel mit der aus Bild 4 eine »Datenbank« bilden, wenn sie Angaben über die »Gebäude« enthielte (Größe, Lage, Adresse); eine mit Angaben über Automarken aber nicht.

An einem simplen Beispiel sehen Sie, was gemeint ist: Ein Makler will seinen PC benutzen, um Daten über Eigentumswohnungen seines Angebotes zu speichern. Für jedes Objekt möchte er beispielsweise aufnehmen: die laufende Nummer, die Adresse (Nummer des Bezirkes und Straße), die Wohnfläche, den Preis, eventuelle Extras, die Telefonnummer des Verkäufers und seinen Namen (diese Liste ist in der Praxis allerdings noch viel umfangreicher). Zur Auswertung wünscht er einen Bericht wie in Bild 1.

Dateien locker aus dem Handgelenk entworfen... aber leider falsch

Nichts ist leichter, als daraus eine Datei (wie gesagt: in Form einer Tabelle) für den PC zu machen, die ein Dateiverwaltungssystem wie dBase III anstandslos akzeptieren würde. Den meisten Anwendern ist bekannt, daß die Definition eines »Schlüssels« bei einer Datei (oder Datenbank) notwendig, zumindest aber sinnvoll ist. Ein Schlüssel ist eine Datenspalte oder ein Feld, das jeden Satz in der Datei eindeutig identifiziert (also nicht zweimal oder noch öfter vorkommt). Zudem ist die Datei meist automatisch nach ihrem Schlüssel sortiert – ein Service der Datenbank Software.

Schlüssel jedes Satzes in der Wohnungsdatei ist also das Feld »Lfd. Nummer« (kurz: Nr.), das jede Wohnung eindeutig identifiziert. Für die Extras bereiten Sie mehrere Felder vor, zum Beispiel drei, so daß Sie maximal drei verschiedene Extras speichern können. So geht es weiter, und schon haben Sie eine einfache Feldaufteilung für Ihre Datei. Ein weiterer Service der Software ist, daß eine »Erfassungsmaske« für die Datei automatisch daraus hergeleitet wird. Darin tippt der Benutzer komfortabel seine Daten ein, wie es Bild 2 zeigt. Gehen Sie davon aus, daß in der Praxis die Längen der Felder nicht so eng bemessen sind wie in unserem Beispiel – es ist ja allgemein bekannt, daß hier bei der Anwendung oft Schwierigkeiten auftauchen. Sollte also als Verkäufer die bekannte »Donaudampfschiffahrtsgesellschaft« in Frage kommen, so muß das Feld entsprechend groß gewählt werden.

Aber leider ist inzwischen schon ein viel gravierenderer Fehler unterlaufen. Alte DV-Hasen kennen das Naturgesetz: »Es ist immer einer mehr oder weniger, als man gedacht hat!«. Irgendwann merkt besagter Makler: »Drei Spalten für Extras sind zu wenig. Nehmen wir lieber fünf.« Was ist zu tun? Er muß die Datei erweitern (neue Felder kommen hinzu) und umladen. Zudem muß er – wegen des geänderten Dateiaufbaus – die Auswerteprogramme ändern. Und wehe, das erste Objekt mit sechs Extras soll in der Datei gespeichert werden. Das Ganze wird immer ärgerlicher, ist aber noch lange nicht das Schlimmste, was einem passieren kann.

NR.	ADRESSE	QM	PREIS	EXTRAS	VERK.TEL.	VERKÄUFER
001	13, Bundesallee 33 a	120	240000	Balkon	123 45 67	Dr. Müller
				Garage		
				Wintergarten		
002	12, Am Weiher 15	85	195000	Anlegesteg	234 56 78	GeSoBau AG
				Garage		

Bild 1. Liste der Eigentumswohnungen

Stellen Sie sich vor, die Datenbank ist ziemlich voll – sagen wir einige hundert Sätze (Dies ist beim Speichervermögen heutiger Systeme auf dem PC ein Klacks). Ein neues Objekt wird bekannt und sofort in seine Erfassungsmaske eingegeben. Dabei passiert leider ein Mißgeschick:

```
VERK.TEL. >2346578<
VERKÄUFER >GeSoBau AG<
```

Was ist passiert? Die Datenbank enthält eine Ungereimtheit, einen Widerspruch – zumindest unter bestimmten Annahmen. In unserem Fall soll es so sein, daß in der Realität ein Verkäufer nur eine Telefonnummer hat. Dann ist der neue Eintrag auf jeden Fall unrichtig: Die »GeSoBau AG« hat die Nummer »234 56 78«. Die Datenbank ist »inkonsistent« geworden.

Wie läßt sich das bereinigen? Im ersten Fall sind die »Extras« sogenannte »Mehrfachfelder« (weil ein Feld mehrfach pro Tabellenzeile beziehungsweise Datensatz vorkommen darf). Regel 1 des »methodischen Datenbankdesigns«, das von Informatik-Wissenschaftlern erarbeitet wurde, verbietet solche Mehrfachfelder. Wie hätte man es machen müssen?

Kombinationsschlüssel schaffen Ordnung, aber auch neue Probleme

»Mehrfachfelder« werden in einer gesonderten Datei gespeichert und nicht in der Originaldatei. Und zwar pro Feld ein Satz, so daß nun beliebig viele Extras (0, 1, 2, ..., viele) pro Wohnung zu erfassen sind. Diese Datei sieht dann beispielsweise aus wie in Bild 3.

Bevor Sie diese Lösung genauer betrachten, schauen Sie sich das Feld »Adresse« noch einmal an. Auch hier wurde ein im wahrsten Sinne des Wortes »elementarer« Fehler gemacht.

Als Feld in einer Datei, als Spalte in einer Tabelle sollte immer nur ein »Datenelement« dienen. Ein Element ist wie in der Chemie eine Einheit, die nicht weiter zerlegt wird (und manchmal gar nicht zerlegbar ist). Nur ein solches Datenelement läßt sich im allgemeinen gesondert betrachten. Unsere Datei läßt sich zum Beispiel hinsichtlich des Bezirkes oder der Straße gar nicht auswerten, zumindest nicht ohne Schwierigkeiten. Das aber ist ja gerade die Stärke von Datenbank-Software, daß Sie Fragestellungen wie: »Welche Objekte im Bezirk 13 sind billiger als 250000 Mark?« mit einem einfachen Befehl der Art

```
list for bezirk = 13 and preis < 250000
```

beantworten. Allerdings erlauben es die meisten Softwarepakete, Feldinhalte zu zerlegen. In unserem Fall könnte man durch die Suche nach dem Komma den Bezirk von der Straßenangabe leicht trennen. In der Praxis ist es jedoch in jedem Fall sauberer, daraus zwei getrennte Felder zu verwenden.

Widmen Sie sich noch einmal den Schlüsseln: Felder, die einen Satz in der Datei (eine Zeile in der Tabelle) eindeutig identifizieren. Was passiert, wenn es kein einzelnes Feld gibt, das dies leistet?

In diesen Fällen benutzt man in der Praxis »Kombinationsschlüssel«: zwei oder mehr Felder, die in Kombination miteinander eindeutig sind. Will man also die monatlichen Gehälter von Mitarbeitern für ein Jahr speichern, so hätte man die Kombination der Felder »Personalnummer« und »Monat« als Schlüssel; wollte man mehrere Jahre speichern, bräuchte man die Jahreszahl als dritte Schlüsselkomponente.

Im Beispiel der »ausgelagerten« Extras in Bild 3 sehen Sie, daß die Datei sogar nur aus ihrem Schlüssel besteht: der Kombination aus »Nr.« und »Extra«. Außer dem Schlüssel ist die Datei hier zufällig leer, es könnten aber genausogut noch (von Wohnung zu Wohnung' verschiedene) Daten gespeichert sein, die sich aber unbedingt auf die Kombination aus beiden Schlüsseln beziehen müssen.

In der Praxis kommt es häufig vor, daß eine Datei nicht nur einen einzigen Schlüssel (wie hier die »Nr.« der Wohnungsdatei) hat, sondern daß eine Kombination aus mehreren Feldern zur Identifizierung eines Satzes herangezogen wird. Vielfach ist dabei das Datum im Spiel. Denken Sie sich eine Datei, in der einzelne Besichtigungen der Wohnungen durch die Interessenten gespeichert werden sollen. Sie hätte als Schlüsselkomponente das Feld »Datum« und zusätzlich... Nun, überlegen Sie selbst, welche Spalten Sie außerdem zur Identifizierung eines Satzes benötigen. Diese aus mehreren Datenelementen (Tabellenspalten) bestehenden Schlüssel, die »Kombinationsschlüssel«, leisten also gute Dienste beim sauberen Entwurf von Dateien – haben aber bei unbedach-

```
NR.        >__<
ADRESSE    >___________________<
QM         >__<
PREIS      >_____<
EXTRAS1    >____________<
EXTRAS2    >____________<
EXTRAS3    >____________<
VERK.TEL.  >______<
VERKÄUFER  >__________<
```

Bild 2. Felder der Wohnungsdatei in der Erfassungsmaske

NR.	EXTRA
001	Balkon
001	Garage
001	Wintergarten
002	Anlegesteg
002	Garage

Bild 3. Datei der Extras der Eigentumswohnungen

NAME	ABT	GEBÄUDE	TEL.
Anders	A12	04	1233
Baier	B02	01	5133
Charcoal	C15	12	1314
Demmer	C15	12	1210
Esser	B02	01	5411
Esser	A12	04	1100

Bild 4. [illegible] fon.dat«

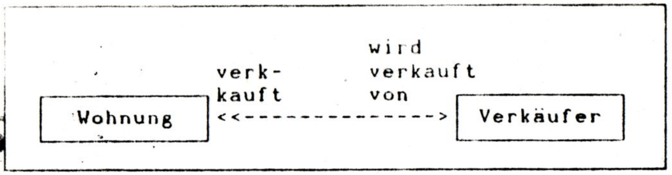


Bild 5. Relationen zwischen zwei Objekten

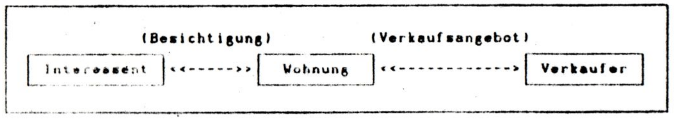


Bild 6. Relationen zwischen drei Objekten

tem Gebrauch auch ihre Tücken. Betrachten Sie folgendes Telefonverzeichnis (Bild 4), bei dem – wie Sie an dem letzten gezeigten Eintrag sehen – nur die Kombination aus »Name« und »Abteilung« eindeutig ist (unter der Annahme, daß nicht zwei »Esser« in derselben Abteilung vorkommen). Sieht doch gut aus, oder? Leider ist es ein Verstoß gegen Regel 2 des Datenbankdesigns: Ein Feld darf nicht von nur einem Teil des (Kombinations-)Schlüssels abhängen! In unserem Fall hängt die in Ziffernform abgekürzte Angabe des Gebäude-Feldes von »Abt« ab. Warum ist das so schlimm? Niemand hindert uns daran, bei einem Neuzugang eine Zeile

```
Huber B02 04 8613
```

mit aufzunehmen – mit falschem Inhalt, da doch die Abteilung »B02« in Gebäude »01« angesiedelt ist. Die Datei enthält somit einen logischen Widerspruch zur Realität. Von der »redundanten« (das heißt mehrfach identisch vorhandenen) Information ganz zu schweigen. Sie kostet uns zumindest Speicherplatz. Die Aussage »Abt. B02 ist in Gebäude 01« wird ja überflüssigerweise in vielen Sätzen gespeichert, obwohl ein Mal reichen würde (nämlich in einer gesonderten Tabelle). Sie werden es schon gemerkt haben: Schon öfter wurde der Ausdruck »unter der Annahme, daß...« gebraucht. Diese Annahmen sind nicht Vereinfachungen unserer Beispiele, sondern wichtige, in der Praxis festzustellende Fakten. Denn wenn es erlaubt ist, daß eine Abteilung über mehrere Gebäude verstreut ist, dann wird aus dem diagnostizierten Designfehler plötzlich eine korrekte Abbildung der Realität. Dieser Sachverhalt läßt sich nur durch Studium der in der Realität herrschenden Regeln und Gesetze feststellen und hat einen bestimmenden Einfluß auf den Entwurf der Datenbanken.

LEISTUNGSMESSUNG

Datum	Vers.Nr.	Ser.Nr.	Typ	Bezeichnung	Sp.[V]	Str[A]	Leistung
18.05.88	2136	712577	GE01	Generator I	220	102	22440
					222	100	22200
					218	99	21681
				Durchschnitt[W]			22107
				Meßabweichg.[%]			3.43
	2137	680455	GE02	Generator II	115	90	10350
					111	93	10323
					110	93	10230
				Durchschnitt[W]			10301
				Meßabweichg.[%]			1.16
19.05.88	2138	712577	GE01	Generator I	219	104	22776
					...	...	.....
... usw.							

Bild 7. Auswertung von Leistungsmessungen

Der Fachmann redet hier von »Objekten« (zufällig derselbe Ausdruck, den der Makler gebraucht). Objekte (oder »Informationsobjekte«) sind Dinge, über die jemand Daten (in Form von Datenelementen) sammelt, und die deshalb durch Schlüssel voneinander unterschieden werden müssen. Die wichtigste Frage ist die nach den Beziehungen der Objekte untereinander – hauptsächlich, ob zu einem Objekt immer und ausschließlich genau ein anderes gehört (und umgekehrt), oder ob es mehrere sein können. Beziehung heißt vornehm (und englisch) »Relation«. Merken Sie etwas? Weil das die wichtigste Aussage in der Datenbank ist, nennt man sie »relational«.

Die Relationen zwischen zwei Objekten müssen immer in zwei Richtungen untersucht werden: hin und zurück. In Bild 5 sehen Sie dies: Eine Wohnung wird von genau einem Verkäufer verkauft (wo käme denn der Makler hin, wenn mehrere Verkäufer dasselbe Objekt anbieten würden?), aber ein Verkäufer verkauft unter Umständen mehrere Wohnungen. Solche Beziehungen nennt der Fachmann »N:1-Beziehungen«. Sie sind von besonderer Bedeutung, wenn sichergestellt ist (per Vorschrift, Gesetz oder gar aufgrund eines Naturgesetzes), daß wirklich nur ein einziges Objekt mit mehreren eine Relation bilden kann. Die Lehrbücher verwenden hier übrigens zur Illustration oft das naheliegende Beispiel »Ehe« zur Darstellung der Varianten »1:1« (Einehe), »N:1«, »1:N« und »N:N« (letzteres als »Chaos« definiert). Diese korrekte Abbildung der Realität ist die wichtigste Voraussetzung für ein einwandfreies Design von Datenbanken.

In Bild 6 sehen Sie eine Erweiterung: Ein neues Objekt namens »Interessent« wird gespeichert. Es steht zur Wohnung in einer N:N-Beziehung: mehrere Interessenten besichtigen ein und dieselbe Wohnung, ein einziger Interessent besichtigt aber seinerseits mehrere Wohnungen.

Es gibt aber noch weitere Regeln zum korrekten Dateientwurf. Ein Problem war ja entstanden, wenn ein Feld nur von einem Teil eines Kombinationsschlüssels abhing. Ein ähnliches Problem tritt auf, wenn Dateifelder nicht von Schlüsselteilen, sondern sogar untereinander abhängig sind. Todsünde Nummer 3!

Ein Beispiel zeigt bereits Bild 1. Hier sind »Verkäufer« und »Verk.Tel« untereinander abhängig. Wieder liegt Redundanz vor und die Möglichkeit zu widersprüchlichen Dateieinträgen, wie sie schon dargestellt wurden. Dabei führen nicht nur Einschübe (»insert«), sondern auch Löschungen (»delete«) und Änderungen (»update«) zu Schwierigkeiten.

Nachdem Sie nun erfahren haben, was man alles falsch machen kann, sollen Sie auch die richtigen Lösungen präsentiert bekommen.

Drei einfache Regeln führen zu sauberem Datenbankdesign

Zum Design einwandfreier »Relationen« (Tabellen, Dateien mit nur einer Satzart) gibt es drei Regeln:

1) Jedes Feld darf im Datensatz nur ein Mal und nicht mehrfach vorkommen.

2) In Dateien mit einem Kombinationsschlüssel muß jedes nicht zum Schlüssel gehörende Feld vom gesamten Kombinationsschlüssel abhangen und nicht nur von Teilen davon.

3) Felder, die nicht Teil des Schlüssels sind, dürfen nicht untereinander abhängig sein.

Die Beachtung dieser drei Regeln liefert Dateien (Relationen), die redundanzfrei sind, und in denen inhaltlich widersprüchliche Sätze nicht vorkommen.

Wäre die Wohnungsdatei nach diesen Regeln entworfen worden, hätten bei Änderung der Zahl der Extras-Felder weder Dateien noch Programme geändert werden müssen. Abgesehen davon lassen solche korrekten Dateien Fragestellungen und Analysen zu, die mit »unsauberen« Dateien nicht realisierbar sind.

Die Vorgehensweise beim Dateientwurf kann nun als Methode wie ein Kochrezept beschrieben werden. Sie sei hier nicht im Detail ausgeführt, aber kurz anhand eines Beispiels verdeutlicht. Dieses wurde bewußt aus einem Bereich gewählt, der den meisten von Ihnen fremd sein wird. Sie werden sehen, daß die Vorgehensweise unabhängig von einer gründlichen Kenntnis des Fachgebietes funktioniert.

In einem Maschinenbaubetrieb soll eine Liste produziert werden, die Meßwerte von Generatoren enthält (Bild 7). Wie gehen Sie nun methodisch an den Dateientwurf heran?

Schritt 1: Welche »Informationsobjekte« (oder »Objekttypen«) sind erkennbar? Auf deutsch: Über welche Gegenstände (im weitesten Sinne) werden Daten gesammelt, und wie (durch welche Schlüssel) werden sie identifiziert?

Im Beispiel erkennen Sie die Schlüssel »Versuchsnr.« und »Seriennr.«. Daraus entnehmen Sie den Objekttyp »Versuch« (Schlüssel »Versuchsnr.«) mit den Daten »Datum«, »Durchschnittsleistung«, »Meßabweichung« und (nicht vergessen!) dem Hinweis auf die »Seriennr.«. Beachten Sie, daß nach Regel 1 Mehrfachfelder nicht erlaubt sind. Es dürfen also nur Datenelemente zugeordnet werden, die pro Informationsobjekt »Versuch« genau ein Mal vorkommen.

Als nächsten Objekttyp identifizieren Sie die »Maschine« (Schlüssel »Seriennr.«) mit den Daten »Typ« und »Bezeichnung«.

Wie Sie den Fall des »Mehrfachfeldes« behandeln, haben Sie schon erfahren – eine separate »Relation« (das heißt eine Tabelle). Das einzige Problem ist der Schlüssel. Die »Versuchsnr.« reicht ja nicht

Tabelle »Versuch«:

SCHLÜSSEL Vers.Nr.	Datum	Seriennr.	Durchschnitt	Meßabweichg.
2136	18.05.88	712577*	22107	3.43
2137	18.05.88	680455	10301	1.16
2138	19.05.88	712577	.....	....

Tabelle »Maschine«:

SCHLÜSSEL Seriennr.	Typ	Bezeichnung
712577	GE01	Generator I
680455	GE02	Generator II

Tabelle »Messung«:

----SCHLÜSSEL--- Vers.Nr.	Lfd.Nr.	Spannung	Strom	Leistung
2136	1	220	102	22440
2136	2	222	100	22200
2136	3	218	99	21681
2137	1	115	90	10350

Bild 8. Erster Entwurf von Relationen

aus, jede einzelne Messung zu identifizieren. Also nehmen Sie eine (auf der Liste nicht sichtbare) laufende Nummer mit hinzu. Daraus entsteht der Objekttyp »Messung« mit dem Schlüssel »Versuchsnr.« und »Lfd.Nr.« sowie den Daten »Spannung«, »Strom« und »Leistung«. Eine übersichtliche Darstellung der Ergebnisse des ersten Schrittes zeigt Bild 8 in Form von »Relationen« (so nennt man auch diese tabellarische Auflistung).

Die Klärung der Beziehungen zwischen verschiedenen »Objekten« ist nicht nur im menschlichen und politischen Leben wichtig. Auch für den Entwurf von sauberen Dateien ist es von höchster Bedeutung, sie korrekt zu berücksichtigen.

Schritt 2: Welche Beziehungen und Häufigkeiten bestehen zwischen den »Objekttypen«?
Im Beispiel erkennen oder erfragen Sie folgende Sachverhalte:

– Es gibt zwar mehrere Messungen pro Versuch, aber jede Messung gehört zu genau einem Versuch (eine »1:N«-Beziehung).

Die präzise Klärung der Beziehungen zwischen den »Objekttypen« ist wichtig

– Es gibt zwar mehrere Versuche pro Maschine (Versuche können wiederholt werden), aber jeder Versuch gehört zu genau einer Maschine.

Ein noch unentdeckter Sachverhalt wird geklärt:

– Es gibt zwar mehrere Maschinen pro Typ, aber jede Maschine gehört zu genau einem Typ.

Schritt 3: »Normalisieren« der Relationen, das heißt Anwendung der drei Regeln auf den ersten Entwurf, um die aufgrund der nun geklärten Abhängigkeiten entdeckten Regelverstöße zu entfernen.
In dem Beispiel erkennen Sie, daß das Vorhandensein der »Bezeichnung« in der Relation »Maschine« ein Verstoß gegen Regel 3 ist – ebenso wie die Speicherung der »Leistung«, da sie ja aus Strom und Spannung direkt errechnet wird. Damit erhalten wir die sauberen Dateientwürfe, wie sie in Bild 9 zu sehen sind. Wären Sie intuitiv, ohne Methode, darauf gekommen?
Diese Relationen sind nicht notwendigerweise vollständig; es könnte etwa zum Maschinentyp ein Datenelement »Nennspannung« und zur Maschine direkt ihr »Herstelldatum« gehören.
Aber Sie sehen sehr schön, wie jetzt durchschaubare Verhältnisse entstanden sind, zum Beispiel durch den »Fremdschlüssel« (das Feld »Seriennr.«) in der Tabelle »Versuch«, der zu weiteren Angaben in der Tabelle »Maschine« führt. Damit läßt sich auch durch Befragen der Fachleute des jeweiligen Anwendungsgebietes sehr leicht der eigene Entwurf überprüfen; hier etwa durch die Frage: »Ist es zutreffend, daß eine Maschine einer bestimmten Seriennummer mehrmals, in mehreren Versuchen gemessen werden kann?«
Die Vorteile der »sauberen« Dateien sind im wesentlichen:

– Daten werden nicht mehrmals (»redundant«) gespeichert. Das spart Platz und (wichtiger!) verhindert sich widersprechende Dateiinhalte.

– Beliebige Fragestellungen sind realisierbar, die in der Regel durch eine in das Datenbanksystem integrierte »Abfragesprache« beantwortet werden können (zum Beispiel: »Welche Leistung(en) wurde(n in eventuell mehreren Versuchen) für Maschine 712577 gemessen?«)

Das interessante Gebiet des methodischen Datenbankdesigns konnte hier nur kurz dargestellt werden. Sicher haben Sie daraus erkannt, wie wichtig es ist, von Anfang an klare und sachlich richtige Verhältnisse herzustellen.

(Jürgen Beetz/zl)

Tabelle »Versuch«:

SCHLÜSSEL

Vers.Nr.	Datum	Seriennr.	Durchschnitt	Meßabweichg.
2136	18.05.88	712577	22107	3.43
2137	18.05.88	680455	10301	1.16
2138	19.05.88	712577	.....	....

Tabelle »Maschine«:

SCHLÜSSEL

Seriennr.	Typ
712577	GE01
680455	GE02

Tabelle »Maschinentyp«:

SCHLÜSSEL

Typ	Bezeichnung
GE01	Generator I
GE02	Generator II

Tabelle »Messung«:

---SCHLÜSSEL---

Vers.Nr.	Lfd.Nr.	Spannung	Strom
2136	1	220	102
2136	2	222	100
2136	3	218	99
2137	1	115	90

Bild 9. Sauberer Entwurf von Relationen

Ein elektronisches Märchen

Es war einmal zur Zeit T=0, da lebte ein armer, aber rechtschaffender Vierpol namens Eddy Wirbelstrom. Er bewohnte einen bescheidenen möblierten Hohlraum mit Dielektrikum und fließend kaltem und warmen Sättigungsstrom. Leider mußte er während der kalten Jahreszeit für die Erwärmung der Sperrschichten noch extra bezahlen. Seinen Lebensunterhalt bestritt er mit einer Verstärkerzucht auf Transistorbasis.
Eddy Wirbelstrom liebte mit der ganzen Kraft seiner Übertragungsfunktion Ionchen. Ionchen, die induktive Spule mit dem kleinsten Fehlwinkel im ganzen Kreis, war die Tochter der einflußreichsten EMK. Ihr remanenter Ferritkörper, ihre symmetrischen Netzintegrale und ihre überaus harmonischen Oberwellen beeinflußten selbst die Suszeptibilität ausgedienter Leidener Flaschen (was viel heißen will!). Ionchens Vater, Cosinus Phi, ein bekannter Industrie-Magnet und Leistungsfaktor, hatte allerdings bereits konkrete Schaltpläne für die Zukunft (T>0) seiner Tochter. Sie sollte nur einer anerkannten Kapazität mit ausgeprägtem Nennwert angeschlossen werden. Aber wie so oft während der Lebensdauer Groß W, der Zufallstrieb wollte es anders.
Als Ionchen eines Tages, zur Zeit T-T, auf ihrem Picofarad vom Frisiersalon nach Hause fuhr – sie hatte sich die neue Sinusstehwelle anlegen lassen – da geriet ihr ein Sägezahn in die Filterkette. Aber Eddy, der die Gegend periodisch frequentierte, eilte mit minimaler Laufzeit hinzu, und es gelang ihm, Ionchens Kippschwingungen noch vor dem Maximum der Amplitude abzufangen und gleichzurichten.
Es ist sicher nicht dem Zufall zuzuschreiben, daß sie sich schon zur Zeit T=T wiedersahen. Eddy lud Ionchen zum Abendessen ins "Ringintegral" ein. Aber das "Ringintegral" war leider geschlossen. "Macht nichts", sagte Ionchen, "ich habe fast 0.2 KHz zu Mittag gegessen und die Sättigungsinduktion hat bis jetzt gehalten. Außerdem muß ich auf meine Feldlinien achten!" Unter irgend einem Vorwand lud Eddy sie daraufhin zu einer Rundfahrt im Rotor ein. Aber Ionchen lehnte ab, ihr würde bei der zweiten Ableitung immer so übel.
Und so unternahmen sie, entgegen den Schaltplänen von Vater Cosinus Phi, einen kleinen Frequenzgang ins nahe gelegene Streufeld.
Der Abend senkte sich über die komplexe Ebene, und am Himmel erglänzten die Sternschaltungen. Eddy und Ionchen genossen die Isolierung vom laufenden Getriebe der Welt. Nur ein einsamer Modulationsbrummer flog vorbei. Sanft plätscherten die elektromagnetische Wellen ans Gestade und leise rauschten die Röhren.
Als sie an der Wheatstone-Brücke angelangt waren, nahm Eddy seinen ganzen Durchgriff zusammen und emmittierte: "Bei Gauß, mein Ionchen, deine lose Rückkopplung hat es mir angetan!".
Der Informationsgehalt dieser Nachricht durchflutete Ionchen mit Groß-Psi. Die Summe über alle n von n=1 bis n dieser Ereignisse war zu überwältigend und sie entglitt der Kontrolle ihrer Zeitkonstanten. Im Überschwang des jungen Glücks erreichten beide vollausgesteuert die Endstufen.
Und wenn sie nicht gedämpft wurden,
dann schwingen sie noch heute.

Dieter Kasper

Ein informatisches Märchen

Fortsetzung der Erlebnisberichte eines Mailbox-Reisenden

--- Fr, 21.10.88 / 03:06:15 von DR.BAKTERIUS

Es war einmal ein kleines, süßes Mädchen, das immer ein Käppchen aus rotem Samt trug. Aufgrund dieses Attributes erhielt es ein Assign unter dem symbolischen Namen "Rotkäppchen". Eines Tages sprach die Mutter:

"Rotkäppchen, die Gesundheit deiner Großmutter hat einen Interrupt bekommen. Wir müssen ein Pflegeprogramm entwickeln und zu ihr bringen, um das Problem zu lösen. Verirre dich aber nicht im Wald der alten Sprachen, sondern gehe nur strukturierte Wege. Nutze dabei immer eine Hochsprache der 4. Generation, dann geht es der Großmutter schnell wieder gut. Und achte immer darauf, daß dein Pflegeprogramm transaktionsorientiert ist, damit es die Großmutter nicht noch mehr belastet".

Da der Weg zum Haus der Großmutter reentrant war, traf Rotkäppchen den bösen Wolf. Er tat sehr benutzerfreundlich, hatte im Background jedoch schon einen Abbruch programmiert. Während Rotkäppchen einen Go-To ins Blumenfeld machte, ging der Wolf im Direktzugriff zur Großmutter und vereinnahmte sie unverzüglich durch einen Delete. Ohne zu zögern gab er sich den Anschein, kompatibel zu sein und nahm die logische Sicht der Großmutter an. Dann legte er sich in ihren Speicherplatz.

Kurz darauf lokalisierte auch Rotkäppchen die Adresse der Großmutter und trat in den Speicherraum. Vor der Installation des Pflegeprogramms machte Rotkäppchen sicherheitshalber einen Verify und fragte:

"Ey Großmutter, warum hast du so große Augen ?"

"Weil ich zufriedene Endbenutzer gesehen habe."

"Ey Großmutter, warum hast du so große Ohren ?"

"Damit ich die Wünsche der User besser verstehen kann."

"Ey Großmutter, warum hast du so ein entsetzlich großes Maul?"

"Damit ich dich besser canceln kann!"

Sprach's und nahm das arme Ding als Input.

Nach einem Logoff begab sich der Wolf zur Ruhe, schlief ein und begann laut zu schnarchen. Als der Jäger auf seinem Loop durch den Wald am Hause der Großmutter vorbei kam, sah er durch ein Window den Wolf im Bett liegen.

"Finde ich dich hier, du alter Error", sprach er," ich habe dich lange gesucht!" Als Kenner der Szene analysierte er sofort, daß nach den Regeln der Booleschen Algebra die Großmutter nur im Bauch des Wolfes sein konnte. Er nahm sein Messer, teilte dem Bauch des Wolfes in mehrere Sektoren und machte, welch' Freude, Großmutter und das Rotkäppchen wieder zu selbständigen Modulen.

Als Input für den leeren Bauch des Wolfes nahmen sie viele Kilobyte Steine und beendeten die Operation mit einem Close. Als der Wolf erwachte, verursachte ihm sein dermaßen aufgeblähter Hauptspeicher solche Schmerzen, daß er an einer Storage Violation jämmerlich zugrunde ging. Da waren alle vergnügt.

Das Pflegeprogramm aktivierte die Großmutter. Rotkäppchen aber dachte:

"Du willst dein Lebtag nie wieder einen Go-To machen, sondern nur noch strukturierte Wege gehen, wie dir's die Mutter geboten hat."

Und wenn sie nicht einem Systemabsturz zum Opfer gefallen sind, duchlaufen sie noch heute die Schaltkreise...

Alexander Schmid

Kostenlos telefonieren

```
+---------------+
! Fun With Fone !
+---------------+
```

aus der B.H.P. #7

Man möchte es gar nicht glauben: Man kann mit dem Telefon auch ohne irgendwelche Gebühren eine Menge Spaß haben. Als Beispiel folgen nun einige Ansagen, die alle dasselbe bedeuten, sich aber trotzdem sehr unterschiedlich anhören.

Die bekannteste Ansage dürfte wohl das gute alte 'Kein Anschluss unter dieser Nummer' sein, das einem - typisch deutsch - mitteilt, dass die gewünschte Nummer nicht existiert, z.B. 0815 - 4711. In München reicht es schon die Ziffern 25 zu wählen. Eine schöne Ansage bietet auch der hohe Norden. Versucht doch mal 04567. Anders sieht es hier im Ausland aus, wo man je nach Land von den verschiedensten Sprechern informiert wird, dass man eine falsche Nummer erwischt hat:

Nordamerika:
001212 1234567 USA/New York
001213 1234567 USA/Los Angeles

Südamerika:
00503 123456 El Salvador
00506 123456 Costa Rica
0054328 123456 Argentinien
005631 123456 Chile

Afrika:
00203 1234567 Ägypten/Alexandria
0021821 123456 Libyen
002542 123456 Kenia
002601 123456 Sambia

Arabien:
009626 123456 Jordanien/Ammann, sehr laut, man beachte die originelle Aussprache.
009662 123456 Saudi-Arabien

Asien:
00822 123456 Rep. Korea
008865 123456 Taiwan

Anmerkung:

Die japanische Zeitansage ist unter 0081 777 777 7... (so lange eine 7, bis was zu hören ist) zu erreichen.

Alexander Schmid

Abschied von der Tastatur

Die komplexen Prozesse zur Informationsverarbeitung, die dem Verstehen natürlicher Sprache zugrunde liegen, werden mit rechnerorientierten Methoden untersucht. Mit Computern, die Sprache verstehen können, ist der Mensch-Maschine-Informationsaustausch schneller, akkurater und effizienter durchführbar und – last, not least – „menschlicher“.

Um gesprochene Sprache zu verstehen, müssen die Schallwellen eines gesprochenen Wortes mit dem Schallmuster eines gespeicherten Wortes verglichen werden, um zunächst die syntaktische (den Satzbau betreffende) Abfolge der Wörter zu analysieren. Das Problem ist nun, daß das Schallmuster eines Wortes je nach Stellung im Satz und je nach Sprecher sehr verschieden sein kann und zudem noch aus einem Geräuschpegel herausgefiltert werden muß.

In der sprachorientierten KI-Forschung unterscheidet man neben der Syntax zwei weitere Ebenen des Verstehens: Die Semantik, die sich mit der Bedeutung des Satzes beschäftigt und die Pragmatik, die sich mit dem praktischen Umsetzen der Äußerung befaßt.

Der Vorteil von Spracherkennungssystemen liegt auf der Hand, denn Tastatureingaben von Texten bedingen die Gefahr, typographische Fehler zu begehen, die unbemerkt bleiben. Derartige Fehler, die nicht auf „logische“ Falscheingaben zurückzuführen sind, können mit den neuen Systemen verhindert werden.

Die technologischen Konsequenzen sind offensichtlich, zumal der Benutzer selbst zum Umgang mit solchen Systemen keinerlei EDV-Kenntnisse mehr, wie etwa bei der Verwendung von Terminals benötigt.

Zur Zeit werden „sprachgesteuerte Schreibmaschinen („voice activated typewriter“) weltweit in zahlreichen Forschungsinstituten entwickelt. Der Wortschatz dieser Spracheingabesysteme umfaßt teilweise bis zu 20 000 Wortformen. In der Bundesrepublik werden sprachverarbeitende Systeme in Berlin, Erlangen, Hamburg und Saarbrücken entwickelt. ddt

81

KI-Anwendung: Einsatz von Spracherkennungs-Computern in der Medizin

Statt Sekretärin oder Diktiergerät benutzen amerikanische Mediziner – wie die Röntgenärztin im linken Bild – zunehmend Spracherkennungs-Computer, die medizinisches Vokabular verstehen. Einer der Vorteile: Die Maschine des KI-Wissenschaftlers Raymond Kurzweil (rechts) versteht selbst Englisch mit stärkstem Akzent. FOTOS: ERNST ULRICH

Wie man den Computer zum Diktat bittet

Schwerer Verkehrsunfall. Zwei lebensgefährlich Verletzte werden in die Unfallstation des Nash General Hospital eingeliefert. Nachdem das eingespielte Team die Verletzten versorgt hat, muß der Chefarzt seine Berichte abfassen, medizinische Buchhaltung sozusagen: „Eine verantwortungsvolle Pflicht, von der Außenstehende gar nicht ahnen, daß sie bis zu 30 Prozent unserer Arbeit als Notfallarzt ausmacht“, so Dr. Arthur Chambers, Direktor der Notfallstation des Nash-Krankenhauses von Rocky Mount in North Carolina.

Bis vor kurzem diktierte Dr. Chambers seine Protokolle auf Tonband. Die besprochenen Kassetten gingen an den Schreibpool des Hospitals, wurden abgeschrieben, dem Doktor zum Gegenlesen vorgelegt, korrigiert, im Zweifelsfall noch einmal geschrieben – ehe sie auf den angestammten Dienstweg gebracht werden konnten. Seit neuestem steht in seinem Dienstzimmer – und in einem Dutzend weiterer US-Hospitäler – ein spezieller Protokoll-Computer, der die Arbeit des Unfallarztes erleichtert und insbesondere die Zeitspanne bis zur Ablieferung der Berichte ganz erheblich verkürzt. Der Computer versteht nämlich, was Dr. Chambers diktiert, mehr noch: Er setzt den gesprochenen Protokolltext sogleich um und schreibt ihn auf den Bildschirm.

Auch die Änderungswünsche, die der Doktor beim Korrekturlesen seiner Texte von sich gibt, führt der Computer auf der Stelle aus. Und auf das (gesprochene) Kommando „Print this“, druckt er schließlich das fertige Protokoll formgerecht auf einer Karteikarte aus. Der Diktier-Computer repräsentiert eine der ersten praktischen Anwendungen dessen, was in der Zunftsprache der Computer-Leute mit den Begriffen wie „Hands-off system“ oder „Voice recognition“, einem Teilgebiet der Künstlichen Intelligenz, umrissen ist.

Es handelt sich bei Dr. Chambers Computer um einen ganz normalen PC, kombiniert mit einem Spracherkennungsmodul aus der Werkstatt der kleinen Bostoner High-Tech-Firma Kurzweil Applied Intelligence Inc., einem Spaltprodukt des legendären Massachusetts Institute of Technology (MIT). Firmengründer Raymond Kurzweil beschäftigt sich seit fast zwei Jahrzehnten mit dem schwierigen Problem, die menschliche Sprache elektronisch zu erfassen, zu analysieren und zu computerisieren.

Durch Zufall wurde 1984 der Röntgenmediziner Alan H. Robbins auf die Experimente Kurzweils aufmerksam – und die Verbindung der beiden blieb nicht ohne Folgen. Dr. Robbins, Professor für Radiologie an der Tufts University School of Medicine in Boston, suchte seit langem eine rationelle Methode, die bei der Analyse von Röntgenaufnahmen erkannten Befunde schnell und sicher zu erfassen: „Der Röntgenarzt ist es gewöhnt zu sprechen, während er das Bild betrachtet; man kann ihn nicht dazu bringen, seine Arbeitsweise zu ändern.“

Deshalb galten in den Augen des Professors die verschiedenen Versuche, Protokollsysteme auf der Basis von Strichcodes oder berührungsgesteuerten („Touch screen“-)Computern zu schaffen, als gescheitert, „weil der Arzt dabei gezwungen ist, den Blick vom Röntgenbild weg, dem Erfassungssystem zuzuwenden“. Versierte Schreibdamen aber, die auch das fachmedizinische Vokabular beherrschen, sind selten und teuer. Robbins: „Ich muß nicht selten mehrere Tage auf das Abtippen meiner Berichte warten.“

So unterzog sich der Röntgenprofessor der Mühe, für das im industriellen Bereich – zum Beispiel bei Qualitätskontroll-Stationen am Fließband – bereits bewährte Kurzweil Voice System (KVS) medizinspezifische Software zu erarbeiten.

Er untersuchte 12 000 Röntgenreporte und dampfte das darin vorkommende Vokabular auf einen Grundbestand von knapp 1000 Wörtern ein: Es sind die 1000 Begriffe, die der Computer verstehen muß, um in der Röntgendiagnose effizient arbeiten zu können. Inzwischen wird „VoiceRAD“, das so entstandene System für Radiologen, in den USA vermarktet und hat seine Bewährungsprobe bestanden. Es kostet etwa soviel wie neun Monate Gehalt für eine geeignete Typistin – rund 16 000 Dollar. Zuverlässigkeit der Spracherkennung: 95 Prozent.

Wer damit arbeiten will, muß sich und seine Stimme zunächst dem Computer vorstellen. Das geschieht, indem der Arzt oder die Ärztin eine Liste von 209 ausgesuchten Wörtern – von „Yes“ bis „Artherosclerotic“ – vorspricht, was gewissermaßen der Einstimmung der Maschine auf den menschlichen Partner dient. Danach folgt ein zweimaliges Einlesen des 1000-Worte-Fachvokabulars.

Aus den personentypischen „Voiceprints“ dieser Wörter baut sich der Computer ein Lexikon, das ihm später während der Routinearbeit als Vergleichsbasis dient: Sobald das Spracherkennungssystem mit Hilfe von acht speziellen Filter-Chips Übereinstimmung eines gesprochenen Wortes mit einem gespeicherten Voiceprint feststellt, schickt es die entsprechenden Buchstaben auf den Bildschirm.

Und den Befehl „Signature“ (Unterschrift) quittiert das System mit ihrem vollen Namen plus Datum und Uhrzeit. Röntgenärztin Dr. Miriam Vincent: „Die meisten Protokolle sind in dem Moment fertig, in dem ich das Röntgenbild in seinen Umschlag zurückstecke.“

Aufgrund der guten Erfahrungen der Röntgenärzte hat die Kurzweil Inc. die Anwendungen der Spracherkennung auch auf andere Bereiche der Medizin ausgedehnt. So gibt es mittlerweile – von Mammographie bis Neuroradiologie – bereits fünf spezifische Sprachmodule für verschiedene Bereiche der Röntgenmedizin. Daneben trimmte der Traumatologe Dr. Chambers zusammen mit Software-Ingenieuren das System auf die Besonderheiten der Notfallmedizin; nach erfolgreicher Testphase im Nash Hospital von Rocky Mount wird es jetzt unter dem Namen Voice-EM (für Emergency medicine) in den USA angeboten.

Geplant sind weitere medizinische Applikationen, etwa in der Chirurgie und Pathologie, sowie eine Erweiterung der Voiceprint-Bestände auf einen Umfang von zunächst 5000, später sogar 10 000 Wörtern.

Ein unerwarteter Vorzug der KI-Computer zeigte sich schon bald nach der Installierung des allerersten VoiceRAD im Veteran's Hospital in Boston. Während die Schreibdamen der Klinik stets Probleme hatten, einwandfreie Transkripte von den Tonbändern des mit stark indischem Akzent sprechenden Röntgenarztes Dr. Srinivasan herzustellen, gelingt das dem Computer tagtäglich so gut wie fehlerfrei. Er ist auf die Stimme des Inders trainiert.

ULRICH BLUMENSCHEIN

Nächsten Montag:

Gespräch mit dem technisch-wissenschaftlichen Direktor des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI), Professor Gerhard Barth.

82

Ein „Wurm" legte Tausende von Computern lahm

Student schleuste Virus ein / Programm so lange kopiert, bis die Speicherkapazitäten erschöpft waren / Von Horst Rademacher

SAN FRANCISCO, 6. November. Russell Brand wollte sich am vergangenen Mittwoch eigentlich einen ruhigen Abend in seinem Haus in den Hügeln von Berkeley machen. Der 26 Jahre alte Informatiker arbeitet am Lawrence-Livermore-Labor, einer der beiden [illegible]-gen, in denen amerikanische Kernwaffen entwickelt werden. Wie an jedem Abend schaltete der Wissenschaftler kurz vor Mitternacht noch einmal seinen Heimcomputer ein und stellte eine Verbindung mit dem Rechner in seinem Büro her. Normalerweise startet Brand dabei Programme, die die Computer in Livermore während der Nachtstunden berechnen. Wenn er am nächsten Morgen ins Büro kommt, liegen bereits die Ergebnisse seiner Berechnungen vor.

Doch in der Nacht zum Donnerstag spielte der von Brand benutzte Computer in Livermore regelrecht verrückt. „Das System antwortete sehr langsam", berichtet Brand. „Nachts werden die Computer sonst kaum benutzt, doch an diesem Abend waren die Rechner voll." Diese ungewöhnliche Aktivität machte Brand stutzig. Er schaute sich die Programme, die der Computer damals berechnete, etwas genauer an. Dabei fiel ihm ein Programm mit dem Namen „SH" auf, das er noch nie vorher gesehen hatte. Es war offenbar von außen in die Maschine geschleust worden war. Brand rief daraufhin jene Kollegen an, die für die Computersicherheit in Livermore zuständig sind. Gemeinsam ging man der Sache nach.

Unterdessen kämpften Darren Griffith und seine Kollegen an der Universität von Berkeley schon seit sechs Stunden mit ihren Computern. Sie waren gegen 18 Uhr vom Abendessen in einer Pizzeria in das Institut für experimentelle Informatik zurückgekehrt. Ein Blick auf die Bildschirme zeigte den Experten, daß irgend etwas mit ihrem System nicht stimmte. Auch sie fanden das mysteriöse SH-Programm in ihren Rechnern. „Es blockierte unseren Rechner vollständig", sagte Griffith.

Weder Brand noch Griffith merkten zu dieser Zeit, daß ihre Computer vom gleichen „Virus" befallen worden waren. Den Nachtarbeitern in vielen Computerzentren in allen Teilen Amerikas ging es ebenso. An den berühmten Universitäten wie Harvard, Stanford, Cornell und dem Massachusetts Institute of Technology (MIT) lief nichts mehr. Die Computer von Unternehmen liefen ebenso Amok wie einige Rechenmaschinen der Nasa und die zahlreicher militärischer Einrichtungen. Am Donnerstagmorgen waren mehr als 6000 Computer in Amerika von diesem Virus befallen. Alle diese Rechenmaschinen hatten nur eines im Sinn: Sie kopierten das SH-Programm immer und immer wieder – so lange, bis ihre Speicherkapazität erschöpft war und sie sich selbst abschalteten.

Am Donnerstag stellte sich heraus, daß nur eine bestimmte Gruppe von Computern infiziert war. Es handelte sich ausschließlich um VAX-Computer und die sogenannten Sun-Workstations. Sie alle arbeiteten unter dem Betriebssystem Unix 4.3 und waren ausnahmslos an eines der größten Datenverbundnetze der Welt angeschlossen, das Arpanet. Raymond Colladay, der Leiter der zum Pentagon gehörenden und Arpanet betreibenden Defense Advanced Research Projects Agency (Darpa), rief den Notstand aus. Das sei der größte Angriff, der je auf amerikanische Computer unternommen worden sei, meinte er. Das ebenfalls zum Verteidigungsministerium gehörende National Computer Security Center (NCSC) in Bethesda (Maryland) richtete ein Krisenzentrum ein. Bei der „New York Times" meldete sich unterdessen ein anonymer Anrufer. Einer seiner Freunde, ein Informatikstudent, habe das Computervirus erzeugt und in den Datenverbund eingeschleust, erklärte er. Allerdings habe er einen Programmierfehler gemacht. Was ursprünglich nur als Spaß gedacht war, legte nun den gesamten Datenaustausch zwischen 300 Universitäten, privaten und industriellen Forschungseinrichtungen sowie verschiedenen militärischen Stellen lahm.

Informatikern am Lawrence-Livermore-Labor und bei der Firma SRI in Palo Alto (Kalifornien) gelang es schließlich, das mysteriöse Programm zu isolieren und zu untersuchen. Es handelte sich um ein relativ kurzes Programm von nur 99 Zeilen Länge. Es war in der Programmiersprache „C" geschrieben. Es wies die befallenen Computer an, das Programm selbst so oft wie möglich zu kopieren. Ein anderer Teil dieses Programms suchte unterdessen in den Dateien nach möglichen Verbindungen zu anderen Computern. Wenn es solche fand, kopierte sich das Programm selbst über Arpanet auf andere Maschinen. Dort begann es sein zerstörerisches Werk von neuem. Das Programm sei äußerst intelligent geschrieben, meinten die Experten. Es mache sich einige bekannte Unzulänglichkeiten in jenen Systemprogrammen zunutze, die Arpanet mit den Unix-Computern logisch verbinden. Nachdem klar war, wie das Programm arbeitete, konnten die befallenen Computer in wenigen Stunden desinfiziert werden. Am Wochenende, so hieß es bei Darpa, habe das Netz wieder normal gearbeitet.

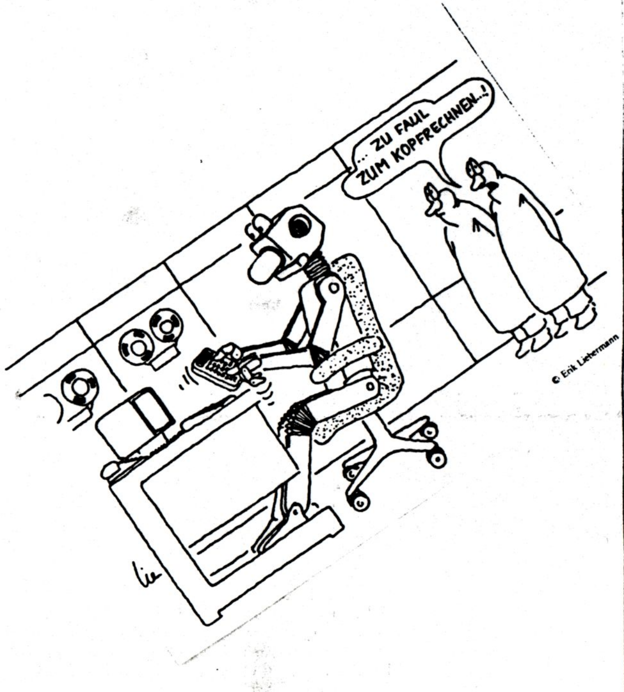
Viele Experten waren überrascht, daß es einem einzigen Programm gelingen konnte, in wenigen Stunden mehr als 6000 Computer lahmzulegen. Noch überraschter waren sie jedoch, als sich der Übeltäter stellte. Es handelte sich um den 23 Jahre alten Robert Morris Jr., einen Informatikstudenten an der Cornell-Universität in Ithaca (New York). Sein Vater, Robert Morris Sr., ist am NSCS als Chefwissenschaftler angestellt und gilt als der Fachmann für Computersicherheit in Amerika. Als Spezialgebiet befaßt sich der Vater mit der Sicherheit von Unix-Systemen.

Auf den ersten Blick sieht es so aus, als habe der Sohn mit dem Computervirus seinem alten Herrn einen Streich spielen wollen. Doch viele Experten hatten seit langem damit gerechnet, daß Arpanet zum Ziel von Hackern werden würde. Erste Computer wurden schon im Jahre 1969 diesem Datenverbund angeschlossen. Damit wollte Darpa, der Forschungsarm des Pentagons, Wissenschaftlern und Ingenieuren, die an von ihr geförderten Projekten arbeiteten, den Datenaustausch und die Kommunikation erleichtern. Im Laufe der Jahre schlossen sich immer mehr Universitäten [illegible] an dieses Netz an. Selbst Computer in Europa und Australien wurden angekoppelt. Im Jahre 1983 entschloß sich das Pentagon, einen neuen Datenverbund einzurichten. Der Austausch geheimer Daten wurde aus Arpanet ausgelagert und wird seitdem über Milnet abgewickelt.

Von den meisten amerikanischen Universitäten ist der Zugang zu Arpanet relativ einfach. Deshalb befürchteten Sicherheitsexperten schon seit langem, daß clevere Informatikstudenten irgendwann ein Virus in dieses System einschleusen würden. Bisher blieb das Netz davon jedoch verschont. Wie Morris junior sagte, hatte er nicht vor, das gesamte Netz lahmzulegen. Er wollte lediglich einen „Wurm" einschleusen, der sich allmählich durch die an Arpanet angeschlossenen Computer „frißt". Dabei sollte er keine Spuren hinterlassen – bis auf eine. Morris hatte seinen Computerwurm so programmiert, daß jedes Programm sich regelmäßig bei einem Computer in Berkeley melden sollte. Auf diesem Computer mit dem Namen „Ernie" hatte Morris eine Datei angelegt. In ihr sollten die Meldungen gesammelt und damit die Ausbreitung des Wurms dokumentiert werden. Morris habe damit zeigen wollen, wie schlecht der Datenverbund gegen unbefugte Benutzung gesichert ist.

Doch durch einen kleinen Programmierfehler geriet ihm der Wurm außer Kontrolle. SH kopierte sich tausendmal schneller, als es Morris vorgesehen hatte. Da jede dieser Kopien ihren Standort an Ernie in Berkeley zurückmeldeten, hatten Griffith und seine Kollegen besondere Schwierigkeiten, ihre Rechner unter Kontrolle zu bringen.

Seit diesem Zwischenfall mit Arpanet wird in Amerika wieder über die Computersicherheit diskutiert. Zwischen August 1986 und Juni vergangenen Jahres war es zum Beispiel einem Hacker aus Hannover gelungen, sich über das Datennetz der Bundespost (Datex-P), dem vom Flugzeughersteller McDonell-Douglas betriebenen „Tymnet" und Arpanet Zugang zu vielen amerikanischen Computern zu schaffen. Er stöberte dort in Dateien herum und suchte offenbar nach geheimen Informationen über SDI. Im September wurde erstmals in Amerika ein Programmierer wegen eines Computervirus verurteilt. Der Verurteilte, Donald Burleson, arbeitete für eine Versicherungsgesellschaft in Fort Worth (Texas). Er wurde schuldig gesprochen, kurz vor seiner Entlassung die Computer der Gesellschaft mit einem Virus infiziert zu haben, das die Informationen über 168 000 Versicherte zerstört hat. Burleson konnte jedoch nur verurteilt werden, weil er wirklich Schaden in den Dateien angerichtet hatte. Im Gegensatz dazu blockierte das Programm des jungen Morris zwar für Stunden die an Arpanet angeschlossenen Computer. Es zerstörte jedoch weder Daten noch hinterließ es – nachdem die Rechner desinfiziert waren – irgendwelche Spuren.



© Erik Liebermann

CLUB-80 Bibliothek

In der Club-Bibliothek sind folgende Neuzugänge zu verzeichnen:

- Tischcomputer-BASIC für Anfänger
  (ca. 200 Seiten)
  meiner Meinung nach nur für absolute Newcommer

- BASIC-übungen für Anfänger
  (ca. 160 Seiten)

- Disk-Interfacing-Guide
  (ca. 50 Seiten)
  für Hardware-Freaks

- Programmieren in Maschinensprache mit Z80
  (ca. 80 Seiten - Hofackerqualität?)

Gleichzeitig haben wir 3 Bücher zu verschenken - und zwar an den, der sich als Erster bei mir meldet:

- Programmieren mit TRS-80
  (ca. 200 Seiten)
  bekanntestes Einsteigerbuch für TRS-80

- TRS-80 Disk And Other Mysteries
  (ca. 100 Seiten)
  erläutert die Diskettenzugriffe

- TRSDOS 2.3 Decoded And Other Mysteries
  (ca. 300 Seiten)
  enthält die Assembler-Listings für TRSDOS 2.3

Das war's von der Club-Bibliothek

Werner

- I M P R E S S U M -

1. Vorsitzende
Gerald SCHRöDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

2. Vorsitzende
Bernd RETZLAFF
Kleiner Sand 98
2082 Uetersen
☎ 04122 /43551

Hardwarekoordinator
Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskothekar
Club-Bücherei
Werner FöRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion
Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Es erfolgt keine Zensur oder Kontrolle der jeweiligen eingeschickten Infobeiträge durch die Redaktion.

Autoren

Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck
Horst-Dieter Schroers
Breslauer Straße 9
8016 Feldkirchen
☎ 089 / 9032615

Bankverbindung des CLUB 80
Postgirokonto Peter STEVENS
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

HEFT
25
Januar
1989

Hallo Club-80er,

e n d l i c h ...
... werdet Ihr sagen, erscheint nun das neueste INFO.
Die Club 80 Redaktion ist aus Ihrer Versenkung wieder aufgetaucht !

Das übliche Sonderheft am Jahresende, mit dem Jahresinhaltsverzeichnis und der Kompletten Auflistung unserer Software sowie der vollständigen Bücherliste unserer Club-80-Bibliothek, ist leider noch nicht fertig.
Werner hat Probleme mit seinem Drucker.
Es wird also alsbald noch einen Nachschlag für dieses INFO geben.

Und das Nächste soll ca. 3-4 Wochen nach Erscheinen des 26. INFO's folgen. Dazu brauche ich aber Eure Mithilfe. Ich suche noch Infomaterial. Ihr werdet ja in der Zwischenzeit schon fleißig für das 27. INFO gesammelt haben? Ich hoffe es.

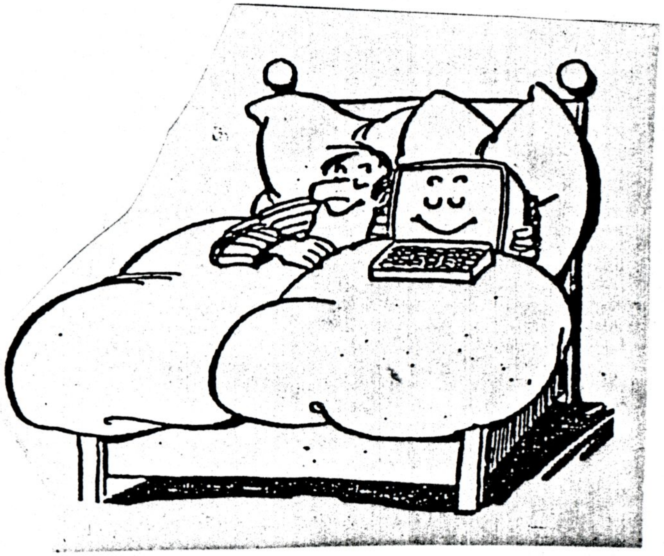
Als weiteres möchte ich ein Sonderheft aus den in INFO 25 als Liste vorgestellten ELCOMP-Artikeln machen. Ich hoffe, daß das dann bis zum Clubtreffen fertig ist.

Diesem INFO liegen wieder zwei Referenzkarten am Heftende bei.
dBase II
Alcor Pascal

Bis zum nächsten INFO wünsche ich Euch eine gute Zeit und vergesst nicht, ich suche noch Infoartikel für das 27. INFO !!
Termin in ca 3 Wochen.

Es grüßt Euch Euer

Jens



CLUB 80 Mitgliederadressenliste

Stand Januar 1989

Bitte überprüft Eure Daten auf Richtigkeit und teilt mir Unregelmäßigkeiten mit.
Die Redaktion

Name	Vorname	Straße	PLZ Stadt	Telefon privat	// geschäftlich
Ahlers	Armin	Rotdornstraße 13	2116 Hanstedt / Nindorf	04185 /7819	// -
Albers	Herbert	Zum Düwelshöpen 14	2117 Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Würfelser-Kaule 30	5060 Bergisch-Gladbach 1	02204 /62781	// -
Bernhardt	Helmut	Hafenstraße 7	2305 Heikendorf	0431 /241907	// 0431 /74047
Betz	Heinrich	St. Wolfgangsstraße 13	8551 Hausen	09191 /31698	// 09191 /611108
Brandl	Hermann	Cäciliastraße 30	8439 Postbauer-Heng	09180/ 493	// 0911 /219-245
Brans	Jörg	Tieloh 55	2000 Hamburg 60	040 /6912716	// -
Braun	Harald	Postfach 8011	2300 Kiel 17	-	// -
Bröcker	Detlev	Eisenstraße 92	4000 Düsseldorf 1	0211 /772436	// -
Böckling	Ulrich	Am Sonnenhang 11	5414 Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200 Wiesbaden-Bierstadt	06121 /508218	// -
Eilers	Hans-Joachim	Theodor-Heuss-Straße 129	2900 Oldenburg	0441 / 53239	// 0441 / 7983845
Förster	Werner	Christoph-Krebs-Straße 9	8720 Schweinfurt	09721 /21841	// 09721 /51256
Hartmann	Hans-Günter	Möwenstraße 9	2876 Berne 2	04406 / 1911	// 0421 / 2483405
Heidenreich	Ulrich	Werderstraße 35	4300 Essen 1	0201 / 282770	// -
Held	Manfred	Stirner Straße 22	8835 Pleinfeld	09144 /6563	// 0911 /2195245
Hentz	Werner	Am Tränkgarten 20	6457 Maintal 2	06109 /66625	// 0691 /583639
Hermann	Klaus	Forchenstr. 8	7401 Pliezhausen	07127 /71945	//
Homann	Matthias	Maschener-Straße 49 A	2105 Seevetal 1	04105 / 50222	// 040 / 7511-278
Issig	Günter	Teutonenstraße 20	7100 Heilbronn	07131 /42601	// -
Jablotschkin	Rainer	Südertor 10	4780 Lippstadt 8	02921 / 23245	// 02921 /70431
Kostya	Mary Jo	Balberstraße 68	CH-8038 Zürich	00411 / 482894	// 00411 / 258170
Krahn	Sven	Am Affenberg 20	7730 Villingen-Schwenningen	07721 /1765	// -
Krispin	Michael	Schwanstraße 8	4130 Moers 3	02841 /73690	// -
Kuhn	Eckehard	Im Dorf 14	7443 Frickenhausen 1	07022 /45417	// 09171 /832665
Lachmann	Wolfgang	Am Ringofen 11	6407 Schlitz	06642 /6940	// 069 /832051
Lohmann	Willi	Honnerskamp 2	4300 Essen 12	0201 / 356319	//
Loose	Gerhard	Viefhaushof 42	4306 Essen 13	0201 /212608	// -
Magnus	Andreas	Pommernstraße 18	4650 Gelsenkirchen	0209 /870230	// -
Mahlert	Herbert	Baumschulstraße 7	4100 Duisburg 14	02135 /81462	// -
Mand	Harald	Kleinflintbeker Straße 7	2302 Flintbek bei Kiel	04347 /3629	// 0431 /3013580
May	Holger	Marienstr. 9	5768 Sundern 2	02935 /1668	// -
Meklenburg	Heinz-Dieter	Amruner Weg 1	2262 Leck /NF	04662 /691	// -
Menk	Christian	Ollsener Straße 52	2116 Hanstedt	04189 /7825	// -
Misioch	Waldemar	Adenauerring 25	8505 Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /107945
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940 Weinheim -Lützelsachsen	06201 /55052	// -
Müller	Kurt	Soltaustraße 24a	2050 Hamburg 80	040 /7246083	// 04151 /8891-37
Neueder	Jens	Rudolf-Then-Straße 32	7178 Gschlachtenbretzingen	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209 Heidenrod 1	06124 /3913	// -

CLUB 80 Mitgliederadressenliste

Name	Vorname	Straße	PLZ	Stadt	Telefon privat // geschäftlich
ObscherningKat	Helmut	1 rue des Bruyeres	F-68360	Soultz	0033/89-762690 // -
Piller	Walter	Rohnenstraße 8	CH-8835	Feusisberg	01 /7847418 //
Preußing	Wolfgang	Dammstraße 68	4300	Essen 14	0201 /530878 // -
Raggan	Hans	Backnanger Weg 36	7146	Tamm	07141 /603611 // 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080	Fürstenfeldbruck	08141 /43791 //
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128	Lauffen am Neckar	07133 /4167 // 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082	Uetersen	04122 /43551 // 04103 /605310
Rinio	Gerd	Rennbahnstraße 9	2000	Hamburg 74	040 /6552630 // -
Ruschinski	Claus	Pommernstraße 21	4370	Marl	02365 /34646 // -
Rychlik	Andreas	Königsberger Allee 120	4100	Duisburg 1	0203 /331383 // 0203 /331383
Schmid	Alexander	St. Cajetan-Straße 38/VII	8000	München 80	089 /495326 // -
Schmitz	Paul-Jürgen	Bremer Straße 9	6236	Eschborn	- // -
Schneider	Bernd	Wirtwis Straße 13	CH-8951	Fahrweid	0041/1/7480348 // -
Schober	Frank-Michael	Weberweg 2	DDR7590	Spremberg	003 /75/4-4565 // -
Schrewe	Christian	Fliederweg 32	4000	Düsseldorf 31	0203 /740897 //
Schroers	Horst-Dieter	Breslauer Str. 9	8016	Feldkirchen	089 / 9032615 // 089 / 9032615
Schröder	Gerald	Am Schützenplatz 14	2105	Seevetal 1	04105 /2602 // -
Schröder	Peter	Theodor-Fahr-Straße 32	2000	Hamburg 62	040 /5311582 // -
Schut	Andre	Sanderstraße 26	1000	Berlin 44	030 / 6917861 // -
Schäfer	Walter	Rathausstr. 4	8160	Miesbach	08025 /1631 // 08025 /41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	5305	Alfter	0228 /643853 // -
Sopp	Arnulf	Wakenitzstr. 8	2400	Lübeck 1	0451 /791926 // -
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506	Hagen a.TW.	05401 /99585 // 05401 /30096
Stober	Reiner	Nelkenstraße 12	3216	Salzhemmendorf 4	05153 /1564 // -
Sörensen	Rüdiger	Thomas-Mann-Straße 3A	6500	Mainz 1	06131 /32860 // 06131 /395268
Tornow	Wilhelm	Görlitzer Straße 16	2190	Cuxhaven 13	04123 /1355 // 0431/362055
Volz	Oliver	Waldenburgstraße 73	7000	Stuttgart 80	0711 /7353817 // -
Waccus	Michel	Mühlhofweg 2a	CH-8266	Steckborn	054/612560 // -
Wacker	Fred	Weingärtenstraße 4	7554	Kuppenheim	07222 /48931 // -
Wucherer	Jürgen	Menzelstraße 1	7750	Konstanz	07531 /54686 // -
Wulf	Hans-Otto	Im BrahmKamp 38	4250	Bottrop	02041 /688972 // -

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

TEXT DATEI

DIE RICHTIGE INFO IST ALLES !

CLUBINFO

27. AUSGABE

KONTAKTADRESSE : CLUB 80 / ALEXANDER SCHMID / ST. CAJETAN STR. 38 V11 / 8000 MÜNCHEN 80
TEL.: 089 / 495326

- I N H A L T S V E R Z E I C H N I S -

Seite:
und Autor:

Clubinternes

Neues von der Prostata .... 01 - 02
Alexander Schmid
Clubtreffen - im Bilde .... 03 - 04
Heinrich Betz, Redaktion
Happy Birthday ! .... 05 - 06
Jutta Obermann
Termine 06
Redaktion
Vorstellungen .... 07 - 10
Andre, Andreas, Egbert, Oskar, Frank
Projekt: Modula-Bibliothek .... 11 - 12
Gerald Schröder

Software

Ein kleiner Fehler im Z80DOS 14
Alexander Schmid
Fraktale Geometrie .... 15 - 18
Egbert Schröder
Möglichkeit & Grenzen des MS-DOS .... 19 - 30
Ulrich Heidenreich
Wie funktioniert ein Vektorgenerator 31 - 38
Z80-Programmtricks in Assembler .... 39 - 40
Traumlandschaften aus dem Computer 41 - 47
Artikel aus Computer Persönlich

Seite:
und Autor:

Hardware

Die Parallelschnittstelle .... 49 - 50
Bewährtes DFü-Protokoll .... 51 - 54
Artikel aus Computer Persönlich

Börse

Wer hat was -- wer will was .... 55 - 58

Sonstiges

Die Mega-Renner von INTEL 48
Artikel aus ...
Welche Sprache für wen ? .... 59 - 60
Alexander Schmid

Die letzten Seiten

Impressum 61
Schluß 62
Redaktion
Mitgliederadressenliste .... am INFO-Ende
Alexander Schmid
CLUB-80 Bücherliste .... Sonder-INFO
CLUB-80 NEWDOS-Diskothek .... Sonder-INFO
POWER .... Sonder-INFO

Neues von der Prostata !

Unser ehrenwerter erster Vorsitzender hat zwar nicht den Löffel, aber zumindest den Vorsitz abgegeben. Trotz heftiger Gegenwehr mußte ich mich schließlich der erdrückenden Mehrheit beugen und so hat man mich auf dem diesjährigen Clubtreffen (vom 28. April - 1. Mai) zur neuen Vorstandsdrüse des Vereins ernannt. Ihr hättet es verhindern können, warum wart Ihr nicht da ?

Wenn man mir die Anwesenheitsliste nicht entführt hätte, könnte ich Euch genau sagen, wieviele Leutchen da waren, aber die Vorstellung war sehr gut besucht. Bei der Sitzung selber waren es 21, dazu kommen noch die, die nur vorher oder danach da waren und, nicht zu vergessen, die Boxenmannschaft, sprich Frauen und Freundinnen. Neben den obligatorischen Tandies und Genies waren natürlich auch wieder einige Kriegsgefangene und Exoten aus dem Atari- und MS-DOS-Lager dabei. Nachdem jede verfügbare horizontale Fläche mit Rechnern, Monitoren oder Handbüchern belegt war wurde es langsam eng, was dem Erfahrungs- und sonstigem Austausch aber keinen Abbruch tat, eher im Gegenteil. Außer der Hackerei, die nach Augenzeugenberichten teilweise bis nach 2 Uhr nachts ging, gab es selbstverständlich auch den offiziellen Teil, über den ich noch etwas mehr sagen möchte.

Im Vorstand hat sich einiges getan, die meisten Posten sind neu besetzt worden, obwohl ich nicht finde, daß die 'Alten' das so schlecht gemacht haben. Von dieser Seite nochmals Danke für die geleistete Arbeit und mögen sie ihren verdienten Ruhestand genießen. Die beiden einzigen Altgedienten sind Jens Neueder, der auch in diesem Jahr wieder als Info-Redakteur tätig werden wird und Jutta Obermann. Jutta hat sich dankenderweise wieder für die Kasse geopfert, die sie im vergangenen Jahr mit wahrer Todesverachtung vor dem Bankrott bewahrt hat. Neuer erster Vorsitzender bin also nun ich, Alexander der Schmid (nein, nicht der Große), zweiter Vorsitzender ist Horst-Dieter Schroers. Neuer Hardware-Koordinator ist Andreas Magnus, der mit seinem Genie IIIs in einem riesigen IBM-Towergehäuse und mit Festplatte für Aufsehen gesorgt hat und die Buch-Bibliothek hat Christian Menk übernommen. Für die Newdos-Programme ist jetzt Oliver Volz zuständig und last, but not least, hat Rüdiger Sörensen die CP/M-Bibliothek vereinnahmt.

Unter anderem wurde auch zum wiederholten mal über MSDOS 'ja' oder 'nein' abgestimmt und glücklicherweise (meine Meinung) wieder abgelehnt. Die Gründe waren im großen und ganzen die, daß wir schon mit unserem eigenen TRS-80, CP/M und sonstigem Z80-Kram genug zu tun haben und mit einem 'echten' MSDOS-Club sowieso nicht konkurrieren könnten. Auch wäre der Aufwand für die Verwaltung der unzähligen PD-Programme viel zu groß. Wir bleiben also ein Z80-kompatibler Club mit bevorzugter Richtung TRS-80. Rüdiger Sörensen hat daraufhin als Alternative angeboten, daß er fast alles, was in der IBM-Welt an Public Domain rumläuft auch besorgen kann. Damit war dieses Thema vom Tisch und wer also ein spezielles (PD !!!) Programm braucht, wende sich bitte an ihn. Außerdem wird niemand standrechtlich erschossen, wenn er mal einen Artikel über Atari oder IBM schreibt. Als Information ist sowas immer herzlich willkommen, wenn es auch nicht ausdrücklich unterstützt wird.

Zur Unterstützung unserer Mitglieder in der DDR wurde beschlossen, daß der Club, wenn jemand seinen alten Compi gerne abgeben möchte und diesen in Richtung Ost verschifft, entweder die Frachtkosten übernimmt, oder selber versucht, das zu organisieren. Bevor er verstaubt und schließlich einfach weggeworfen wird, ist das wohl die eindeutig bessere Lösung. Bei der Gelegenheit kann ich Euch auch gleich ein Buch über Tabellenkalkulation ans Herz legen. Da Frank & Co. verständlicherweise nicht als Bittsteller dastehen möchten, könnten Sie uns interessante Computerbücher aus der DDR besorgen, die dann hier im Club verkauft werden. Von dem Geld können wir ihnen dann Disketten und sonstiges Material schicken, die dort gar nicht oder nur zu extrem hohen Preisen zu bekommen sind. Das Buch heißt 'Tabellenkalkulation mit Personalcomputern' und hat 144 Seiten im A5-Format. Darin werden zwar speziell Programme namens KP und MULTIMAC beschrieben, aber beim Durchblättern haben die mich doch irgendwie an Multiplan o.ä. erinnert. So groß werden die Unterschiede wohl kaum sein und zum ersten Kennenlernen von Spreadsheets dürfte es gar nicht schlecht sein. Wer also was abzugeben hat oder das Buch möchte, soll sich bitte melden. Der Preis richtet sich nach dem, was Ihr geben wollt oder könnt. Neben den unverzagten TRS-Usern gibt es auch noch ein paar C64er, die sich um Gerhard Gruber geschart haben. Wer was für die hat, soll sich bitte auch melden.

Ich hoffe, daß das mit dem Info jetzt besser klappt. Wir haben diesmal mehrere Optionen, wo wir Drucken lassen können und wenn alles schiefgeht, wird doch ein eigener Kopierer angeschafft. Wenn es jetzt nicht hinhaut, dann wohl nie.

Was vielleicht auch noch wichtig ist, ist, daß jetzt endlich mit der Eintragung ins Vereinsregister ernst gemacht werden soll. Das war auch mit einer der Gründe, warum Horst-Dieter zweiter Vorsitzender geworden ist. Um den ganzen notariellen Krempel zu erledigen müssen die beiden Vorsitzenden nämlich zusammen zum Notar gehen und das wäre sonst etwas schwierig geworden.

So, Ihr könnt wieder Luft holen. Das wäre soweit erstmal alles, was ich zum Einstand zu sagen hätte. Schreibt weiter fleißig Eure Artikel und schickt mir bitte noch die letzten Fragebögen zurück, die Ihr irgendwo rumliegen habt.

Meine Anschrift steht zwar im Anhang, aber hier nochmal groß und deutlich wie der Briefkopf aussehen könnte:

An den
hochwohlgeborenen 1. Vorsitzenden
des ehrenwerten Club-80

Alexander Schmid
St. Cajetan Str. 38/VII
8000 München 80

Wenn es sein muß, würden auch die letzten drei Zeilen reichen, der Brief müßte dann rein theoretisch auch noch ankommen.

Ich kann nur hoffen, daß der Club mich überlebt und daß Ihr alle mit mehr oder weniger Begeisterung dabei bleibt. Wenn mir mein Knuddel-Device genug Zeit läßt, verspreche ich, daß ich mein Möglichstes zum Gelingen der Clubarbeit beitragen und immer als Briefkastenonkel zur Verfügung stehen werde.

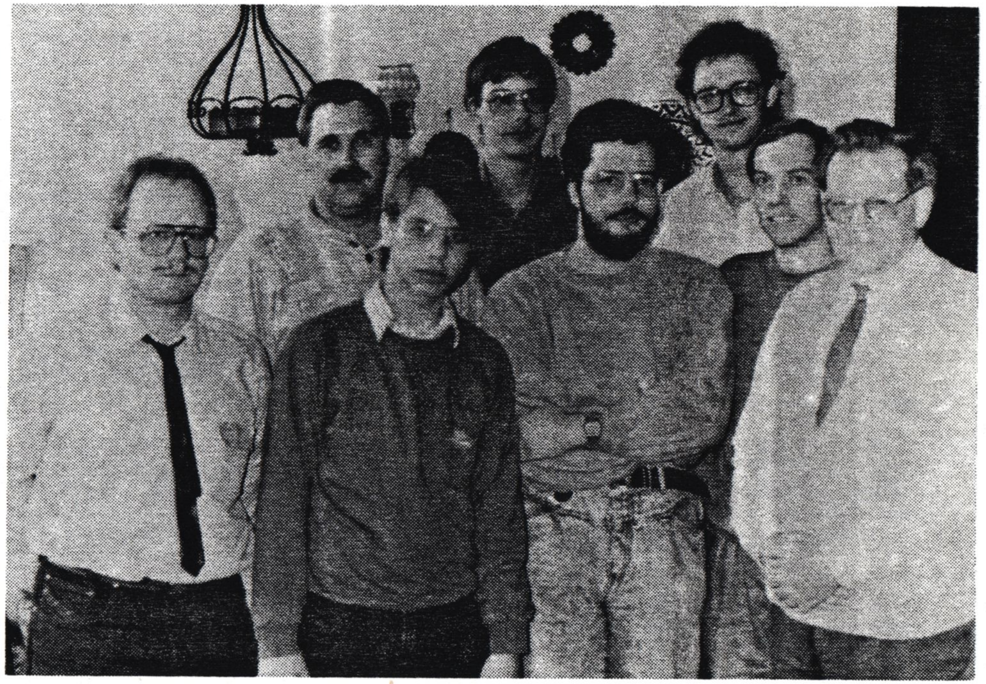
Viele Grüße, frohes Schaffen, happy Hacking wünscht Euch

Alexander

03

04

Die Mitgliederversammlung mit 21
zur Zeit anwesenden Clubmitgliedern
tagt in den Räumen des Idsteiner Hofes.



Der neue CLUB 80 Vorstand:
(von links nach rechts)
Hartmut Übermann........Kassenwart (Vertretung für Jutta)
Jens Neueder............Redaktion
Christian Menk..........Club 80 Bücherei
Andreas Magnus..........Hardware-Koordinator
Alexander Schmid........1. Vorsitzender
Rüdiger Sörensen........CP/M-Diskothekar
Oliver Volz.............NEWDOS-Diskothekar
Horst-Dieter Schroers...2. Vorsitzender

==============================
HAPPY BIRTHDAY !!
==============================

Hallo Ihr "Clubfreunde",

in der Hoffnung, daß Ihr das Clubtreffen gut überstanden habt, und nach Überwindung einiger interner Schwierigkeiten, können wir wieder mit Volldampf in das Clubleben einsteigen.

Da es immer noch nicht sicher ist, wann dieses Clubinfo er - scheint, beginne ich einfach ganz frech mit dem Monat Juni bei den Geburtstagsgrüßen. Denn es wird Euch bestimmt momentan nicht interessieren, wer in den Monaten Februar bis Mai Geburtstag hat, wenn das Info erst im Juni kommt. Den davon betroffenen Clubmitgliedern sei gesagt, daß Ihr bestimmt nächstes Jahr nicht übergangen werdet.

Wie jedes Mal, lege ich Euch ans Herz, mir zu schreiben, wenn Ihr Euch übergangen fühlt oder das Datum nicht stimmt.

Noch eine Bitte an dieser Stelle. Unterstützt Jens bei der Clubarbeit so gut es geht. Scheut Euch nicht, Beiträge an ihn zu schikken. Ihr wißt ja, nur durch das Clubinfo lebt der Club. Deswegen sollte jeder sein Quentchen dazu beitragen.

Aber nun folgen die genauen Daten.

Im Monat Juni haben Geburtstag:

01.	:	Wolfgang Beckhausen
10.	:	Claus Ruschinski
14.	:	Christian Menk
15.	:	Harald Mand
19.	:	Hans-Otto Wulf
20.	:	Eckehard Kuhn
28.	:	Matthias Homann

Im Monat Juli folgen:

07.	:	Armin Ahlers
10.	:	Frank-Michael Schober
18.	:	Hans-Günther Hartmann
19.	:	Hermann Brandl
30.	:	Bernd Retzlaff

Im Monat August gratulieren wir:

02.	:	Günter Issig
05.	:	Hans-Martin Stephan
		Ulrich Heidenreich
10.	:	Helmut Bernhardt
13.	:	Klauspeter Schmidt
15.	:	Waldemar Misioch
16.	:	Harald Braun
24.	:	Hartmut Obermann

Im Namen des Vorstandes wünsche ich allen genannten und ungenannten Clubfreunde ein fröhliches Feiern und alles Gute auch weiterhin.

Jutta

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß ....ca. 5 Wochen
nach Erscheinen dieses INFO's
Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!

-- Messen '89 --

Internationale Funkausstellung ....Berlin 25. August - 9. September 1989
PRODUKTRONIA ....München 7. - 11. November 1989

Hallo Leute,

ganz verschämt möchte ich mich als (nicht mehr ganz so) neues Clubmitglied endlich (nach jetzt über 8 Monaten) vorstellen. Zu einem ganz kleinen bißchen Entschuldigung möchte ich erwähnen, daß ich von dieser Zeit ca. ein halbes Jahr nicht zu Hause war und mir jetzt mein Genie III nach über 2 Monaten der (Wieder-) Eingewöhnung noch immer recht fremd vorkommt.

Mein Name ist André Schut, geb.am 29.3.1951 und ich wohne in Berlin. Ich kam beruflich (allerdings nur als "Anwender") vor ca. 5 Jahren zu meinem ersten Kontakt mit einem Genie III. Vor ziemlich genau 3 Jahren habe ich mir dann das gleiche Teil gebraucht zugelegt und habe zu meiner Schande aber noch immer nicht viel mehr als ein "fundiertes Halbwissen" über die Hardware und gerade mal "Kentnisse" von Basic (weia!) und Cobol. Anfänge von Pascal, die ich vor meinem "Langzeiturlaub" mal ansatzweise in Angriff genommen habe, sind total verschüttet.

Mein Interesse ist 1. wieder "reinzukommen" und 2. mal etwas mehr von Hardware (sowie Elektronik überhaupt) mitzubekommen. Ich würde mich freuen, wenn unter den Hardwarebastlern ein Genie III-Besitzer wäre, der bereit wäre, mir etwas "auf die Sprünge" zu helfen.

Auf hoffentlich bis bald

Hallo Clubfreunde,
Ich möchte mich hier kurz als neues Mitglied vorstellen.

Ich heiße Andreas Magnus, bin auch geboren und zwar am 17.10.1965 in Gelsenkirchen wo ich auch wohne.
Von Beruf bin ich Fernmeldeelektroniker welchen ich aber z.Z. nicht ausübe da ich bei der Bundeswehr bin
Ich befasse mich schon einige Jahre mit Computern und habe einen Genie II sowie einen Genie IIIs
Ich habe immer Wissensdurst, helfe aber auch gerne bei Fragen, speziell zu Hardwareproblemen.
Auf gutes Gelingen.

Hallo -80er Fan´s,

als neues Clubmitglied möchte ich mich kurz vorstellen.

Mein Name ist Egbert Schröer, geb. am 31.05.58, demzufolge (noch) 30 Jahre alt. Verheiratet und wohnhaft in Dorsten, einer Stadt am Rande des Münsterlandes.
Beruflich nutze ich einen HP-Vectra ES/12 Model 41 und einen IBM XT.
Angefangen hat alles mit einem SHARP PC 1251 (nur für Leute mit kleinen Fingern) vor ca 8 Jahren.

Mein TRS 80/Model 1 48K mit Single Density Controler (leider) ist mit Kleinschreibung und HRG1B der Firma RB Elektronik ausgerüstet und läuft unter NEWDOS80 Vers. 2.057. Zwei 40 Spur Floppy´s, ein LW SS das andere DS (leider zur Zeit noch SS angeschloßen), vervollständigen das System.

Auf gute Zusammenarbeit
Egbert Schröer

Hallo Clubfreunde,

ich möchte mich als neues Clubmitglied kurz vorstellen.

Mein **Name** ist Oskar Drechsler und wohne in Bergisch Gladbach. Ich bin 42 Jahre alt, habe Holzkaufmann gelernt, bin anschließend Soldat geworden und geblieben und in der Personalverwaltung tätig.

Ich besitze einen Genie I mit 2 Laufwerken (40 Track SS/DD und 80 Track DS/DD). Den Rechner und die Laufwerke habe ich in ein Alu-Gehäuse eingebaut, mit der HRG 1b-Platine erweitert und das Netzteil gegen ein Schaltnetzteil ausgetauscht. Die Tastatur habe ich gegen eine Tastatur mit Umlauten und 22 Funktionstasten ausgetauscht. Sie liegt vor dem Rechner. Einige Funktionstasten habe ich schon mit Druckersteuerzeichen belegt, für den Rest fehlen mir noch die Ideen. Außerdem habe ich den Rechner an einen ECB-Bus angeschlossen, der zur Zeit nur mit einer Uhr bestückt ist. Ein Tongenerator ist in Vorbereitung.

Als Betriebssystem benutze ich NEWDOS 80 V2.052, habe bisher überwiegend in Basic programmiert und glaube darin ziemlich fit zu sein. Seit einem Jahr programmiere ich auch in Assembler. So habe ich für meine neue Tastatur einen Treiber geschrieben und den von Frank Debatin geschriebenen HRG-Graflix-Editor mit Maschinenroutinen ergänzt. Ich werde das Programm demnächst vorstellen.

Durch meine Clubmitgliedschaft erhoffe ich mir Kontakte zu Hardware-Freaks, weil ich die Speicherkapazität erweitern und die Taktgeschwindigkeit erhöhen möchte. Außerdem bin ich an Kontakten zu Assembler-Programmierern interessiert.

Auf gute Zusammenarbeit

Oskar

Hallo Club 80 !
Als neues "überregionales" Klubmitglied aus der DDR möchte ich mich an dieser Stelle vorstellen.
Mein Name ist Frank-Michael Schober, ich wohne in Spremberg/Lausitz, in der Nähe von Cottbus und bin 37 Jahre alt. Verheiratet bin ich mit dem Computer, ansonsten nicht. Meine Brötchen verdiene ich als Lehrmeister in der Nachrichtentechnik. Dort lasse ich auch meine Nerven.
Angefangen hat das Drame bei mir 1982 mit einem ZX 81-Bausatz. Nachdem ich dann über ZX-Spectrum zu einem ALPHATRONIC PC8 umgestiegen bin, hat es mir die große, weite CP/M-Welt angetan.
Ich bin ein leidenschaftlicher Byte-Wurschtler (Maschinencode-Fan), daher meine Vorliebe für systemnahe Programmierung.
Ich hoffe trotz der Entfernung auf eine gute Zusammenarbeit.

Mit CP/M Kompatiblem Gruß

Projekt: Erstellung/Dokumentation von Modula-Bibliotheken

Ziel : Voll dokumentierte Modula-Bibliotheken
Betriebssystem: CP/M-80
Sprache : Turbo-Modula

vorhanden:
- Turbo-Modula unter CP/M
- einige Turbo-Modula-Bibliotheken mit englischer Dokumentation
- viele Turbo-Pascal-Bibliotheken mit CHIP-Dokumentation

Anforderungen an die Teilnehmer:
- Kenntnisse von Turbo-Pascal (oder Turbo-Modula)
- Bereitschaft, Turbo-Modula zu lernen

Motivation:

Bei der Programmierung verschiedener Programme stellen sich oft ähnliche Probleme. Wenn jedes Mal "das Rad neu erfunden" werden muß, dauert die Programm-Entwicklung zu lange und wird mühselig. Deshalb werden zunehmend Bibliotheken fertiger Programmstücke angeboten, die in eigene Programme integriert werden können und Standard-Aufgaben wie die Eingabe über Menüs, die Ausgabe auf Windows, das Sortieren von Arrays oder Files usw. übernehmen.

Unter CP/M gibt es viele Programmiersprachen, aber nur wenige stark verbreitete. Die bekannteste dürfte Turbo-Pascal sein, für die viele Programme und einige Bibliotheken vorliegen, letztere vor allem aus der Reihe "CHIP-Special".

Leider bietet Turbo-Pascal keine günstigen Voraussetzungen für die Arbeit mit Bibliotheken: Wenn eine Bibliothek ein File ist, muß in das Programm die ganze Bibliothek übernommen werden (per INCLUDE), auch wenn nur ein Teil benutzt wird. Oder der Programmierer muß sich mühselig die benötigten Teile herauskopieren, wobei er evtl. Hilfsprozeduren oder Variablen-Deklarationen übersieht oder zuviel übernimmt. Eine vorherige Syntaxprüfung der Bibliothek ist nur auf Umwegen möglich, die Schnittstellen zwischen Bibliothek und Anwender-Programm können nicht explizit festgelegt und geprüft werden. Der Bibliotheks-File darf seinerseites nicht auf andere Bibliotheken zurückgreifen, denn es sind keine verschachtelten INCLUDEs erlaubt. Somit können Programme nicht hierarchisch aufgebaut werden.

Deshalb scheint die Arbeit mit den Bibliotheken unter Turbo-Pascal nicht so verbreitet zu sein. Die Sprache ist einfach nicht dafür entworfen worden. Außerdem sind die Bibliotheken nicht standardisiert und oft nicht ausreichend dokumentiert. Oder sie liegen nur in MS-DOS-Versionen vor.

Die Lösung scheint mir deshalb in Turbo-Modula zu liegen, das leider nicht so stark verbreitet ist. Es bietet ähnliche Möglichkeiten wie Turbo-Pascal und noch einige dazu. Turbo-Pascal-Programme und -Bibliotheken können fast unverändert übernommen werden und der Programmierer braucht keine neue Sprache zu lernen, denn Modula ist eine Weiterentwicklung von Pascal. Außerdem sind alle Programmteile des Modula-Systems fast genauso wie das Turbo-Pascal-System zu bedienen.

Welchen Vorteil bietet nun Modula? Die Sprache wurde für die intensive Nutzung von Bibliotheken entworfen, die hier "Module" genannt werden. Die Verwaltung der Module samt Syntaxcheck und Überprüfung der Schnittstellen übernimmt das Modula-System. Ein Modul hat immer eine klare Schnittstelle (Definitionsteil), alles andere (also die Implementation) bleibt dem Benutzer verborgen. Ein Modul kann wieder auf andere Module zurückgreifen, was eine Hierarchisierung der Programmteile ermöglicht. Außerdem kann ein Linker herausfinden, welche Teile einer Bibliothek wirklich gebraucht werden und diese herausfiltern; leider nutzt Turbo-Modula diese Möglichkeit nicht aus.

Standardmäßig sind Modula einige Module beigegeben, die vor allem zur (einfachen) Ein-/Ausgabe und zur Speicherverwaltung dienen. Der Benutzer kann nun eigene Module entwerfen (z.B. zur Daten-Verwaltung) und diese dann in seinen Programmen benutzen. Diesen Entwurf, teilweise anhand vorhandener Turbo-Pascal-Bibliotheken, wollen wir in dem Projekt leisten. Außerdem sollen vorhandene und neu erstellte Module einheitlich (neu) dokumentiert werden.

Das Ergebnis werden (hoffentlich) viele gut dokumentierte Module sein, die dann jeder in seinen Programmen einsetzen kann. Dadurch sollte sich später die Programmerstellung vereinfachen.

Auszuführende Arbeiten:

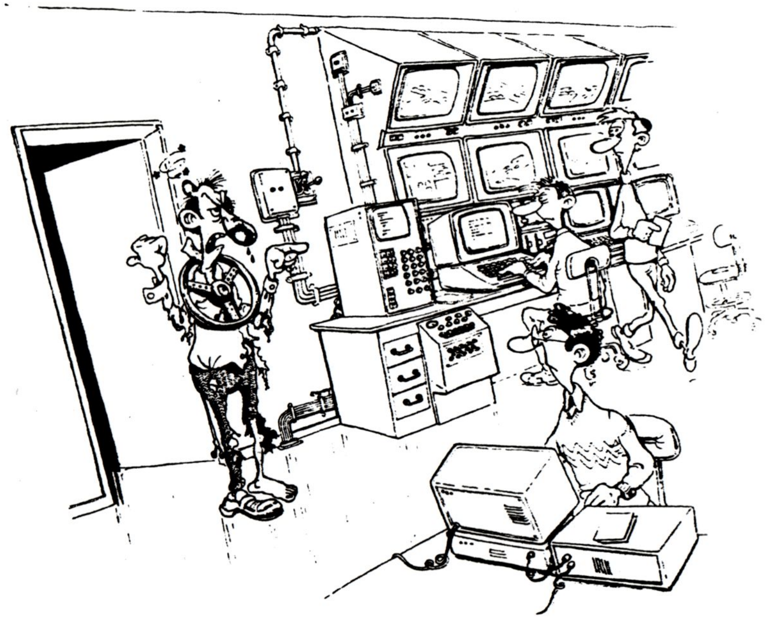
Die vorliegenden Turbo-Modula-Bibliotheken sollten neu dokumentiert werden, denn sie sind bisher nur in Englisch in der Turbo-Modula-Anleitung beschrieben worden. Die Beschreibungen sollten eine einheitliche Form bekommen, damit der Benutzer nicht bei jedem Modul lange nach benötigten Informationen suchen muß. Es geht hier nicht um eine Dokumentation des Source-Codes (der nicht vorliegt), sondern um eine Beschreibung der Schnittstellen und der Benutzung der Module. Dies könnte auch von Anfängern ohne Turbo-Modula-Erfahrung geleistet werden, soweit sie mit der englischen Anleitung zurechtkommen.

Aus der Reihe "CHIP-Special" liegen eine Menge Bibliotheken und Programme für Turbo-Pascal vor. Diese müßten gesichtet und für Turbo-Modula umgeschrieben werden. Teilweise existieren mehrere Versionen einer Bibliothek (z.B. Masken-Verwaltung), so daß eine Zusammenfassung bzw. Auswahl nötig wird. Für diesen Teil werden Programmierer mit Kenntnissen in Turbo-Pascal und möglichst auch in Turbo-Modula benötigt. Wenn Turbo-Modula nicht bekannt ist, müßte der Interessent sich einarbeiten (ausführliche englische Anleitung liegt vor).

Die fertiggestellten Module sollten wie die vorliegenden Module dokumentiert werden, wofür nur Turbo-Modula-Kenntnisse benötigt werden (auch für Anfänger). Außerdem wäre es schön, wenn zu den Modulen Beispielprogramme erstellt werden, um die Benutzung der Module zu demonstrieren.

Wenn aus dem Kreis der Interessenten noch Vorschläge für eigene Bibliotheken (Module) geäußert werden, können wir auch selbst noch welche entwerfen. Vorschläge (auch von Außenstehenden) werden immer gerne angenommen. Außerdem sei jeder Leser aufgerufen, seine eigenen Bibliotheken (seien sie in Turbo-Pascal oder Modula geschrieben) zur Verfügung zu stellen.

Interessenten melden sich bitte bei:
Gerald Schröder
oder Alexander Schmid



Welcher Idiot ist für die Ampeln am Kudamm verantwortlich!?

Ein kleiner Fehler im Z80DOS

Wie bei den CP/M-Benutzern hoffentlich allgemein bekannt, kann man dort die Bildschirmausgabe durch einen Druck auf Control-P auf den Drucker und wieder zurück, nur auf den Bildschirn, umleiten. Normalerweise sollte man davon nichts auf dem Bildschirm sehen, nur der Drucker sollte mitlaufen. Als ich jetzt das antike BDOS durch das Z80DOS ersetzt hatte, erzeugte das Cntl-P plötzlich seltsame Zeichen, die da gar nicht hingehörten. Als übeltäter stellte sich schließlich die Abfrageroutine im Z80DOS heraus. Der Autor hatte einen kleinen Sprung vergessen und so wurde fälschlicherweise in der Ausgaberoutine weitergemacht, statt nur ein Flag zu setzen und stillschweigend auf das nächste Zeichen zu warten. Im Source-File sieht das so aus:

```
; Read buffer
;
rdbuf:  ld      a,(tabcnt)          ; Get current position cursor
        ld      (tabcx1),a          ; Save it
rdbuf0: push    ix                  ; Save start address buffer
        pop     hl                  ; Get it in hl
        ld      c,(hl)              ; Get maximum line lenght
        inc     hl                  ; Increment to line lenght position
        ld      b,0                 ; Clear line lenght counter
        push    hl                  ; Save start line - 1
rdbuf1: push    hl                  ; Save registers
        push    bc
rdbuf2: call    getch               ; Get character
        pop     bc                  ; Restore registers
        pop     hl
        and     07fh                ; Mask character
rdbuf3: cp      ContH               ; Test backspace
        jr      nz,rdbuf4           ; Not then jump
doback: ld      a,b                 ; Test if deleting char from empty line
        or      a
        jr      z,rdbuf1            ; Yes then get next char
        pop     hl                  ; Get start line
        push    hl                  ; And save it again
        call    delch               ; Delete character
        jr      rdbuf1              ; Get next character
rdbuf4: cp      ContP               ; Test print enable/disable
        jr      nz,rdbufC           ; Not then jump
        ld      a,(fContP)          ; Complement print flag
        cpl
        ld      (fContP),a
        jr      rdbuf1              ;  Get next character      <= hier
rdbufc: cp      ContX               ; Test delete line            passierts
        jr      nz,rdbufe           ; Not then jump
rdbufd: pop     hl                  ; Get start line
        ld      a,b                 ; Test if last character deleted
        or      a
        jp      z,RDBUF             ; Yes start routine again
        push    hl                  ; Save pointer
        call    delch               ; Delete last character line
        jr      rdbufd              ; Test last character deleted
```

Wenn jemand es selber benutzt, oder er einen kennt, oder er einen kennt, der einen kennt..., bitte weitersagen. Kleine Ursache, mittelgroße Wirkung.

Alexander Schmid

Fraktal Generatoren für den TRS80 M 1 mit HRG 1B

Teil 1

Spricht man mit einem Computer Freak über Fraktale, so denkt dieser sofort an das bekannte Apfelmännchen.
Entdecker dieser Spezies ist Benoit B. Mandelbrot.
Das Apfelmännchen, oder auch Mandelbrot-Menge, ist aber nur eines von vielen Fraktalen, die man auch mit einem Fraktal Generator auf dem TRS 80 darstellen kann.
Das hier vorgestellte Programm in BASIC erzeugt eine vielzahl von Fraktalen in akzeptabler Zeit.

Bevor es an das Programmieren geht, einiges zur Theorie:

Fraktale (lat. ´fractus´= gebrochen) sind irreguläre, aber recht hübsch anzusehende Gebilde, die eine gemeinsame Eigenschaft haben: Ausschnitte dieser Gebilde sind sich selbstähnlich und Teile dieses Ausschnittes wiederum dem Ausschnitt selbstähnlich (toll).

Diesem geometrischen Prinzip folgen auch natürliche Gebilde, wie z.B. Küstenlandschaften oder Baumrinden um nur zwei zu nennen.
Theoretisch ist dies ein unendlicher Kaskadenprozeß, praktisch gibt es jedoch einen äußeren Abschnitt (Cutoff), der die Größe der Figur bestimmt, sowie einen inneren Cutoff, der theoretisch dann auftritt, wenn man die Größe eines Atoms erreicht. Das läßt sich graphisch naürlich nicht mehr darstellen. Wenn man dem Fraktal Generator Parameter angibt, die graphisch nicht mehr darstellbar sind, führt dies zu einer Fehlermeldung.

Wie sieht nun ein solcher Fraktal-Generator aus.
Man benötigt zunächst einen Initiator, im Programm festgelegt durch ´sides ?´, dann folgt ein ´Wachstumsgenerator´, der den Wachstumsprozeß festlegt, festgelegt durch ´Level ?´.
Der Generator wird dabei so skaliert und gedreht, daß die Endpunkte mit den Endpunkten der Initiator-Strecke zusammenfallen.

Das BASIC-Programm erzeugt Fraktale wie z.B. die Koch´sche Schneeflocke, Drachenflächen und die Serpinsky-Pfeilspitze.
Bei der Option ´Inverse ?´ erfolgt der Wachstumsprozeß nach innen, begrenzt durch den kleinsten darstellbaren inneren Cutoff
Wird dieser überschritten, erfolgt eine Fehlermeldung.
So, genug geschrieben nun folgt das Programm:

```
10 ´*************************************************************
12 ´*********         Creating Fractals           **************
14 ´*********   with High Resolution Graphik      **************
16 ´*************************************************************
18 ´*********       TRS-80  Computerstation       **************
20 ´*********              Dorsten                **************
22 ´*********       Copyright by Egbert Schröer   **************
24 ´*********         Januar 1988     Vers. 1.3   **************
26 ´*************************************************************
28 ´*********          Original Version by        **************
30 ´*********     Michiel van de Panne Dec.1984   **************
32 ´*********        80 MICRO  Dec.´84 side 114   **************
34 ´*************************************************************
100 ´
150 ´***   All remarks can be left out   ***
160 ´
180 #CLS: #OPEN
200 CLEAR 1000 : CLS
300 CF= 3.1416/180
400 ON ERROR GOTO 9600
500 ´
600 ´*** Define point to start drawing ***
700 ´
800 CX%=241: CY%=154: X2=CX%: Y2=199-CY%
900 INPUT"Number of sides ";S%
1000 INPUT"Inverse";I$: I$=LEFT$(I$,1)
1100 ´
1200 ´*** Define numbers of degrees for right and left turn ***
1300 ´
1400 R=-360/S%: L=R+180
1500 IF I$="Y" THEN C$="A": W=R: R=-L: L=-W ELSE C$="R"
1600 A$=C$
1700 IF I$<>"Y"  THEN 2000
1800 AL$="R": S1%=S%-2: FOR N%=1 TO S1%: AL$=AL$+"L": NEXT:
     AL$=AL$+"R"
1900 GOTO 2200
2000 AL$="L": S1%=S%-2: FOR N%=1 TO S1%: AL$=AL$+"R": NEXT
2100 AL$=AL$+"L"
2200 INPUT"Number of levels ";LE%:IF LE%=1 THEN GOTO 3300
2300 ´
2400 ´*** Build string containing right and left ***
2500 ´*** turns to be done.                      ***
2600 ´
2700 FOR N%=2 TO LE%: LN=LEN(A$)
2800 FOR N1%=1 TO LN: B$=B$+MID$(A$,N1%,1)+AL$: NEXT N1%
2900 A$=B$: B$="": NEXT N%
3000 ´
3100 ´*** Lines 125-130 choose length of line segments ***
3200 ´
3300 CLS: IF S%<5 THEN GOTO 3500
3400 LL=452/(S%*3^LE%): GOTO 4000
```

```
3500 LL=144/(3*LE%)
3600 '
3700 '*** Actual drawing loop begins here ***
3800 '
3900 #CLS:#OPEN
4000 FOR Z%=1 TO S%
4100 FOR N%=1 TO LEN(A$)
4200 B1$=MID$(A$,N%,1)+AL$
4300 FOR N1%=1 TO LEN(B1$):B$=MID$(B1$,N1%,1)
4400 IF B$="A" THEN A=A+W: GOTO 4600
4500 IF B$="L" THEN A=A+L ELSE A=A+R
4600 AL=A*CF
4700 X=X+LL*COS(AL): Y=Y+LL*SIN(AL)
4800 XP=INT(X): YP=INT(Y): X1=XP+CX%: Y1=199-(YP/2+CY%)      :
     GOSUB 6400
4900 X2=X1: Y2=Y1
5000 NEXT N1%,N%,Z%
5100 '
5200 PRINT@1019,"Done";
5300 '
5400 '*** Press any key to continue ***
5500 '
5600 A$=INKEY$
5700 IF A$="" THEN 5600 ELSE GOSUB 8300
5800 END
5900 GOTO 5700
6000 '
6100 '*** Check to ensure that endpoints of ***
6200 '*** Line are on screen                ***
6300 '
6400 IF X1<0 OR X1>383 THEN RETURN
6500 IF X2<0 OR X2>383 THEN RETURN
6600 IF Y1<0 OR Y1>191 THEN RETURN
6700 IF Y2<0 OR Y2>191 THEN RETURN
6800 IF X1=X2 AND Y1=Y2 THEN #SET(X1,Y1): RETURN
6900 '
7000 A5=ABS(X2-X1): B5=ABS(Y2-Y1): IF A5>B5 THEN 7100 ELSE 7500
7100 IF X2>X1 THEN C5=X1: D5=X2: E5=Y1: F5=Y2: GOTO 7300
7200 C5=X2: D5=X1: E5=Y2: F5=Y1
7300 G5=D5-C5: H5=(F5-E5)/G5: FOR U5=C5 TO D5
7400 #SET(U5,E5):E5=E5+H5: NEXT: GOTO 7900
7500 IF Y2>Y1 THEN C5=Y1: D5=Y2: E5=X1: F5=X2: GOTO 7700
7600 C5=Y2: D5=Y1: E5=X2: F5=X1
7700 G5=D5-C5: H5=(F5-E5)/G5
7800 FOR U5=C5 TO D5: #SET(E5,U5): E5=E5+H5: NEXT
7900 RETURN
8000 '
8100 '*** Graphik store routine ***
8200 '
8300 #CLOSE
8400 CLS:PRINTCHR$(23)
8500 PRINT@8*64,"Graphik ---> Floppy Y/N ";                   :
     INPUT I$ : IF I$="N" OR I$="n" THEN CLS : RETURN
8600 PRINT@640,"Dateiname :   ............";                  :
     PRINT@668,"";                                             :
     LINE INPUT DAT$
8700 CD$="RENAME FRACTAL/HRG "+DAT$
8800 CLS:#OPEN
8900 #SAVE"FRACTAL/HRG"
9000 #CLOSE
9100 CMD CD$
9200 RETURN
9300 '
9400 '*** Error checking Routine ***
9500 '
9600 CLS: #CLS
9700 PRINTTAB(18)"E r r o r   M e s s a g e"
9800 PRINTTAB(15)STRING$(31,45)
9900 PRINT@4*64,"If Level II Error  : Errorcode ";ERR/2+1      :
     PRINT@6*64,"Line number        :";ERL
     PRINT@8*64,"Last DOS-Error     :";CMD"E"
10000 PRINT:PRINT"Program can't work on !"                     :
      PRINT"Please press <ENTER> to restart"
10100 A$=INKEY$: IF A$<>CHR$(13) THEN 10100 ELSE RUN
```

Alle REM's im Programm können entfernt werden. Anschließend sollte man das Programm mit ACCEL3 compilieren, damit man nicht eine Ewigkeit auf ein Fraktal warten muß.
Der Fraktal Generator ist unter NEWDOS80 und GDOS lauffähig.
Wer sich die Mühe des abtippens ersparen will, kann von mir eine compilierte Form anfordern.
Wie bereits im Titel angedroht folgt noch ein zweiter Teil, der sich mit einem Apfelmännchen Generator beschäftigt.

(c)1989   Egbert Schröer

Literatur:

Benoit B. Mandelbrot: The Fractal Geometry of Nature
W.H. Freeman and Company

Michiel van de Panne: Creating Fractals
80 MICRO, Dezember 1984, Seite 114 u.f.

U L R I C H   H E I D E N R E I C H

Werderstr. 35
4300 Essen 1
priv. 0201 / 282 770
dienstl. 0203 / 278 64

U.Heidenreich,43 Essen,Werderstr.35

CLUB 80
Herrn Gerald Schröder
Am Schützenplatz 14

2105 Seevetal 1

Ihr Schreiben/Zeichen 31.01.89 | Mein Schreiben/Zeichen | Essen, den 08.01.89

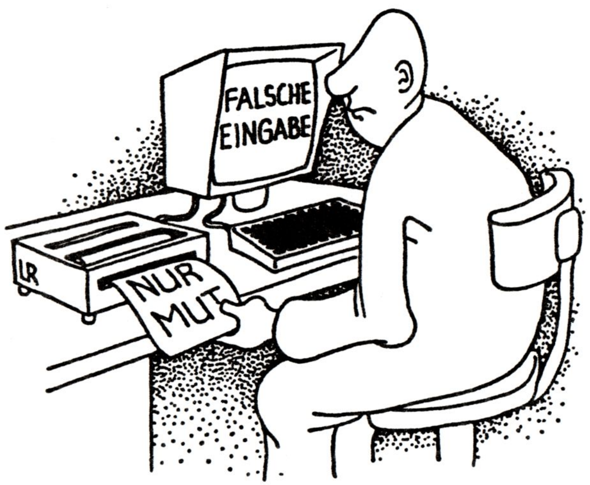
Lieber Gerald, Liebe Clubmitglieder,

meine Pflichten als Clubmitglied habe ich zugegebenermaßen fürchterlich vernachlässigt (vernachlässigen müssen ?). Leider bin ich inzwischen beruflich derart in Anspruch genommen, daß keine Zeit für den Club bleibt. Das ist nicht nur so dahingesagt: Ich war nämlich so dämlich (?), mein Hobby zum Beruf zu machen und habe demzufolge keine Zeit zum Hobby mehr. (Womit auch meine Anwesenheit beim Club-Treffen ausfallen muß, da mein Samstag nun zu den Arbeitstagen zählt!) Nichtsdestowenigertrotz hat mich das Einladungs-Schreiben dazu provoziert, zumindest meinen Senf zum Thema "'Club 80' jetzt auch gleich 'Club 8088' ?" hinzuzugeben:

Um die Allmacht der IBM-Nachbauten - ja gar nicht Mama Blue selbst! - herumzukommen, scheint auch unserm Club kaum möglich zu sein: Sei es, Einsteiger in die Computerwelt einzuführen (das macht heute niemand mit 'nem TRS80 mehr!), sei es aber auch dem zweiten Grundgedanken des Clubs zu folgen, KnowHow zu vermitteln. Unseren Betriebssystem- und Hardwareforschern wird das schier unermesslich undokumentierte Feld MSDOS / IBM auf Jahre hinaus interessanten Stoff bieten.

Auch um auf der 80er Linie zu bleiben, wäre ich der Meinung, die Aktivitäten statt in Richtung Atari, Amiga o.ä. (Nichts gegen die Geräte; Fettnäpfchen läßt grüßen: Ich verkaufe inzwischen sowas!) weiter in Richtung IBM auszudehnen. Wahrscheinlich bin ich nicht der einzige Club-80er, der vom Genie über GDOS/NEWDOS zum - würg! - CP/M-wechselte, dort statt BASIC (in dem wohl manch einer "seit Jahren nicht mehr programmiert") an (Turbo-) Pascal Spaß fand und über diese Sprache nahtlos zum MSDOS wechseln konnte.

Unsere Assembler-Freaks werden darüber hinaus schnell feststellen, daß 8088-Assembler gar nicht mal soooo vom vertrauten Z80 abweicht und sich auch in dieser Materie sofort wohlfühlen.



Es hat sich als sinnvoll erwiesen, wenn der Computer nicht intelligenter ist als der Anwender.

Wobei ich - mit etwas Melancholie! - "unserm" GDOS/NEWDOS keineswegs den Gnadenstoß verpassen möchte. Oft Ray Duncans Kommentare im "mc" zur MSDOS-Szene lesend, kommt mir der Aha-Effekt: Was der Mensch da leise ironisch bemängelt, könnte NEWDOS/GDOS - entsprechend dem Zug der Zeit folgend - allemale! (Was ist eigentlich aus Apparat Inc. geworden?)

Quasi als Produkt meines schlechten Gewissens dem Club gegenüber, als Diskussionsgrundlage für die Öffnung unseres Clubs in Richtung MS-DOS und als abschreckendes Beispiel, wie schlecht ein NEWDOS-Verwöhnter mit dem ach so modernen Dos zufrieden ist, möchte ich doch folgende Seiten für's Club-Info beisteuern.

MfG

Ulrich Heidenreich

Möglichkeiten und Grenzen des MS-DOS -

- Erfahrungsbericht eines Auf(?)steigers

Nach langem Sparen wanderte mit meinem TRS-80 ein NEWDOS/80 und ein noch älteres CP/M auf den Altenteil: Ein AT glänzt nun auf dem Schreibtisch. Die berühmten drei Buchstaben prangen zwar nicht auf dem Typenschild, dennoch gibt sich der Neue alle Mühe, dem großen Vorbild gerecht zu werden: Von der Tastatur - auf der Sie als Cherry-Verwöhnter dank des seitlichen Spiels der Tasten und der frühen Kontaktgabe oft genug mehrere Tasten gleichzeitig treffen - bis hin zur Bildschirmdarstellung steht er Big Blue in nichts nach. So wäre es auch wohl nicht zeitgemäß, den TTL-Monitor mit schmalsten Linien gestochen scharfe Schrift darstellen zu lassen, obwohl's er durchaus auf die Beine bringt. Sans Serif ist halt nur bei Druckern wieder in; auf dem Schirm darf den Augen zuliebe (?) ruhig gekleckst werden.

(Nebenbei: als glücklicher Besitzer einer EGA-Karte könnten Sie eine bessere Darstellung "stylen" . In Kürze: Interrupt 10H, AH=11H, ES:BP auf eigene Zeichenmatrix setzen! - wenn nicht bereits MS-DOS's CLS-Befehl den Original-Zeichensatz wieder aktivieren würde!)

UNIX inklusiv

Aber was soll's. Schließlich hat man sich das Gerät nicht deswegen auf den Schreibtisch gestellt, sondern liebäugelte mit anderen Vorteilen: Neben der obligatorischen Festplatte - endlich keine Assembler-Kaffeepause mehr! - verspricht MS-DOS Möglichkeiten, die CP/M damals noch durch Micro-Shell hinzugefügt werden mußten. Bei MS-DOS dagegen sind mit Piping, Input/Output-Redirecting und seinen Batch-Fähigkeiten diese Unix-Features inklusiv.

So ist MS-DOS in einer seiner neuesten Versionen wohl leistungsfähiger als ein altes CP/M 2.2 mit UNIX-Shell. Schon die einfache Möglichkeit, mehrere Text-Dateien auf den Bildschirm zu holen, begeistert: Kein XPIP, SWEEP oder sonst ein Utility ist nötig, einfach COPY *.TXT CON:! Dies schreibt sogar vor jeder so aufgelisteten Datei als Überschrift deren Namen auf den Bildschirm. Mannigfaltig und interessant sind die Wege, dasselbe auf den Drucker zu bekommen:

COPY *.TXT PRN: kopiert die Datei-Inhalte auf den Drucker, die Dateinamen landen jedoch weiter auf dem Bildschirm. Auch COPY *.TXT CON: >PRN: ist interessant: Jetzt gerät die Liste der Dateinamen auf den Drucker, die Datei-Inhalte erscheinen aber trotz Output-Redirection auf dem Bildschirm. Aber warum einfach, wenn's auch kompliziert geht: COPY *.TXT CON: | MORE bringt Seite für Seite auf dem Schirm, die dann per Taste "PrtScr" zum Drucker geschickt werden könnten. Leider erscheinen so erst alle Datei-Inhalte ungebremst auf dem Schirm, gefolgt von der Liste der Dateinamen. (Letztere, falls genügend lang, anhaltbar!)

Erst die Holzhammer-Methode, alles, was trotz COPY auf PRN: nicht bei PRN: ankommt, zusätzlich zu redirecten, wirkt Wunder:

```
COPY *.TXT PRN: > PRN:
```


Na ja, nach eingehendem Studium des guten DOS-Handbuchs hätte man ja auch sofort auf diese Lösung kommen können! Die Möglichkeiten des MS-DOS sind halt derart mannigfaltig, daß man nicht alles auf Anhieb sieht. So kann man Dank Filter MORE und der Prüfung mit EXIST doch tatsächlich zwei Inhaltsverzeichnisse vergleichen:

Mit DIRCOMP.BAT -

```
echo off
cd %1
echo Files missing in %2
for %%x in (*.*) do if not exist %2\%%x echo %%x
cd %2
echo Files missing in %1
for %%x in (*.*) do if not exist %1\%%x echo %%x
```

- sollten nach DIRCOMP \DIR1 \DIR2 | MORE alle Unterschiede - nämlich die im jeweils anderen Unterverzeichnis fehlenden Dateinamen - Seite für Seite auf dem Schirm gemeldet werden. Bis auf "Seite für Seite" klappt's auch; nur: die Ausgabe geschieht ungebremst. Was habe ich denn nun schon wieder bei der Beschreibung des MORE-Filters mißverstanden? Heißt "MORE" etwa, daß "mehr" als eine Seite ausgegeben wird?

Spanabhebende Datenverarbeitung

Weiter suchte ich - da das Arbeiten mit Disketten bekanntermaßen spanabhebende Datenverarbeitung darstellt - neulich nach einer Möglichkeit, auf meiner Sicherungs-Disk nicht immer wieder alle Dateien abzulegen, sondern nur die noch nicht gesicherten welchen. Mit MS-DOS's Batch-Fähigkeiten war dann auch schnell die entsprechende BATch-Datei kreiert:

```
ECHO OFF
FOR %%X IN (C:*.*) DO IF NOT EXIST A:%%X COPY C:%%X A:
```

Als Krönung dieser Schöpfung hing ich noch ein > NUL: hintenan. Eine Umleitung auf dieses Gerät läßt ja alle Ausgaben im Niemandsland verschwinden, statt sie dort auftauchen zu lassen, wo sie stören könnten - z.B. mitten in Eingabemasken. Mein Erstaunen, als dann doch eine "1 File(s) copied"-Meldung nach der anderen den Bildschirm verzierte, währte aber nicht lange: Seit MS-DOS 3.2 in meinem AT waltet, erledigt REPLACE C:*.* A: /A diese Aufgabe. Und REPLACE's Ausgaben lassen sich widerspruchslos NULlen.

BATch as BATch can!

Vielleicht lag's ja auch daran, daß Output-Redirecting aus .BAT-Dateien ein wenig anders funktioniert als der unbedarfte Anwender meint. Otto-Normalprogrammierer irrt halt, wenn er davon ausgeht, daß alle Ausgaben des Programms links vom ">" an das Gerät rechts desselben gelangen. Auch wenn in TEST.BAT TYPE AUTOEXEC.BAT steht, so heißt das noch lange nicht, daß nach TEST > PRN: eine Kopie der AUTOEXEC.BAT auf dem Drucker erscheint! Aus Stapeldateien heraus redirected MS-DOS nämlich wesentlich einfacher als mit dem schwerverständlichen ">"-Symbol:

Zunächst einmal definiert man je einen Eingabe- sowie Ausgabekanal Namens IDEV und ODEV durch Aufnahme der Zeilen

```
SET ODEV=CON:
SET IDEV=CON:
```

in die AUTOEXEC.BAT. Dann werden ebenso leicht und schnell in allen *.BAT-Dateien alle Ein-/Ausgaben, die Sie zu redirekten wünschen auf <%IDEV% und >%ODEV% umgeleitet. Nach diesen kleinen Vorbereitungen genügt - zum Beispiel - ein einfaches "SET ODEV=PRN:" für mühelose Ausgaben aus diesen Batch-Dateien an den Drucker!

Kanäle

Wem das nun doch nicht so einfach erscheint, sei daran erinnert, daß für solche Spielereien I/O-Redirecting ja auch nicht gedacht war. Die UNIX-Philosophie ist eher, hierdurch Kanäle zur Kommunikation zwischen Programmen bereitzustellen, welche selbst keine eigenen Mechanismen dafür besitzen. Ein äußerst treffendes Beispiel ist hier der bereits zitierte REPLACE-Befehl:

Weder REPLACE kennt einen Kanal, um seine Meldungen - zum Beispiel an ein laufendes BASIC-Programm - weiterzugeben, noch GW-BASIC einen solchen, um Fehlermeldungen eines DOS-Utilities - zum Beispiel REPLACE - einzulesen. I/O-Redirecting kann diesen Kanal nun aber zur Verfügung stellen:

```
10000 REM      Datensicherung
10010 :
10020 SHELL "REPLACE C:*.* A: /A > ERR_CHAN"
10030 OPEN "I",1,"ERR_CHAN"
10040 WHILE NOT EOF(1)
10050  INPUT #1,ERR.MSG$
10060 :
```

Was habe ich bloß falsch gemacht, daß - zum Beispiel bei voller Sicherungsdiskette - REPLACE's Fehlermeldungen trotz allem den Bildschirm ver(un-)zieren? MS-DOS's Piping-Philosophie ist wohl nicht so ganz UNIX-like. Oder liegt's etwa am Denkfehler, daß man zur Fehlermeldung eines Prozesses an den anderen ja auch keine Pipes nutzen darf (warum nicht?), sondern den DOS-Exit-Code - eingefleischten BATchern auch als Errorlevel wohlbekannt - auswertet? Hierüber lassen sich - geeignete Hochsprachen vorausgesetzt - Fehlermeldungen von Child-Prozessen einlesen; TURBO-Pascal 4.0 zählt zu den genannten:

Nach EXEC ('COMMAND.COM','/Ccopy A:DATEN C: > NUL:'); erschiene im Fehlerfalle zunächst die Meldung "A:DATEN not found" und "0 File(s) copied", die sich aber - o Wunder, warum funktioniert's denn jetzt? - in den NUL:-Kanal schicken läßt. Über die Pascal-Funktion DosExitCode kann man dann ja, ohne diese Meldung auf dem Schirm erscheinen zu lassen, die fehlerfreie oder -hafte Ausführung des COPY-Befehls feststellen:

```
If DosExitCode<>0 then Writeln ('Datei "DATEN" fehlerhaft!');
```


Wie sollte es anders sein: Auch hier stellen sich interessante, aber unbrauchbare Effekte ein: Unkritische (?) Fehler - wie zum Beispiel das Fehlen der Datei "DATEN" - erzeugen einen Exit-Code von 0, gaukeln also Fehlerfreiheit vor; bei ernsthafteren Fehlern - z.B. keine Diskette im Laufwerk A: - schlägt Abort, Retry, Ignore zu! (Zu diesem exzellenten Feature später mehr, jetzt erst:)

Nochmal Eingabe-Umleitung

Eingabe-Umleitung - das Holen von Daten statt von der Tastatur aus einer Datei - kann nämlich nicht nur die Tippfaulheit des Anwesenden befriedigen. Nein, auch automatisch ablaufende Prozesse unter Abwesendheit desselben können so bedient werden:

```
:
IF NOT EXIST A:*.* GOTO NODRIVE_A < IGNORE.CTL
COPY *.* A:
GOTO EXIT
:NODRIVE_A
MD C:\SAVECAT
COPY *.* C:\SAVECAT
:EXIT
:
```

In der Eingabe-Datei müssen halt nur die entsprechenden Tastendrücke stehen; IGNORE.CTL - Inhalt "I <cr>" - bedient in diesem Beispiel die Abfrage "Abort, Retry, Ignore ?"

Aber auch hier muß ich wohl die Handbuch-Notiz, das Symbol "<" bedeute, Eingaben für das Programm links desselben aus der Datei rechts desselben zu holen, mißverstanden haben! Trotz Input-Redirecting beharrte MS-DOS hier solange auf dem Tastendruck "I", bis ich am Montag morgen entsetzt feststellen mußte, daß meine automatische Datensicherung über das ganze Wochenende hinweg auf meinen dicken Daumen gewartet hat! Dafür bietet aber MS-DOS mit diesem Abort-Retry-Ignore-Feature Möglichkeiten, die den oft fälligen Warmstart per CTRL-C im alten CP/M bei weitem übertreffen:

Abort, Retry, Ignore über alles!

Kein vergessener {$I-}-Schalter plagt mehr den Turbo-PASCALler, kein BASIC-Programm bedarf mehr großartiger Error-Trap-Routinen, kein ERRORLEVEL braucht umständlich in BATch-Dateien abgefragt zu werden. COMMAND.COM läßt den Fehler - zum Beispiel eine offene Laufwerksklappe - erst gar nicht bis zur Hochsprache durch; Klappe zu und "R" gedrückt und die Sache ist gegessen. Auch ein versehentliches Ansprechen eines nicht vorhandenen Laufwerkes oder des "Off-Linen" Druckers ist kein Problem: Einfach "A"bort gewählt und schon ist das Versehen ungeschehen! Der Abort-Retry-Ignore-Text in der Datenerfassungs-Maske oder mitten in der CAD-Zeichnung - hier meist dekorativ als "Mixed Pixels" - stört ja eh niemanden; und wenn's wirklich mal vollautomatisch gehen soll, stellt man eben einen schweren Gegenstand auf die I-Taste!

Manche Anwenderprogramme nehmen einem den Abort zwar übel und stürzen ins DOS zurück, aber dafür hat man ja die Leerdatei R.COM, um wieder hineinzukommen. Ach nein, Sorry! Das funktionierte ja nur unter CP/M. MS-DOS kennt da ja etwas viel eleganteres: Rufen Sie COMMAND.COM mit dem Schalter /F, so wird kein Anwendungsprogramm mehr abgebrochen, sondern solche Fehler ohne Rückfrage - natürlich (?) dann auch mit Exit-Code 0 für "fehlerfrei"! - "I"gnoriert.

Aber beim Piping ...

Zurück zum Redirecting: CP/M's-WordStar können Sie mit Micro-Shell's Hilfe problemlos per Eingabeumleitung bedienen: Warum WS patchen, eine Steuerdatei des Inhalts "CTRL-OR78 <cr> CTRL-JHO" leistete damals gute Dienste; nur unter MS-DOS geht's so nicht mehr. Ob wohl MS-DOS beim "Redirecten" nicht in die "richtigen" I/O-Routinen eingreift? Dafür spricht dieser - zwar nicht unbedingt als Fehler anzusehende aber doch interessante - Effekt: Der GW-BASIC-Interpreter selbst (why not?) läßt sich nicht durch Piping/Redirecting bedienen, wohl aber das BASIC-Programm PIPETEST.BAS:

```
1000 INPUT A$
1010 PRINT A$
1020 END : REM Hier steht END und nicht SYSTEM!!!
```

Wenn Sie dieses Programm per ECHO Hallo ¦ BASIC PIPETEST mit Daten versorgen, funktioniert es zwar; der BASIC-Interpreter kehrt anschließend jedoch wieder ins DOS zurück. Seltsam, nicht wahr, wie ein Piping, das lediglich Daten in eine Pipe schreibt, den BASIC-Interpreter, der sich ebenso "lediglich" diese Daten aus der Pipe holt, derart beinflussen kann?

Da darüber hinaus Routing - also die Möglichkeit, Ausgabekanäle umzudefinieren (von den besch...eidenen Möglichkeiten des MODE LPT mal abgesehen) - MS-DOS völlig unbekannt ist, griff ich jüngst zur Assemblerlösung, um Ausgaben auf die RS232-Schnittstelle mangels geeigneter serieller Peripherie auf dem Schirm zu sichten:

```
        MOV  AX,2514      ; RS-323-Interrupt 14 via Funktion 25 auf
        MOV  DX,NEWINT    ; neue Routine "umbiegen":
        INT  21
        JMP  HERUM
NEWINT: MOV  DL,AL        ; Statt die RS-232 zu bedienen
        MOV  AH,2         ; Ausgabe auf Bildschirm schreiben
        INT  21
        XOR  AX,AX        ; Fehlerflags löschen
        IRET              ; 'raus ausm neuen INT 14
HERUM:  MOV  DX,HERUM     ; Routing resident
        INT  27           ; verlassen
```

"COPY IRGEND.WAS COM1:" brachte anschließend das Erfolgserlebnis, tatsächlich "IRGEND.WAS" auf dem Bildschirm bewundern zu können. Ein "TYPE IRGEND.WAS > COM1:" sprach dagegen wieder die RS-232-Schnittstelle statt des Schirms an. Es zeugt wohl von meinen schlechten Fähigkeiten als Systemprogrammierer, wenn ich mich (unter der Annahme, ein BIOS stelle hardwarespezifische Routinen zur Verfügung, welche vom DOS unter Unkenntnis der tatsächlichen Hardware über BIOS-Vektoren - und zwar nur über diese! - genutzt werden) verwundert frage: "Wie - zum Teufel - hat dann MS-DOS trotzdem einen Weg zur RS-232 gefunden?!"

... ist MS-DOS konsequent!

Und nun eine Verständnisfrage: Wann wird wohl ein Input-Redirecting beendet: Bei Rückkehr auf System-Ebene oder am EoF der Input-Datei? Die Antwort bekommen Sie spätestens dann, wenn Ihr MS-DOS im siebten Computer-Himmel weilt, weil unerwarteterweise das zu bedienende Programm mehr Eingaben anfordert, als vom Input-File zur Verfügung gestellt werden. Hier ist MS-DOS nämlich konsequent:

§1. Input-Redirecting bleibt solange aktiv, wie das bediente Programm aktiv ist.

§2. Ist Input-Redirecting aktiv, ignoriere man die Tastatur (inklusiv Alt-CTRL-Del!).

§3. Wenn der Input-File leergelesen wurde, tritt §1 in Kraft!

Wer's nicht glaubt, möge einmal mit dem an sich wenig sinnvollen Befehl DEBUG < AUTOEXEC.BAT die Probe auf's Exempel durchführen: Solange DEBUG keinen Q-Befehl bekommt, bleibt DEBUG aktiv. Solange DEBUG aktiv ist, bleibt die Eingabeumleitung aktiv. Solange die Eingabeumleitung aktiv ist, bleibt die Tastatur inaktiv. Solange ... (Hoffentlich verfügt Ihr Rechner über einen Hardware-Reset, sonst hilft nur noch der Griff zum großen roten Schalter hinten rechts!) Ach, wie einfach wär's doch, nach dem Leerlesen der Datei wieder auf die Tastatur zu lauschen!

Absteigende Versionsnummern?

Zu Guter Letzt noch eine MS-DOS-Logik: Wenn Sie davon ausgehen dürfen, daß mit jeder neuen Version eines BASIC-Interpreters bessere Features hinzukommen und ältere Fehler ausgemerzt werden, so scheint Microsoft seine GW-BASIC-Versionsnummern in absteigender Reihenfolge zu vergeben. Wie ich zu dem Schluß komme? Nun, starten Sie einmal dieses BASIC-Progrämmchen

```
100 FOR ZEILE=1 TO 24
110  FOR SPALTE=1 TO 80
120   LOCATE ZEILE,SPALTE
130  NEXT SPALTE
140 NEXT ZEILE
150 GOTO 100
```

so werden Sie als glücklicher Besitzer der BASIC-Version 2.02 den Cursor schön brav über den Bildschirm flitzen sehen; GW-BASIC 3.11 kann dies nicht: Die Cursorposition wird zwar intern nachgeführt, der Cursor selbst bleibt jedoch immer an der Position der letzten Ausgabe hängen. LOCATE mit nachfolgendem PRINT kann so zwar noch sinnvoll programmiert werden; für die Cursor-Nachführung bei bildschirmorientierter Eingabe ist mit dieser Interpreter-Version allerdings Trick 17 nötig:

```
10000 DEF FNCS$=CHR$(SCREEN(CSRLIN,POS(0)))
10010 :
50000 REM
50010 REM    Unterprogramm Cursornachführung
50020 REM
50030 IF SPALTE=1 THEN IF ZEILE>1 THEN CZEILE=ZEILE-1:CSPALTE=80
50040 IF SPALTE>1 THEN CZEILE=ZEILE:CSPALTE=SPALTE-1
50050 LOCATE CZEILE,CSPALTE
50060 PRINT FNCS$;
50070 RETURN
```

Der Editor des GW-BASIC 2.02 versteht diverse CTRL-Sequenzen, die in kaum einem Handbuch erwähnt sind:

```
CTRL ...  B: Cursor ein Wort zurück
          C: Abbruch, wie CTRL-BREAK
          E: Löschen bis Zeilen-Ende
          F: Cursor ein Wort weiter
          G: Beep
          H: Zeichen rechts vom Cursor löschen
          I: TAB
          J: Leerzeile einfügen
          K: Cursor Home
          L: Bildschirm löschen
          M: New Line
          N: Cursor ans Zeilenende
          R: Einfügen An/Aus
          T: Key ON/OFF (nur in Version 3.11)
          U: Eingabe bzw. Zeile löschen
          W: Wort an Cursorposition löschen
          X: Vorhergehende Zeile listen
          Y: Nächste Zeile listen
          Z: Ab Cursorposition bis Bildschirmende
             löschen
```

In der Version 3.2 funktionieren hiervon CTRL T bis CTRL Z nicht; an Stelle dieser Editor-Hilfen verzieren hier die entsprechenden ASCII-Symbole 21 bis 26 den Programmtext! Auch der Abbruch mit CTRL-C ist dort nicht möglich.

SHELL oder nicht SHELL, das ist hier die Frage

Version 2.02 erlaubt ein SHELL "BASIC"; die höheren Versionen 3.11 und 3.20 verbieten dies mit der Meldung "You can not SHELL to BASIC"! Teils zu Recht, da die 3er-Versionen das MS-DOS hierdurch in die ewigen Jagdgründe schicken können; teils zu Unrecht, weil die Version 2.02 beweist, daß ein SHELL-to-BASIC durchaus funktioniert:

Von GW-BASIC V2.02 nach Version 3.xx :
geht problemlos!

Von GW-BASIC V2.02 nach Version 2.02 :
geht problemlos!

Von GW-BASIC V3.xx nach Version 3.xx :
"You can not ..." !

Von GW-BASIC V3.xx nach Version 2.02 :
"You can" zwar, aber beim Ausstieg aus dem gerufenen BASIC hängt MS-DOS!

Peinlich wird so etwas, wenn Sie BASIC-Module mit SHELL "BASIC MODULxxx" als echte Unterprogramme - mit echten lokalen Variablen! - rufen möchten. Vorgesehen hatte ich dies zum Beispiel für eine On-Line-Hilfe mit der berühmten F1-Taste:

```
10020 ON KEY(1) GOSUB 65000
10030 :

20000 REM      Bearbeitung Kontenstamm
20010 TOPIC$="KONTEN":KEY(1) ON
20020 :
21000 KEY(1) OFF:RETURN
21010 :

40000 REM      Einlesen Mehrwertsteuer-Schlüssel
40010 TOPIC$="MWST":KEY(1) ON
40020 :
41000 KEY(1) OFF:RETURN
41010 :

65000 SHELL "BASIC "+TOPIC$+".HLP"
65010 RETURN
```

was auch auf meiner BASIC-Version zur vollsten Zufriedenheit funktionierte, bis ich feststellen mußte, daß die 3er-BASICs dies nicht mehr erlauben! Zudem führen die 3er-Versionen bei jedem SHELL-Aufruf ein "Cursor-Home" aus, was den ganzen Bildaufbau störend beeinflussen kann; Version 2.02 beläßt dagegen den Cursor schön brav, wo er ist.

Nach solchen Erfahrungen glaube ich nun fest an die absteigende Vergabe der Interpreter-Versionsnummern, benutze seitdem das "neueste" GW-BASIC 2.02, nehme dabei in Kauf, zu den weniger leistungsfähigen Versionen ab 3.xx nicht mehr abwärts- (oder aufwärts-?) kompatibel zu sein und hoffe auf GW-BASIC 1.0: Dies wird vielleicht endlich auch eine EGA-Grafik vernünftig bedienen, nach WIDTH "SCRN:",132 tatsächlich den 132-Spalten-Modus verstehen und wahrscheinlich auch den RESUME-Befehl vernünftig ausführen können. Werfen Sie einmal einen Blick auf diese Zeilen:

```
10000 ON ERROR GOTO 10080
10010 OPEN "R",1,"A:MURPHY":REM A:MURPHY existiere bereits!
10020 FIELD 1,128 AS ERFINDER$
10030 LSET ERFINDER$="IDIOT"
10040 PUT 1,1
10050 CLOSE 1
10060 PRINT "Daten OK gesichert!"
10070 END
10080 IF ERR<>70 THEN END
10090 PRINT "Bitte Schreibschutz entfernen. ";
10100 INPUT "Fertig ";DUMMY$
10110 RESUME
```

Offensichtlich fängt hier ON ERROR die Fehlermeldung "Write protected diskette" ab, bittet Sie, den Schreibschutz zu entfernen und nimmt mit RESUME die Verarbeitung wieder auf. Da aber frei nach Murphy kein Programm idiotensicher ist, weil Idioten so erfinderisch sind, wird, falls dieser sprichwörtliche Erfinder den Schreibschutz trotzdem auf der Scheibe läßt, ... Aber probieren Sie doch selbst!

Zurück zum DOS: Jüngst sah ich mich der Situation konfrontiert, ein Programm, welches seine Daten partout auf der Festplatte C: erwartet, auf einer reinen Diskettenmaschine ablaufen lassen zu müssen. Null Problemo: ASSIGN C=A leitet ja alle Zugriffe auf C: stattdessen nach A: um. Denkste! DOS's Kommentar "Illegal Parameter" - logisch (???), da ja Laufwerk C: gar nicht vorhanden ist, demzufolge auch nicht Parameter eines Befehls sein darf - machte jedwede Hoffnung zu nichte.

Liebe Firma Microsoft: Wie wär's, statt immer nur neue Features in die nächsten MS-DOS-Versionen hineinzustricken, zunächst endlich einmal die Bugs aus den alten Versionen zu beseitigen? So bleibt zu befürchten, daß auch OS/2 zwar weitere Möglichkeiten - Sprich: Multitasking - verspricht, uns arme Irren aber, die mehr wünschen, als nur fertige Programme (Wer entwickelt eigentlich unter solchen Voraussetzungen brauchbare Programme???) laufen zu lassen, durch solche Fehler weiterhin zur Weißglut treibt!

In Richtung Multitasking schwant mir sowieso Böses: Schon MS-DOS verspricht mit seinem Drucker-Spooler "PRINT.COM" einen Touch "Multi". PRINT funktioniert offensichtlich auch; nur möchten Sie natürlich gerne die Spool-Dateien nach erledigtem Druckauftrag wieder löschen können. Kein Problem: Sie befehlen "PRINT spoolfile" und "DEL spoolfile". Jedes vernünftige Multitasking wird ja wohl dem Task "DEL" solange den Zugriff auf "spoolfile" verbieten, bis der Task "PRINT" seine Aufgabe erledigt hat! Oder etwa nicht ... ??? Dabei wären nicht einmal großartige (Semaphor-) Klimmzüge nötig; PRINT.COM bräuchte die Spool-Dateien nur für die Dauer des Ausdrucks mit dem Read-Only-Attribut zu versehen, um das Schlimmste zu verhindern!

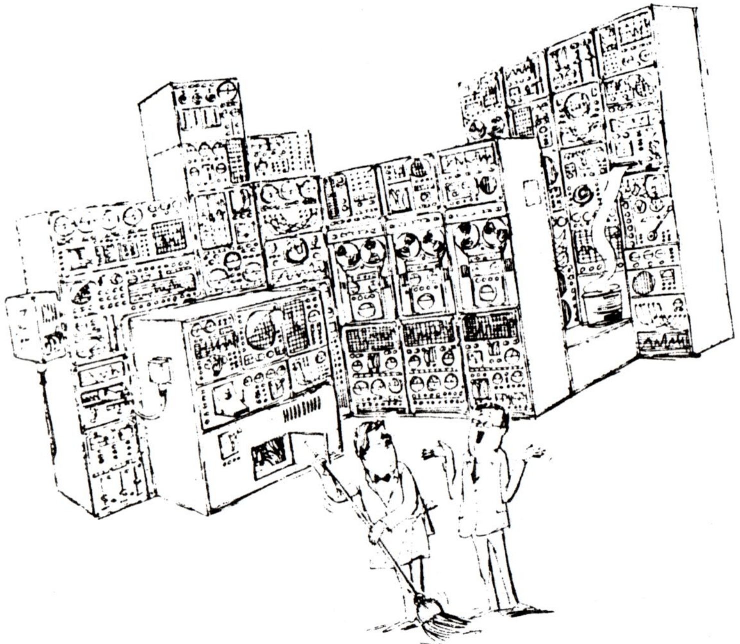
Zum Abschluß noch eine Preisfrage:

Sie befehlen MS-DOS: COPY *.TXT \TEXTE\SUMTEXT

a) Fasst der Befehl alle *.TXT-Dateien in der Datei SUMTEXT im Verzeichnis \TEXTE zusammen?

b) Kopiert dieser Befehl alle *.TXT-Dateien in gleichnamige Dateien ins Verzeichnis \TEXTE\SUMTEXT?

Ach so, was ich am Ende noch bemerken möchte: Falls Sie's etwa als Vorteil ansehen sollten, daß MS-DOS bereits zum Lieferumfang Ihres IBM-Kompatiblen gehört, dann lassen Sie sich eines Besseren belehren. Kaufen Sie Ihr MS-DOS lieber direkt von Microsoft, sonst bekommen Sie bei Rückfragen zum DOS nur den schönen Formbrief, Microsoft unterstütze keine OEM-Versionen seines Betriebssystems!



»Aber Chef, Sie haben doch selbst gesagt, der Computer solle alle Mitarbeiter ihren Fähigkeiten entsprechend einsetzen!«

Wie funktioniert ein Vektorgenerator?

Hochauflösende Grafikkarte für TRS-80 (Teil 2)

Das Listing des Vektorgenerators aus der letzten Ausgabe wird hier genauer erklärt. Zuerst aber soll ein Treiber für die Grafikkarte in Pascal behandelt werden, der mit dem Vektorgenerator in Assembler zusammenarbeitet, anschließend finden Sie den Algorithmus des Vektorgenerators in umgangssprachlicher Form erläutert.

Behandeln wir zunächst kurz den Bildschirmtreiber (Listing 3). Er ist so geschrieben, daß die Parameter der Zeichenbefehle in der Einheit »cm« für einen DIN-A3-Plotter mit einem 36 mal 26 Zentimeter großen Zeichenbereich nicht verändert werden müssen (der Nullpunkt des Koordinatensystems liegt in der Mitte). Trotz »Hochauflösung« ist die Darstellung auf dem Bildschirm im Vergleich zum Plotter recht grob — den 640 Bildpunkten in X-Richtung steht eine Auflösung von 3600 Schritten (10 pro Millimeter) auf einem gewöhnlichen Plotter gegenüber. Die Anpassung dieser Werte an die Anzahl Schritte (das heißt adressierbare Punkte) im Darstellungsbereich der Grafikkarte wird durch Multiplikation mit den Faktoren »cxstepfactor« und »cystepfactor« vorgenommen. Durch die unterschiedlichen Beträge der beiden Konstanten wird auch die etwas unterschiedliche Auflösung in X- und Y-Richtung ausgeglichen. Ein Offset (Verschiebung) für den Nullpunkt ist nicht nötig; dies und das Umdrehen der Y-Achse (siehe weiter unten) erledigt die Assemblerroutine. Die Parameter, welche das Gerät »Bildschirm«, das durch unseren Treiber angesprochen wird, kennzeichnen, werden durch eine Anzahl von Funktionen einem übergeordneten Grafiksystem übermittelt (in einfacheren Anwendungen kann auch darauf verzichtet werden).

Wesentlich sind die beiden Prozeduren »smoveto« und »sdrawto« (das »s« vor dem Namen steht für »screen«). Mit ihnen kann ein imaginärer Zeichenstift an einen neuen Anfangspunkt bewegt (smoveto) oder vom alten Anfangspunkt eine Linie zu einem neuen Zielpunkt (sdrawto) gezogen werden. Die Prozeduren »sdrawtext«, »sdrawreal« und »sdrawinteger« dienen dem Anschreiben von Texten auf dem Bildschirm. Sie sprechen in unserem Fall einfach das normale Video-RAM des Computers an, wobei sie einen Schriftzug in der Zeile und Spalte starten lassen, die der zuvor durch »smoveto« angefahrenen Position am nächsten liegen. Die Möglichkeiten der Schriftausgabe sind daher auf die übliche 80 mal 24-Zeichendarstellung eines Bürocomputers beschränkt.

Die Prozedur »sinitdevice« dient dazu, den Bildschirm zu löschen (weiß oder schwarz) oder das Negativ der Darstellung zu bilden.

Der Bildschirmtreiber ist so ausgelegt, daß er auch Farbbildschirme ansteuern könnte, wovon in unserem Beispiel natürlich kein Gebrauch gemacht werden kann (Prozedur »ssetcolor«).

In den Zeilen 80 und 81 von Listing 3 werden die beiden externen Prozeduren »screenline« und »screeninit« angefordert. Es handelt sich dabei um die schon erwähnten Assemblerroutinen, die nach dem compilieren mit dem Linker eingebunden werden müssen. Sie sind deshalb in Maschinensprache geschrieben, um möglichst schnell zu sein. Bei der Formulierung wurde jedoch in erster Linie Wert auf eine klare Gliederung und Lesbarkeit des Codes gelegt, ein Z80-Freak findet sicher noch Möglichkeiten, den Code geschwindigkeitsmäßig zu optimieren.

Um den Vektorgenerator in Listing 1 (letzte Ausgabe) zu verstehen, wollen wir uns zuerst den zugrundeliegenden Algorithmus überlegen. Sein Auftrag besteht darin, zwischen zwei Punkten eine gerade Linie zu ziehen. Diese Punkte werden durch die Wertepaare (xa, ya) und (xe, ye) bestimmt (siehe Zeile 81 in Listing 3). Da die Punkte völlig beliebig liegen können, kann die geforderte Linie jeden beliebigen Winkel annehmen — also nicht etwa nur waagrecht und senkrecht oder die Vielfachen von 45 Grad. Diese Richtungen wären leicht zu erzeugen — und sie können auch mit hardwaremäßiger Unterstützung der Karte erzeugt werden. Das Raster des Bildschirms läßt nur waagrechte durchgezogene Linien zu. Schon bei senkrechten Linien erkennt man meistens feine Pünktchen, während schräge Linien in mehr oder weniger regelmäßige Treppen aufgelöst werden.

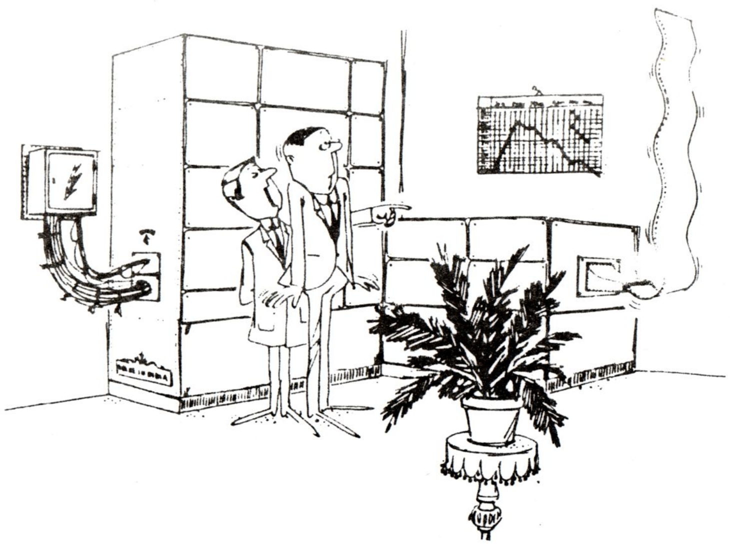
Offenbar erhält man eine Linie, die von links unten im Winkel von 45 Grad nach rechts oben verläuft, durch folgendes Vorgehen:

Man geht einen Schritt (einen Bildpunkt) in X-Richtung und einen Schritt in Y-Richtung, zeichnet einen Punkt und wiederholt dieses Verfahren. Um eine flachere Linie zu erhalten, braucht man bloß die Schritte in X-Richtung seltener auszuführen. Wenn man beispielsweise einen Schritt in Y-Richtung geht, einen Punkt zeichnet, wieder einen Schritt in X-Richtung, einen Punkt zeichnet und erst beim dritten Schritt in X-Richtung auch einen in Y-Richtung macht, so erhält man eine flache Treppe, die der Annäherung einer flachen Linie entspricht. Man kann sagen: Bei einer 45-Grad-Linie entsteht bei jedem Schritt in X-Richtung eine »Schuld« von einem Schritt in Y-Richtung. Bei der flacheren Linie beträgt die »Schuld« in Y-Richtung bei jedem X-Schritt ⅓-Schritte und braucht daher nur alle drei X-Schritte abgetragen zu werden. Nehmen wir an, wir würden vor dem Start des Algorithmus den Betrag der Schuld berechnen — dann könnten wir so verfahren, daß wir bei jedem Schritt in X-Richtung den Schuldbetrag zu einer Schuldsumme hinzuaddieren. Sobald die Schuldsumme den Wert 1 erreicht oder überschreitet, müssen wir einen Schritt in Y-Richtung ausführen — dafür dürfen wir von der Schuldsumme den Betrag 1 abziehen. Dabei gibt es offenbar ein Problem: Wenn die Linie steiler als 45 Grad ist, so wird die »Schuld« bei jedem X-Schritt größer als 1 Schritt in Y-Richtung — das aber kann man leicht dadurch umgehen, daß man zu Beginn entscheidet, ob die Differenz der Start- und Endkoordinaten auf der X-Achse oder auf der Y-Achse größer ist. Falls sie in Y-Richtung größer ist, vertauscht man einfach die Rollen von X und Y und merkt sich beim Fortschreiten in Y-Richtung die Schuld in X-Richtung.

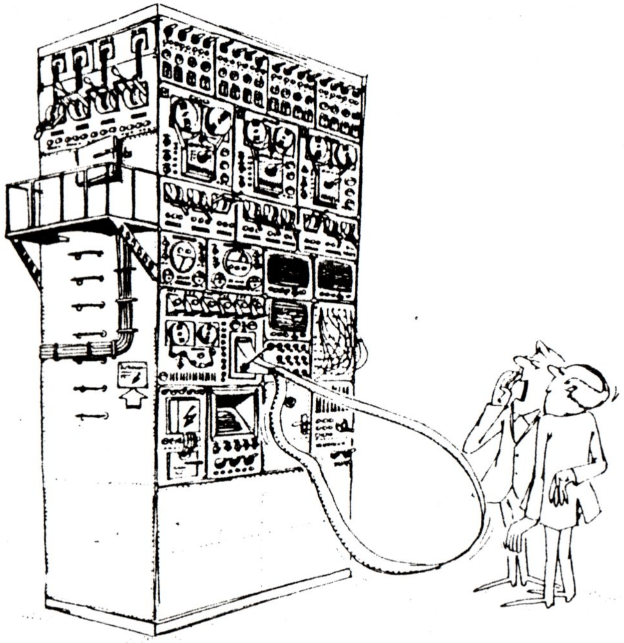
Um zu wissen, wann der Zielpunkt erreicht ist, zählen wir die ausgeführten Iterationen (Wiederholungen des Verfahrens). Dazu wird ein Zähler auf den Betrag von DELTAX (oder DELTAY) gesetzt und bei jeder Iteration um eins heruntergezählt. Sobald der Zähler den Wert Null erreicht hat, brechen wir, nach Zeichnen des Endpunktes, das Verfahren ab.

Offensichtlich treten bei diesem Verfahren, wie es eben geschildert wurde, Bruchzahlen für die Schuldbeträge auf. Das erschwert die Lösung durch eine einfache Assemblerroutine und mindert die geforderte Effizienz. Auch hier gibt es eine einfache Abhilfe: Wir betrachten einfach die gesamte Differenz in Y-Richtung als Schuldbetrag. Als Schwelle für das Einlesen der Schuld nehmen wir die gesamte Differenz in X-Richtung (bei einer Linie, die mehr in X-Richtung geneigt ist). Außerdem addieren wir zu Beginn den halben Schuldbetrag in die Schuldsumme, damit die erste Treppenstufe nur etwa die halbe Länge einer mittleren Treppenstufe beträgt. Das verleiht der approximierten Linie ein ausgewogeneres Aussehen. Wir kommen so mit einfachen Additions- und Subtraktionsbefehlen zur allgemeinen Lösung eines Problems, das nach der Schulmathematik aufwendige Winkelfunktionen erfordern würde (die erwähnte Halbierung des Schuldbetrages, die wir zu Anfang ausführen, erreichen wir durch einfaches binäres Rechtsschieben des fraglichen Differenzbetrages).

Wenden wir uns nun dem Listing 1 zu. Am Anfang findet man die Vereinbarung einiger Konstanten, darunter den jeweiligen Offset zur X- beziehungsweise Y-Koordinate, um den Nullpunkt in die Bildmitte zu verschieben. Die Konstanten INITGRAPH und EXITGRAPH dienen



»Wer hatte eigentlich die famose Idee, den Computer aus Indien zu importieren?«



»Um Himmels willen! Jetzt führt er schon Selbstgespräche!«

zum Ein- beziehungsweise Ausschalten der Grafikkarte. Weiterhin wird eine Anzahl von Speicheradressen durch Define-Word-Anweisungen belegt, die uns als Platzhalter für Variable dienen. Dies ist der Start- und Zielpunkt der Linie, die Schreibfarbe (keine, Weiß, Löschen, Invertieren oder Radar), der laufende Wert für X und Y, Merker, die uns sagen, ob wir die Schritte in positiver oder negativer Richtung ausführen müssen (XINC und YINC) sowie die Differenzbeträge DELTAX und DELTAY. Den Sinn der Variablen GOTIT erkläre ich später im Zusammenhang mit der »Schreibfarbe« RADAR. ERROR bezeichnet unseren Schuldbetrag, ZAEHLER zählt die Anzahl ausgeführter Schritte.

Zunächst müssen die Parameter, welche das Pascal-System beim Aufruf der Routine auf dem Stack deponiert hat, von diesem entnommen und aufbewahrt werden. Diese Vorgänge sind im Assemblerlisting durch Kommentare erläutert. Da auf der Grafikkarte die Y-Adressen von oben nach unten wachsen, ein mathematisches Koordinatensystem aber üblicherweise von unten nach oben positivere Werte annimmt, werden die Y-Beträge zwischen dem Holen vom Stack und dem Abspeichern invertiert (Vorzeichenwechsel). Außerdem wird zu den X- und Y-Beträgen der jeweilige Nullpunkt-Offset addiert. Aus bestimmten programmier-technischen Gründen ist die Routine als Funktion geschrieben, daher muß zuletzt noch ein Leerwort vom Stack genommen werden, das den Platz des Funktionsresultates belegt.

Anschließend werden die Differenzbeträge DELTAX und DELTAY berechnet. Da unsere Linie in alle erdenklichen Richtungen laufen kann, können die notwendigen Schritte in X- und Y-Richtung sowohl Plus 1 als auch Minus 1 betragen. Dies wird gleich bei der Berechnung der Differenz festgehalten, indem in die Variablen XINC und YINC jeweils die Zahl 1 oder —1 eingetragen wird.

Bei einer Berechnung von DELTAX wird, falls dieser Wert negativ ist, XSTART und XEND vertauscht. Wir erreichen dadurch, daß alle Linien von links nach rechts geschrieben werden. Damit soll sichergestellt werden, daß Linien, sofern Start- und Endpunkt identisch sind, sich genau decken — besser gesagt, daß ihre treppenförmigen Annäherungen deckungsgleich ausfallen. Das soll sichergestellt werden, damit man eine zuvor geschriebene Linie durch Überschreiben mit der Untergrundfarbe auch tatsächlich wieder auslöschen kann. Wir haben gesagt, daß wir die erste Treppenstufe nach einer mittleren Länge enden lassen (durch Voraufaddierung des halben Schuldbetrages). Es ist aber nicht vorherzusehen, ob die letzte Treppenstufe der Linie auch gerade die halbe mittlere Länge hat oder vielleicht länger oder kürzer ist als die erste. Würden wir daher


```
unit scrndrive;

(* 19. 12. 83 *)

(* Treiber fuer den TRS-80 MODEL II oder 16 mit Graphikkarte unter BGS-M 2.2 *)
(* - Version mit Vektorgenerator in ASSEMBLER *)

interface

function swmaxx: real;           (* Maximale X-Koordinate in cm (Window) *)
function swmaxy: real;           (* Maximale Y-Koordinate in cm (Window) *)
function swminx: real;           (* Minimale X-Koordinate in cm (Window) *)
function swminy: real;           (* Minimale Y-Koordinate in cm (Window) *)

function svmaxx: real;           (* Maximale X-Koordinate in DOTS (Viewport) *)
function svmaxy: real;           (* Maximale Y-Koordinate in DOTS (Viewport) *)
function svminx: real;           (* Minimale X-Koordinate in DOTS (Viewport) *)
function svminy: real;           (* Minimale Y-Koordinate in DOTS (Viewport) *)
function sxoffset: real;         (* Null-Punkt-Offset X-Achse *)
function syoffset: real;         (* Null-Punkt-Offset Y-Achse *)
function sxpositiv: boolean;     (* X-Achse positiv *)
function sypositiv: boolean;     (* Y-Achse positiv *)
function shascolor: boolean;     (* Verwaltet Zeichenfarbe selbst *)
function shasfile: boolean;      (* Memory-Mapped *)
function sopenport: boolean;     (* kein Default-Geraet *)
function sdevicestatus: integer; (* Geraetestatus abfragen *)
procedure sfilename (var filename: string); (* Dummy-Filename *)

procedure smoveto (x, y: real);
procedure sdrawto (x, y, scale: real);
procedure sdrawtext (s: string; ascale, richtung: real);
procedure sdrawinteger (i, stellen: integer; ascale, richtung: real);
procedure sdrawreal (r: real; vkstell, nkstell: integer;
                     ascale, richtung: real);
procedure ssetcolor (color: integer);
procedure sinitdevice (stat: integer);
procedure sinitfile (name: string);
procedure sclosefile;

implementation

const   cxdiv =        8;
        cydiv =       10;
        clines =      23;

        csxoffset =   40;    (* Versatz fuer Beschriftung *)
        csyoffset =   12;
```

```
        cwmaxx =       18.0;   (* Maximale X-Koordinate in cm (Window) *)
        cwmaxy =       13.0;   (* Maximale Y-Koordinate in cm (Window) *)
        cwminx =      -18.0;   (* Minimale X-Koordinate in cm (Window) *)
        cwminy =      -13.0;   (* Minimale Y-Koordinate in cm (Window) *)

        cvmaxx =       18.0;   (* Maximale X-Koordinate in DOTS (Viewport) *)
        cvmaxy =       13.0;   (* Maximale Y-Koordinate in DOTS (Viewport) *)
        cvminx =      -18.0;   (* Minimale X-Koordinate in DOTS (Viewport) *)
        cvminy =      -13.0;   (* Minimale Y-Koordinate in DOTS (Viewport) *)

        cxoffset =      0.0;      (* Null-Punkt-Offset X-Achse *)
        cyoffset =      0.0;      (* Null-Punkt-Offset Y-Achse *)

        cxstepfactor =  17.77778;      (* x * cxstepfactor = XKoordinate *)
        cystepfactor =  9.230769;      (* y * cystepfactor = YKoordinate *)

        cxpositiv =     true;   (* X-Achse positiv *)
        cypositiv =     true;   (* Y-Achse positiv *)
        chascolor =     true;   (* Verwaltet Zeichenfarbe selbst *)
        chasfile =      false;  (* Memory-Mapped *)
        copenport =     true;   (* kein Default-Geraet *)
        cfilename =     'remout:';      (* Dummy-Filename *)

type    tcolor =    (none, white, black, reverse, radar);

var     (screenfile:    nicht definiert;)
        penstatus:      tcolor;
        dummy, status:  integer;
        newx, newy:     integer;
        oldx, oldy:     integer;

procedure screeninit (col: tcolor); external;
function screenline (xa, ya, xe, ye: integer; i: integer): integer;
                         external;

function swmaxx; begin swmaxx := cwmaxx end;
function swmaxy; begin swmaxy := cwmaxy end;
function swminx; begin swminx := cwminx end;
function swminy; begin swminy := cwminy end;

function svmaxx; begin svmaxx := cvmaxx end;
function svmaxy; begin svmaxy := cvmaxy end;
function svminx; begin svminx := cvminx end;
function sviny; begin sviny := cvminy end;
function sxoffset; begin sxoffset := cxoffset end;
function syoffset; begin syoffset := cyoffset end;
function sxpositiv; begin sxpositiv := cxpositiv end;
function sypositiv; begin sypositiv := cypositiv end;
function shascolor; begin shascolor := chascolor end;
function shasfile; begin shasfile := chasfile end;
function sopenport; begin sopenport := copenport end;
function sdevicestatus; begin sdevicestatus := status end;
procedure sfilename; begin filename := cfilename end;

procedure smoveto;
begin
  newx := round (x * cxstepfactor); newy := round (y * cystepfactor);
  dummy := screenline (oldx, oldy, newx, newy, ord (none));
  oldx := newx; oldy := newy;
end (* smoveto *);

procedure sdrawto;
begin
  newx := round (x * cxstepfactor); newy := round (y * cystepfactor);
  dummy := screenline (oldx, oldy, newx, newy, ord (penstatus));
  oldx := newx; oldy := newy;
end (* sdrawto *);

procedure sdrawtext;
var     x, y:   integer;
begin
  if (ascale >= 1 (*cydiv*)) and (round (richtung / 90) = 0) then begin
    x := ((oldx div cxdiv) + csxoffset);
    if cypositiv then y := (clines - ((oldy div cydiv) + csyoffset))
    else y := ((oldy div cydiv) + csyoffset);
    gotoxy (x, y);
    write (s);
  end (* if darstellbar *);
end (* sdrawtext *);

procedure sdrawinteger;
var     x, y:   integer;
begin
  if (ascale >= 1 (*cydiv*)) and (round (richtung / 90) = 0) then begin
    x := ((oldx div cxdiv) + csxoffset);
    if cypositiv then y := (clines - ((oldy div cydiv) + csyoffset))
    else y := ((oldy div cydiv) + csyoffset);
    gotoxy (x, y);
    write (i:stellen);
  end (* if darstellbar *);
end (* sdrawinteger *);

procedure sdrawreal;
var     x, y:   integer;
begin
  if (ascale >= 1 (*cydiv*)) and (round (richtung / 90) = 0) then begin
    x := ((oldx div cxdiv) + csxoffset);
    if cypositiv then y := (clines - ((oldy div cydiv) + csyoffset))
    else y := ((oldy div cydiv) + csyoffset);
    gotoxy (x, y);
    write (r:vkstell:nkstell);
  end (* if darstellbar *);
end (* sdrawreal *);

procedure ssetcolor;
begin
  if color = 0 then penstatus := none
  else if color in [2..15] then penstatus := white
  else if color = 1 then penstatus := black;
end (* ssetcolor *);

procedure sinitdevice;
begin
  status := stat;
  case status of
    0:  screeninit (black);     (* Graphikkarte einschalten *)
    1:  screeninit (white);
    2:  screeninit (reverse);
  end (* case *);
end (* sinitdevice *);

procedure sinitfile;
begin
  (* Dummy *)
end (* sinitfile *);

procedure sclosefile;
begin
  (* Dummy *)
end (* sclosefile *);

begin (* Unit-Initialisierung *)
  penstatus := white;
  status := 0;
  sinitdevice (status);
  ***;
  screeninit (none);
end.
```

eine Linie von links unten nach rechts oben schreiben und von rechts oben nach links unten löschen, dann würden sich sehr wahrscheinlich die Punkte der Löschlinie nicht genau mit den Punkten der Schreiblinie decken, und es blieben »versprengte« Punkte auf dem Bildschirm stehen. Dadurch, daß Linien immer von links nach rechts geschrieben werden, ist sichergestellt, daß gleiche Strecken immer in der gleichen Richtung durchlaufen werden.

Von DELTAY wird der Absolutbetrag gebildet, um es ebenfalls positiv zu machen, es wird aber in YINC festgehalten, ob die Approximationsschritte in positiver oder negativer Richtung auszuführen sind. Schließlich wird entschieden, ob der Betrag von DELTAX oder der von DELTAY größer ist und je nachdem eine der beiden Routinen DOFORX (Linie stärker zur X-Achse parallel) oder DOFORY aufgerufen. Die auskommentierten JP- und RETURN-Anweisungen mit den Vermerken BREAK 1 und so weiter sind Debug-Anweisungen, die in der ausgetesteten Routine entfallen können.

Die beiden Unterroutinen DOFORX und DOFORY sollten aus der Schilderung des Algorithmus und den Kommentaren im Assemblerlisting heraus zu verstehen sein. Jedesmal, wenn ein Bildpunkt zu schreiben ist, wird die Routine DRAWDOT aufgerufen. Sie entscheidet anhand der Schreibfarbe, ob der Punkt weißgeschaltet, schwarzgeschaltet oder invertiert werden soll. Falls die Schreibfarbe 0 ist (keine), braucht natürlich nichts zu geschehen, es folgt ein RETurn-Befehl. Der Wert 4 (RADAR) soll zunächst außer acht gelassen werden. In diesem Fall wird der Label CALLPOINT erreicht, bei dem das Aufrufen der Routine POINT vorbereitet wird. Dazu werden dieser die X- und Y-Koordinate des gegenwärtig erreichten Bildpunktes auf dem Stack übergeben, sowie die Schreibfarbe. POINT gibt den Stack gesäubert zurück, anstelle des zunächst geladenen Leerwortes hinterläßt sie ein Funktionsresultat, das nur bei der Farbe RADAR Bedeutung hat.

Da POINT eine hardwareabhängige Routine ist, wurde sie getrennt geschrieben, sie ist aber in Listing 1 einkopiert und steht an dessen Ende. Wenn man diesen Teil entsprechend umschreibt, kann der Vektorgenerator auch für abweichende Hardware benutzt werden.

Um die Arbeitsweise von POINT zu verstehen, müssen wir uns näher mit der Architektur der Grafikkarte beschäftigen. Sie wird über vier Ports (Ein- und Ausgabekanäle) angesprochen. Die ersten beiden Ports gestatten das Laden je eines X- und Y-Adreßregisters. Die beiden Register adressieren zusammen ein Byte des Grafikspeichers. Dieses Byte kann über den dritten Port geschrieben oder gelesen werden. Der vierte Port dient zum Steuern bestimmter spezieller Arbeitsweisen der Karte.

Wenn ein Byte in den Grafikspeicher geschrieben wird, so sind davon immer acht nebeneinanderliegende Bildpunkte betroffen. Es kann aber sein, daß eine Linie nur durch einen dieser Punkte führt. Dann dürfen die übrigen sieben Bit des Speicherbytes nicht verändert werden. Daher muß die Routine POINT zunächst einmal die Adresse des Speicherbytes berechnen, in dem sich der zu behandelnde Bildpunkt befindet und dieses lesen. Die ersten Bildpunkte links oben gehören zu dem Byte mit der Adresse (0, 0), die letzten rechts unten zu dem Byte (79, 239). Nun muß die Bitadresse des Punktes in dem Byte bestimmt werden, dieses Bit gesetzt, rückgesetzt oder invertiert werden und schließlich das so modifizierte Byte in den Grafikspeicher zurückgeschrieben werden.

Die Y-Adresse des Bytes kann folglich direkt in das zugehörige Register der Grafikkarte geschrieben werden. Aus der X-Adresse sind dagegen zwei unterschiedliche Werte zu berechnen: zum einen die X-Adresse des Speicherbytes, zum anderen die Bit-Adresse des Bildpunktes in diesem Byte. Betrachten wir die X-Koordinate als Zahl, so ist die Bitadresse offenbar X mod 8, das heißt der Rest, der bei der ganzzahligen Division von X durch 8 bleibt. Die Byteadresse erhält man, indem man das ganzzahlige Ergebnis der Division X durch 8 nimmt. Beide Werte können in einem Arbeitsgang gewonnen werden — die Bitadresse steht bereits in den niederwertigsten drei Bit des Doppelregisters DE, in welchem POINT den Wert X abgelegt hat. Schiebt man DE dreimal nach rechts und bringt den Überlauf aus dem CARRY-Flag durch Rotieren in das Register C, so befindet sich in C die abgetrennte Bitadresse (die allerdings noch vollständig durch C nach rechts geschoben werden muß), in DE aber, da dreimal Rechtsschieben einer Division durch 8 entspricht, bereits die fertige X-Byteadresse. Wir haben also nichts weiter zu tun, als C noch fünfmal nach rechts zu schieben.

Die X-Adresse, die in diesem Fall nicht größer als 255 sein kann (tatsächlich höchstens 79 Dezimal) und folglich in ein 8-Bit-Register paßt, kann dem Register E entnommen und direkt in den X-Adreßport geschrieben werden.

Um die Bit-Adresse in Register C in die entsprechende Bitposition umzuwandeln, benutzen wir einfach eine Tabelle, die zu Anfang durch eine Liste von Konstanten bereitgestellt wurde (ab dem Label MUSTER). Wir laden HL mit der Startadresse der Tabelle (der Adresse MUSTER) und addieren die Bitadresse aus (B)C zu HL, anschließend laden wir C mit dem Inhalt der Adresse, die jetzt in HL steht — und haben damit im Register C das gewünschte Bit stehen. Jetzt kann das Video-Byte (Grafikspeicher) gelesen werden. Je nach Schreibfarbe wird das Bit in Register C zum Videobyte ODERiert (weiß), sein Komplement (Maske) UNDiert (schwarz) oder die EXOR-Funktion ausgeführt (invertieren).

Was hat es nun mit der geheimnisvollen »Farbe« RADAR auf sich? Sie werden schon bemerkt haben, daß wir ja den Inhalt des Grafik-RAMs lesen können. In diesem Zusammenhang kann man feststellen, ob auf einer gedachten Linie, die man sich als Radarstrahl vorstellt, irgendein Objekt vorhanden ist — ob also auf dieser Linie ein Bit gesetzt ist. Wenn nun die Prozedur POINT bei der Farbe »4« die Bits im Grafikspeicher nicht verändert, sondern nur testet und eine Null oder Eins zurückgibt, je nachdem, ob das Bit gesetzt ist oder nicht, so kann man anhand eines Iterationszählers im Vektorgenerator feststellen, wo sich — ausgehend vom Startpunkt der Linie — das nächstgelegene Objekt befindet — wie weit es also entfernt ist. Der Vektorgenerator tastet sich also so lange die gedachte Linie (den Radar-Strahl) entlang, bis er ein Objekt (ein gesetztes Bit) entdeckt — dieses Ereignis wird in der Variablen GOTIT durch Setzen auf 1 festgehalten. Allerdings hat das Sourcelisting in dieser Hinsicht einen kleinen Fehler — bei der Schreibfarbe »Radar« müßte das Vertauschen von Start und Ziel der X-Koordinaten unterbunden werden, weil sonst eine falsche Entfernung herauskommt. Nach Beseitigung dieser Schwäche ist diese Funktion

Der Rat des Computers? – Beten.

beispielsweise für Videospiele recht gut zu gebrauchen.

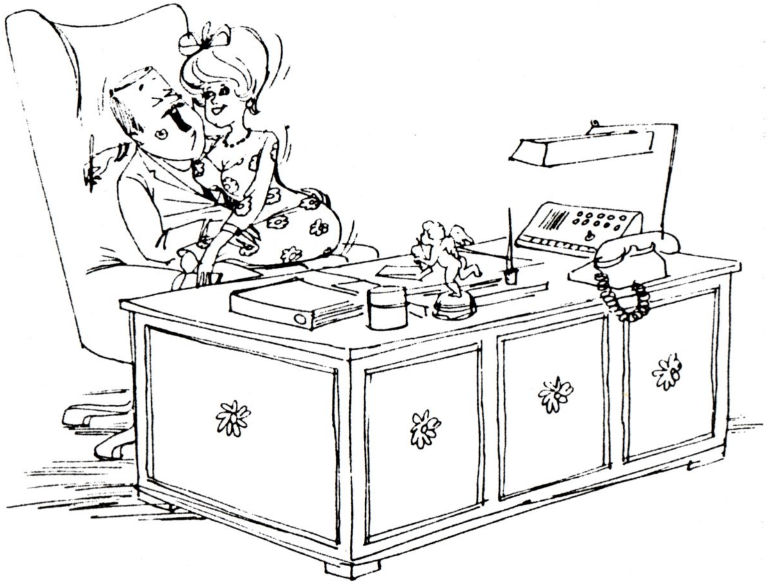
Zu unserem kleinen Satz von Grafikroutinen gehört noch die Routine »Screeninit« in Listing 2 (letztes Heft), die das Löschen (Weiß- oder Schwarzschreiben) sowie Invertieren (Bilden des Negativs) des gesamten Grafikbildschirms erlaubt.

Ich habe sie ebenfalls abgedruckt, möchte sie aber aus Platzgründen nicht ausführlich erläutern. Der interessierte Leser wird sie zusammen mit der Dokumentation der Grafikkarte sicher leicht verstehen.

Selbstverständlich sind zum Verständnis der hier abgedruckten Programme Erfahrungen in der Assemblerprogrammierung erforderlich, besonders dann, wenn sie an eine abweichende Hardware angepaßt werden sollen. Im Rahmen dieses Artikels kann keine Einführung in dieses Gebiet gegeben werden. Die abgedruckten Routinen wurden auf einem TRS-80 Modell 16 (in der Z80-Betriebsart) entwickelt und erprobt, das sich im Frühjahr 83 zum Test in der Redaktion befand; sie funktionierten auch auf dem Modell II des Autors ohne jede Änderung.

Die Karte ist einerseits die gegebene Erweiterung für TRS-80-Computer, um eine hochauflösende Grafik zu erzielen, wobei vielleicht der Preis (siehe unten) angesichts der mäßigen Hardware-Intelligenz etwas hoch erscheinen mag; andererseits bot sie die Gelegenheit, einmal die Grundlagen der Computergrafik an einem praktischen Beispiel darzustellen. Wir hatten zwei Karten im Test, die erste befand sich vor einigen Monaten in einem Testgerät, die Treibersoftware wurde bereits damals entwickelt. Anläßlich dieses Artikels erhielten wir ein anderes Exemplar, das in einem Modell II eingebaut wurde. In beiden Fällen funktionierte die Karte einwandfrei, ohne die sonstigen Funktionen der Testcomputer in irgendeiner Weise zu beeinflussen. Preise: Für Modell 2, 12, 16 kostet die Karte 1595 Mark inkl. Mehrwertsteuer. Eine ähnliche Karte mit gleicher Auflösung, die sich vom Benutzer aus gesehen gleich verhält, gibt es für Modelle 3 und 4 ab Februar '84 ab 768 Mark. (le)

Literaturhinweis: [1] William M. Newman/Robert F. Sproull, Principles of Interactive Computer Graphics, Second Edition, McGraw-Hill.



»Haben sie schon mal darüber nachgedacht, weshalb ich Sie noch nicht durch einen Computer ersetzt habe?«

Z80-Programmtricks in Assembler

Wer den richtigen Assembler besitzt, tippt wesentlich weniger, als der erzeugte Code vermuten läßt. »LD HL,(HL)« zum Beispiel ist da ein gültiger Befehl. Dahinter stecken eingebaute Makros, die dann die richtige Codesequenz generieren. Löst man sie auf, findet man die optimale Lösung für viele Standard-Probleme.

Andere Wege mögen auch nach Rom führen führen, und alle Assembler kann der Autor nicht abklopfen, weshalb in diesem Beitrag ausschließlich die Lösungen aus ALASM (Teil des »Assembler Development System« von Tandy) vorgestellt werden. Je nach Ausrüstung können Sie mit den hier vorgestellten Lösungen Ihre eigene Makro-Lib erweitern, die Routinen als Textfiles einziehen oder schlicht diese CP-Ausgabe als Nachschlagewerk neben den Computer legen. Beginnen wir mit den LD-Befehlen.

Schon »LD HL,BC« mag der Z80 nicht, die Folge »LD H,B:LD L,C« ist nötig. Bei den untrennbaren Pärchen AF, IX und IY schreibt man dann für zum Beispiel »LD HL,IX« die Folge »PUSH IX:POP HL«. Schon lohnender ist das eingangs erwähnte »LD HL,(HL)«, das sehr oft gebraucht wird, und zwar so:

```
PUSH    AF
LD      A,(HL)
INC     HL
LD      H,(HL)
LD      L,A
POP     AF
```

Zählen Sie nach (Bytes und Taktzyklen), der oft gebrauchte Weg via IX kostet mehr.

Und gleich die Umkehr, für »LD (HL),HL« ergibt sich:

```
PUSH    AF
LD      A,H
LD      (HL),L
INC     HL
LD      (HL),A
POP     AF
```

Mit (BC),BC; (BC),DE; (DE),BC und (DE),DE wird's etwas tricky, das geht so:

```
PUSH    1. Operand
EX      (SP),HL
LD      (HL),L-Byte 2. Operand
INC     HL
LD      (HL),H-Byte 2. Operand
EX      (SP),HL
POP     1. Operand
```

Zum Beispiel LD (BC),BC:

```
PUSH    BC
EX      (SP),HL
LD      (HL),C
INC     HL
LD      (HL),B
EX      (SP),HL
POP     BC
```

Hiervon nun wieder die Umkehr geht einfacher, zum Beispiel LD BC,(BC):

```
PUSH    BC
EX      (SP),HL
LD      C,(HL)
INC     HL
LD      B,(HL)
POP     HL
```

Blieben noch so nette Schreibweisen wie »LD (IX+dd),HL. Ganz einfach:

```
LD      (IX+dd),L
INC     IX
LD      (IX+dd),H
```

und ganz trivial: aus LD (IX+dd) wird LD A,(IX+dd):LD (IY+dd),A.

Wie auch immer: Schwierige Probleme waren das alles nicht, nur wenn ein Assembler solche Schreibweisen erlaubt, trägt das zur Lesbarkeit bei, spart einiges an Tipperei, und was das Wichtigste ist: Tippfehlern ist die Chance genommen. Sie wissen, was ein so verwechseltes B mit C für seltsame Fehler und stundenlange Debug-Sessions zur Folge haben kann.

Zwei Makros sollten Sie immer parat haben, nämlich MOVD und MOVI (Move mit Dekrement beziehungsweise Inkrement). Im ALDS gibt es dafür diverse Variationen über »LDIR mit Zubehör«, womit ich Sie nicht langweilen will. Schauen wir uns nur eine an:

```
MOVI nn1,(nn2):   LD      DE,nn1
                  LD      HL,nn2
                  LD      C,(HL)
                  LD      B,O
                  INC     HL
                  LD      A,B
                  OR      C
                  JR      Z,EXIT
                  LDIR
EXIT              EQU     $
```


Ganz im vorgenannten Sinne sollte man auch CMPD und CMPI kreieren. Hier die entsprechende Abart des vorhergehenden Beispiels:

```
CMPI nn1,(nn2):   LD      DE,nn1
                  LD      HL,nn2
                  LD      C,(HL)
                  LD      B,O
                  INC     HL
LOOP              LD      A,B
                  OR      C
                  JR      Z,EXIT
                  LD      A,(DE)
                  CP      (HL)
                  JR      NZ,EXIT
                  LDI
                  JR      LOOP
EXIT              EQU     $
```

Da unsere Subroutinen immer Werte in Registern verlangen, in denen sie beim Aufruf gerade nicht sind, ist Tauschen nicht zu vermeiden. Kombinationen gibt es viele, fünf Lösungen erschlagen alle.

1.) EX AF,BC; EX AF,DE; BC AF,DE an einem Beispiel:

```
EX AF,BC:     PUSH    AF
              PUSH    BC
              POP     AF
              POP     BC
```

2.) EX xx, yy wobei gilt: xx=AF, BC oder DE

```
              yy=IX   oder IY
                      das wird
              PUSH    xx
              EX      (SP),yy
              POP     xx
```

```
3.) EX IX,IY: PUSH    IX
              EX      (SP),IY
              POP     IX
```

```
4.) EX (SP),BC: EX    (SP),HL
                PUSH  BC
                PUSH  HL
                POP   BC
                EX    (SP),HL
```

Mit AF anstelle BC geht das auch

```
5.) EX (SP),DE:  EX    (SP), HL
                 EX    DE,HL
                 EX    (SP),HL
```

Zum Schluß noch ein weiterer Trick:

```
PUSH    AF
EX      (SP),HL
LD      A,L
POP     HL
```

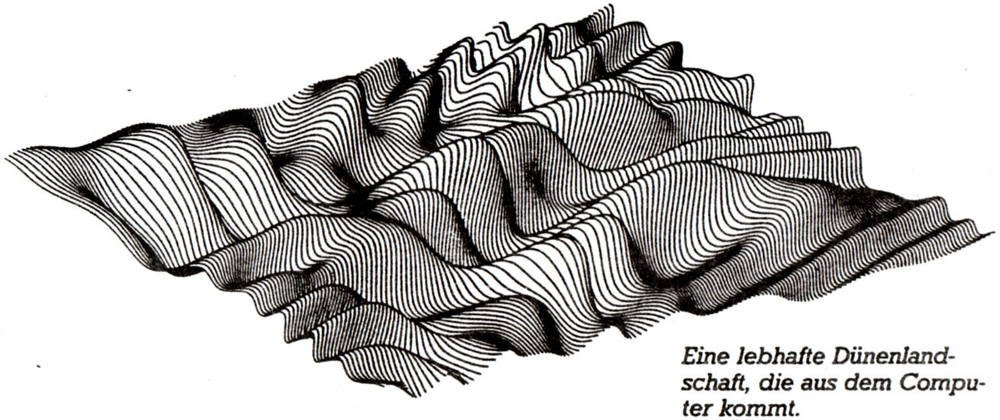
Diese Befehle realisieren schlicht »LD A,F«.

Sind Sie auch schon einmal mit einem Disassembler auf solche Sequenzen gestoßen und dann ins Grübeln geraten? So gesehen ist es doch richtig gemein, daß da einer nur tippt »LD (SP),BC«, und viele dann lange knobeln müssen! Jetzt schreiben wir auch so, und knobeln müssen nur noch die armen Kerlchen, die diese CP nicht gelesen haben.

(Peter Wollschläger/bo)

Traumlandschaften aus dem Computer

Endlich hat auch das deutsche Fernsehen sein vom Computer erzeugtes Markenzeichen. Und was dem Fernsehen recht ist, das ist dem Computerhobbyisten schon lange billig: Grafiken aus dem Mikrocomputer. Grafikprogramme können regelrechte Abenteuerspiele sein — sie schaffen bizarre Mondlandschaften oder sanft gewellte Dünen...



Eine lebhafte Dünenlandschaft, die aus dem Computer kommt.

Was da Anfang Oktober im ersten Programm an unwirklicher Wirklichkeit in die deutschen Wohnstuben flimmerte, ließ selbst dem hartgesottensten Computerignoranten den Atem stocken. Bei einer sanften Dünenlandschaft, die den Hintergrund eines bizarren Balletts aus sich verformenden farbigen Kugeln bildete, fühlte ich mich plötzlich an Computergrafiken erinnert, die ich zwei Tage zuvor mit einem experimentellen Programm angefertigt hatte: »Frequenzy Mountains« habe ich das Programm getauft — weil halt »Frequenzgebirge« so hausbacken klingt. Beides aber bedeutet dasselbe: Die computergenerierten Landschaften sind die dreidimensionale Darstellung mehr oder weniger komplizierter mathematischer Funktionen. Wenn man eine solche komplexe Funktion parametrisiert, kann man eine unerschöpfliche Fülle oft überraschender »Landscapes« erzeugen — es ist ein aufregendes Spiel, bei dem man nach Gefühl oder strategischen Überlegungen möglichst raffinierte Parameter eintippt und dann gebannt die Schreibspur auf dem Grafikbildschirm oder Plotter verfolgt: Was wird diesmal entstehen? Ein simpler Trick, der dafür sorgt, daß Täler, die hinter Gipfeln verborgen sind, unsichtbar bleiben, läßt die Landschaften dreidimensional erscheinen. Mehr darüber in der nächsten Ausgabe.

Das Programm, das diese Traumlandschaften erzeugt, ist mit 366 Zeilen (Turbo-Pascal) gar nicht besonders lang. Wir drucken es in diesem Heft ab, ohne es genau zu erläutern. Das soll in der nächsten Ausgabe geschehen — dort werden Sie auch eine Vielzahl traumhafter Farbfotos finden, die weitere Computerlandschaften zeigen, die von einem Farbgrafiksystem wiedergegeben wurden.

In der hier gezeigten Fassung steuert das Programm direkt den Plotter »MP 1000« an — die beiden Prozeduren »moveto« und »drawto« lassen sich aber leicht auch an andere Plotter anpassen. Ursprünglich wurde das Programm in UCSD-Pascal IV.13 geschrieben und lief auf dem Sage II mit einem universalen

Traumlandschaften aus dem Computer

Grafiksystem (BGS/M). Für Computer persönlich wurde es in Turbo-Pascal umgeschrieben. Die geänderten Stellen sind in der UCSD-Version in Kommentarklammern im Text verblieben, so daß Sie das Programm auch in UCSD eintippen können.

Nach Beseitigung von zwei kleinen Fehlern akzeptierte der Turbo-Compiler auf einem Z80-Computer das Programm. Erstaunlicherweise erhielt ich aber während der Übersetzung die Meldung »Memory Overflow«. Das liegt wohl daran, daß neben dem Turbo-System die verschiedenen Tabellen des Programms keinen Platz mehr im Speicher hatten. Nach Einschalten der Option (Wahlmöglichkeit), eine COM-Datei zu erzeugen anstatt im Arbeitspeicher zu übersetzen, schaffte es der Compiler jedoch. Nun lief das Programm, blieb aber stecken, wenn es den Plotter ansprechen sollte — ich hatte einfach das Kabel vom Sage auf den Centronics-Port des Modell 12 umgesteckt. Wahrscheinlich fehlte ein Bereitschaftssignal, um das sich der Sage nicht kümmert. Da ich aber mit der Parallelschnittstelle dieses Leihgerätes vorher schon ein anderes Problem hatte, wollte ich mich nicht länger damit aufhalten und überspielte das Programm auf den IBM-PC. Dort lief es auf Anhieb — hier verwendete ich ein Watanabe-Kabel, das den Plotter an den parallelen Druckerport des IBM anschließt.

Es zeigte sich aber, daß der 8088 bei Real-Arithmetik alles andere als ein Wunder an Geschwindigkeit ist: Das Programm arbeitete recht langsam, trotz der Prozeduren »fastsin« und »fastcos«, die — anstelle einer aktuellen Berechnung von Sinus und Cosinus — auf eine Tabelle zugreifen.

Da diese Tabelle nur eine Auflösung von 1 Grad bietet, werden die Kurven auf dem Plotter etwas rauh. Schöner (aber noch langsamer) ist es, wenn man statt »fastsin« die Pascal-Funktionen »sin« (bzw. »cos«) direkt verwendet. Einen Geschwindigkeitsrausch erlebte ich dagegen, als ich das Programm mit dem Arithmetik-Prozessor 8087 laufen ließ — es gibt ja eine Turbo-Version, die ihn unterstützt — und zwar direkt mit den mathematischen Pascal-Funktionen (nicht über die Tabellen). Man muß dabei nur darauf achten, daß alle Winkel durch die Konstante »rad« (Zeile 52) geteilt werden — so wie in Zeile 91! Das deshalb, weil die Winkelfunktionen in der Einheit »Pi-Radian« rechnen.

Die gezeigte Abbildung entstand mit folgenden, über das Menü einzustellenden Parametern:

Gewichtung der Grundfrequenzen A, B und C: Je 0.5.
Grundfrequenzen D: 3.0, E: 5.0, F: 7.0.
Amplituden-Modulations-Frequenzen G: 1.0, H: 2.0, I: 3.0.
Amplituden-Gewichtung J: 1.0, K: 0.7, L: 0.5, M: 0.3, N: 0.5, O: 0.7.
Phasen der Grundfrequenzen P: 0.0, Q: 60.0, R: 120.0.
Phasen der Amplitudenmodulation S: 0.0, T: 30.0, U: 60.0.
Punkte pro Kurve und Kurven X: 150, Y: 150.
Gesamtamplitude Z: 1.0.
FM-Frequenzen 1: 0.5, 2: 1.0, 3: 1.5.
FM-Amplituden 4: 0.3, 5: 0.5, 6: 0.7.
FM-Phasen 7: 0.0, 8: 90.0, 9: 270.0.
Bemerkung: FM bedeutet »Frequenz-Modulation«.

Zur Eingabe ist jeweils der Kennbuchstabe (A bis Z) beziehungsweise eine Kennziffer (1 bis 9) anzutippen, dann der entsprechende Zahlenwert und danach die RETURN-Taste.

Wenn Sie keine besonderen mathematischen Ambitionen haben, versuchen Sie lieber nicht, die hochwissenschaftlich wirkenden Einzelheiten des Programms zu verstehen — experimentieren Sie einfach, denn es ist ein Abenteuer-Programm! Ich kann Ihnen versprechen, daß Sie nicht enttäuscht sein werden.

Im nächsten Heft erläutern wir genau, wie das Programm funktioniert und was man alles damit machen kann. (le)

HEFT 27 Juli 1989

Dieses Programm erzeugt wildbewegte Meere, schroffe Gebirge oder sanfte Hügellandschaften.

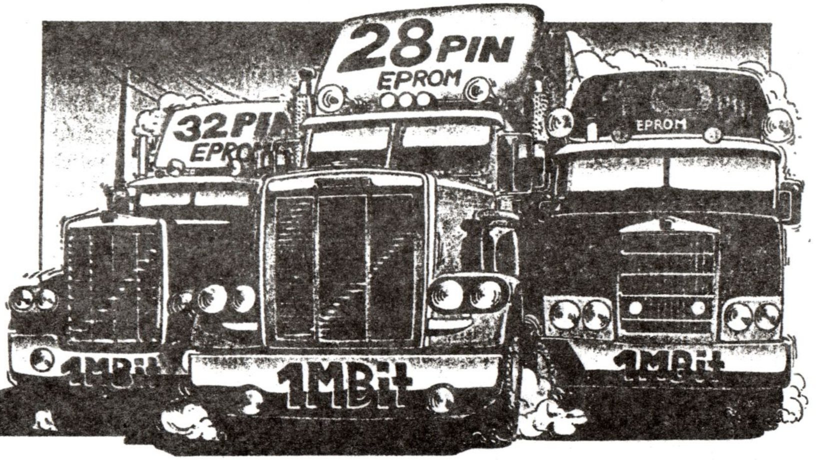
```
program gebirge;

(* vereinfachte Version mit Plotteransteuerung *)
(* fuer Computer Persoenlich/Markt & Technik. *)
(* (C) 1984 by Johannes Leckebusch *)
(* Public domain *)
(* Entworfen in: UCSD-Pascal IV.13 unter B6S/M *)
(* Version: Turbo-Pascal, stand alone *)
(* Plotter: Watanabe MP-1000 *)
(* Stand: 09. 10. 84 *)

(* UCSD IV.1x: *)
(* uses screenops; *)

const   cmaxslot =      511;
        ckurven =       45;
        cmaxpunkt =     45;
        cf1 =           4.00;
        cf2 =           3.00;
        cf3 =           6.00;
        cg1 =           1.0;
        cg2 =           1.0;
        cg3 =           1.0;
        cph1 =          0.0;
        cph2 =          0.0;
        cph3 =          0.0;
        caph1 =         0.0;
        caph2 =         0.0;
        caph3 =         0.0;
        cgroesse =      1.0;
        caf1 =          1.0;
        caf2 =          1.0;
        caf3 =          1.0;
        cag1x =         1.0;
        cag1y =         1.0;
        cag2x =         1.0;
        cag2y =         1.0;
        cag3x =         1.0;
        cag3y =         1.0;

        esc =           27;
        cr =            13;

        cxoffset =      17.95;  (* Null-Punkt-Offset X-Achse *)
        cyoffset =      12.95;  (* Null-Punkt-Offset Y-Achse *)
        cstepfactor =   100.0;  (* Faktor cm/Plotschritt *)
        (* UCSD Name der Plotschnittstelle: *)
        (* cplotname =     'printer:'; *)
        (* CP/M: *)
        cplotname =     'LST:';

        rad =           57.29578;

var     (* Parametertabelle: *)
        xschritt:       real;
        yschritt:       real;
        kurven:         integer;
        punkte:         integer;
        f1, f2, f3,
        g1, g2, g3,
        ag1x, ag1y, ag2x, ag2y, ag3x, ag3y,
        af1, af2, af3,
        ph1, ph2, ph3, aph1, aph2, aph3,
        fmf1, fmf2, fmf3,
        fmg1, fmg2, fmg3,
        fph1, fph2, fph3,
        groesse:        real;
        bef:            char;
        plotfile:       text;
        sintab:         array [0..90] of real;


(* Die folgenden beiden Prozeduren steuern den Plotter "MP 1000" *)

procedure moveto (x, y: real);
begin
  writeln (plotfile, 'M', round ((x + cxoffset) * cstepfactor), ',',
           round ((y + cyoffset) * cstepfactor));
end (* moveto *);

procedure drawto (x, y: real);
begin
  writeln (plotfile, 'D', round ((x + cxoffset) * cstepfactor), ',',
           round ((y + cyoffset) * cstepfactor));
end (* drawto *);

procedure inisintab;
var     w:      integer;
begin
  for w := 0 to 90 do
    sintab [w] := sin (w/rad);
end (* inisintab *);

function fastsin (w: real): real;
var     winind: integer;
        sign:   real;
begin
  winind := round (w) mod 360;
  if winind > 180 then sign := -1.0
  else sign := 1.0;
  winind := winind mod 180;
  if winind > 90 then winind := 180 - winind;
  fastsin := sintab [winind] * sign;
end (* fastsin *);

function fastcos (w: real): real;
begin
  fastcos := fastsin (w + 90.0);
end (* fastcos *);

procedure dosinusgebirge;
(* Diese Prozedur zeichnet ein "Gebirgediagramm" einer Sinusfunktion *)

(* Datum: 28. 09. 84 *)

var     z:              real;
        xoffset, yoffset: real;
        zmax:           array [0..cmaxslot] of real;
        (* Dieses Array haelt fest, wo der hoechste Punkt der alten *)
        (* Kurven liegt, eine neue wird nur gezeichnet, wenn sie *)
        (* hoeher liegt. *)
        zmin:           array [0..cmaxslot] of real;
        (* Dieses Array haelt fest, wo der tiefste Punkt der alten *)
        (* Kurven liegt, eine neue wird nur gezeichnet, wenn sie *)
        (* tiefer liegt *)
        zold:           real;
        scaleversatz:   integer;
        (* Haelt den seitlichen Perspektivischen Versatz der Kurven fest *)
        kurve, punkt:   integer;
        hidden:         boolean;        (* war letzter Punkt sichtbar? *)
        up:             boolean;        (* war letzter Punkt oberhalb? *)

function f (x, y: real): real;
(* Darzustellende Funktion *)

var     k, l, m:        real;
        m1, m2, m3:     real;
        r:              real;

begin
  m1 := (1 - fastcos (fph1 + y * fmf1 / kurven) * fmg1);
  m2 := (1 - fastcos (fph2 + x * fmf2 / punkte) * fmg2);
  m3 := (1 - fastcos (fph3 + (x + y) * fmf3 / (punkte + kurven)) * fmg3);
  k := fastsin (ph1 + x * f1 * m1 / punkte) * g1 *
    (1 - fastcos (aph1 + x * af1 / punkte) * ag1x) *
    (1 - fastcos (aph1 + y * af1 / kurven) * ag1y);

  l := fastsin (ph2 + y * f2 * m2 / kurven) * g2 *
    (1 - fastcos (aph2 + x * af2 / punkte) * ag2x) *
    (1 - fastcos (aph2 + y * af2 / kurven) * ag2y);

  m := fastsin (ph3 + (x - y) * f3 * m3 / (punkte + kurven) (* * 2.0 *)) * g3 *
    (1 - fastcos (aph3 + x * af3 / (punkte)) * ag3x) *
    (1 - fastcos (aph3 + y * af3 / (kurven)) * ag3y);

  f := (k + l + m) * groesse;
end (* f *);

begin (* sinusgebirge *)

  for kurve := 0 to cmaxslot do begin (* Initialisiere Hidden-Arrays *)
    zmax [kurve] := -20.0; (*initialisiere zmax*)
    zmin [kurve] := 20.0; (*initialisiere zmin*)
  end (*for*);
  zold := 0.0;

  xoffset := -18.0; yoffset := -9.0;
  (* Beginne mit dem Diagramm links unten *)
  scaleversatz := 0;

```

Listing »Gebirge«

```
(*******************************************************************)
for kurve := 0 to kurven do begin (* zeichne Funktionskurven *)
  z := 0.0; punkt := 0;
  hidden := false;
  (* Gehe davon aus, dass bei der ersten zu zeichnenden Linie *)
  (* der Startpunkt angefahren werden muss. *)
  moveto (xoffset, yoffset + f (0, kurve));
  (*************************************)
  for punkt := 0 to punkte do begin

    (*******************************************)
    z := f (punkt, kurve); (******************)
    (*******************************************)

    if z + yoffset >= zmax [punkt + scaleversatz] then begin
      (* Die Kurve ist von oben sichtbar: *)
      up := true;
      if hidden then begin moveto (xoffset + (punkt - 0.5) * xschritt,
                              (zmax [punkt - 1 + scaleversatz] +
                               zmax [punkt + scaleversatz]) / 2.0);
                         hidden := false;
                    end (* if hidden *);
      drawto ((xoffset + punkt * xschritt), (yoffset + z));
      zold := zmax [punkt + scaleversatz];
      zmax [punkt + scaleversatz] := z + yoffset;
      (* Neuer Gipfelpunkt der Kurve *)
      if z + yoffset < zmin [punkt + scaleversatz] then
        zmin [punkt + scaleversatz] := z + yoffset;
    end (* von oben sichtbar *)
    else if z + yoffset < zmin [punkt + scaleversatz]
    then begin
      (* Die Kurve ist von unten sichtbar *)
      up := false;
      if hidden then begin moveto (xoffset + (punkt - 0.5) * xschritt,
                              (zmin [punkt - 1 + scaleversatz] +
                               zmin [punkt + scaleversatz]) / 2.0);
                         hidden := false;
                    end (* if hidden *);
      drawto ((xoffset + punkt * xschritt), (yoffset + z));
      zold := zmin [punkt + scaleversatz];
      zmin [punkt + scaleversatz] := z + yoffset;
      (* Neuer Talpunkt der Kurve *)
      if z + yoffset > zmax [punkt + scaleversatz] then
        zmax [punkt + scaleversatz] := z + yoffset;
    end (* von unten sichtbar *)

    else begin (* unsichtbarer Punkt *)
      if not hidden and (punkt > 0)
      then begin (* Der letzte Punkt war sichtbar *)
        if up then
          drawto ((xoffset + (punkt - 0.5) * xschritt),
                  (zold +
                   zmax [punkt + scaleversatz]) / 2.0)
        else
          drawto ((xoffset + (punkt - 0.5) * xschritt),
                  (zold +
                   zmin [punkt + scaleversatz]) / 2.0);
      end (* if not hidden *);
      hidden := true;
    end (* Punkt unsichtbar *);
  end (* for punkt *);
  (*************************************)

    xoffset := xoffset + xschritt; yoffset := yoffset + yschritt;
    scaleversatz := scaleversatz + 1;
    (* Beginne eine Kurve mit perspektivischem Versatz *)
    (* Zum neuen Skalenanfang - um eine Position eingerueckt *)
  end (* for kurve *);
  (*************************************************)
  moveto (17.0, 12.0);
end (* sinusgebirge *);

procedure init;
begin
  kurven := ckurven;
  punkte := cmaxpunkt;
  xschritt := 36 / (punkte + kurven);
  yschritt := 18 / (kurven);
  f1 := cf1 * 360.0; f2 := cf2 * 360.0; f3 := cf3 * 360.0;
  g1 := cg1; g2 := cg2; g3 := cg3;
  ag1x := cag1x; ag1y := cag1y;
  ag2x := cag2x; ag2y := cag2y;
  ag3x := cag3x; ag3x := cag3x; ag3y := cag3y;
  ph1 := cph1; ph2 := cph2; ph3 := cph3;
  aph1 := caph1; aph2 := caph2; aph3 := caph3;
  af1 := caf1 * 360.0; af2 := caf2 * 360; af3 := caf3 * 360;
  fmf1 := 0.0; fmf2 := 0.0; fmf3 := 0.0;
  fmg1 := 0.0; fmg2 := 0.0; fmg3 := 0.0;
  fph1 := 0.0; fph2 := 0.0; fph3 := 0.0;
  groesse := cgroesse;
  (* UCSD: *)
  (* rewrite (plotfile, cplotname); *)
  (* Turbo: *)
  assign (plotfile, cplotname);
  rewrite (plotfile);
  inisintab;
end (* init *);

begin (* Hauptprogramm *)
  init;
  bef := ' ';
  repeat
    (* UCSD: *)
    (* sc_clr_screen; *)
    (* Turbo: *)
    ClrScr;
    writeln ('Gebirge-Generator              ESCape fuer Programmende');
    writeln ('<RETURN> Kurve zeichnen');
    writeln;
    writeln ('A, B, C Gewichtung             1: ', g1:3:3, ', 2: ', g2:3:3,
                 ', 3: ', g3:3:3);
    writeln ('D, E, F Frequenz               1: ', f1/360.0:3:3,
                 ', 2: ', f2/360.0:3:3, ', 3: ', f3/360.0:3:3);
    writeln ('G, H, I Amplituden-Frequenz 1: ', af1/360.0:3:3,
                 ', 2: ', af2/360.0:3:3, ', 3: ', af3/360.0:3:3);
    writeln;
    writeln ('J, K Amplituden-Gewichtung 1x: ', ag1x:3:3,
                 ', 1y: ', ag1y:3:3);
    writeln ('L, M Amplituden-Gewichtung 2x: ', ag2x:3:3,
                 ', 2y: ', ag2y:3:3);
    writeln ('N, O Amplituden-Gewichtung 3x: ', ag3x:3:3,
                 ', 3y: ', ag3y:3:3);
    writeln;
    writeln ('P, Q, R Phase                 1: ', ph1:3:3,
                 ', 2: ', ph2:3:3, ', 3: ', ph3:3:3);
    writeln ('S, T, U Phase Amplitude       1: ', aph1:3:3,
                 ', 2: ', aph2:3:3, ', 3: ', aph3:3:3);
    writeln;
    writeln ('X, Y Punkte pro Kurve:             ', punkte:3,
                 ', Kurven: ', kurven:3);
    writeln;
    writeln ('Z Gesamtamplitude:                 ', groesse:3:3);
    writeln;
    writeln ('1, 2, 3 FM-Frequenz           1: ', fmf1/360.0:3:3,
                 ', 2: ', fmf2/360.0:3:3, ', 3: ', fmf3/360.0:3:3);
    writeln ('4, 5, 6 FM-Amplitude          1: ', fmg1:3:3,
                 ', 2: ', fmg2:3:3, ', 3: ', fmg3:3:3);
    writeln ('7, 8, 9 FM-Phase              1: ', fph1:3:3,
                 ', 2: ', fph2:3:3, ', 3: ', fph3:3:3);

    (* UCSD: *)
    (* read (bef); *)
    (* Turbo: *)
    read (kbd, bef); if bef = chr (cr) then bef := ' ';
    writeln;
    case bef of
      ' ':   begin writeln ('Generator, bitte warten...');
               dosinusgebirge;
             end;
      'y':   begin write ('Kurven: '); readln (kurven);
               xschritt := 36 / (punkte + kurven);
               yschritt := 18 / (kurven);
             end;
      'x':   begin write ('Punkte: '); readln (punkte);
               xschritt := 36 / (punkte + kurven);
               yschritt := 18 / (kurven);
             end;
```

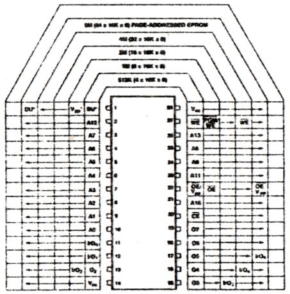
```
          '1':    begin write ('FMF1: '); readln (fmf1); fmf1 := fmf1 * 360.0;
                  end;
          '2':    begin write ('FMF2: '); readln (fmf2); fmf2 := fmf2 * 360.0
                  end;
          '3':    begin write ('FMF3: '); readln (fmf3); fmf3 := fmf3 * 360.0;
                  end;
          '4':    begin write ('FMG1: '); readln (fmg1); end;
          '5':    begin write ('FMG2: '); readln (fmg2); end;
          '6':    begin write ('FMG3: '); readln (fmg3); end;
          '7':    begin write ('FPH1: '); readln (fph1); end;
          '8':    begin write ('FPH2: '); readln (fph2); end;
          '9':    begin write ('FPH3: '); readln (fph3); end;
          'a':    begin write ('G1: '); readln (g1); end;
          'b':    begin write ('G2: '); readln (g2); end;
          'c':    begin write ('G3: '); readln (g3); end;
          'd':    begin write ('F1: '); readln (f1); f1 := f1 * 360.0 end;
          'e':    begin write ('F2: '); readln (f2); f2 := f2 * 360.0 end;
          'f':    begin write ('F3: '); readln (f3); f3 := f3 * 360.0 end;
          'g':    begin write ('AF1: '); readln (af1); af1 := af1 * 360.0 end;
          'h':    begin write ('AF2: '); readln (af2); af2 := af2 * 360.0 end;
          'i':    begin write ('AF3: '); readln (af3); af3 := af3 * 360.0 end;
          'j':    begin write ('AG1X: '); readln (ag1x); end;
          'k':    begin write ('AG1Y: '); readln (ag1y); end;
          'l':    begin write ('AG2X '); readln (ag2x); end;
          'm':    begin write ('AG2Y '); readln (ag2y); end;
          'n':    begin write ('AG3X '); readln (ag3x); end;
          'o':    begin write ('AG3Y '); readln (ag3y); end;
          'p':    begin write ('PH1: '); readln (ph1); end;
          'q':    begin write ('PH2: '); readln (ph2); end;
          'r':    begin write ('PH3: '); readln (ph3); end;
          's':    begin write ('APH1: '); readln (aph1); end;
          't':    begin write ('APH2: '); readln (aph2); end;
          'u':    begin write ('APH3: '); readln (aph3); end;

          'z':    begin write ('Groesse: '); readln (groesse); end;
       end (* case *);

   until bef = chr (esc);
end.
```

Listing »Gebirge« (Schluß)



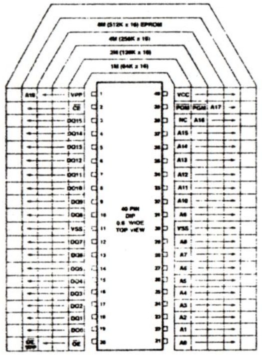
Die Mega-Renner von intel

Gewinnen Sie das Rennen um die höchste EPROM-Speicherdichte zum günstigsten Preis schon heute. INTEL liefert Ihnen Musterstückzahlen drei verschiedener 1-MBit-EPROMs zu einem Preis, der erst Mitte 1987 Wirklichkeit wird. Sie haben sofort Zugriff auf eine Million Bit in drei Gehäusen und drei verschiedenen Konfigurationen:



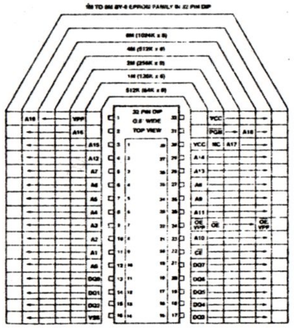
27011

heißt das Megabit-EPROM im Standard-28-Pin-Gehäuse. Organisiert in 8 Seiten zu je 16 KByte, also 8 x 16 K x 6.
Besondere Kennzeichen: Seitenadressierung
Vorteil: Verdoppelung der Speicherkapazität in der laufenden Produktion, z. B. mit bisher 512-KB-EPROMs.



27210

Das erste Megabit-EPROM mit 16-Bit-Wortbreite. Organisiert in 64 K x 16 Bit im 40-Pin-DIL-Gehäuse, bietet diese Version einfachste Anschlußbedingungen für 16-Bit- und 32-Bit-Mikroprozessoren, wie z. B. die Weltstandards 80286/80386 von INTEL.



27010

ist ein Megabit-EPROM im 32-Pin-Gehäuse mit Standardadressierung. Organisiert in 128 K x 8 Bit, ermöglicht es ein Wachstum in Speicherkapazitäten bis zu 8 MBit.

Die Parallelschnittstelle

Die marktführende Position des Druckerherstellers Centronics in den Anfangzeiten des Mikrocomputers hat viele Computerhersteller dazu veranlaßt, eine Parallelschnittstelle in ihre Systeme einzubauen, an die ohne Änderungen die Centronics-Drucker anzuschließen waren. Dies hat zu einem Quasi-Standard geführt, den heute fast alle Drucker- und Systemhersteller übernommen haben.

Als Verbindung zum Computer wird ein Kabel mit 36poligen Anschlußsteckern verwendet, bei dem jeweils ein Aderpaar miteinander verdrillt ist. Davon führt eine Leitung Massepotential und dient somit zur Abschirmung der Datenleitung vor äußeren Störeinflüssen oder vor der Beeinflussung nebeneinanderliegender Datenleitungen. Die Signalbedeutungen und die Belegung der Stecker sind weitgehend einheitlich, jedoch treten Unterschiede hinsichtlich von Erweiterungen und der Interpretation von Schnittstellensignalen durch die verschiedenen Hersteller auf, so daß eine Ausnutzung aller Möglichkeiten, welche die Parallelschnittstelle bietet, in den wenigsten Fällen erfolgt. Wie üblicherweise eine Centronics-Schnittstelle aussieht, zeigt Bild 1.

Die einzelnen Signale der Parallelschnittstelle lassen sich in vier Kategorien unterteilen:

— Datenleitungen

Die acht Datenleitungen übertragen die eigentliche Information und zwar so, daß jeweils ein ASCII-Zeichen komplett, zu einem Zeitpunkt übertragen wird.

— Handshake-Leitungen

Die Handshake-Leitungen regeln den Datenverkehr. Der Rechner erhält eine genaue Information über den Stand der Übertragungen beziehungsweise der Beschäftigung des Druckers. Es wird dadurch verhindert, daß Daten bei der Übertragung verlorengehen.

Die Handshake-Leitungen haben folgende Bedeutung:

Data-Strobe

Wenn das Rechnerinterface auf dieser Leitung einen kurzzeitigen Impuls liefert, wird die Information, die zuvor auf die Datenleitungen gelegt wurde, in den Drucker übernommen. Bevor dieser Vorgang stattfinden darf, muß sich der Computer von der Bereitschaft des Druckers überzeugt haben.

Acknowledge (Anerkennung, Quittung)

Diese Leitung wird vom Drucker benutzt, um dem Computer mitzuteilen, daß sein Zeichen ordnungsgemäß übernommen wurde. Die Mitteilung ist rein formal, eine Bestätigung, ob es sich tatsächlich um die ausgesandte Information handelt, kann bei dieser Übertragungsart nicht erfolgen.

Busy (beschäftigt)

Ist der Drucker in irgendeiner Weise beschäftigt, beispielsweise beim Ausdruck eines Zeichens, beim Wagenrücklauf oder Formularvorschub, so zeigt er seinen momentanen Status durch das Busy-Signal an. Bevor der Computer Daten an den Drucker senden darf, muß er sich in jedem Fall mittels des Busy-Signals vergewissern, daß der Drucker zur Datenaufnahme bereit ist.

— Zustandsmeldungen

Treten Fehlerzustände während der Übertragung oder beim Drucker auf (zum Beispiel Papierende oder Druckkopf klemmt), so kann dies über spezielle Signalleitungen dem Computer mitgeteilt werden, der daraufhin sein Druckprogramm abbrechen und eine entsprechende Fehlermeldung auf den Bildschirm ausgeben kann.

Im einzelnen sind dies:

PE (Paperend)

Kein Papier mehr im Drucker

SELECT

Der Drucker befindet sich im »Online-Mode« und kann Daten aufnehmen.

FAULT

Im Drucker ist ein Fehlerzustand eingetreten, der eine weitere Benutzung nicht möglich macht, beispielsweise Farbband gerissen, Druckkopf überhitzt, Schlitten klemmt etc.

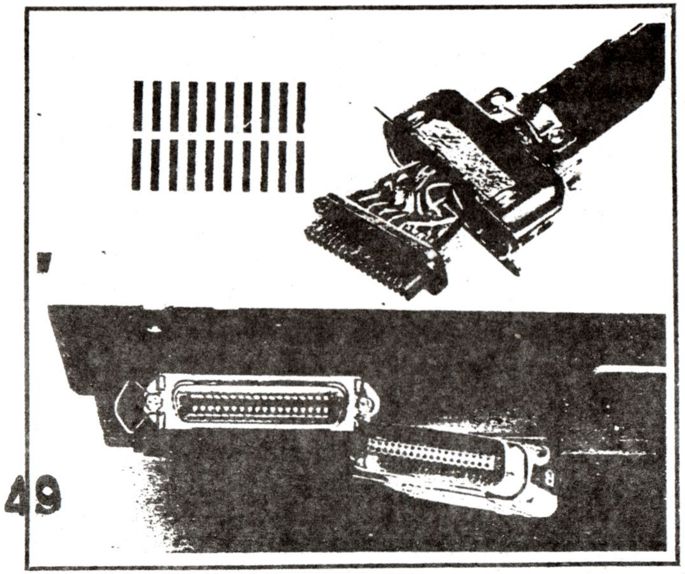


Bild 1. Centronics-Schnittstelle. Druckerkabel mit Cannon-Steckverbindung

— Spezielle Signale und Stromversorgung

Input-Prime oder Init

Mit diesem Signal kann man vom Computer aus eine Neuinitialisierung des Druckers bewirken. Empfängt der Drucker dieses Signal, so nimmt er eine Ausgangsstellung ein, die meist derjenigen nach dem Einschalten entspricht.

Die Tatsache, daß die Parallelschnittstelle keine richtige Norm ist, läßt natürlich eine weitere Belegung der freibleibenden Leitungen zu. Dies wird deutlich, wenn man beispielsweise die beiden Drucker MX80 F/T von Epson und 8510 von C. Itoh miteinander vergleicht (siehe Bild 2).

Beim Epson kann das AUTO-FEED-XT-Signal dazu benutzt werden, um den Drucker zu veranlassen, nach dem Empfang des Zeilenende-Zeichens »CR«, automatisch einen Zeilenvorschub auszuführen.

Der C. Itoh legt auf Pin 18 seine Stromversorgung von 5 V und liefert somit eine eindeutige Aussage, ob der Computer ein- oder ausgeschaltet ist (vergleiche Bild 3).

Besonders gravierend ist der Unterschied bei Abschluß 36. Er dient beim Epson dazu, den Drucker in den Select-Mode zu schalten und ist dementsprechend ein Eingangssignal, während Input-Busy beim Itoh ein Ausgangssignal ist und eine ähnliche Funktion wie das Busy-Signal, jedoch mit einem anderen Zeitverhalten hat.

In Bild 4 ist der Aufbau einer Rechnerschnittstelle aufgezeigt, die zum Parallelanschluß eines Druckers geeignet ist. Mit den heute zur Verfügung stehenden hochintegrierten Bausteinen ist der Aufbau einer solchen Schnittstelle sehr einfach und bei allen Computerherstellern in ähnlicher Form zu finden. Allgemeine Parallelschnittstellen sind für alle CPU-Familien erhältlich und arbeiten gleichartig. Ein Baustein vereinigt meist zwei oder drei 8 Bit breite »Ports« (Tore), die vom Computer her über bestimmte Adressen angesprochen werden können. Welche Adressen das sind, ist einzig und allein von der Verschaltung der Bausteine innerhalb des Computersystemes abhängig und muß jeweils der Systembeschreibung entnommen werden.

Die Parallelschnittstellenbausteine verhalten sich bezüglich ihrer äußeren Anschlüsse vollkommen neutral, das heißt erst durch einen Grundeinstellungsprozeß (Initialisierung) wird vom Programm her festgelegt, welcher Anschluß eines Ports Eingang und welcher Ausgang sein soll oder anders ausgedrückt, die im Computer eingebaute Parallelschnittstelle wird erst durch ein spezielles Programm, das im Betriebssystem vorhanden sein muß, zur Druckerschnittstelle.

	Epson MX80F/T	C. Itoh 8510
Pin 14	AUTO FEED XT	0 V
Pin 18	nc	+5 V
Pin 35	pulled up 5 V	nc
Pin 36	SLCT IN	Input-Busy
nc	heißt not connected = nicht angeschlossen	

Bild 2. Unterschiedliche Pin-Belegung der Centronics-Schnittstelle beim MX80 F/T und 8510

Ganz so flexibel ist die Sache in den meisten Fällen allerdings nicht, denn bevor die Daten und Steuersignale an die Anschlußbuchsen des Computers gelangen, werden sie noch durch sogenannte Leitungstreiber verstärkt. Diese Leitungstreiber sind elektronische Verstärkerbausteine (nicht zu verwechseln mit den Treiberprogrammen), die dafür sorgen, daß die zu übertragenden Signale noch »gut erkennbar« auf der anderen Seite des Kabels, also beim Drucker ankommen. Sie dienen gleichzeitig als Schutz des Schnittstellenbausteines gegen Kurzschlüsse, Fehl- oder Überspannungen im Kabel oder im angeschlossenen Gerät. Meistens werden unidirektionale Treiberbausteine vorgeschaltet, das heißt eine Signalübertragung wird nur in einer Richtung zugelassen. Hardwareseitig ist damit festgelegt, welcher Anschluß der Schnittstelle Eingang und welcher Ausgang ist. Bei der Programmierung der Schnittstelle muß auf diese Gegebenheiten Rücksicht genommen werden.

Strobe	1 ! 19	Masse
DATA-1	2 ! 20	Masse
DATA-2	3 ! 21	Masse
DATA-3	4 ! 22	Masse
DATA-4	5 ! 23	Masse
DATA-5	6 ! 24	Masse
DATA-6	7 ! 25	Masse
DATA-7	8 ! 26	Masse
DATA-8	9 ! 27	Masse
ACKNOW-LEDGE	10 ! 28	Masse
BUSY	11 ! 29	Masse
PAPER END	12 ! 30	Masse
SELECT	13 ! 31	INIT
0V	14 ! 32	ERROR
nc	15 ! 33	0V
0V	16 ! 34	nc
Chassis GND	17 ! 35	nc
x	18 ! 36	x

Bild 3. Pinbelegung bei der Centronics-Parallelschnittstelle

Bis hierhin können wir von einer physikalischen Anpassung sprechen. Die eigentliche Aufgabenstellung, nämlich die Ausgabefunktion des Computers an den Drucker und seine Überwachung, wird durch ein sogenanntes Treiberprogramm erreicht.

Wie ist ein Treiberprogramm für eine Druckeransteuerung aufgebaut?

Treiberprogramme gehören zum innersten Kern eines Betriebssystems, der die Verbindung der Hardware zum rein logischen Gerät des Betriebssystemes herstellt. Will man Korrekturen oder Erweiterungen daran vornehmen, müssen folgende Voraussetzungen bekannt sein:

1. Kenntnis über die Funktion des verwendeten Schnittstellenbausteines, seine Arbeitsweise und seine Programmierung.
2. Die Einbettung des Bausteines in die Architektur des Computers, also Ansprechadressen und Ansprechtechnik (spezielle Ein-/Ausgabebefehle oder Memory-Map).
3. Das Zusammenwirken des Treiberprogrammes mit dem Betriebssystem — Parameterübergabe, Pufferbereiche und zugehörige Pointer, eventuell auch die Einbindung des Treiberprogrammes in eine Interruptstruktur.

Die Anforderungen, die hier gestellt werden, sind also keineswegs gering und es liegt nahezu auf der Hand, daß eine vernünftige Programmierung hier nur in Assembler oder einer systemnahen Sprache, beispielsweise in C, erfolgen kann. Einfacher geht es, wenn zur Druckerausgabe ein Anwenderprogramm verwendet wird, denn hier entfällt das »Einbauen« des Treiberprogrammes in das Betriebssystem; die neu hinzugekommenen Möglichkeiten des Druckprogrammes können dann allerdings nicht von anderen Programmen genutzt werden.

Schema eines Treiberprogrammes für eine Parallelschnittstelle

Die Software zum Betrieb einer Parallelschnittstelle läßt sich in drei Hauptaufgaben gliedern.

Bewährtes DFÜ-Protokoll

Wer zwischen Computern Daten austauschen will, muß sich mit dem Partner über die Übertragungsregeln einig sein. Wir erläutern am Beispiel von »XModem« Aufbau und Arbeitsweise eines solchen Regelkataloges.

Das bekannteste Regelwerk — oder Protokoll — für den Austausch binärer Daten ist XModem. Es ist während der Blütezeit des CP/M-Betriebssystems von Ward Christensen entwickelt worden.

Folgendes Prinzip liegt der fehlerfreien Datenübertragung bei XModem zu Grunde: Die Informationen werden in Blöcken zu je 128 Bytes übertragen. Jeder Block wird beim Absenden mit einem Blockzähler und mit einer Prüfsumme versehen, so daß der Empfänger feststellen kann, ob ein Fehler bei der Übertragung aufgetreten ist.

Der Empfänger sendet nach dem Empfang jedes Blockes ein Antwortzeichen zurück, nämlich das Byte <ACK>, wenn der Block fehlerfrei empfangen wurde und die richtige Blocknummer aufwies, oder das Byte <NAK>, wenn irgend etwas nicht stimmte. Bei einem Fehler wird derselbe Block bis zu zehnmal wiederholt. Wenn allerdings alle zehn Versuche fehlschlagen, dann kapituliert XModem.

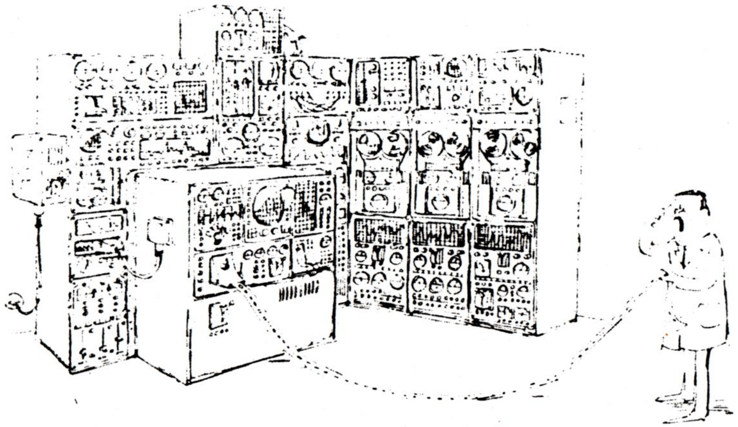
Die folgenden Zeichen werden zum Steuern der Übertragung verwendet:

Code	Englisch	Deutsch	Hexadezimal
<SOH>	Start of Header	Blockanfang	01
<EOT>	End of Transmission	Ende der Übertragung	04
<ACK>	Acknowledge	Bestätigung	05
<NAK>	Not Acknowledge	Keine Bestätigung	15
<CAN>	Cancel	Abbruch	18

Die Übertragung erfolgt asynchron mit 8 Datenbits, ohne Paritätsbit und mit einem Stopbit. Ein Übertragungsblock ist folgendermaßen aufgebaut:

<SOH>
<Blocknummer>
<255 — Blocknummer>
<128 Bytes Daten>
<Prüfsumme>

Hierbei bedeutet:
<SOH> = 01 hexadezimal
<Blocknummer> = binäre Zahl, beginnend mit 01, Zählschritt 1, beginnt nach FF wieder mit 00 (nicht 01)
<255 — Blocknummer> = Komplement der Blocknummer nach der 8080 »CMA« Instruktion, das heißt jedes Bit der 8-Bit-Blocknummer wird invertiert. Dies ist das »Einerkomplement«.
<Prüfsumme> = Summe nur der Datenbytes. Überträge werden vernachlässigt (Modulo 256).



»Und wenn du diesen Fehler jetzt noch einmal machst, dann schreibst du zur Strafe 5000mal: ICH BIN NUR EINE MASCHINE UND HABE KEINEN FREIEN WILLEN!«

1. Initialisierung der Schnittstelle, das heißt Vorbereitung für den Betrieb des Computers mit einem bestimmten Peripheriegerät, hier also einem Drucker.
2. Bedienung einer Ausgabeanforderung des Betriebssystemes (eigentliche Ausgabe)
3. Behandlung von Fehlerereignissen

Punkt 1. haben wir bereits angesprochen. Der Vorgang der Initialisierung wird einmalig beim Start des Computers durchlaufen und stellt die Schnittstelle zum ordnungsgemäßen Zusammenspiel mit einem bestimmten Drucker ein. Zusätzlich können bei diesem Vorgang noch bestimmte Zwischenspeicherbereiche (Puffer) reserviert und systemabhängige Einstellungen vorgenommen werden.

Das eigentliche Ausgabeprogramm wird erst dann angesprochen, wenn ein Anwenderprogramm ein Zeichen auf den Drucker ausgeben möchte. Ist das der Fall, so muß das Treiberprogramm folgendermaßen vorgehen:
1. Ist ein Drucker angeschlossen? nein → Fehlerbehandlung
2. Zeitüberwachung einschalten
3. Ist der Drucker selektiert? nein → Fehlerbehandlung
4. Ist der Drucker beschäftigt? ja → dann warte, gehe zu Punkt 3
5. Daten anlegen
6. Strobesignal erzeugen
7. Auf Quittung (ACK) warten
8. Zeitkontrolle ausschalten
9. Zurück zum aufrufenden Programm.

Im Hintergrund läuft die Zeitkontrolle, die ebenfalls eine Fehlermeldung generiert, wenn der gesamte Vorgang nicht innerhalb einer vorgegebenen Zeitspanne abgewickelt werden kann. Von den hier aufgezählten Schritten sind für eine einfache Ausgabe auch weniger Schritte möglich, und diese werden der Einfachheit halber auch von den meisten Herstellern in einer solchen abgemagerten Form angeboten. Das Ganze sieht dann etwa so aus:
1. Ist der Drucker belegt (entweder beschäftigt, nicht selektiert eventuelle Fehler oder gar nicht angeschlossen)? nein → dann warte
2. Daten anlegen
3. Strobesignal erzeugen
4. Zurück zum aufrufenden Programm.

In der Praxis »wartet« dann meist nicht nur das Ausgabeprogramm, sondern auch der Benutzer, der (wenn er sich nicht gerade auf dem Computersektor auskennt), sich in den wenigsten Fällen zu helfen weiß. Das »Nichtstun« eines Druckers kann viele Ursachen haben:
1. Der Drucker ist ausgeschaltet
2. Der Drucker ist eingeschaltet, aber nicht angewählt (selektiert)
3. Der Drucker ist eingeschaltet und selektiert, aber das Verbindungskabel steckt nicht im Computer
4. Verbindungskabel steckt, ist aber defekt
5. Verbindungskabel ist in Ordnung, aber der Drucker zeigt einen Fehlerzustand an, zum Beispiel kein Papier, Farbband ist verbraucht, Druckkopf klemmt
6. Drucker oder Schnittstelle sind defekt; hier hilft allerdings auch kein komfortables Treiberprogramm, allenfalls ein Diagnose- und Druckertestprogramm.

Wir haben nun die Parallelschnittstelle in groben Zügen besprochen. Sie ist von der Programmierung einfacher zu handhaben, als die serielle Schnittstelle und erlaubt anhand spezieller Signale wie ERROR, SELECT und ACKNOWLEDGE die Richtigkeit der Übertragung zu prüfen und Fehler- oder Betriebszustände des Druckers abzufragen. Dafür bringt sie einen größeren Aufwand bezüglich der Verbindungskabel mit sich und ist nur für relativ kurze Strecken von 1 bis 5 Meter anwendbar.

(Peter Bonsch)

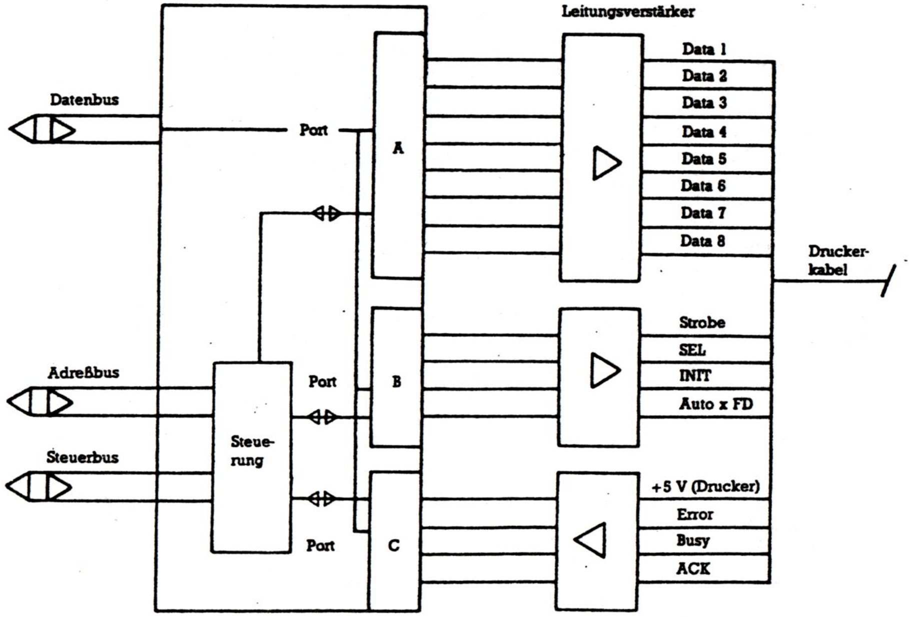


Bild 4. Schaltschema einer Parallel-Schnittstelle für den Anschluß eines Druckers

Das folgende Beispiel zeigt eine typische Übertragung einer drei Blöcke langen Datei. Es enthält die beiden häufigsten Übertragungsfehler — einen falsch übertragenen Block und ein falsch übertragendes <ACK>-Zeichen.

SENDER — EMPFÄNGER (Timeout und Wiederholung nach 10 Sekunden)

Der Sender wartet. — Der Empfänger beginnt mit:

<—<NAK>

Der Sender sendet den ersten Block.

<SOH> <01> <FE> Daten <Prüfsumme>—>

Der Empfänger hat den Block einwandfrei empfangen und antwortet:

<—<ACK>

Der Sender sendet den zweiten Block.

<SOH> <02> <FD> Daten <Prüfsumme>—>

Eine Leitungsstörung tritt auf. Der Empfänger hat den Block nicht einwandfrei empfangen und antwortet mit:

<—<NAK>

Der Sender wiederholt den zweiten Block.

<SOH> <02> <FD> Daten <Prüfsumme>—>

Der Empfänger antwortet mit:

<—<ACK>

Der Sender sendet den dritten Block.

<SOH> <03> <FC> Daten <Prüfsumme>—>

Der Empfänger antwortet mit:

<—<ACK>

Eine Leitungsstörung tritt auf. Das <ACK>-Signal wird verfälscht. Der Sender hat kein einwandfreies <ACK> erhalten und wiederholt daher den dritten Block.

<SOH> <03> <FC> Daten <Prüfsumme>—>

Der Empfänger hatte den dritten Block bereits erhalten und antwortet wieder mit:

<—<ACK>

Das Ende der Datei ist erreicht. Der Sender sendet ein:

<EOT>—>

Der Empfänger antwortet mit:

<—<ACK>

Die Übertragung ist beendet. Die Daten sind fehlerfrei beim Empfänger gespeichert.

Christensen (Seine Mailbox-Adresse ist: CompuServe 76703, 302.) empfiehlt, das Zeichen <CAN> (Cancel, Abbruch) nicht mehr wie bei früheren Implementationen zum Abbrechen der ganzen Übertragung zu verwenden, weil es zu leicht vorkommen kann, daß ein <ACK> oder <NAK> durch eine Leitungsstörung in ein <CAN> umgemünzt wird.

Schließlich müssen viele Timeouts für den heutigen Betrieb an weiten Datennetzen und überfüllten Host-Computern länger ausgelegt werden. Die ursprünglichen 10 Sekunden reichen oft nicht mehr.

XModem ist ein einfaches und zuverlässiges Protokoll. Es ist jedoch nicht ohne Schwächen. Insbesondere hat sich die Welt der Telekommunikation seit der Entwicklung von XModem weiterentwickelt. XModem ist nicht an alle Anforderungen, die wir heute haben, optimal angepaßt.

Eine Schwäche des Protokolls ist die Ungenauigkeit in der Dateilänge. Der letzte Datenblock enthält nämlich, genau wie auch alle vorhergehenden, 128 Daten-Bytes. Die Länge der übertragenen Datei muß aber nicht genau in den 128-Byte-Blöcken aufgehen. Der letzte Block ist also meist nicht voll.

XModem schreibt für den letzten Block nur vor, daß er auch 128 Daten-Bytes enthält. Das Übertragungsprogramm muß also gegebenenfalls diesen letzten Block auf die volle Länge auffüllen. Das empfangende Programm hat bei XModem keine Möglichkeit, die ursprüngliche genaue Dateilänge festzustellen.

In den allermeisten Fällen spielt diese Ungenauigkeit keine Rolle. Man wundert sich nur, wenn man eine übertragene Datei mit der ursprünglichen vergleicht (zum Beispiel mit dem COMP-Befehl in MS-DOS), daß dies manchmal mit der Fehlermeldung »Dateien haben unterschiedliche Länge« endet.

Ein zweites Problem ist das an vielen anderen Stellen verwendete X-ON/X-OFF-Protokoll. Dies ist eine Start-Stop-Steuerung, die sich der beiden Zeichen X-ON und X-OFF bedient. Wenn Daten zum Beispiel über Datex-P geschickt werden und das X-ON/X-OFF-Protokoll aktiv ist, dann ist die Wahrscheinlichkeit, daß die Übertragung plötzlich mitten im schönsten Datei-Transfer stehenbleibt, groß.

Probleme mit X-ON/X-OFF

Dies passiert, wenn in der zu übertragenden Datei zufällig ein X-OFF-Zeichen auftritt, aber im gleichen Block später nicht auch noch ein X-ON-Zeichen vorkommt. Die Datenübertragung bleibt dann sofort stehen und läßt sich mit keinem Mittel zum Weiterlaufen bewegen.

Die einzige Lösung dieses Problems besteht darin, das X-ON/X-OFF-Protokoll vor Beginn der XModem-Übertragung abzuschalten. Anders ausgedrückt, der Übertragungskanal muß »transparent« geschaltet werden.

Es gibt noch einige andere, ähnlich gelagerte Probleme, die sich durch Transparent-Schalten lösen lassen, zum Beispiel das Auftreten von Ctrl-P-Zeichen, die das Datex-P-Netz in den Befehlsmodus umschalten. Auch dann ist es aus mit der Datenübertragung.

Wie man den Übertragungskanal transparent schaltet, hängt vom Kanal selbst ab. Eine Telefonleitung ist immer transparent, solange sie nicht zu viele Störungen aufweist. Datex-P hat ein spezielles Kommando zum Umschalten in ein »transparentes Profil«. Wenn Sie Daten mit XModem über Datex-P übertragen wollen, dann geben Sie vor Beginn ein: Ctrl-P prof 3 und drücken Sie dann genau zweimal die Return-Taste. (Für Ctrl-P halten Sie die Ctrl-Taste fest und drücken dann kurz die Buchstabentaste P.)

Datex-P arbeitet nun transparent, das heißt, jedes Bit kommt völlig unverändert auf der Gegenseite an. Das bedeutet zwangsläufig auch, daß das gewohnte Echo verschwindet. Sie tippen also von nun an blind, aber dafür funktioniert XModem zuverlässig selbst auf den längsten interkontinentalen Strecken.

Dies führt uns gleich zum nächsten Problem mit dem XModem-Protokoll. XModem ist ursprünglich für den Betrieb auf Telefonleitungen entwickelt worden. Mit den transkontinentalen Datennetzen war man damals noch nicht so weit.

Heute stellt sich heraus, daß XModem für unsere modernen X.25-Netze, wie Datex-P, nicht besonders gut angepaßt ist. Datex-P verwendet zum Beispiel eine Paketlänge von 128 Daten-Bytes. Ein XModem-Block ist aber unglücklicherweise mit den zusätzlichen Kontrollinformationen gerade 132 Bytes lang. Für die Übertragung von 128 Byte Daten werden also zwei X.25-Pakete benötigt, von denen das zweite fast leer ist, aber voll bezahlt werden muß.

Das nächste Problem mit den X.25-Netzen ist, daß XModem nach jedem Datenblock wartet, bis die Antwort von der Gegenstelle wieder zurückgekommen ist. Bei einer Langstreckenverbindung, zum Beispiel in die USA, kann das leicht eine Sekunde dauern, weil die Datenpakete ja auf dem Hinweg und die Antworten auf dem Rückweg noch einmal einen kleinen Umweg von je 2 x 36000 km machen müssen. Dies ist nämlich der geostationäre Orbit der Kommunikations-Satelliten. Dadurch wird XModem über Datennetze langsam.

X-Modem und Datex-P

Wer glaubt, durch Wechseln von 300 auf 1200 Baud Geschwindigkeit der Datenübertragung auf die Sprünge helfen zu können, der irrt sich hier. Der Paket-Transfer braucht seine Zeit.

Natürlich kann man Daten auch sehr viel schneller über X.25-Netze übertragen. Eine Methode ist, ganz einfach längere Blöcke zu verwenden. Eine noch raffiniertere ist, auf die Antworten nicht erst zu warten, sondern immer schon einige Blöcke »auf Verdacht« im Voraus zu senden. Die beste schließlich ist es, die unsichere asynchrone Verbindung ganz zu vermeiden und sich einen X.25-Anschluß in den PC zu stecken. Alle diese Wege werden in der Tat beschritten, und eine Reihe neuerer Protokolle für die fehlerfreie Datenübertragung benutzen den einen oder anderen hier genannten Weg, um eine bessere Leistung zu erzielen.

Alternativen

Einige andere Protokolle zur fehlerfreien Datenübertragung binärer Dateien sind XModem CRC, Kermit, CompuServe B, MNP und X.PC. XModem CRC, eine Abwandlung des XModem-Protokolls, wurde von John Mahr entwickelt. Der wesentliche Unterschied ist, daß statt einer einfachen Prüfsumme ein »Cyclic Recundancy Check«, ähnlich wie beim Schreiben eines Blockes auf eine Diskette oder Festplatte, verwendet wird. Der CRC verringert die Wahrscheinlichkeit eines unentdeckten Fehlers auf einen verschwindend geringen Wert.

Die letzteren beiden, MNP und X.PC sind leistungsfähige Protokolle, die nicht nur zum Übertragen von Dateien geeignet sind, sondern den gesamten Datenverkehr kontrollieren und sichern. X.PC ist eine Variante des X.25-Protokolls, wie es zum Beispiel innerhalb des Datex-P-Netzes verwendet wird, jedoch abgeändert für den asynchronen Datenverkehr mit Personal Computern.

Auf Seite 107 finden Sie eine Übersicht populärer Kommunikatinsprogramme. Viele davon übertragen mit XModem. Bei der Auswahl eines DÜ-Programmes ist auf folgende Punkte zu achten:

— Hilfstexte sollen jederzeit einblendbar sein.
— Alle Übertragungsparameter sollen veränderbar sein — auch während eines Dialoges.
— Es soll ein Standard-Protokoll für die Übertragung binärer Dateien implementiert sein.
Es sollen Filter für die Umwandlung von 7- und 8-Bit-Zeichensätzen, speziell auch den IBM-Zeichensatz, mitgeliefert werden. Das gilt insbesondere für Programme, die kein Standardprotokoll für Binärfiles können.
— Funktionstasten sollen frei belegbar sein (für Standard-Prozeduren).
— Die Bedienung soll einfach sein (gefragt sind Menüs, Funktionstasten und gängige Tastenkombinationen zum Aufruf einer Funktion).
— Es sollte ein einfacher Editor zum Offline-Editieren von DÜ-Texten zuschaltbar sein. (H. G. Michna/sm)

Fortsetzung des Schwerpunktes »Kommunikation« auf S. 98

Glossar

asynchron auch: Start-Stop-Betrieb, Form der Datenübertragung, bei der jedes Zeichen mit einem »Startbit« beginnt und mit einem oder mehreren »Stopbits« endet, Gegenteil von »synchron«
binär im Zahlensystem mit der Basis 2 verschlüsselt, das heißt nur mit den Ziffern 0 und 1 darstellbar
Byte Gruppe von in der Regel 8 Bit, die ein Zeichen, zum Beispiel einen Buchstaben, darstellt
CP/M-Betriebssystem
CP/M-Betriebssystem Control Program/Microcomputer, am weitesten verbreitetes Betriebssystem für Microcomputer, die auf den älteren 8-Bit-Prozessoren Intel 8080, 8085 und Zilog Z80 basieren, geschrieben von Digital Research Corporation
CRC Cyclic Redundancy Check, zyklische Redundanz-Prüfzahl, die in der Regel durch eine Polynom-Formel aus einem Datenblock errechnet wird und eine sehr hohe Prüfsicherheit gewährleistet, das heißt wenn im Datenblock ein Fehler auftritt, dann wird aus diesem fehlerhaften Datenblock mit sehr hoher Wahrscheinlichkeit nicht dieselbe CRC-Prüfzahl errechnet
Ctrl Control-Taste
Datenbit Bit, aus dem sich im Gegensatz zum Startbit oder Stopbit die eigentlichen Daten zusammensetzen, in der Regel besteht ein asynchron übertragenes Byte aus einem Startbit, 7 oder 8 Datenbits, eventuell einem Paritätsbit und einem Stopbit
Datex-P In der Bundesrepublik Deutschland von der Post betriebener Teil des globalen Datennetzes nach dem X.25 Standard (paketvermittelndes Netz)
Echo Zurücksenden der eingetippten Zeichen, damit die eintippende Person auf dem Bildschirm sieht, was sie tippt
geostationärer Orbit Kreisbahn eines Satelliten, auf der der Satellit sich so mit der Erde mitdreht, daß er immer über derselben Stelle scheinbar verharrt, wird besonders für Kommunikationssatelliten verwendet, damit die Datenübertragungen nicht durch Auf- oder Untergehen des Satelliten unterbrochen werden und damit die Antennen nicht nachgeführt werden müssen
Host englisch, »Gastgeber«, hier: Computer, der elektronische Dienstleistungen über Datenfernübertragung bereitstellt
implementieren hier: Realisierung in Form eines Programmes
MS-DOS Microsoft-Disk Operating System, meistverbreitetes Betriebssystem für Microcomputer, die auf einem 16-Bit-Prozessor der Intel 8086-Familie basieren
Paritätsbit auch Prüfbit, zu einem Byte gehöriges Bit, das die Anzahl der gesetzten Bits (mit dem Wert 1) je nach Vereinbarung entweder immer gerade oder immer ungerade macht
Protokoll hier: Verfahren zur Datenübertragung, bei dem Fehler automatisch korrigiert werden
Stopbit Bit mit dem Wert 1, das bei asynchroner Übertragung einem Byte nachgestellt wird, bei Übertragungsgeschwindigkeiten bis zu 110 Bit/s werden auch 1 1/2 oder 2 Stopbits verwendet, ab 150 Bit/s in der Regel nur eines
synchron Form der Datenübertragung, bei der mehrere Bytes hintereinander ohne Unterbrechung und ohne Start- oder Stopbits gesendet werden, Gegenteil von »asynchron«
Timeout Zeitüberschreitung
virtueller Kanal unabhängiger Datenübertragungskanal, der scheinbar wie eine eigene Leitung arbeitet, sich jedoch physisch mit mehreren anderen Kanälen dieselbe Leitung teilt
X-ON/X-OFF-Protokoll verbreitetes Datenübertragungs-Protokoll, bei dem der Datenfluß mit einem X-OFF-Zeichen (Ctrl-S, dezimal 19) gestoppt und mit einem X-ON-Zeichen (Ctrl-Q, dezimal 17) wieder gestartet werden kann
X.25 Standard für paketvermittelnde Datennetze, auf dem beispielsweise das Datex-P-Netz beruht, die Daten werden in Paketen, üblicherweise zu je 128 Daten-Bytes, durch das Netz befördert

Verkaufe zum Schrottpreis:
TRS-80
Monitor
2 Laufwerke
diverses Zubehör

Wilfried Knospe
Lesserstr. 126
2000 Hamburg 70
Tel. 040/695 55 76

****** PC-FOUR *** Modell 4 Emulator *** PC-FOUR ****

Wer hat Erfahrung mit dem Modell 4 Emulator
PC-FOUR von Hypersoft ?

Gibt es einen weiteren MS-DOS Emulator für
das TRS-DOS System ?

Klaus Hermann, Tel.: 07127/71945

Verkaufe:
ZX-81, 16 KB, Handbücher: 70,-
Genie I, Centronics, Cassetten: 100,-
Sanyo-Monitor, bernstein, 18 Mhz: 150,-
TRS-80 M. I, Exp.-Interface 48 Kb, Banker 256 Kb (ohne RAMs),
HRG-1B, Zerobrenner, 3xBASF 6106 (je 1x40Track): 400,-
außerdem: Software, Handbücher, Anleitungen

Reiner Stober
Nelkenstraße 12
3216 Salzhemmendorf 4
Tel. 05153/1564

BÖRSE -- BÖRSE -- BÖRSE

Genie IIIs, Z80B-CPU, 7,2/1,77 MHz umschaltbar, 256kB RAM, 12kB ROM-Adreßraum (ein paar kB belegt; erweiterte ROM-Version auf Wunsch), nach Reset durchgehend RAM (kein ROM mehr. Speicherinhalt veränderbar), 64kB Graphik (Auflösung max. 512X512), beliebig variables Bildschirmformat (max. 2kB), ladbare Zeichensätze 8 Pixels breit (Höhe variabel. max. 16 P.). DOS komplett im Speicher, RAM-Disk 64/128kB, 2 Laufwerke 80/DS/DD (andere Formate einstellbar), Echtzeituhr/-kalender batteriegepuffert. G-DOS 2.4 (auf Wunsch erweiterte Version CALVA-DOS 2.4), CP/M 2.2, noch immer voll kompatibel mit TRS-80/Genie I usw., stark erweiterte Tastatur (z. B. 8 Funktionstasten, insg. 96 T.), bisher keine Bernhardt'schen Umbauten (schade!). Neupreis ca. 6000 Mark, Altpreis: Macht mal ein Angebot!
Freut euch nicht zu früh, es handelt sich nicht um meine Maschine, die bleibt euch noch erhalten. Diese gehört einem Freund.
Angebote bitte an Arnulf, Tel. 0451-791926

C compiler

for the model I or 3 using TRSDOS, LDOS, NEWDOS, DOSPLUS, or MULTIDOS; includes full screen text editor and advanced development package

Wer kennt diesen C-Compiler der Fa. ALCOR für NEWDOS ? 09191/31698
Nachricht bitte an Heinrich Betz

Verkaufe:
- altes Shugart-Floppy
- RS 232C und
- Accustikkoppler (dataphon s21d)
- EM 2005 (Banker)

Werner Förster
Christoph-Krebs-Str. 9
8720 Schweinfurt
☎ 09721/21841

BÖRSE -- BÖRSE -- BÖRSE

Sehr geehrter Herr Schröder,

wie ich der C'T 8/88 entnehme, beschäftigt sich Ihr Club mit den guten alten TANDY TRS-80-Rechnern.

Ich bin zwar vor einiger Zeit auf einen ATARI ST umgestiegen, besitze jedoch noch einen TRS-80, den ich nun wegen Platzmangel verkaufen möchte.

Die Anlage besteht aus:

- TRS-80, Mod.I (Bj. 80), 16kB RAM, Level II Basic Groß-/Kleinbuchstaben, eingebauter Joystickanschluß für Standartjoystick (bedient die Pfeiltasten und die Leertaste)

- Tandy-Monitor, grüner Phosphor

- CE-Computer Expansion Interface (Bj. 12/82) mit 32 kB RAM, Floppy Controller und Erweiterungs-Bus interne Erweiterungskarte: DD-Controller für Laufwerke mit doppelter Schreibdichte (Mod. III kompatibel)
- Doppelfloppygehäuse mit 2 * TEAC, 5¼", 40 Spuren, 1-seitig doppelte Schreibdichte (ca. 180 kB/Floppyseite)

- 9-Nadeldrucker Triumpf-Adler, deutscher Zeichensatz, für Einzelblatt, Endlos- und Fernschreiberpapier, 80 cps, (Bj. 83)

- ca. 80 Disketten mit Software (Pascal, Basic, Fortran, M80, FORTH, NEWDOS80, LDOS, Spiele, Anwenderprogramme, SCRIPSIT, Super-SCRIPSIT, etc.)

- ca. 30 kg Dokumentation (Hardware + Software, Zeitschriftenartikel, Schaltpläne, etc.)

Der Anschaffungspreis der Hardware lag bei ca. DM 4000.--, meine Preisvorstellung: DM 450.-- bei Selbstabholung.
Alle Komponenten arbeiten einwandfrei!

Sollten Sie sich für die Anlage interessieren, oder einen Interessenten kennen, bitte ich um Nachricht.

Mit freundlichem Gruß

Peter [signature]

Peter Büttner
Borkumer Straße 21
8500 Nürnberg 90
0911/ 38 21 36

Angebot

Hallo, Leute!

Möchtet Ihr einen Rechner mit 512 KByte RAM, 20 Megabyte Festplatte, einem Diskettenlaufwerk, mehreren Centronics- und V24-Schnittstellen für 1000 bis 2000 DM haben? Nein, es sind keine IBM-Kompatiblen, sondern Z80-Rechner auf ECB-Bus-Basis (Betriebssysteme CP/M, MP/M, OASIS). Ein Münchner Händler hat die Dinger im Keller stehen und wäre froh, sie los zu werden.

Die Rechner sind unterschiedlich ausgestattet. Voll ausgebaut sehen sie so aus: Z80 B mit 6 Mhz Taktfrequenz, 512 KByte RAM, 2 Floppy-Laufwerke (bis 1.2 MByte), 2 Festplatten (je 15-40 MByte) oder eine Festplatte und ein Streamer (entsprechende Größe), mehrere V24- und Centronics-Schnittstellen, Voll-Ausbau beim Spitzengerät auf 8 Terminals und 8 Drucker möglich. Alle Systeme bauen auf dem ECB-Bus auf und es sind noch diverse Einzel-Karten (RAM, Schnittstellen) vorhanden. Die meisten Systeme haben 15.000 bis 25.000 DM gekostet, sind entweder neu oder liefen in Zahnarztpraxen im Mehrbenutzer-Betrieb. Entsprechend sind mehrere Drucker und Terminals anschließbar und auch noch vorhanden. Als Betriebssysteme stehen CP/M, MP/M (Mehrbenutzer-Variante von CP/M) und OASIS (Mehrbenutzer-Betriebssystem) zur Verfügung, außerdem zumindest ein Textverarbeitungsprogramm mit Datenbankfunktionen und evtl. noch Büro-Software. Außerdem Basic, Cobol und Assembler.

Der Mensch kann uns noch Tandberg-Terminals (neu: ca. 7000 DM) für 200 DM, mehrere V24-Terminals und Typenraddrucker anbieten. Die Dinger können auch als Zugabe zu einem MP/M-/OASIS-Rechner geliefert werden, wenn wir das absprechen.

Es gibt drei Rechner-Varianten (Angaben ohne Gewähr):
Junior mit Z80 B, 256/512 KB, Floppy, 15-20 MB Festplatte, 2-4 V24-Schnittstellen; ca. 1000 DM.
16000 mit Z80B, 256/512 Kb, Floppy, Festplatte 20-40 MB, Streamer, 2-8 Schnittstellen (=Terminals=Benutzer!); unter 2000 DM.
ONIX (wird vielleicht anders geschrieben) mit Floppy, Magnetband, Streamer, 4 Schnittstellen; Preis ???

Ich habe diese Sache nur kurz am Telefon besprochen, leider ist dies inzwischen (2.5.89) über drei Monate her. Eigentlich sollte dieses Angebot mit dem letzten Info erscheinen, aber das klappte wohl nicht.

Es sind scheinbar 8 ONIX-, 10 Junior- und 30 16000-Rechner dort gelagert und verstopfen nur den Platz. Der Mensch möchte nur den Materialwert (Festplatten, Floppys, RAM) haben und danach nichts mehr von den Dingern wissen, versichert aber, daß alles im besten Zustand ist. Wenn sich mehrere von uns finden, die gemeinsam den ganzen Ramsch oder den größten Teil nehmen, will er auf den Preis keine Rücksicht mehr nehmen. Wenn Ihr Interesse habt, meldet Euch bei mir. Ich glaube, so ein Angebot bekommt Ihr nicht noch einmal. Allein der Materialwert ist höher als die Preise. Und im Sechserpack wird's billiger! Ich schätze, bei 15 bis 25 Interessenten je 1000 DM bekommen wir den ganzen Kram. Wer weiß?!

Gerald Schröder
(04105/2602)

Welche Sprache für wen ?

Aus einer Mailbox: (hoffentlich kennt zumindest jeder einen, der Englisch kann)

"humor": What's your favourite? (about 100 lines)
--- Mo, 28.11.88 / 00:22:23 von DOPPELKEKS

There are so many programming languages available that it can be very difficult to get to know them all well enough to pick the right one for you. On the other hand most men know what kind of woman appeals to them. So here is a handy guide for many of the popular programming languages that describes what kind of women they would be if programming languages were women.

Assembler - A female track star who holds all the world speed records. She is hard and bumpy, and so is not that pleasant to embrace. She can cook up any meal, but needs a complete and detailed recipe. She is not beautiful or educated, and speaks in monosyllables like "MOV, JUMP, INC". She has a fierce and violent temper that make her the choice of last resort.

FORTRAN - Your grey-haired grandmother. People make fun of her just because she is old, but if you take the time to listen, you can learn from her experiences and her mistakes. During her lifetime she has acquired many useful skills in sewing and cooking (subroutine libraries) that no younger women can match, so be thankful she is still around. She has a notoriously bad temper and when angered will start yelling and throwing dishes. It was mostly her bad temper that made grandad search for another wife.

COBOL - A plump secretary. She talks far too much, and most of what she says can be ignored. She works hard and long hours, but can't handle really complicated jobs. She has a short and unpredictable temper, so no one really likes working with her. She can cook meals for a huge family, but only knows bland recipes.

BASIC - The horny divorcee that lives next door. Her specialty is seducing young boys and it seems she is always readily available for them. She teaches them many amazing things, or at least they seem amazing because it is their first experience. She is not that young herself, but because she was their first lover the boys always remember her fondly. Her cooking and sewing skills are mediocre, but largely irrelevant, its the frolicking that the boys like.
The opinion that adults have of Mrs. BASIC is varied. Shockingly, some fathers actually introduce their own sons to this immoral woman! But generally the more righteous adults try to correct the badly influenced young men by introducing them to well behaved women like Miss Pascal.

PL/I - A bordello madam. She wears silk dresses, diamonds, furs and red high heels. At one time she seemed very attractive, but now she just seems overweight and tacky. Tastes change.

C - A lady executive. An avid jogger, very healthy, and not too talkative. Is an good cook if you like spicy food. Unless you double check everything you say (through LINT) you can unleash her fierce temper. Her daughter C++ is still quite young and prone to tantrums, but it seems that she will grow up into a fine young woman of milder temper and more sophisticated character.

ALGOL 60 - Your fathers wartime sweetheart, petite, well proportioned, and sweet tempered. She disappeared mysteriously during the war, but your dad still talks about her shapely form and their steamy romance. He never actually tasted much of her cooking, but they did exchange simple recipes by mail.

Pascal - A grammar school teacher, and Algol 60's younger sister. Like her sister she is petite and attractive, but very bossy. She is a good cook but only if the recipe requires no more than one pot (module).

Modula II - A high-school teacher and Pascal's daughter. Very much like her mother, but she has learned to cook with more than one pot.

ALGOL 68 - Algol 60's niece. A high-society woman, well educated and terse. Few men can fully understand her when she talks, and her former lovers still discuss her mysterious personality. She is very choosy about her romances and won't take just any man as her lover. She hasn't been seen lately, and rumor has it that she died in a fall from an ivory tower.

LISP - She is an aging beatnik, who lives in a rural commune with her hippie cousins SMALLTALK and FORTH. Many men (mostly college students) who have visited the farmhouse, enthusiastically praise the natural food, and perpetual love-ins that take place there. Others criticize the long cooking times, and the abnormal sexual postures (prefix and postfix). Although these women seldom have full-time jobs, when they do work, their employers praise them for their imagination, but usually not for their efficiency.

APL - A fancy caterer specializing in Greek food. She can cook delicious meals for rows and rows of tables with dozens of people at each table. She doesn't talk much, as that would just slow her work down. Few people can understand her recipes, since they are in a foreign language, and are all recorded in mirror writing.

LOGO - A grade-school art teacher. She is just the kind of teacher that you wish you had when you were young. She is shapely and patient, but not an interesting conversationalist. She can cook up delicious kiddie snacks, but not full-course meals.

LUCID & PROLOG - These clever teenagers show a new kind of cooking skill. They can cook-up fine meals without the use of recipes, working solely from a description of the desired meal (declarative cooking). Many men are fascinated by this and have already proposed marriage. Others complain that the girls work very slowly, and that often the description of the meal must be just as long as a recipe would be. It is hard to predict what these girls will be like when they are fully mature.

Ada - A WAC colonel built like an amazon. She is always setting strict rules, but if you follow them, she keeps her temper. She is quite talkative, always spouting army regulations, and using obscure military

Alexander Schmid

- I M P R E S S U M -

1. Vorsitzender	Alexander SCHMID St.Cajetan Str. 38 VII 8000 München 80 ☎ 089 /495326
2. Vorsitzender	Horst-Dieter SCHROERS Breslauer Str. 9 8016 Feldkirchen ☎ 089 /9032615
Hardwarekoordinator	Andreas MAGNUS Pommernstr. 4 4650 Gelsenkirchen ☎ 0209 /870230
NEWDOS-Diskothekar	Oliver VOLZ Waldburgstr. 73 7000 Stuttgart 80 ☎ 0711 /7353817
CP/M -Diskothekar	Rüdiger SöRENSEN Thomas-Mann-Str. 3a 6500 Mainz 1 ☎ 06131 /32860
Club-Bücherei	Christian MENK Ollsener Str. 52 2116 Hanstedt ☎ 04189 /78251
Redaktion	Jens NEUEDER Gschlachtenbretzingen Rudolf-Then-Straße 32 7178 Michelbach /Bilz ☎ 0791 /42877
Autoren	Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.
Bankverbindung	des CLUB 80

Postgirokonto

	Sonderkonto CLUB 80
Konto-Nummer	285 491 - 465
Postgiroamt	Dortmund
BLZ	440 100 46

Eine Zensur oder Kontrolle der Infobeiträge erfolgt nicht.
Die Redaktion

SCHLUss

Hallo Club-80er,

nachdem wir im April/Mai unsere Jahreshauptversammlung hatten, erscheint nun unser neuestes Clubinfo mit neu kreiertem Deckblatt.
Unser Clubemblem ist als neues Erkennungszeichen eingearbeitet.

Auch sonst hat sich einiges getan.
Der Vorstand ist "erneuert" -ihr habt es sicher im Vorwort gelesen- und ein neuer Weg für den Druck des INFO's wurde gefunden. Hoffen wir, daß es diesmal besser funktioniert als in letzter Zeit.

Eine weitere "Gute Nachricht" für Euch wird wohl das Erscheinen von acht Sonderausgaben sein. Die ersten zwei --
NEWDOS-Diskothek's-Verzeichnis
und
CLUB-80-Büchereiliste
-- liegen diesem Info bei.

Nun aber noch einen Aufruf an Euch:
Wie Ihr aus diesem INFO ersehen könnt fehlt es an "kleinen Artikeln". So ca. 1 bis 4 Seiten lange hatten wir bisher immer genügend, nun aber fehlen genau solche! Ich hatte Euch die "Artikelflaute" ja schon im 26. INFO angekündigt.
Sicher habt Ihr irgendwo noch einen "kurzen Artikel" für unser Clubinfo. Ich bitte Euch ihn mir zuzusenden, damit unser Clubinfo auch weiterhin regelmäßig erscheinen kann.
Wie Ihr wißt ist alles zum Thema Computer interessant. Sei es nun über Hard-Software oder irgendwas anderes Computerspezifisches. Ich rechne mit Eurer Mitarbeit.

Bis auf bald, zum 28. INFO und einigen Sonderheften.

Es grüßt Euch Euer

Jens

HEFT
27
Juli
1989

P.S.: Nicht vergessen - Dein Beitrag zum INFO !!

HARDWARE * SOFTWARE

COMPUTER-
CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

TEXT DATEI

CLUBINFO

28. AUSGABE

KONTAKTADRESSE : CLUB 80 / ALEXANDER SCHMID / ST. CAJETAN STR. 38 VII / 8000 MÜNCHEN 80
TEL.: 089 / 495326

INHALTSVERZEICHNIS

Seite: und Autor:

Clubinternes

Neues vom Vorstand 01
Die Leiden des jungen Helmuts 01
Alexander Schmid
Alle Jahre wieder .... 02 - 03
Andreas Magnus
Fröhlichen Geburtstag ! .... 03 - 04
Jutta Obermann
Termine 04
Redaktion
Vorstellungen .... 05 - 09
Christof,Jürgen,Harald,Wilhelm,Stefan,Norbert

Software

Ein letzter Zehnzeiler 10
Nellemakello 11
Sitzverteilung nach d'HONDT 12
Ordnung muß sein ! .... 13 - 16
PASCAL + GALTON .... 17 - 19
Klaus-Jürgen Mühlenbein
Startprogramm für BASIC-Programme .. 21
Dateiprogramm .... 22 - 24
Zum Schluß zur Speicherbelegung .... 25 - 26
Oskar Drechsler
Geheimnisse eines Zeichengenerators . 27 - 31
Norbert Giese
Ordnung muß sein .... 33 - 34
Arnulf Sopp
Wordstar auf allen Laufwerken ...... 35 - 36
Frank-Michael Schober
Teamwork 37
Alexander Schmid
Tscrips retten 38
Vergleich BASIC mit TURBO-PASCAL ... 39 - 43
Klaus-Jürgen Mühlenbein
Backen und Banken .... 44 - 46
Nochmaliges Banken .... 47 - 51
Unterbrechung gestattet ? 52
ZCPR33 -- die Vierte .... 53 - 57
Rüdiger Sörensen
Prommer 80 unter G-DOS .... 58 - 63
Andreas Magnus
Fensterln mit CP/M .... 64 - 66
Gerald Schröder

Seite: und Autor:

Hardware

Der flüsternde Portable .... 67 - 68
Heinrich Betz
Umlaute auf der Tastatur des Genie I 69 - 70
Oskar Drechsler
Ein klebriges Problem 71
Alexander Schmid
Star NL10 abwechselnd am CP/M-R. +PC 72
Eleganter Adreßdecoder für GENIE I 73 - 77
Es lebe der Standart ! .... 78 - 99
Helmut Bernhardt

Börse

Wer hat was -- wer will was .... 99 - 102

Sonstiges

Teuer, aber überflüssig 48
Artikel aus Ich hasse COMPUTER!
Mikrocomputer-Betriebssysteme - .... 103 - 105
Gerald Schröder
Die Betriebsart Packet-Radio (PR) .. 111 - 116
Hans-Günther Hartmann

Programmbibliothek

Ein neuer Diskothekar meldet sich .. 106
Oliver Volz
Neues aus der CP/M - PD .... 107 - 108
Programm des Monats: BRADFORD ...... 108 - 109
Rüdiger Sörensen

Die letzten Seiten

Impressum 117
Schluß 118
Redaktion
Mitgliederadressenliste .... am INFO-Ende
Alexander Schmid
BIOS / BDOS .... am INFO-Ende

Neues vom Vorstand

Vielleicht hat der eine oder andere von Euch unsere Anzeige in der c't gelesen. Bis jetzt haben sich immerhin acht Interessierte gemeldet, aber nur einer, Detlef Behrendt, hat schon den Sprung ins kalte Wasser gewagt. Er hat ein Genie I mit zwei Laufwerken und einen NEC P6. Die Amateurfunker können sich schon mal auf einen Neuen in Ihren Reihen freuen, das steht nämlich u.a. auch bei seinen Interessen. Mit Hardware- und C-Begeisterten wird er sich wahrscheinlich auch gut verstehen. Herzlich willkommen unter den letzten der Tafelrunde.

Die Infos lassen immer länger auf sich warten und ich danke Euch, daß soviel Vertrauen in den Club habt und weder ausgetreten seid, noch mir eine Bombe geschickt habt. Ich glaube, Ihr werdet DIESEN Grund aber mit Sicherheit verstehen und nicht doch noch auf die Barrkiaden gehen. Unser Jens hat nämlich geheiratet und hat sich in den Flitterwochen mit Recht um alles andere als um den Club gekümmert. Auch hierzu nochmal offiziell herzlichen Glückwunsch und alles Gute. Töpfe, Pfannen und Socken wird er inzwischen wohl genug haben, über rein schriftliche Beileidsbekundungen freut er sich bestimmt genauso.

über Heinrich Betz haben wir jetzt endlich eine Möglichkeit gefunden, das Info zu einem annehmbaren Preis drucken zu lassen. Ohne 'höhere Gewalt' wäre das Info also pünktlich erschienen und das nächste tut das dann wohl hoffentlich mal wieder. Wenn nicht, möchte ich Euch schon mal fröhliche Weihnachten und ein gutes Neues Jahr wünschen.

Soviel vorläufig vom Guru.

Die Leiden des jungen Helmuts

Unser allseits bekannter und beliebter Helmut Bernhardt hat mir in seinem letzten Brief sein Leid über eine leider scheinbar weit verbreitete Unsitte im Club geklagt. Da schreibt er als einer der wenigen wirklich Aktiven viele viele interessante Artikel und bekommt zum Dank viele viele Anfragen, auf die er stundenlang antworten muß (was er, wie er sagt, gerne tut), aber auf die er außerdem noch aus der eigenen Tasche das Porto draufkleben darf. Findet Ihr das fair ? Da wird er auch noch dafür bestraft, daß er was für's Info und den Club tut. Was würdet Ihr davon halten, wenigstens das Rückporto beizulegen ? Vom Arbeitsaufwand spricht ja gar keiner (den könntet Ihr bei Helmuts Durst sowieso nicht bezahlen), aber bei den vielen Anfragen kommen für ihn leicht zwei bis drei große Blaue im Jahr zusammen. Als Alternative könnte der Club natürlich in solchen Fällen die Portokosten übernehmen, oder den Beitrag nach der Menge der geschriebenen Artikel berechnen. In beiden Fällen würden die Beiträge wohl für die meisten Mitglieder eher steigen als sinken. Überlegt es Euch mal, wie wir das regeln könnten. Um Meinungen wird gebeten.

Euer Erster

Alle Jahre wieder

gibt es neue Gesichter im Vorstand des Club 80.
So ist es auch dieses Jahr nicht anders. Apropos Gesicht, da ich erst jetzt dazu gekommen bin zu schreiben wird dieser Text erst im Info NR. 28 erscheinen, so das ihr mein Gesicht schon kennt. Nun aber zu meiner Vorstellung. Ich bin der neue Hardware-Koordinator des Club's. Wie der Name schon sagt, soll ich Hardware koordinieren.(was immer das heissen mag.) Aber dazu später. Erst zu meiner Person. Ich bin erst seit Januar im Club so das viele mich noch nicht kennen. Da ich nicht mehr weis was ich in meiner Vorstellung als neues Clubmitglied geschrieben habe hier noch einmal das wichtigste, meine Hardware. Ich besitze ein Genie IIIs mit 4 Laufwerken sowie einer 20 MB-Festplatte und einer 10-MB in vorbereitung. Das Gerät hat 1 MB Speicher sowie eine 256 KB Grafik. Als Betriebssystem benutze ich G-Dos (Calva-Dos von Arnulf) sowie CP/M.
Außerdem habe ich noch ein Genie II mit 2 Laufwerken (alles 80 Trk), 80-Zeichen-Karte und HRG B1 sowie einigen Erweiterungen (der G II befindet sich in einem PC-Gehäuse und der IIIs in einem Tower.)

Der Haupteinsatz meines Rechners besteht darin Hardwareänderungen einzubauen und die Software dazu zu schreiben. Wobei die Software allerdings überwiegt.Klingt zwar blöd als Hardware-Koordinator aber Software zu schreiben ist billiger. (Ach so: Ich schreibe fast ausschließlich in Assembler) Also wenn Ihr Probleme habt könnt Ihr euch in beiden Fällen an mich wenden.

Nun aber zu meiner Aufgabe im Club.
Erstens: Wenn Ihr irgendwelche Vorschläge zu Hardware-Projekten habt, die bitte sofort an mich. Ich werde dann die Sache ans laufen bringen.
Zweitens: Es ist vor seeehr langer Zeit mal ein ECB-Projekt angefangen worden, welches dann im Sande verlaufen ist. Ich finde das schade denn es ist eine schöne Sache wenn unser Club etwas eigenes hat auf das wir stolz sein können. Es ist abzusehen das dieses Projekt, etwas moderner, aber trotzdem gut, wieder aufgenommen werden kann. Und diesmal sind die, die sich mit der Herstellung der Platienen befassen zuverlässiger als damals. Die genaue Vorstellung wird Helmut geben wenn es soweit ist. Aber Trotzdem schon jetzt meine Frage: Besteht noch (oder erneut) Interesse an so einem Projekt ?
Wenn ja dann sofort bei mir melden (Postkarte genügt). Dann kann ich sehen ob es sich lohnt das Projekt wieder aufzunehmen was ich mir sehr wünsche.(Ich bin auf jeden Fall dabei).

Soweit dazu. Ich selber habe auch meine Vorstellungen.
Da wäre zum einen eine Grafikkarte die's in sich hat. Ich stelle mir da vor: 512 X 512 Punkte Farbgrafik, eigenes RAM sowie ein eigener Prozessor der die Sache Koordiniert. Der Vorteil ist, die Karte wird über Ports gesteuert so das die Software für alle Rechner gleich sein wird.
Außerdem Spiele ich mit dem Gedanken eines Video-Digitalisierers. Eventuell in Verbindung mit der Grafikkarte.
Und eine Sprach-Ein-Ausgabe wäre ein schönes Projekt was zu realisieren Spaß machen würde.

Also Schreibt mir was das Herz begehrt. Ich werde mich um den Rest kümmern.

Noch etwas in anderer Sache. In einem der letzten Info's, ich weiß nicht mehr in welchem, fragte Arnulf(Hallo Arnulf) nach einer G IIIs Ecke, ich wäre sehr ineressiert zumal sich die Anzahl der G IIIs Besitzer mit meinem Eintritt um grob gerechnet 100 % erhöht hat.(das sagt der zweitschönste G-IIIs-Besitzer).

Andreas Magnus

F r ö h l i c h e n   G e b u r t s t a g !

Hallo Ihr Freunde des CLUB 80,

wieder einmal ist es soweit. Das Info ist erschienen und es kann losgehen mit den vielen Grüßen und Wünschen zum Geburtstag.
Auch diesmal wieder appelliere ich an Euch, schreibt mir, wenn Ihr Euch auf der nachfolgenden Liste vermißt. Es hat sich bisher zwar noch niemand beschwert, daß er hier noch nicht vertreten war, doch ich bin sicher, daß ich nicht von allen Mitgliedern die Daten habe.

Meine zweite Bitte lege ich Euch besonders nahe.

L e u t e , s c h r e i b t   A r t i k e l

Wie ich Euch im letzten Info schon erklärt habe, besteht das Info nicht nur aus Witzen und Terminen, sondern vor allem aus E u r e n Artikeln.

So, aber nun zum eigentlichen Grund dieses Artikels.

Im Monat S e p t e m b e r hatten Geburtstag:

Rychlik Andreas 11.09.

Sopp Arnulf 23.09.

Im Monat O k t o b e r haben Geburtstag:

Meklenburg Heinz Dieter 01.10.

Magnus Andreas 17.10.

Held Manfred 30.10.

Im Monat N o v e m b e r haben Geburtstag:

Krispin Michael 02.11.

Hentz Werner 03.11.

Mühlenbein Klaus-Jürgen 07.11.

Schäfer Walter 03.11.

Jetzt wird das Jahr noch voll gemacht.

Im Monat D e z e m b e r hat Geburtstag:

Rinio Gerd 03.12.

Der Vorstand gratuliert allen genannten und ungenannten, hier aufgeführten und nicht aufgeführten, freudigen und nicht freudigen "Geburtstagskindern" ein fröhliches Wiegenfest und alles Gute auch weiterhin.

Eure Jutta

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß .... Mitte / Ende Dezember

Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!

Hallo liebe Clubmitglieder,

als neues Clubmitglied möchte ich mich kurz vorstellen.

Ich heiße Christof Neumann, bin 28 Jahre alt und von Beruf Bauingenieur (Wasserwirtschaft/Tiefbau).

Durch den Beruf kam ich durch Zufall günstig zu einem **Tandy Model II**, der mir mit seinen zwei 8"-Laufwerken und einem dazuerworbenen Typenraddrucker (**JUKI 6000**) seit 1987 hauptsächlich für die Textverarbeitung gute Dienste leistete. Hierzu verwende ich das gute alte Wordstar (gepatcht für den Juki-Drucker) unter CP/M.

Eigene Programme erstelle ich in MBASIC; ich würde aber auch gern in PASCAL oder C programmieren, wenn ich es für das Model II bekommen kann.

Nachdem seit Mitte '88 der Rechner defekt war, hätte ich ihn in meiner Verzweifelung (denn niemand konnte mir helfen) fast zum Mond, oder doch zumindest in die Wüste geschickt, bis ich durch eine kleine Anzeige auf den 'CLUB 80' aufmerksam wurde. So kam es, daß er seit Anfang Mai durch tatkräftige Hilfe aus dem Club wieder läuft, und wenn er nicht gestorben ist ....

Mit dem Wunsch auf gute Zusammenarbeit grüße ich alle Clubmitglieder!

C. Neumann

Hallo Jens,

ich möchte mich hiermit dir und den anderen Clubmitgliedern vorstellen:

Name:	Kemmer Jürgen
Alter:	19 Jahre
Adresse:	Wohnheim: (nur Werktags) 1. Rangierbahnhof Bau 3 8500 Nürnberg 50 Tel.: 0911 / 48 20 90 2. normaler Wohnort: Dorfberg 7 8701 Sulzdorf Tel.: 09334 / 10 50
Beschäftigung:	Ausbildung zum Informationselektroniker in Nürnberg bei der DB (bis nächstes Jahr).
Hobby's:	Computern, Lesen (meistens Science Fiction) Radio Hören, Experimentieren mit Eletronik
Hardware:	Schneider CPC 464 (64k) 9-Nadel-Drucker (EPSON LX-800) 3'-Floppy mit 180K (Schneider-System-Laufwerk) 5 1/4'-Zweitlaufwerk mit 180K Fischertechnik Computing Interface: 4 Motorausgänge,8 Digital- u. 2 Analogeingänge D/A-,A/D-Wandlerkarte fuer PIO-Karte Eprom-Karte für 8 Steckplätze

Hallo Club 80-Freunde,

seid gegrüßt und gestattet, daß ich mich vorstelle:
Ich heiße Harald Hirsekorn, gehöre zum Jahrgang 1927, wohne in 2359 Henstedt-Ulzburg und übe den Beruf eines Elektro-Mechanikers aus.

Vor drei Jahren habe ich mir einen Sharp MZ-3541 angeschafft. Seitdem versuche ich, einigermaßen vernünftige Programme unter CP/M zu schreiben. Der Erfolg ist allerdings recht dürftig. Vielleicht verhilft mir der Kontakt zum Club-80 und seinen Mitgliedern dazu, meine Programmierkenntnisse zu verbessern.

Mit freundlichen Grüßen
Harald Hirsekorn

Harald Hirsekorn

Hallo Clubfreunde, -kameraden, -mitglieder,

als relativ neues Clubmitglied (ab Heft Nr, 23) möchte ich mich stichwortartig vorstellen:
Jahrgang 1938, Berufssoldat, Techniker, verheiratet, 2 Kinder m.

Auf den "Club" gekommen durch Helmut Bernhardt bei einem CP/M-Usertreffen in Kiel. Durch aktive "Hardware-Tips" von Hartmut Obermann und Harald Mand sowie der kameradschaftlichen Stimmung und spontaner Hilfsbereitschaft beim "Nordlichtertreffen" zum Dabeibleiben verpflichtet.

Meine Anlagen:

- Speedmaster 5.3 (Genie II S baugleich) mit 1x 40 Trk SS, 1x 40 Trk DS, 1x 80 Trk SS 5.25" Laufwerke und ein 8" DS Laufwerk sowie ITOH 8510 A Drucker und Akustikkoppler.
- TRS-80 Mod. 4/P mit 2 umschaltbaren 40/80 Trk DS 5.25" Laufwerken und Typenraddrucker Juki 2200.
- ATARI 1040 ST mit zusätzlichem 40/80 Trk 5.25" Laufwerk und einen STAR LC 24-10 Drucker.
- Software für alle Rechner satt.

Aufgrund meines zweiten Steckenpferdes, das der Astronomie, nutze ich meine Rechner mit Astronomieprogrammen ernsthaft, als Schreibmaschine privat als auch dienstlich und zum Spaß als Spielkamerad.

Vom Club erwarte ich, daß er weiterhin bestehen bleibt und sich auch andere (mich eingesclossen) an der Veröffentlichungen von Beiträgen für unsere Clubzeitung aufraffen. Denn dann bleiben auch diejenigen bei der Stange, die bisher alles "gemanagert" haben.

In diesem Sinne Euer

W. Tornow

(Wilhelm Tornow)

Hallo Clubmitglieder

Ich möchte mich kurz vorstellen.

Mein Name ist Stefan Nitschke und wohne in Walzbachtal 1 (bei Karlsruhe). Ich bin 25 Jahre alt und studiere Physik an der Universität Karlsruhe.

Zu CP/M kam ich durch den günstigen Einkauf eines ECB-Bus Rechners Anfang diese Jahres. Es stellte sich jedoch, für mich, als schwierig heraus, Software für CP/M-Rechner zu bekommen, da meine Bekannten Atari ST oder MSDOS Rechner benutzen.

Das Betriebssytem meines Rechners (ZDOS) ist CP/M 2.2 kompatibel. Meine Aktivitäten am Rechner beschränken sich derzeit, wegen Softwaremangel, auf Textverarbeitung mittels Wordstar.

Interresse habe ich an fast allen (Höheren)Programmiersprachen, und Hardwareprojekten. So würde mich das Einbinden einer Harddisk (Soft/Hardware), vielleicht mittels eines RSM-Modules, welches während des booten eingeladen und anschließend aktiviert wird, desweiteren das Betriebssystem MP/M und Numerik interessieren. Für Numerikprogramme habe ich hier noch einen Arithmetikprozessor 8231A von Intel herumliegen, welcher auf Anschluß wartet.
Ich selbst kenne CP/M zu wenig um eigene Hard/Software Projekte zu starten.

Mein Rechner besteht aus:

CPU-Karte	: Z80 mit 4Mhz Takt von Janich+Klass HKM 105X
Floppycontr.	: JK 82 FDC 8/5 ebenfalls von Janich+Klass
Speicher	: 64 KBytes Eigenbau, 2 x 5.25'' 40-Spur-Laufwerke
Terminal	: Infoton nicht graphikfähig
Drucker	: Speedy 100-80

Mit Grüßen

Stefan Nitschke

Hallo Clubfreunde,

als weiterer CP/M-Benutzer möchte ich meine Person und meine hobbybezogene Vergangenheit und Zukunft vorstellen.

Mein Name ist Norbert Giese, 38 Jahre alt, und ich wohne in Gärtringen in der Nähe von Böblingen. Von der Herkunft bin ich eher ein "Nordlicht". Gelernter Starkströmer (etwas zum Anfassen), Elektroniker (lang ist es her), Außendiensttechniker für "etwas größere Computer", Planer (da muß man seine Fehler nicht selber ausbaden), heute Technischer Schreiber (Papier kann sich nicht wehren); kurz gesagt, vom Seitenschneider zum elektronischen Bleistift.

Zum Hobby der Computerei bin ich über einen programmierbaren Taschenrechner (TI-59) etwa 1982 zu einem SHARP MZ-80B nebst originaler Peripherie gekommen. Nachdem ich grob verstanden hatte, was solch ein Ding macht, folgte CP/M 2.2 und diverse Ausflüge in verschiedene Programmiersprachen. Dann wollte ich auch noch wissen, wie ein solches Ding arbeitet und griff wieder zum schon verstaubten Lötkolben. Eine selbstentwickelte doppelte serielle Schnittstellenkarte für den MZ-80B wurde gebaut, ein Akustikkoppler und Ausflüge in diverse Mailboxen folgten. EPROMs wurden geändert, das BIOS wurde vollständig disassembliert, kommentiert, und geändert.

Nun liegt noch eine ECBus-1Mbyte-RAM-Karte auf dem Tisch und muß angeschlossen und ins BIOS implementiert werden, eine Typenradschreibmaschine wartet noch darauf, als "LST:" angesprochen zu werden, eines der beiden 40-Spur Laufwerke soll noch zwei 80-Spur Slimline-Laufwerken weichen, die Graphikkarten des MZ-80B sind auch noch ungenutzt, und, und, und.....

Da dies alles für mich nur ein Hobby ist, sind alle in der Vergangenheit aufkommenden Sehnsüchte nach einer hochintegrierten, lötkolbenfeindlichen Hochleistungs-Produktionsmaschine bisher erfolgreich unterdrückt worden.

Weitere Hobbys: Fotografie mit Schwerpunkt Bildgestaltung, um die andere Gehirnhälfte auch etwas zu trainieren, regelmäßig Standard- und Latein-Tanz zusammen mit der "besseren Hälfte", als ausgleichende Gerechtigkeit für die langen Nächte vor dem Computer.

Ich freue mich auf eine kreative Zusammenarbeit.

* Ein letzter Zehnzeiler *

(Determinanten-Berechnung)

Hallo, Betriebs- und Volkswirte und die es werden wollen! Leider kommt ihr in der Berufspraxis nicht ganz um Mathe und schon gar nicht um die Berechnung von Matrizen (in der Tat: ohne t!) und Determinanten herum.

Nachdem sich einer meiner Schüler eine Zeitlang mit dieser endlosen Zahlenrechnerei herumgequält hatte, baute ich ihm hierfür ein Programmchen - wohl wissend, daß es hierfür natürlich längst eine Anzahl Programme gibt; doch wem das Programmieren ein lieber, aber auch nützlicher Zeitvertreib ist (sogar in BASIC!), der gibt sich erst zufrieden, wenn er eine Aufgabe selbander gelöst hat, und holt nicht Fertiges vom Markt bei ("hob-by" kommt eben nicht von "bei-holen" <Schiffahrt>, sondern von "hob<el>-by-yourself"...)

Der Algorithmus hat einige Tücken. Das Programm mußte deshalb durch eine Anzahl variantenreicher Versuche mit mehrreihigen Determinanten von Hand getestet werden; ich probierte drei- bis fünfreihige aus. Das Ergebis findet ihr nachstehend. Man kommt mit einer einzigen "Subroutine" aus, nämlich für den Fall, daß sich in der Hauptdiagonalen eine Null befindet. Und es geht genau so schnell wie in "TURBO-PASCAL". Wer es nicht glaubt, probiere es aus! (Für den Zeitvergleich bei einer 20-reihigen Determinante will ich mich allerdings nicht verbürgen! Aber wer braucht die schon!)
(Für die Mathematiker: Es wurde nicht der "Laplace'sche Entwicklungssatz" angewendet, sondern das "Dreiecksverfahren", wobei alle Elemente unterhalb der Hauptdiagonalen zu Null gemacht werden und das Produkt der dabei entstandenen neuen Elemente der Hauptdiagonalen den Wert der Determinante ergibt. Letztere Methode finde ich eleganter; sie führt auch schneller zum Ziel.)

Der Eleganz des "Dreiecksverfahrens" ist es auch zu verdanken, daß sich das Programm letztenendes in ~~10~~ 7 Zeilen unterbringen ließ, wie ihr seht (die Kommentare natürlich nicht mitgezählt). Die Flasche guten Pfälzer Weins dafür (vgl. das uralte Preisausschreiben) habe ich mir selbst spendiert. Eine ebensolche spendiere ich demjenigen, der es (in BASIC) kürzer macht...
Gruß! - *Ka-Det*

Und hier das Programm:
(10 Druckzeilen - aber nur 7 Programmzeilen!)

```
10 CLS:DEFINTI-Z:INPUT"Anzahl Reihen der Determinante ";R:DIMA(R,R):PRIN
T:PRINT"Eingabe :":PRINT:FORZ=1TOR:PRINT"Zeile";Z:FORS=1TOR:PRINT"   Spa
lte";S;":";:INPUTA(Z,S):NEXTS:PRINT:NEXTZ
20 '
30 'Spalten unterhalb der Hauptdiagonalen nullen:
40 '
50 FORS=1TOR-1:IFA(S,S)=0THENGOSUB150
60 FORJ=0TOR-S-1:F=A(S+J+1,S)/A(S,S):A(S+J+1,S)=0:FORI=1TOR-S:A(S+J+1,S+
I)=A(S+J+1,S+I)-F*A(S,S+I):NEXTI,J,S
70 '
80 'Berechnung der Determinante:
90 '
100 D=1:FORI=1TOR:D=A(I,I)*D:NEXT:PRINT
110 PRINT"Die Determinante hat den Wert D = ";D*(-1)ÄX:PRINT:PRINT:END
120 '
130 'Subroutine "Zeilentausch"
140 '
150 FORI=S+1TOR:IFA(I,S)<>0THENGOTO160ELSENEXTI
160 FORU=1TOR:CMD"F=SWAP",A(S,U),A(I,U):NEXT:X=2*(I-S)-1:RETURN
```

* * * NELLEMAKELLO * * *

(Ein Beitrag zur Europa-Wahl!)

Wieviele Sitze erhält der CLUB 80 im Europa-Parlament? Wird er sich gegen andere Clubs, Vereine, Lobbys, Parteien und was es sonst noch an Interessengemeinheiten gibt durchsetzen können?
Das ist eine schwerwiegende Frage! Wir wissen nur, daß wir N=70 Stimmen erhalten (Stand Januar 89) - nämlich die unseren! Die "Konkurrenz" scheint mir (was die Masse betrifft) stärker... "Köpfchen" zählen leider nicht nach unserem Recht. So läßt sich's leichter zählen. Sonst müßte man unsere Mitgliederzahl ja "hochnehmen"... (auf den Arm?? Gott bewahre! Ich meinte natürlich: "potenzieren".)
Doch auch den Kleinen soll und kann geholfen werden.
Dafür sorgte das Köpfchen eines Belgiers namens *Victor d'HONDT*. Er erfand 1882 das
<u>Höchstzahlverfahren.</u>
Es fand Eingang in das Bundeswahlgesetz §6, Abs.1. Ich will es euch und mir ersparen, die "Erläuterungen" dazu zu zitieren. Vielleicht verstehst du, lieber Leser, meine folgende:
Die auf die Parteien (oder Personen) auf einer Wahlvorschlagsliste entfallenen Stimmenzahlen werden fortlaufend durch die Folge der natürlichen Zahlen (1,2,3,4,...) dividiert. Dabei ergeben sich immer kleiner werdende Zahlen (logisch!) Man notiert nun, mit der *höchsten Zahl* beginnend (daher der Name) fortlaufend die *nächstniedrigeren* Zahlen und die jeweils dazu gehörende Partei (bzw. Person), und zwar solange, bis man soviele "Größtzahlen" beisammen hat, wie Sitze zu vergeben sind. Nun zählt man, wieviele solcher "Größtzahlen" jede Partei (Person) erhalten hat: soviele Sitze bekommt sie! Ganz einfach, nichtwahr, lieber Leser? Jedenfalls für die Wahlhelfer einfacher als umständliches Proportionalitätsrechnen...
Ein Leckerbissen für den Programmierfreak!

Hier nun mein Programm "d'HONDT" in BASIC. Du ahnst es, l.L.: Hier wird nach allen Regeln der Rechenkunst <u>sortiert:</u> Nämlich mit dem CMD"O"-Kommando! Und zwar absteigend. Das heißt, CMD"O" reiht die Quotienten (das sind die durch die natürlichen Zahlen dividierten Stimmen) nach abnehmender Größe. Deshalb das Minus vor der ersten Variablen (Zeile 120). Natürlich handelt es sich um eine *zweidimensionale* Variable, also eine Matrix. An anderer Stelle in diesem Heft wird die mathematische Seite des 2-dimensionalen Falles gesondert behandelt, da er etwas komplexer ist als der eindimensionale Fall und deshalb (leider) selten angewendet wird.
Durch einen Trick (die Z-Schleife in Zeile 70 und die I-Schleife in Zeilen 140-150) gelang es, die erhaltenen Höchstzahlen den einzelnen "Parteien" wieder zuzuordnen, so daß sie pro Partei ausgezählt werden können. Diese Auszählung wird schließlich in Zeile 170 ausgeworfen.
Vor Anwendung des Programmes vergesse man nicht, die Namen der Parteien (Personen) in die DATA-Zeile 180 einzutragen!

"Das wär's" (wie man sagt...)
Ach so: Was die Überschrift bedeute, fragst du, l.L.?
Na, dann *ORDNE* sie mal! <CMD"O",12,-X(I)>
Vergessen, was das "Minus" bedeutet?
Na, dann lies eben alles noch einmal!

-toJaK

Sitzverteilung nach d'HONDT

```
10 CLS
   PRINT "                d'Hondtsches Höchstzahlverfahren
     zur Ermittlung der Sitzverteilung im Parlament"
   PRINT
20 PRINT "             (C) KaJot Mühlenbein, Weinheim, 6/89
30 CLEAR 70
   PRINT STRING$ (61,"=")
40 DEFSTR P
   DEFINT A,I,M,N,S,Z                  'S=Spalte  Z=Zeile  N="Spalten-Tiefe"
50 PRINT
   INPUT "Anzahl der Parteien (muß mit DATA-Zeile 180 übereinstimme
n)    ";N
   N = N - 1
   INPUT "Anzahl der Sitze im Parlament ";ZZ
   M = (N + 1) * ZZ                         'M=max. Matrix-Größe
   ZZ = ZZ - 1
60 DIM ZP(ZZ),V(N,ZZ),P(N),PP(N,ZZ)
70 FOR S = 0 TO N
   READ P(S)
   FOR Z = 0 TO ZZ
   PP(S,Z) = P(S)
   NEXT Z
80 PRINT "Stimmenzahl für "P(S);
   INPUT V(S,0)                      'V=Votum
   NEXT S
90 PRINT
   PRINT "                     * * * Ich rechne ! * * *
100 FOR Z = 1 TO ZZ
    FOR S = 0 TO N
110 V(S,Z) = V(S,0) / (Z + 1)
    NEXT S,Z
120 CMD "O",M, - V(0,0),PP(0,0)
130 FOR Z = 0 TO ZZ
    FOR S = 0 TO N
140 FOR I = 0 TO N
    IF PP(S,Z) = P(I),SS(I) = SS(I) + 1
150 NEXT I
160 A = A + 1
    IF A = ZZ + 1
    THEN GOTO 170
    ELSE NEXT S,Z
170 FOR I = 0 TO N
    PRINT "Die Partei der "P(I)" erhält" TAB( 40);SS(I)" Sitze"
    NEXT
180 DATARäuber,Stinker,Trinker,Heuler,Hacker   <- Beispiel für 5 "Parteien"
```

KaJot

* * * Ordnung muß sein! * * *

Soeben bin ich erfrischt und gestärkt dem Entmüdungsbecken entstiegen

(der Gebrauch moderner Wortschöpfungen weist mich als modernen Menschen aus, der wenigstens in der Tagessprache up-to-date ist, wenn schon nicht in der Maschinensprache...)

und schon packt mich der Ordnungssinn: Die Seife hierhin, das Handtuch dorthin usw. Dies ist für mich also der erste Maßstab – oder, um langsam zur Sache zu kommen, die erste Dimension!
Doch schon erhalte ich gelinden Widerspruch: "Sie" meint, die Seife müsse dorthin und das Handtuch hierhin. Also ordnen wir beides um. Dies war – i h r e, die z w e i t e Dimension. Die übrigen Sachen durfte ich lassen, wo ich sie hintat.
Nun haben wir also eine zweidimensionale Ordnung neben dem "Entmüdungsbecken", weil ihr und mein Wille geschah.
Könnte man das nicht gleich mittels Computer steuern, bevor Disput entsteht?
Man kann.
Das heißt: "Er" kann.
Sowohl ein- als auch zwei- und mehrdimensional. Und – vor allem – natürlich auch in BASIC!
All dies bewältigt das Kommando:

```
     a)    CMD"O",z,A(0,0,0,...),B(0,0,0,...),...
bzw. b)    CMD"O",z,*I(0,0,0,...),A(0,0,0,...),B(0,0,0,...)...
```

Fangen wir "eindimensional" an.
Hierüber hatte ich mich bereits in einem INFO ausgelassen.
Daher hier in aller gebotenen "Kürze":
Gewöhnliche (feldfreie) Variablen können nicht mit diesem Kommando sortiert werden. Dafür könnte man ein Programm schreiben, das allerdings recht umständlich würde.
Elegant wird das Sortieren erst mit Feldern.
Im Feld A(J) ist J der "Index" oder die "Hausnummer" der Variablen A. Das Feld habe N+1 Elemente. Die Zählung beginnt mit J=0, der höchste Index ist also N, und diesen nennen wir die "Tiefe" der Felddimension. Beim Programmieren wird sie durch das statement "DIM A(N)" festgelegt. A ist, wie man sieht, "eindimensional".
Es können maximal 9 Feldvariablen gleichzeitig sortiert werden. Die Sortierung richtet sich nach dem ersten Feld; alle übrigen Felder machen dieser Neuordnung zwangsläufig mit, d.h. alle Felder mit gleicher "Haus-Nummer" bleiben stets beieinander!
Das 'z' im CMD"O"-Kommando bestimmt, wieviele Elemente des ersten Feldes sortiert werden; der Index, der für die Variable eingegeben wurde, bestimmt, ab welcher "Hausnummer" sortiert wird. Im allgemeinen ist dies die Null, also das erste Element. Es kann jedoch auch ab einem beliebigen anderen Element mit dem Index X sortiert werden. Sortiert wird in aufsteigender (ASCII-)Reihenfolge des ersten Feldes. Soll es absteigend gehen, also mit dem höchsten Wert der A(J) begonnen und abwärts sortiert werden, so wird ein Minuszeichen vor A(J) gesetzt.

Nun ein ganz wichtiger, interessanter Unterschied zwischen Form a) und b).
Benutzt man Form a), so werden die Felder sortiert, indem ihre Indizes (=Mehrzahl von Index; Betonung auf der ersten Silbe, das e wird lang gesprochen! <Also nicht mit den "Indizien" verwechseln...>) so verändert werden, daß sich bei der Auflistung mit einer J-Schleife die neue Reihenfolge ergibt (= "direkte Sortierung").
Benutzt man jedoch Form b), so bleiben die Indizes aller Variablen unverändert. Stattdessen werden die Werte (nicht der Index) des in dieser Form zusätzlich eingeführten, durch den Stern * gekennzeichneten "Indexfeldes" I(J) sortiert! Der Index J weist auf das zugehörige Element A(J) hin; der Wert von I(J) jedoch bestimmt dessen neue Position, letztlich also die Reihenfolge der sortierten Elemente A(J) (="indirekte Sortierung").
Will man diese auflisten, so muß die Schleife lauten:

```
FOR J = 0 TO Z-1
    <L>PRINT A(I(J))         } Z ≤ N+1
NEXT
```

Das Indexfeld I(J) muß übrigens ein Integerfeld sein.

Die "Hausnummern" (Indizes) der Elemente beginnen stets mit Null. Das erste Element ist also stets A(0). Man kann diese Hausnummern deshalb auch als "relative Element-Nummern" (R.E.N.) bezeichnen. Die "REN" (wir lassen die Punkte künftig weg) ist beim eindimensionalen Feld also identisch mit der Hausnummer (dem Index).
Nicht so bei *mehrdimensionalen* Feldern. (Ein Haus kann ja nicht mehrere Nummern haben!) Bei 2-dimensionalen Feldern z.B. wird die REN wie folgt berechnet.
Man denke sich die Elemente in einer Matrix angeordnet:

```
A(0,0) A(1,0) A(2,0) ... A(N,0)
A(0,1) A(1,1) A(2,1) ... A(N,1)
A(0,2) A(1,2) A(2,2) ... A(N,2)
  .      .      .    ...   .
  .      .      .    ...   .
A(0,M) A(1,M) A(2,M) ... A(N,M)
```

Der erste Index ist die Spalten-Nummer, der zweite die Zeilen-Nummer. Nach diesem Prinzip arbeitet CMD"O"!
N ist die Spalten-"Tiefe", M die Zeilen-"Tiefe". Die Anzahl aller Elemente ist mithin (N+1)*(M+1). – Alles klar?
Die REN ist hier nichts weiter wie die Durchzählung aller Matrixelemente von links oben nach rechts unten, und zwar zeilenweise, d.h. von links nach rechts! Also:

```
Den Indizes 0,0 entspricht REN=0,
 "      "   1,0     "      REN=1,
 "      "   2,0     "      REN=2,
            ...            REN=...
 "      "   N,0     "      REN= N (="Tiefe"),
 "      "   0,1     "      REN=N+1,
 "      "   1,1     "      REN=N+2
 "      "   2,1     "      REN=N+3   u.s.w.
```

HEFT 28
Oktober
1989

Allgemein gilt für eine 2-dimensionale Feldvariable mit der Dimensionierung DIM A(N,M), wobei die Spaltennummern S=0...N und die Zeilennummern Z=0...M lauten:

$$\underline{\text{REN von Element } A(S,Z) = Z * (N+1) + S} \qquad (1)$$

wie sich an der Matrix leicht ablesen läßt.
Die Zeilentiefe (M) geht bei dieser Zählweise nicht ein. Sie würde die entsprechende Rolle spielen, wenn man die Elemente spaltenweise (statt zeilenweise) - also senkrecht von oben nach unten - abzählen würde. Das tut CMD"O" jedoch nicht!
Wenn wir mehrdimensionale Felder sortieren wollen, lautet das Kommando genauso wie bei eindimensionalen, nämlich einfach
(Form a:) CMD"O", z, A(J,K<,...>),..<ggf.weitere Feldvariablen>
mit z=Anzahl der zu sortierenden Elemente und J,K,...=Indizes desjenigen Elementes, mit dem die Sortierung beginnen soll; das ist i.a. A(0,0<,0,...>). Sortiert wird diese erste Variable; die übrigen Felder werden lediglich "mitgenommen". Es dürfen wieder insgesamt bis 9 Felder sein.
Der Algorithmus ändert die Indizes so, daß die Elemente in der sortierten Reihenfolge erscheinen, wenn man sie mit dem folgenden Schleifensystem auflistet:
(Bsp.für eine 3-dimensionale Variable A(J,K,L) mit DIM(N,M,P):

```
FOR L = 0 TO P
    FOR K = 0 TO M
        FOR J = 0 TO N
          <L>PRINT A(J,K,L)
        NEXT J
    NEXT K
NEXT L
```

Soviel zur *direkten* Sortierung von mehrdimensionalen Feldern.
Wie sieht aber bei diesen die indirekte Sortierung aus?
Hierbei spielen die oben erklärten "Relativen Element-Nummern" die entscheidende Rolle. Bei der direkten Sortierung werden diese von der Routine automatisch als neue Indizes eingesetzt. Bei der indirekten Sortierung werden die Indizes der Variablen selbst jedoch nicht verändert, sondern lediglich die Werte des Feldes der relativen Elementnummern - des REN-Feldes -, das den Variablen im CMD"O"-Kommando vorangestellt und durch einen * als "Indexfeld" gekennzeichnet wird:
CMD"O", z, *I(0,0,...), ±A(0,0,...), B(0,0,...), ...
Das ± vor der ersten Variablen A (nach der sortiert wird) steht zur Wahl für auf- oder absteigende Sortierung. Für die übrigen Felder gilt das bereits oben Gesagte.
übrigens: Bei allen Sortierverfahren - ganz gleich, welche Dimension die Variablen haben - gelten sog. "Sortierlevel". Das bedeutet: Wenn mehrere Werte der ersten Variablen gleich sind, wird nach den *entsprechenden* Werten der nächsten Variablen (Level 2) solange weitersortiert, bis die erste Vartiable einen neuen Wert aufweist. Sind Werte der zweiten Variablen untereinander auch gleich, wird aufgrund der dritten Variablen sortiert (Sortierlevel 3) usw.

Das Auflisten indirekt sortierter mehrdimensionaler Variablen ist komplizierter als das der indirekt sortierten eindimensionalen. Das könnte der Grund dafür sein, daß indirekte Sortiervorgänge mehrdimensionaler Variablen so selten vorkommen, weil die Wiedergabe der Sortierung etwas umständlich ist. Es genügt ja nicht, wie bei den eindimensionalen Variablen, einen einzigen Index durch die neue REN zu ersetzen. Vielmehr müssen *alle* Indizes einer mehrdimensionalen Variablen irgendwie durch die neue REN, also durch das sortierte Indexfeld, ausgedrückt werden.
Aus obiger Gleichung (1) lassen sich für die REN folgende
Formeln für die neuen Indizes von 2-dim. Variablen
entwickeln:

Spaltenindex (=1.Index):

$$S = I(S,Z) - INT(I(S,Z)/(N+1)) * (N+1)$$

Zeilenindex (=2.Index):

$$Z = INT(I(S,Z)/(N+1))$$

worin N die "Tiefe" des Spaltenindexes, also, wie schon gesagt, N+1 die Anzahl der Spalten ist.
Diese Formeln sind für S und Z im Ka,Jot-Schleifensystem (s.o.) einzusetzen. Die Variablen werden dann in der sortierten Reihenfolge angezeigt bzw. gedruckt. Dabei hat jedes Element seinen alten Index S für die Spalten- und Z für die Zeilen-Nummer behalten! Das kann sehr nützlich sein, wenn die Variablen noch in anderem Zusammenhang in der ursprünglichen Reihenfolge benötigt werden.

Zum Schluß einen wichtigen Hinweis auf eine Bedingung, "die nicht im Handbuch steht" (APPARAT hat hier geschlafen), und hinter die ich erst jetzt nach langer Fehlersuche bei der Anwendung zweidimensionaler Sortierung gekommen bin (und das war mit ein Grund für diesen Beitrag):

> Sämtliche Variablen - einschl. eines Indexfeldes - sind auch dann durch "DIM A(.....) usw." zu dimensionieren, wenn die "Tiefe" der Dimensionen kleiner als 11 ist!

(Anmerkung: Im allgemeinen ist eine DIMensionierung bei N≤10 nicht erforderlich, weil NEWDOS von vornherein je Index 11 Plätze bereithält.)
Vor einem CMD"O"-Kommando muß jedoch auch dann DIMensioniert werden, wenn N≤10 ist, weil die Berechnung der REN sonst nicht funktioniert. Die Routine muß nämlich das Zeilenende erkennen können.
An anderer Stelle in diesem Heft findet sich ein Anwendungsbeispiel für zweidimensionale Sortierung (bei dem allerdings die direkte Sortierung genügte).

Die entsprechenden Rekursionsformeln für drei- und mehrdimensionale Sortierung sind wesentlich komplizierter. Ich habe deshalb beschlossen - nein, nicht "Politiker zu werden", sondern: sie nur auf Anfrage zu erläutern!
Denn es kennt seine Club'nheimer: der
(Sprich: Klapp'nheimer)

KaJot

[-NEXT L,K,J]

* P A S C A L + G A L T O N *

und der liebe Zufall

Nicht für Fakire ist das *Nagelbrett* von GALTON (geb.1822). Es sei denn, einer fühlt sich "als wie ein Fakir", der sich den Leiden der Mathematik unterzieht. Doch Galton war Biologe. Mit der dreieckigen Anordnung von Nägeln auf einem Brett, wie sie in der oberen Hälfte von Bild 1 zu sehen ist, wollte er zeigen, wie sich Erbeigenschaften auf die Nachkommen verteilen, wenn man die Erbbedingungen "dominant oder rezessiv" zugrundelegt, und zwar im Sinne von zwei Erbeigenschaften, die sich gegenseitig ausschließen; wobei das Verhältnis der Wahrscheinlichkeiten beider Eigenschaften durchaus nicht 1:1 sein muß (genauer gesagt, 0,5:0,5, denn die Summe muß stets 1 ergeben), sondern auch 0,7:0,3 oder 0,45:0,55 sein kann. Bild 1 bildet das Verhältnis 0,5:0,5 ab. Für andere Verhältnisse brauchen die Nagelreihen nur entsprechend gegeneinander verschoben zu werden. Doch wir wollen es hier nicht komplizieren.

Galton war ein so großer, genialer Biologe, daß er 1909 - zwei Jahre vor seinem Tode - vom englischen König in den Adelsstand erhoben wurde. Vielleicht war aber *Sir Francis Galton* - wie er seitdem anzureden ist - kein so großer Mathematiker. Sonst hätte er vielleicht auf seinen 200 Jahre älteren Vorläufer PASCAL (geb. 1623) hingewiesen, oder wenigstens auf den nur 45 Jahre älteren C.F.GAUSS, als er mit seinem dreieckigen Nagelbrett die natürliche Entwicklung der *Binomialkoeffizienten* mithilfe eines Zufallsexperimentes darstellte. * Oder hat er? Mein Mathelehrer hat das nicht verraten. (Ich finde, geistes- und naturgeschichtliche Zusammenhänge sollte man aufzeigen!)

Ihr habt das "Spielchen" vielleicht schon einmal irgendwo gesehen. Vor langer Zeit hing es noch als Spielautomat in Gaststuben (bevor Roulette- und Computerspiele Augen, Ohren und Nerven zerstörten...)
Man läßt eine kleine Kugel auf den obersten Nagel fallen. Dieser Ärmste muß sich nun "entscheiden", ob er nach links oder nach rechts fallen will (soll!) Beides ist im Falle des dargestellen "Galtonbrettes", bei welchem die Nägel genau mittig gegeneinander versetzt sind, gleich wahrscheinlich. Fällt die Kugel immer abwechselnd nach rechts und links, so landet sie schließlich genau in der Mitte des Bodens unter dem Brett. (Man stelle sich hier soviele Auffangbehälter vor, wie die letzte Nagelreihe Zwischenräume hat, aus denen die Kugeln fallen können.) Der "Zufall" will es jedoch, daß die Kugel manchmal zwei- oder dreimal nach der <u>gleichen</u> Seite fällt; dann wird sie irgendwo seitlich versetzt auf dem "Boden" landen.
Läßt man genügend Kugeln auf diese Weise durch die Nagelreihen wandern (wobei sie nur der Schwerkraft und dem "Zufall" unterworfen sind), so bilden sich am Boden Kugelhäufchen unterschiedlicher Höhe, die schließlich ein <u>Stapeldiagramm</u> ergeben, das sich um so besser der *GAUSS*ischen "Glockenkurve" nähern würde, je mehr Kugeln man fallen läßt - vorausgesetzt, man verbindet die Stapel durch eine stetige Kurve. Diese läßt sich durch eine etwas komplizierte Integralfunktion berechnen, auf deren Wiedergabe wir aber hier verzichten wollen, weil es nichts mit dem Programm zu tun hat.
Ich weiß wie gesagt nicht, ob *Galton* den *Gauß* kannte; als dieser starb, war *Galton* 33 Jahre alt; aber Engländer wollten ja oft nichts von den Deutschen wissen <was ich verstehen kann>. Da war ihm der Franzose *Pascal* schon lieber.

Dieser baute auch ein Dreieck auf; aber nicht aus Nägeln, sondern <u>Zahlen</u>. Nach dem Schema, das Bild 2 zeigt. Es ist für jeden leicht durchschaubar - oder? (Für den mathematisch Frustrierten: Man schreibe einen Kegelmantel(schnitt) aus lauter Einsen; dann schreibe man in die Zwischenräume immer die Summe der beiden darüberstehenden Zahlen - so einfach ist das!)
Die Zahlen, die man so erhält, sind ebenfalls die sog. "Binomialkoeffizienten", die sich ergeben, wenn man a+b beliebig potenziert. *Pascal* erfand dieses Dreieck als Trick für diejenigen, die nicht gerne mit Formeln umgehen. Denn man kann die Bin... - also die "B.K." auch aus "n über k"

$\binom{n}{k}$ (mit n=Zeilen-Nr., k=Spalten-Nr. im Dreieck, wobei Zeilen- und Spaltenzählung mit Null beginnen)

berechnen.
Doch *Galton* begnügte sich mit Nägeln (statt Zahlen) und mit Stapeln (statt stetigen Kurven) und erhielt so ebenfalls eine Darstellung der "B.K."! Zwar keine zahlenmäßige, aber eine anschauliche. Biologen denken eben praktisch.

Wenn ihr mein kleines Programm eingebt, könnt ihr die Kugeln laufen sehen. So etwa nach 100±20 Kugeln (sie werden links unten mitgezählt) bildet sich ein Stapeldiagramm aus, das sich nicht schlecht der Verteilung einer Fifty-fifty-Wahrscheinlichkeit nähert - also eine "Glockenkurve" andeutet. Natürlich cum grano salis - - - schließlich ist General *Zufall* im Spiel! Aber das "Gesetz der Großen Zahl" ist stärker als jeder Zufall - **oft** genug provoziert, muß er sich "dem Zwange edler Form" ergeben!

übrigens:
Dies Spiel ist einmal nichts für Maschinensprache.
Die Kugeln würden schneller als die eines Maschinengewehrs durch die Nägel rasen und niemand würde was <u>sehen</u>! Denn aufs Sehen kommt es hier an. Es ist fast interessanter, dem Lauf der (hier eckigen) Kugeln zuzusehen, als dem Anwachsen der Stapel!
Es sei denn, der M.L.-Bauer baut komische Pausen zwischen jedem Fall einer "Kugel" ein.
(Wer war das? Bitte nicht lächeln, sondern einsenden!!)

Darstellung einer Zufallsverteilung am GALTON'schen Nagelbrett

```
10 CLS:DEFINTI-K,X,Y
20 PRINT"          G A L -                    T O N";
30 FORY=2TO24STEP2
40 FORX=66-YTO62+YSTEP4
50 SET(X,Y):NEXTX,Y
60 FORX=35TO92:SET(X,47):NEXT
70 X=64:SET(X,0):FORK=1TO50:NEXT
80 FORY=0TO22STEP2:SET(X,Y):RESET(X,Y)
90 RANDOM:R=RND(0):S=SGN(.5-R)
100 SET(X+2*S,Y+1):RESET(X+2*S,Y+1)
110 SET(X+2*S,Y+2):RESET(X+2*S,Y+2)
120 X=X+2*S:NEXT
130 N=N+1:PRINT§972,N;
140 FORJ=25TO47
150 IFPOINT(X,J)THENSET(X,J-1):IFJ=25THEN170ELSEGOTO70
160 NEXT
170 GOTO170
```

HEFT 20
Oktober
1989

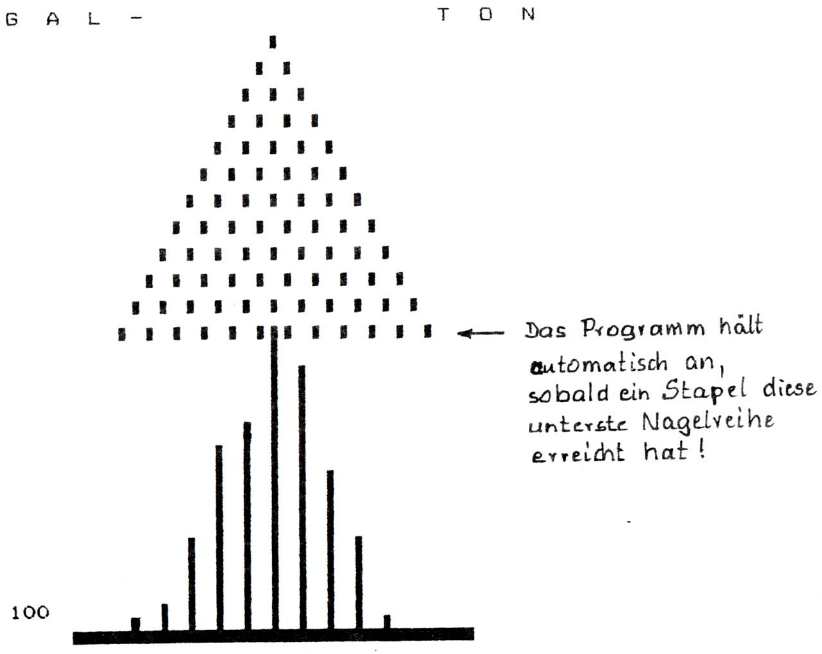
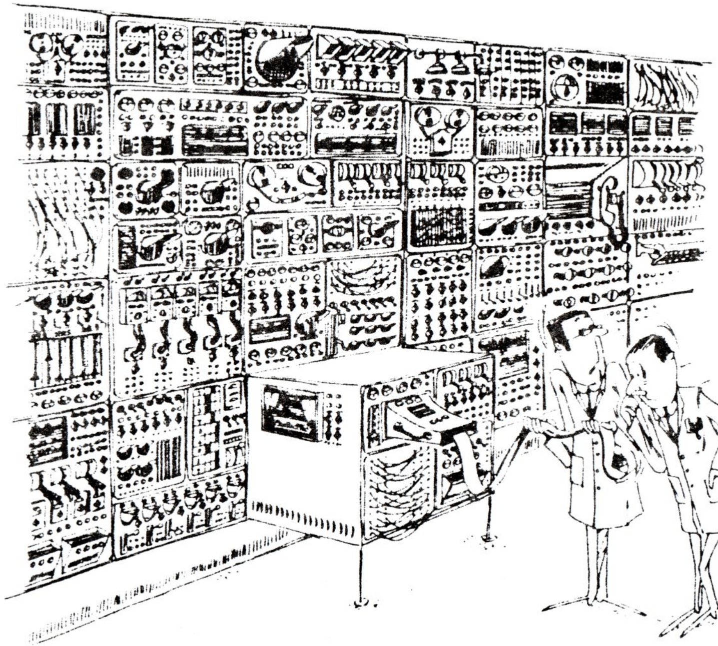


Bild 1

Das PASCAL'sche Dreieck:

```
                              1
                            1   1
                          1   2   1
                        1   3   3   1
                      1   4   6   4   1
                    1   5  10  10   5   1
                  1   6  15  20  15   6   1
                1   7  21  35  35  21   7   1
              1   8  28  56  70  56  28   8   1
            1   9  36  84 126 126  84  36   9   1
          1  10  45 120 210 252 210 120  45  10   1
        1  11  55 165 330 462 462 330 165  55  11   1
      1  12  66 220 495 792 924 792 495 220  66  12   1
 .................................................................
          u.s.w.                              u.s.w.
```

B i l d 2



»Ich werd' verrückt: Das Ding sagt: COGITO ERGO SUM!*«
* ich denke, also bin ich

Oskar Drechsler

Als Neuer möchte ich auch etwas zur Info beisteuern, weiß aber nicht, unter welchem Oberbegriff ich das anbieten soll. Es ist nämlich viel Zeugs, das größtenteils zusammengehört bzw. zusammenarbeitet. Vielleicht fangt Ihr einfach mal von vorne an:

Startprogramm für Basic-Programme

Wer sich Anfang der achtziger Jahre einen Computer gekauft hat, schreibt seine Programme selbst. Da sammelt sich im Laufe der Jahre einiges an. Wenn man Programme wegen höherer Geschwindigkeit über den Accel3 laufen läßt und auch noch eine HPG-Platine besitzt, muß man ständig darauf achten, daß das richtige Hilfsprogramm geladen ist. Außerdem kommt es vor, daß man ein Programm laden will und hat den genauen File-Namen vergessen. Irgendwann ging mir das gewaltig auf den Wecker und so entstand ein Programm, das das Löschen und Laden der Hilfsprogramme erledigt und alle ständig benötigten Programme in alphabetischer Reihenfolge mit einer Nummer auf dem Bildschirm zeigt. Man muß nur noch die Nummer tippen und schon rattert das Laufwerk. Wofür hat man einen Computer?

Das Programm heißt Start/Bas. Dort werden unter DATA die Programm-Namen eingetragen, das Laufwerk, wo sie zu finden sind und die Programmart-Kennung:
A = Accel3-Programm, benötigt Runtime-Routinen
H = HPG-Programm, benötigt HPG/CMD
B = Basic-Programm, wegen der Größe kein Hilfsprogramm zulässig
X = Basic-Programm, kann mit geladenem Accel oder HPG laufen
C = Maschinen-Programm
J = für Accel3-Compiler und PACKER

Alle Hilfsprogramme werden automatisch geladen. HPG wird automatisch gelöscht. Für Löschen von Accel3 wird man aufgefordert '/restore' und 'cont' einzugeben.

Wenn man im Menü des aufgerufenen Programms als Programmende RUN"Start/Bas" programmiert, muß man nur noch Knöppsche drücke.

Das Ablegen der Programm-Namen unter DATA erspart einen unnötigen Disketten-Zugriff und einen zusätzlichen File. Da die Programm-Namen selten verändert werden, gibt man sie in alphabetischer Reihenfolge ein und braucht nicht bei jedem Programmlauf den Diskettenzugriff für CMD"O".

Das Start-Programm alleine ist aber nicht viel wert, wenn der andere Kram unpraktisch ist. Deshalb jetzt ein anderes Thema.

Dateiprogramm

Eigentlich wollte ich das Programm später vorstellen, aber es arbeitet in der vorliegenden Form nur mit den im nächsten Abschnitt beschriebenen Programmen.

Obwohl ich universelle Dateiprogramme nicht leiden kann, möchte ich das nachfolgend beschriebene Programm nicht mehr missen. Es ist schnell zu erlernen, leicht zu bedienen und enthält keine Funktionen die sowieso niemand braucht. Es enthält allerdings keine Datenausgabe-Routine um möglichst viel Platz für Daten zu schaffen.

Das oben ausgesprochene Lob gilt nicht mir, sondern einem Bekannten, der inzwischen zu IBM-AT geflüchtet ist und gegen eine Aufnahme seines Programmes in unsere Programm-Bibliothek nichts einzuwenden hat. Das Programm lief recht ordentlich, hatte aber zwei kleine Fehler. Bei der Fehlersuche hatte ich noch einige Ideen und habe das Programm umgeschrieben und ergänzt. 30-40 % sind von mir.

Mit dem Maskengenerator (Generat/Bas) wird die Bildschirmmaske erstellt. Dabei kann innerhalb des vorgegebenen freien Raumes eine Maske vollkommen nach eigenen Wünschen gebastelt werden. Die Datensatzlänge kann von 1-255 gehen. Es können 1-25 Datenfelder vorkommen. Soll ein Feld linksbündig gefüllt werden, ist das Datenfeld mit '*' zu markieren, für rechtsbündig mit Doppelkreuzen. (Ich schreibe mit Genie-Text, dort sind Doppelkreuz und Dollarzeichen Steuerzeichen und somit nicht darstellbar). Wenn man anschließend die ENTER-Taste drückt, wird die Maske interpretiert und zusammen mit den notwendigen Parametern als xxxxx/MSK abgespeichert.

Datei/Bas nimmt die Maske und die Parameter als Futter und arbeitet damit. Die letzten zwei Zeilen des Bildschirmes geben Bedienerhinweise, sodaß zur Benutzung nicht viel zu sagen ist.

3 Funktionen sind näher zu erklären:
Bei Eingabe des Dollarzeichens an der 1. Stelle eines Datenfeldes wird der Inhalt des Datenfeldes des vorher eingegebenen Datensatzes in dieses Feld kopiert. Das ist recht nützlich, wenn in einem Datenfeld öfter gleiche oder ähnliche Eingaben vorkommen, die dann nur leicht geändert werden brauchen.

Bei der Suche eines Datensatzes kann in jedem Feld gesucht werden. Das Suchwort muß allerdings am linken Rand des Feldes beginnen und kann ein Teilwort sein, das aber am Wortanfang beginnen muß (keine Instring-Suche). Das Suchwort ist mit einem Doppelkreuz abzuschließen. Das Suchwort wird gespeichert. Wenn das Suchwort wieder verwendet werden soll, ist anstatt dessen nur ein Dollarzeichen einzugeben.

Die Daten werden bei der Eingabe nach dem 1. Datenfeld, innerhalb diesem nach dem 2. Datenfeld ... usw. sortiert. Abgespeichert wird sequentiell mit dem FF-File. Alle Zahlen liegen wegen der Speicherung im RAM (nicht als Variable) als String vor und werden auch so abgespeichert.

Bei der Auswahl der Funktionen gab es wie üblich Probleme mit den Anfangsbuchstaben von Speichern und Suchen. Deshalb erfolgt die Auswahl in englisch. Siehe Hardcopy's am Ende.

Das Programm befindet sich einmal als Datei/Ori in Originalform mit Kommentaren auf Diskette, in der es nicht lauffähig ist und einmal als Datei/Bas, gekürzt mit dem PACKER.

Für das Auslesen der Daten gebe ich mein Programm Elohefte/Bas als Muster dazu, ebenfalls die Daten. Das erleichtert den Einstieg.

1. Bild

```
──────────── Eingabemasken - Generator ────────────

Eingabefelder: * = Kennung für Feld linksbündig
               # = Kennung für Feld rechtsbündig
Vor dem ersten und hinter dem letzten '*'#' muß mindestens
ein ' ' stehen.

               Die maximale Feldzahl beträgt 25.

Satzlänge        : 1 ... 255 Zeichen

Steuerzeichen    : <--/--> / Auf-/Abpfeil = Cursorbewegung
                   Sh <--/-->             = Chr löschen/einfügen
                   <New Line>             = Interpretation
   Zum Fortfahren bitte 'Taste' drücken .......
```

Beispiele für Erstellung einer Maske

```
   Add         Update        Delete        Inquire        Save
|Dateiname:               |  |Satznummer:       |  |Satzzahl:     |
─────────────────────────────────────────────────────────────────
Titel      : ***********************************
Fundstelle: ****************
Kode       : *

El      = Elektor
PE      = Populäre Elektronik
ELV
elrad
Buchstabe = Fundstelle im Ordner
─────────────────────────────────────────────────────────────────
```

```
   Add         Update        Delete        Inquire        Save
|Dateiname:               |  |Satznummer:       |  ·Satzzahl:     |
─────────────────────────────────────────────────────────────────
Titel      : ***********************************   ← linksbündig
Fundstelle: ################  ← rechtsbündig
Kode       : *

El      = Elektor
PE      = Populäre Elektronik
ELV
elrad
Buchstabe = Fundstelle im Ordner
─────────────────────────────────────────────────────────────────
```


Datei/Bas

Befehlsmodus erscheint nach Programmstart und Laden der Daten.

```
    Add         Update        Delete        Inquire        Save
|Dateiname: elohefte/SEQ|  |Satznummer:    |  |Satzzahl: 422|
─────────────────────────────────────────────────────────────────
Titel     :|                                       |
Fundstelle:|                  |
Kode      :| |

El      = Elektor
PE      = Populäre Elektronik
ELV
elrad
Buchstabe = Fundstelle im Ordner
─────────────────────────────────────────────────────────────────
 <A>dd    <U>pdate    <D>elete    <I>nquire    <S>ave    <Q>uit
```

wurde gewählt ↙

```
  | Add |      Update        Delete        Inquire        Save
|Dateiname: elohefte/SEQ|  |Satznummer:    |  |Satzzahl: 422|
─────────────────────────────────────────────────────────────────
Titel     :|                                       |
Fundstelle:|                  |
Kode      :| |

El      = Elektor
PE      = Populäre Elektronik
ELV
elrad
Buchstabe = Fundstelle im Ordner
─────────────────────────────────────────────────────────────────
 Shift <--/-->  Chr löschen/einfügen       <CLEAR> Feld löschen
 Bitte Datensatz eingeben... oder   <$> für Befehlsmodus
```

wurde gewählt ↙

```
    Add      | Update |      Delete        Inquire        Save
|Dateiname: elohefte/SEQ|  |Satznummer:    |  |Satzzahl: 422|
─────────────────────────────────────────────────────────────────
Titel     :|                                       |
Fundstelle:|                  |
Kode      :| |

El      = Elektor
PE      = Populäre Elektronik
ELV
elrad
Buchstabe = Fundstelle im Ordner
─────────────────────────────────────────────────────────────────
 Suchbegriff mit <#> abschließen   oder   <$> für Befehlsmodus
```

Zum Schluß zur Speicherbelegung

Ich hatte Routinen aus Basic f&b oben im Speicher liegen und wenn ich nach Benutzung von HRG/CMD vergaß, sie neu zu laden, stürzte der Rechner beim Sprung auf eine der nicht mehr vorhandenen Routinen ab. So faßte ich den Keyboard-Treiber und diese Routinen im File MaschUnt/USR zusammen und legte sie unter Basic-Anfang, der auf 7051H angehoben wird. Somit steht er unbehelligt da unten rum und stört nur, wenn man Basic mit mehr als 3 Filebereichen lädt. Aber wann kommt das schon vor!

Die Diskette, die ich Oliver schicken werde, enthält folgendes:

NEWDOS 2.052 und Basic mit deutschen Fehlerkommentaren, formatiert mit 40 Tr SS/DD.

Mit Autostart wird das Programm 'Anfang/Bas' geladen, das den Basic-Anfang anhebt und das Programm 'Start/Bas' rennen läßt. Dieses Programm lädt 'MaschUnt/USR' nach, startet den Keyboard-Treiber und zeigt alle Programme, die abrufbar sind.

Der Chainfile 'Allstart/JCL' macht beim Aufruf von HRG/CMD und den Runtime-Routinen von Accel3 'AccelRun/CMD' den Automatismus und ruft den 'Accel3/CMD'-Compiler und den PACKER auf.

Die in HRG/CMD enthaltene LPRINT-Routine ist für Epson-kompatible Drucker und beinhaltet ESC K n1 n2. Läuft mindestens auf Epson FX 85, CPA 80 und Nec P 5. Ich habe die Druckerinitialisierung vor und nach LPRINT herausgenommen, um mehrere Hardcopys nahtlos aneinanderfügen zu können.

Wer schon Accel3-Programme hat, muß sie durch Aufruf des Accel3 über das Start-Programm neu compilieren, weil der Basic-Anfang beim Compilieren genauso stehen muß wie beim Programmlauf.

Außerdem stehen folgende Routinen aus Basic f&b zur Verfügung: ID-Array, Quick-Array, LStrip, Search2, Sort3, Movex, RStrip. RStrip befindet sich allerdings auf dem File Basic/CMD zusammen mit CLS für die HRG 1B. Da habe ich noch etwas freien Platz gefunden. Einsprungadressen siehe nachfolgende Speicherbelegung.

Wer noch Fragen hat, kann mich täglich von 1900-2400 Uhr anrufen. Wenn ich ins Bett gehe, stelle ich das Telefon ab. Ihr könnt mich also nie stören. ☎ 02202/55282

Viele Grüße

Speicherbelegung

DB00-FF17	56064-65303	HRG/CMD	
E9E0-FFBF	59872-65471	Accel3/CMD	
E9E0-EFE2	59872-61410	AccelRun/CMD	
EF79-FF79		Packer/CMD	
7001	28673	frei für Basic-Programme	
7000	28672	enthält '0', muß so bleiben	
6F26-6FFF	28454-28671	frei	
6F1B-6F25	28443-28453	KbdNeu ausschalten	
6D88-6F1A	28040-28442	KBDNEU	
6D87	28039	Zeichensatz HRG geladen	
6D86	28038	Programm-Nr. in Start/Bas	
6D85	28037	Programm auf Laufwerk	
6D84	28036	geladenes Hilfsprogramm	
6D83	28035	MaschUnt/USR geladen	
6D82	28034	Kontrollzahl	
6CD8-6D81	27864-28033	Search2	USR8
6C62-6CD7	27746-27863	ID-Array	USR5
6BDD-6C61	27613-27745	Search1	USR4
6B57-6BDC	27479-27612	Quick-Array	USR3
6B38-6B56	27448-27478	LStrip	USR1
6AE0-6B37	27360-27447	Movex	nur: USR0
6A47-6ADF	27207-27359	Sort3	USR2

Die Maschinenprogramme befinden sich auf dem File MaschUnt/USR bzw. MaschUnt/SRC

Folgende Routinen befinden sich auf BASIC/CMD:
siehe auch BasicMod/SRC

669E-66BB	RStrip
6698-669D	Chr$(27) auf Drucker
668B-6697	Zeichen in Akku auf Drucker
6682-668A	Chr$(27);Chr$(33) auf Drucker
6669-6681	#CLS

Basic-Anfang wird durch Anfang/Bas höher gesetzt
Es können nur 3 Files gleichzeitig geöffnet sein

Bei Änderung Basic-Anfang, Accel-Progr. neu kompilieren!

Beim künftigen Start aus dem DOS ist immer wie folgt zu verfahren:
Basic,run'ANFANG/BAS'
Die letzte Zeile von Anfang/Bas muß ein RUN auf ein anderes Basic-Programm enthalten, das als erstes enthalten muß:

```
CMD'load Maschunt/USR':DEFUSR &H6D88:X=USR(X)
```

Geheimnisse eines Zeichengenerators

von Norbert Giese

September 1989

In diesem Artikel möchte ich Euch meine Erfahrungen und Erkenntnisse mitteilen, die ich bei der Änderung zweier Charaktergenerator-EPROMs gemacht bzw. gewonnen habe. Dieses auf den ersten Blick einfache Unterfangen entpuppte sich zu einem umfangreichen Projekt, gab aber auch Einblicke in eine Methode, einen vollständigen Charaktersatz in möglichst wenig Speicherplatz unterzubringen.

Konkret beziehen sich die folgenden Angaben auf einen Sharp MZ-80B Computer mit Z80 CPU und dem dazugehörenden Drucker MZ-80P5 mit 8085 CPU (ab Produktionsdatum Oktober 1981). Beide Geräte haben in ihren Zeichensätzen u. a. einen Unterstrich (ASCII 95, bzw. hexadezimal '5F'), der aber aus mir unerklärlichen Gründen als "Oberstrich" auf dem Bildschirm und auf dem Papier erschien. Die üblichen Pascal-Listings mit ihren Variablennamen und den verbindenden Unterstrichen sahen folglich etwas seltsam aus, auch sonst war mit dem "Oberstrich" nicht viel anzufangen. Dieser Tatbestand sollte geändert werden.

Aus beiden Geräten wurden die entsprechenden EPROMs ausgebaut, auf dem EPROMMER eines Freundes ausgelesen und als Hex-Dump aufs Papier gebracht. Das EPROM 2716 (2 Kilobyte) aus dem MZ-80B enthält den Zeichensatz in Form einer Matrix aus je acht Zeilen und acht Spalten pro Zeichen. Der folgende Auszug aus einem Hex-Dump, schon etwas aufbereitet, zeigt dies deutlich.

```
0000    00 00 00 00 00 00 00 00    ;ASCII 00H
0008    1C 14 14 77 22 14 08 00    ;ASCII 01H
 .
0100    00 00 00 00 00 00 00 00    ;ASCII 20H
0108    08 08 08 08 00 00 08 00    ;ASCII 21H
0110    24 24 24 00 00 00 00 00    ;ASCII 22H
 .
0200    1C 22 4A 56 4C 20 1E 00    ;ASCII 40H
0208    18 24 42 7E 42 42 42 00    ;ASCII 41H
0210    7C 22 22 3C 22 22 7C 00    ;ASCII 42H
 .
02F8    FF 00 00 00 00 00 00 00    ;ASCII 5FH
 .
06F8    00 FF FF FF FF FF FF FF    ;ASCII DFH
 .
07F8    00 00 01 3E 54 14 14 00    ;ASCII FFH
```

Jeweils acht Bytes formen ein Zeichen. Wenn man für jedes dieser acht Bytes deren Bitfolge zu Papier bringt, repräsentiert jedes gesetzte Bit einen Punkt auf dem Bildschirm. Die folgenden Zeichen "A", "B", und der "Oberstrich" zeigen dies.

```
0208 18  ...XX...      0210 7C  .XXXXX..      02F8 FF  XXXXXXXX
0209 24  ..X..X..      0211 22  ..X...X.      02F9 00  ........
020A 42  .X....X.      0212 22  ..X...X.      02FA 00  ........
020B 7E  .XXXXXX.      0213 3C  ..XXXX..      02FB 00  ........
020C 42  .X....X.      0214 22  ..X...X.      02FC 00  ........
020D 42  .X....X.      0215 22  ..X...X.      02FD 00  ........
020E 42  .X....X.      0216 7C  .XXXXX..      02FE 00  ........
020F 00  ........      0217 00  ........      02FF 00  ........
```

Der "Oberstrich" beginnt im Hex-Dump auf der Adresse 02F8H als Folge von acht Bytes, nämlich "0FFH, 000H, 000H, 000H, 000H, 000H, 000H, 000H". Ändert man dies in die Bytefolge "000H, 000H, 000H, 000H, 000H, 000H, 000H, 0FFH", dann wird aus dem "Oberstrich" der gewünschte Unterstrich. Da der Zeichensatz des MZ-80B alle druckbaren Zeichen auch noch in invertierter Form, sprich schwarze Punkte auf weissem Hintergrund, enthielt, wurde auch hier der invertierte "Oberstrich" geändert (Adresse 06F8).

Bevor ich auf das EPROM im Drucker komme, noch eine kurze Rechnung zum Platzbedarf. Der Zeichensatz im MZ-80B spezifiziert die ASCII-Kodes 00 bis 255, also 256 Zeichen. Acht Bytes pro Zeichen mal 256 ergibt 2048 Bytes, also genau 2 Kilobyte. Im Hex-Dump entspricht dies den Adressen 0000H bis 0FFFH. Das EPROM ist also vollständig belegt.

Nun galt es noch das EPROM aus dem Drucker zu ändern. Beide Zeichensätze sollten ja identisch sein. Da es ein 2732 EPROM (4 Kilobyte) ist, vermutete ich je 2 Kilobyte für den Zeichensatz und für das Betriebsprogramm des Druckers. Jedoch lies sich nur gelegentlich ein vollständigen Zeichens entdecken, oft nur die Andeutung davon. Das Service Manual, etwas genauer gelesen, sprach von einem 'verdichteten Charaktersatz', sonst schwieg es beharrlich.

Wie aber sollte ich diesen entschlüsseln? Einen Disassembler für den 8085 Prozessor besitze ich nicht. Dessen Mnemonics wollte ich auch nicht unbedingt lernen.

Irgendwann erinnerte ich mich, dass der 8085 Prozessor 100% software-kompatibel zum 8080 ist, dieser wiederum eine Untergruppe des Z80 ist. Folglich sollte doch mit einem Z80-Disassembler etwas zu machen sein. Gedacht, getan. Der Inhalt des EPROMs 2732 wurde disassembliert und das Listing Stück für Stück kommentiert. Meine Achtung vor den Designern stieg mit jedem Byte. Nie zuvor hatte ich daran gedacht, so weit in das Innenleben eines Druckers vorzudringen. Es dauerte einige lange Nächte, bis ich eine Stelle im Programmlisting gefunden hatte, die offensichtlich ein vom Drucker empfangenes Byte in eine Kette von Bytes für den Druckkopf aufbereitete. Da der Druckkopf nur acht Nadeln hat, lag es nahe, dass jeweils ein Byte die abzufeuernden Nadeln spezifiziert, bevor der Schrittmotor des Druckkopftransportes einen winzigen Schritt in Schreibrichtung macht. Auf diese Weise werden Schritt für Schritt aus einzelnen Bytes Zeichen und letztendlich eine ganze Textzeile. Die Zeichenhöhe ist somit acht Punkte, was sich auch unter einer Lupe auf dem Papier zeigte.

Wie aber wird aus einem einzelnen, vom Drucker empfangenen Byte eine Folge von Bytes für den Druckkopf? Am einfachsten wäre eine Tabelle, in der für jeden empfangenen ASCII-Wert z.B. acht Bytes für den Druckkopf stehen. Diese Tabelle könnte ähnlich wie beim MZ-80B aufgebaut sein und wäre genauso gross. Dies war aber offensichtlich nicht der Fall. Erstens hatte das Betriebsprogramm im EPROM 2732 schon mehr als die Hälfte der 4 Kilobyte belegt, zweitens hätte ich dann ja die einzelnen Zeichen finden müssen, und drittens hiess es ja 'verdichteter' Charaktersatz.

Die Ingenieure von Sharp haben sich damals eine, wie ich meine, elegante Methode einfallen lassen, um dieses Platzproblem zu lösen, vermutlich auch aus Kostengründen. Ein ausgeklügelter Algorithmus und zwei Tabellen sind der Kern der Lösung. Der Algorithmus empfängt einen zu druckenden ASCII-Wert und bereitet zuerst einmal einen internen Puffer von 14 Bytes vor, den er mit 00 füllt. Dies bedeutet, dass später 14 Bytes für jeden empfangenen ASCII-Wert an den Druckkopf weitergegeben werden. Die Darstellung in diesem Puffer bewirkt, das später für jedes gesetzte Bit eines Bytes eine Nadel des Druckkopfes abgefeuert wird. Zwischen den einzelnen Bytes wird natürlich ein Vorschub des Druckkopfes in Schreibrichtung ausgeführt.

Nun gilt es, diese 14 Bytes im Puffer entsprechend des empfangenen ASCII-Werts zu setzen. Dazu bedient sich der Algorithmus der zwei Tabellen, die hier verkürzt und der Übersicht halber nebeneinander gezeigt sind.

```
DEFB 00000001B          DEFB 000H                   ;ASCII 20H
DEFB 00110010B          DEFB 000H, 05FH             ;ASCII 21H
DEFB 11100010B          DEFB 007H,0000H             ;ASCII 22H
DEFB 11000011B          DEFB 014H,07FH,014H         ;ASCII 23H
DEFB 00001001B          DEFB 004H,02AH,000H,02AH    ;ASCII 24H
                        DEFB 055H,02AH,000H,02AH
                        DEFB 010H
DEFB 00001001B          DEFB 003H,040H,023H,010H    ;ASCII 25H
                        DEFB 008H,004H,062H,001H
                        DEFB 060H
.
DEFB 00001001B          DEFB 032H,000H,049H,000H    ;ASCII 40H
                        DEFB 079H,000H,041H,022H
                        DEFB 01CH
DEFB 01000101B          DEFB 07BH,004H,012H,001H    ;ASCII 41H
                        DEFB 010H
.
DEFB 11000011B          DEFB 001H,001H,001H         ;ASCII 5FH
DEFB 10100011B          DEFB 001H,002H,004H         ;ASCII 60H
.
```

Die Tabelle auf der linken Seite enthält für jedes druckbare Zeichen ein Byte, die Tabelle auf der rechten Seite für das korrespondierende Zeichen eine variable Anzahl von Bytes. Jedes Byte der linken Tabelle hat folgende Bedeutung:

Bit	7	Flag für Leerspalten einfügen
Bit	6	Flag für Spiegelung
Bit	5,4	Offset in Puffer
Bit	3,2,1,0	Anzahl Bytes aus rechter Tabelle

Die Bits 3,2,1 und 0 bestimmen die Anzahl der Bytes, welche für den empfangenen ASCII-Wert aus der rechten Tabelle in den Puffer zu stellen sind. Die Bits 5 und 4 geben an, ab welcher Stelle im Puffer diese Bytes zu speichern sind. Ist Bit 6 gesetzt, werden die Bytes aus der rechten Tabelle noch einmal gespiegelt in den Puffer geschoben (Erklärung folgt). Ist Bit 7 gesetzt, wird zwischen jede zu druckende Spalte eine Leerspalte eingefügt.

Dieses ist zugegeben nicht einfach und bedarf eines Beispiels. Der Drucker empfängt z.B. den ASCII-Wert 73 ='I', bzw. 49H. Irgendwann entscheidet sich das Betriebsprogramm des Druckers, dieses Zeichen zu drucken und übergibt es an den besagten Algorithmus. Der interne 14-Byte Puffer wird vorbereitet:

```
Position    1  2  3  4  5  6  7  8  9  10 11 12 13 14
Inhalt      00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Die für Hex 49 gültigen Einträge in den beiden Tabellen sehen wie folgt aus:

```
DEFB    11100010B         DEFB    041H,07FH      ;ASCII 49

   Bit  76543210
```

Die Bits 5 und 4 in der linken Tabelle, hier 0010B, bestimmen den Offset in den Puffer. Zwei Positionen (1 und 2) bleiben also unverändert.


```
Position    1  2
Inhalt      00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Die Bits 3, 2, 1, 0 in der linken Tabelle, hier 10B, bestimmen, wieviele Bytes aus der rechten Tabelle in den Puffer geschoben werden sollen. Also zwei Bytes. Zusätzlich ist aber Bit 7 aktiv, es soll also eine Leerspalte nach jedem Byte eingefügt werden. Der Puffer sieht nun so aus:

```
Position    1  2  3  4  5  6
Inhalt      00 00 41 00 7F 00 00 00 00 00 00 00 00 00
```

Nun ist noch das Bit 6 in der linken Tabelle gesetzt. Es weist den Algorithmus an, die Bytes aus der rechten Tabelle noch einmal gespiegelt in den Puffer zu schieben, dabei aber das letzte Byte zu ignorieren. Also wird 07FH ignoriert und 041H in den Puffer geschoben, wieder gefolgt von einer Leerspalte (Bit 7 aktiv). Der Puffer sieht nun so aus:

```
Position    1  2  3  4  5  6  7  8
Inhalt      00 00 41 00 7F 00 41 00 00 00 00 00 00 00
```

Schickt man diese Folge von Bytes gedanklich zum Druckkopf und schiebt diesen nach jedem Byte einen Schritt nach rechts, ergibt sich folgendes Bild für das Zeichen 'I':

```
Nadel 1  . . X . X . X .
Nadel 2  . . . . X . . .
Nadel 3  . . . . X . . .
Nadel 4  . . . . X . . .
Nadel 5  . . . . X . . .
Nadel 6  . . . . X . . .
Nadel 7  . . X . X . X .
Nadel 8  . . . . . . . .

         00 00 41 00 7F 00 41 00 00 00 00 .....
```

Es ist also gelungen, mit einem Byte aus der linken Tabelle und zwei Bytes aus der rechten Tabelle ein 'I' vollständig darzustellen. Spielt man dieses Beispiel mit anderen Zeichen durch, stellt man sehr leicht fest, dass viele Zeichen unseres Alphabets sich vertikal spiegeln lassen (A, H, I, M, O, T, U, ...). Man braucht folglich nur einen Teil dieser Zeichen in der Tabelle angeben, ohne dass Information verloren-geht. Auch ist der Abstand zum benachbarten Zeichen durch den Offset und die Länge des Puffers eindeutig bestimmt.

Rechnet man dieses Verfahren der Platzoptimierung durch, ergibt sich folgende Bilanz: Der Algorithmus und die Tabellen zur Spezifizierung von 256 Zeichen belegen zusammen etwa 1700 Bytes, dazu im Vergleich das Bildschirm-EPROM im MZ-80B, das genau 2 Kilobyte belegt. Hier muss man aber berücksichtigen, das dieser die einzelnen Zeichen nur in einer Breite von acht Punkten auflöst, dagegen der Drucker eine horizontale Auflösung von 14 Bytes hat. Also eine erhebliche Platzeinsparung.

Einige Leser mögen sich nun fragen, warum man über Bit 7 noch die Einfügung von Leerspalten steuert. Hier hätte man doch offensichtlich die horizontale Auflösung besser machen können. Hier muss ich nun etwas spekulieren, da ich die Sharp-Ingenieure nicht gefragt habe. Würde man die offensichtlich vorhandene horizontale Auflösung dadurch ausnutzen, dass man die Zeichen unter Weglassung der Leerspalten in der linken Tabelle genauer spezifiziert, müsste man dies konsequenterweise für alle Zeichen mit gesetztem Bit 7 machen. Dadurch würde die Tabelle grösser (Platzproblem). So aber kann man die Auflösung selektiv für die Zeichen nutzen, welche schräge Linien haben, wie zum Beispiel das Zeichen 'A'. Hier ist das Bit 7 zurückgesetzt. Ein weiterer Grund wird durch die Mechanik des Druckkopfes und die Druckgeschwindigkeit gegeben sein. Würde man zum Beispiel das Zeichen 'I' unter Weglassung der Leerspalten spezifizieren, dann würden die Nadeln 1 und 7 über die gleiche Wegstrecke fünf anstatt drei mal feuern müssen. Dies verlangt, soll es mit gleicher Geschwindigkeit passieren, eine schnellere Mechanik im Druckkopf und definitiv ein stärkeres

Netzteil im Drucker. Wenn man z.B. das Zeichen 'A' genau betrachtet, hat auch hier jede Nadel eine 'Pause', bevor sie wieder gefeuert wird.

Nach diesen Erkenntnissen war die Lösung des "Oberstrich"-Problems eine Kleinigkeit. Das Byte in der linken Tabelle blieb unverändert, in der rechten Tabelle wurde aus der Bytefolge "001H, 001H, 001H" die Folge "080H, 080H, 080H", und für den invertierten "Oberstrich" wurde aus "0FEH, 0FEH, 0FEH, 0FEH, 0FEH, 0FEH" die Folge "07FH, 07FH, 07FH, 07FH, 07FH, 07FH". Es tat mit fast leid, den Sharp-Ingenieuren ihr Geheimnis entrissen zu haben, als der erste Unterstrich auf dem Bildschirm und dem Drucker erschienen war.

Den Lesern, denen in diesem Text merkwürdige Umlaute und falsche "ss" aufgefallen sind: deutsche Umlaute habe ich noch nicht eingebaut, aber jetzt weiss ich ja, wie so etwas gemacht wird.

Teuer, aber überflüssig

Eigentlich können Sie es auch gleich sein lassen, vor allem dann, wenn Sie versuchen, ein modernes System zu erwerben. Denn beim Computer gilt eine eiserne Faustregel: Spätestens alle drei Monate wird eine neue Maschine auf den Markt geworfen, die nicht nur alles bisher Dagewesene technisch in den Schatten stellt, sondern auch nur noch einen Bruchteil des Vormodells kostet. Außerdem werden die Dinger auch immer kleiner. Computer, die noch vor zwanzig Jahren mehrere Stockwerke eines Rechenzentrums füllten, haben heute bequem in einer Würstchenbude Platz.
Irgend jemand hat mal folgenden Vergleich angestellt: Wenn man diese Entwicklung auf die Automobilindustrie überträgt, dann wäre ein Rolls-Royce, der vor 20 Jahren 80000 Mark kostete, heute schon für fünf Mark achtzig zu haben. Dafür hätte er allerdings auch in einer Streichholzschachtel Platz.
Da stellt sich die Frage: Wer will schon seinen Rolls in der Hosentasche herumtragen? Und ganz nebenbei: Wie soll man da noch den Knopf fürs Autoradio erreichen?
Die Computer könnte man schon längst noch viel kleiner bauen, wenn nur der Mensch nicht so sperrig wäre. Aber unsere Fingerchen sind eben dafür entwickelt worden, Keulen und Kanonen zu bedienen und nicht mit Mikrochips zu hantieren. Merken Sie sich also die Regel Nummer eins beim Kauf der neuen Wundermaschinen:

- **Fragen Sie sich, wofür Sie die Kiste eigentlich brauchen können**

Wenn Sie nicht zufällig der Innenminister sind oder Chef einer großen Versicherungsanstalt, dann überlegen Sie sich die Sache noch mal. Wie haben Sie denn die letzten zehn Jahre Ihre Probleme gelöst? Mal ehrlich: Um Ihren Kontostand zu ermitteln, genügt doch schon eine Kinderrechenmaschine mit Holzperlen, und den Biervorrat im Eisschrank überschauen Sie am besten, indem Sie einfach mal die Türe öffnen und nachzählen: 1, 2, 3!
Leider leben wir in einer Zeit, in der man der Werbung ausgeliefert ist. Jedes Produkt wird mit einem passenden Image versehen: Die harten Kerle rauchen die ungesunden Zigaretten, die Softies ziehen die teuren italienischen Klamotten an. Wer sich für sportlich hält, kauft nur einen weißen Wagen mit schwarzen Gummiflossen und Streifen am Heck. Eine Hausfrau, die nicht mit »Rakete« aufwischt und mit »Blitzy« spült, kann nicht einmal mehr wagen, ihre Freundinnen zum Kaffee einzuladen.

Im technischen Bereich scheint die Sache ebenfalls klar zu sein: Wer heute dazugehören möchte, wer dynamisch, modern, aufgeschlossen, tolerant, aber trotzdem noch ein bißchen individualistisch ist, braucht einen Computer – sagt die Werbung.
So ein Kasten mit dem kleinen grünen Flimmerschirm ist angeblich für alle da. Denn so ein Ding ist einfach »wahnsinnig praktisch«. Das kann alles und kostet dabei nur halb soviel wie ein Auto. Ein Computer gehört einfach zum Image des modernen Menschen.
Das Genialste an der Erfindung des Computers ist, daß die Medien uns erfolgreich eingeredet haben: Jeder, der sich der Kiste verweigert, ist ein weltfremder Typ von vorgestern!
Nie hat uns jemand aufgefordert, einen Fahrkartenautomaten, Zigarettenautomaten oder eine Telefonzelle in unserer Wohnung aufzustellen. Auch eine Druckmaschine oder eine zusammenklappbare Autowerkstatt wären Dinge, mit denen fast jeder etwas anfangen könnte. Aber das ist von den Marketing-Leuten verpennt worden.
Seit der Vermarktung des Automobils hat niemand in der Werbebranche mehr daran geglaubt, daß es jemals wieder etwas geben könnte, das nicht nur unnütz und gefährlich, sondern auch noch immens teuer ist und das man trotzdem jedem halbwegs vernünftigen Mann zwischen sechs und sechzig mühelos andrehen kann. Jetzt ist es endlich gefunden: das teure Ding für jedermann!
Nachdem Sie nun schon einmal festgestellt haben, daß das Leben bisher auch ohne Computer ganz gut funktionierte, fragen Sie sich:

- **Wann habe ich Zeit für den Kasten?**

Schließlich arbeiten Sie ja von neun bis um halb fünf. Wenn Sie nach Hause kommen, will erst mal Ihre Frau von ihnen wissen, wie der Tag gewesen ist, und dann möchte Ihre kleine Tochter aus dem Buch über die Schweine vorgelesen bekommen.
Außerdem ist es höchste Zeit, daß Sie endlich mal wieder ihren Hund kämmen. Und kaum ist das Abendbrot fertig, lockt Frank Elstner mit der neuesten Ausgabe von »Wetten, daß ...?« zu einem gemütlichen Nickerchen vor der Glotze.
Also, wo ist denn da Zeit, um am Computer zu arbeiten?
Von Arbeiten kann sowieso keine Rede sein. Die Untersuchungen zeigen: Mit der programmierbaren Schreibmaschine wird mehr gespielt als gearbeitet. Das ist schon wieder so ein genialer Streich der PR-Strategen. Aufs Knöpfchen gedrückt wurde schon immer gerne: Ob es die Flipper-Automaten in der Kneipe um die Ecke oder die Druckknöpfe an den Jeans der ersten Freundin auf dem Rücksitz des VW-Käfers waren.
Mit dem Computer sind die Spielchen erwachsen geworden. Plötzlich ist die Jagd auf die Ufos kein Kinderkram mehr, sondern wir bereiten uns vor »auf die beruflichen Aufgaben des nächsten Jahrzehnts«.
Gemessen an den meistverkauften Programmen im Bereich der Home-Computer wollen demnach die meisten Menschen in Zukunft Weltraumpiloten werden.

Ordnung muß sein!

Bis vor kurzem kokettierte ich noch gerne mit der nicht ganz wahrheitsgemäßen Bemerkung, ich besäße nur eine Diskette. Es war nämlich so, daß die paar Programme und Datensätze, die ich häufig brauche, sämtlich einträchtig auf meiner Workdisk versammelt waren. Nun hat mir aber Andreas Magnus eine Harddisk eingebaut, die sich wie 5 Laufwerke verhält. Klar, daß ich bei dieser bequemen Art des Zugriffs unterschiedliche Software-Pakete je einem Drive zugeordnet habe, um leichter die Übersicht behalten zu können. Früher hätte ich solche Ordnung nur mit dem leidigen Türchen-auf-Türchen-zu an den Floppy-Laufwerken halten können.

So weit die guten Nachrichten. Hier eine schlechte (eine zweite gute: die schlechte stimmt seit ein paar Minuten nicht mehr): Die SYSTEM-Optionen AN=xx und AO=xx legen fest, welches Laufwerk ohne weitere Parameter beim Befehl DIR und bei der Eröffnung neuer Dateien benutzt werden soll. Andreas hat das DOS so gepatcht, daß ein Teil-Laufwerk der Harddisk hierfür benutzt wird. Gut und schön. Meine Textverarbeitung mit allen Formularen liegt aber auf einem anderen. Neue Texte sollen auch dorthin. Es macht wenig Laune, jedesmal die Laufwerksnummer angeben zu müssen, um die liebe neue Ordnung auch durchzuhalten.

Übrigens tritt dieses Problem auch ohne Harddisk auf, falls ihr die Disziplin habt, grundsätzlich in Lw. 0 nur das System und in einem anderen Laufwerk z. B. den Assembler-Kram zu beharken.

Die Abhilfe ist so simpel, daß ich mich geniere, nicht schon seit Jahren damit zu arbeiten: Das zuletzt angesprochene Laufwerk ist in 4308h vermerkt. Wenn ich gerade ZEUS angekurbelt habe, der bei mir in Drive 6 (Harddisk) wohnt, steht dort also eine 6. Da sollen bitteschön auch alle Sources hingeschrieben werden, die ich gerade schöpfe. Also braucht nur der Inhalt von 4308h nach 43A0h (SYSTEM,AN=xx) und 43A1h (SYSTEM,AO=xx) kopiert zu werden, um das DOS auf das zuletzt benutzte Laufwerk einzuschwören.

Hacker sind faul, deshalb sollte das gerne automatisch passieren, sobald ZEUS (oder z. B. auch TSCRIPS in Lw. 7) aufgerufen wird. Dazu braucht nur eine kleine Maschinenroutine (s. Assemblerlisting) an das Hauptprogramm angehängt zu werden, die das beim Einprung mal eben erledigt:

Die Laufwerksnummer in 4308h wird geladen. Hier kommt das Register L zur Anwendung; wenn anschließend H aus L geladen wird, können gleichzeitig 16 Bit (zweimal die Laufwerksnummer) nach 43A0h und 43A1h kopiert werden. Auf diese Weise werden nur 10 Bytes für das Prográmmchen verschlissen. Nach dieser Verlade muß natürlich das Gastgeberprogramm besprungen werden. Seine Einsprungsadresse wurde zuvor aus dem Ende-Code in seinem letzten Disksektor ermittelt. Die kommentierten Sektordumps sollen diese Zusammenhänge aufklären.

Die alte Frage: Wohin mit solchen Erweiterungen? Beim einfachen Aufruf eines Programms wird vom DOS-Eingabepuffer nur ein bißchen verbraucht, je nach Länge der File Specifications. Der riesige Rest ist frei. Dort kann die Routine arbeiten (s. ORG).

Die fortgeschrittenen DOS-Frevler unter den Lesern werden sich nun fragen, wieso nicht mit lausigen 2 Bytes, nämlich

```
ORG     43A0h     ;Lw# für DIR und CREATE
DEFW    lw#lw#    ;zweimal die Lw#
```

derselbe Job, sogar ohne Rechenzeit erledigt werden soll. Deswegen: Die Diskette liegt vielleicht auch mal nicht im Lieblingslaufwerk. Und wenn das Programm von der RAM-Disk kommt, sieht es wieder anders aus. Daher war die etwas umständlichere Methode vorzuziehen.

Um die Erweiterung an ein Host-Programm anzuhängen, genügt leider nicht der APPEND-Befehl. Man kann es mit der CDD-Funktion von SUPERZAP tun oder "zu Fuß", indem man die Codes einzeln eintippt. Wenn dabei die Sektorgrenze überschritten wird, wird es leider haarig. Mann sollte dann mit APPEND eine kurze Datei, die höchstens einen Sektor lang ist, an das Hauptprogramm anhängen. Wenn der neue Sektor diskrelativ direkt an das Hauptprogramm anschließt, wird die ganze Schlußarie auf einmal mit CDD übertragen, sonst in zwei Raten bis bzw. ab Sektorgrenze. Eine direkte Manipulation am Inhaltsverzeichnis mag schneller sein, ist aber kritischer, besonders dann, wenn der neue Sektor in einem neuen Granule liegt.

Computer heißt auf spanisch *ordenador*, weil er ordnet. Meiner jetzt endlich auch.

Arnulf

```
424E            00001 program EQU   424eh         ;Startadresse Host-Programm, hier ZEUS
                00002
435E            00003         ORG   435eh         ;am Ende des DOS-Eingabepuffers
435E  2A0843    00004 start   LD    HL,(4308h)    ;L <- aktuelle Laufwerknummer
4361  65        00005         LD    H,L           ;HL <- dasselbe zweimal
4362  22A043    00006         LD    (43a0h),HL    ;jetzt Default-Lw. für DIR und CREATE
4365  C34E42    00007         JP    program       ;Host-Programm anspringen
435E            00008         END   start         ;dort Einsprung

00000 Fehler
```

ZEUS/CMD (letzter Sektor) vor der Änderung:
(Einsprungsadresse unterstrichen)

```
FRS C0  E521 0042 1174 0601 1900 DBFA F5E6 DFD3
39  D0  FAED B01E D20E 35ED B0F1 D3FA E13E 13CD
27H E0  3300 C312 5302 024E 4200 0000 0000 0000
    F0  0000 0000 0000 0000 0000 0000 0000 0000
```

Die Wurmfortsatz-Routine:
(die Einsprungsadresse von ZEUS/CMD und ihre eigene unterstrichen)

```
DRV 00  010C 5E43 2A08 4365 22A0 43C3 4E42 0202
2   10  5E43 0000 0000 0000 0000 0000 0000 0000
2H  20  0000 0000 0000 0000 0000 0000 0000 0000
```

ZEUS/CMD (letzter Sektor) nach der Änderung:
(Wurmfortsatz-Routine komplett unterstrichen –
Achtung! Der Ende-Code von ZEUS/CMD muß vom Record-Header des Anhängsels überschrieben werden!)

```
FRS C0  E521 0042 1174 0601 1900 DBFA F5E6 DFD3
39  D0  FAED B01E D20E 35ED B0F1 D3FA E13E 13CD
27H E0  3300 C312 5301 0C5E 432A 0843 6522 A043
    F0  C34E 4202 025E 4300 0000 0000 0000 0000
```

WORDSTAR AUF ALLEN LAUFWERKEN
===

Ist es Euch auch schon so gegangen? Diskette mit Wordstar ins Laufwerk, gestartet und freundliche Mitteilung, die Overlays sind nicht zu finden, obwohl sie da sind ?!

AHA, wieder mal falsches Laufwerk genommen. Diskette wechseln! Alles noch mal von vorn...

Das muß aber keinesfalls so sein. MicroPro hat bei WS lediglich ein paar Bytes "vergessen" einzubauen. Nun soll es ja Betriebssysteme geben, die diese Unterlassungssünde ausbügeln, mein CP/M aber nicht. Der FATALE ERROR liegt in der Adresse 02DCH von WS.COM begründet. Hier steht die Variable DEFDSK, die festlegt, auf welchem Laufwerk WS seine Overlays suchen soll. Steht dort nun "01" (entspricht Drive A) und die Diskette mit WS und seinen Overlays steckt in Drive B, dann sind die eben diese Overlays für WS nicht existent.

Abhilfe läßt sich durch eine Mini-Routine bei Start von WS.COM erreichen. Wie alte CP/M-er wissen, steht der Aktuelle Laufwerks-Code in den unteren 4 Bit der Adr. 0004. Der Inhalt dieser Adresse wird nur bei Aufruf der BIOS-Routine SELDSK beeinflußt. Nach dem Laden von WS.COM wird also dort noch der Laufwerks-Code des Laufwerkes stehen, in dem die Dikette mit WS gesteckt hat. Dieser Code braucht nur in DEFDSK eingetragen werden und schon "weiß" WS.COM auf welchem Laufwerk seine Overlays stehen.

Die einfachste Art diese Maßnahme zu realisieren ist das Umleiten des Startsprunges auf Adr. 0100H zu einer entsprechenden Routine und danach der Sprung zur Original-Startadresse. Die Routine kann in einem freien Bereich von MORPAT, dem Bereich in WS.COM für derartige User-Routinen, (Adr. 02E0-035BH) abgelegt werden.

```
0100    JP     NEWST          ;Start-Adr. ersetzen

NEWST:  LD     A,(0004)       ;Laufwerks-Code holen, A=00, B=01,...
        AND    0F,A           ;nur Bit 0-3
        INC    A              ;Laufwerk A=01, B=02,...
        LD     (02DC),A       ;Laufwerks-Code nach DEFDSK
        JP      OLDST         ;Sprung zu der Adresse, die im Sprung
                              ;auf 0100H im Original stand. Bei Ver-
                              ;sion 2 und 3 meist 2D08H
```

Mit DU oder ZSID ist diese Angelegenheit in 10 Minuten erledigt. Einen Schönheitsfehler hat der Vorschlag allerdings. WS muß vom aktuellen Laufwerk gestartet werden. Wenn etwa folgendes gemacht wird:

A>B:WS

wird WS in Adr. 0004 den Wert 00 (Drive A) vorfinden und eintragen. Unter der Annahme, daß WS und seine Overlays auf Drive B stecken, führt diese Methode mit Sicherheit ins Abseits.

Ursache dafür ist die Arbeitsweise des CCP, der nur für die Dauer des Ladevorganges der WS.COM auf Drive A umschaltet. Vor der Übergabe der Kontrolle an das Programm wird der alte Zustand wieder hergestellt und WS findet daher Drive B als aktuelles Laufwerk vor!

Funktionieren würde es so:

A>B:

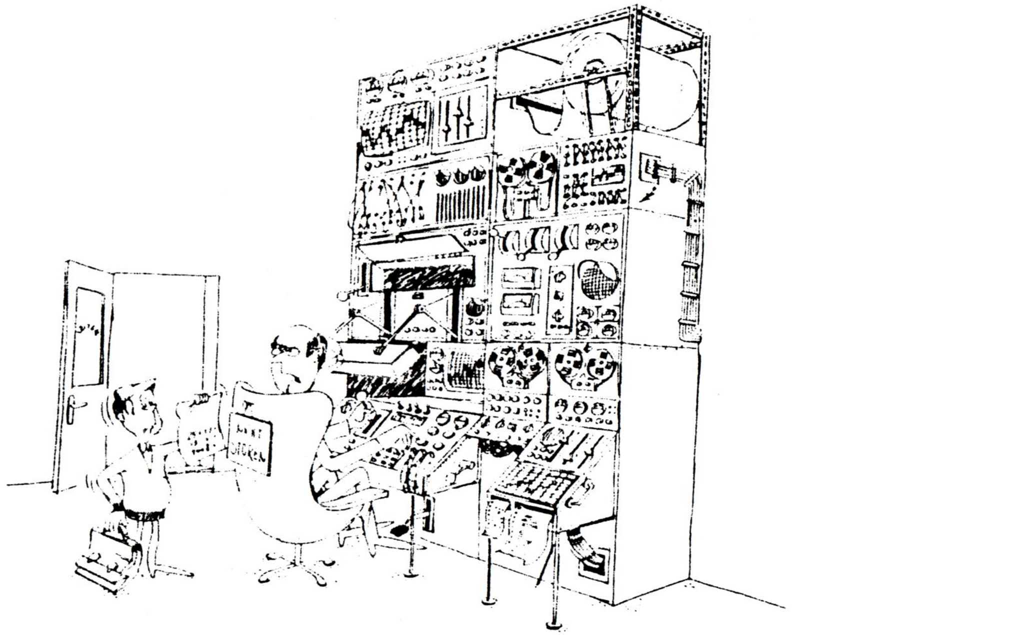
B>WS

Selbst dieser Haken ist, wenn auch mit etwas mehr Aufwand, vermeidbar. Geheimtip: BDOS-Fkt. 17! Wie das geht werde ich im nächsten Info vorstellen.

Noch ein kleiner Tip: Bei WS ist oftmals nicht zwischen harten Trennstrichen und weichen Trennstrichen, die durch Zeilenumbruch erzeugt werden, zu unterscheiden. Nach mehrmaliger Neuformatierung stehen die Trennstriche dann recht sinnverwirrend auf dem Schirm.

Der Zeichencode für den weichen Trennstrich ist in der Variablen SOFHYC (Adr. 03B8H) festgelegt. Der Code hat keinen Einfluß auf den Dateiinhalt oder das Druckbild, d.h. man kann ohne die Dateikompatibilität zu gefährden ändern. Zweckmäßig ist der Ersatz von "-" (2DH) durch "_" (5FH). Damit ist eine eindeutige Zuordnung möglich.

Frank-Michael Schober



»Ist mir egal, ob dein Opa hier der Chef ist – mach' deine Schulaufgaben allein!«

Teamwork

Durch einen Brief von Helmut Bernhardt bin ich auf folgendes Problem gestoßen: man hat eine große Datei mit Namen und Adressen (z.B. unsere Mitgliederverwaltung) unter DBASE II und möchte an eine bestimme Gruppe, z.B. die Nordlichter, ein Rundschreiben schicken, wozu man natürlich (?) Wordstar und Mailmerge nimmt. Wie bringt man die beiden Programme nun zur Zusammenarbeit, damit man nicht alles per Hand abtippen muß ? In einer alten c't (8/84) stand, wie's geht. Ausgehend davon habe ich ein kleines Beispiel gebaut, an dem man hoffentlich sieht, was dabei alles beachtet werden muß. Zuerst sollte man sich die Länge der einzelnen Felder im DBASE ausgeben lassen, sonst gibt's bei dem Vergleich weiter unten Probleme.

```
.. Drucken von Adressenaufklebern mit den Daten aus der Mitgliederverwaltung
.. unter WordStar 4.0
..
..      Erzeugen des Files in DBASE mit:
..
..      .USE pdaten                        <- hier stehen die Adressen
..      .DISPLAY STRUCTURE                 <- Länge der Felder anzeigen
..      .COPY TO xxx DELIMITED WITH ,      <- Komma nicht vergessen !
..
..      Variable         Länge
..
..      NAME        C     020
..      VORNAME     C     020
..      STRASSE     C     025
..      LAND        C     003
..      PLZ         C     005
..      WOHNORT     C     025
..      TELP        C     020
..      TELG        C     020
..      GEBOREN     C     010
..      EINTRITT    C     010
..      BEITRAG     L     001
..      NUMMER      N     004
..
.. Feld im IF muß exakt die Länge haben, die oben angegeben ist !
..
.DF m:xxx.txt                <- diese Datei wird eingelesen
.RV
name,vorname,strasse,land,plz,wohnort,telp,telg,geboren,eintritt,beitrag,
       nummer                <- alle Elemente einlesen
.IF &plz& #<4000             <- z.B. alle Nordlichter. Hinter der '4000' ist
Herrn                        <- ein Leerzeichen, da das Feld 5 Zeichen enthält
&vorname& &name&
&strasse&
&plz& &wohnort&

.EI                          <- Ende des IF
```

Einzelne Aufkleber kann man natürlich einfacher direkt aus DBASE drucken, aber wenn dann noch ein kleiner Brief dazu kommt, braucht man doch ein Textprogramm. Ebenso kann man auch Ausgaben des DBASE-Listengenerators in den Texteditor laden und noch nach Wunsch nachbearbeiten, z.B. um die Texte einzudeutschen oder Kommentare anzufügen, indem man die Ausgabe mit 'SET ALTERNATE TO listout' und 'SET ALTERNATE ON' zusätzlich auf die Floppy lenkt.

Alexander Schmid

* * * TSCRIPS retten * * *

Kurt Müller ergänzte in Heft 26, S.18, meine früheren Ausführungen, wie man einen verunglückten TSCRIPS-Text rettet, durch den Hinweis auf den (NEW)DOS-Befehl <u>DUMP</u>.
Hiermit läßt sich das von mir sz. geschilderte Verfahren wesentlich vereinfachen - vorausgesetzt, daß der Text im Speicher noch nicht zu sehr verunstaltet oder gar ganz kaputt ist.
Leider beschreibt Clubkollege *Kurt* das Vorgehen nicht im Detail, insbesondere nicht, wie man den korrigierten Text nun wieder in das Textsystem TSCRIPS integriert. Ich habe einige Zeit herumprobiert, bis ich es heraushatte. In der Vermutung, daß auch andere mit dem DUMP-Befehl nicht so vertraut sind, möchte ich das Vorgehen hier Schritt für Schritt kurz aufzeigen. Sicher kann man es außer für TSCRIPS auch zum "Retten" anderer Programme anwenden.
Es funktioniert so:

1) Gehe ins DOS
2) Lade SUPERZAP
3) Wähle DM (Display Memory). Der TSCRIPS-Text beginnt in Speicherstelle 9177H (37239d).
4) Korrigiere die Fehler mit MOD (es braucht nur das letzte Byte der Speicheradresse eingegeben zu werden)

Falls gleich weitergeschrieben wird ==> gehe nach 5)
falls der korrigierte Text gespeichert werden soll ==> 5a)

5) Verlasse SUPERZAP mit EXIT
6) Lade TSCRIPS und wähle A<LT>. Drücke SHIFT/Hochpfeil, dann CLEAR

Der korrigierte Text sollte jetzt auf dem Bildschirm sein!

---

* E N D E *

5a) Notiere die Adresse des letzten zu speichernden Bytes nach der korrigierten Stelle = Endadresse.
6a) Verlasse SUPERZAP
7a) Gib ein:

DUMP FILENAME/CMD:dr,37239,Endadresse

8a) Sobald der korrigierte Text mit TSCRIPS weiterverarbeitet werden soll, gib im DOS ein: LOAD FILENAME/CMD:dr
9a) Gehe nach 6)

Kurt, schönen Dank für Deinen Hinweis!

(Man wird alt wie'n Haus
und lernt nicht aus.
Grüß Gott! Jot-Klaus)

Vergleich BASIC mit TURBO-PASCAL

Der Vergleich "VW gegen PORSCHE" ist überhaupt *kein* Vergleich dagegen. Eher schon der Vergleich "Käfer gegen Windhund"! Wer sich noch nicht mit TURBO-PASCAL befaßt hat, sollte sich einmal für diesen ersten kurzen und bescheidenen "Erfahrungsbericht" interessieren. (Wer TP kennt und auch benutzt, sollte zum nächsten Beitrag weitergehen: Er findet hier nichts Neues!)
Als steinalter BASIC-Freak - und auch heute noch der Überzeugung, daß man mit BASIC *alle* Probleme lösen kann und nur dann eine andere Sprache braucht, wenn man es eiliger hat (oder auf irgendeiner engen Schiene dahinfährt) - habe ich mich nun endlich mit *TURBO-PASCAL* befaßt! Das war mit meinem (ur)-alten TRS-80 Model 1 allerdings nur möglich, seit ein geduldiger und aufopferungsbereiter Clubkamerad mir eine CP/M-Karte verpaßt und die nötige Software dafür spendiert hat (oder muß es "CP/M-Banker" heißen? Ich glaube nicht, denn mein CP/M+ ist nicht "gebankt". Aber dieses Wissen ist - mir jedenfalls - völlig unwichtig. Als kleiner "User" bitte ich daher den Fachmann ggf. um Verzeihung!)

Um künftig ein TP-User zu werden, habe ich zunächst einige dünne sowie dicke Bücher über PASCAL studiert; für Interessierte seien sie hier aufgeführt:

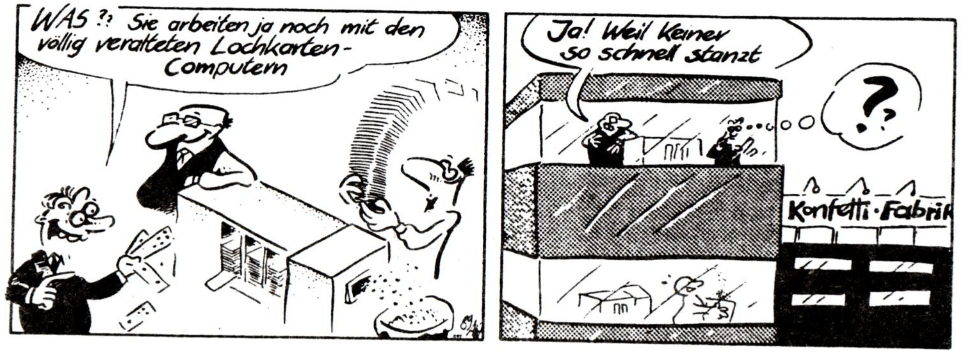
- A.Schmuck: PASCAL - Schritt für Schritt DM 8.80
  (Humboldt-Taschenbuch Nr.551)
- R.Baumann: Programmieren mit PASCAL (antiqu.DM 13.-)
  Vogel-Verlag (sog."CHIP-Wissen")
- H.Feller : PASCAL-Datenstrukturen u. Anwendungen
  Luther-Verlag; auf einer Messe gekauft; Preis?
  (Dieser Verlag scheint ja inzwischen pleite zu sein.)

Speziell für *TURBO-PASCAL* empfehle ich jedoch:
K.-H.Rollke: Grundkurs Turbo-Pascal, Band 1 & 2, je DM 29.80
Das zweibändige Werk ist bis Version 4 zu benutzen. (Inzwischen gibt es schon Version 5; aber das ist für den Einsteiger sicher wurs<ch>t).
Dieses im SYBEX-Verlag erschienene Werk scheint mir am klarsten geschrieben und ist so recht geeignet für geistige Schnecken wie mich. Schwerwiegende Druckfehler halten sich in Grenzen (ich fand bisher nur drei), so daß man das Buch (289 + 320 Seiten) auch nach längerem Lesen noch ertragen kann. SYBEX scheint mir ohnehin der beste EDV-Verlag zu sein.
Ein guter Clubkamerad kopierte mir außerdem 375 Seiten eines Werkes (das hier nicht genannt werden soll), auf dessen Lektüre an Winterabenden ich mich schon besonders freue, weil es recht wissenschaftlich anmutet...

Zu Beginn der Lektüre erschien mir als schlichtem BASIC-Jünger alles furchtbar umständlich. Du meine Güte: Jede noch so kleine Handlung ("Prozedur") bekommt ihre eigene Überschrift miteinem eigenen Namen; dauernd muß etwas erklärt ("definiert") werden; kleine Selbstverständlichkeiten, die man unter NEWDOS sich einfach aus dem ROM holt, müssen ausführlich Punkt für Punkt beschrieben (programmiert) werden, um dann als selbständige "Funktion" zu fungieren (unter NEWDOS meist schlicht mit der Funktion 'DEFFN' darstellbar) und so weiter und auch nicht weiter...

Der "enorme" Vorzug der zwangsläufigen Strukturiertheit dieser förmlich starr fixierten Sprache kann nur demjenigen imponieren, der sich bisher bei seinen Programmierbemühungen stets in einem Urwald von GOTO-Befehlen, Schleifen und sonstigen Anweisungen verlaufen hat und gar nicht gemerkt hat, daß er daran selber schuld war, weil sein Geist so undiszipliniert war. In der Tat: "Strukturiert programmieren" kann man in BASIC auch! Ich habe das in einem früheren Beitrag gezeigt.
Natürlich habe ich das eine und andere TP-Programm von *Rollke* abgeschrieben und ausprobiert - kritisch. Nun aber wollte ich es genau wissen, wo der Vorteil liegt und warum ich mein Hirn mit neuen Vokabeln quälen sollte: Was kann BASIC eigentlich <u>nicht</u> im Vergleich mit Turbo-Pascal?
Ich nahm mir also eins von den ein bißchen mathematischen Programmen des Herrn Rollke vor (er wird mir verzeihen, hoffe ich), welches sich auf S.186 des ersten Bandes befindet. Nachdem man zwei ganze Zahlen eingegeben hat, sagt dieses Programm, welche von beiden "die kleinere" ist (was ja für einen Erstkläßler wirklich enorm schwer zu übersehen ist! Gottlob gibt's dafür Computer...). Es verkündet aber auch, welches der "ggT" (der größte gemeinsame Teiler) dieser beiden Zahlen ist (dies erfordert i.a. schon einen höheren Grad mathematischer Bildung!) und schließlich offenbart es dem Zuschauer, ob die eine Zahl eine Primzahl ist oder nicht, und verrät den gleichen Sachverhalt danach auch hinsichtlich der zweiten eingegebenen Zahl!
Ich hätte mir zwar gewünscht, in jenen Fällen, in denen die eingegebene Zahl *keine* Primzahl ist, auch noch zu erfahren, wodurch sie denn nun teilbar ist - aber das sollte dem Adepten sicher noch nicht zugemutet werden: vielleicht hätte es das Programm gesprengt...
Nun programmierte ich das gleiche Thema in BASIC - nichts leichter als das! Hierbei habe ich mich bewußt nicht um besondere "Strukturierung" bemüht, sondern etliche "GOTOs" benutzt!
Nachdem ich mein BASIC-Programm getestet (man mußte ja einbauen, daß die "1" nicht versehentlich als Primzahl und die "2" nicht als Nichtprimzahl bewertet wird sowie daß die "1" und die eingegebene Zahl selbst sog. "unechte" Teiler sind und daher außer Konkurrenz liegen, d.h. aus dem Rennen auszuscheiden sind; solcherlei Kleinkram kriegt man natürlich erst nach einigen Tests hin) -
aber nun konnte es losgehen: <u>Stoppuhr zur Hand!</u>
Was heißt hier Stoppuhr?
Es genügte der Sekundenzeiger meiner Armbanduhr, die "ich, wo ich gehe, stets bei mir trage" (*Löwe*) - und nachher sogar mußte der Minutenzeiger herbei - je nachdem, was für ein Zahlenpärchen ich eingab. Folgende Liste zeigt die Geschwindigkeits-, genauer gesagt: die Zeitunterschiede zwischen beiden Sprachen:

Gewählte Zahlen		Zeitbedarf in Sek. BASIC	Turbo-Pascal
117	226	3	< 0.5
551	323	8	< 0.5
1001	557	12	0.5
11107	6543	145 (!)	5



Das letzte Beispiel zeigt den Zeitgewinn unter TP überdeutlich: er ist das 29fache! Diese lockeren Beispiele ließen sich bis zur größten darstellbaren Ganzen Zahl (65535) erweitern; doch hatte ich keine Lust mehr, auf mein schläfriges BASIC-Programm zu *warten*. Offenbar steigt der Zeitgewinn logarithmisch an; ich will das jetzt nicht genauer untersuchen.
(Sicher ist er aber nicht so groß, wie er bei Umsetzung des Programmes in Maschinensprache wäre - gell, *Arnulf*?)

Beide Programme findet ihr zum Vergleich anschließend.
Nunmehr werde ich mich aufgrund der Anhebung der Geschwindigkeitsbegrenzung innerhalb meines ~~TRS-80~~-Dorfes mit eitler Lust ins TP-Vergnügen stürzen - denn "Zeit ist Geld" (ZIG). Und billiger kann ich nicht zu Geld kommen.
Wer's noch nicht tut: Wer möchte nicht mit mir Zeit (Geld) sparen? * Der melde sich zum Erfahrungsaustausch! Eins kann ich dem Neuling - falls es ihn noch gibt - versprechen: Er kann aus seinem VW-Käfer in einen TURBO-PORSCHE umsteigen! Fast kostenlos...
Dennoch werde ich auch in Zukunft auf mein Allerwelts-BASIC nicht ganz verzichten; denn:

Wer die Qual
einer Wahl
scheut, der nehme allemal
einfach TURBO-BAS(I)CAL!

Grüß Gott!
Ka-Jot

P.S. Die wesentliche Ursache für die Beschleunigung ist der Umstand, daß Turbo-Pascal eine Compiler-Sprache ist. [BASIC kann aber oft auch kompiliert werden!]

Das Primzahl-Programm in TURBO-PASCAL

```
program primzahlGgt;

var zahl1, zahl2 : integer;
    wort : string(.5.);

function min (a,b:integer):integer;
 begin
  if a<b then min:=a
         else min:=b
 end;

function ggT (a,b:integer):integer;
 var teiler:integer;
  begin
   for teiler:=1 to min(a,b) do
     if a mod teiler = 0 then
     if b mod teiler = 0 then ggT:=teiler;
  end;

function prim(zahl:integer):boolean;
 var teiler:integer;
     test:boolean;

 begin
  teiler:=3;
  test:=true;
    if zahl<2then test:=false else
     if zahl>3then
      if zahl mod 2=0 then test:=false else
       repeat
        if zahl mod teiler=0then test:=false;
        teiler:=teiler+2;
       until teiler>round(sqrt(zahl));
      prim:=test
     end;

function weiter:boolean;
 var ch:char;

 begin
   write('weitermachen (J/N) ?');
   read(ch);
   weiter:=(ch='J')or(ch='j');clrscr
 end;

begin
clrscr;
 repeat
  write('Gib eine erste Zahl ein: ');
  readln(zahl1);
  write('Gib noch eine Zahl ein: ');
  readln(zahl2);
  writeln('Der groesste gemeinsame Teiler');
  writeln('ist ggT = ',ggT(Zahl1,Zahl2):3);
  if prim(zahl1) then wort:='eine' else wort:='keine';
  writeln(zahl1:3,' ist ',wort,' Primzahl');
  if prim(zahl2) then wort:='eine' else wort:='keine';
  writeln(zahl2:3,' ist ',wort,' Primzahl')
 until not weiter
end.
```

Das Primzahl-Programm in BASIC

```
 10 CLS
    PRINT "              PRIMZAHL oder nicht ?"
    DEFINT A - N
    PRINT
    PRINT "Bitte zwei ganze Zahlen <A, B> eingeben!
 20 INPUT A,B
    IF A < B
    THEN MIN = A
    ELSE MIN = B
 30 PRINT MI"ist das Minimum"
    FOR N = 1 TO MI
    IF ((A / N = INT (A / N)) AND (B / N = INT (B / N)))
    THEN GG = N
 40 NEXT
    IF GG < > 1
    THEN PRINT "Der ggT ist"GG

    ELSE PRINT "Kein gemeinsamer Teiler!
 50 IF A = 1
    THEN PRINT "Die 1 ist keine Primzahl!"
         GOTO 110
 60 IF A = 2
    THEN GOTO 90
 70 N = 2
    GOSUB 120
 80 FOR N = 3 TO SQR (A) STEP 2
    GOSUB 120
    NEXT
 90 PRINT A"ist eine Primzahl"
    GOTO 110
100 PRINT A"ist keine Primzahl -"C"ist der größte Teiler"
    [redacted]
110 IF F = 1
    THEN END
    ELSE A = B
         F = 1
         GOTO 50
120 IF A < > N AND A / N = INT (A / N)
    THEN C = A / N
         GOTO 100

    ELSE RETURN
```


Backen und Banken

von Rüdiger Sörensen.

Um eine Diskussion auf dem letzten Clubtreffen und verschiedene Aufrufe verschiedener Mitglieder noch einmal aufzugreifen, will ich ein Zitat aus einem in letzter Zeit recht erfolgreichen Film etwas abgewandelt zum besten geben : "Die Hauptaussage des CLUB 80 ist nicht : 'Jeder kämpft für sich allein.'"

Mir ist aufgefallen, daß vor allen anderen Ka-Jot die Problematik des Bankens untersucht und mit Programmen verdeutlicht hat. An diesem Punkt kann ich etwas hinzufügen, das dem einen oder anderen vielleicht hilft, auf alle Fälle aber zur Diskussion anregen soll! In meinem Model 4 Tischmodell ist ein RAM-Erweiterung um 256K eingebaut, die Anleitung stammt aus der 80 MICRO. Damit verfüge ich über insgesamt 320K RAM, die, wie üblich, nur zu jeweils 64K erreichbar sind. Unter TRSDOS kann man damit eine 256K RAMDisk einrichten, mehr ist mit meinen bescheidenen Systemkenntnissen nicht machbar. Aber unter CP/M! Denn da ist alles, wenn schon nicht selbstgemacht, so doch im Quellcode vorhanden und weitgehend verstanden. Wie bereits mehrfach erwähnt, gliedert sich CP/M in 3 Teile, CCP, BDOS und BIOS. Letzteres nimmt im Model 4 ca. 6K in Anspruch. Davon bestehen 1.5K aus Diskettenpuffern und ähnlichen Bereichen (DPH, DPB etc.).

Die einzelnen Routinen des BIOS habe ich nach Funktionen gegliedert :

CRESET	Kaltstartroutine
WRESET	Warmstartroutine
DISKIO	Lesen und Schreiben einzelner Sektoren
PPDRV	Druckertreiber
SPDRV	RS232-Treiber
VDDRV	Video-Treiber
KBDDRV	Tastatur-Treiber

Das Memory-Layout des Model 4 sieht so aus :

0000H-7FFFH	untere 32K-Bank
8000H-FFFFH	obere 32K-Bank
F800H-FFFFH	Memory-Map für Bildschirm und Tastatur

Die Memory-Map läßt sich Port-gesteuert ein- und ausschalten. Des weiteren wird über einen Port gesteuert, welche der jeweiligen 64K aus der 256K-Erweiterung gerade aktiv sind und als alternativer Speicher parallel zum 64K-Hauptspeicher zur Verfügung stehen. Ferner kann man aus eben den 64K Alternativspeicher eine Bank von 32K in den Hauptspeicher einblenden, und zwar in die obere oder untere Adreßhälfte. Eine 256K Ramdisk ist dann recht einfach zu installieren, aber das reicht mir nicht. Denn man kann ja auch Programme im Alternativspeicher ablegen!

Und genau das habe ich getan. Teile meines BIOS sitzen nicht im Hauptspeicher sondern im AS (Alternativspeicher). Bekanntlich muß das BIOS oberhalb des BDOS residieren. Im Model 4 ist das eh zwingend, wenn man überhaupt Banken will. Also muß in den residenten BIOS-Teil eine Routine, welche das Banken übernimmt, in diesem Fall wird die untere 32K-Bank

ausgewechselt. Für den Transfer von Daten ist das gar kein Problem, es muß lediglich ein Puffer im residenten Teil sitzen, über den der Transfer laufen kann. Beispiel RAMDisk : Wenn ein Anwenderprogramm die DMA-Adresse in der unteren Bank sitzen hat und einen Sektor dorthin haben will, muß das BIOS zuerst die entsprechende Bank berechnen, diese in die untere Adreßhälfte einblenden und den richtigen Sektor in den residenten Puffer laden, die TPA-Bank wieder einblenden und den Puffer in die DMA transferieren. Das hört sich komplizierter an, als es ist.

Man stelle sich nun vor, das einige Routinen des BIOS selbst nicht resident sind, sondern in einer AS-Bank sitzen. Diese kann sich natürlich nur in der unteren Adreß-Hälfte befinden, da ja die Umschaltung vom residenten BIOS in der oberen Hälfte erledigt wird. Die gebankten Routinen müssen ferner auch in dem unteren Adreßraum lauffähig sein, das heißt, mit ORG ØØØØH oder was immer assembliert werden. Welche Routinen kommen nun in Frage? Die DISKIO-Routinen sind schwierig, da sie ihre Puffer im residenten Teil halten müssen. Auch aus einem anderen Grund, nämlich der dynamischen Änderbarkeit des gebankten BIOS, sollte man zunächst davon absehen. CRESET scheidet aus, denn hier wird der gebankte Teil initialisiert und auch (bei mir) in die Bank geladen. WRESET lohnt sich nicht, geht auch nicht, wie später noch zu berichten sein wird.

Bleiben noch die CHARIO-Routinen. Ich habe mich für folgendes Vorgehen entschieden :

Beim Booten des Systems wird CCP, BDOS und BIOS in den residenten Teil des Speichers geladen. Das BIOS enthält allerdings nur rudimentäre CHARIO-Routinen, welche für die untere Adreßhälfte assembliert sind und also oben gar nicht laufen können. Nach dem Laden des BIOS sitzen diese Routinen an der Stelle der Diskettenpuffer, können also deren Länge nicht überschreiten. CRESET schaltet dann die AS-Bank ein, in der das gbankte BIOS sitzen soll, transferiert die Routinen dorthin und initialisiert anschließend die Disk-Buffer. Die Sprungleiste im residenten Teil verweist natürlich nicht auf die gebankten Routinen, sondern auf kleine residente Treiber, die das Banken übernehmen und die entsprechende Routine anspringen.

Im letzten Info stellte Frank-Michael Schober den RSX-Trick vor, mit dem das Nachladen des CCP und BDOS verhindert werden kann. Mit Banken was besseres : CRESET schafft CCP und BDOS ebenfalls in eine Bank, dann kann WRESET aus dem Speicher nachladen! Das geht in Nullzeit (fast). Ein Nachteil der RSX-Methode ist, daß der CCP nicht mehr überschrieben werden kann und damit 2K für Anwenderprogramme verloren gehen. Außerdem bin ich nicht ganz sicher, daß kein Programm das BDOS überschreibt, meines Wissens macht das z.B. TURBO-MODULA2. Aber das ist sicher eine Ausnahme, wenn überhaupt. Weiß jemand Näheres?

Das größte Problem beim Verwenden gebankter Routinen ist der Stack. Wenn er in der unteren Bank residiert, falls XBIOS gerufen wird (sei XBIOS der gebankte Teil des BIOS), resultiert unweigerlich ein Systemabsturz daraus. Die residente Bankroutine muß also unbedingt den Stack sichern und einen Neuen anlegen. Bei mir wird zuerst die XBIOS-Bank eingeschaltet und dann der Stack dorthin verlegt. Dann die Routine geCALLt (grundsätzlich, um wieder in der oberen Bank zu landen, JP's sind sehr gefährlich!) und anschließend der

Stack wieder restauriert und die TPA-Bank eingeschaltet. Das funktioniert einwandfrei. Ein Problem taucht auf : Das BIOS ist nicht reentrant. Man muß sicherstellen, daß Interrupts ausgeschaltet snd, solange in der Bank gewerkelt wird, und daß XBIOS sich nicht selbst via BIOS-Sprungleiste ruft, da dann der Stackpointer verloren geht. Eine entsprechende Verwaltung des Stacks ist viel zu aufwendig, und diese zwei Einschränkungen sind gering.

Was habe ich denn nun damit gewonnen ? Nun, ca. 1.5K an TPA, denn den nehmen nun die gebankten Teile nicht mehr weg. Das ist nicht viel, man muß auch noch etwas abziehen für die hinzugekommenen Verwaltungsroutinen.

ABER!

Wenn man das gebankte XBIOS etwas geschickt aufbaut, ist es änderbar! Bei mir startet das XBIOS bei Adresse ØØØØH mit einer Sprungleiste, genau wie sein residentes Gegenstück. Das hat den Vorteil, daß BIOS gar nicht wissen muß, wo die XBIOS-Routinen liegen, sondern einfach in die Sprungleiste hüpft, die damit das Einzige ist, das in der Reihenfolge und Adreßlage immer beibehalten werden muß. Solange diese Bedingung beachtet wird, kann man nach Herzenslust ein neues XBIOS laden! (Deswegen habe ich auch die Disk-Routinen resident gelassen, sie würden dabei überschrieben werden). Also ein kleines Programm geschrieben, das ein XBIOS-File überprüft, ob es ein solches ist, und an die richtige Stelle schafft. **SOFORT NACH DEM KALTSTART TRITT DIESES PROGRAMM IN AKTION!** Und überhaupt, wie groß darf denn jetzt das BIOS werden? Auf die Systemspuren muß es ja nicht mehr passen, wir haben eine 32K Bank abzüglich CCP und BDOS sowie etwas Stack zur Verfügung! Massig Platz! Zusätzlich zu den CHARIO-Routinen habe ich die Sprunglast im residenten BIOS um einen Sprung erweitert. Dieser führt wieder in das XBIOS, wo anhand des Akku-Inhaltes ein Programm ausgesucht und gestartet wird. Dabei werden BC, DE und HL als Parameter aufgefaßt, je nach Routine. Man kann dahin die Grafik stecken, oder, wie bei mir, Routinen zum Stellen der RS232-Parameter, Modemsteuerung und so weiter. Außerdem kann man die Treiber fast beliebig erweitern. Mein Video-Treiber ist jetzt ein Fenster-Treiber. Default ist ein 8Ø*24 Fenster. Man kann die Grenzen setzen, wie man will, alle Operationen, Zeichenausgabe, Cursorsteuerung, CLS, CLEOL, DELLIN, INSLIN, SCROLL arbeiten im jeweils aktiven Fenster und kein Anwenderprogramm ahnt auch nur etwas davon!

Der Tastaturtreiber kann zwischen deutscher und US-Tastaur umschalten, es gibt eine Bildschirm-Hardcopy-Routine, der Druckertreiber erkennt Grafik-Modus und steuert bei 8-Nadel-Grafik 24 Nadeln an, Tastaturklick (abschaltbar), der Phantasie sind keinerlei Schranken auferlegt.

Ich gerate ins Schwärmen. Noch eine Idee : bekanntlich liegen CCP und BDOS inzwischem als leistungsfähige Z8Ø-Versionen und im Quellcode vor. Warum also nicht CCP und BDOS ebenfalls um gebankte Teile erweitern? Kommandozeilen-Editor? Disk-Buffering? Wo liegt die Grenze ?

An die Arbeit ! *Rüdiger*

Nochmaliges Banken

Ich hatte beschrieben, wie es mir gelungen ist, mehr TPA zu ergattern, indem ich Teile des BIOS gebankt habe. Inzwischen hat sich da einiges geändert. Mein Spezi Lucius, Besitzer eines Model 4 Tischmodells mit 320K RAM, wie ich auch, regte mich an/auf, doch das gesamte Betriebssystem in eine ausführbare Datei zu stecken, so daß ein neues OS einfach durch Aufruf eines Programmes gestartet werden kann. Das habe ich getan. Auf meinen Disketten liegen jetzt zwei Files namens BOOT.COM und ZBOOT.COM rum. Der erste lädt ein 57K TPA System mit Z80DOS 2.34 und Z80CCP. Der zweite lädt ein ZCPR33 System mit 55K TPA und demselben BDOS. Zusätzlich zu den gebankten Teilen aus dem ladbaren XBIOS (wie beschrieben) habe ich nun auch den Floppy-Treiber gebankt. Das XBIOS kann folgendes : Fensterverwaltung, d.h. Cursoransteuerung, Ausgabe von Zeichen, scrolling, insert/delete Line erfolgen in einem definierbaren Fenster. Es können unterschiedliche Fenster definiert werden und das jeweils aktive Fenster gewechselt werden. Der Tastaturtreiber unterhält einen 128 Byte Puffer. Dieser kann von außen mit beliebigen Strings gefüllt werden. Dadurch läßt sich eine Pseudo-Batch-Steuerung realisieren, die sogar direkten Tastatur-I/O unterstützt. Außerdem sind alle Tastaturcodes in einer Tabelle abgelegt, die änderbar ist. Man kann also seine WordStar-Tastaturbelegung laden und danach zur Standardbelegung zurückkehren. Der Video-Treiber ist umschaltbar, auf Tastendruck zeigt er entweder die deutschen Umlaute nebst 'ß' oder die amerikanische Belegung mit geschweiften/eckigen Klammern an. Das geht natürlich nicht bei memory-gemappten Editoren. Es ist übrigens prinzipiell unmöglich, CP/M *beides* auf einmal darstellen zu lassen. XBIOS unterstützt auch die Ausgabe von Zeichen in die HRG. Momentan bin ich gerade dabei, Grafik-Primitive einzubauen. Ferner ist es möglich, Speicher in der XBIOS-Bank zu allokieren und zu lesen/beschreiben. Die RS232-Parameter können gesetzt werden. Der Druckertreiber erkennt Grafik-Codes und setzt sie in 24-Nadel-Grafik um.

Das ganze System benötigt eine 32K-Bank. (It's nice to have 32K for your BIOS...). Damit bleibt mir noch eine RAM-Disk von 224K. Warmstarts laden CCP aus der RAM-Bank nach.
Ich möchte anregen, dieses XBIOS-System zu standardisieren, um eine einheitliche Schnittstelle für Anwenderprogramme zu schaffen. Dazu eine Kurzbeschreibung :

Die Einsprünge in XBIOS erfolgen über einen RST 28H. Diese Adresse ist von DRI freigegeben, es darf damit keine Probleme geben. Beim Warmstart wird der Vektor bei 39H neu initialisiert, falls ein Programm ihn doch einmal zerstört haben sollte. Der Einsprung in's XBIOS muß folgenden Konditionen genügen :
In A steht die Nummer der gerufenen Funktion (0..255).
A = 0 ist die Initialisierung des XBIOS. Nicht implementierte Funktionen RETurnen einfach. Alle anderen Register können zur Parameterübergabe benutzt werden. Register B hat eine Sonderstellung : hier kann eine Unter-Funktion übergeben werden. Alle Register außer IX und IY werden verändert. Über Fehlerbedingungen habe ich mir noch keine Gedanken gemacht, ich denke, das BIOS-Fehlerschema sollte übernommen werden. Die Vergabe der Funktionsnummern an die entsprechenden Aktivitäten ist bei mir historisch bedingt, über Änderungen lasse ich gerne mit mir reden. Hier eine Liste :


```
Nummer in A              Beschreibung

0.......................Initialisierung des XBIOS
1.......................N/A
2.......................N/A
3.......................N/A
4.......................N/A
5.......................N/A
6.......................BaudSet

         IN : C = Baudrate Code.
         OUT : -

         Bedeutung von C :

                 0 = 50 bd
                 1 = 75 bd
                 2 = 110
                 3 = 134.5
                 4 = 150
                 5 = 300
                 6 = 600
                 7 = 1200
                 8 = 1800
                 9 = 2000
                 10 = 2400
                 11 = 3600
                 12 = 4800
                 13 = 7200
                 14 = 9600
                 15 = 19200

7.......................spopts

         Set Serial Port Options

         IN : C = Option Byte

         Bits :
                 0 : 1 = RTS
                 1 : 1 = DTR
                 2 : 1 = BREAK
                 3 : 1 = disable parity
                 4 : 1 = 2 Stop Bits, 0 = 1 Stop Bit
                 5+6 : Word Length
                 7 : 1 = even parity

8.......................WDefine

         IN : HL, BC

                 H = linke Spalte
                 L = obere Zeile
```

B = Anzahl Spalten
C = Anzahl Zeilen

!!! A C H T U N G !!!

Es werden keinerlei checks auf korrekte Parameter durchgefuehrt! Insbesondere führen zu kleine oder zu grosse Fenstergrenzen zum Systemabsturz!

9......................MAlloc

IN : HL OUT : HL

Speicher in der XBIOS-Bank allokieren.

Input : HL = Anzahl der zu allokierenden Bytes
Output :
HL = 0000H : Fehler
HL # 0000H : Adresse des Speichers

10......................MFree

Allokierten Speicher wieder freigeben

IN : HL OUT : HL

Input : HL = Adresse des freizugebenden Speicherbereichs
Output :
HL = 0000H : Fehler
HL # 0000H : Speicher freigegeben.

11......................FKBuf

Den Tastaturpuffer mit einem String fuellen.
IN : HL = Zeiger auf Buffer.
HL muss groesser als 7FFFH sein.
Der Buffer darf hoechstens 128 Byte gross sein und muss mit 00H enden.

12......................RetBufAdr

HL liefert die Adresse des 128 byte langen Buffers (WkBuf) in der oberen RAM-Haelfte, der z.B. fuer XBIOS-Datentransfer benutzt werden kann.

13......................MovBuf

Kopiert den Inhalt des durch (12) gegebenen Buffers WkBuf an die durch HL spezifizierte Stelle in der unteren XBIOS-Bank oder umgekehrt.

IN :

HL = Zeiger auf Buffer (HL > 07FFF -> Fehler)
C = 0 : XBIOS -> WkBuf
C <> 0 : WkBuf -> XBIOS
B = Anzahl der zu uebertragenden Bytes.
Falls B > 128, werden 128 bytes uebertragen.

14......................GetWin

Gibt in HL einen Zeiger auf den aktuellen Fenster-DCB zurueck.

15......................SelectWin

Selektiert einen neuen Fenster-DCB

IN : HL

HL = Zeiger auf neuen Fenster-DCB

16......................NewWin

Erzeugt einen neuen Fenster-DCB.

OUT : HL

HL = Zeiger auf neuen DCB (HL = 0 , falls Fehler).
Die Parameter des neuen DCB entsprechen denen des gegenwaertig aktiven DCB.
Der neue DCB wird nicht selektiert. Das muss explizit durch Aufruf von (15) getan werden.

17......................RestoreDefaults

Loescht alle Fenster-DCB's ausser dem urspruenglichen.
Traegt default-Werte in den DCB ein (80*24).
Alle allokierten Speicherbereiche werden freigegeben.

18......................N/A
19......................N/A
20......................N/A
21......................N/A
22......................N/A
23......................N/A

24......................Grafik-Zeichen schreiben u.a.

IN : B : Funktionsunternummer

B = 0 : Schreibe Zeichen in C in Grafikkarte
B = 1 : Setze Grafik-Cursor nach
H : Spalte (0..79)
L : Zeile (0..239)
B = 2 : Gib zurueck Cursor-Adresse
H : Spalte
L : Zeile
OUT : HL, je nach Funktion

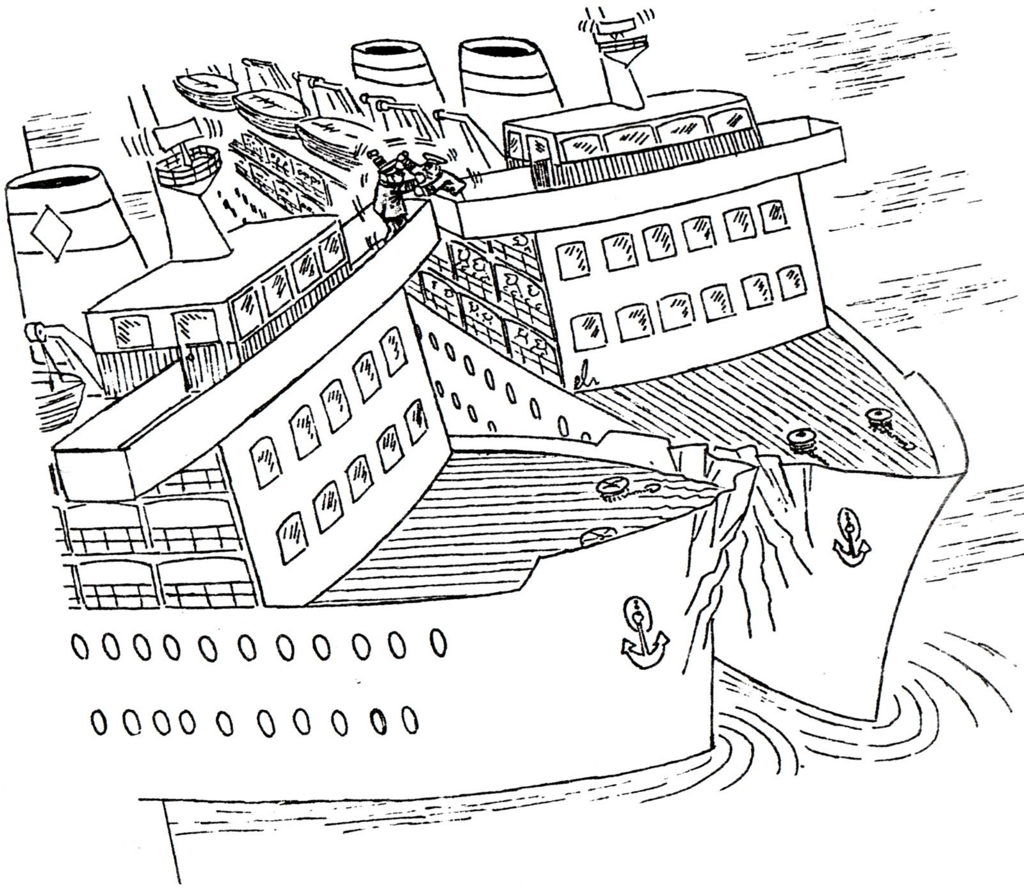
Anm. Der Cursor ist nicht sichtbar und wird intern verwaltet. Steuerzeichen werden nicht erkannt. Legal sind nur Zeichen zwischen 20H und 07FH. Alle anderen werden ignoriert. Bei Erreichen eines Zeilenendes wird ein Umbruch durchgefuehrt. Dabei wird ein Zeilenabstand von 10 Grafikzeilen angenommen.

Anm. Das Erzeugen verschiedener Fenster kann mit (16) geschehen. Die Zeiger auf die DCB's muessen gespeichert erden. Dann alle DCB's nacheinander selektieren (15) und Parameter eintragen(8). Die Ausgabe von Zeichen erfolgt nur in das gegenwaertig aktive Fenster. SetWin kann zwischen den verschiedenen Fenstern wechseln.

Ich denke, daß das Konzept recht interessant ist. Wie denkt Ihr darüber? Wer hat Interesse an einer Zusammenarbeit zwecks Standardisierung? Bitte schreibt mir Eure Meinung dazu!

Ach, übrigens, wer das System haben will, soll mir eine formatierte 80TrDSDD Monte SYSTEM Diskette zukommen lassen. Der Speicher des Model 4/4p muß allerdings 128K groß sein. Als Zugabe erhält jeder Besteller ein vollständiges ZCPR33-System. Die RAM-Disk ist bei 128K Speicher allerdings nur noch 32K groß. Wen's stört...

Euer auf Echo wartender *Rüdiger*



Unterbrechung gestattet?

Im Betriebssystem gibt es immer zeitkritische oder aus sonstigen Gründen anfällige Routinen, bei denen Interrupts gesperrt werden müssen. Das stellt ja auch kein Problem dar, ein DI an der richtigen Stelle sorgt für den ungestörten Verlauf des Programmes. Wenn dieses jedoch fertig ist, sollen dan die Interrupts wieder zugelassen werden ? Oder nicht ? Das hängt davon ab, ob *vor* dem Abschalten Interrupts zulässig waren. Aber wie kann man das herausbekommen ?
Man betrachte die Z80-Befehle

```
        LD    A,I
oder
        LD    A,R
```

Bei jedem dieser Befehle wird das Flag P/V auf den Wert von IFF2 gesetzt, man erhält also Aufschluß darüber, ob Interrupts erlaubt sind! Also : Vor dem DI ein LD A,I, das P/V-Flag in geeigneter Weise speichern und schon kann man hinter der kritischen Routine die Interrupts selektiv zulassen oder sperren!

Rüdiger

ZCPR33 -- die Vierte.

Noch einmal : Installation von ZCPR33 auf den verschiedenen Z80-Maschinen.
Anscheinend ist es doch nicht so leicht, sein CP/M System auf das wesentlich bessere ZCPR-System umzurüsten. Das liegt vermutlich an meinen etwas unverständlichen Ausführungen zu diesem Thema. Da noch keinerlei Erfolgsmeldungen bei mir eingetroffen sind, versuche ich nochmals, in Form einer Schritt-für-Schritt-Anleitung, die Installation zu erklären. Im Übrigen sind die Bugs, die ich das letzte Mal beschrieben hatte, ausgemerzt und können nun vermieden werden.

Was brauche ich ?

HARDWARE

Einen Computer mit Z80, Z180, Z280 oder HD64180 CPU.
Mindestens ein Diskettenlaufwerk (am besten 2x80T).

SOFTWARE

Ein laufendes CP/M 2.2.
Quellen des BIOS.
ZCPR33.LBR (Public Domain, bei mir erhältlich).

Im folgenden differieren die Vorgehensweisen. Mein Ansatz war Standard, deshalb wird er hier behandelt.

Einen Z80-Assembler (z. B. ZAS.COM).
Einen Text-Editor.
SYSGEN.COM
evtl. MOVCPM.COM
Einen Debugger (WADE, DSD, SID, DDT o.ä.)

Das ist alles.

1. Schritt.

Welche Features soll das ZCPR33-System bekommen? Alles kostet seinen Platz. In meinem System habe ich nur auf die Implementation des Input/Output Packages verzichtet. Der Rest nimmt etwa 3.5 KBytes in Anspruch. Jedes Byte lohnt sich! Hier ist der Auszug aus meinem BIOS, der das System beschreibt. Bei mir beginnen die Z-Buffer bei Adresse 0EA00H. Das ist natürlich unterschiedlich. Rechnet aus, wie lang die Buffer werden und verschiebt das CP/M um den entsprechenden Betrag nach unten (mit MOVCPM). Wer die Quellen zu BDOS und BIOS hat, lebt leichter, er muß lediglich neu assemblieren. TESTET DAS NEUE SYSTEM! Es muß genau so arbeiten wie zuvor, nur das die TPA kleiner ist. Alles klar ? Betrachtet den Auszug aus meinem BIOS. Z3RCP liegt hinter dem eigentlichen BIOS und vor den Disk-Tabellen. Das ist aber egal. Die Z-Buffer müssen nicht mal am Stück aneinderhängen, aber das ist von Vorteil.


```
        if ZCPR33

;        I/O package not implemented

z3rcp   equ     0ea00h          ; resident command package

;     RCP beginnt bei 0EA00H (bei mir!)

z3fcp   equ     z3rcp+2048      ; Flow Command Package
z3env   equ     z3fcp+512       ; environment descriptor
z3shstk equ     z3env+256       ; shell stack
z3mbf   equ     z3shstk+128     ; message buffers
z3fcb   equ     z3mbf+80        ; ext. fcb
z3ndrb  equ     z3fcb+48        ; named directory buffer
z3mclb  equ     z3ndrb+256      ; multiple command line buffer
z3xstk  equ     z3mclb+208      ; external stack
z3whb   equ     z3xstk+49       ; Wheel Byte
z3path  equ     z3whb+1         ; external Path
eobios  equ     z3path+48       ; end Of BIOSZ

        else

eobios  equ     $

        endif                   ; IF ZCPR33
```

Wenn das System mit weniger TPA einwandfrei läuft, kommt der 2.Schritt.

Im BIOS gibt es eine Kaltstartroutine. Das BIOS hat immer eine Sprungleiste, diese steht i.allg. am Anfang des BIOS. Der erste Sprung verzweigt in die Kaltstartroutine. Diese muß modifiziert werden. Sie erhält die zusätzliche Aufgabe, die Z-Buffer zu initialisieren.

Hier ein Auszug aus meiner Kaltstartroutine :

```
        if      ZCPR33
z3init

; Alle ZCPR-Buffer mit 00H füllen

        call    clrmem
        DW      z3rcp           ; clear ZCPR3 buffers
        DW      ((z3path+48)-z3rcp) (**)

;    Kaltstartkommando -> externen Kommandopuffer

        ld      hl,intcom
        ld      de,z3mclb
        ld      bc,cmdlsz
        ldir                    ; Initialize for Cold Boot Command

;    externen Path initialisieren

        ld      hl,path
        ld      de,z3path
        ld      bc,4            ; Initialize path
        ldir
```

Wheel-byte auf Supervisor setzen

```
        ld      a,0ffh
        ld      (z3whb),a       ; Initialize Wheel to non-protect
        ret
```

Das ist der Initial-Suchpfad,
A0: -> A15:

```
path    defb    'A'-'§',0,'A'-'§',15

intcom  defw    z3mclb+4                ; nxtchr
        defb    208                     ; bufsiz
        defb    zcmdend                 ; chrcnt
```

Dieser Kommandostring wird beim KALTSTART ausgeführt :

```
zcmdstr defb    'LDR SYS.ENV;STARTUP'
zcmdend equ     $-zcmdstr
cmdlsz  equ     4+zcmdend

        endif
```

Das ist genau der Code, der notwendig ist, um die Z-Buffer zu initialisieren. Mehr braucht es nicht. Und damit ist das Einbinden von ZCPR33, was das BIOS angeht, beendet. Ändert die Kaltstartroutine und TESTET DAS SYSTEM!

3. Schritt.

Notiert alle Adressen der Z-Buffer (RCP, FCP etc.). Ferner die BDOS-Startadresse (steht an Stelle 06H im Speicher) und die Adresse des CCP (BDOS-806H). Kopiert ZCPR33.LBR auf eine leere Diskette. Packt die LBR aus (mit NULU oder sowas) und löscht die LBR von der Diskette. Die Dateien sind gecrunched oder gesqueezed und müssen entsprechend expandiert werden. Löscht dann alle .?Q? und .?Z? Dateien von der Diskette.
Editiert die Datei Z3BASE.LIB. In sie müssen alle Adressen eingetragen werden, die Ihr zuvor notiert habt. Editiert die Datei Z33HDR.LIB und wählt die gewünschten Features des Systems aus. Versucht es mit JUMP, GO und SAVE. Assembliert ZCPR33 in ein HEX-File.

A>ZAS ZCPR33 H

4. Schritt.

Als nächstes wird der alte CCP mit ZCPR33 überschrieben und auf den Systemspuren abgelegt. Dazu wird SYSGEN und ein Debugger benötigt. Das System muß von der Diskette mit dem geänderten BIOS gelesen werden (hier B:)

```
A>SYSGEN
  ...
     PRESS A..P TO READ SYSTEM TRACKS OR <ENTER>:         B
     INSERT DISK IN B: AND PRESS <ENTER>          :       <ENTER>
     PRESS A..P TO WRITE SYSTEM TRACKS OR ENTER   :       <ENTER>

     SYSGEN COMPLETED.

A>SAVE 44 CPMREV.BIN
```


Die Zahl 44 ist von System zu System unterschiedlich, sie gibt die Größe in Pages (256 Bytes) an. SYSGEN sollte die Information rausrücken. Jetzt gibt es eine Datei CPMREV.BIN, die das System enthält. Wo war doch gleich die Adresse des CCP? sagen wir mal, CC00H. Nun muß der Debugger gestartet werden.

A>DDT CPMREV.BIN

Damit wird gleich das System geladen. Wie gesagt, es enthält auch den CCP. Er sollte so um die Adresse 800H im Systemabbild liegen, sucht ihn mit dem Debugger. Am Anfang des CCP steht eine Copyrightnotiz von Digital Research. Sucht um 800H, eher weniger. Sagen wir, Ihr findet den CCP bei 800H.

H800,CC00	**berechnet den Offset für das Laden von ZCPR33**
D400 3C00	**Summe und Differenz von 800H und CC00H**
IZCPR33.HEX	**Initialisiert einen FCB**
R3C00	**lädt ZCPR33.HEX mit einem Offset von 3C00. Prüfen!**
^C	
A>SAVE 44 CPM.BIN	**sichert das neue System.**
A>SYSGEN CPM.BIN usw.	**-> System auf die Spuren schreiben.**

(Anm. Diese Vorgehensweise gilt sinngemäß genauso für Änderungen des BIOS und BDOS. Allerdings eben nur, wenn in ein HEX-File assembliert wird. Es gibt Assembler, die die direkte Erzeugung eines COM-Files unterstützen.)

Uff! Bootet das System. Es sollte sagen :

A0>LDR SYS.ENV;STARTUP?

Dann ist alles in Ordnung.

5. Schritt.

Nun müssen die System-Segmente vorbereitet werden. Dies sind zunächst SYS.ENV, SYS.RCP, SYS.FCP, SYS.NDR und SYS.Z3T. Letztere kann erstmal weggelassen werden. Editiert SYSENV.LIB und SYSENV.ASM und erzeugt daraus ein HEX File.

```
A>MLOAD SYSENV
     Warning! program origin not at 100H!
A>REN SYS.ENV=SYSENV.COM
```

Macht dasselbe mit ZCPR33xx.RCP und .FCP, so daß Ihr die Files SYS.RCP und SYS.FCP erhaltet. Legt diese Dateien in A15: der ZCPR33 Boot-Diskette ab.

Letzter und sechster Schritt.

Es müssen nur noch die Utilities installiert werden, besonders VALIAS.COM und LDR.COM. Erzeugt eine Datei, die, Zeile für Zeile, die Namen (*.COM) aller ZCPR-Utilities enthält. Bringt SYS.ENV, Z3INS.COM und die Liste, die SYS.INS heißen sollte, nach A0:.
A0>Z3INS SYS.ENV SYS.INS
Das dauert ein paar Minuten, dann sind alle Utilities installiert und können benutzt werden. Bootet das System! LDR sollte SYS.ENV laden, über STARTUP? darf er noch meckern. Legt einen STARTUP.COM an mit
A>VALIAS STARTUP
(vorausgesetzt, VALIAS ist installiert)

Er sollte folgende Befehle enthalten :

LDR SYS.RCP,SYS.FCP,SYS.Z3T,SYS.NDR
PATH $$$$: A0: A15:

Legt STARTUP.COM in A0: oder A15: ab.
Das System sollte jetzt booten, außer .Z3T und .NDR alle Segmente laden und eine Suchpfad einrichten über
current drive/user
drive a user 0
drive a user 15

Nun müßt Ihr nur noch mit MKDIR.COM eine Datei SYS.NDR einrichten, wenn Ihr Directories mit Namen haben wollt. Falls SYS.ENV richtig ist, könnt Ihr vorerst auf SYS.Z3T verzichten.

Falls Ihr ZFILER.COM oder ZF.COM habt, startet es, sonst vielleicht VFILER.COM. Es wird sich über fehlende Terminal-Installation schon beschweren. Wenn alles aber funktioniert, dann

*(** Aus drucktechnischen Gründen (!) habe ich die eckigen Klammern, die ZAS für seine Arithmetik verlangt, durch runde Klammern ersetzt. Das müßt Ihr ändern.)*

HERZLICHEN GLÜCKWUNSCH!!!

Rüdiger

Prommer 80 unter G-DOS

Ja, Ihr habt richtig gelesen. Ab jetzt müssen diejenigen von Euch, denen es nicht gegönnt ist, CP/M zu 'fahren', aus welchen Gründen auch immer, nicht mehr auf das EPROM-Brennen verzichten. Es geht auch unter G-Dos bzw. Newdos.

Der Grund dafür ist meine Faulheit und Unerfahrenheit. Da ich noch keine nennenswerten Erfahrungen mit CP/M habe programmiere ich noch unter G-Dos mit ZEUS. Da ich in der letzten Zeit mein Boot-Eprom modifiziert habe, mußte ich den Inhalt immer mit CPMAC auf CP/M-Disketten kopieren, dann CP/M booten, Prommer anwerfen, programmieren and so on. Diese Prozedur ging mir mit der Zeit an meine Nerfen. Also mußte ein Programm her welches unter G-Dos lief. Nein, ich habe keines geschrieben, dazu war ich zu faul (s.o.), außerdem gibt es ja ein sehr gutes, das von Becker. Nur das läuft unter CP/M.

Beim durchstöbern meiner Erinnerung sowie einiger CP/M Bücher fiel mir auf, daß CP/M ein Rechnerunabhäniges Betriebssystem ist und eigentlich alles was mit der Hardware zu tun hat über eine Sofwareschnittstelle erledigt. Als kleinen Exkurs in CP/M für diejenigen die von CP/M noch weniger wissen als ich, hier eine Erläuterung wie CP/M das macht.

Alle Ein- und Ausgaben die den Bildschirm oder die Diskette betreffen werden über das BDOS abgewickelt. BDOS steht für ' Basic Disk Operation System '. Das BDOS wird angesprungen über die Adresse 0005H. In Register C wird die Funktionsnummer übergeben. Weitere Parameter in anderen Registern. (siehe auch Lising) Die Befehlsfolge

```
LD   E,41H
LD   C,02H
CALL 0005h
```

zum Beispiel gibt den Buchstaben 'A' auf dem Bildschirm aus. Genauso macht es auch das Programm von Becker genannt PROM. Soweit zu CP/M.

Ich habe nun das Programm mit CPMAC nach G-Dos kopiert, disassembliert und dann alle Call-Befehle nach 0005H in 'bdos' umbenannt. Jetzt habe ich eine Routine geschrieben die das macht was BDOS im CP/M auch macht. Siehe Listing. Die übergebene Funktionsnummer wird abgefragt und dann zur entsprechenden Routine verzweigt. Dort werden die zusätzlichen Parameter in die, die G-Dos braucht, umgewandelt und dann die entsprechende G-Dos-Funktion aufgerufen. Der Sprung von 'bdos' nach 'system' ist nötig, weil das Hauptprogramm diese Adresse " umbiegt " um die Register zu retten. Im Hauptprogramm mußte fast nichts geändert werden, bis auf das Trennzeichen bei Filenamen. Dieses ist bei CP/M ein Punkt "." und bei G-Dos ein "/". Auch scheinen die Funktionen 'read' und 'write' sehr ungewöhnlich. Zur Erklärung: CP/M benutzt 128-Byte Sektoren und G-Dos 256-Byte. Um einer komplizierten Umrechnung zu entgehen, habe ich einfach Byteweise, und zwar immer 128 an der Zahl, übergeben, das war der einfachste Weg. Ich stimme Euch zu, wenn Ihr der Meinung seid das ginge auch einfacher. (besonders Arnulf) aber mir kam

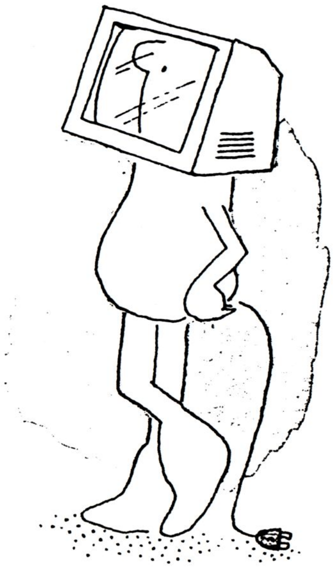
es nur auf die Funktion an und wenn Ihr nach der Geschwindigkeit fragt, so ist G-Dos sowieso viel schneller.

Das abgedruckte Listing zeigt natürlich nicht das ganze Programm, dafür müßte ein Extra-Info erscheinen, sondern nur die Bearbeitung der gebrauchten Funktionen. Es ist nur auf das eine Programm abgestimmt, aber wer Lust hat kann es ja für andere Dinge "mißbrauchen". Da wären wir auch gleich beim Urheberrecht, das bleibt natürlich beim Verfasser, und die kommerzielle Weitergabe ist nicht erlaubt. Anders ist es bei meinem Zusazprogramm, das darf, und soll geändert, erweitert und weitergegeben werden, aber auch nur kostenlos.

Jetzt zu Eurer nächsten Frage, wie Ihr an das Programm rankommt. Bitte schreibt nicht mir wenn Ihr es haben wollt, ich habe das Programm sowie die Scorces dazu an Oliver unseren G-Dos/Newdos-Disk-Jokey geschickt, bitte wendet euch an Ihn. Das lauffähige Programm heist 'PROMGDOS'. Als weiteres gibt es ein Programm 'PROMINST', dieses dient dazu das Hauptprogramm an Euern Rechner und Prommer anzupassen. Wenn Ihr allerdings Fragen zum Programm selbst, oder Beschwerden habt, dann wendet Euch an mich. Apropos Beschwerden, ich übernehme natürlich keine Garantie damit, aber das kennt Ihr ja.

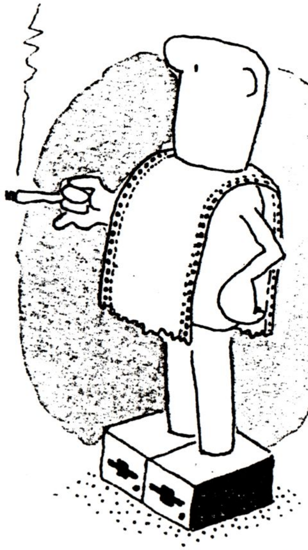
Zu Erwähnen wäre noch, daß PROMINST den Namen mit Extension haben will, sonst stürzt es ab. Als zweites, das Format der Dateien die PROMGDOS generiert, oder haben will ist immer noch CP/M. Das heißt, Ihr müßt die Dateien von Ladecodes befreien, oder diese hinzufügen, das geschieht mit den Programmen CMDER/BAS und COMER/BAS. Die sind auch mit auf der DISK die Oliver hat.

Tschüss



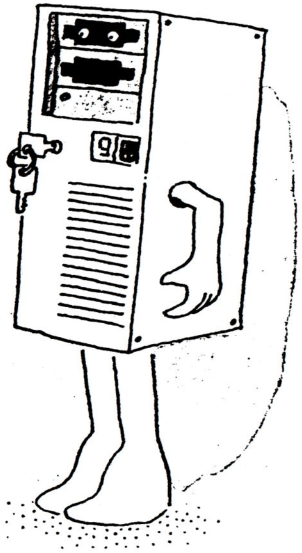

```
;          BDOS/SRC
;*************************************************
;
; Diese Routine simuliert die CP/M-BDOS-Funktionen,
; die für das Programm PROM von Becker gebraucht
; werden, unter G-DOS.
;
; Änderungen, Erweiterungen sowie Unkommerzielle
; Unkommerzielle Weitergabe sind ausdrücklich erwünscht.
;
; (c) Mai 1989 by Andreas Magnus HACKNUS-SOFTWARE
;
;*************************************************
         ORG     5200h
         DI      .
         JP      m0100           ;Start des CP/M-Programmes
.

;SYSTEM-Routinen zur Anpassung von PROM/CMD an G-DOS.

bdos     JP      system          ;SYSTEM ist der Einsprung
in das
                                 ;Beriebssystem

system   LD      A,C             ;Parameter in den ACCU
         CP      01
         JP      Z,getkey        ;warte bis Taste gedrückt
         CP      02
         JP      Z,out           ;Zeichen an Bildschirm
         CP      09
         JP      Z,display       ;Zeichenkette an den
                                 ;Bildschirm
         CP      10
         JP      Z,zeile         ;Zeile von Tastatur lesen
         CP      11
         JP      Z,inkey         ;Tastatur abfragen
         CP      13
         JP      Z,sysres        ;SYSTEM zurücksetzen
         CP      14
         JP      Z,select        ;Laufwerk anwählen
         CP      15
         JP      Z,open          ;Datei öffnen
         CP      16
         JP      Z,close         ;Datei schließen
         CP      20
         JP      Z,read          ;sequentiell lesen
         CP      21
         JP      Z,write         ;sequentiell schreiben
         CP      22
         JP      Z,init          ;neue Datei anlegen
         CP      25
         JP      Z,getdrv        ;Aktuelles Laufwerk abfrag
en
         CP      26
         JP      Z,setdma        ;Sectorpuffer setzen
         RET                     ;Fehler
getkey   PUSH    DE              ;DE wird noch gebraucht
         CALL    0049h           ;auf Zeichen warten
         CALL    0033h           ;Zeichen ausgeben
         POP     DE              ;wieder zurück
         RET                     ;das wars

out      LD      A,E             ;Zeichen in den Accu
out1     CP      0ah             ;ist es LF ?
         RET     Z               ;wird nicht ausgegeben
         CP      08h             ;Backspace ?
         JR      NZ,normal
         LD      A,24            ;24 daraus machen (Rückschritt)
```

```
                                    ;ohne löschen)
normal   PUSH    DE
         CALL    0033h              ;Zeichen ausgeben
         POP     DE
         RET

display  LD      A,(DE)             ;Zeichen in den ACCU
         CP      '$'                ;Ende der Zeichenkette ?
         RET     Z                  ;fertig
         CALL    out1               ;Zeichen ausgeben
         INC     DE                 ;nächstes Zeichen
         JR      display            ;bis alle fertig

zeile    PUSH    DE
         PUSH    HL
         PUSH    BC
         EX      DE,HL              ;Pufferanfang nach HL
         INC     HL
         INC     HL                 ;hier steht das erste Zeichen
         LD      B,80               ;maximale Zahl der Zeichen
         CALL    05d9h              ;B-Zeichen einlesen
         DEC     HL                 ;BUFFER + 1
         LD      (HL),B             ;Eingegebene Zeichen
         DEC     HL                 ;Buffer + 0
         LD      (HL),C             ;Max. Zahl der Zeichen
         POP     BC
         POP     HL
         POP     DE
         RET                        ;fertig

inkey    PUSH    DE
         CALL    002bh              ;Tastatur abfragen
         POP     DE
         RET

sysres   RET                        ;wird bei G-DOS nicht gebraucht

select   RET                        ; dito

open     CALL    umw                ;FCB von CP/M nach G-DOS wandeln
         CALL    4424h              ;OPEN FILE
rück     RET     Z                  ;alles klar
         CP      18h                ;Datei nicht gefunden ?
         RET     NZ                 ;anderer FEhler
fehler   LD      A,0ffh
         RET

close    LD      DE,fcb             ;FCB des zu schließenden Files
         CALL    4428h              ;Dos-Call Datei schließen
         JR      rück               ;weiter wie nach open

read     PUSH    DE                 ;FCB retten
         LD      DE,fcb             ;FCB des geöffneten Files
         LD      HL,(dma)           ;HL = Adresse des Buffers
         LD      B,128              ;128 Byte lesen
readlop  CALL    0013h              ;nächstes Byte lesen
         JR      NZ,err             ;wenn Fehler aufgetreten
         LD      (HL),A             ;und in den Puffer
         INC     HL
         DJNZ    readlop            ;bis alle Bytes gelesen
normaus  XOR     A                  ;alles OK
err      POP     DE
         RET

write    PUSH    DE                 ;FCB retten
```



```
         LD      DE,fcb             ;FCB des geöffneten Files
         LD      HL,(dma)           ;HL= Adresse des Buffers
         LD      B,128              ;128 Byte schreiben
writelo  LD      A,(HL)             ;Byte aus dem Puffer
         CALL    001bh              ;nächstes Byte schreiben
         JR      NZ,err             ;wenn Fehler aufgetreten
         INC     HL                 ;nächste Pufferstelle
         DJNZ    writelo            ;bis alle Byte's geschrieben
         JR      normaus            ;normaler Rücksprung

init     CALL    umw                ;s.o.
         CALL    4420h              ;INIT FILE
         LD      HL,m0080           ;Sectorpuffer für Ein-Ausgabe
         LD      (dma),HL           ;als Sectorpuffer abspeichern
         RET     Z                  ;alles klar
         CP      1ah                ;Inhaltsverzeichnis voll
         JR      Z,fehler           ;dann Fehler
         CP      1bh                ;Disk voll ?
         JR      Z,fehler           ;dann auch
         RET                        ;anderer Fehler

getdrv   RET

setdma   LD      (dma),DE           ;DE in das DMA-Flag
         RET                        ;das wars


umw      INC     DE                 ;DE zeigt auf FILENAMEN
         LD      HL,filenam         ;Zwischenspeicher für Filename
         LD      B,8                ;8 Zeichen übertragen
fnamelo  LD      A,(DE)             ;Zeichen aus CP/M Filename
         CP      20h                ;ist der Filename zuende ?
         JR      Z,ext              ;dann bei der Extension weitermachen
         LD      (HL),A             ;in G-Dos Filename übertragen
```

```
         INC    HL
         INC    DE              ;beide ein höher
         DJNZ   fnamelo         ;bis 8 Zeichen übertragen
ext      CP     ' '             ;war Fname zuende
         JR     NZ,ext1         ;nein
exloop   INC    DE              ;DE + 1
         DJNZ   exloop          ;DE auf EXT stellen
ext1     LD     (HL),'/'        ;Trennungsstrich
         INC    HL              ;hier steht die EXT
         LD     B,3             ;3 Byte für EXT
extloop  LD     A,(DE)          ;Zeichen in Accu
         CP     ' '
         JR     Z,ok            ;EXT zuende
         LD     (HL),A
         INC    HL
         INC    DE
         DJNZ   extloop         ;bis 3 Byte übertragen
ok       LD     (HL),0dh        ;Endekennung für G-Dos
         LD     HL,filenam
         LD     DE,fcb          ;FCB für Files
         CALL   441ch           ;Filename in FCB übertrage
n
         LD     B,0             ;Recordlänge
         LD     HL,buffer       ;Buffer für den File
         RET                    ;zurück

filenam  DM     'FILENAME/EXT',0dh      ;Platz für den FIL
ENAMEN
fcb      DS     32                      ;Platz für den FCB
dma      DW     0000                    ;Platz für Buffera
dresse
buffer   DS     256                     ;Platz für die Dat
en

         END    5200h
```



Fensterln mit CP/M

Erstes Ergebnis des Modula-Projekts

Wer hat sich nicht schon immer gewünscht, diese niedlichen kleinen Fensterchen mit dem Rahmen drumherum auf seinem CP/M-Rechner zu haben? Ich weiß: Keiner. Aber jeder poplige PC kann das auch; und den benutzt Ihr ja auch jeden Tag im Büro! Also habe ich mich an die Arbeit gemacht und gemäß diversen Vorbildern (Chip-Special-Turbo-Pascal, GEM) ein kleines Modul erstellt, das genau solche Fensterchen ermöglicht. Allerdings nur mit Texten und nicht mit Grafik.

Sinn und Zweck: Beispiele

Wozu sind Fenster eigentlich zu gebrauchen? Nun, eigentlich steht uns nur der Bildschirm als "großes" Fenster zur Verfügung. Das reicht auch vollkommen, solange wir nur einen Text editieren wollen, nur ein Menü sehen usw. Die Betonung liegt auf "ein". Windows kommen ins Spiel, sobald wir uns von dem "ein" lösen und "zwei und mehr" (Texte, Menüs usw.) auf unserer Wunschliste stehen. Dazu zwei Beispiele:

Beispiel 1: Wir möchten zwei Texte gleichzeitig editieren. Wie ordnen wir die beiden Text-Ausschnitte auf dem Bildschirm an, so daß sie sich nicht stören? Und wie verhindern wir, daß eine Änderung in einem Text (z.B. das Löschen einer Zeile) sich auf den anderen Text auswirkt? Die Lösung: Jeder Text-Ausschnitt wird in einem Window (mit Rahmen drumrum) ausgegeben. Dabei dürfen sich die beiden Windows ruhig überlappen; wir können die Teile in beliebiger Größe anzeigen und an einer beliebigen Stelle auf dem Bildschirm plazieren. Die Änderungen in einem Text wirken sich nur auf sein Window aus. Das andere Window wird davon überhaupt nicht berührt.

Beispiel 2: Wir haben ein Menüsystem mit einem Hauptmenü; zu jedem Menüpunkt im Hauptmenü gibt es evtl. wieder ein Untermenü usw. Wenn wir nur einen Bildschirm zur Verfügung haben, müssen wir erst das Hauptmenü anzeigen, dann den Bildschirm löschen und das Untermenü anzeigen (das Ganze evtl. mehrere Stufen tief). Bei "Hochsteigen" aus den Menüs muß dann wieder jedes Mal der Bildschirm gelöscht und das Menü angezeigt werden usw. Unter Windows sieht die Lösung anders aus: Am Anfang des Programms öffnen wir für jedes Menü ein Fenster und schreiben den Text dort hinein; der Text erscheint aber noch nicht auf dem Bildschirm. Dann geben wir die Anweisung: "zeige das Fenster *Hauptmenü*" und, schwupp, erscheint es auf dem Bildschirm. Bei jedem Untermenü geht es genauso. Dabei lassen wir das Hauptmenü-Fenster (mit seinen folgenden Untermenü-Fenstern) auf dem Bildschirm stehen. Wenn wir uns jetzt wieder "hochhangeln" geben wir nur die Anweisung "lasse das Fenster *Untermenü x* vom Bildschirm verschwinden" und schon taucht das darunterliegende Fenster mit dem nächsthöheren Menü wieder auf. Das könnte dann so aussehen:

```
*--- Hauptmenü ---*
I  a. Punkt 1     I
I  *--- Untermenü 1 ---*
I  I   x. Punkt 1.1    I
I  I   y. Punkt 1.2    I
I  I  *--- Untermenü 1.2 ---*
I  I  I   1. Punkt 1.2.1    I
I  I  I   2. Punkt 1.2.2    I
*--I  I   3. Punkt 1.2.3    I
   I  I   4. Punkt 1.2.4    I
   *--I   5. Punkt 1.2.5    I
      I   6. zurück zu 1.   I
      *---------------------*
```

Bedienung und Funktionsumfang: Programmbeispiel

Soweit zu den Beispielen. Natürlich brauchen sich Fenster nicht zu überlappen. Sie können auch neben- oder untereinander stehen. Dann dienen sie nur dazu, den Bildschirm übersichtlich aufzuteilen. Beispiel:

```
*------ Menü ------* *------ Zustand ------*
I ...              I I ....                I
I                  I I                     I
...                 ...                  ...
I                  I I                     I
*------------------* *---------------------*
*------------- Ausgabe --------------------*
I ...                                      I
I                                          I
*------------------------------------------*
```

Das läßt sich natürlich auch ohne das Modul "Windows" so oder ähnlich erledigen. Aber warum sollten wir für jedes Programm das Rad neu erfinden? Jetzt müssen wir nur noch sagen: "Öffne ein Fenster an der Position (7,4). Es soll 5 Zeilen lang und 40 Zeichen breit sein. Als Titel soll über dem Fenster "A:Text.DOC" stehen." Das liest sich in Modula so:

```
...
VAR TextWindow: WindowHandle;
...
TextWindow := WdOpen (7,4,40,5);
WdSetTitle (TextWindow, "A:Text.DOC");
...
```

Wir können jetzt Text in dem Fenster ausgeben; dieser Text erscheint aber noch nicht auf dem Bildschirm, sondern wird intern gespeichert:

```
WdWriteC   (TextWindow, "X");          (* ein Zeichen *)
WdGotoXY   (TextWindow, 0,1);          (* Cursor in die nächste Zeile *)
WdWriteStr (TextWindow, "Ein String"); (* ein String *)
WdWriteC   (TextWindow, EoLn);         (* Cursor in die nächste Zeile *)
```

Wie Ihr seht, müssen wir immer den Namen des Windows ("TextWindow") mit angeben, wenn wir Text ausgeben oder den Cursor positionieren usw. ~~aber~~ Nur so kann das Modul wissen, auf welches Window (von den beliebig vielen Windows) wir uns beziehen.

Wenn wir jetzt das Fenster anzeigen lassen, erscheint es an der richtigen Position mit dem richtigen Inhalt:

```
WdTop (TextWindow);    (* es erscheint folgendes: *)
```

```
*------------ A:Text.DOC --------------*
IX                                     I
IEin String                            I
I_                                     I
I                                      I
I                                      I
*--------------------------------------*
```

Evtl. liegen noch vorher geöffnete Windows darunter, daneben usw. Der "_" (Unterstrich) soll hier die Cursor-Position markieren; bei Eurem Rechner blinkt da vielleicht ein Block o.ä. Wie Ihr seht, ist die Innenfläche 40 Zeichen breit und 5 Zeilen hoch. Das ist nun unser Mini-Bildschirm für "A:Text.DOC".

Wenn nun ein anderes Window "nach vorne geholt" wird, verdeckt es evtl. teilweise dieses Window. Auf jeden Fall blinkt der Cursor dann in dem anderen Window. Das heißt aber nicht, daß wir nicht in "TextWindow" noch Text ausgeben könnten; der Cursor ist im "TextWindow" zwar nicht mehr sichtbar, aber er ist noch vorhanden, und nach

```
WdWriteStr (TextWindow, "Noch ein String");
```

würde diese Zeile an der Position angezeigt werden, wo der Cursor vorher stand. Diese Zeile wird auch auf dem Bildschirm angezeigt, wenn der Teil nicht von dem neuen obersten Window verdeckt wird.

Wenn wir das Window nicht mehr auf dem Bildschirm sehen wollen, geben wir

```
WdHide (TextWindow);
```

und schon verschwindet es. Wir können es aber jederzeit mit "WdTop" in alter Form und mit dem alten Inhalt wieder anzeigen lassen. Erst nach

```
WdClose (TextWindow);
```

gibt es unser "TextWindow" gar nicht mehr. Der ganze Inhalt ist jetzt auch verloren.

Wir hätten das Window auch zwischendurch mal woanders plazieren ("WdSetPos") oder seine Größe verändern ("WdSetSize") können. Der Inhalt wäre dabei erhalten geblieben (soweit möglich, wenn das Window kleiner wird).

Außerdem gibt es noch einige Prozeduren zur Text-Manipulation: "WdInsLine" fügt an der Cursorposition im Fenster (und nur dort!) eine Leerzeile ein, "WdDelLine" löscht eine Zeile. "WdScrollUp" bzw. "WdScrollDown" scrollen den ganzen Fensterinhalt eine Zeile hoch bzw. runter. "WdWriteStrLn" schreibt Strings (wie "WdWriteStr"), aber der Text darf länger als eine Zeile im Window sein; der Rest wird dann in die nächste(n) Zeile(n) geschrieben.

Außerdem unterstützt Windows auch noch einige Text-Attribute, wenn Euer Rechner das kann. Bei meinem Televideo-Terminal arbeite ich jetzt mit "HighLight" (Buchstaben leuchten heller), "Revers" (bzw. invers), "UnderLine" (unterstrichen) und "Flash" (die Zeichen blinken). Das kann alles ein-/ausgeschaltet werden, indem das Zeichen "On[Attribut]" bzw. "Off[Attribut]" ausgegeben wird. Entweder erledigt man das per "WdWriteC" oder Ihr baut die Zeichen in den auszugebenden String ein:

```
WdWriteC (TextWindow, On[Flash]);
WdWriteStr (TextWindow, "Tatü-tata, die Post ist da!");
WdWriteC (TextWindow, Off[Flash]);
```

und schon steht der Alptraum jedes echten Hackers blinkenderweise im "TextWindow", das hoffentlich gerade sichtbar ist. Ansonsten ist das selbstgebaute Modem futsch.

Das Windows-Paket (mit Dokumentation, Hilfs- und Test-Modulen) könnt Ihr bei Rüdiger (CP/M-Diskothek) bekommen. Die neuesten Infos gibt's natürlich bei mir, aber ich werde Rüdiger immer auf dem laufenden halten. Für die Italiener bietet sich noch Alexander als Bezugsadresse an. Bei der (kinderleichten) Anpassung helfe ich gerne; die Größe des zusätzlichen Programmcodes kann von 3 bis 14 Kb reichen (je nach gewünschter Geschwindigkeit).

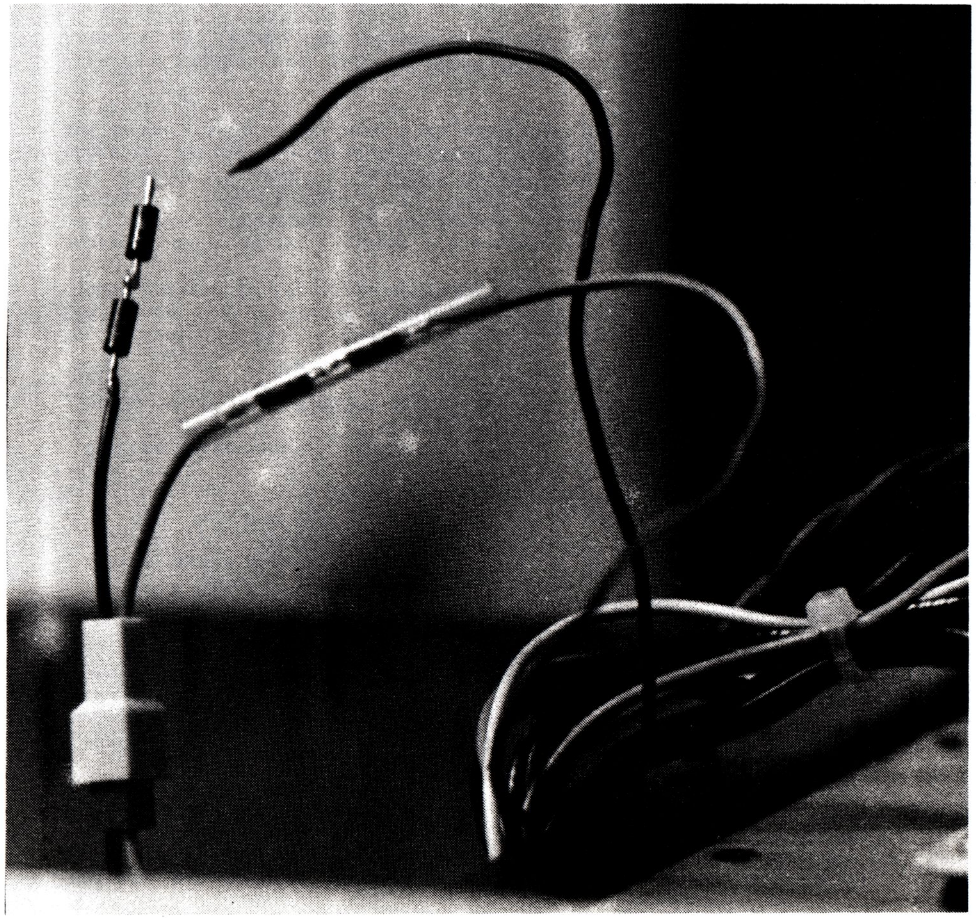
Gerald

Der flüsternde Portable

****

Dank Hartmut Obermanns ausführlicher Anleitung im Info Nr. 26 war der Einbau spurumschaltbarer Diskettenlaufwerke in meinen tragbaren 4p-Computer kein Problem. Die Mitsubishi-Laufwerke haben ihre Befestigungslöcher an anderer Stelle als die original montierten Tandon-Laufwerke, so daß man den Bohrer ansetzen muß. Wenn aber die mechanische Arbeit getan ist, fällt einem schon beim ersten Probelauf das dumpf schnurrende Positioniergeräusch als angenehmer Gegensatz zum lautstarken "Sägen" der Tandon-Drives auf.

Ein Blick ins Datenblatt der Mitsubishi-Laufwerke zeigt, daß sie weniger Strom brauchen als ihre Vorgänger. Dieser Vorteil erlaubt, den 4p-Computer mit weniger Luftdurchsatz und damit noch leiser zu betreiben. Den radikalsten Schritt, den Lüfter ganz abklemmen, wollte ich nicht tun. Nach halbstündigen Hin- und Herkopieren konnte man nämlich bei ausgeschaltetem Gebläse eine deutliche Erwärmung der rechten Gehäusehälfte oberhalb der Diskettenlaufwerke feststellen. Aber zwischen voller Leistung und Null gibt es ja noch einen Mittelweg. Versuche ergaben, daß schon eine geringe Verminderung der Spannung den Lüfter viel langsamer laufen läßt. Mit etwa 2,5 Volt weniger Betriebsspannung wird das singende Lüftergeräusch deutlich gedämpft und die Umwälzung staubhaltiger Luft verringert.



Die einfachste Art, Gleichspannungen zu "vernichten", ist entweder das Vorschalten einer Zenerdiode, oder bei kleinen Spannungen das Vorschalten normaler Dioden in Flußrichtung. Wie das Foto zeigt, habe ich das Zuleitungskabel zum Lüfter aufgetrennt und in beide Adern je zwei Dioden 1N4003 eingelötet. Beim roten (hellen) Kabel ist die Verbindung fertig. Der durchsichtige Schrumpfschlauch wurde schon übergezogen, aber noch nicht erhitzt. Am schwarzen Kabel kann man sehen, wie die Diodenkette verlötet wird.

Heinrich Betz

Umlaute auf der Tastatur des Genie I

Bekanntlich hat die Tastatur keine Umlaute und kein ß. Wenn man jedoch das Gehäuse öffnet, findet man über der Zahlenreihe noch 10 freie Tastenplätze, von denen 8 angeschlossen sind.
Von links nach rechts:

1. nicht angeschlossen
2. Abpfeil
3. Ä oder Hochpfeil
4. ö mit Shift ö
5. ü mit Shift ü
6. - mit Shift ß
7. Chr 95 mit Shift Chr 127
8. Linkspfeil
9. Rechtspfeil
10. nicht angeschlossen

Sicher ist das bekannt. Blöd ist nur, daß die Umlaute in Großschrift und mit SHIFT in Kleinschrift kommen. Außerdem kommt ä überhaupt nicht. Deshalb schrieb ich einen Treiber, der die Funktion dieser Tasten umdreht und das ä erzeugt.

Kaum war ich damit zufrieden, stellte sich mir die Frage: Was soll ich mit den anderen 3 Pfeiltasten? Die Chr 95 und 127 waren auch recht nutzlos. Das Studium der Tastatur-Matrix ergab, daß die Adresse 3880H nur mit der Shift-Taste belegt war und somit noch eine Abfragemöglichkeit für 7 weitere Tasten zur Verfügung stand. So erweiterte ich meinen Keyboard-Treiber um die Abfrage dieser Adresse. Was ich mit den Tasten gemacht habe, weiß ich heute nicht mehr, denn kurz danach bekam ich eine Tastatur mit deutscher Tastenbelegung und 23 Funktionstasten, mehr als die Adresse 3880H hergab. Aber wozu gibt es Dioden? Ich holte also den Lötkolben hervor und heraus kam der Plan, der hoffentlich am Schluß dieses Artikels abgedruckt ist. Jetzt habe ich mehr Abfragemöglichkeiten als Tasten, denn der Plan ist noch längst nicht ausgereizt.

Wer den Plan studiert, soll sich nicht an den Namen der zusätzlichen Tasten stören. Meine Tasten sind eben so beschriftet.

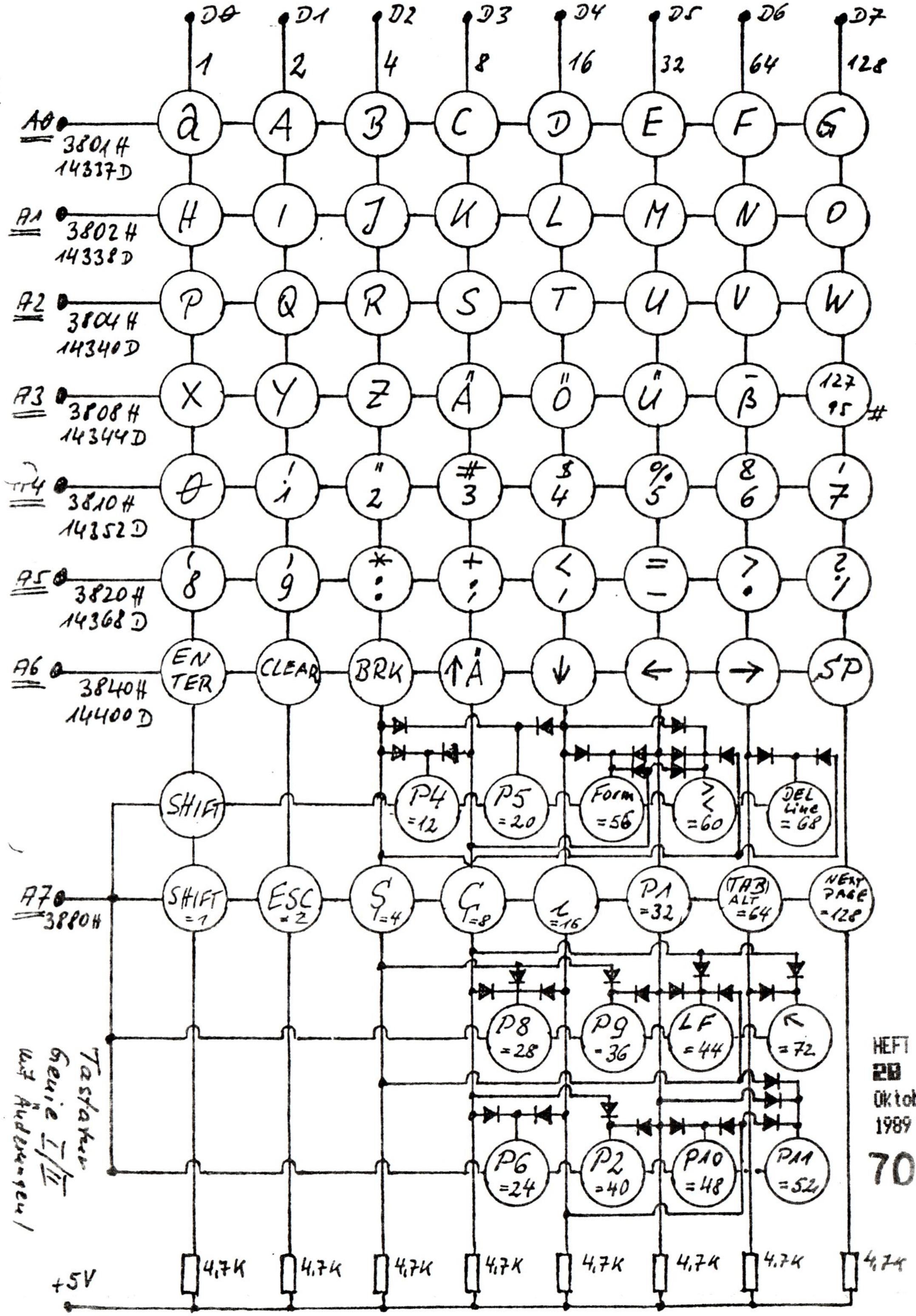
Die Abfrageart der Sondertasten ergibt 4-fache Tastenbelegung:

SHIFT addiert 1 zum abgefragten Tasten-Code.
ESC addiert 2 zum abgefragten Tasten-Code.
Das ist natürlich kein echtes Escape, die Taste ist so beschriftet.

Als Beispiel die Taste mit dem komischen S in meinem Plan (das ist wie die nächsten 2 ein türkisches Zeichen):
Tastendruck = 4
SHIFT+Taste = 5
ESC +Taste = 6
SHIFT+ESC+Taste=7

Belegt habe ich die Sondertasten bisher mit 3 türkischen Zeichen, Open, Close und CLS für die HRG-1B und einigen Druckersteuerungen wie Initialisierung, TOF, LF, FF, Elite, Pica, double strike, Breitschrift, linker Rand 4 und 6. Mit der TAB-Taste (inwischen umbenannt in ALT) und einer 'normalen' Taste werden die Grafikzeichen ausgegeben.

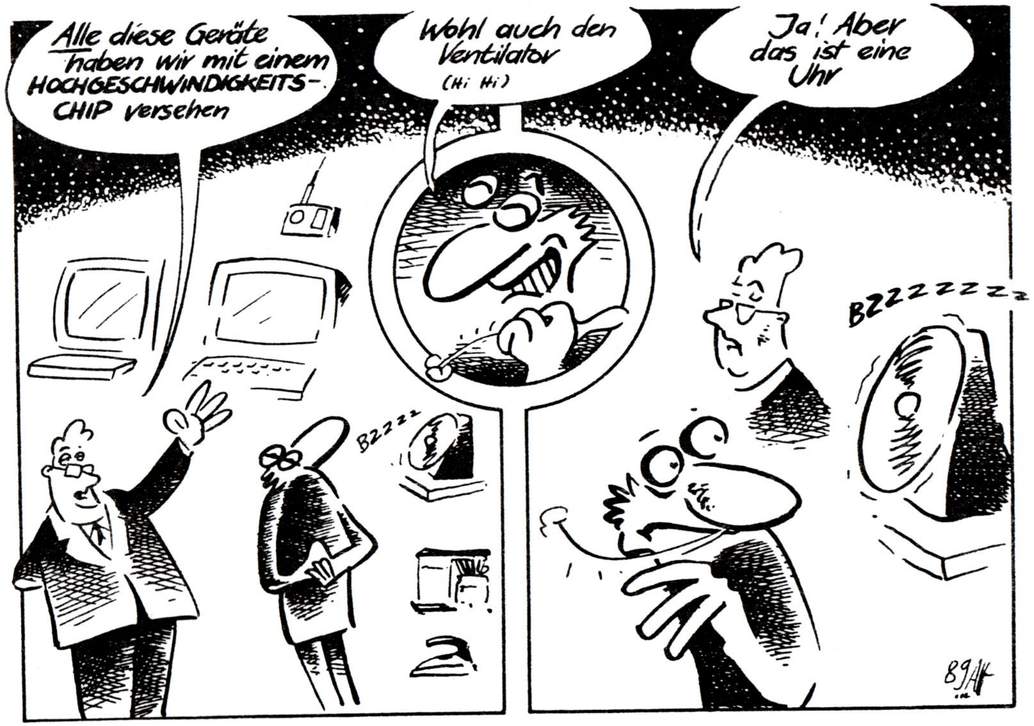
Auf der Diskette, die ich dem Club zur Verfügung stelle, kann man alles in MaschUnt/SRC nachlesen. (geschrieben mit Zeus)



Ein klebriges Problem

Lange Zeit haben mir meine Mitsubishi-Laufwerke jetzt gute Dienste geleistet, aber so allmählich hat die Kiste immer schlechter gebootet und besonders auf einem Laufwerk immer wieder Schreib- und Lesefehler gebracht. Am Ende war es so schlimm, daß ich den Kopf erst etliche Male per Diskmonitor von einem Ende zum anderen fahren mußte, bevor er mit einem deutlich hörbaren Klack aufgesetzt und der Rechner gebootet hat. Nachdem ich das Laufwerk dann ausgebaut und zerlegt hatte, stellte sich schließlich ein Gummi der Marke 'Klebrig spezial' als Übeltäter heraus. Die Idee, auf die Anschläge des Kopfladmagneten einen dünnen Gummi zu kleben ist sicher im Interesse der Geräuschdämpfung, aber im Laufe der Zeit hat sich der, an den der Anker in der Ruhestellung drückt, mehr oder weniger in Wohlgefallen aufgelöst und einen sehr klebrigen Rückstand hinterlassen. Da konnte der Magnet ziehen, wie er will, erst durch die Vibrationen beim Fahren des Kopfes hat sich der Kleister für einige Zeit gelöst; bis zum nächsten längeren Stillstand. Ein Scheibchen Pappe und ein Tropfen Sekundenkleber haben dem dann sehr schnell ein Ende gemacht und seitdem laufen sie wieder einwandfrei.
Es ist gar nicht so schlimm, ein Laufwerk auseinander zu nehmen. Solange man den Köpfen nicht zu nahe kommt, kann eigentlich gar nichts passieren. Natürlich sollte man die leicht erreichbaren Potis nicht zu Reinigungszwecken auch verdrehen, aber mit etwas Vorsicht ist das kein Problem. Die nach dem Zusammenbau übriggebliebenen Schrauben und Teile sind eine gute Ergänzung für den häuslichen Werkzeugkasten.

Alexander Schmid



Star NL10 abwechselnd am CP/M-Rechner und am PC

Helmut Bernhardt

Eigentlich sollte es dem Drucker doch egal sein, ob jenseits des Centronics-Eingangs ein CP/M-Rechner oder ein Big Blue Nachbau Ursache für zu beschmutzendes Papier ist. Das ist es ihm auch und dadurch hat man die Schwierigkeiten mit den deutschen Umlauten, wenn der Drucker nicht der gleichen Kaste angehört wie der Computer. Der Drucker geht davon aus, daß es den ASCII-Code gibt und der für ihn genauso verbindlich wie für alle anderen ist. Schließlich ist der ASCII-Code ja ein allgemeiner Standard.

Das war vor einigen Jahren mal ein einheitlicher Standard. Damals genügten 7 Bit, um damit 128 Zeichen zu definieren. Mit 128 Zeichen waren alle amerikanischen Zeichen und eine Reihe Steuerzeichen zu codieren. Um der Vielfalt der nationalen Sonderzeichen in der übrigen Welt Rechnung zu tragen, hatten Drucker ein Mäuseklavier, auf dem man beispielsweise einstellen konnte, daß anstelle der amerikanischen eckigen und geschweiften Klammern unter den gleichen Codes (5BH-5FH, 7BH-7FH) dann die entsprechenden deutschen Umlaute zu Papier gebracht werden. Andere Einstellungen machten den Drucker auch für Engländer, Franzosen, Spanier und noch manch anderes Völkchen nutzbar.

Bei der Einführung des PC hat der Urheber desselben einen neuen 8 Bit ASCII-Code zum Standard erhoben. Der IBM Zeichensatz Nr 2 wird heute von allen Druckern und natürlich auch von allen PC-Videokarten unterstützt. Diesem Standard haben sich auch andere Computer (z.B. Atari ST) angeschlossen. Der 8 Bit ASCII-Code besteht zur Hälfte (Codes 00H - 7FH) aus dem alten 7 Bit ASCII-Code. Sämtlichen von IBM als berücksichtigenswert erachteten nationalen Sonderzeichen wurden Codes mit gesetztem Bit 7 zugeteilt.

Da Big Blue aber auch so manchen Unsinn wie die Spielkarten-Farben, eine Teilmenge der TRS80-Grafik-Bauklötzchen und eine Menge von Linien-Grafikzeichen (für die Darstellung der furchtbar wichtigen Rahmen bei Pulldown- und Popup-Menüs) als wichtig erachtet hat, sind natürlich etliche Zeichen unter den Tisch gefallen. Anstelle des deutschen "Eßzett" muß man halt ein griechisches "Beta" nehmen.

Mein jüngst erworbener Drucker Star NL10 hat das Glück, in einer Zeit herausgekommen zu sein, als nicht nur PCs sondern auch noch CP/M-Computer eine Rolle spielten. Entsprechend war Star so umsichtig, den Drucker mit verschiedenen Betriebsprogrammen und Zeichensätzen (im 27256-EPROM) herauszubringen.

Wenn man nun einen NL10 mit PC-kompatiblem Betriebsprogramm-EPROM hat und jemanden kennt, der den gleichen Drucker mit dem CP/M-kompatiblen Betriebsprogramm hat, und jemanden kennt, der 27256-EPROMs lesen und 27512-EPROMs brennen kann (falls man das nicht selbst kann), dann kann man sich die Inhalte der beiden 27256-EPROMs in ein 27512-EPROM füllen (lassen) und über A15 des EPROMs (Pin 1) durch Anlegen von +5V oder GND das jeweilige Betriebsprogramm für den jeweiligen Rechner einstellen. Das Umschalten darf natürlich nicht bei eingeschaltetem Drucker erfolgen. Wegen des sonst beim Absturz vom Drucker erzeugten Lärms hat man sich aber schnell an diese Maßnahme gewöhnt.

Eleganter Adreßdecoder für GENIE 1

Helmut Bernhardt

Als ich bei der Fehlersuche in meinem GENIE 1 durch die zahlreichen selbstgeschnitzten Erweiterungen und das damit verbundene Strippengewirr selbst nicht mehr durchblickte, keimte der Gedanke, das alles sehr viel übersichtlicher und völlig neu zu gestalten. Als Ergebnis kam ein kleines Board heraus, das in den Sockel des Z80 zu stecken ist und den Z80 selbst mit aufnimmt. Zusätzlich wird auf dem Board ein 27128-EPROM (das anstelle der dann überflüssigen Original-ROMs das Level-II-Basic enthält), ein 74LS273-Latch, ein PAL16L8 (das ein Freigabesignal für das PAL16L8 und einige zusätzliche Freigabesignale erzeugt) und ein Adreßdecoder-PAL20L8 (das die Freigabesignale für alle memory mapped Systemkomponenten erzeugt) untergebracht.

Die Ausgänge des Latch haben folgende System-Steuerfunktionen:

```
Q0 = 0 : 0000H-2FFFH ist beim Lesen ROM und beim Schreiben RAM
   = 1 : 0000H-2FFFH ist beim Lesen und Schreiben RAM

Q1 = 0 : 37E0H-37EFH ist Floppy, Drucker, INT-Status, Drive-Select
   = 1 : 37E0H-37EFH ist RAM; Floppy u.s.w. sind über die Ports E0H-
                     EFH erreichbar

Q2 = 0 : 3800H-38FFH ist beim Lesen die Tastatur und beim Schreiben
                     RAM
   = 1 : 3800H-38FFH ist beim Lesen und Schreiben RAM

Q3 = 0 : 3C00H-3FFFH ist Video-RAM
   = 1 : 3C00H-3FFFH ist RAM

Q4 = 0 : die memory mapped Systemkomponenten liegen im Bereich
         0000H-3FFFH
   = 1 : die memory mapped Systemkomponenten liegen im Bereich
         C000H-FFFFH
```

Unabhängig vom Zustand von Q0-Q4 ist dauerhaft bei 3000H-37DFH, 37F0H-37FFH und 3900H-3BFFH RAM eingeblendet. Anstelle eines Sonder-ROM war mir RAM bei 3000H wichtiger.

Das Board erzeugt die Freigabe-Signale /ROM, /RAM, /FLO, /KB und /VID und das Steuersignal /MRD für die Lesetreiber des RAMs. Diese Signale sind an die entsprechenden Punkte auf dem CPU-Board zu führen; die ursprünglichen Signale sind zu durchtrennen.

Frei zum CPU-Board und zum RB-EXP1 zu verdrahtende Signale

Signal	Pin des 20L8	zu verbinden mit	zu durchtrennende Leitung	
/RAM	22	Z37,74LS367,Pin15	von Z36,74LS04,Pin2	#1
/MRD	17	Z9,74LS367,Pin1	von Z21,74LS20,Pin8	#2
/KB	20	Z7,74LS368,Pin1	von Z35,74LS32,Pin3	
/VID	19	Interboard-Conn. Pin10	von Z37,74LS32,Pin11	
/FLO	18	RB-EXP1, Z18, 74LS15,Pins2,14	RB-EXP1, Z14, 74LS32,Pin11	#3

#1 bei Aufrüstung auf 64K (256K) RAM besteht diese Verbindung als freie Verdrahtung; dieser Draht wird entfernt
#2 bei Entfernen von Z21,74LS20 entfällt das Durchtrennen
#3 Z13,74LS30 auf dem EXP1 kann entfernt werden
anstelle von /MWR von Z15,74LS32,Pin3 auf dem CPU-Board muß /BWR von Z16,74LS367,Pin11 an das EXP1 geführt werden, damit der Betrieb über die Ports E0H-EFH möglich ist

Das Freigabesignal /ROM ist auf dem Huckepack-Board direkt an die Pins 20 und 22 des EPROMs geführt.

Die Erzeugung dieser Freigabesignale ist aus dem Assembler-Listing für das PAL20L8 zu ersehen. Die Adreßeingänge A14 und A15 dieses PALs sind nicht die Adressen A14 und A15 der CPU. Diese Signale werden aus 2 XOR-Gattern (74LS86) aus den Adressen A14 und A15 der CPU erzeugt. Mit Q4 des Latch läßt sich damit steuern, ob A14 und A15 invertiert oder direkt an das PAL gelangen.

PAL-Assembler-Listing für das Adreßdecoder PAL20L8

```
PAL20L8
GENIE 1 Adreßdecoder mit mm-I/O-Switching
                  04.09.88  H. Bernhardt

A13 A11 MERQ RD A8 A9 A10 Q3 A15 Q1 PEX GND
ADR A14 MRD Q0 ROM FLO VID KB Q2 RAM A12 VCC

/ROM  = /Q0 */MERQ */RD */A15 */A14 */A13
      + /Q0 */MERQ */RD */A15 */A14 * A13 */A12
/FLO  = /Q1 */MERQ */A15 */A14 *A13 *A12 */A11 *A10 *A9 *A8 */ADR
      +  Q1 */PEX
/KB   = /Q2 */MERQ */RD */A15 */A14 *A13 *A12 *A11 */A10 */A9 */A8
/VID  = /Q3 */MERQ */A15 */A14 *A13 *A12 *A11 *A10
/RAM  = /MERQ *ROM *FLO *KB *VID
/MRD  = /MERQ */RD *ROM *FLO *KB *VID
```

Um bei den zur Verfügung stehenden Pins des PAL20L8 auch das Signal /FLO voll ausdecodieren zu können (A4-A15), mußte eine Gruppe von Adressen (A4-A7) schon zusammengefaßt (/ADR) und an einen Pin des PALs geführt werden. Auch für das alternativ von einer Portadresse herzuleitende /FLO war ein vorheriges Ausdecodieren von /IORQ, A4-A7 nötig (/PEX). Diese Signale werden vom PAL16L8 erzeugt, an das für das Erzeugen von /OUTFC (Freigabe des Steuer-Latch) ohnehin schon die Adressen A0-A7 und /IORQ gelangen.

PAL-Assembler-Listing für das I/O-Decoder-PAL16L8

```
PAL16L8
GENIE 1 Portdecoder
                 04.09.88  H.Bernhardt

A7 A6 A5 A4 A3 A2 A1 A0 WR GND
RD OFC IOF0 IOF4 IORQ IFC OFB PEX ADR VCC

/OFC   = /IORQ */WR *A7 *A6 *A5 *A4 *A3 *A2 */A1 */A0   ;Pin12
/IFC   = /IORQ */RD *A7 *A6 *A5 *A4 *A3 *A2 */A1 */A0   ;Pin16
/OFB   = /IORQ */WR *A7 *A6 *A5 *A4 *A3 */A2 *A1 *A0    ;Pin17
/IOF0  = /IORQ *A7 *A6 *A5 *A4 */A3 */A2                ;Pin13
/IOF4  = /IORQ *A7 *A6 *A5 *A4 */A3 *A2                 ;Pin14
/PEX   = /IORQ *A7 *A6 *A5 */A4                         ;Pin18
/ADR   = A7 *A6 *A5 */A4                                ;Pin19
```

Da das PAL16L8 noch einige freie Ausgangs-Pins hatte, wurden noch einige weitere Port-Freigabe-Signale decodiert.

Funktion der Port-Freigabesignale des PAL16L8

/OFC (/OUT FCH) wird intern zur Freigabe des Sytem-Steuerlatch benutzt
/IFC (/IN FCH) steht für die Freigabe eines über Port FCH zu lesenden Treibers zur Verfügung

/OFB (/OUT FBH) Freigabe-Signal, um über Port FBH ein Latch zu beschreiben; kann für die Freigabe des 74LS273 (Pin11) auf dem 256K-Banker benutzt werden (A0-A5 brauchen dann nicht mehr zum Banker geführt zu werden, A6 und A7 sind für das Erzeugen einer 8Bit-Refreshadresse dort aber weiterhin nötig; das 74LS30 auf dem Banker kann entfallen)

/IOF0 und /IOF4 (/In/OUT F0H-F3H und /IN/OUT F4H-F7H) Freigabe-Signale für z.B. Z80-Peripherie-Bausteine mit 4-Portadressen (PIO, SIO, CTC, DMAC, PPI8255...)

Die Verwendung eines 27128-EPROMs (wovon es auch 150ns-Versionen gibt) erlaubt bedeutend höhere Taktraten als die ursprünglichen ROMs. Wenn dann auch noch das Timing der Signale /RAS, /MUX und /CAS anders gelöst wird, spielen auch die RAMs mit. 5,3MHz sind auch ohne große Eingriffe möglich.

Da das ROM nun innerhalb der Treiber der CPU-Signale liegt, muß zur Steuerung der Daten-Lesetreiber auch das Signal /ROM herangezogen werden. Diese Treiber dürfen nicht freigegeben werden, wenn das ROM gelesen wird.

Um dies zu erreichen, wird Pin9 von IC1,74LS14 durchgekniffen und hochgebogen. Auf IC1 wird ein 74LS08 mit den Pins 7 und 14 huckepack aufgelötet. Alle anderen Pins des 74LS08 werden hochgabogen. Der abgekniffenen und hochgebogene Pin9 des IC1 wird mit dem seitlich gebogenen Pin8 des 74LS08 verlötet. Pin10 des 74LS08 wird mit Pin22 des 27128-EPROMs und Pin9 des 74LS08 mit Pin3 von IC14,74LS00 verbunden (Abbildung).

Zur Freigabe der Speicherlese-Treiber erzeugt das PAL20L8 das Signal /MRD, das an Pin15 von IC9,74LS367 zu legen ist. Durch Entfernen von IC21,74LS20 wird das bisherige Signal entfernt.

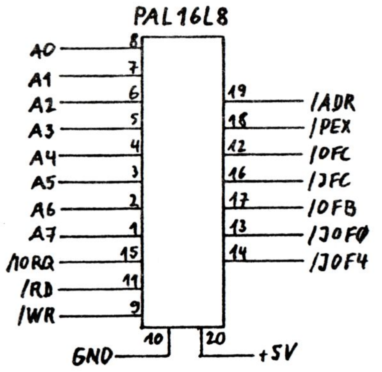
Außer IC21 sind auch IC22,74LS156 und IC25,74LS139 überflüssig geworden. Diese ICs können ebenfalls entfernt werden. Und natürlich können auch die alten ROMs rausgeschmissen werden.

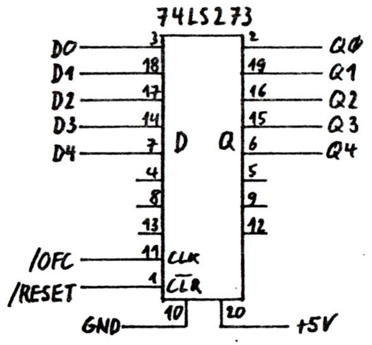
Beim Aufbau des Huckepack-Boards sind folgende Punkte zu beachten:

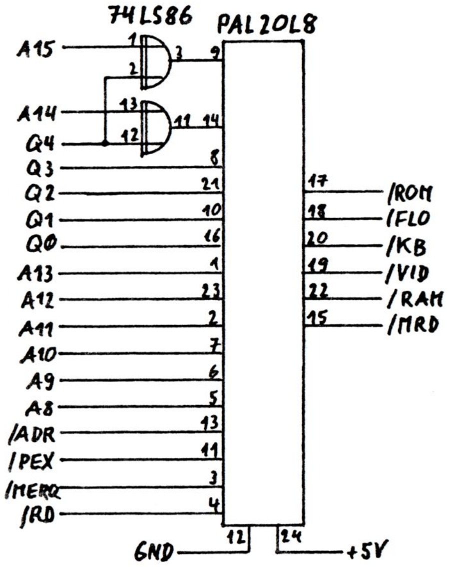
Vor dem Einlöten irgendwelcher Bauteile sind zunächst sämtliche Drahtbrücken zu legen. Als erstes sind dann der Sockel für den Z80 und der Stecker für das Anflanschen des Boards in den Z80-Sockel auf dem CPU--Board in geeigneter Reihenfolge einzulöten. Der Stecker wird auf der Lötseite und der Sockel auf der Bestückungsseite eingebaut. Dabei erschwert das zuerst eingebaute Teil den Einbau des anderen. Was zuerst angelötet wird, sollte eine gründliche Besichtigung der Teile ergeben.

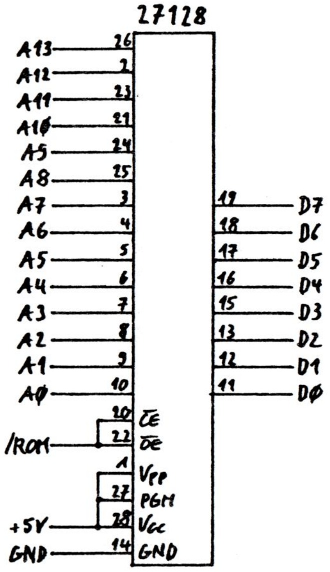
Es sollten alle ICs gesockelt werden. Sowohl beim EPROM als auch bei den PALs kann irgendwann der Wunsch nach geändertem Inhalt aufkommen. Und für Testzwecke bei der Fehlersuche ist es auch bei Standard-ICs zweckmäßig, wenn man diese auswechseln kann oder einfach nur einzelne Pins aus der Fassung biegen kann.

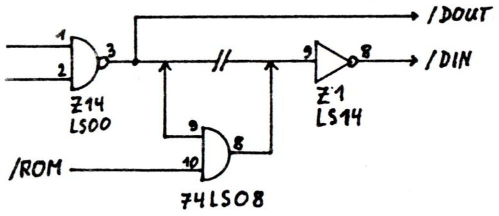
Ungebohrte Boards, mit Level-II befüllte 27128-EPROMs und gebrannte PALs kann ich zum Selbstkostenpreis zur Verfügung stellen.

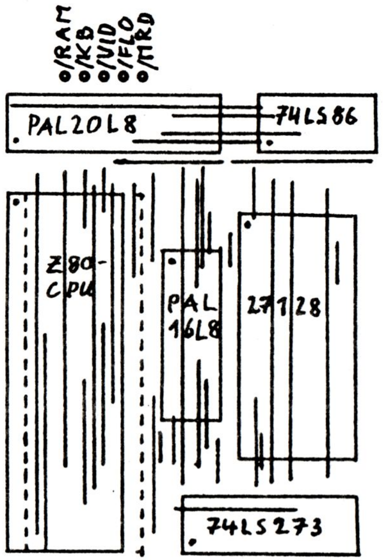


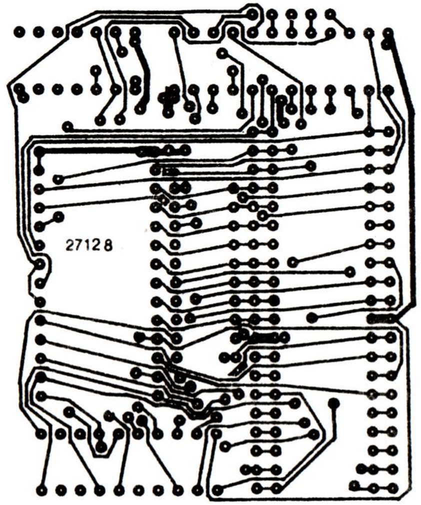


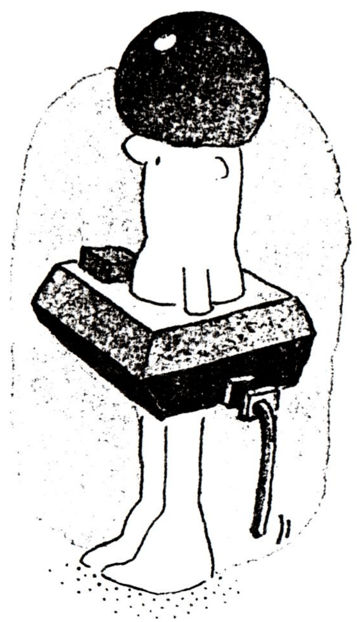












Es lebe der Standard !

Hercules Graphics Adaptor meets Trash 80

Helmut Bernhardt

Genau besehen hat es auch Vorteile, daß der Industrie Standard einen so niedrigen Standard hat. Insbesondere die Herkules-Karte, die dort schon nicht unbedingt als schlechteste Video-Karte gilt, ist schon fast zum TRS-80 kompatibel. Na ja, nicht ganz zu dem, aber doch schon zum GenieIIIs-Video-Interface. Wie im GIIIs gibt es dort eine 2K große Text-Video-Seite (80x25 Zeichen) und ebenso memory mapped. Zusätzlich bietet die Hercules aber auch noch einen 2K großen Attribut-Speicher, womit sich für jedes Zeichen auf dem Bildschirm einzeln Attribute setzen lassen. Und schließlich wird auf der Hercules wie im Genie IIIs mit 2 HRG-Pages mit je 32K Größe (ebenfalls bei beiden memory mapped) Grafik dargestellt. Bei beiden Geräten wird alles von einem CRTC6845 gesteuert.

Die maßgeblichste Inkompatibilität ist die Tatsache, daß bei der Hercules Text- und Attribut-Speicher ineinander verschoben sind. Dort werden innerhalb eines linearen 4K-Bereichs alle geradzahligen Adressen für Textzeichen und alle ungeradzahligen Adressen für die dazugehörigen Attribute verwendet. Weniger störend sind die 10Bit-Portadressen des Industrie-Standards, mit denen die internen Register der Karte angesprochen werden. A8 und A9 kann man der Karte bei Z80--Portzugriffen einfach vorgaukeln.

Folgende Tabelle stellt die Ähnlichkeiten und Abweichungen zwischen Hercules und GenieIIIs-Video-Interface zusammen:

Funktion	Hercules	GenieIIIs
Text-Speicher	B0000-B0FFF (mit A0=0)	3800(3C00)-3FFF
Attribut-Speicher	B0000-B0FFF (mit A0=1)	nicht vorhanden
HRG-Seite 0	B0000-B7FFF	8000-FFFF
HRG-Seite 1	B8000-BFFFF	8000-FFFF
CRTC6845-Ports	3B4H/3B5H	F6H/F7H

Diese Ähnlichkeit ließ mich vor längerer Zeit schon mal darüber brüten, wie man die Hercules anstelle des dort recht lausigen Video--Interface in TRS-80- und Genie-Rechnern einsetzen könnte. Zunächst dachte ich an einen Adapter, der mir die nötige Adreßübersetzung macht und die Verbindung zwischen Genie-CPU-Board und Hercules herstellt. Dann dachte ich daran, daß sich dieser Aufwand für das ebenfalls recht lausige Genie-CPU-Board nicht lohnt. Inzwischen ist um diesen reinen Adapter ein ganzes CPU-Board gewachsen, das noch einiges mehr bietet als das Genie-CPU-Board und dabei auf die Maße einer Europa-Karte geschrumpft ist. Das Board enthält all das, was mir bislang lieb geworden ist:

(1) HD64180-CPU
(2) 2 serielle Schnittstellen
(3) 1MB RAM bestehend aus 8 mal 511000-Megachips
(4) auf 1MB erweiterte MMU des HD64180
(5) 27128-EPROM mit Level II BASIC und Initialisierungen

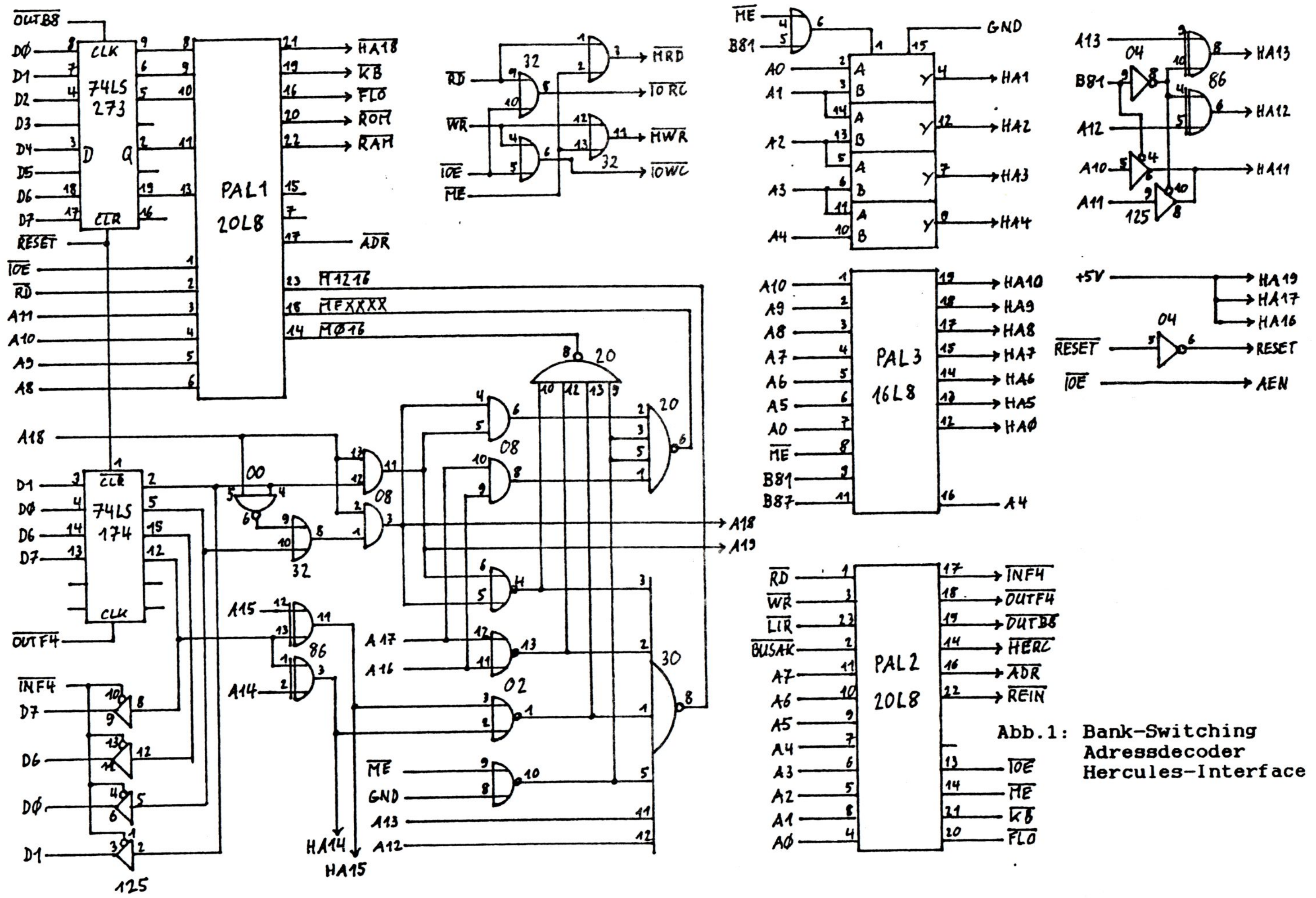
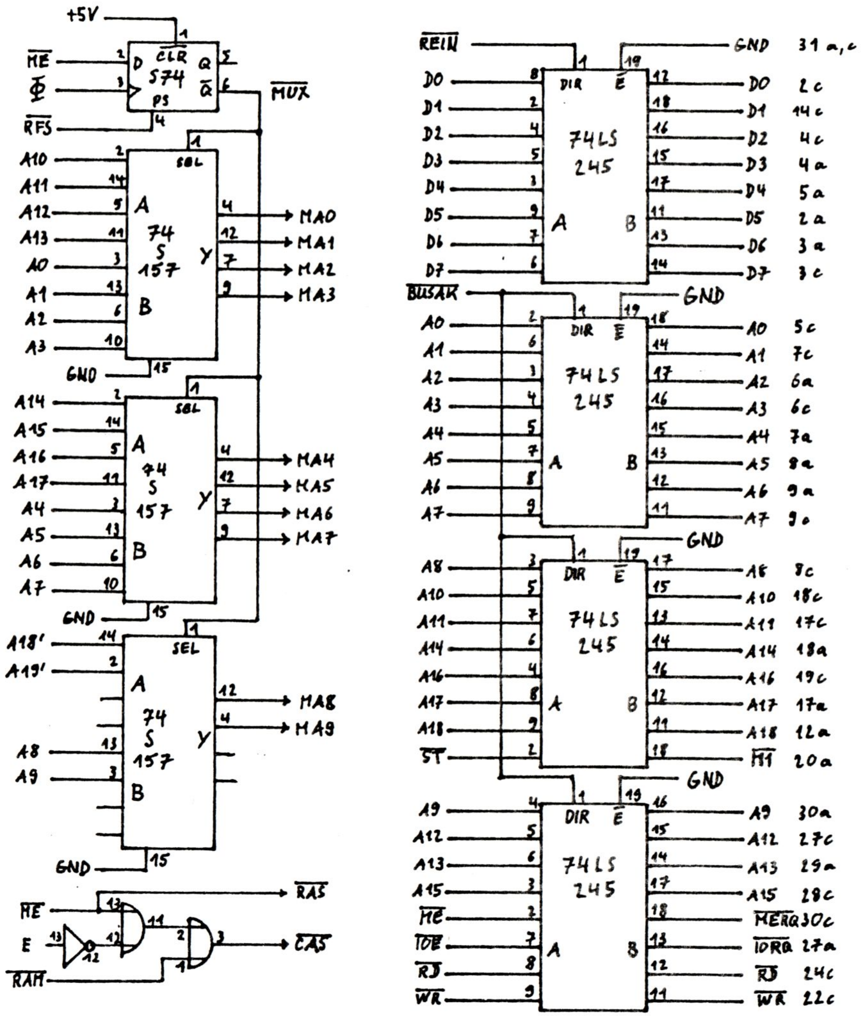


Abb.1: Bank-Switching
Adressdecoder
Hercules-Interface



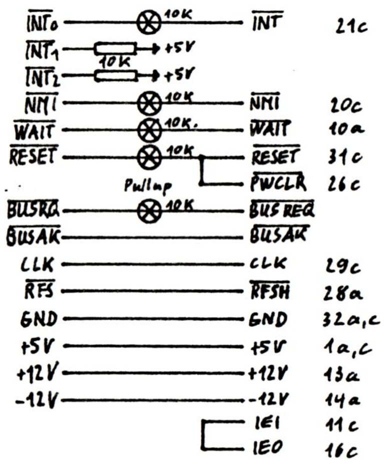


Abb.2: RAM-Steuerung
ECB-Bus-Steuerung

HEFT
28
Oktober
1989

[6] ECB-Bus mit Buspufferung und IM2- und DMA-fähiger Steuerung
[7] Interface zur Hercules
[8] komfortables Switching für memory mapped I/O

Damit steht mir ein voll Geniel-kompatibler (nur Spiele lassen sich ohne Bauklötzchengrafik nicht so richtig spielen) Computer zur Verfügung. Die Hercules stellt mir dafür den Standard-64x16-Bildschirm, eine 80-Zeichen-Karte und zwei hochauflösende Grafikseiten mit je 720x348 Punkten zur Verfügung, und das alles für 100,-DM.

Die eigentlichen Kosten fallen bei der Beschaffung der RAMs an. Zugegeben - 256K hätten's beim Geniel auch getan, aber das Teil-Layout mit CPU, RAMs, EPROM, RAM-Steuerung und Treibern für die V24-Schnittstellen brauchte ich nur aus der Schublade zu ziehen und beim Rest habe ich mir dann auch keine große Mühe mehr gemacht. So ist

Der Aufbau der Karte

zu einem richtigen kleinen Abenteuer geworden. Das Layout beschränkt sich auf einseitige Leiterbahnführung und versucht, dort so viele Verbindungen unterzubringen, wie irgend möglich. Nur im hinteren Teil der Karte (der aus der Schublade), war es möglich, alles mit Drahtbrücken zu erledigen - bis auf das EPROM, das muß auch von Hand verdrahtet werden, dort sind nur D0-D7, +5V und GND angeschlossen. Was von Hand zu verdrahten ist, kann aus der Verdrahtungsliste ersehen werden.

Wer nun immer noch glaubt, daß er dieses Projekt durchstehen kann, und von mir zum Selbstkostenpreis ein geätztes, ungebohrtes Board bekommen hat, muß zunächst mal 1-2 Stunden mit der Bohrmaschine zubringen. Die 3 Stecker für VG64-Leiste, für die Pfostenfeld-Leiste zum Aufstecken der Hercules-Karte und die Pfostenfeld-Leiste für die seriellen Schnittstellen müssen 1mm-Bohrungen erhalten, alle anderen Bohrungen sind in 0,8mm niederzubringen.

Wenn man sich davon wieder erholt hat, legt man sämtliche Drahtbrücken (Drahtbrücken- und Bestückungs-Plan) und fängt dann erst an, Kleinteile und IC-Sockel einzulöten. Da die Drahtbrücken teilweise recht eng liegen, sollte grundsätzlich dünner Kupfer-Lackdraht verwendet werden. Bei Drahtbrücken, die +5V und GND führen sollen (solche Brücken enden meistens an etwas breiteren Leiterbahnen), ist etwas dickerer isolierter Schaltdraht vorzuziehen.

Es sei noch gesagt, daß im dichter besiedelten vorderen Bereich (beim Busstecker) in der ersten Reihe ein IC nicht zu bestücken ist. Weil der Platz halt da war, habe ich da vorsichtshalber mal die Lötaugen hinterlegt, damit es spätere Bastler bei Erweiterungen einfacher haben.

Auf der Busplatine oder frei improvisiert auf dem CPU-Board müssen die CPU- Signale /BUSRQ, /RESET, /WAIT, /NMI und /INT0 mit 10kOhm an +5V gelegt werden. /INT1 und /INT2 müssen direkt am HD64180 über 10k an +5V gelegt werden. Wenn diese Eingänge nicht genutzt werden, genügt ein gemeinsamer Widerstand.

Die 62polige Steckleiste zum Anschluß der Hercules ist für eine spezielle, besonders kleine Karte gedacht, die direkt neben dem für PC-Karten üblichen Platinenrand-Stecker zwei Reihen Durchkontaktierungen aufweist, die freundlicherweise im RM2,54 Rastermaß angelegt sind und eine Weite von 1mm haben. Wenn hier von der Bestückungsseite eine zweireihige Buchsenleiste eingelötet wird, und auf dem CPU-Board die Pfostenfeld-Leiste mit etwa um 1cm verlängerten Beinchen eingelötet wird, können beide Karten direkt aufeinander gesteckt werden.

Der Fernseher darf leider nicht der alte vom Geniel bleiben. Die Hercules benötigt einen TTL-Monitor (ohne Gehäuse und Netzteil bekommt man sowas im Elektronik-Ramsch-Versandhandel für ca. 40,-DM). Andererseits gibt es auch TTL/BAS-Wandler, denen man aber ein aus "Video" und "Intensity" zusammengeführtes Signal anstelle des reinen Video--Signals zuführen sollte, um nicht auf die Helligkeitssteuerung der Hercules verzichten zu müssen.

Programmierung der Hercules-Karte

Interne Ports (die Portadressen sind die auf die 8Bit-Decodierung durch das Interface reduzierten Adressen; im PC müssen zusätzlich A8 und A9 high sein)

B4H Adreßregister des CRTC-6845
B5H Datenregister des CRTC-6845
B8H Display-Betriebsart-Steuerkanal
B9H Display-Status-Kanal
BBH Light-Pen FlipFlop zurücksetzen
BCH Drucker-Datenport
BDH Drucker-Statusport
BEH Drucker-Steuerkanal
BFH Konfigurations-Schalter

Für die Programmierung des CRTC-6845 ist ein Datenblatt erforderlich. Durch Trial-and-Error Methoden soll man es schaffen können, den Monitor dauerhaft zu disablen. Es sei nur soviel gesagt, daß der CRTC eine ganze Menge Datenregister besitzt, wovon nur die ersten 16 relevant sind. Sie haben alle die Adresse B5H. Auf welches dieser Register jeweils über Port B5H zugegriffen werden kann, bestimmt der Eintrag in das Adreßregister B4H. Die Initialisierungsroutine im ROM enthält an der Adresse 3000H 5 Tabellen mit je 16 Daten für die ersten 16 Datenregister des CRTC und eine Routine, die dem CRTC diese Daten einverleibt, wenn HL auf den Anfang der jeweiligen Tabelle zeigt. Dort mag sich der Softi informieren, der lieber noch exotischere Bildschirm-Formate einstellen möchte. Die grundsätzliche Bedeutung der CRTC-Register ist aus mc 1/88 S.113 zu entnehmen.

Der Display-Betriebsart-Steuerkanal (Port B8H) der Hercules reagiert nur auf die Bits 1, 3, 5 und 7. Die geradzahligen Datenbits haben dort keine Funktion. Da die Einstellungen dieses Registers für die Adreßübersetzung des Interface zur Hercules von Bedeutung sind, wurde dieser Port auf dem CPU-Board nachgebildet und um die geradzahligen Bits bereichert. Setzen und Zurücksetzen einzelner Bits dieses Ports durch Auslesen, Bit verändern und Zurückschreiben ist also nicht möglich. Man muß dafür im RAM über den Zustand des Ports Buch führen.

In der folgenden Tabelle sind die Funktionen der einzelnen Bits des

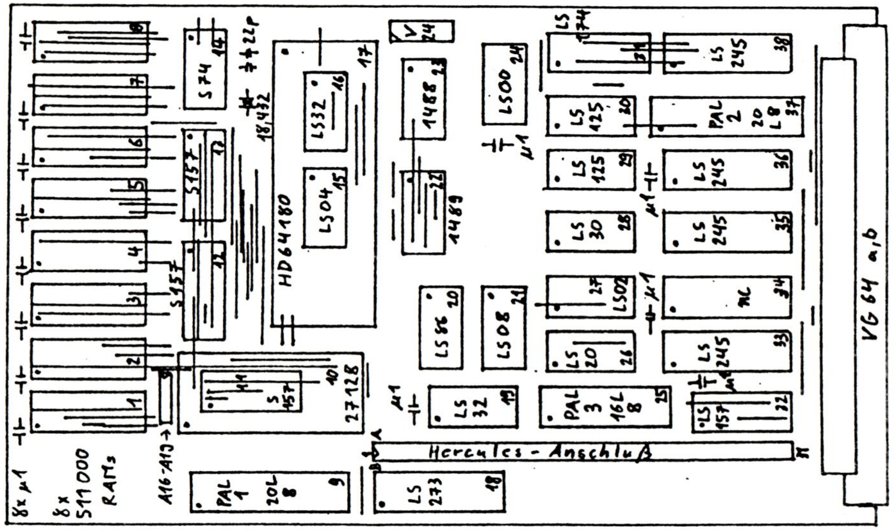
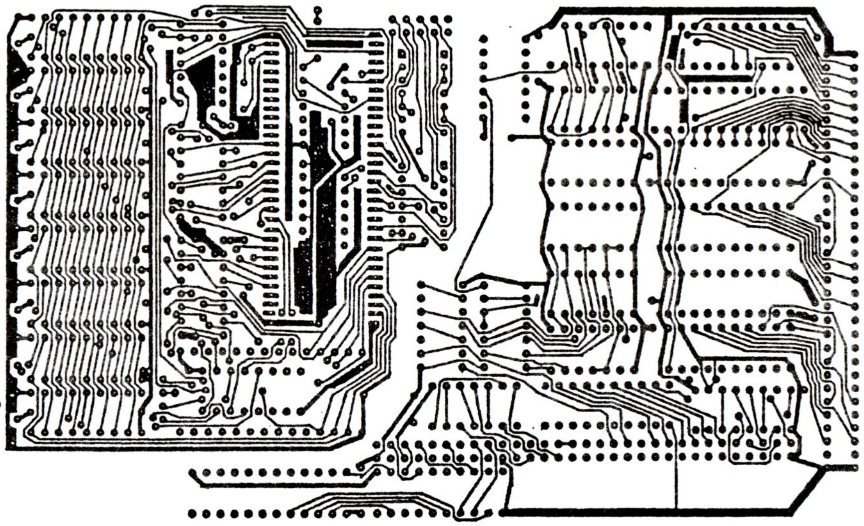


Abb.3: Bestückung- und Dratbrückenplan, Teil-Layout



erweiterten Ports B8H zusammengestellt. Die urgsprünglichen Funktionen des Ports auf der Hercules, die nach wie vor gelten, sind mit # markiert.

Display-Mode-Control-Port, B8H

```
D0 =0 : 1K Video-RAM bei 03C00-03FFF (64x16), Keyboard bei 03800-038FF
   =1 : 2K Video-RAM bei 03800-03FFF (80x25), Keyboard disabled
D1 =0 : Text-Video-Mode enabled, HRG disabled                         #
   =1 : HRG enabled, Text-Video disabled                              #
D2 =0 : ROM bei 00000-036FF (nur bei /RD, bei /WR RAM)
   =1 : ROM disabled, bei 00000-036FF liegt RAM
D3 =0 : die Hercules gibt kein Video-Signal an den Monitor            #
   =1 : das Video-Signal der Karte gelangt zum Monitor                #
D4 =0 : Floppy, Drucker u.s.w. bei 037EX memory mapped
   =1 : 037EX ist RAM, Floppy u.s.w. über Ports EXH
D5 =0 : D7=1 des Attribut-Bytes bedeutet "heller Hintergrund"         #
   =1 : D7=1 des Attribut-Bytes bedeutet "Blinken"                    #
D6 =0 : Text- bzw HRG-RAM sind freigegeben
   =1 : Text- bzw HRG-RAM gesperrt, dort RAM eingblendet
D7 =0 : HRG : Page 0 selected                                         #
        Text: 2K Textspeicher eingestellt
   =1 : HRG : Page 1 selected                                         #
        Text: 2K Attributspeicher eingstellt
```

Die Einteilung des 1MB-Speichers

Der HD64180 (in der hier verwendeten DIL64-Form) kann nur 512K Speicher adressieren. Um den vollen 1MB-Speicher zu nutzen, muß durch zwei weitere Bits eines Ports (D0 und D1 von Port F4H) ein Banking von 256K-Blocks dem Banking der MMU des HD64180 überlagert werden. Die Selektion eines der 4 256K-Blocks erfolgt mit A18 des HD64180 und D0/D1 von Port F4H. Aus diesen 3 Signalen werden die Adressen A18 und A19 für das RAM hergeleitet.

Port F4H D1	D0	CPU A18	RAM A19	A18	256K--Block
x	x	0	0	0	0
0	0	1	0	1	1
0	1	1	0	1	1
1	0	1	1	0	2
1	1	1	1	1	3

Wenn A18 der CPU low ist, wird immer der unterste 256K-Block angewählt. Wenn A18 high ist, wird einer der 3 anderen Blocks selektiert. Welcher dieser 3 Blocks selektiert wird, hängt von den Zuständen von D0 und D1 des Ports F4H ab. Diese Vorgaben gelten sowohl für die MMU als auch für den DMA-Controller.

Nach einem RESET ist die MMU des HD64180 immer so eingestellt, daß der logische 64K-Z80-Adreßraum in den unteren 64K des unteren 256K-Blocks liegt. Um die Standard-Software (NEWDOS/80, CP/M) zu fahren, braucht die MMU nicht umprogrammiert zu werden. Die Decodierung von ROM und

memory mapped I/O schließt auch die Adressen A16-A19 mit ein, so daß diese Baugruppen im physikalischen Adreßraum 00000H-03FFFH liegen. Nur der HRG-Speicher der Hercules-Karte mit 2mal 32K wäre bei 08000H eingeblendet etwas lästig und wurde deshalb an das obere Speicherende bei F0000H verbannt. Für die MMU und den DMAC des HD64180 sind dies die Adressen 70000H-7FFFFH, wobei D0 und D1 von Port F4H gesetzt sein müssen, damit der HRG-Seicher adressiert wird.

Neben der Grundeinstellung (D7 an Port F4H low, Zustand nach RESET) gibt es noch eine andere Lage für die memory mapped Baugruppen. Wenn D7=1 an Port F4H ausgegeben wird, liegen ROM, Floppy, Drucker, Tastatur und Text-Video-RAM bei 0C000H-0FFFFH und die untersten 16K sind RAM.

Und schließlich läßt sich innerhalb dieser 16K (wo auch immer die memory mapped Baugruppen eingeblendet sind) sehr komfortabel umkonfigurieren. Darüber gibt die Tabelle der Bedeutung der Bits des erweiterten Ports B8H und die Map der Speicheraufteilung Auskunft. In allen Bereichen, wo die jeweilige memory mapped Baugruppe ausgeblendet ist, liegt dann automatisch RAM vor.

Der Text-Bildwiederholspeicher

Die Hercules-Karte stellt im Text-Modus einen 2K großen Bildwiederholspeicher für Text und einen 2K großen Attribut-Speicher zur Verfügung. Diese Speicher sind ineinander verschoben; innerhalb eines linearen 4K-Bereichs stellen die geraden Adressen den Text-Speicher und die ungeraden Adressen den Attribut-Speicher dar.

Video-Attribute waren aber damals, als der TRS-80 geboren wurde, noch unbezahlbarer Luxus. Deshalb scheint der Attribut-Speicher der Hercules der Kompatibilität zum TRS-80 im Wege zu stehen. Der TRS-80 braucht einen zusammenhängenden Bildwiederhol-Speicher von 1K Größe (und manche jüngeren Brüder können auch mit 2K Textspeicher umgehen).

Dieses unlösbare Problem läßt sich dadurch lösen, daß beim Text-Modus die Adressen A0-A10 der CPU (2K-Bereich) auf die Adressen A1-A11 der Hercules-Karte gelegt werden. Diese Umsetzung darf aber nur bei Zugriff auf den Text- und den Attribut-Speicher erfolgen. Bei HRG-Betrieb und bei Zugriff auf die Ports der Karte müssen alle Adressen direkt durchgeschaltet werden. Die für das Selektieren von Text- oder Attribut-Speicher zuständige Adresse A0 (zur Hercules) wird im Text--Modus von D7 des erweiterten Ports B8H und bei Zugriff auf Hercules--Ports und HRG-Speicher von A0 der CPU bedient. Diese Adreßsteuerung wird von PAL3 und einem 74LS157-Multiplexer erledigt.

Die damit auf 2K reinen Text-Speicher (und nach Umschalten 2K reinen Attribut-Speicher) beschränkte Bildschirmseite ist aber für den TRS--80, der nur 1K bei 3C00H dafür vorsieht, immer noch zu groß. Deshalb kann mit D0=0 an Port B8H (Einstellung nach der Initialisierung im ROM) die Freigabe nur der oberen Hälfte des Text-Speichers im Bereich 3C00-3FFF vorgegeben werden. Für das TRS-80-Format 64x16 Zeichen muß dann aber auch der CRTC-6845 entsprechend programmiert werden. Dies wird ebenfalls durch die Initialisierung im ROM erledigt. Das Betriebsystem findet dann beim anschließenden Booten eine voll TRS-80--kompatible Maschine vor.

Wenn D0=1 an Port B8H ausgegeben wird, ist der volle 2K-Text-Speicher im Bereich 3800-3FFF verfügbar. Um eine vernünftige Darstellung auf dem Bildschirm zu haben, muß dann aber auch der CRTC-6845 entsprechend initialisiert werden. Im 80x25-Modus ist die Tastatur dann aber nicht mehr zugänglich. Für das Auslesen der Tastatur muß dann immer erst in den 64x16-Modus zurückgeschaltet werden und nach dem Auslesen der Tastatur für weitere Bildschirm-Ausgaben wieder der 80x25-Modus eingestellt werden. Wenn dabei die Register des CRTC-6845 in Ruhe gelassen werden und nur mit D0 von Port B8H geschaltet wird, bleibt die Darstellung auf dem Bildschirm unbeeinträchtigt. Die Umschaltung betrifft dann nur die Zugriffsmöglichkeiten der CPU auf den Bildwiederholspeicher. Das für die Darstellung des Bildspeichers auf dem Monitor verantwortliche Auslesen des 6845 wird dadurch nicht belangt.

Die Initialisierungs-Routine im ROM füllt zunächst den gesamten Attribut-Speicher mit dem Standard-Wert 7. Das bedeutet, daß die Zeichen auf dunklem Hintergrund mit normaler Helligkeit dargestellt werden, wenn der Attribut-Speicher bei der Textausgabe nicht berücksichtigt wird.

Es sind allerdings dies die Zeichen entsprechend Big Blue's Verständnis von ASCII-Code. Die Zeichen mit den Codes 5BH-5FH und 7BH-7FH sind grundsätzlich die amerikanischen Zeichen [{geschweifte und eckige Klammern u.s.w.}]. Den deutschen Umlauten und auch allen anderen nationalen Sonderzeichen sind feste ASCII-Codes mit gestztem Bit7 zugeordnet. Das hat den Nachteil, daß bei allen von Standard-Programmen unter NEWDOS und CP/M ausgegebenen Texten anstelle der deutschen Umlaute die entsprechenden amerikanischen Zeichen erscheinen. Ein Vorteil besteht aber auch darin, daß amerikanische Sonderzeichen und sämtliche nationalen Umlaute gleichzeitig dargestellt werden können. Für Soft-Freaks besteht hier ein weites Betätigungsfeld in der Anpassung von Tastatur-Treibern und Standard-Programmen.

Die Bildschirm-Attribute

Wenn mit D1=1 und D7=1 an Port B8H die Text-Betriebsart und der Attribut-Speicher eingeschaltet werden, kann im Adreßbereich des Text-Speichers ein Attribut-Byte eingetragen werden, das die Darstellung des ASCII-Zeichens auf gleicher Adresse im Text-Speicher beeinflußt. Die Attribut-Bits wirken nur auf das eine Zeichen und nicht auf den gesamten Bildschirm. Wenn mit Attributen gearbeitet werden soll, muß für jedes Zeichen im Text-Speicher auch ein Byte im Attribut-Speicher eingetragen werden. Beim Scrollen muß auch der Attribut-Speicher genauso wie der Text-Speicher verschoben werden.

Die Bits 3 und 7 eines Attribut-Bytes haben eigenständige Funktionen, während die Bits 0-2 und die Bits 4-6 in Kombination miteinander zu benutzen sind:

D7 : die Wirkung von D7 eines Attribut-Bytes hängt davon ab, welcher Wert für D5 an Port B8H ausgegeben wurde.

Port B8H, D5=0:

D7(Attribut) =0 : Zeichen erscheint auf dunklem Hintergrund
=1 : Zeichen erscheint vor halbhellem Hintergrund

Port B8H, D5=1:

D7(Attribut) =0 : kein blinkendes Zeichen
=1 : Zeichen wird blinkend dargestellt
D3 = 0 : normale Helligkeit der Darstellung
= 1 : Highlighting

D0-D2, D4-D6 :

D6	D5	D4	D2	D1	D0	Bedeutung
0	0	0	0	0	0	Zeichen nicht dargestellt
0	0	0	0	0	1	unterstrichen dargestellt
0	0	0	1	1	1	normale Darstellung
1	1	1	0	0	0	inverse Darstellung

Die HRG der Hercules-Karte

Mit 720x348 Punkten Auflösung und 2 Seiten läßt sich schon sehr schöne Grafik auf die Mattscheibe bringen. Diese Auflösung benötigt natürlich auch Speicherraum für die beiden 32K-Pages. Diese Bereiche analog zum GenieIIIs einfach in den logischen Adreßraum der nach RESET eingestellten (unteren physikalischen) 64K zu legen, erscheint etwas unzweckmäßig zu sein. Es müßte dann eine Hälfte des ohnehin nur 64K großen, direkt erreichbaren Arbeitsspeichers ausgeschaltet werden. Für Grafik-Anwendungen müßte man sich festlegen, in welcher Hälfte der 64K die Grafik-Routinen laufen sollen.

Hier ist der HRG-Speicher ans obere Ende des physikalischen 1MB-Speichers verbannt worden. Das Besudeln des Grafik-Speichers erfolgt im einfachsten Fall durch blockweise Übertragung per DMA. Wer aber trotzdem gerne direkt in das HRG-RAM schreiben möchte, kann sich dieses durch die MMU in die untere oder obere Hälfte seines 64K-Arbeitsspeichers einblenden.

Nach einem RESET ist die Text-Betriebsart eingestellt. Mit D1=1 an Port B8H läßt sich auf die HRG-Betriebsart "Half-Mode" umstellen, wenn in Port BFH D0=1 eingetragen ist. Dabei ist die untere Hälfte (Page 0) erreichbar. Um auch die Page 1 erreichen zu können, muß auch D1=1 an Port BFH ausgegeben werden.

Während der Grafik-Betriebsart ist der Text-Bildspeicher nicht zugänglich. Da dieser aber physikalischer Bestandteil der Page 0 der HRG ist, wird der Inhalt des Text-Speichers bei Grafikausgaben an Page 0 überschrieben. Gleiches gilt für den Attribut-Speicher.

Die Tastatur

Dieses Gerät erleidet hier die größten Einbußen in der Präsenz. So ist der Bereich, in dem die Tastatur ausgelesen werden kann, auf den dafür nötigen Platz von 256 Adressen beschränkt. In der 64x16-Text-Betriebsart (D0=0 an Port B8H, die einzige Einstellung, in der die Tastatur erreichbar ist) stehen bei 3900H - 3BFFH ein paar Bytes frei verfügbaren Speichers bereit. Auch der Speicher parallel zur Tastatur kann bei eingeblendeter Tastatur beschrieben werden; auslesen läßt er sich nur, wenn auf Grafik-Betriebsart umgestellt wird.

Beim Arbeiten im 80x25-Textmodus ist die Tastatur nicht erreichbar. Dann liegt ab 3800H Bildwiederhol-Speicher vor. Um die Tastatur zwischendurch auszulesen, muß mit D0=0 an Port B8H vorübergehend auf 1K Video-RAM umgeschaltet werden. Da das aber nicht den Zugriff des CRTC-6845 auf den Bildspeicher beeinflußt, zeigt sich das Umschalten nicht in der Darstellung auf dem Bildschirm.

Wenn es auch Wahnsinn ist, eine Tastatur memory mapped als offene Matrix an einen Computer anzuschließen, weil bei höheren Taktraten und längeren Anschlußleitungen Probleme zu erwarten sind, kommt man bei einem TRS80-kompatiblen Rechner um diesen Unsinn nicht herum, wenn auch der Betrieb von OLDDOS/80 und der darunter laufenden Programme möglich sein soll.

Ein einfaches Interface (Abb.7) kann auf einem Stück Lochraster-Platine in Fädeltechnik aufgebaut werden und mit einer VG64-Steckerleiste auf den ECB-Bus gesteckt werden. Die Tastatur kann daran über ein nicht zu langes, 16poliges Kabel angeschlossen werden.

Die Freigabe des ROMs

Wie es sich für einen Trash-80-kompatiblen Computer gehört, liegen nach dem Einschalt-RESET die 12K Löffel-II-BASIC bei 0000H-2FFFH im ROM rum. Zusätzlich wurde hier auch der Bereich 3000H-36FFH als ROM decodiert. Hier sind die nötigen Initialisierungen der HD64180-CPU und der Hercules-Karte untergebracht. Der Einsprung bei 3050H muß im Level-II-ROM anstelle des ersten Sprungbefehls nach RST 00H gepatcht werden. Die Initialisierung endet mit einem Sprung an das dort vorher vorhandene Sprungziel 0674H.

Wenn D7 an Port F4H high ausgegeben wird, verschiebt sich die Lage des ROM-Inhalts von 0000-36FF nach C000-F6FF. Das ist zwar wenig sinnvoll, weil der Stoff auf dieser Adreßlage nicht läuft, ließ sich aber wegen des für CP/M nötigen Verschiebens des memory mapped I/O nach F7E0-FFFF nicht verhindern. Wenn das ROM aber ausgeschaltet wird (D2 an Port B8H high), stört es dort nicht mehr.

Das Abschalten des ROMs über D2 an Port B8H läßt sich natürlich auch in der ursprünglichen Lage bei 00000H durchführen. Da das ROM ohnehin nur beim Lesen freigegeben wird und beim Schreiben RAM eingeblendet ist (auch wenn das ROM nicht abgeschaltet ist) kann das Kopieren des ROM-Inhaltes in das RAM durch ein einfaches LDIR auf sich selbst (HL = DE = 0) erfolgen. Das wird in der Initialisierung vorsichtshalber auch schon mal gemacht.

Floppy, Drucker und der ganze Rest

Damit die Abwärtskompatibilität zum TRS-80 beibehalten wird, liegen diese Komponenten memory mapped bei 37EXH. Allerdings wurde die Option eingebaut, anstelle dieser Memory-Adressen auch die entsprechenden Ports E0H-EFH für das Ansprechen dieser Einheiten benutzen zu können. Dafür wird mit D4=1 an Port B8H auf Betrieb über Ports umgeschaltet.

Wer von seinem alten Genie oder TRS80 her noch einen RB-EXP1 Controller besitzt, hat damit ein weiterhin brauchbares Gerät vorliegen.

Diesem Board muß nur noch ein VG64-Stecker spendiert werden. Dieser Stecker wird auf einen 2 cm breiten Streifen Lochraster-Platine gelötet, der wiederum mit zwei schmalen Streifen Pertinax oder sonstigem Kunststoff mit dem RB-EXP1 verschraubt wird (Abb.6). Auf der Lötseite werden dann die ECB-Signale des Steckers mit den entsprechenden Punkten der beiden Anschlußstecker des EXP1 verbunden.

Anschluß des RB-EXP1 Controllers an den ECB-Bus

ECB-Bus Pin	Signal	RB-EXP1	
2c	D0	Pin 9	24poliger
14c	D1	Pin 10	Stecker
4c	D2	Pin 11	
4a	D3	Pin 13	
5a	D4	Pin 14	
2a	D5	Pin 15	
3a	D6	Pin 16	
3c	D7	Pin 17	
5c	A0	Pin 8	
7c	A1	Pin 7	
6a	A2	Pin 6	
6c	A3	Pin 5	
1a,c	+5V	Pin 24	
32a,c	GND	Pin 12	
13a	+12V	Pin 11	16poliger
15a	-5V	Pin 12	Stecker
21c	/INT	Pin 4	
31c	/RESET	Pin 5	
z.B. 23a	/FLO	Pins 2,14	74LS155

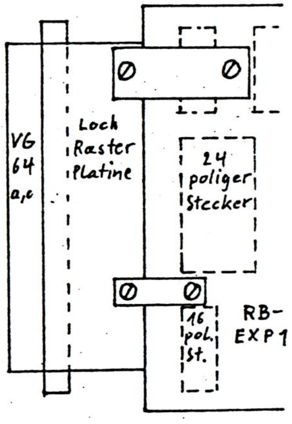


Abb.4: ECB-Bus-Anschluß der RB-EXP1

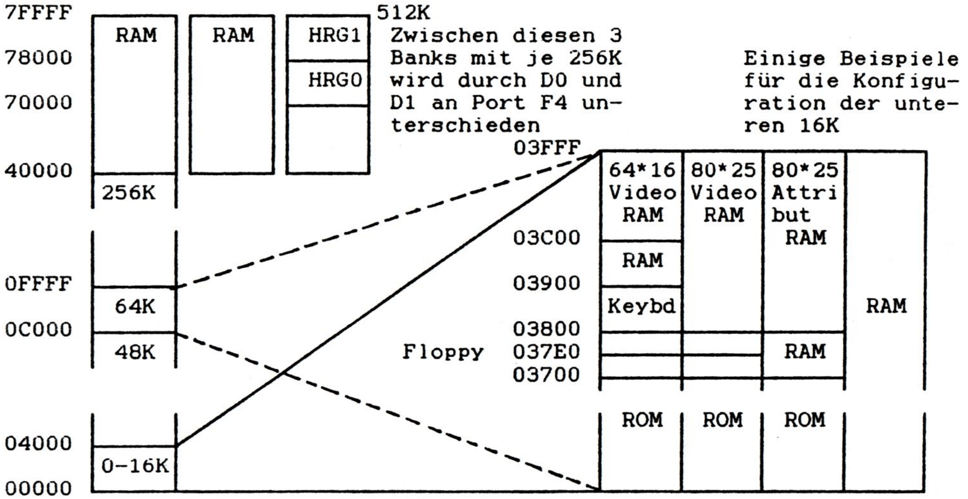
Da das Freigabe-Signal für den Controller bereits auf dem CPU-Board erzeugt wird und auch schon die Umschaltmöglichkeiten beinhaltet, sollte lieber dieses Signal verwendet werden als das auf dem EXP1 hergeleitete Signal /37EX. Dafür wird das Signal /FLO von PAL1 auf dem CPU--Board über eine nicht benutzte ECB-Leitung an den Controller geführt und dort an die Pins 2 und 14 des 74LS155 gelegt. Die beiden 74LS30 beim 16poligen Stecker, die vorher das Signal /37EX und die Drucker--Freigabe über Port FDH geliefert haben, können entfernt werden. Die Verwendung des Signals /FLO erübrigt auch die Zuführung der Signale, die in obiger Tabelle nicht angeführt sind und bei Verwendung des EXP1 im Genie oder TRS80 zusätzlich angeschlossen werden mußten.

RAM-Speicher satt

Grundsätzlich besteht der gesamte physikalische Adreßraum vom 1MB aus RAM-Speicher. Nur dort, wo irgendwelche memory mapped I/O-Baugruppen eingeblendet werden, ist der RAM-Speicher nicht erreichbar. Durch Abschalten der mm I/O-Baugruppen wird dort dann automatisch der entsprechende RAM-Bereich eingeblendet.

Im Bereich des ROMs (00000-036FF) und der Tastatur (03800-038FF), wenn diese im 64x16-Textmodus erreichbar ist, kann der RAM-Speicher auch beschrieben werden, ohne daß ROM bzw. Tastatur ausgeschaltet werden müssen. Nur zum Lesen dieser RAM-Bereiche müssen diese Einheiten ausgblendet werden.

Memory Map



Port 38H					
	D0 :	0	1	1	x
	D1 :	0	0	0	x
	D2 :	0	0	0	1
	D4 :	0	0	1	1
	D6 :	0	0	0	1
	D7 :	0	0	1	x

Ein entscheidender Nachteil des Industrie-Standards sei zum Schluß noch erwähnt. Um das vom Genie her bekannte Flickern bei Ausgaben auf den Bildschirm zu verhindern, lassen modernere Hercules-Karten einen Zugriff auf das Video-RAM nur zu, wenn der Elektronenstrahl vom rechten zum linken Bildrand zurückläuft und dabei dunkel gesteuert ist. Die CPU wird während der Zeit bis zum Strahlrücklauf mit einem Wait (/IO_CH_RDY, Pin A10 des PC-Slots) ruhiggehalten. Offensichtlich ist das Wait-Timing bei HD64180 anders als beim 80286 im PC-AT, denn die Hercules macht im Genie nur bis 5MHz Systemtakt mit; bei höheren Takten erscheint nur noch Müll auf dem Fernseher.

Es gibt aber auch Hercules-Karten, die sich wie ein normales Genie--Video-Interface ohne Rücksicht auf Flickern beschreiben lassen. Eventuell läßt sich damit dann der für das CPU-Board als machbar zu erwartende Takt von 9,216MHz fahren. Wahrscheinlich muß dann aber ein Wait aus dem Freigabesignal /HA18 für die Speicher der Hercules-Karte erzeugt werden, das den Zugriff der CPU um 1 oder 2 Takte verlängert. Da ich eine solche Karte aber nicht habe, will ich darüber auch nicht weiter nachdenken.

Das System verfügt noch nicht über eine Reset-Schaltung. Diese kann im

einfachsten Fall durch Kurzschließen von /RESET (ECB-Bus, 31c) mit GND über einem Taster bereitgestellt werden. Da aber während der Zeit, in der der Taster gedrückt ist, kein Refresh der dynamischen RAMs erfolgt, gehen bei einem Reset Daten verloren. Eine einfache Schaltung mit einem 555-Timer und einem Taster erzeugt sowohl beim Einschalten als auch beim Betätigen des Tasters ein getimtes /RESET-Signal, das ein längeres Ausbleiben des Refresh verhindert (Abb.5).

Freie Verdrahtung auf der Lötseite

Signal	IC,Typ,Pin —— IC,Typ,Pin ...				
D0	18,273,8	10,27128,11			
D1	18,273,7	10,27128,12			
D2	18,273,4	PC-Slot,A7	10,27128,13		
D3	18,273,13	10,27128,15			
D4	18,273,8	PC-Slot,A5	10,27128.16		
D5	18,273,14	10,27128,17			
D6	18,273,17	10,27128,18			
D7	18,273,18	10,27128,19			
D0	17,64180,34	30,125,5			
D1	17,64180,35	30,125,2			
D2	17,64180,36	38,245,4			
D3	17,64180,37	38,245,5			
D4	17,64180,38	38,245,3			
D5	17,64180,39	38,245,9			
D6	17,64180,40	30,125,11			
D7	17,64180,41	31,174,13			
/HALT	17,64180,56	VG64-Leiste,25c			
/REF	17,64180,57	VG64-Leiste,28a			
/IOE	1,PAL1,1	17,64180,58	19,32,10	PC-Slot,A10	33,245,7
	37,PAL2,13				
/ME	16,32,4	17,64180,59	19,32,13	33,245,2	37,PAL2,14
/LIR	17,64180,61	37,PAL2,23			
/WR	17,64180,62	19,32,12	33,245,9	37,PAL2,3	
/RD	1,PAL1,2	17,64180,63	19,32,1	33,245,8	37,PAL2,1
CLK	17,64180,64	VG64-Leiste,29c			
/WAIT	17,64180,4	VG64-Leiste,10a	PC-Slot,A10		
/BUSAK	17,64180,5	37,PAL2,2			
/BUSRQ	17,64180,6	VG64-Leiste,11a			
/RESET	18,273,1	17,64180,7	15,04,5	31,174,1	VG64-Leiste,31c
/NMI	17,64180,8	VG64-Leiste,20c			
/INT0	17,64180,9	VG64-Leiste,21c			
/ST	17,64180,12	35,245,2			
A0	17,64180,13	10,27128,10	25,PAL3,7	32,157,2	38,245,2
A1	17,64180,14	10,27128,9	32,157,14/3	36,245,6	
A2	17,64180,15	10,27128,8	32,157,13/5	36,245,3	
A3	17,64180,16	10,27128,7	32,157,11/6	36,245,4	
A4	17,64180,17	10,27128,6	25,PAL3,16	32,157,10	36,245,5
A5	17,64180,18	10,27128,5	25,PAL3,6	36,245,7	
A6	17,64180,19	10,27128,4	25,PAL3,5	36,245,8	
A7	17,64180,20	10,27128,3	25,PAL3,4	36,245,9	
A8	1,PAL1,6	10,27128,24	17,64180,21	25,PAL3,3	35,245,3
A9	1,PAL1,5	10,27128,25	17,64180,22	25,PAL3,2	33,245,4
A10	1,PAL1,4	10,27128,21	17,64180,23	25,PAL3,1	35,245,5
A11	1,PAL1,3	10,27128,23	17,64180,24	35,245,7	
A12	17,64180,25	10,27128,2	20,86,5	28,30,12	33,245,5
A13	17,64180,26	10,27128,27	20,86,9	28,30,11	33,245,6
A14	17,64180,27	20,86,2	35,245,6		
A15	17,64180,28	20,86,12	33,245,3		
A16	17,64180,29	12,157,5	27,02,11	35,245,4	
A17	17,64180,30	12,157,11	27,02,12	35,245,8	
A18	17,64180,31	24,00,5	21,08,13/2	35,245,9	
A18'	21,08,4	11,157,14			
A19'	21,08,5	11,157,2			
B81	18,273,6	25,PAL3,9	29,125,4	16,32,5	15,04,9
/B81	15,04,8	20,86,4	29,125,10		
B87	18,273,16	25,PAL3,11			
Q0	30,125,5	16,32,10			
Q1	30,125,2	21,08,12			
X	16,32,8	21,08,1			
Y	24,00,6	15,32,9			
/M016	26,20,8	1,PAL1,14			
/M1216	28,30,8	1,PAL1,22			
/MFxxxx	26,20,6	1,PAL1,18			
/OUTB8	37,PAL2,19	9,273,11			
/OUTF4	37,PAL2,18	31,174,9			
/INF4	37,PAL2,17	30,125,10			
/ADR	37,PAL2,15	1,PAL1,17			
/ROM	1,PAL1,20	10,27128,20	10,27128,22		
/FLO	1,PAL1,16	37,PAL2,19	VG64-Leiste,z.B.23c		
/KB	1,PAL1,19	37,PAL2,20	VG64-Leiste,z.B.23a		
/RAM	1,PAL1,22	16,32,1			
/HA18	1,PAL1,21	PC-Slot,A13			
HA15	20,86,11	PC-Slot,A16			
HA14	20,86,3	PC-Slot,A17			
HA11	29,125,6	PC-Slot,A20			
RESET	15,04,6	PC-Slot,B2			
SEL	16,32,6	32,157,1			
+5V	22,1489,14	20,86,14	17,64180,32	10,27128,28	4,RAM,9
+5V	1,PAL1,24	2,RAM,9			
+5V	25,PAL3,20	PC-Slot,A12/A14/A15/B29			
GND	10,27128,14	20,86,7	26,20,7		
GND	17,64180,33	15,04,7	V24-Stecker,9/10		
+12V	23,1488,14	VG64-Leiste,13a			
-12V	23,1488,1	VG-Leiste,14a			

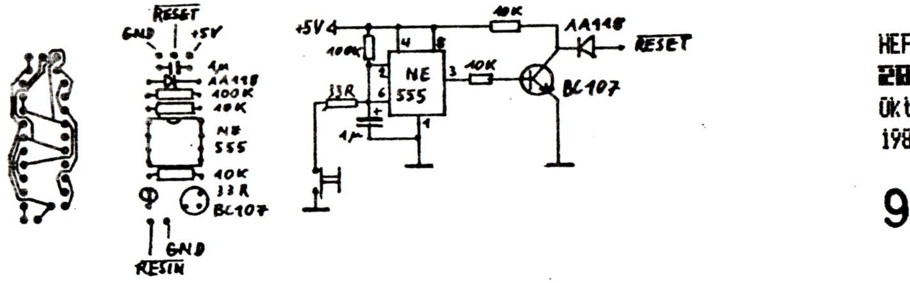


Abb.5: Getimeter /RESET beim Einschalten und Drücken des Tasters

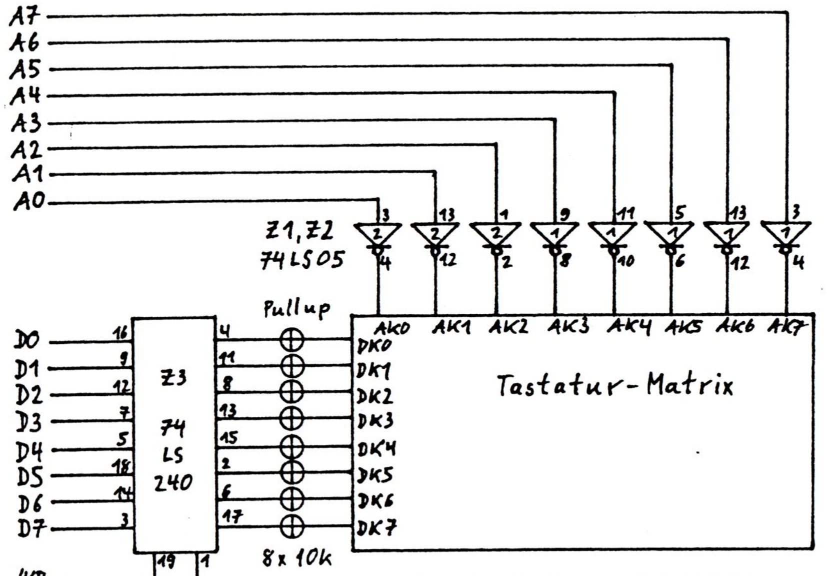
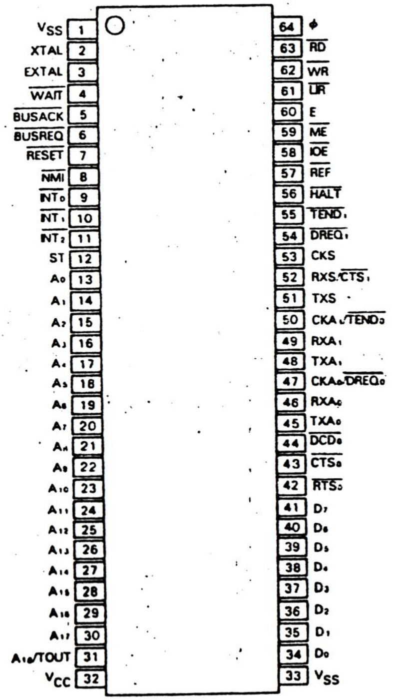
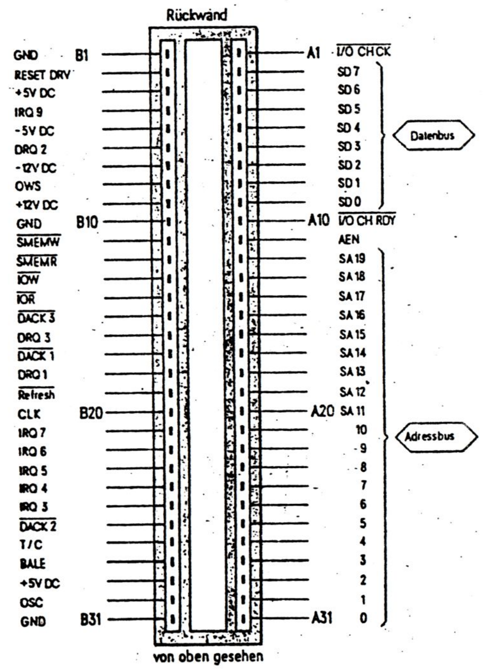


Abb.6: Tastatur-Interface



Pinout der HD64180-CPU



Pinout des PC-Slots


```
;Initialisierung der Hercules-Karte und der HD64180-CPU
;im Sonder-ROM-Bereich ab 3000H
;
;Diese Routinen werden aus dem ROM heraus angesprungen
;und übergeben zum Schluß wieder an die TRS-80 Boot-
;Routine. Dafür muß das ROM ab 0000H folgendermaßen
;geändert werden:
;    vorher                          nachher
;0000   DI                          0000    DI
;0001   XOR     A                   0001    XOR     A
;0002   JP      0674H               0002    JP      3050H
;
;Die Initialisierung schließt mit JP  0674H ab.
;
;
attrib  EQU     07h         ;D7=0 nicht blinkend bzw
                            ;   normaler Hintergrund
                            ;D6-D4=0,D2-D0=1 normale
                            ;   Darstellung
mode    EQU     0           ;D7=0   HRG Page 0
                            ;D5=1   Blinken erlaubt
                            ;D3=1   Video-Signal aktiv
                            ;D0=0   Text-Betriebsart
config  EQU     3           ;D0=1   HRG erlaubt
                            ;D1=1   HRG-Page2 enabled
;
;
        ORG     3000h               ;Anfang Sonder-ROM
;
;Initialisierungdaten für den CRTC6845
;64*16-TRS80-Standard-Einstellung
tab     DEFB    6eh,40h,50h,0ah,14h,06h,10h,12h
        DEFB    02h,0eh,29h,09h,04h,00h,00h,00h
;80*25-Modus
        DEFB    6eh,50h,58h,0ch,1fh,02h,19h,1bh
        DEFB    02h,09h,29h,09h,00h,00h,00h,00h
;64*24-Modus
        DEFB    6eh,40h,50h,0ah,1bh,09h,18h,19h
        DEFB    02h,0ah,29h,09h,02h,00h,00h,00h
;64*32 interlaced
        DEFB    6eh,40h,50h,0ah,13h,02h,10h,11h
        DEFB    03h,0fh,2fh,0fh,00h,00h,00h,00h
;92*22-Modus
        DEFB    70h,5ch,61h,08h,1dh,08h,16h,19h
        DEFB    02h,0bh,29h,09h,00h,00h,00h,00h
;
        LD      SP,8000h
        LD      A,3                 ;obere 256K einstellen
        OUT     (0f4h),A
        LD      A,81h               ;Attribut-Speicher mit
        OUT     (0b8h),A            ;Standard-Wert füllen
        LD      HL,3800h            ;Video-RAM ab 3800h
        LD      A,attrib
        LD      (HL),A
        LD      DE,3801h
        LD      BC,07ffh            ;2k Speicher
        PUSH    HL
        PUSH    DE
        PUSH    BC
        LDIR
;
        POP     BC
        POP     DE
        POP     HL
        LD      A,1                 ;Umschalten auf Text-
```

```
        OUT     (0b8h),A          ;-Speicher
        LD      A,' '             ;mit Blanks füllen
        LD      (HL),A
        LDIR
;
        LD      HL,0              ;ROM-Inhalt ins RAM
        LD      DE,0              ;kopieren
        LD      BC,3700h
        LDIR
;
;HD64180 initialisieren
        LD      A,10h             ;WAITs: Memory 0, I/O 2
        DB      0edh,39h,32h      ;OUT0 (DCNTL),A
        LD      A,83h             ;Refresh-Control
        DB      0edh,39h,36h      ;optimieren
        LD      A,44h             ;Common 0-32K, Bank 32-
        DB      0edh,39h,3ah      ;64k einstellen
        XOR     A                 ;Common1 als Bank benutzen
        DB      0edh,39h,38h      ;Bank bei 08000-0FFFF
;
;CRTC6845 für 64*16 Zeichen initialisieren
        LD      HL,tab            ;HL => CRTC-Parameter
        CALL    setcrtc           ;raus an CRTC
;
;Video_Signal einschalten
        LD      A,08h             ;Video-Signal aktiv
        OUT     (0b8h),A          ;Video bei 3c00-3fff
                                  ;Text-Modus eingesch.
                                  ;Blinken gesperrt
        JP      0674h             ;und weiter im TRS80-
                                  ;Boot-Gebaren

;CRTC-6845 initialisieren (HL => Parameterliste)
setcrtc XOR     A                 ;für Video-Mode-Umschal-
        OUT     (0bfh),A          ;tung nötig
        LD      B,16              ;16 Bytes ausgeben
        LD      C,0b5h            ;CRTC-Daten-Port
        XOR     A                 ;CRTC-Register-Nr ab 0
crtclp  OUT     (0b4h),A          ;1. Register
        INC     A                 ;A=> nächstes Register
        OUTI                      ;raus das Zeugs
        JR      NZ,crtclp
        LD      A,config          ;HRG voll freigeben
        OUT     (0bfh),A
        RET

        END
```


Assembler-Listing für PAL1

```
PAL20L8
Helmut Bernhardt            22.01.89
Erzeugen der mm-Freigabesignale

ioe rd a11 a10 a9 a8 nc b80 b81 b82 b84 gnd
b86 m016 nc flo adr mfxxxx kb rom ha18 ram m1216 vcc

/ha18 = /b86 */b81 */b80 */m1216 *a11 *a10
      + /b86 */b81 * b80 */m1216 *a11
      + /b86 * b81 */mfxxxx

/kb   = /b86 */b80 */rd */m1216 *a11 */a10 */a9 */a8

/flo  = /b84 */m1216 */a11 *a10 *a9 *a8 */adr
      +  b84 */ioe */adr

/rom  = /b82 */m016 *m1216 */rd
      + /b82 */m1216 */a11 */a10 */rd
      + /b82 */m1216 */a11 * a10 */a9 */rd
      + /b82 */m1216 */a11 * a10 * a9 */a8 */rd

/ram  = ioe *ha18 *kb *flo *rom
```

Assembler-Listing für PAL2

```
PAL20L8
Helmut Bernhardt            24.01.89
I/O-Decoder und Datentreiber-Steuerung

rd busak wr a0 a2 a3 a4 a1 a5 a6 a7 gnd
ioe me adr herc inf4 outf4 outb8 flo kb rein lir vcc

/inf4  = /ioe */rd *a7 *a6 *a5 *a4 */a3 *a2 */a1 */a0

/outf4 = /ioe */wr *a7 *a6 *a5 *a4 */a3 *a2 */a1 */a0

/outb8 = /ioe */wr *a7 */a6 *a5 *a4 *a3 */a2 */a1 */a0

/adr   = a7 * a6 * a5 * /a4

/herc  = /ioe *a7 */a6 *a5 *a4

/rein  = busak */ioe */lir
       + /ioe */rd *inf4 *herc
       + /busak *rd
       + /flo */rd
       + /kb */rd
```

99

Assembler-Listing für PAL3

```
PAL16L8
Helmut Bernhardt                    23.01.89
Adreßübersetzung für die Hercules-Karte

a10 a9 a8 a7 a6 a5 a0 me b81 gnd
b87 ha0 ha5 ha6 ha7 a4 ha8 ha9 ha10 vcc

/ha0   = /me * /b81 * /b87
       + /me *  b81 * /a0
       +  me * /a0

/ha5   = /me * /b81 * /a4
       + /me *  b81 * /a5
       +  me * /a5

/ha6   = /me * /b81 * /a5
       + /me *  b81 * /a6
       +  me * /a6

/ha7   = /me * /b81 * /a6
       + /me *  b81 * /a7
       +  me * /a7

/ha8   = /me * /b81 * /a7
       + /me *  b81 * /a8

/ha9   = /me * /b81 * /a8
       + /me *  b81 * /a9

/ha10  = /me * /b81 * /a9
       + /me *  b81 * /a10
```

Pinbelegung des V24-Steckers CN2

Signal	Pin	Pin	Signal
/CTS1	1	2	TXS1
/RTS0	3	4	RXA0
/CTS0	5	6	RXA1
TXA0	7	8	TXA1
GND	9	10	GND

Bei welchem VIDEO-GENIE- Besitzer läuft eine RB V.24-Schnittstelle.
Ich bin sehr an Erfahrungsaustausch interessiert. Meine
Schnittstelle läuft nämlich nicht.

Hans-Günther Hartmann, Adresse und Teflonnummer siehe Info!

100

Betr.: Ihr Beitrag im c't-Club 7/89

Ich möchte zwar nicht dem Club beitreten, habe aber für den Tandy Modell 1 jede Menge an Software.

Den Tandy biete ich zum Verkauf an.

1 Tandy Modell 1 mit Expansion-Interface alles in einem Gehäuse untergebracht.

2 serielle Schnittstellen eingebaut
1 Tandy Monitor
1 Tandy Drucker 100 ← Line Printer 100

Zeitschriften aus den USA "MICRO 80"

Alles zusammen für 400,-DM

Ich würde mich freuen von Euch zu hören.

Mit freundlichen Grüßen P. S.

Peter Schopen
Lochnerstr. 25
4030 Ratingen

Verkaufe

1 x TRS-80 Model I, Level II
mit 10er Block auf der Tastatur und HRG,
eingebaut in ein Stahlblechgehäuse
2 x Laufwerke 40 Spuren, DD
1 x entspiegelter Monitor

dazu diverse Software und Handbücher

Preis: VS

W. Camphausen
Florastr. 14
5600 Wuppertal 1
Tel.: 0202-750399

Suche folgende Zeitungsartikel leihweise oder als Kopie:

Creative Computing:	Juli 79, S.34-37
Dr.Dobbs:	22:10-13; 48:34-37; Feb.77, S.30-43; Feb.80, S.18-25
Interface Age:	Aug.78, S.152-156; Dez.78, S.130-135; Aug.80, S.88-90; Sept.80, S.80-86,130-131; Okt.80, S.106-107, 137-141
Microsystems:	Jul/Aug.80, S.32-35

Kosten werden erstattet. Wer kann helfen?

Norbert Giese A.-Stifter-Weg 10 7034 Gärtringen 07034/29980

Suche für SHARP MZ-80B:

Informationen über Bankumschaltung, speziell Funktion von PAL (IC 2 im Schaltbild).

Schaltbild oder Idee für einfachen EPROMMER, der die nötigen Programmierspannungen aus +5 Volt generiert. Gerät soll nur an Parallelport angeschlossen werden; ohne externes Netzteil.

Norbert Giese A.-Stifter-Weg 10 7034 Gärtringen 07034/29980

TRS 80 - Modell 1 mit HRG1B und 64 KB Ram,3 BASF 5.25 " 1D DD, Exp.Interface,256 KB - Banker Platine,CP/M - Platine,Netzteile, Monitor Sanjo bernsteinfarbig.Preis VS.

Genie 1 EG3003 16 KB mit Centronicsinterface,200.- DM. Zero Eprommer mit Textolsockel für Genie jedoch Software auf Diskette,80.- DM.

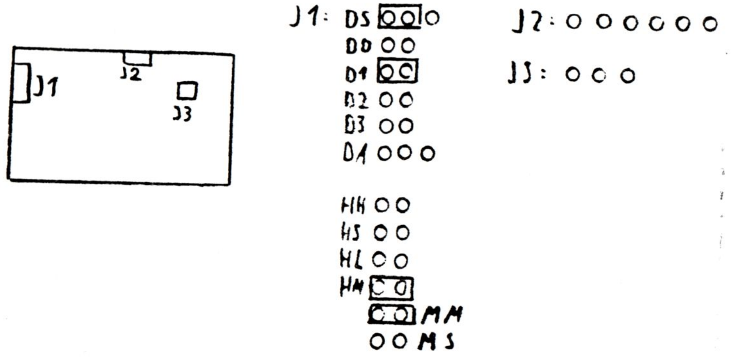
Reiner Stober Tel.:05153/1564 ab 20.00Uhr

Außerdem möchte ich gleich mal eine Anfrage im Clubinfo starten:
Wie schon erwähnt haben meine beiden Laufwerke nur eine Kapazität von 180K. Ich möchte nun wissen, was ich an Hard- und Software ändern muß, um die Kapazität zu erhöhen.
Bei dem 3′-Floppy ist wahrscheinlich nichts mehr zu machen, dafür kann man bestimmt aus dem 5 1/4′-Floppy noch mehr herausholen. Laut Beschreibung handelt es scih um ein Original-Commodore-Amiga-Laufwerk, das ein meinen Bus angepaßt wurde. Zum Umschalten der beiden Köpfe wurde einfach ein Schalter vorne am Gehäuse angebracht.
Hinweise und Tips, die zur Erhöhung der Speicherkapazität beitragen, werden dankbar angenommen.

Bis zum nächsten Mal

Jürgen Hennen

Skizze meines 5 1/4"-Laufwerks



GENIE III : Einzelteile zu verkaufen
CPU-Board, FDC-Board, Video-Board : je 50.-DM Netzteil : 60.-DM Gehäuse mit Einbaurahmen und Buskarte gibt's umsonst dazu Video-Teil (60Hz-Monitor) und Floppy Drives sind defekt Tastatur wollte ich nicht unbedingt loswerden: darüber läßt sich aber reden. Helmut Bernhardt 0431/241907

Ein guter Meter TRS-80-Literatur zu verschenken
Enthalten sind die goldenen Jahrgänge von "80 Micro", die "Blah, blah, blah... and other Mysteries" - Bücher und noch so manche anderen Schmankerl. Das alles verschenkt Christof Ueberschaar, wenn die Papierberge in Wilhelmshaven oder Kiel abgeholt werden oder die Portokosten übernommen werden. Da Christof momentan nicht weiß, wo es ihn in den nächsten Monaten hinverschlägt, und ich nicht weiß, wann das nächste Info erscheint, stelle ich mich als Kontaktadresse zur Verfügung. Helmut Bernhardt 0431/241907

c't 12/83 bis 3/89 zu verkaufen
Alles, was c't über 8-Bit-Computer gebracht hat, ist hier zu finden. Bernd Bauer (Tel. 0431/673819. ab 19:Uhr) will das nicht noch mal alles bauen und sich daher von dem Meter Literatur trennen. Wer Interesse hat, kann sich mit ihm über den Preis streiten.

Suchmeldung,

suche Double Density Controler für Model 1. Wer kann helfen. Anleitung zum Eigenbau zwecklos, da Niete in Hardware

Egbert Schröder
Joachimstr. 18
4270 Dorsten 1
Tel.: 02362/75840

Mikrocomputer-Betriebssysteme - eine Einführung

von Gerald Schröder

Dieser Artikel soll der Auftakt zu einer kleinen Serie über CP/M sein. Allerdings werden auch Newdos sowie (am Rande) MS-DOS erwähnt. Hier soll es erst einmal um die Grundlagen gehen, die für alle Betriebssysteme gelten, die auf kleineren Rechnern zum Einsatz kommen. Bei Großrechnern sind ganz andere Anforderungen wichtig.

Ich habe mich bemüht, einen lockeren Ton anzuschlagen. Damit sollen vor allem die Leute angesprochen werden, die sich noch keine Gedanken über dieses Thema gemacht haben. Weiterhin muß ich zugeben, daß ich mich auf keinerlei Literatur berufen kann, weil sich die mir bekannten Bücher nur mit Großrechner-Betriebssystemen befassen. Alle Ausführungen sind also meine private Meinung, gewonnen aus 8 Jahren Erfahrung mit Mikrocomputern, die mich zu folgenden Forderungen brachten:

Freiheit für unsere Programme
Gleichheit aller Rechner-Umgebungen
Unabhängigkeit von der Hardware

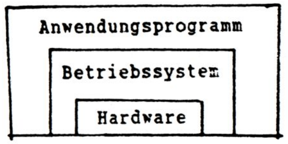
Am Anfang möchte ich Euch die Frage stellen: "Was erwartet Ihr von einem Rechner samt Betriebssystem?"

Meine Antwort: Ich möchte einen Rechner, bei dem ich mich nicht um jede Kleinigkeit selbst kümmern muß, auf dem ich schnell und einfach Programme herstellen kann, die auch auf anderen Rechnern laufen, bzw. auf dem ich problemlos Programme laufen lassen kann, die andere Leute auf anderen Rechnern geschrieben haben. Ich schätze, so dürften Eure Bedürfnisse auch aussehen; ich sehe von bekannten Ausnahmen ab, die den Rechner in- und auswendig kennenlernen wollen, nur Programme zum Eigengebrauch schreiben und auf fremde Programme ganz verzichten.

Nun sind Computeranlagen meist sehr unterschiedlich zusammengesetzt: Der eine hat das Geld für vier 1,2-Mb-Laufwerke nebst Festplatte, arbeitet grundsätzlich nur mit Laserdrucker, läßt eine Komfort-Tastatur eigens für seine Handgröße entwerfen und verwöhnt seine Augen mit einem hochauflösenden Grafik-Monitor. Der andere dagegen rödelt mit einem alten 100-Kb-Laufwerk, kann sich gerade die gehäuselose Tastatur von Bühler für 5,20 DM leisten, muß sich den Fernseher zum Monitor umbauen und hat einem ausgemusterten Fernschreiber das Drucken beigebracht. Beide möchten dieselben Programme benutzen und vielleicht sogar Programme austauschen. (Die echten Computer-Freaks kennen eben keine Skrupel bei der Überwindung von Klassengrenzen; Marx wäre schockiert.)

Da kommt unser Betriebssystem ins Spiel, mag es nun TRaShDos, Ce-Pe-eM, DoMeSDOS, oder TU-nix heißen. Man nehme ein beliebiges dieser Betriebssysteme, passe es so an, daß es einmal auf dem Turbo-GSI-Injection und einmal auf der Schrottmühle läuft, und schon können die beiden munter Programme tauschen, die mit etwas Glück auf beiden Computern laufen.

Wo ist der Trick? Ganz einfach: Jedes (Anwendungs-)Programm, das einer unser Freaks geschrieben hat, findet auf jedem der Rechner die gleiche Umgebung, nämlich das Betriebssystem. Das Programm merkt nichts davon, welcher Computer sich dahinter verbirgt. Bildlich sieht das so aus:



Das Betriebssystem "kennt" die Hardware und das Anwendungsprogramm "kennt" das Betriebssystem. Aber das Anwendungsprogramm weiß nichts über die Hardware. Das Programm hat nun meistens einige Bedürfnisse, die das Betriebssystem befriedigen muß: das Lesen und Schreiben von Texten (o.ä.) von/auf eine Diskette/Festplatte; das Einlesen von Tastendrücken des Users (einer unserer Freaks); die Ausgabe von Reaktionen darauf (meist auf einen mehr oder minder teuren Bildschirm); die Beschmutzung von Papier per Drucker; usw. Generell handelt es sich hier um Ein- und Ausgaben des Programms von/auf irgendwelche/n Geräte/n.

Das alles erledigt das Betriebssystem für das Anwenderprogramm. Das Programm sagt also (stark vereinfacht): "Ich hab' hier einen Text und der soll irgendwo so gespeichert werden, daß ich ihn später mal zurückholen kann; das Gerät soll A: sein und als Name wäre mir FILENAME.EXT recht."

Dies geschieht durch einen oder mehrere Unterprogrammaufrufe, wobei die Unterprogramme Teile des Betriebssystems sind, die von dem Programm aufgerufen werden. Nun kann das Betriebssystem nachsehen, was denn da so an Massenspeichern vorhanden ist, z.B. Diskettenlaufwerken, und welcher Massenspeicher die Bezeichnung "A:" trägt. Sollte es keinen davon geben, sagt das BS: "Nee, tut mir leid, A: haben wir heute nicht. Geht nicht was anderes?" Dann muß sich das Programm was anderes ausdenken.

Normalerweiser haben wir aber einen Massenspeicher A:; dabei ist egal, was für ein Typ Massenspeicher dies ist, also Diskette oder Festplatte oder umgebauter CD-Player o.ä. Das Betriebssystem sorgt nun dafür, daß dieser Text auf dem Massenspeicher so abgelegt wird, daß es ihn später unter dem Namen FILENAME.EXT wiederfinden kann. Je nach Betriebssystem kann/muß dieser Name und die Geräte-Bezeichnung übrigens verschieden aussehen. Später sagt das Programm dann mal: "Ey, ich hätte gern einen Text zurück, der auf dem Gerät A: steht und den Namen FILENAME.EXT trägt." In einem Prozent der Fälle bekommt das Programm den Original-Text fehlerlos zurück; streng nach Murphy tritt bei den restlichen 99% irgendein Fehler auf (Diskette im Kaffee gebadet, Bits gerade auf Reisen oder anderweitig unabkömmlich usw.).

Wie wir sehen, ist es dem Programm vollkommen schnurzpiepe, was für Gerät sich nun eigentlich hinter diesem ominösen "A:" verbirgt, solange die Daten da nur über das Aussschalten des Rechners hinaus verfügbar bleiben. Theoretisch könnte das beim Schreiben ein Drucker sein, der massenhaft Papier ausspuckt, und beim Lesen eine Tastatur, auf der eine unterbezahlte junge Dame wieder eintippt, was vorher auf das Papier gedruckt wurde. Da dies nicht so effizient ist, wurde die Idee trotz des drängenden Arbeitslosenproblems wieder fallengelassen.

Ähnlich sehen die Sachen nun für Tastatur, Bildschirm, Drucker usw. aus. Nur wird diesmal meistens nur ein Zeichen übergeben, was dann beim Bildschirm so ausieht: "Ich hab' hier ein Zeichen 'a', das der blöde User da irgendwie mitkriegen soll; schmier das mal auf 'nen Bildschirm oder so." Dabei ist es dem Programm egal, ob das Betriebssystem das Zeichen wirklich auf einen Bildschirm malt, oder auf einem Drucker ausgibt (weil der Bildschirm gerade zum 42.ten Mal kaputt ist), oder gar einen Text auf einer Diskette daraus baut (weil der User gerade auf'm Klo Geschäfte abwickelt und zu gegebener Zeit nachsehen möchte, was denn so in den letzten Stunden schiefgelaufen ist).

Für den Drucker sieht die Sache genauso aus, nur weiß das BS jetzt, daß das auszugebende 'a' etwas dauerhafter zur Verfügung stehen soll, z.B. zum Ablegen des verschmutzten Papiers in einen Briefumschlag zwecks Bereicherung der Post. Trotzdem kann sich das BS dafür entscheiden, dieses 'a' auf einem Bildschirm oder gar nicht auszugeben, wenn es das für sinnvoll hält

bzw. von dem User vorher diesen Befehl erhalten hat (der User meinte natürlich was ganz anderes).

Die entsprechende Frage für die Tastatur lautet: "Hat der User inzwischen mit seinen Fettfingern eine unschuldige Taste angefaßt?", worauf das BS antwortet: "Nee, der Junge braucht mal wieder 'ne Gedenkminute oder -stunde." oder "Ja, das arme 'a' mußte dran glauben. Hilft Dir auch nicht weiter, was?" Dabei kümmert sich unser Programm nicht darum, ob der User wirklich auf der Tastatur rumhämmert oder ob das BS sich diese Tastendrücke einfach ausdenkt.

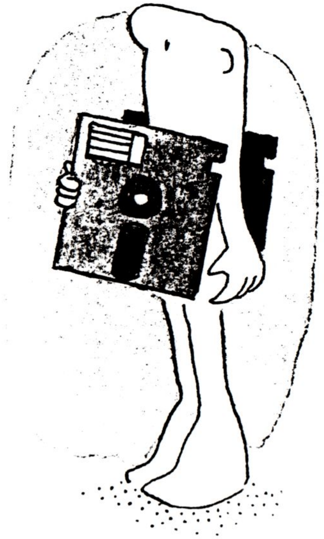
Niedlich ist es z.B., wenn der Mensch seine ganzen Tastendrücke in einem Text auf der Diskette ablegt und das BS diese nun Stück für Stück wieder rausholt und dem Programm gegenüber so tut, als hätte der User gerade seine Grabbeln an der Tastatur gehabt (während er in Wirklichkeit ganz was anderes begrabbelt). Das nennt sich "Batch-Betrieb" und der Text auf der Diskette heißt "Batch-Datei"; wenn Ihr jetzt im Wörterbuch über "Batch = Stapel" stolpert, denkt daran, daß in grauer Vorzeit mal Lochkartenstapel (statt Dateien) benutzt wurden, die sich das BS dann Karte für Karte reingezogen hat.

Wie Ihr seht, weiß das Anwenderprogramm überhaupt nicht, was denn eigentlich passiert, nachdem es dem BS die Aufgabe übertragen hat, etwas zu tun. Wenn es Pech hat, macht das BS sich einen faulen Lenz und das Programm merkt noch nicht einmal was. Das schönste Beispiel ist die Umlenkung der Drucker-Ausgabe auf ein Null-Device (das ist ein Gerät, das es nicht gibt, also eine Fata Morgana, eben nichts - null). Da vergißt das BS einfach alle Zeichen, die es eigentlich ausdrucken sollte - sagt aber dem Anwenderprogramm dauernd: "Alles klar, Kumpel, ich druck' das schon für Dich." Fragt mich nicht, was das soll, fragt unsere IBM-/MS-DOS-Freaks.

Was für einen Vorteil hat es für uns, daß das Betriebssystem alle diese Dinge (und noch mehr) für uns erledigt? Nun, wir (als Programmierer in irgendeiner Form) brauchen uns nicht darum zu kümmern, wie der Laserdrukker/Fernschreiber angesprochen werden muß, müssen nicht dauernd die Tastatur überwachen (sofern wir überhaupt wissen, wie das geht), kennen nicht jeden Bildschirm / jede Grafikkarte / jedes Terminal, sondern überlassen alle diese Sorgen dem Betriebssystem bzw. dem armen Schwein, das ein BS schreiben oder an den jeweiligen Rechner anpassen muß.

Ansonsten müßten wir jeden Fall, der irgendwann auf irgendeinem Rechner auftreten könnte, einkalkulieren und unser Programm darauf einrichten. Wir müßten jedes Gerät (Kassetten-/Disketten-/Festplattenlaufwerk, Bildschirm, Tastatur, Drucker...) kennen, wissen, wie es auf jedem Rechner angesprochen wird und was jeweils zu beachten ist usw. Das ist sogar schon bei der relativ kleinen Klasse der TRS-80/Genie-Rechner unmöglich. Wer's nicht glaubt, versuche mal ein Programm ohne jeglichen Newdos-Aufruf zu schreiben, das sowohl auf einem Original-TRS-80 als auch auf Arnulfs Genie-IIIs-Turbo läuft.

Fazit: Ohne Betriebssystem wäre unser Programmierer-Leben weitaus schwerer und wir würden alle ein Einzelgänger-Dasein fristen. Hier habe ich nur etwas dazu gesagt, wie Betriebssysteme uns helfen, daß unsere Programme sich auch auf vollkommen fremden Rechnern zurechtfinden. Mit Hilfe des Betriebssystems können sie auf jedem Rechner feststellen, welche Tasten der User gerade gedrückt hat, sie können einen angeschlossenen Drucker drucken lassen usw. Über andere Probleme werde ich mich in einem weiteren Artikel auslassen. Also: Keep listening to this radio station!



Ein neuer Diskothekar meldet sich zu Wort

Wie Ihr an anderer Stelle gelesen habt bin ich der neue Diskothekar für Neudos, HDOS, GDOS, etc.. Ich hoffe, daß ich rege Post von Euch bekomme. Am liebsten wäre es mir wenn Ihr mir Disketten 80 Spur DS/DD schicken würdet, da ich dieses Format benutze. Ihr solltet aber auf jeden Fall, bei Programmanforderungen, die Scheiben mit euerem Rechner formatieren da es sonst später Probleme bei Euch geben wird, vor allem bei 40 Tracklaufwerken. So jetzt will ich Euch noch verraten in welchen Formaten Ihr mir sie senden könnt bzw. in denen ich sie beschreibe.

80 Track DS/DD
TI=EHK,TD=G,SP=80,SEK=36,SWZ=3,EIB=6,SBIV=48,AEIV=6

40 Track DS/SD
TI=EKL,TD=G,SP=40,SEK=36,SWZ=3,EIB=6,SBIV=24,AEIV=6

40 Track SS/SD
TI=A,TD=A,SP=40,SEK=10,SWZ=3,EIB=2,SBIV=20,AEIV=2

Bei den Anforderungen von Kopien kann es etwas dauern denn ich habe noch ein paar Probleme mit der Dateiverwaltung. Ihr solltet also nicht verzweifeln wenn Ihr nicht gleich am nächsten Tag Post von mir bekommt.
Jetzt noch eine Frage, kann mir jemand bei TSCRIPS oder wie das sonst noch heist helfen.
Gehe ich richtig in der Annahme, daß es eine zu kopieren erlaubte Software ist und wie komme ich an ein für mich lauffähiges File und eine Bedienungsanleitung. Ich habe einen Genie IIS und der Drucker läuft vermutlich über einen PORT.

Am Schluß möchte ich mich bei allen bedanken die mir beim Clubtreffen Infomaterial versprochen und zahlreich zugesand haben. Ich hoffe daß ich bis zum nächsten Info alles im Griff haben werde, in Bezug auf unsere Diskothek.

Neues aus der CP/M - Public Domain!

Es tut sich was auf dem Sektor BDOS-Replacements. Es stehen momentan folgende Pakete zur Verfügung (aufgelistet nach dem Datum ihres Erscheinens) :

SUPRDOS2.LBR
eines der ersten Pakete, teilweise fehlerhaft, aber ist zu Ruhm gelangt durch die Tatsache, daß die meisten anderen DOSse auf ihm aufbauen.

Z80DOS10.LBR
Z8D-UTL1.LBR
direkter Nachfolger von SUPRDOS, nicht weniger Bugs.

Z80D20A.LBR
Neuere Beta-Test-Version

DOSPLS25.ARK
Das ist eigentlich ein ganzes Betriebssystem, natürlich ohne BIOS, aber mit erweitertem CCP. Erweitertes Batch-Processing, IF/THEN/ELSE/GOTO usw. eigentlich Shareware, vom Autor jetzt in die PD gegeben, inkl. Quellen. Wenn es jemand ausprobieren möchte, wäre ich für einen Erfahrungsbericht dankbar.

P2DOS23.LBR
Habe ich selbst eine Zeitlang benutzt, gut dokumentiert.

Z80DS231.LBR
Das Ergebnis der Fehlerkorrekturen von Vers. 2.0, den ^P Bug hat wohl niemand entdeckt. Der Autor hat sich große Mühe gegeben, Disk-Zugriffe, speziell Directory-Zugriffe zu beschleunigen.

NOVADOSH.LBR
Das sieht *sehr* interessant aus. Wie Z80DOS, aber zusätzlich ein Suchpfad definierbar. Natürlich auch Uhr, Datestamps, Public Files etc.

Z80DOVL.LBR
Z80D24SR.LBR
Zwei Files aus einem Paket von dreien, wobei mir die dritte noch fehlt. Die neueste Version, stammt aus der ersten Jahreshälfte 1989. ^P geht immer noch nicht. Das wichtigste sind sog. "Harte Disks", Disketten, die nicht gewechselt werden, kann man "hart" machen. Dann muß die Disk Allocation Table nicht jedesmal neu berechnet werden und das Einloggen geht schneller.

Zu allen BDOS-Replacements liegen die Quellen vor. Die Installation ist in der Regel simpel, man muß lediglivh die BDOS-Startadresse eintragen, Optionen setzen und assemblieren, mit Hilfe von SYSGEN und DDT das neue System auf die Systemtracks schaffen.
Ein Tip zum Assembler : alle Pakete sind in Zilog-Mnemonics abgefaßt. Allerdings oft für unterschiedliche Assembler, meist für den SLR. Ich benutze ZAS. Die Änderungen an den Quellen beschränken sich auf zweierlei:

Alle arithmetischen Ausdrücke, die geklammert sind. ZAS verlangt ECKIGE Klammern.

Das Einbinden einer Include-Datei sieht so aus :

```
      .in  ANYCODE.LIB
```

oder

```
      maclib ANYCODE.LIB
```

Die Extension der Datei *muß* .LIB sein!

Programme des Monats

Eins fällt in die Kategorie Print-Utilities und heißt BRADFORD. Unterstützt werden grafikfähige 8/9 Nadeldrucker, befehlskompatibel zu Epson FX80/MX80 oder Star Gemini. Das dürfte für die Mehrzahl der Drucker zutreffen. BRADFORD liest eine Text-Datei, die spezielle Kommandos beinhalten kann, und gibt sie unter Benutzung eigener Fonts aus. Mit Fonteditor. Nachteil : BRADFORD braucht viel TPA, läuft nicht unter Standard-Monte-CP/M. Aber unter meiner 128K-Version davon. Ein Demo-Ausdruck liegt bei.

Das zweite Programm ist ein Editor namens VDE, inzwischen Version 2.66. Er ist kompatibel zu Turbo-Pascal / WordStar. Sehr schnell, mit begrenzten Fähigkeiten als Textverarbeiter. Gut konfigurierbar, Macro-Fähigkeiten. Läßt sich memory-mappen. Nachteil : Er frißt nur Dateien bis Speichergröße.

So, das war's für diesmal. Bestellungen an :

Rüdiger Sörensen
Thomas-Mann-Straße 3a
6500 Mainz 1

Bitte legt eine Diskette ein, bevorzugt eine formatierte. Ich versende nur Monte 80 Tr DSDD SYSTEM. Punkt.

Bis denn, *Rüdiger*

Bradford

The printing program
Copyright 1988, Aaron Contorer
A Concom Enterprises production

Are you tired of the low-quality printing you get from your dot-matrix printer? Then it's time to use Bradford, the printing program.

Bradford will print your ASCII and WordStar files in very high quality, using your choice of fonts (typestyles). Bradford requires no hardware adjustment and costs only $39.95. The version that you are now using is a completely functional copy, not just a crippled "demo version."

Bradford 2.0 new features include:

Up to five fonts PER line.

Much faster on many computers.
Understands more WordStar commands.
Double-height fonts (three shown above).
Automatically recognizes ends of paragraphs.
New three-pass mode on some printers.

Subscripts and superscripts.
Automatic chapter numbering.
Can overstrike two lines.
Headers and footers.
Macros.

Bradford can now do equations: $\mu_0 = 4\pi \cdot 10^{-7}$ Wb A^{-1} m^{-1}

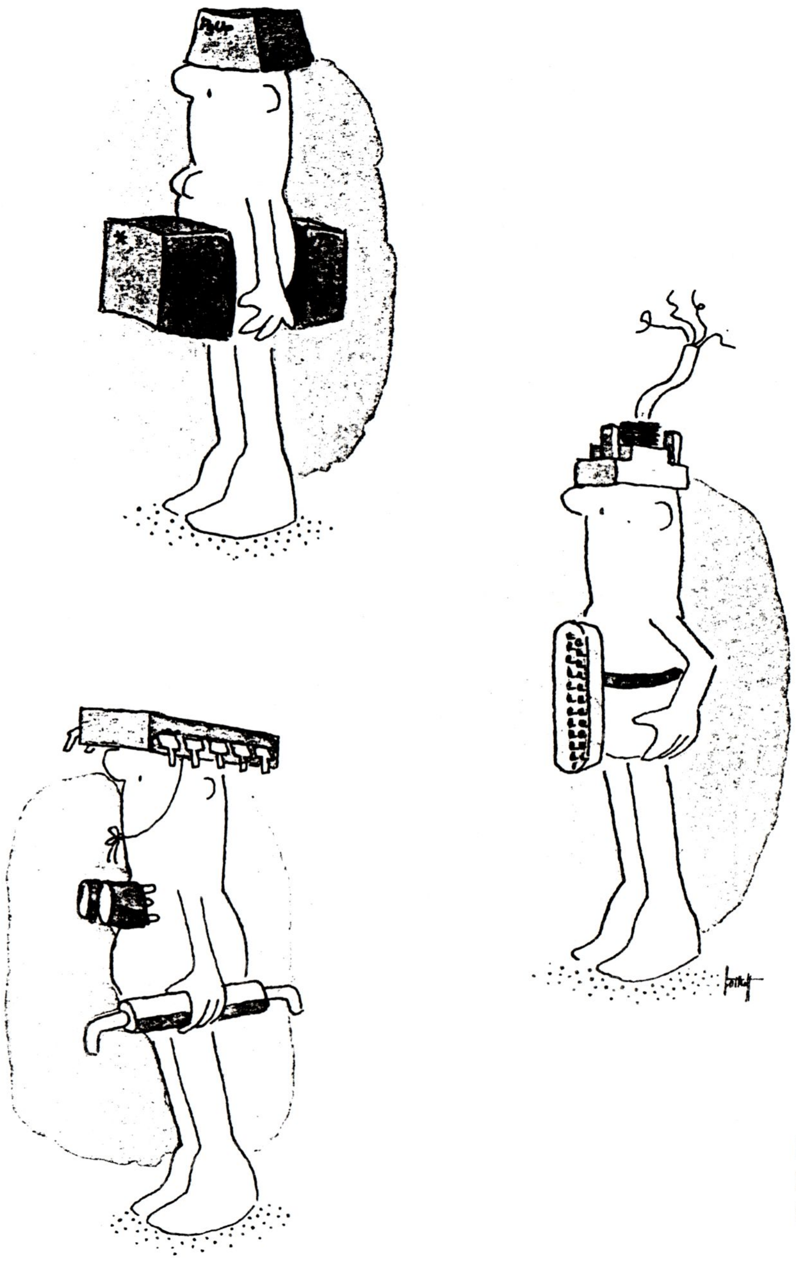
Bradford can now print Greek text: ΙΧΘΙΣ ιχθισ

Try Bradford for a reasonable period of time. A set of basic operating instructions has been provided on the disk for you. If you like what Bradford can do for you, then it's time to pay for a "legitimate" copy. Included with this copy will be a thorough, printed instruction manual (over 50 pages long), and one or more disks containing the latest version of Bradford.

To order your own copy of *Bradford 2*, send a check or money order in U.S. dollars for $39.95 (which includes shipping) to:

Concom Enterprises
Post Office Box 5056
Champaign, IL 61820, USA

(Add $3 if you're outside North America.) Be sure to specify what type of computer you are using (so we will know what type of disks to send) and what type of printer you are using. If you do not wish to purchase your own copy of Bradford 2, you may order the manual alone for $25.



Die Betriebsart Packet-Radio (PR)

(Konzept eines Vortrags, gehalten von Hans-Günther Hartmann, DF7BX, anläßlich des Nordlichter-Treffens des Club 80 vom 20. bis 22.10.89 in Kiel)

A. Zusammenfassung

Die von den Funkamateuren genutzte, neueste Betriebsart **Packet-Radio** ist Gegenstand dieses Vortrags. Hierbei wird zunächst auf die herkömmlichen Betriebsarten im Amateurfunkdienst eingegangen. Weiter soll der Weg erkennbar werden, wie man überhaupt Funkamateur wird, welche Rechte und Pflichten ein Funkamateur hat. Nach einem kurzen "technischen" Streifzug durch die möglichen Betriebsarten im Amateurfunk soll endlich näher auf PR eingegangen werden.
Wer keine Lust hat, etwas über Funkamateure und Gesetze und Verordnungen zu lesen, fängt beim Punkt C 1 an.

B. Der Amateurfunkdienst

1. Was ist Amateurfunk?

Nach der Vollzugsordnung für den Funkdienst (VO Funk) ist der Amateurfunk ein Funkdienst, der von Funkamateuren für die eigene Ausbildung, für den Verkehr der Funkamateure untereinander und für technische Studien wahrgenommen wird. Funkamateure sind ordnungsgemäß ermächtigte Personen, die sich mit der Funktechnik aus rein persönlicher Neigung und nicht aus wirtschaftlichem Interesse befassen.
Für die Verbreitung eignen sich nur Meldungen von geringem Nachrichteninhalt sowie technischem Inhalt. Der Amateurfunk ist kein Ersatz für das Telefon.
Der Amateurfunk existiert weltweit.
Es gibt rund 1 Million Amateurfunk-Stationen auf der Erde, nahezu 57000 befinden sich in der Bundesrepublik.

2. Wie wird man Funkamateur?

Funkamateur kann werden, der
- seinen Wohnsitz in der Bundesrepublik Deutschland einschl. Berlin(West) hat
- mindestens 18 Jahre alt ist (Ausnahmen möglich)
- nicht vorbestraft ist
- eine fachliche Prüfung für Funkamateure abgelegt hat

a) Anforderungen der Deutschen Bundespost an die Fachkunde

Im Rahmen der fachlichen Prüfung werden Kenntnisse in folgenden Bereichen erfragt:
- Kenntnis von Vorschriften
- betriebliche Kenntnisse
- technische Kenntnisse
- Morsekenntnisse

b) Lizenzklassen

Es gibt hier z.Zt. 3 Lizenzklassen: A, B und C.

Die "große" Lizenz der Klasse B erlaubt den Amateurfunkbetrieb auf allen freigegebenen Frequenzen auf KW und UKW in allen Betriebsarten.
Die Klasse A erlaubt eingeschränkten Betrieb auf KW, schränkt den Betrieb auf UKW jedoch nicht ein.
Die "kleine" Lizenz der Klasse C erlaubt nur den Funkbetrieb im UKW-Bereich. Für diese Klasse sind keine Morsekenntnisse erforderlich.

3. Betriebsarten im Amateurfunk

Neben den herkömmlichen Betriebsarten wie Telegrafie (Morsen; hierbei wird die Information in Form eines Punkt/Strich-Codes übermittelt)und Telefonie (Sprechfunk; hier gibt es verschiedene Modulationsarten wie wir es alle aus dem Rundfunk kennen: AM und FM)gibt es mittlerweile eine Vielzahl von "moderneren" Betriebsarten, die nachfolgend kurz erläutert werden sollen.

a) RTTY (Radio Tele Typing)

RTTY, auch Funkfernschreiben genannt, ist die am weitesten verbreitete Betriebsart.
Zunächst mit ausgemusterten Fernschreibern betrieben, wird heute immer mehr mit Hilfe von Computern Funkfernschreibverkehr betrieben. Platzbedarf und Lärmpegel konnten so erheblich verringert werden.
RTTY ist sehr nahe mit Btx verwandt. Der digitale Code eines Zeichens wird in Töne umgesetzt, per Funk ausgesendet, beim Empfänger wieder decodiert und auf den Bildschirm des Computers gebracht. Die Übertragungsgeschwindigkeit auf UKW liegt wie beim Btx auf 1200 Baud. Im herkömmlichen Funkfernschreiben liegt die Geschwindigkeit bei 45,45 Baud.

b) ATV (Amateur Television)

Das ATV ist die mit dem höchsten technischen Aufwand verbundene Betriebsart. Aufbereitung und Aussendung eines ATV-Signals erfolgt nach den gleichen Prinzipien und Normen wie beim öffentlichen Fernsehen.
Wegen der erforderlichen Bandbreite von 5,5 MHz eignet sich diese Betriebsart nur zur Aussendung im UKW-Bereich (70 cm-Band). Die verwendeten geringen Leistungen von ca. 10 W erfordern drehbare Richtantennen.

c) SSTV (Slow Scan Television)

Entgegen der Möglichkeit, die beim Fernsehen üblichen bewegten Bilder zu übertragen, kann man beim SSTV - auch Schmalbandfernsehen genannt - nur stehende Bilder übertragen. Die Reichweite ist allerdings nahezu unbegrenzt, da die Signale üblicherweise auf KW übertragen werden.
Die von Funkamateuren entwickelte Technik erlaubt das Abspeichern eines normalen Fernsehbildes und späterer Aussendung mit 120 Zeilen. Die Übertragung eines kompletten Bildes dauert rund 8 Sekunden.

d) FAX (Faksimile-Funk)

Mittels des Faksimile- oder Bildfunks kann man vom sendeseitigen Original über Funk auf der Empfangsseite eine Kopie anfertigen.
Man erhält also auf einem Papier eine Hard-Copy.
Als populärstes Beispiel sei das Aufzeichnen von Aussendungen der Wettersatelliten erwähnt.
Neben der Elektronik ist der Anteil der Elektromechanik nicht unerheblich, damit derartige Aussendungen auf das Papier gebannt werden können. FAX-Geräte sind fast nicht zu kaufen, daher muß der Funkamateur fast alles selber bauen.

e) Amateurfunk-Satelliten

Der erste Amateurfunksatellit wurde im Jahre 1961 gestartet. Seitdem hat diese extrem kostenaufwendige Betriebsart einen "raketenhaften" Aufstieg hinter sich. Waren zunächst noch Batterien an Bord, so wird die Stromversorgung mittlerweile mittels Solarzellen und aufladbaren Akkus sichergestellt.
1983 erfolgte der Start des ersten deutschen OSCAR.
Mit Hilfe dieser Satelliten ist es selbst Amateuren mir der Lizenzklasse C möglich, weltweite Funkverbindungen herzustellen.
Auch Packet-Radio und ATV können über Satelliten abgewickelt werden. Zum Betrieb über Satelliten sind horizontal und vertikal drehbare Richtantennen nötig.

f) AMTOR (Amateur Teleprinting Over Radio)

Amtor eignet sich hervorragend zur Datenübertragung selbst unter schwierigsten Bedingungen auf Kurzwelle. Hiermit ist praktisch fehlerfreie Datenübertragung zwischen entsprechend ausgerüsteten Amateurfunkstationen möglich. Die Fehlerfreiheit geht allerdings auf Kosten der Übertragungszeit.
Im Gegensatz zum asynchronen 5er-RTTY-Code verwendet Amtor die synchrone Aussendung eines 7-Bit-Codes.

C. Die Betriebsart Packet-Radio (PR)

1. Allgemeines

PR ist eine noch sehr neue Betriebsart im Amateurfunk. PR ermöglicht eine gesicherte Übertragung von Texten über Funk.

PR kann man auf KW (sehr langsam ==> 300Bd) und auf UKW betreiben.
Die Übertragungsgeschwindigkeit beträgt im allgemeinen zwischen den Funkgeräten 1200 Baud, während Computer und Modem mit 9600 Baud kommunizieren können.

Der Hauptzweck beim PR-Betrieb muß die Beschäftigung mit Datenübertragungstechniken sein. Daher wird Binärfile-Übertragung auch nicht gerne gesehen. Hier ist zumindest zweifelhaft, ob es sich dabei noch um die vorgeschriebene Übertragung von "offener" Sprache handelt.

2. Der Netzaufbau am Beispiel DB0IL in Kiel

PR hat eine so rasante Entwicklung genommen, daß heute schon fast ganz Europa mit Umsetzern -sogenannten Digipeatern --> Digi von Digitaltechnik; Repeater von Umsetzer- versorgt ist.
Diese Umsetzer sorgen für eine größere Reichweite meiner Aussendungen, die im UKW-Bereich sonst höchstens Stationen im Umkreis von 30 km erreichen würden. Dies ist natürlich abhängig vom Antennenaufwand.

Mit Hilfe dieser Digipeater also kann ich PR-Verbindungen auf UKW von Norwegen bis nach Sizilien oder Spanien aufbauen.

Im Kurzwellenbereich ist die Sendung weltweit zu hören. Hierfür sind dann auch keine Umsetzer nötig.

In Kiel steht z.B. der Digipeater mit dem Rufzeichen DB0IL. Daran angeschlossen ist eine Mailbox, die in etwa 60 Rubriken zu allen möglichen Themen Dateien gespeichert hat. Diese Dateien können per Funk ausgelesen und dann auf meinem Datenträger, sei es nun Diskette oder Festplatte, gespeichert werden.

Weiter unten ist kurz aufgezeichnet, welche Verbindungen von Kiel aus möglich sind. Einerseits kann ich in Richtung Dänemark/Schweden versuchen, Verbindungen aufzubauen, andererseits ist der Aufbau einer Verbindung Richtung Süden über Lüneburg oder Hamburg möglich. Von jedem Digi gehen spezielle Abzweigverbindungen per Funk zu anderen Umsetzern.

Seit kurzem sind auch Funkamateure in der DDR in PR betriebsbereit und haben ihrerseits schon Digipeater aufgebaut.


```
Scandinavian Net
     VHF/UHF
         !
         !
  +-----------+
  :  OZ5BBS   :
  :  Odense   :
  :  JO55EJ   :
  +-----------+
         !
         !              :::::::::::::::::
         !              :KICONV:DB0IL-2:
+---------------+       @@@@@@@@@@@@@@@@@
:    DB0OQ      :       @  KIEL:DB0IL   @
:    Kiel       :--+    @     Kiel      @
:438.025 JO54AH:   !    @  R97  JO54AH  @
+---------------+  !    @@@@@@@@@@@@@@@@@
                   +----@KIEL23:DB0KI-1@
                        @@@@@@@@@@@@@@@@@
                                  !
   +------------------------------+
   !
   !
   !
   !                     ::::::::::::::::::
   !                     :SECONV:DB0HHN-4:
   !                     @@@@@@@@@@@@@@@@@@
   !        +------------@ #SE77:DB0HHN-7@
   !        !            @@@@@@@@@@@@@@@@@@
   ! ++++++++++++++++    @ SE70:DB0HHN-0 @
   ! +STADE:DL8XAW-2+    @   Henstedt    @
   ! + Grünendeich  +    @438.150 JO43XT @
   ! +   JO43TN     +    @@@@@@@@@@@@@@@@@@
   ! ++++++++++++++++    @ SE23:DB0HHN-3 @
   !        !            @@@@@@@@@@@@@@@@@@
   ! ++++++++++++++++             !
   ! +BRV70:DL4BBD-7+             !
   ! + Bremervörde  +             !
   ! +   JO43ML     +             !
   ! ++++++++++++++++             !
   !                              !
   !                    +---------+
   !                    !
@@@@@@@@@@@@@@@@@@@     !
@ #HHW4:DB0HHW-4@       !
@@@@@@@@@@@@@@@@@@@     !   @@@@@@@@@@@@@@@@@@
@ HHW70:DB0HHW   @      !   @ HHM70:DB0DJ   @
@ Hambg/Uetersen@       !   @    Hamburg     @
@438.125  JO43TQ@       !   @  R97  JN43XN  @
@@@@@@@@@@@@@@@@@@@     !   @@@@@@@@@@@@@@@@@@
@#HHW23:DB0HHW-3@---+---@ HHM23:DB0DJ-3@
@@@@@@@@@@@@@@@@@@@         @@@@@@@@@@@@@@@@@@
                                  !
                       weiter nach Lüneburg
```

HEFT
28
Oktober
1989

Jeder Digipeater weiß, welche anderen Umsetzer er erreichen kann. Diese Information kann ich abrufen und mich dadurch in die gewünschte Richtung durcharbeiten.

Nehmen wir einmal an, daß ich von Kiel aus nach München möchte. Dazu brauche ich bei bestehender Verbindung von Kiel aus nur den Befehl "C DB0MWE" (DB0MWE ist das Rufzeichen des Digipeaters in München) einzugeben und schon weiß der Digi, was er zu tun hat. Nach etwa 3-5 Minuten bin ich in München angekommen, was mir der dortige Digi an meinem Bildschirm mit einer Meldung anzeigt.

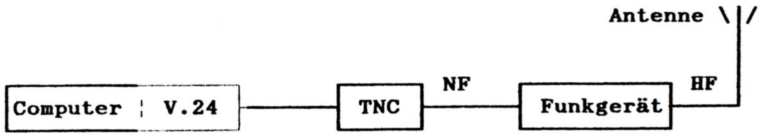
a) Digipeater,Links, Gateways

Nach Vorgaben des Deutschen Amateur-Radio-Clubs (DARC) muß der **Link** zwischen **Digipeatern** im 23cm-Band (1240 - 1300 MHz) erfolgen. Als Einstieg ist das 70cm-Band (430 - 440 MHz) vorgesehen.
Linkverbindungen laufen mit einer Leistung von 0,5 W bis ca.5 W. Als Antennen werden Richtantennen mit bis zu 20 dB Gewinn verwendet. Es gibt heute schon Richtfunkverbindungen, die im Vollduplexverkehr 9600 Bit/s übertragen.

Zu Verbindung von absolut inkompatiblen Netzen, die also mit unterschiedlichen Übertragungsprotokollen arbeiten, werden sogenannte Gateways benötigt. Jede PR-Station kann einen Gateway darstellen; z.B. zur Verbindung von Stationen einerseits mit AX.25-Protokoll und andererseits mit TCP/IP-Protokoll.

3. Was benötigt man für PR?

a) Hardware



Benötigt wird ein Computer mit V.24-Schnittstelle, ein TNC (**T**erminal **N**ode **C**ontroller), ein Funkgerät für das 70 cm-Band sowie eine Antenne.
Der TNC, der einfach an den NF-Ausgang des Funkgerätes angeschlossen wird, ist ein einfacher Computer mit Z80A-CPU, EPROM-Software, einer RS 232-Schnittstelle mit Z80A-SIO/0 und einem Modem mit dem IC TCM 3105.

Für manche Rechner (C64, Apple) gibt es PR-Software, so daß der TNC entfallen kann.

b) Software

Die erforderliche Software ist natürlich abhängig vom verwendeten Rechnertyp. Auf alle Fälle wird ein Terminalprogramm benötigt, das im Polling- oder Interrupt-Modus arbeiten kann. Nach meiner Erfahrung mit dem TRS 80 Model 4p kann ich sagen, daß die Verwendung von Programmen mit Polling-Modus die Transfergeschwindigkeit sehr verlangsamt. So ist z.B. der PR-Betrieb mit dem Model 4p nur mit 1200 Bd möglich, während sonst Übertragungsraten von 9600 Bd üblich sind.

Die Software ist nur lauffähig, wenn ich vorher mein Rufzeichen eingegeben habe. Hiermit soll verhindert, daß Nichtlizensierte diese Betriebsart anwenden.
Software gibt es für alle gängigen Rechner. Der Komfort der Programme für PC's und Atari-Rechner ist durchaus beachtlich. Der für mich größte Vorteil dieser komfortablen Programme liegt in der dreifachen Aufteilung des Bildschirms.
Die Größe dieser 3 Bereiche kann mit den Cursortasten verändert werden.
Der obere Bereich ist als "Vorschreibschirm" gedacht, nach Drücken der Return-Taste wird der oben geschriebene Text ausgesendet und erscheint im mittleren Bildschirmteil, in dem übrigens die für die eigene Station bestimmten Pakete erscheinen. Der untere Teil des Schirms ist als Monitorkanal zur Überwachung der Frequenz gedacht. Hiermit kann der Betrieb auf der Frequenz beobachtet werden. Diese Programme sind von Funkamateuren für Funkamateure geschrieben worden und ausdrücklich Public Domain.

Diese eben beschriebenen Vorteile haben die mir bekannten, für uns als TRS80-User verfügbaren Programme leider nicht, weder in den Systemen NEWDOS/GDOS (GDOS habe ich hier noch eingefügt, nachdem ich aus Kiel zurückkam, weil mir Arnulfs Rechner mit Festplatte so imponiert hatte) noch unter CP/M!
Entweder muß hier ganz auf die Monitorfunktion verzichtet werden (was natürlich möglich ist, gleich bei welchem Rechner), oder fremde und eigene Pakete kommen gemischt auf meinen Bildschirm; es ist dann nur noch mit Mühe ein für mich bestimmter Text lesbar.

Angeblich soll das beste Programm für CP/M "Modem7" sein. Ich habe bisher aber nicht geschafft, die notwendigen Parameter einzustellen, so daß ich zu diesem Programm nichts sagen kann.

Was sich hier so (hoffentlich) schön liest, war dann leider in der Praxis doch nicht so schön, weil die Funkverbindung zur Mailbox recht mickerig war. Das lag beileibe nicht an der Unterkunft -hier war alles bestens organisiert-, sondern an der mit 1,5 Watt geringen Sendeleistung meines Funkgerätes und der nur ca. 15 cm langen Antenne. Aber die Hauptsache war, eine Verbindung aufzubauen ,das Prinzip zu erläutern und einmal mehr darzustellen, was man noch alles mit einem Computer machen kann außer Briefe schreiben.

Zum Schluß, Dir, lieber Helmut, nochmals herzlichen Dank für die Organisation des Treffens. Es hat mir sehr gut gefallen!

IMPRESSUM

1. Vorsitzender
Alexander SCHMID
St.Cajetan Str. 38 VII
8000 München 80
☎ 089 /495326

2. Vorsitzender
Horst-Dieter SCHROERS
Breslauer Str. 9
8016 Feldkirchen
☎ 089 /9032615

Hardwarekoordinator
Andreas MAGNUS
Pommernstr. 4
4650 Gelsenkirchen
☎ 0209 /870230

NEWDOS-Diskothekar
Oliver VOLZ
Waldburgstr. 73
7000 Stuttgart 80
☎ 0711 /7353817

CP/M -Diskothekar
Rüdiger SöRENSEN
Thomas-Mann-Str. 3a
6500 Mainz 1
☎ 06131 /32860

Club-Bücherei
Christian MENK
Ollsener Str. 52
2116 Hanstedt
☎ 04189 /78251

Redaktion
Jens NEUEDER
Gschlachtenbretzingen
Rudolf-Then-Straße 32
7170 Michelbach /Bilz
☎ 0791 /42877

Autoren
Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto
Sonderkonto CLUB 80
Obermann,H. 6209 Heidenrod
Konto-Nummer 496 071 - 605
Postgiroamt Frankfurt
BLZ 500 100 60

Eine Zensur oder Kontrolle der Infobeiträge erfolgt nicht.
Die Redaktion

SCHLUSS

Hallo Club-80er,

endlich kann ich Euch nun wieder einmal ein umfangreiches INFO präsentieren. Dafür möchte ich mich bei Euch, den Autoren, bedanken.

Nachdem beim letzten Termin -zu dem ich Euch leider kein aktuelles INFO liefern konnte- recht wenig an INFO-Material vorhanden war, habt Ihr nun recht kräftig aufgeholt und ca. 4/5 dieses Club-INFO's angeliefert. Ich hoffe und wünsche es uns, daß es auch in Zukunft so weiter geht. Es wäre ja schade, wenn die Kommunikation im Club abreißen würde.

Als nächsten Redaktionsschluß möchte ich Mitte/Ende Dezember ansetzen, sodaß ich Euch nach dem Jahreswechsel das nächste Club-INFO zusenden kann.

Weiterhin habe ich noch eine Bitte an Alle, wenn Ihr Artikel für unser INFO einsenden möchtet:

- Bitte verwendet kein Recycling-Papier, da es beim Verkleinern kein so guten Kontrast zum Text abgibt.
- Bitte haltet die Zeilenzahl von 60 Zeilen ein. Ich brauche auch ein bißchen Rand.

Für Eure Mitarbeit danke ich im voraus.

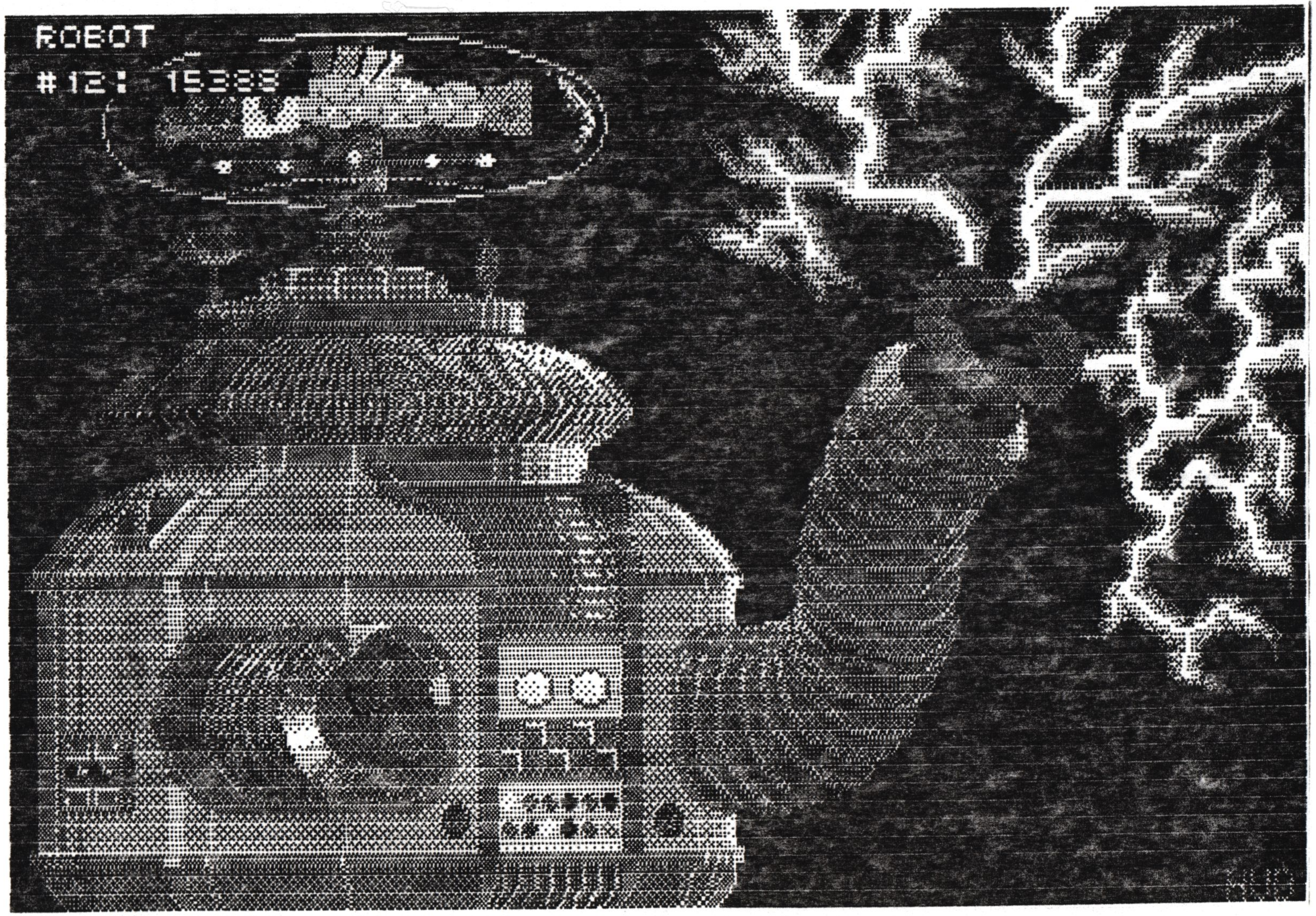
Als Beilage zu diesem INFO findet Ihr einmal das Sonderinfo
Floppy-Controller 1771 / 179x / 279x
sowie BIOS- / BDOS- Funktionen.

Nun wünsche ich Euch viel Spaß damit, an der Computerei und dem Trubel der kommenden Tage.

Es grüßt Euch Euer

Jens

HEFT 28
Oktober 1989


ROBOT
#12: 15388

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

CLUBINFO

29. AUSGABE

KONTAKTADRESSE : CLUB 80 / ALEXANDER SCHMID / ST. CAJETAN STR. 38 VII / 8000 MÜNCHEN 80

TEL.: 089 / 495326

- I N H A L T S V E R Z E I C H N I S -

Seite: und Autor:

Clubinternes

Neues vom Vorstand 01
Alexander Schmid
Fischköppe-Treffen 89 02
Hartmut Übermann, Gerald Schröder
Vorstellung 03
Stefan Nitschke
Termine 04
Redaktion
Weissagung .... 05 - 06
Klaus-Jürgen Mühlenbein

Software

Entdecke die Unbekannten .... 07 - 08
Klaus-Jürgen Mühlenbein
mittelmäßig ... . 09 - 10
Kurt Müller
TSCRIPS retten - wenn's geht, heute 11
Arnulf Sopp
Modifikation des HRG-Treibers RB 12 - 17
Egbert Schröer
Grafikpaket für die HRG 1B-Platine 18 - 31
Oskar Drechsler
Austausch illegaler Z80-Befehle ..... 33 - 36
Artikel aus MC

Seite: und Autor:

Hardware

Ein Versuch .... 37 - 52
Analog mit drei IC's .... 53 - 55
Kurt Müller

Börse

Wer hat was -- wer will was .... 56 - 57

Sonstiges

Mikrocomputer-Betriebssysteme -2+3+4 59 - 64
Gerald Schröder
Tips für den Computer-Kauf .... 65 - 66
Artikel aus ...
Der Computer .... 67 - 74
Artikel aus Hamburger Hochschulzeitung

Die letzten Seiten

Impressum 75
Schluß 76
Redaktion
Mitgliederadressenliste .... am INFO-Ende
Alexander Schmid
Inhaltsübersicht .... am INFO-Ende
Sonderinfo HD 64180 .... am INFO-Ende

Neues (?) vom Vorstand

Lange ist's her, daß das letzte Info erschienen ist. Eigentlich wollte ich die neuen Mitglieder immer einzeln begrüßen und willkommen heißen, aber nehmt es nicht persönlich, wenn ich jetzt nur sage: 'Herzlich willkommen, alle, die seit dem letzten Mal dazugekommen sind'. An den Vorstellungen habt Ihr ja gesehen, wer da neu zu uns gestoßen ist und wer was macht.

Es liegt mir fern, unverschämt oder bösartig zu sein, aber ich möchte sagen, daß viele wohl endlich mal gemerkt haben, was Sache ist. Die, die im Info immer um Artikel bitten und betteln werden mit einem Lächeln als Miesmacher und Störenfriede abgetan, aber jetzt jetzt hat es jeder selber gesehen: ohne neue Artikel läuft absolut garnichts. Das Info lebt nunmal von Euren Ideen und wenn nichts kommt, kommt auch nichts zurück. Wie es scheint, sind die ewigen Vielschreiber etwas kürzer getreten und jetzt ist die breite Masse gefordert. Irgendein Problem wird doch wirklich jeder schon gehabt haben, das ihm schlaflose Nächte bereitet hat. Ihr müßt ja nicht schreiben, daß Ihr eine Woche drangesessen seid, ein oder zwei Tage glaubt auch jeder und keiner lacht, daß Ihr Euch 'bei so einem lächerlichen Problem' so blöd angestellt habt. Wer sich da lustig machen kann, der soll sich selber an die Nase fassen und fragen, warum er das nicht schon längst im Info gebracht hat. Ich hoffe, daß keiner so schadenfroh ist und sich freut, daß er die Lösung hat, an der sich andere vielleicht die Zähne ausbeißen. Dann können wir uns gleich in 'Rambo-Club, Vereinigung der härtesten aller Einzelkämpfer' umbenennen.

Jetzt aber genug gelästert. Als ich das letzte Vorwort geschrieben habe, war von Weihnachten weit und breit nichts zu sehen und jetzt wird es wohl etwas zu spät sein, aber trotzdem wünsche ich Euch noch ein gutes Neues Jahr und dem Club viele fleißige Autoren.

Alexander Schmid

Fischköppe-Treffen 1989 in Heikendorf

Im Gegensatz zu den Nord-Italienern im Club haben es die Nordlichter auch dieses Jahr wieder geschafft, ein Regionaltreffen auf die Beine zu stellen. Diesmal hat sich ein Veteran des Clubs (Helmut Bernhardt) die Arbeit gemacht, eine geeignete Tagungsstätte zu finden und die Organisation zu übernehmen.

Anwesend waren diesmal (das dreckige Dutzend):

Helmut Bernhardt	(mit Tandon-PC)
Jörg Brans	(mit Model IVp und HP DeskJet)
Harald Braun	
Hans-Günther Hartmann	(mit Packet-Radio-Ausrüstung)
Andreas Magnus	(mit Genie IIIs im Tower-Gehäuse)
Kurt Müller	(mit Mega-ST)
Gerhard Neebe	(mit TRS-80 im PC-Gehäuse)
Hartmut Obermann	(mit Toshiba-PC)
Frank-Michael Schober	
Gerald Schröder	(mit Genie IIs)
Arnulf Sopp	(mit Genie IIIs)
als Gast: Erik-Jan Plog	

Der "harte Kern" reiste schon am Freitag an, während sich die Masse erst am Sonnabend von ihren Familien losreißen konnte. Neben dem offiziellen Teil, der sich aus vier Vorträgen zusammensetzte, wurde gefressen, geraucht, gesoffen und natürlich die Hardware begrabbelt. Folgende Themen kamen zum Vortrag:

Zunächst stellte Hans-Günther die Möglichkeiten des Packet-Radio (Datenübertragung per Funk) vor. Da er zur Zeit noch kein geeignetes Programm für CP/M und NEWDOS zur Verfügung hat, lief die Demonstration auf Helmuts PC (den Helmut nur zu diesem Zweck mitgebracht hatte).

Nach einer kurzen Verschnaufpause und einigen Bierchen führte Kurt seinen selbstgestrickten Analog-Digital-Wandler vor. Der Wandler läuft an jedem Rechner mit Centronics-Schnittstelle und eignet sich z.B. zur Aufnahme von Meßwerten (Temperatur, Spannung usw.). Als Beispiel führte Kurt auf seinem Atari vor, wie man aus einem Rechner ein Speicher-Oszilloskop macht.

Zu diesen beiden Vorträgen werden sich noch Veröffentlichungen im Info gesellen, damit auch das faule Volk, das dieses Jahr (wieder) nicht zu unserem Treffen erschienen ist, davon was hat (Zitat Hartmut). Die folgenden Geistes-Ergüsse bleiben dagegen dem dreckigen Dutzend vorbehalten.

Da wäre als erster Arnulf zu nennen, der in unvergleichlicher Manier vorführte, wie die moderne Technik in Form eines Diktiergeräts Souffleur und Manuskript ersetzt (fast sogar den Vortragenden selber). Inhaltlich bezog sich sein Vortrag auf die Optimierung von Assembler-Programmen, wobei er besonders einige sehr leistungsfähige, aber wenig benutzte Z80-Befehle hervorhob. Andreas ergänzte dieses Kapitel mit einem Referat über die unterschiedlichen Möglichkeiten, einen Speicherbereich mit einem Byte zu füllen.

Abschluß des Vortrags-Marathons bildete Gerald, der sich über die Kommentierung und Dokumentation von Programmen, besonders Assembler-Programmen, ausließ. Dabei wurde deutlich, daß optimierte Assembler-Programme Gefahren in sich bergen, wenn sie nicht sehr ausführlich kommentiert sind. Obwohl die beiden Vorträge eigentlich sehr unterschiedliche Aspekte der Programmierung beleuchteten, ergänzten sie sich doch.

Besonders erwähnen möchten wir noch unsere Überraschung und Freude, als plötzlich unser DDR-Vertreter Frank-Michael auftauchte. Helmut hatte Franks Anreise geheim gehalten, so daß vor allem Hartmut, der gerade mit dem Booten von Foodware beschäftigt war, das Gebiß aus dem Gesicht fiel (Zitat Gerald).

Abschließend bedanken wir uns ganz herzlich bei Helmut, der uns Quartier gab und seine Wohnung auch bis 5 Uhr morgens für die Hartgesottenen als Tagungsort zur Verfügung stellte.

Hartmut und Gerald

HEFT 29 Januar 1990

Hallo Clubmitglieder

Ich möchte mich kurz vorstellen.

Mein Name ist Stefan Nitschke und wohne in Walzbachtal 1. Ich bin 25 Jahre alt und studiere Physik an der Universität Karlsruhe.

Zu CP/M kam ich durch den günstigen Einkauf eines ECB-Bus Rechners Anfang diess Jahres. Es stellte sich jedoch für mich als schwierig heraus, Software für CP/M-Rechner zu bekommen, da meine Bekannten Atari ST oder MSDOS Rechner benutzen.

Das Betriebssytem meines Rechners (ZDOS) ist CP/M 2.2 kompatibel.

Interresse habe ich an fast allen (Höheren)Programmiersprachen, und Hardwareprojekten. Für Numerikprogramme habe ich hier einen Arithmetikprozessor 8231A von Intel herumliegen.
Das Einbinden einer Grafikkarte in CP/M 2.2 interessiert mich ebenfalls. Ich kenne das Angebot für ECB-Bus Systeme nicht außer Grip5 von Conitec welche in den Anzeigen die ich davon habe runde 700 DM kostet (sind allerdings schon etwas alter). Wenn einer von Euch ein preisgünstigeres Angebot kennt z.B. auch eine Selbstbau-Karte meldet Euch doch bitte.

Ich programmiere in Turbo-Pascal, Turbo-Modula, Assembler und ein wenig in Fortran.
Turbo-Pascal habe ich an den 8231A Coprozessor angepasst und bring nur für einen Z80-Rechner eine genehme Rechenleistung. Eine entsprechende Anpassung an Modula2 habe ich noch nicht geschrieben, da ich den Compiler erst seit kurzem habe. Wenn einer von Euch schon eine Einbindung vorgenommen hat oder mir ein paar Tips geben kann schreibt mir doch bitte, oder schreibt an Jens Neueder.

Mein Rechner besteht aus:

CPU-Karte	: Z80 mit 4Mhz Takt von Janich+Klass HKM 105X
Floppycontr.	: JK 82 FDC 8/5 ebenfalls von Janich+Klass
Speicher	: 64 KBytes Eigenbau, 2 x 5.25'' 40-Spur-Laufwerke
Terminal	: Infoton nicht graphikfähig, keine Attribute, maximal (nur) 9600 Baud
Drucker	: Speedy 100-80

Meine Adresse:

Stefan Nischke
Germanenstr.5
7519 Walzbachtal 1
Tel.: 07203/452

Mit Grüßen

Stefan

PS.
Ich hoffe Ihr seid mittlerweile nicht alle vom PC, AT oder ST Fieber angesteckt.

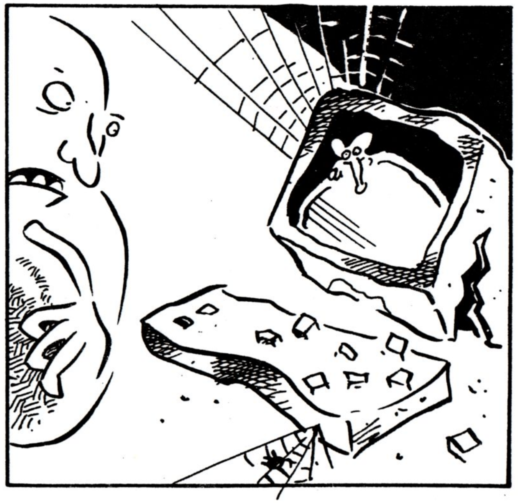
-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß .... Mitte / Ende Februar

Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!

-- Messen '90 --

Int. Spielwarenmesse .... Nürnberg ..... 08.02. - 14.02.
mit Fachmesse Modellbau, Hobby u. Basteln
CeBIT .... Hannover ..... 21.03. - 28.03.
Telematica .... Stuttgart .... 11.06. - 14.06.
electronica .... München ...... 06.11. - 10.11.



* * * W E I S S A G U N G * * *

Der Satz:

"Mit Statistik kann man alles beweisen"

muß ganz entschieden bestritten werden!

Wer das behauptet (und das tun viele) will entweder Böses aushecken oder hat keine Ahnung von den mathematischen Grundlagen wahrer Statistik (sog. "Stochastik")

Das Folgende darf daher nicht so ernst genommen werden (falls einer meiner Beiträge überhaupt je ernst genommen wurde). Es geht schlicht um eine Behauptung; vielleicht auch eine Weissagung - insoweit aufgrund einer Statistik extrapoliert werden darf.
Doch keine Furcht: Falls die "Weissagung" berechtigt ist, bedeutet sie nicht mehr und nicht weniger, als daß die Zukunft als ein "weißes Blatt" vor uns - dem "CLUB80" - liegt, d.h. als ein leeres Blatt, und dieses mag außer seiner Weißheit vielleicht sogar einige Weisheit auf-weisen... (Im Malerjargon auch "aus-weißen".)
Als ich dem "CLUB80" beitrat, waren bereits 9 Hefte erschienen. Diese kaufte (!) ich sz. nach. Ihre Eingangsdaten kenne ich deshalb nicht. Als Neuling wurde mir jedoch mitgeteilt, daß alle zwei Monate ein Heft, jährlich also sechs Hefte erscheinen würden. Da bei mir eingehende Post einen Eingangsvermerk erhält (falls sie mir wichtig oder interessant erscheint), war es ein leichtes, die nachstehende "Statistik" anzufertigen.

Sie soll diejenigen einmal nachdenken machen, die mit dem Z80 und seinen Nachfolgern umgehen und daran interessiert sind, Erfahrungen von Gleichgesinnten (Gleichsündern) und Gleichgesonnenen (gleichen "Sonnen") zu erfahren - wenn schon nicht mit diesen auszutauschen .

Ich weiß nicht, ob es noch einen entsprechenden Club geben wird, zu dem diese in 1991 ausweichen können, falls das Tabellenende (die "Statistik") stimmt! Wahrscheinlich nicht; sondern nur noch Clubs der "Kompatiblen" - sprich: IBM-Jünger!

Also: Geniert Euch nicht, Ihr Stummen !! Packt aus, was Ihr bisher angepackt habt an Eurem stummen Diener! Nur ER ist zur Stummheit = Dummheit verdammt. Auch wenn er inzwischen noch so hochgerüstet wurde!
Packt Eure Aktivitäten aus, seien sie auch noch so bescheiden! (Ich tu's auch...) Denn wer schreibt, der denkt. Wer denkt, der kann irren (muß aber nicht), und wer irrt, ist überhaupt nicht dumm - im Gegenteil, denn er dachte; - nur wer schweigt und nie was sagt oder auch fragt - ja, der ist wahrscheinlich wirklich dumm!
Dieses statement soll die nicht treffen, die bisher keinen Beitrag geleistet haben, weil sie absolut keine Zeit hatten (??), sondern die, die in Zukunft noch ...

Na denn, schaut Euch erst mal die Tabelle an. Vielleicht geht manch einem von selbst ein Licht auf!

* * * INFO-Statistik des CLUB-80 (extrapoliert)* * * *

(oder: "Die Rückwärtsreihe der Natürlichen Zahlen")

Heft Nr.	Eingang	Anz.	Preis DM/Heft
10	Nov.85		
11	Jan.86		
12	25.3.86		
13	27.5.86	(6)	8.33
14	26.7.86		
15	2.10.86		
16	Weihn.86		
17	21.1.87		
18	12.3.87		
19	23.6.87	(5)	10.-
20	25.8.87		
21	17.11.87		

Heft Nr.	Eing.	Anz.	Preis DM/Heft
22	22.1.88		
23	14.4.88		
24	23.7.88	(4)	12.50
25	12.10.88		
26	10.5.89		
27	26.7.89	(3)	16.67
28	9.12.89		
29	?.??.90		
30	?.??.90	(2)	25.-
31	?.??.91	(1)	50.-

Lohnt sich das?

Preisbasis: 50.-DM/a

*P.S.: „extrapoliert" does not mean: « Extra polished », like „extra poliert" (= geschönt) but « extended » ...

NB: Ich weiß natürlich, daß außerdem Werte vorgehalten werden! [Bücher, Programme] + Porti ...

HEFT 29 Januar 1990

* * Entdecke die Unbekannten ! * *

Boucuse kocht nicht nach Rezepten.
Die Rezeptur entwickelt er im Kopf.
(Trotzdem schmeckt es vielen...)

Zwar bin ich kein Koch. Und schon gar kein *Boucuse*. Doch reizt es auch mich nicht, meine Suppe nach fertigen Lösungsrezepten – wir nennen sie "Programme" – zu kochen. Die Lösung programmiere ich lieber selbst. Auch wenn es schon zwei bis hundert "bessere" Programme gibt.
Es soll sogar Köche geben, denen ihr eigenes Gebräu und Gebrutzel am besten schmeckt. Weil nämlich die eigene Zuwendung und Arbeit drinsteckt.
In diesem (oder dem letzten) Heft findet ihr mein Programm zur Lösung von Determinanten beliebiger Ordnung.
Ein System aus m linearen Gleichungen für n Unbekannte ist dann besonders sympathisch, wenn genau so viele Gleichungen zur Verfügung stehen wie Unbekannte zu lösen sind, d.h. wenn n=m.
Noch sympathischer ist es, wenn mindestens eine dieser Gleichungen von Null verschieden, also wenn sie 'inhomogen' ist. Dann kann nämlich das Lösungsverfahren nach *CRAMER* angewendet werden. Dies läuft folgendermaßen:
Zuerst werden die Unbekannten in allen Gleichungen mitsamt ihren Koeffizienten in der gleichen Reihenfolge auf die linken Seiten gebracht und die Konstanten (die sog. "absoluten Glieder") auf die rechten Seiten. Die Koeffizienten bilden dann eine Matrix. Deren Determinante wird berechnet. (Wir nennen sie die "Hauptdeterminante" oder den "Nenner").
Dann werden die Koeffizienten der ersten Unbekannten in allen Gleichungen durch die jeweiligen absoluten Glieder ersetzt. Die Determinante dieser leicht veränderten Matrix wird berechnet und durch den Nenner dividiert. Das Ergebnis ist die erste Unbekannte.
Zur Berechnung der übrigen Unbekannten wird in gleicher Weise verfahren.
Wie man sieht, ist dies ein stures Schema, also bestens für unseren Rechenknecht geeignet.
Nachdem ein Programm zur Lösung von Determinanten beliebiger Ordnung (!) vorlag (s.o.), war es naheliegend, es bei dem hier beschriebenen Verfahren nach *CRAMER* anzuwenden.
Wie das geschieht, zeigt mein nachstehendes "wohlstrukturiertes" Programm. Was bei seinem Ablauf gerade geschieht, wird auf dem Bildschirm laufend mitgeteilt. Diese Mitteilungen dienen im Listing gleichzeitig als Kommentare für die Unterprogramme. (Auch in PASCAL sind Unterprogramme das Konzept der Strukturierung.)
Wenn wenig zu rechnen ist (wenige Unbekannte), laufen diese Mitteilungen so schnell ab, daß man sie kaum lesen kann – was ja auch gar nicht so wichtig ist; nur bei längeren Berechnungen möchte man sicher sein, daß der Rechner was tut.
Schließlich wird der Bildschirm gelöscht und die Ergebnisse, d.h. die Lösungen der n Unbekannten werden angezeigt.
Das Programm wurde mit mehreren Gleichungssystemen verschiedener Ordnung getestet. (Wer evtl. trotzdem noch einen Fehler findet, teile ihn mir bitte mit.) Seine Geschwindigkeit ist befriedigend. Das entsprechende Programm von *Rugg und Feldman*, das der eine oder andere von euch vielleicht kennt, läuft auch nicht schneller.
Εὕρηκα!
(H-eure-ka-jot)

Lösung eines inhomogenen linearen Gleichungssystems

```
10 'Mühlenbein,  Weinheim, 27.10.1989
20 CLS:PRINT"  Lösung eines inhomogenen Gleichungssystems":
PRINT:PRINT"Das Gleichungssystem muß in der 'Normalform' vo
rliegen:":PRINT"Links vom Gleichheitszeichen die Unbekannte
n mit ihren
Koeffizienten, rechts die Konstanten!":PRINT
30 DEFINTI-Z:INPUT"Anzahl Unbekannte (= Anzahl Gleichungen!
)";R:DIMA(R,R),C(R,R),B(R),D(R):PRINT:PRINT"Eingabe :":PRIN
T:FORZ=1TOR:PRINT"Zeile";Z:FORS=1TOR:PRINT"   Spalte";S;":"
;:INPUTA(Z,S):C(Z,S)=A(Z,S):NEXTS:PRINT
40 INPUT"Rechte Seite (Konstante) ";B(Z):PRINT:PRINT:NEXTZ
50 FORSS=0TOR
60 PRINT:PRINT"Rechte Seite einsetzen":GOSUB110
70 PRINT:PRINT"Berechnung der Determinante":GOSUB130
80 PRINT:PRINT"Hauptmatrix regenerieren":GOSUB120
90 NEXTSS
100 CLS:FORS=1TOR:PRINT:PRINT"   X(";MID$(STR$(S),2,3);") =
 ";D(S)/D(0):NEXT:END
110 FORZ=1TOR:A(Z,SS)=B(Z):NEXT:RETURN
120 FORZ=1TOR:FORS=1TOR:A(Z,S)=C(Z,S):NEXTS,Z:RETURN
130 FORS=1TOR-1:IFA(S,S)=0THENGOSUB150
140 FORJ=0TOR-S-1:F=A(S+J+1,S)/A(S,S):A(S+J+1,S)=0:FORI=1TO
R-S:A(S+J+1,S+I)=A(S+J+1,S+I)-F*A(S,S+I):NEXTI,J,S:D=1:FORI
=1TOR:D=D*A(I,I):NEXT:D(M)=D*(-1)ÄX:M=M+1:RETURN
150 FORI=S+1TOR:IFA(I,S)<>0THENGOTO160ELSENEXTI
160 FORU=1TOR:CMD"F=SWAP",A(S,U),A(I,U):NEXT:X=2*(I-S)-1:RE
TURN
```

mittelmäßig...

In meinem Programm zum ANAloginterface benutze ich zur Anzeigeberuhigung eine Mittelwertbildung, die nicht erst n Werte aufsummiert und dann das Ergebnis präsentiert. Die dadurch entstehende Wartezeit erschien mir unpraktisch zumal bei sehr großem n die Wartezeiten wirklich stören.

Hiervon ausgehend, habe ich mir überlegt, wie der Vorgang der Summenbildung so umgestaltet werden kann, daß er mit jedem eingelesenen Meßwert einen Mittelwert ausgibt. Bei der "normalen" Lösung wird - nehmen wir an über 10 Werte - solange alles auf einen Haufen gepackt, bis eben jene 10 Werte eingelesen sind. Dann erfolgt die Division durch 10, womit der Mittelwert berechnet ist. Schwankt der Meßwert nun während dieser Zeit, so werden natürlich auch die einezelnen Meßwerte unterschiedlich sein. Die schwankungsbreite dividiert durch 10 ergibt dann die noch verbleibende "Unruhe" in der Anzeige. Daran ändert auch meine Lösung nichts, aber es spricht nichts dagegen, diese Mittelwertbildung bis zum erreichen eines Endwertes anzuzeigen. Womit die Forderung pro Meßwert eine Ausgabe konform geht.

Das Problem in einen Algorithmus aufgedröselt sieht dann so aus.

<u>alter Weg:</u>

$$\hat{u} = \left(\sum_{1}^{n} u\right) / 10$$

û -> anzeigen

<u>neuer Weg:</u>

û(1..n) = 0 :Alle Elemente Null setzen

DO

$\hat{u} = \hat{u} - u_{n-1}$:u_{n-1} = letzter alter Meßwert

$\hat{u} = \hat{u} + u_n$:u_n = neuer Meßwert

û / 10 -> anzeigen

LOOP

Damit meine Methode funktioniert, ist ein eindimensionales Array mit n Felder notwendig, in das die Meßwerte eingtragen werden. Ausprogrammiert sieht das dann so aus:

```
OPTION BASE 0               !Es gibt ein "Nulltes Element"
summier_faktor%=10          !Über 10 Werte Mitteln
DIM mittel%(summier_faktor%)
ARRAYFILL mittel%(),0       !Alle Elemente Null setzen
mittel%(0)=1                !Feld 0 von mittel%() = Array-Pointer ins Feld !
```



```
summe%=0
adc%=0
REPEAT
  rd_adc(ska!,off!)         !Hole einen Meßwert vom ADC
  mittel(adc%,summe%,mittel%()) !Bilde Mittelwert
  PRINT AT(1,1);adc%        !Wert ausgeben
UNTIL INKEY$<>""
END
'
PROCEDURE mittel(adc_wert%,VAR summe_wert%,mittel_wert%())
  IF summier_faktor%>1 THEN
    SUB summe_wert%,mittel_wert%(mittel_wert%(0))
    ADD summe_wert%,adc_wert%
    mittel_wert%(mittel_wert%(0))=adc_wert%
    INC mittel_wert%(0)
    IF mittel_wert%(0)>summier_faktor% THEN
      mittel_wert%(0)=1
    ENDIF
    adc%=summe_wert%/summier_faktor%
  ELSE
    adc%=adc_wert%
  ENDIF
RETURN
```

Für diejenigen, die sofort mit klammen Fingern auf's Hackerbrett zustürzen und meine kühnen Behauptungen erproben wollen, zuvor noch einige Zeichenerklärungen. Das Ausrufezeichen wie auch das einfache Hochkomma leiten eine Kommentar im Programmbeispiel ein. Im Unterprogramm "MITTEL()" taucht bei den Übergabeparametern die Anweisung "VAR" auf. Das bedeutet, daß die nachfolgenden Variablen selbst übergeben werden. Mit anderen Worten, das Unterprogramm kann den Inhalt selbiger verändern. Die Variablen werden jedoch auf den Namen im Prozedur-Kopf umgetauft. Die Zuordnung erfolgt in der Reihenfolge der Nennung:

```
           mittel(adc%,          summe%,      mittel%())
                     :               :            :
                     v               v            v
PROCEDURE mittel(adc_wert%,VAR summe_wert%,mittel_wert%())
```

Arrays werden dabei ohne Angabe der Dimmension übergeben. Übrigens kann unter Benutzung des Unterprogramm MITTEL() sehr einfach die 1. Ableitung des Meßwertes gebildet werden. Um diese zu erhalten muß lediglich vom aktuelle Meßwert der durch MITTEL() berechnete Wert subtrahiert werden. Also:


```
rd_adc(ska!,off!)              !Hole einen Meßwert vom ADC
mittel(adc%,summe%,mittel%())  !Bilde Mittelwert
diff%=mittel%(mittel%(0)-1)-adc% !1. Ableitung berechnen
```

Logisch, das jetzt natürlich sofort der Regelungstechniker einhackt und sich im stillen fragt: "Wenn das so ist, wie baue ich mir dann einen PI-, PID- oder PD-Regler zusammen...?" Ich für meinen Teil verdrück mich jetzt nach unten zu Frauchen an den Wehnachtstisch und überlaß den Hackern das Feld. .....Tschüß

TSCRIPS retten –
wenn's geht, heute noch!

Lese ich richtig? Da ist mir also bei irgendeinem Systemabsturz des Bierglases auf das CPU-Board ein TSCRIPS-Text verstümmelt worden. Den soll ich nun erstens bis sechstens bzw. gegf. über fünf-a-tens und DUMP und SUPERZAP ... Nö, ehrlich!

Die Rede ist von einem Verfahren zum Restaurieren von Speicher-Datensätzen (auch ZEUS-Sources usw.), bedichtet von Kajot im Info Nr. 28. Sehen wir uns das Vorher und Nachher genauer an: Falsche Bytes, die TSCRIPS nicht alleine auf die Reihe kriegt, werden geändert ... Warum guckt ihr so ungeduldig? Damit ist es doch schon erschöpfend beschrieben, mehr ist es nicht!

Das einfachste Werkzeug für Speichermanipulationen ist immer noch der Debugger. Man braucht für ihn nicht einmal zwei gesunde Hände, nur drei Finger. Die legt man – möglichst gleichzeitig – auf die Zifferntasten 1, 2 und 3 (im normalen Alpha-Block der Tastatur). Anschließend das Loslassen nicht vergessen, sonst setzt im jetzt bereits ungeduldig mit dem Cursor zappelnden Debugger die Tastenwiederholung ein! Wie jener betan wird, steht in der Anleitung zum DOS. Und davon ist auch nur das M-Kommando wichtig.

Die Verstümmelung wird man wohl während der Arbeit mit TSCRIPS bemerkt haben. Warum aussteigen? Mit 1-2-3 kann man nicht nur komfortabel vorübergehend fremdgehen – mit G, ENTER kommt man ebenso sutsche auch wieder rein, und zwar genau da, wo man die Arbeit unterbrochen hat. Nur der Bildschirm sieht jetzt wild aus. Mit Shift-Hochpfeil oder irgendeiner anderen Scrollerei ist auch das wieder eingerenkt.

Aber nicht nur die Bequemlichkeit spricht für Debug. SUPERZAP (oder wie sie alle heißen) ist etliche kB groß. Der zu rettende Text wird in der Kombination TSCRIPS-SUPERZAP noch nicht überschrieben, vielleicht aber bei einer anderen Paarung von Datenschaufel und -doktor. Der Debugger aber ist ein SYS-File, belegt also kein Bit des Anwenderspeichers. Damit sind die Daten sicher.

Zwei Tips noch:

Oft will man aus TSCRIPS aussteigen, ohne den Text zu drucken. Er soll nur erst mal auf die Diskette, weil er z. B. noch nicht fertig ist. Der Drucker ist nicht eingeschaltet. In diesem Fall fragt ihn TSCRIPS hartnäckig ab und kehrt erst ins DOS zurück, wenn man endlich klein beigibt und ihn anmacht. Das geht elegant auch mit 1-2-3: Jetzt sind wir im Debugger. Mit dem Kommando Q, ENTER springt dieser das DOS an.

Der Text für TSCRIPS steht keineswegs bei allen Versionen ab 9177h. Kaum ein Programm wurde von so vielen Leuten so oft modifiziert wie TSCRIPS. Es empfiehlt sich daher, zuvor nachzusehen. Mit Debug.

Arnulf Sopp

Modifikation des HRG Treibers von RB-Elektronik

Umlaute mit NewDos/80 Vers.2.0; neuer Tastaturtreiber

Der hier beschriebene Patch des Supertreibers ermöglicht die Eingabe von Umlauten, die Rückkehr vom BASIC zum DOS ohne die Tastenkombination 1,2,3. Außerdem erübrigt sich eine neue Definition des G-Zeichensatzes zur Darstellung von Umlauten durch einlesen des Zeichensatzes. Die Umlaute sind jederzeit aktiv. Nachteil ist das stete Geflimmer der HRG, aber hier kann eventuell ein Hardware-Fachmann helfen ?

Grundidee dieses Patches war eigentlich nur ein neuer Tastaturtreiber, der es mir ermöglichen sollte jederzeit Umlaute ohne große Mühe einzugeben. Die Bedingung dazu war, daß dies mit der gleichen Tastenkombination geschehen sollte, wie bei dem von mir durchgeführten SCRIPSIT-Patch (siehe SONDERINFO 27.5). Außerdem sollte die eigentliche Funktion der Tasten nicht verändert werde. Als dieser Tastaturtreiber fertiggestellt war, stürzte ich mich auf den HRG-Supertreiber V1.1 von RB-Elektronik. Es ist nämlich nicht einzusehen, daß man nach dem Laden des Treibers und Basic nun auch noch den G-Zeichensatz beispielsweise mit Umlauten neu definieren muß. Zusätzlich störte mich schon immer, daß man über DEBUG (Tasten 1,2,3) in das DOS zurückkehren mußte.
Hilfsmittel für die nun folgende kriminalistische Kleinarbeit war das Monitorprogramm TASMON.

Doch zunächst zum neuen Tastaturtreiber, der ganze 124 Bytes belegt. Min. 25 Bytes könnte man einsparen, indem man Teile des Tastaturtreibers in eine freigewordene Stelle des HRG Supertreibers installiert. Ich hielt dies nicht für sinnvoll, möchte aber auf die Möglichkeit hinweisen.

Listing 1: Tastaturtreiber:

```
;*****************************************************************
;*                 Keyboard Driver für HRGDOS                    *
;*****************************************************************
;*                Copyright by Egbert Schröer                    *
;*                      Version 1.1                              *
;*                    October  1989                              *
;*****************************************************************
;
        ORG     4049H           ;setzt HIMEM neu
        DEFW    KBDNEU-1
```

```
        ORG     0DA84H              ;oder wo man möchte
;
;Entry Point muss dem HRGDOS angepaßt werden
;
KBDNEU: LD      HL,(4016H)          ;Adresse KBDNEU in 4016H
        LD      (DRIVER+1),HL
        LD      HL,DRIVER
        LD      (4016H),HL
        JP      0EB36H              ;umschalten auf HRG
DRIVER: CALL    $-$                 ;Tastaturabfrage -> Taste
;                                   gedrückt ? Wenn ja,dann
        OR      A                   ;sichere den Tastaturcode
        RET     Z                   ;in AKKU
        LD      (AKKU),A
;
;Wenn § gedrückt, dann warte auf nächste Taste. Ist
;es kein Vokal, wird § ausgegeben. Folgt ein
;Vokal, wird dieser in seinen entsprechenden
;Umlaut umgewandelt.
;
        LD      A,(3801H)           ;Abfrage Tastaturzeile
        BIT     0,A                 ;BIT 0 gesetzt ?
        JR      Z,BACK              ;Nein, dannn ->ENDE
        CALL    0049H               ;Abfrage nächste Taste
        LD      A,(3880H)           ;und warten. Wenn <SHIFT>
        CP      1                   ;dann Großbuchstabe
        JR      Z,GROSS
        LD      A,(3801H)
        CP      2
        JR      Z,UMF1
        LD      A,(3802H)
        CP      128
        JR      Z,UMF2
        LD      A,(3804H)
        CP      32
        JR      Z,UMF3
        LD      A,(3820H)
        CP      4
        JR      Z,UMF4
KEINV:  LD      A,40H               ;kein Vokal ->dann ALPHA
        JR      ENDE                ;in Akkumulator
GROSS:  LD      A,(3801H)
        CP      2
        JR      Z,GUMF1
        LD      A,(3802H)
        CP      128
        JR      Z,GUMF2
        LD      A,(3804H)
        CP      32
        JR      Z,GUMF3
        JR      KEINV               ;kein Vokal
;kein § gedrückt -> zurück und weiter mit Zeichen
;in AKKU !
```



```
BACK:   LD      A,(AKKU)
        RET
ENDE:   RET
;*********************************************************
;*              UNTERROUTINEN                            *
;*********************************************************
;
UMF1:   LD      A,7BH               ;ä
        JR      ENDE
UMF2:   LD      A,7CH               ;ö
        JR      ENDE
UMF3:   LD      A,7DH               ;ü
        JR      ENDE
UMF4:   LD      A,7EH               ;ß
        JR      ENDE
;
GUMF1   LD      A,5BH               ;Ä
        JR      ENDE
GUMF2   LD      A,5CH               ;Ö
        JR      ENDE
GUMF3   LD      A,5DH               ;Ü
        JR      ENDE
;
AKKU    DEFS    1
        END     KBDNFU
```

Die Abfrage der Tasten erfolgt direkt über die Abfrage der entsprechenden Tastaturzeile. Der Entry Point DA84H ist auf den HRG Supertreiber abgestimmt. Der Tastaturtreiber kann aber auch im Speicher frei verschoben und ohne HRG benutzt werden.
JP EB36H wird später erläutert.

Nun zum HRG-Treiber. Um die Umlaute stets präsent zu haben, wurden diese einmalig in den G-Zeichensatz eingelesen und anschließend das Programm von DB00H bis FF24H mit Entry Point EB00H auf Diskette gebannt.

Die Darstellung der Umlaute sollte mit der Option #PRINT(2) erfolgen, die den normalen Bildschirminhalt auf die HRG umleitet und dabei den definierten G-Zeichensatz benutzt. Anschließend sollte ein Rücksprung in das DOS erfolgen.
Listing 2 zeigt das disassemblierte Programm von EB00H bis EBFCH, Listing 3 den ausgeführten Patch von EB00H bis EB78H und den Patch ab F6C5H (#Print(2)-Befehl: Patch ermöglicht Rücksprung zum DOS).

Listing 2: HRG-Supertreiber V1.1

```
EB00    LD      HL,EB79H
```

```
         LD      (4ØØ4H),HL
         LD      HL,ØDAFFH
         LD      (EB74H),HL
         LD      HL,ØAAFFH
         LD      (EB76H),HL
         LD      HL,(4Ø1EH)
         LD      (ØF6B7H),HL
         LD      A,(4Ø12H)
         CP      ØFBH
         JP      Z,EB32H
         LD      HL,(EB74H)
         LD      (4Ø49H),HL       ;setzt HIMEM
         LD      HL,EB51H         ;HL auf Text ab EB51H
         CALL    4467H            ;Ausgabe Text RB-EL HRG usw.
         LD      HL,EB4BH         ;HL auf Text ab EB4BH=BASIC
         JP      4419H            ;DOS Call-> BASIC
EB32H    LD      HL,ZEB51H
         CALL    28A7H
         LD      HL,(EB74H)
         LD      (4ØB1H),HL
         LD      DE,ØFFCEH
         ADD     HL,DE
         LD      (4ØAØH),HL
         CALL    1B4DH
         JP      1A19H
EB4BH    DEFM    'BASIC'
         DEFB    ØDH
EB51H    DEFM    'RB-EL HRG SUPER-TREIBER V1.1'
         DEFB    ØDH
         NOP
         LD     B,L               ;Version EPS48
         LD     D,B
         LD     D,E
         INC    (HL)
         JR     C,EB75H
EB75H    NOP
EB76H    NOP
         NOP
EB78H    NOP
EB79H    EX      (SP),HL
         LD      A,H
         CP      1DH
         JR      Z,EB83H
         CP      2ØH
         JR      NZ,EB8AH
EB83H    LD      A,L
         CP      5BH
         JR      Z,EB8AH
         CP      5ØH
EB8AH    EX      (SP),HL
         JP      NZ,1D78H
         CALL    1D78H            ;RST 1ØH: analysiert Programmtext
         PUSH    AF               ;HL dient als Zeiger
```

Hier kann geändert werden!


```
         CP      23H              ;-> #
         JP      NZ,EC46H         ;nein, dann zurück
         INC     HL               ;nächstes Zeichen;holt Z. in A
         LD      A,(HL)
         CP      83H
         JP      Z,ØEC55H         ;#SET
         CP      82H
EBAØH    JP      Z,ØEC6DH         ;#RESET
         CP      ØC6H
         JP      Z,ØEC98H         ;#POINT
         CP      ØA2H
         JP      Z,ØECBØH         ;#OPEN
         CP      ØA6H
EBAFH    JP      Z,ØECD8H         ;#CLOSE
         CP      9CH
         JP      Z,ØEDØ9H         ;#LINE
         CP      84H
         JP      Z,ØECE8H         ;#CLS
         CP      ØB8H
         JP      Z,ØEC48          ;#CLEAR
         CP      ØAFH
         JP      Z,ØFE76H         ;?
         CP      ØCBH
         JP      Z,ØEEC7H         ;?
         CP      ØA4H
         JP      Z,ØEEE9H         ;#GET
         CP      ØA9H
         JP      Z,ØEF95H         ;#NAME
         CP      ØA5H
         JP      Z,ØF38AH         ;#PUT
         CP      ØDBH
         JP      Z,ØFBB9H         ;?
         CP      ØAØH
         JP      Z,ØFB96H
         CP      ØF2H
         JP      Z,ØF5EDH
         CP      ØBØH
         JP      Z,ØF64BH
         CP      ØB2H
         JP      ZØF6C5H          ;#PRINT(2)

         usw.
```

Listing 3: Patches im HRG Treiberprogramm

```
EBØØH    LD      HL,EB79H
         LD      (4ØØ4H),HL
         LD      HL,ØDAFFH
         LD      (EB74H),HL
         LD      HL,ØAAFFH
         LD      (EB76H),HL
```

```
       LD     HL,(401EH)
       LD     (0F6B7H),HL
       LD     HL.(EB74H)
       LD     (4049H),HL
       CALL   01C9H            ;CLS-Befehl
       CALL   04FCH            ;Umschalten auf 32 cpl
       LD     HL,3C04H         ;Cursor nach 3C04H
       LD     (4020H),HL
       LD     HL,EB44H         ;HL auf Text ab EB44H
       CALL   4467H            ;Text ausgeben
       LD     HL,EB3DH         ;HL auf DOS Command ab EB3DH
       JP     4419H            ;DOS Command ausführen= KBDHRG
EB36H  LD     A,00H
       OUT    (01H),A          ;HRG einschalten
       JP     0F6E5H           ;HRG löschen und #PRINT(2)
EB3DH  DEFM   'KBDHRG'
       DEFB   0DH
EB44H  DEFM   'HRG-DOS Version 1.6'
       DEFB   0AH
       DEFB   0AH
       DEFM   ' (c) by E.Schröer '89'
       DEFB   0DH

F6F2H  JP     1DEH             ;Steuerung Programmausführung
Wird ersetzt durch:
F6F2H  JP     402DH            ;dadurch Rücksprung zum DOS
```

Nachdem der Tastaturtreiber geladen wurde, wird ein Sprung von diesem nach EB36H ausgeführt und dort die HRG eingeschaltet. Dann wird der normale Bildschirmtreiber vollwertig ersetzt. Alle Ausgaben erfolgen mittels G-Zeichensatz durch die HRG.

Anstatt die Message ab EB44H auszugeben, kann man den freigewordenen Speicherplatz, wie ich eingangs erwähnte, anders nutzen oder Teile des Tastaturtreibers dort unterbringen.

Das Tastaturtreiberprogramm wird als KBDHRG/CMD abgespeichert, HRG/CMD habe ich in HRGDOS/CMD umbenannt.

Für Anregungen, Verbesserungen, Kritik und weitere Möglichkeiten wäre ich dankbar.

(c)1989 Egbert Schröer

Oskar Drechsler

'Grafikpaket' für die HRG 1B-Platine

Vor langer Zeit fand ich in der 'Computer persönlich' das Grafik-Programm Graflix/Bas von Frank Debatin und war natürlich begeistert, denn es gab ja nichts vergleichbares für meine HRG-Platine. Irgendwann ging mir aber der zeitlupenhafte Kopiervorgang von Bildteilen so gewaltig auf den Wecker, daß ich auf Abhilfe sann. Da ich den einzigen Ausweg in Assembler sah, nahm ich mir erstmal das Z80-Handbuch von Rodney Zaks vor, das schon Jahre im Schrank rumlag. Über die ersten 30 Seiten kam ich nie hinaus. Aber nun hatte ich ein Ziel und biß mich durch.

Dann disassemblierte ich HRG/CMD und versuchte die Steuerung der Platine herauszufinden. Als ich das kapiert hatte, war mein Kopf plötzlich so voller Ideen, nach deren Verwirklichung das ursprüngliche Graflix/Bas nur noch als Rumpf-Programm anzusehen ist.

Bei der Beschreibung des Programmes gehe ich mal davon aus, daß das Original-Programm und die Systematik der Funktionen und der Tastenbenutzung bekannt ist.

Das Menü sieht wie folgt aus:

1 - Neue Grafik erstellen
2 - Momentane Grafik ändern
3 - Bildschirm auf Diskette
4 - Diskette auf Bildschirm
5 - Diskette mit Bildschirm mischen

6 - Grafikmuster neu initialisieren
7 - Alle Grafikmuster zeigen
8 - Bildschirm aus GEAP übernehmen

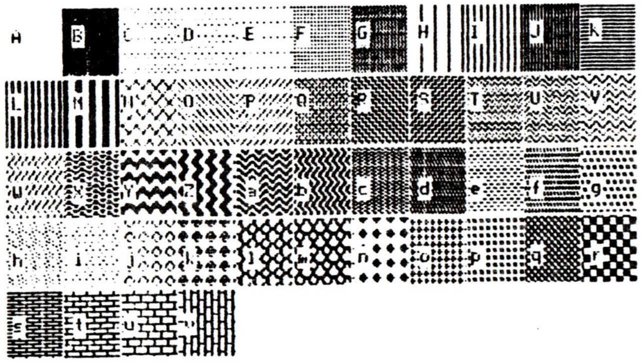
9 - Programmende

Die Punkte 1 und 2 sind identisch mit dem Originalprogramm.

Bei Punkt 3 - 5 wird erst das Directory von Laufwerk 1 mit Extension /DIS gezeigt, und dann nach dem Filenamen gefragt, der gespeichert, geladen oder gemischt werden soll. Da beim HRG/CMD von RB-Electronik das LOAD, SAVE oder MERGE in Verbindung mit einem String als Filenamen nicht immer funktioniert, gibt es dazu noch eine Fehlerbehandlungsroutine. Mehr gibt's dazu nicht zu sagen.

Punkt 6: Im Programm sind einige Grafikmuster (Raster) fest installiert, die immer an die Cursorposition angepaßt werden und deshalb immer aneinanderpassen (Tun sie auch in 98 % aller Fälle. Warum die 2 % spinnen, habe ich noch nicht herausgefunden). Wenn der von mir vorgegebene Musteranfang nicht paßt, kann man ihn verschieben. Wenn man vor lauter Verschieberei nicht mehr durchblickt, stellt Punkt 6 den Grundzustand wieder her.

Punkt 7 zeigt alle gespeicherten Grafikmuster und die Buchstaben, mit denen sie abgerufen werden können:



Punkt 8: Nähere Erläuterungen sind aus dem Programm-Listing ab Zeilen-Nr. 8000 ersichtlich. Wer GEAP hat, weiß sicherlich was ich meine.

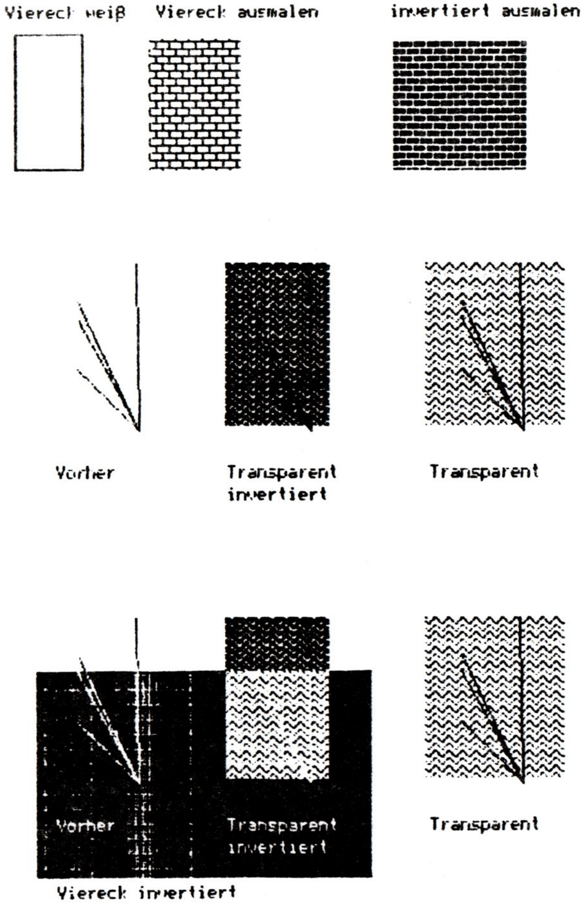
Dem Basic-Grundprogramm habe ich versucht eine Struktur zu geben und durch eine geordnete Variablentabelle und mir bekannte Tricks etwas Beine zu machen. Alles was trotzdem noch langweilig war (Bildteile kopieren, Flächen mit Muster ausfüllen) habe ich in 'Maschine' umgeschrieben.

Nun zu den einzelnen Funktionen und Möglichkeiten:

Folgende Funktionen habe ich nicht verändert:
- (A)uslöschen Punkt
- (P)unkt setzen
- (M)arkiere Punkt
- (E)ingabe Cursorkoordinaten
- (G)rafikausschnitt (markiert) festlegen (etwas erweitert)
- (H)ilfe
- (I)nvertiere den gesamten Bildschirm
- (K)reis: (W)eiß (S)chwarz (A)usmalen (L)öschen (I)nvertieren
- (L)inie: (W)eiß (S)chwarz (I)nvertieren

ergänzt wurde:
- (V)iereck: (W)eiß nur Rand
  (S)chwarz nur Rand
  (L)öschen ganzes Viereck
  (I)nvertieren ganzes Viereck
  (i)nvertiert nur den Rand
  (A)usmalen
  (T)ransparent ausmalen
  (t)ransparent invertiert ausmalen
  (a)ausmalen invertiert
  wobei nach der Wahl der Ausmal-Funktion immer noch die Frage nach dem Muster kommt

Zum besseren Verständnis gebe ich auf der folgenden Seite einige Beispiele:



Kopieren von Bildteilen

Zuerst wird wie gehabt das zu kopierende Feld unten rechts markiert und der Cursor dann an die linke obere Ecke gestellt und dann 'G' gedrückt.

Dann erscheint folgendes:

Für Bildteile kopieren ohne Überlappung: (des Zielfeldes)

seitengleich	: Cursor in linke obere Ecke,	dann O und c
nach links kippen	: Cursor in rechte obere Ecke,	dann O und l
nach oben kippen	: Cursor in linke untere Ecke,	dann O und o
nach ob/li kippen	: Cursor in rechte untere Ecke,	dann O und g

Bei transparent kopieren, anstatt O ein T drücken
Bei invers kopieren, anstatt O ein i drücken
Bei inv./trans. kopieren, anstatt O ein t drücken

<Taste> drücken, damit das Bild wieder erscheint !

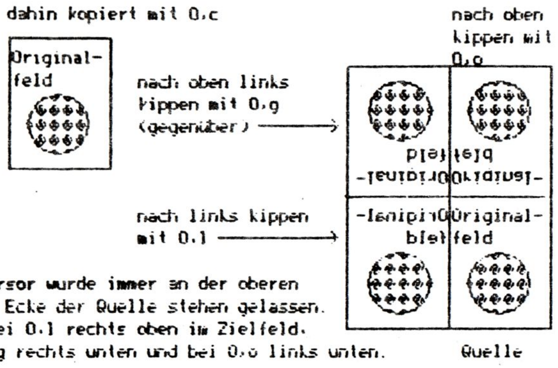
Bei der Buchstabenbenutzung ging ich von folgender Überlegung aus:

O wie Original

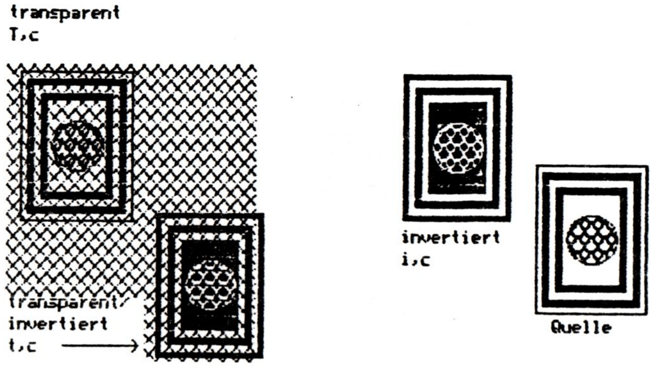
c = copieren wie Originalfeld
l = links für nach links kippen
o = oben für nach oben kippen
g = gegenüber für nach oben links kippen

Die Anfangsbuchstaben der anderen Kommandos sind wohl klar. Die Tasten sind nacheinander zu drücken.

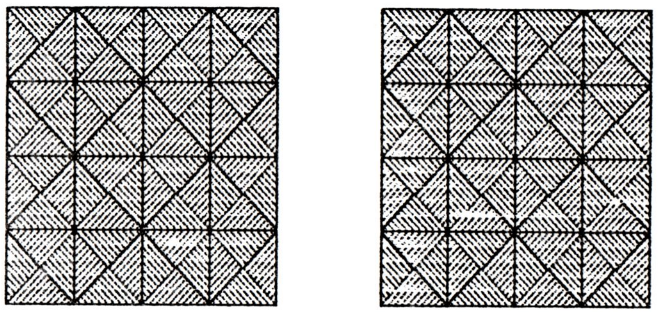
Nun wieder einige Beispiele:

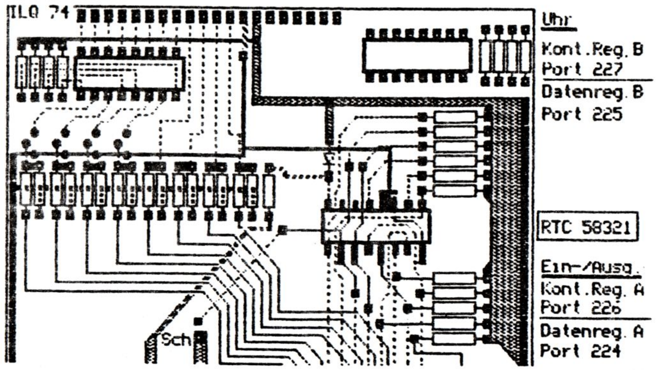


Der Cursor wurde immer an der oberen linken Ecke der Quelle stehen gelassen. d.h. bei O,l rechts oben im Zielfeld, bei O,g rechts unten und bei O,o links unten.



Dieses Feld dahin kopiert dauert 25 sec





HEFT
29
Januar
1990

Die letzte Hardcopy zeigt das 'Teil-Layout' meiner Hardware-Uhr. Es ist für eine 2-seitige Leiterplatte gedacht. Die durchgezogenen Linien liegen auf der Bestückungsseite. Wenn man z.B. mehrere Widerstände schön in eine Reihe haben will, braucht man nur einen zu zeichnen und zu markieren und dann den Cursor nur um festgelegte Punkte weiterzuschieben und immer O,c zu drücken und fertig ist der Kram.
Bei meinem Drucker paßt zufällig auch das Rastermaß 2.54 mm. Deshalb habe ich die Uhr komplett am Bildschirm entworfen. Da er nicht ganz ausreichte, habe ich alles auf 2 Bildschirme verteilt und dann nahtlos aneinandergedruckt. Vorher habe ich bei dem Layout für die Oberseite die unterbrochenen Linien gelöscht und beim Layout für die Unterseite die durchgezogenen Linien. Anschließend habe ich die Hardcopy nachgeschwärzt, alles dreifach auf Folie kopiert, diese aufeinandergelegt und die Platine belichtet. Man muß sehr genau arbeiten, aber es geht.

Nun noch ein Wort zum kopieren, wenn sich das Quellfeld und das Zielfeld überlappt. Man kann nur mit Überlappung kopieren, wenn das Zielfeld über oder/und links vom Quellfeld stehen soll. Das liegt daran, daß das Quellfeld von oben links nach unten rechts Punkt für Punkt auf das Zielfeld übertragen wird. Außerdem ist noch zu sagen, daß beim Zielfeld kein Bildschirmrand erkannt wird. Das habe ich aus Gründen der Verarbeitungsgeschwindigkeit weggelassen. Wird das nicht beachtet, hängt das Programm sich zwar nicht auf, aber das Bild wird an einer falschen Stelle weitergemalt.

Nun weitere Funktionen. Dazu muß ich vorausschicken, daß ich den Artikel mit Genie-Text schreibe. Hier sind Doppelkreuz, Dollarzeichen, größer-als und kleiner-als-Zeichen Steuerzeichen und somit nicht darstellbar.

Größer als : kopiert Bildschirm in den 2. Speicher
Kleiner als: kopiert den 2. Speicher in den Bildschirm

(m)ischen Bildschirm mit 2. Speicher
(w)echseln Bildschirm mit 2. Speicher (vertauschen)

(ab Cursorposition waagrecht schreiben mit G-Zeichensatz
) ab Cursorposition senkrecht schreiben mit G-Zeichensatz

Doppelkreuz: wie (, nur K-Zeichensatz
Dollarzeichen: wie), nur K-Zeichensatz

Hardcopy vom Bildschirm:

Club 80
l
u
b

8
0

Aus diesen Funktionen kommt man mit dem Klammeraffen wieder heraus. Geschrieben wird transparent.
Der Cursor wandert mit der Schrift nicht mit, sondern steht nach Drücken des Klammeraffen wieder an der alten Stelle.

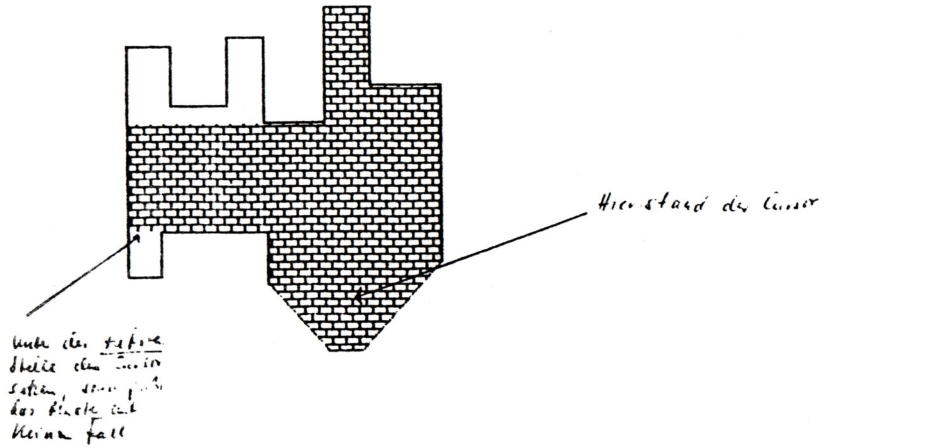
Nun die neu hinzugefügten Funktionen:
****

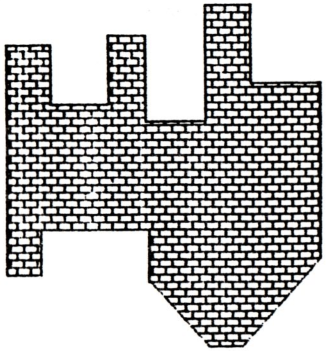
(U)nregelmäßiges Feld mit Muster ausfüllen

Hier mußte ich leider einige Einschränkungen machen. Bei der Architektur dieser Grafik-Karte ist das Ansprechen eines Punktes schon recht zeitraubend. Wenn man nun noch ein Muster bei der Cursorbewegung mitrotieren läßt, die Begrenzung eines Feldes erkannt werden muß und wenn diese vergessen wird, auch der Bildschirmrand erkannt werden soll, kommt man schnell an einen Punkt, wo auch reine 'Maschine' zu langsam wird, wenn man noch mehr will.
Ich bin folgendermaßen vorgegangen: Zuerst wird der Cursor irgendwo in das weiß begrenzte Feld gestellt. Nach drücken von 'U' und Auswahl des Musters beginnt die Suche ab Cursorposition nach oben, bis ein gesetzter Punkt gefunden wird oder der obere Bildschirmrand erreicht ist. Dann wird nach links gesucht, bis ein gesetzter Punkt oder der linke Bildschirmrand erreicht ist. Dabei wird in 'Gedanken' immer das Muster mitrotiert. Sodann wird nach rechts gezeichnet, bis ein gesetzter Punkt gefunden oder der rechte Bildschirmrand erreicht wird. Jetzt erfolgt ein Sprung zu der Stelle, an dem bei der 'Aufwärtssuche' auf 'Linkssuche' umgeschaltet wurde und untersucht, ob eine Zeile tiefer ein Punkt gesetzt oder der untere Bildschirmrand erreicht ist. Wenn nein, beginnt die Suche wieder nach links usw.

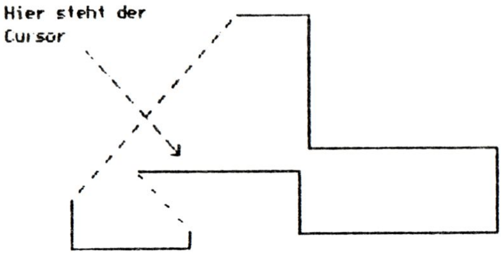
Daraus geht hervor, daß man am besten den Cursor in die Spalte setzt, die am weitesten nach oben und/oder nach unten reicht. Bei Feldern, die Taschen enthalten, die bei der Suche nicht mit erreicht werden, muß man halt nochmal ansetzen. Da das Muster mitrotiert, paßt das Muster bis auf die bereits erwähnten 2 % zusammen.

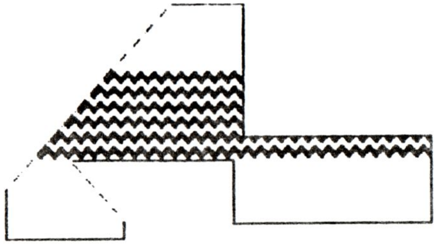
Dazu wieder einige Beispiele:

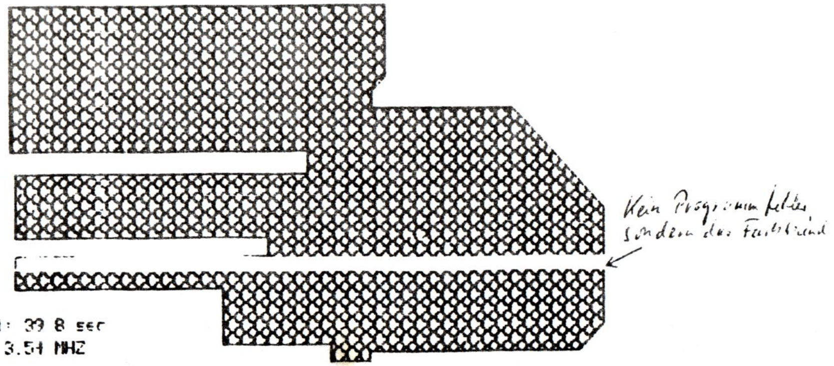




So sieht es fertig aus
Auch das Muster paßt!







Zeit: 39 8 sec
bei 3.54 MHZ

(u)nregelmäßiges Feld weiß ausfüllen

Hier gibt es nur die Einschränkung, daß aus Geschwindigkeitsgründen der Bildschirmrand nicht erkannt wird. Ist das begrenzte Feld irgendwo offen, daß der Cursor den Bildschirmrand erreichen kann, kommt der Cursor irgendwo am Bildschirm wieder zum Vorschein und malt alles weiß, was ihm über den Weg kommt.

Taschen werden erkannt, gespeichert und nachträglich ausgemalt. Und das alles in akzeptabler Geschwindigkeit.

Beispiel:

Die TrocKenlegung des
Mittelmeeres dauert

53 SEC

HEFT
29
Januar
1990

& Dotprint-Zeichen auf dem Bildschirm

Zuerst wird der Cursor 1 Punkt über die Stelle gebracht, an der das 1. Zeichen beginnen soll, dann wird & gedrückt. Anschließend wird das Directory von Laufwerk 1 mit /FON gezeigt, damit man weiß, welche Font's zur Verfügung stehen. Dann wird der Abstand zwischen den Buchstaben in Dot's abgefragt und dann nach dem auszugebenden Wort oder Satz gefragt. Ist der Satz länger als noch Platz in der Zeile am Bildschirm ist, wird er entsprechend abgeschnitten.
Die Buchstaben werden transparent gezeichnet. Mehr gibt's dazu nicht zu sagen.

Bei meinen Arbeiten mir Graflix suchte ich oft mit müden Augen den winzigen Cursorpunkt. Versuche, den Cursor größer zu machen scheiterten an der Verarbeitungsgeschwindigkeit und der genauen Definition des Cursor-Mittelpunktes, der dann invers zum umgebenden Rand sein sollte. Deshalb habe ich in Anlehnung an das GEM meines AT im Büro das 'Gummi-Viereck' erfunden:

g zeichnet kurz ein Viereck um Markierung und Cursorposition

Da die Line-Funktion der HRG bei großen Abstand zwischen Markierung und Cursor zu langsam war, habe ich mich für die Line-Funktion der Block-Grafik entschieden. Hier kommt es ja nicht auf Genauigkeit an, sondern nur um den Cursor zu finden oder eine Vorstellung zu haben, wie groß ein zu zeichnendes Viereck wird. Das geht natürlich nur, wenn man im NEWDOS das Sys29/Sys hat. Auf der Diskette, die ich Oliver schicken werde, ist alles notwendige drauf.

Z zeichnet bei jeder Cursorbewegung kurz ein 'Gummi-Viereck'
  wie oben beschrieben. Z hebt das wieder auf
----

l Muster um 1 Punkt nach links rotieren
----

r Muster um 1 Punkt nach rechts rotieren
----

a Muster um 1 Punkt abwärts rotieren
----

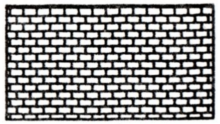
o Muster um 1 Punkt nach oben rotieren

Diese o. a. Funktionen verschieben die fest eingespeicherten Muster. Bei der Installierung der Muster wollte ich erreichen, daß beim Ausfüllen verschiedener Flächen mit gleichem Muster diese zueinanderpassen, wenn diese Flächen überlappen oder einander berühren. Das hat zur Folge, daß die Muster quasi fest am Bildschirm stehen. Da das nicht immer gut aussieht oder nicht paßt, muß man die Muster verschieben können.

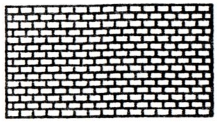
Beispiel: Will man ein Haus mit Ziegelmuster zeichnen, kann es vorkommen, daß das Muster nicht zur Hausecke paßt. Die Ziegel müssen ja im Verbund um die Hausecke gehen.

Hierzu wieder ein Beispiel:

So wirkt das nicht gut



2 x nach links und 1 x nach oben und nochmal V,A,u und schon sieht das besser aus.



Dann habe ich noch die NAME-Funktion eingebaut, die mir aber nicht viel bringt. Deshalb habe ich auch keine Lust sie zu erläutern. Die 'Hilfe' erläutert das aber ausreichend. Die (H)ilfe besteht übrigens aus 4 Bildschirmen. Weiterschalten mit irgendeiner Taste oder mit Klammeraffe Abbruch und Rückkehr zum Grafikbildschirm.

Mit den Pfeiltasten bewegt man den Cursor um einen Punkt, mit SHIFT/Pfeiltasten um 10 Punkte. Will man also ganz an den Rand, darf man die letzten 10 Punkte SHIFT nicht benutzen.

Mit Klammeraffe kommt man zurück zum Hauptmenü

Als die bisher beschriebenen Funktionen liefen, war ich endlich zufrieden und auch etwas stolz auf mich. Fast ein halbes Jahr lang. Doch ganz tief in meiner Computerseele piekste mich immer noch was. Vor allem dann, wenn ich im Büro mit GEM spielte und neidisch den schönen großen still dastehenden Pfeil der Maus sah. Das hat mich fertiggemacht.
Zu allem Unglück fiel mir dann auch noch das türkische Sprichwort ein: 'Wenn Du etwas nicht erreicht hast, hast Du nicht richtig gewollt.'
Fazit: ein maschinengesteuerter Cursor mußte endlich her! Ich will nun nicht alle Irrungen wieder aufwärmen, aber das Resultat ist nach meiner Ansicht das Beste was aus dem Genie I bei 3.54 MHZ herauszuholen ist. Wenn mein geplanter 'Helmut Bernhardt I' fertig ist und ich neben der Hercules-Karte auch noch Lust habe, die HRG 1B-Platine mit anzuschließen und das auch möglich (und sinnvoll) ist, bieten sich sicher wieder neue Möglichkeiten.

+ Sprung von der Basic-Ebene in die Maschinenebene.

Hier erscheint ein rechter Winkel als Cursor. Er ist weiß und blinkt nicht. Die Punkte, die der Cursor abdeckt, werden im RAM zwischengespeichert und nach einer Bewegung wieder zurückgebracht. Die Spitze des rechten Winkels ist der eigentliche Cursorpunkt und blinkt über dem überdeckten Punkt. Wenn der Cursor sich einem Bildschirmrand nähert, wechselt er seine Form, um bis an den Rand heranzukommen. Da bei einer rein weißen Fläche der Cursor nicht sichtbar wäre (außer der winzigen blinkenden Spitze), befindet sich innerhalb des rechten Winkels noch ein negativer Winkel.

Das klingt sicher etwas verwirrend, eine nachstehende Grafik verdeutlicht das aber sicherlich.

Mit SHIFT/Pfeiltasten geht alles etwas schneller. Hier wird aber nicht um 10 Punkte weitergesprungen, sondern eine Zeitschleife wird übersprungen.
Es sind auch Kombinationen der Pfeiltasten möglich, mit denen man sich also schräg vorwärtsbewegen kann. Das ist sehr nützlich bei der Zeichenfunktion, die gleich beschrieben wird. Der Bildschirmrand wird natürlich erkannt.

In dieser Ebene sind folgende Funktionen möglich:

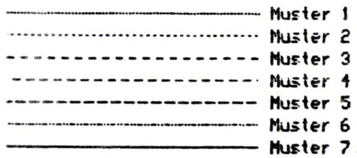
p oder P = Punkt setzen
a oder A = auslöschen Punkt
m oder M = markiert Punkt
Klammeraffe = Rücksprung in Basic-Ebene

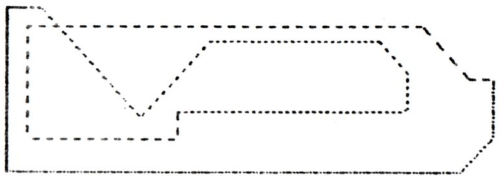
Die Markierung wird übrigens in die Basic-Ebene mit übertragen, wenn man zurückspringt.

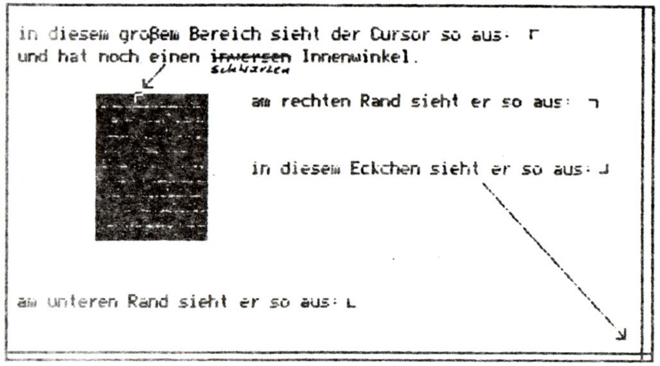
Drückt man die Enter-Taste, erscheint in der rechten oberen Ecke ein '?'. Dann drückt man die Tasten Ø-7.
Ø = löschen mit dem Cursor
1 - 7 = zeichen mit dem Cursor (Punktraster und Strich)

Drückt man anstatt der ENTER-Taste 'T', zeichnet der Cursor transparent.

CLEAR-Taste hebt das Zeichnen/Löschen mit dem Cursor wieder auf.







Die Cursorbewegung ist ohne SHIFT ausreichend langsam, mit SHIFT ausreichend schnell. Geschwindigkeit von links nach rechts:
ohne SHIFT: 19 sec
mit SHIFT: 5 sec
Wem das zu langsam oder zu schnell ist, kann das ändern.
In 6DD2H steht: LD BC,1300H, das ist für die Zeitschleife in 0060H.
In 6DD3H steht also 00H und in 6DD4H steht 13H (19D)
Entweder ändert man den Source-Code von HRGGRA/SRC und assembliert neu oder nimmt in Graflix eine Zeile neu auf und pokt so lange in 6DD4H herum, bis das paßt.

Die Diskette enthält folgendes:

Autostart mit Graflix/JCL. Dieser Chain-File setzt den Basic-Anfang hoch und lädt HRGGra/CMD. HRGGrafl/CMD macht MEM1 und RENEW und läßt Graflix/HRG rennen.

HRGGra/CMD enthält fast alle von mir hinzugefügten Maschinenroutinen und einen Tastaturtreiber. Siehe hierzu HRGGra/SRC.

HRGGrafl/CMD ist das ursprüngliche HRG/CMD. Hier habe ich in der Lprint-Routine (für Epson-kompatible Drucker) die beiden Initialisierungsroutinen vor und nach der Druck-Routine herausgenommen, um Hardcopy's nahtlos aneinanderfügen zu können. Auch habe ich den etwas unschönen G-Zeichensatz durch die Zeichen ersetzt, die der normale Charaktergenerator erzeugt. Da HRG/CMD nur bis FF24H geht, habe ich hier noch ab FF25H die noch fehlenden Maschinen-Routinen angefügt (siehe Grafli1/SRC) und das alles HRGGrafl/CMD genannt.

Graflix/Ori ist das Original-Programm, das mit Mem1 nicht läuft.

Graflix/HRG ist das um die REM's gekürzte Basic-Programm.

GraflixA/CMD ist ein Bildschirm mit einem Vorstellbild.

GraflixD/HRG sind Data-Zeilen mit den Mustern, das gemergt wird und nach Abarbeitung wieder gelöscht wird. (Platz ist knapp)

GraflixH/HRG sind die Hilfe-Zeilen, die bei Bedarf ebenfalls gemergt werden.

XXX/BI ist ein mit GEAP erzeugtes Bild als Test, das mit Menüpunkt 8 geladen werden kann.

Zum Schluß ein Wort zu den fest programmierten Grafikmustern. Wer die ändern oder ergänzen will muß die Systematik kennen.

Ich will mal das Muster 'u', die großen Ziegel erklären:

Die Data-Zeile 61560 lautet: 8.255.4.4,4.255,64,64,64
8 = 8 Byte hoch (höchstens)

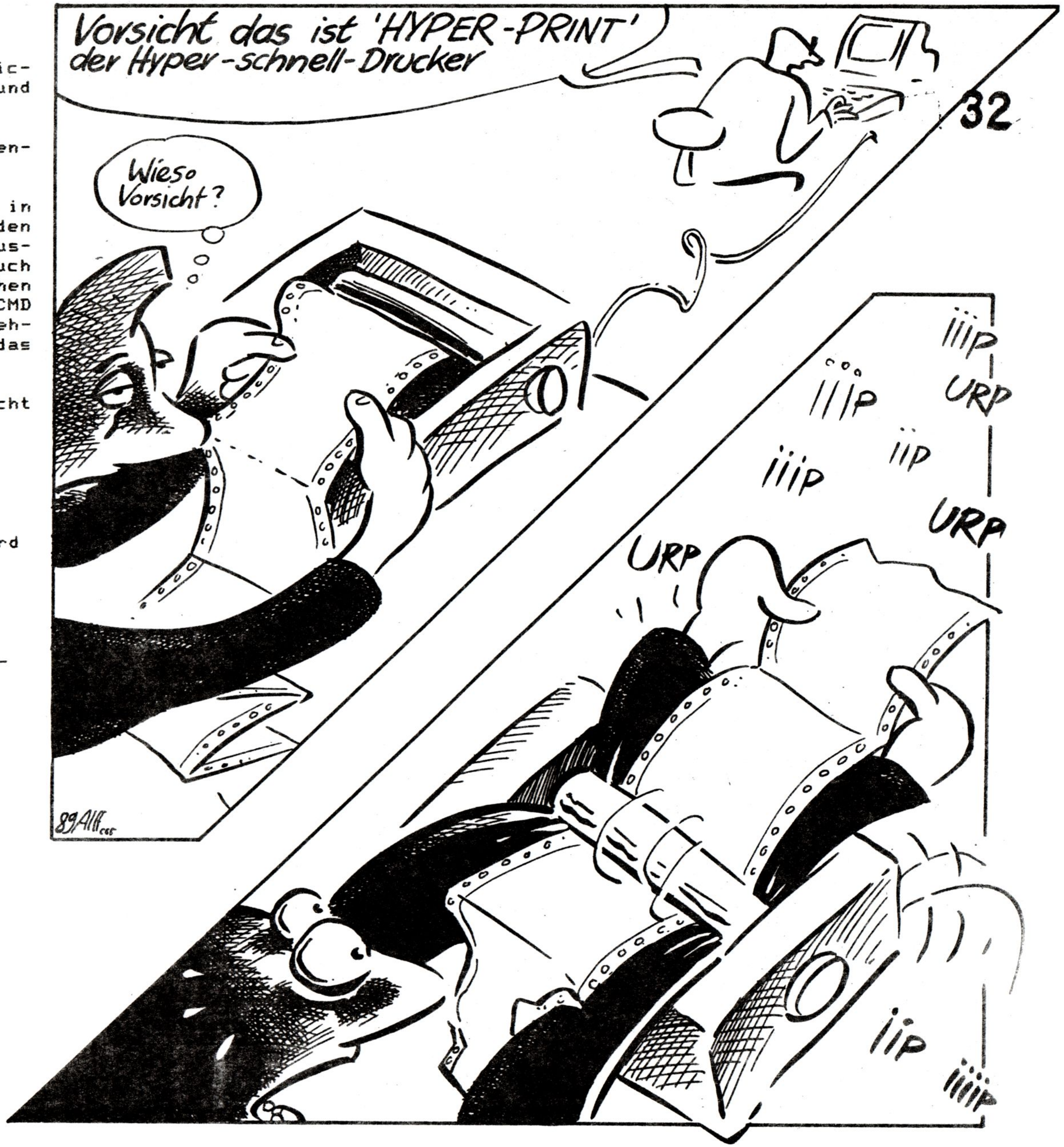
Darstellung der Byte's:

```
!x!x!x!x!x!x!x!x!   255 = 11111111 binär
! ! ! ! ! !x! ! !     4 = 00000100
! ! ! ! ! !x! ! !     4 = 00000100
! ! ! ! ! !x! ! !     4 = 00000100
!x!x!x!x!x!x!x!x!   255 = 11111111
! !x! ! ! ! ! ! !    64 = 01000000
! !x! ! ! ! ! ! !    64 = 01000000
! !x! ! ! ! ! ! !    64 = 01000000
```

oder die Data-Zeile 61170: 4,136,68,34,17 Diagonale
4 = 4 Byte hoch

```
!x! ! !x! ! ! !     136 = 10001000 binär
! !x! ! ! !x! ! !    68 = 01000100
! ! !x! ! ! !x! !    34 = 00100010
! ! ! !x! ! ! !x!    17 = 00010001
```

Wo ein x steht, wird der Punkt gesetzt. Und aneinandergereiht sieht das alles recht gut aus.



Dr. Hans Hehl

Austausch illegaler Z80-Befehle

Ersatzroutinen für die Basic-Interpreter XITAN und HEBAS

In vielen Z80-Programmen werden illegale Befehle verwendet. In Z80-Emulationsprogrammen und bei der neuen CPU 64180 können diese Befehle jedoch nicht verwendet werden. Unser Autor zeigt hier am Beispiel der Basic-Interpreter XITAN und HEBAS, wie diese illegalen Z80-Befehle ersetzt werden können. Ebenso wird ein neuer Basic-Befehl USER für den mc- und den NDR-Klein-Computer vorgestellt.

Im mc 1/82 wurden 422 neue Z80-Befehle veröffentlicht, welche durch eifriges Experimentieren gefunden wurden. Diese Befehle waren jedoch bereits auf einem anderen Kontinent bekannt und wurden dort auch verwendet. Damit stehen 1116 Befehle der Z80-CPU zur Verfügung. Besondere Bedeutung erlangen die zusätzlichen Befehle für die einzelnen Hälften der Indexregister IX und IY. Diese werden so zu vollwertigen Registerpaaren wie die Register HL, DE, oder BC.

Ein Beispiel soll dies erläutern, genauere Hinweise sind in [1] zu finden. Der 8-Bit-Ladebefehl LD C,H (4C) des HL-Registers lautet für das IX-Register LD C,HX (DD 4C). HX steht für die Bits 8...15 von IX. Damit wird das High-Byte des IX-Registers nach Register C kopiert. Entsprechend gibt es zum Befehl LD C,L (4D) den illegalen Teil LD C,LX (DD 4D), wobei LX für die Bits 0...7 steht. Als Ergänzung zu dem in [1] aufgeführten Artikel seien die beiden illegalen Befehle OR HX (DD B4) und OR LX (DD B5) des IX-Registers nachgetragen.

Assembler und Emulatoren steigen aus

Ein großer Nachteil bei der Verwendung der illegalen Befehle besteht darin, daß Hilfsprogramme wie DDT, DDTZ und viele Assembler diese Befehle nicht beherrschen und meistens nur mit einem erstaunten ??? reagieren. Bei einem Debugger kann man sich mühsam durch Setzen von Breakpoints behelfen. Bei Assemblern müssen die Befehle per Byte-Definition direkt gesetzt werden.

Die günstige Preisentwicklung bei Speicherbausteinen und die Entwicklung von 16-Bit- bzw. 32-Bit-CPUs lassen das Arbeitspferd Z80 zunehmend in den Hintergrund treten.

Doch so mancher Umsteiger von CP/M 2.2 auf CP/M-68k würde seine unter CP/M 2.2 erstellten Basic-Programme gerne weiter verwenden. Dazu gibt es einen Z80-Emulator der Firma Soft-Design [2], der auf jedem CP/M-68k-System ein CP/M-80-System Vers. 2.2 nachbildet [3]. Leider werden aber die illegalen Z80-Befehle vom Emulator nicht akzeptiert. Gleiches gilt für die Nachfolger des Z80, z.B. für die CPU 64180, die 512 KByte direkt adressieren kann.

XITAN und HEBAS werden „akzeptabel“

Bei den Basic-Interpretern XITAN von Neil Colvin und dem verbesserten Nachfolger HEBAS (Franzis-Software-Service und Firma Graf, Kempten) kann man die illegalen Opcodes durch legale ersetzen. Nun laufen auf dem 680XX-NDR-System mit Z80-Emulator beide Interpreter mit den schon vorhandenen Basic-Programmen. Auch die neue CPU-64180-Baugruppe der Fa. Graf macht keine Schwierigkeiten bei den angegebenen Basic-Interpretern.

Ein Nachteil sei nicht verschwiegen. Es können die Speicherbereiche über 64 KByte nicht angesprochen werden, da der Befehl LD (nn),HL (22 XXXX) nur FFFFH als höchste Adresse ansprechen kann. Das gilt auch für CP/M 2.2.

Für die Programmteile des Interpreters, welche die illegalen Opcodes enthalten, wird jeweils die Ersatzroutine angegeben. Da diese mehr Platz benötigen, wurde am Ende des Interpreters Platz reserviert, der von Basic-Programmen nicht überschrieben wird. Der Rücksprung aus der Erweiterung zu gleichen Programmteilen erfolgt zunächst mit Relativsprüngen, dann geht es mit RET (C9) zurück ins Hauptprogramm des Interpreters.

Die Änderungen

Im Rahmen dieses Artikels kann keine genaue Erläuterung aller hier geänderter Programmteile erfolgen. Dies ist einer vollständigen Dokumentation des überarbeiteten Interpreters vorbehalten, die als Kurzfassung (ca. 90 Seiten) z. Zt. auf Diskette vom Autor bezogen [4] und als Dokumentation vom Elektronikladen (Detmold) gedruckt erworben werden kann. Bevor jedoch die aufgezeigten Änderungen (*Bilder 1..16*) durchgeführt werden können, muß sowohl bei XITAN als auch bei HEBAS der Inhalt der Adresse 46BAH in A0 47H umgeändert werden. In Adresse 479FH muß dann der Wert 0 enthalten sein. Die Routine in *Bild 2* verändert Low- und High-Byte der Register BC, DE und IX durch den Befehl RRA. Dabei wird der Inhalt des Akkumulators nach rechts verschoben, wobei der Inhalt des Übertragbits nach Bit 7, der Inhalt von Bit 0 ins Übertragbit geschoben wird.

Bei der Ausgabe einer Zahl muß das interne Format in die ASCII-Darstellung umgewandelt werden (*Bild 5*). Dies geschieht durch wiederholten Abzug der in einer Tabelle abgelegten, elf sedezimalen Konstanten (1 000 000 000 bis 1) von den in den Registern DE, IX und C enthaltenen Werten. Elf Konstanten liegen deshalb vor, da maximal eine elfstellige Zahlenausgabe möglich ist. In Register B wird die ASCII-Zahl, mit 30H beginnend, hochgezählt und dann jeweils in das ASCII-Zahlen-Flag F92H bis F99H

```
alte Folge:                        neue Folge:

28E1  4E     LD C,(HL)             28E1  4E     LD C,(HL)
      DD 69  LD LX,C                     23     INC HL
      23     INC HL                      46     LD B,(HL)
      4E     LD C,(HL)                   C5     PUSH BC
28E6  DD 61  LD HX,C               28E5  DD E1  POP IX
      23     INC HL                      23     INC HL
                                         00     NOP
28E9  4E     LD C,(HL)             28E9  4E     LD C,(HL)
```

Bild 1. Sechs Bytes werden ab der durch das HL-Register angegebenen Speicherstelle in die Register DE, IX und BC kopiert

```
alt:                               neu:

25C3  DD 7C    LD A,HX             25C3  E5        PUSH HL
      1F       RRA                       DD E5     PUSH IX
      DD 67    LD HX,A                   E1        POP HL
      DD 7D    LD A,LX                   CD 40 47  CALL 4740
      1F       RRA                       E1        POP HL
      DD 6F    LD LX,A                   00 00     NOP NOP
25CD  CB 1A    RR D                25CD  CB 1A     RR D

                                   ****************************
                                   4740  CB 1C     RR H
                                         CB 1D     RR L
                                   4744  E5        PUSH HL
                                   4745  DD E1     POP IX
                                   4747  C9        RET
```

Bild 2. Diese Routine verändert Low- und High-Byte der Register BC, DE und IX durch den Befehl RRA

```
alt:                               neu:

26BC  DD 7C    LD A,HX             26BC  E5        PUSH HL
      1F       RRA                       DD  E5    PUSH IX
      DD 67    LD HX,A                   E1        POP HL
      DD 7D    LD A,LX                   CD 40 47  CALL 4740
      1F       RRA                       E1        POP HL
      DD 6F    LD LX,A                   00 00     NOP NOP
26C6  CB 1C    RR H                26C6  CB 1C     RR H
```

Bild 3. Bei der Multiplikationsroutine finden sich ähnliche illegale Opcodes wie in Bild 2. Es kann der gleiche Einsprung bei 4740H verwendet werden

```
alt:                               neu:

2512  DD 4C    LD C,HX             2512  E5        PUSH HL
      DD 7D    LD A,LX                   DD E5     PUSH IX
      DD 67    LD HX,A                   E1        POP HL
      DD 6A    LD LX,D             2516  CD 48 47  CALL 4748
251A  AF       XOR A               2519  E1        POP HL

                                   ************************
                                   4748  4C        LD C,H
                                         7D        LD A,L
                                         67        LD H,A
                                         6A        LD L,D
                                   474C  18 F1     JR 4744
-----------------------------------------------------------
alt:                               neu:

2911  5F       LD E,A              2911  E5        PUSH HL
      DD 67    LD HX,A                   CD 4E 47  CALL 474E
      DD 6F    LD LX,A                   E1        POP HL
2916  B7       OR A                2916  B7        OR A

                                   ************************
                                   474E  5F        LD E,A
                                         67        LD H,A
                                         6F        LD L,A
                                   4751  18 F1     JR 4744
```

Bild 4. Illegale Ladebefehle werden ersetzt

```
alt:                               neu:

4287  DD 7D    LD A,LX             4287  D5        PUSH DE
      9E       SBC A,(HL)                DD E5     PUSH IX
      DD 6F    LD LX,A                   D1        POP DE
      23       INC HL                    7B        LD A,E
      DD 7C    LD A,HX                   9E        SBC A,(HL)
      9E       SBC A,(HL)                5F        LD E,A
      DD 67    LD HX,A             428E  CD 53 47  CALL 4753
4292  23       INC HL                    D1        POP DE
                                   4292  23        INC HL

                                   ************************
                                   4753  23        INC HL
                                         7A        LD A,D
                                         9E        SBC A,(HL)
                                         57        LD D,A
                                         D5        PUSH DE
                                   4758  18 EB     JR 4745
```

Bild 5. Umwandlung des internen Formats in die ASCII-Darstellung

```
alt:                               neu:

25AC  DD 55    LD D,LX             25AC  E5        PUSH HL
      08       EX AF,AF'                 DD E5     PUSH IX
      DD 7C    LD A,HX                   E1        POP HL
      DD 6F    LD LX,A                   55        LD D,L
      08       EX AF,AF'                 6C        LD L,H
      DD 61    LD HX,C             25B2  CD 5A 47  CALL 475A
25B6  0E 00    LD C,0                    E1        POP HL
                                   25B6  0E 00     LD C,0

                                   ************************
                                   475A  61        LD H,C
                                   475B  18 E7     JR 4744
```

Bild 6. Die Halb-Bytes der Register BC, IX, und DE werden nach rechts verschoben

```
alt:                               neu:

2693  DD 55    LD D,LX             2693  E5        PUSH HL
      08       EX AF,AF'                 DD E5     PUSH IX
      DD 7C    LD A,HX                   E1        POP HL
      DD 6F    LD LX,A                   55        LD D,L
      08       EX AF,AF'                 6C        LD L,H
      DD 61    LD HX,C                   CD 5A 47  CALL 475A
      4F       LD C,A                    E1        POP HL
269E  C9       RET                       4F        LD C,A
                                   269E  C9        RET
```

Bild 7. Dieser Abschnitt ist fast identisch mit dem ab 25ACH

```
alt:                               neu:

24F9  DD 9D     SBC A,LX           24F9  D5        PUSH DE
      DD 6F     LD LX,A                  DD E5     PUSH IX
      23        INC HL                   D1        POP DE
      7E        LD A,(HL)                9B        SBC A,E
      DD 9C     SBC A,HX                 5F        LD E,A
      DD 67     LD HX,A                  CD 5D 47  CALL 475D
2503  23        INC HL                   D1        POP DE
                                   2503  23        INC HL

                                   ************************

                                   475D  23        INC HL
                                         7E        LD A,(HL)
                                         9A        SBC A,D
                                         57        LD D,A
                                         D5        PUSH DE
                                   4762  18 E1     JR 4745
```

Bild 8. Von den in den Flags F8AH bis F8EH enthaltenen Werten werden die in den Registern DE, IX und C enthaltenen Werte abgezogen

```
alt:                               neu:

2597  DD 9D     SBC A,LX           2597  D5        PUSH DE
      DD 6F     LD LX,A                  DD E5     PUSH IX
      7D        LD A,L                   D1        POP DE
      DD 9C     SBC A,HX                 9B        SBC A,E
      DD 67     LD HX,A                  CD 64 47  CALL 4764
25A0  7D        LD A,L                   D1        POP DE
                                   25A0  7D        LD A,L

                                   ************************

                                   4764  5F        LD E,A
                                         7D        LD A,L
                                         9A        SBC A,D
                                         57        LD D,A
                                         D5        PUSH DE
                                   4769  18 DA     JR 4745
```

Bild 9. Ersatz eines illegalen Subtraktions- und Ladebefehls

```
Test auf den Wert FF bei den Registern DE, IX und BC

alt:                               neu:

2560  DD 2C     INC LX             2560  E5        PUSH HL
      C0        RET NZ                   CD 6B 47  CALL 476B
      DD 24     INC HX                   E1        POP HL
2565  C0        RET NZ             2565  C0        RET NZ

                                   ************************

                                   476B  DD E5     PUSH IX
                                         E1        POP HL
                                         2C        INC L
                                         E5        PUSH HL
                                         DD E1     POP IX
                                         C0        RET NZ
                                         24        INC H
                                   4774  18 CE     JR 4744
```

Bild 10. Test auf den Wert FFH bei den Registern DE, IX und BC

```
alt:                               neu:

2578  DD 8D     ADC A,LX           2578  D5        PUSH DE
      DD 6F     LD LX,A                  DD E5     PUSH IX
      23        INC HL                   D1        POP DE
      7E        LD A,(HL)                8D        ADC A,E
      DD 8C     ADC A,HX                 5F        LD E,A
      DD 67     LD HX,A                  CD 76 47  CALL 4776
2582  23        INC HL                   D1        POP DE
                                   2582  23        INC HL

                                   ************************

                                   4776  23        INC HL
                                         7E        LD A,(HL)
                                         8A        ADC A,D
                                         57        LD D,A
                                         D5        PUSH DE
                                   477B  18 C8     JR 4745
```

Bild 11. Hier werden die Werte der Flags F8AH bis F8DH zu den Werten der Register DE, IX und C addiert

abgelegt. Dann kann diese Zeichenkette ausgegeben werden.
In *Bild 8* werden von den Werten der Flags F8AH bis F8EH die Werte der Register DE, IX und C abgezogen und die Ergebnisse wieder in den Registern DE, IX und C abgelegt. Das HL-Register dient als Zeiger auf die Flags F8AH bis F8EH. Das Register B wird gesondert behandelt.
In diesem Zusammenhang sei auf das Buch von Röckrath aus der Serie Computerpraxis des Franzis-Verlages [5] verwiesen, das grundsätzliche Erläuterungen der Arbeitsweise von Basic-Interpretern enthält.

Eigene Basic-Erweiterungen

Allerdings soll kurz auf die Möglichkeit eingegangen werden, wie eigene Erweiterungen in Maschinensprache im Interpreter resident abgelegt werden können. Es kann der Speicherbereich nach den Systemmeldungen des Kaltstarts Verwendung finden, der aber nicht von Basic-Programmen überschrieben werden darf. Die Anfangsadresse eines Basic-Programmes steht in Adresse 46BAH. Bei XITAN steht dort 45D8H, bei HEBAS je nach Version eine höhere Adresse, z.B. 4740H. Setzt man bei Adresse 46BAH z.B. den Wert 4900H ein, so werden ab dieser Adresse die Basic-Programme abgelegt. Natürlich verringert sich dadurch der zur Verfügung stehende Speicherplatz. Wichtig ist, daß das Byte vor dem ersten Byte des Basic-Programmes den Wert 0 besitzt. Ein Ausführungsbeispiel für einen neuen Basic-Befehl USER soll dies verdeutlichen.

USER-Befehl unter XITAN- und HEBAS-Basic

Der unter CP/M zur Verfügung stehende USER-Befehl kann mit folgender Änderung auch in Basic anstelle des selten benötigten Basic-Befehls USR als Direktbefehl eingerichtet werden. Der USER-Befehl ermöglicht die Zusammenfassung bestimmter Dateigruppen, z. B. nur Systemprogramme oder nur Basic-Programme in einem eigenen Bereich des Inhaltsverzeichnisses. Sie erscheinen dann beim DIR-Befehl nur nach Umschaltung auf den gewählten User-Bereich. Damit bei beiden Interpretern mit dem DIR-Befehl alle Dateien mit beliebigen Dateizusatz angezeigt werden, müssen ab Adresse 35B9H die drei Bytes 42H, 41H und 53H jeweils durch 3FH ersetzt werden (alte Maske ????????BAS, neue Maske ??????????? beim DIR-Kommando).

Damit Programme auf eine höhere USER-Ebene kopiert werden können, muß das Kopierprogramm PIP.COM auf der gewählten USER-Ebene vorhanden sein. Dies kann mit DDT.COM geschehen, wie in [6] beschrieben und in *Bild 15* gezeigt ist. Bei Verwendung von Micro-Shell oder anderen komfortablen Dienstprogrammen entfällt der Vorgang, da nach dem zu kopierenden Programm auch auf anderen USER-Bereichen gesucht wird.
Der neue Basic-Befehl USER verwendet nur die USER-Bereiche 0 bis 9. Die Umschaltung erfolgt z.B. für den Bereich 5 mit CALL USER(5). Der aktuelle USER-Bereich kann mit CALL USER(X) abgefragt werden, es wird dann die Zahl ausgegeben. Ein Basic-Programm wird mit LOAD von der eingeschalteten USER-Ebene geladen. Dann wird mit CALL USER(5) z.B. der 5. USER-Bereich aufgerufen und mit SAVE das Programm dort abgespeichert. *Bild 16* zeigt die Änderungen für beide Interpreter.
Bei den 5 1/4-Zoll Laufwerken mit 80 Spuren und etwa 780 KByte Speicherkapazität wird das Directory doch recht unübersichtlich. Unter Basic mit dem USER-Befehl kann man es nun in bis zu 10 Teile aufspalten. So kann man die Systemprogramme auf dem User 0 belassen und Basic-Programme auf höhere User-Ebenen verlagern.
Hebas gibt es inzwischen auch in einer Version für CP/M 3.

Literatur

[1] *Haßelberg, M.:* 422 neue Z80-Befehle, mc 1982, Heft 1, Seite 27.
[2] *Thies, H.:* Z80-Emulator für 680XX-Systemen, Soft-Design 8 München 19, Hedwigstr. 3.
[3] *Bittner, F.:* CP/M auf dem mc-68000-Computer, mc 1985, Heft 2, Seite 89.
[4] *Hehl, H.:* LOOP, Zeitung für Computerbauer, G. Graf GES GmbH, Kempten.
[5] *Röckrath, L.:* Microsoft-Basic, Konzepte, Algorithmen, Datenstrukturen, Franzis-Verlag.
[6] *Plate, J.:* Betriebssystem CP/M, Franzis-Verlag.


```
alt:                               neu:

2771  B3        OR E               2771  E5        PUSH HL.
      DD B4     OR HX                    CD 7D 47  CALL 477D
      DD B5     OR LX                    E1        POP HL
2776  20 9F     JR NZ, 2717        2776  20 9F     JR NZ, 2717
                                   **************************
                                   477D  B3        OR E
                                         DD E5     PUSH IX
                                         E1        POP HL
                                         B4        OR H
                                         B5        OR L
                                   4783  18 BF     JR 4744
```

Bild 12. Die Veroderungen des High- und des Low-Bytes des IX-Registers werden durch legale Befehle ersetzt

```
alt:                               neu:

2875  DD 7C     LD A,HX            2875  D5        PUSH DE
      BE        CP (HL)                  DD E5     PUSH IX
      C0        RET NZ                   D1        POP DE
      2B        DEC HL                   CD 85 47  CALL 4785
      DD 7D     LD A,LX                  D1        POP DE
      BE        CP (HL)            287D  C0        RET NZ
287D  C0        RET NZ
                                   ************************
                                   4785  7A        LD A,D
                                         BE        CP (HL)
                                         C0        RET NZ
                                         2B        DEC HL
                                         7B        LD A,E
                                         BE        CP (HL)
                                   478B  C9        RET
```

Bild 13. Die Werte der Register BC, IX und DE werden mit denen in den Flags F8FH bis F8AH verglichen.

```
alt:                               neu:

292A  DD 2B     DEC IX             292A  E5        PUSH HL
      DD 7C     LD A,HX                  CD 8C 47  CALL 478C
      DD A5     AND LX                   E1        POP HL
2930  3C        INC A                    00        NOP
                                   2930  3C        INC A
                                   **********************
                                   478C  DD 2B     DEC IX
                                         DD E5     PUSH IX
                                         E1        POP HL
                                         7C        LD A,H
                                         A5        AND L
                                   4793  C9        RET
```

Bild 14. Illegale Opcodes von Logikbefehlen werden ersetzt

```
Befehl:                 Hinweise:

DDT PIP.COM             PIP.COM mit DDT.COM geladen
                        unter NEXT-Anzeige steht 1E00
                        (1E00 = 1*16 + 14 = 30 pages
                        zu je 256 Byte)

G 0                     DDT verlassen
USER 3                  gewünschter USER-Bereich
SAVE 30 PIP.COM         PIP.COM auf USER-Bereich ablegen

Kopieren von Programmen auf den gewählten User-Bereich:

PIP a: = a:DAT.BAS ÄGOVÜ     Ä bzw. Ü = eckige Klammer

Programm DAT.BAS wird aus User 0 nach User 3 kopiert (glei-
ches Laufwerk).
```

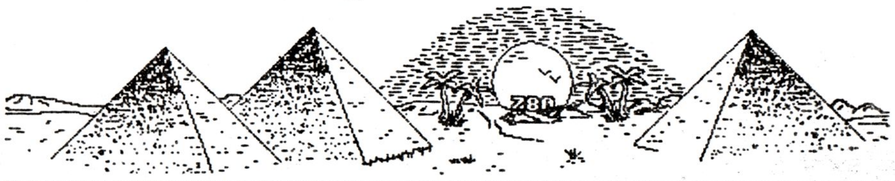
Bild 15. Mit dem Debugger DDT.COM wird ein Kopierprogramm auf eine höhere USER-Ebene unter CP/M 2.2 gebracht

Adresse	altes Byte	neues Byte
116F	46	13
1170	01	47
1C71	52	45
1C72	20	52
46BA	D8	A0
46BB	45	47

```
bei HEBAS: folgende Bytes werden ab Adresse 4713 eingegeben:

4713    21 58 11 0E 20 7E FE 28 C2 48 20 23 CD
4720    C7*44 D6 30 FE 28 28 0B FE 0A D2 48 20 5F CD 05
4730    00 18 DD 1E FF CD 05 00 C6 30 CD 94 32 18 D1 00

bei XITAN: gleiche Eingabe, es ändern sich nur die zwei
           unterstrichenen Bytes:

4732:  C4   und   473E:  B8

also: CALL USER(X): Ausgabe des aktuellen User-Bereiches
      CALL USER(9): Umschaltung z.B. in User-Bereich Nr. 9
```

Bild 16. Änderungen für den neuen Basic-Befehl USER bei XITAN und HEBAS

CLUB 80 HARD - - - CLUB 80 HARD



Kurt Müller Sophie-Scholl-Ring 3B 2054 Geesthacht CLUB 80

Ein Versuch ...

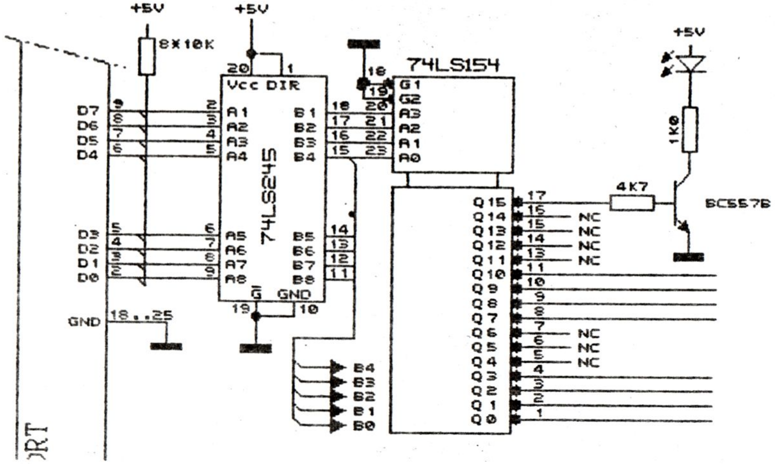
Hallo Clubfreunde,
das Nordlicht-Treffen ist wieder einmal vorüber. Sicher steht am Anfang des Infos der Tagungsbericht, aus dem alles zu entnehmen ist. Meine Absicht ist es, das von mir auf dem Treffen vorgestellte Analoginterface an dieser Stelle noch einmal näher zu beschreiben. Diesmal aber so, daß auch die weniger mit Lötzinn Bekleckerten es verstehen können.

Kommen wir zum Thema. Bevor ich mich in Einzelheiten vergrabe, als erstes ein paar Worte zum Sinn der Übung. Entstanden ist die ganze Sache aus dem Frust heraus, einen tollen Composter unter den Händen zu haben, der aber zu nichts anderem benutzt wird, außer zum Schreiben von Briefen. Das erschien mir dann doch zu wenig. Um diesem Mißstand abzuhelfen, habe ich mir ein Analoginterface mit zwei analogen Eingängen (12 Bit Auflösung) und einem 5-bit breiten Datenbus (nur Ausgabe, warum s. u.) zusammengebaut. Das die hier beschriebene Schaltung nur eine Auflösung von 8 Bit hat, liegt einfach am Bauteilpreis. Außerdem bereitet ein Nachbau bei 8 Bit Auflösung weniger Probleme.

Ein Einbau ***in*** den Rechner scheidet schon wegen der notwendigen Dekodierlogik und der ungewissen Frage nach dem Adress- bzw. Portbereich aus. Außerdem ist es nicht jedermanns Sache alles in seinem Compi fest zu installieren, zumal selbst das größte Gehäuse nur begrenzt Platz bietet. Von den vielfältigen Möglichkeiten, sein liebstes Stück beim Einbau in die Ewigen Jagdgründe schicken, einmal ganz zu Schweigen. Was also von Außerhalb kommt, muß also eine reguläre Buchse benutzen. In den meisten Fällen gibt es an jeden Rechner einen Druckeranschluß, "Centronic" genannt. Dieser hat aber häufig deutlich weniger als 8 Eingangsleitungen. Ein Genie I z.B. vier und bei meiner Kiste sogar nur einer (!). Echt sparsam, oder ? Damit ist die von mir eingeschlagene Richtung eigentlich beschrieben. Bleibt noch der Unkostenbeitrag auch "Spaßsteuer" genannt (unabhängig davon, ob die Aktion in Rauch aufgeht oder nicht), zu nennen. Deutlichen Einfluß auf die Rechnugshöhe hat hier die eigene Bastelkiste, wer aus ihr Platinen, Netztrafo, A/D-Wandler und ein Gehäuse zocken kann, spart spürbar (gute ²/₃ vom Gesamtpreis). Der Kleinkram wie Sockel, Widerstände, Kondensatoren, TTL-IC's und die unvermeidlichen Buchsen und Stecker vereinen den Rest auf sich. Wer keinen Fundus hat, wird für die 8-bit Lösung ca. DM 100.-- an Materialkosten locker machen müssen. der Zeitaufwand für den Zusammenbau des Gerätes liegt bei etwa 20 Std. Die Ehrenrunden für die messtechnische Erforschung einiger Verdrahtungsfehler einmal nicht mitgerechnet. Die meiste Zeit geht in das Verdrahten der Bauteile. Ich habe mich hier einem Gemisch aus Drahtverbindungen und Fädeltechnik bedient. Der Analogteil ist normal verdrahtet (Analogschaltungen sind zum fädeln ungeeignet), die digitalen Teile sind gefädelt, da die Leitungen anders nicht unterzubringen sind. Als Platine habe ich eine Version mit Squarepads auf der Unterseite und einer Massefläche auf der Oberseite verwendet. Diese Kupferfläche ist wichtig, garantiert sie doch eine saubere Masse (man hat sie immer genau dort wo sie auch benötigt wird !). Hersteller dieser Platinen ist die Fa. Bicc-Vero. Sie kostet mit ca. DM 16.-- gutes Geld, aber ich kann sie nur empfehlen, die Investition lohnt sich.

... mit der Elektronik

Damit ist es nun soweit, ich nähere mich der Schaltungsbeschreibung. Den Vorstoß wagt als erstes der Befehlsdekoder. Er besteht aus dem 74LS245 als Buffer und Sabotageschutz. Sollte nämlich jemand einmal einen Kurzschluß auf den Digitalleitungen zustande bringen, erwischt es nur dieses IC und nicht den Drucker-Port im Rechner. Daran anschließend folgt ein 74LS154, als Befehlsdekoder. Er macht nichts anderes, als die oberen 4 Bit (D7...D4) einer 4-zu-16 Umwandlung zu unterziehen und mit der richtigen Ausgangsleitung dann die zugehörige Aktion zu starten. Diese sind den Sample and Hold zu steuern, den A/D-Wandler zu starten, das Schieberegister mit dem Ergebnis der Wandlung zu laden und den Wert unter Anleitung der CPU seriell auszulesen. Außerdem wird noch ein Monoflop zum erzeugen eines Schreibclocks für den 5-bit Datenbus getriggert. Die übrigen Ausgänge werden nicht benutzt.



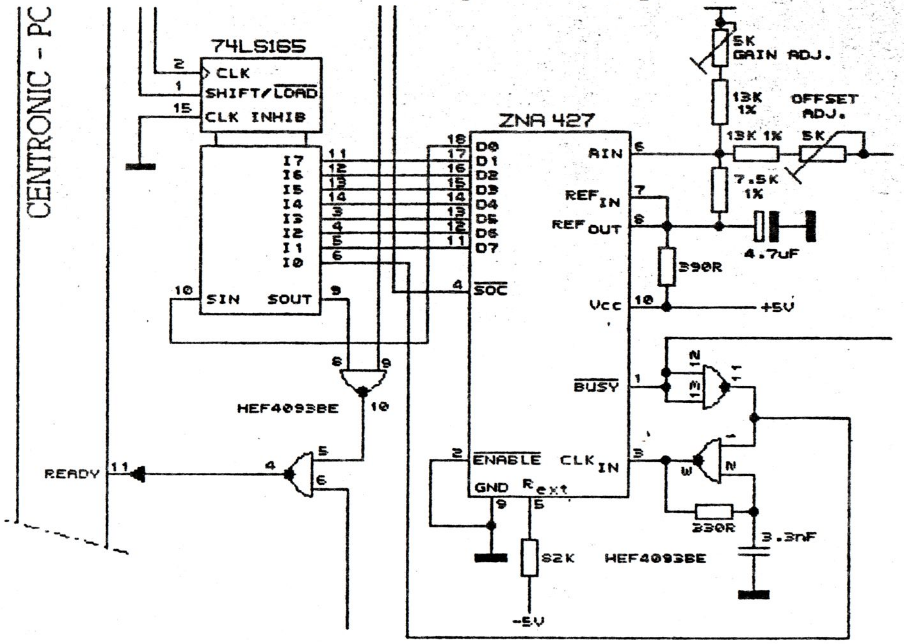
Wem es noch nicht aufgefallen ist, der sei schon hier darauf gestoßen, es wird so ziemlich alles zu Fuß, d.h. durch die CPU per Programm erledigt. Eine Ablaufsteuerung, die alles automatisch managed, ist wegen des unnötigen Aufwandes (=Geld) nicht vorgesehen. Dafür hat aber der Programmierer auf alles Einfluß und kann sich seine Ablaufsteuerung ausdenken.

Die Leitung Q15 ist die BUSY-Anzeige des Interface, d.h. solange dieser Ausgang HIGH ist (=Interface angesprochen), leuchtet die LED. Es läßt sich so leichter erfassen, wann und ob überhaupt vom Rechner auf das Interface zugegriffen wird. Eine besondere Stellung hat D4. Diese Leitung geht gleichzeitig mit D3...D0 auf den externen Datenbus. D4 ist von mir dazu ausersehen worden, als

HEFT 29 Januar 1990

Statusleitung zu dienen. Ist sie HIGH (=1), so liegt ein 4-bit Kommando an, mit dem die externe Hardware gesteuert werden kann. Ist es LOW (=0), so übertragen D3...D0 lediglich Daten, die verarbeitet werden sollen. Diese Vorgehensweise ist mit dem OP-Code der Z80 CPU vergleichbar, der auch aus einem Befehls- und einem Operandenteil besteht.

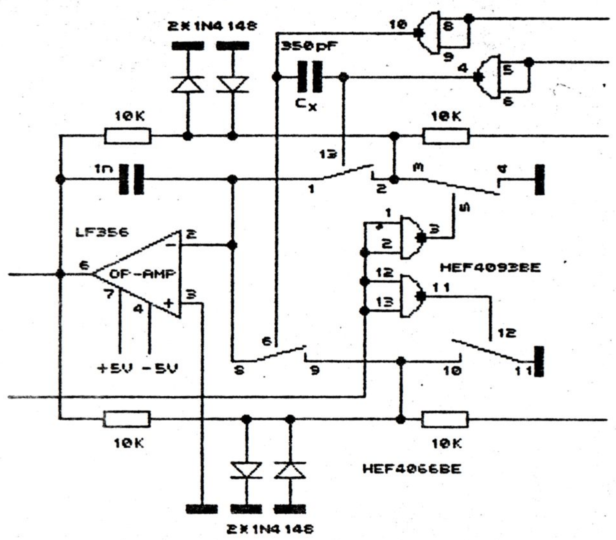
Der Dreh- und Angelpunkt der ganzen Sache kommt jetzt als A/D-Wandler mit Schieberegister. Wie ich schon weiter oben angedeutet habe, hat nicht jeder Centronic-Port acht Eingangsleitungen. Um diesem Problem abzuhelfen, benutze ich die BUSY-Leitung zum Einlesen der Daten, denn die gibt es immer. Aus dem Grund auch das Schieberegister. Eine Baudrate gibt es nicht, die Auslesegeschwindigkeit ergibt sich unmittelbar aus der Arbeitsgeschwindigkeit der CPU bzw. aus der Schnelligkeit des Programms.



Der Pin 'SOUT' am Schieberegister wird schlauer Weise über ein UND-Gatter auf die READY-Leitung geführt. So ist es problemlos möglich, bei Bedarf von Außen auf den READY-Pin ein Signal zu schalten. Damit aber in jedem Fall das externe Signal gelesen werden kann, wird über ein zusätzliches Gatter eine eventuelle '1' am Ausgang 'SOUT' überschreibbar gehalten.

Auffallend ist auch der Poti-Kneuel am Eingang 'AIN' des ZNA427. Diese Lösung stammt von FERRANTI und ist dem Datenblatt entnommen. Die Dimensionierung ist so gewählt, daß sich ein Eingangsspannungsbereich von ±1.25V ergibt. Also ziemlich genau 10mV/Digit. Ich würde aber empfehlen, die Eingangsverstärker vor dem Sample-and-Hold in der Verstärkung so einzustellen, daß sich eine Empfindlichkeit von 1mV/Digit ergibt. Abschwächen ist einfacher bewerkstelligt, wie zu verstärken. Die Berechnung für den Fußpunktwiderstand im Gegenkopplungszweig erfolgt nach der Gleichung: R1 = R2 / (Vu - 1). Für 1mV/Digit ist eine Verstärkung von 10fach notwendig. R1 also: 10K/(10-1)=1.11K. Der Eingangsspannungsbereich ist dann ±0.125V. Mehr gibt es zu diesem Schaltungsteil nicht zu sagen, außer noch folgendes: Da das benutzte Schieberegister nur 8 Bit breit ist, ich aber 9 Bit verarbeiten muß, ist das Bit 0 vom Wandler auf den 'SIN'-Eingang des 74LS165 gelegt und das BUSY-Signal des Wandlers auf den Eingang I0 des Schieberegisters ! Durch diesen Trick ist es möglich, per Programm festzustellen, wann der ZNA427 fertig ist und muß nicht blind warten, in der Hoffnung, daß er in dieser Zeit zum Ende kommt. Damit gewinnt aber die Vorgehensweise beim Heraustakten der Bits an Bedeutung. Man muß jetzt nämlich erst Takten und dann an 'SOUT' nach dem logischen Pegel sehen. Sonst wird das BUSY-Signal unversehens zum MSB. Um das BUSY-Signal des Wandlers lesen zu können, genügt es, die SHIFT/LOAD-Leitung am Schieberegister zu Takten. Der Zustand von 'SOUT' ist nach der Übernahme in das Register immer gleich 'I0'.

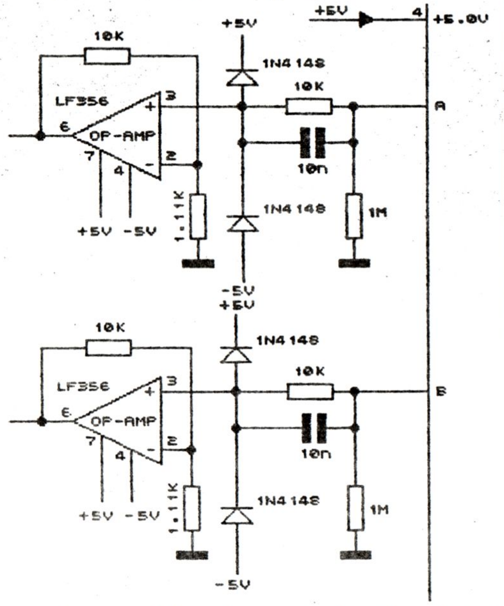
Kommen wir zum Sample-and-Hold. Die Schaltung sieht etwas verworren aus. Das liegt aber nicht etwa daran, daß ich hier den letzten Schaltungstrick ausgegraben und zu Papier gebracht habe (Stichwort 'ELEKTOR'), sondern an dem leidigen Problem, mit möglichst wenig Bauteilen ein Maximum an Funktion in die Stufe zu packen. Dazu kommt, daß die Analogschalter (HEF4066BE) mit TTL-Pegeln angesteuert werden, diese selbst aber mit Spannungspegel von maximal ±15V konfrontiert werden. Bei einer Versorgungsspannung von nur +5V würde der erlaubte Bereich von -0.3V und +5.3V mehr als nur überschritten werden. Damit dies aber nicht geschieht, habe ich die Analogschalter in den virtuellen Nullpunkt des OP's gelegt und mit jeweils zwei Dioden die Spannungshöhe am Teilerausgang auf < ±0.7V bei offenem Gegenkopplungspfad begrenzt. Damit liege ich zwar nicht ganz in der Spec, aber wer will kann die Silizium-Dioden ja



durch Germanium-Typen ersetzen. Bei mir sind diese Dinger nicht mehr allzu stark verbreitet und damit nicht zum Einsatz gekommen. Außerdem dient mir der S&H gleichzeitig als Analogmultiplexer, deshalb auch die zwei Spannungsteiler. Welcher 'Kanal' abgetastet wird, hängt davon ab, welchen Zweig ich takte. Eine Besonderheit ist der Kondensator Cx. Er verringert das Übersprechen beim Ein- und Ausschalten der Kanäle untereinander. Ohne ihn würde die Anzeige von z.B. Kanal A sich geringfügig mit dem Signal am Kanal B verändern (umgekehrt natürlich dto.). Der Wert von 350pF ist experimentell ermittelt und kann sich u. U. in

einem anderen Aufbau etwas ändern. Insbesondere, wenn IC's anderer Hersteller eingesetzt werden wie in meinem Aufbau. Die Größe des Haltekondensators (1nF) hat ebenfalls einen Einfluß auf die gegenseitige Beeinflußung. Je größer er dimensioniert wird, umso geringer ist sie. Gleichzeitig steigt aber auch die Einschwingzeit des S&H an. Also ein echtes Kompromiss-Problem. So wie die Sache gewählt ist, benötigt der S&H etwa 70usec. um auf 0.1% und 100usec., um auf 0.01% genau auf den Meßwert einzuschwingen. Bei diesem Tempo ist Sound-Sampling bestenfalls mit Telefonqualität möglich. Für Geschwindigkeitsfanatiker sicher nichts, aber für mehr Tempo heißt es auch mehr DM's spendieren. Die Pinbelegung der Analogschalter sollte unverändert bleiben, da sie so gewählt ist, daß sich ein Minimum an Störungen durch die Digitalsignale ergibt.

Kommen wir zu den Eingangsverstärkern.



Von diesen gibt es wegen der zwei Kanäle entsprechend zwei. Die Beschaltung birgt keine sonderlichen Geheimnisse. Damit nicht gleich beim ersten Meßausflug der große Manitou Rauchsignale empfängt, sind die Eingänge durch eine Schutzschaltung gegen Überspannung geschützt. Koppelkondensatoren zum Abtrennen von Gleichspannungsanteilen sind nicht eingezeichnet, da die sicherlich jeder nach eigenem Gusto einbauen will, speziell direkt am zugehörigen Umschalter.

Das Thema Netzteil ist besonders schnell abgehandelt. Es besteht nur aus zwei 5V-Reglern. Die 2.5V-Referenz ist dort nicht so gut aufgehoben und sollte sich ein lauschiges Plätzchen auf der Hauptplatine suchen. Im Übrigen ist mit Stützkondensatoren (~100nF) an den Versorgungspins der einzelnen IC's nicht zu sparen. Eine saubere Versorgung ist bei den meisten Schaltung der halbe Weg zu Sieg ! Man betrachte hierzu auch den Verdrahtungsplan der Stromversorgungsanschlüsse für die IC's. Der Analogteil ist hier über 50 Ohm Widerstände und einem 47uF Elko entkoppelt.

Die Software. Ohne diese so gefragte Substanz geht auch hier nichts. Da ich z. Zt. nur ein Programm für meinen Atari habe, gebe ich dieses zum Besten (Es spielt Digitalvoltmeter). Allerdings ist es mit dem Komfort nicht weit her. Es war und ist auch nur als Demo bzw. Testkrücke für die Inbetriebnahme meines Analoginterfaces auf die Welt gekommen. Hier also nun erst einmal das Prögrämmchen (im Anschluß gehts weiter !):


```
' Allgemeine Adress- und Datenvereinbarungen
' ------------------------------------------------
OPTION BASE 0
DIM mittel_a%(90),mittel_b%(90)
ARRAYFILL mittel_a%(),0             !Erledige einige Vorbreitungen ...
ARRAYFILL mittel_b%(),0             !Feld 0 von mittel_wert%() = Array-Pointer ...
mittel_a%(0)=1                      !ins Feld !!!
mittel_b%(0)=1
summen_wert%=2
summe_a%=0
summe_b%=0
offset_a%=0
offset_b%=0
ctrl_register|=15           !Steuer-Register fürs Programm:
'                            Bit-No.: 7 6 5 4 3 2 1 0
'                                     : : : : : : : :
'                                     : : : : : : : Kanal A (1=aktiv)
'                                     : : : : : : Kanal B (1=aktiv)
'                                     : : : : : Mittelwert-Bildung Cha. A (1=aktiv)
'                                     : : : : Mittelwert-Bildung Cha. B (1=aktiv)
'                                     : : : Frei
'                                     : : Frei
'                                     : Frei
'                                     Frei
'
port_b|=&H8F                !Port-B = Centronic-Port (Atari-Spezifisch
mfp%=&HFFFA01               !Adresse des BUSY-Signal (Atari-Spezifisch)
busy|=1                     !Bit-Maske fürs BUSY-Bit (Atari-Spezifisch)
adc%=0                      !Empfangsregister für ADC-Wert
k=0.128/128                 !128mV/128 Digit = 1mV/Digit
' ------------------------
' Die Interface-Kommandos
' ------------------------
nzd|=&H0                    !... nicht definiert
clk|=&H10                   !Schiebe Register ein Bit nach links
lds|=&H20                   !Lade Schieberegister
ads|=&H30                   !A/D-Wandlung starten
x01|=&H40                   !Verstärkung auf x1 stellen
x10|=&H50                   !Verstärkung auf x10 stellen
skb|=&H60                   !Kanal B samplen
ska|=&H70                   !Kanal A samplen
nzd|=&H80                   !... nicht definiert
dah|=&H90                   !Schreibe Kommando-Nibble ins Interface
dal|=&HA0                   !Schreibe Daten-Nibble ins Interface
nzd|=&HB0                   !... nicht definiert
nzd|=&HC0                   !... nicht definiert
nzd|=&HD0                   !... nicht definiert
sum|=&HE0                   !Sample Kanal A u. B als Summe
off|=&HF0                   !Letztes Kommando löschen und Access-LED aus
' -------------
' Hauptprogramm
' -------------
CLS
rahmen                      !Zeichne einige Kringel ...
PRINT AT(10,10);"Offset-Korrektur :    OFF";
PRINT AT(10,11);"Mittelwertbildung:    ON ";
PRINT AT(10,12);"Mittelwertfaktor :    ";summen_wert%;
DO
  taste$=INKEY$
  IF taste$<>"" THEN
    IF taste$="a" THEN
```

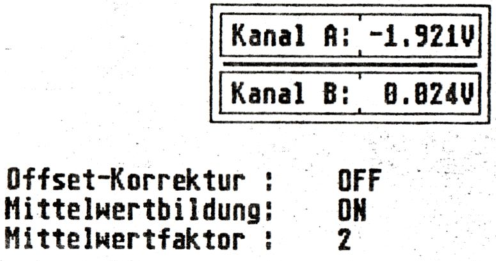
HEFT
29
Januar
1990

```
      ctrl_register|=BCLR(ctrl_register|,0)
      PRINT AT(33,5);" ##.###";
    ELSE IF taste$="A"
      ctrl_register|=BSET(ctrl_register|,0)
    ELSE IF taste$="b"
      ctrl_register|=BCLR(ctrl_register|,1)
      PRINT AT(33,7);" ##.###";
    ELSE IF taste$="B"
      ctrl_register|=BSET(ctrl_register|,1)
    ELSE IF taste$="O"
      PRINT AT(10,10);"Offset-Korrektur :    ON ";
      set_zero
    ELSE IF taste$="o"
      PRINT AT(10,10);"Offset-Korrektur :    OFF";
      offset_a%=0
      offset_b%=0
    ELSE IF taste$>="0" AND taste$<="9"
      IF taste$="0" THEN
        ADD summen_wert%,10
        IF summen_wert%>90 THEN
          summen_wert%=90
        ENDIF
      ELSE
        summen_wert%=VAL(taste$)
      ENDIF
      PRINT AT(10,12);"Mittelwertfaktor :    ";summen_wert%;"  ";
      ARRAYFILL mittel_a%(),0
      ARRAYFILL mittel_b%(),0
      mittel_a%(0)=1
      mittel_b%(0)=1
      summe_a%=0
      summe_b%=0
      IF summen_wert%=1 THEN
        PRINT AT(10,11);"Mittelwertbildung:    OFF";
      ELSE
        PRINT AT(10,11);"Mittelwertbildung:    ON ";
      ENDIF
    ENDIF
  ENDIF
  IF BTST(ctrl_register|,0)=TRUE THEN
    rd_adc(ska|,off|)        !Wecke den ADC aus seinem Schlaf
    IF BTST(ctrl_register|,2)=TRUE THEN
      mittel(adc%+offset_a%,summe_a%,mittel_a%())
    ENDIF
    PRINT AT(25,5);USING "Kanal A: ##.###V",CFLOAT(adc%)*k
  ENDIF
  IF BTST(ctrl_register|,1)=TRUE THEN
    rd_adc(skb|,off|)      !dto. für Kanal B
    IF BTST(ctrl_register|,3)=TRUE THEN
      mittel(adc%+offset_b%,summe_b%,mittel_b%())
    ENDIF
    PRINT AT(25,7);USING "Kanal B: ##.###V",CFLOAT(adc%)*k;
  ENDIF
  EXIT IF taste$="x"    !Prüfe, ob's reicht ...
LOOP
EDIT
' ----------------------------
' Die Unterprogramm-Biliothek
' ----------------------------
PROCEDURE wr_intf(cmd|)         !Schicke ein Kommando ans Interface
  ~XBIOS(28,V:cmd|,V:port_b|)
```



```
RETURN
'
PROCEDURE rd_intf                !Schieberegister auslesen
  LOCAL i%
  adc%=0
  FOR i%=1 TO 8
    ~XBIOS(28,V:clk|,V:port_b|)     !Einmal schieben wegen EOC-Bit
    ~XBIOS(28,V:off|,V:port_b|)
    adc%=SHL(adc%,1) OR (PEEK(mfp%) AND busy|)      !Bit abholen
  NEXT i%
RETURN
'
PROCEDURE rd_adc(cmd|,off_mode|)        !Kanal A/B abtasten u. auslesen
  LOCAL i%
  adc%=0
  ~XBIOS(28,V:cmd|,V:port_b|)                       !Sample Kanal A/B
  ~XBIOS(28,V:ads|,V:port_b|)                       !ADC starten
  REPEAT
    ~XBIOS(28,V:lds|,V:port_b|)                     !ADC Status holen
  UNTIL (PEEK(mfp%) AND busy|)=0                    !ADC fertig ?
  ~XBIOS(28,V:lds|,V:port_b|)                       !ADC-Wert laden
  FOR i%=1 TO 16                                    !Schieberegister auslesen
    adc%=SHL(adc%,1) OR (PEEK(mfp%) AND busy|)
    ~XBIOS(28,V:clk|,V:port_b|)
    ~XBIOS(28,V:off_mode|,V:port_b|)
  NEXT i%
  adc%=-(adc%-&H80)
RETURN
'
PROCEDURE set_zero               !Den Offset-Abgleich per Programm
  rd_adc(ska|,off|)              !Bestimme Offset von Kanal A
  offset_a%=-adc%
  rd_adc(skb|,off|)              ! dto. für Kanal B
  offset_b%=-adc%
  adc%=0
RETURN
'
PROCEDURE rahmen            !Zeichne einen Fensterrahmen
  BOX 259,60,322,83         !Box um Kanal A
  BOX 185,60,259,83         !Box um "Kanal A:"
  LINE 187,87,320,87        !Trennlinie zwischen A & B
  LINE 187,88,320,88        ! dto.
  BOX 259,92,322,115        !Box um Kanal B
  BOX 185,92,259,115        !Box um "Kanal B:"
  BOX 180,55,327,120        !Box um Alles
RETURN
'
PROCEDURE mittel(adc_wert%,VAR summe%,mittel_wert%())
  IF summen_wert%>1 THEN
    SUB summe%,mittel_wert%(mittel_wert%(0))
    ADD summe%,adc_wert%
    mittel_wert%(mittel_wert%(0))=adc_wert%
    INC mittel_wert%(0)
    IF mittel_wert%(0)>summen_wert% THEN
      mittel_wert%(0)=1
    ENDIF
    adc%=summe%/summen_wert%
  ELSE
    adc%=adc_wert%
  ENDIF
RETURN
```

Da das Programm nicht unbedingt wie 'gewöhnliches' BASIC aussieht, einige erklärende Worte dazu. Zuerst zu dem was es so macht. Nach dem Start wird per Grafik ein Rahmen gezeichnet, in den die ins dezimale umgerechneten Analogwerte geschrieben werden (siehe Bild). Soll ein Kanal nicht angezeigt werden, so ist über die Tastatur ein kleines "a" bzw. "b" einzugeben. In der entsprechende Anzeige erscheinen dann Nummernzeichen. Unterhalb des Rahmen wird ausgewiesen, ob ein Offsetabgleich durchgeführt wurde (Großes "O" für Abgleich, kleines "o" für Abgleich löschen). In den beiden Zeilen darunter ist nachzulesen, ob und wenn ja, über wie viele Messwerte gemittelt wird. Bei einer "1" wird die Mittelwertbildung abgeschaltet. Es können Werte in Einerschritte 1,2,3,...9,11,12,13.. bis 90 eingestellt werden. Für die Zehnerschritte dient die "0", deshalb sind Mittelwertfaktoren von 10,20,30... nicht möglich. Aber wen stört das ? Verantwortlich für die Mittelwertbildung ist die Procedure "mittel". Das Einlesen der Messwerte erfolgt über die Procedure "rd_adc" im Hauptprogramm. Ein Wort noch zu den Zeilen "~XBIOS(...). Hier wird eine Routine im Betriebsystem aufgerufen, mit der ich den Centronic-Port beschreibe. Hier kann/muß abhängig vom Rechner auch ein 'OUT port_b%,cmd%' stehen. Die Wirkung ist Gleich (hoffentlich !). Beim Tandy-Basic würde die Routine 'rd_adc' in etwa so aussehen:

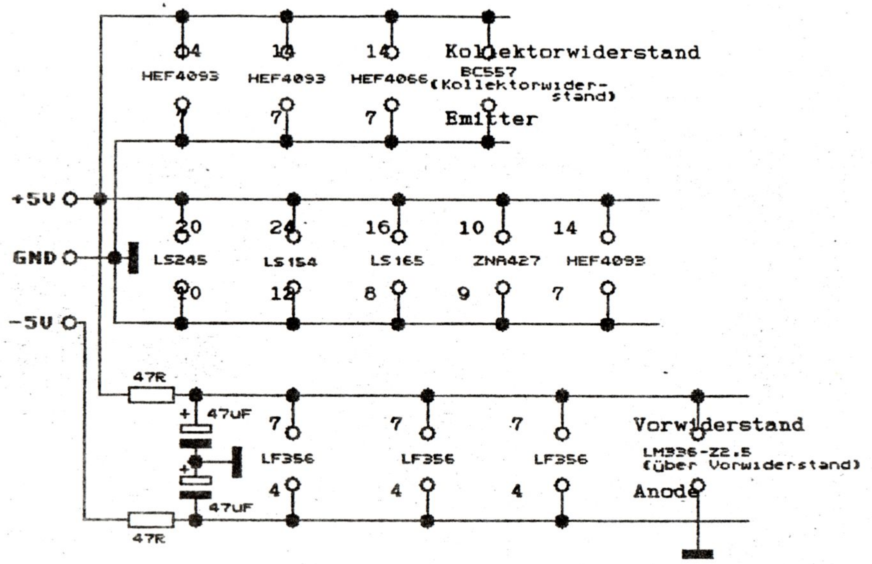



```
1'PROCEDURE rd_adc(cmd%,off%)                   !Kanal A/B abtasten
2  adc%=0
3  OUT port_b%,cmd%                             !Sample Kanal A/B
4  OUT port_b%,ads%                             !ADC starten
6  OUT port_b%,lds%                             !ADC Status holen
7  IF NOT((INP(mfp%) AND busy%)=0) GOTO 6       !ADC fertig ?
8  OUT port_b%,lds%                             !ADC-Wert laden
9  FOR i%=1 TO 8                                !Schieberegister auslesen
10   OUT port_b%,clk%                           !Erst schieben...
11   OUT port_b%,off%
12   adc%=(adc%+adc%) OR (INP(mfp%) AND busy%)  !...dann lesen
13 NEXT i%
14 adc%=-(adc%-&H80)                            !ADC-Wert korregieren
15 RETURN
```

Zum Schluß noch einen Tip zur Mittelwertbildung. Diese Routine summiert nicht erst X_n-Werte auf und gibt das Ergebnis dann aus, sondern hier wird ein gleitender Mittelwert gebildet. Ich meine damit, daß bei Anlegen einer Spannung die Anzeige mit jedem Abtastschritt eine Ausgabe liefert und sich mit fortschreitender Anzahl an Abtastungen dem Endwert annähert (Die zu messende Spannung ändert sich währenddessen nicht !!!). Schwankt die zu messende Größe, so erfolgt je nach Höhe des eingestellten Mittelwertfaktors, eine mehr oder weniger starke Integration der Schwankungen auf einen mittleren Wert (ähnlich verhält sich ein

Tiefpaß)

Als letztes noch der Anschlußplan der einzelnen IC's an die 5V bzw. gefilterten 5V:

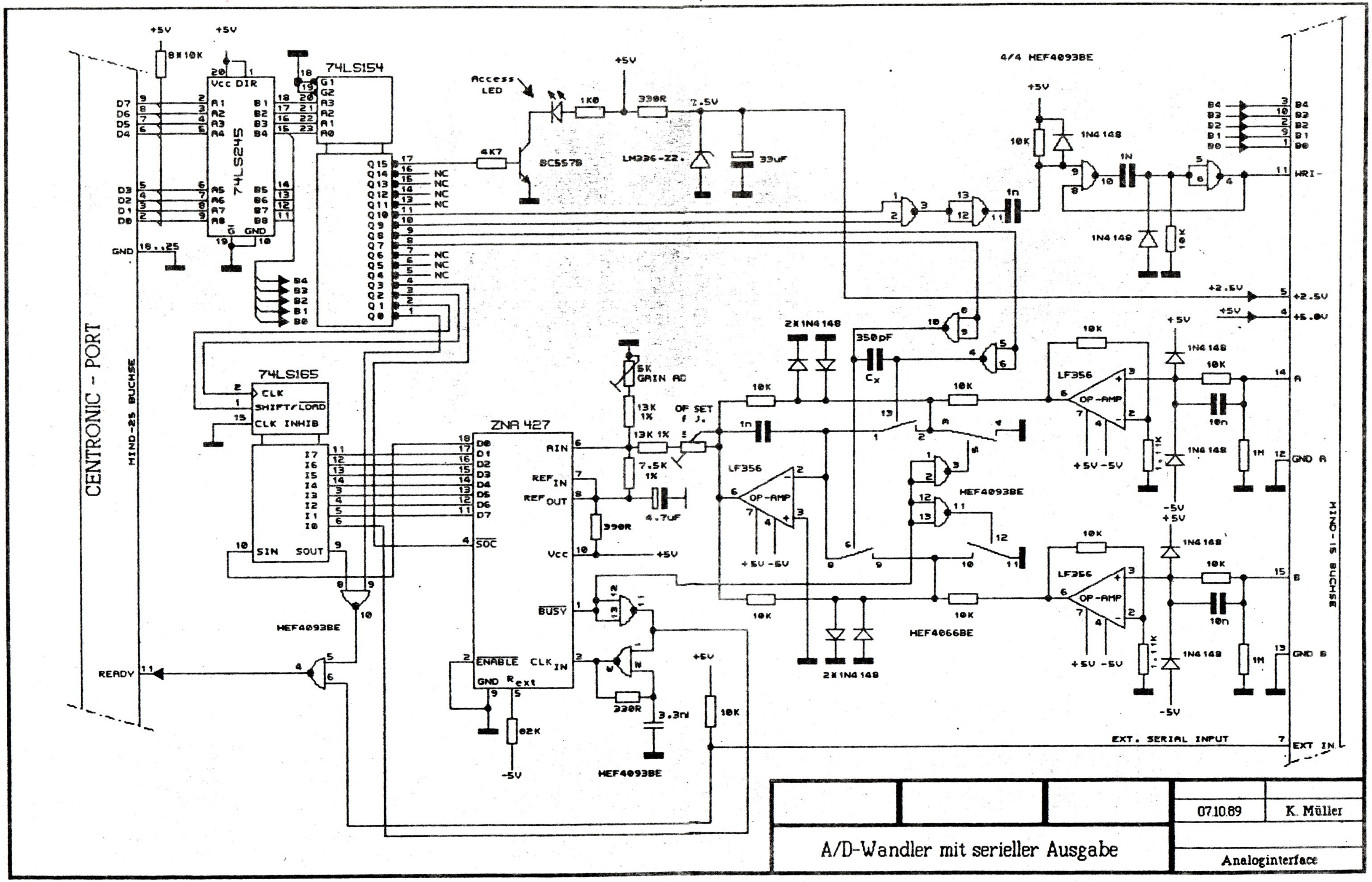


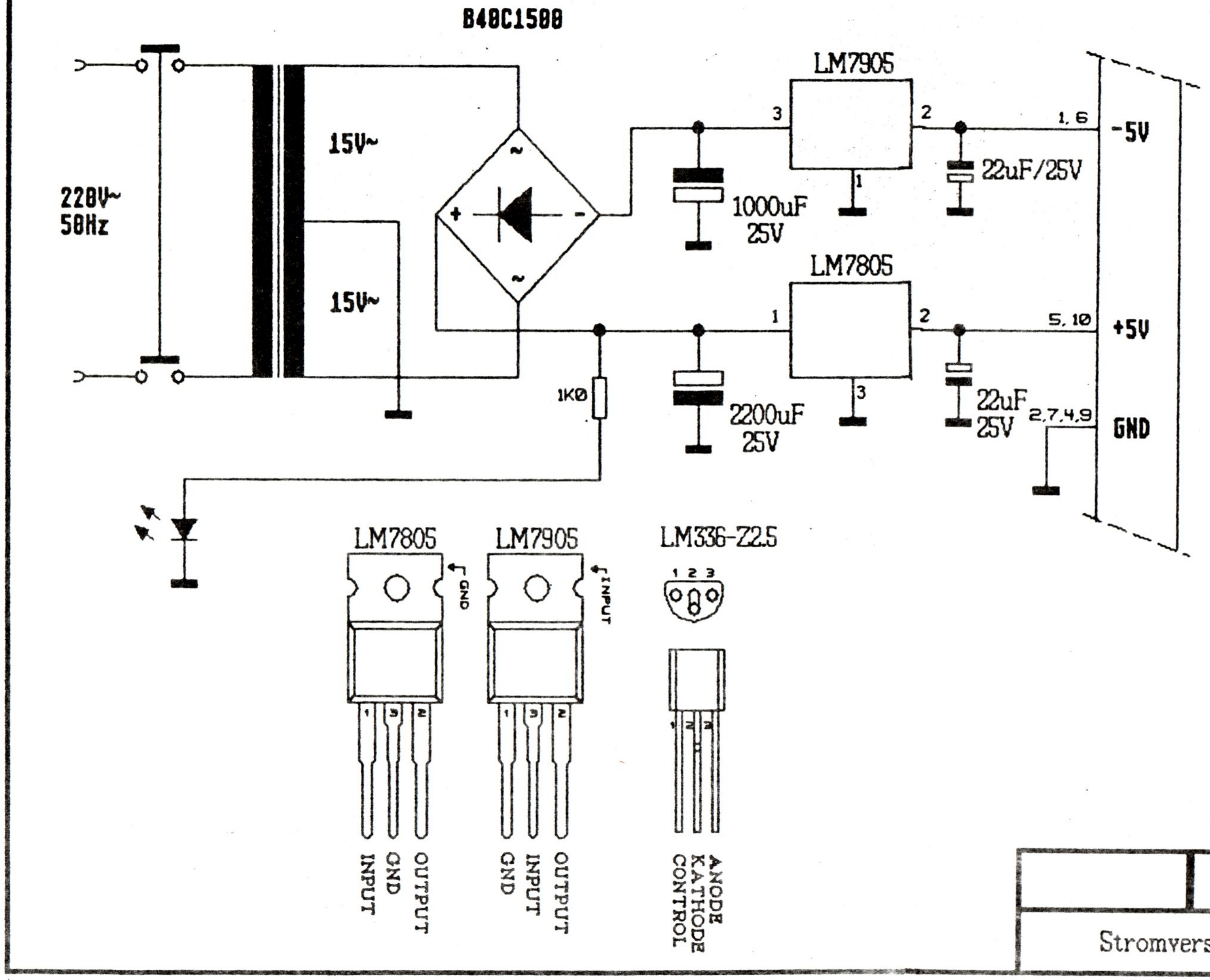
Ich glaube, nun ist es genug.

Tschüß

KuA

HEFT 29 Januar 1990


CENTRONIC - PORT
MINI-25 BUCHSE
74LS245
74LS154
74LS165
ZNA 427
Access LED
BC557B
LM336-Z2.
33uF
4/4 HEF4093BE
HEF4093BE
HEF4066BE
LF356
OP-AMP
5K GAIN AD
OF SET f J.
350pF
2x1N4148
1N4148
4.7uF
3.3n
330R
390R
82K
10K
READY
GND
EXT. SERIAL INPUT
EXT IN
MINI-15 BUCHSE
+2.5V
+5.0V
GND A
GND B
WRI-
07.10.89
K. Müller
A/D-Wandler mit serieller Ausgabe
Analoginterface



HEFT
29
Januar
1990

			19.09.89	K. Müller
Stromversorgung Analoginterface			Analoginterface	

Analog-Input:

1 - D0
2 - D2
3 - D4 (Command)
4 - +5V
5 - +2.5V
6 - NICHT BELEGT
7 - Ext. Serial Input
8 - existiert nicht
9 - D1
10 - D3
11 - WRI-
12 - GND A
13 - GND B
14 - A+
15 - B+
16 - existiert nicht

Centronic-Input:

1 - Strobe (nicht benutzt)
2 - D0
3 - D1
4 - D2
5 - D3
6 - D4
7 - D5
8 - D6
9 - D7
10 - NICHT BELEGT
11 - READY
12-17 - NICHT BELEGT
18-25 - GND

Access-LED:

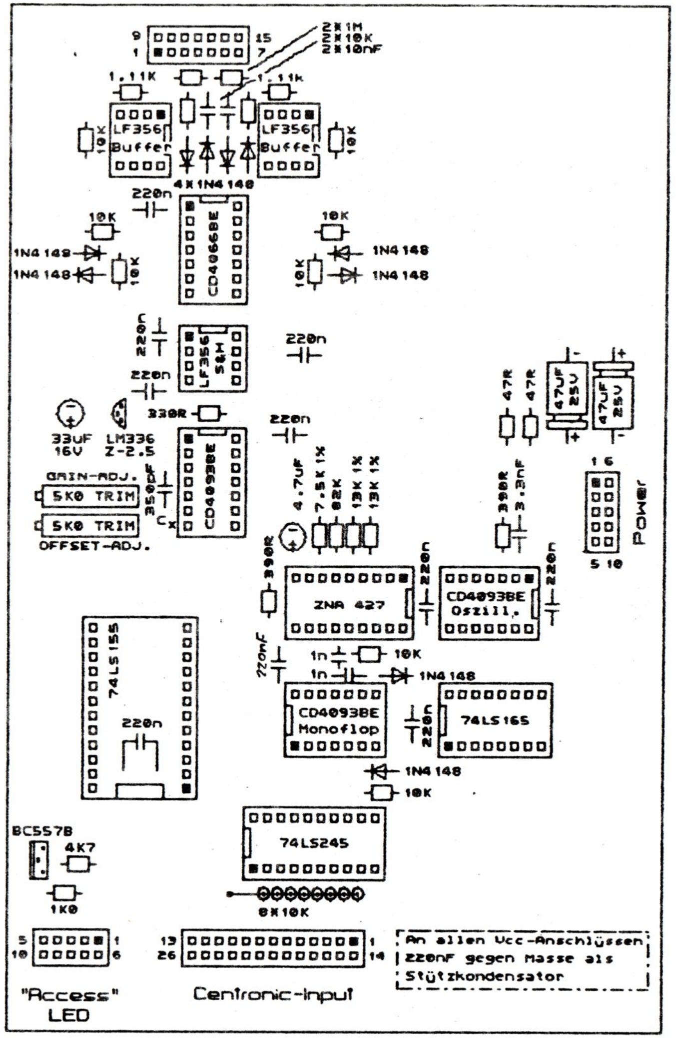
1,6 - GND
2,3,7,8 - nicht belegt
5,10 - Anode
4,9 - Kathode

Power:

1,6 - -5V
2,7 - GND
3,8 - NICHT BELEGT
4,9 - GND
5,10 - +5V

			19.09.89	K. Müller
Steckerbelegung A/D-Wandler			Analoginterface	

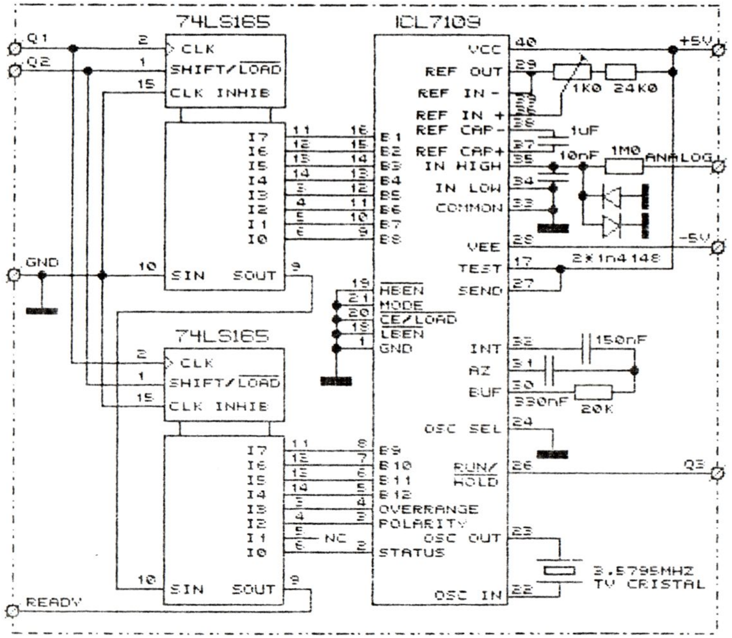
Analog-Input



			19.09.89	K. Müller
Bestückungsplan A/D-Wandler			Analoginterface	

Analog mit drei IC's

Nachdem ich bereits meine "große" Version des Analog-Interface an Jens abgeschickt hatte, erhielt ich vom Helmut einen Brief, in dem er mich fragte, wie statt des von mir eingesetzten ADC's sein bevorzugter Wandler (ICL7109) zu benutzen sei. Nach einigen Hin- und Her mit dem Datenblatt (dieser Maikäfer hatte einfach ein paar Anschlüsse zu viel) kam ich ich dann auf die vorliegende Lösung. Mit dem Messen muß man sich jedoch etwas gedulden, schafft das gute Stück doch *nur* 7.5 Wandlungen pro Sekunde. Die Ansteuerung erfolgt wie gehabt über den Centronic-Port, ausgelesen wird das Ergebnis über die BUSY-Leitung. Da alle sonstigen Dinge vom ICL7109 selbst erledigt werden - nachdem ihm über die RUN/NHOLD-Leitung der Startschuß gegeben wurde - beschränkt sich die Hauptaktivität der CPU auf trivialitäten: Starten... auslesen... vermarkten... starten...



Die Belegung der Leitungen am Centronic-Port mit den Anschlüssen der Schaltung (mit Ausnahme eben von READY oder auch BUSY) kann sich im Grunde jeder selbst zurechtlegen. Mit welchen Ausgabewerten dann welche Funktion ausgelöst wird, darf sich in dem Fall dann auch selbst ausgedacht werden. Zum Erhalt der Einheit (nicht der deutschen, soweit sind wir noch nicht !) mache ich aber einen Vorschlag:

Centronic:		**Wandler:**
D0	(2)	Q1 (CLK)
D1	(3)	Q2 (SHIFT/NLOAD)
D2	(4)	Q3 (RUN/NHOLD)
D3	(5)	n.b.
D4	(6)	n.b.
D5	(7)	n.b.
D6	(8)	n.b.
D7	(9)	n.b.
BUSY	(11)	READY
GND	(19..29)	GND

Die Stromversorgung von ±5V ist separat, das Analogsignal zwischen den Punkten "ANALOG" und "GND" bzw. Pin "COMMON" am ICL7109 anzuschließen. Die Empfindlichkeit für Vollausschlag beträgt ±0.2V. Der genaue Anzeigewert kann mittels des 1K Trimmpoti eingestellt werden. Hierzu sind natürlich 200mV zuvor anzulegen !

Zum Schluß noch ein paar Worte zur programmtechnischen Seite. Damit die Sache in Ruhestellung gehen kann, ist der Hex-Wert 03 auszugeben: CLK=1, SHIFT/NLOAD=1 u. RUN/NHOLD=0. Von diesem Wert gehen im Weiteren dann alle Aktionen aus. Die Befehlssequenz zum Auslesen des Schieberegisters hat folgendes Aussehen:

```
02H     CLK = LOW setzen
03H     CLK = HIGH und 1 Bit nach links schieben
        BUSY-Zustand abfragen
02H     CLK = LOW setzen
03H     CLK = HIGH und 1 Bit nach links schieben
        BUSY-Zustand abfragen
 :
 :
```

Und das insgesamt 16 mal oder innerhalb einer Schleife. Die letzte "Anweisung" ist immer 03H, es sei denn es folgen andere Kommandos. Wenn man's nicht macht, ist das übrigens auch nicht tragisch, aber so hat die Sache immer einen definierten Endzustand. Zum Starten der Wandlung gilt die Sequenz:

```
07H     RUN/NHOLD = HIGH, Umsetzung starten
03H     Wandler Stop, sobald Umsetzung beendet
```

Wie nun das Schieberegister geladen wird, ist klar. Der Vollständigkeit wegen, aber auch das noch:

```
01H     SHIFT/NLOAD = 0, Register laden
03H     Ruhezustand
```

oder ...

```
01H     SHIFT/NLOAD = 0, Register laden
02H     CLK = LOW setzen, SHIFT = TRUE
03H     CLK = HIGH und 1 Bit nach links schieben
        BUSY-Zustand abfragen
```

Die Bedeutung der Bit's im ausgelesenen Registerinhalt ist wie folgt:

Bit 0..11	12bit Analogwert, binär kodiert	
Bit 12	Überlauf	(1 = Überlauf)
Bit 13	Polarität	(1 = positiv, 0 = negativ)
Bit 14	nicht benutzt	
Bit 15	Status	(1 = ADC busy)

Laut Datenblatt genügt es, RUN/NHOLD für eine halbe Taktperiode auf HIGH zu halten, damit der Wandler korrekt losläuft. Die Integration des Meßsignals beginnt dann 7 Taktperioden später. Zeitlich bedeutet das 7 * (1/3.5795MHz) ~ 2usec. Erst jetzt wechselt das Status-Signal auf HIGH. Man sollte deshalb besser auf diesen Wechsel warten und dann erst RUN/NHOLD auf LOW-Pegel zurücksetzen. BASIC-Freaks werden jetzt zwar abwinken - der Interpreter schafft es wohl eher mit knapper Not mit den 7.5 Wandlungen mitzuhalten - aber in Assembler und mit entsprechender Taktfrequenz könnte man durchaus schneller sein. Die Daten sind gültig, sobald der STATUS-Ausgang auf LOW wechselt. Soll kontinuierlich gewandelt werden, genügt es, RUN/NHOLD ständig auf HIGH zu belassen. Gültige Daten können jedoch nur gelesen werden, solange STATUS auf LOW steht !

Eins sei an dieser Stelle und zum Schluß noch gesagt: Ich habe die Schaltung selbst noch nicht aufbauen können. Bis auf die Schieberegister ist alles, was um das ICL7109 herum an passiven Bauteilen zu sehen ist, dem Datenblatt entnommen. Die Bauteildimensionierung ist aber mit Sicherheit korrekt, da das IC so für die spezifizierte Funktion beschaltet werden muß. Mir erschien die Einfachheit der Lösung aber so gut, daß ich ausnahmsweise einmal ohne einen vorherigen Schaltungsaufbau die Sache kundtue. Sollte sich nichts tun, erst einmal weglegen und mindestens eine (w)einstündige Kaffepause einlegen, bis sich die Nerven beruhigt haben, dann alles nochmals in Ruhe kontrollieren. Wenn dann immer noch nichts geht, langsam anfangen zu fluchen. Ich bin mir aber sicher, daß es so arg nicht kommen wird.

Tschüß

Dierk

BÖRSE -- BÖRSE -- BÖRSE -- BÖRSE

Werner Förster
Christoph-Krebs-Str. 9
8720 Schweinfurt
☎ 09721/21841

Zu verkaufen

- TRS-80 M I
- Expansion-Board
- 80-Zeichen-Karte
- HRG II
- 3 Diskettenlaufwerke 80TR DS/DD
- reparaturbedürftiger Epson RX80
- viel Software / 14 Ordner Literatur

BÖRSE -- BÖRSE -- BÖRSE

Ein ehemaliges Mitglied des Bremerhavener Computerclubs betreibt in Oldenburg eine PC-Firma. Hartmut und ich haben den Waldemar Grundmann am 07. Oktober in Oldenburg besucht. Was dabei herauskam, könnt ihr hier lesen.

1. Eine Verkaufsanzeige der Fa. Computer-Service

2. Als kleines Dankeschön für die Veröffentlichung der Anzeige habe ich einige Bauteile kostenlos bekommen, die noch aus alten Tagen bei der Firma herumlagen. Diese Teile biete ich den Clubmitgliedern zum Abruf kostenlos an. Schon alleine der Bausatz HRG 1B hat seinerzeit mehr als 200,-DM gekostet. Die übrigen Bauteile kommen zusammen nochmals auf ca. 120,-DM.

1. IC - Sockel
(Doppel-Federkontakte)

24 x 14 pins
58 x 16 pins
23 x 24 pins
23 x 28 pins

2. IC's

2 x 74 LS 00
2 x 74 LS 04
1 x 74 LS 05
1 x 74 LS 20
1 x 74 LS 32
2 x 74 LS 74
1 x 74 LS 123
1 x 74 LS 155
2 x 74 LS 166
1 x 74 LS 174
2 x 74 LS 244
1 x 74 LS 245
1 x 74 LS 367
1 x 74 LS 368
1 x 74 LS 373

3. Pfostenstecker

9 Stück 26polig
(ohne Zugentlastung)

4. 1 Bausatz HRG 1B

(TRS80 Model 1 und VG I/II)

Platine u. fast alle Bauteile (incl. RAMs) und Bauanleitung

5. 1 Double Density Controller (Model 1/Video Genie) zum Einbau in Expansion-Interface mit Bauanleitung

6. RAM - IC's

2 x 41256
4 x 4164
1 x 4116

7. Für mich nicht definierbar:

3 IC's mit der Aufschrift:
F 21LO22PC
7735

13 x 8041016

Ab hier kostet es Geld:

Derselbe Waldemar Grundmann (Telefonnummer für Verkaufsgespräche siehe Anzeige) verkauft noch:

1 Tandy TRS 80 Model III für 100,- DM

1 Shuffle-Board, angeblich Speicheraufrüstung auf 64k, CP/M 2.2, incl. Software und Handbücher; Preis ?

Für VG IIs: 1 Speichererweiterungskarte (kenne ich nicht, es sind jedenfalls 24 Stück IC's vom 4864 drauf); Preis 50,-DM

Immer noch für VG IIs: 1 Schnittstellenkarte mit 2fach Z80-SIO/0 und 2fach Z80-PIO; Preis 50,-DM

Also auf Jungs! Ran ans Telefon! Wer zuerst kommt, malt zuerst! Da die Telefonnummer jetzt richtig im Info steht, verweise ich auf die Mitgliederliste. Wer suchet, der findet.

Hans-Günther, Dezember 1989
2876 Berne 2

Computer Service

Moslestr. 74 (Victoria-Haus) • 2900 Oldenburg • Tel.: 0441 / 25072
Im TZ • Am Wollelager 8 • 2870 Delmenhorst • Tel.: 04221 / 16303
Maiglöckchenweg 5 • 2910 Westerstede • Tel.: 04488 / 1629

Jetzt sind sie da: CS Computer

CS AT 286
- 12 MHz
- 640 KB Ram
- Festplatte 20 MB
- 14" Monitor
- 102 Tasten Tastatur

2425.-

- mit 16 MHz NEAT Board

2755.-

- 12 MHz mit 40 MB Festplatte

2775.-

VGA Karte 800 x 600 **375.-**

CS AT 386
- 20 MHz Takt
- 2 MB Ram
- Festplatte 40 MB
- 14" Monitor
- 102 Tasten Tastatur

4685.-

CS AT 386 SX
- 20 MB Platte
- sonst wie AT 386

3525.-

KHK Private Line 79,50
PD - Software je 7,50

Selbstverständlich sind: 12 Monate Garantie, FTZ-Nr. und Deutsches Handbuch

Einführung in Mikrocomputer-Betriebssysteme
2. Teil: Das Klassen-Problem
von Gerald Schröder

Ich habe Euch im ersten Teil zwei sehr unterschiedliche Computersysteme vorgestellt, auf denen aber (hoffentlich) dasselbe Programm laufen kann - nur einmal druckt es eben auf einem Laserdrucker und einmal auf einem Fernschreiber usw. Aber das ist doch nicht so selbstverständlich, oder?! Beim Super-Turbo-System sagt das Programm zum Betriebssystem: "Druck' mal ein 'a'!" und das BS druckt es auch wirklich auf dem Laserdrucker - aber was ist bei dem Fernschreiber-System?

Würde die Aufforderung lauten: "Drucke das Zeichen auf dem Laserdrucker!", hätte das Fernschreiber-System große Schwierigkeiten. Aber dem ist ja nicht so! Das Programm sagt eben nur: "*Drucke* das Zeichen!", ohne dabei den Drukker näher zu bezeichnen. Und somit ist die Art des Druckers egal. Aber woher weiß das BS, daß ein Fernschreiber "fast" das gleiche wie ein Laserdrucker ist - also daß beide *Drucker* sind?! Warum "druckt" das BS hier das Zeichen nicht auf eine Lochkarte? Das wäre doch peinlich, gelt?!

Da kommt eine Eigenschaft des Betriebssystems ins Spiel, die wir näher beleuchten müssen: die Aufteilung aller Geräte in Geräte-"Klassen". Alle Geräte einer Klasse leisten prinzipiell das gleiche, allerdings sind sie meistens unterschiedlich ausgestattet. Beispiel: Die Klasse der Drucker besteht aus Fernschreibern, Typenrad-, Matrix-, Tintenstrahl-, Laserdrukkern usw. Alle drucken (in erster Linie) Zeichen auf Papier. Manchmal können Klassen auch noch weiter aufgeteilt werden, wie Ihr gleich sehen werdet. Ich zähle mal auf, was mir an Klassen so einfällt - ohne Anspruch auf Vollständigkeit.

Einfach ist noch die grobe Aufteilung in Massenspeicher und Ein-/Ausgabe-Geräte. Zwar ist diese Aufteilung nicht ganz eindeutig, denn schließlich geht's beim Speichern auch um das Ein- und Ausgeben, aber da gibt es einige Unterschiede: a) Auf Massenspeicher wird etwas geschrieben, das wir auch wieder vom gleichen Gerät lesen wollen. Das trifft auf andere Ein-/Ausgabe-Geräte normalerweise nicht zu. b) Bei Massenspeichern geben wir uns nicht mit einzelnen Zeichen ab, sondern erledigen die Ein-/Ausgabe blockweise (im CP/M "Record" genannt und 128 Bytes lang, im Newdos dagegen ein "Sektor" mit 256 Bytes).

Bei den Massenspeichern wird selten weiter unterschieden, jedenfalls von Disketten aufwärts (Kassetten sind "out", auch wenn IBM noch gerne eine Kassettenschnittstelle einbaut, was die Fortschrittlichkeit dieser Geräte unterstreicht). Einziges Unterscheidungsmerkmal bei Massenspeichern: wechselbare gegenüber festen Medien; typische Vertreter der beiden Gattungen: wechselbare Disketten und feste Festplatten. Daß wiederum Disketten-/Festplatten- und sonstige Laufwerke äußerst unterschiedlich groß sein können (nicht nur äußerlich, sondern auch von der Kapazität her), dürfte klar sein. Da helfen uns die Klassen nicht weiter.

Aber die Ein-/Ausgabe-Geräte lassen sich sehr schön weiter unterscheiden. Eine klassische Klasse ist die Konsole (engl. "console"), die in Urzeiten aus einer Schreibmaschine bestand, auf der ein Computer auch was drucken durfte. Getipptes ging außer auf's Papier auch an den Rechner. Heutzutage nennt sich sowas eher "Terminal" und besteht aus einer umfangreichen (gegenüber der Schreibmaschine) Tastatur und einem Bildschirm, auf dem sich manchmal Zeichen blicken lassen. Dies ist die Geräte-Klasse, über die jedes Programm normalerweise Kontakt zum User aufnimmt, von der es Eingaben erwartet und auf das es Ausgaben macht. Aber es ist Sache des Betriebssystems, was wirklich hinter der Konsolen-Ein-/Ausgabe steckt (Schreibmaschine, Terminal, Spracherkennung und -ausgabe ...).

Auch sehr klassisch ist die Klasse der Drucker. Drucker sind Teile, die Papier zeichen-, zeilen- und seitenweise verschmutzen, wobei das Papier wie eine Klorolle gegen endlos strebt, aber im falschen Moment zuende geht. Probleme bekommen wir aber bei grafikfähigen Druckern - dazu unten mehr bei den "Grafischen Ausgabe-Geräten".

Etabliert hat sich noch die Klasse der Kommunikations-Geräte, wie Modem und Akustikkoppler, über die Rechner Verbindung zur Außenwelt aufnehmen, wobei die Außenwelt oft auch nur aus anderen Rechnern besteht. Interessant wäre hier auch das Stichwort "Netzwerke", wo mehrere Rechner sich gegenseitig Daten zuschaufeln. Leider muß ich mangels Erfahrung da passen. Aber eins ist sicher: An sowas haben weder die Newdos- noch die CP/M-Entwickler gedacht.

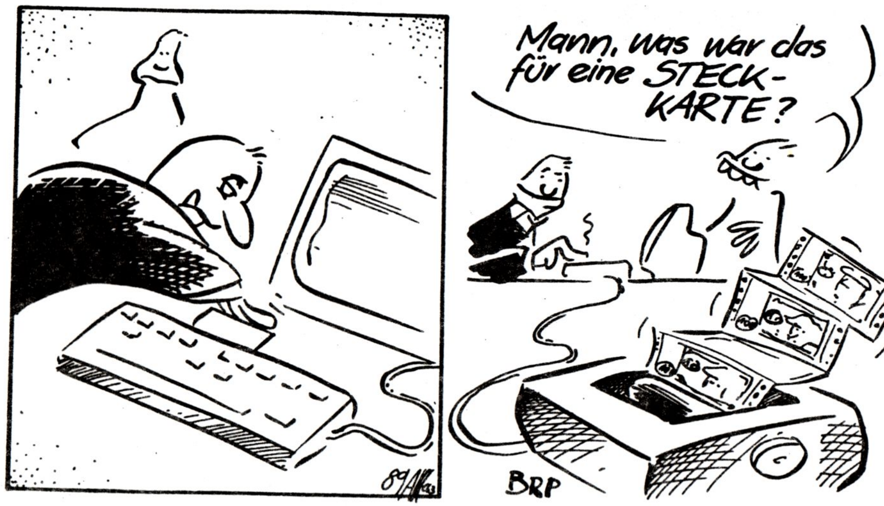
Neuerdings kommen noch zwei Geräte-Klassen hinzu: Die grafischen Ein- und Ausgabe-Geräte, als da auf der einen Seite wären: Grafik-Karten nebst Plottern und grafikfähigen Druckern, während die andere Seite von Mäusen, Trackballs, Grafik-Tabletts, "Didjieteisern" und Lightpens belebt wird. Bei den Ausgabe-Geräten muß natürlich unterschieden werden zwischen "flüchtigen" (Bildschirm-) und "nicht-flüchtigen" (Drucker-) Ausgaben. Bei der Eingabe wiederum gibt es "Lokalisierer", mit denen sehr genau eine Koordinate angewählt werden kann (z.B. Digitalisier-Tisch), und "Auswähler", mit denen nur grob auf etwas (auf dem Bildschirm) gezeigt werden kann (z.B. Maus). Hier hat selbst MS-DOS noch Nachholbedarf, denn außer der (scheinbar nicht sehr erfolgreichen) Unterstützung einiger Grafikkarten wird der Rest hoheitsvoll ignoriert.

Natürlich kann ich hier etwas vergessen haben, aber die hauptsächlichen Geräte-Klassen für Mikrocomputer sollten dabei sein. Bei größeren Rechnern kommen vielleicht noch welche dazu, bei kleineren ganz bestimmt. Warum gerade bei kleineren? Nun, da geht es oft auch um Steuerungsaufgaben, für Waschmaschinen, Walzstraßen oder Ampeln (diese Assoziationskette ist hoffentlich einleuchtend). Diese Steuerungsrechner haben also andere Probleme als unsere Mikrocomputer; dafür fehlt ihnen aber auch oft das Betriebssystem.

Erwähnen möchte ich aber noch die Schnittstellen zu den Geräten. Schnittstellen sind einerseits diese niedlichen kleinen Buchsen und Stecker an der Rückseite des Rechner-Gehäuses, andererseits aber auch die Belegung der Kabel und die Art der elektrischen Signale. (Daneben werden auch die Verbindung vom Anwenderprogramm zum BS, die Verbindung von Programmen untereinander und viele andere Verbindungen "Schnittstelle" oder Neu-Deutsch "Interface" genannt, was zur Verwirrung beiträgt. Demnächst werden auch die schlagenden Studenten-Verbindungen in "Schnittstellen" umbenannt.)

Es gehört sich für ein Anwenderprogramm nicht, dem BS zu sagen, mit welcher Schnittstelle es arbeiten soll. Warum? Weil es schließlich egal ist, an welcher Schnittstelle welches Gerät hängt. Zwar ist die Schnittstelle zum Drucker meist parallel nach der Centronics-Norm ausgeführt und die zum Kommunikations-Gerät meist seriell nach der Norm RS-232 bzw. V.24. Aber auch andersrum oder nur mit einer Sorte geht's. Es ist Aufgabe des Betriebssystems, sich zu merken, an welcher Schnittstelle welches Gerät hängt. Das Anwenderprogramm sagt nur, welches Gerät es benutzen möchte.

Was haben wir nun von den Geräte-Klassen? Für jede Klasse von Geräten sieht das Betriebssystem besondere Aufrufe/Anfragen vor (wie schon gesehen: "Programm an Betriebssystem: Gib mal auf der Konsole (bzw. einem Gerät der Klasse "Konsole") ein 'a' aus!"). Das Programm sieht dann nur noch diese Klassen (bzw. die entsprechenden Aufrufe des Betriebssystems) und das Betriebssystem kümmert sich darum, das richtige Gerät für diese Klasse zu finden. Und das auf jedem Rechner!



Hier nochmal eine Zusammenfassung der Klassen mit typischen Vertretern (ohne Anspruch auf Vollständigkeit):

Massenspeicher	
wechselbare	(Diskette, CD-ROM)
feste	(Festplatte, RAM-Disk)
Ein-Ausgabe-Geräte	
Konsolen-Eingabe	(Tastatur)
Konsolen-Ausgabe	(Bildschirm, Drucker)
Drucker	(Laserdrucker, Fernschreiber)
Kommunikation	(Modem, Akustikkoppler)
Graphische Eingabe	
Lokalisieren	(Grafik-Tablett, Digitalisier-Tisch)
Identifizieren	(Maus, Trackball)
Graphische Ausgabe	
Flüchtige	(Grafikkarte)
Nicht-flüchtige	(Plotter, grafikfähiger Drucker)

Einführung in Mikrocomputer-Betriebssysteme
3. Teil: Was treibt ein Treiber?
4. Teil: Was treibt das "Basic" im Betriebssystem?
von Gerald Schröder

Zur Erinnerung: Im letzten Teil haben wir gesehen, was das Betriebssystem für das Anwenderprogramm an Geräte-Klassen zur Verfügung stellt. Allerdings leistet ein Betriebssystem mehr, andere wieder weniger - also versteift Euch nicht darauf. Zum Beispiel werdet Ihr weder im Newdos noch im CP/M eine Routine zum Zeichnen einer Linie oder zur Abfrage einer Maus finden. Das war eben damals (bei der Entwicklung dieser Betriebssysteme) noch nicht aktuell.

Es gibt dauernd neue Geräte, die beachtet sein wollen. Gestern war noch das 8-Zoll-Diskettenlaufwerk "in" (Größe Aktenkoffer bis Klein-Kühlschrank), heute ist es ein Laufwerk in der Größe einer Zigarettenschachtel. Früher waren 500 Kilobyte auf einer Diskette nicht schlecht, heute stößt eine löschbare Opto-Platte in den Gigabyte-Bereich vor (1000x1000 Kilobyte). Also muß ein Betriebssystem flexibel sein.

Aber "flexibel sein" ist (wie so vieles) relativ: TRS-80/Newdos war mit der Unterstützung von 4 (in Worten: vier) Laufwerken ziemlich flexibel, denn: wer konnte sich damals (1870-1979) schon 4 Laufwerke leisten! CP/M war mit 16 Drives etwas flexibler (fragt mal Helmut, ob das reicht). MS-DOS konnte (bis vor kurzem) nur Teile von Festplatten mit einer Größe von 32 Megabyte verwalten; bei den neuen Festplatten von 500 MB haben wir dann locker 13 Partitionen (=Teile, entspricht logisch einem Diskettenlaufwerk). Und bei allen Betriebssystemen gilt: Wehe, ein File wird mal zu lang (wie hoch die Grenze auch sein mag).

Machen wir uns noch einmal klar, was das BS leisten soll und wo das Problem sitzt. Es soll leisten: Für jedes Programm die Ressourcen des Rechners verwalten und verfügbar machen. Ressourcen sind alle "Möglichkeiten" des Rechners, also die angeschlossenen Peripheriegeräte, aber oft auch Co-Prozessoren (für Arithmetik o.ä.) und der verfügbare Speicher.

Das Problem: Bei der Entwicklung des Betriebssystems müssen alle denkbaren Ressourcen beachtet werden; dies ist besonders bei Peripheriegeräten schwer, denn oft gibt es diese gerade erst auf dem Papier. Aber trotzdem möchte jedermann/-frau diese neuen Geräte nutzen können, wenn sie auf den Markt kommen, und das möglichst mit den alten Programmen!

Aber wie geht das? Wie kann ein MS-DOS von anno dazumal mit optischen Platten umgehen, die erst viel später die geheimen Entwicklungslabors verließen? Und warum nagt heute trotz aller Abwehrversuche eine Maus an den meisten IBM-Rechnern, obwohl MS-DOS doch (streng konservativ erzogen) eine Maus gar nicht kennt?

Einen Teil der Lösung kennt Ihr bereits: Die Geräte-Klassen. Wenn ein neues Gerät auf den Markt kommt, ordnen wir es in eine der Klassen ein. Da unser Anwender-Programm nur diese Klassen sieht (und nicht die "echten" Geräte), bemerkt es den Unterschied nicht und kann sofort mit dem neuen Gerät arbeiten. Das neue Gerät ist einfach ein weiterer Vertreter aus einer der altbekannten Klassen. Ärgerlich wird es dann, wenn für das neue Gerät keine Klasse paßt. Z.B. gibt es weder im Newdos noch im CP/M Geräte-Klassen für Grafik; keine Grafikkarte, kein Plotter, keine Maus läßt sich dort in eine Klasse einordnen.

Also ist die Einführung eines neuen Geräts für das Anwenderprogramm kein Problem, sehr wohl aber für das Betriebssystem! Denn dieses hat die Aufgabe, mit dem Gerät umzugehen, es zu steuern. Wie bringen wir dem BS bei, wie es das neue Gerät zu bedienen hat? Schließlich ist ein Unterschied, ob wir als Vertreter der Klasse "Drucker" einen Fernschreiber, einen Matrix-Drucker oder einen Laserprinter anschließen.

Das Zauberwort heißt: Treiber. Das sind nicht diese bärtigen Jungs mit Panflöte, Hirtenstock und Schäferhund, sondern niedliche, kleine Programmteile, die sich nachträglich irgendwo im oder am BS einnisten und die Arbeit übernehmen, die das BS nicht alleine machen kann. Für jedes Gerät gibt es einen Treiber. Ein Treiber kann aus mehreren Unterprogrammen bestehen, z.B. bei Treibern für Geräte der Klasse "Massenspeicher": ein Unterprogramm zum Auswählen eines Sektors, eines zum Lesen eines Sektors, eines zum Schreiben usw.

Wir können jetzt festlegen:

1. Für jedes Gerät muß es einen Treiber geben.
2. Die Treiber für alle Geräte einer Klasse müssen Teile (Unterprogramme) enthalten, die das Gleiche leisten. Z.B. müssen sowohl die Treiber für Diskettenlaufwerke als auch die Treiber für Festplattenlaufwerke ein Unterprogramm zum Lesen eines Sektors beinhalten, da beide zur Klasse "Massenspeicher" gehören.
3. Das BS muß es ermöglichen, daß neue Geräte eingeführt werden, indem neue Treiber in das BS "eingebunden" werden. Nach dem "Einbinden" kennt das BS diese neuen Treiber und benutzt sie, um die neuen Geräte anzusteuern.

Gerade die letzte Forderung ist nicht zu verachten. Was nützt uns die Klasse "Massenspeicher" im CP/M, wenn wir es nicht schaffen, den Treiber für unsere neue 20-MB-Festplatte in das CP/M einzubinden? Dann kann das CP/M nie was mit der Festplatte anfangen, weil der Treiber ihm eben nicht bekannt ist. Und: Wat de Bur nich kinnt, dat frett he nich. (Für unsere Südländer: Woas i ned woaß, macht mi ned hoaß.)

Nun könnten wir ja denken, daß ein BS nur aus Treibern besteht. Dem ist aber nicht so! Treiber dienen nur der grundlegenden Anpassung an die speziellen Geräte, mit denen sich unser Betriebssystem rumschlagen muß. In einem Treiber wird es kein Unterprogramm geben, das ein Inhaltsverzeichnis (einer Diskette, Festplatte o.ä.) verwaltet. Das erledigen andere Teile des Betriebssystems, und zwar für alle Geräte einer Klasse gleich. Außerdem ist dieser Teil des Betriebssystems auf allen Rechnern gleich, also können sich alle Anwenderprogramme auf ihn verlassen.

Damit haben wir zwei Teile eines Betriebssystems: Einen festen Teil, der sich nie ändert, egal auf was für einem System er läuft und welche Geräte daranhängen, und einen variablen Teil, der sich je nach Rechner und Umgebung ändert. Der variable Teil sieht für jedes Gerät einen eigenen Treiber vor, während der feste Teil alle Geräte einer Klasse gleich behandelt und dabei auf die Treiber aus dem variablen Teil zurückgreift. Also ist es für den festen Teil egal, ob er es mit Disketten und/oder Festplatten zu tun hat, weil die Treiber für ihn absolut identisch wirken.

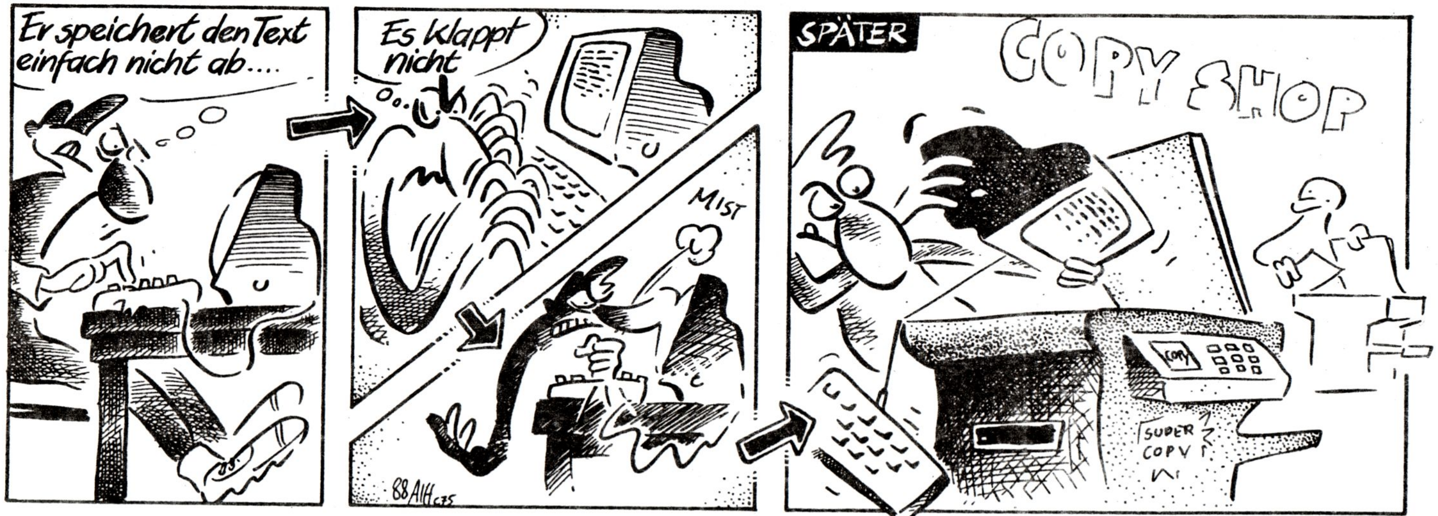
Die CP/M- und MS-DOS-Freaks haben es schon gemerkt: Der variable Teil ist das BIOS, das "Basic Input/Output System" (die Grundlage (engl.: basic) aller Ein-/Ausgaben in einem Rechner), der feste Teil dagegen wird BDOS genannt, "Basic Disk Operating System", wobei dieser Name etwas irreführend ist, denn schließlich geht es dem Betriebssystem nicht nur um Disketten. Das "Basic" hat übrigens nichts mit dem Löffel-Zwo- oder sonstwie-BASIC zu tun, mit dem immer noch die meisten Freaks ihre Gehirngänge verstopfen (sorry); da ist BASIC übrigens eine Abkürzung für "Beginners All Purpose Instruction Code" bzw. "Mehrzweck-Programmiersprache für Anfänger".

Wenn wir uns jetzt von dem Weihnachtsgeld ein neues Gerät (Grafikkarte, Laserdrucker, Festplatte, ...) kaufen und es an unseren Computer anschließen (bzw. einbauen), muß das BIOS geändert werden, aber nicht das BDOS. CP/Mler haben es hier schwer: Normalerweise muß man sich den Treiber selber schreiben und irgendwie in das BIOS bringen, wobei das "irgendwie" darauf hinausläuft, daß nur "echte" Programmierer (die Assembler mit der Muttermilch aufgesaugt haben) so ein neues Device genießen können. Sowieso kann CP/M mit vielen der neuen Geräte nicht ordentlich umgehen, wenn es auch gegenüber Newdos einige Klassen besser* ist (aber das Alter!).

Bei MS-DOS sieht die Sache anders aus: Hier gibt es nicht ein BIOS, sondern die meisten Geräte bringen ihr eigenes auf einer Steckkarte mit (was auch nicht problemlos ist); oder es müssen nur irgendwo einige Tabellen geändert werden; oder bei jedem Booten des Rechners zieht der sich von Festplatte/Diskette die neuen Treiber rein. Das Betriebssystem wächst also dynamisch (oft im Wildwuchs). Das BDOS ändert sich trotzdem nicht.

Soweit, so schlecht. Beim nächsten Mal soll es dann noch um virtuelle Geräte gehen; das sind Geräte, die es nicht gibt (genial, gelt?!).

* bzw. flexibler



Tips für den Computer-Kauf

Beste Gelegenheit, sich über das aktuelle Angebot an Personal-Computern zu informieren, bieten spezielle Messen. Möchte man sich lieber daheim mit dem Thema Computer beschäftigen, sind Computer-Magazine das geeignete Mittel, um zu einer Kaufentscheidung zu kommen. Testberichte vermitteln nicht nur einen Eindruck von Preis und Leistung des einzelnen Systems, sondern setzen sie auch ins Verhältnis zu Konkurrenzprodukten.

Anzeigen in Zeitschriften, Zeitungen und auch örtliche Händler bieten die komplette Hardware-Palette an. Die wichtigste Regel lautet, daß das billigste Angebot nicht auch das preiswerteste ist. Zählt man nicht zu den Profis, erhält die Beratung durch den Händler besonderes Gewicht. Grundsätzlich sollte das Angebot mehrerer Händler geprüft werden. Von Interesse ist auch die Frage, welches Gerät der Händler benutzt, er hat sich bestimmt nicht das schlechteste genommen. Gute und preiswerte PC schickt auch der Versandhandel ins Haus. Computer-Versender bieten inzwischen einiges an Leistung und Service.

Manche Versender bieten einen qualifizierten Beratungs-Service per Telefon an. Damit nicht schon die Telefonkosten den Geldbeutel plündern, können Interessenten dort zum Ortstarif anrufen.

Besonders sicher kann sich der Käufer fühlen, wenn er für seinen PC 30 Tage Rückgaberecht hat. Wenn der Rechner nicht den Vorstellungen entspricht, wird er einfach zurückgeschickt. Eine Garantiezeit von zwölf Monaten und ein kostenloser Reparatur-Service vor Ort wirken sehr beruhigend, vor allem wenn der PC beruflich genutzt wird. Außerdem kann eine Firma sich einen solchen Service wohl nur leisten, wenn die Produkte von hoher Qualität sind.

Und wenn dann noch einige Standardprogramme wie Textverarbeitung, Datenbank und anderes im Lieferumfang enthalten sind, kann der PC-Spaß so richtig losgehen.

Der Computer

Folgender Artikel stammt aus einer Zeitung der "Marxistischen Gruppe" der Uni Hamburg. Halt, laßt nicht gleich die Scheuklappen runter! Bis auf einige klassenkämpferische Anmerkungen ist der Artikel meiner Meinung nach faktisch einwandfrei und bringt das Thema "Technischer Aufbau und soziale Wirkungen von Computern" (relativ) kurz und präzise rüber. Zwar sind die Formulierungen teilweise etwas hochgestochen (ein Hoch auf die Fremdworte!), aber es sind eine Menge Anregungen zum Nachdenken und Diskutieren enthalten. Die Hervorhebungen stammen übrigens vom Original-Autor (wer immer das auch gewesen sein mag - wahrscheinlich Marx persönlich).

Gerald Schröder

I.

Wozu ein Computer auch verwendet werden mag: Er ist eine *Rechenmaschine*. Im Unterschied zum volkstümlichen Taschenrechner und dessen mechanischen Vorgängern, wie sie noch vor wenigen Jahren die Büros bevölkerten, ist ein Computer ein *Automat*. Er wurde erfunden, um komplizierte und langwierige Berechnungen ohne menschliche Intervention zwischen den einzelnen Schritten durchzuführen. Das heißt, er kann einen vorweg festgelegten Plan der Rechnung auf die jeweils gegebenen Werte anwenden und dabei sein tatsächliches Verhalten nach den anfallenden Zwischenergebnissen richten, also etwa Fälle unterscheiden oder in Fehlersituationen anhalten. Diesem Zweck des Computers entspricht seine funktionelle Gliederung, die, wie auch immer variiert und verfeinert, bis heute grundlegend ist. An die Stelle des menschlichen Rechners, der sich mit einer herkömmlichen Tischrechenmaschine und einem Notizzettel an die Arbeit macht, tritt ein "Leitwerk", das ein "Rechenwerk" sukzessive mit der Ausführung elementarer Operationen betraut und Zahlenwerte von einem "Speicherwerk" festhalten bzw. liefern läßt. Hinzu kommen noch "Ein-/Ausgabewerke".

Dabei ist ein Computer kein Spezialist. Während ein Zigarettenautomat kein Bier verkaufen kann und ein Maschinensystem, das beispielsweise Autos fabriziert, bei jeder Variation des Produkts größere Umrüstungsarbeiten erfordert, ist die technische Gestalt des Computers, seine "Hardware", unabhängig von der besonderen Aufgabe fixiert, aber für schlechterdings jede tauglich. Entscheidend ist dafür die Konstruktion des Leitwerks. Es muß den jeweils bezweckten Gang der Dinge steuern und dabei selber nach einem invarianten Schema verfahren. Dies tut es, indem es die einzelnen auszuführenden Operationen erst aus Daten ableitet, die ihm genauso wie die Zahlen, mit denen gerechnet werden soll, im Speicher übergeben werden. Die Spezifik der Aktivität fällt somit ganz auf die Seite dieser "Software"; insofern ein Computer nicht ein besonderes Rechenverfahren verkörpert, sondern so eingerichtet ist, daß er geeignete Darstellungen solcher Verfahren - "Programme" - umsetzt und damit ausführt, ist er eine *universelle* Rechenmaschine.

Die primitiven Fähigkeiten des Computers, wie komplex auch immer ihr letztendlicher Gebrauch, sind sehr bescheiden. Wenn ein Mensch schriftlich addiert oder multipliziert, benutzt er Zahlenzeichen, Ziffern in Stellenschreibweise, und das Rechnen selbst ist ein Manipulieren des äußerlichen Materials, Ersetzen zweier Ziffern durch eine dritte, Übertragen in eine anderen Position, wobei Art und Abfolge dieser Operationen durch Regeln - einen Algorithmus - festgelegt sind. Genauso verfährt ein Computer. Sein Geschäft ist mechanisch und läßt sich deshalb z.B. mit Hilfe von Zahnrädern, deren Stellung und Ineinandergreifen, realisieren. Tatsächlich werden elektronische Bauteile verwendet, insbesondere wegen der damit erreichbaren Geschwindigkeit, und wenn sich dann aus technischen Rücksichten die binäre Form der Darstellung (Schalter auf, Schalter zu) aufdrängt, so belegt das augenfällig die abstrakte, ganz inhaltslose Bestimmheit als Prinzip der Zahl und des Rechnens. Diese Tätigkeit ist nun allerdings zu unterscheiden von dem Zweck, ein numerisches Resultat hervorzubringen, und in diesem Sinne wird der Computer oft als *Zeichen* verarbeitende Maschine definiert. Man kann mit Hilfe seiner Bits (binary digits, Binärzeichen) auch ganz andere Dinge codieren, z.B. Buchstaben oder Merkmale vorgestellter Dinge, und dann durch geeignete Manipulationen an dieser Darstellung z.B. Schriftstücke aufbereiten oder Inventarlisten führen; die Geschichte der Verbreitung des Computers ist auch eine der Entdeckungen, was sich alles auf diese Weise bewerkstelligen läßt. Dabei ist der Umgang des Computers mit Zeichen, laut DIN 44 300 "Elemente einer vereinbarten endlichen Menge" und sonst nichts, nicht zu verwechseln mit dem Gebrauch, den die Intelligenz von Zeichen macht. Für ihn sind sie nicht die Manifestationen von Gedanken und Vorstellungen, sondern in ihrem physikalischen Dasein Material einer absolut gedankenlosen Aktivität. Ein Computer *transformiert* Bitketten in andere Bitketten, wie etwa ein Setzer ein Manuskript ohne Rücksicht auf den Inhalt typographisch umsetzt. Und er tut dies, indem er auf andere Bitketten, die seinem festen Satz von Befehlen angehören, *reagiert*, so wie eine Telefonvermittlung auf eine gewählte Nummer mit der Herstellung des zugehörigen Anschlusses antwortet.

Dem Prinzip, daß ein Computer ein universeller Rechenautomat ist, gab es in seiner nunmehr 45jährigen Entwicklungsgeschichte natürlich nichts hinzuzufügen; Leistungsunterschiede zwischen modernen und historischen, aber auch großen und kleinen Exemplaren sind quantitativer Natur, betreffen also Merkmale wie Geschwindigkeit, Speicherkapazität, Ausfallrate usw. und haben nur insofern Bedeutung für den Kreis der Anwendungen. Typische Geräte schaffen heute einige Millionen Instruktionen pro Sekunde und ähneln, ganz im Gegensatz zu beliebten Bebilderungen des machtvollen Elektronengehirns, in Größe und äußerer Gestalt üblichen Büromöbeln. Insbesondere wegen der Fortschritte der Halbleitertechnik sind sie in einem Preisverfall begriffen, dessen Ausmaß und Dauer in der Geschichte des Kapitalismus ohne Beispiel dastehen sollen, jedenfalls von einem stinknormalen Geldausgeber noch nie beobachtet werden konnten. Sie erlauben eine Vielzahl von Ein- und Ausgabegeräten, insbesondere die aus Bildschirm und Tastatur bestehenden Arbeitsplätze, und eine eindrucksvolle Fülle gleichzeitiger Arbeiten: Der eine Benutzer entwickelt neue Software, der zweite tippt Kundendaten ein, der dritte spielt Mondlandung, und zu Nutz und Frommen der Geschäftsleitung läuft außerdem noch ein Programm, das über die Aktivitäten der lieben Mitarbeiter Buch führt.

Eine derart "komfortable Benutzeroberfläche" kommt dem Computer nicht von Haus aus zu - im Gegenteil. Um ihn zweckmäßig verwenden zu können, sind erhebliche Kenntnisse seiner technischen Details, mathematische Bildung und Erfindungsgabe sowie peinlichste Sorgfalt bei der Bitfummelei erforderlich - mit anderen Worten akademische Grade und die Qualitäten von Schwachsinnigen: Wollte man das Ding über den Kreis seiner Erbauer hinaus verbreiten und zu einem Geschäftserfolg machen, war es nötig, einen Teil der Arbeiten, die aus dem Gebrauch eines Computers entspringen, wieder an diesen selbst zu übertragen. Statt den Anwender mit der nackten Apparatur zu konfrontieren, wird ihm mit Hilfe der *"Systemsoftware"* eine wesentlich leichter zu bedienende (aber immer noch universelle) Maschine dargeboten; das heißt, seine Programme werden nicht unmittelbar, sondern durch die Wirkung anderer Programme ausgeführt.

Das *"Betriebssystem"* übernimmt die Funktionen eines Operateurs und geht dabei weit über das hinaus, was ein Mensch zu leisten vermöchte. Es lädt, startet, überwacht Programme, regelt den Gebrauch von Betriebsmitteln wie Speicher oder Drucker und erlaubt erst, die Geschwindigkeit der Maschine durch Mehrprogrammbetrieb (scheinbar gleichzeitig verzahnte Ausführung) auszunutzen. *"Programmiersprachen"* und dazugehörige Übersetzungsprogramme steigern die Produktivität des Programmierers, indem sie ihn von der schier prohibitiven Mühsal und Fehleranfälligkeit der binären Codierung befreien. An die Stelle der eigentlichen Maschinenbefehle treten zunächst "aussagefähige Namen", die sich ein Mensch besser merken kann als die Bitkombinationen selbst, und des weiteren "höhere Sprachkonstrukte", die aus der Formelsprache der Mathematiker entlehnt oder aus den Erfahrungen und Bedürfnissen des Programmierens heraus erfunden wurden und vom Übersetzerprogramm in komplizierte Gebilde aus Maschinenbefehlen transfomiert wurden.

Das an die Verwendungen von Computern gern geknüpfte Fehlurteil, es handele sich um eine intelligente Maschine, findet an seinem durch die Systemsoftware vermittelten Erscheinungsbild eine Stütze. Namentlich bei den Programmiersprachen kann man sich einbilden, daß der Computer verstehe, was man ihm in den eigenen Worten sage. Dabei handelt es sich um ein Werkzeug des Programmierens, so "soft" es auch sein mag, da heißt, um ein Mittel, dieses Geschäft für das Subjekt zu erleichtern. Wenn für Programmiersprachen Elemente verwendet werden, die auch in richtigen Sprachen ihre Bedeutung haben, so darf man sich doch nicht gerade diese denken, und die Bedeutung für - oder besser Wirkung auf - den Computer, die der Programmierer lernen muß, ist von derselben Qualität wie die des Knopfdrucks, mit dem man den Strom abschaltet. Deshalb, gibt es bei Programmierneulingen stets die Enttäuschung, daß der Kasten, der so schön aufs Wort hört, nicht tut, was man gemeint hat, und noch die ältesten Hasen müssen sich über die Folgen banaler Schreibfehler ärgern, die ein Mensch auch beim dritten Hinsehen einfach überliest.

Als potentieller Alleskönner ist der Computer an sich zu nichts nutze; er benötigt *Programme*. Das Programmieren besteht darin, zu dem Zweck, dem ein Computer dienen soll, einen Algorithmus anzugeben und in einer Programmiersprache zu fixieren, das heißt, die jeweilige Vorgabe, ***was*** die Maschine leisten soll, muß übersetzt werden in ein Verfahren, ***wie*** sie es tun kann und soll. Die berüchtigte Schwierigkeit dieses Geschäfts, bei dem auch nicht laufend das Pulver erfunden wird, besteht darin, sich überaus komplexe und verschlungene Abläufe vorzustellen und ein Urteil über deren Effekt zu bewahren bzw. zu bilden. Mit dem Computer kommt so eine neue Sorte geistiger Arbeit in die Welt und des weiteren die "Softwarekrise", die nach allgemeiner Meinung noch bis ins nächste Jahrtausend den Engpaß der schönen Technik bildet. Das Problem ist weniger der Mangel an Personal als dessen Neigung, einen eigenen Kopf zu haben, wenn es sich denselben schon so "kreativ" zerbricht. Die Antwort auf das halbkünstlerische Gebaren dieser Leute heißt "egoless programming" und beinhaltet neben rigider Führung, Arbeitsteilung und Kontrolle die womöglich computergestützte Variation und Kombination früher erarbeiteter Programmstücke.

II.

Das typische Urteil über den Computer lautet, daß er die *geistigen* Fähigkeiten des Menschen erweitere, genauso wie die stoffumwandelnde Maschinerie sein körperliches Vermögen ausdehnte. Dieses Urteil täuscht sich über die Natur sowohl dieses Arbeitsmittels wie der Arbeitsprozesse, denen es dient.

Allgemein gilt für die Beziehung des Computers zum Geist, daß seine Domäne die ganz *gedankenlosen* Tätigkeiten sind, die aber auf seiten eines Menschen, der sie ausüben soll, allerhand Einsatz seines ***Hirns*** verlangen, Aufmerksamkeit, Gedächtnis, Kenntnis und Beachtung von Regeln. Auch diese Tätigkeiten sind beim Menschen, als Verrichtungen seiner Vorstellungskraft und seines Verstandes, anderer Natur als die Prozesse in einem Rechner. Sie lassen sich aber durch Schaltungen darstellen, ohne das Ergebnis zu verfälschen. Die Leistungsfähigkeit von Computern liegt also nicht daran, daß sie das Denken gelernt hätten, sondern hat ihren Grund in einer Eigentümlichkeit *der* theoretischen Leistungen, die bei der Rechenmaschine so gut aufgehoben sind. Das Gedächtnis ***ist*** zwar kein Speicher, sondern abstraktes Vorstellen von Sachverhalten, die man hinsichtlich ihnen wesentlicher Merkmale und Bestimmungen einmal zur Kenntnis genommen, also eben "gedacht" hat. Fürs Gedächtnis sind die gehabten Gedanken aber ein verfügbarer "Stoff", die getätigten ***Abstraktionen*** verwendbare ***Vorstellungen***, nicht jeweils von neuem zu ***denken***, sondern ***für*** allerlei theoretische Tätigkeiten "abrufbar". Die Mathematik ***ist*** zwar keine "Mechanik des Geistes", sondern die Erkenntnisleistung, die Abstraktion der Quantität in den ***ihr*** eigenen Bestimmungen zu entwickeln. Jede erkannte mathematische Regel - daher auch jedes mathematisch darstellbare Naturgesetz und jede daraus abgeleitete Technologie - läßt sich aber im Gedächtnis "aufbewahren" als vorstellungsmäßige ***Anweisung*** fürs Kombinieren quantitativer Bestimmungen. Das Rechnen lernt jedes Kind an seinen Fingern; und deren abstraktes Vorstellungsbild ist immerhin so vollständig zum Medium für die Anwendung einfacher mathematischer Regeln im Abendland geworden, daß mancher gebildete Mensch sich das Rechnen überhaupt nur als Dezimalsystem vorstellen kann. Umgekehrt erlaubt die Deckungsgleichheit von mathematischem Wissen und abstraktem Vorstellen etwas in jeder anderen Wissenschaft so Unmögliches wie mathematische *Genies*.

Das "Elekronengehirn" mit seinen binären Schaltkreisen macht ernst mit der von jedem Kind und jedem Rechengenie geübten Bequemlichkeit, die Mathematik in der Gestalt begriffsloser Kombinationsregeln zur Anwendung zu bringen. Und diese Bequemlichkeit hat ganz außerordentlich das Bemühen beflügelt, die verschiedenartigsten Gedächtnisinhalte - z.B. Vokabeln verschiedener Sprachen - in eindeutigen Zahlenwerten darzustellen und ihre jeweils zweckmäßige Anwendung in Kombinationsregeln für diese Zahlenwerte zu "übersetzen".

Das Ergebnis heißt zu Recht *"Datenverarbeitung"*: Es geht um das Aufbewahren, Zurverfügungstellen und Verknüpfen eigentlich numerischer und ähnlich abstrakt aufgefaßter Information. Sie operiert nicht mit Schlüssen, sondern Prüfungen auf Gleichheit und Ungleichheit; ihr Element ist nicht das Urteil, sondern der "Datensatz", d.h. eine Kombination von Angaben wie Beruf, Gehalt, Steuerklasse. Brauchbar sind solche Fähigkeiten des Computers in den verschiedensten Umständen; als epochemachend gilt er aber in einer Produktionsweise, in der *immaterielle Produktion* weitgehend *identisch* ist mit solcher *Datenverarbeitung*. Der Zweck der kapitalistischen Produktion, der abstrakte Reichtum, der Wert, hat nicht nur die elementaren geistigen Potenzen als Maschinensteuerung und -kontrolle von der köperlichen Arbeit getrennt, sondern auch die unmittelbare Produktion mit einem Drumherum von gigantischen Ausmaßen versehen. Das betriebliche Rechnungswesen verkörpert in seiner Selbständigkeit wie in seinen Inhalten den Standpunkt der Profitmacherei gegenüber der eigentlichen Fertigung; es wird gekauft und verkauft und überhaupt furchtbar viel Geld furchtbar viel gezählt mit all den Feinheiten des wann und wo, für was und für wen, die erst über einen gelungenen Überschuß entscheiden.

HEFT 29 Januar 1990

Der Computer kann es einem Menschen *ersparen*, seinen Kopf auf ganz mechanische Weise anzustrengen, wo das Nachdenken eher stört und zu Fehlern führt, und die Maschine übertrifft hier die Intelligenz lässig im Umfang der bewältigten Aufgaben. Daraus folgt nicht, daß *tatsächlich* irgendeinem Menschen irgendetwas erspart bleibt. Dank seiner Universalität legt ein Computer weit weniger als andere Maschinerie schon technisch fest, wie an und mit ihm gearbeitet wird. Namentlich im wissenschaftlichen Bereich findet man, daß der Rechenautomat als *Organ* und *Werkzeug* einzelner oder kooperierender *Subjekte* fungiert. Sie bedienen sich seiner souverän gemäß ihren wechselnden Zwecken; schreiben Programme selber oder wählen sie aus vorhandenen Bibliotheken; lassen rechnen oder benutzen die Elektronik als Karteikasten oder Vehikel der Kommunikation mit Kollegen, und sie ziehen es oft vor, ihren Publikationen gleich selbst mit einem Textsystem statt einer Sekretärin die endgültige Form zu geben.

Das glatte Gegenteil trifft man in den Niederungen der Wirtschaftswirklichkeit; wenn sich das Kapital des Computers bedient, so folgt, daß *viele Leute den Computer bedienen*. Das Grundprinzip aller kapitalistischer Produktion, daß nämlich "nicht der Arbeiter die Arbeitsbedingung, sondern umgekehrt die Arbeitsbedingung den Arbeiter anwendet" (Kal Marx), ist hier so handgreiflich wie in der industriellen Fertigung. Der ordinäre Bildschirmarbeiter, ob er nun Buchhalter oder Lagerverwalter war oder einen solchen ersetzt, sieht sich mit einem laufenden Programm konfrontiert, das ihm den Tag hindurch *vorschreibt*, was er zu machen hat. Nämlich den vom Programm benötigten Input, Daten oder gelegentlich Kommandos eines "Menues" in die Tastatur zu geben. Die Maschine subsumiert den Arbeiter als ihren *Handlanger*. Sie befreit ihn nicht von der Arbeit, sondern nimmt seiner Arbeit den Inhalt; erspart ihm Routinetätigkeiten, um ihm eine noch viel einförmigere und damit umso quälendere Routine aufzuzwingen; räumt Hindernisse und Friktionen aus dem Weg und eleminiert damit jede Abwechslung und Pause. Statt einer Erweiterung seiner geistigen Fähigkeiten erlebt er seine endgültige *Verblödung* und trägt noch allerlei Augen-, Rücken- etc. Beschwerden davon.

Erfunden wurde der Computer für den Staat, insbesondere für seine militärische Abteilung, und die ist noch heute, wo jede Klitsche auf Computer umstellt, der größte Sponsor und Motor der technischen Entwicklung. Zum Geschäftsmittel wurde der Computer durch die simple Entdeckung, daß sich seine automatische Arbeitsweise außer für die langen Berechnungen, wie sie bei der Fabrikation von Atombomben und Artillerietabellen anfielen, auch genausogut für eine Reihe kleinerer Berechnungen, wie sie zum Beispiel in der Lohnbuchhaltung vorkommen, verwenden läßt. Die *kommerzielle Datenverarbeitung* beginnt damit, die überkommenen *isolierten* Tätigkeitsfelder des Rechnungswesens an den Computer zu übertragen; es gibt Programme für Gehaltsabrechnung, Kundenkonten etc. und die entsprechende Datenhaltung. *Integration* heißt das Ziel fortan. Die verschiedenen Bereiche arbeiten mit denselben Daten, und an die Stelle von Dateien, die wie klassische Akten unter einem Verarbeitungsgesichtspunkt organisiert sind, treten *Datenbanken*, die eine Vielzahl solcher Sichtweisen erlauben. Die überkommene Arbeitsteilung wird obsolet; der typische "Vorgang", der von einem Schreibtisch zum nächsten wandert, kann ohne große Ansprüche an seinen Bearbeiter in *einem* Schritt erledigt werden, und dank der dem Computer einverleibten Fähigkeiten kann der Systemanalytiker die Informationsströme und Verarbeitungsflüsse eines Unternehmens ganz *ohne* Rücksicht auf die in einem Spektrum von Berufen festgeschriebenen Fähigkeiten *entflechten und neuordnen*. Das *Papier*, charakteristisches Mittel der Büroarbeit und zugleich Inbegriff ihrer Diskontinuität, wird in doppeltem Sinne zur *Randerscheinung*; und was

noch in Briefform nach außen fließen muß, erledigt die moderne Schreibkraft mit ihrem Textsystem: kaum Verzögerungen mehr durch Fehler, Revisionen und Kopien fürs Archiv, und statt dessen jede Menge fertiger Bausteine, deren Auswahl der Sachbearbeiter womöglich schon selber besorgt.

Die bloß *ideelle Einheit* manufakturmäßig geteilter Büroarbeit (ähnliches gilt für den Bereich Konstruktion) erhält *leibhaftige Existenz* in einem Computer, der mit seiner zentralen Datenhaltung und einem System von Programmen eine Fülle von Detailtätigkeiten an Sichtgeräten oder auch Kleincomputern treibt und zusammenfaßt.

Die Analogie zur Dampfmaschine als zentraler Beweger der alten Fabrik oder auch zum modernen Fließband ist deutlich, und eben weil die Unterwerfung des Arbeiters unter die Maschine und die Überflüssigmachung seiner Geschicklichkeit in der *materiellen* Produktion schon vollendet ist, findet der Computer hier zögernder Verwendung und wirkt *nicht mehr revolutionär*. Eigentliche Roboter, wiewohl Lieblingsikonen der Ideologie von der Abschaffung der Arbeit, sind relativ seltene Erscheinungen, beschränkt auf Fälle, die sich weniger durch die Tätigkeit selbst als deren für einen Menschen widrige Umstände empfehlen. Ein Roboter äfft den Arbeiter nach, der, indem er eine Maschine bedient, auf seine natürlichen Potenzen reduziert ist, auf Wahrnehmung, Muskelkraft und Koordination seiner Gliedmaßen, der also ganz einfache, unausgebildete Arbeit leistet. Aber für einen Computer gilt die Hierarchie der Berufe nicht: Es ist unendlich schwer, ihm zum Beispiel beizubringen, Schrauben aus einer Kiste zu greifen. Weit größere Bedeutung hat er - oft in spezialisierter Form - für die *Vervollkommnung konventioneller Maschinen oder Produktionssysteme*, die nicht auf der *Nachahmung* menschlicher Tätigkeit, sondern ganz *objektiver Analyse* der bezweckten Stoffumwandlung beruhen. Als Organ der *Prozeßsteuerung*, ob in Erdölraffinerien, Papierfabriken oder Flugzeugkanzeln (und in allem möglichen militärischen Gerät), leitet er aus Meßwerten die nötigen Korrekturen ab und bringt sie selbsttätig auf den Weg. Und auf Werkzeugmaschinen *verpflanzt*, vererbt er ihnen seine programmierte *Flexibilität* und erübrigt die fachmännische Einstellung nach Zeichnungen und Maßen.

Wie im Büro, so fungiert der Computer auch hier, wo er bereits vorhandene Automatisierung perfektioniert, als *verselbständigtes Hirn* der jeweiligen Aktivität, und deshalb bemüht man sich fleißig, die fundamentale Scheidung von Büro und Fabrikation zwar nicht aufzuheben, aber doch einem ganz neuen Maß von Integration zu unterwerfen: Die ideele Lagerhaltung kooperiert automatisch mit der leibhaftigen Bewegung von Beständen; Konstruktion und Kapazitätsplanung gehen direkt über in Maschinensteuerung.

III.

Mit dem Titel eines "Jobkillers" wird dem Computer und seinen diversen Einkleidungen zugeschrieben, in beispielloser Weise Arbeit überflüssig zu machen und de facto abzuschaffen. Damit bestände an und für sich Grund zum Jubel - denn wo Arbeit *nicht nötig* ist, muß *Reichtum* herrschen. Aber dieser Schluß existiert heutzutage noch nicht einmal als naiver Ingenieuroptimismus: Die Technik dient so hohen Zwecken wie der nationalen Konkurrenzfähigkeit und läßt für die Menschheit nichts zurück als *Probleme*. Insbesondere das Problem, daß die Mehrheit der Leute zu nichts mehr nutze ist, aber leider nicht in persona abgeschafft werden könne, sondern Betreuung brauche.

Der harte Kern der Jobkillerei spielt bei solchen Menschheitsfragen keine Rolle, die Tatsache nämlich, daß mit dem Arbeitsplatz seinem Inhaber der *Lebensunterhalt* genommen wird. So wenig diese Konsequenz die *Wirkung* eines technischen Gerätes sein kann, so wenig ist es überhaupt wahr, daß der Computer *per se* Einfluß auf die Arbeitswelt hätte: Dies bleibt seinem Herrn und Meister vorbehalten. Die Anwendung des Computers durch das Kapital soll Kosten senken, d.h. nicht Arbeit *überhaupt*, sondern *bezahlte* Arbeit einsparen, Arbeit als Anspruch auf Lohn. Dieser Rationalisierung genannte Zweck hat zwei Konsequenzen. Erstens werden Arbeiter *freigesetzt*, ob sie nun massenhaft auf die Straße fliegen oder "bloß" entdecken müssen, daß das Wachstum ganz ohne Nachfrage nach Personal blendend vorankommt. Die Beweisführung, daß, was vom Computer an Arbeit verdrängt wird, auch durch ihn wieder Arbeit erhalte, ist eine apologetische Milchmädchenrechnung: Es handelt sich bei solcher Nachfrage nicht nur um ganz andere Berufe, sondern es werden auch nicht an der einen Stelle Lohnkosten gespart, um sie als Teil der Maschinenkosten an anderer Stelle zu zahlen.

Und eben weil zweitens der Angriffspunkt der Rationalisierung der Lohn ist, heißt ihr Resultat *profitablere Ausnutzung der Arbeit*. Tätigkeiten, die zuvor nicht jeder ausüben konnte, werden jetzt von billigen, ungebildeten Kräften erledigt. Wenn die Maschine die Arbeit erleichtert, so erlaubt sie auch deren *intensivere* Verausgabung, ja dient als Mittel, diese zu erzwingen. Mit der alten Arbeitsteilung verschwindet manche *technische Notwendigkeit der Kooperation*, und dem Kapital eröffnen sich neue *Freiheiten*, Zeit und Ort der Arbeit nach seinen eigenen Bedürfnissen zu gestalten.

Die ganz andere Sorge aufgeklärter Zeitgenossen gilt ganz direkt der Fähigkeit des Computers, große *Datenmengen* nutzbar zu machen. Bei betrieblichen Informationssystemen bei Banken, Versicherungen und vor allem beim Staat befürchtet man, daß der *"verdatete Bürger"* Objekt von Maßnahmen wird, die ihm nicht passen können. Dabei wird die Legitimität solcher Institutionen und Interessen *nicht* bestritten und Kritik an ihnen kommt gar nicht erst auf. Der absurde Gedanke, mit dem ganz wohlmeinend Bedenken angemeldet werden, heißt Aushöhlung bestehender Rechte durch technischen Wandel oder auch Machtverschiebung, die eigentlich nicht so gemeint war. Beispiele dafür lassen sich ernsthaft nicht geben - es soll ja nicht bös' gemeint sein -, und deshalb erfolgt regelmäßig der Übergang zu den *Möglichkeiten*, die ein Hitler heute hätte.

Der Staat gebraucht Computer natürlich nicht zur Machtverschiebung - er hat schon die Macht -, sondern um die *Wahrnehmung seiner Aufgaben*, die ihm niemand bestreiten will, zu *verbessern*. Ein schul-, wehr-, steuer- und sonstwie -pflichtiger Bürger ist schon immer in mannigfacher Weise verdatet gewesen. Und was für die guten Bürger gilt, gilt für die schlechten erst recht, und für Verbrecher und Kommunisten waren jede Menge Akten und Karteien nötig. Was neu ist, ist die Leichtigkeit, solches Material verfügbar zu machen und zu verwalten. Datenbanken und Computernetze sparen nicht nur Zeit und Kosten, sondern vergrößern auch den Erfolg mancher Anfrage; und Informationen, die - weil räumlich und organisatorisch getrennt - nie aufeinander bezogen worden wären, finden jetzt ganz automatisch zueinander.

Das Unbehagen am Computer argumentiert mit einem *Quidproquo*: Negative Wirkungen, die die *Anwendung* des Gerätes hat, werden als die *seinen* aufgefaßt, bloß weil sie *ohne* es nicht existierten. Der Fehler dieses Arguments ist der der Maschinenstürmer zu Beginn der Industrialisierung - bloß *bekämpften* diese in der Maschine - also falsch - *das Kapital*, während heute

von solcher Gegnerschaft nichts mehr zu spüren ist. Die Sorge um die Zukunft ohne Arbeit, aber mit allwissendem Großen Bruder ist eine Manier, die Gegenwart für sehr *gemütlich* zu befinden. Deshalb fällt es diesen Mahnern nicht im Traum ein, die Zwecke zu kritisieren, die sich mit dem Computer realisieren, und sie gehen konsequent dazu über, *an ihm selbst* bedenkliche Züge zu entdecken. Er ist keine *"menschengerechte"* Technik. Zu kompliziert und voller Eigenleben, kann er nicht wirklich verstanden und beherrscht werden. Und an der Möglichkeit des "Kriegs aus Versehen" soll man ermessen, was es heißt, sich auf diese Dinger einzulassen. Als ob diese Möglichkeit nicht zuallererst eine kriegerische Situation *voraussetzte*!

Genauso schlimm wie der "Computer, der durchdreht", ist der "Computer, der nur funktioniert". Bloße Realität und Logik werden ihm attestiert, als ob diese Attribute, die ihm übrigens nicht zukommen, zu einem Vorwurf taugen. Was ihm - wie "vorprogrammiert" - abgehen soll, sind Kreativität, Problembewußtsein und innere Werte. So hat das Ding immerhin das Gute, die Arbeitsplätze von Pfaffen, Professoren und anderen Fernsehstars zu garantieren. Wir meinen allerdings: Auch daran ist nicht der Computer schuld.

aus: Hamburger Hochschulzeitung, Marxistische Gruppe, Sonderausgabe Informatik, Novenmber 1989

- I M P R E S S U M -

75

1. Vorsitzender — **Alexander SCHMID**
St.Cajetan Str. 38 VII
8000 München 80
☎ 089 /495326

2. Vorsitzender — **Horst-Dieter SCHROERS**
Breslauer Str. 9
8016 Feldkirchen
☎ 089 /9032615

Hardwarekoordinator — **Andreas MAGNUS**
Pommernstr. 4
4650 Gelsenkirchen
☎ 0209 /870230

NEWDOS-Diskothekar — **Oliver VOLZ**
Waldburgstr. 73
7000 Stuttgart 80
☎ 0711 /7353817

CP/M -Diskothekar — **Rüdiger SÖRENSEN**
Thomas-Mann-Str. 3a
6500 Mainz 1
☎ 06131 /32860

Club-Bücherei — **Christian MENK**
Ollsener Str. 52
2116 Hanstedt
☎ 04189 /78251

Redaktion — **Jens NEUEDER**
Gschlachtenbretzingen
Rudolf-Then-Straße 32
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto
Sonderkonto CLUB 80
Obermann,H. 6209 Heidenrod
Konto-Nummer 496 071 - 605
Postgiroamt Frankfurt
BLZ 500 100 60

Eine Zensur oder Kontrolle der Infobeiträge erfolgt nicht.
Die Redaktion

Schluss

76

Hallo Club-80er,

nachdem nun der Jahreswechsel erfolgreich -das hoffe ich für jeden von Euch- hinter uns gebracht ist, erscheint nun wie versprochen unser nächstes CLUB-INFO.

Für die geleistete Mitarbeit an unsern CLUB-INFO's möchte ich mich bei allen Mitwirkenden recht herzlich bedanken.

Ich hoffe und wünsche uns, daß es im Neuen Jahr noch besser weiter geht und ich Euch regelmäßig ein "sattes" INFO erstellen kann.

Als nächsten Redaktionsschluß möchte ich Mitte/Ende Februar ansetzen, sodaß ich noch vor unserer Jahreshauptversammlung das nächste, das 30., Club-INFO zusenden kann.

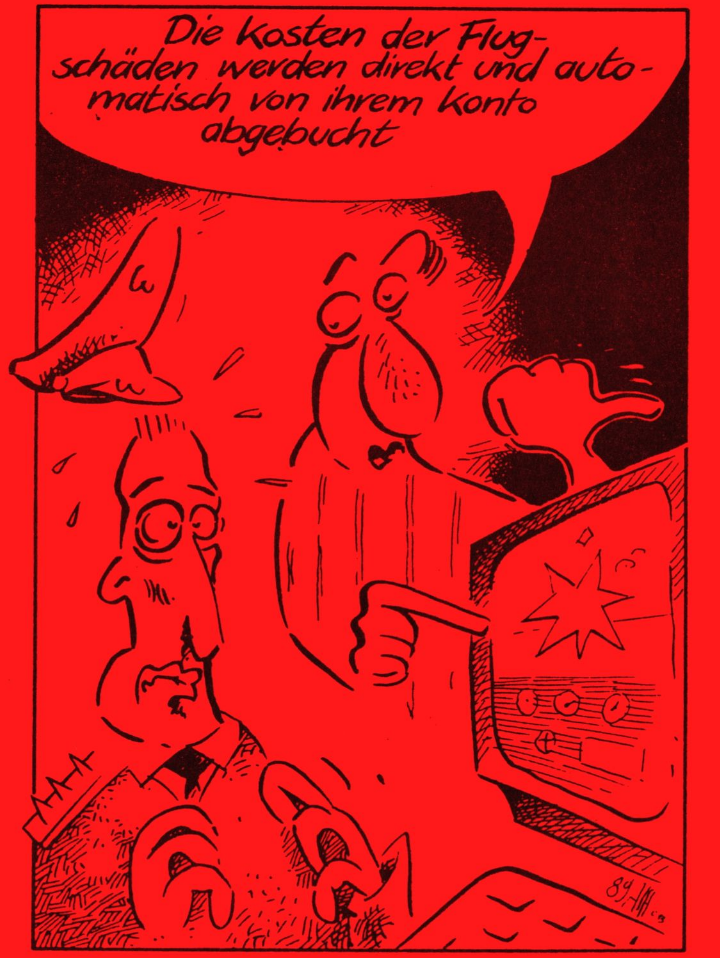
Als Beilage zu diesem INFO findet Ihr einmal das Sonderinfo
HD 64180
sowie eine aktuelle Inhaltsübersicht

Ich wünsche Euch zum Abschluß Spaß am neuen INFO,
"happy computing 90"
und einen "Guten Rutsch" gehabt zu haben.

Es grüßt Euch Euer

Jens

"Bitte beachtet den Überweisungsträger, für den CLUB-Beitrag 1990, der diesem INFO beiliegt."



CLUBINFO

30. AUSGABE

KONTAKTADRESSE : CLUB 80 / ALEXANDER SCHMID / ST. CAJETAN STR. 38 VII / 8000 MÜNCHEN 80
TEL.: 084 / 445326

- I N H A L T S V E R Z E I C H N I S -

Seite:
und Autor:

Clubinternes

Neues vom Vorstand 01
Alexander Schmid
Gedanken zur CLUBINFO 02
Oskar Drechsler
Vorstellung 03
Jens Günther
Messen '90 03
Termine 04
Redaktion
Vorschlag für nächste Clubtagung ... 04
Oliver Volz
Club international .... 05 - 06
Alexander Schmid, Uwe Brunjes
BASIC-Tricks und Tips für Anfänger 07 - 08
Oskar Drechsler

Software

Diophantisch .... 09 - 10
Klaus-Jürgen Mühlenbein
Dotprint mit EPSON LQ 550 .... 11 - 16
#Lprint mit EPSON LQ 550 .... 17 - 23
Oskar Drechsler
Programme ohne Grenzen 26
dBASE II und die Hardwareuhr .... 27 - 28
Das Turbo-Monster 28
Platz sparen mit WordStar 29
Nachtrag zum Banker 30
Port-I/O die Zweite 31
Alexander Schmid
Ergänzung zu RGB-Modifikation von RB 31
Oskar Drechsler

Seite:
und Autor:

Hardware

Turbotape, der Jedermann-Streamer .. 32
Alexander Schmid

Börse

Wer hat was -- wer will was .... 33 - 34

Sonstiges

Mikrocomputer-Betriebssysteme -5 ... 35 - 39
Gerald Schröder

Die letzten Seiten

Impressum 41
Schluß 42
Redaktion
Mitgliederadressenliste .... am INFO-Ende
Alexander Schmid
MC-Inhaltsübersicht .... am INFO-Ende
THE PUBLIC SOFTWARE LIBRARY ...   am INFO-Ende

Neues vom Vorstand

Ich hoffe, daß KaJot mit diesem Info ein dicker Strich durch seine Nullfolge gemacht wird und daß das Info in Zukunft wieder regelmäßiger erscheinen wird. Da nach der Jahreshauptversammlung wohl ein neuer Steuermann das Ruder übernehmen wird, sind die Voraussetzungen dafür ja günstig. Schön, daß doch so viele optimistisch geblieben und nicht abgesprungen sind. Bei dieser Gelegenheit möchte ich posthum noch ein Danke an die sagen, die wenigstens eine Austrittserklärung geschickt haben und sich nicht noch freuen, daß sie noch ein paar kostenlose Infos auf Clubkosten bekommen. Eine einfache Postkarte reicht doch und erspart uns eine Menge Arbeit. Zum Zeitpunkt des Schreibens (Mitte Februar) sind wir immerhin (oder nur noch ?) 73. Nicht schlecht, für einen Rechner, der seinen Zenit schon um ein paar Tage überschritten hat. Wenn wir uns anstrengen überleben wir noch die MSDOS-Welle.

Wahrscheinlich wird es wie jedes Jahr wieder eine Diskussion über MSDOS usw. geben, aber ich glaube, daß wir mit der bisherigen Regelung ganz gut gefahren sind. Gegenteilige Meinungen können in der Diskussionsrunde vorgetragen werden. Apropos Diskussionsrunde, viele Anregungen sind bis jetzt noch nicht gekommen, scheinbar ist soweit alles in Ordnung, auch für Referate und Vorträge sind nur 1 1/2 Vorschläge da. Winterschlaf oder Frühjahrsmüdigkeit ? Nur ein Punkt wurde auf den Anmeldungen ziemlich oft erwähnt, nämlich das spärliche Erscheinen des Infos. Das gibt sicher einiges an Gesprächsstoff her, wie wir dieses Problem am besten lösen können. Dem Vorstand samt Redaktion die Schuld zu geben ist natürlich relativ leicht, wenn auch recht unergiebig. Jens schreibt die Artikel doch nicht selber, was soll er machen, wenn er kein Material hat ? Eine Möglichkeit wäre vielleicht, den Beitrag pro geschriebenem Artikel zu senken und/oder für die "Nur-Konsumenten" anzuheben, aber ob das was bringen würde und wie weit das zu verwirklichen wäre, bleibt die Frage. Lassen wir uns überraschen. Strengt Eure grauen Zellen an und bringt ein paar Ideen mit.

Vom "Casale" habe ich inzwischen einen Prospekt bekommen und muß sagen, daß die Landschaft dort wirklich schön ist. Wer nicht nur hacken will, dem sei eine Erkundung der Umgebung wärmstens empfohlen. Es gibt dort u.a. die einzige Kristallhöhle Deutschlands, einen Wildpark und einen wunderschönen Renaissance-Schloßhof. Zum besseren Finden kann ich noch folgendes sagen: Zuerst Richtung Stadtmitte fahren, dann geht es über eine Brücke in die "obere" Stadt und da in der Gegend muß es dann irgendwo sein. Noch eine Bitte an die Frühaufsteher: Seid nicht böse, wenn ich um 10 Uhr noch nicht freudestrahlend vor dem Hotel auf Euch warte, aber ich habe ungefähr 4-5 Stunden Anfahrtzeit und bis jetzt hat noch keiner darauf bestanden, daß ich schon am Freitag anreise. Räumt schon mal Eure Zimmer ein, sucht Euch die schönsten Plätze für Eure Rechner und macht es Euch gemütlich.

Alexander Schmid

Gedanken zur **Clubinfo** von Oskar Drechsler

Soeben habe ich die 29. Clubinfo erhalten und muß mal meine Gedanken zum Dauerthema Clubinfo verstreuen.

Auf Seite 5 und 6 hat KaJot einen Rundumschlag verteilt, der vielleicht berechtigt ist. Ich bin zwar erst seit ca. einem Jahr Mitglied, finde aber, daß unser Club doch recht aktiv ist. Wenn man bedenkt, daß unser Club über die ganze BRD verteilt ist und die meisten sich nicht persönlich kennen, läuft der Laden doch ganz gut.

Ein Club wie unserer lebt nur durch die Clubzeitschrift und die entsteht durch die Aktivität der Mitglieder. Aber jeder Club hat nun mal viele passive und nur wenig aktive Mitglieder. Damit müssen wir uns abfinden.

Manche sind Neulinge, die sich vom Club Informationen erhoffen und nur lernen wollen und können. Von ihnen ist nichts zu erwarten. Ich erinnere mich noch sehr gut an meine Lehrjahre der Computerei. In dieser Zeit habe ich nur (Informationen) genommen. Geben konnte ich nichts.

Dann gibt es noch die, die glauben, daß die anderen viel mehr wissen und nicht den Mut haben, was von sich zu geben. Denkt doch mal an die vom vorherigen Absatz! Anfänger können mit großen Projekten und Strukturen nicht viel anfangen. Sie benötigen jede Menge kleine Tips und Informationen.

Neue Mitglieder haben nicht die alten Hefte (ich auch nicht). Warum soll man nicht ein Thema nochmal ansprechen? Auch das 21. Programm zur Berechnung des Benzinverbrauchs unterscheidet sich von den anderen 20 Versionen und enthält vielleicht eine Routine, die man woanders sehr gut brauchen kann.

Warum (lieber Jens!) soll man nicht mal einen Beitrag aus einem alten Heft, das neue Mitglieder nicht haben, wieder abdrucken? Das Fernsehen macht das seit Jahren mit der Begründung, daß immer wieder junge Menschen nachwachsen.

Müssen wir unbedingt alle 2 Monate eine Info herausbringen? Die Anzahl der Hefte pro Jahr ist keine Prestige-Frage. Der Inhalt und der Wert der Informationen ist wichtig. Da wir alle keine Journalisten oder Redakteure sind, ist es nicht einfach, einen guten Gedanken verständlich auf Papier bzw. in den Speicher zu bringen. Man braucht seine Zeit dafür. Nebenbei haben wir alle noch einen Beruf und/oder Familie.

Mir würde es genügen, nur alle 3 Monate eine Clubinfo zu erhalten. Dann kann auch der bereits 2 Wochen nach Erhalt einer Info gesetzte Redaktionsschluß hinausgeschoben werden. Dies hätte den Vorteil, daß, wenn man auf einen Artikel reagieren möchte, auch genügend Zeit hat, was Fundiertes vom Stapel zu lassen.

>> Hallo Club-ACHTZIG-Members,

als neues Mitglied möchte ich mich hiermit kurz vorstellen:

Name:	Jens Günther
Alter:	19
Adresse:	Bannerscheidt 7 5231 Neitersen Telefon: 02681 / 1553 (oft ab 17 h)
Ausbildung als:	Schornsteinfeger
System:	GENIE II Doppellaufwerk 40 Spuren SS/DD und DS/DD EPSON FX-80 Crt 4 A-HiRes-Grafik Genie-DOS 2.2

Programmiere hauptsächlich in BASIC bzw. Z-BASIC, selten auch in Assembler. Hauptinteressen liegen in den Bereichen: Textverarbeitung, Grafik, Utilities und Spiele (hier besonders Adventures und sog. Rollenspiele, aber auch sog. "Phaser-Games"). Software- und Ideenaustausch ist immer willkommen!

Weitere Hobby's: Sci-Fi (mainly CYBERPUNK)
Independent-Music ... <<

Viele Grüße,

Jens Günther

-- Messen '90 --

Telematica .... Stuttgart .... 11.06. - 14.06.
electronica .... München ...... 06.11. - 10.11.

Sicher schon allen längst bekannt !!
neue Adresse von Hartmut:
Hartmut Übermann
Mozartring 23
Postfach 1430
<u>8870 Günzburg</u>

Hallo Clubkameraden oder sind wir schon Vereinskameraden?

nach dem sich das Jahr dem Ende neigt und das letzte Clubtreffen in vier Monaten auch schon ein Jahr her ist würde ich vorschlagen, daß man sich um das 90er-Treffen kümmern sollte. Beim 89er-Treffen hat unser Horst-Dieter Schroers mir gesagt er kenne eine Gaststätte (Hotel) in dem man das Treffen ausrichten könnte. Vielleicht haben andere Mitglieder auch noch einige Vorschläge zu möglichen Treffpunkten. Der Letzte war vom Aufenthaltsraum mittlere Preislage aber die Zimmer fand ich nicht besonders. Ich glaube desshalb sind auch die Frauen die im Jahr davor dabei waren nicht wieder gekommen. Für den Fall, daß Niemand einen besseren Vorschlag macht hätte ich einen der besonders den Nordlichtern gefallen wird. Ich war geschäftlich schon öfters im Emsland, genauer gesagt in Sögel. Wer sich unter Emsland nichts vorstellen kann dort gibts die Magnetschwebebahn und viele "glückliche" Kühe. Zum Hotel die Tagungsräume sind für 200 Personen reichhaltig (kann verkleinert werden) das Essen war hervorragend die Zimmer schön. Die Preise sind tragbar, näheres müßte ich erst erfragen. Die Zugverbindung ist nach Lathen sehr gut. Lathen liegt an der Eisenbahnstrecke Köln-Norddeich Eilzugstation. Ich weis die "Südlichter" schreien jetzt mordieo sollten Sie aber nicht sondern einen besseren Vorschlag machen. Ich will auch nicht unter allen Umständen so weit fahren, aber lieber nach Sögel als nach Idstein.

-- Termine -- Termine -- Termine --

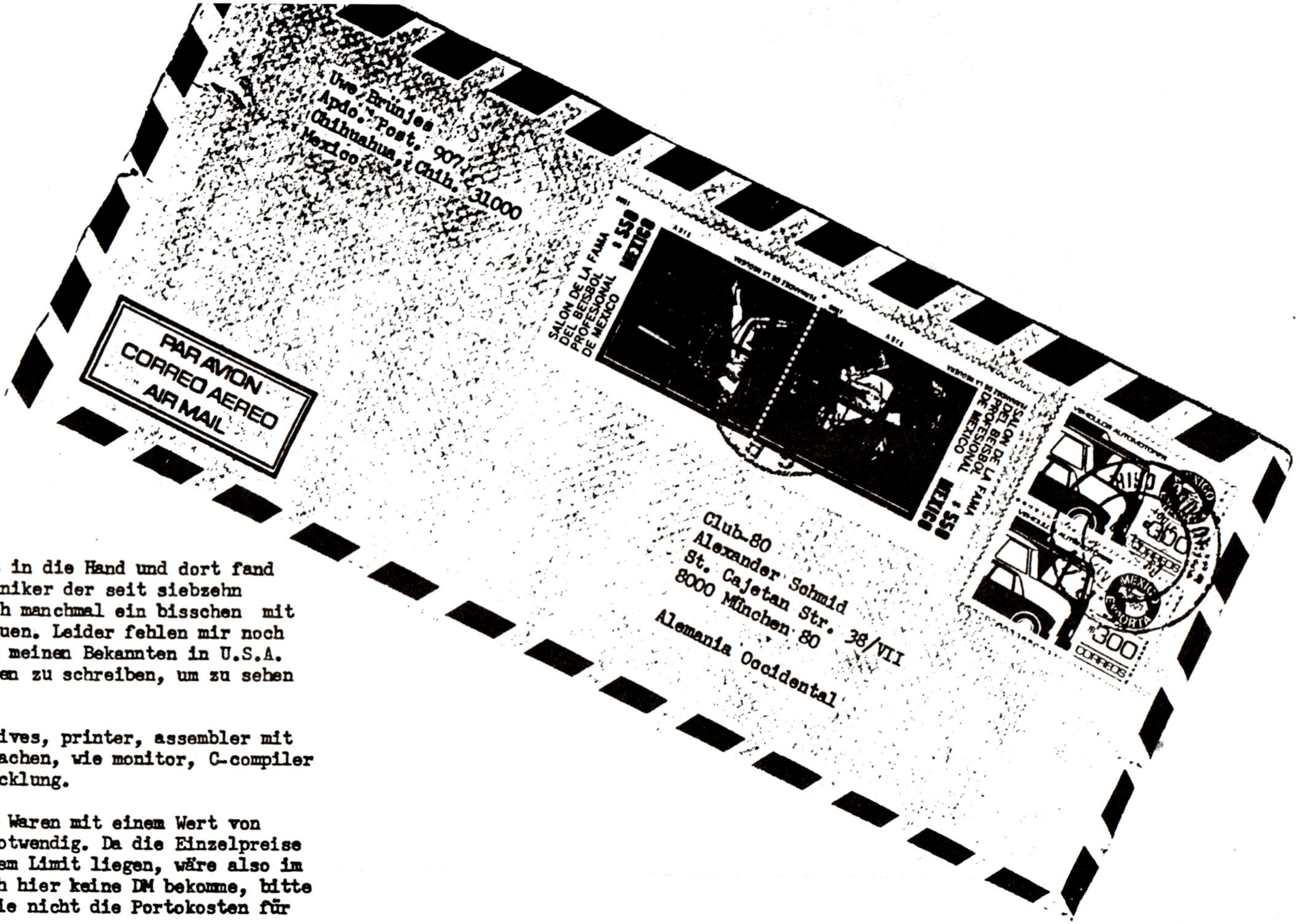
Nächster Redaktionsschluß .... Mitte / Ende Juni

Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!

Club international

Es ist erstaunlich, wie weit sich der Club-80 inzwischen rumgesprochen hat. Wie Ihr an dem Brief sehen könnt, sind wir schon bis Mexiko bekannt.

Leider hatte ich mit dem Model II noch nichts zu tun. Wenn jemand meint, daß er Uwe Brünjes helfen kann, scheibt entweder mir, damit ich es weiterleiten kann, oder gleich direkt über den Großen Teich.



Uwe Brünjes
Apdo. Post. 907
Chihuahua, Chih. 31305 000
Mexico

8. Februar 1990

Club-80
Alexander Schmid
St. Cajetan Str.38/vll
8000 München 80

Liebe Computer-Freunde.

Zufällig fiel mir vor Kurzem eine alte Ausgabe von c't in die Hand und dort fand ich Ihre Adresse. Ich bin ein ehemaliger Computer-Techniker der seit siebzehn Jahren in Mexico lebt und in meiner Freizeit spiele ich manchmal ein bisschen mit meiner TRS-80, die ich versuche aus Resten zusammenzubauen. Leider fehlen mir noch einige Geräte, die ich bisher weder in Mexico noch bei meinen Bekannten in U.S.A. erhalten konnte. Daher fand ich es eine gute Idee, Ihnen zu schreiben, um zu sehen ob Sie mir weiterhelfen können.

Mir fehlt fasst alles: ein disk-controller mit zwei drives, printer, assembler mit Dokumentation (auf diskette) und wenn möglich andere Sachen, wie monitor, C-compiler und andere Hilfsmittel für hardware- und software-Entwicklung.

Zur Zeit ist die Einfuhr nach Mexico sehr einfach: für Waren mit einem Wert von weniger als U.S.$ 300.00 sind keine Zollfarmalitäten notwendig. Da die Einzelpreise von gebrauchten Geräten offensichtlich weit unter diesem Limit liegen, wäre also im äussersten Fall nur getrennte Sendung notwendig. Da ich hier keine DM bekomme, bitte ich Sie, mir die Preise in U.S.$ anzugeben. Vergessen Sie nicht die Portokosten für versicherte Postpakete.

Viele Grüsse aus Mexico.

Uwe Brünjes

P.S. Meine TRS*80 ist Modell II

Basic-Tricks und Tips für Anfänger ?

Oskar Drechsler

In den letzten Club-Info's war aus einigen Vorstellungen von neuen Mitgliedern der Wunsch zu lesen, daß sie sich vom Club Informationen und Hilfestellung für die ersten oder weiteren Schritte in Basic erhoffen.

Dieses Thema möchte ich gerne aufgreifen, weil ich ein leidenschaftlicher Anhänger von Basic bin. Ich bin es, obwohl ich seit ca. 2 Jahren auch sehr gerne und ausdauernd in Assembler programmiere. Zwischendurch habe ich auch schon mal was in Fortran und Alcor-Pascal gemacht und mich mal etwas in PL/I eingearbeitet. Dienstlich habe ich mal 2 Jahre an einem Großrechner mit einer Art Abfragesprache ca. 10.000 Zeilen 'programmiert' und in diesem Jahr habe ich ca. 3000 Zeilen auf einem AT mit dem Programm-Paket Smart (wer es nicht kennt, hat nichts verpaßt) geschrieben. Ich glaube also schon zu wissen, warum ich Basic so gerne mag und welchen Wert es hat.

Den letzten Anlaß, mich mal über Basic auszulassen, war eine Äußerung eines Kollegen, der im Augenblick einen Anfängerlehrgang in Cobol macht. Kaum daß er weiß wie man 'programmieren' schreibt, hat er über Basic was von Spaghetti-Code gebrabbelt.

Ich halte Basic (und hier meine ich nicht z.B. das Basic vom C 64) für die Programmiersprache, die man am schnellsten erlernen kann, die am fexibelsten ist und mit der man wirklich alles machen kann, solange man nicht im Betriebssystem des Rechners rumfummeln will (Kleinigkeiten gehen auch) oder großen Wert auf hohe Verarbeitungsgeschwindigkeit legt. Letzteres spielt nur bei großen Datenmengen eine Rolle oder bei hochauflösender Grafik.
Sicher verleitet Basic zum schludern. Aber das ist so wie mit einem Auto. Manche fahren damit, manche bewegen es nur.

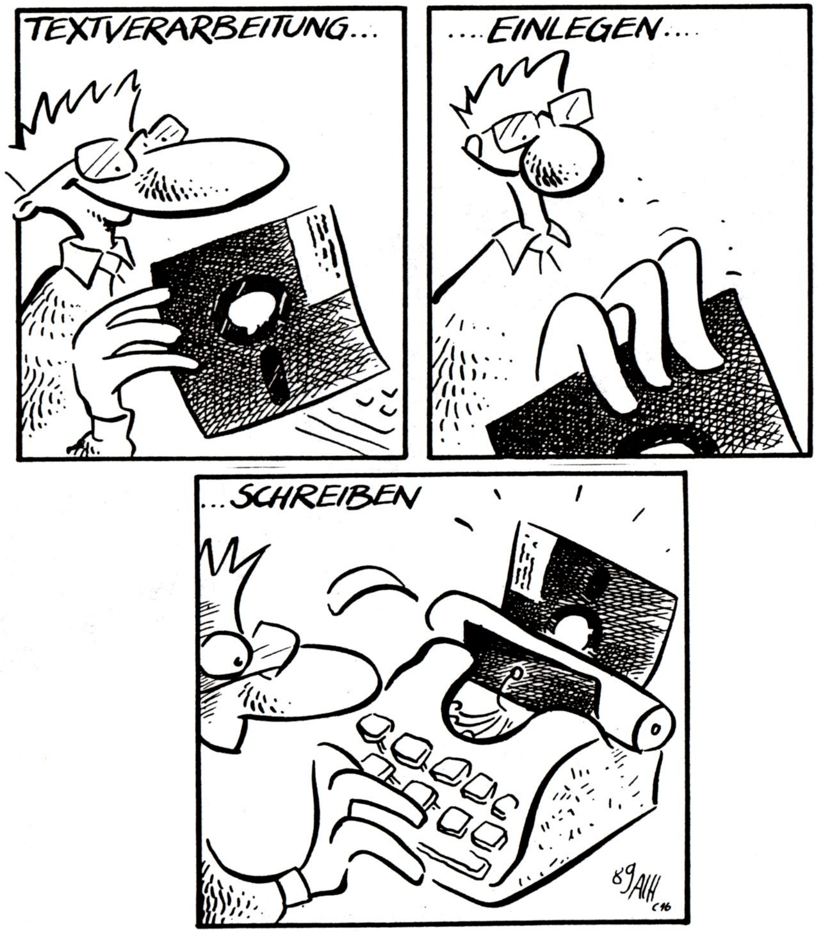
Ich habe nun vor, in jedem Info einen Beitrag zu bringen. Die Themen stelle ich mir wie folgt (grob) vor:

1. Beispiel für eine Programm-Struktur
2. Dynamische Variablen-Verwaltung des Microsoft-Basic und Verminderung/Vermeidung von Garbage
3. Geplante Verwendung von Variablen und sonstige Tricks zur Erhöhung der Verarbeitungsgeschwindigkeit
4. Interessante Funktionen aus 'Basic faster & better' (das Buch ist seltener vorhanden als man glaubt)
5. Verwendung von 'AND', 'OR' und Simulation von 'EXOR' zur Manipulation mit Bit's und sonstige Trick's, um unendliche 'IF ... THEN' Abfragen zu vermeiden.

Und sonstiger Kleinkram, den ich aus Büchern, Zeitschriften habe, bzw. von mir erarbeitet wurde.

Dies erfordert natürlich viel Zeit, die ich nicht umsonst verschleudern will.
Ich bitte deshalb diejenigen, die an diesem Thema Interesse haben, mir eine Postkarte zu schicken und mir dies mitzuteilen. Ich glaube, das ist nicht zuviel verlangt.
Meine Freizeit ist im Augenblick nämlich sehr knapp. Außerdem habe ich Helmut Bernhardt versprochen, ihm bei seinem Projekt zu helfen, das mir auch sehr am Herzen liegt. Allerdings muß ich mich dazu erst mal gründlich in CP/M einarbeiten und meine Assembler-Kenntnisse vertiefen. Da will ich natürlich wissen, ob es sich lohnt, nebenbei noch die Beiträge für Basic zu schreiben.

Also, auf geht's. Der Post geht's auch nicht so gut, daß sie auf Euer Porto verzichten kann.



* * * DIOPHANTISCH * * *

Weder ELE- noch OTTI-, sondern DIO-PHANTisches ist heute dran! Man lehrte uns, daß zur Berechnung von Unbekannten angeblich soviele "Gleichungen" erforderlich sind, wie Unbekannte zu berechnen sind. Doch: (Ziemlich) weit gefehlt! Jedenfalls, was die Formulierung betrifft. Es muß heißen: "Es sind soviele Bedingungen nötig wie Unbekannte"!
So lassen sich z.B. zwei Unbekannte auch aus einer Gleichung berechnen, wenn außerdem noch wenigstens eine Bedingung gegeben ist.
Dies erkannte schon vor 1740 Jahren (wie immer!) ein Grieche: * *DIOPHANTOS*! * Ihm ist dieser kleine Beitrag gewidmet. Er blickte mir beim Programmieren seines Algorithmus (von "oben" oder "unten"? Wo ruhen sich die Mathematiker nach kurzem Hiersein für immer aus?) über die Schulter und schimpfte kräftig, als er sah, daß ich noch nicht einmal das Unterprogramm für einen *Kettenbruch* parat hatte!!
Der Algorithmus für zwei Unbekannte X und Y ist nur e i n Beispiel für die sog. Diophantischen Gleichungen. Damit soll es dann auch sein Bewenden haben. Programme für mehr als zwei Unbekannte (und entsprechend mehr Bedingungen) mag ein anderer entwerfen, der mehr von Mathe versteht...
Die die Gleichung für zwei Unbekannte:

Ax + By = C

ergänzende Bedingung lautet: Die auftretenden Größen A,B,Cx,y müssen ganze Zahlen sein. Außerdem (kleine Nebenbedingung) müssen A und B teilerfremd sein. Das ist schon alles - und hört sich sehr einfach an!

(Anm.: Auch negative ganze Zahlen sind erlaubt. Man nennt diesen Zahlenbereich einfach "Ø".)

Aber sooo einfach ist das auch wieder nicht. Finde mal die passenden ganzzahligen Werte von X und Y, so daß die Gleichung stimmt, z.B. für

31X - 164Y = 7

Das ist eine ganz schöne Herumprobiererei, über die leicht ein Nachmittag oder - eine Nacht vergeht! Dabei gibt es - hier sei es verraten - sogar unendlich viele Lösungen, also Wertepaare X,Y!
Um sie zu berechnen, muß der Kettenbruch für X/Y gebildet werden, sodann für Zähler und Nenner des vorletzten Näherungsbruches die passenden Vorzeichen gefunden werden, so daß die linke Seite obiger Gleichung zunächst gleich Eins wird, woraus sich dann X und Y nach Formeln ergeben, die *Diophantos* unvergessen machen...
Die Entwicklung des Kettenbruches geschieht in den (nur!) drei Zeilen 90-110. In der anschließenden Schleife 120-140 werden die richtigen Vorzeichen für Zähler und Nenner bestimmt. Danach wird durch einen optionalen Parameter P der Umstand berücksichtigt, daß es unendlich viele Zahlenpaare X,Y gibt, die die Gleichung erfüllen (Zeilen 150/160). Ab Zeile 170 werden am Programmanfang vorgenommene notwendige Vertauschungen wieder rückgängig gemacht und schließlich die Ergebnisse angezeigt. *
Eine Kleinigkeit noch für den Anfänger:
Der ganzzahlige Teil einer Dezimalzahl wird bekanntlich durch "INT(X)" gefunden. Dies stimmt aber nur bei positiven Zahlen. Bei negativen erhält man mit diesem Befehl einen um 1 zu niedrigen Wert. Für solche Zahlen dient der wohl weniger bekannte Befehl "FIX(X)" (s.Zeile 90). Er gibt nicht wie der INT-Befehl die größte ganze Zahl, die kleiner als die Dezimalzahl ist, sondern die kleinste ganze Zahl wieder, die größer als diese ist - wie es bei negativen Zahlen ja sein muß. (Unklar? Dann BRANCH==> IF NOT: Einfach mal hinschreiben! * IF NOT NOT: OK!)
Viel Spaß beim Spiel mit dem OTTI- ach nein: dem DIO-phanten!

KaJot


```
1 '*****************************************************************
2 '*   Filename:             D I O P H A N T                      *
3 '*        <C> K.-J. Mühlenbein, Weinheim, 10.02.90              *
4 '*   Beschreibung :                                              *
5 '*   Für die Berechnung von z w e i Unbekannten steht nur        *
6 '*          e i n e  Gleichung zur Verfügung.                     *
7 '*   Die Lösung wird möglich durch die Nebenbedingung,           *
8 '*   daß alle Größen ganze Zahlen (positiv oder negativ)         *
  *        sein müssen!  (Nach Diophantos 250 n.d.Z.)             *
9 '*****************************************************************
10 CLS:DEFINTA-P,R-Z:DIMQ(50):I=0:U=1:V=0
20 PRINT:PRINT"Es wird die 'Diophantische Gleichung'
30 PRINT"              Ax + By = C
40 PRINT"gelöst, worin A, B und C ganze Zahlen und A,B teilerfremd sind!
50 PRINT:PRINT"Gib 3 ganze Zahlen A, B, C ein !
Die 3 Zahlen dürfen positiv oder negativ sein;
52 PRINT"A und B  m ü s s e n  teilerfremd sein
und dürfen deshalb auch nicht gleich 1 sein!
55 PRINT:INPUT"A, B, C = ";A,B,C
60 IFB<0B=-B:F1=1
65 IFA<0A=-A:F3=1
70 IFB<A,CMD"F=SWAP",A,B:F2=1
80 Z=A:N=B
90 Q(I)=FIX(Z/N):R=Z-Q(I)*N:IFR=0THEN110
100 Z=N:N=R:I=I+1:GOTO90
110 FORJ=I-1TO1 STEP-1:V=Q(J)*U+V:CMD"F=SWAP",U,V:NEXT
120 FORK1=1TO2:FORK2=1TO2
130 IF((A*U*(-1)ÄK1)+(B*V*(-1)ÄK2))=1THEN140ELSENEXTK2,K1
140 U= (-1)ÄK1*U:V=(-1)ÄK2*V
150 PRINT"Gib eine beliebige ganze Zahl ein!
  (Sie darf auch negativ sein)"
:INPUT"Bitte ";P
160 X=C*U+B*P:Y=C*V-A*P
170 IFF2=1:CMD"F=SWAP",A,B:CMD"F=SWAP",X,Y
175 IFF1=1,Y=-Y:B=-B
177 IFF3=1,X=-X:A=-A
180 PRINT:PRINT"Die beiden Lösungen lauten für diese Zahl:
190 PRINT"       x = ";X;"     y = ";Y
200 PRINT:PRINT"In der Tat ist:     "A" x "X"   +   "B" x "Y" = "C" !":END
```

DOTPRINT MIT 24-NADELDRUCKER EPSON LQ 550

Oskar Drechsler

Nun will ich auch noch die Anpassung des Dotprint an den Epson LQ 550 unters VolK werfen. Er ist zwar abwärtsKompatibel und arbeitet auch mit nur 8 Nadeln, aber wenn man einen 24-Nadel-DrucKer hat, dann soll gerade das Paradepferd für TextausdrucKe die Q ualität des DrucKers demonstrieren Können.

Da ich das Maschinenprogramm WPML/EPS nicht verändert habe und mir nicht vorstellen Kann, daß da schon dranrumgebastelt worden ist, dürften bei der ModifiKation Keine Probleme entstehen.

Es ist lediglich im Basic-Programm DOTPRINT/EPS in der Zeile 100 die PoKerei wie im anschließend abgedruckten Programm abzuändern und die Zeilen 130 bis 190 einzufügen. Die Zeilen-Nummern müssen nicht mit den Nummern in anderen Programmen identisch sein, aber das Programm ist so Kurz, daß man leicht die entsprechenden Stellen findet. Für die, die es genau wissen wollen, habe ich noch die Teile des Assemblerlistings von WPML/EPS beigefügt, in die die Änderungen gepoKt werden.

Im Listing von WPMLEr/Zeu stammen die Zeilen 24 bis 74 mit Ausnahme der Zeile 63 aus WPML/EPS. Ein Ändern der Zeile 63 in WPML war nicht möglich, da die Routine in unveränderter Form für einen anderen Ansprung benötigt wird. So mußte ich die Routine no chmal Komplett ausführen.

Wer zu faul ist, das Listing abzutippen, Kann mir eine DisKette schicKen.

Nun noch eine Bitte:

Wer weiß, woran es liegt, daß das Programm manchmal völlig unmotiviert ein BlanK drucKt. Dabei ist es egal, welches Wort verwendet wird. Der BlanK steht immer an der gleichen Stelle. An meiner ModifiKation Kann es nicht liegen. Das ist auch schon bei der Originalversion passiert. Bei meinen Nachforschungen Kam ich nur so weit, daß ich weiß, daß es nur an WPML/EPS liegen Kann und nicht an WP/Eps.


```
1 ' Dotprint/Eps geändert auf 24-Nadeldrucker EPSON LQ 550
2 ' soll nur ein 8-Nadeldrucker verwendet werden, sind die
3 ' Zeilen 130-190 zu löschen
4 '
10 CLS
20 PRINT"INITIALIZINGTIME$OUTRANDOMRESUMEIFIFIFLPRINTUSINGUSROUTRANDOMRESUMEIFIFIFIFLPR
INTUSROUTRANDOMRESUMEIFIFIFIFLPRINT
ORKILLUSELSEKILLUSINGTORESTORELSETUSRKILLUSELSEKILLUSINGUSINELSEKILLUSRKILLUSRUSINGUSIN
ELSEKILLUSINGUSING
ORKILLUSELSEKILLUSINGUSIROUTR
30 PRINT"USINGUSING
STRING$LETLETCMDLETERLLETGOTOCLSERLGOTOLETCLSCMDTHENUSINGUSINGUSINGUSINGUSINGUSING

INKEY$OUTRANDOMRESUMEIFIFIFIFLPRINTUSROUTRANDOMRESUMEIFIFIFIFLPRINTUSROUTRANDOMUSINGRES
UMEIFIFLPRINTUSINGUSROUTRANDOMRESUMEIFUSINGUSINGRESUMELPRINTUSROUTR
40 PRINT".
INKEY$KILLDELETEDEFDEFDEFDEFDEFSTRFORUSRKILLDELETEDEFDEFDEFDEFDEFSTRFORUSRKILLDELETEDEF
DEFDEFDEFDEFSTRFORUSRKILLDELETEDEFDEFDEFDEFDEFSTRFORUSRKILLDELETEDEFDEFDEFDEFDEFSTRFORS
TRING$MASON";
100 POKE16561,&H84:POKE16562,&HB0:CLEAR500:DEFINTI-Y:K=0:DIMM(11)
110 M(0)=32717:M(1)=8970:M(2)=9054:M(3)=-12970:M(4)=17456:M(5)=1320:M(6)=289:M(7)=6144:
M(8)=8451:M(9)=0:M(10)=-25917:M(11)=10
120 A$="WPML/EPS"+CHR$(13)+STRING$(50," "):DEFUSR=VARPTR(M(0)):K=USR(VARPTR(A$)):IFKTHE
NCLS:PRINT"DISK ERROR!":END
130 CMD"load wpmler24/cmd"
140 POKE &HC67C,&H86:POKE &HC67D,&HB0' Call B086
150 POKE &HCAB0,&HA4:POKE &HCAB1,&HB0' Call B0A4
170 POKE &HCAE4,&H86:POKE &HCAE5,&HB0' Call B086
180 POKE &HCAE7,&HA4:POKE &HCAE8,&HB0' Call B0A4
190 POKE &HCB19,&HC3:POKE &HCB1A,&H86:POKE &HCB1B,&HB0' JP B086
200 RUN"wp/eps
```

```
                  00001 ; Ergänzung von WPML/CMD (Dotprint) für 24-Nadeldrucker
                  00002 ; WPMLEr24/Zeu  bzw./CMD
                  00003
B086              00004         ORG     0B086h
BA32              00005 DruZwSp EQU     0BA32h
CA8F              00006 Druck   EQU     0CA8Fh
BA34              00007 AdrZaeh EQU     0BA34h
BA3C              00008 AnzAdr  EQU     0BA3Ch
BA8E              00009 Flag    EQU     0BA8Eh
BAA2              00010 ErstAdr EQU     0BAA2h
BAAA              00011 leAdr   EQU     0BAAAh
                  00012
B086  211B00      00013 BitIm   LD      HL,1Bh          ;ESC
B089  2232BA      00014         LD      (DruZwSp),HL
B08C  CD8FCA      00015         CALL    Druck
B08F  212A00      00016         LD      HL,2Ah          ;*
B092  2232BA      00017         LD      (DruZwSp),HL
B095  CD8FCA      00018         CALL    Druck
B098  212700      00019         LD      HL,39D          ;39 = dreifache Dichte
B09B  2232BA      00020         LD      (DruZwSp),HL
B09E  CD8FCA      00021         CALL    Druck
B0A1  C32BCB      00022         JP      0CB2Bh          ;wieder in Original
                  00023
B0A4  F3          00024 TextAu  DI
B0A5  2AE2BA      00025         LD      HL,(0BAE2h)
B0A8  7D          00026         LD      A,L
B0A9  B4          00027         OR      H
B0AA  2832        00028         JR      Z,TextA2
B0AC  2A8EBA      00029         LD      HL,(Flag)       ;Inh. von Flag in HL
B0AF  7D          00030         LD      A,L             ;A=L
B0B0  B4          00031         OR      H               ;ist beides 0 ?
B0B1  282B        00032         JR      Z,TextA2        ;ja, springe
B0B3  210000      00033         LD      HL,0h           ;ansonsten HL=0
B0B6  228EBA      00034         LD      (Flag),HL       ;und in Flag
B0B9  2AAABA      00035         LD      HL,(leAdr)      ;letzte Adr. in HL
B0BC  ED5BA2BA    00036         LD      DE,(ErstAdr)    ;erste Adr. in DE
B0C0  AF          00037         XOR     A               ;Z-Flag auf 0
B0C1  ED52        00038         SBC     HL,DE           ;HL-DE, Ergebnis in HL
B0C3  223CBA      00039         LD      (AnzAdr),HL     ;Ergebn. = Anzahl Adr.
B0C6  7D          00040         LD      A,L             ;L in Akku
B0C7  B4          00041         OR      H               ;ist HL 0 ?
B0C8  2814        00042         JR      Z,TextA2        ;ja, springe
B0CA  2AAABA      00043         LD      HL,(leAdr)      ;letzte Adr. in HL
B0CD  5D          00044         LD      E,L             ;HL in DE
B0CE  54          00045         LD      D,H             ;laden
B0CF  1B          00046         DEC     DE              ;und - 1
B0D0  ED4B3CBA    00047         LD      BC,(AnzAdr)     ;Anzahl Adressen in BC
B0D4  1A          00048 TextA1  LD      A,(DE)          ;le Adr-1 in Akku
B0D5  B6          00049         OR      (HL)            ;OR mit letzte Adr
B0D6  77          00050         LD      (HL),A          ;Ergebn. in letzte Adr
B0D7  2B          00051         DEC     HL              ;letzte Adr. -1
B0D8  1B          00052         DEC     DE              ;(letzte Adr.-1) -1
B0D9  0B          00053         DEC     BC              ;Anzahl Adressen -1
B0DA  79          00054         LD      A,C             ;Anzahl Adressen
B0DB  B0          00055         OR      B               ;jetzt 0 ?
B0DC  20F6        00056         JR      NZ,TextA1       ;nein, Schleife nochmal
B0DE  2AA2BA      00057 TextA2  LD      HL,(ErstAdr)    ;Erste Adresse in HL
B0E1  2234BA      00058         LD      (AdrZaeh),HL    ;und in Adr.-Zähler
B0E4  2AAABA      00059         LD      HL,(leAdr)      ;letzte Adr. HL
B0E7  E5          00060         PUSH    HL              ;und retten
B0E8  2A34BA      00061 TextA3  LD      HL,(AdrZaeh)    ;Adr.-Zähler in HL
B0EB  5E          00062         LD      E,(HL)          ;1. Adresse in E
```

```
B0EC  CD00B1      00063  -->    CALL    Test7           ;und an Drucker
B0EF  2A34BA      00064         LD      HL,(AdrZaeh)    ;Adr.-Zähler in HL
B0F2  23          00065         INC     HL              ;Zähler +1
B0F3  2234BA      00066         LD      (AdrZaeh),HL    ;und wieder zurück
B0F6  C1          00067         POP     BC              ;le Adr. in BC (von HL)
B0F7  C5          00068         PUSH    BC              ;und wieder retten
B0F8  03          00069         INC     BC              ;le Adr. +1
B0F9  AF          00070         XOR     A               ;Z-Flag auf 0
B0FA  ED42        00071         SBC     HL,BC           ;momentane Adr - le Adr
B0FC  38EA        00072         JR      C,TextA3        ;Übertrag, Schleife nochm
B0FE  C1          00073         POP     BC
B0FF  C9          00074         RET
                  00075
                  00076 ;hier wird der Wert aus E (für 8-Nadel-Drucker)
                  00077 ;verdreifacht in D, B und C umgeschaufelt
B100  7B          00078 Test7   LD      A,E             ;8 Nadelwert in Akku
B101  CB2F        00079         SRA     A               ;rechts rotieren
B103  CB2F        00080         SRA     A               ;rechts rotieren
B105  E6E0        00081         AND     0E0h            ;11100000 Bit 0-4 löschen
                  00082
B107  CB73        00083 Test6   BIT     6,E             ;Bit 6 testen
B109  2802        00084         JR      Z,Test5         ;nicht gesetzt, springe
B10B  F61C        00085         OR      01Ch            ;sonst 00011100 setzen
                  00086
B10D  CB6B        00087 Test5   BIT     5,E             ;Bit 6 testen
B10F  2807        00088         JR      Z,End5          ;nicht gesetzt, springe
B111  F603        00089         OR      03h             ;sonst 00000011 setzen
B113  57          00090         LD      D,A             ;und in D speichern
B114  3E80        00091         LD      A,80h           ;dann 1000000
B116  1802        00092         JR      Test4
B118  57          00093 End5    LD      D,A             ;und in D speichern
B119  AF          00094         XOR     A               ;Akku löschen
                  00095
B11A  CB63        00096 Test4   BIT     4,E             ;Bit 4 testen
B11C  2802        00097         JR      Z,Test3         ;nicht gesetzt, springe
B11E  F670        00098         OR      070h            ;sonst 01110000 setzen
                  00099
B120  CB5B        00100 Test3   BIT     3,E
B122  2802        00101         JR      Z,Test2
B124  F60E        00102         OR      0Eh             ;00001110
                  00103
B126  CB53        00104 Test2   BIT     2,E
B128  2807        00105         JR      Z,End2
B12A  F601        00106         OR      01h             ;00000001
B12C  47          00107         LD      B,A
B12D  3EC0        00108         LD      A,0C0h          ;11000000
B12F  1802        00109         JR      Test1
B131  47          00110 End2    LD      B,A
B132  AF          00111         XOR     A
                  00112
B133  CB4B        00113 Test1   BIT     1,E
B135  2802        00114         JR      Z,Test0
B137  F638        00115         OR      38h             ;00111000
                  00116
B139  CB43        00117 Test0   BIT     0,E
B13B  2802        00118         JR      Z,End0
B13D  F607        00119         OR      07h             ;00000111
B13F  4F          00120 End0    LD      C,A
                  00121
                  00122 ; 3 mal 8 Nadeln ausgeben
B140  7A          00123         LD      A,D             ;obere 8 Nadeln in Akku
B141  3232BA      00124         LD      (DruZwSp),A
```

```
B144  CD8FCA    00125          CALL   Druck
B147  78        00126          LD     A,B               ;mittl. 8 Nadeln in Akku
B148  3232BA    00127          LD     (DruZwSp),A
B14B  CD8FCA    00128          CALL   Druck
B14E  79        00129          LD     A,C               ;unter. 8 Nadeln in Akku
B14F  3232BA    00130          LD     (DruZwSp),A
B152  CD8FCA    00131          CALL   Druck
B155  C9        00132          RET
                00133
0000            00134          END

00000 Fehler
```

```
CB04  2232BA    01823          LD     (DruZwSp),HL
CB07  CD8FCA    01824          CALL   Druck
CB0A  2A9EBA    01825 Ausgal   LD     HL,(Spei29)
CB0D  7D        01826          LD     A,L
CB0E  B4        01827          OR     H
CB0F  C0        01828          RET    NZ
CB10  210D00    01829          LD     HL,0Dh            ;CR
CB13  2232BA    01830          LD     (DruZwSp),HL
CB16  CD8FCA    01831          CALL   Druck
CB19  211B00    01832 BitIm    LD     HL,1Bh            ;Änderung: C386B0
CB1C  2232BA    01833          LD     (DruZwSp),HL
CB1F  CD8FCA    01834          CALL   Druck
CB22  214C00    01835          LD     HL,4Ch            ;L
CB25  2232BA    01836          LD     (DruZwSp),HL
CB28  CD8FCA    01837          CALL   Druck
CB2B  2AAEBA    01838          LD     HL,(N1)           ;N1
CB2E  2232BA    01839          LD     (DruZwSp),HL
CB31  CD8FCA    01840          CALL   Druck
CB34  2AACBA    01841          LD     HL,(N2)           ;N2 * 256
CB37  2232BA    01842          LD     (DruZwSp),HL
CB3A  CD8FCA    01843          CALL   Druck
CB3D  2AA6BA    01844          LD     HL,(Spei1)
CB40  7D        01845          LD     A,L
CB41  B4        01846          OR     H
CB42  C8        01847          RET    Z
CB43  210000    01848          LD     HL,0h
CB46  22B2BA    01849          LD     (0BAB2h),HL
CB49  2AA6BA    01850          LD     HL,(Spei1)
CB4C  22B6BA    01851          LD     (0BAB6h),HL
CB4F  CD64CB    01852          CALL   BitImU1
CB52  2A16BB    01853          LD     HL,(0BB16h)
CB55  7D        01854          LD     A,L
CB56  B4        01855          OR     H
CB57  C8        01856          RET    Z
CB58  2A18BB    01857          LD     HL,(0BB18h)
CB5B  ED5BA2BA  01858          LD     DE,(ErstAdr)
CB5F  19        01859          ADD    HL,DE
CB60  3EFF      01860          LD     A,0FFh
CB62  77        01861          LD     (HL),A
CB63  C9        01862          RET
                01863

00000 Fehler
```



```
                00001 ;WPML/Zeu            von BBA5 - D23F
                00002
                01284
C655  AF        01285          XOR    A
C656  ED52      01286          SBC    HL,DE
C658  FA46CC    01287          JP     M,0CC46h
C65B  210001    01288          LD     HL,100h
C65E  ED5B36BA  01289          LD     DE,(0BA36h)
C662  CD9024    01290          CALL   IntDiv
C665  CD7F0A    01291          CALL   CINT
C668  22ACBA    01292          LD     (N2),HL
C66B  21FF00    01293          LD     HL,0FFh
C66E  ED5B36BA  01294          LD     DE,(0BA36h)
C672  7B        01295          LD     A,E
C673  A5        01296          AND    L
C674  6F        01297          LD     L,A
C675  7A        01298          LD     A,D
C676  A4        01299          AND    H
C677  67        01300          LD     H,A
C678  22AEBA    01301          LD     (N1),HL
C67B  CD19CB    01302          CALL   BitIm             ;Änderung: CD86B0
C67E  2ABCBA    01303          LD     HL,(0BABCh)
C681  ED5B74BA  01304          LD     DE,(Spei10)
C685  AF        01305          XOR    A
C686  ED52      01306          SBC    HL,DE
C688  22EEBA    01307          LD     (0BAEEh),HL
C68B  210000    01308          LD     HL,0h
C68E  22F0BA    01309          LD     (0BAF0h),HL
                01310
                01790
CAA9  210100    01791 Ausgab   LD     HL,1h
CAAC  228EBA    01792          LD     (Flag),HL
CAAF  CD8DCB    01793          CALL   TextAu            ;Änderung: CDA4B0
CAB2  2AA2BA    01794          LD     HL,(ErstAdr)
CAB5  22A0BA    01795          LD     (leAdr0),HL
CAB8  2AD0BA    01796          LD     HL,(0BAD0h)
CABB  7D        01797          LD     A,L
CABC  B4        01798          OR     H
CABD  204B      01799          JR     NZ,Ausgal
CABF  211B00    01800          LD     HL,1Bh            ;ESC
CAC2  2232BA    01801          LD     (DruZwSp),HL
CAC5  CD8FCA    01802          CALL   Druck
CAC8  213300    01803          LD     HL,33h            ;3
CACB  2232BA    01804          LD     (DruZwSp),HL
CACE  CD8FCA    01805          CALL   Druck
CAD1  210100    01806          LD     HL,1h             ;1/180 Zeilenabstand
CAD4  2232BA    01807          LD     (DruZwSp),HL
CAD7  CD8FCA    01808          CALL   Druck
CADA  210D00    01809          LD     HL,0Dh            ;CR
CADD  2232BA    01810          LD     (DruZwSp),HL
CAE0  CD8FCA    01811          CALL   Druck
CAE3  CD19CB    01812          CALL   BitIm             ;Änderung: CD86B0
CAE6  CD8DCB    01813          CALL   TextAu            ;Änderung: CDA4B0
CAE9  2AA2BA    01814          LD     HL,(ErstAdr)
CAEC  22A0BA    01815          LD     (leAdr0),HL
CAEF  211B00    01816          LD     HL,1Bh            ;ESC
CAF2  2232BA    01817          LD     (DruZwSp),HL
CAF5  CD8FCA    01818          CALL   Druck
CAF8  213300    01819          LD     HL,33h            ;3
CAFB  2232BA    01820          LD     (DruZwSp),HL
CAFE  CD8FCA    01821          CALL   Druck
CB01  211700    01822          LD     HL,17h            ;23/180 Zeilenabstand
```

ßLprint mit 24-Nadel-Drucker EPSON LQ 550

Oskar Drechsler

Zum Weihnachtsfest habe ich mir einen Epson LQ 550 geschenkt und sogleich den Lprint-Befehl der HRG 1B auf 24 Nadeln umgeschrieben.

Ich nehme an, daß im Club noch mehrere 24-Nadeldrucker vorhanden sind. Deshalb will ich mal meine Version zum Besten geben.

Da die Ansteuerung der Nadeln nicht bei allen Druckern gleich sein dürfte, zuerst das System des Epson LQ 550: Es liegen 3 Nadelreihen zu je 8 Nadeln vertikal versetzt nebeneinander. Diese 3 Reihen stellen fiktiv eine Reihe von 24 Nadeln untereinander und somit eine Druckspalte dar. Wann bei welcher Druckkopfposition welche Reihe angesprochen wird regelt der Drucker und braucht den Programmierer nicht zu interessieren. Bis hierher und auch das nachfolgende Schema dürfte bei allen Druckern noch gleich sein.

```
X ----------------Nadel 24
  X --------------Nadel 23
    X ------------Nadel 22
X ----------------Nadel 21
  X               usw.
    X
X
  X
    X
X
  X
    X
X
  X
    X
X
  X
    X
X
  X
    X
X
  X
    X
```

Die Nadeln sind so angeordnet, daß sie überlappen und somit ein lückenloses Druckbild ergeben.

Ab hier kommen wohl die Unterschiede. Der LQ 550 hat den Befehl: ESC * m n1 n2 Daten Daten Daten ....

ESC * = 24-Nadel-Modus
m = Druckmodus z.B. 32 = einfache Dichte, 33 = dopp. Dichte
n1,n2 = LSB,MSB: = die Zahl der zu druckenden Spalten a 24 Nadeln. Der Drucker nimmt n1 * 1 und n2 * 256.
230 muß also umgesetzt werden in: 4,1
Daten = pro angegebener Spalte muß ein Byte für die untersten 8 Nadeln, 1 Byte für die mittleren und 1 Byte für die oberen 8 Nadeln übergeben werden.

Mir hat jemand erzählt, es gäbe Drucker, bei denen die Reihenfolge von oben nach unten geht oder jedes übergebene Byte eine komplette Nadelreihe ansteuert. Ich kann das nicht bestätigen und somit nur kommentarlos weitergeben.

Aus dem eben gesagten ist ersichtlich, daß der Rechner bei einfacher Dichte die 3-fache Datenmenge erzeugen und in den Drukker schaufeln muß als bei 8 Nadeln und somit der Ausdruck einer Grafik länger dauert. Bei doppelter Dichte, wie ich sie angewendet habe, ist es die 6-fache Menge, also rund 55 kByte. (Ob mir KaJot diese Zahl widerlegen kann, so wie Prof. Haber den Beginn des Jahrzehnts ?)
Bei dieser Datenmenge ist der Puffer meines Druckers hoffnungslos überfordert und veranlaßt mich nebenbei zu der Frage: hat jemand im Club schon den Mega-Puffer aus ELEKTOR nachgebaut? Wenn ja, wäre ich über einen kurzen Erfahrungsbericht dankbar. Wer noch nichts davon gehört hat: man kann den Puffer in Stufen von 32 K, 64K, 128K ... bis 1 MByte ausbauen.

Nachfolgend ist das Assembler-Listing der 8-Nadel Druck-Routine und die von mir erweiterte Version für 24 Nadeln abgedruckt. Soweit ich weiß, liegt die Druck-Routine bei allen Versionen am Ende von HRG/CMD. Ob sie immer an der gleichen Stelle beginnt, ist nicht wahrscheinlich, denn an diesem Programm haben schon viele rumgefummelt.

Assembler-Freaks brauche ich wohl nicht zu erklären, wie man die Änderung einfügt.
Für die, die von Assembler keine Ahnung haben, ist eine Erläuterung wenig sinnvoll. Wer mir eine Diskette mit seinem HRG/CMD schickt, dem füge ich gerne die Änderung hinzu. (Rückporto nicht vergessen!)
Ich habe 2 Laufwerke: 0 = 40 Tr SS DD, 1 = 80 Tr DD DD.

```
                 ; #Lprint-Befehl von HRG/CMD für 8-Nadeldrucker
                 ; In der Originalversion stand noch 2 x 'ESC §'. Ich habe
                 ; das herausgenommen, um Bilder nahtlos aneinander drucken
                 ; können.
FE4B                         ORG     0FE4Bh

0002             XWert       DS      2
0002             YWert       DS      2
FE4F  00         NadNum      NOP
FE50  00         NadWer      NOP

FE51  F5         DruTes      PUSH    AF
FE52  DBFD                   IN      A,(0FDh)
FE54  FEFF                   CP      0FFh
FE56  200A                   JR      NZ,DruIn
FE58  F1                     POP     AF
FE59  21E837                 LD      HL,37E8h
FE5C  CB7E       DruOn       BIT     7,(HL)
FE5E  20FC                   JR      NZ,DruOn
FE60  77                     LD      (HL),A
FE61  C9                     RET
FE62  DBFD       DruIn       IN      A,(0FDh)
FE64  CB7F                   BIT     7,A
FE66  20FA                   JR      NZ,DruIn
FE68  F1                     POP     AF
FE69  D3FD                   OUT     (0FDh),A
FE6B  C9                     RET

FE6C  0601       LprLink     LD      B,01h           ;B = 1
FE6E  FE00       LprLin1     CP      00h             ;ist Akku 0 ?
FE70  C8                     RET     Z               ;ja, zurück
FE71  3D                     DEC     A               ;Akku -1
FE72  CB20                   SLA     B               ;Arithmet. linksschieben
FE74  18F8                   JR      LprLin1         ;weiter

FE76  CDD5F5     Lprint      CALL    Vorber          ;Einsprungadresse
FE79  E5                     PUSH    HL
                 ;           hier stand mal ESC §
FE7A  3E1B                   LD      A,1Bh           ;ESC
FE7C  CD51FE                 CALL    DruTes
FE7F  3E41                   LD      A,'A'           ;'A'
FE81  CD51FE                 CALL    DruTes
FE84  3E08                   LD      A,08h           ;8/60 Zoll Zeilenabstand
FE86  CD51FE                 CALL    DruTes
FE89  210000                 LD      HL,0h
                 ; ESC A ist bei manchen Druckern n/72 Zoll Zeilenabstand.
                 ; Wenn man dann 8 stehenläßt, wird ein Kreis zum Ei.
                 ; (Hallo KaJot!)

                 ; Hier beginnt die Schleife für Zeilen je 8 Bit hoch
FE8C  224DFE     LpZeil      LD      (YWert),HL      ;YWert auf 0
FE8F  3E1B                   LD      A,1Bh           ;ESC
FE91  CD51FE                 CALL    DruTes
FE94  3E4B                   LD      A,'K'           ;K
FE96  CD51FE                 CALL    DruTes
FE99  3E80                   LD      A,80h           ;=128d
FE9B  CD51FE                 CALL    DruTes
FE9E  3E01                   LD      A,01h           ;1x256d
FEA0  CD51FE                 CALL    DruTes          ;=384 Spalten
FEA3  210000                 LD      HL,0h
                 PAGE
```



```
                 ; Hier beginnt die Schleife für 384 Spalten
FEA6  224BFE     LpSpa       LD      (XWert),HL      ;XWert auf 0
FEA9  AF                     XOR     A
FEAA  3250FE                 LD      (NadWer),A      ;Nadelwert auf 0
FEAD  324FFE     LpSpa1      LD      (NadNum),A      ;Nadelnummer auf 0
FEB0  2A4DFE                 LD      HL,(YWert)      ;YWert in HL
FEB3  1600                   LD      D,00h
FEB5  3A4FFE                 LD      A,(NadNum)      ;Nadelnummer in Akku
FEB8  5F                     LD      E,A             ;und in E
FEB9  19                     ADD     HL,DE           ;YWert + Nadelnummer
FEBA  E5                     PUSH    HL              ;und retten
FEBB  2A4BFE                 LD      HL,(XWert)      ;XWert in HL
FEBE  D1                     POP     DE              ;HL in DE
FEBF  CD80EE                 CALL    XYBere          ;X, Y am Bildsch. berech.
FEC2  A0                     AND     B               ;Punkt gesetzt ?
FEC3  CAD7FE                 JP      Z,LpSpa2        ;ja, springe
FEC6  3A4FFE                 LD      A,(NadNum)      ;ans. Nadel-Nr. in Akku
FEC9  47                     LD      B,A             ;und in B
FECA  3E07                   LD      A,07h           ;Akku = 7
FECC  90                     SUB     B               ;Akku - B
FECD  CD6CFE                 CALL    LprLink         ;linksschieben, nach oben
FED0  3A50FE                 LD      A,(NadWer)      ;Nadelwert in Akku
FED3  80                     ADD     A,B             ;Akku + B
FED4  3250FE                 LD      (NadWer),A      ;Nadelwert wieder zurück
FED7  3A4FFE     LpSpa2      LD      A,(NadNum)      ;Nadelnummer in Akku
FEDA  3C                     INC     A               ;+1
FEDB  FE08                   CP      08h             ;sind es schon 8 Nadeln ?
FEDD  C2ADFE                 JP      NZ,LpSpa1       ;nein, springe
FEE0  3A50FE                 LD      A,(NadWer)      ;ans. Nadelwert in Akku
FEE3  CD51FE                 CALL    DruTes          ;und ausdrucken
FEE6  2A4BFE                 LD      HL,(XWert)      ;XWert in Akku
FEE9  23                     INC     HL              ;+1
FEEA  7C                     LD      A,H             ;H in Akku
FEEB  FE01                   CP      01h             ;ist es 1 ?
FEED  C2A6FE                 JP      NZ,LpSpa        ;nein, springe
FEF0  7D                     LD      A,L             ;ans. L in Akku
FEF1  FE80                   CP      80h             ;ist es 128d ?
FEF3  C2A6FE     LpSpaE      JP      NZ,LpSpa        ;nein, springe

FEF6  3E0D                   LD      A,0Dh           ;Akku mit 'CR' laden
FEF8  CD51FE                 CALL    DruTes          ;neue Zeile drucken
FEFB  2A4DFE                 LD      HL,(YWert)      ;YWert in Akku
FEFE  110800                 LD      DE,8h           ;DE mit 8 laden
FF01  19                     ADD     HL,DE           ;YWert + 8
FF02  7D                     LD      A,L             ;L in Akku
FF03  FEC0                   CP      0C0h            ;ist es 192 ?
FF05  C28CFE                 JP      NZ,LpZeil       ;nein springe

                 ;           hier stand mal ESC §
FF08  E1                     POP     HL
FF09  C31E1D                 JP      1D1Eh           ;Programmausführung

EE80             XYBere      EQU     0EE80h          ;berechnet X und Y-Wert
F5D5             Vorber      EQU     0F5D5h          ;Argumente aus Basic
                                                     ;soweit vorhanden
0000                         END

00000 Fehler
```

```
                00001 ; #Lprint-Befehl HRG/CMD für 24 Nadel-Drucker
                00002 ; EPSON LQ 550 und Kompatible
                00003
FE4B            00004          ORG     0FE4Bh
                00005
0002            00006 XWert    DS      2h
0002            00007 YWert    DS      2h
FE4F  00        00008 NadNum   NOP
FE50  00        00009 NadWer   NOP
FE51  F5        00010 Druck    PUSH    AF              ;Zeichen retten
FE52  DBFD      00011 TestPor  IN      A,(0FDh)        ;Port FD testen
FE54  FEFF      00012          CP      0FFh            ;Drucker ausgeschaltet?
FE56  200A      00013          JR      NZ,DruIn        ;nein, springe
FE58  F1        00014          POP     AF              ;ans. Zeichen zurück
FE59  21E837    00015          LD      HL,37E8h        ;Adresse in HL
FE5C  CB7E      00016 DruOff   BIT     7,(HL)          ;Drucker Off-Line ?
FE5E  20FC      00017          JR      NZ,DruOff       ;ja, nochmal Abfrage
FE60  77        00018          LD      (HL),A          ;ausgeben über Adresse
FE61  C9        00019          RET
FE62  DBFD      00020 DruIn    IN      A,(0FDh)        ;Port FD testen
FE64  CB7F      00021          BIT     7,A             ;Drucker Offline ?
FE66  20FA      00022          JR      NZ,DruIn        ;ja, nochmal Abfrage
FE68  F1        00023          POP     AF              ;Zeichen zurück
FE69  D3FD      00024          OUT     (0FDh),A        ;ausgeben über Port
FE6B  C9        00025          RET
                00026
FE6C  0601      00027 LprLink  LD      B,01h           ;B = 1
FE6E  FE00      00028 LprLin1  CP      00h             ;ist Akku 0 ?
FE70  C8        00029          RET     Z               ;ja, zurück
FE71  3D        00030          DEC     A               ;Akku -1
FE72  CB20      00031          SLA     B               ;Arithmet. linksschieben
FE74  18F8      00032          JR      LprLin1         ;weiter
                00033
FE76  CDD5F5    00034 Lprint   CALL    Vorber          ;Einsprungadresse
FE79  E5        00035          PUSH    HL
FE7A  3E1B      00036          LD      A,1Bh           ;ESC
FE7C  CD51FE    00037          CALL    Druck
FE7F  3E41      00038          LD      A,'A'           ;'A'
FE81  CD51FE    00039          CALL    Druck
FE84  3E08      00040          LD      A,08h           ;8/60 Zoll Zeilenabstand
FE86  CD51FE    00041          CALL    Druck
FE89  210000    00042          LD      HL,0h
                00043
                00044 ; Hier beginnt die Schleife für Zeilen je 8 Bit hoch
FE8C  224DFE    00045 LpZeil   LD      (YWert),HL      ;YWert auf 0
FE8F  3E1B      00046          LD      A,1Bh           ;ESC
FE91  CD51FE    00047          CALL    Druck
FE94  3E2A      00048          LD      A,'*'           ;* 24 Nadeln <-------
FE96  CD51FE    00049          CALL    Druck
FE99  3E21      00050          LD      A,33d           ;Mode 33 doppelte Dichte
FE9B  CD51FE    00051          CALL    Druck
FE9E  3E00      00052          LD      A,00h           ;=0  (Nadelanzahl *1)
FEA0  CD51FE    00053          CALL    Druck
FEA3  3E03      00054          LD      A,03h           ;3x256=768 Sp. (384*2)
FEA5  CD51FE    00055          CALL    Druck
FEA8  210000    00056          LD      HL,0h
                00057
                00058 ; Hier beginnt die Schleife für 384 Spalten
FEAB  224BFE    00059 LpSpa    LD      (XWert),HL      ;XWert auf 0
FEAE  AF        00060          XOR     A
FEAF  3250FE    00061          LD      (NadWer),A      ;Nadelwert auf 0
FEB2  324FFE    00062 LpSpa1   LD      (NadNum),A      ;Nadelnummer auf 0
```



```
FEB5  2A4DFE    00063          LD      HL,(YWert)      ;YWert in HL
FEB8  1600      00064          LD      D,00h
FEBA  3A4FFE    00065          LD      A,(NadNum)      ;Nadelnummer in Akku
FEBD  5F        00066          LD      E,A             ;und in E
FEBE  19        00067          ADD     HL,DE           ;YWert + Nadelnummer
FEBF  E5        00068          PUSH    HL              ;und retten
FEC0  2A4BFE    00069          LD      HL,(XWert)      ;XWert in HL
FEC3  D1        00070          POP     DE              ;HL in DE
FEC4  CD80EE    00071          CALL    XYBere          ;X, Y am Bildsch. berech.
FEC7  A0        00072          AND     B               ;Punkt gesetzt ?
FEC8  2811      00073          JR      Z,LpSpa2        ;ja, springe
FECA  3A4FFE    00074          LD      A,(NadNum)      ;ans. Nadel-Nr. in Akku
FECD  47        00075          LD      B,A             ;und in B
FECE  3E07      00076          LD      A,07h           ;Akku = 7
FED0  90        00077          SUB     B               ;Akku - B
FED1  CD6CFE    00078          CALL    LprLink         ;linksschieben, nach oben
FED4  3A50FE    00079          LD      A,(NadWer)      ;Nadelwert in Akku
FED7  80        00080          ADD     A,B             ;Akku + B
FED8  3250FE    00081          LD      (NadWer),A      ;Nadelwert wieder zurück
FEDB  3A4FFE    00082 LpSpa2   LD      A,(NadNum)      ;Nadelnummer in Akku
FEDE  3C        00083          INC     A               ;+1
FEDF  FE08      00084          CP      08h             ;ist es schon 8 ?
FEE1  C2B2FE    00085          JP      NZ,LpSpa1       ;nein, springe
FEE4  CD0DFF    00086          CALL    Test7           ;und ausdrucken <-----
FEE7  2A4BFE    00087          LD      HL,(XWert)      ;XWert in Akku
FEEA  23        00088          INC     HL              ;+1
FEEB  7C        00089          LD      A,H             ;H in Akku
FEEC  FE01      00090          CP      01h             ;ist es 1 ?
FEEE  C2ABFE    00091          JP      NZ,LpSpa        ;nein, springe
FEF1  7D        00092          LD      A,L             ;ans. L in Akku
FEF2  FE80      00093          CP      80h             ;ist es 128d ?
FEF4  C2ABFE    00094 LpSpaE   JP      NZ,LpSpa        ;nein, springe
FEF7  3E0D      00095          LD      A,0Dh           ;Akku mit 'CR' laden
FEF9  CD51FE    00096          CALL    Druck           ;neue Zeile
FEFC  2A4DFE    00097          LD      HL,(YWert)      ;YWert in Akku
FEFF  110800    00098          LD      DE,8h           ;DE mit 8 laden
FF02  19        00099          ADD     HL,DE           ;YWert + 8
FF03  7D        00100          LD      A,L             ;L in Akku
FF04  FEC0      00101          CP      0C0h            ;ist es 192 ?
FF06  C28CFE    00102          JP      NZ,LpZeil       ;nein springe
FF09  E1        00103          POP     HL
FF0A  C31E1D    00104          JP      1D1Eh           ;Programmausführung
                00105
                00106 ; Ergänzung des #Lprint-Befehls für 24-Nadeldrucker
                00107 ; Bit's von NadWer testen und verdreifachen
FF0D  2150FE    00108 Test7    LD      HL,NadWer       ;Wert für 8 Nadeln
FF10  7E        00109          LD      A,(HL)          ;in Akku
FF11  CB2F      00110          SRA     A               ;rechts schieben
FF13  CB2F      00111          SRA     A               ;und nochmal
FF15  E6E0      00112          AND     0E0h            ;11100000 Rest auf 0
                00113
FF17  CB76      00114 Test6    BIT     6,(HL)
FF19  2802      00115          JR      Z,Test5
FF1B  F61C      00116          OR      01Ch            ;00011100
                00117
FF1D  CB6E      00118 Test5    BIT     5,(HL)
FF1F  2807      00119          JR      Z,End5
FF21  F603      00120          OR      03h             ;00000011
FF23  57        00121          LD      D,A
FF24  3E80      00122          LD      A,80h           ;10000000
FF26  1802      00123          JR      Test4
FF28  57        00124 End5     LD      D,A
```

HEFT
[illegible]
April
1990

```
FF29  AF                      XOR    A
FF2A  CB66      Test4         BIT    4,(HL)
FF2C  2802                    JR     Z,Test3
FF2E  F670                    OR     070h            ;01110000
FF30  CB5E      Test3         BIT    3,(HL)
FF32  2802                    JR     Z,Test2
FF34  F60E                    OR     0Eh             ;00001110
FF36  CB56      Test2         BIT    2,(HL)
FF38  2807                    JR     Z,End2
FF3A  F601                    OR     01h             ;00000001
FF3C  5F                      LD     E,A
FF3D  3EC0                    LD     A,0C0h          ;11000000
FF3F  1802                    JR     Test1
FF41  5F        End2          LD     E,A
FF42  AF                      XOR    A
FF43  CB4E      Test1         BIT    1,(HL)
FF45  2802                    JR     Z,Test0
FF47  F638                    OR     38h             ;00111000
FF49  CB46      Test0         BIT    0,(HL)
FF4B  2802                    JR     Z,End0
FF4D  F607                    OR     07h             ;00000111
FF4F  4F        End0          LD     C,A
                ; Nadelreihen 2 x ausgeben um bei doppelter Dichte
                ; die gleiche Breite zu erreichen
FF50  0602      Schleif       LD     B,02            ;B=2, Schleife 2 x
FF52  7A        NadOb         LD     A,D             ;obere 8 Nadeln aus D
FF53  CD51FE                  CALL   Druck
FF56  7B        NadMit        LD     A,E             ;mittlere 8 Nadeln aus E
FF57  CD51FE                  CALL   Druck
FF5A  79        NadUnt        LD     A,C             ;untere 8 Nadeln aus C
FF5B  CD51FE                  CALL   Druck
FF5E  10F2                    DJNZ   NadOb
FF60  C9                      RET
EE80            XYBere        EQU    0EE80h          ;berechnet X und Y-Wert
F5D5            Vorber        EQU    0F5D5h          ;Argumente aus Basic
                                                     ;soweit vorhanden
0000                          END

00000 Fehler
```


Dezi mal anders

Vor längerer Zeit stellte mal jemand eine Routine aus TSCRIPS vor, die Binärzahlen dezimal umwandelt. Zwar kann ich den Artikel in Kajots Gesamtinhaltsverzeichnis nicht finden, aber ich meine, daß ich eine verkürzte, beschleunigte Version im nächsten Info veröffentlichte. Aus gegebenem Anlaß habe ich mir die Source noch einmal vorgeknöpft und weiter gestreamlined.

Nach wie vor rechnet sie ganz banal einen binären 16-Bit-Wert in dezimale ASCII-Ziffern um, die in einem Puffer abgelegt werden. Jetzt gibt es aber drei verschiedene Entries in das Programm, die das Problem der führenden Nullen unterschiedlich angehen:

Beim Einsprung am Label ldzero werden führende Nullen als solche ausgegeben. Das mag bei irgendwelchen Anwendungen seinen Sinn haben. Interessanter ist die Möglichkeit, führende Nullen durch Leerzeichen zu ersetzen (Entry am Label ldblank). Dabei ergibt sich optisch ein Dezimaltabulator. Diese beiden Varianten ergeben immer einen fünfstelligen String, gegf. aus führenden Blanks und den Ziffern. Aber schließlich kann auch (bei Einsprung am Label nolead) der Ziffernstring auf die tatsächlich erforderlichen Stellen begrenzt werden. Das ist dann interessant, wenn die Dezimalzahl etwa nicht in einer Tabelle, sondern in einem Text erscheinen soll.

Das Programm setzt ans Ende des Strings als EOS das Byte 03h. So kann die Dezimalzahl beispielsweise über die DOS-Routine an 4467h (TEXTTV) oder 446Ah (TEXTLP) auf dem Bildschirm oder dem Drucker ausgegeben werden. Zur Vereinfachung steht HL bereits als Zeiger auf dem Pufferanfang. Zusätzlich zeigt DE auf das EOS-Byte. So kann es z. B. leicht für einen Text mit einem Blank überschrieben werden.

Hieraus resultiert ein kleiner Zusatzaufwand im Programm: Um die DOS-Konvention der Übergabeparameter einzuhalten, mußte DE den anzuzeigenden Wert und HL den Zeiger enthalten. So wird das Programm auch verlassen.

Arnulf

```
                ;       DE als Dezimalstring nach (HL)
                ;
                ; Eingang:      HL: Pufferadresse für den ASCII-String
                ;               DE: Integerwert 0000h-FFFFh
                ; Ausgang:      HL: unverändert
                ;               DE: letzte Pufferstelle, (DE)=EOS
                ; verändert:    DE
5200                    ORG     5200h
                ;Einsprung 1: Führende Nullen sollen ausgegeben werden (immer 5stellig)
5200  F5        ldzero  PUSH    AF              ;retten
5201  3E30              LD      A,'0'           ;führende 0
5203  1807              JR      decasci         ;dort weiter
                ;Einsprung 2: Führ. Nullen sollen durch Blank ersetzt werden (wie Tabul.)
5205  F5        ldblank PUSH    AF              ;retten
5206  3E20              LD      A,' '           ;führendes Blank
5208  1802              JR      decasci         ;dort weiter
                ;Einsprung 3: Führende Nullen sollen keine Stelle beanspruchen
520A  F5        nolead  PUSH    AF              ;retten
520B  AF                XOR     A               ;A <- Flag 0: kein führendes Zeichen
520C  FDE5      decasci PUSH    IY              ;alles retten
520E  E5                PUSH    HL
```

```
520F C5                PUSH  BC
5210 FD215952          LD    IY,ten^x      ;Tabelle der Zehnerpotenzen
5214 47                LD    B,A           ;B <- 00 bzw. Zeichen für führende Null
5215 0E00              LD    C,00h         ;C <- 00, Flag: bisher führende Nullen
5217 EB                EX    DE,HL         ;DE <- Pufferzeiger, HL <- Binärzahl
5218 1B                DEC   DE            ;weil gleich INC DE folgt
5219 13       getdiv   INC   DE            ;nächste Pufferstelle
521A D5                PUSH  DE            ;Pufferstelle retten
521B 3E30              LD    A,'0'         ;A <- Ziffer 0, Ausgangswert für Ziffer
521D FD5E00            LD    E,(IY+00h)    ;DE <- Zehnerpotenz
5220 FD5601            LD    D,(IY+01h)
5223 B7       testdiv  OR    A             ;Cy löschen
5224 ED52              SBC   HL,DE         ;Zahl minus Zehnerpotenz
5226 3803              JR    C,cmpquot     ;falls Zahl kleiner als Zehnerpotenz
5228 3C                INC   A             ;nächste Ziffer (Diff. > 0)
5229 18F8              JR    testdiv       ;Ziffer entspr. weiter erhöhen
522B 19       cmpquot  ADD   HL,DE         ;letzte Zehnerpotenz addieren (Pot.-Rest)
522C FD23              INC   IY            ;nächste Zehnerpotenz
522E FD23              INC   IY
5230 FE30              CP    '0'           ;ist die Ziffer =0?
5232 200E              JR    NZ,ciphok     ;falls höher
5234 0C                INC   C             ;ist C noch =0? (noch keine Ziffer >0,
5235 0D                DEC   C             ;noch immer führende Nullen)
5236 200B              JR    NZ,chrok      ;falls schon höher .
5238 78                LD    A,B           ;sonst führende 0 entspr. behandeln
5239 B7                OR    A             ;0 als Flag für Stellenzahl nach Bedarf?
523A 2007              JR    NZ,chrok      ;falls ja
523C 1D                DEC   E             ;schon Einerstelle erreicht? (E = 10^1)
523D 2805              JR    Z,samedig     ;falls ja
523F E3                EX    (SP),HL       ;HLalt retten, HLneu <- Pufferstelle
5240 2B                DEC   HL            ;zurückstellen
5241 E3                EX    (SP),HL       ;HL und Stack restaurieren
5242 4F       ciphok   LD    C,A           ;Ziffer erstmals >0: C entjungfern
5243 1D       chrok    DEC   E             ;schon Einerstelle erreicht? (E = 10^1)
5244 D1       samedig  POP   DE            ;Pufferstelle
5245 12                LD    (DE),A        ;Ziffer (oder Blank bei führ. 0) ablegen
5246 20D1              JR    NZ,getdiv     ;falls noch nicht Einerstelle erreicht
5248 EB                EX    DE,HL         ;HL <- Pufferzeiger
5249 0C                INC   C             ;Überhaupt keine Ziffer >0?
524A 0D                DEC   C             ;(nur führende Nullen: Zahl = 0000h)
524B 2002              JR    NZ,donepd     ;falls mindestens eine Ziffer >0
524D 3630              LD    (HL),'0'      ;wenigstens eine 0 einschreiben
524F 23       donepd   INC   HL            ;Pufferstelle nach der Einerstelle
5250 3603              LD    (HL),03h      ;EOS hinter die ASCII-Zahl
5252 EB                EX    DE,HL         ;DE <- Pufferende
5253 C1                POP   BC            ;alles restaurieren
5254 E1                POP   HL
5255 FDE1              POP   IY
5257 F1                POP   AF
5258 C9                RET                 ;zurück
                  ;Tabelle der Zehnerpotenzen für max. 5stellige Dezimalzahlen
5259 1027     ten^x    DW    10000
525B E803              DW    1000
525D 6400              DW    100
525F 0A00              DW    10
5261 0100              DW    1
0000                   END                 ;Entry bei ldzero, ldblank oder nolead

00000 Fehler

chrok   5243   ciphok 5242   cmpquot 522B   decasci 520C   donepd 524F   getdiv  5219
ldblank 5205   ldzero 5200   nolead  520A   samedig 5244   ten^x  5259   testdiv 5223
```


Programme ohne Grenzen

Als ich mit Klaus Thielecke, einem unserer beiden Mitstreiter in der DDR, Programme austauschen wollte, ging natürlich wieder die Sucherei nach dem richtigen Format los. Erstaunlicherweise ist das Format der Robotron Z1013-Rechner, die dort sehr weit verbreitet sind, dem des Genie IIs äußerst ähnlich und auch das Model-IV scheint seine Finger im Spiel gehabt zu haben. Hat da wer über den Zaun geschielt ? Die Reihenfolge der Sektoren ist etwas unterschiedlich, aber ansonsten haben sie vieles gemeinsam. Wer sonst noch Kontakt mit "drüben" hat und dem ein Utility zum Einstellen der Diskparameter zur Verfügung steht, wird hoffentlich mit den folgenden Daten zum Erfolg kommen:

```
Robotron Z1013 (80T, DS, DD, 780K)

Logische Sektoren je Spur (Vorder- und Rückseite)      (40)?
Tracks je Diskette                                     (160)?
Bytes je Block                                        (2048)?
Einträge im Directory                                  (128)?
System-Tracks                                            (4)?
Bytes je physikalischer Sektor (128-1024)             (1024)?
Doppelseitig                                             (J)?
 Ungerade Tracks auf der Rückseite                       (J)?
Doppelte Schreibdichte                                   (J)?
Doppel-Step (40er-Disks in 80er-Laufwerk)                (N)?
Inverse Daten                                            (N)?

Die Sektor-übersetungs-Tabelle enthält 5 Einträge.
Geben Sie die erste Sektor-Nr. ein                       (1)?
Skew-Faktor                                              (1)?
XLT-Tabelle: 1, 2, 3, 4, 5
```

Eine doppelseitige Diskette mit 80 Spuren wird dort wie eine einseitige mit 160 behandelt. Ich weiß nicht, ob das wirklich so ist, aber bei dem Programm von Gerald Schröder muß ich es so einstellen. Beim Model-IV-Format habe ich übrigens dasselbe Phänomen. Die Tabellen aus DISK.FDF sehen bei mir so aus:

```
*Video Genie 3s (80T, DS, DD, 768K)
80,5,31,3,191,255,192,0,64,3,5,3,80,192
0,1,2,3,4

*Video Genie IIs (80T, DS, DD, 780K)
80,5,31,3,194,255,192,0,64,2,5,3,80,192
0,1,2,3,4

*Montezuma Model IV (80T, DS, DD, 710K)
36,4,15,0,354,127,192,0,32,2,9,1,160,196
1,3,5,7,9,2,4,6,8

*Robotron Z1013 (80T, DS, DD, 780K)
40,4,15,0,389,127,192,0,32,4,5,3,160,204
1,2,3,4,5
```

Alexander Schmid

dBASE II und die Hardwareuhr

Da hat man sich endlich eine Hardwareuhr gebaut, diese sogar noch (kompatibel zum CP/M Plus) ins CP/M 2.2 eingebunden und dann muß man bei dBASE trotzdem jedesmal das Datum per Hand eingeben. Das muß doch wirklich nicht sein, oder?

Folgendes Programmchen macht das ab jetzt automatisch:

```
*************************************************************
* AUTODAT.CMD                                               *
* Setzt das Datum automatisch durch Einlesen der            *
* Systemzeit aus der Hardware-Uhr.                          *
* Der BDOS-Call ist im CP/M 3.0 standardmäßig vorhanden,    *
* unter CP/M 2.2 nur mit Z80DOS, P2DOS o.ä.                 *
*************************************************************

SET TALK OFF
SET INTENSITY OFF

POKE 42000,17,25,164,14,105,205,5,0,201
SET CALL TO 42000
CALL
STORE PEEK(42009) + 256*PEEK(42010) TO JULIAN
STORE JULIAN + 722099 TO JULIAN
STORE INT(JULIAN/365.26)+1 TO JAHR
STORE JULIAN + INT(395.25 - 365.25*JAHR) TO TAG

IF INT(JAHR/4)*4 = JAHR
  STORE 1 TO SCHATAG
ELSE
  STORE 2 TO SCHATAG
ENDIF

IF TAG>(91-SCHATAG)
  STORE TAG+SCHATAG TO TAG
ENDIF

STORE INT(TAG/30.57) TO MONAT
STORE TAG-INT(30.57*MONAT) TO TAG
IF MONAT>12
  STORE 1 TO MONAT
  STORE JAHR+1 TO JAHR
ENDIF
STORE JAHR-1900 TO JAHR

STORE STR(TAG,2) TO STAG
DO WHILE $(STAG,1,1)=" "
  STORE "0"+$(STAG,2,1) TO STAG
ENDDO

STORE STR(MONAT,2) TO SMONAT
DO WHILE $(SMONAT,1,1)=" "
  STORE "0"+$(SMONAT,2,1) TO SMONAT
ENDDO
```



```
STORE STR(JAHR,2) TO SJAHR
DO WHILE $(SJAHR,1,1)=" "
  STORE "0"+$(SJAHR,2,1) TO SJAHR
ENDDO

STORE STAG+"."+SMONAT+"."+SJAHR TO DATUM
?
?
?
?
?
§ 20,10 SAY "Das Datum vom "+DATUM+" wurde ins System übernommen"

SET DATE TO &DATUM

RELEASE TAG,MONAT,JAHR,JULIAN,SCHATAG,DATUM,STAG,SMONAT,SJAHR
```

Wer jetzt einen neuen Stern am dBASE-Himmel aufgehen sieht, der ihm alle Fragen beantworten kann, den muß ich leider enttäuschen, ich habe das Programm nämlich in einer alten Composter Persönlich gefunden. Die Mitgliederverwaltung des Clubs ist zwar in dBASE geschrieben, aber ich bin nur ein reiner Anwender, den lediglich die nervige Tipperei gestört hat. Mit Datenbanken hatte ich bisher noch nichts zu tun.

Trotzdem ein paar kurze Erläuterungen. POKE schreibt ein kurzes Maschinenprogramm ins freie RAM, das dann per BDOS-Call 105 die Uhr ausliest. Der Rest ist normale Datumsarithmetik. Das eine JULIAN + 722099 ist notwendig, weil die Rechnerei für ein Julianisches Datum mit Startwert 1.1.1600 gilt, das BDOS aber ab dem 1.1.1900 rechnet. So kann der Rechner 300 Jahre länger arbeiten, ohne ein falsches Datum zu haben. (ist das schon die April-Ausgabe ?)

Alexander Schmid

Das Turbo-Monster

Turbo-Pascal hat eine sehr unangenehme Eigenschaft, es macht sich im Speicher unheimlich breit. Wenn man mit Turbo ein *.COM-File erzeugt, merkt sich der Compiler, wieviel RAM ihm zur Verfügung steht und das COM-File will später mindestens ganausoviel Platz. Das kann jeder leicht selber ausprobieren, indem er zuerst ein Programm schreibt, das z.B. einfach nur eine Zeile Text am Bildschirm ausgibt. Wenn Ihr dann ein residentes Utility, z.B. MicroShell ladet, läßt sich das Turbo-Compilat nicht mehr starten, obwohl in der TPA noch massig Platz ist. Wenn MicroShell aber VOR dem Compilieren geladen wird, läuft das Turbo-File nacher auch, wenn MicroShell noch im Speicher ist (natürlich erst recht, wenn nicht).

Fazit: Man sollte im (O)ptionen-Menü die obere Grenze für den Speicher so weit wie möglich herabsetzen, dann läuft das Programm auch auf Maschinen mit kleinerer TPA oder wenn man noch ein paar residente Utilities im RAM hat.

Alexander Schmid

Platz sparen mit WordStar

Ich weiß nicht, wie Ihr Euren Briefverkehr organisiert, ich habe für einen neuen Brief immer einen alten mit der richtigen Anschrift im Briefkopf geladen und dann nur den Text neu geschrieben. Je nachdem, wie umfangreich der Kopf ist, bläht der den Brief ziemlich stark und absolut unnötigerweise auf. Am einfachsten wäre es, wenn man im jeweiligen Brief nur ganz kurz den Namen und eventuell das Datum hat und den langen Briefkopf erst beim Ausdruck und nicht auf der Floppy dazulädt. Beim WordStar 4.0 (beim 3er auch ???) kann man die Mailmerge-Funktion folgendermaßen mißbrauchen:

1. schreibt man seinen schönsten Sonntagsbriefkopf und setzt an den strategischen Punkten Variable, wie auch sonst bei der Verwendung von Mailmerge, z.B.

```
..-----------------------------------------------------------------------
An Herrn                                                    &datum&
&name&
&strasse&

&wohnort&
..-----------------------------------------------------------------------
```

2. setzt man an den Anfang seines Briefes die benötigten Informationen, z.B.

```
.sv name   =Rudi Ratlos
.sv strasse=Hackergasse 5
.sv wohnort=4711 Stonsdorf
.sv datum  =12.34.56
.fi BRIEFKO.TXT

Lieber Rudi,

bla, bla, bla...
```

Durch die Zuweisung mit ".sv" werden die Variablen auch ohne Daten-File belegt. Der Trick ist nun, daß mit ".fi" der Briefkopf eingelesen wird und die Variablen durch die vorgegebenen Texte ersetzt werden. Daß man den Brief jetzt mit "M" wie Mailmerge statt mit "P" wie Print ausdrucken muß ist nur Gewöhnungssache und im Grunde ohne Bedeutung.

Das Ergebnis sieht dann folgendermaßen aus:

```
An Herrn                                          12.34.56
Rudi Ratlos
Hackergasse 5
4711 Stonsdorf

Lieber Rudi,

bla, bla, bla...
```

Alexander Schmid

Nachtrag zum Banker

Sollte es wahr sein, daß bis jetzt außer Arnulf Sopp und mir noch niemand Probleme mit dem Banker von Helmut Bernhardt hatte ? Keine unmotivierten Abstürze oder Aufhänger ? Na gut, vielleicht hängt das Auftreten auch etwas mit der Taktgeschwindigkeit zusammen und die ist bei uns halt etwas höher. Trotzdem hat die Schaltung einen echten Fehler, den ich hier mit freundlicher Genehmigung von Helmut breittreten darf.

Nach über einem Jahr Suchen und Ärgern habe ich ihn endlich gefunden. Der Käfer besteht darin, daß am 74LS74 auf dem Banker die Pins 11 und 12 vertauscht sind, was auf einen Druckfehler in einer alten c't zurückgeht, aus der Helmut einen Teil der Schaltung übernommen hat. Da vergleicht man immer wieder das Layout mit dem Schaltplan, wundert sich, daß alles stimmt und trotzdem nichts geht und dann sowas. Diese Verwechslung hat die unangenehme Folge, daß der Refresh für die dynamischen RAMs nicht richtig funktioniert und die Bits unter Umständen nach Lust und Laune umkippen. Beim TRS-80/Genie unter NEWDOS oder GDOS macht sich das im Normalfall nicht bemerkbar, weil der Prozessor dort durch die Interrupts kreuz und quer durch's RAM springt und so schon alleine durch die Speicherzugriffe während der Interruptroutinen für eine ausreichende Erfrischung sorgt. Wenn die Interrupts aber abgeschaltet sind, was z.B. bei mir unter CP/M der Fall war, kann es Ärger geben wenn z.B. eine Schleife zu kurz wird.

Klarheit kann man sich mit folgendem Programm schaffen:

```
        ORG  6000h          ;oder sonstwo

start   DI                  ;keinen Mogelrefresh machen !
        LD   A,0            ;Zähler
        LD   B,A            ;  "
        LD   C,A            ;  "
loop    INC  C
        JR   NZ,loop        ;innere Schleife
        INC  B
        JR   NZ,loop
        INC  A
        JR   NZ,loop        ;äußere Schleife
        JP   4400h          ;ins DOS

        END start
```

Durch die kurzen Schleifen bleibt der Speicherzugriff auf wenige Adressen beschränkt und der ganze Rest kann nur durch die Logik auf dem Banker behandelt werden. Wenn sich der Rechner nach 1-2 Wochen noch nicht wieder gemeldet hat, könnt Ihr davon ausgehen, daß im restlichen RAM nur noch Schrott steht, womit sich auch das DOS verabschiedet hat. Nach dem Neubooten dürfte dann überall nur noch wirres Zeug stehen, was Ihr mit Debug o.ä. überprüfen könnt. Wenn alles stimmt, sollte nach ungefähr zwei Minuten wieder das DOS-Ready erscheinen und der Rest des Speicherinhalts unverändert sein. Wem das zu lange dauert, der kann A mit einem größerem Wert laden, wobei es aber auch wieder nicht zu schnell gehen darf, weil die RAMs, wenn sie einen guten Tag haben, durchaus ein paar Sekunden stabil sein können. Die paar Minuten sollte das doch wert sein. Zur überprüfung könnt Ihr vorher noch irgendein Programm in die Ramdisk kopieren, die ja denselben Speicher benutzt. Wenn das Programm nachher noch läuft, ist mit ziemlicher Sicherheit alles in Ordnung. Wenn nicht, solltet Ihr schnellstens den Lötkolben heizen.

Alexander Schmid

Port-I/O die Zweite

Gerald Schröder hat in einem der letzten Infos beschrieben, wie er in seinem Genie die Kommunikation mit Drucker und Floppy umschaltbar von memory-mapped auf Ports gemacht hat. Die Umschaltung erledigt er über ein Bit im Kassettenport, den normalerweise sowieso niemand mehr braucht und den er deshalb radikal mit allen ICs rausgeworfen hat. Da ich den Platinen aber nur sehr ungern Gewalt antue, zumal es keinen Ersatz mehr gibt, habe ich eine andere Möglichkeit gesucht und gefunden. Im Genie IIs, und natürlich auch in anderen CP/M-tauglichen TRS-80 und Genies, gibt es ein Signal mit Namen CPM o.ä. mit dem man die Umschaltung sozusagen umsonst bekommt. Dieses Signal sagt der Banking-Logik nämlich, ob sie den I/O-Bereich freigeben soll, oder ob durchgehend RAM gewünscht wird, womit die mögliche Ein-/Ausgabeart bereits mehr oder weniger festliegt.

Unter GDOS, wenn der I/O-Bereich sowieso freigegeben ist, erfolgt die Ausgabe voll kompatibel zum Original memory-mapped, unter CP/M, bei durchgehendem RAM, über Ports. So erfolgt die Umschaltung automatisch und man muß nicht noch zusätzlich auf irgendwelchen anderen Ports rumfummeln. Wie die Bytes rübergeschaufelt werden ist letztlich ja egal, funktionieren tut beides, nur mußte man unter CP/M bis jetzt immer erst den I/O-Bereich einblenden, das Byte rüberschieben und den I/O-Bereich wieder abschalten. Ziemlich lästig und außerdem braucht es unnötig Speicher. Ganz auf Ports umzustellen wäre natürlich auch noch eine Möglichkeit, aber damit verbaut man sich den Weg, auch schmutzige Programme zu fahren, die unter NEWDOS/GDOS direkt auf die memory-mapped Adressen zugreifen. Ganz abgesehen von der Arbeit mit der Patcherei. Ich finde, daß sich das nur beim CP/M lohnt, wenn man auch den Source-Code hat. Dann wird das BIOS wirklich kürzer und enthält nicht nur zusätzliche NOPs, wenn man z.B. die drei Bytes für LD (37E8H),A gegen die zwei Bytes für OUT (0E8H),A austauscht.

Der praktische Nährwert dieses Artikels ist für die große Mehrheit auch nicht größer, als der von Geralds, aber ich möchte mich seinen Worten voll anschließen: wenn Euch was stört, ändert es einfach.

Alexander Schmid

Ergänzung zu 'Modifikation des HRG-Treibers von RB-Elektronik' Clubinfo 29 Seite 12 ff.

Die Fragezeichen auf Seite 16 möchte ich wie folgt ergänzen:

```
JP Z,0FE76h   ;#Lprint
JP Z,0EEC7h   ;#Not
JP Z,0FBB9h   ;#Inp
JP Z,0FB96h   ;#Out
JP Z,0F5EDh   ;#Fix
JP Z,0F64Bh   ;#Def
```

Übrigens, von FB13h bis FB73h liegen CLoad, CMerge und CSave. Hier kann man noch was reinschreiben. Mit Kassette arbeitet doch niemand mehr.
Hinter #Lprint, so ab FFA0h (je nach Druckerversion) ist auch noch Platz bis FFFFh.

Oskar Drechsler

Turbotape, der Jedermann-Streamer

Beim Durchstöbern meiner Programmsammlung ist mir ein Programm mit dem Namen "Turbotape" aufgefallen, das ein komfortables Backup von Dateien auf jedem normalen Kassettenrecorder erlaubt. Die Schaltung und auch die Bedienung und die Meldungen des Programms erinnern sehr an das "Supertape" aus der c't, wahrscheinlich ist auch das Aufzeichnungsformat identisch. Mangels Programm konnte ich das leider nicht nachprüfen. Im Gegensatz zum original Supertape-Programm arbeitet Turbotape aber auch mit Wildcards, sodaß man z.B. mit *.* als Dateinamen eine ganze Floppy auf einen Schlag sichern und wieder zurücklesen kann. Der Aufwand ist denkbar gering und ich hatte das Platinchen innerhalb einer Stunde mit den Beständen aus der Bastelkiste bestückt und zu meiner Verwunderung funktioniert es einwandfrei. Der Aufnahmezweig ist ein simpler Bandpaß und der Wiedergabezweig braucht neben ein paar Widerständen und Kondensatoren nur ein Feld-, Wald- und Wiesen-IC zur Impulsformung. Die ganze Schaltung sieht folgendermaßen aus:

```
        100n        9 I------I       5 I------I
           CC   I------I      I  10   I---I      I 4
Rekorder )-CC--I      I 4093 I0-----I   I 4093 I0---I---( Daten
Ausgang    CC   I------I      I      I---I      I    I      lesen
        5V      I    8 I------I       6 I------I    I      Bit
         I      RR                                  I
   Poti1 I      RR 22k                              I
   10k  PP      RR                1M                I
      +--PP----I--------------RRRRRR---------------I
         PP
         I
       Masse                     33k    100n  CC
              I---------I-----RRRRRR--------CC----------( Daten
    Poti2 10k PP     1n  I                   CC         schreiben
Rekorder )----PP--+     CCCCC                            Bit
Eingang       PP         I    -Poti1 so abgleichen, daß es in der
              I          I     Mitte der Schaltschwelle von
            Masse      Masse   Ausgang 4 liegt.
                              -Poti2 an Rekorder anpassen.
```

Statt des 4093, den ich in meinem Datenbuch nicht mehr finde, habe ich einen 74HCT00 genommen, der genauso funktioniert. In der c't wird dafür ein 4001 oder 4011 vorgeschlagen, wahrscheinlich kann man so ziemlich alles nehmen.

Die Installation ist sehr einfach. Wenn man einen Parallelport mit einer Z80-PIO hat, braucht man dem Installationsprogramm nur die Portadresse und die Taktgeschwindigkeit zu sagen, der Rest geht von alleine. Wenn man einen anderen Portbaustein einsetzt, muß man die Initialisierungsbytes halt von Hand eingeben, aber das war's dann auch schon.

Die übertragung findet mit 3600 Baud statt. Nicht überwältigend, aber eine 700K-Floppy ist nach rund 20 Minuten gesichert und so oft braucht man seine Backups ja hoffentlich nicht.

Wer seine Backups lieber auf Floppy macht, der soll das bei den heutigen Preisen ruhig tun, aber man kann ja z.B. auch Programme zwischen Rechnern austauschen, deren Format man beim besten Willen nicht lesen kann, sei es weil die nur 3"-Floppies haben oder warum auch immer. über größere Entfernungen ist das bestimmt billiger, als per Modem mit 300 oder 1200 Baud.

Das Programm hat nirgendwo eine Copyrightnotiz, wahrscheinlich wird es der ein oder andere irgendwo in seinen Beständen haben. Wenn nicht, stehe ich gerne zur Verfügung.

Alexander Schmid

BÖRSE -- BÖRSE -- BÖRSE

Aus Alt macht Neu

Oliver Volz

Vielleicht können sich einige von Euch noch an die etwas seltsame Namensgebung unserer Programme von früher erinnern? Da gab es solche Namen wie S0152A04/BAS was heute VBRN555/BAS heißt. Leider habe ich keine Dekodiertabelle von Neu nach alt, wesshalb ich teilweise manche Bestellungen nicht befriedigen kann. Jetzt kramt doch schnell mal diesen Stapel Papier raus und überlegt Euch ob Ihr den wirklich noch braucht, wenn nicht schickt ihn mir.

Hallo Arnulf und sonstige Betriebssystemprogramierer,

verratet mir doch bitte wie ich von meiner DD/DS (720k) Systemdiskette eine SS/SD machen kann. Nachdem ich von Heinrich Betz ein selbiges als Diskothekar zur Verfügung gestellt bekommen habe wollte ich dort mein Betriebssystem deponieren was vor allem beim kopieren von DD/DD nach DD/DD mit meinen beiden anderen Laufwerken sehr von Vorteil ist.
Seid auch noch so nett und erklährt mir warum DIRCHECK/CMD bei mir mein Inhaltsverzeichnis nicht aufräumt.

Zilog-CPU mit 20 MHz

Zilog entwickelte jetzt eine 20-MHz-Version ihrer Z-80-CPU. Der neue Prozessor Z84-C0020 soll doppelt so schnell sein wie jede andere 8-Bit-CPU. Die maximale Rechenleistung liegt bei 5 MIPS. Zilog-Kunden können problemlos ein Upgrading ihrer bestehenden Systeme vornehmen, oft reicht der Austausch von CPU und Quarz. Der Z84C0020 ist voll pin- und software-kompatibel zu den herkömmlichen Z-80-Prozessoren. (gs)

Tel. 02671/5100 **RU-KY** FAX: 02671/8285

Computer - Zubehör - Schulung - Beratung - Verkauf - Service

Rudolf Kyek - Hauptstr. 65 - 5592 Klotten

FARBBÄNDER

Farbbänder Bestellmenge:

A = Stück
B = 5er Pack/Stck
C = 10er Pack/Stck
D = 20er Pack/Stck
E = 50er Pack/Stck

Gruppe 173 C- Film Art.Nr.173450
IBM 6746, u. a.
A= 7,30DM B= 7,10DM C= 6,90DM D= 6,40DM E= 5,90DM

Gruppe 174 Lift-off-Kas. Art.Nr.174450
IBM 6746, u.a.
A= 7,30DM B= 7,10DM C= 6,90DM D= 6,40DM E= 5,90DM

Gruppe 186 C- Film Art.Nr.186450
Nakajima AX 200,210 u.a.
A= 6,50DM B= 6,40DM C= 6,20DM D= 5,60DM E= 5,10DM

Gruppe 187 C- Film Art.Nr.187450
Olympia,Nakajima, u.a.
A= 5,90DM B= 5,70DM C= 5,50DM D= 5,10DM E= 4,60DM

Gruppe 188 C- Film Art.Nr.188450
TA-Gabriele 7007/9009 u.a.
A= 5,80DM B= 5,60DM C= 5,40DM D= 4,80DM E= 4,30DM

Gruppe 143 Lift-off-Kas. Art.NR.143450
Nakajima,Olympia, TA, u.a.
A= 1,20DM B= 1.10DM C= 1,00DM D= 0,85DM E= 0,75DM

Gruppe 615 Nylon schwarz Art.Nr.615450
MT 400,1000,1612,Commodore 8024,Nixdorf ND 15/24/25 Präsident 6310/6313, u.a.
A= 7,90DM B= 7,70DM C= 7,50DM D= 6,60DM E= 5,60DM

Gruppe 621 Nylon schwarz Art.Nr.621450
Citizen 120D,Commodore MPS 1200,u.a.
A= 7,40DM B= 7,20DM C= 7,00DM D= 6,60DM E= 6,20DM

Gruppe 678 Nylon schwarz Art.Nr.678450
Seikosha SP 800/1000/SP1200, Schneider PCW 8256,u.a.
A=10,50DM B=10,30DM C=10,10DM D= 9,70DM E= 9,30DM

Gruppe 680 Nylon schwarz Art.Nr.680450
Seikosha SP 80,Commodore VC 1515,u.a.
A= 9,50DM B= 9,30DM C= 9,10DM D= 8,30DM E= 7,40DM

Gruppe 686 Nylon schwarz Art.Nr.686450
Star NL 10/ND 10/NG 10,u.a.
A= 8,50DM B= 8,30DM C= 8,10DM D= 7,40DM E= 6,50DM

Gruppe 771 Nylon schwarz Art.Nr.771450
Star LC-10/NX 1000,
A= 6,40DM B= 6,20DM C= 6,00DM D= 5,80DM E= 5,50DM

Gruppe 786 Nylon schwarz Art.Nr.786450
Star LC 24-10,
A= 9,90DM B= 9,70DM C= 9,50DM D= 9,10DM E= 8,70DM

Gruppe 624 Nylon schwarz Art.Nr.624450
Commodore MPS 803,Atari Serie 600/800, Brother M1009, Centronic GLP 3000/3101,Schneider NLQ 401,u.a.
A= 7,90DM B= 7,70DM C= 7,50DM D= 7,10DM E= 6,60DM

Gruppe 625 Nylon schwarz Art.Nr.625450
Centronic 150/152/154/159,Commodore, Fujitsu Micro 7/168, u.a.
A= 8,40DM B= 8,20DM C= 8,00DM D= 7,50DM E= 6,60DM

Gruppe 629 Multicarbon Art.Nr 629550
Commodore MPS 802,Atari SMM/XMM 801, Mannesmann Tally MT-80/81,
A= 6,90DM B= 6,70DM C= 6,50DM D= 6,30DM E= 6,10DM

Gruppe 629 Nylon schwarz Art.Nr.629450
Commodore MPS 802,Atari SMM/XMM 801, Mannesmann Tally MT-80/81,
A= 9,90DM B= 9,70DM C= 9,50DM D= 8,90DM E= 8,20DM

Gruppe 633 Nylon schwarz Art.Nr.633450
Citizen HQP 40, Epson LQ 500/800/850,u.a.
A= 9,90DM B= 9,70DM C= 9,50DM D= 8,80DM E= 7,70DM

Gruppe 635 Nylon schwarz Art.Nr.635450
Commodore 4020, Epson FX 80/85/800,MX 80,FT 80, Citizen LSP 10,MSP 10/20 u.a.
A= 6,40DM B= 6,20DM C= 6,00DM D= 5,80DM E= 5,50DM

Gruppe 636 Nylon schwarz Art.Nr.636450
Epson MX 100,FX100/105/1000,RX 100,u.a.
A= 9,90DM B= 9,70DM C= 9,50DM D= 8,80DM E= 7,90DM

Gruppe 638 Nylon schwarz Art.Nr.638450
Epson LX 80/90,GX 80,Commodore MPS 1000, u.a.
A= 7,50DM B= 7,30DM C= 7,10DM D= 6,70DM E= 6,10DM

Gruppe 650 Nylon schwarz Art.Nr.650450
Itoh 1550/8500/8510A/C-310,Apple Imagewriter, Commodore 8023 BC,NEC PC 8023 A-B-C /8025,u.a.
A= 8,50DM B= 8,30DM C= 8,10DM D= 7,50DM E= 6,50DM

Gruppe 666 Nylon schwarz Art.Nr.666450
Commodore MPS 2000,NEC P1/P2/P6, u.a.
A= 8,90DM B= 8,70DM C= 8,50DM D= 7,90DM E= 7,30DM

Gruppe 783 Nylon schwarz Art.Nr.783450
NEC P6/P/7 Plus,
A=12,90DM B=12,70DM C=12,50DM D=10,90DM E= 9,90DM

Gruppe 762 Nylon schwarz Art.Nr.762450
NEC P 2200,
A= 9,90DM B= 9,70DM C= 9,50DM D= 9,30DM E= 9,00DM

PC-Druckerkabel 25-Pol.Sub-D Stecker an 36-pol. Centronic-Stecker (Standard IBM)
Art.Nr.171400 9,90DM Stck/ 8,90DM 20erPack/Stck

Andere Farbbänder auch farbige auf Anfrage lieferbar.

AB DM 100,- frei Haus per Nachnahme oder Vorkasse

(bis 100,-DM Bestellwert DM 5,50 Versandkosten)
(Mindestbestellwert: DM 30,-)
Preise incl. Mwst.

Einführung in Mikrocomputer-Betriebssysteme
5. Teil: Geisterstunde: Virtuelle Geräte

von Gerald Schröder

a. Gut gedruckt ist halb gegeistert

Wie versprochen: Die virtuellen Geräte sind am rumgeistern, auch wenn es sie nicht gibt. Was sucht ein Gerät, das es nicht gibt, in/am unserem Rechner? Ganz einfach: Es narrt unser selbstgeschriebenes Programm und gaukelt ihm vor, es gäbe ein Gerät, wo es in Wirklichkeit gar keins gibt.

Halt, so natürlich nicht! Natürlich gibt es schon ein Gerät, nur eben nicht das Gerät, von dem das Programm glaubt, daß es da ist. Seht Euch zum Beispiel mal das Handbuch Eures Druckers an. Dort findet Ihr irgendwo einen Hinweis, der ungefähr so lautet: "Kompatibel zu / emuliert: Epson-LQ-1500, NEC-P5/6/7/CP6/CP7/P5XL/P9XL" (so im Handbuch zum NEC-P2200).

Was hat das nun mit "virtuellen Geräten" zu tun, wo doch unser Drucker leibhaftig noch da steht? Aber seht mal genau hin: Da steht nur ein mieser kleiner P-2200, kein Epson-LQ-1500 neben einem NEC-P6 usw. Aber "virtuell" sind alle diese teueren Druckr vorhanden, denn mein P-2200 macht (fast) dasselbe wie sie. Und in meinem Textverarbeitungsprogramm sage ich dann großspurig: "Ausdruck bitte auf einem Epson-LQ-1500" und es klappt (meistens) auch! Das Programm "denkt" in dem Moment, daß der Epson da steht und druckt, und mein P2200 lacht sich kaputt.

Einen kleinen Haken hat die Sache natürlich: Der P2200 kann nicht alles, was die anderen Drucker können. Und wenn das Textverarbeitungsprogramm etwas davon benutzt, merkt es selber zwar nichts, aber der P2200 schaut schön blöd aus der Wäsche, weil er keine Ahnung hat, wie er z.B. DIN-A4 quer und in Farbe drucken soll.

Natürlich gilt das nicht nur für meinen Drucker. Wer hat nicht schon von "Postscript" gehört, was nun jeder Laserdrucker können sollte; oder dem "Hayes-Modem" (die Firma (?) "Hayes" baut nun nicht millionenfach Modems, sondern andere Firmen bauen welche, die sich genauso verhalten); oder den "SCSI-Festplatten" usw. (SCSI=Small Computer Systems Interface, wobei dann sehr unterschiedliche Festplatten o.ä. dahinter stehen)

Verallgemeinern wir das Ganze mal: Unser Programm "denkt", daß irgendwo ein bestimmtes "virtuelles Gerät" (Epson-Drucker, Postscript-Drucker, Hayes-Modem, ...) vorhanden ist und gibt diesem Gerät die Befehle, die es verstehen müßte (eben "drucke 150 Zeichen/Zeile in roter Schrift"). Unser echtes ("physikalisch vorhandenes") Gerät muß nun damit klarkommen.

Was machen wir aber, wenn unser Textverarbeitungsprogramm nur mit IBM-Druckern (als virtuellen Geräten) umgehen kann, unser armer NEC-P2200 das nun aber partout nicht versteht? (Was sich meistens in einem wirren Buchstabensalat äußert.) Da kommt unser Betriebssystem ins Spiel.

b. Geister-Beschwörung

Was soll das BS denn nun plötzlich tun? Nun, bisher haben wir gesagt, unser "echtes" Gerät muß sich so verhalten wie das "virtuelle", auf das unser Programm sich verläßt. Aber wenn wir das Betriebssystem dazwischenschalten, können wir sagen: Das Betriebssystem muß sich so verhalten wie das virtuelle Gerät und hat dafür zu sorgen, daß das echte Gerät keinen Mist baut. Also gibt das Textverarbeitungsprogramm dem Betriebssystem alle Befehle wie für einen IBM-Drucker. Unser Betriebssystem sieht sich die Befehle an, entscheidet, wie die entsprechenden Befehle für unseren echten Drucker aussehen und läßt ihm diese (statt der IBM-Befehle) zukommen.

Ein fiktives Beispiel:
Wordstar an BS : "drucke 10 Zeichen/Zeile" = 1Bh 55h
BS an NEC-P2200: "drucke 10 Zeichen/Zeile" = 1Ch 01h 00h

Wie wir sehen, hat sich die Sache etwas verlagert: Vorher hat unser Drukker so getan, als wäre er ein anderer, nun macht das unser Betriebssystem. Heutzutage ist (noch?) der erste Fall aktuell: die echten Geräte (Drucker, Modem, Festplatte, ...) sind selber dafür zuständig, sich richtig (wie das virtuelle Gerät) zu verhalten. Da es aber z.B. kein virtuelles Gerät "DRUCKER" gibt, das alle echten Geräte emulieren (umgangssprachlich: "nachmachen"), und somit jeder Drucker etwas mehr oder weniger als ein anderer kann, müssen wir immer noch unserem Wordstar sagen, mit was für einem Drucker er es zu tun hat. Also werden tausend "Drucker-Treiber" mitgeliefert (zu jedem neuen Textverarbeitungsprogramm wieder tausend) und wir müssen den auswählen, der für unseren Drucker gilt. Falls es den nicht gibt (z.B. habe ich noch keinen für den P2200 gesehen), müssen wir einen für einen Drucker wählen, den unser Drucker emulieren kann. Womit wir wieder bei den "virtuellen Geräten" wären.

c. Was treiben die Geister?

Hat es bei dem Wort "Treiber" nicht bei Euch geklingelt?! Wenn nicht, schaut Euch mal den 3. Teil dieser Serie an ("Was treibt ein Treiber?"). Ja, unser Betriebssystem hat auch massig Treiber, meistens BIOS genannt. Warum gibt es dann nicht die oben erwähnten tausend Drucker-Treiber für unser Betriebssystem? Wir müßten dann nur einmal (beim Kauf des Druckers/-Betriebssystems) den richtigen Druckertreiber auswählen und in unser Betriebssystem "einbinden". Danach könnte <u>jedes</u> Textverarbeitungsprogramm mit unserem Drucker umgehen, ohne jegliche eigene Treiber. Oder?!

Also ich würde das für den natürlichen Weg halten, aber es gibt ein Problem: Was wäre unser "virtuelles Gerät"? Vielleicht ein Fernschreiber mit 70 Zeichen/Zeile, ohne jegliche Schriftarten/-breiten usw.? Da würden sich die Laserdrucker-Besitzer (und nicht nur die) freuen, daß ihr teurer Drukker nur 70 Zeichen/Zeile druckt (aber das wahnsinnig schnell). Nix mit hochauflösender Grafik, Umschaltung zwischen Courier, Roman, Helvetica, doppelt-breit, siebenfach-hoch usw. Das kann unser virtuelles Gerät "Fernschreiber" nicht, also können wir es unserem Betriebssystem nicht sagen, auch wenn unser echtes Gerät viel mehr kann.

Also die zweite Möglichkeit: Wir nehmen den besten Drucker, den es gibt, und der wird unser virtuelles Gerät. Aber halt: Was machen all die armen Schlucker, die sich nur einen P2200 (oder einen Fernschreiber) leisten können? Sense, die können nicht mal ein Zehntel der Fähigkeiten des virtuellen Geräts mit ihrem echten Gerät erreichen. Oder: Der virtuelle Drukker arbeitet mit 360 Punkten/Zoll, unser Drucker aber nur mit 300. Das ist fast genauso gut, aber eben nicht ganz. Entweder verzichten wir ganz auf den Grafik-Ausdruck oder unsere Grafiken sehen (bestenfalls) etwas seltsam aus.

Und noch ein Punkt: Welches ist "das beste Gerät"? Es gibt höchstens ein <u>momentan</u> bestes Gerät, das aber morgen schon wieder veraltet ist (360 Punkte/Zoll? Quark, 400, 500, 1000 müssen es sein!).

d. Geister-Video

Gehen wir mal weg von den Druckern zu einem Bereich, wo die virtuellen Geräte schon verbreiteter sind: den Bildschirmen. Den alten CP/Miern wird es schon bekannt sein: "ADM-3A kann jeder". Was soll das heißen? Nun, es gab mal irgendwann ein Terminal, das wohl "ADM-3A" hieß und folgende Sachen konnte:

- 80 Zeichen x 24 Zeilen darstellen
- den Bildschirm löschen
- den Cursor auf einen beliebigen Punkt auf dem Bildschirm setzen
- ein Zeichen an der Cursorposition anzeigen
- invertierte Schrift ein- und ausschalten.

Das war's (glaube ich jedenfalls).

Darauf stellten sich die Programme ein. Mehr gab es nicht und mehr wurde nicht benutzt. Auch wenn ein später erschienenes Terminal mehr kann (z.B. Televideo 950: helle/dunkle Schrift, unterstreichen, blinkende Schrift, Zeilen löschen/einfügen, ...), benutzt das kaum ein Programm: Es ist eben auf ADM-3A ausgerichtet und basta. Vielleicht können wir noch einstellen, ob wir statt "invertieren" lieber "hell/dunkel" haben möchten, aber beides gleichzeitig geht eben nicht!

Hier haben wir ein typisches virtuelles Gerät, das gerade genug kann, aber doch so einfach ist, daß jedes Terminal das schafft. Übrigens sollte sich jeder von Euch mal ein CP/M-BIOS für TRS-80 (mit oder ohne Schmidtke-Karte) ansehen; da könnt Ihr dann sehen, wie schwer es ist, auf diesem /Zensur/ Gerät allein das zu schaffen, was das ADM-3A kann. Nicht auszudenken, wenn das Televideo-Teil Standard (=virtuelles Gerät) geworden wäre! Dann wären der TRS-80 und CP/M sich ewig fremd geblieben. (MS-DOS scheint das dann verwirklicht zu haben, während Atari sich auf ein "VT-52" stützt, was auch nicht viel mehr als das ADM-3A kann.)

e. Disketten-Geister

Und noch ein virtuelles Gerät aus dem CP/M: Unsere "logischen" Disketten. Jeder Newdos-CP/M-Umsteiger hat ein Problem: Warum kann ich (meistens) nicht einstellen, daß meine Disketten 40 Tracks in doppelter Schreibdichte haben? Warum heißt es (beim Montezuma-CP/M): Die Diskette hat 160 Tracks, obwohl doch kaum das Laufwerk nur 80 hat? Warum gibt es im CP/M nur einseitige Disketten? (Obwohl doch auf der Rückseite auch was steht...?) usw.

Das hat einen einfachen Grund: Das BDOS. Klick, klick, na, fällt der Groschen? Der feste Teil unseres Betriebssystems, ja, siehe 4. Teil dieser Serie. Das BDOS im CP/M geht bei den Massenspeichern (Diskette, Festplatte) logisch immer davon aus, daß diese Massenspeicher in Tracks und Sektoren aufgeteilt sind. Jeder Sektor hat 128 Bytes und jeder Track kann bis zu 65535 Sektoren haben und eine Diskette kann (wahrscheinlich) 65535 Tracks haben (Allerdings gibt es noch einige interne Einschränkungen, so daß CP/M-Laufwerke m.W. nur 32 MByte groß sein dürfen.)

Das BDOS sagt zum BIOS immer nur: "Beschaffe mir Sektor x von Track y". Also keine Vorder-/Rückseite! So kommt es zu den obengenannten 160 Tracks: 80 vorne, 80 hinten! Beim nächsten CP/M sind es dann nur 40 Tracks, dafür hat jeder Track aber viermal soviele Sektoren, so daß wir insgesamt dieselbe Diskettenkapazität vorliegen haben (also das gleiche echte Laufwerk!).

Wieviele (logische) Sektoren je (logischen) Track und wieviele (logische) Tracks je (logische) Diskette es gibt, legt eine Tabelle im BIOS fest ("Disk Parameter Block", s. CP/M-Literatur). Auch hier finden wir konsequenterweise keine Angaben über Seiten oder Schreibdichte o.Ä., sondern nur noch einige weitere Parameter, die aber nichts über unser echtes Gerät (Disketten-/Festplattenlaufwerk) aussagen. Alle Angaben darüber sind im BIOS versteckt und sehen auf jedem CP/M-Rechner anders aus. Wie hätten auch die CP/M-Entwickler in den siebziger Jahren schon wissen sollen, daß es mal Laufwerke mit einfacher, doppelter und hoher Schreibdichte geben wird (SD, DD, HD)? Oder daß Sektoren außer einer Länge von 128 Byte auch 256, 512, 1024 usw. Byte lang sein können?

Damit erklärt sich auch, daß eine Einstellung der echten Geräte-Daten über das CP/M nicht möglich ist. Wenn, dann geht diese Einstellung nur auf einem bestimmten CP/M-Rechner für eine bestimmte BIOS-Version, aber beim nächsten BIOS oder Rechner sieht es wieder ganz anders aus. Wir können beim CP/M nur die Daten des virtuellen Geräts ändern (über den DPB).

Falls jemand oben über die "logische Diskette" gestolpert sein sollte: Warum nehmen wir nicht die Vorderseite der echten Diskette als eine logische (z.B. "A:"), die Rückseite aber als eine andere (z.B. "B:"). Oder warum soll nicht die gleiche echte Diskette einmal logisch als Diskette "A:" mit 160 Tracks und als Diskette "C:" mit 40 Tracks angesprochen werden können? Das kann man im CP/M alles problemlos machen!

Nur am Rande: Eigentlich sollte sich ein Programmierer unter CP/M keine Sorgen um (logische/virtuelle) Tracks/Sektoren machen, denn seine Schnittstelle zu den CP/M-Massenspeichern sind die logischen Laufwerke ("A".."P") und die Files! Für deren Verwaltung ist das BDOS eigentlich da.

f. Geister in der Schule: Geisterklassen

Einen hab' ich noch! Nein, kein virtuelles Gerät. Aber erinnert sich vielleicht noch jemand an die Klassen aus Teil 2? Da war auch schon die Rede von "Massenspeichern" und "Druckern" als Klasse. Und wirklich: Bei den Massenspeichern haben wir ja einen Zusammenhang zwischen der Klasse und den virtuellen Geräten gefunden. Es würde heute niemandem einfallen, zu jeder Textverarbeitung tausend Disketten- und Festplattentreiber mitzuliefern, damit wir uns dann die aussuchen, die für unser System passen. Das macht alles das Betriebssystem mit seinen Treibern.

Ganz anders dagegen bei den restlichen Klassen: Während kaum jemand weiß, wie der Schreib-/Lese-Kopf eines Diskettenlaufwerks angesteuert wird, müssen wir uns um den Druckkopf des Druckers sehr wohl Gedanken machen. Er will genaue Befehle darüber, welche Nadeln er wie verwenden soll. Glücklicherweise nehmen uns das die Treiber der Textverarbeitungen ab, aber wehe sie können mal was nicht (z.B. eine Grafik ausdrucken). Dann gute Nacht!

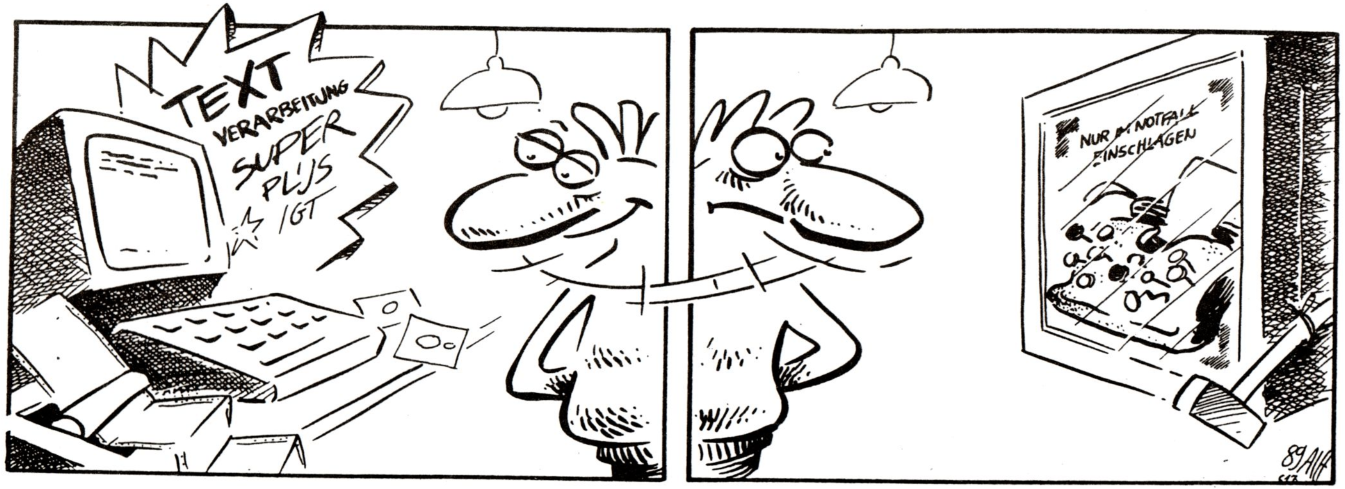
9. Die Geister, die ich rief...

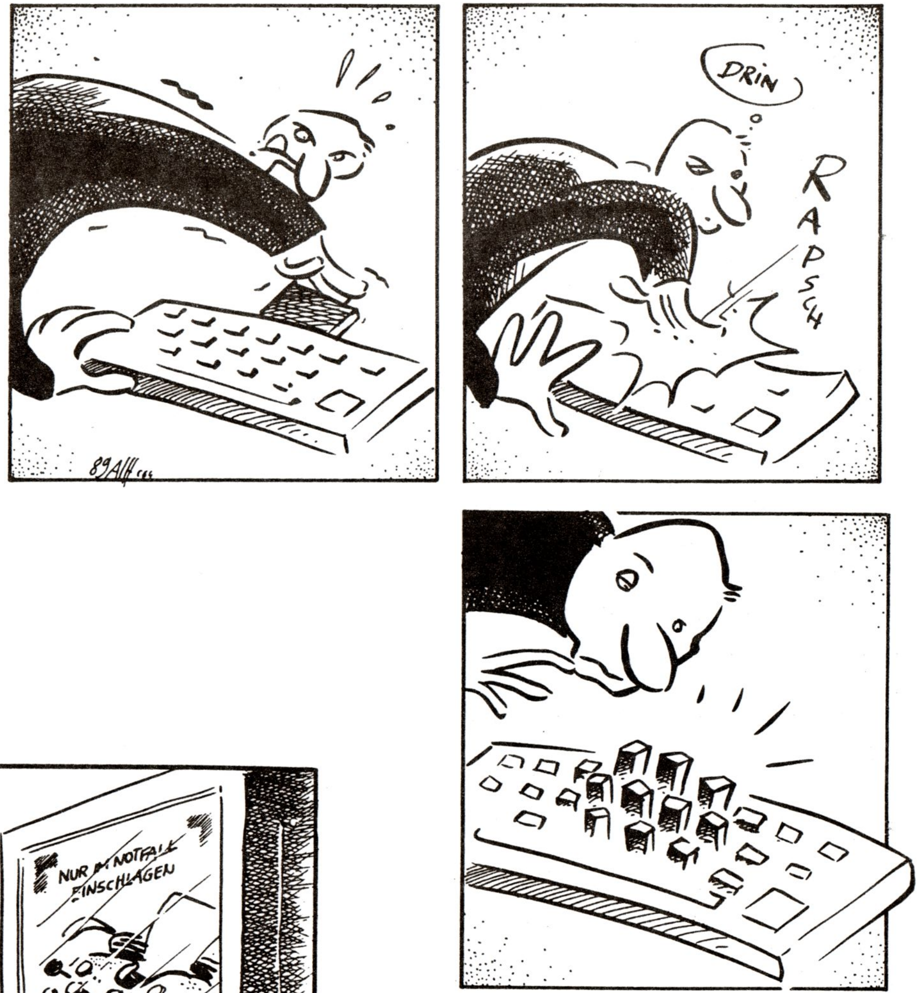
Fassen wir zusammen:

Virtuelle Geräte stehen zwischen dem Anwenderprogramm und dem echten Gerät, wobei der 'Standpunkt' heutzutage meistens im echten Gerät zu suchen ist, so sich dieses überhaupt an irgendein virtuelles Gerät hält. Ausnahme: Massenspeicher.

Wünschenswert wäre es, unser Betriebssystem aus lauter virtuellen Geräten aufzubauen und ihm die Umsetzung auf die echten Geräte zu überlassen, aber das ist heute meistens nicht der Fall. Es ist vor allem ein Problem, 'das' virtuelle Gerät zu finden (welches dann unser BS emuliert).

Die Grundlagen sind nun gelegt. Über Betriebssysteme gibt es wohl noch viel zu sagen, aber eigentlich wollte ich ja einen Vergleich zwischen GDOS und CP/M machen. Das muß ich nun in dem einen (oder anderen?) Artikel nachholen. Falls Ihr noch Anregungen oder Kritik auf dem Herzen habt: Her damit! Sei es inhaltlich, stilistisch oder sonstwas. Je nachdem werde ich mich dann entschließen, noch was dranzuhängen (wie wär's mit 'Speicherverwaltung', 'BS-Oberflächen', 'Geräte-Ansteuerung', 'Files'?) oder das Schreiben aufzugeben (falls keine Reaktionen kommen).





- I M P R E S S U M -

1. Vorsitzender

Alexander SCHMID
St.Cajetan Str. 38 VII
8000 München 80
☎ 089 /495326

2. Vorsitzender

Horst-Dieter SCHROERS
Breslauer Str. 9
8016 Feldkirchen
☎ 089 /9032615

Hardwarekoordinator

Andreas MAGNUS
Pommernstr. 4
4650 Gelsenkirchen
☎ 0209 /870230

NEWDOS-Diskothekar

Oliver VOLZ
Waldburgstr. 73
7000 Stuttgart 80
☎ 0711 /7353817

CP/M -Diskothekar

Rüdiger SöRENSEN
Thomas-Mann-Str. 3a
6500 Mainz 1
☎ 06131 /32860

Club-Bücherei

Christian MENK
Ollsener Str. 52
2116 Hanstedt
☎ 04189 /78251

Redaktion

Jens NEUEDER
Gschlachtenbretzingen
Rudolf-Then-Straße 32
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren

Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung

des CLUB 80

PostgiroKonto

	Sonderkonto CLUB 80
	Obermann,H. 6209 Heidenrod
Konto-Nummer	496 071 - 605
Postgiroamt	Frankfurt
BLZ	500 100 60

SCHLUSS

Hallo Club-80er,

Kurz vor unserem Club-Treffen in Weilburg habe ich nun das 30. der CLUB-Infos fertiggestellt.
Eigentlich wollte ich ja schon viel früher mit dem Info herauskommen, aber leider fehlten mir mal wieder die Artikel.

Aus diesem Grunde finde ich es auch ganz gut, daß das Thema Clubinfo ab und zu mal wieder zur Diskussion gestellt wird.
Ich möchte einmal Euer aller Meinung zu diesem Thema hören.

Den "Gedanken" von Oskar kann ich zustimmen. Nur zu dem 2-monatlichen INFO-Termin möchte ich anmerken, daß wir damit eine regelmäßige und aktuelle CLUB-80-Kommunikation erreichen wollten. Dies sollte unter anderem auch der Dicke unserer CLUB-Info dienlich sein. Schade, daß es nur ein guter Vorsatz blieb. Der große Teil der Club-Mitglieder entschied sich dafür, passiv zu bleiben und dafür lebhaft nach dem Erscheinen der nächsten Clubinfo zu fragen.

Ich hoffe, daß wir uns zur Jahreshauptversammlung auf ein neues Schema zur Puplikation unseres INFO's einigen können.

Bis zum Erscheinen des folgenden CLUB-Info's viel Spaß

Als Beilage zu diesem INFO findet Ihr diesmal:

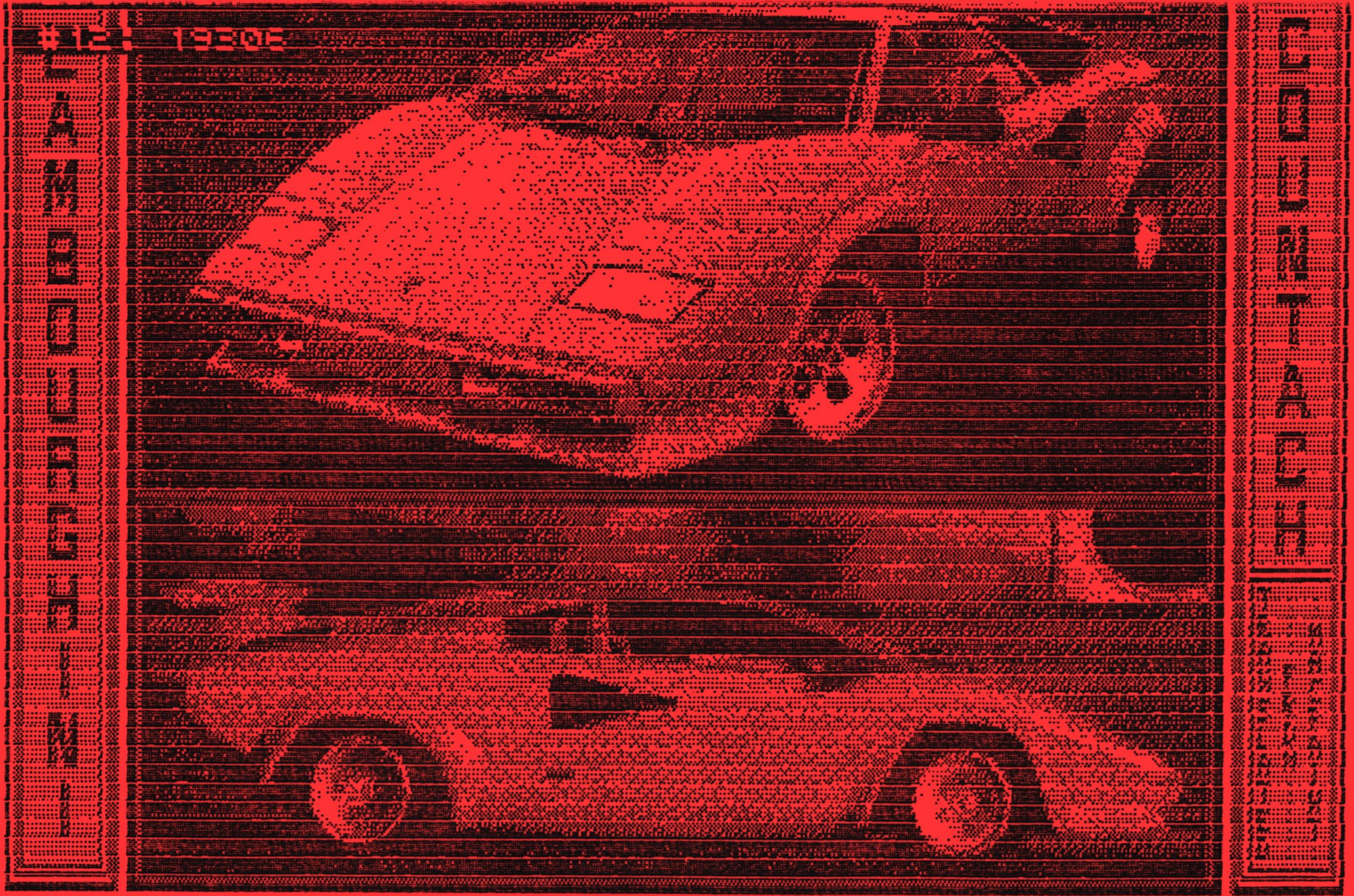
einen Ausschnitt aus THE PUBLIC SOFTWARE LIBRARY
sowie eine MC-Inhaltsübersicht über TRS-80 & ....

Es grüßt Euch Euer

Jens

HEFT
30
April
1990

Eine Zensur oder Kontrolle
der Infobeiträge erfolgt nicht.
Die Redaktion



CLUBINFO

31. AUSGABE

TADRESSE : CLUB 80 / FRITZ CHWOLKA / SAARSTRAßE 34 / 5173 ALDENHOVEN
TEL.: 02464 / 8920

- I N H A L T S V E R Z E I C H N I S -

Seite: und Autor:

Clubinternes

Neues vom Vorstand .... 01 - 03
Gerald Schröder
Clubtreffen in Spremberg/DDR .... 04 - 05
Nachbemerkung zu Spremberg 06
Bilderseite vom Clubtreffen .... 07 - 08
Arnulf Sopp
Termine 46
Redaktion

Software

Turbo-Pascal und ZCPR .... 09 - 11
Alexander Schmid
ZEUS und der HD64180 oder Z180/280 . 12 - 13
Programme relozieren .... 15 - 16
Arnulf Sopp
Harddisk-Einbindung ins CP/M-Bios .. 17 - 22
Helmut Bernhardt, Peter Petersen

Hardware

Die Clubeigene Grafikkarte .... 23 - 24
Andreas Magnus
Club 80 Projekt .... 25 - 32
Helmut Bernhardt
IMB-Banker mit IMB 33
Alexander Schmid

Börse

Wer hat was -- wer will was 34

Sonstiges

Schnittstellen-Programmierung ...... 35 - 41
Von 80 auf 180 oder 280? .... 43 - 44
Gerald Schröder

Club Bibliothek

Buchvorstellung 45

Die letzten Seiten

Impressum 47
Schluß 48
Redaktion
Bildseiten Sprembergtreffen .... am INFO-Ende
Arnulf Sopp
Mitgliederadressenliste .... am INFO-Ende
Fritz Chwolka

Club 80-Hauptversammlung 1990 - das Ereignis des Jahres

Vom 28.4. bis 1.5. fand die diesjährige Hauptversammlung statt, deren Ergebnisse ja schon im letzten Info kurz wiedergegeben wurden. Hier nochmal ein etwas längerer Bericht.

Anwesend waren (und haben damit das Klassenziel erreicht): Helmut Bernhardt, Heinrich Betz, Jörg Brans, Fritz Chwolka, Manfred Held, Jürgen Kemmer, Andreas Magnus, Kurt Müller, Jens Neueder, Stefan Nitschke, Hartmut Obermann, Alexander Schmid, Gerald Schröder, Arnulf Sopp, Rüdiger Sörensen, Oliver Volz. (sowie kurzzeitig Horst-Dieter Schroers)

Versammlungsort war diesmal Weilburg, ein kleines Städtchen an der Lahn, das der mitgereisten holden Weiblichkeit genügend Sehenswürdigkeiten und das richtige Wetter zum Erkunden bot. Wir Herren/Hacker dagegen bekamen nur den Keller in unserem Hotel zu sehen, denn da tagten (?) wir bis spät in die Nacht. Da das Hotel gleichzeitig ein italienisches Restaurant beherbergte, brauchten wir nicht mal zum Essen das Gebäude verlassen.

Im Tagungsraum fand sich eine bunte Mischung aller möglichen Rechner, aber es war diesmal kein TRS-80 Model I/III/Genie I mehr zu sehen. Das zeigt m.E. den allmählichen Wandel des Club 80 in Richtung der neueren Nachbauten (Model IV(p), Genie IIs/IIIs) und der CP/M-Welt. Natürlich wurden zu Vergleichszwecken auch einige "Gastrechner" neben "die schnellsten ...-Rechner der Welt" (c't anno dazumal) gestellt.

Im einzelnen waren vertreten: Model IVp (3x), Genie IIIs (2x), Genie IIs (2x), Model IV, Prof-180X, Tatung (CP/M). Als Gäste: Acorn Archimedes, MS-DOS (2x), Atari-Mega-ST.

Bemerkenswert finde ich, daß 5 Rechner hauptsächlich unter CP/M liefen, wobei drei als Oberfläche ZCPR benutzten. Außerdem konnten wir in Arnulfs Genie IIIs einen Z180-Prozessor bewundern, während Fritz schon Werbung für einen Z280-Rechner machte.

Es stellte sich als eine der wichtigsten Aufgaben des Treffens heraus, das CP/M-Format von Fritz' Tatung-Rechner zu knacken, was Rüdiger und Hartmut dann auch gelang. Übrigens halten nach den Megabyte RAM auch Harddisks in die Z80-Welt Einzug: Jörgs Model IVp, Hartmuts Prof-180X sowie Arnulfs und Andreas' Genie IIIs sind damit ausgestattet. (Der Tatung steht kurz davor.)

Neben dem regen Gedankenaustausch fand natürlich auch ein (kurzer) offizieller Teil statt. Der wurde aber auf der Terrasse im strahlenden Sonnenschein durchgeführt, weil unsere Herren Hacker sonst gar nicht hätten sagen können, wie das Wetter draußen eigentlich aussieht. Also brieten 16 Club-Mitglieder (und einige beratende bessere Hälften) in der Sonne und lauschten den direkt nebenan vorbeirauschenden Pkws und Lkws. Alle Abstimmungen fielen dementsprechend 16:0 aus, d.h. es gab einfach keine Gegenstimme, wobei nachher einige Teilnehmer von den Ergebnissen überrascht waren (dafür aber sehr interessante Gespräche mit ihren Nachbarn geführt hatten).

Zuerst mußten wir entsetzt zur Kenntnis nehmen, daß die Kassenlage katastrophal ist. Falls die letzten 20 säumigen Mitglieder ihren Beitrag auch noch bezahlen sollten, würden wir fast DM 7000 (siebentausend) in der Kasse haben, was ein Rekord ist. Allerdings müssen wir je (erscheinendem) Info im Moment mit ca. DM 600 Kosten rechnen. Aber die Hoffnung, daß wir die Kohle so wieder loswerden, schwindet zusehends, denn unser Club-Redakteur Jens Neueder mußte uns berichten, daß keine Artikel mehr vorliegen. Daraufhin wurde beschlossen, ab jetzt solange "Not-Infos" (Deckblatt & Impressum) herauszubringen, bis Ihr Euch wieder bequemt, Eure Gedanken, Erfahrungen und Telefongespräche auf Papier zu bringen.

Eine andere Möglichkeit, diesen Geldberg abzubauen, wurde abgelehnt: Die Beiträge werden nicht gesenkt und es werden auch keine Staffel-Beiträge oder Honorare für fleißige Autoren eingeführt. Diese Maßnahmen hätten nur zur Verwirrung geführt und zur Belustigung beigetragen, aber sonst kaum Ergebnisse gezeigt.

Die Zahl der Mitglieder hält sich relativ konstant bei 70 Mann und einer Dame (Emanzipation, wo bist Du geblieben?). Allerdings könnte sich diese Zahl durch "verdeckte" Austritte, indem einfach der Beitrag nicht bezahlt wird, noch verringern.

Wie in allen aktuellen Diskussionen, mußten wir uns natürlich auch mit der DDR befassen. Wir haben ja schon die ersten Ost-Mitglieder, hatten bisher aber Probleme mit Disketten-Tausch und persönlichen Besuchen (vielleicht erinnert sich noch jemand an "damals"?). Das sollte sich jetzt grundlegend ändern und in absehbarer Zeit wird auch die Bezahlung des Clubbeitrages, wenigstens von der Währung her, einfacher werden. Ergebnis: Bis zur nächsten Hauptversammlung gilt: alle Interessenten aus der DDR können kosten- und beitragslos Mitglied werden. Im nächsten Jahr sehen wir weiter.

Übrigens vermuten wir ein größeres Mitglieder-Potential in den Ost-Gebieten, denn dort ist der Z80 ja noch up-to-date, was man (leider oder glücklicherweise?) hier nicht mehr sagen kann. Schätzungsweise wird "drüben" auch mehr gebastelt als in unserer "Wegwerf-Gesellschaft", was zu interessanten Einblicken führen könnte.

Nochmal kurz zum Info: Das Drucken klappt, wobei Jens sogar weniger Arbeit als vorher hat, weil die Infos schon gebunden werden. Eine zweite Druck-Möglichkeit hat Helmut aufgetan, wobei die Kosten ähnlich günstig sind wie die jetzigen. Dementsprechend war der Kopierer-Kauf auch kein Thema mehr. Über die Artikel-Ebbe hatte ich ja schon weiter oben berichtet.

Weiterhin wurde beklagt, daß auf veröffentlichte Artikel selten eine Reaktion erfolgt. Auch Aufrufe (so der von Rüdiger zur Mitarbeit an einem erweiterten BIOS) liefen ins Leere. Wir möchten uns/Euch deshalb bitten, die Artikel nicht nur zu lesen, sondern auch zu überlegen, ob sie eine Antwort wert sind. Ähnlich sieht es bei unseren anderen Anlaufstellen aus: Sowohl im Hardware- wie auch im Software-Bereich seid Ihr scheinbar wunschlos glücklich bzw. voll ausgestattet. Glückwunsch!

Womit wir schon beim Vorstand bzw. den mehr oder weniger offiziellen Posten des Clubs wären. Das Gros behält seine Posten, nur die Spitze wurde ausgetauscht und Kurt zum neuen Bücher-Menschen ernannt (weil nicht bekannt war, ob Christian das weitermachen möchte). Die Verteilung sieht jetzt so aus:

1. Vorsitzender	: Fritz Chwolka
2. Vorsitzender	: Gerald Schröder
Kassenwart	: Familie Obermann
Info-Redakteur	: Jens Neueder
Hardware-Koordinator	: Andreas Magnus
Bücher	: Kurt Müller
Newdos-Programme	: Oliver Volz
CP/M-Programme	: Rüdiger Sörensen
Model-IV(p)	: Jörg Brans

Der letzte "Posten" ist neu: Jörg hat sich angeboten, die Model IV(p)-User zu betreuen. Durch Jörgs Kontakte zum großen Bruder (USA) hat er einige interessante Erweiterungen (RAM, Textverarbeitung, Festplatte) und Literatur aufgetan, die auch für die anderen Model IV(p)-Besitzer interessant sein könnten. Vielleicht meldet Ihr Euch mal bei ihm?

Der neue Vorstand hat hoch und heilig versprochen, genauso viel Arbeit in den Club zu stecken, wie das die Mitglieder offensichtlich tun (z.B. beim Schreiben von Artikeln). Weiterhin wird unser neuer Erster auch noch etwas Werbung in seiner CP/M-Gemeinde (er betreut eine Liste mit ca. 100 CP/M-Usern) und in der DFÜ-Szene machen. Begleitet wird diese Aktion von der Veröffentlichung der neuen Adresse in diversen Zeitschriften. Übrigens könnt Ihr auch von Euch aus Werbung für den Club (z.B. in Anzeigenblättern oder Mailboxen) machen. Oder teilt dem Vorstand mit, wo wir noch inserieren sollten.

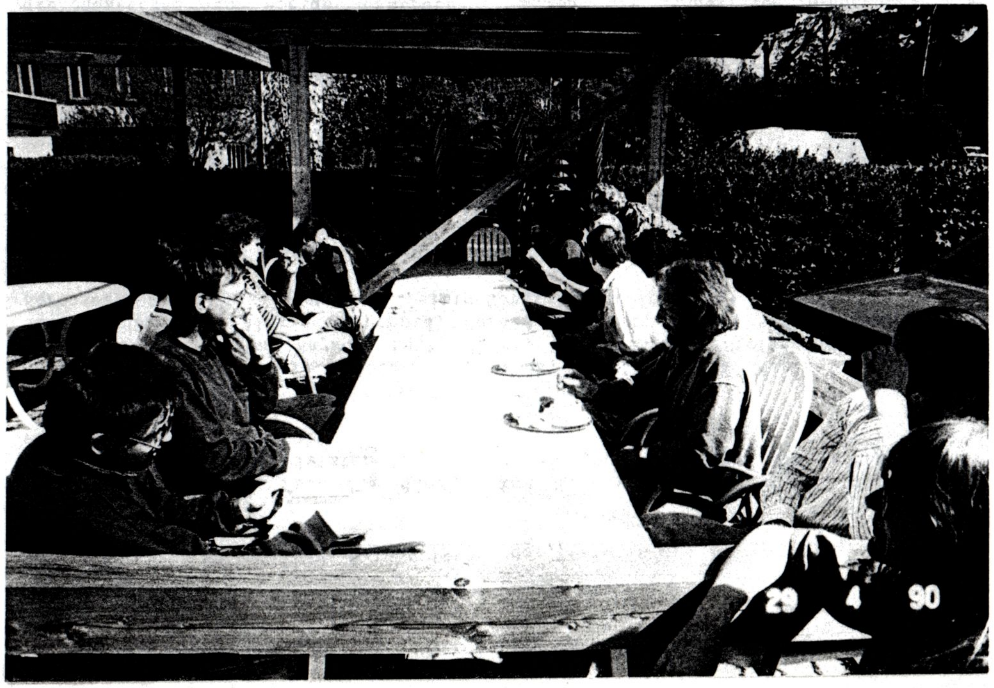
Ach ja, zwei Dauerbrenner habe ich noch vergessen:

1. Die Eintragung ins Vereinsregister wird endgültig nicht stattfinden.
   a) dürfte das inzwischen verjährt sein
   b) meint unser clubeigener Steuerprüfer, daß wir so nicht prüfungswürdig sind
   c) machen wir auch noch Gewinn (s. Kassenstand)
2. Die offizielle Einführung von MS-DOS in den Club wurde gar nicht erst diskutiert.

Durch wundersame Wendungen des Schicksals (bzw. die Weisungen unserer geliebten Hardthöhe) hat es den agilen Hartmut Obermann nach Nord-Italien (nach Günzburg) verschlagen, so daß wir jetzt guten Mutes auf das erste Südländertreffen warten. Im Norden wird das schon traditionelle Nordlichter- bzw. Fischköppe-Treffen (Nr. 4) diesmal von Kurt Müller organisiert. Vielen Dank an diese beiden Aktivisten!

Mehr fällt mir zum Clubtreffen nicht ein, außer, daß es wieder superaffengeil-abgefahren-sahnemäßig war. Wie jedes Clubtreffen eben.

Gerald



Clubtreffen in Spremberg/DDR

Spremberg? Nie zuvor davon gehört. Es liegt in der DDR. Seine Umgebung, die landschaftlich etwa an das Ruhrgebiet erinnert, erinnert auch optisch daran, wie jenes früher war: Ach nein, wir wollen in diesen Zeiten nur Nettes sagen. Um nicht lügen zu müssen, darf ich dann allerdings nur über die wenigen Spremberger berichten, die ich kennenlernte. Die Stadt (und nicht nur sie) hat das alte Regime nämlich geschafft.

Ringsum Tagebau. Im Süden die Stadt Schwarze Pumpe mit einem Braunkohle-Energiekombinat. Frank erzählt, daß man manchmal noch in Spremberg seine hellen Fußabdrücke im Ruß auf der Straße sieht.

Es gibt Oasen. Da wäre zum Beispiel Franks Wohnung (siehe Mitgliederliste). Mein Feinsliebchen und ich hatten in den drei Tagen, die wir uns dort aufhielten, immer das Gefühl, willkommen zu sein. Frank war vorbildlicher Gastgeber, Fremdenführer, auch Mitsäufer und ein amüsanter Gesellschafter. - Eigenartig; da erzählt einer, wie er als Angehöriger seines Bürgerkomitees die Stasi ausgenommen hat, und man amüsiert sich darüber, ignoriert seine eigene Gänsehaut. Psychische Notwehr?

Ach so, das Clubtreffen. Nein, tut mir leid, noch nicht, denn Computer sehe ich hier pausenlos. Was mir von dieser Reise in Erinnerung bleiben wird, sind nicht die flimmernden Bildschirme, das hektische Programmtauschen, die Fachsimpelei über Entwicklungen, von denen man bestenfalls gelesen hat (kommt uns das alles nicht sehr bekannt vor?). Es ist die unbeschreibliche Exotik der Situation, der Mief von Braunkohleöfen und Trabbis, Franks Mutter, die in 40 Jahren DDR nicht gealtert zu sein scheint, es ist ein Museumsdirektor-Ehepaar, das ebenso schlitzohrig wie mutig die Machthaber verarschte und damit den Bismarckturm (scheußlich-schöner, aber authentischer Historismus) wenigstens weitgehend rettete, es ist die Kellnerin, die uns das Essen am liebsten an den Kopf geschmissen hätte, weil Gäste nun mal die Ruhe stören, es ist aber auch die andere Kellnerin anderswo, die sich offenbar tatsächlich dafür interessierte, ob es uns geschmeckt hat. Es ist Spremberg, es ist die DDR.

Es war einmal ein Clubtreffen. Sollte ich gottbehüte jemals Enkel auf meinem Schoß sitzen haben, werde ich ihnen das ungefähr so erzählen. Wirklicher als ein Märchen kam es mir nämlich nicht vor: Elf oder zwölf Besitzer eines Computers. Sagt da gerade jemand, "Na und?" Freunde, wir sind in der DDR! Elf oder zwölf Besitzer eines Computers, die sich seit ein paar Jahren als Club verstehen. Normal? Wer einen Rechner besitzt, kann damit selbstredend nur beabsichtigen, das Regime zu stürzen. Wie, das wußte das Regime auch nicht so genau. Wenn aber im Prinzip schon die Steinzeittechnik des Küchenmessers dazu ausreicht, dann ist ab der Glühbirne äußerste Wachsamkeit geboten! Wenn sich ein paar real existierende Computeraner treffen, ist das eine Zusammenrottung. Wer diese kriminelle Vereinigung beherbergt, ist auch mit dran. Pardon: War. Elf oder zwölf Digitalpartisanen.

Kleiner Seitenhieb nach Westen: Computer, auch in dieser Größenordnung, stehen immer noch auf der Cocom-Liste. Man kann ja nie wissen.

Merkwürdig - nur zwei von ihnen sprechen sächsisch. Ist das nicht die Hochsprache der DDR? Ich sollte meine Vorurteile überdenken. Die anderen haben den örtlichen Lausitzer Tonfall, der sehr an Berlinerisch erinnert.

Die Kneipe ist nach hiesigem Sprachgebrauch das Vereinslokal der örtlichen Laubenpieper. Im lokalen Idiom hört sich das so an: "Spartengaststätte der Sparte 'Eigener Aufbau'". Ach so.

Auf den Tischen standen ungefähr zehn Meter Computer, hauptsächlich Sinclair ZX-Spectrum. Ein vermutlich IBM-kompatibler Laptop mit Plasma-Display schien irgendwie nicht dazuzugehören. Wer soeben mitleidig lächelte,

wird später möglicherweise vor Neid erblassen. Abenteuerliche Floppy-Controller, sämtlich Eigenbau, sämtlich ohne Gehäuse, standen daneben. Die Laufwerke wieder eine Kabellänge weiter. Die meisten haben ebenfalls kein Gehäuse. Diese Maschinen kriegen CP/M auf die Reihe. Mit unseren TRS-80 ist das schon schwieriger.

Computer, Controller, Floppy, Bildschirm. Vier Verbraucher. Franks Steckdosenleiste ist ein Holzbrett mit vier Aufputzdosen. "Der DDR-Bürger ist Weltmeister im Improvisieren," sagt er. Natürlich gibt es auch fertige Leisten, wie wir sie kennen. Zur Zeit vielleicht beispielsweise in Leipzig oder in Königs Wusterhausen.

Uwe hat seinen Schneider Joyce nicht mitgebracht. Wenn ein Computer den Transport nicht übersteht, ist erst mal Feierabend. HO hat keine Teile. Alles muß man sich umständlich und ziemlich langwierig "besorgen". Ein Wort, das man verdächtig oft hört.

Einer ist ziemlich sicher, daß eine Drahtbrücke umgepolt gehört. Der Wessi lötet aus und um und läßt es darauf ankommen. Der Ossi hat jetzt ein Problem. Natürlich gibt es Lötkolben. Aber was, wenn dabei ein Teil die weiße Fahne schwenkt? Der Wessi kauft ein neues, falls er es nicht schon auf Halde liegen hat. Der Ossi hat jetzt ein Problem.

Auf einem Tisch steht ein Fotokopierer. Nein, keine Pershing 2. Ein Fotokopierer. Sein Eigner hat ein Joint Venture mit einem Bundi gemacht. Zu beiderseitigem Nutz und Frommen (vermutlich, solange es dem Bundi nutzt und frommt). Karlheinz kopiert wie besengt. Schon mal beim R19 die Türen durch die Scheibe per Miniatur-Infrarot-Fernbedienung im Autoschlüssel auf- und zugeschlossen? Es muß für Karlheinz ungefähr dasselbe Feeling sein. Bislang waren Kopierer hier etwa so einfach zugänglich wie die Goldvorräte in Fort Knox.

Frank hat zwei Floppy-Laufwerke dabei. Wer einen Artikel schreibt, kriegt eins. Er nimmt seine Laufwerke wieder mit. Frank hat dasselbe Problem wie unser Jens-80. Schade, Jens zahlt nicht dieselben Honorare wie Frank.

Er hat auch einen Stapel elektronischer Bauteile aus ehemaligen Stasi-Abhöranlagen mitgebracht. Wer möchte, kann sich kostenlos bedienen. Auch hiervon bringt er das meiste wieder mit nach Hause. Hat man die Brauchbarkeit dieser Teile nicht erkannt? Oder fürchtet man das kalte Grausen, jedesmal, wenn man eins davon einlötet?

Zwischendurch gibt es Essen. Das Schnitzel ist gebraten. Meine sind meistens gekocht. Die Bundi-Bauern haben das unheimlich drauf, mit Hormonen oder Hortensien oder Wasweißich das Fleisch mit Wasser anzureichern, um Gewicht zu machen. Wasser zum Fleischpreis ist ein enormes Geschäft.

Das Clubtreffen hatte auch einen offiziellen Teil. Vereinsmeierei hier wie da. Wesentlicher Punkt der erfrischend kurzen Tagesordnung: Sollen wir dem Club-80 überhaupt, einzeln, en bloc oder nicht beitreten? Können wir ihm gem. Artikel 23 GG einfach mitteilen, daß Franks Mitgliedschaft jetzt für alle von Franks Club gilt? Was kostet das? Und für uns?

Und wenn sie keinen System Crash hatten, dann hacken sie noch heute.

Natürlich werden es mir meine Enkel nicht glauben, aber ihre ihnen auch nicht. Die Zeiten sind vorbei. Oder?

Arnulf

Nachbemerkung zu Spremberg

Das Clubtreffen fand am 21. 4. dieses Jahres statt. Wir haben heute den 5. 7. 90. Seither hat sich allerhand geändert, verbessert. "Ossi-Raten", hier in Lübeck, direkt an der Grenze, noch vor kurzem ein populärer Sport, ist schwierig geworden: Die verräterische blaß-zweifarbige Windjacke aus Dederon, die Prinz-Heinrich-Mütze (dort ausgerechnet "Schmidt-Mütze" genannt), der Trabbi und die fünf prallvollen Plastetüten sind immer seltener das ehemals unveränderliche Kennzeichen der Brüderundschwestern.

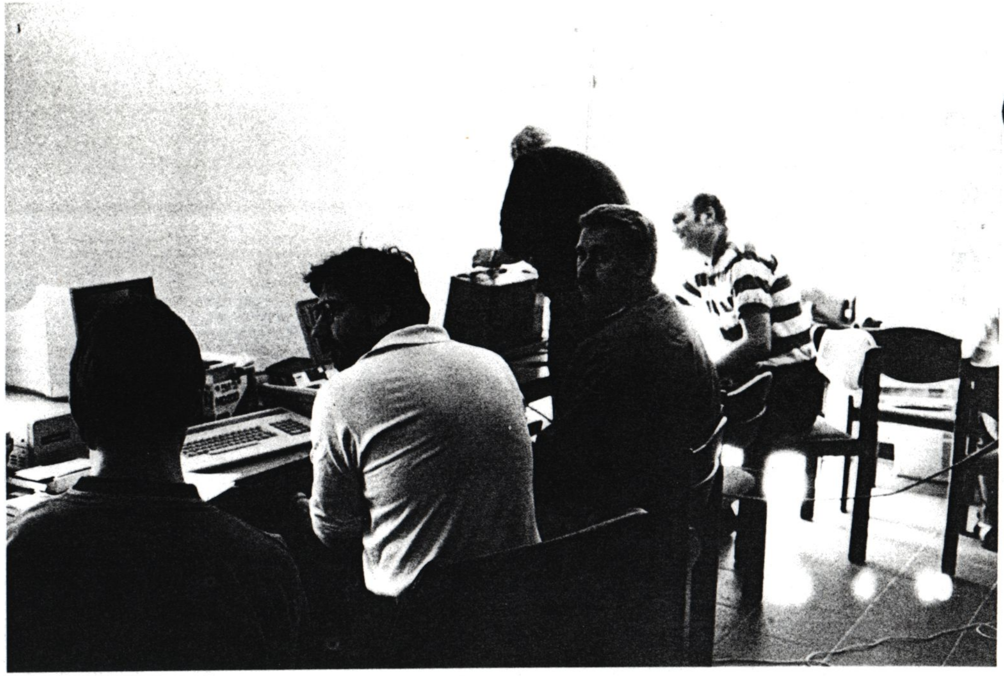
Auch auf dem technischen, natürlich auch dem digitalen Sektor hat sich eine Menge getan. Die DM hat offiziell Einzug gehalten, nachdem sie schon seit Monaten inoffizielles Zahlungsmittel für "Bückware", Schmierstoff für Handwerker und Eintrittskarte für die Intershops mit Westware war. (Dabei fällt mir ein, daß Schalck-Intershopski immer noch frei in der BRD herumläuft, denn er hat ja nicht falsch geparkt.)

Zur Sache: Wir dürfen uns auf die 15 neuen Clubmitglieder aus der DDR freuen. Nicht nur, daß viele von ihnen jetzt einen neuen, einen "richtigen" Computer kaufen werden, der besser in die Hardware-Landschaft des Club-80 paßt. Sie haben auch vier Jahrzehnte lang trainiert, aus nichts Erstaunliches zu zaubern. Auf ihre Bastel- und Programmieranleitungen, die wir demnächst im Info lesen werden, kann man gespannt sein.

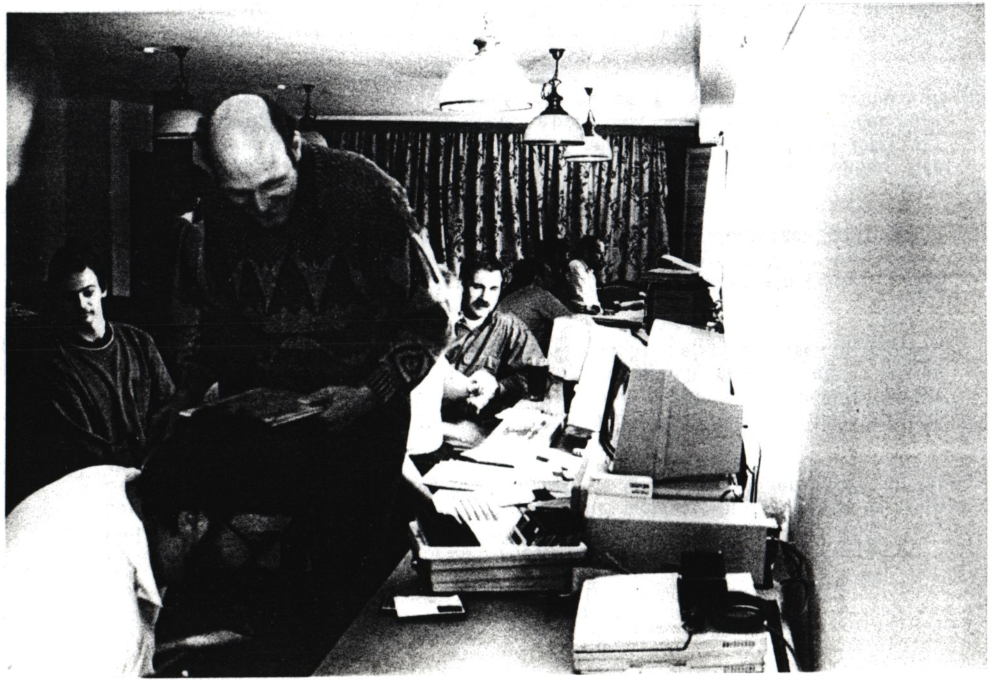
Auf unserer eigenen Jahreshauptversammlung haben wir beschlossen, daß die 15 Interessierten aus Franks Dunstkreis und jeder andere DDR-Bürger, der dem Club beitreten will, für ein Jahr beitragsfrei geführt wird.

Arnulf

Eine Fotoseite zum Spremberg-Treffen ist am Ende dieses INFO's beigelegt.



Unser Erster und wir.
Einige Eindrücke vom letzten Clubtreffen in Weilburg.



Turbo-Pascal und ZCPR

Turbo-Pascal auf der einen Seite ist neben dem Assembler in CP/M-Kreisen wohl die am weitesten verbreitete Programmiersprache und liefert eine ganze Menge mächtige Werkzeuge, die vom direkten Systemzugriff bis zu informatischen Höhenflügen ziemlich alles abdecken. Auf der anderen Seite steht das ZCPR-System, das mit seinen Suchpfaden, Shells und Error-Handlern usw. alles das bietet, was einem zum vollkommenen Glück noch fehlen kann. Leider hat Turbo keinen blassen Schimmer, in welch komfortabler Umgebung es laufen kann, aber dem kann abgeholfen werden.

Zuerst kurz ein paar Grundlagen. Ein ZCPR-Utility ist dadurch gekennzeichnet, daß der Code am Anfang folgendermaßen aussieht:

```
    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
100 C3 xx xx  Z  3  E  N  V 01 00 00 xx xx xx xx xx
```

Bei Adresse 100 steht ein Sprung, der in das eigentliche Programm führt und von 103-107 steht genau dieser Text als Erkennungsmerkmal. Das Byte bei 108 gibt an, um welche Art Programm es sich handelt, 01 bedeutet hier 'externes Utility', sprich gewöhnliches COM-File. Beim Laden des Programms schreibt der Lader in die Bytes 109 und 10A die Adresse des sog. Environments und da beginnt die Sache interessant zu werden. Wenn man nämlich diese Adresse kennt, sind alle anderen Adressen, wo z.B. der Suchpfad, die Prozessorgeschwindigkeit, die Steuerzeichen für das verwendete Terminal stehen, auch zugänglich.

Turbo-Pascal kommt einem in dieser Beziehung in bemerkenswerter Weise entgegen, hinter dem Sprung bei 100 kommt nämlich gleich eine Copyright-Notiz, die entsprechend überschrieben werden kann. Wem das nicht gefällt, der kann sich auch ein kleines Alias (eine Art SUBMIT) bauen, das den Code erst beim Laden patcht, sodaß das Copyright unverletzt bleibt:

```
GET 100 TURBO.COM                          <- lädt Turbo
POKE 103 5A 33 45 4E 56 01 00 00           <- macht Patches
GO                                         <- startet Turbo
```

Im Pascal-Programm selber kann man nun mit

```
VAR Z3EADR : Integer Absolute $109;
```

auf diese Adresse zugreifen und nach Lust und Laune mit den Möglichkeiten des ZCPR arbeiten. Um auf die einzelnen Werte einfacher zugreifen zu können, ist ein Record eine ganz brauchbare Lösung. Das sieht dann so aus:

```
TYPE ENVREC = Record
         env      : Byte;
         cbios    : Integer;
         z3id     : Array Ä1..5Ü of Char;
         envtyp   : Byte;
         expath   : Integer;
         expaths  : Byte;
            .
            .
            .

VAR ZENV : ^ENVREC;

ZENV := Ptr(Z3EADR);
```

Mit WriteLn('Länge des Pfads = ',envrec^.expaths); kann man sich z.B. bequem die Länge des Suchpfades ausgeben lassen. Damit kann man Turbo endlich beibringen, Files auf mehreren Laufwerken zu suchen. Natürlich kann man durch Schreiben auch gezielt Eingriffe ins System vornehmen, z.B. den Suchpfad so zu verändern, wie man ihn gerade braucht oder die diversen Flags bearbeiten.

Das Grundgerüst für die Zusammenarbeit von Turbo-Pascal mit ZCPR ist in einer Library namens NZ-TURBO.LBR zu finden, die auf zahlreichen PD-Disketten im Umlauf sein dürfte. Wenn nicht, Rüdiger hilft bestimmt gerne weiter (ich übrigens auch).

Hier noch ein kleines Demoprogrämmchen, wie man den Suchpfad abklappern kann.

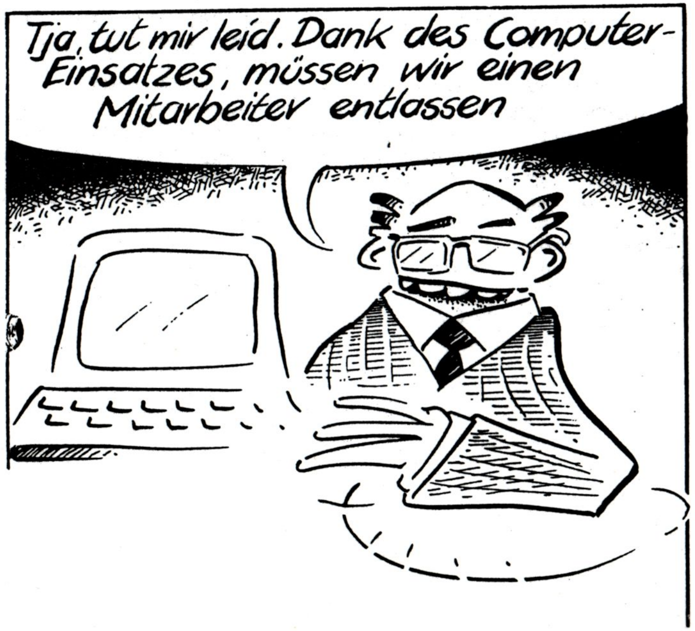
```
(* Testprogramm fuer den Zufgriff auf das ZCPR-Environment *)

(*$I NZ-TURBO.CON *)
(*$I NZ-TURBO.TYP *)   (* hier stehen die Records usw. drin *)
(*$I NZ-TURBO.VAR *)   (*  "      "     "    "     "    "  *)
(*$I HEX.FUN      *)

TYPE pointer=File;
     filename=StringÄ13Ü;

VAR aktlw  : Integer;
    suchlw: Char;
    laufw  : StringÄ17Ü;
    start  : Integer;
    p_elem: Integer;
    fileptr:pointer;
    fehler: Integer;

FUNCTION Oeffnen(name:filename;laufw:Char):Integer;
BEGIN
  (*$I-*)
  Assign(fileptr,suchlw+':'+name);
  Reset(fileptr);
  (*$I+*)
  Oeffnen:=IOResult;
END;

BEGIN
  envptr:=Ptr(z3eadr);

  Write('Path-Adr. : ',envptr^.expath,' = ');
```


HEFT 31
Juli
1990

```
WriteLn(HEX(envptr^.expath,1),'H');
WriteLn('Path-Len. : ',envptr^.expaths);

laufw:='ABCDEFGHIJKLMNOP';
aktlw:=Bdos($19)+1;
WriteLn('Aktuelles Laufwerk: ',LaufwÄaktlwÜ);

WriteLn('Suchpfad              : ');
start:=envptr^.expath-2;

p_elem:=aktlw;                (* zuerst aktuelles Lw durchsuchen *)
WHILE (p_elem<>0) DO          (* Ende des Suchpfades              *)
BEGIN
  IF (p_elem<>36) THEN        (* $ bedeutet auch akt. Lw im Pfad *)
  BEGIN
    suchlw:=laufwÄp_elemÜ;
    fehler:=Oeffnen('SYS.ENV',suchlw);
    Write('Laufwerk ',suchlw,':');
    IF fehler=0 THEN WriteLn('gefunden') ELSE WriteLn('nix');
  END;
  start:=start+2;
  p_elem:=MemÄstartÜ;
 END;
END.
```

Eine weitere Arbeitserleichterung ist der mögliche Zugriff auf die auf dem jeweiligen Terminal vorhandenen Steuerzeichen. Damit kann sich jedes Programm vollautomatisch selber installieren und die ewige Sucherei und Tipperei hat ein Ende.

Alexander Schmid



ZEUS und der HD64180 oder Z180/280

Ein paar Computer in unserem Club laufen inzwischen nicht mehr mit dem guten alten Z80, sondern einem seiner Nachfolger HD64180 oder Z180. Außerdem läuft gerade ein Selbstbauprojekt an, bei dem der Z280 das Herz bildet.

Diese neuen Mikroprozessoren haben einige Befehle, die der Z80 noch nicht kannte. Ihre Codes, die sog. Illegals, befolgte jener zwar auch, aber die neueren CPUs stellen damit etwas anderes an. Helmut hat das im letzten Info eingehend beschrieben und in einer Tabelle zusammengefaßt. Deshalb sei es hier nicht mehr erklärt.

Diese Befehle haben, ihrer anderen Funktion gemäß, neue Namen. Und die sind ZEUS leider unbekannt.

Bisher schrieb ich diese Befehle umständlich in der folgenden Form:

```
00029 ;          OTDMR
00030            DB           0edh,9bh
```

Der Befehl mit seinem richtigen Namen, den aber ZEUS nicht kennt, steht in einer Kommentarzeile. Die nächste Zeile enthält ausführbaren Code, der aber leider als DB (DEFB) eingegeben werden muß (dieser Zwei-Byte-Befehl hätte auch DW 9bedh lauten können).

Dessen leid, schrieb ich nun einfach alle neuen Befehle in eine kleine Library von Hilfslabels (s. Listing). Dort stehen sie als EQUates. Sie können in den ZEUS-Editor geladen werden. In der Source erscheinen nun die Befehle mit ihrem richtigen Namen, aber ohne sie mit dem Semikolon als Kommentare entschärfen zu müssen:

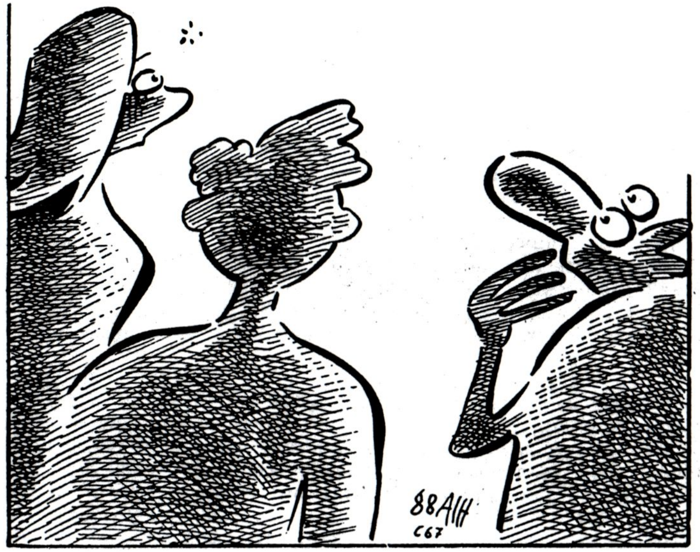
```
00029            DW           otdmr
```

Einige der neuen Befehle bestehen aus 3 Bytes. Da ist es mit einem DEFW leider nicht mehr getan. Das dritte Byte, in allen Fällen die Angabe eines I/O-Ports, muß dann doch noch in einer weiteren Zeile als DEFB-Statement nachgereicht werden (s. Listing). Dennoch wird die Eintipperei damit immer noch etwas vereinfacht gegenüber der alten Methode.

Natürlich ist das auch nur eine Krücke. Eleganter wäre es, den Befehl OTDMR in ZEUSens Befehlsvorrat einzubauen. Vor Jahren haben Gerald und ich das mal versucht. Es ist aber reichlich kompliziert, deshalb blieb die Arbeit irgendwann liegen. So geht es schließlich auch.

Arnulf

```
;Hilfslabels für Befehle des Z180 bzw. des HD64180, die ZEUS nicht kennt

;Inputs in Register g vom I/O-Port m in der I/O-Zero-Page
;(die Angabe des Ports muß als DEFB-Statement folgen)
in0_b   EQU     00edh
in0_c   EQU     08edh
in0_d   EQU     10edh
in0_e   EQU     18edh
in0_h   EQU     20edh
in0_l   EQU     28edh
in0_f   EQU     30edh
in0_a   EQU     38edh

;Outputs aus Register g auf den I/O-Port m in der I/O-Zero-Page
;(die Angabe des Ports muß als DEFB-Statement folgen)
```



```
out0_b  EQU     01edh
out0_c  EQU     09edh
out0_d  EQU     11edh
out0_e  EQU     19edh
out0_h  EQU     21edh
out0_l  EQU     29edh
out0_a  EQU     39edh

;(wiederholte) Block-Outputs mit Inkrement des Ports in der I/O-Zero-Page
otim    EQU     83edh
otimr   EQU     93edh
otdm    EQU     8bedh
otdmr   EQU     9bedh

;Multiplikationen (wwL * wwH = ww)
mlt_bc  EQU     4cedh
mlt_de  EQU     5cedh
mlt_hl  EQU     6cedh
mlt_sp  EQU     7cedh    ;SPL * SPH = SP

;non-destruktives AND
tst_b   EQU     04edh
tst_c   EQU     0cedh
tst_d   EQU     14edh
tst_e   EQU     1cedh
tst_h   EQU     24edh
tst_l   EQU     2cedh
tst_hl  EQU     34edh    ;A AND (HL)
tst_a   EQU     3cedh
tst_m   EQU     64edh
tstio   EQU     74edh    ;(C) AND m (I/O-Port m als DEFB-St. anzufügen)

;SLEEP-Befehl
slp     EQU     76edh
```

Programme relozieren

Häufig kommt es vor, daß ein Programm nicht gleich dorthin geladen werden kann, wo es laufen soll. In günstigen Fällen kommt man darin mit relativen Sprüngen aus, die an jeder Adresse ohne Änderung funktionieren. Denn sie erstrecken sich über eine Sprungdistanz, sie peilen kein unveränderliches Sprungziel an.

Gibt es jedoch JPs, CALLs oder Speicher-Ladebefehle, ist die Adresse leider ORG-abhängig. Wenn also eine Routine unter G-DOS an 5200h oder unter CP/M an 0100h geladen wird, dort aber gar nicht laufen soll, müssen diese Adressen reloziert, also der endgültigen Ladeadresse angepaßt werden.

ZEUS leistet für einfachere Fälle eine gute Hilfe: Da mit seinen Labels ein paar logische und mathematische Operationen erlaubt sind, kann beispielsweise ein Offset eingeführt werden:

```
offset      EQU      38f4h
            LD       A,(byte-offset)
            ...
byte        DB       00h
```

Das Byte "byte" steht nach dem Laden von Diskette meinetwegen an 5FD3h, soll aber später 38F4h Bytes tiefer liegen. Der Offset mit dem Label "offset" wird von ZEUS korrekt mit der ersten Ladeadresse von "byte" verrechnet.

Schwieriger ist es, wenn ein und dasselbe Programm an mehreren Stellen laufen soll, z. B. in mehreren Banks an verschiedenen Adressen. Genau dieser Fall trat jetzt bei mir auf, als ich eine TRAP-Routine für den Z180 sowohl in Bank 0 im unbenutzten BASIC-Bereich von G-DOS, als auch in Bank 1 ganz unten ab 0000h bereitlegen wollte.

Die kritischen Adressen habe ich zunächst mit den Labels addr1 – addr15 sowie dstlog versehen. Diese Adressen wiederholten sich dann in einer Relokationstabelle (s. Listing). Danach brach ich mir fürchterlich einen ab, um mit HL als Quellzeiger, DE als Zielzeiger, BC als Hilfszeiger, Helmut als Bergsteiger usw. die Adreß-Operanden der Befehle mit den neuen Werten zu beglücken. Es entstand ein umfängliches Programm, bei dem es mir einfach um den Speicherplatz leidtat. Das ging dann wiederum mir auf den Zeiger.

In der jetzigen Form ist die Routine erfrischend kurz: Zweimal wird der Stack verbogen. Zunächst wird er in die Tabelle der zu relozierenden Adressen gelegt. Mit POP gelangt die Adresse nach HL. Der Stackpointer hat sich dabei auf die nächste Adresse in der Tabelle erhöht. Er wird nun aus HL mit der Programmstelle geladen, an der eine Adresse reloziert werden soll. Was dort bereits steht, wird wieder mit POP erforscht. Die Subtraktion mit dem zweiten Offset (SBC HL,DE) ergibt den neuen Wert. Der wird mit PUSH gepatcht. Fertig.

Der ursprüngliche Stack und der zweite in der Relokationstabelle müssen natürlich gerettet werden, um den Super-Soft-GAU zu vermeiden. Leider existiert dafür nur ein Speicherladebefehl, was etwas unelegant ist (LD (nn),SP) und ein paar Nanosekunden länger dauert als Registerladebefehle. Da es in meinem Programm aber nur 16mal vorkommt, kann man damit gut leben.

Die TRAP-Routine ist die von Gerald. Da sie hier nicht (sonst aber sehr) von Interesse ist, habe ich im Listing fast alles davon mit LIST OFF ausgespart. Nur ein wenig Code mit den Labels addr... ist übrig. Das erkennt man an den springenden Zeilennummern. Auch in meinem eigenen Programm ist an dieser Stelle vieles nicht von Belang. Der gelistete Teil stellt lediglich die Relokation vor.

Arnulf


```
5200            00001          ORG     5200h
                00002
1650            00037 target   EQU     1650h             ;im BASIC-Bereich (vorläufig)
3BCF            00038 offs     EQU     $-target          ;Abstand zur Arbeitsadresse
526D  32B016    00097 gotrap   LD      (const-offs),A    ;evtl. Konstante in den Programmtext
526E            00098 addr5    EQU     $-2
5270  78        00099          LD      A,B               ;DD/FD zurück
5271  32AC16    00100          LD      (ddfd1-offs),A    ;in den Programmtext setzen
5272            00101 addr6    EQU     $-2
5274  32B216    00102          LD      (ddfd2-offs),A    ;dort auch
5275            00103 addr7    EQU     $-2
530B  115016    00204          LD      DE,target         ;Offset der Adressen von target zu 0000
530E  0610      00205          LD      B,endtab-addrtab/2     ;Anzahl zu relozierender Adressen
5310  ED732753  00206          LD      (spbuff2),SP      ;Stack retten
5314  315253    00207          LD      SP,addrtab        ;Stack auf Adressentabelle setzen
5317  E1        00208 addrlop  POP     HL                ;eine Adresse abholen
5318  ED732253  00209          LD      (spbuff1),SP      ;Tabelle als Stack retten
531C  F9        00210          LD      SP,HL             ;dortige Adresse = Stack
531D  E1        00211          POP     HL                ;dortiges Word abholen
531E  ED52      00212          SBC     HL,DE             ;auf Basisadresse 0000 umrechnen
5320  E5        00213          PUSH    HL                ;neues Word ablegen
5321  310000    00214          LD      SP,0000h          ;Stack wieder in der Tabelle
5322            00215 spbuff1  EQU     $-2
5324  10F1      00216          DJNZ    addrlop           ;bis alle Adressen bearbeitet sind
5326  310000    00217          LD      SP,0000h          ;alten Stack restaurieren
5327            00218 spbuff2  EQU     $-2
                00256
                00257 ;Tabelle der zu relozierenden Adressen in der TRAP-Routine
5352  2B52      00258 addrtab  DW      addr1,addr2,addr3,addr4,addr5,addr6,addr7,addr8
5362  8952      00259          DW      addr9,addr10,addr11,addr12,addr13,addr14,addr15,dstlog
                00260 endtab

00000 Fehler
```

Harddisk-Einbindung ins CP/M3-BIOS

hier am Beispiel des OMTI 5527 RLL Controllers und einer Platte mit 615 Zylindern und 4 Köpfen (32MB im RLL-Format)

Helmut Bernhardt, Peter Petersen

Die Hardware

Der Aufwand in Hardware, um einen PC-Harddisk-Controller wie die verschiedenen OMTI-Typen an den ECB-Bus oder auch an andere Z80-Bus-Systeme anzuschließen, ist minimal. Beim ECB-Bus bestehen die Probleme hauptsächlich in der ungeeigneten Geometrie des Controllers, der die Breite der Europakarten von 10cm überschreitet. Die Montage auf eine ECB-Bus-Trägerkarte, die auch die beiden zur Busanpassung nötigen ICs aufnimmt, muß für jedes Gerät individuell gelöst werden.

Die Busanpassung beschränkt sich auf die Umverlegung der ECB-Adressen an die Slot-Adressen des Controllers, damit bei einem Zugriff der CPU unter einer für den Controller vorgesehenen Portadresse am Controller die Portadresse 320H(-323H), unter der der Controller im PC decodiert ist, anliegt. Außerdem müssen aus den ECB-Signalen /RD, /WR und /IORQ die PC-Bus-Signale /IOR und /IOW mit zwei OR-Gattern eines 74LS32 erzeugt werden. Auch das low aktive /RESET des ECB-Bus muß über einen 74LS04-Inverter in das high aktive RESET des PC-Bus umgesetzt werden.

Da 4 aufeinanderfolgende Portadressen benötigt werden, müssen A0 und A1 direkt an den Controller gelangen. Für eine Adreßlage bei D0H-D3H am ECB-Bus ist in Abb.1 eine Beschaltung gezeigt, die folgende Pegelübersetzung bewirkt:

Adresse	:	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ECB Port D0H	:	-	-	1	1	0	1	0	0	x	x
PC Port 320H	:	1	1	0	0	1	0	0	0	x	x

Außerdem ist an AEN und A10-A19 des Controllers GND und an /DACK3 und /MEMR +5V zu legen. Außer dem OMTI 5510 benötigen alle OMTI-Controller am Slot-Pin b9 +12V Versorgungsspannung. Selbstverständlich müssen die Daten D0-D7 an die gleichen Eingänge des Controllers gelegt werden.

Die Software

Wesentlich anspruchsvoller ist die Software-Einbindung der Festplatte in das BIOS des CP/M plus.

Mit den unter CP/M bestehenden Möglichkeiten zur Verwaltung der Files auf einer Platte (Unterteilung in User 0 bis User 15) lassen sich nur begrenzte Mengen an Files übersichtlich handhaben. Deshalb ist es bei größeren Platten sinnvoll, diese in mehrere logische Laufwerke zu partitionieren. Noch handhabbar sind dabei Größen von einigen Megabyte Kapazität. Danach könnte eine 32MB-Platte in 4 Partitionen aufgeteilt werden. Die sich anbietende Aufteilung, daß alle Spuren unter einem Kopf als ein Laufwerk behandelt werden, bedeutet, daß nach je 26 Sektoren der Kopf eine Spur weiterbewegt werden muß. Der Spurwechsel ist bei Festplattenzugriffen aber die zeitraubendste Aktion und sollte so wenig wie möglich vorkommen.

Deshalb wurde die Partitionierung hier auch anders durchgeführt: Eine logische Spur umfaßt alle 4 Spuren (Köpfe) eines Zylinders. Es werden nacheinander erst die 4 Spuren unter allen 4 Köpfen gelesen/geschrieben, bevor der Kopfschlitten um eine Position weiterbewegt wird. Dadurch können nacheinander 104 Sektoren verholt werden, ohne zwischendurch Steppen zu müssen. Einer Partition wird dabei ein zusammenhängender Bereich von Zylindern zugeordnet; die Nummer des letzten absoluten Zylinders einer Partition ist die Anzahl Spuren und die Nummer des letzten absoluten Zylinders der vorherigen Partition ist die Anzahl reservierter Systemspuren.

```
Spur-Nr.    0 1        154         307         460         615
Partit. 0   ============-----------------------------------------
Partit. 1   ------------============-----------------------------
Partit. 2   ------------------------============-----------------
Partit. 3   ------------------------------------=================

     -----   reservierte Systemspuren
     =====   Datenspuren der Partition
```

Das BDOS kennt für Disk-I/O nur die Adresse einer Tabelle mit dem Namem @dtbl (drive table). In dieser Tabelle stehen die Adressen von 16 Disk Parameter Headern (DPHs) für 16 Laufwerke in der Reihenfolge ihrer logischen Namen (A, B, ...P). Nicht existierende Laufwerke erhalten hier den Eintrag 0. Wenn in einem Sourcecode-Modul nur die Drive Table und die DPHs und DPBs für RAM-Floppy und Harddisk enthalten sind, könnte das so aussehen:

```
        public  @dtbl                   ; Drive Table
        public  fdsd0,fdsd1,fdsd2,fdsd3 ; dph's fuer 4 Floppies
        public  fdram                   ; dph RAM-Floppy
        public  hdsk0,hdsk1,hdsk2,hdsk3 ; dph's fuer harddisk 0 - 3

        extrn   hdlogi,hdinit,hd1init    ; Harddisk Routinen
        extrn   hdwrit,hdread

        maclib  xcpm3           ; mit hdpb-Macro fuer grosse Drives

        cseg                    ;Drivetable und dpb's im Common

;Drive Table (zeigt auf die DPBs der verfuegbaren Laufwerke)

@dtbl   dw      fdsd0,fdsd1,fdsd2,fdsd3 ; Floppy Drives
        dw      fdram                   ; RAM-Floppy
        dw      hdsk0,hdsk1,hdsk2,hdsk3 ; Harddisk
        dw      0,0,0,0,0,0,0           ; nicht benutzt

;Die Einträge in @dtbl weisen auf Adressen der DPHs, wobei die Lables
;nicht am Anfang der xDPHs stehen. Der Aufbau der xDPHs sieht so aus:

        dseg            ; die DPHs koennen in gebanktem RAM stehen

; extended disk parameter header (xDPH) der Harddisk-Partitionen

drva:   dw      hdwrit          ; Routine zum Schreiben
        dw      hdread          ;   und Lesen eines Sektors
        dw      hdlogi          ; Einloggen des Drives
        dw      hdinit          ; Initialisierung des HDC
        db      0               ; relative drive 0
        db      0               ; media flag     : not used
hdsk0   dph     0,hddpb0        ; Skew-Tabelle und xDPB

drvb:   dw      hdwrit,hdread,hdlogi
        dw      hd1init         ; kann auf RET zeigen
        db      1               ; relative drive 1
        db      0               ; media flag     : not used
hdsk1   dph     0,hddpb1

drvc:   dw      hdwrit,hdread,hdlogi
        dw      hd1init         ; kann auf RET zeigen
```


HEFT 31 Juli 1990

```
        db      2               ; relative drive 2
        db      0               ; media flag     : not used
hdsk2   dph     0,hddpb2

drvd:   dw      hdwrit,hdread,hdlogi
        dw      hdlinit         ; kann auf RET zeigen
        db      3               ; relative drive 3
        db      0               ; media flag     : not used
hdsk3   dph     0,hddpb3
```

'dph' ist ein Macro-Aufruf, der in CPM3.LIB aufgelöst wird. Am Anfang des Moduls, in dem der xDPH steht, ist ' maclib cpm3' anzugeben, damit RMAC den Macro bearbeiten kann (aus später benannten Gründen sollte lieber XCPM3.LIB benutzt werden).

Da der Skew der Festplatte bereits beim Formatieren durch physikalischen Versatz der Sektoren auf der Spur vorgegeben wird, ist ein Soft-Skew nicht nötig; anstelle der Adresse einer Skew-Tabelle kann als erster Parameter für dph '0' angegeben werden.

Die Einträge hddpb0 bis hddpb1 geben die Adressen der xDPB's der vier Harddisk-Partitionen an. Diese können wieder mit einem Macro erzeugt werden:

```
        cseg            ; die DPHs müssen im Common liegen

; hard disk parameter block macros

hddpb0  hdpb    512,104,154,4096,2048,1,8000h   ; partition 0
hddpb1  hdpb    512,104,307,4096,2048,154,8000h ; partition 1
hddpb2  hdpb    512,104,460,4096,2048,307,8000h ; partition 2
hddpb3  hdpb    512,104,615,4096,2048,460,8000h ; partition 3

;                !   !   !   !    !    !    !
;                !   !   !   !    !    !    ---- flag für nonre-
;                !   !   !   !    !    !         movable media
;                !   !   !   !    !    ---- reserv. Systemspuren
;                !   !   !   !    ---- max. Anzahl DIR-Einträge
;                !   !   !   ---- Blockgröße (die durch einen
;                !   !   !        DIR-Eintrag verwaltet wird)
;                !   !   ---- Anzahl Zylinder des Laufwerks
;                !   ---- Sektoren pro Zylinder (2 tracks mit 26 secs)
;                ---- Bytes pro Sektor
```

Aufgrund des optionalen Eintrags 8000h wird das Laufwerk vom BDOS als 'nonremovalble media' angesehen, woraufhin nicht bei jedem Zugriff ein neues Einloggen durchgeführt wird.

'hdpb' ist (ähnlich dem dpb in CPM3.LIB) ein in XCPM3.LIB zusätzlich vorhandener Macro, der auch für Laufwerke mit einer Kapazität von mehr als 8 MB einen richtigen Disk Paramter Block erzeugt.

Das Source-Modul (angenommen, es heißt MOVE.ASM), das diese Datenstrukturen enthält, wird mit 'rmac move $pz sz' assembliert (in einem SUBMIT File sind 2 $-Zeichen einzutragen). M80 kann die Macrobibliotheken CPM3 und XCPM3 nicht benutzen. Da für die Datenstrukturen aber keinerlei Maschinenbefehle nötig sind, braucht man sich auch nicht unbedingt mit den von RMAC einzig verstandenen 8080-Mnemonics herumzuschlagen. Die eigentlichen Disk-I/O-Routinen können in einem anderen Modul formuliert werden, wobei man sich die Fähigkeit von M80 zunutzemachen kann, Z80-Mnemonics zu vestehen.

Im Kopf des Moduls mit diesen Datenstrukturen sind sämtliche Lables wie @dtbl, auf die aus anderen Modulen zugegriffen wird, mit 'public' bekanntzugeben, und alle Einträge in diesen Tabellen, die nicht in diesem Modul selbst als Lable auftauchen, mit 'extrn' als außerhalb stehend zu benennen.

Bei ausreichend bereitgestelltem RAM-Speicher legt GENCPM Directory-Buffer in Bank 0 (Systembank) und Data-Buffer in anderen dafür angegebenen Banks an (siehe unten). Die Routinen 'hdwrit' und 'hdread' müssen also den Datentransfer zwischen Controller und verschiedenen Banks unterstützen. Beim Aufruf bekommen sie in den in BIOSKRNL zugewiesenen Variablen entsprechende Parameter übergeben:

@dbnk	(8Bit)	Source-/Ziel-Bank für Holen/Ablegen eines Sektors beim Schreiben/Lesen
@dma	(16Bit)	RAM-Adresse in der Bank, von wo ab mit aufsteigender Adresse die Daten des Sektors zu holen/abzulegen sind
@adrv	(8Bit)	absolute Laufwerks-Nummer (entsprechend Position in @dtbl)
@rdrv	(8Bit)	relative Laufwerks-Nummer (Zählung der Laufwerke für einen Controller)
@trk	(16Bit)	Spur des Laufwerks
@sect	(16Bit)	Sektor der Spur

Diese Daten stehen alle in der Systembank 0 und müssen ausgewertet werden, bevor die in @dbnk vorgegebene Bank eingestellt wird und diese Daten dann nicht mehr zur Verfügung stehen.

Die Übertragung von Daten erfolgt direkt vom/zum Controller zur/von der Buffer-Bank. Da diese Bank dann eingeschaltet sein muß, kann die Übertragung nur von einer Routine im Common durchgeführt werden. Das hat den Vorteil, daß der Common-Bereich keinen 512Byte-Zwischnspeicher enthalten muß, was auf Kosten der TPA geht, und daß nur ein Datentransfer nötig ist, was Geschwindigkeit bringt.

Befehle an den OMTI-Controller bestehen aus einer Folge von 6 Bytes, die nacheinander an die relative Portadresse 0 des Controllers zu übergeben sind. Das erste Byte ist das eigentliche Befehlsbyte und die folgenden 5 Bytes sind Paramter des Befehls. Diese Bytefolge wird zunächst im RAM erzeugt und dann an den Controller ausgegeben. Das Parameter-Feld im RAM sieht so aus:

```
cfield:                 ;Befehlsparameter-Feld für HDC
head:   db      0       ; D7 = Bit 10 der Zylinder-Nummer
                        ; D6 = 0, D5 = Nr. des zu bedienenden Lauf-
                        ; werks (0 oder 1, an den OMTI lassen sich
                        ; 2 Festplatten anschließen) nicht mit der
                        ; Partition eines Laufwerks verwechseln !
                        ; D4-D0 : Head-Nummer
sect:   db      0       ; D6,D7 : Bit 9 und 8 der Zylinder-Nummer
                        ; D5-D0 : Sector-Nummer
cylind: db      0       ; D7-D0 : Bit 7 - 0 der Zylinder-Nummer
bcount: db      1       ; Interleave-Count beim Formatieren
                        ; Block-Count bei MultIO
termin: db      2       ; D7-D5 : Command-Controll-Byte
                        ; D4-D0 : nicht benutzt
```

Für die Beschreibung des 'termin'-Bytes sei auf das Handbuch der OMTI-Controller verwiesen (das wollen wir uns hier sparen).

Vor der Übertragung dieser Parameter ist der eigentliche Befehl direkt an den Controller auszugeben. Der Inhalt dieses Feldes wird in der Routine hdrw_common berechnet und in die Tabelle eingetragen. 'hdscmd' übergibt die Bytes an den Controller.

Die zwingend im Common anzusiedelnden Routinen 'hdrd' und 'hdwr' so-

wie der von beiden genutzte Ausgang 'termcmd1', der wieder die Bank 0 einschaltet, machen zusammen knapp 60 Bytes des kostbaren Common-Speichers aus, dem 512 Bytes eines Zwischenspeichers bei der anderen Variante gegenüberstehen.

Bei einem Prof mit 9,2MHz Systemtakt ist damit ein Interleave 1:1 möglich. Wegen insgesamt bereitgestellter Data-Buffer von 64K RAM erscheint ein erster Zugriff auf die Festplatte recht langsam, weil dann immer gleich riesige Datenmengen bewegt werden. Nachfolgende Zugriffe können dann aber aus dem Buffer leben und erreichen dann die Geschwindigkeit einer RAM-Floppy.

Weiteres Tuneup am BIOS

Da die Speicheraufteilung des Prof 180 recht ungünstig ist und dadurch bedingt auch die RAM-Floppy-Routinen umständlich mit '?xmove' über einen 128 Byte Zwischenspeicher im common arbeiten, wurde auch die ganze Speicherkonfiguration nach Conitec's Vorstellung aufgegeben und durch eine sinnvollere ersetzt.

Speicherbereich	Adreßlage nach Conitec	neue Adreßlage
Systembank	00000h-0efffh	00000h-0efffh
Common	0f000h-0ffffh	0f000h-0ffffh
TPA	40000h-4efffh	10000h-1efffh
CCP-Buffer	4f000h-4ffffh	1f000h-1ffffh
weitere Banks	----------	20000h-2ffffh
RAM-Floppy	10000h-3ffffh + 50000h-7ffffh	30000h-7ffffh

Die ursprüngliche Einteilung von Conitec ermöglicht den Einsatz von 4164-RAMs, wobei die Selektion der beiden 64K-Blöcke durch den Pegel von A18 erfolgt (was bei 41256-RAMs auch sinnvoll ist). Entsprechend liegen diese beiden Blöcke auf den physikalischen Adressen 0xxxxh und 4xxxxh. Da bei den heutigen RAM-Preisen der Einsatz von 4164-RAMs unwahrscheinlich ist und hier 41256-RAMs stecken, wurde auf die Möglichkeit des Abrüstens auf 128K verzichtet, so daß der Bereich, der für Banks und/oder RAM-Floppy zur Verfügung steht, zusammenhängt. Das vereinfacht die RAM-Floppy-Routinen gewaltig.

Der Zugriff auf die RAM-Floppy erfolgt nun im DMA-Betrieb und zur Vereinfachung (und Beschleunigung) wurde die Sektorgröße von 128 auf 1024 Bytes erhöht und die Größe einer Spur auf 64K festgelegt. Die Track-Nr kann dann direkt (durch Addition des Offsets des RAM-Floppy-Anfangs) in das Adreß-Zusatzbyte für den DMAC umgesetzt werden und aus der Sektor-Nummer kann durch Shiften um 2 Bits der Adreßanteil A15-A10 erhalten werden; A9-A0 sind immer 0 für die Anfangsadresse eines Sektors.

Die RAM-Floppy-Routinen sind in RFLO.MAC versammelt. DPH und DPB der RAM-Floppy sowie die Routinen und Tabellen für Bankumschaltung und DMA sind neben den DPHs und DPBs für die Festplatte in MOVE.ASM enthalten.

Durch die neue Speichereinteilung muß GENCPM bei der Erzeugung der Memory Segment Table auch mit anderen Parametern versorgt werden, als den in GENCPM.DAT bisher enthaltenen. Bei 64K Data-Buffer sind insgesamt 3 Segmente anzugeben (Common und TPA sowie CCP-Buffer werden hier nicht mitgezählt). Für jedes Segment sind dann je 3 Bytes anzugeben, die folgende Bedeutung haben:

1) high Byte der logischen Anfangsadresse der Bank
2) high Byte der Länge der Bank
3) Bank-Nummer

Eine Angabe '00,80,02' bedeutet, daß die Bank Nr 2 von 0000h bis 7fffh

reicht (logische Adressen). Wo diese Bank physikalisch im Speicher liegt, spielt hier keine Rolle.

Für die Bank 0 sind diese Angaben (speziell beim Prof 180) etwas komplizierter. Der Prof 180 benutzt 4KByte des Boot-ROMs (bzw. des auf gleiche Adressen ins RAM kopierten ROM-Inhalts) vom BIOS des CP/M aus. Deshalb muß für die Systembank als high Byte der Anfangsadresse 10 angegeben werden, damit diese ROM-Routinen nicht überschrieben werden. Als Länge ist die Differenz des von GENCPM vorher ausgegebenen Anfangs des gebankten BDOS abzüglich dieser Adresse anzugeben. Die Bank Nr ist natürlich 00.

Um unseren CP/M-Diskothekar nicht mit Format-Konvertier-Problemen zu überlasten, bitte ich darum, bei Interesse an den BIOS-Sources für die Festplatte und die neue Variante der RAM-Floppy mir eine formatierte Diskette mit den gerade verfügbaren Angaben zum Format und möglichst einigen Text-Files darauf (damit ich prüfen kann, ob ich das Format richtig eingestellt habe) zu schicken. Beiliegendes Rückporto wird auch gerne gesehen.
Helmut

Zum Thema Harddisk unter CP/M geben folgende Artikel etwas her:

Andreas Zippel 'Hart, schnell und sicher', c't 8/86, S.52ff; c't 9/86, S.105ff; c't 10/86, S.144ff

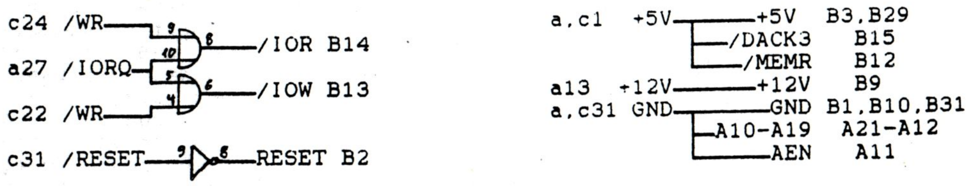
Martin Rost, Detlef Grell 'Harddisk-Controller einmal anders', c't 4/87, S.138ff; c't 9/87, S.138ff

Christian Persson 'OMTI-Alternative', c't 5/89, S.244

Anschluß des OMTI-Controllers 5527 an den ECB-Bus

ECB-Bus Pin-Nr	ECB-Bus Signal	PC-Slot Signal	PC-Slot Pin-Nr
c9	A7	A9	A22
a9	A6	A8	A23
a8	A5	A7,A6	A24,A25
a7	A4	A5	A26
c6	A3	A4	A27
a6	A2	A3,A2	A28,A29
c7	A1	A1	A30
c5	A0	A0	A31

ECB-Bus Pin-Nr	ECB-Bus Signal	PC-Slot Signal	PC-Slot Pin-Nr
c2	D0	D0	A9
c14	D1	D1	A8
c4	D2	D2	A7
a4	D3	D3	A6
a5	D4	D4	A5
a2	D5	D5	A4
a3	D6	D6	A3
c3	D7	D7	A2



HEFT 31 Juli 1990

Die Clubeigene Grafikkarte

Nachdem im vorletzten INFO mein Artikel erschienen ist, habe ich mit einer Flut von Zuschriften gerechnet. Dem war leider nicht so. Dem konnte ich entnehmen, daß Ihr alle sehr schreibfaul seid, oder kein Interesse an Projekten habt. Ich will hoffen, daß es der erste Grund ist. Aber immerhin habe ich ja zwei Zuschriften bekommen. Aber vieleicht waren euch ja die Angaben auch zu dürftig. Jetzt sieht es anders aus. Auf dem Clubtreffen habe ich mich mit Helmut über eine Grafikkarte unterhalten. Wir sind zu dem Entschluß gekommen es so einfach wie möglich zu machen und vor allem das Rad nicht neu zu erfinden. Man kann nicht wiederlegen, das auch IBM seine Vorteile hat, auch wenn es nur der ist, dafür gesorgt zu haben 'billige' Grafikkarten auf den Markt zu schmeißen. Und diesen Vorteil wollen wir nutzen.

Aber nun im einzelnen. Wir haben beschlossen einen Grafikadapter zu entwickeln. Dieser hat die Aufgabe eine Herkules- oder EGA-Karte über Port's an jeden Rechner anzuschließen. Der Adapter wird 'Intelligent', d.h. er bekommt eine eigene CPU. Alles was zur Ansteuerung der Grafikkarte notwendig ist, erledigt ein Z-80, genauso wie die Kommunikation mit dem HOST-Rechner.

Die Kommunikation mit der Karte erfolgt über zwei Port's, einer lesend, einer schreibend, die mit ein paar Bauteilen leicht in jeden Rechner einzubauen sind. Über diese Port's werden Daen und Befehle ausgetauscht. Die Adapterkarte enthält eine CPU, 32Kb RAM, 32Kb ROM sowie die Anschaltung der Grafikkarte. Es wird also nicht zu viel. Für ganz eilige: Die genaue Schaltung ist noch nicht fertig, weil Helmut und ich noch ein genaues Konzept ausarbeiten müssen, aber das ist jetzt auch noch nicht ganz so schlimm, weil noch ein paar andere Dinge festzulegen sind, die aber die Software angehen. Wer aber trotzdem mit Rat und Tat zur Seite stehen will, wende sich bitte an Helmut oder mich, auch was die Zoffware angeht.

Aber nun zu der Karte selber. Wir haben uns zwar schon ein paar Gedanken gemacht was das Teil können muß, aber wie schon gesagt, es ist noch in der Entwicklung, und außerdem sollt Ihr eure Meinung auch dazu beitragen. Die Liste erhebt keinen Anspruch auf Vollständigkeit, sondern ist ein erster Überblick. Wenn Ihr eigene Ideen habt, laßt sie mich wissen, auch wenn Ihr nicht wißt wie sie zu bewerkstelligen sind, denn darüber machen wir uns dann später Gedanken.

- Programm Dounload (Eigene Programme die die Karte erledigt)
- Punkt setzen
- " löschen
- " testen
- Linie ziehen mit Farbangabe
- Viereck zeichnen mit Farbangabe
- Fläche ausfüllen mit Farbangabe
- Kreis zeichnen
- Schirm invertieren

- Bereich invertieren
- Fläche inverieren
- Bereichsfarbe ändern
- Flächenfarbe ändern
- Bereich spiegeln
- Bereih drehen
- Bereich verschieben
- Objekt verschieben
- ASCCI anzeigen
- Zeichensatz laden
- Terminalfunktion
- Letzten Befehl wiederholen
- Letzten Befehl zurücknehmen
- HRG auslesen (zum abspeichern)
- HRG einschreiben (von Disk)
- HRG mischen (MERGE)
- beliebigen CURSOR erzeugen
- CURSOR bewegen (plazieren)
- ....

Das ist das, was mir auf die schnelle so einfiel. Ich will es erstmal dabei belassen, um mir über den genauen Aufbau Gedanken zu machen. Vieleicht gibt es im nächsten INFO schon Schaltungen oder Programme dazu. Aber wie gesagt, Eure Mitarbeit und Eure Anregungen sind ausdücklich erwünscht.

Bis bald, Euer Hardware-Koordinator.

Club 80 Projekt:

Bau einer Terminal-Karte auf Basis der PC-Hercules-Karte

Der GENIE 1 mit Hercules-Karte als Video-Interface ist ja ganz witzig aber für die meisten Club-Mitglieder uninteressant, weil die Strippenzieherei einfach zu aufwendig ist und die Festlegung auf die GENIE--Kompatibilität zu beengend ist.

Um das günstige Preis/Leistungs-Verhältnis der Hercules allgemeiner zugänglich zu machen, wurde angeregt, als Projekt im Club 80 eine Terminal-Karte zu bauen, die zusammen mit der Hercules in jedem Computer als Konsole eingesetzt werden kann. Das Terminal soll mit Z80-Power und entsprechender Betriebssoftware fast ein eigenständiger Text-/Grafik-Computer sein, der in Arbeitsteilung mit der CPU Video-Ausgaben erledigt.

Da eine EGA- oder VGA-Karte ähnlich wie die Hercules-Karte funktioniert (erweiterter I/O-Adreßbereich, anderer Memory-Adreßbereich), ist das Prinzip auch auf diese Karten anwendbar. Es bestehen dort aber bei den über die IBM-Vorgaben hinausgehenden Features unterschiedliche Lösungen, die nicht ausreichend dokumentiert sind. Außerdem benötigen diese Karten Monitore, deren Preise weit über den Aktionsradius des Bastlers hinausgehen, so daß wir uns zunächst auf die Hercules-Karte beschränken sollten. Wenn wir das im Griff haben, können wir immer noch überlegen, ob wir nach den Sternen greifen wollen.

Von Andreas Magnus habe ich als letzte Meldung vor längerer Zeit erfahren, daß er auch schon einen entsprechenden Artikel ins Info gesetzt hat; ich weiß allerdings nicht, was er dort schon angekündigt hat. Deshalb möge man mir nachsehen, wenn einige Aussagen hier zum zweiten mal erscheinen. Da inzwischen schon etwas festere Vorstellungen bestehen, ist dieser Beitrag aber unumgänglich.

Die Kommunikation der Terminal-Karte mit der CPU soll über einen bidirektionalen Parallelport (Datenaustausch) und einen Statusport (Handshaking) erfolgen. Zwischen CPU und Terminal werden Befehle der CPU an das Terminal vereinbart, die das Terminal zur selbständigen Bearbeitung der Hercules anregen. Die bisherige Vorstellung sieht dabei im Textmodus die Emulation eines Televideo 950 Terminals und im Grafikmodus die eines Tektronix 4014 Grafikterminals vor.

Während für die Hardware des Terminals schon feste Vorstellungen bestehen, ist für die Betriebssoftware noch alles offen. Zur Erstellung eines Betriebsprogramms für das Terminal sind alle Club-Mitglieder herzlich eingeladen. Der Umfang der zu erstellenden Software ist so groß, daß jedes Gramm Gehirnschmalz im Club gefragt ist.

Durch das Herumwerkeln mehrerer Programmierer ist die Funktion eines Software-Koordinators nötig (Freiwillige bitte vortreten !). Dieser Job bedeutet, einen Überblick über die Gesamtarbeit zu gewinnen, diese in Module zu unterteilen und die Zusammenarbeit der einzelnen Module durch Festlegen von gemeinsam genutzten Datenfeldern und zu übernehmenden und zu übergebenden Parametern der einzelnen Module (in den CPU-Registern) zu gewährleisten. Diese Vorgaben sind von den Programmierern der einzelnen Module einzuhalten. Wer dazu Vorstellungen hat, möge bitte nicht durch bescheidene Zurückhaltung versuchen, einen guten Eindruck zu machen.

Auch wenn jemand zu Teilaspekten der Betriebssoftware Ideen hat und die Programmierung dieses Parts übernehmen möchte (oder auch nur seine Idee einbringen möchte), kann er einen wertvollen Beitrag zu dem Projekt leisten.

Als Bestechung in Naturalien bekommt jeder, der sich an dem Projekt

beteiligt, von mir eine geätzte Terminalkarte geschenkt, die er allerdings selbst und mit eigener Finanzierung der Bauteile aufbauen muß. Hercules-Karten kann ich auch günstig beibringen (zur Zeit für 40,-DM). Wer den Job des Software-Koordinators übernehmen will, bekommt auch die Hercules geschenkt (wenn der Club mit seinem gewaltigen Konto das nicht übernehmen will, stifte ich die Karte).

Um das ganze Projekt etwas konkreter darzustellen, will ich im folgenden die bestehenden Vorstellungen zur Hardware und Software, wie Andreas Magnus und ich uns das vorstellen, breittreten. Wie gesagt, davon ist noch nichts endgültig. Insbesondere bei der Software-Vorstellung kann noch alles ganz anders gestaltet werden. Das Folgende ist also nur als Diskussionsgrundlage zu werten.

So könnte die Hardware aussehen:

Das Terminal ist ein Z80-Computer mit 32K Arbeitsspeicher und 32K Hercules-Video-RAM. Nach dem Einschalten besteht der Arbeitsspeicher beim Lesen aus EPROM und beim Schreiben aus RAM. Durch ein LDIR wird der EPROM-Inhalt bei der Initialisierung ins RAM kopiert. Danach wird das EPROM abgeschaltet und beim Lesen und Schreiben in den unteren 32K erreicht der Z80 das RAM.

Die oberen 32K des Z80 sind dem Video-RAM der Hercules vorbehalten. Hier können die beiden 32K HRG-Pages der Hercules bzw. das Text-/Attribut-Video-RAM eingeblendet werden. Im Textmodus bietet die Hercules-Karte 8 linear hintereinander liegende Seiten mit je 4K Text-/Attribut-Speicher.

Im I/O-Adreßraum des Terminal-Z80 sind die Hercules-internen Ports, das Interface zum Host-Computer, ein Latch zur Einstellung der Konfiguration des Terminals und ein Tastatur-Interface unterzubringen. Zur Vereinfachung der Hardware scheinen folgende Portadressen die sinnvollsten zu sein:

B0H-BFH	interne Ports der Hercules-Karte (siehe Artikel zur Hercules im GENIE)
00H	Status-Register des Interface zum Host-Computer
01H	Daten-Register des Interface zum Host-Computer
02H	Konfigurations-Latch
03H	Tastatur

Das Konfigurations-Latch unterstützt bislang folgende Befehle:

OUT (02H),0	Einschalten des Text-/Attribut-Speichers im Textmodus bzw. der HRG-Seite 0 im Grafik-Modus
OUT (02H),1	Einschalten der HRG-Seite 1; im Text-Modus ist der Text-Speicher nicht verfügbar
OUT (02H),2	im Bereich 0-32K ist bei Leseoperationen das EPROM eingeblendet
OUT (02H),3	sowohl beim Lesen als auch beim Schreiben liegt im Bereich 0-32K RAM vor
OUT (02H),5	und danach
OUT (02H),4	Zurücksetzen des Interface der PC-XT-Tastatur

Die Ausgabe der Werte 6-15 kann für weitere Konfigurations-Einstellungen benutzt werden. D0 gibt dabei jeweils den Pegel einer entsprechenden Steuerleitung wieder (Q3-Q7 des 74LS259-Latch) und D1-D3 geben vor, welche Steuerleitung geschaltet werden soll.

Die Portadressen, unter denen das Terminal für den Host-Computer erreichbar sein soll, lassen sich über Jumper in Schritten von 4 beliebig einstellen. Die Basisadresse X+0 ist das Status-Register. D0 des Status-Registers zeigt an, ob das Terminal ein weiteres Byte annehmen kann:

DO=0 : das Terminal kann ein weiteres Byte annehmen
=1 : es darf noch kein weiteres Byte an das Terminal ausgegeben werden

D7 gibt an, ob das Terminal bei der Ausgabe von Daten an den Host ein neues Byte bereitgestellt hat:
D7=0 : es steht kein neues Byte des Terminals bereit
=1 : das Terminal hat ein neues Byte ausgegeben

Je nach Zustand der Bits 0 und 7 des Status-Ports kann die CPU an den Daten-Port X+1 ein weiters Byte ausgeben bzw. von dort ein weiteres Byte abholen.

Für den Z80 des Terminals liegen die Adressen der Ports des Interface fest (00H = Status-Register; 01H = Daten-Register). Durch Lesen des Satus-Registers kann der Terminal-Z80 feststellen, ob er Daten entgegennehmen kann/muß bzw. ausgeben kann/muß:
D0=0 : es kann/muß ein weiteres Byte an den Host ausgegeben werden (der Host hat das letzte Byte abgeholt)
=1 : es darf noch kein weiteres Byte an den Host ausgegeben werden
D7=0 : es liegt kein neues Byte vom Host vor
=1 : der Host hat ein neues Byte ausgegeben, das abzuholen ist

Zur Vereinfachung der Kommunikation zwischen Host und Terminal wird am Terminal-Z80 ein Interrupt erzeugt, wenn der Host ein neues Datum ausgegeben hat.

Als Tastatur kann wahlweise eine 8Bit-Parallel-ASCII-Tastatur oder eine serielle PC-XT-Tastatur verwendet werden. Beide Interfaces sind bei der bislang bestehenden Prototypen-Karte vorhanden und können durch Jumper selektiert werden. Wenn von der Tastatur ein Byte gekommen ist, wird am Terminal-Z80 ein /NMI ausgelöst.

Und das sollte die Software unterstützen:

In allen vorgesehenen Betriebsmodi des Terminals sind Umschaltbefehle möglich, die in einen jeweils anderen Modus umschalten oder sonstige Spezialaufgaben erledigen:

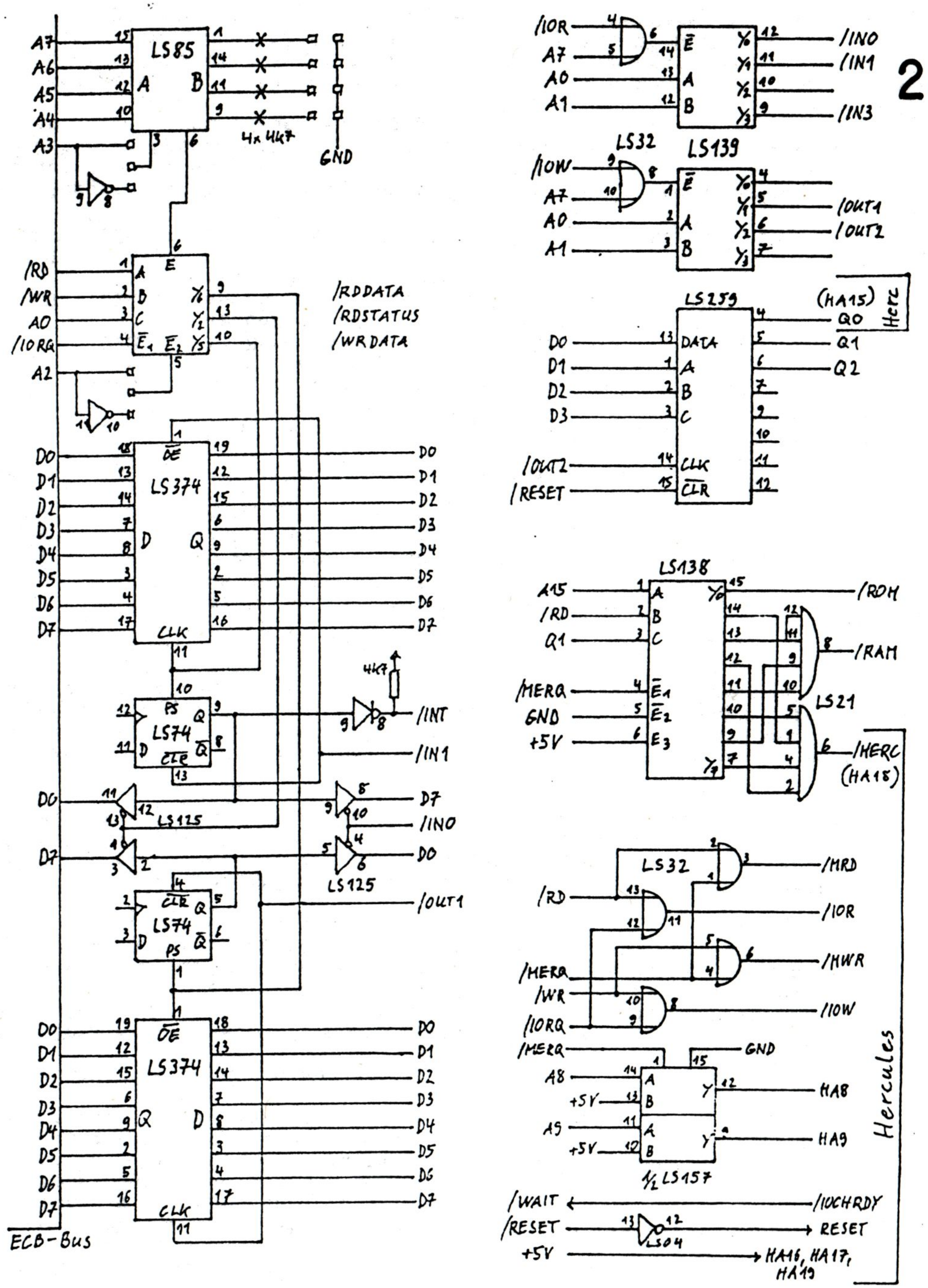
1) Einschalten des TVI 950 Textmodus (Grundeinstellung)
2) Einschalten des Tektronix 4014 Grafikmodus
3) Umschalten zwischen Local Mode und OnLine-Betrieb
4) Übernehmen eines Maschinenprogramms vom Host; Der Befehl enthält als Parameter Ladeadresse, Länge und Call-Adr.
5) Umschalten zwischen Video- und Drucker-Ausgabe
6) Übernahme eines Initialisierungsstrings und der Bit-Grafik-Einleitung für den jeweils an der Hercules angeschlossenen Drucker
7) Hardcopy des Bildschirmes (sowohl Text als auch Grafik, was gerade eingestellt ist)

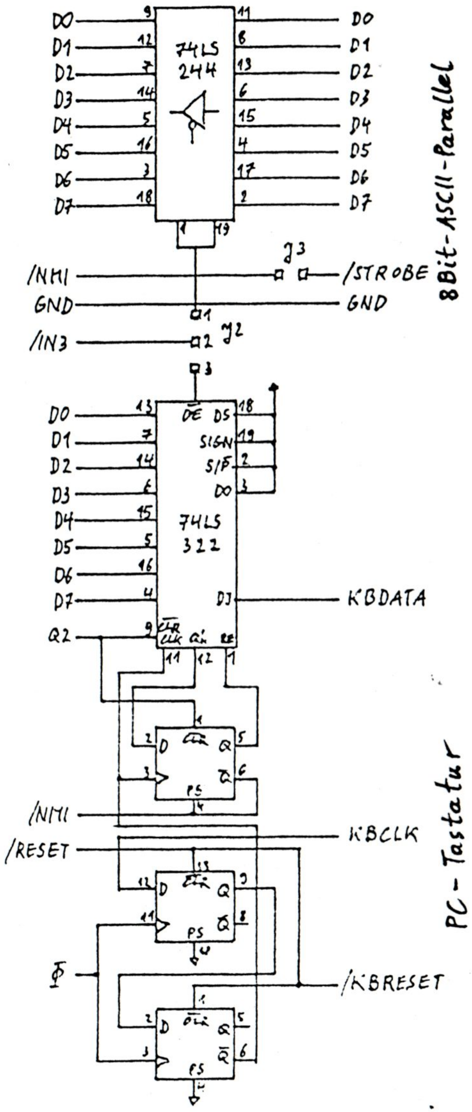
Die ersten 3 Punkte machen schon ein recht ansehnliches Terminal aus. Die Übernahme und optionale Ausführung von Maschinenprogrammen hat haupt

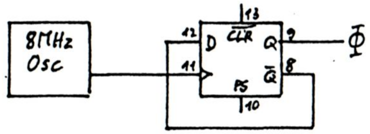
sächlich in der Entwicklungsphase der Software ihre Bedeutung. Außerdem erlaubt sie auch später noch den Mißbrauch der Z80-Power des Terminals für Anwenderzwecke.

Der Punkt 5 erschlägt mit dem Terminal auch gleich eine Drucker-Schnittstelle. Die Punkte 6 und 7 drängen sich bei der gemeinsamen Lage von Video- und Drucker-Interface geradezu auf.

Als Grundstock an Treiberroutinen müssen die Initialisierung des Textbildschirms und des Grafikbildschirms sowie ein Tastatur-Treiber vorhanden sein. Für die beiden Video-Modi muß je ein Interpreter für die im jeweiligen Modus gültigen Contol-Codes und Escape-Sequenzen existieren, der auch immer die oben angeführten globalen Umschaltbefehle mit berücksichtigt.







Schaltungs-Auszüge

links:
Portdecodierung der Terminalkarte und ECB-Bus-Interface

Mitte:

Erzeugen interner I/O-Freigabesignale

Konfigurations-Latch (LS259)

Erzeugen der memory mapped Freigabesignale

Busanpassung für PC-Slot

rechts:

Interfaces für wahlweise 8Bit-ASCII-Parallel-Tastatur oder IBM-PC-Tastatur

Systemtakt-Erzeugung

Die Befehle des Terminals Grip 5.x

Um irgendwo anzufangen, habe ich mir die Leistungen der GRIP 5.X von Conitec als Vorlage für das Hercules-Terminal herangezogen. Dort werden ebenfalls ein TVI 950 Textterminal und ein Tektronix 4014 Grafikterminal unterstützt, eine Tastatur und ein Drucker angeschlossen und ähnliche Zusatzleistungen bereitgestellt.

Die mit dem Hercules-Terminal möglichen Befehle der GRIP seien im Folgenden mit knapper Beschreibung zusammengestellt. Alle Features der GRIP, die sich bei derselben als unschön erwiesen haben, sind hier garnicht erst aufgeführt.

ESC ESC 'A'	1BH 1BH 41H	TVI 950 Textmodus einschalten (Einstellung nach RESET)

TVI 950 Steuercodes:

^H	08H	Cursor nach links
^I	09H	Cursor nach rechts (mit Scroll)
^K	0BH	Cursor eine Zeile hoch
^V	16H	Cursor eine Zeile runter (mit Scroll)
^J	0AH	Line Feed (wie ^V)
^M	0DH	Carriage Return, Cursor an den Zeilenanfang
^_	0FH	Neue Zeile (:= Carriage Return + Line Feed)
^^	1EH	Cursor Home (Zeile 0, Spalte 0) ohne ClrScr
^Z	1AH	Bildschirm löschen
^L	0CH	Form Feed (:= Clear Screen + Cursor Home)
^G	07H	Rabatz auslösen (muß außerhalb des Terminals erzeugt werden)

TVI 950 Escape-Sequenzen:

ESC '+'		Form Feed (wie ^L)
ESC ';'		Form Feed (wie ^L)
ESC '.'	a	Cursor-Attribute setzen
	a = '0'	Unsichtbarer Cursor
	'1'	Blinkender Block (nach Reset)
	'2'	Stehender Block
	'3'	binkender Unterstrich
	'4'	stehender Unterstrich
	'5' '6' '7'	diverse User-Cursor
ESC '='	y x	Corsorposition setzen
ESC '?'		Cursorposition abfragen, ergibt 3 Bytes als Antwort: y x <CR>
	y	Zeilennummer + 20H
	x	Spaltennummer + 20H
		bei der Antwort auf die Abfrage ist in [y, x, CR] das Bit7 gesetzt
ESC '1'		Bildschirm um 1 Zeile hochscrollen
ESC '2'		Bildschirm um 1 Spalte nach rechts scrollen
ESC '3'		Bildschirm um 1 Zeile runterscrollen
ESC '4'		Bildschirm um 1 Spalte nach links scrollen
		der Cursor bewegt sich mit dem Text mit bis zum Rand
ESC 'E'		Leerzeile an der Cursorposition einfügen
ESC 'R'		Zeile an Cursorposition löschen
		die nachfolgenden Zeilen werden entsprechend bewegt
ESC 'T'		Zeile ab Cursor bis zum Ende löschen
ESC 'Y'		Seite ab Cursor bis zum Ende löschen
ESC 'G'	x	Zeichenattribute setzen

x:	D0=1:	Zeichen unsichtbar	möglich
	D1=1:	Zeichen duchgestrichen	nicht möglich
	D2=1:	Zeichen invers dargestellt	möglich
	D3=1:	Zeichen unterstrichen	nicht möglich
	D5,D4 = 0,0:	Subscript	nicht möglich
	0,1:	Indexschrift	nicht möglich

HEFT 31 Juli 1990

1,0:	Superscript	nicht möglich
1,1:	normale Schrift (default)	möglich
D6=1:	Breitschrift	nicht möglich
D7	ist immer low auszugeben	

ESC ')'	Invertieren einschalten	
ESC '('	Invertieren ausschalten	
ESC 'n'	Bildschirm eingeschaltet	
ESC 'o'	Bildschirm dunkel	
ESC '$'	Strichgrafik ein	Multiplan-kompatible
ESC '%'	Strichgrafik aus	Zeichen anstelle der Großbuschstaben

ESC ESC 'T' Tektronix-Grundmodus einschalten

Tektronix-Control-Codes

^M	0DH	Alphamodus ein, Cursor an den Anfang der Zeile der letzten Grafikposition setzen
^_	1FH	Alphamodus ein, Cursor auf die letzten Graphik-Koordinaten setzen
^Ö	1CH	Punktmodus ein
^Ü	1DH	Vektormodus ein; der erste Vektor wird nicht gezeichnet, die Koordinaten werden als neuer Anfangspunkt übernommen
^G	07H	Sonderfunktion (nach ^Ü): der Vektor wird doch gezeichnet
^^	1EH	Inkrementalmodus ein
^Q	11H	lösche mit Schwarz (Hintergrundfarbe)
^R	12H	zeichne invers (Vordergrund: schw<->weiß)
^S	13H	zeichne mit weiß (->Vordergrundfarbe)

die letzten 3 Steuercodes sind nicht Bestandteil des Tektronix-Befehlssatzes sondern Erweiterungen der GRIP

Tektronix-Escape-Sequenzen

ESC '^L'	Clear Screen, Alphamodus ein, Cursor home
ESC '1'	Bidschirm um 1 Zeile hochscrollen
ESC '2'	Bidschirm um 1 Spalte nach rechts scrollen
ESC '3'	Bidschirm um 1 Zeile runterscrollen
ESC '4'	Bidschirm um 1 Spalte nach links scrollen

die letzte Grafik-Koordinate scrollt mit, die Position realtiv zum Nullpunkt ändert sich also

ESC 'G' x	Zeichenattribute für Alphamodus setzen Attribute sind die gleichen wie bei TVI 950
ESC '`'	Vektortyp := durchgehende Linie (default)
ESC 'a'	Vektortyp := gepunktete Linie
ESC 'b'	Vektortyp := Strichpunkt-Linie
ESC 'c'	Vektortyp := kurz gestichelte Linie
ESC 'd'	Vektortyp := lang gestichelte Linie
ESC 'F'	Fülle Fläche, in der der Grafik-Cursor steht mit dem Muster des Zeichens 7FH

Der Vektormodus (mit ^Ü einschalten)

Bei der Hercules beträgt die Grafikauflösung 720(x) * 348(y) Punkte. Der Ursprung (0,0) liegt unten links. Eine Koordinate wird durch 2 10Bit-Werte angegeben, die als 4 Bytes mit 5 signifikanten Bits (D0-D4) an das Terminal übertragen werden. Welche Funktion das jeweilige Byte hat, wird durch D5-D7 codiert:

0,0,1	obere 5 Bit der Y-Koordinate
0,1,1	untere 5 Bit der Y-Koordinate
0,0,1	obere 5 Bit der X-Koordinate
0,1,0	untere 5 Bit der X-Koordinate

Die Übertragung erfolgt in der Reihenfolge: Yhigh, Ylow, Xhigh,Xlow
Folgende Formate zur Koordinaten-Übergabe sind auch erlaubt:

Yhigh Xlow	für die nicht übergebenen Anteile
Ylow Xhigh Xlow	der Koordinaten werden die zuletzt
Ylow Xlow	übergebenen Werte wieder verwandt
Xlow	

Mit dem Empfang des Xlow-Bytes ist die Übertragung eines Koordinaten-Wertepaares abegschlossen, das Terminal zeichnet dann einen Vektor vom letzten Koordinatenpunkt zum neu angebenen Punkt.

Der Punktmodus

funktioniert wie der Vektormodus, es wird aber keine Linie sondern nur der Endpunkt gezeichnet. Lösch- und Invertier-Befehle wirken auch hier.

Der Inkrementalmodus

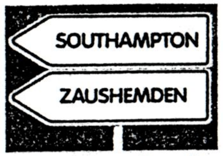
zeichnet von der letzten Grafik-Koordinate in 1-Punkt-Schritten weiter; die Werte (X,Y) des Grafik-Cursors laufen mit. Die Richtung wird durch die folgenden Zeichen vorgegeben:

F	D	E	'F' nach oben links
			'D' nach oben
B	*	A	'E' nach oben rechts
			'B' nach links
J	H	I	'A' nach rechts u.s.w.

Durch ' ' 'P' 'Q' 'R' 'T' werden die Zeichenmodi eingestellt. Ein Modus ist so lange wirksam, bis ein neuer eingeschaltet wird. Nach dem Einschalten des Inkrementalmodus muß einer dieser Befehle folgen:

' '	Punkte nicht zeichnen, nur Grafik-Cursor bewegen
'P'	Punkte zeichnen (weiß, wie ^S)
'Q'	Punkte löschen (schwarz, wie ^Q)
'R'	Punkte invertieren (wie ^R)

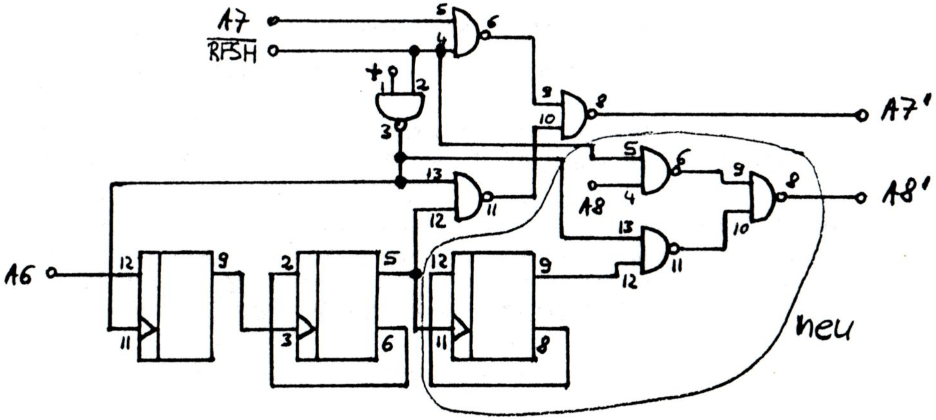
Helmut Bernhardt



Klasse! Zum Beginn der Reisesaison haben sich die Engländer etwas einfallen lassen: Speziell für deutsche Touristen ist die Beschilderung jeweils unter der oft unverständlichen örtlichen Bezeichnung in deutscher Sprache ergänzt. Wir meinen: ein liebenswürdiger Sörwiß!

1MB-Banker mit 1MB

Da Helmuts Banker ein ganzes Megabyte verwalten kann liegt es natürlich nahe, den Speicher auch soweit auszubauen. Die Möglichkeit mit vier übereinander gestapelten 256er RAMs funktioniert zwar und sieht phänomenal interessant aus, ist aber sicher nicht der Weisheit letzter Schluß. Einmal braucht man viel Platz nach oben und zweitens werden die Dinger ganz schön warm, entsprechend ist der Strom, der da verheizt wird. Inzwischen sind aber die Mega-Chips auch schon in bezahlbaren Regionen angelangt und damit löst sich fast alles in Wohlgefallen auf. Probleme bereiten nur noch die Größe der Chips und der Refresh. Da die neuen Käfer zwei Beine mehr haben als die alten und die Belegung von der der 256er etwas abweicht habe ich einen Platinenstreifen mit den 18-poligen Fassungen bestückt und die Signale einfach zu den alten Fassungen rübergefädelt. Geht einwandfrei und läuft auch mit 10MHz problemlos. Aber jetzt kommts. Für die 256er mußte der Refresh-Zähler bereits auf 8 Bit aufgebohrt werden und die Mega-Chips brauchen sogar einen 9-Bit-Refresh. Mit zwei weiteren ICs läßt sich aber auch dieses Problem ohne allzugroßen Aufwand lösen. Wenn man beim Umbauen mit dem Refresh anfängt, muß sich der Rechner hinterher noch ganz genauso verhalten wie vorher, da es die 256er RAMs überhaupt nicht stört, wenn sie mit einer 9-Bit-Adresse betan werden.



Alexander Schmid

BÖRSE -- BÖRSE -- BÖRSE

H. G. Lütz

Heintzmannstr. 147
4630 Bochum 1
Tel. 0234-707070

Alexander Schmid
St. Kajetanstr. 38/IV
8000 München 80

Sehr geehrter Herr Schmid!

Vielen Dank für die Übersendung einer Club-Zeitschrift.
Nach Systemumstieg biete ich folgendes an:

1 TRS 80 Mod. 2
2 8-Zoll Laufwerke
1 Line Printer 3

Preis 333,- DM

Mit frdl. Gruß

Hans-Georg Lütz

Schnittstellen-Programmierung

von Gerald Schröder

In jedem Betriebssystem (und auch in manchen anderen Programmen) gibt es Treiber, die Peripherie-Geräte ansteuern. Z. B. verfügen die meisten Textverarbeitungsprogramme über eigene Drucker-Treiber, weil die des Betriebssystems ewig auf ausgeschaltete Drucker warten (oder das gar nicht bemerken), zu langsam sind, gewisse Zeichen verschlucken, kein Spooling ermöglichen usw.

Nun fragt sich natürlich: Wie programmiere ich einen Treiber? Die Frage läßt sich nicht so einfach beantworten, denn es kommt auf zwei Dinge an:

1. Um was für eine Art (Klasse) Peripheriegerät handelt es sich?
2. Welche Hardware zur Ansteuerung habe ich zur Verfügung?

Geräte-Klassen: Zeichen- und blockorientierte Geräte

Die erste Frage führt uns zu einer Unterscheidung in zwei Klassen: Die *zeichen-* und die *blockorientierten* Geräte.

Zeichenorientierte Geräte

Bei den zeichenorientierten Geräten wird immer nur ein Zeichen zur Zeit ein- oder ausgegeben. Darunter fallen Drucker, Bildschirme, Tastaturen und serielle Schnittstellen. Nach der Ausgabe eines Zeichen ist das Gerät meist eine gewisse Zeit mit der Verarbeitung beschäftigt, bevor es das nächste Zeichen empfangen kann; so lange müssen wir warten bzw. etwas anderes tun. Andererseits müssen wir dem Gerät nicht immer Zeichen liefern, wenn es (z. B. der Drucker) bereit ist. Bei der Eingabe rechnen wir nur sporadisch mit Zeichen, wobei es Probleme geben kann, wenn wir nicht rechtzeitig das Zeichen abholen (z. B. von der seriellen Schnittstelle). Dann überschreibt das nächste ankommende Zeichen das vorherige, was (falls bemerkt) dann Neu-Deutsch als „Overrun“ bezeichnet wird.

Oft wird noch weiter abstrahiert, indem wir sagen: Es gibt einen „Strom“ von Bytes („Byte Stream“) vom oder zum Gerät. Programmierer kennen das vom Datentyp „sequentieller File“, wobei darunter z. B. bei TURBO-PASCAL auch der Drucker fällt: Es kann immer nur sequentiell das nächste Byte gelesen oder geschrieben werden, eine Adressierung („Byte Nr. 10“) gibt es meistens nicht.

Blockorientierte Geräte

Bei den blockorientierten Geräten dagegen, zu denen vor allem Massenspeicher wie Floppy-Laufwerke gehören, tritt nie ein Zeichen allein auf, sondern es werden immer Blöcke übertragen. Meistens sind die Blocklängen festgelegt und typischerweise finden wir 128, 256, 512 oder 1024 Bytes je Block. Bei jedem Schreiben oder Lesen wird hier dem Gerät erstmal über ein Kommando mitgeteilt, daß jetzt geschrieben oder gelesen werden soll, und um welchen Block es sich handelt. D. h. hier findet eine Adressierung („Block Nr. 10“) statt!

Beim Lesen oder Schreiben des Blocks haben wir es dann natürlich wieder mit einzelnen Bytes zu tun, die aber meist so schnell hintereinander gelesen/geschrieben werden müssen, daß die Zusammenzufassung zu einem Block gerechtfertigt ist. Der Zeitfaktor ist hier ganz anders gelagert als im zeichenorientierten Fall: Beim Schreiben *muß* das nächste Zeichen geliefert werden, wenn das Gerät dies verlangt („Data Request“), sonst gehen Zeichen verloren („Data lost“). Ebenso beim Lesen: Wenn das Zeichen nicht schnell genug abgeholt wird, kommt es wie oben zu einem „Overrun“, wobei jetzt aber der ganze Block nicht mehr stimmt!

Soweit zu den beiden Klassen. Jetzt wird es etwas konkreter: Welche Möglichkeiten für die Ein-/Ausgabe gibt es?

Programmierung: Polling, Interrupt, DMA

Die sich uns bietenden Möglichkeiten zur Ein-/Ausgabe-Programmierung hängen vor allem von der vorhandenen Hardware ab. Oben habe ich immer nur von „Geräten“ gesprochen; meistens finden wir aber zwischen dem Hauptprozessor und dem Gerät noch einen oder mehrere Chips, die „Controller“ genannt werden. Diese Controller sind Spezial-Prozessoren, die dem Hauptprozessor einige Arbeit abnehmen. Drei unterschiedliche Controller-Typen spielen im Bereich „Ein-/Ausgabe“ eine Rolle:

- **Ein-/Ausgabe-Controller**: Das sind Chips, die sich nur um ein bestimmtes Gerät bzw. dessen Ansteuerung kümmern. Wichtig sind für uns Floppy-Controller und SIO-Controller („Serial Input/Output“, serielle Schnittstelle). Wenn ich im folgenden Teil von „Gerät“ spreche, meine ich meistens den zu diesem Gerät gehörenden Controller, denn für den Hauptprozessor sieht der Controller so aus, als wäre er selbst das Gerät.
- **Interrupt-Controller**: Manchmal hat ein Rechner oder ein Hauptprozessor intern (z. B. der HD64180) auch einen Interrupt-Controller. Dieser verwaltet für den Hauptprozessor die verschiedenen Interrupt-„Quellen“, wobei es sich meist um die Controller von Geräten handelt. Die Controller melden sich meistens über Interrupts, wenn eine Zustandsänderung auftritt, wenn sie also wieder bereit sind, ein Byte zu empfangen, oder eins senden möchten oder ein

Fehler von ihnen erkannt wurde. Der Interrupt-Controller entscheidet dann, was zu passieren hat. Normalerweise wird der Hauptprozessor bei der Abarbeitung seines normalen Programms unterbrochen. Wenn mehrere Interrupts gleichzeitig auftreten oder gerade ein vorhergehender noch bearbeitet wird, entscheidet der Interrupt-Controller, welcher Interrupt Vorrang hat. Er kann auch einige oder alle Interrupts sperren, so daß deren Auftreten den Hauptprozessor bei seiner Arbeit nicht stören.

- **DMA-Controller**: „DMA" bedeutet „Direct Memory Access". Ein DMA-Controller darf also direkt auf den Hauptspeicher (RAM) zugreifen, ohne den Hauptprozessor bemühen zu müssen. Der DMA-Controller liest und/oder schreibt selbst Bytes, während der Hauptprozessor weiter sein eigentliches Programm abarbeitet. Für DMA-Controller gibt es zwei Anwendungen: Verschieben von Speicherblöcken, Lesen oder Schreiben von Blöcken von/auf Geräte(n). Uns interessiert natürlich nur der letzte Fall. Besondere Zustandsänderungen (z. B. das Ende eine Übertragung) werden wieder an den Hauptprozessor gemeldet, evtl. über einen Interrupt-Controller.

Einerseits müssen also in unserem Rechner die nötigen Controller vorhanden sein, vor allem ein Interrupt- und ein DMA-Controller, andererseits muß die Architektur von Rechner und Betriebssystem darauf abgestimmt sein. So verfügen z. B. die Z80-Prozessoren von Haus aus über eine sehr mächtige Interrupt-Behandlung in Form des Interrupt-Modus 2 (IM 2), aber nur wenige Rechner bzw. Betriebssysteme nutzen dies aus (NEWDOS gar nicht und CP/M selten). Noch besser sind der HD64180 und der Z280, die beide auf dem Hauptprozessor-Chip einen Interrupt- und mehrere DMA-Controller beherbergen, was aber auch selten ausgenutzt wird.

Bei jeder Programmierung gibt es drei verschiedene Fälle zu beachten:

- **Eingabe** bzw. **Empfangen**: Bei der Eingabe bzw. dem Empfangen müssen die ankommenden Bytes rechtzeitig abgeholt werden, damit nachfolgende Bytes sie nicht überschreiben. Hier kommt es also auf schnelle Abholung an.

- **Ausgabe** bzw. **Senden**: Bei der Ausgabe bzw. dem Senden ist zu beachten, daß wir die Bytes nicht beliebig schnell ausgeben können. Meist braucht das Gerät (bzw. dessen Controller) nach jeder Ausgabe erst einmal eine „Denkpause", bevor wir das nächste Byte senden können. Hier dürfen wir also nicht zu schnell sein.

- **Fehler**: Neben der normalen Ein-/Ausgabe können natürlich auch Fehler auftreten, wenn wir etwas falsch programmiert haben (Bytes zu langsam angeholt oder zu schnell ausgegeben) oder ein physikalischer Fehler auftrat (Diskette kaputt, Leitung unterbrochen, Checksummen-Fehler usw.). Wir müssen dauernd auf Fehler gefaßt sein und geeignet darauf reagieren. Der Rechner sollte nicht abstürzen (in einer Endlosschleife hängenbleiben); wenn möglich, sollten wir den Fehler beheben (nochmal versuchen); falls das nicht klappt, ist die Ein-/Ausgabe-Operation mit einer Meldung an den Benutzer abzubrechen.

Als Fallbeispiel für die verschiedenen Möglichkeiten werde ich die Ein-/Ausgabe von/auf eine serielle Schnittstelle nehmen. Sehr ähnlich sieht die Verbindung zum Drucker (wenn auch nur Ausgabe) oder der Tastatur (nur Eingabe) aus. Dabei werde ich keinen Z80-Code benutzen oder eine bestimmte Hardware (Adressen, Ports, Chips) voraussetzen, sondern alles in einer „Meta-"Sprache, einer Art Pascal, formulieren. Das muß dann jeder für sich auf seine Hardware und Lieblings-Sprache übersetzen.

Polling

Die einfachste und am häufigsten benutzte Methode der Ein-/Ausgabe-Programmierung ist das „Polling" (ein entsprechendes deutsches Wort fällt mir nicht ein). Dabei erledigt der Hauptprozessor, bei uns also der Z80, die Haupt-Arbeit und steht währenddessen für nichts anderes zur Verfügung. Er muß dauernd die Status-Leitung(en) des Geräts (bzw. Controllers) abfragen und warten, bis das Gerät zur Ein- oder Ausgabe bereit ist. Das entsprechende Programmstück sieht so aus:

```
REPEAT                                      (* wiederhole                   *)
 WHILE No_RW_Request AND No_Error DO        (* solange E/A-Anforderung u.   *)
   Nothing                                  (* kein Fehler: warten          *)
 END_WHILE                                  (*                              *)
 IF No_Error THEN                           (* falls kein Fehler:           *)
   Read_Or_Write_Byte                       (* lies/schreibe das nächste Byte *)
 ELSE                                       (* sonst:                       *)
   Try_To_Correct_Error                     (* versuche, Fehler zu beheben  *)
 END_IF                                     (*                              *)
UNTIL End_Of_Block OR Fatal_Error           (* bis Blockende o. fataler Fehler *)
IF Fatal_Error THEN                         (* falls fataler Fehler:        *)
  Show_Error_Message                        (* zeige Fehlermeldung          *)
END_IF                                      (*                              *)
```

Hier stellen sich gleich mehrere Probleme:

- **Senden oder Empfangen**: Das Gerät (hier: die serielle Schnittstelle) kann entweder ein Byte verlangen oder liefern. Beim Senden *können* wir ein Byte schreiben (bei einem zeichenorientierten Gerät) oder *müssen* es sogar (bei einem blockorientierten Gerät). Beim Empfangen hingegen *müssen* wir das Byte abholen, denn sonst wird es vom nächsten überschrieben. Also: Je nach Fall (Senden oder Empfangen) und Gerät (zeichen- oder blockorientiert) ist eine andere Reaktion angebracht. Aber vielleicht wollten wir *nur* senden, empfangen aber plötzlich ein Zeichen! Was machen wir dann damit? Eine Lösung wären zwei Buffer: ein Sende- und ein Empfangs-Buffer (taucht weiter unten auch wieder auf).

- **Timing**: Sobald der Request kommt, sollten wir sofort reagieren, sonst kann es zu Fehlern kommen. Also darf der Prozessor nichts anderes machen, denn sonst

überschreiten wir evtl. die kritische Zeit zwischen dem Request und dem Abholen oder Senden des Zeichens. Besonders fällt dies bei Floppy-Zugriffen ins Gewicht: Schon die falsche Anordnung der Befehle kann zu Timing-Problemen führen! Auf jeden Fall darf kein dann kein Interrupt auftreten, also werden sie bei Floppy-Zugriffen meist gesperrt. Deshalb verlieren interruptgesteuerte Uhren wie die des NEWDOS bei jedem Floppy-Zugriff einige Millisekunden.

- **Blockgröße:** Wann können wir die äußere Schleife verlassen? Bei blockorientierten Geräten ist das leicht: ein Block hat meist eine feste Länge. Bei zeichenorientierten Geräten wissen wir nur beim Senden, wieviele Zeichen zu übertragen sind, beim Empfangen aber (meistens) nicht. Bei einer seriellen Schnittstelle müssen wir eigentlich *immer* auf empfangene Zeichen warten und dürfen die Schleife nie verlassen! Wenn wir es doch tun, z. B. um ein Zeichen von der Tastatur zu holen oder auf dem Bildschirm anzuzeigen, verpassen wir (besonders bei hohen Übertragungsraten) evtl. das nächste empfangene Byte. Mögliche Lösungen: Niedrige Baudraten, Stop- („Bitte nichts mehr senden!") und Start-Bytes („Ich bin wieder soweit!"), feste Blockgrößen auch bei zeichenorientierten Geräten (evtl. enthält das erste Byte die Blockgröße).

Interrupt

Wie Ihr seht, ist das Polling nicht das Gelbe vom Ei, jedenfalls nicht für zeichenorientierte Geräte. Ärgerlich ist vor allem, daß der Prozessor seine ganze Zeit in den Schleifen damit vertrödelt, auf das Gerät zu warten. Es wäre doch viel schöner, wenn der Prozessor seine normale Arbeit erledigt und dabei irgendwann von dem Gerät (oder Controller) unterbrochen wird.

Da haben wir schon unser Stichwort: „Unterbrechung", auf Neu-Deutsch „Interrupt". Der Prozessor fragt also nicht mehr dauernd den Status des Geräts ab, um auf eine Änderung zu warten, sondern das Gerät meldet selbständig eine Änderung des Zustands an den Prozessor. Dieser verzweigt dann *automatisch* in die entsprechende Routine, die eine angemessene Reaktion ausführt. Nach Beendigung der Routine wird das Programm bei dem Befehl fortgesetzt, den der Prozessor gerade bearbeiten wollte, als der Interrupt auftrat.

Ich zeige das Prinzip hier am Beispiel des Sendens:

```
PROCEDURE Send (Char: BYTE)
BEGIN
 IF Ready_To_Send THEN
    Send_It(Char)
 ELSE
    Put_Into_Buffer(Char)
 END_IF
END Send
```

Wir warten jetzt nicht mehr in einer Schleife, bis das Gerät bereit ist, sondern senden das Zeichen sofort, wenn das Gerät vorher durch einen Interrupt bekanntgegeben hat, daß es bereit ist. Ansonsten merken wir uns das Zeichen und führen unser Hauptprogramm fort.


```
PROCEDURE Send_Int
VAR Char: BYTE
BEGIN
 IF Char_In_Buffer THEN
    Get_From_Buffer(Char)
    Send_It(Char)
 ELSE
    Ready_To_Send := TRUE
 END_IF
END Send_Int
```

Wenn das Gerät dann irgendwann die Sende-Bereitschaft meldet, wird automatisch (!) diese Interrupt-Routine aufgerufen. Falls sich noch ein Zeichen im Buffer befindet, wird dieses gesendet. Ansonsten wird unser Flag Ready_To_Send gesetzt, damit bei einem folgenden Aufruf der Prozedur „Send" (s.o.) das nächste Zeichen sofort gesendet wird.

Ähnlich sehen die Routinen zum Empfangen aus. Aber eine Frage stellt sich noch: Was passiert bei Fehlern? Auch bei einem Fehler wird eine Interrupt-Routine aufgerufen, aber wahrscheinlich wartet das Programm gar nicht auf eine Fehlermeldung, sondern werkelt irgendwo anders herum. Wir müssen uns also (fatale) Fehler merken und dem Programm mitteilen, wenn es dazu bereit ist. Das wäre entweder möglich, wenn „Send" (s.o.) aufgerufen wird (Rückmeldung: „Es gab da mal einen Fehler!") oder über eine spezielle Routine, die das Programm aufrufen kann, um aufgetretene Fehler oder andere Meldungen (z. B. „Alle Bytes wurden gesendet.") abzufragen. Etwas ähnliches findet Ihr im nächsten Abschnitt bei der Routine `Send_Status?`.

DMA

Über DMA-Controller habt Ihr ja schon weiter oben etwas erfahren. Sie eignen sich vor allem für blockorientierte Geräte: Ein Block von Bytes, dessen Länge normalerweise bekannt ist, soll von einem bestimmten Speicherbereich aus an ein Gerät ausgegeben bzw. von einem Gerät in einen bestimmten Speicherbereich gelesen werden. Währenddessen soll der Hauptprozessor weiter werkeln.

```
PROCEDURE Send_Block (Adress: ADDRESS;
                      Length: INTEGER)
BEGIN
 End_of_Send := FALSE
 Error       := FALSE
 Start_DMA (Adress, Length)
END Send_Block
```

So sieht die Routine zum Senden eines Blocks aus. Die Variable `End_of_Send` wird `TRUE`, sobald der Block vollkommen übertragen wurde, `Error` dagegen, wenn ein Fehler auftrat.

```
PROCEDURE DMA_Int
BEGIN
 End_of_Send  := TRUE
END DMA_Int

PROCEDURE Error_Int
BEGIN
  Stop_DMA
  Error  := TRUE
END Error_Int
```

Natürlich muß der Hauptprozessor irgendwie mitbekommen, wie er die Variablen zu besetzen hat. Das wird normalerweise wieder mit Interrupts erledigt. Die beiden Routinen `DMA_Int` und `Error_Int` werden von den Interrupts gestartet, wenn eine Übertragung (mit oder ohne Fehler) beendet wurde.

```
TYPE Status = (Busy, Ready, Error)

PROCEDURE Send_Status? : Status
BEGIN
  IF    Error        THEN RETURN Error
  ELSIF End_of_Send  THEN RETURN Ready
  ELSE                    RETURN Busy
  END_IF
END Send_Status
```

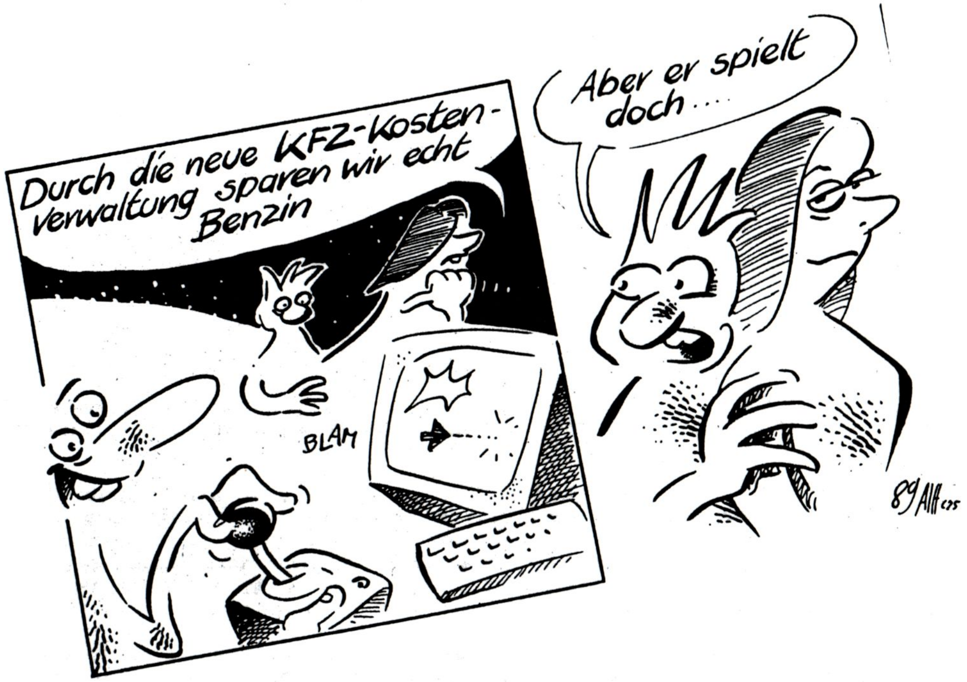
Unser Hauptprogramm kann die Vollzugs- oder Fehler-Meldung über die Routine `Send_Status?` abholen, wobei es folgende Möglichkeiten gibt: Die Übertragung läuft noch („Busy"), sie ist fertig („Ready") oder sie wurde aufgrund eines Fehlers abgebrochen („Error").

Schlußwort

In diesem Artikel habe ich Euch ein bißchen darüber berichtet, was *allgemein* bei der Ein-/Ausgabe-Programmierung möglich ist. Naturgemäß konnte ich dabei nicht sehr tief in die Materie einsteigen, z. B. fehlt noch die Buffer-Verwaltung, der Zugriff aus den Anwendungsprogrammen heraus und die Einbindung in das Betriebssystem.

Interessant wird es erst, wenn Ihr vor einem konkreten Problem steht und selber programmieren (oder vorher noch die Hardware dafür stricken) müßt. Dann erst könnt Ihr anhand der Möglichkeiten (Hardware, Betriebssystem), des Geräts (zeichen-/blockorientiert, Daten-Durchsatz...) und Eurer Anwendung entscheiden, was sinnvoll ist. Ihr werdet dann feststellen, daß vieles bei Euch von vornherein nicht möglich ist, weil Euer Rechner oder das Betriebssystem oder die Anwendungsprogramme es nicht unterstützen. C'est la vie!

PS: Vielen Dank an Hans-Günther Hartmann für die Anregungen und das Korrekturlesen. Und an das Informatik-Rechenzentrum HH für den Laser-Druck per LaTeX.



Von 80 auf 180 oder 280?

von Gerald Schröder

Ihr habt es ja alle schon gelesen: Der Z80 ist "out". "In" sind dagegen der Z180 (bzw. HD64180) und der Z280. Ich möchte hier nun keine genauen Beschreibungen der Prozessoren abliefern. Für den 180er gab's das schon (Sonderheft HD64180), für den 280er dürfte es bald kommen. Aber es stellen sich folgende Fragen:
Was sind die großen Verbesserungen an den beiden Neuen?
Lohnt sich das Umrüsten?
Wenn ja: für wen?

180

Die erste Frage ist für den 180er leicht beantwortet: Man nehme ein komplettes Z80-Board mit: SIO-Controller (serielle Schnittstellen), Timer/Counter, Banking-Logik, DMA- und Interrupt-Controller (jedesmal mindestens ein Chip). Das packe man auf einen Chip, der etwas größer ist als vorher der Z80 und einige Pins mehr hat, und schon hat man einen 180.

Wie Ihr seht: wenn Ihr einen ordentlich aufgebauten Z80-Rechner mit allem drum und dran habt, braucht Ihr kaum einen 180. Er ist vielleicht etwas schneller (aber auch der Z80 kommt bis zu 12 Mhz, wenn ich Alexander glauben darf), hat einige kleine Befehle mehr, die aber kein Programm benutzt, hat eine "Multi-Prozessor-Schnittstelle", die aber kein Board und keine Software unterstützt. Also: Für Euch ist das nichts.

Aber für alle anderen ist das Ding ein Sprung um eine Größenklasse nach vorn. Da kommt Helmut mit seinem HD64180-Adapter, der den Z80 ersetzt, ganz recht. (Helmut wird mich wegen der Schleichwerbung wieder steinigen, aber von Kiel bis Hamburg muß er schon eine ganz schöne Wurfkraft haben...)

Was bringt's Euch? Nun, unter NEWDOS so gut wie nichts, aber unter CP/M wird es interessant. Dort lassen sich eine Menge der neuen "Fietschers" (Heikendorf-Friesisch für "Features") ausnutzen, besonders wenn jemand CP/M-Plus fahren möchte, denn dann ist das Banking des 180er gefragt. Das ist haargenau auf CP/M-Plus zugeschnitten und wartet nur darauf, damit zusammengebracht zu werden. Allerdings legt der TRS-80 seinen Besitzern tausend Steine in den Weg, doch jeder "normale" Rechner verträgt sich ganz gut damit.

280

Den Z280 kenne ich bisher noch nicht "persönlich", aber sein Datenbuch ist doch sehr informativ. Erstmal kann er ähnlich viel wie der 180, hat aber nur eine serielle Schnittstelle (der HD64180 deren zwei sowie eine spezielle). Was kann er mehr?

- bis zu 16 MB Adreßraum (HD64180: 512 Kb oder 1 Mb)
- virtueller Adreßraum mit Pages (statt Banking)
- unheimlich viele neue Befehle
- Cache (256 Byte) und 3-stufige Pipeline
- Daten und Befehle können voneinander getrennt werden
- das Betriebssystem kann vom Anwenderprogramm getrennt werden
- Co-Prozessoren könnten eingesetzt werden (wenn es sie gibt)

Einige von Euch werden jetzt sagen: "Wau, ein ultramoderner Super-High-Tech-Prozessor." Das sind die Freaks, die eh' mehr in der Hardware und dem Betriebssystem stecken und ansonsten ihre Verwaltung lieber auf Karteikarten machen sowie ihre Briefe per Hand schreiben (hallo, Helmut!).
Der größte Teil dürfte sagen: "Was'n dat? Kenn ich nich'." Müßt Ihr auch nicht, denn: Für die Z80-Welt sind das alles Begriffe, die nie im Gespräch waren. Wer einen PC hat, kennt schon eher einige davon, ohne sich groß Gedanken darüber zu machen (dort zählen nur Mhz und Waitstates).

Eins sollte beiden Gruppen gesagt werden: Unter CP/M(-Plus) (geschweige denn NEWDOS) seht Ihr von den Leistungsmöglichkeiten des Z280 nur Bruchteile. Er ist dann wie ein Porsche-Motor im Käfer: Das Drumherum kann die PS nicht unterstützen und mehr als 120 hält die Kiste dann auch nicht aus. Jedenfalls hat noch nicht mal das "modernste" CP/M (3.0) irgendwelche Unterstützung dafür parat.

Nun aber die gute Nachricht: Für die Freaks unter Euch sei gesagt: Eine viel bessere Plattform zum Experimentieren könnt Ihr wohl kaum bekommen. Das Ding kann mehr als mancher IBM-Kompatibler, ist aber flexibler und offener. Für mich sieht das Ding aus wie ein UNIX-Prozessor. Aber: Wer hat schon von Z80-UNIX-Maschinen gehört? Ich nicht. Also bastelt mal schön!

Fazit

Für den normalen Z80-User, der endlich CP/M-Plus komfortabel genießen möchte (und dafür die BIOS-Anpassung selbst vornehmen kann), bietet sich der 180er an. Er ersetzt einfach eine Menge Chips, die auf einer normalen Platine kaum Platz haben und hat alle Möglichkeiten der Z80-Welt nebst einigen Schmankerln.

Für den Power-User (UNIX nennt es "Super-User") bietet sich der Z280 mit seinem Superuser-Modus geradezu unverschämt an. Natürlich kann er auch sein CP/M-Plus darauf fahren, aber das allein bringt es nicht die Bohne.

Eins sollte ich natürlich nicht vergessen: Wenn Ihr vorhabt, Eurem alten TRS-80 oder sonstigen Z80-Rechner endlich die letzte Ruhe zu gönnen, aber der Z80-Welt nicht ganz abhanden kommen wollt, macht's wie ich: Legt Euch einen HD64180- oder Z280-Rechner zu. Dann habt Ihr wenigstens einen Rechner, den nicht jeder hat. Und obendrein habt Ihr eine Menge Möglichkeiten zum Rumspielen in Hard und Soft, die PCs und STs nicht haben.

PS: Natürlich ist das alles meine persönliche Meinung und Ihr dürft alle ganz anderer Meinung sein.

Hallo Clubfreunde,

als neuer Bibliotekar liefere ich meinen "Einstand" gleich mit zwei Buchvorstellungen ab. Die Bücher wurden mir auf der diesjährigen Hauptversammlung übergeben. Beide sind aus dem VEB Verlag Technik in Berlin und behandeln das Thema Technische Informatik. Buch 1 beleuchtet das Thema "CP/M in der Praxis", der andere Band trägt den Titel "MS-DOS". Während der erste Band sicherlich viele aufhorchen läßt, dürfte der MS-DOS Band wohl mehr als Reizstoff empfunden werden. Immerhin wurde ja erneut bekräftigt, daß der Z80 und sein Umfeld weiterhin "Hahn im Korb" bleiben sollen. Aber gehen wir der Reihe nach vor.

Erst wollte ich die Inhaltsverzeichnisse abtippen, wegen deren Länge habe ich diese Idee dann aber sehr schnell fallen gelassen. Stattdessen also neben meinem Kommentar die Kurzbeschreibung von der Rückseite jedes Bandes:

CP/M in der Praxis

"Der Band vermittelt dem Anfänger bei der Arbeit mit einem 8-Bit-Personalcomputer das notwendige Einstiegswissen und ist dem fortgeschrittenen Anwender eines Fachgebietes ein ausreichendes Nachschlagewerk. Es werden das Betriebssystem CP/M mit seinen wesentlichen Systembestandteilen, das Dienstprogramm POWER und das Textverarbeitungssystem WordStar mit seiner Komponente Mailmerge beschrieben"

Meiner Ansicht nach, ist zum Thema CP/M im Ganzen zu wenig zu lesen. Eine umfangreiche Beschreibung des inneren Aufbaues vom CP/M sollte man besser nicht erwarten. Der Schwerpunkt liegt dafür eindeutig auf der Beschreibung der Dienstprogramme, deren Handhabung und Befehlssatz. In diesem Punkt wird dann auch nicht mit Informationen gegeizt. Wer zu den genannten Programmen keine Beschreibung greifbar hat oder einfach mehr zur Bedienung wissen will, wird mit einer Menge an Information bedient. Welche WordStar-Version als Grundlage diente, konnte ich beim Durchsehen nicht feststellen.

MS-DOS

"MS-DOS ist das international am meisten genutzte Betriebssystem für 16-Bit-PC. Das Buch enthält alle wichtigen Informationen für die System- und Anwendungsprogrammierung. Dazu gehören: Hardwareübersicht, BIOS- und DOS-Funktionen, Kommandos, Stapelverarbeitung, Programmierwerkzeuge und wichtige Anwendungsprogramme."

Was ich beim CP/M-Band aufgrund des Titels erwartet hatte, gibt es hier für MS-DOS wirklich satt. Jede Systemfunktion wird beschrieben, inklusive der Belegung der CPU-Register für den Aufruf. Ebenso wird die Struktur von MS-DOS, Datenformate und dergleichen mehr erklärt. Kurz - wer mehr als nur die Schreibweise von AUTOEXEC.BAT und CONFIG.SYS wissen will, wird bestens bedient.

Soviel vorerst zu den "Neuzugängen" und bis zur nächsten Wortmeldung

Tschüß

Kurt

-- Termine -- Termine -- Termine -- Termine --

Nächster Redaktionsschluß .... Anfang November

Bitte denkt an kleinere INFO-Artikel (ca. 1 - 4 Seiten) für unser Clubinfo !!
Natürlich nehme ich auch größere Werke an !

-- Messen '90 --

Systec 90 .... 22.10. - 26.10.
electronica .... München ...... 06.11. - 10.11.

Clubtreffen Süd
Am Sonntag zur Hobbytronik in Stuttgart währe eine gute Gelegenheit sich wieder einmal treffen.
Treffpunkt ist ab 10.00 Uhr (und dann alle zwei Stunden) im Eingangsbereich.
Vielleicht bietet sich die Möglichkeit Fahrgemeinschaften zu bilden. Specht
Euch bitte ab.

- I M P R E S S U M -

1. Vorsitzender
Fritz CHWOLKA
Saarstraße 34
5173 Aldenhoven
☎ 02464 / 8920

2. Vorsitzender
Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 / 2602

Hardwarekoordinator
Andreas MAGNUS
Pommernstr. 4
4650 Gelsenkirchen
☎ 0209 /870230

NEWDOS-Diskothekar
Oliver VOLZ
Waldburgstr. 73
7000 Stuttgart 80
☎ 0711 /7353817

CP/M -Diskothekar
Rüdiger SÖRENSEN
Thomas-Mann-Str. 3a
6500 Mainz 1
☎ 06131 /32860

Club-Bücherei
Kurt MÜLLER
Sophie-Scholl-Ring 38
2054 Geesthacht
☎ 04152 / 70643

Redaktion
Jens NEUEDER
Gschlachtenbretzingen
Rudolf-Then-Straße 32
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren
Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung
des CLUB 80
Postgirokonto

	SonderKonto CLUB 80
	Übermann,H. 6209 Heidenrod
Konto-Nummer	496 071 - 605
Postgiroamt	Frankfurt
BLZ	500 100 60

Eine Zensur oder Kontrolle
der Infobeiträge erfolgt nicht.
Die Redaktion

SCHLUSS

Hallo Club-80er,

nachdem nun scheinbar Frühjahrsmüdigkeit, Sommerloch und andere Unannehmlichkeiten überwunden sind, kamen auch einmal wieder ein paar INFO-Beiträge in meinem Briefkasten an.

Obwohl ich doch noch nicht alle Sparten unserer INFO belegen konnte, habe ich trotzdem mal wieder ein INFO fertiggestellt. Dazu kann ich Euch zusätzlich noch ein Sonderheft

TANDY und GENIE ...ein Hardwareüberblick

bieten.
Zur Zeit habe ich noch zwei Sonderhefte in der Mache. Ansonsten fehlen mir leider halt doch immer die Beiträge für das nächste INFO, um ein geregeltes Erscheinen zu sichern.

Als nächsten Redaktionsschluß habe ich mir nun Anfang November vorgemerkt. Es hängt natürlich auch von Eurer Mitarbeit ab, ob die pünktliche Herausgabe gewährleist werden kann.

Bis demnächst ...
Es grüßt Euch Eure Redaktion

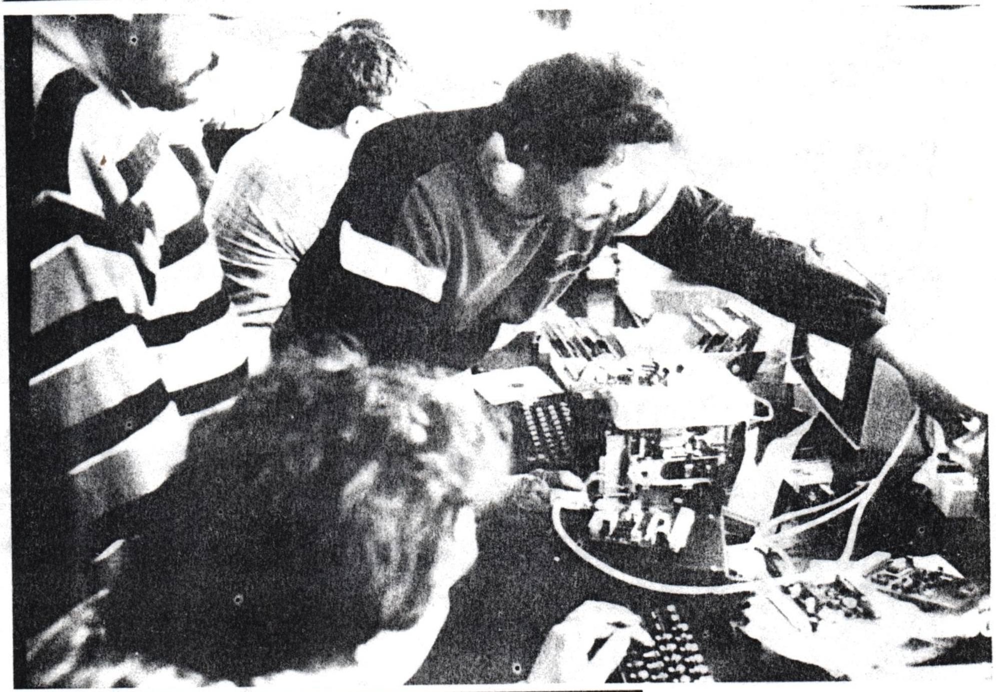
Jens

Karlheinz kopiert
und kopiert ...

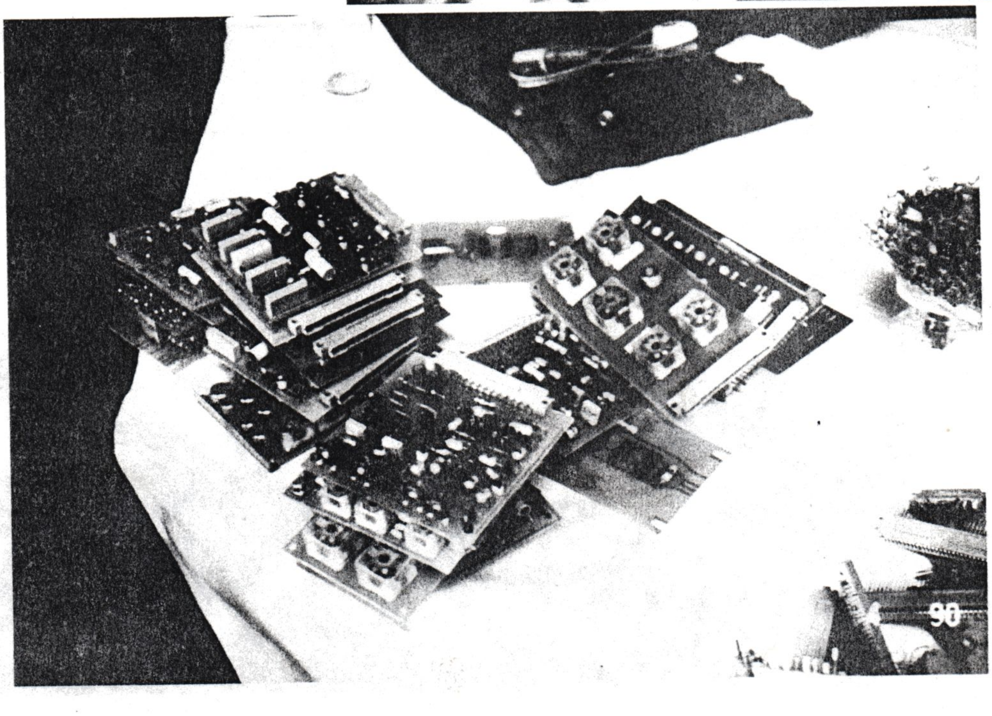
Hacken macht hungrig

... und nachts
bei Frank
die Theorie
(mit Flasche:
der Hausherr)

Dasselbe Bild in West
wie Ost
(links:
Frank-Michael Schober)



Sinclair ZX-Spectrum
+ Technik = Computer



Beute von der Stasi

CLUBINFO

32. AUSGABE

KONTAKTADRESSE : CLUB 80 / FRITZ CHWOLKA / SAARSTRAßE 34 / 5173 ALDENHOVEN
TEL.: 02464 / 8420

Inhaltsverzeichnis

Clubinternes	Autor & Seite
Neues vom Vorstand	1
Neue Mitglieder	2
	Fritz Chwolka
Vorstellung	3
	Claus Littmann
Termine	4
	Redaktion
Traurige Nachricht	4
	Jutta & Hartmut Obermann
Einige Gedanken zum Fortbestand des Club 80	5
	Hans-Martin Stephan
In den Sternen steht's	6 - 7
	Klaus-Jürgen Mühlenbein

Software	
Schneller Booten	9
Fehler in EASE.COM	9
Die RS232 des Model 4p	10
	Alexander Schmid
Bildschirmaufbau ohne viel Aufwand	11 - 12
	Oliver Volz
WINDOW.NDR	13 - 19
	Claus Littmann
Umlaute in LeScript 1.65	20 - 21
	Jens Günther
Download noch einfacher	22 - 27
	Alexander Schmid, c't 12/90
After a long, long Time... Von CP/M nach MSDOS	28 - 32
	Hartmut Obermann
Die Interrupts im Griff	33 - 34
	Arnulf Sopp
ZCPR -- Information und Beschreibung der TCAP	35 - 41
	Fritz Chwolka
Korrekte Syntaxbeschreibungen in der BNF	43 - 44
	Michael Müller

Hardware	
Drucker-Reset	45
	Alexander Schmid
Megabit-RAMs einfach erfrischt	46
2MB-RAM-Erweiterung für Prof 180x	47 - 52
	Helmut Bernhardt

Börse	Autor & Seite
Wer hat was -- wer will was	53 - 56
Werbung	57 - 60

Sonstiges	
Neue Drucker braucht das Land	61 - 62
	Alexander Schmid
DFÜ - was bringts?	63 - 64
Befehlssatz Hayes-kompatibler Modeme	65 - 67
	Hartmut Obermann
Newdos, CP/M, MS-DOS, UNIX	68 - 74
	Gerald Schröder
Menschliches Versagen	75 - 78
Railroad Tycoon	79 - 81
	Hans-Martin Stephan
Zilogs späte Ernte	82
	Heinrich Betz
Zen and the Art of CP/M	83 - 89
	Rüdiger Sörensen, Robert Colem

Club-Diskothek	
CP/M Public Domain	91 - 92
Kompression von Dateien unter CP/M	93 - 94
Der Editor	95 - 97
	Rüdiger Sörensen

Die letzten Seiten	
Impressum	98
Schluß	99 - 100
	Redaktion
Mitgliederadressenliste	am INFO-Ende
GEGRAS-Sonderinfo	Beilage
Genie 3s Z180-Sonderinfo	Beilage

C L U B 80

Fritz Chwolka

5173 Aldenhoven, den 03.12.1990
Saarstr.34
(02464)8920

Liebes Clubmitglied!
Ich hoffe den meisten bin ich inzwischen ausreichend bekannt, auch wenn dies nicht unbedingt in meiner Funktion als erster Vorsitzender des Clubs ist.
Für die anderen möchte ich ein Wenig zu meiner Person erzählen.
Ich bin 32 Jahre alt, von Beruf Betriebsschlosser auf einem Kohlekraftwerk und seit ca. 1979 mit dem Z80 verbunden.
Neben diversen Rechenschiebern habe ich auch ein paar von diesen Kisten, welche sich Computer nennen. Ohne jetzt eine alphabethische Liste erstellen zu wollen zähle ich die *Kisten* mal auf. Da wären ein Tatung TPC-2000, ein Commodore C128, Epson PX-8, ein Morrow MD-3 und mein Lieblingsmodell der Genie IIIs. Im Moment baue ich, wie einige andere auch, den Z280 Rechner vom Dipl.Ing. Tilman Reh auf. Tilman sieht zwar nicht unbedingt gerne die Voranstellung des Berufstitels, aber Ehre wem Ehre gebührt.
Ich hoffe, daß in 1990 wenig Klagen über den Vorstand zu hören waren, außer vielleicht über die manchmal lange Wartezeit beim Beantworten der Briefe, aber dies ließ sich nicht immer vermeiden.
Zwischendurch habe ich den Herrn Thomas Holte ausfindig gemacht und angebaggert. Herr Holte hat für den Genie IIIs eine gute Addaption für CP/M+ geschrieben und mir (inclusive Quellcode) zugesandt. Gleichfalls wurde mir erlaubt den Quellcode des Herrn Holte zum privaten Gebrauch weiterzugeben. Wer also das CP/M+ BIOS für GENIE IIIs im Quellcode haben möchte schicke mir bitte drei rückadressierte Disketten zu.
Abschließend wünsche ich Euch erhohlsame und freudige Feiertage und einen schönen und ruhigen Jahreswechsel.

Mit vielen Grüßen

Rein informativ möchte ich Euch noch die neuen Mitglieder kurz vorstellen.

Neue Mitglieder seit meiner Amtsübernahme:

Achim Benner, Vorm Mühlberg 1, 5190 Kreuztal 3
Rechner CPC, Genie IIIs. Achim hat einiges an PD von mir erhalten,

Willi Johnen, Hansemannstr.1, 5160 Düren
Rechner Genie IIIs mit 10MB HD, Willi sitzt auf der GDOS Schiene und ließ sich noch nicht zum CP/M überreden. Er hat viel mit dem GBASIC gemacht und läßt Uns hoffentlich was zukommen.

Franz Mössel, Schaferstr.12, I-39012 Meran
Rechner CPC. Franz ist auch im SCUG Mitglied und inzwischen Z3PLUS Anwender.

Jürgen Peters, Heukoppel 14, 2000 Hamburg 71
Rechner SHARP MZ800. Jürgen macht Amateurfunk (YAPP) und hat für das Z-MODEM Programm aus der PD eine Anpassung für den Sharp geschrieben. Nebenbei ist Jürgen noch im SCUG-München.

Werner Schilling, Ehndorfer Str.340, 2350 Neumünster
Rechner CPC 6128, Applle kompat., Werner ist Packet-Radio User mit Rufzeichen DB2OU.

Rainer Schmitz, Dornierstraße 17, 7320 Göppingen
Rechner Joyce + Vortex 5.25". Rainer ist Modelleisenbahn-Fan und macht da auch einiges mit dem Joyce. Er ist ebenfalls Z3PLUS Anwender.

Uwe Schobert, Petrus-Waldus-Str. 14, 7136 Oetisheim
Rechner Triumpf-Adler P3 (8085). Uwe sucht ständig nach Programmen für den 8085 Prozessor und arbeitet (noch) mit MICRO SHELL. Momentan versuchen wir ZCPR für 8085 zu installieren.

Frank Schoof, Elpke 5, 4800 Bielefeld 1
Rechner Genie I (64KB). Frank arbeitet mit GDOS/NEWDOS und hat eine grafische Benutzeroberfläche entwickelt, welche er uns demnächst wohl vorstellen wird.

Svend A. Sørensen, Bogholder Allee 76A, DK-2720 Vanløse
Svend hat einen Eigenbaurechner und mir auf mein letztes Schreiben noch nicht geantwortet. Vielleicht hat jemand Interesse an Dänemark und schreibt Ihm ? Ebenso kann Svend einiges an PD gebrauchen.

Stefan Stumpferl, Hasenbergl-Straße 57, 8000 München 45
Rechner: CPC6128, Genie IIIs, PROF 180x
Stefan arbeitet an Betriebssystemerweiterungen und hätte gerne Informationen zu Erweiterungen für den Genie und PD-Software.

Wolfgang Schwarz, Schwedenring 6, 8850 Donauwörth
Rechner C128(D). Wolfgang hat noch nicht viel an PD für seinen C128, kann aber Kaypro-Format zwecks Konvertierung lesen. Ich hoffe daß wir da etwas helfen können.

Dies als Kurzinformation über die neuen Mitglieder, welche sich aber hoffentlich 'persöhnlich' vorstellen werden.

Lieber Jens,

da ich nun seit kurzem auch diesem Club angehöre, bin ich mal so frech und fange gleich mit ´Du´ an. Ich hoffe, niemand nimmt mir das übel. Wie ich dem Probeheft des Club-Infos entnommen habe, ist es üblich sich erst mal vorzustellen. Diesen Brauch finde ich gut. Ich will gleich damit anfangen.

Ich bin schon ein ziemlicher Oldie. Seit kurzem bin ich stolze 36 Jahre. Von daher fällt mir manches nicht ganz so leicht, wie den übrigen Vollprofis in diesem Club. Vor allem auch, weil ich ganz und gar kein Profi bin. Ich habe zwar meinen Computer selbts (nach)gebaut. Aber ich glaube, das kann im Prinzip jeder, der es will. Ich habe bei mir den NDR-Klein-Computer stehen. Er läuft mit einer stinknormalen Z80 mit gut 6 MHz. Die Grafikkarte kann 256 x 512 Punkte auf 4 Seiten darstellen und hat eine eigene CPU (EF 9366) und natürlich eigenen Speicher.

Ansonsten habe ich nichts Weltbewegendes zu bieten. Ich kann mit einem Spezial-Bios verschiedene Formate bearbeiten (leider nicht alle). Den Einstieg in die DFÜ habe ich noch nicht gefunden. Daran wird aber gerade gearbeitet.

Ich arbeite meisten mit NZ-COM (phantastisch!!). Projekte, die ich anfange, brauchen meist ziemlich viel Zeit. Derzeit stehen an: DFÜ, Installation von CP/M+, ZCPR usw.

Für Hilfe bin ich immer dankbar. Natürlich versuche ich auch gern zu helfen.

Nebenbei bin ich bei der hiesigen Sparkasse beschäftigt. Ich habe die unangenehme Aufgabe ausstehende Beträge ´einzutreiben´. Ich bin verheiratet und Vater einer Tochter.

Soviel zu mir selbst.

Claus Littmann
Plockhorst, Zum Spring 15, 3155 Edemissen

Termine... Termine... Termine... Termine... Termine

Clubtreffen Süd	Röfingen	25.01.- 27.01.91
Jahreshauptversammlung	....	......- 91
Clubtreffen Nord	....	......- 91

Redaktionsschluß für die Clubinfo's 1991 ist jeweils der letzte Tag der folgenden Monate:

März
Mai
Juli
Oktober
Dezember

Internationale Musikmesse	Frankfurt	März 91
CEBIT	Hannover	13.03.- 20.03.91
IAA	Frankfurt	12.09.- 22.09.91
Büro-data	Berlin	16.10.- 19.10.91
SYSTEMS	München	21.10.- 25.10.91

Traurige Nachricht!
Mit großem Bedauern mußten wir erfahren, daß unser langjähriges Clubmitglied und guter persönlicher Freund Walter Piller am Montag den 07. Januar plötzlich verstorben ist.

Walter, der in der Nähe von Zürich zuhause war, hatte trotz seiner siebzig Lenze noch viele Pläne. Noch am Wochenende vor seinem Tod hat er uns angerufen und angekündigt, daß er bei gutem Wetter zum Südländertreffen kommen möchte. Um so unerwarteter erreichte uns die Nachricht von seinem plötzlichen Ableben durch Herzversagen.

Meine Frau und ich haben, auch im Namen des CLUB 80. Walters Frau, Helene Piller, herzliches Beileid ausgesprochen.

Jutta und Hartmut Obermann

Einige Gedanken zum Fortbestand des Club 80

P'anta rh'ei, alles fließt, wußte schon der griechische Philosoph Heraklit vor über 2000 Jahren zu sagen. Das ist heute aktueller denn je! Wenn ich mir betrachte, welche Fortschritte die Computertechnik allein in den letzten 10 Jahren gemacht hat, kann ich nur staunen. Leider hat unser Club 80 sich diesem Tempo nicht anpassen können.

Trotz eindringlicher und warnender Appelle ist nun passiert, was der Jens vorausgesagt hat: Mangels Masse war er nicht in der Lage, mehr als zwei INFOS in diesem Jahr herauszubringen. Nun ist auch der beste Redakteur nicht in der Lage, aus null Beiträgen ein INFO zu machen. Aber warum ist die früher so reichlich sprudelnde Quelle an Beiträgen versiegt? Liegt das vielleicht auch daran, daß sich viele Mitglieder heimlich oder unheimlich mit dem ach so geschmähten MS-DOS befassen und bis zur Stunde keine Basis in unserem Club 80 gefunden haben?

Warum nur ist dieses MS-DOS in unseren Reihen so verpönt? Kann man den nicht auch unter diesem Betriebssystem in Assembler oder den verschiedensten Hochsprachen programmieren? Ist das das DOS so ausgereift, daß es daran nicht zu verbessern gibt? Und kann man nicht auch an diesen Rechnern schrauben und löten?

Meiner Meinung nach befinden wir uns an einem Scheideweg. Machen wir weiter so wie bisher, wird der Club 80 bald sanft entschlafen sein. Es möge daher jedes Mitglied eindringlich mit sich zu Rate gehen, ob wir uns nicht den Realitäten beugen und MS-DOS in unseren Reihen zulassen sollen.

Ich würde eine Öffnung sehr begrüßen. Es wäre schade, wenn eine Einrichtung wie der Club 80 so sang- und klanglos untergehen würde. Aber ich bin zuversichtlich, daß wir auch diese Kurve kriegen und den Club 80 zu seiner alten Vitalität zurückführen können.

"In den Sternen steht's...

...nicht geschrieben", was aus dem CLUB 80 wird!
Oder haben wir auch einen Astrologen unter uns? Der verzeihe mir meine "Ungläubigkeit" und erstelle uns auch sogleich ein Horoskop, bitte! Ist es positiv, so wollen wir daran glauben. Andernfalls nicht.
Auch ohne der Astrologie zu frönen, hatte ich im INFO 28 ein "Horoskop" veröffentlicht. Ohne "Tierkreiszeichen" und ähnliche Mätzchen - Verzeihung: Mystik, sondern mit jener Mathematik, die man Extrapolation nennt. Extrapolationen sind - sofern überhaupt zulässig - ihrem Wesen nach stets ungenau. So sind es bis heute (wir schreiben im Moment den 6.11.1990) zwar nicht, wie von mir extrapoliert, nur zwei, aber doch bloß ganze drei INFOs in 1990, also genau so wenig wie in 1989!
Gewiß kommt es nicht auf die Anzahl, also die Quantität, sondern auf die Qualität an; dennoch:
Erinnert sei an den eindrucksvollen Bericht des holländischen TRS-80-Users *Joop Groenendijk*, Sekretär der "TRS-80 Gebruikers Vereniging", im INFO 21 (bereits September 1987!) Bei dem klangvollen "Grünen Deich" sonnten sich 2000 (zweitausend!) Z80-Anhänger! Und früher waren es sogar noch mehr! Dagegen sind wir ein klägliches Häuflein von z.Z. 63. Und über noch mehr km² verstreut als die Holländer!
Was können wir von *Joop* lernen?
Sinngemäßes Zitat: "Diese Zahl (2000) steigt z.Z. wieder, seit wir den Entschluß gefaßt haben, daß wir uns auch auf den Gebrauch des PC einrichten!" (Hört hört!)
In ähnlichem Sinn äußerte sich *Paul-Jürgen Schmitz* auf der nächsten Seite jenes INFOs.

Nachdem ich nochmal alle INFOs auf die Früchte von (Hard- und Software-)Bastlerfleiß abgeklopft und diese Früchte (=Arbeitsergebnisse, besser: Hobby-Ergebnisse) mit der Leistungsfähigkeit meines neuen, erweiterten XT's von EPSON (mit Festplatte und - pfui! - MS-DOS) verglichen hatte, ging mir ein Licht auf, worauf die Treue zum TRS-80 bzw. GENIE beruht: Es ist das Vergnügen, mit eigenen (!) Einfällen und Anstrengungen in diese Geräte das hineinzupacken und aus ihnen das herauszuholen, was die "Kompatiblen" (die "IBM-Diener"), die nun unwiderruflich en vogue sind, mit Leichtig- und Selbstverständlichkeit an Leistung und Komfort von Haus aus mitbringen!!

Dieses Vergnügen kann in Frust enden - oder gelingen; je nach Geschick und Geschicklichkeit. Maßstab für den Erfolg dieses Verg- ← *die „Trenn-Automatik"!* nügens bzw. Bemühens, wie gut das Ziel erreicht wurde, wird stets und in Zukunft immer stärker der "Kompatible" sein. An dem müssen wir unsere Künste und deren Früchte messen lassen.

"Messen" aber heißt "Vergleichen"!
(Hier spricht ausnahmsweise mal der Physiker.)

Sonst schweben wir im luftleeren Raum - und somit "entschwebt" uns einer nach dem anderen... und früher oder später sind es nur noch 6 statt 60. (Gewiß, auch das ist noch eine hübsche Zahl. Aber diese Sechs hocken ohnehin seit eh und je eng zusammen.)

Kurzum:
Ihr "Halbaussteiger" (das ist wohl die Hälfte aller), schreibt über eure Erfahrungen mit MS-DOS und Zubehör, vergleicht (z.B. Programme, die ihr für beide Systeme geschrieben habt; Utilities beider oder auch mehrerer Systeme - etwa nebst CP/M - u.a.) und zieht Schlüsse daraus, die für uns alle von Nutzen sind!

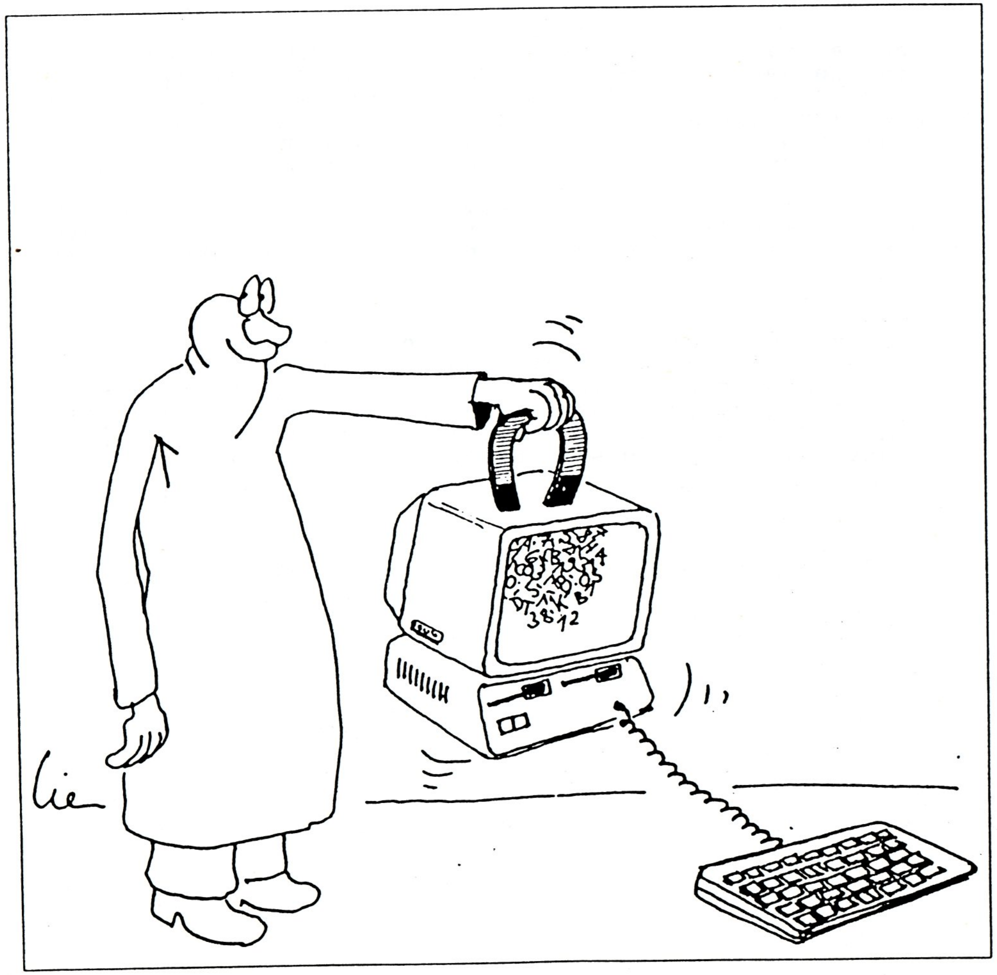
übrigens: INFO Nr.31, Seite 6 ("Nachbemerkung" von Arnulf) läßt hoffen! Denn:

EX ORIENTE LUX!

Das Tolle dabei:
"Oriente" ist diesmal nicht der Orient oder Nah- oder Fernost, sondern sehr nahes neues altes Deutschland!
Deren Freaks werden sich wohl bald auch die neuen Kisten mit MS-DOS anschaffen – es sei denn, ihr schenkt ihnen eure alten (und macht dann selbst mit den "Kompatiblen" weiter!)
Also heißt es dann sowieso:

Macht)
Schnell)
–)
Das Tor) ⟹ z.B.: MS-DOS!
Offen für andere)
Systeme!)

In diesem Sinn: Euer KH 7. XI. 90



Adventure

Schneller Booten

Wenn man das System, einfach so wie es ist, nur bis zum DOS-Prompt hochfährt, läßt sich das kaum beschleunigen, es dauert halt seine Zeit, bis die Systemspuren gelesen sind. Wenn aber anschließend per Autostart noch einige Programme für diverse Einstellungen aufgerufen werden, läßt sich einiges tun. Wenn der Rechner beim Booten ewig auf der Floppy rumsägt und von Anschlag zu Anschlag hetzt, ist es dafür allerhöchste Zeit. Beim NEWDOS/GDOS kann man leider nicht viel an der Dateiorganisation drehen, aber beim CP/M ist es so, daß Files immer schön aufsteigend in die nächsten freien Sektoren hinter dem Directory kopiert werden. Der Trick ist nun, daß man die beim Booten benötigten Programme als erste auf eine leere Floppy kopiert. Dann hat man die Bootspur(en) ganz außen, anschließend kommt das Directory und gleich dahinter die Startup-Files. So muß der Kopf nur über höchstens 4 bis 5 Spuren bewegt werden und es geht auf einmal viel schneller und leiser.

Alexander Schmid

Fehler in EASE.COM

EASE ist zur Zeit mein liebster Error-Handler, weil er zwei Features miteinander verbindet. Einmal kann man eine fehlerhafte Befehlszeile editieren (wie sich das für einen Error-Handler nunmal gehört) und zweitens kann man vergangene Eingaben zurückholen, er ist also Error-Handler und History-Shell in einem. Wenn man will, zeigt er einem mit einem Patch auch zusätzlich noch die aktuelle Uhrzeit an.
Nur einen kleinen Fehler hat das Ding. Wenn man einen Befehl aus dem RCP mit einem Parameter (z.B. P 200) aufruft wird jedesmal das Byte an der Adresse 100 gelöscht. Das ist sehr ärgerlich, weil dort ausgerechnet ein Sprung steht (stehen muß) und so der Restart-Trick mit der leeren Datei bzw. das GO nicht mehr funktionieren kann. Auch nach SAVE wird man sich wundern, daß das gerettete Programm nicht mehr läuft.
Statt nun dort jedesmal das 'C3' von Hand wieder einzufügen sollte man lieber gleich EASE patchen, sodaß das ein für allemal aufhört. Die gesuchte Stelle ist bei 0D4DH (Version 2.0), wo ein LD BC,0080 steht, womit der Buffer von 80H bis FFH gelöscht werden soll. Für diese Aktion reicht aber ein LD BC,007F und das Byte bei 100 bleibt anschließend verschont.

Alexander Schmid

Die RS232 des Model 4p

Bei der Anpassung von Terminalprogrammen an den eigenen Rechner stellen sich immer wieder dieselben Fragen, nämlich auf welche Ports man wie zugreifen muß und welche Bits man braucht. Am besten macht man sich einen kleinen Zettel und legt oder klebt ihn ins Handbuch, weil man das sonst mit Sicherheit bis zum nächsten mal wieder vergessen hat.

Für das Model 4p sieht das folgendermaßen aus:

Stellt fest, ob ein Byte zum Abholen bereit steht:
```
INPSTAT   IN   A,(0EAh)
          AND  80h
          RET
```

Liest ein Byte von der seriellen Schnittstelle ein:
```
READ      IN   A,(0EBh)
          RET
```

Stellt fest, ob ein Byte gesendet werden kann:
```
OUTPSTAT  IN   A,(0EAh)
          AND  40h
          RET
```

Gibt ein Byte an die serielle Schnittstelle aus:
```
WRITE     OUT  (0EBh),A
          RET
```

Einige Programme erwarten auch, daß die Leseroutine erst dann RETurned, wenn ein Byte eingelesen wurde. Das geht dann so:

```
READ      IN   A,(0EAh)
          AND  80h
          JR   Z,READ
          IN   A,(0EBh)
          RET
```

Die Baudrate kann man folgerndermaßen einstellen:

```
BAUD      LD   A,xx
          OUT  (0E9h),A
          RET
```

Die Initialisierung (Anzahl der Datenbits, Parity usw.) erfolgt zumindest beim CP/M beim Booten, sodaß diese kurze Routine, wenigstens dann, als ich es auf Hartmuts Rechner ausprobiert hatte, ausreichend ist. Der Wert von xx steht im Handbuch und ist je nach gewünschter Baudrate einzusetzen. Einige Werte sind FFh für 19200 Baud, EEh für 9600, CCh für 4800, AAh für 2400, 77h für 1200, 66h für 600 und 55h für 300 Baud.

Alexander Schmid

28.10.90

Hallo Clubkameraden,

Euer Diskothekar hat seit unserem letzten Treffen von Eurer Seite nichts mehr zu tun bekommen. Liegt es daran, daβ ich keine Programme vorgestellt habe, oder unsere Zeitschrift nur noch sehr selten erscheint, oder haben alle ihren Z80 Rechner verkauft und machen jetzt in MS-DOS oder TOS? Zu meiner Entschuldigung möchte ich vorbringen, daβ ich seit Juli stolzer Vater bin und vor kurzem (vielleicht auch schon länger??) habe ich mir, aus geschäftlichen Gründen auch so eine Maschine zugelegt. Das heiβt aber nicht, daβ ich nichts mehr mit NEW_DOS zu tun haben will. Schon von meinem Brötchengeber her muβ ich, was Betriebssysteme angeht, sehr flexibel sein. Da ich MS-DOS, VMS (auf unserem Groβrechner), RT11 und RSX (auf unseren Prozessrechnern) fahren darf.

Ich will Euch heute zeigen wie man einen sauberen Bildschirmaufbau (ansprechende Masken) ohne sehr viel Aufwand in verschiedenen Programmiersprachen realisieren kann. Als erstes das Prinzip. Wie Ihr sicher alle wiβt läβt sich der Bildschirm in Zeilen und Spalten unterteilen. Ihr könnt sogar jede einzelne Position lesen und schreiben, was bei meinem Programm aber nicht nötig ist. Die Sache läβt sich in zwei Abschnitte einteilen.
Man nehme irgend einen Editor und gestalte sich sein Menü. Einzige Anforderung, Ihr müsst für jede Zeile ein Zeilenendezeichen (CR) stehen haben. Wie Ihr seht alles ganz einfach. Es sind sogar Sonderzeichen erlaubt die man sonst nur über CHR$() auf den Monitor bekommt.
Jetzt kommt der Trick, wie vorher schon angedeutet, geht einfach davon aus der Monitor sehe aus wie ein Vektor (Array,Feld) oder wie man es sonst noch nennen will. Jede Zeile ist ein Element dieses Vektors und die Dimension hängt nur von der Zeilenzahl ab die Ihr verwendet. Auf jeden Fall dimensioniert man ein Feld DIM MASKE$(16) für 16 Zeilen. Dieses Feld ist bei mir z.B. maximal 16 * 64 = 924 Byte groβ. 16 Zeilen mit maximal 64 Zeichen pro Zeile und jedes Zeichen braucht 1 Byte Platz. Wenn man jetzt viele Masken in einem Programm hat kann es schon zu Speicherknappheit kommen, wesshalb es zwei Wege gibt. Entweder schnelles Maskenwechseln und viel Speicherplatzverbrauch oder wenig Speicherplatzverbrauch und lange Wartezeiten bis von der Diskette die Maske nachgeladen ist.

Die Beschreibung für eine Maske im schnellen Verfahren.

Sprung ins Unterprogramm fürs Maskenladen. Hier öffnet man den Maskenfile als sequentielle Datei. Nun folgt eine FOR NEXT - Schleife die den File Zeile für Zeile in unseren Vektor MASKE$(I) ladet, I ist der Schleifenzähler. Wie oft die Schleife durchlaufen wird hängt von der Anzahl der Zeilen ab. So jetzt sind die Daten (die Maske) im Speicher und wenn Ihr sie braucht gebt sie einfach mit PRINT und einer FOR NEXT - Schleife aus, aber vergesst nicht beim letzten PRINT den Zeilenvorschub zu unterbinden sonst ist die Maske oben mit der ersten Zeile schon wieder rausgerutscht. Solltet Ihr mehere Masken brauchen, dann müsst Ihr das obige für jede Maske einmal durchführen.
Ist der Speicher knapp so kann man das Unterprogramm immer dann aufrufen, wenn die Maske gebraucht wird. Wenn viel Platz vorhanden ist ladet man die Maske(n) am Anfang in den Speicher und hat sie anschlieβend alle jederzeit zur Verfügung. Man muβ sie dann nur noch ausgeben.

Im Folgenden zeige ich Euch wie so etwas in BASIC und FORTRAN aussieht. Beim BASIC verwende ich die Befehle von GENIE-BASIC beim FORTRAN die von MS-FORTRAN unter MS-DOS. Übrigens vielleicht hat einer Unserer Assemblerspezialisten Lust die Ausgabe auf den Bildschirm in Assembler zu schreiben, dann läuft alles noch schneller.

Datei einlesen:

```
10 DIM MASKE$(16)
20 OPEN"I",1,"MASKE1.DAT:1"
30 FOR I=1 TO 16
40    LINEINPUT#1,MASKE$(I)
50 NEXT I
60 CLOSE 1
```

Maske auf den Bildschirm ausgeben:

```
1000 FOR I = 1 TO 15
1010    PRINT MASKE$(I)
1020 NEXT I
1030 PRINT MASKE(16);            vergesst das Semikolon nicht!!!!!
```

Die obigen Routinen könnt Ihr zum Beispiel im Editorprogramm EDITOR1/BAS auf Diskette 2 verwenden das von mir kommentiert wurde und jetzt EDITOR/BAS heiβt.

Nun das ganze in MS-FORTRAN

Als erstes die notwendige Definitionen

```
      CHARACTER MASKE(24)*80     das Feld MASKE mit 24 Elementen a 80 Zeichen
                                 definieren
```

Den File einlesen

```
      OPEN(10,FILE='MASKE.001')  Fiele öffnen
      REWIND 10                  Zurückspulen, den Zeilenzähler natürlich
      READ(10,'(A)')MASKE        den ganzen File in den Vektor
      CLOSE(10)                  den File schlieβen nicht vergessen
```

Maske auf den Bildschirm

```
      WRITE(*,'(24A\)')MASKE     den Vektor auf den Bildschirm ausgeben
```

Cursor positionieren und numerische Variable I einlesen

```
      CALL CURPOS (1,2)          Cursor positionieren auf Zeile 1, Spalte 2
      READ(*,'(I1)')I            I einlesen
```

Das Ganze funktioniert mit dem Treiber Namens ANSI.SYS dem eine entsprechende Steuersequenz geschickt wird.

```
      SUBROUTINE CURPOS (Z,SP)
*******  AUFRUF : CALL POS (1,10)
*******     POSITIONIEREN AUF DEM BILDSCHIRM Z= ZEILE SP = SPALTE ***********
*******     ERSTES ZEICHEN IM HAUPTPROGRAMM IST KEIN STEUERZEICHEN MEHR *****
*******     Z = 1-25              SP = 1-80
      INTEGER Z,SP
      WRITE (*,'(1X,A,I2.2,A,I2.2,A)') ' [',Z,';',SP,'H'
      END
```

Wie Ihr seht ist es ganz einfach einen ansprechenden Bildschirmaufbau zu machen ohne, daβ man sich mit PRINT und einer Menge Blanks herumschlägt.

WINDOW.NDR
eine Umsetzung von WINDOW.BIB
aus Chip Spezial Turbo-Pascal Nr. 3
an die speziellen Moeglichkeiten von FlomonCG

Mitte 1989 habe ich die allgemeine Version von WINDOW.BIB an den NKC angepasst. Kurz darauf lief dann auch FlomonCG auf meinem Rechner. Es erwachte fast sofort der Wunsch, die Fenstertechnik auf Monitorebene zu nutzen. Da in der Reihe Turbo-Spezial von Chip viele gute Programme die in Heft 3 veroeffentlichte Bibliothek WINDOW.BIB verwenden, sollte die neue Bibliothek natuerlich kompatibel dazu sein. Lediglich die kostbare TPA muss entlastet werden. (WINDOW.BIB speichert alle Ausgaben in ein Fenster in der TPA. Der Speicher wird da recht schnell knapp.) Das Ergebnis ist ein kurzes Programm in Turbo-Pascal. Dieses Programm nutzt nicht alle Moeglichkeiten von FlomonCG. Hier ist (in Grenzen) noch Raum fuer eigene Kreativitaet. Die vorhandenen Prozeduren sollten jedoch nicht in ihrer Funktion veraendert werden, um die Kompatibilitaet zu erhalten.

Allgemeine Hinweise

WINDOW.BIB in der allgemeinen Version erwartet verschieden Parameter (z. B. Zeichen fuer Rand, Anzahl Zeilen etc). Diese Parameter waren in der Datei WINDOW.PAR gespeichert. Diese wird, wie auch WINDOW.BIB, vom Quellprogramm mit der Include-Anweisung (z. B. {$I WINDOW.BIB}) geladen. Die von Chip veroeffentlichten Programme erwarten eine Datei mit diesem Namen. Die Parameter sind fuer WINDOW.NDR ohne Bedeutung. Die Datei kann also leer sein. Natuerlich kann auch die entsprechen Include-Anweisung im Quelltext geloescht werden. Da dies meist vergessen wird, halte ich eine leere Datei mit dem Namen WINDOW.PAR auf meiner Bibliotheks-Diskette bereit.

WINDOW.NDR arbeitet nach meinen bisherigen Beobachtungen mit allen bisher von Chip veroeffentlichten Programmen. Auch die Bibliotheken SELECT.BIB, CHOOSE.BIB etc. koennen benutzt werden. Inwieweit die Maus funktioniert, konnte ich mangels Maus bisher nicht testen.

Die einzelnen Prozeduren

InitWindows

Parameter: keine
Aufruf nicht erforderlich. Prozedur wurde nur der kompatibilitaet halber eingefuegt. Prozedur wird beim ersten Aufruf von OpenWindow mit aufgerufen, wenn 'Initialisiert' nicht TRUE ist.

ExitWindows

Parameter: keine
Alle Fenster werden geschlossen. Initialisiert wird auf FALSE gesetzt.

OpenWindow

Parameter: x1,y1,x2,y2
Oeffnet ein Fenster mit den Koordinaten
x1,y1 = Ecke links oben und
x2,y2 = Ecke rechts unten.

CloseWindow

Parameter: keine
Schliesst das letzte Fenster.

SelWindow

Parameter: Nr
Das Fenster mit der ID 'Nr' wird zum aktuellen Fenster. Alle Ausgaben erfolgen jetzt in diesem Fenster. Das Fenster wird, wenn erforderlich, in den Vordergrund geholt. In der allgemeinen Bibliothek bezieht sich diese Prozedur nur auf sich ueberlappende Fenster. (Wer kompatible Programme schreiben will, sollte das beachten!)

ChangeWindow

Parameter: Nr
Funktion wie SelWindow. In der allgemeinen Version bezog sich diese Prozedur nur auf nicht ueberlappende Fenster. Wer kompatible Programme schreiben will, die dann auch problemlos auf anderen Rechnern (nicht NDR) laufen, sollte diesen Unterschied streng beachten. Auf dem NKC funktionieren beide Prozeduren gleich. ChangeWindow ruft hier lediglich SelWindow auf.

Funktionen

WhereX

Parameter: keine
Rueckgabe: Integer
Ermittelt die Spalte, in der der Cursor steht.

WhereY

Parameter: keine
Rueckgabe: Integer
Ermittelt die Zeile, in der der Cursor steht.

Globale Variablen

Initialisiert

Typ: Boolean
Gibt an, ob Fenstertechnik initialisiert wurde. Fuer FlomonCG ist das nicht wichtig, aber in der allgemeinen Version geht ohne initialisierung gar nichts.

Inverse

Typ: Boolean
Wenn Inverse = True, dann erfolgt die Ausgabe in inverser Darstellung.

ScreenPtr

Typ: Byte
Gibt die ID des gerade aktiven Fensters an.

MaxScreen

Typ: Byte
Enthaelt die Anzahl der geoeffneten Fenster.

Zum Schluss noch einige Bemerkungen

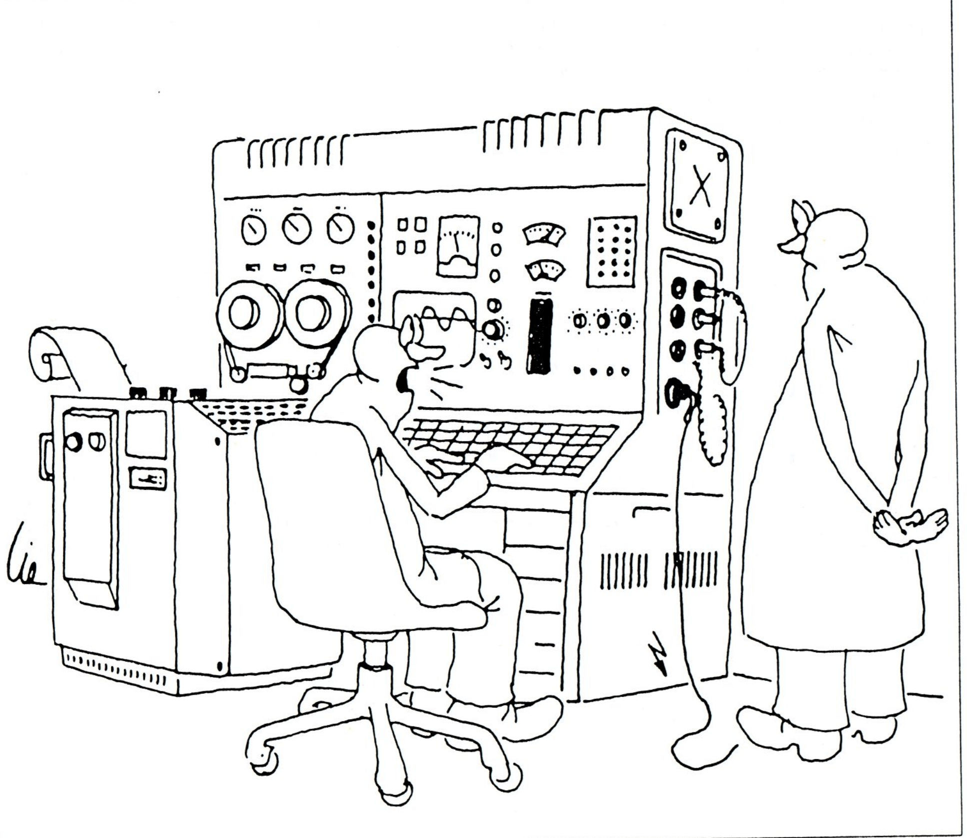
Ich moechte nochmals davon Abraten, die Bibliothek voll an die vorhanden Moeglichkeiten des FlomonCG anzupassen. Mit Sicherheit kann das Ganze einfacher und eleganter geloest werden. Die Verwendung der vorliegenden Bibliothek hat aber den grossen Vorteil, dass es eine auf jedem Rechner arbeitende Version gibt. Programme, die diese Bibliothek verwenden, laufen also auf jedem

Rechner. Sie muessen lediglich unter der Verwendung von WINDOW.BIB und WINDOW.PAR neu kompiliert werden. Es ist lediglich darauf zu achten, dass in der TPA ausreichend Platz zum speichern der Fenster ist. Da in der Regel aber nur sehr wenige Fenster gleichzeitig geoeffnet sind,ist das meistens kein Problem.

Wer sich das Eintippen sparen moechte, kann mir auch eine formatierte Diskette (5 1/4 oder 3 1/2) schicken. Aber bitte ausreichend Rueckporto beifuegen. Ueber Anregungen und Beispielprogramme wuerde ich mich natuerlich auch freuen.

Ich habe in diesem Artikel uebrigen bewusst auf Umlaute etc. verzichtet, da mir fuer BRADFORD noch keine Deutschen Zeichensaetze zur Verfuegung stehen. Da ich aber die ausserordentlichen Faehigkeiten dieses Druckprogramms nutzen wollte, um das ganze lesbarer zu gestalten, musste der normale ASCII-Zeichensatz genuegen. Naeheres zu diesem (und weiteren interessanten) Programmen soll in weiteren Artikeln folgen.

Ausgabe
»Die Computer-Diagnose ergab, daß wir uns gar keinen Computer mehr leisten können.«



Und nun endlich das Programm

```
*******************************************************************
* Bibliotheks-Modul: WINDOW.NDR*
**
* Spezielle Anpassung von WINDOW.BIB aus Chip Spezial Turbo-Pascal
* an den NDR-Klein-Computer mit FlomonCG und GDP64HS. Ob das ganze
* auch mit der alten GDP64 funktioniert, wurde bisher nicht getest
* Die Urversions von Chip benoetigt noch die Include-Datei WINDOW.
* Diese Datei kann hier leer sein oder fehlen. Um Aenderungen an*
* bestehenden Programmen so gering wie moeglich zu halten, wird *
* empfohlen, diese Datei zu erstellen.*
**
* Das Modul liefert die folgenden Bezeichner und Routinen:*
**
* p InitWindowsurspruenglich zur Initialisierung gedacht*
*bei FlomonCG leere Procedure*
* p ExitWindowwie InitWindows*
* p OpenWindow (x1,y1,x2,y2)*
* oeffnet ein Fenster von (x1,y1) = LinksOben*
*bis (x2,y2) = RechtsUnten*
* p CloseWindowschliesst das letzte Fenster*
* p SelWindow (NR)Holt das Fenster (Nr) nach von und macht es*
*aktiv (in der Urversion fuer ueberlappende *
*Fenster gedacht)*
* p ChangeWindow (Nr) macht Fenster (Nr) aktiv; Aktion wird von*
*SelWindow erledigt. Ist erforderlich, um alte*
*Programme nicht aendern zu muessen. In der*
*Urversion war die Procedure fuer nicht*
*ueberlappende Fenster zustaendig. Fuer *
*FlomonCG gibt es da keinen Unterschied.*
* v inverse : boolean gibt an, ob gerade LowVideo gilt.*
* v ScreenPtr : bytegibt die Nr. des gerade aktiven Fensters *
* v initialisiert : boolean*
*gibt an, ob InitWindows aufgerufen wurde. Hat*
*eigentlich keine Bedeutung mehr. Da aber alte*
*Programme dies evtl. abfragen, wird das Flag*
*entsprechend gesetzt.*
* v MaxScreen : byteist die Anzahl der geoeffneten Fenster*
* f WhereX : integerCurser-Spalte*
* f WhereY : integerCurser-Zeile*
* *
* Neu definiert wurden noch die Proceduren NormVideo und HighVideo
* Bei korrekter Installation von Turbo Pascal waere auch dies nich
* erforderlicht. Aber wer hat schon immer alles korrekt installier
**
* Das urspruengliche Modul geht auf eine Veroeffentlichung von *
* Chip (Turbo-Pascal Spezial Nr. 3) zurueck. Es wurden jedoch *
* lediglich die Bezeichner der Proceduren und Variablen uebernomme
* um mit diesem Modul kompatibel zu bereits geschriebener Software
* zu bleiben. Das Copyright dieses Moduls liegt bei mir. Die *
* Verwendung und Verbreitung wird zu den ueblichen Public-Domain-*
* Bedingungen gestattet. Die gewerbliche Nutzung (was immer das au
* sein mag) ist nur mit meiner ausdruecklichen Genehmigung gestatt
* Auch eine Veroeffentlichung ist vorher mit mir abzustimmen (ich
* erwarte in der Regel die Zusendung eines Belegexemplares).*
**
* Dies ist die Version 1.0 von 11/89*
**
* Fehlerhinweise, Verbesserungen und Fragen bitte an:*
**
```

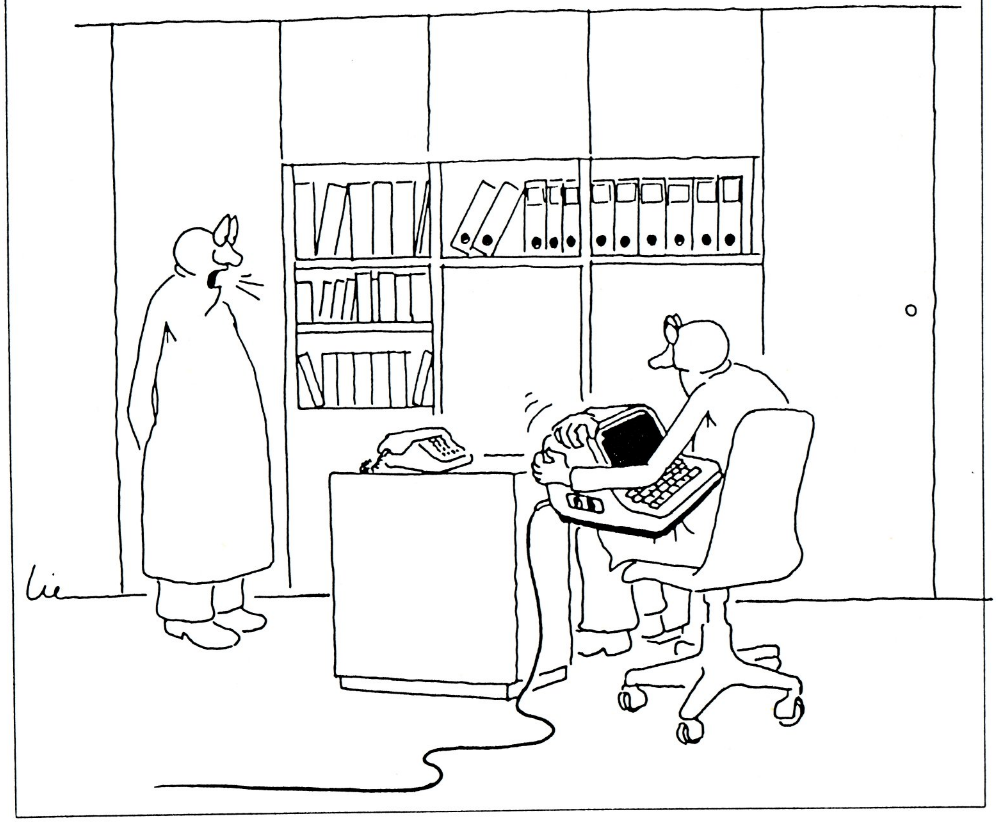
```
 * Claus Littmann, Plockhorst, Zum Spring 15, 3155 Edemissen*
 * Tel. 05372-7796 (bitte nicht nach 21.00!!)*
 ***************************************************************

Const   ScreenPtr       : Byte = 0;    { das momentan aktive Fenster }
        MaxScreen       : Byte = 0;    { Anzahl der geoeffneten Fenster
        Initialisiert   : Boolean = False; { wg. Kompatibilitaet }
        Inverse         : Boolean = False; { Flag wg. Inverser Darstel

Procedure InitWindows;         { nur wg. kompatibilitaet noetig }

Begin   Initialisiert := True;  End;

Procedure ExitWindows;  (alle Fenster schliessen, Fenster 0 dann akt

Begin
   Writeln (#27,'$@');{ Fenster normieren = alle Fenster schliessen,
  Fenster 0 mit Standardgroesse oeffnen }
   ScreenPtr := 0; MaxScreen := 0; { alle Pointer auf 0 setzen)
   Initialisiert := False;    { der Komptibilitaet wegen }
End;

Function WhereX : Integer; { Wo steht der Curser }

Var St1   : String[2];

Begin
   Write (#27,'?'); Read (Kbd,St1); { Curser mit Flomon abfragen }
   WhereX := Ord(Copy(St1,2,1))-32; { Wert fuer x-Koordinate normier
End;

Function WhereY : Integer;      { Wie WhereX, nur fuer Y-Koordinate }

Var St1   : String[2];

Begin
   Write (#27,'?'); Read (Kbd,St1);
   WhereY := Ord(Copy(St1,1,1))-32;
End;

Procedure NormVideo;    { Darstellung mit normalen Zeichen }

Begin
   Write (#27,'(');   { Flomon-Befehl Invers aus }
   Inverse := False; { Flag setzen, damit man weiss, was los ist }
End;

Procedure LowVideo;{ LowVideo hier als Invers interpretiert }

Begin
   Write (#27,')');{ Flomon-Befehl Invers an }
   Inverse := True;{ Und wieder das Flag setzen }
End;

Procedure HighVideo;{ = NormVideo, da wir sowas nicht haben }

Begin NormVideo End;
```



```
Procedure SelWindow (x : Byte);{ Ein Fenster anwaehlen }

{  Das Fenster x wird in den Vordergrund gebracht; alle A
erfolgen  jetzt  in diesem Fenster. Fenster muss  definiert
Ausserdem muss das Flag 'initialisiert' = TRUE sein. Dies
FlomonCG   zwar   nicht   erforderlicht,   aber   in   der   allg
Version     dieser     Bibliothek     sehr     wichtig,     da     in
Initialisierungsprocedure  die Ausgaberoutiene  von  Turbo
auf  eine eigene Routiene umgebogen wird. Um alle Programm
diese Bibliothek verwenden, ohne Aenderungen weiter verwen
koennen,  wurde dieses hier eigentlich nicht  erforderlich
gesetzt. }

Begin
   If Not initialisiert Then Halt;
   If X <= MaxScreen Then
      Begin
         Write (#27,'$S',x);{ Fenster in den Vordergrund }
         Writeln;
         GotoXY(0,0);{ Curser setzen }
         ScreenPtr := x;{ Fenster ID aktualisieren }
      End;
End;

Procedure ChangeWindow (x : Byte);    { wg. kompatibilitaet }

{   Diese Procedure war urspruenglich erforderlich,  d
allgemeine Version einen Unterschied zwischen hintereinand
nebeneinander  liegenden  Fenstern  machte.  Um  Aenderung
bestehenden  Programmen unoetig zu machen, wurde diese  Pr
erhalten. Die Arbeit macht aber SelWindow. }

Begin
   SelWindow(x);
End;

Procedure OpenWindow (a1,b1,a2,b2 : Byte);

{ Hier wird ein Fenster geoeffnet. Die Koordinaten sind :
  a1,b1 = links oben bis a2,b2 = rechts unten. Diese Angab
  werden auf das Flomon-Format umgerechnet. }

Var  Zeilen,Spalten,
     Offx,Offy        : Char;
     ID               : Integer;

Begin
   If Not initialisiert Then InitWindows;
   Spalten := Chr(32+(a2-a1)); Zeilen := Chr((b2-b1)+32);
   Offx   := Chr(a1+32);       Offy   := Chr(b1+32);
   ScreenPtr := Succ(ScreenPtr);  { enthaelt die ID des Fenstern }
   ID := ScreenPtr;
   Write (#27,'$O',ID,Zeilen,Spalten,Offy,Offx); { Fenster oeffnen }
   Writeln (#27,'$E');       { und sichtbar machen }
   MaxScreen := ScreenPtr    { Anzahl der geoeffneten Fenster }
End;

Procedure CloseWindow;

{ Mit dieser Procedure wird das letzte geoeffnete Fenster
  geschlossen. }
```

```
Begin
     If (ScreenPtr=0) Or Not initialisiert Then Halt;
                                          { kein Fenster offen }
     Writeln (#27,'$C');  { letztes Fenster schliessen }
     ScreenPtr := Pred(ScreenPtr); { Zeiger berichtigen }
     MaxScreen := ScreenPtr:  { Anzahl der Fenster berichtigen }
End;
```

Büro
»Sie müssen einfach vergessen, Meier, daß Sie vor der Rationalisierung ein herzliches Verhältnis zu einer Sekretärin hatten.«



*** Umlaute in *LeScript 1.65* ***

Wer von Euch mit *LeScript* (Vers. 1.65) seine Texte schreibt, wird das Problem sicher kennen: Das kleine "ö" (ö!) läßt sich auf dem Bildschirm nicht darstellen, da es als Trennungszeichen zwischen den einzelnen Buchstaben im Breitschriftmodus (CLEAR-X) verwendet wird. überall dort, wo *LeScript* im Text ein "ö" vorfindet, schaltet der Druckertreiber auf gedehnte Schrift. Dies hat auch den Nachteil, daß der Text, der in Breitschrift geschrieben wurde, fast nicht mehr zu erkennen ist:

öBöröeöiötösöcöhöröiöföt ö(öCöTöRöLö-öXö)

Durch die unten aufgeführten Änderungen wird die Tastatur ein wenig anders belegt, so, daß nun auch Umlaute erreichbar sind. Außerdem wird das "ö" im CLEAR-X-Modus durch ein Grafikzeichen ersetzt. Hierfür habe ich den Grafikblock B0h (176d) gewählt, der - meiner Meinung nach - den Text am Besten in Breitschrift simuliert. Die Einstellung läßt sich natürlich beliebig ändern, bis auf einige Ausnahmen:

- 7Fh (127d) = "hartes Leerzeichen"
- 83h (131d) = Blockmarkierung Anfang/Ende
- 88h (136d) = Markierung Absatzende
- 8Ah (138d) = Einleitung eines Druckersteuerbefehls
- 8Ch (140d) = Satzendzeichen

Die oben genannten Codes also vermeiden, damit sie von *LeScript* nicht in die entsprechenden Kommandos umgesetzt werden.

Die Änderungen sind, z.B. mit SUPERZAP o. ä. , im Hauptmodul von *LeScript* vorzunehmen (LESCRIPT/CIM bzw. LESCRIPT/CMD).

1. Patch - Umbelegung der Tastatur für die Umlaute:

Sektor:	Byte:	Alt:	Neu:	Zeichen:
15	C2	1B	5B	"Ä"
15	C4	1C	5C	"Ö"
15	C6	1D	5D	"Ü"
15	C8	1E	5E	"^"
15	CA	1F	5F	"_"
15	CC	5E	7B	"ä"
15	CE	7E	7C	"ö"
15	D0	5B	7D	"ü"
15	D2	5D	7E	"ß"
15	D4	00	1B	ESCAPE
15	D8	7B	00	(Frei)
15	DA	5F	00	(Frei)
15	DC	7D	00	(Frei)
15	DE	5C	00	(Frei)

2. Patch - Grafikblock statt "ö" bei CLEAR-X:

Sektor:	Byte:	Alt:	Neu:	Zeichen:
32	F7	7C	B0	Grafik
69	FA	7C	B0	Grafik
70	06	7C	B0	Grafik

Die Tastatur ist nun folgendermaßen belegt:

<SHIFT>-<CLEAR> "1" = Ä
<SHIFT>-<CLEAR> "2" = ö
<SHIFT>-<CLEAR> "3" = ü
<SHIFT>-<CLEAR> "4" = ^
<SHIFT>-<CLEAR> "5" = _
<SHIFT>-<CLEAR> "6" = ä
<SHIFT>-<CLEAR> "7" = ö
<SHIFT>-<CLEAR> "8" = ü
<SHIFT>-<CLEAR> "9" = ß
<SHIFT>-<CLEAR> "*" = ESCAPE-Code (erscheint als "Ä")

Die Tasten <CLEAR>-<SHIFT>-",", <CLEAR>-<SHIFT>-"-", <CLEAR>-<SHIFT>-"." und <CLEAR>-<SHIFT>-"/" lassen sich frei belegen. (Sektor 15, Bytes D8, DA, DC und DE)

In den Sektoren 14 und 15 läßt sich mit einem Disketteneditor die gesamte Tastaturbelegung von *LeScript* ändern.
Einer "DIN"-ähnlichen Belegung steht also nichts im Wege ...

Deutsche Version von *LeScript* (mit DIN-Belegung) kann gegen eine formatierte Diskette und Rückporto bei mir angefordert werden.

Jens Günther

(PS: Suche außerdem neuere Version von *LeScript* (1.7 ? oder so)

Download noch einfacher

Wenn man seinem Drucker, sofern er softwaremäßig Zeichensätze laden kann, neue Zeichen beibringen will, kann einen das manchmal den letzten Nerv kosten. Bei 9-Nadel-Druckern geht es ja noch, aber bei 24ern wird's lästig. Eine Matrix mit 24 Zeilen und 32 Spalten auszuzählen macht nicht sehr viel Spaß und ist noch dazu einigermaßen fehlerträchtig. In einem CHIP-Special war zwar mal ein bildschirmorientierter Editor, aber der arbeitet stur mit 8x11 Punkten, die eben nur für einen 9-Nadler reichen. Die Lösung: in der c't 12/90 ist ein Programm, das mit beliebigen Formaten und also auch mit beliebigen Druckern arbeiten kann. Das Geheimnis liegt darin, daß man die Matrix und die notwendigen Steuerzeichen in ein Textfile schreibt, das dann vom Programm interpretiert wird. Ein Beispiel für so ein File könnte folgendermaßen aussehen:

```
File für DOWNLOAD.PAS aus der c't 12/90
Download-Zeichen für HP DeskJet mit FX-80-Emulation

#27#58#00#00#00              ROM ins RAM kopieren

#27#38#00#35#35#139          umdefiniertes Zeichen: #
.M 8 11 X                    Punkt-Befehl: 8 Zeilen, 11 Spalten
XXXXXXXXXXX
XXX..X..XXX
.....X.....
.XXXXXXXXX.
X.........X
X.........X
XXXXXXXXXXX
...........

#27#38#00#36#36#139          umdefiniertes Zeichen: $
.M 8 11 X
.....X.....
.....X.....
.XXXXXXXXX.
.....X.....
.....X.....
...........
.XXXXXXXXX.
...........

#27#37#01#00                 Zeichensatz im RAM aktivieren
```

Die Datei kann wirklich genauso aussehen, wie sie hier abgedruckt ist. Mit dem "Punktbefehl" M 8 11 X sagt man dem Programm, daß die Zeichenmatrix, die in der nächsten Zeile beginnt, 8 Zeilen und 11 Spalten hat und daß die gesetzten Punkte durch ein X repräsentiert werden. Alle anderen Zeichen werden ignoriert, daher kann man mit den Kommentaren sehr großzügig sein ! Die Zahlen hinter einem # werden als ASCII-Werte betrachtet und als ein Zeichen an den Drucker geschickt. So könnte man auch, ohne ein neues Zeichen zu defi-

nieren, einfach nur ein paar Steuerzeichen (Initialisierung, Fett, Doppeldruck...) an den Drucker schicken.
Zu beachten wäre noch, daß der "echte" FX-80 (und auch andere Nadeldrucker) aufgrund der langsamen Mechanik keine zwei direkt benachbarten Punkte drucken können. Die Matrix müßte dann so aussehen (bitte ausprobieren):

```
X.X.X.X.X.X
X.X..X..X.X
.....X.....
.X.X.X.X.X.
X.........X
X.........X
X.X.X.X.X.X
...........
```

Welche Steuerzeichen Euer Drucker im einzelnen braucht, müßt Ihr leider im Handbuch nachsehen. Auch in der c't ist das ganze noch etwas ausführlicher beschrieben. Wer nicht gerne tippt, kann das Programm gegen Rückporto von mir auf Floppy haben.

Noch eine Idee am Schluß. Thema: Artikelflut für's Info. Wenn die meisten sich nicht in der Lage sehen, großartige eigene Artikel über den ultimaten Patch im DOS oder sonstwas zu schreiben, wie wäre es denn mit Artikeln wie diesem ? Jeder wird doch wohl irgendeine Zeitung (vielleicht nicht gerade die BILD) lesen und den ein oder anderen Artikel finden, der von allgemeinem Nährwert ist. Pascal- und BASIC-Programme laufen, wenn sie nicht zu tief ins System einsteigen, auf fast jedem Rechner oder lassen sich zumindest relativ einfach anpassen.

```
(*****************************************************************)
(*                                                               *)
(* Dieses Programm wertet eine als Parameter übergebene Datei    *)
(* aus und schickt die darin enthaltenen Zeichen zum Drucker,    *)
(* um einen Sonderzeichensatz zu installieren.                   *)
(*                                                               *)
(* File          : DOWNLOAD.PAS                                  *)
(* Sprache       : TURBO PASCAL 3.0                              *)
(* Autor         : Heinz Hagemeyer                               *)
(*                                                               *)
(* (c) September 1990 Verlag Heinz Heise  Redaktion c't          *)
(*                                                               *)
(* c't 12/90, Seite 304                                          *)
(*                                                               *)
(*****************************************************************)

PROGRAM Download;

CONST Trennzeichen = '#';
      Blank        = ' ';
```



```
TYPE  Zeile          = STRING[255];
      Strg80         = STRING[80];

VAR   DateiName      : Strg80;
      Datei          : TEXT;
      Zeilennr       : INTEGER;
      Druckerfehler: BYTE;
      Code           : INTEGER;

PROCEDURE WriteXY(x,y:BYTE; was:Zeile);
BEGIN
  GotoXY(x,y);
  Write(was);
END;

PROCEDURE Error(t:Strg80; VAR x:TEXT; ZeilenNr:INTEGER);
VAR d:CHAR;
BEGIN
  WriteXY(36,2,'Fehler: '+#7);
  WriteXY((80-Length(t)) DIV 2,3,t);
  IF ZeilenNr>0 THEN
  BEGIN
    WriteXY(35,4,'Zeile: ');
    Write(ZeilenNr);
  END;
  (*$I-*)
  Close(x);
  (*$I+*)
  HALT;
END;

FUNCTION Umwandlung(VAR Datei:TEXT; DateiName:Strg80; x:Zeile;
                    ZeilenAnzahl:INTEGER):Zeile;
VAR zk:Zeile;
    p :INTEGER;

  FUNCTION ASCII(zk:Zeile):CHAR;
  VAR Fehler,Zahl:INTEGER;
  BEGIN
    Val(zk,Zahl,Fehler);
    IF (Fehler=0) AND (Zahl>=0) AND (Zahl<=255) THEN
      ASCII:=Chr(Zahl)
    ELSE
      Error('Zahl falsch in Datei '+DateiName,Datei,ZeilenAnzahl);
  END;

BEGIN
  p:=Pos(Blank,x);
  IF p>0 THEN
    x:=Copy(x,1,p-1);
  zk:='';
  REPEAT
    p:=Pos(Trennzeichen,x);
    IF p>1 THEN
```

```
      zk:=zk+ASCII(Copy(x,1,p-1))
    ELSE
      IF p=0 THEN
        zk:=zk+ASCII(x);
    x:=Copy(x,p+1,255);
  UNTIL (p=0) OR (x='');
  Umwandlung:=zk;
END;

PROCEDURE SendeZeichen(VAR Datei:TEXT; DateiName:Zeile;
                       ZeilenAnzahl:INTEGER);
CONST Ausdehnung=128;
      Punkt     ='.M';
VAR   i,Zeilen,Spalten:INTEGER;
      h,x             :Zeile;
      Eins            :CHAR;

  PROCEDURE Dimension(h:Zeile; VAR z,s:INTEGER; VAR Eins:CHAR);
  VAR i:BYTE;

    PROCEDURE GibZahl(VAR z:INTEGER);
    VAR Fehler:INTEGER;
        st    :STRING[3];
    BEGIN
      st:='';
      WHILE (h[i]<'0') OR (h[i]>'9') DO
        i:=i+1;
      WHILE (h[i]<='9') AND (h[i]>='0') DO
      BEGIN
        st:=st+h[i];
        i:=i+1;
      END;
      Val(st,z,Fehler);
      IF (z<1) OR (z>Ausdehnung) THEN
        Error('Dimension der Matrix falsch in '
              +DateiName,Datei,ZeilenAnzahl);
    END;

  BEGIN (* Dimension *)
    i:=1;
    h:=h+' ';
    GibZahl(z);
    GibZahl(s);
    WHILE h[i]=' ' DO
      i:=i+1;
    Eins:=h[i];
  END;

  PROCEDURE SendeMatrix(Nadeln,Spalten:BYTE; Eins:CHAR;
                        DateiName:Zeile);
  VAR i,ByteAnzahl,
      Anzahl        :BYTE;
      Zahl,Spalte   :BYTE;
      Matrix        :ARRAY [1..Ausdehnung] OF STRING[Ausdehnung];
```



```
  BEGIN
    ByteAnzahl:=Nadeln DIV 8;
    FOR i:=1 TO Nadeln DO
    BEGIN
      (*$I-*)
      ZeilenAnzahl:=ZeilenAnzahl+1;
      ReadLn(Datei,Matrix[i]);
      (*$I+*)
      IF IOResult<>0 THEN
        Error('Matrix in Datei '+DateiName+' fehlerhaft',
              Datei,ZeilenAnzahl);
    END;
    FOR Spalte:=1 TO Spalten DO
    BEGIN
      FOR Anzahl:=0 TO ByteAnzahl-1 DO
      BEGIN
        Zahl:=0;
        FOR i:=1 TO 8 DO
          IF Matrix[i+Anzahl*8][Spalte]=Eins THEN
            Zahl:=2*Zahl+1
          ELSE
            Zahl:=2*Zahl;
        (*$I-*)
        Write(Lst,Chr(Zahl));
        (*$I+*)
        IF IOResult<>0 THEN
          Error('Drucker nicht bereit',Datei,0);
      END;
    END;
  END;

  BEGIN (* SendeZeichen *)
    WHILE NOT EOF(Datei) DO
    BEGIN
      REPEAT
        ZeilenAnzahl:=ZeilenAnzahl+1;
        ReadLn(Datei,h)
      UNTIL (Pos(TrennZeichen,h)>0) OR EOF(Datei) OR
            (Pos(Punkt,h)=1);
      IF Pos(TrennZeichen,h)>0 THEN
        Write(Lst,Umwandlung(Datei,DateiName,h,ZeilenAnzahl))
      ELSE
        IF Pos(Punkt,h)=1 THEN
        BEGIN
          Dimension(h,Zeilen,Spalten,Eins);
          SendeMatrix(Zeilen,Spalten,Eins,DateiName);
        END;
    END;
  END;

BEGIN (* Main *)
  ClrScr;
  IF ParamCount=0 THEN
    Error('Falscher Aufruf, Dateiname fehlt',Datei,0)
```

```
    ELSE
    BEGIN
      DateiName:=ParamStr(1);
      Assign(Datei,DateiName);
      (*$I-*)
      Reset(Datei);
      (*$I+*)
      IF IOResult<>0 THEN
        Error(DateiName+' nicht gefunden',Datei,0);
      SendeZeichen(Datei,DateiName,ZeilenNr);
      Close(Datei);
    END;
  END.
```

Alexander Schmid

Brainstorming

After a long, long Time ...

Von CP/M nach MSDOS

Vor ein paar Wochen unterrichtete mich mein lieber Freund Kajot Mühlenbein von einem Unglück, an das zu denken jedem wirklichen Z8Ø-Freak den Angstschweiß auf die Stirne treibt. Nachdem sein, von Helmut Bernhard mühsam bis auf CP/M-Fähigkeit aufgebohrtes, Model 1 seinen Geist aufgegeben hat (das kann ja schon mal passieren und ist halb so wild) kaufte er sich, noch dazu zu einem unverschämt günstigen Preis, einen MSDOS-kompatiblen PC (was die eigentliche Katastrophe darstellt)!

Spätestens jetzt, vielleicht aber schon nach dem Entziffern des letzten Wortes der Überschrift, haben die oben angesprochenen orthodoxen Z8Ø-Freaks diese Seite überblättert und sind wahrscheinlich gerade dabei, Kajot's und meine Adresse aus ihren Adressbüchern, Computern und Gehirnen zu tilgen! All diejenigen, die sich bis hierher durchgerungen haben und noch dazu ein wenig an dBase interessiert sind, kann ich beruhigen: das folgende kann man durchaus auch dann sinnvoll verwenden, wenn man nicht zu den abtrünnigen Überläufern ins MSDOS-Lager gehört!

Nachdem die oben schon beschriebene Katastrophe ihren irreversiblen Lauf genommen hatte, bat mich Kajot, wohl wissend, daß auch ich seit geraumer Zeit nicht mehr mit ungetrübtem Gewissen den Z8Ø-Gurus unter die Augen treten darf, um einen Gefallen. Er hatte sich unter CP/M, dBase II benutzend, eine mehrere hundert Records (=Datensätze) große Datenbank angelegt. Da er sich die Arbeit ersparen wollte, alle Datensätze nochmals unter MSDOS einzutippen (die Z8Ø'er hätten es ihm wohl gegönnt), richtete er die Anfrage an mich, ob man die Daten wohl zwischen den Systemen transferieren könne. Die Antwort auf diese Frage erinnert ein wenig an Radio Eriwan: "Im Prinzip ja, aber ...".

Bei der Übertragung von Daten zwischen CP/M- und MSDOS-Systemen treten zunächst einmal zwei grundsätzliche Probleme auf:

1. MSDOS-Systeme können mit CP/M-Disketten (ohne spezielle Programme) nichts anfangen (umgekehrt natürlich auch nicht!)

2. die Zeichensätze unter CP/M und MSDOS unterscheiden sich bei den Umlauten, beim 'ß' sowie bei manchen Sonderzeichen (z.B. §).

Zusätzlich besteht bei der Übertragung von dBase-Dateien ein Unterschied zwischen den von dBase II und dBase III angelegten Datendateien.

Das Problem der unterschiedlichen Zeichensätze läßt sich, sowohl unter CP/M als auch unter MSDOS, durch recht einfache Programme in den verschiedensten Sprachen lösen. Wer nicht selbst zum Interpreter oder Compiler greifen will, kann von mir einen entsprechenden Turbo PASCAL-Quelltext bekommen.

Das Problem der physikalischen Übertragung der Daten ist da schon schwieriger zu knacken. Prinzipiell gibt es vier Lösungsansätze:
1. ein Programm, mit dem der CP/M-Rechner MSDOS-Disketten bearbeiten kann;
2. ein Programm, mit dem der MSDOS-Rechner CP/M-Disketten bearbeiten kann;
3. die serielle Übertragung der Daten zwischen CP/M- und MSDOS-Rechner;
4. die parallele Übertragung der Daten zwischen CP/M- und MSDOS-Rechner.

Wer glücklicher Besitzer eines TRS 8Ø Model 4/4p ist, hat es gut. Für diesen Rechner gibt es von Montezuma Micro/JOB ein Programm zum Bearbeiten von MSDOS-Disketten (DBLCROSS). Ganz nebenbei ermöglicht DBLCROSS auch noch den Datentransfer mit dem Model III-TRSDOS 1.3 und dem Model 4-TRSDOS 6.x. Weiterhin bekommt man für fast alle New-/TRSDOS-Versionen das Programm HyperCross, welches sowohl den Umgang mit CP/M- als auch mit MSDOS-Disketten beherrscht. Wer keinen Tandy-Rechner sein Eigen nennt, dem kann eventuell mit den Programmen TRANSFER oder PC-READ geholfen werden. Beide Programme liegen im Turbo PASCAL-Quellcode vor und sind Public Domain.

Unter MSDOS gibt es eine recht große Anzahl von Programmen, mit denen man CP/M-Disketten lesen kann. Leider findet man nur selten eines, welches gerade das eigene, meist recht exotische CP/M-Format beherrscht. Noch seltener sind solche Programme, bei denen man die Liste der lesbaren CP/M-Formate ändern/erweitern kann. Zu diesem Thema will ich mich aber an dieser Stelle nicht weiter auslassen, eventuell gibt es dazu demnächst mal einen gesonderten Artikel.

Auch die serielle Übertragung von Daten ist eigentlich relativ einfach zu bewerkstelligen. Außer einer seriellen Schnittstelle, über welche aber die meisten CP/M-Rechner verfügen, braucht man dazu nur zwei Terminalprogramme (jeweils eines unter CP/M und eines unter MSDOS), welche die gleiche Sprache sprechen (soll heißen, welche das gleiche Übertragungsformat benutzen). Hier bieten sich KERMIT und MODEM7 an, beides Programme, die sowohl unter CP/M als auch unter MSDOS zur Verfügung stehen. Zusätzlich benötigt man noch ein Übertragungskabel, welches aber, zumindest bei nicht allzu hoch geschraubten Geschwindigkeitsansprüchen, mit drei Adern auskommt (TxD, RxT und Masse).

Wer weder eine serielle Schnittstelle hat, noch eines der oben genannten Programme zum Laufen bringt, der kann eventuell auf die zuletzt genannte Möglichkeit, die parallele Übertragung, zurückgreifen. Hierbei werden die Daten auf dem CP/M-Rechner ausgedruckt. Statt dem Drucker hängt aber ein PC mit entsprechendem Programm an der Schnittstelle. Informationen zu dieser, zugegeben etwas exotischen Art der Datenübertragung, findet man in der Zeitschrift c't 6/88 Seite 166 ff (PC Bausteine: Details über die parallele Schnittstelle).

Nun, nach mehr als einer Seite, sind wir endlich beim Kernthema dieses Artikels (Arnulf und Kajot haben bestimmt schon mit Rotstift die Randbemerkung "Thema verfehlt!" angebracht), der Inkompatibilität zwischen den Daten-(dbf-) dateien von dBase II und dBase III. Wie wir gleich sehen werden, läßt sich dieses Problem aber, im Verhältnis zu den oben genannten, relativ einfach lösen!

Zunächst könnte man, welch schrecklicher Gedanke, auch unter MSDOS mit dBase II arbeiten. Tatsächlich gibt es eine Version II des wohl bekanntesten Datenbankprogramms auch für MSDOS. Wer aber einmal, als eh nur sporadischer und damit ständig handbuchblätternder dBase II-Nutzer, dBase III gestartet und mit ASSIST (einem Menüsystem für Leute, die nicht gerne Handbücher wälzen) gearbeitet hat, der wird dieses Ansinnen als unannehmbar ablehnen!

Nach längerer Suche im Handbuch findet sich dann aber doch eine Möglichkeit, sowohl die Fingerkuppen als auch die Tastatur des MSDOS-Rechners zu schonen und die "alten" Daten zu übernehmen. Dazu geht man wie folgt vor:

1. unter dBase II die gewünschten Daten exportieren (Funktion COPY)
2. die exportierten Daten auf MSDOS übertragen (siehe oben)
3. unter dBase III die Daten wieder importieren (Funktion APPEND).

Dabei ist zu beachten, daß die Struktur der neu angelegten dBase III-Datenbank der exportierten dBase II-Daten entsprechen muß. Um die Vorgehensweise deutlicher zu machen, folgt hier als Beispiel die Übertragung bestimmter Felder aller Datensätze der Mitgliederverwaltung des CLUB 80.

Als Quelle wird die Datei PDATEN.DBF benutzt. Wie schon erwähnt, gehört sie zur Mitgliederverwaltung des CLUB 80, geschrieben von unserem ehemaligen Mitglied Harald Mand in dBase II auf einem Model 4p unter CP/M 2.2. Nachdem dBase gestartet und die Datumsabfrage beantwortet wurde, erhält man den obligatorischen Punkt als Prompt (auch Eingabeaufforderung genannt). Nun muß zunächst die Datenbankdatei geöffnet werden. Dies geschieht mit dem Befehl:

```
. USE PDATEN.DBF
```


Danach muß man sich unbedingt über die Struktur der Datenbank im klaren werden. Dazu kann man die Befehle DISPLAY STRUCTURE oder LIST STRUCTURE benutzen. Die Antwort von dBase II sieht wie folgt aus:

```
STRUCTURE FOR FILE:  A:PDATEN  .DBF
NUMBER OF RECORDS:   00074
DATE OF LAST UPDATE: 29/04/90
PRIMARY USE DATABASE
FLD       NAME       TYPE WIDTH   DEC
001      NAME         C    020
002      VORNAME      C    020
003      STRASSE      C    025
004      LAND         C    003
005      PLZ          C    005
006      WOHNORT      C    025
007      TELP         C    020
008      TELG         C    020
009      GEBOREN      C    010
010      EINTRITT     C    010
011      BEITRAG      L    001
012      NUMMER       N    004
** TOTAL **              00164
```

Nun kann man sich überlegen, welche Daten man übernehmen will. Ich habe für dieses Beispiel die Felder NAME, VORNAME, STRASSE, PLZ, WOHNORT und TELP ausgewählt. Diese Felder können nun mit dem Befehl COPY in eine Datei kopiert werden. Dieser Befehl hat so viele mögliche Parameter, daß vor dem Beispiel erst einmal ein paar Erläuterungen angebracht sind. Der Befehlssyntax lautet:

```
COPY TO <Dateiname> [<Geltungsbereich>] [FIELD <Felderfolge>]
     [FOR <Ausdruck>] [WHILE <Ausdruck>]
     [SDF] [DELIMITED [WITH <Begrenzung>]
```

Hierbei bedeuten:

Dateiname	:	Name der Datei, in die die Datensätze geschrieben werden sollen
Geltungsbereich	:	ALL = alle Datensätze (Voreinstellung!) NEXT (n) = vom aktuellen Datensatz ab die nächsten n Sätze RECORD (n) = nur der Datensatz mit der Nummer n
FIELD <Felderfolge>	:	die Felder, die übernommen werden sollen (wenn keine Felder angegeben werden, werden alle Felder übernommen)
FOR <Ausdruck>	:	alle Sätze, die die logische Bedingung des <Ausdrucks> erfüllen
WHILE <Ausdruck>	:	alle Sätze, die auf den aktuellen Datensatz folgen, bis die im <Ausdruck> angegebene logische Bedingung nicht mehr erfüllt ist.
SDF	:	Anlegen einer Datei im Standard-ASCII-Format (als Dateityp wird, wenn nicht anders angegeben, .TXT eingesetzt!)
DELIMITED	:	in der Zieldatei werden alle Felder durch Komma getrennt und die Felder vom Typ "Zeichenkette" in Hochkomma (') eingeschlossen.
WITH <Begrenzung>	:	sollen die Zeichenreihen durch ein anderes Sonderzeichen eingeschlossen werden, so kann das gewünschte Zeichen hier angegeben werden. Handelt es sich dabei um ein Komma, werden nicht belegte Stellen am Ende von alphanumerischen Feldern bzw. am Anfang von numerischen Feldern entfernt. Außerdem wird der Wert alphanumerischer Felder nicht in Sonderzeichen gesetzt.

Das war jetzt eine ganze Menge recht trockene Information, hier nun zwei Beispiele. Mit dem Befehl

. COPY TO PDATEN FIELD NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP SDF

wird eine Standard-ASCII-Datei mit Namen PDATEN.TXT erzeugt. Die Datei enthält die Felder NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP aller in der aktuellen .DBF-Datei enthaltenen Datensätze. Die einzelnen Datensätze sehen wie folgt aus:

```
0        10        20        30        40        50        60        70        80        90        100       110
.........!.........!.........!.........!.........!.........!.........!.........!.........!.........!.........!.....
Ahlers              Armin               Rotdornstr. 13           2116 Hanstedt/Nindorf     04184    /7667
Bernhardt           Helmut              Hafenstr. 7              2305 Heikendorf           0431     /241987
```

Wie man sieht, wird jedes Feld linksbündig geschrieben und bis auf seine maximale Länge mit Leerstellen aufgefüllt. In diesem Beispiel sind keine numerischen Felder enthalten, dort wäre es aber genau anders herum!

Der Befehl

. COPY TO PDATEN.DEL FIELD NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP FOR PLZ > '5999' DELIMITED

erzeugt eine Standard-ASCII-Datei mit Namen PDATEN.DEL. Die Datei enthält die Felder NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP aller Clubmitglieder, deren Postleitzahl größer als 5999 ist. Die einzelnen Sätze sehen wie folgt aus:

```
0        10        20        30        40        50        60        70        80
.........!.........!.........!.........!.........!.........!.........!.........!..
'Betz','Heinrich','St.Wolfgangstr. 13','8551','Hausen','09191    /31698'
'Brandl','Hermann','C{ciliastr. 30','8439','Postbauer-Heng','09180    /493'
```

Wie nach der Lektüre der Beschreibung des COPY-Befehl nicht anders erwartet, sind die einzelnen Felder durch Komma getrennt und die Zeichenketten in Hochkomma eingeschlossen. Dem aufmerksamen Beobachter wird weiterhin auffallen, daß die Zeichensatzkonvertierung hier noch nicht durchgeführt ist.

Welchen Vorteil hat nun das DELIMITED- gegenüber dem SDF-Format? Zunächst einmal fällt auf, daß die DEL-Datei um einiges kleiner ist, als die TXT-Datei. Die Platzersparnis hängt allerdings vom Füllungsgrad der einzelnen Felder ab. Läßt man alle Daten der Mitgliederdatei PDATEN.DBF in eine SDF-Datei kopieren, hat diese einen Umfang von 10 Kilobyte. Eine DEL-Datei mit gleichem Inhalt belegt nur 6 kByte. Dieser Größenunterschied könnte (bei größeren Datenbanken) beim Übertragen über eine serielle Schnittstelle ein Grund sein, sich für das DELIMITED-Format zu entscheiden.

Als zweites Argument könnte dienen, daß es sowohl unter CP/M als auch unter MSDOS eine stattliche Anzahl von Programmen gibt, die zwar das DEL-, nicht aber das SDF-Format importieren können. Insbesondere die Klassiker WordStar und Mailmerge sowie einige Tabellenkalkulationsprogramme bieten diese Möglichkeit. Man sollte bei der Wahl eines mit WITH gewählten Trennzeichens übrigens beachten, daß dBase selbst nur DEL-Dateien wieder einwandfrei importieren kann, bei denen die Felder in Hochkomma oder Anführungsstrichen eingeschlossen sind.

Und gerade das wollen wir ja! Die exportierte SDF- oder DEL-Datei wird nun auf den MSDOS-Rechner übertragen und der Zeichensatz angepaßt. Fertig? Dann kann es jetzt mit dBase III weitergehen!

Nach dem Start von dBase III und dem Erscheinen des obligatorischen '.' als Prompt, fühlt man sich fast wieder auf einen CP/M-Rechner versetzt (ich weiß, das ist bei weitem nicht die einzige Gemeinsamkeit zwischen CP/M- und MSDOS-Rechner!) und kann daher ungehemmt draufloshacken.

Zunächst muß man mit dem Befehl

. CREATE PDNEU.DBF

eine neue Datenbank eröffnen und deren Struktur definiert werden. Dabei muß, sonst klappt der spätere Import der alten Daten nicht, die Struktur der unter dBase II exportierten Daten streng eingehalten werden.

	Feldname	Typ	Länge	Dez
1	NAME	Zeichen	20	
2	VORNAME	Zeichen	20	
3	STRASSE	Zeichen	25	
4	PLZ	Zeichen	5	
5	WOHNORT	Zeichen	25	
6	TELEFON	Zeichen	20	

Da die Eingabe der Feldnamen, -typen und -längen (sowie, falls erforderlich der Anzahl der Dezimalstellen) ebenfalls nur im Detail von der Vorgehensweise unter CP/M abweicht, dürfte auch dieser Schritt keine Probleme mit sich bringen. Aber selbst ohne dBase-Erfahrung ist diese Hürde recht leicht zu nehmen. Wenn alle erforderlichen Felder definiert sind, kann man die obligatorische Frage, ob gleich Daten eingegeben werden sollen, verneinen. Schließlich wollen wir ja Daten importieren!

Dazu brauchen wir den Befehl APPEND. Dieser hat erheblich weniger mögliche Parameter als COPY. Neben der Importdatei und deren Format (SDF oder DELIMITED WITH) kann man noch die Parameter FOR <Ausdruck> und WHILE <Ausdruck> angeben. Für unseren Fall reicht folgender Befehl:

. APPEND FROM PDATEN.TXT SDF

oder

. APPEND FROM PDATEN.DEL DELIMITED WITH '

falls man das platzsparendere DEL-Format gewählt hat.

Was, ihr glaubt nicht, daß das alles war? Doch, doch! Keine Hexerei, kein Woodou, keine Anwendung von Zaubersalz oder sonstigen Hilfsmitteln! Ehrlich!!!

Nachworte

I. für die notorischen Nörgler und Besserwisser:

1. Ich weiß, daß die Befehle COPY und APPEND noch weitere mögliche Parameter und Einsatzmöglichkeiten haben. Der Grund für die Unterschlagung dieser Möglichkeiten ist der, daß sie für den beschriebenen Zweck nicht von Interesse sind.

2. Schon vor Jahren hat mir Gerald Dreyer erzählt, daß es ein Programm mit Namen DCONVERT geben soll, welches nicht nur die Anpassung von dBase II-Daten-, sondern auch Programmdateien erlaubt. Dies wurde mir vor kurzem von Fritz Chwolka bestätigt. Da ich dieses Programm aber nicht hatte (was nicht ist, kann ja noch werden), ...

II. für die zufriedenen CP/M-User (ein paar soll es ja tatsächlich geben)

Ich wünsche euch, daß euer treuer CP/M-Rechnknecht mindestens so lange funktioniert wie mein Model 4p, welches auf meinem Schreibtisch, trotz MSDOSe, immer noch einen Stammplatz hat!

I'm back now!

H.O.

Die Interrupts im Griff

Insbesondere bei der Bastelei am Betriebssystem kommt es häufig vor, daß die Interrupts mit DI vereitelt werden müssen, weil sonst eine Katastrophe passieren würde. Es ist absolut kein Problem, sie anschließend wieder mit EI zuzulassen. Was aber, wenn der CALLer bereits mit disableten INTs auf dieses Programm zugriff? Er wird seine ehrenwerten Gründe dafür gehabt haben. Also den Befehl EI weglassen? Wenn aber nun der CALLer auf INTs angewiesen ist, weil z. B. die Software-Uhr weiterlaufen oder das Tastatur-Repeat aktiv bleiben soll?

Es gibt eine Möglichkeit, den Interrupt-Status schon beim Einsprung festzustellen, um ihn nach der Bearbeitung wieder zu restaurieren. Fragt mich nicht, wieso, ich weiß es auch nicht: Beim Auslesen des R- und des I-Registers mit LD A,R oder LD A,I wird der gegenwärtige Zustand des Interrupt-Flipflops 0 (IFFO) in das P/V-Flag geladen. Ist er 1, dann sind Interrupts zugelassen, ist er 0, dann nicht. Man kann also nach dem Lesen von R oder I mit einem bedingten Sprung (JP PO,nn, JP PE,nn, CALL PO,nn, CALL PE,nn) an eine Stelle verzweigen, wo EI oder DI neu befohlen wird, je nach altem INT-Status.

Weil es mir nun mal so in den Kram paßte, geht das hier vorgestellte Programm (Listing 1) einen anderen Weg: Um für möglichst viele Zwecke dienlich zu sein, lädt es je nach INT-Status lediglich den Akku mit dem Opcode für EI (FBh) oder DI (F3h) und kehrt zurück. Das ist schön kurz, verändert aber den Akku-Inhalt und alle Flags außer dem Carry-Bit. Man kann dem mit PUSH AF begegnen, wenn es erforderlich ist (anschließend POP AF nicht vergessen!).

Die Ladeadresse der Routine ist vollkommen egal. Bei mir lautet sie 3BB4h, weil ich da gerade noch Platz hatte. Dort ist sie auch nicht im Bereich des User-RAMs ab 5200h (unter G-DOS), so daß ORG-Konflikte nicht zu erwarten sind.

Listing 2 zeigt ein Programm, das sich dieser Routine bedient. Es blendet das HRG-RAM in den Speicher ein. Das ist bei meinem Genie IIIs ein hochexplosiver Betriebszustand; jeder Interrupt hätte jetzt einen bühnenreifen Absturz zur Folge. Zugriffe auf die Graphik kommen sowohl mit enableten als auch mit disableten INTs vor. Während des Zugriffs müssen sie auf jeden Fall disablet sein. Mit Hilfe der neuen Routine kann nun der alte Zustand, egal, wie er war, wiederhergestellt werden.

So weit die guten Nachrichten. Die schlechte: Es gibt auch ein Interrupt-Flipflop 1 (Rodnay Zaks numeriert die beiden mit 1 und 2; ich erlaube mir, nach altem Hacker-Brauch von Nr. 0 und 1 zu sprechen. Sorry, Rodnay-Honey!). Tritt ein Interrupt auf, dann wird der gegenwärtige INT-Status aus IFFO nach IFF1 kopiert. Der Sinn ist, daß dann mit den Befehlen RETI und RETN der alte Zustand restauriert werden kann: IFFO holt sich sein früheres Bit aus IFF1 zurück. Das hat Folgen:

Wenn während einer Interrupt-Bearbeitung die Routine *chckint* geCALLt wird, kann nur DI im Akku stehen, weil zumindest nicht maskierbare Interrupts (NMI) sich als erstes die Konkurrenz von weiteren INTs vom Halse halten. Das machen sie, indem sie IFFO, also das für die CPU gegenwärtig gültige INT-Enable-Flag, auf 0 setzen. IFF1 bleibt auf dem alten Wert. Wird nun mit der neuen Routine DI befohlen (EI kann hier nicht auftreten, s. o.), dann setzt dieser Befehl IFFO *und* IFF1 auf 0. So kann während eines NMI-Service die Suppe versalzen werden, weil auf jeden Fall DI dabei herauskommt.

Aber Kopf hoch: Der geneigte User soll eben aufpassen, daß seine NMI-Routine oder ein Glied in der INT-Kette von *chckint* die Flossen läßt. Das ist überhaupt kein Problem. G-DOS ist bezüglich der Interrupt-Bearbeitung wasserdicht (auch CP/M, wie ich hoffe). Programme, die in diese Falle stolpern könnten, sind also immer auf dem Mist des geneigten Lesers gewachsen. Demnach darf diese Routine dort nicht aufgerufen werden, wo ein INT bedient wird. Das trifft z. B. auf die Erkennung der 3-Tasten-Befehle des G-DOS zu (JKL, 123, DFG) und auf Programme, die über einen CALL 4410h in die INT-Kette eingefügt werden sollen. Aber das macht eh' keiner.

Arnulf

```
                00001 ; Listing 1:    Den Interrupt-Status gemäß IFFO feststellen
                00002 ;               und entsprechend EI oder DI in den Akku laden
                00003
3BB4            00004         ORG     3bb4h           ;oder wo auch immer
3BB4  ED5F      00005 chckint LD      A,R             ;IFFO ins P/V-Flag kopieren
3BB6  3EF3      00006         LD      A,0f3h          ;Opcode DI
3BB8  E0        00007         RET     PO              ;falls IFFO = 0
3BB9  3EFB      00008         LD      A,0fbh          ;sonst Opcode EI,
3BBB  C9        00009         RET                     ;falls IFFO = 1
3BB4            00010         END     chckint

00000 Fehler

chckint 3BB4
```

```
                00001 ; Listing 2:    beliebiges Programm, das die Interrupts disablen muß
                00002 ;               und hinterher den alten INT-Status wieder restauriert
                00003
5200            00004         ORG     5200h
5200  CDB43B    00005 start   CALL    chckint         ;je nach INT-Status EI oder DI -> Akku
5203  321452    00006         LD      (eidi),A        ;Befehl dort patchen
5206  DBFA      00007         IN      A,(0fah)        ;Systemport, höchst sensibel!
5208  F5        00008         PUSH    AF              ;Zustand retten
5209  F640      00009         OR      40h             ;Graphik ab 8000h eingeblendet
520B  F3        00010         DI                      ;um Himmels Willen keine Interrupts!
520C  D3FA      00011         OUT     (0fah),A        ;Systemport neu schreiben
520E  2AFC80    00012         LD      HL,(80fch)      ;zwei Graphikbytes abholen
5211  F1        00013         POP     AF              ;alten Systemport
5212  D3FA      00014         OUT     (0fah),A        ;restaurieren
5214  00        00015 eidi    NOP                     ;hier landet EI oder DI
5215  C9        00016         RET
3BB4            00017 chckint EQU     3bb4h           ;Adresse des IFFO-Detektorprogramms
5200            00018         END     start

00000 Fehler

chckint 3BB4    eidi    5214    start   5200
```

ZCPR - nicht nur eine Erweiterung zum CP/M sondern ....

(Allgemeine Information und Beschreibung der TCAP)

Inzwischen wurde schon einiges in den letzten Clubinfos über ZCPR geschrieben, wobei ich unter anderem an die Texte von Rüdiger und Alexander denke.
Hier möchte ich mal allgemein erläutern, was dieses ominöse ZCPR eigendlich ist, und einige Informationen zur Terminal-installierung geben.

Geschichtliches :

Wann und wie CP/M entstand, kann man überall nachlesen und soll hier draußen bleiben. Das Schöne am CP/M ist die Struktur des Betriebssystemes, welche eine Anpassung an fast alle Z80 (8080,8085) Systeme ermöglicht.
Wie bekannt enthält das Betriebssystem einen hardwareunabhängigen Teil , welcher im eigentlichen Sinne das CP/M darstellt, und einen Teil zur Anpassung an die entsprechende Hardware.
Der von DRI erhältliche Programmteil zum CP/M ist in 8080 geschrieben und beinhaltet keine Hardwareanpassung. Die Hardwareanpassung wird vom Distributor des Rechners erstellt oder selbst geschrieben (XBIOS,CBIOS oder ähnliches).
Leider ist das CP/M sehr sparsam mit Befehlen ausgestattet worden, und CP/M 2.x hat nur 5 interne Befehle zur Bearbeitung. Diese sind DIR, TYPE, ERASE, SAVE und RENAME.
Die Ausführung der Befehle ist Sache des Command Controll Prozessors, kurz CCP genannt. Er übernimmt die Tasteneingaben und führt die o.a. Befehle mit Hilfe des restlichen Betriebssystemes aus. Bei CP/M+ existiert der CCP als eigene Datei (CCP.COM), unter CP/M 2.x befindet sich der CCP im Speicher unterhalb dem BDOS (Basic Disk Operating System) und kann zwecks Änderung exakt lokalisiert werden.

Wie kann man aber den CCP ändern ?

Mit Einführung des Z80 bekam man eine Möglichkeit. Durch den umfangreicheren Befehlssatz des Z80 konnte man den CCP im Z80 Code schreiben und hatte noch Platz für Erweiterungen, da das Programm durch die Möglichkeiten des Z80 kürzer wurde. So wurde aus dem CCP der ZCPR (Z80 ControlPRozessor). Die vielen Zwischenstationen und andere gleichartige Programme (XCCP,CCP105..usw) werden kaum noch genutzt und nur der Vollständigkeit halber erwähnt.
Die heutige Version des ZCPR erweitert nicht nur die Befehle des CCP, sondern bildet eine neue Commandoumgebung und verbindet hierfür geschriebene Programme untereinander. Viele Funktionen wurden -ähnlich UNIX- eingebunden, wie z.B. die Beschreibung des Terminals (TCAP).
Nun aber mal ans Eingemachte. Im INFO wurde bis jetzt nur die Public Domain Version des ZCPR behandelt, die Version 33. Deren Installierung setzt aber doch genaue Kenntnisse des Systems voraus und es sollte das Programm MOVCPM vorhanden seien. Oft ist aber nur SYSCOPY, SYSGEN o.a. vorhanden, mit denen man nicht ein neues CP/M für andere Speichergröße erzeugen kann, sondern nur die Systemspuren kopiert. (Anm. Mit MOVCPM kann das CP/M entsprechend der Speichergröße generiert werden. MOVCPM beinhaltet ein Abbild des CP/M's).
Auch ist man bei einer 'Handinstallierung' auf eine Speichergröße festgelegt. Dies muß aber nicht sein, denn inzwischen bekommt man für CP/M2.2 und CP/M+ ein selbstinstallierendes ZCPR, welches allerdings COPRIGHTED und folglich nicht ganz um sonst ist.

Die ZCPR-Version für CP/M2.2 heißt 'NZCOM' , die für CP/M+ 'Z3PLUS'. Im weiteren Text schreibe ich aber für alle Versionen ZCPR , da ich mich auf alle beziehe und bei Unterschieden darauf hinweise.
Beide copyrighted ZCPR - Umgebungen sind in ca.5 Minuten installiert und bieten eine größere Flexibilität als die Pd-Version.

T E R M I N A L - B E S C H R E I B U N G

Eine Grundfunktion innerhalb des ZCPR ist der Installierung der entsprechenden Terminalbefehle zuzuordnen. Diese Installierung kann per Programm oder zu Fuß erfolgen.
Folgende Programme kann man zur Erstellung der Terminalbeschreibung, im weiteren Text TCAP (TerminalCAPability) genannt, nutzen:

TCMAKE.COMP	Erstellt interaktiv eine TCAP - Datei
TCSELECT.COM	Wählt aus einer Datendatei ein Terminal aus und erstellt eine TCAP - Datei.
TCVIEW.COM	Zeigt die Parameter des eingestellten Terminals auf und hilft so bei der Kontrolle eines 'zu Fuß' installierten Terminals.
TCCHECK.COM	Testet eine TCAP-Datei auf korrekten Aufbau und gibt Fehler an.
TCSRC.COMP	Erstellt einen kommentierten SOURCE-CODE (MYTERM.Z80) aus einem TCAP-File und dient zur korrekten Installierung eines TCAP für ZCPR33/34 da hier zum Original von Richard Conn einige Verbesserungen bzw. Erweiterungen innerhalb der TCAP gemacht wurden.
TCAP.COM	Beispielprogramm zur Nutzung des TCAP von Basic aus .

WAS soll diese Terminalbeschreibung ?
Programme, welche für ZCPR geschrieben werden, kennen die Existenz der TCAP und können so ohne gesonderte Installierung ablaufen. Hiermit können Programme direkte Cursorsteuerung nutzen und laufen auf jedem Rechner mit ZCPR. Neue Funktionen wie PULL-DOWN Menues oder Buisinessgraphic werden hiermit auf einfacher Weise möglich und dies ohne eine jeweilige Neuinstallierung.

Die Terminalbeschreibung ab März 1989:

Ab ZCPR-Version 3.3 wurde die Terminalbeschreibung um einige Funktionen erweitert. Vor allem ist nun die Möglichkeit der Grafikeinbindung gegeben und einige Beispielprogramme sind in der 'SAYGE-PD' hierfür enthalten

TCAP Definition

Die TCAP ist ein 128 Byte langes Feld und befindet sich direkt hinter der Umgebungbeschreibung (ENV). Hier möchte ich den Aufbau und die Struktur der TCAP anhand eines Beispieles aufzeigen.

Aufbau und Strucktur der Terminalbeschreibung (TCAP):

BYTE	Bedeutung
0	Terminalname , bestehend aus 15 alphanumerische Zeichen. das erste Zeichen darf kein 'SPACE' sein, da so ein uninstalliertes Terminal definiert wird.
15	Grundlegende Eigenschaft des Terminals.

Jedes der 8 Bit von BYTE 15 hat eine festgelegte Bedeutung, welche nachfolgend aufgezeigt werden.

BIT	Wert	Definition
0	1	= Revers oder helle Darstellung
0	0	= Dunkle oder in der Helligkeit reduzierte Darstellung
1	1	= AutoLineFeed beim Schreiben in der letzten Spalte (weiterschreiben in neuer Zeile)
1	0	= Kein Auto-LF
2	1	= Scrollen beim Schreiben in letzter Spalte und/oder Zeile.
2	0	= Kein Scrollen beim Schreiben in letzter Spalte und/oder Zeile.
3	1	= 10 Sekunden Delay beim Einschalten
3	0	= Kein Delay

Die folgenden vier Bytes definieren die CURSOR-Tasten des Terminals. Falls Euer Terminal keine CURSOR-Tasten oder anderweitig definierte Cursorsteuerung hat, kann man jeden Controll- oder ESC-Code mit 1 byte Länge wählen. Hier sind auf jeden Fall die Cursor-Codes von WORDSTAR vorzuziehen und werden allgemein benutzt.

Wordstar- 'DIAMANT'

Control-E (Hoch)

Control-S (Links) (Rechts) Control-D

Control-X (Runter)

BYTE	Bedeutung	
16	Cursor Up	Cursor UP Befehl des Terminals (^E)
17	Cursor Down	Cursor DOWN Befehl des Terminals (^X)
18	Cursor Right	Cursor RIGHT befehl des Terminals (^D)
19	Cursor Left	Cursor LEFT Befehl des Terminals (^S)

Einige Terminals brauchen DELAYS zur Durchführung spezieller Funktionen. Hier werden die Wartezeiten (Delays) im MILLIsekunden angegeben. Zum Erreichen der maximalen Ausgabegeschwindigkeit des Terminals sollten die Wartezeiten aus den Serviceunterlagen übernommen werden.

Wartezeiten:

BYTE	Bedeutung	
20	Clear Screen	Zeit in Millisekunden (0 to 255)
21	Move Cursor	Zeit in Millisekunden (0 to 255)
22	Clear to EOL	Zeit in Millisekunden (0 to 255)

Ab Byte 23 vom Beginn der TCAP werden die Befehlzeichenketten mit einem NULL-Byte abeschlossen. Hierdurch ist die Befehlslänge nicht auf nur ein oder zwei Bytes begrenzt.
Falls einige der vorgesehenen Möglichkeiten im eigenen Terminal nicht vorhanden ist, wird dies durch eine einzelne Null '0' angegeben (binäre 0).

Befehl		Bedeutung
Clear Screen	(CL)	- Löscht Bildschirm; Cursor linke obere Ecke
Cursor Movement	(CM)	- Gesamte Zeichenfolge zur Cursorpositionierung.Die Einzelheiten sind am Ende der Auflistung angegeben oder in 'THE MANUAL' von Richard Conn nachzuschlagen. Hier wird am Ende der Sequenz ein '%N' angehängt um eine binäre 0 ans Terminal zu senden.
Clear to End-of-Line	(CE)	- Lösche von Cursor bis Zeilenende.
Standout Begin	(SO)	- Normaler Darstellungsmodus, im allg. halbe Helligkeit und nicht Revers.
Standout End	(SE)	- Ende Standout Modus , ab hier REVERS oder/und doppelte Helligkeit.
Initialize Terminal	(TI)	- Initialisiert das Terminal
De-initialize Term	(TE)	- Setzt die Initialisierung zurück
Delete Line	(LD)	- Löscht die Cursorzeile und scrollt die unteren Zeilen um 1 höher.
Insert Line	(LI)	- Fügt vor der Cursorzeile eine leere Zeile ein.
Clear to EOS	(CD)	- Löscht Bildschirm ab Cursor bis Schirmende.

Die nun folgenden Definitionen implementieren Grafikmöglichkeiten.

Graphics Delay - Delay for Graphics On/Off (0-255). Sollte größer als 2 sein (Angabe in Millisekunden).

Graphics Mode On	(GO)	- Grafik an
End Graphics Mode	(GE)	- Grafik aus, Textwidergabe an
Cursor Off	(CDO)	- Cursor aus, wird für Pull-Down Menues benötigt.
Cursor On	(CDE)	- Cursor an , s.o.

Die Grafikzeichen sind bei jedem Terminal anders, und es könnten sogar normale Textzeichen verwendet werden, zum Beispiel das Asterix '*' als Ecke, '!' als vertikale Linie und '-' als horizontale Linie. Hiermit ist auch auf einem nicht-grafikfähigem Terminal Buisiness-Grafik möglich.

```
Befehl                         Bedeutung
---------------------------------------------------------------------
Upper-Left Corner        (GULC)  -   **STRING***
Upper-Right Corner       (GURC)  -   "     "     "
Lower-Left Corner        (GLLC)  -   "     "     "
Lower-Right Corner       (GLRC)  -   "     "     "
Horizontal Line          (GHL)   -   "     "     "
Vertical Line            (GVL)   -   "     "     "
Solid (Full) Block       (GFB)   -   "     "     "
Hashed Block             (GHB)   -   "     "     "
Upper Intersect          (GUI)   -   "     "     "
Lower Intersect          (GLI)   -   "     "     "
Intersection             (GIS)   -   "     "     "
Right Intersect          (GRTI)  -   "     "     "
Left Intersect           (GLTI)  -   "     "     "
```

Der Rest der TCAP sollte mit binären Nullen aufgefüllt werden, um zukünftige Ergänzungen einfach zu handhaben.
Das folgende Beispiel könnte man auch als Assembler-File für das eigene Terminal ändern, assemblieren und als TCAP übernehmen.

Beispiel:

```
; Z3TCAP:   HEATH19.Z80
; Author:   Harold F. Bower

; Z3 Termcap implementing the Heath/Zenith-19 command set

ESC      EQU      27               ; Escape character

; New Terminal Capabilities Data

Z3TCAP: DEFB      'Heath/Zenith-19'  ; Name of terminal (15 chars)

TRMMOD: DEFB      00000111B         ; B3 = 0 - Term Ready quickly
                                    ; B2 = 1 - Reverse Video,
                                    ; B1 = 1 - Wrap $ EOL,
                                    ; B0 = 1 - Scroll $ EOP

         DEFB     'E'-'$'           ; Cursor up   (WS Diamond)
         DEFB     'X'-'$'           ; Cursor down
         DEFB     'D'-'$'           ; Cursor right
         DEFB     'S'-'$'           ; Cursor left

; Heath-19 doesn't need delays

         DEFB     00                ; Cl delay
         DEFB     00                ; Cm delay
         DEFB     00                ; Ce delay

; Strings start here.

         DEFB     ESC,'E',0         ; CL str (Clear, Home Cursor)
         DEFB     ESC,'Y%+ %+ ',0   ; CM str (Cursor positioning)
         DEFB     ESC,'K',0         ; CE str (Clear to End-of-Line)
         DEFB     ESC,'p',0         ; SO str (Go to Revers Video)
         DEFB     ESC,'q',0         ; SE str (Return Normal Video)
         DEFB     0                 ; TI str (Initialize Terminal)
         DEFB     0                 ; TE str (De-initialize Term)

; Extensions to Standard TCAP
```



```
; Extensions to Standard TCAP
         DEFB     ESC,'M',0         ; LD str (Delete Line)
         DEFB     ESC,'L',0         ; LI str (Insert Line)
         DEFB     ESC,'J',0         ; CD - Clear to EOS String
         DEFB     0                 ; GO/GE - Graphics On/Off Delay
         DEFB     ESC,'F',0         ; GO - Graphics Mode On
         DEFB     ESC,'G',0         ; GE - Graphics Mode End
         DEFB     ESC,'x5',0        ; CDO - Cursor Off string
         DEFB     ESC,'y5',0        ; CDE - Cursor Enable string
         DEFB     'f',0             ; GULC - Upper Left Corner [*]
         DEFB     'c',0             ; GURC - Upper Right Corner [*]
         DEFB     'e',0             ; GLLC - Lower Left Corner [*]
         DEFB     'd',0             ; GLRC - Lower Right Corner [*]
         DEFB     'a',0             ; GHL - Horizontal Line [-]
         DEFB     ''',0             ; GVL - Vertical Line [¦]
         DEFB     'i',0             ; GFB - Full Block String [*]
         DEFB     'w',0             ; GHB - Hashed Block String [ ]
         DEFB     'u',0             ; GUI - Upper Intersection [+]
         DEFB     's',0             ; GLI - Lower Intersection [+]
         DEFB     'b',0             ; GIS - Intersection [+]
         DEFB     'v',0             ; GRTI - Right Intersection [+]
         DEFB     't',0             ; GLTI - Left Intersection [+]
;  File unused space with Nulls

           REPT   128-[$-Z3TCAP]
           DEFB   0
           ENDM

         END
;------------ End of Sample TermCap ------------
```

Soweit mit dem Beispiel. Wie man sieht, geht Grafik auch ohne grafikfähigem Terminal und diese 'grafische Darstellung' war für Buisinessgrafik durchaus üblich.
Da die Hilfsprogramme TCSELECT und TCMAKE nur die Grundfunktionen des Terminals installieren, muß man zur Komplettinstallation das Programm TCSRC nehmen und das erstellte *.Z80 File auf das gewünschte Terminal ändern. Nach assemblieren vom MYTERM.Z80 und umbenennen auf den gewünschten Namen kann diese TCAP ins ZCPR eingebunden werden. Hierbei ist natürlich TCVIEW zur Kontrolle eine große Hilfe.

Noch eine Anmerkung zu den ZCPR-Versionen. Ich gebe persöhnlich den selbstinstallierenden ZCPR-Versionen NZCOM und Z3PLUS den Vorzug, habe aber auch auf meinem MORROW-MD3 ZCPR33 per 'Hand' installiert. Diejehnigen unter Euch, welche auch eine 'Handinstallierung' machen wollen sollten in der PD-Software nach einer eventuellen Installierung für Ihren Rechner nachschlagen. In der SIG ist z.B. für Kaypro, Applle (?) und Morrow ein installiertes ZCPR vorhanden. Ebenfalls ist für 8080/85 eine Version zum ZCPR2 vorhanden, allerdings laufen die neueren Utillities nicht unter 8080/85.
Eine große Hilfe ist das von Helmut Jungkunz zusammengestellte 'ZCPR - Installation-Kit'. Es beinhaltet aus der PD die nötigen Dateien und Informationen zur 'Handinstallierung'. Diese Disketten sind demnächst auch bei Rüdiger Sörensen erhältlich .
Ebenfalls sollte jeder Interessent an PD oder ZCPR die SAGE-PD beim Rüdiger anfordern.
Diese (25 * 780KB) Disketten wurden durch die Initiative von Helmut Jungkunz durch G.Sage, einem der Programmierer des ZCPR's ,nach GERMANY gesandt. Hier findet man viele neue Programme und unter anderem auch eine Menge zum ZCPR.

Literaturhinweise:

The Z-System User's Guide(Bruce Morgen.Richard Jakobson). Eine Beschreibung des Z-Systemes für weniger versierte Anwender.

The ZCPR3.3 User's Guide(G.Sage). Dies ist eine Beschreibung des ZCPR Version 3.3 Commandoprozessor. Es beinhaltet viele Beispiele für die vorteilhafte Nutzung des ZCPR und gilt meist auch für ZCPR Version 3.4 (ZCPR 3.4 ist copyrighted).

ZCPR3: The Manual(Richard Conn). Dies war die 'Bibel' für ZCPR3, aber vieles darin ist heute nicht mehr aktueller Stand. Die Beschreibungen darin zum Z-Help , den Menü Shells und der TCAP (einschließlich TCSELECT,TCMAKE,TCCHECK) sind sehr hilfreich.

Für diejehnigen unter euch, welche das ZCPR selber installieren wollen ist folgendes zu empfehlen:

ZCPR3:The Libraries(Richard Conn). Dieses Buch beschreibt komplett einige hundert Routinen, welche das Schreiben vom Z-System Programmen so einfach macht wie 'Programming in high level language', obwohl man in assembler programmiert. Die letzte Version der Routinen (V4LIB) ist auf SAGE.PD 25.

ZCPR3 Shells(David McCord).Eine Beschreibung des SHELL Systemes unter ZCPR, welche zur effizienten Nutzung und Programmierung beiträgt.

Hier noch einige Literaturangaben zum CP/M:

Betriebssystem CP/M(Jürgen Plate). Dieses Buch wurde ergänzend zum NDR oder KLEIN - Rechner geschrieben und erläutert den Aufbau und die Programmstrucktur von CP/M2.x. Es ist nicht unbedingt als Bedienungsanleitung zu nutzen.

CP/M Handbuch mit MP/M(Rodnay Zaks). Der Autor dürfte vielen für seinen lehrreichen Schreibstiel bekannt sein. Diese Buch ist eine ausführliche Bedienungsanleitung für CP/M, geht aber nicht näher auf Systemprogrammierung ein.

CP/M Anwenderhandbuch(Thom Hogan). Ein Handbuch zum CP/M (Version bis 2.x) und CP/M86, welches keinem fehlen sollte. Thom Hogan hat auch das Handbuch zum OSBORNE 1 geschrieben.

****

Die oben vorgestellten Programm(packet)e NZCOM und Z3PLUS sind copyrighted und dürfen nicht an Dritte weitergegebn werden. Ebenso ist ZCPR3.4 copyrighted.
ZCPR3.3 ist Public Domain Software und zum *privaten Nutzen* freigegeben.

Da Herr Helmut Jungkunz in Deutschland alleiniger Distributor der Fa. Alpha Systems,711 Chatsworth Place, San Jose, CA 95128 ist, ist er für weitere Informationen zu NZCOM und Z3P1US , ebenso wie für Copyrighted Software wie Turbo Pascal, Wordstar 4 etc.. der Ansprechpartner.

Mein Dank auch G.Sage, Bridger Mitchel, Helmut und allen anderen, welche sich für den Z80 (und kompatible) Prozessor verdient machen.

F. Chwolka.

A

AAA

Die drei großen A's des Computerfreaks bedeuten Auspacken, Anschließen und Abhauen. Wer nicht abhaut, muß eben anfangen und wird Byte-süchtig.

Absturz

a) Verlust aller Sinne bei leichtsinnigem Umgang mit Alkohol.
b) Verlust aller Programmdaten bei leichtsinnigem Umgang mit dem Computer, meist nach vorangegangenem leichtsinnigen Umgang mit Alkohol.

Action

a) Computerspielgrenze, bei der es kracht und zischt, daß es eine Freude ischt.

b) Mahnungen diverser Inkassobüros an Anwender, die mit ihren Raten für den Computer im Rückstand sind.

Download vom Brett "MS_DOS-Anwendungen" der BRAINSTORM
Datum: 18-Nov-90 15:29
Von : Michael Mueller
An : All

Betreff : Korrekte Syntaxbeschreibung von Befehlszeilen !

Korrekte Syntaxbeschreibungen in der Backens-Naur-Form (BNF)

Immer wieder taucht nach dem Herunterladen von Programmen, vor allem von Hilfsprogrammen, wie Entpackern, das Problem auf, wie man Befehle und Parameter korrekt eingibt. Dazu ist es wichtig, die mitgelieferten Syntaxbeschreibungen richtig zu verstehen oder selbst eine korrekte Beschreibung zu erstellen.

Fuer diesen Zweck gibt es die sog. "Backens-Naur-Form" (BNF), eine "Metasprache", d.h. eine Sprache, die eine andere Sprache beschreibt. Mit der BNF lassen sich auch ganze Programmiersprachen spezifizieren. Besonders geeignet fuer Mailboxen ist sie, weil sie ohne graphische Hilfsmittel auskommt und daher als einfache ASCII-Datei uebertragen werden kann. Viele Dokumentationen verwenden daher dieses Format, wenn auch nicht immer ganz richtig.

Wie ist die BNF nun aufgebaut?
Im Grunde laeuft es so ab, dass alle Elemente der Programmiersprache oder der Eingabezeile nacheinender definiert werden. Das sieht so aus:

SYMBOL::=LISTE

Die das SYMBOL ist die Bezeichnung, fuer das, was definiert werden soll, z.B. eine Befehlszeile. Die LISTE muss alle(!) Moeglichkeiten enthalten, die fuer das Symbol erlaubt sind. Die LISTE kann weitere SYMBOLE enthalten, die an anderer Stelle definiert werden. Solche SYMBOLE heissen auch "Nichtterminale Symbole".

Hierzu gibt es einige Zeichen oder Zeichenfolgen, die Teile der Metasprache BNF sind, und nicht zur beschriebenen Sprache gehoeren muessen. Solche Zeichen heissen auch "Metazeichen".

Folgende Metazeichen gibt es:

- ::= Das haben wir schon kennengelernt. Es kommt in jeder Definition vor und steht immer nach dem zu definierenden Symbol.
- | Der senkrechte Strich steht zwischen Symbolen, zwischen denen eines (!) ausgewaehlt werden muss.
- " Anfuehrungszeichen schliessen Symbole ein, die nicht mehr anderswo naeher definiert werden, sondern Teil der zu beschreibenden Sprache sind. Das heisst, dass das, was in den Anfuehrungszeichen steht, woertlich so geschrieben werden muss. Solche Symbole heissen auch "Terminale Symbole".
- {} Geschweifte Klammern schliessen Elemente ein, die "freiwillig" (alternativ) verwendet und beliebig oft wiederholt werden koennen.
- [] Eckige Klammern sind eigentlich kein Teil der BNF (man kommt auch ohne sie aus), aber man stoesst trotzdem haeufig auf sie, da man mit ihnen manches kuerzer ausdruecken kann. Sie schliessen dann Elemente ein, die alternativ verwendet koennen, aber NICHT wiederholt werden duerfen.

WICHTIG: Jedes verwendete, nichtterminale Symbol MUSS anderswo definiert werden. Wenn in dessen Definition wieder nichtterminale Sybole verwendet werden dann muss auch dieses definiertwerden und so weiter. Man darf die so entstehende Kette erst dann abbrechen, wenn sich jedes nichtterminale Symbol ausschliesslich auf terminale Symbole zurueckfuehren laesst.

Rekursive Definitionen sind erlaubt, d.h. in der LISTE darf das zu definierende Symbol erneut auftauchen.

ACHTUNG: BNF beschreibt nur die reine Syntax, d.h. welche Eingaben erlaubt sind. Die Bedeutung der Befehle muss extra beschrieben werden.

Beispiele aus der Programmiersprache BASIC:

EINGABEZEILE::=ZEILENNUMMER BEFEHLSFOLGE|BEFEHLSFOLGE

Das Gleiche mit eckigen Klammern:

    EINGABEZEILE::=[ZEILENNUMMER]BEFEHLSFOLGE

ZEILENNUMMER und BEFEHLSFOLGE sind nichtterminale Symbole.
Sie muessen naeher definiert werden:

    ZEILENNUMMER::=ZIFFER{ZIFFER}

Das bedeutet, dass man mindestens einmal ZIFFER schreiben muss. Die maximale Anzahl der Ziffern ist noch nicht bestimmt, und muss in einer Erlaeuterung angegeben werden. Erlaeuterungen erfolgen in der Ragel erst nach allen Definitionen z.B. in Form einer Fussnote. Erlaeuterungen gehoeren NICHT zur BNF.

Weiter in der Definition: Auch ZAHL ist nichtterminal:

    ZIFFER::="0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"

Nun ist man endlich bei terminalen Symbolen, naemlich den Ziffern angekommen und hat damit ZEILENNUMMER vollstaendig definiert.

Nun zur Definition von BEFEHLSFOLGE:

    BEFEHLSFOGE::=BEFEHL{":"BEFEHL}

Man koennte anstelle der geschweiften Klammern auch mit einer rekursiven Definition arbeiten:

    BEFEHLSFOLGE::=BEFEHL|BEFEHL":"BEFEHLSFOLGE

Die Rekursion bricht ab, sobald man die erste Alternative waehlt.
Jetzt muss BEFEHL definiert werden.

    BEFEHL::=PRINTBEFEHL|INPUTBEFEHL|ZUWEISUNG usw.

Jetzt muessen saemtliche Befehle definiert werden. Das wuerde hier zu weit fuehren. Eine komplette Syntaxdefinition von Standard-BASIC wuerde etwa 5 engbedruckte Seiten fuellen, aber auch saemtliche Zweifen an der richtigen Schreibweise eines Befehls ausraeumen.

Ich hoffe, das genuegt, um BNF hinreichend und verstaendlich zu beschreiben. Wer ein Programm verbreitet, das irgendwelche Texteingaben verlangt, sollte in der Dokumentation dazu, die Eingaben auf diese Art definieren. Das mag zwar manchmal etwas umstaendlich erscheinen, erhoeht jedoch die Qualitaet der Dokumentaion enorm. Merke: Bei professionellen Programmen mit guter Dokumentation, wird mit der Erstellung derselben ein aehnlicher Aufwand getrieben wie mit der Erstellung des Programmes selbst! Das wuerde niemand machen, wenn es nicht wirklich sinnvoll waere!!

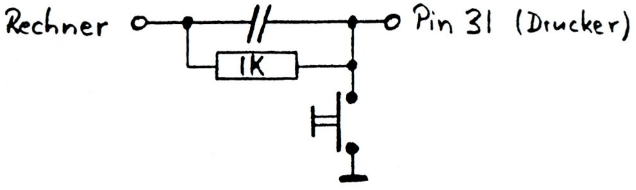
Mit diesem Hinweis: Ciao

Michael Mueller

..Hardware ..Hardware ..Hardware

Drucker-Reset

Neuere Drucker haben in der Regel einen eingebauten Buffer mit 16 bis 64KB Kapazität und mehr. Normalerweise ist das ganz angenehm, wenn der Rechner fast sofort nach dem Start des Ausdrucks schon wieder zur Verfügung steht, aber manchmal kann das auch zu Problemen führen. Dann nämlich, wenn man im Ausdruck einen Fehler bemerkt und den Drucker gerne anhalten würde, bevor er noch den Rest seines Buffers völlig sinnlos auf's Papier kleckert. Einige Drucker, wie z.B. der HP Deskjet, haben dafür einen Reset-Knopf, die meisten anderen lassen sich aber nur materialzermürbend mit dem Holzhammer (Netzschalter) bremsen. Die Centronics-Schnittstelle bietet nun aber die relativ unbekannte Möglichkeit, dasselbe viel sanfter zu erledigen. Wie Ihr in der folgenden Schaltung, die aus der Elektor stammt, sehen könnt, ist der Aufwand denkbar gering. Hoffentlich haben sich alle Hersteller an die Norm gehalten und nicht gemeinerweise dort den Selbstzerstörungseingang hingelegt. Garantien irgendwelcher Art möchte ich lieber nicht übernehmen.



Alexander Schmid

Megabit-RAMs einfach erfrischt

Helmut Bernhardt

Im letzten Info hat Alexander eine aufwendige Erweiterung der Erweiterung der 7Bit-Refreshadresse des Z80 auf 8 Bit beim 256K-Banker und nun 9 Bit für Megabit-RAMs vorgestellt. Einerseits wäre ich ziemlich stolz gewesen, sowas selbst ersonnen zu haben, andererseits bin ich aber viel zu faul, die bisherige aus der c't übernommene Schaltung zu durchblicken und dann noch auf 9 Bit zu erweitern - es geht nämlich auch ohne.

Wie aus Datenblättern und den beiden üppigen Knowledge-Anhäufungen zu DRAMs in den Heften 10 und 11 aus c't 1990 von Johannes Assenbaum zu entnehmen ist, gibt es für alle 1M-Bit-DRAMs die Möglichkeit, das Refreshing durch den /CAS-before-/RAS-Rfresh zu erledigen. Bei 256K-Bit--DRAMs wird das nur bei einigen Herstellern unterstützt - hier sollte man in entsprechenden Schaltungen diese Technik nicht anwenden, weil nicht gesichert ist, ob die darin eingesetzten Chips den Sport mitmachen.

Beim /CAS-before-/RAS-Refresh muß nur ein low aktives /CAS an die RAMs gelegt werden, bevor ein /RAS kommt. Dann wird im RAM mit einem internen Zähler eine Columnadresse erzeugt und die damit adressierten Speicherzellen werden refreshed. Ein externer Memorycontroller (bei uns z.B. der Z80) braucht sich dann nicht mehr um eine von außen zugeführte Refreshadresse zu kümmern. Damit ist es auch nicht mehr nötig, daß in der Breite der von den RAMs verwendeten Refreshadresse die niedrigen Adressen der CPU zusammen beim /RAS durch die Adreßmultiplexer durchgeschaltet sein müssen. Die CPU-Adressen können in wilder Folge an die A- und B-Eingänge der Multiplexer gelegt werden - wie es für das Layout gerade am besten kommt.

Und wie erzeuge ich einen /CAS-before-/RAS-Refresh? Ich leite wie immer ein aus Adreßdecodierung für die RAM-Bank und Timing (relativ zu /RAS und MUX) zusammengesetztes /CAS her und lege es über ein AND-Gater mit /RFSH bzw. /RFS beim HD64180 zusammen und verabreiche den RAMs das Produkt dieser geringen Mühe an den /CAS-Eingängen.

Beim Refresh der CPU erscheint das /RFSH-Signal sehr viel früher als das als /RAS verwendete /MERQ. Damit ist die Bedingung der RAMs für den /CAS-before-/RAS-Refresh erfüllt. Bei normalen RAM-Zugriffen wird das herkömmliche /CAS, das zeitlich versetzt nach /RAS und MUX aktiv wird, zu den RAMs durchgeschaltet.

Wer mehr dazu wissen möchte, kann an anderer Stelle in diesem Info eine Anwendung in Form einer 2MB-RAM-Floppy finden, wo diese Refreshart den Schaltungsaufwand wesentlich reduziert hat.

2MB-RAM-Erweiterung für Prof 180x

Helmut Bernhardt

Zur Erweiterung des Speichers über die 512K auf dem CPU-Board des Prof 180x hinaus hat Conitec die Turbo-RAM vorgesehen. Das 'Turbo' in dieser Bezeichnung ist aber mit Vorsicht zu genießen. Da diese Speichererweiterung auf dem ECB-Bus steckt, liegen zwischen CPU und diesem Speicher 2 Datentreiber. Bei 9,2 MHz ist die Turbo-RAM nur mit einem Wait zu betreiben, was dann auch beim Zugriff auf die ursprünglichen 512K enorm bremst. Hier sei eine Alternative vorgeführt, die ohne Waits eine Erweiterung um 2MB in Form eines Huckepack-Boards auf dem CPU-Board bringt.

Um zu erklären, wie das machbar ist, soll zunächst auf die Verhältnisse eingegangen werden, wie im Prof der Zugriff auf verschiedene Speicher geregelt ist. da dies alles hauptsächlich im PAL 14L8 gesteuert wird, sei das disassemblierte PAL-Listing hier wiedergegeben:

PAL14L8
ausgelesen mit c't PAL-Brenner, 1988 Be
Grundversion nach Conitec

```
me ioe rd lir busak a18 e mm0  mm1 adr a5 gnd
a2 a7 a19 rom cas0 cas1 fdc gate iosel dir a19 vcc

/dir     = /ioe * /rd * /adr * a7 * /a5             ;Richtungssteuerung
           + /me * /rd * /a19                       ; ECB-Bus-Daten-
           + rd * lir * busak                       ; -treiber
/iosel   = /ioe * lir * /a18 * /adr * a7 * /a5 * /a2
/fdc     = /ioe * lir * /a18 * /adr * a7 * /a5 * a2
/gate    = mm0 * /mm1                               ;Freigabe ECB-Bus-
           + mm1                                    ; -treiber
/a19     = /mm1                                     ;Freigabesignal für
           + a18 * /mm0                             ; externe Speicher
           + /a18 * mm0
/rom     = /me * /a18 * /rd * /mm0 * /mm1
/cas0    = /me * /a18 * e * rd * /mm0 * /mm1
           + /me * /a18 * e * mm0
/cas1    = /me *  a18 * e * /mm0
           + /me *  a18 * e *  mm0 * /mm1
```

'adr' ist ein Eingangssignal für das PAL, in dem A3, A4 und A6 entsprechend den zu erzeugenden I/O-Freigabesignalen im Bereich D8H-DFH decodiert sind. 'E' ist ein spezielles Taktsignal des HD64180, das hier zum Timing der /CAS-Signale verwendet wird. MM0 und MM1 sind Ausgänge eines 74LS259-Latches mit einzeln zu setzenden Bits. Die übrigen Eingangssignale des PALs sind die normalen Signale der CPU.

Die I/O-mapped Freigabesignale /IOSEL und /FDC sind hier uninteressant. Für die Steuerung des ECB-Businterface und der memory-mapped Resourcen werden im PAL die Signale A19, /GATE, /ROM, /CAS0 und /CAS1 erzeugt. /ROM, /CAS0 und /CAS1 sind low aktive Freigabesignale für das ROM und für je einen 256K-Block RAM auf dem CPU-Board. /GATE, das an die Enable-Pins 19 der 74ALS245-Bustreiber gelegt ist, steuert die grundsätzliche Möglichkeit der CPU, auf den ECB-Bus zugreifen zu können. Dieses Signal muß high sein, wenn die Prof 180x-Karte im Slave--Modus für sich allein werkeln soll und eine andere CPU-Karte den Bus kontrollieren soll.

/DIR steuert die Richtung, in der der 74ALS245-Datentreiber die Daten vom/zum ECB-Bus treibt: /DIR = 0 bedeutet, daß die Daten vom Prof 180 zum ECB-Bus getrieben werden. In die Richtungssteuerung geht nicht nur ein, ob gelesen oder geschrieben wird (/RD) sondern auch, ob die adressierte Baugruppe auf dem CPU-Board oder 'draußen' auf dem Bus liegt. Wenn eine Baugruppe auf dem CPU-Board gelesen wird, muß der Bustreiber zum Bus hin treiben, um auf dem internen Datenbus keinen Datenkonflikt zu erzeugen. Außerdem wird auch das Signal /BUSAK ausgewertet, das bei Zugriff eines externen DMA-Controllers auf Baugruppen des CPU-Boards die Treiberrichtung in Bezug auf /RD umkehrt.

A19 ist ein high aktives Freigabesignal, das auf den ECB-Bus gelegt ist und bei Conitec u.a. für die Steuerung der Turbo-RAM benutzt wird.

Die Analyse des disassemblierten PAL-Listings ergibt für die Steuerung der Freigaben von ROM und internem und externem RAM die Beziehungen:

Steuersignale MM1	MM0	A18	freigegebene Baugruppe	Pegel v. /GATE	Pegel v. A19
0	0	0	ROM bei /RD; RAM bei /WR	1	0
0	0	1	RAM Bank 1	1	0
0	1	0	RAM Bank 0	0	0
0	1	1	RAM Bank 1	0	0
1	0	0	externes RAM	0	1
1	0	1	RAM Bank 1	0	0
1	1	0	RAM Bank 0	0	0
1	1	1	externes RAM	0	1

Wenn MM0 und MM1 beide low sind, ist /GATE high und es kann nicht auf Speicher auf dem ECB-Bus zugegriffen werden. Durch Umschalten auf MM0=1 wird das beim Lesen in die unteren 256K eingeblendete ROM abgeschaltet. Mit MM1=1 wird der grundsätzliche Zugriff auf externe Speicher ermöglicht. Dann wird mit MM0 gesteuert, ob der externe Speicher in die unteren oder die oberen 256K des HD64180-Adreßraums eingeblendet wird.

Für das Bereitstellen des bei Conitec 'externen' Erweiterungsspeichers nun innerhalb des CPU-Boards muß die Bussteuerung nun etwas anders gestaltet werden. Außerdem müssen Freigabesignale für zwei weitere RAM-Banks mit je 1 MB geschaffen werden. Dafür muß das PAL zunächst entrümpelt werden. Auf das Signal /GATE kann man verzichten, wenn man die CPU-Karte nur als Master fahren will. Die Pins 19 der 74ALS245--Bustreiber werden einfach an Masse gelegt.

Da kein externer Speicher zu steuern ist, wird auch das Signal A19 nicht mehr benötigt. Es sind dann zwei Ausgänge und ein Eingang am PAL freigeworden. Die beiden Ausgänge sollen die Freigabesignale für die beiden zusätzlichen 1MB-RAM-Blocks ergeben und der Eingang wird für die Selektion eines dieser beiden Blocks mit einem Signal Q2 beaufschlagt. Dann bekäme ein neues PAL den Inhalt:

PAL20L8
08.08.90 H. Bernhardt
Erweiterung des Prof 180x auf 2,5 MB RAM

```
me ioe rd lir busak a18 e mm0  mm1 adr a5 gnd
a2 a7 cas2 rom cas0 cas1 fdc cas3 iosel dir q2 vcc
```

```
/dir     = /ioe * /rd * /adr * a7 * /a5
         +/me * /rd
         + rd * lir * busak
/iosel   = /ioe * lir * /a18 * /adr * a7 * /a5 * /a2
/fdc     = /ioe * lir * /a18 * /adr * a7 * /a5 *  a2
/rom     = /me * /a18 *      /mm0 * /mm1 * /rd
/cas0    = /me * /a18 * e * /mm0 * /mm1 *  rd
         + /me * /a18 * e *  mm0
/cas1    = /me *  a18 * e * /mm0
         + /me *  a18 * e *  mm0 * /mm1
/cas2    = /me * /a18 * e * /mm0 *  mm1 * /q2
         + /me *  a18 * e *  mm0 *  mm1 * /q2
/cas3    = /me * /a18 * e * /mm0 *  mm1 *  q2
         + /me *  a18 * e *  mm0 *  mm1 *  q2
```

Das Steuersignal Q2 und auch noch zwei Pseudoadressen zu Selektion eines von 4 Blocks mit 256K innerhalb eines 1MB-Blocks im Erweiterungsspeicher werden mit einem zusätzlich bereitgestellten Latch 74LS174 erzeugt. Dieses Latch wird durch einen OUT-Befehl an Port D7H beschrieben. D0 und D1 geben darin die Pegel der Pseudoadressen A18 und A19 und D2 den Pegel von Q2 vor.

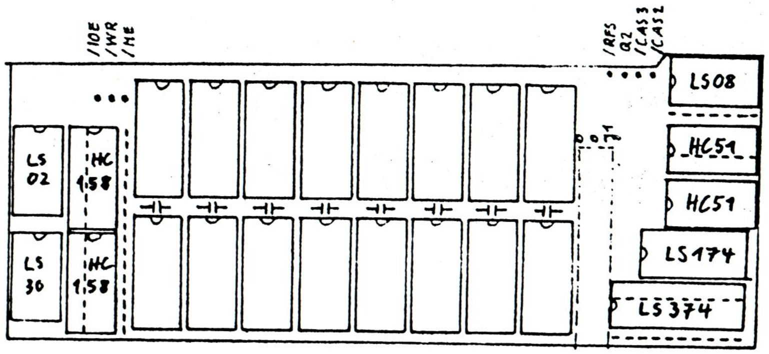
Die gesamte Erweiterungsschaltung incl. RAMs kommt auf einem Huckepackboard unter, das in die Sockel der Adreßmultiplexer Z17 und Z18 (74HC158) sowie Z11 (74HC51) und des Datenlatch Z15 (74LS374) der Druckerschnittstelle gesteckt wird und diese ICs mit aufnimmt. Damit stehen alle gemultiplexten Adressen, die Daten, die Steuersignale /RAS, MUX und /MUX auf dem Huckepackboard zur Verfügung. Die Pseudoadressen A18' und A19' werden mit einem zusätzlichen 74HC51-Gatter gemultiplext.

Es müssen dann nur noch folgende Signale durch freie Verdrahtung zwischen Huckepackboard und CPU-Board verbunden werden:

/IOE (74LS02, Pin6)	von Pin5 des Z7, 74ALS245
/WR (RAMs, Pins2)	von Pin6 des Z7, 74ALS245
/ME (RAMs, Pins3)	von Pin2 des Z7, 74ALS245
/RFS (74LS08, Pins1,13)	von HD64180, Pin57
Q2 (74LS174, Pin5)	an Pin23 des PAL
/CAS3 (74LS08, Pin12)	von Pin18 des PAL
/CAS2 (74LS08, Pin1)	von Pin13 des PAL

Und wofür ist so viel Speicher gut? Natürlich wird der erstmal als RAM-Floppy genutzt. Anstatt diesen Speicher zusammen mit den restlichen 320K, die im Grundspeicher schon als RAM-Floppy genutzt werden, als eine RAM-Floppy zu betreiben, soll hier eine zweite RAM-Floppy von 2MB Kapazität eingerichtet werden. Das vereinfacht die Software etwas, wenn nicht ständig geprüft werden muß, ob ein zu übertragender Sektor nun auf der Erweiterung oder im Grundspeicher zu holen ist. Der benötigte Treiber läßt sich aus dem für die 'alte' RAM-Floppy bestehenden Treiber ohne große Änderungen umformen. Es kommt beim Login nur hinzu, daß MM1 auf high gesetzt werden muß.

Bei den Lese- und Schreib-Routinen ist noch nachzurüsten, daß der in (@trk) übergebene Wert aufzuteilen ist in D0 und D1, die zusammen mit einem gesetzten D2 das Adreßzusatzbyte des DMAC ergeben, und daß D2-D4 nach zweimaligem Rechtsshiftenals D0-D2 an den Port D7H ausgeben werden. In der Login-Routine der 'alten' RAM-Floppy sollte vorsichtshalber grundsätzlich MM1 low gesetzt werden.



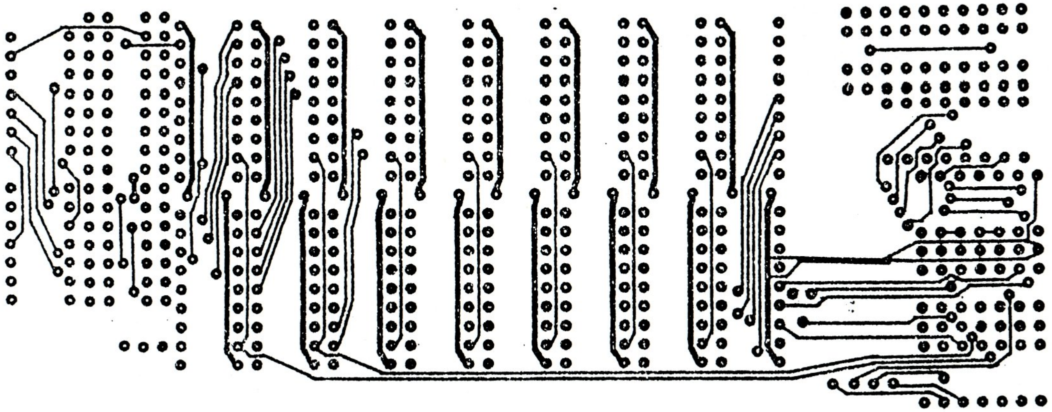
Bestückungs- und Anschlußplan; J1 bleibt bei 1MB-Chips offen

Folgende Datenstrukturen sind für das BDOS vorzugeben:

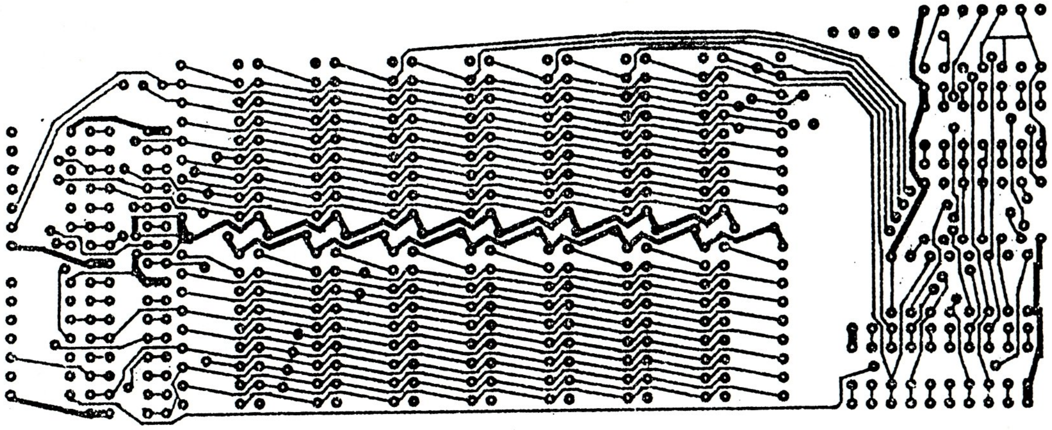
```
        cseg            ;Drive Table
@dtbl   dw      fdsd0,fdsd1,fdsd2,fdsd3 ; 4 Floppy Drives
        dw      fdram                   ; 'alte' RAM-Floppy
        dw      f2dram                  ; 2MB-RAM-Floppy
        dw      hdsk0,hdsk1;hdsk2,hdsk3 ; 4 Harddisk-Partitionen
        dw      0,0,0,0,0,0

        dseg            ;dph's koennen in Bank 0 liegen
;       extended disk parameter header 2MB-RAM-Floppy
        dw      f2write         ;Routinen, um Sektoren
        dw      f2read          ; zu schreiben und zu lesen
        dw      f2login         ;2MB-RAM-Floppy einloggen
        dw      f2init          ;Initialisierung
        db      0,0             ;kein Sektor-Skew
f2dram  dph     0,dpbr2         ;Zeiger auf DPB

        cseg            ;disk parameter block fuer 2MB-RAM-Floppy
dpbr2   dpb     1024,64,2048,4096,512,0,8000h
                ;1024 Bytes/Sektor,  64 Sektoren/Track,
                ;2048 Tracks, Blockgröße 4K, 512 DIR-Entries,
                ;keine Systemspuren, unremovable media
```

Da diese 2MB (wie das sonst meistens bei RAM-Floppies für 8Bit-Computer aussieht) nicht I/O-mapped sind, kann man darin auch Programme laufen lassen. Es liegt dann allerdings am Können des Programmierers, mit dem BDOS ein Abkommen zu treffen, daß dieses nach dem Abarbeiten einer BDOS-Funktion wieder ordnungsgemäß die Bank und auch noch die entsprechende 256K-Portion einschaltet, in der das Programm den BDOS-Call abgesetzt hat.

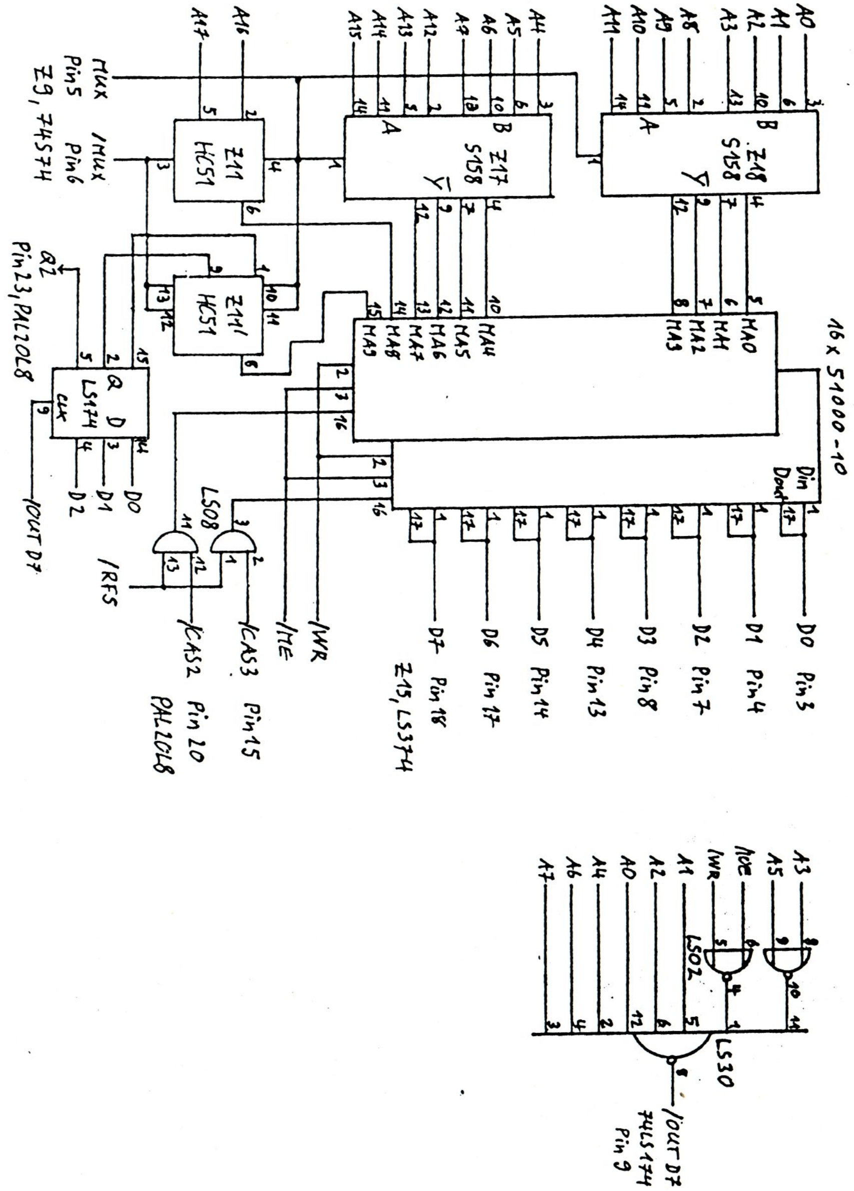
Dieses Problem sei an die Soft-Cracks weitergegeben.



Layout der Bestückungsseite der 2MB-Erweiterung



Layout der Lötseite der 2MB-Erweiterung



Club 80 Börse --- Club 80 Börse

ich habe einen CP/M-Rechner abzugeben:

Morrow MD10 (Micro Decisions) mit 2 Laufwerken 5.25" und bei Bedarf etwas Software (Turbo Pascal o.ä.) Preis: VHB 150,- DM

Sven Krahn Preßburger Str. 5 D-7500 Karlsruhe 41

Mit freundlichem Gruß,

Suche folgende Programme für Genie II <**40 Spuren DD**> (gegen Tausch oder Bezahlung):

"**Warrior of RAS Trilogy**" von MED Systems:
- Volume I DUNZHIN
- Volume II KAIV
- Volume III THE WYLDE

"**DunjonQuest**"-Serie von Automated Simulations (EPYX):
- Hellfire Warrior
- Keys Of Acheron
- Sorcerer Of Siva
- Temple Of Apshai
- Upper Reaches Of Apshai

"**MACES & MAGIC**"-Serie von Adventure International:
- Balrog
- Morton's Fork
- Reign Of The Red Dragon und Hidden Valley

Data-Writer von Software Options Inc.

Copy Art II von Simutek

WordStar (Vers. 3.00) von Micropro unter NewDOS/80

sowie **L-DOS 5.1.1** und **MultiDOS** unter Double Density ...

Bitte melden bei Jens Günther (Tel.: 02681 / 1553)

Und für die Börse habe ich auch etwas. Da mir mein TRS-80 Mod 4P genügt, möchte ich mich von meinen Speedmaster (80 Zeichen-, Ton-, Uhren- und RS 232-Karte) trennen. Mitgeliefert werden: Monitor, Laufwerkgehäuse mit 1x 40 Trk SS - 1x 40 Trk DS - 1x 80 Trk 5.25 Lw, 1x 8" SS Lw und 8" DS Lw, eine Typenradschreibmaschine (Juki 2200 mit Paralell-Interface), Software satt (8" und 5.25 Disk) und Literatur. Für 500.- DM geht alles an einen Selbstabholer weg.

Wilhelm Tornow
Görlitzer Str. 16

K Ü N D I G U N G

Hiermit kündige ich meine Mitgliedschaft im CLUB 80 zum Ende dieses Jahres.

Grund: Ich gebe auf.

Gleichzeitig biete ich meinen MZ 3541 zum Kauf an.

Und hier einige technische Einzelheiten zu diesem Gerät:
Es handelt sich um einen Personal-Computer mit Z80-CPU und einem Arbeitsspeicher mit einer Größe von 256 KB, von denen bis zu 176 KB als schnelle RAM-DISK verwendet werden können.
Zusätzlich verfügt das Gerät über einen Gafikspeicher von 96 KB. Dadurch ist es möglich, Grafik mit einer Auflösung von 640 mal 400 Punkten und 8 Farben auf dem Bildschirm darzustellen. Es können 4 Bildschirmseiten (3mal Grafik und 1mal Text) gleichzeitig oder einzeln oder auch gemischt auf dem Bildschirm ausgegeben werden. Die Grafikbildschirmseiten können nicht gescrollt werden.
Es sind 2 Stück Doppelseitige 5,25"-Laufwerke mit je 385 KB eingebaut.
Ferner sind vorhanden: eine parallele Druckerschnittstelle nach Centronics-Norm, eine serielle Schnittstelle sowie zwei Bildschirmausgänge, die über getrennte Kanäle bedient werden können. Außerdem befindet sich auf der Rückseite noch ein Anschluß für zwei externe Minifloppy-Laufwerke.
Zum Lieferumfang gehört ein SW-Monitor (grün).
Zwei Betriebssysteme gehören auch dazu:
"FDOS", ein reines Basic-System.
Und "EOS-V3", ein schnelles CPM-kompatibles Betriebssystem.
Von beiden Betriebssystemen werden die grafischen Fähigkeiten des Rechners mit einem komfortablen Befehlssatz voll unterstützt. Die mitgelieferten Handbücher sind als Lehrbücher für Anfänger bestens geeignet.

Wer Interesse daran hat, sollte sich das Gerät vorher bei mir anschauen. (Nach tel. Vereinbarung: 04193/6125)

Ich denke, DM 1000.-- ist ein angemessener Preis.

Henstedt-Ulzburg, den 26.10.1990
Mit den besten Grüßen
Harald Hirsekorn

Harald Hirsekorn
Beckersbergring 131
2359 Henstedt-Ulzburg 1
Telefon (0 41 93) 61 25

Sehr geehrter Herr Chwolka,

da ich mich beruflich mit MS/DOS und UNIX beschäftigen muß, habe ich keine Zeit mehr für meinen CP/M-Rechner.

Ich verkaufe deshalb:

Hardware:	
1 Z80 Rechner	alphaTronic PC mit eingebauter Bicom-Graphic
1 Diskettenlaufwerk	Original TA F1(TEAC) mit Controller für 2 Laufwerke
1 Laufwerkgehäuse	Original TA für F2
1 Farbmonitor	PRISM QL14
Software:	
1 Betriebssystem	TA-PC Disc-Basic V. 5.26B ■
1 Betriebssystem	CP/M 2.2/3.1 + CP/M 2.2 mit Super BIOS
1 Programmier-Sprache	Microsoft Basic ■
1 Programmier-Sprache	Turbo Pascal 3.0 + TURBO TUTOR 1.0 ■
1 Textprogramm	Wordstar 3.0 ■
1 Datenbank	dbase II ■
	(■ mit Handbuch)

Kurt Klucina
Flurstraße 4
8891 Adelzhausen
Tel.: 08258-348

Gesamt-Preis: DM 300,-

Club 80 Börse --- Club 80 Börse

Weiterhin verkaufe ich verschiedene Bücher:

Das alphaTronic PC-Buch (Basic-Lehrgang)	Erhard Unger	Heim	DM 15,-
BASIC mit dem AT-PC Band1	S. Lumma	Heim	DM 18,-
BASIC mit dem AT-PC Band2	S. Lumma	Heim	DM 18,-
alphtronic PC BASIC Handbuch	Karl-Heinz Hauer	SYBEX	DM 10,-
Assembler-Tricks am AT-PC	Patrick V.Thomas	Heim	DM 15,-
TAalpaTronicPC Almanach 85/86 (Band 1 und 2)	H.Stöber	Stöber	DM 18,-
Vom Umgang mit CP/M	Bernd Pol	IWT	DM 20,-
Betriebssystem CP/M	Plate	Franzis	DM 25,-
Programmieren mit CP/M	Alan R.Miller	SYBEX	DM 20,-
computer compact CP/M	Joseph Reymann	Goldmann	DM 4,-
CP/M kompakt	Plate	Franzis	DM 4,-
dbase II Band1 Einführung	W.Eggerichs	Hüthing	DM 15,-
dbase II Band3 Aufbau und Nutzung v. Datenbanken	W.Eggerichs	Hüthing	DM 15,-
Mikrocomputer Grundkurs Turbo Pascal	Kaier/Rudolfs	Vieweg	DM 12,-
Mikrocomputer Grundkurs Turbo Pascal Übungen	Kaier/Rudolfs	Vieweg	DM 10,-
Mikrocomputer Aufbaukurs Turbo Pascal	Kaier/Rudolfs	Vieweg	DM 20,-
Turbo Pascal 3.0 QUICKMANUAL	L.Lessner	MERKUR	DM 12,-
Wordstar-Tuning	Werner Borsbach	HEISE	DM 10,-
Hardware-Erweiterungen für Z80-Rechner	Martin Aschoff	Vogel	DM 12,-
Microsoft-Basic: Konzepte, Algorithmen, Datenstrukturen	Röckrath	Franzis	DM 12,-

Gesamt-Preis DM 250,-

F&S VERTRIEB
computer division

Ihr Partner für
Hard- und Software

Telefon 05042 1253
05042 4215

An
Club 80
Herrn Schmid
St. Cajetan Str. 38
W-8000 Muenchen 80

20. Oktober 1990

SONDERRABATT FÜR COMPUTERCLUB-MITGLIEDER

Sehr geehrte Clubmitglieder,

Aus aktuellem Anlaß machen wir Ihnen als Computerclub ein Angebot:

Wir wollen Ihr Clublieferant werden !

Wir haben Ihnen eine Auswahl aus unserer derzeit gültigen Preisliste für Hard- und Software zusammengestellt, die wir aufgrund der hohen Qualität und des guten Preis-Leistungsverhältnisses empfehlen möchten.
Für Sammelbestellungen aus dieser Preisliste gewähren wir Ihnen

5% SONDERRABATT !

Sammelbestellungen sind schriftliche Bestellungen von drei Verpackungseinheiten oder mehr. Dieses Angebot ist bis auf Widerruf gültig.

Auf gute Zusammenarbeit

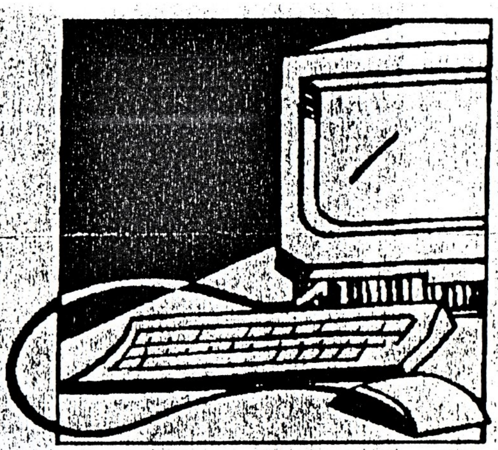
Fiedler & Schuster
-computer division-

P.S. Sollten Sie einen Artikel auf unserer Preisliste vermissen, schreiben Sie uns. Wir besorgen Ihnen jeden Artikel zu dem günstigen Preis-Leistungsverhältnis, für das wir bekannt sind.

Bankverbindung:

100 DISKETTEN

	NONAME	BASF EXTRA
5.25" 2 D 360 KB	DM 69,-	159,-
5.25" HD 1.2MB	DM 139,-	289,-
3.5" 2 D 720 KB	DM 139,-	289,-
3.5" HD1.44MB	DM 299,-	599,-

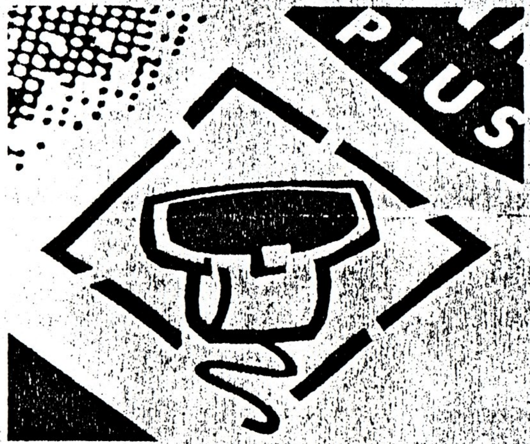


10 BASF CARTRIDGES

DC 300 L	DM 439,-
DC 450 L	DM 475,-
DC 450 LS	DM 489,-
DC 600 H	DM 539,-
DC 600 HS	DM 629,-
DC 2000 L	DM 489,-

5 DRUCKER FARBBÄNDER

Epson FX80 (G635) / STAR LC 10	DM 49,-
Epson LQ400 /LQ500/LQ550 (G633)	DM 69,-
NEC P6+/P7+	DM 79,-
NEC P2 / P6 (G666)	DM 79,-



MÄUSE + SCANNER

SCANMAN PLUS DM 419,-
mit PAINTSHOW PLUS: Deutsche Originalversion!

LOGIMOUSE S9 DM 199,-
Hochauflösende Profi - Maus mit Software POP-UP DOS und MOUSEWARE. Als BUS, SERIELL oder PS/2 Version, Auflösung 320 dpi!

LOGIMOUSE PILOT DM 119,-
Profi - Maus mit Software PAINTSHOW PLUS und MOUSEWARE in deutsch ! Serielle Version für den normalen und einfachen Anschluß an Ihren Rechner. Auflösung 200 dpi!

LOGITECH TRACKMAN DM 299,-
Hochauflösende stationäre Profi - Maus mit Hochleistungssoftware. Auflösung 50 bis 15 000 dpi!

LOGI FINESSE VERSION 3.1 DM 499,-
Das Desktop Publishing Programm, mit dem dieses Blatt erstellt wurde!

Sonderpreisliste gültig ab 01. Oktober 1990
Kurzfristige Preisänderungen müssen vorbehalten bleiben. Lieferung an Einzelpersonen nur gegen Vorkasse oder Nachnahme. Versandkosten werden extra berechnet. Bitte bestellen Sie nur schriftlich, Postkarte oder kurze Mitteilung genügt.

Fiedler & Schuster GbR
HARD u. SOFTWARE Vertrieb
POSTFACH 1116
3252 BAD MÜNDER 1

COMPUTER

PT-AT TOP

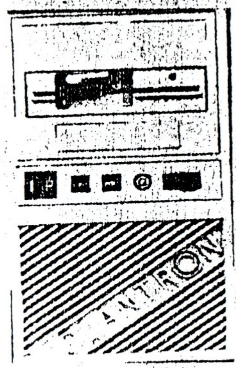
- 80286 CPU
- 8/12 MHz (0 Wait State)
- 1 MB RAM
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Festplattenlaufwerk 42 MB 28 ms
- Große deutsche Tastatur

NUR DM 2899,-

PT-386 SX/20-1 Tower

- 80386/SX CPU
- 10/20 MHz (0 Wait State)
- 1 MB RAM Neat
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Festplattenlaufwerk 42 MB 28 ms
- Große deutsche Tastatur

NUR DM 3898,-



PT386 25 CACHE Tower

- 80386-25 CPU
- 25 MHz
- 2 MB RAM
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Diskettenlaufwerk 1.44 MB, 3.5"
- Festplattenlaufwerk 65 MB 28 ms
- Große deutsche Tastatur

NUR DM 6898,-

PT386 33 CACHE Tower

- 80386-33 CPU
- 33 MHz
- 2 MB RAM
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Diskettenlaufwerk 1.44 MB, 3.5"
- Festplattenlaufwerk 65 MB 28 ms
- Große deutsche Tastatur

NUR DM 8498,-

386-16 SX Tower AT

- 80386-16 SX CPU
- 12/16 MHz
- 1 MB RAM (0 Wait State)
- 4x16 Bit und 1x8 Bit Erweiterungsslots
- Monochrom-Grafikkarte, Herkules-kompatibel
- 2 parallele Schnittstelle, 2 serielle Schnittstellen
- Marken Diskettenlaufwerk 1.2 MB, 5.25"
- Marken Diskettenlaufwerk 1.44 MB, 3.5"
- FDD/HDD Controller
- MF-II kompatible Tastatur mit 102 Tasten

NUR DM 2799,-

DISKETTENLAUFWERKE

5.25" 360KB	179,-
5.25" HD 1.2MB	199,-
3.5" 720KB	189,-
3.5" HD 1.4MB, mit Rahmen	199,-

FESTPLATTEN

VOM MARKTFÜHRER SEAGATE

ST 225	21.4 MB		409,-
ST 238R	32.7 MB	70 ms	499,-
ST 251-1	42.8 MB	28 ms	699,-
ST 296N	85 MB	28 ms	999,-
ST 124	21.4 MB	40 ms	649,-
ST 125-1	21.4 MB	28 ms	669,-
ST 157N	48.8 MB	28 ms	799,-
ST 151	43.5 MB	28 ms	799,-

DRUCKER

NEC PINWRITER P6 PLUS	1399,-
NEC PINWRITER P7 PLUS	1599,-
CITIZEN SWIFT 24	899,-
HP LASERJET II P	2999,-
HP LASERJET III	4899,-

MONITORE

14" TTL FLAT MONITOR AMBER	299,-
14" TTL FLAT MONITOR PAPERWHITE	299,-
14" VGA MONO MONITOR	399,-
14" VGA COLOR MONITOR	799,-
14" NN MULTISCAN MONITOR	999,-
14" NEC MULTISYNC 3D	1799,-
14" NEC MULTISYNC 2A	1299,-

GRAFIKKARTEN

KOMPATIBEL ZU IBM'S CGA,MGA,EGA,VGA

VGA KARTE 256KB		199,-
JUKO VGA PREMIUM	16 BIT 256KB	299,-
JUKO VGA PREMIUM II	16 BIT 512KB	399,-
TVGA VGA TRIDENT	16BIT 512KB	299,-

Alle Preise sind in DM und enthalten die gesetzl. Mehrwertsteuer. Versandkosten werden extra in Rechnung gestellt.

Kurzfristige Preisänderungen müßen leider vorbehalten bleiben. Lieferung solange Vorrat reicht.

EMERSON ACCUCARD

Notstromversorgung für jeden XT/AT 286-386-486 oder kompatiblen.

2/3-lange 8-Bit Slotkarte, die bei einem Stromausfall automatisch ein Backup fährt und alle Daten, die im Augenblick genutzt werden, auf die Festplatte sichert, und dann das System abschaltet. Kommt der Strom wieder, kommen auch Ihre Daten wieder! Nach der normalen Startprozedur des Rechners wird automatisch ein Restore gefahren und das System nimmt seinen Dienst an exakt der Stelle wieder auf, wo es zuvor durch den Stromausfall unterbrochen wurde!

Stück DM 999,-

Der Laserdrucker zum Preis eines Matrixdruckers!

Qume CrystalPrint WP Plus

das preiswerte Einstiegsmodell - entwickelt für die Anwender, die einen kleinen und leisen Arbeitsplatz-Drucker benötigen - zu einem Preis, der trotz perfekter Schriftqualität und Grafikfähigkeit in einer vernünftigen Relation zum Rechner steht. Durch Flüssigkristall-(LCS) Technologie beträgt die Stellfläche der Drucker nur noch ca 50 % derer von normalen Laserdruckern. Auch die Geräuschentwicklung liegt niedriger.

Crystal-Print WP Plus

Nur DM 1999,-

286-12 Desktop

Intel 80286-12 mit 6 und 12 MHz (Landmark 16), 1 MB RAM bestückt, onboard erweiterbar bis 4 MB RAM EMS-kompatibel, Sunlac Chip-Set, extended Bios mit Set-Up, 1 Floppydisklaufwerk 1,2 MB (Markenfabrikat), 1 Floppydisklaufwerk 1,44 MB (Markenfabrikat), FDD/HDD Kombi Controller 1:1 mit Kabelsatz, Echtzeituhr, Schnittstellen: 1 x parallel, 1 x seriell und Gameport, Monochrom-Grafikkarte mit Printerport, 6 x 16 Bit und 2 x 8 Bit Erweiterungsslots, 200 Watt Schaltnetzteil, Qualitätstastatur KT 2000 mit 102 Tasten, verwindungsstabiles Desktopgehäuse, Reset- und Turboschalter, Speed-Display, Ringbuchschuber, 80 seitiges deutsches Handbuch, ZZF-Zulassung, DR. CETERA DOS 3,41, lizenzierte Version des aktuellen DR. DOS 3,41, komplett mit deutschen Handbüchern, wahlweise auch auf 3,5" FD. Inklusive Norton Commander 2.01.

PREIS NUR 1599,-

Neue Drucker braucht das Land

Nachdem mir mein RX-80 seit 6 Jahren treue Dienste geleistet hat und noch immer keine Neigung zeigt, kaputt zu gehen, habe ich ihn nun zwangsbeurlaubt und mir einen HP Deskjet 500 gekauft. Warum nun ausgerechnet den ?
Mein Forderungskatalog sah folgendermaßen aus:

- 24-Nadleln (wenn schon, denn schon)
- automatischer Einzelblatteinzug (muß einfach sein, s.u.)
- leise (Eltern schlafen nebenan)
- gute Schriftqualität (fast selbstverständlich)
- gute Verarbeitung (auch klar)
- nach Möglichkeit Epson-kompatibel (Tscrips ade)

Also bin ich von Geschäft zu Geschäft gepilgert und habe mich umgehört. 24-Nadler gibt es zwar schon ab rund 500 Mark, aber deren Verarbeitung und auch die Einzelblatteinzüge sahen nicht besonders vertrauenerwechend aus und wenn man dauernd einen Papierstau hat, kann man gleich beim Handbetrieb bleiben. Warum überhaupt einen automatischen Einzug ? Ganz einfach, weil mir die Einspannerei per Hand im Laufe der Zeit gewaltig auf die Nerven gegangen ist. Wenn man ein File mit 50 Seiten ausdruckt und nicht weggehen kann, weil man jede Minute eine neue Seite nachlegen muß ist das einfach lästig. Der Einzelblatteinzug ist beim HP übrigens schon fest eingebaut, sodaß die Konstruktion solide und sicher ist. Verarbeitet werden zwar nur A4-Blätter (die per Befehl auch quer bedruckt werden können !), aber ich weiß schon gar nicht mehr, wann ich das letzte mal auf Endlospapier gedruckt habe. Briefumschläge schluckt er, nebenbei bemerkt, auch.
Die nächste Frage war, ob Nadeln oder Tintenspritzer. Von der Qualität her geben die sich nicht viel, obwohl beim HP auch im Draft-Modus keine Sägezähne, wie bei allen Nadeldruckern, die ich kenne, im Schriftbild zu erkennen sind. Der Ausdruck ist lediglich etwas heller und auf Recyclingpapier sogar besser zu lesen, weil die Schrift im Schönschriftmodus dort wegen der größeren Tintenmenge und der rauheren Oberfläche fast verläuft. Auch braucht man längst kein spezielles Tintendruckpapier mehr, normales Kopierpapier tut es genauso. Ein großer Vorteil ist aber einmal die Geräuscharmut und zweitens, daß einem keine Nadeln abbrechen oder sich verbiegen können. Wenn die Tintenpartone leer ist wird sie mitsamt dem Druckkopf wegeworfen und man hat wieder einen völlig Neuen. Im Gegensatz zu anderen Druckern, z.B. von Epson, wird nichts nachgefüllt, sodaß auch keine Schläuche verstopfen können. Bei den billigeren Nadeldruckern kostet ein neuer Kopf mindestens die Hälfte des Neupreises und ob sich das dann bezahlt macht, ist eine andere Frage. Da es kein Farbband gibt, daß immer mehr ausgelutscht wird, bleibt die Druckqualität natürlich bis zum Ende immer gleichmäßig und gut.
Ein besonderes Bonbon, womit schließlich auch der letzte Punkt erfüllt wäre, ist die Epson-Emulations-Kassette, die einfach in einen der beiden Erweiterungsschächte gesteckt wird. Mit dieser Kassette emuliert der HP einen FX-80 und zwar so gut, daß man keine weißen Streifen im Grafik-Ausdruck hat ! Ich habe es selber

ausprobiert, es funktioniert. So kann man alle alten Programme ohne irgendwelche Softwarekrücken (Arnulf, nimm's bitte nicht persönlich) weiterverwenden, die sich nicht oder nur schwer an andere Drucker anpassen lassen (TSCRIPS usw.). Der HP selber ist zum LaserJet kompatibel, für den sich in neueren Programmen mit Sicherheit ein Treiber findet. Bis auf die Tatsache, daß er wegen des fehlenden RAMs keine ganzen Grafikseiten im Speicher aufbauen kann, kann er den Laserdrucker vollständig ersetzen, sogar die Auflösung mit 300 dpi ist identisch.
Im Prinzip könnte man sich dann ja gleich einen Laser kaufen, die billigsten gibt es schon für knapp 2000 Mark, aber bei denen sind die Verbrauchskosten nochmal doppelt so hoch, wie beim Tintenspritzer und der kostet schon rund das doppelte eines Nadeldruckers.
Zusätzlich zu den eingebauten drei Schriftarten gibt es natürlich noch Schriftkassetten, man kann aber auch Fonts downloaden. Features wie Proportionalschrift usw. sind ja sowieso selbstverständlich.
Leider hat auch der beste Drucker seine Schattenseiten, das 'Handbuch' ist nämlich nur eine bessere Referenzkarte. Scheinbar geht HP davon aus, daß der moderne DOS-Geschädigte den Drucker sowieso nur aus den diversen Softwarepaketen heraus per Menü anspricht und nichts selber machen will oder kann. Beispiele sucht man weitgehend vergebens und gerade die Grafik wird sehr stiefmütterlich behandelt.
Wenn Ihr Euch den HP in einem Geschäft mal anseht, paßt auf, daß man Euch den richtigen zeigt. Letztes Jahr im Oktober war erst ein Test des HP DeskJet in der c't, etwas später kam der DeskJet Plus und seit kurzem gibt es schließlich den DeskJet 500. Ich glaube zwar nicht, daß es bis zum Erscheinen des Infos schon wieder einen neuen gibt, aber möglich ist alles. Darum Vorsicht, mir hat man auch versucht, noch den Plus anzudrehen, obwohl der nur eine eingebaute Schrift hat und sogar teurer ist, als der 500. Ich habe für den Drucker 1450.- gezahlt und die Epson-Emulation kommt auf 200.- Mark.

Alexander Schmid

DFÜ — was bringts?

Vor einigen Monaten bin ich, warum eigentlich weiß ich heute selbst nicht mehr zu sagen, in die Übertragung von Daten über Telefon eingestiegen. Am Anfang war DFÜ für mich Selbstzweck, sprich ich betrieb die Datenübertragung der Datenübertragung wegen. Mittlerweile hat sich das aber erheblich geändert. Inzwischen ist dieser Nebenzweig der Computerei aus meinem Tagesablauf gar nicht mehr wegzudenken. Heute leere ich mein elektronisches "Postfach" in der Mailbox genauso selbstverständlich, wie meinen Briefkasten und mein Postfach im Postamt und nutze die vielfältigen Möglichkeiten intensiv.

Dabei beschränken sich meine Aktivitäten (zur Zeit noch) nur auf den "Besuch" von Mailboxen und den Gastzugang zu BTX. Die Möglichkeiten von DATEX-P sind mir zur Zeit noch aus finanziellen Gründen verschlossen. Demnächst werde ich hoffentlich auch vollwertiger FIDO*-Teilnehmer sein, zur Zeit funktioniert aber mein Mailerprogramm noch nicht richtig!

Mailboxen bieten aber heutzutage schon so viel, daß man durchaus auch auf BTX (geht eh nur mit C64, ATARI, PC oder speziellen BTX-Geräten), DATEX-P und FIDO (bzw. andere Netze) verzichten kann. Neben den öffentlichen Brettern (PIN's genannt) und dem privaten Postfach gibt es vor allem Downloadareas für viele verschiedene Rechner.

Um euch einmal eine Übersicht über die Vielfalt der öffentlichen PIN's zu geben, hier eine Liste aus der BRAINSTORM, meiner "Heimatbox":

BUECHER.PIN	BUECHER	POP.PIN	POP-MUSIC
ANDY_AMI.PIN	SW-TIPS AMIGA	VIREN.PIN	VIREN
HACKER.PIN	NICE HACKING	UMWELT.PIN	UMWELT
KONTAKTE.PIN	KONTAKTE	ESOTERIK.PIN	ESOTERIK
DFUE.PIN	DFUE/FIDO	ANDY_AND.PIN	SW-TIPS DIVERSE
ANDY_MS.PIN	SW-TIPS MS-DOS	DATEX_P.PIN	DATEX-P
MS_DOS.PIN	MS-DOS ECKE	NEWSYSOP.PIN	NEUES VOM SYSOP
POLITINL.PIN	POLITIK	NUTS.PIN	- NUTS -
SOFTWARE.PIN	SOFTWARE-TIPS	KASSE.PIN	KASSE BRAINSTORM
ALLGMAIL.PIN	ALLGEMEINE MAIL	SUCHE.PIN	SUCHE/TAUSCHE
ATARI.PIN	RUND UM ATARI	TIERVERS.PIN	TIERVERSUCHE
TIPSBAU.PIN	HARDWARE TIPS/BAUANLEITUNGEN	WITZE.PIN	LACH-HAFT
DX.PIN	DX-PIN	MECKER.PIN	MECKERECKE
ADALLG.PIN	ALLERLEI/ADVENTURES	GREENPEA.PIN	GREENPEACE
AMIGA.PIN	AMIGA ALLGEMEIN	HEAVY.PIN	HEAVY NEWS/VORST.
CD-EXTRA.PIN	CD CHARTS TOP 15	PDIVERS.PIN	ALLERLEI (P)
LABER.PIN	- RABARBERSUELZ -	GERUECHT.PIN	GERUECHTE-KUECHE
PSTORY.PIN	GESCHICHTEN FÜR ERWACHSENE	VERKAUF.PIN	VERK./VERSCH.
MVA.PIN	MUELLVERBRENNUNGSANLAGEN		

Wie jeder echte Z80-Freak sofort feststellen wird, gibt es tatsächlich eine Menge interessanter Sachen, aber nichts für sein 8-Bit-Herz. Wie mir aber erst gestern Gerd Neebe am Telefon sagte, gibt es in vielen Mailboxen noch CP/M-Bretter. Meine Wahl fiel nur deshalb auf die BRAINSTORM, da sie in meinem Ortsbereich liegt und damit im 8- bzw. 12-Minutentakt zu erreichen ist. Mit den Downloadbereichen, aus denen man oft sehr gute Public Domain Programme herauskopieren kann, sieht es ähnlich aus. Fast in allen Boxen findet man Programme für MSDOS, ATARI und C64, aber auch CP/M-Programme gibt es noch häufiger, als man sich das vorstellt!

Daß durchaus auch nicht computerspezifische Informationen über Mailboxen laufen, kann man an der oben abgedruckten PIN-Liste schon sehen. Daß es aber auch Bereiche gibt, die man nicht (oder doch zumindest ich nicht) in Mailboxen vermuten würde, zeigt die Box "Lebensinterface" in Aalen. Dort gibt es unter anderem eine Bibeldatenbank "Neues Testament" mit folgenden Unterpunkten:

*FIDONet ist ein privates, internationales Rechnernetzwerk, das derzeit (März '90) aus etwas mehr als 5000 Rechnern ("Nodes") besteht.

<1>	Matthäus - Evangelium	<2>	Markus - Evangelium	<3>	Lukas - Evangelium
<4>	Johannes - Evangelium	<5>	Apostelgeschichte	<6>	Römer
<7>	1. Korinther	<8>	2. Korinther	<9>	Galater
<A>	Epheser	<B>	Philipper	<C>	Kolosser
<D>	1. Thessalonicher	<E>	2. Thessalonicher	<F>	1. Timotheus
<G>	2. Timotheus	<H>	Titus	<I>	Philemon
<K>	1. Petrus	<L>	2. Petrus	<M>	1. Johannes
<N>	2. Johannes	<O>	3. Johannes	<P>	Hebräer
<Q>	Jakobus	<R>	Judas	<S>	Offenbarung

Wie ihr seht, ist das Angebot vielfältig, teilweise sogar unüberschaubar, auf jeden Fall aber sehr interessant!

Für mich ist, ich habe es oben schon erwähnt, der Bereich der Übertragung "privater" Post von großer Wichtigkeit. Als kleines Beispiel hier eine Nachricht, die mir Hans Günther Hartmann kürzlich zukommen lies:

MsgNr: 1
*MB*1002/HANS-G HARTMANN am 23.11.90 um 09:13:50*

Betreff: zcpr3 config fuer 512k RAM

Moin Hartmut.
Ich habe dir in die priv softbiblio eine zip-Datei gelegt.
Hoffentlich kannst Du sie entzippen!
Wenn nicht, kurze msg. Falls Du den Fehler in den dateien findest,
die das Booten auf dem 4p verhindern------> auch kurze msg.
Gruss
Hans-G, UID 1002

Wie man der Nachricht entnehmen kann, ist es auch möglich, eigene Programme oder Daten ohne Zugang für andere Mailboxnutzer zu übertragen. Solche Nachrichten, offline geschrieben (d.h. mit einem Texteditor erstellt, bevor man die Mailbox anruft) und dann übertragen, sind oft billiger und auf jeden Fall schneller als Briefpost.

Für den Zugang zu einer Mailbox braucht man, neben dem Computer, ein Modem (direkt ans Telefonnetz anschließbar) oder einen Akustikkoppler ("Anschluß" über den Telefonhörer) und ein Datenübertragungsprogramm. Dabei würde ich, wegen der besseren Übertragungssicherheit, ein Modem jederzeit einem Akustikkoppler vorziehen. Leider gibt es zur Zeit prinzipiell nur zwei Alternativen beim Modem-Kauf. Entweder illegal (weil ohne Postzulassung) und billig (um ca. 250-300 DM) oder legal und doppelt bis dreifach so teuer. Zum Glück gibt es unter CP/M wenigstens genug Übertragungsprogramme zum Nulltarif (z.B. KERMIT und MODEM7). Bei der Wahl des Terminalprogramms sollte man aber darauf achten, daß das Programm wenigstens das XMODEM-Protokoll (Übertragungsprotokoll) beherrscht. Dies ist bei einigen älteren Programmen unter NewDOS nicht der Fall!

Für alle die, die schon DFÜ betreiben und all jene, denen ich den Mund wässrig gemacht habe und die demnächst in die Datenfernübertragung einsteigen, hier die Nummer und die Parameter der Mailbox, unter der ich zu erreichen bin: BRAINSTORM 07309/6753, 8N1, 300/1200/2400 User-ID: 1061.

Ich würde mich sehr freuen, wenn ich in meinem Postfach mal eine Message von euch finden würde.

In diesem Sinne allzeit Good Hacking,

Hartmut Obermann

PS: Übrigens gibt es in unseren Reihen sogar einen Mailbox-Betreiber. Klaus Hermann ist SYSOP der DISCOVERY, die unter der Nummer 07127/70107 mit den Parametern 300/1200/2400 N81 zu erreichen ist.

Befehlssatz Hayes-kompatibler Modeme

Der Begriff 'Hayes' ist schnell erklärt. Es handelt sich dabei um einen amerikanischen Modemhersteller, dessen 'Smartmodem' mittlerweile als das Referenzgerät für den Hayes-Befehlssatz gilt. Es ist heute nahezu undenkbar, daß auf dem internationalen Markt ein Modem hergestellt wird, welches nicht diesen De-facto-Standard erfüllt.

Daß der Hayes-Befehlssatz ein Industriestandard wurde, hat seine Ursache in dem einfachen Konzept, das hinter der Modemansteuerung steht. Völlig unabhängig reagiert das Modem auf einfache ASCII-Zeichen, die über die serielle Schnittstelle an das Modem gelangen. Das Modem verfügt über einen Prozessor, der diese Zeichen in Befehle umwandelt. So lassen sich auf einfachste Art Kommunikationsprogramme schreiben. Ein weiterer Grund für die Etablierung des Standards ist die Fähigkeit des Modems, sich selbständig auf die Parameter eines Anrufers einstellen zu können.

Hier nun kurz die Beschreibung der wichtigsten Befehle:

'D' - Dial
Das Modem wählt eine Nummer. Dabei gibt es zwei Spielarten:

- 'DP' Pulse-Dialing (in Europa üblich)

- 'DT' Touch-Tone-Dialing (in den USA üblich)

Beide Wählarten können auch gemischt werden. Der Befehl 'ATDT0,P01188' würde beispielsweise mit Touch-Tone eine Null wählen (Amtsleitung anfordern), kurz warten und danach mit Pulse-Dialing die Fernsprechauskunft anrufen.

'A/' - Wiederholung
Das Modem wiederholt die letzte Funktion. Häufigste Anwendung ist, die letzte Nummer noch einmal zu wählen.

'H0' - Auflegen
Die Verbindung wird abgebrochen, der 'Hörer aufgelegt'.

'H1'- Abheben
Der 'Hörer' wird 'abgehoben', die Leitung erscheint nach außen hin besetzt.

'A' - Antworten
Das Modem sendet einen 'Antwortton' aus. Es versucht eine Verbindung aufzubauen, ohne vorher zu wählen. Nach diesem Befehl befindet man sich nicht mehr im Befehlsmodus, den erreicht man erst wieder über die 'Escape'-Sequenz.

'S0=<n>' - Automatisch antworten
Dieser Befehl schaltet den Automatisch-antworten-Modus ein und aus. Bei n=0 antwortet das Modem nicht auf ankommende Anrufe, jede andere Zahl zwischen 1 und 255 legt fest, beim wievielten Klingeln das Modem 'abhebt' und antwortet.

'+++' - Escape
Nachdem man eine Verbindung aufgebaut hat, befindet sich das Modem nicht mehr im Befehlsmodus. Alle Zeichen, die man nun an das Modem schickt, leitet dieses ins Telefonnetz weiter. Empfängt es jedoch nach einer Sekunde Pause die drei Pluszeichen und danach wiederum für eine Sekunde kein Zeichen, schaltet es sofort in den Befehlsmodus zurück. Nun kann man wieder mit 'AT' Parameter ändern oder einfach den 'Hörer auflegen'.

Die restlichen Befehle und Register-Funktionen sind in der folgenden Tabelle zusammengefasst:

Hayes-Modem-Befehle:

AT	- Beginn eines Hayes-Befehls
+++	- Zurück in den Kommandomodus (vor und nach dem Befehl muß eine Sekunde gewartet werden)
A/	- Automatische Wiederholung des letzten Befehls
D	- Wählen > Ziffern: 0 - 9
	> Symbole: # und * (zur übersichtlichen Gestaltung der Nummer)
	> Befehle: P Pulse-Dialing (Europa)
	T Touch-Tone-Dialing (USA)
	, 2 Sekunden Wählpause
	/ 125 ms Wählpause
	; nach dem Wählen in den Befehlsmodus
	R ein Originate-Modem anrufen
A	- Beantworten eines Anrufs ohne zu wählen
B	- B0 = CCITT V.21, V.22 (Europa)
	B1 = Bell 103/212A (USA)
C	- C0 = Carrier-Signal aus
	C1 = Carrier-Signal an
E	- E0 = Echo aus
	E1 = Echo an
F	- F0 = Halb-Duplex
	F1 = Voll-Duplex
H	- H0 = Modem legt auf
	H1 = Modem hebt ab
I	- I0 = Ausgabe des Produktcodes
	I1 = Ausgabe der Firmware-Version
	I2 = Selbsttestfunktion
M	- M0 = Lautsprecher aus
	M1 = Lautsprecher an, bis Carrier detect
	M2 = Lautsprecher immer an
0	- zurück in Online-Modus
Q	- Q0 = Sende Modem-Antwort an Rechner
	Q1 = keine Antwort Senden
Sr?	- Zeige Wert von Register r
Sr=n	- Setze Register r auf Wert n
V	- V0 = Numerische Antworten
	V1 = Ausgeschriebene Antworten
X	- X0 = Grundantworten (OK und CONNECT)
	X1 = Antworten (CONNECT <Baudrate>)
	X2 = Erkenne "NO DIALTONE"
	X3 = Erkenne "BUSY"
	X4 = Erkenne "NO DIALTONE" und "BUSY"
Z	- Software-Reset

Hayes-Register-Funktionen:

S0	- Anzahl der Klingelzeichen vor dem Abheben (Parameter: 0-255)
S1	- Zähler für die Klingelzeichen (Parameter: 0-255)
S2	- ESCAPE-Zeichen (Parameter: 0-127 ASCII)
S3	- RETURN-Zeichen (Parameter: 0-127 ASCII)
S4	- LINE-FEED-Zeichen (Parameter: 0-127 ASCII)
S5	- BACKSPACE-Zeichen (Parameter: 0-127 ASCII)
S6	- Warte auf Freizeichen (Parameter: 2-255 Sekunden)
S7	- Wartezeit auf einen Carrier nach Wahl oder Antwort
S8	- Pausenzeit für ein Komma (Parameter: 0-255 Sekunden)
S9	- Carrier-Erkennungszeit (Parameter: 1-255 in 1/10 Sekunden)
S10	- Wartezeit zwischen Carrier-Ausfall und dem Auflegen (Parameter: 1-255 in 1/10 Sekunden)
S12	- Wartezeit auf ein ESCAPE-Zeichen (Parameter: 0-255 in 1/10 Sekunden
S16	- Selbsttest des Modems (Parameter: 0-2)

Wer sich etwas mit hayeskompatiblen Modemen auskennt, dem wird auffallen, daß die obrige Liste nicht ganz komplett ist. Das hat seinen Grund darin, daß es inzwischen viele Modeme gibt, die gegenüber dem ursprünglichen 'Smartmodem' von Hayes viele erweiterte Funktionen haben. Vor allem das Kommando 'B' ist bei vielen neueren Modemen erweitert, da sie zusätzliche Normen erfüllen (z.B. CCITT V.23 75/1200 Baud für BTX). Da diese erweiterten Kommandos jedoch nicht genormt sind, wurden nur die Funktionen aufgeführt, die in allen hayeskompatiblen Modemen vorhanden sind.

Der Text und die Liste basieren auf einem Artikel aus der c't 2/88 und erheben nicht den Anspruch der Vollständigkeit! Der Text wurde mehrfach korrigiert und vor allem auf seine sachliche Richtigkeit überprüft. Sollte sich trotzdem ein Tipppfeelrteuffel eingeschlichen haben, bitte ich um Entschuldigung.

Hartmut Obermann

PS: Diesen Text habe ich als Antwort auf eine Anfrage in der BRAINSTORM geschrieben und dort als Textfile in der Download-area DFUE abgelegt. Inzwischen habe ich den Text schon in mehreren anderen Boxen im süddeutschen Raum wiedergefunden, und die Verbreitung wird wohl noch ein bißchen weitergehen, bevor das Thema uninteressant geworden ist (was natürlich in meinem Sinne ist).

Die beiden Terminalprogramme MODEM7 und KERMIT, die es unter CP/M als PD-Software gibt, können beide mit hayeskompatiblen Modemen zusammenarbeiten. Die Geräte gibt es, allerdings ohne FTZ-Prüfnummer und damit zum Betrieb am Netz des Gilb (TELECOM) nicht zugelassen (Anschluß strafbar!), schon ab ca. 250,-- DM.

Adventure

Computerspiel mit vorweihnachtlicher Spielidee. Meist geht es bei dem besinnlichen Game um die aufregende Suche nach einem Schatz und die brutale Vernichtung von möglichst vielen Gegnern - ganz wie bei dem letzten verkaufsoffenen Samstag vor Heiligabend.

AT

Ein durch vielerlei kostspieligen Schnickschnack Ausgebauter Taschenrechner.

Newdos, CP/M, MS-DOS, UNIX

von Gerald Schröder

Nun habe ich Euch lange genug mit (All-)Gemeinheiten über Betriebssysteme gequält. Also jetzt mal was Konkretes. Heute werde ich versuchen, die vier oben genannten Betriebssysteme vorzustellen. Für die meisten von Euch sind das natürlich alte Hüte, aber vielleicht kennt ja nicht jeder alle Systeme. Außerdem werde ich sie aus meiner Sicht vergleichen. "Meine Sicht" bedeutet: Nicht als Anwender, sondern als Programmierer, und besonders als notorischer Nörgler.

[Wer im folgenden TOS (Atari ST) vermißt, sollte einfach "MS-DOS" durch "TOS" ersetzen (und "IBM" durch "Atari"), schon stimmt es (fast) wieder. Wer AmigaDOS (Commodore) oder ähnlich Exoten vermißt, möge Nachsicht mit mir haben, denn alles kenne ich nun auch nicht.]

Fangen wir mit unseren altbekannten BSen an: NEWDOS und CP/M. Beide wurden für Computer mit Z80- (bzw. 8080-) Prozessoren entworfen, womit ein Haupt-Parameter festgelegt war: 8-Bit-Prozessor mit 16-Bit-Adressen = 64 Kb Speicher (RAM und/oder ROM). Damit hören aber schon die Gemeinsamkeiten auf.

NEWDOS

Beim Entwurf von NEWDOS (bzw. dessen Vorläufer TRSDOS) gab es feste Vorgaben: Das BS sollte nur auf einem bestimmten Computer laufen, dem TRS-80. Dieser ist mit einem 12 Kb großen ROM(-BASIC) ausgestattet, das bei 0000h anfängt. Daran kann das BS schon mal nichts ändern. Um den Platz aber nicht ganz zu verschwenden, sollten natürlich möglichst viele ROM-Routinen benutzt werden. Dann liegen irgendwo mitten im Adreßraum (bei 3xxxh) alle Verbindungen zu den Peripherie-Geräten ("Memory Map"), als da wären: Tastatur, Bildschirm, Disk(-Controller) und Drucker(-Interface). Da kann das BS auch nichts dran ändern, aber es darf sich darauf verlassen, daß die Sachen bei jedem TRS-80 dort an der selben Stelle stehen und immer gleich behandelt werden.

Darüber ist dann endlich RAM angesagt, ab 4000h aufwärts, also 48 Kb. Hier muß sich das BS einnisten, aber nicht nur das BS, sondern auch die Anwender-Programme sollen noch reinpassen! Für die Anwender-Programme wurde der Platz ab 5200h (bis 0FFFFh) vorgesehen, also muß unser NEWDOS mit genau 4,5 Kb auskommen, kein Stück mehr!

Eins war damals (scheinbar) klar: Der Computer würde sich nie mehr ändern. Also bliebe das BS immer unverändert und die Entwickler brauchten sich um "Änderungsfreundlichkeit" keine Gedanken zu machen. Aber sie hatten ein viel größeres Problem: In die 4,5 Kb läßt sich kein ordentliches BS pressen, also griffen sie in die Trickkiste und holten "Overlays" heraus. Das bedeutet schlicht: Ein (kleiner) Teil des BSs bleibt immer im Speicher (hier: SYS0/SYS), alle anderen Teile werden je nach Bedarf (von der Diskette) nachgeladen, und zwar immer an den gleichen Platz (übereinander). Das bedeutet auch, daß zwei Overlays nie gleichzeitig im Speicher stehen können. Es geht eine Menge Zeit (für das Suchen und Laden der Overlays) und eine Menge Platz (auf der Diskette) verloren.

Um von einem Overlay aus ein anderes (bzw. eine Routine daraus) aufrufen zu können, kommt natürlich ein einfacher Unterprogrammaufruf (CALL) nicht in Frage, sondern es ist komplizierter: Das rufende Overlay gibt dem residenten Teil (SYS0) bekannt, welches Overlay es mit welchem "Befehl" (z.B. "formatiere Diskette") aufrufen möchte, SYS0 lädt dann das entsprechende Overlay und gibt die Meldung weiter.

Moderne Betriebssysteme nennen so etwas gerne "virtuelle Speicherverwaltung", d.h. virtuell gibt es ein sehr großes BS, von dem aber immer nur ein kleiner Teil real im Speicher vorhanden ist. Allerdings reicht bei modernen Prozessoren doch wieder ein "CALL" aus, ohne das Overlay-Gedöns. Damit kann dann das BS (oder ein sehr großes Programm) doch ganz im Speicher gehalten werden, wenn der groß genug ist. Ansonsten wird doch wieder von Festplatte (wer arbeitet denn noch mit Disketten?) nachgeladen.

Um Mißverständnissen vorzubeugen: Mit "modern" ist sicherlich nicht der IBM PC (samt Kompatiblen) gemeint (siehe dazu auch weiter unten). Der kennt neben den Overlays noch ganz andere Krücken (Stichworte: "Expanded" und "Extended" Memory), um mit seinen Konstruktionsfehlern leben zu können.

Wie Ihr seht, ist NEWDOS also ein hochmodernes BS, das nur einen kleinen Fehler hat: Es ist total auf den TRS-80 zugeschnitten. In jedem Overlay wird garantiert irgendwas benutzt, was nur auf dem TRS-80 so geht. Das BS ist zwar schön modularisiert (durch die Overlays, die jedes für sich eine andere Aufgabe übernehmen), so daß Änderungen eigentlich leicht möglich sein sollten, aber leider gibt es dazu keine Dokumentation (mal abgesehen von Grossers "DOS-Buch"). Vor allem werden weder der Aufbau der Overlays bekanntgegeben noch die gegenseitigen Aufrufe erklärt. Sprich: Es fehlt ein "Systems Guide", wie es zu jeder CP/M-Version existiert, wo ziemlich genau die Anpassung des BS (also CP/M) an beliebige andere Z80-Rechner beschrieben und außerdem festgelegt wird, was auf jedem Rechner gleich sein muß (Einsprungadressen usw.).

Von der Seite der Anwenderprogramme aus, sieht die Sache so aus: Es gibt keine einheitliche Schnittstelle zwischen Anwenderprogramm und Betriebssystem (sprich: einen bestimmten Punkt, an dem das BS per "CALL" betreten werden kann), sondern in der Anleitung stehen gleich mehrere! Das hat zwei große Nachteile für den, der NEWDOS ändern möchte: a) Diese Adressen sind fest, hier darf also nie was anderes als vorgesehen passieren. b) Wo sollen neue (eigene) Funktionen eingefügt werden?

Natürlich sind auch alle Anwender-Programme so geschrieben: Sie sollten nur auf dem TRS-80 und nur mit diesem BS laufen. Wenn irgendein Bit nicht dem im Original-TRS-80/-NEWDOS entspricht, kann das fatale Folgen haben, denn es werden fröhlich alle (auch die undokumentierten) Eigenschaften des TRS-80 und des NEWDOS ausgenutzt.

Eine Kleinigkeit sollte ich noch erwähnen, auf die mich Hartmut hingewiesen hat: Im Newdos gibt es keine Trennung zwischen BS und Kommando-Interpreter! In allen folgenden Betriebssystemen werden alle Benutzer-Eingaben wie COPY, DIR usw. von einem Programm (genannt: CCP, Kommando-Interpreter, COMMAND.COM, Shell, o.ä.) entgegengenommen und dann in BS-Aufrufe umgewandelt. So kann die Oberfläche (der Kommando-Interpreter) leicht gegen eine andere (ein anderes Programm) ausgetauscht werden, z.B. gegen eine grafische Oberfläche wie GEM oder WINDOWS. Das geht bei Newdos nicht, weil dort eine enge Verzahnung von BS und Kommando-Interpreter besteht und der

Kommando-Interpreter kein eigenständiges Programm ist. Was übrigens das Newdos größer macht, als es eigentlich sein müßte. Dafür kennt der integrierte Kommando-Interpreter aber viel mächtigere Befehle als alle anderen mir bekannten (man siehe nur den COPY-Befehl mit den tausend Parametern). Was aber dem Programmierer nicht viel bringt, wenn er ein Programm schreibt.

Um es noch einmal zusammenzufassen: Der TRS-80 ist eine vollkommen festgelegte Maschine und das NEWDOS ein genau darauf zugeschnittenes BS. Beide werden vom Anwenderprogramm als unveränderlich angenommen.

CP/M

Ganz anders die Ausgangssituation bei CP/M: Hier gibt es nicht einen Computer, auf dem das BS laufen soll. Nein, jeder Computer mit Z80-/8080-CPU sollte CP/M "fahren" können. Allerdings werden auch bei CP/M einige Vorgaben gemacht, die aber nicht von einem existierenden Computer vorgegeben wurden, sondern allein aus den Erfordernissen des BSs entspringen. Das zu lösende Problem hieß schlicht: Wie schaffen wir es, (Anwender-)Programme auf jedem Computer laufen zu lassen, ohne daß die Konfiguration des Rechners bekannt ist? "Konfiguration" soll hier heißen: Ausstattung des Rechners mit Speicher und Peripherie-Geräten wie Floppy-Controllern, Bildschirmen, Tastaturen usw.

Wie wir schon mal gehört haben, soll das BS die Verwaltung und das Ansprechen der Peripherie-Geräte für die Anwenderprogramme übernehmen, und zwar vollkommen. Beim TRS-80/NEWDOS kann das leicht umgangen werden, denn jeder weiß ja, wie die Maschine aussieht und kann selber daran rum"PEEKen" und -"POKEn" (was auch jeder tut). Beim CP/M sollte aber gerade das vermieden werden, denn jedes Programm soll ja auf jeder Maschine laufen. Also wurde genau festgelegt, was auf jeder Maschine gleich aussieht. Diese Dinge darf der Anwendungsprogrammierer benutzen, alles andere ist für ihn tabu! Moderner ausgedrückt: Das BS stellt dem Programm eine "virtuelle Maschine" (also nicht eine real existierende) zur Verfügung, deren Eigenschaften auf jedem Rechner die gleichen sind.

Unsere virtuelle Maschine sieht so aus: Das RAM muß bei 0000h beginnen. Anwenderprogramme werden ab 0100h geladen. Das CP/M steht irgendwo am oberen Ende des Speichers. Es ist nicht festgelegt, wieviel RAM zur Verfügung steht. Das Programm kann über einen Wert erfahren, wieviel RAM es benutzen darf (darüber gehört alles dem BS). Wenn das Programm irgendwas vom Betriebssystem will, muß es einige Register laden und dann "CALL 0005h" ausführen. Damit wird dann irgendwas vom BS erledigt, und zwar auf jedem Rechner.

Natürlich gehören noch einige Sachen mehr zu den Festlegungen, z.B. welche Register-Inhalte beim "CALL 0005h" denn nun was bedeuten und was man zurück bekommt. Das würde jetzt aber zu weit führen. Wichtig ist nur: Es gibt eine genau dokumentierte Schnittstelle zwischen Anwenderprogramm und BS. Das Programm darf sich nur auf die festgelegten Sachen verlassen und auf kein Stück mehr. Wie das BS die von ihm geforderten Tätigkeiten erledigt, ist allein seine Sache, und kann sich von Zeit zu Zeit bzw. von Rechner zu Rechner auch mal ändern. Dem Anwenderprogramm muß es vollkommen egal sein, auf welchem Rechner es läuft (d.h. es darf nichts voraussetzen, z.B. daß ein "PEEK (3880h)" die SHIFT-Taste abfragt).

Für das Anwenderprogramm ist damit die Sache klar. Aber was passiert, wenn wir einen vollkommen jungfräulichen Z80-Rechner gebaut oder gekauft haben und nun gerne unsere CP/M-Programme darauf laufen lassen möchten? Nun, ein CP/M muß her! Danach läuft alles.

Damit nicht für jeden Rechner ein vollkommen neues CP/M geschrieben werden muß, teilten die CP/M-Urväter das Ding in zwei Teile: Ein Teil, das BDOS, läuft auf jedem Rechner, ein anderer Teil, das BIOS, nur auf einem bestimmten. Nun muß für jeden neuen Rechner nur noch ein neues BIOS geschrieben werden. Und wenn sich an einem Rechner etwas ändert, z.B. eine Festplatte eingebaut wird, ändert man (wer immer das auch sein mag) nur noch einen Teil des BIOS. Das ganze Prinzip hatte ich ja schon mal erklärt.

Wichtig ist hier der Unterschied zum NEWDOS: Dort gibt es kein BIOS, das man mal kurz anpaßt, wenn eine Festplatte eingebaut wird. Entweder ändert man das ganze BS und läuft dabei Gefahr, daß einige Programme mit diesen Änderungen nicht zurechtkommen, oder läßt es bleiben. Im CP/M dagegen ist so etwas vollkommen unkritisch, denn es bekommt ja kein Programm mit, daß sich etwas geändert hat.

Aber natürlich hat auch ein CP/M seine Tücken: Beim Erstellen der Schnittstelle mußte ja schon bedacht werden, wie ein Rechner aussehen könnte bzw. was er alles kann. Also wurde die oben erwähnte "virtuelle Maschine" so entworfen, daß ihre Eigenschaften "irgendwie" auf die (damals) real existierende Hardware übersetzt werden konnten.

Diese virtuelle Maschine hatte einen 8080-Prozessor (deswegen werden alle schönen Eigenschaften des Z80 nicht benutzt), bis zu 16 Laufwerke, eine Konsole, einen Lochstreifen-Stanzer und -Leser usw. Wie gesagt: virtuell, also nicht vorhanden. In Wirklichkeit hat unser realer Rechner vielleicht einen HD64180 als Prozessor, nur 2 Laufwerke nebst Festplatte, ein Grafik-Terminal und eine serielle Schnittstelle. Das BS übernimmt es nun für das Anwenderprogramm, diese Umsetzung von der virtuellen auf die reale Maschine (möglichst sinnvoll) vorzunehmen.

Wenn es zu einem virtuellen Gerät keine Entsprechung auf der realen Maschine gibt, es uns das ziemlich egal (dann wird eben das 16. Laufwerk nicht benutzt), aber umgekehrt wird die Sache ärgerlich: Wir haben ein reales Gerät, das nirgends in unsere virtuelle Maschine paßt, z.B. eine Grafikkarte mit 256 Farben. Ist das nun eher ein Laufwerk oder ein Lochkarten-Stanzer? Weder noch, es paßt eben nicht. Das wurde damals nicht bedacht und damit aus.

Halt, so einfach geben wir ja nicht auf! Die virtuelle Maschine muß einfach nur (z.B. um ein virtuelles Gerät) erweitert werden und schon sind wir wieder im Geschäft. Das hat Digital Research mit den verschiedenen CP/M-Versionen auch gemacht. Trotzdem ist unsere Grafikkarte immer noch nicht drin, was nur beweist, wie kurzsichtig einige BS-Bauer (nicht mit Landwirten zu verwechseln!) sind. Aber auch mit einer Erweiterung um eine Grafik-Schnittstelle würde kein älteres Programm sie benutzen können (vorher gab es sie eben nicht) und neuere Programme würden auf den alten BS-Versionen nicht mehr laufen. Das Problem kennen nicht nur CP/Mler, sondern auch die MS-DOSler, die inzwischen bei Version 5 anlangen, während mit OS/2 schon der Nachfolger auf der Matte steht, genauso wie damals MS-DOS in der Blütezeit des CP/M.

Dieses Prinzip der virtuellen Maschine (wie in CP/M) statt der realen Maschine (wie in NEWDOS) findet sich in den meisten modernen Betriebssystemen. Ansonsten dürfte ein Rechner und sein darauf zugeschnittenes BS nie geändert werden und nach dem Veralten des Systems wären alle darauf geschriebenen Programme Müll. Wie wir gesehen haben, können sie ansonsten auch Müll werden, aber vielleicht nicht ganz so schnell.

MS-DOS

(Alle MS-DOS-Freaks überspringen bitte dieses Kapitel. Ich bitte darum, von Briefbomben abzusehen, sondern lieber auf das nächste Clubtreffen zu warten, wo Ihr dann persönlich handgreiflich werden könnt.)

Verlassen wir den Z80-Bereich und stoßen wir weiter vor ins Land der begrenzten Möglichkeiten, "IBM-Country". IBM entwickelte mal in grauer Vorzeit (also vor ca. 10 Jahren) einen Rechner, der ungefähr einem um den Faktor 10 vergrößerten TRS-80 entspricht: Prozessor 8088/8086 (leicht aufgemotzter Nachfolger des 8080), 20-Bit-Adressen, also 1 MB Adreßraum, aber höchstens 640 KB Hauptspeicher (RAM), ROM(-BASIC) eingebaut, alles drin und drumherum genau festgelegt (Memory-Map, Bildschirmspeicher usw.). Allerdings wurde das Ding manchmal schon mit einer Grafikkarte ausgeliefert!

Dafür brauchte IBM natürlich ein Betriebssystem (auch wenn die Kassetten-Schnittstelle wie beim TRS-80 nicht fehlen durfte), wobei seltsamerweise NEWDOS nicht in die engere Wahl kam. Aber CP/M war dabei, nur hatten wir ja keinen 8080/Z80 mehr da drin, also hätte es (leicht) umgeschrieben werden müssen. Irgendwie vergrätzte Digital Research aber IBM und somit bekam Microsoft die Aufgabe, ein BS für den IBM PC zusammenzuschustern. Was sie dann auch genauso taten (eben "zusammenschustern").

Eigentlich handelte es sich dabei dann schließlich nur um ein aufgemotztes CP/M, damit die Anwender-Programmierer aus dem CP/M-Bereich nicht umlernen mußten und die CP/M-Programme schnell angepaßt werden konnten. Nur wurde statt des "CALL 0005h" ein "INT 21h" genommen, was das gleiche ist, nur etwas kürzer. (Beim Z80 gab's das auch schon, hieß nur "RST xxh" und wurde unter CP/M nicht benutzt, obwohl es der 8080 auch konnte (glaube ich). Aber NEWDOS hat es benutzt!)

Ansonsten wurden wieder zwei Teile genommen, BDOS und BIOS, die aber diesmal getrennt auf der Diskette stehen. Teile des BIOS sind auch im ROM (bzw. manchmal auch mehreren ROMs) vorhanden. Außerdem mußten natürlich einige Erweiterungen an der "virtuellen Maschine" vorgenommen werden, damit alle vorhandenen und geplanten Eigenschaften des IBM PC eine Entsprechung hatten.

Bis hierher sieht die Sache noch rund aus. Aber nun kam ein (meiner Meinung) fataler Fehlschritt: Statt das MS-DOS als Grundlage (=virtuelle Maschine) für alle Anwender-Programme zu nehmen, wurde von IBM die Hardware genau fest- und offengelegt, wie beim TRS-80! (Während beim MS-DOS einiges im Dunkeln blieb, wie beim NEWDOS.) Außerdem erwies sich das MS-DOS als nicht besonders schnell, war unflexibel und fehlerhaft. Also dachten sich unsere Anwender-Programmierer: "Egal, ich kenne ja die Hardware, greife ich doch direkt darauf zu!" Und schon war die Grafik schneller und farbiger o.ä.

Aber: Jede anders aufgebaute Hardware eines anderen Herstellers (sei es nun ein ganzer Rechner oder nur eine neue Grafik-Karte) läßt sich von solchen Programmen nicht benutzen, weil sie ja genau auf die IBM-Hardware zugeschnitten sind. Dementsprechend liegen jedem Programm auch tausend Treiber (meist zu Grafikkarten und Druckern) bei, weil das MS-DOS eben zu blöde ist. Und deshalb heißt es "IBM-kompatibel" statt "MS-DOS-kompatibel". Ein kleiner Vorteil: Jedes MS-DOS müßte auf jedem IBM-Kompatiblen sofort laufen, ohne Anpassung des BIOS.

Womit wir wieder bei den TRS-80/NEWDOS-Verhältnissen wären. Jeder Rechner, auf dem MS-DOS-Programme laufen sollen, muß genauso aussehen wie der Original-IBM PC. Rechner, die von dem (Hardware-)"Standard" auch nur ein Bit abweichen, bekommen böse Probleme (woher kennen wir diesen Satz?). Die "virtuelle Maschine" wurde also zugunsten der realen Maschine fallengelassen.

Bleiben wir aber beim MS-DOS: Im Gegensatz zum NEWDOS, und das ist immerhin ein Fortschritt, sind die genauen Adressen des MS-DOS (was steht wo?) nicht mehr festgelegt. Es gibt nur die von CP/M bekannte Schnittstelle per CALL/INT. Leider wurden, wie gesagt, alte CP/M-Hüte übernommen, so daß wirkliche Neuerungen (z.B. Unterverzeichnisse und die File-Handles statt der FCBs) erst beim XT (MS-DOS 2.0) eingeführt wurden (Info von Hartmut). Natürlich muß man kompatibel bleiben und so bleiben alte Sünden lange fortbestehen... Die "Neuerungen" stammen übrigens aus UNIX, womit wir beim nächsten Thema wären.

<u>UNIX</u>

Lassen wir die profane PC-Welt hinter uns und steigen ein ins Großrechner- und Hochschul-Geschäft. Als CP/M gerade anfing, die Mikrocomputer zu erobern, gab es ja auch noch große und sehr große Rechner. Einige davon stehen immer in irgendwelchen Entwicklungslabors und die Leute, die daran arbeiten, sind meistens mit den gelieferten BSen nicht zufrieden. So werden immer neue BSe entwickelt (genauso wie Programmiersprachen), die aber kaum mal die Labors verlassen.

Anders dagegen UNIX: Zuerst war es ein Public-Domain-BS, das besonders in Entwicklungs-Labors und Hochschulen populär wurde, weil es auch einfach und sehr gut dokumentiert war. Erst mit dem wachsenden wirtschaftlichen Erfolg wurde das "Public Domain" wieder zurückgezogen. Trotzdem ist UNIX samt Nachfolgern und Nachmachern (z.B. ULTRIX, XENIX eben alles mit "-IX" am Ende) vor allem noch im Hochschulbereich anzutreffen, denn es ist eine Art Lehrstück für "gute" (hardware-unabhängige, leicht übertragbare, erweiterungsfähige) BSe.

Damit hätte ich auch die Hauptsache schon gesagt: Dieses BS ist nun wirklich von keinerlei Hardware mehr abhängig; selbst der Prozessor ist nicht mehr wichtig! Wie das? Nun, UNIX und alle wichtigen Programme selbst sind nicht in Assembler geschrieben, sondern in C, einer "höheren" Programmiersprache. C wurde parallel zu und mit UNIX entwickelt, um das BS und die Utilities zu schreiben. (Ähnlich wurde CP/M in PL/M entwickelt, das aber nie bekannt wurde.)

Wenn nun ein neuer Rechner mit UNIX samt Programmen ausgestattet werden soll, müssen a) ein C-Compiler, der Code für diesen Prozessor erzeugt, und b) einige hardwarenahe Teile des BS neu geschrieben werden. Wenn dann das BS und seine Programme, die ja alle im C-Quellcode vorliegen sollten, neu compiliert sind, gehört der neue Rechner zur Familie der UNIX-Rechner.

Leider weiß ich auch nicht mehr darüber, denn ein "Systems Guide", wie es Digital Research zu jeder CP/M-Version liefert und in dem alle interessanten Informationen für die Installation auf einem neuen Rechner stehen, habe ich für UNIX noch nicht gesehen. Dem interessierten Leser empfehle ich Tanenbaum's MINIX, ein UNIX-Abkömmling, der wieder Public Domain ist.

Nun fragt sich natürlich: Was hebt UNIX über die Masse von NEWDOS, CP/M und MS-DOS noch hinaus? Nun, da UNIX aus dem Großrechner-Bereich kommt, wurde alles vorbereitet, um parallele Prozesse, viele Benutzer, viele Dateien und viele Peripherie-Geräte zu unterstützen. Die "viele" sind wörtlich zu nehmen - "fast unendlich viele" wäre vielleicht besser.

Haupt-Ansatzpunkt dabei ist das File-System. Während bisher immer noch eine Trennung zwischen Speicher- und Peripherie-Geräten stattfindet, wird hier alles zusammengeworfen und als Files behandelt. D.h., eine Diskette oder Festplatte ist nur ein File, entweder ein sehr großer oder unterteilt in mehrere kleine, wie wir es schon kennen. Aber auch ein Drucker ist nur ein File (auf den wir normalerweise nur schreiben), eine Tastatur ebenfalls (meistens nur gelesen) usw. Da die Anzahl der Files (fast) unbegrenzt groß ist, können auch beliebig viele Peripherie-Geräte angeschlossen werden - jedes ist ja nur ein neuer File. Sie bekommen dann so nette Namen wie "PRT" (Drucker), "FD40DSDD" (Floppy, 40 Track, doppelseitig, doppelte Schreibdichte), "TTY" (Bildschirm) usw. Kennen wir ja schon aus CP/M, nur war das Konzept da etwas anders.

Somit haben wir ein BS, das sich a) leicht auf (fast) jeden Rechner übertragen und b) (fast) beliebig erweitern läßt. Warum ist es dann trotzdem noch nicht so verbreitet? Dafür mag es mehrere Gründe geben, über die ich hier nur spekulieren kann. Wahrscheinlich reiten alle Hersteller von Computern lieber auf ihren eigenen Entwicklungen rum (besonders IBM). Außerdem ist das BS für (heutige) Mikro- und Personalcomputer zu groß (Tanenbaum spricht von einer Minimal-Größe von 128 KB). Parallele Prozesse und beliebige Erweiterbarkeit sind im Hobby- und Büro-Bereich noch nicht gefragt und wurden (von der Hardware-Seite) auch nicht unterstützt. Das ändert sich langsam. Und nicht zu vergessen: UNIX steht und fällt mit C, das zwar immer mehr Anhänger findet, aber auch viele Kritiker hat (ich zähle mich gern dazu).

<u>Schluß</u>

Jetzt habe ich mich einmal quer durch den Garten der Betriebssysteme genörgelt. Besonders IBM/MS-DOS und NEWDOS haben ihr Fett abbekommen, aber das liegt an meiner Sichtweise. Bei UNIX könnte ich auch sicherlich noch einige Nachteile finden, wenn ich nur so ein System näher kennen würde. Ich habe es hier nur der Vollständigkeit halber aufgenommen.

Natürlich dürft Ihr über alle Punkte anderer Meinung sein, insbesondere andere Aspekte in den Vordergrund stellen und somit zu einer anderen Wertung kommen. Aber dazu müßt Ihr schon selbst mal zur Feder greifen (bzw. in die Tasten hauen), um mir "Kontra" zu geben. Ich nutze hier jedenfalls schamlos Eure Schreibfaulheit aus, um meine Meinung als die (scheinbar) allgemeingültige kundzutun. Atschibätsch!

Der Kapitän starrte verblüfft auf die linke Hand seines Copiloten, der den Anflug auf Frankfurt durchführte. Sie bediente die drei Schubhebel der DC 10. Es war böig, und wegen der starken Geschwindigkeitsschwankungen mußte bald Motorleistung nachgeschoben, bald stark zurückgenommen werden. Das automatische Schubhebelsystem (Automatic Throttle System), das normalerweise die Schubhebel reguliert, war 'übungshalber' abgeschaltet worden. Doch seltsam: Der Pilot bediente die Hebel genauso ruckartig und 'überreagiert', als sei die wenig sensible Automatik eingeschaltet. Tatsächlich bedient jeder Pilot die Schubhebel von Hand gefühlvoller: seine Erfahrung sagt ihm zum Beispiel, das ein plötzlicher Böenschlag, der Geschwindigkeitszuwachs bringt, überhaupt nicht berücksichtigt zu werden braucht, weil im nächsten Augenblick die gegenläufige, ausgleichende Tendenz einsetzt. Eine Automatik hingegen, sei sie noch so feinfühlig, muß auf jeden Impuls von außen reagieren.

Hinterher stellte sich heraus, daß der Copilot, unter Streß, instinktiv die Bewegungen des Auto-Throttle-Systems an den Schubhebeln nachgeahmt hatte, obwohl er weitaus sensibler hätte fliegen können.

Dieses Beispiel aus dem Anfang der achtziger Jahre läßt Prokrustes wiederauferstehen: Wenn die Technik die Forderungen nicht erfüllen kann, muß der Mensch sich eben nach ihren Möglichkeiten einrichten. Bei der eben erwähnten 'Bettstreckung' erfolgte die Anpassung sogar, wenn auch unter Streß, freiwillig.

Fälle, bei denen die Technik die Forderungen ihrer Benutzer nicht erfüllen kann, gibt es mehr, als der mit den Erfolgsmeldungen aus dem Elektronikbereich überschüttete naive Laie glaubt. Einer von vielen - ebenfalls aus der Luftfahrt: Für die halbjährlichen Überprüfungen der Zivilpiloten hatten die Verkehrsfluggesellschaften ein schwieriges fliegerisches Manöver eingebaut, das als Canyon Approach bekannt ist: Ein Flugzeug macht einen Anflug auf einen Platz in einem Canyon, der am Ende 'verschlossen' ist. Der Anflug mißlingt, es muß durchgestartet werden. Gleichzeitig mit dem komplizierten Verfahren des Powergebens, Fahrwerk- und Klappeneinfahrens muß nun auch noch eine steile Kehrtwende um 180 Grad eingeleitet werden.

Diese Verfahren war äußerst kompliziert, es schied aber auch die 'Könner' von den 'Nichtkönnern', weil dabei ein äußerst schwieriges Zusammenspiel von Triebwerkbedienung, exakter Ruderbedienung und rascher Entscheidungsfähigkeit gefordert wurde. Kein Wunder, daß dieser Canyon Approach von den ersten Kolbenflugzeugen a la Superconstellation bis zu den Jets die Hohe Schule des Verkehrsfliegens war. Wer damit nicht klarkam, fiel durch den Check.

Doch dann wurden diese Überprüfungen nicht mehr im echten Flugzeug, sondern im Simulator praktiziert. Es stellte sich heraus, daß Simulatoren der damaligen Generation diese komplizierten Flugvorgänge nicht befriedigend simulieren konnten. Schickte man die unbrauchbaren Geräte an ihre Hersteller zurück? Natürlich nicht: Man strich schlicht diese Übung aus dem Überprüfungsprogramm. Plötzlich war nicht mehr wichtig, was über zwanzig Jahre lang so manchem Prüfling Durchfälle beschert hatte - in doppelter Bedeutung.

Wir stehen heute vor der Tatsache, daß unsere elektronischen 'Hilfen' nicht das leisten, was wir von ihnen fordern, sondern daß wir uns mit unseren Forderungen ihren Leistungen angepaßt haben. Ein primitives Beispiel aus dem Haushaltsalltag: Früher konnten wir unsere Bank- und Postgiroüberweisungen handschriftlich ausfüllen, wie uns die Klaue gewachsen war. Hauptsache: überhaupt lesbar. Weil der moderne Computer dazu absolut unfähig ist, haben wir nun, als Mehrarbeit, gefälligst sorgfältig in Großdruckbuchstaben zu schreiben, und, bitte, nur ein Buchstabe pro vorgedrucktem Kästchen. Sonst muß der Herr Lesecomputer den Dienst verweigern. Und niemand ist da, der diesen, als modernsten Fortschritt deklarierten Zivilisationsschrott den Banken und der Bundespost als Plunder an den Kopf wirft, mit der Auflage, endlich einen Computer zu liefern, der das kann, was jeder Aushilfslehrling kann: individuelle Handschrift lesen. (Daß es sehr wohl Computer gibt, die Handschriften lesen, identifizieren und vergleichen können, ist kein Gegenargument, solange sie so teuer und fehleranfällig sind, daß sie sich ein normaler Betrieb nicht leisten kann.)

An diesem allgemein verständlichen Beispiel zeigt sich ein Dilemma, das von Computergeneration zu Computergeneration akuter wird: Der Computer ist unfähig zu einer echten 'Companionship' mit seinem Partner, dem Menschen. Eine 'Zusammenarbeit' kann nur stattfinden, sofern der menschliche Teil des Teams sich den Eigenarten, Unarten und mangelnden Fähigkeiten der Elektronik anpaßt. Diese Art einer Partnerschaft wirft eigentlich alles über den Haufen, was zum Beispiel von den Fluggesellschaften seit mehr als einem Jahrzehnt als Heilige Kuh der Cockpitarbeit dogmatisiert worden ist: Teamwork. Ein Besatzungsmitglied - Captain eingeschlossen - das zu einer fruchtbaren Anpassung und einem Zusammenspiel mit den anderen nicht fähig ist, fällt beim nächsten Check durch. Bezeichnenderweise werden die halbjährlichen Überprüfungen schon lange nicht mehr für eine Einzelperson (Captain, Copilot, Flugingenieur) durchgeführt, sondern gleichzeitig für die gesamte Cockpitcrew. Als Besatzungsmitglied wäre der Computer wegen seiner

mangelnden Anpassungsfähigkeit und seiner Eigengesetzlichkeit eine Katastrophe. Und dabei soll er doch zumindest den Flugingenieur, zum Teil auch noch Copilot und Captain ersetzen! Er spielt nur mit, wenn man bereit ist, auf seine Art einzugehen, wie er 'bedient' werden möchte. (Das Wort ist verräterisch.)

Kein Wunder, daß bei der eben noch hochgepriesenen Bordcomputergeneration die Fehlerlimits bei automatischen Flugvorgängen höher lagen, als wenn der Pilot das gleiche Verfahren von Hand flog. Er mußte zum Beispiel von Hand exakter Geschwindigkeiten einhalten, als wenn er die Automatik dafür einschaltete. Mit ihr durfte schlampiger geflogen werden. Von einer gleichwertigen Partnerschaft kann also nicht die Rede sein. Hingegen ist der Computer dort überlegen, wo er blitzschnell Informationen liefern soll oder/und Abflugrouten, Luftstraßen und so weiter optisch darstellt. Die Dutzende von Kursen und Richtungsänderungen, die früher im Kopf der Piloten nur mühsam zu einem Bild wurden, werden jetzt in Sekundenschnelle auf den Schirm projiziert, mitsamt dem sich bewegenden Flugzeug. So läßt sich auch, ein Beispiel von Dutzenden, beim Abstieg aus der Reiseflughöhe erkennen, an welchem geografischen Punkt das Flugzeug bei einer bestimmten Sinkrate in einer bestimmten Höhe ankommen wird. Eine Information, die früher mühsame Kopfrechenvorgänge benötigte.

Fast könnte man sagen: Der Computer entwickelt so etwas wie ein Schamgefühl, weil er sich der menschlichen Art zu denken, zu erkennen, zu reagieren, nur mangelhaft anpassen kann. Er kompensiert diesen Mangel durch eine wahre Sturzflut von Informationsmöglichkeiten und Automatikfähigkeiten, die so exzessiv nun auch wieder nicht gebraucht werden. Ein Beispiel von vielen:

Schon auf der DC 10 gab es die Möglichkeit, Gewitterfronten automatisch zu umfliegen, indem man bestimmte Kurse in den Navigationscomputer eintippte, so daß nach dem Umfliegen die Maschine sofort wieder auf den alten Kurs zurück gebracht wurde. In der Praxis kenne ich niemanden, der das, außer bei einem Checkflug, jemals gemacht hätte: Das Vorprogrammieren war weitaus mühsamer, als wenn man in alter Weise schlicht am Kursknopf des Autopiloten drehte und nach Sicht und eigenem Gusto die Wolken umkurvte, unter Umständen sogar von Hand.

Nun ist es freilich nicht der Computer, der ein Schamgefühl entwickelt. Vielmehr wird er ja von einem Menschen konstruiert. Dieser Mensch freilich ist keinesfalls der, der ihn dann - bleiben wir im Luftfahrtbereich - benutzt, um mit ihm den Nordatlantik zu überqueren. Im Gegenteil, dieser Ingenieur hat von den Problemen eines gestreßten Flugkapitäns, der auf 30W in eine nicht vorausgesagte Gewitterfront gerät, nicht die geringste Ahnung. Er kann sie daher seinem Gerät auch nicht als 'einprogrammierte Gene' mitgeben. Nach wie vor kann der Computer nur dann vollwertige Arbeit leisten, wenn er vorschriftsmäßig fehlerfrei programmiert wird. Damit aber fällt er im Vergleich zu einem simplen Copiloten, der gerade hundert Stunden Flugerfahrung hat, auf den Status eines Vollidioten zurück. Denn ein solcher Copilot wird eine unter Streß gemachte falsche Anordnung eines anderen Crewmitglieds erkennen und sie nicht ausführen, sondern zumindest darüber diskutieren.

Das völlige Versagen des Computers beim Fehlverhalten seines Betreibers (von der 'Partnerschaft' sollte endgültig nicht mehr geredet werden) ist aber genau der Gefahrenpunkt, der von Fortschritt zu Fortschritt größer wird. Jeder Autofahrer, der in einem computerüberwachten Parkhaus parkt, kennt das Problem: Obwohl pausenlos Fahrer hinausfahren und bestätigen, es sei Platz genug, zeigt der Rechner besetzt an und verweigert die Einfahrt. Der Grund: Er zählt schlicht und einfach die Einfahrenden, ohne zu berücksichtigen, daß viele illegal ihren Wagen auf verbotenen Parkflächen abstellen (und somit echte Parkflächen freihalten). Jeder Hilfsarbeiter, der bis zwanzig zählen kann, wäre hier als Parkwächter dem kostspieligen Computer überlegen.

Und während alle Journalistenwelt, aus den Hochglanzwerbebroschüren zitierend, die vollautomatischen Fähigkeiten einer Elektronikgeneration lobt, die längst und besser nicht nur den Flugingenieur, sondern auch schon fast den Copiloten ersetzt hat, wird einfach unterschlagen, daß nun kein Flugingenieur mehr da ist, der über die Schulter der beiden Piloten mit überprüft, ob die Checklisten richtig gelesen werden und der hinausblickt, um beim Anflug auf die Kollisionsgefahr mit anderen Flugzeugen zu achten.

Die absolute Mißachtung menschlicher Fehlleistungen ist genau die Hauptfehlleistung von Computeranlagen. In Tschernobyl war die Automatik genauso wenig in der Lage, die falschen Absichten der Programmierer am Pult (Simulation bestimmter Notzustände) zu erkennen, wie die Computer im Airbus 320, als der französische Pilot seinen Airbus bei einer Vorführung in den Wald flog. Wollte der Pilot das wirklich? Keinesfalls. Er wurde nur durch seinen Ehrgeiz getrieben, ein wenig riskanter und tiefer als normal zu fliegen. Immerhin war er als Chefpilot ja ausgewählt worden, andere Piloten auf diesem neuartigen Flugzeug, 'das keine Pilotenfehler mehr zuläßt', auszubilden. Wer hatte da wirklich 'menschlich versagt'? Von den Airline-Managern, den Psychologen, den Vorstandsmitgliedern, die diesen Mann zum Chefpiloten gemacht hat-

ten, war nie die Rede. Und die neuartige Elektronik hatte nicht einmal ein bißchen menschliche Eitelkeit mit ihren Fehlleistungen erkennen können. Wir alle müssen wohl weitaus anspruchsvoller gegenüber dem technischen Fortschritt werden.

In Zukunft darf keine neue Computergeneration eine Chance haben, die menschliche Fehlleistungen, die nun mal 'normal' sind, nicht bewältigen kann. Sei es, daß ein Girokontenformularausfüller die Großbuchstaben zu undeutlich schreibt, Falschparker die elektronische Zählung verfälschen, ehrgeizige Piloten risikoreicher als einprogrammiert fliegen. Und wieso eigentlich verlangen wir von unserer Kaffeemaschine nicht, daß sie das im Mitternachtssuff einprogrammierte 15:08 Uhr rechtzeitig als vertippt und als 08:15 Uhr für die erste Tasse Kaffee erkennt? Unsere technischen Wunder der siebenten Computergeneration können es nicht. Wie gesagt, man sollte anspruchsvoller werden.

Die Frage, ob die immer wieder sensationell herausgestellten Leistungen moderner Computersysteme wirklich einen Fortschritt bedeuten, entscheidet sich an der Klippe der Umfelderweiterung. Soll die gegenwärtige Computerentwicklung zu einem echten Fortschritt werden, müßte sie zumindest die schlichten Fehlleistungen aus dem weiteren Umkreis erkennen und ausgleichen. Allerdings ist den fortschrittsgläubigen Ingenieuren noch völlig unbekannt, daß sich gerade durch ihre fortschrittlichste Technik neuartige juristische Probleme ergeben könnten:

Seit Mitte der sechziger Jahre wurden immer wieder Meldungen lanciert, nach denen nun die vollautomatische Landung bei Nullsicht gelungen sei. Keinem der aus den Hochglanzprospekten abschreibenden Reporter fiel jemals auf, daß diese Demonstrationslandung a) stets ohne Passagiere mit einer speziell ausgebildeten Testcrew oder b) bei Wetterlagen durchgeführt wurden, die weitaus besser als Nullsicht waren.

Jetzt, mit einer mehr als zwanzigjährigen Verspätung nach der ersten triumphalen Vollzugsmeldung, sind endlich diese Landungen in den Bereich des Machbaren (mit Passagieren) gerückt. (Freilich mit der Einschränkung, daß bestimmte Querwindstärken nicht überschritten werden dürfen.) Und schon treten völlig andersartige Probleme auf, mit denen die von der Wirklichkeit isolierten Ingenieure nicht gerechnet hatten: Die Piloten sehen sich außerstande, wegen der zu kurzen Reaktionszeiten für das manuelle Eingreifen noch irgendeine Verantwortung zu übernehmen. In Zukunft also werden 'technisch machbare' Projekte scheitern, weil das menschliche Umfeld stets völlig ausgeklammert wurde. So zeigt sich in den reduzierten Leistungsmöglichkeiten der Computer die Unfähigkeit ihrer Hersteller, größere Zusammenhänge im menschlichen Bereich herzustellen.

Versagen ist auch immer dort vorprogrammiert, wo es zum sogenannten Multiple Failure oder sogar Triple Failure kommt: Ein Triebwerk gerät in Brand, gleichzeitig fällt die Feuerwarnanlage aus, und auf einem anderen Triebwerk fällt der Öldruck auf Null. Jetzt geht es um Prioritäten: Welche Störung muß zuerst behoben werden? Diese Rangordnung kann nie durch den Computer bestimmt werden, sondern nur durch den Captain aufgrund der akuten Situation. Sie wird jedesmal anders ausfallen, je nachdem, ob er sich über der einsamen Sahara, über Mitteleuropa mit seinen Hunderten von Flughäfen befindet, ob er den Dienst frisch angetreten hat oder seine zwölfte Dienststunde absolviert, ob er in eine Schlechtwetterzone einfliegen muß oder einen Rückwind von 200 Knoten hat.

Wir alle sind betroffen. Fällt in einem Atomkraftwerk nicht nur ein Kühlsystem, sondern am Kontrollpult durch eine elektrische Störung zusätzlich das Warnsystem und die Darstellung des Störverlaufs aus, dürfte der wachhabende Ingenieur schon dadurch total überfordert sein, zumindest, wenn auch die Abschaltautomatik gestört wird. Er dürfte kaum halbjährlich auf Triple Failures trainiert werden, wie ein Airline-Captain. Gerade aber, wenn das gesamte Abschaltprogramm vollautomatisch ablaufen würde, ließe sich vorstellen, daß es für bestimmte Störfälle gar kein Automatikprogramm gibt, einfach, weil sich die Programmierer einen solchen Fall nicht vorstellen können (oder dürfen).

Ein simpler Fall aus der Luftfahrt von 1959, als Computer noch gar keine Rolle spielten, verdeutlicht das Problem, das heute verstärkt auftritt. Da die damalige viermotorige Superconstellation bei einem Vollaststart nur schwer vom Boden zu kriegen war, hatte sich ein Flugkapitän als Privatverfahren zur Auftriebserhöhung folgendes ausgedacht: Für den Start werden die Klappen auf 'Startstellung' gefahren. Doch bevor sie, bei zunehmender Geschwindigkeit, den Auftrieb vergrößern können, bieten sie beim Anrollen nur Widerstand und Beschleunigungsverzögerung. Also legte er vor dem Anrollen den Klappenhebel zwar auf 'Start', zog jedoch vorher die elektrische Sicherung, die das Ausfahren der Klappen bewirkt. Sie blieben also auf Null. Als dann der Moment der notwendigen Auftriebserhöhung da war, wurde die Sicherung eingedrückt, die Klappen fuhren auf 'Startstellung', die Maschine hüpfte in die Luft. Das Privatverfahren bewährte sich bis zu jenem Tag, als genau beim Eindrücken der Sicherung die Elektrik total versagte. Die Klappen fuhren nicht aus. Ergebnis: Aufschlagbrand in Paris Le Bourget, in dem

alle Insassen umkamen.

Es gehört zur Alltagsphilosophie moderner Elektronikprogramme, daß derartige menschliche Reaktionsweisen weder vorgesehen noch abgedeckt sind. Für die Programmierer sind sie schlechterdings 'unvorstellbar'.

Die Möglichkeit zu einer echten Partnerschaft zwischen Mensch und Computer zu gelangen, wird freilich nicht nur durch die Unfähigkeit der Softwaresysteme selber eingeschränkt, sondern mehr noch durch die überzogenen Anpreisungen, mit denen die zuständige Industrie ihre Produkte auf Ansprüche hochzupushen versucht, die diese nun ganz bestimmt nicht erfüllen können. Die unzählbaren unerwarteten Störfälle in unseren Atomkraftwerken (die angeblich sichersten überhaupt, auch wenn seit Biblis, einem Musterfall für 'menschliches Versagen', etwas weniger laut das Lob der deutschen KKWs hinausposaunt wird), das Auftreten von 'Phantomerscheinungen', die sich die ohnehin recht mangelhafte Phantasie der im klinisch-sterilen Raum arbeitenden Ingenieure nie vorstellen konnten, die Materialermüdungen, die Haarrisse in den Kühlanlagen der Reaktoren, die elektronischen Störungen der Warnsysteme, die sich schon mit einem aufs Kontrollpult gelegten Schlüsselbund erzeugen ließen - all diese 'Abfallprodukte' einer maßlos optimistischen Aberdaslläßtsichdochallesmachen-Philosophie werden am besten illustriert durch ein typisches Beispiel aus dem Waffengeschäft.

Als die Pershing-II-Raketen an den Mann gebracht werden sollten (das gilt ähnlich für alle anderen Raketensysteme auch), fand vor den Militärkunden ein Demonstrationsschießen statt. Weil die Rakete nicht annähernd zielgenau war, verringerte man nicht nur die Reichweite des Demonstrationszieles drastisch, sondern installierte dort auch einen Sender, der die ziemlich ziellos herumirrende Rakete auf einer bestimmten Frequenz heranlocken mußte. Trotzdem lag die Trefferrate von 21 Raketen bei weniger als 20 Prozent. Nach dem gleichen Procedere werden wohl auch unsere Atomkraftwerke verkauft.

Allerdings hat die Atom- und Computerlobby stets einen Verantwortlichen als Sünder griffbereit: den, der am Pult des Atomkraftwerkes, im Airbuscockpit oder im Führerstand des ICE-Triebkopfes 'menschlich versagt' hat.

Der Schluß daraus wird nur ungern oder nie gezogen: Wenn Mensch und Computer zusammen eine Situation nicht meistern können, kann eigentlich der Mensch nicht schuld sein (es sei denn, er ist ein Selbstmörder). Seine Verhaltensschemata sind seit Jahrtausenden bekannt. Hingegen ändern sich die Möglichkeiten technischer Systeme von Jahr zu Jahr. Hat sich auch die neueste fortschrittliche Computergeneration wieder einmal nicht dem Menschen anpassen können, sein Versagen nicht ausgleichen können? Wieso ist sie dann fortschrittlicher als die vorige?

Der Mensch, so der allgemeine Tenor der Fluggesellschaften, wird sich in absehbarer Zeit im Cockpit nicht ersetzen lassen. Er ist einzigartig dadurch, daß er seine Erfahrung, seinen Intellekt, Instinkt, seine Prioritätenentscheidung der akuten Situation anpassen kann. Der Mensch reagiert auf unvorhergesehene Störungen und Systemausfälle auf jeder Fall flexibler als ein Computer, dem die Reaktionen darauf gar nicht einprogrammiert wurden, weil sie im Bewußtsein des Programmierers nicht existent waren.

So könnten sich beide Systeme bestens ergänzen und zu einer echten Partnerschaft kommen: Der Mensch mit seiner Flexibilität und seinem Erfahrungsschatz, der Computer mit seinem Überangebot an Informationen u nd Rechenvorgängen.

Dieser Forderung steht zur Zeit noch der maßlose Anspruch der Technologen gegenüber, durch den Computer einen 'besseren Menschen' schaffen zu wollen, der an seiner Perfektion zur Zeit nur noch durch die Fehlleistungen des niederen Menschen gehindert werde.

- +++ -

Rudolf Braunburg war 20 Jahre lang Flugkapitän und lebt heute als freier Schriftsteller in Waldbröl. Er ist Autor mehrerer Reise- und Sachbücher und beschäftigt sich vor allem mit Fragen des Fliegens und der Flugsicherheit.

Railroad Tycoon

Eine komplexe Wirtschaftssimulation, rund um die Eisenbahn!

Als ich meinen Aufruf an den Jens abgeschickt hatte, kam mir die Idee, einfach mal mit einer MS-DOS Ecke anzufangen und einige meiner Lieblingsprogramme zu beschreiben. Vielleicht fühlt sich der eine oder andere 'Untergrundkämpfer' angeregt, auch über seine Erfahrungen mit Dingen rund um MS-DOS zu berichten.

In irgendeiner Computerzeitung wurde ein Spiel namens Railroad Tycoon begeistert beschrieben. Da ich auch ein Eisenbahnfan bin, habe ich es noch zur Stunde bestellt. Der erste Eindruck verlief allerdings recht enttäuschend! Einen solch ewig langen Vorspann habe ich sogar bei der Shareware selten angetroffen. Auch von der Grafik hatte ich mir mehr versprochen, als ich zu Beginn die Option VGA gewählt hatte. Aber die Handlung hat es in sich und verspricht interessante Beschäftigung über Tage und Wochen.

Grundlage des Spiel ist das Errichten und Betreiben von Eisenbahngesellschaften. Es stehen verschiedene Landschaften und Zeitpunkte für den Beginn zur Auswahl:

Ostküste der USA um 1830
Westküste der USA um 1866
England und Wales um 1828 und
Mitteleuropa um 1900

Das Spiel beginnt damit, daß eine Eisenbahngesellschaft gegründet wird. Es werden 100.000 Aktien im Wert von 5 $ (bzw. 5 £) ausgegeben, sowie ein Darlehen über 500.000 $ aufgenommen. Mit diesem Geld kann innerhalb der angezeigten Landschaft eine beliebige Eisenbahnlinie gebaut werden.

Die Kosten für eine Meile sind je nach Gelände unterschiedlich hoch. Die Bahnhöfe (Depot, Station, Terminal) kosten unterschiedlich viel Geld, haben dafür aber auch unterschiedlich große Einzugsbereiche. Je nach Zeitpunkt und Örtlichkeit stehen verschiedene Lokomotiven zur Verfügung. Um 1830 kleine, wenig leistungsfähige Maschinchen. Man muß bei Streckenlänge und Transportvolumen starke Einschränkungen machen. Mit fortschreitenden Spieldauer kommen größere und leistungsfähigere Lokomotiven auf den Markt. Dann kann man die Zugmaschinen austauschen und längere Züge mit größeren Umläufen zusammenstellen.

Dann muß mit dem Aufstellen eines Fahrplanes begonnen werden. Beim Errichten der Bahnstation wird angezeigt, mit wievielen Wagen an Passagieren, Post und Gütern im Jahr gerechnet werden kann.

Wie im richtigen Leben verläuft die wirtschaftliche Entwicklung in Wellen. Es gibt Zeiten, da boomt das Transportaufkomen. Dann ist es, gerade mit den kleinen Maschinen, fast unmöglich, alle gefülltem Waggons zu befördern. In Zeiten der Rezession stehen die Züge auf den Bahnhöfen herum und warten auf Transportaufgaben.

Es ist natürlich möglich, beliebig viele Züge einzusetzen, um den vielfältigen Transportaufgaben nachzukommen. Jede Lokomotive hat aber, je nach Alter und Auslastung, bestimmte Betriebskosten, die sich am Jahresende auf das Betriebsergebnis auswirken.

Wenn Güter nicht innerhalb eines bestimmten Zeitraumes von einem Bahnhof befördert werden können, gehen sie auf anderen Transportwege über. Es empfiehlt sich deshalb, die Bahnhöfe mit Güterschuppen, Postämtern, Restaurants und Hotels für die Reisenden auszustatten. An Bahnhöfen mit viel Verkehr dauert das Umsetzten der Züge besonders lange, kostbare Zeit geht verloren. Die Anlage eines Rangierbahnhofes mit Drehscheibe ist hier notwendig. Um die Betriebskosten der Lokomotiven niedrig zu halten, können an beliebigen Stationen Werkstätten eingerichtet werden.

All das kostet, besonders in der Startphase, eine Menge Geld. Es kann zu Beginn mit einer Minimalausstattung operiert werden und die Investitionen aus dem laufenden Gewinn erbracht werden. Leider arbeiten aber noch vier konkurrierende Unternehmen in dem gleichen Gebiet! Um eine gewisse Strecke zu belegen ist es vielleicht besser, weitere Darlehen aufzunehmen. Der Zinsaufwand richtet sich dabei nach der augenblicklichen Wirtschaftslage und schmälert natürlich das Betriebsergebnis.

Jeweils nach zwei Jahren wird Bilanz gezogen. Alle Erträge und Aufwendungen werden gegenüber gestellt und der Gewinn (positiv wie negativ) ermittelt. War der Gewinn positiv, sind die Aktionäre zufrieden. Ist er negativ freut es sie weniger. Erwirtschaftet der Manager über fünf Jahre hinaus Verluste, wird er gefeuert und das Spiel ist beendet!

Wenn die Eisenbahnlinie dann gut läuft und sich das Geld häuft, kann man zu allerhandlei Schandtaten aufbrechen. Ist eine konkurrierende Eisenbahnlinie in der Nähe, kann ich mich an einen ihrer Bahnhöfe anschließen. Wer in den nächsten zwei Bilanzperioden mehr Güter abfährt als die Konkurrenz, hat gewonnen und wir alleiniger Betreiber. Es ist aber auch möglich, eine Eisenbahngesellschaft über den Aktienmarkt zu übernehmen. Dazu muß man mehr als 50% der entsprechenden Aktien aufkaufen. Wird gekauft, steigen natürlich die Preise. Also sollte man in Zeiten der Rezession kaufen, wenn die Kurse fallen. Natürlich ist es auch möglich, die Aktien des eigenen Unternehmens zu kaufen. Sind mindesten 50% im Bestand, kann die Gesellschaft nicht übernommen werden, noch können die Aktionäre bei Verlusten den Manager feuern!

Neben der Planung und Anpassung des Fahrplanes und den Geldgeschäften kann sich, wer will, auch als Fahrdienstleiter beschäftigen. Die Strecken können mit Signalbrükken in Blocks zerlegt werden. Die 'eingebauten' Fahrdienstleiter sehen alles ein wenig konservativ und öffnen das Blocksignal erst, wenn der Block frei ist. Hier kann man eingreifen, und die Signale früher öffnen, oder auch schließen um Züge zu stoppen.

Nachdem ich ganze Nächte damit verbracht habe, dutzende von Eisenbahngesellschaften zu errichten, ist mir das Spiel immer noch nicht langweilig geworden. Leider habe ich eine englische Version erwischt. Das Handbuch ist aber leicht und flüssig geschrieben und bereitet wenig Schwierigkeiten.

Allerdings gibt es auch einige Mängel. So gibt es spaßige Einlagen, wie den Brückenbau. Die ersten zehn Male ist das ja ganz lustig, dann fängt es aber an zu nerven. Auch ist die Auswahl der Lokomotiven nicht ganz ideal. Man sollte sich nicht daran stören, daß z.B. die rhätische Ge 6/6 (Meterspur!!!) oder englische Typen durch ganz Europa sausen. Auch ist die Verteilung der Ressourcen ab und zu ein wenig verwunderlich, oder hat man z.B. schon von Weinbergen bei Königsberg gehört?

Aber alles in allem: Railroad Tycoon ist ein spannendes Strategiespiel (es muß nicht immer Kriegsspiel sein) das langen Spielspaß verspricht.

Zilogs späte Ernte

Der amerikanische Halbleiterhersteller Zilog, Entwickler des in Rekord-Stückzahlen verkauften Mikroprozessors Z80, hat sich wieder an sein "verlorenes Kind" erinnert. Während der Zeit, als Zilog zum Ölmulti Exxon gehörte, meinte man mit Neuentwicklungen das große Geschäft machen zu können. Doch diese Entwicklungen konnten in keinem Fall an den Erfolg des Z80 anknüpfen. Zum Teil waren sie sogar Flops am Markt. Als der erhoffte Umsatz ausblieb, wurde die Halbleiter-Tochter für Exxon zu kostspielig. Der Unternehmensteil wurde verkauft, zum großen Teil an die eigenen (Zilog-) Mitarbeiter.

Seitdem entfalten die Californier eine rege Aktivität in Entwicklung und Marketing. Der serielle Communications-Controller Z8530 ist einer der weit verbreiteten Schaltkreise in modernen Personal Computern. Aber auch auf die Qualitäten und Markterfolge des einstigen Zugpferdes Z80 besann man sich. Inzwischen erkannte man bei Zilog, daß man mit hochintegrierten Z80-Controllern gute Geschäfte machen kann, denn das Know-How und die Programmiererfahrung für diesen Mikroprozessor ist weit verbreitet. Lest dazu einen Auszug aus der Zeitschrift "Elektronik-Informationen" Heft 10/90.

Heinrich Betz

1989 war für Zilog aber ein weiteres Erfolgsjahr. Es wurden mehr neue Bausteine vorgestellt als in jedem anderen Jahr. Welche Schwerpunkte nun Zilog setzt und welche weiteren Bausteine zu erwarten sind, erläutert **Dipl.-Ing. Eckhard Stock,** Managing Director der deutschen Niederlassung.

ELEKTRONIK INFORMATIONEN: Herr Stock, welche Auswirkungen hatte die Übernahme auf die deutsche Niederlassung, und haben alle Mitarbeiter Anteile übernommen?

E. Stock: Weltweit wurde ca. 500 Mitarbeitern angeboten, sich an der Übernahme von Zilog zu beteiligen. Dies gilt auch für alle Mitarbeiter der deutschen Niederlassung. Dieses Angebot wurde von allen angenommen, so dass es bis heute ca. 500 Zilog-Angestellte mit Anteilen an der Firma gibt. Sie können sich sicher vorstellen, dass die Motivation, für eine Firma zu arbeiten, von der einem ein Teil gehört, grösser ist, zumal man sich vom Gang an die Börse einen finanziellen Vorteil verspricht.

ELEKTRONIK INFORMATIONEN: Spielt heute der Z80 für Zilog noch eine Rolle?

E. Stock: Der Z80 spielt für Zilog nach wie vor eine grosse Rolle. Wenn man sich heute den Markt für Mikroprozessoren ansieht, kann man feststellen, dass die 8-Bit-Prozessoren nach wie vor stückzahlenmässig eine dominierende Rolle spielen. Innerhalb der 8-Bit-Prozessoren basieren mehr als die Hälfte auf dem Z-80-Befehlssatz. Der Z-80-Befehlssatz, den man sicher als Industriestandard bezeichnen kann und für den eine grosse Menge an Software existiert, ist heute noch Basis für viele Neuentwicklungen unserer Kunden. Dem Wunsch der Entwickler nach ständiger Leistungssteigerung der Systeme und höherer Integrationsdichte hat Zilog schon seit einigen Jahren entsprochen, indem man die Z-80-Produkte in CMOS weiterentwickelte, die Taktraten kontinuierlich erhöhte (den Z80 gibt es heute als 20-MHz-Variante), Z-80-Bausteine zu höher integrierten Bausteinen entwickelte und verschiedene Gehäuseformen wie PLCC oder Quad-Flat-Pack für höher integrierte Bausteine herausbrachte.

Ein wenig Nachdenkliches, Wissenswertes und Bekanntes für CP/M-Freunde. Leider in Englisch, das Übersetzen müßt ihr selbst besorgen. Den Artikel bekam ich über INFOCPM§FINHUTC.BITNET, der "Weiterreicher" David Goodenough ist Autor vieler PD-Programme, darunter ZSM, UUCP und QTERM. Viel Spaß beim Entziffern, Rüdiger.

A04980 (401 lines) CPM.PuDoSe 11/04/90 0700.8 fwt Sun cpm
To: äforum >site>t>digests>cpmü

INFO-CPM Digest   Thu, 1 Nov 90   Volume 90 : Issue 169

Today's Topics:
Zen and the art of CP/M

Date: 1 Nov 90 05:30:59 GMT
From:
usc!zaphod.mps.ohio-state.edu!think.com!mintaka!spdcc!mirror!pallio!dg§ucsd.edu (David Goodenough)
Subject: Zen and the art of CP/M
Message-ID: <XX00011f10§pallio.UUCP>

The following article appeared in the PIPMAG online magazine on GEnie. I found it interesting because of the history it contains, but also because it explains why I spend my time beating my head against a 14" green phosphor tube, covered with the Z80 assembler neumonics that make up QTERM, when I could do it in C on the 386 UNIX crate at work, and get it written much quicker.

It is reproduced with permission of the author.

```
--
    dg§pallio.UUCP - David Goodenough          +---+
                        IHS     ö +-+-+
    ..... !harvard!xait!pallio!dg              +-+-+ ö
AKA:    dg%pallio.uucp§xait.xerox.com                +---+
```

Zen and the Art of CP/M

by

Robert Coleman (R.COLEMAN3)

It was late. Maybe midnight, and quiet in the house. Upstairs, my wife and son were probably asleep. I could hear the faint sound of the TV droning; that and the soft breeze outside my opened window. Whoosh! Drive A on my computer was busy at work. I was backing up some files and doing general housekeeping after a long session at the keyboard. Tic-tic, whoosh, tic-tic! Drive B kicked in. I watched. I had to smile. It was noisier than drive A -- always was (I could close my eyes and tell which of my drives were running, just by the sounds they made). Drive B was perhaps the better of the two. It was newer. But they were both slow; just as slow and tired as I was now. I looked up at the window and felt the cool breeze on my skin. No, they weren't the fastest drives. Not like the 1.2 Meg on my 386 at work. That was much faster; faster by a country mile. And I pushed it too -- pushed the hell out of it. But that was work. Now, just being here late at night with this old CP/M computer, it was somehow different...

Zen and the Art of CP/M? (...like Motorcycle Maintenance? A10) Crazy? Hey, I'm not the first to get this feeling of religious fervor over a computer operating system. How about 'The SOUL of CP/M' and 'The CP/M BIBLE' A2,30. Mr. Waite and associates, they too had reverence for this IDEA. In spite of MS-DOS, countless thousands still cling to CP/M. Why? Good question. Perhaps it has more to do with the intangibles -- faith, acceptance, quiet devotion, and a brand of grass-roots, seat-of-the-pants wisdom. The CP/Mer assumes a tongue-in-cheek posture; all the while, watching the rest of the computer world spin its wheels in a deluge of glitter, gloss and Madison Avenue hype. It appears that regardless of the odds, in an era of disposable widgets and planned obsolescence, in the face of such staunch and resolute adversity, CP/M is here to stay. In the words of one PIPMAG author, Jim Taylor, CP/M is 'the peoples operating system'.

It is interesting that in spite of CP/M's continued popularity, the name 'CP/M' has been interpreted to mean several things: The CP/M Bible refers to it as 'control program for microprocessor'. In one Digital Research, Inc. CP/M Users Manual dated 1979, it is referred to as 'control program for microcomputer'. Still, I have heard it referred to as 'control program monitor'. These discrepancies of course only add to the mystique. I guess any CP/Mer worth his salt has read the basics, but just where DID CP/M come from? Where does it fit into the computer world as we enter the 1990s?

Perhaps the largest MAINFRAME computer, and the grand-daddy of them all, was the UNIVAC computer of 1950. Built entirely of vacuum tubes, it was so big it filled a large room. And the PDP-8 MINICOMPUTER, costing a mere $50,000, brought computing to corporate business in the mid-60s. But it was the advent of the microprocessor that made the MICROCOMPUTER a reality.

In 1969, engineers at Datapoint Corporation of San Antonio, Texas, designed a simple central processing unit and contracted both Intel and Texas Instruments to implement the design on a single piece of silicon. Intel succeeded, but the product ran ten times slower than had been agreed upon, so Datapoint backed out of the deal. Intel decided to go ahead and market the device and called it the 8008 microcomputer A40. Little did they know that this humble device signaled the beginning of the microcomputer revolution.

The 8008 was truly limited, and was used as a controller such as in robot stepper motors and push button TVs. The 8080, more-or-less its direct descendent, was the first microprocessor with an instruction set powerful enough to do real data processing. But these were still humble beginnings. In 1974, a microcomputer kit was sold to hobbyists called the Altair 8800. It featured switches to enter data, and LEDs as a monitor device. There was no keyboard, no CRT, and no floppy drives, which meant no stored programs. All programs had to be tediously entered by hand, byte by byte. This was indeed the stone age of microcomputing. They say that necessity is the mother of invention -- thus the birth of an operating system. CP/M was originally developed by Gary Kildall in 1973 for testing 8" Shugart disk drives. Later, in 1976, he founded Digital Research, Inc. and released CP/M, targeted at 8080 microcomputer hobbyists A50. But the 8080 was slow, and needed three different voltage levels and several support ICs to operate. It was soon replaced by the 8085 which required only +5 volts to run. Then, later came the Z80. It was not

only 8080/8085 software compatible, but included many additional, more powerful instructions. By 1980, with the Z80, CP/M had truly become the industry standard.

This is where Big Blue enters the picture. Microcomputers by this time were starting to find use in the business world. IBM realized the vast commercial potential and was gearing up for a major onslaught. They had been associated with quality mainframe computers, and most experts agreed that they would become the leader in the microcomputer field -- in essence, they would decide the future of microcomputing. IBM chose the 8088 microprocessor by Intel, and with Microsoft, they developed a new operating system called PC-DOS (essentially the same as MS-DOS -- Microsoft's stand alone version). Today, PC doesn't just mean personal computer -- it means IBM compatible.

For the uninitiated, let's take a look inside these machines and see what all the fuss is about. The 8088/8086 microprocessor, which is used in the PC-XT and its clones, is vaguely similar to our 8080 (the NEC V20 version can run 8080 instructions). It has an 8-bit data bus but 16-bit internal architecture and is considered a 16-bit microprocessor. The 8086 is an upgrade with a full 16-bit data bus. The major feature of the 8088/8086 is the ability to address up to 1M of RAM using 'segment address-extension' registers. This method is similar to CP/M 3.0's method of using multiple banks, only in the 8088/8086, this technique is used to the fullest. Much like the 8080, the 8086 can only use 64K at a time, but it has 10 banks of 64K each. This makes a total of 640K of RAM. By using other tricks it can access many megabytes of additional memory.

Another member of the 8086 family is the 80286. It provides 12 Mhz operation, a crude form of multi-tasking (running several SMALL programs at once) and is used in the PC-AT and its clones. The 80386 is another upgrade which provides up to 33 Mhz operation, a full 32-bit bus and addressing up to 4 Gigabytes of memory. And the 80486 is yet the latest upgrade, which provides further streamlining with a built-in cache memory controller and a coprocessor to speed up floating point calculations. These microprocessors are currently considered the 'leading edge technology' and are the darlings of the micro world Ä60. But even now, there is talk of MS-DOS being replaced in the 1990s with UNIX, a true multi-user/multi-tasking operating system.

There is another class of super microprocessors, such as the Motorola 88000. These new devices use RISC technology (RISC -- 'Reduced Instruction Set Computer'). They provide even faster execution by implementing a more efficient set of instructions and executing these within one machine cycle. Whew!

Okay, so what does this mean for CP/M? The microcomputer, which started out as an experiment and a boon for hobbyists has now become the workhorse of the corporate business establishment. The 80286-386-486 explosion is being fueled by big business, where money is no object and there is a constant demand for ever more speed and efficiency. Through CAD (Computer Aided Design), these powerful microcomputers are now being used by scientists and engineers. By connecting them together with LANs (Local Area Networks), some experts predict microcomputers will be direct competition for the minicomputer market which has already taken a beating in recent times. But while these advances are impressive, it leaves one piece of the puzzle missing. What about the consumer? It is true, most people can afford

a simple XT. What does that mean really? Today, most GOOD software written for MS-DOS is expensive. And to enjoy the fruits of the leading edge, you have to constantly upgrade your equipment. The latest software offerings for PCs aren't designed to run on two floppies -- a hard drive is MANDATORY. A VGA monitor alone can cost more than an entire monochrome XT package. Getting caught up in this spiraling staircase of madness is frightening and can lead to gross over-spending and perhaps even what I like to call 'PC-phobia' -- the fear of owning an inferior computer. In reality, the PC revolution could be thought of as a plush banquet, where Big Business is picking up the tab, and the gamers are picking up the crumbs, hanging on to Sugar Daddy's coat tails.

The dream among countless, now defunct computer manufacturers in the early 80s of fully automated lifestyles with a computer in every home was just that, a dream, and one that I fear will not soon be realized. Let's face it, using a computer requires some effort, some learning on the part of the user. You can't just turn it on like a VCR or a Nintendo. Even the 'point and click' fad is too strenuous for most of the general public. I venture to say that there is NO well defined home computer market. All the non-IBM/clone platforms are beginning to fall away; ie: Commodore C-series, Atari, even some Apple II-series. The only true home computer market is really the COMPUTER HOBBYIST, and CP/M is still the ideal operating system for such a phenomenon.

It is conceivable that a CP/M supercomputer could be designed to have many of the bells and whistles of MS-DOS. The HD64180 is an integrated Z80 microcontroller with several built-in features: 10 Mhz operation, two DMA controllers, two serial ports, and a programmable timer. But the most notable feature is the ability to address up to 1 Meg of memory using a built-in MMU (Memory Management Unit -- similar to the 'segment address-extension registers' in the 8086 family, previously mentioned). A SCSI bus (Small Computer System Interface) could be used to interface up to eight devices; ie: hard drives, floppy drives, tape drives, printers, etc. This system could have high-resolution bit-mapped graphics, and perhaps even some form of multi-tasking (MP/M-II). The only problem is that CP/M as we know it has no provisions for handling graphics, let alone 1 Meg of memory! Alas, I have found myself fantasizing at times about such a machine. I have personally designed several 68000 microprocessor-driven systems, and it would not be exceptionally difficult to implement the 64180. Of course, outside of a few crazies like myself, such a machine would have no commercial value in light of the current technologies. It would merely be one CP/Mers dream of power and glory. (NOTE: anyone out there who has similar ideas of building a CP/M supercomputer or who knows of such a beast, please drop me a line via GEnie mail. ÄNote - I posted asking about that large multi-user CP/M system in the Netherlands a few days ago, I'd like to pass that information on to Rob Coleman -- dgÜ

Anyway, what makes CP/M so special? Well, to begin with, it was the first successful disk operating system. Living life as a microcomputerist before CP/M was akin to living without indoor plumbing. And to think that a boni fide computer system like the UNIVAC, that once filled an entire room, could now sit on a table top, I find astonishing. And further, the BDOS/BIOS concept is ingenious. The compatablitily of CP/M operated systems allows the user to share the experience with dissimilar hardware platforms -- thus a whole computing community is enriched. And even more, anyone who lives in the MS-DOS world can feel comfortable with CP/M. After all, MS-DOS

really grew out of CP/M; many of the commands and file handling schemes are similar. But more than anything else, maybe SIMPLICITY is what makes CP/M so appealing. As in Art, there is a beauty in simplicity and symmetry, combined with functionality. And CP/M is just that, a simple, straight-forward solution to operating a personal microcomputer system.

I guess personal computing is different for others. There's Dan-the-man, my snippidy next door neighbor, the one with the super AT with VGA. I have been to his house to marvel over his computer before, and it IS a remarkable machine. It would be ridiculous to ignore the fact that this is no doubt a far superior machine to anything in the CP/M world. The speed is overwhelming, the graphics dazzling. It is awe inspiring, the shear, unbridled power! (Excuse me, I get excited about computers -- any flavor.) Dan-the-man is indeed one happy camper, as well he should be. But I am baffled. What I find unsettling is the way he PERCEIVES personal computing.

Case in point: No sooner did he get this amazing machine, then he was planning to buy another, more powerful one. Why? I'm not sure. Is it the 'first on the block' syndrome, or is it that he really needs that kind of computing power? Dan-the-man plays many computer games and experiments with various software packages. Now six months later, he's ready for a new machine. He hasn't even learned how to use THIS one yet. Speaking of falling in love with one's computer; Dan-the-man not only has no loyalty, he is a technological flirt! In this relationship, there seems to be something missing. Hmmm...

Anyone who would care to notice, would be astounded at the longevity of the Commodore 64 computer. True, this little machine is edging its way to obsolescence, but what a journey it has seen! Nearly eight years after its debut, there is still a large group of devotees. There are, as a rough guesstimate, about seven or eight million of these machines around the world. Not all gamers either I remind you; there are multitudes of application programs that have been written for it. With its handful of ROM-based kernel calls and its graphics capabilities, the C64 is a veritable programmer/hacker's delight. How many imaginations have been stirred by these 8-bit wonders? How many computer software careers do you suppose have been launched by these little 'toy' computers? I have read articles in some trade magazines by professional programmers who STILL enjoy hacking on them. These people really love the little computer. In this relationship, there seems a great deal of dedication. Obviously, personal computing means different things to different people.

Meanwhile, back in my den, whoosh-tic! Both drives came to a halt. I read the directory of the backed-up floppy. Everything looked good. I wondered, should I check for bad sectors? But then again, the format command should have detected any of these. I yawned. Now I yearned for sleep. Inadvertently, I ran scan.com to look for a bad sector, only I entered the A drive by mistake. I cringed as I realized that my boot disk was now being scanned for bad sectors. This is NOT what I had intended. So much for working when I was tired. But behold! There was a bad sector detected on my boot floppy! I glared at the message on the monitor with disbelief. I thought to myself, maybe the A drive was overheating. That had to be it. I ran scan.com again with the boot disk in the B drive. Again -- a bad sector -- the same one. I decided to fix the problem NOW. No matter how tired I was, I couldn't go to sleep having my boot disk corrupted. No way. That would be sacrilege.

I pulled out my back-up boot disk (yes, I had a back-up, but that is a different story!). I pip'd the contents to my RAM drive and then reformatted the corrupted boot disk. Outside my window, I heard a bird call, close in, near the house. I looked briefly to the open window, then back to the task at hand. Once again, I ran scan.com. Good! No bad sectors. Patiently, I waited as the RAM disk contents were copied back onto the boot floppy.

'Booting' -- what an odd term. In the early 1970s, practically all memories were magnetic core types. These memories had tiny toroid-shaped magnets spaced between a lattice-work of wires. The direction of the current through one wire determined the logic value, and current through the other determined whether that value was read or written into the magnet-element. These devices were non-volatile, and held their contents even when the power was removed. But when this data was corrupted, a small program had to be entered to load the system code. This was a tedious process in those days -- done either with switches or paper tape input. The solution? A small loader program was entered manually which loaded another program which actually loaded the system code. This small loader program was called a 'bootstrap' loader. The system was loaded, or pulled up by its own bootstraps. Hence the term -- 'boot' A70.

Whoosh-tic! The A drive came to a stop. I then rebooted. It was done. All was fine in the world. Again, there was the bird outside my window. I got up and walked over to have a look. Again, I heard the cooing. It came from the bushes near the tree out front. It was really too dark to see. Up in the sky, I could see the crescent moon hanging lazily in the heavens, amidst a slight veil of wispy cloud. I listened and heard silence. In the dim light, I glanced back at my computer; my reconditioned boot disk. Yes... all WAS fine in the world. I was ready for sleep.

It has been said that perhaps the true delight is not in arriving at the destination, but in the journey itself; to occasionally stop and enjoy the view. If even just to simply smell the fragrance of flowers, or ponder over a private thought. I, myself, find the personal computing experience to be just that -- PERSONAL. It is a quiet time alone, away from the stress and strain of everyday life. And it is during these times that I learn about strength and weakness, success and failure; the value of work; and play for the sake of play; and ways of acceptance. Perhaps it is in this process of DOING that I find the real reward. Knowing the joy of getting a five page, hand typed program to finally run. Learning to accept that slow, noisy drive. Learning that there is always another corner to turn, another page to read. Learning that maybe there is wisdom in discovering even just one little, simple thing each day. If only for this, it has been worth the while. Oh yes, it could be model cars, or stamp collecting, or flower arranging. But it is when these things cease to be mere objects and become vehicles, taking us someplace else, to some secret, private place, they bring us closer to the inner self. By pouring over the tired guts of my old CP/M computer, cursing myself for having misplaced my screw driver AGAIN, perhaps I learn a little more about myself.

References / Odds and Ends

1. Zen and the Art of Motorcycle Maintenance, Robert M. Pirsig, Bantam. This book is not about fixing motorcycles, or even CP/M, but I strongly recommend it.

2. The Soul of CP/M, The Waite group, Howard Sams. Out of print, but a real gem for programmers. I have read excerpts from some old copies of Computers and Electronics magazine, but I don't own a copy myself. I'm still looking but to no avail. (If anyone knows of an available copy please let me know.)

3. The CP/M Bible, Waite and Angermeyer, Howard Sams. Out of print. This book is useful for someone running several versions of CP/M. It includes a handy cross reference section for 1.X, 2.X, and 3.X systems. I found a NEW hard-bound copy in a used book shop for $4. Note: Many CP/M computer books are now out of print but can be found at 'fire-sale' prices in used book stores. With little money, you can build an impressive CP/M library.

4. An Introduction to Microcomputers, Vol. 1 Basic Concepts, Adam Osborne, Osborne/Mcgraw Hill. A good book for newcomers or anyone who wants to know generally how a microcomputer works.

5. TC128 Compendium #1, Issue #2, CP/M Update, by Todd Madson. Any one who uses a C128 should subscribe to Twin Cities 128 magazine. It deals ONLY with the Commodore 128. But I must say, I wish there was more CP/M coverage.

6. EDN magazine, #24, Nov. 23, 1989. The Annual uP/uC directory. This magazine is written for professional electronic design engineers and is a good source of info on new technology.

7. CP/M Assembly Language Programming, Ken Barbier, Prentice Hall. As far as I know, this book is out of print. However, it is an excellent tutorial on CP/M system programming with the 8080 instruction set.

----- About the Author -----

Robert Coleman is an electronics engineer for Data General, Telecommunication Products Division. Besides designing microprocessor-based electronic circuits, and fiddling around with CP/M in his spare time, he enjoys reading, writing and playing bass in a part-time rock 'n' roll band.

Ausgabe

a) Kosten, die bei weitem einen Jahresverdienst überschreiten und die zur Anschaffung einer Computerausrüstung notwendig sind.
b) Die – mehr oder weniger – freiwillige Herausgabe von Daten auf einem Bildschirm oder Drucker, mit denen ein Computer vergeblich versucht, die unter a) gemachten Ausgaben halbwegs zu rechtfertigen.

Auflösung

a) Zustand, in dem sich ein User befindet, nachdem er festgestellt hat, daß es möglich ist, mit einem kleinen Befehl eine große Festplatte zu löschen. Der Auflösung des Users folgt – falls vorhanden – ein handfester Streit mit der Ehefrau oder – falls vorhanden – ein heiteres Haareausraufen.
b) Je größer die Anzahl und je kleiner die Punkte sind, aus denen eine Bildschirmgrafik zusammengesetzt ist, desto besser ist die Auflösung und damit der optische Eindruck. Wann werden dies die Hersteller begreifen?

Die Public Domain - Ecke

FATCAT

Die Leute mit Festplattenlaufwerken werden immer zahlreicher, und eine Platte mit 20 MByte ist heute nichts besonderes mehr, auch nicht unter CP/M. Hinzu kommt die Fülle von Software, die auf hunderten von Disketten ihr magnetisches Dasein fristet. Einen Überblick über diese Sammlungen zu bekommen und zu pflegen, ist die Aufgabe eines Disk-Katalog-Programmes. FATCAT ist so ein Programm, und es ist, wie immer in dieser Ecke, ein PD-Programm.
In der neuesten Version 2.4 ist FATCAT sehr schnell und intelligent, Nach dem Aufsetzen verschiedener Parameter muß man nur noch die Disketten einschieben, die in den Katalog aufgenommen werden sollen. Erst wenn alle Disketten gelesen sind, beginnt der zeitaufwendige Teil des Sortierens. Das ist sehr anwenderfreundlich. Andererseits verlangt FATCAT auf jeder Diskette eine Datei mit einem Namen, der einer speziellen Syntax genügen muß. Diese Dateien müssen vor oder während des Katalogisiervorganges angelegt werden. Das kann etwas bremsen.
FATCAT ist in der Lage, .LBR Dateien zu lesen und die einzelnen Mitglieder der Libraries in den Katalog aufzunehmen. Ein sehr hübsches Feature!

Die eigentliche Stärke von FATCAT liegt im Wiederfinden von Dateien. Wie war das doch gleich... ich hatte doch mal ein DBASE-Programm... aber wo?

FATCAT erlaubt es, Bereiche anzugeben, in denen gesucht werden soll, also z.B. die Disketten 100 bis 199. Und, FATCAT-Suchbegriffe sind rudimentäre reguläre Ausdrücke, es sind also Wildcards erlaubt. Die Suche nach der obigen Datei gestaltet sich also, was den Suchbegriff angeht, vielleicht so : "*.CMD".

FATCAT ist in Turbo-Pascal geschrieben und kommt ohne Quellen, muß für das System konfiguriert werden. Das ist erst nach Studium der DOCs möglich. Die Bedienung ist sehr einfach, wenn man nur die Grundfunktionen benötigt, für spezielle Sachen kann es etwas aufwendig werden. FATCAT Version 2.4 gibt es beim CP/M Bibliothekar, es benötigt CP/M ab Version 2.2.

BUSH

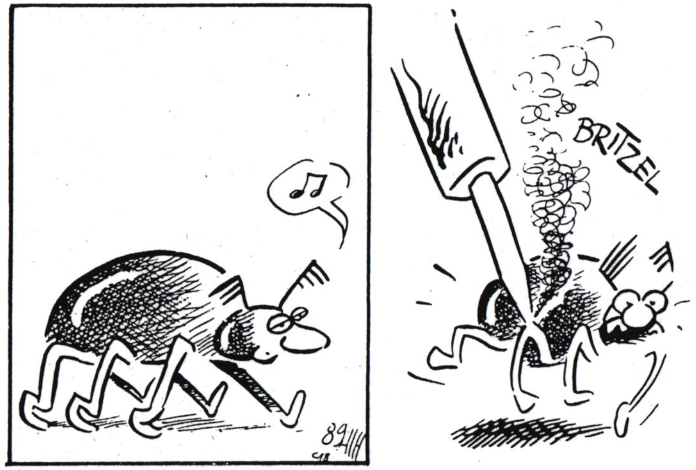
BUSH ist eine Z-Utility, benötigt also ZCPR3x. BUSH ist die Abkürzung für BackUpSHell, also eine Oberfläche, mit der Backups von Disketten oder Festplatten schnell und komfortabel gemacht werden können. Auch hier reduziert sich die Aufgabe des Bedieners weitgehend auf das Einlegen der Disketten.
BUSH muß konfiguriert werden, speziell die Druckersteuerung, denn BUSH kann ausgefeilte Reports erstellen, damit man nach dem Backup auch weiß, was wo steht.
Interessant ist folgendes: man kann BUSH mitteilen, das gewisse Filetypen, wie .BAK, .SYM usw. nicht berücksichtigt werden sollen, und man kann Files in ARChive zusammenfassen lassen.
BUSH arbeitet recht flott, wenn man bedenkt, das nur DOS-Calls verwendet werden, um ein Backup zu erstellen. Der Vorteil liegt darin in der leichten Zugreifbarkeit auf Backups, da sie als Files abgelegt werden, kann man sie ohne BUSH wieder retrieven. Das muß man allerdings auch manchmal. Mir ist es bisher nicht gelungen, ein komplettes Festplattenbackup einer Partition wieder von Disketten auf die Platte zu schaffen. Laut DOCs ist es möglich, alle USER abzuklappern, aber es funktioniert nicht. Bleibt nur die CP/M-Methode mit PIP oder sonstwas.
Aber ein Backup ist ja in erster Linie eine Sicherheitskopie, und die legt man sicher weitaus öfter an als man sie benötigt. Und dafür ist BUSH sehr gut zu gebrauchen.

BUSH ist in Z80-Assembler geschrieben (keine Quellen) und benötigt CP/M 2.2 oder höher und ZCPR3x.

PACK

Zum Schluß noch ein kleines Progrämmchen, das wahre Wunderdinge leistet: PACK. Es handelt sich um einen Disk-Arranger, also ein Programm, das die Fragmentierung von Disks/Festplatten aufhebt oder zumindest verringert. Bei CP/M ist das sehr wichtig, eine Festplatte, die lange nicht mehr "aufgeräumt" wurde, ist wesentlich langsamer als eine frisch formatierte, auf der die Files noch alle schön in einer Reihe stehen. Darüber hinaus ist der Zugriff auf Dateien, die im Directory weit vorn stehen, schneller, denn CP/M 2.2 sucht linear. PACK erlaubt, eine Prioritätenliste anzugeben, mit der die Files arrangiert werden, so können z.B. oft benötigte Utilities an den Anfang der Platte geschoben werden. Ich benutze PACK regelmäßig, denn merke: Je öfter PACK gestartet wird, desto schneller ist es fertig!

PACK ist in Z80-Assembler geschrieben (keine Quellen) und benötigt CP/M 2.2 oder höher sowie ZCPR3x.



Kompression von Dateien unter CP/M.

Als CP/M Diskothekar bekomme ich immer wieder Anfragen, was diese seltsamen Files mit der Extension .LBR, .ARK oder .DZC wohl bedeuten mögen. Dabei handelt es sich um komprimierte Dateien. Man unterscheidet zwei prinzipielle Arten der Kompression:

1. Datei-Kompression
2. Daten-Kompression

zu 1).

Datei-Kompression bedeutet, daß mehrere Dateien in eine einzige zusammengefaßt werden, um Directory-Slots und Diskettenplatz zu sparen. Unter CP/M gibt es zwei bekannte Verfahren : Libraries und Archive. Zunächst zur Motivation. Bekanntlich vergibt CP/M Diskettenplatz nicht in Sektoren oder gar byteweise, sondern in Blöcken (Groups). Ein Block kann unterschiedliche Größen haben, üblich sind Werte von 1 KByte bis 8 KByte. Im letzten Fall würde eine Datei, die nur aus einem einzigen Byte besteht, 8 KByte Diskettenplatz und einen Directoryeintrag verbraten. Das ist nicht sehr effektiv. Wenn nun mehrere Dateien, die überdies vielleicht auch noch logisch zusammenhängen, wie etwa ein Compiler nebst Overlays, Linker und Assembler, zusammen gespeichert werden sollen, organisiert man sie in einer Library bzw. einem Archiv. Man kann natürlich alle möglichen Dateien in einer LBR oder einem Archiv zusammenfassen, z.B. um ein Backup einer Diskette herzustellen (was allerdings auf Speicherprobleme stoßen dürfte).
LBRs und ARKs haben ihrerseits ein Inhaltsverzeichnis, das die Größe und den Offset einer Datei in dieser Library angibt. Der Vorteil liegt auf der Hand : Es wird nur noch ein Directoryeintrag verbraucht und der Diskettenplatz wird ökonomischer verwaltet, denn in der Library nimmt jede Datei nur den tatsächlich beanspruchten Platz ein. Dasselbe gilt für Archive. Libraries sind an der Extension .LBR zu erkennen, Archive heißen entweder .ARK oder .ARC. Erstere sind unter CP/M zusammengestellt, letztere unter MSDOS, aber beide können unter CP/M ausgepackt werden.
Sowohl Libraries als auch ARKs lassen u.U. das Einfügen von zusätzlichen Dateien zu.

Standard-Utility für LBRs ist NULU.COM, für ARKs UNARC.COM. Zum Herstellen von Archiven ist neuerdings ARK Version 1.1 erhältlich, ein CP/M-Hacker namens Tillmann Reh aus Siegen hat eine Version 2.0 erstellt, um die ich mich bemühe. Es hat lange (ca. 2 Jahre) gedauert, bis die erste ARK-version verfügbar war, damals noch Version 0.7, voller Bugs.

Es gibt zunehmend auch noch andere Archivierungsmöglichkeiten, die unter CP/M Verbreitung finden, so zum Beispiel ZIPs oder ZOOs. Im Moment ist ihr Aufkommen jedoch noch sehr gering.
Die Extension .LBR ist NICHT mit .LIB zu verwechseln! Eine LIB ist normalerweise keine Library, sondern z.B. ein INCLUDE-File für einen Assembler. Auch TURBO-Modula-2 benützt LIBs, dabei handelt es sich tatsächlich um Libraries, allerdings nicht im LBR - Format.

zu 2).

Daten-Kompression ist die Reduktion von Dateien, d.h. die Entfernung redundanter Daten. Solche komprimierten Dateien sind am mittleren Buchstaben ihrer Extension zu erkennen, das ist ein Q, Z oder Y.
Kanditaten für DatenKompression sind alle Dateien, speziell Texte und vor allem Grafik-Daten. Dabei werden nicht selten Reduktionen von 50 % erreicht.
Ich weiß nicht genau, welche Algorithmen bei diesen Kompressionsmethoden benutzt werden. Es gibt eine Reihe verschiedener Algorithmen : Run-Length-Encoding, Squeezing, Crunching (GEL und LZH), Packing, Squashing...
Unter CP/M sind drei verbreitet.

a) SQUEEZING. Solche Dateien tragen in der Extension den Buchstaben Q. Viele Utilities erlauben das Entsqueezen von Dateien: WASH, NULU u.a.

b) CRUNCHING (GEL). Eine etwas effektivere Methode. Zu erkennen am Z in der Extension. Utility : UNCR.

c) CRUNCHING (LZH). Das Neueste. Die letzten Uncruncher erkennen die Methode und behandeln die Datei entsprechend. Zu erkennen am Buchstaben Y in der Extension.

Die neueste Utility, UNCRUNCH.COM Version 1.1, behandelt GEL, LZH und gesqueezete Dateien gleichermaßen.
Archive als auch Libraries enthalten in der Regel komprimierte Dateien, die nach dem Auspacken zu expandieren sind. Neuere Programme, z.B. LBREXT17.COM, expandieren automatisch.

Unter ZCPR33 gibt es eben dieses LBREXT17, ferner LPUT, LGET und VLU. Damit ist das Arbeiten wesentlich angenehmer. VLU ist z.B. Screen-orientiert und erlaubt das selektive Binden von Dateien in eine Library, außerdem das Betrachten von Library-Members, auch, wenn diese gecrunched oder gesqueezed sind.

Rüdiger.

Der Editor.

Der Editor heißt PMATE. Es gibt ihn für CP/M, MSDOS und CP/M 86. Er kann alles.
PMate ist unter CP/M (und diese Version, genauer gesagt 3.02, beschreibe ich hier) ein ca. 20 KByte großer COM-File. Das ist recht viel, aber PMate kann Texte verarbeiten, die länger sind als der vorhandene Speicher.
Wenn man PMate startet, sieht man eine recht spartanische Infozeile, in der der Name der Arbeitsdatei und die Cursorposition steht, ab und zu tauchen auch noch andere Meldungen auf. Unter der Infozeile ist die Kommandozeile, denn PMate ist ein Kommando-orientierter Editor.
Ein Kommando besteht aus einer Zeichenfolge, die mit der Eingabe zweier ESC beendet wird. Das ESC wird als '$' dargestellt. Das Kommando :

```
10m$$
```

bewegt den Cursor 10 Zeichen nach rechts. Um den gleichen Betrag nach links verrücken läßt er sich mit

```
-10m$$
```

Ganz logisch. Eine Zeile nach unten : 1$$. eine Zeile nach oben : -1$$. Zehn Zeilen zurück : -10l$$. Ganz klar. Lösche das Zeichen under dem Cursor : d$$. Füge einen String an der Cursorposition ein : idas steht jetzt hier$$. Natürlich kann PMate auch suchen und ersetzen. smate$$ findet den String MATE, mate oder MaTe, cmate$matt$$ ersetzt jedes Vorkommen von MATE durch den String matt. (Das $ steht immer für ESC).
Im Gegensatz zu WordStar-like Editoren sind die meisten PMate-Kommandos nicht an eine Taste bzw. Tastenkombination gebunden, sondern sie müssen als Kommando eingegeben werden. Dazu gibt es neben Insert- und Overwrite-Modus noch den Kommandomodus.
Hier eine kurze Aufstellung der an Tasten gebundenen Kommandos mit Erläuterungen (^A = CONTROL-A):

Cursortasten	bewegen den Cursor
^A	zum Anfang/Ende der Datei
^D	lösche Zeichen
BS	lösche Zeichen vor dem Cursor
^P	Wort rechts
^O	Wort links
^Q	lösche Wort links
^W	lösche Wort rechts
^U	Seite zurück
^B	Seite vor
^T	Setze Marke
^E	lösche von Marke bis Cursor
^R	hole vom Garbage Stack
^C	Breche Kommando ab
^X	Kommando Modus
^V	Overwrite Modus
^N	Insert Modus
^L	Neue Zeile
^Y	lösche Zeile

Hoffentlich habe ich nicht viel vergessen. Alle diese Tasten sind konfigurierbar, die Liste ist die Belegung meines PMate. Es kommen noch hinzu einige Tasten-Kommandos zur Text-Formatierung, aber PMate ist kein Textverarbeiter, sondern ein Editor, hier liegen seine starken Seiten.

PMate hat 11 Buffer, sie heißen T und 0 bis 9. Buffer T ist der Text-Buffer, mit ihm wird normalerweise gearbeitet. Aber man kann auch in die anderen Buffer Texte laden und sie modifizieren und speichern, auch zwischen den Buffern hin- und herschieben. Buffer 0 spielt eine besondere Rolle: hier wird abgelegt, was bei einem ^E gelöscht wurde. Deshalb kann man auch die WordStar Block-Operationen ^KB, ^KK, ^KV und ^KC nebst ^KY nachbilden. Wenn man will. Die Buffer 1 bis 9 dienen normalerweise als Hilfsbuffer für komplexe Kommandofolgen.

Die Kommandosprache von PMate ist unglaublich mächtig. Man kann damit regelrechte Programme schreiben, mit Arithmetik, Bedingungen, Wildcards, Schleifen, Breakpoints, Variablen, Ein/Ausgabe. Wenn ein solcher Kommandostring in einem der Buffer 0 - 9 steht, kann er als "Macro" aufgerufen werden, indem einfach ein ".", gefolgt vom Buffernamen, eingegeben wird. Ein Beispiel : Buffer 2 enthält den folgenden Text:

```
[M(@T=" )]
```

Das ist eine Kommandofolge. Sie könnte auch in der folgenden, kommentierten Form dort stehen:

```
[               ; Anfang Schleife
M               ; Cursor nach rechts
(               ; Bedingung
@T              ; Zeichen unter dem Cursor
=" )            ; gleich Leerzeichen/Ende Bedingung
]               ; Ende Schleife
```

Gehen wir zurück in den T-Buffer mit dem Kommando bte$$ (Buffer T Enter). Mit .2$$ wird nun das Programm ausgeführt. Solange das Zeichen unter dem Cursor kein Space ist, wird der Cursor nach rechts bewegt. Dann wird die Schleife und damit das Programm verlassen, der Cursor steht nun auf dem nächsten Leerzeichen hinter seiner Startposition.
Einfaches Drücken der ESC-Taste führt das Kommando nochmals aus. Sehen wir uns ein anderes, nützlicheres Macro an.

```
a               ; go to start of buffer
{               ; start loop ({ und [ ist beides m|glich)
(@t=";)                 ; if @t = ';'
[               ; then
k               ; kill rest of line
13i             ; insert chr(13)
1qr             ; update screen
]               ; end then
[               ; else
m               ; move 1 char
]               ; end else
}               ; end loop
```

Dieses Macro wird mit einem Fehler beendet, wenn das Ende der Datei erreicht wurde. Fehler können abgefangen werden. Man kann, neben der Schleifenkonstruktion, die if-then-else

Bedingung erkennen. Das Macro durchsucht die Datei nach dem Semikolon. Wenn es eines findet, wird ab dieser Position die Zeile gelöscht und ein <CR> wieder eingefügt. Sonst wird der Cursor auf das nächste Zeichen positioniert. Auf diese Weise kann ein Assemblerprogramm oder ein PMate-Macro von Kommentaren befreit werden. Ein '?' in einem Macro stellt einen Breakpoint dar, der Programmlauf stoppt und man kann im single-step durchlaufen.
Es gibt drei Schleifentypen : REPEAT_UNTIL ,WHILE_END und die Endlosschleife. Labels können definiert und, auch bedingt, angesprungen werden.

Im Zusammenhang mit Bedingungen ist es nützlich, wenn man PMates Variablen kennt. Hier sind einige:

@0-@9	: Benutzervariablen, numerisch.
@b	: der aktive Buffer
@e	: Error-Flag
@l	: Zeilennummer
@x	: Spaltennummer
@t	: das Zeichen unter dem Cursor
@m	: der freie Speicher

Einige Operatoren :

+	: Addition
-	: Subtraktion
*	: Multiplikation
/	: Division (integer)
!	: ODER
&	: UND
'	: NICHT

Die behandeleten Macros können natürlich als Datei gesichert und bei Bedarf geladen werden, man kann sie aber auch "permanent" machen. Dazu hat PMate eine seltsame Eigenschaft: er kann sich selbst klonen. hat man während einer PMate-Sitzung einige Installationen verändert, so sagt man einfach:

XDNEUPM.COM$$

und es gibt einen neuen PMate mit Namen NEUPM.COM auf der Disk. Das könnte z.B. ein PMate mit WordStar-Eigenschaften sein.

Die Kommandos, die PMate beherrscht, sind bei weitem zu zahlreich, als das sie hier aufgeführt werden könnten. Zusammen mit der mächtigen Macro-Sprache läßt sich PMate vermutlich dazu bringen, C-Quellcode in Assembler zu übersetzen. Ich habe ein Macro, das mit zwei Argumenten aufgerufen wird und den Textbuffer auf "matching parenthesis" untersucht, das können Klammerpaare sein, wie '{' und '}', aber auch Schlüsselworte wie 'begin' und 'end' oder '(*' und '*)'. Ein Pascal-Programmierer weiß sowas zu schätzen.

Habe ich etwa übertrieben?

Rüdiger

Impressum

1. Vorsitzender
Tel.: 0 24 64/89 20
Fritz Chwolka
Saarstraße 4
5173 Aldenhoven

2. Vorsitzender
Tel.: 0 41 05/26 02
Gerald Schröder
Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator
Tel.: 02 09/87 02 30
Andreas Magnus
Pommernstraße 4
4650 Gelsenkirchen

Newdos-Diskothekar
Tel.: 07 11/7 35 38 17
Oliver Volz
Waldburgstraße 73
7000 Stuttgart 80

CP/M-Diskothekar
Tel.: 0 61 31/3 28 60
Rüdiger Sörensen
Thomas Mann Straße 3a
6500 Mainz 1

Clubbücherei
Tel.: 0 41 52/7 06 43
Kurt Müller
Sophie-Scholl-Ring 38
2054 Geesthacht

Redaktion
Tel.: 07 91/4 28 77
Jens Neuder
Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

Autoren
Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 6209 Heidenrod
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Schluß

Hallo Club 80'er,

wieder einmal ist ein Jahreswechsel in greifbarer Nähe und somit auch die Zeit jedem das Beste für die Feiertage und einen erfolgreichen Sprung nach 91 zu wünschen. Gleichzeitig möchte ich mich bei allen Autoren und Mithelfenden für die geleistete Arbeit bedanken.
Ohne Euch wäre die TRS-80-Welt fast ausgestorben. Ich hoffe, daß wir auch im nächsten Jahr wieder gemeinsam einige INFO's und Sonderhefte zum Druck und natürlich an das Clubmitglied bringen werden.

Mit diesem Info werden Euch diesmal zwei Sonderhefte erreichen. Zum Einen über GEGRAS eine Benutzeroberfläche -welche die HRG auf dem TRS-80/Genie und HRGPACK benötigt (Softwarepaket GEGRAS ist in der Club-PD)- und Genie 3s Z180 -ein Hard- und Softwaretunup für den 3s-.
Wir hoffen, daß Ihr Gefallen an den Werken findet. Es steckt ja doch recht viel Arbeit dahinter, bis so etwas entsteht. Die Autoren werden sich sicher über ein kleines "Feedback" eurerseits freuen. Vielleicht ist es auch eine Animation für den einen oder anderen gleiches zu tun, und mir ein Sonderheft zur "Endfertigung" zuzusenden.

Als weiteres möchte ich, um die Qualität des INFO's zu steigern, wieder einmal einen Hinweis loswerden und ein Angebot machen. Zuerst zum Hinweis:
Vermeidet bitte die Verwendung von Umweltschutzpapier und älteren Farbbändern! Wie Ihr -teilweise in diesem INFO- seht, leidet der Kontrast und somit die INFO-Qualität darunter. Ich bin natürlich auch für Umweltschutz und optimaler Ausnutzung der Energie, aber für das INFO möchte ich doch um "ungeknicktes in schwarz auf weiß" bitten.
Nun zu meinem Angebot: Wie Ihr sicher am Druckbild dieses Textes erkennen könnt, habe ich die Möglichkeit meine schwarzen Buchstaben mit dem Laser zu brennen. Wenn Bedarf besteht kann ich ja für Euch Euren vorgefertigten Text ausdrucken. Dies erfordert aber, daß ich von Euch den Text auf IBM-kompatiblen Scheibchen (peinlicherweise doch wieder DoMesTos) zugeliefert bekomme. Größe und Art der Bespurung sind egal, sofern sie sich auf normale IBM-XT/AT-Formate beschränken. Der Text sollte am besten unter Word erstellt werden, da ich hiermit arbeite, es geht auch Text im ASCII-Format.
Sollten besondere Wünsche beim Layout berücksichtigt werden, ist ein Probeausdruck als Muster für mich sicher sehr hilfreich. Zumindest brauche ich dann eine genaue Anleitung. Bilder und Grafiken sind bitte extra mitzusenden, da Word nicht jedes Grafikformat im Text mit verarbeitet.

Zum Abschluß dieses INFO's möchte ich mich bei Euch bis zum nächsten INFO oder Treffen verabschieden. Viel Spaß mit dem neuen Lesestoff wünscht Euch

Jens Neueder

Club 80 Mitgliederadressenliste

Detlef Behrendt	Schlosserbreite 1a	D-	8018	Grafing
Achim Benner	Vorm Mühlberg 1	D-	5910	Kreuztal 3
Helmut Bernhardt	Hafenstraße 7	D-	2305	Heikendorf
Heinrich Betz	St. Wolfgangstraße 13	D-	8551	Hausen
Jörg Brans	Tieloh 55	D-	2000	Hamburg 60
Harald Braun	Postfach 8011	D-	2300	Kiel 17
Ulrich Böckling	Jochaczstraße 61	D-	5410	Hör-Grenzhausen
Fritz Chwolka	Saarstraße 34	D-	5173	Aldenhoven
Oskar Drechsler	Ducksteratherr Busch 2	D-	5060	Bergisch Gladbach 2
Hans-Joachim Eilers	Theodor-Heuss-Straße 129	D-	2900	Oldenburg
Richard Frey	Wolfsgrube 45	D-	3422	Bad Lauterberg
Werner Förster	Christoph-Krebs-Straße 9	D-	8720	Schweinfurt
Jens Günther	Bannerscheid 7	D-	5231	Neitersen
Hans-Günther Hartmann	Möwenstraße 9	D-	2876	Berne 2
Manfred Held	Stirner Straße 22	D-	8835	Pleinfeld
Werner Hentz	Am Tränkgarten 20	D-	6457	Maintal 2
Klaus Hermann	Forchenstraße 8	D-	7401	Pliezhausen
Matthias Homann	Hermesweg 21	D-	2000	Hamburg 90
Willi Johnen	Hansemannstraße 1	D-	5160	Düren
Jürgen Kemmer	Dorfberg 7	D-	8701	Sulzdorf
Mary Jo Kostya	Balberstraße 68	CH-	8038	Zürich
Eckehard Kuhn	Im Dorf 14	D-	7443	Frickenhausen
Claus Littmann	Plockhorst, Zum Spring 15	D-	3155	Edemissen
Walter Lorenz	Mahräckerstraße 9	D-	6000	Frankfurt /Main 50
Andreas Magnus	Pommernstraße 18	D-	4650	Gelsenkirchen
Herbert Mahlert	Baumschulstraße 7	D-	4100	Duisburg
Harald Mand	Kl. Flintbeker Straße 7	D-	2302	Flintbek /Kiel
Holger May	Marienstraße 9	D-	5768	Sundern 2
Heinz-Dieter Meklenburg	Amrumer Weg 1	D-	2262	Leck/NF
Christian Menk	Ollsener Straße 52	D-	2116	Hanstedt
Franz Mössel	Schafferstraße 12	I-	39012	Meran
Klaus-Jürgen Mühlenbein	Am Mönchgarten 28	D-	6940	Weinheim-Lützelsachsen
Kurt Müller	Sophie-Scholl-Ring 3b	D-	2054	Geesthacht
Gerhard Neebe	Märkische Straße 186	D-	4600	Dortmund 1

Club 80 Mitgliederadressenliste

Jens Neueder	Rudolf-Then-Straße 32	D-	7178	Gschlachtenbretzingen
Christof Neumann	Theodor-Heuss-Straße 8	D-	7265	Oberhaugstett
Stefan Nitschke	Germanenstraße 5	D-	7519	Walzbachtal 1
Hartmut Obermann	Mozartring 23	D-	8870	Günzburg
Bernd Retzlaff	Kleiner Sand 98	D-	2082	Uetersen
Gerd Rinio	Rennbahnstraße 9	D-	2000	Hamburg 74
Claus Ruschinski	Pommernstraße 21	D-	4370	Marl
Werner Schilling	Ehndorfer Straße 340	D-	2350	Neumünster
Alexander Schmid	St. Cajetan Straße 38/VII	D-	8000	München 80
Paul-Jürgen Schmitz	Bremer Straße 9	D-	6236	Eschborn
Rainer Schmitz	Dornierstraße 17	D-	7320	Göppingen
Frank-Michael Schober	Weberweg 2	O-	7590	Spremberg
Uwe Schoberth	Petrus-Waldus-Straße 14	D-	7136	Oetisheim
Frank Schoof	Elpke 5	D-	4800	Bielefeld 1
Horst-Dieter Schroers	Breslauer Straße 9	D-	8016	Feldkirchen
Gerald Schröder	Am Schützenplatz 14	D-	2105	Seevetal 1
Peter Schröder	Theodor-Fahr-Straße 32	D-	2000	Hamburg 62
Egbert Schröer	Joachimstraße 18	D-	4270	Dorsten 1
Andre Schut	Sanderstraße 26	D-	1000	Berlin 44
Wolfgang Schwarz	Schwedenring 6	D-	8850	Donauwörth
Walter Schäfer	Rathausstraße 4	D-	8160	Miesbach
Jörg Seelmann-Eggebert	Henri-Spaak-Straße 96	D-	5305	Alfter 4
Svend A. Soerensen	Bogholder Allee 76A	DK-	2720	Vanloese
Arnulf Sopp	Wakenitzstraße 8	D-	2400	Lübeck
Hans-Martin Stephan	Am Glasesch 9a	D-	4506	Hagen a. TW
Stefan Stumpferl	Hasenberglstraße 57	D-	8000	München 45
Rüdiger Sörensen	Thomas-Mann-Straße 3a	D-	6500	Mainz 1
Klaus Thieleke	Ahrenshooper Straße 47	O-	1093	Berlin
Wilhelm Tornow	Görlitzer Straße 16	D-	2190	Cuxhaven 13
Richart Vollkmer	Am Spörkel 69	D-	6700	Dortmund 50
Oliver Volz	Waldenburgstraße 73	D-	7000	Stuttgart 80
Michel Waccus	Mühlhofweg 2a	CH-	8266	Steckborn
Heinz Wittkamp	Hindenburgstraße 37	D-	5630	Remscheid
Hans-Otto Wulf	Im Brahmkamp 38	D-	4250	Bottrop

04723/1355

HARDWARE * SOFTWARE
COMPUTER-CLUB 80
TANDY-/ GENIE- ANWENDER
ERFAHRUNGSAUSTAUSCH

Computer Unser

Computer unser, der Du bist in der Zentrale
Geheiligt sei dein Bildschirm
Deine Eingabe komme
Dein Wille geschehe wie im Speicher
So auch auf dem Drucker
Unsere täglichen Listen gib uns heute
Und vergib uns unsere Fehler
Obwohl wir nicht vergeben denen
die falsch programmiert haben
Lass uns nicht zu lange warten
Und erlöse uns von teuren Ausgaben
Denn Dein ist die Firma
Die Macht und das Personal
In Ewigkeit

Enter

CLUBINFO

33. AUSGABE

KONTAKTADRESSE : CLUB 80 / FRITZ CHWOLKA / SAARSTRAßE 34 / 5173 ALDENHOVEN
TEL.: 02464 / 8920

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Nordlichtertreffen 90 — 1
Kurt Müller
Termine — 2
Redaktion
Vorstellung — 3 - 4
Uwe Schoberth, Gert Halgasch

Software

Expertensystem in Basic ??!! — 5 - 24
Masken generieren mit TRS 80 MI und GENIE I — 25 - 27
Egbert Schröer
Hardcopyroutine für SEKOSHA GP 550A — 29
Hardcopyroutine für BROTHER M 1009 — 30
Schnelle Kreisfunktion unter BASIC80 — 31
Grafikdemo für uLabs Grafikkarte (M IV/IVp/III) — 32
Frank Schoof
Schönere Programme mit Turbo-Pascal — 33 - 37
Kleiner Terminkalender — 38 - 40
Alexander Schmid
WIPE — 41 - 44
Uwe Schoberth
Unter falschem Namen ALIASes mit ZCPR — 45 - 46
DU3 für CP/M Plus — 47 - 48
Alexander Schmid
Adreßautomat, die Zweite — 49 - 53
Arnulf Sopp
Modell IV/IVp-INFO's — 55 - 56
Jörg Brans
LeSript Druckersteuerzeichen — 57 - 58
Frank Schoof

Autor & Seite

Börse

Wer hat was -- wer will was — 59 - 60

Sonstiges

TRX-280 The Next Generation ot TRS-80 — 61 - 64

Club-Bücherei

Buchvorstellungen — 65 - 66
Kurt Müller

Die letzten Seiten

Impressum — 67
Schluß — 68
Redaktion
Mitgliederadressenliste — am INFO-Ende
M IV/IVp Sonderinfo — Beilage

Nordlichtertreffen 90

Auch dieses Jahr wieder eintreffen im Norden. Allerdings hatte ich mit dem Treffen so meine liebe Not. Bis zum Anmeldungsschluß lagen sage und schreibe drei Anmeldung vorliegen. Aufgrunddessen mußte ich das Treffen zu mir in den Keller verlegen, da ich nicht mit so wenig Leuten das halbe Lokal belegen konnte.

Am alles entscheidenden Tag ergab die Zählung der Häupter dann sechs Personen, mich eingerechnet:

> Jörg Brans, Gerd Rinio, Gerald Schröder, Hans-Dieter Hartmann, Andreas Magnus und meine Wenigkeit

Andreas hatte sich kurzfristig per Telefon angekündigt und Gerd noch am Samstag eingefunden.

An digitalem Handwerkszeug waren vertreten zwei Model 4P (Jörg & Hans-Dieter) sowie ein Model II im Towergehäuse (Andreas). Meine Bitschleuder mußte mangels Monitor im Stall bleiben.

Tagesordnung gab es keine, es wurde nachdem alles "hergerichtet" (oder auch hingerichtet ?) war, munter an den mitgebrachten Arbeitspaketen gewerkelt. Gerald versuchte aus der Hüfte geschossen eine Software zu erstellen, welche in der Lage war, die Filegröße und einige weitere interessante Dinge die CP/M nicht so ohne weiteres herausrückt, zu lesen. Jörg zeigte mir die neueste Version der Textverarbeitung LeScript. Leider hatte er mit seiner eingedeutschten Version bei den Umlauten formatierprobleme, die, wie er sagte, dem Hersteller Anitek wohl noch einige Schweißperlen abringen werden. Andreas kämpfte mit Refresh-Problemen. Seine Z280 CPU verstand sich nicht so ganz mit dem Rest der Welt (=Hardware). Leider gelang es auch mit vereinten Kräften nicht, die Ursache einzukreisen.

Gegen Sonntag Mittag, nach dem Essen wurde dann das große Einpacken und Heimreisen gestartet.

Reit

Termine... Termine... Termine... Termine... Termine

Jahreshauptversammlung	Weilburg	09.05.- 12.05.91
Clubtreffen Nord	....	......- 91

Redaktionsschluß für die Clubinfo's 1991 ist jeweils der letzte Tag der folgenden Monate:

Mai
Juli
Oktober
Dezember

IAA	Frankfurt	12.09.- 22.09.91
Büro-data	Berlin	16.10.- 19.10.91
SYSTEMS	München	21.10.- 25.10.91

Liebe Club 8Ø - Mitglieder, lieber Jens,

dem Trend des allgemeinen Du folgend hoffe ich auf Einverständnis mit dieser Anrede. Ich bin nun seit Herbst 9Ø Mitglied und möchte mich kurz vorstellen, nachdem Fritz Chwolka uns Neulinge dazu ja ausdrücklich aufgefordert hat. (Die Sitten eines Clubs muß man ja erst lernen).

Ich bin 45 Jahre alt, insofern also auch ein "Fossil", zwar nicht aus der Steinzeit, wie Fritz meinen Computer mal bezeichnete, aber doch nur eben noch ein Nachkriegsjahrgang und somit nicht mehr ganz jung. Bin verheiratet, habe zwei Töchter im juristischen Erwachsenenalter (persönlich wird man's ja erst etwas später), wohne im eigenen Haus (teure Sache) mit großem Garten, der viel Arbeit mit sich bringt und bin ganz froh, daß unsere kleine Ortsgemeinde mit nur ein paar Hundert Einwohnern gerade noch ganz abgelegen liegt.

Beruflich bin ich Ingenieur bei der Deutschen Bundespost Telecom (Neudeutsch!) und bekam hier diesen "meinen" Computer, einen Triumpf Adler AlphaTronic P3. Betriebssystem CP/M 2.2. Dies führte mich damals über Uwe Herczeg zu Fritz Chwolka und dem Club 8Ø. Meinen ersten Eindruck davon bekam ich beim Treffen in Mannheim, wo ich mich dann abends beim Grillessen heimisch zu fühlen begann. Dank sei hier nochmals vor allem Fritz Chwolka gesagt, der sich auch im Nachhinein voll für mich engagierte.

Und damit bin ich beim Eigentlichen. Mein Rechner hat eine 8Ø85-CPU. Die ist leider nur 8Ø8Ø-kompatibel und viele der PD-Programme haben Z8Ø-Befehle. Also, wer interessante 8Ø8Ø-Programme hat, wird bei mir immer auf Interesse stoßen. Und noch eine Bitte habe ich. Fritz Chwolka hat's schon angedeutet. Wegen eines fehlenden MOVECPM-Programmes für meinen Rechner können wir die Installation von ZCPR nicht (vorerst ?) durchführen. Wer kennt jemanden, der dies Programm für Triumpf Adler AlphaTronic P3 hat ?

Ansonsten bemühe ich mich einerseits, Assemblersprache zu verstehen und andererseits die Steuerung der Diskettenlaufwerke zu begreifen. Ohne eine BIOS-Beschreibung ist das natürlich schwer.

Mit freundlichem Gruß

Uwe Schoberth
Petrus-Waldus-Str. 14
Schönenberg
W-7136 Ötisheim

Vorstellungen neuer Clubmitglieder

Lieber Jens,

da ich mich nun doch entschlossen habe Mitglied im Club zu werden, möchte ich mich kurz vorstellen.

Auch wenn ich mich schon mehr als sieben Jahre mit Mikroelektronik befasse, bin ich doch mit meinen fast vierzig Jahren auf der Computerstrecke ein blutiger Anfänger. Ein Computer war für viele Interessierte in der Ex-DDR ein schwer erfüllbarer Wunsch. Für meinen Ersten, einen Philips VG 8010, mußte ich mir das Rauchen abgewöhnen. (Man braucht ja für die Frau Argumente !) Und dann habe ich mir den Kasten für unsere "stabile Ostmark" für den Preis gekauft, für den ich heute einen IBM-Compatiblen mit wenigstens 40-er Festplatte haben könnte. Im vorigem Jahr habe ich mir dann aus der Konkursmasse von Robotron den Alba-PC gekauft. Das ist eine modifizierte Variante des Bildungscomputers, kurz BIC genannt. Dieses Gerät sollte in allen Schulen und Bildungseinrichtungen der DDR eingesetzt werden. Es arbeitet mit der CPU U880 (Z80), ist mit 3,75 MHz getaktet und hat folgende Speicher: 40 kbyte ROM (Treiber,BASIC-Interpreter)
64 kbyte RAM
64 k x 16 bit VRAM
Grafik: 320 x 200 (max 6 Seiten)
640 x 200 (max 3 Seiten)
Diskeinheit : 2 x 5,25 Zoll, 780 kbyte

Durch Zufall fand ich in einer Zeitschrift die Adresse von Claus (Vorstellung im 32. Clubinfo). Er half mir über die ersten Hürden und interessierte mich für den Club.

Meine nächsten Ziele: System ausbauen (ich arbeite immer noch mit einem stinknormalen Kofferfernseher), RAM-Floppy, Maus..., und mehr in die Programmierung (TPascal) schnüffeln. Vorausgesetzt mir bleibt die nötige Zeit.
Ich bin Ingenieur für Kraftwerksanlagen und arbeite als technischer Leiter eines Fernwärmeversorgungsbetriebes. Privat interessiert mich Alles, was mit dem Thema Heizen, Heizanlagen, Eneriesparen usw. zu tun hat und natürlich auch alle Software zu diesem Thema.
Das sollte für eine kurze Vorstellung genügen.
Ich freue mich über jeden Kontakt.

Gert Halgasch
Großschönauer Str.26
O-8805 Jonsdorf

PS: Wer jetzt Jonsdorf sucht- es liegt im Dreiländereck Polen, Tschechoslowakei, Deutschland- in der Nähe der Stadt Zittau.

Expertensystem in Basic ??!!

KI Programmierung mit TRS 80/GENIE Computern

1. Einführung
Durch Zufall fiel mir vor ca. einem Jahr ein Buch mit folgendem Titel auf: KI Expertensysteme programmieren
Einführung in die Funktionsweise mit Beispielen in BASIC
Das Thema begann mich zu faszinieren und es reifte der Gedanke die im Buch vorgestellten Routinen in BASIC auf unserere Small System Computer zu übertragen.
Schwierigkeiten gab es genügend, aber mittlerweile liegt eine lauffähige Version vor, die ich hier vorstellen möchte. Das Programmsystem setzt sich, um Speicherplatz zu sparen, aus mehreren Einzelprogrammen zusammen und gibt auch für andere Problemlösungen in BASIC gute Programmierbeispiele.

Es stellte sich beim Programmieren und Durcharbeiten des Buches heraus, daß Expertensysteme nichts mystisches und keine KI im eigentlichen Sinn, sondern nur eine besondere Programmiertechnik sind.

Aus diesem Grund habe ich mich trotz der Länge der Sources enschlossen diese abzudrucken. Falls jemand Lust bekommt den ganzen Krempel bis ins Detail nachzuvollziehen: die Zeilennum- merierung entspricht den Programmierbeispielen im Buch
Was leistet nun der ganze Krempel ?
Als Testkomponente enthält die Disk eine Wissensbasis um ein Radio leidlich zu reparieren. Vorstellbar wäre das sich einige Hardware "Experten" eine Wissensbasis z.B. zur Reparatur des Trash 80 erarbeiten.
Allerdings: ein fertiges Expertensystem (entgegen manchen Versprechungen bei "großen Systemen) eignet sich immer nur für wenige gleichartige Probleme.
Das vorgestellte Programmsystem liefert aber ein funktionsfähiges Skelett, das sich zu einem voll einsatzfähigem Expertensystem ausbauen läßt.
Das Beispiel zur Arbeitsweise ist dem Eingangs vorgestellten Buch entnommen.

1.1 Arbeitsweise
Ein Expertensystem ahmt die Arbeitsweise einer Person nach, die aufgrund ihres Fachwissens Probleme in einem begrenzten Fachgebiet lösen kann.
Beispiel:
Mein Transistorradio gibt seinen Geist auf. Ich habe einen Bekannten mit dem löblichen Beruf des Radiotechnikers. Folgender Dialog könnte sich entwickeln:

Ich : Mein Radio ist defekt
Experte : Spielt das Radio, kann man Sprache oder Musik hören ?
Ich : Nein.
Experte : **Hört man bei voller Lautstärke mindestens ein leises Rauschen ?**
Ich : **Ja.**
Experte : **Endstufe oder Lautsprecher sind wahrscheinlich in Ordnung. Schlägt die Empfangsanzeige aus ?**
Ich : **Ja, etwas.**
Experte : **Empfangsstufe ist auch oK. Der Fehler liegt zwischen Eingangsstufe und Endstufe, wahrscheinlich ist der NF-Vorverstärker defekt.**

Ein Experte stellt also Fragen um Fakten zu sammeln. Anschließend erfolgt eine Auswertung/Bewertung der gesammelten Fakten und eine Begründung.

Das Expertensystem soll diese Vorgehensweise nachahmen. Deshalb besteht es aus einer sogenannten Wissensbasis und einem Diagnosesystem.

2. Die Wissensbasis
Die Wissensbasis enthält das "Wissen" des Experten. Mit diesem Wissen arbeitet das Diagnosesystem (Inferenzkomponente), welches Eingabedaten vom Benutzer erwartet in der Form von, um bei dem Beispiel des Radios zu bleiben:
- nicht - EMPFANGSANZEIGE
Trifft diese Aussage zu Ja/Nein ?
Die Inferenzkomponente arbeitet also mit der Wissensbasis, die sinnvoll strukturiert sein muß. Z.B. kann man auf das Wissen eines Lexikons durch die alphabetische Sortierung nach Schlag- worten leicht zugreifen.
Das Wissen kann in zwei Untereinheiten untergliedert werden, in Faktenwissen (ein Computer hat einen Prozessor) und Regelwissen (wenn der Computer einen Prozessor hat und keinen Strom, dann läuft er nicht).

Fakten sind im einfachsten Fall wahr oder falsch:

Aussage	Darstellung	Zustand
Der Computer hat einen bernsteinfarbenen Monitor	BERNSTEIN-MONITOR	wahr
Der Computer hat eine Floppy Disk	TAPE-RECORDER	falsch

Das Regelwissen hat eine göße̸re Bedeutung für unser Expertensystem, da Regeln zu weiterem Wissen führen, wenn bestimmte Fakten vorhanden sind.
Beispiel:
Aus DIE-SONNE-SCHEINT ist wahr folgt ES-IST-TAG ist wahr.
Genug der grauen Theorie, es folgt die Praxis.

2.1 Aufbau der Wissensbasis
In der Wissensbasis sind FRAGEN% mit MF% (max. Fragen) Fragen enthalten und werden wie folgt in Felder unterteilt:

```
DIM FTEXT$(FRAGEN%),(FA$(2,FRAGEN%),FSTATUS%(FRAGEN%),
    FOBJEKT%(FRAGEN%)
```

FTEXT$(N%)	Text der N-ten Frage
FA$(1,N%)	Antwort, wenn der Zustand des zur Frage gehörigen Objektes wahr sein soll
FA$(2,N%)	s.o., aber Zustand falsch
FSTATUS%(N%)	Zustand der Frage (0=unbestimmt, 1=erste Antwort wurde ausgewählt, -1=zweite Antwort wurde ausgewählt
FOBJEKT$(N%)	Name des Objektes, auf das sich die Frage bezieht

Für die Regeln wird nach folgendem Schema verfahren:
Die Regel

```
      Wenn gilt 1.Prämisse-1      und
                2.Prämisse-2      und
                usw.
      dann      AKTION
```

Beispiele:
Fragenummer : 1

```
Fragetext           : Klingt Musik dumpf ?
Antworttext 1       : Ja
Antworttext 2       : Nein
Bezugsobjekt-Name:  KLANG-DUMPF

Regelnummer         : 1
Regel/Diagnose      : Diagnose
Prämisse 1          : EMPFANGSTEIL-IN-ORDNUNG
Prämisse 2          : ENDSTUFE-KEIN-EINGANGSSIGNAL
Prämisse 3          : -
Prämisse 4          : -
Prämisse 5          : -
Aktion/Diagnose     : NF-VORVERSTARKER-DEFEKT

Objektnummer        : 1
Objekttyp           : Regel
Objektname          : KLANG-DUMPF
```

Für weiter Details verweise ich auf den Literaturhinweis

2.2 Der Editor

Das Programm WBEDIT ist der Editor um Fragen, Regeln, Objekte, Diagnosen und Fakten (Constraints) in die Wissensbasis einzutragen und zu speichern.

```
1 'Wissenserwerbskomponente
2 'Dr. H.-J. Soll         1987
3 'modifiziert für Tandy TRS80 Mod.1
4 'von Egbert Schröer     1990
5 CLEAR 7000
10 MO%=200
20 MF%=50
30 MR%=100
40 MC%=50
50 MQ%=20
60 DIM OA$(MO%),OSTATUS%(MO%),ODIAG%(MO%)
70 DIM FTEXT$(MF%),FA$(2,MF%),FSTATUS%(MF%),FOBJEKT$(MF%)
80 DIM RSTATUS%(MR%),WENN$(5,MR%),WF%(5,MR%),DANN$(MR%),TYP%(MR%)
90 DIM CN$(2,MC%),CS%(MC%)
100 DIM QU$(MQ%)
110 DIM W$(5)
120 CLS
130 PRINT §5,"Was möchten Sie ....."
140 PRINT §133,"1. Bestehende Wissensbasis bearbeiten"
150 PRINT §197,"2. Neue Wissensbasis erzeugen"
160 PRINT §329,"Bitte wählen Sie : ";
170 I$=INKEY$:IF I$="" GOTO 170
180 PRINT I$:I%=VAL(I$)
190 PRINT §581,"Name der Wissensbasis : ",:LINEINPUT WN$
200 ON I% GOTO 230,510
220 PRINT §348," ":PRINT §606,"                    ":GOTO 160
230 OPEN "I",1,WN$
240 INPUT #1,OBJEKTE%:INPUT #1,FRAGEN%:INPUT #1,REGELN%
250 INPUT #1,CO%:INPUT #1,NQ%
260 IF OBJEKTE%=0 GOTO 300
270 FOR N%=1 TO OBJEKTE%
280 INPUT #1,OA$(N%):INPUT #1,OSTATUS%(N%):INPUT #1,ODIAG%(N%)
290 NEXT N%
300 IF FRAGEN%=0 GOTO 340
310 FOR N%=1 TO FRAGEN%
320 INPUT #1,FTEXT$(N%):INPUT #1,FA$(1,N%):
    INPUT #1,FA$(2,N%):INPUT #1,FSTATUS%(N%):
    INPUT #1,FOBJEKT$(N%)
330 NEXT N%
340 IF REGELN%=0 GOTO 420
350 FOR N%=1 TO REGELN%
360 INPUT #1,RSTATUS%(N%)
370 FOR M%=1 TO 5
380 INPUT #1,WENN$(M%,N%):INPUT #1,WF%(M%,N%)
390 NEXT M%
400 INPUT #1,DANN$(N%):INPUT #1,TYP%(N%)
410 NEXT N%
420 IF CO%=0 GOTO 460
430 FOR N%=1 TO CO%
440 INPUT #1,CN$(1,N%):INPUT #1,CN$(2,N%):INPUT #1,CS%(N%)
450 NEXT N%
460 IF NQ%=0 GOTO 480
470 FOR N%=1 TO NQ%:INPUT #1,QU$(N%):NEXT N%
480 CLOSE #1
490 GOTO 1010
500 'Neue Wissensbasis erzeugen
510 OBJEKTE%=0:FRAGEN%=0:REGELN%=0:CO%=0:QU%=0
520 GOTO 1010
1000 'Wissensbasis bearbeiten
1010 CLS
1020 PRINT §5,"Was möchten Sie ....."
1030 PRINT §133,"1. Fragen bearbeiten"
1040 PRINT §197,"2. (Diagnose-)Regeln bearbeiten"
1050 PRINT §261,"3. Objekte bearbeiten"
1060 PRINT §325,"4. Sichern und Beenden"
1070 PRINT §584,"Bitte wählen Sie : ";
1080 I$=INKEY$: IF I$="" GOTO 1080
1090 PRINT I$:I%=VAL(I$)
1100 ON I% GOTO 2010,5010,8010,10010
1140 PRINT §603," ":GOTO 1070
2000 'Fragen bearbeiten
2010 CLS
2020 PRINT §5,"Was möchten Sie ....."
2030 PRINT §133,"1. Frage hinzufügen"
2040 PRINT §197,"2. Fragen auflisten"
2050 PRINT §261,"3. Fragen löschen"
2060 PRINT §325,"4. Eingangsfragen festlegen"
2070 PRINT §584,"Bitte wählen Sie : ";
2080 I$=INKEY$:IF I$="" GOTO 2080
2090 PRINT I$:I%=VAL(I$)
2100 ON I% GOTO 2210,2610,2810,12010
2140 PRINT §603," ":GOTO 2070
2200 'Frage hinzufügen
2210 GOSUB 4010
2220 IF FRAGEN%=MFRAGEN% GOTO 11010
2230 FRAGEN%=FRAGEN%+1
2240 PRINT §83,"";:PRINT CHR$(15);:PRINT FRAGEN%;
2250 ZEILE%=2:K%=20
2255 ZEILE%=ZEILE%+1:IF ZEILE%=9 THEN ZEILE%=10:K%=22
2257 IF ZEILE%>10 THEN 2290
2260 PRINT §ZEILE%*64+K%,"";:PRINT CHR$(14);
2265 A$=INKEY$:IF A$="" THEN 2265
2270 IF A$=CHR$(13) THEN PRINT CHR$(15);:GOTO 2255
2275 IF POS(0)=61 THEN PRINT A$;:PRINT CHR$(15);:GOTO 2255:ELSE
PRINT A$;:GOTO 2265
2290 PRINT §910,"Eintrag in Ordnung ? (j/n)";
2300 I$=INKEY$:IF I$="" GOTO 2300
2310 IF (I$<>"j" AND I$<>"J") THEN FRAGEN%=FRAGEN%-1:GOTO 2210
2320 GOSUB 4110:GOTO 1010
2600 'Fragen auflisten
```

```
2610 CLS
2620 IF FRAGEN%<>0 GOTO 2670
2630 PRINT §584,"*** keine Fragen in der Wissensbasis ***"
2640 PRINT §652,"*** Bitte eine Taste drücken ***"
2650 I$=INKEY$:IF I$="" GOTO 2650
2660 GOTO 1010
2670 PRINT §83,"":GOSUB 4010
2680 FOR N%=1 TO FRAGEN%
2690 PRINT §83,N%;
2700 PRINT §212,LEFT$(FTEXT$(N%),42);:
     PRINT §212+64,RIGHT$(FTEXT$(N%),42);
2710 PRINT §340,LEFT$(FA$(1,N%),42);:
     PRINT §340+64,RIGHT$(FA$(1,N%),42);
2720 PRINT §468,LEFT$(FA$(2,N%),42);:
     PRINT §468+64,RIGHT$(FA$(2,N%),42);
2730 PRINT §662,FOBJEKT$(N%);:PRINT CHR$(15);
2740 PRINT §913,"Zum Abbruch <E> drücken"
2750 PRINT §968,"Zur Fortsetzung eine andere Taste drücken";
2760 I$=INKEY$:IF I$="" GOTO 2760
2770 IF (I$="E" OR I$="e") GOTO 1010
2780 NEXT N%
2790 GOTO 1010
2800 'Frage löschen
2810 CLS
2820 PRINT §261,"Geben Sie bitte die Nr. der zu"
2830 PRINT §325,"löschenden Frage ein  :  ";:INPUT N%
2840 IF N%<=FRAGEN% GOTO 2890
2850 PRINT §584,"***** Fragennummer zu groß *****"
2860 PRINT §651,"Bitte eine Taste drücken"
2870 I$=INKEY$:IF I$="" GOTO 2870
2880 GOTO 1010
2890 IF NQ%=0 GOTO 2930
2900 FOR M%=1 TO NQ%
2910 IF FOBJEKT$(N%)=QU$(M%) THEN CLS:
     PRINT §133,"*** Frage ist Eingangsfrage ***":GOTO 2860
2920 NEXT M%
2930 PRINT §83,"":GOSUB 4010
2940 PRINT §83,N%;
2950 PRINT §212,LEFT$(FTEXT$(N%),42);:
     PRINT §212+64,RIGHT$(FTEXT$(N%),42);
2960 PRINT §340,LEFT$(FA$(1,N%),42);:
     PRINT §340+64,RIGHT$(FA$(1,N%),42);
2970 PRINT §468,LEFT$(FA$(2,N%),42);:
     PRINT §468+64,RIGHT$(FA$(2,N%),42);
2980 PRINT §662,FOBJEKT$(N%);:PRINT CHR$(15);
2990 PRINT §910,"Soll diese Frage gelöscht werden (j/n) ";
3000 I$=INKEY$:IF I$="" GOTO 3000
3010 IF (I$<>"j" AND I$<>"J") THEN FRAGEN%=FRAGEN%-1:GOTO 1010
3020 IF N%=FRAGEN% GOTO 3090
3030 FOR M%=N%+1 TO FRAGEN%
3040 FTEXT$(M%-1)=FTEXT$(M%)
3050 FA$(1,M%-1)=FA$(1,M%)
3060 FA$(2,M%-1)=FA$(2,M%)
3070 FOBJEKT$(M%-1)=FOBJEKT$(M%)
3080 NEXT M%
3090 FRAGEN%=FRAGEN%-1
3100 GOTO 1010
4000 'Fragenmaske
4010 CMD"load maske1/dum"
4070 RETURN
4100 'Einlesen aus Fragemaske
4110 PRINT CHR$(15);:I2%=1:I3%=2
```



```
4120 FOR I1%=0 TO 2
4130 FOR I%=I2% TO I3%:FOR K%=0 TO 41
4140 A$(I1%)=A$(I1%)+CHR$(PEEK((2+I%)*64+15380+K%)):
     POKE(2+I%)*64+15380+K%,46:NEXTK%,I%:I2%=I2%+2:I3%=I3%+2:
     NEXT I1%
4145 FOR K%=0 TO 39:
     A$(3)=A$(3)+CHR$(PEEK(10*64+15382+K%)):
     POKE(10*64)+15382+K%,46:NEXTK%
4150 FTEXT$(FRAGEN%)=A$(0)
4160 FA$(1,FRAGEN%)=A$(1)
4170 FA$(2,FRAGEN%)=A$(2)
4180 FOBJEKT$(FRAGEN%)=A$(3)
4185 FOR I1%=0 TO 3:A$(I1%)="":NEXT I1%
4190 RETURN
5000 'Regeln bearbeiten
5010 CLS
5020 PRINT §5,"Was möchten Sie ....."
5030 PRINT §133,"1. (Diagnose-)Regeln hinzufügen"
5040 PRINT §197,"2. (Diagnose-)Regeln auflisten"
5050 PRINT §261,"3. (Diagnose-)Regeln löschen"
5060 PRINT §584,"Bitte wählen Sie : ";
5070 I$=INKEY$:IF I$="" GOTO 5070
5080 PRINT I$:I%=VAL(I$)
5090 ON I% GOTO 5210,5810,6110
5120 PRINT §603," ":GOTO 5060
5200 'Regeln hinzufügen
5210 GOSUB 7010
5220 IF REGELN%=MR% GOTO 11010
5230 REGELN%=REGELN%+1
5240 PRINT §18,REGELN%;
5250 PRINT §147,"";:INPUT DIAGNOSE$
5260 DIAGNOSE%=0:
     IF (LEFT$(DIAGNOSE$,1)="d" OR LEFT$(DIAGNOSE$,1)="D")
     THEN DIAGNOSE%=1
5270 TYP%(REGELN%)=DIAGNOSE%
5280 PRINT §275,"";:INPUT W$(1):V$=W$(1):GOSUB 5690
5290 PRINT §339,"";:INPUT W$(2)
5300 PRINT §403,"";:INPUT W$(3)
5310 PRINT §467,"";:INPUT W$(4)
5320 PRINT §531,"";:INPUT W$(5)
5330 PRINT §659,"";:INPUT D$
5340 PRINT §910,"Eintrag in Ordnung ? (j/n)";
5350 I$=INKEY$:IF I$="" GOTO 5350
5360 IF (I$<>"j" AND I$<>"J") THEN REGELN%=REGELN%-1:GOTO 5210
5370 FOR N%=1 TO 5
5380 OBJEKT$=W$(N%)
5390 IF W$(N%)<>"- oder -" GOSUB 5610:OTYP%=0:GOSUB 5510
5400 WENN$(N%,REGELN%)=OBJEKT$:WF%(N%,REGELN%)=VO%
5410 NEXT N%
5420 OBJEKT$=D$
5430 OTYP%=DIAGNOSE%:GOSUB 5510
5440 DANN$(REGELN%)=OBJEKT$
5450 GOTO 1010
5500 'Objekte für Regeln erzeugen
5510 IF OBJEKT$="" THEN RETURN
5520 IF OBJEKTE%=0 GOTO 5570
5530 FOR NUMMER%=1 TO OBJEKTE%
5540 IF OBJEKT$=OA$(NUMMER%) THEN RETURN
5550 NEXT NUMMER%
5560 IF OBJEKTE%=MOBJEKTE% GOTO 11010
5570 OBJEKTE%=OBJEKTE%+1
5580 OA$(OBJEKTE%)=OBJEKT$:ODIAG%(OBJEKTE%)=OTYP%
```

```
5590 RETURN
5600 ´"Nicht" bei Regel abtrennen
5610 VO%=1
5620 IF LEN(OBJEKT$)<7 THEN RETURN
5630 U$=LEFT$(OBJEKT$,6):GOSUB 20020:IF U$<>"NICHT" THEN RETURN
5640 OBJEKT$=RIGHT$(OBJEKT$,LEN(OBJEKT$)-6)
5650 IF LEFT$(OBJEKT$,1)=" " THEN
     OBJEKT$=RIGHT$(OBJEKT$,LEN(OBJEKT$)-1):GOTO 5650
5660 VO%=-1
5670 RETURN
5680 ´oder-Regel
5690 IF (LEFT$(W$,1)=" " OR LEFT$(W$,1)="-")
     THEN W$=RIGHT$(W$,LEN(W$)-1):GOTO 5690
5700 IF (RIGHT$(W$,1)=" " OR RIGHT$(W$,1)="-")
     THEN W$=LEFT$(W$,LEN(W$)-1):GOTO 5700
5710 U$=W$:GOSUB 20020:IF U$<>"ODER" THEN RETURN
5720 V$(1)="- oder -"
5730 RETURN
5800 ´Regeln auflisten
5810 CLS
5820 IF REGELN%<>0 GOTO 5870
5830 PRINT @584,"*** keine Regeln in der Wissensbasis ***"
5840 PRINT @657,"Bitte eine Taste drücken"
5850 I$=INKEY$:IF I$="" GOTO 5850
5860 GOTO 1010
5870 FOR N%=1 TO REGELN%
5880 GOSUB 7010
5890 PRINT @18,N%;
5900 DIAGNOSE$="Regel                ":
     IF TYP%(N%)=1 THEN DIAGNOSE$="Diagnose                "
5910 PRINT @147,DIAGNOSE$;
5920 FOR M%=1 TO 5
5930 W$=WENN$(M%,N%)
5940 IF WF%(M%,N%)=(-1) THEN W$="nicht "+W$
5950 IF W$="" THEN W$=" -"
5960 PRINT @64*M%+211,W$;
5970 NEXT M%
5980 PRINT @659,DANN$(N%);
5990 PRINT @848,"Zum Abbruch <E> drücken"
6000 PRINT @904,"Zur Fortsetzung eine andere Taste drücken"
6010 I$=INKEY$:IF I$="" GOTO 6010
6020 IF (I$="E" OR I$="e") GOTO 1010
6030 NEXT N%
6040 GOTO 1010
6100 ´Regeln löschen
6110 CLS
6120 PRINT @197,"Geben Sie bitte die Nr. der zu"
6130 PRINT @261,"löschenden Regel ein : ";:INPUT N%
6140 IF N%<=REGELN% GOTO 6190
6150 PRINT @584,"***** Regelnummer zu groß *****"
6160 PRINT @651,"Bitte eine Taste drücken"
6170 I$=INKEY$:IF I$="" THEN 6170
6180 GOTO 6110
6190 GOSUB 7010
6200 PRINT @18,N%;
6210 DIAGNOSE$="Regel                ":
     IF TYP%(N%)=1 THEN DIAGNOSE$="Diagnose                "
6220 PRINT @147,DIAGNOSE$;
6230 FOR M%=1 TO 5
6240 W$=WENN$(M%,N%)
6250 IF WF%(M%,N%)=(-1) THEN W$="nicht "+W$
6260 IF W$="" THEN W$=" -"
```



```
6270 PRINT @64*M%+211,W$;
6280 NEXT M%
6290 PRINT @659,DANN$(N%);
6300 PRINT @904,"Soll diese (Diagnose-)Regel gelöscht werden ?
     (j/n)";
6310 I$=INKEY$:IF I$="" GOTO 6310
6320 IF (I$<>"j" AND I$<>"J") GOTO 1010
6330 PRINT @787,"--- Bitte warten ---"
6340 FOR K%=1 TO 5
6350 OBJEKT$=WENN$(K%,N%):GOSUB 7110
6360 NEXT K%
6370 OBJEKT$=DANN$(N%):GOSUB 7110
6380 IF N%=REGELN% GOTO 6460
6390 FOR M%=N%+1 TO REGELN%
6400 FOR K%=1 TO 5
6410 WENN$(K%,M%-1)=WENN$(K%,M%)
6420 WF%(K%,M%-1)=WF%(K%,M%)
6430 NEXT K%
6440 DANN$(M%-1)=DANN$(M%):TYP%(M%-1)=TYP%(M%)
6450 NEXT M%
6460 REGELN%=REGELN%-1
6470 PRINT @777,"                                    "
6480 GOTO 1010
7000 ´Regelmaske
7010 CMD"load maske2/dum"
7100 RETURN
7110 ´ggf. Objekt löschen
7120 ANZAHL%=0
7130 FOR M%=1 TO OBJEKTE%
7140 IF OBJEKT$=WENN$(1,M%) THEN ANZAHL%=ANZAHL%+1
7150 IF OBJEKT$=WENN$(2,M%) THEN ANZAHL%=ANZAHL%+1
7160 IF OBJEKT$=WENN$(3,M%) THEN ANZAHL%=ANZAHL%+1
7170 IF OBJEKT$=WENN$(4,M%) THEN ANZAHL%=ANZAHL%+1
7180 IF OBJEKT$=WENN$(5,M%) THEN ANZAHL%=ANZAHL%+1
7190 IF OBJEKT$=DANN$(M%) THEN ANZAHL%=ANZAHL%+1
7200 NEXT M%
7210 IF ANZAHL%>1 THEN RETURN
7220 FOR M%=1 TO OBJEKTE%
7230 IF OA$(M%)=OBJEKT$ GOTO 7260
7240 NEXT M%
7250 RETURN
7260 IF N%=OBJEKTE% GOTO 7320
7270 FOR L%=M%+1 TO OBJEKTE%
7280 OA$(L%-1)=OA$(L%)
7290 OSTATUS%(L%-1)=OSTATUS%(L%)
7300 ODIAG%(L%-1)=ODIAG%(L%)
7310 NEXT L%
7320 OBJEKTE%=OBJEKTE%-1
7330 RETURN
8000 ´Objekte anzeigen
8010 CLS
8020 PRINT @5,"Was möchten Sie ....."
8030 PRINT @133,"1. Objekte auflisten"
8040 PRINT @197,"2. Constraints auflisten"
8050 PRINT @261,"3. Constraints eintragen"
8060 PRINT @325,"4. Constraints löschen"
8070 PRINT @584,"Bitte wählen Sie : ";
8080 I$=INKEY$:IF I$="" GOTO 8080
8090 PRINT I$:I%=VAL(I$)
8100 ON I% GOTO 8210,8610,9010,9410
8140 PRINT @603," ":GOTO 8070
8200 ´Objekte auflisten
```

```
8210 CLS
8220 IF OBJEKTE%<>0 GOTO 8270
8230 PRINT §584,"*** keine Objekte in der Wissensbasis ***"
8240 PRINT §657,"Bitte eine Taste drücken"
8250 I$=INKEY$: IF I$="" GOTO 8250
8260 GOTO 1010
8270 ZEILE%=2
8280 FOR N%=1 TO OBJEKTE%
8290 IF ZEILE%=2 THEN PRINT §0,"Objektnr.  Typ  Objektname"
8300 PRINT §64*ZEILE%+4,N%
8310 I$="R": IF ODIAG%(N%)=1 THEN I$="D"
8320 PRINT §64*ZEILE%+12,I$
8330 PRINT §64*ZEILE%+15,OA$(N%)
8340 ZEILE%=ZEILE%+2
8350 IF ZEILE%<14 GOTO 8400
8360 PRINT §904,"Bitte eine Taste drücken     <E>=Ende"
8370 I$=INKEY$: IF I$="" GOTO 8370
8380 IF (I$="E" OR I$="e") GOTO 1010
8390 CLS: ZEILE%=2
8400 NEXT N%
8410 PRINT §840,"******** Ende der Liste ********"
8420 PRINT §908,"Bitte eine Taste drücken"
8430 I$=INKEY$: IF I$="" GOTO 8430
8440 GOTO 1010
8600 'Constraints auflisten
8610 CLS
8620 IF CO%<>0 GOTO 8670
8630 PRINT §584,"*** keine Constraints in der Wissensbasis ***"
8640 PRINT §657,"Bitte eine Taste drücken"
8650 I$=INKEY$: IF I$="" GOTO 8650
8660 GOTO 1010
8670 ZEILE%=2
8680 FOR N%=1 TO CO%
8690 IF ZEILE%=2 THEN
     PRINT §0,"Nr.   Faktum 1  bedingt  Faktum 2"
8700 PRINT §64*ZEILE%,N%
8710 I$=CN$(1,N%)+"  ===>  "
8720 IF CS%(N%)=(-1) THEN I$=I$+"nicht  "
8730 I$=I$+CN$(2,N%)
8740 PRINT §64*ZEILE%+6,I$
8750 ZEILE%=ZEILE%+2
8760 IF ZEILE%<14 GOTO 8810
8770 PRINT §904,"Bitte eine Taste drücken     <E>=Ende"
8780 I$=INKEY$: IF I$="" GOTO 8780
8790 IF (I$="E" OR I$="e") GOTO 1010
8800 CLS: ZEILE%=2
8810 NEXT N%
8820 PRINT §840,"********* Ende der Liste ********"
8830 PRINT §908,"Bitte eine Taste drücken"
8840 I$=INKEY$: IF I$="" GOTO 8840
8850 GOTO 1010
9000 'Constraints eintragen
9010 CLS
9020 PRINT §133,"Faktum 1 :"
9030 PRINT §261,"bewirkt ==>"
9040 PRINT §389,"Faktum 2 :"
9050 CO%=CO%+1: IF CO%>MC% GOTO 11000
9060 PRINT §133+12,"";: INPUT CN$(1,CO%)
9070 PRINT §389+12,"";: INPUT I$
9080 CS%(CO%)=1
9090 IF LEN(I$)<7 GOTO 9140
9100 U$=LEFT$(I$,6): GOSUB 20020: IF U$<>"NICHT " GOTO 9140
```



```
9110 I$=RIGHT$(I$,LEN(I$)-6)
9120 IF LEFT$(I$,1)=" " THEN I$=RIGHT$(I$,LEN(I$)-1): GOTO 9120
9130 CS%(CO%)=-1
9140 CN$(2,CO%)=I$
9150 PRINT §648,"Eintrag in Ordnung ?  (j/n)";
9160 I$=INKEY$: IF I$="" GOTO 9160
9170 IF (I$<>"j" AND I$<>"J") THEN CO%=CO%-1: GOTO 9010
9180 GOTO 1010
9400 'Constraints löschen
9410 CLS
9420 PRINT §261,"";:
     INPUT"Nr. des zu löschenden Constraints : ";N%
9430 IF N%<=MC% GOTO 9480
9440 PRINT §584,"***** Constraint-Nr. zu hoch *****"
9450 PRINT §643,"Bitte eine Taste drücken"
9460 I$=INKEY$: IF I$="" GOTO 9460
9470 GOTO 9410
9480 I$=CN$(1,N%)+"  ===>  "
9490 IF CS%(N%)=(-1) THEN I$=I$+"nicht  "
9500 I$=I$+CN$(2,N%)
9510 PRINT §325,I$
9520 PRINT §584,"Soll dies gelöscht werden ? (j/n)"
9530 I$=INKEY$: IF I$="" GOTO 9530
9540 PRINT §654,"--- Bitte warten ---"
9550 IF (I$<>"j" AND I$<>"J") GOTO 1010
9560 FOR M%=N%+1 TO CO%
9570 CN$(1,M%-1)=CN$(1,M%)
9580 CN$(1,M%-1)=CN$(2,M%)
9590 CS%(M%-1)=CS%(M%)
9600 NEXT M%
9610 CO%=CO%-1
9620 PRINT §654,"                    "
9630 GOTO 1010
10000 'Sichern/Beenden
10010 OPEN "O",1,WN$
10020 PRINT #1,OBJEKTE%: PRINT #1,FRAGEN%: PRINT #1,REGELN%
10030 PRINT #1,CO%: PRINT #1,NQ%
10040 IF OBJEKTE%=0 GOTO 10080
10050 FOR N%=1 TO OBJEKTE%
10060 PRINT #1,OA$(N%): PRINT #1,OSTATUS%(N%): PRINT #1,ODIAG%(N%)
10070 NEXT N%
10080 IF FRAGEN%=0 GOTO 10120
10090 FOR N%=1 TO FRAGEN%
10100 PRINT #1,FTEXT$(N%): PRINT #1,FA$(1,N%):
      PRINT #1,FA$(2,N%): PRINT #1,FSTATUS%(N%):
      PRINT #1,FOBJEKT$(N%)
10110 NEXT N%
10120 IF REGELN%=0 GOTO 10200
10130 FOR N%=1 TO REGELN%
10140 PRINT #1,RSTATUS%(N%)
10150 FOR M%=1 TO 5
10160 PRINT #1,WENN$(M%,N%): PRINT #1,WF%(M%,N%)
10170 NEXT M%
10180 PRINT #1,DANN$(N%): PRINT #1,TYP%(N%)
10190 NEXT N%
10200 IF CO%=0 GOTO 10260
10210 FOR N%=1 TO CO%
10220 PRINT #1,CN$(1,N%)
10230 PRINT #1,CN$(2,N%)
10240 PRINT #1,CS%(N%)
10250 NEXT N%
10260 IF NQ%=0 GOTO 10280
```

```
10270 FOR N%=1 TO NQ%:PRINT #1,QU$(N%):NEXT N%
10280 CLOSE #1
10290 CLS:END
11000 'Wissensbasis zu klein
11010 CLS
11020 PRINT §10,"*****  Wissensbasis zu klein  *****"
11030 PRINT §140,"P r o g r a m m a b r u c h  !":PRINT §584,"";
11040 STOP
12000 'Eingangsfragen
12010 CLS
12020 PRINT §5,"Was möchten Sie ....."
12030 PRINT §133,"1. Eingangsfragen hinzufügen"
12040 PRINT §197,"2. Eingangsfragen auflisten"
12050 PRINT §261,"3. Eingangsfragen löschen"
12060 PRINT §584,"Bitte wählen Sie : ";
12070 I$=INKEY$:IF I$="" GOTO 12070
12080 PRINT I$:I%=VAL(I$)
12090 ON I% GOTO 12210,12610,13010
12120 PRINT §603," ":GOTO 12070
12200 'Eingangsfrage hinzufügen
12210 CLS
12220 NQ%=NQ%+1:IF NQ%>MQ% GOTO 11010
12230 PRINT §133,"Fragenummer oder Bezugsobjekt-Name :"
12240 PRINT §261,"";:INPUT" ===> ";I$
12250 M%=VAL(I$)
12260 IF M%>FRAGEN% THEN
      PRINT §325,"*** Fragenumer zu groß ***":GOTO 12320
12270 IF M%<>0 THEN QU$(NQ%)=FOBJEKT$(M%):GOTO 12350
12280 FOR M%=1 TO FRAGEN%
12290 IF I$=FOBJEKT$(M%) THEN QU$(NQ%)=I$:GOTO 12350
12300 NEXT M%
12310 PRINT §453,"*** ";I$;" nicht vorhanden ***"
12320 PRINT §653,"Bitte eine Taste drücken"
12330 I$=INKEY$:IF I$="" GOTO 12330
12340 NQ%=NQ%-1:GOTO 12210
12350 PRINT §261,FTEXT$(M%)
12360 PRINT §325,FOBJEKT$(M%)
12370 PRINT §584,"Eintrag in Ordnung ? (j/n)";
12380 I$=INKEY$:IF I$="" GOTO 12380
12390 IF (I$<>"j" AND I$<>"J") THEN NQ%=NQ%-1:GOTO 12210
12400 GOTO 1010
12600 'Eingangsfragen auflisten
12610 CLS
12620 IF NQ%<>0 GOTO 12670
12630 PRINT §584,"*** keine Eingangsfragen vorhanden ***"
12640 PRINT §655,"Bitte eine Taste drücken"
12650 I$=INKEY$:IF I$="" GOTO 12650
12660 GOTO 1010
12670 ZEILE%=2
12680 FOR N%=1 TO NQ%
12690 IF ZEILE%=2 THEN PRINT §0,"Eingangsfragen :"
12700 PRINT §64*ZEILE%,N%
12710 FOR M%=1 TO FRAGEN%
12720 IF QU$(N%)=FOBJEKT$(M%) THEN I$=FTEXT$(M%):GOTO 12750
12730 NEXT M%
12740 I$="*** Frage nach "+QU$(N%)+" nicht vorhanden ***"
12750 PRINT §64*ZEILE%+2,I$
12760 ZEILE%=ZEILE%+2
12770 IF ZEILE%<14 GOTO 12820
12780 PRINT §908,"Bitte eine Taste drücken      <E>=Ende"
12790 I$=INKEY$:IF I$="" GOTO 12790
12800 IF (I$="E" OR I$="e") GOTO 1010
```



```
12810 CLS:ZEILE%=2
12820 NEXT N%
12830 PRINT §840,"******** Ende der Liste ********"
12840 PRINT §908,"Bitte eine Taste drücken"
12850 I$=INKEY$:IF I$="" GOTO 12850
12860 GOTO 1010
13000 'Eingangsfragen löschen
13010 CLS
13020 PRINT §261,"Geben Sie bitte die Nr. der zu"
13030 PRINT §325,"";:INPUT"löschenden Eingangsfrage ein : ";N%
13040 IF N%<=NQ% GOTO 13090
13050 PRINT §584,"***** Fragennummer zu groß *****"
13060 PRINT §652,"Bitte eine Taste drücken"
13070 I$=INKEY$:IF I$="" GOTO 13070
13080 GOTO 13010
13090 CLS
13100 FOR M%=1 TO FRAGEN%
13110 IF QU$(N%)=FOBJEKT$(M%) GOTO 13170
13120 NEXT M%
13130 PRINT §261,"*** ";QUESTION$(N%);" nicht vorhanden ***"
13140 PRINT §652,"Bitte eine Taste drücken"
13150 I$=INKEY$:IF I$="" GOTO 13150
13160 GOTO 1010
13170 PRINT §133,"Frage   : ";FTEXT$(M%)
13180 PRINT §261,"Objekt  : ";FOBJEKT$(M%)
13190 PRINT §584,"Soll diese Frage gelöscht werden ? (j/n)"
13200 I$=INKEY$:IF I$="" GOTO 13200
13210 IF (I$<>"j" AND I$<>"J") THEN NQ%=NQ-1:GOTO 13010
13220 N%=M%
13230 IF N%=NQ% GOTO 13270
13240 FOR M%=N%+1 TO NQ%
13250 QU$(M%-1)=QU$(M%)
13260 NEXT M%
13270 NQ%=NQ%-1
13280 GOTO 1010
20000 'Upper $
20010 '-> U$ ->
20020 UU$=""
20030 IF U$="" THEN RETURN
20040 FOR U%=1 TO LEN(U$)
20050 UT$=MID$(U$,U%,1)
20060 IF (ASC(UT$)>96 AND ASC(UT$)<123)
      THEN UT$=CHR$(ASC(UT$)-32)
20070 UU$=UU$+UT$
20080 NEXT U%
20090 U$=UU$
20100 RETURN
```

Auf eines ist beim Aufbau einer Wissensbasis zu besonders zu achten:
Damit beim späteren Ablauf des Expertensystems eine kausale Kette entstehen kann, muß auch in Regeln die sich aufeinander beziehen derselbe Objektname angegeben werden. Dabei ist besonders auf Schreibfehler zu achten !!

3. Die Inferenz(=Diagnose/Begründungs)komponente
Die Inferenzkomponente hat die Aufgabe, aus den eingegebenen Daten für eine Wissensbasis zu Ergebnissen zu kommen. Dies ge- schieht durch eine Rückwärtsverkettung, bei dem versucht wird von einem Zielzustand den Anfangszustand zu erreichen. Alle Diagnosen sind Ziele, die durch Inferenz bestätigt oder widerlegt werden. Das System bewegt sich rückwärts zu den Prämissen der (Diagnose-)Regeln

und versucht festzustellen, ob diese wahr sind. Sind in der Wissensbasis keine Fakten vorhanden, die eine Ent- scheidung über den Zustand (wahr oder falsch) zulassen, so werden weitere Regeln herangezogen, die die Gültigkeit dieser Prämissen bestätigen können.
Beispiel:

```
Fakten : A, B, C, D, E, F (D=Diagnose)
Regeln : 1) wenn A und B dann D
         2) wenn C       dann B
         3) wenn E und F dann A
```

Zur Bestätigung der Diagnose ergibt sich folgende Verkettung:

```
         D
      A     B
E oder F      C
```

Um zu einem Ergebnis=Diagnose zu kommen sind dann die Fragen E, F und C zu stellen.

3.1 rekursive Suche/Programmierung

Im Programm wird durch die rückwärtsverkettete Suche für eine Regel, bei der eine Prämisse unbestimmt ist, eine weitere Regel gesucht um den Wahrheitsgehalt der ersten Prämisse zu ermitteln. Vom Prinzip ist dies ein rekursiver Prozeß. Die Methode, mit der dies im Programm realisiert wird, ist die Verwaltung eines STACK, auf dem alle möglichen Zwischenergebnisse (Diagnosen) abgelegt werden, wenn die Prämisse unbestimmt ist. Wenn ein AKTIONsteil einer Regel schließlich ausgeführt werden kann, wird die oberste Regel wieder vom STACK geholt und die vorher unbekannte Prämisse kann weiter bearbeitet werden.
Beispiel:

```
Regel  : wenn R1         dann D1 (D1=Diagnose)
         wenn F1 und R3 dann R1
         wenn F2 und F3 dann R3
Fragen: F1, F2, F3
---------------------------------------------
Schritt 1: Stack: -    zu untersuchende Diagnose D1
                       unbekannte Prämissen      R1
        2:        -    D1 auf Stack ablegen
        3:        D1   zu untersuchendes Faktum  R1
                       unbekannte Prämissen      F1, R3
        4:        D1   R1 auf Stack ablegen
        5:        D1,  Frage F1 stellen
                  R1   unbekannte Prämisse       R3
        6:        D1,  zu untersuchendes Faktum  R3
                  R1   unbekannte Prämissen      F2, F3
        7:        D1,  Frage F2 stellen
                  R1   unbekannte Prämissen      F3
        8:        D1,  Frage F3 stellen
                  R1   unbekannte Prämissen      -
        9:        D1,  R3 bestimmen (F2,F3) sind bekannt.
                  R1
       10:        D1,  R1 vom Stack holen
                  R1
       11:        D1   R1 bestimmen (F1,R1) sind bekannt
       12:        -    D1 vom Stack holen
       13:        -    D1 bestimmen (R1) ist bekannt
       14:        -    D1 ausgeben und Ende
```

Auf in die Praxis !

3.2 Das Programm

Die Definition des Stack erfolgt durch SRNR%(STACK%) und PZAEHLER%(STACK%)=Prämissenzähler, sowie dem Wert MSTACK% (max. Stackgröße) und der Variablen STACK%, in der die aktuelle

Stacktiefe als Stackpointer enthalten ist. Um das stellen Stellen von überflüssigen Fragen zu vermeiden, werden die Constraints=CO% eingesetzt, die in die Wissensbasis aufgenommen wurden.
Beispiel:
RADIO-SCHWEIGT --> nicht MW-EMPFANG-GUT
MW-EMPFANG-GUT --> nicht RADIO-SCHWEIGT
Mit dem Verfahren, daß Objekte einen Status haben, lassen sich die Auswirkungen der Constraints nur schlecht nachvollziehen. Deshalb wird zusätzlich noch mit einem Traceprotokoll, eine Art Buchführung über die durchgeführten AKTIONEN, gearbeitet. Auf den Aufbau will ich nicht näher eingehen, denn jeder kann sich bei einem Probelauf die Wirkung mit Ausgabe der Diagnosebegründung ansehen.

```
1 'Expertensystem / Inferenzkomponente
2 'Dr. H.-J. Soll       1987
3 'modifiziert für Tandy TRS80 Mod.1
4 'von Egbert Schröer  1990
10 CLS
15 CLEAR 9000
20 MSTACK%=100
25 MTRACER%=200
30 DIM SRNR%(MSTACK%),PZAEHLER%(MSTACK%)
40 PRINT §581,"Name der Wissensbasis : ";:LINEINPUT WN$
50 OPEN "I",1,WN$
60 INPUT #1,OBJEKTE%:INPUT #1,FRAGEN%:INPUT #1,REGELN%
70 INPUT #1,CO%:INPUT #1,Z2%
80 DIM OA$(OBJEKTE%),OSTATUS%(OBJEKTE%),ODIAG%(OBJEKTE%)
90 DIM FTEXT$(FRAGEN%),FA$(2,FRAGEN%),FSTATUS%(FRAGEN%),
   FOBJEKT$(FRAGEN%)
100 DIM RSTATUS%(REGELN%),WENN$(5,REGELN%),WF%(5,REGELN%),
    DANN$(REGELN%),TYP%(REGELN%)
105 DIM CN$(2,CO%),CS%(CO%)
107 DIM TRACER$(MTRACER%)
110 IF OBJEKTE%=0 GOTO 150
120 FOR N%=1 TO OBJEKTE%
130 INPUT#1,OA$(N%):INPUT#1,OSTATUS%(N%):INPUT#1,ODIAG%(N%)
140 NEXT N%
150 IF FRAGEN%=0 GOTO 190
160 FOR N%=1 TO FRAGEN%
170 INPUT #1,FTEXT$(N%):INPUT #1,FA$(1,N%):
    INPUT #1,FA$(2,N%):INPUT #1,FSTATUS%(N%):
    INPUT #1,FOBJEKT$(N%)
180 NEXT N%
190 IF REGELN%=0 GOTO 270
200 FOR N%=1 TO REGELN%
210 INPUT #1,RSTATUS%(N%)
220 FOR M%=1 TO 5
230 INPUT #1,WENN$(M%,N%):INPUT #1,WF%(M%,N%)
240 NEXT M%
250 INPUT #1,DANN$(N%):INPUT #1,TYP%(N%)
260 NEXT N%
270 IF CO%=0 GOTO 279
271 FOR N%=1 TO CO%
272 INPUT#1,CN$(1,N%)
273 INPUT#1,CN$(2,N%)
274 INPUT#1,CS%(N%)
275 NEXT N%
279 CLOSE#1
280 N%=0
290 FOR M%=1 TO REGELN%
300 IF TYP%(M%)=1 THEN N%=N%+1
```

```
310 NEXT N%
320 CLS
330 PRINT §133,FRAGEN%;" Fragen"
340 PRINT §197,REGELN%;" Regeln"
350 PRINT §261,N%;" Diagnosen"
355 PRINT §325,CO%;" Abhängigkeiten"
360 PRINT §657,"Bitte drücken Sie eine Taste"
370 I$=INKEY$:IF I$="" GOTO 370
380 PRINT §657,"Bitte etwas Geduld             "
390 PRINT §5,"Die Wissensbasis besteht aus :"
1000 'Status zurücksetzen
1010 FOR N%=1 TO OBJEKTE%
1020 OSTATUS%(N%)=0
1030 NEXT N%
1040 FOR N%=1 TO FRAGEN%
1050 FSTATUS%(N%)=0
1060 NEXT N%
1070 FOR N%=1 TO REGELN%
1080 RSTATUS%(N%)=0
1090 NEXT N%
1092 FOR N%=1 TO MTRACER%
1093 TRACER$(N%)=""
1094 NEXT N%
1095 TRACER%=1
1100 'alle möglichen Diagnosen auf Stack ablegen
1110 STACK%=0
1120 FOR N%=1 TO REGELN%
1130 IF TYP%(N%)=1 GOSUB 3220:TRACER$="1="+DANN$(N%):
     GOSUB 2810:'Stack ablegen
1140 NEXT N%
2000 'eigentliches Infernzprogramm
2010 IF STACK%=0 GOTO 4010:'Diagnosen ausgeben
2020 IF RSTATUS%(SRNR%(STACK%))=0 GOTO 2505
2030 STACK%=STACK%-1
2040 GOTO 2010
2500 'Regeln bearbeiten
2505 PRINT §517,"Regeln werden bearbeitet !"
2510 IF (PZAEHLER%(STACK%))>5 AND WENN$(1,SRNR%(STACK%))<>
     "- oder -") THEN RSTATUS%(SRNR%(STACK%))=1:
     NM$=DANN$(SRNR%(STACK%)):TRACER$="2="+NM$+"=1="+
     STR$(SRNR%(STACK%)):GOSUB 2810:GOSUB 3430:
     OSTATUS%(N%)=1:GOSUB 5010:GOTO 2030
2520 IF (PZAEHLER%(STACK%))>5 AND WENN$(1,SRNR%(STACK%))=
     "- oder -") THEN RSTATUS%(SRNR%(STACK%))=-1:
     NM$=DANN$(SRNR%(STACK%)):TRACER$="5="+NM$+"=-1="+
     STR$(SRNR%(STACK%)):GOSUB 2810:GOSUB 3430:
     OSTATUS%(N%)=-1:GOTO 2030
2530 NM$=WENN$(PZAEHLER%(STACK%),SRNR%(STACK%))
2540 IF NM$="" THEN PZAEHLER%(STACK%)=6:GOTO 2510
2550 IF NM$="- oder -" THEN PZAEHLER%(STACK%)=PZAEHLER%(STACK%)+1
     :GOTO 2510
2560 GOSUB 3430
2570 IF OSTATUS%(N%)=0 THEN 3020
2580 IF WENN$(1,SRNR%(STACK%))="- oder -" GOTO 2645
2590 'und-Regel
2600 WF%=WF%(PZAEHLER%(STACK%),SRNR%(STACK%))
2605 IF OSTATUS%(N%)*WF%<>1 THEN RSTATUS%(SRNR%(STACK%))=(-1):
     NM$=DANN$(SRNR%(STACK%)):TRACER$="2="+NM$+"=-1="+
     WENN$(PZAEHLER%(STACK%),SRNR%(STACK%))+"="+STR$(WF%):
     GOSUB 2810:GOSUB 3430:OSTATUS%(N%)=(-1):GOTO 2030
2610 PZAEHLER%(STACK%)=PZAEHLER%(STACK%)+1
2620 GOTO 2510
```



```
2630 'oder-Regel
2640 WF%=WF%(PZAEHLER%(STACK%),SRNR%(STACK%))
2645 IF OSTATUS%(N%)*WF%=1THEN RSTATUS%(SRNR%(STACK%))=1:
     NM$=DANN$(SRNR%(STACK%)):TRACER$="5="+NM$+"=1="+
     WENN$(PZAEHLER%(STACK%),SRNR%(STACK%))+"="+STR$(WF%):
     GOSUB 2810:GOSUB 3430:OSTATUS%(N%)=1:GOSUB 5010:GOTO 2030
2650 PZAEHLER%(STACK%)=PZAEHLER%(STACK%)+1
2660 GOTO 2510
2800 'Tracer eintragen
2810 IF TRACER%=MTRACER% THEN TRACER$="9=**** Überlauf ****"
2820 IF TRACER%<=MTRACER% THEN TRACER$(TRACER%)=TRACER$
2830 TRACER%=TRACER%+1
2840 RETURN
3000 'Objektstatus unbekannt
3010 '-> n%  : Nr des unbestimmten Objektes
3020 FOR K%=1 TO REGELN%
3030 IF DANN$(K%)=OA$(N%) THEN M%=N%:N%=K%:GOSUB 3220:N%=M%:
     GOTO 2020
3040 NEXT K%
3050 GOSUB 4410:'Fragen stellen
3060 OSTATUS%(N%)=I%:IF OSTATUS%(N%)=1 THEN GOSUB 5010:'Constraints
testen
3070 GOTO 2580
3200 'Regeln auf stack legen
3210 '-> n% : Nummer der (Diagnose-)Regel
3220 IF STACK%=MSTACK% THEN CLS:
     PRINT"**** Stack-Überlauf ****":END
3230 STACK%=STACK%+1
3240 SRNR%(STACK%)=N%:PZAEHLER%(STACK%)=1
3250 RETURN
3400 'suchen der Objektnummer
3410 '->NAMEN$ : Name des Objektes
3420 '<-N%     : Nummer des Objektes
3430 FOR N%=1 TO OBJEKTE%
3440 IF OA$(N%)=NM$ THEN RETURN
3450 NEXT N%
3460 CLS
3470 PRINT"**** Objekt ";NM$;" nicht vorhanden ****"
3480 END
4000 'Diagnosen ausgeben
4010 ANZAHL%=0
4020 FOR N%=1 TO REGELN%
4030 IF (TYP%(N%)=1 AND RSTATUS%(N%)=1) THEN ANZAHL%=ANZAHL%+1:
     DIAG%=N%
4040 NEXT N%
4050 CLS
4060 PRINT §5,"D I A G N O S E N"
4070 IF ANZAHL%=0 THEN PRINT §256,
     "**** Es konnte keine Diagnose bestätigt werden ****":
     GOTO 4240
4080 IF ANZAHL%=1 THEN PRINT §256,
     "Bestätigte Diagnose : ";DANN$(DIAG%): GOTO 4240
4090 PRINT §128,"Bestätigte Diagnosen : "
4100 ZEILE%=10
4110 FOR N%=1 TO REGELN%
4120 IF (TYP%(N%)<>1 OR RSTATUS%(N%)<>1) GOTO 4210
4130 IF ZEILE%<18 THEN PRINT §ZEILE%*64+5,DANN$(N%):
     ZEILE%=ZEILE%+2:GOTO 4210
4140 PRINT §968,"Zur Fortsetzung eine Taste drücken";
4150 I$=INKEY$:IF I$="" THEN 4150
4160 CLS
4170 PRINT §5,"D I A G N O S E N"
```

```
4180 PRINT §128,"Fortsetzung bestätigte Diagnosen :"
4190 ZEILE%=10
4200 GOTO 4130
4210 NEXT N%
4220 GOTO 4240
4230 ´Programmende
4240 PRINT §968,"Bitte eine Taste drücken";
4250 I$=INKEY$:IF I$="" THEN 4250
4260 CLS
4270 PRINT §5,"Was möchten Sie ....."
4280 PRINT §133,"1. Programm beenden"
4290 PRINT §197,"2. Begründung ausgeben"
4300 PRINT §261,"3. Neuer Durchlauf"
4310 PRINT §584,"Bitte wählen Sie : ";
4320 I$=INKEY$:IF I$="" THEN 4320
4330 PRINT I$:I%=VAL(I$)
4340 ON I% GOTO 4350,6010,280
4350 CLS:PRINT §5,"Programm beendet":PRINT:END
4370 GOTO 4270
4380 ´Dialog-Komponente: Frage stellen
4390 ´-> n% : Nr des unbestimmten Objektes
4400 ´<- i% : Antwortcode
4410 FOR L%=1 TO FRAGEN%
4420 IF OA$(N%)=FOBJEKT$(L%) THEN 4520
4430 NEXT L%
4440 ´keine Frage vorhanden
4450 CLS
4460 PRINT §128,"*** Es fehlt die Frage für ´";OA$(N%);"´ ***"
4465 PRINT §128+64,"Bitte Antwort eingeben"
4470 PRINT §256+64,"Trifft diese Tatsache zu ?"
4480 PRINT §320+64,"Antwort  (Ja,Nein) : ";:INPUT I$
4490 IF (LEFT$(I$,1)="J" OR LEFT$(I$,1)="j") THEN I%=1:
     TRACER$="4="+OA$(N%)+"=1=*** Frage fehlt ***= Trifft
     nicht zu !":GOSUB 2810:CLS:RETURN
4500 I%=(-1):TRACER$="4="+OA$(N%)+"=-1=*** Frage fehlt ***=
     Trifft nicht zu !":GOSUB 2810:CLS:RETURN
4510 ´gefundene Frage nun stellen
4520 CLS
4530 PRINT §128,FTEXT$(L%)
4540 PRINT §256,"1 -  ";FA$(1,L%)
4550 PRINT §384,"2 -  ";FA$(2,L%)
4560 PRINT §968,"Welche Antwort ist korrekt ? (1,2) ";
4570 I$=INKEY$:IF I$="" THEN 4570
4580 PRINT I$
4590 IF I$="1" THEN I%=1:FSTATUS%(L%)=I%:TRACER$="4="+
     FOBJEKT$(L%)+"=1="+FTEXT$(L%)+"="+FA$(1,L%):GOSUB 2810:CLS:
     RETURN
4600 IF I$="2" THEN I%=-1:FSTATUS%(L%)=I%:TRACER$="4="+
     FOBJEKT$(L%)+"=-1"+FTEXT$(L%)+"="+FA$(2,L%):
     GOSUB 2810:CLS:RETURN
4620 FOR I%=1 TO 5:CLS:PRINT §968,"*** Antwort ist falsch ***";:
     NEXT
4630 GOTO 4520
5000 ´ Constraints auswerten
5010 IF  OSTATUS%(N%)<>1 THEN RETURN:´ Trifft normalerweise
     nicht zu
5020 IF CO%=0 THEN RETURN
5030 FOR Z%=1 TO CO%
5040 IF CN$(1,Z%)<>OA$(N%) THEN 5110
5050 FOR ZZ%=1 TO OBJEKTE%
5060 IF CN$(2,Z%)<>OA$(ZZ%) THEN 5100
5070 IF OSTATUS%(ZZ%)<>0 THEN 5110
```



```
5080 OSTATUS%(ZZ%)=CS%(Z%)
5090 TRACER$="3="+CN$(2,Z%)+"="+STR$(OSTATUS%(ZZ%))+
     "="+CN$(1,Z%):GOSUB 2810
5100 NEXT ZZ%
5110 NEXT Z%
5120 RETURN
6000 ´Begründungskomponente
6010 CLS
6020 PRINT §5,"B E G R Ü N D U N G E N"
6030 PRINT §133,"Folgende Ausgaben sind möglich:"
6040 PRINT §197+64,"1. Auflistung aller möglichen Fakten"
6050 PRINT §261+64,"2. Auflistung bestimmter Fakten"
6060 PRINT §325+64,"3. Anzeige einer bestimmten Regel"
6070 PRINT §389+64,"4. Ausdruck aller bestimmter Regeln"
6080 PRINT §453+64,"5. Beenden der Begründungskomponente"
6085 IF TRACER%=MTRACER% THEN PRINT §517+64,"***** Tracer
     Überlauf *****"
6090 PRINT §848,"Bitte wählen Sie : ";
6100 I$=INKEY$:IF I$="" THEN 6100
6110 PRINT I$:I%=VAL(I$)
6120 ON I% GOTO 6210,6310,7010,10010,4260
6170 GOTO 6090
6200 ´Auflistung aller moglichen Fakten
6210 WG%=0:´alle Fakten
6220 GOTO 6340:´Auflistung von Fakten
6300 ´Auflistung der bestimmten Fakten
6310 WG%=1:´nur bestimmte Fakten
6320 GOTO 6340:´Auflistung von Fakten
6330 ´Auflistung von Fakten
6340 CLS
6350 IF WG%=0 THEN PRINT"Auflistung aller möglichen Fakten"
6360 IF WG%=1 THEN PRINT"Auflistung der bestimmten Fakten"
6370 ZEILE%=2
6380 FOR M%=1 TO OBJEKTE%
6381 FOR N%=1 TO TRACER%-1
6382 GOSUB 6730:IF (TYP%<>1 AND OA$(M%)=NM$) THEN 6400
6384 NEXT N%
6385 I$="?":GOTO 6420
6400 IF SU%=1 THEN I$="+"
6410 IF SU%=-1 THEN I$="-"
6420 IF WG%=0 THEN PRINT §ZEILE%*64,M%;".":PRINT §ZEILE%*64+6,I$;
     " ";OA$(M%):ZEILE%=ZEILE%+2
6430 IF WG%=1 AND I$<>"?" THEN PRINT §ZEILE%*64,M%;".":
     PRINT §ZEILE%*64+6,I$;" ";OA$(M%):ZEILE%=ZEILE%+2
6440 IF ZEILE%<15 THEN 6510
6450 PRINT §968,"Bitta eine Taste drücken ! ( E = Abbruch)";
6460 I$=INKEY$:IF I$="" GOTO 6460
6465 IF I$="e" OR I$="E" THEN GOTO 6000
6470 CLS
6480 IF WG%=0 THEN PRINT "Fortsetzung: Alle möglichen Fakten"
6490 IF WG%=1 THEN PRINT "Fortsetzung: Nur bestimmte Fakten"
6500 ZEILE%=2
6510 NEXT M%
6520 PRINT"-------------------------------------------"
6530 PRINT §968,"Ende - bitte eine Taste drücken !";
6540 I$=INKEY$:IF I$="" THEN 6540
6550 GOTO 6010
6700 ´Tracer aufschlüsseln
6710 ´ -> N%
6720 ´ <- TYP$,NM$,SU%,N2$
6730 TRACER$=TRACER$(N%)
6740 TYP%=VAL(LEFT$(TRACER$,1))
```

```
6750 TRACER$=RIGHT$(TRACER$,LEN(TRACER$)-2):IF TRACER$=""
     THEN RETURN
6760 FOR K%=1 TO LEN(TRACER$)
6770 IF MID$(TRACER$,K%,1)="=" GOTO 6800
6780 NEXT K%
6790 NM$=TRACER$:RETURN
6800 NM$=LEFT$(TRACER$,K%-1)
6810 TRACER$=RIGHT$(TRACER$,LEN(TRACER$)-K%)
6820 IF LEFT$(TRACER$,1)="1" THEN
SU%=1:TRACER$=RIGHT$(TRACER$,LEN(TRACER$)-1):GOTO 6850
6830 IF LEFT$(TRACER$,2)="-1" THEN SU%=-1:TRACER$=RIGHT$(TRACER$,
     LEN(TRACER$)-2):GOTO 6850
6840 PRINT §968,"**** Fehler in Tracer ****":STOP
6850 IF LEN(TRACER$)=0 THEN RETURN
6860 TRACER$=RIGHT$(TRACER$,LEN(TRACER$)-1)
6880 FOR K%=1 TO LEN(TRACER$)
6890 IF MID$(TRACER$,K%,1)="=" GOTO 6920
6900 NEXT K%
6910 N2$=TRACER$:RETURN
6920 N2$=LEFT$(TRACER$,K%-1)
6930 N3$=RIGHT$(TRACER$,LEN(TRACER$)-K%)
6950 RETURN
7000 ´Anzeige einer bestimmten Regel
7010 CLS
7020 PRINT §5,"Anzeige einer bestimmten Regel:"
7030 PRINT §645,"Bitte Nr. oder Namen eingeben : ";:LINEINPUT BN$
7050 IF LEN(BN$)=0 GOTO 7030
7060 FOR N%=1 TO LEN(BN$)
7070 I$=MID$(BN$,N%,1)
7080 IF (ASC(I$)<48 OR ASC(I$)>57) AND I$<>" " GOTO 7130:´Name !
7090 NEXT N%
7100 N%=VAL(BN$)
7110 BN$=OA$(N%)
7120 GOTO 7190
7130 FOR M%=1 TO OBJEKTE%
7140 IF OA$(M%)=BN$ GOTO 7190
7150 NEXT M%
7160 PRINT §197+64,"*** Objekt ´";BN$;"´ nicht vorhanden ***"
7180 GOTO 7030
7190 FOR N%=1 TO TRACER%-1
7200 GOSUB 6730
7210 IF (NM$=BN$ AND TYP%<>1) GOTO 7275
7220 NEXT N%
7230 CLS:PRINT §197+64,"Über ";BN$;"liegen keine Informationen
     vor."
7240 PRINT §968,"Bitte eine Taste drücken"
7250 I$=INKEY$:IF INKEY$="" GOTO 7250
7260 GOTO 6010
7270 ´Begründung anzeigen
7275 ON TYP% GOTO 7281,7520,8310,7310,7810
7281 PRINT §968,"**** Fehler in Trace ****":STOP
7300 ´Frage anzeigen
7310 CLS
7320 PRINT §5,BN$;" trifft ";
7330 IF SU%=(-1) PRINT "nicht ";
7340 PRINT "zu"
7350 PRINT §133,"Dies wurde durch die Frage ermittelt :"
7360 PRINT §197+64,N2$
7370 PRINT §325+64,"Die Antwort war :"
7380 PRINT §389+64,N3$
7390 PRINT §968,"Bitte eine Taste drücken"
7400 I$=INKEY$:IF I$="" GOTO 7400
7410 GOTO 6010
7500 ´und-Regel anzeigen
7520 IF SU%=-1 GOTO 7550
7530 IF SU%=1 GOTO 7640
7540 ´Regel trifft nicht zu
7550 PRINT §133,NM$;" ist nicht erfüllt, weil"
7570 I$=N2$
7580 IF VAL(N3$)=(-1) THEN I$="nicht "+I$
7590 PRINT §197+64,I$;"   <=== nicht zutrifft"
7600 PRINT §968,"Bitte eine Taste drücken"
7610 I$=INKEY$:IF I$="" GOTO 7610
7620 GOTO 6010
7630 ´Regel erfüllt
7640 PRINT §133,NM$;" ist erfüllt weil"
7650 PRINT §197,"alle Prämissen erfüllt sind :"
7660 N%=VAL(N2$)
7670 FOR M%=1 TO 5
7680 I$=WENN$(M%,N%)
7690 IF I$="" GOTO 7720
7700 IF VF%(M%,N%)=(-1) THEN I$="nicht "+I$
7710 PRINT §261+64*M%,I$
7720 NEXT M%
7730 PRINT §968,"Bitte eine Taste drücken"
7740 I$=INKEY$:IF I$="" GOTO 7740
7750 GOTO 6010
7800 ´oder-Regel anzeigen
7810 CLS
7820 IF SU%=1 GOTO 7880
7830 IF SU%=-1 GOTO 8110
7870 ´Regel trifft zu
7880 PRINT §133,NM$;" ist erfüllt, weil"
7890 PRINT §197,"mindestens eine Prämisse der"
7900 PRINT §261+7,"oder - Regel"
7910 PRINT §325,"erfüllt ist"
7920 I$=N2$
7930 IF VAL(N3$)=(-1) THEN I$="nicht "+I$
7940 PRINT §453,I$;"   <=== trifft zu"
7950 PRINT §968,"Bitte eine Taste drücken"
7960 I$=INKEY$: IF I$="" GOTO 7960
7970 GOTO 6010
7980 ´Regel unbestimmt
7990 PRINT §968,"Bitte eine Taste drücken"
8000 I$=INKEY$:IF I$="" GOTO 8000
8010 GOTO 6010
8020 RETURN
8030 FOR M%=1 TO 5
8040 IF WENN$(M%,N%)<>"" THEN PRINT §197+M%*64,I$
8050 NEXT M%
8060 PRINT §968,"Bitte eine Taste drücken"
8070 I$=INKEY$:IF I$="" GOTO 8070
8080 RETURN
8090 ´Regel nicht erfüllt
8110 PRINT §133,NM$;" ist nicht erfüllte, weil"
8120 PRINT §197,"keine Prämisse der oder-Regel"
8130 PRINT §261+20,"----------"
8140 PRINT §325,"erfüllt ist"
8150 N%=VAL(N2$)
8160 FOR M%=2 TO 5
8170 I$=WENN$(M%,N%)
8180 IF VF%(M%,N%)=(-1) THEN I$="nicht "+I$
8190 PRINT §261+M%*64,I$;"  <=== trifft nicht zu"
8200 NEXT M%
8210 PRINT §968,"Bitte eine Taste drücken"
8220 I$=INKEY$: IF I$="" GOTO 8220
8230 GOTO 6010
8300 ´Constraint
8310 CLS
8320 PRINT §133,BN$;"trifft ";
8330 IF SU%=(-1) THEN PRINT "nicht ";
8340 PRINT "zu"
8350 PRINT §197,"Dies gilt aufgrund des Constraint"
8360 PRINT §261,N2$;" ===> ";
8370 IF SU%=(-1) THEN PRINT "nicht ";
8380 PRINTNM$
8390 PRINT §968,"Bitte eine Taste drücken"
8400 I$=INKEY$:IF I$="" GOTO 8400
8410 GOTO 6010
10000 ´Begründung ausdrucken
10010 PRINT §968,"**** Druckroutine nicht installiert ****"
10020 GOTO 6010
```


3.3 Ausblick

Die vorgestellte Inferenzkomponente kann noch verbessert werden, wenn, dies ist in dem Editor schon berücksichtigt, durch festgelegte Eingangsfragen Hypothesen erzeugt werden. Die Hypothesen sind eine Auswahl möglicher Diagnosen, die am Anfang der Auswer- tung durch festgelegte Eingangsfragen erzeugt werden. Das würde einer Vorwärtsverkettung entsprechen. Die weitere Diagnose er- folgt dann wieder durch Rückwärtsverkettung.
Die Routine, um Diagnosen etc. auszudrucken, ist noch nicht installiert.
Ich hoffe, daß Programm ist Anregung genug für BASIC-Programmierer

März ´91
Egbert

Literaturhinweis:

1. Soll, KI Expertensysteme programmieren, Franzis Verlag München

Masken generieren mit TRS 80 MOD 1 und GENIE I Computern

Ergänzung zum Artikel von Oliver Volz im Info 32

Im Info 32 stellte uns Oliver die Möglichkeit vor, in Basic bzw. Fortran Masken zu erzeugen. Ich möchte diese Möglichkeiten ergänzen. Unter NewDos/Gdos kann man über den Befehl DUMP,Filename/...:lw,Startadresse,Endadresse,Programmstart nicht nur den Inhalt eines Programmspeicherinhaltes, sondern auch den Bildschirminhalt auf Diskette speichern. Hierauf baut das vorgestellte System auf.
Wenn man sich eine Maske erzeugt hat, sei es mit einem Editor (s. Info 32) oder über Basic Programmzeilen, so speichert man nach Ausgabe auf den Monitor sofort mit dem Befehl DUMP,MASKE/...,3C00H,3FFFH,3C00H den Bildschirminhalt ab.
Im eigentlichen (Basic)Programm kann dann über LOAD"MASKE/..." die Maske wieder eingeladen werden und zu Ein/Ausgabezwecken genutzt werden.

Bei Ausgaben mit der erzeugten Maske läßt sich durch ein und ausschalten des Cursors die Maske relativ einfach erhalten. Will man eine Maske aber als Eingabe- Maske nutzen, ist das aber Schwieriger, vor allen Dingen dann, wenn eine Eingabe über mehrere Zeilen erfolgen soll. Hierzu eine Routine die über PEEK/POKE zunächst Zeichenweise aus einer Maske Eingaben in einen String (A$) übernimmt und die eingelesenen Zeichen durch einen Punkt (.) ersetzt. der Cursor springt dabei immer an die Eingabestellen der Maske und ist dabei abgeschaltet. Dadurch wird die Maske nicht zerstört.

Zunächst wird die für das eigentliche Programm benötigte Maske mit ein paar Basic-Zeilen generiert (oder nach der Methode von Oliver). Die letzte Zeile enthält dann den DUMP Befehl

```
4000 'Fragenmaske
4010 CLS
4020 PRINT $66,"Fragennummer     : ..."
4030 PRINT $194,"Fragetext        : "
4032 PRINT $212,"..........................................."
4035 PRINT $276,"..........................................."
4040 PRINT $322,"Antworttext 1    : "
4042 PRINT $340,"..........................................."
4045 PRINT $404,"..........................................."
4050 PRINT $450,"Antworttext 2    : "
4052 PRINT $468,"..........................................."
4055 PRINT $532,"..........................................."
4060 PRINT $642,"Bezugsobjekt-Name : ........................
.................."
4070 CMD"DUMP MASKE1/DUM:1,3C00H,3FFFH,3C00H"
```

Die gespeicherte Maske läßt sich nun aus dem DOS mit LOAD MASKE1/DUM am Bildschirm darstellen.
In einem Programm kann man die Maske durch CMD"LOAD MASKE1/DUM:1" aufrufen und seine Eingaben machen, vorausgesetzt man achtet auf die Cursorsteuerung, da sonst die Maske zerstört wird.

Ein Programm kann dann wie folgt aussehen:


```
1000 'Hauptprogramm
1010 'Abfragen z.B.:
1020 'Möchten Sie Fragen hinzufügen, auflisten, asudrucken
1020 'oder wie auch immer die "Fragemaske/Eingabemaske
1030 'auch aussieht

2100 'Unterroutine "Fragen hinzufügen"
2200 'Frage hinzufügen
2210 GOSUB 4010                         Unterprogramm Maske laden
2230 FRAGEN%=FRAGEN%+1
2240 PRINT $83,"";:                     Cursor positionieren
     PRINT CHR$(15);:                   Cursor ausschalten
     PRINT FRAGEN%;                     Nummer der Frage
2250 ZEILE%=2:K%=20
2255 ZEILE%=ZEILE%+1:IF ZEILE%=9 THEN ZEILE%=10:K%=22
2257 IF ZEILE%>10 THEN 2290             Ende Eingabemaske
                                        erreicht
2260 PRINT $ZEILE%*64+K%,"";:
     PRINT CHR$(14);                  . Cursor einschalten und
2265 A$=INKEY$:IF A$="" THEN 2265
2270 IF A$=CHR$(13) THEN PRINT CHR$(15);: alles in A$ bis <ENTER>
                                        gedrückt
     GOTO 2255
2275 IF POS(0)=61                       Cursor-Position prüfen;
       THEN PRINT A$;:                  wenn Cursor über
                                        Zeilenende, dann Cursor
                                        aus
         PRINT CHR$(15);:GOTO 2255:     und nächste Zeile
     ELSE PRINT A$;:GOTO 2265
2290 PRINT $910,"Eintrag in Ordnung ? (j/n)";

2300 I$=INKEY$:IF I$="" GOTO 2300
2310 IF (I$<>"j" AND I$<>"J")
     THEN FRAGEN%=FRAGEN%-1:
       GOTO 2210
2320 GOSUB 4110:                        Alles klar, dann Einlesen
     GOTO 1010


2600 'Fragen auflisten
2610 CLS
2620 IF FRAGEN%<>0 GOTO 2670
2630 PRINT $584,"***    keine Fragen vorhanden          ***"
2640 PRINT $652,"*** Bitte eine Taste drücken ***"
2650 I$=INKEY$:IF I$="" GOTO 2650
2660 GOTO 1010
2670 PRINT $83,"":GOSUB 4010            Ausgabemaske auf Monitor
2680 FOR N%=1 TO FRAGEN%                Cursor steht nun auf
                                        Position $83
2690   PRINT $83,N%;                    vorher über Maske
                                        eingegebenen Fragetext
                                        mit den entsprechenden
                                        Begrenzungen und
                                        Positionen ausgeben
2700   PRINT $212,LEFT$(FTEXT$(N%),42);:
       PRINT $212+64,RIGHT$(FTEXT$(N%),42);
2710   PRINT $340,LEFT$(FA$(1,N%),42);:
       PRINT $340+64,RIGHT$(FA$(1,N%),42);
2720   PRINT $468,LEFT$(FA$(2,N%),42);:
       PRINT $468+64,RIGHT$(FA$(2,N%),42);
2730   PRINT $662,FOBJEKT$(N%);:
```

```
         PRINT CHR$(15);                      nach Ausgabe Cursor
                                              auschalten, damit
                                              Ausgabemaske nicht
                                              zerstört wird.
2740     PRINT @913,"Zum Abbruch <E> drücken"
2750     PRINT @968,"Zur Fortsetzung eine andere Taste drücken";
2760  I$=INKEY$:IF I$="" GOTO 2760
2770  IF (I$="E" OR I$="e") GOTO 1010
2780  NEXT N%
2790  GOTO 1010

4000  'Fragenmaske
4010  CMD"load maske1/dum"
4070  RETURN
4100  'Einlesen aus Fragemaske
4110  PRINT CHR$(15);:                        Cursor ausschalten:
                                              Positionen der Zeilen und
                                              Spalten auf Monitor
                                              festlegen und dann
                                              zunächst alles in A$
                                              packen
        I2%=1:
        I3%=2
4120  FOR I1%=0 TO 2
4130    FOR I%=I2% TO I3%:
          FOR K%=0 TO 41
4140        A$(I1%)=A$(I1%)+CHR$(PEEK((2+I%)*64+15380+K%)):
            POKE(2+I%)*64+15380+K%,46:
          NEXTK%,I%:
          I2%=I2%+2:
          I3%=I3%+2:
      NEXT I1%
4145  FOR K%=0 TO 39:
        A$(3)=A$(3)+CHR$(PEEK(10*64+15382+K%)):
        POKE(10*64)+15382+K%,46:
      NEXTK%
```

Jetzt kommt alles in die eigentlichen Variablen. Bleiben wir beim Beispiel Fragen, dann sei FTEXT$ der Fragentext in der Maske und FA$ die Antworttexte. Alle drei sind zu speichern.

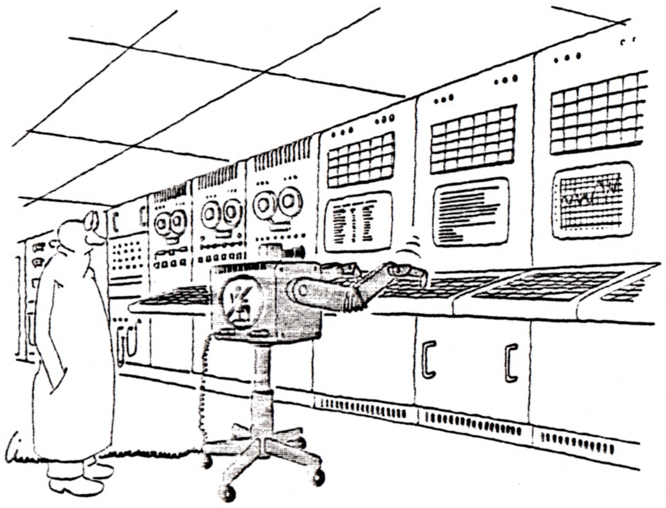
```
4150  FTEXT$(FRAGEN%)=A$(0)
4160  FA$(1,FRAGEN%)=A$(1)
4170  FA$(2,FRAGEN%)=A$(2)
4180  FOBJEKT$(FRAGEN%)=A$(3)
```

Damit alles seine Ordnung hat, wird A$ nun gelöscht.

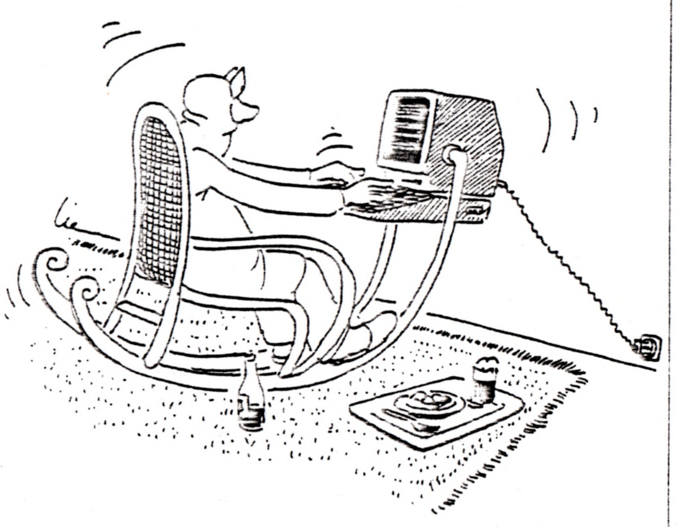
```
4185  FOR I1%=0 TO 3:
        A$(I1%)="":
      NEXT I1%
4190  RETURN
```

Viel Spaß beim Masken generieren
Egbert Schröder

Datenerfassung



Direktzugriff



Hardcopyroutine für SEIKOSHA GP 550A

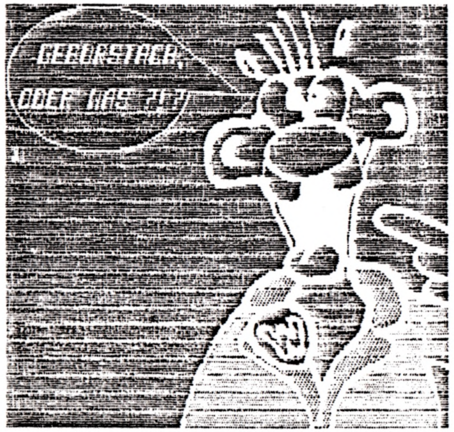
Die meisten Hardcopyroutinen sind für die Drucker geschrieben, die EPSON kompatibel sind. Doch gibt es auch Drucker, die andere Steuerzeichen benötigen um Grafik ausdrucken zu können.

Hierzu gehört auch der SEIKOSHA GP 550 A. Dieser Drucker hat ein gutes Schriftbild und ist auch Grafikfähig.

Das folgende Programm läuft auf einem GENIE I mit HRG 1B und der Software HRGPACK zusammen mit einem SEIKOSHA GP 550A.

Viel Spaß

```
10 DEFINT A-Z
20 LPRINT CHR$(27);"T16";
30 FOR Y=0 TO 191 STEP 8
40       LPRINT CHR$(27);"G383";
50       FOR X=0 TO 383
60                FOR O=0 TO 7
70                         QDOT(X.Y+O.F)
80                         IF F=-1 THEN DE=DE+2AO
90 PRINT"*";
100                NEXT
110                OUT 253.DE
120               DE=0
130      NEXT
140      LPRINT
150 NEXT
```



Das nachfolgende Programm stellt eine Hardcopyroutine für den BROTHER M 1009 und den TRS-80 Model IV/p in Verbindung mit der u-Lab HRG Karte da.

Unter der Treibersoftware steht zwar ein Hardcopybefehl zur Verfügung, jedoch arbeitet dieser nicht zufriedenstellend mit dem M1009 zusammen. Deshalb dieses Programm.

Viel Spaß

```
1 DEFINT A-Z:CLS
50091 SOUND 1,.5
60000 LPRINT CHR$(27);"A";CHR$(8);CHR$(27);"2";
60010 FOR Y=20 TO 239 STEP 8
60011            LPRINT CHR$(27);"L";CHR$(128);CHR$(2);
60020    FOR X=0 TO 639 STEP 1
60030            FOR O=7 TO 0 STEP -1
60040                    FARBE=@POINT(X,Y+O)
60050                    IF FARBE=1 THEN DE=DE+2^(7-O)
60060            NEXT
60070 LPRINT CHR$(DE);:ZAHL=ZAHL+1
60071 DE=0
60080    NEXT
60090 LPRINT CHR$(10);
60091 SOUND 1,.5
60100 NEXT
```

HARDCOPY FUER TRS IV/P UND BROTHER M/1009

(C) 1990/91 FRANK SCHOOF

Schnelle Kreisfunktion unter BASIC30

In den Zeitschriften wie c-t etc. werden bzw. wurden Programme veroeffentlicht, in denen es immer um das besonders schnelle Zeichnen von Kreisen geht. Jedoch sind diese Programme alle nicht die wahren, wenn man die Geschwindigkeit liebt.

Das nachfolgende Programm soll einmal zeigen, dass es noch schneller geht als alles davor da gewesene. Es ist sogar schneller als der @CIRCLE-Befehl des GBASIC.

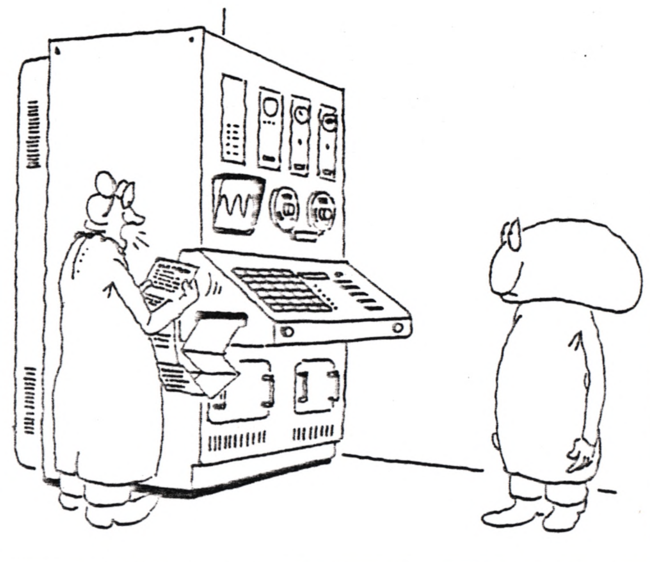
Der Trick: Es wird eine SIN/COS-Tabelle erstellt und diese dann beim Zeichnen ausgelesen.

Das nachfolgende Programm zeigt das Zeichnen eines Kreises mit den Radien 200 und 80. Dazu wird eine Zeit von nur einer(1) Sekunde benoetigt!!!

Wird die Schrittweite in Zeile 30 von 10 erhoeht, so lassen sich auch Vielecke zeichnen.

```
2 REM
3 REM (c) 1991 by Frank Schoof
4 REM
10 DIM XX(100),YY(100)
20 PI=4*ATN(1)
30 FOR W=0 TO 360*PI/180 STEP 10*PI/180
40    X=COS(W)
50    Y=SIN(W)
60    XX(Z)=X
70    YY(Z)=Y
80    Z=Z+1
90 NEXT
100 Z1=Z
110 @ON
120 @CLS
130 R=200:R1=80
140 CLS
150 X1=XX(0)*R+320:Y1=YY(0)*R1+120
160 PRINT TIME$
170 FOR I=0 TO Z-1
180    X=XX(I)*R+320
190    Y=YY(I)*R1+120
200    @LINE(X,Y,X1,Y1,1)
210    X1=X
220    Y1=Y
230 NEXT
240 PRINT TIME$
```


Drucker
»Die Computertherapie sieht vor, ein Weilchen auf Pilzgerichte ganz zu verzichten.«



Grafikdemo fuer uLabs Grafikkarte (Model IV/IVp/III)

Ein kleines Programm zur Auflockerung der alltaeglichen Computertipperei.

```
10 DEFINT A-Z
20 Y=30
30 @CLS
40 CLS
50 @ON
60 FOR X=0 TO 239 STEP RND(20)+3
70    @LINE(Y,X,Y1-Y,120-(X/2),2)
80    Y=Y+10
90 NEXT
100 Y1=RND(600)
110 @CLS
120 Y=RND(40):GOTO 60
```

Frank Schoof

Schönere Programme mit Turbo-Pascal

Über den Sinn oder Unsinn von einheitlichen Schreibweisen läßt sich sicher schön streiten, aber ich finde z.B. Pascal-Programme, die durchgehend klein geschrieben sind, furchtbar unleserlich. Modula bietet hier, neben anderen Verbesserungen, den 'Vorteil', daß der Compiler meckert, wenn man sich nicht haargenau an die Vorschriften hält. Pascal ist in dieser Beziehung zwar viel liberaler, aber wenn man mal ein Programm von Pascal nach Modula umsetzen will, kann man das große Fluchen bekommen. Man kann es dann nämlich trotz der großen Ähnlichkeit der Sprachen fast neu abschreiben, weil der Modula-Compiler sich alle Nas' lang beschwert.

Aus einem Programm, das in einem Listing zur Verbesserung der Lesbarkeit die Schlüsselwörter hervorheben sollte, habe ich mir daher nun eines gebaut, das genau die gewünschte Umsetzung der reservierten Wörter vornimmt. Bei der Schreibweise habe ich mich dabei an die von Modula gehalten, damit, wie schon gesagt, eine Umsetzung möglichst einfach ist und ich mir keine zwei Versionen merken muß. Wenn im Hauptprogramm Include-Files benutzt werden, werden diese automatisch auch mitbearbeitet, Wörter in Kommentaren und Strings werden übergangen.

Wer das Listing nicht abtippen will, kann es von mir gerne gegen Rückporto auf Floppy haben.

```
PROGRAM Markres;

(* Schreibt alle reservierten Woerter in einer einheitlichen *)
(* Schreibweise                                              *)

(*$V-*)        (** Kein Range-Checking bei String-Uebergabe **)

TYPE    Line            = STRING [128];
        Linelength      = 0..128;
        Bez             = STRING [20];

CONST Standardbezeichner : ARRAY [1..102] OF Bez =
        ('Abs','Addr','Append','Arc','ArcTan','Assign',
         'Bdos','BdosHL','Bios','BiosHL','BlockRead','BlockWrite','BOOLEAN','BYTE',
         'Chain','CHAR','Chr','Close','ClrEol','ClrScr','Con','Concat',
         'Copy','Cos','CrtExit','CrtInit',
         'Delete','Delay','DelLine','Dispose','EOF','EOLN','Erase','Execute',
         'Exp','FALSE','FilePos','FileSize','FillChar','Frac','FreeMem',
         'GetMem','GotoXY','HALT','HeapPtr','Hi','HighVideo','Insert',
         'InsLine','Int','INTEGER','IOResult','Kbd',
         'KeyPressed','Length','Ln','Lo','LowVideo','Lst','Mark','MaxAvail','MaxInt',
         'Mem','MemAvail','Move','New','NormVideo','Odd','Ord','ParamCount','ParamStr',
         'Pi','Port','Pos','Pred','Ptr','Random','Randomize','Read','ReadLn','REAL',
         'Release','Rename','Reset','Rewrite','Round',
         'Seek','Sin','SizeOf','Sqr','Sqrt','Str','Succ','Swap','Trm','TRUE','Trunc',
         'UpCase','Usr','Val','Write','WriteLn');
```



```
CONST ResWords : ARRAY [1..46] OF STRING [10] =
        ('ABSOLUTE','AND','ARRAY','BEGIN','CASE','CONST',
         'DIV','DO','DOWNTO','ELSE','END','END.','EXTERNAL',
         'FILE','FOR','FORWARD','FUNCTION','GOTO','IF','IN',
         'INLINE','LABEL','MOD','NIL','NOT','OF','OR','OVERLAY',
         'PACKED','PROCEDURE','PROGRAM','RECORD','REPEAT','SET',
         'SHL','SHR','STRING','TEXT','THEN','TO','TYPE','UNTIL',
         'VAR','WHILE','WITH','XOR');

VAR     InlineFlag,
        KommentarFlag,
        StringFlag      : BOOLEAN;
        EinFn, AusFn,
        IniFn, InlFn2   : Bez;
        EinFp,AusFp,
        IniFp,InlFp2    : TEXT;
        Progzeile       : Line;
        Check           : INTEGER;

FUNCTION Gross ( [redacted] X : Line ) : Line;
VAR I : Linelength;
BEGIN
  FOR I:=1 TO Length (X) DO
    X[I]:=UpCase(X[I]);
  Gross:=X;
END;

FUNCTION Standard( VAR N : Bez ) : BOOLEAN;
VAR B     : Bez;
    I,J,M : INTEGER;
    flag  : BOOLEAN;
BEGIN
  B:= Gross(N); I:=1; J:=102;
  IF NOT(KommentarFlag) AND NOT(StringFlag) THEN
  REPEAT
    M:=(I+J) DIV 2;
    IF B<=Gross(Standardbezeichner[M]) THEN J:=M ELSE I:=M+1
  UNTIL I=J;
  flag:=(Gross(Standardbezeichner[i])=B);
  IF flag THEN N:=Standardbezeichner[I];
  Standard := flag;
END;

FUNCTION Reserviert ( VAR X : Line ) : BOOLEAN;
VAR I,J,M : 1..50;
    H     : BOOLEAN;
    W     : Line;
BEGIN
  H:=FALSE : I:=1; J:=46; W:=Gross(X);
  IF NOT(KommentarFlag) AND NOT(StringFlag) THEN
  BEGIN
    REPEAT
      M:=(I+J) DIV 2;
```

```
      IF W<=Gross(ResWords[M]) THEN J:=M ELSE I:=M+1
    UNTIL I=J;
    H:=(ResWords[I]=W)
  END;
  IF H THEN X:=ResWords[I];
  Reserviert:=H;
END;

PROCEDURE Markiere ( VAR X : Line; VAR Pointer:TEXT );
BEGIN
  Write(Pointer,X);
  LowVideo; Write(X); NormVideo;
END;

PROCEDURE Drucke ( VAR X : Line; VAR Pointer:TEXT );
VAR I : Linelength;
    A : Line;
BEGIN
  A:=''; I:=1;
  WHILE I<Length(X)+1 DO
  BEGIN
    IF X[I] IN ['^','A'..'Z','a'..'z'] THEN
    BEGIN
      A:=X[I]; I:=Succ(I);
      WHILE (X[I] IN ['.','^','0'..'9','A'..'Z','a'..'z','_']) AND (I<=Length(X)) DO
      BEGIN
        A:=A+X[I]; I:=Succ(I)
      END
    END
    ELSE
    BEGIN
      IF A<>'' THEN
        IF Reserviert(A) OR Standard(A) THEN
          Markiere(A,Pointer)
        ELSE
        BEGIN
          Write(Pointer,A); Write(A);
        END;
        A:='';
        Write(Pointer,X[I]); Write(X[I]);
        IF NOT(StringFlag) AND ((X[I]='{') OR ((X[I]='(') AND (X[I+1]='*'))) THEN
          KommentarFlag:=TRUE;

        IF NOT(StringFlag) AND ((X[I]='}') OR ((X[I]='*') AND (X[I+1]=')'))) THEN
          KommentarFlag:=FALSE;

        IF NOT (KommentarFlag) AND (X[I]=Chr(39)) THEN
          StringFlag:=NOT(StringFlag);

        IF KommentarFlag AND (X[I]='$') AND ((X[I+1]='I') OR (X[I+1]='i')) AND
          (X[I+2]<>'+') AND (X[I+2]<>'-') THEN
          InlineFlag:=TRUE;

        I:=Succ(I)
```



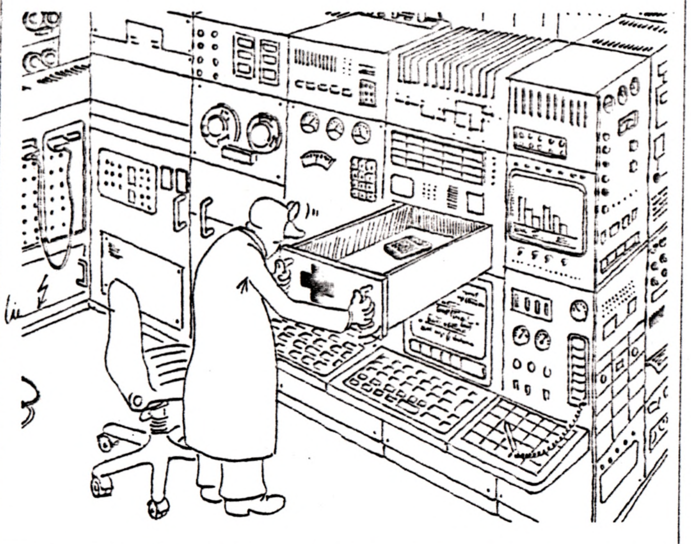
```
      END;
    END;
    IF A<>'' THEN
      IF Reserviert(A) OR Standard(A) THEN
        Markiere(A,Pointer)
      ELSE
      BEGIN
        Write(Pointer,A); Write(A);
        Write(Pointer); Write;
        StringFlag:=FALSE
      END;
      WriteLn(Pointer); WriteLn;
END;

BEGIN
  IF ParamCount=0 THEN
  BEGIN
    ClrScr;
    WriteLn('Umwandlung aller reservierten Woerter in Grossbuchstaben');
    WriteLn;
    Write('Filename: '); ReadLn(EinFn);
  END
  ELSE
    EinFn:=ParamStr(1);
  IF Pos('.',EinFn)=0 THEN
    EinFn:=Concat(EinFn,'.PAS');
  Assign(EinFp,EinFn);
  Reset(EinFp);
  AusFn:='TEMP$$$.TMP';
  Assign(AusFp,AusFn);
  Rewrite(AusFp);
  WHILE NOT EOF(EinFp) DO
  BEGIN
    ReadLn(EinFp,Progzeile);
    Drucke(Progzeile,AusFp);

    IF InlineFlag THEN
    BEGIN
      InlineFlag:=FALSE;
      Check:=Pos('$',Progzeile)+2;
      InlFn:=Copy(Progzeile,Check,Length(Progzeile)-Check);
      writeln(InlFn);
      Assign(InlFp,InlFn); Reset(InlFp);
      Assign(InlFp2,'TEMP2$$.TMP'); Rewrite(InlFp2);
      WriteLn;WriteLn; WriteLn;
      WHILE NOT EOF(InlFp) DO
      BEGIN
        ReadLn(InlFp,Progzeile);
        Drucke(Progzeile,InlFp2);
        IF InlineFlag THEN
        BEGIN
          WriteLn('Verschachtelte INCLUDEs gehen noch nicht');
          HALT
        END
```

```
      END;
      Close(InlFp); Close(InlFp2);
      Erase(InlFp); Rename(InlFp2,InlFn);
    END;

  END;
  Close(EinFp);
  Close(AusFp);
  Erase(EinFp);
  Rename(AusFp,EinFn);
END.
```

Das Programm erhebt keinerlei Anspruch auf besondere Originalität oder guten Programmierstil, aber es erfüllt seinen Zweck. Ich hatte einfach keine Lust, das Originalprogramm vollständig zu überarbeiten.

Alexander Schmid

EDV



Kleiner Terminkalender

Ist es Euch auch schon passiert, daß Ihr das Wochenende durchgehackt und dabei irgendeinen wichtigen Termin oder Ereignis verpaßt habt ? Aus genau diesem Grund habe ich mir ein kleines Programmchen gebastelt, das mich jedesmal beim Booten daran erinnert, was in nächster Zeit alles zu tun ist. Das Eingabefile MEMO.DAT ist sehr einfach folgendermaßen aufgebaut: Datum des Termins, Leerzeichen, beliebiger Text, z.B.

19.05.91 Jahrstreffen des Club-80
22.06.91 Geburtstag von Hugo
24.12.91 Weihnachten

Das Programm gibt immer die Termine der nächsten 8 Wochen aus, die (hoffentlich nicht verpaßten) der letzten Woche werden als letzte Mahnung hervorgehoben dargestellt. Natürlich muß das DOS dafür eine Funktion haben, über die man die aktuelle Zeit erfahren kann, ansonsten bliebe nur die Möglichkeit, das Datum per Hand einzugeben. In meinem Startup-File wird übrigens die Existenz einiger Programme auf der Ramdisk abgefragt, sodaß der Kalender nur beim Kaltstart und nicht bei jedem Reset aufgerufen wird, aber das bleibt jedem selbst überlassen.

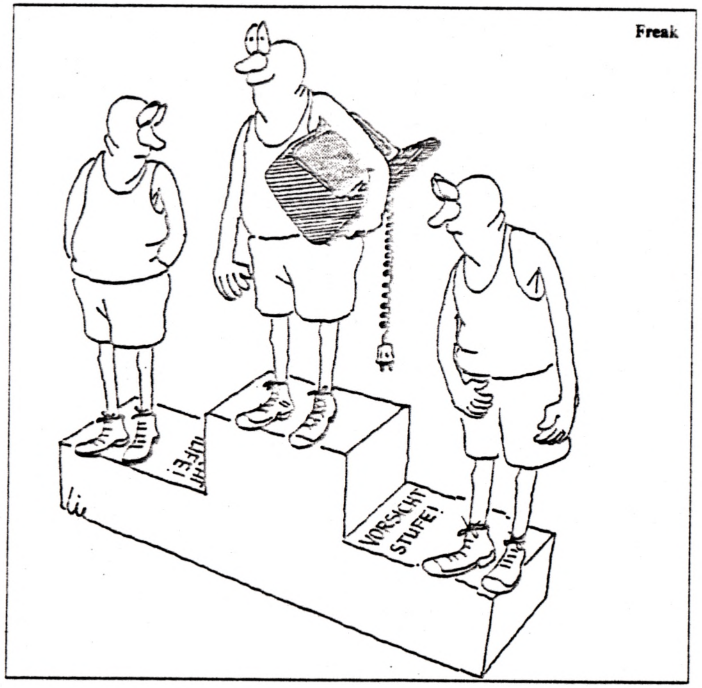
```
PROGRAM Memo;

CONST Tage:ARRAY[0..6] OF STRING[10] = ('Samstag   ','Sonntag   ','Montag    ',
                                         'Dienstag  ','Mittwoch  ','Donnerstag',
                                         'Freitag   ');
CONST days:ARRAY[1..12] OF INTEGER = (31,28,31,30,31,30,31,31,30,31,30,31);

VAR Datum           :INTEGER;
    Tag,Monat,Jahr:INTEGER;
    Filename        :TEXT;
    Zeile           :String[80];
    Ad1,Ad2,Ad3     :String[2];
    Aj1,Aj2,Aj3     :INTEGER;
    AktJul          :INTEGER;
    Dummy           :INTEGER;

PROCEDURE GetDate(VAR datum:INTEGER);
VAR dat  :ARRAY[0..4] OF BYTE;
    s    :INTEGER;
BEGIN
  s:=BDOS(105,ADDR(dat[0]));
  datum:=dat[0]+256*dat[1];
END;
```

```
FUNCTION Leap(jahr:INTEGER):BOOLEAN;
BEGIN
  Leap:=((jahr DIV 4)*4 = jahr);
END;

FUNCTION daysperyear(jahr:INTEGER):INTEGER;
BEGIN
  IF Leap(jahr) THEN
    daysperyear:=366
  ELSE
    daysperyear:=365;
END;

FUNCTION dayspermonth(month,jahr:INTEGER):INTEGER;
BEGIN
  IF Leap(jahr) AND (month=2) THEN
    dayspermonth:=29
  ELSE
    dayspermonth:=days[month];
END;

PROCEDURE BinToJul(julian:INTEGER; VAR tag,monat,jahr:INTEGER);
BEGIN
  tag:=1; monat:=1; jahr:=78;
  WHILE julian>daysperyear(jahr) DO
  BEGIN
    julian:=julian-daysperyear(jahr);
    jahr:=jahr+1
  END;
  WHILE julian>dayspermonth(monat,jahr) DO
  BEGIN
    julian:=julian-dayspermonth(monat,jahr);
    monat:=monat+1;
  END;
  tag:=julian;
END;

PROCEDURE JulToBin(tag,monat,jahr:INTEGER; VAR datum:INTEGER);
VAR   year:INTEGER;
BEGIN
  datum:=0;
  year:=jahr;
  WHILE jahr>78 DO
  BEGIN
    datum:=datum+daysperyear(jahr-1);
    jahr:=jahr-1;
  END;
  WHILE monat>1 DO
  BEGIN
    datum:=datum+dayspermonth(monat-1,year);
    monat:=monat-1;
  END;
  datum:=datum+tag;
END;
```



```
BEGIN
  GetDate(Datum);
  Assign(Filename,'E:MEMO.DAT');
  Reset(Filename);
  WHILE NOT EOF(Filename) DO
  BEGIN
    ReadLn(Filename,Zeile);
    Ad1:=Copy(Zeile,1,2); Ad2:=Copy(Zeile,4,2); Ad3:=Copy(Zeile,7,2);
    Val(Ad1,Aj1,Dummy); Val(Ad2,Aj2,Dummy); Val(Ad3,Aj3,Dummy);
    JulToBin(Aj1,Aj2,Aj3,AktJul);
    IF (AktJul<Datum) AND (Datum-AktJul<8) THEN
    BEGIN
      LowVideo; Write(TageÄAktJul MOD 7Ü,' ',Zeile); NormVideo; WriteLn;
    END;
    IF AktJul=Datum THEN
    BEGIN
      WriteLn;
      LowVideo; Write(TageÄAktJul MOD 7Ü,' ',Zeile); NormVideo; WriteLn;
      WriteLn;
    END;
    IF (AktJul>Datum) AND (AktJul-Datum<61) THEN
      WriteLn(TageÄAktJul MOD 7Ü,' ',Zeile);
  END;
  Close(Filename);
END.
```

Alexander Schmid



Liebe Freunde im Club 80 !

Nachdem mich Fritz Chwolka nun ermutigt hat, es gäbe sicher noch einige "Nichtexperten" im Club, die sich genauso wie ich mit dem eigenen Rechner herumplagen, möchte ich hier schildern, wie ich in letzter Zeit Interesse an Assembler und ein wenig Kenntnis davon gewann.

Es fing damit an, daß ich von mehreren Freunden CP/M-PD-Software bekam, jedoch oft nur die COM-Datei. Vieles hat mich dabei gefreut, nur einige Programme nicht, denen man am Programmnamen nicht ablesen konnte, was sie wohl tun würden. Beim Durchprobieren der Programme ist mir dann immer wieder manch Malheur passiert, wenn sich so ein Programm verselbständigte und nicht mal mehr mit ^C abzubrechen war.

Mein Ärger auf die oft unbekannten Programmierer war groß, zumal sich die Verfasser auch nicht im Programmcode verewigt hatten. Es folgte mein Entschluß : Die Programme wollte ich so ändern, daß ich bei einem zukünftigen Aufruf vor unkontrolliertem Ablauf geschützt war.

Ja, wie mach' ich das nun? Disassembler hatte ich einige, der DASM8080 gefiel mir am Besten. Hier muß ich wohl erwähnen, daß mein Rechner noch eine 8085-CPU hat. Beschreibung zum DASM war zwar nicht zur Hand, aber die Befehle sind so gut wie identisch zum RESOURCE-Disassembler des Ward Christensen (diese DOC-File hatte ich).
Weiterhin brauchte ich einen Texteditor, einen Assembler und einen Linker. Ich habe da Wordstar, M80 und L80 (Macroassembler und Linker) benutzt. Es wird sich nachher zeigen, daß wegen der Besonderheiten des M80 einige Details ins Programm einzufügen sind, die vielleicht bei anderen Assemblern nicht notwendig sind. Ich habe da noch keine Ahnung.

Als Beispiel möchte ich hier das Programm WIPE anführen. Nach dem Aufruf löscht es auf der aktuellen Diskette sämtliche Dateien mit der Erweiterung .PRN, .BAK, .SYM, .REL und .HEX . Dies wird zwar mit einer Textausgabe am Bildschirm bekanntgegeben, doch dann ist es schon zu spät und der Löschvorgang läuft.

Wie ging ich nun vor ? Mit DASM WIPE versucht Dasm verschiedene noch nicht vorhandene WIPE-Files zu lesen und fragt schließlich, ob es WIPE.COM lesen solle. Hier natürlich Y für yes. Nun sollte man schon ein wenig Kenntnis vom Disassembler haben. Lest doch mal das DOC-File. Es folgt das Kommando A100,200. Die Endadresse 200 gab mir Dasm bekannt, der Start bei Adr. 100h ist bei meinem CP/M üblich. Mit diesem A-Kommando werden "mögliche" DB's gekennzeichnet (Speicherplätze mit 8 oder mehr aufeinander folgenden ASCII-Werten). Nun folgt B100,200. Es werden Symbols (Labels) gekennzeichnet. Mit L100 z. B. kann man nun die Assemblerbefehle ansehen und bei verständlichen Stellen einen Kommentar einfügen. Bei meinem Beispiel sind das die Kommentare in Großbuchstaben. Die BDOS-Aufrufe CALL 5 waren unübersehbar.

Hier lachte schon fast mein Herz, denn mir schien fast alles klar. Doch nur sachte. Meine ganze üble Erfahrung will ich hier nicht schildern, nur die Konsequenz daraus. Jetzt rasch die Assemblerfile sichern. SWIPENEU.ALL speichert alle notwendigen WIPE-Dateien und fragt, ob eine .ASM-Datei gewünscht wird. Ja

natürlich in meinem Fall. In die offene Datei wird zuerst mit P100,200 der sogenannte Prolog, der Assemblervorspann geschrieben, danach mit L100,200 die Mnemo-Befehle und dann die Datei mit dem Z-Befehl geschlossen, welcher ein 'END' an den Schluß anfügt. Das erste Drittel war getan.

Probeassemblieren, und schon zeigte der Assembler einen Fehler, das heißt, daß dieser so superschlaue Disassembler Murks gemacht hat (warum ? wer's weiß soll doch mal schreiben). Doch schwierig wars hier nicht. Es fehlte nur das Label L016F vor der Zeichenkette, die ausgegeben werden sollte (WIPE Version ...). Nun tritt der Texteditor in Aktion, mit dem das Verbessern kein Problem ist. Gleichzeitig habe ich dann die anderen Kommentare eingefügt, jetzt alle in Kleinbuchstaben (damit man's besser unterscheiden kann).

Das eingefügte Programmstück will ich hier nicht näher erläutern, es ist deutlich gekennzeichnet. Es fehlen nur noch die nachzutragenden Definitionen für bdos und wboot sowie der neu auszugebende Text (EXIT mit X, ...) und das ende:-Label. Für den M80-Assembler war dann noch die Voranstellung von 'aseg' nötig und eine Titelangabe für den Printerausdruck.

Halt, da war doch der Ärger mit dem unbekannten Programmierer ! Also schnell zu Anfang noch etwas Text in DB's eingefügt. Und wo startet nun das Programm? Mit Text bei Adresse 100h? Nee, so gehts nicht. Also 'ORG 103h', am Ende 'END start', start:-Label beim ersten Befehl und die Sache läuft, denn nun beginnt der Text bei 103, und der Linker hat von 100 bis 102 Platz, den Sprung auf die Startadresse einzutragen. Ob alle Linker das können weiß ich jedoch nicht.

Wieder assemblieren, linken und starten. Da zeigt sich dann, ob alles richtig gemacht wurde. Ich habe recht lange gebraucht, um diese ganze komplexe Angelegenheit soweit verstehen zu können. Bin halt ein Laie in Programmiertechniken. Ich hoffe jedoch, daß sich mehrere Clubfreunde finden, die manches besser wissen und dies hier dann schreiben, so daß ich irgend wann mal kein Anfänger mehr sein werde.

Herzliche Grüße

Uwe Schoberth
P.-Waldus-Str. 14
W-7136 Ötisheim

```
         TITLE 'WIPE  (VERSION 1.1) ,     März 1991 '      ; für M8Ø
         aseg                       ; für M8Ø
         ORG      Ø1Ø3H             ; von 1ØØh auf 1Ø3h (wegen Sprung auf start:)
         db       'WIPE (Version 1.Ø), gefunden in der PD-Software'
         db       'Änderung Uwe Schoberth, W-7136 Ötisheim.'
         db       'Neue Versions-Nr. 1.1, März 91'

                                    ; Das Programm wurde mit DASM rückassembliert
                                    ; und eine Abfrage eingefügt, um einen ungewollten
                                    ; Programmablauf (löschen von Dateien) zu verhindern.
LØØØØ    EQU      ØØH
LØØØ5    EQU      Ø5H
bdos     equ      1ØØØ5             ; neu
wboot    equ      bdos              ; neu

; BEZUGSLAUFWERK ERMITTELN
start:   MVI      C,19H             ; 'start:' ist neu
         CALL     LØØØ5
         ADI      'A'
         STA      LØ1DØ
; ZEICHENKETTE AUSGEBEN
         MVI      C,9
; " WIPE VERSION ...
         LXI      D,LØ16F
         CALL     LØØØ5
                                    ; hier ist die richtige Stelle für eine Abfrage vor
                                    ; Programmablauf
         mvi      c,9h              ; BDOS-Funktion Zeichenkette ausgeben
         lxi      d,text            ; die Zeichenkette steht beim Label 'text:'
         call     bdos
         mvi      c,1h              ; BDOS-Funktion Konsoleneingabe erwarten
         call     bdos
         cpi      'X'               ; ist es ein 'X' ?
         jz       ende              ; ja, Schluß jetzt
         cpi      'x'               ; oder ein 'x' ?
         jz       ende              ; dann auch Schluß
                                    ; Ende der Einfügung

         LXI      H,LØ13B
LØ115:
         MOV      A,M
         INX      H
         STA      LØ157
         CPI      '$'
         JZ       LØØØØ
         MOV      A,M
         INX      H
         STA      LØ158
         MOV      A,M
         INX      H
         STA      LØ159
; DATEI LöSCHEN
LØ129:
         MVI      C,13H
         LXI      D,LØ14E
         PUSH     H
         CALL     LØØØ5
         POP      H
         CPI      ØFFH
         JZ       LØ115
         JMP      LØ129
```



```
LØ13B:
         DB       'PRN'
         DB       'BAK'
         DB       'SYM'
         DB       'REL'
         DB       'HEX'
         DB       'SYM$'
LØ14E:
         NOP
         DB       '????????'
LØ157:
         DB       ' '
LØ158:
         DB       ' '
LØ159:
         DB       ' '
         DS       ØØ17H
LØ16F:
         DB       'Wipe  Version 1.1',ØDH,ØAH       ; neue Versionsnummer
         DB       '-----------------',ØDH,ØAH
         DB       ØAH,'Erasing *.PRN, *.HEX, '
         DB       '*.SYM, *.REL and *.BAK from '
         DB       'Drive '
LØ1DØ:
         DB       ' '                               ; Speicher für akt. Laufwerk
         DB       ':',ØAH,ØDH,'$'
text:    db       Øah,Ødh,'EXIT mit X ,  weiter mit <RET> . $'    ; neu
ende:    jmp      wboot                                            ; neu
         END      start                                            ; neu
```

Hacker

Unter falschem Namen
ALIASes mit ZCPR

Bei Betriebssystemen, die hauptsächlich mit Floppies arbeiten, ist man ja immer bestrebt, nach Möglichkeit Platz zu sparen. Unter CP/M bieten sich Libraries dazu förmlich an, vor allem, da man so auch das Problem mit der begrenzten Anzahl an Files elegant umgehen kann. Ich habe z.B. ein File namens CHAIN.LBR, in dem Turbo-Pascal Chain-Files, und eine zweite Library namens COMMAND.LBR, in der sonstige COM-Files sind. Zwar gibt es Programme, die Files direkt aus solchen Libraries holen und ausführen können, aber die Tipperei ist dabei direkt proportional zum Platzgewinn. Viel Platz -> viel Tipperei, wenig Tipperei -> wenig Platzgewinn. Der Aufruf müßte dann z.B. folgendermaßen aussehen: "-R DDTZ". Sieht ja gar nicht so schlecht aus, aber das geht leider nur, wenn die Library auf dem aktuellen Laufwerk ist. Ansonsten heißt die Kommandozeile z.B. schon "-R -E:COMMAND.LBR DDTZ" und da hört der Spaß langsam auf.

Außer man scheibt sich ein ALIAS, dann kann man nämlich trotzdem einfach nur DDTZ eintippen und der Rest geht von ganz alleine. Zauberei ? Keineswegs ! Da der Command-Processor in diesem Fall natürlich kein DDTZ.COM findet, aktiviert er den ECP, den sog. Enhanced Command Processor und dieser sieht dann nach, ob er ein (Text-)File namens ALIAS.CMD findet. In diesem steht nun hoffentlich, was der Befehl DDTZ in Wirklichkeit bewirken soll und zwar ohne Netz und doppelten Boden so:

```
DDTZ      -R -E:COMMAND.LBR DDTZ $*          (mit Laufwerksangabe)
DDTZ      -R -UTIL:COMMAND.LBR DDTZ $*        (mit Directorynamen)
```

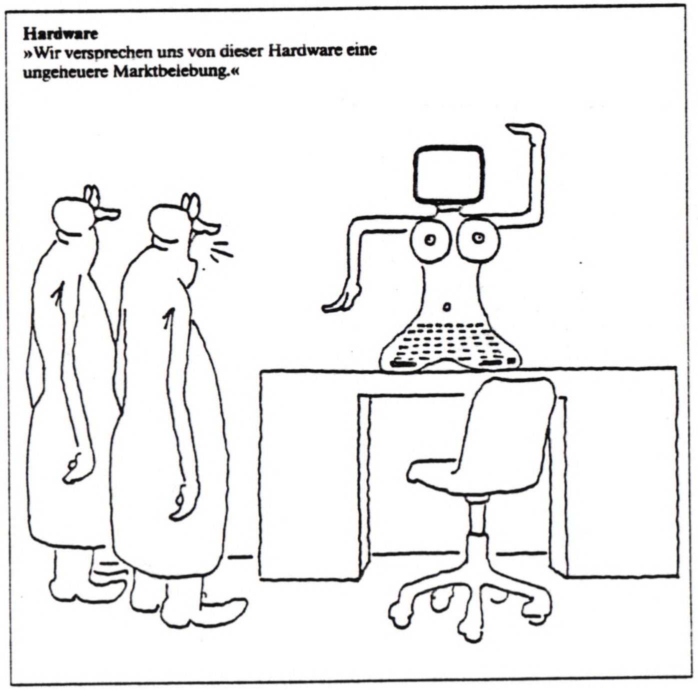
Statt DDTZ wird der gesamte Rattenschwanz rechts davon ausgeführt, sodaß DDTZ.COM aus der Library ausgepackt und mit sämtlichen Parametern aus der ursprünglichen Kommandozeile (das ist das $*) gestartet wird. Entsprechendes geht mit den Chain-Files:

```
DPAR      -C -E:CHAIN.LBR DPAR $*
```

Es ist also völlig egal, ob die Files als 'echte' COM-File vorliegen, oder in irgendeiner Library, womöglich auf einem ganz anderen Laufwerk (Stichwort Harddisk), eingepackt sind. Den ganzen Verwaltungsaufwand nimmt uns ab jetzt das DOS ab. Wenn man will, kann man das ganze natürlich bis zum Abwinken treiben und noch ein paar 'Schmankerl' einbauen.

```
LOOK      IF EX $1;TYPE $1;ELSE;ECHO Schlafmütze;FI
```

Wenn das File vorhanden ist (EXist), wird es aufgelistet, ansonsten bekommt man einen netten Hinweis. Im Grunde ist nur die Phantasie die Grenze. Man könnte z.B. noch prüfen, ob es ein lesbares TXT-File ist und bei COM- und anderen binären Files die Floppy formatieren...



Mit folgendem Feature kann man z.B. DOSlern das Leben leichter machen:

```
ERA,SE=DEL,ETE      ERA $*
```

Durch diesen Trick bewirken ERA, ERASE, DEL und DELETE alle dasselbe, nämlich daß daß der interne ERA-Befehl ausgerufen wird. Die Verwendung ist also keineswegs auf den Start von Programmen beschränkt.

Noch ein Tip am Rande: Wenn man sicher weiß, daß der nächste Befehl für den ECP, d.h. als Alias gedacht ist, kann man mit einem Schrägstrich oder einem Blank vor dem eigentlichen Befehl diesen sofort an den ECP weiterreichen. So spart man die Sucherei nach dem COM-File und es geht schneller.

Übrigens, hatte da mal jemand gesagt, CP/M sei tot ? Ich würde sowas gerne mal unter DOMESDOS sehen (aber bitte nicht mit hunderten von Batch-Files, die für 20 Bytes Text 1K auf der Harddisk belegen). Die ALIASes stehen natürlich alle zusammen im ALIAS.CMD, fein säuberlich untereinander:

```
DDTZ               -R -E:COMMAND.LBR DDTZ $*
DPAR               -C -E:CHAIN.LBR DPAR $*
LOOK               IF EX $1;TYPE $1;ELSE;ECHO Schlafmütze;FI
ERA,SE=DEL,ETE     ERA $*
```

Alexander Schmid

DU3 für CP/M Plus

Für ZCPR33 (CP/M2.2) gibt es ein tolles Disk-Utility mit dem Namen DU3. Damit kann man auf der Floppy alles machen, was man im täglichen Leben so braucht, aber ich will hier gar keine Werbung dafür machen. Gorm Helt-Hansen hat sich nun die Arbeit gemacht, das Programm an Z3PLUS anzupassen, aber die Benutzer von CP/M Plus (ohne Z3) schauen weiterhin in die Röhre, weil das Programm sich seine Steuerzeichen für den Bildschirm aus dem sog. TCAP des Z3PLUS holt, wo diese praktischerweise stehen. Da im normalen CP/M Plus die Terminalfunktionen nirgendwo allgemein zugänglich sind, funktioniert fast nichts mehr und das Ding ist absolut wertlos. Abhilfe: man spiele Z3PLUS und baue sich sein TCAP selber. Die richtigen Steuerzeichen müßte man theoretisch auch per Debugger in das COM-File eintragen können, aber die Initialisierungsroutinen stammen aus einer Library, deren Source-Code ich nicht habe und daher nicht weiß, wo was stehen muß. Aber warum kompliziert, wenn es auch viel einfacher geht.
Der Anfang des ersten Sektors von DU3 sieht so aus:

```
0100   C3 9A 01 5A  33 45 4E 56  01 00 00 49  4E 49 54 49   C..Z3ENV..>INITI
0110   41 4C 20 4D  41 43 52 4F  53 3A 47 30  2C 44 0D 2D   AL MACROS:GO,D.-
0120   44 0D 2B 44  0D 2D 38 44  0D 2B 38 44  0D 54 30 2C   D.+D.-8D.+8D.TO,
```

An der Stelle 109/10A trägt der Lader im Z3PLUS die Adresse des sog. Environments ein, womit die Steuerzeichen für die Initialisierungsroutinen zugänglich sind und genau hier greifen wir ein. An dieser Stelle kann nämlich eine beliebige Adresse stehen, sie wird vom Lader später einfach sang- und klanglos überschrieben, aber eben NUR unter Z3PLUS. Das Programm geht im Urzustand bis 3E7F, sodaß wir unser Pseudo-TCAP einfach bei 3E80 dahinter hängen und diese Adresse natürlich bei 109/10A eintragen. Das alte CP/M Plus verändert diese Adresse nun eben NICHT (wieso auch) und bei der Initialisierung wird unser Kuckucksei verwendet. Der Sektor sieht dann so aus:

```
0100   C3 9A 01 5A  33 45 4E 56  01 80 3E 49  4E 49 54 49   C..Z3ENV..>INITI
0110   41 4C 20 4D  41 43 52 4F  53 3A 47 30  2C 44 0D 2D   AL MACROS:GO,D.-
0120   44 0D 2B 44  0D 2D 38 44  0D 2B 38 44  0D 54 30 2C   D.+D.-8D.+8D.TO,
```

Das benötigte TCAP steht nun genau 128 Bytes hinter dem Anfang des Environments, sodaß wir bei 3E80 etwas Platz lassen müssen. Der Speicher hinter DU3 muß also folgendermaßen gefüllt werden:

```
3E80   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3E90   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3EA0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3EB0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3EC0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3ED0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3EE0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
3EF0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

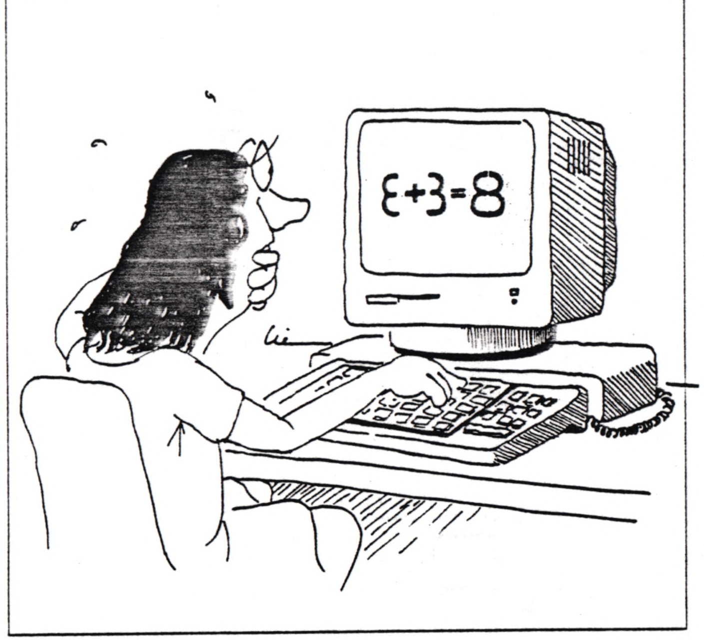


```
3F00   43 50 55 32  38 30 2C 56  54 35 32 20  20 20 20 01   CPU280,VT52    .
3F10   00 00 00 00  00 07 00 1B  45 1B 48 00  1B 59 25 2B   ........E.H..Y%+
3F20   20 25 2B 20  00 1B 4B 00  1B 70 00 1B  71 00 00 00    %+ ..K..p..q...
3F30   1B 4D 00 1B  4C 00 1B 4A  00 00 1B 46  00 1B 47 00   .M..L..J...F..G.
3F40   1B 78 35 00  1B 79 35 00  66 00 63 00  65 00 64 00   .x5..y5.f.c.e.d.
3F50   61 00 60 00  1B 70 20 1B  71 00 69 00  73 00 75 00   a.'..p .q.i.s.u.
3F60   62 00 74 00  76 00 00 00  00 00 00 00  00 00 00 00   b.t.v...........
3F70   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Die Steuerzeichen hier gelten für ein VT52, das TCAP für z.B. ADM3A kann Euch aber jeder Z3PLUS-Besitzer geben.
Die Initialisierungsroutine denkt nun, daß das TCAP hier steht und holt sich die Steuerzeichen wie gewünscht ab. Damit läuft das Programm natürlich (unter CP/M Plus !!!) nur mit diesem einen Terminal. Unter Z3PLUS paßt es sich aber weiterhin automatisch an.

Alexander Schmid

Keyboard



Adreßautomat, die Zweite

Vor Urzeiten stellte ich hier einmal einen Patch in TSCRIPS vor, mit dem häufig gebrauchte Texte, in der Hauptsache natürlich Adressen, rationell in andere Texte, also z. B. Briefköpfe, eingebaut werden können. Das war im Clubinfo Nr. 19. Hier soll nur noch kurz angeschnitten werden, was man dazu wissen und machen muß:

Auf der Arbeitsdiskette befindet sich ein File namens ADRESSEN/LIB. Es enthält eine Bibliothek der wiederkehrenden Texte. Jeder ist genau einen Sektor lang. Seine beiden ersten Zeichen sind das Suchkriterium, etwa die Anfangsbuchstaben des Namens.

Im Haupttext, also z. B. einem Briefformular, ist an der gewünschten Stelle ein Block von genau 254 Bytes Länge mit dem Blocknamen A markiert. In die ersten beiden Stellen des Blocks werden die beiden Suchzeichen eingetippt. Drückt man nun die Tastenkombination @-K, dann sucht TSCRIPS selbsttätig in ADRESSEN/LIB nach der Adresse und überträgt sie in den Block. Nebenbei wird damit auch das aktuelle Datum in einen evtl. vorhanden Block namens D eingetragen. Wer seinerzeit den notwendigen Patch in TSCRIPS nicht vornahm, braucht es allerdings gar nicht erst zu versuchen.

Diese Erleichterung ist besonders dann willkommen, wenn eine öfter gebrauchte Adresse mit kompliziert einzugebenden Steuercodes gespickt ist. Hier ein besonders fieses Beispiel:

Marilyn Monroe
c/o C P G I
Station du Bénin
B. P. 08-0948
Cotonou
R. du Bénin

In der TSCRIPS-Codierung sieht das folgendermaßen aus (Druckercodes des NEC P6, auf IBM-Zeichensatz eingestellt - weitgehend Epson-"Norm"):

```
SW&01&Marilyn Monroe
&1B5300&c&1B54&/&1B5301&o&1B54& &1C4502&C P G I
&1C4501&Station du B&82&nin
B. P. 08-0948&1C4502&
Cotonou&1C4501&
R. du B&82&nin&1C4500&
```

So weit der Schnee von gestern. Die Adressenautomatik versagte leider beim Beschriften von Adreßaufklebern. TSCRIPS hat nämlich den Trieb, nach jedem Text ans Ende der Seite zu gehen. Dabei überspringt der Drucker dann die nächste Million kleiner Aufkleber, bevor der folgende zum Bedrucken an der Walze bereitsteht. Dem ist nun abgeholfen:

Ein kleines Anwenderprogramm macht im Prinzip dasselbe wie der Adressenautomat von TSCRIPS, verkneift sich aber dessen Unarten. Zusätzlich wird die Absenderadresse mit ausgedruckt, so daß man sich auch diese auf dem Umschlag schenken kann. Das kann dann z. B. so aussehen:

<u>Arnulf Sopp, Wakenitzstr. 8, D-2400 Lübeck 1</u>

Herrn
H. Bernhardt
Hafenstr. 7

2305 Heikendorf

Das Programm berücksichtigt auch die TSCRIPS-Sonderzeichen $ und &, allerdings keine weiteren. Mehr davon einzubauen, erschien mir überflüssig, weil mehr in einer Adresse nicht zu erwarten ist. Gegf. kann man sich anders behelfen: Alles, was TSCRIPS mit Hilfe bestimmter reservierter Codes kann, kann der Drucker schon lange. So läßt sich z. B. eine Unterstreichung ohne weitere TSCRIPS-Unterstützung einfügen mit $-&01& (Epson

usw.).

Ein anderes TSCRIPS-Feature wurde jedoch berücksichtigt, das durchaus vorkommen kann: Wenn nämlich der User z. B. zusätzlich die Telefonnummer, den Geburtstag oder etwas anderes über den Adressaten in einer Kommentarzeile (eingeleitet mit ">*") abgelegt hat. Auch andere Kommandos, etwa Randeinstellungen, (eingeleitet mit ">") im Adreßblock wären denkbar. Sie werden nicht mit ausgedruckt.

Ein einzelner Aufkleber ist fix mit der Hand beschriftet; man wird dafür kaum das Schreibpapier aus dem Drucker nehmen, die Endlos-Aufkleber einführen und das Programm starten. Es geht in der Praxis also wohl immer um mehrere Adressen. Anschriften von Gruppen (z. B. Mitgliederliste des Clubs) stehen sinnvollerweise hintereinander in ADRESSEN/LIB. Die Suche nach einem Namenskürzel braucht also nicht jedesmal von vorne in der Datei zu beginnen. Dies berücksichtigt das Programm, indem es nach Erreichen des Dateiendes noch einmal von vorne sucht, dann aber endgültig den Suchfehler anzeigt.

Im Hinblick darauf, daß in der Regel zusammengehörige Gruppen, also auf dem File wohl auch benachbarte Adressen ausgedruckt werden sollen, gibt es auch ein "von-bis"-Feature. Man kann mit der Eingabesyntax AA-ZZ den Ausdruck mehrerer Adressen veranlassen. Diese Bequemlichkeit machte ein zusätzliches Feature erforderlich:

Stellen wir uns vor, die erste eingegebene Adresse gibt es wohl, die zweite aber nicht (Tippfehler). Das Programm findet die erste. Es sucht überhaupt nur nach ihr, um anschließend alle folgenden zu drucken, bis ein Vergleich mit der zweiten Eingabe zeigt, daß schließlich auch sie gefunden wurde. Wurde sie natürlich nicht, weil es sie in unserem Beispiel nicht gibt. Also weiter drucken, bis ADRESSEN/LIB vollständig ausgelutscht ist!

Dem ist mit einem gemeinen Trick abgeholfen: Ein Probelauf ohne Druckerbeteiligung wird durchgeführt, wobei das zweite Kürzel das Suchkriterium ist. Sollte er einen Fehler zur Folge haben, der auch gemeldet wird, wird das Unternehmen abgebrochen und nach einem oder zwei neuen Kürzeln gefragt. Bei erfolgreicher Suche wird der FCB wieder auf den Fileanfang gestellt, denn das erste Kürzel meint natürlich eine davor liegende Adresse. Ein solcher Probelauf mit dem ersten Kürzel erübrigt sich, weil nach ihm tatsächlich gesucht wird.

Das Programm ist ganz und gar auf die Aufkleber abgestimmt, die ich z. Zt. benutze: 88.9 X 35.7 mm. Nötigenfalls braucht für andere Abmessungen nur die Länge des Zeilenvorschubs pro Aufkleber geändert zu werden.

Es versteht sich, daß alle im Programm vorkommenden Druckercodes gegf. dem eigenen Gerät angepaßt werden müssen. Die Kommentare erklären, was da gerade gesteuert wird. Und dann wäre da noch das heikle Problem mit den DOS-Aufrufen. Mein DOS hat mit dem Original nur noch eine entfernte Ähnlichkeit. Sollte etwas nicht klappen, muß der geneigte Leser das dort aufgerufene Feature (s. Kommentare) eben selber programmieren oder gegf. die bei ihm zutreffende Adresse im Original-DOS verwenden.

Arnulf

```
                 ;        Utility zum Beschriften von Adreßaufklebern
                 ;        die Adreßbibliothek ADRESSEN/LIB muß vorhanden sein

5200                      ORG     5200h

                 ;Anschriftenfile ADRESSEN/LIB eröffnen
5200  217053     start    LD      HL.filenam      ;Dateiname
5203  118453              LD      DE.fcb          ;FCB für ADRESSEN/LIB
5206  CD1C44              CALL    441ch           ;Namen in den FCB übertragen
5209  210054              LD      HL.secbuff      ;Sektorpuffer
520C  45                  LD      B,L             ;logische Recordlänge = 256 (L = 00h)
520D  CD2444              CALL    4424h           ;Datei eröffnen
5210  206D                JR      NZ.disperr      ;Ende. falls ADRESSEN/LIB nicht existiert

                 ;Drucker auf Bereitschaft testen. Font des Druckers einstellen
```

```
5212  3EFE      00016          LD     A,0feh         ;Requestcode für SYS28/SYS
5214  0E05      00017          LD     C,05h          ;Zeiger auf Druckertest-Routine in SYS28
5216  CD0244    00018          CALL   4402h          ;dort RST 28h: SYS28/SYS aufrufen
5219  211E53    00019          LD     HL,prtinit     ;Initialisierungsdaten des Druckers
521C  CD6A44    00020          CALL   446ah          ;ausgeben
                00021
                00022 ;eine Adresse oder 'von-bis' mit Kürzel/n erfragen
521F  210153    00023 newname  LD     HL,getaddr     ;Frage nach dem/den Adreßkürzel/n
5222  CD6744    00024          CALL   4467h          ;anzeigen
5225  217E53    00025          LD     HL,inpbuff     ;Eingabepuffer
5228  0605      00026          LD     B,05h          ;max. 5 Zeichen für das/die Namenskürzel
522A  CDD905    00027          CALL   05d9h          ;Zeichen einlesen
522D  DA2D40    00028          JP     C,402dh        ;ENDE, falls mit BREAK abgebrochen
5230  3A8053    00029          LD     A,(inpbuff+2)  ;dort CR oder 'bis'-Zeichen
5233  FE2D      00030          CP     '-'            ;für 'bis'? (zwei Kürzel 'von-bis')
5235  2A7E53    00031          LD     HL,(inpbuff)   ;1. Namenskürzel
5238  2022      00032          JR     NZ,oneaddr     ;falls nur ein Kürzel
                00033
                00034 ;es wurden zwei Kürzel für 'von-bis' eingegeben
523A  E5        00035          PUSH   HL             ;1. Kürzel retten
523B  2A8153    00036          LD     HL,(inpbuff+3) ;2. Kürzel
523E  227E53    00037          LD     (inpbuff),HL   ;als 1. patchen
5241  3EC9      00038          LD     A,0c9h         ;Opcode RET
5243  329752    00039          LD     (found2),A     ;Druckroutine unwirksam machen
5246  3E01      00040          LD     A,01h          ;Flag: nur 1 Durchlauf der Suche, weil am
5248  08        00041          EX     AF,AF'         ;Fileanfang begonnen
5249  CD6752    00042          CALL   readsec        ;Programm aufrufen (Probelauf)
524C  E1        00043          POP    HL             ;1. Kürzel
524D  227E53    00044          LD     (inpbuff),HL   ;restaurieren
5250  3E21      00045          LD     A,21h          ;Opcode LD HL,nn
5252  329752    00046          LD     (found2),A     ;Druckroutine restaurieren
5255  20C8      00047          JR     NZ,newname     ;falls Adresse nicht vorhanden
5257  CD3F44    00048          CALL   443fh          ;FCB auf Fileanfang zurückstellen
525A  1803      00049          JR     alladdr        ;dort weiter
                00050
                00051 ;eine einzeine Adresse suchen und ausdrucken
525C  228153    00052 oneaddr  LD     (inpbuff+3),HL ;1. Kürzel als 2. patchen (wg. Vergleich)
                00053
                00054 ;gemeinsam weiter bei 1 und 2 Kürzeln
525F  CD6452    00055 alladdr  CALL   prtaddr        ;Such- und Druckprogramm aufrufen
5262  18BB      00056          JR     newname        ;von vorne weiter
                00057
                00058 ;eine Adresse gem. dem Kürzel finden und ausdrucken
5264  3E02      00059 prtaddr  LD     A,02h          ;Flag: 1. Durchgang bei der Suche
5266  08        00060          EX     AF,AF'         ;retten
5267  CDD652    00061 readsec  CALL   loadsec        ;Sektor laden
526A  200D      00062          JR     NZ,notfnd      ;falls Ende der Datei oder anderer Fehler
                00063
                00064 ;im Adressenfile nach demselben Kürzel suchen
526C  2A0054    00065          LD     HL,(secbuff)   ;Namenskürzel in ADRESSEN/LIB
526F  ED4B7E53  00066          LD     BC,(inpbuff)   ;1. eingegebenes Namenskürzel
5273  ED42      00067          SBC    HL,BC          ;bei Gleichheit ist das Z-Flag gesetzt
5275  2820      00068          JR     Z,found2       ;falls die richtige Anschrift gefunden
5277  18EE      00069          JR     readsec        ;bis zum 1. gefunden Kürzel
                00070
                00071 ;ADRESSEN/LIB ist überschritten, diese Anschrift gibt es evtl. nicht
5279  FE1C      00072 notfnd   CP     1ch            ;Fehler 'Ende der Datei angetroffen'?
527B  2805      00073          JR     Z,seekerr      ;falls ja
527D  FE1D      00074          CP     1dh            ;Fehler 'Hinter Ende der Datei'?
527F  C20944    00075 disperr  JP     NZ,4409h       ;ENDE, falls anderer Fehler
5282  08        00076 seekerr  EX     AF,AF'         ;Durchgangszähler
5283  3D        00077          DEC    A              ;schon zweiter Durchgang?
5284  2806      00078          JR     Z,adrfail      ;falls ja: Adresse nicht vorhanden
5286  08        00079          EX     AF,AF'         ;falls nein: Zähler wieder retten
```



```
5287  CD3F44    00080          CALL   443fh          ;FCB auf Anfang des Files stellen
528A  18DB      00081          JR     readsec        ;ab Fileanfang weiter versuchen
                00082
                00083 ;Fehlerroutine: Adresse nicht vorhanden
528C  215F53    00084 adrfail  LD     HL,errmssg     ;Fehlermeldung
528F  CD6744    00085          CALL   4467h          ;anzeigen
5292  B7        00086          OR     A              ;NZ-Flag setzen (Fehlerbedingung)
5293  C9        00087          RET
                00088
                00089 ;für 'von-bis'-Ausdruck fortlaufende Sektoren laden
5294  CDD652    00090 found1   CALL   loadsec        ;nächsten Sektor laden
                00091
                00092 ;1. Anschrift gefunden: ausdrucken
5297  212453    00093 found2   LD     HL,sender      ;Absenderadresse
529A  CD6A44    00094          CALL   446ah          ;ausdrucken
529D  210254    00095          LD     HL,secbuff+2   ;Beginn der eigentlichen Empfängeradresse
52A0  7E        00096 prtloop  LD     A,(HL)         ;Zeichen aus dem String
52A1  23        00097          INC    HL             ;auf nächstes Zeichen
52A2  FE24      00098          CP     '$'            ;ESC in TSCRIPS-Codierung?
52A4  2002      00099          JR     NZ,noesc       ;falls nein
52A6  3E1B      00100          LD     A,ESC          ;sonst ASCII-ESC daraus machen
52A8  FE26      00101 noesc    CP     '&'            ;TSCRIPS-Flag für Hex-String?
52AA  2833      00102          JR     Z,hexloop      ;falls ja
                00103
                00104 ;normale Zeichen ausdrucken
52AC  FE8D      00105          CP     CR+80h         ;Zeilenende? (TSCRIPS setzt Bit 7 bei CR)
52AE  200A      00106          JR     NZ,nocr        ;falls nein
52B0  7E        00107          LD     A,(HL)         ;Zeichen an der nächsten Stelle laden
52B1  FE3E      00108          CP     '>'            ;Beginn einer Befehlszeile?
52B3  3E8D      00109          LD     A,CR+80h       ;Zeichen restaurieren
52B5  2003      00110          JR     NZ,nocr        ;falls nein
52B7  44        00111          LD     B,H            ;Länge der Suchstrecke groß genug
52B8  EDB1      00112          CPIR                  ;Zeiger hinter abschließenden CR stellen
52BA  E67F      00113 nocr     AND    7fh            ;Bit 7 = 0 wg. TSCRIPS-Eigentümlichkeiten
52BC  CDB405    00114          CALL   05b4h          ;Zeichen ausdrucken
52BF  7C        00115          LD     A,H            ;MSB der Pufferadresse
52C0  FE54      00116          CP     secbuff/100h   ;Sektorende überschritten?
52C2  28DC      00117          JR     Z,prtloop      ;falls noch nicht
                00118
                00119 ;Adresse ist gedruckt; Walze auf nächsten Aufkleber stellen
52C4  215B53    00120          LD     HL,nextlbl     ;Zeilenschaltungen zum nächsten Aufkleber
52C7  CD6A44    00121          CALL   446ah          ;ausdrucken
52CA  2A0054    00122          LD     HL,(secbuff)   ;Namenskürzel dieses Sektors
52CD  ED4B8153  00123          LD     BC,(inpbuff+3) ;2. Namenskürzel
52D1  ED42      00124          SBC    HL,BC          ;gleich? (2. Kürzel erreicht, Adr. gedr.)
52D3  C8        00125          RET    Z              ;fertig, falls ja
52D4  18BE      00126          JR     found1         ;sonst weiter drucken
                00127
                00128 ;einen Sektor in den Puffer laden
52D6  210054    00129 loadsec  LD     HL,secbuff     ;Sektorpuffer
52D9  118453    00130          LD     DE,fcb         ;FCB-Adresse
52DC  C33644    00131          JP     4436h          ;Sektor laden und zurück
                00132
                00133 ;Bearbeitung von Hex-Codes zwischen &-Zeichen
52DF  CDF652    00134 hexloop  CALL   gethex         ;1. Hex-Ziffer einlesen
52E2  07        00135          RLCA                  ;unteres ins obere Nibble rotieren
52E3  07        00136          RLCA
52E4  07        00137          RLCA
52E5  07        00138          RLCA
52E6  4F        00139          LD     C,A            ;puffern
52E7  CDF652    00140          CALL   gethex         ;2. Hex-Ziffer einlesen
52EA  B1        00141          OR     C              ;jetzt eine Hex-Zahl komplett im Akku
52EB  CDB405    00142          CALL   05b4h          ;Zeichen ausdrucken
52EE  7E        00143          LD     A,(HL)         ;nächstes Zeichen
```

```
52EF  FE26     00144          CP      '&'           ;abschließendes &-Zeichen?
52F1  20EC     00145          JR      NZ.hexloop    ;falls es in Hex weitergeht
52F3  23       00146          INC     HL            ;&-Zeichen überspringen
52F4  18AA     00147          JR      prtloop       ;weiter drucken
               00148
               00149 ;eine Hexziffer einlesen
52F6  7E       00150 gethex   LD      A.(HL!        ;Hex-Zeichen laden
52F7  23       00151          INC     HL            ;auf nächstes Zeichen
52F8  FE41     00152          CP      'A'           ;höher als 0-9?
52FA  3802     00153          JR      C,ciphok      ;falls nein
52FC  D607     00154          SUB     07h           ;Hex an Dez angleichen
52FE  E60F     00155 ciphok   AND     0fh           ;ASCII -) binär
5300  C9       00156          RET
               00157
               00158 ;verschiedene Codes. Texte und Puffer
0000           00159 NUL      EQU     00h     ;null
0003           00160 ETX      EQU     03h     ;end of text
0007           00161 BEL      EQU     07h     ;beep
000D           00162 CR       EQU     0dh     ;carriage return
001B           00163 ESC      EQU     1bh     ;escape
5301  4B       00164 getaddr  DM      'Kürzel für die Adresse(n)? '.BEL.CR    ;Aufforderung
531E  1B       00165 prtinit  DB      ESC.'§'.ESC.'p',01h.ETX :Druckerinitialisierungen:
               00166                                          :Nahezu-Reset. Proportionalschr.
5324  1B       00167 sender   DB      ESC.'-'.01h.ESC.'S',01h :Unterstreichung, Subscript
532A  41       00168          DM      'Arnulf Sopp. Wakenitzstr. 3. D-2400 Lübeck 1'  :Absender
5356  1B       00169          DB      ESC.'T',ESC.'-',NUL     :zurück auf Normalschrift
535B  1B       00170 nextlbl  DB      ESC,'J',0fh.CR          ;15/180 Zoll Vorschub. CR
535F  41       00171 errmssg  DM      'Adresse fehlt in '     ;Fehlertext
5370  41       00172 filenam  DM      'ADRESSEN/LIB'.BEL.CR   ;Filename
0006           00173 inpbuff  DS      06h                     ;Puffer für Tastatureingaben
0020           00174 fcb      DS      20h                     ;Puffer für den FCB
5400           00175 secbuff  EQU     $+100h&0ff00h           ;Sektorpuffer an gerader Adresse
               00176
5200           00177          END     start

00000 Fehler

BEL     0007   CR      000D   ESC     001B   ETX     0003   NUL     0000   adrfail 523C
alladdr 525F   ciphok  52FE   disperr 527F   errmssg 535F   fcb     5384   filenam 5370
found1  5294   found2  5297   getaddr 5301   gethex  52F6   hexloop 52DF   inpbuff 537E
loadsec 52D6   newname 521F   nextlbl 535B   nocr    52BA   noesc   52A8   notfnd  5279
oneaddr 525C   prtaddr 5264   prtinit 531E   prtloop 52A0   readsec 5267   secbuff 5400
seekerr 5232   sender  5324   start   5200
```


Copyright

a) Hinweis für Raubkopierer („Kopier – aber richtig").
b) Das Urheberrecht verhindert durch das Copyright, daß sich feige Programmierer in der Anonymität verstecken können.

CPU

Gerichtsstand des Rechners, in dem sämtliche Entscheidungen (meist zu Ungunsten des Anwenders) gefällt werden. Der Chip sieht aus wie ein Käfer mit 32 Beinen. Bewegt er sich, dann ist es sogar einer.

D

Datei

a) Jede halbwegs locker zusammengehörende Anzahl von Bytes auf einem Speichermedium, gleichgültig, ob sie überhaupt einen Zweck erfüllen, und welcher dies nun gerade sein könnte.
b) Datenbankinhalt, jene auf immer und ewig verschwundenen, vor langer Zeit eingetippten Angaben über irgend etwas, was man nicht vergessen wollte.

Datenbus

Vielpolige, gelegentlich auch ganzvielpolige Leitung in der CPU zum Datentransport. Immer viel zu langsam und immer mit zu geringer Kapazität. Deshalb in Anlehnung an öffentliche Verkehrssysteme „Bus" genannt.

Datenerfassung

Neue Volkssucht, die, ausgehend von öffentlichen Institutionen, auf alle Computerbesitzer übergriff. Datenerfasser, die mit Lust bei der Sache sind, haben in ihrer Datenbank nicht nur die Maße und Haarfarben aller Bunnies (Retusche-Häschen) des Jahrgangs 1967–1989, sondern können diese auch nach der metrisch spezifizierten Größe ordnen und ausdrucken (siehe auch Volkszählung).

Datenreise

Nächtliche DFÜ-Verbindung mit einer Datenbank. Für Hacker ist dies mit Rauschgiftkonsum gleichzusetzen. Der übernächtigte, bleiche User sitzt mit geweiteten Pupillen und eingefrorenem Grinsen vor seinem Bildschirm. Gelegentlich wird die Datenreise zum Horrortrip, etwa weil das Password unüberwindlich ist. Um die Kosten ihrer Sucht, die Telefonrechnung, zu bezahlen, schrecken Hacker auch vor kriminalistischen Handlungen nicht zurück. Sie verfassen Handbücher oder Programme.

Defender

a) Computerspiel, bei dem es darum geht, die Erde gegen angreifende Weltraummonster und glitschige Aliens zu verteidigen.
b) User, dessen Lebensinhalt darin besteht, den Computer gegen eifersüchtige Ehefrauen und neidische Kinder zu verteidigen.

Directory

Verzeichnis auf einer Diskette oder Festplatte. Dabei verhalten sich die Dateien oft wie Socken oder Unterhosen in einer Kommodenschublade; sie verschwinden auf Nimmerwiedersehen.

Direktzugriff

a) Unangenehme, tätliche „Anmache" eines Vorstadt-Casanovas. Führt gelegentlich zur Ohrfeige.
b) Direkter Zugriff eines Programms auf Daten der Festplatte. Führt gelegentlich zum Headcrash (siehe dort).

Liebe Computer-Freunde,

mit diesem kleinen Artikel habe ich endlich meine guten Vorsätze etwas zur Clubzeitschrift beizutragen in die Tat umgesetzt. Ich werde auch zukünftig (hoffentlich regelmäßig) für die Clubfreunde die ein Modell 4 oder 4P besitzen über PD - Software, "Neuigkeiten" usw. berichten. Da es für mich nur ein gutes Betriebssystem gibt (LSDOS 6.3) werden sich meine Tips und Patches in der Regel nur auf dieses Betriebssystem beschränken. Ein kleiner Hinweis: Alle Tips und Patches sind nicht auf meinem Mist gewachsen, sondern stammen in der Regel aus der Zeitschrift Computer News 80 die ich aus Amerika beziehe.

Selbstverständlich habe ich, soweit das möglich ist, alle Patches vorher getestet. Es wird sich aber nie ausschließen lassen, daß es zu Problemen kommen kann. Also Vorsicht bei allen Änderungen des Betriebssystems.

So nun soll es genug sein mit der langen Vorrede, jetzt endlich zu den versprochenen Informationen.

Im letzten Jahr habt ihr zusätzlich zum Clubinfo eine Aufstellung über PD - Software (Mod. 4) erhalten. Ein Teil dieser Programme ist jetzt vom Club gekauft worden, und steht den Mitgliedern auf Wunsch zur Verfügung. Diese Software wird von mir verwaltet und evtl. bei Zustimmung durch den Vorstand um weitere interessante Programme erweitert.
Zur Zeit sind folgende Disketten im Bestand:

MOD4UTL Nr.: 3 - 5, 9 - 12, 14, 16 - 18, 20 - 31, 33 - 36,
38 - 40, 43 - 48, 53 - 54, 60

MOD4COM Nr.: 26

Eine kurze Beschreibung der Programme ist aus dem Verzeichnis ersichtlich. Ich habe auch vor einige Programme in der Clubzeitschrift vorzustellen, möchte aber jetzt bereits an jeden die Bitte richten selbst nach Erhalt der Software einen kleinen Erfahrungsbericht bei Jens abzuliefern.

Da die Programme vom Club gekauft wurden entstehen also nur Kosten Diskette/Postgebühr. Wenn mir Leerdisketten zugesandt werden dann bitte ordentlich verpackt.

Bei dieser Gelegenheit etwas zum aktuellen Stand der Technik beim Modell 4 hinsichtlich Speichererweiterungen. Die Firma Anitek (USA) bietet ihre MegaMem - Boards mit bis zu 8 MByte an. Es gibt verschiedene Boards von 1 bis 8 MByte (130 bis 200 US $/ohne Speicherb.) die sich mit SIP Bausteinen bestücken lassen. Der Einbau soll angeblich ohne löten möglich sein. Das Board hat nach Angaben von Anitek eine Größe von

3.5 x 5.4 inch. Ein z.B. 6 MByte Board kann auch mit lediglich 1, 2 usw. MByte bestückt werden, so daß einem späteren "Vollausbau" nichts im Wege steht. Nachdem man das Board mit Speicherbausteinen bestückt hat muß lediglich die CPU entfernt werden, das Board in den Sockel der CPU und die CPU in den Sockel auf dem Board gesteckt werden. Hört sich doch wirklich gut an.

Da es bisher lediglich ein Programm gab (LeScript v. Anitek) welches Speichererweiterungen über 128 K nutzt, konnte mann im Normalfall die Speichererweiterung lediglich als schnelle Ramdisk nutzen. Jetzt gibt es aber das Programm HYPERSVC, ebenfalls von Anitek, welches völlig neue Aspekte eröffnet. Dieses Programm patcht das Betriebssystem (LS/TRS-DOS 6.xx) beim Start in der Art und Weise, das auch andere Programme wie z.B. PRO-WAM, Double-Douty (neue Vers.), Little Brother Data Base, Busy Calc und Busy Calc Extra, Fast Term (Comm. Progr.), M-Ras Assembler, M-C Compiler und PRO-ENH Basic Compiler die Fähigkeit verleiht die Speichererweiterung über 128K zu nutzen. Das ganze soll rd. 20 US $ kosten. Ob dieses Programm auch mit dem an die deutsche Tastatur angepassten Betriebssystem funktioniert, werde ich noch klären.

Anitek bezeichnet das ganze als "intelligent patch". Unabhängig vom benutzten Betriebssystem sucht sich das Programm den Bereich der geändert werden soll selbst.

Zum Schluß ein kleiner Tip für alle die in Basic programmieren. Statt des Befehles SYSTEM reicht es völlig aus das Ausrufungszeichen ! einzugeben. Hierzu ein Beispiel:

```
! "DIR 1" entspricht SYSTEM "DIR 1"
```

Da mir nicht genau bekannt ist wer alles ein Mod. 4/4P besitzt, bin ich auf eure Reaktionen sehr gespannt. Wenn ich etwas Resonanz finde werde ich euch in Kürze z.B. Informationen über mehrere wirklich gute "Patches" für LSDOS 6.3 und andere Programme liefern.

Jörg Brans

Hallo LeScript User,

bestimmt fahren viele von euch das Textverarbeitungsprogramm LeScript und haben auch einen Drucker angeschlossen (Wer schreibt schon fuer die Diskette?). Jedoch gibt es bei den verschiedenen Drucker Probleme beim Ausdruck. So habe ich einen kleinen (jedoch sehr guten) Drucker, den BROTHER M1009 an meinem IVp. Er druckte zwar alles, jedoch keine Steuerzeichen wie Breit, Tief, etc.!
Diesem kann durch Steuerzeichen im Text abgeholfen werden. Das einzigste Problem: er kann noch keine Umlaute (aber das bekomme ich auch noch hin).

Die nachfolgende Liste zeigt, welche Tasten fuer welche Funktion stehen.

Steht in der Tabelle CLEAR-SHIFT-1 so ist die CLEAR-Taste zusammen mit der SHIFT-Taste und der Taste 1 zu druecken. Es sollte dann ein halbes Dreieck erscheinen, was den CHR$(27) darstellt. Nun koennen weitere Zeichen folgen.

Soll also unterstrichen gedruckt werden, so werden folgende Tasten gedrueckt.

CLEAR-SHIFT-1 dann
- (Minus) und dann noch
CLEAR-SHIFT-A

Alles klar?! So lassen sich alle Steuerzeichen auf dem BROTHER 1009 drucken. Auch Exoten wie den SEIKOSHA GP 550A habe ich auch das richtige Drucker beigebracht.

Viel Spass und wenig Papierverbrauch!

Frank

Steuerzeichen Brother M1009

Unterstreichen EIN	CLEAR-SHIFT-1 - CLEAR-SHIFT-A
Unterstreichen AUS	CLEAR-SHIFT-1 - CLEAR-SHIFT-@
Breit EIN	CLEAR-SHIFT-N
Breit AUS	CLEAR-SHIFT-T
Schmal EIN	CLEAR-SHIFT-O
Schmal AUS	CLEAR-SHIFT-R
Hochgestellt	CLEAR-SHIFT-1 S CLEAR-SHIFT-@
Tiefgestellt	CLEAR-SHIFT-1 S CLEAR-SHIFT-A
Normal	CLEAR-SHIFT-1 T
Vergr. und Breit	CLEAR-SHIFT-1 W CLEAR-SHIFT-A
Normal	CLEAR-SHIFT-1 W CLEAR-SHIFT-@

Druckersteuerzeichen für den SEIKOSHA GP550A

Unter der Textverarbeitung *LeSript* kann der Drucker SEISKOSHA GP 550A mit den nachfolgenden Steuerzeichen angesteuert werden.

Möchte man z. B. die LQ ELITE einschalten, so drückt man die Tasten SHIFT CLEAR und die Taste 1 zusammen. Es erscheint dann ein Dreieck auf dem Schirm. Nun noch das Steuerzeichen Q eingeben und der Drucker druckt alles in LQ ELITE. Die Steuerzeichen können auch gemischt im Text verwendet werden.

Alle Zeichen beziehen sich auf die eingedeutschte Version.

Frank Schoof

Breitschrift EIN:	SHIFT CLEAR 1 SHIFT CLEAR N
Breitschrift AUS:	SHIFT CLEAR 1 SHIFT CLEAR O
Standard PICA :	SHIFT CLEAR 1 N
Standard ELITE :	SHIFT CLEAR 1 E
Schmalschrift :	SHIFT CLEAR 1 C
LQ PICA :	SHIFT CLEAR 1 H
LQ ELITE :	SHIFT CLEAR 1 Q
Italic kursiv :	SHIFT CLEAR 1 B
Proportional :	SHIFT CLEAR 1 P
Superscript :	SHIFT CLEAR 1 U
Subscript :	SHIFT CLEAR 1 D
Unterstreichen :	SHIFT CLEAR 1 X
Ausschalten :	SHIFT CLEAR 1 Y

Club 80 Börse --- Club 80 Börse

Wer benötigt für seine Rechner GENIE I, GENIE II, III, TRS-80 III, etc. noch IC's ???

Habe noch über 40 IC's wie z. B. 4116, 4164, Z80, EPROMS, ROMS, etc. sowie einen Floppy-controler (GENIE I), 1 Floppy 40 DD/DS 1 Floppy 40 DD/SS, 1 Busadapter VG>TRS, 1 Centronics Interface GENIE I, 1 Minitool, 1 HRG 1B unbestückt sowie 2 Platinen zum ausschlachten !!!

Anfragen bitte schriftlich an:
Frank Schoof
Elpke 5
4800 Bielefeld 1

HRG-fähiger, sonst aber eher einfacher Matrix-drucker mit Epson-kompatiblem Befehlssatz gesucht. Angebote bitte an

Arnulf Sopp
Wakenitzstr. 8
2400 Lübeck 1
T. 0451-793054

Suche Bilder/Grafiken auf
IBM oder ATARI
Format dürfte bis hin zu VGA fast egal sein.
Hauptsächlich schwärme ich für farbiges.
Wer tauschen möchte:

Jens Neueder
0791/42877

TRX-280
The Next Generation of TRS-80
by Peter Ray (Anitek)

How would you like it if your TRS-80 had the expandability of an IBM XT, yet had more built-in features, ran faster, had more on-board memory, and all without having to make a single modification to your TRS-80 mother board? Sound fantastic? It is actually possible, and may soon even become a reality, with the proposed "Next Generation" replacement mother board, called the "TRX-280". This marvel of modern technology is already in the earlier stages of development as this article is being written.

Anitek Software Products has already begun developing a totally new replacement mother board for the TRS-80 that would empower your Model 4 and 4P to new levels of performance that were only before dreamed about by TRS-80 enthusiasts, like yourself. Our design criteria for the TRX-280 were simple. First, design a mother board the same size and dimensions of the TRS-80 mother board that would already have built into it most, if not all, the features that could, up until now, only be added to a TRS-80 by purchasing and installing a full compliment of the currently available modification kits. Second, design into the TRX-280 mother board many features that are not currently available to TRS-80 users even as modification kits - features that the original TRS-80 designers should have designed into it in the first place, but didn't - features like on-screen underlining, bold and italic which is a most needed feature in both the word processing and data processing environments. And third, even with all this new power, the TRX-280 had to be designed to continue to be totally TRS-80 compatible so that any programs that now run on a stock TRS-80 Model 4 or 3 will still run on the new TRX-280 mother board (except programs that use the cassette port).

The following is a list of the proposed features that Anitek believes can be designed into the TRX-280 replacement mother board, the size of the TRS-80 mother board, using the state-of-the-art technology that is available today.

1. A Z-280 microprocessor running at 10 MHz (which is equivalent to the speed of a Z-80 running at about 15 MHz).
2. On-board RAM expandable to 4 megabytes (possibly even 16 Meg).
3. Built-in bit-mapped graphics.
4. Built-in clock/calender.
5. Built-in hard drive host adapter.
6. On-screen text underlining.
7. On-screen text bolding.
8. On-screen italics.
9. On-screen expanded text.
10. Ability to display IBM characters, TRS-80 characters, block graphics, bit-mapped graphics, inverse video, bold text, underlined text, italics, expanded characters and normal size characters all on the screen at the same time.
11. A floppy disk controller that is capable of handling 180K, 360K, 720K, 1.2 Meg, and 1.4 Meg floppy drives.
12. Ability to boot directly from a hard drive without the need to first insert a floppy.
13. Ability to boot Model 3 DOSs directly on a 4P without MODELA/III.
14. One, or possibly even two, expansion slots, compatible with the IBM PC 8-bit expansion card architecture.

The on-board expansion slots would make it possible for the user to further expand his system with inexpensive half-size expansion cards currently available for the IBM PC. Some examples might include: hand scanners, modem cards, color monitor cards, mouse cards, and hard drive interface cards, just to name a few.

Sound too good to be true? Admittedly, it sounds fantastic, but it is absolutely possible! Much of the conceptual design work has already been done. In fact, about one year from today it could even be possible for you to purchase one of these new TRX-280 mother boards and have all of these features in your TRS-80.

Parenthetically let me state that even though we are very sure that we could design and build a multi-featured mother board the size and dimensions of the TRS-80 mother board, and even though we are very sure we could design a mother board with all the above features, it is still a little early yet to know for sure whether or not we can do both - i.e. make the mother board the size of TRS-80 mother board and at the same time give it every one of the new features mentioned above. The physical size of the mother board may impose a limit on the number of new features that we could design into it. However the TRX-280, if produced, will be the size of the TRS-80 mother board (that is a must) and it will have most, if not all, the above-mentioned features - all of the most important features for sure.

Yet, before Anitek Software Products can move from the safe and cozy arena of conceptual design on the TRX-280 project into the financially risky arena of actually developing and manufacturing such an awesome product, it will be necessary to first do a some careful market analysis to determine just how successful a product like the TRX-280 mother board will be in the TRS-80 marketplace. We already know that a mother board could be designed that could have most, if not all, of the features mentioned above, plus possibly even some additional features that were not mentioned. What we don't know is, are there enough people like yourselves who own TRS-80s that would be interested in buying the new TRX-280 mother board to justify to us the huge investment of time and money that it would take to develop and produce it. This is where you can help us out.

We would like to see what kind of interest a proposed product of this type will generate in the TRS-80 marketplace. If the TRX-280 is to become a reality, we will need to have this information early on, before we begin full-scale development. If the TRX-280 mother board were available today, would you buy it for your computer? Why or why not? If it were available a year from now (which is about the earliest it could be ready) would you buy it? Do the features sound interesting to you? Do you feel that some of the proposed features are of more value to you than other features? If so, which features do you rate the highest and which the lowest? Does the fact that you could have virtually every modification known for the TRS-80 built right into the mother board as standard equipment interest you? Would having a TRS-80 that you could plug off-the-self IBM compatible expansion card into interest you? Does a TRS-80 that could out perform an IBM XT interest you?

What other features have you thought would be nice to have built into a TRS-80 that are not mentioned in the above list?

We want your responses. If you have questions or comments regarding the TRX-280 mother board project, we want to hear from you. But we need to have you write your letter to CN 80. The editors have very graciously and enthusiastically agreed let us use this publication as an open forum for sharing ideas and questions regarding the TRX-280. By you sharing your thoughts with the magazine, you will also be giving the other readers the opportunity to see what you are thinking about the "Next Generation", TRX-280 concept. And at the same time, you will have the advantage of seeing what others are saying about this fantastic idea, as well. If any of the letters submitted to CN 80 need our attention for commenting on, the editors will make sure Anitek sees a copy of the letter ahead of time so that our comments can be published along with your letter.

If you are feeling like you can't live another day without one of these TRX-280 mother boards for your computer, or if you just think the concept sounds interesting, please let me strongly encourage you to, in the next day or two, write it down on a piece of paper and send it in to CN 80. And when you read the comments from other readers in next month's issue, if you have anything other ideas you would like to voice, write those down, too, and send them in. Your positive interest and enthusiasm shared in this publication is what it may take to generate in the other readers a strong interest as well. Excitement breeds excitement. Enthusiasm breeds enthusiasm. This way we can get the fires of interest in this "Next Generation" of TRS-80s burning faster and hotter. No comment or question is too small. If you have something you wish to voice about the TRX-280 before we begin development, please send it CN 80. It's going to take all of us to turn the TRX-280 from merely a great possibility into a great reality.

In anticipation of some of the questions that you might already have, I have answered what I feel may be the more obvious questions. Keep in mind though that no one can predict the future with absolute certainty. The answers given below are

based on possibilities and expectations of future events.

Q. How much will the new TRX-280 mother board cost?
A. Probably somewhere between $700 to $900 depending on the number of boards that we feel will be purchased. We will have a better idea after we have had the chance to measure the market interest and to tally the advanced orders before we go into the final design stages of production. The price could even be considerably less if a good number of people commit ahead of time to the purchase of the TRX-280 board by placing advanced orders. We are also looking into the possibility of offering an extra discount as an incentive to those who are able to place advanced orders, probably in the form of granting a discount equal to 1/2 the purchaser's advanced down payment. This could bring the price down to as little as $500 - $600 for those who are able to put that much down on an advanced order. Under this proposed plan a customer making an advanced payment of 2/3 of the cost of a TRX-280 mother board would get the last 1/3 free. This would be a fantastic way to substantially lower the cost of owning a TRX-280 mother board. But even at the expected normal price of $700 to $900, this is really still quite a bargain when you consider that it would cost over twice this amount to buy even half these features as individual modification kits, not to mention the days or weeks it would take you to install each one. By the way, if advanced orders are received with advanced payment but the number of which is not large even to justify the final development and manufacturing of the TRX-280, then the advanced payments would be returned.

Q. Will my other TRS-80 programs still work on the new TRX-280 mother board?
A. Yes. This is our number one most important design objective. We will do everything in our power to ensure that all TRS-80 Model 3/4 software (except programs that access the cassette port) will work as well or better on the new replacement mother board as they do on a stock TRS-80. It is a little early yet to know whether or not any DOS device drivers might require patches to work at the faster CPU speed. If there is any clever way to design around this in hardware, we will.

Q. Will the bit-mapped graphic circuitry of the TRX-280 be compatible with the Grafyx Solution board.
A. Yes. It is our number two most important design objective to make sure that all your Grafyx Solution programs work as well on the TRX-280 as they do on the TRS-80 mother board.

Q. How soon will the new TRX-280 mother board be ready to buy?
A. Once we begin full-scale development we expect it will only take about 12 months. The sooner we know that the TRX-280 is a product that that TRS-80 marketplace will buy, the sooner we will be able to start the full-scale development phase, and the sooner it will be available for you to purchase.

Q. Will I have to put a heaver power supply in my computer?
A. At this point we feel that it won't be necessary to change the power supply as long as it is in good condition and is not one of the earlier supplies that were small and under powered. Our objective is to design the board in such a way that it will work with the same power supply that you now have.

Q. Can I trade in my old mother board and get a credit toward the purchase of a TRX-280 mother board?
A. We may offer some sort of credit if you wish to trade in your old mother board, but it may only be for about $25 to $50. The fact is that the technology will be so much more advanced on the new mother board that there will be only a few components on the old TRS-80 mother board that will be usable on the new TRX-280 boards.

Q. Can I put one of the new TRX-280 mother boards in my Model 3.
A. Yes. Since the Model 3 has basically the same case and mother board housing as the non-gate-array Model 4, as far as we can tell you would be able to put a TRX-280 mother board in a Model 3.

Q. Will technical information be published to explain how to take advantage of all the new features?

A. Yes. We want you to get the most out of the product as possible, so we will publish what all the port addresses are and what they do, and the memory maps and all hardware technical info we can think of to benefit those who want that type of information.

Q. Will the TRX-280 mother board be available for the Model 4 first, or for the 4P?
A. That all depends on which one of the two versions we get the most advanced orders for before we begin final development.

Q. Will the new mother board be able to run MS-DOS programs?
A. No. We don't feel that there is any need for us (or anyone else) to come up with another IBM clone. If we thought you wanted a computer that could run MS-DOS programs we would have thought that you would have bought an MS-DOS computer. The TRX-280 mother board is for people who love their TRS-80s and don't care to switch over to the MS-DOS world. Yet, at the same time, they see what the MS-DOS machines can do and they long for a way that they could have the same power and performance on there TRS-80.

If you have any other questions, and I am sure you do, please send them into CN 80 as soon as possible so they can get into the next issue.

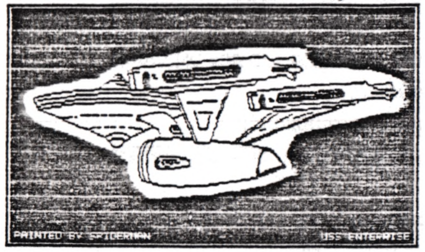
In closing, let me stress that the outcome of the TRX-280 replacement mother board project is now in the hands of you, the readers. I know you have a great attachment to your TRS-80, or you wouldn't be reading this magazine. We at Anitek also share in this love for what was and is still one of the greatest little computers ever made. But our strong appreciation for the TRS-80 only makes us more keenly aware of its several designed-in limitations and makes us long for the TRS-80 that could be, the TRS-80 that is now possible with the advent of the TRX-280 replacement mother board. But the bottom line is this, we know that we can design and build the TRX-280, but whether this product ever sees the light of day is something that you now have the opportunity to decide. Are you ready to usher-in the "Next Generation" of power, speed, features, capabilities, and expandability - or are you satisfied with the current generation of TRS-80 just the way it is? Cast your vote of support for the TRX-280 by writing a letter to this publication. And if you are already sold on the idea and would like to place an advanced order to lock-in a lower price for yourself, let us know that too, so we can send you the details on how to place an advanced order. But again, please do so through CN 80 so that all readers can see the degree of interest this new product is generating. I am eager to hear your votes, answer your questions, respond to your comments, and discuss more of the aspects of the "Next Generation" with you again next month.
-Peter Ray (Anitek)

MOUSE ON YOUR MODEL 4
(Without an Interface) Part II
by David Goben

I've received numerous complaints that I did not follow my first mouse article with the follow-up in the very next issue, or the one after that. Actually, I had only said that I would cover some things in my -next- article on the mouse, and did not promise to cover it immediately. Finally, I apologize for the technical depth of this article, but due to space constrains, I cannot expect to be able to use the entire issue of CN80 to explain the intricate working of a mouse in a more comprehensible format.

The idea of communicating with a serial mouse may have seemed complicated, because no one seemed to think it could be done. It was probably just one of those things where one simply assumes that such a task to be incredibly complex. Fortunately, Scott McBurney did not think so, and blazed the trail toward using a mouse on a TRS-80, thus sparking my own underlying interest. My contribution to this effort was to improve on the basic driver design, simplify the interfacing process, give it more sophisticated commands, and allow it to work on both a 2-button and a 3-button

Hallo Clubfreunde,



heute gehe ich auf Entdeckungsreise in der Club-Bobliothek. Jeder kennt sicherlich noch den Spruch: "Unendliche Weiten (und weit und breit kein Spock zu sehen) ...". In Anlehnung hieran gilt wohl ähnliches für unsere emsigen Club-Bücher-Leser. Unendliche Mengen an Bücher, nur keiner will sie lesen. Bei mir stehen sie - wie wohl auch zuvor bei den anderen Bibliothekaren - zwar warm und trocken, trotzdem wäre es wesentlich angebrachter ihnen die Weiten Deutschlands und die Wohnorte der Club 80er zu zeigen. Scheinbar wähnt jeder sein eigenes Regal viel interessanter als das was hier so steht. Weit gefehlt liebe Leute ! Hier könnt ihr noch stöbern und dabei so manchen lukrativen "Hecht" an Land ziehen. Bereits beim ersten Griff ins gedränge werde ich fündig:

Etliche Bände zum TRSDOS, TRS-80 und Innenleben ...

1. *BASIC FASTER AND BETTER &* OTHER MYSTERIES
   A guided tour of BASIC programming tricks and techniques.

2. *TRS-80 DISK &* andere Geheimnisse
   Ratgeber zur Datenwiederauffindung

3. *TRSDOS 2.3 DECODED &* OTHER MYSTERIES
   Commented Source and guide to TRSDOS 2.3 for the Model I

4. *BASIC DISK I/O FASTER AND BETTEER &* OTHER MYSTERIES
   TRS-80 Random and Sequential File Programming: Beginner, Intermidiate, Advanced

5. *THE CUSTOM TRS-80 &* OTHER MYSTERIES
   The Complete guide to Customizing TRS-80 Software and Hardware

... eine gebündelte Ladung an echt guten Dingen. Nun mal nicht gleich abgewunken !!! Das etwas gealterte TRSDOS ist zwar inzwischen der letzte Schrei, aber wir sind doch wohl immer noch ein Club der Z80-Freunde, *ODER* ?! Na also.

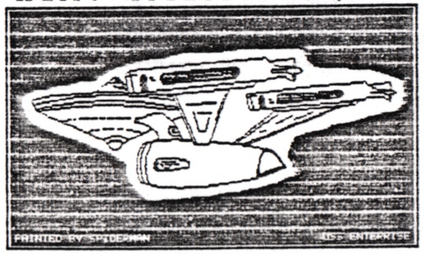
Gut ! Kommen wir zum Inhalt. Am interessantesten - und wer hat schließlich schon problemlos Zugriff auf das Listing eines Betriebssystems - scheint mir der Band von Punkt 3 zu sein. Hier kann man nach aller Herzenslust in den SYS-Modulen des TRSDOS blättern und sich über die Funktionsweise unterrichten lassen oder auch nur nach Tips, Tricks und guten Ideen ausschau halten.

Wem das nicht so zusagt, kann sich auch den Punkt 1 näher ansehen und sich dort einige Inspirationen zum Thema BASIC abholen. Der Band enthält eine ganze Reihe von interessanten Utilities zur BASIC-Erweiterung oder um sich das debuggen von Basic-Programmen zu erleichtern. Lohnenswert allemal ...

Ein Problem will ich jedoch nicht verschweigen, etwas Englisch sollte man beherrschen. Oder man gibt sich mit den Listings zufrieden, denn was dort steht ist ohne Untertreibung auch schon "sein Geld wert".

---------------<#>-----------------

Hallo Clubfreunde,



es ist ein Neuzugang in unsere Bibliotek zu verzeichnen. Oliver Volz hat es mir geschickt. Wie ich feststellen konnte, handelt es sich um die englischsprachige Ausgabe der bereits in der Clubbibliotek vorhandenen deutschen Version. Beim Vergleich beider Bücher fiel mir auf, daß in der englischen Version das Inhaltsverzeichnis deutlich detailierter und besser gegliedert ist (!). Vom Inhalt her scheint aber in beiden das Gleiche zu stehen.

Beim zwanglosen Durchblättern fand ich in der Rubrig "4. Betriebssysteme" eine Kommentierung der Generationsfolge der TRS-80 Systemsoftware. Angefangen mit TRSDOS 2.1 ... 2.2, NEWDOS 2.2 ... 2.2, VTOS etc. Sogar ein Ausblick auf zukünftiges in dieser Richtung ist zu finden (NEWDOS 80 wird als heiser Tip gehandelt und seine tollen neuen Fähigkeiten angerissen !). Leute !, die Geschichte gibt mehr her, als man der Zukunft nachsagt. Auf jeden Fall ist klar zu übersehen, was man verpaßt hat.

So weit, so gut. Gehen wir wieder einmal auf Entdeckungsfahrt. Beim Stöbern bin ich auf zwei Bücher gestoßen, deren Inhalt sich recht interessant liest. Die Buchtitel lauten:

1. *Listing-Verwaltung in MBASIC*
   aus dem W.-D. Luther-Verlag

2. *Mikrorechner in der Amateurmeßtechnik*
   aus dem Militärverlag der DDR (VEB)

Buch 1:
Es wird in allen Einzelheiten ein Verwaltungsprogramm zum Erfassen von Buch-, Zeitschriftentiteln und einzelnen Artikeln entwickelt. Erfaßbare Daten sind:

> Zeitungstitel, Titel, Ausgabe, Seite, Seitenumfang, Serien, Stichwörter, Rechnertypen, Programmiersprachen, Textsprachen und Programmarten.

Das ganze ist in MBasic für einen CP/M-Rechner geschrieben. Leichte Anpassungen in Bezug auf das Zusammensetzen der File- und Laufwerksbezeichnungen zum vollen Dateinamen sind entsprechend dem eingesetzten Betriebsystem notwendig. Das Programm scheint so gut parametrisiert zu sein, daß Änderungen nur an wenigen stellen zu tätigen sind um zur gewünschten Anpassung zu kommen. Zentrale Drehscheibe des Gesamtprogramms ist ein Menüprogramm namens "START". Von hier aus werden die einzelnen Funktionen per RUN gestartet. Es könne also u.U. Teile einfach ausgelassen werden, bzw. die Eingabe des Programmsystems kann übersichtlich in handlichen Schritten erfolgen. Wer also eine umfangreiche Loseblattsammlung an Artikeln aller Art sein Eigen nennt, hat hier die möglichkeit, ein für allemal Ordnung zu schaffen. Das Buch hat einen Umfang von ca. 110 Seiten und ist zu gut 80% Listing. Es sind 8 Module mit durchschnittlich 500...600 Zeilen Umfang abzutippen. Etwas Geduld sollte also vorhanden sein. Aber es lohnt sich (wer wagt es ???).

Buch 2:
Wer sich für Hardware in Verbindung mit Rechneransteuerung interessiert, findet hier eine Menge Anregungen zum Experimentieren. Vom Autor werden rechneransteuerbare Sinusgeneratoren, Netzgeräte D/A und A/D-Wandler, Schrittmotoransteuerungen, Frequenzzähler etc. besprochen. Die zur Ansteuerung erforderlichen Routinen sind überwiegend in Z80-Assembler verfaßt. Es handelt sich aber nicht um vollständige Programme, sondern nur um die Treiber (!). Die Applikation muß der Anwender selbst schreiben. Etwas gewöhnungsbedürftig sind die Schaltbilder. Hier wird man mit dem DDR-System konfrontiert. Es ist aber entzifferbar. Mit den IC-Bezeichnungen ist es auch so ein Ding. Die Bezeichnungen sind nicht zu gebrauchen. Hier muß aus der Funktion und dem Text auf das entsprechende TTL-IC geschlossen werden ! Da die Schaltungen aber meist von geringem Umfang sind (4...8 TTL-IC's bzw eine Handvoll analoger Bauteile wie OP's und Transistoren), ist es kein Problem. Am Ende des Buches wird auch noch einmal auf unterschiedliche Verfahren zur Meßwertglättung eingegangen (stärken & schwächen) und eine Assemblerroutine dazu vorgestellt. Wer häufiger Meßprobleme hat und eine "schnelle Lösung" sucht, sollte hier einen Blick hineinwerfen.

------------<#>-----------------

Impressum

1. Vorsitzender — Fritz Chwolka
Tel.: 0 24 64/89 20 — Saarstraße 4
5173 Aldenhoven

2. Vorsitzender — Gerald Schröder
Tel.: 0 41 05/26 02 — Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator — Andreas Magnus
Tel.: 02 09/87 02 30 — Pommernstraße 4
4650 Gelsenkirchen

Newdos-Diskothekar — Oliver Volz
Tel.: 07 11/7 35 38 17 — Waldburgstraße 73
7000 Stuttgart 80

CP/M-Diskothekar — Rüdiger Sörensen
Tel.: 0 61 31/3 28 60 — Thomas Mann Straße 3a
6500 Mainz 1

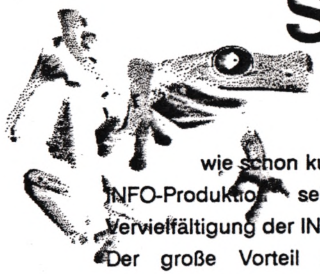
Clubbücherei — Kurt Müller
Tel.: 0 41 52/7 06 43 — Sophie-Scholl-Ring 38
2054 Geesthacht

Redaktion — Jens Neuder
Tel.: 07 91/4 28 77 — Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 8870 Günzburg
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.



Schluß

Hallo Club 80'er,

wie schon kurz berichtet, haben wir uns nun mit der INFO-Produktion selbstständig gemacht. Layout und Vervielfältigung der INFO's erfolgt nun aus einer Hand.
Der große Vorteil des clubeigenen Kopierers ist der Zeitgewinn für die Fertigstellung des INFO's. So habe ich jetzt die Möglichkeit auch nebenher die eine oder andere Seite vorzubereiten. Auch die Ergebnisse, die wir mit unserem clubeigenen Gerät erzielen, sind doch recht annehmbar.

Leider hatte ich nicht genügend Material, um in allen Sparten unseres Club-INFO's etwas zu bringen. Ich hoffe aber, das dies bis zum nächten INFO der Fall sein wird. Zumindest werde ich mich, für meinen Teil, zukünftig bemühen, das jeweils fällige INFO ca 3/4 Wochen nach dem Redaktionsschluß fertig zu haben. Der nächste Redaktions-Schluß ist, nochmals zur Erinnerung, Ende Mai.

Als Beilage gibt es diesmal ein Sonder-Heft für die Modell-4-User:
Portbelegung und BOOT-ROM-Dokumentation des M IV

Abschließend hoffe ich, daß Euch unsere neueste Club-80-Paperware zusagt und Ihr viel Spaß daran - und Anregungen darin findet.

Übrigens:
Vergesst nicht, Euch zum Clubtreffen, vom 09.05.91 bis 12.05.91 in Weilburg, anzumelden!
Das in der Anmeldung **"Ich reise am ..... April gegen ..... Uhr an."** war natürlich ein kleiner Scherz zum 1. dieses Monates. Es heißt natürlich **"Mai"**! Sicher haben dies alle bemerkt. Eine Anmeldung kann aber selbstverständlich trotzdem mit diesem Blatt erfolgen.

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

CLUBINFO

34. AUSGABE

KONTAKTADRESSE : CLUB 80 / FRITZ CHWOLKA / SAARSTRAßE 34 / 5173 ALDENHOVEN
TEL.: 02464 / 8920

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Das Ausgefallene Clubtreffen 1991 — 1
Gerald Schröder

Termine — 2
Redaktion

Software

ADRESS - TEXT - 90 — 3 - 8
Willi Johnen

Text-Konvertierung IBM-PC nach CP/M — 9 - 12
Helmut Bernhardt

C-Einführung — 13 - 15
BNF für (wahre) Programmierer — 16 - 18
Alexander Schmid, Gerald Schröder

The Producer — 19 - 23
Jens Günther

Hardware

Club-80-Terminal auf Basis der PC-Hercules-Karte — 25 - 33
Erweiterung ECB-Interface auf 4 Portadressen 34
Helmut Bernhardt, Stefan Nitschke

Börse

Wer hat was -- wer will was — 35 - 38

Sonstiges

NEWDOS, CP/M, MS-DOS UNIX
(Leserbrief zu INFO 32) — 61 - 64
Helmut Bernhardt

Die letzten Seiten

Impressum — 41
Schluß — 42
Redaktion

Mitgliederadressenliste — am INFO-Ende

Das ausgefallene Clubtreffen 1991

von Gerald Schröder

In diesem Jahr fand kein Haupttreffen des Club 80 stand, denn es hatten sich nur einige wenige Mitglieder angemeldet. Der Vorstand bleibt also dieses Jahr unverändert im Amt. Und es ändert sich nichts am Club, insbesondere wird der Club immer noch nicht in "Club 86" wg. MS-DOS umbenannt, wenn auch Artikel zu dem Thema gern gesehen wären ... wenn jemand etwas dazu schreiben würde!

Hartmut war also gezwungen, allen angemeldeten Mitgliedern abzusagen. Gerald beschwerte sich, daß er die Bahnfahrkarte schon gekauft habe, und aus lauter Verzweiflung lud Hartmut ihn zu sich ein, um Gerald zu seiner Bahnfahrt zu verhelfen. Das sprach sich unter den angemeldeten anderen Club-Mitgliedern rum und sie luden sich auch zu Hartmut ein. Den packte nun angesichts der auf ihn zustürmenden Horden die nackte Panik und er schickte Jutta auf die Suche nach einer geeigneter Zuflucht.

Und so fanden sich mehr oder weniger zufällig vom 9. bis 12. Mai im Hotel "Zum Bären" in Leibheim zusammen:
Helmut Bernhardt, Fritz Chwolka, Manfred Held, Andreas Magnus, Jens Neueder, Christof Neumann, Hartmut Obermann, Alexander Schmid, Rainer Schmitz, Gerald Schröder. Und als Special Guests: Jutta Obermann, Karin und Jenny Held.

Natürlich wurde wieder gehackt ohne Ende. Diesmal waren sogar die MS-DOS-Rechner in der Minderzahl, dafür hatten wir zwei Z280-Rechner (Alexander und Fritz), Andreas x-mal umgebautes Genie IIIs, Rainers Schneider Joyce und Christofs Model IVp und II. Verbunden wurden die meisten Rechner durch Rainers Netzwerk, das in einem eigenen Artikel vorgestellt wird und damit offiziell als "Club-Netzwerk" eingeführt wird.

Termine... Termine... Termine... Termine... Termine

....- 91

Clubtreffen Nord

Redaktionsschluß für die Clubinfo's 1991 ist jeweils der letzte Tag der folgenden Monate:
Juli
September
Dezember

IAA	Frankfurt	12.09.- 22.09.91
Büro-data	Berlin	16.10.- 19.10.91
SYSTEMS	München	21.10.- 25.10.91

In einem unserer letzten Gespräche habe ich Dir ja versprochen,etwas für die Clubnachrichten zu tun. Hier ist nun mein Beitrag:

A D R E S S - T E X T - 90 ,so heißt daß von mir 1982 entwickelte Adress-Text-Programm - bis zum 1.3.1990 unter ADRESS-TEXT-80 - mit seinen manigfachen Möglichkeiten. In den Anfangsjahren lief es auf den GENIE I + II-Rechnern, 1984 auf GENIE III und Speedmaster und 1986 auf GENIE IIIS.
Das Programm ist so ausgereift, daß theoretisch keine Fehlerquellen mehr vorhanden sind.
ADRESS-TEXT-80 wurde von mir außer in Deutschland auch nach Spanien,Frankreich und Österreich verkauft. Mittlerweile - seit 3 Jahren - verkaufe ich dieses Programm nicht mehr, da andere Adress-Text-Programme der feineren Art den Wettbewerb so hart machten, daß sich ein Verkauf nicht mehr lohnte,weil jedes Programm von mir für Benutzer individuell umgearbeitet wurde.

Trotzdem habe ich ADRESS-TEXT-80 weiterentwickelt,nachdem ich auch im Besitz der hochauflösenden Graphik der Firma TCS war.Das neue ADRESS-TEXT90 arbeitet nur noch mit hochauflösender Graphik. Da die Abspeicherung der hochauflösenden Graphiken viel Platz einnimmt, mußte ich mir eine Festplatte anschaffen.
Für das komplette ADRESS-TEXT-90 braucht man nachfolgende Geräte:

*** 1 Gerät GENIE III S mit 1x10MB-Platte u. 2 LW DD
mit eingebauter Echtzeituhr
mit hochauflösender Graphik

*** 1 Drucker ITOH 85100 oder 8510

Clubmitglieder die Interesse hieran haben, können sich gerne mit mir in Verbindung setzen.
Das ganze Programm ist in G-Basic geschrieben. Auf andere Sprachen kann ich mich anfreunden, da das entwickelte G-Basic von TCS beispielhaft auf dem Internationalen Markt war und auch noch heute sein dürfte. Dies gilt vor allen Dingen dem Programmieren der hochauflösenden Graphik mit seinen unwahrscheinlichen Möglichkeiten - soetwas gibt es in CPM nicht !!

Als Einführung in das ADRESS-TEXT-90 anbei 3 Seiten Listing der Textverarbeitung, die sogar einen ganzseitigen Bildschirmeditor intregiert hat ! Wo findest Du so ein Programm in der Kürze bei CPM ??

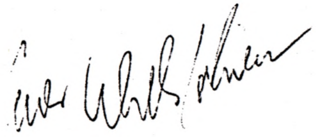
Hinweise: Dieses Programm läuft nur mit hochauflösender Graphik und einer
--------- installierten Festplatte (LW5 + LW6)

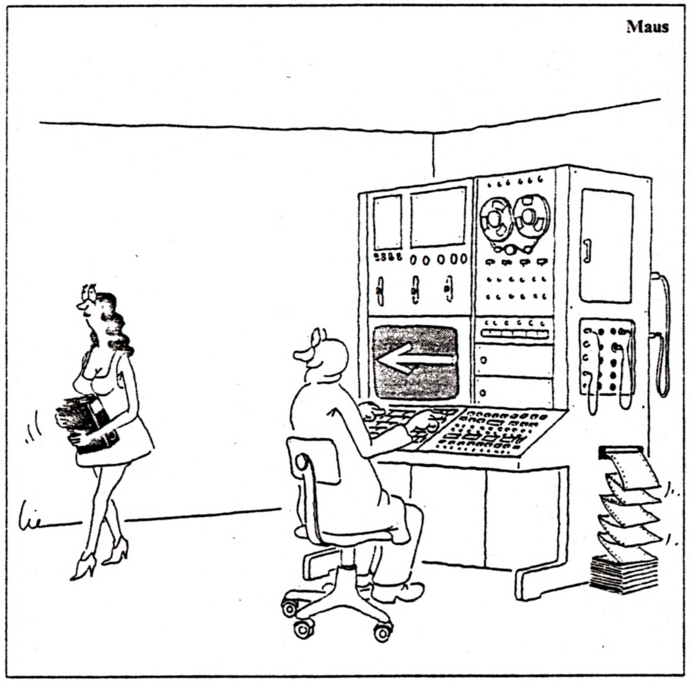
Nachfolgende Zeilen müssen jenachdem geändert werden:
Z.0110 Hochauflösende Graphik
Z.0250 Anzahl der möglichen Zeilen eines Textes

Z.0750 Anzahl der möglichen Zeilen eines Textes
Z.0850 Begrenzung der Zeilen in der Bildschrimanzeige
Z.0980 Name des Programm-Inhabers
Z.1540 "DIR 5,/TXT = Laufwerksänderung wenn keine Festplatte
Z.1590 TEXTBILD/RDL:5 muß selber gezeichnet werden
Z.1670 Laufwerksangabe ändern wenn keine Festplatte
Z.1680 Sprung in ein anderes Hauptmenü

Die Textbilder sind zum selber-zeichnen beigelegt.
ACHTUNG: Es gibt von der Firma TCS 6 Versionen von hochauflösenden Graphiken. Eine davon ist nach Auskunft durch ehemalige Mitarbeiter der Firma TCS nur einwandfrei lauffähig. Diese Version muß lauten: GENIE - DOS 2.4. 1985 PHÖNIX. Dies ist eine Folgeversion des GENIE - DOS 2.4. 1984 TCS. Die Folgeversion wurde von Mitarbeitern der TCS, die anschließend bei Phönix arbeiteten, weiter ausgearbeitet.

Und nun viel Spaß beim Eintippen des Programms, keine Angst, es läuft !!





```
10 '#>> Programmname  << JOHTEXTV/BAS >>                <<#
20 '#>> Textverarbeitung des ADRESS-TEXT-90             <<#
30 '#>> Stand: 20.07.1990                               <<#
40 '#>> Hochauflösende Graphik                          <<#
50 '#>> Textspeicherung in LW 5  (ohne HD in LW 1)      <<#
60 '#>> Graphikbild:  >JOHNEN01/RDL< + >TEXTBILD/RDL<   <<#
70 HOFF:HCLS:HON:CLEAR12000:DEFSTRA-D,G-H,J-K:DEFINTN,X,Z,W:DIMA3(160)
80 '#--------------------- Graphik $ -----------------------#
90 CC=STRING$(15,140):CY=STRING$(78,128):AD="TEXT-NEUEINGABE":AE="TEXT-
":AF="TEXT-EINFÜGEN":AS="TEXT-SPEICHERN"
100 '#------------------Menüanzeige ----------------------#
110 CLS:HLOAD"JOHNEN01/RDL":HCODE"BLOCK/RZS":HBIG65,225,1,1,1,"* TEXTVE
,1,1,1,LEFT$(TIME$,8);:HBIG536,202,1,1,1,RIGHT$(TIME$,8);
120 ONERRORGOTO0:PRINT§461,CC:PRINT§541,"PROGRAMM - MENÜ":PRINT§621,CC:
§939,"(2) ";AE:PRINT§1099,"(3) ";AI:PRINT§1259,"(4) HAUPT-MENÜ";
130 WU=VAL(INKEY$):IFWU<1ORWU>4THEN130
140 ONWUGOTO160,370,1430,1680
150 '#--------------- Textneueingabe ---------------------#
160 GOSUB1560:GOSUB1540:PRINT§131,AD;:GOSUB1150
170 ONERRORGOTO220
180 OPEN"I",1,KA
190 IFEOF(1)THENCLOSE:GOTO160
200 N2=N2+1:LINEINPUT#1,A3(N2)
210 GOTO190
220 RESUME230
230 GOSUB1590
240 CLS:PRINT§2,AD;:GOSUB1580
250 NN=160        'Anzahl der möglichen Zeilen
260 IFAY!=1THEN270ELSEII=0
270 L!=73          'Anzahl der möglichen Zeichen in einer Zeile
280 BP%=245        'Cursorbeginn bei Neueingabe
290 K=""
300 IFAZ!=2THENBP%=BP%+79:II=II+1:A3$(II)=K:K=""
310 IFII=NNTHENN2=II:PRINT:AY!=0:GOSUB850:KQ="":GOTO520
320 IFLEFT$(A3$(II),1)="/"THENN2=II-1:PRINT:AY!=0:CLS:GOTO460
330 BP%=BP%+1:IFBP%>1920THENAY!=1:AZ!=0:PRINT§159," ";:PRINTUSING"###";
0250
340 GOSUB1190:GOSUB1400
350 GOTO300
360 '#--------------- Textänderung ----------------------#
370 GOSUB1560:GOSUB1540:PRINT§131,AE;:A8="Ä":GOSUB1150
380 ONERRORGOTO430
390 OPEN"I",1,KA
400 IFEOF(1)THEN450
410 N2=N2+1:LINEINPUT#1,A3(N2)
420 GOTO400
430 RESUME370
440 '#---- Texterfassung (Änderung/Menüanzeige) ----------#
450 GOSUB1590
460 PRINT§239," ";:NL=0:N1=0
470 NM=13:IFNM>N2THENNM=N2
480 NX=0
490 NX=NX+1:IFNL=160THEN520
500 NL=NL+1:PRINTUSING"###";NL;:PRINT".Z:";A3(NL):IFNL=N2THEN520
510 IFNX=21THEN520ELSE490
520 JQ="":AZ!=0:GA="":CLOSE:PRINT§2,"M E N ü :   <Ä>ndern   <E>infügen
chern   <L>öschen";
530 GA=INKEY$:IFGA="s"THENGA="":A8="":GOTO870
540 IFGA="e"THENGA="":GOTO690
550 IFGA="b"THENGX="B":GOTO660
560 IFA8="Ä"THENIFGA="1"THENPRINT§2,CY;:PRINT§2,"TEXT-LÖSCHEN";TAB(53)"T
OTO70
570 IFGA="ä"THEN590ELSE530
580 '#----------- Texterfassung (Änderung) --------------#
590 KQ="Ä":IFGX="B"THENII=N1+1ELSEII=1
600 PRINT§2," Zum Löschen einer ganzen Zeile als 1.Zeichen der Zeile ein < § > setzen    ";
610 BP%=246:AZ!=0:L!=73:K=A3(II):GOSUB1630:PRINT§BP%,CHR$(14);
620 SL!=1:ONERRORGOTO630:GOSUB1250:A3(II)=K:K=""
630 RESUME640
640 II=0:CLS:KQ="":GOTO460
650 '#---------- Texterfassung (Änderung) ----------------#
660 N1=N1+21:IFN1=<N2THENPRINT§239," ";:FORI=1TO20:PRINTCY:NEXTI:PRINT§239," ";:GOTO470
670 GOTO460
680 '#---------- Texterfassung (Einfügen) ----------------#
690 JQ="X":IFN2>=160THENGOSUB850:GOTO520
700 II=0:NN=0:PRINT§2,CY;:PRINT§2,AF;" -- Max.21 Zeil.-- Anzahl der Zeilen ";:INPUTNN:IFNN=0ORN
>21THEN700
710 PRINT§57,"-- Nach Zeilennr.";:INPUTII:IFII=0ORII>N2THEN700
720 IFNN>(160-N2)-1THENNN=160-N2:GOSUB850
730 N4=N2+NN:FORNS=N4TOII+1STEP-1:NP=NS-NN:IFNP<IITHENA3(NS)="§":GOTO750
740 A3(NS)=A3(NP)
750 NEXT:CLS:PRINT§2,CY;:PRINT§2,AF;" -- ";A4;" -- ";:PRINT§159," ";:PRINTUSING"###";II;:PRINT"
Z:";A3(II)
760 QR=0:L!=73:BP%=245:K=""            'BP%=Startcourser bei Texteinfügen
770 FORNS=II+1TOII+NN:IFAZ!=2THENBP%=BP%+79:K=""
780 BP%=BP%+1:QR=QR+1
790 IFNN<10THENPRINT§BP%-4,USING"#";QR;:GOTO810
800 IFNN=>10THENPRINT§BP%-5,USING"##";QR;
810 PRINT".E: ";:GOSUB1190:A3(NS)=K
820 IFLEFT$(A3(NS),1)="/"THENA3(NS)="§":NS=II:A3(NS+NN)="§":GOTO840
830 NEXT
840 N2=N2+NN:CLS:GOTO460
850 PRINT§2,CY;:PRINT§17,"Die Länge des Briefes ist auf 140 Zeilen begrenzt  !";CHR$(7);:FORI=1
01200:NEXTI:PRINT§2,CY;:RETURN
860 '#------- Text speichern mit Kontrolldruck -----------#
870 PRINT§2,CY;:PRINT§2,AS;" ----- Kontrolldruck ?    < J/N >";
880 GA=INKEY$:IFGA="j"THEN890ELSEIFGA="n"THENGA="":PRINT§42,"< Nein >";:ONERRORGOTO0:GOTO1000EL
E880
890 PRINT§42,"< Ja >          Textkopf ?    < J/N >";
900 GS=INKEY$:IFGS="j"THENPRINT§69,"< Ja > ";ELSEIFGS="n"THENPRINT§69,"< Nein >";ELSE900
910 GOSUB1700
920 '#--------------- Drucker einstellen -----------------#
930 IFINP(253)=63THEN940ELSEPRINT§2," Ihr Drucker ist nicht startbereit.     Prüfen Sie den Dru
cker (SEL-Lampe)! ";:FORQ=1TO200:NEXTQ:PRINT§2,CY;CHR$(7);:GOTO930
940 PRINT§2,"  Ist das Formular richtig eingestellt ?    <D>rucken   <M> zurück zum Menü  ";CHR$
(7);
950 GY=INKEY$:IFGY<>"d"ANDGY<>"m"THEN950
960 IFGY="m"THENCLOSE:GOTO70
970 '#--------------- Mustertextdruck -------------------#
980 ONERRORGOTO0:IFGS="n"THEN990ELSELPRINTTAB(7)"Willi Johnen, Düren -----  Textname:  ";A4;"  "
+"Düren,den ";LEFT$(TIME$,8):LPRINT:LPRINT
990 GOTO1010
1000 GOSUB1700
1010 A4=KA:XP=0:CLS:PRINT§2,AS;TAB(53)"TEXTNAME: "+A4;
1020 PRINT§174,"< H > = Text anhalten    < W > = Text wieder starten":FORI=1TO400:NEXTI:PRINT
1030 FORNU=1TON2:IFLEFT$(A3(NU),1)="§"THEN1130ELSEPRINT#1,A3(NU)
1040 GC=INKEY$:IFGC<>"h"THEN1060
1050 GC=INKEY$:IFGC<>"w"THEN1050
1060 PRINTTAB(2)" ";:PRINTA3(NU)
1070 IFGY="d"THENXP=XP+1
1080 IFGS="n"THENIFXP=61THENAJ="2":GOSUB1600:GOTO1120
1090 IFGS="n"THENIFXP=120THENAJ="3":GOSUB1600:GOTO1120
1100 IFGS="j"THENIFXP=58THENAJ="2":GOSUB1610:GOTO1120
1110 IFGS="j"THENIFXP=117THENAJ="3":GOSUB1610:GOTO1120
1120 IFGY="d"THENLPRINTTAB(7)A3(NU)
1130 NEXT:CLOSE:GOTO70
1140 '#----------- Brief/Textnamen-Abfrage --------------#
1150 N2=0:GOSUB1670
1160 IFLEN(A4)<>14THENPRINT§1858,"Textname ist zu lang oder falsch eingegeben ";CHR$(7);:FORI=1T
01500:NEXT:PRINT§1840,CY;:GOTO1150
```

```
1170 KA=A4:RETURN
1180 '#---------------- Editfunktion ---------------------#
1190 GOSUB1630
1200 IFJQ<>"X"THEN1220
1210 PRINT§BP%,K;:PRINT§BP%,CHR$(14);:GOTO1240
1220 PRINT§BP%,K;:IQ=II+1        'IQ=Zähler für Neueingabe-Zeilen
1230 PRINT§BP%-7,IQ:PRINT§BP%,CHR$(14);
1240 SL!=1
1250 IFSL!>L!THENSL!=L!ELSEIFSL!<1THENSL!=1:PRINTCHR$(15);
1260 PRINT§BP%+SL!-1,CHR$(14);
1270 K1=INKEY$
1280 IFK1=""THEN1270
1290 IFKQ="Ä"THEN1310
1300 IFK1=CHR$(10)ORK1=CHR$(11)THEN1270ELSE1330
1310 IFK1=CHR$(10)THENA3(II)=K:K="":BP%=BP%+80:II=II+1:K=A3(II):IFBP%>1846THEN1390ELSE1250
1320 IFK1=CHR$(11)THENA3(II)=K:K="":BP%=BP%-80:II=II-1:K=A3(II):IFBP%<246THEN1390ELSE1250
1330 IFK1>=CHR$(32)THENPRINT§BP%+SL!-1,K1;:MID$(K,SL!,1)=K1:SL!=SL!+1:GOTO1250
1340 IFK1=CHR$(8)THENSL!=SL!-1:GOTO1660
1350 IFK1=CHR$(9)THENSL!=SL!+1:GOTO1660
1360 IFK1=CHR$(31)THENSL!=1:K=STRING$(L!,32):PRINT§BP%,K;:K="§":GOTO1660
1370 IFK1=CHR$(25)THENK=LEFT$(K,SL!-1)+" "+LEFT$(RIGHT$(K,L!-SL!+1),L!-SL!):PRINT§BP%,K;:GOTO166
0
1380 IFK1=CHR$(24)THENIFSL!=1THEN1250ELSEK=LEFT$(K,SL!-2)+RIGHT$(K,L!-SL!+1)+" ":PRINT§BP%,K;:SL
!=SL!-1:PRINTCHR$(15);:GOTO1250ELSEIFK1=CHR$(11)THENAZ!=1:GOTO1390ELSEIFK1=CHR$(10)ORK1=CHR$(13)
THENAZ!=2:GOTO1390
1390 PRINTCHR$(15);:RETURN
1400 IFJQ<>"X"THENE!=VAL(K)
1410 RETURN
1420 '#-------------- Inhaltsverzeichnis Briefe ----------#
1430 GOSUB1560:GOSUB1540:PRINT§131,AI;
1440 GOSUB1570:PRINT§1857,"Wollen Sie einen Brief/Text löschen ?   < J/N >";
1450 D=INKEY$:IFD="j"THEN1470ELSEIFD="n"THEN70ELSE1450
1460 '#------------- Texte/Briefe löschen ----------------#
1470 ONERRORGOTO1500
1480 GOSUB1670
1490 IFLEN(A4)<>14THEN1480ELSEKILLA4:GOTO1520
1500 RESUME1510
1510 ONERRORGOTO0:GOTO1470
1520 D="":GOSUB1570:PRINT§1856,"Wollen Sie weitere Texte löschen ?    < J/N >";
1530 D=INKEY$:IFD="n"THEN70ELSEIFD="j"THENGOSUB1570:GOTO1480ELSE1530
1540 PRINT§205," ":CMD"DIR 5,/TXT":PRINT§458,"Nebenstehende Briefe";:PRINT§538,"oder Texte sind
ge-";:PRINT§618,"speichert !";:RETURN
1550 '#------------- Bildschirmroutinen ------------------#
1560 X6=298:FORX=1TO18:PRINT§X6,STRING$(20,32);:X6=X6+80:NEXTX:RETURN
1570 PRINT§1840,CY:RETURN
1580 PRINT§18,"-- ";A4;" -- S1=33/S2=85 -- </>=ENDE -- DAT= ";:DA=LEFT$(TIME$,8):PRINTDA;:RETURN
1590 ONERRORGOTO0:CLS:HCLS:HLOAD"TEXTBILD/RDL:5":RETURN
1600 GOSUB1620:LPRINTTAB(34)"Seite ";AJ:LPRINT:LPRINT:LPRINT:AJ="":RETURN
1610 GOSUB1620:LPRINTTAB(7)"Seite ";AJ;TAB(38)"TEXTNAME: ";A4:LPRINT:LPRINT:LPRINT:RETURN
1620 FORP=1TO10:LPRINT:NEXTP:RETURN
1630 IFLEN(K)>L!THENK=STRING$(L!,32)
1640 IFLEN(K)<L!THENK=K+STRING$(L!-LEN(K),32)
1650 RETURN
1660 PRINTCHR$(15);:GOTO1250
1670 PRINT§1339,"Textname-Eingabe:":PRINT§1499,"<< TEXTNAME/TXT >>";:PRINT§1742,STRING$(12,95)+"
  ":PRINT§1742,;:LINEINPUTA4:A4=A4+":5":IFLEFT$(A4,1)="/"THEN70ELSERETURN
1680 HOFF0:HON2:HNEG:CLEAR50:RUN"JOHMENÜ/BAS"
1690 '#----------- Text-Datei öffnen -------------------#
1700 ONERRORGOTO1710:OPEN"O",1,KA:RETURN
1710 RESUME1720
1720 PRINT§5,"Sie haben als Textnameneingabe unerlaubt Zahlen verwendet !";:FORQ=1TO100:NEXTQ:PR
INT§2,CY;CHR$(7);:Q1=Q1+1:IFQ1=>5THEN1730ELSE1720
1730 PRINT§160,"Neuer Textname für die Speicherung:":PRINT§195,STRING$(12,95)+"    ":PRINT§195,;:
LINEINPUTKA:GOTO1700
```

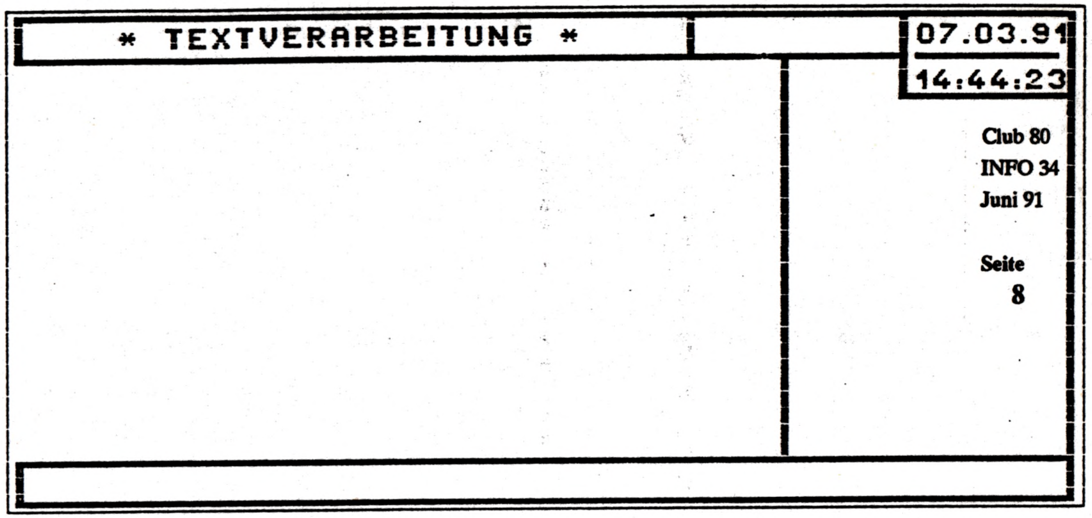


Bild 1: JOHNEN01/RDL

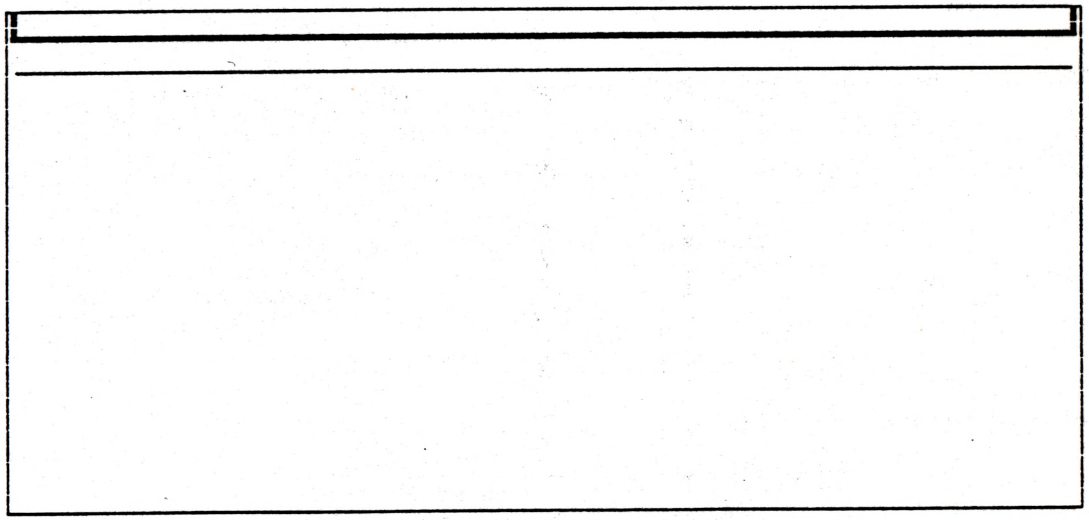


Bild 2: Textbild/RDL

Text-Konvertierung IBM-PC nach CP/M

Helmut Bernhardt

Wer abwechselnd unter CP/M und unter DRDOS (oder womöglich gar MSDOS) Texte schreibt, hat stets Probleme, diese Texte zwischen den beiden Systemen auszutauschen. Das geringere Übel ist dabei, daß die auf dem PC eingegebenen deutschen Umlaute und Grafikzeichen unter CP/M nicht mehr zu erkennen sind; viel schlimmer ist, daß gelegentlich ab einer bestimmten Stelle im Text unter CP/M vermeintlich nichts mehr da ist.

Woran liegt das? -- Daran:

Unter CP/M gibt es den 7Bit-ASCII-Code, mit dem 128 verschiedene Zeichen codiert werden können. Die ersten 32 Codes werden noch für Steuerzeichen für das Terminal, den Drucker oder wen auch immer belegt, so daß mit 96 Codes nicht allzuviele darstellbaren Zeichen unterschieden werden können. So hat man schon für die verschiedenen nationalen Sonderzeichen die Konvention eingeführt, daß deren Codes immer bei 5Bh-5Eh und 7Bh-7Eh liegen. Welches nationale Sonderzeichen mit einem dieser Codes erzeugt werden soll, kann z.B. beim Drucker über ein Mäuseklavier eingestellt werden oder allgemein über eine ESC-Sequenz vorgegeben werden.

Bei der Druckerausgabe ist es noch möglich, durch Umschalten mit entsprechenden ESC-Sequenzen abwechselnd die Sonderzeichen aller nationalen Zeichensätze innerhalb eines Textes zu benutzen. Dann müssen in dem Text-File diese Steuersequenzen enthalten sein und es handelt sich nicht mehr um einen reinen ASCII-Text, der universell übertragbar ist.

Ein Terminal kann aber immer nur eines der mit einem der Codes 5Bh-5Eh und 7Bh-7Eh gemeinten Zeichen darstellen, das dann in allen Fällen auf dem Bildschirm erscheint. Gleichzeitige Darstellung von z.B. [und Ä ist nicht möglich.

Diese Übel wurde beim PC durch den 8Bit-ASCII-Code beseitigt. Das 8te Bit wird dafür verwendet, weitere 128 darstellbare Zeichen zu codieren. Die Codes 5Bh-5Eh und 7Bh-7Eh sind eindeutig den Zeichen [\]^{|}~ zugeordnet und alle anderen nationalen Sonderzeichen haben eindeutige Codes mit gesetztem Bit 7. Daneben gibt es noch etliche weitere Sonderzeichen und auch Grafikzeichen zum Erzeugen von Linien und Rahmen in einfacher, doppelter und gemischter Linienführung.

Da unter CP/M das Bit 7 eines ASCII-Codes nicht definiert ist, benutzen einige Textprogramme und auch Terminals es zur Unterbringung eines Attribut--Bits oder ignorieren es einfach. In diesem Fall werden die PC-Zeichen mit Codes 80h-9Fh als Steuerzeichen 00h-1Fh erkannt (z.B. Wordstar), was im Fall von 'Ü' mit dem Code 9Ah als 1Ah =^Zerkannt und als 'End of File' interpretiert wird. Der Rest des Files hinter dem Ü ist dann nicht zugänglich.

Da mit dem unter CP/M 3.0 laufenden Programm MSDOS.COM, dessen Turbo Pascal Source in dem Artikel 'CP/M 3.0 goes MSDOS' in der c't abgedruckt war, 5¼"-360K-MSDOS-Disketten gelesen und geschrieben werden können, ist ein Transfer von Texten zwischen den Systemen physikalisch möglich. Es muß nur noch eine Anpassung der ASCII-Codes vorgenommen werden, falls das Text-File von DRDOS nach CP/M übertragen werden soll. In anderer Richtung ergibt einen Codeumwandlung wenig Sinn, weil nicht feststeht, welches nationale Sonderzeichen anstelle der eckigen oder geschweiften Klammer gemeint ist, oder ob nicht hier tatsächlich mal eine solche Klammer gemeint ist.

Eine Codeanpassung der Zeichen des PC-Zeichensatzes №2 an den 7Bit-ASCII--Zeichensatz leistet das beiliegende kleine Pascal Programm, das einfache 8Bit-ASCII-Files (von Quick Star, dem Editor von Turbo Pascal, EDITOR.EXE von DRDOS...) umcodiert. Texte von MS-Word, Word Perfect, Star Writer PC, Word Star... lassen sich nicht (solange sie nicht mit einem entsprechenden Utility in reine ASCII-Texte konvertiert wurden) übertragen.

Das Programm läßt sich mit Turbo Pascal unter DRDOS und unter CP/M kompilieren und läuft unter beiden Betriebssystemen. Es werden nicht nur die deutschen Umlaute übersetzt sondern auch die Selbstlaute mit Akzenten in entsprechende Selbstlaute ohne Akzent übersetzt. Die Grafikzeichen werden je nach Form in die Zeichen ! - + übersetzt, wobei man unter CP/M die Linien leidlich erkennen kann, die im PC-Zeichensatz dargestellt sind. Für nicht übertragbare Zeichen wird ein Blank erzeugt.

Tabelle: Zeichen des PC-Zeichensatz №2 und die daraus für den 7Bit-ASCII-Zeichensatz erzeugten Entsprechungen

	8xh	9xh	Axh	BxH	Cxh	Dxh	Exh	Fxh
0	Ç C	É E	á a	░	└ +	╨ -	α a	≡ =
1	ü }	æ ë	í i	▒	┴ -	╤ -	β ~	± +
2	é e	Æ [	ó o	▓	┬ -	╥ -	Γ T	≥ >
3	â a	ô o	ú u	│ !	├ !	╙ +	π	≤ <
4	ä {	ö |	ñ n	┤ !	─ -	╘ +	Σ	⌠
5	à a	ò o	Ñ N	╡ !	┼ +	╒ +	σ	⌡
6	å a	û u	ª a	╢ !	╞ !	╓ +	µ	÷
7	ç c	ù u	º o	╖ +	╟ !	╫ +	τ	≈
8	ê e	ÿ y	¿	╕ +	╚ +	╪ +	Φ	°
9	ë e	Ö \	⌐	╣ !	╔ +	┘ +	Θ	∙
A	è e	Ü]	¬	║ !	╩ -	┌ +	Ω	-
B	ï i	¢ C	½	╗ +	╦ -	█	δ	√
C	î i	£	¼	╝ +	╠ !	▄	∞	ⁿ
D	ì i	¥ Y	¡	╜ +	═ -	▌	ø	² 2
E	Ä [	₧	« <	╛ +	╬ +	▐	∈	■
F	Å A	ƒ	» >	┐ +	╧ -	▀	∩	

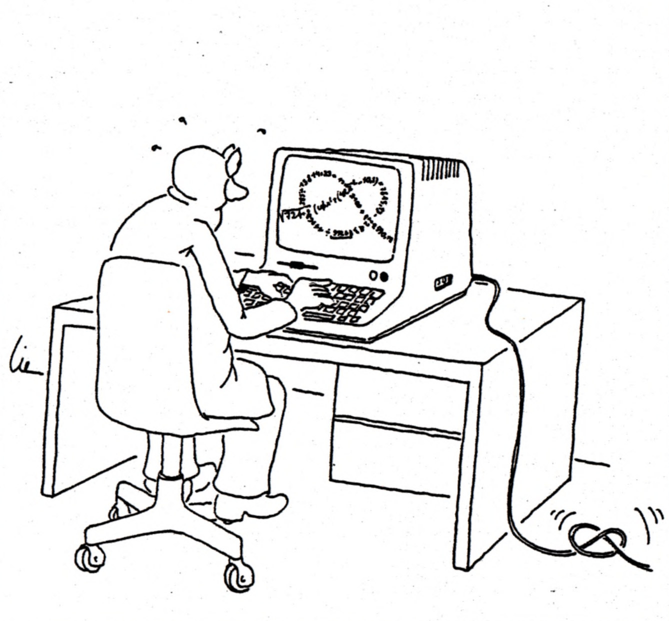
Da diese Umcodierung über eine Tabelle erledigt wird, kann man sich dort noch weitere Stellvertreter-Zeichen für den 7Bit-Zeichensatz einbauen, ohne an der Programmlogik herumschnitzen zu müssen.

```
program PC2CPM;
{uses crt;    nur unter Turbo Pascal 4.0 nötig, kennt sonst kein UPCASE}

const logo : string = 'PC => CPM  Textkonvertierung,  (C) 06.05.91  Be';

      Umcode : array [0..127] of byte =
      ($43,$7D,$65,$61, $7B,$61,$61,$63, $65,$65,$65,$69, $69,$69,$5b,$65,
       $45,$7b,$5b,$7c, $7c,$6f,$75,$75, $79,$5c,$5d,$63, $20,$59,$20,$20,
       $61,$69,$6f,$75, $6e,$4e,$61,$6f, $20,$20,$20,$20, $20,$20,$3c,$3e,
       $20,$20,$20,$21, $21,$21,$21,$2b, $2b,$21,$21,$2b, $2b,$2b,$2b,$2b,
       $2b,$2d,$2d,$21, $2d,$2b,$21,$21, $2b,$2b,$2d,$2d, $21,$2d,$2b,$2d,
       $2d,$2d,$2d,$2b, $2b,$2b,$2b,$2b, $2b,$2b,$2b,$20, $20,$20,$20,$20,
       $61,$7e,$54,$20, $20,$20,$20,$20, $20,$20,$20,$20, $20,$20,$20,$20,
       $3d,$2b,$3e,$3c, $20,$20,$20,$20, $20,$20,$20,$20, $20,$32,$20,$20);
          {Position in der Tabelle ist PC-ASCII-Code abzüglich 128.
           Wert in der Tabelle ist der entsprechende CPM-ASCII-Code.
           Für nicht uebertragbare Zeichen wird ein Blank erzeugt.}

var  PC, CPM               : file of byte;
     Quelle, Ziel          : string;
     Ptr, scan, i, Zeichen : byte;

procedure GetSource;
begin
  repeat
    write('PC-Text-File : ');             readln(Quelle);
    for scan:=1 to length(Quelle) do Quelle[scan] := upcase(Quelle[scan]);
    assign(PC,Quelle);       {$I-}         reset(PC);          {$I+}
  until ioresult=0;
end;

begin
  if ParamCount<>0 then begin
    Quelle := ParamStr(1);     for scan:=1 to length(Quelle) do
       Quelle[scan] := upcase(Quelle[scan]);
    assign(PC,Quelle);       {$I-}      reset(PC);      {$I+}
    if ioresult<>0 then begin
      writeln('kann ',Quelle,' nicht oeffnen!');        GetSource;
    end
  end
  else GetSource;

  Ziel:='';    scan:=1;
  repeat
    Ziel := Ziel + Quelle[scan];            scan := scan+1;
    if Quelle[scan] = '.' then scan:=length(Quelle);
  until scan=length(Quelle);

  Ziel:=Ziel + '.CPM';
  assign(CPM,Ziel);           {$I-}     rewrite(CPM);      {$I+}
  if IOresult <> 0 then begin
    writeln('kann ',Ziel,' nicht anlegen!');      Halt;
  end;

  writeln(logo);
```



```
  repeat
    read(PC,Zeichen);
    if (Zeichen and $80) <> 0 then begin
      Ptr := (Zeichen and $7F);         Zeichen := Umcode[Ptr];
    end;
    write(CPM,Zeichen);
  until eof(PC);
  close(PC);        close(CPM);
  writeln(Quelle,' in ',Ziel,' konvertiert !')
end.
```

Monitor



C-Einführung

von Gerald Schröder und Alexander Schmid

Falls Ihr noch kein C-Listing "gelesen" habt, dürftet Ihr das in dem Artikel über die BNF für Programmierer abgedruckte als recht kryptisch empfinden. Deshalb hier einige Erläuterungen.

Fangen wir oben im Listing an: Kommentare stehen zwischen "/*...*/". Die Dinger dürfen nicht geschachtelt werden! Dann folgt "#include...": damit werden die entsprechenden Dateien hier eingefügt, wie das auch im Turbo-Pascal möglich ist. Alle Befehle, die mit einem "#" beginnen, gehören nicht direkt zu C, sondern sind Anweisungen an den Compiler, irgendwas zu tun. "#define" dient dazu, Zeichenfolgen zu ersetzen, Mit "#define Hugo 1" kann man sich z.B. eine Konstante namens Hugo, definieren, aber auch wesentlich kompliziertere Sachen anstellen, was ggf. in einem anderen Artikel ausführlicher beschrieben wird. Selbstverständlich passiert das nicht in Eurem wertvollen Quellfile, sondern in einem internen Zwischenfile, das der Compiler anlegt. Die Datei "stdio.h" ist nötig, um Ein- und Ausgaben machen zu können. I/O (und Strings) gehören also (wie in Modula-2) nicht zur Sprache, anders als in Pascal.

Nun werden globale Variablen deklariert, und zwar immer in der Form "typ variable;". Wir deklarieren "ch" vom Typ "char", während "line" ein "Array (0..128) of char" ist, was in C sehr viel kürzer ausgedrückt wird. Das letzte Zeichen eines Strings muß immer 00h (bzw. #0) sein, deshalb muß der String immer ein Zeichen größer als benötigt deklariert werden. Als Typen kommen ansonsten hier nur "int" für Integer- und "double" für Real-Zahlen zum Zuge.

Dann folgen "Prototypen" der im Programm deklarierten Funktionen. Das ist dasgleiche wie die "FORWARD"-Deklarationen in Pascal. Funktionen (Prozeduren ohne Rückgabewert gibt es in C offiziell nicht) werden wie Variablen deklariert:
"typ fktname (paramtyp...);"
- "typ" gibt den Rückgabewert der Funktion an
- durch die Klammern werden Funktionen von Variablen unterschieden, also sind sie auch bei Funktionen ohne Parameter nötig
- in den Klammern kann eine Liste mit den Typen der Parameter stehen
- "void" wird in die Klammern geschrieben, falls kein Parameter übergeben wird

"readline" ist die erste Funktion. Sie bekommt als Parameter ein "Array of Char", wobei die Größe egal ist. Die Funktion beginnt mit dem "(". (Davor darf kein ";" stehen!) Dann folgt die Deklaration lokaler Variablen, wobei die Variable "count" auf 0 initialisiert wird. Die in der Funktion "nextchar" deklarierte Variable "index" behält übrigens zwischen zwei Aufrufen ihren Wert, wird also nur beim ersten Aufruf auf 0 gesetzt; dies bewirkt das "static".

Die wichtigste Funktion ist "main": sie wird beim Aufruf des Programms gestartet, bildet also das Hauptprogramm. Funktions-Deklarationen dürfen übrigens nicht geschachtelt werden, also gibt es nur lokale Variablen, aber nicht lokale Funktionen.

Nun zu den Anweisungen:

Funktionen müssen immer mit "fktname(params)" aufgerufen werden. Falls die Funktion keine Parameter erwartet, lautet der Aufruf "fktname()"; "fktname" allein genügt nicht!

"return EXPR" beendet eine Funktion und gibt den angegebenen Wert EXPR zurück.

"if(EXPR) STMT else STMT" ist die IF-Anweisung. In der Klammer steht ein Ausdruck. Wenn der Ausdruck einen Wert <>0 ergibt, wird die erste Anweisung ausgeführt, sonst die nach dem "else". Das "else STMT" kann übrigens auch fehlen. Statt einer (mit ";" abgeschlossenen) Anweisung kann auch ein Block mit mehreren Anweisungen stehen (BEGIN = {", END = "}"). Eine besonders schöne Alternative ist in C mit dem "?"-Operator möglich. EXPR1 ? EXPR2 : EXPR3 liefert den Wert von EXPR2, wenn EXPR1 TRUE ist, andernfalls den Wert von EXPR2. Da es in C auch keine Boole'schen Variablen gibt, nimmt man der Einfachheit halber einfach Integer-Werte. Null bedeutet FALSE, alles andere TRUE. Pascal-Fans können sich hier mit "#define BOOLEAN INTEGER" Erleichterung verschaffen.

"while(EXPR) STMT" ist eine Schleife; ansonsten siehe "if...".

Ausdrücke:
"EXPR == EXPR" vergleicht die beiden Ausdrücke, z.B. "x == y".
"!=" prüft auch Ungleichheit (normalerweise "<>").
"&&" bedeutet soviel wie AND, "||" soviel wie OR.

"VAR++" oder "++VAR" erhöht einfach den Wert der Variablen um eins. In anderen Sprachen heißt das INC(VAR) oder auch nur VAR :=VAR+1.

"VAR *= EXPR" ist eine Kurzform für VAR = VAR * EXPR, genauso geht das mit "-=", "+=" und "/=".

"VARIABLE = EXPR" weist der Variablen einen Wert zu. Das lustige daran ist, daß dies überall auftauchen kann und den Wert ergibt, der zugewiesen wird. Statt "var = x; if(var == y) ..." schreiben wir also: "if((var = x) == y) ...".

"printf(...)" dient zur Ausgabe auf dem Bildschirm, wobei als erster Parameter ein String übergeben wird, der beschreibt, wieviele und welche Parameter noch folgen. In "main()" bedeutet das "\lf", daß als zweiter Parameter eine double-Zahl (long float) erwartet wird, die an der Stelle ausgegeben wird.

"getchar()" holt ein Zeichen von der Tastatur.

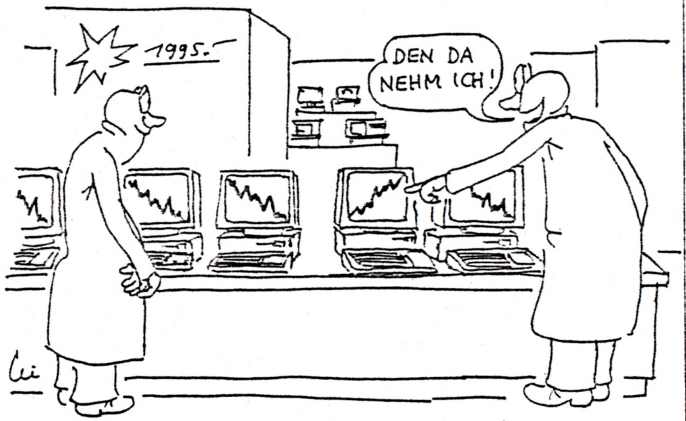
"\n" steht für "Newline", also das Carriage Return.

"\0" steht für den "Null"-Character also das ASCII-Zeichen mit dem Code 0 und zeigt das Stringende an, das in C so markiert wird.

"EOF" ist eine symbolische Konstante, Funktionen, die Dateien lesen, liefern diesen Wert, wenn das Dateiende erreicht ist.

Ansonsten: Nicht verzweifeln, wenn Euer Programm nicht läuft. Der Compiler prüft so gut wie gar nichts, d.h. fast alles, was Ihr ihm vorsetzt, baut er irgendwie zu einem Programm zusammen. Nur was das dann tut, ist die Frage...

Noname



BNF für (wahre) Programmierer

von Alexander Schmid und Gerald Schröder

In einem der letzten Infos war ein Bericht über die BNF (Backus Naur Form). Nun wollen wir doch mal sehen, was ein echter Programmierer damit anfangen kann, außer seine heißgeliebte Programmiersprache besser zu verstehen oder unbedarfte Gemüter schocken.

Wenn wir irgendwas in BNF beschrieben haben, können wir daraus sehr leicht ein Programm entwickeln. Als Beispiel haben wir diesmal einen kleinen Taschenrechner gebaut. Da ein Rechner keinen Grips hat, müssen wir ihm die Regeln, was er wie machen soll ganz genau vorkauen. Das sieht in BNF so aus:

```
Expression = Term {AddOp Term}
AddOp      = '+' | '-'
Term       = Faktor {MulOp Faktor}
MulOp      = '*' | '/'
Faktor     = [Sign] Number | '(' Expression ')'
Sign       = '+' | '-'
Number     = {Digit} ['.' {Digit}]
Digit      = '0' .. '9'
```

Damit wäre z.B. dies ein Expression: -5 * 7 + (4 - 2)
Wenn wir nun ein Programm zum Berechnen so eines Ausdrucks schreiben wollen, ist das unheimlich einfach: die Namen auf der linken Seite der Regeln werden Funktionsnamen, wenn die Namen auf der rechten Seite auftauchen, so werden die entsprechenden Funktionen aufgerufen. Wenn Ihr die Funktionen und die Definitionen in der BNF nebeneinander legt, werdet Ihr die Umsetzung leicht nachvollziehen können.

Das ganze sieht in C, der Sprache für echte Programmierer, dann so aus (hier in ANSI-C; Kommentare sind nicht nötig, da der Code auch so selbstdokumentierend ist):

```
/* Copyright Gerald Schröder & Alexander Schmid    */
/* Formelparser für arithmetische Ausdrücke        */
/* 10.05.91  Clubtreffen des Club 80               */


#include <stdio.h>

#define MAXLEN 128

char   ch, line[MAXLEN+1];
double expr(void);
double term(void);
double faktor(void);
double number(void);
```

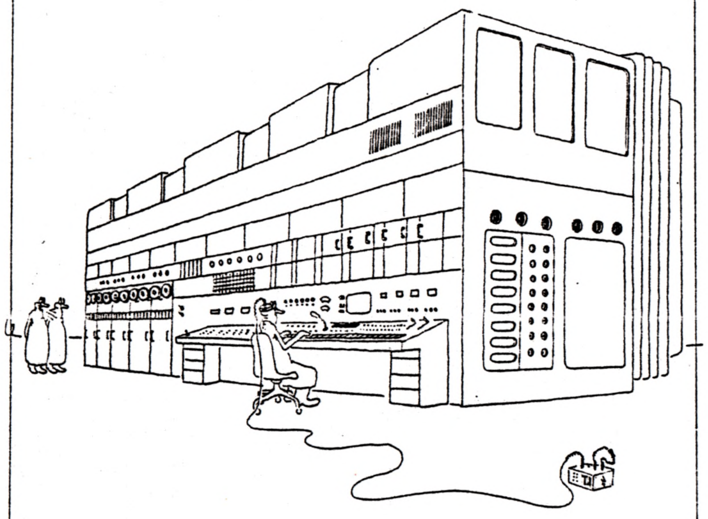
```
void readline(char inputline[])
{
  int ch,count=0;

  while((ch=getchar()) != EOF && ch != '\n' && count<MAXLEN)
    inputline[count++] = ch;
  inputline[count] = '\0';
  }

void nextchar(void)
{
  static int index=0;
  while((ch = line[index++]) == ' ');
  }

double expr(void)
{
  int    addop;
  double term1,term2;

  term1 = term();
  while(ch=='+' || ch=='-'){
    addop = ch=='+';
    nextchar();
    term2 = term();
    term1 = addop ? term1+term2 : term1-term2;
    }
  return term1;
}

double term(void)
{
  int    mulop;
  double fak1,fak2;

  fak1 = faktor();
  while(ch=='*' || ch=='/'){
    mulop = ch=='*';
    nextchar();
    fak2 = faktor();
    fak1 = mulop ? fak1*fak2 : fak1/fak2;
    }
  return fak1;
  }

double faktor(void)
{
  int    minus;
  double express;

  minus = ch=='-';
  if(minus || (ch=='+')) nextchar();
  if(ch=='('){
    nextchar();
    express = expr();
```



```
    nextchar();
    }
  else
    express = number();
  return minus ? -express : express;
  }

double number(void)
{
  double akku=0;

  while(ch>='0' && ch<='9'){
    akku = 10 * akku + (ch-'0');
    nextchar();
    }
  if(ch=='.'){
    double divi=10;
    nextchar();
    while(ch>='0' && ch<='9'){
      akku = akku + (ch-'0')/divi;
      divi *= 10;
      nextchar();
      }
    }
  return akku;
  }

main()
{
  printf("Eingabe : ");
  readline(line);
  nextchar();
  printf("Ausgabe : %lf\n",expr());
  }
```

Erstellt auf einem Z280-Rechner unter CP/M Plus mit Hi-Tech C. Muß aber auch auf einem (Domes)DOS-WC z.B. mit Turbo-C laufen (ANSI-C). Wer nur einen Compiler nach dem K&R-Standard hat, muß die ganzen 'void' und die Funktions-Prototypen einfach wegstreichen, dann sollte es auch damit gehen.

Operator
»Ist es nicht tröstlich, zu wissen, daß hinter alldem letztlich der Mensch steht?«



PRODUCER (Version 3.0, deutsch ...)

Die deutsche Version des Programmgenerators "PRODUCER" läuft nun auch unter Genie-DOS und erstellt Euch, nach ca. einer halben Stunde, eine komplette Dateiverwaltung (Ja, auch in deutsch) mit Bildschirmeditor, Hilfsfenstern, BTREE-Feilhändling und anderen wundersamen Funktionen, nachzulesen in mehrseitigen Anzeigen der 80 MICRO ... schaut Euch die (hoffentlich) nachfolgend abgedruckten Hardcopy's an, schickt mir zwei formatierte 40-Spur Disketten und Rückporto und hört auf, mit FIELD, GET, PUT und CMD"O" herumzuspielen, löscht auch sämtliche Maskengeneratoren von Euren Disketten ... (meinetwegen auch den "CREATOR")
Zwei Demonstrationsprogramme sind auch dabei ... Handbuch nicht!

Jens Günther

The Producer™

Ein PROGRAMM-ENTWICKLUNGS-SYSTEM 1982 von ROGER SMITH

Deutsche Version
& Bearbeitung 1990
von Jens Günther

Drücke: <A> oder <B>
für gewünschtes Menü

<A>	<B>
<1> Plannungsliste	<1> Maske erstellen
<2> Programm-Name	<2> Ändere Quellcode
<3> Referenzliste	<3> Generiere Zeilen
<4> Pflege Dateien	<4> Erzeuge Reports
<5> Weiteres Menü	<5> Erzeuge Programm
<6> Hilfsprogramme	<6> Rückkehr ins DOS

PRODUCER: B1-BILDSCHIRM-GENERATOR
ERSTELLE / DEFINIERE EINGABEMASKE

<1> ANZEIGEN eines Buffers
<2> LADEN eines Buffers von Disk
<3> LÖSCHEN ausgewählter Buffer
<4> ZUKÜNFTIGE Erweiterung
<5> SPEICHERE Buffer auf Disk
<6> ÄNDERN: Maskendefinition
<7> KOPIERE Buffer auf Buffer
<8> HILFE: Zeige EDITOR-Befehle
<9> RÜCKKEHR ins Hauptmenü

FUNKTION DURCH ZAHL WÄHLEN....>

BUFF# SPEICHER STATUS

BUFF#	SPEICHER	STATUS
1	55009	LEER
2	56033	LEER
3	57057	LEER
4	58081	LEER
5	59105	LEER
6	60129	LEER
7	61153	LEER
8	62177	LEER
9	63201	LEER

PRODUCER!

CTL Q=Sichern auf Buffer 9 & ENDE

CTL C=Zentriere Text (CTL C am
Textanfang & Ende)

CTL B=Erzeuge BASIC-Zeilen

CTL G=Graphik-Zeichen CHR$(191)
Mit Zahlentasten 89
gewünschte Pixels 56
ein-/ausschalten 23

CTL R=Zeichen unter Cursor
mit PFEILTASTEN bewegen

CTL S=Einfügen eines Leerzeichens

CTL D=Lösche Zeichen am Cursor

CTL Z=Cursorposition anzeigen

CTL F=Anzeige des § Symbols

CTL E=Löscht den markierten Block

CTL O=Löscht die Blockmarkierungen

CTL L=Riesenbuchstaben AN/AUS

CLEAR=Lösche Maske ab Cursor

PFEILTASTEN=Bewege den Cursor

SHIFT PFEIL-Rechts=Cursor zum
rechten Bildschirmrand

SHIFT PFEIL-Links=Cursor zum
linken Bildschirmrand

SHIFT PFEIL-Hoch=Cursor oben

ENTER=Cursor in nächste Zeile

§=Beginnt Masken-Definierung

CTL K=Einfügen einer Zeile

CTL N=Lösche Zeile am Cursor

CTL T=Block kopieren/bewegen
Links oben & Rechts unten
mit CTL T markieren, dann
Cursor auf Ziel & CTL T!

The Producer

Ein PROGRAMM-ENTWICKLUNGS-SYSTEM

A6: Diskettenoperationen durchführen

Wähle Funktion!
Tasten: <1> &
<0> für RÜCKKEHR
ins Hauptmenü!

<1> Disketteninhalt
<2> Freier Diskplatz
<3> Disketten-Backup
<4> Formatiere Disk
<5> Kopiere Programm
<6> Programmtransfer
<7> Erstelle RUN-Disk
<8> Lösche Programme
<9> Zukünftige Funktion
<0> RÜCKKEHR Hauptmenü

1 PRO/CLC	2 PRO/L1	3.PRO0D	4.PRO12D
5.PRO13D	6.PRO1D	7 PRO1D/CTR	8.PRO2D
9.PRO3D	10.PRO4AD	11.PRO4D	12 PRO5D
13.PRO6D	14.PRO7D	15.PROD/JOB	16 TMLIC/CMD

<#> Markiere/Demarkiere Nummer   <M> Eingeben der Maske

<E> ENDE und Übertragung   <X> ENDE OHNE Übertragung

<N> Nächste Seite anzeigen   <*> Alles Demarkieren >

BIBLIOTHEK

Deutsche Version des >Reader's Guide< geschrieben mit dem
PRODUCER Programm-Entwicklungs-System am
24.12.90 von: Jens Günther

Programm-Menü

<1> Datensätze hinzufügen
<2> Daten ändern/anzeigen
<3> Report-Druck-Menü
<4> Optionenmenü aufrufen
<5> Beenden des Programms

Dateistatus

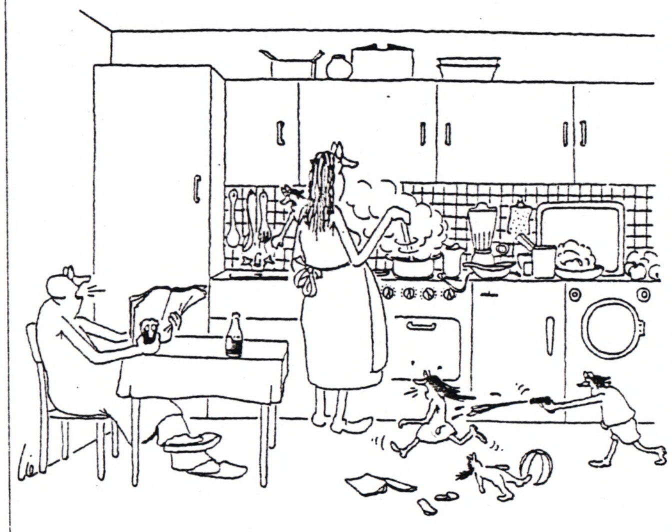
Zugriff letzter Satz->0
Aktive Datensätze---->1
Gelöschte Datensätze->1
Dateiname----->BIBLIO/DAT
Fehlerstatus-->

Bitte FUNKTION mit ZAHL wählen:

```
1.Schlagwort---->...............
2.Titel--------->                         5-7.Datum->  /  /
3.Autor--------->                         8.Ausgabe->
4.Publikation--->                         9.Seite--->

10.Report-Titel->                     11.Report-Code-->
11.Report-Text-->
12.Beschreibung1>
13.Beschreibung2>
    System-Meldungs-Bereich

    Hilfstext-Bereich
Geben Sie ein Schluesselwort oder einen Text ein, der diesen
Artikel bzw. dieses Buch beschreibt.
```

```
     Ihre Datei wird nun bearbeitet, eine Liste
der Sätze wird zur Wiedereinfügung optimiert      BEARBEITE
und nach dieser Liste wieder verbunden.
Sie können während der BEARBEITE und OPTIMIERE    OPTIMIERE
Sektion den Vorgang durch <§> abbrechen ohne
Ihre Datei dabei zu zerstören.                    VERBINDE

Anzahl Datensätze-->         Zugriff auf Satz->
Verbleibende Sätze->         Text>

Satz zum Verbinden
     Satz #>
Schlüssel 1>
Schlüssel 2>
```


PC
»Ich befürchte, daß du durch diese neuen PC's eines Tages ohne Arbeit dastehst . . .«



..Hardware ..Hardware ..Hardware

Club-80-Terminal auf Basis der PC-Hercules-Karte

Helmut Bernhardt, Stefan Nitschke 29.05.91

Nun ist es schon wieder ein Jahr her, daß auf dem Club-Treffen 1990 der Bau einer Terminalkarte, die sich des günstigen Leistungs-/Preis-Verhältnisses der Herculeskarte für den PC bedient, beschlossen wurde. Nachdem ein erfolgsversprechendes Konzept stand, der Hardware-Koordinator noch schriftlich um Stellungnahme gebeten wurde (die bislang immer noch nicht vorliegt), auf diese Stellungnahme artig ein paar Wochen gewartet wurde, haben wir das damalige Konzept für das Projekt an Jens Neueder abgeschickt. Wahrscheinlich erscheint es zusammen mit diesen Beiträgen irgendwann in den nächsten Jahren mal in einem Info.

Da wir nicht warten wollten, bis eventuell mal ein Info erscheint und dann eventuell doch noch mehr Mitstreiter bei dem Projekt helfen, haben wir schon mal angefangen und eine lauffähige erste Version fertiggestellt. An dem ursprünglichen Hardwarekonzept hat sich inzwischen nichts geändert und auch das, was als Software bislang läuft, entspricht dem TVI950-Standard (mit den paar zusätzlichen und sehr nützlichen Features wie z.B. Program-Download).

Es bleibt noch die Einbindung der Tektronix 4014 Grafikbefehle, desDruckertreibers, eines Hardcopy-Befehls und einiger zusätzlicherBefehle (u.a. Snapshot: Abspeichern des Bildschirminhalts auf Disk). Diese recht umfangreichen Aufgaben stehen immer noch an.

Zusätzlich haben wir uns überlegt, daß man die 8 Textpages der Herculeskarte dazu benutzen kann, um damit Window-Technik zu realisieren. Durch einen Befehl werden dem Terminal die Koordinaten des oberen linken Punktes sowie die Anzahl Zeilen und Spalten im Fenster mitgeteilt. Das Terminal kopiert daraufhin die derzeitige Textseite in die näschste, löscht darin den angegebenen Fensterbereich und positioniert den Corsor in der oberen linken Ecke des neuen Fensters. Der Befehl 'Fenster löschen' schaltet einfach wieder zurück auf die vorherige Textseite. Nötige Statusinformationen werden in Window Control Blocks für jede Fensterebene geführt, so daß nach dem Entfernen eines Fensters der vorherige Bildschirm mit allen Einstellungen (Attribut, Cursor) wieder restauriert wird. Das ist aber zur Zeit noch im Entstehen.

Das Terminal ist ein Z80-Computer mit 32K Arbeitsspeicher und 32K Hercules-Video-RAM. Nach dem Einschalten besteht der Arbeitsspeicher beim Lesen aus EPROM und beim Schreiben aus RAM. Durch ein LDIR wird der EPROM-Inhalt bei der Initialisierung ins RAM kopiert. Danach wird das EPROM abgeschaltet und beim Lesen und Schreiben in den unteren 32K erreicht der Z80 das RAM.

Die oberen 32K des Z80 sind dem Video-RAM der Hercules vorbehalten. Hier können die beiden 32K HRG-Pages der Hercules bzw. das Text-/Attribut-Video-RAM eingeblendet werden. Im Textmodus bietet die Hercules-Karte 8 linear hintereinander liegende Seiten mit je 4K Text-/Attribut-Speicher. Innerhalb einer Textseite stehen die geradzahligen Adressen als Textspeicher und die ungeradzahligen Adressen als Speicher für die zugehörigen Attribute zur Verfügung. Jeweils zwei Speicheradressen stehen für eine Druckposition auf dem Bildschirm.

Die Bits eines Attributbytes haben folgende Funktion:

```
    D7 = 0  nicht blinkend bzw. normaler Hintergrund
    D7 = 1  blinkend bzw. halbheller Hintergrund
              (siehe D5 des Display Mode Control Ports B8h)
    D6-D4 = 0, D2-D0 = 1  normale Darstellung
    D6-D4 = 1, D2-D0 = 0  inverse Darstellung
```

Das gilt aus der Sicht des Z80, der den Speicher der Hercules mit Zeichen und Attributen zu befüllen hat. Der CRTC6845 auf der Hercules hat während der Darstellung eines Zeichens (bzw. der Dotzeile eines Zeichens) gleichzeitig das Zeichen und Attribut auszulesen und und während des Shiftens der Bits zum Videosignal die Informationen des Attributs auf das Videosignal wirken zu lassen. Für den 6845 ist eine Textseite in 2K-Worten organisiert - das ist bei der Berechnung des Inhaltes des Scan-Offset-Registers des 6845 zu berücksichtigen.

Im I/O-Adreßraum des Terminal-Z80 sind die Hercules-internen Ports, das Interface zum Host-Computer, ein Latch zur Einstellung der Konfiguration des Terminals und ein Tastatur-Interface unterzubringen. Zur Vereinfachung der Hardware scheinen folgende Portadressen die sinnvollsten zu sein:

```
  00H        Status-Register des Interface zum Host-Computer
  01H        Daten-Register des Interface zum Host-Computer
  02H        Konfigurations-Latch
  03H        Tastatur-Datenregister
  B0H-BFH    interne Ports der Hercules-Karte
    B4H      Adressregister des CRTC-6845
    B5H      Datenregister des CRTC-6845
    B8H      Display-Mode-Control-Port der Hercules
               D1 =0 : Text-Video-Mode enabled, HRG disabled
                  =1 : HRG enabled, Text-Video disabled
               D3 =0 : die Hercules gibt kein Video-Signal an den
                       Monitor
                  =1 : das Video-Signal der Karte gelangt zum
```

```
            D5 =0 : D7=1 des Attribut-Bytes bedeutet "heller
                    Hintergrund"
               =1 : D7=1 des Attribut-Bytes bedeutet "Blinken"
            D7 =0 : HRG : Page 0 selected
               =1 : HRG : Page 1 selected
B9H         Display-Status-Kanal (nach mc)
BAH         Display-Statusport  (nach Computer 2000)
            D0 =1 : horizontaler Zeilenrücklauf
            D3 =1 : Video-Echo, Signal zum Monitor
            D7 =0 : vertikaler Strahlrücklauf
BBH         Light-Pen FlipFlop zurücksetzen
          wird kaum noch in Hardware ermöglicht
BCH         Drucker-Datenport  (Schreiben)
BDH         Drucker-Statusport (Lesen)
            D3 =0 : Error
            D4 =1 : Selected & Ready
            D5 =1 : Paper empty
            D6 =0 : (für 5 Microsecs) /ACK low aktiv
            D7 =1 : Ready high aktiv
BEH         Drucker-Steuerkanal
            D0 =0 : /STROBE, mindestens 5 Microsecs
            D1 =1 : Linefeed
            D2 =0 : Reset Printer, mindestens 50 Microsecs
            D3 =1 : Select, Drucker freigegeben
            D4 =1 : IRQ7 (high aktiver Interrupt) freigegeben,
                    wenn der Drucker ein Byte übernommen hat
BFH         Konfigurations-Schalter
            D0 =1 : HRG erlaubt, gilt nur für Page 1
            D1 =1 : auch HRG-Page 2 erlaubt
```

Das Konfigurations-Latch (Port 02h) unterstützt bislang folgende Befehle:

```
 OUT  (02H),0  Einschalten des Text-/Attribut-Speichers im Text-
               modus bzw. der HRG-Seite 0 im Grafik-Modus
 OUT  (02H),1  Einschalten der HRG-Seite 1; im Text-Modus ist der
               Text-Speicher nicht verfügbar
 OUT  (02H),2  im Bereich 0-32K ist bei Leseoperationen das EPROM
               eingeblendet
 OUT  (02H),3  sowohl beim Lesen als auch beim Schreiben liegt im
               Bereich 0-32K RAM vor
 OUT  (02H),5  und danach
 OUT  (02H),4  Zurücksetzen des Interface der PC-XT-Tastatur
```

Die Ausgabe der Werte 6-15 kann für weitere Konfigurations-Einstellungen benutzt werden. D0 gibt dabei jeweils den Pegel einer entsprechenden Steuerleitung wieder (Q3-Q7 des 74LS259-Latch) und D1-D3 geben vor, welche Steuerleitung geschaltet werden soll. Auf der Karte ist noch ein Lochrasterfeld frei, in dem entsprechende Hardware-Erweiterungen untergebracht werden können. U.a. scheint es zweckmäßig zu sein, dort noch Rabatz-I/O-Hardware unterzubringen, die über Q3 (mit den Steuercodes 06H/07H) zu löblichem Tun angeregt wird.

Zur Vereinfachung der Kommunikation zwischen Host und Terminal wird am Terminal-Z80 ein Interrupt erzeugt, wenn der Host ein neues Datum ausgegeben hat. Diese Technik erfordert in der Software-Entwicklung zwar etwas mehr Sorgfalt, sie ermöglicht aber auch, daß der Z80 des Terminals bei Konjunkturflauten mit anderen sinnvollen Aufgaben beschäftigt werden kann und zwischendurch mal eben ein Byte vom Host übernehmen und ins Fernsehen bringen kann.

Als Tastatur kann wahlweise eine 8Bit-ASCII-Paralleltastatur oder eine serielle PC-XT-Tastatur verwendet werden. Für den Anschluß einer 8Bit-ASCII-Paralleltastatur besteht ein 74LS244 Lesetreiber. Der /Strobe-Eingang der Tastatur ist direkt an den /NMI-Pin des Z80 angeschlossen. Die Übernahme von Tastatur-Daten erfolgt in Echtzeit im Hintergrund. Alle Tastatureingaben werden in einem 32Byte-Ringpuffer zwischengespeichert. Der Ringpuffer wird in der Hauptprogrammschleife ständig überwacht und bei Bedarf werden die Tastatureingaben an den Host entsorgt. Dadurch ist gewährleistet, daß keine Tastatureingaben verlorengehen, und daß bei der Übertragung sonstiger Daten (z.B. Bildschirmdump) an den Host nicht Tastatureingaben dazwischengeraten. Während solcher Jobs befindet sich der Z80 nicht in der Hauptprogrammschleife.

Alternativ kann auch eine PC-XT-Tastatur bzw. eine in den XT-Modus geschaltete MF2-Tastatur angeschlossen werden. Diese Teile geben keine ASCII-Daten sondern Scancodes aus. Solch ein Scancode wird gesendet, wenn eine Taste gedrückt und auch wenn eine Taste losgelassen wurde. Make- und Break-Codes einer Taste unterscheiden sich dabei nur durch D7 des übertragenen Bytes.

Die Codes der PC-Tastatur werden seriell mit TTL-Pegeln übertragen. Zur Synchronisation liefert die Tastatur auch ein CLK-Signal. Auf der Terminal-Karte besteht ein Seriell/Parallel-Wandler, der selbständig die 8 Bit der Tastatur sammelt und nach dem achten Bit einen /NMI erzeugt. Der Z80 kann dann über den Port 03H den Scancode der Tastatur als 8Bit-Wort auslesen und durch aufeinanderfolgende Ausgabe der Werte 4 und 5 an Port 02H den Seriell/Parallel-Wandler für die nächste Tastatureingabe zurücksetzten.

Die /NMI-Serviceroutine führt Buch über den Zustand der Tasten mit Umschaltfunktion und erzeugt mit diesen Informationen aus den Scancodes der normalen Tasten ASCII-Codes, die im Ringspeicher abgelegt werden.

Wenn eine PC-Tastatur angeschlossen wird, braucht der 74LS244 nicht bestückt zu werden, und wenn eine Paralleltastatur zum Einsatz bestimmt, ist kein 74LS322 nötig. Durch den Jumper JP2 wird vorgegeben, welcher Baustein aktiv sein soll:

```
JP2  1-2 gebrückt: Interface für eine PC-Tastatur
     2-3 gebrückt: Interface für Paralleltastatur
```

Bei Anschluß der PC-Tastatur muß außerdem JP3 gesteckt werden, damitam Z80 ein /NMI ankommen kann.

Das Terminal ist mit dem Host über eine bidirektionale parallele Schnittstelle verbunden. Für den Z80 des Terminals liegen die Adressen der Ports des Interface fest (00H = Status-Register; 01H = Daten-Register). Durch Lesen des Satus-Registers kann der Terminal-Z80 feststellen, ob er Daten entgegennehmen kann/muß bzw. ausgeben kann bzw. muß:

D0=0 : es kann/muß ein weiteres Byte an den Host ausgegeben werden (der Host hat das letzte Byte abgeholt)
=1 : es darf noch kein weiteres Byte an den Host ausgegeben werden
D7=0 : es liegt kein neues Byte vom Host vor
=1 : der Host hat ein neues Byte ausgegeben, das abzuholen ist

Die Portadressen, unter denen das Terminal für den Host-Computer erreichbar sein soll, lassen sich über Jumper in Schritten von 4 beliebig einstellen. Die Basisadresse X+0 ist das Status-Register. D6 des Status-Registers zeigt an, ob das Terminal ein weiteres Byte annehmen kann:

D6=1 : das Terminal kann ein weiteres Byte annehmen
=0 : es darf noch kein weiteres Byte an das Terminal ausgegeben werden

D7 gibt an, ob das Terminal bei der Ausgabe von Daten an den Host ein neues Byte bereitgestellt hat:

D7=1 : es steht kein neues Byte des Terminals bereit
=0 : das Terminal hat ein neues Byte ausgegeben

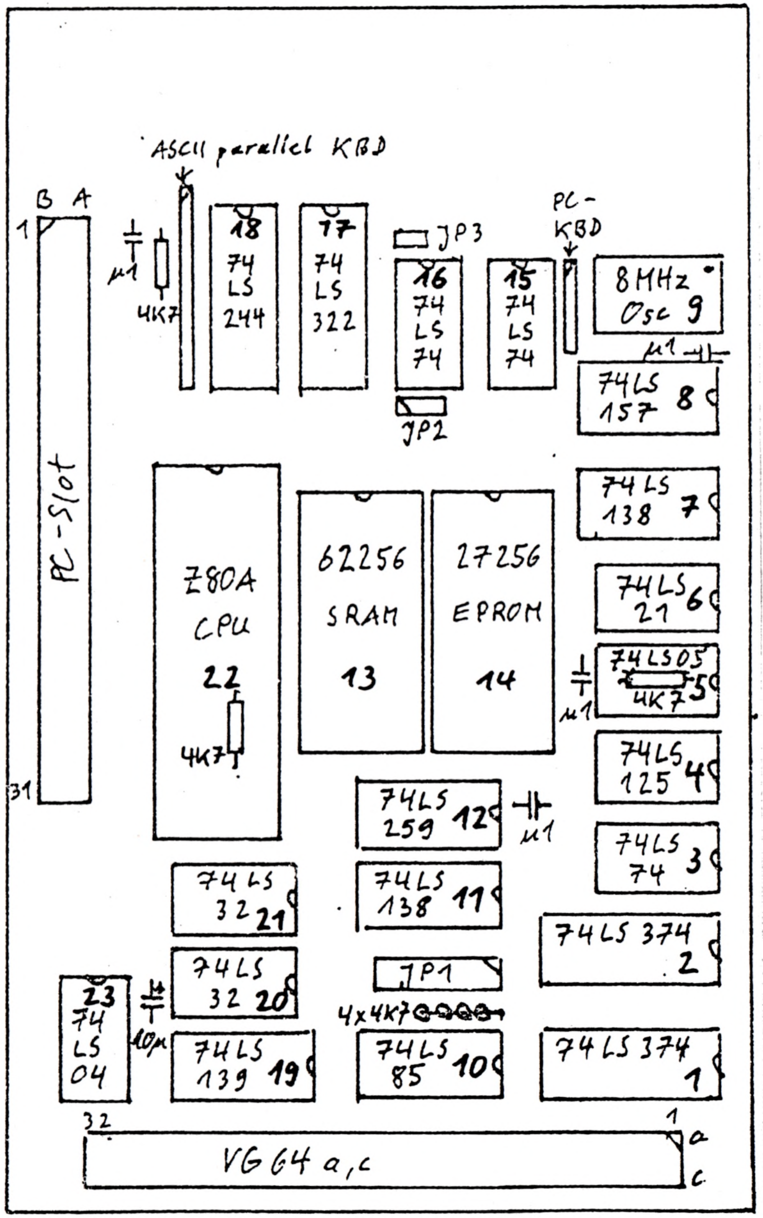
Je nach Zustand der Bits 6 und 7 des Status-Ports kann die CPU an den Daten-Port X+1 ein weiters Byte ausgeben bzw. von dort ein weiteres Byte abholen.

Über das Jumperfeld JP1 wird die Portadresse des Terminals eingestellt. Die Jumper haben folgende Funktion:

1 - 2 offen:	A7 der Portadresse ist high
gesteckt:	A7 der Portadresse ist low
3 - 4 offen:	A6 der Portadresse ist high
gesteckt:	A6 der Portadresse ist low
5 - 6 offen:	A5 der Portadresse ist high
gesteckt:	A5 der Portadresse ist low
7 - 8 offen:	A4 der Portadresse ist high
gesteckt:	A4 der Portadresse ist low
11 - 9 gesteckt, 11 -13 offen:	A2 der Portadresse ist high
11 -13 gesteckt, 11 - 9 offen:	A2 der Portadresse ist low
12 -10 gesteckt, 12 -14 offen:	A3 der Portadresse ist high
12 -14 gesteckt, 12 -10 offen:	A3 der Portadresse ist low

A1 wird nicht decodiert; die Ports X+0 und X+1 sind in den Adressen X+2 und X+3 gespiegelt.

Um kompatibel zur GRIP von Conitec zu sein, müssen die Portadressen C0h un C1h decodiert werden, dafür sind folgende Jumper zu setzen:
5-6, 7-8, 11-13, 12-14.

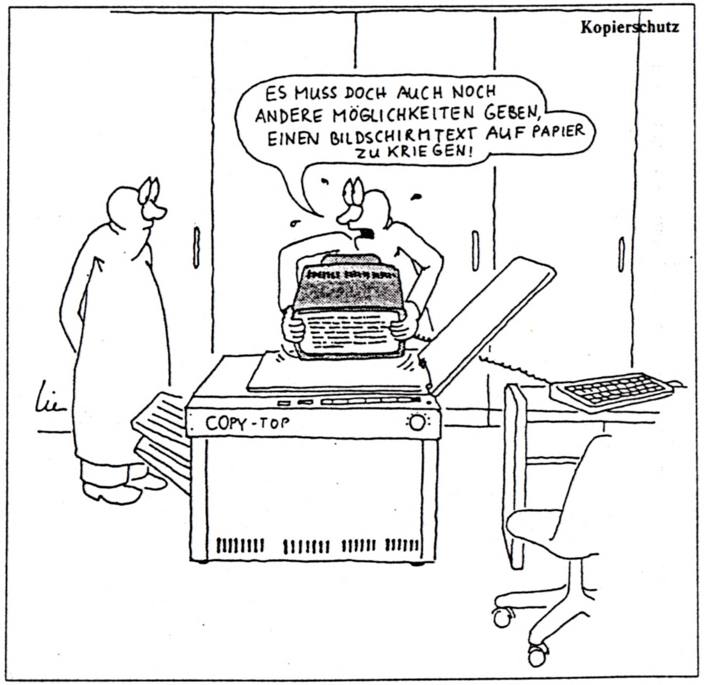


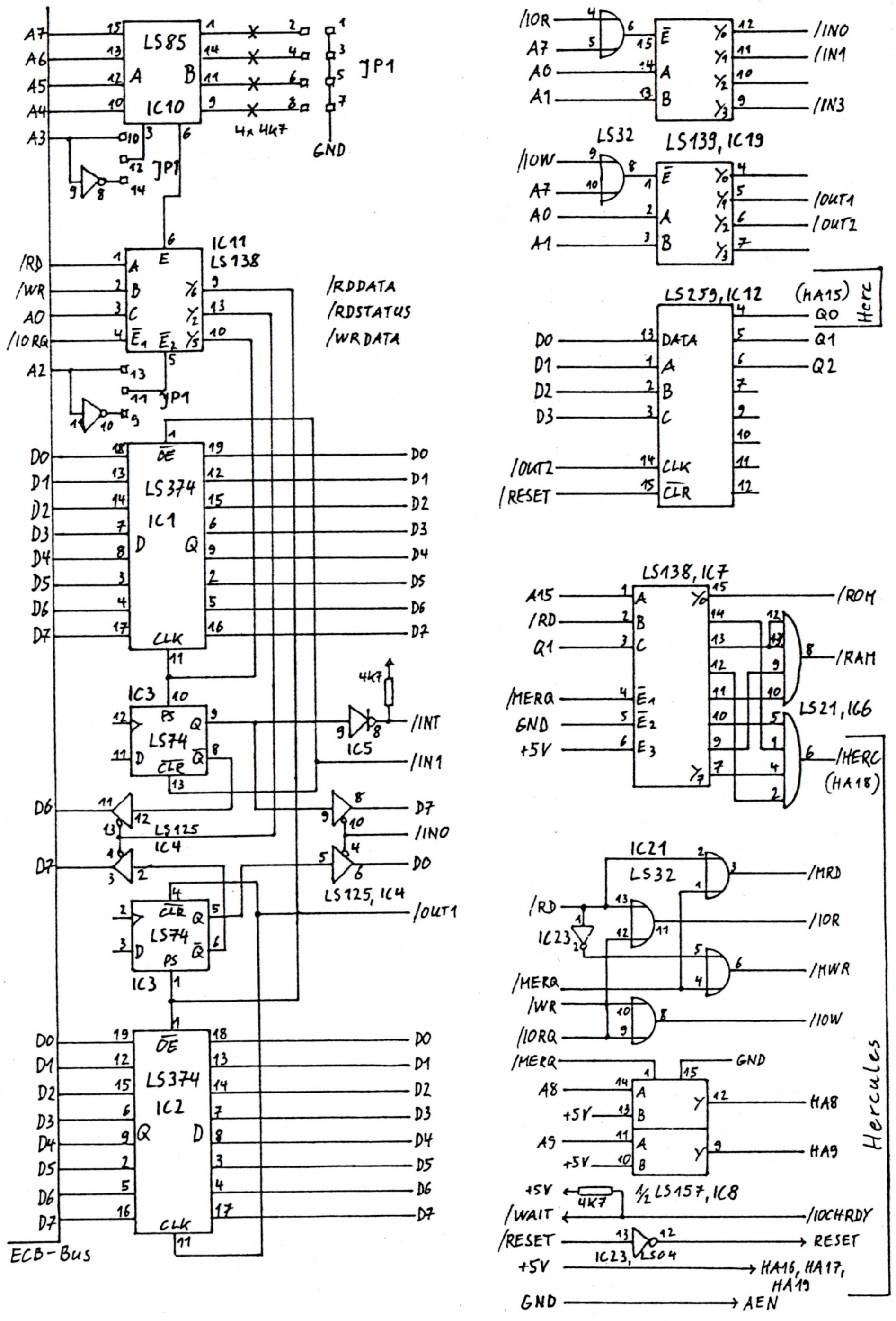
Der Aufbau der Karte

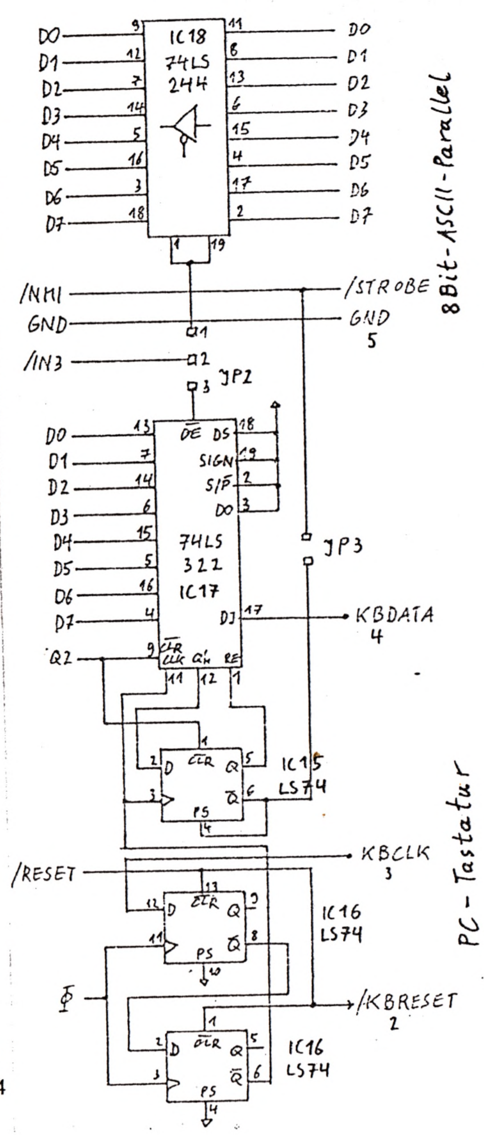
ist wie bei allen Bastlerlösungen etwas komplizierter als bei industiell gefertigten, durchkontaktierten Karten. Wer schon mal eine doppelseitig geätzte Karte selbst durchkontaktieren mußte, weiß ein Lied davon zu singen. Da kann es durchaus einfacher sein, auf einem einseitigen Board die nötigen Querverbindungen in Drahtbrücken anzulegen. Bei dieser Karte ist beides möglich (auch gemischt). Entweder verwendet man die Leiterbahnen auf der Bestückungsseite direkt und legt die nötigen Durchkontaktierungen an, oder man benutzt sie nur als Hinweis dafür, wo Drahtbrücken gezogen werden müssen. Das Layout der Bestückungsseite weist keine sich überkreuzenden Drahtbrücken auf (was im Layout auch schlecht ohne Kurzschluß möglich ist); wo die Leiterbahnen nicht zu eng beieinander liegen, kann auch unisolierter Draht verwendet werden. Bei den etwas dicker geäzten Leiterbahnen sollte als Brücke Klingeldraht und bei den dünnen Leiterbahnen dünner Kupferlackdraht verwendet werden (wenn man lieber Drahtbrücken legen möchte).

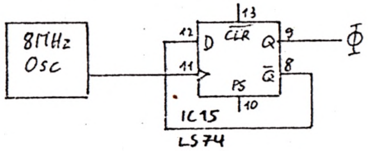
Für den Anschluß der Hercules-Karte benötigt man einen PC-Slot-Stecker. Die Verbindungen zur Terminal-Karte stellt man mit isolierter, flexibler dünner Litze her. Mit Leitungslängen von bis zu 5cm entstehen dann noch keine Probleme auf dem Bus des Terminal-Z80 und es bleibt genügend Bewegungsfreiheit für die Montage der Hercules-Karte in Sandwich-Technik.

Kopierschutz









Schaltungs-Auszüge

links:
Portdecodierung der Terminalkarte und ECB-Bus-Interface

Mitte:

Erzeugen interner I/O-Freigabesignale

Konfigurations-Latch (LS259)

Erzeugen der memory mapped Freigabesignale

Busanpassung für PC-Slot

rechts:

Interfaces für wahlweise 8Bit-ASCII-Parallel-Tastatur oder IBM-PC-Tastatur

Systemtakt-Erzeugung

Erweiterung des ECB-Interface des Hercules-Terminals auf 4 Portadressen für die gesonderte Druckerausgabe

Helmut Bernhardt, Stefan Nitschke

Für die gleichzeitige Ausgabe von Daten für die Console und den Drucker über das gleiche Interface ohne besondere Unterscheidungsmöglichkeiten des Terminals anhand irgendwelcher Statusbits lassen sich grundsätzlich Probleme erwarten.

Es wäre möglich (wie bei der GRIP) durch ein gesetztes D7 im Datum dem Terminal zu signalisieren, daß dieses Byte für den Drucker gedacht ist. Damit ist aber die Möglichkeit verbaut, einen Drucker mit IBM-Zeichensatz 2 voll zu nutzen.

Hier soll aus der Not eine Tugend gemacht werden und die unvollständige Decodierung des ECB-Interface auf 4 Portadressen für unterschiedliche Ausgabeports für Console und Drucker genutzt werden. Die Ausgabe an die Console erfolgt nach wie vor über die Adressen

Basis + 0 : Consolen-Status (Lesen)
Basis + 1 : Console Daten (Lesen und Schreiben)

und der Drucker ist über die Adressen

Basis + 2 : Drucker-Status D7=0: Ready (Lesen)
Basis + 3 : Drucker Daten (Schreiben)

erreichbar. Physikalisch bleiben die Latches für Consolen- und Drucker-Datenport identisch. Das Terminal kann nun anhand von D6 im Statusport des Interface erkennen, ob das Datum an die Console oder an den Drucker ausgegeben wurde.

D6 in Port 00h ist 0 : Datum geht an Console
D6 in Port 00h ist 1 : Datum geht an Drucker

Dafür muß der Pegel von A1 des ECB-Bus bei der Ausgabe des Host an das Terminal in einem Flip Flop gespeichert werden und der Z80 des Terminals muß diesen Wert über einen Treiber auslesen können.

In Hardware

ist auf IC3, 74LS74 ein weiteres 74LS74 mit den Pins 7 und 14 huckepacks aufzulöten und auf IC4, 74LS125 ist ein weiteres 74LS125 mit den Pins 7(GND), 10(/IN0) und 14(+5V) huckepacks aufzulöten.
Die Pins 9 dieser beiden aufgelöteten ICs sind miteinander zu verbinden.
An Pin 12 des 74LS74 wird A1 des ECB-Bus (Pin 7c) gelegt.
Pin 11 des 74LS74 wird mit Pin 10 von IC11, 74LS138 verbunden (/WRDATA).
Pin 8 des 74LS125 wird mit Pin 4 von IC2, 74LS374 verbunden (D6).

In Software

ist in das Betriebsprogramm des Terminals in die Interrupt-Service-Routine bei Ausgaben vom Host nach der Ermittlung des Status des ECB-Interface anhand des Pegels von D6 in Port 00h festzustellen, ob das Byte in Port 01h an die Console (D6=0) oder an den Drucker (D6=1) ausgegeben werden soll.

Ich will meinen alten Compie loswerden, damit ich wieder Platz in meinen Regalen für Neuanschaffungen bekomme:

Die Lieferliste enthält:

Hardware:

1 * Videogenie EG3003 mit 256KB RAM. Gebankt in 32KB-Scheiben. Der Banker paßt zu LeSctipt ab Ver. 1.8 und stammt direkt von Anitek. Der Zeichensatz enthält auch Kleinbuchstaben auf dem Bildschirm.

1 * HRG 1B Grafikkarte

1 * Diskontroller mit DD-Erweiterung. Der Kontroller ist in den Rechner eingebaut.

1 * Taktumschaltung 1.78Mhz <-> 3.5Mhz. Die Umschaltung kann während des Betriebes ohne Rechnerabsturz erfolgen ! Das von mir benutzte NewDos80 hat im Bootteil einen Batch, der sicherstellt, das mit 1.78Mhz gebootet wird (sonst bleibt die Kiste hängen) und schaltet dann selbstständig auf den höheren Takt um. Vor einem erneuten booten (per Reset oder Boot-Befehl) muß allerdings von Hand zurückgeschaltet werden ! Centronics-Port für den Drucker ist auf dem Kontroller-Board vorhanden. Kabel wird mitgeliefert !

1 * Doppellaufwerkstation 2 * 360KB mit eigener Stromversorgung. Laufwerke TEAK FD-55. Das Ganze ist in einem sehr kompakten Gehäuse untergebracht.

1 * Monitor in s/w. Es handelt sich dabei um einen umgebauten Fernseher. Die Auflösung wurde aber von mir deutlich verbessert, sodaß sich ein scharfes Bild ergibt. Auch bei der HRG-Darstellung.

Achtung: Die zweite Casetten-Recorder Buchse wurde von mir seinerzeit umgebaut und werkelt nun als RS232C-Schnittstelle zum seriellen Senden von Daten mit Hardware-Handshake. Es Fehlen jedoch alle anderen Signale die es sonst noch so gibt ! Mein erster Drukker war seriell und kam mit dieser Konfiguration klar. Eventuell habe ich noch das Treiberprogramm zum Übertragen der *ASCII*-Dateien.

Software:

1 * "Sackweise" Disketten. Konkret: ca. 80 Stück mit daten und ca. 20 frisch aus der Packung. Die genaue Anzahl ist mir nicht bekannt (und zum Zählen bin ich zu faul). Das alles befindet sich in einem 5 1/4" Diskettenkasten in Schubladenform. Diese Plastikschachtel (sehr stabile Ausführung) hat mich seinerzeit einen Blauen gekostet. Es handelt sich also um was "haltbares". Die Software stammt übrigens in zählbarer Menge von Kamerad Hanke. Wegen der vielen Arbeit habe ich die Disketten jedoch *nicht* umkopiert ! Das PDrive-Format habe ich aber noch irgendwo herumliegen. Wer in diesem Zusammehang außerdem an LeScript Ver. 2.02 (topaktuell aus 1990 !!!) interessiert ist, sollte allerdings dafür eine extra Frage stellen - wegen der DM's, es werden aber nicht viele sein, schließlich will ich das Zeug loswerden (da mir hat das Scheibchen ein paar Mark gekostet hat und ich diese Software nicht so einfach verschenken möchte).

1 * Einiges an Handbüchern zur vorhandenen Software. In etwa 2m Regal sind schon notwendig, um es unterzubringen. Ein Auszug vom Vorhandenem (ohne Anspruch auf vollständigkeit):

- Fortran 80 System (im Original)
- Series I Editor Assembler (im Original)
- M80, L80
- Pascal 80 (im Original)
- RPNL Ver. 2 (mit Source-Code)
- MMSForth
- u0Lisp
- ZBasic V2.2
- Basic-Compiler
- Accel 3
- Visicalc
- Zeus-Assembler
- MZal-Assembler
- Misosys Disassembler Disasm
- Trekcess (Diskaccess Utility)
- Superutility
- Kermit (m. Source-Code)
- ....

Zu den meisten der genannten Programme habe ich auch die Handbücher. Übrigens auch für NewDos80. Hierfür sogar in deutsch und als Buch gebunden.

Desweiteren habe ich auch noch so manches Buch zum Genie (Grosser und Co.) sowie das System-Manual mit den Schaltplänen. Dto. für die ganze zusätzlich eingebaute Hardware.

Nun zum Geschäftlichen. Den ganzen Kram gedenke ich ***komplett*** abzustoßen für DM500.-- Ich denke in Anbetracht der Menge an Material ist der Preis nicht zu hoch gegriffen. Ich muß aber dazu sagen, das der Rechner nicht mehr versandfähig ist wegen der vielen Einbauten. Es ist also nur bei direkter Abholung möglich, ihn zu erwerben. Eine technische Betreuung wie Reparatur, Wartung etc. kann ich nicht gewähren. Hier ist der neue Besitzer selbst gefordert. Mir fehlt für diesen Support einfach die Zeit. Im übrigen gilt "gekauft wie besehen". Was nicht ausschließt, daß sich der Interessent das Gerät in Funktion betrachten kann. Bei eventuellen und alseits in solchen Momenten gefürchteten Vorführeffekten bitte ich um Nachsicht und Geduld. Nicht nur mir flattern dann die Nerven, auch dem Rechner. Wer interessiert ist und mich zwecks Kauf besuchen möchte, sollte sich allerdings vorher telefonisch oder wie auch immer rechtzeitig anmelden, da meine Freizeit erst abends ab 19.00Uhr beginnt, ich also nicht früher im Hause bin.

Meine Anschrift nebst "sprechendem Draht":

Kurt Müller
Sophie-Scholl-Ring 3B
2054 Geesthacht

04152/70643

– Club 80 Börse

Club 80 Börse

Verschenke:

Z 80 Assembler-Sprache
Benutzer-Handbuch (deutsch)
KONTRON-Verlag
(293 Seiten)

+

Microsoft BASIC
Decoded & other mysteries
von J. Farvour
(310 Seiten Fotokopie)

Verschicke unfrei!

K.-J. Mühlenbein [Adr. s. Mitgl.-Verz.]

Jörg Brans
Tieloh 55
2000 Hamburg 60

Hmb., 17. April 1991
040 - 690 65 31
691 27 16

Club 80 Börse

Im Laufe der Zeit hat sich mein Gerätepark derartig vergrößert, so daß ich aus Platzgründen einiges verkaufen will. Folgende Hardware ist daher zu verkaufen:

1.) Computer / gebraucht

TRS 80 Mod. IV P - 512 K / 8 Mhz / Laufw. 80/40
Schnittstellen: Par./Ser. und
für externe Laufwerke Prs. 800,- DM

2.) Hard und Software für externe Festplatte
Mod. IV bzw IV P

Alles Neuteile / ungebraucht (vorsorglich als Ersatzteile gekauft)

a.) SCSI Hostinterface von Misosys
b.) Controller Adaptec 4000 A
c.) Verbindungskabel Hostinterface - Controller (50 Pin)
d.) Verbindungskabel Controller - Festplatte (20 Pin)
e.) Software für Mod. 4 (TRS/LSDOS 6.3) u. Mod. III (LSDOS) incl. diverser Utilitys.

Prs. 350,- DM

Hinweis:

Das Hostinterface "übersetzt" alle Signale und Befehle die zum Controller gehen, der einen SCSI Eingang und einen ST406/512 (MFM) Ausgang besitzt.

Anschließen läßt sich jede Festplatte die nach ST412/506 Norm im MFM - Aufzeichnungsverfahren arbeitet. Die Amerikaner bieten bevorzugt Seagate Festplatten an.

3.) Drucker

Ein gebrauchter Drucker Tandy DMP 430 (DIN A 3)

mit Par. u. Ser. Schnittstelle
umschaltbar auf IBM Zeichensatz

ohne Druckkopf (defekt) gegen Gebot zu verkaufen.

Wer Interesse hat soll sich bitte bei mir schriftlich oder telefonisch melden.

Jörg

Leserbrief zu
"Newdos, CP/M, MS-DOS, UNIX" in Info Nr.32

Gerald, grundsätzlich sprichst Du mir da aus der Seele. Ähnlich ketzerische Äußerungen zu MS-DOS und PC und ihrer Verwandtschaft zu NEWDOS und Trash-80 habe ich - glaube ich - vor längerer Zeit auch schon mal im Info verbreitet. Dazu stehe ich auch heute voll und ganz. Hier will ich aber mal versuchen, die Sicht des Anwenders dagegen zu stellen und ein paar positive Seiten der beiden Anachronismen hervorheben.

Gehen wir aber zuerst mal gute 10 Jahre zurück: Damals hat ein kleiner Bauchladen Namens Radio Shack einen Billigst-Computer herausgebracht und in den USA über die Tandy-Kette verkauft. Wegen des niedrigen Preises war das Gerät ein Riesenerfolg und wurde in großen Stückzahlen verkauft. Daraufhin schossen etliche Software-Firmen aus dem Boden und brachten gute Betriebs-Systeme und gute Programme auf den Markt. Dieses für damalige Verhältnisse riesige Software-Angebot hätte es bestimmt nicht gegeben, wenn der TRS-80 nicht aufgrund seines niedrigen Preises eine solche Verbreitung gefunden hätte und dadurch für die Softfirmen ein entsprechender Markt bestanden hätte.

Big Blues PC hat sich zunächst natürlich nicht des günstigen Preises (der kam erst sehr viel später durch Taiwan-Konkurrenz) sondern wegen der Machthörigkeit der ganzen Branche (auch der Fachpresse) verbreitet, obwohl die Erfinder des Gerätes sich alle Mühe gegeben hatten, ein möglichst primitives System zu schaffen, das man durchaus billiger verkaufen könnte. Das Vorhandensein von DMA-Controller und Timer war nicht etwa eine großzügige Ausstattung zugunsten von mehr Performance sondern für den Refresh der dynamischen RAMs nötig, weil die CPU das nicht wie ein Z80 selbst erledigt. An diesem Murks orientieren sich auch heute noch alle Neuschöpfungen des Industrie-Standards, sodaß Software für den PC-XT auch noch auf einem EISA-486er-AT läuft.

Bei diesem Softwaremarkt stürzten sich alle Softfirmen auf den großen Kuchen und niemand hatte mehr etwas für andere Syteme zu bieten. Die Fülle der Anwenderprogramme führte dazu, daß die Programme nicht mehr nur die eigentlich bezweckten Funktionen erfüllen mußten, um gegenüber der Konkurrenz bestehen zu können; sie mußten auch durch originelle äußerliche Gestaltung das Interesse des Käufers auf sich lenken. Die 'Benutzeroberfläche' war erfunden.

Sowas läßt sich aber nur programmieren, wenn man die volle Kontolle des Bildschirms hat. Ein unter CP/M übliches Terminal ist dagegen ein Datengrab, das kaum Kontolle darüber zuläßt, was wo im Bildschirm steht. So erfolgen die Bildschirmausgaben bei den meisten Programmen unter CP/M zeilenweise. Wer unter CP/M mit DU.COM gearbeitet hat, sehnt sich nach SUPERZAP von NEWDOS/80 oder DEBUG zurück, wo das alles schon sehr viel eleganter durch Bewegen des Cursors über den ganzen Bildschirm möglich war und das auch nur wegen der festgelegten Hardware und des memory mapped Bildwiederhol-Speichers, den die CPU selbst verwalten muß bzw. manipulieren kann. Fenstertechnik ist nur möglich, wenn Bereiche des Bildwiederhol-Speichers ausgelagert werden können, um nach Schließen des Fensters wieder restauriert werden zu können.

Programme unter CP/M, die ansatzweise diesen Komfort zu bieten haben, müssen installiert werden. Die dabei durchzustehenden Fragen bringen auch einen alten Hasen gelegentlich aus dem Tritt, wenn er nach dem ersten Durchgang merkt, daß seine Annahme, er müsse die Daten in Hex eingeben, falsch war, und er nochmal wieder die entsprechenden Informationen zu seinem Terminal im Handbuch nachschlagen muß.

Auf dem PC bedeutet Installieren im schlimmsten Fall, der Software bekanntzugeben, was für eine der 3 denkbaren Videokarten denn nun im Gerät steckt; das kriegt noch jeder hin, weil er sich beim Kauf des PCs wegen der sehr unterschiedlichen Preise darüber Gedanken gemacht haben muß. Meistens bemerkt das Inatallationsprogramm die Existenz der jeweiligen Videokarte aber selbst und belästigt den Anwender garnicht erst mit entsprechenden Fragen.

Ähnlich verhält es sich mit der Tastatur: Wie beim TRS-80 erhält die CPU nur Informationen von der Tastatur, welche Taste gedrückt bzw. losgelassen wird. In beiden Fällen muß die CPU sich aus der Identität der gedrückten Tasten selbst ein ASCII-Zeichen basteln oder sich sonst einen Reim daraus machen. Und da man mit TSR-Programmen beim PC selbst in die Auswertung der Tastatur-Scancodes eingreifen kann, ist ein kleiner Geschmack von Oligo-Tasking möglich ("Multi-Tasking" wäre etwas zu hoch gestochen).

Ein weiterer Schachzug der Softwarefirmen war, in die Programme eine jederzeit zugängliche Online-Hife einzubauen, die ein ständiges Blättern im Handbuch (und schließlich das Handbuch selbst) unnötig machte. Jeder konnte recht schnell mit einem Programm umgehen. Natürlich kostet das Bereitstellen von Hilfstexten und auch die üppige Ausstattung der Programme mit Menüs Speicherplatz.

Dem Anwender sind Kriterien wie Portabilität und Hardware-Unabhängigkeit ziemlich schnuppe; er will möglichst schnell sein Programm beherrschen können und vom Programm eine geschmackvolle Bildschirmgestaltung geboten bekommen. Er kann ohnehin nicht beurteilen, mit welchem Aufwand an Gehirnschmalz die einzelnen Funktionen des Programms realisiert worden sind. Wenn er ein neues Programm vor sich hat, dort die gewohnte SAA-Oberfläche wiederfindet und auch gleich ein paar minimale Funtionen bewirken kann, ist das Erfolgserlebnis größer, als wenn er nach vielen Schiffbrüchen endlich in der Lage ist, durch kryptische Befehle gewisse Funktionen auszulösen.

Helmut Bernhardt

Impressum

1. Vorsitzender	Fritz Chwolka
Tel.: 0 24 64/89 20	Saarstraße 4
5173	Aldenhoven
2. Vorsitzender	Gerald Schröder
Tel.: 0 41 05/26 02	Am Schützenplatz 14
2105	Seevetal 1
Hardewarekoordinator	Andreas Magnus
Tel.: 02 09/87 02 30	Pommernstraße 4
4650	Gelsenkirchen
Newdos-Diskothekar	Oliver Volz
Tel.: 07 11/7 35 38 17	Waldburgstraße 73
7000	Stuttgart 80
CP/M-Diskothekar	Rüdiger Sörensen
Tel.: 0 61 31/3 28 60	Thomas Mann Straße 3a
6500	Mainz 1
Clubbücherei	Kurt Müller
Tel.: 0 41 52/7 06 43	Sophie-Scholl-Ring 38
2054	Geesthacht
Redaktion	Jens Neuder
Tel.: 07 91/4 28 77	Rudolf-Then Straße 32
7178	Gschlachtenbretzingen

Autoren: Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 8870 Günzburg
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Schluß

Hallo Club 80'er,

eigentlich wollte ich das INFO einen Monat früher herausbringen, aber leider hatte ich keine Artikel dazu. Erst jetzt, in den letzten zwei Wochen, kam noch etwas INFO-Material per Post, sodaß ich endlich redaktionell tätig werden konnte. Aus diesem Grund ist diesmal unser INFO auch wieder etwas dünner -aber das habt Ihr ja selbst schon gemerkt.

Schade fand ich auch, daß es diesmal mit dem Club-Treffen nicht so richtig geklappt hat. Zu guter Letzt konnten wir aber dann doch noch in geselliger Runde Erfahrungen und Daten in Leibheim austauschen. Ein Dankeschön an die Initiatoren und den Wunsch, daß es im nächsten Jahr wieder ein größerer Kreis sein wird, der zusammentrifft. Vielleicht klappt es auch dieses Jahr noch -im Herbst?

Zum Abschluß dieses INFO's bleibt mir noch, Euch einen schönen Urlaub und der Club-INFO viele Artikel zu wünschen.

Viel Spaß-
bis zum nächsten mal

Jens

Adressen-Liste Club80

Stand vom 30-06-1991
Seite 1

Nachname	Vorname	Straße	PLZ	Ort	Telefon privat	Telefon geschäftlich
Behrendt	Detlef	Schlosserbreite 1a	W- 8018	Grafing	08092 / 9173	
Benner	Achim	Vorm Mühlberg 1	W- 5910	Kreuztal 3	02732 / 3780	
Bernhardt	Helmut	Hafenstraße 7	W- 2305	Heikendorf	0431 / 241907	0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	W- 8551	Hausen	09191 / 31698	09191 / 611108
Brans	Jörg	Tieloh 55	W- 2000	Hamburg 60	040 /6906531	
Braun	Harald	Postfach 8011	W- 2300	Kiel 17		
Böckling	Ulrich	Jochaczstraße 61	W- 5410	Hör-Grenzhausen	0264 / 4861	02631 / 895168
Chwolka	Fritz	Saarstraße 34	W- 5173	Aldenhoven	02464 / 8920	
Drechsler	Oskar	Duckterather Busch 2	W- 5060	Bergisch Gladbach 2	02202 / 55282	0228 / 128114
Gill	Thomas	Maria-Eich-Straße 34	W- 8000	München 60		
Günther	Jens	Bannerscheid 7	W- 5231	Neitersen	02681 / 1553	
Halgasch	Gert	Großschonauer Straße 26	O- 8805	Jonsdorf		
Hartmann	Hans-Günther	Möwenstraße 9	W- 2876	Berne 2	04406 / 6911	0421 /2483405
Held	Manfred	Stirner Straße 22	W- 8835	Pleinfeld	09144 / 6563	0911 /2195245
Hermann	Klaus	Forchenstraße 8	W- 7401	Pliezhausen	07127 / 71945	
Johnen	Willi	Hansemannstraße 1	W- 5160	Düren	02421 / 501305	
Kemmer	Jürgen	Dorfberg 7	W- 8701	Sulzdorf	09334 / 1050	
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 /4828948	00411 / 258170
Kuhn	Eckehard	Im Dorf 14	W- 7443	Frickenhausen	07022 / 45417	
Littmann	Claus	Plockhorst, Zum Spring 15	W- 3155	Edemissen	05372 / 7796	
Lorenz	Walter	Mahräckerstraße 9	W- 6000	Frankfurt /Main 50	069 / 531656	
Magnus	Andreas	Bismarkstraße 29	W- 4650	Gelsenkirchen	0209 / 870230	
Mahlert	Herbert	Hohenbudbergerstraße 112 A	W- 4100	Duisburg	02135 / 47217	
Mand	Harald	Kl. Flintbeker Straße 7	W- 2302	Flintbek /Kiel	04347 / 3629	0431 /3013580
May	Holger	Marienstraße 9	W- 5768	Sundern 2	02935 / 1668	
Menk	Christian	Ollsener Straße 52	W- 2116	Hanstedt	04184 / 7825	
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	W- 6940	Weinheim-Lützelsachsen	06201 / 55052	
Müller	Kurt	Sophie-Scholl-Ring 3b	W- 2054	Geesthacht	04152 / 70643	04151 / 889137
Mössel	Franz	Schafferstraße 12	I- 39012	Meran		
Neebe	Gerhard	Märkische Straße 186	W- 4600	Dortmund 1	0231 / 416549	
Neueder	Jens	Rudolf-Then-Straße 32	W- 7178	Gschlachtenbretzingen	0791 / 42877	0791 /44-8279
Neumann	Christof	Theodor-Heuss-Straße 8	W- 7265	Oberhaugstett	07307 / 5147	
Nitschke	Stefan	Germanenstraße 5	W- 7519	Walzbachtal 1	07203 / 452	
Obermann	Hartmut	Mozartring 23	W- 8870	Günzburg	08221 / 30248	
Peters	Jürgen	Heukoppel 14	W- 2000	Hamburg 14	040 /6412371	
Reit	Hermann	Vechter Hof 40	W- 4500	Osnabrück	0541 / 16331	
Retzlaff	Bernd	Kleiner Sand 98	W- 2082	Uetersen	04122 / 43551	04103 / 605310
Rinio	Gerd	Rennbahnstraße 9	W- 2000	Hamburg 74	040 /6552630	
Ruschinski	Claus	Pommernstraße 21	W- 4370	Marl	02365 / 34646	
Schilling	Werner	Ehndorfer Straße 340	W- 2350	Neumünster	04321 / 61116	
Schmid	Alexander	St. Cajetan Straße 38/VII	W- 8000	München 80	089 / 495326	
Schmitz	Paul-Jürgen	Bremer Straße 9	W- 6236	Eschborn		
Schmitz	Rainer	Dornierstraße 17	W- 7320	Göppingen	07161 / 22549	
Schober	Frank-Michael	Weberweg 2	O- 7590	Spremberg	0037574/ 4565	
Schoberth	Uwe	Petrus-Waldus-Straße 14	W- 7136	Oetisheim	07041 / 7254	
Schoof	Frank	Elpke 5	W- 4800	Bielefeld 1	0521 / 237749	
Schroers	Horst-Dieter	Breslauer Straße 9	W- 8016	Feldkirchen	089 /9032615	089 /9032615
Schröder	Gerald	Am Schützenplatz 14	W- 2105	Seevetal 1	04105 / 2602	
Schröder	Peter	Theodor-Fahr-Straße 32	W- 2000	Hamburg 62	040 /5311582	
Schröer	Egbert	Joachimstraße 18	W- 4270	Dorsten 1	02362 / 75840	02365 / 495183
Schut	Andre	Sanderstraße 26	W- 1000	Berlin 44	030 /6917861	
Schwarz	Wolfgang	Schwedenring 6	W- 8850	Donauwörth	0906 / 3092	
Schäfer	Walter	Rathausstraße 4	W- 8160	Miesbach	08025 / 1631	08025 / 41247
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	W- 5305	Alfter 4	0228 / 643853	
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanloese		
Sopp	Arnulf	Wakenitzstraße 8	W- 2400	Lübeck	0451 / 793054	
Stephan	Hans-Martin	Am Glasesch 9a	W- 4506	Hagen a. TW	05401 / 99585	05401 / 30096
Stumpferl	Stefan	Hasenberglstraße 57	W- 8000	München 45	089 /3144001	
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	W- 6500	Mainz 1	06131 / 32860	06131 / 395268
Thieleke	Klaus	Ahrenshooper Straße 47	O- 1093	Berlin	0372 /3223392	
Tornow	Wilhelm	Görlitzer Straße 16	W- 2190	Cuxhaven 13	04723 / 1355	04723 / 3063
Vollkmer	Richart	Am Spörkel 69	W- 6700	Dortmund 50		
Volz	Oliver	Waldenburgstraße 73	W- 7000	Stuttgart 80	0711 /7353817	
Waccus	Michel	Mühlhofweg 2a	CH- 8266	Steckborn		
Werner	Heiko	Reichenberger Straße 5	O- 8032	Dresden		
Wittkamp	Heinz	Hindenburgstraße 37	W- 5630	Remscheid	02191 / 75132	
Wulf	Hans-Otto	Im Brahmkamp 38	W- 4250	Bottrop	02041 / 688972	

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

HARDWARE * SOFTWARE
COMPUTER-CLUB 80
TANDY-/ GENIE- ANWENDER
ERFAHRUNGSAUSTAUSCH

Achtung !!
wichtiger Termin !

Hobbytronik-Treffen

am Sonntag, den 10.11.91

gleich morgens zum Einlaßbeginn
oder
mittags, 12.00 Uhr
bei der Kartenkontrolle
am unteren Eingang

CLUBINFO

35. Ausgabe

Kontaktadresse: **Club80** / Fritz Chwolka / Saarstraße 34 / 5173 Aldenhoven
Tel.: 0 24 64 / 89 20

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Anmerkung zur Jahreshauptversammlung 91	Fritz Chwolka	1
Neue Mitglieder in Kurzform		2
Termine	Redaktion	2
Vorstellung neuer Mitglieder	Rainer, Volker, Manfred	3 - 5
Mitteilung	Willi Johnen	6

Software

Axonometrie	Christof Neumann	7 - 8
Die Formatknackerbande schlägt zu	Hartmut, Gerald, Hans-Günther	9 - 18
ADRESS - TEXT - 90	Willi Johnen	19 - 24
Windows 3.0 unter DR-Dos 5.0		26
Patch für DR-Dos 5.0 Kommando-Interpreter		26
ADDPath	Harald Braun	26 - 28
Seriell mit PIP	Fritz Chwolka	29 - 33

Autor & Seite

Hardware

Club-80-Netzwerk	Rainer Schmitz	35 - 40
Anmerkung zum Club-Terminal, erste Erfahrungen	Alexander Schmid	41 - 42
PC-Hardware	Harald Braun	43 - 47
Floppylaufwerke an CPC-Computern	Manfred Hürdler	48 - 53
Anmerkungen Floppylaufwerke CPC	Fritz Chwolka	54

Börse

Wer hat was – wer will was		55 - 58

Sonstiges

Betriebssysteme: Oberflächen	Gerald Schröder	59 - 61
Information Framework	Paul-Jürgen Schmitz	62
NEAT-AT		63 - 64
MFM-Festplatten	Harald Braun	63 - 64
Heiße Liebesgrüße aus dem PC	(FN 29.10.91)	65 - 66

Die letzten Seiten

Impressum		67
Schluß	Redaktion	68
Mitgliederadressenliste		am INFO-Ende

Anmerkung zur Jahreshauptversammlung 1991:

Einleitend möchte ich nun doch noch etwas zu unserem letzten Clubtreffen schreiben, obwohl ich dies nicht gerne mache.
Vor einiger Zeit war unsere alljährliche Jahreshauptversammlung angesagt , und die Einladungen wurden allen Clubmitgliedern (und Interessierten) zugesandt. Eine Woche vor dem Tagungstermin hatte Hartmud Obermann gerade zwei (in Worten 2) Anmeldungen (!?!)

Das ist schon fast eine Beleidigung dem Vorstand gegenüber und das Interesse der Personen, welche bis jetzt aktiv am Clubleben teilnehmen, ist dadurch bestimmt nicht größer geworden.
Wenn Ihr Wünsche, Vorschläge oder sonstige Anliegen habt so teilt sie uns doch bitte mit. Durch Kommunikation ist meistens mehr zu erreichen als durch Ignoranz und Passivität. Ein Club lebt nur durch seine Mitglieder und deren Aktivitäten. Der Vorstand soll leitend einwirken, diplomatisch und angemessen die Aktivitäten koordinieren und den Überblick betreff der Finanzlage behalten etc..
Manche von Euch haben sich über dies oder jenes beschwert, zum Beispiel über die Erscheinungsdaten der Clubinfos.
Meckern und zu hause sitzen bleiben ist natürlich einfach , aber fühlt IHR EUCH dabei wohl ?
Gerade der persöhnliche Kontakt auf den Clubtreffen lasssen jedes Treffen zu einem besonderen Ereigniss werden und auch Ihr solltet daran teilhaben.
Ganz so nebenbei möchte ich den Zauderern erzählen, daß ich neben acht CP/M Rechnern auch noch eine MSDOS 286er Kiste habe, aber keine zehn Meter zu einem MSDOS'ler Treffen fahren würde.
Wir haben halt einige besondere Schätzlein unter unseren CP/M Kisten , mit Geld garnicht zu bezahlen und Briefmarken (oder Bierdeckel) sammelt halt jeder.
Na ja, vielleicht wart Ihr alle verhindert und das nächste Clubtreffen beschehrt uns einen neuen Teilnehmerrekord.
Ebenso höre ich mir gerne EURE Ausreden zum nächsten Clubtreffen an und werde wohl eine erhöhte Telefongebühr haben, da ich - wenn möglich - jeden (!) persöhnlich anrufen werde.
Bitte nehmt Euch dies zu herzen.
Es ist nicht gut wenn sich welche wie Jutta und Hartmud Obermann, Jens Neueder , Alexander Schmid, Helmud Bernhard und andere für den Club einsetzen und keine Reaktion von EUCH kommt. Ich habe hier bewußt eine Wertung (positiv oder negative Reaktion) rausgelassen. Wir freuen uns schon wenn überhaupt eine Reaktion entsteht .
Also nehmt diesen Text bitte persöhnlich und fühlt EUCH angesprochen, denn nur so kann der Club weiter so aktiv arbeiten und allen zu Nutze sein .

Euer Vorsitzender

Frih. Chwolka

Neue Mitglieder in Kurzform:

Ilse Berndt, Stachelsgut 24, 5060 Bergisch Gladbach 1
Telefon: 02204/65254 Genie III + IIIs (GDOS)

Volker Dose, Dorfstrasse 10, 2304 Brodersdorf
Telefon: 04343/1357 Genie I (NEWDOS)

Dietmar Kauka, Strasse des Friedens 37, O-7201 Neukirchen
Telefon: 0037404/850112 Sharp MZ-800

Manfred Hürdler, Niederhoferstrasse 29, 8709 Rimpar
Telefon: 09365/4235 CPC6128

==================================

Termine... Termine... Termine... Termine... Termine

Clubtreffen Nord	....	......- 91
Jahreshauptversammlung 92	....	......- 92
Clubtreffen Süd	....	......- 92
Hobbytronik	Stuttgart	06.11.- 10.11.91

Redaktionsschluß für die Clubinfo's ist jeweils der letzte Tag der folgenden Monate:

Dezember 91
Februar 92
April 92

Hallo Leute!

Eigentlich bin ich recht spät dran mit meiner Vorstellung, aber besser spät als nie! Zunächst mal was zur Person: ich bin 29 Jahre alt, ledig und von Beruf Dipl. Ing (FH), Fachrichtung allgemeine Elektrotechnik. Wie man meiner Berufsbezeichnung entnehmen kann, weiß ich, an welchem Ende der Lötkolben warm wird. Das muß ich auch, da ich beruflich die freien Räume von Modellbahnfahrzeugen mit Digital-Decodern füllen muß. Ebenso gehört die Entwicklung elektronischer Steuergeräte (auch mit Microcontrollern) zu meinem Aufgabenbereich bei Gebr. Märklin und Cie. in Göppingen. Da ich mich auch in meiner Freizeit mit der großen und kleinen Eisenbahn beschäftige, kann ich mich zu den wenigen rechnen, die für ihr Hobby bezahlt werden!

Meine "Computerleidenschaft" begann schon zu Studienzeiten mit einem Eigenbau: von Motorola gab es mal eine "Industrial Control Unit", einen Ein-Bit-Prozessor, für den es meines Wissens noch nicht mal einen Assembler gibt und der nur die grundlegenden Boole'schen Verknüpfungen beherrscht. Mit diesem Teil wollte ich eine Automatik für den Abstellbahnhof meiner Modellbahn bauen, ich war sogar schon recht weit, da ereilte mich das Schicksal in Gestalt eines Arbeitsplatzes bei Märklin und ich hatte auf einmal ganz andere Möglichkeiten zur Anlagensteuerung.

Der nächste Computer war ein Bausatz des gefürchteten ZX 81, der auf Anhieb lief und von mir gleich in ein anderes Gehäuse mit ordentlicher Kühlung für den Spannungsregler eingebaut wurde. Die Krankheit mit den thermischen Aussteigern des Gate-Arrays wollte ich von vornherein ausschließen!

Als die Diplomarbeit anstand, kratzte ich meine letzten Kröten zusammen und kaufte mir einen Schneider Joyce, und zwar als Textsystem. Vergleichbare Systeme aus dem PC-Bereich gab es damals nicht oder nicht zu dem Preis und ich

hatte keine Lust, mich durch den Marktdschungel zu kämpfen, um zu einer funktionsfähigen Kombination zu kommen. Daß das Ding ein richtiger CP/M-Computer ist, habe ich erst nach und nach entdeckt, ebenso wie die Möglichkeiten zum Speicherausbau und zum Anschluß eines weiteren Floppylaufwerks. Danach gab es kein Halten mehr: ich habe ein externes Laufwerk mit Netzteil eines Thomson-Rechners ergattert und meine Kiste mit einen Anschluß für den Floppybus versehen, so daß ich nach Lust und Laune 3½ und 5¼ Zöller anschließen kann. Der Speicher ist auf 512 k aufgebohrt und wird zum Teil von einer 368k großen Ramdisk belegt.

Mein Zweitrechner ist ein Wavemate Super Bullet mit CP/M+ und MP/MII, vorbereitet als Mehrplatzsystem mit 4 seriellen Schnittstellen, haufenweise Floppies (bis zu 10) und einem SCSI-Anschluß für die Festplatte. Der Rechner stellte 1984 in der CP/M-Szene das dar, was heute der 486er ist! Dazu gehört ein Terminal Zenith Z-49, das ich von Helmut Jungkunz erwerben konnte.

Neu in der Sammlung ist ein Farb-CAD-System für Leiterplatten (auch SMD!) mit einem 6809 nebst Z80-Coprozessor, es läuft unter Flex (???) und bei Bedarf unter CP/M. Der Hersteller ist Eltec, Mainz, aufgebaut ist das Ganze um einen EUROCOM II herum, der Videocontroller ist zu Fuß (!) aufgebaut. Erfreulicherweise ist das System mit deutscher Gründlichkeit dokumentiert, sogar die PAL-Listings sind dabei! Ansonsten ist aber noch einiges an Forschung fällig, bis das Teil richtig läuft. Momentan arbeite ich mich in Z3PLUS ein, um das Letzte aus meinen Rechnern herauszuholen. Ansonsten passe ich PD-Software für den Hausgebrauch an und entwickle Hardware (siehe den Artikel "Das Club 80 Netzwerk").

Dem Club 80 bin ich in Mannheim beigetreten, ich dachte vorher, ich sei der letzte Mohikaner! Gut, daß dem nicht so ist; ich mache auch bei jeder Gelegenheit Werbung für unseren Club, wenn ich eines anderen CP/Mlers ansichtig werde. Wer mich in Natura sehen will, sollte sich auf Clubtreffen sehen lassen, die Gefahr ist sehr groß, daß ich auch da bin!

Hallo Computerfreunde!
Ich heisse Volker Dose, bin 25 Jahre alt und Informationselektroniker von Beruf. Ich habe seit Februar 1991 ein Videogenie mit zwei 80 Track DS/DD Laufwerken und Drucker.
Nachdem ich mich mit mehreren Softwarewellen, welche über mich ergingen, herumgeschlagen habe und nun auch mit NewDos ein wenig umgehen kann, habe ich mich ans Programieren gewagt. Vorerst in Basic, aber Assembler habe ich mir ernsthaft vorgenommen und eine I/O-Karte mit Abfrage in Maschinensprache gebastelt. Ich würde mir ein mäßiges Halbwissen über Computer bescheinigen und hoffe noch einiges, auch durch den Club, dazuzulernen.
Da ich mich aus Preisgründen für das Genie entschieden habe und auch das Erschließen dieses Rechners als Perspektive völlig genügt hoffe ich, daß einige von Euch weiter mit dem Genie arbeiten.
Das Schreiben von Artikeln würde ich mir schon zutrauen, bin mir aber nicht sicher ob mein Niveau dem eurer Zeitung entspricht.
(Mein bisher größtes Werk ist ein Apfelmännchenprogramm samt Hilfswerkzeug und eine I/O-Karte)
Auf hoffentlich ersprießliche Zusammenarbeit..

Volker Dose

(Anm.d.R.: Wir warten nur so auf Artikel, vor allem für die Neulinge.)

HALLO Z80 Freunde !

Ich möchte mich hier als neues Club 80 Mitglied vorstellen. Vor 31 Erdenjahren habe ich mich auf diesem Planeten materialisiert und seit 3 Jahren bin ich mit den Schaltkreisen meines Computers in Verbindung. Leider wird dieser Kontakt durch Schichtarbeit und Familienpflichten des Öfteren unterbrochen. Da mein Interesse im Bereich der Microelektronik und der Programmierung liegt, genügt mir die Leistungsfähigkeit meines Amstrad CPC 6128 vollkommen. Meinen CPC gebe ich meistens CP/M plus zu fressen und garniere diese Mahlzeit mit verschiedenen Pascalcompilern. Gerne würde ich noch Assembler und Forth erlernen, jedoch fehlt mir die Zeit und ein bischen Hilfe. Vielleicht finde ich bei Euch einen Brieffreund der die gleichen Probleme hat, auch Anfängern möchte ich Hilfe anbieten, dazu hier meine Anschrift.

Manfred Hürdler / Frühlingstr.43 / 8709 Rimpar
ab 1.8.91 : Niederhoferstr.29 / 8709 Rimpar
Telefon : 09365 / 4235

LUNG - MITTEILUNG - MITTEILUNG - MITTEILUNG - MITTEILUNG - MITTEILUNG - M

AN ALLE CLUB80-MITGLIEDER !!

Ich bin gerne bereit auch weiterhin Programme zu verschicken, die das GENIE-III/IIIS anbetreffen, wenn ein paar Spielregeln beachtet werden.

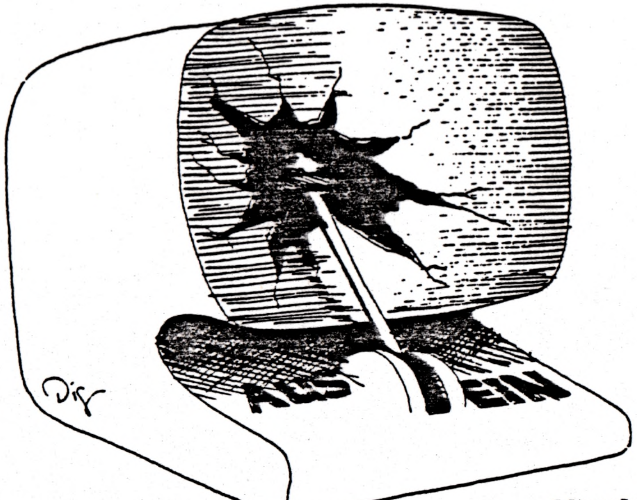
In der Vergangenheit habe ich Programme auf Disketten verschickt,nachdem mir vom Mitglied zugesagt wurde, daß ich auch Programme zugeschickt bekäme. Ich habe nicht nur die Kosten der Diskette gehabt, sondern auch die Portokosten. Die Zusage der Rücksendunge wurde aber bei bestimmten Mitgliedern nicht eingehalten.

Es ist meines Erachtens nicht Sinn und Zweck einer Allgemeineinrichtung wie der CLUB80, daß bestimmte Leute nur auf sich selber bedacht sind und das noch auf Kosten anderer.

Die Mitglieder die hiermit angesprochen sind, wissen schon Bescheid. ich dürfte doch bitten, die Zusagen auch nach Monaten noch einzuhalten.

Willi Johnen, Hansemannstr.1, 5160 Düren Tel.02421/501305

TEILUNG - MITTEILUNG - MITTEILUNG - MITTEILUNG - MITTEILUNG - MITTEILUNG



© Dietmar Große

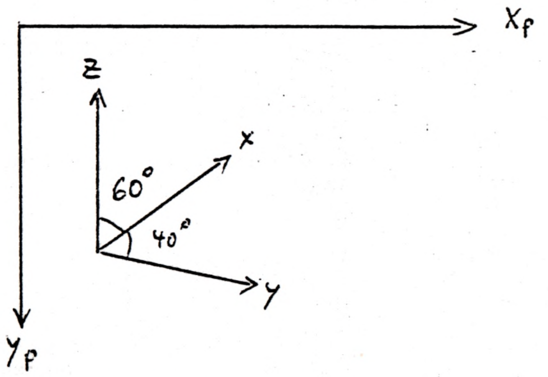
A X O N O M E T R I E

Im CAD-Zeitalter mag etwas seltsam anmuten, wenn ich jetzt ein kleines Programm vorstelle, mit dem man eine perspektivische Darstellung erzeugen kann.

Vielleicht hat aber ein Grafikprogrammierer auch schon 'mal darüber gegrübelt und fängt mit diesen Gedanken etwas an.

1. Grundlagen der AXONOMETRIE

Die Axonometrie ist eine vereinfachte räumliche Darstellung ähnlich der sogenannten "Kavalierperspektive", jedoch noch etwas gefälliger.



Die Aufgabe ist klar: Ich habe eine Anzahl von Punkten mit dreidimensionalen Koordinaten (x,y,z), die jeweils z.B. Anfangs- und Endpunkt einer Linie sein können. Mit einer Ansammlung von Linien kann man dann auch räumliche Gebilde erzeugen.

Zur Ausgabe auf dem Bildschirm (oder Drucker/Plotter) wandeln wir diese Koordinaten so um, daß sie nur noch zweidimensional aber perspektivisch erscheinen.

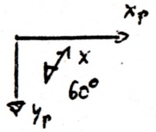
Das Verhältnis der Achsen x, y und z zueinander ist in der Skizze durch die Winkel definiert. Hinzu kommt, daß man die Länge von x etwas reduziert, um eine brauchbare Optik zu erzielen (du kannst es ja 'mal ohne Reduktion probieren ...).

Also: Lx : Ly : Lz = 3 : 4 : 4

oder auf deutsch: Länge x hat nur 3/4 (=0,75) der Ausgangslänge!

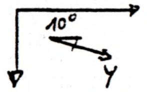
2. Mathematische Aufarbeitung

Der mathematische "Trick" besteht darin, daß man für jede Einzelachse(x,y,z) die jeweiligen Xp- und Yp-Anteile errechnet und diese dann aufsummiert.

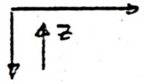


x-Anteil: XOX = +X * SIN 60° * 0.75
YOX = -X * COS 60° * 0.75

(Wir erinnern uns: 0.75 für die Optik ...)



y-Anteil: XOY = +Y * COS 10°
YOY = +Y * SIN 10°



z-Anteil: XOZ = 0
YOZ = -Z

Xp = XOX + XOY + XOZ

Yp = YOX + YOY + YOZ

3. Programmierung der Axonometrie

Das nachfolgende BASIC-Programm geht davon aus, daß in der Datei "AXO" Einzelpunkte mit den x, y und Z-Werten stehen.

```
100 '       < Einlesen >
110 DIM PX(100),PY(100),PZ(100),XP(100),YP(100)
120 OPEN "I",#1,"AXO"
130 N=0
140 N=N+1
150 INPUT #1,PX(N),PY(N),PZ(N)
160 IF EOF(1) THEN 170 ELSE 140
170 MN=N
180 '        < neue Koordinaten >
190 FOR N=1 TO MN
200 XOX=PY(N)*SIN(60*3.1416/180)*.75
210 YOX=(-1)*PY(N)*COS(60*3.1416/180)*.75
220 XOY=XOX+PX(N)*COS(10*3.1416/180)
230 YOY=YOX+PX(N)*SIN(10*3.1416/180)
240 XOZ=XOY
250 YOZ=YOY-PZ(N)
260 XP(N)=XOZ
270 YP(N)=YOZ
280 NEXT N
```

Eine veränderte Ansicht (Richtung!) erhält man leicht, wenn man Zeile 195 einfügt:

```
195 PX(N)=PX(N)*(-1)
```

Viel Spaß dabei!

Christof

Die Formatknackerbande schlägt zu

Ein Bericht vom Tatort

von Hans-Günther Hartmann - die Arbeit hatten Hartmut und Gerald -

Als ich vor einigen Jahren einen Computer namens "Video-Genie II" von einem Kollegen kaufte und ich mich an meiner Arbeitsstelle auch mit Rechnern beschäftigen mußte, wollte ich irgendwann auch mal Dateien von anderen Rechnern auf mein Genie übertragen. Dabei stieß ich auf ungeahnte Probleme.

Zuhause hatte ich einen Rechner (der/das Video-Genie), der mit einer Schmidtke-Karte CP/M 2.2 fahren konnte und im Büro hatte ich einen Alphatronic-Rechner P3, der ebenfalls das Betriebssystem CP/M 2.2 hatte.
Es mußte doch funktionieren, so dachte ich!

Die Praxis belehrte mich eines besseren. Doch dank des erwähnten Kollegen erfuhr ich von einem Programm, mit dessen Hilfe ich Dateien vom NEWDOS nach CP/M und umgekehrt übertragen konnte. Das Programm heißt CPMac. Im Info 18, Seite 44 ff., ist das Programm vorgestellt worden.

Erst viel später erfuhr ich, daß Ulrich, der Programmautor, auch Mitglied im Club ist. Von ihm bekam ich auch ein Konvertierungsprogramm, das die Disketteninhalte des P3 mit einem Zwischenschritt übertragbar machte. Die Übertragung der Daten war jetzt zwar möglich, aber mit welchem Zeitaufwand! Die Sache war für mich deshalb nicht gerade attraktiv und schlief ein, bis ich das Modell 4p des TRS 80 bekam.

Mit dem zugehörigen Montezuma CP/M 2.2 kann man viel anstellen, u.a. auch viele andere Diskettenformate einstellen und damit für das Modell 4 lesbar machen. Doch mit einer P3-Diskette funktionierte es nicht.

Beim vorletzten Nordlichtertreffen des Club 80 - Schande über mich, daß ich jetzt erst darüber berichte - sollte es jedoch zum Durchbruch kommen. Um es vorweg zu nehmen, am Abreisetag konnte ich die P3-Disketten mit meinem Modell 4p lesen und beschreiben. Das ging sogar so weit, daß eine P3-Bootdiskette nach dem Beschreiben auch immer noch im P3 booten konnte.
Aber jetzt mal der Reihe nach:

Hartmut, der in Norditalien beheimatete, berüchtigte Formatknacker, beschäftigte sich mit der P3-Diskette und fand schnell das Problem heraus: Die Alphatronic-Entwickler hatten sich nicht an den verbreiteten Standard der Spurnumerierung von CP/M-Disketten gehalten und die Directory-Spuren anders gelegt. Deshalb konnte man das Inhaltsverzeichnis nicht lesbar machen:

Directory
- bei Montezuma und anderen
  Spur 1, Seite 0 (Vorderseite)

- bei Alphatronic P3
  Spur 0, Seite 1 (Rückseite)

Das Ergebnis der Format-Untersuchungen stelle ich mal im Vergleich dar:
Dabei handelt es sich in allen 3 Fällen um Disketten mit 80 Spuren, doppelseitig und doppelter Dichte.

Parameter	Monte CP/M	P3 CPMac	P3 Original
RPT= **R**ecords **P**er **T**rack	36	40	40
BSH= **B**lock **SH**ift count	4	4	4
BLM= **BL**ock **M**ask	15	15	15
EXM= **EX**tent **M**ask	0	0	0
DSM= **D**isk **S**torage **M**aximum	354	394	394
DRM= **D**i**R**ectory **M**aximum	127	191	127
AL0= **AL**location vector 0	192	224	192
AL1= **AL**location vector 1	0	0	0
CKS= **D**irectory **C**heck **S**ize	32	32	32
OFF= Track **OFF**set	2	2	2
SPT= **S**ectors **P**er **T**rack	18	5	5
SSZ= **S**ector **S**i**Z**e Code 0=128, 1=256, 2=512, 3=1024	1	3	3
NTK= **N**umber of **T**rac**K**s	80	80	80
OPT= **OPT**ion bits	192	194	192
STT= **S**kew **T**ranslation **T**able			

Monte CP/M : 1 3 5 7 9 11 13 15 17 2 4 6 8 10 12 14 16 18
P3 CPMac : 1 2 3 4 5
P3 Original: 1 2 3 4 5

Wer mehr zum Aufbau der einzelnen Disk-Parameter wissen möchte, der kann im Club-Info 23, Seiten 49-63, nachlesen. Dort hat Gerald sich in seinem Artikel "CP/M intern - kurz und kompakt" u.a. über die Berechnung der einzelnen Parameter ausgelassen. Wer selbst mal versucht hat, diese Rechenschemata anzuwenden, wird gemerkt haben, daß es leider so einfach nicht ist.

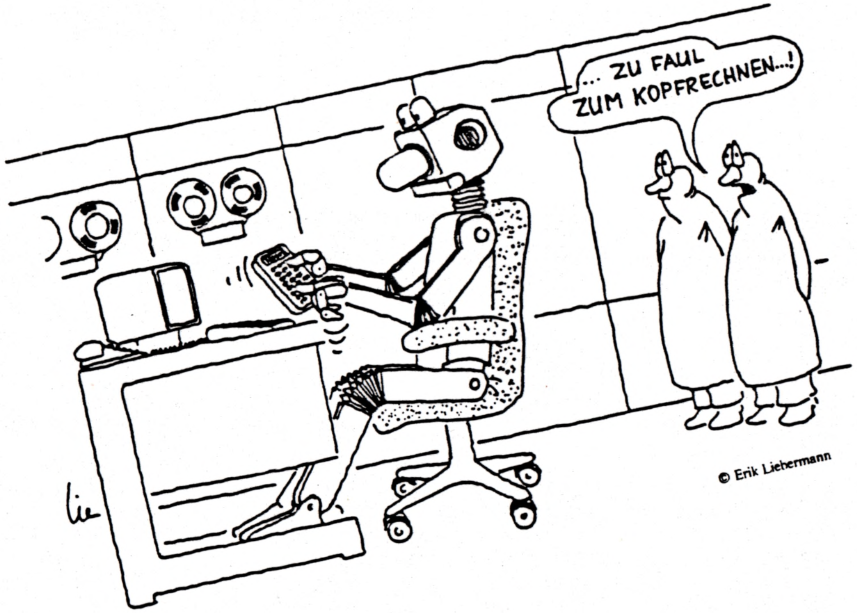
Außerdem ist das Thema der Disk-Parameter auch in dem im Franzis-Verlag erschienenen Buch von Jürgen Plate "Betriebssystem CP/M" behandelt, hier allerdings für die CPU 8080, aber das ist hierfür wohl unwichtig.

Als nun das Format entblößt vor uns lag, war klar, daß die Disketten ohne weitere Zutaten nicht lesbar sein konnten. Da der Experte zum Thema Software auch seine kostbare Freizeit mit uns teilte, konnte er überredet werden, sich des Themas anzunehmen. Gerald (ihr werdet es längst erraten haben) setzte sich also hin und entwickelte ein geniales Programm, daß die Directory-Spur 0 / Rückseite und Spur 1 / Vorderseite der P3-Diskette einfach vertauschte, so daß sie vom Modell 4 gelesen werden konnten
Ich kann nur vermuten, daß Ulrich mit seinem Konvertierprogramm zu CPMac ebenfalls diesen Weg gegangen ist, nur eben nicht unter CP/M sondern unter NEWDOS, oder?

Das Programm, P3COPY genannt, fragt zuerst nach dem Laufwerk, in dem die P3-Diskette liegt. Nach fehlerfreiem Spurtausch gibt es eine Erledigungsmeldung. Jetzt kann die P3-Diskette gelesen und beschrieben werden. Nach erledigter Arbeit P3COPY wieder aufrufen und schon kann der P3 die Disketten wieder lesen (wenn Jens noch Platz im Info hat, findet ihr das Listing nach diesem Geschreibsel).

Eine Besonderheit ist bei P3-Disketten noch zu beachten:
Im Gegensatz zu den mir sonst bekannten Rechnern, bei denen die offene Schreibschutzkerbe ein Beschreiben der Diskette ermöglicht, ist es beim P3 umgekehrt; eine offene Schreibschutzkerbe heißt für den P3, daß diese Diskette schreibgeschützt ist.

Nebenbei bemerkt, wer unter Euch einen P3 sein eigen nennt - ich kenne nur einen -, kann das Programm bei Rüdiger bekommen. Wer jetzt meint, seinem heimlich gehüteten C 64 oder Apple-Computer könnte auf ähnliche Weise geholfen werden, den muß ich enttäuschen; daraus wird nichts, weil diese Teile nun wirklich nicht annähernd dem Standard entsprechen.




```
;       Kopierprogramm für Alphatronic-P3-Disks
;       vertauscht auf einer Alphatronic-Diskette Spur 0 / Rückseite und
;       Spur 1 / Vorderseite (in Track-Zählweise: Tracks 1 und 2)
;       je nachdem wird die Diskette für "normale" Rechner oder für den
;       Alphatronic lesbar
;       die physikalischen Disk-Daten müssen stimmen (bzw. eingestellt sein),
;       sonst gibt es Schrott
;
;       von: Hartmut Obermann, Gerald Schröder und Hans-Günther Hartmann
;       für: Macro-80 unter CP/M-80
;       erstellt auf dem Nordlichtertreffen des CLUB 80 im Jahre 1989
;       letzte [nderung: 18.12.89 (GS)

            NAME  ('P3COPY')

            .Z80
            cseg

CR          EQU   13                ;Ausgabe auf Bildschirm: Carriage Return
LF          EQU   10                ;  "       "       "     : Line Feed
EOStr       EQU   '$'               ;Ende-Markierung für Strings, die das BDOS ausgibt

Track1      EQU   1                 ;erste Track-Nr.
Track2      EQU   2                 ;zweite Track-Nr.
                                    ;(gegen diesen Track wird getauscht)


                                    ;Offsets für BIOS-Aufrufe
OHome       EQU   21                ;Lw. auf Track 0
OSelDsk     EQU   24                ;Lw. anwählen
OSetTrk     EQU   27                ;Track auswählen
OSetSec     EQU   30                ;Sektor auswählen
OSetDma     EQU   33                ;DMA-Adr. auswählen
ORead       EQU   36                ;Sektor lesen
OWrite      EQU   39                ;Sektor schreiben
OSecTrn     EQU   45                ;Sektor-Nr. übersetzen (log.=>phys.)

BDOS        EQU   0005              ;Einsprung ins BDOS (ätzenderweise)
PrtStr      EQU   09                ;String ausgeben (BDOS-Fkt.)
Conin       EQU   01                ;Taste abfragen (BDOS-Fkt.)

Sektor1     EQU   00                ;Nummer des ersten Sektors
AnzSek      EQU   40

                                    ;Daten:
Laufw:      DB    00                ;gewünschtes Laufwerk
Track:      DW    0000              ;akt. Track
Sektor:     DW    0000              ;akt. Sektor
BufAdr:     DW    0000              ;akt. Buffer-Adresse
Zaehler:    DW    0000              ;soviele Sektoren noch zu lesen
TrnTab:     DW    0000              ;Adresse der Sektor-Translation-Tabelle

;-----------------------------------------------------------------------------

Start:      LD    (RettSP),SP          ;SP sichern
            LD    SP,NewStack          ;neuen Stack setzen
            LD    HL,(0001)            ;Anfangsadresse des BIOS
            LD    (BIOSAnspr),HL       ;und merken für Aufrufe des BIOS
```

```
;       1. Akt: Abfrage, welches Laufwerk benutzt werden soll
;       -----------------------------------------------------
LwNochmal:
                LD      DE,LwFrage      ;Zeiger auf Frage-String setzen
                LD      C,PrtStr        ;Frage ausgeben
                CALL    BDOS
                LD      C,Conin         ;A/B/C/D... eingeben
                CALL    BDOS
                AND     0DFh            ;'a'=>'A' usw.
                SUB     'A'             ;A=>0, B=>1
                CP      'P'-'A'+1       ;Eingabe:
                JR      Z,Abgang        ; q,Q => Abgang (Abbruch)
                JR      NC,LwNochmal    ;falsche Eingabe: nochmal
                LD      (Laufw),A       ;A..P / a..p merken

                LD      HL,OSelDsk      ;Laufwerk anwählen
                LD      C,A
                Call    BiosCall
                LD      A,H             ;gibt's das Laufwerk?
                OR      L
                JR      Z,LwNochmal     ;nee, dad Ganze von vorn
                LD      (TrnTab),HL     ;Adr. der Sektor-Translation-Tab. merken

                LD      HL,OHome        ;Lw. auf Track 0 fahren
                CALL    BiosCall

;       2. Akt: Einlesen der Tracks 1 und 2 in zwei Buffer
;       --------------------------------------------------

                LD      L,Track1        ;erste Tracknummer nach HL
                LD      H,0
                LD      DE,AnzSek       ;40 Sektoren
                LD      BC,Buffer1      ;Adresse des Buffers für Track 1
                CALL    ReadTrack       ;ganzen Track einlesen

                LD      L,Track2        ;zweite Tracknummer nach HL
                LD      H,0
                LD      DE,AnzSek       ;40 Sektoren
                LD      BC,Buffer2      ;Adresse des Buffers für Track 2
                CALL    ReadTrack       ;ganzen Track einlesen

;       3. Akt: Schreiben der Tracks 1 und 2 (vertauscht)
;       -------------------------------------------------

                LD      L,Track1        ;erste Tracknummer nach HL
                LD      H,0
                LD      DE,AnzSek       ;40 Sektoren
                LD      BC,Buffer2      ;Adresse des Buffers für Track 2 (!!!)
                CALL    WriteTrack      ;ganzen Track schreiben

                LD      L,Track2        ;zweite Tracknummer nach HL
                LD      H,0
                LD      DE,AnzSek       ;40 Sektoren
                LD      BC,Buffer1      ;Adresse des Buffers für Track 1 (!!!)
                CALL    WriteTrack      ;ganzen Track schreiben
```



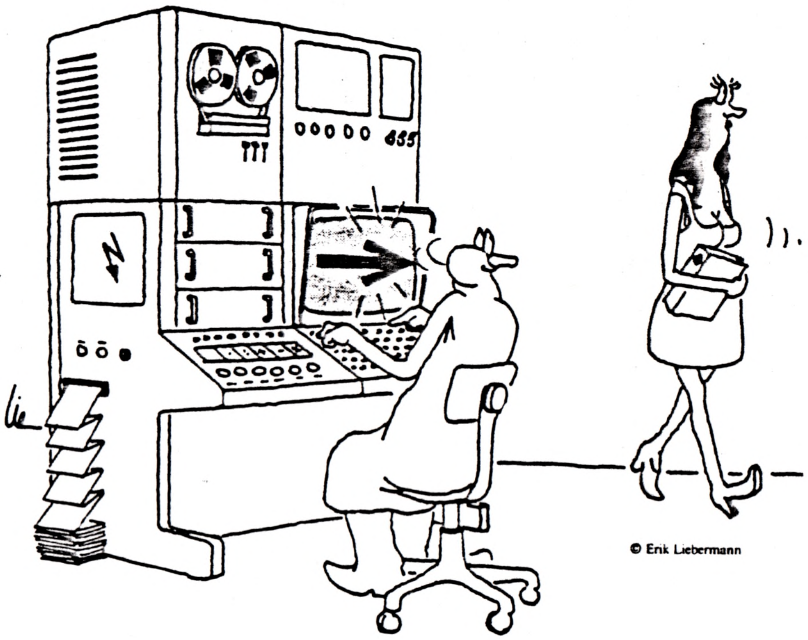
```
;       4. Akt: Schlußszene nach Vollzugs-Meldung
;       -----------------------------------------

Ende:
                LD      DE,OKMeld       ;Zeige auf OK-Meldung
                LD      C,PrtStr        ;ausgeben
                CALL    BDOS

;       kontrollierter Abgang aus dem Stück, auch bei Feuer (Fehler)
;       ------------------------------------------------------------

Abgang:
                LD      SP,0000         ;Stack zurück
RettSP          EQU     $-2
                JP      0               ;Warmstart

;-----------------------------------------------------------------------
;       *******************     Unterprogramme      ******************
;-----------------------------------------------------------------------
;
;       UP BiosCall: springt BIOS an
;       IN: HL=Offset ins BIOS
;           DE,BC=Parameter
;       OUT: je nach BIOS-Call
;       geänderte Register: ?
BiosCall:
                PUSH    DE              ;DE retten
                LD      DE,0000         ;Anfangsadresse des BIOS-3
BIOSAnspr       EQU     $-2
                ADD     HL,DE           ;Einsprungsadresse berechnen
                POP     DE              ;DE zurück
                JP      (HL)

;-----------------------------------------------------------------------
;
;       UP ReadTrack: liest einen Track ein
;       IN: HL: Tracknummer
;           DE: Anzahl Sektoren
;           BC: Bufferadresse
;       OUT: /
;       verändert: alles
ReadTrack:
                LD      (Track),HL      ;Tracknummer retten
                LD      (Zaehler),DE    ;Zähler retten
                LD      (BufAdr),BC     ;Bufferadr. retten
                LD      HL,Sektor1      ;erste Sektornummer
                LD      (Sektor),HL     ;Sektornummer retten
RLoop:
                CALL    SetParams
                LD      HL,ORead        ;Sektor lesen
                CALL    BiosCall
                OR      A
                JR      NZ,RError       ;Fehler
                CALL    NextSektor
                JR      NZ,RLoop
                RET
```

```
;      Fehler beim Lesen:

RError:     LD      HL,RTrNr          ;wo soll die Tracknummer hin?
            LD      DE,(Track)        ;Tracknummer laden
            CALL    SetAscii          ;in ASCII umwandeln
            LD      HL,RSekNr         ;wo soll die Nr. des fehlerhaften Sektors
                                      ;angezeigt werden?
            LD      DE,(Sektor)       ;Sektornummer laden
            CALL    SetAscii          ;umwandeln Sektornr. in ASCII
            LD      DE,RErrStr        ;Zeiger auf Fehlermeldung
            LD      C,PrtStr          ;ausgeben
            CALL    BDOS
            JP      Abgang            ;raus aus dem Programm

;-----------------------------------------------------------------------------
;
;      UP WriteTrack: schreibt einen Track raus
;      IN: HL: Tracknummer
;          DE: Anzahl Sektoren
;          BC: Bufferadresse
;      OUT: /
;      verändert: alles
WriteTrack:
            LD      (Track),HL        ;Tracknummer retten
            LD      (Zaehler),DE      ;Zähler retten
            LD      (BufAdr),BC       ;Bufferadr. retten
            LD      HL,Sektor1        ;erste Sektornummer
            LD      (Sektor),HL       ;Sektornummer retten
WLoop:
            CALL    SetParams
            LD      HL,OWrite         ;Sektor lesen
            LD      C,0
            CALL    BiosCall
            OR      A
            JR      NZ,WError         ;Fehler
            CALL    NextSektor
            JR      NZ,WLoop
            RET

;      Fehler beim Schreiben:

WError:     LD      HL,WTrNr          ;wo soll die Tracknummer hin?
            LD      DE,(Track)        ;Tracknummer laden
            CALL    SetAscii          ;in ASCII umwandeln
            LD      HL,WSekNr         ;wo soll die Nr. des fehlerhaften Sektors
                                      ;angezeigt werden?
            LD      DE,(Sektor)       ;Sektornummer laden
            CALL    SetAscii          ;umwandeln Sektornr. in ASCII
            LD      DE,WErrStr        ;Zeiger auf Fehlermeldung
            LD      C,PrtStr          ;ausgeben
            CALL    BDOS
            JP      Abgang            ;raus aus dem Programm
```



```
;-----------------------------------------------------------------------------
;
;      UP SetAscii: setzt Zahl in Ziffern um
;      IN: DE=in ASCII umzuwandelnde Zahl
;          HL=Zeiger, wo Zahl angezeigt werden soll (4 Stellen)
;      OUT: /
;      veränderte Register: AF,HL
SetAscii:
            LD      A,D               ;oberes Byte holen
            CALL    SetByte           ;und umwandeln
            LD      A,E               ;unteres Byte holen
SetByte:
            PUSH    AF                ;retten
            AND     0F0h              ;nur oberes Nibble
            RRCA                      ;nach unten
            RRCA
            RRCA
            RRCA
            CALL    SetNibble         ;in den String setzen
            POP     AF                ;Byte zurück
            AND     0Fh               ;nur unteres Nibble
SetNibble:
            CP      10                ;Ziffer ('0'..'9')?
            JR      C,Ziffer          ;ja
            ADD     A,7               ;nein, Buchstabe 'A'..'F'
Ziffer:     ADD     A,'0'             ;in '0'..'F'
            LD      (HL),A            ;in den String
            INC     HL                ;Zeiger auf nächste Stelle
            RET

;-----------------------------------------------------------------------------
;
;      UP SetParams: setzt Parameter für Lesen/Schreiben eines Sektors
;      IN: (Track)=Tracknummer
;          (Sektor)=Sektornummer
;          (BufAdr)=Buffer-Adresse
;      OUT:
;      verändert: alle
SetParams:
            LD      BC,(Track)        ;Tracknummer setzen
            LD      HL,OSetTrk        ;Offset
            CALL    BiosCall
            LD      BC,(Sektor)       ;Sektornummer übersetzen
            LD      DE,(TrnTab)
            LD      HL,OSecTrn
            CALL    BiosCall
            LD      B,H
            LD      C,L               ;Sektornummer setzen
            LD      HL,OSetSec        ;Offset
            CALL    BiosCall
            LD      BC,(BufAdr)       ;Buffer-Adresse
            LD      HL,OSetDma        ;Offset
            JP      BiosCall
```

```
;---------------------------------------------------------------------------
;
;       UP NextSektor: schaltet weiter zum nächsten Sektor
;       IN: (Sektor) = akt. Sektornr.
;           (Zaehler) = wieviele Sektoren noch zu lesen
;           (BufAdr) = Bufferadresse
;       OUT: (Sektor)+1
;            (Zaehler)-1
;            (BufAdr)+128
;            Zero-Flag: keinen Sektor mehr lesen
NextSektor:
            LD      HL,(Sektor)
            INC     HL
            LD      (Sektor),HL
            LD      HL,(BufAdr)
            LD      DE,128
            ADD     HL,DE
            LD      (BufAdr),HL
            LD      HL,(Zaehler)
            DEC     HL
            LD      (Zaehler),HL
            LD      A,H
            OR      L
            RET

;---------------------------------------------------------------------------
;---------------------------------------------------------------------------
;
;       Messages
;       --------

LwFrage:
            DB      CR,LF,CR,LF
            DB      'Achtung: Dieses Programm vertauscht zwei Tracks auf einer'
            DB      CR,LF
            DB      '          Diskette. Dies ist nur bei Alphatronic-P3-Disketten'
            DB      CR,LF
            DB      '          sinnvoll.'
            DB      CR,LF,CR,LF
            DB      'In welchem Laufwerk befindet sich die Alphatronic-Diskette?'
            DB      CR,LF
            DB      ' Wählen Sie: Laufwerk A..P oder Q für Abbruch: '
            DB      EOStr

OKMeld:     DB      CR,LF,CR,LF
            DB      'Alles roger! Ready and gone.'
            DB      CR,LF,EOStr

RErrStr:
            DB      CR,LF,CR,LF
            DB      'Fehler beim Lesen von Track '
RTrNr:      DB      '0000'
            DB      ', Sektor '
RSekNr:     DB      '0000'
            DB      '. Abbruch. Sorry!'
            DB      CR,LF,EOStr
```


```
WErrStr:
            DB      CR,LF,CR,LF
            DB      'Fehler beim Schreiben von Track '
WTrNr:      DB      '0000'
            DB      ', Sektor '
WSekNr:     DB      '0000'
            DB      '. Abbruch. Sorry!'
            DB      CR,LF,EOStr

;---------------------------------------------------------------------------
;
;       Buffer
;       ------

Buffer1:
            DS      AnzSek*128          ;Buffer für Track 1
Buffer2:
            DS      AnzSek*128          ;Buffer für Track 2


            DS      256                 ;Platz für Stack
NewStack:


            END     Start
```

Hallo alle GENIE III S -Besitzer !!

Wie bereits in der Ausgabe 34 des CLUB-INFO versprochen, veröffentliche ich heute den II.Teil des ADRESS-TEXT-90. Entschuldigt bitte,daß ich Euch nicht in der letzten Ausgabe den I.Teil zuerst gebracht habe.Dies geschah mit Absicht,da ich Euch erst einmal die Textverarbeitung vorstellen wollte um Euch für das ADRESS-TEXT-90 zu interessieren.

Die Anrufe nach der Veröffentlichung waren zwar gleich "NULL". Das soll aber noch nicht heißen, daß ich den Mut verliere weiterzumachen. Vielleicht ist doch jemand da, der mit ADRESS-TEXT-90 arbeiten möchte, wenn er erst einmal gemerkt hat, wie komforttabel ADRESS-TEXT-90 ist und wie einfach es ist, dieses auf seine eigenen Bedürfnisse abzuändern.

Nun zum Programm !

Damit dieses Programm überhaupt lauffähig ist, benötige ich außer der in Ausgabe 34 angegebenen Hardware auch die Original-Diskette des G-DOS 2.4 mit hochauflösender Graphik, mit nachfolgenden Files:

1.Druckertreiber: PRINTER/DRV - ITOH/DRV - STAR/DRV - PRSTAR/DRV - RADIX 151/DRV
ADRESS-TEXT-90 selber ist auf ITOH eingestellt,bei anderen Druckern,sind de Druckparameter zu ändern.

2.Basic: RDL/BASIC

3.DOS-Programme: JOB/CMD - SYSCOPY/CMD - PDAUTO/CMD - STD

4.Sonstige : Alle /SYS-Dateien und Overlay-Dateien

5.Schriften : SCRIPT/RZS - BLOCK/RZS - NORMAL/RZS - ASCII

Starteinstellung:

Mit dem Programm JOB/CMD wird eine JOB/Datei programmiert, die dann als AUTO-Befehl mit DO TEXT90ST/JOB aufgerufen wird. Die Reihenfolge der JOB-Programmierung ist wie folgt:
SYSCOPY - PDAUTO - ITOH - ZL STD - 80 - RDL/BASIC - RUN"TEXT90ST/BAS"

Nach diesem Start erscheint auf dem Bildschirm Bild 2.

Das Programm POM sowie WPS werde ich aus arbeitsrechtlichen Gründen nicht veröffentlichen. Meine Veröffentlichung bezieht sich nur auf das PRI-Programm. Der Ablauf des Programms erfolgt im Bildschirmdialog.

Zum Listing: die Passwörter können selber abgeändert werden.

Nach Aufruf von < PRI > - < DATA > kommt Ihr in das < JOHMENü/BAS >-Programm. Es erscheint das Bild 1 mit seinem Hauptmenü.
Es stehen nun folgende Programme zur Verfügung:

1.) AV Adressen-Verwaltung -->> Adressenverwaltung mit Sortierprogramm nach Familienname bzw. Firmenbezeichnung max. 2000 Adressenblöcke
2.) AK Adressenaufkleber ---->> Adressenaufkleber drucken mit 2verschiedenen Formaten

3.) BR Briefe-drucken ------->> Drucken von Briefen mit oder ohne Serienbriefe, die vorher mit der Textverarbeitung abgespeichert wurden. Anschreiben von bestimmten Personengruppen in bestimmten PLZ-Bereichen oder unter Eingabe einer bestimmten Adresse etc.
4.) VK Videokasetten -------->> Verwaltung von Videokasetten nach einem bestimmten System mit Sortierprogramm nach Kasetten etc.
5.) DA Diskettenaufkleber --->> Drucken von Diskettenaufklebern
6.) MV Musikkasetten -------->> Verwaltung von Musikkasetten max.1500 Titel mit Sortierprogramm
7.) TV Textverarbeitung ----->> komfortable Textverarbeitung, die in Ausgabe 34 schon veröffentlicht wurde max 160 Zeilen pro Brief
8.) HY Hypotheken ----------->> Berechnung von Hypotheken und Tilgungen bei fester Rate oder var. Rate, fester Zinssatz oder var. Zinssatz etc.
9.) BI Biorythumus ---------->> Berechnung und Ausdruck des Biorythumus

Nur obige Programm werden von mir veröffentlicht, alle anderen im M E N ü angegebenen Programme sind für mich rein privat.

Zum Listing:

Und nun viel Spaß beim eintippen. In der nächsten Ausgabe folgt der III.Teil mit Veröffentlichung der ADRESS-VERWALTUNG.

Originalbeschreibung eines Spiels (aus einem Computer-Magazin):
»In einem zur Hälfte fertiggestellten Rohbau, dessen Etagen durch Leitern verbunden sind, werden Sie von einer Horde wildgewordener Äpfel gejagt. Um den Ungeheuern zu entgehen, müssen Sie Löcher graben, in die die Äpfel reinfallen – doch das ist nicht alles . . .«

Wer kennt es?

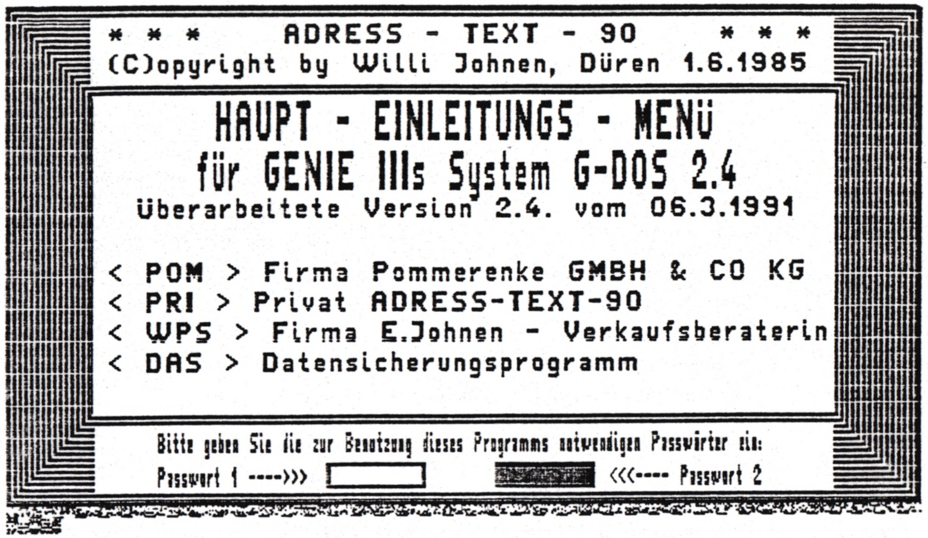
```
10 CLS:CMD"80"
20 '#>> Programmname      << JOHMENÜ/BAS >>                 <<#
30 '#>> Menüprogramm zum ADRESS-TEXT-90 - Privat            <<#
40 '#>> Stand: 03.10.1990                                   <<#
50 '#>> (C) by W.Johnen,5160 Düren - erstellt am 1.4.1984   <<#
60 '#>> Keine hochauflösende Graphik                        <<#
70 CLS:CMD"80":HOFF:HCLS:CLEAR1000:DEFSTRA-H
80 AA=CHR$(128):AD=CHR$(131):AM=CHR$(140):AN=CHR$(141):AO=CHR$(142):AP=CHR$(143):AV=CHR$(149):AJ
=CHR$(137):AG=CHR$(134):AQ=CHR$(144):AX=CHR$(151)
90 BC=CHR$(156):BS=CHR$(172):BD=CHR$(157):BU=CHR$(174):BZ=CHR$(179):BQ=CHR$(170):BR=CHR$(171):BW
=CHR$(176):BY=CHR$(178):BX=CHR$(177)
```

```
                              HAUPT-MENÜ

< av > Adressen-Verwaltung              | < tv > Text-Verarbeitung
< ak > Adressenaufkleber drucken        | < ar > Artikelnr.-Verzeichnis drucken
<    >                                  | <    >
<    >                                  | <    >

< br > Briefe - drucken                 | <    >
<    >                                  | <    >
<    >                                  | <    >
<    >                                  | <    >

< vk > Videokasetten-Verwaltung         | < hy > Hypotheken-Berechnung
< da > Diskettenaufkleber drucken       | < dl > DIA-Liste drucken
< le > Lebenslauf drucken               | < ka > Karnevalszug-Verträge drucken

< mv > Musikkasetten-Verwaltung         | < bi > Biorythmus - Programm
<    >                                  | < pe > Programmende

                 Welches Programm wünschen Sie ? **
```

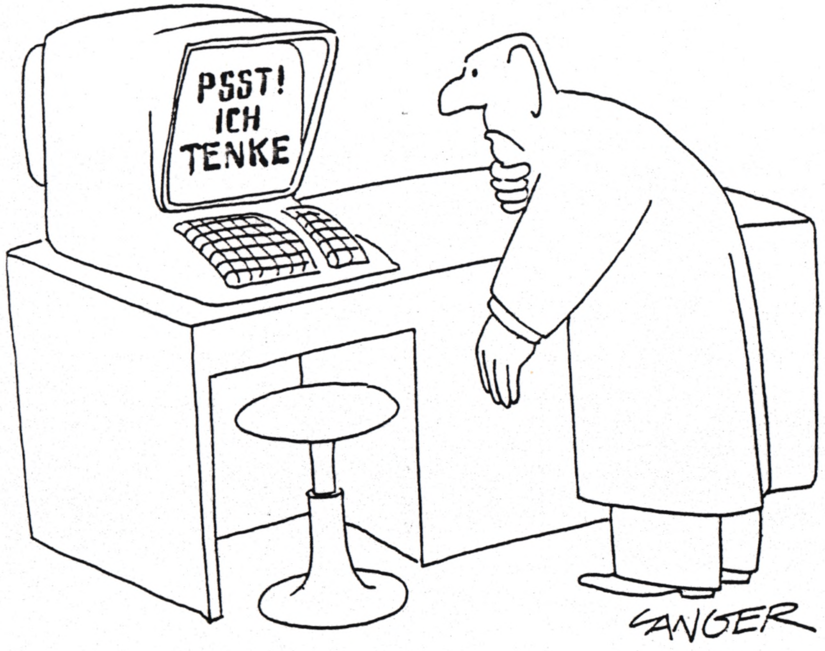
-------------------- Bild 1 zum Programm < JOHMENÜ/BAS -------------------------


```
100 CL=CHR$(191):CI=CHR$(188)
110 CA=STRING$(38,AM)+BC+STRING$(39,AM):CB=STRING$(78,AM):CC=BZ+BZ+BZ+BZ:CD=AD+BZ+AD+AD+BZ
120 PRINTCL;STRING$(14,AD);CD;AD;CC;CD;AD;CC;AD;CC;AD;AD;AD;CD;AD;CC;CD;CD;STRING$(16,AD);CL;
130 PRINTCL;STRING$(15,AA);CL;AM;AM;CL;AA;CL;AM;AM;CL;AA;CL;AA;AA;CL;AA;CL;AM;AM;AP;AA;AA;BQ;AV;
AA;AA;AM;AM;AA;CL;AJ;AG;CL;AA;CL;AM;AM;AA;AA;CL;AJ;AQ;CL;AA;CI;AA;AA;CI;STRING$(16,AA);CL;
140 PRINTCL;STRING$(15,BW);BZ;BW;BW;BZ;BW;BZ;BW;BW;BZ;BW;BZ;BZ;BZ;BZ;BW;BZ;BW;BW;BW;BW;BW;BY;BX;
BW;BW;BW;BW;BW;BZ;BW;BW;BZ;BW;BZ;BZ;BZ;BZ;BW;BZ;BW;BY;BZ;BW;BZ;BZ;BZ;BZ;STRING$(16,BW);CL;
150 PRINTBC;CB;BS;
160 PRINTAV;"< av > Adressen-Verwaltung              ";AV;"< tv > Text-Verarbeitung
";BQ;
170 PRINTAV;"< ak > Adressenaufkleber drucken        ";AV;"< ar > Artikelnr.-Verzeichnis drucken
";BQ;
180 PRINTAV;"<    >                                  ";AV;"<    >
";BQ;
190 PRINTAV;"<    >                                  ";AV;"<    >
";BQ;
200 PRINTBD;CB;BU;
210 PRINTAV;"< br > Briefe - drucken                 ";AV;"<    >
";BQ;
220 PRINTAV;"<    >                                  ";AV;"<    >
";BQ;
230 PRINTAV;"<    >                                  ";AV;"<    >
";BQ;
240 PRINTAV;"<    >                                  ";AV;"<    >
";BQ;
250 PRINTBD;CB;BU;
260 PRINTAV;"< vk > Videokasetten-Verwaltung         ";AV;"< hy > Hypotheken-Berechnung
";BQ;
270 PRINTAV;"< da > Diskettenaufkleber drucken       ";AV;"< dl > DIA-Liste drucken
";BQ;
280 PRINTAV;"< le > Lebenslauf drucken               ";AV;"< ka > Karnevalszug-Verträge drucken
";BQ;
290 PRINTBD;CB;BU;
300 PRINTAV;"< mv > Musikkasetten-Verwaltung         ";AV;"< bi > Biorythmus - Programm
";BQ;
310 PRINTAV;"<    >                                  ";AV;"< pe > Programmende
";BQ;
320 PRINTAN;CB;AO;
330 PRINTBC;CB;BS;
340 PRINTAV;"                  Welches Programm wünschen Sie ?                  ";BQ;
350 PRINTAN;CB;AO;
360 PRINT§1815,STRING$(2,42);"      ";:PRINT§1815,;:LINEINPUTA
370 PRINTAN;CB;AO;
380 IFLEN(A)<>2THEN360
390 HNEG
400 IFA="av"THENRUN"JOHADRES/BAS"
410 IFA="ak"THENRUN"JOHADREK/BAS"
420 IFA="br"THENRUN"JOHBRIEF/BAS"
430 IFA="tv"THENRUN"JOHTEXTV/BAS"
440 IFA="da"THENRUN"JOHDISKE/BAS"
450 IFA="ka"THENRUN"JOHKARNE/BAS"
460 IFA="le"THENRUN"JOHLEBEN/BAS"
470 IFA="mv"THENRUN"JOHMUSIK/BAS"
480 IFA="dl"THENRUN"JOHDIASV/BAS"
490 IFA="vk"THENRUN"JOHVIDEO/BAS"
500 IFA="pe"THENRUN"TEXT90ST/BAS"
510 IFA="hy"THENRUN"HYPOTH80/BAS"
520 IFA="bi"THENRUN"BIORYT80/BAS"
530 GOTO340
```

```
10 CMD"§,KEINE"
20 '#>> Programmname      << TEXT90ST/BAS >>                    <<#
30 '#>> Startprogramm zum ADRESS-TEXT-90                        <<#
40 '#>> (C) by Willi Johnen,5160 Düren  - erstellt 1990         <<#
50 '#>> Stand: 19.07.1991      17.00 Uhr                        <<#
60 '#>> Mit Hochauflösender Graphik                             <<#
70 '#>> O H N E     H A R D D I S K                             <<#
80 CLEAR2000:DEFSTRA-C,G:DEFINTN,X,Z,W
90 TT=0
100 CLS:HOFF:HCLS:HON:HCODE"SCRIPT/RZS"
110 W1=0:W2=0:W3=639:W4=0:W5=639:W6=249:W7=0:W8=249:W9=0:W0=0
120 FORX=1TO15
130 W1=W1+4:W2=W2+3:W3=W3-4:W4=W4+3:W5=W5-4:W6=W6-3:W7=W7+4:W8=W8-3:W9=W9+4:W0=W0+3
140 HMOVEW1,W2,1:HDRAWW3,W4,1,:HDRAWW5,W6,1,:HDRAWW7,W8,1,:HDRAWW9,W0,1
150 NEXTX
160 HNEG
170 HCAR059,211,578,242,1
180 HCAR062,6,578,38,1
190 HCODE"BLOCK/RZS"
200 HCAR059,46,578,203,2
210 HBIG70,225,2,1,1,"* * *    ADRESS - TEXT - 90    * * * "
220 HBIG70,210,2,1,1,"(C)opyright by Willi Johnen, Düren 1.6.1985"
230 HBIG145,170,2,1,2,"HAUPT - EINLEITUNGS - MENÜ"
240 HBIG132,145,2,1,2,"für GENIE IIIs System G-DOS 2.4"
250 HBIG90,137,2,1,1,"überarbeitete Version 2.4. vom 19.07.1991"
260 HBIG70,123,2,1,1,"ACHTUNG! Diese Version ist ohne HARD-DISK"
270 HBIG70,90,2,1,1,"< PRI > Privat ADRESS-TEXT-90"
280 HBIG70,75,2,1,1,"< WPS > Firma E.Johnen - Verkaufsberaterin"
290 HBIG70,60,2,1,1,"< DAS > Datensicherungsprogramm"
300 HCAR062,6,578,38,1
310 HBIG105,20,2,0,1,"Bitte geben Sie die zur Benutzung dieses Programms notwendigen Passwörter
ein:"
320 HBIG105,3,2,0,1,"Passwort 1 ---->>>                          <<<---- Passwort 2"
330 HCAR0223,8,340,20,2:HCAR0293,8,410,20,2
340 '### --------------- Programmsicherung -----------------<<#
350 HCAR0227,10,288,18,2:TT=TT+1:PRINT§1871,;:LINEINPUTA
360 HCAR0345,10,406,18,2:PRINT§1885,;:LINEINPUTB
370 IFA="pri"ANDB="data"THENCLS:GOTO510
380 IFA="wps"ANDB="data"THENCLS:GOTO520
390 IFA="das"ANDB="data"THENCLS:GOTO530
400 IFTT<4THENHOFF:HNEG:GOTO100
410 CLS:HCAR062,46,578,160,1
420 HBIG78,125,2,2,1,"    ! ACHTUNG !"
430 HBIG200,100,2,1,1,"Fehlerhafte Codeeingabe"
440 HBIG100,85,2,1,1,"Dieses Programm ist laut Bestimmungen"
450 HBIG125,70,2,1,1,"des Bundesdatenschutzgesetzes gegen"
460 HBIG160,55,2,1,1,"unbefugte Benutzung gesichert"
470 HCAR062,6,578,38,1
480 HBIG82,20,2,1,1,"Sie sind nicht der berechtigte Benutzer !!"
490 HBIG72,3,2,1,1,"Ein erneuter Versuch zerstört das Programm"
500 FORX=1TO1250:NEXTX:GOTO470
510 HCLS:CLEAR50:RUN"JOHMENü/BAS"
520 HCLS:CLEAR50:RUN"JOHSELST/BAS"
530 HCLS:CLEAR50:RUN"TEXT90SI/BAS"
```

------------------------- Bild 2 zum Programm < TEXT90ST/BAS -----------------



© Heinz Langer

Hallo Club 80´er,

auch ich gehöre zu den Anwendern die sich verstärkt mit MS-Dos-Rechnern befassen. Wer schon einmal mit Windows 3.0 und Winword gearbeitet hat, kann dies auch verstehen. Fast alle Programme laufen unter Windows 3.0 einwandfrei auf einem Neat-AT. Speicherplatzsorgen sind aus diesem Grund auf einem AT-Rechner kein Thema. Multitask ist bei dem Rechner nicht drin, aber dafür haben die 386´iger im Multitask-Betrieb auch ihre Probleme. Ein fehlerhaftes Programm kann den Multitask-Betrieb zum Erliegen bringen und damit sind alle Inhalte verloren, der Rechner hängt.

Windows 3.0 unter DR-Dos 5.0

Allen Gerüchten zum Trotz, auch DR-Dos 5.0 läuft unter Windows 3.0 einwandfrei. Es gibt nur wenige Programme (z.B einige Debugger) die unter DR-Dos nicht hochgefahren werden können. Vereinzelt kommt es auch vor, das einige Programme mit Cache.EXE nicht zusammenarbeiten wollen. Weiter ist es möglich auf einem Neat-AT auch Shadow-Ram einzustellen bezw. mit Hi-Load auch Programme über die Adressen der Grafik-Karte hochzuladen. Wie sich herausstellte ist die Adressierung des Protect-Modes nicht mit MS-Dos kompatibel. Aus diesem Grund darf in der Config.SYS der Befehl nicht B=FFFF lauten sondern B=auto. Um die Adressleitung A 20 anzusteuern muß das File Himem.SYS verwendet werden. Unter diesen Umständen stehen dann auch ca. 621 KB unter DR-DOS 5.0 zur Verfügung, wenn Windows 3.0 hochgefahren werden soll.

Patch für DR-Dos 5.0 Kommando-Interpreter

Vielleicht ist es dem einen oder anderern User auch schon passiert das er die Umgebungsvariablen des Command.Com erhöhen mußte. Dies geschieht mit dem Parameter /E: . Springt man aus der Shell einer Anwendung heraus, zwingt Dr-Dos wie MS-Dos zur Verwendung der Standardumgebungsvariablen (256). Dieser Wert ist im Command.Com intern festgelegt. Um diesen Wert zu ändern muß man den Kommando-Interpreter "patchen".Dies kann mit dem SID86 wie folgt geschehen:

```
Sid86 starten
# r command.com
# sw16ff
```

Zeigt der Sid86 den Wert (Standardgröße des Umgebungsspeicher) der Adresse in Byte 100 hex, decimal 256 so kann das Wort an dieser Adresse geändert werden. Um beispielsweise 512 Bytes für die Umgebungsvariable zu vereinbaren,wird jetzt der Wert 0200 eingegeben und mit Enter quititert.

```
. Enter
#w command.com
#q
```

Danach wird der Command.com zum Testen aufgerufen. Dieser Patch sollte natürlich vorsichtshalber nur auf einem Betriebssystem gemacht werden, das sich auf der Diskette befindet.

ADDPath

Durch immer anwenderfreundlichere Programme werden leider die Festplatten-Kapazitäten auch nach oben verlagert. Durch Verschachtelung von Root und mehreren Subdirectorys ist der Path bald erschöpft. DR-Dos wie MS-Dos sind nur in der Lage 128 Buchstaben im Path zu verwalten. Alle folgenden Anweisungen werden aus diesem Grund ignoriert. Dies würde bedeuten das der Anwender unter Umständen mehrere Bat-Dateien anlegen muß oder andere Spielereien zur Anwendung bringen muß. Doch es geht auch einfacher mit ADDpath.

Listing und Files sind aus der MC. Diese stelle ich dem Club80 zu Verfügung.

z.B. könnte die Autoexec.bat so aussehen:

```
@ECHO OFF
C:\DOS\KEYB GR
Path=C:\;C:\DRDOS
SET ADDPATH1=c:\
SET ADDPATH2=C:\
SET ADDPATH3=C:\
C:\TOOLS\ADDPATH.EXE
PROMPT $P$G
```

```
;  ***************************************
;  *  ADDPATH.ASM                   *
;  *  PATH länger als 128 Zeichen   *
;  *                                *
;  *  27.März 1991  Peter Engels    *
;  ***************************************
;
; assemblieren: MASM / TASM  ADDPATH
;          LINK / TLINK ADDPATH
;
                    dosseg
                    .model small
                    .stack 100h

NULL                equ 0
DOSINT              equ int 21h

                    .DATA
parent_env_seg      dw 0
path                db 'PATH='
addpath             db 'ADDPATH'
addpath_nr  db '0='
addpath_flag        db 0
path_ofs     dw 0
addpath_ofsdw 0

copyr_msg  db  'ADDPATH V 1.0  by: P.Engels  27.03.1991'
                    db  10,13,'$'
dos_ver_error       db  10,13,'DOS-Version ab 3.0 erforderlich !'
                    db  10,13,'$'
nopath_msgdb  10,13,'Es wurde kein "PATH=" Eintrag gefunden !'
                    db  10,13,'$'
noaddpath_msg       db  10,13,'Es wurde kein "ADDPATHx=" Eintrag gefunden !'
                    db  10,13,'$'
follow_msg  db  10,13,'Der "PATH" und alle "ADDPATHx="-Einträge müssen '
                    db  'direkt aufeinander folgen !'
                    db  10,13,'$'
total_msg   db  10,13,'Der Suchpfad wurde durch '
total_nr    db  '0 ADDPATH-Einträge ergänzt'
                    db  10,13,'$'

                    .CODE
start:              mov ax,@DATA            ;Setze die Segmente
                    mov ds,ax
                    lea dx,copyr_msg        ;Titel ausgeben
                    mov ah,9
                    DOSINT

                    mov ah,30h              ;DOS-Version holen
                    DOSINT                  ;über INT21h Funktion 30h
                    cmp al,3
                    jge search_env          ;grösser/gleich 3 -> ok
                    lea dx,dos_ver_error    ;wenn DOS-Version < 3.0
                    mov ah,9                ;dann Fehler ausgeben
                    DOSINT
                    jmp ende
;***************************************
;Suche Segment des Parent-Environment
;***************************************
search_env: mov ah,62h              ;hole PSP-Segment nach BX
                    DOSINT                  ;über INT 21h Funktion 62h

                    mov es,bx               ;Word 16h im PSP
                    mov bx,es:16h           ;zeigt auf Parent-PSP

                    mov es,bx
                    mov bx,es:2ch           ;Word 2Ch im Parent-PSP
                    mov parent_env_seg,bx   ;zeigt auf Parent-Environment
```

```
                    mov es,bx               ;ES: = Segment des Parent-Environments
;***************************************
;Suche PATH-Eintrag im Parent-Environment
;***************************************
                    mov ax,0ffffh           ;Zaehler initialisieren
                    mov dl,NULL
next_byte:  inc ax
                    mov di,ax               ;ES:DI zeigen in Parent-Environment
                    cmp es:[di],dl                  ;bei zwei NULL-Bytes ist das
                    jne no_zero             ;Ende des Environments erreicht
                    inc di
                    cmp es:[di],dl
                    jne ok                          ;Ende noch nicht erreicht ->
weiter
                    jmp no_path_found       ;Ende erreicht -> Fehlermeldung
ok:                 dec di

no_zero:    lea si,path             ;DS:SI zeigen auf 'PATH=' Text
                    mov  cx,addpath-path    ;Länge=5
                    repe cmpsb              ;suche PATH=

                    jcxz path_found                 ;gefunden !
                    jne next_byte                   ;nein nächstes Byte

path_found: mov path_ofs,ax                 ;Offset des Path-Eintrags speichern
;***************************************
;Suche ADDPATHx= im Parent-Environment
;***************************************
search_addpath:     mov es,parent_env_seg
                    inc addpath_nr                  ;Nr. des ADDPATH erhöhen
                    mov ax,0ffffh           ;AX initialisieren
                    mov dl,NULL
next_byte1: inc ax
                    mov di,ax               ;ES:DI zeigen in Parent-Environment
                    cmp es:[di],dl                  ;bei zwei NULL-Bytes ist das
                    jne no_zero1                    ;Ende des Environments
erreicht
                    inc di
                    cmp es:[di],dl
                    je no_addpath_found     ;Ende erreicht -> not found
                    dec di

no_zero1:   lea si,addpath          ;DS:SI zeigen auf ADDPATHx= Text
                    mov  cx,addpath_flag-addpath ;Länge
                    repe cmpsb              ;suche ADDPATHx=

                    jcxz addpath_found      ;gefunden !
                    jne next_byte1                  ;nein nächstes Byte

addpath_found:      mov addpath_ofs,ax      ;Offset des ADDPATH-Eintrags speichern
;***************************************
;suche das Ende der PATH-Variablen
;***************************************
                    mov es,parent_env_seg   ;Ende der PATH-Variablen wird
                    mov di,path_ofs                 ;durch NULL-Byte
                    mov al,NULL             ;markiert, also nach
search_pathend:     inc di                          ;NULL-Byte suchen
                    cmp es:[di],al
                    jne search_pathend      ;noch nicht gefunden -> weitersuchen
;*****************************************************
;testen ob ADDPATH direkt hinter PATH-Variablen steht
;*****************************************************
                    inc di                          ;wenn Ende+1 von PATH
ungleich
                    cmp di,addpath_ofs      ;Anfang von ADDPATHx= ist
                    jne follow_path                 ;dann Fehler ausgeben

                    dec di                          ;Ende-Kennung des PATH-
```

```
                    mov dl,';'              ;durch ';' ersetzen
                    mov es:[di],dl
;*****************************************************************
;Rest des Environments 9 Bytes nach links schieben
;somit wird der Inhalt von ADDPATHx ans Ende der PATH-Variablen gehängt
;*****************************************************************
                    mov es,parent_env_seg   ;ES:DI zeigen auf MOVE-Ziel
                    mov di,addpath_ofs      ;= Start von ADDPATHx =
                    push ds                         ;DS retten
                    mov ds,parent_env_seg   ;DS:SI zeigen auf MOVE-Quelle
                    mov ax,di
                    mov bx,addpath_flag-addpath
                    add ax,bx
                    mov si,ax               ;SI = DI + len('ADDPATHx=')
loop1:              movsb                           ;Byte verschieben
                    mov al,es:[di]                  ;Testen ob das Ende
                    inc di                          ;des Environments erreicht
                    add al,es:[di]                  ;Ende = zwei NULL-Bytes
                    dec di
                    cmp al,NULL
                    jne loop1
                    movsb                           ;das zweite NULL-Bytes auch
schieben
                    pop ds                          ;DS wieder zurückholen
;***************************************
;Vorbereitung für nächste ADDPATH-Suche
;***************************************
                    inc total_nr            ;Zaehler erhöhen
                    inc addpath_flag        ;FLAG <> NULL setzen
                    jmp search_addpath      ;nächste ADDPATH Variable suchen
;***********************************
;Routinen zur Ausgabe vom Meldungen
;***********************************
no_path_found:      lea dx,nopath_msg       ;Fehlermeldung ausgeben wenn
                    mov ah,9                ;PATH-Eintrag nicht gefunden wurde
                    DOSINT
                    jmp ende

no_addpath_found:
                    mov al,addpath_flag     ;Anzahl der angehängten Pfade
                    cmp al,NULL             ;ausgeben, wenn mindestens ein
                    jne give_total                  ;ADDPATH-Eintrag bearbeitet
wurde.
                    lea dx,noaddpath_msg    ;kein ADDPATH-Eintrag gefunden ->
                    mov ah,9                ;Fehlermeldung ausgeben
                    DOSINT
                    jmp ende

follow_path: lea dx,follow_msg      ;Fehlermeldung ausgeben wenn
                    mov ah,9                ;ADDPATH= nicht auf PATH= folgt
                    DOSINT
                    mov al,addpath_flag
                    cmp al,NULL             ;falls schon min. ein ADDPATH
                    je ende                         ;angehängt wurde, Meldung
ausgeben
give_total:   lea dx,total_msg
                    mov ah,9
                    DOSINT

;*****************
;Programm beenden
;*****************
ende:               mov ax,4c00h                    ;beende Programm
                    DOSINT

                    end start
```

SERIELL mit PIP

Hallo Clubmitglieder !

Vor einiger Zeit traf sich der Club auf der Jahreshauptversammlung , zumindest war eine anberaumt worden.
Viele von Euch kennen den Ablauf eines Clubtreffens und auch diesmal verlief das Treffen wie gewohnt. Andreas hatte seinen Genie 3s inclusive Lötkolben und Oskar dabei , diverse REH-280 Rechner im Siemens-Gehäuse waren gekommen , Hartmuds M4P und Christopher mit seinem TANDY Modell II inclusive drei 8"-Drives waren dabei und viel Zeit zum fachsimpeln.
Meine Probleme fingen mit dem TANDY Modell II an, denn Christopher wollte ein Terminalprogramm auf dem TANDY haben um Dateien vom Modell 4P auf das Modell II zu übertragen.
Nun , der Genie 3s von Andreas wollte partut nicht mit den 8"-Drives umgehen, und ansonsten war auf dem TANDY Modell II nichts DFUe-mäßiges vorhanden.

Außer ... PIP.

Tja , mit PIP, da ging doch auch was mit der seriellen Verstöpselung. Wenn man genau weiß wies geht ist es einfach , aber ich möchte hier auch erzählen wie man es nicht machen sollte.
Man nehme nicht auf der einen Seite einen KAYPRO II mit TERM und auf der anderen Seite einen TANDY MII und PIP. Beide verstehen sich (gleiche Baudrate vorausgesetz) prächtig, aber wenn der TANDY speichert sendet das Terminalprogramm auf dem Kaypro fleißig weiter, wodurch natürlich einige BITs fehlen und das übertragene Programm kaum laufen dürfte.
Mir ging es so und eigendlich hätte ich dies wissen müßen, aber da ich den KAYPRO auf dem Clubtreffen erst bekam war ich natürlich ganz neugierig auf diesen Rechner und beachtete so eine wichtige Voraussetzung für die serielle Übertragung binärer Daten nicht, welche im normalen Terminalmodus selten gebraucht wird.
Es muß ein Handshake her. Als ich mir dies alles wieder klar gemacht hatte war schon einiges an kohlensäurehaltigem Gebräu die Kehle hinunter, alkohohlfrei versteht sich und ein Vormittag hiermit verbracht.
PIP gehört zur Standardsoftware eines CP/M - Rechners und kann auch die seriellen Schnittstellen bedienen, vorausgesetzt PIP (für CP/M2.2) ist hierfür gepatcht. PIP für CP/M 3.x macht dies ohne Änderungen, da die Devicetreiber AUXIN und AUXOUT die serielle Schnittstelle bedienen und mit dem Programm DEVICE die Parameter der Schnittstellen eingestellt werden können.
Leider ist bei der seriellen Datenübertragung das Controll-Z die Dateiendmarke und in einer binären Datei ist oft ein Controll-Z als Bildschirmsteuerzeichen oder sonstiges mitten im Programmcode enthalten.
Also geht es nicht so einfach mit der Übertragung einer binären Datei, welche schon vorzeitig durch Controll-Z beendet wird, aber es gibt da Abhilfe.

Die binäre Datei wird in eine HEX-Datei umgewandelt.

Mit dem Programm **UNLOAD** wird aus einer binären Datei eine Filename.HEX Datei erstellt.

Die HEX-Datei besteht nur aus hexadezimalen Zahlen und enthält so keine bei der serielle Übertragung störenden Zeichen.
Nach der korrekten Übertragung auf den Zielrechner wird die HEX-Datei mit **LOAD**, **HEXCOM**, **DDT(Z)** oder dergleichen in eine COM-Datei umgewandelt. LOAD und DDT wird dem CP/M2.x mitgeliefert und HEXCOM ist dem CP/M3.x beigefügt.

Die Syntax dieser Programme ist einfach:

Befehl	Eingabe		Ergebniss
UNLOAD	Filename.COM	----->	Filename.HEX
LOAD	Filename.HEX	----->	Filename.COM
HEXCOM	Filename.HEX	----->	Filename.COM

DDT(Z) und SID(Z) laden auch HEX-Dateien und speichern diese Dateien dann als COM-Datei ab. Bei DDT muß mit SAVE die Datei abgespeichert werden, während SID den Befehl W(rite) mitbringt.
Ihr seht es gibt einige Möglichkeiten eine HEX/COM - Datei zu bearbeiten.
Diese **HEX**-Datei ist also Voraussetzung für unsere problemlose serielle Datenübertragung. Nachdem dies nun bekannt ist und auf beiden Rechnern PIP zur Verfügung steht sucht man sich ein NULL-MODEM Kabel und meistens klappts. Falls beide Rechner die gleiche Belegung der RS232 Schnittstelle haben, sollte eine Probeübertragung klappen, ansonsten muß man an einem der Stecker wahrscheinlich PIN 2(TXD) und PIN 3(RXD) tauschen.
Hier noch mal zur Erinnnerung - zwei Rechner sind vorhanden, gleiche Baudzahl eingestellt, Kabel ist richtig konfiguriert und nun sollte ein ASCII-Text zum Schnittstellentest übertragen werden. Glücklicherweise war das PIP des TANDY MII schon entsprechend an die serielle Schnittstelle angepaßt und mein Rechner ist ein CP/M3 Gerät, und dessen PIP kann die serielle Schnittstelle von Haus aus bedienen.
Die Änderung an PIP unter CP/M 2.x beschreibe ich am Ende es Artikels.

<u>Testen der seriellen Verbindung:</u>

SENDER:	**PIP PUN:=b:Filename[EB]**
EMPFÄNGER:	**PIP b:Filename:=RDR:[EB]**

<u>Bedeutung der Optionen E und B:</u>
PIP schickt mit der OPTION "E" die Datei auch auf das angeschlossene Display, was aber nur bei Textdateien einen Sinn ergibt und hier nur zum Testen der ASCII-Übertragung gebraucht wird.
Mit der Option "B" arbeitet PIP im **BLOCK-Modus** und die übertragene Datei wird in Blöcken zu je 256 Bytes gesendet. Die Daten werden blockweise in den Puffer gelesen und zum Zielgerät übertragen, bevor ein neuer Block gelesen werden kann. Das Blockende-Zeichen ist der ASCII-Code 19, d.h. CTRL-S oder auch X-OFF.

Der Block-Modus ist also ein X-ON/X-OFF Protokoll und verhindert, daß beim Speichern der Daten welche verloren gehen, bzw. nicht empfangen werden.
Also, bei Datenübertragung mit PIP über die RS232 Schnittstelle nur im BLOCK-Modus arbeiten.
Falls der oben gezeigt "Test" der Verbindung fehlerfrei funktioniert hat, kann jetzt die eigendliche Datei im HEX-Format übertragen , und die HEX-Datei anschließend auf dem Zielrechner mit HEXCOM oder LOAD in eine COM-Datei umgewandelt werden.
Ein kleines Problem haben wir aber nur kurz angesprochen und wollen das hier nachholen.
PIP, unter CP/M3, ist für AUXIN/AUXOUT-Device generiert worden. Jeder CP/M 3.x (CP/M+) Rechner kann mit PIP für CP/M 3.0 die serielle Schnittstelle bedienen.
Unter CP/M 2.x haben wir leider nicht diese Standardisierung und PIP muß von der Lage der seriellen Schnittstelle informiert werden.
Im allgemeinen wird der serielle Baustein über IN/OUT-PORTS angesprochen und diese sind hoffentlich in den Handbüchern zum Rechner beschrieben.
PIP muß also wissen an welcher Portadresse der serielle Baustein angesprochen wird und wann der Baustein dazu in der Lage ist. Hierfür ist in PIP im Bereich von 10AH bis 1FFH Platz zur Verfügung gestellt worden.
Es wurden zwei Labels vorgesehen, und zwar INP:(INPUT) an der Stelle 103H. PIP springt, um Daten vom zu übernehmen, die Adresse 103H per CALL-Befehl an. Dann wird in 110H eine kleine Routine durchgeführt, welche das eingetroffene Zeichen an der Stelle 109H ablegt und mit RET nach PIP zurückspringt. Das Zerro-Bit (BIT 8) muß auf Null gesetzt sein.
Ähnliches geschieht auch beim Senden der Daten. Hier wird das Label OUT:(OUTPUT) an der Adresse 106H angesprungen , eine kleine Routine auf 120H angesprungen , welche das zu sendende Zeichen aus Register C liest und an die serielle Schnittstelle weitergibt.
Natürlich müssen die Baudraten der gekoppelten Rechner stimmen, aber da es die verschiedensten seriellen Bausteine gibt kann hier nur empfohlen werden, über ein SETUP Programm oder dergleichen die Baudrate vorher einzustellen. Meist sind Programme vorhanden, mit denen der Druckeranschluß, die Laufwerke und die serielle Schnittstelle konfiguriert werden können. Falls nichts dergleichen machbar ist muß man einen Fachmann fragen und ist in diesem Artikel nicht weiter ausgeführt.

Hier noch eine kleine Routine mit welcher PIP gepatcht werden kann. Damit es für alle nützlich ist auch in 8080 Code:
Die hier genutzten Portadressen sind für den TATUNG TPC2000 gültig.

Prozessor 8080:

```
IN:
     103H      JMP 110H   ; Sprung zur Eingabeunterroutine
     106H      JMP 120H   ; Sprung zur Ausgabeunterroutine
     110H      IN  1DH    ; Port 1DH liefert das Statuswort
     112H      ANI 2      ; Testen ob Zeichen eingetroffen
     114H      JZ  110H   ; Schleife bis Zeichen da ist
     117H      IN  1CH    ; Port 1CH liefert das Zeichen von
                          ; der seriellen Schnittstelle
     119H      ANI 7FH    ; höchstes Bit ausblenden
     11BH      STA 109H   ; Empfangenes Zeichen in 109H ablegen
     11EH      RET        ; und nach PIP zurück
OUT:
     120H      IN  1DH    ; liefert Statuswort
     122H      ANI 1      ; testet ob serieller Baustein bereit
                          ; ist zum senden
     124H      JZ  120H   ; Schleife bis senden möglich
     127H      MOV A,C    ; Auszugebendes Zeichen aus C in A
     128H      OUT 1DH    ; Zeichen über Port  1DH ausgeben
     12AH      RET        ; Sprung aus Sendeunterroutine  zu PIP
```


Prozessor Z80:

```
IN:
     103H      JR  110H      ; Sprung zur Eingabeunterroutine
     106H      JR  120H      ; Sprung zur Ausgabeunterroutine
     110H      IN  A,(1DH)   ; Port 1DH liefert das Statuswort
     112H      AND 2         ; Testen ob Zeichen eingetroffen
     114H      JR  Z,110H    ; Schleife bis Zeichen da ist
     117H      IN  A,(1CH)   ; Port 1CH  liefert das Zeichen von
                             ; der seriellen Schnittstelle
     119H      AND 7FH       ; höchstes Bit ausblenden
     11BH      LD  (109H),A  ; Empfangenes Zeichen in 109H ablegen
     11EH      RET           ; und nach PIP zurück
OUT:
     120H      IN  A,(1DH)   ; liefert Statuswort
     122H      AND 1         ; testet ob serieller Baustein bereit
                             ; ist zum senden
     124H      JR  Z,120H    ; Schleife bis senden möglich
     127H      LD  A,C       ; Auszugebendes Zeichen aus C in A
     128H      OUT (1DH),A   ; Zeichen über Port 1DH ausgeben
     12AH      RET           ; Sprung aus Sendeunterroutine wieder
                             ; zu PIP
```

Mit dieser kleinen Anleitung müßte jeder sein PIP für CP/M2.2 patchen können um die seriellen Schnittstellen anzusprechen. Etwas Geduld braucht man hierzu, vor allem da die Konfiguration der seriellen Schnittstelle, bzw. der Stecker doch unterschiedlich sein können.
Unterschieden wird hier zwischen DTE (Data Terminal Equipment) oder deutsch DEE (Daten-Endeinrichtung/en) also Rechner; und DCE (Data Communication Equipment) beziehungsweise DÜE (Daten-Übertragungseinrichtung/en) also Modem.

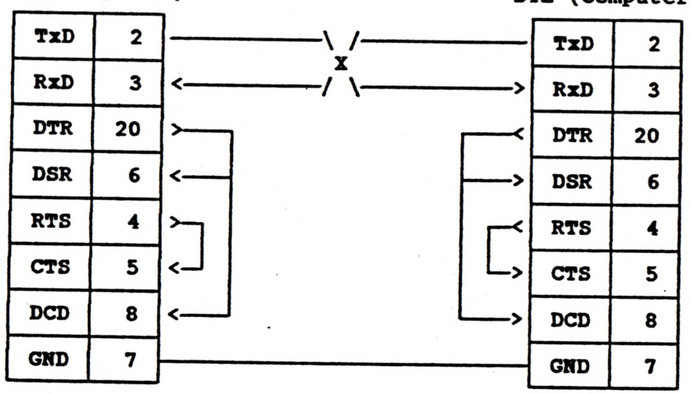
Modem und Computer passen direkt zusammen, da hier Sende- auf Empfangsleitung geschaltet ist. Manchmal ist aber auch der Computer (am Tatung kann dies gejumpert werden) als DCE konfiguriert.
Bei zwei Rechnern, welche als DTE beschaltet sind sollte das nachfolgende Nullmodem für eine funktionierende Verbindung der beiden Rechner sorgen.

In der Regel sind als RS232 Stecker 25-polige D-SUB Typen eingebaut, oder oft bei IBM und kompatiblen platzsparende 9polige D-SUB .
Da ein XON/XOFF - Protokoll genutzt wird reicht ein NULL-Modemkabel aus und braucht nur aus drei Adern zu bestehen, eventuell als vierte Ader eine Abschirmung. Die PINS 4,5,6,8,20 werden am Stecker miteinander verbunden.
Falls jemand Probleme mit der seriellen Übertragung bei seinem Rechner hat stehe ich gerne mit RAT & TAT beiseite. Für folgende Rechner sind bei mir funktionfähige Terminalprogramme zu erhalten: TATUNG TPC2000 - EPSON PX-8 - GENIE 3s - Morrow MD3 - KAYPRO II und Generic CP/M+ Versionen der gängigen Terminalprogramme.

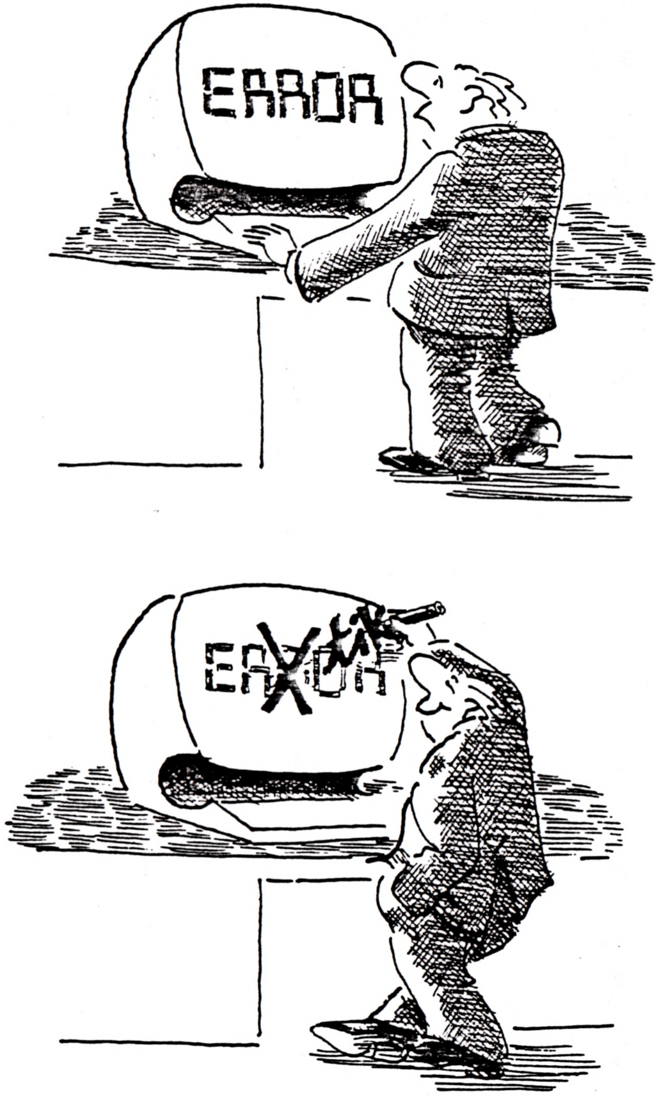
/ Meine E-Mail Adresse: fritz chwolka@solaris.hamm.sub.org /

oder Postadresse laut INFO.

Belegung des Nullmodemkabels bei 25 D-Sub.



Anmerkung: Dieser Artikel wurde mit Wordstar 4 geschrieben und auf einem Panasonic KX-P1124 mit Einzelblatteinzug der Firma Orbitronic ausgedruckt.



© Dietmar Große

..Hardware

Das Club 80 Netzwerk

oder: Poor Man's LAN

von: Rainer Schmitz, Göppingen
Stand: 11. August 1991

Warum?

Jeder kennt wohl die Situation, wenn man auf einen CP/M-Rechner trifft, dessen Diskettenformat man (beziehungsweise die eigene Kiste!) nicht verarbeiten kann und es steht ein Datentransfer an. Sofern beide Rechner über eine serielle Schnittstelle und passende Modemprogramme (LYNC, ZMP, QTERM, MEX, KERMIT oder ähnliche) verfügen, kann man den Datentransfer zwar langsam, aber sicher über ein Nullmodem-Kabel durchführen. Schwierig wird die Sache aber, wenn mehrere Rechner miteinander in Verbindung treten sollen, wie es bei umfangreichen "Gebietsrechenzentren" à la Fritz Chwolka oder Helmut Jungkunz oder bei Clubtreffen (auch ausgefallenen...) vorkommen kann. Man steht dann vor der Alternative, entweder bei den Hobbykollegen mit dem Formatmanager (hallo Z280-Besitzer!) Schlange zu stehen, um die Datenpumpe anlaufen zu lassen oder mit einem möglichst langen Nullmodemkabel und einigen Adaptern und "Geschlechtswechslern" zu versuchen, die RS-232-"Norm" zu umgehen und die Kommunikation zu beginnen. Dabei stört mich, daß man den "Stolperdraht" regelmäßig umstecken muß, um mit einem anderen Comp zu kommunizieren. Kurz, beide Lösungen können nicht als optimal betrachtet werden.

Die Super-Luxus-Lösung des Problems wäre natürlich ein LAN (Local Area Network) in PC-Manier, dem steht allerdings entgegen, daß kaum jemand über ein paar überzählige Ethernet-Karten und ein passendes CP/M-NetBIOS verfügt, außerdem ist die zum Betrieb nötige Kabelage nebst Steckverbindern nicht gerade billig. Zusätzlich stellt sich die Frage, ob ein LAN nicht unnötiger "Overkill" für die oben angesprochene Aufgabenstellung ist.
Der Wunschzettel, den ich daraufhin im Geiste aufstellte, sah folgendermaßen aus:

- kein Umbau am Rechner selbst erforderlich
- keine besondere Hard- und Software nötig, speziell kein "Rumfummeln" am Betriebssystem, möglichst Verwendung vorhandener Modemprogramme
- niedriger Materialaufwand, damit niedriger Geldaufwand
- einfach nachzubauen, auch für Nichtelektroniker
- störungssicher und robust

..Hardware

Wie?

Einen brauchbaren Lösungsansatz fand ich im Leitartikel von Herwig Feichtinger in der mc Heft 7/1986 : dort beschrieb er ein einfaches Netzwerk, das in der mc-Redaktion eingesetzt wurde, um Texte zwischen den Redakteuren auszutauschen, da diese es leid waren, dauernd Disketten hin und her zu tragen. Es basierte darauf, daß die seriellen Schnittstellen in einer Art "verdrahtetem Oder" miteinander verbunden waren. Der Materialaufwand war extrem gering (drei Widerstände, eine Diode), als Verbindungskabel war Koaxkabel vorgeschlagen. Ich habe diese Schaltung aufgebaut und ausprobiert und dabei festgestellt, daß die "handelsüblichen" Modemprogramme damit bei Datentransfer nicht liefen: die Ursache war schnell gefunden. Wenn man sich die Schaltung (Bild 1) anschaut, wird man feststellen, daß der Sender sein eigenes Echo mithört! Das Modemprogramm erwartet aber bei einem Datentransfer nicht irgendetwas, sondern ein entsprechendes Quittungszeichen, mit dem das empfangende Programm den erfolgreichen Empfang eines Datenblocks kundtut (oder auch nicht). Beim Terminalbetrieb stört das Echo nicht weiter, da man dann gleich am Schirm sieht, was man eingetippt hat, sofern man nicht im Terminalmodus "Local Echo" eingestellt hat, denn dann hat man den "Jägermeister-Effekt": man sieht alles doppelt!

Die Schaltung funktioniert so: der Widerstand R3 in der Masseleitung soll ganz einfach Ausgleichsströme zwischen den Rechnern (unterschiedliche Massepotentiale) auf ein ungefährliches Maß begrenzen. Durch den Widerstand R2 in der RXD-Leitung wird der Empfängereingang hochohmiger gemacht, so daß mehr Eingänge parallelgeschaltet werden können. Der Senderausgang TXD zieht bei Ruhepegel den Anschluß Netzwerk+ über den Widerstand R1 unter die Empfängerschaltschwelle, bei aktivem Pegel wird R1 durch D1 überbrückt. Die Abhilfe des Problems beim Datentransfer erforderte kurzes Nachdenken und noch ein paar Bauteile der billigen Natur. In Bild 2 ist zu erkennen, wie's gemacht wird: der Transistor T1 schließt ganz einfach den Empfängereingang kurz, wenn der zugehörige Senderausgang aktiv wird. Angesteuert wird er mit R4 von TXD aus. D2 verhindert, daß der Transistor durch die negative Ausgangsspannung von TXD bei Ruhepegel zerstört wird.

Bei geschicktem Aufbau ist es möglich, die gesamte Schaltung in eine Steckerhaube für einen Submin-D-Stecker (so die offizielle Bezeichnung) einzubauen. Damit der Kontakt zum restlichen Netzwerk hergestellt werden kann, sollte jeder Anschluß mit zwei parallelgeschalteten Cinch-Buchsen (auch als RCA-Buchsen bekannt) versehen werden. Zum Anschluß gehört natürlich auch das passende Verbindungskabel: ich empfehle aus eigener Erfahrung ganz normales NF-

Kabel, und zwar die einadrige abgeschirmte Sorte. Bei den niedrigen Kabelpreisen halte ich eine Länge von drei Metern pro Station für angemessen, das reicht in den meisten Fällen (sprich: Clubtreffen). Der Innenleiter (Seele) kommt an den Netzwerk + Anschluß, die Abschirmung an den Netzwerk - Anschluß, das entspricht jeweils Stift und Massehülse beim Cinch-Stecher. Den Stecker am freien Kabelende steckt man dann in die freie Netzwerk-Buchse des nächsten Rechners.

Versuche ergaben noch bei ca. 30 m Kabellänge einwandfreie übertragungen bei 9600 Bd, was ja wohl ausreicht. Ich sehe aber keinen Grund, warum es bei den höheren Baudraten nicht auch gehen soll! Das Oszilloskop zeigte jedenfalls bei 19200 Bd und ca. 250 m Kabel (!) am freien Ende noch schöne saubere Flanken. Ach ja, noch was: wenn der Rechner auf Hardware-Handshake Wert legt, sollte man ihm den Gefallen tun und RTS mit CTS verbinden (Pins 4 und 5), dasselbe gilt für die Pins 6,8 und 20 (das Dreieck rund um Masse, Pin 7). Diese Pinbelegung gilt aber nur für 25-polige Verbinder nach Norm! Bei meiner Kiste (Joyce) kann ich aber wählen, ob ich Handshake haben will oder nicht (via SETSIO). Besitzer anderer Rechner sollten lieber ins Handbuch schauen (so vorhanden!). Gerade bei Normbelegungen sind die Hersteller ja besonders einfallsreich...

Eine weitere Frage wäre noch in einem Feldversuch zu klären: nämlich die, wieviele Stationen das Netz verträgt. Nach meinen Berechnungen sollten bis zu sechs Stationen möglich sein, das hängt aber von verschiedenen Randbedingungen ab. Der in den meisten Schnittstellenschaltungen verwendete 1488 (75488) kann durchschnittlich 10 mA Ausgangsstrom liefern. Da jeder Eingang eines Empfängers 1489 ca. 1.2 mA Eingangsstrom oberhalb der Schaltschwelle benötigt, kann man sich schnell ausrechnen, wann Schluß ist. In der Realität gibt es aber Streuungen der Bauteilewerte, die diese Grenze nach oben oder unten verschieben können, und das probiert man wohl am besten aus. Bei der Originalschaltung aus der mc war bei drei Stationen am Netz Schluß, wie man durch Berechnung der Ströme im Netzwerk leicht nachweisen kann! Daher habe ich auch einige Werte gegenüber der Ursprungsschaltung verändert, ohne daß bei den Versuchen Probleme auftraten.

Womit?

Wie weiter oben schon angedeutet wurde, war angestrebt, das Netzwerk mit Standardsoftware zu betreiben, um den bei Software-Neuentwicklungen üblichen Zyklus Editieren - Assemblieren - Systemabsturz zu umgehen (ich gebe zu, mir fehlte dazu die Motivation und Erfahrung). Bei den ersten Versuchen setzte ich (oh Schande!!) mangels eines weiteren CP/M-Rechners einen kompandebilen Computer (mit MeSsy-DOS) mit Procomm ein, dazu meinen "Taschenknecht"

Portfolio. Alle diese Rechner verstanden sich unter XMODEM prächtig! Auf CP/M-Seite setzte ich ZMP ein, das ich inzwischen auch auf meinen neuen Rechner, einen Wavemate Super Bullet, angepaßt habe (falls jemand den Quelltext für den Overlay braucht, bei mir melden!).

Einen ersten Härtetest bestand das Netzwerk bein Treffen des Club 80 am 9. bis 12. Mai in Leipheim (nicht Leibheim, Gerald!). Wie bereits im letzten Clubinfo vermerkt, funktionierte das Ganze mit ca. 20 m Kabel recht gut. Es wurde hauptsächlich ZMP und LYNC zur Kommunikation eingesetzt, wobei man mit LYNC sogar den anderen Rechner "fernsteuern" kann, also Laufwerke wechseln, Directories anzeigen und dergleichen mehr (nur Disketten wechseln muß man von Hand!). Auf diese Weise gelangten einige Programmpakete aus der PD von einem zum anderen Rechner, und das über nur zwei Drähte! Mit 9600 Bd geht der Filetransfer sogar recht flott, so daß eigentlich bei einem der nächsten Treffen ein Rechner mit einer mit dem Vereinsarchiv gefüllten Platte und LYNC in der Ecke stehen und als "Fileserver" dienen könnte. Denkbar wäre auch ein Mailboxprogramm, mit dem man sich unterhalten könnte. Dabei muß man sich aber bewußt sein, daß dann nur Kommunikation mit dem Fileserver oder der Mailbox möglich ist, nicht aber mit anderen Rechnern. Das Netzwerk funktioniert ähnlich wie CB-Funk, wenn einer spricht, ist halt der Kanal belegt und die anderen müssen warten. Wenn zwei gleichzeitig sprechen, kommt Müll raus, aber das ist der Preis der Einfachheit. Besonders lustige Effekte kann man beobachten, wenn man mehr als zwei Stationen unter LYNC betreibt: da das Programm versucht, bei jeder Aktion eine Verbindung herzustellen, und jedesmal zwei antworten, sind bald alle drei damit beschäftigt, sich gegenseitig "die Hand zu schütteln". Die Konsequenz der Eigenheiten dieses Netzwerks ist daher: Disziplin der Benutzer! Bevor man die Leitung belegt, sollte man im Terminal-Modus nachschauen, ob gerade Ruhe herrscht, und erst dann loslegen. Wenn dann doch zwei gleichzeitig senden, passiert außer einer versauten übertragung nichts weiter, das heißt, ihr müßt in diesem Fall nicht mit kleinen Atompilzen, die eurem Rechner entsteigen, rechnen! (Das gilt aber nur, wenn ihr bei der Hardware keinen Mist gebaut habt, auch 220 V sind von der Netzwerk-Teilnahme ausgeschlossen!)

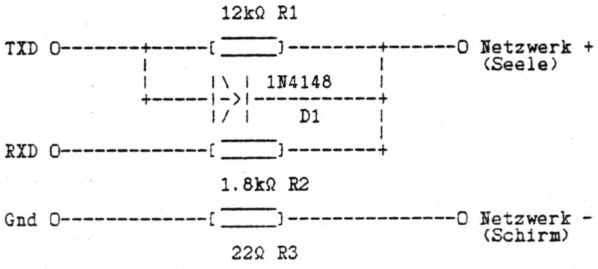
Zusammenfassung (für Diagonalleser)

In diesem Artikel habe ich ein Netzwerk für den Filetransfer "nach Holzfällerart" beschrieben, das für unsere Clubtreffen und den heimischen Gebrauch gedacht war. Seine Einfachheit wird durch ein paar Funktionseinschränkungen erkauft, diese stellten sich aber in der Praxis als tolerabel heraus, besonders bei Berücksichtigung des Aufwands. Hinzu kommt, daß das Netzwerk unabhängig vom Betriebssystem funktioniert, also auch für PC's, Atari's, Amiga's und andere geeignet ist, sofern diese über eine serielle Schnittstelle nach RS 232 (V-24) mit TXD und RXD verfügen. Ob es zu einer Kommunikation kommt, hängt dann nur noch von der (Modem-)Software ab.
Soweit mein Bauvorschlag, seht zu, was ihr draus macht, beim nächsten Clubtreffen will ich jede Menge CP/M-Rechner mit Netzwerkanschluß sehen!

Ausblick (auch das noch!)

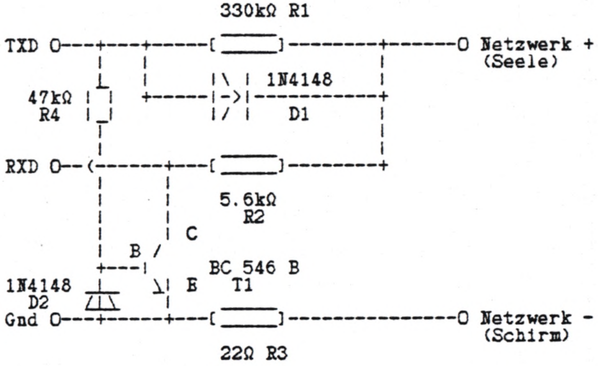
Seit ich mit diesem Artikel angefangen habe, erhielt ich einige faszinierende Informationen aus der Welt des Amateurfunks: dort ist im UKW-Band ein drahtloses Netzwerk namens "Packet Radio" entstanden. Es handelt sich dabei um ein modifiziertes X.25-Protokoll à la ISBN. Dadurch ist es möglich, auf einem Funkkanal mehrere Verbindungen gleichzeitig zwischen verschiedenen Stationen aufzubauen. Speziell die Hardware dürfte für uns interessant sein: der typische TNC (das ist das Modem zum Anschluß eines Computers an ein Funkgerät) ist mit einer Z80 CPU und SIO (wegen dem synchronen X.25-Protokoll) aufgebaut. Ein Bekannter besorgt mir Schaltpläne und ein Eprom mit der Betriebssoftware, so daß man sich die Sache mal anschauen kann. ZMP enthält ja ein Packet Radio Overlay namens ZMYAPP, das erleichtert die Sache gewaltig. Soweit also die neuesten Infos zum Thema Ausbau des Netzwerks.

Bild 1



Netzwerkankopplung nach mc 7/1986

Bild 2



Erweiterte Netzwerkankopplung

Anmerkungen zum Club-Terminal und erste Erfahrungen

Nachdem Helmut noch jede Menge andere Projekte am laufen und sein Terminal noch nicht ganz fertiggestellt hat, sind Stefan Nitschke und ich bis jetzt leider die einzigen, die in den vollen Genuß dieses Teils kommen (bei dieser Gelegenheit möchte ich Andreas Magnus nochmal herzlich danken, der das Ding auf dem "ausgefallenen" Clubtreffen in stundenlanger Fleißarbeit debuggt hat). Wie schon in einem der letzten Infos passend bemerkt, ist der Aufbau nicht ganz einfach, besser gesagt muß man aufpassen wie ein Schießhund, aber wenn es mal läuft, ist es allererste Sahne und braucht den Vergleich mit kommerziellen Terminals im keinster Weise zu scheuen. Im Gegenteil, ich würde es mit keinem anderen Terminal tauschen, das ich bis jetzt gesehen habe.

Einige der Punkte, die es gegenüber anderen Terminals hervorheben:

Punkt 1: Geschwindigkeit

Helmut hat als Taktfrequenz zwar nur 4MHz vorgesehen, aber bei mir läuft es ohne Probleme mit 10MHz und damit bringt es eine maximale Übertragungsrate von knapp 12000 Zeichen (nicht Bit !) pro Sekunde. Mit einem Takt von 12MHz bin ich zwar schon auf 15000 Zeichen gekommen, aber das wollte ich dem Z80 auf die Dauer nicht zumuten, da ihm dabei etwas sehr warm ums Herz geworden ist.
Zum Vergleich: Bei einer seriellen Übertragung bräuchte man dafür rund 120000 Baud (1 Startbit + 8 Datenbits + 1 Stopbit = 10 Bit pro Zeichen) und das dürfte man kaum irgendwo finden, 38400 Baud sind meistens das höchste der Gefühle, wenn nicht gar nur 19200 oder 9600. Wie man im Info nachlesen kann, wird das Hercules-Terminal ja über Ports und somit mit 8 Bit auf einmal betan, daher der Geschwindigkeitsunterschied. Wen es vielleicht wundert, daß die Datenrate nicht noch wesentlich höher ist, dem sei gesagt, daß da die Hercules-Karte ganz gewaltig bremst, weil sie den Z80 Zwecks störungsfreier Ausgabe nur während des Zeilenrücklaufs, also wenn der Elektronenstrahl nicht schreibt, auf den Bildschirmspeicher zugreifen läßt (Besitzer alter TRASH-80 und Video Genies wissen, was ich meine). Daneben braucht das BIOS natürlich auch seine Zeit, die Zeichen zu verarbeiten und auszugeben.

Punkt 2: Bildschirm und Tastatur

Da das Teil mit einer handelsüblichen Feld-, Wald- und Wiesen-Hercules-Karte arbeitet, kann man jeden beliebigen TTL-Monitor anschließen, egal ob grün, orange oder weiß und genauso geht es mit der Tastatur. Die Komponenten sind einzeln austauschbar und man kann das System so zusammenstellen, wie man es gerne hätte. Im Idealfall schmeißt man den PC weg und führt Keyboard und Monitor dem Terminal zu. Austauschtastaturen für Terminals kosten wegen der kleineren Stückzahlen mindestens das Doppelte einer guten PC-

Tastatur (300 Mark aufwärts) und der Fernseher liegt bestimmt auch weit jenseits eines Hercules-Monitors, wenn es ihn überhaupt einzeln bzw. als Reparaturteil gibt. Wenn die Hercules selber mal den Geist aufgeben sollte, kommt man mit rund 30 Mark direkt billig weg. Die Freude am Club-Terminal dürfte damit also für lange Zeit gesichert sein.
Die Bildschirmausgabe selber ist über jeden Zweifel erhaben, gestochen scharf und flimmerfrei und die Zeichen haben etwas mehr "Körper" und sind dadurch angenehmer zu lesen als die übliche "Computer-Schrift", die wie bei einem alten 9-Nadel-Drucker aussieht . Insgesamt stehen 10 Attribute (normal, hell, normal unterstrichen, hell unterstrichen, invers und das alles nochmal blinkend) zur Verfügung, was man auch nur bei den Spitzenmodellen unter den normalen Terminals findet. Zur Textdarstellung kann man aus dem vollen 8-Bit-IBM Zeichensatz mit Semigrafik und allen Sonderzeichen schöpfen, wobei nationale Umcodierungen natürlich unterstützt werden.

Punkt 3: Software

Eine faszinierende Funktion des Hercules-Terminals, die ich so oder ähnlich noch nirgends gesehen habe, ist, daß man ihm vom Host-Rechner ein neues Betriebsprogramm schicken kann. Das Terminal ist ja praktisch ein eigenständiger Einplatinencomputer, den man mehr oder weniger beliebig programmieren kann. Die Software steht vollständig als Sourcecode zur Verfügung und wenn einem etwas nicht paßt, ändert man es und schickt es anschließend zum Terminal. So braucht man nicht den halben Rechner zu demontieren und keine EPROMs zu brennen, um kleine Updates auszuprobieren oder z.B. die Belegung der Funktionstasten zu ändern.

Natürlich gibt es auch gute Terminalkarten, wie etwa die GRIP von Conitec, zu kaufen, aber die sind sauteuer und die Software muß man erstmal disassemblieren, bevor man was daran ändern kann (hoffentlich ist es kein Microcontroller...). Bei den "richtigen" Terminals ist man sowieso auf das angewiesen, was das Ding von Haus aus kann. Das Club-Terminal kann man für rund 100 Mark aufbauen und damit dürfte es im Preis-/Leistungsverhältnis sehr einsam und verlassen weit an der Spitze stehen. Kostenlos bekommt man auch noch eine hochauflösende Grafik mit 720 x 348 Punkten und je nach Hercules-Karte eine parallele Druckerschnittstelle dazu. Wer für etwas weniger Geld und dafür etwas mehr Arbeit eine professionelle Mensch-Maschine-Schnittstelle sucht, für den ist das Club-Terminal schlichtweg die ideale Lösung.
Was das Terminal am Ende alles kann, ist eigentlich nur durch die Phantasie begrenzt und davon habt ihr bestimmt eine Menge (und traut Euch nur nicht, die Ideen im Info zu veröffentlichen...).
Wenn sich jemand mal so richtig an der Software mit Assembler, Linker und allem Zubehör, austoben will, die hochauflösende Grafik und der Druckertreiber stehen im Moment noch zur freien Verfügung.

Alexander Schmid

Hallo Club 80´ er!

Noch einmal PC-Hardware, auch wenn diese in der Clubzeitschrift nicht gern gesehen wird.
Wie ich in einigen Fachgeschäften festgestellt habe, legen einige Verkäufer bei der IRQ-Jumperung die Ohren an, sofern mehr als 2 serielle Schnittstellen im System enthalten sind. Es gibt unter Umständen einige I/O Karten, wo die Interrupts der parallelen Schnittstelle benutzt werden können, bezw. durch auftrennen einiger Leiterbahnen die IRQ´s 5 und 7 an die serielle Schnittstelle herangeführt werden können. Eine Doppelbelegung von Interrupts ist nicht möglich (außer es wird ein Umschalter im Slotblech eingebaut, was sehr unpraktisch ist). *Doch es geht auch einfacher.* Auf einem AT-Rechner sind auf dem AT-Bus noch einige Interrupts frei. Diese können mit einem Hardware-Platinenschnipsel (einseitige Lochrasterplatine) und einigen Zentimetern Litze an die I/O-Karte herangeführt werden. Natürlich müssen auf die COM-Schnittstelle dann Programme geladen werden, wo der IRQ und die I/O Adresse einstellbar sind (z.B Telix(DFÜ)).

Weiter habe ich festgestellt, das es einige Drucker oder Hardwarezubehörgeräte gibt, die auf der paralelien Schnittstelle einen IRQ 5 oder 7 verlangen. Auch kommt es vor, das einige Handscanner mit IRQ 5 gejumpert werden wollen. Aus diesem Grund, sollte man es sich überlegen, ob der IRQ 5 und 7 an der seriellen Schnittstellen benutzt werden soll.

Moderne Betriebssysteme wie Ms-Dos 5.0 oder DR-Dos 5.0 können 4 serielle Schnittstellen verwalten und die Adressen in das Betriebssystem einbinden. Dieser Vorgang ist nach dem Booten des Rechners erkennbar, denn die I/O Adressen werden auf dem Monitor eingeblendet.

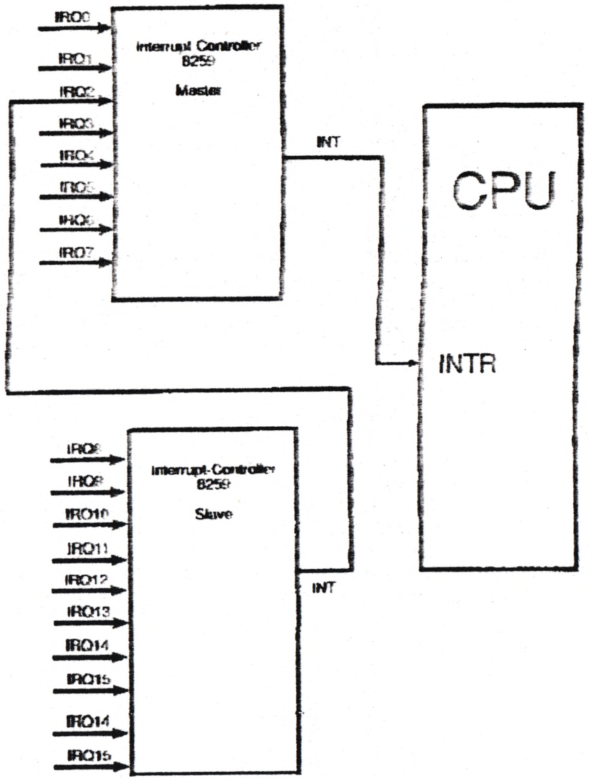
I/O Adressen	Schnittstellen	Interrupt	
3F8	COM 1	IRQ 4	
2F8	COM 2	IRQ 3	
2E8	COM 3	IRQ 7	z.B. reservierter Druckerinterruppt LPT 1
2E0	COM 4	IRQ 5	" " " LPT 2

oder

2E8	COM 3	IRQ 10	AT-Buspin
2E0	COM 4	IRQ 11	" "

Interruptbehandlung —> Interrupt-Controller 8259A

Im XT-kompatiblen PC waren 8 Interrupts möglich. Da davon fast alle durch interne Funktionen belegt waren, erwiterte man die Anzahl beim AT auf 16. Für die Bearbeitung dieser 16 Hardware-Interrupts (IRQ 0 bis IRQ 15) wurden im AT zwei Interrupt-Controller des Typs 8259A KASKIERT. Dies war notwendig, da die Prozessoren nur einen Interrupteingang besaßen. Um bei den ersten acht Interruptkanälen kompatibel zum XT zu bleiben, wird für den durch die Kaskadierung belegten Interrupt 2 der Interrupt 9 verwendet. Das ROM-BIOS parametriert diesen Interrupt während der Initialisierungsphase des Rechners auf den Software-Interrupt-Vektor 0A Hex um, den Vektor für Interrupt 2. Mit dem INTR-Signal stellt der Interrupt-Controller 8259A (Master) eine Unterbrechungsanforderung an den Prozessor 80286 oder 80386.



Das INTR-Signal kann von einem der 15 Einzelinterrupts ausgelöst werden. Natürlich kann es auch vorkommen, daß mehrere Interrupts gleichzeitig ausgelöst werden.

PRIORITÄT DER INTERRUPTS

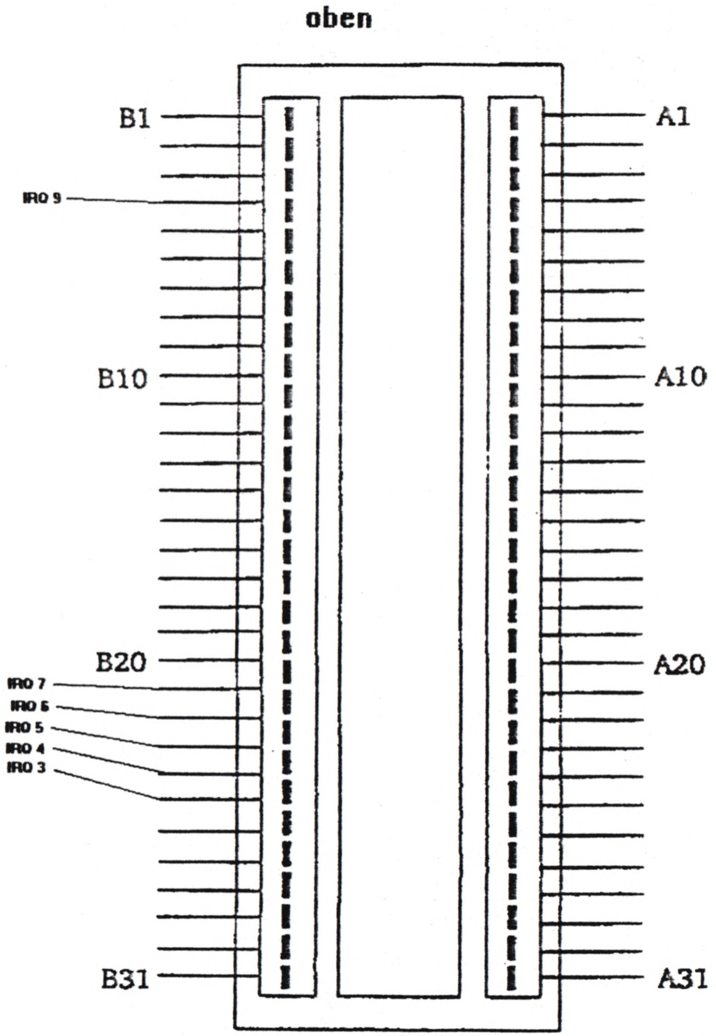
Die Priorität der Interrupts verläuft in umgekehrter Reihenfolge wie deren Nummerirung. Der Interrupt IRQ 0 hat die höchste, der Interrupt IRQ 7 die niedrigste Priorität. Für die Auflösung von IRQ2 hat Interrupt IRQ 8 die höchste und IRQ15 die niedrigste Priorität. Dadurch haben die Interrupts IRQ 8 bis IRQ 15 höhere Priorität als die Interrupts IRQ 3 bis IRQ 7.

Wurde ein Interrupt ausgelöst, so unterbricht der Prozessor seine Arbeit, liest aus der Interrupt-Vektor-Tabelle die Adresse der zugehörigen Interrupt-Service-Routine aus, arbeitet diese ab und kehrt zum eigentlichen Hauptprogramm zurück. Diese Programmier-Technik ist besonders in der Datenerfassung unter Echtzeitbedingungen wichtig.

Hardware-Interrupts beim AT

NMI	Ram-Parität
IRQ 0	Timer Ausgang 0
IRQ 1	Tastatur (Ausgabepuffer voll)
IRQ 2	Interrupt von Controller 2
IRQ 3	2. serielle Schnittstelle
IRQ 4	1. serielle Schnittstelle
IRQ 5	2. parallele Schnittstelle
IRQ 6	Floppy-Disk
IRQ 7	1. parallele Schnittstelle
IRQ 8	Uhr
IRQ 9	Software-Interrupt nach INT 0AH
IRQ 10	frei
IRQ 11	frei
IRQ 12	frei
IRQ 13	Coprozessor
IRQ 14	Festplatte
IRQ 15	frei

XT - Bus

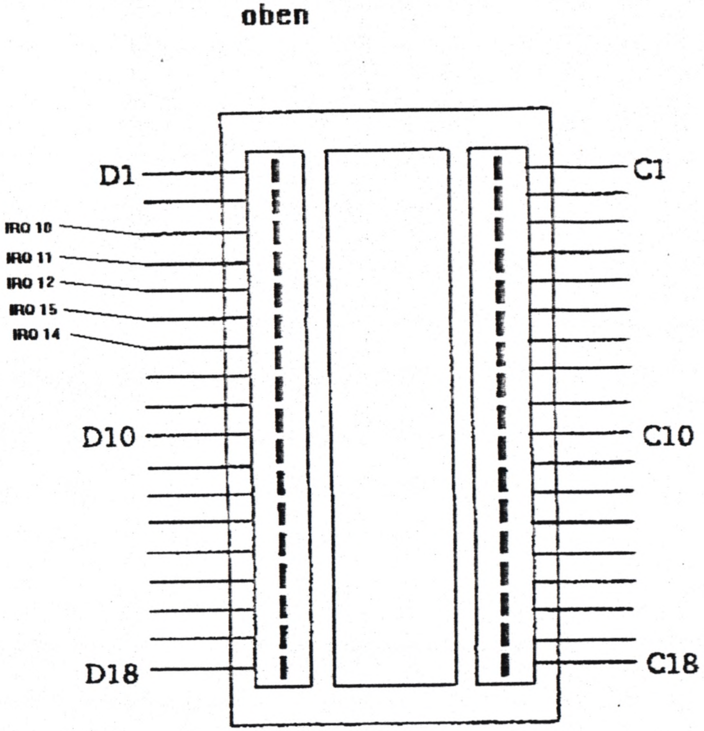


Bauelement Seite

I/O Buspin-Nummerierung

[62 Pin]

AT - Bus



Bauteile-Seite
I/O Buspin-Nummerierung (36 Pin)

Mit freundlichen Grüßen

Harald

Anschluß und Steuerung von Floppylaufwerken an CPC - Computern

Seit seinem Einstand auf dem deutschem Markt überzeugte das Computersystem CPC der Firmen Amstrad/Schneider mit seinen Komplettangebot und seiner Vielfalt an Möglichkeiten.
Besonders Computerfans die mit Anwendungen ihren Rechner nutzen, oder mit CP/M in die Welt der Informatik blicken wollen, sind gut mit diesem System bedient.
Die CPC- Computer lassen sich sehr leicht zu professionellen Geräten aufrüsten, denn es sind reine CP/M- Rechner. Leider wurde der CPC 464 zu einem Homecomputer abgespeckt und das macht sich besonders bemerkbar wenn man ein Laufwerk anschließen will.
So fehlt beim CPC464 ein Floppycontroller. Man muß ein spezielles 3" Laufwerk der Firma Amstrad/Schneider beziehen (DDI-1), das mit einem Controllermodul und einem CP/M-System ausgeliefert wird.
Solche Speziallaufwerke werden auch von verschiedenen Firmen mit CP/M-System angeboten (z.B. Vortex, Dobbertin). Laufwerke dieser Fremdhersteller bieten wesentlich mehr an Leistung, z.B.zweiseitig nutzbare Diskettenformate von je 360 KB. Einfacher haben es da die 664 u. 6128 Besitzer, diese haben ein komplettes System mit eingebauten 3" Laufwerk und einem CP/M Softwarepaket.
An einem Computer gehören jedoch wenigstens zwei Floppys und auf das unübliche 3" Format kann man auch verzichten.
Ich möchte hier den Anschluß von Laufwerken an einem CPC 6128 beschreiben und im allgemeinen gilt folgendes auch für den CPC 664.

Was man braucht:

Diskettenlaufwerk	- 3.5 oder 5.25 Zoll mit 80 Tracks (unbedingt empfohlen) ein - oder doppeltseitig (besser) IBM - Standard / Shugart - Bus
Netzteil	- +12 V und +5 V je 1.5 Ampere (min.)
Kabel	- 34 Pole (Flachbandkabel ist zu empf.)
Platinenstecker	- 34pol. (klemmbar)
Centronics-Stecker	- 36pol.! (klemmbar)
Floppystecker	

Wer sich für die internen Besonderheiten der CPC Typen 664/6128 interessiert, dem möchte ich hier die nötigsten Informationen geben. Leider gibt es auf dem Buchmarkt keine entsprechende Literatur mehr. CPC-Besitzern empfehle ich jedoch dieses Buch :

Das Schneider CPC Systembuch / Günter Woigk
Verlag - Sybex

In den CPC Computern 664 und 6128 befindet sich ein spezieller Baustein zur Verwaltung des Datentransfers und zur Steuerung der angeschlossenen Laufwerke - der FDC 765 (Floppy Disk Controller). Dieser Mikroprozessor ist baugleich von verschiedenen Firmen erhältlich.

<u>Firma / Typ :</u>		
	NEC	uPD 765 A
	ROCKWELL	R 6765
	INTEL	8765

Technische Daten :

Speisespannung	5 V	max. 7 V
Input low	min. -0.5 V,	max. 0.8 V
Input high	min. 2 V,	max. 5.5 V
Output low		max. 0.45 V
Output high	min. 2.4 V,	max. 5 V
Strombedarf	150 mA	
Ein - Phasen - Takt	4 oder 8 Mhz	
Temperatur	min. - 10, max. 70 Grad	

Datenformat :

IBM / 3740 - Single Density
IBM / System 84 - Double Density

(Commodore, Apple, Genie - Tandy Formate *1
können nicht gelesen werden.)

Anschlußmöglichkeiten :

DMA-Betrieb - Hier arbeitet der FDC mit einem DMA-Controller zusammen und entlastet somit die CPU, daher ist ein sehr schneller Betrieb möglich.

Interrupts - Datentransfer mit Interrupt über CPU Pro Datentransfer ein Interrupt. Daten- Befehlsbyte wird vom Prozessor gelesen oder geliefert.

Polling - Diese wird im CPC benutzt. Sie benutzt ebenfalls Interrupts. Der Prozessor muß regelmäßig in den Registern des FDC die nächste Aktion abfragen. Als Datenseparator dient im CPC das IC SMC 9216.

Leistungsumfang :

Verwaltung und Steuerung von 8", 5.25", 3.5" und 3" - Laufwerken
einseitig od. doppelseitig mit 40, 80 oder mehr Spuren
vier Laufwerke anschließbar
Sektorlänge und Laufwerksdaten programmierbar

Einschränkung des FDC im CPC :

Durch eine Gatterschaltung können vom FDC nur zwei Laufwerke verwaltet werden. (US1 - Unit Select 1 ist nicht angeschloßen)
8 Zoll - Floppyanschluß (Head Load) nicht ge - schaltet.
Rom (Software) unterstützt nur einseitige Lauf - werke mit 40 Spuren.
Gate Array (ULA 40010 - CPC 6128) speißt FDC mit 4 Mhz.

Anschlussbelegung :

```
RESET   (1)   -> #1----|_|-----40#       Vss +5 Volt
RD      (0)   -> #|              |#  ->     RW/SEEK
WR      (0)   -> #|              |#  ->     LCT/DIR
CS      (0)   -> #|              |#  ->   FLTR/STEP
A 0           -> #|              |#  ->     HD LOAD
D 0          <-> #|              |# <-    (1) READY
D 1          <-> #|              |# <-     WR PT/DS
D 2          <-> #|      7       |# <-    FLT/TRK 0
D 3          <-> #|      6       |#  ->        PS 0
D 4          <-> #|      5       |#  ->        PS 1
D 5          <-> #|              |#  ->     WR DATA
D 6          <-> #|              |#  ->        US 0
D 7          <-> #|              |#  ->        US 1
DRQ          <-  #|              |#  ->        SIDE
DACK    (0)   -> #|              |#  ->         MFM
TC      (1)   -> #|              |#  ->      (1) WE
INDEX   (1)  <-  #|              |#  ->    VCO SYNC
INT           -> #|              |# <-      RD DATA
CLK           -> #|              |# <-       RD WIN
Vss 0 Volt       #|______________|# <-       WR CLK
```

Beschreibung der Pinbelegung
============================

Vss und Vcc
Stromversorgung - +5 Volt , GND (Vss)

RESET
Im Normalbetrieb auf Null-Potential. Wird durch ein High-Signal in einen definierten Zustand gebracht. (High-aktiv)

RD - WR - CS
Ist CS (Chip Select) auf Null so sind die Steuersignale RD (Read) u. WR (Write) aktiv.

A0 - Adresse Null
Unterscheidet bei einem Schreib- od. Lesezugriff zwischen dem Haupt-Statusregister (0) und dem Datenregister (1).

D0 ---> D7 (Datenbus)
Verbunden mit dem Datenbus der CPU.

INT-errupt
Im CPC nicht benutzt

DRQ u. DACK - DMA Request u. DMA Acknowledge
Im CPC nicht benutzt.

TC - Terminal Count
Unterbricht durch einen Highpegel den Datentransfer. Im CPC mit RESET verbunden - nur beim Einschalten des Computers wird ein Impuls erzeugt.

HDLOAD - Head Load Signal für 8 Zoll-Laufwerke
Im CPC nicht benutzt

US0 - US1 (Unit Select)
Über diese Leitungen können 4 Laufwerke betrieben werden. Im CPC ist nur US0 geschaltet. Die Leitung für das zweite Laufwerk ist nicht an US1 angeschlossen, sondern über einen Inverter ebenfalls an US0.

SIDE - Head Select
Seitenwahl bei Doppelkopflaufwerken. Im CPC angeschlossen, wird jedoch vom AMSDOS-ROM nicht unterstützt.

INDEX
Signal des Indexloches der Diskette.

WE - Write Enadle
Daten auf Floppy schreiben (1).

READY - Laufwerk bereit.

WRDATA - RDDATA (Write Data - Read Data)
Serieller Bitstrom von und zur Floppy. CPC-eigener Datenseparator erzeugt mit einem Taktsignal aufzeichenbare Signale für die Diskette. Umgekehrt werden die Signale beim Lesen der Diskette in für den FDC verwendbare Impulse verwandelt.

MFM - Mode Select
Diese Leitung meldet dem FDC in welchen Modus die Diskette beschrieben werden soll.
MFM -> doppelte Datendichte
MF -> einfache Datendichte
AMSDOS-ROM arbeitet nur mit MFM.

VCO SYNC - VCO Synchronisation
Im CPC nicht benutzt.

PS1 - PS0 (Pre Shift early and late)
Melden dem Datenseparator MFM-Modus.

RDWIND - Read Data Window
Datenseparator kennzeichnet über diesen Pin Lesedaten.

RW/SEEK - Read/Write/Seek
Mit diesem Ausgang schaltet der FDC die folgenden vier Signalleitungen zwischen jeweils zwei verschiedenen Funktionen um. Damit kommt der 765 dem Shugart-Bus sehr entgegen, bei dem man mit diesem Trick vier Adern im Verbindungskabel gespart hat. Ein Null-Pegel an diesem Ausgang signalisiert Schreib-/Lesefunktionen (RW), eine eine Eins Spursuchen (Seek). Die Bezeichnungen der folgenden Leitungen enthalten jeweils links die RW-Bedeutung und rechts die für Seek.

FLT/TRK0 - Fault/Track 0
Fault (1) -> Fehler- Flag überprüfen.
(evtl. vorhandenes Fehler-Flip/Flop am Laufwerk)
Durch Schaltung im CPC wird immer "Kein Fehler" angezeigt.
Track (0) -> Signal / Lichtschranke - Indexloch Stellung des Schreib- Lesekopfes kennt der FDC durch mitzählen.

FLTR/STEP - Fault Reset/Step
Nach Fehler -> Flip/Flop zurücksetzen. Im CPC wird diese Funktion durch ein Gatter mit jedem Null-Pegel an RW/SEEK erzeugt. Step -> eine Spur weiter.

LCT/DIR
Low Current - Singnalamplitude
Direction - In Verbindung mit STEP Richtung angeben.

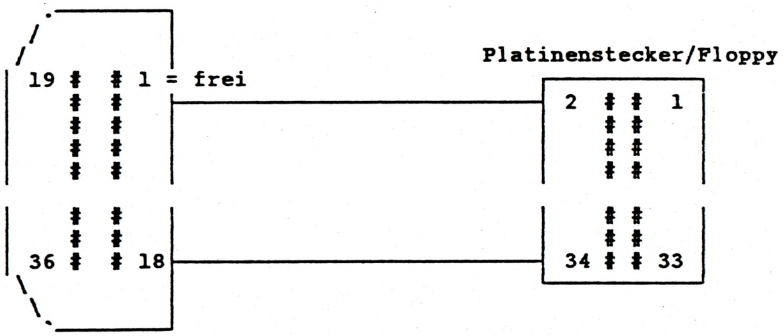
WRPT/DS - Write Protect / Double Sided
Write Protect -> Diskette Schreibgeschützt ?
Double Sided -> Doppeltkopflaufwerk ?

D E R A N S C H L U S S

Die CPC-Computer werden seit 1984 in verschiedenen Ländern hergestellt, daher kann es Abweichungen in der Anschlußbelegung geben. Mir persönlich sind zwar keine Änderungen bekannt, jedoch sollte man immer mit Problemen rechnen.
An meinem CPC 6128 habe ich ein Tandon-Laufwerk und später zwei Laufwerke aus Taiwan (IBM kompat.) ohne Probleme angeschlossen.
Für <u>CPC 6128-Besitzer wichtig - die im Handbuch beschriebene Anschlußbelegung entspricht der des CPC 664 und ist somit nutzlos.</u>
Um die Verwirrung optimal zu erhöhen wurde eine Centronicsbuchse eingebaut an der die Pins 1 und 19 nicht belegt sind.Weiter wurde die Belegung um 180 Grad verdreht so das die Masseanschlüsse auf der oberen Leiste der Buchse sind.
Also der Steckerpin Nr.36 kommt an Pin Nr.1 der Buchse - Steckerpin Nr.20 kommt an Pin Nr.33 der Buchse, - aber vielleicht sollte man darauf keine Garantie geben.

Mit diesem Anschlußkabel sollte es jedoch keine Probleme geben:

(* Für CPC 6128 *)



Belegung der Stromversorgung / Floppystecker

```
 ___________________
|  *   *   *   *    |
 \_________________/
    ^   ^   ^   ^
    /   /   /  + 5 Volt
   /  Masse
+ 12 Volt
```

Soll ein Doppelkopf - Laufwerk beidseitig genutzt werden muß man am Flachbandkabel Pin 32 (Head Select) auftrennen und einen einpoligen Schalter anlöten. Will man die Stellung mit einer Leuchtdiode ablesen können muß der Schalter zweipolig sein.

Anschluß Doppelkopflaufwerk :

```
      * -> Pin 32                 * -> +5 V
      |                           |
+- - -|- - - - - - - - - - - - - -|- - - +
:     *                           *      :  <-- Schalter (Wechsler)
: (1) /--------------------------/(2)    :      mit 2 x UM
:    /                          /        :
: ┌o/     o┐               ┌o/     o┐    :
+-|- - - - |- - - - - - - -|- - - - |- -+
  |        |               |        |
 GND     frei            GRÜN-------ROT
                                |
                                █ -> Widerstand 330 Ohm
                                |
                               GND
```

So - damit kann man jetzt schon gut leben und CP/M "fahren". Ich hoffe das bei euch der Anschluß ohne Probleme funktioniert. Man möchte jedoch die 80 Tracks der Floppy ausnützen, denn AMSDOS (ROM-Speicher) unterstützt ja nur 40 Spuren. Hierzu kann man bei den meisten Computerclubs das B360K Format beziehen (B360K.COM u. Formatierungsprog.). Dieses Programm ist Public-Domain und unter-anderem erhältlich bei der Schneider/Amstrad User Group München, H. Jungkunz - Zacherlstr.14 , 8045 Ismaning.
Nun mit B360K hat man 348 KB freien Platz auf der Diskette, denn 12 KB werden für Systemspuren reserviert, falls man einmal das leidige 3" durch ein zweites 5.25" oder 3.5" Laufwerk ersetzen will. Den Anbau eines zweiten A-Laufwerkes (wahlweise umschaltbar auf das 3" A-Laufwerk) möchte ich jedoch in einen weiteren Beitrag beschreiben.

Viel Glück beim Anschluß

von eurem Clubkameraden

Manfred Hürdler

Anmerkungen:

Anmerkungen:
*1) Tandy- und Genie Diskettenformate sind durchaus mit dem 765 lesbar. Beide Rechner schreiben in MF- oder MFM Verfahren die Daten auf Disk. Das Problem ist vielmehr die Diskettenkapazität, die Diskettengröße (3") und die verschiedenen Diskettenformate. Da der CPC im normalfall nur einseitiges Format lesen kann muß man auf anderen Rechnern ein einseitiges Format benutzen. Ich habe die Diskette mit Manfred's Text auf dem Genie bearbeitet und dann auf einen Rechner mit 286 CPU übertragen. Unter CP/M sind diese Formate untereinander lesbar, sobald die Möglichkeit eines Eingriffes in die Formatbeschreibungstabelle des CP/M-BIOS gegeben ist. Allerdings sind auf Grund anderer Aufzeichnungs-struckturen COMMODORE und APPLE - Formate nicht lesbar. Hier wird mit GCR (Group Code Modulation) aufgezeichnet. Commodore macht die Berechnung der Aufzeichnung mit im Laufwerk eingebautem Prozessor und ohne einen speziell hierfür vorgesehenen Controller. Unter TANDY - TRSDOS oder GENIE G-DOS Betriebssystem wird die Diskette physikalisch genau so wie unter CP/M betan. Allerdings sind die logischen Struckturen der Aufzeichnung anders und dadurch unter CP/M nicht verwertbar. Das Directory liegt in der Mitte der Disk, ist anders aber ähnlich aufgebaut und man kann vom DOS her schon fast jedes (un)mögliche Format über Parameter einstellen.

(F.Chwolka)

bitte veröffentliche untenstehende Anzeige in der nächsten Ausgabe 35 des Clubinfos! Danke Willi

Suche dringend XEBEC-Controller für Genie IIIs mit Eprom 2732 !!
Wer kann mir helfen ?
Wer weiß, wie man eine 20 oder 50 MB-Platte anschließen kann, ohne Schäden anzurichten ?
Tel. 02421/501305
Willi Johnen, 5160 Düren

SUCHE LAUFWERK !

Für mein Tandy Model 4P brauche ich ein neues Laufwerk 5 1/4":

80 Spur/Double Density/Double Sided

Wer kann mir helfen ???????????????

Christof Neumann
Wikingerstr. 2
7910 Gerlenhofen
Tel. 07307/5147

Lieber Club 80-Freunde,

aus beruflichen Gründen bin ich auf MS-DOS umgestiegen und gebe meinen TRS 80, Modell 4, und 1 Drucker Tandy DMP 110 gegen Gebot in gute Hände ab.

Ausstattung: 128 kb RAM, 2 LW 720 kb, 1 LW 180 kb.

Software: Multiplan, Scripsit, Assembler etc.

Ich bitte um Abdruck des Textes im Clubinfo.

Mit freundlichen Grüßen

Richard Frey
Diplom-Ingenieur

Club 80 Börse --- Club 80 Börse

W I C H T i G - dringend gesucht:

S Y L O C O P Y für den GENIE IIIs...

mit Sylocopy kann man vom Programm Genietext ein Clone erzeugen und so die Installationen mit abspeichern. SYLOCOPY wurde angeblich als letzte Version für den Genie IIIs verkauft. Bitte melden bei:

Bildhauerin
Keramik - Lehrkurse

ILSE BERNDT-JOCHUM

Entwürfe figürlicher Keramiken

02202/65254 - Stachelsgut 24 - 5060 Bergisch Gladbach 1

GESUCHT .. Teile für GENIE III...
Ich suche für ein GENIE III alles an Hardware und Beschreibungen.
Speziell Beschreibungen und Schaltplan für den TCS - Disketten-
controller mit 2793-Controllerbaustein da dieser (?) bei meinem
Gerät defekt ist
Des weiteren suche ich alles an Hardware mit dem Z80 Prozessor.
Fritz Chwolka, Saarstraße 34, 5173 Aldenhoven (02461)8920

VERSCHENKE:

1) „CP/M-Spezial": [CHIP-Sonderheft]
Das CP/M-Plus Anwenderbuch
2) ROM-Listing & RAM-Adressen
3) ZILOG Z80-CPU Programming Reference Card

Abholung HIER oder Versand (unfrei!)

K.-J. Mühlenbein
Am Mönchgarten XXVIII (1C)
MMMMMMCMXL Weinheim (1B1C)
Tel. 06201-55052

JOHANN RATH
Oberstraße 116
5000 Köln 90
priv: 02203/16367
gesch: 02238/6097

Johann Rath, Oberstr. 116, 5000 Köln 90

Herrn
Fritz Chwolka
Saarstr. 34

5173 Aldenhoven

Köln, 30.09.91

Sehr geehrter Herr Chwolka,

ich habe erfahren, daß Sie einen Computer-Club 80 leiten.
Ich vermute sehr, daß es sich hier um Freunde der alten Tandy-TRS80-Produkte handelt. Sollte das so sein, wäre ich mit einer Anfrage bei Ihnen und Ihren Freunden richtig.

Ich suche für eine Anwendung, die von Schwerbehinderten für Spiel-und Lernbetrieb benutzt wird, Ersatzhardware-Drucker DMP 200 oder 400 und 420. Mit diesen Typen liege ich fest, da eine Umschulung auf heute aktuelle Hardware praktisch einen Stillstand von 2 Jahren bedeuten würde.

Könnten Sie mir Angebote von Ihren Freunden besorgen?

Für Ihre Vermittlung wäre ich Ihnen sehr dankbar.

Mit freundlichen Grüßen

Johann Rath

Betriebssysteme: Oberflächen

von Gerald Schröder

Sicherlich habt Ihr Euch auch schon gewundert, warum ich in der ganzen BS-Serie kein Wort über Kommandos verloren habe (zum Kopieren oder Löschen von Dateien, zum Formatieren von Disketten, zum Anzeigen von Inhaltsverzeichnissen usw.). Warum vergleiche ich nicht die Kommando-Vielfalt von NEWDOS mit den paar spartanischen Befehlen, die CP/M bietet?

Die Antwort darauf: Für mich als *Programmierer* stellt sich das BS als eine Schnittstelle zwischen meinem Programm und dem Rechner dar. Und in meinem Programm steht nicht etwa „`COPY File1 TO File2`", sondern ungefähr folgendes:

```
OPEN File1
CREATE File2
REPEAT
  READ (File1, x)
  WRITE (File2, x)
UNTIL End of File1
CLOSE File1
CLOSE File2
```

Natürlich sieht die Sache je nach Betriebssystem und verwendeter Programmiersprache etwas anders aus, aber das wesentliche habt Ihr schon erraten: Meine BS-*Kommandos* sind BS-*Aufrufe*.

CP/M-Oberflächen

Soviel also zur Sichtweise. Aber es gibt noch einen Grund: Normalerweise ist die BS-*Oberfläche*, also die Schnittstelle zum (menschlichen) Anwender (im Gegensatz zum Anwendungsprogramm), selber ein (Anwendungs-)*Programm*! Am deutlichsten wird das unter CP/M: Kaum jemand benutzt ernsthaft die mitgelieferte Oberfläche von Digital Research, den CCP (Command Console Processor). Dies Ding wird einfach durch ein normales Programm ersetzt, und zwar je nach Geschmack des Benutzers durch ein anderes.

So benutze ich gerne NSWEEP, ein einfaches, menü-gesteuertes Programm mit 1-Buchstaben-Befehlen, z. B. „M" zum Kopieren mehrerer Files. Alexander hat lange MICROSHELL benutzt, ein Programm, das auch Batch-Dateien abfahren kann und vieles mehr. Rüdiger und Fritz (neben diversen anderen CP/Mlern) schwören dagegen auf ZCPR, von dem Ihr ja auch schon gelesen habt, ein Programm, das eine UNIX-ähnliche Umgebung zur Verfügung stellt. Hartmut kommt immer mit POWER an, einem an NEWDOS-SUPERZAP erinnernden Utility.

Wie Ihr seht: Jeder hat (unter CP/M) eine Oberfläche, die ihm besonders gefällt. Der CCP als „normale" Oberfläche ist eher die Ausnahme. Ähnlich sieht es übrigens bei MS-DOS aus. Nur heißen hier die entsprechenden Oberflächen DOSSHELL, VIEWMAX, NORTON-COMMANDER usw.

Was ist eine BS-Oberfläche?

Was machen nun diese Programme? Sie stellen eine BS-Oberfläche, eine Schnittstelle zum Anwender, zur Verfügung. D. h. der Benutzer gibt seine Kommandos ein (je nach Oberfläche nur „C" oder „COPY 0 1,,,CBF" usw.) und das Programm setzt sie in BS-Aufrufe um (wie oben beim COPY-Kommando gezeigt). Es „interpretiert" also die eingegebenen Kommandos, was zu dem Namen „Kommando-Interpreter" für einige dieser Programme führt.

Natürlich kann das Programm dazwischen/dabei noch viele andere Sachen erledigen, z. B. vor dem Kopieren/Löschen nach den zu kopierenden oder zu löschenden Files fragen (wozu es auch wieder ganz unterschiedliche Möglichkeiten gibt); kopierte Files löschen; anzeigen, wieviel schon kopiert wurde, usw.

Batch-Dateien

Wie schon erwähnt, stellen einige Oberflächen auch eine ausgefeilte *Batch*-Unterstützung zur Verfügung. Batch-Dateien sind die aus NEWDOS bekannten „Jobs" (bzw. „JCL"=Job Control Language), die noch aus den Tagen der Lochkartenstapel-Verarbeitung stammen. Jede Karte war ein Kommando, ein Stapel ein *Job* oder *Batch*.

In modernen Systemen nehmen wir statt Lochkartenstapeln lieber Dateien, wobei jede Zeile dann ein Kommando ist. Aber auch hier gibt es

ganz unterschiedliche Möglichkeiten: Einige Batch-„Sprachen“ bestehen nur aus den normalen Oberflächen-Befehlen, wie z.B. bei NEWDOS, einige kennen außerdem noch Kontroll-Anweisungen, z.B. „IF ... THEN ... ELSE ... “ usw. Es sind also fast Programmiersprachen!

Wozu braucht man Batch-Dateien? Nun, wir können ja kleine Programme damit schreiben, die irgendetwas erledigen, was öfter als einmal zu tun ist. So müssen wir nicht immer alle Befehle von Hand eingeben, aber das wißt Ihr ja selbst. Interessant ist eben nur folgendes:

Unter NEWDOS sind die Kommandos und damit die Oberfläche sowie die verfügbare Batch-Sprache fest in das BS integriert. Es gibt kein eigenständiges Programm, das die eingegebenen Kommandos oder Batch-Dateien in BS-Aufrufe umsetzt. Also können wir auch nicht einfach (mal kurz) eine neue Oberfläche nehmen. Bzw. wir können schon, verschenken aber zumindest eine Menge Platz (der für die gleichen Befehle im BS verbraten wird) und sind durch die mangelnde Unterstützung von NEWDOS doch sehr eingeschränkt.

Grafische Oberflächen

Einen hab' ich noch: Bisher haben wir ja nur von Kommando-Oberflächen gesprochen, aber es gibt noch mehr. Jeder ATARI-ST-Besitzer lacht sich über die Leute nur schlapp, die noch auf der Tastatur rumhacken und beim dritten Anlauf immer noch nicht den File kopiert haben, weil der Befehl „COPI“ zufälligerweise nicht von dem Kommando-Interpreter verstanden wird. Inzwischen hat er mit einem kraftvollen Druck auf den Mausknopf und einer lässigen Handbewegung gleich die ganze Diskette kopiert.

Gegenbeispiele gibt es natürlich auch zuhauf. Man versuche mal, mit der Maus alle Dateien „*.?X?“ von einer Festplatte zu kopieren. Oder eine Batch-Datei mit Mausbewegungen zu erstellen...

Schluß

Und da haben wir sie alle (unvollständig) beisammen: Kommando-Oberflächen, Menü-Oberflächen, grafische Oberflächen, Batch-Dateien, Mischformen davon. Jedem das seine, lautet meine Devise.

Als *Programmierer* brauche ich keine Oberfläche: solange man mir einen Compiler und eine Betriebssystemanbindung gibt, kann ich mir alle Oberflächen selber schreiben. Als *Anwender* muß ich sagen: Dazu bin ich viel zu faul. Ich bin sogar zu faul, „COPY file1 TO file2...“ einzugeben. Deshalb benutze ich mit Vorliebe eine grafische Oberfläche mit Maus. Aber wie gesagt: Jeder suche sich das aus, was ihm gefällt und werde glücklich.

Nur eins kann sich kaum einer aussuchen: Das Betriebsystem, das unter seiner Oberfläche rumwerkelt. Und daran hapert's eben meistens. Nicht umsonst ist „Windows“ auf IBM-PCs nicht nur einfach eine Oberfläche, sondern ein ganzes Betriebssystem. MS-DOS bietet einfach nicht genug Möglichkeiten.

© Reinhold Löffler

Paul-Jürgen Schmitz
Dipl.-Ökonom

Information

Seit geraumer Zeit arbeite ich mit dem integrierten Programm Framework unter MS-DOS (beim Genie gab es leider kein solches System, daß ich mir dort immer gewünscht hatte). Umfang: Textverarbeitung, Datenbank (mit dBASE-Schnittstelle), Tabellenkalkulation, Grafik, DFÜ und eingebaute höhere Programmiersprache (FRED) mit sowohl Makros, als auch komplexe Programme erstellt werden können (Einbinden von Assembler oder C möglich); also ein System, mit dem man komfortabel alle gängigen Aufgabenstellungen lösen kann.
Für Interessenten/-innen gebe ich gerne weitere Auskünfte dazu.
Daneben freue ich mich, auf ein interessante Neuerscheinung hinweisen zu können:

FRAMEWORK III ANWENDEN: Eine Einführung für die Praxis, Hüthig-Verlag, Heidelberg 1991, 381 Seiten, geb., DM 68,-
ISBN 3-7785-1919-0

Neat-AT

Wenn jemand von Euch einen Neat-At besitzt, dann wird er sicherlich festgestellt haben, das man wunderbar im Neat-Setup herumspielen kann. Genau dies habe ich auch gemacht und festgestellt , nach dem der Rechner ohne zu meckern seinen Hardware-Check und Bootvorgang beendet hatte, das dieser in einigen Benchmarktestprogrammen beachtliche Leistungen zeigte. Leider zeigte sich nach einigen Betriebsstunden, das ein merkwürdiger Geruch aus dem Rechner in meine Nase drang und der Verdacht aufkam, das auf dem Motherboard irgendetwas überlastet wurde. In meinem Handbuch wurde diesbezüglich zu der Einstellung der Neat-Register keine Warnungen angegeben. Also ging die Sucherei los. Wie sich später herausstellte wurde der DMA-Controller-Chip total überlastet. Dieser stammt im AT wohl noch aus der XT-Zeit. Im XT wird dieser Baustein mit ca. 4,77 MHz getaktet und im AT mt ca. 4 MHz. Durch den Chip & Techniks Chipsatz kann ich die Taktfrequenz weit nach oben verlagern und dadurch eine totale Überlastung des Bausteins hervorrufen.

Faszit : Benutze nur die Standardeinstellung mit dem beigelegten Quickset-Programm. Sonst könnte es passieren das ein junges Motherboard bald zum *Alten Eisen* gehört.

MFM-Festplatten

Wer hin und wieder in die Anzeigen einiger Computeranbieter schaut, wird feststellen das 42 MB--MFM -Festplatten für ca. 248 DM zu haben sind. Diese können manchmal auch unter einem RLL-Festplattencontroller betrieben werden. Doch wie gesagt nur manchmal. Durch die höhere Aufzeichnungsdichte des Rll-Controllers kann es bei einer MFM-Festplatte Timing-Probleme geben. Diese lassen sich nur beheben, wenn auf der Elektronic-Karte der Festplatte die Filter geändert werden. Dies dürfte oftmals nicht so einfach sein, wegen fehlender Unterlagen. So erging mir das auch. Beim Booten wurde die Festplatte erkannt und angesprochen. Die Lampe ging an und blieb an, denn der Rechner hatte sich aufgehängt. Die Low-Level-Formatierung auf 26 Sektoren klappte hervorragend. Auch das Betriebssystem ließ sich einwandfrei auf die Platte schreiben. Aber mehr war aber auch nicht drin. Nach einigem Suchen habe ich dann doch noch einen brauchbaren Kompromiß gefunden. Die Laufwerkslampe wird auf einem Anschlußbein aus der Platine ausgelötet, dadurch bekommt die Festplattenelektronikkarte keine Rückmeldung.

Ergebnis: Festplatte läßt sich einwandfrei ansprechen.Die Festplattenkontroll-Lampe C: wird über den Festplattenkontroller der über einen seperaten Kontroll-Lampenanschluß verfügt, der auch für Festplatte D: zuständig ist, angezeigt. Dieser Hardwarepatch funktioniert auf meinem Rechner schon seit ca. 15 Monaten. Die RLL formatierte MFM-Platte arbeitet genauso zuverlässig wie meine RLL-Platte unter D:.

Mit freundlichen Grüßen

Harald

Heiße Liebesgrüße aus dem PC

FN 251 29.10.91

Neue Computerprogramme helfen auch einfallslosen Casanovas aus der Patsche

Von unserem Mitarbeiter
Karl Heinz Reger

Der Casanova von gestern kaute lange am Bleistift, bis er eine Formulierung fand, mit der er sich von seiner Geliebten angemessen verabschieden konnte. Der Casanova von heute ruft ein Computer-Programm „Anti-Liebesbrief" auf. Der Text, der anfängt: „Lisa, Häschen, nimm das doch alles nicht so persönlich ..." paßt eigentlich immer, man muß nur den Vornamen ersetzen.

Der Portier von gestern kontrollierte sorgfältig den Werksausweis, wenn er nicht sicher war, ob der Ankommende auch wirklich in die Firma darf. Der Portier von morgen hat zwei Fernseh-Bildschirme vor sich. Zieht ein Ankommender seinen Werksausweis durch den Ausweiskarten-Leser, erscheint auf dem linken dessen Bild und auf dem rechten das im Computer archivierte elektronische Paßfoto. Sekundenschnell kann verglichen werden, ob der Betreffende mit dem Zugangsberechtigten identisch ist.

Nur zwei Beispiele, eines eher scherzhaft, das andere sehr konkret über neue Computer-Programme, die auf der jetzt in München zu Ende gegangenen Computer-Messe „Systems" vorgestellt wurden.

Die großen und kleinen der Branche präsentierten ihre neuesten Geräte und Computer-Programme für jede denkbare Anwendung. Das Preis-Spektrum reicht von 19,80 DM für den erwähnten „Elektronischen Liebesbrief" (Rossipaul-Verlag) bis zu siebenstelligen Beträgen. Den schlichten PC-Privatnutzer wird aber interessieren, daß immer mehr ursprünglich für die Industrie und Forschung entwickelten Programme in etwas abgespeckter Form, aber zu passablen Preisen, auch für den Privatmann erschwinglich werden.

Die Hersteller haben inzwischen gemerkt, daß weder mit Büchern auf Diskette noch durch Computer-Programme für die Küche ein überaus großes Geschäft zu machen ist. „Bücher wollen die Leute mit ins Bett nehmen, das geht mit dem PC nicht. Und die Kühltruhe mit einem Programm zu verwalten, ist zu umständlich. Da schauen sie lieber gleich nach, was noch drinnen ist."

Renner sind dagegen Programme, die entweder die erwähnten reduzierten Kopien ursprünglicher Industrieprogramme darstellen oder die Möglichkeiten des Computers besonders ausnützen, die berechnen und auch bewegte Bilder zeigen können.

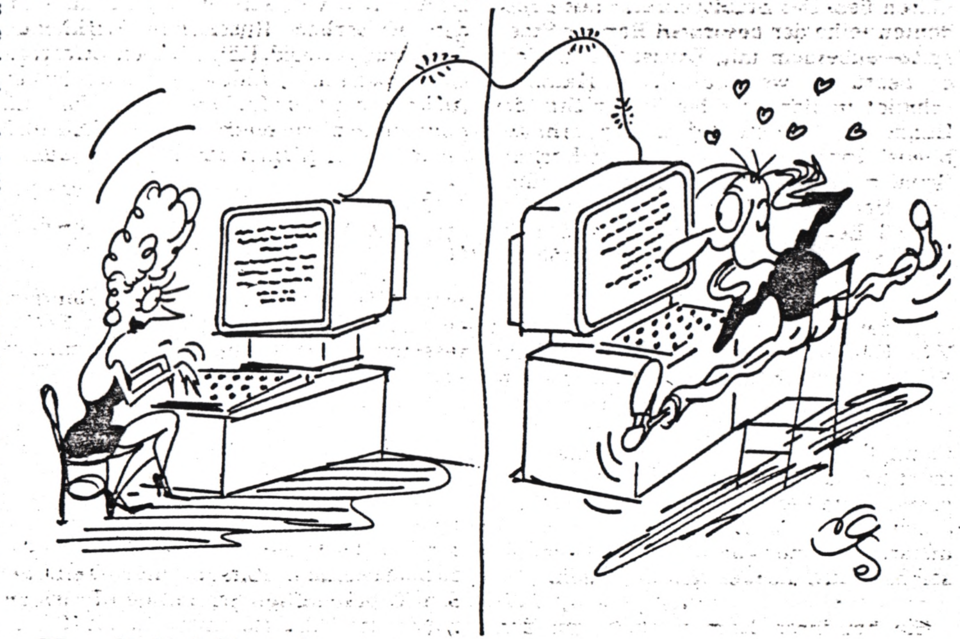
Geoworks (Heureka-Verlag) ist zum Beispiel so ein Programm, das für 399 DM Tausende von Adressen verwaltet (und mit Hilfe eines Modems die jeweiligen Telefonnummern gleich anwählt), über Rechner, Schreib- und Datenfernübertragungsprogramme verfügt und außerdem mit einem „Solitaire"-Spiel ausgestattet ist, das Napoleon erfunden haben soll, als er sich im Exil auf Elba langweilte.

„Markt & Technik Software Partner" (MSPI) in Haar setzt auf die Computer-Ergänzung der häuslichen Bibliothek. „PC-Geos" vom letzten Jahr, das weltweit geographische Daten liefert, war ein Riesen-Erfolg. Es wurde heuer ergänzt durch „PC-Cosmos" (159 DM), das Planeten und Sterne auf den Monitor zaubert, „PC Astrologie", das Horoskope erstellt (aber keine Beurteilungen liefert – DM 199,95) und „PC Mensch", ein anatomisches Lernprogramm (179,95 DM).

Mit dem letzteren können sich auch Medizinstudenten trainieren. Gezeigt werden nicht nur alle Knochen, Muskeln, Organe. Man kann sich farbig Details darstellen lassen – und durch Umschalten wählen, ob die Bezeichnungen in deutsch oder im Medizin-Latein der Ärzte erscheinen sollen. Wenn der Arm gestreckt wird, ist zu sehen, wie der Bizeps wächst. Unter dem Stichwort „Zeugung" bahnen sich die Spermien ihren Weg zur Befruchtung. Es gibt Extra-Programme zur Verwaltung der eigenen medizinischen Daten und jener der Kinder. Man kann sich sogar die persönlichen „fruchtbaren und unfruchtbaren Tage" ausrechnen lassen.

Trotzdem: „Keines der Programme geht so in die Tiefe wie ein ausführliches Lexikon", sagt MSPI-Geschäftsführer Rudolf Wollner, „wenn es aber um spielerische Wissensvermittlung geht, sind sie wegen der bewegten Bilder nicht zu übertreffen." Deshalb glaubt der Software-Produzent, daß neben reinen Computergeschäften zunehmend auch der Buchhandel die von ihm und von konkurrierenden Anbieter-Firmen produzierten Programme anbieten könnte.

Ein anderes Zauberwort, das den Privatverbraucher faszinieren soll, heißt „CD-Rom", was die Abkürzung für „Compact Disc Read Only Memory" darstellt. Commodore führte in Halle 16 so ein Gerät vor, das an das Fernsehgerät im Wohnzimmer angeschlossen wird. Aber auch andere große Namen der Branche, von Apple über Atari, Hewlett & Pakkard, Hitachi, IBM, NEC, Philips, Sanyo, Sony und Toshiba sind mit im Geschäft. Im Unterschied zu Disketten oder Festplatten können diese CD-Rom nur gespeicherte Daten wiedergeben. Man kann sie aber nicht verändern und neu abspeichern. Dafür bieten sie pro Diskette Speicherplatz für 270 000 Schreibmaschinenseiten. Der Inhalt eines 21bändigen Lexikons ist locker auf einer Scheibe unterzubringen.



„Herzliche" Signale

Karikatur: Ryss

Club 80
INFO 35
Okt. 91
Seite

Impressum

1. Vorsitzender
Tel.: 0 24 64/89 20
Fritz Chwolka
Saarstraße 34
5173 Aldenhoven

2. Vorsitzender
Tel.: 0 41 05/26 02
Gerald Schröder
Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator
Tel.: 02 09/87 02 30
Andreas Magnus
Bismarckstraße 29
4650 Gelsenkirchen

Newdos-Diskothekar
Tel.: 07 11/7 35 38 17
Oliver Volz
Waldenburgstraße 73
7000 Stuttgart 80

CP/M-Diskothekar
Tel.: 0 61 31/32 08 60
Rüdiger Sörensen
Thomas-Mann-Straße 3a
6500 Mainz 1

Clubbücherei
Tel.: 0 41 52/7 06 43
Kurt Müller
Sophie-Scholl-Ring 3b
2054 Geesthacht

Redaktion
Tel.: 07 91/4 28 77
Jens Neueder
Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 8870 Günzburg
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Schluß

Hallo Club 80'er,

nachdem nun mein Urlaub und einige wichtige, geschäftliche Verpflichtungen abgewickelt sind, komme ich nun endlich dazu, Euch wieder ein neues Club-INFO zu präsentieren.

Die lange Wartezeit hat natürlich auch etwas für sich, bei termingerechter INFO-Auflage wäre nicht einmal die Hälfte der erschienenen Artikel zu lesen gewesen. Das soll natürlich nicht heißen, daß ich nun einen anderen Erscheinungsrhythmus, als den 2-monatigen, befürworte, aber das Infomachen macht mehr Spaß und lohnt sich auch erst richtig, wenn genügend Material vorhanden ist. (Der allgemeine Aufwand bleibt immer der gleiche).
Ich hoffe, daß sich nun in der "dunkleren" Jahreszeit die Freizeitaktivitäten wieder mehr zum Computer hin verlagern und sich Euer Output in Richtung Redaktion enorm verstärkt. Zum Jahreswechsel möchte ich Euch gern das nächste INFO liefern. Vielleich ist auch ein Sonderheft dabei?

Zum Abschluß dieses INFO's wünsche ich:
... Euch viel Spaß an der neuen Club-INFO
... daß man sich zur Hobbytronik in Stuttgart trifft
... mir viele Artikel für's nächste INFO

Viel Spaß-
bis zum nächsten mal

Jens

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

CLUBINFO

36. Ausgabe

Kontaktadresse: **Club80** / Fritz Chwolka / Saarstraße 34 / 5173 Aldenhoven
Tel.: 0 24 64 / 89 20

Inhaltsverzeichnis

	Autor & Seite
Clubinternes	
Vorwort	1
	Fritz Chwolka
"Formatknackerbande"	2
	Uwe Schobert
Termine	2
	Redaktion
Software	
Noch serieller mit PIP	3 - 4
Prozessor piep einmal ...	5
	Alexander Schmid
BONUM BENE	6 - 15
	Volker Dose
Ein bißchen C, ein bißchen Mathe...	17 - 23
	Kurt Müller
Hardware	
Umrüstung ade !	24
	Uwe Schobert
CPU280 - die Z80-VAX	25 - 26
Einfachste A/D- und D/A-Wandler	27 - 28
	Alexander Schmid

	Autor & Seite
Börse	
Wer hat was – wer will was	29 - 30
Sonstiges	
Der Rechner mit der Handbremse	31 - 33
Schönere Schildchen	33
Hacker's Hirnkrücke	34
	Alexander Schmid
Exceptions, Interrupts und Traps	35 - 38
Vom Single zum Multi	39 - 45
	Gerald Schröder
XT - Extra viel Trouble ???	47 - 48
	Kurt Müller
Keine Glosse, sondern Wirklichkeit	49 - 50
	Klaus-Jürgen Mühlenbein
Public Domain Software	51 - 54
	Fritz Chwolka
Die letzten Seiten	
Impressum	55
Schluß	56
	Redaktion
Mitgliederadressenliste	am INFO-Ende

Hallo Freunde!

Diesmal möchte ich nicht über fehlende Manuscripte schreiben, auch wenn der Jens sich bitter darüber beklagt. Aber warum bringen denn so wenige von EUCH ihre Gedanken zu Papier ? Es muß doch nicht immer ein super-hyper-megabytemäßiges Sebstbauprojekt sein!

Aber ich wollte ja etwas anderes ansprechen. Der Club nennt sich CLUB 80 , weil er sich dem Z80 und damit verbundener Peripherie widmet. Nun hat aber fast jeder von uns neben einem Z80 Rechner auch Kontakt zu z.B. einem MSDOS-kompatiblen Computer, einem MAC oder auch einem Atari. Ich möchte auch nicht verschweigen, daß ich neben einigen CP/M-Rechnern auch einen AT habe.
Meiner Meinung nach können wir uns durchaus in Artikeln oder Fragen auf diese Rechner beziehen, wie es auch im letzten oder vorletzten Info geschehen ist. Durch unsere Arbeit mit dem Z80, ich meine natürlich unser Hobby, haben wir mehr von Rechnern und deren Strucktur kennengelernt als es einem normalen Anwender je gestattet ist. Warum sollen wir denn nicht auch etwas darüber mitteilen.
Hierdurch wird der Grundgedanke des Clubs nicht umgestoßen, sondern vielleicht der Ramen etwas weiter gesteckt.
Auch dürften die Artikel hierzu nicht überhand nehmen, da doch an jedem Kiosk die Computerhefte kiloweise erhältlich sind und Ihr hoffentlich nicht abschreibt.

<u>Hier nun ein Gedanke und Aufruf zur Hauptversammlung</u>

Einer der Gedanken des Clubs ist, mindestens einmal im Jahr eine **Hauptversammlung** abzuhalten. Mein terminlicher Vorschlag wäre der Monat März oder April. Da die Familie Obermann aus beruflichen Gründen diesmal warscheinlich als Veranstalter ausfällt (schade), sollte sich einer von EUCH dazu bereiterklären. Die Aufgabe ist ja klar, durch Rücksprache mit Vorstand den Termin abklären und einen geeigneten Ort mit Unterkunftsmöglichkeit suchen. Auch sollte der Veranstaltungsort so ziemlich in der Mitte unseres Einzugsgebietes liegen. Ich hoffe auf reichliche Anrufe und habe schon ein extralanges Band in den Anrufbeantworter gesteckt.

Fritz Chwolka
Fritz Chwolka

Liebe Clubfreunde !

Nachdem Hans-Günther Hartmann in seinem Bericht '... Formatknackerbande ...' den einzigen erwähnt hat, der einen AlphaTronic P3 betreibt und den er kennt, so möchte ich mich gleich freiwillig melden und ihm für diesen guten Artikel danken.
Dies ist zumindest ein erster Schritt für mich, andere Diskettenformate zu lesen, da mein Betriebssystem das leider gar nicht unterstützt. Ich weiß, daß da noch viel auf mich wartet, doch vielleicht löse ich auch das einmal. (Es ist erstaunlich, wie sehr einen eine solche Schwierigkeit reizt, wo manch anderer Computer das doch im Handumdrehen schon kann.)

Ich hoffe jedenfalls, nicht zuletzt mit Eurer indirekten oder direkten Hilfe, irgendwann einmal unser Club-Diskettenformat (gibts denn ein einheitliches ?) direkt lesen zu können.

Mit freundlichen Grüßen

Uwe Schoberth   Tel.: priv. 07041/7254

Uwe Schoberth

<u>Termine... Termine... Termine... Termine... Termine</u>

Jahreshauptversammlung 92	....	Mai...- 92
Clubtreffen Süd	....	......- 92

DOMOTECHNICA	Köln	18.02.- 21.02.92
CeBIT	Hannover	11.03.- 18.03.92
CAT-Anwenderkongreß	Stuttgart	05.05.- 08.05.92
Orgatec	Köln	22.10.- 27.10.92
Electronica	München	10.11.- 14.11.92

<u>Redaktionsschluß</u> für die Clubinfo's ist jeweils der letzte Tag der folgenden Monate:

März 92
Mai 92
Juli 92

Noch serieller mit PIP

Im Info Nr. 35 hat "uns Fritz" geschrieben, wie man im Notfall auch ohne kompatible Laufwerke, bzw. wenn keine Informationen über das Disketten-format vorliegen, Dateien mit Hilfe von PIP von einem Rechner zum anderen übertragen kann. Dazu noch zwei kleine Anmerkungen/Ergänzungen:

Wenn man PIP als Parameter für die Übertragung zusätzlich noch ein "H" (Hex) mitgibt, schaltet es in den Hex-Modus und überprüft, ob mit der übertragenen Datei (in der Z80-Welt üblicherweise im sog. Intel-Hex-Format) alles stimmt. Diesen Service hat man nur mit "B" (Block) alleine nicht und kann verfälschte Zeichen höchstens daran erkennen, daß der HEX-COM-Wandler (HEXCOM, LOAD o.ä.) hinterher meckert, oder das Programm nicht läuft. Im Hex-Modus stellt PIP anhand der Prüfsumme in jeder Zeile fest, ob möglicherweise ein Fehler aufgetreten ist und meldet dies dann sofort. Wenn man nur beim sendenden Rechner ein E (Echo) als Parameter angibt, wird dieser weiter gebremst, was dem empfangenden Rechner etwas mehr Zeit gibt, die ankommenden Zeichen zu verarbeiten. Normalerweise dürfte das nur bei einem relativ langsamen Rechner auf der einen und einem schnelleren Rechner auf der anderen Seite notwendig sein, aber aus Erfahrung würde ich sagen, daß man das immer so machen sollte. Auch ist es nicht unbedingt gesagt, daß es mit einer kleineren Baudrate immer besser geht, ich habe es schon erlebt, daß es mit 1200 Baud einwandfrei ging, mit 300 Baud aber nur Schrott angekommen ist. Warum ? Keine Ahnung, aber wenn alle Geheimnisse gelöst wären, wäre es ja langweilig. Außerdem sollte man beim sendenden Rechner die Schnittstelle tunlichst auf das XON/XOFF-Protokoll einstellen, sonst funktioniert der Blockmodus nicht. PIP sendet zwar XOFF- und XON-Zeichen, aber die Übertragung wirklich anhalten muß der Schnittstellentreiber bzw. das BIOS. Wenn sich der empfangende Rechner nach der Übertragung nicht mehr zurückmeldet, kann man ihm mit "PIP PUN:=EOF:" bzw. "PIP OUT:=EOF:" noch ein explizites Dateiende zukommen lassen.

Was in Fritz' Artikel vielleicht auch nicht ganz klar rausgekommen ist und viel Zeit und Nerven kosten kann: Der Test mit

PIP PUN:=filename[EB] und PIP filename=RDR:[EB]

(hinter dem Filenamen darf, entgegen der Angabe von Fritz, übrigens KEIN Doppelpunkt kommen, sonst gibt's eine Fehlermeldung !!)

funktioniert nur, wenn das CP/M das IO-Byte unterstützt ! Es gibt Implementationen, die das Byte einfach nicht benutzen und dann wundert man sich, daß nichts geht. Ausprobieren kann man das mit "PIP LST:=CON:", mit dem alle Eingaben von der Tastatur auf den Drucker umgelenkt werden sollten. Da LST, genauso wie EOF weiter oben, ein logisches Gerät ist, muß hier zur Unterscheidung von einem Filenamen nun tatsächlich ein Doppelpunkt dahinter stehen. Wenn man aber ein komfortabes BIOS hat, das u.a. bei einem leeren Empfangspuffer brav selber auf ein neues Zeichen wartet und der Testtext so fehlerfrei angekommen ist, ist man schon fertig und braucht die ganze Patcherei garnicht. Dann kann man mit PUN: bzw. RDR: arbeiten und im Prinzip sofort loslegen und beim CP/M Plus, wo man in beiden Fällen nur über AUX: geht, muß man sowieso nicht fummeln und kann auch gleich anfangen.

Wenn es aber doch sein muß, sei es weil das mit dem IO-Byte nicht funktioniert, oder dauernd Zeichen ankommen, ohne daß der andere Rechner

sendet, und man PIP schließlich so gepatcht hat, daß es die Ports direkt anspricht, muß die Ein- und Ausgabe folgendermaßen aussehen: (PUN: und RDR: gehen nach wie vor über das Betriebssystem, da helfen alle Patches nichts !)

PIP OUT:=filename[EBH] und PIP filename=INP:[BH]

Wenn diese Hürde dann schließlich genommen ist und die Übertragung klappt, sollte man so schnell wie möglich LYNC rüberziehen. Das Programm besteht aus dem eigentlichen COM-File (LYNC.COM), einem Installer (LYNC-INST.COM) und einer Treiber-Bibliothek (IO.DRV). Am besten konfiguriert man das Programm schon vorher, dann braucht man nur ein File zu übertragen. Das File ist auch als Hex-File so klein, daß es zusammen mit PIP in den Arbeitsspeicher paßt, sodaß im Block-Modus erst nach Ende der Übertragung auf die (langsame) Floppy zugegriffen werden muß und es so keine Probleme mit dem Protokoll geben kann.

Warum aber nun ausgerechnet LYNC ? Ganz einfach, es ist verhältnismäßig klein und so ziemlich das einfachste und sicherste Programm für diesen Job, bei dem man eigentlich nichts falsch machen kann. Der Installer arbeitet menügeführt und man muß nur die Ports der seriellen Schnittstelle, die man ja eventuell schon von den Patches in PIP kennt, per Hand eingeben; für viele Rechner (u.a. TRS-80 Model I und IV) gibt es einen fertigen Menüpunkt, den man nur anzuwählen braucht. Danach kann man von einem der beiden Rechnern beliebig Files senden und empfangen, ohne daß am anderen Rechner jemand sitzen muß. Man kann sich z.B. sogar das Directory des anderen Rechners ferngesteuert ansehen und dann mit SEND und FETCH loslegen. Das ist viel bequemer als mit Kermit oder sonstigen Terminalprogrammen, wo man immer noch einen Helfer braucht. Die Übertragung wird durch ein LYNC-eigenes Protokoll gesichert und nichtmal das mutwillige Unterbrechen der Leitung bringt es aus dem Tritt. Nachdem die Verbindung wieder steht, geht es sofort weiter. Außerdem überprüft sich LYNC beim Starten selber, sodaß man dadurch nochmal eine Kontrolle hat, ob die Übertragung des Hex-Files fehlerfrei war.

Alexander Schmid

© Heinz Langer

Prozessor piep einmal...

wenn ein Programm auf einen bestimmten Prozessor angewiesen ist, ist es ganz nett, wenn man auf dem Bildschirm wenigstens eine Meldung ausgibt, daß das Programm z.B. einen Z80 braucht und auf einem 8080 nicht lauffähig ist, statt kommentarlos ins Betriebssystem zurückzuspringen. Für den Z80 und HD64180 gibt es schon einige Lösungen, aber für den Z280 ist mir noch keine über den Weg gelaufen. Wenn man ein neues Feature des Z280 (die Queue) ausnutzt, ist das ohne großen Aufwand mit folgender Routine möglich:

```
     XOR  A
     LD   (flag),A
     LD   A,1
flag EQU  $-1
```

Auf einem "normalen" Z80 wird der Akku in der ersten Zeile gelöscht, in der zweiten Zeile wird diese Null in den Ladebefehl in Zeile drei gepatcht und in Zeile drei bekommt der Akku folglich den Wert 0. Soweit ganz klar und verständlich, aber nicht so beim Z280, wenigstens auf den ersten Blick. Der Z280 hat nämlich die Eigenschaft, daß er immer etwas "vorausschaut", d.h. schon die nächsten Bytes lädt, wenn er den aktuellen Befehl gerade ausführt. Im Beispiel heißt das, daß er, wenn er Zeile zwei ausführt, schon den Code von Zeile drei in die Queue (Warteschlange) geladen hat und daher gar nicht mehr mitbekommt, daß da inzwischen ein anderer Wert hineingeschrieben wurde. Sowas ist natürlich nicht gerade die feine englische Art, aber im Normalfall macht man das ja auch nicht, oder ? Bei einem Z80 enthält der Akku also anschließend eine 0, bei einem Z280 eine 1 (wie das dann weiter verarbeitet wird ist hier nebensächlich). Ganz nebenbei bemerkt ist das auch ein prima Debug-Schutz, der Trick mit der Queue funktioniert nämlich nur, wenn das Programmstück auf einmal durchlaufen wird. Wenn ein Debugger im Einzelschritt-Modus hinter jedem Befehl einen Sprung einfügt wird die Queue jedesmal geleert und der Prozessor verhält sich im obigen Beispiel wie ein gewöhnlicher Z80. Bei einem speziellen Z280-Programm würde man dann immer mit einer Fehlermeldung im DOS landen und sich fürchterbar wundern, warum der Z280 einfach nicht erkannt wird.
Im Prinzip gilt das auch für den 80x86, nur variiert dort die Länge der Queue von Typ zu Typ.

Alexander Schmid

BONUM BENE

Obwohl die Benutzung von G-DOS/NewDOS80 oder auch Calvados eine Kategoriesierung der User in die Sparte "Ewiggestrige" durchaus gerechtfertigt, hat dieses Betriebssystem auch seine Vorz>ge.
Die ganzen schoenen Features die moderne Betriebssysteme aufweisen, fehlen bei uns zwar, aber man kann sie wenigstemns noch selber programmieren. Zumindest einige.

Mich hat beim CP/M und MS-DOS immer die Benutzung von Wildcards, also * und ?, bei Angabe von Filenamen beeindruckt.
Insbesonders beim Kopieren von grossen Datenmengen zeigen sich im Copy-Befehl des NewDOS ernsthafte Schwaechen. Zum einen werden vor dem Kopiervorgang die Filenamen schon ins Directory eingetragen; wenn die Diskette dann aber voll ist, sind die Files 'leer', man muss dann von Hand die nicht kopierten Files suchen und dann File fuer File loeschen. Und die Eingabemoeglichkeit bei dem Parametern ',EDK,FRAG,...' ueberzeugt mich auch nicht immer. Andauernd gebe ich ein 'J' oder auch mal ein 'N' zuviel und ich muss mir den File dann aufschreiben, um ihn hinterher zu loeschen oder zu koperen, je nach dem. Und besonders praktisch ist das Sternchen, wenn man viele Files mit dem selben Anfangsbuchstaben hat die in einem ursaechliche Zusammenhang stehen, bei mir zum Beispiel Apfelmaenchendateien, die APFEL/CMD, und dann eben APFEL1/GRF bis APFEL29/GRF und dann noch die /DAT oder /NEW oder/HMH Dateien in welchen andere Daten oder die Bilder selbst verstecken. Oder ein anderes Beispiel, Kopiere alle Dateien die Sourcecodes von Sysfiles sind; hiesse dann Kopiere SYS*/SRC.

Nun ja, meine ersten Versuche machte ich in Basic. Aber trotz Compilierung war es einfach unertraeglich langsam, es nutzte also nichts, Assembler musste gelernt werden. Sehr geholfen haben mir dabei das DOS-Buch von H. Grosser sowie die kommentierten Assemblerroutinen die sich immer wieder in alten Club 80-Infen finden.

Also flugs ZEUS angeworfen und los gings. Sehr schnell stellte ich fest, das sich Z80-Assembler von Hochsprachen doch ziemlich unterscheidet, besonders die Beherrschung der bedingten Spruenge und Unterprogrammaufrufe machte mir zunaechst ziemliche Schwierigkeiten (ist das Z-Flag nun gesetzt oder nicht oder was ?). Aber nach einigen Tagen hatte ich dann endlich ein lauffaehiges, narrensicheres und vor allen Dingen sauschnelles Kopierutility vor mir.

Nun zum Programmablauf noch einige Erlaeuterungen. Die ersten 19 Zeichen des Programms sind reserviert und werden vom Programm mit Parametern gefuellt. Die Eingabe wird in ein Format uebertragen das dem Format des FDE (File Directory Eintrag) im Inhaltsverzeichnis der Disketten gleicht, also immer 11 Zeichen und kein Schraegstrich. Eventuelle Sterne im

Eingabestring werden in Fragezeichen umgewandelt. Danach werden die Laufwerksnummern von Source- und Destinationdrive gefunden und eingetragen. Wenn die Eingabe fehlerhaft war, wird eine Fehlermeldung mit der richtigen Syntax ausgegeben.

Die Anzahl der FDE-Sektoren einer Diskette, in der maximal 8 Eintraege fuer Files stehen koennen, findet sich im ersten Sektor des Inhaltsverzeichnisses in Byte 1Fh. Die Zahl wird ermittelt und eingetragen. Es werden genau soviele FDE-Sektoren mit dem Dosunterprogramm 'DIRR' geladen und wenn der Eintrag besteht (gesetztes bit 4 des ersten Bytes des Eintrages) und es sich weder um ein System-File (gesetztes bit 6) noch um einen Folgeeintrag (gesetztes bit 7) handelt, wird der Filename mit den 11 Bytes am Anfang des Programmes, in dem der geforderte Filename steht, verglichen. Dabei gilt ein Fragezeichen soviel wie jedes anders Zeichen, welches als Filenamezeichen erlaubt ist (Ziffern und Umlaute sowie beim Genie IIIs der untenstehende Strich).

Entspricht der Filename im Directory den geforderten Spezifikationen, dann wird er in einen Buffer am Ende des Programmes eingetragen. Davor wird aber jedesmal ein '> ', dies ist der COPY-Befehl des G-DOS eingertragen. Hinter den Filename wird noch 'LW(source),LW(dest)<ENTER>' eingetragen, um den Befehl mit einme Dos-Call ablaufen zu lassen. Dieser Vorgang wird wiederholt, bis alle FDE-Sektoren der Source-Disktette gelesen sind, so das sich am Programmende, in dem Buffer, eine Reihe von gueltigen Doskommandos (kann man sich mit DEBUG sehr schoen anschauen) befindet. Der Anfang des Buffers ist in 'buf2' zwischengespeichert, 'buf2' zeigt also auf den ersten Kopierbefehl. Sofern die Anzahl der zu kopierenden Files nicht null ist(ergibt dann eine Meldung) oder die Eingabe 'Kopiere */* ...' getaetigt wurde (bewirkt eine nochmalige Nachfrage 'Sollen alle Files kopiert werden', die mit einem grossen 'J' beantwortete werden muss) wird ein Kopierkommando nach dem anderen zunaechst angezeigt und dann ausgefuehrt.

Sollte eine Fehlermeldung auftreten, die eine volle Dikette oder ein volles Inhaltsverzeichnis anzeigt, dann wird der letzte zu kopierende File wieder geloescht, da er auf der Zieldiskette nicht ordnungsgemaess abgespeichert wurde.

Der einzige Fehler, den das Programm aufweist ist das im Eingabestring keine Ziffern fuer die Filespezifikation benutzt werden koennen. Das duerfte aber auch relativ nebensaechlich sein.

Wenn man das Programm anders nennen moechte, muss man im Sourcelisting in Zeile 330 eine Aenderung vornehmen. Der Eingabepuffer des DOS liegt ab Speicherstelle 4318h. Hier steht nachdem man <ENTER> eingegeben hat die vor dem ENTER liegende Eingabe, in diesem Fall zum Beispiel 'KOPIERE */*:0,:1'. Die Erkennung der Eingabe '*/*' sowie die Behandlung derselben habe ich folgendermasen geloest: An der Speicherstelle 4320h steht der erste der Sternchen. Wenn das Programm umbenannt ist, zum Beispiel in 'KOPY' wuerde ab 4318h folgendes stehen : 'KOPY */*:0,:1'. In Speicherstelle 4320h steht dann der Doppelpunkt vor der Null, das Programm kann den String '*/*' also gar nicht erkennen. Die Zahl in 330 muesste in diesem Fall also 431Dh heissen. An dieser Stelle befindet sich dann der erste Stern.
Sinngemaess gilt das auch fuer die Zeilen 333 und 337. Hier muessen die Zahlenwerte jeweils incrementiert werden

Das Grundgeruest dieses Programmes liesse sich fuer viele andere Geschichten nutzen, die Implemention eines 'ERASE B*/*' um Filegruppen zu loeschen (zum Beispiel genau die, welche gerade auf eine Sicherheitskopie mit 'KOPIERE' kopiert wurden) oder auch 'INV' bzw 'VIS' um Files unsichtbar oder sichtbar zu machen oder auch nur um unsichtbare Files anzuzeigen (und zwar nur die) waeren meiner Meinung nach huebsche Features. Wahrscheinlich fallen jedem von euch auch noch andere Geschichten ein, je nach dem was euch gerade beschaeftigt.

Das Programm ist so fuer jeden Rechner nutzbar, auf dem irgendeine Version des G-DOS lauft, weil ich nur ziemlich allgemein gehaltene DOS-Routinen aus dem SYS0/SYS benutze. Das sollte auf jedem G-DOS gleich sein.

Die NewDOS80-Benutzer muessen die Routine 'ein>' ab Zeile 274 bis Zeile 280 loeschen und die ';' also die Kommentarzeichen vor den Zeilen 285 bis 300 loeschen, so das dieser Teil jetzt die Routine 'ein>' darstellt.

Wer das Programm nicht abtippen will, kann mir eine Diskette im G-DOS Standartformat fuer 80-Spuren DS/DD schicken, und denn auch bitte Rueckporto beifuegen.
Noch mal die PDRIVE-Parameter:
TI=EHK TD=G SP=80 SEK=36 SWZ=0 EIB=6 SBIV=48 AEIV=2
TJa, in diesem Sinne
'DEM GUTEN DAS BESTE '
Happy Hacking !!!!!

PS.: In einer lauschigen Stunde habe ich das Sourcelisting umgeschrieben, auf das es jetzt die Funktion 'ERASE' darstellt.Die Hauptarbeit lag darin, die Kommentarzeilen umzuschrieben, der Rest ist wirklich ziemlich einfach.

Statt der Routine 'ein>' muss eine Routine geschrieben werden, welche den Befehl 'KILL' in den Buffer einschreibt. Ausserdem muessen die Stellen, wo Source- und Destinationlaufwerksnummern eingeschrieben oder abgefragt werden geaendert werden. Das Programm 'ERASE' werde ich auch zum Diskothekar schicken.

PPS.: Seit einigen Wochen bin ich im Besitz eines GENIE IIIs mit dem 1MByte-Umbau von Helmut Bernhardt. Sehr schade finde ich, dass es keinen vernuenftigen Grafikcompiler bzw. eine Library fuer den BASCOM gibt. Ich habe zum Beispiel fuer das Genie I im HRG-Pack-Paket eine Library fuer BASCOM. Jetzt hatte ich die Idee, diese File zu disassemblieren und umzuschreiben auf das es die HRG des G IIIs beharken moege. Wenn jemand schon mal an soetwas gedacht hat oder es gar geschafft hat oder es versuchen will oder auch nur so, kann er oder sie sich ja mal bei mir melden.
PPPS.: Ich suche noch eine BASCOM-Version fuer Genie-DOS, welche CMD-Files generiert, mein Linker erzeugt nur CHN-Files die mit BRUN gestartet werden.

```
                ;         K O P I E R E
                ;
                ;Kopieren von einer diskette auf eine andere
                ;mit angabe von wildcards
                ;       ? = beliebiger buchstabe
                ;       * = nachfolgende zeichen bis '/' beliebig
                ;
                ;
                ;   Syntax :=  kopiere FILENAME/EXT:lw#,:lw#<ENTER>
                ;
                ;
                ;
outch   EQU     0033h
getkey  EQU     0049h
dosrdy  EQU     402dh
doscal  EQU     4419h
extrfs  EQU     441ch
sendms  EQU     4467h
grosbu  EQU     45b5h
dirr    EQU     490ah
dirw    EQU     491fh
tesdsk  EQU     445eh
debug   EQU     440dh
doserr  EQU     4409h
        ;
        ;
        ORG     8000h
dri1    DM      'x'       ;source-lw#
dri2    DM      'x'       ;destination-lw#
secanz  DM      'x'       ; anzahl der zu lesenden
                          ; dirsectoren mit FDE's
buf1    DM      'xx'      ; zeiger auf buffer
buf2    DM      'xx'      ; zeigt auf anfang Bufferfeld
filzae  DM      00h       ; Anzahl der zu kopierenden Files
fname   DM      'FILENAME';ebenso vom Progr. eingesetzt
exten   DM      'EXT'
ende    DM      0DH       ;in den zur}ckliegenden 11 bytes
                          ;wird der Vergleichsfilespec
                          ;eingeschrieben um verglichen
                          ;zu werden
schlech DM      'Syntax := KOPIERE FILENAME/EXT'
        DM      ':<lw#s>,:<lw#d>ENTER'
        DM      0dh       ; Bei falscher Eingabe
start   LD      B,09h     ; max. 8Zeichen f}r Filename
        LD      DE,fname; Zeiger auf
loop1   LD      A,(HL)    ; zeichen holen
        CALL    grosbu    ; a->A, soweit m|glich
        CP      '?'       ; wenn ja einfach eintragen
        JR      Z,wei1
        CP      'A'
        JR      C,kzeich; wenn Carry gesetzt, ist
                          ;das Zeichen in der
                          ;ASCII-Tabelle kleiner als 'A'
        CP      '['
        JR      NC,kzeich; bzw. es ist oberhalb von 'Z'
wei1    LD      (DE),A    ; buchstaben eintragen
        INC     DE
        INC     HL        ; zeiger einen weiter
        DJNZ    loop1     ; bis 8 Zeichen
        JR      error     ; mehr als 8 !!
kzeich  CP      ':'       ; laufwerkstrennzeichen?
        JR      Z,error   ; enter ohne ext ist unzul{ssig
        CP      '/'       ; einziges zul{ssiges zeichen
        JR      Z,slash   ; weiter im text
        CP      '*'       ; jetzt wirds interessant
        JR      Z,stern   ; mal sehen
```

```
        JR      error     ; anderes zeichen !!
slash   LD      A,B       ; wegen DJNZ
        CP      0h        ; 8tes Zeichen ?
        CALL    NZ,nulauf; ausnullen mit spaces
        LD      DE,exten  ; zeiger auf extensionstring
ext     LD      B,04h     ; h|chstens drei Buchstaben !
loop2   INC     HL        ; eingangs z{hler ein weiter
        LD      A,(HL)
        CALL    grosbu    ; selbe routine wie oben
        CP      '?'       ;einfach weiter
        JR      Z,wei2
        CP      'A'
        JR      C,keexze; wenn enter, dann o.k.
        CP      '['
        JR      NC,keexze; dito
wei2    LD      (DE),A    ; buchstabe eintragen
        INC     DE
        DJNZ    loop2     ; bis 3 Zeichen
        CP      ':'       ; 13. Zeichen LW-Doppelpunkt ?
        JR      NZ,error; wenn nicht !!!!!!!!!!!
        JR      kopie     ; soweit in ordnung
keexze  CP      '*'       ; mit ?? auf}llen
        JR      Z,stern1
        CP      ':'
        CALL    NZ,error; jetzt folgt die LW#
        CALL    nulauf
        JR      kopie
nulauf  LD      A,' '     ; space
fill    LD      (DE),A    ; in fname eintragen
        INC     DE
        DJNZ    fill
        RET
stern   LD      A,'?'     ; fuellbyte '?'
        CALL    fill
        INC     HL        ; m}~te auf '/' zeigen
        JR      slash
stern1  LD      A,'?'     ; f}llbyte
        CALL    fill      ; ausf}llen,
        INC     HL        ; soll auf ':' zeigen
        LD      DE,ende   ; aufs ende von fname
        LD      A,(HL)    ; zeichen holen
        CP      ':'       ; wegen LW#
        JR      NZ,error
        JR      kopie     ; soll funktionieren
error   LD      HL,schlech
        CALL    sendms    ; error meldung ausgeben
        JP      dosrdy    ; vorl{ufiger ausstieg
istzahl SUB     '0'       ; ascii korrektur
        JR      C,error
        CP      10
        RET     C
        JR      error
kopie   INC     HL        ; jetzt wird erst mal auf
                          ; die Befehlssyntax
        LD      A,(HL)    ; also :lw#,:lw# getestet
        CALL    istzahl
        LD      (dri1),A; als   source laufwerk
        INC     HL        ; auf komma
        LD      A,(HL)
        CP      ','
        JR      NZ,error; keine andere eingabe erlaubt
        INC     HL        ; auf zweiten doppelpunkt
        LD      A,(HL)
        CP      ':'
        JR      NZ,error
        INC     HL        ; auf destination laufwerk
```

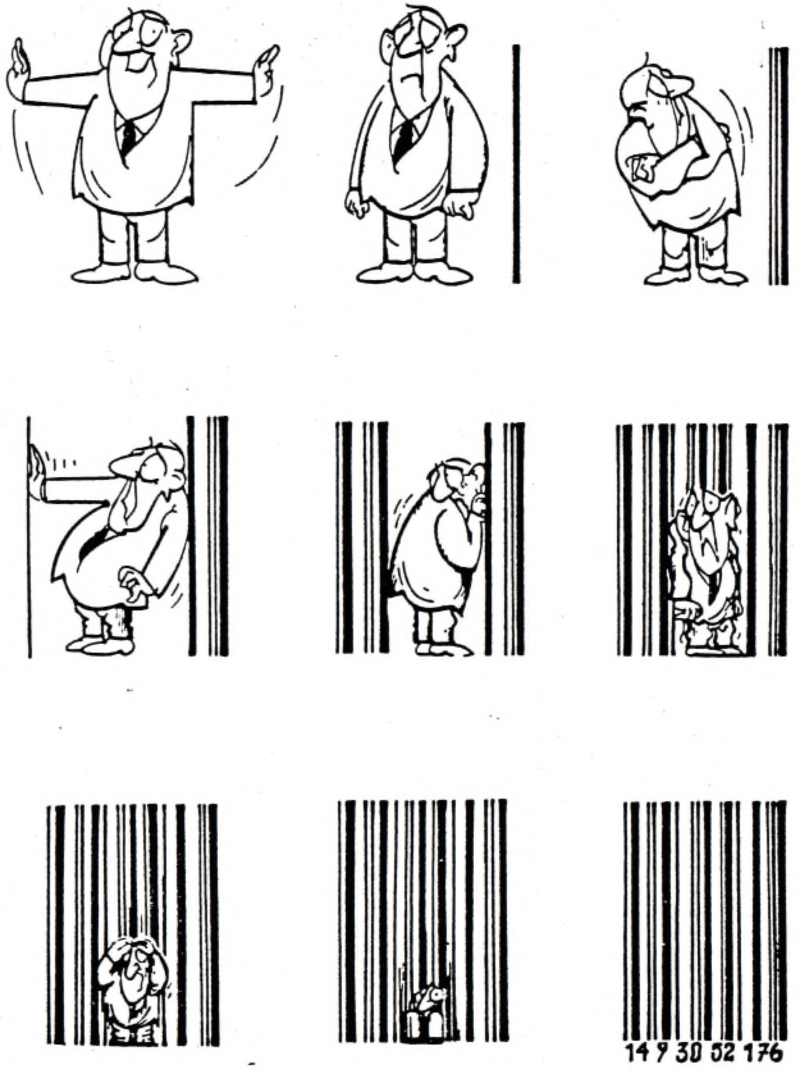
```
               LD      A,(HL)
               CALL    istzahl
               LD      (dri2),A; als dest-lw
               INC     HL      ; auf ENTER
               LD      A,(HL)
               CP      0dh
               JR      NZ,error;
               ; weiter im text, die Eingabe ist richtig
               ; und der vergleichs filename steht.
               ; die laufwerksnummern sind auch da.
               ; jetzt kann wirklich kopiert werden.
               ; es werden ersteinmal die eintr(ge in den
               ; buffer geschrieben
               LD      HL,buffer
               LD      (buf1),HL
               LD      (buf2),HL; anfang des Buffers
                                ; muss erhalten bleiben
       regsec  LD      A,(dri1); sourcelaufwerk eintragen
               CALL    tesdsk  ; disketten zugriff testen
               JP      NZ,doserr
               CALL    ermi    ; wieviele Dir-sectoren
               LD      A,(secanz)
               LD      B,A
               LD      A,1     ; erster dir-sector mit fde's
       liesec  INC     A       ; ist der Sector Nummer 2
               PUSH    AF      ; retten !!
               PUSH    BC      ; ebenso
               CALL    dirr    ; les einen sector in DOS-buffer
               CP      06h     ;'Lesevers. markierter satz'
               JR      Z,wei3  ; dann ist alles in ordnung
               JP      doserr  ; es ist ein fehler aufgetr.
       wei3    CALL    auswert ; g)ltige filespecs errech.
               POP     BC
               POP     AF      ; accu vom stack als z(hler
               DJNZ    liesec
               JP      copy    ; jetzt kann es richtig losgehen
       auswert LD      B,7     ; acht eint(ge pro sector
               LD      HL,4200h; erstes byte zeigt ob
                               ; ein Eintrag vorhanden ist
               LD      DE,4205h; zeigt auf ersten FILENAMEEXT
               CALL    verglei ; pr)ft ob der eintrag den
       loop3   LD      A,20h   ; spezifikationen
               ADD     A,L     ; entspricht. Ein FDE ist
               LD      L,A     ; 20h Byte lang.
               LD      A,20h
               ADD     A,E
               LD      E,A
               CALL    verglei ; siehe 8 Zeilen weiter oben
               DJNZ    loop3
               RET             ; acht eintr(ge sind gepr)ft
       verglei PUSH    HL      ; HL und DE werden gerettet
               PUSH    DE
               PUSH    BC      ; mu^ auch gertettet werden
               LD      B,0bh   ; 11 zeichen ist fname lang
               LD      A,(HL)  ; sehen ob eintrag besteht
               CP      00h     ; null, wenn nicht
               JR      Z,zur)  ; erst noch pop de und hl
               CP      90h     ; ist es ein Folgeeintrag ???
               JR      Z,zur)  ; dann nicht eintragen !!
               BIT     6,A     ; bit 6 ist '1' bei /SYS-Files
               JR      NZ,zur) ; weil die nicht kopiert
                               ; werden d)rfen (oder sollen)
                               ; vor allen Dingen nicht Boot und
                               ; Inhalt/SYS !!!!!!!!!!!!!!!!!!
               LD      HL,fname; enth(lt spezifizierten fname
       wei4    LD      A,(DE)  ; zeichen im buffer
               CP      (HL)    ; vergleiche (hl) mit (de)
               JR      NZ,wei7 ; wenn nicht gleich, pr)fen ob ?
       wei5    INC     HL
               INC     DE      ; zeiger auf n(chstes zeichen
               DJNZ    wei4    ; 11 zeichen vergleichen
       eintr   NOP             ; alle zeichen entsprechen den
                               ; spezifikationen, jetzt m)ssen
                               ; sie in einen Buffer
                               ; eingetragen werden, allerdings
                               ; in der Form FILENAME/EXT
               LD      A,E     ; DE mu^ um 11 erniedriegt werden
               SUB     11      ; damit DE wieder auf Anfang des
               LD      E,A     ; filespecs im FDE zeigt
               LD      A,(filzae); z(hler f)r zu kop. Files
               INC     A       ; + 1
               LD      (filzae),A     ; und speichern
               LD      HL,(buf1); buf1 ist der zeiger auf buffer
               CALL    ein>    ; der DOS-Befehl '> 'wird
                               ; vor die filespecs
                               ; eingetragen wegen doscall
               LD      B,8     ; acht zeichen h!chstens
       loop4   LD      A,(DE)  ; erstes zeichen im buffer
               CP      ' '     ; ist es space ?
               JR      Z,nein
               LD      (HL),A  ; zeichen eintragen
               INC     HL
       nein    INC     DE      ; zeiger weiter
               DJNZ    loop4   ; bis 8 zeichen
               LD      A,'/'
               LD      (HL),A  ; Trennzeichen eintragen
               INC     HL
       ext2    LD      B,3     ; 3 extension zeichen
       loop5   LD      A,(DE)  ; erstes zeichen
               CP      ' '
               JR      Z,nein2
               LD      (HL),A
               INC     HL
       nein2   INC     DE
               DJNZ    loop5
               DEC     HL      ; abfrage, ob gar keine extension
               LD      A,(HL)  ; vorhanden war, dann mu^ der '/'
               CP      '/'     ; n(mlich auch wieder weg
               JR      Z,wei6
               INC     HL
       wei6    LD      A,':'   ; die zeichen ':lw#,:lw#',0dh
               LD      (HL),A  ; werden hinter jeden filenamen
               INC     HL      ; eingetragen
               LD      A,(dri1)
               ADD     A,48    ; ascii-ausgleich !! accu+48d
               LD      (HL),A  ; dann wird aus 01h 49h,'A' !!
               INC     HL
               LD      A,','
               LD      (HL),A
               INC     HL
               LD      A,':'
               LD      (HL),A
               INC     HL
               LD      A,(dri2) ; destination drive
               ADD     A,48    ; ascii-ausgleich !!!!!!!!!!
               LD      (HL),A
               INC     HL
               LD      A,0dh   ; ENTER kennung f)r COPY befehl
               LD      (HL),A  ; eintragen
               INC     HL      ; zeigt auf das zeichen nach 0dh
               LD      (buf1),HL       ; buf1 aktualisieren
               ; der filename wurde in den buffer eingetragen
```

```
00265           ; es wird eine ebene zur)ck gesprungen, ob der
00266           ;  nexte eintrag g)ltig ist.
00267 zur)      POP     BC      ; die parameter werden
00268           POP     DE      ; zur)ckgepopt
00269           POP     HL
00270           RET
00271           ; die Routine ein) f)gt den GDOS-Befehl '> '
00272           ; also 'COPY ' ein.
00273           ; HL zeigt auf den Buffer
00274.ein)     LD      A,'>'
00275           LD      (HL),A
00276           INC     HL
00277           LD      A,' '
00278           LD      (HL),A
00279           INC     HL
00280           RET
00281 ;die folgende Routine ist der Ersatz fuer die 'ein)'
00282 ; in Zeile 273 bis 279 falls das Betriebssystem
00283 ; NewDOS 80 benutzt wird.
00284 ; Zeile 274 bis 280 mussen dann geloescht werden.
00285 ;ein)     ld      a,'C'
00286 ;         LD      (HL),A
00287 ;         INC     HL
00288 ;         LD      A,'O'
00289 ;         ld      (hl),a
00290 ;         INC     HL
00291 ;         ld      a,'P'
00292 ;         ld      (hl),a
00293 ;         inc     hl
00294 ;         ld      a,'Y'
00295 ;         ld      (hl),a
00296 ;         inc     hl
00297 ;         ld      a,' '
00298 ;         ld      (hl),a
00299 ;         inc     hl
00300 ;         ret
00301 wei7      LD      A,'?'   ; test ob der joker gesetzt ist
00302           CP      (HL)    ; vergleiche mit fname
00303           JR      Z,wei5  ; wenns ein '?' war, ist der
00304                           ; Buchstabe o.K., ansonsten
00305           JR      zur)    ; braucht fname nicht weiter
00306                           ; getestet werden, er entspricht
00307                           ; nicht
00308                           ; der geforderten Spezifikation
00309 ermi      LD      A,1     ; im ersten sektor des dir
00310           CALL    dirr
00311           LD      A,(421fh); an dem byte 1fh befindet sich
00312           ADD     A,08h   ; die angabe , wieviele fde-
00313                           ; sectoren das directory
00314                           ; enth(lt.
00315           LD      (secanz),A      ; speichern
00316           RET
00317 ;
00318 ;         Jetzt folgt der eigentliche Kopiervorgang
00319 ;         es wird vorher getestet auf anzahl der Files =  0
00320 ;         und auf */*, also kopiere alles
00321 ;         au~erdem wird nach dem letzten Eintrag als
00322 ;         Endemarkierungein 'E' eingesetzzt
00323 ;
00324 copy      LD      A,'E'           ; Endemarkierung
00325           LD      (HL),A          ; nach dem letzten Befehl
00326           LD      A,(filzae)      ; keine Files  kopieren ?
00327           CP      00h
00328           JR      Z,keifil        ; Programm beenden
00329                                   ; vorher Meldung ausgeben
00330           LD      A,(4320h)       ; erstes Zeichen nach
00331                                   ; 'KOPIERE '
00332           CP      '*'             ; falls '*/* folgt
00333           JR      NZ,goon
00334           LD      A,(4321h)
00335           CP      '/'
00336           JR      NZ,goon
00337           LD      A,(4322h)
00338           CP      '*'
00339           JR      NZ,goon
00340           LD      HL,meld1
00341           CALL    sendmg
00342           LD      HL,meld2
00343           CALL    sendmg
00344           CALL    getkey
00345           CP      'J'
00346           JP      NZ,dosrdy
00347           LD      A,0dh   ; noch ein ENTER ausgeben
00348           CALL    outch   ; auf dem bildschirm ausgeben
00349           JR      goon
00350 keifil    LD      HL,meld3 ; kein File der Source-Disk
00351                           ; entspricht den
00352           CALL    sendmg  ; geforderten Spezifikationen
00353           JP      dosrdy
00354 goon      LD      HL,(buf2)       ; anfang des Buffers
00355 wei8      CALL    sendmg  ; der Copy-Befehl wird auf dem
00356                           ; Bildschirm angezeigt.
00357           LD      HL,(buf2)
00358           CALL    doscal  ; der Befehl wird ausgef)hrt
00359           CALL    NZ,keinfe; falls ein Fehler aufgetr. ist
00360           LD      A,(3840h)       ; tastaturzeile mit BREAK
00361           BIT     2,A     ; Abbrechen mit <BREAK>-Taste
00362           JR      Z,wei9
00363           LD      A,39h   ; fehler 'Function Aborted'
00364           JP      doserr
00365 ; Suchroutine um den n(chsten Befehl zu finden
00366 wei9      LD      HL,(buf2); anfang des letzten Befehls
00367           LD      B,0
00368           LD      C,9; der n(chste Befehl kommt fr)hestens
00369           ADD     HL,BC           ; nach 9 Zeichen.
00370 loop6     LD      A,(HL)
00371           CP      0dh     ; kommt schon der n(chste ?
00372           JR      Z,wei10
00373           INC     HL              ; zeiger ein weiter
00374           JR      loop6           ; ohne ende .
00375 wei10     INC     HL              ; ein hamwa noch
00376           LD      A,(HL)
00377           CP      '>'             ; normaler Befehl ?
00378           JP      NZ,dosrdy  ;  wenn nicht, wars der letzte
00379           LD      (buf2),HL  ; buf2 zeigt auf den n(chsten
00380           JR      wei8     ;Befehl und so weiter und so fort
00381 keinfe    CP      1ah     ; fehlermeldung 'Directory voll'
00382           JR      Z,loefil; letzten File wieder loeschen
00383           CP      1bh     ; 'Diskette voll '       ??????
00384           JR      Z,loefil
00385           CP      1eh     ; 'Directory voll bei Erw. ' ??
00386           JR      Z,loefil
00387           CP      21h     ; 'Kein Platz auf Diskette ' ??
00388           JR      Z,loefil
00389           JP      doserr
00390 loefil    LD      HL,4326h; letztes (moegliches)
00391                           ; signifikantes
00392                           ; Zeichen des Copy-Befehls
00393           LD      DE,4329h        ; die ganze Zeichenkette
00394                                   ; soll um 3 Zeichen nach
00395                                   ;hinten verschoben werden
00396           LD      BC,000dh        ; sie ist 14 Zeichen lang
```

```
                                  ; und zwar 'Filename/Ext'
                                  ; + <ENTER> + ein blank
                                  ; nach dem '>' !!
                LDDR              ; verschiebebefehl
                                  ; (Mit DEC der Parameter !)
                LD      HL,kill   ; source ist 'KILL '
                LD      DE,4318h  ; Anfang  Befehlsbuffer
                LD      BC,0005h  ; }bertrage 5 Zeichen
                LDIR              ; also los
                EX      DE,HL     ; Austausch HL <> DE
        findde  INC     HL        ; suche den Doppelpunkt
                                  ; f}r Destination
                LD      A,(HL)    ; Drive Eintrag
                CP      ':'
                JR      NZ,findde ;solange bis ':' gefunden
                INC     HL        ; zeiger auf Drive-#
                LD      A,(dri2)  ; Destination-Drive
                ADD     A,'0'     ; ASCII-Ausgleich
                LD      (HL),A    ; eintragen
                INC     HL
                LD      A,0dh     ;<cr> am Schlu~ eintragen
                LD      (HL),A
                LD      HL,4318h  ; Befehlsbuffer des DOS
                CALL    sendmg    ; auf Display anzeigen
                LD      HL,4318h
                CALL    doscal    ; ausf}hren
                JP      Z,noerr   ; zur}ck ins DOS
                LD      HL,meld4  ; ansonsten Fehlermeldung
                CALL    sendmg    ; ausgeben
                JP      dosrdy    ; und zurueck ins DOS
        noerr   LD      HL,meld5  ; 'Diskette voll' ausg.
                CALL    sendmg
                JP      dosrdy
        kill    DM      'KILL '           ; DOS-BEFEHL KILL
        meld1   DM      'Wirklich alle Files kopieren ?',0dh
        meld2   DM      'Dann jetzt <J> dr}cken ',03h
        meld3   DM      'Keine Files der Spezifikation '
                DM      'vorhanden.',0dh
        meld4   DM      'Der letzte kopierte File muss '
                DM      'noch gel|scht werden.',0dh
        meld5   DM      'Diskette voll !!!!!!',0dh
        buffer  EQU     $
                END     start
```

Leider ist mein Drucker noch älter als mein Computer, er kennt keine Umlaute.

Ä = {
Ö = \
Ü = }
ß = ~



© Walter Hanel

Ein bißchen C, ein bißchen Mathe . . .

Vor einiger Zeit hatte ich die Idee, mein C-System um einige mathematische Routinen zu ergänzen. Mit den vier Grundrechenarten war ich nicht ganz so zufrieden, wie es sich die Entwickler des C-Compilers vieleicht gedacht hatten. Bevor so ein Vorhaben aber zu brauchbaren Algorithmen führt, stellte sich mir das Problem selbige erst einmal ausfindig zu machen. Wer hat schon soetwas als Nichtinformatiker in der Schublade liegen.

Gestecktes Ziel war, die fehlenden Funktionen - immerhin alle trigonometrischen Operationen - mit möglichst schnellen Algorithmen aufzubauen. Ausscheiden würden somit schon einmal alle höheren Operationen oberhalb "+" und "-". Es mußte also alles so weit als möglich mit Addition und Subtraktion erledigbar sein - zumindest in den zeitfressenden Berechnungsschleifen. Gemacht, getan. Da mir nichts mit viel Mathematik in gedruckter Form zur Hand gehen konnte, war mein erster Schritt nach einem Buch oder auch mehreren, zu suchen und aus diesen die gesuchten Rechenanweisungen herauszuziehen.

Die Rechenfunktionen basieren alle auf "Unendlichen Zahlenfolgen". Beim "TAN(x)" habe ich auf "SIN(x)/COS(x)" zurückgebriffen, da hier eine Folge zur Berechnung der Bernoulischen Zahlen erfoderliche ist. Die Berechnung ist recht aufwendig und deshalb habe ich mir diesen Akt verkniffen. Anzumerken ist noch, das die vorgestellten Routinen noch nicht der "Weisheit lezter Stuß" darstellen. Eine der Algorithmen konvergieren bei bestimmten Winkeln nur sehr schlecht, d.h., sie rechnen und rechnen und .... Zum Teil auch deshalb, weil die Genauigkeit des Fließkomma-Routinen meines C-Compilers nur 6 ... 7 Stellen umfaßt, was, wie ich mit GFA-Basic feststellen konnte, eine der Ursachen ist. Grundsätzlich bestimmt "FLT_EPSILON" die Genauigkeit der Ergebisse. Zu beachten ist aber, daß die erreichte Genauigkeit des errechneten Wertes geringer ausfällt als "FLT_EPSILON" angibt. Bei "FLT_EPSILON" gleich 1E-6 wird auf 6 stellen hinter dem Komma gerechnet. Wenn sich dann das (n-1)-te in der Differenz von n-ten Ergebnis um weniger als "FLT_EPSILON" unterscheidet, wird die Berechnung abgebrochen. Bei 1E-9 ist also nach 9 stellen Schluß. Damit diese Genauigkeit aber auch erreicht wird, muß die benutzte Fießkomma-Arithmetik mit einer größeren Stellengenauigkeit rechnen als man erreichen will !!! Sonst geht die Sache wegen der unvermeidlichen Rechenfehler schief. Das sicherste ist, die Stellenzahl zwei Stellen weniger als möglich einzustellen. Dann bleibt genügend Luft für Rundungsfehler etc.

Das Buch, aus dem ich alles entnommen habe nun als letztes (S. 603):

> *"Rechnen und Mathematik"*
> Das Lexikon für die Schule und Praxis
>
> Dudenverlag Manheim/Wien/Zürich
> 4. Auflage, ISBN 3-411-02423-2
> Preis: DM 36.--

In dem Schinken steht auch noch einiges mehr. Wer sich in Sachen Algorithmen noch weiter vergraben will, kann es hier ausgiebig tun.

Tschüß

Kurt


```
/*
 * Die nachfolgenden "#define"-Anweisungen legen einige Konstanten
 * zu den Algorithmen fest
 *
 */

#define M_PI            3.14159265
#define M_PI_2          (M_PI)/2
#define M_LN10          2.30258509
#define FLT_EPSILON     1E-6

/*
 * Funktion zur Berechnung des FMOD(X,Y)
 * Es wird der Gleitkomma-Rest der Modulo-Funktion
 * X/Y berechnet ! Hierzu wird bestimmt, wie
 * oft X ganzzahlig durch Y dividierbar ist und mit
 * dem gefundenen Wert rückgerechnet.
 *
 */
float fmod(x,y)
float x,y;
{
    if (x>=0)
       return (x-(float)((int)(x/y))*y);
    else
       return (x-(float)((int)((x/y)+.5))*y);
}

/*
 * Die Funktion 'frac(x)' liefert den Nachkomma-
 * teil einer gebrochen rationalen Zahl.
 *
 */
float frac(x)
float x;
{
    if (x>=0)
       return (x-(float)((int)(x)));
    else
       return (x-(float)((int)((x)+.5)));
}

/*
 * Funktion zur Berechnung des SIN(X).
 * 'X' ist im Bogenmaß !
 *
 *              x^3   x^5   x^7
 * sin(x)=x - --- + --- - --- + ...     (für x Element von R)
 *              3!    5!    7!
 */
float sin(x)
float x;
{
    register
    float n    = 2.,
```

```
          x2    = 0.,
        x_alt = 0.,
          y     = 0;

    if(abs(x)>M_PI) x = fmod(x,M_PI);
    x2   = x*x;
    y    = x;
        do
    {
        x_alt = y;
  x   *= (-x2)/(n*(n+1.));
        y  += x;
        n  += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
        return(y);
}

/*
 * Funktion zur Berechnung des SINH(X).
 * 'X' ist im Bogenmaß !
 *
 *                x^3   x^5   x^7
 * sinh(x)=x + --- + --- + --- + ...     (für x Element von R)
 *                3!    5!    7!
 */
float sinh(x)
float x;
{
    register
    float n     = 2.,
          x2    = 0.,
        x_alt = 0.,
          y     = 0;

/*    if(x>M_PI_2) x = fmod(x,M_PI_2);*/
    x2   = x*x,
    y    = x;
    do
    {
        x_alt = y;
        x   *= x2/(n*(n+1.));
        y  += x;
        n  += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
        return(y);
}

/*
 * Funktion zur Berechnung des cos(X).
 * 'X' ist im Bogenmaß !
 *
 *                 x^2   x^4   x^6
 * cos(x)= 1 - --- + --- - --- + ...     (für x Element von R)
 *                 2!    4!    6!
 *
 */
```



```
 */

float cos(x)
float x;
{
    register
    float n     = 1.,
          x2    = 0.,
        x_alt = 0.,
          y     = 1.;

    if(abs(x)>M_PI) x = fmod(x,M_PI);
    x2   = x*x,
        x = 1.;
    do
    {
        x_alt = y;
        x   *= (-x2)/(n*(n+1.));
        y  += x;
        n  += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
        return(y);
}
/*
 * Funktion zur Berechnung des cosh(X).
 * 'X' ist im Bogenmaß !
 *
 *                 x^2   x^4   x^6
 * cosh(x)=1 + --- + --- + --- + ...     (für x Element von R)
 *                 2!    4!    6!
 *
 */

float cosh(x)
float x;
{
    register
    float n     = 1.,
          x2    = 0.,
        x_alt = 0.,
          y     = 1.;

/*    if(x>M_PI) x = fmod(x,M_PI);*/
    x2   = x*x;
        x = 1.;
    do
    {
        x_alt = y;
              x   *= x2/(n*(n+1.));
        y  += x;
        n  += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
        return(y);
}
```

```
/*
 * "X" im Bogenmaß
 *
 *                  1       1
 * atan(x) = x - - x^3 + - x^5-...+...
 *                  3       5
 *
 */
float tan(x)
float x;
{
    return((abs(x)<0.005) ? x : (sin(x)/cos(x)));
}

float atan(x)
float x;
{
    register
    float n     = 3.,
          x2    = x*x,
          x_alt = 0.,
          y     = x;

    do
    {
        x_alt = y;
        x  *= (-x2);
        y  += (x/n);
        n  += 2;
    } while (abs(x_alt-y)>FLT_EPSILON);
        return(y);
}

float cot(x)
float x;
{
    return((abs(x)<0.005) ? (1/x) : (cos(x)/sin(x)));
}


/*
 * |X| < 1
 *
 * 1     1+x        1       1
 * - ln --- = x + - x^3 + - x^5 + ...
 * 2     1-x        3       5
 *
 */
float ln(x)
float x;
{
  register
  float d = (x-1.)/(x+1.),
        d2= d*d,
        n = 3.,
```



```
        x = d,
        y = 0.;

  do
  {
    y  = x;
    d *= d2;
    x += (d/n);
    n += 2.;
  } while (abs(x-y)>FLT_EPSILON);

  return(x+x);
}


/*
 *  x Element der reellen Zahlen
 *
 *                    x^2   x^3
 *  exp(x) = 1 + x + --- + --- + ...
 *                    2!    3!
 *
 */
float exp(x)
float x;
{
    register
    float n    = 1.,
        x_alt = 0.,
            y  = 1.,
            z  = 1.;

    do
    {
        x_alt = y;
        z     *= x/n;
        y     += z;
        n     += 1.;
    } while (abs(y-x_alt)>FLT_EPSILON);
        return(y);
}

/*
 * Von der rechnung her wie "EXP(X)", es wird nur entsprechend
 * umgerechnet
 *
/*
float exp10(x)
float x;
{

    register
    float n    = 1.,
        x_alt = 0.,
            y  = 1.,
```

```
          z  = 1.;

    x *= M_LN10;
    do
    {
        x_alt = y;
        z     *= x/n;
        y     += z;
        n     += 1.;
    } while (abs(y-x_alt)>FLT_EPSILON);
        return(y);
}

/*
 * Hier ist mir der Zettel verloren gegangen. Pech, aber die
 * Rechnung kann sich jeder selbst rekonstruieren
 *
/*
float sqrt(x)
float   x;
{
    register
    float   x1 = 1.;
            x2 = x1;
    int      i = 0;

    do
    {
      x1=x2;
      x2=(x1+(x/x1))/2;
      i++;
    } while (abs(x1-x2)>FLT_EPSILON);

    return(x2);
}


main()
{
    float x = 8.9;

    printf("Funktionsargument x = %2.2f\n",x,"\n");/**/
    printf("FMOD(x,8)= %3.7f\n",fmod(x,8.));         /**/
    printf("FRAC(x)  = %3.7f\n",frac(x));             /**/
    printf("SIN(x)   = %3.7f\n",sin(x));              /**/
    printf("COS(x)   = %3.7f\n",cos(x));              /**/
    printf("SINH(x)  = %3.7f\n",sinh(x));             /**/
    printf("COSH(x)  = %3.7f\n",cosh(x));             /**/
    printf("TAN(x)   = %3.7f\n",tan(x));              /**/
    printf("COT(x)   = %3.7f\n",cot(x));              /**/
    printf("ATAN(x)  = %3.7f\n",atan(x));             /**/
    printf("LN(x)    = %3.7f\n",ln(x));               /**/
    printf("EXP10(x)= %3.7e\n",exp10(x));             /**/
    printf("EXP(x)   = %3.7e\n",exp(x));              /**/
    printf("SQRT(x)  = %3.7f\n",sqrt(x));             /**/
}
```


..Hardware ..Hardware

Umrüstung ade !

Da hatte mir Fritz Chwolka vor längerer Zeit mal am Telefon gesagt, er wüßte von einem Knaben, der seinen 8085-Computer umgerüstet hätte auf Z80.

Das war natürlich die Sensation für mich! Ja, ich benütze noch immer diese Kiste mit CPU 8085A und bin es zuweilen auch zufrieden damit (AlphaTronic P3 !! hi hi). So bemühte ich mich denn monatelang, bei der von Fritz genannten Adresse anzurufen, bis ich endlich Erfolg hatte. Aus den mir zugesandten Unterlagen konnte ich die Adresse einer Firma rekonstruieren, die mir dann weitere Unterlagen zusandte und mir anbot, mir versuchshalber eine Umrüstplatine zuzusenden.

Da ich mir der technischen Details noch nicht ganz sicher war, beließ ich es erstmal bei den Unterlagen.
Diesen Unterlagen zufolge handelt es sich um eine Platine mit Baustein NSC 800 (das ist eine Z80-CPU) und einem Chip 74LS240, die anstelle des 8085 eingesteckt wird. Das ist fast alles. Je nach Taktfrequenz sind das dann verschiedene Typvarianten. Bei meinen 6 MHz Takt (nicht schlecht, was ?) sollte das Ganze dann ca. 140 DM kosten.

Ich hatte mich schon fast dazu entschlossen, diesen Betrag zu investieren, als ich doch noch mal mein techn. Handbuch des P3 durchblätterte, und feststellte, daß der Drucker über SID/SOD - Leitungen angesteuert wird. Das ist eine eigene kleine serielle Schnittstelle, die die CPU 8085 direkt anbietet.
Und das wars denn auch, denn als Z80-Processor hat der NSC800 diese Leitungen natürlich auch nicht und ich hätte nach der Umrüstung nicht mehr drucken können.

Aus und vorbei! (Änderungen des Betriebssystems und des Bios ziehe ich wegen fehlender Unterlagen gar nicht in Betracht).

Sollte jemand dennoch Interesse an der Sache haben, Fritz bekam die Unterlagen von mir und auch ich gebe gern Auskunft.

Mit freundlichen Grüßen

Uwe Schoberth

CPU280 – die Z80-VAX

Ich hatte schon befürchtet, daß ich mich vor Drohbriefen nicht mehr retten kann, wenn ich den 101. Artikel zum Z280 verzapfe, aber bis jetzt scheint (noch) niemand was darüber schreiben zu wollen. Müssen erst die Namen der Besitzer veröffentlicht werden, damit die was von sich hören lassen ?

Na gut, dann will ich mal mit gutem Beispiel voran gehen. Worum geht es bei diesem Rechner, über den hin und wieder so geheimnisvolle Andeutungen in diversen Artikeln aufgetaucht sind ? Ganz einfach, im Prinzip geht es um einen Computer, der als Prozessor keinen schnöden Z80, sondern einen Z280 hat. Der Z280 ist das, was vor langer Zeit mal der Z800 werden sollte und wo der HD64180 bzw. Z180 auf halbem Weg verhungert ist. Man könnte den Z280 als den 80286 der Zilog-Welt bezeichnen und im Grunde handelt es sich mehr um einen 16-Bit- als um einen 8-Bit-Prozessor. Um nur einige Highlights kurz zu nennen: Rund 600 Befehle, die über den Befehlssatz des Z80 hinausgehen (16-Bit-Multiplikation, 32-Bit-Division, relative Adressen und Sprünge über die gesamten 64K...), 3-stufige Befehlspipeline, eine on-chip MMU, die 16MB verwalten kann, 256 Byte Cache, drei 16-Bit Zähler/Timer, vier DMA-Kanäle und eine serielle Schnittstelle. Von der Leistung her ist der Chip dem 80286 mindestens ebenbürtig, wenn nicht sogar überlegen. Trotzdem ist er 100%ig kompatibel zum Z80 und das macht ihn überhaupt erst wirklich so interessant.

Aber nun zum eigentlichen Rechner, der CPU280 (wer als erster errät, woher der Name kommt, bekommt ein Autogramm). In der "Grundausstattung" besteht der ganze Compi lediglich aus einer Europakarte, die den Prozessor mit 64 oder 128KB ROM und bis zu 4MB (!) dynamischem RAM, den Floppycontroller (kompatibel zum 765), zwei interruptfähige serielle Schnittstellen und eine akkugepufferte Echtzeituhr mit 50 Bytes statischem Ram für Konfigurationsdaten beherbergt. Mit einem Netzteil, einem Terminal und zwei Laufwerken hat man schon einen kompletten CP/M-Rechner zusammen und kann loslegen. Das CP/M wird normalerweise "in Nullzeit" direkt aus dem EPROM gebootet, zu Testzwecken kann man das System aber auch von der Floppy laden. Die Taktrate beträgt im Moment 12.288MHz, was in der Rechenleistung einem Z80 bei etwa 16-20MHz entspricht, je nach Programm. Wenn man (=Stefan Nitschke) mal wirkliche Rechenpower demonstrieren will und dazu echten Z280-Code benutzt, liefert die Karte ein Apfelmännchen mit 720x348 Punkten und 40 Schritten Iterationstiefe in ca. einer halben Minute ab. Damit die Karte mit der alten ECB-Welt kompatibel ist, wird das Businterface mit einem Takt von rund 6MHz betrieben, intern wird alles aber selbstverständlich 16 bittig und ohne Waitzyklen mit voller Geschwindigkeit betan.

Worin unterscheidet sich das Board nun von anderen CP/M-Rechnern ? Da bis zu 4MB RAM, im platzsparenden ZIP-Gehäuse, bestückt werden können, hat man beim Vollausbau, wenn man die diversen Puffer abzieht, eine Ramdisk von rund 2-3MB zur Verfügung und da der Z280 hier im 16-Bit-Burst-Modus (4 Worte werden mit 2 Speicherzyklen gelesen) betrieben wird, eine verdammt schnelle. Zum Thema Disketten hat sich Tilmann Reh auch etwas ganz feines einfallen lassen. Man kann auf die Disketten einen Parameterblock schreiben, sodaß der Rechner das Format automatisch erkennen kann. Damit kann man in einem Laufwerk z.B. beliebig zwischen HD-Floppies mit 1.44MB (5

...Hardware

1/4" !) und 40 Track SS SD mit 70KB wechseln, ohne daß man etwas per Hand einstellen muß, wodurch sich dieser Rechner hervorragend als Formatkonverter und Kopierstation eignet. Auf 3 1/2"-Disketten gehen sogar 1.77MB, IBM war da unnötigerweise etwas sehr großzügig mit seinen Formaten.

Den Rechner mit CP/M zu fahren ist eigentlich wie ein Porschemotor in einem Goggo, aber CP/M ist leider nun mal DAS Betriebssystem für den Z80. Wenn man nur CP/M laufen lassen will, kann man sich genausogut einen anderen Composter zulegen, obwohl man wohl länger suchen muß, bis man für knapp 600 Mark diese Ausstattung auf einer Karte findet. Wer aber Lust am Experimentieren hat, der ist mit diesem Chip bestens bedient und wenn dann noch das Club-Terminal als Ein-/Ausgabe dient, dürfte die CPU280 einer der leistungsfähigsten CP/M-Rechner sein, die es momentan gibt. Natürlich läuft die Version 3.1, die selbstredend alle Features der Karte voll unterstützt, und wer ZCPR kennt, weiß was Z3PLUS auf so einer Maschine bedeutet. In nächster Zukunft gibt es noch eine AT-Bus-Karte, sodaß man auch in Sachen Massenspeicher ganz vorne mitmischen kann. Da läuft einem doch das Wasser im Munde zusammen, oder ? Eine der wenigen Alternativen wäre natürlich das Z280-Board für das Model I oder IV, aber ob das jemals kommt ? Auf dem nächsten Clubtreffen sind mit Sicherheit wieder mehrere CPU's anzutreffen, dann könnt Ihr mal "Probefahren". Für Anfragen stehe ich natürlich gerne zur Verfügung, wer sich aber lieber gleich an den Entwickler wendet, seine Adresse ist:

Tilmann Reh
In der Großenbach 46
5900 Siegen

Alexander Schmid

© Uli Stein

»Warum gehst du nicht raus, beschäftigst dich ein bißchen im Garten und läßt mich in Ruhe arbeiten?«

Einfachst A/D- und D/A-Wandler

Es hat auch in unserer Postille schon diverse Bauanleitungen für A/D- und D/A-Wandler gegeben, aber beides zusammen, und dann noch in einem IC, habe ich noch nicht gesehen. Mit noch weniger Chips geht's ja wirklich nicht, hier dürfte die Untergrenze an Bauteilen endgültig erreicht sein.

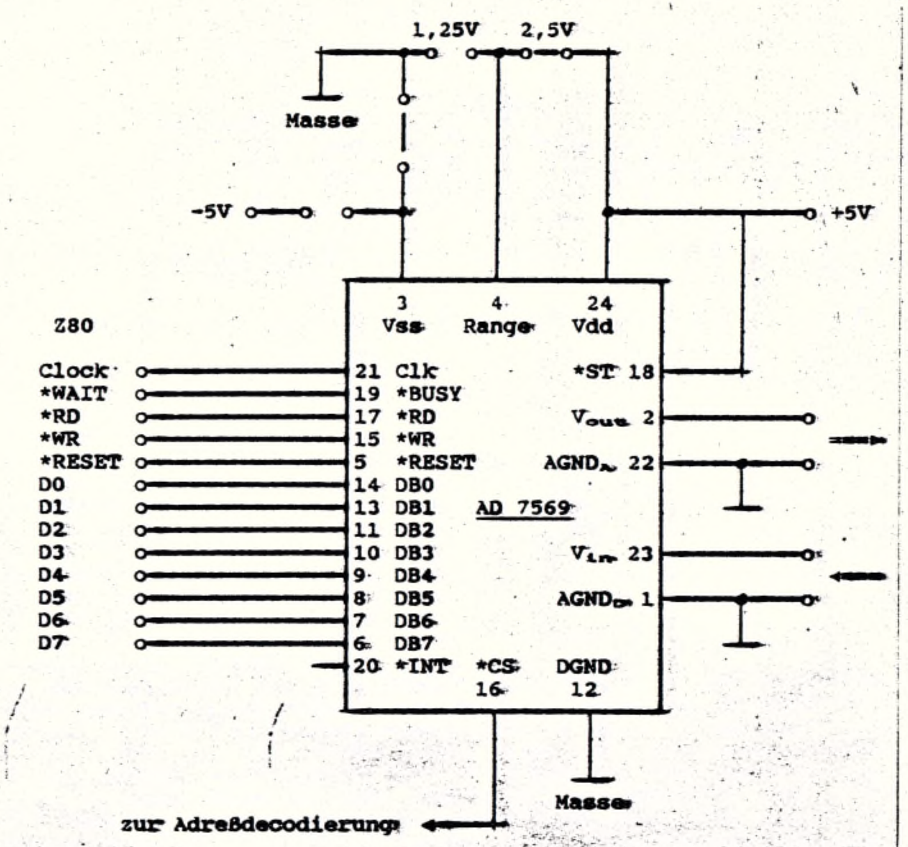
Der AD 7569 von Analog Devices enthält alles was man dazu braucht, einen 8-bit-A/D-Wandler mit 2us Wandlungszeit, einen D/A-Wandler mit 1us, die Referenzspannungsquelle und ein Businterface für die direkte Kopplung mit dem Rechnerbus. Bei einer unsymmetrischen Versorgung mit 5V stehen als Ein- und Ausgangsspannungsbereiche 0...1,25V oder 0...2,5V (abhängig vom logischen Pegel am RANGE-Eingang) zur Verfügung. Bei einer symmetrischen Versorgung gehen die Werte entsprechend nochmal genauso weit in den negativen Bereich. Die Adreßdecodierung ist nicht mit auf dem Chip enthalten, je nach Aufwand braucht man dazu nochmal ein oder zwei ICs. Wie sowas geht, könnt Ihr in fast allen Artikeln nachlesen, die sich mit Hardware beschäftigen, ich habe mir das hier erspart, ICs mit dem Texteditor zu zeichnen ist etwas mühsam.

Bei Schreibzugriffen wird der D/A-Wandler aktiv und setzt die Daten, die er auf dem Bus findet, in eine entsprechende Ausgangsspannung um. Bei einem Lesezugriff wird die A/D-Wandlung gestartet, wobei der Chip den Prozessor über die Busy-Leitung solange anhält, bis er fertig ist. Bei einem BASIC-Programm ist aber schon die Berarbeitungszeit der Befehle so lang, daß man davon mit Sicherheit nichts merkt. Auch auf einem 4MHz-Z80 dauert der kürzeste Maschinenbefehl schon 1us, das als Vergleich. Da lohnt es sich noch nicht mal, die Wandlung zu starten und dann auf einen Interrupt zu warten, während der Rechner etwas anderes macht, das kostet durch den Verwaltungsaufwand viel mehr Zeit. Bei einem 486er mit 33MHz mag das anders aussehen, aber von sowas reden wir hier ja garnicht.

Bei dem Chip von einer nenneswerten Stromaufnahme zu reden ist schon fast lächerlich, gerade mal 12mA braucht er und das kann man wohl jedem Netzteil zusätzlich zumuten. Für den Preis gilt ähnliches, rund 21 Mark sind für diesen Chip sicher nicht zuviel.

Apropos Netzteil, damit kann man z.B. ein billiges programmierbares Netzteil äußerst einfach aufbauen. Über den D/A-Ausgang steuert der Rechner die Leistungstransistoren direkt an und mit dem A/D-Eingang lauert er, ob an den Klemmen die gewünschte Spannung tatsächlich erreicht wird. Natürlich ist die Regelung etwas träger als eine mit einem integrierten Spannungsregler oder Operationsverstärker, aber wenn die Laständerung nicht zu sprunghaft ist, dürfte es durchaus genügen. Im Gegensatz zu "normalen" Netzteilen, deren Ausgangsspannung immer mehr oder weniger stark lastabhängig ist, kann der Rechner die Spannung dank der Messung (natürlich im Rahmen der 8 Bit) genau auf dem eingestellten Wert halten. Wer sich für (digitale) Regelungstechnik interessiert, könnte hier prima die Wirkungsweise von verschiedenen Regelungsalgorithmen ausprobieren. Auch zum rechnergestützten Aufnehmen von Kennlinien eignet sich sowas gut, denn da braucht man sowieso immer beides, sowohl eine Spannungs- oder Stromquelle als auch ein Meßgerät.

Alexander Schmid



Club 80 Börse --- Club 80 Börse

Genie IIIs, 256 kB RAM, 1,77/7,2 MHz, 2x Floppy 80/ds/dd (bisher ohne Helmut'sche Umbauten)

Star Gemini 10x 9-Nadel-Drucker (Epson-Emulation), Centronics-Interface, HRG- und download-fähig, NLQ, Traktor

Zusammen VB 600,- DM, einzeln 550,- DM (Computer) und 150,- DM (Drucker)

Anfragen an Arnulf Sopp, Katharinenstr. 27, 2400 Lübeck 1 (neue Telefon- und Fax-Nummer nach Umzug noch nicht bekannt).

"Wegen Geschäftsaufgabe"

verschenke bzw. verkaufe ich folgende Literatur für TRS80/GENIE:

Maschinencode und besseres BASIC (Stewart)	DM 10.--
Machine Language Disk I/O (Wagner)	20.--
Maschinensprachebuch für TRS80 und GENIE (Röckrath)	-----
Z80 Assemblersprache (deutsches Benutzerhandbuch)	20.--
TRS80/Z80 Assembly Language Library (Lindley) Original (NP DM 145)	45.--
How to program the Z80 (R.Zaks) Original (NP 50.--)	25.--
Disassembled Handbook for TRS80	-----
TRSDOS & DISK BASIC Reference Manual	15.--
Erfolg mit VISICALC (Hergert) Original (NP ca.40.--)	18.--
BASIC FASTER & BETTER (Rosenfelder) - viele nützliche Routinen in BASIC und Maschinensprache	30.--
AIDS III V.1.5 - Handbuch für eine sehr schnelle Datenbank	-----

Versand unfrei oder gegen Portoerstattung (in Briefmarken).

*K.-J.Mühlenbein, Keplerstr.13 * 6940 Weinheim * Tel 06201/183640*

Suche für NewDos/GDos:

Codecen.CMD & Optimice.CMD
aus ALCOR-Pascal-Compiler-System

für Grape21:
von M. Winter
Treiberprogramm für Genie IIIs

Volker Dose 04343/1357

Der Rechner mit der Handbremse

Da neben den Tandy's immer mehr "Fremdrechner" den Weg zu uns finden, möchte ich denen auch mal einen Artikel widmen. Die CPC's waren ja schon immer etwas anders als andere Rechner und so ist es auch beim Timing der CPU. Damit der Z80 dem Video-Controller nicht ins Gehege kommt, bekommt er sog. WAIT-Zyklen aufgebrummt, was zu ca. 20% Verlust an Rechenleistung führt; als effektive Taktfrequenz bleiben rund 3.2MHz übrig. Wenn man für Zeitschleifen u.ä. aber die genaue Zahl der Taktzyklen braucht, hilft ein Blick ins Prozessorhandbuch leider nicht weiter, weil die unterschiedlichen Befehle von den Wait-Zyklen unterschiedlich beeinflußt werden und die Zahl der Takte somit mehr oder weniger stark von einem ungebremsten Z80 abweicht. In der c't 9/85 hat sich mal jemand die Mühe gemacht und diese Zeiten bestimmt. Hier nun das Ergebnis, vielleicht kann es der ein oder andere ja brauchen. Die Zahl der Taktzyklen ist angegeben als "ohne Bremse (=anderer Rechner) / mit angezogener Handbremse (=CPC)". Durch dieses merkwürdige System beträgt die Ausführungszeit eines Befehls immer ein vielfaches von 4 Takten und bei längeren Befehlen kommt meistens noch eine "Ehrenrunde" dazu.

***** 8-Bit Ladebefehle *****

Befehl	Takte
LD r,r'	4 / 4
LD r,n	7 / 8
LD r,(HL)	7 / 8
LD r,(IX/IY+d)	19 / 20
LD (HL),r	7 / 8
LD (IX/IY+d),r	19 / 20
LD (HL),n	10 / 12
LD (IX/IY+d),n	19 / 24
LD A,(ss)	7 / 8
LD A,(nn)	13 / 16
LD (ss),A	7 / 8
LD (nn),A	13 / 16
LD A,I/R	9 / 12
LD I/R,A	9 / 12

***** 16-Bit Ladebefehle *****

Befehl	Takte
LD dd,nn	10 / 12
LD IX/IY,nn	14 / 16
LD (HL),nn	16 / 20
LD dd,(nn)	20 / 24
LD IX/IY,(nn)	20 / 24
LD (nn),HL	16 / 20
LD (nn),dd	20 / 24
LD (nn),IX/IY	20 / 24
LD SP,HL	6 / 8
LD SP,IX/IY	10 / 12
PUSH qq	11 / 16
PUSH IX/IY	15 / 20
POP qq	10 / 12
POP IX/IY	14 / 16

***** 8-Bit Arithmetik *****

Befehl	Takte
ADD/ADC/SUB/SBC/ AND/OR /XOR/CP	
-- r --	4 / 4
-- n --	7 / 8
-- (HL) --	7 / 8
-- (IX/IY+d) --	19 / 20
INC/DEC	
-- r --	4 / 4
-- (HL) --	11 / 12
-- (IX/IY+d) --	23 / 24

***** 16-Bit Arithmetik *****

Befehl	Takte
ADD HL,dd	11 / 12
ADC HL,dd	15 / 16
SBC HL,dd	15 / 16
ADD IX/IY,dd	15 / 16
INC/DEC dd	6 / 8
INC/DEC IX/IY	10 / 12

***** Exchange, Blocktransfer und Suchbefehle *****

Befehl	Takte
EX DE,HL	4 / 4
EX AF,AF'	4 / 4
EXX	4 / 4
EX (SP),HL	19 / 24
EX (SP),IX/IY	23 / 24
LDI/LDD	16 / 20
LDIR/LDDR	
-- B<>0 --	21 / 24
-- B=0 --	16 / 20
CPI/CPD	16 / 16
CPIR/CPDR	
-- B<>0 --	21 / 24
-- B=0 --	16 / 16

***** Rotieren und Verschieben *****

Befehl	Takte
RLCA/RLA/RRCA/RRA	4 / 4
RL /RLC/RR /RRC/ SLA/SRA/SRL	
-- r --	8 / 8
-- (HL) --	15 / 16
-- (IX/IY+d) --	23 / 28
RLD/RRD	18 / 20

***** Sprünge *****

Befehl	Takte
JP nn / JP cc,nn	10 / 12
JR e / JR cc,e	
-- cc wahr --	12 / 12
-- cc falsch --	7 / 8
JP (HL)	4 / 4
JP (IX/IY)	8 / 8
DJNZ e	
-- cc wahr --	13 / 16
-- cc falsch --	8 / 12

***** CALL und RETURN *****

Befehl	Takte
CALL nn / CALL cc,nn	
-- cc wahr --	17 / 20
-- cc falsch --	10 / 12
RET	10 / 12
RET cc	
-- cc wahr --	11 / 16
-- cc falsch --	5 / 8
RETI / RETN	14 / 16
RST p	11 / 16

***** INPUT / OUTPUT *****

Befehl	Takte
IN / OUT A,(n)	11 / 12
IN / OUT r,(C)	12 / 16
INI / IND	16 / 20
INIR / INDR	
-- B<>0 --	21 / *
-- B=0 --	16 / *
OUTI / OUTD	16 / 20
OTIR / OTDR	
-- B<>0 --	21 / *
-- B=0 --	16 / *

***** Bit-Verarbeitung *****

Befehl	Takte
BIT	
-- b,r --	8 / 8
-- b,(HL) --	12 / 12
-- b,(IX/IY+d) --	20 / 28
SET/RES	
-- b,r --	8 / 8
-- b,(HL) --	15 / 16
-- b,(IX/IY+d) --	23 / 28

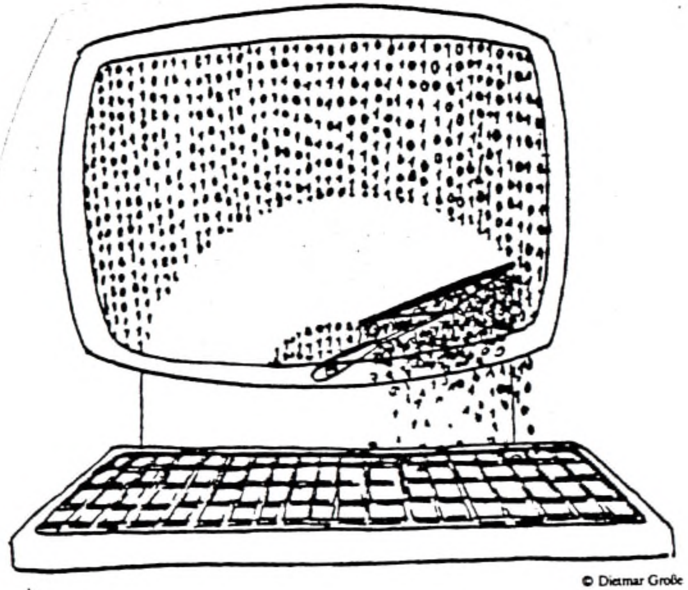
```
***** sonstige Befehle *****

Befehl                        Takte

DAA / CPL / CCF / SCF /
NOP / HALT/  EI / DI          4 / 4

NEG / IM 0/1/2                8 / 8

-------------------------------------------------------------

r,r': A,B,C,D,E,H,L           b  : Bitnummer
dd  : BC,DE,HL,SP             p  : Restartnummer
ss  : BC,DE,HL                d  : 8-Bit-Offset
qq  : AF,BC,DE,HL             cc : Bedingung
n   : Byte oder Portadresse   e  : relative 8-Bit-Adresse
nn  : 16-Bit-Adresse

*) diese Befehle können beim Schneider nicht verwendet werden
```

Alexander Schmid

Schönere Schildchen

Wenn man seinem Rechner eine neue Schnittstelle gönnt oder sonstige Veränderungen vornimmt, kommt man immer wieder zu dem Punkt, wo es darum geht, der Öffentlichkeit kundzutun, wozu dieser und jener Stecker dient. Handgeschriebene Aufkleber haben leider den Nachteil eben genauso auszusehen. Neidvoll blickt man immer wieder auf die professionellen Beschriftungen "ordentlicher" Rechner, aber es gibt einen recht einfachen Weg, das auch selber hinzukriegen. Man nehme selbstklebende Etiketten, wie es sie für fast jeden Drucker gibt und spanne sie in denselben ein. Danach drucke man den gewünschten Text möglichst fett und dunkel, damit die Schrift nicht so nach Nadeldrucker aussieht. Das Wichtigste kommt jetzt: damit nichts mehr verwischen oder verkratzen kann nehme man einen Streifen Tesa o.ä. und klebe ihn auf das Etikett ! Mit dem Fingernagel ordentlich festreiben und anschließend kann man das Schildchen mit der Schere in Form bringen und auf den Compi kleben, wenn nötig mit Uhu. Das Ergebnis kann sich durchaus sehen lassen und man braucht seinen Rechner nicht mehr schamvoll zu verstecken. Aufreibebuchstaben sehen zwar auch gut aus, aber um damit eine gerade Linie hinzukriegen muß man schon viel üben, besonders wenn einem Stecker und Knöpfe im Weg sind und billiger ist das sicher nicht. Ganz abgesehen davon, daß diese Zeichen recht empfindlich sind und man sie für den rauhen Alltagsbetrieb am besten mit Klarlack schützen sollte.

Alexander Schmid



© Dietmar Große

Hacker's Hirnkrücke

Ihr werdet Euch sicher wundern, was ein Taschenrechner in dieser Postille macht, aber wenn man PC als "Pocket Calculator" interpretiert, kommt's doch hin. Es geht einfach darum, daß ich endlich einen brauchbaren Hardware-Sidekick gefunden habe, der neben allen von mir benötigten Funktionen auch in Hex rechnen kann. Besagter Rechner ist der CASIO fx-115D und zählt sich zu den sog. wissenschaftlichen Rechnern, d.h. er hat alle möglichen trigonometrischen Funktionen und was man in der höheren Mathematik sonst noch so braucht. Dazu hat er natürlich statistische Funktionen und kann sogar Permutationen und Kombinationen berechnen (wer nicht weiß, was das ist, braucht es auch nicht, wer damit aber schon mal zu tun hatte, weiß, was das für ein Aufwand ist). Daß er mit komplexen Zahlen umgehen kann, ist schon nicht mehr so selbstverständlich und für den Programmierer schließlich bietet er beliebige Umrechnungen zwischen hex, oktal, binär und dezimal. Als logische Funktionen hat er NEG (2er-Komplement), NOT (1er-Komplement), AND, OR, XOR und XNOR, dazu kommen selbstverständlich noch die artihmetischen Funktionen (+, -, *, /) in allen Zahlensystemen (was war doch gleich 123h * 4...). Mehr kann man wohl nicht verlangen. Ansonsten ist es ein ganz normaler Taschenrechner, wie es ihn sicher auch von anderen Firmen gibt, fragt sich nur, ob man dort auch soviele Funktionen zusammen hat. Programmierbare Rechner von HP oder Texas waren mir bis jetzt immer eine Nummer zu groß und zu teuer, den CASIO gibt's schon für 35 Mark. Warum soll man mit Kanonen auf Spatzen schießen und mehrere hundert Mark ausgeben, etwas selber denken tut hin und wieder auch ganz gut.

Alexander Schmid

Exceptions, Interrupts und Traps

von Gerald Schröder

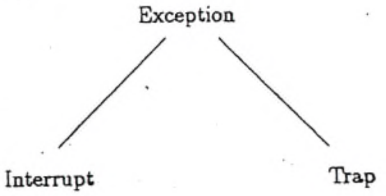
Dieser Artikel fällt wieder in das Themengebiet „Betriebssysteme", wie sollte es anders sein. Die drei obengenannten Begriffe sind auf vielen Ebenen zu finden, von der Maschinenebene (Mikroprozessor und Konsorten) bis zur Ebene der Programmiersprachen (PL/1 und Nachfolger). Die ganze Sache ist für ProgrammiererInnen interessant, die sich nahe an der Hardware tummeln, sei es in Assembler oder einer (höheren) Programmiersprache. Übrigens finden sich die Begriffe bei *allen* Prozessortypen (selbst bei den Intel-80x86), obwohl der Z80 natürlich alles am schönsten, besten, einfachsten macht...

Leider werden die Begriffe nicht einheitlich verwendet. Ich werde hier *meine* Auffassung der Begriffe darlegen, da sich die verschiedenen Auslegungen nicht unter einen Hut bringen lassen. Deshalb kann es sein, daß Euch die Begriffe in einer etwas anderen Verkleidung über den Weg laufen, aber erkennen und einordnen könnt Ihr sie dann hoffentlich trotzdem.

Ich werde zuerst die Begriffe so vorstellen, wie ich sie verstehe, und dann Einsatzbereiche nennen.

Exception

Eine Exception (dt.: Ausnahme) ist für mich entweder ein Interrupt oder ein Trap, also ein Oberbegriff:



Der Begriff „Exception" wird oft im Bereich von höheren Programmiersprachen gebraucht, z. B. bei Ada[1].

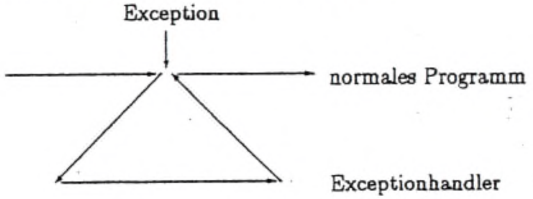
Eine Exception tritt auf, wenn irgendein *Ereignis* den normalen Programmfluß *unterbricht*. Denkbar sind z. B. folgende Ereignisse und Reaktionen:

[1] In C bzw. UNIX heißen Exceptions „Signale", sind aber ähnlich.

Ereignis	mögliche Reaktion
ein unzulässiger Wert wurde eingegeben	Wiederholung der Eingabe
ein Sektor soll eingelesen werden	der Sektor wird gelesen
Division durch 0	Ausgabe einer Fehlermeldung
Überschreiten der Indexgrenzen eines Feldes	Erweiterung des Feldes
es wird auf einen Speicherbereich zugegriffen, der gerade ausgelagert ist	Einlagerung des Speicherbereichs
ein Zeichen ist an der seriellen Schnittstelle angekommen	das Zeichen wird abgeholt
beim Lesen eines Sektors von der Diskette oder Festplatte ist ein Fehler aufgetreten	das Lesen wird wiederholt und ggf. eine Fehlermeldung ausgegeben
ein Sektor wurde von der Diskette oder Festplatte in einen Buffer (dt.: Puffer, Zwischenspeicher) gelesen	der Sektorinhalt wird aus dem Buffer abgeholt
ein Timer (dt.: Zähler) ist abgelaufen	ein Weckton wird ausgegeben

Alle diese Ereignisse finden während des *normalen* Programmablaufs statt, sollen aber *sofort* bearbeitet werden. Also wird das Programm unterbrochen und ein Unterprogramm aufgerufen, das sich um die *Behandlung der Exception* kümmert; dieses Unterprogamm nennen wir **Exceptionhandler**. Nach der Behandlung der Exception wird das Programm an der Stelle fortgesetzt, an der es unterbrochen wurde.

Der Programmfluß sieht grafisch folgendermaßen aus:



Interrupt

Ein Interrupt (dt.: Unterbrechung) liegt vor, wenn ein Gerät bzw. ein Controller die Arbeit des Prozessors unterbricht. Ein Interrupt wird also *extern* verursacht und dann *intern* (im Prozessor) behandelt.[2] Die Behandlung wird von einem **Interrupt-**

[2] Heute finden wir *intern* in einem Mikroprozessor auch Einheiten vor, die früher *extern* (als eigene Chips) realisiert wurden, wie serielle Schnittstellen, DMA-Kanäle (DMA: „Direct Memory Access", dt.: direkter Speicherzugriff) oder Timer. *Logisch* sind diese Einheiten aber weiterhin *externe* Einheiten, denn der Prozessor würde auch ohne sie funktionieren. Sprich: Interne Einheiten sind solche, ohne die ein Prozessor nicht funktioniert (kein Programm abarbeiten kann).

handler (oft auch „Interruptroutine“ genannt) erledigt. (Siehe dazu auch Info 31, Artikel „Schnittstellenprogrammierung“.)

Trap

Bei einem Interrupt ist ein *externes* Ereignis dafür verantwortlich, daß der Programmfluß unterbrochen wird. Wenn ein *internes* Ereignis die Ursache für eine Unterbrechung ist, nennen wir die Unterbrechung **Trap** (dt.: Falle). Für die Behandlung ist ein **Traphandler** zuständig.

Interne Ereignisse werden durch einen *Befehl* direkt oder indirekt ausgelöst. Eine *direkte* Auslösung ist ein spezieller Trap-Befehl, der dann wie ein Unterprogrammaufruf `CALL xxxx` wirkt, wobei xxxx die Adresse des Traphandlers ist. *Indirekt* wird ein Trap ausgelöst, wenn dies bei der Ausführung eines Befehls geschieht (der kein Trap-Befehl ist). Beispiele:

- Beim Versuch, durch 0 zu teilen (was nicht möglich ist), wird ein Trap „Division durch 0“ausgelöst, was die Ausgabe einer Fehlermeldung zur Folge haben kann.
- Falls ein unbekannter Befehl ausgeführt werden soll, wird ein Trap „Unbekannter Befehl“ ausgelöst, so daß der Traphandler diesen Befehl evtl. emulieren kann (siehe Z80-Illegals, Info 17).

Einsatzbereiche

Wo lassen sich Exceptions, also Interrupts und Traps, sinnvoll einsetzen?

Interrupts

Die Einsatzbereiche für Interrupts wurden schon im Artikel „Schnittstellenprogrammierung“ in Info Nr. 31 beschrieben: Wenn der Prozessor „ziemlich lange“ warten muß, weil ein Gerät oder Controller zu *langsam* ist (z. B. Drucker), oder wenn sich nicht voraussehen läßt, *wann* etwas passiert (z. B. Empfang eines Zeichens an der seriellen Schnittstelle), lassen sich Interrupts einsetzen. In der zwischen den Interrupts liegenden Zeit kann der Prozessor einen Haufen „nützlicher“ Arbeit verrichten, statt in Warteschleifen zu kreisen.

Traps

Trap-Befehle bieten sich als Schnittstelle zwischen Anwenderprogramm und Betriebssystem an. Im Gegensatz zu einem normalen Unterprogrammaufruf per

`CALL xxxx` sind Traps meistens schneller und nicht von einer bestimmten Adresse xxxx abhängig, also änderungsfreundlicher. Andere Traps erhöhen vor allem den Komfort und die Sicherheit von Programmen, denn es muß z. B. nicht vor jeder Division `x DIV y` geprüft werden, ob der Divisor y = 0 ist.

Weitere Vorteile

Da die Aufrufe der Exception- (Interrupt-, Trap-) Handler meistens nicht direkt, sondern über eine Tabelle erfolgen, in der die Adressen der Handler eingetragen sind, lassen sich leicht neue Handler für einen Interrupt oder Trap installieren: In der Tabelle wird die Adresse des alten Handlers durch die Adresse des neuen Handlers ersetzt. Wenn die Tabelle groß genug ist, lassen sich auch leicht neue Exceptions einführen.

Alles klar?

Ich hoffe, daß Ihr mit mir der Meinung seid, daß Exceptions eine tolle Sache sind und viel mehr ausgenutzt werden sollten.

Ich muß an dieser Stelle nochmal darauf hinweisen, daß die Begriffsbildung nicht einheitlich ist. Bei den 680x0 werden auch Interrupts als „Traps“ bezeichnet, bei den 80x86 ist es umgekehrt (der Befehl `INT` ist ein Trap, kein „Interrupt“). Nachdem ich jetzt die Sache ein für allemal klargestellt habe, sollten sich so unbedeutende Firmen wie Motorola (680x0) und Intel (80x86) gefälligst an meine Begriffe halten, meine ich!

Vom Single zum Multi

Gerald Schröder

Um was geht's?

Neuerdings taucht in viele Artikeln, Anzeigen und Gesprächen das Schlagwort „Multi" auf: Multitasking, Multiprocessor, Multiprocessing, Multiuser. Diese Begriffe werden uns alten CP/Mlern an den Kopf geworfen, mit dem Vorwurf, ein „Singleprocessor, Singleuser, Singletasking System" sei doch wohl nicht mehr zeitgemäß.

Ich möchte hier versuchen, die Brücke vom alten „Single" zum modernen „Multi" zu schlagen. Dazu werde ich (bei ausreichender Ausdauer) einige Artikel zu diesem Thema schreiben. Anfangen möchte ich hier mit einer Einführung in die obengenannten neuen Begriffe. Entferntes Ziel dieser Serie ist es, den Kern eines modernen Betriebssystems auf Basis des Z280 zu entwerfen. Ich hoffe, daß der Weg dorthin Euch hilft, den ganzen Humbug zu durchschauen, damit Ihr aus vollem Herzen sagen könnt: „So ein Quatsch!" oder „Stark! Super! Genial!", je nach Geschmack.

Damit sowohl die englisch- als auch die deutschsprachigen Leser zu ihrem Recht kommen, habe ich ein Glossar englischer Begriffe, die in diesem Text auftauchen, erstellt und als Anhang beigelegt. Es läßt sich darüber streiten, ob eher die englischen Originale, die „Eindeutschungen" oder die Übersetzungen benutzt werden sollen.

Für die Wissensdurstigen habe ich als zweiten Anhang einige Literaturhinweise verfaßt, die allerdings nur einen minimalen Ausschnitt aus der unübersehbaren Masse an Büchern zu diesem Thema darstellen.

Prozeß und Prozessor

Was ein Prozessor ist, weiß hoffentlich jeder. Sagen wir einfach: Ein Prozessor ist ein Ding, das Befehle ausführt und Daten bearbeitet. Aber was ist ein Prozeß? Eine mögliche Antwort ist: Ein Prozeß ist ein Programm, wie wir es alle kennen. Genauer ist folgende Definition:

Ein Prozeß besteht aus drei Teilen:

1. Einer Folge von Befehlen.
2. Einem Satz von Daten.
3. Einem inneren und einem äußeren Zustand.

Was Befehle und was Daten sind, dürfte klar sein. Für uns ist nur interessant, daß der Prozessor die Befehle ausführt und mit den Daten arbeitet, insbesondere sie ändert.[1]

Der *innere* Zustand beschreibt den *Abarbeitungsstand* des Prozesses:

- Zustand der Daten.
- Befehlszeiger (Zeiger auf den nächsten auszuführenden Befehl).

Von *außen* gesehen, können wir auch mehrere Zustände eines Prozesses unterscheiden, z. B.:

inaktiv: Der Prozeß wurde noch nicht gestartet.

laufend: Der Prozeß läuft gerade.

wartend: Der Prozeß hat sich selbst in einen Wartezustand versetzt (und wartet auf ein Signal von außen).

unterbrochen: Der Prozeß wurde von außen unterbrochen (und wartet auf seine Fortsetzung).

Damit ein Prozeß laufen kann, muß er einen Prozessor zugeteilt bekommen, der die Befehle des Prozesses ausführt. Dadurch wiederum wird der *innere* Zustand verändert, denn erstens ändern die Befehle normalerweise die Daten und zweitens zeigt nach Abarbeitung eines Befehls der Befehlszeiger auf den nächsten Befehl.

Multitasking bzw. Multiprocessing

Der Normalzustand, den wir alle aus der täglichen Praxis kennen, ist *ein* Prozeß auf *einem* Prozessor. Sobald wir es mit *mehreren* Prozessen zu tun haben, gibt es mehrere Möglichkeiten der Ausführung:

1. Wir lassen die Prozesse *nacheinander* auf *einem* Prozessor laufen. Diese Möglichkeit ist trivial und interessiert uns nicht weiter.
2. Wir lassen die Prozesse *konkurrent* (bzw. nebenläufig) auf *einem* oder *mehreren* Prozessoren ablaufen. Zwei Prozesse sind konkurrent, wenn zu einem Zeitpunkt beide begonnen, aber keiner abgeschlossen wurde [Gil81, Seite 49]. Nun gibt es wieder zwei Unterklassen:
   (a) Bei *einem* Prozessor ist zu einem Zeitpunkt immer genau *ein* Prozeß aktiv, während alle anderen entweder unterbrochen sind oder freiwillig warten. Ich nenne dies *quasi parallel*.

[1] Befehle dürfen normalerweise nicht geändert werden. Selbstmodifizierender Code ist also tabu.

(b) Bei *mehreren* Prozessoren können *mehrere* Prozesse gleichzeitig aktiv sein. Zwei Prozesse sind *echt parallel*, wenn zu einem Zeitpunkt beide Prozesse aktiv sind.

Was ist genau der Unterschied zwischen *quasi parallel* und *echt parallel*? Nehmen wir an, wir haben es mit zwei Prozessen zu tun, die ausgeführt werden sollen. Im *quasi parallelen* Fall werden einige Befehle des ersten Prozesses ausgeführt und dieser dann von außen unterbrochen. Nun werden einige Befehle des zweiten Prozesses ausgeführt, bis dieser unterbrochen wird. Jetzt kommt wieder Prozeß 1 an die Reihe usw. Im *echt parallelen* Fall dagegen werden die Befehle eines Prozesses auf einem Prozessor und die Befehle des anderen Prozesses *gleichzeitig* auf einem anderen Prozessor ausgeführt.

Das quasi oder echt parallele Ausführen von Prozessen nennt sich Multitasking oder Multiprocessing. Interessanterweise bieten die meisten modernen Betriebssysteme (z. B. UNIX, OS/2) Multitasking an, aber nur auf *einem* Prozessor. Mit *mehreren* Prozessoren kann kaum ein Betriebssystem umgehen. Hier gibt es noch viel Entwicklungsarbeit zu tun.

Threads

In letzter Zeit tauchen neuerdings Threads auf. Threads sind eine Abart der Prozesse. Bei Prozessen gibt es das Problem, daß (normalerweise) jeder Prozeß für sich abgeschlossen ist, d. h. zwei Prozesse sich nicht dieselben Befehle oder Daten teilen dürfen. Wenn zwei Prozesse Daten austauschen oder mit denselben Daten arbeiten wollen, müssen dafür besondere Maßnahmen getroffen werden, indem zum Beispiel ein Prozeß einem anderen Daten schickt (was wiederum Zeit kostet).

Bei Threads ist dies genau umgekehrt: Threads teilen sich sowohl Befehle als auch Daten. Allerdings benutzt jeder Thread einen eigenen Stack. Außerdem besitzt jeder Thread seinen eigenen Zustand, wobei der Zustand der Daten für alle Threads gleich ist.

Ansonsten gelten dieselben Aussagen wie für Prozesse: Auch Threads können konkurrent ausgeführt werden, wobei wir wieder zwischen der quasi und der echt parallelen Ausführung unterscheiden.

Threads sind normalerweise den Prozessen untergeordnet. D. h. ein Prozeß kann aus mehreren Threads bestehen, aber nicht umgekehrt. Es bietet sich zum Beispiel an, mehrere Prozesse auf mehreren Prozessoren echt parallel ablaufen zu lassen. Diese Prozesse können dann wiederum aus mehreren Threads bestehen, die quasi parallel auf dem einen Prozessor laufen.

Multiprozessorsysteme

Bisher haben wir uns nur mit der Software-Seite beschäftigt. Und was gibt es auf der Hardware-Seite? Wir kennen alle unsere Rechner mit *einem Hauptprozessor*.

Neu und interessant sind Rechner mit *mehreren Prozessoren*, Multiprozessorsysteme genannt.

Dazu wieder einige Klarstellungen. Ein Prozessor eines Multiprozessorsystems sollte auch *allein* in einem Rechner arbeiten können, also macht ein Arithmetik-Coprozessor oder ein Grafikprozessor noch kein Multiprozessorsystem. Es gibt aber höchst unterschiedliche Multiprozessorsysteme, die wir nach folgenden Merkmalen klassifizieren können [Gil81, Kapitel 10]:

Kopplung:

- **starke:** Die Prozessoren sind über einen *gemeinsamen Speicher* (Shared Memory) verbunden.
- **schwache:** Die Prozessoren können nur Nachrichten (Messages) austauschen, z. B. über einen Bus.

Kontrolle:

- **zentral:** Ein Prozeß übernimmt die Kontrolle über alle anderen Prozesse und die Prozessoren.
- **verteilt:** Kein Prozeß hat die Kontrolle.

Hardware:

- **homogen:** Alle Prozessoren sind gleich, z. B. nur Z280-Prozessoren.
- **inhomogen:** Unterschiedliche Prozessoren, z. B. Z280 neben 80386.

Symmetrie:

- **symmetrisch:** Alle Prozessoren spielen dieselbe Rolle, können also jede Funktion im System übernehmen.
- **asymmetrisch:** Die Prozessoren haben verschiedene Rollen, z. B. kümmert sich ein Prozessor um die Ausgabe auf Drucker, während ein anderer die Ein-/Ausgabe von/auf Festplatten erledigt.

Daraus kann sich nun jeder sein Multiprozessorsystem zusammenbasteln (Gruß an Helmut), z. B. ein System,

- in dem mehrere Z280 werkeln,
- die alle einen eigenen (lokalen) Speicher besitzen und Nachrichten über eine serielle Schnittstelle austauschen,

- wobei alle Prozessoren dieselbe Rolle spielen (also alle Zugriff auf Drucker, Tastatur, Bildschirm usw. haben) und
- ein Prozeß (auf irgendeinem Prozessor) alles kontrolliert,

nennt sich ein „homogenes, schwach gekoppeltes, symmetrisches Multiprozessorsystem mit zentraler Kontrolle".

Multiuser

Der Schritt vom mehreren Prozessen zu mehreren Benutzern ist nicht mehr groß. Im normalen Fall sitzt ein Benutzer (User) vor dem Rechner. Es gibt dann meistens einen Prozeß, der Tastatur und Bildschirm dieses Benutzers überwacht. Wenn wir nun noch eine Tastatur und einen Bildschirm an den Rechner anhängen und einen zweiten Prozeß erzeugen, der diese kontrolliert, haben wir ein Multiusersystem. Es gibt zwei Prozesse, also haben wir es mit Multiprocessing tun.

Unser Multiusersystem kann nur einen Prozessor haben, was natürlich bedeutet, daß sich die Benutzer diesen *teilen* müssen, was den englischen Begriff „time sharing" geprägt hat (die *Zeit* des Prozessors wird *aufgeteilt*). Wenn mehrere Prozessoren vorhanden sind, können die Prozesse auf diese verteilt werden.

Schluß

Das soll für heute reichen. Wir haben jetzt die grundlegenden Begriffe im Zusammenhang mit „Multi..." kennengelernt. Sicherlich gibt es Fragen, Unklarheiten oder Wünsche für weitere Erklärungen. Schickt sie mir! Ich freue mich über jede Reaktion.

Anhänge

Glossar

Task (Aufgabe): Wie → Process.

Process (Prozeß): Ein Programm, bestehend aus Befehlen, Daten und einem (inneren und äußeren) Zustand.

Thread (Faden): Ein Teil eines Prozesses (→ Process), der mit anderen Threads Befehle und Daten des Prozesses teilt.

Multitaskingsystem : Wie → Multiprocessingsystem.

Multiprocessingsystem (Mehrprozeßsystem): Mehrere → Prozesse laufen auf einem Rechner.

Multiprocessorsystem (Mehrprozessorsystem): Mehrere Prozessoren in einem Rechner.

Multiusersystem (Mehrbenutzersystem): Mehrere Benutzer teilen sich einen Rechner.

Shared Memory (geteilter Speicher): Wie → Global Memory.

Global Memory (globaler Speicher): Speicher, auf den mehrere Prozessoren zugreifen können. Siehe auch → Local Memory.

Local Memory (lokaler Speicher): Speicher, auf den nur ein Prozessor zugreifen kann. Siehe auch → Global Memory.

Literatur

Wer sich näher mit diesem Thema befassen will, dem sei hier etwas Literatur empfohlen.

Von Seiten der Hardware werden die modernen Rechnerarchitekturen in [Gil81] beleuchtet. Das Buch ist 10 Jahre alt und dementsprechend sind einige der vorgestellten Rechner längst in der Rumpelkammer verschwunden. Trotzdem sind die grundlegenden Konzepte weiterhin gültig. Ich habe mich hier weitgehend an dieses Buch gehalten.

Bei den Einführungen in Betriebssysteme finden sich viele Bücher, die dieses Thema behandeln, allerdings auf einer sehr allgemeinen Ebene. Konkrete Implementierungen werden zumeist im Bereich von UNIX beschrieben, z. B. [ea].

Den theoretischen Hintergrund beleuchten Jessen und Valk in [JV87]. Sie entwickeln Modelle, mit denen Rechensysteme (Hardware und Software) beschrieben werden können, z. B. Petri-Netze. Außerdem werden Probleme und Lösungen in diesem Bereich besprochen, z. B. die Synchronisation von Prozessen oder Bedienstrategien, d. h. wie quasi-parallele Prozesse auf einem System (mit einem oder mehreren Prozessoren) abgearbeitet werden können. Das Buch kann ich nur Leuten mit soliden mathematischen Kenntnissen empfehlen.

Wer lieber alles zu den den klassischen Themen der Informatik aus einer Hand haben möchte, dem sei A. S. Tanenbaum empfohlen. In [Tan87] bzw. [Tan90a] befaßt er sich mit Betriebssystemen, sowohl allgemein als auch mit einer konkreten Implementierung (MINIX, ein UNIX-Clone, Quelltexte in C werden mitgeliefert). [Tan90b] beschreibt die Hardware von Rechnern näher, setzt aber auf einer tieferen Ebene an als das oben angeführte Buch von Giloi. [Tan88] widmet sich der Verbindung zwischen Rechnern auf den verschiedenen Ebenen. Neben MINIX hat Tanenbaum auch noch an einem verteilten Betriebssystem namens AMOEBA mitgearbeitet, zu dem meines Wissens leider bisher nur Arbeitsberichte erschienen sind.

Mit einer Empfehlung für besondere Zeitschriften kann ich nicht dienen. In fast jeder Zeitschrift findet sich etwas zu dem Thema. Einiges wird z. B. in dem Artikel

[BR91] angesprochen, der auch zeigt, daß dieses Thema noch aktueller Forschungsgegenstand ist.

Literatur

[BR91] T. Bemmerl und B. Ries. Obenliegend. Integrierter Einsatz von Parallelrechnern an der TU München. *iX*, 1991(7):74–79, Juli 1991.

[ea] S. J. Leffler et al. *The Design and Implementation of the 4.3 BSD UNIX Operating System.* ~~?, ?, ? Seiten.~~ Addison-Wesley, 1989. 471 Seiten.

[Gil81] W. K. Giloi. *Rechnerarchitektur.* Springer, 1981. 380 Seiten.

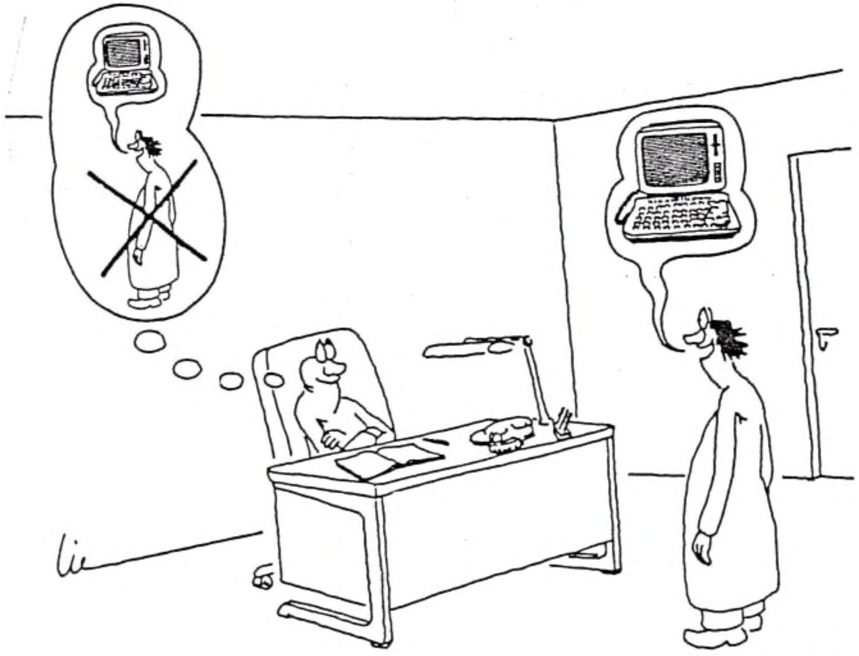
[JV87] E. Jessen und R. Valk. *Rechensysteme — Grundlagen der Modellbildung.* Springer, 1987. 560 Seiten.

[Tan87] A. S. Tanenbaum. *Operating Systems — Design and Implementation, 3rd Ed.* Prentice Hall, 1987. 719 Seiten.

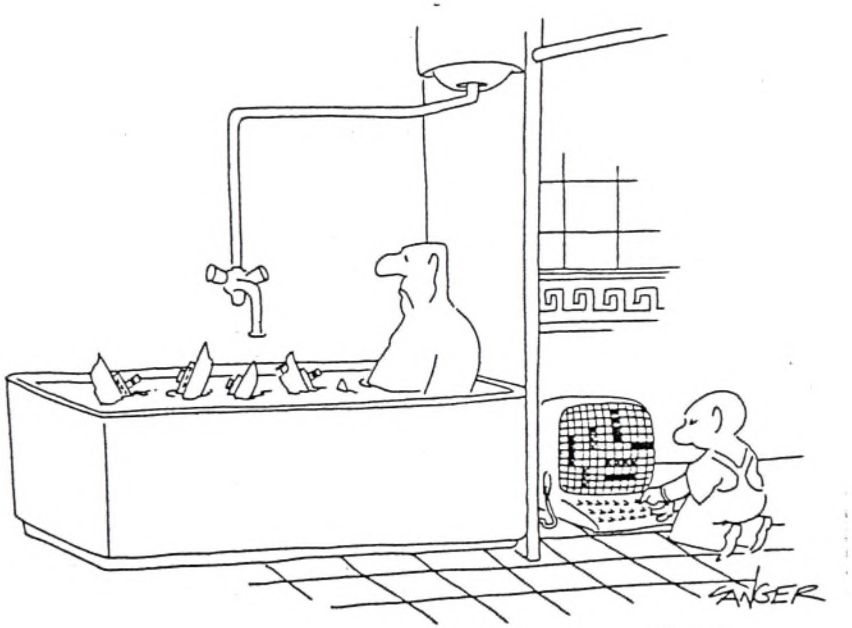
[Tan88] A. S. Tanenbaum. *Computer Networks, 2nd Ed.* Prentice Hall, 1988. 658 Seiten.

[Tan90a] A. S. Tanenbaum. *Betriebssysteme — Entwurf und Realisierung (2 Bände).* Hanser, 1990. 322 bzw. 268 Seiten.

[Tan90b] A. S. Tanenbaum. *Structured Computer Organization, 3rd Ed.* Prentice Hall, 1990. 587 Seiten.



© Erik Liebermann



© Heinz Langer

XT = Extra viel Trouble ???

Anfang 1990 hatte ich mich in die Idee verrannt, einen PC200 der Fa. "Sir Sinclair" mit einer Festplatte auszurüsten. Ein einige Zeit zuvor gelesener Testbericht über XT's (der PC200 war ebenfalls getestet worden und bekam eine akzeptable Kompatibilität bescheinigt) unterstützte mich in der Ansicht, daß das Unterfangen durchführbar sei. Also wurde beschlossen, eine 20MB-Platte nebst ST11M-Controller zu beschaffen und beides in der Kiste zu installieren.

Das Wörtchen "akzeptabel" hätte mich hellhörig machen sollen, denn was mit dem Kauf der Platte an Ereignissen losgetreten wurde, reicht aus, um einem den Spaß an der Computerei zu verderben. Da eine Stromversorgung der Platte aus dem Rechner nicht möglich war, mußte ein VERO-Schaltnetzteil herhalten (klappt übrigens prima). Nach Einsetzen der Karte, Anstöpseln der Stromversorgung und Anbringen aller Kabel konnte es losgehen. Rechner einschalten, Platte auf Drehzahl hochlaufen lassen und auf gehts in schnelle HD-Vergnügen ... Doch back to the roots ! Ohne Formatierung geht's nicht. Das Bios ist so schlau und läßt mir keine andere Wahl. Gemacht, getan !

Ab hier wird die Sache nun skuril. Mit Abschluß der sogenannten Low-Level-Formatierung verabschiedet sich das ganze Vergnügen mit der Meldung "Error: Faulty Memory (Parity-Error)". Von diesem letzten Statement ist die Kiste dann auch nicht mehr abzubringen. Jeder Power-On stoppt an dieser Stelle. Man ist ja Kummer gewohnt und hat gelernt, mit diesen Dingen zu leben.

Das nächste zu erreichende "Opfer" ist der Händler, der mir die Hardware verkauft hat. Auf mein Drängen hin erhalte ich von ihm die Platte für die sofortige Benutzung formatiert, partitioniert und mit dem DOS zum booten beschrieben. Erster Gedanke zu Hause vor dem Rechner: Jetzt muß ja alles bis zu DOS-Prompt in einem Rutsch laufen. Also Kontroller einstöpseln, Platten & Stromversorgung anschließen und alles einschalten. Der Selbsttest des Rechners läuft an und währenddessen leiert die Platte hoch. Die BIOS-Meldung kommt und ... "Error: Faulty Memory (Parity-Error)". Mittlerweile dämmert es mir, daß das Problem wohl größeren Kalibers ist, denn ziehe ich das Kabel mit den Steuersignalen ab - die Platte wird dann nicht gefunden - läuft alles sauber durch. Es hat den Anschein, als wenn der Rechner mit der Karte so recht nicht will.

Zu denken gibt nur, daß mir der Händler beide Teile in einem AT einwandfrei funktionierend vorgeführt hat. Der Gedanke ist also naheliegend, daß meinem XT nicht so kompatibel wie notwendig ist. Um das abtesten zu können, verabrede ich kurzerhand ein Treffen mit Helmut Bernhard, er besitzt ebenfalls einen XT. Bei ihm eingesteckt läuft alles so wie es soll. Ein vorübergehend ausgeliehener WD1006-Kontroller offenbart eine neue Fehlermeldung "...controller malfunction..." An ein ordnungsgemäßes Booten des Rechner ist nicht zu denken. Da nun insgesamt zwei Karten in meinem Unikum den Dienst verweigern, sehe ich mich zu einem Brief an den Hersteller genötigt. Da der in England seinen Sitz hat radebreche ich mein Problem in "bestem" Englisch zu Papier und schicke alles auf die Reise. Nach drei Wochen kommt dann endlich die Antwort: Man könne sich meine Schwierigkeiten nicht erklären, der Rechner sei vollkommen "kompatibel", müsse also mit der übliche

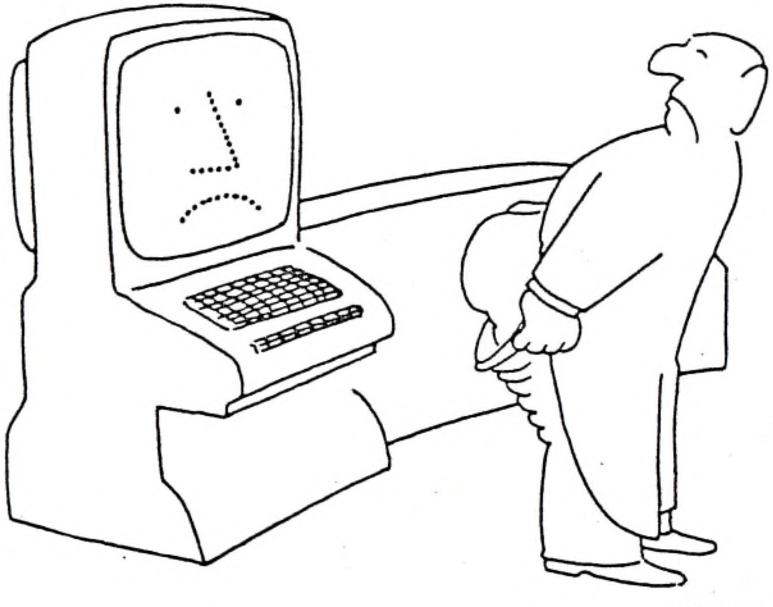
PC-Hardware laufen. Möglicherweise liegt ein Fehler im Netzteil vor (da die Platte extern versorgt wird, ist die Richtigkeit dieser Diagnose fraglich). Ich sollte mich an die deutsche AMSTARD-Vertretung zwecks Unterstützung wenden. Freundlicherweise liefert man im Antwortschreiben auch gleich die Anschrift und Telefonnummer mit. Ein entsprechendes Telefonat beschert mir eine reichlich niederschmetternde Antwort: Da der Rechner hier in Deutschland nicht verkauft wurde, könne man auch keinen Support gewähren, im übrigen hätte man auch keine Serviceunterlagen zur Hand und könne schon aus diesem Grunde nicht helfen (Gott sei Dank ist der Kunde König, nur in dieser Sache scheinbar ohne Amt und Würden !). Als Trostpflaster erhalte ich eine neue Nummer, wo mir angeblich weitergeholfen werden kann.

Bis heute ist es mir nur gelungen herauszubekommen, daß sich hinter dieser Nummer ein Techniker verbirgt, der solche Kisten wohl schon einmal repariert hat. Nur konnte ich ihn bis heute nicht an den Apparat bekommen.

Solchermaßen gesteßt, wurde das Problem mittlerweile recht elegant gelöst: bei HW-Elektronik in Hamburg habe ich mir alle Teile für einen 386SX gekauft und der XT zur Verschrottung freigeben.

Tschüß

Olaf



© Heinz Langer

* Keine Glosse, sondern Wirklichkeit...*

Es wäre übertrieben, nachfolgenden Einakter als "Drama" zu bezeichnen.
Aber ein Lustspiel ist es eigentlich auch nicht. Höchstens für den Außenstehenden.
Für den betroffenen Akteur hätte es eine Tragödie sein können; aber der ist solchen "Kummer" gewohnt.
Nennen wir es : einen traurigen (nicht tragischen) Sketch...

Ort der Handlung: Mannheim. Ein bekannter Supermarkt für elektronische Erzeugnisse: Nicht nur Computer, sondern auch Videogeräte, Fernseher, Radios, Kameras und und... Also lauter so'n technischer Kleinkram.
Zeit der Handlung: An der Schwelle des dritten Jahrtausend. Also weder in der Vergangenheit noch in der Zukunft. Nüchterne, unbarmherzige Gegenwart. Genauer gesagt: Mittagsstunde.
Bühnenbild: Riesenraum. Halb gedämpftes, weißliches Licht (kein Tageslicht, trotz Mittagsstunde). Viele Warengondeln, auf denen sich hochinteressante Technik im Kleinen türmt. Dazwischen Leute - Leute - teils mit suchenden Blicken, teils mit ernstem Blicke prüfend, teils gedankenvoll, teils -leer. Durchschnittsalter zwischen 10 bis 40, dazwischen Kinder (unter 10) und Greise (über 40). Überwiegend männlichen Geschlechts - ausgenommen bei den Fernsehern. Akustisch überwiegt Stille, gelegentlich durchdrungen von plötzlichen Schreien aus einer Ecke. Es ist die Ecke der CD-Player. Die Stimme hat KJ schon öfter (aus Versehen) gehört; nur kann er nie entscheiden, ob es Bryan Adams oder Adam Bryans ist. (Der Leser kennt sich da besser aus.)
Personen der Handlung:

1. Der Fachverkäufer in der Drucker-Abteilung (hier kurz "FV", sprich "EFF-AU!", genannt) ;
2. Der Kunde (hier kurz K.-J., sprich "KAJOT", genannt).

KAJOT, seit einem Jahrzehnt fast stolzer Besitzer eines EPSON RX80 F/T, also "nur Neunnadeldruckers", interessiert sich für einen möglichst EPSON-, auf jeden Fall aber IBM-kompatiblen Drucker mit 15 Nadeln mehr. (Wer mehr hat, is<s>t mehr!)

Vorhang auf!

KJ: *Guten Tag! Gestatten, ich möchte einen 24-Nadeldrucker!*
FV: *Sehr wohl, mein Herr! Wir haben da jede Menge!*
(Was schon schlechter ist. Anm. der Regie.)
FV (zeigt auf einen PANASONIC): *Was halten Sie z.B. von diesem hier? Der ist im Sonderangebot!*
KJ: *Was soll ich davon halten? Bitte, halten Sie mal meine Tasche.*
(KJ kramt darin herum und holt schließlich einen Fragebogen heraus, den er sich zuhause zusammengestellt hat.)
KJ: *Vielleicht halte ich viel davon, wenn er auch Proportionalschrift kann!*
FV: ? ? ?
KJ: *Ich meine: "Propro- äh - Popo- na, zum Gugguck, Sie machen mich ganz unsicher mit Ihrer Unsicherheit! Ich meine: P r o p o r t i o n a l- - - und dann kommt -schrift!*
FV: ? ? ? ?
KJ: *Ja also, das ist so: man denkt, jeder Buchstabe sei gleich breit. Is aba nich. I is dünner als O -*
FV (*unterbricht mit leuchtenden Augen*): *Aja, naklar: I heißt IN- und O heißt OUTPUT. Jaja, einen Output hat er. Sehen Sie mal ...*
(jetzt läßt FV doch tatsächlich eine ganze Seite ausdrucken, nachdem er elegant umständlich einen Bogen DIN A4 eingefädelt hat - was für KJ schon mal sehr wichtig ist, weil das beim EPSON glatter und somit schneller geht)

FV: *Ja, was meinten Sie doch?*
KJ: *'Proportionalschrift ist, wenn...' - - ja wenn jeder Buchstabe nur soviel Platz - (ach Gott, ich laß es lieber; hat wohl keinen Zweck) Sehen Sie hier im Ausdruck den Unterschied zwischen der Breite des i und des w ?*
FV: ??? ???
KJ: *Na schön. Dann sagen Sie mir bitte: Druckt der hier auch kursiv?*
FV: ??
KJ (*ich glaube, ich muß lauter sprechen*) *Ich meine: Schreibt der auch Italics?*
FV: ? ? ?
KJ: *Wissen Sie, das ist Schrägschrift.*
FV: *O ja, das tut er! Aber es ist doch besser, wenn der Drucker gerade steht. Eine kleine Wasserwaage bekommen Sie auch in unserem Hause.*
KJ: *Dankeschön! Die habe ich zu Hause. Ich stelle ihn also "gerade".*
("Vielleicht läuft sonst die Tinte aus..." denkt KaJot)
FV: ?
KJ: *Nun gut. Der Drucker scheint interessant. Besonders vom Preis her. Was kostet er doch nochmal?*
FV (*beflissen*): *hmhmhm Emmchen. Deutsche, versteht sich.*
KJ: *Sooo? In Ihrem Prospekt steht aber nur hmhm Emmchen. (Deutsche, versteht sich.)*
FJ: *Na ja, wenn Sie ihn kaufen, bekommen Sie ihn auch für hmhm Emmchen.*
KJ: *Ach so? Und was muß ich bezahlen, wenn ich ihn nicht kaufe??*
FV: ??

P A U S E

(Beide erholen sich. KaJot guckt in den Prospekt, der neben dem Drucker liegt. Ihm gehen einige Lichter auf. * * * Nach gebührender Warteschleife:)
KJ: *Nun eine letzte, nein, lieber eine allerletzte Frage: Im Prospekt steht nicht, ob der Drucker auch einen Download-Speicher besitzt. Das wissen Sie doch bestimmt!? O Gott - Sie sind ja plötzlich so bleich! Ist Ihnen schlecht? Soll ich einen Kollegen rufen?*
(Hier zieht der FV - sprich "Fachverkäufer" - kurz vor dem Zusammenbruch plötzlich ein kleines Ding aus der Rocktasche - KJ springt blitzschnell zur Seite, denn er kennt die rachsüchtige Unberechenbarkeit "technischer Kaufleute" und will Leben und Gesundheit noch etwas für seine Familie erhalten - - da spricht FV in das Ding leise hinein:
FV: *Otto! Otto bitte zu Nummer 7a!*
FV Otto kommt nach einigen bangen Minuten und wiederholter drahtloser Aufforderung.
KJ erblaßt vor soviel funktechnischer Sach- und Fachkenntnis und flüstert nur noch:
KJ: *Guten Tag Herr Otto! Bitte geben Sie mir diesen PANASONIC. Aber nicht diesen, sondern Original-verpackt, bitte!*
Otto: *Zahlen Sie bar oder mit Schech? Oder lieber auf Raten?*
KJ: *Ja was kostet der denn überhaupt?*
Otto: *Mein Herr, das ist ein Sonderangebot. Der kostet heute nur DM hm!*
KJ: *Soso; vorhin hätte ich fast hmhm bezahlt! Also gut. Packen Sie ein. Inzwischen gehe ich zahlen.*
Otto: ?
KJ: *Jaja - ratlos - - ich meinte: ratenlos...in "DÄHMARK"! Und vergessen Sie die Gebrauchsanleitung nicht, bitte!*
Otto geht ab. -

Gekauft. (Heißt soviel wie: " E N D E ")

Otto heißt Otto. Das versteh' ich. Weil man's auch von hinten lesen kann.
Was FV heißt, weiß ich aber nicht. Vielleicht: "Fröhlicher Vunker" ? ?

PS: Die Gebrauchsanleitung war wirklich drin! Ich bin sooo froh... Denn:
Ein Fachverkäufer ist versiert,
drum kaufe blind, was er serviert!
* * * * * * * * * * * * * * * * * * * *)

Public Domain Software

Hier ist natürlich CP/M als Betriebssystem gemeint und auch das Thema ist nicht gerade neu. Im Jahr 1985/86 veröffentlichte Ted Silveira in "Computer Currents, 2550 9th Street Berkley, CA" eine Serie von Artikeln. Diese Artikel habe ich für unsere Verhältnisse überarbeitet , beziehungsweise als Vorbild genommen. Die originalen Texte sind bei mir zu erhalten, falls jemand daran interessiert sein sollte. Eigentlich gab das Lesen der Texte von Ted Silveira den Anstoß für die mich. Meist nimmt man sein eigenes Wissen und die persönliche Erfahrung als selbstverständlich hin vergisst aber, daß es oft Neueinsteiger gibt und viele einfach nicht so gut mit der Fremdsprache Englisch zurechtkommen. Da ich auch eine Menge an PD-Software aus den USA erhalten habe und vor obigem Problem stand habe ich viele PD-Programme geteste bzw. jetzt im Gebrauch. Natürlich ist der Einsatz meiner 68 MB Festplatte hier von Vorteil und vereinfacht die Handhabung sehr.
Die Texte von Ted Silveira sind copyrighted und ich habe hier nicht die Texte übersetzt, diese wären dann doch nicht gerade zeitgerecht, sondern angeregt dadurch einige Artikel verfaßt, welche in den nächsten INFO's erscheinen.
Meine Artikel orientieren sich aber an der Vorlage und Beschreibungen von Programmen sind natürlich ähnlich.

Vor zwei Jahren kaufte ich ein Modem und hatte im Nahbereich eine Mailbox mit CP/M PD-Software. Natürlich stiegen meine Gebühren um das 5fache und oft war das empfangene Programm nicht die Zeit wert.
Die Qualitäten der Programme sind sehr verschieden und wenn ihr etwas Einblick in die MSDOS - PCSIG habt (10000 PD-Progamme), könnt ihr euch den Aufwandt, ein geeignetes Programm für den eigenen Bedarf zu finden, vorstellen.
Aber der Mensch (ich auch) ist natürlich neugierig und so verbrachte ich oft Stunden vor dem Monitor und einigen Kisten mit PD - Software.

Was ist Public Domain Software ?

Public Domain heißt auf deutsch öffentlicher Bereich, aber eine wörtliche Übersetzung ist hier wohl nicht ganz korrekt.
Einige Leute verbringen ihre Zeit am liebsten mit dem Schreiben von Programmen. Nicht weil sie dadurch Millionär werden wollen, sondern aus Spaß am konstruktiven Denken , weil sie in die Technik vernarrt sind oder eben jeder so seine kleinen Hobbies pflegt.
Wenn nun die Arbeit der "Programmierer" beendet ist werden die Programme der "Öffentlichkeit" anvertraut. Das heißt, jeder kann die Programme kostenlos bekommen und keine Lizenzbestimmungen müssen unterzeichnet werden. Man kann die Programme kopieren, ändern, weitergeben ,also machen was man möchte. Damit aber keiner sich mit fremden Federn schmückt haben einige Autoren dies nur unter Beibehalt ihres Copyrechtes gestattet, wie zum Beispiel bei *MEX* oder den Programmen zum *ZCPR*. Im allgemeinen sollte der Urheber der Programme sowieso immer erwähnt werden.

Ich bleibe mal bei dem Kürzel PD für Public Domain Programme.
Viele PD-Programme sind nützliche Hilfen und ich möchte einige nicht missen. Einige Programme sind nur etwas für Programmierer, aber die meisten sind für den normalen Anwender.

XDIR zum Beispiel listet das Direktorie alphabetisch sortiert, zeigt die Dateilängen und den freien Platz auf der Diskette an.
BAD oder *FINDBAD* testet die Diskette und/oder Festplatte auf Fehler, zeigt die defekten Sektoren an und erstellt eine Datei welche diese Sektoren belegt, damit nicht darauf zugegriffen wird. *NSWEEP*, eines meiner Favoriten, kopiert, listet, druckt und vereinfacht das Kopieren enorm.
Wenn auch die Hilfsprogramme den überwiegenden Teil der PD ausmachen, so gibt es Programme für fast alle Bereiche. Für Datenübertragung, Anwenderprogramme für Datenbanken, Textprogramme , Erweiterungen zum CP/M (mein Lieblingsthema) und und und...
Ich erspare mir hier ein paar Seiten und möchte eine oft gestellte Frage aufgreifen.

Wie gut ist Public Domain Software ?

Sicher wollen viele das wissen und PD-Programme können sehr gut sein, aber die meisten sind eben doch nur 'Eigenbau', was aber

bitte nicht abwertend aufgefaßt werden soll.
Die meisten Programme sind älter und CP/M oder auch NEWDOS sind halt in den Anfängen der Computertechnik entstanden. Aber wer kann ermessen, mit welcher Freude ich meinen Genie einschalte oder mit dem MORROW die diversen DFÜ-Programme ausprobiere ?
Letztendlich haben auch so manche PD-Programme die Richtung der Softwareentwicklung bestimmt (nicht nur unter CP/M) .
Hierzu gehören zum Beispiel *MDM7* , *MEX*, *ZMP*, im DFÜ-Bereich, *NSWEEP* oder *NULU* bei den Utilities und so großartige Sachen wie Richard Conn's ZCPR , oder die Autoinstallversionen *NZCOM* und *Z3PLUS* um nur einige zu nennen.

Alle PD-Programme haben aber eins gemeinsam, wenn es nicht klappt und die Arbeit der letzten Stunden verloren sind ist man alleine und fast immer auf sich selber angewiesen. Man kann Freunde fragen oder in Mailboxen um Hilfe bitten, aber es gibt keine Unterstützung und keine Hotline wie oft bei kommerziellen Produkten. Leider ist der Konkurenzkampf heute so groß, daß der Endanwender oft als Beta-Tester herhalten muß und dann ist es mit der Unterstützung durch den Softwareentwickler eh nicht weit her.

Wie bekommt man PUBLIC DOMAIN SOFTWARE ?

Einer der einfachsten und angenehmsten Arten, PD-Software zu erhalten ist, dies über Mailboxen zu machen. Leider haben viele von Euch kein Modem und selten gibt es Mailboxen, welche alte Software , wie ich die für Newdos und CP/M hier bezeichnen möchte, auf der Festplatte haben.
Man darf hier nicht vergessen, daß alte Rechner unser Hobby sind, und diese sind halt nicht mehr Up-To-Date. Also ,am besten erhaltet Ihr die gewünschte PD-Software über einen User-Club, durch Anfrage bei unseren Soft-Archivaren, oder von Gleichgesinnten.
Es gibt durchaus einige Clubs ähnlich wie unser , deren Adressen werde ich aber erst im nächsten Info bringen.
Ich selber installiere gerade eine ZNODE auf einem ALTOS 580 Rechner. Diese ZNODE wird als DEMO-Rechner bei mir ab Mitte Februar laufen. Der Rechner wurde mir von Herrn Helmut Jungkunz freundlicherweise zur Verfügung gestellt und so können sich Interessenten am Z-System selber ein Bild machen.
Danke Helmut!

ZNODE wird die Mailbox genannt , weil die Mailboxsoftware auf einem Z-80 Rechner nur mit den Utilities der CP/M Erweiterung NZCON oder ZCPR läuft und es halt meist um den Z80 geht.
Hierzu aber später mehr.

An PD-Software gibt es die Software der ursprünglichen USER-Groups , wie CP/M-UG, SIGM, PICONET und die deutsche Osborne User-Group, um nur einige zu nennen. Diese Gruppen haben sich im allgemeinen aufgelöst , oder sind zu moderneren Betriebssystemen gewechselt. Hierdurch entsteht oft das Problem, einen bestehenden alten Rechner in Gang zu bekommen, wenn keiner mehr weiss wie. Ich könnte euch da einige längere Geschichten von meinen Sammelstücken erzählen und konnte so manchem mit Hilfe der Clubkollegen zur Seite stehen.
Die Softwarelisten der User-Groups und neuer Software aus den USA sind bei mir , oder unseren Archivaren zu erhalten und es wird wohl kaum ein nicht lesbares Diskettenformat für uns geben.
In den nächsten Ausgaben des Club-Infos werde ich gute PD-Software vorstellen und zum Teil näher darauf eingehen. Da ich neben meinem MSDOS-Rechner fast nur mit CP/M arbeite sind die NEWDOS und GDOS-User hier leider etwas benachteiligt.

> Deshalb !
> Wer mit NEWDOS und/oder GDOS arbeitet möchte dem Club doch bitte seine Erfahrung mit der Soft- und Hardware mitteilen. Es gibt noch viele von uns, welche sich sehr aktiv mit diesen Rechnern befassen, unter anderem zu Steuerungszwecken oder als Wetterstation.

Genau ! IHR seid gemeint und nun trau EUCH mal zu schreiben.

Und was schreib ICH im nächsten INFO ?
Natürlich werde ich über DFÜ und die dazu nötigen Programme schreiben, vor allem da ich mich wegen der ZNODE gerade damit beschäftige.
Wie heißt es so schön

WER SCHREIBT - DER BLEIBT,

allerdings in einem anderen Zusammenhang, aber es könnte ja auch hier stimmen.

RCH

Impressum

1. Vorsitzender — Fritz Chwolka
Tel.: 0 24 64/89 20 — Saarstraße 34
5173 Aldenhoven

2. Vorsitzender — Gerald Schröder
Tel.: 0 41 05/26 02 — Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator — Andreas Magnus
Tel.: 02 09/87 02 30 — Bismarckstraße 29
4650 Gelsenkirchen

Newdos-Diskothekar — Oliver Volz
Tel.: 07 11/7 35 38 17 — Waldenburgstraße 73
7000 Stuttgart 80

CP/M-Diskothekar — Rüdiger Sörensen
Tel.: 0 61 31/32 08 60 — Thomas-Mann-Straße 3a
6500 Mainz 1

Clubbücherei — Kurt Müller
Tel.: 0 41 52/7 06 43 — Sophie-Scholl-Ring 3b
2054 Geesthacht

Redaktion — Jens Neueder
Tel.: 07 91/4 28 77 — Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 8870 Günzburg
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Schluß

Hallo Club 80'er,

wieder einmal haltet Ihr Euer neues Club-INFO in den Händen, und wieder einmal ist ein Jahreswechsel vorübergegangen. Den an dem Club-INFO mitwirkenden Kameraden möchte ich meinen Dank für die Mitarbeit aussprechen.

Gleichzeitig auch -im Nachhinein noch- meine allerbesten Wünsche für das jetzt begonnene Jahr an alle Club-Mitglieder. Hoffen wir, daß es im neuen Jahr mindestens so gut weitergeht wie im vergangenen Jahr.

Zum Abschluß dieses INFO's habe ich noch zwei Dinge, die mir am Herzen liegen:
Bitte sendet mir Euere Paper-Ware Artikel auf **weißem** Papier unter Verwendung eines frischen Farbbandes. Es ist für die Erstellung des INFO's wichtig, kontrastreiche Vorlagen zu haben.

Als Zweites, würde ich mich freuen, wenn sich eine BASIC-Anwender-Gruppe hier im Club finden würde. BASIC ist ja als Programmiersprache ein verbindendes Objekt zwischen all unseren Computern. Ich arbeite derzeit mit Quick-Basic 4.5 und PDS 7.1 als BASIC-Dialekte und bekenne mich als BASIC-Programmierer. Gibt es noch Gleichgesinnte?

Viel Spaß-
bis zum nächsten mal

Jens

Adressen-Liste CLUB80

Stand vom 12-02-1992
Seite 1

Nachname	Vorname	Straße	PLZ	Ort	Telefon privat	Telefon geschäftlich
Behrendt	Detlef	Schlosserbreite 1a	W- 8018	Grafing	08092 / 9173	
Benner	Achim	Vorm Mühlberg 1	W- 5910	Kreuztal 3	02732 / 3780	(ab 18.00Uhr)
Berndt	Wolfgang	Friedberger Straße 92c	W- 6360	Friedberg 2	06031 / 2963	
Berndt-Jochum	Ilse	Stachelsgut 24	W- 5060	Bergisch-Gladbach 1	02204 / 65254	
Bernhardt	Helmut	Hafenstraße 7	W- 2305	Heikendorf	0431 / 241907	0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	W- 8551	Hausen	09191 / 31698	09191 / 611108
Brans	Jörg	Tieloh 55	W- 2000	Hamburg 60	040 /6906531	
Braun	Harald	Postfach 8011	W- 2300	Kiel 17		
Böckling	Ulrich	Jochaczstraße 61	W- 5410	Hör-Grenzhausen	0264 / 4861	02631 / 895168
Chwolka	Fritz	Saarstraße 34	W- 5173	Aldenhoven	02464 / 8920	
Dose	Volker	Dorfstraße 10	W- 2304	Brodersdorf	04343 / 1357	
Drechsler	Oskar	Duckterather Busch 2	W- 5060	Bergisch Gladbach 2	02202 / 55282	0228 / 128114
Gill	Thomas	Maria-Eich-Str. 34	W- 8000	München 60	089 /8349527	
Günther	Jens	Bannerscheid 7	W- 5231	Neitersen	02681 / 1553	
Halgasch	Gert	Großschonauer Straße 26	O- 8805	Jonsdorf		
Hartmann	Hans-Günther	Möwenstraße 9	W- 2876	Berne 2	04406 / 6911	0421 /2482419
Hebecker	Ulrich	Büsumerstraße 15	W- 7000	Stuttgart	0711 / 734800	
Held	Manfred	Stirner Straße 22	W- 8835	Pleinfeld	09144 / 6563	0911 /2195245
Hermann	Klaus	Forchenstraße 8	W- 7401	Pliezhausen	07127 / 71945	
Hürdler	Manfred	Niederhofer Straße 29	W- 8709	Rimper	09365 / 4235	
Johnen	Willi	Hansemannstraße 1	W- 5160	Düren	02421 / 501305	
Kauka	Dietmar	Straße des Friedens 37	A- 7201	Neukirchen	0037404/850112	
Kemmer	Jürgen	Dorfberg 7	W- 8701	Sulzdorf	09334 / 1050	
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 /4828948	00411 / 258170
Kuhn	Eckehard	Im Dorf 14	W- 7443	Frickenhausen	07022 / 45417	
Littmann	Claus	Plockhorst, Zum Spring 15	W- 3155	Edemissen	05372 / 7796	
Lorenz	Walter	Mahräckerstraße 9	W- 6000	Frankfurt /Main 50	069 / 531656	
Magnus	Andreas	Bismarckstraße 29	W- 4650	Gelsenkirchen	0209 / 870230	
Mahlert	Herbert	Hohenbudbergerstraße 112 A	W- 4100	Duisburg	02135 / 47217	
Mand	Harald	Kl. Flintbeker Straße 7	W- 2302	Flintbek /Kiel	04347 / 3629	0431 /3013580
Menk	Christian	Ollsener Straße 52	W- 2116	Hanstedt	04184 / 7825	
Müller	Kurt	Sophie-Scholl-Ring 3b	W- 2054	Geesthacht	04152 / 70643	04151 / 889137
Mössel	Franz	Schafferstraße 12	I- 39012	Meran		
Neebe	Gerhard	Märkische Straße 186	W- 4600	Dortmund 1	0231 / 416549	
Neueder	Jens	Rudolf-Then-Straße 32	W- 7178	Gschlachtenbretzingen	0791 / 42877	0791 /44-4722
Neumann	Christof	Zeitblomstraße 22/2	W- 7910	Ulm		
Nitschke	Stefan	Germanenstraße 5	W- 7519	Walzbachtal 1	07203 / 452	
Obermann	Hartmut	Mozartring 23	W- 8870	Günzburg	08221 / 30248	
Peters	Jürgen	Heukoppel 14	W- 2000	Hamburg 14	040 /6412371	
Reit	Hermann	Vechter Hof 40	W- 4500	Osnabrück	0541 / 16331	
Retzlaff	Bernd	Kleiner Sand 98	W- 2082	Uetersen	04122 / 43551	04103 / 605310
Riechmann	Michael	Letelner Heideweg 12	W- 4950	Minden	0571 / 36627	
Rinio	Gerd	Rennbahnstraße 9	W- 2000	Hamburg 74	040 /6552630	
Ruschinski	Claus	Pommernstraße 21	W- 4370	Marl	02365 / 34646	
Schilling	Werner	Ehndorfer Straße 340	W- 2350	Neumünster	04321 / 61116	
Schimmer	Jörg	Castellring 55	W- 6369	Nidderau 1	06187 / 25503	
Schmid	Alexander	St. Cajetan Straße 38/VII	W- 8000	München 80	089 / 495326	
Schmitz	Rainer	Dornierstraße 17	W- 7320	Göppingen	07161 / 22549	
Schober	Frank-Michael	Weberweg 2	O- 7590	Spremberg	0037574/ 4565	
Schoberth	Uwe	Petrus-Waldus-Straße 14	W- 7136	Oetisheim	07041 / 7254	0711 /2011455
Schroers	Horst-Dieter	Breslauer Straße 9	W- 8016	Feldkirchen	089 /9032615	089 /9032615
Schröder	Gerald	Am Schützenplatz 14	W- 2105	Seevetal 1	04105 / 2602	
Schröder	Peter	Theodor-Fahr-Straße 32	W- 2000	Hamburg 62	040 /5311582	
Schröer	Egbert	Joachimstraße 18	W- 4270	Dorsten 1	02362 / 75840	02365 / 495183
Schulte	Hartmut	Entenschnabel 8	W- 3162	Uetze	05173 / 1248	
Schut	Andre	Sanderstraße 26	W- 1000	Berlin 44	030 /6917861	
Schwarz	Wolfgang	Schwedenring 6	W- 8850	Donauwörth	0906 / 3092	
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	W- 5305	Alfter 4	0228 / 643853	
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanloese		
Stumpferl	Stefan	Hasenberglstraße 57	W- 8000	München 45	089 /3144001	
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	W- 6500	Mainz 1	06131 / 320860	06131 / 395268
Tornow	Wilhelm	Görlitzer Straße 16	W- 2190	Cuxhaven 13	04723 / 1355	04723 / 3063
Vollkmer	Richart	Am Spörkel 69	W- 4600	Dortmund 50		
Volz	Oliver	Waldenburgstraße 73	W- 7000	Stuttgart 80	0711 /7353817	
Werner	Heiko	Reichenberger Straße 5	O- 8032	Dresden		
Wittkamp	Heinz	Hindenburgstraße 37	W- 5630	Remscheid	02191 / 75132	
Wulf	Hans-Otto	Im Brahmkamp 38	W- 4250	Bottrop	02041 / 688972	

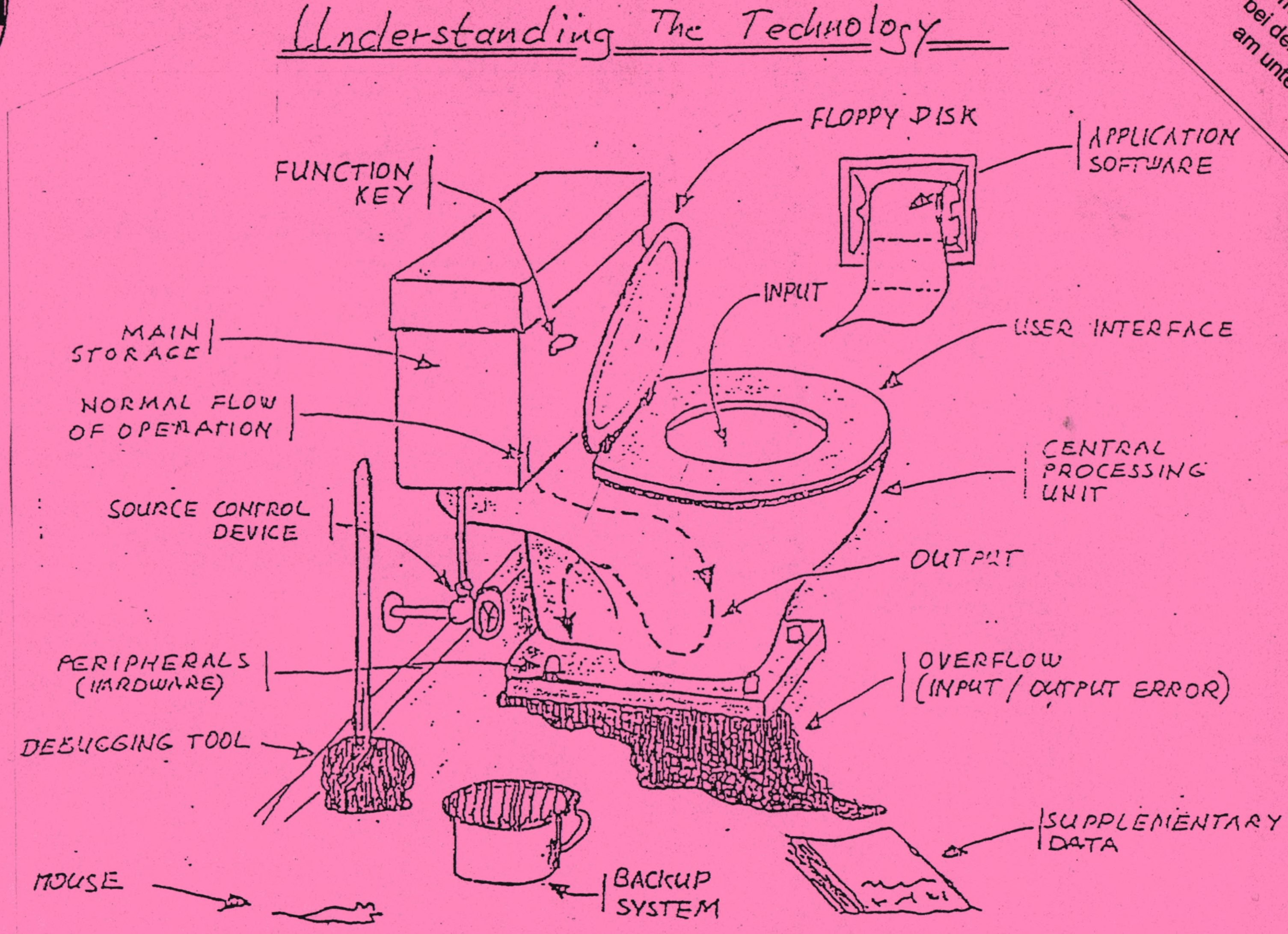
HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

Achtung !!
wichtiger Termin !
Hobbytronik-Treffen
am Sonntag, den 8.11.92
gleich morgens zum Einlaßbeginn
oder
mittags, 12.00 Uhr
bei der Kartenkontrolle
am unteren Eingang



CLUBINFO

37. Ausgabe

Kontaktadresse:
Club 80
Fritz Chwolka
Saarstrasse 34
5173 Aldenhoven
Tel.: 02464/8920

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Vorwort 1
Fritz Chwolka

Termine 2
Redaktion

Vorstellung 3
Wolfgang Berndt

Software

Anpassung von Pascal-80 an Turbo-Pascal 4
PIO1.Pas PC--Turbo-Pascal 5
PIO1.Pas TRS-80--Pascal-80 6
Einblicke für künftige Schüler 7 - 8
FUNK3/PAS 8 - 14
Wolfgang Berndt

Der Atari Portfolio, ein echter MSDOS-Computer? 15 - 18
Egbert Schröer

ZEUS spricht Intel-Hex 19 - 24
Heinrich Betz

Aufbacken der harten Scheiben BACKUP/CMD 25 - 27
Es lebe der Fortschritt 28 - 30
Nie mehr das verfluchte DOS-Handbuch suchen 31 - 32
INV/CMD 33 - 36
Wer sucht der FINDet 37 - 44
Volker Dose

BASIC Vorworte, Meinungen 45
Willi Johnen, Jens Neueder

BASIC -Neuen Wind für eine totgeglaubte Sprache 46 - 48
Christof Neumann

ADRESS-TEXT-90 Teil III 49 - 54
Willi Johnen

Autor & Seite

Hardware

Der Ton macht die Musik 55 - 56
Artikel aus WIN

Börse

Wer hat was – wer will was 57 - 58

Sonstiges

Zehn Jahre ZCRP 59 - 67
Übersetzung Hartmut Schulte

Deterministisches Chaos als ergänzende ... 68 - 70
Fraktale -Vorkommen und Anwendung ... 71 79
Artikel aus CLB

Die letzten Seiten

Impressum 2
Schluß 80
Redaktion

Mitgliederadressenliste am INFO-Ende

Achtung !!
wichtiger Termin !

Hobbytronik-Treffen

am Sonntag, den 8.11.92

gleich morgens zum Einlaßbeginn
oder
mittags, 12.00 Uhr
bei der Kartenkontrolle
am unteren Eingang

VORWORT

oder andersrum: We are back again!

Endlich habe ich wieder etwas Zeit. Ich muß mich bei Euch allen erst mal für das lange Ausbleiben des Clubinfos und das teilweise Nichtbeantworten von Briefen entschuldigen. Neben meiner beruflichen Tätigkeit im Wechselschichtbetrieb auf einem Kraftwerk machte ich per Fernschule den Maschinenbautechniker und habe jetzt alle Prüfungen abgeschlossen und die ersehnte Qualifikation.

Dieses Jahr hatten wir ein kurzfristig von Hartmut Obermann initiertes Clubtreffen, welches natürlich auch von mir besucht wurde. In einer zu nächtlicher Stunde anberaumten Sitzung hat der anwesende Vorstand eine Senkung des Mitgliedsbeitrages verabschiedet.

Der Vorstand war Mangels Anwesenheit leider nicht zur Wahl eines neuen Vorsitzenden beschlußfähig. Ich hätte mein Amt gerne zur Verfügung gestellt, vor allem wegen Zeitmangel , dies ist aber auf eine nächste Vorstandssitzung verschoben. Ich würde gerne die Arbeit des Softwarearchivatord uebernehmen, da sich die meisten doch direkt an mich wenden und Ruediger auch kaum noch Zeit hat.

Meine Meinung: Ich arbeite gerne im Club und helfe wo ich kann, wenn nur nicht diese administrativen Tätigkeiten währen.

Allgemein ist der Mitgliederstand konstant geblieben. Neuzugaenge und Abmeldungen halten sich die Waage. Allerdings werden die Nutzer von CP/M und Erweiterungen dazu (ZCPR...) mehr.

Vielen Dank fuer Eure Geduld mit mir
und viele Gruesse

F.Chwolka

Impressum

1. Vorsitzender — Fritz Chwolka
Tel.: 0 24 64/89 20
Saarstraße 34
5173 Aldenhoven

2. Vorsitzender — Gerald Schröder
Tel.: 0 41 05/26 02
Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator — Andreas Magnus
Tel.: 02 09/87 02 30
Bismarckstraße 29
4650 Gelsenkirchen

Newdos-Diskothekar — Oliver Volz
Tel.: 07 11/6 85 30 13
Am Ochsenwald 37A
7000 Stuttgart 80 (Rohrerhöhe)

CP/M-Diskothekar — Rüdiger Sörensen
Tel.: 0 61 31/32 06 60
Thomas-Mann-Straße 3a
6500 Mainz 1

Clubbücherei — Kurt Müller
Tel.: 0 41 52/7 06 43
Sophie-Scholl-Ring 3b
2054 Geesthacht

Redaktion — Jens Neueder
Tel.: 07 91/4 28 77
Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 8870 Günzburg
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Termine... Termine... Termine... Termine... Termine

Orgatec	Köln	22.10.- 27.10.92
Electronica	München	10.11.- 14.11.92

Redaktionsschluß für das nächste Clubinfo ist:

Mitte Dezember 92, damit rechtzeitig zum Jahreswechsel das INFO auch fertig wird.

Liebe Clubfreunde,

seit zwei Monaten bin ich Mitglied im Club-80 und möchte mich kurz vorstellen. Ich bin 42 Jahre alt, Physiklehrer (auch Chemie und Informatik) und habe 1980 bei einer Lehrerfortbildung an meiner Schule den TRS-80 Model I von TANDY kennengelernt (BASIC-Lehrgang). Weil die Geräte unverwüstlich sind, funktioniert die ganze Anlage bis heute. Ein Reporter der FRANKFURTER RUNDSCHAU hat am Tag der offenen Tür (8.2.1992) ein Photo von unserem Computerraum II (Im Computerraum I stehen neue 386-er SX-Rechner) geschossen. Weil die alten TRS-80-Geräte alle mit PIOs, AD-Wandlern usw. ausgestattet sind und mein Interesse dem Gebiet Messen, Steuern und Regeln gilt, bin ich mehr im Computerraum II als im Computerraum I zu finden.
Nachdem ich jahrelang mit BASIC programmiert hatte, habe ich mich nun mehr der Sprache PASCAL zugewandt. Leider habe ich mit ALCOR-PASCAL nicht den Durchbruch geschafft. Ich glaube, daß die von mir benutzte Version fehlerhaft war. Dann entdeckte ich PASCAL-80 und bin damit sehr zufrieden. Leider ist die Beschreibung sehr knapp. Wenn man wissen will, wie man z.B. den Stapelspeicher verändern will, so erhält man keinen Hinweis.
Da wir neben den TRS-80 noch drei alte PCs haben, auf denen inzwischen die TURBO-PASCAL-Version 6.0 läuft, war ich gezwungen, die Programme so zu schreiben, daß sie mit wenig Änderung auf beiden Geräten lauffähig sind. Das Programm zur Dekodierung der Funkuhrsignale läuft auf einem modernen PC genauso wie auf dem Model I. Dies ist möglich, weil kleine Zusatzprogramme wie TURBO1/PAS und PIO1/PAS die Unterschiede zwischen PASCAL-80 und TURBO-PASCAL ausgleichen. Es folgt nun das Programm mit ein paar Hinweisen, wie eine PIO (hier 8255) an den TRS-80 angeschlossen wird. Ich bin gerne bereit, weitere Auskünfte zu dieser Arbeit zu geben. Gibt es noch jemanden, der mit PASCAL-80 arbeitet? Bitte melden!

Ich wünsche allen noch viele neue Ideen auf den alten Rechnern

Wolfgang Brock


```
(*****************************************************************)
(*                                                               *)
(*    TURBO1/PAS      VERSION 1.0       -        18.01.1992      *)
(*                                                               *)
(*    ---- T R S - 8 0    ----     P A S C A L  8 0  ----        *)
(*                                                               *)
(*    ERLAUBT:      ANPASSUNG VON PASCAL-80                      *)
(*                  AN TURBO-PASCAL                              *)
(*                                                               *)
(*****************************************************************)

PROCEDURE DELAY (D: INTEGER);
VAR I: INTEGER;
BEGIN
  FOR I:=1 TO D DO
END;

PROCEDURE CLRSCR;
BEGIN
  CLS;
END;

FUNCTION KEYPRESSED: BOOLEAN;
BEGIN
  IF INKEY=CHR(0) THEN KEYPRESSED:=FALSE
                  ELSE KEYPRESSED:=TRUE
END;

FUNCTION READKEY: CHAR;
VAR CH: CHAR;
BEGIN
  REPEAT
    CH:= INKEY
  UNTIL ORD(CH)>0;
  READKEY:= CH
END;

PROCEDURE SETTIME(STUNDEN, MINUTEN, SEKUNDEN, HUNDERTSTEL:
          INTEGER);
BEGIN
  POKE(16449,SEKUNDEN);
  POKE(16450,MINUTEN);
  POKE(16451,STUNDEN);
END;

PROCEDURE SETDATE(JAHR, MONAT, TAG: INTEGER);
BEGIN
  POKE(16452,JAHR-1900);
  POKE(16453,TAG);
  POKE(16454,MONAT)
END;

(* WIRD FORTGESETZT *)
```

```
(***************************************************************)
(*                                                             *)
(*   PIO1.PAS      VERSION 1.0      DATUM :  05.03.1992        *)
(*                                                             *)
(*   ------ P C ------   T U R B O - P A S C A L -----          *)
(*                                                             *)
(*                                                             *)
(*   ERLAUBT:      PIO('AUS','EIN','AUS');                     *)
(*                 OUT('A',BYTE);                              *)
(*                 BYTE:=INP('A');                             *)
(*                                                             *)
(*   ACHTUNG:      BEI MANCHEN COMPUTERN SIND NUR EIN ODER     *)
(*                 ZWEI KANÄLE ('A','B') ZU BENUTZEN!          *)
(*                                                             *)
(***************************************************************)

PROCEDURE OUT(CH: CHAR; BYTE: INTEGER);
BEGIN
  IF (CH='A') OR (CH='a') THEN PORT[888]:=BYTE;
END;

FUNCTION INP(CH: CHAR): INTEGER;
BEGIN
  IF (CH='A') OR (CH='a') THEN INP:=PORT[888]
END;

PROCEDURE PIO(A,B,C: STR3);
BEGIN
  IF (A='AUS') OR (A='aus') OR (A='Aus') THEN OUT('A',0);
  IF (A='EIN') OR (A='ein') OR (A='Ein') THEN OUT('A',255)
END;
```



```
(***************************************************************)
(*                                                             *)
(*   PIO1/PAS      VERSION 1.0      -      18.01.1992          *)
(*                                                             *)
(*   ---- T R S - 8 0  ----   P A S C A L  8 0  ----           *)
(*                                                             *)
(*   ERLAUBT:      PIO('AUS','EIN','AUS');                     *)
(*                 OUT('A',2);                                 *)
(*                 OUT('C',5);                                 *)
(*                 X:=INP('B');                                *)
(*                                                             *)
(***************************************************************)

PROCEDURE OUT(CH: CHAR; B: INTEGER);
VAR A,BYTE,NAME: INTEGER;
BEGIN
  CASE CH OF
    '7': A:=7;
    'A': A:=4;
    'B': A:=5;
    'C': A:=6
  END;
  POKE(-5536,  62);
  POKE(-5535,   B);
  POKE(-5534, 211);
  POKE(-5533,   A);
  POKE(-5532, 201);
  NAME:=CALL(-5536,BYTE)
END;

FUNCTION INP(CH: CHAR): INTEGER;
VAR A: INTEGER;
BEGIN
  CASE CH OF
    'A': A:=4;
    'B': A:=5;
    'C': A:=6
  END;
  POKE(-5530, 219);
  POKE(-5529,   A);
  POKE(-5528, 201);
  INP:=CALL(-5530,0)
END;

PROCEDURE PIO(A,B,C: STR3);
VAR X,Y,Z,ZAHL: INTEGER;
BEGIN
  X:=0;
  Y:=0;
  Z:=0;
  IF A='EIN' THEN X:=16;
  IF B='EIN' THEN Y:= 2;
  IF C='EIN' THEN Z:= 9;
  ZAHL:=128+X+Y+Z;
  OUT('7',ZAHL)
END;
```

Viele neugierige Besucher beim Tag der offenen Tür in der Friedberger Gesamtschule

0 8. FEB. 1992

Einblicke für künftige Schüler

Etliche Besuchergruppen mußte Walter Scheibner, der Leiter des Realschulzweigs, am Samstag durch die Friedberger Gesamtschule — im Bild einer der EDV-Räume — führen. (FR-Bilder: Eberhardt)

FRIEDBERG. „Die Anlage hier ist ganz neu. Wir haben sie erst vor drei Tagen bekommen", sagt Walter Scheibner. „Maximal 18 Schüler — jeweils zwei an einem Platz — können hier arbeiten." Zuvor hatte sich die buntgemischte Gruppe von Müttern, Vätern und Kindern einen anderen EDV-Raum angesehen, wo fleißig getüftelt wurde. Wie es sich für einen Hausherrn gehört, führte der Leiter des Realschulzweigs der Friedberger Gesamtschule am Samstag seine Gäste durch das Haus. Und die kamen beim „Tag der offenen Tür" in Scharen.

Während andere erst mal gucken wollten, war für Thomas die Entscheidung bereits gefallen. Ab dem kommenden Schuljahr soll der Junge, der im April, zehn Jahre alt wird, den Realschulweig der Gesamtschule besuchen, erzählt seine Mutter. Da ist der „Tag der offenen Tür" eine gute Gelegenheit, schon einmal einen Blick in die künftige „Lernstätte" zu werfen. Und wie gefällt sie ihm? „Ganz nett", findet Thomas und trollt sich.

Im Foyer sind Werke aus dem Kunst-Unterricht ausgestellt. Ein Katzenkopf, der aus einem Kupferblech getrieben wurde, Bilder in Pop-Art, Federzeichnungen, Foto-Collagen. Mehr als das interessieren Thomas aber die Computerterminals, an denen einige ältere Schüler arbeiten: Ein Elektronenhirn nimmt über vier Fotowiderstände an den Fenstern Informationen von draußen auf. Ein anderes fragt die Außentemperatur ab. Spätestens im Sommer eine ganz nützliche Einrichtung, um nachzusehen, ob nicht bald die Temperatur für „hitzefrei" erreicht ist.

Beim Gang durch die Räume der Gesamtschule mit ihren über 900 Schülern beläßt es Scheibner aber nicht bei solchen Vorzeigeprojekten. Der Lehrer weist auch auf Probleme der Schule hin: Neue Anstriche für einzelne Räume sind

gen machen Scheibner Pläne, der Schule einen Teil ihres Gebäudekomplexes wegzunehmen. Der Kreis möchte dort einen Jahrgang der Augustinerschule unterbringen.

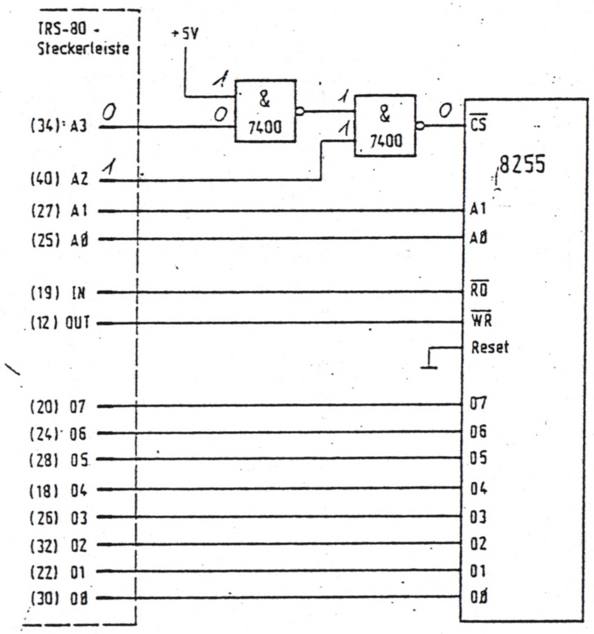
Für die Schule hat das tiefgreifende Folgen für das pädagogische Konzept: Wo Hauptschul-, Realschul- und Gymnasiumsklassen eines Jahrgangs nicht mehr wie bisher in einem Flügel untergebracht werden können, wird auch der Wechsel zwischen den einzelnen Schulzweigen

```
(**********************************************************************)
(*                                                                    *)
(*   PROGRAMM  :  >FUNK3/PAS<            DATUM  :    05.03.1992       *)
(*                                                                    *)
(*                                                                    *)
(*   COMPUTER  :  TRS-80        UND      PC                           *)
(*                                                                    *)
(*   SPRACHE   :  PASCAL-80     UND      TURBO-PASCAL-6.01            *)
(*                                                                    *)
(*                                                                    *)
(*   GESAMTSCHULE FRIEDBERG - WAHLPFLICHTKURS KLASSE 10               *)
(*                                                                    *)
(**********************************************************************)

PROGRAM FUNK3;

 USES CRT; DOS;      (*        <--- BEI TURBO-PASCAL          *)

TYPE STR3     = ARRAY(.1..03.) OF CHAR;                  (* FUER PIO *)
     WERTTYP  = ARRAY(.0..59.) OF INTEGER;
     CHARTYP  = ARRAY(.0..59.) OF CHAR;
VAR  ZEIT                   : BOOLEAN;
     WERT                   : WERTTYP;
     WERTZEICHEN            : CHARTYP;
     MINUTEN, STUNDEN,
     TAG, WOCHENTAG,
     MONAT, JAHR, X,
     ZAHL1, ZAHL2, PC       : INTEGER;
```

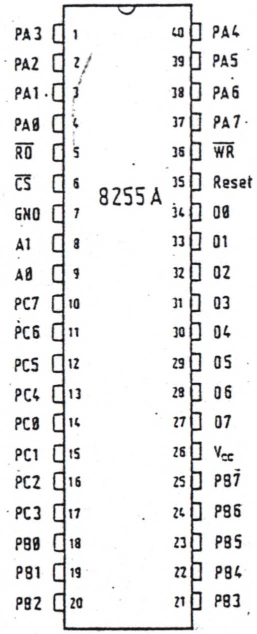
```
(****************************************************************)
(*                                                              *)
(*   COMPUTERTYP:                    ----------->  !   ENTFERNEN *)
(*                                                              *)
(*--------------------------------------------------------------*)
(*                                                              *)
(*   TRS-80  ----->    *)          (*!$  TURBO1/PAS             *)
(*   TRS-80  ----->    *)          (*!$  PIO1/PAS               *)
(*                                                              *)
(*   PC      ----->    *)          (*$I PIO1.PAS               *)
(*                                                              *)
(****************************************************************)

PROCEDURE COMPUTER_TYP;
VAR WAHL: CHAR;
BEGIN
  WRITELN('AUSWAHL DES COMPUTERS:');
  WRITELN;
  WRITELN('>1<    TRS-80              ');
  WRITELN('>2<    PC                  ');
  REPEAT
    WAHL:= READKEY
  UNTIL WAHL IN (.'1','2'.);
  IF WAHL='1' THEN PC:=1
              ELSE PC:=110
END;

PROCEDURE INIT;
BEGIN
  CLRSCR;
  COMPUTER_TYP;
  ZAHL1:= 70*PC;
  ZAHL2:=  7*PC;
  PIO('EIN','AUS','AUS')
END;

PROCEDURE MINUTENANFANG_SUCHEN;
VAR ZAEHLER, MARKE: INTEGER;
BEGIN
  WRITELN('MINUTENANFANG SUCHEN, BITTE WARTEN');
  REPEAT
    ZAEHLER:=0;
    REPEAT
      MARKE:= INP('A');
      MARKE:= MARKE MOD 2;
      IF MARKE=1 THEN ZAEHLER:= ZAEHLER+1
    UNTIL MARKE=0;
    WRITE(ZAEHLER,' ');
  UNTIL ZAEHLER>ZAHL1;
END;

PROCEDURE INFO_EINLESEN;
VAR ZAEHLER, MARKE: INTEGER;
BEGIN
  WRITELN;
  WRITELN('INFOEINLESEN, BITTE 1 MINUTE WARTEN');
  X:= 0;
  REPEAT
    ZAEHLER:=0;
    REPEAT
      MARKE:=INP('A');
      MARKE:=MARKE MOD 2;
      IF MARKE=0 THEN ZAEHLER:= ZAEHLER+1
    UNTIL MARKE=1;
    IF ZAEHLER>0 THEN
      BEGIN
        IF ZAEHLER>ZAHL2 THEN WERT(.X.):=1
                         ELSE WERT(.X.):=0;
        X:=X+1;
        WRITE(X)
      END
    UNTIL X=59;
  WRITELN;
END;

PROCEDURE INFO_AUSGEBEN;
BEGIN
  FOR X:=1 TO 59 DO
    BEGIN
      IF WERT(.X.)=0 THEN WERTZEICHEN(.X.):='.'
                     ELSE WERTZEICHEN(.X.):='-';
      WRITE(WERTZEICHEN(.X.))
    END;
  WRITELN
END;

PROCEDURE INFO_DEKODIEREN;
BEGIN
  WRITELN('INFO DEKODIEREN');
  MINUTEN    := WERT(.21.)* 1+WERT(.22.)* 2+WERT(.23.)* 4+
                WERT(.24.)* 8+WERT(.25.)*10+WERT(.26.)*20+
                WERT(.27.)*40;
  STUNDEN    := WERT(.29.)* 1+WERT(.30.)* 2+WERT(.31.)* 4+
                WERT(.32.)* 8+WERT(.33.)*10+WERT(.34.)*20;
  TAG        := WERT(.36.)* 1+WERT(.37.)* 2+WERT(.38.)* 4+
                WERT(.39.)* 8+WERT(.40.)*10+WERT(.41.)*20;
  WOCHENTAG  := WERT(.42.)* 1+WERT(.43.)* 2+WERT(.44.)* 4;
  MONAT      := WERT(.45.)* 1+WERT(.46.)* 2+WERT(.47.)* 4+
                WERT(.48.)* 8+WERT(.49.)*10;
  JAHR       := WERT(.50.)* 1+WERT(.51.)* 2+WERT(.52.)* 4+
                WERT(.53.)* 8+WERT(.54.)*10+WERT(.55.)*20+
                WERT(.56,)*40+WERT(.57.)*80+1900;
END;

PROCEDURE ZEIT_HOLEN;
BEGIN
  MINUTENANFANG_SUCHEN;
  INFO_EINLESEN;
  INFO_AUSGEBEN;
  INFO_DEKODIEREN;
END;

PROCEDURE UHR_STELLEN;
BEGIN
  SETDATE(JAHR,MONAT,TAG);
  SETTIME(STUNDEN,MINUTEN,0,0)
END;

FUNCTION MINUTEN_KONTROLLE: BOOLEAN;
VAR I, BIT_SUMME: INTEGER;
BEGIN
  BIT_SUMME:= 0;
  FOR I:=21 TO 28 DO
    IF WERT(.I.)=1 THEN BIT_SUMME:= BIT_SUMME+1;
  IF BIT_SUMME MOD 2=0 THEN MINUTEN_KONTROLLE:=TRUE
                       ELSE MINUTEN_KONTROLLE:=FALSE
END;

FUNCTION STUNDEN_KONTROLLE: BOOLEAN;
VAR I, BIT_SUMME: INTEGER;
BEGIN
  BIT_SUMME:= 0;
  FOR I:=29 TO 35 DO
    IF WERT(.I.)=1 THEN BIT_SUMME:= BIT_SUMME+1;
  IF BIT_SUMME MOD 2=0 THEN STUNDEN_KONTROLLE:=TRUE
                       ELSE STUNDEN_KONTROLLE:=FALSE
END;

FUNCTION TAG_KONTROLLE: BOOLEAN;
VAR I, BIT_SUMME: INTEGER;
BEGIN
  BIT_SUMME:= 0;
  FOR I:=36 TO 58 DO
    IF WERT(.I.)=1 THEN BIT_SUMME:= BIT_SUMME+1;
  IF BIT_SUMME MOD 2=0 THEN TAG_KONTROLLE:=TRUE
                       ELSE TAG_KONTROLLE:=FALSE
END;
```

```
PROCEDURE ZEIT_KONTROLLE;
VAR M, S, T: BOOLEAN;
BEGIN
   M:= MINUTEN_KONTROLLE;
   S:= STUNDEN_KONTROLLE;
   T:= TAG_KONTROLLE;
   IF (M=TRUE) AND (S=TRUE) AND (T=TRUE) THEN ZEIT:= TRUE
                                         ELSE ZEIT:= FALSE
END;

PROCEDURE ZEIT_ANZEIGEN;
BEGIN
   CLRSCR;
   WRITE('HEUTE IST ');
   CASE WOCHENTAG OF
      1: WRITE('MONTAG     ');
      2: WRITE('DIENSTAG   ');
      3: WRITE('MITTWOCH   ');
      4: WRITE('DONNERSTAG ');
      5: WRITE('FREITAG    ');
      6: WRITE('SAMSTAG    ');
      7: WRITE('SONNTAG    ')
   END;
   WRITE('DER  ');
   WRITE(TAG:2,'.',MONAT:2,'.',JAHR:4,'      ');
   WRITE(STUNDEN:2,':',MINUTEN:2,' UHR');
END;

PROCEDURE ENDE;
BEGIN
   REPEAT
   UNTIL KEYPRESSED
END;


BEGIN (* HAUPTPROGRAMM *)
   INIT;
   REPEAT
      ZEIT_HOLEN;
      UHR_STELLEN;
      ZEIT_KONTROLLE;
   UNTIL ZEIT=TRUE;
   ZEIT_ANZEIGEN;
   ENDE
END.
```


Spezial-Interface



Schaltbild des Spezial-Interface. Die angegebenen Zahlen sind die Sockelstift-Nummern des TRS-80 Steckers



Anschlußbezeichnungen

D_0 - D_7	Daten-Bus (Zweiweg)
RESET	Rücksetz-Eingang
CS	Baustein-Auswahl
$\overline{RD}$	Lese-Eingang
$\overline{WR}$	Schreib-Eingang
A0, A1	Kanal-Adresse
PA_0 - PA_7	Kanal A (Bit 0 bis 7)
PB_0 - PB_7	Kanal B (Bit 0 bis 7)
PC_0 - PC_7	Kanal C (Bit 0 bis 7)
V_{cc}	Versorgungsspannung (+ 5V)
GND	Masse (0 V)

Anschluß-Belegung und Anschlußbezeichnungen des Bausteins 8255

Tabelle 1. Schema der beim Sender DCF77 verwendeten Zeitcodierung

Sek.	Bed.	Pegel	Wert	Zuordnung	Beispiel	
0	M	0	–	Minutenbeginn	0	
1...						
14	–	X	–	z. Z. nicht belegt	–	
15	R	X	–	Reserveantenne	0	
16	A	X	–	Ankündigung Sommerzeit	0	
17	Z1	X	2^1	Zonenzeit-bits	1	
18	Z2	X	2^0		0	
19	–	X	–	z. Z. nicht belegt	–	
20	S	1	–	Start der Zeitcod.	1	
21		X	1		1	5
22		X	2		0	
23		X	4	Einer	1	
24		X	8		0	
				Minuten		
25		X	10		1	3
26		X	20	Zehner	1	
27		X	40		0	
28	P1	X	–	Prüfbit 1	0	
29		X	1		1	9
30		X	2		0	
31		X	4	Einer	0	
32		X	8		1	
				Stunden		
33		X	10		1	1
34		X	20	Zehner	0	
35	P2	X	–	Prüfbit 2	1	
36		X	1		1	5
37		X	2		0	
38		X	4	Einer	1	
39		X	8		0	
				Kalendertag		
40		X	10		1	1
41		X	20	Zehner	0	
42		X	1		0	4
43		X	2	Wochentag	0	
44		X	4		1	
45		X	1		1	7
46		X	2	Einer	1	
47		X	4		1	
48		X	8		0	
				Kalendermonat		
49		X	10	Zehner	0	0
50		X	1		0	2
51		X	2		1	
52		X	4	Einer	0	
53		X	8		0	
				Kalenderjahr		
54		X	10		0	8
55		X	20		0	
56		X	40	Zehner	0	
57		X	80		1	
58	P3	X	–	Prüfbit 3	1	
59	–	–	–	Marke fehlt	–	

Pegel: 0 = fest LOW; 1 = fest HIGH; X = LOW oder HIGH
Beispiel: Do (= 4), 15. 07. 82, 19.35 Uhr

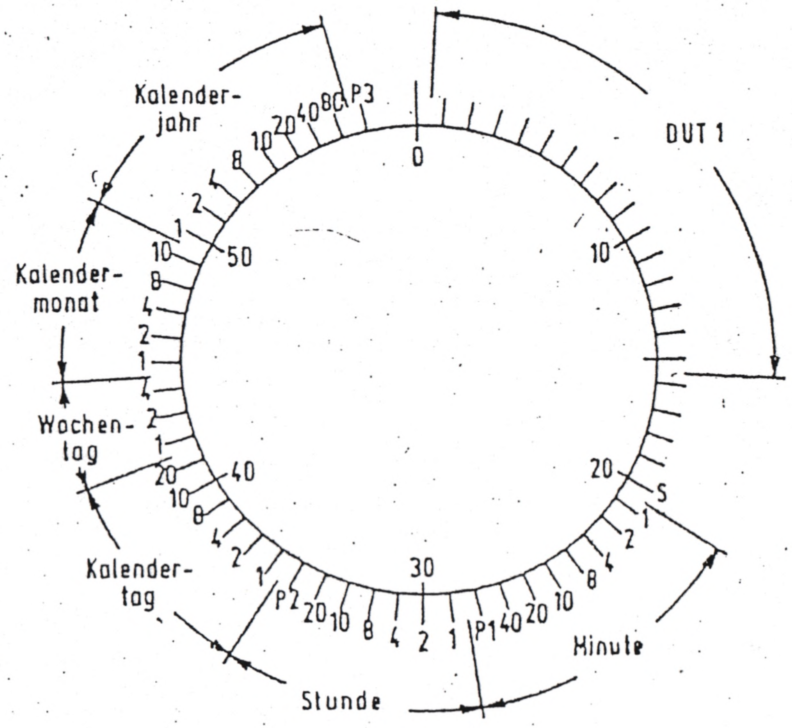


Bild 1. Schema der codierten Zeitinformation. P 1, P 2, P 3 Prüfbits, S auf 0,2 s verlängerte Sekundenmarke zum Start der Codierung der Zeitinformation

Technik aktuell

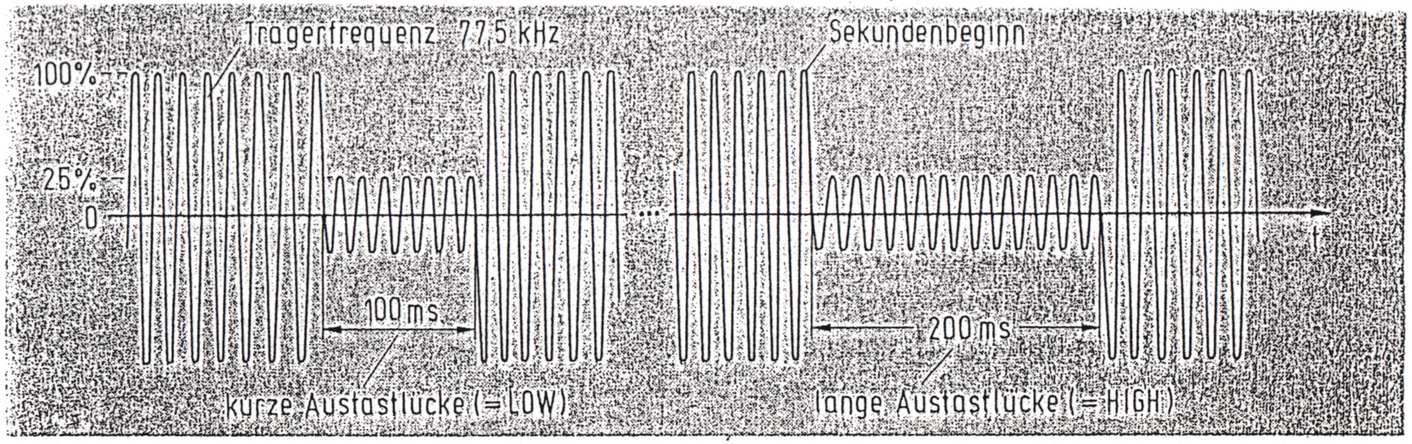


Bild 1: Die codierte Zeitinformation „steckt“ in den unterschiedlich langen Trägerabsenkungen (Sekundenmarken).

Der Atari Portfolio,
ein "echter" MSDOS-Computer ?

Möglichkeiten eines Pocket PC's

1. Vorwort
Trotzdem der Artikel einen Außenseiter unter den Rechnern (und einen MSDOS-Kompatiblen dazu) beschreibt, wird dieser hoffentlich für die Club Mitglieder interessant sein.
Zumindest von einem im Club (Hallo Claus !) ist mir bekannt, daß er einen Atari Portfolio besitzt. Da die neueste SMD-Technologie aber auch andere im Club interessieren dürfte, wird dieser Artikel hoffentlich nicht nur für 2 Club-Mitglieder und zum Füllen der Info-Seiten geschrieben worden sein.

2. Allgemeines
Die Reaktionen auf das 20x10x2 cm große schwarze Kästchen gleichen sich: "Was ist denn das für ein Taschenrechner ?" oder "Ich wußte gar nicht, daß Atari Taschenrechner herstellt". Die nächste Phase des Gespräches hört sich dann etwa so an: "DAS soll ein PC sein" gefolgt von einem verächtlichen herabziehen der Lippen. Es besteht aber auch die Möglichkeit das zwei Freak's aufeinander treffen (soll es auch außerhalb der Clubtreffen geben): "Haben Sie ihre Schnittstelle dabei ?". Das 40x8 Zeichen große LCD--Display wird durch Betätigung eines Schiebeschalters hochgeklappt, daß 63 Tasten umfassende "Mäuseklavier" steht zur Verfügung und der Daten, Text oder Programmtransfer via parallele oder serielle Schnittstelle kann beginnen. Was verbirgt sich nun in dem knapp 500 g schweren "Taschen-PC" an Hardware ?

3. Hardware
Kennt jemand noch die Daten des aus dem Jahr 1983 stammenden original IBM-PC ?: 64 KByte Hauptspeicher und 4,77 MHz Taktfrequenz. Der Portfolio wartet da in seiner ursprünglichen Version mit 4,93 MHz, 128 KByte Speicher und 256 KByte ROM auf. Den Hauptspeicher von 128 KByte RAM bilden vier S-RAM Bausteine vom Typ 43256-15. Zwei SMD-ROM's beherbergen das Betriebssystem sowie die internen Programme. Es gibt inzwischen Anbieter, die den Portfolio mit 640 KByte intern aufgerüstet ausliefern. Der Portfolio besitzt aufgrund seiner Größe keinen Bildschirm von 80x25, wie bei MDA-Darstellung üblich, sondern ein 40x8 Zeichen großes nicht hintergrundbeleuchtetes LC-Display (bei ungünstigem Blickwinkel prima als Rasierspiegel zu gebrauchen). Die Punktmatrix der Buchstaben beträgt 8x8 Pixel. Dadurch rücken Unterlängen wie 'g' etwas unschön nach oben.
Anstelle der herkömmlichen 5.25"- oder 3.5" Laufwerke können zur Speicherung Chipkarten verwendet werden. An der linken Gehäuseseite befindet sich der Schlitz für die Speicherkarten. Dieses "Laufwerk A:" kann sowohl ROM als auch RAM-Cards lesen.
RAM-Cards gibt es von 32KByte bis 1MByte. Durch einen Schalter an den RAM-Cards kann man diese gegen ungewolltes Schreiben schützen.
Erweitert man den Portfolio extern mit der von Atari vertriebenen 256K Speichererweiterung, so erhält man zusätzlich noch ein "Laufwerk B:".
Durch diese Speichererweiterungen verliert der Portfolio sein handliches Format, allerdings ist das Laufwerk B: recht nützlich. An der rechten Gehäuseseite hinter einer Plastikkappe verborgen befindet sich der Erweiterungsport an dem man wahlweise die Erweiterungsmodule (max. 2) und/oder eine RS 232 Schnittstelle oder eine parallele Schnittstelle anschließen kann. Stellt man sich nun die maximale externe Hardware-Erweiterung von 2 Speichererweiterungen a 256K + parallele oder serielle Schnittstelle vor, so erreicht der 20 cm Breite Rechner einen 21 cm langen "Anbau". Da Atari keine Abhilfe schaffte, wird eine Erweiterung von anderen Firmen vertrieben, welche 512K + Laufwerk B: + parallele/serielle Schnittstelle enthält. Länge: Sagenhafte 5 cm. Die Portbelegung des Portfolio ist mir leider nicht bekannt.

4. Steckerbelegung der Schnittstellen
4.1. Parallelschnittstelle

Pol	Richtung	Signal
1	AUS	Strobe
2	AUS	Data0
3	AUS	Data1
4	AUS	Data2
5	AUS	Data3
6	AUS	Data4
7	AUS	Data5
8	AUS	Data6
9	AUS	Data7
10	EIN	Acknowledge
11	EIN	Busy
12	EIN	Papierfehler
13	EIN	Select
14	AUS	Auto Feed
15	EIN	Fehler
16	AUS	Drucker Initialisieren
17	AUS	Select Input
18-25		Ground

4.2. Serielle Schnittstelle

Pol	Signal	Funktion	
1	CD	Carrier Detect	(Empfangssignal)
2	RD	Receive Data	(Empfangsdaten)
3	TD	Transmit Data	(Sendedaten)
4	DTR	Data Terminal Ready	(DEE betriebsbereit)
5	GND	Signal Ground	(Signal Erde)
6	DSR	Data Set Ready	(Betriebsbereitschaft)
7	RTS	Request To Send	(Sendeteil einschalten)
8	CTS	Clear To Send	(Sendebereitschaft)
9	RI	Ring Indicator	(Ankommender Ruf)

5. Eingebaute Software
Die ROMs des Portfolio enthalten einige nützliche Anwendungsprogramm. Alle Programme belegen die Funktionstasten einheitlich, so daß man sich schnell an deren Funktion gewöhnt. Alle Funktionen können über Popdown-Menüs aufgerufen werden. Ein Klemmbrett (Clipboard) ermöglicht den Austausch von Daten zwischen den integrierten Programmen. Aufgerufen werden die Programme über die 'ATARI'-Taste oder über ein Menü, welches mit dem Befehl APP gestartet wird. Verfügbar sind:

- Adreßbuch
- Rechner
- Zeitplaner
- Textverarbeitung
- Tabellenkalkulation

Hinzu kommen noch die Möglichkeiten der Systemvorbereitung (RS232 initialisieren etc.)
Das Adreßbuch dient zum Speichern von Namen, Adressen und Telefonnummern und wählt diese automatisch.
Der einfache Rechner besitzt 5 Speicher, eine korrigierbare

Eingabezeile, Quadratwurzel, verschiedene Zahlenformate und Druckausgabe (Datei/Drucker).
Der einfach zu handhabende Texteditor besitzt die üblichen Funktionen wie Blockbildung, Verschieben, Kopieren, Suchen und Ersetzen. Den Ersatz eines Notizblocks gewährleistet dieses Programm allemal.
Eine recht interessante Anwendung ist der Zeitplaner, der einen Kalender und ein Tagebuch mit Weckfunktion bei jedem Eintrag enthält. Termine können abgelegt und einmalig, täglich, arbeitstäglich, wöchentlich, monatlich oder jährlich wiederholt werden.
Die Tabellenkalkulation ist Lotus 1-2-3 kompatibel und generiert 127 Spalten mal 255 Zeilen.
Das Problem des zu kleinen Bildschirm hat ATARI bei diesem Programm oder auch MSDOS-Programmen, die nicht auf den Portfolio Bildschirm angepaßt wurden elegant umsegelt: der intere Bildschirmspeicher umfaßt die PC-üblichen 80x25 Zeichen. Je nach Einstellung im Menü "Systemvorbereitung" wandert das 40x8 Zeichen Fenster dynamisch mit dem Cursor mit oder zeigt statisch einen Ausschnitt an, der mit der ATARI und den Cursor-Tasten über den virtuellen Bildschirm geschoben wird.
Dieses Verfahren ist sehr gewöhnungsbedürftig. Betrachtet man den im Vergleich zum kleinen LCD-Bildschirm riesigen PIEPSER (Lautsprecher wäre wohl übertrieben), so hätte man ruhig etwas mehr als 40x8 Zeichen investieren können.
Eingebaut ist auch die serielle Datenübertragung, welches den Rechner nun auch für die Z80-Welt interessant macht. Via serieller Datenübertragung ist damit ein Austausch zwischen PC und Z80 Rechner möglich.

6. Externe Software
An externen Programmen gibt es mittlerweile eine ganze Menge. Beispielsweise existiert ein BTX-Programm, diverse Modem-Programme inklusive Hardware (Mini-Modem), ein auf das ATARI-Display abgestimmtes dBASE, diverse Fremdwörter Lexika, welche den Portfolio in einen echten Übersetzungscomputer verwandeln. Generell kann man sagen, daß alle *.COM Files der MSDOS Welt fehlerfrei auf dem Portfolio laufen, bei *.EXE Files kann es schon mal Schwierigkeiten geben.
Turbo Pascal 3.0 läuft auf dem Portfolio ebenso fehlerfrei wie GW-BASIC. Erstaunlich ist, daß die entsprechenden Befehle zur Steuerung einer Farbgraphikkarte in den Programmen keinerlei Schwierigkeiten bereiten.
Einen CP/M Rechner kann man durch entsprechende Emulationsprogramme leider nicht aus dem Portfolio machen. Zwar läßt sich beispielsweise das Programm Z80MU, ein CP/M 2.2 Emulator für MSDOS-Rechner, noch fehlerfrei auf dem Portfolio installieren und starten. Der entsprechende Prompt taucht auch auf dem Display auf und der Speicher wird auch in 64K-Segmente unterteilt, allerdings lassen sich nicht mehr als ca. 10 Zeichen in der Tastaturzeile eingeben. Danach stürzt der Rechner ab.
Das man auch die Dos-Bios Interrupts per Programm nutzen kann, zeigt folgendes Beispiel-Programm in Turbo-Pascal 3.0:
Das Programm sperrt Dateien gegen versehentliches löschen und wird durch cmode s dateiname für Sperren, bzw. cmode f dateiname für Freigabe einer Datei aufgerufen.

```
program filechmode;
type
  registers = record
    ax, bx, cx, dx, bp, si, di, ds, es, flags:integer;
  end;

var
  fn   : string!64!;     -> ! = eckige Klammer
```



```
  mode : string!2!;
  regs : registers;
  atr  : byte;

begin
  if paramcount <> 2 then begin
    write('Dateiname: ');
    readln(fn);
    write('Mode F(rei S(perren: ');
    readln(mode);
   end

   else begin
     mode := paramstr(1);
     fn   := paramstr(2);
end;

case mode!1! of
  'S','s' : atr := 1;
  'F','f' : atr := 0;
end;

fn := fn + chr(0);
with regs do begin
  ax := $4301;
  cx := atr;
  ds := seg(fn);
  dx := ofs(fn) + 1;
  msdos(regs);
  if (flags shl 15) shr 15 = 1 then begin
    write('Error ',ax);
    case ax of
      3 : writeln(': Datei existiert nicht');
      5 : writeln(': Kein gültiger Dateiname');
    end;
    halt;
  end;
  writeln('Done');
end;
end.
```

Alles in allem representiert das Maschinchen eine beachtliche Ingenieur-Leistung.

Egbert Schröer, April 1992

7. Literaturnachweis
1. Portfolio-Bedienungshandbücher (Portfolio-Grundgerät, RS232c-Erweiterung, parallelschnittstelle, Speichererweiterung)
2. Systemhandbuch zum ATARI-Portfolio, Fischel GmbH 1991, S. Gad
3. ct Juli 1990, S.114, "Ungleiche Brüder"
4. Happy Computer Sept. 1989, S. 116, "Der kleinste PC der Welt"

ZEUS spricht Intel-Hex

"ZEUS 1.Ø,(c) 1983 C.E.C" erscheint auf dem Bildschirm, wenn ich meinen Editor-Assembler auf dem Model 4P aufrufe. Neun Jahre ist das gute Stück Software also schon alt. An der Lebensdauer heutiger Programme gemessen ist er ein Relikt aus der Computer-Steinzeit. Doch das Werkzeug, das ich schon viele Jahre benutze, ist keineswegs stumpf geworden. Im Gegenteil: Mit einem Assemblerprogramm habe ich dem ZEUS neue Anwendungen erschlossen, die ihn über seinen ursprünglich geplanten Einsatz hinaus zum Universalwerkzeug für die Z8Ø-Programmentwicklung werden lassen.

Lohnt es sich heute in der Zeit der High Performance Chips noch, den Z8Ø Mikroprozessor für Neuentwicklungen zu verwenden? Schon ein Blick auf die Verkaufszahlen beantwortet diese Frage. Zahlenmäßig dominieren die 8-Bit-Prozessoren nach wie vor den Markt. Etwa die Hälfte von ihnen basiert auf dem Z8Ø-Befehlssatz. Wegen seiner geradlinigen I/O-und Interruptstruktur, sowie der großen vorhandenen Programmiererfahrung gehört der Z8Ø und seine Nachfolgetypen für Anwendungen in der Prozeßtechnik und Datenkommunikation immer noch zur ersten Wahl.

Natürlich gibt es inzwischen Assembler für den Z8Ø, die auf MSDOS-Rechnern laufen. Aber versucht einmal, eine Kombination von Editor und Assembler zu finden, die ähnlich leistungsfähig und leicht zu bedienen wie der ZEUS ist. Bis jetzt kenne ich nur ein Produkt, das Gleiches (und mehr) leistet. Es ist der ECAL von Maxim Technology, für den man jedoch über DM 600,- anlegen muß.

Der ZEUS war konzipiert als eine Art verbesserter EDTASM, um Programme zu entwickeln, die auf dem TRS8Ø laufen. Er erzeugt /CMD-Files auf Diskette, die man dann von der DOS-Ebene aus starten kann. Nicht vorgesehen ist der Fall, daß der TRS8Ø gar nicht das Zielsystem für die selbst entwickelte Software ist, sondern ein anderes Gerät, das über die serielle Schnittstelle an den TRS8Ø angeschlossen ist. Da der ZEUS den Z8Ø-Code mit jeder beliebigen Startadresse erzeugt, braucht man nur noch ein Hilfsprogramm, das den auf der Diskette vorhandenen Objectcode umformatiert und über die Schnittstelle überträgt.

Für die serielle Übertragung von Binärdaten ist das am häufigsten angewendete Format der Intel-Hex-Code. Die Erklärung dazu habe ich der Einfachheit halber aus der Zeitschrift c't auskopiert (siehe Information im Kasten). Mein Entwicklungsziel war es, die Objectfiles des ZEUS so umzucodieren, daß sie der Intel-Hex-Norm entsprechen und über die serielle Schnittstelle ausgegeben werden können. Das ist nicht ganz einfach, denn ZEUS und EDTASM lassen Blocklängen bis zu 256 Bytes zu, während Intel-Hex mit Blocklängen von 64 Zeichen oder weniger arbeitet. An den passenden Stellen müssen Adressmarken und Prüfsummen eingesetzt werden. Die Versuche fingen zunächst mit einem Basicprogramm an. Das lief zwar im Prinzip, war aber viel zu langsam. Erst mit dem Assemblerprogramm konnte ich die im TRS8Ø mögliche Baudrate von 96ØØ nutzen.

Der Intel-Hex-Ausgang aus dem TRS-8Ø ist wie ein Schlüssel zu neuen Möglichkeiten. Auf einmal ist der gute alte ZEUS wieder up to date. Man kann mit ihm Objectcodes in EPROM-Programmiergeräte oder EPROM-Simulatoren laden, man kann Programme ins RAM von Single-Board-Computern hinunterladen oder per Modem zu anderen Computern übertragen. Auch die modernsten Entwicklungssysteme schicken genau den gleichen Intel-Hex-Code über ihre seriellen Schnittstellen.

Die Gebrauchsanweisung für DLHEX/CMD:

1) Das Programm assemblieren z.B. mit der ZEUS-Anweisung ANO

2) Als Namen des Objectfiles immer OBJ eingeben. Nur diesen Namen findet und bearbeitet DLHEX/CMD.

3) Den ZEUS verlassen mit dem Dreifingergriff Q-Enter-Clear

4) DLHEX aufrufen. Man sieht am Bildschirm die gleichen Daten, die über die serielle Schnittstelle gesendet werden.

5) DLHEX/CMD führt nach beendetem Programmlauf zu NEWDOS oder startet wahlweise gleich wieder den ZEUS.

Heinrich Betz

© Heinz Langer

```
Marke | Anzahl Bytes | Adresse | Typ | Daten                          | Prüf-summe
:10200000 10CE80007F180386FFB71802860 4B718 29
:10201000034FB718028E0000301F26FC43B718028A
:092020008E0000301F26FC20E8B0
:0000000000
```

Intel-Hex-Format

Das Intel-Hex-Format ist eine Kodierungsvorschrift für beliebige byteorientierte, adreßbezogene Daten, wie sie etwa Maschinenprogramme für 8-Bit-Mikroprozessoren darstellen. Zur Kodierung der Daten werden ausschließlich Textzeichen verwendet (ASCII), wobei der Wert eines Bytes, der üblichen Schreibweise entsprechend, in zwei Hex-Ziffern (0..9, A..F) erscheint.

Der Hex-Ziffern-Strom der umgesetzten Daten ist nun aber nicht 'endlos', sondern in Blöcke unterteilt, die jeweils mit der Steuercodefolge CR/LF enden, so daß eine Datei in diesem Format wie eine normale Textdatei behandelt werden kann. Dies betrifft insbesondere die Übertragung über Schnittstellen, die keine 8-Bit-Transfers erlauben (vornehmlich serielle Schnittstellen).

Jeder Block stellt eine in sich geschlossene, mit einem Vorspann und einer Prüfsumme versehene Informationseinheit dar, was bei Übertragungsfehlern eine relativ genaue Aussage über den Fehlerort zuläßt beziehungsweise nicht gleich die Wiederholung der kompletten Sendung verlangt. Das erste Zeichen des Vorspanns ist ein Doppelpunkt, gefolgt von zwei Hex-Ziffern (= ein Byte), die die hexadezimale Anzahl Datenbytes im jeweiligen Block verkörpern. Die nächsten vier Hex-Ziffern bilden die (absolute) 16-Bit-Adresse, unter der das erste Datenbyte des Blockes im Speicher abgelegt werden soll. Den Vorspann beschließt ein Byte, dessen Wert den Typ des Blockes angibt: 0 = Datenblock, 1 = Endblock. Auf diese Unterscheidung kann jedoch verzichtet werden, wenn sich ein Endblock auch durch eine Blocklänge gleich Null eindeutig kennzeichnen läßt. (So verfahren die meisten Assembler unter CP/M, auch der XASM09; das Typbyte ist dann immer Null).

Nach den Datenbytes, als letztes vor dem Block- oder Zeilenende (CR/LF), steht die 8-Bit-Prüfsumme, die als negativer Wert der Summe aller übrigen Bytes des Blocks gebildet wird. Oder anders ausgedrückt: Die Summe aller Bytes eines Blocks (einschließlich der Prüfsumme) beträgt Null. Durch die Addition auftretende Überträge bleiben dabei unberücksichtigt.


```
                 ØØØØ1 ;DLHEX/SRC       29.Ø8.92            Heinrich Betz
                 ØØØØ2
                 ØØØØ3 ;Das Programm speichert das File OBJ/CMD ins RAM (OBJBUF)
                 ØØØØ4 ;und laedt es von dort ueber die serielle Schnittstelle
                 ØØØØ5 ;im Intel-Hex-Code zum EPROM-Programmer oder zum SBC.
                 ØØØØ6 ;Schnittstellen-Parameter: siehe Zeilen 14 bis 19.
                 ØØØØ7
4424             ØØØØ8 OPEN     EQU     4424H              ;Open file
4436             ØØØØ9 READ     EQU     4436H              ;Read one record
4428             ØØØ1Ø CLOSE    EQU     4428H              ;Close file
44Ø9             ØØØ11 ERRDSP   EQU     44Ø9H              ;Display DOS errors
                 ØØØ12
52ØØ             ØØØ13          ORG     52ØØH
                 ØØØ14 ;Baudraten: Ø=5ØBd,;17=75Bd;34=11ØBd;52=15ØBd;85=3ØØBd
                 ØØØ15 ;1Ø2=6ØØBd;119=12ØØBd;17Ø=24ØØBd;2Ø4=48ØØBd;238=96ØØBd
52ØØ  3EAA       ØØØ16 START    LD      A,17Ø              ;Baudrate..
52Ø2  D3E9       ØØØ17          OUT     (233),A            ;waehlen
52Ø4  3EA5       ØØØ18          LD      A,165              ;ev. Prty, 1 Stopbit
52Ø6  D3EA       ØØØ19          OUT     (234),A            ;7 Datenbits, DTR=true
52Ø8  3EØF       ØØØ2Ø          LD      A,15               ;Master Reset..
52ØA  D3E8       ØØØ21          OUT     (232),A            ;UART
52ØC  216654     ØØØ22          LD      HL,OBJBUF          ;ab OBJBUF den..
52ØF  226154     ØØØ23          LD      (PTSTOR),HL        ;Objcode abspeichern
5212  21Ø553     ØØØ24          LD      HL,MSOBJ           ;Text 'OBJ/CMD'
5215  114153     ØØØ25          LD      DE,FCB
5218  Ø1Ø8ØØ     ØØØ26          LD      BC,MSOEND-MSOBJ
521B  EDBØ       ØØØ27          LDIR                       ;File Name -> FCB
521D  216153     ØØØ28          LD      HL,FILBUF
522Ø  114153     ØØØ29          LD      DE,FCB
5223  Ø6ØØ       ØØØ3Ø          LD      B,Ø
5225  CD2444     ØØØ31          CALL    OPEN               ;Open File
5228  C2EE52     ØØØ32          JP      NZ,SYNERR          ;evtl. Error anzeigen
522B  114153     ØØØ33 RDSEC    LD      DE,FCB
522E  CD3644     ØØØ34          CALL    READ               ;read sector
5231  C24652     ØØØ35          JP      NZ,RDERR
5234  216153     ØØØ36          LD      HL,FILBUF
5237  ED5B6154   ØØØ37          LD      DE,(PTSTOR)
523B  Ø1ØØØ1     ØØØ38          LD      BC,256
523E  EDBØ       ØØØ39          LDIR                       ;store -> RAM
524Ø  ED536154   ØØØ4Ø          LD      (PTSTOR),DE
5244  18E5       ØØØ41          JR      RDSEC
                 ØØØ42
5246  FE1C       ØØØ43 RDERR    CP      1CH                ;Check for EOF
5248  CA5Ø52     ØØØ44          JP      Z,FILEND           ;EOF=File gelesen
524B  FE1D       ØØØ45          CP      1DH                ;Check if past end
524D  C2E952     ØØØ46          JP      NZ,ERROR
525Ø  114153     ØØØ47 FILEND   LD      DE,FCB
5253  CD2844     ØØØ48          CALL    CLOSE              ;Close File
5256  C2Ø944     ØØØ49          JP      NZ,ERRDSP          ;Display error, goto DOS
                 ØØØ5Ø ;.........................................................
                 ØØØ51 ;Programmteil fuer Download -> SBC
5259  CDC9Ø1     ØØØ52          CALL    Ø1C9H              ;CLS
525C  216654     ØØØ53          LD      HL,OBJBUF          ;Anfang Obj.-File
525F  7E         ØØØ54 EPLOOP   LD      A,(HL)             ;1.Byte=Satzkennz.
526Ø  FEØ1       ØØØ55          CP      Ø1                 ;Ø1=Normaler Datensatz
5262  C2F352     ØØØ56          JP      NZ,ENDE            ;Ø2=End-Datensatz
5265  23         ØØØ57          INC     HL                 ;HL pt ZEUS-Bytezahl
5266  7E         ØØØ58          LD      A,(HL)             ;Zahl der Bytes
5267  3D         ØØØ59          DEC     A                  ;Zeus hat 2 zu viel,
5268  3D         ØØØ6Ø          DEC     A                  ;(zaehlt Startadr. mit)
5269  F5         ØØØ61          PUSH    AF
526A  D1         ØØØ62          POP     DE                 ;Flag in E
526B  326554     ØØØ63          LD      (GESBYT),A
526E  23         ØØØ64          INC     HL                 ;HL pt LSB der Startadr.
526F  7E         ØØØ65          LD      A,(HL)
527Ø  326354     ØØØ66          LD      (STARTA),A
5273  23         ØØØ67          INC     HL                 ;HL pt MSB der Startadr.
```

```
5274  7E                    LD    A,(HL)
5275  326454                LD    (STARTA+1),A
5278  3E3A      STMARK      LD    A,':'             ;Startmarke..
527A  CD3353                CALL  SIOOUT            ;ausgeben
527D  CB73                  BIT   6,E               ;Z-Flag von vorher
527F  CBB3                  RES   6,E
5281  2ØØ8                  JR    NZ,BYT256         ;256 Bytes?
5283  3A6554                LD    A,(GESBYT)
5286  FE21                  CP    33                ;<= 32 Bytes?
5288  DA9F52                JP    C,KLGL32
528B  3E2Ø      BYT256      LD    A,32              ;Bytezahl=32..
528D  47                    LD    B,A
528E  CD1E53                CALL  BIN2HX            ;ausgeben
5291  3A6554                LD    A,(GESBYT)
5294  D62Ø                  SUB   32
5296  326554                LD    (GESBYT),A
5299  CDAD52                CALL  SEND
529C  C37852                JP    STMARK
529F  3A6554    KLGL32      LD    A,(GESBYT)        ;Zahl der restl. Bytes
52A2  47                    LD    B,A               ;fuer DJNZ -> B..
52A3  CD1E53                CALL  BIN2HX            ;und ausgeben
52A6  CDAD52                CALL  SEND
52A9  23                    INC   HL
52AA  C35F52                JP    EPLOOP
52AD  3A6454    SEND        LD    A,(STARTA+1)      ;MSB der Startadr.
52BØ  F5                    PUSH  AF
52B1  8Ø                    ADD   A,B               ;Checksu. bilden
52B2  4F                    LD    C,A               ;C=Checksu.-Speicher
52B3  F1                    POP   AF
52B4  CD1E53                CALL  BIN2HX            ;in Hex-ASCII ausgeben
52B7  3A6354                LD    A,(STARTA)        ;LSB der Startadr.
52BA  F5                    PUSH  AF
52BB  81                    ADD   A,C               ;Checksu. aufaddieren
52BC  4F                    LD    C,A
52BD  F1                    POP   AF
52BE  CD1E53                CALL  BIN2HX            ;LSB Startadr. ausgeben
52C1  AF                    XOR   A                 ;Record Type (ØØ)..
52C2  CD1E53                CALL  BIN2HX            ;ausgeben
52C5  AF                    XOR   A
52C6  81        SATZLP      ADD   A,C               ;Checksu. aufaddieren
52C7  4F                    LD    C,A
52C8  23                    INC   HL                ;HL pt Datenbyte
52C9  7E                    LD    A,(HL)            ;Daten..
52CA  F5                    PUSH  AF
52CB  CD1E53                CALL  BIN2HX            ;senden
52CE  F1                    POP   AF
52CF  DD216354              LD    IX,STARTA
52D3  DD34ØØ                INC   (IX)
52D6  2ØØ3                  JR    NZ,NUR1X          ;Uebertrag?
52D8  DD34Ø1                INC   (IX+1)            ;Hi Byte erhoehen
52DB  1ØE9      NUR1X       DJNZ  SATZLP
52DD  81                    ADD   A,C               ;Checksu. bilden..
52DE  ED44                  NEG                     ;und..
52EØ  CD1E53                CALL  BIN2HX            ;ausgeben
52E3  3EØD                  LD    A,ØDH             ;CR
52E5  CD3353                CALL  SIOOUT
52E8  C9                    RET
```



```
52E9  F5        ERROR   PUSH  AF
52EA  CD2844            CALL  CLOSE
52ED  F1                POP   AF
52EE  F64Ø      SYNERR  OR    4ØH
52FØ  C3Ø944            JP    ERRDSP
52F3  Ø6ØC      ENDE    LD    B,12              ;Laenge v. EMSG
52F5  21ØD53            LD    HL,EMSG
52F8  7E        E1      LD    A,(HL)
52F9  CD3353            CALL  SIOOUT
52FC  23                INC   HL
52FD  1ØF9              DJNZ  E1
52FF  211953            LD    HL,ZMSG           ;ZEUS aufrufen mit..
53Ø2  C3Ø544            JP    44Ø5H             ;DOS Einsprung
                ;..............................................
53Ø5  4F        MSOBJ   DM    'OBJ/CMD',ØDH
ØØØØ            MSOEND  DS    Ø
53ØD  3A        EMSG    DM    ':ØØØØØØ1FF',ØDH
5319  5A        ZMSG    DM    'ZEUS',ØDH
                ;BIN2HX, gibt das Byte in A als zwei ASCII-Zeichen
                ;'ØØ'..'FF' ueber die serielle Schnittstelle und den
                ;Bildschirm des TRS8Ø aus. Zerstoert den Inhalt von A.
                ;Beispiel: '?' (3FH) wird als '3F' (33H,46H) ausgegeben.
531E  F5        BIN2HX  PUSH  AF
531F  E6FØ              AND   ØFØH
5321  ØF                RRCA
5322  ØF                RRCA
5323  ØF                RRCA
5324  ØF                RRCA
5325  CD2B53            CALL  NASCII
5328  F1                POP   AF
5329  E6ØF              AND   ØFH
532B  FEØA      NASCII  CP    1Ø
532D  38Ø2              JR    C,NAS1
532F  C6Ø7              ADD   A,7
5331  C63Ø      NAS1    ADD   A,'Ø'
5333  F5        SIOOUT  PUSH  AF
5334  DBEA      RSLP    IN    A,(ØEAH)
5336  CB77              BIT   6,A
5338  28FA              JR    Z,RSLP
533A  F1                POP   AF
533B  D3EB              OUT   (ØEBH),A          ;Zch. ausgeben
533D  CD3AØ3            CALL  Ø33AH             ;darstellen
534Ø  C9                RET
                ;Variable:
ØØ2Ø            FCB     DS    32                ;File Control Block
Ø1ØØ            FILBUF  DS    256               ;File Buffer
ØØØ2            PTSTOR  DS    2                 ;Pointer ->RAM
ØØØ2            STARTA  DS    2                 ;Startadr. f. Intel-Hex
ØØØ1            GESBYT  DS    1                 ;Gesamt-Bytezaehler
ØØØØ            OBJBUF  DS    Ø                 ;Hier beginnt der Buffer
52ØØ                    END   START

ØØØØØ Fehler
```

Aufbacken der harten Scheiben.

Wer ein Winchesterlaufwerk in seinem Computer eingebaut hat, sollte ab und zu diesselbe 'backup'-pen. Ich habe auf meinem DOS-Laufwerk 5 zum Beispiel bei der Installation ein defektes Directory angelegt. Nach dem ich gerade alle meine Lieblingsprogramme auf der Scheibe installiert hatte, und eines Abends noch zwei Programme dazutun wollte, meldete der Rechner beim Kopieren plötzlich 'Lesefehler Inhaltsverzeichnis'. Eins der beiden Programme wurde auf die ersten beiden Sektoren des Inhaltsverzeichnissses kopiert und mein Gesicht wurde länger und länger.

Falls mir so etwas noch einmal passieren sollte, will ich gewappnet sein. Das Maschinenprogramm 'BACKUP/CMD' kopiert alle Files, die sich auf der Festplatte befinden, auf 5 Disketten und löscht dann die Bearbeitungskennzeichen der Files auf der Harddisk. Nur die Files, die seit dem letzten Backup neu auf die Platte gekommen sind und solche, die über DOS-Routinen verändert oder erweitert wurden, werden beim nächsten Backup übertragen.

Der 'COPY'-Befehl benutzt hierbei den Parameter 'BEA' für 'Bearbeitungskennzeichen gesetzt' und kopiert eben gerade nur solche Files, bei denen dieses Flag gesetzt ist. Das BK wird allerdings nicht gesetzt, wenn man mit SUPERZAP zum Beispiel sektorweise auf einen File, man müßte hier von Sektor und nicht von File reden, zugegriffen wird.

Wird 'BACKUP <ENTER>' eingegeben, wird von allen 5 Laufwerken ein Backup auf Diskette erstellt, das Programm fordert dann zu 5 Diskettenwechseln auf. Falls so eine Sicherungsdiskette einmalvoll ist, schließlich fassen Festplattenlaufwerke die mit dem 'HDSYSOEA/SRC' von Andreas Magnus betan werden immerhin 1.9MByte, muß man eben eine mehere Backup-Disketten benutzen. Man kann auch nur ein Laufwerk sichern, indem, man die betreffende Laufwerksnummer mit angibt.

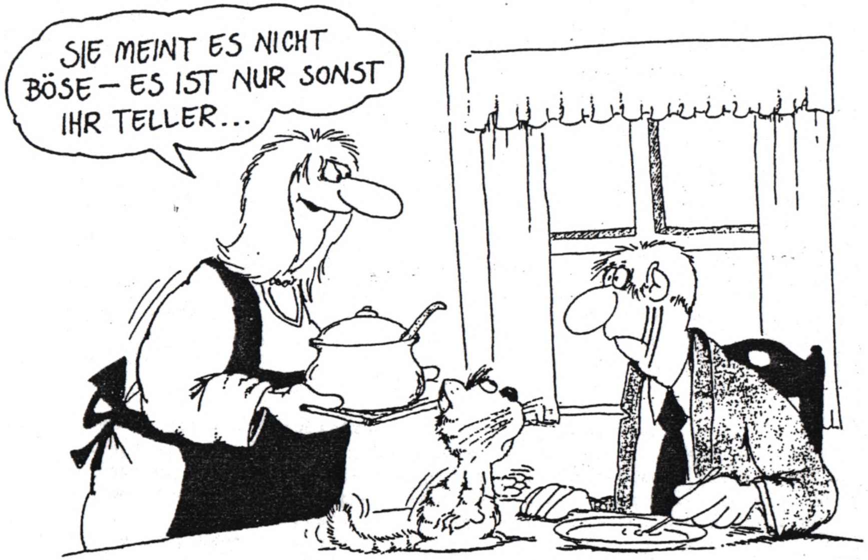
Das Programm ist eigentlich ziemlich einfach, es können sehr viele DOS-Calls benutzt werden. Man könnte dieses Backupprogramm übrigens auch sehr leicht als JOB-File schreiben. Die Startadresse ist 3000h, weil dieser Speicherbereich beim Kopieren garantiert nicht überschrieben wird. Soviel zum Thema Datensicherheit beim Betrieb von Festplattenlaufwerken.

Ein weiterer Vorteil ist noch, daß durch regelmäßiges Sichern zum einen ältere Versionen von Dateien erhalten bleiben und man kann relativ gefahrlos Files löschen wobei man eben sicher sein kann, diese File einfach wiederzufinden, wenn man der Meinung ist, dieses File ist denn nun doch nicht so unnütz wie früher vermutet.

Volker Born


```
00001 ;        BACKUP/CMD
00002 ;
00003 ;
00004 ; dieses Programm soll von der Festplatte Backup's erstellen.
00005 ; Es tut dies indem nur Files kopiert werden, deren Bearbeitungs-
00006 ; kennzeichen gesetzt ist. Dies ist der Fall, wenn ein File neu
00007 ; in ein Directory eingetragen wurde oder wenn das File verändert
00008 ; wurde. Man wird aufgefordert, die Backup-Disketten 5 bis 9
00009 ; nacheinander einzustecken.
00010 ;
00011 ;
00012 ;
00013 gibaus   EQU    4467h
00014 dosrdy   EQU    402dh
00015 doscal   EQU    4419h
00016 doserr   EQU    4409h
00017 datum    EQU    4470h
00018 ;
00019 ;
00020 ;
00021          ORG    3000h
00022 start    LD     A,(HL)
00023          CP     0dh      ; nur <ENTER> ohne Parameter ?
00024          JR     Z,alle
00025          SUB    '5'               ; gültige LW# sind 5,6,7,8,9
00026          JR     C,error
00027          CP     10
00028          JR     C,eintrag
00029 error    LD     A,32
00030          OR     A
00031          JP     doserr
00032 eintrag  ADD    A,'5'    ; restauriert die in Z.25 ausgef. Subtraktion
00033          LD     (cpnum),A
00034          LD     (prnum),A
00035          LD     (lgnum),A
00036          LD     (afnum),A
00037          CALL   backup
00038          JR     ende
00039 alle     LD     BC,0500h          ;5 HD-laufwerke
00040 loop     PUSH   BC                ; BC wegen Registerveränderungen pushen
00041          LD     A,(speich)
00042          INC    A                 ; a => '5', usw.
00043          LD     (speich),A        ; zwischenspeichern
00044          LD     (cpnum),A
00045          LD     (prnum),A
00046          LD     (lgnum),A
00047          LD     (afnum),A
00048          CALL   backup
00049          POP    BC                ; BC vom Stack, schön warm und trocken
00050          DJNZ   loop
00051 ende     LD     HL,endtex
00052          CALL   gibaus
00053          JP     dosrdy
00054 endtex   DM     'Der Backupvorgang ist abgeschlossen.',0dh
00055 ;
00056 backup   LD     HL,auffor
00057          CALL   gibaus
00058 loopi    LD     A,(3840h)
00059          CP     01h               ; <ENTER> gedrückt??
00060          JR     NZ,loop1
00061          LD     HL,losgeh
00062          CALL   gibaus
00063          LD     HL,copy
```

```
        CALL    doscal
        CALL    NZ,doserr
        LD      HL,prot
        CALL    doscal
        CALL    NZ,doserr
        LD      HL,buff
        CALL    datum   ; datum trägt das aktuelle Datum in 'buff' ein
        LD      HL,prot0              ; neuestes Datum auf Disk 0
        CALL    doscal
        CALL    NZ,doserr
        RET
;
auffor  DM      07h,20h,0bh,'Lege Backup-Disk für HD LW# '
afnum   DM      '5'
        DM      ' ein. Dann <ENTER>.',0dh
losgeh  DM      'Backup von Festplattenlaufwerk '
lgnum   DM      '5'
        DM      '.',0dh
copy    DM      '> '
cpnum   DM      '5'
        DM      ',0,,nfmt,kdwa,edk,bea',0dh
prot    DM      'prot :'
prnum   DB      '5'
        DM      ',bkl',0dh
prot0   DM      'prot :0,datum='
buff    DM      '00.00.00',0dh
speich  DM      '4'
        END     start
```



Es lebe der Fortschritt.

Als glücklicher Besitzer eines Genie IIIs (übrigens jetzt auch mit Harddisk) genieße ich es sehr, auf dem Bildschirm bis zu 2000 Zeichen im Format 80x25 darstellen zu können. In diesem Bildschirmmodus befindet sich der Rechner fast immer, wenn ich mit ihm herumspiele.

Leider sind die meisten Programmme für diese Rechnergeneration original für den TRS-80 bzw. VideoGenie geschrieben, der ja bekanntlich nur 64x16 Zeichen anzeigt, so auch das Textverarbeitungsprogramm, mit dem dieser Artikel entstand. Wenn ich also so ein älteres Programm nutzen will, muß ich beim GIIIs zunächst den DOS-Befehl '64' eingeben und dann das Programm starten. Habe ich genug von diesem oder jenem Programm, so mußte ich nach Beendigung so einer Session den Dos-Befehl '80' eingeben. Diese beiden Befehle hab ich mir auf die ganz linke und ganz rechte Funktionstaste gelegt, dies beschleunuigt den Umschaltvorgang schon ganz schön. Aber das genügte mir nicht.

Ich entsann mich der Organisation von /CMD-Programmen und dem DOS-Call 0033h, der ein Zeichen auf den Bildschirm bringt. Wenn man das Zeichen 10h oder 16d über diese Routine ausgibt, wird der Videocontrollerbaustein umprogrammiert und ein Clear-Screen wird ausgegeben. Danach erfolgt die Bildschirmausgabe im 64x16 Zeichen-Modus. Um den 80x25 Zeichen-Modus einzuschalten, muß das Zeichen 13h (19d) ausgegeben werden.

/CMD-Programme sind folgendermaßen auf Diskette abgespeichert: Das erste Byte ist immer 01h. Dann folgt die Angabe über die Menge der in den Speicher zu ladenden Zeichen plus zwei und dann die Adresse, an der die Zeichen in den Speicher geladen werden sollen. Ein Zähler wird mit dem zweiten Byte geladen, und wenn er auf 0 runtergezählt wurde,zeigt der Zeiger wieder auf so eine 01, was eine neue Menge von Zeichen bedeutet. Ist diese Zeichen aber 02, ist das nächste auch 02. Dies ist die Ende-Kennung für dieses File, danach folgen noch zwei Bytes, die die Startadresse des Programms angeben.

Es gibt noch den Sonderfall, das /CMD-Files zunächst oder auch mitten in der Datei Kommentare enthalten, dann ist dieses erwähnte Zeichen welches übrigens Kennbyte des Object Code Records genannt wird auch einmal 00 oder sogar 03 bis 0FFh. Das ist allerdings relativ selten.

Ich habe nun bei einigen mir wichtigen Programmen dieselben so geändert, daß vor dem Programmstart zunächst auf 64x16 Zeichen geschaltet wird und bevor dieselben wieder den Ausgang über DOS-Ready (DOS-Call 402Dh ist hier ziemlich oft verwendet) finden, wird wieder auf 80x25- Zeichen zurückgeschaltet. Der zweite Teil funktioniert nicht bei Programmen, die als Unterprogramm programmiert sind, die enden nämlich mit einem 'RET'. Um so eine Umschaltung in die Files einzuZAPen, benutze man am besten den 'FED' oder noch besser das Programm 'DED', was dieselben Funktionen wie 'FED' kann, allerdings auch Systemfiles bearbeitet.

In ihm wird mit der Funktion 'Z', was das Aufsuchen des nächsten Object Record Headers bedeutet, so lange, bis man den Start Header mit der Kennung 02 02 <startadresse> gefunden hat. Wenn man diese äußerst praktischem Programme nicht hat, muß man halt von Hand den letzten Sektor nach der Endekennung duchsuchen.

Die Umschaltung auf 64x16 in Assembler:

```
3E 10          64x16  LD     A,10h
CD 33 00              CALL   0033h
C3 XX XX              JP     <startadresse>
```

Das ist schon fast alles. Anstatt der Ende Kennung 02 02 usw. muß nun dieses kleine Programm eingezapt werden, erst muß jedoch die erste 02 in 01 umgewandelt werden (nicht Endekennung) und als zweites Byte muß die Anzahl der Objektbytes plus zwei (damit der Zeiger auf dem nächsten Kennbyte steht) eingetragen werden. Es sind also 8 + 2 Bytes, was hexadezimal 0Ah heisst. Jetzt muß die Ladeadresse der kleinen Routine kommen, sie kann entweder nach dem eigentlichen Programm stehen' oder am Ende des Eingabepuffer ab 4318h oder im Prinzip sonstwo. Die 8 Bytes können ja getrost überschrieben werden. Natürlich darf die Routine nicht im Code des eigentlichen Programmmes liegen. Nehmen wir an, das Programm an dem der Zusatz angehängt werden soll, liegt von 5200h bis 5400h. Unser Zusatz kann also zum Beispiel bei A000h liegen, einzutragen ist folgender Code :

```
01 0A  00 A0    3E 10  CD 33 00  C3 00 52
```

Falls man im Besitz des FED oder DED ist, sollte man sich anschauen, wo das letzte Byte des Programms geladen werden soll, und die Routinen ab der nächsten Speicherstelle positionieren. Jetzt soll nach dem Ausstieg des Programms wieder auf 80x25 zurückgeschaltet werden. Das ist etwas komplizierter. Es muß zunächst der Ausgang des Programmes gefunden werden. Hierzu gibt man bei 'FED' folgende Sequenz ein

```
F       - für FIND HEX
Od 42   - für 420dh im Z80-Format, also erst MSB dann LSB
```

Der Cursor sollte dann auf die Bytes OD 42 zeigen, falls sich das Programm auf diesem Wege verabschiedet. Vor dem OD 42 muß ein 'C3' stehen, es heißt 'JUMP' und stellt sicher, das man nicht den Code in einer Tabelle gefunden, sondern den tatsächlichen Ausgang. Es können auch mehrer solcher Sprünge nach 402Dh im Programm sein.Hier muß jetzt ein Sprung zu einer Routinen stehen, die zunächst auf 80x25 Zeichen schaltet, und erst dann nach DOS-Ready springt. Sie sieht ähnlich aus wie '64x16', bloss daß das Zeichen, mit dem der Accu geladen wird, ein 13h ist.

Diese Routine darf aber nicht irgendwo stehen, da das Rumpfprogramm diese Stelle ja überschreiben könnte, sondern sie muß an einem sicheren Plätzchen verborgen bleiben. Ich würde das Ende des DOS-Eingabepuffers empfehlen, er ist schließlich 80 Zeichen lang, und so lange DOS-Befehle wird man wohl kaum in einem Programm eingeben. Im folgenden also 80x25:

```
4338  3E 13       LD    A,13h
433A  CD 33 00    CALL    0033h
433D  C3 2D 40    JP      402Dh
```


Dieser Code sieht in der Datei dann folgendermaßen aus

```
01 0A 38 43  3E 13 CD 33 00 C3 2D 40
```

Und jetzt zum Schluß die Endekennung :

```
02 02 00 A0
```

, damit zunächst das dort liegende '64x16' abgearbeitet wird.

Schon fertig.
Aber ich denke die Arbeit lohnt sich. Falls dieses Späsken auf einem anderen Rechner laufen soll, muß erst das Handbuch zu Rate gezogen werden, um die Codes für die Bildschirmumschaltung in Erfahrung zu bringen.

Bei einigen Programmen sind Variationen denkbar, der 'DED' zum Beispiel läßt mitten im Programmtext einige Bytes für Erweiterungen frei, das '80x25' oder sogar auch '64x16' läßt sich also dort unterbringen. Das wärs für heute abend

Happy Hacking

Nie mehr das verfluchte DOS-Handbuch suchen.

Es gibt für das Betriebssystem NewDOS-80 V.2 ein Programm namens 'HELP/CMD' von Matt Wakeley, das zu jeden DOS-Befehl einen mehr oder weniger aussagekräftigen Erläuterungstext auf den Bildschirm bringt. Dieses Programm ist genial einfach programmiert. Der Maschinencode selbst ist nur knapp 400 Bytes lang. Die nächsten zwei Sektoren enthalten eine Liste der DOS-Befehle, jeder Befehl hat maximal 6 Zeichen und danach folgen immer zwei Zeichen, deren Funktion wir später kriegen.

Diese Liste hat als Endekennung ein 00h, und sie kann im Prinzip auch länger als zwei Sektoren sein. Nach den Sektoren mit der Liste folgen mehere Sektoren mit den 'HELP'-Texten, sie sind getrennt durch ein '00h'. Die jeweiligen Texte sind folgendermaßen strukturiert :

Zunächst kommt die Beschreibung des Befehls, dieser Text wird nach der Wiederholung des gesuchten Begriffs angezeigt. Dann folgen Textzeilen, die mit einem Zeichen größer C0h (größer 191) beginnen. Diese Codes stehen für eine bestimmte Menge Leerzeichen, C5h steht für 5 Spaces, das spart halt Platz. Abgeschlossen werden die einzelnen Zeilen mit 0Dh, ENTER. Somit können sie über den DOS-Call 4467h (gibt Zeichenketten bis zur Endekennung 03h oder 0Dh aus) bequem auf den Bildschirm gebracht werden. Nach der letzten Textzeile steht dann auch '00h'. In der ersten Zeile steht immer derBefehl mit allen Parametern.

So ein Hilfe-File wollte ich für G-DOS schon immer haben, bei Befehlen die ich selten benutze, muß ich sonst halt immer das Handbuch rauskramen, und das gibt ja auch nicht immer alles her, was man sich so wünscht. Ich habe mit TSCRIPS also für jeden Befehl aufgeschrieben, was mir wichtig und notwendig erschien. Um Platz zu sparen, habe ich vor jede Zeile, die mit so einem Tabulatorspezialzeichen (CXh) anfangen sollte, einen Stern '*' gemacht. Zwischen den einzelnen Texten habe ich immer ein 0Dh, Carriage Return, gelassen. Als der Text endlich fertig war, schrieb ich mir in Basicprogramm, welches solche 0Dh's in 00h und jeden Stern in 'C5h' umwandelte. Mit 'CREATE DUMMI/CMD,ANZ=5' hatte ich mit ein File erzeugt, welcher genau so groß war, wie der Maschinendcode von 'HELP/CMD' plus 3 Sektoren für die Befehlstabelle, da mein CalvaDOS mehr als 64 Befehle ausführen kann. Mit Superzap kopierte ich die ersten 5 Sektoren von 'HELP/CMD' nach 'DUMMI/CMD' und hängte die Textdatei 'HILF/TXT' mit 'APPEND HILF/TXT DUMMI/CMD' an den Rumpf meines im werden begriffenen 'HILF/CMD'. Die Tabelle ab Sektor 2 im 'HELP/CMD' ist nämlich genial einfach aufgebaut : Nach dem Befehlswort folgen 2 Bytes, die angeben in welchem Sektor ab welchem Byte der gewünschte Hilfe-Text steht.

'HELP/CMD' liesst dann den angegebenen und die folgendenSektoren und gibt die Textzeilen aus, bis die Endekennung 00 erreicht ist. Die Schwierigkeit war dann eben genau dies herauszufinden und in die ersten beiden Sektoren hereinzuZAPen. Aber dank 'FED' war es dann doch relativ schnell getan. Das Programm ließe sich auch noch erweitern, wenn man kleine Hilfen zu Programmen oder eine Liste der Controlcodes für den Videocontroller ab und zu mal braucht. Man schreibt sich den Text, ändert die Sterne in 'C5h', erzeugt ncoh die Endeklennung und hängt das File an 'HILF/CMD' heran. Dann muß nur ncoh der Suchbegriff in der Tabelle und die Position des Hilfstextes im File eingetragen werden. Eine praktische Sache, finde ich. Als Kostprobe hab ich den Text zum Befehl 'DR' ausgdruckt, die Kontollcodes für den Drucker kann und will ich mir einfach nicht in den Schädel hämmern. Das File ist allerddings auch ziemlich lang

geworden, bis jetzt knapp 37KByte, es ist also für Festplattenbenutzer prädestiniert. Aber auch auf ein 80 Spuren Diskette passen ja fast 720kByte. Der Abdruck des Programms selbst erscheint mir ziemlich sinnlos, wer Interesse hat, soll sich bei mir melden, er oder sie kann dann die Files zugeschickt bekommen. Im Folgenden also 'HILF DR<ENTER>' :


```
HILF (Ver.1) für G-DOS und CalvaDOS 2.4
nach einer Idee von Matt Wakeley

DR --- Bemerkung drucken.
    DR,<$>,<#>,<text>
     ermöglicht die direkte Ausgabe von Text und Sonderzeichen an

    den Drucker. Maximale Zeilenlänge ist 76 Zeichen.
    '$' ersetzt das Zeichen ESC (ASCII 27).
    '#' bedeutet die Umwandlung des nachfolgenden Buchstabens in
    einen Controlcode (z.B. #N ergibt 14).
    Eingige Umschaltsequenzen für Schriftarten:
    DR #§   =>  ist ASCII 00, #A ist ASCII 01
    DR $W#A=>   Doppelt breite Zeichen
    DR $W#§=>   Doppelt breite Zeichen aus
    DR #N   =>  Breitschrift
    DR #O   =>  Komprimierter Druck
    DR #Q   =>  Aufhebung des komprimierten Druckens
    DR #S   =>  Aufhebung der Breitschrift
    DR $P   =>  Elite-Schrift
    DR $§   =>  Drucker-Reset
    DR $-   =>  Unterstreichen
    DR $S   =>  Tieferstellen
    DR $S#§=>   Hochstellen
    DR $E   =>  Hervorgehobener Druck
    DR $F   =>  Ausschalten von $E
    DR $G   =>  Doppeltdruck
    DR $H   =>  Ausschalten von $G
    DR $4   =>  Kursiver Druck
    DR $5   =>  Kursivdruck aus
:7 Workbench >
```

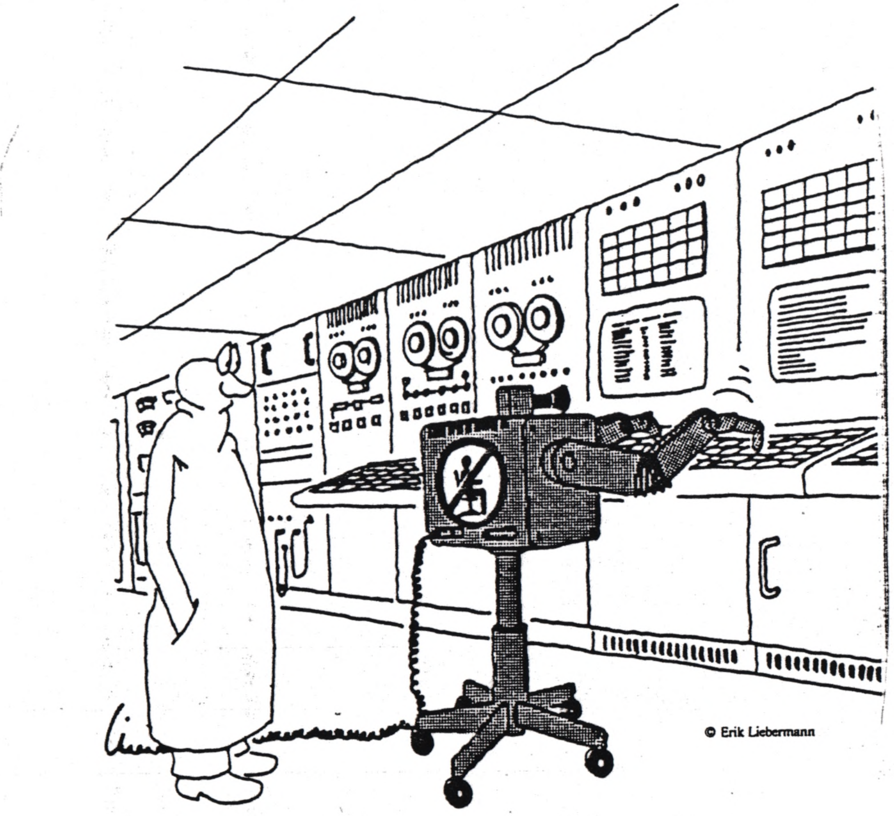
Volk Da

```
00002 ;*****************************************************************************
00003 ;        I      N      V      /      C      M      D
00004 ;
00005 ;       INV/CMD zeigt auf dem angegebenen Laufwerk nur die
00006 ;       unsichtbaren Files an.
00007 ;
00008 ;       SYNTAX :=  INV <LW#><ENTER>
00009 ;
00010 ;
00011 ;*****************************************************************************
00012 ;
00013 dosrdy  EQU     402dh
00014 doserr  EQU     4409h
00015 debug   EQU     440dh
00016 dirrea  EQU     490ah
00017 gibaus  EQU     4467h
00018 tesdsk  EQU     445eh
00019 ;
00020 ;
00021 ;
00022 ;
00023 ;
00024         ORG     5200h
00025 lwnum   DM      '0'
00026 debuf   DW      1806h    ;18h=24 Zeilen mit 6 Filenames /Zeile
00027 fdeanz  DW      0000h    ; hier steht wieviele Files angezeigt wurden
00028 secanz  DB      01h      ; Anzahl der DIR-Sektoren, wird incrementiert
00029 secbuf  DB      00h      ; Hier steht die Anzahl immer !
00030 ;
00031 ;
00032 ;
00033 ;
00034 error   LD      A,2fh    ; 'schlechte parameter
00035         JP      doserr
00036 istzahl SUB     '0'      ;ascii korrektur
00037         JR      C,error
00038         CP      10
00039         JR      C,eindir
00040         JR      error
00041 ;
00042 ;
00043 ; Hier fängt das Hauptprogramm an
00044 ;
00045 start   LD      IX,fdeanz      ; IX wird der Zähler für die angz.Files
00046         LD      (IX+0),00h
00047         LD      A,(HL)
00048         CP      0dh            ; wenn ENTER, hole lw# von 43a0h
00049         CALL    NZ,istzahl
00050         LD      A,(43a0h)      ; hier aktuelles laufwerk
00051 eindir  LD      (lwnum),A      ; hier die Routine für genau 1 Laufwerk
00052         ADD     A,'0'          ; binär=>ASCII
00053         LD      (dnum),A       ; ist für U-Prog. 'zeile'
00054         CALL    zeile          ; hier wird die Lw# sowie der Name
00055         CALL    diskdir        ; des Lw's auf den Bildschirm gebracht
00056         LD      A,(IX+0)       ; Wieviele Files sind angezeigt ?
00057         CP      00h            ; wenn kein File der Spezi. genügte
00058         JP      NZ,dosrdy      ; dann wird die 'zeile' wieder gelöscht
00059         LD      A,1bh          ; Cursor eine Zeile höher
00060         CALL    0033h          ; auf den Bildschirm
00061         JP      dosrdy         ; jetzt aber Abgang
00062 diskdir CALL    ermittl
00063 dirwei  LD      A,(secanz)
00064         INC     A
00065         LD      (secanz),A     ; 1 erhöhen und zurückschr.
00066         PUSH    HL
00067         LD      HL,secbuf
00068         CP      (HL)
00069         POP     HL
00070         RET     Z              ; hier ist der AUSGANG der U-Routine
```



```
00071 dirsec  CALL    dirrea         ; ließt einen Dirsector
00072         JP      NZ,doserr      ; falls ein Fehler auftrat
00073         CALL    auswert
00074         JP      dirwei
00075 ermittl LD      A,01h          ; HIT-Sector
00076         CALL    dirrea
00077         LD      A,(421fh)      ; hier steht die Anzahl der
00078         ADD     A,0ah          ; Dirsec, minus 10 inkl.HIT&GAT
00079         LD      (secbuf),A
00080         RET
00081 auswert LD      BC,0800h       ; 8 Einträge pro Sector
00082 awloop  LD      A,(HL)         ; erstes Zeichen des FDE's
00083         CP      18h            ; ist es ein FPDE ?
00084         JR      NZ,fxde        ; wenn nicht war b7=1 oder b4=0
00085         LD      A,L            ; jetzt soll HL auf den Filename zeigen
00086         ADD     A,05h          ; der ist bei FDE+5
00087         LD      L,A            ; HL auffrischen
00088         CALL    verglei        ; vergleichen und anzeigen
00089         LD      A,1bh          ; 5 weniger als bei keinem Eintrag
00090                                ; weil der Zeiger im Sektorpuffer
00091                                ; in Zeile 86 mit 5 addiert wurde
00092         JR      fpde
00093 fxde    LD      A,20h          ; alle 20h Zeichen beginnt ein FDE
00094                                ; (FileDirectoryEintrag)
00095 fpde    ADD     A,L
00096                                ; FPDE=FilePrimaryDirectoryEntry
00097                                ; also erster Eintrag eines Files
00098 ;
00099                                ; FXDE=FileExtensionDirectoryEntry
00100                                ; also Erweiterungseintrag
00101         LD      L,A            ; HL zeigt auf den nächsten FDE
00102         DJNZ    awloop         ; bis 8 FDE's verglichen wurden
00103         RET                    ; und wieder eine Ebene höher
00104 verglei PUSH    BC             ; einen Zähler brauch ich selbst
00105         PUSH    HL
00106 eintrag LD      DE,(debuf)
00107         CALL    ausgabe
00108         LD      (debuf),DE
00109         INC     (IX+0)         ; fdeanzahl=fdeanz+1
00110 zurück  POP     HL
00111         POP     BC
00112         RET
00113 dirfert JP      dosrdy
00114 ;
00115 ;
00116 ;
00117 ;
00118 zeile   LD      A,(lwnum)      ;Hier wird die Laufwerksnummer und der
00119         CALL    tesdsk         ;Diskettenname auf dem Bildschirm
00120         JP      NZ,doserr      ; angezeigt.
00121         LD      A,00h          ; Name ist im GAT-Sector an Position
00122         CALL    dirrea         ; Byte D0h bis D7h
00123         LD      HL,42d0h
00124         LD      DE,dname
00125         LD      BC,0008h       ; 8 Zeichen
00126         LDIR
00127         LD      HL,zeitxt
00128         CALL    gibaus         ; den gesamten String ausgeben
00129         RET                    ; ist ein Unterprogramm
```

```
zeitxt  DM      'in :'
dnum    DB      00h
        DM      '
dname   DM      'DISKNAME !,0dh
;
;
;
;
;
;       Ab hier liegen Unterroutinen, die ich von Andreas Magnus
;       einfach übernommen habe. Moin, moin Andreas !
;       Sie stammen aus seinem höchst nützlichen Programm
;
;           DIRVERGL/CMD
;
;       Und machen die Bildschirmausgabe so schön einfach.
;
;UP AUSGABE: gibt den Filenamen in HL auf dem Bildschirm oder Drucker aus
;            und zwar 6 Files nebeneinander und 24 files pro Schirm

ausgabe LD      B,08h           ;8 Byte Filenamen
        LD      C,H             ;H sichern, wird noch gebraucht
        CALL    aushl           ;und ausgeben
        LD      A,20h           ; ein Space zum Trennen
        CALL    ausa
        LD      B,03h           ;3 Byte Extension
        CALL    aushl           ;auch ausgeben
        LD      A,':'
        CALL    ausa            ; hier den Doppelpunkt
        DEC     E               ;(Filenamen nebeneinander)
        LD      H,C             ;H vorsichtshalber zurück
        RET     NZ              ;alles klaro

        LD      E,6             ;wieder auf 6 stellen
        DEC     D               ;Zeilenzähler -1
        JR      NZ,neuz         ;neue Zeile anzeigen
        LD      D,18h           ;wieder auf 24 Zeilen stellen
        EX      DE,HL           ;Zähler sichern
        CALL    warte           ;auf Tastendruck warten
        EX      DE,HL           ;und Zähler zurück
neuz    LD      A,0dh           ;CR => A
        CALL    ausa            ;und anzeigen und zurück
        LD      H,C             ;H jetzt entgültig zurück
        RET

;UP AUSHL: gibt B Zeichen lang (HL) ff aus

aushl   LD      A,(HL)          ;Zeichen in den ACCU
        CALL    ausa            ;und ausgeben
        INC     HL              ;auf nächste Stelle
        DJNZ    aushl           ;bis alle Angezeigt
        RET

;UP AUSA: gib den ACCU auf dem Bildschirm oder Drucker aus

ausa    PUSH    DE              ;Zeilenzähler retten
        CALL    0033h           ;oder 3bh für Drucker
        POP     DE
        RET

;UP WARTE: wartet bis CR gedrückt ist

warte   CALL    0049h           ;auf Tastendruck warten
        CP      0dh             ;war es Enter ?
        RET     Z               ;zurück wenn ja
        JR      warte           ;weiter warten
;
;
;
;
        END     start
```

© Erik Liebermann

Wer suchet, der FINDet.

Auch auf die Gefahr hin mich zu wiederholen:
Ich bin überglücklicher Besitzer einer Harddisk !

Aufbauend auf dem Rumpf meiner Programme 'KOPIERE' und 'ERASE' entschloß ich mich, diese Wildcards auch beim Aufruf des Inhaltsverzeichnisses nutzbar zu machen. Das Programm 'FIND/CMD' kann dies, mit 'FIND APFEL*/*:9' werden alle Dateien, die irgendwas mit Äpfeln zu tun haben (bei mir alles, was mit Mandelbrot zusammenhängt) auf den Bildschirm gebracht.

Außerdem macht mir seit einiger Zeit mein Gedächnis ein wenig zu schaffen, deswegen kann ich mich mit FIND davon überzeugen, ob sich ein Programm, welches ich auf einer meiner Disketten finde und welches ich eigentlich ganz interessant finde schon auf einem meiner Harddisklaufwerke befindet.
Dies wird erreicht durch das Weglassen der Laufwerksbezeichnung im Parameterstring. Jetzt werden nämlich die logischen Laufwerke 3 bis 9 nach irgendeinem oder sogar nach meheren Dateien durchsucht, das ist ganz nützlich, wenn man den genauen Namen einer Datei vergessen hat oder wenn man nicht mehr weiß, auf welchem Laufwerk sich die Datei denn nun verbirgt. Gibt man als Laufwerksbezeichner das Zeichen '*' an, so werden auch die Diskettenlaufwerke mit durchsucht, das dauert ja bekanntlich seine Zeit und muß ja nicht immer sein.

Das Programm arbeitet folgendermaßen :
Nachdem festgestellt wurde, welche Laufwerke oder welches denn durchsucht werden soll, wird ein Unterprogramm aufgerufen, daß einen Inhaltsverzeichnissektor nach dem anderen in den Speicher lädt und jeden Eintrag mit der eingegebenen Maske vergleicht. Ist der Dateiname würdig angezeigt zu werden, wird er das auch und zwar mit einer Routine, die ich wieder von Andreas Magnus übernahm. Ich habe den Teil, der fast original aus 'KOPIERE/SRC' stammt, nicht ausgedruckt, er stand ja im letzten INFO schon.

Das Programm ist zwar nicht besonders elegant programmiert, aber es erfüllt halt seinen Zweck. Der einzige Fehler, den ich finden konnte, geht so : Wenn man FIND */* eingibt, so stürzt der Rechner nach einigen ausgegebenen Zeilen einfach ab, wahrscheinlich habe ich die Unterroutinen von Andreas nicht sauber genug eingefügt. Dies wird man aber wohl ohnehin selten tun und ich war bis jetzt zu faul zum Suchen. Meinen Zwecken genügt es vollkommen.

Nach einiger Zeit und dem Besuch eines Freundes, der mir auf seinem IBM-Kompaktiblen das Programm 'WO/COM' zeigte, hatte ich das Gefühl, das mein 'FINDER' einfach zu langsam sei. Ich schloß messerscharf, das es daran liegt, das ich jeden DIR-Sektor nacheinander einlesen mußte, um ein Laufwerk zu durchforsten. Ich errinnerte mich daran, wie das DOS das Problem des Findens löst: Jeder Eintrag im Inhaltsverzeichnis hat seine Entsprechung in der HIT-Tabelle, welche den zweiten Sektor des 'INHALT/SYS' belegt. HIT steht für Hash Index Tabelle. In dieser Tabelle ist zu erkennen, ob ein Eintrag frei oder belegt ist, ist er frei, so ist der Eintrag in der HIT ein 00h. Jedem FDE im Inhaltsverzeichnis ist ein Byte in der HIT zugeordnet, außerdem entspricht die Position in der HIT genau dem DEC (Directory Entry Code), welcher sich über einen DOS-Call bequem einlesen läßt. Der Filename wird nach einem bestimmten und sehr einfachen Algorithmus kodiert und dieser 'Hash-Code' wird dann in der HIT eingetragen. Soll ein File vom DOS gefunden werden, so wird aus dem

Filenamen der Hash-Code errechnet und die HIT des jeweiligen Laufwerkes wird nach diesem Code durchsucht. Wird der Code in der Tabelle gefunden, wird der zuständige FDE eingeladen und dieser Filenameneintrag wird mit dem zu findenden verglichen. Ist er nicht gleich, was durchaus nicht unüblich ist, da die Information von 11 Zeichen (FILENAME/EXT) nicht in einem Byte gespeichert werden kann, wird die HIT weiter durchsucht. Wenn der errechntete Hash-Code gar nicht in der HIT zu finden ist, kann sich das File nicht auf der Diskette befinden.

Diese Funktion realisiert das Programm 'WO/CMD'. Es durchsucht alle angeschlossenen Laufwerke in aufsteigender Folge nach einem einzigen Dateinamen. Findet es diesen, so wird eine Meldung auf dem Bildschirm ausgegeben. Dieser Vorgang ist erheblich schneller als der beim 'FIND/CMD' angewendetet, so daß jetzt immer auch die Disketenlaufwerke mit durchsucht werden. Es kann selbstverständlich keine Wildcards im Eingabestring akzeptieren.

Diese Programme sind zwar nur für Leute mit Harddisk interessant (das FIND stellt allerdings auch ein 'DIR */*:lw#' zur Verfügung) aber ich hoffe daß der oder die eine so ein Gerät eingebaut hat. Ansonsten muß der oder die das eben bald tun.

Mit wunden Augen und platten Fingern

Schöne Grüße aus dem Norden

```
;
;
;         F    I    N    D    /    C    M    D
;
;         soll auf der Hard-, Ram- und eventuell Floppydisk
;         Programme finden, ähnlich einem DIR */*:*
;         also Filenamen, Extensions und auch Laufwerknummer
;         können beliebig angegeben werden.
;         Die Directoryunterfunktionen I(-nvisible),S(-ystem),
;         A(-lles) werden nicht implementiert.
;         File ist File. Die Laufwerksangabe ist * für alle,
;         bzw. explizite Lw#-angabe.
;         Man kann auch den :* weglassen und enfach ENTER nach der
;         Filespezifikation angeben
;
dosrdy  EQU     402dh
doserr  EQU     4409h
debug   EQU     440dh
dirrea  EQU     490ah
gibaus  EQU     4467h
tesdsk  EQU     445eh
;
;
;
;
;
;         Dies ist ein Teilstück des Programmes
;                   KOPIERE/CMD
;         und zwar wird hier die richtige Eingabe des Filenamens
;         kontrolliert, und zwar Test auf ?,* inklusive.
;         Man kann jetzt allerdings auch Zahlen und das Zeichen '_'
;         eingeben(allerdeings auch an erster Stelle, was eigentlich
;         nicht in Ordnung ist).
;
        ORG     5200h
fname   DM      'FILENAME';wird vom Progr. eingesetzt
exten   DM      'EXT'     ;hier steht dann der Vergleichsstring
ende    DM      0dh
lwnum   DM      '0'
debuf   DW      1806h   ;18h=24 Zeilen mit 6 Filenames /Zeile
fdeanz  DW      0000h   ; hier steht wieviele Files angezeigt wurden
secanz  DB      01h     ; START der FDE-Sektoren in DIR/SYS
secbuf  DB      00h     ; Hier wird eingetragen, wie viele DIR-Sektoren
;
;   Hier fängt das Hauptprogramm an
;
progr   LD      IX,fdeanz       ; IX wird der Zähler für die angz.Files
        LD      (IX+0),00h
        INC     HL      ; zeigt jetzt auf lwnummer
        LD      A,(HL)
        CP      '*'
        JP      Z,alle
        CALL    istzahl
eindir  LD      (lwnum),A       ; hier die Routine für genau 1 Laufwerk
        ADD     A,'0'           ; binär=>ASCII
        LD      (dnum),A        ; ist für U-Prog. zeile
        CALL    zeile           ; hierwird die Lw# sowie der Name
        CALL    diskdir         ; des Lw's auf den Bildschirm gebracht
        LD      A,(IX+0)        ; Wieviele Files sind angezeigt ?
        CP      00h             ; wenn kein File der Spezi. genügte
        JP      NZ,dosrdy       ; dann wird die 'zeile' wieder gelöscht
        LD      A,1bh           ; Cursor eine Zeile höher
        CALL    0033h           ; auf den Bildschirm
        JP      dosrdy          ; jetzt aber Abgang
```



```
diskdir CALL    ermittl
dirwei  LD      A,(secanz)
        INC     A
        LD      (secanz),A      ; 1 erniedriegen und zurückschr.
        PUSH    HL
        LD      HL,secbuf
        CP      (HL)
        POP     HL
        RET     Z               ; hier ist der AUSGANG der U-Routine
dirsec  CALL    dirrea          ; ließt einen Dirsector
        JP      NZ,doserr       ; falls ein Fehler auftrat
        CALL    auswert
        JP      dirwei
ermittl LD      A,01h           ; HIT-Sector
        CALL    dirrea
        LD      A,(421fh)       ; hier steht die Anzahl der
        ADD     A,0ah           ; Dirsec. minus 10 inkl.HIT&GAT
                                ; + noch einen weiß der Geiger warum
        LD      (secbuf),A
        RET
auswert LD      BC,0800h        ; 8 Einträge pro Sector
awloop  LD      A,(HL)          ; erstes Zeichen des FDE's
        AND     90h             ; alle außer b7 & b4 werden gelöscht
        CP      10h             ; ist es ein FPDE ?
        JR      NZ,fxde         ; wenn nicht war b7=1 oder b4=0
                                ; herumrechnen zu müssen.
        LD      A,L             ; jetzt soll HL auf den Filename zeigen
        ADD     A,05h           ; der ist bei FDE+5
        LD      L,A             ; HL auffrischen
        CALL    verglei         ; vergleichen und anzeigen
        LD      A,1bh           ; 5 weniger als bei keinem Eintrag
        JR      fpde
fxde    LD      A,20h
fpde    ADD     A,L
        LD      L,A             ; HL zeigt auf den nächsten FDE
        DJNZ    awloop          ; bis 8 FDE's verglichen wurden
        RET                     ; un wieder eine Ebene höher
verglei PUSH    BC              ; einen Zähler brauch ich selbst
        PUSH    HL
        LD      B,11            ;
        LD      DE,fname        ; Vergleichsfilespezifikation

verloop LD      A,(DE)
        CP      '?'
        JR      Z,veregal               ; wenn ?, dann ist alles erlaubt
        CP      (HL)            ; ist es gleich dem FDE ?
        JR      NZ,zurück       ; falls nciht, noch mal POPen
veregal INC     DE
        INC     HL              ; beide Zeiger ein Weiter
        DJNZ    verloop         ; 11 zeichen
        POP     HL
        PUSH    HL
eintrag LD      DE,(debuf)
        CALL    ausgabe
        LD      (debuf),DE
        INC     (IX+0)          ; fdeanzahl=fdeanz+1
zurück  POP     HL
        POP     BC
        RET
dirfert JP      dosrdy
;
;
```

```
;
;
zeile   LD      A,(lwnum)       ;Hier wird die Laufwerksnummer und der
        CALL    tesdsk          ;Diskettenname auf dem Bildschirm
        JP      NZ,doserr       ; angezeigt.
        LD      A,00h           ; Name ist im GAT-Sector an Position
        CALL    dirrea          ; Byte D0h bis D7h
        LD      HL,42d0h
        LD      DE,dname
        LD      BC,0008h        ; 8 Zeichen
        LDIR
        LD      HL,zeitxt
        CALL    gibaus          ; den gesamten String ausgeben
        RET                     ; ist ein Unterprogramm
zeitxt  DM      'in :'
dnum    DB      00h
        DM      '
dname   DM      'DISKNAME ',0dh
;
;
;       Einsprung in 'alle', wenn nur RAMDisk und HARDDISK durchsucht
;       werden soll
abram   LD      A,03h           ; RAMDISK ist LW# 3
        JR      aloop           ; und sonst genau wie alle
;
alle    LD      A,00h           ; alle heißt das ein oder mehrer Files
aloop   LD      (lwnum),A       ; auf allen verfügbaren Laufwerken gesuch
        CALL    tesdsk          ; ist überhaupt erreichbar ?
        JR      NZ,is_nich      ; zerobit =0 bei keinem fehler
is_doch LD      A,(lwnum)
        ADD     A,'0'           ; binär => ASCII
        LD      (dnum),A        ; in zeitxt
        LD      (IX+0),00h      ; fdezähler
        CALL    zeile           ; Diskname usw.
        CALL    diskdir         ; Inhaltsverzeichnis durchstöbern
        LD      A,(IX+0)        ; ist mindestens ein File gefunden worden
        CP      00h
        JR      NZ,is_nich
        LD      A,1bh           ; cursor eine Zeile höher
        CALL    0033h
is_nich LD      A,0bh           ; zeilenvorschub wenn nicht schon
        CALL    0033h
        LD      A,01h           ; bei 'alle' muß jedesmal der secanz neu
        LD      (secanz),A      ; auf den ersten DEC-Sektor gesetzt werdn
        LD      A,(lwnum)
        INC     A
        LD      (lwnum),A
        CP      0ah             ; 9 laufwerke durch ?
        JR      NZ,aloop        ; sonst weiter suchen
        JP      dosrdy
;
;
;       Ab hier liegen Unterroutinen, die ich von Andreas Magnus
;       einfach übernommen habe. Moin, moin Andreas !
;       Sie stammen aus seinem höchst nützlichen Programm
;
;          DIRVERGL/CMD
;       Und machen die Bildschirmausgabe so schön einfach.
;
        END     start
```



```
;
;               WO/CMD
;
;               Programm, um genau ein bestimmtes Programm auf allen
;               angeschlossenen Laufwerken zu finden.
;               Der Suchvorgang funktioniert mit der Errechnung des
;               HASH-Codes, es muß von jedem Laufwerk also nur der HIT-
;               Sector eingelesen werden.
;               Das geht natürlich erheblich schneller.
;
dosrdy  EQU     402dh
gibaus  EQU     4467h
doserr  EQU     4409h
debug   EQU     440dh
tstdsk  EQU     445eh
dirrea  EQU     490ah
getfde  EQU     4936h
;
        ORG 5200h
filspe  DM      20h,20h,20h,20h,20h,20h,20h,20h; 8 Zeichen Filespec.
extspe  DM      20h,20h,20h     ; 3 Zeichen für Extension
aktlw   DM      00      ; Suche starten bei LW# 0, wird hochgezählt
hashco  DM      00h     ;hier wird der errechnete Hashcode eingetragen
;
start   PUSH    HL      ; zeigt auf den eingegenben Filenamen
        LD      DE,filspe
        LD      B,09h   ; 8 Zeichen für
nloop   LD      A,(HL)
        CP      0dh     ; ENTER ?
        JR      Z,rechne
        CP      '/'     ; Trennzeichen zur Extension ?
        JR      Z,ext
        LD      (DE),A
        INC     DE
        INC     HL
        DJNZ    nloop   ; bis 8 Zeichen in Buffer übertragen(Höchstens)
        CP      '/'     ; folgt jetzt der slash ?
        LD      A,13h
        JP      NZ,doserr
ext     LD      B,03h   ; 3Zeichen für Extension
        INC     HL
        LD      DE,extspe
eloop   LD      A,(HL)
        CP      0dh
        JR      Z,rechne
        LD      (DE),A
        INC     DE
        INC     HL
        DJNZ    eloop
        LD      A,(HL)
```

```
        CP      0dh         ; hernach also hier muß!! Enter sein
        LD      A,13h
        JP      NZ,doserr
rechne  LD      HL,filspe       ; diese Routine errechnet den HASH-Code
        LD      B,0bh           ; sie ist abgeschrieben aus SYS2/SRC
        XOR     A
rloop   XOR     (HL)
        INC     HL
        RLCA
        DJNZ    rloop
        JR      NZ,fertig
        INC     A
fertig  LD      (hashco),A      ; abspeichern.
;       Der HASH-Code ist errechnet, jetzt müssen alle HIT-Sectoren
;       eingelesen und der Inhalt derselben mit dem hashco verglichen
;       werden. Dann muss der jeweilige FDE geholt werden und mit
;       filspe verglichen werden. Erst dann ist der Filename gefunden.
;
drloop  LD      A,(aktlw)       ; LW# laden
        CALL    tstdsk          ; ist eine Disk eingelegt ??
```

```
        CALL    Z,liehit        ; wenn ja, lies den HIT-Sector
        LD      A,(aktlw)
        INC     A
        CP      0ah             ; schon über 9 hinaus ?
        JP      Z,ende
        LD      (aktlw),A
        JR      drloop
ende    LD      HL,text1
        CALL    gibaus
        JP      dosrdy
liehit  LD      A,01h           ; HIT-Sector
        CALL    dirrea
        JP      NZ,doserr       ; falls Fehler(sehr unwahrscheinlich)
        LD      B,0ffh
        LD      A,(hashco)
sloop   CP      (HL)            ; sucht den gesamten Sectorpuffer ab
        CALL    Z,su_fde        ; wenn der hashcode stimmt, muß der
        INC     HL              ; zuständige FDE mt filspe verglichen
        DJNZ    sloop           ; werden.

        RET
su_fde  PUSH    HL
        PUSH    AF
        PUSH    BC
        LD      A,L             ; L ist LSB des Sectropufferzeigers,
        CALL    getfde          ; und ist genau der FDE im Directory
        LD      DE,filspe
        LD      BC,0005h        ; FILENAME steht an 5.ter Stelle
        ADD     HL,BC           ; jetzt zeigt HL auf den FILENAMEEXT
        LD      B,0bh
fdloop  LD      A,(DE)
        CP      (HL)
        JR      NZ,nein
        INC     DE
        INC     HL
        DJNZ    fdloop
ja      LD      A,(aktlw)
        ADD     A,'0'           ; binär => ASCII
        LD      (lwnum),A
        LD      HL,text2
        CALL    gibaus
nein    LD      A,01h           ; der HIT-Sector muss wieder eingelesen
        CALL    dirrea          ; werden, er wurde überschrieben
        POP     BC
        POP     AF
        POP     HL
        RET
text2   DM      'File gefunden auf LW# '
lwnum   DM      '0.',0dh
text1   DM      'Alle Laufwerke sind durchsucht.',0dh
        END     start
```

Das Schlußwort unseres Club-Freundes Jens, war in der 35.Ausgabe des Clu-Infos ein Aufruf an alle Computerbesitzer die mit BASIC arbeiten.

Auch ich wäre der Meinung, daß man sich innerhalb des CLUB 80 zu einer BASIC-Anwendergruppe zusammentun sollte. Ich selber arbeite seit 8 Jahren mit G-DOS der Firma Trommelschläger, wovon ich nicht abzubringen bin. Mit diesem G-DOS,was ja die Ableitung bzw. Erweiterung von NEWDOS 80 ist, habe ich mich sehr stark angefreundet.

Wer hat Interesse an der Gründung einer BASIC-Anwendergruppe ?

Meldet Euch bitte bei: Willi Johnen, Hansemannstr.1
5160 Düren, Tel. 02421/501305

BASIC

Liebe Basic-Freunde,

wie die Zuschriften hier im INFO und auch Gespräche zeigten, ist BASIC ein Treffpunkt, an dem alle Computer unseres Club's wieder zusammenfinden.

Mich freut es ganz besonders, wieder einmal eine Gemeinsamkeit innerhalb des Club's -trotz der verschiedensten Rechnertypen-gefunden zu haben. Aus diesem Grund ist es, bei entsprechender Beteiligung, sinnvoll eine BASIC-Ecke ins Leben zu rufen. Interessenten melden sich bitte bei Willi Johnen oder bei mir.

Sicherlich wird es nicht ganz einfach sein, die verschiedenen BASIC-Dialekte unter einen Hut zu bringen. Was bei dem einen als Befehl im Basic schon vorhanden ist, kann ein anderer erst mit Hilfe seiner Toolkiste ausführen. Hier liegt, meiner Meinung nach, ein wichtiger Aktivitätspunkt der BASIC-Anwendergruppe:
Bereitstellen von Tools, die die Arbeit erleichtern und sich in andere Basicprogramme einfügen oder anpassen lassen.
Es muß ja nicht jeder seine eigene ...xy-Routine geschrieben haben. Gleichzeitig können durch die Gruppe Anregungen zur Optimierung gemacht werden. Somit haben eigentlich alle Beteiligten einen Nutzen an dieser Aktion.

Ich hoffe auf vielfältige Aktivität und freue mich schon auf die nächsten Veröffentlichungen -und das nicht nur wegen der CLUB-INFO!

Jens Neueder

B A S I C - Neuen Wind für eine totgeglaubte Sprache

Mittlerweile gibt es viele, viele schlaue, hilfreiche, übersichtliche, effektive, schnelle und was weiß ich nicht noch alles für Programme und Programmiersprachen, so daß kaum noch Eine(r) von der antiquarisch-amateurhaft anmutenden Sprache BASIC redet.

S C H A D E !

Jens hat sich im Info Nr. 36 etwas verschämt dazu bekannt und ich möchte diese Anregung einer BASIC-Anwendergruppe aufgreifen und für das INFO eine 'BASIC-Ecke' eröffnen.

'Damals' habe ich das Programmieren auch mit BASIC gelernt, daher liegt mir diese Sprache verständlicherweise immer noch sehr leicht auf der Zunge. Hauptsächlich aus folgenden Gründen heraus ist sie bei mir auch jetzt noch beliebt, wenn auch nicht mehr für alle Anwendungen:

- ich kann Ein- und Ausgabedialoge komplett steuern und so gestalten, wie ich es schön und praktisch finde
- ich kann Fehler abfragen und das Programm ohne Abbruch weiterführen

Um dies verständlicher zu machen, gibt es jetzt ein paar Beispiele (bekanntlich läßt man sich davon ja am besten stimmulieren) und ich bin der festen Überzeugung, daß es dann zukünftig nur so von Beispielen, Anfragen, Tips und Tricks aus den CLUB-Reihen für unsere neue Ecke hagelt...

Euer Christof

P.S.: Meine Adresse lautet richtig:

Christof Neumann
Zeitblomstr. 22/2
7900 Ulm
Tel. 0731/6022568

1. ALLGEMEINES VORWEG

Grundsätzlich habe ich die meiste Zeit mit einem Microsoft-BASIC namens MBASIC unter dem Betriebssystem CP/M gearbeitet; alle Programmierbeispiele sind damit verfaßt. Mein Computer hört auf den Namen 'Tandy Model II' und alle zitierten Steuerzeichen sind für ihn gedacht.

Je nach Computer und verwendeter BASIC-Version ergeben sich also Unterschiede, die zu beachten sind.

Je größer und rechenintensiver ein BASIC-Programm ist, umso wichtiger wird die Übersetzung des BASIC-Quellprogrammes in die Maschinensprache: Das COMPILIEREN. Hierzu gibt es spezielle Software, wobei mir allerdings ein Überblick über Angebot, Leistungen und Möglichkeiten fehlt; ich weiß nur, daß mein COMPILER auch auf meinem Model IVp mit dem Montezuma-CP/M läuft.

2. STEUERUNG DER DIALOGE

Ich habe es ganz gerne, wenn meine Programme idiotensicher auch von anderen bedient werden können. Dazu ist nicht nur eine Bedienungsanleitung und ein eindeutiger Bildschirmdialog nötig, sondern obendrein auch - je nach Bedarf - eine Abfrage dessen, was dem Computer eingepickt wird.

In dem folgenden Beispiel sind ein paar grundsätzliche Definitionen aufgelistet, die bei fast allen Programmen so dastehen.

```
DEF FNCU$(X,Y)  =  Positionierung des Cursors auf X,Y
                (das geht bei anderen Rechnern sicher einfacher)
EIN$            =  Inverse Bildschirmdarstellung ein
AUS$            =  dasselbe wieder aus
CLS$            =  gesamten Bildschirm löschen
```

```
170 WIDTH 255
175 DIM BV$(7),BVK$(7),GMNR(7),GMKG$(7),KGMK$(7),ANFL(7),SBET#(7),SBEI#(7)
180 DEF FNCU$(X,Y)=CHR$(27)+CHR$(61)+CHR$(32+Y)+CHR$(32+X)
190 EIN$=CHR$(14):AUS$=CHR$(15):CLS$=CHR$(26)
200 PRINT CLS$
210 PRINT FNCU$(9,2);EIN$;SPC(62)
220 FOR I=1 TO 5 :PRINT FNCU$(9,2+I);"  ";FNCU$(69,2+I);"  ":NEXT
230 PRINT FNCU$(9,8);SPC(62);AUS$
```

Die Zeilen 210 - 230 malen mir ein kleines Eingabefenster auf den Bildschirm.

Jetzt kommt ein Eingabebeispiel, wo ich an einer vorgegebenen Stelle auf dem Bildschirm (Zeile 1680) ein Zahl mit maximal 2 Stellen eingeben lasse (andere Zeichen sind nicht zugelassen):

```
1670 FL$=""
1680 PRINT FNCU$(41,Y%+5);"";
1690 ANT$=INKEY$:IF ANT$="" THEN 1690
1700 IF ANT$=CHR$(13) THEN 1770
1710 IF ANT$=CHR$(8) AND LEN(FL$)>0 THEN FL$=LEFT$(FL$,LEN(FL$)-1):GOTO 1740
1720 IF ASC(ANT$)<48 OR ASC(ANT$)>57 THEN 1690
1730 IF LEN(FL$)<2 THEN FL$=FL$+ANT$
1740 PRINT FNCU$(41,Y%+5);"  "
1750 PRINT FNCU$(41,Y%+5);FL$;
1760 GOTO 1690
```

Weitere Möglichkeiten, die denkbar sind:

- Umwandlung von Kleinbuchstaben automatisch in Großbuchstaben (z. Bsp. für Dateinamen)
- freies Definieren von Sonderzeichen (z.Bsp. wenn man Komma eingibt, wandelt der Rechner dies in Punkt um)

Der Phantasie sind keine Grenzen gesetzt. Da dies bei der Eingabe von vielen Werten im Programm einen Wust von Programmtipperei gibt, kann man das ganze auch als Unterroutine anspringen; hier z. Bsp. zur Eingabe von Texten mit einer Eingabemaske, die die Wortlänge mit der Variablen 'EL' beschränkt:


```
9000 '     --> UP Eingabe Texte <--
9010 PRINT FNCU$(EX,EY);">";STRING$(EL,"_");"<"
9020 PRINT FNCU$(EX+1,EY);"";
9030 LINE INPUT "",E$
9040 E$=LEFT$(E$,EL):RETURN
```

3. FEHLERBEHANDLUNG

In dem folgenden Beispiel wird vorausgesetzt, daß am Programmanfang irgendwo ein 'ON ERROR GOTO 10000' steht. Findet das Programm ein bestimmtes File auf der Diskette nicht, gibt es normalerweise eine Fehlermeldung und das Programm bricht ab. Hier wird in Zeile 10010 stattdessen bei Auftreten des Fehlers (Fehlernummer 53) die gewünschte Datei angelegt und eine Null reingeschrieben. Alle anderen Fehler werden mit Nummer und Programmzeile angegeben und das Programm steigt aus.

```
10000 '    --> Fehlerroutine  <--
10010 IF ERR=53 THEN CLOSE:OPEN "O",#1,DAT$:PRINT #1,"0":CLOSE:RESUME
10020 PRINT FNCU$(17,22);EIN$;"Fehler Nr. ";ERR;" in Zeile ";ERL;AUS$;:END
```

4. AUSBLICK AUF WEITERE THEMEN

Ich möchte in einer späteren Folge auf die Themen

- Schnelligkeit durch richtiges Programmieren
- Sortieren mit BASIC - mit welcher Logik?

eingehen und außerdem interessiere ich mich auch noch für das Ansprechen der Schnittstellen mit BASIC. Wer kann mir dabei mit seinen Erfahrungen, Literatur, Beispielen etc. hilfreich unter die Arme greifen ???

Hallo alle GENIE III S -Besitzer !!

Ich hoffe Ihr habt geneuso wie ich die Karnevals- bzw. Faschingstage gut überstanden, was auch der Grund war, wieso ich Euch in der Nr.36 des CLUB 80-INFOS habe "hängen" lassen. Da ich im Karneval sehr aktiv bin, war ich nicht in der Lage meinen Beitrag zum CLUB-Info zu schreiben. Ich bitte Euch herfür Verständnis zu haben.

In dieser Ausgabe liefere ich Euch den III.Teil des ADRESS-TEXT-90, die Adressenverwaltung. Mit diesem Programm könnt Ihr

*** ca. 2000 Adressen speichern

*** den gesammten Adressenbestand sortieren nach Familiennamen

*** eine Adressenliste formatiert ausdrucken

Hinweise: Dieses Programm läuft nur mit hochauflösender Graphik und einer
--------- installierten Festplatte (LW5 + LW6)

Nachfolgende Zeilen müssen jenachdem geändert werden:
Z.0150 Hochauflösende Graphik
Z.0160 Laufwerksangabe
Z.1670 Laufwerksangabe
Z.1700 Laufwerksangabe
Z.1730 Laufwerksangabe
Z.1740 Laufwerksangabe

Die Hochauflösende Graphik zu diesem Programm wurde bereits in Nr.34 des CLUB-INFOS vorgestellt.

Bereits erfolgte Veröffentlichungen:

I.Teil - Club-Info Nr.34: Textverarbeitung
II.Teil - Club-Info Nr.35: Startprogramm und Hauptmenü


```
10 '#)) Programmname  (( JOHADRES/BAS ))                 ((#
20 '#)) Adressenverwaltung des ADRESS-TEXT-90 - Privat   ((#
25 '#)) (C)opyright by W.Johnen,5160 Düren               ((#
30 '#)) Stand: 19.01.1992          12.00 UHR             ((#
40 '#)) Hochauflösende Graphik                           ((#
50 '#)) Graphikbild:  )JOHNEN01/RDL(                     ((#
60 '#)) Datenspeicherung in LW 6                         ((#
70 '#)) Sortierprogramm                                  ((#
80 '#)) M I T     H A R D D I S K                        ((#
90 HOFF:HCLS:HON:CLEAR10000:DEFSTRA-D,G-H,J-K:DEFINTN,X,Z,W:DEFSNGY:DIMDX(7)
100 '#--------------------- Graphik $ ----------------#
110 CC=STRING$(15,140):CY=STRING$(79,32):KR=STRING$(20,144):KY=STRING$(30,144):KC=STRING$(16,144)
120 AD="NEUEINGABE":AE="ÄNDERUNG":AI="LISTE-DRUCKEN":AF="SORTIERUNG"
130 HX="####"
140 '#--------Datendiskette prüfen --------------------#
150 CLS:HLOAD"JOHNEN01/RDL":WP=0:HBIG30,225,1,1,1,"* ADRESSEN - VERWALTUNG *":HBIG536,225,1,1,1,LEFT$(TIME$,8);:HBIG536,202,1,1,1,RIGHT$
(TIME$,8);:IFWA=6THENRETURN
160 GA="":ONERRORGOTO170:OPEN"I",1,"JOHADRES/DAT:6":CLOSE:GOTO220
170 RESUME180
180 PRINT§405,"Im Laufwerk 6 ist keine  (JOHADRES/DAT - Datei)":PRINT§494,"Prüfen sie nochmal mit ( W )";
190 GA=INKEY$:IFGA="w"THENGOSUB200:GOTO160ELSE190
200 X7=405:FORX=1TO6:PRINT§X7,CY;:X7=X7+80:NEXTX:RETURN
210 '#-------- Erstellung der Programmnamen -----------#
220 K1="1 Anrede                  : ":K2="2 Vorname/Firmenbez.    : ":K3="3 Name/Firmenname        : ":K4="4 Straße/Postf./Hausnr. : ":K5="
5 PLZ/Wohnort             : ":K6="6 Telefonnummer         : ":K7="7 Ansprechpartner       : "
230 GOTO260
240 '#---------------- Menüanzeige -------------------#
250 CLOSE:WA=6:GOSUB150:WA=0
260 ONERRORGOTO0:PRINT§461,CC:PRINT§541,"PROGRAMM - MENü":PRINT§621,CC:PRINT§779,"(1) ";AD:PRINT§939,"(2) ";AE:PRINT§1099,"(3) ";AI:PRIN
T§1259,"(4) ";AF:PRINT§1419,"(5) HAUPT-MENü";
270 WU=VAL(INKEY$):IFWU<1ORWU>5THEN270ELSEONWUGOTO300,360,810,1640,280
280 CLEAR50:RUN"JOHMENü/BAS"
290 '#----------- Neueingabe von Daten ---------------#
300 CLS:WU=0:GOSUB1560:PRINT§131,"Neueingabe";:GOSUB1530:GOSUB1200
310 LSETDX(1)=DT(1):LSETDX(2)=DT(2):LSETDX(3)=DT(3):LSETDX(4)=DT(4):LSETDX(5)=DT(5):LSETDX(6)=DT(6):LSETDX(7)=DT(7)
320 PUT1,LQ:CLOSE
330 WA=0:GOSUB1600:PRINT§1848,"( W ) eitere Neueingaben machen          ( M ) zurück zum Menü";CHR$(7);
340 GA=INKEY$:IFGA="w"THEN300ELSEIFGA="m"THEN250ELSE340
350 '#----------- Ändern von Daten ------------------#
360 CLS:GOSUB1560:PRINT§131,"Ändern";
370 IFGQ="y"THEN400
380 GOSUB1590:GOSUB1620:PRINT§1842,"Nach welchem Suchbegriff wollen Sie suchen ? ";KR:PRINT§1887,;:LINEINPUTGN:IFLEFT$(GN,1)="/"THEN250
390 IFGN=""THEN400ELSEIFLEN(GN)>30THEN380
400 WP=9:GOSUB1530:WP=0
410 IFEOF(1)THENGOSUB770:CLOSE:GOTO360
420 IFGQ="y"THENX1=X4:GQ="":X4=0
430 X1=X1+1:GET1,X1
440 KZ=DX(1)+DX(2)+DX(3)+DX(4)+DX(5)+DX(6)+DX(7)
450 PRINT§141,USING":#### ";X1;:IFINSTR(KZ,GN)>0THEN470ELSE410
460 '#--- Änderung / Bildschirmanzeige des D-Block----#
470 GOSUB1590:PRINT§321,K1;DX(1):PRINT§481,K2;DX(2):PRINT§641,K3;DX(3):PRINT§801,K4;DX(4):PRINT§961,K5;DX(5);:PRINT§1121,K6;DX(6):PRINT§
1281,K7;DX(7)
480 PRINT§1841," <---( Z ) - ( V )---> - ( N )eusuchen - ( Ä )ndern - ( L )öschen - ( M )enü ";:ONERRORGOTO1500
490 GA=INKEY$:IFGA="ä"THEN560
500 IFGA="n"THENCLOSE:GOTO360
510 IFGA="z"THENGA="":X1=X1-1:GET1,X1:GOTO440
520 IFGA="v"THENGA="":GOTO410
530 IFGA="m"THEN250
540 IFGA="l"THEN690ELSE490
550 '#-------- Ändern des Datenblcks ---------------#
560 PRINT§131,"Änderung";:ONERRORGOTO0:GA="":PRINT§459,"Neue Daten eingeben";:GOSUB1180
570 '#----- Speichern des geänderten D-Blocks --------#
```

```
580 IFDT(1)=""THENDU(1)=DX(1)ELSEDU(1)=DT(1)
590 IFDT(2)=""THENDU(2)=DX(2)ELSEDU(2)=DT(2)
600 IFDT(3)=""THENDU(3)=DX(3)ELSEDU(3)=DT(3)
610 IFDT(4)=""THENDU(4)=DX(4)ELSEDU(4)=DT(4)
620 IFDT(5)=""THENDU(5)=DX(5)ELSEDU(5)=DT(5)
630 IFDT(6)=""THENDU(6)=DX(6)ELSEDU(6)=DT(6)
640 IFDT(7)=""THENDU(7)=DX(7)ELSEDU(7)=DT(7)
650 LSETDX(1)=DU(1):LSETDX(2)=DU(2):LSETDX(3)=DU(3):LSETDX(4)=DU(4):LSETDX(5)=DU(5):LSETDX(6)=DU(6):LSETDX(7)=DU(7)
660 PUT1,X1:CLOSE
670 WA=0:PRINT§1841," Die Änderung wurde durchgeführt!   <W>eitere Adressdaten ändern      <M>enü ";CHR$(7);:GOTO730
680 '#----- Löschen eines Adressblocks ----------------#
690 GA="":PRINT§131,"Löschung";
700 FORQ=1TO7:LSETDX(Q)="":NEXTQ
710 PUT1,X1:CLOSE
720 PRINT§1841," Die Löschung wurde durchgeführt !   <W>eitere Adressblöcke löschen   <M>enü ";CHR$(7);
730 GA=INKEY$:IFGA="m"THEN250
740 IFGA="w"THENCLOSE:GQ="y":X4=X1:GOTO360  'Datenrücksprung bei Änderungswiederholung
750 GOTO730
760 '#---- Ende der Adress-Datei (EOF) ----------------#
770 PRINT§1841,"  Kein weiterer Adress-Block vorhanden   <W>eitersuchen m.a.Suchkr.  <M>Menü ";CHR$(7);
780 GA=INKEY$:IFGA="m"THEN250
790 IFGA="w"THENRETURNELSE770
800 '#----- Adress-Liste drucken ----------------------#
810 ONERRORGOTO0:CLS:GOSUB1560:PRINT§131,"Drucken";
820 PRINT§1841," Welche Adress-Liste wollen Sie drucken ?   ( G )esamtbestand --- ( A )auswahl ";CHR$(7);
830 B1=INKEY$:IFB1="a"THEN840ELSEIFB1="g"THEN870ELSE830
840 GOSUB1590:GOSUB1620:PRINT§1842,"Nach welchem Suchbegriff wollen Sie drucken? ";KY:PRINT§1887,;:LINEINPUTGN:IFLEFT$(GN,1)="/"THEN250
850 IFGN=""THEN840ELSEIFLEN(GN))30THEN840ELSEIFASC(GN)(48ORASC(GN))126THEN840ELSE870
860 '#---------- Drucker einstellen ------------------#
870 XY=0:IFINP(253)=63THEN890ELSEPRINT§1841," Ihr Drucker ist nicht startbereit.     Prüfen Sie den Drucker (SEL-Lampe) !";:PRINT§1841,
CY;CHR$(7);:GOTO870
880 '#---------- Formular einstellen -----------------#
890 PRINT§1841,"     Ist das Formular richtig eingestellt ?          <D>rucken    <M>enü     ";CHR$(7);
900 GA=INKEY$:IFGA="d"THEN920ELSEIFGA="m"THEN250ELSE900
910 '#--------- Druck der Daten-Liste ----------------#
920 WP=9:GOSUB1530:WP=0
930 IFEOF(1)THENCLOSE:GOTO1150
940 LQ=LQ+1:GET1:KZ=DX(1)+DX(2)+DX(3)+DX(4)+DX(5)+DX(6)+DX(7)
950 IFB1="g"THEN970
960 IFINSTR(KZ,GN))0THEN970ELSE930
970 KN=DX(1):KV=DX(2):KU=DX(3):KS=DX(4):KO=DX(5):KW=DX(6):KP=DX(7)
980 JB=CHR$(14):JC=CHR$(15):JG=CHR$(27)+"N":JH=CHR$(27)+"E":JI=CHR$(27)+"Q":JN=CHR$(27)+"!":JO=CHR$(27)+CHR$(34)
990 IFX8=9THENLPRINT:LPRINT:LPRINT:X8=0:X7=0:GOTO1030
1000 IFXY)0THEN1060
1010 LPRINTJH+JN+JB;TAB(18)"ADRESS-VERZEICHNIS":LPRINT:LPRINTTAB(16)"P R I V A T"+JC:LPRINT:LPRINT
1020 LPRINTTAB(35)"STAND: ";LEFT$(TIME$,8):LPRINT:LPRINT
1030 LPRINTJI+JO+"  "+STRING$(128,45):LPRINT"  ANREDE"
1040 LPRINT"  VORNAME / FIRMENBEZEICHNUNG      PLZ / WOHNORT"
1050 LPRINT"  NAME / FIRMENNAME                STRASSE / POSTFACH / HAUSNR.     TELEFONNUMMER               ANSPRECHPARTNER":LPRINT"
  "+STRING$(128,45):LPRINT
1060 X7=X7+1:LPRINTCHR$(27)+"(003."
1070 LPRINTCHR$(09)KN;
1080 LPRINTCHR$(27)+"(003,036."
1090 LPRINTCHR$(09)KV;CHR$(09)KO;
1100 LPRINTCHR$(27)+"(003,036,069,101."
1110 LPRINTCHR$(09)KU;CHR$(09)KS;CHR$(09)KW;CHR$(09)KP;:LPRINT
1120 XY=XY+1
1130 FORX=0TOZA:S(X)=0:NEXTX:IFXY=12THENLPRINT:LPRINT:LPRINT:X8=9
1140 IFX7=15THENLPRINT:LPRINT:LPRINT:X8=9:X7=0:GOTO930ELSE930
1150 PRINT§1841," Die Adress-Liste ist fertig !       <W>eiterdrucken m.a.Suchkr.  <M>Menü    ";CHR$(7);
```

```
1160 GA=INKEY$:IFGA="w"THEN810ELSEIFGA="m"THEN250ELSE1160
1170 '#--------- Dateneingabe mit Maske ---------------#
1180 GOTO1450
1190 '#---Anrede             K1/KN #
1200 PRINT§321,K1;KY;CY;:GOSUB1600:GOSUB1620
1210 PRINT§1850,"Herrn - Frau - Eheleute - Architekt - Firma - Dr. - Sonstige";CHR$(7);
1220 PRINT§347,;:LINEINPUTDT(1):IFLEFT$(DT(1),1)="/"THENCLOSE:GOTO250
1230 IFLEN(DT(1))>30THEN1200ELSEIFWA=1THEN1430
1240 '#---Vorn./Firmenbez.   K2/KV #
1250 PRINT§481,;K2;KY;CY;:GOSUB1590:PRINT§1854,"Vornamen oder Firmenbezeichnung eingeben!";CHR$(7);
1260 PRINT§507,;:LINEINPUTDT(2):IFLEN(DT(2))>30THEN1250ELSEIFWA=2THEN1430
1270 '#---Name/Firmenname    K3/KU #
1280 PRINT§641,K3;KY;CY;:GOSUB1600:PRINT§1858,"Familiennamen oder Firmennamen eingeben!";CHR$(7);
1290 PRINT§667,;:LINEINPUTDT(3):IFLEN(DT(3))>30THEN1280ELSEIFWA=3THEN1430
1300 '#---Straße/Hausnr.     K4/KS #
1310 PRINT§801,K4;KY;CY;:GOSUB1600:PRINT§1861,"Straße und Hausnr. oder Postfach eingeben!";CHR$(7);
1320 PRINT§827,;:LINEINPUTDT(4):IFLEN(DT(4))>30THEN1310ELSEIFWA=4THEN1430
1330 '#---PLZ/Wohnort        K5/KO #
1340 PRINT§961,K5;KY;CY;:GOSUB1600:PRINT§1862,"PLZ und Wohnort eingeben!";CHR$(7);
1350 PRINT§987,;:LINEINPUTDT(5):IFLEN(DT(5))>30THEN1340ELSEIFWA=5THEN1430
1360 '#---Telefonnummer      K6/KW #
1370 PRINT§1121,K6;KC;CY;:GOSUB1600:PRINT§1874,"Telefonnummer eingeben !";CHR$(7);
1380 PRINT§1147,;:LINEINPUTDT(6):IFLEN(DT(6))>16THEN1370ELSEIFWA=6THEN1430
1390 '#---Ansprechpartner    K7/KP #
1400 IFDT(1)="Herrn"ORDT(1)="Frau"ORDT(1)="Fräulein"ORDT(1)="Eheleute"THEN1430
1410 PRINT§1281,K7;KY;CY;:GOSUB1600:PRINT§1854,"Ansprechpartner eingeben - z.B. >Herr Bong< >Frau Meier<";CHR$(7);
1420 PRINT§1307,;:LINEINPUTDT(7):IFLEN(DT(7))>30THEN1400ELSEIFWA=7THEN1430
1430 N=0:FORZ1=1TO7:N=N+1:K(N)=DT(N):NEXTZ1:PRINT§1600,CY;:GA="":GOSUB1600:PRINT§1857,"Sind die obigen Eingaben richtig   < J/N > ?";CHR
$(7);
1440 GA=INKEY$:IFGA="j"THENRETURNELSEIFGA="n"THEN1450ELSE1440
1450 WA=0:GOSUB1600:PRINT§1858,"Geben Sie die zu ändernde Zahl ein  < 1-7 >";CHR$(7);
1460 WA=VAL(INKEY$):IFWA=0ORWA>7THEN1460
1470 IFWU=2THENN=0:FORZ2=1TO7:N=N+1:K(N)=DX(N):NEXTZ2
1480 N=WA:PRINT§1602,"Alte Eingabe ----->>>>>  ";K(N);BN;BN;:DT(N)=""
1490 ONWAGOTO1200,1250,1280,1310,1340,1370,1400
1500 RESUME1510
1510 ONERRORGOTO0:GOSUB770:CLOSE:GOTO360
1520 '#--------- Datei öffnen ------------------------#
1530 LQ=0:OPEN"R",1,"JOHADRES/DAT:6":X1=0:FIELD1,30ASDX(1),30ASDX(2),30ASDX(3),30ASDX(4),30ASDX(5),16ASDX(6),30ASDX(7)
1540 LQ=LOF(1):IFWP=9THENRETURNELSELQ=LQ+1:PRINT§141,USING":####";LQ;:RETURN
1550 '#--------- Daten auf 0 setzen ------------------#
1560 FORQ=1TO7:DX(Q)="":NEXTQ:FORQ=1TO7:DU(Q)="":NEXTQ:FORQ=1TO7::DT(Q)="":NEXTQ
1570 KN="":KV="":KU="":KS="":KO="":KW="":KP="":RETURN
1580 '#---- Rechte Textspalte auf 0 setzen -----------#
1590 X6=298:FORX=1TO18:PRINT§X6,STRING$(20,32);:X6=X6+80:NEXTX
1600 PRINT§1840,CY;:RETURN
1610 '#------------ Texteingabe ---------------------#
1620 PRINT§540,"Nach der Eingabe";:PRINT§619,"< ENTER >  drücken";:PRINT§778,"Mit < / > kommen Sie";:PRINT§860,"zurück ins Menü!";:RETUR
N
1630 '#------------ Sortierprogramm -----------------#
1640 CLEAR5000:DEFSTRA-G:CV=STRING$(19,140):CW=STRING$(54,140)
1650 CLS:PRINT§131,"Sortieren";
1660 ONERRORGOTO1690
1670 KILL"JOHADRES/BAK:6":GOTO1710
1680 GOTO1710
1690 RESUME1700
1700 OPEN"R",1,"JOHADRES/BAK:6":CLOSE:GOTO1660
1710 CLOSE:ONERRORGOTO0:PRINT§322,CW;:PRINT§402,"Das gesamte  JOHADRES/DAT-File  wird nun bereinigt.Die":PRINT§482,"alten und neuen Date
n werden alphabetisch sortiert und":PRINT§562,"     erhalten eine neue fortlaufende Datennummer.":PRINT§642,CW;
1720 PRINT§379,CV;:PRINT§459,"Daten einlesen";:PRINT§539,"LOF-Datensatz:";:PRINT§619,"Daten-Nummer :";:PRINT§699,CV;
1730 CMD"N JOHADRES/DAT:6 JOHADRES/BAK"
1740 OPEN"R",1,"JOHADRES/BAK:6":OPEN"R",2,"JOHADRES/DAT:6"
1750 FIELD#1,90ASX$      '#>> Länge des zu sortierenden Blocks <<#
1760 L1=LOF(1):L2=L1:PRINT§553,L1;      '#>> Dateilänge alt/neu <<#
1770 DIMN$(L1),NS(L1):FORN=1TOL1:GET1,N:PRINT§633,N;
1780 Z$=MID$(X$,61,30)      '#>> Sortieren nach Familiennamen bzw.Firmenbez.
1790 N$(N)=Z$      '#>> Sortierkriterium <<#
1800 NS(N)=N      '#>> Zugehörige Sektor-Nummer <<#
1810 IFLEFT$(N$(N),1)=" "THENN$(N)=CHR$(255):L2=L2-1      '#>> Gelöschter Eintrag <<#
1820 GOSUB1890:NEXTN:CMD"O",L1,N$(1),NS(1)
1830 PRINT§813,"Das Sortierprogramm ist beendet.";:PRINT§891,"Das neue Datenfile wird geschrieben.";:PRINT§962,CW;
1840 PRINT§779,"Daten speichern";:PRINT§859,"LOF-Datensatz:";L2;:PRINT§939,"Daten-Nummer :";:PRINT§1019,CV;
1850 FIELD#2,255ASY$:FIELD#1,255ASX$:FORN=1TOL2
1860 GET1,NS(N):PRINT§953,N;:LSETY$=X$      '#>> Nach sortierter Reihenfolge <<#
1870 PUT2,N:NEXTN:CLOSE      '#>> Neue Datei ist richtig sortiert <<#
1880 CLEAR50:RUN
1890 IFINSTR(N$(N),"Ä")THEN1960
1900 IFINSTR(N$(N),"Ö")THEN1970
1910 IFINSTR(N$(N),"Ü")THEN1980
1920 IFINSTR(N$(N),"ä")THEN1990
1930 IFINSTR(N$(N),"ö")THEN2000
1940 IFINSTR(N$(N),"ü")THEN2010
1950 IFINSTR(N$(N),"ß")THEN2020ELSERETURN
1960 A="Ae":D="#Ä":H=INSTR(N$(N),"Ä"):GOSUB2030:GOTO1920
1970 A="Oe":D="#Ö":H=INSTR(N$(N),"Ö"):GOSUB2030:GOTO1920
1980 A="Ue":D="#Ü":H=INSTR(N$(N),"Ü"):GOSUB2030:GOTO1920
1990 A="ae":D="#ä":H=INSTR(N$(N),"ä"):GOSUB2030:RETURN
2000 A="oe":D="#ö":H=INSTR(N$(N),"ö"):GOSUB2030:RETURN
2010 A="ue":D="#ü":H=INSTR(N$(N),"ü"):GOSUB2030:RETURN
2020 A="ss":D="#ß":H=INSTR(N$(N),"ß"):GOSUB2030:RETURN
2030 B=LEFT$(N$(N),H-1):C=MID$(N$(N),H+1):N$(N)=B+A+C+D:RETURN
```

..Hardware ..Hardware ..Hardware

D/A-Wandler selbstgebaut

Musikwiedergabe mit dem PC muß nicht teuer sein. Wenige Bauteile schaffen eine kleine Alternative zu Soundkarten.

Der Ton macht die Musik

Nichts mehr hören und sehen. Das Gekreische und Gepiepse einfach abschalten und der wohltuenden Stille lauschen. Diesen Wunsch hegt bestimmt jeder, der schon mal versucht hat, Musik-Files über den PC-Lautsprecher wiederzugeben. Denn schöne Töne gehören nicht zu dessen Stärken. Unüberhörbar sind die Unzulänglichkeiten des winzigen Lautsprechers – Tonqualität läßt er schmerzlich vermissen, von ausreichender Lautstärke ganz zu schweigen.

Das sind Schwächen, mit denen die hier vorgestellte Schaltung Schluß macht. Sie wird an der parallelen Schnittstelle betrieben und wandelt die digitalen Daten in analoge Signale um – erheblich besser, als es der im Rechner vorhandene Baustein nebst Lautsprecher tun könnte.

Das Herzstück der Platine bildet ein DAC-Baustein (DAC = Digital/Analog-Converter). Er stellt die benötigten analogen Signale zur Verfügung, die dann über eine leistungsfähige HiFi-Anlage oder Aktivbox wiedergegeben werden können. Das schont nicht nur das Gehör, sondern nützt auch dem Geldbeutel.

Rund 25 Mark kostet der Eigenbau in Mono-Ausführung. Die Stereo-Version schlägt dagegen mit 50 Mark zu Buche. Dann allerdings kann sich die Schaltung mit so mancher Soundkarte messen. Herkömmliche Soundkarten, die ebenso wie die einfache Ausführung des Eigenbaus nur Mono-Wiedergabe erlauben, läßt sie jedoch weit hinter sich.

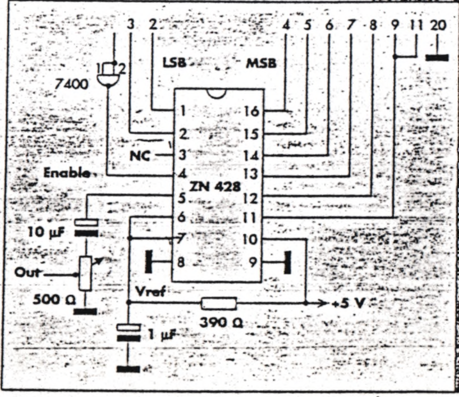
So bietet beispielsweise die alte Soundblasterkarte von der Firma Creative Labs ebenfalls nur 8-Bit-Wandlung. Mit der neuen Soundblasterkarte, die Stereoklang und 8-Bit-Wandlung pro Kanal beherrscht, kann der Eigenbau ebenfalls konkurrieren. Die Klangqualität einer Next-Workstation jedoch, die mit einem 16-Bit-Wandler für jeden Kanal CD-Sound bietet, erreicht sie nicht.

Mitentscheidend für Tonqualität und Dynamik ist die Abtastfrequenz. Und in diesem Punkt besitzt die analoge Technik einige Schwächen: So bereiten ihr Höhen und leise Passagen Probleme; denn je höher der Ton, desto höher muß die Abtastfrequenz sein. Hier stoßen herkömmlichen Schaltungen schnell an ihre technischen Grenzen. Die Umwandlung der digitalen Informationen wird ungenau, die Störungen nehmen zu.

Leistungsfähiger D/A-Wandler

Der CHIP-Tip stößt zwar ab 10 kHz Tonfrequenz an seine Grenzen, darunter gibt er aber das Audiosignal relativ sauber und exakt wieder. Für diese Maximalfrequenz müssen aber bereits 20 000 Signale pro Sekunde an dem D/A-Wandler bereitgestellt werden. Soll es höher hinausgehen, benötigt man professionelle Soundkarten. Mit einem Geräuschspannungsabstand von ungefähr 48 Dezibel liegt er jedoch gar nicht schlecht im Rennen. Und leistet ebensoviel wie preiswerte Kofferradios.

Außer an den D/A-Wandler stellt die Wiedergabe digitaler Musikstücke auch hohe Anforderungen an die Rechenkapazität des PC. Der Grund liegt auf der Hand. Um den Schwingungsverlauf auch der hohen Töne möglichst exakt wiedergeben zu können, müssen pro Sekunde 20 000 Werte an den D/A-Wandler ausgegeben werden. Das bedeutet einen enormen Rechenaufwand: 20 000mal je Sekunde muß eine Programmschleife abgearbeitet werden, die aus den Musikdaten die Ausgabewerte berechnet. Das überfordert PC der Klasse 286 in den meisten Fällen.



Der Aufbau der Schaltung ist recht einfach, da sie aus sehr wenigen Bauteilen besteht:

- eine Platine;
- ein DAC (ZN 428);
- ein TTL-Baustein 7400;
- ein Elko mit1Mikro-Farad/16Volt;
- ein Elko mit10Mikro-Farad/16Volt;
- ein Widerstand mit 390 Ohm;
- ein Drehpotentiometer mit 500 Ohm;
- eine Cinchbuchse;
- ein Kabel mit 25poligem Sub-D-Stecker sowie ein Netzteil oder ein Akku.

Im wesentlichen besteht die Schaltung aus zwei integrierten Schaltkreisen, aus dem DAC ZN428, der die digitalen Musiksignale in analoge umwandelt, und dem TTL-Baustein 7400, der ein Steuersignal anpassen muß. Signal 1 des Druckerports wird an Pin 1 und 2 angeschlossen und invertiert an Pin 3 ausgegeben. Wichtig ist: Pin 7 muß mit Masse und Pin 14 mit +5 Volt Versorgungsspannung verbunden sein.

Die 8-Bit-Signale werden über die Anschlüsse 1, 2 und 11 bis 16 über die parallele Schnittstelle an den DAC-Baustein gesendet. Liegen sie richtig an, erfolgt die Freigabe der Daten über die Stromleitung. Das Ausgangssignal kann an Pin 5 abgegriffen werden. Hier verhindert zusätzlich der zwischengeschaltete Elektrolytkondensator (Elko) Störungen zwischen DAC-Baustein und angeschlossenem Verstärker (siehe Schaltplan).

Zusätzlich läßt sich über das Drehpotentiometer der Ausgangspegel variieren. Sollte der angeschlossene Verstärker die Musikstücke nämlich nur verzerrt wiedergeben, so liegt dies häufig daran, daß der Ausgangspegel zu hoch liegt – Abhilfe schafft das Verdrehen des Widerstands. Der Pegel wird an den jeweiligen Verstärkereingang angepaßt, so daß die Störungen entfallen.

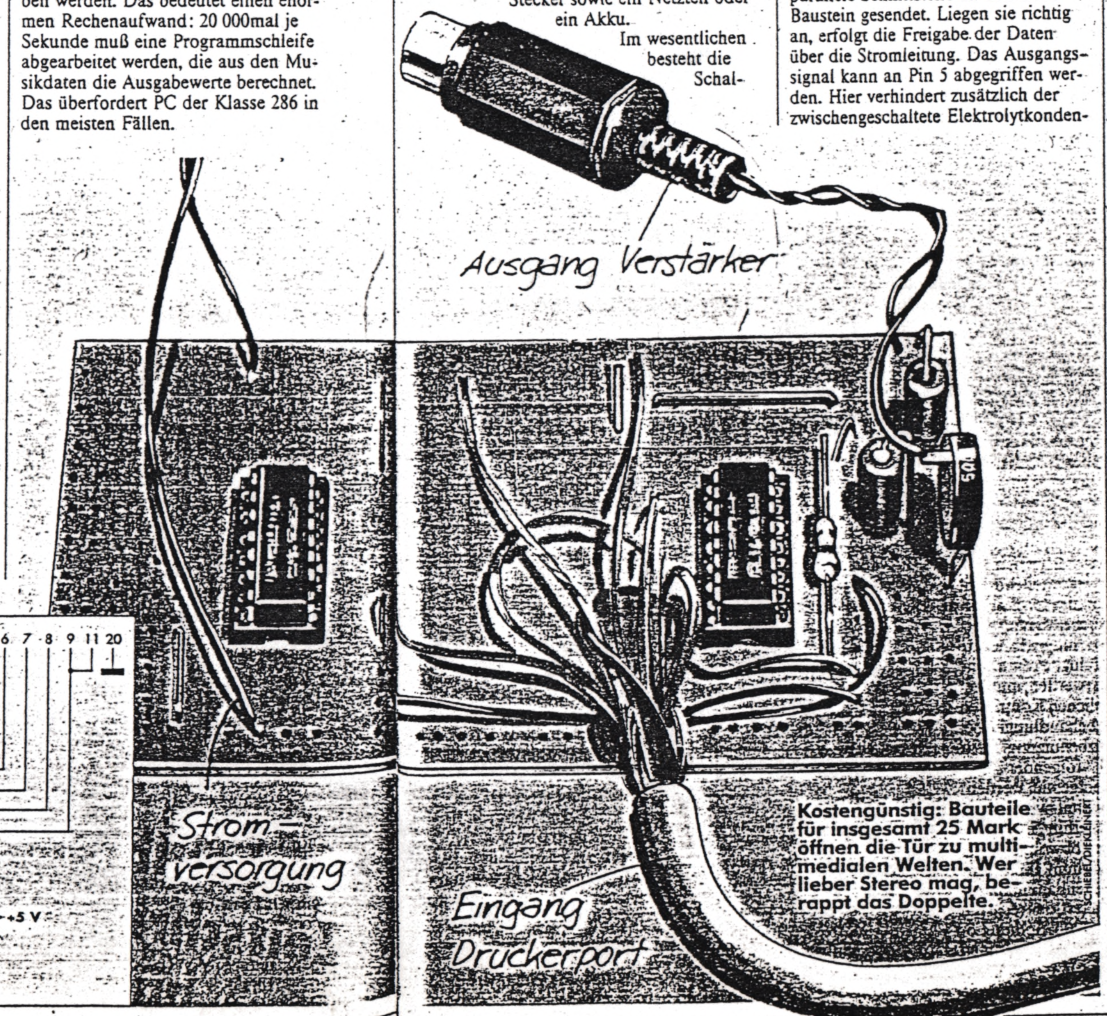
Die Stromversorgung der Schaltung übernimmt entweder ein herkömmliches externes Netzteil (5 Volt, 500 mA) oder ein Akku. Auch die Verbindung zum Druckerport des PC bereitet keine Probleme: Die Anschlußbelegung für den 25poligen Druckerport-Stecker ist im Schaltplan angegeben.

Ausgangspegel muß angepaßt werden

Etwas mehr Aufwand erfordert die Wiedergabe in Stereo: Für jeden Kanal werden je eine Schaltung und Schnittstelle benötigt. Dann allerdings steht dem ungetrübten Klangvergnügen dank zahlreich angebotener Publicmain- und Sharewareprogramme nichts mehr im Wege.

So können etwa die MOD-Player Trackblaster 2.0 von Volker Zinke oder WOW von Jan Ole Suhr direkt den Hardwaretip ansteuern. Sie spielen die breite Palette der vom Amiga bekannten Soundmodule (Mod-Files) auf dem PC ab – egal, ob in Mono oder in Stereo.

Andreas Filip



Kostengünstig: Bauteile für insgesamt 25 Mark öffnen die Tür zu multimedialen Welten. Wer lieber Stereo mag, berappt das Doppelte.

Know-how im Überblick

Hardwaretip
280 D/A-Wandler selbstgebaut
Praxis für Profis
282 Tips für Programmierer mit Ideen für Programmodule in BASIC und Pascal
Workshop
286 Ganz schön übersichtlich: Schneller arbeiten mit Open Access
Tips & Tricks
294 Tips, Tricks und Makros für Standardsoftware

aufen - zu verkaufen - zu verkaufen - zu verkaufen - zu verkaufen - zu ve

GENIE III - 4 MHz - Echtzeituhr - Bildschirm 12" grün entspiegelt = mit 2 Floppy 80/ds/dd 1.4 Mbyte Speicherkapazität

Anschlüsse: RS 232C, Druckerparallel, 2 weitere Laufwerke

Betriebssysteme: G-DOS, NEWDOS80, CPM 2.2

Programmiersprachen: BASIC Level II, Disk-Basic, Assembler, Cobol,Fortra5

Compiler: Microsoft, Z-Basic, Accel/

Anwenderprogramme: Deutsches Handbuch, ADRESS-TEXT 90, GENIE-TEXT, Mini-Daten-Bank-System (MIDABAS), ca. 100 Spiele, ca.100 Utillites

Komplettpreis: VB 650.00 DM

Anfragen: Willi Johnen,5160 Düren,Hansemannstr.1,Tel, 02421/501305

fen - zu verkaufen - zu verkaufen - zu verkaufen - zu verkaufen - zu verk

Zwecks Platzgewinnung habe ich folgende Teile billig abzugeben:

1 * ATARI Megafile 30 für DM350.--

1 * PC-Steckerweiterung PC-Spedd V1.4 für DM100.--
(Nur für ATARI-Besitzer geeignet !)

1 * ATARI GFA-Assembler V1.5 für DM50.--

1 * ATARI GFA-BASIC 2.02 mit Compiler für DM25.--

1 * ATARI GFA-Draft v2.0 (CAD) für DM20.--

1 * Video-Genie EG3003 m. 256KByte RAM über Banker, 3,5 MHz Takt, HRG1B incl. HRG-Basicerw., Monitor u. 2*360K Disklaufwerke, NewDOS80 V2.0 und noch div. andere Software.

Da ich den Kram beim letzten Versuch nicht losgeworden bin, hier nochmal das super Sonderangebot: alles was ich hier genannt habe plus das was sonst noch zum System rumliegt, für DM 100.-- Wegen der vielen Einbauten ist der Rechner jedoch nicht mehr ohne entsprechende Maßnahmen postversandtfähig. Auf Wunsch versuche ich jedoch die Innereien des Rechners (und nur diesen betrifft es) durch Füllmaterial so zu fixieren, daß er nach Empfang noch komplett ist. Der Versandt erfolgt UNFREI !, da das ganze Material mit sicherheit mehrere Packete umfaßt und ich bei dem Angebotspreis nicht auch noch die Post finanzieren will.

Meine Anschrift:

KURT MÜLLER
Sophie-Scholl-Ring 3 b
2054 Geesthacht

Club 80 Börse --- Club 80 Börse

Zehn Jahre ZCPR
von Jay Sage

(Anm des Übersetzers: Dieser Artikel wurde der Ausgabe 54 des "Computer Journal" entnommen und mit freundlicher Genehmigung des Verlages veröffentlicht. Jay Sage ist - was ZCPR betrifft - ein Mann der ersten Stunde. Ich habe mich bemüht, den Artikel so wörtlich wie möglich zu übersetzen. Teilweise ist dabei der Stil etwas auf der Strecke geblieben. Einige Fachausdrücke, bei denen ich davon ausgehen mußte, daß eine Übersetzung sie nur unverständlich machen würde, habe ich übernommen. Hartmut Schulte)

Am 02. Februar 1992 sind genau zehn Jahre seit der Herausgabe der ersten Version des ZCPR vergangen. Mehr oder weniger habe ich mich während der gesamten Zeit damit beschäftigt und ich glaube, es ist erstaunlich, wie kraftvoll die Aktivitäten auf diesem Gebiet noch sind.

Der Herausgeber (des Computer Journal, d.Ü.), Chris McEwen, hatte ursprünglich gehofft, daß wir aus diesem Heft eine spezielle ZCPR-Jubiläumsausgabe mit Beiträgen von einigen der ursprünglichen Entwickler, in erster Linie Richard Conn, machen konnten. Ich wechselte mehrere Briefe mit Richard darüber, er griff die Idee aber leider nicht auf. Wenn ich davon ausgehe, daß ich von all denen, die in der Z-Gemeinde aktiv sind, mit am längsten dabei bin, scheint es mir passend zu sein, daß ich nunmehr den Artikel schreibe.

Die Geschichte des ZCPR

Die Masse des Materials für diese Abhandlung habe ich dem Einführungskapitel meines Buchs "ZCPR33 User Guide" entnommen. Als Echelon meine Version des ZCPR3 als offizielle Version herausgab, wurde selbstverständlich ein begleitendes Handbuch gefordert. Neben all den notwendigen technischen Informationen, z.B., was der neue Command Processor macht und wie er installiert werden sollte, fügte ich zwei für mich sehr wichtige Kapitel hinzu:
eine Feststellung, was ich versuchte, mit ZCPR33 zu erreichen und die Geschichte, die zu seiner Entwicklung führte.
Im Computer Journal schreibe ich oft über die Ziele des Z-Systems, heute geht es darum, einiges aus seiner Vergangenheit zu beleuchten.

ZCPR1

"Was, Du hast noch nichts von ZCPR gehört?" Ich erinnere mich sehr gut daran, daß ich mit diesem Ausruf von einem der Clubveteranen begrüßt wurde, als ich als frischgebackener Computerbenutzer an dem Treffen eines CP/M-Computerclubs teilnahm.
Er konnte es nicht fassen, daß es noch jemanden gab, der das Standard-CP/M benutzte. Bald dachte ich genauso und tue es auch heute noch!
Das ZCPR, das er meinte, würden wir heutzutage ZCPR1 nennen. ZCPR, was soviel bedeutet wie "Z80 Command Processor Replacement", war das Ergebnis der Arbeit einer Gruppe von Computerliebhabern, die sich "The CCP Group" nannte. Sie bestand aus Frank Wancho, Keith Peterson, Ron Fowler, Charlie Strom, Bob Mathias und Richard Conn. Wie wir noch sehen werden, war Richard die treibende Kraft.

(C) 1992 Computer Jurnal
PO Box 12, S Plainfield
NJ 07080 USA
and Jay Sage

Reprintet with Permission

Ron Fowler ist wohl bekannt als der Autor des MEX Kommunikationsprogramms, daß ich immer noch benutze und liebe. Keith Peterson schrieb eine vereinfachte Version von Ward Christensens CBBS, dem ursprünglichen computerisierten Schwarzen Brett. Keith Programm nannte sich MINICBBS, bis heute läuft es auf meiner Z-Node in einer von mir angepaßten Version. Um ehrlich zu sein, ist es überholt, aber es ist für mich eine Art Verbindung zur Vergangenheit, die ich sehr schätze.

Keith Peterson war lange Zeit Sysop eines der besten BBS Systeme im Lande, dem Royal Oak. Obwohl es zur MS-DOS Software expandierte, ließ es CP/M nie außer acht. Vielleicht war es einzigartig, daß sich Anrufer bei Royal Oak sofort am Prompt des Operationssystems wiederfanden. Man brauchte weder Namen noch Passwort. Wollte man MINICBBS benutzen, mußte man es selbst aufrufen.

Frank Wancho ist an der Verwaltung des SIMTEL20-Computers auf dem White Sands Raketenschießstand beteiligt. Dieser Rechner beherbergt ein riesiges Archiv von CP/M-Programmen (und anderen natürlich). Keith arbeitet bei SIMTEL20 an der Bearbeitung der Sammlungen. In den Informationen, die mir Frank via Computer gab, tauchten ihre Namen immer wieder auf.

Um 1981 begeisterte Richard Conn die Gruppe, indem er vorschlug, den CP/M Console Command Processor oder CCP neu zu schreiben, um die Vorteile der leistungsfähigeren und eleganteren Opcodes des neuen Zilog Z80 Mikroprozessors zu nutzen. Da die Mitglieder der CCP-Gruppe weit auseinander wohnten, hielten sie, ebenso wie wir heutzutage, über Computer Kontakt zueinander. Für diesen Zweck stellte Frank Wancho den Zugriff zum Rechner sicher.

Nachdem im CCP etwas Platz geschaffen worden war, konnten die Programmierer einige brauchbare neue Eigenschaften einfügen. Die allerwichtigste war die Schaffung eines Suchpfades für ausführbare Dateien. Ab CP/M 2.0 hatte Digital Research Usernummern eingeführt, jedoch waren sie faktisch wertlos, weil es keine Möglichkeit gab, von einem Userbereich zum anderen Programme laufen zu lassen oder sie anzusprechen. Mit ZCPR und seiner Fähigkeit, automatisch Laufwerk A/ User 0 zu suchen, wurde dieses Problem aus der Welt geschaffen, so daß sich die Möglichkeit ergab, die neugeschaffenen Userbereiche effektiv zu nutzen.

Eine weitere Neuheit des ZCPR war das Kommando GO, daß den sofortigen Neustart des letzten ausgeführten transienten Programms erlaubte, ohne daß dies neu von Diskette geladen werden mußte. Das war in den Zeiten der langsamen Diskettenlaufwerke ein echter Segen. Die residenten Befehle wurden in vielen nützlichen Kleinigkeiten verbessert. So brechen Befehle wie REN oder SAVE unter CP/M einfach ab, wenn sie auf eine schon existierende Zieldatei treffen. Der Nutzer muß dann langwierig diese Datei löschen und wieder von vorne beginnen. Unter ZCPR wird der Nutzer einfach gefragt, ob die alten Dateien überschrieben werden sollen.

Das original ZCPR wurde auf einer Diskette der SIG/M (Special Interest Group/Microcomputers) veröffentlicht, dem Zweig der Amateur Computer Group of New Jersey (ACGN), der für die Verteilung von Public Domain Software verantwortlich war. Es war Ausgabe Nr. 54 vom 02. Februar 1982. Interessanterweise wurde der Artikel, den Sie gerade lesen, zehn Jahre später in der Ausgabe 54 des Computer Journal herausgegeben!

Andere Programmierer verfeinerten ZCPR weiter zum NZCPR (New ZCPR), so daß Ende Oktober 1982 mit der SIG/M Nr.77 die Version 1.6 des NZCPR herausgegeben werden konnte. Schließlich wurde Version 2.1 erreicht, die allerdings nie auf Diskette, sondern nur über Netzwerke verteilt wurde.

Jim Byram von der Boston Computer Society CP/M Group verfaßte eine persönlich verteilte Version des NZCPR, die nur Intel 8080 Codes benutzte und an der man erkennen konnte, daß nicht einfach nur die Anwendung der neuen Z80 Opcodes, sondern vielmehr wirtschaftliches Programmieren der wichtigste Faktor bei der Verbesserung des Command Processors war. Übrigens, ich glaube, es war Jim, der die eingangs erwähnte Frage bezüglich ZCPR an mich stellte. Ich wurde schließlich Leiter dieser Gruppe, die sich mit anderen zusammenschloß und letztendlich die Zi/Tel Group wurde, bei der ich heute der Verantwortliche für CP/M und der Sysop des Schwarzen Bretts bin. Die Gruppe unterstützt CP/M, das Z-System und MS-DOS.

ZCPR2

Obwohl ZCPR1 eine deutliche Verbesserung gegenüber CP/M bedeutete, so war es doch keine revolutionäre Entwicklung. Richard Conn träumte aber von einem wirklich fortschrittlichen Betriebssystem und setzte darum die Entwicklung fort. Fast genau ein Jahr nach dem Erscheinen des ZCPR1 wurde ZCPR2 auf zehn SIG/M Disketten (98-107) herausgegeben, ein beispielloser und großartiger Beitrag zur Public Domain Software.

Der konzeptionelle Fortschritt des ZCPR2 war beachtlich: es nutzte Memoryspeicher im geschützten Memory oberhalb des BIOS zur Speicherung neuentwickelter Module des Betriebssystems. Die Befehlszeile, die vorher Bestandteil des Command Processors war, wurde in einem dieser Puffer gespeichert, so daß sie nicht durch Warmstarts, bei denen eine neue Kopie des Command Processors von Diskette geladen wird, zerstört werden konnte. Auf diese Weise konnten mehrere Befehle auf einer Zeile aneinandergefügt werden.

Der Befehlssuchpfad befand sich ebenfalls in einem dieser Puffer, anstatt ihn fest im Command Processor einzuprogrammieren. Auf diese Weise konnte der Suchpfad jederzeit vom Nutzer geändert werden. Gleichzeitig wurde das Konzept der mit Namen versehenen Directories eingeführt, wobei ein weiterer Memorypuffer zur Aufnahme des Namensindex dient.

Viele uns heute wohlvertraute Utilities erschienen zum ersten Mal unter ZCPR2, so z.B. ZEX, WHEEL, HELP, PATH, PWD, MKDIR und MENU. Das letztgenannte Programm nutzte ein neues Shellkonzept, indem es jedesmal, wenn es einen Befehl in den Befehlszeilenpuffer schrieb, seinen Namen an das Ende der Befehlsfolge anfügte, so daß nach Abarbeitung der Befehle die Kontrolle wieder an MENU überging. Dieses Prinzip funktionierte hervorragend, solange es nicht mehr als eine Shellebene gab. Unter ZCPR2 wurde auch die erweiterte Befehlsverarbeitung eingeführt.

Alleine die ZCPR2-Dokumentation war größer als ein halbes Megabyte! Sie bestand aus einem konzeptionellen Handbuch, einem Installationshandbuch, einem Benutzerführer und einem Handbuch, das die Grundprinzipien erklärte und wohl entstand, weil Rick sich und uns beweisen mußte, daß es sich bei dem Ganzen nicht um eine Marotte handelte.

Kurz nach der Herausgabe durch die SIG/M erschien Version 2.3 auf Diskette 108. Bisher war ZCPR2 den Traditionen des ZCPR1 gefolgt und nutzte Zilog Opcodes. Da die Eigenschaften des ZCPR2 jedoch so überzeugten, wollten immer mehr Besitzer von Rechnern, die mit Intel 8080 und 8085 Mikroprozessoren arbeiteten, am neuen System teilhaben. Charlie Strom, ein Mitglied der ursprünglichen CCP Group und später wohlbekannt als Sysop der Compuserve CP/M Special Interest Group, schrieb den Kode des Command Processors und einige wichtige Utilities um in einen Intel-kompatiblen Kode und veröffentlichte das Ergebnis auf SIG/M 122. Ob Sie es glauben oder nicht, zu jener Zeit benutzte ich bei meiner Arbeit ein Intel MDS-800 miroprozessorgestütztes Entwicklungssystem, eben das, für das Gary Kildall während seiner Zeit bei Intel CP/M erfand und ich erinnere mich gut an das Erscheinen der 8080-Version des ZCPR2. Es war großartig!

ZCPR3

Doch mit ZCPR2 endete keineswegs die Entwicklung. Am französischen Nationalfeiertag, dem 14. Juli 1984, noch nicht ganz eineinhalb Jahre nach Herausgabe des ZCPR2, bot Richard Conn eine Version 3 in Form von weiteren neun SIG/M-Disketten (184-192) an. Bis zu diesem Zeitpunkt stammten mehr als 10% aller Software, die die SIG/M je herausgebracht hatte, aus der Feder eines Mannes - Richard Conn!

Während eines Gesprächs mit Richard muß ich wohl meine Bewunderung über diese unglaubliche Menge an Software, die er geschrieben und veröffentlicht hat, geäußert haben. Richards Antwort beeindruckte mich ebenso. Er sagte, daß die Programme, die andere veröffentlicht hatten, ihn soviel gelehrt und ihm so geholfen hatten, daß er sich selbst weit über das normale Maß hinaus verpflichtet fühle, das an die Gemeinschaft zu verteilen, was er geben können. Das hat er mit Sicherheit getan! Und von genau diesem Geist ist die 8-Bit-Gemeinde durchdrungen.

ZCPR3 brachte sowohl bedeutsame neue Konzepte als auch größere Verfeinerungen. Drei wesentliche Neuerungen waren flow control, error handling und der message buffer.

Flow control erlaubte einen erheblich höheren Grad der automatischen Verarbeitung, weil der Command Processor nicht mehr vom Bediener und seinen Eingaben abhängig war, sondern weitreichende Entscheidungen selbst treffen konnte. Der message buffer schuf die Möglichkeit zur Kommunikation zwischen dem Command Processor und den Programmen bzw zwischen nacheinander ablaufenden Programmen.

Mit error handlern war es möglich, falsch eingegebene Befehle zu korrigieren, eine wichtige Hilfe im Zusammenhang mit Mehrfachbefehlen in einer Zeile. Es war schon schlimm genug, nach einem Fehler einen Befehl neu zu tippen, aber wegen eines einzigen Fehlers eine lange Befehlsfolge neu eingeben zu müssen, konnte einem die Freude an der Nutzung der Mehrfachbefehle schnell nehmen.

Übrigens war ZCPR3 anders als seine Vorgänger so geschrieben, daß es sowohl mit Intel als auch mit Zilog Opcodes assembliert werden konnte. Im ersten Fall war der Code verständlicherweise länger und es konnten weniger Möglichkeiten eingebaut werden, aber er konnte auf einem 8080- oder 8085-Rechner laufen.

ZCPR31

Im März 1985 begann jene Kette von Verbesserungen des ZCPR3, die zur Version 3.3 führte, mit meiner privaten Experimentalversion ZCPR31, die ich auf meiner Z-Node nutzte. Sie war so abgewandelt, daß der Command Processor die Werte für das letzte Laufwerk und den letzten Userbereich vom Environment Descriptor (dazu später mehr) erhielt.

Dies war meine erste Begegnung mit dem Kode des Operationssystems, vor dem ich bisher, wie bestimmt viele andere auch, großen Respekt hatte. Das Wort "Operationssystem" hat etwas Mystisches, das einen vermuten läßt, daß nur die allererfahrensten Programmierer eventuell den Kode verstehen könnten. Tatsächlich stellte ich dann fest, daß er sich gar nicht so sehr von dem ordinärer Utilityprogramme unterschied. Zu meinem Erstaunen war ich in der Lage, lauffähige Änderungen durchzuführen, die den CCP verbesserten. Die offensichtlichsten Fortschritte machte ich im August 1985, als mir gleich drei Verbesserungen gelangen.

Als erstes unterbrach ich die Endlosschleife, in die ZCPR30 geriet, wenn der vorher aufgerufene Error Handler nicht gefunden wurde, vielleicht, weil der Suchpfad oder der Programmname geändert worden war. In dieser Situation würde ein Befehlsfehler den Error Handler aufrufen. Wenn dieser nicht gefunden würde, würde dies einen weiteren Fehler bewirken, der wiederum den Error Handler aufrufen würde und so weiter, bis jemand die Resettaste drückt oder den Strom ausschaltet.
Zweitens änderte ich den Kode so, daß er aus dem Environment die Adressen der RCP-, FCP- und NDR-Module entnehmen und so dynamisch auf Änderungen reagieren konnte.
Schließlich schrieb ich den Kode so um, daß der erweiterte Command Processor die Kontrolle an den Command Processor zurückgeben konnte, falls er nicht in der Lage war, das Kommando auszuführen. In diesem Fall rief der Command Processor dann den Error Handler auf. Damit war der erweiterte Command Processor als Erweiterung des CCP tatsächlich flügge geworden und ein ZCPR3-System konnte Vorteile sowohl aus der erweiterten Befehlsverarbeitung als auch aus dem Error Handling ziehen, das nunmehr auch von normalen Programmen eingeleitet werden konnte.

Im Januar 1986 wurden dann erste Schritte unternommen, um einige schwerwiegende Fehler bei der Berechnung des kürzesten Suchpfads zu beseitigen. Leider lösten wir damit andere Fehler aus, so daß es Howard Goldstein erst im Juni 1986 gelang, eine vollständige und saubere Lösung anzubieten.

Zum nächsten größeren Verbesserungsschub kam es im März 1986, als Al Hawley, Sysop der Z-Node #2 und heute ein bekannter Autor des "Computer Journal", verschiedene neue Ideen einbrachte. Eine davon war ein neuer Ansatz, wheel-geschützte Befehle (Befehle, die nur ein berechtigter Nutzer geben darf) einzufügen. Unter ZCPR30 wurde dieser Schutz direkt in den Command Processor (und den RCP) einprogrammiert, so daß eine Fehlermeldung erschien, wenn ein eingeschränkter Befehl bei ausgeschaltetem Wheel gegeben wurde. Al schlug vor, das höchstwertige Bit des ersten Buchstabens des Befehls zu setzen, um anzuzeigen, daß dieser Befehl für die Allgemeinheit unzulässig war.

Dieses Konzept hatte mehrere wichtige Vorteile: Erstens verkürzte sich der Kode. Zweitens wandte der neue Kode automatisch die selbe Technik bei Befehlen des residenten und flüchtigen Befehlsspeichers (RCP und FCP) an, so daß der Kode, der die Zugriffsberechtigung überprüfte, aus diesen Modulen verschwinden konnte.

Drittens verschwanden geschütze Programme einfach, soweit es den Command Processor betrifft, statt eine Fehlermeldung auszulösen. Damit konnten transiente Programme oder Alias mit dem selben Namen wie der geschützte residente Befehl automatisch eingreifen und das tun, was der Systemverwalter in einem solchen Fall von ihnen wünschte.

Al Hawley führte auch zwei Konzepte ein, die die Behandlung geschützter Systeme einfacher machten. Er ermöglichte es dem Command Processor, flexibel festzulegen, ob er die DU: Form des Inhaltsverzeichnisses entsprechend der gesetzten DUOK Flag im Environment erkennen sollte oder nicht und er umging die Überprüfung des Paßworts, wenn das Wheelbyte gesetzt war. Diese beiden Neuschöpfungen erleichterten Sysops und Systemergänzern den Umgang mit gesicherten Systemen, ohne es dem eingeschränkten Nutzer leichter zu machen.

Der letzte größere Fortschritt bei der Entwicklung des ZCPR31 entsprang einer Unterhaltung, die ich im Juli 1986 mit Bruce Morgen hatte. Wir diskutierten über die lästige Art, wie ZEX in Shells funktionierte, weil das Shellprogramm für jede Befehlszeile neu geladen wurde, nur um festzustellen, daß ZEX gerade lief und dann die nächste Befehlszeile aus dem ZEX in den Mehrfachbefehlszeilenpuffer (multiple command line buffer) zu laden. Ich wechselte eine Kleinigkeit im Kode und das Problem verschwand wie der Blitz.

ZCPR33

In den letzten Tagen des Januar 1987 erhielt ich einen Anruf von Echelon. Richard Conn wollte nicht mehr am ZCPR3 weiterarbeiten und Echelon fragte mich, ob ich auf der Basis meiner Experimentierversion 31 an einer offiziellen Version 3.3 weiterarbeiten wollte. Ich stimmte zu. Von Februar bis April 1987 war ein enormer Aufwand an zusätzlichen Entwicklungen zu bewältigen, deren Ergebnisse ich im Detail im ZCPR33 User Guide beschrieben habe. Hier will ich nur einige Kernpunkte erwähnen.

Die Entscheidung, nicht länger 8080/8085-Rechner zu unterstützen, fiel. Der Kode wurde in Zilog mnemonics geschrieben und es wurde intensiver Gebrauch von Z80-spezifischen Befehlen einschließlich relativer Sprünge, Blockbewegungen, direkter Wortübertragungen an Registerpaare außer HL, 16-Bit Subtraktionen und das alternative Registerset gemacht. Diese Entwicklung wurde bis zum heutigen ZCPR34 fortgeführt. Soweit ich weiß, hat niemand auch nur versucht, eine 8080-Version zu schreiben - es gibt ja auch nicht mehr allzu viele Rechner dafür.

Eine der schönsten Eigenschaften des ZCPR33 war die automatische Installation von Programmen. Bis zu diesem Zeitpunkt mußte auch ein Programm, das später unter ZCPR laufen sollte, auf das spezielle System eingestellt werden. Falls das vergessen wurde, konnte sich ein Programm unter Umständen ganz komisch benehmen, was sowohl junge als auch erfahrene Nutzer in Schwierigkeiten bringen konnte.

Unter ZCPR2 war die Installation ein aufwendiges Verfahren, bei dem ein großer Kodeblock mit GENINS gepatcht werden mußte. Unter ZCPR3 wurden Informationen über die Systemzusammensetzung in einem Puffer gespeichert (dem sogenannten Environment oder ENV), zu dem alle Programme Zugriff hatten. Oder was noch wichtiger war: die Konfiguration des Systems konnte geändert werden, ohne gleichzeitig alle Programme ändern zu müssen. Das Patchen der Programme konnte auf das Einfügen der ENV-Adresse reduziert werden.

Sobald ich hörte, daß Richard Conn einen Weg gefunden hatte, um diesen lästigen Installationsschritt aus der Welt zu schaffen, war mir die Lösung auch klar. Wenn der Command Processor schon das Programm von der Diskette lädt und er gleichzeitig auch die ENV-Adresse kennt, warum kann er dann nicht gleich automatisch die Adresse in das geladene Programm schreiben? Das war doch die Lösung...

Eine wirklich revolutionär neue Idee wurde mit ZCPR33 eingeführt. Bisher wurden unter CP/M alle transienten Programme ab der Standardadresse 100H geladen und von dort aus gestartet. Unter CP/M gab es keinen Grund, dies zu ändern, wohl aber unter ZCPR3. Schon seit ZCPR1 hatte ich mich sehr daran gewöhnt, Programme mit GO wiederzustarten. Umso mehr verwunderte es mich, daß GO unter ZCPR3 manchmal merkwürdige Ergebnisse lieferte.

Unter CP/M werden Programme nur geladen, wenn das System angewiesen wird, diese laufen zu lassen. Unter ZCPR3 jedoch wurden eine ganze Menge von Programmen automatisch vom Command Processor geladen und ausgeführt. Dazu gehörten erweiterte Command Processoren, Error Handlers, Shells und transiente COM-Versionen ansonsten residenter Befehle, wie z.B. ERA oder REN. So kam es ab und zu, daß diese Programme statt der vom Nutzer erwarteten wiedergestartet wurden, wenn man GO eingab.

Als ich eines Tages am ZCPR33-Kode arbeitete, bemerkte ich, daß es nur einer Kleinigkeit bedurfte, damit der Command Processor eine Datei in eine andere Adresse als 100H lud. Das, so war mir klar, konnte das Problem mit dem GO lösen. Anwenderprogramme konnten, wie üblich, bei 100H geladen werden, aber automatisch durch den Command Processor aufgerufene Programme konnten in eine höhere Adresse wie z.B. 8000H deponiert werden. Auf diese Weise wurden Anwenderprogramme nicht überschrieben und GO könnte sie jederzeit wieder laufen lassen.

Es gab noch eine weitere Gruppe von Neuerungen. ZCPR30 bot eine Anzahl an Sicherheitsmaßnahmen, die es besonders geeignet für Datenfernübertragungssysteme (BBS) machten. Das sogenannte Wheelbyte steuerte den Zugriff sowohl zu residenten als auch zu transienten Programmen. Beliebig veränderbare Grenzen im Bereich der Laufwerke und Usernummern und Inhaltsverzeichnisse mit Namen und Paßwort konnten Anrufer aus bestimmten Bereichen fernhalten.

Diese Vorsichtsmaßnahmen ermöglichten es Nutzern, das System direkt vom Befehlszeilenprompt aus fernzusteuern, dies im scharfen Gegensatz zu MS-DOS Systemen, bei denen ein Nutzer, der erstmal beim Prompt angelangt ist, frei über Nutzung oder auch Zerstörung des Systems entscheiden kann.

War das Sicherheitssystem voll aktiviert, konnte es unter ZCPR30 allerdings auch ein unnützes Ärgernis sein. So konnte man sich Situationen vorstellen, in denen ein Nutzer zwar mit einem Namen in ein Directory kam, weil es kein Passwort hatte, jedoch die Möglichkeit, nur mit Laufwerk und User hineinzukommen, versperrt blieb, weil es hier außerhalb der erlaubten Reichweite lag. Unter ZCPR33 ist ein Directory auch mit Laufwerk/User erreichbar, wenn es unter einem Namen angesprochen werden kann.

ZCPR34

Der augenblickliche Bearbeitungsstand des ZCPR Command Processors ist Version 3.4. Die erste Herausgabe erfolgte irgendwann im März 1988 zusammen mit NZCOM und Z3PLUS; sie wurde im Heft 32 des Computer Journal beschrieben. Im Vergleich zu ZCPR33 war es ein Fortschritt, eine Verfeinerung. Es gab keine radikal neuen Ideen wie beim ZCPR33. Trotzdem waren die Änderungen bedeutsam und nützlich. Seit der ersten Veröffentlichung gab es mehrere kleinere Überarbeitungen und Erweiterungen.

Eine dieser Erweiterungen unter ZCPR34 war die Schaffung eines erweiterten environment descriptors ("Umgebungsbeschreibers"). Nach Entfernung einiger unwichtigen Details fügten wir neue Informationen hinzu. Die wichtigste war die Festlegung einzelner Laufwerke mit einem 16-bit Wort. Schon immer gab es im ENV ein Byte, das das letzte in einem System ansprechbare Laufwerk festlegte. Dies reichte jedoch für ein System, in dem nur einzelne Laufwerke, die nicht fortlaufend durchbuchstabiert waren, nicht aus, z.B. A:, B: und E:. Mit dem neuen Wort konnte jetzt exakt jedes Laufwerk definiert werden, indem "sein" Bit gesetzt wurde.

Der neue ENV enthält auch die Adressen des CCP, BDOS und BIOS und die Größen der ersten beiden. Dies ist eine Maßnahme für die Zukunft, wenn wir uns nicht mehr unbedingt an die Standardgrößen des alten CP/M halten werden. Hal Bower und Cam Cotrill, haben schon im Rahmen der Entwicklung eines neuen gebankten ZSDOS (das kurz vor der Veröffentlichung steht) mit einem CCP experimentiert, der größer ist als die üblichen 2K und einem gebankten DOS, das deutlich kleiner sein wird als die üblichen 3.5K.

Ich erwähnte schon, daß unter ZCPR33 jedes Directory, das unter einem Namen erreichbar ist, auch über DU: aufzurufen ist, obwohl es außerhalb der vom Environment gesetzten Grenzen liegt. Unter ZCPR34 war die Symetrie vollständig. Falls hier ein paßwortgeschütztes Directory mit dem DU: frei angesprochen werden kann, wird das Paßwort ignoriert. Somit können Directories jetzt immer auf beide Arten erreicht werden.

Das Interface des erweiterten Command Processors wurde liberalisiert, so daß Kommandos, die früher nicht erlaubt waren und einen sofortigen Fehler reproduzierten, wie z.B. Jokerzeichen ("?" oder "*") oder bestimmte Dateitypen, an den ECP weitergegeben werden. Damit kann etwa das ALIAS.CMD, das Alias für den für ARUNZ erweiterten Befehlsprozessor definiert, solche Befehle wie "?" enthalten, mit dem dann HELP aufgerufen würde.

Der offensichtlichste Fortschritt unter ZCPR34 war die Unterstützung der sogenannten Typ 4-Programme. Typ 3-Programme werden, wie ich schon ausführte, an eine andere Adresse als 100H geladen, aber diese Adresse ist dann auch für das Programm festgelegt. Für mich war beim Schreiben des ZCPR33 schon klar, daß es ideal sein würde, wenn die zu ladende Adresse erst beim Laden durch den CCP festgelegt würde.
Ich entschloß mich damals für den sehr einfachen Kode, der für Typ 3-Programme ausreichte. Mit diesem Kompromiß war Joe Wright nicht einverstanden und er schrieb darum eine erste Version für Typ4-Programme, der den Kode automatisch an das obere Ende des freien Memories setzte. In echter Gemeinschaftsarbeit polierten wir so lange, bis es wirklich schön funktionierte und brauchten dann dem Command Processor nur sehr wenig Kode hinzuzufügen.

Das Geheimnis lag darin verborgen, daß Joe ein PRL (page relocateable) Programm für ausführbare Dateien anwendete. Einzelheiten hierzu stehen in der Ausgabe 32 des Computer Journal. Die normale PRL-Datei beginnt mit zwei 128-Byte Header Records und ich schlug vor, daß der zum Berechnen der Adresse und zum Verschieben benötigte Kode hier statt im Command Processor stehen sollte. Joe fand einen wunderbaren Weg, dies in die Tat umzusetzen.

Hierdurch wurde nicht nur der Kode des CCP kürzer, sondern das Typ-4 Programm wurde flexibler, weil es vom Command Processor unabhängig war. In meinem nächsten Artikel im Computer Journal werde ich über die ersten oben beschriebenen Laderoutinen für Typ 4-Programme schreiben. Sie können vom Nutzer in jedes bestehende Type 4-Programm installiert werden. Auch dies ist wieder ein wunderschönes Beispiel für den modularen Aufbau, der eines der herausragenden Kennzeichen des ZCPR ist.

Weiter werde ich über die letzte Überarbeitung des ZCPR34, die Version 3.4E berichten. Vorbereitet wurde diese Version von Howard Goldstein, indem er eine Anzahl neuer Ideen einfügte, am erwähnenswertesten ein kleiner Wechsel im Typ 4-Lader, der das Ganze noch flexibler macht.

So entwickelt sich nach zehn Jahren, einer Ewigkeit für die Computerindustrie, ZCPR - das Konzept, das Richard Conn entworfen hat - immer noch weiter und fordert immer noch die Kreativität von Programmierern und Nutzern gleichermaßen. Wie immer können solche Entwicklungen nur in einer großen Gemeinschaft von Menschen wachsen, die willens und begierig sind, ihre Ideen miteinander zu teilen.

(Eine letzte Anmerkung des Übersetzers: So, das war die Übersetzung eines - hoffentlich nicht nur im Original - wirklich interessanten Artikels aus dem "Computer Journal". Wer jetzt wissen möchte, wo und wie man dieses Journal beziehen kann oder was so komische Abkürzungen und Begriffe wie "RCP", "Wheel" oder "Error Handler" bedeuten, der sollte sich mit Vorstandsmitgliedern unseres Clubs oder aber mit mir in Verbindung setzen. Ich würde auch gerne wissen, ob Bedarf an weiteren Übersetzungen oder Abhandlungen über ZCPR besteht - so einfach ins Leere zu arbeiten, ist doch etwas zu frustierend. Hartmut Schulte, Tel 05173/1248)

Deterministisches Chaos als ergänzende Betrachtungsweise

Eckhard Freuwört, Seesen · Teil 2 · (Teil 1 CLB-Heft 5/90)

Fraktale Geometrie

Die nichtlineare fraktale Geometrie erregt seit den siebziger Jahren mit der Abbildung der nach ihrem Entdecker *Benoit Mandelbrot* benannten, aus Julia- und Cantor-Mengen bestehenden Mandelbrot-Menge Aufsehen. Diese Struktur errechnet sich rekursiv nach

Gl. 19 $Z_{i+1} = Z_i^2 + W$ (Abb. 1).

Z und W sind dabei komplexe, d. h. aus Real- und Imaginäranteil bestehende Zahlen, wobei W den Kontrollparameter der Rekursion bildet. Jede Zahl „W", die den Betrag des Ergebnisses nach unendlich vielen Iterationsschritten endlich läßt (der Realanteil von „W" liegt dabei etwa zwischen -1,6 und +1,6 und der Imaginäranteil nimmt Werte etwa zwischen -1,2 und +1,2 an), gehört zu einer zusammenhängenden Julia-Menge (nach dem französischen Mathematiker *Gaston Julia*, 1893–1978, benannt). Bildet man diese Zahl „W" der Iterationsfolge von Gl. 19 nun als Computergraphik in der Ebene der komplexen Zahlen ab (wobei z. B. für jedes „W" einer zusammenhängenden Julia-Menge ein schwarzer Punkt gesetzt wird), so erhält man die Mandelbrot-Menge [17, 18]. Betrachtet man die Ränder dieser Menge vergrößert, so findet sich eine zur Umgebung unendlich dünne Trennlinie – eine scharfe, definierte Trennung ist jedoch nicht vorhanden. Hingegen tauchen immer wieder kleinere, der Mandelbrot-Menge ähnliche Strukturen auf. Diese Eigenschaft der Selbstähnlichkeit wird als S k a l e n i n v a r i a n z bezeichnet und stellt Symmetrie in verschiedenen Maßstäben dar, wobei man ständig auf dieselben Grundelemente trifft. In diesem Sinne bildet die M a n d e l b r o t -Menge einen Bildspeicher von unerhörten Ausmaßen, denn relativ geringfügige Variationen von Gl. 19, etwa

Gl. 20 $Z_{i+1} = Z_i^3 + W$

erlauben die rekursive Darstellung von Bildern durch eine einzige Formel – so führt Gl. 20 beispielsweise zum Bild einer Radiolarie [19]. Die Fachrichtung des „Artificial Life" der Biologie wendet solche Rekursionen u. a. zur Computersimulation von Wachstums- und Evolutions-Prozessen an [20]. Die damit erzeugten Abbildungen von Lebewesen werden B i o m o r p h e genannt.

Doch zurück zur Skaleninvarianz. Schon bei der Untersuchung von Turbulenzen bzw. Konvektionsströmungen in der Physik und bei der Bildung von Rohrleitungsverkrustungen oder der Entwicklung von Schlamm-Rinnen auf Behälterböden in der chemischen Technik treten selbstähnliche Strukturen auf. Die Bezeichnung für diese Strukturen lautet Fraktale; sie werden mit den Methoden der linearen fraktalen Geometrie untersucht. Zu diesem Zweck verwendet man als Maß für die Selbstähnlichkeit die sogenannte Hausdorff-Besicovitch-Dimension (*Felix Hausdorff*, 1868–1942, war deutscher Mathematiker), die häufig der Fraktaldimension gleichgesetzt wird. Das sich von dem englischen Wort „fraction" und dem lateinischen Verb „frangere" für Bruch bzw. Brechen ableitende Wort F r a k t a l weist bereits darauf hin, daß die besagte Dimension nicht ganzzahlig ist.

Die Bestimmung dieser Maßzahl soll nun näher erläutert werden: Wie bereits mit Gl. 15 und Gl. 18 angedeutet wurde, bewegt sich der Endwert eines chaotischen Systems in einem bestimmten Bereich, ohne sich je zu wiederholen. Dieser Bereich ist graphisch darstellbar, indem die sich verändernden Einzelergebnisse der rekursiven Berechnungen gegeneinander aufgetragen werden. Die dabei entstehende Struktur (das Phasenbild bzw. der Attraktor im Phasenraum) weist bei vergrößerter Betrachtung Selbstähnlichkeit auf – es handelt sich um ein Fraktal. Die *Brown*sche Molekularbewegung eines Teilchens in der Ebene kann hierfür als anschauliches Beispiel dienen. Dessen irreguläre Bewegung führt zu einem verwobenen Muster ineinander verworrener, zurückgelegter Strecken. Betrachtet man dieses Muster bei verschiedenen Vergrößerungen und zählt letztlich immer nur die Anzahl

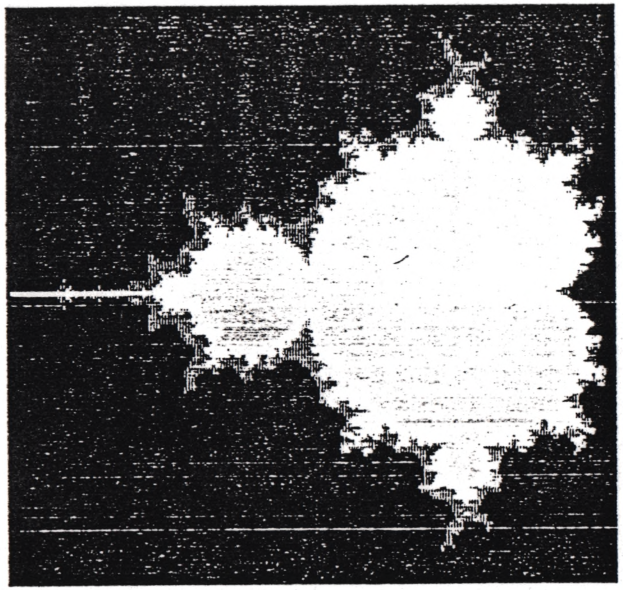


Abb. 1

a der gleichgroßen Ereignisse (hier die Anzahl der mit gleicher Länge zurückgelegten Strecken) in einem willkürlich gewählten Skalierungs-Ausschnitt mit dem Radius r, so erhält man eine Beziehung, welche einem Potenzgesetz folgt:

Gl. 21 $a = r^D$

mit D als Fraktaldimension. Nach Umformung und Logarithmieren der Gleichung läßt sich deren Wert näherungsweise bestimmen [18, 21]:

Gl. 22 $D = \frac{\ln a}{\ln r}$

In der Praxis stellt diese Berechnung die Bestimmung der Steigung einer Geraden im doppelt logarithmischen Netz mit gegeneinander aufgetragenen Meßwerten für a und r dar (lineare Regressionsanalyse, Listing 1). Die Kenntnis der Fraktaldimension zur Charakterisierung der geometrischen Eigenschaften eines seltsamen Attraktors ermöglicht dann die vergleichende Untersuchung komplexer Systeme auf dem Wege der Computersimulation: z. B. beim Flockengrößenwachstum (als Umkehrung der Rohrleitungsverkrustung) [22] oder beim Mischen hochviskoser Flüssigkeiten [23], wie sie bei der Polymerverarbeitung oder in der Geochemie bei Gesteinsschmelzen auftreten. Ferner wurde empirisch festgestellt, daß sich das Aussehen einer Oberfläche und damit deren Topographie anhand der Fraktaldimension gut beurteilen läßt – dies ist z. B. in der Oberflächenmeßtechnik anhand der Oberflächenmaßzahlen (nach DIN 4762 und DIN 4768) allein nicht möglich. Im Gegensatz zu der eingangs erwähnten deterministischen Mandelbrot-Menge handelt es sich bei diesen zuletzt genannten Beispielen um zufällige Fraktale – wobei Zufall als allgemeine Tendenz in Richtung auf Entropieerhöhung zu verstehen ist. Hierbei werden die Kontrollparameter der den jeweiligen Simulationen zugrundeliegenden rückgekoppelten Gleichungen durch (statistische) Verteilungsgesetze beeinflußt. Da dies ohne Auswirkungen auf die Selbstähnlichkeit bleibt, läßt sich argwöhnen, daß derartige Untersuchungen am Computer realitätsfern sind – die Laserforschung belegt jedoch die Korrektheit solcher Simulationen [24]. Die Computersimulationen zeigen dann, daß sich entwickelnde komplexe Systeme qualitativ unterschiedliche Zustände annehmen können. Im Chaos ist die Bildung von Inseln der Ordnung

```
10 CLS:CLEAR:KEY OFF
20 PRINT "ANZAHL WERTEPAARE "::INPUT N1
30 IF N1<=0 OR N1<>INT:ABS(N1)) THEN GOTO 10
40 DIM L(2.N1):B=0:X=0:Y=0:X1=0
50 Y1=0:X2=0:Y2=0:Z=0:R=0:A=0
60 FOR N=1 TO N1:FOR Q=1 TO 2
70 IF Q=2 THEN PRINT "EREIGNIS ":N:" (z.B.ERHOHUNG/VERTIEFUNG) ":
80 IF Q=1 THEN PRINT "UMFELD   ":N:" (z.B.KREISRADIUS/FLACHE ) ":
90 INPUT L(Q.N):IF L(Q.N)<1 THEN GOTO 90
100 L(Q.N)=LOG(L(Q.N))
110 NEXT Q:NEXT N
120 FOR N=1 TO N1
130 X=X+L(1.N):Y=Y+L(2.N):X2=X2+L(1.N)*L(1.N):Y2=Y2+L(2.N)*L(2.N)
140 Z=Z+L(1.N)*L(2.N)
150 NEXT N
160 X1=X/N1:Y1=Y/N1
170 IF N1*X2-X*X=0 THEN GOTO 190
180 B=(N1*Z-X*Y)/(N1*X2-X*X)
190 A=Y1-B*X1
200 X0=SQR(N1*X2-X*X):Y0=SQR(N1*Y2-Y*Y)
210 IF X0*Y0=0 THEN R=0:GOTO 230
220 R=(N1*Z-X*Y)/(X0*Y0)
230 PRINT "FRAKTALDIMENSION D= ":B
240 PRINT "KORRELATIONSKOEFFIZIENT r= ":R
250 PRINT :PRINT "NOCHMAL ?  (J/N)      "::INPUT AS
260 IF AS="J" OR AS="JA" THEN GOTO 10
270 END
```

Listing 1

durchaus normal. Unter welchen Bedingungen dies erfolgt, ist Gegenstand der Untersuchungen der Synergetik.

Synergetik

Die Synergetik als Lehre vom Zusammenwirken untersucht sich entwickelnde, gegenseitig beeinflussende komplexe Systeme [25], wobei ein System als Kontrollparameter (Ordner) des anderen auftritt. Die Systementwicklung ist dabei durch die Rekursion einer als Instruktionssatz zur Durchführung eines Prozesses aufzufassenden rückgekoppelten Funktion vorgegeben und basiert auf positiver Rückkoppelung. Wie bereits früher dargestellt wurde, kommt es dann über kurz oder lang zur Ausbildung eines instabilen Gleichgewichts (Bifurkationspunkt), an dem das System – ähnlich einem Phasenübergang – einen qualitativ neuen Zustand einnehmen kann. Dies ist mit der Zufuhr von Aktivierungsenergie zum Starten einer exothermen Reaktion vergleichbar und soll am Beispiel der Bildung von FeS aus den Elementen beschrieben werden. Bekannterweise erfolgt die exotherme Reaktion nach

Gl. 23 $Fe + S \rightarrow FeS$,

wobei eine ausreichende Aktivierungsenergie Voraussetzung ist. Betrachtet man eine Pulvermischung dieser Elemente als System, so kann dieses 3 Arten von Stabilität aufweisen:

1. Stabilität im Sinne von Beharrlichkeit bzw. Statik, wobei beide Elemente über einen langen Zeitraum hinweg unverändert vorliegen (Pulvermischung).
2. Stabilität im Sinne von Robustheit, wobei diese angibt, wie ein System mit dem Zuwachs bzw. Verlust von System-Komponenten (beispielsweise der Änderung des Energiehaushalts) zurechtkommt. Wird die Aktivierungsenergie unterschritten, so setzt keine Reaktion ein, und das System erweist sich als robust.
3. Stabilität im Sinne von Elastizität als Maß dafür, wie schnell ein System nach einer Störung (z. B. durch Energiezufuhr) in einen energetisch günstigen Zustand (dies kann die Ausgangslage oder aber ein qualitativ neuer Zustand sein) übergeht. In Gl. 23 entspricht dies der FeS-Bildung.

Bei ausreichender Zufuhr von Aktivierungsenergie erfolgt daraufhin exotherme Reaktion. Wird die Energiezufuhr jedoch etwas knapper bemessen, so wird ein höchst instabiles Gleichgewicht erreicht. An diesem Verzweigungs- bzw. Bifurkationspunkt kommt es zu Schwankungen – sogenannten kritischen Fluktuationen –, welche entsprechend der Fremdbeeinflussung durch die zugeführte Energie einer statistischen Verteilung unterliegen. Aufgrund von positiver Rückkoppelung verstärken sich dann an sich vernachlässigbar kleine, zufällige Störungen selbst. Dieses Verhalten findet man u. a. auch bei der Untersuchung des

Quellungsgrades von Gelen [26] (welcher bei der Entsorgung amphotermetallhaltiger Abwässer die Schlammalterung beeinflußt) oder im Bereich der Ökologie [27]. Die beschriebenen Störungen durch Fremdbeeinflussung (z. B. durch ein anderes System) führen bei dem rekursiven skaleninvarianten Prozeß zum Symmetriebruch und somit zur Bildung eines zufälligen Fraktals, das mittels linearer fraktaler Geometrie untersucht werden kann. Ferner bewirkt die Weiterentwicklung des Symmetriebruchs durch positive Rückkoppelung u. U. eine dramatische Systemänderung, die zu einer neuen Ordnung führt. Diese wiederum beeinflußt ihrerseits Fremdsysteme und deren Entwicklung. Ähnliche Prinzipien versucht man heute bei der Expertensystem-Programmierung mit Ansätzen wie Certainty-Factors, Fuzzy-Logic und Plausibilitätsintervallen [28] zu nutzen. Das *Lorenz*sche Wasserrad [29] mag als Gedankenexperiment zur Veranschaulichung dienen:

System A sei ein Wasserrad. System B sei ein Wasserstrom, der nicht wie üblich tangential, sondern mittig von oben auf das Wasserrad trifft, und System C sei eine Mechanik, die durch die Drehung des Wasserrades angetrieben wird und in ihrer Wirkung drehrichtungsabhängig ist. Erfolgt nun ein langsamer Wasserzufluß, so wird die Reibung des Rades nicht überwunden, und es kommt nicht zur Drehung. Bei verstärktem Zufluß erfolgt dann stetige Rotation, wobei die Drehrichtung und damit die Wirkung von System C nicht vorherbestimmbar, jedoch konstant ist. Bei sehr starkem Wasserzufluß hingegen entwickelt sich die Drehung chaotisch, da die Füllmenge der Schaufeln des Rades von der aktuellen Drehgeschwindigkeit abhängt – womit sich die Wirkung von System C mehrfach umkehrt. Beeinflußt System C nun seinerseits partiell durch Öffnen oder Schließen eines Schiebers den Wasserstrom des Systems B, so werden alle drei Systeme ständig zwischen den Zuständen konstante Rotation, indeterministisches Verhalten und instabilem Stillstandsgleichgewicht hin und hergeschleudert. Bildet man dieses Verhalten im Phasenraum ab, so erhält man einen seltsamen Attraktor mit fraktaler Struktur, bei dem an Bifurkationspunkten über die neue Ordnung entschieden wird. Normalerweise vernachlässigbar kleine Einflüsse führen dann die Entscheidung herbei und bewirken neue Verhaltensweisen aller drei Systeme.

Berücksichtigt man nun energetisch günstigere und daher stabilere Zustände, so ist die Zusammenlagerung von Subsystemen – die durch fraktale Strukturen darstellbar sind – nach dem Prinzip des kleinsten Widerstandes ebenfalls möglich (übrigens drängt sich bei diesem Prinzip die Ähnlichkeit mit der sehr rechenintensiven Problematik des kürzesten Netzwerkes von Ver- und Entsorgungsleitungen auf [30]). Die Zusammenlagerung kann zur Selbstorganisation führen, wobei Subsysteme ihre alten Eigenschaften zugunsten eines neuen Systemverhaltens verlieren – die Aussage der Gestaltpsychologie „Das Ganze ist mehr als die Summe seiner Teile" wird hierdurch bestätigt. Dies läßt sich mittels eines Beispiels aus der Elektrotechnik veranschaulichen (Abb. 2):

Im Subsystem R-C-Glied kommt es nach Schließen des Stromkreises bedingt durch die Aufladung des Kondensators zur Spannungszunahme. Im Subsystem R-L-Glied tritt durch den Widerstand der Spule aufgrund von elektromagnetischer Induktion das Gegenteil ein. Bei deterministischer Betrachtungsweise ließe sich durch die Koppelung dieser Subsysteme ein konstanter Spannungsverlauf erwarten – dies erfolgt jedoch nicht, da sich ein Schwingkreis gebildet hat. Ähnliche Zusammenlagerungen von sich ergänzenden Eigenschaften führen in Biologie und Biochemie zur sogenannten kumulativen Selektion [31]. Von Interesse sind in diesem Zusammenhang beispielsweise die Theorie zu den an kristallinen Tonmineralen katalytisch synthetisierten organischen Replikator-Molekülen des englischen Chemikers *Graham Cairns-Smith* [32] sowie der Hyperzyklus der autokatalytischen Replikation von Biomolekülen von *Manfred Eigen* [33]. Beide Untersuchungen führen in letzter Konsequenz zu den von der Biochemie aufgeklärten metabolischen Zyklen wie u. a. dem Zitratzyklus. Derartige Zyklen sind zwar weit vom niedrigsten energetischen Gleichgewicht (wie es z. B. in einer kristallinen Struktur vorliegt) entfernt, erhöhen jedoch die Entropie entsprechend dem 2. Hauptsatz der Thermodynamik und stellen somit nur weitere Inseln der Ordnung auf der Linie der Gleichgewichtspunkte dar. Die Untersuchung solcher Zusammenhänge durch Computersimulation steht erst am Anfang: die entsprechende Forschungsrichtung befaßt sich mit dem deterministischen Chaos.

Fortsetzung folgt

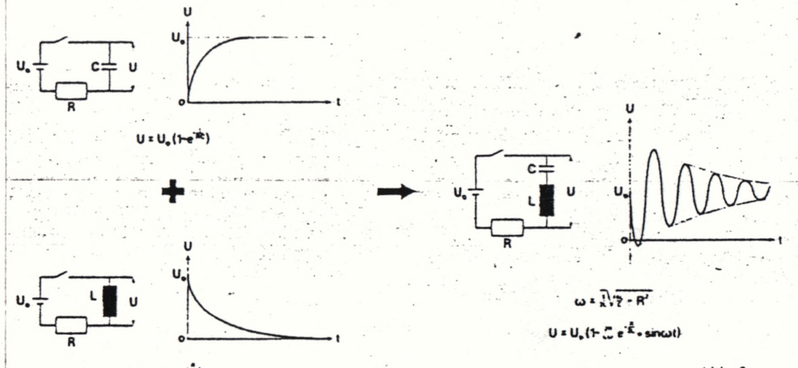


Abb. 2

Fraktale – Vorkommen und Anwendung, Simulation auf dem Computer

Dipl.-Chem. Heiko Schinke, Doz. Dr. sc. Martin Schleiff, Dipl.-Biochem. Mirco Wahab, Dr. Wolfgang Brandt, Prof. Dr. habil. Alfred Barth
Biotechnikum und Sektion Mathematik der Martin-Luther-Universität Halle-Wittenberg, O-4020 Halle an der Saale

Insbesondere in populärwissenschaftlichen Zeitschriften finden sich in den letzten Jahren immer wieder Veröffentlichungen zum Thema „Fraktale", auch im Zusammenhang mit der theoretischen Beschreibung von Ordnung und Chaos. Besonders eindrucksvoll sind dabei die vielfältigen Computergrafiken, welche die Schönheit der einerseits so bizarren, aber andererseits unendlich regelmäßigen fraktalen geometrischen Gebilde anschaulich machen. Auch in unterschiedlichen Fachzeitschriften werden interessante Ergebnisse der Anwendung der Fraktale auf spezielle naturwissenschaftliche Probleme beschrieben. Man kann annehmen, daß viele Erscheinungen der uns umgebenden Natur fraktalen Charakter haben und sich mit Hilfe der fraktalen Geometrie adäquat simulieren lassen.

In diesem Beitrag soll zunächst eine kurze, allgemeinverständliche Einführung in die Fraktaltheorie gegeben werden. Danach werden Beispiele aus verschiedenen Gebieten der Naturwissenschaften betrachtet, die die vielfältigen Anwendungsmöglichkeiten des Fraktalbegriffs auf konkrete Erscheinungen nachweisen. Dabei wird sowohl in der Auswahl und im Umfang als auch in der angeführten Literatur kein Anspruch auf Vollständigkeit erhoben. Der Schwerpunkt der Beispiele liegt, dem Charakter dieser Zeitschrift entsprechend, auf den Gebieten der Chemie und Biochemie.

Aber auch auf die Erzeugung von Fraktalen mit Hilfe des Computers und ihre Darstellung als Computergrafik wird kurz eingegangen. Es ergeben sich so bei geeigneter Farbgebung Bilder von hoher ästhetischer Schönheit.

Was sind Fraktale?

Benoit Mandelbrot, der Begründer der modernen Fraktaltheorie, hat den Begriff „Fraktal" zunächst nicht exakt (im formal-mathematischen Sinn) definiert. Um die typischen und wesentlichen Eigenschaften herauszuarbeiten, betrachtet er ein anschauliches Beispiel: Er stellt die Frage nach der Länge der Küstenlinie von Großbritannien. In unterschiedlichen Nachschlagewerken findet man jeweils andere Werte. Mandelbrot schreibt hierzu: „Die Länge einer typischen Küstenlinie... ist schlecht bestimmt. Wenn man verschiedene Küstenlinien bezüglich ihrer Ausdehnung vergleichen will, dann ist die Länge ein ungeeigneter Begriff" [1].

Zur Verdeutlichung dieser Aussage sei ein Landvermesser betrachtet, der die Küste vollständig vermessen soll. Nach jeweils einer gleichen Länge l_1 (geradlinige Entfernung, Luftlinie) setzt er einen Markierungspunkt. Die Länge $L_1(l_1)$ der Küste ist dann die Anzahl dieser Markierungspunkte multipliziert mit dem Abstand l_1 zwischen zwei solchen Punkten. Nun wird dieser Meßprozeß wiederholt mit einem kleineren Abstand l_2 zwischen jeweils zwei Meßpunkten. Man hat dann eine größere Anzahl von Meßpunkten und erhält $L_2(l_2)$, einen Wert, der größer als $L_1(l_1)$ sein wird. Eine weitere Verkleinerung auf l_3 führt zu einem dritten Wert $L_3(l_3)$, der noch größer ist.

Während bei genügend glatten geometrischen Gebilden, z. B. einem Kreis, die so ermittelten Längen L_n schnell einem Grenzwert zustreben (in der Mathematik als Bogenlänge bezeichnet), scheinen bei natürlichen Grenzlinien, wie der Küstenlinie, die Werte L_n zunächst immer weiter zu wachsen.

Wodurch kommt dieser Effekt zustande, der die Küstenlinie deutlich von den glatten Kurven unterscheidet? Bei jeder Vergrößerung des Maßstabes werden neue Feinheiten der Küste zugänglich: Buchten teilen sich in Unterbuchten, Halbinseln lassen neue Unter-Halbinseln erkennen. Leicht nachvollziehbar ist diese Erscheinung auch auf Landkarten, wenn die gleichen Gebiete auf Karten mit unterschiedlichem Maßstab betrachtet werden. Das Typische für das andersartige Verhalten der Küstenlinie und anderer natürlicher Grenzen ist also die **Selbstähnlichkeit**. Halbinseln und Buchten sind auf Karten mit unterschiedlichem Maßstab ähnlich. In Abstraktion von dem konkreten Beispiel der Küstenlinie kann man also definieren: Eine Punktmenge heißt fraktal, wenn sie selbstähnlich ist, d. h. jeder noch so kleine Teil der Menge hat bei passender Vergrößerung das gleiche Aussehen wie die ursprüngliche Menge.

Mit dieser Definition wird das Typische und Gemeinsame solcher geometrischen Gebilde wie einer Küstenlinie herausgearbeitet. Gleichzeitig wird die typische Eigenschaft, die Selbstähnlichkeit, in der Weise umfassend verallgemeinert, daß sie bei beliebigen Ausschnittsvergrößerungen gültig sein soll. Auf diese Weise wird unter Abstraktion von dem konkreten Erscheinungsbild eine fraktale Menge der formal-mathematischen Untersuchung zugänglich. Man kann nun auf rein mathematischem Weg allgemeine Schlußfolgerungen herleiten, die dann automatisch für alle konkreten fraktalen Gebilde gültig sind. Das ist der Vorteil bei einer Anwendung mathematischer Methoden zur Untersuchung konkreter Gebilde. Man kann die universell gültigen mathematischen Ergebnisse nutzen, um unmittelbar zu Aussagen über konkrete Gebilde zu gelangen.

Allerdings muß dabei eine wesentliche Einschränkung gemacht werden. Konkrete Gebilde besitzen die fraktalen Eigenschaften nur in einem bestimmten Größenordnungsbereich. Bei der Anwendung von Fraktalen in der Chemie hat es beispielsweise keinen Sinn, eine Verkleinerung bis zu atomaren Größenordnungen vorzunehmen. Die aus der allgemeinen Theorie hergeleiteten Aussagen haben daher bei der Anwendung auf konkrete Gebilde nur bei einer bestimmten Größenordnung einen Sinn.

Mathematische Eigenschaften der Fraktale

Die hier gegebene mathematische Definition der fraktalen Menge geht von einer fertigen Menge aus. Es wird dabei nicht angegeben, wie eine solche Menge tatsächlich konstruiert werden kann (beispielsweise auf dem Bildschirm eines Computers). Im folgenden soll ein solcher konstruktiver Prozeß angegeben werden. Man geht von einer beliebigen Menge aus. Dann wird angegeben, wie verkleinerte Abbilder der ursprünglichen Menge (meistens drei oder vier Bildmengen) in bezug auf die Ausgangsmenge zu positionieren sind (Mehrfach-Verkleinerungs-Kopier-Regel). Dieser Prozeß wird nun auf alle Bildmengen angewendet und dann das Ganze immer wiederholt (iterativer, rekursiver Prozeß). Da dieser Prozeß auch auf jede Teilmenge beliebig oft angewendet wird, ist es klar, daß man dadurch eine selbstähnliche Menge, also eine fraktale Menge, erhält. Ein wichtiger mathematischer Satz sagt nun, daß die entstehende fraktale Menge unabhängig von der Ausgangsmenge ist, mit der der Prozeß begonnen wurde. Die fraktale Menge ist also allein durch die wenigen Abbildungsvorschriften der Mehrfach-Verkleinerungs-Kopier-Regel vollständig bestimmt. An den Bildern wird aber deutlich, welche bizarre Vielfalt von Punktmengen man so erhalten kann.

In dem einleitenden Beispiel von der Küstenlinie war die Frage nach der Länge dieser Linie gestellt. Es war ersichtlich, daß die in der üblichen Weise definierte Länge einer Kurve bei einer fraktalen Menge nicht zu sinnvollen Werten führt. Man führt daher als Maßzahl für die Größe einer fraktalen Menge die sogenannte **fraktale Dimension** ein, die auf eine Arbeit des Mathematikers *Hausdorff* (1868 – 1942) aus dem Jahre 1919 zurückgeht. Diese **Hausdorffdimension** ist folgendermaßen erklärt: Man überdeckt die Menge mit einem Quadrat (bzw. im Raum mit einem Würfel). Nun wird jede Seite des Quadrats in n gleiche Teile geteilt, so daß man ein Gitter von kleinen Quadraten erhält. Mit a_n wird die Anzahl derjenigen kleinen Quadrate bezeichnet, die einen Punkt der fraktalen Menge enthalten. Dann ist der Grenzwert

$$D_H = \lim_{n \to \infty} \frac{\ln (a_n)}{\ln (n)}$$

der im Fall einer fraktalen Menge stets existiert, die Hausdorffdimension. Man kann diese Formel auch noch anders deuten. Wenn man die Anzahl der überdeckenden kleinen Quadrate für verschiedene Unterteilungen n ermittelt und die Werte n und a_n als Punkte in einem doppeltlogarithmischen Koordinatenpapier einzeichnet, so liegen diese Punkte annähernd auf einer Geraden, deren Anstieg die Hausdorffdimension ist. Der Vorteil dieses Verfahrens ist, daß man damit auch die fraktale Dimension realer Objekte wie beispielsweise einer Küstenlinie bestimmen kann. Die Hausdorffdimension kann entsprechend ihrer Definition eine gebrochene Zahl sein. Aus der Tatsache der **gebrochenen Dimension** ist die Bezeichnung Fraktal entstanden (vgl. z. B. Fraktur, Fraktion). Für die Kochsche Schneeflockenkurve in Abb. 2 ist $D_H = 1{,}262$.

Es sei hier darauf hingewiesen, daß es in der Theorie der Punktmengen noch weitere Dimensionsbegriffe gibt. So hat eine Punktmenge die topologische Dimension D_t, wenn man sie durch Punktmengen der Dimension D_t-1 zerteilen kann. Die Kochsche Schneeflockenkurve hat also die topologische Dimension 1, da man sie durch einzelne Punkte, die die Dimension Null haben, zerlegen kann. Für fraktale Mengen gilt $D_t < D_H$. Diese Ungleichung bringt die anschauliche Tatsache zum Ausdruck, daß eine fraktale Menge stets verzweigt und bizarr ist.

In der Mathematik begegnet man fraktalen Mengen bei der Untersuchung dynamischer Systeme. Der Einfachheit wegen werden nur dynamische Systeme betrachtet, die durch wenige Zustandsparameter (z. B. drei Parameter) beschrieben werden. Es interessiert nun, welche Werte die Parameter nach einer langen zeitlichen Entwicklung annehmen werden.

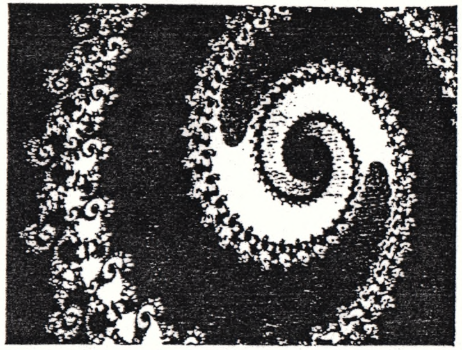


Abb. 1: Apfelmännchen in 256 Farben (Ausschnitt aus Abb. 8). Das Bild wurde mit dem Programm FRAKFILM (auf der Programm-Diskette) erzeugt.

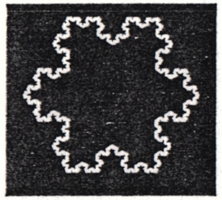


Abb. 2: Kochsche Schneeflocke (erzeugt mit dem Programm TURTLE)

Fraktale dürften für die meisten Leser neu sein. Deshalb bietet CLB denjenigen Lesern, die keinen Computer haben und sich die angebotene Diskette nicht anschaffen können, einen besonderen Leserservice an. Wir liefern 10 Fraktal-Motive (darunter auch zwei der hier farbig abgebildeten) in Postkartengröße für DM 25,–. Jedes der 10 Motive ist auch als Miniposter in der Größe DIN A4 lieferbar und kostet dann DM 15,– (bei Bezug von mehr als 1 Miniposter nur DM 12,50). Alle Bilder sind echte Fotos in Hochglanzausführung. Geben Sie Ihre Bestellung an die CLB-Redaktion auf: Postfach 1247, 5840 Schwerte. Es werden auch Euroschecks angenommen, im anderen Fall erhält der Besteller eine Rechnung. Verwenden Sie als Kennwort Fraktal-Postkarten und/oder Fraktal-Miniposter und geben Sie den von Ihnen ausgerechneten Rechnungsbetrag an. Die genannten Preise sind Endpreise und enthalten sowohl die Versandkosten als auch die Mehrwertsteuer. Die Belieferung erfolgt in der Reihe des Eingangs der Bestellungen.

Ein bekanntes Beispiel sind die idealen Planetenbahnen, wie sie sich aus den Keplerschen Gesetzen ergeben. Die Grenzmenge ist in diesem Fall eine gewöhnliche Kurve ($D_t = D_H = 1$). Bei komplizierteren dynamischen Systemen ist auch die Grenzmenge komplizierter, sie kann fraktale Eigenschaften aufweisen. Man spricht dann von chaotischem Verhalten. Es ist in diesem Fall nämlich nicht möglich, langfristig vorauszusagen, auf welchem Teil der fraktalen Grenzmenge die Parameterwerte nach einiger Zeit liegen werden.

Bei anderen dynamischen Systemen kann es mehrere verschiedene Grenzzustände geben. Man interessiert sich dann dafür, aus welchem Einzugsgebiet ein bestimmter Zustand erreicht wird. Die Grenze eines solchen Einzugsgebietes (z. B. Juliamenge, auch als Apfelmännchen bezeichnet) hat häufig fraktale Eigenschaften. Man sieht, daß die moderne Theorie dynamischer Systeme, der Wechsel von Ordnung und Chaos, vielfach mit dem Begriff der Fraktale zusammenhängt [1, 2, 13].

Fraktale in unserer Umwelt

Nachdem bisher die Fraktale mehr von ihrer theoretischen Seite betrachtet wurden, sollen nun Beispiele für das Vorkommen von fraktalen Eigenschaften in der Natur folgen. Dabei sei noch einmal darauf hingewiesen, daß der fraktale Charakter natürlicher Gebilde stets nur in einem bestimmten Größenordnungsbereich ausgeprägt ist.

Als einführendes Beispiel wurde bereits die fraktale Natur von Küstenlinien genannt. Allgemein haben natürliche Grenzen zwischen zwei Systemen (z. B. zwischen Land und Meer) häufig fraktalen Charakter. Auch Flüsse und die Einzugsgebiete von Flüssen weisen fraktale Eigenschaften auf, d. h. alle Flüsse und alle Flußeinzugsgebiete sind untereinander ähnlich. Die fraktale Dimension beträgt bei diesen Beispielen etwa 1,2.

In der Natur gibt es noch andere unregelmäßige Begrenzungslinien, beispielsweise die Oberfläche von Gebirgen. Aus dem Prozeß der Gebirgsentstehung wird der fraktale Charakter dieser Oberfläche deutlich: Ausgehend von einem Plateau (in der Computersimulation einer Dreiecks-Fläche, Abb. 3a) werden einzelne Punkte durch den Druck aus dem Erdinnern angehoben, andere durch die Erosion abgetragen (Abb. 3b). Dieses so entstandene, unregelmäßige Gebilde wird erneut teilweise angehoben und wieder abgetragen, d. h. den gleichen Vorgängen unterworfen (Abb. 3c – 3d). Im Laufe der Zeit entstehen so die unregelmäßigen Oberflächenformen der Erde. Dieser Entstehungsprozeß ist ein iterativer, rekursiver Prozeß, der hier zu einer Grenzfläche mit fraktalen Eigenschaften führt. Betrachtet man Wolken am Himmel, so stellt man fest, daß sie unregelmäßig begrenzt sind, aber andererseits selbstähnlich sind. Wolken stellen also fraktale Gebilde dar. Die fraktale Dimension ihrer Umfangslinie (der Grenzlinie zum wolkenfreien Gebiet) beträgt etwa 1,33.

Auch in der belebten Natur haben die Grenzflächen zwischen zwei Systemen häufig fraktalen Charakter. Allgemein bekannt ist, daß das Gehirn von Säugetieren gefurcht ist. Diese Furchungen sind sehr unregelmäßig, aber in gewisser Weise selbstähnlich. Die fraktale Dimension der Gehirnoberfläche konnte zu rund 2,75 ermittelt werden. Auch die Lungenoberfläche ist fraktal. Ermittelt man die

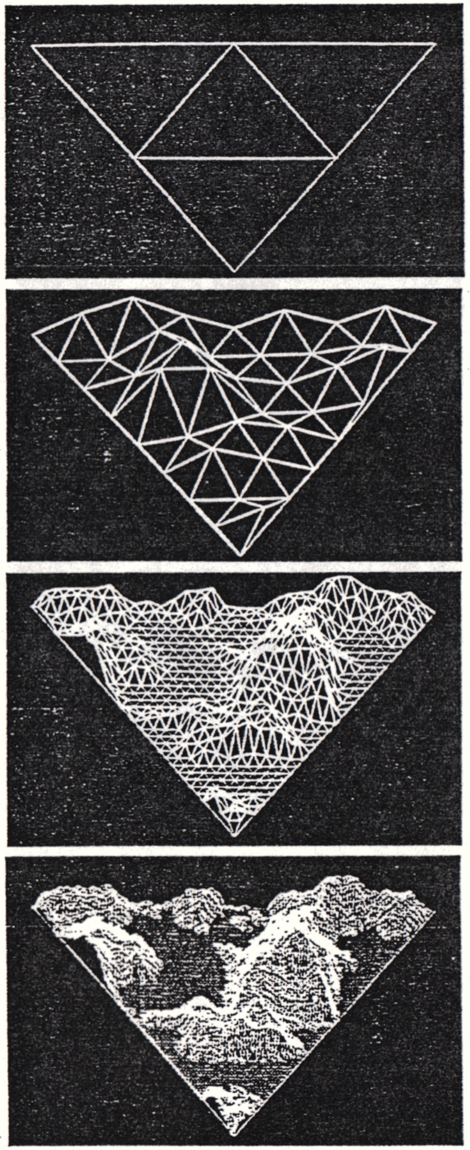


Abb. 3: Entstehung eines Gebirges auf dem Computer in 4 Abschnitten, dargestellt sind das 1., 3., 5. und 7. Bild (erzeugt mit dem Programm BERG)

Fläche der menschlichen Lungenalveolen mit Hilfe des Lichtmikroskops, so findet man 80 m², während man mit dem Elektronenmikroskop (und somit bei einer höheren

Auflösung) 140 m² findet. Diese Fläche wird also besser durch ihre fraktale Dimension charakterisiert, die bei $D_H = 2{,}17$ liegt.

Es sei hier noch einmal an die Bedeutung der fraktalen Dimension erinnert. Ein Wert von D_H zwischen 2 und 3 bedeutet, daß das zu beschreibende Gebilde zwischen einem ebenen Gebilde und einem räumlichen Gebilde einzuordnen ist. Je näher der Wert von D_H bei der Zahl 3 liegt, um so stärker wird durch das geometrisch zweidimensionale Gebilde der Raum erfüllt.

Pflanzen bilden mit ihrem Wachstum ebenfalls fraktale Gebilde aus. So wird in der Struktur eines Baumes, bestehend aus Stamm, Ästen, Zweigen und Blättern, der selbstähnliche Charakter deutlich (Abb. 4a). Auch Blätter für sich haben häufig eine fraktale Struktur. Bäume und Blätter (insbesondere Farnblätter, Abb. 4b) werden daher gern als Demonstrationsbeispiele genutzt, um zu zeigen, wie gut die vom Computer erzeugten fraktalen Gebilde natürliche Gegenstände simulieren können. Mit etwas Zufall lassen sich noch zahlreiche weitere Formen erzeugen (Abb. 4c). Auch die Abb. 5a und 5b wurden mit einem Computer erzeugt und stellen einen simulierten Schnitt durch einen Pflanzenstengel dar.

Fraktale in der Chemie

Viele chemische Prozesse laufen an Oberflächen ab (z. B. heterogene Katalyse). Es ist daher zu erwarten, daß hierbei auch Oberflächen mit fraktalen Eigenschaften eine Rolle spielen werden. In Abb. 6 ist ein Schnitt durch den sogenannten Mengerschwamm (in zwei Auflösungen) dargestellt, einem fraktalen Gebilde mit der Dimension 2,73, das als idealisiertes Modell für einen Katalysator mit durchgehenden Poren dienen kann. Wie bei einem realen Katalysator sind die Poren unregelmäßig verteilt und streuen in ihrer Größe über einen weiten Bereich. Die Oberfläche des Katalysators kann man daher, genauso wie bei der bereits beschriebenen Küstenlinie, nicht direkt bestimmen, da bei jeder Vergrößerung wieder neue Poren zugänglich werden. Zur Bestimmung der Oberfläche eines realen Katalysators wird diese mit einer Monoschicht von Adsorbatmolekülen (z. B. Stickstoff, Edelgase; idealisiert als Kugeln) bedeckt. Die Anzahl der bedeckenden Moleküle wird bestimmt und daraus die Oberfläche berechnet. In Abhängigkeit von den verwendeten Molekülen erhält man unterschiedliche Werte für die Oberfläche. Wenn man die Oberfläche mit verschieden großen Molekülen (Kugeln) mit den Radien r_1, r_2, ... bedeckt, die jeweilige Anzahl a_1, a_2, ... ermittelt, so kann man aus diesen Werten in der schon beschriebenen Weise die fraktale Dimension der Katalysatoroberfläche bestimmen (vgl. ausführlich in [19a – 19c]). Wie bereits oben erwähnt, hat diese empirisch bestimmte Dimensionszahl nur in dem Größenbereich einen Sinn, der durch die Radien der benutzten Moleküle eingegrenzt wird. Das ist stets zu beachten, wenn die fraktale Dimension als Katalysatorkenngröße verwendet wird.

Die in Abb. 2 dargestellte Kurve wird als Schneeflockenkurve bezeichnet. Sie kann erzeugt werden, indem, ausgehend von einem gleichseitigen Dreieck, aus jeder Kante jeweils ein neues Dreieck ausgestülpt wird. Die Bezeichnung als Schneeflocke ist kein Zufall, denn auch bei der Bildung von Schneeflocken werden die neuen H_2O-Mole-

Abb. 4: Fraktale Botanik auf dem Computer (erzeugt mit dem Programm STRAUCH): 4a) regelmäßiger Busch; 4b) fraktaler Farn; 4c) fraktaler Busch mit Zufallsparametern

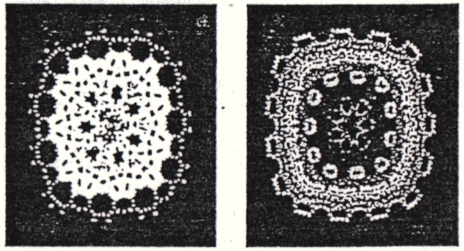


Abb. 5: Schnitt durch einen fraktalen Pflanzenstengel (erzeugt mit dem Programm PFLANZE). Geändert wurde nur der Parameter von b, in Bild a ist b = 0,01, in Bild b ist b = 0,005.

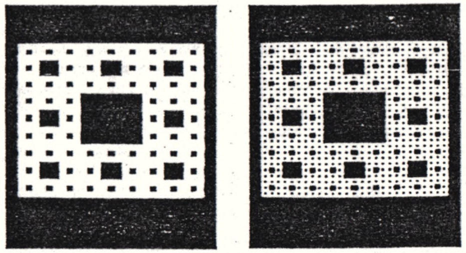


Abb. 6: Schnitt durch den Mengerschwamm in zwei verschiedenen Auflösungen

küle bevorzugt an geraden Kanten angelagert. Bei anderen Anlagerungsprozessen entstehen blumenartige oder baumartige Gebilde, beispielsweise bei mikrobiologischem Wachstum, bei elektrolytischen Kupferabscheidungen und bei Kolloidagglomerationen. Auch das sind fraktale Strukturen (siehe die Computersimulationen einer Anlagerung

die Berechnungsprogramme dafür wurden [9] entnommen).

Die Oberfläche von gefalteten Proteinen weist ebenfalls fraktalen Charakter auf. Die fraktale Dimension beträgt hier etwa 2,2. Auch die RNA-Sekundärstruktur wird in der Literatur [17] als fraktal beschrieben. Dabei wurde eine gute Übereinstimmung zwischen dem Entwicklungsniveau der RNA und ihrer fraktalen Dimension beobachtet.

Weitere Beispiele für Fraktale in der Chemie seien hier nur kurz aufgezählt: Vernetzte und verzweigte Polymere in der makromolekularen Chemie, Struktur von Zeolithen, Aggregationsprozesse, Diffusionsfronten. Gute Übersichten bieten das Buch von *Avnier* [12] sowie das PC-Lernprogramm von *J. Bargon* [13].

Fraktale auf dem Computer

Die Berechnung fraktaler Gebilde ist mit einem iterativen, rekursiven Verfahren möglich. Das ist eine verhältnismäßig einfache Rechenvorschrift, die aber sehr oft angewendet werden muß (siehe die Programme). Die Ausführung einer solchen Rechnung mit Zettel und Bleistift ist sehr eintönig und ermüdend und wird daher kaum durchgeführt.

Umgekehrt eignen sich solche iterativen Rechenvorschriften sehr gut für die Ausführung auf einem Computer. Man hat meist ein einfaches und kurzes Rechenprogramm, welches wiederholt, mitunter tausendfach, abgearbeitet werden muß. Mit der Entwicklung einer leistungsfähigen Rechentechnik wurde es möglich, fraktale Gebilde in großer Vielfalt zu berechnen und auf dem Bildschirm grafisch darzustellen. Heute gibt es für jeden Heimcomputer und Personalcomputer fertige Programme zur Berechnung fraktaler Gebilde. Da diese Programme einfach aufgebaut sind, kann man sie leicht selbst abwandeln und so nach Belieben völlig neue Bilder von meist bizarrer Schönheit erzeugen.

Einen Eindruck geben die Abb. 1 sowie 8, 9 und 10. Sie stellen ein fraktales Gebilde dar, das sich als Grenze des

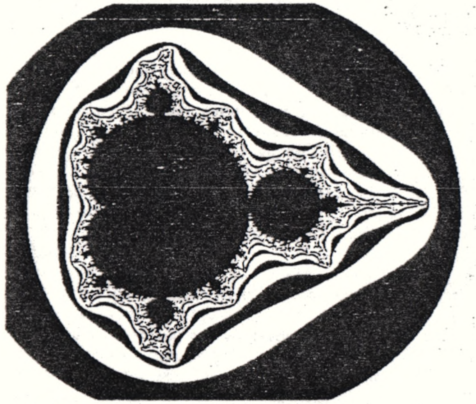


Abb. 8: Apfelmännchen (Gesamtansicht)

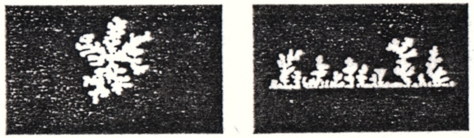


Abb. 7: Anlagerungsprozesse in der Chemie als Computersimulation: 7a) Anlagerung (Kristallisation); 7b) Sedimentation

Einzugsbereiches eines bestimmten dynamischen Prozesses ergibt (häufig als Apfelmännchen bezeichnet, siehe Listing 1). Abb. 8 stellt das vollständige Gebilde dar (in 2-Farbdarstellung). Dabei wurde der Bereich in der Mitte (das „Chaos", d. h. die Menge der Punkte, die nach einer bestimmten Anzahl von Iterationsschritten noch keine eindeutige Zugehörigkeit zu einem bestimmten Einzugsbereich erkennnen lassen) schwarz gezeichnet. Die weißen und schwarzen Ringe darum herum lassen die Anzahl der Iterationsschritte erkennen, die an den entsprechenden Punkten notwendig sind. Die Bezeichnung dieses Gebildes kommt von der Ähnlichkeit des inneren schwarzen Berei-

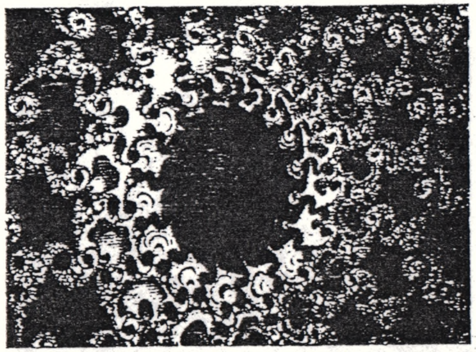


Abb. 9: Apfelmännchen (Ausschnitt aus Abb. 8; erzeugt mit dem Programm FRAKFILM)

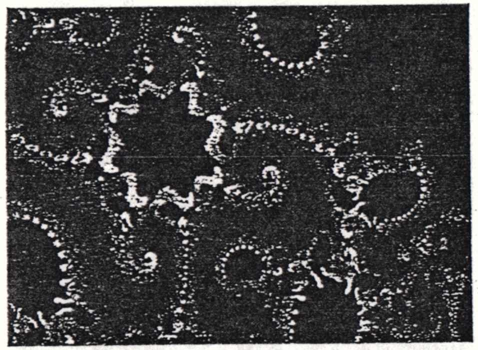


Abb. 10. Apfelmännchen (Ausschnitt aus Abb. 8; erzeugt mit dem Programm FRAKFILM)

len jeweils Ausschnitte daraus dar, wobei auch unterschiedliche Möglichkeiten der Farbgebung verwendet wurden. Die Berechnungsvorschrift ist auch hier iterativ und rekursiv, aber mathematisch von anderer Struktur als die Berechnungsvorschrift, die zur Schneeflockenkurve oder zu baumartigen bzw. blattartigen Fraktalen führt. Für interessierte Leser enthält das Literaturverzeichnis Hinweise auf Programme zur Berechnung fraktaler Strukturen auf dem Commodore C64 sowie auf dem Personalcomputer [3 – 11], außerdem ist zu diesem Artikel eine Programm-Diskette mit Fraktal-Programmen erhältlich (s. u.).

An den zwei Teilbildern der Abb. 5 soll noch einmal gezeigt werden, wie minimale Änderungen eines Ausgangsparameters zu grundlegenden Veränderungen in der Struktur des sich als Grenzfall ergebenden Bildes führen (siehe dazu Listing 3). Der Wert von b beträgt in Abb. 5a 0,01, in Abb. 5b 0,005. Nach 50 000 Iterationsschritten (Punkten auf dem Bildschirm) wurden diese beiden Bilder erhalten, zwischen denen sich kaum eine Ähnlichkeit finden läßt.

Es ist also offensichtlich, daß man durch eine andere Wahl gewisser Parameter im Berechnungsprogramm völlig neue Strukturen und Bilder erzeugen kann.

Der Vorteil der Bilderzeugung in der beschriebenen Weise besteht darin, daß man zur Erzeugung eines kompletten Bildes nur eine einfache und kurze Berechnungsvorschrift braucht. So werden Gebirgslandschaften in Computerspielen oft als Fraktale berechnet (z. B. bei dem Spiel „Rescue of Fractalus" für 8-bit Homecomputer). Das ist effektiver, als wenn das komplette Landschaftsbild gespeichert werden müßte. Auch Bäume und ganze Wälder werden auf diese Weise als Fraktale „synthetisch" erzeugt und liegen nicht als fertige Bilder vor (siehe Abb. 4c), wobei in begrenztem Umfang der Zufall bei der Berechnung mit eine Rolle spielt.

Diese vom Computer erzeugten Bilder sind häufig von einer faszinierenden Schönheit und stellen in der Beziehung durchaus Kunstwerke dar. Demgegenüber wird oft eingewandt, daß die Berechnungsvorschrift vollständig determiniert und reproduzierbar ist. Das ist richtig, aber die Wahl der Parameter, die zum fertigen Bild führt, und die passende Farbgebung erfordern die Intuition. Daher stellen diese „künstlichen" Fraktale durchaus kreative Leistungen dar. Der Computer ist dabei nur das Hilfsmittel, um die Ideen des Autors gegenständlich zu realisieren. Deutlich wird diese Aussage z. B. in den Abb. 1, 9 und 10 sowie bei dem Programm FRAKFILM auf der zu diesem Beitrag erhältlichen Programmdiskette (s. u.).

Einen umfassenden Eindruck von solchen „künstlichen" Bildern bietet [2]. Zahlreiche fertige Computerprogramme zum Thema Fraktale (geschrieben in der Programmiersprache C) sind in [3] veröffentlicht. Programmbausteine (geschrieben in PASCAL) zum Selberexperimentieren mit Fraktalen finden sich in [4].

Zu den Programmen

Zum eigenen Experimentieren mit Fraktalen am Computer werden hier drei kurze Programm-Listings vorgestellt. Sie sind alle in C geschrieben, wobei für die Grafik die speziellen Funktionen des Turbo C-Compilers der Firma Borland verwendet werden. Zur Compilierung auf anderen Compilern müssen die entsprechenden Funktionen

sten Programmzeilen).

Die Programme laufen ohne Änderung auf allen Farbgrafikkarten (CGA, EGA oder VGA). Ein Coprozessor ist – wegen der teilweise doch recht großen Rechenzeit – sinnvoll. Alle Zeichen zwischen „/*" und „*/" (teilweise auch über mehrere Zeilen reichend) sind Kommentare und brauchen nicht abgetippt zu werden. Aus Platzgründen wurde in allen Programmen auf Eingabemöglichkeiten über den Bildschirm verzichtet. Die Startwerte für die Berechnungen (in den Zeilen mit „define") können mit einem Editor verändert werden; danach sind die Programme erneut zu compilieren. Auf der Programm-Service-Diskette sind u. a. auch diese 3 Listings (in einer erweiterten Form mit interaktiver Parametereingabemöglichkeit) als Quelltexte sowie als lauffähige EXE-Files vorhanden.

Listing 1: APFEL.C

Mit diesem Programm können Apfelmännchen mit maximal 16 verschiedenen Farben berechnet werden. Die angegebenen Werte für rmin, rmax, imin und imax liefern das vollständige Bild. Ausschnitte sind durch die Wahl kleinerer Werte möglich. Gleichzeitig muß dann jedoch die Iterationstiefe (Variable „Tiefe") erhöht werden, was automatisch zu einer Erhöhung der Rechenzeit führt.

Zur Farbgebung werden die Anzahl der benötigten Iterationsschritte bis zum Erreichen des Ergebnisses gezählt, und dieser Zahl wird dann eine Farbe zugeordnet. Sollte kein Ergebnis erreicht werden, wird als Farbe „schwarz" gewählt.

Listing 2: DRACHE.C

Eng mit dem Apfelmännchen verwandt ist der San-Marcos-Drachen. Hiermit kann er als „Gebirge" dargestellt werden, d. h. die Anzahl der benötigten Iterationsschritte wird durch eine unterschiedliche Höhe der Punkte über der Grundfläche kenntlich gemacht. Speziell bei Verwendung einer CGA-Grafikkarte müssen die Farben verändert werden (COLOR1 = 1, COLOR2 = 2, COLOR3 = 3).

Listing 3: PFLANZE.C

Vollständig andere Bilder können mit diesem Programm erzeugt werden. Sie haben Ähnlichkeit mit Querschnitten durch Pflanzenstengel (siehe Abb. 5a – 5b). Für a, b und c kann jeder beliebige Wert eingesetzt werden, bei großen Werten sollten die Zoomfaktoren „x_zoom" und „y_zoom" verkleinert werden. Mit „bunt" kann die Farbigkeit des Bildes verändert werden, d. h., nach soviel Punkten, wie dort angegeben, ändert sich die Zeichenfarbe.

Aus technischen Gründen werden die drei Listings in der Beilage CLB-Memory abgedruckt. Wir bitten um Beachtung.

Erläuterungen zur Programm-Service-Diskette

Als Service für alle an Fraktalen auf dem Computer interessierten Leser ist zu diesem Artikel eine Programm-Service-Diskette erhältlich. Sie enthält neben den bereits hier abgedruckten Listings noch einige andere als C-Quelltexte, dazu von allen Programmen die lauffähigen EXE-Files (für MS-DOS), einige Beispielbilder zum sofortigen Anse-

hen, für jedes Programm eine Tabelle mit interessanten Parametern, Hinweise zur Compilierung der Quelltexte (sie sind Compiler-unabhängig) sowie eine ausführliche Beschreibung jedes der einzelnen Programme. Die Disketten (wahlweise 2 Stück 5 1/4 Zoll 360 KB oder 1 Stück 3 1/2 Zoll 720 KB, bei der Bestellung bitte mit angeben) sind für einen Zeitraum von sechs Monaten nach Erscheinen dieses Beitrags für DM 35,- (incl. Porto, Verpackung, Mehrwertsteuer) bei der CLB-Redaktion (Postfach 1247, 5840 Schwerte) erhältlich. Danach werden sie mit in das Vertriebsprogramm der CLB-Software aufgenommen und sind dann für DM 70,- (zuzüglich Porto, Verpackung, Mehrwertsteuer) erhältlich.

Auf den Disketten befinden sich im einzelnen die folgenden Programme (einschließlich Quelltexte):

APFEL.C

Das als Listing hier bereits abgedruckte Apfelmännchen-Programm, erweitert um die Möglichkeit des Bildspeicherns, der Ausschnittbestimmung unter Verwendung einer Maus sowie der Möglichkeit der interaktiven Parametereingabe.

DRACHE.C

Das hier als Listing abgedruckte Programm für den 3D-San-Marcos-Drachen, erweitert um die Möglichkeit der interaktiven Parametereingabe.

PFLANZE.C

Das als Listing hier abgedruckte Programm eines Schnittes durch einen Pflanzenstengel, ebenfalls erweitert um die Möglichkeit der interaktiven Parametereingabe. Es können hierbei gleichzeitig mehrere Muster auf dem Bildschirm dargestellt werden.

BERG.C

Programm zur Berechnung fraktaler Berge (siehe Abb. 3) mit interaktiver Parametereingabe. Das Gebirge kann durch Linien oder Polygone aufgebaut werden; die einzelnen Flächen können jeweils unterschiedlich farblich modelliert werden.

STRAUCH.C

Programm zur Berechnung fraktaler Bäume, Sträucher und Wälder (siehe Abb. 4) auf dem Bildschirm und Ausgabe auf einen HP-kompatiblen Plotter.

TURTLE.C

Programm zur Erzeugung von Fraktalen mit Turtle-Grafik (Schneeflocke, Hilbertkurve etc., siehe Abb. 2). Das Programm enthält einen Turtle-Editor, mit dessen Hilfe sich einfach selbstähnliche Strukturen generieren lassen. Die Ausgabe der Grafiken kann ebenfalls auf dem Bildschirm oder auf einem HP-kompatiblen Plotter erfolgen.

FRAKFILM.C

Dies ist das Spitzenprogramm der Programmsammlung. Mit diesem Programm können, ähnlich wie mit APFEL.C, Apfelmännchen berechnet werden, die Darstellung erfolgt aber mit bis zu 256 Farben gleichzeitig. Erforderlich ist dazu jedoch eine VGA-Grafik-Karte. Nach der Berechnung können die Farben in unterschiedlichen Variationen kontinuierlich geändert werden. So ergeben sich sehr schöne Filmeffekte, da die Farben durch die komplizierten Figuren ineinanderlaufen, und schöne Standbilder (siehe Abb. 1, 9 und 10).

Literaturübersicht (Auswahl)

[1] Mandelbrot, B.B.: Die fraktale Geometrie der Natur. Akademie-Verlag Berlin, 1987.

[2] Peitgen, H.O.: Richter, P.H.: The Beauty of Fractals. Springer-Verlag, Berlin 1986.

a) Fraktale und Computer

[3] Stevens, R.T.: Fractal-Programming in C (englisch). M & T Publishing Inc., Redwood City, 1989.

[4] Becker, K.-H.: Dörfler, M.: Dynamische Systeme und Fraktale. Vieweg-Verlag, Braunschweig/Wiesbaden 1989.

[5] Vilsmeier, S.: Reise in die fraktale Faszination. 64'er, 5/1987.

[6] Vilsmeier, S.: Vorstoß ins Chaos. 64'er, 9/1987, 10/1987, 12/1987, 2/1988, 3/1988, 4/1988.

[7] Chaos und Fraktale. Spektrum der Wissenschaft Verlagsgesellschaft, Heidelberg 1989.

[8] Computer-Kurzweil. Spektrum der Wissenschaft Verlagsgesellschaft, Heidelberg 1989.

[9] DOS extra, Sonderheft Grafik, Nr. 12, 1990. DMV-Verlag.

[10] Bandt, C.; Kuschel, T.: Fraktale mittels „Igelgeometrie". Wissen. Fortschr. 38 (1989) 264.

[11] Urian, R.: Wir bauen uns ein Monster. c't, 5/1988.

b) Fraktale und Wissenschaft

[12] Avnir: The Fractal Approach to Heterogeneous Chemistry. Surfaces, Colloids, Polymers. John Wiley & Sons, 1989.

[13] Bargon, J., u.a.: Fraktale in der Physikalischen Chemie. IBM-Anwendungsbrief Nr. 16, Bonn 1987.

[14] Pfeifer, P.: CHIMIA 39 (1985) 120.

[15] Amann, A.; Gans, W.: Angew. Chem. 101 (1989) 277-285.

[16] Purugganan, M.D.: Naturwissenschaften 76 (1989) 471.

[17] Luo, L.; Tsai, L.: Chin. Phys. Lett 5 (1988) 421.

[18] Verhandlungen der Gesellschaft Deutscher Naturforscher und Ärzte, 115. Versammlung, Tagungsband: Ordnung und Chaos in der unbelebten und belebten Natur. Wissenschaftliche Verlagsgesellschaft mbH. Stuttgart 1989.

[19a] Ackermann, W.-G.; Spindler, H.: Z. phys. Chemie 269 (1988) 1000.

[19b] Ackermann, W.-G.; Spindler, H.; Kraft, M.: Z. phys. Chemie 269 (1988) 1233.

[19c] Spindler, H.; Kärger, J.: Z. phys. Chemie 270 (1989) 225.

[20] Spindler, H.: Z. Chem. 28 (1988) 421.

[21] Heidenreich, E.: Chem. Techn. 38 (1986) 65.

[22] López-Quintela, M.A.; Tojo, C.; Buján-Núñez, M.C.: Molec. Phys. 65 (1988) 1195.

[23] Mögel, H.-J.: Wiss. Z. Univ. Halle XXXVII (1988), H. 5, 20.

[24] Leuenberger, H.; Holman, L.; Usteri, M.; Winzap, S.: Pharm. Acta Helv. 1989 (2) 34.

[25] Blumen, A.; Schnörer, H.: Angew. Chem. Int. Ed. Engl. 29 (1990) 113.

Listings zu dem Aufsatz über Fraktale in diesem CLB-Heft

```
Listing 1: APFEL.C
------------------

/*******************************************************
              Das Apfelmännchen in 'C'

 Für die graphische Darstellung werden folgende
 Routinen der Grafikbibliothek von TURBO-C (Version
 1.5, 2.0, TC++) verwendet:
 initgraph   :  Einschalten des Grafikmodus
 getmaxx     :  liefert höchste X-Koordinate
 getmaxy     :  liefert höchste Y-Koordinate
 getmaxcolor :  liefert höchste Farbnummer
 putpixel    :  zeichnet Punkt bei x,y in der Farbe
 closegraph  :  Zurück in den Textmodus

*******************************************************/

#include <graphics.h>     /* TURBO-C Grafikheader */
#include <conio.h>

/*******************************************************
  Definierte Konstanten für die Darstellung des
  Apfelmännchens. Diese Werte sind für das Aussehen
  der Figur verantwortlich.  Sinnvolle Werte liegen
  für rmax/rmin und imax/imin jeweils zwischen
  -2.5 und 2.5.
  Die Iterationstiefe (tiefe) sollte nicht unter 20
  liegen. Es gilt: je höher dieser Wert, desto
  detaillierter die Darstellung, aber auch um so
  länger die Rechenzeit.
*******************************************************/

#define tiefe  30 /* Maximalzahl der Iterationen */
#define rmin   -2.5  /* Wertebereich für den     */
#define rmax    1.0  /* Realteil sowie für den   */
#define imin   -1.5  /* Imaginärteil des be-     */
#define imax    1.5  /* trachteten Bereichs      */

/*******************************************************/

 int main()
{
 int     tc_driver, tc_mode;
 int     maxcolor, xres, yres, xr, yi, v, color;
 double ar, ai, r, x, y, i, xalt, xx, yy;

 tc_driver = DETECT;      /* Automatische Erkennung */
                          /* des Grafiksystems      */
 initgraph(&tc_driver,&tc_mode,"");
 xres = getmaxx() + 1;  /* maximale Auflösung X   */
 yres = getmaxy() + 1;  /* maximale Auflösung Y   */
 maxcolor = getmaxcolor() + 1;  /* max. Farbzahl  */
                             /* Skalierung von    */
 ar = (rmax - rmin) / xres;  /* Wertebereich zu   */
 ai = (imax - imin) / yres;  /* Bildschirm        */
 v = 0;                      /* Iterationszähler  */

 for(yi=0; yi<yres; yi++)       /* Schleife über Y */
   for(xr=0; xr<xres; xr++) {   /* Schleife über X */
      r = rmin + xr * ar;
      i = imin + yi * ai;
      x = y = xx = yy = 0.0;
      v = 0;

        do {                           /* Iterationen */
           xx = x * x;
           yy = y * y;
           xalt = x;
           x = xx - yy  + r;
           y = 2.0 * xalt * y + i;
        }
        while(!(++v >=tiefe || (xx + yy > tiefe)));
       /* Farbe für den nächsten Punkt bestimmen */
      color = (v == tiefe) ? BLACK : (v % maxcolor);
      putpixel(xr, yi, color);

      if(kbhit()) {   /* Nach betätigen von Tasten */
      getch();         /* erfolgt Schleifenabbruch */
        yi = yres, xr = xres;
      }
    }
                          /* Ende der X-Schleife  */
                          /* Ende der Y-Schleife  */
 getch();                 /* Erst nach Betätigen  */
 closegraph();            /* einer Taste wird das */
 return 0;                /* Programm beendet.    */
}
/*******************************************************/
```

Listing3: PFLANZE.C

```
/*****************************************************
            Das Programm 'Pflanze' in 'C'

 Für die graphische Darstellung werden folgende
 Routinen der Grafikbibliothek von TURBO-C (Version
 1.5, 2.0, TC++) verwendet:
 initgraph    :  Einschalten des Grafikmodus
 getmaxx      :  liefert höchste X-Koordinate
 getmaxy      :  liefert höchste Y-Koordinate
 getmaxcolor  :  liefert höchste Farbnummer
 putpixel     :  zeichnet Punkt bei x,y in der Farbe
 closegraph   :  Zurück in den Textmodus

 *****************************************************/

 #include <graphics.h>      /* TURBO-C Grafikheader */
 #include <conio.h>
 #include <math.h>

/*****************************************************
  Definierte  Konstanten  für  die  Darstellung der
  'Pflanze'. Diese Werte sind für das Aussehen
  der Figur verantwortlich.  Sinnvolle Werte liegen
  etwa zwischen -3.0 und 3.0 für a, b und c.
  Die Konstante 'bunt' dient dazu,  erst  nach  dem
  Zeichnen einer größeren Menge von Punkten einen
  Farbwechsel zuzulassen.
 *****************************************************/

 #define a        -1.0    /* Parameter für Form und */
 #define b         1.0       /* Farbe der 'Pflanze' */
 #define c        -2.0
 #define x_zoom  5.0  /* Zoomfaktoren für X und Y */
 #define y_zoom  5.0 /* für Größe der Darstellung */
 #define bunt  4000 /* Verzögerung f. Farbwechsel */
/*****************************************************/

                        /* Makro für Absolutbetrag */
 #define FABS(x)   (((x) > 0.0) ? (x) : (x) * -1.0)

/*****************************************************/

 int main()
{
 int  tc_driver, tc_mode, maxc;
 int  x_mitte, y_mitte, x_coord, y_coord, color, cl;
 double  x0, y0, x1;

 tc_driver = DETECT;      /* Automatische Erkennung */
                          /* des Grafiksystems      */
 initgraph(&tc_driver,&tc_mode,"");
 x_mitte = getmaxx() / 2;     /* Zentrierung von X */
 y_mitte = getmaxy() / 2;     /* Zentrierung von Y */
 maxc = getmaxcolor() + 1;       /* Anzahl Farben */
                                /* cl wird 'über'- */
 cl = 0;                     /* laufend hochgezählt */
 x0 = y0 = 0.0;                     /* Startwerte */

 while(!kbhit()) {         /* Bildschirmkoordinaten */
    x_coord = (int)(x0 * x_zoom) + x_mitte;
    y_coord = (int)(y0 * y_zoom) + y_mitte;

    color = ((++cl / bunt)  % maxc) + 1;
    putpixel(x_coord, y_coord, color);
                            /* neue Funktionswerte */
    x1 = y0 - sqrt(FABS(b * x0 - c))
          * ((x0 < 0.0) ? -1.0 : 1.0);
    y0 = a  - x0;  x0 = x1;
 };
             /* Da eine Taste gedrückt wurde, wird */
 getch();                /* das Zeichen eingelesen, */
 closegraph(); /* der Textmodus wiederhergestellt */
 return 0;           /* und das Programm beendet. */
}
/*****************************************************/
```

Listing 2: DRACHE.C

```
/*****************************************************
       Der San Marcos-Drachen als 3D-Bild in 'C'

 Für die graphische Darstellung werden folgende
 Routinen der Grafikbibliothek von TURBO-C (Version
 1.5, 2.0, TC++) verwendet:
 initgraph    :  Einschalten des Grafikmodus
 getmaxx      :  liefert höchste X-Koordinate
 getmaxy      :  liefert höchste Y-Koordinate
 getmaxcolor  :  liefert höchste Farbnummer
 line         :  zeichnet Linie von x1,y1 nach x2,y2
 closegraph   :  Zurück in den Textmodus

 *****************************************************/

 #include <graphics.h>     /* TURBO-C Grafikheader */
 #include <conio.h>

/*****************************************************
  Definierte  Konstanten  für  die  Darstellung des
  Drachens. Diese Werte sind für das  Aussehen  der
  Figur verantwortlich.  Sinnvolle Werte liegen für
  rmax/rmin und imax/imin zwischen -2.5 und 2.5.
  Die Iterationstiefe (tiefe) sollte nicht unter 20
  liegen, es gilt: je höher dieser Wert,  desto de-
  tailierter die Darstellung, aber auch um so länger
  die Rechenzeit.
 *****************************************************/

 #define tiefe  50     /* Max.zahl der Iterationen */
 #define hoehe 100     /* Höhe über Grundfläche    */
 #define rmin   -2.5   /* Wertebereich für den     */
 #define rmax    2.5   /* Realteil sowie für den   */
 #define imin   -1.3   /* Imaginärteil des be-     */
 #define imax    1.3   /* trachteten Bereichs      */

 #define COLOR1  6          /* Farbe Vorderseite   */
 #define COLOR2  7          /* Farbe rechte Seite  */
 #define COLOR3  14         /* Farbe Oberfläche    */

/*****************************************************/

 int main()
{
 int    tc_driver, tc_mode;
 int    m, n, k, xm, ym;
 int    u, v, ul, vl;
 double a, x, x2, y, yl, y2, xc, yc, dx, dy;

 tc_driver = DETECT;     /* Automatische Erkennung */
                         /* des Grafiksystems.     */
 initgraph(&tc_driver,&tc_mode,"");
 xm = getmaxx() / 2;     /* maximale Auflösung X   */
 ym = getmaxy() / 2;     /* maximale Auflösung Y   */

 dx = (rmax - rmin)/(double)xm; /* Skalierung von */
 dy = (imax - imin)/(double)ym; /* Wertebereich   */
                                /* zu Bildschirm  */
 xc = 1.0;
 yc = 0.0;
 a  = (double)tiefe / (double)hoehe;

/* Verhältnis Rechentiefe zu Höhe der Darstellung */

 for(n=0; n<ym; n++) {          /* Schleife über Y */
       yl = imin + n * dy;
    for(m=0; m<xm; m++) {       /* Schleife über X */
          x = rmin + m * dx;
      y = yl;
      k = 0;
          while(++k <= tiefe) {   /* Iterationen */
                x2 = x * x;
                y2 = y * y;
                y  = 2 * x * y - yc;
                x  = x2 - y2 - xc;
                if(x2 + y2 >=(double)hoehe) break;
   /* Abbruch, wenn Höhe der Darstellung erreicht */
          }
        /* Berechnung der Bildschirmkoordinaten */
          u  = m + xm/2 - n/2 ;
          ul = u + 1;
          v  = n + ym;
          vl = v - (int)((double)k / a - 1.0);

      setcolor(COLOR1);     /* Farbe vorne setzen */
      line(u,v, u,vl);          /* Linie zeichnen */
      setcolor(COLOR2);    /* Farbe rechts setzen */
      line(ul,v, ul,vl);        /* Linie zeichnen */
      setcolor(COLOR3);      /* Farbe oben setzen */
      line(u,vl, ul,vl);        /* Linie zeichnen */
      if(kbhit()) n = ym, m = xm; /* Nach Betätigen
                       einer Taste Schleifenabbruch */
    }
 }
 getch();                  /* Erst nach Betätigen  */
 closegraph();             /* einer Taste wird das */
 return 0;                 /* Programm beendet.    */
}
/*****************************************************/
```


Schluß

Hallo Club 80'er,

endlich, werdet Ihr sagen, gibt es mal wieder eine INFO! Ein langer Zeitraum war jetzt zwischen INFO 36/37 - aber das Material aus vorliegendem INFO ist alles, was zwischenzeitlich bei mir eingetrudelt ist. Das lange Warten auf versprochene Beiträge hat sich nicht gelohnt -sie fehlen bis heute. Schade!

Vielleicht ergibt sich über die BASIC-Ecke mehr Feedback. Interesse dazu wurde bisher bekundet. Auf alle Fälle möchte ich Mitte Dezember mit der Produktion des nächsten INFO's beginnen. Ich hoffe, daß bis dahin noch einige Artikel bei mir ankommen.

Gleichzeitig wurde anläßlich Clubtreffen Süd beschlossen, daß zukünftig Artikel von mindestens eineinhalb DIN A 4 Seiten, die im INFO abgedruckt, mit 5,-DM vergütet werden. Dies in der christlichen Hoffnung auf eine sich bessernde Beteiligung am Club-INFO zum Gemeinwohl aller. Entlohnt werden maximal bis zu vier Beiträge pro Jahr. Der Betrag wird den jeweiligen Autoren zum Jahresende ihrem Jahresbeitragskonto gutgeschrieben. Wir hoffen auf diese Weise auch die Portokasse der Autoren etwas zu schonen. Begonnen wird die Aktion mit den Artikeln zum 38. (nächsten) INFO.

Ich wünsche Euch eine schöne Zeit
bis zum nächsten mal

Jens

Adressen-Liste Club80

Stand vom 30-10-1992
Seite 1

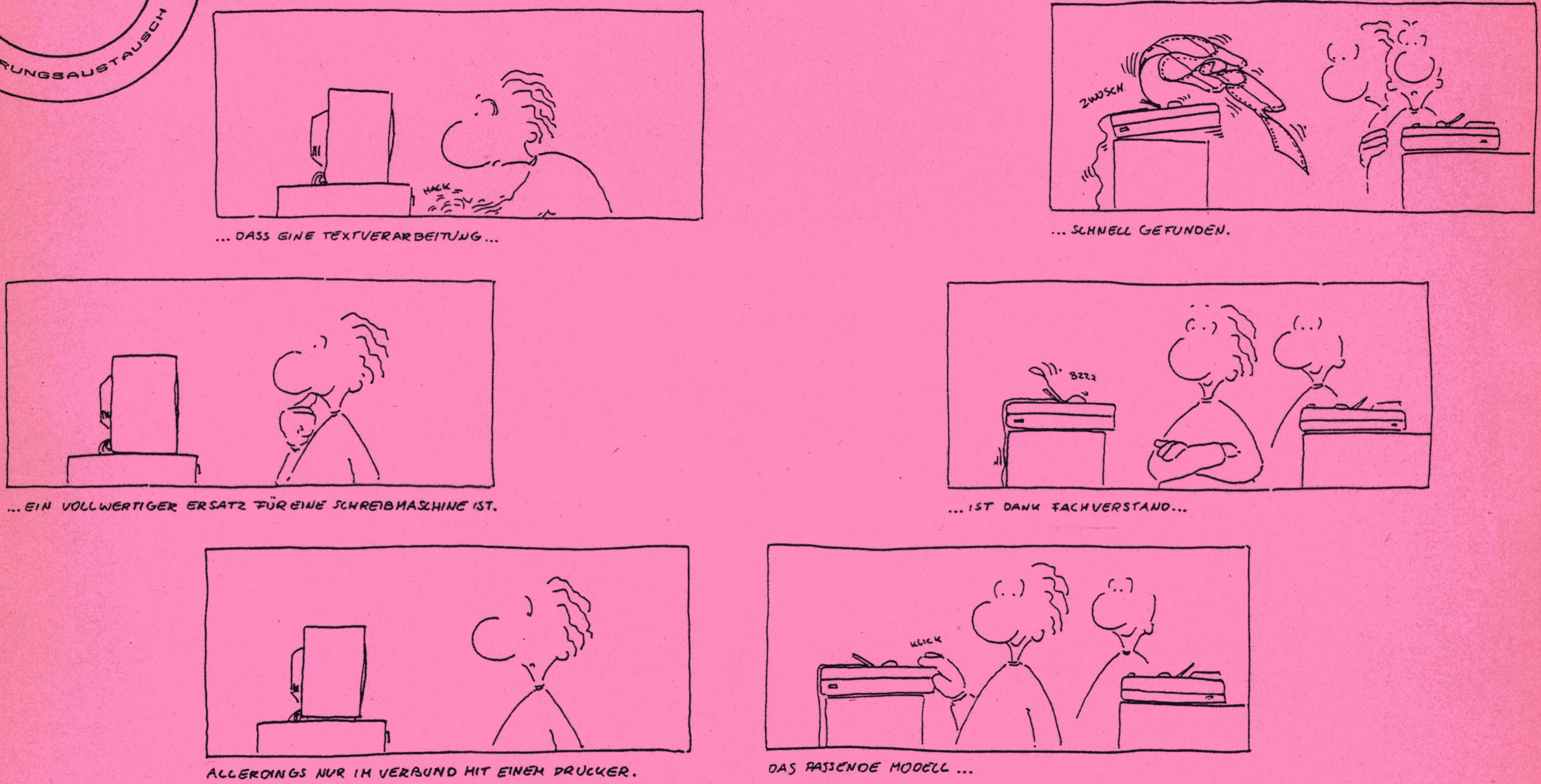
Nachname	Vorname	Straße	PLZ	Ort	Telefon privat	Telefon geschäftlich
Behrendt	Detlef	Schlosserbreite 1a	W- 8018	Grafing	08092 / 9173	
Benner	Achim	Vorm Mühlberg 1	W- 5910	Kreuztal 3	02732 / 3780	(ab 18.00Uhr)
Berndt	Wolfgang	Friedberger Straße 92c	W- 6360	Friedberg 2	06031 / 2963	
Berndt-Jochum	Ilse	Stachelsgut 24	W- 5060	Bergisch-Gladbach 1	02204 / 65254	
Bernhardt	Helmut	Hafenstraße 7	W- 2305	Heikendorf	0431 / 241907	0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	W- 8551	Hausen	09191 / 31698	09191 / 611108
Brans	Jörg	Tieloh 55	W- 2000	Hamburg 60	040 /6906531	
Braun	Harald	Postfach 8011	W- 2300	Kiel 17		
Braun	Günter W.	Postfach 800 226	W- 8000	München 80		
Böckling	Ulrich	Jochaczstraße 61	W- 5410	Hör-Grenzhausen	0264 / 4861	02631 / 895168
Chwolka	Fritz	Saarstraße 34	W- 5173	Aldenhoven	02464 / 8920	
Dose	Volker	Dorfstraße 10	W- 2304	Brodersdorf	04343 / 1357	
Gill	Thomas	Maria-Eich-Str. 34	W- 8000	München 60	089 /8349527	
Günther	Jens	Bannerscheid 7	W- 5231	Neitersen	02681 / 1553	
Halgasch	Gert	Großschonauer Straße 26	O- 8805	Jonsdorf		
Hartmann	Hans-Günther	Möwenstraße 9	W- 2876	Berne 2	04406 / 6911	0421 /2482419
Hebecker	Ulrich	Büsumerstraße 15	W- 7000	Stuttgart	0711 / 734800	
Held	Manfred	Stirner Straße 22	W- 8835	Pleinfeld	09144 / 6563	0911 /2195245
Hermann	Klaus	Forchenstraße 8	W- 7401	Pliezhausen	07127 / 71945	
Hürdler	Manfred	Niederhofer Straße 29	W- 8709	Rimper	09365 / 4235	
Johnen	Willi	Hansemannstraße 1	W- 5160	Düren	02421 / 501305	
Kauka	Dietmar	Straße des Friedens 37	A- 7201	Neukirchen	0037404/850112	
Kemmer	Jürgen	Dorfberg 7	W- 8701	Sulzdorf	09334 / 1050	
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 /4828948	00411 / 258170
Kuhn	Eckehard	Im Dorf 14	W- 7443	Frickenhausen	07022 / 45417	
Littmann	Claus	Plockhorst, Zum Spring 15	W- 3155	Edemissen	05372 / 7796	
Lorenz	Walter	Mahräckerstraße 9	W- 6000	Frankfurt /Main 50	069 / 531656	
Magnus	Andreas	Bismarckstraße 29	W- 4650	Gelsenkirchen	0209 / 870230	
Mahlert	Herbert	Hohenbudbergerstraße 112 A	W- 4100	Duisburg	02135 / 47217	
Mand	Harald	Kl. Flintbeker Straße 7	W- 2302	Flintbek /Kiel	04347 / 3629	0431 /3013580
Menk	Christian	Ollsener Straße 52	W- 2116	Hanstedt	04184 / 7825	
Müller	Kurt	Sophie-Scholl-Ring 3b	W- 2054	Geesthacht	04152 / 70643	04151 / 889137
Mössel	Franz	Schafferstraße 12	I- 39012	Meran		
Neebe	Gerhard	Märkische Straße 186	W- 4600	Dortmund 1	0231 / 416549	
Neueder	Jens	Rudolf-Then-Straße 32	W- 7178	Gschlachtenbretzingen	0791 / 42877	0791 /44-4722
Neumann	Christof	Zeitblomstraße 22/2	W- 7900	Ulm	0731 /6022568	0731 /9749720
Nitschke	Stefan	Germanenstraße 5	W- 7519	Walzbachtal 1	07203 / 452	
Obermann	Hartmut	Mozartring 23	W- 8870	Günzburg	08221 / 30248	
Peters	Jürgen	Heukoppel 14	W- 2000	Hamburg 14	040 /6412371	
Reit	Hermann	Vechter Hof 40	W- 4500	Osnabrück	0541 / 16331	
Retzlaff	Bernd	Kleiner Sand 98	W- 2082	Uetersen	04122 / 43551	04103 / 605310
Riechmann	Michael	Letelner Heideweg 12	W- 4950	Minden	0571 / 36627	
Rinio	Gerd	Rennbahnstraße 9	W- 2000	Hamburg 74	040 /6552630	
Ruschinski	Claus	Pommernstraße 21	W- 4370	Marl	02365 / 34646	
Schilling	Werner	Ehndorfer Straße 340	W- 2350	Neumünster	04321 / 61116	
Schimmer	Jörg	Castellring 55	W- 6369	Nidderau 1	06187 / 25503	
Schmid	Alexander	Entmannsdorf 5	W- 8640	Kronach/Gehülz		
Schmitz	Rainer	Dornierstraße 17	W- 7320	Göppingen	07161 / 22549	
Schober	Frank-Michael	Weberweg 2	O- 7590	Spremberg	0037574/ 4565	
Schoberth	Uwe	Petrus-Waldus-Straße 14	W- 7136	Oetisheim	07041 / 7254	0711 /2011455
Scholz	Hans-Werner	Spitalstraße 54	W- 4054	Nettetal		
Schroers	Horst-Dieter	Breslauer Straße 9	W- 8016	Feldkirchen	089 /9032615	089 /9032615
Schröder	Gerald	Am Schützenplatz 14	W- 2105	Seevetal 1	04105 / 2602	
Schröder	Peter	Theodor-Fahr-Straße 32	W- 2000	Hamburg 62	040 /5311582	
Schröer	Egbert	Joachimstraße 18	W- 4270	Dorsten 1	02362 / 75840	02365 / 495183
Schulte	Hartmut	Entenschnabel 8	W- 3162	Uetze	05173 / 1248	
Schut	Andre	Sanderstraße 26	W- 1000	Berlin 44	030 /6917861	
Schwarz	Wolfgang	Schwedenring 6	W- 8850	Donauwörth	0906 / 3092	
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	W- 5305	Alfter 4	0228 / 643853	
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanloese		
Sonnemann	Harald	In den Eckwiesen	W- 6101	Fischbachtal	06131 /320860	
Stumpferl	Stefan	Hasenberglstraße 57	W- 8000	München 45	089 /3144001	
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	W- 6500	Mainz 1	06131 / 320860	06131 / 395268
Tornow	Wilhelm	Görlitzer Straße 16	W- 2190	Cuxhaven 13	04723 / 1355	04723 / 3063
Vollkmer	Richart	Am Spörkel 69	W- 4600	Dortmund 50		
Volz	Oliver	Am Ochsenwald 37A	W- 7000	Stuttgart 80 (Rohrerhöh	0711 /744051	0711 /6853013
Werner	Heiko	Reichenberger Straße 5	O- 8032	Dresden		
Wittkamp	Heinz	Hindenburgstraße 37	W- 5630	Remscheid	02191 / 75132	
Wulf	Hans-Otto	Im Brahmkamp 38	W- 4250	Bottrop	02041 / 688972	

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

HARDWARE * SOFTWARE

COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80

ERFAHRUNGSAUSTAUSCH



CLUBINFO

38. Ausgabe

Kontaktadresse:
Club 80
Fritz Chwolka
Saarstrasse 34
5173 Aldenhoven
Tel.: 02464/8920

Inhaltsverzeichnis

	Autor & Seite
Clubinternes	
Vorwort	1 - 2
	Hartmut Obermann
Termine	2
	Redaktion
Vorstellung	3 - 6
	Ilse Berndt Jochum
	Günther W. Braun
Börse	
Wer hat was -- wer will was	7
Software	
Der Wahnsinn des Monats	9
ADRESS-TEXT-90 Teil IV	10 - 20
	Willi Johnen
QB-Tool RAHMEN	16 - 18
	Jens Neueder
Hardware	
TRS80 Model I Tuneup	21 - 27
	Helmut, Egbert, Claus
Sonstiges	
Kleines DFÜ-Glossar für Einsteiger	28 - 39
DFÜ-Fachbegriffe	40 - 42
	Hartmut Obermann
Das FIDO-Netz	43 48
	Artikel aus DOSshareware
Die letzten Seiten	
Impressum	8
Schluß	48
	Redaktion
Mitgliederadressenliste	am INFO-Ende

Liebe CLUB 80-Freunde,

es ist höchste Zeit, daß ich mich mal wieder zu Wort melde, denn es gibt zwei "heiße Eisen" aufzugreifen, die mir schon lange auf dem Herzen liegen.

Jahresbeitrag und Clubtreffen 1993

Zunächst zum Jahresbeitrag, der sich ab 1993 (bis auf weiteres) wohl seinen Namen nicht mehr verdient. Klingt seltsam!? Also weiterlesen!!!

Wie ihr sicher alle wisst, führen wir (die eigentliche Arbeit macht ja meine Frau Jutta) nun schon seit einigen Jahren die Kasse des CLUB 80. Es gab Zeiten, da war auf dem Konto, genau wie in unserem Clubinfo, so richtig was los. Diese Zeiten sind leider vorbei. Genau wie im Clubinfo herrscht auf dem Konto Flaute!

Nicht das ihr mich falsch versteht. Es herrscht nicht etwa Ebbe in der Kasse, vielmehr haben wir einen ständig wachsenden Kontostand. Durch die selteneren und spärlicheren Infos wird kaum noch Geld verbraucht und so hat sich ein richtiges kleines Vermögen angesammelt (genau 4137.64 am 03.12.92). Aus diesem Grunde sehe ich mich gezwungen, die Beitrags-Notbremse zu ziehen!

Würden wir im Januar alle Mitgliederbeiträge kassieren, stiege der Kontostand nochmals um ca. 3500 DM. Das steht Ausgaben von nur noch wenigen hundert Mark pro Jahr für Info und sonstige Aktivitäten des Clubs entgegen. Aus diesem Grunde verkünde ich hiermit:

ab sofort (und bis auf weiteres) keine Mitgliedsbeiträge mehr!!!

Ich glaube, damit dürften wohl alle einverstanden sein!? Sollte das nicht der Fall sein, können wir das ja beim Jahrestreffen 1993 besprechen. Womit mir beim nächsten Thema wären.

Das Jahrestreffen des CLUB 80 findet am zweiten Märzwochenende, also vom 12. bis 14.03.1993 statt. Der Ort steht noch nicht genau fest, da aber ich wohl derjenige sein werde, der das Treffen ausrichtet, wird es wohl in der Umgebung von Ulm stattfinden.

Sicher werden jetzt die Nordlichter entsetzt aufstöhnen und das für ungerecht halten. Daran kann aber nur etwas geändert werden, wenn sich jemand aus dem Norden oder der Mitte Deutschlands bereit erklärt, das Treffen auszurichten. Wir hatten 1991 schon einmal versucht, ein Treffen zu veranstalten, das mehrere hundert Kilometer von unserem Wohnort entfernt stattfinden sollte. Dabei sind wir fast "auf den Bauch" gefallen und hatten großen Ärger. Daraufhin fand das Treffen hier "gleich um die Ecke" statt (siehe Info 34) und es wurde ein voller Erfolg.

Wie gesagt, der Termin steht fest und nur der genaue Ort ist noch offen. Ihr könnt euch also das Datum schon mal merken und es wäre schön, wenn möglichst viele sich das Wochenende frei halten und kommen könnten. Außerdem würde ich mich sehr über Anregungen zur Tagesordnung freuen. Mit Sicherheit steht die Wahl eines neuen Vorsitzenden auf dem Programm, das hat mir Fritz Chwolka schon "angedroht". Und auch über die weitere Zukunft des CLUB 80 muß auf dem Treffen gesprochen werden. Denn über eines sind sich fast alle einig: so wie in den letzten zwei Jahren kanns nicht weitergehen!

In diesem Sinne wünschen wir Euch ein frohes Weihnachtsfest
und ein gutes, gesundes und glückliches Jahr 1993,

Jutta und Hartmut Obermann.

Termine... Termine... Termine... Termine... Termine

UTEC Umwelttechnologieforum	Berlin	15.02.- 19.02.93
Electronic Partner	München	27.02.- 28.02.93
CeBIT	Hannover	24.03.- 31.03.93
Büro,Kommunikation,Computer	Nürnberg	06.05.- 08.05.93
CAT	Stuttgart	11.05.- 14.05.93
Computer-Schau	Dortmund	12.05.- 16.05.93
Hobby-tronic	Dortmund	12.05.- 16.05.93
Eltec 93	Nürnberg	17.06.- 19.06.93
HAM RADIO	Friedrichshafen	26.06.- 27.06.93
IFA	Berlin	27.08.- 05.09.93
Fachmesse Elektrotechnik	Dortmund	01.09.- 04.09.93
MICRO SYSTEMS TECH. 1993	Berlin	21.09.- 24.09.93
SYSTEMS	München	18.10.- 22.10.93
PRODUKTRONICA	München	09.11.- 13.11.93
Nord Elektro	Hamburg	18.11.- 20.11.93

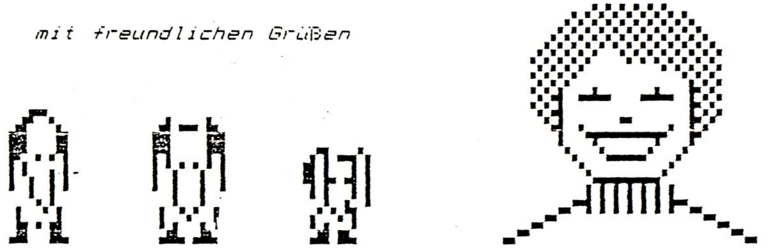
Redaktionsschluß für das nächste Clubinfo ist

Mitte März 93, damit rechtzeitig vor der Jahreshauptversammlung das INFO auch fertig wird.

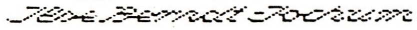
im November 1991

Da bin ich, die Computer-Oma. Es wird Zeit, daß ich mich vorstelle. Im Herzen bin ich 18, auf dem Papier Jahrgang 18. Von Beruf Bildhauerin, Kunstkeramikerin und Buchhalterin. Ich wohne mit drei Dackeln in einem 200 Jahre alten Häuschen. Meine sehr klein geratene Rente bessere ich durch Unterrichten in Kunstkeramik auf. Ich leite den Verein *"die KUNST und KERAMIKFREUNDE e.V."* Die Mitglieder sind Hobbykeramiker. Mein Hobby ist der Computer. Als ich noch 59 war, las ich in der Zeitung die Reklame einer Schreibmaschinenfabrik, mit der Überschrift: *"Bitsy beißt nicht"*. (ich hielt Bitsy für einen Mädchennamen, inzwischen weiß ich wer Bitsy ist). Das wollt ich genau wissen und fuhr in Köln's größtes Bürohaus. Da gab's zwar Geräte, aber niemanden, der sie vorführen konnte, das war 1978. Ich bekam einen Termin, an dem Tag war dann einer vom Werk da, der auf dem Bildschirm mit einem beweglichen Punkt in einem Brief herumfuhrwerkte (so sah ich das damals) und dabei erklärte was er macht, wie löschen, einfügen, ändern usw., das kennt Ihr ja alle. Die Diskette steckte bereits beim Kommen der bestellten Interessenten im Fach und der Brief stand schon auf dem Bildschirm (Irreführung: daß das über Diskette funktioniert und ein Programm erforderlich ist, wurde uns nicht gesagt). 15.000 Mark sollte der Scherz kosten. War zwar begeistert, wie vorteilhaft ich damit Artikel für die Schulen schreiben und vor allem ständig dem neuen Stand anpassen könnte (ich unterrichtete Kunst an Schulen), aber so recht kam ich nicht hinter die Sache. Da las ich in der hiesigen Zeitung die Anzeige einer Computer-Firma in Bergisch Gladbach, das gefiel mir besser, wenn ich nicht klar käme, war das wenigstens hier. Also hin. Ein junger Mann erkundigte sich erst, wofür ich das Gerät nutzen wolle und sagte, dann müsse ich eins mit Diskettenfächern haben. Er steckte ein Textprogramm in ein Gerät und ließ mich gleich selbst daran rumprobieren, wobei er erklärte, wie das vor sich geht. In Erinnerung an die Vorführung in Köln sagte ich, daß muß es können, und das muß es können, was ich so alles in Köln beobachtet hatte. Manches mußte der Händler selbst erst ausknobeln, aber das Textprogramm machte alles mit, was ich in Köln erlebt hatte und dann ein Mann, der sehr gut erklärte und mir erlaubte, ihn notfalls auch Sonntags anzurufen, fand ich doll. So kam ich an einen GENIE, damals der GENIE IIs mit einem Drucker ITOH 8510, Textprogramm und G-DOS kostete die Sache nicht ganz die Hälfte als in Köln. Ich bekam noch eine Diskette geschenkt, die unter anderem ein Schachspiel enthielt. Die deutsche Beschreibung enthielt eine sehr gute Erklärung für das Arbeiten mit BASIC und damit war mein Interesse geweckt. Mein neues Hobby war geboren. Daß der Händler genau so neu war wie ich, aber fachlich versiert, erfuhr ich später von ihm. Heute arbeite ich mit dem GENIE IIIs und III und dem Drucker ITOH C 310. Für heute

mit freundlichen Grüßen



PETER PAULA ROSI und ICH



Dieser Brief, die Zeichnungen und die Unterschrift sind mit dem GENIE-Text-Programm geschrieben.

Anmerkung der Redaktion:
Das Datum stimmt. Leider habe ich die Vorstellung erst jetzt erhalten.

Günther W. Braun
Postfach 800 226
8000 München 80

Guten Tag, liebe Freunde von CP/M-Computern,

mit meinem Commodore 128 D gehöre ich zu den Neuen, die 1992 zum Club gefunden haben. Als gelernter Kaufmann hatte ich die Möglichkeit, mich durch Fortbildung zum "Datenverarbeitungs-Kaufmann" zu qualifizieren, und dadurch entdeckte ich auch mein neues Hobby: Die Anwendung des Heimcomputers zu allen möglichen privaten Zwecken.

Da ich also kein Techniker bin und auch keine Ambitionen habe, das Rad neu zu erfinden, beschäftige ich mich seit Jahren mit dem Sammeln vorhandener CP/M-Software, die so vor dem Untergang gerettet wird. Meine Liste umfasst inzwischen bereits über 200 Marken-Programme oder Programmpakete und wird vielleicht demnächst in diesen Clubinfos hier vorgestellt (wer nicht so lange warten will, kann sie gegen 5 Mark in Briefmarken anfordern).

Auf meinem Rechner läuft aber auch die Software des kleineren Commodore 64, die nicht für das CP/M-Betriebssystem geschrieben wurde. Dabei handelt es sich wohl um die bisher grösste Auswahl an Bildungsprogrammen überhaupt; weshalb ich auch hierüber eine Liste zusammengestellt habe (die Themenübersicht findet Ihr anbei). Für die Zukunft erhoffe ich mir davon Anregungen für die Arbeit unter CP/M.

Beim Umfang des Programmangebots und der Möglichkeit, Disketten auch mit PCs auszutauschen, ist CP/M also immer noch aktuell. Und nicht nur die Software: Die englische Firma Amstrad baut auch immer noch solche Rechner, wie der Prospekt eines modernen Notepad Computers mit Z-80 Prozessor vom Oktober 1992 beweist!

C-64/128-Computereinsatz in Schulunterricht und Fortbildung

I n h a l t :

I. Allgemeines - 2

II. Lexika und Wörterbücher - 6

III. Amateurfunk - 7

IV. Astronomie (*) - 8

V. Biologie (*) - 10

VI. Chemie (*) - 11

VII. Daten-Fernübertragung - 12

VIII. Fremdsprachen - 14

IX. Führerscheine - 16

X. Geographie (*) - 17

XI. Gesundheit - 17

XII. Graphik - 18

XIII. Informatik (*) - 20

XIV. Mathematik (*) - 23

XV. Musik - 27

XVI. Ökologie (*) - 28

XVII. Ökonomie (*) - 29

XVIII. Physik (*) - 29

XIX. Programmiersprachen - 31

XX. Religion - 33

XXI. Schach - 34

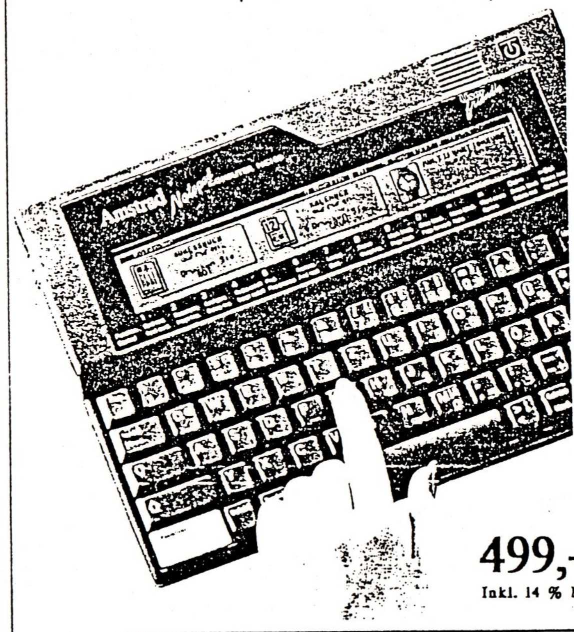
XXII. Statistik - 35

XXIII. Technische Anwendungen (*) - 35

XXIV. Verschiedenes - 37

(*) In diesem Fachgebiet ist eine Teilnahme am Wettbewerb "jugend forscht" in zwei Altersklassen (bis 15, oder von 16 bis 21 Jahre) möglich. Anmeldeschluss ist der 30. November des jeweiligen Jahres. Einen Informationsprospekt gibt's bei: Stiftung Jugend forscht e.V., Beim Schlump 58, 2000 Hamburg 13

NOTEPAD NC100



499,- DM
Inkl. 14 % Mehrwertsteuer

Computer mit Textverarbeitung, Datenbank, Kalender, Rechner- und Alarmfunktion

NOTEPAD NC100

Technik im Detail

Soviel Computer wie möglich und sowenig Technik wie nötig: das ist die Philosophie des Amstrad NOTEPAD NC100. Er ist der ideale Computer für alle, die eigentlich nie einen haben wollten. Mit vier Tasten steuern Sie die integrierten Programme und wechseln bequem zwischen einer komfortablen Textverarbeitung, Datenbank mit Adressen, Terminkalender, Rechner und Alarmfunktionen. Der NC100 ist ideal für unterwegs, denn er verbindet völlig unkompliziert zwei Welten: Organizer-Funktionen mit einer wirklich funktionsfähigen Schreibmaschinen-Tastatur plus Textverarbeitung und Kommunikation.

TASTATUR
Komfortable Tastatur im Schreibmaschinen-Layout, die auch die bequeme Eingabe größerer Textmengen erlaubt.

TEXT
Leistungsstarke Textverarbeitung mit integrierter Rechtschreibprüfung (erkennt 48.000 Wörter). „Suchen und Ersetzen" Funktion. Anzeige von Text in fett, *kursiv* und unterstrichen. Direkte Ausgabe an die meisten Computerdrucker.

ADRESSEN
Speicherung von Namen, Adressen und Telefonnummern. Einträge werden automatisch alphabetisch sortiert. Suchfunktion.

KALENDER
„100jähriger" Terminkalender bis zum Jahr 2099. Alarmfunktion mit Notizeintrag. Schneller Zugriff auf alle Einträge.

ALARM
Weltzeituhr mit 6 programmierbaren Zeitzonen und Weckfunktion. 6 unabhängige Weckrufe mit Notiz. Automatische tägliche, wöchentliche oder monatliche Wiederholung von Weckrufen.

RECHNEN
Rechner mit besonders großformatiger 12stelliger Anzeige. Speicher- und Konstantenfunktion.

SPEICHER
Standardmäßig mit 64 KB Speicher ausgestattet. Bis zu 1024 KB zusätzlicher Speicher in Form von PCMCIA 2.0-Speicherkarten nach Industriestandard (optional).

KOMMUNIKATION
Serielle Schnittstelle nach Industriestandard für den Anschluß an serielle Drucker, PCs oder andere Computer. Parallele Schnittstelle für den Anschluß an die meisten Computerdrucker.

SICHERHEIT
Paßwortschutz für geheime Eingaben.

Kompakt. Komplett. Kompatibel.

Der Amstrad NOTEPAD NC100 wird komplett mit Netzgerät, Transporttasche, Batterien und Handbuch geliefert.

Amstrad. Computertechnik mit System.

Das perfekt abgestimmte Amstrad PC-Programm reicht vom 486er über Standard-PCs, Laptops, Notebooks und Laserdrucker bis zur Bürokommunikation.

Technische Daten NOTEPAD NC100

Prozessor:	Z80
Hauptspeicher:	64 KB, auf bis zu 1024 KB aufrüstbar durch zusätzliche Speicherkarten nach PCMCIA 2.0-Industriestandard (optional)
Grafik:	Supertwist-LCD mit 8 Zeilen à 80 Zeichen (480 x 64 Punkte)
Tastatur:	Schreibmaschinen-Layout mit 64 Tasten, deutsch
Schnittstellen:	1 x parallel, 1 x seriell
Stromversorgung:	4 x 1,5 V-Batterien „AA", 1 x 3 V-Knopfzelle C 2032
Besonderheiten:	Uhr, Lautsprecher, BBC-BASIC Programmiersprache
Maße:	297 (B) x 27 (H) x 210 (T) mm

Lieferumfang:

Tasche, Netzteil, Batterien, Handbuch

Club 80 Börse

Hallo CP/M- und Oldie-Freaks,

gestern ist mir ein NCR Decision Mate V in die Haende gefallen, den sein Besitzer ausschlachten wollte (Disklaufwerke, Netzteil usw.). Das konnte ich natuerlich nicht zulassen und habe das Teil gerettet!

Das mich die Rettung einen Hunni gekostet hat, konnte meine bessere Haelfte ja noch verkraften. Aber als sie die Dimensionen des Geraetes sah, bekam ich was zu hoeren. Aber sie hat ja Recht. Ich habe einfach keinen Platz fuer noch eine Kiste, mein Buero quillt eh schon ueber und als Wohnzimmermoebel eignet sie sich auch nicht. Was also tun!?

Ich suche also jemand, der den NCR Decision Mate V von mir uebernimmt (Handbuecher und MSDOS 2.11 sind dabei). Natuerlich will ich an dem Geraet, das voll funktionsfaehig ist, nichts verdienen. Den Hunderter haette ich allerdings schon gerne wieder zurueck, und natuerlich muss der neue Besitzer den Transport uebernehmen.

Wer also an dem Geraet, das uebrigens 256k RAM und zwei Diskettenlaufwerke sowie einen eingebauten Monitor hat (und das soweit ich weiss sowohl CP/M als auch MSDOS booten kann) Interesse hat, kann sich bei mir melden.

Hier meine Adresse: Hartmut Obermann
Mozartring 23
8870 Guenzburg
Tel.: 08221/30248
FAX : 08221/33575

Allways good Hacking, H.O.!

Herr Doktor, hier ist meine Frau – sie bildet sich ein, ein Computer zu sein!

Impressum

1. Vorsitzender — Fritz Chwolka
Tel.: 0 24 64/89 20 — Saarstraße 34
5173 Aldenhoven

2. Vorsitzender — Gerald Schröder
Tel.: 0 41 05/26 02 — Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator — Andreas Magnus
Tel.: 02 09/87 02 30 — Bismarckstraße 29
4650 Gelsenkirchen

Newdos-Diskothekar — Oliver Volz
Tel.: 07 11/6 85 30 13 — Am Ochsenwald 37A
7000 Stuttgart 80 (Rohrerhöhe)

CP/M-Diskothekar — Rüdiger Sörensen
Tel.: 0 61 31/32 08 60 — Thomas-Mann-Straße 3a
6500 Mainz 1

Clubbücherei — Kurt Müller
Tel.: 0 41 52/7 06 43 — Sophie-Scholl-Ring 3b
2054 Geesthacht

Redaktion — Jens Neueder
Tel.: 07 91/4 28 77 — Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 8870 Günzburg
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

CLUB-80 INFO

BASIC-ECKE FÜR GENIE-BESITZER

Hallo Volker !

Mit etwas Entsetzen und auch wiederum etwas Hoffnung las ich im CLUB-INFO Nr.37 über Dein Problem mit der Textverarbeitung.

Wenn Du Dir einmal das von mir veröffentlichte ADRESS- TEXT-90 anschaust, wirst Du sehen, daß dieses gesamte Programm im Bildschirmformat 80x25 geschrieben ist. Die Textverarbeitung ist auch wunderbar. Es ist kein Programm,daß ich irgendwo abgeschrieben habe. Dieses ADRESS-TEXT-90 habe ich selber in der BASIC-Sprache entwickelt und bis heute immer wieder verfeinert. Zu der damaligen Trommelschläger-Zeit 1984-1986 ist dieses Programm unzählige Male verkauft worden. Ich habe heute noch Kontakte zu Käufern, denen ich die Programme immerwieder erneuere.
Die neuere Version 2.8.A hat nun sogar eine Videokasetten-Verwaltung.

Wenn Du Interesse hast,so schicke mir eine Diskette und ich werde Dir eine neue Version schicken. Die verschiedenen Sachen müßtest Du Dir dann selber ändern.

Nun zurück zur Textverarbeitung. Mit meiner Textverarbeitung kannst Du ganze Seiten ändern und auch einfügen,innerhalb einer Zeile einfügen oder löschen.

Aber da ich nun einmal dran bin,ich weiß auch nicht alles, obwohl ich nur im BASIC arbeite.
Vielleicht kannst Du mir helfen!!

Problem 1: Wie kann ich innerhalb eines Datenstrings$ der RANDUM-Datei
---------- die volle Länge von 255 Zeichn ausnützen,so daß ich Neueingaben,Änderungen und Sortierungen (CMD-File) machen kann?

Problem 2: Wie kann ich die hochauflösende Graphik des GENIE III S compi-
---------- lieren?

D E R   W A H N S I N N   D E S   M O N A T S !!!

Vielleicht etwas zu kompliziert,vielleicht etwas zu wahnsinnig und - eigentlich dürfte es so garnicht funktionieren. Was denn, kein FORMAT,READY und REPEAT-Statement im GENIE-Basic? -- Na und ? -- Das Ding läuft trotzdem,was immer es ach tut... --- Wer blickt durch ??

```
1CLEAR:ONERRORGOTO1:M=2:FORMAT=0TTO5:READY:FORD=21TORSTEP-MÄP:X=MÄD:PRINT
STRING$(M,MÄ7+MÄ6*(-(Y=>X)+(Y<X))-MÄ(-5*(Y<X)))STRING$((MAD=5)*(D=0)+MÄ-D
,-13*(D=0));:Y=Y+X*(Y=>X):NEXTD,MARK:REPEAT:DATA4058962,2435411,2435411,4
058962,2427474,24272231,MCW
```

+++

Hallo Ihr GENIE III S - Besitzer !!

Na, wie funktioniert Euer ADRESS-TEXT-90? Hat es schon jemand abgetippt?
Heute darf ich nun den IV.Teil vorstellen:

**** Briefe drucken mit und ohne Serienadresse ****

Mit diesem Programm kann jeder seinen gesamten Adressen-Bestand anschreiben, wobei die Adressen ausgewählt werden können und sich automatisch in den Brief einfügen.

Das Programm bietet folgende Möglichkeiten um Briefe zu drucken:

* Alle gespeicherten Adressen mit einem bestimmten Brief anschreiben.
** An einzelne Adressen einen bestimmten Brief schreiben.
*** Serienbriefe drucken nach bestimmten Auswahl-Parametern. Die Auswahlparameter sind gleich wie bei ADRESSLISTEN drucken.

Die Briefe sind vorher als ASCII-Datei auf der Diskette mit der TEXTVERARBEITUNG geschrieben worden oder werden neu eingegeben,wenn kein Serienbrief gewünscht wird.

Unter der Betreffzeile kann ein Text von max. 50 Zeichen-Länge eingegeben werden.
Die Wahleingabe der Bankverbindung ist auch möglich.
Weiterhin kann Rollenpapier oder Einzelblatteinzug gewählt werden.

Hinweis: Dieses Programm läuft auch nur mit hochauflösender Graphik und
-------- einer installierten Festplatte.

Nachfolgende Zeilen müssen je nach Bedarf geändert werden:

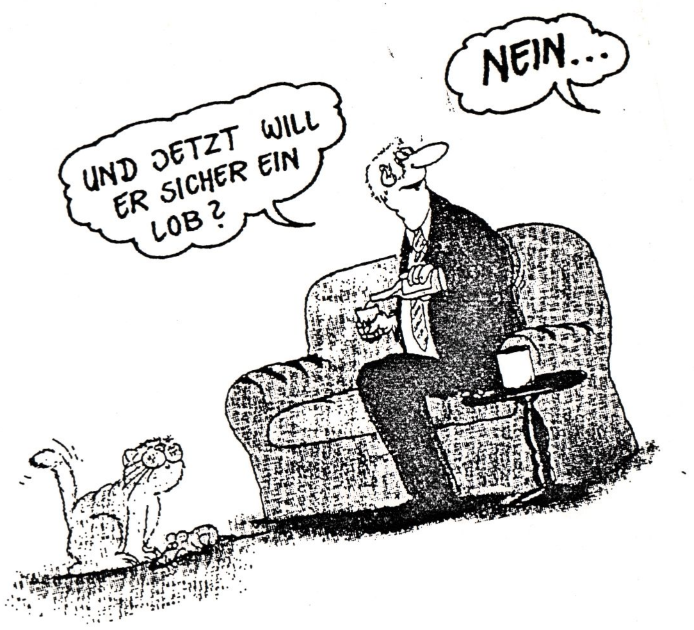
Z.0170	Hochauflösende Graphik
Z.0130	Laufwerksangabe
Z.0170	Anzahlen der Zeilen eines Briefes KT(160) / A3(160)
Z.1810	Namensangabe
Z.1930-1960	Anschrift
Z.1980-2000	Anschrift
Z.2050-2060	Absender in Anschriftenfeld
Z.2180-2300	Bankverbindungen
Z.2320	Ortsangabe
Z.3030	Laufwerksangabe

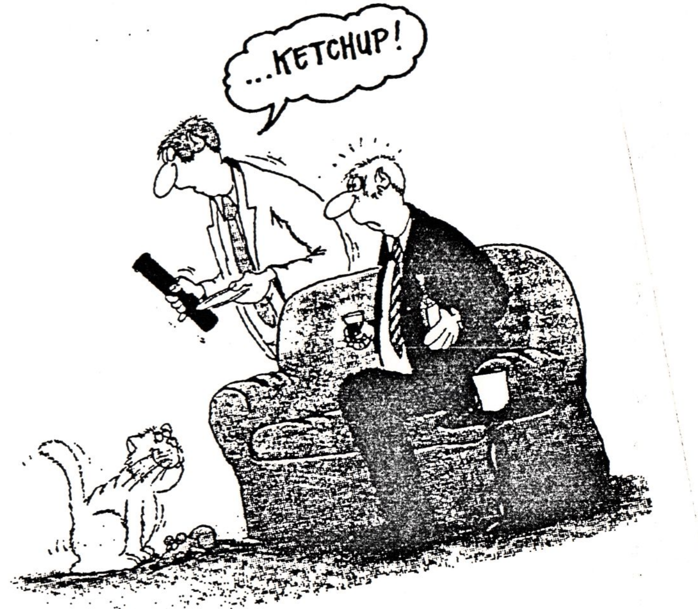
Bereits erfolgte Veröffentlichungen zum ADRESS-TEXT-90:

I.Teil - Club-Info Nr.34: Textverarbeitung
II.Teil - Club-Info Nr.35: Startprogramm und Hauptmenü
III.Teil - Club-info Nr.37: Adressenverwaltung

```
10 '#>> Programmname  << JOHBRIEF/BAS >>                           <<#
20 '#>> Brief-und Textverarbeitung zum ADRESS-TEXT 90 - Privat     <<#
30 '#>> Stand: 07.12.1992                                          <<#
40 '#>> Hochauflösende Graphik                                     <<#
50 '#>> Adressenspeicherung in Laufwerk 6                          <<#
60 '#>> Graphikbild:  >JOHNEN01/RDL< + >TEXTBILD/RDL<              <<#
70 '#>> M I I      H A R D D I S K                                 <<#
80 '#>> Nur für Endlospapier geeignet                              <<#
90 '#>> Drucker ITOH 8510                                          <<#
100 '#------------ Prüfen,ob Datendiskette im Laufwerk -----------#
110 CLEAR:DEFSTRG:DEFINTW
120 GA="":ONERRORGOTO140
130 OPEN"I",1,"JOHADRE1/DAT:6":CLOSE:GOTO170
140 RESUME150
150 WA=6:GOSUB200:PRINT§405,"Im Laufwerk 6 ist keine  <JOHADRE1/DAT - Datei>":PRINT§495,"Prüfen
Sie nochmal mit  < W >.";
160 GA=INKEY$:IFGA<>"w"THEN160ELSE110
170 HOFF:HCLS:HON:CLEAR12000:DEFSTRA-D,G-H,J-L:DEFINTN,Q,X,Z,W:DEFSNGP:DIMDX(7),K(7),KT(160),DA(
7),A3(160),Q(13),CB(13)
180 '#---------------- Graphik -----------------------------------#
190 CC=STRING$(15,140):CY=STRING$(79,32):KQ=STRING$(30,144):HX="####.##":AF="TEXT-EINFÜGEN"
200 GOSUB2890
210 WP=0:HBIG30,225,1,1,1,"BRIEF-UND TEXTVERARBEITUNG":HBIG536,225,1,1,1,LEFT$(TIME$,8);:HBIG536
,202,1,1,1,RIGHT$(TIME$,8);:IFWA=6THENRETURN
220 '#-------------- Erstellung der Programmnamen ----------------#
230 K1="1 Anrede                 : ":K2="2 Vorname/Firmenbez.     : ":K3="3 Name/Firmenname
: ":K4="4 Straße/Postf./Hausnr. : ":K5="5 PLZ/Wohnort            : ":K6="6 Telefonnummer
: ":K7="7 Ansprechpartner        : "
240 GOTO270
250 '#---------------------- Menü-Anzeige ----------------------#
260 WA=6:GOSUB200:WA=0
270 ONERRORGOTO0:PRINT§461,"Programm - Menü":PRINT§541,CC:PRINT§618,"<1> Spezial - Brief":PRINT§
702,"ohne Serienadr.":PRINT§858,"<2> Serien - Brief":PRINT§942,"ohne Serienadr."
280 PRINT§1098,"<3> Spezial - Brief":PRINT§1182,"mit Serienadr.":PRINT§1338,"<4> Serien - Brief"
:PRINT§1422,"mit Serienadr.":PRINT§1578,"<5> Textverarbeitung":PRINT§1738,"<6> Haupt-Menü"
290 WU=VAL(INKEY$):IFWU<1ORWU>6THEN290ELSECLS:ONWUGOTO320,330,690,690,300,310
300 HOFF0:HON2:HNEG:CLEAR50:RUN"JOHTEXTV/BAS"
310 HOFF0:HON2:HNEG:CLEAR50:RUN"JOHMENÜ/BAS"
320 PRINT§459,"Spezial - Brief":PRINT§539,"ohne Serien-Adresse":PRINT§619,"Eingabe der Adresse":
GOTO370
330 PRINT§459,"Serien - Brief ":PRINT§539,"ohne Serien-Adresse":PRINT§619,"Eingabe der Adresse":
GOTO370
340 PRINT§81,"<< Spezial-Brief mit Serienadresse  >>";CY;:GOTO730
350 PRINT§81,"<< Serien -Brief mit Serienadresse  >>";CY;
360 '#------- Adresseneingabe für Programmwahl 1-2 --------------#
370 '#---Anrede          DT(1)-KN
380 PRINT§321,K1;KQ;CY;:GOSUB2950:GOSUB2930:PRINT§1180,"Nach der Eingabe";:PRINT§1259,"< ENTER >
 drücken";:PRINT§1418,"Mit < / > kommen Sie";:PRINT§1500,"zurück ins Menü!";
390 PRINT§1849,"Herrn - Frau - Eheleute - Architekt - Firma - Dr. oder Sonstige";CHR$(7);
400 PRINT§347,;:LINEINPUTDT(1):IFLEFT$(DT(1),1)="/"THENCLOSE:GOTO260
410 IFLEN(DT(1))>30THEN380
420 IFWA=1THENPRINT§1602,CY;:GOTO590
430 '#---Vorn.Firmenb. DT(2)-KV
440 PRINT§481,K2;KQ;CY;:GOSUB2940:PRINT§1854,"Geben Sie den Vornamen oder die Firmenbezeichnung
ein";CHR$(7);
450 PRINT§507,;:LINEINPUTDT(2):IFLEN(DT(2))>30THEN440ELSEIFWA=2THEN630
460 '#---Name Firmenn. DT(3)-KU
470 PRINT§641,K3;KQ;CY;:GOSUB2950:PRINT§1855,"Geben Sie den Familiennamen oder den Firmennamen e
in";CHR$(7);
480 PRINT§667,;:LINEINPUTDT(3):IFLEN(DT(3))>30THEN470ELSEIFWA=3THEN630
490 '#---Straße/Hausnr DT(4)-KS
500 PRINT§801,K4;KQ;CY;:GOSUB2950:PRINT§1861,"Geben Sie die Straße und Hausnummer ein";CHR$(7);
510 PRINT§827,;:LINEINPUTDT(4):IFLEN(DT(4))>30THEN500ELSEIFWA=4THEN630
```

```
520 '#---PLZ Wohnort   DT(5)-KQ
530 PRINT§961,K5;KQ;CY;:GOSUB2950:PRINT§1857,"Geben Sie die 4-stellige PLZ und den Wohnort ein";
CHR$(7);
540 PRINT§987,;:LINEINPUTDT(5):IFLEN(DT(5))>30THEN530ELSEIFWA=5THEN630
550 '#---Telefonnummer DT(6)-KP
560 PRINT§1121,K6;STRING$(16,144);CY;:GOSUB2950:PRINT§1863,"Geben Sie die Telefonnummer ein";CHR
$(7);
570 PRINT§1147,;:LINEINPUTDT(6):IFLEN(DT(6))>16THEN560ELSEIFWA=6THEN630
580 '#---Ansprechpart. DT(7)-KB
590 IFDT(1)="Herrn"ORDT(1)="Frau"ORDT(1)="Fräulein"ORDT(1)="Eheleute"THEN630
600 PRINT§1281,K7;KQ;CY;:GOSUB2950:PRINT§1842," Ansprechpartner mit Anrede eingeben!  z.B. >Herr
 Thomas< oder >Frau Schmitz<";CHR$(7);
610 PRINT§1307,;:LINEINPUTDT(7):IFLEN(DT(7))>30THEN600ELSEIFWA=7THEN630
620 IFWA=1THEN630
630 N=0:FORZ1=1TO7:N=N+1:K(N)=DT(N):NEXTZ1:PRINT§1600,CY;:GA="":GOSUB2950:PRINT§1857,"Sind die o
bigen Eingaben richtig   < J/N > ?";CHR$(7);
640 GA=INKEY$:IFGA="j"THEN690ELSEIFGA="n"THEN650ELSE640
650 WA=0:GOSUB2950:PRINT§1858,"Geben Sie die zu ändernde Zahl ein  < 1-7 >";CHR$(7);
660 WA=VAL(INKEY$):IFWA=0ORWA>7THEN660
670 N=WA:PRINT§1602,"Alte Eingabe ----->>>>>  ";K(N);BN;BN;:DT(N)=""
680 ONWAGOTO380,440,470,500,530,560,600
690 '#--------- Start von der Adresseneingabe her ---------------#
700 IFWU=1ORWU=3THEN730
710 IFWU=2THEN1570
720 '#---- Spezialbrief mit u. ohne Serienadr. / Texteingabe ----#
730 CLS:GOSUB2880:PRINT§2,"Spezial-Brief -- TEXT-NEUEINGABE -- S1=-35/S2=-88/S3=-160 Zeilen -- <
/>=Ende"
740 NN=160                    'Anzahl der möglichen Zeilen
750 IFAY!=1THEN760ELSEII=0
760 L!=73                     'Anzahl der möglichen Zeichen in einer Zeile
770 BP%=245                   'Cursorbeginn bei Texteingabe
780 K=""
790 IFAZ!=2THENBP%=BP%+79:II=II+1:A3$(II)=K:K=""
800 IFII=NNTHENN2=II:PRINT:AY!=0:GOSUB1250:KQ="":GOTO930
810 IFLEFT$(A3$(II),1)="/"THENN2=II-1:PRINT:AY!=0:CLS:GOTO870
820 BP%=BP%+1:IFBP%>1920THENAY!=1:AZ!=0:PRINT§159," ";:PRINTUSING"###";II;:PRINT".Z:";A3(II):GOT
O740
830 GOSUB1290    'editfunktion für Texteingabe
840 GOTO790
850 '#----------------- Textänderung ----------------------------#
860 N2=N2+1:PRINT§319,"";LINEINPUTA3(N2)
870 PRINT§239," ";:NL=0:N1=0
880 NM=13:IFNM>N2THENNM=N2
890 NX=0
900 NX=NX+1:IFNL=160THEN930
910 NL=NL+1:PRINTUSING"###";NL;:PRINT".Z:";A3(NL):IFNL=N2THEN930
920 IFNX=21THEN930ELSE900
930 JQ="":AZ!=0:GA="":PRINT§2,"M E N Ü :   <Ä>ndern   <E>infügen   <B>lättern   <D>rucken";
940 GA=INKEY$:IFGA="ä"THEN990
950 IFGA="b"THENGX="B":GOTO1060
960 IFGA="e"THEN1090
970 IFGA="d"THENGA="":GOTO1590ELSE940
980 '#------------------ Textänderung ----------------------------#
990 KQ="Ä":IFGX="B"THENII=N1+1ELSEII=1
1000 PRINT§2,"Zum Löschen einer ganzen Zeile als 1.Zeichen der Zeile ein < § > setzen     ";
1010 BP%=246:AZ!=0:L!=73:K=A3(II)
1020 SL!=1:ONERRORGOTO1030:GOSUB1290:A3(II)=K:K=""
1030 RESUME 1040
1040 II=0:CLS:KQ="":GOTO870
1050 '#--------------- Text Blättern ----------------------------#
1060 N1=N1+21:IFN1=<N2THENPRINT§239," ";:FORI=1TO20:PRINTCY:NEXTI:PRINT§239," ";:GOTO880
1070 GOTO870
1080 '#--------------- Text einfügen ----------------------------#
1090 JQ="X":IFN2>160THENGOSUB1250:GOTO930
1100 II=0:NN=0:PRINT§2,CY;:PRINT§2,AF;" -- Max.21 Zeil.-- Anzahl der Zeilen ";:INPUTNN:IFNN=0ORN
N>21THEN1100
```



```
1110 PRINT§57,"-- Nach Zeilennr.";:INPUTII:'IFII=0ORII)N2THEN3310
1120 IFNN)(160-N2)-1THENNN=160-N2:GOSUB1250
1130 N4=N2+NN:FORNS=N4TOII+1STEP-1:NP=NS-NN:IFNP(IITHENA3(NS)="§":GOTO1150
1140 A3(NS)=A3(NP)
1150 NEXT:CLS:PRINT§2,CY;:PRINT§2,AF;:PRINT§159," ";:PRINTUSING"###";II;:PRINT".Z:";A3(II)
1160 QR=0:L!=73:BP%=245:K=""          'BP%=Startcourser bei Texteinfügen
1170 FORNS=II+1TOII+NN:IFAZ!=2THENBP%=BP%+79:K=""
1180 BP%=BP%+1:QR=QR+1
1190 IFNN(10THENPRINT§BP%-4,USING"#";QR;:GOTO1210
1200 IFNN=)10THENPRINT§BP%-5,USING"##";QR;
1210 PRINT".E: ";:GOSUB1290:A3(NS)=K
1220 IFLEFT$(A3(NS),1)="/"THENA3(NS)="§":NS=II:A3(NS+NN)="§":GOTO1240
1230 NEXT
1240 N2=N2+NN:CLS:GOTO870
1250 GA="":PRINT§2,CY;:PRINT§17,"Die Länge des Briefes ist auf 160 Zeilen begrenzt !";CHR$(7);:F
ORI=1TO1200:NEXTI:PRINT§2,CY;:RETURN
1260 WT=N2:FORI=1TOWT:KT(I)=A3(I):A3(I)="":NEXT:GOTO1270
1270 GOTO1590
1280 '#---------- Editfunktion für Texteingabe -------------------#
1290 IFLEN(K))L!THENK=STRING$(L!,32)
1300 IFLEN(K)(L!THENK=K+STRING$(L!-LEN(K),32)
1310 IFGA="ä"THEN1340
1320 IFJQ()"X"THEN1350
1330 PRINT§BP%,K;
1340 PRINT§BP%,CHR$(14);:GOTO1370
1350 PRINT§BP%,K;:IQ=II+1       'iq=Zähler für neueingabe-Zeilen
1360 PRINT§BP%-7,IQ:PRINT§BP%,CHR$(14);
1370 SL!=1
1380 IFSL!)L!THENSL!=L!ELSEIFSL!(1THENSL!=1:PRINTCHR$(15);
1390 PRINT§BP%+SL!-1,CHR$(14);
1400 K1=INKEY$
1410 IFK1=""THEN1400
1420 IFGA="ä"THENIFK1=CHR$(10)THENA3(II)=K:K="":BP%=BP%+80:II=II+1:K=A3(II):IFBP%)1846THEN1540EL
SE1380
1430 IFGA="ä"THENIFK1=CHR$(11)THENA3(II)=K:K="":BP%=BP%-80:II=II-1:K=A3(II):IFBP%(246THEN1540ELS
E1380
1440 IFGA="ä"THEN1460
1450 IFK1=CHR$(10)THEN1400
1460 IFK1)=CHR$(32)THENPRINT§BP%+SL!-1,K1;:MID$(K,SL!,1)=K1:SL!=SL!+1:GOTO1380
1470 IFK1=CHR$(8)THENSL!=SL!-1:PRINTCHR$(15);:GOTO1380
1480 IFK1=CHR$(9)THENSL!=SL!+1:PRINTCHR$(15);:GOTO1380
1490 IFK1=CHR$(31)THENSL!=1:K=STRING$(L!,32):PRINT§BP%,K;:PRINTCHR$(15);:IFGA="ä"THENK="§":GOTO1
380ELSE1380
1500 IFK1=CHR$(25)THENK=LEFT$(K,SL!-1)+" "+LEFT$(RIGHT$(K,L!-SL!+1),L!-SL!):PRINT§BP%,K;:PRINTCH
R$(15);:GOTO1380
1510 IFK1=CHR$(24)THENIFSL!=1THEN1380ELSEK=LEFT$(K,SL!-2)+RIGHT$(K,L!-SL!+1)+" ":PRINT§BP%,K;:SL
!=SL!-1:PRINTCHR$(15);:GOTO1380ELSEIFK1=CHR$(11)THENAZ!=1:GOTO1520ELSEIFK1=CHR$(10)ORK1=CHR$(13)
THENAZ!=2:GOTO1520
1520 IFGA="ä"THENRETURN
1530 IFJQ()"X"THENE!=VAL(K)
1540 PRINTCHR$(15)
1550 RETURN
1560 '#--------- Serienbriefdruck ---------------------------------#
1570 GOSUB2750:ONERRORGOTO0:PRINT§131,"TN="+KA;:GOTO1610
1580 '#------------ Abfrageroutinen ------------------------------#
1590 IFWU=1ORWU=2ORWU=3THENGOSUB2890
1600 '#------------- Abfrage Betreffzeile ------------------------#
1610 GOSUB2940:PRINT§1601,"9 Betreff - Zeile        : ":PRINT§1681,STRING$(75,144);"        ";:PRIN
T§1681,;:LINEINPUTKF
1620 IFLEN(KF))75THEN1610ELSEIFWU(3THEN1740ELSEGOSUB2700
1630 '#---------------- Druck mit Serienadresse -------------------#
1640 PRINT§1841,CY;:PRINT§1200,"Suchbegriff - Eingabe    : ";KQ;"       ";:PRINT§1226,;CHR$(7);:LI
NEINPUTGN
1650 IFGN=""THEN1660ELSEIFLEN(GN))30THEN1640ELSEIFASC(GN)(32ORASC(GN))126THEN1640
1660 GOSUB3030
```


Kleine QB-Tool-Kiste

Jens Neueder

Hier, unter dieser Rubrik, möchte ich Euch kleine Tools vorstellen, die das BASIC-Leben leichter machen und sich in fast jedes Basic-Programm einbinden lassen.
Die Tools sind für/in QB oder auch PDS 7.1, lassen sich sicher auch für jedes andere Basicprogramm anpassen. Gleichzeitig hoffe ich Euch dazu animieren zu können, auch eine Tool-Kiste (wenn nicht schon vorhanden) anzulegen. Vielleicht könnt Ihr das eine oder andere Beispiel dazu beisteuern.

SUBroutine RAHMEN

Zeichnen eines Rahmens unter Verwendung der ASCII-Zeichen mit Übergabe aller Parameter zur Erzeugung des Rahmens.

Der Aufruf aus dem Programm erfolgt durch:

CALL Rahmen(Kopf$, Zlio%, Slio%, ZL%, SL%, ColR%, ColI%, RT%, S%)

Hierbei bedeuten:

Kopf$	Überschrifttext in der Rahmenzeile
Zlio%	Beginnzeile des Rahmens (Ecke links oben)
Slio%	Beginnspalte (Ecke links oben)
ZL%	Rahmenlänge (nach unten in Zeilen)
SL%	Spaltenbreite (nach rechts in Spalten)
ColR%	Textfarbe des Rahmens
ColI%	Textfarbe des Innenraumes
RT%	Angabe des Rahmentyps (RTA) (1-8 möglich)
S%	Angabe für Schatten (1=Ja/0=Nein)

zuvor wird SUB natürlich deklariert:

DECLARE SUB Rahmen(Kopf$, Zlio%, Slio%, ZL%, SL%, ColR%, ColI%, RT%, S%)

```
SUB Rahmen(Kopf$, Zlio%, Slio%, ZL%, SL%, ColR%, ColI%, RT%, S%)

'* Rahmenzeichen definieren
        DIM RTA$(1 TO 8)
        RTA$(1) = "    - "
        RTA$(2) = "      "
        RTA$(3) = "    - "
        RTA$(4) = "      "
        RTA$(5) = "■■■■■■"
        RTA$(6) = "      "
        RTA$(7) = "??????"
        RTA$(8) = "!!!!!!"
```

```
1670 IFEOF(1)THEN170ELSELPRINTCHR$(27)"L"+"000";
1680 GQ="":GET1
1690 KZ=DX(1)+DX(2)+DX(3)+DX(4)+DX(5)+DX(6)+DX(7)
1700 IFINSTR(KZ,GN))0THEN1710ELSE1670
1710 IFX1=0THEN1720
1720 KN=DX(1):KV=DX(2):KU=DX(3):KS=DX(4):KO=DX(5):KP=DX(6):KB=DX(7):GOTO1770
1730 '#--------------------- Anschrift drucken ------------------#
1740 GOSUB2700
1750 IFWU=1THENGOSUB2900:PRINT§108," -Druck des Brief/Text";:GOTO1790
1760 IFWU=2THEN1790
1770 IFWU=3THENGOSUB2910:PRINT§108," -Druck des Brief/Text";:GOTO1800
1780 IFWU=4THENGOSUB2920:PRINT§108," -Druck des Brief/Text";:GOTO1790
1790 KN=DT(1):KV=DT(2):KU=DT(3):KS=DT(4):KO=DT(5):KP=DT(6):KB=DT(7)
1800 GOSUB2990
1810 PRINT§1842,"Welchen Briefkopf ? -- <E>lisabeth Muster - <W>olfgang Mus - <G>enie-Pr.-E. ";
1820 B2=INKEY$:IFB2<>"e"ANDB2<>"w"ANDB2<>"g"THEN1820
1830 IFB2="w"ORB2="g"THEN1860
1840 PRINT§1841,"                Wünschen Sie den Eindruck der Bankverbindung ?  < J/N >        ";
1850 B5=INKEY$:IFB5<>"j"ANDB5<>"n"THEN1850ELSE1880
1860 PRINT§1841,"                Wünschen Sie den Eindruck der Bankverbindung ?  < J/N >        ";
1870 B4=INKEY$:IFB4<>"j"ANDB4<>"n"THEN1870
1880 GOSUB2950:YY=0:LPRINTCHR$(27)"r";:LPRINT:LPRINT:LPRINT:LPRINT:LPRINTCHR$(27)"f";
1890 POKE16424,76:POKE16425,1                '76 Zeilen pro Seite festlegen
1900 LPRINTCHR$(27)CHR$(69);CHR$(27)"!";CHR$(14):LPRINTCHR$(27)"L"+"004";
1910 IFB2="g"THEN1980
1920 IFB2="w"THEN1940
1930 LPRINT"  ELISABETH MUSTER":GOTO1950
1940 LPRINT"  WOLFGANG MUSTER"
1950 LPRINT"  6666 MUSTERSTADT";TAB(29)"TEL.06666/888888"
1960 LPRINT"  MUSTERSTRASSE 1"
1970 GOTO2010
1980 LPRINT"   GENIE - SOFTWARE";TAB(37)"WOLFGANG MUSTER"
1990 LPRINT"       ENTWICKLUNG";TAB(33)"6666 MUSTERSTADT"
2000 LPRINT"   TEL.06666/888888";TAB(31)"MUSTERSTRASSE 1"
2010 LPRINTCHR$(27)CHR$(34):LPRINTCHR$(15):LPRINTCHR$(27)"E"
2020 '#------------ Absender in Anschriftenfeld drucken ---------#
2030 LPRINTCHR$(27);"Q";:LPRINT:LPRINT
2040 IFB2="w"ORB2="g"THEN2060
2050 LPRINTTAB(9);CHR$(27)"X"+"E.Muster - 6666 Musterstadt - Musterstr.1":LPRINTCHR$(27)"Y";:GOT
02070
2060 LPRINTTAB(9);CHR$(27)"X"+"W.Muster - 6666 Musterstadt - Musterstr.1":LPRINTCHR$(27)"Y";
2070 LPRINT:LPRINTCHR$(27)"E";:LPRINT
2080 '#--------------- Anschrift drucken -------------------------#
2090 LPRINTCHR$(27)"L";"010";
2100 IFB2="w"ORB2="g"THEN2210
2110 IFB5="j"THEN2160
2120 LPRINTKN
2130 IFRIGHT$(KV,1)=" "THENKV=LEFT$(KV,LEN(KV)-1):GOTO2130
2140 LPRINTKV;" ";KU
2150 LPRINTKS:LPRINT:GOTO2310
2160 LPRINTKN;TAB(65)CHR$(27)"X";"Bankverbindung:"CHR$(27)"Y"
2170 IFRIGHT$(KV,1)=" "THENKV=LEFT$(KV,LEN(KV)-1):GOTO2170
2180 LPRINTKV;" ";KU;TAB(60)"Sparkasse Musterstadt"
2190 LPRINTKS;TAB(60)"Kontonummer :9999999999"
2200 GOTO2300
2210 IFB4="j"THEN2260
2220 LPRINTKN
2230 IFRIGHT$(KV,1)=" "THENKV=LEFT$(KV,LEN(KV)-1):GOTO2230
2240 LPRINTKV;" ";KU
2250 LPRINTKS:LPRINT:GOTO2310
2260 LPRINTKN;TAB(65)CHR$(27)"X";"Bankverbindung:"CHR$(27)"Y"
2270 IFRIGHT$(KV,1)=" "THENKV=LEFT$(KV,LEN(KV)-1):GOTO2270
2280 LPRINTKV;" ";KU;TAB(60)"Sparkasse Musterstadt"
2290 LPRINTKS;TAB(60)"Kontonummer :8888888888"
2300 LPRINTTAB(60)"Bankleitzahl:300 000 00"
```



```
'* Oberste Rahmen-Zeile erzeugen
RM$ = LEFT$(RTA$(RT%), 1) + STRING$(SL% - 2,
                MID$(RTA$(RT%), 5, 1)) + MID$(RTA$(RT%), 2, 1)
CALL PrintF(Slio%, Zlio%, RM$, ColR%)
IF LEN(Kopf$) > 0 THEN
        IF LEN(RM$) > SL% - 4 THEN RM$ = LEFT$(Kopf$, SL% - 4)
                                              ELSE RM$ = Kopf$
        CALL PrintF(Slio% + (SL% - LEN(RM$)) / 2, Zlio%, RM$,
                                                        ColR%)
END IF

'* Mittlere Rahmen-Zeilen erzeugen
RM$ = RIGHT$(RTA$(RT%), 1): F$ = RIGHT$(RTA$(RT%), 1)
FOR i% = 1 TO ZL% - 2
        CALL PrintF(Slio%, Zlio% + i%, RM$, ColR%)
        CALL PrintF(Slio% + SL% - 1, Zlio% + i%, F$, ColR%)
NEXT i%

'* Unterste Rahmen-Zeile erzeugen
RM$ = MID$(RTA$(RT%), 3, 1) + STRING$(SL% - 2, MID$(RTA$(RT%), 5,
                                    1))+ MID$(RTA$(RT%), 4, 1)
CALL PrintF(Slio%, Zlio% + i%, RM$, ColR%)

'* Innenzeilen mit Blanks in Textfarbe löschen/füllen
RM$ = SPACE$(SL% - 2)
FOR i% = Zlio% + 1 TO Zlio% + (ZL%-2)
        CALL PrintF(Slio% + 1, i%, RM$, ColI%)
NEXT i%

'* gegebenenfalls Schatten darstellen
IF S% THEN
        IF MONOCHROM% THEN DEF SEG = &HB000 ELSE DEF SEG = &HB800
        Unt% = (Zlio% + ZL% - 1) * 160 + (Slio% - 1) * 2 + 3
        FOR j% = 0 TO SL% - 1
                Platz% = Unt% + 2 * j%
                POKE Platz%, PEEK(Platz%) MOD 16
        NEXT j%
        Rech% = Zlio% * 160 + (Slio% + SL% - 1) * 2 + 1
        FOR j% = 0 TO ZL% - 1
                Platz% = Rech% + 160 * j%
                POKE Platz%, PEEK(Platz%) MOD 16
        NEXT j%
        DEF SEG

END IF
'* PrintF(x%,y%,Z$,Farbe%)      schnelle Assemblerausgaberoutine
'* RM$                 Ausgabestring zum Erzeugen des Rahmens
'* Unt%                Lage der unteren Schattenlinie
'* Rech%               Lage der Schattenlinie rechts
'* Platz%              Lage des Schattenzeichens im Bildschirmspeicher
'* i%, j%              Laufvariable für Schleifen
END SUB
```

```
2310 LPRINTKO:LPRINT:LPRINT:LPRINT
2320 LPRINTTAB(60)"Muster, den ";LEFT$(TIME$,8):LPRINT:LPRINT
2330 IFLEN(KF)=0THENLPRINT:LPRINT:LPRINT:GOTO2360
2340 IFLEN(KF))0THENLPRINTCHR$(27)"X";KF;JQ:LPRINT:LPRINT
2350 '#----------- Anredeschlüssel -------------------------------#
2360 IFRIGHT$(KU,1)=" "THENKU=LEFT$(KU,LEN(KU)-1):GOTO2360    'Name
2370 IFRIGHT$(KB,1)=" "THENKB=LEFT$(KB,LEN(KB)-1):GOTO2370    'Ansprechpartner
2380 IFLEFT$(KV,3)="Dr."THENIFLEFT$(KN,4)="Herr"THENLPRINT"Sehr geehrter Herr Dr. ";KU+"!":GOTO2
520
2390 IFLEFT$(KV,3)="Dr."THENIFLEFT$(KN,4)="Frau"THENLPRINT"Sehr geehrte Frau Dr. ";KU+"!":GOTO25
20
2400 IFLEFT$(KN,5)="Herrn"THENLPRINT"Sehr geehrter Herr ";KU+"!":GOTO2520
2410 IFLEFT$(KN,4)="Frau"THENLPRINT"Sehr geehrte Frau ";KU+"!":GOTO2520
2420 IFLEFT$(KN,8)="Fräulein"THENLPRINT"Sehr geehrtes Fräulein ";KU+"!":GOTO2520
2430 IFLEFT$(KN,8)="Eheleute"THENLPRINT"Sehr geehrte Frau ";KU+"!":LPRINT"Sehr geehrter Herr ";K
U+"!":GOTO2520
2440 IFLEFT$(KN,9)="Architekt"THENLPRINT"Sehr geehrter Herr Architekt ";KU+"!":GOTO2520
2450 IFKB=""THEN2460ELSE2470
2460 LPRINT"Sehr geehrte Damen und Herren !":GOTO2520
2470 IFLEFT$(KB,4)="Herr"THENLPRINT"Sehr geehrter ";KB+"!":GOTO2520
2480 IFLEFT$(KB,3)="Frl"THENLPRINT"Sehr geehrtes ";KB+"!":GOTO2520
2490 IFLEFT$(KB,4)="Frau"THENLPRINT"Sehr geehrte ";KB+"!":GOTO2520
2500 GOTO2460
2510 '#------------------- Textdruck ----------------------------#
2520 LPRINT
2530 IFWU=2THEN2560
2540 IFGA="X"THEN2560
2550 WT=N2:FORI=1TOWT:KT(I)=A3(I):A3(I)="":NEXT
2560 FORI=1TOWT:IFLEFT$(KT(I),1)="§"THEN2610
2570 IFYY=35THENLPRINTCHR$(12):GOSUB2990:KI="2":GOSUB3000:GOTO2600
2580 IFYY=88THENLPRINTCHR$(12):GOSUB2990:KI="3":GOSUB3000:GOTO2600
2590 IFYY=141THENLPRINTCHR$(12):GOSUB2990:KI="4":GOSUB3000:GOTO2600
2600 YY=YY+1:LPRINTJU;KT(I)
2610 NEXTI:LPRINTCHR$(12):PRINT§1841," Der Druck ist fertig!    <K>opie drucken  <*>
     <M>enü";CHR$(7);                                   (*) <W> Weitere Drucke
2620 GA="":GA=INKEY$:IFGA="w"THEN170ELSEIFGA="k"THEN2630ELSEIFGA="m"THEN2650ELSE2620
2630 GA="X":GOTO1880
2640 '#-------- Adresse nach Briefdruck abspeichern -------------#
2650 IFWU<3THEN2660ELSEIFWU=3THEN1670ELSE170
2660 GOSUB2950:PRINT§1851,"Wollen Sie die Kundenanschrift abspeichern ?   < J/N >"
2670 GA=INKEY$:IFGA="n"THEN170ELSEIFGA="j"THEN2680ELSE2670
2680 GOSUB3030:GOTO170
2690 '#--------- Drucker einstellen -----------------------------#
2700 IFINP(253)=63THEN2710ELSEPRINT§1841," Ihr Drucker ist nicht startbereit.     Prüfen Sie de
n Drucker (SEL-Lampe)! ";:PRINT§1841,CY;CHR$(7);:GOTO2700
2710 PRINT§1841,"  Ist das Formular richtig eingestellt ?    <D>rucken   <M> zurück zum Menü  ";
CHR$(7);
2720 GA=INKEY$:IFGA<>"d"ANDGA<>"m"THEN2720ELSEIFGA="m"THENCLOSE:GOTO170
2730 JB=CHR$(14):JC=CHR$(15):JG=CHR$(27)+"N":JH=CHR$(27)+"E":JI=CHR$(27)+"Q":JN=CHR$(27)+"!":JO=
CHR$(27)+CHR$(34):JQ=CHR$(27)+"Y":JU=CHR$(27)+"L"+"010":JV=CHR$(27)+"L"+"012":RETURN
2740 '#--------------- Textabfrage -------------------------------#
2750 CLS:PRINT§205," ":CMD"DIR 5,/TXT"
2760 PRINT§458,"Nebenstehende Briefe";:PRINT§538,"oder Texte sind ge-";:PRINT§618,"speichert !";
2770 GA="":PRINT§1441,"8 Text-Bezeichnung      : ":PRINT§1467,STRING$(8,144)+"/TXT:5";CY;:PRINT§
1467,;:LINEINPUTKA:KA=KA+"/TXT:5"
2780 IFLEN(KA)<>14THEN2840
2790 ONERRORGOTO2830
2800 OPEN"I",2,KA:WT=0:I=0
2810 IFEOF(2)THENCLOSE2:WT=I:RETURN
2820 I=I+1:LINEINPUT#2,KT(I):GOTO2810
2830 RESUME2840
2840 CLOSE:GOSUB2950:PRINT§1841,"Text nicht da oder falsch eingegeben  ---- < W >iederholung
< M >enü";
2850 GA=INKEY$:IFGA<>"m"ANDGA<>"w"THEN2850
2860 KT="":IFGA="w"THEN2750ELSE170
```



```
2870 '#--------------- Bildschirmroutinen ----------------------#
2880 ONERRORGOTO0:CLS:HCLS:HLOAD"TEXTBILD/RDL":RETURN
2890 CLS:HLOAD"JOHNEN01/RDL":RETURN
2900 PRINT§81,CY;:PRINT§82,"Spezial-Brief o. Serienadr.":RETURN
2910 PRINT§81,CY;:PRINT§82,"Spezial-Brief m. Serienadr.":RETURN
2920 PRINT§81,CY;:PRINT§82,"Serien -Brief m. Serienadr.":RETURN
2930 X6=1098:FORX=1TO6:PRINT§X6,STRING$(20,32);:X6=X6+80:NEXTX:IFGQ="Q"THENGQ="":RETURNELSERETUR
N
2940 X6=298:FORX=1TO18:PRINT§X6,STRING$(20,32);:X6=X6+80:NEXTX:IFGQ="Q"THENGQ="":RETURN
2950 PRINT§1840,CY;:RETURN
2960 X6=81:FORX=1TO16:PRINT§X6,CY;:X6=X6+80:NEXTX:RETURN
2970 HBIG30,225,1,1,1,"                              ":RETURN
2980 '#------------- Drucker-Routinen --------------------------#
2990 POKE16424,72:POKE16425,1:RETURN
3000 LPRINT:LPRINT:LPRINT
3010 LPRINTJU;:LPRINTTAB(30)"- Seite ";KI;" -":LPRINT:LPRINT:LPRINT:RETURN
3020 '#----- Abspeicherung der Kundenanschrift ------------------#
3030 FQ=0:OPEN"R",1,"JOHADRE1/DAT:6":X1=0:FIELD1,30ASDX(1),30ASDX(2),30ASDX(3),30ASDX(4),30ASDX(
5),16ASDX(6),30ASDX(7):IFWU=3THENRETURN
3040 FQ=LOF(1):FQ=FQ+1:PRINT§139,USING" Adress-Nr.:#### ";FQ;
3050 LSETDX(1)=KN:LSETDX(2)=KV:LSETDX(3)=KU:LSETDX(4)=KS:LSETDX(5)=KO:LSETDX(6)=KP:LSETDX(7)=KB
3060 PUT1,FQ:CLOSE:RETURN
```

..Hardware ..Hardware

TRS80 Model I Tuneup

Helmut Bernhardt, Egbert Schröer, Claus Ruschinski

1. Vorwort

In einem Club Info Jahrgang 1990 bot Hans-Günther Hartmann Hardware Restposten unter anderem für das Model I an. Dazu gehörte auch das Big Mem Board, welches den TRS80 Model 1 durch Banking auf 94 K erweiterte. Kurz nach Erscheinen des Restposten-Angebots war ich (E.S.) im Besitz dieses Boards und mußte mir die dazugehörige Paperware von 1cm Dicke bei Helmut besorgen.
Der wiederum schlug mir vor, anstatt des Big Mem Boards seinen Adreßdecoder und Banker einzubauen. Nach Sichten der Möglichkeiten schien mir dies auch der eleganteste Weg zu sein. Allerdings mußte die Schaltung von Helmut, der diesen Umbau bisher nur für den Genie I ausgeführt hatte, ferndiagnostisch ausgeguckt und in endlosen Telefonsessions mit mir abgeglichen werden (vielen Dank Helmut), da gleichzeitig auch ein Double Density Controler zu kreieren war. Da ich keinerlei Hardware Kenntnisse besitze bestand meine Arbeit nur aus dem Nachbau nach Helmut´schen Angaben und Beschreibung der auftretenden Fehler. Alle Angaben und Fehler wurden von mir dokumentiert um Nachbauern die Arbeit zu erleichtern und Helmut sich die Mühe zu ersparen nochmals den Kram auszugucken.

Es sei an dieser Stelle nicht verschwiegen, daß der Umbau auf dem Model I nie vollständig beendet wurde, da eine Überspannung im VEW-Netz meinem zu diesem Zeitpunkt 8 Jahre alten Rechner plus einem nagelneuen 80 Track Laufwerk den Garaus machte. Mittlerweile hat Claus Ruschinski das Teil soweit repariert, daß es wieder mit Single Density läuft; Double Density ist noch nicht möglich, obwohl der Doubler in Ordnung ist (vielen Dank Claus).
Den Umbau habe ich aber - mit Helmuts Hilfe - auf einem Genie I fortgeführt und bis auf eine falsch geschaltete Adresse (A13; tja Helmut immer noch !!) läuft die Sache.

Der Umbau Plan sieht also so aus:

Das Model 1 wird mit 8 Stück 41256-RAMs anstelle der 4116er auf dem CPU-Board bestückt. Dafür sind vorher noch einige Umrüst-Arbeiten fällig (dazu später mehr). In dem Sockel des Z80 wird das FC-Switchboard mit einem Z80B (oder höher) gesteckt, welches die Freigabe-Signale für alle Memory Mapped Signale beisteuert. Das Level II Basic befindet sich auch auf diesem Board. Anschließend wird der 256K Banker eingebaut, um mit den 256K auch vernünftig umgehen zu können. Dann wird von Hand eine Schaltung gestrickt, die das Umschalten entweder der unteren oder oberen 256K mit dem Banker steuert, also je nach zu bootendem Betriebssystem NewDos oder CP/M Anforderung. Und völlig unabhängig davon wird eine Minimal-Speed-Umschaltung mit einem Handschalter konstruiert.

Das Tuneup besteht somit aus 4 Teilen:
1) Aufbau eines Double Density Boards auf Basis eines WD2793
2) Anschluß eines neuen Adreßdcoders
3) Einbau eines 256 K (oder mehr) Bankers
4) Speed up (von 1,7 auf 5,3 MHz)

Beachten: Bei der Japan Version des Trash 80 müssen die beschriebenen Änderungen geprüft und verglichen werden !!!!

2. Der Doubler

Unter dem Titel SELBSTBAU DOUBLER FÜR EXP1 ist in einem älteren Club-Info hierzu ein Artikel von Helmut veröffentlicht worden. Diese Bau-Anleitung betrifft nur den GENIE I. Für den TRS 80 sind einige kleine Modifkationen notwendig.
Grundlage des Doublers ist der WD2793 Floppycontroler. Für Einzelheiten zu diesem Board wird auf den Artikel verwiesen.
Folgende Änderungen müssen für den TRS 80 Model I durchgeführt werden: (die Bezeichnungen der IC´s und Widerstände entsprechen den Angaben im Hardware Handbuch Expansion Interface Cat. Nr. 26-1140/1141/1142).

1) Die invertierenden Treiber Z50 und Z51 (74LS240) müssen ausgelötet und, am besten gesockelt, durch die nicht invertierenden pinkompatiblen Treiber 74LS245 ersetzt werden. Sollten Fehler auftreten, können durch einfaches Aushebeln die alten Treiber eingesetzt werden.

2) FD DATA geht von Pin 27 des durch ein Huckepack-Board zu ersetzenden WD1771 FDC an Pin 8 des IC´s Z32, Typ 74LS04. Man lötet dieses IC am besten aus und sockelt ein IC gleichen Typs ohne Pin 8 und 9 (abkneifen oder besser aus der Fassung biegen).

3) Von Pin 9 des Z32 geht über den Widerstand R22 READ DATA* an den Floppy Disk Bus J5 (Pin 30). Der Widerstand R22 muß ausgelötet werden.

Dies sind alle Änderungen. Der WD1771 FDC wird nun aus seinem Sockel herausgelöst und durch das Huckepackboard mit WD 2793 FDC ersetzt.

Durch Booten und gleichzeitiges Drücken der BREAK-Taste sollte man nun in das LEVEL 2 Basic gehen, und die einwandfreie Funktion des neuen FDC mit folgendem kleinen Basic Programm testen:

```
10:FOR X = 14304 TO 14319
20:  PRINT PEEK X;
30:NEXT X
40:PRINT
50:GOTO 10
```

Es werden folgende 16 Bytes ausgegeben:

```
ersten   4: 63 127 191 ???       ? für uns ohne Bedeutung
nächsten 4: ?? ??? ??? ???
         4: ?? ??? ??? ???
         4: 128 0   0   0        0 zufällig
```

Von den Adressen 14317 an sollte man nun mit POKE Adresse Zahl Werte Schreiben und mit PEEK wieder auslesen können.
Ist dies möglich, müßten alle Anschlüsse einwandfrei sein.

Das Doubler Board wird, wie in obigem erwähnten Artikel beschrieben, für die einwandfreie Funktion von Datenseparator und Schreibvorkompensation an den Potentiometern P1 und P2 und dem Kondesator C1 mittels Oszilloskop oder wie bei mir in Ermangelung eines Selben durch wiederholte Bootversuche und ständiges Verändern von P2 und C1 die günstigste Einstellung für das Lesen einer Diskette gefunden. Mit P1 (Schreiben einer Diskette) kann ähnlich Verfahren werden (Methode: Try and Error).

3. Der neue Adreßdecoder und das Banking Board

Eleganter Adreßdecoder für GENIE I. Unter dieser Überschrift veröffentlichte Helmut in Info Nr.28, Seite 74 ff eine Schaltung, die einen BIGMEM-ähnlichen Banker für den GENIE I darstellt.

Diese Variante kann dem TRS 80 zugänglich gemacht werden. Dabei wird das gleiche Adapter Board mit den gleichen PALs und einem 27128 Eprom mit Level II Basic eingesetzt. Hier soll nur noch beschrieben werden, wo die von diesem Adapter erzeugten Signale angelegt werden, und welche Änderungen durchzuführen sind, damit auf dem CPU Board des TRS 80 4164er oder 41256er RAM-Bausteine eingesetzt werden können. Die Banking Logik und den Anschluß ohne neuen Adreßdecoder wurde in einem Info unter dem Titel "256K RAM für Z80-System" von Helmut beschrieben.
Auf geht´s !
Zum CPU-Board und zum Expansion Interface zu verdrahtende Signale des Adapters:

/RAM (CPU) Z67,74LS367,Pin 15
die Brücken Pins 12,5 und Pins 11,6 im Jumperfeld X71 müssen offen sein.

/MRD (CPU) Z67,74LS367,Pin 1
(CPU) Z68,74LS367,Pin 1
diese Pins sind bereits miteinander verbunden; die Verbindung dieser Pins zu Z74,74LS00, Pin 6 muß durchtrennt werden.

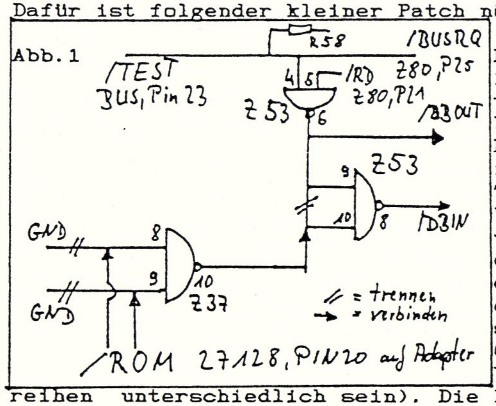
/KB (CPU) Z3,74LS368,Pin 1
(CPU) Z4,74LS368,Pin 1
diese Pins sind bereits miteinander verbunden; die Verbindung dieser Pins mit Z36,74LS32, Pin 11 muß durchtrennt werden.

/VID (CPU) Z31;Z49;Z64 alle 74LS157 jeweils Pin 1
diese Pins sind miteinander verbunden; die Verbindung dieser Pins zu Z36, 74LS32, Pin 8 muß durchtrennt werden.

/FLO (EXP) Z39,74LS155, Pins 2,14
diese Pins sind miteinander verbunden; die Verbindung dieser Pins zu Z40, 74LS139, Pin 12 muß durchtrennt werden.

Im Expansion Interface sind bei Umrüsten auf 64K oder 256K (oder mehr) im CPU-Board alle RAMS 4116 RAMs und die Treiber Z29 und Z31, beide vom Typ 74LS244 zu entfernen.
Da das ROM nun innerhalb der CPU-Datentreiber liegt, dürfen die Lesetreiber nicht freigegeben werden, wenn das ROM gelesen wird. Dafür ist folgender kleiner Patch nötig:

Abb. 1



Die Verbindung der Pins 9 und 10 von Z53,74LS132 ist zu trennen. Der Pin (dieser beiden), der nicht mehr mit Pin 6 dieses ICs verbunden ist, wird mit Pin 10 von Z37, 74LS02 verbunden. Die Verbindung der Pins 8 und 9 von Z37, 74LS02 mit GND wird durchtrennt (ACHTUNG !! eventuell werden andere ICs dadurch von GND abgetrennt; solche müssen dann wieder an GND gelegt werden. Dies kann bei den verschiedenen Baureihen unterschiedlich sein). Die Pins 8 und 9 von Z37, 74 LS02 werden mit /ROM vom Adapter-Board verbunden (dort 27128 Pin 20).

Um auf dem CPU-Board 4164er oder 41256er RAMs einsetzen zu können, sind - aufgrund der Konzeption des Adapters - folgende Änderungen durchzuführen:

1) Die Leiterbahnen, die alle Pins 9 der RAM-Sockel miteinander verbindet ist von +5V abzutrennen (eventuell an mehreren Stellen). Alle Kondensatoren, die dann noch Verbindung mit den Pins 9 der RAMs haben, werden ausgelötet.

2) Die Leiterbahn, die alle Pins 8 der RAM Sockel miteinander verbindet, ist von -5V abzutrennen (eventuell wird -5V noch an anderer Stelle benötigt -> unbedingt prüfen !! ob dort noch vorhanden). Alle Kondensatoren, die dann noch Verbindung mit den Pins 1 der RAMs haben, werden ausgelötet.

3) Die Leiterbahnen, die alle Pins 8 der RAM Sockel miteinander verbindet, ist von +12V abzutrennen (eventuell wird +12V noch an anderer Stelle benötigt -> prüfen, ob dort noch vorhanden !!). Die Kondensatoren, die mit den Pins 8 verbunden sind, bleiben bestehen. Die Pins 8 der RAM-Sockel werden mit +5V verbunden.

4) Das Signal A7 wird von Pin 6 des Z35, 74LS157 abgetrennt. Dieser Pin wird mit Pin 11 von Z38, 74LS367 verbunden (Signal A14). Auch hier ist zu prüfen, ob A7 dadurch nicht eventuell von einem anderen IC abgetrennt wurde. In diesem Fall wäre A7 dorthin frei zu verdrahten. Wenn 4164er RAMs eingesetzt werden sollen, wird A7 (von Pin 11 des Z39, 74LS367) an Pin 14 des Z51, 74LS157 gelegt und A15 (von Pin 9 des Z38, 74LS367) wird an Pin 13 des Z51, 74LS157 gelegt. Der Pin 12 des Z51 wird mit den Pins 9 der RAM-Sockel verbunden. Wenn mit dem 256K-(1MB-)Banker 41256er-RAMs eingesetzt werden sollen, wird auf Z35 oder Z51 ein weiteres IC vom Typ 74LS 157 mit den Pins 1,8,15 und 16 huckepack aufgelötet. Alle anderen Pins dieses ICs werden leicht abgespreizt, um Kontakt mit den Pins des ICs darunter zu vermeiden. An die Pins 2,3,5 und 6 dieses Huckepack-ICs ist mit den Pins 9 der RAM-Sockel und Pin 7 mit den Pins 1 der RAM-Sockel zu verbinden.

Bei gleichzeitigem Einsatz des 256K-Bankers und des Adreßdecoder-Adpaters bieten sich folgende Vereinfachungen und Verbesserungen:

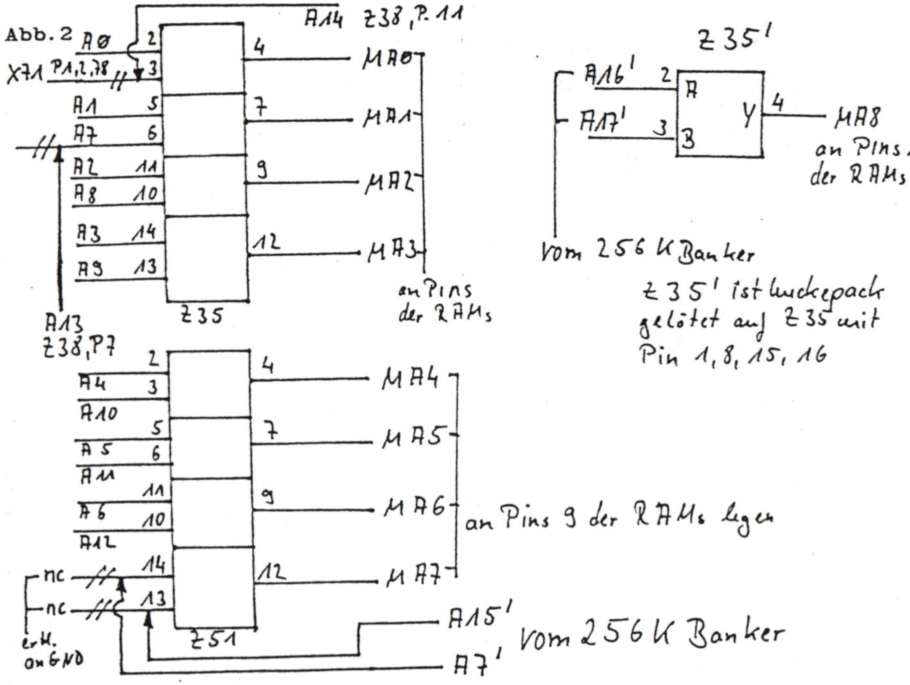
1) Das Freigabesignal des Latches 74LS273 auf dem 256K-Banker wird nicht auf dem Banker erzeugt; stattdessen wird das Signal /QFB (Pin 17 des PAL16L8 auf dem Adapter) an Pin 11 des 74LS273 auf dem 256K-Banker geführt. Die Selektion einer Bank erfolgt dann durch Ausgabe der Bank Nr. an Port FBH.

2) Das Signal A15, das dem Banker zugeführt wird, sollte für die Benutzung unter CP/M (Banking der unteren 32K des Z80-Adreßraums) direkt A15 sein und für die Benutzung unter NewDos/80 (Banking der oberen 32K) ein invertiertes A15. Um das Banking abwechselnd in beiden Betriebssystemen nutzen zu können, soll durch Software einstellbar sein in welcher logischen Speicherhälfte gebankt werden kann. Auf dem Adapter erfolgt bereits eine softgesteuerte Umschaltung von A15 in /A15 mit dem XOR-Gatter (Pins 1,2 und 3) vom 74LS86. Das an das PAL20L8 geführte Signal an Pin 3 dieses Gatters braucht nur noch invertiert zu werden und erfüllt dann die Anforderung, als Eingangssignal A15 des 256K-Bankers zu dienen. Dafür werden die Pins 3 und 4 des Sockels des 74LS86 auf dem Adapter miteinander verbunden. Pin 5 wird mit Pin 14 (+5V) verbunden und Pin 6 ergibt das gewünschte Signal A15 für den Banker. Dieses Signal ist durch den Pegel von D4 an Port FCH steuerbar, womit auch vorgegeben wird, ob die Memory Mapped Baugruppen in den unteren 16K oder an entsprechender Stelle in den oberen 16K des Z80-Adreßraumes liegen.

Nochmals zur Verdeutlichung:
Da die 41256er RAMs 8 Bit Refresh Adressen brauchen, müssen A0-A7 zusammen durch den Adreßmultiplexer geführt werden. Da das A7 des Z80 für das Refreshing keine Funktion hat, ist stattdessen A7´ des 256K Bankers verwendet worden. Zur Erleuterung siehe folgende

Abbildung. Die dort zu verdrahtenden Signale A7´, A15´, A16´ und A17´ stammen vom 256K-Banker.

Abb.2



Zum Schluß dieses Bauabschnittes muß ich nochmals darauf hinweisen, daß beim Durchtrennen von Leiterbahnen ein Signal eventuell nicht nur von der gewünschten Stelle abgetrennt werden, sondern auch noch von anderen ICs. Manchmal ist es nicht ohne weiteres zu erkennen (wenn z.B. die Leiterbahnen unter dem IC weiterführen), daß an dieser Leitung noch mehr angeschlossen ist (Ich empfehle deshalb alle betreffenen ICs auszulöten und anschließend zu sockeln). Wenn das der Fall ist, ist die Leitung auf beiden Seiten des Pins - der nicht mehr damit verbunden sein soll - zu durchtrennen und das Signal ist vom Ausgangspin zu dem anderen Zielpin mit isolierter dünner Litze oder Wrapdraht zu verbinden.

Da die in Kapitel 3 beschriebenen Umbauten aufgrund der mir vorliegenden Schaltpläne meines TRS80 Model 1 von Helmut ausgeguckt wurden, kann für den Einzelfall nicht angegeben werden, ob eine oder zwei Auftrennungen nötig sind und/oder nachverdrahtet werden muß.

4. Speed up auf 5.3 Mhz (oder mehr??)

Nachdem nun der schwierigste Teil hoffentlcih überstanden ist kommt nun eine Minimal Speedup Handumschaltung.

Dazu wird die Verbindung zwischen den Pins 5 und 12 des Z69, 74LS74 durchtrennt und die Pins 13 und 12 dieses Ics miteinander verbunden. Wenn die zu durchtrennende Leitung nicht erreichbar ist - Modellabhängig - wird Pin 5 durchgekniffen. Diese Änderung kann auch bei einer eventuell nötigen Abrüstung des Speed up bestehen bleiben. Man kann an Z56 verschieden Taktraten einstellen: Pins 8,9,11,12 haben Taktraten von 1,7 MHz, 3,5 MHz, 5,3 MHz usw.. Die Pins 8 und 9 werden an die Außenkontakte eines Umschal-

ters gelegt. Der Mittelkontakt wird mit Pin 12 von Z72, 74LS367 verbunden. Die bisherige Verbindung Z56, Pin 8 -> Z72, Pin 12 wird durchtrennt. Man sollte zunächst 3,5 MHz ausprobieren. und anschließend anstelle von Pin 9 auch Pin 11 von Z56 an den Umschalter legen - 5.3 MHz - und testen. Die 41256 RAMs machen das ohne weiteres mit, wenn die Leitungen zum Umschalter möglchst kurz gehalten werden.

5. Zusammenfassung

Nochmals den Einbau Adreßdecoder (FC-Switchboard) Banker und Speed up in Kurzform:

RAM:

+12V, -5V, +5V	von den Pins 1,8,9 abtrennen; Prüfen !!!
+5V	an Leitung, die mit Pins 8 der RAMs untereinander verbunden sind.
Alle Kondensatoren, entfernen.	die die Pins 1 bzw. 9 untereinander verbinden

---

Refresh der 41256er	RAMs: Abb.2 und entsprechender Info Artikel
Z35, 74LS157	2 Leitungen trennen - 3 und 6 - 1x huckepack löten
Z51, 74LS157	2 Leitungen trennen - 13 und 14 -

Alte ROMs Z33/34 entfernen (eventuell kann, muß nicht, Z21, 74LS156 entfernt werden ->Geschwindigkeit !)

/RAM Z67, 74LS367	/RAM mit Pin 15 verbinden, bestehende GND Verbindung trennen.
/KBD Z3, Z4, 74LS368	/KBD mit Pins 1 von Z3/Z4 verbinden, alte Leitungen dorthin durchtrennen
/VID Z31, Z49, Z64 74LS157	
Z7, 74LS74	/VID an Pins 1 von Z31,49,64 und Pin 4 von Z7 verbinden
Z36, 74LS32	Verbindung zu Pin 8 von Z36 trennen

Auf dem Expansion Board:

/FLO Z39, 74LS1	/FLO an Pins 2 und 14 von Z39,
Z40, 74LS139	Verbindung dieser Pins mit Pin 12 von Z40,74LS139 trennen

Im Expansion Board alle RAMs und die Treiber Z19, Z31,74LS244 entfernen. ?FLO kann anstelle von A15 über das 40 polige Buskabel geführt werden.

Geschwindigkeit:

Z69, 74LS74	*Pins 5 und 12 Verbindung trennen. Pins 12 und 13 verbinden.

* falls die Leitung nicht erreichbar, wird Pin 5 durchgekniffen.

Z56, 74LS92	8,9,11,12 -->Taktfrequenz Pins 8 und 9 an Außenkontakt - eventuell Pin 11 für 5,3MHz - des Umschalters
Z72, 74LS367	Mittelkontakt mit Pin 12 von Z72, 74LS367 verbinden
Z56 -> Z72	Verbindung Z56, Pin 8 nach Z72, Pin 12 durchtrennen

6. Hardware Bauteile

Zum Ätzen der Boards und Brennen der unschuldigen PAL´s müßt Ihr schon den Helmut bequatschen. Über die entstandenen Unkosten werdet ihr sicherlich mit ihm einig. Kopien der Schaltpläne TRS80 Model I CPU und Expansion Board stelle ich zur Verfügung - frankierten Rückumschlag beilegen -. Vom Japan CPU-Board habe ich auch einen Schaltplan der Firma RB-Elektronik.

7. Software

Für den Betrieb von CP/M 3.0 auf dem Model I ist als Bootlaufwerk ein 80 Track Laufwerk nötig. Es gibt ein gebanktes und ungebanktes CP/M 3.0, dessen Loader auf den PAL Banker gepatcht ist. Die Sources sind von Peter Petersen, Stand vom 22.2.89.

Das TRS 80 CP/M 3.0 Diskformat sieht wie folgt aus:

80 DS DD 5,25"

Spurverfahren	0	1.Sektor	1
Spuren 80		letzter Sektor	10
Sektoren/Spur	10	Density	double
Dir-Größe	256	Bytes/Sektor	512
K-Bytes/Block	2	Wr Precomp Spur	81
Systemspuren	2		
Skew	nein		
Head Bit Side0	0		
Head Bit Side1	1		

Allzeit genügend Lot auf dem Kolben

Egbert Schröer, November 1992

„Wer legt eigentlich bei euch fest, was ferngesehen wird? Du oder deine Frau?"
„Wir stimmen immer darüber ab. Allerdings, bei Stimmengleichheit entscheidet meine Frau."

Kleines DFUe-Glossar fuer Einsteiger

Wie war doch frueher alles einfach, als es nur Akustikkoppler mit 300 Baud gab. Heute werden Modems angeboten, die mehr als 9600 Bit/Sekunde ueber die Telefonleitung senden. Dieses kleine Glossar soll den DFUe-Einsteigern helfen, die einzelnen Begriffe richtig einzuordnen. Heute gibt es eine ganze Reihe unterschiedlicher Uebertragungsverfahren.

Zunaechst etwas Grundlegendes ueber die Telefonleitung, auf der die Datenuebertragung stattfindet. Das normale Telefonnetz ist fuer das Sprachband kozipiert und uebertraegt Frequenzen zwischen 300 und 3400Hz. Dies ist auch die Bandbreite, an die sich die Modems halten muessen.

Die Uebertragungsgeschwindigkeit wird in Bit pro Sekunde (BPS) gemessen. Leider wird hier oft faelschlicherweise die Einheit "Baud" verwendet (benannt nach dem Franzosen E. Baudot), die aus der Fernschreibtechnik stammt. In Baud wird die Anzahl der Infor- mationsaenderungen pro Sekunde angegeben. Die Datenrate (BPS) kann sich aber von der Schrittgeschwindigkeit (Baud) unterscheiden. Dazu ein Beispiel, das den Sachverhalt verdeutlicht: Wir defini- eren fuer unsere Uebertragungsstrecke zwei binaere Zustaende 0 und 1. Die 0 soll einer Spannung von 0 Volt entsprechen, die 1 einer Spannung von 5 V. Hier ist die Baudsrate gleich der Anzahl der uebertragenen Bits/Sekunde (bis 300 BPS stimmen Baudrate und Bit- rate auch tatsaechlich ueberein).

Weil wir eine analoge Uebertragungsleitung haben, koennen wir auch eine andere Vereinbahrung treffen: Es werden vier Spannungswerte verwendet, 0V, 5 V, 10 V und 15 V. Die Bits werden nun zu Paaren (Dibits)zusammengefasst. Die Zuordnung wird folgendermassen ge- waehlt:

00	0 V
01	5 V
10	10 V
11	15 V

Nun lassen sich mit gleicher Baudrate (!) doppelt soviele Informationsbits uebertragen. Man hat dann z.B. 300 Baud, aber 600 BPS.

Das Verfahren kann man noch erweitern, indem man 3 oder 4 Bits zu einer Einheit zusammenfasst. Weitere Verfahren zur Uebertragungstechnik werden weiter unten behandelt.

Es stellt sich nun die Frage, wie hoch sich die Datenrate bei der Telefonleitung ueberhaupt schrauben laesst. Die Telefonverbindung hat einen zulaessigen Frequenzbereich von 300 Hz bis 3400 Hz. Bedingt durch die Daempfung lassen sich nur etwa maximal 2500 Hz nutzen. Die maximale Baudrate betraegt das Zweifache der Grenz- frequenz f, also 5000 Baud. Bei einer analogen Verbindung fliesst als weitere Groesse das Rauschen ein. Fuer die Telefonleitung ergibt sich ein Dynamikbereich von -40 dB bis etwa -3 dB um Uebersprechen zwischen einzelnen Leitungen zu vermeiden. Fuer den Einfluss des Rauschens gibt es eine Formel aus der Nachrichtentechnik fuer die maximale Uebetragungskapazitaet K:

$$K = 2 * f * ld(1 + (I/R))$$

Der Quotient I/R gibt das Verhaeltnis Information/Rauschen an; fuer eine gute Verbindung kann man hier etwa 30 dB setzen. Daraus ergibt sich eine maximale Kapazitaet von etwas ueber 24000 BPS. Moderne Modems mit 14400 BPS kommen diesem Idealwert schon recht nahe.

Modulationsarten

- Bei der Amplitudenmodulation wird die Amplitude (Signalspannung) des Signals veraendert, das eine konstante Frequenz besitzt.

- Bei der Frequenzmodulation wird die Frequenz (Tonhoehe) bei einen Signal konstanter Amplitude veraendert.

- Bei der Phasenmodulation hat das Signal wieder konstante Frequenz. Es werden hier Phasenspruenge in die Sinusschwingung "eingebaut" (Das Ganze ist ohne Bild etwas schwer zu erklaeren; ich versuch es mal). Stellen Sie sich eine Sinusschwingung vor. Ein Phasensprung fuehrt dann zu einer bestimmten Amplitude, die vom Phasenwinkel abhaengt, d.h. Die Sinuswelle wird in ihrem Schwingungsanfang um den entsprechenden Phasenwinkel veraendert.

Doch nun zu den einzelnen Uebertragungsstandards nach CCITT (Comite Consultativ International Telegraphique et Telephonique):

V.21 (300 bps)

Dies ist der aelteste Standard, der bei den Akustikkopplern verwendet wird. Die Bits werden in Toene unterschiedlicher Frequenz gewandelt (Frequenzmodulation). Fuer zwei Kanaele benoetigt man vier Frequenzen:

	Senden	Empfangen
Kanal 0	1180 Hz	1850 Hz
Kanal 1	980 Hz	1650 Hz

V.23 (1200/75 bps)

Dieser Standard wird fuer die BTX-Uebertragung verwendet, bei der die Datenmengen ind den beiden Richtungen hoechst unterschiedlich sind. Vom BTX-Rechner zum Benutzer weden die Daten mit 1200 bps uebertragen, von der Tastatur des Benutzers zum BTX-Rechner mit 75 bps. Fuer die Uebertragung mit 1200 bps wird fast das gesamte Frequenzband belegt, die 75 bps kommen gerade noch durch. Auch verwendet fuer 1200/1200
halbduplex.

V.22 (1200 bps)

Hier werden jeweils zwei Bits zu einen sogenannten "Dibit" zusammengefasst. Diesen Dibits wird dann ein Phasenwinkel zugeordnet: 00 = 0 Grad, 01 = 90 Grad, 10 = 270 Grad, 11 = 180 Grad. Es werden also 600 Zustaende/Sekunde (= 600 Baud), aber 1200 BPS uebertragen. V.22 gilt auch fuer 600 BPS (nur 2 Phasenwinkel). Grafisch in die Ebene projiziert sieht das dann so aus:

```
                 90
                  |
                  O
                  |
                  |
                  |
                  |
 180  ----O-------+-------O----  0
                  |
                  |
                  |
                  |
                  O
                  |
                 270
```

Die Uebertragung laeuft hier Vollduplex ab. Beide Modems senden die Informationen auf einem eigenen Traeger: der rufende Modem (Originate) uebertraegt mit 1200 Hz, der antwortende Modem mit 2400 Hz.

V.22bis (2400 bps)

Auch hier wird wieder mit Phasenverschiebung gearbeitet, jedoch wird zusaetzlich die Amplitude moduliert --> Quadratur-Amplitudenmodulation. Zusaetzlich werden die Phasenwinkel 45 Grad, 135 Grad, 225 Grad und 315 Grad eingefuehrt. Mit einer Baudrate von 600 lassen sich so bei jedem Schritt 4 Bit uebertragen. Zusaetzlich ist V.22bis auch zu V.22 kompatibel, so dass auch 1200 bps moeglich sind. Die Grafik ist schon komplexer:

```
                 90
     135         |         45
      0    0     |    0    0

      0    0     |    0    0
                 |
180 -------------+------------ 0
                 |
      0    0     |    0    0

      0    0     |    0    0
    225          |         315
                270
```

V.27ter (4800 bps) und V.29 (9600 bps)

Wenn man bei dem vorhergehenden Verfahren die Schrittrate verdoppelt (1200 Baud) kann man 4800 bps uebertragen. Eine weitere Verdoppelung fuehrt dann zu 2400 Baud und 9600 bps, denn es wer- den je Schritt ja immer 4 Bit uebertragen. Diese Verfahren koennen aber nur noch halbduplex uebertragen, das die gesamte Bandbrei- te belegt wird. Sie kommen hauptsaechlich bei der Telefax-Ueber- tragung zum Einsatz, da hier der Datenstrom in eine Richtung geht und der Empfaenger nur kurz quittiert.

Da die Telefonleitung keineswegs stoerungsfrei ist (Knacken, Rauschen, usw.) muss hier auch technisch hoher Aufwand getrieben werden um die Uebertragung zu ermoeglichen. Fax-Geraete besitzen auch die Eingenschaft, bei schlechten Leitungen von 9600 bps auf 7200 bps oder 4800 bps zurueckzuschalten (Fallback). Das V.29-Phasendiagramm sieht so aus:

```
                  90
                   |
     135           0           45

      0     0      |     0     0
                   0
      0     0      |     0     0
                   |
180 -0-----0-------+------0-----0- 0
                   |
      0     0      |     0     0
                   |
                   0
      0     0      |     0     0
                   |
     225           0          315
                   |
                  270
```

Es gibt noch einige Abarten von V.29, bei denen es einen zusaetzlichen Rueckkanal mit 300 BPS gibt. Auf diesem Kanal kann der Empfaenger trotzt des Halbduplexbetriebs dem Sender Nachrichten zukommen lassen (Empfangsbestaetigung, Abbruch, etc.).

V.32 (9600 bps)

Beim Modem will man natuerlich Duplexuebertragung haben. Hier wird mit Hilfe von Signalprozessorbausteinen das Sendesignal aus dem Signalgemisch herausgefiltert und so kann die Information des Partners erkannt werden. Die Modulationsfrequenz ist hier 1800 Hz. Beide Stationen benoetigen eine Trainingsphase, bei der immer nur ein Partner sendet und so seine Echosperre anpassen kann. Hier wird auch das "Trellis-Verfahren" angewendet, bei dem vier Datenbits in fuenf Bits codiert uebertragen werden. Das fuenfte Bit wird aus den vier Datenbits errechnet und wirkt aehnlich wie ein Pruefbit bei der Codierung von sieben Bit in einem Byte. Damit ergibt sich ein um 3 dB besseres Signal/Rauschverhaeltnis gegenueber der reinen Quadraturmodulation. Das Trellis-Verfahren subtrahiert zur Echobeseitigung die Sendedaten vom Signalgemisch und extrahiert so die Empfangsinformation. Dieses komplexe Verfahren wird uebrigens auch bei ISDN verwendet.

V.32bis (14 400 BPS)

Eine nochmalige Erhoehung von Schrittgeschwindigkeit (2400 Baud) und die Hinzunahme weiterer Phasenwinkel wird mit dem Trellis-Verfahren diese phantastische Geschwindigkeit erreicht. Der Aufwand an Elektronik ist hier aber auch betraechtlich.

Multicarrier-Verfahren

Die Multicarrier-Technik stammt aus der militaerischen Anwendung, wo man die Daten verschluesseln wollte. Dabei werden im Frequenzband des Telefons bis zu 500 Traegerfrequenzen aufgebaut. Diese vielen bitparallelen Traeger ermoeglichen auch eine dynamische Anpassung an den Zustand der Leitung - gestoerte Frequenzen werden ausgeblendet, d.h. nicht verwendet. Die vielen Traeger ermoeglichen Datenraten bis zu 19200 BPS. Das bekannteste Verfahren dieser Art ist PEP (Packetized Ensemble Protocol), das aber noch in keiner Norm festgeschrieben ist. Daher existieren derzeit auch noch mehrere Varianten.

Datex-P und Datex-L

Bei Verwendung der Telefonleitung haengen die Kosten lediglich von den anfallenden Telefongebuehren ab, die bei Fernverbindungen schnell eine betraechtliche Groesse erreichen koennen. Die Post bietet zusaetzliche Dienste an, bei denen die Gebuehren teilweise nach dem Datenvolumen, d.h. nach der Menge der uebertragenen Daten berechnet werden, was bei Fernverbindungen wesentlich guenstiger ist.

Datex-P

Basis fuer diesen Dienst sind die CCITT-Empfehlungen der Serie X.25. Das Prinzip sieht hier die Uebertragung der Daten in Bloecken, sogenannten "Paketen" vor. Diese Paket- oder Speichervermittlung stuetzt sich auf die Bereitstellung von "virtuellen" Verbindungen: die Teilnehmer sind hier nicht direkt ueber eine Telefonverbindung miteinander gekoppelt, sondern tauschen ihre Informationen paketweise ueber das gesamte Netz aus. Dabei kann ein interner Uebertragungsweg durchaus von mehreren Benutzern verwendet werden, da die Daten-Pakete jedem Empfaenger eindeutig zugeordnet werden koennen. Andererseits kann es vorkommen, dass aufenanderfolgende Pakete eines Benutzers auf unterschiedlichen Wegen zum Ziel gelangen. Die Datenpakete werden innerhalb des Netzes in jedem Vermittlungsknoten zwischengespeichert, auf Fehler geprueft (dann eventuell nochmals angefordert) und zum naechsten Knoten weitergeleitet. Zwischen den Vermittlungsknoten laeuft die Uebertragung mit 64000 BPS, also wesentlich schneller, als zwischen Modem und der Vermittlungsstelle. Daher erfolgt im Datex-P-Knoten auch eine Protokollanpassung, was den Vorteil bietet, dass auch Modems unterschiedlicher Bauart miteinander kommunizieren koennen.

Datex-P bietet asynchrone Uebertragung mit 300, 1200 und 2400 BPS und synchrone Uebertragung mit 2400, 4800, 9600, 48000 und 64000 BPS. Neben den virtuellen Wahl-Verbindungen sind auch permanetne virtuelle
Verbindungen moeglich, die wie eine Standleitung benutzt werden koennen. Datex-P eignet sich besonders fuer kurze Dialoge (Datenbankabfragen, Buchungsvorgaenge, etc.) ueber weltweit 150 Datennetze in 80 Laendern. Die Gebuehrenstruktur von Datex-P ist recht kompliziert; die Kosten setzen sich aus Verbindungszeit, Entfernung und uebertragener Datenmenge zusammen. In Deutschland sind Vermittlungsstellen in allen groesseren Staedten.

Bei Datex-P erhaelt jeder Teilnehmer eine Benutzerkennung wobei zwischen reinen Informationsanbietern (z. B. Mailboxen) und "normalen" Benutzern unterschieden wird. Der Anbieter erhaelt einen Datex-P-Hauptanschluss mit einer eigenen Datex-P-Telefonnummer (NUA = Network User Address). Ueber diesen Hauptanschluss ist nur Datenaustausch moeglich. Der Teilnehmer beantragt bei der Telekom eine Benutzerkennung (NUI = Network User Identification), die es dem Vermittlungsrechner gestattet, die Gebuehren abzurechnen. Es sind aber auch R-Gespraeche moeglich, bei denen der Angerufene die Kosten uebernimmt. Zur Aufnahme der Verbindung wird der naechstgelegene Vermittlungsknoten angerufen und dann die NUI eingegeben. Danach kann man dann den gewuenschten Partner angeben und die Verbindung herstellen lassen. Danach laeuft dann alles wie vorher mit der Modem-Verbindung ueber die normale Telefonleitung.

Datex-L

Das "L" steht fuer "Leitungsvermittlung", d.h. statt der virtuellen Verbindung von Datex-P besteht hier fuer die gesamte Dauer der Uebertragung eine physikalische Verbindung - wie beim Telefonieren. Daher koennen hier nur Modems mit der gleichen Datenrate miteinander kommunizieren. Datex-L bietet asynchrone Verbindungen mit 300 BPS und synchrone Verbindungen mit 2400, 4800, 9600 und 64000 BPS. Es ist hauptsaechlich fuer den nationalen Gebrauch verwendbar, da bisher nur 9 auslaendische Netze angeschlossen sind. Datex-L eignet sich wegen des raschen Verbindungsaufbaus und der hohen Uebertragungsguete besonders fuer grosse Datenmengen. Die Gebuehren haengen von Verbindungsdauer, Datenrate, Entfernung, Tages- zeit und Wochentag ab. Als besondere Features werden automatische Wahl, Kurzwahl, Direktruf, Gebuehrenuebernahme durch den Angerufenen und vieles mehr geboten.

Datenkompression

Durch Stoerungen auf der Telefonleitung gibt es ab und zu Uebertragungsfehler. Das "Microcom Networking Protokoll" (MNP) der Firma Microcom ist ein Fehlerkorrekturverfahren, mit dem auch bei gestoerter Leitung 100% fehlerfreie Uebertragung moeglich ist - sofern beide Seiten das Protokoll beherrschen. Es gibt mehrere Klassen dieses Protokolls, wobei MNP4 und MNP5 die wichtigsten sind. Fuer den europaeischen Bereich gibt es die Protokolle nach V.42 und V.42bis.

MNP4 sorgt, aehnlich wie die zuvor erwaehnten Protokolle, fuer fehlerfreie Uebertragung, jedoch auf Hardwareebene. Das funktioniert natuerlich nur, wenn beide miteinander verbundene Modems diese Moeglichkeit beherrschen. Damit die Datenrate sich nicht durch die zusaetzliche Pruefinformation verringert, wird die asynchrone Uebertragung in eine synchrone umgesetzt, wodurch Start- und Stopbits wegfallen.

MNP5 kann zusaetzlich die Daten komprimieren, sodass die Uebertragungszeit kuerzer wird (1,3- bis 2-mal so schnell). Wenn die Daten schon in komprimierter Form vorliegen, hilft das natuerlich nicht viel. Zusaetzlich wird der Bitstrom laufend analysiert und bei guter Uebertragungsqualitaet die Blocklaenge verlaengert (Senken des Overhead durch die Pruefinformation). Bei schlechten Leitungen wird die Blocklaenge verkuerzt, damit bei Wiederholungen die Zeit fuer einen Block kuerzer ist.

V.42 entspricht in seiner Leistung dem MNP4-Protokoll, wobei dieser Standard sogar MNP4-kompatibel ist. V.42 hat jedoch ein sein eigenens, besseres Protokoll - LAPM (Link Access Procedure for Modems). Wie bei MNP4 werden auch hier die fehlerhaft uebertragenen Datenbloecke wiederholt.

V.42bis ist der Datenkompressions-Standard des CCITT und er liefert eine um ca. 35% hoehere Kompressionsrate als MNP5. Ein V.42bis-Modem kann zudem erkennen, ob die Daten bereits in komprimierte Form vorliegen (in den meisten Mailboxen sind die Daten bereits "gepackt" verfuegbar), und fuehrt die Kompression nur bei solchen Daten durch, die auch komprimiert werden koennen. V.42bis setzt die Fehlererkennung von V.42 voraus. Das Verfahren ist nicht MNP5-kompatibel, kann aber die Fehlererkennung von MNP4 verarbeiten.

Die V.24- und RS-232-C-Schnittstelle

Die V.24-Schnittstelle, der die amerikanische Schnittstelle RS-232-C weitgehend entspricht (bei V.24 sind einige Dinge genormt, die bei RS-232-C nicht festgelegt sind), ist eine asynchrone, serielle Schnittstelle. Die einzelnen Zeichen werden als Folge von Einzelbits uebertragen, die von einem Startbit mit Nullpegel angefuehrt wird und von einem oder zwei Stopbits mit Einspegel abgeschlossen wird. Sinn und Zweck dieser Norm ist es, eine Basis fuer die Datenuebertragung zu schaffen.

Neben der Masseleitung und den Datenleitungen gibt es noch eine ganze Reihe von Leitungen, die den Verkehr zwischen Rechner und E/A-Geraet (Drucker, Plotter oder DFUe-Modem) steuern. Die Leitungen mit ihren Bezeichnungen sind in der Tabelle unten aufgelistet. Meist interessieren aber nur einige Leitungen um den Verkehr zwischen Computer und Peripherie oder zwischen zwei Computern aufrechtzuerhalten; Die anderen Leitungen bleiben unbeschaltet oder werden auf einen festen Pegel gelegt. Die Pegel bei der V.24-Schnittstelle sind -3...-15 V fuer logisch 1 und +3...+15 V fuer logisch 0 bei den Datenleitungen und +3...+15 V fuer logisch 1 und -3...-15 V fuer logisch 0 bei den Steuerleitungen.

Die wichtigsten Leitungen:

E2 bildet das gemeinsame Massepotential fuer die Datenleitungen.
D1 fuehrt die Sendedaten des Computers zum Modem.
D2 liefert die Daten vom Modem zum Rechner.
S2 gibt dem Modem bekannt, dass der Computer zur Datenuebertragung bereit ist.
M2 signalisiert die Bereitschaft des Modems, Daten zu empfangen. Manchmal wird fuer diesen Zweck auch die Leitung M1 verwendet (z.B. bei Druckern).

Damit waeren die fuenf wichtigsten Leitungen aufgefuehrt. Oft sind noch die Leitungen M1, S1 und M5 belegt, die dann meist auf die entsprechenden Anschluesse des Schnittstellenbausteins fuehren. Normalerweise kann man diese Leitungen in der Buchse verbinden, damit S1 die richtigen Pegel fuer M1 und M5 liefert. Mit den Leitungen D1, D2, M2 und S2 laesst sich ein Handshake-Verkehr realisieren: Ist der Empfaenger (z. B. Drucker) bereit,

Daten zu verarbeiten, legt er M2 auf 1, danach kann der Computer ein Zeichen absenden. Umgekehrt zeigt der Computer durch eine 1 auf der Leitung S2 an, dass er bereit ist, Daten zu empfangen.

V.24-Schnittstellensignale

Kurzzeichen CCITT	DIN	Stecker-belegung	Beschreibung	Richtung Modem Terminal
101	E 1	1	Schutzerde	o—o
102	E 2	7	Signalerde	o—o
103	D 1	2	Sendedaten (TD)	<—o
104	D 2	3	Empfangsdaten (RD)	o—>
105	S 2	4	Sendeteil einschalten (RTS)	<—o
106	M 2	5	Sendebereitschaft (CTS)	o—>
107	M 1	6	Betriebsbereitschaft (DSR)	o—>
108.1	S 1.1	20	Uebertragungsleitung einsch.	<—o
108.2	S 1.2	20	Terminal bereit (DTR)	<—o
109	M 5	8	Empfangspegel (DCD)	o—>
110	M 6	21	Empfangsguete	o—>
111	S 4	23	Uebertragungsgeschwindigkeit (Wahl vom Terminal)	<—o
112	M 4	23	Uebertragungsgeschwindigkeit (Wahl vom Modem)	o—>
125	M 3	22	Ankommender Ruf (RI)	o—>
113	T 1	24	Sendetakt vom Terminal	<—o
114	T 2	15	Sendetakt vom Modem	o—>
115	T 4	17	Empfangstakt	o—>
118	HD 1	14	Sendedaten Rueckkanal	<—o
119	HD 2	16	Empfangsdaten Rueckkanal	o—>
120	HS 2	19	Rueckkanal Sender einschalten	<—o
121	HM 2	13	Rueckkanal Sendebereitschaft	o—>
122	HM 5	12	Rueckkanal Empfangspegel	o—>

Standard-PC-RS-232 Modemkabel (25-Pin)

```
     (PC)                         (Modem)

     DTE                            DCE
    +---+                          +---+
GND | 1 | <----------------->  | 1  | GND
TxD | 2 | ------------------>  | 2  | TxD
RxD | 3 | <----------------->  | 3  | RxD
RTS | 4 | ------------------>  | 4  | RTS
CTS | 5 | <------------------  | 5  | CTS
DSR | 6 | <------------------  | 6  | DSR
GND | 7 | <----------------->  | 7  | GND
DCD | 8 | <------------------  | 8  | DCD
DTR |20 | ------------------>  | 20 | DTR
    +---+                          +---+
```

Standard-PC-DB-9 Modemkabel (9-Pin)

```
     (PC)                         (Modem)

     DTE                            DCE
    +---+                          +---+
GND | 1 | <----------------->  | 1  | GND
TxD | 3 | ------------------>  | 2  | TxD
RxD | 2 | <----------------->  | 3  | RxD
RTS | 7 | ------------------>  | 4  | RTS
CTS | 8 | <------------------  | 5  | CTS
DSR | 6 | <------------------  | 6  | DSR
GND | 5 | <----------------->  | 7  | GND
DCD | 1 | <------------------  | 8  | DCD
DTR | 4 | ------------------>  | 20 | DTR
    +---+                          +---+
```

Kommandos fuer den Modem

Die Steuerung des Modems geschieht nach einem De-Facto-Standard, der von der amerikanischen Firma Hayes entwickelt und dann von anderen Herstellern uebernommen wurde. Vielleicht haben Sie schon von "Hayes-kompatiblen" Modems gehoert. Der Modem ist in der Lage auf bestimmte Befehle zu reagieren, die ueber die serielle Schnittstelle gesendet werden. Diese Befehle beginnen immer mit den beiden Buchstaben "AT" (AT steht fuer "Attention"). Dann folgen die Befehlscodes fuer bestimmte Funktionen, z.B.:

```
ATH0      Hoerer auflegen
ATH1      Hoerer abheben
ATDP123   Die Nummer 123 waehlen
```

Das Waehlen kann uebrigens nicht nur mit dem bei uns gebraeuchlichen Impulsverfahren, sondern auch mit der Tonfrequenzwahl arbeiten. Auf die gleiche Weise lassen sich die internen Steuerregister des Bausteins setzen. So laesst sich der Modem veranlassen, bei einem Anruf das Gespraech automatisch entgegenzunehmen. Man kann so einen unbedienten Computer als Mailbox laufen lassen, der Informationen eines Anrufers speichert, die dann zu einem spaeteren Zeitpunkt von einem anderen Anrufer wieder "abgeholt" werden kann. Die Verbindungsaufnahme geschieht normalerweise durch die Anwahl der Gegenstation, z. B. mit

```
ATDP12345678
```

Der Modem waehlt die Nummer und die Gegenstation nimmt das Gespraech automatisch an. Bei Modems mit hoeheren Uebertragungsraten und automatischer Anpassung der Bitrate tauschen beide Stationen nun verschiedene Tonsignale aus, die dazu dienen, die Datenrate abzustimmen. Ueber den Lautsprecher des Modems koennen Sie den Verbindungsaufbau verfolgen. Danach meldet der Modem "CONNECT" und der Datenaustausch kann beginnen.

Damit waehrend der Datenuebertragung ein zufaelliges Auftreten der Buchstabenfolge "AT" nicht als Befehl an den Modem interpretiert wird, schaltet sich die Kommandoerkennung nach Aufbau der Verbindung ab. Erst wenn nach einer Pause von mindestens 1 Sekunde die Zeichenfolge "+++" an den Modem gesendet wird und anschliessend eine weitere Pause von 1 Sekunde folgt, werden AT-Kommandos wieder erkannt. Nach dem Uebertragen der Kommandos kann die Kommandoerkennung mit "ATO" wieder abgeschaltet werden.

Die Leitung wird vom Modem staendig ueberwacht und bei Stoerungen (oder wenn der Partner einfach auflegt) die Verbindung automatisch unterbrochen.

Fachbegriffe

ASCII
Abkuerzung fuer "American Standard Code for Information Interchange". Gelaeufiger Code fuer die Datenuebertragung.

AT-Befehlssatz
Von der Firma Hayes eingefuehrter und zu einem Standard gewordener Befehlssatz fuer Modems.

Autodial
Automatisches Waehlen eines Modems (im Gegensatz zum manuellen Waehlen).

Baudrate
Anzahl der Signalwechsel in Einheiten pro Sekunde (Fernschreibtechnik). Nicht verwechseln mit Bitrate (in der Regel nicht identisch mit Baudrate).

bps
Abkuerzung fuer "Bits per Second". Masseinheit fuer die Anzahl der uebertragenen Datenbits pro Sekunde (in der Regel nicht identisch mit der Baudrate).

BBS
Abkuerzung fuer "Bulletin Board System" - Mailbox(-Programm).

BZT
(Bundesamt fuer Zulassungen in der Telekommunikation) Neuer Name des -> ZZF

Carrier
Analoges Signal mit konstanter Frequenz und Amplitude (Traegersignal)

CCITT
Unterorganisation der Vereinten Nationen, in der Fernmeldeunternehmen aller Welt vertreten sind (Comite Consultatif International Telegraphique et Telephonique)

CEPT
Europaeische Vereinigung der Postverwaltungen (Conference Europeene des Administrations des Postes et Telecommunications)

CRC
Abkuerzung fuer "Cyclic Redundancy Check". Verfahren zur Fehlerkorrektur.

Datex-P
Datenuebertragungsnetz mit Datenpaket-Vermittlung, das von der Deutschen Bundespost Telekom betrieben wird. Der Datenaustausch an der Netzschnittstelle erfolgt nach der Empfehlung X.25.

Download
Datenempfang von einem anderen Computer.

DTMF-Wahl
(Dial Tone Multi Frequency) Mehrfrequenz-Tonwahl

Duplex
Die Datenuebertragung kann gleichzeitig in beide Richtungen erfolgen.

FCC
(Federal Communications Commission) Amerikanische-Fernmelde-Aufsichtsbehoerde

FSK
(Frequency Shift Keying) Datenuebertragung durch Frequenzumtastung

FTZ (*)
(Fernmeldetechnisches Zentralamt) Postbehoerde zur Erarbeitung von Vorschriften und Standards fuer die Telekommunikation

(*) Anmerkung der Redaktion: ... seit Dezember 92 offiziell als **Forschungs- und Technologiezentrum** bezeichnet.

ISDN
(Integrated Services Digital Network) Kommunikationsdienst fuer Sprache, Daten, Text, Bild auf 2 64KBit- und 1 16KBit-Kanaelen.

ISO
(International Organisation for Standardiziation) Internationaler Zusammenschluss von Normungsausschuessen.

Kermit
Uebertragungsprotokoll mit Fehlerkorrektur.

Login
Einwahl in ein System, z.B. Mailbox, Computer, etc.

Logoff
Auswahl aus einem System - Verbindung beenden.

Modem
Abkuerzung fuer "Modulator-Demodulator". Eine Datenuebertragungseinrichtung zum Anschluss an das Telefonnetz.

PAD
Abkuerzung fuer "Packet Assembly and Disassembly". Einrichtung zum paketweisen Uebertragen von Daten (DATEX-P).

PABX
(Private Automatic Branch Exchange) Telefon-Nebenstellenanlage

Protokoll
Vereinbarung ueber den Aufbau, die Ueberwachung und den Abbau von Verbindungen.

RS232C
Amerikanischer Standard fuer die serielle Schnittstelle bis 19200 Bit/s. -> V.24

Simplex
Datenuebertragung jeweils nur in einer Richtung wechselweise).

Synchrone Uebertragung
Datenbits werden in exakten Zeitintervallen vom Sender zum Empfaenger uebertragen. Dadurch sind keine Start- und Stopbits notwendig. Die synchrone Uebertragung ist schneller und effizienter als die asynchrone Uebertragung.

TEMEX
(Telemetry Exchange) Moeglichkeiten von Fernmessen, Fernwirken, Fernsteuern ueber vorhandenen Telefonleitungen.

V.24
Definition der Schnittstellenleitungen zwischen Datenendeinrichtungen (DTE) und Datenuebertragungseinrichtungen (DCE); vom CCITT festgelegt worden.

ZZF
(Zentralamt fuer Zulassungen im Fernmeldewesen) Zulassung von Telekommunikationseinrichtungen. Neuer Name: BZT

Ist man aktiv in der DFÜ, wird man oft gefragt, was das FIDO-Netz nun eigentlich sei. Eine kurze Antwort wäre: "Ein internationaler, privater Rechnerverbund", doch ist das zu ungenau. FIDO ist nicht nur der weltweit größte privat betriebene, geordnete Zusammenschluß von Computern, sondern wesentlich mehr: Poststelle, Flohmarkt, Beratungsstation, Treffpunkt oder gar Kummerkasten.

DFÜ

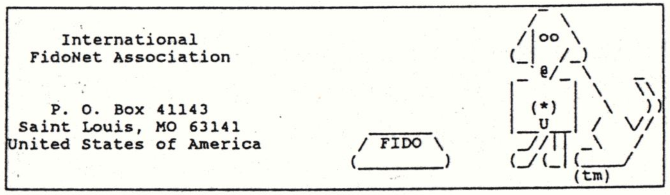
MAILBOX-NETZE IN DEUTSCHLAND

Das FIDO-Netz

Wenn man so etwas noch nie live erlebt hat, fällt es auch nach ausführlichster Schilderung durch begeisterte DFÜ-User oft schwer, es nachzuvollziehen - zumal ja im Volksmund diejenigen, die etwas mit Computern und Datenfernübertragung zu tun haben, oft als "Hacker" verschrien sind - eine Verallgemeinerung ohne jede Grundlage. Deshalb ein konkretes, eigentlich für jeden nachzuvollziehendes Beispiel aus der Praxis: nachdem mein alter Wagen seinen Geist aufgab und ich mich von ihm trennen mußte, blieben mir von ihm noch einige Teile übrig, die ich verkaufen wollte. Eine Möglichkeit nun, sie anzupreisen, bestand in einer nicht billigen Kleinanzeige in der Samstagsausgabe der örtlich beschränkten Lokalzeitung. Die, für mich aufgrund ihrer bundesweiten und und zudem kostenlosen Verbreitung, jedoch wesentlich interessantere Alternative war die DFÜ. Über meinen PC, der per Terminal-Software und Modem mit dem Telefonnetz verbunden ist, loggte ich mich also in eine Mailbox des FIDO-Netzes ein, die im wesentlichen ein per Modem an das Telefonnetz angeschlossener Computer mit geeigneter (Mailbox-) Software ist. Im Nachrichtenbereich der Box wählte ich die Area "Flohmarkt" an, und tippte - bequem zuhause über meine PC-Tastatur - meine Verkaufsanzeige für die Autoteile ein. Diesen Text speicherte dann auf Tastendruck die Mailbox, und ich loggte mich wieder aus. Meine Anzeige nun machte die Runde in fast allen bundesdeutschen Mailboxen, die dem FIDO-Netzverbund angeschlossen sind - weltweit sind es zur Zeit etwa 12000 Mailboxen mit insgesamt wohl drei bis vier Millionen Mitgliedern. Ich hatte in der Anzeige sowohl meine Telefonnummer als auch die spezielle FIDO-Nummer der Mailbox angegeben, in der ich die Nachricht aufgab. Zwei Tage später rief mich jemand aus Ulm an und kaufte alle Teile en bloc. Ein Verkaufserfolg, der mich lediglich 90 Pfennig Telefongebühr kostete - anstelle bis zu DM 70.- für eine herkömmliche Kleinanzeige in der Zeitung.

Verkaufsanzeige per FIDO-Netz - kein Problem

Eine andere Gelegenheit ergab sich, als ich etwas suchte - sowohl eine seltene Software als auch etwas Rat dazu. In der Mailbox wählte ich im Nachrichten-Bereich die Area "Request" (= engl.: Nachfrage, Gesuch) an, und tippte meine Suchanzeige. Da es zu dieser Art Software auch eine spezielle Area mit allerlei Mitteilungen, eben Tips + Tricks, gibt, wählte ich diese Area auch noch an, und tippte hier meine Fragen zur Software ein. Während mein Kaufgesuch wieder sowohl mit meiner Telefonnummer als auch der FIDO-Nummer der Mailbox versehen war (drei Tage später erhielt ich einen Anruf mit einem preiswerten Angebot und kaufte), hatte ich meinen Text mit den Fragen zum Programm nur mit der FIDO-Nummer der Mailbox versehen. Zwei Wochen lang trudelten nun in schöner Regelmäßigkeit Antworten auf meine Fragen ein, gesandt von zahlreichen FIDO-Mitgliedern. Die Antworten wurden mir immer angezeigt, sobald ich mich in die Mailbox einloggte, - zwei kamen sogar aus Übersee: eine aus Washington, DC (USA), die andere aus Südafrika - den Ort weiß ich nicht mehr. Ich hatte so - von zuhause aus - nicht nur ein Programm gesucht, gefunden und gekauft, sondern gleich auch noch viele wertvolle Tips für den Umgang damit fast aus aller Welt erhalten. So etwas schafft keine Anzeige in einer Zeitung, welcher Art auch immer.



Das alles liest sich nun sehr einfach (was es eigentlich auch ist) und verheißt einen schnellen Weg, mit vielen Leuten in Kontakt zu kommen - was auch stimmt. Doch der Weg zum heutigen FIDO-Netz war nicht ganz so einfach. 1984 kamen die Amerikaner Tom Jennings in Baltimore und John Madill in San Francisco auf die Idee, Daten zwischen ihren beiden DEC-Rechnern per Telefonleitung und Modem auszutauschen. Innerhalb von Wochen wurde das die lästige Handwahl ablösende Mailbox-Programm, das die beiden erstellt hatten, immer weiter ausgebaut und fand zahlreiche Interessenten, so daß das FIDO-Netz dann, ebenfalls noch 1984, gegründet wurde. Zweck war, ein privates Netz zwischen mehreren Rechnern auf der Basis der Datenfernübertragung zu schaffen, mit dem beliebige, an das Netz angeschlossene Mitglieder der Programmdaten und vor allem Nachrichten bzw. Mitteilungen austauschen können. Der ehemalige APPLE-Systemprogrammierer Jennings installierte den ersten FIDO-NODE auf einer DEC Rainbow, einem Rechner der mittleren Datenverarbeitung. Bei der Namensfindung für das Netz war ihm ein weiterer guter Freund behilflich: auf den Namen "Fido" hörte sein Hund. Zwei Monate nach der offiziellen Gründung war das noch junge FIDO-Net schon auf über 50 Rechner mit Standorten überall in den USA angewachsen.

Gleich nach Gründung wuchs das Netz

Ein böser Spruch sagt: "... wenn drei Kaufleute zusammentreffen, gründen sie einen Verein und sind Vorstand, Kassierer und Protokollführer ..." und nimmt damit das Verwaltungswesen aufs Korn, das überall gut (oder schlecht, je nachdem) entwickelt ist, wo viele Menschen zusammentreffen. Ein solches Verwaltungssystem wurde aber dringend notwendig, als Tom Jennings und seine Freunde feststellten, daß aus ihrem recht losen Haufen DFÜ-Enthusiasten in den USA plötzlich eine große Schar Anwender geworden war. Der Name FIDO war bekannt im Land, und viele Leute ärgerten sich, wenn nachts das Telefon klingelte, sie schlaftrunken dran gingen - und außer dem Piepsen eines Modems nichts hörten.

Ein User hatte sich verwählt, oder gar eine Nummer angewählt, die vor Wochen mal zu einer Mailbox gehörte - der SysOp (und die Box) waren aber umgezogen - und die Nummer gehörte nun anderen, in die Wohnung als Nachmieter eingezogenen Leuten. Als sich die Beschwerden von "Nicht-Usern" häuften, die mit Modem-Gepiepse aus dem Schlaf gerissen wurden, begannen die FIDO-Gründer über eine Verwaltung des noch jungen FIDO-Netzes nachzudenken, die zudem dezentral und demokratisch arbeiten sollte.

Erste Regel wurde, daß, wer sich jetzt am FIDO beteiligen wollte, zunächst die Software von Tom Jennings auf seinem Rechner zum Laufen bringen mußte. Funktionierte das, mußte per DFÜ eine Mitteilung an ihn abgesandt werden, woraufhin er dem neuen User die FIDO-Node-Nummer, auch Netz-Adresse genannt, zusandte, die letztendlich erst die Teilnahme am aus Mailboxen bestehenden Netzbetrieb ermöglichte - ganz ähnlich übrigens, wie das überall auf der Welt mit den Telefonen geschieht: auch hier stellt man einen Antrag, bekommt eine Leitung gelegt, einen Telefonapparat und schließlich die Telefonnummer, unter der man angerufen werden kann.

Das FIDO-Netz-System ähnelt dem Telefon-System

Was als kleiner Verbund interessierter Freaks begann, weitete sich rasend schnell weltweit aus - schneller, als man schauen konnte. Und ebenso schnell wuchs dann notgedrungen auch das Verwaltungssystem, das bis dahin ja nur auf die USA beschränkt war. Um nun Daten aller Art weltweit versenden zu können, richtete man "Zonen", "Regionen", "Netze" und "Hubs" ein, wobei letztere Mailboxen, also "Nodes" beliefern. Dazu wurde auch ein spezielles, von jedem FIDO-Mailbox-Programm verwendbares Adressierungssystem eingeführt, das für jeden Bereich, also z.B. Zone, Region, Netz oder Node eine Nummer vergibt. Das Ganze ähnelt stark dem weltweiten Telefonsystem. Folgende Zonen mit folgenden "Adress-Nummern" gibt es heute:

1 für Nordamerika
2 für Europa usw.
3 für Australien und Taiwan
4 für Südamerika
5 für Afrika
6 für Asien

Zonen, Regionen, Netze usw. stellen zum einen eine Verwaltungshierarchie da, zum anderen handelt es sich dabei aber auch um Computer, die die Daten per Telefonnetz übertragen. So erhält eine "Zone" Daten von anderen Zonen und leitet diese, je nach Adresse, an ihre eigenen Regionen weiter, die zumeist identisch sind mit politischen Ländern. Eine

"Region" leitet weiter an mehrere, nur ihr direkt zugehörende "Netze", die wiederum an mehrere, ihnen unterstellte "Hubs" (Verteiler-Stationen) senden. Ein "Hub" schickt die entsprechenden Daten dann an "seine" "Nodes" (Mailboxen). Im Klartext heißt das: wenn nun eine Mitteilung, aufgegeben z.B. vom User einer Box, von dieser versandt wird, geht die Mitteilung erst an den zuständigen Hub und dann an das zuständige Netz. Das Netz sendet die Mitteilung dann, je nach Adresse, an einen seiner Hubs oder an ein anderes Netz (unter Umständen über die Region an die Zone und von da aus in eine andere Zone, andere Region), das - wieder je nach Adresse - an einen Hub weitergibt, der

DFÜ

wiederum den Empfänger-Node beliefert. Das Ganze funktioniert nicht nur innerhalb eines Netzes oder einer Region, sondern - dem Schema entsprechend - auch von Zone zu Zone, also weltweit. Der Absender erhält dann vom Empfänger - z.B. einem der empfangenden Box angehörenden User - auf demselben Wege kurzfristig eine Antwort, falls nötig. Der Absender sitzt z.B. in Österreich, der Empfänger in Taiwan. Beide stehen so in ständigem "Mitteilungs-Kontakt" - können sich aber auch andere Daten, z.B. Programme oder Datendateien schicken. Da solche Kontakte rund um den Globus riesige Entfernungen zurücklegen, könnte man denken, daß das die beiden User viel Telefongebühren kostet - mitnichten, da die Mails und Dateien über das Netz gehen - und sich so die Kosten aufteilen.

Die Menschen hinter den "Zonen", "Regionen", "Netzen" und "Hubs" nennt man "Koordinatoren", die von der FIDO-Gemeinde demokratisch gewählt werden. So gibt es "ZC's" (Zone-Koordinatoren), "RC's" (Regional-Koordinatoren), "NC's" (Netz-Koordinatoren) und "HC's" (Hub-Koordinatoren), die nicht nur einige Kosten haben, sondern durch die ihnen übertragenen Verwaltungsaufgaben (z.B. Aufnahme neuer Regionen, neuer Netze, neuer Nodes, Zuordnung von Adressen, Durchführung von Wahlen, Belieferung der ihnen untergeordneten Stellen mit Daten aller Art, Schlichtung von Streitigkeiten usw.) auch Verantwortung tragen und so ziemlich alle Freizeit opfern müssen. Als normaler User kennt man lediglich den SysOp der nächstgelegenen Box näher.

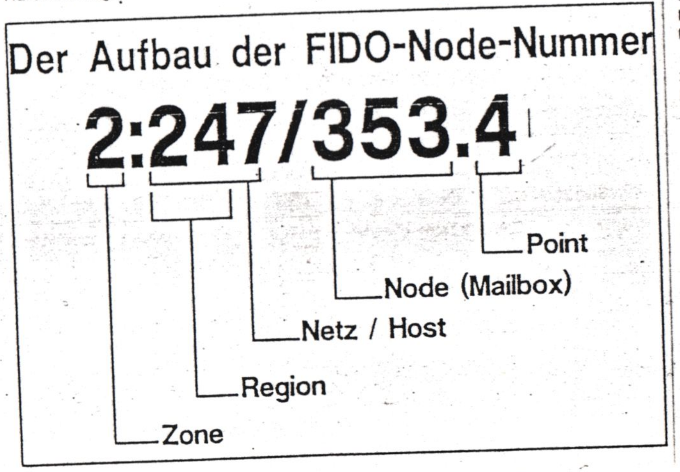
Das Vorgenannte war eine schematische Beschreibung der FIDO-Struktur, um zunächst das "Versand-System" darzustellen. Jedoch ist das Ganze noch wesentlich verwobener, als es sich hier liest: ein ZC zum Beispiel kann neben seiner Tätigkeit als "Zone" und deren Koordinator selbst durchaus auch Node sein - gleiches gilt für RC's, NC's und Hub's. Somit kann ein User durchaus Mitglied der Box eines ZC sein - was durchaus (zeitliche) Vorteile hat, sitzt der User doch direkt am Verteiler dran und bekommt Mails am schnellsten. Dieses - für den Laien - augenscheinliche Durcheinander kann nur deshalb funktionieren, weil es eben diese starre, aber für solche Zwecke hervorragend geeignete Verwaltungs- und Vermittlungsstruktur (siehe Abbildung) innerhalb des FIDO-Netzes gibt. Und es funktioniert auch deshalb, weil diejenigen im FIDO, die Verantwortung haben, sich zumeist streng an die Struktur und die dafür auch notwendige Disziplin halten.

FIDO - größtes privates Computer-Netz der Welt

Durch diesen Netzverbund und das darin herrschende "Daten-Verteilungssystem" wird ermöglicht, daß sich die Kosten für das, ja ausschließlich privat betriebene FIDO-Netz in Grenzen halten: denn jeder FIDO-Offizielle, beginnend beim Node bis hin zum ZC oder gar hinauf zum IC (Internationaler Koordinator - der eigentliche FIDO-Oberboß, Chef aller ZC's, NC's usw.) betreibt seine Tätigkeit als Hobby. Und da das Hobby auf teure Hardware angewiesen ist, und zudem mit dem ebenfalls nicht kostenlos nutzbaren Telefonnetz arbeitet, gehört die aktive Mitarbeit an einem privaten Computer-Netz zu den teureren Freizeitbeschäftigungen, was außer der Geldfrage manchmal auch Partnerschaften und Familien ins Wanken brachte - und bringt.

Die Datenübermittlung funktioniert nach dem genannten Schema, dem eine Adressierung, die sogenannte "Node-Number" (Mailbox-Nummer) zugrundeliegt, die verwaltungstechnisch auf gleicher Stufe mit einer Telefonnummer steht. Eine typische Node-Nummer ist z.B. 2:247/353. Sie beschreibt die Zone 2 für Europa, die Region 24 ,für Deutschland, das Netz (auch "Host") 247 für das "Hessen-Net" (kann aus einer oder mehr Ziffern bestehen) und 353 für den Node (Mailbox, ebenfalls eine oder mehr Ziffern) (siehe Abbildung). Die Software eines jeden Übermittlungsrechners, egal, ob Zone, Region, Netz oder Hub, erkennt anhand dieser Ziffern- und Zeichenfolge aufgrund der, ähnlich einem Telefonbuch, vorliegenden Node-Liste, wohin eine Nachricht kommen muß. Diese Node-Liste enthält die weltweite hierarchische Struktur des FIDO-Netzes in Form dieser Adressierung. In ihr sind neben der jeweiligen FIDO-Adresse auch die Telefonnummern und Namen der SysOps und ihrer Boxen sowie der ZC's, NC's usw. enthalten. Die Liste wird wöchentlich "von ganz oben" aktualisiert und ist für Dritte und auch normale Box-User nicht zugänglich.

Soweit zur Technik. Was bietet FIDO nun den Anwendern, den Usern, die sich in die Mailboxen des Netzes einloggen? Zunächst einmal recht strenge Regeln, die sogenannte "Policy", die nicht unbedingt vorschreibt, jedoch klar festmacht, was im FIDO erlaubt ist, und was nicht. Man soll freundlich, sachlich und hilfsbereit miteinander umgehen. Und man darf das Netz weder kommerziell, noch politisch oder institutionell ausnutzen. Auf all diese Punkte wird rigoros geachtet - und User, die sich regelmäßig daneben benehmen, oder aber in ungewünschter Richtung tätig werden, sind schnell "persona non grata" - und fliegen aus FIDO raus.



FIDO und die Mailboxen verfügen bezüglich der Nachrichten/Mitteilungen (Mails) oder Software (PD und Shareware) über sogenannte "Area's" (Rubriken oder auch Bretter). Diese Areas sind in jeder FIDO-Mailbox weltweit vorhanden und können von allen Usern gelesen und beschrieben werden. Im Mail-Bereich kann man im Rahmen des bereits beschriebenen Mitteilungsempfangs und -versands so ziemlich über alles diskutieren, was es an Themen gibt: Computer, Autos, Kochen, Medizin, Recht usw. - für jeden Bereich, jedes Thema eine Area. Der eingangs beschriebene An- und Verkauf wird beispielsweise in der "FLEA.GER", der Flohmarkt-Area, abgewickelt. Fast jeder Area steht ein demokratisch von den Area-Nutzern gewählter "Anstandswauwau", der "Moderator" vor, der die Einhaltung der FIDO-Policy und eventueller, Area-spezifischer Regeln überwacht und auch die Anweisung für den Ausschluß eines Users aus der Area wegen Regelverstoßes geben kann. Diese Anweisung wird an alle Mailboxen weitergegeben, die den "randalierenden" User dann für diese Area sperren.

Es gibt verschiedene Mitteilungsarten, zum Beispiel die "NetMail" und die "EchoMail". Die NetMail sendet man direkt über die Netzadresse (Node-Number) an einen Anwender in einer Box, während EchoMails öffentliche Mitteilungen in den Areas sind, die von jedermann in jeder Mailbox gelesen werden können und auch an jede Box verteilt werden. Gerade die EchoMails tragen zur Anregung der vielfältigen Diskussionen und Hilfestellungen bei, für die das FIDO-Netz mittlerweile bekannt ist. Das System der NetMail, also der "direkten Nachricht" funktioniert wie folgt: in der Mailbox (oder in einer speziellen Software zur Mailbearbeitung) wählt man die Nachrichteneingabe an. Im dazu notwendigen Editor erscheint in einem Adress-Feld bereits der eigene Name. Man gibt den Namen des Empfängers und die Node-Number seiner "Haus+Hof-Mailbox" an, über die man ihn erreichen kann. Die mit dieser Adressierung versehene Mail kommt dann über das Netz in der Empfänger-Mailbox an. Loggt der Empfänger sich in die Box ein, was namentlich geschehen muß, werden ihm sämtliche für ihn bestimmten Mitteilungen, also öffentliche, jedoch namentlich an ihn gerichtete EchoMails und die privaten NetMails, sofort nach dem Ein-log-Vorgang angezeigt.

Interessant ist, daß der Mitteilungsverkehr dritterseits völlig unzensiert ist - FIDO ist privat, und man achtet auch strikt darauf, daß niemand diesen Status stört. Das Mitteilungslesen und -schreiben kann man entweder online erledigen, oder aber, was wesentlich kostengünstiger ist, die Nachrichten offline bearbeiten. Zwei Wege führen dazu: entweder bedient man sich eines sogenannten Offline-Mail-Readers, einer Software, die von der Box gesandte "Mail-Packets" verarbeitet, oder aber man wird als User zum "Point" - einer Art kleine Mailbox innerhalb einer FIDO-Box. Die Points lesen und schreiben zuhause Mitteilungen mit spezieller Point-Software, ähnlich wie User, die einen Offline-Mail-Reader haben. Zusätzlich sind Points aber eingebunden in die FIDO-Hierarchie mit einer eigenen FIDO-Adresse: der sogenannten Point-Number, die sich aus der schon bekannten Node-Number, also z.B. 2:247/353 und einem Zusatz ".x", also etwa 2:247/353.4, zusammensetzt. Für die Points gibt es eine eigene Point-Liste. Die dafür notwendige Point-Software ist aber nicht nur zum Offline-Bearbeiten der Mails gedacht, sondern stellt zudem ein Terminalprogramm dar, mit dem man seine Mail-Pakete bekommt und wegschickt und Shareware aus dem Fundus der Box anfordern kann.

DFÜ

Mitteilungen weltweit sind Hauptzweck von FIDO

FIDO-Boxen bieten ständig Neues schon beim "Guten-Tag-Sagen".

Während bei anderen Netzen meistens eine einzige Mailboxsoftware mit überall einheitlicher Oberfläche Verwendung findet, dürfen die SysOps der FIDO-Boxen ihre Box-Oberflächen frei gestalten. Das bedeutet für datenreisende User innerhalb FIDO so manche Überraschung. Während manche Box zunächst recht wüst aussah und man sich kaum zurecht finden konnte, achten mittlerweile immer mehr SysOps im FIDO darauf, daß ihre Box zwar individuell gestaltet, aber trotzdem übersichtlich und leicht bedienbar bleibt - man arbeitet meist mit den "Hotkeys" - der Ein-Tasten-Bedienung, wo Funktionen, wie z.B. das "Download" et-

wa durch den Druck auf die Taste "D", aktiviert werden können. Dafür haben User im FIDO-Netz den Vorteil, daß alle sich mit ihrem wirklichen Namen einloggen müssen - Alias-Namen jeder Art sind verboten - und ein SysOp bekommt auch schnell heraus, wenn sich jemand mit einem Falschnamen eingeloggt hat. Diese Vorgehensweise hat mit dazu beigetragen, daß das FIDO-Netz relativ sicher ist vor allerlei mehr oder weniger mutwilligem Schabernack oder ungewollter Überwachung.

DAS FIDO-NETZ

GRÖSSTES, PRIVAT BETREUTES COMPUTER-NETZ DER WELT. DAS SIND 12.000 MAILBOXEN MIT CA. 3 BIS 4 MILLIONEN USERN, WELTWEIT!

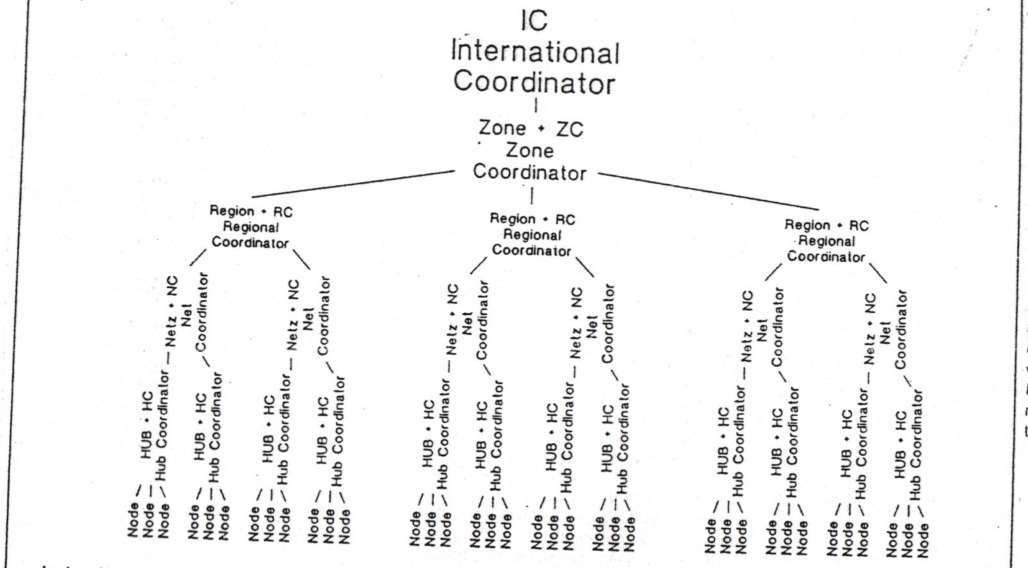
Obwohl die meisten Rechner des FIDO-Netzes unter MS-DOS arbeiten, bedeutet das nicht, daß das Netz ein Synonym für MS-DOS-Rechner ist. Es definiert lediglich die Art und Weise, über Modems und Telefonleitung nach einer bestimmten Vorgehensweise Nachrichten auszutauschen, sowie die interne Nachrichtenstruktur, ohne dabei an eine bestimmte Computer-Hardware gebunden zu sein. Geeignete Terminalsoftware und ein Modem vorausgesetzt, kann man sich in FIDO auch z.B. mittels eines Apple MacIntosh oder eines Commodore Amiga einloggen. Es steht allen offen, die bereit sind, sich mit den Netz-Regeln vertraut zu machen und sich daran zu halten und dient - was viele nicht wissen - in erster Linie dem reinen Nachrichten-Austausch weltweit. Der Vertrieb von PD- und Shareware-Programmen ist mehr oder weniger zweitrangig, wenngleich die PD/Shareware weltweit erst durch die DFÜ (und FIDO) zu ihrem Erfolg gekommen ist. Die Anwender im FIDO-Netz sind ebenso wie in anderen Netzen zumeist ganz normale Leute aus allen Bevölkerungsschichten, die keinerlei Ambitionen haben, sich als "Hacker" in den nächstmöglichen Großcomputer einzukoppeln - das, was im Volksmund oft geäußert wird, mag auf einzelne zutreffen - in keinen Fall jedoch pauschal für alle DFÜ-ler.

Grundsätzlich kann jeder, der ein Modem besitzt und am Austausch von Informationen, Tips und Tricks interessiert ist, Teilnehmer am FIDO-NET werden. Zuerst besteht die Möglichkeit, daß er sich als User in einer Mailbox registrieren läßt, die Mitglied im FIDO ist. Wenn man als User oder Point soviel Erfahrungen und auch Interesse gesammelt hat, daß man sich nun entschließt, nicht nur sich selbst mit interessanten Nachrichten aus dem FIDO zu versorgen, sondern dies auch andern Interessenten aus der eigenen Umgebung zugänglich machen möchte, hat man durchaus die Möglichkeit, selbst eine Mailbox aufzumachen, und die Nachrichten allen potentiellen Usern anzubieten, die vielleicht im eigenen Einzugsgebiet wohnen. Im FIDO-Netz sind die (DOS-) Programme "RemoteAccess" als Mailboxprogramm und "D'Bridge" als Mailer aufgrund ihrer einfachen Bedienung und soliden Programmierung weit verbreitet - jedoch gibt es auch andere, ähnliche Programme auf dem (Shareware-) Markt - auch für andere Hardware als die PC. Als SysOp in spe hat man ja schon Kontakt zur Szene und stellt über einen befreundeten SysOp einen Antrag auf Aufnahme ins FIDO-Netz, der, sofern nichts dagegensteht, schnell und positiv beantwortet wird - man bekommt eine eigene Node-Nummer und die für SysOps üblichen Anweisungen und darf dann, nach Absprache mit der "übergeordneten Stelle", dem Hub, loslegen. Als SysOp sollte man mindestens einmal täglich beim Hub anrufen und Mails pollen (abrufen), damit die der neuen Box hoffentlich recht schnell angehörenden User entsprechend versorgt sind. Und was auch sehr wichtig ist: als SysOp mit Box sollte man sich vorab mit der Familie besprechen - damit unter Umständen nachher nicht das einzige, was einem von beiden bleibt, die Box ist...

FIDO in all seinen Bereichen ausführlich zu beschreiben, würde den Umfang dieser Ausgabe bei weitem sprengen, so daß es nur zu diesem groben Überblick reicht. Tatsache ist, daß das FIDO-Netz einen hervorragenden Service und ein richtiges, eigenes Flair bietet, wenngleich es aufgrund der unterschiedlichen Mailbox-Oberflächen etwas gewöhnungsbedürftig ist und Flexibilität erfordert, aber auch erlaubt. Da FIDO - im Gegensatz zu manch anderem Netz - weniger streng themenorientiert ist und zudem ausschließlich privat organisiert wird, ist es ein ideales Netz für eine DFÜ-Mitgliedschaft - aber bitte immer die monatliche Telefonrechnung im Auge behalten...

Thomas Kallay

Die Verwaltungsstruktur des FIDO-Netzes



Jeder Node hat mehrere Points, die in die FIDO-Hierarchie mit eingebunden sind, und unzählige User

Schluß

Hallo Club 80'er,

"schon" wieder steht ein Jahreswechsel vor der Tür, an dem man üblicherweise über das vergangene Jahr zurückblickt.
Hier in aller Kürze: Dieses Jahr war fast nichts los!

Trotz bester Vorsätze konnte ich Euch in diesem Jahr nur drei Club-Info's bieten. Leider!
Auch die geplante Jahreshauptversammlung fiel aus, wurde aber durch ein spontanes und gelungenes "Südländer-Treffen" ersetzt.

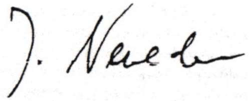
Was ist los mit dem CLUB80 im Jahre 93?

Wichtig für eine bessere Zukunft wäre auf alle Fälle, daß:

- wir erfahren, welche Themen Euch interessieren,
- soviel Mitglieder wie möglich an der Jahreshauptversammlung teilnehmen,
- wir mit Eurer Unterstützung bei den vielfältigen Aufgaben im Club rechnen können.

Ich bin -wie es Hartmut im Vorwort schon angekündigt hat- auch für ein beitragfreies Club-Jahr. Eine Entscheidung darüber sollte man aber anläßlich der künftigen Jahreshauptversammlung finden. Zumindest möchten wir Eure Meinung dazu hören. Sollte man vielleicht ganz aufhören? Oder, um z.B. die Aktualität zu wahren, das INFO streichen und dafür eine (nachdem ein Modem doch recht günstig zu haben ist) clubeigene Mailbox betreiben?
Fragen über Fragen, die wir ohne Eure Mitarbeit nicht beantworten können. Es ist jedenfalls an der Zeit, daß wir uns alle wieder aktiver mit unserem Club befassen.

Mit den besten Wünschen
zu den Feiertagen und zum NEUEN JAHR
bis zum nächsten mal



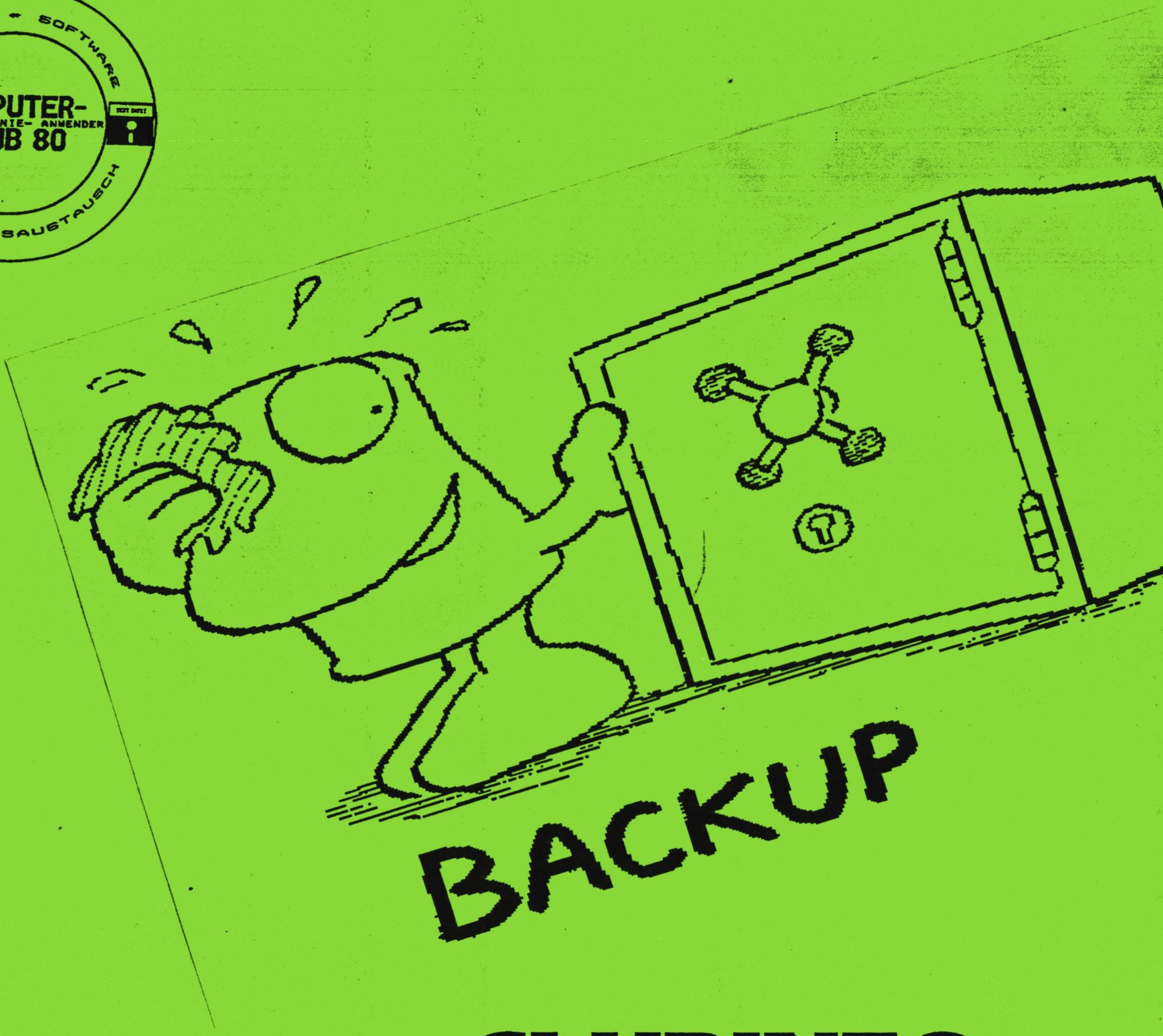
Adressen-Liste Club80

Stand vom 10-01-1993
Seite 1

Nachname	Vorname	Straße	PLZ	Ort	Telefon privat	Telefon geschäftlich
Behrendt	Detlef	Schlosserbreite 1a	W- 8018	Grafing	08092 / 9173	
Benner	Achim	Vorm Mühlberg 1	W- 5910	Kreuztal 3	02732 / 3780	(ab 18.00Uhr)
Berndt	Wolfgang	Friedberger Straße 92c	W- 6360	Friedberg 2	06031 / 2963	
Berndt-Jochum	Ilse	Stachelsgut 24	W- 5060	Bergisch-Gladbach 1	02204 / 65254	
Bernhardt	Helmut	Hafenstraße 7	W- 2305	Heikendorf	0431 / 241907	0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	W- 8551	Hausen	09191 / 31698	09191 / 611108
Brans	Jörg	Tieloh 55	W- 2000	Hamburg 60	040 /6906531	
Braun	Harald	Postfach 8011	W- 2300	Kiel 17		
Braun	Günter W.	Postfach 800 226	W- 8000	München 80		
Böckling	Ulrich	Jochaczstraße 61	W- 5410	Hör-Grenzhausen	0264 / 4861	02631 / 895168
Chwolka	Fritz	Saarstraße 34	W- 5173	Aldenhoven	02464 / 8920	
Dose	Volker	Dorfstraße 10	W- 2304	Brodersdorf	04343 / 1357	
Drechsler	Oskar	Duckterather Busch 2	W- 5060	Bergisch Gladbach 2	02202 / 55282	0228 / 128114
Gill	Thomas	Maria-Eich-Str. 34	W- 8000	München 60	089 /8349527	
Günther	Jens	Bannerscheid 7	W- 5231	Neitersen	02681 / 1553	
Halgasch	Gert	Großschonauer Straße 26	O- 8805	Jonsdorf		
Hartmann	Hans-Günther	Möwenstraße 9	W- 2876	Berne 2	04406 / 6911	0421 /2482419
Hebecker	Ulrich	Büsumerstraße 15	W- 7000	Stuttgart	0711 / 734800	
Held	Manfred	Stirner Straße 22	W- 8835	Pleinfeld	09144 / 6563	0911 /2195245
Hermann	Klaus	Forchenstraße 8	W- 7401	Pliezhausen	07127 / 71945	
Hürdler	Manfred	Niederhofer Straße 29	W- 8709	Rimper	09365 / 4235	
Johnen	Willi	Hansemannstraße 1	W- 5160	Düren	02421 / 501305	
Kauka	Dietmar	Straße des Friedens 37	A- 7201	Neukirchen	0037404/850112	
Kemmer	Jürgen	Dorfberg 7	W- 8701	Sulzdorf	09334 / 1050	
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 /4828948	00411 / 258170
Kuhn	Eckehard	Im Dorf 14	W- 7443	Frickenhausen	07022 / 45417	
Littmann	Claus	Plockhorst, Zum Spring 15	W- 3155	Edemissen	05372 / 7796	
Lorenz	Walter	Mahräckerstraße 9	W- 6000	Frankfurt /Main 50	069 / 531656	
Magnus	Andreas	Bismarckstraße 29	W- 4650	Gelsenkirchen	0209 / 870230	
Mahlert	Herbert	Hohenbudbergerstraße 112 A	W- 4100	Duisburg	02135 / 47217	
Mand	Harald	Kl. Flintbeker Straße 7	W- 2302	Flintbek /Kiel	04347 / 3629	0431 /3013580
Menk	Christian	Ollsener Straße 52	W- 2116	Hanstedt	04184 / 7825	
Müller	Kurt	Sophie-Scholl-Ring 3b	W- 2054	Geesthacht	04152 / 70643	04151 / 889137
Mössel	Franz	Schafferstraße 12	I- 39012	Meran		
Neebe	Gerhard	Märkische Straße 186	W- 4600	Dortmund 1	0231 / 416549	
Neueder	Jens	Rudolf-Then-Straße 32	W- 7178	Gschlachtenbretzingen	0791 / 42877	0791 /44-4722
Neumann	Christof	Zeitblomstraße 22/2	W- 7900	Ulm	0731 /6022568	0731 /9749720
Nitschke	Stefan	Germanenstraße 5	W- 7519	Walzbachtal 1	07203 / 452	
Obermann	Hartmut	Mozartring 23	W- 8870	Günzburg	08221 / 30248	
Peters	Jürgen	Heukoppel 14	W- 2000	Hamburg 14	040 /6412371	
Reit	Hermann	Vechter Hof 40	W- 4500	Osnabrück	0541 / 16331	
Retzlaff	Bernd	Kleiner Sand 98	W- 2082	Uetersen	04122 / 43551	04103 / 605310
Riechmann	Michael	Letelner Heideweg 12	W- 4950	Minden	0571 / 36627	
Rinio	Gerd	Rennbahnstraße 9	W- 2000	Hamburg 74	040 /6552630	
Ruschinski	Claus	Pommernstraße 21	W- 4370	Marl	02365 / 34646	
Schilling	Werner	Ehndorfer Straße 340	W- 2350	Neumünster	04321 / 61116	
Schimmer	Jörg	Castellring 55	W- 6369	Nidderau 1	06187 / 25503	
Schmid	Alexander	Entmannsdorf 5	W- 8640	Kronach/Gehülz		
Schmitz	Rainer	Dornierstraße 17	W- 7320	Göppingen	07161 / 22549	
Schober	Frank-Michael	Weberweg 2	O- 7590	Spremberg	0037574/ 4565	
Schoberth	Uwe	Petrus-Waldus-Straße 14	W- 7136	Oetisheim	07041 / 7254	0711 /2011455
Scholz	Hans-Werner	Spitalstraße 54	W- 4054	Nettetal		
Schroers	Horst-Dieter	Breslauer Straße 9	W- 8016	Feldkirchen	089 /9032615	089 /9032615
Schröder	Gerald	Am Schützenplatz 14	W- 2105	Seevetal 1	04105 / 2602	
Schröder	Peter	Theodor-Fahr-Straße 32	W- 2000	Hamburg 62	040 /5311582	
Schröer	Egbert	Joachimstraße 18	W- 4270	Dorsten 1	02362 / 75311	02362 / 494392
Schulte	Hartmut	Entenschnabel 8	W- 3162	Uetze	05173 / 1248	
Schut	Andre	Sanderstraße 26	W- 1000	Berlin 44	030 /6917861	
Schwarz	Wolfgang	Schwedenring 6	W- 8850	Donauwörth	0906 / 3092	
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	W- 5305	Alfter 4	0228 / 643853	
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanloese		
Sonnemann	Harald	In den Eckwiesen	W- 6101	Fischbachtal	06131 /320860	
Stumpferl	Stefan	Hasenberglstraße 57	W- 8000	München 45	089 /3144001	
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	W- 6500	Mainz 1	06131 / 320860	06131 / 395268
Tornow	Wilhelm	Görlitzer Straße 16	W- 2190	Cuxhaven 13	04723 / 1355	04723 / 3063
Vollkmer	Richart	Am Spörkel 69	W- 4600	Dortmund 50		
Volz	Oliver	Am Ochsenwald 37A	W- 7000	Stuttgart 80 (Rohrerhöh	0711 /744051	0711 /6853013
Werner	Heiko	Reichenberger Straße 5	O- 8032	Dresden		
Wittkamp	Heinz	Hindenburgstraße 37	W- 5630	Remscheid	02191 / 75132	
Wulf	Hans-Otto	Im Brahmkamp 38	W- 4250	Bottrop	02041 / 688972	

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

HARDWARE + SOFTWARE
COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80
ERFAHRUNGSAUSTAUSCH



CLUBINFO

39. Ausgabe

Kontaktadresse:
Club 80
Hartmut Obermann
Mozartring 23
Postfach 1430
8870 Günzburg
Tel.: 08221/30248
Anrufbeantworter &
FAX: 08221/33575

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Das CLUB 80-Treffen 1993	1 - 4
	Hartmut Obermann
Hallo Clubfreunde	5
	Fritz Chwolka
Termine	6
	Redaktion
FIDO-Netz	6
	Hartmut Obermann
DFÜ im Club-80	7
	Alexander Schmid
Im Club 80 hält DFÜ Einzug	8 - 12

Software

CRR Offlinereader für CP/M	13 - 17
	Joerg Schimmer
	Alexander Schmid
Arbeiten mit Quickbbs	19 - 22
	Artikel aus TOOL
Herstellung einer Terminalbeschreibung	23 - 27
WDR Grafik Files & HRG Modell 1 Grafik Files	28 - 30
	Egbert Schrör

Autor & Seite

Hardware

ATOnce, ATTwice...	31 - 33
	Kurt Müller
Schneid-Klopf-Verbinder	34
	Alexander Schmid

Sonstiges

Wa-Tor	35 - 46
	Artikel aus ...
SMILIES	47 - 49
	Hartmut Obermann
	Alexander Schmid

Die letzten Seiten

Impressum	51
Schluß	52
	Redaktion
Mitgliederadressenliste	am INFO-Ende
Sonderhefte	am INFO-Ende
Fragebogen	am INFO-Ende

Das CLUB 80-Treffen 1993

Liebe CLUB 80-Mitglieder,

im folgenden Artikel will ich euch kurz über das Jahrestreffen 1993, welches vom 12. bis 14. März in Burgau-Röfingen stattfand, berichten.

Vorwort

Zunächst zu den weniger erfreulichen Aspekten des diesjährigen Treffens. Am Freitag waren neben meiner Frau und mir nur Alexander Schmid und Rainer Schmitz anwesend. Samstags waren insgesamt neun Mitglieder anwesend - das Treffen war also eines der am schlechtesten besuchten seit Bestehen des CLUB 80. Ebenfalls weniger erfreulich war die geringe Präsenz der Z80- bzw. CP/M-Rechner. Neben drei PX-8 CP/M-Notebooks und Alexander Schmids Z280-Maschine waren nur MSDOSen vertreten. Auch wenn einer dieser MSDOS-Rechner meiner war, muß ich sagen: "Für einen Z80- und CP/M-Club ein schwaches Bild!".

Trotzdem möchte ich das Treffen insgesamt als gut und die während des offiziellen Teils erzielten Ergebnisse als positiv bezeichnen (auch wenn mir der eine oder andere Teilnehmer da eventuell nicht ganz zustimmen wird). Doch dazu später mehr.

DFÜ und FIDO

Vielleicht ist euch schon aufgefallen, daß in den letzten CLUB-Infos mehrere Beiträge über Datenfernübertragung und das FIDO-netz zu finden waren. Auch in dieser Ausgabe finden sich einige Beiträge zu diesem Thema.

Datenfernübertragung an sich ist ja auch für CP/M-Jünger nichts neues. Im Gegensatz dazu ist, zumindestens für mich, der auf dem Treffen vorgeführte (und hier im Info vorgestellte) FIDO-Offline-Reader für CP/M ein großer Fortschritt. CP/M-Nutzer mit Modem werden dadurch erstmals in die Lage versetzt, mit ihrem Rechner kostengünstig das FIDO-Netz zu nutzen. Wer Interesse an dem Programm hat, kann sich bei Alexander Schmid melden, der als Experte und Ansprechpartner zur Verfügung steht.

Klaus Hermann, der eine Mailbox im LIFE-Netz betreibt, hatte seine MSDOS-Festplatte mit der gesamten Box-Software mitgebracht. Nachdem diese in meine MSDOSe eingebaut war, konnte er zusammen mit Alexander Schmid und dessen Z280-Maschine das Prinzip des Offline-Readers live demonstrieren. Aber auch der Mailbox-Onlinebetrieb konnte durchgespielt werden.

Im DFÜ-Bereich wird sich in nächster Zeit einiges tun, womit wir dann auch schon beim offiziellen Teil des Treffens, der Mitgliederversammlung, wären.

Protokoll der Mitgliederversammlung

Als Protokollführer stellte sich Rainer Schmitz zur Verfügung. Wie es sich für einen echten Computerfreak gehört, tippte er das Protokoll gleich in seinen mit WordStar ausgerüsteten CP/M-Notebook EPSON PX8. Wie abgesprochen erlaube ich mir allerdings, den Text etwas "mit Leben" zu füllen und die zeitliche Reihenfolge etwas durcheinanderzuwirbeln bzw. zu ordnen.

Anwesend (alphabetische Reihenfolge):
Günther Braun
Manfred Held
Klaus Hermann
Jens Neueder
Hartmut Obermann
Alexander Schmid
Rainer Schmitz
Uwe Schoberth

Wie der Anwesenheitsliste schon zu entnehmen ist, waren weder der erste noch der zweiter Vorsitzende anwesend und so konnte der alte Vorstand nicht entlastet werden. Allerdings hat uns Fritz in einem Schreiben, welches sich auch in diesem Info findet, seinen Rücktritt vom Amt des ersten Vorsitzenden erklärt. Das brachte uns in den Zugzwang, einen neuen ersten Vorsitzenden wählen zu müssen.

Die Kassenprüfung übernahmen Klaus Hermann und Jens Neueder. Es wurden die Jahre 1991 und 1992 geprüft und das Ergebnis war sehr positiv und erbrachte keine Abweichungen. Der Kassenführer wurde daraufhin von der Versammlung mit einer Enthaltung entlastet und zudem in seinem Amt bestätigt.

In diesem Zusammenhang war jedoch die Frage von Bedeutung, ob der erste Vorsitzende zugleich Kassenführer sein darf. Mangels weiterer Bewerber (Günther Braun hatte an seine Kanditatur Bedingungen geknüpft, die die Anwesenden nicht akzeptieren wollten, und sie daher zurückgezogen) war auch die Wahl des 1. Vorsitzenden nur eine Formsache und so habe ich nun zwei Ämter im CLUB 80 inne.

Die Vorstandschaft sieht also wie folgt aus:

1. Vorsitzender:	Hartmut Obermann
2. Vorsitzender:	Gerald Schröder
Kassenführer:	Hartmut Obermann
Info-Redakteur:	Jens Neueder
Hardware-Koordinator:	Andreas Magnus
NewDOS-Diskothekar:	Oliver Volz
CP/M-Diskothekar:	Fritz Chwolka
C-128-Diskothekar:	Günter Braun
Bücherei:	Kurt Müller

Bezahlung für Artikel

Schon vor der Wahl des neuen Vorstands wurden einige Punkte teilweise heftig und kontrovers diskutiert. Einer davon war Günter Brauns Antrag, Autoren für im Info veröffentlichte Artikel zu bezahlen. Dieser Antrag wurde mit 8 zu 1 Stimmen abgelehnt.

Mitgliedsbeitrag/Aufnahmegebühr

Ebenfalls keine Zustimmung fand Günters Meinung, den ausgesetzten Mitgliederbeitrag wieder einzuführen. Statt dessen folgte die Mehrheit der Anwesenden meinem Vorschlag, zwar weiterhin eine Aufnahmegebühr für Neumitglieder zu erheben, ansonsten aber bis auf weiteres auf einen jährlichen Mitgliedsbeitrag zu verzichten. Die Aufnahmegebühr beträgt bis auf weiteres 30,-- DM.

CLUB 80-Info

Die Entscheidung über einen von Günter Braun und Uwe Schobert eingebrachten Vorschlag, das Info ab sofort im DIN A4-Hochformat zu veröffentlichen, soll eine Abstimmung aller Clubmitglieder bringen. Es wurde verabredet, daß Günter Braun einen "pro"-, Jens Neueder einen "kontra"-Artikel zu diesem Thema schreiben. Die Abstimmung erfolgt dann mit dem Formular, mit dem auch die neuen Postleitzahlen usw. ermittelt werden.

Eine heftige Diskussion entfachte Günter Brauns Vorschlag zur Erhöhung der Kontinuität des Clubinfos. Seiner Meinung nach sollten in "fetten Zeiten", also wenn viele Artikel vorliegen, einige davon gesammelt und erst dann veröffentlicht werden, wenn zum nächsten (oder übernächsten) Redaktionsschluß zu wenig Material vorhanden ist. Da die Zurückhaltung von Artikeln eine Art Zensur bedeuten würde, wurde ein Kompromiss gesucht und gefunden.

Ab sofort gelten die Redaktionstermine 01. März, 01. Juni, 01. September und 01. Dezember. Was bis zu diesen Terminen bei Jens eingetroffen ist wird veröffentlicht. Liegt viel vor, gibt es ein dickes Info. Liegt wenig vor kommt eben nur ein Minimalinfo (Deckblatt und Impressum). Es liegt also ganz an der Mitarbeit der Clubmitglieder, wie ein Info ausfällt!

CP/M aktuell

Mit Günter Brauns "Kreis von Commodore-128-Anwendern" mit dem Namen "CP/M aktuell" wurde eine Zusammenarbeit vereinbart, die wie folgt aussieht:

1. Günter Braun sammelt Artikel seiner Mitglieder und übersendet sie Jens zur Veröffentlichung
2. im Gegenzug dazu bekommt Günter Braun 50 Infos des CLUB 80 zum Selbstkostenpreis von 10 Pfennigen pro Blatt.

Selbstverständlich sind alle Mitglieder von "CP/M aktuell" recht herzlich dazu eingeladen, Mitglied im CLUB 80 zu werden. Die Aufnahmegebühr beträgt wie gesagt 30 DM. Ein Anmeldeformular und weitere Informationen über den CLUB 80 können direkt bei mir angefordert werden.

CLUB 80 und DFÜ

Wie im Vorwort schon erwähnt, betreibt Klaus Hermann eine Mailbox im Life-Net. Schon vor längerem bot er dem CLUB 80 an, darin ein Brett für die Clubmitglieder einzurichten. Dies wird nun realisiert. Der CLUB 80 bekommt in der "Discovery" eine Message- und File-Area, es können also sowohl Nachrichten als auch Programme dort abgelegt und abgeholt werden. Als "Moderator" werde ich fungieren. Alles weitere dazu findet ihr in einem gesonderten Artikel in diesem Info. Ich würde mich sehr freuen, wenn alle Mitglieder die neue Kommunikationsmöglichkeit rege nutzten! Die Areas sind übrigens öffentlich, so daß natürlich auch "CP/M aktuell"-Mitglieder Zugriff haben.

BASIC-Ecke

Last but not least wäre da Jens Neueders BASIC-Ecke zu erwähnen, die leider bisher zu keiner allzu großen Resonanz geführt hat. Bis Ende 1993 wird Jens die Ecke noch alleine bestreiten, sollten danach keine Rückmeldungen interessierter BASIC-Programmierer eintreffen, wird er sie wohl einstellen.

Soweit zum Clubtreffen und den dort besprochenen Themen. Nun noch zu einem Thema, das mich zur Zeit beschäftigt, der Mitgliederkartei.

Mitgliederkartei

Aus verschiedenen Gründen halte ich es für notwendig, die Mitgliederkartei zu aktualisieren. Einer der Gründe sind zum Beispiel die neuen Postleitzahlen, die ab 01. Juli 93 gelten werden. Ein weiterer ist die Erfassung eventuell neu dazugekommener Hard- und Software oder neuer Interessensgebiete (z.B. DFÜ usw.).

Ein anderer, ebenso wichtiger Grund ist die "Beerdigung" von Karteileichen. Mitgliedern also, die nie eine offizielle Abmeldung geschickt haben und durch die "Null-Beitrags-Regelung" auf ewig dem Club angehören würden. Ich gehe davon aus daß, wer den Fragebogen nicht bis zum 30. Juni an mich zurückschickt, kein Interesse mehr am CLUB 80 hat und austreten möchte.

Ich weise darauf hin, daß die Daten mit Sicherheit in meinem Computer landen, versichere jedoch, daß sie nur für Clubzwecke genutzt und nicht an Dritte weitergegeben werden!

So, das war's jetzt endgültig. Ich hoffe, daß ich dem CLUB 80 für das vor uns liegende Jahr bis zur nächsten Hauptversammlung ein guter Vorsitzender sein werde und würde mich freuen, wenn mich alle Mitglieder nach Kräften unterstützen würden.

In diesem Sinne

Allways good Hacking, H.O.!

Hallo Clubfreunde!

Leider kann ich aus beruflichen Gründen nicht an der Versammlung teilnehmen. Ich hoffe aber ,daß einige der neueren Mitglieder der Versammlung beiwohnen und die Themen so interessant sind wie sonst. Prost!
Da mir eine geregelte Clubarbeit selten möglich ist lege ich den Vorsitz nieder.
Der hoffentlich neu gewählte Vorstand (und Vorsitzende) wird aber weiterhin meiner Unterstützung sicher sein. Nach Absprache mit dem Rüdiger Sörensen werde ich den SOFTWARE-Archivar für CP/M machen,natürlich nur falls die Teilnehmer der Hauptversammlung mich dafür ausgucken.
Auch wenn nicht immer Interesse an der Clubarbeit vorhanden war hat mir der Vorsitz gefallen, und dazu möchte ich noch einige kurze Sätze schreiben.

Meist hatte ich mit den gleichen Personen (auch Nichtclubmitglieder) Kontakt und konnte manchmal helfen , wobei nicht nur CP/M sondern auch andere Betriebssysteme behandelt wurden.
Den Club verstehe ich nicht "nur" als Z80 und CP/M -Club, sondern als letzte Anlaufstelle bei Problemen und der Suche nach Gleichgesinnten bezüglich alter HARD- und Software. Ebenso habe ich auch Hilfe bei anderen Betriebssystemen (UNIX,LINUX,CP/M86) erhalten und dafür sind die Kontakte da.
Viele von UNS haben meist einen anderen (kompatiblen) Rechner, sei es ein IBM-Clown, Atari oder NeXT. Auch auf diesen Geräten gibt es gute Software wie Minix, Coherent, Linux etc. und die Spezialisten unter uns sollten auch darüber was schreiben.
Ich werde in meinem nächsten Brief etwas von meiner Erfahrung mit LINUX berichten.
Da ich gerne am Betriebssystem , der Hardware und der Software bastell werde ich versuchen meine Erfahrungen weiterzugeben. Egal für welches Betriebssystem und dahin sollte der Club sich bewegen.
Der Kern der Mitglieder sind keine Anwender , sondern wollen selber was tun und verstehen und mit keiner Hardware kann man besser den COMPUTER (und billiger) kennenlernen als mit 8Bit-prozessoren. Da aber fast jeder auch mit neuer Hard- und Software Erfahrung hat, wobei ich durch das Hobby Z80 eine gute Basis habe, sollte er sich auch darüber auslassen können.
Dies als Anmerkung und in der Hoffnung auf weitere gute INFO's.
Abschließend möchte ich Euch ein schönes Clubtreffen wünschen und mich für die zurückliegende Zusammenarbeit , persöhnlichen Gesprächen und die Clubgemeinschaft bedanken.

Viele Grüße

Euer

Fritz

Termine... Termine... Termine... Termine... Termine

Büro,Kommunikation,Computer	Nürnberg	06.05.- 08.05.93
CAT	Stuttgart	11.05.- 14.05.93
Computer-Schau	Dortmund	12.05.- 16.05.93
Hobby-tronic	Dortmund	12.05.- 16.05.93
Eltec 93	Nürnberg	17.06.- 19.06.93
HAM RADIO	Friedrichshafen	26.06.- 27.06.93
IFA	Berlin	27.08.- 05.09.93
Fachmesse Elektrotechnik	Dortmund	01.09.- 04.09.93
MICRO SYSTEMS TECH. 1993	Berlin	21.09.- 24.09.93
SYSTEMS	München	18.10.- 22.10.93
PRODUKTRONICA	München	09.11.- 13.11.93
Nord Elektro	Hamburg	18.11.- 20.11.93

<u>Redaktionsschluß</u> für das nächste Clubinfo ist der

01. Juni 93.

FIDO-Netz

Im letzten Info konntet Ihr einen sehr interessanten Artikel über das FIDO-Netz lesen. Sog. FIDO-Mailer (z.B. FrontDoor) gibt es für fast alle Computer (MSDOSen, ATARI ST, AMIGA) aber leider nicht für CP/M-Rechner (es würde mich freuen, wenn ich mich in diesem Punkt irren würde).

Trotzdem sind einige wenige Mitglieder des CLUB 80 über FIDO zu erreichen (die werden doch nicht heimlich mit einem PC arbeiten)!? Hier die "Adressen", auch FIDO-Node genannt:

Hans-Günther Hartmann, 2:240/300.24
Klaus Hermann, 2:247/770.740
Hartmut Obermann, 2:241/7902.10

Es kann durchaus sein, daß diese leider recht kurze Liste nicht vollständig ist. Sollten noch weitere Info-Leser über FIDO erreichbar sein, mögen sie mir dies bitte mitteilen. Im nächsten Info gibt es dann vielleicht schon eine umfangreichere Liste.

Sollte jemand noch nicht Mitglied im FIDO sein, aber Interesse daran haben, kann er sich bei mir melden. Ich bin gerne bereit, beim Einstieg zu helfen.

Mt+ Oz

DFÜ im Club-80

Es ist kaum zu übersehen, daß auch in unserem Info das Thema DFÜ immer wieder auftaucht. Viele dürften inzwischen ja auch einen zweit-PC haben und damit an irgendeinem Netz hängen und so würde ich vorschlagen, daß neben den Telefonnummern auch die Fido- oder sonstige Netzadresse in die Adressenliste aufgenommen wird. Dieses Info paßt gerade zum nächsten Haupttreffen und jeder, der kommt, kann sich schon mal Gedanken zu den Ideen hier machen. Ob sich allerdings eine eigene Club-Mailbox lohnt, wage ich zu bezweifeln. Egal, wer die betreibt, für fast alle anderen sind es Ferngespräche und so wird kaum jemand regelmäßig dort reinschauen, ob es was neues gibt. Man könnte es natürlich so machen, daß die Box z.B. nur am Wochenende ereichbar ist, dann müßte der Betreiber seinen Compi auch nicht Tag und Nacht laufen lassen und damit mehr oder weniger vollständig zu blockieren. Bei unserem Kapital könnten wir uns eventuell einen gebrauchten PC für ein paar Hunderter als Boxrechner leisten, damit niemand seinen eigenen zur Verfügung stellen muß.
Eine ganz andere Möglichkeit wäre, daß wir uns an eine schon vorhandene Box anhängen. In München gibt es die SCUG, die Schneider-CPC-User-Group, auch ein Z80-Club, der sich mit den CPC's und natürlich auch mit CP/M beschäftigt. Einer von denen betreibt schon länger seine eigene Mailbox und hat dort ein Brett "SCUG-Intern" speziell für die Mitglieder eingerichtet. Im Fido-Netz gibt es die Möglichkeit, "eigene" Bretter über das Netz weiter zu verteilen, aber da sollten sich mal die Experten zu Wort melden, wie man das am besten machen könnte. Dann könnte da jeder über seine Hausbox zum Ortstarif rankommen und wer nicht im Fido ist, für den ist es so oder so ein Ferngespräch, egal wer die Box betreibt. Im Moment ist da nicht viel los und die haben mit Sicherheit nichts dagegen, wenn sie mal wieder ein paar interessante Artikel zum Thema Z80 lesen. Der 1. Vorsitzende der SCUG ist übrigens Stefan Stumpferl und ganz zufällig ist er auch im Club-80; wenn sich da nichts machen läßt...
Solange aber nicht jeder ein Modem hat, finde ich, daß wir das gedruckte Info nicht einstellen sollten, sondern die DFÜ als Ergänzung für die ganz eiligen nehmen. Besonders die Artikel mit Schaltplänn lassen sich mangels Grafikstandart (nicht jeder hat einen PC !) auch schlecht per DFÜ übertragen. Für Listings ist die DFÜ aber nahezu ideal, dann hätte die elendige Abtipperei endlich ein Ende und das geht einfacher und schneller als über die Diskothek. Wenn man nicht unbedingt ein High-Speed-Modem oder eines mit Fax braucht, kann man schon ab 140 Mark (neu !) für ein 2400er dabei sein, gebraucht u.U. natürlich noch billiger.

Alexander Schmid

Im Club 80 hält die DFÜ Einzug

1 Vorbemerkung

Nachdem der praktische Versuch, per CP/M-Offline-Reader aus einer normalen DOS Mailbox Nachrichten zu laden, gelungen ist und wir beim Clubtreffen beschlossen haben, ein Schwarzes Brett für den Club 80 einzurichten, will ich hier beschreiben, wie das praktisch aussehen kann.

Da sich eine Mailbox alleine für den Club 80 nicht lohnen würde und ich bereits eine Mailbox betreibe, habe ich mich gerne bereit erklärt, für den Club 80 ein Nachrichtenbrett und eine Filearea einzurichten. Bei einem entsprechenden hohen Aufkommen könnten diese zwei Bereiche sicher auch erweitert werden.

2 Grundsätzliches zu einer Mailbox

Zunächst ein paar grundsätzliche Dinge.

Als erstes braucht man natürlich ein ***Modem***, wobei ich mir habe sagen lassen, daß 1200 Baud oder 2400 Baud für CP/M Rechner ausreichend sind bzw. der CP/M Rechner keine höheren Geschwindigkeiten verkraftet.

Daneben benötigt man ein ***Terminalprogramm***, das es unter CP/M sicher im Shareware / Public Domain Bereich geben dürfte.

Damit man nicht "stundenlang" in einer Mailbox online sein muß, um die gewünschten Nachrichten zu lesen, empfiehlt sich ein ***Offline-Reader***.
In der Discovery Mailbox habe ich das Gegenstück zum Offline-Reader eingebaut. Mit diesem ist es möglich, das oder die gewünschten "Schwarzen Bretter" auszuwählen. Die Nachrichten in diesem Brett werden dann in eine Datei gepackt, komprimiert und anschließend an den Mailboxbenutzer (User) übertragen. Dieser kann sich danach aus der Mailbox ausloggen. Sein Offline-Reader entpackt nun diese Datei zu Hause auf dem Rechner und er kann dort in aller Ruhe die Nachrichten lesen. Der umgekehrte Weg ist natürlich auch möglich: Zu Hause mit dem Offline-Reader Nachrichten schreiben und beim nächsten Anruf in die Box uploaden.

Der Vorteil liegt bar auf der Hand. Durch die kurzen Onlinezeiten in der Mailbox spart man auf Dauer Geld..

3 Anruf in der Discovery Mailbox

Zunächst die technischen Daten zur Discovery

- Tel.: 0 71 27 / 7 01 07
- 24 Stunden online
- 300 - 14400 bps (V.21, V.22, V.22bis, V32, V32.bis, MNP 2-5, V.42, V.42bis)
- 8 Datenbits, kein Paritätsbit, 1 Stopbit
- Standort: 7401 Pliezhausen

Die Discovery ist seit August 1990 online und dem "LifeNet - Verbund christlicher Mailboxen" angeschlossen.

3.1 Einloggen in die Mailbox

Nach der Anwahl der Mailbox wird der Eingangsbildschirm und einige Infos angezeigt. Danach wird

- der Vorname und der Nachname eingegeben.
- Ist man bereits eingetragener User wird nach dem Passwort gefragt.
- Loggt man sich zum ersten Mal ein, müssen einige Angaben zur Person und technische Angaben gemacht werden. Außerdem ist das Ausfüllen des Userantrags notwendig. Beim Userantrag wird u.a. gefragt, woher die Nummer der Mailbox bekannt ist. Tragt dort bitte **Club 80** ein.
- Wenn diese Dinge alle erledigt sind, gelangt man ins Hauptmenü.

```
                 +---------------------+
                 ¦  H a u p t m e n u e ¦
                 +---------------------+

+----------------------------------+----------------------------------+
¦ S  Systemeinstellungen/-infos    ¦ N  Nachrichtenbereich            ¦
¦ R  Sysop rufen                   ¦ F  Filebereich                   ¦
+----------------------------------+----------------------------------+
¦ U  Hobby  Unterhaltung  Infos    ¦ G  Glaube / Religion             ¦
¦ W  Wissenswertes                 ¦ E  Erlebtes                      ¦
¦ P  Preis-Quiz                    ¦ C  Geschichten/Flopsi Compuso    ¦
+----------------------------------+----------------------------------+
¦ 1  1992 - Jahr mit der Bibel     ¦ !  Logoff/Ende                   ¦
+----------------------------------+----------------------------------+
Donnerstag , 18-03-93 , 20:29 Uhr - Restzeit: 90 Min. / AUSWAHL ==>>
```


Für die Clubmitglieder sind zunächst mal 3 Menüs von Bedeutung

- `Systemeinstellungen/-infos`
  - hier können die individuellen Parameter eingestellt werden
- `Nachrichtenbereich`
  - hier befindet sich das Nachrichtenbrett **CLUB80.DISC**
- `Filebereich`
  - hier befindet sich die Dateiensammlung für CP/M bzw. Club 80 Software

3.2 Nachrichtenbereich

Um in den Nachrichtenbereich zu kommen, drückt man einfach den Buchstaben **N**

```
   [ Nachrichtenbereich ]

[S] Suche nach neuen Nachrichten an mich (alle Bretter)

[ Privates Postfach ]
[1] Privates Postfach Discovery ...
[2] Privates Postfach LifeNet ...

[ Oeffentliche Nachrichtenbretter ]
[D] Discovery lokal ...
[L] LifeNet ...                    [P] PARABEL.LIFE

[A] Alle Nachrichtenbereiche (Kombinierte Bretter) ...
[Q] QWK Mail

-  voriges Menue   * Hauptmenue      ! Logoff      H Hilfe

Auswahl ->
```

Das Brett des Club 80, ist dem öffentlichen Nachrichtenbereich `Discovery lokal` angegliedert und heißt **CLUB80.DISC**

Hier hat man nun grundsätzlich zwei Möglichkeiten. Entweder man wählt direkt den Download der Nachrichten für den Offline-Reader `QWK Mail` aus oder man schaut sich die Nachrichten im Brett in der Box direkt an.

3.2.1 Download für Offline-Reader.

Hierzu wählt man **Q** für `QWK Mail` aus. Daraufhin wird ein Programm geladen, das die QWK-Nachrichtenpakete bearbeitet. Zunächst sollte man den Menüpunkt für die Konfiguration auswählen und vor allem den gewünschten Archivierer / Packer und das Transfer-Protokoll wählen. Für letzteres empfehle ich das ZMODEM Protokoll. Wenn die Einstellungen gemacht sind, wählt man aus dem Menü das gewünschte Brett (CLUB80.DISC) aus und überträgt es anschließend per Download zu sich nach Hause. (Da sich das QWK-Programm auf Mailboxseite noch ändern kann, unterbleibt an dieser Stelle ein Menüabdruck - die Funktionen sind jedoch in den div. Programmen im wesentlichen identisch)

3.2.2 Brett in der Box direkt anschauen

Hierzu wählt man **D** für `Discovery lokal` aus, worauf ein weiteres Menü mit verschiedenen Möglichkeiten angezeigt wird.

```
[ Nachrichten einzeln ]    eingestelltes Brett:
..........

[N] Neue Nachrichten in folgenden Brettern:

[W] Brett waehlen
[V] Ein Brett vorwaerts blaettern
[R] Ein Brett rueckwaerts blaettern
[I] Inhaltsverzeichnis
[B] Inhaltsverzeichnis Betreff
[L] Lesen
[S] Schreiben

- voriges Menue        * Hauptmenue        ! Logoff
```

Mit **W** wählt man das Brett CLUB80.DISC aus und kann anschließend mit Lesen die Nachrichten vorwärts, rückwärts etc. lesen.

3.3 Filebereich

In den Filebereich, der für den Club 80 eingerichtet wird, können CP/M Programme, aber auch Listen von Shareware-Serien upgeloadet werden.
Ähnlich wie die Auswahl der Nachrichtenbretter funktioniert auch die Auswahl der Filebereiche. Mit **F** wählt man den Filebereich aus und erhält dieses Menü:

```
 [ Files Menue ] - eingestellte Area:  ...........

Zugriff auf _alle_ Fileareas nur fuer eingetragene User

[A] Area wechseln
[F] Files in dieser Area               [L] Neue Uploads seit letztem
                                                          Anruf

[W] Suche nach Stichwort (alle Areas)  [N] Suche nach Files (alle
                                                           Areas)

[Z] Inhalt von Archiv
[U] Upload (senden)                    [D] Download (empfangen)
[S] Statistik                          [H] Hilfe

[T] Fileliste der Discovery downloaden
[K] Download LifeNet-Infoprogramm

- voriges Menue      * Hauptmenue       ! Logoff

Auswahl ->
```

Mit **A** kann man die Filearea auswählen: **CLUB 80 (CP/M)**. Man kann sich nun die Files in dieser Area ansehen (**F**) und anschließend Files uploaden (<u>zur</u> Mailbox übertragen) oder downloaden (<u>von</u> der Mailbox übertragen). Auch hier sollte man das ZMODEM Protokoll auswählen.

<u>Wichtig:</u> Wenn ihr Software uploadet, achtet bitte darauf, daß es sich ausschließlich um Shareware oder Public Domain Software handelt. Die Haftung hierfür übernimmt der User, der die Software uploadet.

3.4 Nachrichten schreiben

3.4.1 Öffentliche Nachrichten

Im öffentlichen Brett des Club 80 können Nachrichten geschrieben werden, die alle User lesen können. Natürlich kann auf diese Nachrichten auch wieder geantwortet werden.

Dies kann zunächst mal zu Hause über den Offline-Reader geschehen. Allerdings kann es auch mal sein, daß man kurzfristig eine Nachricht online schreiben will. Hierzu wählt man **S** für Nachrichten schreiben. Im ASCII - Modus wird ein einfacher Zeileneditor aufgerufen, im ANSI - Modus kann man einen komfortableren Bildschirmeditor benutzen (Einstellung hierfür im Menü `Systemeinstellungen - Parameter`)

3.4.2 Private Nachrichten

Neben den öffentlichen Nachrichten kann man den Usern auch private Nachrichten schreiben, die nur der Absender und der Empfänger lesen können. Hierzu wählt man das Brett **POST.DISC** aus und schreibt dort die Nachricht w.o. beschrieben.

So, das wäre wohl das Wichtigste gewesen. Wenn ihr mal nicht weiter wißt oder Probleme habt, könnt ihr euch natürlich gerne auch an mich wenden. Das ist entweder während dem Mailboxbesuch durch den *Sysopruf* möglich (wenn ich da bin, melde ich mich kurz drauf). Oder ihr schreibt mir in der Mailbox eine persönliche Nachricht. Daneben könnt ihr mich natürlich auch anrufen.

Zu dem CP/M Offline-Reader kann ich allerdings nichts sagen, da ich kein CP/M System mehr habe. Allerdings kann euch da sicher der Alexander Schmid weiterhelfen, der Erfahrung mit dieser Software hat.

Bis demnächst in der Mailbox !

Klaus Hermann

Hallo CP/M Freunde !

ich moechte heute mal ein tolles Shareware-Programm fuer CP/M 2.2 und CP/M Plus vorstellen. Es handelt sich um einen Offline-Reader für das FIDO Netz.
Erst mal kurz ein paar Daten zu mir:
Meinen Schneider CPC 664 habe ich im Juni 86 erstanden. Den 1. Akkustik-Koppler mit 300 - Baud habe ich mir im April 87 zugelegt. Schon bald folgte ein externes 5,25" Laufwerk, mittels Kippschalter 2x 178kb und eine RAM-Disc von Dobbertin mit 444k. Mittlerweile steht mein Schneiderlein im Keller und ich arbeite mit einem 486er mit OS/2 2.0 Betriebssystem. Beruflich arbeite ich mit einem HP 1000 mit A400er CPU (falls den jemand kennt).
Mein Interesse an DFUE flaute schnell ab, als die Anfangseuphorie vorbei war. Mit Datex-P kam auch nicht der grosse Durchbruch und BTX nutzte ich nur gelegentlich für meine Kontoführung. Alles aenderte sich, als ich das weltweit verbreitete FIDO System kennenlernte. Endlich auch Kontakte in andere Laender und das alles zum Ortstarif. Die Datenflut brach anfangs total ueber mich herein und ich saß ganze Nächte lang vor der Flimmerkiste. Neidisch schaute ich auf andere User mit Amiga-Rechnern oder einer MS-DOSe mit ihren Pointprogrammen. Ich musste alles Online lesen und meine Telefonrechnung wurde immer höher, bis ich eines Tages in der Fido-Area CPM.GER von einem OFFLINE-READER fuer CP/M gelesen habe.
Gut also nix wie ran ans Modem und das Programm downloaden sagte ich zu mir -- gesagt getan -- hier meine Erfahrungen mit dem:

CRR Offlinereader fuer CP/M 2.2 und Plus mit mind. 42kb TPA
Autor:
Paul Martin, 85 Hollin Lane, Middleton, Manchester, GB-M24 3FE

1. Hinderniss: benoetigt wird laut DOC-File ein ARKer (ARK v0.4 bzw. ARK v1.0) und ein Editor (WS, VDE usw. oder irgendein anderer)
Ein entZIPer ist dabei (nur 3kb! Autor: Dave Goodenough)
Das Programmpaket enthält ein Install Programm und ein Default - File in dem Anpassungen vorgenommen werden müßen.
2. Hinderniss: um das Programm auszuprobieren muss der Sysop der Box erst eine Software installieren, die zu CRR kompatibel ist.
CRR ist das Äquivalent zu XRS unter MS-DOS vom Autor Mike Ratledge. Es arbeitet zusammen mit dem Programm XRSDOOR das auch bekannt ist unter den Namen: QMX, RAX, RAQMX und RQSeX. XRSDOOR laeuft unter QBBS, RemoteAccess und SuperBBS Mailbox Systemen. Mittlerweile gibt es auch eine Erweiterung damit man QWK - Formate bearbeiten kann. Falls die interface Software dafuer geschrieben ist laeuft es auch auf Opus, Maximus und XBBS boards und neuerdings auch mit XCS (PKT2XRS) ab v0.47 von Rudi Kuster.
Nach einigen Mühen und gutem Zureden von mir war es soweit:
Der Menüpunkt XRSDOOR v2.0 steht in der Mailbox. Ich nix wie rein und alles konfiguriert:
1. Auswahl aller mich interessierenden Areas.
2. Pack-Methode (ZIP, ARC, LHA)
3. Transfer-Methode (XMODEM, YMODEM, ZMODEM)
4. Option Switches (Suche ausserhalb der angegebenen Areas nach Post, usw.)
Nachdem alles eingestellt war habe ich mein 1. Paket geladen. Als maximale Anzahl der Mails gab ich 250 an, da ich eine RAM-DISC von 444k hatte, sonst entsprechend mehr oder weniger. Nach dem Logoff der Mailbox habe ich das Paket entZIPt bzw. spaeter entARCt (schneller). Das File der Mailbox enthaelt immer Dateien folgender Struktur.
AREAS1.XRS (die verfuegbaren Areas Deiner BOX)
BAT1MAIL.XRS (das Mailpaket selbst)
MAIL1IDX.XRS (der Index der Mails)
SUMMARY1.XRS (das von wem: betreffs: -File der Mails)
USER1.XRS (Name,Fido-Adresse von Dir)
oder andere Filenamen mit gleicher Struktur und Endung *.XRS

Nun kam der Moment und ich startete CRR und es erschien:
(R)ead messages (all)
(M)y mail only
(O)ne area only
(S)elect message from summary
(E)nter new message
(N)etmail enter (Anmerkung: falls es der Sysop erlaubt)
(V)iew replies
(P)ack the responses
(Q)uit
Ob Ihr es nun glaubt oder nicht -- alles funktioniert wie angegeben. Mails können vorwärts/rueckwärts gelesen werden bzw. direkt beantwortet werden mit autom. Adressierung usw. Als letztes wählt man `P` fuer packen und die Antworten/Mails werden geARKt und können beim nächsten Besuch der Box gesendet werden. Falls Ihr den Sysop überzeugt, kann er es ermöglichen gleichzeitig mit dem Senden der Antworten auch Dateien seiner Box automatisch mit dem Empfang des nächsten Paketes downzuloaden bzw. Areas an- oder abzubestellen. (so eine Art File-request)
Da CRR Shareware ist muss man nach 1 Monat angezeigte Passwörter eingeben oder besser sich fuer 15 Pfund Sterling (ca.45,-DM) bei Paul Martin registrieren lassen. Lässt man sich registrieren bekommt man die neueste Version von CRR (beta v1.13) bzw. bald die v2.0 und einen Key den man im DEFAULTS.CRR installieren muß. Briefwechsel von Frankfurt nach Manchester und zurück ca. 2 Wochen (normaler Brief), via Fido geht es schneller !
So ich hoffe ich konnte einige von Euch neugierig auf CRR machen und kann demnächst von Euch zu lesen =8-) (grins). Meine FIDO- Point-Adresse ist die 2:249/70.9 dort bin ich unter

Joerg Schimmer zu erreichen. Sollten Fragen auftauchen meine Neue Adresse ist: Stettinerstr.28, 6000 Frankfurt-60 Tel.: 06109/35336.

Tschüß Euer

Jörg

Als Point im FIDO unter CP/M

Wenn man das erste Mal in eine Fido-Mailbox geht und sich dort etwas umsieht wird man wohl bald auf all die schönen Bretter stoßen, in denen man fast über alles und jeden diskutiernen und Informationen bekommen kann (je nachdem, was der Sysop an Brettern anbietet). Wenn man sich nun jede Mail einzeln ansieht geht das sehr schnell ins Geld und außerdem werden sich die anderen User bald beschweren, wenn da jemand stundenlang die Box blockiert. Da wäre es doch viel praktischer, wenn man sich einfach alle neuen Mails zusammen und möglichst noch komprimiert abholen und dann in aller Ruhe zu Hause durcharbeiten könnte. Diese Idee hatten natürlich auch schon andere und so gibt es dafür u.a. die sogenannten Offline-Reader. Im DOSen-Bereich gibt es noch viel umfangreichere und komfortablere Software (sog. POINT-Software), nur für CP/M habe ich sowas leider noch nicht gesehen. Aber ein Offline-Reader ist auch schon eine wesentliche Erleichterung und so funktioniert das im Prinzip:

Man loggt sich mit einem Terminal-Programm in eine FIDO-Box ein und sucht sich den Menüpunkt QWK, Offline-Reader oder so ähnlich. Da gibt es dann einen Punkt, in dem man die Bretter auswählt, die man lesen will. Anschließend wählt man einen Packer aus, damit das Mailpaket möglichst klein wird. Als letzte Aktion geht man dann zum Download und die Box sucht in allen markierten Brettern die Mails zusammen, die man noch nicht gesehen hat. Wenn das Mailpaket übertragen ist, kann man sich schon wieder ausloggen und die Onlinezeit reduziert sich auf ein Minimum. Zu Hause packt man das Mailpaket mit einem entsprechenden Programm aus und startet den eigentlichen Offline-Reader. Mit dem kann man sich dann alle Mails in aller Ruhe ansehen und bei Bedarf auch selber welche schreiben. Wenn man nun selber was geschrieben hat, werden diese Mails vom Offline-Reader auch wieder gepackt und man kann sie beim nächsten Einloggen, wenn man sowieso ein neues Mailpaket abholt, uploaden.

Das Programm, um das es hier geht ist der

CRR Offlinereader für CP/M 2.2 und Plus
von Paul Martin, 85 Hollin Lane, Middleton, Manchester, GB-M24 3FE.

Zum Funktionieren des CRR (CP/M Read & Reply) braucht man noch einen externen Packer (z.B. ARC.COM, einen Entpacker (z.B. UNARC. COM) und einen Editor (z.B. VDE, ZDE, ...), aber die gibt es als PD fast überall zum Nulltarif. Das Programm ist menügeführt und auch ohne das recht ausführliche Handbuch sicher zu bedienen. Als kleines Appetithäppchen hier ein kleiner Überblick über die Menüs. Das Hauptmenü sieht folgendermaßen aus:

CP/M Read & Reply 1.40 (c) Copyright 1990,1991 Paul Martin

Thu 11 Feb 93

(R)ead messages ÄallÜ
(M)y mail only
(O)ne area only
(S)elect message from summary

(E)nter new message

(V)iew replies
(P)ack the responses

(X)ternal command
(Q)uit

Die ersten Punkte dürften einigermaßen klar sein, hier kann man die neuen Mails lesen, gezielt nach seiner Post suchen oder auch eine Auswahl aus den verschiedenen Brettern treffen. Wenn man dann mit "Enter" eine eigene Mail schreiben will, bekommt man eine Übersicht über die vorhanden Bretter, aus denen man das gewünschte aussuchen kann. Hier ist ein Beispiel aus der Graffiti-Box in München.

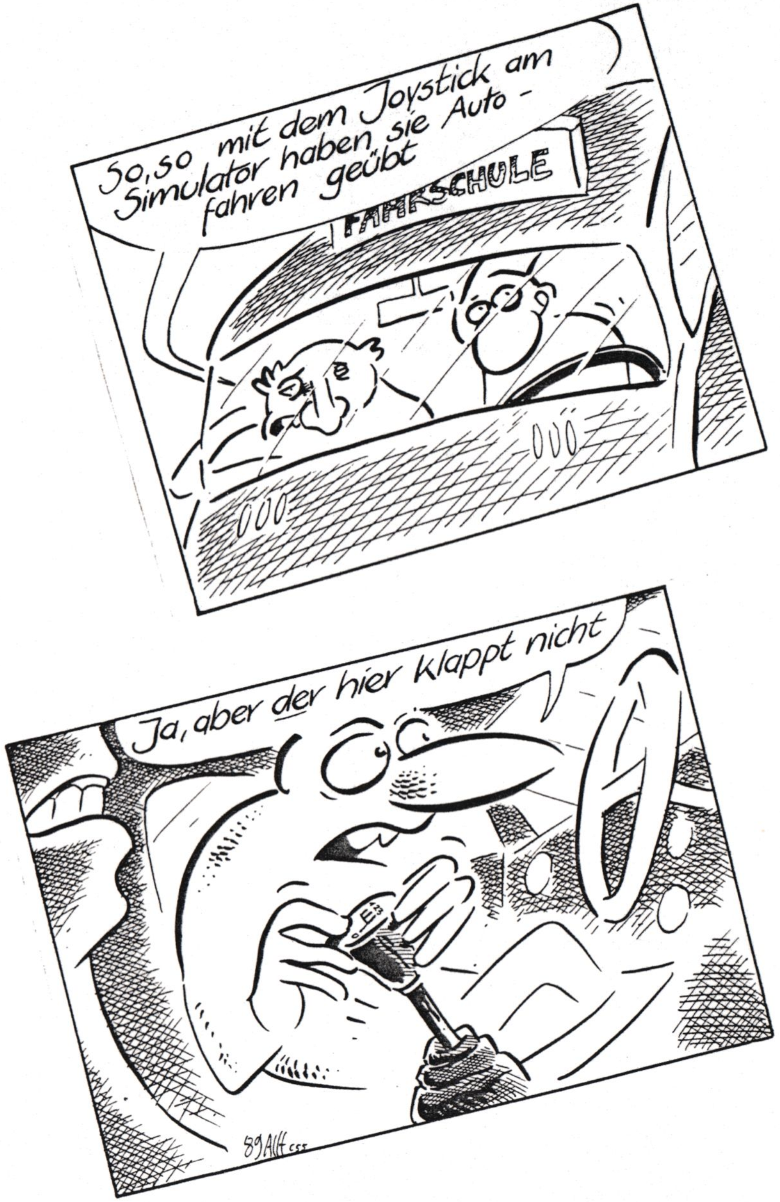
Num	Area Name	Num	Area Name	Num	Area Name
00	LOCAL	01	privat	05	Bazar.muc
08	DFUE.GER	09	disp	10	dtp.ger
11	ebbauser.g	12	fidosoft.g	13	filemgr.ge
14	*flea.ger	15	Frauen.ger	18	golded.ger
19	grafik.ger	20	Hardrock.g	21	hardware.g
22	ibm.ger	23	ibm.muc	24	kommerz.ge
26	lan.ger	27	medizin.ge	28	mmp.muc
29	msdos	30	novell.ch	33	printer.ge
34	request.ge	36	shareware.	37	soft.muc
39	storage.ge	40	superbbs.g	41	windows.ge
42	zyxel.ger	44	chauvi.ger	45	dcc.ger
46	erotika.ge	47	kontakt.ge	48	
49	novell.ger	50	os2.lan.ge	54	sex.ger
56	bazar.inte	57	btx.ger	58	cdrom.ger
59	coherent.g	60	comms.ger	62	cpm.ger
63	dbase.ger	64	dbridge.ge	67	fdecho.ger
68	fidoguide.	71	general.mu	72	graffiti.i
74	hs_modems.	76	kochen.ger	77	
79	modems.ger	80	modem_boer	81	msdos5xx.g
82	LAN.GER	83	os2.ger	87	pc_geos.ge

Which area or (Q)uit? (+ to see more)

Wenn man das Brett ausgewählt hat, muß man nur noch den Namen des Empfängers ("Harry Hirsch" oder auch einfach "alle") und das Subjekt angeben und landet automatisch im Editor. Wenn man dann mit allem fertig ist, geht man zum Punkt "Pack" und schließlich und endlich hat man auf der Floppy ein File mit der Extension .REP (für reply). Darin sind alle Antworten gepackt und man kann es in der Mailbox "abgeben".

Zwei Nachteile sollen aber nicht verschwiegen werden. Einmal muß in der Mailbox eine Software laufen, die Offline-Reader unterstützt, ggf. muß man den Sysop halt solange nerven, bis er sich dazu bereit erklärt. Der zweite "Nachteil" ist, daß das Programm Shareware ist. Einen Monat kann man das Programm ausprobieren, dann kommen immer wieder sehr lästige Anforderungen von Passwörtern, spätestens dann ist es an der Zeit, sich für 15 Pfund (rund 45 Mark) registrieren zu lassen. Ich denke, daß das Programm das auf alle Fälle wert ist, zumal es noch immer Updates gibt und man die letzten aktiven Softwareschreiber für CP/M nicht hängen lassen sollte.

Alexander Schmid



Kurs:

Mailbox-Bau Teil 1

Arbeiten mit Quickbbs

Mailboxen erfreuen sich seit ein paar Jahren großer Beliebtheit in der Szene der DFÜ-Freaks. Eine gute Mailbox bietet dem Benutzer eine Reihe von Serviceleistungen: beispielsweise Nachrichtenforen mit weltweitem Briefverkehr oder Dateibereiche, aus denen die aktuellste Shareware bezogen werden kann. Befinden Sie sich nur ein Ortsgespräch von Ihrer nächsten Mailbox entfernt, ist diese gewiß die kostengünstigste Gelegenheit, um in Sachen Branchengerüchte, Neuigkeiten und Software »auf dem laufenden« zu bleiben.

Die Preise für schnelle Modems sind inzwischen stark gesunken. Deshalb entdecken immer mehr PC-Anwender die Datenfernübertragung (kurz DFÜ) als neues Hobby. Haben Sie erst einmal einige Monate als Benutzer einer Mailbox in die Welt der DFÜ »hineingeschnuppert«, möchten Sie vielleicht sogar selbst Ihre eigene Mailbox eröffnen.

Vor wenigen Jahren hat dies noch einiges an Programmieraufwand bedeutet, da fertige Mailbox-Software praktisch nicht existierte. Heute hingegen können Sie auf eine breite Palette fertiger Systeme zurückgreifen, die die Gestaltung einer Mailbox mit individueller Note sehr komfortabel machen.

Eines der am weitesten verbreiteten Programme ist »Quickbbs«, ein Shareware-Produkt, das bereits sämtliche Module für eine Mailbox enthält. Quickbbs beziehen Sie aus diversen Mailboxen sowie von Shareware-Anbietern meist als komprimierte Datei (beispielsweise mit der Dateinamenserweiterung ».zip« oder »lha«).

Zur Installation der Software müssen Sie lediglich die in der komprimierten Archivdatei enthaltenen Programm- und Datendateien »entpakken« und in ein leeres Unterverzeichnis auf der Festplatte kopieren (hierzu benötigen Sie die entsprechenden, meist mitgelieferten Dekompressionsprogramme wie beispielsweise »pkunzip.exe« oder »lha.exe«). Richten Sie sich das Verzeichnis »c:\qbbs« mit dem MS-DOS-Befehl »md c:\qbbs« ein, wechseln Sie mit »cd qbbs« in dieses Unterverzeichnis, und generieren Sie auf die gleiche Weise die Verzeichnisse »c:\qbbs\menus« und »c:\qbbs\txtfiles«.

Das Hauptverzeichnis »c:\qbbs« dient zur Aufnahme aller Dateien, die zu Quickbbs direkt gehören. Die Unterverzeichnisse »c:\qbbs\menus« und »c:\qbbs\txtfiles« werden zum Speichern der Menüs Ihrer Mailbox sowie der Textdateien, die in der Mailbox abgerufen werden können, verwendet.

Eine Mailbox-Grundkonfiguration benötigt einen »Logon-Screen«, der jeden Anrufer nach seinem Benutzernamen und dem zugehörigen Paßwort befragt. Quickbbs erledigt diese Aufgabe alleine, so daß Sie sich nicht weiter darum zu kümmern brauchen. Ein weiteres wichtiges Element einer Mailbox ist das Hauptmenü, von dem aus alle weiteren Funktionen der Mailbox aktiviert werden. Das Hauptmenü muß vom Betreiber der Mailbox – also von Ihnen – programmiert werden. Ebenso sind für einen Nachrichtenbereich und einen Dateibereich Untermenüs vonnöten, die Sie, als

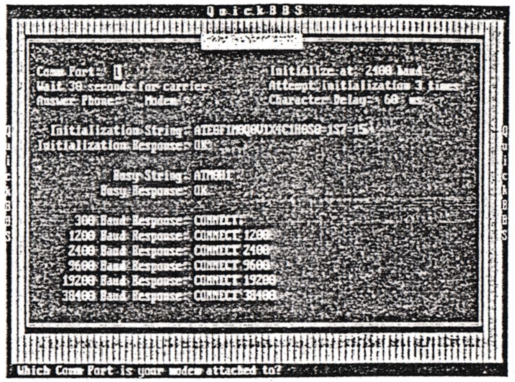


Bild 1. Das Konfigurationsprogramm »qconfig.exe« dient zur Einstellung der wichtigsten Parameter von »Quickbbs«

Sysop Ihrer Mailbox, selbst erzeugen müssen.

Die Programmierung von Quickbbs-Menüs geschieht auf recht einfache Art und Weise mit einem Menügenerator wie beispielsweise »mm.exe« (»Menu Maker«), der bei Quickbbs normalerweise mitgeliefert wird. Ein weiteres essentielles Zusatzprogramm für Ihre Mailbox ist ein »Usereditor«, mit dem Sie die Zugriffsrechte der Mailboxbenutzer, die Vergabe von Paßwörtern oder die Festlegung von Zeitlimits steuern. Ein geeignetes Programm ist »qufe.exe«, ebenfalls im Lieferumfang von Quickbbs. In der neuen Version 2.75 von Quickbbs tragen die beiden Hilfsprogramme die neuen Bezeichnungen »menuedit.exe« und »useredit.exe«.

Zur Grundkonfiguration von Quickbbs starten Sie zunächst das Installationsprogramm »qconfig.exe« (Bild 1). Die entscheidenden Menüpunkte von »qconfig.exe« sind »Messages«, »Files« und »General«, da hier die Grundeinstellungen für die Nachrichten- und Dateibereiche sowie für die Grundeinstellungen der Software (Modemansteuerung, Sicherheitsabfragen) vorgenommen werden.

Wählen Sie zunächst den Punkt »Messages« an, und verzweigen Sie im folgenden Menü zum Punkt »Message Boards«. In der dritten Bildschirmzeile finden Sie die Nummer des Nachrichtenbereichs, der nun editiert werden kann (»Message Area #1«). Die Parameter, die auf dem sich präsentierenden Bildschirm (Bild 2) individuell konfiguriert werden, sind nur zum Teil für die Funktion Ihrer Mailbox wichtig. Unbedingt ausfüllen sollten Sie die Punkte »Type«, »Kinds« und »Security«.

Unter »Type« geben Sie an, welche Art von Nachrichten im Nachrichtenbereich gezeigt wird. Zur Auswahl stehen »Netmail«, »Echomail« und »Local«. Die ersten beiden Bereiche dienen lediglich der Organisation des

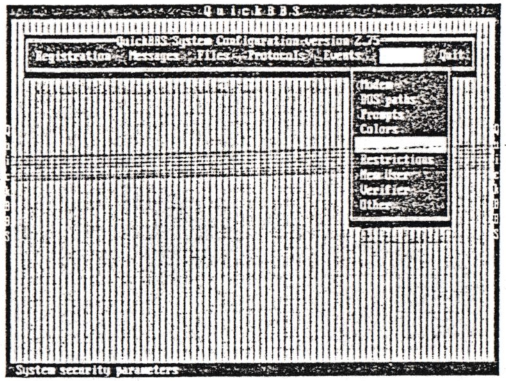


Bild 2. Die einzelnen Nachrichtenbereiche werden in diesem Menü konfiguriert

Nachrichtenverkehrs in Computernetzwerken. Im weltweiten »Fidonetz« beispielsweise sind »Netmails« private Nachrichten, »Echomails« hingegen öffentliche Mitteilungen, die von allen Interessenten bezogen werden können und meistens in Diskussionsforen zu einem bestimmten Thema gehören. Lokale Nachrichten werden nicht weitergeleitet und verbleiben auf der Mailbox, so daß nur die Benutzer der Mailbox Zugriff darauf haben.

Der Parameter »Kinds« gibt an, ob Mitteilungen im Nachrichtenbereich privat, öffentlich, privat und öffentlich oder »read only« sein sollen. Je nachdem, welche Funktion der von Ihnen gerade bearbeitete Nachrichtenbereich später haben soll, wählen Sie einen geeigneten Wert. Möchten Sie beispielsweise ein Diskussionsforum einrichten, in dem alle Benutzer Ihrer Mailbox miteinander Nachrichten öffentlich austauschen, wählen Sie die Einstellung »public«.

Wollen Sie dagegen einen Nachrichtenbereich, in dem einzelne Teilnehmer private Nachrichten an andere Mailboxbenutzer schicken, ist »private« der geeignete Parameter. Benötigen Sie einen Nachrichtenbereich, in dem nur Sie selbst Nachrichten schreiben – beispielsweise für Hinweise an die Benutzer Ihrer Mailbox – so stellen Sie »read only« ein.

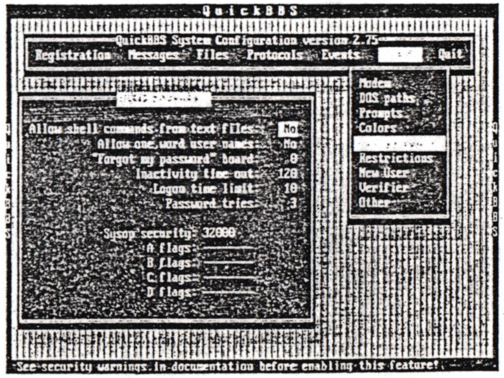


Bild 3. Im Filearea-Menü legen Sie fest, auf welche Programme die Mailboxbenutzer Zugriff haben

Der Bereich »Security« dient der Sicherheit in Ihrer Mailbox und ist daher besonders wichtig. Ein Anrufer, der noch keine Benutzerkennung (englisch »Account«) auf Ihrem System besitzt, erhält einen Anfangssicherheitscode (»Level«) von beispielsweise »5«. Eingetragene User bekommen dann die höhere Zugriffsberechtigung »20«, und Sie selbst als »Sysop« haben mit der höchsten Stufe »32000« Zugang. In die Security-Felder tragen Sie daher den Level ein, den ein Benutzer mindestens braucht, um im betreffenden Nachrichtenbereich Mitteilungen zu lesen (»read«) oder zu schreiben (»write«).

Zu Beginn richten Sie in Ihrer Mailbox verschiedene Mitteilungsbereiche ein. Vorausgesetzt, Sie sind an kein Computernetzwerk angeschlossen, sollte je ein Bereich für private Nachrichten der Benutzer untereinander (»private«, »local«) sowie ein öffentliches Forum (»public«, »local«) existieren. Benutzen Sie die Tasten [Bild aufwärts] und [Bild abwärts], um zwischen den verschiedenen Nachrichtenbereichen (es können maximal 200 sein) hin- und herzuschalten.

Notieren Sie sich jeweils die Nummer jedes Nachrichtenbereichs (beginnend bei »Message Area #1«), und »taufen« Sie jeden Bereich unter dem Punkt »Area Name« mit einer passenden Bezeichnung – beispielsweise »Briefe« für den privaten Nachrichtenbereich und »Diskussion« für den öffentlichen Bereich. Sind Sie an ein Computernetzwerk angeschlossen oder möchten Sie Diskussionsforen zu bestimmten Themen anbieten, so schaffen Sie einfach weitere Nachrichtenbereiche mit den entsprechenden Einstellungen.

Für den Menüpunkt »Files« von »qconfig.exe« gilt ähnliches wie für die Konfiguration von Mitteilungsbereichen. Der Unterschied ist lediglich, daß die hier vorzunehmenden Einstellungen für den Dateibereich bestimmt sind, aus dem Benutzer Programme holen (»downloaden«) oder auch dorthin schicken (»uploaden«).

Unter »Name« geben Sie eine passende Bezeichnung für den Datenbereich an – beispielsweise »PC-Utilities« für MS-DOS-Programme – und unter »File Path« das Unterverzeichnis, in dem die zum Dateibereich gehörenden Dateien gespeichert sind. Die Unterverzeichnisse müssen Sie auf der MS-DOS-Kommandozeile selbst einrichten.

Beachten Sie, daß Quickbbs nur dann mit allem verfügbaren Komfort arbeitet, wenn in jedem Unterverzeichnis im Dateibereich eine Datei »files.bbs« existiert, die alle im Unterverzeichnis enthaltenen Dateien mit Namen und einer kurzen Erläuterung enthält. Die Syntax für einen Eintrag in »files.bbs« lautet »Dateiname Erklärung«, also beispielsweise

```
qbbs275.zip
QuickBBS Version
2.75
```

Richten Sie wie bei den Nachrichtenbereichen mehrere Dateibereiche für Ihre Mailbox ein – beispielsweise »MS-DOS-Programme«, »DFÜ-Software« und »Computerspiele« –, und fügen Sie bei der Konfiguration in der Sparte »Download Security« jeweils den Mindest-Zugriffs-Level ein, den ein Anwender haben muß, um Dateien aus dem jeweiligen Bereich downloaden zu dürfen.

Die Einträge »Ignore K Limit« und »Ignore Ratios« benötigen Sie nur, wenn Sie den Benutzern Ihrer Mailbox ein Maximum pro Tag (beispielsweise maximal 500 KByte) für das Downloaden von Programmen einräumen möchten. Dies ist dann sinnvoll, wenn einige Anrufer den Telefonanschluß Ihrer Mailbox ständig lange belegen und andere Interessenten nicht mehr durchkommen.

Eine Alternative zur Benutzung des Dateimenüs von »qconfig.exe« ist die Textdatei »flsearch.ctl«, die Sie mit Ihrem ASCII-Texteditor er-

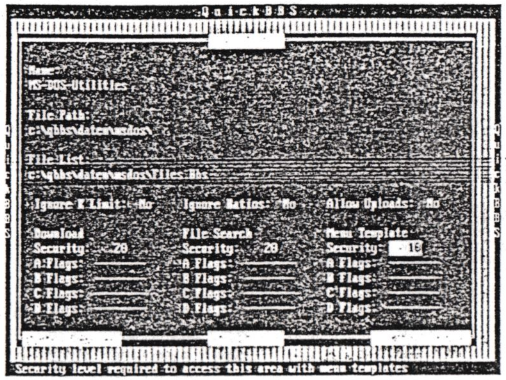


Bild 4. Um das Zusammenspiel mit dem Modem zu organisieren, stellen Sie hier alle erforderlichen Parameter ein

zeugen. Diese enthält jeweils zeilenweise Einträge, die das Unterverzeichnis, den Zugriffs-Level sowie den Namen aller verfügbaren Dateibereiche enthält, beispielsweise

```
c: \daten \utils
20 MS-DOS-Utilities
c: \daten \dfue
20 DFUe-Programme
c: \daten \spiele
20 Computerspiele
```

Sie sollten »flsearch.ctl« auf jeden Fall erzeugen, weil dann ohne explizite Angabe eines Datenbereichs auf beliebige Dateien aus den angegebenen Unterverzeichnissen zugegriffen werden kann.

Der dritte Menüpunkt von »qconfig.exe« ist »General« und dient der Einstellung diverser Systemparameter. Unter »Modem« finden Sie ein Konfigurationsmenü, in dem Sie alle wichtigen Angaben zu Ihrem Modem wie beispielsweise die Nummer der verwendeten seriellen Schnittstelle, die Übertragungsgeschwindigkeit sowie die Befehlssequenzen zum Initialisieren des Modems vornehmen müssen. Sehen Sie im technischen Handbuch Ihres Modems nach, welche Einstellungen für Ihr Gerät nötig sind.

Unter »DOS paths« teilen Sie Quickbbs alle Unterverzeichnisse mit, in denen wichtige Dateien zu finden sind. Der »Menu Path« muß »c:\qbbs\menus\« enthalten, da hier – wie zu Beginn festgelegt – die Menüs Ihrer Mailbox gespeichert werden sollen. Unter »Text File Path« (hier: »c:\qbbs\txtfiles\«) sucht Quickbbs später nach Textdateien.

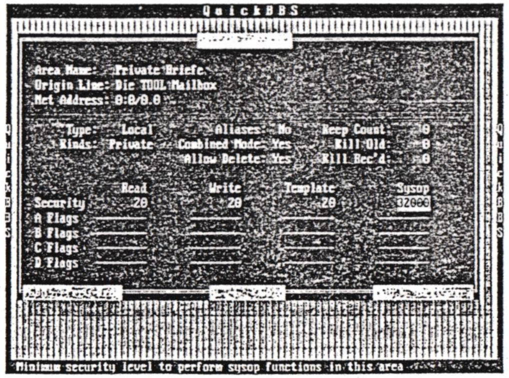


Bild 5. Um Hackern das Leben schwer zu machen, sollten Sie besonderes Augenmerk auf das Security-Menü legen

»Message Base Path« gibt an, in welchem Verzeichnis die »Message Base« zu finden ist. »Message Base« ist eine Datei, die sämtliche (private und öffentliche) Nachrichten Ihrer Mailbox enthält. Geben Sie hier »c:\qbbs\« an, um das Quickbbs-Hauptverzeichnis zum Speichern der »Message Base« zu verwenden. Den Eintrag »Nodelist Path« benötigen Sie nur, wenn Ihre Mailbox Teilnehmer in einem Computernetzwerk ist.

Verwenden Sie die spezielle »Overlay-Version« von Quickbbs, müssen Sie unter »Overlay Path« nochmals das Quickbbs-Hauptverzeichnis »c:\qbbs\« anführen.

Ein sehr wichtiger Menüpunkt ist weiterhin »Security«: Hier geben Sie an, wie viele Fehlversuche ein Anrufer bei der Angabe seines Paßworts (»Password tries«) und welchen Zugriffs-Level der Sysop – also Sie – hat (»32000« ist der Maximalwert). Ebenfalls zum Bereich »Sicherheit« gehört das Menü »New User«, in dem Sie den Standard-Zugiffs-Level eines neuen Anrufers (»Security Level«) angeben. Wählen Sie hier nicht die Null, da dies dazu führt, daß neue Anrufer nicht zugelassen werden und Ihr Modem sofort die Verbindung unterbricht. Haben Sie alle Einstellungen vorgenommen, wählen Sie den Menüpunkt »Quit« und speichern anschließend Ihre Quickbbs-Konfiguration mit »Save«.

Die ersten Schritte sind nun getan. Bis zum ersten Testlauf fehlen nun lediglich die Mailbox-Menüs, über die ein Anrufer alle Funktionen der Mailbox steuert.

Im nächsten Teil erfahren Sie, welche Menüpunkte für ein reibungsloses Funktionieren Ihrer Mailbox notwendig sind und wie diese mit »mm.exe« (beziehungsweise »menuedit.exe«) programmiert werden.

Thomas Rueß/ru

Erstellung einer Terminalbeschreibung

Am Beispiel Genie IIIs unter Kämpf CP/M 3.0

Egbert Schröer, November 1992

1. Vorwort

Nachdem nun in einigen CLUB INFO´s schon über Z3PLUS und auch die Erstellung einer TCAP - Info 32, Seite 35 ff, F.Chwolka - geschrieben wurde, möchte ich diese mit folgendem Artikel ergänzen und damit allen Neulingen die Erstellung einer eigenen Terminal-Beschreibung unter Z3PLUS erleichtern. Das User Manual sagt zu diesem Thema leider nichts und hinkt der Entwicklung etwas hinterher.

2. Terminal unter Z3PLUS

Als Standard-Installtion wird bei Kauf von Z3PLUS das Programm TCSELECT mit einer aktuellen TCAP Library ausgeliefert. Damit wird das Terminal-Capability-File (1) für das entsprechende CP/M-System bzw. den Computer-Typ ausgewählt. Beim CP/M 3.0 von Kämpf wird beispielsweise ein Televideo 920 Terminal emuliert.
Wenn man die Grafikfähigkeit des Genie IIIs unter Gdos kennt ist diese Installation mehr als mager, da ja auch Z3PLUS Grafik verarbeiten kann.

3. VLIBxx - Grafikfähigkeit unter Z3PLUS

VLIB ist eine Library von Routinen ursprünglich von Richard Conn geschrieben.
Inzwischen wurden viele Änderungen und Ergänzungen durchgeführt.
VLIB liegt jetzt in der Version 4.2d vor (2).
Programme, die diese Routinen nutzen, sind in der Lage das Video Display um die Funktionen

- o direkte Cursor-Adressierung
- o spezielle Video Eigenschaften
- o Pull-Down Menüs

zu erweitern.
Dazu müssen Informationen über den Rechner und/oder die verwendete CP/M 3.0 Implementation vorliegen (z.B. Character-Set etc.).

4. Erstellung eines Source Codes für das Terminal

Mit dem Programm TCSRC14.COM (3) erzeugt man - nachdem mit TCSELECT ein Terminal gewählt wurde - einen kommentierten Z80 Source Code des TCAP im Environment. Dieser File wird als Default mit Z3TCAP.Z80 benannt und sieht für das Televideo 920 Terminal wie folgt aus:

```
;
; Z3TCAP file:   Z3TCAP.Z80
;
ESC      EQU      27                    ; Escape character
;
; The first character in the terminal name must not be a space.  For
; Z3TCAP.TCP library purposes only, the name terminates with a space
; and must be unique in the first eight characters.
;
TNAME:  DB       ´TVI920          ´ ; Name of terminal (13 chars)
```



```
;
GOFF:    DB       GOELD-TNAME       ; Graphics offset from Z3TCAP start
;
;  Terminal configuration bytes B14 and B15 are defined and  bits
;  assigned
; as follows.  The remaining bits are not currently assigned.  Set
;  these
; bits according to your terminal configuration.
;
;        B14 b7: Z3TCAP Type.... 0 = Standard TCAP  1 = Extended TCAP
;
;        bit:    76543210
B14:     DB      00100000B          ; Configuration byte B14
;
;     B15 b0: Standout....... 0 = Half-Intensity 1 = Reverse Video
;     B15 b1: Power Up Delay. 0 = None           1 = Ten-second delay
;     B15 b2: No Auto Wrap... 0 = Auto Wrap      1 = No Auto Wrap
;     B15 b3: No Auto Scroll. 0 = Auto Scroll    1 = No Auto Scroll
;     B15 b4: ANSI........... 0 = ASCII          1 = ANSI
;
;        bit:    76543210
B15:     DB      00100000B          ; Configuration byte B15
;
; Single character arrow keys or WordStar diamond
;
         DB      ´K´-40H            ; Cursor up
         DB      ´J´-40H            ; Cursor down
         DB      ´L´-40H            ; Cursor right
         DB      ´H´-40H            ; Cursor left
;
; Delays (in ms) after sending terminal control strings
;
         DB      50                 ; CL delay
         DB      0                  ; CM delay
         DB      0                  ; CE delay
;
; Strings start here
;
CL:      DB      ´Z´-40H,0          ; Home cursor and clear screen
CM:      DB      ESC,´=%+ %+ ´,0    ; Cursor motion macro
CE:      DB      ESC,´T´,0          ; Erase from cursor to end-of-line
SO:      DB      ESC,´)´,0          ; Start standout mode
SE:      DB      ESC,´(´,0          ; End standout mode
```

Ab hier werden die eigenen Terminal Definitionen statt 0 eingefügt.

```
TI:      DB      0                  ; Terminal initialization
TE:      DB      0                  ; Terminal deinitialization
;
; Extensions to standard Z3TCAP
;
LD:      DB      0                  ; Delete line at cursor position
LI:      DB      0                  ; Insert line at cursor position
CD:      DB      0                  ; Erase from cursor to end-of-screen
;
; The attribute string contains the four command characters to set
; the following four attributes for this terminal in the following
; order:         Normal, Blink, Reverse, Underscore
```

```
SA:      DB      0                   ; Set screen attributes macro
AT:      DB      0                   ; Attribute string
RC:      DB      0                   ; Read current cursor position
RL:      DB      0                   ; Read line until cursor
;
; Graphics TCAP area
;
GOELD:   DB      0                   ; Graphics On/Off delay in ms
;
; Graphics strings
;
GO:      DB      0                   ; Graphics mode On
GE:      DB      0                   ; Graphics mode Off
CDO:     DB      0                   ; Cursor Off
CDE:     DB      0                   ; Cursor On
;
; Graphics characters
;
GULC:    DB      0                   ; Upper left corner
GURC:    DB      0                   ; Upper right corner
GLLC:    DB      0                   ; Lower left corner
GLRC:    DB      0                   ; Lower right corner
GHL:     DB      0                   ; Horizontal line
GVL:     DB      0                   ; Vertical line
GFB:     DB      0                   ; Full block
GHB:     DB      0                   ; Hashed block
GUI:     DB      0                   ; Upper intersect
GLI:     DB      0                   ; Lower intersect
GIS:     DB      0                   ; Mid intersect
GRTI:    DB      0                   ; Right intersect
GLTI:    DB      0                   ; Left intersect
;
;  Fill remaining space with zeros
;
          REPT   128-($-TNAME)
          DB     0
          ENDM

         END
;
; End of Z3TCAP
;
```

Für das Genie IIIs unter Kämpf CP/M 3.0 sieht der geänderte Source Code nun wie folgt aus:

```
;
; Z3TCAP file:  G3SK.Z80
;
ESC      EQU     27                  ; Escape character
;
; The first character in the terminal name must not be a space.  For
; Z3TCAP.TCP library purposes only, the name terminates with a space
; and must be unique in the first eight characters.
;
TNAME:   DB      'G3S-KAEMPF    '    ; Name of terminal (13 chars)
;
GOFF:    DB      GOELD-TNAME         ; Graphics offset from Z3TCAP start
```



```
;
;  Terminal configuration bytes B14 and B15 are defined and  bits
;  assigned
; as follows.  The remaining bits are not currently assigned.  Set
;  these
;  bits according to your terminal configuration.
;
;       B14 b7: Z3TCAP Type.... 0 = Standard TCAP  1 = Extended TCAP
;
;       bit:    76543210
B14:    DB      10000000B       ; Configuration byte B14
;
; B15 b0: Standout....... 0 = Half-Intensity 1 = Reverse Video
; B15 b1: Power Up Delay. 0 = None           1 = Ten-second delay
; B15 b2: No Auto Wrap... 0 = Auto Wrap      1 = No Auto Wrap
; B15 b3: No Auto Scroll. 0 = Auto Scroll    1 = No Auto Scroll
; B15 b4: ANSI........... 0 = ASCII          1 = ANSI
;
;       bit:    76543210
B15:    DB      00000001B       ; Configuration byte B15
;
; Single character arrow keys or WordStar diamond
;
        DB      'K'-40H         ; Cursor up
        DB      'V'-40H         ; Cursor down
        DB      'L'-40H         ; Cursor right
        DB      'H'-40H         ; Cursor left
;
; Delays (in ms) after sending terminal control strings
;
        DB      0               ; CL delay
        DB      0               ; CM delay
        DB      0               ; CE delay
;
; Strings start here
;
CL:     DB      ESC,'*',0       ; Home cursor and clear screen
CM:     DB      ESC,'=%+ %+ ',0 ; Cursor motion macro
CE:     DB      ESC,'T',0       ; Erase from cursor to end-of-line
SO:     DB      ESC,')',0       ; Start standout mode
SE:     DB      ESC,'(',0       ; End standout mode
TI:     DB      0               ; Terminal initialization
TE:     DB      0               ; Terminal deinitialization
;
; Extensions to standard Z3TCAP
;
LD:     DB      ESC,'R',0       ; Delete line at cursor position
LI:     DB      ESC,'E',0       ; Insert line at cursor position
CD:     DB      ESC,'Y',0       ; Erase from cursor to end-of-screen
;
; The attribute string contains the four command characters to set
; the following four attributes for this terminal in the following
; order:        Normal, Blink, Reverse, Underscore
;
;  I had no informations about the screen attribute macro  of
; Kämpf CP/M ; 3.0.
; I wrote to Klaus Kämpf for more informations and for the source
; codes in september 1992, but there was no response until now
; (February 1993).
```

```
; If anybody is able to complete this TCAP please inform me or call:
; Egbert Schröer
; Joachimstraße 18
; W-4270 Dorsten
; Germany
; Phone: private: 02362/75311
;          on business: 02365/49/9649
;
SA:     DB      0               ; Set screen attributes macro
AT:     DB      0               ; Attribute string
RC:     DB      0               ; Read current cursor position
RL:     DB      0               ; Read line until cursor
;
; Graphics TCAP area
;
GOELD:  DB      0               ; Graphics On/Off delay in ms
;
; Graphics strings
;
GO:     DB      ESC,'$',0       ; Graphics mode On
GE:     DB      ESC,'%',0       ; Graphics mode Off
CDO:    DB      ESC,'.0',0      ; Cursor Off
CDE:    DB      ESC,'.4',0      ; Cursor On
;
; Graphics characters
;
GULC:   DB      86H             ; Upper left corner
GURC:   DB      87H             ; Upper right corner
GLLC:   DB      85H             ; Lower left corner oK
GLRC:   DB      88H             ; Lower right corner oK
GHL:    DB      8BH             ; Horizontal line oK
GVL:    DB      8AH             ; Vertical line oK
GFB:    DB      160D            ; Full block oK
GHB:    DB      7FH             ; Hashed block Testing
GUI:    DB      8EH             ; Upper intersect oK
GLI:    DB      8FH             ; Lower intersect oK
GIS:    DB      89H             ; Mid intersect oK
GRTI:   DB      8CH             ; Right intersect oK
GLTI:   DB      8DH             ; Left intersect oK
;
;  Fill remaining space with zeros
;
        REPT    128-($-TNAME)
        DB      0
        ENDM

        END
;
; End of Z3TCAP
;
```

Da sich Herr Kämpf leider bezüglich Informationen zu seinem CP/M bedeckt hält - trotz mehrmaliger Anschreiben - , konnten die Character Sets nur mit dem Programm CHARSET ermittelt werden. Weiter Informationen liegen zum Terminal nicht vor, bzw. sind aus der Dokumentation zum CP/M 2.2, erschienen bei Röckrath, zu entnehmen.
Die Disketten Sage xxx bekommt man bei dem nächsten Verteiler oder direkt bei mir.

Egbert

(1) siehe User Manual
(2) siehe auch VLIB-Help Files auf Sage 41-Disk
(3) Sage 38-Disk

WDR Grafik Files und HRG Model 1 Grafik Files

Egbert Schröer, Februar 1993

Im Laufe der Jahre und aus guten alten Computerclub Zeiten haben sich doch bei dem ein oder anderen Grafik Files angesammelt, die man nicht mehr missen möchte. Sollen die verloren gehen, nur weil man den Rechner wechselt ? Natürlich nicht. Im folgenden zwei Basic Programme zum Einlesen von WDR Grafik-Files und HRG-Grafik-Files im Model 1 Format in die Grafik des Genie IIIs.
Bei den Grafik Files im WDR Format sind einige Informationen verborgen, die man zur optimalen Darstellung auf der Grafik des Genie IIIs heranziehen kann.
Dies wird in der Unterroutine "Monitor einstellen" erledigt.
Zur gefälligen Beachtung auch die Error Routine zur Fehlerbehandlung.

```
100 REM ************************************
110 REM BILD 2 (LESER) *
120 REM FüR Genie IIIs  *
130 REM MIT RDL-Basic *
140 REM W.BACK MAI 85 geä. 5/92 E.Schröer *
150 REM ************************************
160 CLEAR 1000 : ON ERROR GOTO 65010 : CLS : HCLS : HON
170 DEFINT A-Z : DEFFNAT$(Z,S)=CHR$(28)+STRING$(Z,26)+
    STRING$(S,25) : NG$=CHR$(22)
180 GOSUB 8010 : 'Titel
190 PRINT "DATEINAME : ";:LINE INPUT NAM$
200 PRINT"LAUFWERK : ";:LINE INPUT LAUF$
210 OPEN"I" , 1 , NAM$ + ":" + LAUF$ : CLS
220 LINE INPUT #1,N$ : IF N$<>"BILD"AND N$<>"BILD" THEN 220
230 LINE INPUT #1,R$
240 LINE INPUT #1,V$ : VE% = VAL(MID$(V$,2,3))
250 LINE INPUT #1,H$ : HO% = VAL(MID$(H$,2,3))
260 LINE INPUT #1,D$ : LINE INPUT #1,D$
265 GOSUB 5010
270 FOR V = 0 TO VE%
280 IF EOF(1) THEN CLOSE : GOTO 400
290 LINE INPUT #1 , A$ : X = 1
300 FOR H = 0 TO HO% STEP 6 : X = X + 1
310 A = ASC(MID$(A$,X,1)) - 32 : IF A = 0 THEN 380
312 V2% = VE% - V
320 IF A AND 1 THEN  HPLOT H ,V2%,2
330 IF A AND 2 THEN  HPLOT H+1,V2%,2
340 IF A AND 4 THEN  HPLOT H+2,V2%,2
350 IF A AND 8 THEN  HPLOT H+3,V2%,2
360 IF A AND 16 THEN HPLOT H+4,V2%,2
370 IF A AND 32 THEN HPLOT H+5,V2%,2
380 NEXT H , V
400 IF INKEY$="" THEN 400
500 END
5000 REM Monitor einstellen
5010 IF HO% < VE% THEN 5020 ELSE 5040
5020 VE%=VE%-1 : IF HO%-VE% < 0 THEN 5020
5030 CMD"##,h" : RETURN
```

```
5040 IF HO% = VE% THEN CMD"##,v" : RETURN
5050 IF HO% > 511 AND VE% > 239 THEN CMD"##,x" : RETURN
5060 IF HO% < 512 AND VE% < 240 THEN CMD"##,h" : RETURN
5070 RETURN
8000 REM Zeichensatz laden und Programmtitel ausgeben
8010 CMD"z,y" : HCODE"script/rzs"
8020 CMD"##,h" : HCLS : Z=63 : S=6
8030 HBIG 55,220,1,2,1,"Bild - Lader"
8040 HTEXT 45,150,2,"nach einer Idee von V. Back"
8050 HTEXT 60,100,2,"WDR - Computer - Club"
8060 HCODE "normal/rzs"
8070 FOR X=0 TO 2500 : NEXT X
8080 HCLS : RETURN

6000 ´ hier steht Routine "File auswählen"

7000 ´ hier steht Routine "Info zu WDR-File"
7010 ´ Info zu Files werden ausgegeben:
       HON 0,1
       H$ -> horizontal
       V$ -> vertikal
       N$ -> Datei-Art
       R$ -> Info zu Bild (Rechner Typ)

65000 ´ Error Routine
65010 IF V>VE% THEN RESUME 400
65020 IF ERR/2+1=5 THEN RESUME 380
65030 X=1
65040 FE$(1)="Fehler #"+STR$(ERR/2+1)+" in Zeile"+STR$(ERL)+
      " (bitte notieren)"
65050 FE$(2)="Eingabe : 1=nochmal, 2=weitermachen, 3=Neustart,
      4=Menü"
65060 PRINT FNAT$(Z,S);CHR$(30);NG$;FE$(X);NG$;CHR$(7);
65070 GOSUB 65080 : GOTO 65060
65080 FOR Y=1 TO 100 : X$=INKEY$ : XX=VAL(X$)
65090 IF XX>=1 AND XX <=4 THEN PRINT FNAT$(Z,S);CHR$(30);
65100 IF XX=1 THEN RESUME ELSE IF XX=2 THEN RESUME NEXT
65110 IF XX=3 THEN 160 ELSE IF XX=4 THEN 65140
65120 NEXT Y : IF X=1 THEN X=2 ELSE X=1
65130 RETURN
65140 CLOSE : RUN"menu/bas"
65150 RUN
```

Und als Listing 2 das Programm zum <u>direkten</u> Einlesen von Grafik Files im Model 1 HRG Format.

```
10 REM ************************************
20 REM Model 1 HRG 1b - Bildleser
30 REM für Genie 3s
40 REM Hartmut Obermann
45 REM geändert für Genie 3s 05.92 E. Schröer
50 REM ************************************
60 CLEAR 20000 : CLS :HCLS
70 DEFINT C-Z : DIMA$(191)
80 PRINT:PRINTTAB(10)"* * * * * HRG 1b - L A D E R * * * *":
   PRINT
90 PRINT"Mit diesem Programm können HRG-Bilder des Model 1 (HRG 1b)
   in die Genie 3s - HRG geladen und dann abgespeichert werden!":
   PRINT
100 PRINT "Dateiname : ";:LINE INPUT NAM$
110 PRINT"Laufwerk : ";:LINE INPUT LAUF$
120 OPEN"R" , 1 , NAM$ + ":" + LAUF$
130 FIELD 1, 64 AS B$(0), 64 AS B$(1), 64 AS B$(2), 64 AS B$(3)
140 FOR S1=0 TO 191 STEP 4 : GET 1
150 FOR S2=0 TO 3
160 A$(S1+S2)=B$(S2)
170 NEXT S2 : NEXT S1
180 CLOSE : CLS : HON
190 I=-1
200 FOR V1=11 TO 0 STEP -1
210 FOR V2=0TO191 STEP 12
220 V=240-V2+V1
230 X=0 : I=I+1
240 FOR H = 0 TO 383 STEP 6 : X = X + 1
250 A = ASC(MID$(A$(I),X,1))
260 IF A AND 1 THEN HPLOT H ,V,2
270 IF A AND 2 THEN HPLOT H+1,V,2
280 IF A AND 4 THEN HPLOT H+2,V,2
290 IF A AND 8 THEN HPLOT H+3,V,2
300 IF A AND 16 THEN HPLOT H+4,V,2
310 IF A AND 32 THEN HPLOT H+5,V,2
320 NEXT H : NEXT V2 : NEXT V1
330 HOFF : PRINT"Bild geladen!"
340 PRINT"Unter welchem Namen soll das Bild gespeichert werden?"
350 PRINT"<NAME/EXT:DNR> "; : LINEINPUT N$
360 CLS : HON : HSAVE N$
370 HOFF : PRINT"Bild gespeichert!" : END
```

Das war´s.

Bitte an alle Grafik Fan´s mit MSDose: Schreibt doch mal eine Routine zum Abspeichern von Grafiken auf dem PC im WDR-Format !!

Egbert

ATOnce, ATTwice...

Da hat man nun einen schönen Rechner mit allem Schnick-Schnack, kann sich also nicht beklagen und dennoch hat die "Liebe Seele" keine Ruh'.

Wovon ist die Rede ?..., natürlich vom schielen ins PC-Lager schlechthin. Bekanntlich sind ja die Rechner der anderen Sparte immer die besseren und es reizt stets auch hier "mitzureden". In diesem Punkt waren und sind die ATARI-Rechner schon immer eine Art Camälion gewesen. CP/M-, C64-, ZX81- und seit längeren ja auch PC-Emulatoren für XT- und AT-Rechner. Da ich von Berufs wegen mit einem PC mein Auskommen finden muß, lag der Gedanke nahe, mir einen PC-Emulator anzuschaffen und so den Brückenschlag ins andere Lager zu vollziehen. Kostenmäßig war dieser Schritt weit günstiger wie die Anschaffung eines "richtigen" PC's, zumal Dinge wie Festplatte, Monitor etc. bereits vorhanden sind. Nicht zu unterschlagen der Gedanke, "gelegentlich" den einen oder anderen Hack auch zu Hause vornehmen zu können . Das Studium div. Testberichte in den ATARI-Zeitschriften ließ denn auch ein hinreichend positives Bild von der Sache entstehen. Immerhin konnte den Ergebnissen der Tester entnommen werden, daß alle Programme denen die konkrete Hardware schnuppe ist (hauptsache die DOS-Schnittstelle stimmt) auf einem so umgekrempelten Rechner laufen würden. Nach intensivem Anzeigenstudium (das Teil kostet immerhin ein paar Blaue) hatte ich dann den preisgünstigsten Anbieter herausgefiltert. Der Rest ist reine Fleißarbeit und etwas Geduld bis die Post das begehrte Teil in den Kasten wirft.

Praxistest...

Der Einbau gestaltet sich wirklich einfach, sofern man bereits bei anderer Gelegenheit die CPU auf einen Sockel gelupft hat. Ansonsten ist dieser Vorgang jetzt nachzuholen. Nach Installieren der Software (auch dieser AT braucht sein BIOS) kann der erste Start erfolgen...

Es ist schon faszinierend, wenn auf einem ATARI mit einem mal der Bootvorgang eines PC's abgespult wird (bevor das DOS von der Platten geladen werden kann, ist allerdings vorerst noch der Boot von Diskette angesagt, anschließend den Aufruf von FDISK und FORMAT auf die PLatte loszulassen und dann ...). Sobald der DOS-Prompt endlich erscheint, kann an die CONFIG.SYS und AUTOEXEC.BAT gedacht werden. Hier gibt es keine großen Probleme, man hat ja Vorkenntnisse, einige Male im Handbuch zum ATonce nachgesehen, desgleichen im DOS-Manual fleißig geblettert und gesucht, dann sind alle Angaben zusammengestellt. Mit ALT-CTRL-DEL die CPU aus ihrem Trott gerissen und diesmal das DOS von der Platte kratzen lassen. Wow..., es funktioniert, der DOS-Prompt erscheint wieder. Der Appetit kommt bekanntlich beim Essen, warum also nicht auch noch gleich Windows installieren ?

W-i-n-n-d-o-o-w-w-s-s-s...

Von der Idee bis zur Ausführung derselben stellt sich ersteinmal die Frage, wie man 1.44 MByte Disketten durch ein 720 KByte Laufwerk schleusen soll. Klar, geht so natürlich nicht ! Der findige Mensch weiß sich zu helfen, ein echter PC muß her (da soll doch ...). Da sich Windows bekanntlich ohne Murren von der Platte installieren läßt, ist nur dafür zu sorgen, daß alles dorthin geschaufelt wird. 1.44MB paßt aber nicht auf 720KB (womit eine Endlosschleife entstanden ist). Rettung bringt "EXIT IF PC=vorhanden".

Damit ich dem ersehnten Ziel, Windows auf meinem ATARI laufen zu sehen näher komme, sind auf der "Hilfshardware" alle Disketten abzulegen und anschließend per BACKUP auf ca. 13 Stück 720KB Disk umzubetten. Diese Datenkekse stecke ich dann meinem Laufwerk zur Vereinnahmung in den Schacht und es kann endlich losgehen (um bis hier her zu kommen bedarf es etwa 3 Tage, da ich meinen PC im Büro hierfür benutzen muß und nur gelegentlich Zeit zum Kopieren habe). Nach einem weiteren durchinstalliertem Abend kann ich endlich "WIN" eintippen. Das Logo macht einen etwas verkorksten Eindruck, wen wunderts, Farbe läßt sich auf einem reinen SW-Bildschirm (Marke TTL) nicht anders wiedergeben. Ab da erstrahlt dann aber der typische Windows-Look auf der Mattscheibe, ...faszinierend !!!

... und was nun ?

Ich hasse Fragen wie diese. Können sie doch so manchen "tollen Gedanken" mangels Anwendungsmöglichkeiten (wie z.B. nach einem erfolgten ruinösen Spontankauf) zum Absturz bringen. Aber für diesen Fall habe ich noch ein Trump im Ärmel: Autosketch ! Wem das nichts sagt, sei erklärt, daß ich häufig Stromlaufpläne erstellen muß und selbiges mit diesem Programm zu tun gedenke. Da kein Meister vom Himmel zu fallen pflegt, ist zuvor eine gewisse Einarbeitung von Nöten, was nichts anders bedeutet: installiere es auf deinem PC und arbeite damit ... (1.44MB -> 720KB etc.).

Das Lied von den Haken und Ösen...

Ein Punkt, der gelegentlich doch recht schmerzlich sein kann. Der Emulator ermöglicht zwar eine ganze Reihe von Videomodi (Herkules, Olivetti, EGA, VGA und CGA), bei allen aber muß man sich mit Einschränkungen der einen oder anderen Art zufrieden geben. In der Hauptsache ergeben sie sich aus dem Umstand, daß alles auf dem SW-Bildschirm des ATARI dargestellt werden muß, was gleichbedeutend ist mit: Was mit 640x400 Pixel auskommt wird dargestellt, der Rest muß entweder horizontal oder vertikal geschoben werden, will man den nicht dargestellten Bildschirmteil sehen. Unter Windows ist das kein so großes Handycap, es genügt das Fenster auf Bildschirmmaße einzustellen. Anders bei Programmen, die ohne diese Oberfläche arbeiten. Da kann es schon lästig werden mit dem links/rechts bzw. oben/unten. Gleicher Verdruß kann bei Programmen entstehen, die zum Betrieb ihren Dongel konsultieren wollen (mein diesbezüglicher Versuch Autocad zu fahren, scheiterte an diesem Sachverhalt), da die reale Hardware eben doch keinen "Industriestandard" wiederspiegelt. Klammert man diese "Spezialfälle/-probleme" einmal aus, hatte ich erstanlich wenige Hürden zu nehmen.

Und die Moral...

Die gibt es nicht, wohl aber eine Erkenntnis: Nicht alles was als "Standard" angesehen wird ist es auch. Bei der oben beschriebenen Aktion (und ich habe dabei noch nicht einmal alle Fallen, Herzrythmusstörungen und Ehrenrunden beschrieben) habe ich meinen ATARI schätzen gelernt, der beweißtdoch immerhin, daß es zum Glück auch noch andere Rechnerkonzepte gibt, die einfacher und damit benutzerfreundlicher angelegt sind. Im unmittelbaren Vergleich muß ich feststellen, daß, wäre da nicht die große Verbreitung der PC's, ein Rechnersystem wie das von ATARI vorzuziehen ist. Und das schon allein wegen des linearen Speicherkonzeptes. Keinerlei Kopfschmerzen über "expanded" oder "extended memory", "DOS=HIGH" und dergleichen. Man packt einfach die Speichererweiterung in den Rechner und fühlt sich wohl, gleichwohl welche Applikation gestartet wird ! Allein mit diesen PC-typischen Problemen habe ich mich eine ganze Weile auseinandersetzen müssen, nur um "meinen" optimal zu Konfigurieren. Auf der anderen Seite ist es erstaunlich, mit welch verhältnismäßig geringem Aufwand ein ansonsten inkompatibler Rechner in einen PC mit beachtlich hohem Kompatibilitätsgrad verwandelt werden kann. Trotzallem sollte aber jeder ATARIaner mit Seitenblick auf die PC-Welt lieber überlegen, ob er die Emulatorlösung wählt. Sobald ernstere Absichten bestehen, würden ich zum Umstieg auf einen "Echten" raten. Die inkaufzunehmenden Einschränkungen sind nicht zu vernachlässigen (Keine Farbe, keine Steckplätze, SW-Bildschirmformate werden nicht von jeder Software akzeptiert, leichte Inkompatibilitätsprobleme, u.U. gerade bei der Wunsch-Software, kein 1.44MB-Disklaufwerk (!!!) etc.). Wer unter Finanznot

..Hardware ..Hardware ..Hardware ..Hardware

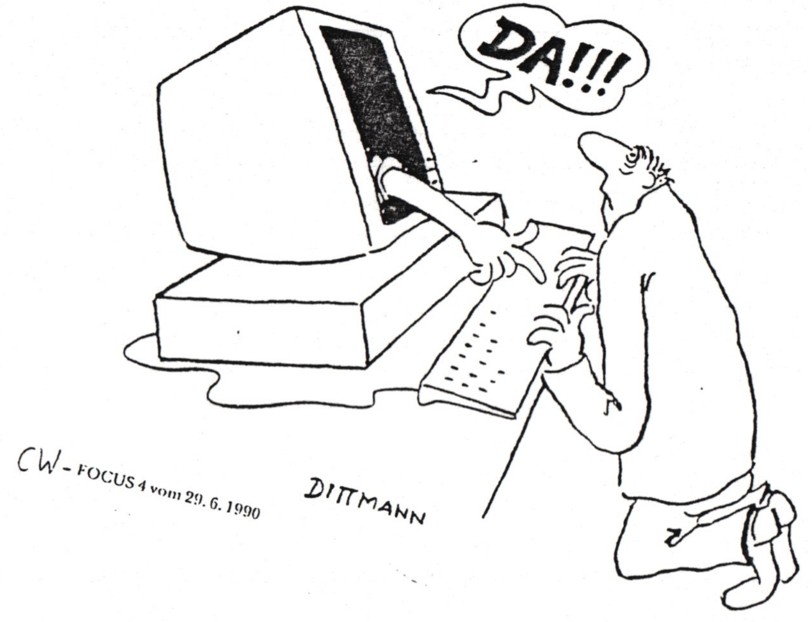
leidet und einfache Anwendungen wie z.B. Textverarbeitung auf dem PC betreiben will ohne gleich sein "altes Gerät" mit der ganzen vorhandenen Software verschrotten zu wollen, kann mit der Emulatorlösung durchaus eine gute Wahl treffen. Anzuraten ist dann aber ein Speicherausbau auf mindestens 4MB, da in der PC-Welt ansonsten kein Hut mehr zu gewinnen ist (bei Windows schon gar nicht). Ähnliches gilt für die Festplatte. Meine Erfahrung ist, daß jeder Gedanke mit einer 20MB MFM-Platte etwas werden zu wollen schon im Ansatz scheitern muß. Die Software-Pakete sind mittlerweile so umfangreich, daß derartige Platten sowohl zu klein als auch zu langsam sind. Windows z.B. benötigt komplett installiert gut 8MB. Im Protected Mode kommen dann noch einmal ca. 8MB als virtueller Speicher dazu. Damit wären dann bereits 16MB belegt. DOS selbst sollte man ebenfalls nochmals 4MB in Form einer eigenen Partition gönnen - macht in der Summe 20MB. Jede weitere Applikation müßte dann von Diskette gestartet werden. Ein unmöglich Ding. Ich habe für die DOS-Welt 52MB auf einer SCSI-Platte (m. 1MB/s u. 19ms Zugriffszeit) freigestellt und komme bei meinen Lieblings-Programmen damit gerade zurecht (als da wären: PSpice, Windows, MS-DOS 5, AUTOSKETCH 3, WinWord + div. Kleinkram mit wenig Speicherbedarf). Das im PC-Bereich 120MB-Platten aus genannten Gründen mittlerweile Standard sind ist überflüssig zu erwähnen.

8086 ... 80286 ... 80386 ...

Das höchste der Gefühle auf dem ATARI ist z.Zt. ein 80386SX mit 16MHz. Unter dem ist es m. E. auch nicht sinnvoll (Auf den 1040ST's aus Platzgründen nur bedingt einbaubar !). Alle anderen Emulatoren sind zu langsam. Nichts kann mehr stören, wie ein schleppendes Arbeiten am Rechner, nur weil dieser zu langsam ist. Der ATonce 386SX kommt laut Norton SI auf einen Index von ca. 16, was die Untergrenze in Sachen Tempo darstellt (meine Erfahrung jedenfalls). Wer zudem noch viel Numbercrunching betreiben will, sollte sich auch gleich über die Preise bei den Coprozessoren informieren, denn der scheint sich immer mehr zum Standardzubehör zu entwickeln (so'n Teil steckt immerhin in jeden 486, und die werden mittlerweile an jedem Bahnhofskiosk gehandelt - zumindest bis dieser Text im Info erscheint).

Bis dahin gilt wie immer: Nachbars Rasen ist immer am grünsten ...

Tschüß



Schneid-KLOPF-Verbinder

Nein, es handelt sich nicht um eine neue Erfindung, es geht noch immer um das leidige Problem mit den allseits beliebten Schneid-Klemm-Verbindern. Jeder, der schon mal einen Floppystecker aufpressen mußte, weiß, daß man dazu mindestens fünf Hände braucht. Entweder nimmt man einen Schraubstock oder eine Kombizange (die Spezialzange dafür leistet sich ja doch keiner) und vielleicht noch zwei Brettchen zum Unterlegen, damit man den Stecker nicht zu sehr deformiert. Überschlägig ergibt das: Stecker + Kabel + Zange + 2 Brettchen = 5 Hände. Es geht aber auch wesentlich einfacher, nämlich mit einem kleinen (ca. 100g) Hämmerchen. Mit einer Hand kann man das Kabel im Stecker einklemmen, legt ihn dann mit der Rückseite nach oben auf eine flache harte Unterlage und klopft ihn mit vorsichtigen Schlägen zusammen. Wenn man nicht gerade einen Vorschlaghammer nimmt, geht das sehr gut und wenn nach Möglichkeit alles eben auf dem Tisch liegt und nicht irgendwo schief runterhängt, ist es gar keine große Kunst. Ich habe das inzwischen bei mindestens 10 Steckern praktiziert und noch keinen in die ewigen Byte-Gründe geschickt.

Alexander Schmid

Wa-Tor

Die kybernetische Saga vom Planeten Wa-Tor
oder: Wie man die Populationsdynamik eines Ökosystems auf dem Computer simuliert.

Von A. K. Dewdney

Irgendwo, in einer Richtung, die man nur spaßwärts nennen kann, und einer Entfernung, der Ihre Programmierkünste allein Grenzen setzen, schwimmt der Planet Wa-Tor im Sternenmeer. Er hat die Gestalt eines Torus, sprich: Autoreifens, und ist ganz von Wasser bedeckt. Bewohnt wird er von Fischen und Haien – so genannt, weil sie diesen Erdbewohnern am meisten ähneln. Die Haie Wa-Tors ernähren sich von den Fischen, während die Nahrung der Fische nicht bekannt, aber offenbar stets im Überfluß vorhanden ist.

Dieses schlichte Ökosystem könnte stabil, ja sogar gähnend langweilig erscheinen. In Wahrheit aber unterliegt die Populationsgröße von Fischen und Haien extremen Schwankungen. Schon oft in der Vergangenheit waren die Fische nahezu ausgerottet, während ein andermal die Haie vor dem Aussterben standen (auch wenn es reichlich Fische gab). Und dennoch – Haie und Fische überlebten. Um herauszufinden wie das kam, entwarf ich ein Computerprogramm, das ihr Freß- und Fortpflanzungsverhalten simulieren sollte.

Bevor ich allerdings selbst Zeuge dieser ökologischen Rhythmen am Bildschirm wurde (Bild 1), grübelte ich lange über die Regeln und die Details des WATOR-Programms nach. In solche Überlegungen vertieft, fand ich mich eines Tages beim Mittagessen am gleichen Tisch mit David Wiseman, dem Computer-Hexenmeister in meiner Abteilung an der Universität von Western Ontario. Nachdem ich ihm mein Projekt beschrieben hatte, sah ich, daß Magi (so Wisemans Spitzname bei uns) geheimnisvoll lächelte. Am nächsten Morgen führte er mich stolz in sein Büro.

„Schau", sagte er, und drückte auf eine Taste an seinem Terminal. Auf dem Bildschirm erschien ein zufällig verteilter Haufen von Fischen und Haien, die langsam und scheinbar ohne System von Punkt zu Punkt irrten. Einige Haie bekamen nichts zu fressen und verschwanden. Andere hatten Nachkommen, die genauso gefräßig waren wie sie selbst. Einige Fische, die das Glück hatten, sich in einer Gegend aufzuhalten, in der gerade keine Haie lauerten, vermehrten sich zu einem großen Schwarm. Alsbald wurde dieser jedoch von ein paar Haien entdeckt, die sich an seinem Rand zusammenrotteten und sich ein Stück in ihn hineinfraßen. Ein paar Minuten später resümierte Magis Programm den Stand der Dinge: Es gab nun 578 Fische und 68 Haie.

Jemand schaute in Magis Büro und huschte wieder eilig davon. Es dauerte keine fünf Minuten, und der Raum war voller Leute, die den Haien zujubelten. Langsam umzingelte eine Phalanx dieser Räuber die unglücklichen Fische. An einer anderen Stelle des Schirms wuchs indessen unbemerkt erneut ein kleiner Fischschwarm heran. Unter dem bedauernden Gemurmel der Zuschauer wurde der große Schwarm schließlich niedergemetzelt. Die Haie waren nun ohne Beute, und ziellos umherirrend, starb einer nach dem anderen. Ich dachte schon daran, die Regeln so zu ändern, daß sich die Haie auch gegenseitig auffressen können. Aber dann sah ich ein, daß die Selbstzerfleischung ihre Existenz nicht wesentlich verlängern würde, den anderen Schwarm aber in seiner Entwicklungsphase gefährden könnte. Schließlich stießen zwei umherstreifende Haie auf diesen Schwarm, und der ganze Zyklus begann von neuem.

Grundzüge des WATOR-Programms

Das Programm für Wa-Tor ist weder sehr lang noch kompliziert. Leser mit einem Personal Computer zu Hause sollten in der Lage sein, ein solches Programm zu schreiben, selbst wenn sie nur wenig Programmiererfahrung haben. Es lohnt sich in jedem Fall. Zu Beginn jedes Laufs muß man Parameter wie die Brutdauer, die Zeitspanne bis zum Verhungern und die anfänglichen Populationsgrößen festlegen. Danach kann man sich entspannt zurücklehnen und zuschauen, wie aus einem Durcheinander von Fischen und Haien langsam ökologische Muster entstehen.

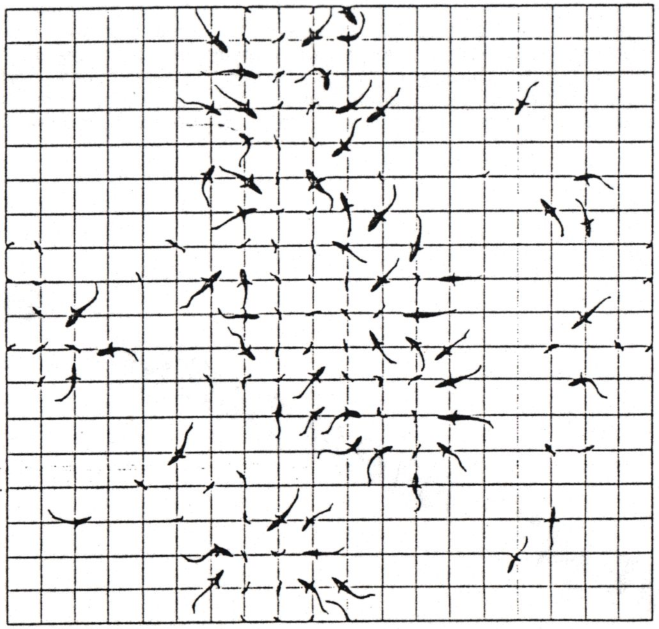


Bild 1: Eine realistische Wiedergabe von Haien auf der Jagd nach Fischen.

Das WATOR-Programm enthält einige einfache Regeln, die das Verhalten der Fische und Haie bestimmen. Der Ozean, in dem sie sich tummeln, besteht aus einem rechteckigen Gitter, dessen gegenüberliegende Seiten „verheftet" werden. Das heißt einfach, daß ein Fisch oder Hai auf einem Gitterpunkt am rechten Rand, der nach Osten (also rechts) schwimmt, auf dem entsprechenden Gitterpunkt am linken Rand wieder auftaucht. Analoges gilt für das Hinausschwimmen über den oberen oder unteren Rand. Das Verheften einer Fläche ergibt im dreidimensionalen Raum einen Torus (Bild 3), und als solcher war Wa-Tor ja auch definiert.

Es bleibt jedem Schöpfer eines WATOR-Programms unbenommen, die Feinheit des Gitters und damit die Zahl der Gitterpunkte selbst festzulegen. Magi zum Beispiel, dessen Programm auf einer VAX läuft, benutzt einen Ozean aus 80 × 23 Gitterpunkten. Meine WATOR-Version ist für einen IBM PC geschrieben und begnügt sich mit einem 32 × 14-Gitter.

Die Zeit verstreicht in diskreten Schritten, von mir Chrononen genannt. Während eines Chronons kann jeder Fisch oder Hai auf den nächsten Gitterpunkt im Norden, Osten, Süden oder Westen überwechseln, sofern dieser nicht schon von einem Mitglied der eigenen Art besetzt ist. Wohin sich das Tier tatsächlich bewegt, entscheidet ein Zufallszahlengenerator.

Für Fische ist die Anweisung einfach: Wähle ein unbesetztes Nachbarfeld zufällig aus und ziehe dorthin. Sind alle vier Nachbarfelder bereits besetzt, bewegt sich der Fisch gar nicht. Da für die Haie die Jagd wichtiger ist als bloßes Herumstreunen, gestalten sich deren Zugregeln etwas komplizierter: Wähle aus denjenigen Nachbarpunkten, auf denen sich Fische befinden, einen zufällig aus, ziehe dorthin und verschlinge den Fisch. Falls sich in der Nachbarschaft kein Fisch aufhält, ziehe entsprechend der Regel für Fische auf einen unbesetzten Nachbarpunkt.

Vor jedem Simulationslauf von WATOR sind fünf Parameter zu wählen. Die Parameter *nfische* und *nhaie* legen die Anzahl der Fische und Haie zu Beginn fest. Das Programm verteilt die Tiere in der verlangten Menge zunächst zufällig und mehr oder weniger gleichmäßig auf der Oberfläche des Planeten. Die Parameter *fbrut* und *hbrut* geben die Zahl der Chrononen an, die ein Fisch oder Hai existieren muß, bis er genau einen Nachkommen hat. (Offenbar vermehren sich beide Arten durch Jungfernzeugung.) Schließlich bedeutet *fasten* die Zahl der Chrononen, die ein Hai ohne Nahrung auskommt. Wenn er länger herumschwimmt, ohne etwas Freßbares aufzutreiben, verhungert er und sinkt auf den Grund des Ozeans, wo er nicht mehr zu sehen ist.

Pro Chronon bewegt WATOR jeden Fisch und jeden Hai genau einmal und zeigt das Ergebnis anschließend auf dem Bildschirm. Diese simplen Regeln genügen, um zu simulieren, wie das Ökosystem von Wa-Tor von einer Krise zur nächsten taumelt.

Eine rätselhafte Instabilität

Magi und ich waren Zeugen mehrerer Fünf-Parameter-Szenarien, bei denen Wa-Tors Ozean von Fischen überschwemmt wurde, wodurch sich die Haie jedoch so stark vermehrten, daß sie alle Fische fraßen und dann verhungerten. In anderen Fällen sahen wir Haie Fische eines großen Schwarms verschlingen und anschließend zugrunde gehen, weil sie eine kleine Gruppe von Fischen in ihrer Nähe nicht fanden.

Bei einigen wenigen Gelegenheiten immerhin konnten sich Räuber und Beute über zwei oder sogar drei Zyklen halten, ehe die Hai-Population schließlich doch noch unterging. Leider gaben uns die für die Szenarien gewählten Parameter keinen Hinweis, unter welchen Bedingungen der Bestand des Ökosystems auf Dauer gesichert wäre. Wie aber hatten die Bewohner von Wa-Tor überlebt?

Es heißt, Biologie sei Schicksal. Magi und ich möchten hinzufügen: Ökologie ist Geometrie, wenigstens was Wa-Tor betrifft. Der Ausgang eines Szenarios scheint nicht von der ursprünglichen Verteilung und Größe der beiden Populationen abzuhängen. Ebensowenig ist er eine Folge der zufälligen Bewegungen der Fische und Haie. Entscheidend für die Stabilität des Ökosystems scheinen vielmehr die geometrischen Beziehungen zwischen den beiden Populationen zu sein, wie sie auf unserem Bildschirm zum Ausdruck kommen: Je höher organisiert und lokalisiert eine Population wird, desto wahrscheinlicher ist sie zum Untergang verurteilt.

Aus dieser Erkenntnis heraus fragten wir uns, durch welche Wahl der fünf Parameter sich die Geometrie auflockern ließe. Schließlich ging uns ein Licht auf: Wenn sich die Haie am Rand eines Fischschwarms zusammengeschart hatten, könnte man ihre Phalanx dadurch aufbrechen, daß man sie sich weniger schnell vermehren läßt. Die Zusammenballung beruhte schließlich weniger darauf, daß viele Haie herbeischwammen, als daß die Anwesenden sich ungehemmt vermehrten.

Vor diesem Einfall hatten wir mehr oder weniger gleiche Brutraten für Fi-

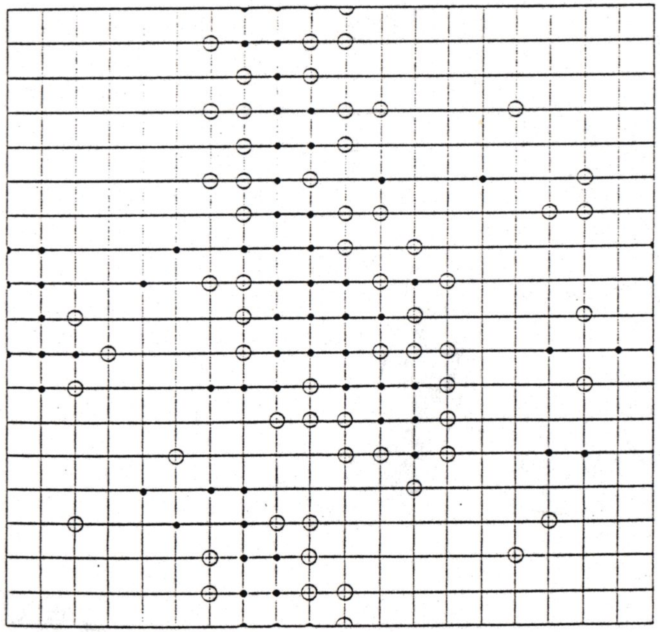


Bild 2: Schematische Darstellung von Bild 1 mit Kreisen als Haien und Punkten als Fischen.

sche und Haie angenommen. Dadurch dachten wir ausgeglichene Populationen zu begünstigen. Derart verschwommenes Denken ist vermutlich für viele Auswüchse unserer hochtechnisierten Welt verantwortlich. Jedenfalls setzte ich nunmehr 200 Fische und 20 Haie in meinen 32 × 14-Ozean und ließ die Fische nach drei, die Haie aber erst nach zehn Chrononen Junge haben. Für die Zeit, nach der Haie ohne Nahrung verhungern, wählte ich relativ willkürlich drei Chrononen.

Nachdem wir meinem ziemlich langsamen Programm etwa eine Viertelstunde lang zugesehen hatten, registrierten wir mit Befriedigung einen ersten Erfolg: Nach anfänglichem Abfall erholten sich die Populationen vollständig. Außerdem existierte zwar eine geometrische Struktur, doch war sie nicht streng, sondern nur andeutungsweise vorhanden (Bild 1). Schwärme bestanden jetzt aus formlosen Haufen mit ausgefransten Rändern. An manchen Stellen zogen Fische und Haie auch einsam ihre Kreise.

Ich ließ das Programm den ganzen Nachmittag laufen und warf nur ab und an zwischen wichtigeren Arbeiten einen Blick auf den Schirm. Das Programm lief auch noch die ganze Nacht, und als ich nach meiner Vorlesung am Morgen wieder in mein Büro trat, fand ich Fische und Haie immer noch quicklebendig beim Durchlaufen ihrer Populationszyklen. Das war Wa-Tor!

Das WATOR-Programm im Detail

Ein WATOR-Programm läßt sich auf viele Arten implementieren, am einfachsten aber geht es mit einer Anzahl zweidimensionaler Felder. Ich benutze fünf Felder, die ich FISCHE, HAIE, FISCHZUG, HAIZUG und HUNGER nenne. In diesen Feldern, alle von der Größe 32 × 14, sind die momentanen Positionen und das Alter der Fische und Haie verzeichnet. So gibt FISCHE(I,J) an, ob sich in Position (I,J) ein Fisch befindet oder nicht. Ist kein Fisch dort, enthält das Feld den Wert −1. Anderenfalls steht darin das Alter des betreffenden Fisches in Chrononen. Das Gleiche gilt für die Matrix HAIE.

Im Feld FISCHZUG ist für jeden Gitterpunkt vermerkt, ob während des aktuellen Zyklus ein Fisch dorthin versetzt wurde. So verhindert man, daß ein Fisch in ein und demselben Chronon zweimal gezogen wird. HAIZUG erfüllt dieselbe Aufgabe bei den Haien. Das Feld HUNGER enthält den Zeitpunkt, zu dem der an dem entsprechenden Gitterpunkt sitzende Hai zuletzt gefressen hat. Befindet sich kein Hai an dieser Stelle, wird −1 eingetragen.

Die einfachste Möglichkeit, das Geschehen auf Wa-Tor auf dem Bildschirm darzustellen, besteht darin, für jede Reihe der Felder FISCHE und HAIE eine Zeile mit Zeichen auszugeben. Dabei steht ein Leerzeichen für ein freies Feld, ein Punkt (.) für einen Fisch und eine Null (0) für einen Haifisch (Bild 2). Mag diese Darstellung auch primitiv scheinen – sie ist überraschend informativ und hübsch anzusehen.

In der Anfangsphase verteilt WATOR die angegebene Zahl von Fischen und Haien zufällig, aber möglichst gleichmäßig über das toroidale Meer. Danach durchläuft das Programm periodisch die unten beschriebenen drei Unterprogramme. Jeder Programmzyklus dauert ein Chronon.

FISCHE SCHWIMMEN UND VERMEHREN SICH:

Für jeden Fisch im Feld FISCHE legt das Programm eine Liste der freien Nachbarfelder an und verschiebt den Fisch wahllos auf eines dieser Felder. Das bedeutet, daß im Feld FISCHE an der alten Position eine −1 und an der neuen Position das aktuelle Alter des Fisches einzutragen ist. Das Feld FISCHZUG wird, wie oben beschrieben, auf den neuesten Stand gebracht. Ist das Alter des Fisches gleich *fbrut*, so wird an die alte Position ein neuer Fisch gesetzt, und beide Fische erhalten das Alter 0. Auch der neue Fisch muß im Feld FISCHZUG eingetragen werden. Sind alle benachbarten Felder bereits besetzt, kann der Fisch sich weder bewegen noch Nachkommen haben.

HAIE JAGEN UND VERMEHREN SICH:

Für jeden Hai im Feld HAIE legt das Programm eine Liste der benachbarten Felder an, auf denen sich Fische befinden (falls es solche Felder überhaupt gibt). Der Hai wählt dann willkürlich eines dieser Felder aus, zieht dorthin und frißt den Fisch. Das heißt, das Programm muß nicht nur die Felder HAIE und HAIZUG, wie bei den Fischen beschrieben, modifizieren, sondern auch den entsprechenden Eintrag im Feld FISCHE auf −1 setzen. Ebenso ist in HUNGER an dieser Stelle eine 0 einzutragen. Befindet sich auf keinem Nachbarfeld ein Fisch, so zieht der Hai wie ein Fisch. Sofern der Hai das Alter *fasten* erreicht hat, stirbt er, und an seiner Position wird im Feld HAIE eine −1 eingetragen. Ist das Alter des Hais dagegen gleich *hbrut*, so setzt das Programm wie bei den Fischen einen neuen Hai ins Leben.

ANZEIGE:

Das Programm durchsucht zeilenweise die Felder FISCHE und HAIE und gibt für jeden Fisch einen Punkt und für jeden Hai eine Null aus. Das kann auf einmal geschehen oder getrennt, nachdem erst die Fische und dann die Haie gezogen haben.

Um den Ozean zu Anfang der Simulation mit Fischen zu bevölkern, konstruiert der Programmierer eine Schleife, die *nfische*-mal zwei Zufallszahlen erzeugt. Diese Zufallszahlen werden mit einem Faktor multipliziert, der so gewählt ist, daß die skalierten Zahlen den Ozean in seiner vollen Größe gerade abdecken. Dann erzeugt der Rechner an jeder so ermittelten Position einen Fisch im Feld FISCHE und gibt ihm willkürlich ein Alter zwischen 0 und *fbrut*. Entsprechend werden die Haie verteilt. In beiden Fällen ist vorher zu prüfen, ob der Gitterpunkt frei ist. Die Fische und Haie erhalten ein zufällig gewähltes Alter, damit später nicht alle zur gleichen Zeit Junge bekommen – was sehr verwirrend und unnatürlich wirken würde.

Simulation eines Betrunkenen

Für ungeübte Programmierer mag diese Beschreibung immer noch zu allgemein sein. Sie sollten vielleicht mit einem Programm beginnen, das den Zickzackkurs eines Betrunkenen simuliert. Ein solches Programm könnte aus einer einzigen großen Schleife bestehen, die sieben Anweisungen umfaßt. Diese werden am besten zunächst in einer algorithmischen Pseudosprache und nicht in einer wirklichen Programmiersprache formuliert. Für Zuweisungen zeichnet man einen Pfeil nach links, und für die Koordinaten des Betrunkenen setzt man die Variablen X und Y. Diese werden in Abhängigkeit vom Wert einer ganzzahligen Zufallsvariablen *Richtung* verändert. Je nachdem, ob dieser Wert 0, 1, 2 oder 3 ist, torkelt der Betrunkene (ein Punkt auf dem Bildschirm) nach Norden, Osten, Süden oder Westen.

Richtung ← ganzzahliger Teil von (*Zufallszahl* × 4)
Wenn *Richtung* = 0, dann $X \leftarrow X + 1$
Wenn *Richtung* = 1, dann $X \leftarrow X - 1$
Wenn *Richtung* = 2, dann $Y \leftarrow Y + 1$
Wenn *Richtung* = 3, dann $Y \leftarrow Y - 1$
Zeige (X, Y) an

Erzeugt Ihr Zufallszahlengenerator eine dezimale *Zufallszahl* zwischen 0 und 1, dann wird sie durch obige Skalierung in den Bereich zwischen 0 und 3,999 gebracht. Der ganzzahlige Teil dieser Zahl muß dann 0, 1, 2 oder 3 sein.

Ich will nicht behaupten, daß die Beobachtung eines über den Bildschirm irrenden Lichtpunktes an Faszinationskraft dem ökologischen Drama zwischen Haien und Fischen gleichkommt, aber das Schreiben dieses Programmchens vermittelt doch eine gewisse Vorstellung davon, wie sich Teile von WATOR erstellen lassen.

Erfahrenen Programmierern werden beim Lesen dieses Artikels natürlich noch andere Möglichkeiten eingefallen sein, ein WATOR-Programm zu schreiben. So läßt sich der Rechenaufwand wesentlich verringern, wenn man Fische und Haie in verketteten Listen führt. Er ist dann für einen Zyklus nur proportional zur jeweils vorhandenen Zahl von Fischen und Haien statt zur Größe des Ozeans.

Der Zusammenhang mit irdischen Ökosystemen

WATOR kann uns natürlich auch etwas über Tierpopulationen auf der Erde lehren. In einfachen Räuber-Beute-Ökosystemen folgen die Populationsgrößen von Räubern und Beutetieren ähnlichen miteinander verwobenen Zyklen wie auf Wa-Tor. Solche zyklischen Schwankungen zeigten beispielsweise die Populationsgrößen der Schneehasen und des Kanadaluchses in der kanadischen Subarktis, über die Daten der Hudson's Bay Company zwischen 1847 und 1903 indirekt Aufschluß geben. In Bild 4 ist aufgetragen, wieviele Hasen und Luchse jeweils innerhalb eines Jahres gefangen wurden. Vermutlich sind die Zahlen der gefangenen Tiere proportional zur Größe des gesamten Bestandes während des betreffenden Zeitraums.

Wenn das zutrifft, lassen sich die Zyklen leicht dadurch erklären, daß sich die Luchse bei wachsender Hasenpopulation wegen des reichlichen Futterangebots immer stärker vermehren. Durch ihre wachsende Zahl dezimieren sie jedoch die Hasenpopulation. So wird für die vielen Luchse bald das Futter knapp, und sie beginnen zu verhungern oder weniger Junge zu bekommen – vielleicht auch beides. Sobald die Luchspopulation genug geschrumpft ist, kann sich die Hasenpopulation wieder erholen.

Zum Vergleich zeigt Bild 5 zwei theoretische Kurven, die sich als Lösung der Lotka-Volterra-Gleichungen ergeben. Diese Differentialgleichungen wurden 1931 von Vito Volterra, einem italienischen Mathematiker, erstmals formuliert. Sie simulieren eine Art kontinuierliche Population von Räubern, die kontinuierlich eine kontinuierliche Beutepopulation zu dezimieren sucht. Die Lösungen dieser Gleichungen zeigen eine zyklische Schwankung, die auf den ersten Blick gut mit den empirischen Luchs-Hasen-Daten übereinstimmt. Die Biologen sind allerdings uneins, ob sich die Populationsgrößen von Luchsen und Hasen durch derart simple Überlegungen erklären lassen. Auf jeden Fall mischen wenigstens zwei weitere Feinde der Hasen mit: Mikroben und Menschen.

Das heißt freilich nicht, daß auch die Statistiken über Fische und Haie auf Wa-Tor wertlos sind. Magi und ich haben solche Statistiken aufgestellt, und unsere letzten graphischen Darstellungen kommen der Luchs-Hasen-Population näher als die Lotka-Voterra-Kurven. Immer

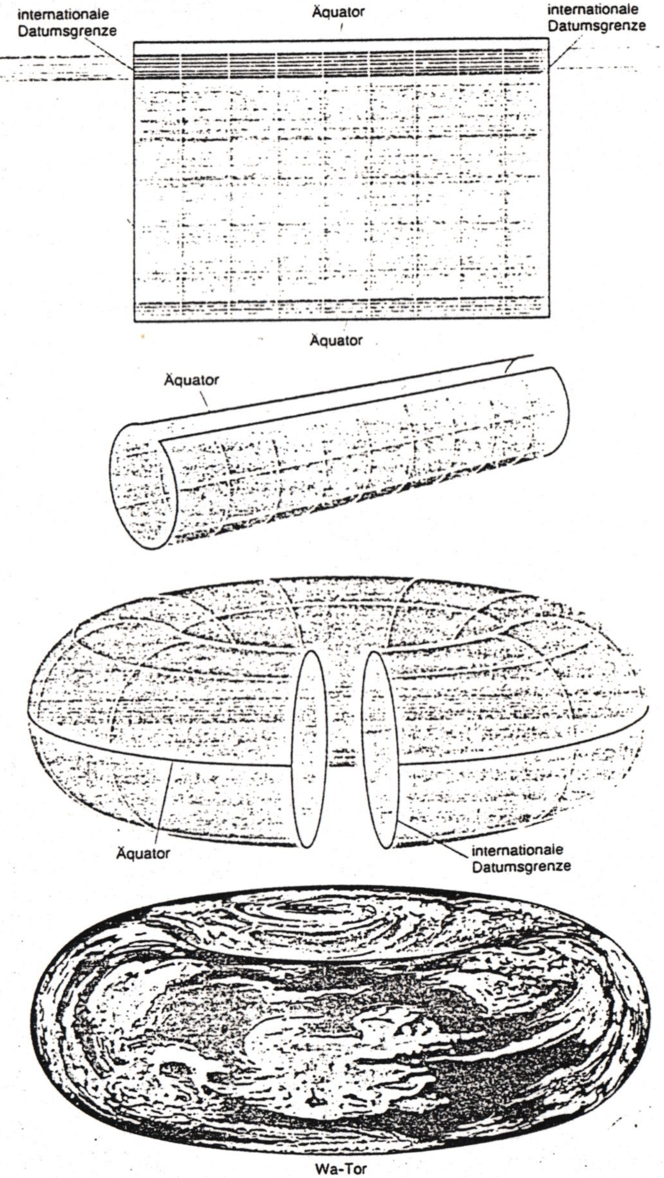
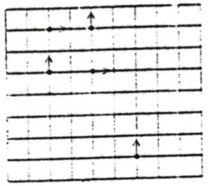


Bild 3: Der Planet Wa-Tor und seine Darstellung auf einer ebenen Karte (oder einem Bildschirm).

noch aber rätseln wir über die merkwürdige langfristige Instabilität des Systems bei gewissen Parameterkombinationen. Vielleicht finden einige Leser, die mit Ihrem eigenen WATOR-Programm herumspielen, eine vernünftige Erklärung. Gibt es eine allgemeine Regel, mit der sich aus der Parameterkombination die Stabilität oder Instabilität des Ökosystems ablesen läßt? Inwieweit folgen die zyklischen Fluktuationen den Lotka-Volterra-Gleichungen?

Daß die Oberfläche von Wa-Tor torusförmig ist, hat einen praktischen Grund: Es macht viel weniger Mühe, ein Programm für einen Ozean ohne Rand oder Küsten zu schreiben. Wenn die Breite des Ozeans zum Beispiel 32 Einheiten beträgt, rechnet man die *X*-Koordinaten der Fische und Haie einfach modulo 32. Wenn also ein Fisch zu einer bestimmten Zeit am rechten Rand des Bildschirms mit der *X*-Koordinate 31 zu sehen ist und sich nach Osten bewegt, hat er ein Chronon später die *X*-Koordinate 32 = 0 und taucht somit am linken Bildschirmrand auf. Vertikal funktioniert das genauso.

Die toroidale Gestalt von Wa-Tor bedingt mitunter sehr merkwürdige Effekte, wie die beiden folgenden Rätsel zeigen. Das erste hängt mit einem Fehler in einer meiner ersten WATOR-Versionen zusammen. Er bewirkte, daß die Fische immer nur nach Norden und die Haie nach Osten schwammen. Wieviele Fische aus dem unten gezeigten Ozean wurden unter dieser Bedingung nie von den Haien gefressen?



Bei dem zweiten Rätsel billigen wir den Fischen und Haien Intelligenz zu. Angenommen, jeder Fisch und jeder Hai zieht abwechselnd zu einem seiner vier Nachbarpunkte. Dabei zeigt sich, daß auf Wa-Tor ein einzelner Fisch bei genügend Intelligenz einem einzelnen Hai immer entkommen kann, wie geschickt sich dieser auch anstellen mag. Bei zwei Haien, die Jagd auf einen Fisch machen, sieht es für diesen nicht so gut aus. Dennoch kann der Fisch auch in diesem Fall entwischen, selbst wenn es die Haie noch so schlau anstellen und sogar gemeinsame Sache machen. Finden Sie heraus wie? Das Ergebnis hängt nicht von der Größe des Ozeans ab.

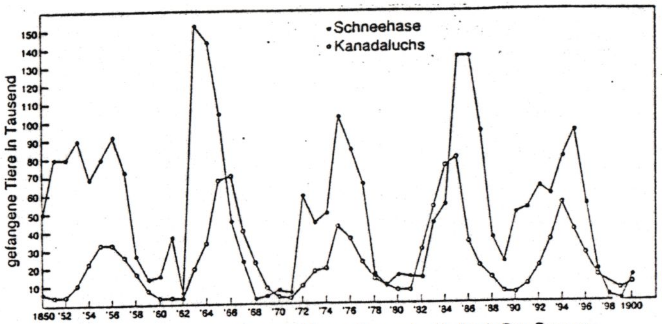


Bild 4: Mengen von Luchsen und Hasen, die von der Hudson's Bay Company zwischen 1850 und 1900 gefangen wurden.

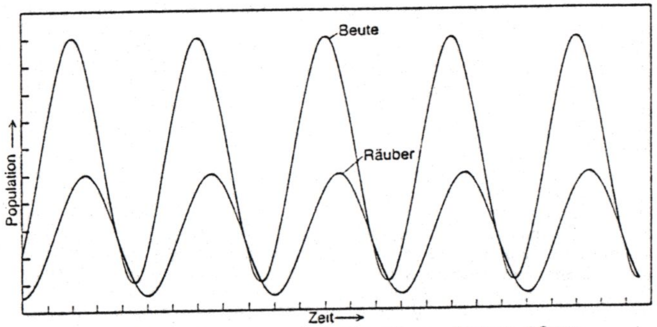


Bild 5: Zyklische Schwankungen in der Population von Räuber und Beute, die sich aus der Lotka-Volterra-Gleichung ergeben.

Ein zweiter Besuch auf Wa-Tor

Mehrere Leser haben moderne Theorien beschrieben, die bei der Analyse von Wa-Tor hilfreich sind. Das letzte Wort darüber, ob stochastische Matrizen die Berechnung spezifischer Überlebenswahrscheinlichkeiten für beliebige Parameterkombinationen ermöglichen, ist noch nicht gesprochen. Es ist jedoch interessant, daß die Lotka-Volterra-Gleichungen seit ihrer Formulierung durch Vito Volterra im Jahre 1931 dahingehend erweitert wurden, daß auch die Diffusion als ein Faktor, der das Gleichgewicht zwischen Räuber und Beute beeinflußt, berücksichtigt wird. Diffusionsterme geben den einfachen Lösungskurven der Lotka-Volterra-Gleichungen eine kompliziertere Gestalt. Der Doppelname dieser Gleichung rührt daher, daß Alfred J. Lotka, ein amerikanischer Mathematiker, etwa ein Jahrzehnt vor Volterra bereits ähnliche Gleichungen angegeben hatte.

Um die Verhältnisse auf Wa-Tor denen auf der Erde anzugleichen, verfeinerten einige Leser ihre Programme und sahen etwa eine Lebenskraft bei den Haien, Mutationen, zwei verschiedene Arten von Fischen und schließlich Plankton vor. Während ich nur geschrieben hatte, daß die Fische von einer unbekannten, im Überfluß vorhandenen Nahrung lebten, sah Jim Lemon aus El Segundo in Kalifornien Plankton explizit als Nahrungsquelle vor. Am Anfang füllt es jeden Punkt, der nicht gerade von einem Hai oder Fisch besetzt ist. Im Verlauf der Simulation vermehrt es sich immer wieder in unbesetzte Felder hinein; es steht zu den Fischen in der gleichen Beziehung wie diese zu den Haien.

Die Haie von Edgar F. Couda aus Park Ridge in Illinois gewinnen oder verlieren an Lebenskraft – je nachdem, wieviel sie zu fressen bekommen. Daher können sie viel länger ohne Nahrung auskommen als die einfachen Haie vom Standardplaneten Wa-Tor. Couda schickte Computerdiagramme, die den auf den Daten der Hudson Bay Company beruhenden Populationskurven bemerkenswert ähneln.

Stephen R. Berggren aus Satellite Beach in Florida schrieb ein System namens EVOLUTION bereits vor zwei Jahren. Es ähnelt WATOR, nur daß sich die Tiere hier Umwelteinflüssen anpassen und sich so fortentwickeln können.


```
(*-----------------------------------------------------------------------*)
(*                             Wator.pas                                  *)
(*-------------------------=============---------------------------------*)
(* Simulation des oekologischen Systems des wasserbedeckten, torus-      *)
(* foermigen (Ring-) Planeten Wator mit seinen 1080 einzelnen Lebens-    *)
(* raeumen (Zellen), in denen entweder kein Lebewesen, ein Fisch oder    *)
(* ein Hai lebt. Bei ungeeigneter Wahl der oekologischen Parameter       *)
(* sterben die Haie oder sogar die Haie und die Fische aus.              *)
(*-----------------------------------------------------------------------*)
(* Nach einer Idee von A.K. Dewdney                                      *)

(* Zellulaerer Automat: Bitbedeutung einer Zelle:                        *)

(* 11xxxxxx xxxxxxxx : Ein Hai   lebt in der Zelle                       *)
(* 10xxxxxx xxxxxxxx : Ein Fisch lebt in der Zelle                       *)
(* 00xxxxxx xxxxxxxx : Die Zelle ist leer                                *)
(* xx111111 xxxxxxxx : max.  63 Chrononen, bis zum Hungertot             *)
(* xxxxxxxx 11111111 : max. 255 Chrononen, bis Nachwuchs folgt           *)

const  hai    = $c000;
       fisch  = $8000;
       leer   = $0000;

(* hier können beliebige Zeichen, je nach Terminal, eingesetzt *)
(* werden                                                       *)

       haibild   = '*';
       fischbild = '+';

       anzf   : Integer = 200; (* Anzahl der Fische   *)
       brutf  : Integer =   2; (* Brutzeit der Fische *)
       anzh   : Integer =  20; (* Anzahl der Haie     *)
       bruth  : Integer =   3; (* Brutzeit der Haie   *)
       essh   : Integer =   2; (* Esszeit der Haie    *)

Type   TypeFeld    = Array[-60..1139] of Integer;

var    alt, neu    : TypeFeld;

       c           : Integer;   (* Chronoen (=Anzahl der Zeiteiheiten)  *)

       k           : Char;
       key         : Boolean;

(*-----------------------------------------------------------------------*)

procedure Rahmen;

var i:Integer;

begin
  clrscr;
  gotoxy(8, 3); for i:=1 to 64 do write('-');
  gotoxy(8,24); for i:=1 to 64 do write('-');
  for i:=4 to 23 do begin
      gotoxy( 7,i); write('!');
      gotoxy(72,i); write('!');
```

```
  end;
end;

procedure ausgabe(var alt:TypeFeld);

const fa:integer = 0;       (* Alter Fischbestand *)
      ha:integer = 0;       (* Alter Haibestand   *)

var  x,y  : integer;         (* Bildschirmkoordinaten *)
     i    : integer;         (* Array Index *)
     f,h  : integer;         (* Neuer Fisch- Haibestand *)

begin
  f:=0; h:=0;
  i:=0;
  key:=false;
  for y:=5 to 22 do begin
    gotoxy(10,y);
    for x:=1 to 60 do
      begin
        if keypressed then begin
            key:=true;
            read(kbd,k);
            lowvideo;
            exit
        end;
        case alt[i] and hai of
          fisch : begin
                    highvideo;
                    write(fischbild);
                    f:=succ(f)
                  end;
          hai   : begin
                    lowvideo;
                    write(haibild);
                    h:=succ(h)
                  end;
          else write(' ');
        end;
      i:=succ(i);
      end;
    end;
  highvideo;
  gotoxy(14,2); write('Chrononen:',C);
  gotoxy(32,2); write('Fische:',f,'(',fa,')  ');
  gotoxy(55,2); write('Haie:',h,'(',ha,')  ');
  fa:=f; ha:=h
end;

(*-----------------------------------------------------------------

procedure Neustart;

var i : integer; (* Array Index *)
    z : integer; (* Zaehler     *)

begin
```



```
  clrscr;
  writeln('Wator - Simulation eines einfachen oekologischen Systems');
  writeln('========================================================');

  writeln(#10,'Biologische Parameter eingeben:');
  fillchar(alt[-60],2400,0);
  write(#10,anzf:3, ' Anzahl   Fische ? '); readln(anzf);
  write(#10,brutf:3,' Brutzeit Fische ? '); readln(brutf);
  writeln;
  write(#10,anzh:3, ' Anzahl   Haie   ? '); readln(anzh);
  write(#10,bruth:3,' Brutzeit Haie   ? '); readln(bruth);
  write(#10,essh:3, ' Esszeit  Haie   ? '); readln(essh);
  C:=0;
  fillchar(alt[-60],2400,0);
  clrscr;
  Rahmen;
  if anzh>1080 then anzh:=1080;
  for z:=1 to anzh do
    begin
      lowvideo;
      repeat i:=random(1080) until alt[i]=0;
      gotoxy(10+i mod 60,5+i div 60);
      write(haibild);
      alt[i]:=hai or swap(succ(random(essh))) or random(bruth);
    end;
  if anzf>1080-anzh then anzf:=1080-anzh;
  for z:=1 to anzf do
    begin
      highvideo;
      repeat i:=random(1080) until alt[i]=0;
      gotoxy(10+i mod 60,5+i div 60);
      write(fischbild);
      alt[i]:=fisch or random(brutf);
    end;
  k:=#0;
end;

(*-----------------------------------------------------------------*)

procedure   sim  (alt:TypeFeld; var neu:TypeFeld);

var      i: integer;  (* Array Laufindex             *)
         j: integer;  (* Aktuelle Zellenbesetzung    *)
         m: integer;  (* Map, gibt freie Plaetze an*)
         a: integer;  (* Anzahl der freien Plaetze *)

label    nexthai,nextfisch;

  procedure TestAltFisch(var m,a: integer);
  begin
    m:=0; a:=0;
    if (alt[succ(i)] and hai)=fisch then begin m:=1; a:=1 end;
    if (alt[  i-60  ] and hai)=fisch then begin m:=m or 2; a:=succ(a)
end;
    if (alt[pred(i)] and hai)=fisch then begin m:=m or 4; a:=succ(a)
end;
    if (alt[  i+60  ] and hai)=fisch then begin m:=m or 8; a:=succ(a) end
```

```
begin (* sim *)
  move(alt[1020],alt[-60],120);
  fillchar(neu,2400,0);
                (* ---------- Hai - Zug ------------- *)
  for i:=0 to 1079 do begin
    if (alt[i] and hai)=hai then
      begin
        if keypressed then begin key:=true; read(kbd,k); exit end;
        j:=alt[i];
        TestAltFisch(m,a);          (* m=0000..1111; a=0..4 *)
        if (hi(j) and 63)=0 then
          if a=0 then goto nexthai;                (* verhungert *)
        if (j and 255)=0 then
          begin                                    (* Nachwuchs Hai *)
            if neu[i]=0
              then neu[i]:=hai or swap(essh) or bruth
              else
                begin
                  TestNeuLeer(m,a);
                  if a>0 then
                    begin
                      nummer(m,a);
                      settier(hai or swap(essh) or bruth,a);
                    end;
                  TestAltFisch(m,a);
                end;
            j:=(j and $ff00) or bruth (* Brutrate neu *)
          end;
        if a>0 then
          begin                   (* Fisch ist da *)
            nummer(m,a);          (* a=1..4 neue Pos *)
            settier(hai or swap(essh)or pred(j and 255),a);
            LoeschAltFisch(a);
            goto nexthai
          end;
        TestNeuLeer(m,a);         (* m=0000..1111; a=0..4 *)
        j:=hai or
           swap(pred(hi(j)and 63)) or (* Hunger verstaerken
*)
           pred(j and 255);       (* Nachwuchs-Wartezeit verkuerzen
*)
      if a=0 then
        begin                     (* weder fisch noch leer *)
          if neu[i]=0 then neu[i]:=j; (* bleibt am selben Ort  *)
        end
      else                        (* leerfelder vorhanden *)
        begin
          nummer(m,a);            (* a=1..4 neue Pos *)
          settier(j,a);
        end;
    end;(* if (alt[i].. *)
    nexthai:
      if i=540 then               (* Torus: von Oben unten an *)
        begin
          move(alt,alt[1020],240);
          move(neu,neu[1020],240)
        end;
```



```
end;

procedure TestNeuLeer(var m,a: integer);
begin
  m:=0; a:=0;
  if neu[succ(i)]=0 then begin m:=1; a:=1 end;
  if neu[ i-60   ]=0 then begin m:=m or 2; a:=succ(a) end;
  if neu[pred(i)]=0 then begin m:=m or 4; a:=succ(a) end;
  if neu[ i+60   ]=0 then begin m:=m or 8; a:=succ(a) end
end;

procedure nummer(var m,a:integer);
(* gibt in a den zufaellig gewaehlten Ort zurueck *)
var n:integer; (* Nummer 1,2,3 od. 4 der neuen Zelle *)
    z:integer; (* Zaehler *)
  procedure nextpos;      (* sucht Ort in Map m *)
  begin
    while not odd(m) do
      begin
        m:=m shr 1; n:=succ(n)
      end;
  end; (* nextpos *)
begin (* nummer *)
  a:=random(a);             (* Zufallsauswahl *)
  z:=0;
  n:=1;                     (* 1=rechts,2=oben,3=links,4=unten *)
  nextpos;
  while z<a do
    begin
      m:=m shr 1;
      n:=succ(n);
      nextpos;
      z:=succ(z)
    end;
  a:=n
end; (* nummer *)

procedure settier(tier:integer; var a:integer);
begin
  case a of
   1 : neu[succ(i)]:=tier;
   2 : neu[i-60]:=tier;
   3 : neu[pred(i)]:=tier;
   4 : neu[i+60]:=tier
  end
end; (* settier *)

procedure LoeschAltFisch(var a:integer);
begin
  case a of
   1 : alt[succ(i)]:=leer;
   2 : alt[ i-60   ]:=leer;
   3 : alt[pred(i)]:=leer;
   4 : alt[ i+60   ]:=leer
  end
end; (* LoeschAltFisch *)
```

```
    end; (* for i *)
    move(alt[1020],alt,240);              (* Torus: von Unten oben an *)
    move(neu[1020],neu,240);
    for i:=0 to 1079 do begin
      if (alt[i] and hai)=fisch then
        begin
          if keypressed then begin key:=true; read(kbd,k); exit end;
          j:=alt[i];
          if (j and 255)=0 then
            begin
              j:=fisch or brutf;
              if neu[i]=leer then neu[i]:=j
              else
                begin
                  TestNeuLeer(m,a);
                  if a>0 then
                    begin
                      nummer(m,a);
                      settier(j,a)
                    end
                  else goto nextfisch;
                end;
            end; (* if (j *)
          TestNeuLeer(m,a);               (* m=0000..1111; a=0..4 *)
          j:=fisch or pred(j and 255);
          if a>0 then
            begin
              nummer(m,a);
              settier(j,a)
            end
          else
            if neu[i]=0 then neu[i]:=j;
        end; (* if (alt[i].. *)
      nextfisch:
      if i=480 then
        begin
          move(alt,alt[1020],240);
          move(neu,neu[1020],240)
        end;
    end; (* for i *)
    move(neu[1020],neu,240);
  end;(* sim *)

  procedure simuliere;
  begin
    Rahmen;
    ausgabe(alt);
    key:=false;
    repeat
      sim(alt,neu);
      if key then exit;
      alt:=neu;
      C:=succ(C);
      ausgabe(alt);
    until key;
  end;
```



```
(*-----------------------------------------------------------------------*)

begin (* main *)
  Neustart;
  repeat
    clrscr;
    highvideo;
    writeln('  Wator Simulation  ',#10,#10);
    writeln('N Neustart');
    writeln('< Ende');
    writeln('A Aktuelle Population zeigen');
    writeln('S Simulation');
    read(kbd,k);
    k:=upcase(k);
    repeat
      case k of
       'N' : Neustart;
       'A' : begin Rahmen; ausgabe(alt); read(kbd,k) end;
       'S' : simuliere;
      end;
    until pos(k,'NAS')=0;
  until k='<';
end.
```

Die Birne glüht ...

"Do you have four-volt, two-watt bulbs?"
"For what?"
"No, two."
"To what?"
"Yes!"
"No!"
"..."

SMILIES

Im FIDO-Netz wird teilweise eine sehr unkonventionelle Schreibweise der deutschen Sprache geplegt. Das fängt beim sogenannten Quoten (dem Übernehmen von Textpassagen aus der Message, auf die man gerade antwortet) an und hört bei den Smilies noch lange nicht auf.

Smilies sind Symbole, die an Sätze oder Bemerkungen angefügt werden, und ein gewisses Stimmungsbild des Schreibers ausdrücken sollen. Da im FIDO-Netz nur 7-Bit ASCII-Zeichen erlaubt sind, müssen die Smilie-Symbole aus ganz normalen Zeichen zusammengesetzt werden. Hier eine kleine Auswahl davon:

:-) der Schreiber ist fröhlich gestimmt
;-) die vorhergehende Bemerkung sollte man nicht so ernst nehmen
:-(den Schreiber hat irgendetwas geärgert
:-I irgendwo dazwischen
:-> sarkastisches Grinsen
(-: der Schreiber ist Linkshänder
%-) der Schreiber sitzt schon 15 Stunden vor dem Bildschirm
:*) der Schreiber hat schon ein paar Flaschen Bier o.ä. intus
8-) der Schreiber ist Brillenträger
B-) nochmal ein Brillenträger, diesmal mit Hornbrille
|-O der Schreiber ist müde
:-Q der Schreiber ist Raucher
<|-) der FIDO-Teilnehmer ist Chinese

Die Reihe ließe sich fast unendlich fortsetzen. Ich habe aus einer Mailbox eine Datei gefischt, die fast 500 Beispiele für Smilies und ihre Deutung enthält. Natürlich bleibt es jedem selbst überlassen, weitere Smilies zu "erfinden", der Fantasie sind kaum Grenzen gesetzt. In der MOTORRAD.GER, der deutschen Area für motorradfahrende FIDO-User findet man z.B. häufig folgende Smilies:

(|:-) Motorradfahrer mit Integralhelm
(8-) Motorradfahrer mit Halbschalenhelm und Schutzbrille
=:-) der Knallkopf fährt ohne Helm!

Wundert Euch nicht, wenn Ihr mal in einem meiner Briefe einen Smilie findet. Ich habe mich durch das FIDO-Netz so daran gewöhnt, daß ich mich ab und zu nicht zurückhalten kann ;-)

The Unofficial Smilie Dictionary

In den Fido-Mails tauchen immer wieder diese seltsamen Hieroglyphen auf, die einfach keinen Sinn zu ergeben scheinen. Wenn man den Kopf aber um 90 Grad nach links oder rechts neigt, kommt man dem Geheimnis recht schnell auf die Spur. Hier ist eine kleine Auswahl der verschiedenen Smilies, die es als File in fast jeder Fido-Box gibt.

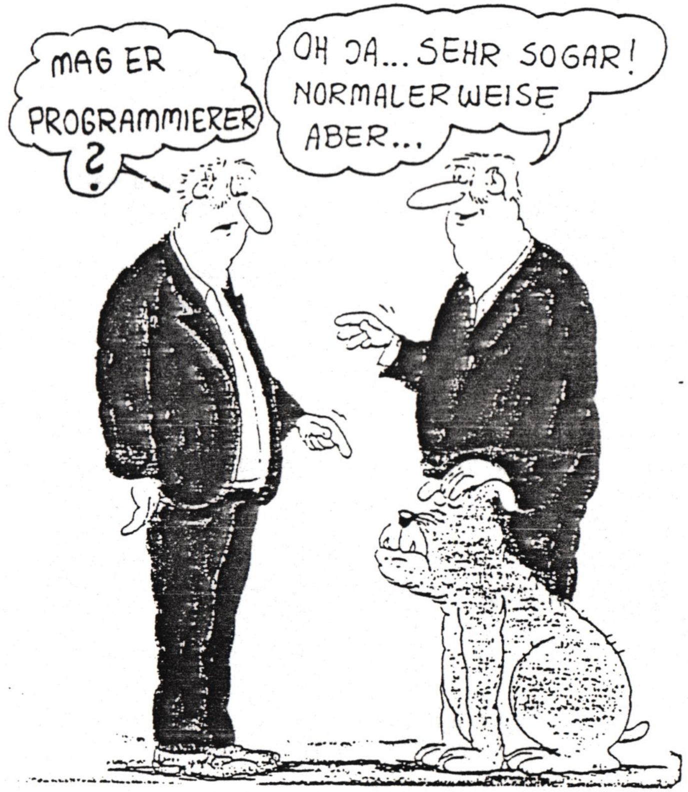
:-) basic smilie
;-) winky smilie
:-(frowning smilie
:-I indifferent smilie
:-> sarcastic smilie
>:-> making a really devilish remark
>;-> winky and devilish combined

Those are the basic ones... Here are some somewhat less common ones:

(-: user is lefthanded
%-) user has been staring at a green screen for 15 hours strait
:*) user is drunk
8-) user is wearing sunglasses
B-) user wears horn-rimmed glasses
8:-) user is a little girl
:-)-8 user is a big girl
:-#) user has a mustache
#:-) user wears a toupee
:-(toupee in an updraft
:-C user is a vampire
:-E bucktoothed vampire
:-F bucktoothed vampire with one tooth missing
:-7 user just made a wry statement
:-* user just ate something sour
:'-(user is crying
:'-) user is so happy that she/he is crying
:-O user is screaming
:-# user wears braces
:n) user has a broken nose
:u) user has a broken nose but it is the other way
:-& user is tongue tied
=:-) user is a hosehead
--:-) user is a punk rocker
--:-(real punk rockers don't smile
:=) user has two noses
+-:-) user it the pope or holds some other religious office
':-) user shaved one of his eyebrows this morning
|-I user is asleep
|-O user is jawning/snoring
:-Q user is a smoker
:-? user smokes a pipe
O-) megatron man on partol ! (or else user is a scuba diver)
O :-) user is an angel
:-P - no comment -
:-S user just made an incoherent statement
:-D user is laughing (at you!)
:-X users lips are sealed
<|-) user is a chinese
<|-(user is a chinese and doesnt like this kind of jokes

```
:-/      user is sceptical
@-*      user is pro-nuclear war
:-o      uh oh !!
(8-o     it's Mr. Bill !!
*:O)     Bonzo the clown
:-9      user is licking his/her lips
%-9      user is braindead
[:-)     user is wearing a walkman
(:I      user is an egghead
<:-I     user is a dunce
{:-)     user is wearing a turban
:-:      mutant smile
.-)      user only has one eye
,-)      dito, but he is winking
X-(      user just died
8 :-)    user is a wizard

Note: a lot of these can be typed without noses to make midget smilies !!

:)       midget smilie
---------------------------------------------------------------------
```

Alexander Schmid



„Die Angst vor einem Mißbrauch der Volkszählungsdaten sei jedoch unbegründet. Anders, als mancherorts vermutet, werden die gewonnenen Daten nämlich nicht in Computern, sondern auf Magnetbändern und Platten gespeichert."

Oberregierungsrat Lorenz, neutraler Referent auf einem Informationsabend in Markt Wolnzach; zitiert nach einem Bericht im Pfaffenhofener Kurier/Wolnzacher Anzeiger vom 28. April 1987.

Impressum

1. Vorsitzender: **Hartmut Obermann**
Mozartring 23
Postfach 14 30
8870 Günzburg
Tel.: 0 82 21/ 3 02 48
FAX: 0 82 21/ 3 35 75

2. Vorsitzender: **Gerald Schröder**
Am Schützenplatz 14
2105 Seevetal 1
Tel.: 0 41 05/ 26 02

Hardwarekoordinator: **Andreas Magnus**
Bismarckstraße 29
4650 Gelsenkirchen
Tel.: 02 09/ 87 02 30

NewDOS-Diskothekar: **Oliver Volz**
Am Ochsenwald 37A
7000 Stuttgart 80 (Rohrerhöhe)
Tel.: 07 11/ 74 40 51

CP/M-Diskothekar: **Fritz Chwolka**
Saarstraße 34
5173 Aldenhoven
Tel.: 0 24 64/ 89 20

C-128-Diskothekar: **Günter W. Braun**
Tel.:
Postfach 80 02 26
8000 München 80

Clubbücherei: **Kurt Müller**
Sophie-Scholl-Ring 3b
2054 Geesthacht
Tel.: 0 41 52/ 7 06 43

Redaktion: **Jens Neueder**
Rudolf-Then-Straße 32
7178 Gschlachtenbretzingen
Tel.: 07 91/ 4 28 77
FAX: 0 79 71/ 2 50 55

Bankverbindung: **Club 80**
Postgiro Sonderkonto CLUB 80
Obermann H., 8870 Günzburg
Postgiroamt Frankfurt
BLZ: 500 100 60
Kto.Nr.: 496 071 - 605

Autoren: Die Redaktion bedankt sich bei den im **Inhaltsverzeichnis genannten Autoren** für die Mitarbeit an der Club-INFO.
Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.

Schluß

Hallo Club 80'er,

sicher habt Ihr schon alles Neue, was es zu berichten gibt, auf den ersten Seiten des INFO's gelesen und seit über die aktuelle Clubsituation gut unterrichtet. Aus diesem Grund erscheint das INFO auch erst jetzt und nicht, wie angekündigt, vor dem Clubtreffen.

Auch zum Thema Club-INFO gab es einige wichtige Entscheidungen. Gut finde ich die festen Redaktionstermine zum jeweiligen 01. im März, Juni, September und Dezember. Damit lässt sich auf alle Fälle schön planen.

Die Pro/Kontra-Geschichte mit dem INFO-Format steht noch aus, da der Günter seine "Kontra-Quer-Version" erst zum nächsten INFO liefern wird. Von mir aus sei in Kürze dazu gesagt, daß ich am Querformat festhalte, weil:
1. ...die Seitenlayouts entsprechend gerichtet sind
2. ...ich damit die hälfte Arbeit habe
   (beim Kopieren und Sortieren -**für mich ganz wichtig!**)
3. ...Kostenersparnis: Papier, Toner und Strom sind halbiert
   (sicher auch ein Umweltaspekt!)
4. ...seit 01. April gibt es die günstige Versendungsform der Drucksache nicht mehr, sodaß sich die Portokosten allein schon verdoppeln. Die Versendung der Info's wird künfig jeweils über 200.-DM ausmachen. Die angesparten Talers in unserer Kasse werden also recht schnell schwinden.

Dies als kleine Entscheidungshilfe für die Beantwortung der Formatfrage.

Nun aber zu dem INFO, daß diesmal durch die Sonderbeilagen, recht umfangreich geworden ist.
DFÜ ist angesagt, dazu haben wir ein ganzes Paket an Informationen zusammengeschnürt. Wir hoffen, daß die Info's auch für "Nicht-Datenreisende" interessant sind.
Als weiteres haben wir eine aktuelle Liste unserer Clubbücherei aufgelegt, bitte macht regen Gebrauch von dieser Clubeinrichtung.

So, daß wars auch schon wieder,
bis zum nächsten mal Euer

Adressen-Liste Club80

Stand vom 06-04-1993
Seite 1

Nachname	Vorname	Straße	PLZ	Ort	Telefon privat	Telefon geschäftlich
Behrendt	Detlef	Schlosserbreite 1a	W- 8018	Grafing	08092 / 9173	
Benner	Achim	Vorm Mühlberg 1	W- 5910	Kreuztal 3	02732 / 3780	(ab 18.00Uhr)
Berndt	Wolfgang	Friedberger Straße 92c	W- 6360	Friedberg 2	06031 / 2963	
Berndt-Jochum	Ilse	Stachelsgut 24	W- 5060	Bergisch-Gladbach 1	02204 / 65254	
Bernhardt	Helmut	Hafenstraße 7	W- 2305	Heikendorf	0431 / 241907	0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	W- 8551	Hausen	09191 / 31698	09191 / 611108
Brans	Jörg	Tieloh 55	W- 2000	Hamburg 60	040 /6906531	
Braun	Harald	Postfach 8011	W- 2300	Kiel 17		
Braun	Günter W.	Postfach 800 226	W- 8000	München 80		
Böckling	Ulrich	Jochaczstraße 61	W- 5410	Hör-Grenzhausen	0264 / 4861	02631 / 895168
Chwolka	Fritz	Saarstraße 34	W- 5173	Aldenhoven	02464 / 8920	
Dose	Volker	Dorfstraße 10	W- 2304	Brodersdorf	04343 / 1357	
Drechsler	Oskar	Duckterather Busch 2	W- 5060	Bergisch Gladbach 2	02202 / 55282	0228 / 128114
Gill	Thomas	Maria-Eich-Str. 34	W- 8000	München 60	089 /8349527	
Günther	Jens	Bannerscheid 7	W- 5231	Neitersen	02681 / 1553	
Halgasch	Gert	Großschonauer Straße 26	O- 8805	Jonsdorf		
Hartmann	Hans-Günther	Möwenstraße 9	W- 2876	Berne 2	04406 / 6911	0421 /2482419
Hebecker	Ulrich	Büsumerstraße 15	W- 7000	Stuttgart	0711 / 734800	
Held	Manfred	Stirner Straße 22	W- 8835	Pleinfeld	09144 / 6563	0911 /2195245
Hermann	Klaus	Forchenstraße 8	W- 7401	Pliezhausen	07127 / 71945	
Hürdler	Manfred	Niederhofer Straße 29	W- 8709	Rimper	09365 / 4235	
Johnen	Willi	Hansemannstraße 1	W- 5160	Düren	02421 / 501305	
Kauka	Dietmar	Straße des Friedens 37	A- 7201	Neukirchen	0037404/850112	
Kemmer	Jürgen	Dorfberg 7	W- 8701	Sulzdorf	09334 / 1050	
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 /4828948	00411 / 258170
Kuhn	Eckehard	Im Dorf 14	W- 7443	Frickenhausen	07022 / 45417	
Littmann	Claus	Plockhorst, Zum Spring 15	W- 3155	Edemissen	05372 / 7796	
Lorenz	Walter	Mahräckerstraße 9	W- 6000	Frankfurt /Main 50	069 / 531656	
Magnus	Andreas	Bismarckstraße 29	W- 4650	Gelsenkirchen	0209 / 870230	
Mahlert	Herbert	Hohenbudbergerstraße 112 A	W- 4100	Duisburg	02135 / 47217	
Mand	Harald	Kl. Flintbeker Straße 7	W- 2302	Flintbek /Kiel	04347 / 3629	0431 /3013580
Menk	Christian	Ollsener Straße 52	W- 2116	Hanstedt	04184 / 7825	
Müller	Kurt	Sophie-Scholl-Ring 3b	W- 2054	Geesthacht	04152 / 70643	
Mössel	Franz	Schafferstraße 12	I- 39012	Meran		
Neebe	Gerhard	Märkische Straße 186	W- 4600	Dortmund 1	0231 / 416549	
Neueder	Jens	Rudolf-Then-Straße 32	W- 7178	Gschlachtenbretzingen	0791 / 42877	0791 /44-4722
Neumann	Christof	Zeitblomstraße 22/2	W- 7900	Ulm	0731 /6022568	0731 /9749720
Nitschke	Stefan	Germanenstraße 5	W- 7519	Walzbachtal 1	07203 / 452	
Obermann	Hartmut	Mozartring 23	W- 8870	Günzburg	08221 / 30248	
Peters	Jürgen	Heukoppel 14	W- 2000	Hamburg 14	040 /6412371	
Reit	Hermann	Vechter Hof 40	W- 4500	Osnabrück	0541 / 16331	
Retzlaff	Bernd	Kleiner Sand 98	W- 2082	Uetersen	04122 / 43551	04103 / 605310
Riechmann	Michael	Letelner Heideweg 12	W- 4950	Minden	0571 / 36627	
Rinio	Gerd	Rennbahnstraße 9	W- 2000	Hamburg 74	040 /6552630	
Ruschinski	Claus	Pommernstraße 21	W- 4370	Marl	02365 / 34646	
Schilling	Werner	Ehndorfer Straße 340	W- 2350	Neumünster	04321 / 61116	
Schimmer	Jörg	Stettinerstraße 28	W- 6000	Frankfurt 60	06109 / 35336	
Schmid	Alexander	Entmannsdorf 5	W- 8640	Kronach/Gehülz	09261 / 53496	
Schmitz	Rainer	Küferweg 12/1	W- 7327	Adelberg	07166 / 1397	
Schober	Frank-Michael	Weberweg 2	O- 7590	Spremberg	0037574/ 4565	
Schoberth	Uwe	Petrus-Waldus-Straße 14	W- 7136	Oetisheim	07041 / 7254	0711 /2011455
Scholz	Hans-Werner	Spitalstraße 54	W- 4054	Nettetal		
Schroers	Horst-Dieter	Breslauer Straße 9	W- 8016	Feldkirchen	089 /9032615	089 /9032615
Schröder	Gerald	Am Schützenplatz 14	W- 2105	Seevetal 1	04105 / 2602	
Schröder	Peter	Theodor-Fahr-Straße 32	W- 2000	Hamburg 62	040 /5311582	
Schröer	Egbert	Joachimstraße 18	W- 4270	Dorsten 1	02362 / 75311	02362 / 494392
Schulte	Hartmut	Entenschnabel 8	W- 3162	Uetze	05173 / 1248	
Schut	Andre	Sanderstraße 26	W- 1000	Berlin 44	030 /6917861	
Schwarz	Wolfgang	Schwedenring 6	W- 8850	Donauwörth	0906 / 3092	
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	W- 5305	Alfter 4	0228 / 643853	
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanloese		
Sonnemann	Harald	In den Eckwiesen 9	W- 6101	Fischbachtal 1	06166 / 8512	
Stumpferl	Stefan	Hasenberglstraße 57	W- 8000	München 45	089 /3144001	
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	W- 6500	Mainz 1	06131 / 320860	06131 / 395268
Tornow	Wilhelm	Elbblick 46	W- 2153	Neu Wulmstorf	040 /7007280	04723 / 3063
Vollkmer	Richart	Am Spörkel 69	W- 4600	Dortmund 50		
Volz	Oliver	Am Ochsenwald 37A	W- 7000	Stuttgart 80 (Rohrerhöh	0711 /744051	0711 /6853013
Werner	Heiko	Reichenberger Straße 5	O- 8032	Dresden		
Wittkamp	Heinz	Hindenburgstraße 37	W- 5630	Remscheid	02191 / 75132	
Wulf	Hans-Otto	Im Brahmkamp 38	W- 4250	Bottrop	02041 / 688972	

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

CP/M Literaturrecherche

Literatur-Recherche

Egbert Schröer, März 1993

Wer sich mal die Frage gestellt hat, was an CP/M Literartur in Buchform in Deutschland erschienen und noch erhältlich ist hier eine Antwort. Nach dem Schlagwort CP/M gesucht erhält man folgendes Ergebnis:

```
=> e cp/m /ct
E1          1       COZUMEL/CT
E2          1       CP-INVARIANZ/CT
E3         37 -->   CP/M/CT
E4          6       CP/M 2.2/CT
E5          3       CP/M 3.0/CT
E6          1       CP/M 68 K/CT
E7          1       CP/M 80/CT
E8          5       CP/M PLUS/CT
E9          1       CP/M: UNTERRICHTSMATERIAL/CT
E10        32       CPC/CT
E11        39       CPC 464/CT
E12        13       CPC 6128/CT

=> s e3-e9
           37 "CP/M"/CT
            6 "CP/M 2.2"/CT
            3 "CP/M 3.0"/CT
            1 "CP/M 68 K"/CT
            1 "CP/M 80"/CT
            5 "CP/M PLUS"/CT
            1 "CP/M: UNTERRICHTSMATERIAL"/CT
L1         52 ("CP/M"/CT OR "CP/M 2.2"/CT OR "CP/M 3.0"/CT OR
              "CP/M  68 K"/CT OR "CP/M 80"/CT OR "CP/M  PLUS"/CT
              OR "CP/M: UNTERRICHTSMATERIAL"/CT)

=> dis l1 1-52

L1   ANSWER 1 OF 52  COPYRIGHT 1993
AN   (90)901247294  BIBLIODATA
TI   Betriebssystem CP, M: Aufbau und Anwendung.
AU   Kramer, Manfred
AS   Manfred Kramer
PB   Berlin : Brandenburg. Verl.-Haus, 1990.
SO   1. Aufl. - 128 S. ; 23 cm. - ISBN 3-327-01024-2 kart. :
     DM 7.00.
TS   (Amateur-Bibliothek)
DN   90,A41,0521
```



```
L1   ANSWER 2 OF 52  COPYRIGHT 1993
AN   (90)900552883  BIBLIODATA
TI   CP M in der Praxis
AU   Bernert, Joachim; Burow, Michael; Hanisch, Christian
AS   Joachim Bernert ; Michael Burow ; Christian Hanisch
PB   Berlin : Verl. Technik, 1990.
SO   2., unver*aend. Aufl. - 224 S. : 126 graph. Darst. ; 23 cm. -
     ISBN 3-341-00567-6 kart. : M 23.00.
TS   (Technische Informatik)
DN   90,A17,0618

L1   ANSWER 3 OF 52  COPYRIGHT 1993
AN   (89)890623325  BIBLIODATA
TI   CP,M, CP,M-plus, BASIC 80
AU   M*ueller, Frank
AS   Frank M*ueller
PB   Rinteln : Merkur-Verl., X1987U.
SO   48 Bl. ; 30 cm. - ISBN 3-8120-2601-5 Kunststoff : DM 34.50.
TS   (Quickmanual)
DN   89,A17,0712

L1   ANSWER 4 OF 52  COPYRIGHT 1993
AN   (89)890471088  BIBLIODATA
TI   CP M in der Praxis
AU   Bernert, Joachim; Burow, Michael; Hanisch, Christian
AS   Joachim Bernert ; Michael Burow ; Christian Hanisch
PB   Berlin : Verl. Technik, 1988.
SO   1. Aufl. - 224 S. : graph. Darst. ; 23 cm. -
     ISBN 3-341-00567-6 kart. : M 23.00.
TS   (Technische Informatik)
DN   89,A13,0603

L1   ANSWER 5 OF 52  COPYRIGHT 1993
AN   (89)890440115  BIBLIODATA
TI   Das CP-M-plus-Anwenderbuch: f*uer Schneider CPC 6128, Joyce,
     Commodore 128 und alle CP/M-plus-Rechner.
AU   Dereser, Karlheinz; Ruprecht, Chris
AS   XAutor dieser Ausg.: Karlheinz Dereser ; Chris RupprechtU
PB   W*uerzburg : Vogel, 1987.
SO   1. Aufl. - 114 S. ; 30 cm. - kart. : DM 28.00.
TS   (Chip special : Besser programmieren)
CR   Rupprecht, Chris -> Ruprecht, Chris
DN   89,A12,0616

L1   ANSWER 6 OF 52  COPYRIGHT 1993
AN   (88)880988851  BIBLIODATA
TI   Softregister CP/M
PB   Berlin : Vahle, 1985.
SO   X140U S. ; 14 cm. - ISBN 3-9800252-6-8 spiralgeh. : DM 13.80.
DN   88,A36,0602
```

L1 ANSWER 7 OF 52 COPYRIGHT 1993
AN (88)880428821 BIBLIODATA
TI Das grosse Buch der Public-Domain-Software: freie CP/M-Programme f*uer Commodore-128, Schneider CPC u. Joyce.
AU Kotulla, Martin
AS Martin Kotulla
PB Hannover : Heise, 1988.
SO 1. Aufl. - 229 S. ; 23 cm. - ISBN 3-88229-159-1 kart. : DM 34.80.
DN 88,A34,0768; 88,N15,0205

L1 ANSWER 8 OF 52 COPYRIGHT 1993
AN (88)871454165 BIBLIODATA
TI ATARI ST: Arbeiten mit CP/M.
AU Bachmann, Bernhard
AS Bernhard Bachmann
PB D*uesseldorf ; San Francisco ; Paris ; London : Sybex, 1987.
SO 1. Aufl. - 255 S. : Ill. ; 22 cm. - ISBN 3-88745-665-3 Pp. : DM 48.00.
DN 88,A01,0490

L1 ANSWER 9 OF 52 COPYRIGHT 1993
AN (87)871436825 BIBLIODATA
TI Arbeiten mit CP, M
AU Steinhoff, Alfons; R*uebsam, Klaus Michael
AS Alfons Steinhoff ; Klaus Michael R*uebsam
PB D*uesseldorf ; San Francisco ; Paris ; London : SYBEX, 1987.
SO 1. Aufl. - 251 S. : graph. Darst. ; 22 cm. - ISBN 3-88745-622-X Pp. : DM 38.00. - Auf d. Umschlag: Klaus-Michael R*uebsam.
CR R*uebsam, Klaus-Michael -> R*uebsam, Klaus Michael
DN 87,A51,0638

L1 ANSWER 10 OF 52 COPYRIGHT 1993
AN (87)870979086 BIBLIODATA
TI Commodore 128, Arbeiten mit CP-M Plus
AU Bachmann, Bernhard
AS Bernhard Bachmann
PB D*uesseldorf ; San Francisco ; Paris ; London : Sybex, 1987.
SO 1. Aufl. - 282 S. ; 22 cm. - ISBN 3-88745-638-6 Kunststoff : DM 48.00.
DN 87,A35,0573

L1 ANSWER 11 OF 52 COPYRIGHT 1993
AN (87)870973665 BIBLIODATA
TI Den Joyce programmieren: raffinierte Programmier-Tricks.
AU Kotulla, Martin
AS Martin Kotulla
PB M*uenchen : Franzis, 1987.
SO 158 S. ; 21 cm. - ISBN 3-7723-8861-2 kart. : DM 38.00.
DN 87,A45,0565; 87,N35,0292

L1 ANSWER 12 OF 52 COPYRIGHT 1993
AN (87)870601849 BIBLIODATA
TI Installieren und Pachten von WordStar: 3.0 unter CP/M u. 3.4X unter MS-DOS.
AU Weinberg, Werner
AS Werner Weinberg
PB W*uerzburg : Vogel, 1987.
SO 1. Aufl. - 334 S. ; 22 cm. - ISBN 3-8023-0175-7 Kunststoff : DM 43.00.
TS (Chip-Wissen)
DN 87,A35,0588; 87,N19,0173

L1 ANSWER 13 OF 52 COPYRIGHT 1993
AN (87)861135210 BIBLIODATA
TI Apple CP, M: Assembler-Programmierung.
AU Kersten, Heinrich
AS Heinrich Kersten
PB Heidelberg : H*uethig, 1987.
SO 253 S. : graph. Darst. ; 30 cm. - ISBN 3-7785-1379-6 kart. : ca. DM 35.00.
DN 87,A17,0802; 86,N48,0122

L1 ANSWER 14 OF 52 COPYRIGHT 1993
AN (87)860944794 BIBLIODATA
TI Programmer's utilities und SID f*uer CP, M 2.2 und CP, M 3.0 (Plus): in engl. Sprache. (Programmer's utilities guide and symbolic instruction debugger).
CS Digital Research <Pacific Grove, Calif.>
AS Digital Research
PB Haar bei M*Uenchen : Markt-und-Technik-Verlag, 1986.
SO XIII, 294, 72 S. ; 24 cm. - ISBN 3-89090-372-X Pp. : DM 49.00.
TS (Technik Originaldokumentation)
DN 87,A06,0584; 86,N40,0263

L1 ANSWER 15 OF 52 COPYRIGHT 1993
AN (87)860783561 BIBLIODATA
TI CP-M: e. Betriebssystem stellt sich vor.
AU Weiss, Roman
AS Roman Weiss
PB Heidelberg : H*uethig, 1986.
SO 178 S. : graph. Darst. ; 21 cm. - ISBN 3-7785-1266-8 kart. : DM 39.00.
DN 87,A03,0607; 86,N34,0255

L1 ANSWER 16 OF 52 COPYRIGHT 1993
AN (86)860944883 BIBLIODATA
TI XCP-M-Plus-CP-M-Version-Drei-BetriebssystemU CP-M-Plus-(CP-M-Version-3)-Betriebssystem: Benutzerhandbuch (dt.) u. Programmierhandbuch (engl.).
CS Digital Research <Pacific Grove, Calif.>
AS Digital Research
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 281, X230U S. ; 24 cm. - ISBN 3-89090-371-1 Pp. : DM 38.00.
TS (Originaldokumentation)

DN 86,A51,0734; 86,N40,0248

L1 ANSWER 17 OF 52 COPYRIGHT 1993
AN (86)860697940 BIBLIODATA
TI C 128 - alles *ueber CP, M 3.0: Beschreibung d. CPM-Befehle ; Struktur von CP/M ; CPM-Dateien ; Programmieren unter CP/M (Turbo-Pascal, Microsoft-Basic).
AU Becker, Wolf-J*uergen
AS Wolf-J*uergen Becker
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 299 S. ; 24 cm. - ISBN 3-89090-370-3 Pp. : DM 52.00.
TS (Commodore-Sachbuch)
DN 86,A45,0565; 86,N31,0239

L1 ANSWER 18 OF 52 COPYRIGHT 1993
AN (86)860689026 BIBLIODATA
TI Einf*uehrung in Turbo Pascal unter CP M 80
AU Harbeck, Gerd
AS Gerd Harbeck
PB Braunschweig ; Wiesbaden : Vieweg, 1986.
SO XIII, 160 S. : 1 graph. Darst. ; 23 cm. - ISBN 3-528-04440-3 kart. : DM 32.00.
TS (Programmieren von Mikrocomputern ; Bd. 22)
DN 86,A45,0571; 86,N30,0228

L1 ANSWER 19 OF 52 COPYRIGHT 1993
AN (86)860600874 BIBLIODATA
TI CP-M st*aendig im Griff
AU Weiss, Roman
AS Roman Weiss
PB Heidelberg : H*uethig, 1986.
SO 91 S. ; 17 cm. - ISBN 3-7785-1267-6 spiralgeh. : DM 21.80.
DN 86,A41,0648; 86,N26,0269

L1 ANSWER 20 OF 52 COPYRIGHT 1993
AN (86)860583732 BIBLIODATA
TI XCP-M-drei-null-Anwenderhandbuch C 128Ʊ CP-M-3.0-Anwenderhandbuch C 128: e. unentbehrl. Nachschlagewerk f*uer d. prakt. Arbeit mit CP/M Plus u. seinen Hilfsprogrammen ; mit zahlr. Beispielen u. ausf*uehrl. systemspezif. Daten zur internen Speicherorganisation u. zu Schnittstellen.
AU H*ueckst*aedt, J*uergen
AS J*uergen H*ueckst*aedt
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 250 S. ; 24 cm. - ISBN 3-89090-196-4 Pp. : DM 52.00, sfr 47.80, S 405.60.
DN 86,A38,0674; 86,N25,0129

L1 ANSWER 21 OF 52 COPYRIGHT 1993
AN (86)860454495 BIBLIODATA
TI Betriebssystem CP-M: vom Monitorprogramm zum Mehrbenutzersystem.
AU Plate, J*uergen
AS J*uergen Plate
PB M*uenchen : Franzis, 1986.
SO 2., neubearb. u. erw. Aufl. - 422 S. ; 23 cm. - ISBN 3-7723-7522-7 kart. : DM 58.00.
TS (Franzis-Computer-Praxis)
DN 86,A30,0413; 86,N19,0195

L1 ANSWER 22 OF 52 COPYRIGHT 1993
AN (86)860336883 BIBLIODATA
TI Das Handbuch des CP-M-2.2-Betriebssystems in englischer Sprache (CP, M operating system manual).
CS Digital Research <Pacific Grove, Calif.>
AS Digital Research
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 388 S. in getr. Z*aehlung ; 24 cm. - ISBN 3-89090-369-X Kunststoff : DM 90.00.
TS (Markt-&-Technik-Originaldokumentation)
DN 86,A27,0372; 86,N14,0154

L1 ANSWER 23 OF 52 COPYRIGHT 1993
AN (86)860275019 BIBLIODATA
TI Anwenderhandbuch CP-M-68 K: Einstieg in d. Betriebssystem f*uer d. 68000-Familie.
AU Plate, J*uergen
AS J*uergen Plate
PB M*uenchen : Franzis, 1986.
SO 144 S. ; 23 cm. - ISBN 3-7723-9751-4 kart. : DM 48.00.
TS (Franzis-Computer-Praxis)
DN 86,A26,0417; 86,N12,0235

L1 ANSWER 24 OF 52 COPYRIGHT 1993
AN (86)860223299 BIBLIODATA
TI Turbo-Pascal auf CP-M
AU Berendt, Gerhard
AS Gerhard Berendt
PB Neubiberg b. M*uenchen : Elektra, 1986.
SO 143 S. ; 21 cm. - ISBN 3-922238-63-7 kart. (Pr. nicht mitget.).
DN 86,A10,0479

L1 ANSWER 25 OF 52 COPYRIGHT 1993
AN (86)860222071 BIBLIODATA
TI TURBO-Pascal auf MUPID-CP-M
AU Br*ueckler, Helmut; Fritz, Werner
AS von Helmut Br*ueckler ; Werner Fritz
PB Mannheim ; Wien ; Z*uerich : Bibliographisches Institut, 1986.
SO X, 309 S. : graph. Darst. ; 21 cm. - ISBN 3-411-03131-X kart. : DM 34.00.
DN 86,A34,0451; 86,N09,0214

L1 ANSWER 26 OF 52 COPYRIGHT 1993
AN (86)860204650 BIBLIODATA
TI CP-M-Plus-Anwender-Handbuch CPC 6128, Joyce: e. unentbehrl. Nachschlagewerk f*uer d. prakt. Arbeit mit CP/M Plus u. seinen Hilfsprogrammen ; mit zahlr. Beispielen u. ausf*uehrl. systemspezif. Daten zur internen Speicherorganisation u. zu Schnittstellen.
AU H*ueckst*aedt, J*uergen
AS J*uergen H*ueckst*aedt
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 256 S. ; 24 cm. - ISBN 3-89090-197-2 Kunststoff : DM 46.00.
DN 86,A25,0568; 86,N09,0217

L1 ANSWER 27 OF 52 COPYRIGHT 1993
AN (86)860204642 BIBLIODATA
TI Programmentwicklung unter CP, M 2.2 auf dem CPC 464, 664: Beschreibung d. CP/M-Systems mit seinen BDOS/BIOS-Routinen ; Verbindung zum BASIC-Betriebssystem u. Schnittstellen ; Anwendungsbeispiele f*uer d. selbst*aendige Programmentwicklung.
AU Tischer, Helmut
AS Helmut Tischer
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 340 S. ; 24 cm. - ISBN 3-89090-209-X Pp. : DM 52.00.
DN 86,A21,0513; 86,N09,0228

L1 ANSWER 28 OF 52 COPYRIGHT 1993
AN (86)860075680/ BIBLIODATA
TI XCP-M-zwei-zwei-Anwender-HandbuchU
CP-M-2.2-Anwender-Handbuch: CPC 464/664/6128 ; e. unentbehrl. Nachschlagewerk f*uer d. prakt. Arbeit mit d. Betriebssystem CP/M 2.2. ; mit vielen CPC-systemspezif. Informationen: Speicheraufteilung, Schnittstellen, Sprungtab.
AU H*ueckst*aedt, J*uergen
AS J*uergen H*ueckst*aedt
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1986.
SO 212 S. ; 24 cm. - ISBN 3-89090-204-9 Pp. : DM 46.00.
DN 86,A15,0589; 86,N02,0138

L1 ANSWER 29 OF 52 COPYRIGHT 1993
AN (86)850918693 BIBLIODATA
TI Pocket guide CP, M <dt.>
Terminalbuch CP, M
AU Blackburn, Lawrence; Taylor, Marcus
AS von Lawrence Blackburn u. Marcus Taylor. XDt. *Uebers.: Thomas Fr*uehaufU
PB M*uenchen ; Wien : Oldenbourg, 1985.
SO 116 S. ; 24 cm. - ISBN 3-486-29331-1 kart. : DM 24.80.
DN 86,A07,0230; 85,N45,0154

L1 ANSWER 30 OF 52 COPYRIGHT 1993
AN (85)850744318 BIBLIODATA
TI CP, M: Einf. in d. popul*aerste Betriebssystem.
AU Reymann, Joseph
AS Joseph Reymann. X*Uebers.: J*uergen Janas ; Sabine JanasU
PB M*uenchen : Goldmann, Copyright 1984.
SO 1. Aufl. - 96 S. ; 18 cm. - ISBN 3-442-13117-0 kart. : DM 9.80.
TS (Goldmann ; 13117 : computer compact)
DN 85,A37,0318

L1 ANSWER 31 OF 52 COPYRIGHT 1993
AN (85)850297028 BIBLIODATA
TI Mastering CP/M <dt.>
Programmieren mit CP/M
AU Miller, Alan R.
AS Alan R. Miller. XDt. *Uebers.: Peter M*uehlbachU
PB Berkeley ; Paris ; D*uesseldorf : SYBEX, 1985.
SO 1. Aufl. - 418 S. ; 22 cm. - ISBN 3-88745-077-9 kart. : DM 52.00.
DN 85,A14,0391

L1 ANSWER 32 OF 52 COPYRIGHT 1993
AN (85)850110645 BIBLIODATA
TI CP-M kompakt: e. Kurzhandbuch d. Betriebssystems CP/M.
AU Plate, J*uergen
AS J*uergen Plate
PB M*uenchen : Franzis, 1985.
SO 87 S. ; 18 cm. - ISBN 3-7723-4011-3 kart. : DM 9.80.
TS (RPB-electronic-Taschenb*uecher ; N.F., Nr. 1)
DN 85,A16,0358; 85,N04,0189

L1 ANSWER 33 OF 52 COPYRIGHT 1993
AN (85)850078741 BIBLIODATA
TI Programmmieren unter CP, M mit dem C 64: TURBO-PASCAL, NEVADA-FORTRAN, MBASIC 80 ; e. Anleitung f*uer d. Implementierung von Standard-Programmiersprachen ; mit Schaltpl. zur eigenen Fertigung d. CP-M-Moduls.
AU Becker, Wolf-J*uergen; Folprecht, Miloslav
AS Wolf-J*uergen Becker ; Miloslav Folprecht
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1985.
SO 298 S. ; 23 cm. - ISBN 3-89090-091-7 kart. : DM 52.00.
DN 85,A41,0419; 85,N02,0213

L1 ANSWER 34 OF 52 COPYRIGHT 1993
AN (85)841097186 BIBLIODATA
TI CP-M revealed <dt.>
CP-M - Anatomie eines Betriebssystems: BDOS - BIOS - CCP ; Struktur vom Inhaltsverz., Disketten u. Dateien ; n*uetzl. Utilities u. Informationen, d. nicht im Handbuch stehen.
AU Dennon, Jack D.
AS Jack D. Dennon. Dt. *Uebers.: Peter Rosenbeck
PB Haar bei M*uenchen : Markt-und-Technik-Verlag, 1984.
SO 321 S. ; 23 cm. - ISBN 3-89090-087-9 kart. : DM 68.00.
TS (Computer pers*oenlich)
DN 85,A19,0302; 84,N48,0138

L1 ANSWER 35 OF 52 COPYRIGHT 1993
AN (85)840776632 BIBLIODATA
MAIN:
TI Basic aus der Praxis: geschrieben in MBASIC f*uer Ratlose, Einsteiger, Aufsteiger, Anwender.
AU Hillerkus, Karl W.
AS K. W. Hillerkus
PB Vaterstetten : IWT
SO 21 cm.
PART:
TI 4. Typische Programm-Beispiele f*uer Kaufleute
PB 1984.
SO 1. Aufl. - 281 S. - ISBN 3-88322-082-5 spiralgeh. : DM 44.00, sfr 44.00.
DN 85,A10,0481; 84,N32,0143

L1 ANSWER 36 OF 52 COPYRIGHT 1993
AN (84)840914849 BIBLIODATA
TI Einf*uehrung in die Anwendung des Betriebssystems CP,M: mit 76 Beispielen u. 48 *Uebungsaufgaben.
AU Schneider, Wolfgang
AS Wolfgang Schneider
PB Braunschweig ; Wiesbaden : Vieweg, 1984.
SO 1. Aufl., Nachdr. 1984. - 146 S. ; 23 cm. - ISBN 3-528-04252-4 kart. : DM 29.80.
TS (Programmieren von Mikrocomputern ; Bd. 9)
DN 84,A40,0416

L1 ANSWER 37 OF 52 COPYRIGHT 1993
AN (84)840908466 BIBLIODATA
TI CP-M-Anwenderhandbuch
AU Hogan, Thom
AS Thom Hogan
PB Hamburg ; New York ; SXainUt Louis ; San Francisco ; Auckland ; Bogota ; Guatemala ; Johannesburg ; Lissabon ; London ; Madrid ; Mexiko ; Montreal ; New Delhi ; Panama ; Paris ; San Juan ; Sao Paulo ; Singapur ; Tokio ; Toronto : MacGraw-Hill, 1984.
SO 303 S. : graph. Darst. ; 24 cm. - ISBN 3-89028-005-6 kart. : DM 39.80. - Literaturverz. S. 276 - 283.
DN 84,A40,0409

L1 ANSWER 38 OF 52 COPYRIGHT 1993
AN (84)840845294 BIBLIODATA
TI The CP-M handbook with MP-M <dt.>
CP-M-Handbuch
AU Zaks, Rodnay; Musstopf, G*uenter XBearb.U
AS Rodnay Zaks. XDt. *Uebers.: Klaus-J*uergen Schmidt. *Ueberarb.: G*uenter Musstopf u. Winfried Wolf. Zeichn.: Daniel LeNoury. Techn. Ill.: J. Trujillo SmithU
PB Berkeley ; Paris ; D*uesseldorf : SYBEX, 1984.
SO 5. Aufl., (2., *ueberarb. u. erw. Ausg.). - 315 S. ; 21 cm. - ISBN 3-88745-053-1 kart. (Pr. nicht mitget.). - Fr*uehere Aufl. u.d.T.: Zaks, Rodnay: CP-M-Handbuch mit MP-M.
DN 84,A37,0325

L1 ANSWER 39 OF 52 COPYRIGHT 1993
AN (84)840796382 BIBLIODATA
TI Betriebssystem CP-M: vom Monitorprogramm zum Mehrbenutzersystem.
AU Plate, J*uergen
AS J*uergen Plate
PB M*uenchen : Franzis, 1984.
SO 351 S. : graph. Darst. ; 24 cm. - ISBN 3-7723-7521-9 Pp. : DM 56.00.
TS (Franzis-Computer-Praxis)
DN 84,A45,0473; 84,N34,0185

L1 ANSWER 40 OF 52 COPYRIGHT 1993
AN (84)840609930 BIBLIODATA
TI MBASIC-Wegweiser f*uer Mikrocomputer unter CP, M und MS-DOS
AU Kaier, Ekkehard
AS Ekkehard Kaier
PB Braunschweig ; Wiesbaden : Vieweg, 1984.
SO X, 234 S. : graph. Darst. ; 23 cm.- ISBN 3-528-04294-X kart.: DM 48.00. - Mit 86 vollst. Programmen u. zahlr. Bildern.
DN 84,A42,0496; 84,N23,0134

L1 ANSWER 41 OF 52 COPYRIGHT 1993
AN (84)840078021 BIBLIODATA
TI Software-Schnellkurs CP, M: e. Lern- u. Nachschlagewerk.
AU Maass, Wolfgang
AS Wolfgang Maass
PB Haar bei M*uenchen : Verlag Markt u. Technik, 1984.
SO 79 S. ; 23 cm. - ISBN 3-922120-55-5 spiralgeh. : DM 37.00.
TS (Computer pers*oenlich)
DN 84,A15,0434; 84,N02,0175

L1 ANSWER 42 OF 52 COPYRIGHT 1993
AN (84)831221437 BIBLIODATA
TI Mikrocomputer-Betriebssysteme CP M, CDOS, DOS
AU Schmidt, Klaus-J*uergen; Renner, Gerhard
AS Klaus-J*uergen Schmidt ; Gerhard Renner
PB W*uerzburg : Vogel, 1984.
SO 1. Aufl., (unver*aend. Nachdr.). - 152 S.: Ill. ; 21 cm. - ISBN 3-8023-0655-4 kart. : DM 25.00.
TS (Chip-Wissen)
DN 84,A38,0333; 83,N50,0183

L1 ANSWER 43 OF 52 COPYRIGHT 1993
AN (84)831098201 BIBLIODATA
TI Einf*uehrung in die Anwendung des Betriebssystems CP, M: mit 76 Beispielen u. 48 *Uebungsaufgaben.
AU Schneider, Wolfgang
AS Wolfgang Schneider
PB Braunschweig ; Wiesbaden : Vieweg, 1983.
SO IX, 146 S. : graph. Darst. ; 23 cm. - ISBN 3-528-04252-4 kart. : DM 29.80.
TS (Programmieren von Mikrocomputern ; Bd. 9)
DN 84,A11,0465; 83,N44,0155

L1 ANSWER 44 OF 52 COPYRIGHT 1993
AN (84)831087382 BIBLIODATA
MAIN:
TI Basic aus der Praxis: geschrieben in MBASIC f*uer Ratlose, Einsteiger, Aufsteiger, Anwender.
AU Hillerkus, Karl W.
AS K. W. Hillerkus
PB Vaterstetten : IWT
SO 21 cm.
PART:
TI X2U. Programm-Beispiele f*uer kaufm*aennisch orientierte Berufe
PB 1983.
SO 1. Aufl. - 183 S. - ISBN 3-88322-042-6 spiralgeh. : DM 40.00.
DN 84,A06,0446; 83,N43,0163

L1 ANSWER 45 OF 52 COPYRIGHT 1993
AN (84)831078286 BIBLIODATA
TI CBASIC, CB 80: f*uer Personalcomputer mit CP/M-Betriebssystem ; e. Einf.
AU Possin, David
AS David Possin
PB Vaterstetten : IWT, 1983.
SO 1. Aufl. - IV, 187 S. : graph. Darst. ; 21 cm. - ISBN 3-88322-034-5 spiralgeh. : DM 56.00.
DN 84,A05,0539; 83,N43,0166

L1 ANSWER 46 OF 52 COPYRIGHT 1993
AN (84)831065206 BIBLIODATA
MAIN:
TI Basic aus der Praxis: geschrieben in MBASIC f*uer Ratlose, Einsteiger, Aufsteiger, Anwender.
AU Hillerkus, Karl W.
AS K. W. Hillerkus
PB Vaterstetten : IWT
SO 21 cm.
PART:
TI X1U. Typische Programm-Beispiele f*uer alle Berufe
PB 1983.
SO 1. Aufl. - 163 S. : graph. Darst. - ISBN 3-88322-031-0 spiralgeh. : DM 40.00.
DN 84,A06,0445; 83,N42,0175

L1 ANSWER 47 OF 52 COPYRIGHT 1993
AN (84)830912495 BIBLIODATA
TI Basic 80 and CP-M <dt.>
Basic 80 und CP-M: wie man Personal-Computer programmiert - e. Lehrbuch, d. Sie von d. Anf*aengen bis zur sicheren Beherrschung von BASIC begleitet.
AU Purdum, Jack Jay
AS Jack Jay Purdum. Dt. *Uebers.: Peter L*ueke
PB Haar bei M*uenchen : Verlag Markt u. Technik, 1983.
SO 296 S. ; 23 cm. - ISBN 3-922120-40-7 kart. : DM 48.00.
TS (Computer pers*oenlich)
DN 84,A14,0490; 83,N36,0210

L1 ANSWER 48 OF 52 COPYRIGHT 1993
AN (83)831143835 BIBLIODATA
TI Benutzeranleitung zum CP-M-Betriebssystem
AU Richter, Gerhard
AS G. Richter. FEOLL, Inst. f*uer Unterrichtswiss., Medieninformation
PB Paderborn : Forschungs- u. Entwicklungszentrum f*uer Objektivierte Lehr- u. Lernverfahren, 1981.
SO 25, 9 S. ; 30 cm. - kart. (Pr. nicht mitget.).
TS (Arbeitspapier / FEOLL : Mikrocomputer in d. Schule)
DN 83,B24,0253

L1 ANSWER 49 OF 52 COPYRIGHT 1993
AN (83)830994084 BIBLIODATA
TI Programmieren mit CP, M: Grundlagen ; Befehlsbeschreibungen ; CP/M-Assembler ; M-BASIC, MS-DOS.
AU Stepanienko, Reinhard
AS XR. Stepanienko. Mit Erfahrungsbericht von Prof. Dr. LauermannU
PB Holzkirchen : W. Hofacker, 1982.
SO 1. Aufl. - 122 S. : Ill. ; 21 cm. - ISBN 3-921682-92-4 kart.: DM 19.80.
DN 83,B21,0250

L1 ANSWER 50 OF 52 COPYRIGHT 1993
AN (82)820115045 BIBLIODATA
TI CP,M und Wordstar: Anwender-Handbuch ; dt.-sprachiges Handbuch f*uer d. Betriebs-Systeme CP,M u. MP,M u.d. Textverarbeitungs-System WordStar.
AU Paul, R*uediger; Riedel, Martin
AS R*uediger Paul ; Martin Riedel
PB M*uenchen : te-wi, 1981.
SO 1. Aufl. - 122 S. : graph. Darst. ; 21 cm. - ISBN 3-921803-14-4 kart. : DM 29.80.
DN 82,A05,0516

L1 ANSWER 51 OF 52 COPYRIGHT 1993
AN (82)820051969 BIBLIODATA
TI Vom Umgang mit CP-M: e. allgemein-verst*aendl. Einf.
AU Pol, Bernd
AS Bernd Pol
PB Vaterstetten bei M*uenchen : IWT-Verlag, 1982.
SO 1. Aufl. - XIII, 376 S. : Ill. ; 22 cm. - ISBN 3-88322-004-3 Pp. : DM 42.00.
TS (CP/M f*uer die Praxis ; 1)
DN 82,A42,0467; 82,N01,0205

L1 ANSWER 52 OF 52 COPYRIGHT 1993
AN (82)810804379 BIBLIODATA
TI CP-M und WordStar: Anwender-Handbuch ; dt.-sprachiges Handbuch f*uer d. Betriebs-Systeme CP/M u. MP/M u.d. Textverarbeitungs-System WordStar.
AU Paul, R*uediger; Riedel, Martin
AS R*uediger Paul ; Martin Riedel
PB Haar bei M*uenchen : Verlag Markt u. Technik, 1981.
SO 1. Aufl. - 122 S. : 9 graph. Darst. ; 21 cm. - ISBN 3-922120-10-5 kart. : DM 29.80. - In Deutsche Bibliographie, Reihe N, angezeigt u.d.T.: CP-M- und Wordstar-Anwender-Handbuch.
TS (Computer pers*oenlich ; 1)
DN 82,A33,0664; 81,N37,0303

Club80-Bibliothek

Stand: 14.02.1993
von Kurt Müller

Bedienungsanleitung zum Inhaltsverzeichnis der Club-Bibliothek

Nachfolgend ein Block aus der Club-80 Bücherliste mit Erklärung des Aufbaues. Wer Bücher ausleihen will, muß mir die Buch-Nr. mitteilen. Alle anderen Angaben sind optional. Besonders wichtig für den Empfänger ist die Gewichtsangabe, es kann so jeder nachrechnen, ob er eine Sackkarre mitnehmen muß oder ob es ein Esel (Ehefrau ?) auch tut.

Ansonsten möchte ich noch daran erinnern, daß es sicherlich keine Untugend ist, ausgeliehene Bücher nach "Verbrauch" sobald als möglich wieder zurückzuschicken - möglichst ohne zwei bis drei Mahnungen abzuwarten. Teilweise habe ich ausgeliehene Bücher erst nach einem Jahr zurückerhalten (das sind keine "Ausleihfristen", sondern grenzt bereits an "Sozialisierung", sprich Dauerleihe). Hier drängte sich mir in der Vergangenheit manchmal der Verdacht auf, als wolle man sich den Frust über den Mangel an Clubinfos auf dem Bücherwege "von-der-Seele-leihen".

14.02.93 Club80-Bücherliste Seite 1

```
Buch-Nr:  Aufl:Jahr:  Seiten: Gew.:              \ Texte sprechen für sich ...
2         1  1982     300     250                / Das Gewicht ist in Gramm angegeben !
Markt & Technik                                  } Buchverlag, Fehler von mir, da 2-mal
Taschenlexikon für alle Personal-Comp.           } Buchtitel
Michael Scharfenberger                           } Name des Autors
Markt & Technik                                  } Buchverlag
Kurzbeschreibung:                                } spricht für sich ...
Recht brauchbar, da die ganz alten Da-           } Eine kurze Inhaltsbeschreibung. Nicht
tenschleudern aus der goldenen Computer-         } immer vorhanden, aber immer öfter ...
ära ausführlich behandelt werden.                }
```

Clubbücherei
Kurt Müller, Sophie-Scholl-Ring 3b, 2054 Geesthacht
Tel.: 0 41 52 / 7 06 43

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
2 1 1982 300 250
Markt & Technik
Taschenlexikon für alle Personal-Comp.
Michael Scharfenberger
Markt & Technik
Kurzbeschreibung:
Recht brauchbar, da die ganz alten Datenschleudern aus der goldenen Computerära ausführlich behandelt werden.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
4 1 0 66 100
Vogel-Verlag
Mein Homecomputer - Eine Verbraucherfibl
./.
Vogel-Verlag
Kurzbeschreibung:
Einige Tips & Erklärungen rund um den Computer

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
6 1 1983 125 150
Virgin Books, USA
Games for your TRS-80
./.
Virgin Books, USA
Kurzbeschreibung:
Einige Spiele in BASIC, Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
8 1 1980 196 480
Creativ Computing Press
More BASIC Computer Games
David H. Ahl
Creativ Computing Press
Kurzbeschreibung:
Spiele satt, aber in Englisch !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
10 1 1984 128 350
Franzis' Verlag
Matrix-Operationen
Rudolf Busch
Franzis' Verlag
Kurzbeschreibung:
Eine Software-Sammlung in BASIC

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
3 2 1983 300 400
Vogel-Verlag
Computersp. & Knobeleien progr. in BASIC
Rüdeger Baumann
Vogel-Verlag
Kurzbeschreibung:
Der Titel spricht für sich. Sehr gut aufgemacht.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
5 1 1982 148 250
W. Hofacker Verlag
Programmieren mit dem ZX81
E. Flögel
W. Hofacker Verlag
Kurzbeschreibung:
Wie benutze ich ein Spielzeug zum Programmieren. DAs Buch enthält ein bunte Sammlung an BASIC-Progr. zu den unterschiedlichsten Themen.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
7 3 1979 142 220
Dilithium Press, Portland, Oregon, USA
Introduction to TRS-80 Graphics
Don Inman
Dilithium Press, Portland, Oregon, USA
Kurzbeschreibung:
Es wird sehr gut auf die Technik der Grafikprogr. eingegangen. Die Vorgehensweise & Technik wird anhand guter Beispiele erklärt. Englischkenntn. erfordl.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
9 1 1984 104 300
Franzis' Verlag
Dateien, Listen & Verzeichnisse
Rudolf Busch
Franzis' Verlag
Kurzbeschreibung:
Eine Software-Sammlung in BASIC. Spiele & etc.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
11 7 1980 267 390
Delithium Press Beaverton, Oregon, USA
TRS-80 Programms
Tom Rugg & Phil Feldmann
Delithium Press Beaverton, Oregon, USA
Kurzbeschreibung:
Diverse Programme in BASIC mit Erläuterungen (Spiele etc.) Englischkenntnisse erforderlich !

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
12 1 1983 110 350
nicht bekannt !
Programme & Tricks für Genie I & II
Clemens Becher, Frank Seger
nicht bekannt !
Kurzbeschreibung:
Jede Menge Listings in BASIC (Spiele etc.)

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
16 4 1985 89 500
Franzi' Verlag
Das Modem Sonderheft
./.
Franzi' Verlag
Kurzbeschreibung:
Das Modem-Sonderheft mit Grundlagen & Schaltungen.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1985 58 200
TCS Computer GmbH
Genie Data
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Anwenderzeitung für Genie-Besitzer

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1985 42 125
TCS Computer GmbH
Genie Data
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Anwenderzeitung für Genie-Besitzer

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1984 58 200
TCS Computer GmbH
Genie Data
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Anwenderzeitung für Genie-Besitzer

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
13 1 1984 70 250
Franzis' Verlag
Alles über Peek & Poke
Heiko Requardt
Franzis' Verlag
Kurzbeschreibung:
Eine Software-Sammlung zum Thema Tastatur.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1984 66 200
TCS Computer GmbH
Genie Data
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Eine Art Club-Zeitung für Besitzer eines Genie 1/2/3 Computers

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1984 54 175
TCS Computer GmbH
Genie Data
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Anwenderzeitung für Genie-Besitzer

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1984 24 75
cc Computer Studio GmbH
Genie Data
Fa. TCS Computer
cc Computer Studio GmbH
Kurzbeschreibung:
Anwenderzeitung für Genie-Besitzer

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1985 58 150
TCS Computer GmbH
Genie Data
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Anwenderzeitung für Genie-Besitzer

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
17 1 1984 62 200
TCS Computer GmbH
Genie Data (TCS Computer GmbH)
Fa. TCS Computer
TCS Computer GmbH
Kurzbeschreibung:
Eine Art Club-Zeitung für Besitzer eines Genie 1/2/3 Computers

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
19 1 1981 288 650
IJG Inc. USA
BASIC Faster & Better & o. Mysteries
Lewis Rosenfelder
IJG Inc. USA
Kurzbeschreibung:
Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
22 1 1982 300 650
IJG Inc. USA
TRSDOS 2.3 Encoded & other Mysteries
James Lee Farvour
IJG Inc. USA
Kurzbeschreibung:
Das gesamte Betriebssystem offengelegt !
Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
24 1 1981 306 490
John Wiley & Sons, Inc.
TRS-80 Data File Programming M-1/3
Radio Shack
John Wiley & Sons, Inc.
Kurzbeschreibung:
Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
27 2 1980 351 550
John Wiley & Sons Inc., USA
TRS-80 Level-II BASIC
Robert L. Albrecht
John Wiley & Sons Inc., USA
Kurzbeschreibung:
Das Handbuch zum Level-II BASIC.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
18 1 1983 430 1000
IJG Inc. USA
BASIC Disk-I/O faster & better
Lewis Rosenfelder
IJG Inc. USA
Kurzbeschreibung:
Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
20 1 1982 335 800
IJG Inc. USA
The Cutom TRS-80 & other Mysteries
Dennis Bathory
IJG Inc. USA
Kurzbeschreibung:
Wer gern an seinem TRS-80 bastelt liegt mit diesem Buch genau richtig. Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
23 1 1982 124 350
Interface Age, Münschen
TRS-80 Disk & andere Geheimnisse
H. C. Pennington
Interface Age, Münschen
Kurzbeschreibung:
Endlich einmal etwas über den TRS-80 in deutsch !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
25 1 1981 224 260
RAdio Shack
Programming Technics for L-II BASIC
William Barden, Jr.
RAdio Shack
Kurzbeschreibung:
Englischkenntnisse erforderlich ! Wer mit dem Enlischen zurechtkommt, findet eine gute Grundlageneinführung in die Programmierung des TRS-80 vor.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
28 1 1980 342 800
Radio Shack
TRS-80 Programming Style
Thomas Dwyer
Radio Shack
Kurzbeschreibung:
Englischkenntnisse erforderlich

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
29 1 1979 56 600
Ing. W Hofacker Verlag
The First Book of 80-US
80-US
Ing. W Hofacker Verlag
Kurzbeschreibung:
Ein Re-Print einiger Ausgaben der US-Zeitschrift 80-US

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
31 1 1979 144 380
Franzis-Verlag
Z80-Applikationsbuch
Michael Klein
Franzis-Verlag
Kurzbeschreibung:
Eine Einführung in die Programmier- & Interfacetechnik zum Z80

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
33 1 1980 606 800
Sybex-Verlag
Programmierung des Z80
Rodney Zaks
Sybex-Verlag
Kurzbeschreibung:
Die Bibel zum Z80-Prozessor

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
35 1 1981 185 300
W. Hofacker Verlag
Anwenderprogramme für TRS-80
M. Stübs
W. Hofacker Verlag
Kurzbeschreibung:
Eine Sammlung von Geschäftsprogr. Utilities in BASIC für den TRS-80.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
36 2 1981 202 275
W. Hofacker Verlag
Programmieren mit dem TRS-80
M. Stübs
W. Hofacker Verlag
Kurzbeschreibung:
Einführung in den TRS-80 u. seine Programmierung nebst einigen BASIC-Programmen.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
30 1 1983 236 380
Wayne Green Books
The Rest of 80
./.
Wayne Green Books
Kurzbeschreibung:
Ein Software-Querschnitt rund um den TRS-80. Englischkenntnisse erforderlich!

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
32 1 1980 424 450
Ing. W. Hofacker
Z80-Assembler Handbuch
Elcomp
Ing. W. Hofacker
Kurzbeschreibung:
Einer Auflistung aller Z80-Befehle mit Erläuterungen.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
34 1 1984 299 400
Vogel-Verlag
Daten - Disketten - Dateien
Hans-Joachim Sacht
Vogel-Verlag
Kurzbeschreibung:
Effektive Dateienverwaltung mit MBASIC

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
36 2 1981 202 275
W. Hofacker Verlag
Programmieren mit dem TRS-80
M. Stübs
W. Hofacker Verlag
Kurzbeschreibung:
Zweiter Band mit gleichem Inhalt.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
38 1 1984 222 330
W.-D. Luther Verlag
Textsystem in BASIC & Z80-Assembler
Gerd Scheller
W.-D. Luther Verlag
Kurzbeschreibung:
Wie No. 36. aber deutlich verbesserte Version des TESY-Textsystems. Auf einem TRS-80 geschrieben.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
39 1 1985 145 250
W.-D. Luther Verlag
Textsystem in BASIC & Z80-Assembler
Wolfgang Weber
W.-D. Luther Verlag
Kurzbeschreibung:
Wer gern an seiner "eigenen" Textverarbeitung "basteln" möchte, kann sich hier nach belieben austoben. Geschrieben für das Level-2 BASIC des TRS-80

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
41 1 1984 285 450
Vogel-Verlag
Start in die Computergrafik
Fred Wagenknecht
Vogel-Verlag
Kurzbeschreibung:
Viele kurze Progr.. die aufzeigen. wie man mit der TRS-80 Klötzchengrafik Grafik erzeugen kann mit guten Erläuterungen.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
43 1 1984 222 350
Elektor-Verlag
BASICODE - Die komp. Computersprache
M. Wiegand. M. & H. Fillinger
Elektor-Verlag
Kurzbeschreibung:
Die kompatible Computersprache. 14 Prog. für über 25 Homecomputer. Enthält viele BASIC-Listings (allg. Themen)

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
45 1 1980 183 460
Creativ Computing Press
BASIC Computer Games
David H. Ahl
Creativ Computing Press
Kurzbeschreibung:
Spiele satt. aber in englisch !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
47 1 1979 221 275
Radio Shack
TRS-80 Assembly-Language Programming
William Barden. Jr.
Radio Shack
Kurzbeschreibung:
Text in Englisch !

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
40 1 1984 145 400
Birkhäuser Verlag
Einfache Peripheriegeräte im Selbstbau
Owen Bishop
Birkhäuser Verlag
Kurzbeschreibung:
Es werden einige Hardware-Projekte wie z.B. eine Kaffemschinen-Steuerung, EPROM Karte etc beschrieben. Durchaus anregen.

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
42 1 1976 180 460
Radio Shack
Level-II BASIC Reference Manual
Radio Shack
Radio Shack
Kurzbeschreibung:

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
44 1 1983 112 150
Franzis' Verlag
Kleiner BASIC-Wortschatz
Rufdolf Busch
Franzis' Verlag
Kurzbeschreibung:
Eine Kommentierte Auflistung von BASIC-Befehlen mit Beispielen.

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
46 1 1980 350 400
Tab Books Inc.
PASCAL
David L. Heisermann
Tab Books Inc.
Kurzbeschreibung:
Eine Einführung in PASCAL und wie man diese Programmiersprache richtig anwendet. Viele Listings. Englischkenntnisse erforderlich !

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
49 1 1981 192 430
McGraw-Hill
Some Common BASIC Programms
Lon Poole, Mary Borchers, Karl Koessel
McGraw-Hill
Kurzbeschreibung:
Enthält etliche interessante Programme

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
51 2 1986 288 700
Markt & Technik
Der Einstieg in C

Paul M. Chirlian
Markt & Technik
Kurzbeschreibung:
Soll eine Einweisung in C sein. Es gibt besseres zu diesem Thema.

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
54 2 1986 443 600
Vieweg Verlag
Mikrocomputer Aufbaukurs
E. Kaier, E. Rudolfs
Vieweg Verlag
Kurzbeschreibung:
Das Buch informiert durch ausführliche Beispiele grundlegend über die Anwendungsmöglichkeiten die Turbo Pascal unter CP/M, MSDOS und MSX-DOS bietet.

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
56 1 1987 682 1000
Markt & Technik
Das Z80-Buch
Eberhard Zehender
Markt & Technik
Kurzbeschreibung:
Die Z80-Bibel ! Es werden hier querbeet alle für den Z80 relevanten Gebiete in Assembler programmiert, sogar Gleitkomma routinen !

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
58 1 1985 111 190
W.-D. Luther Verlag
Listing-Verwaltung in MBASIC
H.-J. Rothe
W.-D. Luther Verlag
Kurzbeschreibung:
Ein kompl. BASIC-Progr. zum Verwalten von Zeitschriften-Artikeln aller Art. Ausführliche Erkl. mit Beispielausdrucke

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
61 1 1984 58 150
Vieweg Verlag
PASCAL - Datenstrukt. & prakt. Anwendg.
D. Herrmann
Vieweg Verlag
Kurzbeschreibung:
Die Programmierung von div. Algorithmen Sehr ausführlich m. Flußdiagramm und Ergebnisausdrucken.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
52 6 1986 303 500
Oldenburg-Verlag
PASCALund PASCAL-Systeme

Rudolf Herschel
Oldenburg-Verlag
Kurzbeschreibung:
Beschreibung der Programmierspr. PASCAL &

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
55 2 1986 295 300
Teubner Studuienskripten
Programmierung in Modula-2
Dal Cin, Lutz Risse
Teubner Studuienskripten
Kurzbeschreibung:
Eine Einführung in das modulare Progr. mit MODULA-2. Hervorragender Inhalt ! Eignet sich zum Einarbeiten in das Konzept der Sparache.

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
57 3 1985 321 600
Markt & Technik
CP/M - Anatomie eines Betriebssystems
Jack D. Dennon
Markt & Technik
Kurzbeschreibung:
Ein Wissen-Torpedo mit Informationen zum Thema CP/M. Sehr brauchbar.

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
59 1 1984 91 120
W.-D. Luther Verlag
dBASE-II im Überblick
Thomas LAuer
W.-D. Luther Verlag
Kurzbeschreibung:

===

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
62 1 1979 80 150
Frech-Verlag Stuttgart
BASIC - Grundlagen & Beispiele
Günter Abeldt
Frech-Verlag Stuttgart
Kurzbeschreibung:
Es werden mathematische Probleme behandelt. Empfehlenswerter Inhalt !

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
63 1 1985 224 330
W.-D. Luther Verlag
Program. in MBASIC unter CP/M
Wilfried Plieth
W.-D. Luther Verlag
Kurzbeschreibung:
Tips & Trick zum optimaleren Programmieren in BASIC unter CP/M. Viele Beispiel-Listings !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
66 1 1982 136 290
Markt & Technik
Personal Computer Lexikon
Computer Persönlich
Markt & Technik
Kurzbeschreibung:
Wie der Titelö es schon sagt. Der Inhalt erscheint mir hinreichend vollständig. Ein "richtiges" Computerlexikon (von Duden-Verlag) ist jdoch besser.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
68 4 1979 295 350
W. Hofacker Verlag
Z8, Z80, Z8000 Mikrocomputer-Technik
H.-P. Blomeyer-Bartenstein
W. Hofacker Verlag
Kurzbeschreibung:
Es werden der Z8, Z80 & Z8000 behandelt

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
70 1 1979 199 250
R. Oldenbourg Verlag
Tischcomputer - BASIC für Anfänger
Juilius Schärf
R. Oldenbourg Verlag
Kurzbeschreibung:
Ein sehr gutes Übungsbuch für Anfänger, die sich fundiert in BASIC einarbeiten wollen. Empfehlenswert !

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
72 1 1981 55 120
80-Northwest Publishing, Inc.
Disk Interfacing Guide (TRS-80 M-1)
William Barden, Jr.
80-Northwest Publishing, Inc.
Kurzbeschreibung:
Eine Auslassung über das Diskinterface des TRS-80.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
64 1 1984 190 120
W.-D. Luther Verlag
RPNL - Eine FORTH-ähnliche Sprache
Gustav Wostrack
W.-D. Luther Verlag
Kurzbeschreibung:
Der Compiler ist in Z80-Assembler abgedruckt. Läuft in der abgedruckten Form unter CP/M.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
67 2 1980 245 475
Vogel-Verlag
Mein Computer 2^10 Fragen & Antworten
Rich Didday
Vogel-Verlag
Kurzbeschreibung:
Viele Fragne mit Antworten in Form eines bebilderten Computerlexikons.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
69 2 1984 270 275
Röckrath Verlag
Das DOS-Buch
Hartmut Grosser
Röckrath Verlag
Kurzbeschreibung:
Wer dieses Buch nicht kennt ...Eine fundierte Erklärung von NewDOS80, GDOS, und Colour-DOS für den TRS-80

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
71 2 1981 167 225
R. Oldenbourg Verlag
BASIC-Übungen für Anfänger
Julius Schärf, Tobias Schläpfer
R. Oldenbourg Verlag
Kurzbeschreibung:
Ein gutes Buch zum Einarbeiten ins Programmieren mit BASIC. Allein die vielen Beispiele sind ein Blick wert.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
73 1 1983 110 200
W. Hofacker Verlag
Progr. in Assembl. mit Z80, Bd 2
Dr. E. Schmitter
W. Hofacker Verlag
Kurzbeschreibung:
Eine kleine Befehlsreferenz mit Beisp.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
74 1 1986 223 200
Verlag Die Wirtschaft
Der Personal-Computer 1715
Prof. Dr.-Ing. Rolf Zeth
Verlag Die Wirtschaft
Kurzbeschreibung:
Eine Art Bedienerhandbuch zum Robotron 1715

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
76 1 1988 112 140
Militärverlag d. DDR
Mikrorechner in der Amateurmeßtechnik
A. Bogatz
Militärverlag d. DDR
Kurzbeschreibung:
Interface-Schaltungen in TTL (DDR-Symbole, deshalb gewöhnungsbedürftig !)

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
78 1 1988 65 150
VEB Verlag Technik Berlin
radio fernsehen elektronik nr.11
./.
VEB Verlag Technik Berlin
Kurzbeschreibung:
Nr. 11, Elektronikzeitschrift

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
80 1 1988 50 100
Zeitschrift der GST
Funkamateur 6/88
./.
Zeitschrift der GST
Kurzbeschreibung:

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
82 2 1990 224 250
VEB Verlag Technik, Berlin
CP/M in der Praxis
Joachim Bernert
VEB Verlag Technik, Berlin
Kurzbeschreibung:
Beschreibt den Umgang mit CP/M und seiner Kommandos.

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
75 2 1988 272 300
Militärverlag der DDR
Praktische Mikrocomputertechnik
Manfred Kramer
Militärverlag der DDR
Kurzbeschreibung:
Das Buch zeigt ausführlich den Bau von Rechnerhardware für den Z80. Mit ausführlicher Beschreibung und Listings.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
77 1 1988 224 300
Verlag Technik der DDR
CP/M in der Praxis
Joachim Bernert
Verlag Technik der DDR
Kurzbeschreibung:
Beschreibung von CP/M und einigen Anwendungen mit Beschreibung.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
79 1 1988 47 100
Zeitschrift der GST
Funkamateur 11/88
./.
Zeitschrift der GST
Kurzbeschreibung:

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
81 1 1989 240 300
VEB-Verlag Technik, Berlin
W. Grafik
Wilfried Grafik
VEB-Verlag Technik, Berlin
Kurzbeschreibung:
Es wird der Umgang mit DBase III beschrieben.

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
83 1 1979 300 700
IJG Inc., USA
TRSDOS 2.3 Decoded & other Mysteries
James Lee Farvour
IJG Inc., USA
Kurzbeschreibung:
Es wird das Betriebsystem TRSDOS 2.3 in seinen SYS-Modulen beschrieben. Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
84 1 1982 158 250
Luther-Verlag
BASIC im Büro
W. Brümmer, D. Hemmerling, H. Fritsch
Luther-Verlag
Kurzbeschreibung:
BASIC-Programme mit ausführlicher Programmbschreibung

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
86 1 1984 129 123
./.
TRS-80 & Video-Genie ROM-Listing
Luidger Röckrath
./.
Kurzbeschreibung:
Das komplette ROM-Listing für den TRS-80

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
88 1 1988 128 350
VEB Verlag Technik, Berlin
Programmieren mit C
Matthias Clauß
VEB Verlag Technik, Berlin
Kurzbeschreibung:
Das Buch beschreibt ausführlich die Anwendung der Sprach C. Gut aufgemacht.

==

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
85 1 1979 180 340
IJG Inc. USA
TRS-80 Disk & other Mysteries
H. C. Pennington
IJG Inc. USA
Kurzbeschreibung:
Englischkenntnisse erforderlich !

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
87 2 1984 402 500
Ing. W. Hofacker
Programmieren in Maschinenspr. mit Z80
P. Immerz
Ing. W. Hofacker
Kurzbeschreibung:

==

Buch-Nr: Aufl:Jahr: Seiten: Gew.:
89 2 1989 240 123
VEB Verlag Technik Berlin
MS-DOS
J. Hübener
VEB Verlag Technik Berlin
Kurzbeschreibung:
Wer mit PC's arbeitet und Details zu MS-Do DOS sucht liegt hier richtig.

==

Protokoll einer Mailboxsession

Informationen zu DFUE

E.Schröer

Durch einen Zahlendreher gelangte ich per Pattex-D zum Host des Neusser Computer Club. Mein (ex USA) Terminalprogramm protokollierte braf die kurze Session mit, hatte allerdings bei einigen MSG's Schwierigkeiten mit den deutschen Sonderzeichen. Die MSG's fand ich von allgemeinen (Club) Interesse, sodaß ich hier den Text mal zur Verfügung stelle. Eine vollständige Nachbearbeitung wäre etwas aufwendig gewesen, somit seien noch kleine Fehler verziehen; aber ich denke das man den Text noch gut verstehen kann.

```
              S t a r t   i n ' s   N C C - N E T   :

N     N    CCCCC     CCCCC          N     N   EEEEEEE   TTTTTTT
NN    N   C     C   C     C         NN    N   E            T
N N   N   C         C               N N   N   E            T
N  N  N   C         C         ===   N  N  N   EEEEE        T
N   N N   C         C               N   N N   E            T
N    NN   C     C   C     C         N    NN   E            T
N     N    CCCCC     CCCCC          N     N   EEEEEEE      T

DATEX-P NUA: 0262 45 2131 20133                      24h online
Phone 49 (0) 2131 277799   from 300 up to 14400 baud v42bis

==================================================================
                   W E L C O M E  to the biggest German BBS
==================================================================
 The BBS runs with 22 phonelines up to 38.400 Baud (MNP5 V42bis),
                   8 X.25 lines with   4.800 Baud (X.28)
 Hardware: IBM PC 486 Eisa with 48 MB Memory and 2.1 Gigabyte
           Harddisk

 Open your new account with LI or use ? for HELP

Gast, Du bist der 23550. Anrufer auf Port ttyB09
```

```
Befehl:
?
               * * * * * * *  N C C - M E N U  * * * * * * *

AM = Allgemeine Mails                          AS = Assembler
BA = Bedienungsanleitung                       BU = Buhmann
CE = Clubecke
CI = Clubinfo                                  DE = Diskussionsecke
DF = Dfue                                      DS = Disk Space
EM = Electronic Mail                           FG = Fundgruben
FI = Fido Bretter                              FM = File Menu (Up &
                                                    Download)
IN = Info                                      KE = KulturEcke
KW = Komerz. Werbung                           MI = Mitfahrgelegenheit
ML = Mailbox Liste                             PS = Programmier-Sprachen
SP = Sportecke                                 SY = Sysop Board
TB = Technik Board                             TD = ToDay
TT = Tips & Tricks                             UG = User Gruppen
UX = Unixecke                                  WI = Witzecke
CO = Color on / off                            CR = ChatRequest  on / off
CH = CHAT ( Multiuser )                        FM = FileMenue
OG = Online Games                              PM = Private Mail
LI = Login / Userantrag stellen                LO = LogOff
NW = News seit dem letzten Login               ON = Wer ist ONLINE ?
UL = Userliste                                 US = User-Setup
VK = Visitenkarten                             VT = Voting Tool

Befehl:
ML
Mailbox Liste
1 = Nicht vorhanden
2 = Mailboxliste Bereich 02....
3 = Nicht vorhanden
4 = Nicht vorhanden
5 = Nicht vorhanden
6 = Nicht vorhanden
7 = Nicht vorhanden
8 = Nicht vorhanden
9 = Nicht vorhanden

X = Exit
Bitte waehlen:
2

  < 1 > = lesen     / read  / ler
  < 4 > = string suchen / search / procurar
  < 5 > = Inhaltsverzeichnis anzeigen
  < X > = Exit

Bitte waehlen:
?
MSG Nr. 01  von MIGHTY vom 31.10.1992  Betr. 02xxx

Befehl:

 - ? / Hilfe / Help / Ajuda -
```

```
Befehl:
1

 - ? / Hilfe / Help / Ajuda -

Befehl:
ML
Mailbox Liste
1 = Nicht vorhanden
2 = Mailboxliste Bereich 02....
3 = Nicht vorhanden
4 = Nicht vorhanden
5 = Nicht vorhanden
6 = Nicht vorhanden
7 = Nicht vorhanden
8 = Nicht vorhanden
9 = Nicht vorhanden

X = Exit
Bitte waehlen:
2

   < 1 > = lesen      / read   / ler
   < 4 > = string suchen / search / procurar
   < 5 > = Inhaltsverzeichnis anzeigen
   < X > = Exit

Bitte waehlen:
?
MSG Nr. 01  von MIGHTY vom 31.10.1992  Betr. 02xxx
BEREICHsliste 02xxx
Sortiert nach: Orten


----- AACHEN ----------------------------------------------------
0241    24378       Cheops              2400   PC-Net

----- ALPEN -----------------------------------------------------
02802   1824        Midimail           16800   MagicNET

----- BAD HONNEF ------------------------------------------------
02224   89432       TECHNICS            2400   TechnoNet

----- BERGHEIM --------------------------------------------------
02271   44730       Erftbox            16800
02271   61650       BPB                 2400                ab 20.00

----- BERGISCH GLADBACH -----------------------------------------
02204   21356       SPEED-BOX           2400   Art-Net

----- BOCHUM ----------------------------------------------------
0234    17796       ABAKUS              2400
0234    231764      PCB                 2400   APNET
0234    294715      KKB                 2400
0234    294715      KKB                 2400   Aquarius
0234    341755      ONLINE BBS          9600   Fido
0234    412240      PENTA               2400   PC-Net
0234    430325      CONTACT             2400   AmNet 2
0234    770457      Saphir

----- BONN ------------------------------------------------------
0228    254020      MAUS BONN           2400   Maus         BN
0228    259146      Rubin                      PC-Net
0228    259147      Diamant             9600   PC-Net
0228    317752      TERRANIA CITY      14400   Fido
0228    4009742     BUND GESCHAEFTS     2400   Zerberus     BUNDBonn
0228    628516      BUNDESWEHR           300
0228    638217      GRUENE-BV           2400   Zerberus     GRUEN-BV
0228    680509      NEPTUN              2400
0228    856349      WASSERTURM          1200   ProNET

----- BOTTROP ---------------------------------------------------
02041   27585       Westworld          14400   18-21.30
----- BRHL ------------------------------------------------------
02232   28112       SPACEBALL           2400   Art-Net
02232   29681       Jungle             16800   PC-Net
02232   31591       HBB-BRUEHL          2400   Art-Net
02232   47805       ATLANTA             2400   PC-Net

----- CASTROP-RAUXEL --------------------------------------------
02305   23916       CODS                2400   PC-Net
02305   82045       TELE DATA LINE      1200

----- DINSLAKEN -------------------------------------------------
02064   13795       HIS-Box            14400   MagicNET
02134   18205       Atlantis           14400   MagicNET GIGA
02134   98380       DDS-Box             2400   MagicNET Art-Net

----- DORMAGEN --------------------------------------------------
02133   62224       Phobos              2400   ANET SUPPORT-MAILBOX

----- DORTMUND --------------------------------------------------
0231    145779      COLT BOX DO         2400   Star Net
0231    146121      HOLIDAY BOX         2400                22-10 Uh
0231    171725      GLOBALDATA          2400   Zerberus     GLOBAL
0231    174674      MAUS DORTMUND       2400   Maus
0231    271871      TRASH               1200
0231    356030      Forum               2400   PC-Net
0231    401550      RIAS 2 LINE 1       2400
0231    401559      RIAS 2 LINE 2       2400
0231    403668      QBBS RIAS 1         2400   Fido
0231    409594      RIAS 2 LINE 3       2400
0231    409597      RIAS 2 LINE 4       9600   Fido
0231    458665      BLACK HOLE          2400   Fido
0231    458732      MEGA MAIL            300
0231    468767      DO-MAIL             2400   Fido
0231    47126       BONNY MAILBOX S     9600   Fido
0231    512636      TPBOARD ATLANTI     2400   Fido
0231    601502      MELMAC              2400   APNET
0231    616372      SC BOX              2400
0231    616383      SVENNYS BOX         2400
0231    752694      BOSKOPP             2400   UUCP
0231    753099      BIG BOX             2400
```

0231	753099	BIG BOX	2400	Aquarius	
0231	822008	MAMNIX	2400	UUCP	PASSWORT
0231	824577	AIB DORTMUND	2400	AmNet 2	
0231	827059	MOONLIGHT	2400	UUCP	
0231	827872	FISCHLADEN	2400		15-5 Uhr
0231	8280279	DAS OFFENE OHR	2400	Zerberus	DOO
0231	894460	CO-DO 2	2400	AmNet 2	

----- DREN ---

02421	66365	Positron	9600	PC-Net

----- DSSELDORF ---

0211	212494	NADESCHA	2400	FREE-NET	
0211	236995	KARLCHEN	2400		
0211	237817	KEY TO PARADISE	2400	Fido	
0211	2480296	Mode Line	14400		
0211	319412	Attach	9600	MagicNET	
0211	346008	KOKORO	2400	FREE-NET	Server
0211	369599	SILENTS	2400	ANET	
0211	375191	FUTURE	9600	APNET	
0211	393628	Electric Dreams	16800	Zerberus	
0211	4180259	DUNE	2400	Zerberus	DUNE
0211	4542766	FFB	2400	ANET	
0211	465698	TOM'S BOX	2400		
0211	490121	ARTWORKS	2400		
0211	613764	Don`s Box	16800	ZENtron	
0211	615100	Omegabox	14400	MagicNET	
0211	6588940	HOTLINE	2400	AmNet	
0211	705817	DAB	2400	AmNet	
0211	7270519	ATLANTIS	2400	APNET	
0211	783178	WEL-COM		COM-NET	

----- DUISBURG --

0203	26115	PC-DMC	2400		
0203	408799	RADIO CHAOS	14400	Fido	
0203	475277	HAMMERBOX	2400		
0203	559499	TERRASOFT		COM-NET	
0203	63201	Offline	2400	ZAK COM	Zer.MagicNET Gay
0203	701806	IBM USER SYSTEM	2400	Zerberus	IUS
0203	773263	IMABOX PORT 2	2400		
0203	773664	IMABOX	2400		
0203	780344	BLACK TOWER	4800	Fido	2:245/7
0203	790841	STARLINE	2400	FREE-NET	
02136	370389	HCB		COM-NET	

----- ERFTSTADT ---

02235	76392	Chappi	2400	PC-Net

----- ERKRATH ---

02104	43653	Arboleda	9600	MagicNET
02104	47915	Funmail	14400	MagicNET
0211	2480269	Heiwoe	9600	MagicNET

----- ESSEN ---

0201	225118	Snapshot	16800		
0201	232899	APOLONIA	2400	Fido	
0201	233249	PUMA	2400	ProNET	
0201	256885	ELOI	2400	Zerberus	ELOI
0201	288200	AXEL'S BOX	9600	Fido	
0201	303899	HEADCRASH	2400		
0201	312400	BL BOX	2400		Test
0201	483735	MUG	9600	Fido	
0201	494959	CAMEL	9600	Fido	
0201	501952	CENTURY	2400	ProNET	
0201	627018	MILLIWAYS	2400	ProNET	
0201	711882	DIGILINK	2400	ProNET	
0201	734045	CREDO	2400	Zerberus	CREDO
0201	764731	Lemmings !	9600		
0201	787425	Blackout	14400		

----- EUSKIRCHEN --

02251	4566	CRAISH	2400	ANET
02251	74173	DIAMANT	2400	PC-Net

----- FRECHEN ---

02234	23238	Carthago	2400	PC-Net	
02234	22983	FKK	300		C-64
02234	3332	JINXED	2400	PC-Net	
02234	48695	DISKETTE	2400		
02234	49086	SPARK	2400	PC-Net	
02234	57646	Waterloo	9600	PC-Net	
02234	58234	ABACUS	2400		
02234	64751	COCOS E.V.	2400		
02234	76495	SeaSide			

----- GELSENKIRCHEN ---

0209	130194	Viking	16800	
0209	135269	CBFC	2400	
0209	146372	DISCOVERY	2400	ProNET
0209	148235	Century	16800	
0209	273094	Ost-West	2400	Branchenadressen in OST-D
0209	53559	Hase	14400	MagicNET
0209	54920	SHIVA	2400	UUCP
0209	598560	NO NAME	19200	Zerberus
0209	815885	Space Center	2400	22:00-5:00
0209	873022	Willi_S	14400	
0209	873547	COLUMBIA	2400	

----- GEVELSBERG --

02332	65564	PROWARE	2400	AtariST-N

----- GODORF/WESSELING --

02236	45892	HCS	2400	Art-Net
02236	47010	DARKLINE	2400	Art-Net
02236	63371	MAGIC MOUNTAIN	2400	

----- GREVENBROICH --

02181	41606	BEDY-BOX	14400	ProNet
02181	71649	DUCK TALES	2400	Power-Net

----- HAAN --

02129	4891	ANDI'S BBS	4800	Fido	2:244/10

----- HAGEN ---

02331	338606	TSHA	1200		
02331	339322	COMPOINT	2400		
02331	67555	WILDCAT HAGEN	2400	Fido	

----- HENNEF ---

02242	2764	KHC	2400	Art-Net	

----- HERDECKE ---

02330	13353	STONES HERDECKE	2400	Fido	

----- HERNE ---

02323	18757	UPTODATE	2400		
02323	33849	LIFE FAMILY	1200		
02323	43363	MONOPOL	1200		
02323	55158	E.DE.KA.	2400	Fido	

----- HILDEN ---

02103	8692	VOGON'S BBS	2400	Fido	2:244/15

----- HRTH ---

02233	15569	HDC Port 1	14400	MagicNET	
02233	33053	DOCKERS	1200		
02233	43747	HDC Port 2	14400	MagicNET	
02233	46488	HACKER	2400	Art-Net	
02233	77303	HPB	2400	MCNET	18-6 Uhr
02233	77863	TIME SCAN	2400	MCNET	22-1 Uhr

----- JLICH ---

02461	56291	Piranha	9600	MagicNET	
02463	3253	Shadow	2400	MagicNET	

----- KEMPEN ---

02152	6354	ISH	14400	UUCP	19.00-7.
02152	7030	ISH	2400		

----- KERPEN ---

02237	8733	WALL-STR	2400	PC-Net	
02237	3044	DARKNESS	2400	MC/SUB-Ne	
02237	54103	Broadway	9600	PC-Net	

----- KOELN ---

02203	12400	MAUS Kln 2	2400	Maus	
02203	2448	Data	9600	PC-Net	
02203	28501	Detour			
02203	26860	DATASERV PORZ	2400		
02203	33021	Pal	16800	PC-Net	
02203	63720	1Message	14400	MagicNET	
02203	69073	CREAM 0 PORZ	2400		
02203	83108	MEGAMAIL	2400	MagicNET	MEGAMAIL
0221	137603	BSB	2400	PC-Net	
0221	1702328	EXIL	2400		
0221	172717	VERTICO	2400	MCNET	
0221	176609	ASD	2400		
0221	210515	WDR-KOMM PORT 1	300		
0221	210516	WDR-KOMM PORT 2	300		
0221	210517	WDR-KOMM PORT 3	2400		
0221	216129	BETA-BOX	2400	MCNET	
0221	242748	Logomedia	2400	PC-Net	
0221	248066	Promt	2400	PC-Net	
0221	248285	SPACER	2400	Zerberus	SPACER
0221	249508	PRONET 1		ProNET	
0221	321035	ZOPI-BOX	2400	Art-Net	
0221	343390	US-POWER	2400	Art-Net	
0221	3601908	MCP	14400	Art-Net	
0221	3605415	BASIC	1200		
0221	361634	Senil	14400	MagicNET	
0221	361997	MARLBORO	2400	PC-Net	
0221	426760	Sunil	2400	PC-Net	
0221	441801	DARIUS	2400		
0221	488084	Matrix_2	2400	PC-Net	
0221	488085	Jinxed	9600	PC-Net	
0221	493176/77	MBK 1	2400		PASSWORT
0221	512640	HBK	14400		
0221	519755	SMART	14400	Art-Net	
0221	526849	MACRO	2400	MCNET	
0221	526862	HBK CHAT 1	9600		
0221	554219	TUBORG COLOGNE	2400	Zerberus	TBC
0221	558336	BITDSCHUNGEL	2400	Links-Zer	LINK-K
0221	5906556	FOTO	14400	Art-Net	
0221	5907407	HGS BBS	14400	Fido	
0221	592837	NIGHTLIFE	2400	Art-Net	
0221	601799	Syntron	16800	PC-Net	
0221	603027	NOFI	2400		
0221	616678	MICRO	2400		
0221	621278	Linex	9600	Art-Net	
0221	629996	PICCADILLY	2400	APNET	
0221	6320244	SONIC	2400	AmNet	
0221	639632	Digger	14400	MagicNET	
0221	668362	Guinness	16800	PC-Net	
0221	6801253	HAPPY	2400	Art-Net	
0221	6804471	RCL	9600	PC-Net	
0221	693767	GENIUS INFOSYST	9600		
0221	728201	TITANIC	2400		
0221	735664	DINO-BOX	2400		
0221	738904	Teucoms Terminal	9600	MagicNET	
0221	7604312	STILLE WASSER	2400	Fido	244/18.4
0221	7604312	C.O.N.	2400	Kultnet	
0221	82752731	Einstein	2400	PC-Net	
0221	852554	MEDBOX	2400	PC-Net	
0221	856888	KIBO	2400	Art-Net	
0221	8703864	MAUS KOELN-KALK	2400	Maus	K
0221	882898	Milka	16800	PC-Net	
0221	8903821	CTBBS	2400	Fido	
0221	892853	DEEPEST DUNGEON	2400		

----- KNIGSWINTER ---

02223	23225	EC-MAIL	2400	Fido	
02244	5552	Telebox	2400	TechnoNet	PC-Net
02244	7751	MAUS KOENIGSWIN	2400	Maus	SU
02244	81901	Telebox 2		PC-Net	

----- KREFELD ---

02151	304865	TREFFPUNKT	2400	Zerberus	TREFF
02151	307007	Maus Krefeld	2400	Maus	

```
02151  313124      KREFCOM            2400  UUCP        15-23 Uhr
02151  399895      MIRAMAR            2400
02151  474239      KWCK1              2400  PC-Net
02151  503433      WALBOX             2400              19-7 Uhr

----- KRTEN -----------------------------------------------------
02268  6715        FAC                2400  Art-Net

----- LEICHLINGEN -----------------------------------------------
02174  3672        TUBORG LEICHLIN    2400  Zerberus    TBL

----- LEMGO/SANKT AUGUST/SPICH ----------------------------------
02241  313453      AMS                2400  Art-Net
02241 404980       MESSAGE Port 1    16800  Zerberus ProNET GenieNET
                                            PhoenixNET
02241 405960       MESSAGE Port 2    16800  MagicNET KlnNET PcNET
02241  336014      PEANUT             2400  TechnoNet
02241  64425       CALYPSO            2400  TechnoNet  19-8 Uhr

----- LEVERKUSEN ------------------------------------------------
0214   45195       Twilight          14400  MagicNET
0214   94738       LOS               14400  MagicNET

----- LNEN ------------------------------------------------------
02306  12616       LEVIATHAN          2400
02306  44974       MOON44            16800

----- MEERBUSCH -------------------------------------------------
02159  50643       MISS               2400

----- METTMANN --------------------------------------------------
02104  16185       Crazy Zoo         14400  MagicNET
02104  73016       MEGABASE           2400

----- MOERS -----------------------------------------------------
02841  72868       Falke             14400  MagicNET

----- MNCHENGLADBACH/KORSCHENBROICH -----------------------------
02161  22501       ZIPPI              2400  Art-Net FIDO
02161  391169      PD-KISTE           2400  Art-Net
02161  648874      Atlantis          16800  ZENtron
02166  44713       SHARE             14400  Art-Net
02166  48521       MAB               14400
02166  80254       PD-Access         14400  ZENtron

----- MONHEIM ---------------------------------------------------
02173  31748       CHAOS-BOX          1200
02173  67331       WOA               14400  Art-Net

----- MLHEIM AN DER RUHR ----------------------------------------
0208   32554       FERGIE BOX         2400
0208   400327      HOT-LINE           2400  Zerberus    HOT
0208   470399      Tec-Box           14400  MagicNET
0208   479309      Alfa               9600
0208   592082      Softpro 2          9600  MagicNET
0208   762546      Omega-Box         14400

----- NEUSS -----------------------------------------------------
02131  222071      KGB        v32      9600  NCCNET
02131  21029       MAD        v32bis  14400  NCCNET
02131  27035       FBI   PEP  19K2    19200  NCCNET
02131  278074      FBI        v32      9600  NCCNET
02131  274332      BND                 2400  NCCNET
02131  25841       BND                 2400  NCCNET
02131  274621      Spektrum BBS        2400  NCCNET
02131  274623      Spektrum   v32bis  14400  NCCNET
02131  274624      Spektrum   v32bis  14400  NCCNET
02107  12207       ONLINE              2400

----- OBERHAUSEN ------------------------------------------------
0208   630466      OLDIEBOX            2400  Art-Net
0208   852252      Brabox

----- PULHEIM ---------------------------------------------------
02238  14661       ALIBABA             9600  GTN
02238  50800       CHROME              2400
02238  54702       SUNSHINE            2400  APNET
02238  55616       GOBLIN-BOX          2400  MCNET

----- RATINGEN --------------------------------------------------
02054  2345        R.C.B.              2400  Fido          2:244/3
02102  24670       B-Bogen             9600  MagicNET
02102  471453      Smartbox           16800  MagicNET
02102  68004       R.T.C.              4800  Fido          2:244/2
02102  68273       DESTROY             9600  Fido          2:507/13

----- RECKLINGHAUSEN --------------------------------------------
02361  185613      Atomic              2400
02361  186370      BBOX               16800
02361  492314      Moskito             2400  MagicNET

----- REMSCHEID -------------------------------------------------
02191  27627       PEN                 1200  AmNet 2
02191  291796      Edi-1               9600  PC-Net
02191  294924      Medicine            9600  PC-Net
02191  52688       RODS 2              9600  Fido
02191  54914       RODS 1              2400  PC-Net
02191  65496       SHADES
02191  74180       SUNRISE             2400  Art-Net
02191  75101       ABACUS              2400  Fido

----- SANKT AUGUST/SPICH/LEMGO ----------------------------------
02241  313453      AMS                 2400  Art-Net
02241 404980       Message Port 1     16800  Zerberus ProNET GenieNET
                                             PhoenixNET
02241 405960       Message Port 2     16800  MagicNET KlnNET PC-Net
02241  336014      PEANUT              2400  TechnoNet
02241  64425       CALYPSO             2400  TechnoNet  19-8 Uhr

----- SCHWERTE --------------------------------------------------
02304  18651       GALACTICA           2400  MSA-Net
02304  18799       SUNSHINE            2400
02304  187999      TRITON             14400  Zerberus    TRITON
```

Protokoll einer Mailboxsession
Seiten 11/12

```
----- SOLINGEN ---------------------------------------------
0212   45711       SIB              2400  AmigaNet
0212   652437      SBL              2400  PC-Net

----- SPROCKHOEVEL -----------------------------------------
02324  74896       PEDDER           1200

----- UEDEM ------------------------------------------------
02825  10173       UDS             14400

----- UNNA -------------------------------------------------
02303  61584       ORACLE           2400  PC-Net

----- VELBERT ----------------------------------------------
02051  63663       NETCOMM          2400  Art-Net
02053  48556       VOPATEPATU       2400  Zerberus   VOP

----- VIERSEN ----------------------------------------------
02162  21068       VCB              9600  PC-Net
02162  58352       Aircraft         9600  PC-Net

----- WEGBERG ----------------------------------------------
02434  25063       SBMSYS                 PC-Net

----- WEILERSWIST ------------------------------------------
02251  71479       Pegasus          9600  PC-Net
02251  74745       Titan                  PC-Net

----- WERMELSKIRCHEN ---------------------------------------
02196  4602        GRAF             2400  Fido

----- WESEL ------------------------------------------------
0281   7931        Science                MagicNET

----- WETTER -----------------------------------------------
02335  66001       CYGNUS WETTER    9600  Fido
02335  66282       NOVA             2400  ProNET

----- WITTEN -----------------------------------------------
02302  63783       PARADISE         1200             16-5 Uhr
02302  690011      WITTENER ROSEN   2400  Fido

----- WUPPERTAL --------------------------------------------
0202   305803      THE 49ER'S      14400  Fido
0202   308278      Byte             2400  PC-Net
0202   449515      NRW-BOX          2400
0202   462114      UNNET PORT 2     2400  AmNet 2
0202   463678      RONSDORFER MAIL  2400  Zerberus   RMB
0202   464259      Wods             2400  PC-Net
0202   4660234     UNNET PORT 1     2400  AmNet 2
0202   473086      Toelleturm       9600  Zerberus   TTB
0202   477483      EXPERT           2400  PC-Net     Expert
0202   477638      TTB II           9600  Fido
0202   500198      XENOX            2400  UUCP
0202   528544      HEIBOX           2400  PC-Net     Heibox
0202   595293      EXCALIBUR        2400
0202   595908      WDB II           2400  AmigaNet
0202   597645      WDB I            2400  AmigaNet
0202   605434      METAL            2400  PC-Net
0202   734751      Heavy            9600  PC-Net
0202   783901      WUPPER           2400  Art-Net
0202   80951       WUPPERT. BBS     2400  Fido

----- WLFRATH ----------------------------------------------
02058  72169       MAC              2400
02058  73228       AMIGA-BOX        2400  Space-Net

----- WRSELEN ----------------------------------------------
02405  18067       VOBIS Mailbox    2400

------------------------------------------------------------
02043  57821       Highway          2400  12:30-2:30
02043  39876       Abyss            2400
------------------------------------------------------------
02056  23915       Bytehoven       14400
------------------------------------------------------------
02150  6361        MIB              2400
------------------------------------------------------------
02192  6131        P.O.S.T          2400  Fido
------------------------------------------------------------
02291  5036        BABEBOX          2400  MCNET      20-9 Uhr
------------------------------------------------------------
02307  21968       THE WIZZARDS IN  2400
------------------------------------------------------------
02309  76577       AMIGOS           2400
------------------------------------------------------------
02325  49550       XANADU           2400             18-8.00
02325  52544       CTP-BOX          2400  ProNET
------------------------------------------------------------
02327  320077      CONDOR          14400  Fido
02327  74426       SOFTWORLD        2400  Fido
------------------------------------------------------------
02368  1428        Dark Empire     14400  MagicNET ProNET

All rights reserved. Unauthorized copying, reproducing, hiring,
lending, public performance and broadcasting prohibited.
 ... :-))))))
End of Message
Befehl:
DF
Dfue
0 = Dfue Allgemein
1 = Dfue Mailboxen
2 = Dfue Datex P
3 = Dfue BTX

X = Exit
Bitte waehlen:
2

   < 1 > = lesen     / read   / ler
   < 4 > = string suchen / search / procurar
   < 5 > = Inhaltsverzeichnis anzeigen
   < X > = Exit
```

Bitte waehlen:

MSG Nr. 16 von MAGICIAN vom 09.10.1992 Betr. GEnie
Hi,
wer sich mal in die zweitgroesste Mailbox der Welt eintragen will, hier die Nummern von GEnie.
- Datex-P-PAD anwaehlen (Duesseldorf - 2400 Baud: 134533 , Ortstarif)
- Nach der Connect-Meldung einen Punkt eingeben und <Return> druecken
- Jetzt muesste der Schriftzug DATEX-P und einige Zahlen erscheinen
- R 45221040880 eingeben (Verbindung zu GEnie, Anruf uebernimmt der Angerufene)
- Beim Login XTX99628,GEnie als U# eingeben
- Jetzt befindet man sich im Programm, um ein GEnie-Kennwort zu beantragen und muss einige Fragen beantworten. Die Kosten betragen fuer die Nutzung von GEnie betragen im Normaltarif 27$ und im Billigtarif $18. Bezahlung erfolgt nur mit Kreditkarte (Mastercard/Eurocard, Visa, American Express)

In GEnie gibt es ueber hundert Roundtables zu den unterschiedlichsten Themen bereichen (Computer, Hobby, Unterhaltung, Musik, Religion, Spaceport, SciFi etc.). Weiterhin steht ein Online-Lexikon (Grolier's Encyclopedia) zur Verfuegung. Software zum Downloaden ist natuerlich auch reichlich vorhanden.
Man kann sich auf ueber 40 Lines mit anderen Usern unterhalten. Dann gibt es noch jede Menge Online-Spiele: Schach, Reversi, Backgammon, Black Jack, Poker, A-MAZE-ING, Air Warrior und jede Menge Fantasy-Rollenspiele). Es finden oefters sogenannte Entwicklerkonferenzen statt, d.h. offizielle Mitarbeiter der grossen Computefirmen beantworten Fragen von Usern waerend des Chats.

Tschau,
MAGICIAN

MSG Nr. 10 von POSEIDON vom 20.03.1992 Betr. Datex
¦ GERMAN TEXT ONLY ¦ GERMAN TEXT ONLY ¦ GERMAN TEXT ONLY ¦

```
+-------------------------------------------------------------+
¦               AllES ÜBER BTX UND DATEX-P v1.0               ¦
¦               ===============================               ¦
¦released by the Phantom in 1992, as free public domestos edition¦
¦                                                             ¦
¦ring now Breaking The Wall at +49 - (0)30 8544619 to contact me!¦
+-------------------------------------------------------------+
¦ greetings to all contacts and friends I have met on my way  ¦
¦ until now                                                   ¦
+-------------------------------------------------------------+
```

Inhalt

Einleitung

1. Definitionen
   - Allgemein
     - Btx
     - Datex-P

2.1. Über Btx
   - Geschichte von Btx
   - Videotex-Standard
   - Datenschutz und Datensicherheit
     - Kennwvrter
   - Anbietervergütungen/Entgelte

2.2. Über Datex-P
   - Wichtige PAD-Befehle
     - Parameterbedeutung

3.1. Zugang zu Btx
   - Vffentliche Btx-Gerdte
   - Gastzugang

3.2. Zugang zu Datex-P
   - Herstellen der Verbindung zur PAD-Einrichtung
     - Eingeben der Teilnehmerkennung
     - Herstellen einer Datex-P-Verbindung
     - Besonderheiten bei Auslandsverbindungen
   - Auslvsung einer Verbindung
   - Dndern der Teilnehmerkennung Teil B
   - Datex-P-Prüfmvglichkeit "Echo"

4.1. Funktionsweise von Btx
   - Aufbau

4.2. Funktionsweise von Datex-P
   - Aufbau

4.5. Datenschutz
- Der Btx-Hack
- Der Nasa-Hack

5. Anhang
- Telefonnummern für den Zugang zu Btx (Tabelle)
- Telefonnummern für den Zugang zu Datex-P (Tabelle)
- Datex-P-Auslandsverkehrsbeziehungen (Tabelle)

Schlußbemerkungen

Einleitung

Dieser Text wurde notwendig, um endlich einmal (und vor allen Dingen vollständig, so hoffe ich) über diese beiden Datennetze zu berichten. Er soll erldutern, erkldren und Fragen beseitigen dort wo sie auftreten; Zusammenhdnge verdeutlichen und vereinfachen. Viele werden vielleicht sagen:
>viel zu weit ausgeholt, kannte ich doch alles schon!< oder dhnliche Bemerkungen. Aber denen sei gesagt, daß man bei diesem Thema einfach nicht weit genug ausholen kann, um solch ein komplexes System (vor allen Dingen denen, die auf diesem Gebiet nicht so bewandert sind) verstdndlich darstellen zukvnnen.
Ich hoffe, daß mir wenigsten dies gelungen ist, denn darauf kam es mir hauptsdchlich an!
Zu dem Inhalt ist noch folgendes anzumerken: BTX- und NASA-Hack Informationen wurden, wenn auch durch mich ziemlich verkürzt, dem Chaos Computer Buch entnommen. Ein entsprechender Vermerk befindet sich am Ende der Artikel.
In den Definitionen finden sich alle dem Laien auf Anhieb aber vielleicht auch dem Profi schwerverstdndliche Begriffe sowie allgemeingebrduchliche Abkürzungen genau erldutert wieder.
PS: Es war nicht gerade einfach einen solchen Text zu erstellen.
Nun aber viel Spaß beim Lesen...

1. Definitionen

ALLGEMEIN

Akustikkoppler:
Vorläufer des --> Modem. Nachteile: langsam, hohe Fehlerrate

BTX:
Bildschirmtext
Digitales Daten- bzw. Informationssystem in der Bundesrepublik Deutschschland; Betreiber: Deutsche Bundespost TELEKOM

CCC:
Chaos Computer Club
Bekanntester Computerclub in Deutschland, gegründet 1984, Sitz Hamburg. Erregte u. a. Aufsehen in der ganzen Welt mit dem Btx und NASA-Hack.

DEC:
Digital Equipment Corporation
Hersteller der --> VAX und des --> VMS.

Hacker:
Bezeichnung für einen Computerfreak, der sich mit Hilfe seines Heim- oder Personalcomputers über Datenfernverbindungen (z. T. widerrechtlich) Zugang zu Datenbanken zu verschaffen sucht (Meyers Enzyklopddie).

HEPNET:
High Energy Physics Network
Datennetzwerk für die weltweite Forschung auf dem Gebiet der Physik; bietet auch Verbindung zum --> SPAN.

Login:
Login nennt man den Vorgang, wenn man sich zu einen Rechner durch benutzen eines Schutzmechanismus (z. B. Paßwort) Zugang verschafft.

Logoff:
Siehe --> Logout.

Logout:
Logout nennt man den Vorgang, wenn man einen Rechner, zu dem man sich zuvor Zugang durch benutzen eines Schutzmechanismus verschafft hatte, wieder verläßt.

Modem:
MOdulator-DEModulator (Verschlüssler-Entschlüssler)
Dient der Datenfernübertragung (DFÜ), stellt Verbindung zwischen zwei oder mehr Rechnern auch über weite Entfernungen via Telefonleitung her.

NASA:
National Aeronautic Space Administration
Staatliche Raumfahrtbehvrde der Vereinigten Staaten von Amerika.

Netzwerk:
Verbund mehrerer Rechner zu einem Netz, um z. B. eine Steigerung der Rechenleistung zu erreichen oder einfach aus funktionellen Gründen (Vernetzung von Rechnern in verschiedenen Etagen eines Hauses).

Offline:
Ein Rechner ist >Offline<, wenn er nicht --> Online ist.

Online:

Zustand eines Rechners, der zur Zeit gerade mit einem anderen Rechner (meist durch Telefonleitung) verbunden und somit >Online< ist. Begriff findet auch Verwendung für Rechner, die an Datennetze gekoppelt sind (meist per Telefon) und Daten zum Abruf bereithalten; sie sind somit >Online<.

SPAN:
Space Physics Analysis Network
Datennetzwerk aufgebaut von der --> NASA verbunden mit dem europäischen EURO-SPAN betrieben von der European Space Agency (ESA) zum Datenaustausch auf dem Gebiet der Raumfahrt; bietet auch Verbindung zum --> HEPNET.

Trojanisches Pferd:
Nach der griechischen Sage: Ein Holzpferd, das als Geschenk der Griechen an die Osmanen nach Troja (Stadt in der heutigen Türkei) gebracht wurde.
In seinem Inneren befanden sich Krieger, welche schließlich in der Nacht durch den Überraschungseffekt die Kontrolle von Troja und dadurch den Sieg über die Osmanen erreichten.
Hier: Computerprogramm, welches in einem fremden Rechner durch Aktivierung vvllige oder zumindest teilweise Kontrolle über jenen erreicht.

VAX:
Großrechner von --> DEC mit Betriebssystem --> VMS.

VMS:
Betriebssystem einer --> VAX, hergestellt von --> DEC.

VTX:
Videotext
Über normale Fernsehkandle ausgestrahltes hauptsdchlich Nachrichten- und Informationsangebot, das nur mit Hilfe eines entsprechenden Decoders zu entschlüsseln ist. Mit diesem System ist kein Datenaustausch mvglich !!!

BTX

Anbieter:
Jeder der in Btx Informationen zur Verfügung stellt. Jeder Anbieter erhdlt von der Deutschen Bundespost TELEKOM eine Leitseite, unter der sein Angebot abrufbar ist.

Anbieterverzeichnis:
Ihnen steht ein alphabetisches Verzeichnis aller --> Anbieter in Btx zur Verfügung.

Anschlußkennung:
12stellige Kennung, die zur Feststellung der --> Zugangsberechtigung dient.

Antwortseiten:
Seiten, die von den --> Anbietern innerhalb ihres Programms für Ihre Anfrage oder Bestellung bereitgestellt werden.

Bereichskennzahl (BKZ):
Kennzeichnung eines --> Regionalbereiches für bereichsspezifische Angebote. Ein Regionalbereich entspricht im allgemeinen dem Regierungsbezirk.

Btx-Anschlußbox:
Wird von der Deutschen Bundespost TELEKOM als besonders komfortabler "Modem" installiert. Übernimmt beim Zugang automatisch die Anwahl der Btx-Vermittlungsstelle und übermittelt die --> Anschlußkennung.

Btx-Gastzugang:
Ermvglicht Interessenten, den Btx-Dienst, auch ohne eigene --> Zugangsberechtigung, kennenzulernen.
Bei Anwahl mit Modem oder Akkustikkoppler wird die --> Anschlußkennung mit 000 829 017 874 vorbesetzt. Wenn Sie diese Kennung mit # überspringen, kommt der Gastzugang zustande.

Btx-Nummer:
Ihre "Adresse in Btx", wenn Mitteilungen an Sie gesendet werden.
Die Btx-Nummer besteht aus 2 Teilen:
- Teilnehmer-Nummer -- sie ist identisch mit Ihrer Telefonnummer inkl. Vorwahl, aber ohne Nummer der Nebenstelle.
- Mitbenutzerzusatz -- zur Unterscheidung von Teilnehmern und --> Mitbenutzern.

Btx-Seite:
Alle Informationen in Btx werden in Form von Btx-Seiten dargestellt. Jede Btx-Seite ist mit einer Seitennummer gekennzeichnet und umfaßt 20 oder 24 Zeilen mit je 40 Zeichen.

Elektronischer Briefkasten:
Jeder Teilnehmer (und Mitbenutzer) hat seinen persvnlichen Briefkasten in Btx. Eintreffende Mitteilungen dort abgelegt und kvnnen nur von ihm abgerufen werden.

Externer Rechner:
Alle privaten Rechner (= Computer), auf die über die Btx-Vermittlungstelle zugegriffen werden kann.

Geschlossene Benutzergruppe (GBG):
--> Anbieter kvnnen die Mvglichkeit, Ihr Programm oder Teile daraus abzurufen, auf einen bestimmten Kreis von Teilnehmern - die Geschlossene Benutzergruppe - beschrdnken. Der Anbieter bestimmt, wer Mitglied in seiner GBG wird.

Kennwvrter:
Schutz bestimmter Nutzungsbereiche in Btx.

Kurzwahl:
Bis zu 10 Btx-Seitennummern mit einem kurzen Kommentar kvnnen gespeichert und durch Eingabe einer Auswahlziffer aufgerufen werden.

Meldung:
In der letzten Zeile einer Seite wird mitgeteilt, was gerade passiert oder welcher Schritt als ndchster folgen soll. Die Meldungen weisen auch aufStvrungen und Bedienungsfehler hin.

Mitbenutzer:
Neben dem Teilnehmer kvnnen weitere Personen (Mitbenutzer) Btx-Einrichtung en nutzen. Die Mitbenutzer werden vom Teilnehmer eingerichtet und verwaltet. Durch einen Mitbenutzerzusatz (0002 - 9999) werden die Mitbenutzer unterschieden.

Mitteilungsdienst:
Jeder Teilnehmer/Mitbenutzer kann jedem anderen Teilnehmer/Mitbenutzer (oder Anbieter) Mitteilungen zusenden oder von anderen Teilnehmern/Mitbenutzern Mitteilungen empfangen.

Mitteilungsseiten:
Spezielle Seiten zur Nutzung des Mitteilungsdienstes. Sie enthalten Eingabefelder für die --> Btx-Nummer des Empfdngers (als Adresse) und für die eigentliche Mitteilung. Es gibt Mitteilungsseiten für Text, --> transparente Daten und grafisch gestaltete Seiten für verschiedene Anldsse.

Regionalbereich:
Der --> Anbieter entscheidet, ob sein Programm bundesweit oder in einem bestimmten Regionalbereich abrufbar sein soll. Es gibt 32 Regionalbereiche, sie entsprechen weitestgehend den Regierungsbezirken.

Schlagwvrter:
Suchhilfe in Btx, wenn Btx-Angebote zu einem bestimmten Thema gesucht werden.

Transparente Daten:
Bezeichnung für Daten mit beliebiger bindrer Struktur, die in Btx übermittelt werden kvnnen.

Übergabeseite:
"Pforte" zu einem externen Rechner. Nach dem Absenden der Übergangsseite wird die Verbindung zum --> externen Rechner aufgebaut.

Übergdnge zu anderen Diensten:
Mit den Übergdngen in Btx kvnnen Sie andere Dienste der Deutschen Bundespost TELEKOM nutzen, ohne selbst Teilnehmer an diesen Diensten zu sein.

Vergütungen:
Sie werden von --> Anbietern für den Abruf ihrer Angebote verlangt. Es gibt Vergütungen je abgerufene Seite (max. 9,99 DM) und Vergütungen für die Nutzungsdauer eines Angebotes. Vergütungen werden von der Deutschen Bundespost TELEKOM mit der Fernmelderechnung dem Teilnehmer in Rechnung gestellt und dem --> Anbieter gutgeschrieben.

Zugangsberechtigung:
Zur Feststellung der Zugangsberechtigung dient die 12stellige --> Anschlußkennung (Soft- oder Hardwarekennung), die unter der Anschlußnummer im Btx-Dienst gespeichert ist.

DATEX-P

Datenendgerdt:
Rechner, mit dem man Daten empfdngt, sichtet und bearbeitet.

Gegenstelle:
Rechner, mit dem man nach Anwahl durch eine Datenleitung verbunden ist.

NUA:
Network User Adress (Netzwerk Benutzer Adresse, 11-15stellige Zahl) Adresse eines Rechners zum Anwdhlen innerhalb des Datensystems.

NUI:
Network User Information (Netzwerk Benutzer Informationen) Besteht aus Benutzerkennung + Passwort; Zugangsberechtigung zum System.

2.1. Über Btx

BTX ist ein digitales Dateninformationsdienstsystem der Deutschen Bundespost (TELEKOM). Es ermvglicht den Zugriff auf unendlich viele Informationen. Diese kann man grundsdtzlich in zwei Arten unterteilen:

1. Daten/Informationsabruf
   (Nachrichten: politisches Weltgeschehen, Bvrse, Wetter etc.; sonstige Informationen: Auskünfte, Angebote, Werbung etc.)

2. Daten/Informationsaustausch
   (Dialoge führen: Überweisungen ausführen, Bestellungen tätigen etc.;
   Telesoftware: Datentransfer UL/DL, Onlinespiele benutzen etc.)

Geschichte von Btx:

Die Btx-Philosophie basiert auf der einfachen Grundüberlegung, daß nahezu jeder Haushalt in der Bundesrepublik Deutschland über einen Fernseher und ein Telefon verfügt. Damals Lieblingskind von Bundespostminister Christian Schwarz-Schilling (auch unter >Blackpenny< bekannt) wurden Btx bis Mitte der achtziger Jahre 1 Million Teilnehmer prognostiziert. Bildschirmtext sollte damals das grvßte Computersystem der Welt werden, mit der Entwicklung wurde daher I.B.M beauftragt. Nach ersten Feldversuchen in Berlin und Düsseldorf wollte die Bundespost auf der Funkausstellung 1983 den bundesweiten Start von Btx im europdischen CEPT-Standard persvnlich einläuten.
Nur I.B.M war noch nicht fertig, und so behalf man sich zundchst mit einer Übergangslvsung. Als der I.B.M-Zentralrechner in Ulm mit

neun Monaten Verspdtung dann endlich in Betrieb ging, bekam er von Btx-Kennern sofort den Namen >Yo Yo< - denn wie beim Kinderspiel ging es mit ihm auf und ab, bis zum ndchsten Absturz.
Darüberhinaus hatte Btx nur wenig zu bieten: Die Programmangebote waren meist unübersichtlich und umstdndlich zu handhaben, die Blockgrafikbildchen waren grob und ungelenk. Nur mdßig stiegen die Teilnehmerzahlen, statt der erwarteten 150.000 waren es Ende 1984 gerade erst 19.000 Nutzer.

Videotex-Standard:

Bildschirmtext entspricht in seinem Darstellungsumfang dem "erweiterten harmonisierten Videotexdienst", den die Konferenz der europdischen Verwaltungen für Post und Fernmeldewesen (CEPT) in der Empfehlung T/CD-06-01 festgelegt hat. Neben diesem Basisstandard (Alphamosaik, Farben, DRCS) beschreibt die Empfehlung zusdtzliche Verfahren wie Geometrik, Telesoftware, Bildplattensteuerung und Photografik. Die Geometrik kann wie die anderen Ergdnzungen zusdtzlich zum Basisstandard in die Btx-Endgerdte implementiert werden und steht somit als Option für bestimmte Anwendungen wie Business-Grafik, Design-Grafik, Ingenieur- und Architekturzeichnungen usw. zur Verfügung. Im Juli 1985 hat die CEPT die Geometrik an die Funktionen des grafischen Kernsystems (GKS) angepaßt, so daß deren Codierungsschemata nunmehr zur Computergrafik kompatibel sind. Der Btx-Dienst der DeutschenBundespost wird seinen eingeführten Darstellungsumfang unverdndert beibehalten, so daß die heutigen Endgerdte langfristig nutzbar bleiben. Die im CEPT-Standard enthaltenen Optionen sind wdhlbare Ergdnzungen zum Grundstandard, die vergleichbar einem PKW-Anhdnger nur in besonderen Bedarfsfdllen genutzt werden.

Datenschutz und Datensicherheit:

Die wdhrend der Benutzung von Bildschirmtext übertragenen Daten werden durch den Btx-Dienst im allgemeinen nicht erfaßt. Ausgenommen davon sind die Daten, die zur Berechnung von Entgelte und Anbietervergütungen dienen. Dem Anbieter werden Ihre persvnlichen Daten (Name und Anschrift) nur mitgeteilt, wenn Sie die aufgekommenen Anbietervergütungen nicht fristgerecht entrichtet haben.
Ihre persvnlichen Daten werden dem Anbieter wdhrend einer laufenden Verbindung grundsdtzlich nicht bekanntgegeben. Spricht Sie ein Anbieter innerhalb seines Seitenangebotes persvnlich an, so wurden die persvnlichen Angaben vom Btx-Dienst automatisch eingesetzt.
Bei Dialog- und Mitteilungsseiten entscheiden Sie selbst, ob Sie die Seite absenden oder nicht. Nur wenn Sie die Seite absenden, werden Ihr Name, Ihre Anschrift und Ihre Btx-Nummer an den Empfdnger (den Anbieter oder einen vorbestimmten Teilnehmer) übermittelt. Dies gilt auch im Dialog mit Externen Rechnern. Enthdlt eine Dialogoder Mitteilungsseite persvnliche Daten, die vom Btx-Dienst automatisch eingesetzt wurden, so erkennen Sie dies an einem "P!" in der letzten Zeile auf Ihrem Bildschirm. Eine solche Seite wird erst übertragen, wenn sie von Ihnen mit "19" bestdtigt wird. Mit *05# kvnnen Sie sich zur Kontrolle die Datenfeldinhalte der Seite anzeigen lassen. Wenn Sie die Seite wieder vollstdndig sehen wollen, so geben Sie *00# ein.

Kennwvrter:

- Welche Kennwvrter gibt es?

Die unterschiedlichen Kennwvrter dienen dem Schutz Ihres Btx-Anschlusses!

- Das persvnliche Kennwort zum Zugang
- Das Nutzungskennwort für die Teilnehmer/Mitbenutzerverwaltung; Bereich *7#
- Das Kennwort für den Zugang zum Anbieterbereich
- Das Kennwort für Bulk-Updating über Magnetband

Solange keine gesonderten Kennwvrter eingerichtet sind, gilt für alle Bereiche das persvnliche Kennwort.

- Bildungsregeln:

Kennwvrter müssen mindestens 4 und dürfen hvchstens 8 Zeichen lang sein. Bei weniger als 8 Zeichen müssen Sie nach dem letzten Zeichen "#" eingeben. Als Zeichen kvnnen sowohl Ziffern als auch Groß- und Kleinbuchstaben (auch gemischt) verwendet werden. *,# und Leerzeichen sind nicht erlaubt! Kennwvrter mit Einfachstruktur (123456, 0815, 1333, abcd, AAABBB usw.) kvnnen nicht eingerichtet werden. Die letzten Stellen der Btx-Nr. als Kennwort sind ebenfalls nicht zugelassen.

- Fehleingaben/Anschlußsperre:

Wird das persvnliche Kennwort 3 mal hintereinander falsch eingegeben, erfolgt Verbindungsabbruch. Bei 9-maliger Fehleingabe an einem Tag wird derbenutzte Btx-Anschluß gesperrt. Falscheingaben Ihrer Mitbenutzer an diesemAnschluß werden mitgezdhlt.
Fehleingaben der anderen Kennwvrter führen nicht zur Anschlußsperre; jedoch wird die Verbindung zu Btx nach dreimaliger Falscheingabe getrennt. Wenn Sie eines Ihrer Kennwvrter vergessen haben oder Ihr Btx-Anschluß gesperrt wurde, wenden Sie sich bitte persvnlich oder schriftlich an die Anmeldestelle Ihres Fernmeldeamtes.

- Dndern Kennwvrter:

Ihre Kennwvrter sollten Sie zu Ihrem eigenen Schutz gelegentlich dndern und sicher aufbewahren. Das persvnliche Kennwort ist jederzeit über *72#, das Nutzungskennwort über *77# dnderbar.
Teilnehmer mit Anbieterberechtigung und Mitbenutzer mit Editierberechtigung kvnnen das Kennwort für den Anbieterbereich und ggf. für Bulk-Magnetband nach Aufruf von *910# dndern.

Anbietervergütungen/Entgelte:

Neben den festen Kosten für die Zugangsberechtigung fallen auch nutzungsabhdngige Kosten an. Es ist zwischen Entgelten und Anbietervergütungen zu unterscheiden. Alle Kosten für Btx werden mit der Telefonrechnung dem Btx- Teilnehmer in Rechnung gestellt. Nach Eingabe von *92# erhalten Sie eine Übersicht zu den Kosten der

bestehenden Verbindung, gegliedert nach Entgelte und Anbietervergütungen. Geben Sie auf dieser Übersicht # ein, so wird eine Auswahlseite Abrechnungsdaten angezeigt. Von dort erreichen Sie die Abrechnungsdaten für den laufenden und den vorhergehenden Rechnungsmonat.

- Entgelte:
Entgelte erhebt die Deutsche Bundespost TELEKOM für:
  - Telefonverbindungen zur Btx-Vermittlungsstelle
  - Zugriffe auf fremde Regionalbereiche
  - Senden/Speichern von Mitteilungen
  - Senden von Btx-Telex Mitteilungen
  - Speichern von Mitbenutzern
  - Nutzen des Anbieterbereiches

Der Teilnehmer kann seine Mitbenutzer mit einer "Entgeltsperre" (außer Telefon- und Editor-Entgelte) belegen.

- Anbietervergütungen seitenabhdngig:
Anbietervergütungen werden von den Anbietern für den Abruf Ihrer Angebote oder einzelner Seiten verlangt. Die Anbietervergütungen werden dem Teilnehmer von der Deutschen Bundespost TELEKOM mit der Telefonrechnung in Rechnung gestellt und den Anbietern gutgeschrieben. Die Hvhe der Anbietervergütungwird vom Anbieter festgelegt. Wenn Sie eine Seite aufrufen, die mit einer Anbietervergütung belegt ist, werden Sie vor der Anzeige in der letzten Zeile auf die Hvhe der Anbietervergütung hingewiesen. Erst nach Bestdtigung mit # wird die Seite angezeigt und Ihr Btx-Konto mit dem Vergütungsbetrag belastet.
Bei Mitteilungs-, Antwort- und Übergabeseiten wird Ihr Btx-Konto erst nach dem Absenden der Seite mit dem Vergütungsbetrag belastet; das Ausfüllen der Seite ist kostenlos, soweit das Angebot nicht mit einer nutzungszeitabhdngigen Anbietervergütung belegt ist. Nach dem Ausfüllen der Seite wird in der letzten Zeile auf die Anbietervergütung und ggf. auf das Entgelt hingewiesen. Abgesandt wird die Seite mit "19"! Teilnehmer kvnnen für Ihre Mitbenutzer eine Anbietervergütungssperre oder eine obere Vergütungsschwelle einrichten.

- Anbietervergütungen nutzungszeitabhdngig:
Mit einer nutzungszeitabhdngigen Anbietervergütung kann der Anbieter entweder sein gesamtes Angebot oder nur Übergabeseiten und nachfolgende Seiten im Externen Rechner belegen. Rufen Sie die erste Seite eines Angebotes mit nutzungszeitabhdngiger Anbietervergütung auf, wird mit einer Meldung in der letzten Zeile auf die Kosten XX;XXDM/Min hingewiesen. Werden in einem Angebot mit zeitabhdngiger Anbietervergütung Folgeseiten zusdtzlich mit einer Seitenvergütung belegt, muß die Seitenvergütung für jede Seite extra bestdtigt werden. Das Zeichen "*" in der Mitte der letzten Zeile signalisiert, daß der "Zeittakt" noch lduft bzw. nach einem Rücksprung (z.B. mit *#, *55#, *Seitennr.# oder *Anbietername#) aus einem zeitvergütungsfreien Angebot weiterlduft.
Teilnehmer kvnnen in der Mitbenutzerverwaltung den Rahmen, in dem ihre Mitbenutzer Anbietervergütungen verursachen, bestimmen. Sie haben dazu zwei Mvglichkeiten: die Vergütungssperre und die obere Vergütungsschwelle. Bei zeitabhdngiger Anbietervergütung werden keine Seiten angezeigt, deren Anbietervergütung/Min die Vergütungsschwelle übersteigt.

2.2. Über Datex-P

Das Datex-P-System ist ein sehr viel komplexeres System, als das nur in der Bundesrepublik existierende Btx-System. Es handelt sich hierbei also um ein internationales digitales Dateninformationsnetzsystem, welches mit oder durch viele andere, ihm dhnlicher Systeme, verbunden ist und somit Zugang in die ganze Welt verschafft.

Wichtige PAD-Befehle:

CLEAR	....	Auslvsen einer gewdhlten virtuellen Verbindung
PAR?	....	Abfrage der aktuellen PAD-Parameter
PROF?n	....	Anzeige des Profils n (n = 1-7, 90, 91)
PROF n	....	Auswahl des Profils n
SET 6:5	....	Setzen des Parameters 6 auf den Wert 5
STAT	....	Statusabfrage (>frei< oder >belegt<)

Folgende Tabelle wird interessant, wenn man im PAD >PAR?< + return (enter) eingibt. Anhand nachstehender Tabelle lassen sich die Parameter erkldren.

Parameterbedeutungen (Datex-P-Profil für Einwdhlzugdnge):

Parameter	Wert	Bedeutung
001	001	Austritt aus DFÜ-Phase mit CTRL + P möglich
002	001	PAD-Echo eingeschaltet
003	002	Datenweiterleitung mit return (enter)
004	000	Keine Zeitstufe für Weiterleitung
005	000	Keine Steuerung zusdtzlicher Eingabegeräte
006	001	PAD-Meldungen werden ausgegeben
007	021	BREAK unterbricht die Ausgabe
008	000	Normale Datenausgabe zum Endgerdt
009	002	Zwei Füllzeichen (NUL) werden nach return eingefügt
010	000	Keine Zeilenldnge-Begrenzung im PAD
011	Der Wert hdngt von der Übertragungsgeschwindigkeit ab.	
012	000	DC1/DC3 für Empfang ausgeschaltet
013(126)	004	PAD sendet LF (Line Feed) nach Empfang von return
014	000	Kein Füllzeichen (NUL) nach LF (Line Feed)
015	000	Kein Editieren wdhrend der Datenübertragungsphase
016(118)	127	Lvschen eines eingegebenen Zeichens im PAD mit Hilfe des Zeichens (DEL)
017(119)	024	Lvschen einer eingegebenen Zeile im PAD mit Hilfe des Zeichens (CAN)
018(120)	018	Wiedergabe einer eingegebenen Zeile im PAD mit Hilfe des Zeichens (DC2)
019	000	Keine Editier-PAD-Meldungen
020	000	Kein Zeichen vom ECHO ausgeschlossen
021(123)	001	Paritdtsbit-Prüfung im PAD eingeschaltet
022	000	Funktion >Seite abwarten< ausgeschaltet
121	000	Kein zusdtzliches Weiterleitungszeichen
125	000	Keine Verzvgerung von Ausgaben im PAD

3.1. Zugang zu Btx

Um Zugang zum Btx-System zu bekommen gibt es grundsdtzlich 2 Mvglichkeiten.

1. Btx-Anschluß der Deutschen Bundespost TELEKOM
2. Gastzugang

- durch der Vffentlichkeit zugdnglichen von der Deutschen Bundespost TELEKOM aufgestellten Btx-Gerdten (manche auch mit direktem Zugang!).
- per Telefon mit Home- bzw. Personalcomputer via Modem mit Hilfe eines Btx-Decoderprogrammes, das die ankommenden Signale in für den Computer verstdndlichen und damit darstellbaren Zeichen umwandelt.
  Bei Modems mit Hayes-Standard (AT-Befehlssatz) ist unbedingt zu beachten, daß für eine fehlerfreie Verbindung AT&M0 gesetzt ist.

Öffentliche Btx-Terminals:
Die Deutsche Bundespost TELEKOM stellt Btx-Terminals für den allgemeinen Gebrauch zur Verfügung, genannt vBtx. Diese vBtx-Gerdte stehen in Postdmtern und vffentlichen Gebduden. An diesen Gerdten kvnnen Sie ebenfalls Btx betreiben, aber keine vergütungspflichtigen Seiten abrufen. Auch sind entgeltpflichtige Seiten vom Abruf gesperrt. Der Mitteilungsversand ist ermvglicht.
Eine Aufstellung über die Standorte der vBtx-Gerdte finden Sie auf der Seite *10423#. Einige vBtx-Gerdte sind anschlußfreizügig geschaltet, so daß Sie mit Ihrer eigenen Telefon-Nummer btx'en kvnnen.

Gastzugang:
Interessierten wird mit dem Gastzugang Gelegenheit gegeben, Btx ohne eigene Zugangsberechtigung kennenzulernen. Voraussetzungen: Modem oder Akustikkoppler und btx-fdhiges Endgerdt. Keine Mvglichkeit mit der Btx-Anschlußbox. Das Feld für die Anschlußkennung ist auf der Btx-Einstiegsseite vorbesetzt. Wird diese Kennung mit # übersprungen, kommt der Gastzugang zustande. Dem "Gast" ist die Btx-Nummer 99999-1 zugeordnet; entgelt- und vergütungspflichtige Aktionen kvnnen nicht durchgeführt werden.

3.2. Zugang zu Datex-P

Herstellen der Verbindung zur PAD-Einrichtung:

- von einem Telefonanschluß
1. Einschalten des Datenendgerdtes.
2. Datenendgerdt auf Betriebsart >ONLINE< einstellen.
3. Zugang Datex-P anwdhlen (Rufnummernverzeichnis siehe Anhang) und Datenton abwarten
4a Bei Betrieb mit Modem Datentaste (linke Taste) am Telefon sofort drücken und den Telefonhvrer auflegen.
4b Bei Verwenden eines Akustikkopplers den Telefonhvrer sorgfdltig und fest in der vorgeschriebenen Lage in die Aufnahme einsetzen. Dafür stehen 22 Sekunden zur Verfügung.

Jetzt haben Sie ca. 90 Sekunden Zeit, das Dienstanforderungssignal sowie ggf. Ihre Teilnehmerkennung einzugeben und eine Datex-P-Verbindung herzustellen.
5. Dienstanforderungszeichen (Satzzeichen Punkt >.<) sofort eingeben (ggf. wiederholen).

Das erfolgreiche Herstellen der Verbindung zur PAD-Einrichtung wird angezeigt mit: DATEX-P: 44 XXXX XXXXX (Dabei bezeichnet 44 XXXX XXXXX die Datex-P-Rufnummer des erreichten Zugangs bzw. Ihres Datex-P20H-Anschlusses). In den nächsten 60 Sekunden müssen Sie eine Datex-P-Verbindung herstellen, ansonsten wird die Zugangsverbindung automatisch ausgelvst.

- von einem Datex-L300-Anschluß
Zugang Datex-P mit der Rufnummer 140 anwdhlen, sonst wie oben (ohne 4).

- von einem Datex-P20H-Anschluß
Datenendgerdt einschalten, Betriebsart >ONLINE< einstellen und Dienstanforderungssignal eingeben; die Zeitgrenzen gelten nicht.

Eingeben der Teilnehmerkennung:

Die Teilnehmerkennung dient dazu, den Bediener am Datenendgerdt zu identifizieren und mit den Datex-P-Verbindungsgebühren zu belasten. Beim Zugang von einem Telefon- oder Datex-L300-Anschluß ist sie für internationale Datex-P-Verbindungen immer erforderlich, bei nationalen Verbindungen kann sie entfallen, wenn der gerufene Datex-P-Anschluß die Gebühren übernimmt (Eingabe R oder REV beim Befehl zur Verbindungsanforderung). Beim Benutzen einer Teilnehmerkennung am Datex-P20H-Anschluß (= es ist auf Wunsch mvglich, einen Datex-P20H-Anschluß so zu programmieren, daß die Eingabe einer Teilnehmerkennung oder die Gebührenübernahme durch den gerufenen Anschluß erforderlich ist, Verbindungen zu Lasten des Anschlusses sind dann verhindert) wird nicht der Anschluß, sondern die Teilnehmerkennung mit den Verbindungsgebühren belastet. Nach dem Herstellen der Verbindung zur PAD-Einrichtung erfolgt die Eingabe des Teils A der Teilnehmerkennung: >NUI DAAAAA< + return (enter).

Das Netz antwortet mit: DATEX-P: Passwort
XXXXXX

Der Cursor fdhrt automatisch an die Stelle des ersten X zurück. Jetzt Teil B der Teilnehmerkennung in das X-Feld eingeben!: >BBBBBB< + return (enter).
Datex-P unterdrückt bei Duplex-Betrieb die Anzeige Ihrer Eingabe, bei Halbduplex-Betrieb wird die Eingabe mit >MMMMMM< überschrieben, darauf folgt die Netzmeldung: DATEX-P: Teilnehmerkennung DAAAAA aktiv (sofern Ihre Eingaben richtig waren). Die Netzmeldung: DATEX-P: Teilnehmerkennung Fehler, zeigt an, daß ein Fehler vorliegt: - Schreibfehler, - falsches Paßwort, - nicht gültige Teilnehmerkennung. Sie kvnnen erneut Ihre Teilnehmerkennung, Teile A und B, wie vorstehend beschrieben eingeben.

Herstellen einer Datex-P-Verbindung:

Nachdem die Verbindung zur PAD-Einrichtung besteht und ggf. eine Teilnehmerkennung aktiv ist kann eine Datenverbindung über Datex-P hergestellt werden.
Dazu ist der >Befehl zur Verbindungsanforderung< in einer Zeile einzugeben:

">REV <YY YYYY YYYYY>, Benutzerdaten<" + return (enter).

dabei bedeutet

REV	die Anforderung der Gebührenübernahme (Reverse Charging),
YYYYYYYYYYY	die Rufnummer des gerufenen Anschlusses,
Benutzerdaten	bis zu 12 alphanumerische Zeichen, die von der Rufnummer durch Komma zu trennen sind.

REV kann mit R abgekürzt werden.
Falls Sie bei der Eingabe einen Fehler gemacht haben oder wenn die Verbindung nicht mvglich ist, empfangen Sie eine entsprechende Meldung. Das erfolgreiche Herstellen der Verbindung wird angezeigt mit:

DATEX-P: Verbindung hergestellt mit YY...YY
(LKN) (i,n,Paket-Laenge: 128) (XXX)

dabei bedeutet

YY..YY	die Rufnummer des erreichten Anschlusses bei Auslandsverbindungen einschließlich Netzkennzahl,
LKN	die logische Kanalnummer bei der Gegenstelle,
i	Auslandsverbindung (entfdllt bei nationalen Verbindungen),
n	normale Übertragung,
Paket-Laenge	maximale Anzahl von Zeichen je Datenpaket,
XXX	Knoten und Leitungsnummern bei internationalen Verbindungen.

Jetzt sind Sie mit der gerufenen Gegenstelle verbunden und die Berechnung der Datex-P-Gebühren beginnt. Ihre weiteren Eingaben hdngen von der Gegenstelle (Host) ab; bitte fragen Sie bei Schwierigkeiten dort nach, hier kann Ihnen die Deutsche Bundespost TELEKOM leider nicht helfen.

Besonderheiten bei Auslandsverbindungen:

- Die wahlfreie Angabe REV ist nicht zuldssig.
- Beim Zugang von Telefon- oder Datex-L300-Anschlüssen muß eine gültige Teilnehmerkennung verwendet werden, auch wenn der gerufene Anschluß am besonderen Gebührenübernahmeverfahren teilnimmt.
- Der >Befehl zur Verbindungsanforderung< sieht bei internationalen Verbindungen wie folgt aus:
  O ZZZZ YY..YY >,Benutzerdaten< + return (enter).
  dabei bedeutet

O	die Zugangskennziffer, die dazu dient, einen internationalen von einem nationalen Verbindungswunsch zu unterscheiden,
ZZZZ	die internationale Netzkennzahl des Netzes, an dem die gerufene Endeinrichtung angeschlossen ist,
YY..YY	die nationale Datenrufnummer der gerufenen Endeinrichtung (die Länge hängt vom jeweiligen Netz ab),
Benutzerdaten	wurden oben schon erläutert.

In manchen Ldndern (z. B. Großbritannien) ist auch national die Netzkennzahl mitzuwdhlen, d. h. die nationale Rufnummer ist gleich der internationalen Rufnummer. In anderen Netzen (z. B. TELENET) weicht die nationale Datenrufnummer von der im internationalen Verkehr zu verwendenden ab. Bitte fragen Sie den Betreiber Ihrer Gegenstelle nach seiner >Internationalen Network User Adress (NUA)<.

Auslvsung einer Verbindung:

Normalerweise lvst Ihre Gegenstelle die Verbindung aus, wenn Sie die Sitzung ordnungsgemdß beenden. Dies zeigt Ihnen das Netz mit einer der folgenden Meldungen an:

DATEX-P: Ausloesung - Anforderung durch Gegenstelle
oder
DATEX-P: Ausloesung - Veranlassung durch Gegenstelle

dabei bedeutet - >Anforderung<, daß die Gegenstelle den PAD aufgefordert hat, die Verbindung nach Ausgabe aller Daten auszulvsen.
- >Veranlassung<, daß die Gegenstelle die Verbindung direkt ausgelvst hat; vom PAD noch nicht ausgegebene Daten gehen verloren.

Sie kvnnen die Datex-P-Verbindung auch selbst auslvsen mit CTRL & P >CLEAR< + return (enter) oder CTRL & P >CLR< + return (enter). (In Abhdngigkeit von der Parametereinstellung durch die Gegenstelle ist an Stelle von >CTRL + P< das Zeichen >BREAK< zu verwenden oder ein Austritt aus der Datenübertragungsphase ist nicht mvglich. Im letzten Fall kvnnen Sie die Datex-P-Verbindung nur über die Gegenstelle (z. B. mit LOGOFF) oder mit der >Notbremse< auslvsen. Diese Notbremse bedeutet, daß Sie die Zugangsverbindung durch Ausschalten der Betriebsart >ONLINE< bzw. Ihres Datenendgerdtes trennen.)

Das Netz bestdtigt mit:
DATEX-P: Ausloesung - lokale Veranlassung
und fordert mit
DATEX-P: 44 XXXX XXXXX
zu einem erneuten Verbindungsaufbau auf. Bei Bedarf ist jetzt das Lvschen Ihrer Teilnehmerkennung mit: NUI OFF + return (enter) mvglich. Sie empfangen daraufhin die Netzmeldung: DATEX-P: Teilnehmerkennung nicht aktiv. Nun sind Sie wieder am Anfangspunkt von Datex-P. Wenn Sie keine weitere Datex-P-Verbindung herstellen wollen, ist beim Benutzen eines Zugangs der Telefon- bzw. Datex-L-Anschluß durch Ausschalten der Betriebsart >ONLINE< freizugeben und ggf. der Telefonhvrer aufzulegen.

Hinweis: Beim Zugang von Telefonanschlüssen sind pro Einwahl nur vier Datex- P-Verbindungen bzw. Verbindungsversuche mvglich, d. h. nach der vierten Auslvsung in Datex-P wird automatisch die Telefonverbindung zur PAD-Einrichtung getrennt.

Dndern der Teilnehmerkennung Teil B:

Inhaber von Teilnehmerkennungen Datex-P kvnnen das Paßwort ihrer Teilnehmerkennung im Online-Dialog selbst dndern, wenn sie dazu eine Datex-P-Verbindung zum Teilnehmererkennungssystem aufbauen. Darüber hinaus sind noch folgende statistische Informationen abrufbar:

- Anzahl der Tage seit der letzten Paßwortdnderung
- Anzeige der letzten zehn Validierungen mit Datum und Uhrzeit (Validierung = Teilnehmerkennung aktivieren bzw. nutzen)
- Anzahl der Fehlvalidierungen seit der letzten Abfrage

Anleitung:
1. Verbindung zum PAD herstellen und die Teilnehmerkennung eingeben (siehe oben).
2. Datex-P-Rufnummer des Systems eingeben 45010019000 (oder 45020029000), daraufhin kommt die Meldung >DATEX-P: Verbindung hergestellt mit ...<
3. Eingangsdialog
   Das System stellt sich mit >DATEX-P-Teilnehmerkennungssystem der Deutschen Bundespost TELEKOM< vor und fordert nun den Benutzer auf, sich ebenfalls vorzustellen.
   \>> Bitte Teilnehmerkennung Teil A
   eingeben: DAAAAA
   \>> Bitte Teilnehmerkennung Teil B
   (Passwort) eingeben:
   HHHHHH
   ****** (Das Paßwort wird nicht abgedruckt.)
   Eingabe bitte: (? Gibt Ihnen eine Uebersicht ueber die moeglichen Befehle)
   \>>
   Jetzt kvnnen Sie folgende Befehle eingeben:
   1 --> Dndern des Paßwortes
   2 --> Informationen zur Teilnehmerkennung
   3 --> Ende des Dialogs mit dem System

```
   ? --> Hilfe
4. Dndern des Paßwortes (Teilnehmerkennung Teil B)
   Geben Sie dazu die Zahl 1 ein!
   >> 1
   Bitte beachten Sie bei der Eingabe, daß das Paßwort
   aus genau sechs Zeichen bestehen muß. Das Paßwort
   wird nicht abgedruckt.
   >> Geben Sie den neuen Teil B (Passwort)
      Ihrer Teilnehmerkennung ein:
      HHHHHH
      ******
   >> Bitte wiederholen:
      HHHHHH
      ******
   Teilnehmerkennung Teil B (Passwort) geaendert

5. Abfragen von Informationen zur Teilnehmerkennung
   >> 2
   Die Anzahl der Tage seit der letzten Passwort-
   aenderung 40
   Folgende Validierungsdaten fuer diese Teilnehmer-
   kennung
   17.01.90/07.30
   23.10.89/09.58
   23.10.89/08.10
   Anzahl der Fehlvalidierungen seit der letzten
   Abfrage 2
6. Ende des Dialogs
   >> 3
   DATEX-P: Ausloesung - Anforderung durch Gegenstelle
```

Hinweis: Punkt 6 der Anleitung ist nur notwendig, wenn lediglich Punkt 4 oder Punkt 5 durchgeführt wurden. Bei Durchführung der Punkte 4 und 5 bzw. doppelter Eingabe eines der beiden Punkte erfolgt danach die automatische Auslvsung der Verbindung durch das Teilnehmerkennungs system.

Datex-P-Prüfungsmvglichkeit "Echo":

Sie kvnnen die Betriebsfdhigkeit des Netzes für Verbindungen zu allen Datex-P-Vermittlungsstellen mit Hilfe der Echo-Funktionen prüfen. Bitte geben Sie als Befehl zur Verbindungsanforderung folgende Zeichen ein:
45 ONKZ 49002,ECHO + return (enter)
dabei ist ONKz die Telefon-Vorwahl ohne erste Nummer (immer 4stellig, ggf. rechts mit Null ausfüllen) der gewünschten Datex-P-Vermittlungsstelle. Zum Beispiel erreichen Sie das Echo in München mit 45 8900 49002,ECHO + return (enter). Verbindungsanforderungen mit REV (siehe oben) werden abgewiesen. Der Test beginnt, nachdem Sie die Meldung: >DATEX-P: Verbindung hergestellt mit...< empfangen haben. Geben Sie bis zu 128 beliebige Zeichen ein, die Sie mit return (enter) zum Echo absenden. Die Testdaten werden sofort zurückgesandt und Sie kvnnen eingegebene und empfangene Zeichen vergleichen, sie müssen übereinstimmen.
Verbindungsauslvsung mit CTRL + P CLEAR + return (enter), wie zuvor beschrieben.

Einige sicherlich interessante Beispiele zum Ausprobieren, sofern man Zugang zu Datex-P (oder sich eine NUI "ausgeborgt") hat:

Systemname:	NUA:	Bemerkungen:
Outdial	03110 203 00121	Format: ATDT+Nr. (Europavorwahl 011)
QSD-System	02080 570 40540	Internationales Chatsystem in F.Reich
Washington Post	03106 005 84401	Amerikanische Zeitung; PW: KRON
?	02342 123 00387	Chatsystem in Großbritannien; PW: C 20

4.1. Funktionsweise von Btx

Aufbau:

```
                      West-German City          Btx
Intruder -->              Telephone        -->  Digital
(Anrufer)                 Exchange              Network
                    (Vermittlungsstelle)        (TELEKOM)
                                                    |
                                                    |
                                                Extern Computer
                                                Network (Btx-Anbieter)
```

4.2. Funktionsweise von Datex-P

Aufbau:

```
                                                      University
                                               z. B. Karlsruhe
                                                      West-Germany
                                                        /     \
                                                       /       \
                 West-German City              Datex-P         Datex-P
Intruder -->        Telephone          -->     Digital -->     International
(Anrufer)           Exchange                   Network         Gateway \
                (Vermittlungsstelle)              \              /      \
                                                   \            /        \
                                             University             International
                                      z. B.  Bremen                 Record
                                             West-Germany           Carrier
                                                                  ____|____
                                                                 / / /  |  \ \ \
                                                                        |
                                                                     Tymnet

                                                                z. B. X.25
                                                                     Network
                                                                        |
                                                                        |
                                                                 Long Distance
                                                               Dialed Telephone
                                                                     Network
```

(siehe Anhang: Datex-P-Auslandsverkehrsbe-
ziehungen -->)

4.5. Datenschutz

Der Datenschutz ist eines der wichtigsten Themen unserer heutigen Zeit. Sogar die Politik hat dies, wenn auch noch verbesserungswürdig, erkannt, indem sog. Datenschutzbeauftragte beim Bund und in den einzelnen Ldndern existieren, die Datenmißbrauch aufdecken sollen bzw. für den Schutz der (unserer) Daten Verantwortung tragen.
Das dies jedoch nicht ausreichend ist, um Daten wirklich zu schützen, bewies umsomehr die Pioniertdtigkeit des CCC auf diesem Gebiet.

Der Btx-Hack:
Auf der DAFTA am 15. und 16. November 1984 in Kvln, einer Datenschutz Fachtagung, prdsentierten die Veranstalter einen richtigen Hacker. Ein Mitglied des CCC gab einige Erkenntnisse über das Btx-Sytem bekannt.
So z. B. wie man eine Btx-Anschlußbox so manipuliert, daß man auf fremde Kosten im System herumhacken kann. Der CCC habe im I.B.M-Programm schwere Fehler entdeckt, berichtete er, die unter Umstdnden dazu führen kvnnten, daß persvnliche Daten und geheime Paßwvrter in fremde Hdnde gelangen.
Kriminelle kvnnten diesen Fehler ausnutzen und arglose Btx-Teilnehmer um Riesenbetrdge prellen. Der Vertreter des Postministeriums auf der DAFTA stritt hingegen solche Mdngel rundheraus ab. Das System sei absolut dicht.
Der CCC war sich damals bereits sicher, Beweise für einen Software-Fehler in Hdnden zu halten. Und weil die Post auf der Datenschutztagung wenig kooperativ war, so ein spdterer CCC-Kommentar, mußte man mit diesem Beweis an die Vffentlichkeit, zur Warnung und zur Abschreckung potentieller Btx-Teilnehmer. Die Schlagzeilen am 20.11.84 ließen bei Post und I.B.M dann die Sirenen aufheulen: >ELEKTRONISCHER BANKRAUB IN BTX! - COMPUTER FANS ZAPFTEN DER HASPA 135.000.- DM VOM KONTO!<. Was war geschehen? Mit dem geheimen Paßwort der Hamburger Sparkasse sind die Daten-Chaoten ins Btx-System geschlüpft, wie mit einer elektronischen Tarnkappe. Dann haben sie - auf Kosten der Sparkasse - eine gebührenpflichtige Spendenseite aus dem eigenen CCC-Btx-Programm abgerufen, aber nicht nur einmal, sondern rund 13.500mal, eine ganze Nacht lang. Von Hand wdre das zu mühsam gewesen, deshalb wurde diese Arbeit von einem Computer erledigt. Da ein Abruf der Spendenseite (Text: >Es erforderte ein bemerkenswertes Team, den Gilb (Anmerkung: Gilb = Post) zurückzuweisen und ein Volk von 60 Millionen Menschen zu befreien.<) nicht weniger als 9,97 DM kostete, kamen rund 135.000.- DM zusammen, die der Hamburger Sparkasse berechnet und den Chaos Leuten mit der November-Telefonrechnung gutgeschrieben worden wdren. Die historische Gutschrift über knapp 135.000.- DM gibt es tatsdchlich, doch der Club hatte sofort verkündet, daß er das Geld gar nicht haben will. Zweck der Aktion sei ja nur gewesen, >die bei Btx vorhandenen Mdngel vffentlich darzustellen. Wir hdtten das auch mit 10 Pfennigen machen kvnnen, nur hdtte sich dann niemand dafür interessiert<.
>Hut ab vor dieser Leistung<, zollte der sichtlich irritierte Vorstandsvorsitzende der Hamburger Sparkasse den selbsternannten Datenschutz-Testern vor der Fernsehkamera respekt. >Blamabel und dußerst schmerzhaft< gestand die Post nach dem ersten Schreck. Wie das Chaos-Team letztlich an das geheime Paßwort der Hamburger Sparkasse gelangen konnte, dafür gab es später nach monatelangen Analysen von Post und I.B.M, verschiedene Erklärungen.
Version des CCC: >Die Post hat uns das Paßwort frei Haus auf den Bildschirm geliefert - durch einen Systemfehler<, hieß es in der CCC-Pressekonferenz.
Genauer: durch unkontrollierten Überlauf von Decoderseiten. Wenn ein Programmanbieter eine Btx-Seite gestaltet, passen genau 1626 Zeichen drauf. Mehr geht nicht. Was passiert aber, wenn jemand in Fleißarbeit genau 1626 Zeichen unterbringt und die Seite dann zum Abspeichern in den Rechner schickt? Das probierten die Jungs vom CCC aus. Sie füllten eine Seite bis zum Rand mit Zeichen, speicherten sie ab und riefen sie dann wieder auf.
Und dabei, sagten sie, sei urplvtzlich das Paßwort der Hamburger Sparkasse (usd70000) über die Mattscheibe geflimmert.
Dagegen die Version von Post und I.B.M: Einen unkontrollierten Seitenüberlauf kann es mvglicherweise gegeben haben, doch dieser Fehler sei nach Bekanntwerden sofort beseitigt worden. Niemals aber hdtte ein Paßwort aus dem System herauskommen kvnnen. Viel wahrscheinlicher sei, daß Mitglieder des Chaos Computer Clubs das Paßwort bei einer vffentlichen Vorführung der Sparkasse mitbekommen, also ausgespdht hdtten. Beigelegt ist dieser Konflikt bis heute nicht. Was die Postler wirklich wurmt: Sie kvnnen dem CCC einfach nicht positiv beweisen, daß das fragliche Paßwort nicht - mir nichts, dir nichts - auf den Bildschirm geflattert kam.
Nach eingehender Prüfung wollte übrigens auch der Bundesdatenschutzbeauftragte im Januar 1986 nicht mehr ausschließen, daß der Systemfehler >in Ausnahmefdllen< aufgetreten sein kvnnte.
Wie auch immer: Die Nachricht vom Btx-Coup schlug ein wie eine Bombe. Der CCC war in den Schlagzeilen und hatte Auftritte in Hvrfunk und Fernsehen.
Und die Post hat seither die Schlappe mit der Sparkasse nicht verwunden.
Derzeit hat das Bildschirmtextsystem nur rund 200.000 Teilnehmer, obwohl es nach den Hoffnungen der Gründer schon ldngst mehr als eine Million sein müßten. (Aus: Das Chaos Computer Buch, erschienen 1988, ISBN 3805204744)

Der Nasa-Hack:
Durch einen Fehler im Betriebssystem VMS der weltweit verbreiteten und fast überall im Einsatz befindlichen VAX, wurde dieser Hack, wohl weitaus bedeutender im Vergleich zum eher noch harmlosen Btx-Hack, mvglich.
In der Version VMS 4.4 vom Mdrz 1986 steckt ein Fehler, der die Integrität der Systeme erheblich tangiert. Das Betriebssystem stellt dem Benutzer einige hundert Systemaufrufe für Anwenderprogramme zur Verfügung. Das beschriebene Sicherheitsloch bezieht sich auf den Systemaufruf $SETUAI und erlaubt allen - also auch unberechtigten - Benutzern Schreibzugriffe auf die geschützte Datei SYSUAF.DAT. In dieser werden die Kennwvrter und Privilegien

der Benutzer verwaltet. Immerhin wird der Versuch, über die entsprechende Systemfunktion die Zugangskontrolldatei zu dndern, mit einer Fehlermeldung beantwortet. Durch den vorhandenen Softwarefehler kann jedoch die Fehlermeldung ignoriert werden: Die Datei bleibt gevffnet und kann nach Belieben modifiziert werden. Durch ldngeres Probieren und geschicktes Ausnutzen des VMS-Fehlers konnte wie folgt Kontrolle über die Rechnersysteme erlangt werden: Zundchst erfolgte der Rechnerzugang unter einem Gdsteeintrag oder über Netzwerkfunktionen (z. B. NETDCL), unabhdngig davon, welche Privilegien für den benutzten Zugang gesetzt waren. Mittels eines Maschinenprogramm wurden anschliessend durch Systemaufruf und weiterer Operationen alle Privilegien des verwendeten Zugangs nach belieben gesetzt. Nach einem wiederholten Einwdhlenunter dem verdnderten Benutzereintrag verfügten die Hacker über uneingeschrdnkten Systemzugriff. Danach war es ihnen mvglich, das jeweilige System erheblich zu manipulieren.
Das VMS-Sicherungssystem verschlüsselt die Kennwvrter nach der Eingabe mit einem Einwegverfahren und vergleicht die Ergebnisse mit dem jeweiligen bei der Kennwortvergabe einwegverschlüsselten Eintrag in der SYSUAF.DAT. Da es nahezu unmvglich ist, ein entsprechendes Entschlüsselungsverfahren zu finden, suchten und fanden die Hacker einen anderen, phantasievollen Weg. Beim Identifizieren gegenüber dem System wurde das Benutzerkennwort mittels einer eingebrachten Programmdhderung im Klartext abgefangen und verschleiert für die Hacker in freien Bereichen der Zugangskontrolldatei abgelegt. Je nach Belieben konnten die Hacker nun die so gesammelten Kennworte abrufen.
Um den privilegierten Zugang auch nach Systemdnderungen durch den Betreiber zu ermvglichen, wurde die Kennwortüberprüfung des Systems verdndert. Danach wird jede Kennworteingabe vor der systemüblichen Überprüfung mit einem von den Hackern eingerichteten Generalkennwort verglichen. Wird statt des Benutzerkennwortes der Generalschlüssel eingegeben, gestattet das System den Zugriff mit sdmtlichen Privilegien. Alle Zugangsbeschrdnkungen und Kontrollmechanismen sind dabei ausgeschaltet. In allen >besuchten< Rechnern wurde der gleiche Generalschlüssel hinterlegt, damit das Hacken nicht zu kompliziert wurde. Als >eigenen< Sicherheitsmechanismus verwandten die Hacker ein Kennwort, in dem auch unzuldssige Eingabezeichen vorkamen; ein Eindringen durch einen Tippfehler eines legitimen Benutzers wurde damit ausgeschlossen.
Durch Verdnderungen der entsprechenden Systemvariablen wurde die Anzeige so erzielter Zugdnge systemintern unterdrückt. Die Zugriffe wurden nicht protokolliert und dem Systemoperator sowie anderen Benutzern nicht angezeigt.
Die Hacker waren somit unsichtbar. Wdhrend ihrer über Monate andauernden Versuche gelang es der Hackergruppe schließlich, diese Manipulationen zu automatisieren. Die letzten Versionen ihrer trojanischen Pferde liefen als Rechenprozeß unbemerkt im Hintergrund ab, d. h. auch ohne Anwesenheit eines Hackers im Rechner. Es wdre durchaus mvglich gewesen, alle Systeme eines Netzes, die mit dem fehlerhaften Betriebssystem arbeiten, automatisch mit einem trojanischen Pferd auszustatten. Der Zeitaufwand betrug je System nur wenige Minuten.
Die fehlerhafte, im Mai 1986 ausgelieferte Version 4.4 des VMS-Betriebssystems wurde im Dezember 1986 durch die Version 4.5 ersetzt, die die gleichen fehlerhaften Mechanismen enthielt. Etwa seit Mai 1987 bot DEC eine >obligatorische<, aber nicht kostenfreie Nachbesserung des Sicherheitsprogramms an. Bei Nichtbeachtung kvnne, so die Ankündigung zu dem Programm, die Integritdt des Systems Schaden nehmen. Insgesamt waren es am Ende 135 Rechner, die sich in den Hdnden der Hacker befanden. Der Zugang zu diesen Rechnern wurde jedesmal erst u. a. durch Datex-P mvglich.
(Aus: Das Chaos Computer Buch, erschienen 1988, ISBN 3805204744)

5. Anhang

Die beiden folgenden Aufstellungen der Zugangstelefonnummern von BTX und DATEX-P sind für den Einsatz eines PC + Modem + Terminalprogramm (BTX: Decoderprogramm) gedacht (wenn auch nicht ausschließlich!).

Telefonnummern für den Zugang zu BTX - (Local Ports to reach BTX)

BTX Vermittlungs-stelle	Ortsnetz-kennzahl (Vorwahl)	Rufnummern für 1200/75 bit/s CCITT V.23	Übertr.geschw. 2400 bit/s CCITT V.22bis	64 kbit/s ISDN Leitung
AUGSBURG	0821	19 300	19 304	------
BAYREUTH	0921	19 300	19 304	------
BERLIN	030	19 300	19 304	19306
BIELEFELD	0521	19 300	19 304	------
BONN	0228	19 300	19 304	------
BREMEN	0421	19 300	19 304	------
DETMOLD	05231	19 300	19 304	------
DORTMUND	0231	19 300	19 304	------
DÜSSELDORF	0211	19 300	19 304	19 306
ESSEN	0201	19 300	19 304	------
FRANKFURT	069	19 300	19 304	19 306
FREIBURG	0761	19 300	19 304	------
GIEßEN	0641	19 300	19 304	------
HAMBURG	040	19 300	19 304	19 306
HANNOVER	0511	19 300	19 304	19 306
KAISERSLAUTERN	0631	19 300	19 304	------
KARLSRUHE	0721	19 300	19 304	------
KASSEL	0561	19 300	19 304	------
KIEL	0431	19 300	19 304	------
KOBLENZ	0261	19 300	19 304	------
KVLN	0221	19 300	19 304	------
KREFELD	02151	19 300	19 304	------
MAINZ	06131	19 300	19 304	------
MANNHEIM	0621	19 300	19 304	------
MESCHEDE	0291	19 300	19 304	------
MÜNCHEN	089	19 300	19 304	19 306
MÜNSTER	0251	19 300	------	------
NORDEN	04931	19 300	19 304	------
NÜRNBERG	0911	19 300	19 304	19 306

OFFENBURG	0781	19 300	19 304	------
OSNABRÜCK	0541	19 300	19 304	------
RECKLINGHAUSEN	02361	19 300	19 304	------
REGENSBURG	0941	19 300	19 304	------
SAARBRÜCKEN	0681	19 300	19 304	------
SALZGITTER	0534	19 300	19 304	------
SINGEN	0773	19 300	19 304	------
STUTTGART	0711	19 300	19 304	19 306
WIESBADEN	0611	19 300	------	------
WÜRZBURG	0931	19 300	19 304	------

Ist Ihr Ortsnetz nicht dabei, suchen Sie den ndchstgelegenen Ort aus der Liste und wdhlen dessen Ortsnetzkennzahl vor 193xx!

Telefonnummern für den Zugang zu DATEX-P - (Local-Ports PAD für Datex-P!)

AUGSBURG	0821	3 67 91	3 67 81	3 67 61	----
BERLIN	030	21 06 01	21 06 81	21 06 61	21 02 11
BIELEFELD	0521	5 90 11	5 90 21	5 90 41	----
BREMEN	0421	17 01 31	1 42 91	1 50 77	----
DORTMUND	0231	5 70 11	5 20 11	5 20 81	----
DÜSSELDORF	0211	32 93 18	86 31	32 07 48	13 45 33
ESSEN	0201	78 70 51	79 10 21	79 30 03	----
FRANKFURT	069	2 02 81	2 02 91	2 02 01	2 02 51
HAMBURG	040	44 12 31	44 12 61	44 12 81	44 12 91
HANNOVER	0511	32 66 51	32 74 81	32 75 91	54 81 81
KARLSRUHE	0721	6 02 41	6 03 81	6 05 81	----
KVLN	0221	29 11	29 31	29 51	29 71
MANNHEIM	0621	40 90 85	3 99 41	3 99 51	----
MÜNCHEN	089	22 87 30	22 86 30	22 87 58	29 99 78
NUERNBERG	0911	2 05 71	2 05 41	2 05 01	6 60 51
SAARBRÜCKEN	0681	81 00 11	81 00 31	81 00 61	81 00 81
STUTTGART	0711	29 91 71	29 90 61	29 92 91	87 03 21
WIESBADEN	06121 <0611>	3 60 11	3 60 41	3 60 81	----

PS: Nach Anwahl von Datex-P und einem hoffentlich fehlerfreien Connect, geben Sie >.< + return (enter) ein.

Folgende Aufstellung zeigt alle internationalen Datennetze, die via Datex-P zu erreichen sind. Durch diese Tabelle kann man z. B. anhand der ersten 4 Ziffern einer NUA feststellen, an welchem Datennetz und in welchem Land sich der Rechner befindet (z. B. 3106 = Vereinigte Staaten von Amerika = Tymnet), den Sie anwdhlen mvchten.

DATEX-P-Auslandsverkehrsbeziehungen - (Alle internationalen Datennetze)

In der folgenden Tabelle bedeutet:
A = noch festzulegende Netzkennziffer
--- = noch nicht bekannt

( ) = Angabe unter Vorbehalt
* = diese Netze sind z. Z. über Datex-P noch nicht erreichbar, aber Verbindungen aus diesen Netzen zu Datex-P-Anschlüssen sind mvglich ** = Übergdnge mit anderen Netzen (Telefon bzw. Telex)

LAND	NETZNAME	INTERNATIONALE NETZKENNZAHL
Dgypten	DATA TRANSMISSION TECHNICAL SERVICE (EGYPTNET)*	602A
Antigua und Barbuda	(AGANET)	3443
Argentinien	ARPAC ARPAC	7220 7222
Australien	AUSTPAC DATA ACCESS SERVICE TELETEX	5052 5053 5054
Bahamas	BATELCO	3640
Bahrain	BAHNET	4263
Barbados	IDAS*	3423
Belgien	DCS Sonderknoten DCS** DCS**	2062 2063 2068 2069
Bermuda	C&W (IDAS)*	3503
Brasilien	INTERDATA RENPAC RENPAC** RENPAC**	7240 7241 7248 7249
Bulgarien	BULPAC	2841
Chile	E-COM CHILEPAC TOMNET	7302 7303 7305
China	PKTELCOM*	(4600)
China (Taiwan)	PACNET I PACNET II UDAS	4872 4873 4877
Costa Rica	RACSAPAC RACSAPAC** RACSAPAC**	7122 7128 7129

Curacao	UDTS*	3620
Ddnemark	DATEX** DATAPAK DATAPAK	2381 2382 2383
Dominikansche Republik	UDTS*	3700
Elfenbeink¦ste	SYTRANPAC	6122
Finnland	DATEX** DATAPAK DIGIPAK	2441 2442 2443
Frankreich	TRANSPAC NTI VX 32* TRANSPAC** TRANSPAC**	2080 2081 2089 842A 933A
Franzvsische Antillen (Guadeloupe/ Martinique)	TRANSPAC	2080
Franzvsisch-Guayana	TRANSPAC	2080
Franzvsisch-Polynesien	TOMPAC-PF	5470
Gabun	GABONPAC	6282
Griechenland	HELPAK HELLASPAK	2022 2023
Grvnland	KANUPAK	2901
Großbritannien	BTI IPSS BT PSS MERCURY PNS HULL TELEMATIC	2341 2342 2350 2352
Guam	PACNET	5351
Guadeloupe	TRANSPAC	2080
Guatemala	GUATEL*	704A
Honduras	HONDUTEL*	7080
Hongkong	INTELPAK DAS DATAPAK	4542 4544 4545
Indien	GPSS	4042
Indonesien	SKDP	5101
Irland	EIRPAC	2724
Island	ICEPAK/ISPAK	2740
Israel	ISRANET	4251
Italien	ITAPAC ITAPAC	2222 2227
Jamaica	JAMANTEL*	3380
Japan	DDX-P VENUS-P	4401 4408
Jugoslawien	YUPAC	2201
Jungferninseln	(UDTS)*	3320
Kaimaninseln	---	3463
Kamerun	CAMPAC	6242
Kanada	DATAPAC GLOBEDAT-P INFOGRAM INFOSWITCH	3020 3052 3028 3029
Katar	via Bahrain*	427A
Kolumbien	COLDAPAQ*	7320
Korea (Republik)	DACOMNET	4501
Kuba	KUPAC*	368A
Kuwait	via Bahrain*	419A
Libanon	CEDARPAC	4155
Luxemburg	LUXPAC LUXPAC**	2704 2709
Malaysia	MAYPAC	5021
Malta	MALTAPAC	2782
Marokko	---	604A
Martinique	TRANSPAC	2080
Mauritius	MAURIDATA	6170
Mexiko	TELEPAC	3340

Namibia	SWANET	6490
Neukaledonien	TOMPAC-NC	5460
Neuseeland	PACNET	5301
Niederlande	DATANET 1 DATANET 1 DATANET 1**	2040 2041 2049
Norwegen	DATAPAK DATEX	2422 2427
Vsterreich	DATEX-P	2322
Portugal	TELEPAC-P	2680
Puerto Rico	UDTS*	
Reunion	TRANSPAC	20
Schweden	DATEX** DATAPAK DATAPAK	2401 2402 2403
Schweiz	TELEPAC	2284
Senegal	SENPAC	6081
Simbabwe	ZIMNET	6482
Singapur	TELEPAC	5252
Spanien	NID IBERPAC	2141 2145
S¦dafrika	SAPONET-P SAPOPAC	6550 6559
Thailand	(IDARC)*	520A
Trinidad und Tobago	TEXTEL DATANET	3740 3745
T¦rkei	ETPSDN (TURPAC)	2862
Tunesien	RED25	6050
UdSSR	IASNET	2502
Ungarn	DATEX-L**	2160
Uruguay	URUPAC*	7482
Vanuata	VIAPAC	5410
Venezuela	(PDVSA-P)	734A
Vereinigte Arabische Emirate	EMDAN TELETEX	4243 4241
Vereinigte Staaten von Amerika (USA)	ADP-AUTONET ALASKACOM ALASKANET AMERITECH PPDN AT&T ACCUNET BELL ATLANTIC ADV. SYSTEM BELL SOUTH PULSELINK CINCINATTI BELL COMPUSERVE CSC INFONET* DATA AMERICA FEDEXPRESS FTCC PST SERVICE* GEISCO MARKNET GLOBENET MCI DATA TRANSPORT MCI-MAIL NETEXPRESS NYNEX INFOPATH PACIFIC TELESIS PPSS RCA-LSDS SNET CONNNET SW BELL MICROLINK II TELENET TELENET TRT DATAPAK TYMNET U.S. WEST DIGIPAC WANG PAC SERVICE W.UNIONWORLDCOM INFOMASTER WU-WORLDCOM UDTS I WU-WORLDCOM UDTS II WUTCO PTN-1	3126 3135 3143 3134 3141 3142 3148 3132 3137 3151 3138 3124 3136 3150 3102 3104 3139 3144 3145 3113 3140 3146 3110 3127 3119 3106 3147 3149 3114 3107 3103 3101

Schlußbemerkungen

Fazit:
Auch heute noch ist es mvglich mit nur wenig Grundwissen und den richtigen Verbindungen in den Genuß zu kommen Datex-P kostenlos nutzen zu kvnnen.
Das Btx-System ist insgesamt sicherer und schwieriger zu knacken geworden, doch bei weiterer Benutzung von einfachen Paßwvrtern als simplen Schutz des Systems, kann auch hier ein erneuter Hack schnell zur Realitdt werden. Überhaupt kann man sagen, das Systeme, die lediglich einen Paßwortschutz verwenden niemals zu 100% sicher sein kvnnen und auch in Zukunft nicht sein werden.
Remember: Every day is a nice day to explore the digital highways, fly high and try to touch the sky!

Wegen Verbesserungsvorschldgen, Anmerkungen, Fragen oder was auch immer, bin ich immer dankbar und in der BBS Breaking the Wall unter +49 (0)30 854 46 19 zu erreichen. Bestimmt haben einige von Euch noch ganz andere Kenntnisse oder Erfahrungen mit den beiden Systemen gemacht, die für ein eventuell spdter erscheinendes Update dieses Textes von Bedeutung sein kvnnten!? Wenn dies der Fall sein sollte, so wdre es nett, wenn der oder diejenigen sich dazu mit mir in Verbindung setzen würden. Außerdem, so denke ich, kvnnten diese Informationen evtl. auch in englisch erscheinen, um so auch dem Ausland Gelegenheit zu geben, sich mit diesen Datennetzen vertraut zu machen (vielleicht findet sich da ja jemand!?, nicht, daß mein Englisch so mies ist aber ich habe einfach nicht die Zeit dafür diese Arbeit auch noch zu übernehmen). Deshalb meldet Euch bitte! Hmm, hat also doch tatsdchlich jemand diesen Text bis hierhin gelesen? Nunja, also wenn, dann hat sich die Arbeit ja doch gelohnt...!!!???

Phantom

MSG Nr. 08 von SPEE vom 13.12.1991 Betr. Datex-P 1 u.2 Teil
Paeckchen mit Datex - P
======================

Die Dfue ist ein Bereich der EDV, der viele Computer Begeisterte besonders fasziniert. Aehnlich der Bewegung in Raum und Zeit reizen die Datenreisen am meisten, die am weitesten in die Frage fuehren. Waehrend den Dfue - Spezialisten die Mailboxen in den heimischen Gefilden des Telefonortsbereichs ueber kurz oder lang einmal langweilig werden, versprechen weit entfernte Gegenstation und Mailboxen im Ausland Quellen Fuer heisse Informationen und neueste Public - Domain - Programme zu sein. Waehrend ihnen der Datenverkehr im Ortsnetzbereich keine groesseren Loecher in den Geldbeutel reisst, saugt Telekom bei der Reise ueber Kontinente doch schmerzende Betraege von ihrem Konto. Der Datenplausch beispielsweisse von nur einer Stunde mit einer Mailbox in den USA verschlingt gleich um die 250 Mark Telefongebuehren. Fuer den durchschnittlichen Steuerzahler bleibt somit haeufig, lange und regelmaessige Nutzung einer US - Verbindung per Modem unerschwinglich. doch zu dieser Datenreise haben Sie Alternativen. Denn neben dem bewaehrten Telefonnetz, welches aus Zeiten stammt, in denen die Wortwurzeln von >> digital << allenfalls im Lateinunterricht Erwaehnung fanden, hat die post eigens fuer den Datenkehr andere Leitungsnetze geschaffen.
Nach dieser Lektuere wissen Sie diese Datennetze besser zu nutzen, damit der sparsame Umgang mit den Ressourcen auch und insbesondere bei Ihrer Telefonrechnung greift, ohne auf die reizvollen Angebote gerade vieler internationaler Mailboxanbieter verzichten zu muessen. Fuer knapp 15 Dollar geht die Post das naechste Mal ueber das Datex - P - Netz ab, womit Sie schon eine Stunde in der regen DFUE - Szene bei Uncle Sam ueber dem Atlantik schnuppern koennen. Das Preissgefaelle zugunsten von Datex - P - Verbindungen ist in den meisten Faellen leicht geklaert: Das Datex - Netz der DBP - Telekom stellt ein voellig autonomes Leitungsystem dar, das vom Kernsystem her nicht mit Telefonnetz vergleichbar ist. Datex dient von Planung und Konzeption her ausschliesslich der Uebertragung von Computerdaten Worte koennen Sie also fernmuendlich ueber Datex nicht wechseln. Das liegt darin begruendet, das die Technik des Datex - und des Fernsprechnetzes zu verschieden ist, wobei Datex mit einer atemberaubenden Hoechstgeschwindigkeit von bis zu 64 KBit/s in seinem Kernbereich aufwarten kann. Dabei liegt die Anwahldauer meist noch bei weniger als zwei Sekunden. Und eine Rate von nur einem Fehler bei einer Million uebertragener Bits koennen Sie ja nun nahezu vergessen. Experten unterscheiden zwischen Datex - L und Datex - P. Datex - L aehnelt insofern dem Telefonnetz, qls Sie zwischen zwei Datenstationen fuer die Dauer der Uebertragung eine Verbindung aufbauen. Jede Datenstation identifizieren Sie mit einer eigenen Nummer. Waehrend der gesamten Kommunikation verbindet Sie eine Leitung mit der Gegenstation. Kosten tragen Sie nach Dauer und nach der Entfernung der Verbindung.
Doch wegen der monatlichen Fixkosten mit zusaetzlichem Hardware - Aufwand nutzen Datex - L vorrangig diejenigen, die Dfue als Job betreiben muessen.
Im Gegensatz dazu lohnt sich die Datex - P - Reise sowohl fuer DFUE-Weltenbummler als auch fuer gelegentliche Kurzurlauber in fern Informationsgefilde, obgleich die damit einhergehende Technik

HARDWARE * SOFTWARE

COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80

ERFAHRUNGSAUSTAUSCH

CLUBINFO

40. Ausgabe

Kontaktadresse:

Club 80

Hartmut Obermann

Mozartring 23

Postfach 1430

8870 Günzburg

Tel.: 08221/30248

Anrufbeantworter &

FAX: 08221/33575

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Neues vom Vorstand — 1
Hartmut Obermann

Termine — 2
Redaktion

Gedanken zu den Clubinfos — 3 - 4
Günther W. Braun

... zum Clubinfo — 5
Redaktion

Sonstiges

128er Sonderheft 89 — 6 - 12
Günther W. Braun

Der Mailboxdienst GEnie — 13 - 42
Egbert Schrör

Die letzten Seiten

Impressum — 43
Schluß — 44
Redaktion

Mitgliederadressenliste — am INFO-Ende

Neues vom Vorstand

Wie (hoffentlich) in jedem Info möchte ich an dieser Stelle einige Neuigkeiten aus der "Vorstandsetage" des CLUB 80 loswerden. Fangen wir gleich mit etwas erfreulichem an.

1. CLUB80.DISC
   Das "Schwarze Brett", welches Klaus Hermann in seiner Mailbox für den CLUB 80 eingerichtet hat, funktioniert inzwischen einwandfrei. Zu Beginn gab es noch ein paar kleine Probleme mit dem Offline-Reader bzw. den unterschiedlichen Packformaten unter MSDOS und CP/M, aber diese Probleme scheinen nun auch behoben zu sein.

   Die Resonanz der CLUB 80-Mitglieder auf dieses neue Kommunikationsmedium läßt zwar leider noch etwas zu wünschen übrig, aber ich bin guter Hoffnung, daß die Zahl der DFÜ-Freaks mit der Zeit noch ansteigen wird.

   Inzwischen liegen auch einige Dateien und Programme zum Download bereit. Neben dem Offline-Reader CRR0140 finden sich auch einige Programme zum Komprimieren und Archivieren von Dateien und Programmen (ARK, ZIP). Demnächst werden wohl auch die Bücherliste von Kurt Müller sowie die Listen der verfügbaren Shareware in der DISCOVERY zu finden sein. Kurt Müller hat mir seine Liste, die auf einem ATARI erstellt wurde, inzwischen im ASCII-Format zukommen lassen. Leider konnte ich sie aber bisher noch nicht so aufbereiten, daß ein Import in dBase möglich wäre. Aber nach dem Urlaub gehe ich an die Lösung dieses Problems - versprochen!

2. FzAdM
   Auch der Fragebogen zur Aktualisierung der Mitgliederkartei war ein voller Erfolg. Zwar sind bisher nur zwanzig Fragebogen bei mir eingetroffen, aber ich schreibe das eher der späten Auslieferung der neuen Postleitzahlenbücher als dem Austrittswillen der CLUB 80-Mitglieder zu. Wer den Fragebogen allerdings nicht spätestens bis zum Erscheinen der Info-Ausgabe 41 an Jens oder mich zurückgeschickt hat, ist raus aus dem CLUB 80. Also unbedingt die neue Postleitzahl organisieren, den Fragebogen möglichst vollständig ausfüllen und an Jens oder mich schicken!

Damit bin ich für diesmal schon am Ende angelangt. Ich wünsche euch, egal ob ihr Urlaub habt oder arbeiten müßt, einen schönen Sommer und weiterhin

Allways good Hacking,

PS: Man liest sich in der CLUB80.DISC!

Fliegen für Senioren billiger

Aus: B. Z.

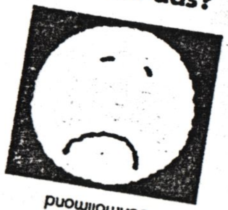
Fein! Und was sollen sie damit machen?

Anfrage

Warum sind einige Kollegen so gehässig gegenüber der sympathischen Moderatorin der Sportnachrichten? Kürzlich sprach einer von „Sissi, dem Aas"!

Fritz-Jürgen Busch, Hamburg

Was ist das?



Schmollmond

Lieber 'ne frohe Kunde als 'ne traurige Verkäuferin.

Termine... Termine... Termine... Termine... Termine

HAM RADIO	Friedrichshafen	26.06.- 27.06.93
IFA	Berlin	27.08.- 05.09.93
Fachmesse Elektrotechnik	Dortmund	01.09.- 04.09.93
MICRO SYSTEMS TECH. 1993	Berlin	21.09.- 24.09.93
SYSTEMS	München	18.10.- 22.10.93
PRODUKTRONICA	München	09.11.- 13.11.93
Nord Elektro	Hamburg	18.11.- 20.11.93

Redaktionsschluß für das nächste Clubinfo ist der

01. September 93.

Gedanken zu den Clubinfos

Da bei der Hauptversammlung im März weniger als 10 % der Gesamt-Mitglieder anwesend waren, wurde ich aufgefordert, meine Verbesserungs-Vorschläge allen schriftlich im Info zugänglich zu machen. Durch Fragebogen besteht dann eine Möglichkeit zur Abstimmung. Und vielleicht hat der eine oder andere, dem der Anfahrtsweg nach Burgau zu weit war, trotzdem ein Interesse an der Mitgestaltung unserer Zeitschrift.

Seit Jahren wird geklagt, es kämen immer weniger Beiträge zum Abdrucken, sodass im vergangenen Jahr nur noch drei Hefte in unregelmässigen Abständen herausgegeben werden konnten. Aufgrund dieser unregelmässigen Erscheinungsweise fragen sich manche Leser ("Abonnenten"), wofür sie eigentlich noch Beitrag zahlen, und springen ab. Durch die geringer werdende Mitgliederzahl wird aber automatisch auch die Zahl der Artikel-Schreiber geringer: Der Teufelskreis schliesst sich. Und dabei ist eigentlich jedem klar, dass eine Clubzeitschrift, die keine Honorare an Profischreiber zahlt, nur von der Mitarbeit der Mitglieder existieren kann!

Was könnte man also noch tun, als die Mitglieder immer wieder zu ermuntern, die Erfahrungen mit ihren Rechnern doch schriftlich niederzulegen und in der Clubinfo den anderen auch zugänglich zu machen? - Meiner Meinung nach könnte man bei dieser Gelegenheit die Infohefte selbst und ihre Erscheinungsweise noch verbessern und professioneller gestalten:

Ein erster Schritt dazu wurde von Redakteur Jens bereits vorgeschlagen: Anstatt Berichte auf grauem Umweltschutzpapier einzusenden (die oft kaum kopiert werden können), wäre eine Einsendung auf IBM-kompatibler Diskette vorteilhaft. Jens könnte die Berichte dann leichter ins Heft-Format bringen, und auch ein einheitlicheres gut lesbares Schriftbild wäre dadurch möglich.

Ein weiterer Schritt wurde von der Hauptversammlung dieses Jahr beschlossen und muss jetzt konsequent verwirklicht werden. Egal wie wenig Beiträge zum jeweiligen Redaktionsschluss vorliegen, die Hefte sollten regelmässig vierteljährlich erstellt und zum Versand gebracht werden. Mit welchen durchschnittlichen Seitenzahlen der Infos wir dabei rechnen können, zeigt eine kleine Statistik der Heftstärken der vergangenen beiden Jahre:

Mrz. 91	68 Seiten	
Juni 91	42 "	
Okt. 91	68 "	178:4 = 44 Seiten (1991)
Jan. 92	56 "	
Okt. 92	80 "	
Dez. 92	48 "	184:4 = 46 Seiten (1992)

Teilt man also die Gesamtzahl der eingegangenen Beiträge auf 4 Hefte auf, so hätte ein Heft zwischen 40 und 50 Seiten. Hefte mit 100 und mehr Seiten (wie zuletzt im Dezember 1990) gehören somit - bei regelmässiger Erscheinungsweise - der Vergangenheit an.

Der nächste Schritt, der nun gegangen werden müsste, um eine Zeitschrift herzustellen, die auch professionellen Ansprüchen genügt, wäre eine *Umstellung des Formats auf Hochformat*. Oder kennt in unserem Mitgliederkreis wirklich einer eine professionelle Zeitschrift im Querformat, bei der eine Seite zwei Seitennummern hat? Dieses Verfahren war früher zweckmässig, um bei der grossen Anzahl an eingehenden Artikeln die Hefte nicht zu dick werden zu lassen. Das hatte Auswirkungen auf die Versandkosten bei der Post, und natürlich auch auf die Freizeit von Jens, dem wir ja nicht zumuten wollen, dass er am Feierabend noch Berge von Kopien herstellt und sortiert. Doch bei jetzt durchschnittlich 40-50 Seiten - das sind also nur noch 20-25 Blätter pro Info - ist er wahrscheinlich nicht überfordert. Notfalls würde sich vielleicht sogar ein interessiertes Mitglied in seiner Nähe finden, das ihm bei der Herstellung und beim Verschicken zur Hand geht?

Ein paar Bedenken, die gegen das Hochformat laut wurden, kann man - glaube ich - leicht zerstreuen. So ist es in Zukunft nicht nötig, dass das Layout auf jeder Seite "Club 80 Info-Nr. und Datum" vorsieht. Eine ganz normale Seitennummer tut's auch; und wenn zunehmend mehr Berichte auf Diskette eingeschickt werden, können die Seitennummern vielleicht sogar automatisch eingesetzt werden. - Was den Umweltschutz bei höherem Papierverbrauch betrifft, so könnte man künftig das moderne chlorfreie weisse Umweltschutzpapier verwenden. - Und bezüglich der Portokosten ist die Sachlage seit Einführung der neuen Tarife gar nicht so schlecht. Durch den Wegfall von "Drucksachen" will die Post zwar die Flut von Werbesendungen zu verbilligten Tarifen eindämmen. Aber werbungsfreie Drucksachen gelten jetzt als "Büchersendung", und da ist es völlig gleichgültig, ob unser Info 5 oder 50 Blätter dick ist, das Porto beträgt bis zu 500 g nur DM 1,50.

Wer die beiden grünen "Fragebogen zur Aktualisierung der Mitgliederkartei" aus Info 39 ausfüllt, kann auch darüber abstimmen, ob unsere Hefte in Zukunft im Hochformat erscheinen sollen. Wer nichts verändern will, kann auch nichts verbessern! Das gilt dann beispielsweise auch für unsere Mitglieder-Listen in den Infos. Wo mit den Fragebogen auch Angaben zu Hardware und Betriebssystem gemacht werden, könnten diese Angaben doch auch in den Mitglieder-Listen künftig aufgeführt werden. Ebenso könnte dort vermerkt sein, ob jemand BTX hat oder in welcher Mailbox er zu erreichen ist. Wäre das kein Fortschritt? Ich weiss zwar nicht, wann Euch dieses Heft erreicht. Wer seine Fragebogen aber noch nicht abgeschickt hat, sollte das jetzt sofort nachholen. Auch wenn es nach dem 1. Juli ist, wird sie Hartmut Obermann sicher nicht wegwerfen!

Günther W. Braun

... zum Clubinfo

weitere Anmerkungen der Redaktion

Liebe Club 80'er, noch immer steht das Thema Club-INFO auf dem Tablo: Zu den mir wichtigsten Punkten, die ich schon im Schluß von INFO 39 aufgezeigt habe, möchte ich eigentlich weiter nichts hinzufügen. Zu einigen von Günthers Anmerkungen will ich aber doch die "historische Geschichte" oder das "Warum und Wiso" hinzufügen.

Zuerst möchte ich einmal den Zeitaufwand und die Tätigkeiten zur Infoerstellung aufführen, damit Ihr einmal seht, daß das auch nicht zwischen Frühstück und Vesper erledigt ist. Als Beispiel soll das letzte INFO dienen, der Zeitaufwand kann dann aufgrund der Verhältnisrechnung für jedes Info nachvollzogen werden.
Auflage des Heftes: 80 Stück davon je 31 Blatt doppelseitig und 5 Blatt einseitig kopiert. Dies sind 5360 Kopien wobei unser Kopierer ca. 15 Kopien pro Minute kann. Somit liegt der Zeitaufwand allein für das Vervielfältigen bei ca. 6 Stunden. Zeiten für Wartung, Papier und Toner nachfüllen sind noch nicht mitgerechnet! Für das Layout, Sortieren der Kopien und Kleben sowie das Versandfertigmachen der Hefte ist fast nochmals der gleiche Zeitbedarf zu rechnen. Somit war ich knapp <u>12 Stunden</u> mit dem INFO voll beschäftigt (ich bin selbst überrascht wieviel Zeit dafür draufgeht, ich hatte vorher noch nie Zeitmessungen betrieben.). Eine Verdoppelung des Zeitaufwandes durch ein Verdoppeln der Kopienzahl und der zugehörigen Arbeiten ist für mich nicht akzeptabel.

Als weiteres wurde zum Clubtreffen, um die Aktualität zu wahren, vereinbart alles was bis zum Stichtag von Euch kommt auch zu bringen. Hefte mit immer gleicher Blattanzahl wird es somit sicher nicht geben, auch wenn sich das schön rechnen läßt.

Nun zu der "Professionalität" unseres INFO's. Ich glaube nicht, daß wir gegen eine komerzielle Zeitschrift konkurieren zu haben. Die Rechts/Links-Seitenbeschriftung war übrigens ein Mitgliederwunsch, der ab Heft 7 eingeführt wurde. (Rein in die Kartoffeln, raus aus die Kartoffeln!) Nichts desto Trotz möchte ich behaupten, daß unser INFO im Vergleich mit anderen Club-Info's, die auch in Heimarbeit erstellt werden, sicher bestehen kann. Ansonsten darf ich noch anmerken, daß dieses Format als vorbereitendes DIN A5-Format zwischenzeitlich auch bei der TELEKOM Verwendung findet. Auch hier zählen Kostenminimierung und Umweltschutz.

Zum Abschluß nun noch zur Versendungsform. Unser Clubinfo gehört laut Aussage der Post nicht zu den Büchersendungen. Ich hatte diese Versendungsform auch schon gewählt. Aber nach mehreren Konsultationen auf dem Postamt, zu denen ich hinzitiert wurde, gehe ich den teureren aber sicheren Weg. So erhält jeder sein INFO auch pünktlich zugesandt. Postens kontrolliert nämlich die Büchersendungen und abhängig vom Prüfer, ist es eine oder nicht (wie Lotto). Nachdem ich, als Versender, immer den Anschiß bekam habe ich das Thema Büchersendung gestrichen. Wenn es als Büchersendung klappt, kann man sich freuen.

Ich denke, Ihr könnt das Clubinfo in seiner bisherigen Form weiterhin akzeptieren. Wichtig ist vor allem, finde ich, der Informationsgehalt eines Clubinfos und der ist abhängig von der Aktivität und Mitarbeit der Clubmitglieder. Mir persönlich würde es, darauf lege ich Wert, sehr viel zusätzliche Arbeit ersparen!

Euer Jens

<u>128er Sonderheft Nr. 89 von M & T:</u>
<u>Professionelles Heft mit CP/M-Beiträgen</u>

Einigermassen regelmässig alle halbe Jahre bringt der Markt & Technik Verlag immer noch Sonderhefte für den Commodore 128 heraus; das vorige mit der Nummer 82 war im September 1992 erschienen. Das Heft vom April dieses Jahres enthält ausser Programm-Beschreibungen zum Commodore-Betriebssystem auch Neuigkeiten über CP/M-Software (vergl. Inhaltsverzeichnis) - erstaunlich, oder nicht?

<u>128'er Software</u>

Eigentlich ist das Heft seinen Preis schon wegen der 128'er Programme wert, die in bewährter Weise auf Diskette gleich mitgeliefert werden. Darunter befinden sich zwar auch Sachen, die einen langjährigen Leser des 64'er Magazins nicht mehr vom Stuhl reissen: So wieder einmal eine BASIC-Erweiterung, und Konvertierprogramme ins PC-DOS Format sind auch nichts Neues mehr! Die Vorstellung einiger neuer Versionen aus dem gesamten Software-Spektrum bringt aber sicher für jeden etwas Lohnendes - von der Musik über die Graphik bis zu Zusatzprogrammen zur Textverarbeitung "Mastertext" (davon gibt's bei uns übrigens ein Update zur Version 3.0). Die meisten Anwender ziehen es ja vor, sich in verbesserte Updates eines bekannten Programmes einzuarbeiten, als ihre bisherige Software immer wieder durch völlig andere neue zu ersetzen.

Ein wirklicher Leckerbissen ist das Update des Disketten-Monitors "BURSTMON 4.0", mit dem man auf 1571-Laufwerken auch MFM-Disketten (also CP/M-Formate) verarbeiten kann. Das übersichtliche Menü erlaubt ein problemloses Initialisieren eingelegter Disketten, worauf zunächst das Format analysiert und angezeigt wird. Anschliessend kann man beispielsweise auf jeder beliebigen Spur jeden Sektor lesen und auch beschreiben, wirkliche Spitzenklasse! Die ursprüngliche Version des "BURSTMON" gehört zum Fachbuch "Die Floppy 1570/1571" von Karsten Schramm, das bereits 1986 bei Markt & Technik erschienen ist. Dementsprechend werden auch die Besitzer dieses Buches ein besonderes Interesse am jetzigen Programm-Update haben. Da im Sonderheft aber die Funktionen des Programms ausführlich beschrieben sind, ist das Programm auch für Leute, die nicht im Besitz des Buches sind, voll geeignet.

<u>CP/M Public Domain Software</u>

In einem mehrseitigen Artikel zu diesem Thema werden zunächst alle bedeutenden amerikanischen Computerclubs erwähnt, die Public Domain Sammlungen anlegten und diese Software somit erst der Öffentlichkeit zugänglich machten. Dass der Autor des Artikels klagt, diese Clubs würden Disketten leider nur an Mitglieder abgeben, ist allerdings nicht ganz verständlich: Schliesslich ist es doch das Wesen eines Clubs, dass die Mitglieder soli-

darisch miteinander und füreinander etwas unternehmen (wozu auch finanzielle Leistungen notwendig sind), und nicht, dass man Aussenstehenden, die keinerlei Verpflichtungen eingehen wollen, möglichst kostenlos Beratung und Software anbietet!

Als Software-Diskothekar für C-128 Besitzer - und für diesen Anwenderkreis habe ich den "Markennamen" "CP/M aktuell" gewählt - kann ich hier mitteilen, dass all' diese Public Domain Disketten durch mich erhältlich sind. Z. Z. haben wir allein über 500 KB Katalog-Dateien, die auf zwei Disketten im Format mit 40 Spuren pro Seite erhältlich sind. In unserem 1986 gegründeten Club befinden sich einige alte Hasen, die heute noch laufend Beziehungen zu den USA unterhalten und die CP/M-Software - wenn auch oft mit einiger Verzögerung - zu uns herüber bringen.

Bei dieser Gelegenheit gleich noch ein paar Worte zu den Preisen: Die Beschaffung und Verwaltung dieser Software kostet natürlich Geld, und daher kann auch P-D-Software nicht ganz kostenlos sein. Wenn sich der Diskothekar schon die Arbeit macht, sollte man nicht erwarten, dass er einem die Disketten auch noch schenkt. Und wenn dazu im 128er Sonderheft erwähnt wird, dass von Computerclubs für P-D-Disketten Kopierkosten von DM 3,-- pro Stück verlangt werden, dann ist das erst die halbe Wahrheit. Dazu kommen dann nämlich die Kosten für die Diskette, sowie Kosten für den Versand (die Post arbeitet ja auch nicht kostenlos). Wenn ein Club dann aber verlangt, dass man Leerdisketten mit frankiertem Umschlag erst einschicken soll, dann kann es teurer werden als zehn Mark, denn nach den neuesten Posttarifen kostet ein Brief mit einer Diskette ja schon allein drei Mark.

Die Frage nach diskettenplatz-sparender Speicherung, die im Zusammenhang mit P-D-Software oft auftritt, wird im 128er Sonderheft ebenfalls ausführlicher erörtert. Dabei fehlt jedoch die klare Unterscheidung zwischen komprimierten und gepackten Dateien: die letzteren werden selbst wiederum aus (möglicherweise komprimierten) Einzeldateien zu einer einzigen "zusammengepackt". Weiterhin sollte auch nicht der Eindruck entstehen, die Wieder-Gangbarmachung derartiger Dateien sei eine komplizierte Arbeit, die eine Bibliothek von Hilfsprogrammen und eine Menge Fachwissen nötig mache. Wer von uns P-D-Disketten bezieht, bekommt die eventuell erforderlichen Hilfsprogramme mit Anleitung selbstverständlich dazu.

Ein Thema, mit dem sich C-128 Anwender hierzulande meist noch nicht beschäftigt haben, ist die ZCPR-Software; möglicherweise, weil der C-128 (D) mit seinem Z-80 Prozessor und dem CP/M-3 Betriebssystem einer der letzten und modernsten CP/M-Rechner ist, und seine Besitzer nicht als erstes am Betriebssystem herumbasteln. Für die älteren CP/M-2-Rechner mit Z-80 - also auch die kleinen Amstrad/Schneider CPC 464 und 664 - wurde in den USA jedoch schon vor Jahren eine Betriebssystem-Verbesserung entwickelt, wobei man den zum Standard-Betriebssystem gehörenden Console Command Processor (beim C-128 die selbständige Datei "CCP.COM") durch einen ZCPR (Z 80 Console Processor Replacement) ersetzt. Das hat dann die Vorteile komfortablerer Bildschirm-Bedienung und eines Geschwindigkeits-Gewinns, und die speziell dafür entwickelte ZCPR-Software läuft damit.

Auf dem C-128 hat der ZCPR allerdings nicht nur Vorteile. Er besetzt im nicht gerade üppigen Arbeitsspeicher (TPA) beispielsweise noch rund 5 KB, und von den bekannten Marken-Programmen werden nur zwei durch den ZCPR unterstützt: Die Textverarbeitung "Wordstar 4.0" und der C-Compiler "BDS-C". Man muss also schon ganz schön experimentierfreudig sein, wenn man DM 99,-- für das ZCPR-Installations-Paket "Z3PLUS" ausgibt, nur um anschliessend die 40 Mega-Bytes ZCPR-P-D-Software anschaffen und durchforsten zu können!

CP/M Disketten-Konvertierprogramm "Juggler"

Ein Marken-Programm, das erst 1988 in Canada entwickelt wurde und somit zu den jüngsten CP/M-Programmen zählt, wird im 128er Sonderheft zum ersten Mal vorgestellt: JUGG'LER - 128 V 3.4 mit deutschem Handbuch. Es handelt sich dabei um die Commodore-Anpassung eines komfortablen Konvertierprogramms von und nach 170 verschiedenen Disketten-Formaten (vergl. nachfolgende Liste). Schade, dass uns so ein Programm nicht schon 1988 zur Verfügung stand! Das Programm enthält auch Funktionen zur Format-Analyse und zum Formatieren in den entsprechenden Formaten auf dem C-128. Da dieser Rechner mit seinen Laufwerken allerdings keine 5 1/4"-Disketten mit 80 Spuren pro Seite verarbeiten kann (wie es bei moderneren Rechnern verbreitet ist), kann das Programm derartige Formate auch nicht herbeizaubern. Wer die Formatliste durchsieht kann selber feststellen, ob sich Disketten-Formate darunter befinden, die eine Anschaffung des Programms rechtfertigen. Nur wegen der MS-DOS Formate für IBM-kompatible Rechner beispielsweise, lohnt es sich nicht: Dafür haben wir auch ein preisgünstiges P-D-Programm in unserer Sammlung.

Die Übersetzung des Handbuchs besorgte Helmut Jungkunz, Software-Diskothekar im SCUG (Schneider Computer User Group in München), bei dem auch das ganze Programm-Paket für DM 70,-- erhältlich ist. Von seinen Schneider-Anwenderkollegen wird er offensichtlich wohl nicht mehr ausreichend beschäftigt? Doch nachdem die letzte Amstrad/Schneider-Zeitschrift in Deutschland, die "Amstrad PC International" zum Ende des vergangenen Jahres das Handtuch geworfen hat, weiss man nicht, ob auch dieses C-128 Sonderheft von Markt & Technik vielleicht das letzte seiner Art sein wird!

Jugg'ler-128 V 3.4 unterstützt folgende Disketten-Formate:

A B C - 80	SS	Ampro	(1581)	SS
A B Dick Magna III	DS	Archive		
Access Matrix		Arisia		SS
Actrix	SS	ATR-8000		SS
Actrix	DS	ATR-8000		DS
Adler Textwriter	SS	Avatar		DS
Adv. Digt'l Super-6	SS	Beehive		DS
Adv. Digt'l Super-6	DS	Bitelex		SS
Adv. Digt'l T-DOS 312k	DS	Bondwell 12		SS
Adv. Digt'l T-DOS 366k	DS	Bondwell 14		DS
Amigo	SS	Bondwell 2	(1581)	SS
Amigo	DS	Burr-Brown		DS
Ampro	SS	C-64 CP/M (GCR)		SS

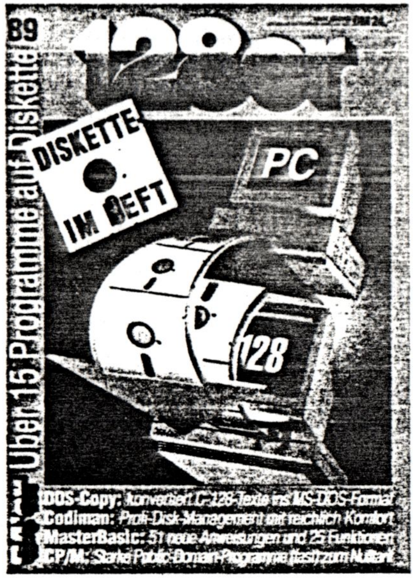
C-128 CP/M (GCR)	SS
C-128 CP/M (GCR)	DS
C-128 CP/M (1581)	DS
Cal PC	DS
CCS (256 byte/sector)	DS
CCS (512 byte/sector)	DS
CCS (1024 byte/sector)	DS
CDI-5000	DS
Coleco ADAM CP/M	
Commodore DOS (GCR)	
Compugraphic MCS 5	SS
Compustar 30	DS
Cromemco C-10	
Cromemco CDOS	SSSD
Cromemco CDOS	DSSD
Cromemco CDOS	SS
Cromemco CDOS	DS
Cromemco Int'l Term	SS
Cromemco Int'l Term	DS
C/WP Cortex	SS
Cykey	DS
DEC Robin	
DEC VT-180	SS
DEC II/Rainbow (1581)	
Digilog	DS
Direct 1025	DS
Epson QX-10 (256)	DS
Epson QX-10 (256-Euro)	DS
Epson QX-10 (512)	DS
Formula 1	DS
Fujitsu Micro 8	DS
Fujitsu Micro 16-s	DS
Gemini (1581)	
Glass Typewriter II	
Glass Typewriter (1581)	
Gnat Sys. 10	DS
Heath Magnolia	SS
Heath Magnolia	DS
Heath Magnolia (1581)	
Hewlett Packard 86, 87	DS
Hew.-Pack. 125 (1581)	SS
IBM-8 (CP/M-86)	SS
IBM-8 (CP/M-86)	DS
ICM Turbo DOS	DS
IMS-5000	SS
IMS-5000	DS
Insight Enter.	DS
Jet-80	DS
KayPro II, 2	SS
KayPro 4, 10	DS
Labtam (1581)	
Lexoriter	SS
Librarian 1571	DS
Librarian 1581 (1581)	DS
Lobo Max-80 (256 byte)	SS
Lobo Max-80 (256 byte)	DS
Lobo Max-80 (512 byte)	DS
L.N.W.-2	SS

Magic	DS
MAXI 1571	DS
MAXI 1581 (1581)	DS
MG 1581 (1581)	DS
Mega-88 Turbo DOS	DS
MicroBee	DS
Micron Quark	DS
Micro Source M-6000	DS
Molecular S-9	DS
Morrow MD2	SS
Morrow MD3,5,11,16,34	DS
Morrow Turbo DOS	DS
Mountain Control	DS
MS-DOS (8 sector)	SS
MS-DOS (8 sector)	DS
MS-DOS (9 sector)	SS
MS-DOS (9 sector)	DS
Multiflex	
NCR Decision Mate	DS
NEC PC-8001	SS
NEC PC-8801	DS
NEC PC-8431 (1581)	SS
Nelma Persona	SS
Olivetti 240 (1581)	
Olivetti ETV-300	SS
Olivetti M20	DS
Olympia ETX-II	SS
Olympia EX-100	DS
Osborne 1	SSSD
Osborne 1/Executive	SS
Osborne G2	DS
Osborne Nuevo/Vixen	DS
Otrona Attache	DS
Panasonic KXE-828	DS
Pegasus Data Logger	
Philips PC-2010	SS
PMC MicroMate	DS
Portapak (1581)	
Royal Alphatronic	DS
Sage IV	DS
Sanyo MBC-1000, 1150	DS
Seequa Chameleon	SS
Seequa Chameleon	DS
Sharp MZ-80	
Slicer	DS
Sony SMC-70 (1581)	SS
Spectravideo SV-328	DS
SuperBrain Jr.	
SuperBrain 40 Track	
SuperBrain	
Systel 2	SS
Systel 3	DS
Tektronix 4170	DS
Telectronics CP/M 681	
Teletek SystemMaster	DS
TeleVideo 801, 806	DS
TeleVideo T-DOS	DS
TI Pro. CP/M-86	SS

Tomcat 1800G	
Toshiba T-100	DS
Toshiba T-200	DS
TRS-80 Model 1	
TRS-80-1/Omicron	SSSD
TRS-80 Color Power II	SS
TRS-80 Mod 3/Holmes	DS
TRS-80 Mod 3/M.M.	SS
TRS-80 Mod 4/MTZ (256)	SS
TRS-80 Mod 4/MTZ (256)	DS
TRS-80 Mod 4/MTZ (512)	SS
TRS-80 Mod 4/MTZ (512)	DS
TRS-80 Mod 4 CP/M+	SS
Wang-Writer	DS
Wavemate Bullet	SS

Xerox 16/8	DS
Xerox 820	SSSD
Xerox 820-II	SS
Xerox 820-II	DS
Xerox/Sunrise Laptop	
XOR-5	DS
Zenith Z-37	SSSD
Zenith Z-90	SS
Zenith Z-89/90	DS
Zenith Z-89 (1581)	
Zenith Z-100	SS
Zenith Z-100	DS
Zorba	DS

(Günther W. Braun)



Anwendungen

Turbo-Diskmanagement
»CODIMAN 128«: Unser vollständig menügesteuertes Tool ordnet Ihre Diskettensammlung im Handumdrehen: Bis zu 2000 Einträge lassen sich pro Datei verwalten! **4**

Lieferant für Mastertext
»Datenmanager 128«: Diese Adressenverwaltung läßt in puncto Komfort keine Wünsche mehr offen - Sahnehäubchen: Die Datensätze sind kompatibel zur Mastertext-128-Rundbrief-Funktion. **8**

Meter sind gefragt!
»Umrechnen«: Gallons, Miles, Inches, Fahrenheit - da raucht der Kopf, wenn man solche Maßeinheiten in hierzulande verständliche Werte umwandeln will. Benutzen Sie doch unser Programm! **11**

Drucker

Phantastische Schriften mit dem NEC-P 2200
»Zenec 2.0 und Masterfont«: Jetzt zeigt Ihr 24-Nadler erst, was er zu leisten vermag: Mit unserem exquisiten Font-Editor läßt sich jeder beliebige Druckerzeichensatz kreieren. **14**

CP/M

CP/M-Programme (fast) zum Nulltarif
Unglaublich, wie viele Programme, Utilities und Tools es noch fürs CP/M-Betriebssystem gibt. Die Zauberformel heißt Public Domain. Wir haben einige Perlen herausgegriffen und zeigen Ihnen, wie Sie an diese Software kommen. **21**

...aber bitte mit Format!
»Jugg'ler«: Ab sofort steht es weit offen, das Tor zur Welt der CP/M-Diskformate: Diese kommerzielle Software erkennt, liest und beschreibt mehr als 170 unterschiedliche Versionen! **24**

Erweiterung

Die höchste Stufe
Welcher C-128-Basic-Dialekt hat einen komfortableren, besser strukturierten und umfangreicheren Befehlssatz als das integrierte Basic 7.0? Masterbasic 128 – mit 51 zusätzlichen Befehlen und 25 neuen Funktionen! **26**

Floppy 1571

Das Tor zur DOS-Welt
»DOS-Copy«: Komfortabler geht's nicht – unser Tool für die Floppy 1571 konvertiert Commodore-Texte ins PC-DOS-Format (und umgekehrt): ohne RS-232-Kabel und Terminal-Software! **36**

Floppy auf Trab gebracht!
»Burstmon 4.0«: Die völlig neu überarbeitete Version des Super-Diskmonitors für den C128 liest und manipuliert Disketten im Commodore-GCR-, CP/M- und PC-DOS-Format. **40**

Tips & Tricks

So machen's Profis!
Zweifeln Sie daran, daß man die Effektivität von Mastertext, der meist verwendeten Textverarbeitung des C128, noch steigern kann? Unsere Profi-Tips überzeugen Sie vom Gegenteil! **43**

Schnell und seriell
Die Textverarbeitung Superscript 128 konnte man bislang nur mit einem Userport-Parallelkabel nutzen. Wir zeigen, wie serielle Drucker willig mitspielen und haben eine übersichtliche Tastatur-Referenz der Programmfunktionen für Sie parat. **44**

Jetzt wird's bunt!
»LC-10 Color«: Trotz Farbdrucker nur schwarzweiße Ergebnisse? Mit unserem Utility läßt sich der Star LC-10 Color (und andere kompatible Geräte) vor dem Ausdruck auf Farbe einstimmen! **45**

POKE, PEEK & Co.
Nutzen Sie die Systemroutinen des C128! Unsere Liste bringt jede Menge nützlicher POKE- und SYS-Anweisungen **45**

Aufbruchstimmung
Planen Sie, in nächster Zeit in höhere Computer-Regionen aufzusteigen? Eine optimale Alternative zum MS-DOS-PC/AT ist zweifellos die Archimedes-RISC-OS-Rechnergeneration von Acorn. **46**

Bytes, auf die's ankommt
»Patch Music«: Darauf haben viele gewartet: Die Anpassung von Music Master 7.70 (128er-Sonderheft 76) an seriell angeschlossene Nadeldrucker ist da! **47**

Cursor per Maus
»Mauscursor«: Nie ist der Cursor da, wo man ihn gerade braucht (vor allem bei Dateneingaben in Bildschirmmasken). Mit unserem Utility folgt er jeder Maus- oder Joystick-Bewegung wie ein Schatten! **48**

Farbenpracht 128
»Moving Colors 80«: Was sind schon die 16 Standardfarben des C128? Unser Grafik-Utility zaubert 136 verschiedene Farben flimmerfrei auf den 80-Zeichen-Bildschirm des VDC. **48**

Start frei für die Micro-Floppy!
»Autoboot 128«: Der Autoboot-Maker von Commodores Test-/Demo-Disk, raffiniert aufgepeppt: Jetzt funktioniert er auch mit den 3½-Zoll-Disketten der Floppy 1581! **49**

Labels mit High-Speed
»DiskEtti 128«: Noch schneller, übersichtlicher und komfortabler! Das Disketten-Etikettier-Tool im 128er-Sonderheft 76 wird jetzt höchsten Profi-Ansprüchen gerecht. **49**

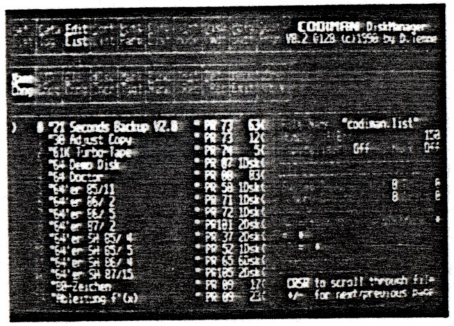
Disk-Hilfen
Sie sind nicht jedermanns Sache, die komplizierten DOS-Anweisungen des Basic 7.0 (formatieren, Header oder ID ändern usw.). Bei unserem Tool erledigt alles ein Knopfdruck! **50**

Sonstiges

Diskettenseiten **18**

Impressum **20**

Vorschau 64'er-Sonderheft 90 **50**



Diskettenverwaltung der Extraklasse:
CODIMAN 128: erfaßt Ihre Diskothek und registriert maximal 2000 Einträge pro Datei **Seite 4**

Der Mailboxdienst GEnie

Egbert Schröer

Auf der Suche nach neuer Software unter CP/M stößt man recht schnell auf den Mailboxdienst GEnie, ein ähnlicher kommerzieller Anbieter wie Compuserve. Wenn man allerdings Mailboxdienst sagt, ist dies sehr untertrieben, denn GEnie bietet weit mehr. Insbesondere die Special Interest Group ist für uns Interessant, oder möchte sich etwa keiner der Z3PLUS Besitzer online mit Jay Sage oder anderen Größen auf diesem Gebiet unterhalten ? (Schwer vorstellbar). Die Sache hat nur einen Haken: der Zugang zu GEnie ist nur für Besitzer eines Datex-P Anschlusses über NUA: R 45221040880 - s.a. letztes Club Info Protokoll einer Mailboxsession - oder für Nutzer des Genios Hosts für Wirtschaftsdatenbanken zugänglich. Bei Zugang über Datex-P muß man zusätzlich im Besitz einer Kreditkarte sein. Genios ist über Telefonleitung zugänglich und schaltet dann zu GEnie (Menü-Punkt 06) via Gateway durch.
Was alles GEnie bietet und wie man mit dem Host arbeitet wird -so hoffe ich- in diesem Club-Info veröffentlicht:

1. Auszug aus einem Artikel über prof. Mailboxanbieter in PC-Professionell (etwas zu sehr pro CIS eingefärbt, CIS bietet für CP/M nichts und in anderen Bereichen nicht alles was GEnie bietet; ist auch teurer)

2.Protokoll einer Session in der Special Interest Group (seht euch mal die Veranstaltungen für den 18.04.93 an !!!)

3.CPMHELP-Text (File 7300)

4. Auszug aus der File Liste CP/M Software (Public Domain, Download Zeit wird als Online Zeit mit 18$/Stunde berechnet)

5. Das GEnie Handbuch (Das Dowloading kostet Zeit und Geld -Bestellung für 29$+shipping online möglich- deshalb, so hoffe ich, der Abdruck im Sonder-INFO)

See you on GEnie

Egbert

Das oben, unter Punkt 5 erwähnte Handbuch (ca. 80 Seiten in englisch) kann von der Redaktion per Postkarte oder FAX 07971/25055 angefordert werden.
Gleichzeitig möchte ich den Egbert bitten, das dazu sicher vorhandene Text-File in die CLUB80.DISC einzulagern. Vielleicht sind auch schon weitergehende Erfahrungen vorhanden, die dann dort besprochen werden können.
Die Redaktion.

GENIE

Der Mailboxdienst Genie weist viele Parallelen zu CIS auf: Er wird ebenfalls von einem amerikanischen Unternehmen, der General Electric Company, betrieben. Den Genie-Dienst mit weltweit circa 400 000 Teilnehmern wickelt die Tochterfirma GE Information Services ab. In Deutschland existiert eine Filiale in Hürth-Efferen bei Köln.

Genau wie Compuserve versteht sich Genie als Breitbandinformationssystem, das Informationen beliebiger Art für eine weitgefaßte Zielgruppe bereithält, so daß Hobbyanwender ebenso wie geschäftliche Nutzer angesprochen werden. Auch ist die Box strukturiert, über mehrere, hierarchische Menüs findet man die gewünschten Daten.

Damit eine Direktanwahl schnell vonstatten geht, werden sämtliche Genie-Infos mit Seitennummern versehen. Durch die Angabe eines Move-Befehls und der Seitennummer, zum Beispiel *move 200* für das Mailmenü, springt das Infosystem direkt zu der Seite. An jedem Prompt darf der Benutzer auch eine begrenzte Anzahl von Kommandos oder Keywords eingeben. Durch die direkte Angabe eines Keywords wie *pcsource* oder *hayes* ruft der Anwender das mit diesem Keyword verbundene *Round Table*, das schwarze Brett, auf.

Weitgehende Ähnlichkeit mit Compuserve besteht auch bei der Boxstruktur von Genie: Schwarze Bretter beziehungsweise Foren im CIS-Slang oder Round Tables (RT) in der Genie-Terminologie sind in die drei Rubriken *Text*, *Dateien* und *Chat* gegliedert. Neben den reinen schwarzen Brettern existieren Datenbankverbindungen und Reisedienste, die es auch bei CIS gibt (OAG, Eassy Sabre). Sehr illuster sind die Support Round Tables. Hier unterstützen, wie bei CIS, große, amerikanische EDV-Firmen wie Microsoft, Borland, Hayes, Wordperfect, Geoworks etc. ihre Kunden über den Draht. In einer Genie-Mall preisen die meist amerikanischen Anbieter ihre Produkte an.

Auf den zweiten Blick allerdings hat Genie gegenüber CIS einige Defizite: So befindet sich lediglich ein deutschsprachiges Brett im gesamten System. Elektronische Gateways zu anderen Mailboxen und Netzen gibt es nicht, lediglich Telex, Fax und Briefpost werden angesprochen.

Sehr beschränkt sind auch die Zugangswege zu Genie, nur über Datex-P mit maximal 2400 Bit/s kommt Genie zum Anwender nach Hause, Highspeed-Modem-Zugänge, ISDN oder Knotenrechner stehen nicht zur Verfügung.

Außerdem ist für Genie-Anwender der Besitz einer Kreditkarte Pflicht, da eine andere Zahlungsmethode nicht vorgesehen ist. Nur Benutzer, die mit dem zu Genie verwandten Genios-Datenbankservice arbeiten, kommen ohne aus. Die übersichtliche Preisstruktur von Genie ist dagegen ein Pluspunkt: 27 Dollar effektive Kosten inklusive Kommunikationszuschlag, Reverse-Charging und Onlinegebühr im Normaltarif sowie 18 Dollar während des nächtlichen und wochenendlichen Niedrigtarifs verlangen dem Benutzer keine großen Rechenleistungen ab. Zu diesen Gebühren kommen noch die Kommunikationsaufwendungen für die Modemtelefonverbindung zum nächsten Datex-P-PAD.

Unter dem Strich sieht die Rechnung für Genie jedoch nicht so gut aus: Bei Niedrigtarif und einem PAD, der nur über die Weitzone zum Telefontarif von 0,23 Mark für 42 Sekunden zu erreichen ist, betragen die Kosten pro Stunde circa 50 Mark. Nutzt ein CIS-Anwender den gleichfalls in der Weitzone befindlichen CIS-Einwählknoten mit 9600 Bit/s, addieren sich sämtliche Kosten (Onlinegebühr 22,50 Dollar, Kommunikationszuschlag 2,50 Dollar, Telefonkosten Zone 3, eine Stunde circa 20 Mark) auf ungefähr 60 Mark. Allerdings wird dabei mit CIS während der gleichen Zeit die vierfache Menge an Daten transferiert. In der Rechnung wurden die als Gemeinkosten anzusehenden monatlichen Grundkosten und die einmaligen Aufwendungen für CIS (Starterkit) nicht berücksichtigt.

Teuer wird es für Genie-Kunden auch, wenn sie sich häufig an die telefonische Hotline von GE Information Services wenden. Anders als bei CIS kann man nicht kostenlos anrufen, die GE-Supporter sind jedoch bereit, den Anfragenden zurückzurufen. Das GE-Büro ist allerdings nur während der üblichen Bürozeiten besetzt und nicht wie bei den Compuservern von 9.00 bis 20.00 Uhr.

Einen Onlinesupport anderer Art erfahren Genie-Benutzer durch *Aladdin.* Die Software ist das GE-Pendant zu Compuserve Information Manager. Doch trotz guter Funktionalität und kontextsensitiver Hilfe ist Aladdin nicht so leistungsfähig wie CIM. Viele in CIM implementierte Features fehlen Aladdin, außerdem weisen weder Genie noch Aladdin ein dem HMI-Protokoll vergleichbares Verfahren auf.

FAZIT: GENIE

Es ist deutlich, daß Genie in einigen Punkten vom Modell einer idealen Mailbox abweicht. Auch beim Vergleich mit dem amerikanischen Konkurrenten Compuserve werden Defizite sichtbar, zum Beispiel bei den Zugängen: kein Highspeed-Modem, keine Einwählpunkte, kein ISDN. Darüber hinaus sind die Infoangebote auf amerikanische Erfordernisse abgestimmt, es sind keine deutschen oder europäischen Round Tables oder Foren vorhanden, ebenso keine Net-Gateways. Allerdings planen GE Information Services nach eigenen Angaben, deutsche Round Tables einzurichten. Künftig sollen auch E-Mail-Gateways für andere Netze zur Verfügung stehen, die noch in der Testphase sind.

Die Special Interest Group in der GEnie Computing Area
Seite: -1-

```
Entering the Computing on GEnie area.

GEnie                     COMPUTING                      Page 510
                      Computing on GEnie

 1.[*]GEnie Computing News (930418)     2.  GEnieLamp Newsletter
 3.  Apple/Macintosh RoundTables        4.  Atari RoundTables
 5.  Commodore & Amiga RoundTables      6.  IBM PC/Tandy RoundTables
 7.  Aladdin Support RoundTables        8.  Product Support RoundTables
 9.  Special Interest Group RTs        10.  Computing Conference Center
11.  PC Catalog                        12.  Computer NewsCenter

Enter #, <P>revious, or <H>elp?
   Welcome to COMPUTING ON GENIE. This page, which serves as the gateway to
the Computing RoundTables, is also your source of daily information about
events and occurrences inside the Computing RoundTables.

   Regardless of the computer you may be using, there's always something
happening in the Computing RoundTables that concerns you and will be of
interest to you.

Today in COMPUTING ON GENIE:  Sunday, April 18, 1993
---------------------------------------------------------------------------
(All times listed are Eastern Time Zone)

Time:   Keyword   Page   Subject
---------------------------------------------------------------------------

12:00n:  A2          (645): Apple II Kids Real Time Conference
 3:00pm: AMIGA       (555): 5-Minute News Network-Conference Room 10
                        >>  Capture to read later; available until 3:00am
 6:00pm: LAPTOPS     (655): Laptops Real Time Conference
                        >>  Hardware & Applications
 6:00pm: CHAT        (400): Macintosh HELP DESK-Channel 3
  till 8:00             >>  You can seek answers to your Mac questions at the
                            Help Desk. The Help Desk will be a resource for
                            RoundTable and GEnie members alike; get answers in

                            real-time, from an experienced Macintosh user, to
                            your hardware, software or general GEnie and Mac
                            RT questions. Poelo Mileto, coordinator
 6:00pm: AMIGA       (555): Amiga Hardware Clinic
 7:15pm: WP          (521): WordPerfect & DrawPerfect Real Time Conference
                        >>  Graphics
 8:00pm: BORLAND     (765): Borland Real Time Conference
 8:00pm: CHAT        (400): Unix & Internet Help Desk-channel 4
 8:30pm: A2PRO       (530)  World-Famous A2Pro Real Time Conference
 8:30pm: FORTH       (710): FORTH Figgy Bar Tutorial Real Time Conference
 8:30pm: GEOWORKS   (1050): GEOS Developers Real Time Conference
 9:00pm: UNIX        (160): UNIX Tutorial (Real Time Conference)-Room 1
 9:00pm: AMIGA       (555): Amiga Help Desk-Conference Room 4
 Ngm: ST          (475): Atari ST Help Desk
 9:00pm: MAC         (605): Mac Sunday Nights Fight Real Time Conference-Room 3
```

9:00pm: TANDY (635): Tandy Classical Real Time Conference
>> Regular "FreYK\K5All" Where Anything CxCppen
9:30pm: A2 (645): Apple II][Speak Real Time Conference
>> Host: Don Arrowsmith
9:30pm: IBMPC (615): IBM General Meeting Real Time Conference
10:00pm: PORTFOLIO (950): Portfolio Real Time Conference
10:00pm: AMIGA (555): Graphics Real Time Conference
10:00pm: DTP (590): Desktop Publishing Weekly Real Time Conference
10:00pm: CBM (625): CBM Weekly Real Time Conference-Room 2-Host T.ADAMS18
10:00pm: MAC (605): Macintosh HELP DESK Real Time Conference-Room 1
till 12:00am >> You can seek answers to your Mac questions at the Help Desk. If you didn't have a chance to call earlier or you have more questions, this is the place! All GEnie and Mac questions are welcome.
11:00pm: CBM (625): Commodore 64/128 Help Desk-Room 3, host YLR.ROSE

* * * * * * * * * * * * *

Tonight in COMPUTING ON GENIE: Monday, April 19, 1993

(All times listed are Eastern Time Zone)

Time: Keyword Page Subject

6:00pm: MAC (605): GE-MUG FlashNews Express Desk
till 10:00pm >> The most current Apple press releases will be broadcast rotationally in the GE-MUG Real Time Conference Room #4. Set your macro to get this 10 minute press update and find out what your favorite computer company is saying.
8:00pm: CHAT (400): Macintosh HELP DESK Real Time Conference-Channel 3
till 10:00 >> You can seek answers to your Mac questions at the Help Desk. The Help Desk will be a resource for RoundTable and GEnie members alike; get answers in real-time, from an experienced Macintosh user, to your hardware, software or general GEnie and Mac RT questions.
8:30pm: ENABLE (465): Enable Applications Real Time Conference
9:00pm: AMIGA (555): Amiga Real Time Conference-Help Desk
9:00pm: TI (575): Texas Instruments Real Time Conference
9:30pm: A2 (645): Apple !!; New Users Real Time Conference-Room 1
>> Host: Dave Ciotti
9:45pm: MAC (605): Educational Mac Real Time Conference-Room2
The Macintosh as an educational computer; how does it compare with the alternative (Apple II, IBM) in terms of hardware appropriateness, software vailability and quality, and interfaces for special education. Hosted by Robert Whitelock, a teacher currently working as a Computer Education Resource Specialist for the Bruce County Board of Education, in Chesley, Ontario.
10:00pm: PCALADDIN PCAladdin Help Desk/Real Time Conference
till 12:00am (110): If you have any questions, this is the place...
10:00pm: UNIX (160): NeXT Night with Eric Tremblay-Room 1

10:00pm: CBM (625): Commodore Real Time Conference-Room 5
till 12:00am Host: F.Ogle
10:00pm: AMIGA (555): DeskTop Publishing Amiga Style!
10:00pm: ATARI (475): Atari ST Desktop Publishing Real Time Conference
special guest Nathan Potechin of DMC Publishing visits to discuss Outline Art 3.0-the Color Vector
giveaway Graphics program. Someone will win Outline Art 3.0 or the equivalent value in Calamus fonts
10:30pm: MACPRO (480): MacPro Real Time Conference-Room 1
11:00pm: MAC (605): Macintosh HELP DESK Real Time Conference-Room 1
till 1:00am >> You can seek answers to your Mac questions at the Help Desk. If you didn't have a chance to call earlier or you have more questions, this is the place! All GEnie and Mac questions are welcome.

* * * * * * * * * * * * * *\

Tonight in COMPUTING ON GENIE: Tuesday, April 20, 1993

(All times listed are Eastern Time Zone)

Time: Keyword Page Subject

8:30pm: ENABLE (465): ENABLE Weekly Real Time Conference
9:00pm: AMIGA (555): Amiga Help Desk-Conference Room 4
8:00pm: CHAT (400): Macintosh HELP DESK-Channel 3
till 10:00 >> You can seek answers to your Mac questions at the Help Desk. The Help Desk will be a resource for RoundTable and GEnie members alike; get answers in real-time, from an experienced Macintosh user, to your hardware, software or general GEnie and Mac RT questions.
9:30pm: A2 (645): A2 Real Time Conference- Host: Lynda Botez
9:30pm: LAPTOPS (655): Laptops Real Time Conference
9:30pm: DTP (590): Design to Print Real Time Conference-Helpline
9:30pm: IBMPC (615): IBM TeleCommunications Real Time Conference
9:45pm: MAC (605): HyperCard Real Time Conference-Room 3
>> Learn: HyperCard; how to build stacks; the secrets of HyperTalk, and a lot more... Feel free to bring your "basic" or "advanced" questions to our own, in-house, HyperGurus: Chris Innanen and Remo Barter.
10:00pm: BBS (610): BBS TalkAbout-WWIV (Real Time Conference)
10:00pm: MAINFRAME (1145): Mainframe Real Time Conference
special guest Tony Wetmore of System Development on the GEnie Bulletin Board system
10:00pm: UNIX (160): Unix Real Time Conference-room 1
to 11:30pm >> Weekly tutorials for learning the INTERNET
10:30pm: MAC (605): About Powerbooks Real Time Conference-Room 2
>> Shopping for a PowerBook? Want to know more about PowerBooks? Stop by the RTC and find answers to your questions.
10:00pm: AMIGA (555): Mikey's Mad Music Night Real Time Conference
10:00pm: CBM (625): GEOS and Wizard Real Time Conference-Room 4

till 12:00am Host: CMD-Doug
10:00pm: GADGETS (690): Gadgets by Small Weekly Real Time Conference
11:00pm: MAC (605): Macintosh HELP DESK Real Time Conference-Room 1
till 1:00am >> You can seek answers to your Mac questions at the Help Desk. If you didn't have a chance to call earlier or you have more questions, this is the place! All GEnie and Mac questions are welcome.
11:00pm: TANDY (635): Color Computer, OS-9, and OSK Real Time Conference

* * * * * * * * * * * * * * *

Tonight in COMPUTING ON GENIE: Wednesday, April 21, 1993

(All times listed are Eastern Time Zone)

Time: Keyword Page Subject

6:30pm: A2PRO (530): Resources Real Time Conference with Marc Wolfgram
7:00pm: SOFTRONICS (630): Softronics Real Time Conference
8:00pm: CHAT (400): Macintosh HELP DESK-Channel 3
till 10:00 >> You can seek answers to your Mac questions at the Help Desk. The Help Desk will be a resource for RoundTable and GEnie members alike; get answers in real-time, from an experienced Macintosh user, to your hardware, software or general GEnie and Mac RT questions.
9:00pm: DBMS (485): Database RoundTable Regular Meeting-room 1
9:00pm: TANDY (635): TRS-80 Model 1, 3, & 4 Family Real Time Conference
9:00pm: CHIPSOFT (170): Chipsoft Real Time Conference
9:00pm: AMIGA (555): Amiga Help Desk-Conference Room 4
9:30pm: A2 (645): Apple II Real Time Conference
>> Hypermedia with host HangTime
9:30pm: IBMPC (615): IBM Operating Systems Real Time Conference
9:30pm: AMIALADDIN (1055): Ami-Aladdin Real Time Conference
10:00pm: PCALADDIN (110): PC Aladdin Weekly Real Time Conference
till 12:00am Do you have any questions about this wonderful tool? This is the place to ask them.
10:00pm: CBM (625): Commodore Flagship Real Time Conference-Room 2
>> Host- T.Adams18
10:00pm: GEOWORKS (1050): GeoWorks Real Time Conference
10:00pm: AMIGA (555): AmiGames with Zach! Real Time Conference
10:00pm: PRO/AM (670): Amiga Programmers Weekly Real Time Conference
10:00pm: CP/M (685): CP/M General Real Time Conference
10:00pm: ST (475): Atari ST General Real Time Conference
Purple Mountain Computers (PMC)-Oscar, Darek, Don
10:30pm: MAC (605): Ma,K=Mh Games Real Time Conference-Room 3
>> This is the place to find out the latest scoop on the newest and hottest games! If you like: Arcade, Action/Adventure, Role-Playing, Strategy Sports, Simulations, Children's Games, etc., this conference is for you! Join host Gann Matsuda (G.Matsuda) for the week's latest!
11:00pm: CBM (625): Commodore Flagship Real Time Conference-Room 3
>> Host-YLR.ROSE

11:00pm: MAC till 1:00am (605): Macintosh HELP DESK Real Time Conference-Room 1
>> You can seek answers to your Mac questions at the Help Desk. If you didn't have a chance to call earlier or you have more questions, this is the place! All GEnie and Mac questions are welcome.

* * * * * * * * * * * *

Other items of interest: Keyword:

TONY WETMORE RTC on GEnie Bulletin Board software 4/20........MAINFRAME
MS-DOS Emulator for TMS 9900 Processors.......................TI
SAVE hard disk space and WIN free software in.................MAC-PS
Nathan Potechin presents Outline Art 3.0......................ST
Uncrackable encryption using RIPEM. To get it, visit.........VIRUS

* * * * * * * * * * * *

GEnie COMPUTING Page 510
Computing on GEnie

1.[*]GEnie Computing News (930418)	2. GEnieLamp Newsletter
3. Apple/Macintosh RoundTables	4. Atari RoundTables
5. Commodore & Amiga RoundTables	6. IBM PC/Tandy RoundTables
7. Aladdin Support RoundTables	8. Product Support RoundTables
9. Special Interest Group RTs	10. Computing Conference Center
11. PC Catalog	12. Computer NewsCenter

Enter #, <P>revious, or <H>elp?

GEnie SIGS Page 516
Special Interest Group RoundTables

1. BBS RoundTable	2. Borland RoundTable
3. CP/M RoundTable	4. Forth RoundTable
5. Laptops RoundTable	6. Microsoft RoundTable
7. Mini-Mainframe RoundTable	8. PostScript RoundTable
9. TI & Orphans RoundTable	10. Unix RoundTable

Enter #, <P>revious, or <H>elp?

Entering the CP/M RoundTable area.

=======================================

Welcome to the

CP/M RoundTable!

Supporting all varieties of CP/M, ZCPR3
ZSDOS and Z3PLUS

=======================================

File/B.B. Help: File #7300 CPMHELP.TXT
General help: Send GEmail to CPM$

Welcome to the CP/M RoundTable!
****

For IMPORTANT information please use selection #4, About the RoundTable from the main CP/M RoundTable menu at Page 685.

****
W E L C O M E A B O A R D !
****

Chances are this is your first time here so jump right in and feel out your new message system. Although you'll find that the menus are pretty straight forward, let me offer FOUR pointers for using this system:

1) If you haven't recieved your manual, type MANUAL at the next GEnie prompt. You may either DOWNLOAD the manual or ORDER a manual for delivery. There are HELP FILES on most all the menus. If a menu doesn't seem to have a HELP option, try typing HELP anyway.

2) The real brawn of the CP/M RoundTable Bulletin Board is in the finely-tuned Command Structure we use. Commands may be selected either by entering the corresponding number (found on the the Bulletin Board Full Prompt menu) - OR - by typing the actual command. Only the first three letters are actually required. You can mix any combination of command words and numbers. You'll want to become familiar with the BB Command Mode if you plan to actively participate in the discussions.

3) To get here DIRECTLY when you logon to GEnie, add the following to the end of your GEnie U#: ';685' (without the quotes). You can add a menu command to it if you like -> 'XYZ#####,XXXXX;685,1' will automatically put you into the CP/M RoundTable Bulletin Board every time you logon.

4) IF YOU HAVE A PROBLEM -> At any menu page type 'PO' (which stands for POrt#. Hopefully it won't be used enough to win the obvious nickname!). Note the Port and MRC number GEnie displays - and the time - then type FEEDBACK at any menu page to report your problem. Depending on the nature of your problem, you may want to tell us the type of modem and software you are using.

You'll only see this bulletin just this one time, so if you have any problems at all, don't by shy! ASK!

ABOUT THE CP/M ROUNDTABLE...

(Use Control-S to Pause text/Control-Q to Continue the text & your BREAK Key to leave this listing.)
================================

Your CP/M RoundTable Sysops are:

All CP/M:	Keith Petersen	(W8SDZ)
All CP/M:	Brian Moore	(BRIAN-CPM)
C128 CP/M:	Peter Jacobson	(PETER-CPM)
Communications:	David Goodenough	(DAVID-CPM)
Z-System/ZCPR:	Jay Sage	(JAY.SAGE)

RoundTable Conference Schedule (RTC)
====================================
WED 10:00pm EST CP/M Help Desk
SAT 10:00pm EST CP/M Help Desk

First Wednesday of Every Month 10pm EST:

Z-System RTC with Jay Sage

Frequently Requested Files and Locations
==
New CP/M RoundTable Help File:
File #7300 CPMHELP.TXT .

What is the CP/M RT? It's thousands of CP/M users just like you, helping each other with questions and answers in the CP/M RT Bulletin Board. It's thousands of CP/M enthusiasts sharing public domain software with each other, uploading and downloading files from the huge CP/M RT Software Libraries.

If you are new to the world of telecommunications, you may feel that you're on unfamiliar ground. What follows are answers to the most common questions we're asked:

Q: HOW DO I DOWNLOAD A PROGRAM, AND WHAT DO I HAVE TO DO TO USE THE FILE?

A: First, make sure your terminal software is capable of supporting binary file transfers. QTERM, MEX, and IMP are the most popular terminal programs which are binary file capable. Next, check your terminal settings. They should be 8 bits, no parity, 1 stop bit, and local echo (unless you have selected node echo at login). Now, pick a file, any file. You can LIST out the contents of the SL's (Software Libraries); you can BROWSE through the SL's file by file; you can SEARCH for a file using keywords, name of uploader, specifying the age of the file by days. If you're using the BROWSE option, you will be prompted after each file description to either download, list, or skip the file. When you see READY TO SEND, tell your terminal software to 'receive a file using XMODEM', or if your communications program supports YMODEM the file will name itself on your disk.

Q: IS THERE ANYTHING MORE I HAVE TO DO TO USE THE DOWNLOADED SOFTWARE?

A: That depends. If the file's name ends in a '.LBR' or '.ARK' suffix, it's

what is known as a Library file or Archive file. Files stored in this format save you time and money - but they require an extra step before you can use them. Please download file #7300, CPMHELP.TXT, for full details.

Some files may need a certain language interpreter to use them. For instance, a few files are in Microsoft Basic source code. You'd need the Microsoft Basic (MBASIC) program to run these. Whenever a program needs another program in order to run, it'll say so in the file's description so PLEASE pay special attention to the information provided on a file in the long file descriptions.

Q: HOW DO I UPLOAD SOFTWARE TO CP/M RT?

A: If you can download properly, your settings are already correct for uploading (to send software) to the CP/M RT Software Library. Give it a try - the GEnie prompts will walk you through the procedure and remember, uploads to GEnie are FREE during non-prime time hours!!!!

Q: HOW CAN I GET THE MOST OUT OF THE CP/M RT BULLETIN BOARD?

A: That depends on what you WANT out of the Bulletin Board. If you need to ask for help or advice on a particular subject, first type CAT to get a list of the available BB Categories. Next, type SET categoryNumber to get to that category. Now, type TOPics (only the first three letters are needed) to see if your topic has already been covered. If so, see what's already been said using the REAd topicNumber command. If you need to start a new topic, type STArt. Write down the categoryNumber and topicNumber so you can get back to it in a hurry next time you're on. That's it!. Remember, ALWAYS SET TO THE CORRECT CATEGORY BEFORE STARTING A TOPIC!! YOU'LL GET FASTER RESULTS, AND THE CP/M SYSOPS' LIFE WILL BE EASIER!!

Q: WHAT'S AN RTC?

A: A RTC is like a live meeting of all the users who want to attend, and someone from the CP/M RoundTable will be there to 'chair' the meeting. Real Time Conferences are scheduled on Wednesdays and Saturdays. Check the CP/M RT Front Door for times and special details.

Q: I HAVE A PROBLEM. WHO DO I ASK FOR HELP?

A: First, what kind of problem? If it's timeout errors in file transfers, lost carrier signal, or erratic characters are appearing on your screen, type PORT at any GEnie menu, logoff, and report it to GEnie Client Services. A downloaded file won't work? Check your steps, and if the problem still has you stumped, send a GE Mail message to CPM$.

Hope this helps. If this file doesn't cover enough of the bases, just holler via E-Mail!!

- Your CP/M SYSOPS - CPM$ (GEmail)

CP/M RoundTable

1. CP/M Bulletin Board
2. CP/M Real-Time Conference
3. CP/M RT Libraries
4. About the RoundTable
5. RoundTable News (911122)
6. Take a CP/M RT Survey

Enter #, <P>revious, or <H>elp?

(Use Control-S to Pause text/Control-Q to Continue the text & your BREAK Key to leave this listing.)

================================

=-> New CP/M Help File <-=
File #7300

We have updated our CP/M RoundTable Help File to include the many new features now available in our RoundTable. File #7300 CPMHELP.TXT is yours for the download!

=-> New CP/M Menu Option <-=

The Main Menu on page 685 has a new option. Called The PIPMAG, it is option #7 and will bring you to page 688, The CP/M PIPMAG. On this submenu, you can select any of the current PIPMAG articles for buffering or on-line reading. You

may also select to download the latest issue in the CP/M Libraries on page 686. You will also find our NEW and extensive CP/M HELP FILES on page 688, which are menu options #12 (Libraries and Files), #13 (Bulletin Board), and #14 (Getting Help, General).

-=> Expanded Canadian Service offered through iNET 2000(TM) is now available in 166 Canadian Cities. For further information on iNET 2000 call their Customer Assistance Center (in Canada) toll free 1-800-267-8480, or (in U.S.) 613-781-5128. GEnie service will still be offered through the four local nodes located in Montreal, Totonto, Calgary and Vancouver.

Remember - Your userid and password are like the keys to your home. You should guard them carefully, NEVER give your userid and password to another user online and be sure to change your password frequently.

Genie will never request your userid or password from you after you have completed your logon. If it appears that your userid and/or password are being requested, logoff and start a new session from scratch.

If you have not changed your password in the last two months, at the next menu prompt please type M900;2<RETURN> and change it NOW!

Just a reminder that upload time is still FREE on GEnie. Every minute from the time you hit the "5" until you are done with the upload is credited against your account. There's not enough time in the day for the SYSOP's to call around the country looking for new software. We must also depend upon you, the users. So hey guys and gals, check your local system for new stuff and do your fellow CP/M'ers a favor by uploading it on GEnie!!

PIPMAG Issue #7 is now available for downloading!! See files numbers 7663 (PIPMAG7.ARK) for the ARK version --OR-- file number 7662 (PIPMAG7.ZIP) for the ZIP version!!

Ya wanted it both ways, so here's your request..... we listen!!!

In response to numerous suggestions, GEnie will be changing the entry method to the Software Libraries. Currently, the software libraries exist on separate mainframes within the GEnie processing environment. The menus which are displayed on the Software Library and Software Library Set pages exist on one set of mainframes, while the actual processing of a user's selected function is handled by another set of mainframes. GEnie will be consolidating the Software Libraries into a "single unit" on April 8, 1991.

Users should notice faster response when selecting a Software Library function. The "please wait..." message will be eliminated. The Software Library menus will now be consistent across all of GEnie.

PLEASE NOTE that Script users will have to make appropriate modifications to existing scripts.

The following user visible differences should also be noted:

"Software Library" and "Software Library Set" Page numbers are being eliminated. For example, you will no longer be able to move directly to page 686 (our Software Library Page). Instead, users should select the Software Library menu selection from the "main RoundTable page". For example, for the CPM Roundtable Software Library, issue the MOVE 685;3 command.

The indication of which library is currently set, which appears as the "Library:...." line at the top of the main roundtable page, is being removed. This information will only appear in the Software Library section.

Wizard users should see no changes when they use their auto-passes, but must be careful to take into account the new changes when they log on manually.

Users should also be aware that only the MOVE, EXIT, and QUIT commands, plus valid keywords, will be accepted within a Software Library.

If you have any questions, please post them in the GENIEus Bulletin Board (Page 150 or *GENIEUS), Category 4, Topic 17 or the CPM Bulletin Board.

```
GEnie                        CPM                        Page 685
                        CP/M RoundTable

 1.  CP/M Bulletin Board
 2.  CP/M Real-Time Conference
 3.  CP/M RT Libraries
 4.  About the RoundTable
 5.  RoundTable News (911122)
 6.  Take a CP/M RT Survey

Enter #, <P>revious, or <H>elp?
```

In this area, you will find people with shared interests. You may want to start out by reading "About the RoundTable" which contains a description of the RT and may contain helpful tips on how to use the RoundTable (RT).

Each RT has its own "Bulletin Board". This is an area where you can electronically post messages or respond to messages posted by others. It is a perfect place to exchange information, find solutions to problems, or discuss issues and topics of interest.

A RT's "Real Time Conference" (RTC) is an electronic meeting place where members gather to discuss topics of interest. Guest speakers are ocassionally invited to lead the RTC and discuss their area of specialty.

"RoundTable News" displays announcements including news about upcoming RTCs.

The "Software Libraries" contain software files uploaded by other GEnie members for your use. Uploads are free on GEnie at 300, 1200 and 2400 baud.

If you have any further questions about a specific RT, you should send GE Mail to the Sysop.

```
GEnie                        SIGS                       Page 516
              Special Interest Group RoundTables

 1.  BBS RoundTable                   2.  Borland RoundTable
```


CPMHELP.TXT   900320

Introduction:

The CP/M Help File is in the Library to assist your on-line GEnie needs.
If you have any problems, please contact us through the CP/M Bulletin Board, our regular scheduled RoundTable Conferences, or send GEmail to: CPM$ (just type 'S' at any GEnie System Prompt (?) and at the To: enter CPM$). The CPM$ is called a Group Mail address, and is sent to all of your CP/M SYSOPS.

This help file is divided into 3 main sections:
1. Use of the Libraries, where you can upload and download files
2. Use of the Bulletin Board for public messgaes, questions, etc.
3. Real Time Conferences, interactive chatting and discussions.

The Main CP/M RoundTable Menu:

Page 685

```
1. CP/M Bulletin Board         <---- Public Messages
2. CP/M Real-Time Conference  <---- Online discussions, chats
3. CP/M RT Libraries           <---- Files in 38 different libraires
4. About the RoundTable        <---- General Information/Help
5. RoundTable News             <---- Newsworthy items of interest
6. Take a CP/M Survey          <---- Special Items (changes from time to time
7. The CP/M PIPMAG             <---- Our own CP/M Magazine
```

Part I   Libraries
==================

UPLOADING
----

1. Select an appropriate Library to send your file to by choosing #8 from the Roundtable Software Library menu (page 686).

2. Choose #5, Upload a file from the Software Library menu.

3. Fill in the information requested about the file you are uploading.

4. Choose the Uploading protocol you are using. Please use ONLY the XMODEM or XMODEM CRC options.

5. When GEnie says 'Ready for INPUT', start your XMODEM Upload.

6. Say 'Yes' when asked 'Put in Library y/n?'

7. Easy, right???? If you wish to try out uploading, go ahead!

CP/M File Compression

This document is intended to give a brief overview of files stored in the CP/M RoundTable Software libraries. For current information, see Category 4 of the Bulletin Board area. This file is an updated copy.

Many of the files in the Software Libraries have been compressed and/or grouped together in a single file by using one of the utilities available to the public domain that either squeezes, crunches, libraries, or archives files. This has been done to minimize disk storage requirements and also to minimize download time.

File Tails Signify Compression

These processed files are specially named with a filetype and can be identified by the last 3 letters of a filename after the "." that signifies the conversion. These are:

.ARK for files archived (same as MS-DOS "ARC").
.LBR for library files.
.?Q? for squeezed files (middle letter is a Q).
.?Z? for crunched files (middle letter is a Z).

A library is a group of files collected together into one file in such a way that the individual files may be recovered intact. A library file can be identified by the "LBR" as the extent of the file name. LU (Library Utility) is a CP/M utility used to maintain libraries of files. LU does not perform any compression. Because of this, most people will squeeze or crunch files before adding them to a library if they want to save space. If you want to remove the component files (members) from a LBR file, you should have a copy of LU310.COM, NULU152A.COM, or other LBR extractor utility. (LU and NULU are 8080/8085/Z80 compatible.)

ARKS

Recently, popular CP/M Public Domain software files and information files are being distributed using ARCHIVE files. Archive files are similar to library (LBR) files, in that they take a logical group of files and put them together in a single file. The main difference, is that the members of the "ARC" or "ARK" file are automatically compressed. The compression algorithm

chosen is one of three which will produce the smallest file.

Archive files have been available to the MS-DOS and PC-DOS areas, and been made useful in the CP/M environment with the introduction of the "UNARC" program. The current version is 1.6, and is available with extensive documentation, and two executable COM files, a 8080/8085 version and a Z80 version. The Z80 version takes advantage of the expanded Z80 (and equivalent) instruction set for speed and size, and therefore is machine dependent. There is also a modification overlay to adapt the program to non-standard CP/M 2.2 and 3.0 operating systems, such as CP/M68k and CP/M emulators. Programs are available on many machines to process "ARC" files, the Atari ST, systems running the UNIX, systems running MS-DOS, and CP/M. Please note that the CP/M ARK is not compatible with most Native Mode C-128 ARC.

ARK11.ARK (file #7117) is the CP/M utility available to make an "ARK" file. Archive files may also be made in the MS-DOS/PC-DOS or UNIX environment. (ARKs made on other systems may produce different results as the CP/M ARK program -- UNARC16 is able to extract and decompress these files.)
To avoid confusion on RCP/Ms that support both MS-DOS and CP/M callers, Archive files have a file extension of "ARK" for CP/M software and information files.

Details of File Compression

Some files in the CP/M RoundTable Software Libraries have been compressed, using one of the standard public domain utilities, to minimize download time and to save storage space.

USQ120.COM is used to unsqueeze, or expand files that have a "Q" as the middle letter of the filetype. Such files have been squeezed, or compressed with SQ111.COM or similar utility. These programs use Huffman Encoding to reduce the size of the target file. Depending on the distribution of data in a file it can be reduced in size by 30% to 60% by squeezing it. If you download a file with a filetype indicating that it is squeezed, you will need USQ120.COM to expand it before you can use it. There are other programs available, written in different languages and take advantage of special hardware, but USQ120 is 8080/8085/Z80 compatible.

Other utilities are available that have the unsqueeze coding imbedded and function with squeezed or unsqueezed files. There are programs that perform file maintenance functions (NSWP), bi-directional display utilities (BISHOW), and string search programs, (FYNDE and FINDU). This method of compressing files has been used for some time now and programs to uncompress the files are available to several microprocessors and main frame computers.

CRUNCH uses the Lempel-Ziv-Welch (LZW) techniques to compress files. These "crunched" files are identified with a "Z" as the second character of

file type. This method is fast and offers compression ratios around 55%. Highest compression is achieved with graphics data, values of 90% are typical, followed by text, with 50%, and COM files and other binary file of around 20%.
See CRUNCH24.LBR for the Z80 CRUNCH and UNCRunch programs. See FCRNCH11.LBR for the 8080/8085 and V20 CRUNCH and UNCRunch programs.

Since this method of compression is relatively new, there are only a few utilities available that processes crunched files are "typer" utilities such as TYPELZW, TYPEQZ, LT, and QL which also type members of libraries and squeezed files. If running a 8080/8085 processor, check the documentation to determine if the utility will only process files on systems using the Z80 processor. UNCR232.ARC contains a program to uncrunch files in the MS-DOS/PC-DOS environment.

MicroSoft BASIC programs are always saved as ASCII files. (Saved with the ",A" operand.) They may than be squeezed or crunched. This has been done to allow them to be converted for use with other BASIC compilers without the need to convert them with MBASIC first.

Some executable files have a file extension of "OBJ". These are really "COM" files that have been renamed to "OBJ" to avoid execution on RCP/M systems. Rename them back to "COM" for use them as commands on your system.

So where are these Utilities?

Below is a list of utilities in the CP/M software libraries that work with the above mentioned files. For the current version, check the directory or look in Category 1 in the Bulletin Board area for any new information.

FCRNCH11.COM	5995	Uncruncher for 8080/8085 systems
UNARC16.COM	5997	Unarcer for Z80 processors
UNARCA16.COM	5993	Unarcer for 8080/8085 CP/M systems
UNCR24.COM	5998	Uncruncher for Z80 processors
USQ120.COM	5996	Unsqueezer for all CP/M systems
NULU152A.COM	5298	Library maintenance utility

Comming soon.... LZHUF (.?Y?)

Part II Bulletin Board

GEnie's Bulletin Board--Quick Tutorial

So ya wanna get active in our BB area, but don't know how? This file will give you everything you need to know to jump right in.

DELETE CHARACTER

Before you do ANYTHING, make SURE you have your delete character set properly. This will tell GEnie which key your computer uses to delete a character. Since you are using an Amiga, you want to use the Back Space key. You need to set your delete character to 8. So, how do you do this? At any command prompt outside the BB, type 'SET'. Then pick option 3. Terminal Settings. Then ya pick 2. Terminal Settings. This will give you a list of various term options. You want #3. This should be set to 8. After you set it, you want to SAVE it with option 8 on the menu. That's it, your set!

Remember, too, that you are calling a computer which does not accept your cursor keys as editing commands <smile>. There is a line editor provided for errors and mistakes which you can no longer correct with your backspace key. See your GEnie manual.

USING THE Bulletin Board

When you first enter the BB you are set for FULL MENU prompt. That means you get the menu at each prompt. You will quickly get tired of it and want to go into either BRIEF or NONE mode. Easy enough. Just type
PRO BRIEF or PRO NONE. If you decide you want the FULL mode back, type PRO FULL. If you want even more information, type HELP at the B.B. prompt.

Okay, you are here... If you have a special handle or nickname you like to use, now is the time to set it. Just type NAME. It will prompt you for a Nickname. This will be included with your MAIL name on every message you write in the BB.
You can change your Nickname whenever you like in the BB. Just type NAME again. But, all messages written while the old Nickname was set will still have that name in their header.

Now, you have all these billions of messages in the BB that are marked as NEW for you. If you are made of money, or just ENJOY sitting for several days reading old messages you can catch up to the current day by reading everything (By typing BRO ALL).

=========== VERY ==========
=========== IMPORTANT ==========

Most dont want to do this. That's what IGNORE is for. Typing IGNORE ALL at the BB prompt will go through the BB and mark every message as OLD. That way, when you type BRO ALL (more on this later) you will only be shown those messages

written since you last read messages. This only takes a few minutes! Once IGN ALL is completed, you may now use (with ease) the BRO and RAM commands!!!!

==================================

After IGNORE ALL works...

You can still read the old messages whenever you want by using the READ command. It has its good points and its bad points. We have, as of this writing, 17 CATEGORIES of messages. READ only works in one CAT at a time. So... you will to type SET 1, SET 2 etc... after each READ to get to the next CATEGORY. This is only on READ, tho... the BRO command can cover the entire BB. Anyway... You are new, and have typed IGNORE ALL. BUT, you decide you want to read all messages left in, say, the past week. Easy enuf. When you are in the CAT you want (Using the SET command) type:

READ ALL DATE>900210

This will show you every message in the current CAT written on or after 1990, Feb 10. The format for the DATE option is YYMMDD.
So... 900210 means Feb 10, 1990.
Just do this in each CAT and you can read all messages left since Feb 10. The > in the above command denotes GREAT THAN, or AFTER. If you wanted to read all message BEFORE that date you would use <. You can also read all messages left only on that date using = there. Okay, now you are caught up with IGNORE ALL and have read the past weeks messages using READ ALL DATE>900210 in each CATEGORY. Now, whenever you return to the BB all you have to type is BRO ALL and you will be shown ALL new messages in ALL CATEGORIES since you last read messages.

READ also works on TOPIC numbers. Typing REA 5 will show you the entire contents of TOPIC 5. You can also read a certain message in a TOPIC using READ. Say you want to look at message (reply) 24 in TOPIC 5.
You would type:

REA 5 24

You can give it a range of message #'s. REA 5 24-30 will show you messages 24 to 30 in TOPIC 5.

WRITING A MESSAGE IN THE BB

So you've read a message and you would like to reply to it...
Each CATAGORY is divided into message groups called TOPICS. A TOPIC is a thread of messages all covering the same thing. To get a list of all the TOPICS in the CATEGORY you are in, just type TOPIC.

Each topic has a title, and a descriptive header. It also usually has a string of REPLIES. When you get to the bottom of all the REPLIES in a TOPIC you are reading, you will be asked if you want to REPLY. Type Y here if you do. If you dont want to reply, just hit <RETURN>.

Okay, you typed Y and want to leave a message. You will get the prompt:
1>
That means you are on the first line of your message. Start typing whatever you want to say, hitting RETURN at the end of each line. When you are done with the message, and made no mistakes that you want to fix, type *s to save it. You can always type *h for help. See Survival Kit item #10 for a complete list and examples of using the Bulletin Board editor's * commands.

Okay, now you know how to write a message, and fix any mistakes you might make in it. But, suppose you dont see a TOPIC in the CATEGORY that covers what you want to say/ask? No problem. Just start your OWN TOPIC.
At the BB command prompt, or Reply prompt enter "START".

** NOTE ** The sysops and other members appreciate it a lot if you will leave your new topic in the appropriate category! <smile> Use the SET command to set to the category before you STArt a new topic.

Back to STArting now...

You will be asked for a Title. Enter a brief title about what you will be talking about. Then it will ask you for a few lines describing your TOPIC. Enter them as desired, but remember, only the first 3 lines you type here will be shown every time someone reads a reply in that TOPIC.
(If that didnt make sense, dont worry... Just remember to limit yourself to 3 lines here)
Once you save this (using the *s) you will be asked if you want to enter the first message. Type Y if you do, N if you dont.
Generally, you will type Y. After all.. if you STARTed the TOPIC, you will have something you want to say, yes? Sometimes the 3 line Description will be enough for you, maybe you just had a short question. In that case, you wouldn't need to enter the first message, so you would type N when asked.

NOTE At a TOPIC REPly prompt you can read a message in that TOPIC again by typing its number. If you want to read Reply message 17 again, typing 17 at this prompt will do it.

Well... That about covers everything you need to know to get started in GEnie's CP/M Bulleten Board.

READ

Everyone wants to READ, so let me explain a couple of things about the READ command. There are actually *2* read commands, one is CALLED READ and the other is called BROWSE.

BROwse is a shortcut command for READ NEW messages. You can specify BRO CAT and just see the messages in *that* category that are NEW or you can specify BRO ALL and see *all* the new messages on the bulletin board in the categories you are attending. You *can* CANCELL a category and still use BRO ALL without messages in that category being displayed to you until you JOIN the cancelled category again.

READ is a command which works *only* in the discussion topics in the category you are in. There are several parameters of information, some optional which you can specify.

ALL of the new READ command options are listed in the HELP text that you get when you type HELP at the Bulletin Board menu. To see them, just specify HELP READ, buffer it, its about a page of information, which you just might want to print out and put in your GEnie manual.

For the very basic, here is the structure of READ:

```
READ 9 5-8 NOR
 :  :  :  :
 -> :  :  :   --READ command, by itself prompts you for more info.
    -> :  :   --TOPIC # you wish to read *WITHIN* this category.
       -> :   --Message number or range of messages in the topic to read.
          ->  --Options, this is NOReply. IE, its a complete message dump
                which will not pause. Other options include:
          NOT --no text, scans message headers only.
          AUT=BOB
              --lists all messages within given READ parameters from mail
                address BOB, and no others:
READ ALL AUT=BOB will read all messages from me in the category you are
                set to at the time. This works on MAIL addresses only
                and not nicknames.
          DAT>900210
              --lists messages ONLY since the date specified. No spaces
                in the command!!!
READ ALL DAT>900210
              --is the specific command that lets you view all messages
                written in the category you are SET to after that date.
READ 18 DAT>900210
              --reads ONLY messages in topic 18 which were after Feb 10'90
```

To continue reading after a specific date for other categories of interest, just issue the SET command and enter the new category. SET 2, SET 8, etc.

These special read parameters work within a large topic, also for instance:

READ 7 AUT=APPLE.CPM will show you only messages from APPLE.CPM in topic 7 of the current category.

And remember, these things CAN be typed in all in one line or you can let the bulletin board prompt your way thru them, regardless of your prompt mode. I'd seriously suggest that once you are familiar with the looks of the bulletin board, Buffer out a listing of CATEGORIES <these NEVER change>, a listing of the menu <again, its the same>, and have them handy on a print out when you log in!! THEN set to PROmpt NONe!!! <<smile!>>

Judicious use of the IGNORE command can make topics, entire categories or every topic and every category on the Bulletin Board appear as *OLD* messages to you. This will enable you to keep up with just the newest postings with the BROwse command as described above. Set to the category you want to IGNore...remember, this marks all messages there as if you had read them.

SET 2

Then type in:

IGN CAT

You could also just ignore one topic in the CATegory:

IGN 5

Anything you IGNore is not lost forever, it just won't appear when you issue a READ NEW command or a BRO ALL command. You can still read and search through topics which have been ignored with the REA command as described above.

Searching the topics on the Bulletin Board can save you some time, too. There are two ways to do this. The first command, INDEX, will give you a list of EVERY single topic name on the BBoard. Buffer and print out this list.

There is a SEARCH command to help you find items. SEARCH will search through the topic names <only> and match the characters you specify. If you type in SEARCH all by itself, the BBoard will prompt you for all additional information. First, it will want to know what search string you want. This is marked by slash marks at the beginning of the string and at the end:

/MODEM/

And then you'll need to indicate if you want ALL categories searched, or just the one you are in currently.

Once again, SEArch only searches throught TOPIC names, or SUBJECT lines of the topics, but will match your desired string if it appears ANYWHERE within the subject line. For instance, /MODEM/ might bring up topics having to do with XMODEM, YMODEM, MODEMS, HAYES, etc.

There is a lot of powerful commands here-let them work for you!

How to Use the Message Editor

You can change lines, insert, delete, and other commands here while you are leaving a message. The message editor is about the same as the editor in GE-Mail.

First....all commands are preceded with an asterisk. To see a list of all commands, at any line number in your message, as the first character, type: *H

You might want to buffer this list and print it out for future reference, in a handy offline binder!! It'll save you time in the long run!

Now, remember, before trying any command, the asterisk MUST be the first character in the line.

The editor here is a LINE EDITOR. This means you edit only one line at a time, AFTER you enter it. Each line is Entered into the system with a RETURN at the end. It is best to type up to around 80 characters and then press RETURN.

The most often used commands are CHANGE, DELETE and INSERT. Lets take a quick look at them first.

*CHA
====

Here's the command to change a sequence of characters in a line. You can choose to have the line VERIFIED to you automatically by using the command:

*CHAV
=====

The CHA command needs several arguments within the fields of the syntax:
-* The LINE Number you want to edit
-* The OLD STRING you want to correct
-* The NEW String you want entered
-* The number of times you want it corrected <<Usually one>>

The CHA command is followed by a single comma, the line number and then slashes which separate all other fields of the line:

```
*CHAV,39/separaate/separate/1*
: : : :          :       :
: : : :          :       1 Time
: : : :          NEW String
: : : OLD String
: : Line #
: Command <change with line verify on>
ASTERISK
```

*DEL
====

The delete command is very simple....you type in the command, a comma and the line number to delete:

*DEL,50
You can also specify a range of lines:
*DEL,58-60

*I <<Insert>>
==

Insert allows you to add lines between previously typed text lines. You will need to give it one piece of information, the line number of the text you want your new lines to be entered after. Lets say you want to add 3 lines of text after line 5, you'd enter:

*I,5

Then you'd see line 6 come up, and you can enter text until you are done with the information you wanted to add. You are automatically placed in INSERT mode as you enter each new line. Exit INSERT with a RETURN on a blank line, and you'll be placed back where you started in your message.

*L <<List>>
==

You can always LIST your message anytime you'd like, with the *L command. Enter it as the first characters of any new line. Without any additional parameters, it will list the entire messae. You can specify a range of lines, too, tho, like this:

*L,80-86

Well, so much for B.B. Commands. Any other questions about the this, and any-else, please holler!!!

Part III Online Conferences

==============================

Real Time Conferencing
In the CP/M RT (Wednesday Nights at 10pm, eastern room #2)

You can type to other RoundTable members in the Conference area. They will be able to instantly respond to you and you can see what they are saying or doing as it happens. We call this the Real Time Conference. Your messages will be transmitted as soon as you press RETURN, and will appear on the screens of everyone else in that "room". Split Screen terminal programs are best, and a good one is Q-TERM.

Most of the conferences do not require formal protocols, but there are many things you can do to make yourself understood a little better. Follow these quick tips for your first time on Conference. You'll always get a notice when you enter the RoundTable which announces how many people are on the Real Time Conference area. Sometimes there are none, other times we have a regular crowd for specific nights and many times you can make or find impromptu discussions and tutorials with fellow members.

First and foremost, remember that your line is NOT sent until you press RETURN. If you have a split screen or chat window, its very helpful to use that-see your terminal program manual for instructions about that.
If you cannot use a split screen, and an incoming line splits up the line you were typing on, don't worry. GEnie still remembers what you ALREADY typed, you can finish your sentence and press RETURN, and everything will be there, what you typed before the incoming line and what you typed after it!

Many times you will want to type out a complete sentence or thought and need more than one line to do it in. If this happens, just place three periods at the end of the line and press RETURN. This indicates to those who are attending the conference with you that you have MORE to say.
A single period at the end of a line indicates you have finished your thoughts.

Lines can be up to 250 characters long on GEnie's conference. But, be aware that when you type in something very very long that other people might be waiting for you to respond with something. It is many times much more helpful and courteous to other people to go ahead and break up a very very long line like that into two or more lines, adding those three periods and a return

between each part:

<Bob> Hi George, How are you tonight?! By the way, did you see the..
<Bob> incredible new PIPMAG File just uploaded to the Library?!

Abbreviations on Conference

Because people are people, they many times want to say a lot in as small a space of time as they can. Common abbreviations and phrases for people on conferences follow:

Abbreviation	Meaning:
BB	Bulletin Board
BRB	Be Right Back - away from terminal
BTW	By The Way
CU	See You
FYI	For Your Information
G/A	Go Ahead (you're done)
OIC	Oh, I See
OTF	On The Floor - as in hysterics!
PLZ	Please
RSN	"Real Soon Now"
TNX	Thanks!
:)	Smiling
;)	Winking

Private Messages

Everyone has the capability of sending a private message to other people attending the conference. The way you can tell that you have received one is that instead of just a <NAME> in front of the message, you will also see a Job number: like this:

<Bob> This is a real, public message.
<Bob> (Job 1) This is a private message, and only appears on YOUR screen!

^^^^^ See the (Job 1) in parenthesis?! That means that it is private, and came from the username on the line, who is job 1. The Job number is important because that is how you respond privately. To send a reply, all you

have to type is:

/SEN 1 Got the message, Bob!

The NAM command is also important. This functions lets others know what your 'handle' is, instead of your GEMail address. Type /NAM BOB to display your handle, along with your regular GEnie name:

<C.KENT> /NAM Clark

will now display as:

<[Clark] C.KENT>

PROCEDURES AND ETIQUETTE FOR FORMAL CONFERENCES

The RTC CO area (page 685;2) is used for several purposes, including: holding formal and informal conferences, or just chatting with other members. When the meeting is strictly an informal one, there are no special rules, and participants generally talk when they please. However, for more formal meetings, some rules must be established to keep the CO running smoothly and give all a chance to participate. This help file explains the general procedures for running a formal conference in RTC CO area.

First, try to be in the designated room before the announced time of the CO. Almost all of our COs begin at the time specified, and it helps a great deal if people are there from the start. Punctuality results in fewer interruptions, and also fewer repeats of questions already asked.

Second, if someone comes late to the CO, it is impolite for that person to break into the proceedings with a "Hi all, sorry I'm late!". This almost always results in a temporary halt while members exchange greetings, and wastes the time and money of others who are there.

Third, wait for your turn to speak. Typically, COs are held in two parts: the guest speaker (or the CO leader, if there is no guest), makes some opening remarks or comments. During this time, all CO attendees must remain quiet. When the speaker is finished, there is a short break while the CO leader makes a list of questioners. Anyone who has a question to ask, or comment to make, types in: /RAI and that person's name is added to the list. When the list is complete, it is read back, and anyone who was missed the first time will be added. Please observe any limits on the number of questions that may be asked at one time. If there is a large turnout, members may be limited to 1 question and 1 followup per turn, in order to give everyone his or her chance to talk. Members who have more questions can always ask to be put on the list again for the next round.

Fourth, when making your comments (or asking questions), use "..." at the end of each line you type, to indicate there is more to come. When you have finished your question, type in GA (for Go Ahead), so the guest or CO leader knows, and can respond. When your turn is over, type in "G/A" so the next person on the list can be called.

Fifth, it is best to stay in CO until your turn, because if you are called and you are not there, your name will be crossed off the list, and the next person will take your place. If you come back after that, it is up to the CO leader to decide whether or not to add your name back to the bottom of the list.

Sixth, remember that the CO leader, be it a SysOp or someone else, is in charge of the conference. If you have any problems or complaints with the way the CO is being handled, wait until after the conference to straighten things out. Rowdy or disruptive users will be gagged or even locked out of the RTC.

We have held many COs using the above rules, and the results have been very good. The conferences move smoothly, and everyone gets a chance to talk or ask questions. Courtesy and patience on the part of the audience goes a long way to making a good CO. If these guidelines are followed, a conference can be an enjoyable and rewarding experience for all.

Here are some of the more used CO commands, and their explanations:

!! - entered when you have a comment on the current topic

/ECH - echo to sender on; it is helpful to come into our CO's in full duplex, with a SplitScreen or window type terminal program so that your sentences are not mixed in with others on your screen.
/EXI - leave the RTC; returns you to page 685
/MON 1 - Monitor or listen to what is being typed in room 1
/NAM Mark - add your name in RTC
/PRI 3 - go private with JOB 3
/RAI- get attention of leader
/ROO 3 - move to room 3
/SEN 2 Hello! - sends to job 2 a message m of "HELLO"
/STA - see users in room by name and location
/TIM - display time and day
/USE - see number of users
/XEC - exit the echo mode
/XMO - exit the monitor mode
/XPR - exit private mode

Real Time Conferencing is fun and informative! Enjoy!

About the CP/M PIPMAG
=====================

The PIPMAG is our way to bring you information from your CP/M Sysops and our CP/M Members about general and specific CP/M information. You are free to upload this information to your local non-profit Bulletin Board, or distribute it to your favorite User Group. Please follow the Copyright notices.

PIPMAG, The National CP/M RoundTable on GEnie can be read in two different ways:

1. As an ARK file in the CP/M Library
   == OR ==
2. Read On-line, buffered, captured, etc. in MENU FORM! Select item #7. The CP/M PIPMAG from the Main CP/M RoundTable Menu (page 685), or type 'M688' at any GEnie System Prompt.

The PIPMAG, in menu form, can be updated by your PIPMAG Editor in sections, and sometimes contain different and new information from the PIPMAG.ARK file. Check the file dates that are to the right of the general PIPMAG articles for the current status of that column.

GEnie Page 688
The CP/M PIPMAG

1. Copyright Notice	900214	
2. Data Base Column	900219	
3. Word Wrangling Column	900219	
4. Hardware Column	900318	
5. Tele-Comm Column	900215	
6. Z-Systems Column	900214	
7. PIPMAG Feature	900220	
8. GEnie CP/M Column	900219	
9. Guest Column	900219	
10. Editorial	900221	
11. Submissions Explained	900215	
12. Getting Help, Libraries	900220	<---- Help Files
13. Getting Help, B.B.	900220	<---- Help Files
14. Getting Help, General	900220	<---- Help Files

============

<END>

Auszug der Files CP/M Round Table **Seite: -1-**
Download Host GEnie ; Egbert Schröer

ALL Libraries.

```
No. File Name            Type Address     YYMMDD Bytes   Access Lib
----- ---------------------- - ------------ ------ -------

8109 RCPM0493.LBR         X M.FINN       930416  42880    1  26
     Desc: International RCP/M Systems Lists
8108 FORMAT23.LBR         X BRIAN-CPM    930409  48128    1  11
     Desc: Morrow MDHD Formatter
8107 RENAMZ20.LBR         X BRIAN-CPM    930409  76160    5  33
     Desc: Powerful REName for Z3 and CP/M
8106 MIX-C-DS.LBR         X BRIAN-CPM    930409   9600    1   3
     Desc: Doc for MIX C Dynamic strings
8105 D23.LBR              X BRIAN-CPM    930406  41728   10  33
     Desc: Handy DIR replacement for Z3 & CP/M
8104 CMAZE25.LBR          X BRIAN-CPM    930406  42752    4  33
     Desc: Z3 & CP/M 'Maze' game
8103 CRR0150A.ZIP         X BRIAN-CPM    930406  57344   26  20
     Desc: Shareware CP/M XRS/QWK mail reader
8102 ELAPSE13.LBR         X BRIAN-CPM    930406  11392    7  33
     Desc: Ampro/YASBEC elapsed timer for Z33+
8101 CLOCK22.LBR          X BRIAN-CPM    930406  47104    3  33
     Desc: Set/Read Kenmore/Anapro/H89 clocks
8100 BPCLKS.LBR           X BRIAN-CPM    930406   5248    9  33
     Desc: Fixed YASBEC clock routines
8099 OSB1ZMP.ARK          X C.COTHAM     930330  70656    2  20
     Desc: ZMP 1.5 Ready to run for Osborne 1
8098 GRAIL.LBR            X J.RUSSELL43  930319  41216   11  15
     Desc: CP/M Text Adventure Game
8097 TCFZFEST.MZG         X JAY.SAGE     930317   2048   10  18
     Desc: Trenton '93 Z-Fest Announcement
8095 CPMUNARJ.ARK         X M.FINN       930312  24064   15  29
     Desc: .ARJ file extractor for Z80 CP/M
8094 TCJINFO.TXT          X B.KIBLER     930227   3456   14  13
     Desc: TCJ new features and sub rates
8093 DTBS0393.LBR         X M.FINN       930225  15360   11  18
     Desc: PACS CP/M SIG's Data Bus Articles
8092 BIOS-R62.LBR         X BRIAN-CPM    930220 148224   11  34
     Desc: Enhanced BIOS for C128
8091 GAME40.LBR           X BRIAN-CPM    930220 128512    7  15
     Desc: Adventure Game Compiler!
8090 ZMP-SRC.LBR          X BRIAN-CPM    930217 108800   10  20
     Desc: C Source code to ZMP
8089 RZMP-SRC.LBR         X BRIAN-CPM    930217  43776    5  20
     Desc: Source to Remote ZMP (for BBS's)
8088 Z80DSR.LBR           X BRIAN-CPM    930217  99456    7   1
     Desc: Pascal Source to Z80DISassembler
8087 FAUST.LBR            X BRIAN-CPM    930217 124928    9  15
     Desc: Text Adventure game
8086 ZERR16C.LBR          X BRIAN-CPM    930207  49792    7  33
     Desc: ZCPR3 Error Handler (works with LSH)
8085 BCOMPSRC.LBR         X BRIAN-CPM    930207  37632    4  33
     Desc: Source to BCOMP30 - Binary Compare
8084 BCOMP30.LBR          X BRIAN-CPM    930207  25728    9  33
     Desc: Binary file comparison/patcher
8083 LSH12B.LBR           X BRIAN-CPM    930207  58368    7  33
     Desc: Z-System History Shell
8082 XOX11SRC.LBR         X BRIAN-CPM    930207  50944    5  33
     Desc: Source to Z-System file viewer
8081 XOX11.LBR            X BRIAN-CPM    930207  25600    5  33
     Desc: Z-System text file viewer
8080 CTUTOR2.ZIP          X MBLINDMAN    930130  96256    4   3
     Desc: Part 2 of ctutor
8079 CTUTOR1.ZIP          X MBLINDMAN    930130 113024    2   3
     Desc: part 1 of c tutorial (generic)
8078 QT-TRSM2.ZY          X D.FARNHAM    930127   3968    1  20
     Desc: QTERM Patch for the TRS-80 Model II
8077 QT-SANY1.LBR         X D.FARNHAM    930126   8192    1  20
     Desc: QTERM patch for Sanyo MBC-1150
8076 FLIFE12.LBR          X BRIAN-CPM    930124  29440    4  15
     Desc: Z3 & CP/M game of Life (assembly)
8075 Z80TEST.LBR          X BRIAN-CPM    930124  20736    6   1
     Desc: Assembler Test routines
8074 ZSDHLP2.LBR          X BRIAN-CPM    930124  24448    7   1
     Desc: HLP files for ZSDOS programming
8073 LSH12SRC.LBR         X BRIAN-CPM    930124  62464    6  33
     Desc: Source for Z3 History Shell
8072 LSH12.LBR            X BRIAN-CPM    930124  59008    6  33
     Desc: Z3 History Shell
8071 DSTATS14.LBR         X BRIAN-CPM    930124  22912    9  33
     Desc: ZCPR3 disk/user statistics utility
8070 CPMLIBS.DYF          X BRIAN-CPM    930123 125824    4   7
     Desc: dBase II CP/M RT file list
8069 YAS_PALS.ZIP         X P.CHIDLEY    930122  10368    9  38
     Desc: Source & Jedec files used on Yasbec
8068 CPMLIBS.LYT          X BRIAN-CPM    930122 133632    6  18
     Desc: ASCII capture of CP/M RT file list
8067 TPATCH.LBR           X M.FINN       930118  74368    8  24
     Desc: Turbo Pascal Patches
8066 PCTAX92.LBR          X DWGREEN      930113   7040    5  30
     Desc: Form 1040 tax template
8064 ZMO1805.ZZ0          X MBLINDMAN    930106   9344    4  20
     Desc: corrected version of zmo-1805.zz0
8063 COMM122.LBR          X BRIAN-CPM    930102  41856   18  20
     Desc: X/Y/ZMODEM & Emulation for Z3PLUS!
8062 ZCNFG24.LBR          X M.FINN       921231 120064   22  33
     Desc: configuration utility for COM files
8061 XDRO10.LBR           X BRIAN-CPM    921229   8448    7  11
     Desc: Set/Reset disk read/write flag
8060 BIGDIR12.LBR         X BRIAN-CPM    921229  24192    5  33
     Desc: BBS utility : directory of all NDR
8059 SZRZ100.LBR          X BRIAN-CPM    921229  14336   17  20
     Desc: ZMODEM for CP/M+
8057 LT31.LBR             X M.FINN       921210  64768   25  29
     Desc: Types and extracts compressed files
8056 DTBS1992.LBR         X M.FINN       921125  50944   14  18
     Desc: PACS CP/M SIG Data Bus Articles '92
8054 OS2CPM.ZIP           X M.FINN       921115  78848    6  38
     Desc: CP/M Emulator for OS/2 Systems
8053 LIBHL45A.LBR         X M.FINN       921114 180224   15  33
     Desc: HELP files for Version 4.5 LIBs
8052 ZERR16B.LBR          X BRIAN-CPM    921105  53632   10  33
     Desc: ZCPR34 Error Handler
8051 TCJ54.WZ             X TCJ$         921105  12800    8  33
     Desc: Jay Sage column from TCJ #54
8050 TCJ53.LBR            X TCJ$         921105  16000    5  33
     Desc: Jay Sage column from TCJ #53
8049 TCJ52.LBR            X TCJ$         921105  21120    5  33
     Desc: Jay Sage column from TCJ #52
8048 TCJ51.WZ             X TCJ$         921105  12800    6  33
     Desc: Jay Sage column from TCJ #51
8047 TCJ50.LBR            X TCJ$         921105  16640    5  33
     Desc: Jay Sage column from TCJ #50
8046 MOVE23.LBR           X BRIAN-CPM    921101  31360   16  33
     Desc: Change user # of a file
8045 NZBYE011.LBR         X BRIAN-CPM    921101  99840   16  33
     Desc: New Z-System BYE
8044 ZPM3N08.ARK          X R.FAIRBANKS2 921025  50432   30   4
     Desc: Simeon Cran's updated version of CP/
8043 MYZ80103.ZIP         X I.COTTRELL   921024  76800   25  38
     Desc: Z80 Emulator for MS-DOS
8042 KMDZCFG.LBR          X BRIAN-CPM    921008  33024    4  33
     Desc: CFG source files for KMDZ
8041 KMDZ0C.LBR           X BRIAN-CPM    921008 112000    7  33
     Desc: KMDZ -- KMD for NZBYE and ZCPR3
8040 ZP17A.LBR            X M.FINN       921007  96640    6  33
     Desc: ZCPR3 record patcher
8039 RTOS.LBR             X P.CHIDLEY    921002  18304   11  38
     Desc: Real Time Operating System for Z180
8037 C128-XGR.LBR         X E.R.FLINN    920926   4224   11  34
     Desc: Superior Z-System TCAP for C128
8036 LIBHLP45.LBR         X BRIAN-CPM    920926 165248   10  33
     Desc: Help for version 4.5 Libraries
8035 LIBS45.LBR           X BRIAN-CPM    920926  74880   12  33
     Desc: SYSLIB/VLIB/Z3LIB 4.5  DSLIB 4.4
8034 LBREXT35.LBR         X BRIAN-CPM    920926  37504    7  33
     Desc: Library extract/decompress utility
8033 SALIS16B.LBR         X BRIAN-CPM    920926  29696   10  33
     Desc: Screen-oriented Alias editor
8032 LT30.LBR             X BRIAN-CPM    920926  57216   12  33
     Desc: Library Type/Uncompress
8030 ZGOLF22.LBR          X BRIAN-CPM    920926  19840    6  33
     Desc: Z-System Solitaire Game
8029 TCJ SURVEY FOR 9/92  X B.KIBLER     920923   1280   12  38
     Desc: TCJ survey for readers input.
8028 THE COMPUTER JOURNAL X B.KIBLER     920923   7680   18  38
     Desc: TCJ update as of Sept 1992
8027 X10-S34.LBR          X R.SWENTON    920918 104960    7  38
     Desc: X10 CP-290 Control Software Source
8026 X10-C34.LBR          X R.SWENTON    920918  67072    7  38
     Desc: X10.COM CP-290 Software for Z80
8025 NZBOVL.LBR           X BRIAN-CPM    920914  19072    8  33
```

Desc: NZBYE sample overlays
8024 NZBYE08.LBR X BRIAN-CPM 920914 89856 4 33
Desc: New Z-BYE for PBBS and Z-MSG and Z3
8023 EDZCM11.LBR X BRIAN-CPM 920914 36352 6 33
Desc: Edit NZCOM .ZCM files
8022 ARC20.ARC X BRIAN-CPM 920914 70144 16 29
Desc: Pascal Source to Tillman Reh's ARC
8021 Z-FOR-AT.ARK X BRIAN-CPM 920914 13312 20 33
Desc: Article on Z-System on an AT system
8019 ARCCPM20.COM X PRESS54 920823 27264 35 29
Desc: True ARC CPM FAST! Turbopas
8018 AMP8ZU13.LBR X BRIAN-CPM 920821 95360 3 33
Desc: Ampro disk and config utils for Z3
8017 LBCLKS12.LBR X BRIAN-CPM 920821 17920 3 33
Desc: ZCPR33+ Ampro LB Clock utilities
8016 DIF12.LBR X BRIAN-CPM 920821 29184 8 33
Desc: Compare two text files (redir IO)
8014 MYZ80100.ZIP X STEVE.HIRSCH 920801 58496 30 1
Desc: Z80/CPM Emulator for MS-DOS 80x86
8012 RCOPY22.LBR X BRIAN-CPM 920727 28032 5 33
Desc: Copy files from a list
8011 SILENT16.LBR X BRIAN-CPM 920727 12032 2 33
Desc: Force a program to be silent
8010 PWSKIM.LBR X J.BREWER7 920709 6528 3 12
Desc: Rescues lost text, skims ASCII chars
8009 BUIHD09.LBR X BRIAN-CPM 920707 68736 7 17
Desc: Build WS file from .HLP files
8008 WSZDS10B.LBR X BRIAN-CPM 920707 5120 10 33
Desc: Patch for WordStar 4 and ZSDOS
8007 SCOPY10.LBR X M.FINN 920702 27904 7 33
Desc: Z-System Screen oriented copier
8006 EDISK30.LBR X BRIAN-CPM 920621 31744 5 11
Desc: Foreign Disks for Ampro and Z3
8005 ZDM.LBR X BRIAN-CPM 920621 76544 22 1
Desc: Z80/Z180 Debuger (like DDT)
8004 UNZ80.ZIP X SMJ 920531 29056 16 1
Desc: Z80 Threading Disassembler: 8086 Src
8002 ZDE16XRX.PAT X G.BARNES 920511 3200 7 12
Desc: ZDE16 patch, Xerox 16/8 cursor keys
8001 PERFRM13.LBR X I.COTTRELL 920510 17408 4 20
Desc: Accessory for FOR.COM for RCP/M.
8000 UNZIP18.LBR X I.COTTRELL 920510 25472 41 19
Desc: Explode MESS-DOS .ZIP files.
7998 MU317.LBR X I.COTTRELL 920510 27136 11 33
Desc: Memory Utility for Z-System.
7997 FILT83.LBR X I.COTTRELL 920510 27904 4 12
Desc: Update to Irv Hoff's FILT program.
7996 SLRHLP2.LBR X BRIAN-CPM 920507 9088 10 1
Desc: HLP files for SLR assemblers
7995 MENU42.LBR X BRIAN-CPM 920507 45568 13 33
Desc: ZCPR Menu Shell -- now "AT" aware
7994 XRUN25.LBR X BRIAN-CPM 920507 38656 12 33
Desc: Extended Command Processor for Z33
7993 ZDB18C.LBR X I.COTTRELL 920426 43776 6 33

Desc: ZCPR3 DataBase
7992 ZBIB11A.LBR X I.COTTRELL 920426 34944 3 33
Desc: Z3 Bibliographic Dbase File Manager
7991 SNAP13.LBR X I.COTTRELL 920426 15232 4 33
Desc: Saves system package segs to a file
7990 IOPZXR12.LBR X I.COTTRELL 920426 13056 12 33
Desc: Debugging IOP for Z3
7989 Z3HELP.CZT X Z-SUS$ 920413 5376 14 33
Desc: Z-SUS Help File Ordering Info
7988 Z3COM.CZT X Z-SUS$ 920413 7168 9 33
Desc: Z-SUS .COM File Ordering Info
7987 TCSELE31.LBR X BRIAN-CPM 920412 30720 27 33
Desc: TermCap installer for CP/M & Z3
7986 AUNTIE.KIT X AUNTIE 920410 5632 9 17
Desc: AUNTIE HELP KIT
7985 DSSW2.ZZ0 X L.MOORE15 920409 4096 5 38
Desc: set a TV803H heartbeat with ds1612E
7983 PCXSLIDE.LBR X S.GOLDSMITH2 920408 117888 19 34
Desc: C128 CP/M 80 column PCX Slide Show!
7982 ZPM3.LBR X I.COTTRELL 920405 31488 36 4
Desc: Replacement BDOS for CP/M Plus
7981 DINOSIG.DZK X GLEEPY 920404 7168 25 18
Desc: DinoSIG Boot Disk Listing
7980 ZSLIB36.LBR X I.COTTRELL 920322 108544 10 33
Desc: Additional library routines f/ZCPR3
7978 CCOUNT10.LBR X I.COTTRELL 920322 25088 6 12
Desc: Counts characters in a file.
7977 SZAP4A.LBR X I.COTTRELL 920313 75392 10 11
Desc: Disk editor update for Z-System.
7976 LHH11.LBR X I.COTTRELL 920313 24576 10 33
Desc: ZCPR LBRHLP for .?Y? files.
7975 LBRHLP21.LBR X I.COTTRELL 920313 48128 12 33
Desc: Execute .HLP files within a library.
7974 IOPZXR11.LBR X I.COTTRELL 920313 17408 7 33
Desc: ZCPR IOP - system exam & patch tool.
7973 AREA20.LBR X I.COTTRELL 920313 16128 8 38
Desc: Area code database. City/state/prov
7971 MAINT13.LBR X I.COTTRELL 920308 38912 13 11
Desc: Directory (disk) maint util.
7970 ZMO-YB20.MZC X I.COTTRELL 920308 8832 4 20
Desc: ZMP overlay for YASBEC.
7969 QT4-YB20.MZC X I.COTTRELL 920308 7424 5 20
Desc: Qterm overlay for YASBEC.
7968 GT-ZSDOS.Z X I.COTTRELL 920308 2432 3 20
Desc: UUCP time insert for ZSDOS.
7967 LOGMK3.LBR X I.COTTRELL 920308 18944 14 20
Desc: Qterm chat script generator.
7966 DATSTP19.LBR X I.COTTRELL 920308 62080 8 33
Desc: Z-System date stamp utility.
7965 RCPM0392.BZF X I.COTTRELL 920308 9472 14 26
Desc: RCP/M List for March (short vers).
7964 RCPM0392.LZT X I.COTTRELL 920308 34944 17 26
Desc: International RCP/M List for March.
7963 XFOR16.LBR X I.COTTRELL 920308 46080 4 33

7962 ZDT13.LBR X I.COTTRELL 920308 41472 7 33
Desc: Z-System Day Timer.
7960 KEYIN12.LBR X BRIAN-CPM 920307 11520 6 33
Desc: NZCOM keyboard input buffer
7959 ZF10Q.LBR X BRIAN-CPM 920307 120192 16 33
Desc: ZFiler -- Z Shell/File manager
7958 ZERR16.LBR X Z-SUS$ 920307 24576 9 33
Desc: ZCPR34 Error handler
7957 Z3HELP42.LBR X Z-SUS$ 920307 49536 13 33
Desc: Update #42 to Z-SUS Help collection
7954 PUBTOOL1.LBR X L.CAVANAUGH 920222 22656 3 20
Desc: Handy sysop util. for file requests
7953 WAIT13.LBR X I.COTTRELL 920217 11392 7 33
Desc: ZCPR3 delayed-execution tool
7952 LD13.LBR X I.COTTRELL 920217 49280 12 33
Desc: Library directory lister
7951 TCJ-CO.TXT X PETER-CPM 920216 30976 13 18
Desc: TCJ Conference transcript 2/15/92
7950 TRENTON.LBR X I.COTTRELL 920215 4096 11 18
Desc: Details of 8-bit at Trenton CF.
7946 EXTEND14.LBR X Z-SUS$ 920131 9600 12 33
Desc: Text file extender for CP/M & ZCPR3
7945 LPUT22.LBR X Z-SUS$ 920131 34944 15 33
Desc: PUT files into a Library for Z3
7944 LX22.LBR X Z-SUS$ 920131 48512 15 33
Desc: Run programs from library files
7942 NTS11B.LBR X Z-SUS$ 920131 31104 12 33
Desc: Screen-oriented NoteTaker for Z3
7941 LSH11.LBR X Z-SUS$ 920131 66944 11 33
Desc: ZCPR34 History Shell/Error Handler
7939 ZCRCK14.LBR X Z-SUS$ 920131 16384 5 33
Desc: Display/Print and save file CRC's
7938 REMIND14.LBR X Z-SUS$ 920129 36224 7 33
Desc: Appointment reminder for Z-System
7937 LDIRB21.LBR X Z-SUS$ 920129 25088 9 33
Desc: LBR DIRectory for Z3 & CP/M
7936 DSTATS12.LBR X Z-SUS$ 920129 18944 10 33
Desc: Display disk usage
7935 CPD10.LBR X Z-SUS$ 920129 20736 5 33
Desc: Compare Disk Directories for Z3
7934 DIALER.LBR X Z-SUS$ 920129 9088 6 33
Desc: Phone dialer for Z3 and Hayes modems
7933 PSEUDO2.DOC X S.MITCHELL19 920128 11008 17 1
Desc: A comparison of assembler pseudo-ops
7931 ZMAC.DOC X S.MITCHELL19 920119 9856 14 1
Desc: Info on ZMAC assembler package
7930 EPEX11.ARK X L.CAVANAUGH 920110 62208 15 33
Desc: An easily-installed ZCPR3 look-alike
7928 CHED80.LBR X S.GOLDSMITH2 920110 72832 22 34
Desc: 80 col. color char editor for C128
7927 PCTAX91.LBR X DWGREEN 920110 9088 6 30
Desc: Perfect Calc 1040 template
7922 BATCH+.LBR X C128-BILL 920108 13824 29 27

Desc: Conditional SUBMIT for CP/M+
7920 SCSIDOCS.LBR X I.COTTRELL 920107 115968 27 18
Desc: Docs of SCSI interface
7919 INPUTCPM.TQT X G.STAGLIANO 920107 36352 20 18
Desc: Begining CP/M for C128
7916 CPM011.ARK X C128-BILL 920106 92288 6 27
Desc: FOG Disk: TOUR outline Processor
7915 CPM004.ARK X C128-BILL 920106 87424 9 27
Desc: WS3 Patches and WS Utiliites
7914 CPM159.ARK X C128-BILL 920106 134400 8 27
Desc: Wordstar Print enhancement programs

Impressum

1. Vorsitzender:	**Hartmut Obermann** Mozartring 23 Postfach 14 30	Tel.: 0 82 21/ 3 02 48 FAX: 0 82 21/ 3 35 75
89304	Günzburg	
2. Vorsitzender:	**Gerald Schröder** Am Schützenplatz 14	Tel.: 0 41 05/ 26 02
21218	Seevetal 1	
Hardwarekoordinator:	**Andreas Magnus** Bismarckstraße 29	Tel.: 02 09/ 87 02 30
45879	Gelsenkirchen	
NewDOS-Diskothekar:	**Oliver Volz** Am Ochsenwald 37A	Tel.: 07 11/ 74 40 51
70565	Stuttgart 80 (Rohrerhöhe)	
CP/M-Diskothekar:	**Fritz Chwolka** Saarstraße 34	Tel.: 0 24 64/ 89 20
52457	Aldenhoven	
C-128-Diskothekar:	**Günter W. Braun** Postfach 80 02 26	
81602	München 80	
Clubbücherei:	**Kurt Müller** Sophie-Scholl-Ring 3b	Tel.: 0 41 52/ 7 06 43
21502	Geesthacht	
Redaktion:	**Jens Neueder** Gschlachtenbretzingen Rudolf-Then-Straße 32	Tel.: 07 91/ 4 28 77 FAX: 0 79 71/ 2 50 55
74544	Michelbach / Bilz	
Bankverbindung:	**Club 80** Postgiro Sonderkonto CLUB 80 Obermann H., 8870 Günzburg	**Postgiroamt Frankfurt** **BLZ: 500 100 60** **Kto.Nr.: 496 071 - 605**
Autoren:	Die Redaktion bedankt sich bei den im **Inhaltsverzeichnis** genannten Autoren für die Mitarbeit an der Club-INFO. Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.	

Schluß

Hallo Club 80'er,

sicher habt Ihr das INFO schon gelesen und festgestellt, daß wir recht NEUDEUTSCH angehaucht sind. Ich hoffe, Ihr findet auch diesmal wieder viel Gefallen am aktuellen INFO.

Die Entscheidung über das Clubinfoformat steht, wie Ihr seht, immer noch aus. Aus diesem Grund **-und die für uns ganz wichtige Beantwortung aller allgemeinen und clubbezogenen Fragen-**, möchten wir Euch bitten, sofern nicht schon geschehen, an der Fragebogenaktion teilzunehmen. Das bisherige Feedback war noch nicht so berauschend. Wir würden gerne etwas für Euch/uns alle tun, aber dies geht nur mit Eurer Mithilfe.

Als weiteres gelten ab Juli, natürlich Allen bekannt, die neuen Postleitzahlen. In dem anhängenden Adressenverzeichnis wurden sie, soweit möglich, schon eingearbeitet. Bitte überprüft Eure Daten und teilt mir aktuelleres rechtzeitig mit. Ab INFO 41 wird nach den neuen Zahlen versandt und... ...wie bekannt.

So, daß wars auch schon wieder, ich wünsche Euch eine schöne Urlaubszeit!
(vergesst nicht einen Artikel fürs nächste INFO an mich einzutüten)
bis zum nächsten mal Euer

Jens

Adressen-Liste CLUB80

Stand vom 20-06-1993
Seite 1

Nachname	Vorname	Straße	PLZ	Ort	Telefon privat	Telefon geschäftlich
Behrendt	Detlef	Schlosserbreite 1a	85567	Grafing	08092 / 9173	
Berndt	Wolfgang	Friedberger Straße 92c	61169	Friedberg	06031 / 2963	
Berndt-Jochum	Ilse	Stachelsgut 24	51427	Bergisch-Gladbach	02204 / 65254	
Bernhardt	Helmut	Hafenstraße 7	24226	Heikendorf	0431 / 241907	0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	91353	Hausen	09191 / 31698	09191 / 611108
Brans	Jörg	Tieloh 55	22307	Hamburg	040 /6906531	
Braun	Harald	Postfach 8011		Kiel		
Braun	Günter W.	Postfach 800 226	81602	München		
Böckling	Ulrich	Jochaczstraße 61	56203	Hör-Grenzhausen	0264 / 4861	02631 / 895168
Chwolka	Fritz	Saarstraße 34	52457	Aldenhoven	02464 / 8920	
Dose	Volker	Dorfstraße 10	24235	Brodersdorf	04343 / 1357	
Gill	Thomas	Maria-Eich-Str. 34	81243	München	089 /8349527	
Günther	Jens	Bannerscheid 7	57638	Neitersen	02681 / 1553	
Halgasch	Gert	Großschonauer Straße 26	02796	Jonsdorf		
Hartmann	Hans-Günther	Möwenstraße 9	27804	Berne	04406 / 6911	0421 /2482419
Hebecker	Ulrich	Büsumerstraße 15		Stuttgart	0711 / 734800	
Held	Manfred	Stirner Straße 22	91785	Pleinfeld	09144 / 6563	0911 /2195245
Hermann	Klaus	Forchenstraße 8	72124	Pliezhausen	07127 / 71945	
Hürdler	Manfred	Niederhofer Straße 29		Rimper	09365 / 4235	
Johnen	Willi	Hansemannstraße 1	52351	Düren	02421 / 501305	
Kauka	Dietmar	Straße des Friedens 37	A- 7201	Neukirchen	0037404/850112	
Kemmer	Jürgen	Dorfberg 7	97232	Sulzdorf	09334 / 1050	
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 /4828948	00411 / 258170
Kuhn	Eckehard	Im Dorf 14	72636	Frickenhausen	07022 / 45417	
Littmann	Claus	Plockhorst, Zum Spring 15	31234	Edemissen	05372 / 7796	
Lorenz	Walter	Mahräckerstraße 9	60431	Frankfurt /Main 50	069 / 531656	
Magnus	Andreas	Bismarckstraße 29	45879	Gelsenkirchen	0209 / 870230	
Mahlert	Herbert	Hohenbudbergerstraße 112 A	47229	Duisburg	02135 / 47217	
Mand	Harald	Kl. Flintbeker Straße 7	24220	Flintbeck	04347 / 3629	0431 /3013580
Menk	Christian	Ollsener Straße 52	21271	Hanstedt	04184 / 7825	
Müller	Kurt	Sophie-Scholl-Ring 3b	21502	Geesthacht	04152 / 70643	
Mössel	Franz	Schafferstraße 12	I- 39012	Meran		
Neebe	Gerhard	Märkische Straße 186	44141	Dortmund	0231 / 416549	
Neueder	Jens	Rudolf-Then-Straße 32	74544	Michelbach / Bilz	0791 / 42877	0791 /44-4722
Neumann	Christof	Zeitblomstraße 22/2	89077	Ulm	0731 /6022568	0731 /9749720
Nitschke	Stefan	Germanenstraße 5	75045	Walzbachtal	07203 / 452	
Obermann	Hartmut	Mozartring 23	89312	Günzburg	08221 / 30248	
Peters	Jürgen	Heukoppel 14	22179	Hamburg	040 /6412371	
Reit	Hermann	Vechtaer Hof 40	49088	Osnabrück	0541 / 16331	
Retzlaff	Bernd	Kleiner Sand 98	25436	Uetersen	04122 / 43551	04103 / 605310
Riechmann	Michael	Letelner Heideweg 12	32423	Minden	0571 / 36627	
Rinio	Gerd	Rennbahnstraße 9	22111	Hamburg	040 /6552630	
Ruschinski	Claus	Pommernstraße 21	45770	Marl	02365 / 34646	
Schilling	Werner	Ehndorfer Straße 340	24537	Neumünster	04321 / 61116	
Schimmer	Jörg	Stettinerstraße 28	60388	Frankfurt	06109 / 35336	
Schmid	Alexander	Entmannsdorf 5	96317	Kronach/Gehülz	09261 / 53496	
Schmitz	Rainer	Küferweg 12/1	73099	Adelberg	07166 / 1397	
Schober	Frank-Michael	Weberweg 2	02742	Spremberg	0037574/ 4565	
Schoberth	Uwe	Petrus-Waldus-Straße 14	75443	Oetisheim	07041 / 7254	0711 /2011455
Scholz	Hans-Werner	Spitalstraße 54	41334	Nettetal		
Schroers	Horst-Dieter	Breslauer Straße 9	85622	Feldkirchen	089 /9032615	089 /9032615
Schröder	Gerald	Am Schützenplatz 14	21218	Seevetal	04105 / 2602	
Schröder	Peter	Theodor-Fahr-Straße 32	22419	Hamburg	040 /5311582	
Schröer	Egbert	Joachimstraße 18	46284	Dorsten	02362 / 75311	02362 / 494392
Schulte	Hartmut	Entenschnabel 8	31311	Uetze	05173 / 1248	
Schut	Andre	Sanderstraße 26	12047	Berlin	030 /6917861	
Schwarz	Wolfgang	Schwedenring 6	86609	Donauwörth	0906 / 3092	
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	53347	Alfter	0228 / 643853	
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanloese		
Sonnemann	Harald	In den Eckwiesen 9	64405	Fischbachtal	06166 / 8512	
Stumpferl	Stefan	Hasenberglstraße 57	80933	München	089 /3144001	
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	55122	Mainz	06131 / 320860	06131 / 395268
Tornow	Wilhelm	Elbblick 46	21629	Neu Wulmstorf	040 /7007280	04723 / 3063
Vollkmer	Richart	Am Spörkel 69	44227	Dortmund		
Volz	Oliver	Am Ochsenwald 37A	70565	Stuttgart (Rohrerhöhe)	0711 /744051	0711 /6853013
Werner	Heiko	Reichenberger Straße 5	01129	Dresden		
Wittkamp	Heinz	Hindenburgstraße 37	42853	Remscheid	02191 / 75132	
Wulf	Hans-Otto	Im Brahmkamp 38	46242	Bottrop	02041 / 688972	

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

Beachtet die Umstellung der
Postleitzahlen!

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

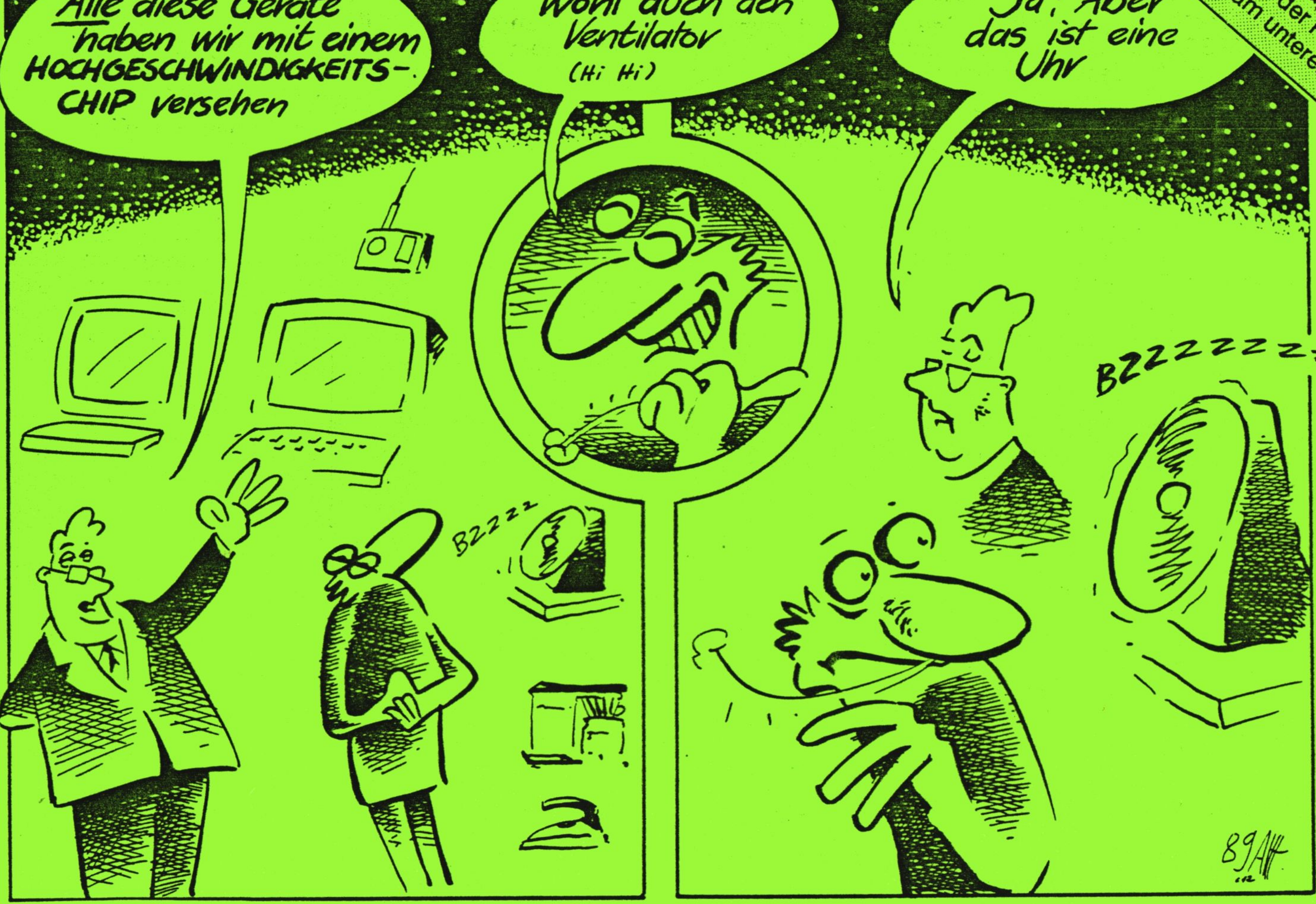
TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

Achtung !!
wichtiger Termin !!

Hobbyelektronic-Treffen 93

am Sonntag, den 07.11.1993
gleich morgens zum Einlaßbeginn
oder
mittags, 12.00Uhr
nach der Kartenkontrolle
am unteren Eingang



CLUBINFO

41. Ausgabe

Kontaktadresse:

Club 80

Hartmut Obermann

Mozartring 23

Postfach 1430

8870 Günzburg

Tel.: 08221/30248

BTX: 08221/30248

Anrufbeantworter &

FAX: 08221/33575

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Zusammenfassung Fragebogenaktion 1 - 5
Jens Neueder

Neues vom Vorstand 6 - 8
Club 80-Neumitglieder aus "CP/M-aktuell"-Usergruppe
Auswertung der Fragebogen zur Mitgliederkarteiaktualisierung

Nachträge zum Vorwort 9 - 10
Hartmut Obermann

Clubinfo 11 - 12
Termine 13
Jens Neueder

Vorstellung 13 - 14
Jörg Lindner

Software

Austausch von Grafiken DOS -> CP/M 15 - 18
Alexander Schmid

Grafik des GIIIs unter Holte CP/M+ 19 - 26
Volker Dose, Egbert Schröer

Model 4 intern: Dem Interpreter aufs Bit geschaut 27 - 35
Artikel aus CP

MSDOS-Trick: Command /F 35
Insider-Info: Windows-Fehler 68
Hartmut Obermann

Microsoft Link80 und .SYM-Datei 36
Uwe Schobert

Software-Technik und Compilerbau 37 - 38
Artikel aus PASCAL

IMP - ein DFÜ-Programm für CP/M 39 - 44
Günther W. Braun

Internet und UUCP mit CP/M 45 - 58
Volker Dose

Internet via Genie 59 - 62
GE-Mail aud Internet Access 63 - 64
Egbert Schröer

Autor & Seite

Hardware

Floppy-Tester 65 - 66
Artikel aus Elektor

Club 80 Börse

Suche ... 67
Uwe Schobert

Sonstiges

Neuigkeiten für CP/M 69
Alexander Schmid

Genie News: Neue Preisstrukturen 70 - 71
Wissenswertes ZCPR oder Z3Plus 72 - 73
Literaturrecherche: USA, England, Deutschland 73 - 85
Egbert Schröer

Wichtige Tips ... Elektrikers 86
Alexander Schmid

Die letzten Seiten

Impressum 87
Schluß 88
Redaktion

Mitgliederadressenliste am INFO-Ende
Sonderheft: Wissenswertes rund um den Z280 am INFO-Ende

Fragebogenaktion

Auswertung/Zusammenfassung

Von 70 versandten Fragebögen erhielten wir bisher 43 mehr oder weniger ausgefüllt wieder zurück. Die folgenden Seiten geben eine Zusammenfassung Eurer Antworten/Angaben wieder. Im Vorwort von Hartmut wird auch auf manchen Punkt eingehend Stellung bezogen. Zu den gegebenen Fragebogen-Antworten könnt Ihr aber vielleicht noch weitere Vorschläge zu allgemeinen Aktivitäten im Club machen.

Verhältnis aktiv / passiv:

Neben 14 aktiven Mitgliedern meldeten sich 27 Club80'er als passiv. 2 Mitglieder ließen diesen Punkt unbeantwortet.

Gruppenarbeit mithelfen:

Für eine Mitarbeit in einer Gruppe könnten sich 19 Leute erwärmen.

Artikel schreiben:

Neben 3 Hardwareschreiber fanden sich noch 17 Mitglieder, die in der nächsten Zeit einen Softwareartikel schreiben würden. Dazu sind folgende Themen in Aussicht:

Hardwarebeiträge:

MS-DOS-Rechner,
VGA-Videokarten für 8Bit-Rechner

Softwarebeiträge:

Betriebssysteme (allgemein), ZCPR
Programmiersprachen,Object Vision
BASIC, besondere BASIC-Programmierung,
Compilerbau
Shareware
DFÜ, Fido-Netz
Grafik + Speicherausbaueinbindung in TPascal + BS
Physikalische, mathematische Simulation

vorhandene Erfahrungen:

21 Club-Mitglieder würden gern ihre Hard- und/oder Softwareerfahrungen den anderen Clubmitgliedern bereitstellen. Folgende Themengebiete können angeboten werden:

Allgemein:

DFÜ(3), Amateurfunk, Astronomie
Übersetzungen aus dem Englischen
sehr große, kommerzielle Anwendungen

Hardware:

Z80 + Peripherie, Z80-Prozeßsteuerung
PC-Hardware, Diskettenlaufwerke
Rechnerkopplung
Amstrad CPC's (Hardwareprogrammierung)

Software:

Betriebssysteme (allgemein)
LS-DOS, G-DOS
CP/m 2.2, ZCPR, Alphatronik P3
Genie IIIs mit Holte CP/M+
CP/M Software, DFÜ mit CP/M
CRR-Offlinereader
Compilerbau
Programmiersprachen (allgemein) (2)
BASIC (2), Objekt Vision
Mathematik mit Pascal
Shareware MS_DOS und Windows

Erfahrungsaustauschwünsche:

Über folgende Themen wurde von den Fragebogenbeantwortern ein Erfahrungsaustausch gewünscht. Teilweise deckt sich der "Bedarf" mit den vorhandnen Erfahrungen. Hier besteht auch die Möglichkeit, daß INFO aktuell zu beleben. Stellt Eure Fragen öffentlich im INFO. Sicher lässt ich für vieles eine Antwort finden. Vielleicht gibt auch mal eine Diskussion unter Fachleuten? Hier nun die Wunschthemen der Mitglieder:

Allgemein:
DFÜ
Astronomie
Computer und Software der 80'er

Hardware:
S100-BUS
Z180 usw., HD 64180
Netzwerke
CP/M-Rechner
Steuerungen mit Rechner (I/O) (2)
Hardware allgemein

Software:
OS/2
G-DOS-Programmierung
Illegalverarbeitung
Compilerbau,
CP/M BIOS-Tuning (RamDisk, HD)
CP/M 86, CP/M 2.2, MDVCPM, Z3-Plus-Umgebung
C-Compiler unter CP/M
C-Programmierung (2)
BASIC, TurboPascal
Grafikbearbeitung
Umsteigen + IBM-Software

Interesse einfach mal so:

Für folgende Themen wurde zusätzlich Interesse bekundet.

Hardware:
Z280 im GIIIs
Reperaturanleitungen, -Erfahrungen

Software:
Z80/8088-Assemblerprogrammierung
VZI (Z80 Unix)???
ZCPR auf 8085CPU
neue CP/M-Software (nach dem Tod der Z-SUS)
wie ein Compiler funktioniert

Clubleben:

Hier die Sammlung Eurer Vorschläge für die Verbesserung des Clublebens:

- mehr Aktivität in Richtung ZCPR3, da CP/M doch langsam stirbt
- alternative PD-Betriebsysteme (z.B. Linux) unterstützen
- Software-Diskothekare für die wichtigsten Rechnergruppen wählen
- Utilitie-Sammlung für CP/M
- mehr Kontakte mit Mitgliedern herstellen
- weitere Mitglieder zur Mitarbeit anregen
- mehr Regionaltreffen, mal ein Treffen im Norden,
- nicht immer am selben Ort treffen, Ort mit Bahnstation wäre wünschensert
- Veranstaltungen für die "Restfamilie" des Mitglieds beim Clubtreffen.
- S/G/M-Archiv im Club-Brett (platzsparend mit ARJ gepackt)
- Fido-Zugang zum Club-Brett (Zugangsverbilligung, mehr Kommunikation)
- in der Fido-Area CPM.GER öfters mal diskutieren bzw. diese beleben

Clubtreffen:

Anreiseentfernung:

Es meldeten sich für eine einfache Anreise bis:
200km 35 Clubmitglieder, 400km 23 Clubmitglieder, 600km 15 Clubmitglieder und über 600km noch 11 Clubmitglieder. Bei 3 Mitgliedern, die absolut nicht zu einem Clubtreffen kommen möchten, gab es bei den 200/400km sehr oft den Wunsch, sich mehrmals im Jahr zu treffen. Dies ist ein starkes Votum nach Regionaltreffen an strategisch günstigen Orten!
Teilweise wurde zu diesem Punkt auch keine Angaben, bzw eine Abhängigkeit von Terminen und anderweitigen Verpflichtungen gemacht.

Regionales Treffen organisieren:

8 Club-Mitglieder sehen eine Möglichkeit, ein regionales Treffen organisieren zu können.

Workshops:

Als (Mit-)Vortragende würden sich 6 Mitglieder zur Verfügung stellen. Angeboten wurden folgende Themenvorschläge:
Hard- und Software allgemein, Object Vision, Kommunikation
Als Workshopwünsche wurde folgendes benannt:

Hardware:
Hardwarebastelei, alles was mit/am Z80 machbar ist (2), Z80 durch Z280 ersetzen, systematische Fehlersuche,
I/O-Karten zum Steuern von ... (Licht,Rolladen,Markise...)

Software:
Z80/8088-Assemblerprogrammierung, Z80 und Interrupts
Textverarbeitung, Datenbanksysteme, DFÜ, Grafik
BASIC, C, TurboPascal

... an allgemeinen Workshops mit Programmierung oder Hardwarebastelei wurde sehr oft Interesse bekundet.

Clubinfo:

Formatfrage:

Bei 1 -egal wie, nur der Inhalt zählt- entschieden sich 34 Mitglieder für das Querformat und 6 Mitglieder für das Hochformat. Weiterhin gab es 2 Enthaltungen.

Zum Thema GUT/SCHLECHT machten 26 Mitglieder Angaben. Der Rest (17) enthielt sich dazu.

Als störend wurde genannt:

nichts(2),
unregelmäßige Erscheinungsweise(3), Info kommt zu selten,
manchmal zu kleine Schrift, das Format, Vereinsmeierei,
zu viel wird vorausgesetzt -ich hoffe den Anschluß zu bekommen,
alte Beiträge werden nicht wieder neu aufgelegt,
wo soll man Artikel hinschicken? (Redaktionsadresse fehlt),
manchmal ist der Autor von Beiträgen nicht erkennbar,
das Artikel oft nur unterschrieben werden und mann nicht lesen kann wer der Autor ist (Klartextname dazu),
mehr freier Rand für Lochung, die endlosen Listings,
seitenlange Mailbox-Sitzungs-Mitschnitte
keine E-Mail-Adresse, an die Clubbeiträge gesendet werden können,
schlechte Resonanz der Mitglieder auf Veröffentlichungen,
das ich keine Artikel schreibe, BASIC-Artikel,
der 2. Vorsitzende,
Mitgliederliste: Rechnertypenangabe fehlt, weitere Erreichbarkeit (per BTX/Modem) fehlt

Für gut wurde befunden:

Vielseitigkeit, Aufmachung/Gliederung/Layout/Format(4),
sehr locker, der Preis!, Clubatmosphäre, gute Führung,
die kostendeckende und nicht gewinnträchtige Arbeitsweise,
Öffnung gegenüber anderen Systemen, die Comics, das es das INFO gibt,
daß das INFO wieder regelmäßig erscheint, wenn das INFO erscheint,
Grundlagen- und Intensiv-Artikel über Grenzgebiete und "Veraltetes",
nahezu alle Artikel wegen des Engagement der Autoren,
das es Leute gibt, die Artikel schreiben,
die fachlichen Auskünfte einiger Mitglieder,
die 99% die sonst im Clubinfo abgedruckt sind,
das es noch nicht (Domes)DOS durchsetzt ist,
Artikel von Helmut und Gerald,
die Frau des 1. Vorsitzenden,

Wünsche für zukünftige INFO's:

lassen wie es ist,
mehr BASIC, mehr kurze Tips und Tricks,
DOS sollte fester Bestandteil werden, da kein Weg mehr daran vorbei führt,
Softwareecke ->Programminformationen, Shareware (da der Markt zu unübersichtlich)
Messen, Steuern, Regeln (EMR) am PC,
chemische/physikalische Simulationen am PC,
Lernsoftware für Schule und Berufsausbildung

Neues vom Vorstand

Diesmal gibt es zwei wichtige Themen, die ich in meinem Vorwort behandelt möchte. Das erste betrifft eigentlich weniger die Mitglieder des CLUB 80 als vielmehr die, die es noch werden wollen. Danach folgt dann die Auswertung der Fragebogen zur Aktualisierung der Mitgliederkartei.

I. CLUB 80-Neumitglieder aus den Reihen der "CP/M-aktuell"-Usergruppe

Wie ich schon in meinem Bericht vom CLUB 80-Treffen 93 berichtet habe, wird unser Club-Info seit seiner 39'sten Ausgabe auch in der "CP/M-aktuell"-Usergruppe verteilt und sicher auch mit Interesse gelesen.

Dazu kam es, nachdem eine von Günter Braun, dem Leiter von "CP/M-aktuell", vorgeschlagene Fusion mit dem CLUB 80 nach hartnäckigen und teilweise sehr heftig geführten Grundsatzdiskussionen verworfen wurde.

Günter bekommt von Jens jeweils die angeforderte Anzahl Infos für 10 Pfennig pro Blatt zugeschickt. Die Verteilung übernimmt er in eigener Regie. Ob er er dies kostenlos tut (die Mitgliedschaft bei "CP/M- aktuell" ist kostenfrei) oder die Infos verkauft, geht den CLUB 80 bzw. mich im Prinzip nichts an und ist mir auch nicht bekannt.

Selbstverständlich können "CP/M-aktuell"-User jederzeit Mitglied im CLUB 80 werden. Dazu müssen sie allerdings einen Mitgliedsantrag bei mir anfordern und ausgefüllt zurückschicken. Weiterhin sind 30,-- DM auf das Konto des CLUB 80 zu überweisen.

"CP/M-aktuell"-User, die ihren CLUB 80-Mitgliedsbeitrag an Günter Braun geschickt bzw. überwiesen haben, sind solange nicht Mitglied des CLUB 80, bis 1. das Geld auf dem CLUB 80-Konto und 2. ihr Anmeldebogen bei mir eingegangen ist!

Nur dann kommen sie in den vollen Genuß der Vorteile einer CLUB 80-Mitgliedschaft wie z.B. den Zugriff auf die Clubbücherei, die Programmbibliotheken, Einladungen zu Regionaltreffen, die Adressen- und Rechnerliste (die den durch Günter verschickten Infos nicht beiliegen) usw..

Es gibt aber noch einen wesentlich gravierenderen Nachteil. Sollte der CLUB 80 auch im nächsten Jahr für "Altmitglieder" beitragsfrei sein, kommen diejenigen, die die 30,-- DM an Günter Braun überwiesen haben, nicht in den Genuß der Beitragsfreiheit (es sei denn, der Betrag landet bis zum Jahresende auf dem CLUB 80-Konto und das Anmeldeformular bei mir.)!

Soweit zu diesem unerquicklichen Thema, das meiner Meinung nach aber zur Sprache gebracht werden mußte. Nun aber endlich zum Hauptthema dieses Info-Vorwortes.

II. Auswertung der Fragebogen zur Aktualisierung der Mitgliederkartei

1. Danksagung

Zunächst möchte ich mich an dieser Stelle recht herzlich bei Jens Neueder bedanken! Er war es nämlich, der die mit dem Fragebogen gesammelten Daten in ein von ihm programmiertes Verwaltungsprogramm eingegeben und die Antworten ausgewertet hat.

Bedanken muß ich mich auch bei all denen, die den Fragebogen ausgefüllt und zurückgeschickt haben. Sie haben damit bewiesen, daß sie weiterhin ein Interesse am CLUB 80 und seinen Aktivitäten haben.

2. Mitgliederzahl

Auf einem der Fragebögen befand sich unter dem Satz "Wer den Fragebogen nicht zurückschickt, bekundet sein Desinteresse am CLUB 80 und wird nicht mehr als Mitglied geführt!!!" die handschriftliche Bemerkung "Ist das rechtlich zulässig?". Ob dieses Verfahren zulässig ist oder nicht kann wohl nur ein Jurist beantworten. Für den CLUB 80-Vorstand lautete die Frage einfach, wie man in einem Club ohne Mitgliedsbeitrag die Leute aussondert, die absolut kein Interesse mehr an seinen Aktivitäten haben. Und dazu eignet sich eine solche Fragebogenaktion recht gut.

So ist dann leider auch die Mitgliederzahl des CLUB 80 von ursprünglich 70 auf 44 zusammengeschrumpft. Das entspricht einem Schwund von immerhin 37 %. Ob sich dieser Schwund negativ auf den CLUB 80 auswirkt? Ich glaube es nicht! Die aus der Mitgliederliste Gestrichenen hatten sich eh teilweise schon seit Jahren nicht mehr gemeldet und ihr Info wanderte wohl auch ungelesen in irgend einen Ordner oder, was wahrscheinlicher ist, direkt den Papierkorb.

3. Mitgliederaktivität

Im Prinzip hat sich dadurch auch das Verhältnis zwischen aktiven und passiven Mitgliedern verbessert. Allerdings würde ich es immer noch nicht als gut bezeichnen. Schließlich sehen sich nur etwas mehr als ein Drittel der CLUB 80-Mitglieder als eher aktives Mitglied.

Immerhin kann sich gut die Hälfte aller Mitglieder vorstellen, in einer Hard- oder Software-Gruppe mitzuarbeiten. Und fast genauso viele haben sich bereit erklärt, in nächster Zeit einen Artikel fürs Info zu schreiben. Wobei allerdings die Software, mit einer Themenvielfalt von B wie BASIC bis Z wie ZCPR, gegenüber der Hardware wesentlich stärker vertreten sein wird.

4. Erfahrungsaustausch

Sehr interessant ist meiner Meinung nach die Liste der Erfahrungen, über die die verschiedenen Clubmitglieder verfügen und die sie bereit sind, an andere weiter zu geben. Wichtig ist in diesem Zusammenhang, Leute, die bestimmte Probleme haben, mit den Leuten zusammenzubringen, die eventuell eine Lösung dafür kennen. Dafür eignet sich meiner Meinung nach das Info besonders gut.

Wenn also jemand ein Problem mit seinem Rechner oder seiner Software hat oder an einem Projekt arbeitet, bei dem er Unterstützung braucht oder gerne annehmen würde, sollte er dies im Info veröffentlichen. Bei der Fülle von Erfahrungen, die die Mitglieder des CLUB 80 in teilweise langjähriger Beschäftigung mit Computern gesammelt haben, sollte es doch möglich sein, eine Lösung für ein Problem oder einen Partner für ein Projekt zu finden!

Eine gezieltere Suche ist natürlich über die Mitgliederkartei, in der praktisch alle Informationen aus der Fragebogenaktion gespeichert sind, möglich. Da es aus Datenschutzgründen meiner Meinung nach nicht ratsam ist, die Datei über DFÜ allgemein zugänglich zu machen, muß zur Zeit eine Suche über mich oder Jens laufen. Dazu genügt ein kurzer Anruf oder ein Brief - Antwort kommt prompt!

5. Vorschläge zum Club-Leben

In der Rubrik "Vorschläge zum Club-Leben" wurden einige interessante Vorschläge gemacht, deren Umsetzung sich aber leider teilweise als schwierig erweist. So wird der Vorschlag, "in der FIDO-Area CPM.GER öfters mal zu diskutieren bzw. diese zu beleben" wohl daran scheitern, daß das Interesse der Clubmitglieder an DFÜ leider immer noch relativ gering ist. Daran liegt es auch, daß das CLUB80-Brett in der Diskovery bisher leider nur von recht wenigen zum Informationsaustausch genutzt wird (wobei wohl auch die recht hohen Telefonkosten bei relativ niedrigen Übertragungsraten im Ferntarif eine gewisse Mitschuld trägt).

Andere Vorschläge sind dagegen praktisch schon umgesetzt (z.B. CP/M-Utilitie-Sammlung bei Fritz Chwolka) oder werden bei nächster Gelegenheit aufgegriffen und verwirklicht (z.B. Veranstaltungen für Familienmitglieder bei Clubtreffen, mehr Aktivitäten in Richtung ZCPR 3 usw.). Allerdings wird dazu vor allem auch die Mitarbeit der Mitglieder notwendig sein, ohne die z.B. der Vorschlag, mehr Regionaltreffen durchzuführen, nicht umsetzbar ist.

6. Clubtreffen

Womit wir beim Thema Clubtreffen wären. Hier würden immerhin 30 Leute die Anreise zu einem bis zu 200 km entfernten Treffen nicht scheuen. 19 Mitglieder würden auch bis 400 km, 13 bis 600 km und noch 10 über 600 km anreisen. Die mäßige Beteiligung beim letzten Treffen im Frühjahr dieses Jahres ist denmach wohl auf die Wahl des Termins zurückzuführen. Ich hoffe, daß wir damit im nächsten Jahr mehr Glück und damit eine erfreulichere Beteiligung haben.

Wichtig ist meiner Meinung nach auch die Organisation von Regionalen Treffen, für deren Organisation sich immerhin 8 Club-Mitglieder zur Verfügung stellen würden. Die Initiative für ein solches Treffen muß von diesen Mitgliedern ausgehen, auch wenn sie natürlich jederzeit mit Unterstützung durch den Vorstand rechnen dürfen! Interessante Themen für die Treffen gibt es auf jeden Fall, wie die Liste der Workshops, für die Interesse bekundet wurde bzw. für die sich Vortragende zur Verfügung stellen würden, beweist.

7. Club-Info

Last but not least zum Thema Club-Info, das ich allerdings zum größten Teil unserem Redakteur Jens Neueder überlassen werde. Nur zwei Dinge will ich kurz ansprechen.

Dabei handelt es sich erstens um die Frage "Hoch- oder Querformat", die durch die Mitgliederbefragung klar entschieden wurde. Während sich nur 6 Mitglieder für das Hochformat aussprachen, entschie- den sich 32 Info-Leser für das Querformat. Damit ist zunächst einmal eine Formatänderung vom Tisch - das Info bleibt, zumindest im Format, wie´s ist!

Zweitens wird immer wieder die Frage aufgeworfen, ob Beiträge über MSDOS und andere Betriebssysteme (z.B. LINUX) bzw. Rechner veröffentlicht werden sollten. Hierzu möchte ich nochmals mitteilen, daß im CLUB 80 (fast) nichts zensiert wird. Ob ein Mitgied einen Artikel zu MSDOS, LINUX, einem ATARI ST oder einem C 64 schreibt - er wird veröffentlicht. Allerdings sollte das Hauptziel des CLUB 80, nämlich die Unterstützung der Z80- und CP/M-Rechner, nicht aus dem Auge verloren werden!

So, das war´s. Den "Rest" zum Thema Clubinfo überlasse ich wie gesagt Jens.

Weiterhin viel Spaß beim Computerhobby,

PS: Nachträge zum Vorwort

Jetzt sind mir doch tatsächlich noch ein paar Sachen eingefallen, die eigentlich ins Vorwort gehören und die ich vergessen habe.

III. WDR-Grafik-Files
In seinem Artikel "WDR-Grafik-Files ..." im Info 39 hat Egbert Schröer alle MSDOSen-Besitzer dazu aufgerufen, Grafiken vom PC- ins WDR-Format zu übertragen. Da ich eigentlich grundsätzlich ein fauler Mensch bin, habe Alexander Schmid dazu angeregt, statt dessen unter CP/M ein Programm zu schreiben mit dem man das auf PC´s sehr verbreitete PCX-Format lesen kann. Das Ergebnis seiner Bemühungen sollte eigentlich schon in diesem, spätestens aber im nächsten Info zu finden sein.

IV. Compiler-Grundlagen
Harald Sonnemann hat in seinem FzAdM angegeben, daß ihn "einfach mal so interessieren" würde wie ein Compiler funktioniert. Als ich diesen Satz las, erinnerte ich mich dunkel daran, in einer Zeitschrift mal einen Grundlagen-Artikel zu diesem Thema gelesen zu haben. Nach einigem Suchen in meinem Archiv hat sich der Beitrag angefunden.

Es handelt sich dabei um einen Artikel aus der PASCAL international (die übrigens schon seit ein paar Jahren nicht mehr existiert) vom August 1987. Die Redaktion hatte damals die Idee, einen Selbstbau-PASCAL-Compiler zu veröffentlichen. Aus dieser Serie von insgesamt 6 Artikeln wird im Clubinfo nur der erste veröffentlicht. Die Kopieen der restlichen Artikel und die Diskette mit den Quelltexten zu dem Projekt können bei mir angefordert werden.

V. FIDO / MAUS
Anfang Juli 1993 ist das FIDO durch eine "gewaltsame" Regionalisierung der Netze quasi in zwei Teile zerbrochen. Die Initiatoren dieser Zwangsregionalisierung wollten die bis dahin nach dem Sympathieprinzip geknüpften Verbindungen trennen und nach Bundesländern wieder zusammenstricken. Das hat (leider) nur zum Teil geklappt und dazu geführt, daß es neben dem FIDO nun ein sogenanntes FIDO-Classic gibt.

Mein SysOp (der Betreiber meiner FIDO-Mailbox) hat sich den Konterrevoluzionären (der Ausdruck wurde dabei wirklich gebraucht und zeigt, wie ernst es den verantwortlichen Leuten ist) angeschlossen und ist im FIDO-Classic zu finden. Das hat zur Folge, daß ich z.B. Hans-Martin Stephan (ein ehemaliges CLUB 80-Mitglied), dessen SysOp sich den Revoluzern angeschlossen hat, nicht mehr per Net-/Crash-Mail erreichen kann. Zum Glück gibt es inzwischen wenigstens für einige Areas Verbindungen zwischen den beiden FIDO-Ablegern.

Vielleicht wäre ja das MAUS-Netz eine Alternative!? Dabei handelt es sich um ein auf den deutschsprachigen Raum Europas begrenztes Netz, welches im Gegensatz zum FIDO demokratisch geführt wird (im FIDO sind z.B. die NetCoordinators nicht von "unten" gewählt sondern von oben bestimmt). Nachteil von MAUS ist die gegenüber FIDO relativ geringe Verbreitung. So wäre für mich der nächste Knotenrechner erst in Stuttgart. Ich werde aber mal versuchen, für das nächste Info etwas mehr über das MAUS-Netz herauszufinden.

VI. Kostenloses Modem
Kürzlich habe ich mich mit einem Brancheninsider über BTX unterhalten und folgende Info erhalten. Unter der Telefonnummer 0130/808606 (kostenfrei) ist eine Interessengemeinschaft von BTX-Anbietern zu erreichen, die einen Verleihservice für Modems anbieten.

Ein Anruf bei den Leuten brachte folgendes Ergebnis:

1. der Service beinhaltet ein 1200-Baud-Modem incl. Softwaredecoder (für MSDOSen) und dem benötigten Anschlußmaterial.

2. die Anmeldung für den BTX-Dienst der TeleCom erfolgt durch die Interessengemeinschaft. Dadurch entfällt die Anmeldegebühr von 50,-- DM, die bei der Anmeldung normalerweise fällig wird.

3. die einzigen Gebühren, die anfallen, sind die BTX-Grundgebühr von 8,-- DM monatlich. Die vertelefonierten Einheiten und die Gebühren für die benutzen Seiten/Leistungen (z.B. 0,30 DM für die Benutzung des elektronischen Telefonbuchs der TeleCom) müssen selbst-verständlich ebenfalls bezahlt werden! Sie werden über die normale Telefonrechnung abge-rechnet.

4. das Modem kann jederzeit zurückgegeben werden. Auch dabei entstehen, außer dem Porto, keine Kosten.

5. ob das Modem für andere Zwecke (z.B. Anwahl einer Mailbox) "mißbraucht" werden kann, konnte mir die nette Dame leider nicht mitteilen. Einen Versuch wäre es sicher Wert.

6. das Ganze ist ein Werbegag der BTX-Anbieter, die sich durch mehr BTX-Nutzer mehr Gewinn erhoffen und den Zugang zum BTX daher möglichst günstig gestalten wollen.

7. leider habe ich vergessen zu fragen, ob die Aktion zeitlich begrenzt ist.

Mehr dazu kann ich euch vielleicht im nächsten Info berichten, wenn ich mein Leihmodem ausprobiert habe.

VII. Postfach-Adresse
Sowohl im Impressum als auch in der Adressenliste ist meine Hausadresse abgedruckt. Das soll auch so bleiben, schließlich habe ich keine Angst davor, daß mich mal jemand anruft oder besucht.

Postsendungen sollten aber, wenn möglich, an mein Postfach geschickt werden! Vor allem wenn es sich um Disketten oder anderes wichtiges oder empfindliches Material handelt. Aus diesem Grund hier nochmal meine vollständige Postfachadresse:

CLUB 80 - Hartmut Obermann, Postfach 1430, 89304 Günzburg

Man beachte die, gegenüber der Hausadresse unterschiedliche, Postleitzahl!

So, daß wars jetzt aber wirklich!

Allways good Hacking, H.O.!

Clubinfo

Liebe Club'er, auf diesen Seiten möchte ich gleich zum Fragebogen-Thema Clubinfo "Rede & Antwort" zu stehen.

Zuersteinmal möchte ich mich bedanken, daß so viele das Querformat akzeptiert haben, und ich somit keine zusätzlichen Arbeiten mit dem INFO habe. Wie aus der Auswertung ersichtlich bleibt es bei dem Querformat des INFOs.

Zu den von Euch genannten Mängeln am INFO möchte ich in Kürze folgendes anmerken:
Die Erscheinungsweise wurde aus bekannten Gründen auf vier Ausgaben im Jahr festgelegt (Info39/03).
Die Adresse der Redaktion ist, wie die der anderen Vorstandsmitglieder im Impressum genannt. Als E-Mail-Adresse kann die Discovery verwendet werden. Ich bin ab und zu in der Discovery. **Auf alle Fälle spätestens kurz vor dem Redaktionsschluß.** Nachrichten/Artikel dann bitte direkt an mich privat adressieren. Der Klaus hilft Euch sicher weiter, wenn Ihr Probleme haben solltet. Durch ein entsprechend schnelles Modem kann ich auch größere, ganz und suupergroße Artikel recht schnell downloaden. Ihr braucht Euch also (bis auf die Begrenzung Eurer Telefonrechnung) keinem Zwang unterlegen.
Eine Bitte betreffend der Artikel per DFÜ/Diskette habe ich hier noch, da ich unter WORD arbeite, wäre mir natürlich ein Textfile in entsprechendem Format am liebsten. Wer dies nicht kann, sollte seinen Text als ASCII-File ohne [CR] an jedem Zeilenende **[CR] bitte nur am Ende eines Absatzes!** an mich senden. Als Bilddatei kann ich eigentlich mit jedem DOS-Format (z.B.: TIFF, GIF, JPG, BPM, ...) arbeiten.
Die Autoren der Beiträge sind, sofern bekannt, im Inhaltsverzeichnis des jeweiligen INFOs immer im Klartext zu finden. Sind mehrere Artikelverzeichnisse direkt übereinander, sind sie vom gleichen Autor.
Die Mitgliederliste ist auf Euren Wunsch hin neu überarbeitet worden, leider hat sie nicht viel mehr Informationsgehalt wie vorher, da die Meisten, wenn überhaupt, ja doch nur ihre private Erreichbarkeit angegeben haben. Einziger Gewinn ist hier die Rechnerliste. **Wichtig auf alle Fälle ist, daß, sollten sich Änderungen ergeben, die Mitglieder Ihre Daten unverzüglich aktualisieren. Dazu Nachricht an Hartmut oder mich. Sonst ist die beste Dateiverwaltung doch nur Datenmüll auf der Platte!** Es hat viel Mühe gekostet, die Datei wieder zu bereinigen, deshalb seit bitte hilfreich sie zu pflegen!

Einige von Euch fanden die endlosen Listings und seitenlangen Mailboxmitschnitte etwas störend. Ich möchte hier zukünftig folgendes vorschlagen: Listings und Mailboxmitschnitte, die 4 Einzel-Seiten überschreiten sollen nur kurz beschrieben(Extrakt mit Hinweis auf das entsprechende File) werden. Der/das ausführliche Text/Listing wird dann in der Discovery abgelegt. Dies hat auch den Vorteil, daß die Listings bei Weiterverwendung nicht erst lange abgetippt werden müssen. Ich hoffe, es ist im Sinne der Autoren und User.
Für die Autoren und solche die es werden wollen, wird im Dezember-Info eine "Anleitung" erscheinen, in der ich einige Tips zum Format, Schriftgröße usw. geben werde. Ich hoffe dadurch den Sachen wie: zu kleine Schrift, mehr Rand für Lochung Abhilfe zu schaffen. Wir hatten früher schon einmal eine Anleitung aufgelegt, diese wird gerade aktualisiert.
Weiterhin wurde von den Autoren beklagt, daß es auf die Artikel eigentlich nie einmal ein "Feedback" gibt, nicht einmal in privater Korrespondenz. Ich finde dies schade und auch für die Autoren demoralisiernd. Das es daran liegt, daß die Themen so danebenliegen glaube ich nicht. Sicher liegt es mehr daran, daß jeder nur konsumiert. Als Beispiel die BASIC-Ecke: Nach diversen Aufrufen habe ich gerade Einen gefunden, der noch Interesse hätte. Mit dem, selber auch gut drauf in BASIC, komme ich auch "privat" klar. Um uns auszutauschen ist das INFO eigentlich nicht erforderlich. Nach Auswertung der Fragebogen aber, gibt es viele, die noch Interesse an der BASIC-Gruppe hätten. Meine Bitte, meldet Euch zu den Themen zu Wort, nur so entsteht eine Diskussion und ein lebendiges Clubinfo.
Zum Abschluß, weils gerade so passt, zu den Wünschen für zukünftige INFOs. Wir freuen uns natürlich über jeden, der dem Info beiseitesteht. Für die unter Wünsche genannten Themen muß sich nur jemand bereiterklären mitzuarbeiten. Natürlich können wir auch -wie gewünscht- einige "Oldies" wieder aufleben lassen. Bei der Menge an schon behandelten Themen fällt die Wahl schon schwer. Sicher ist einiges von "Früher" auch schon total überholt.

Nochmals meinen Dank für Eure Mitarbeit,
-man liest sich in der INFO-
die Redaktion.

Termine... Termine... Termine... Termine... Termine

SYSTEMS	München	18.10.- 22.10.93
Hobbyelektronik	Stuttgart	03.11.- 07.11.93
PRODUKTRONICA	München	09.11.- 13.11.93
Nord Elektro	Hamburg	18.11.- 20.11.93

Redaktionsschluß für das nächste Clubinfo ist der

01. Dezember 1993.

Zunächst möchte ich mich kurz vorstellen:

Mein Name ist Jörg Linder. Ich bin 23 Jahre alt und wohne in Seelow, einer Kleinstadt im äußersten Osten Deutschlands.

Ursprünglich habe ich mal Elektronikfacharbeiter gelernt, aber damit kann man hier nicht mehr viel anfangen. Deshalb mache ich zur Zeit eine Umschulung zum Bürokaufmann.

Dabei habe ich auch relativ oft mit Computern zu tun. Mein Interesse für diese "Kisten" war aber schon Jahre zuvor durch die damaligen Kleincomputer (kurz KC) aus der DDR geweckt worden. Zwar bin ich beruflich mehr mit den DOSen beschäftigt, aber privat arbeite ich mit meinem KC 85/4 (Z80-Rechner mit CP/M) - da ist alles noch so schön einfach und verständlich!

Im Moment bin ich damit beschäftigt, alles notwendige für DFü zusammenzusammeln. Von Alexander Schmid habe ich auch schon ein paar nützliche Tips erhalten. Ich hoffe, daß ich bald einsteigen kann.

Ansonsten interessiert mich besonders Programmierung in TurboPascal und/oder C. Auf dieser Strecke habe ich noch keinerlei Erfahrung und wäre für Tips und Anregungen dankbar. Bisher konnte ich mich nicht dazu durchringen, eine Programmiersprache zu favorisieren. Wenn mir dabei jemand eine Entscheidungshilfe geben könnte (für und wider beider Sprachen)?!

Doch nun möchte ich meinen Rechner vorstellen:

Wie bereits erwähnt, habe ich einen KC 85/4 aus DDR-Produktion. Er war das letzte Produkt einer langjährigen Reihe (es gab auch /1, /2 und /3). Unter den Homecomputern der DDR war er sozusagen "High-End". Ich habe dazu den Floppy-Aufsatz, wodurch man zu einem CP/M-kompatiblen Betriebssystem kommt.

Das Grundgerät ist ein eigenständiger Rechner mit 20 kB ROM und 2x64 kB RAM, wovon 64 kB für den Bildspeicher genutzt werden. Mit dem Floppy-Aufsatz erhält man einen zweiten Rechner, der über einen KoppelRAM mit dem Grundgerät kommuniziert bzw. umgekehrt. Damit sind zwei Betriebsarten möglich. Einmal die normale KC-Betriebsart, bei der der Floppy-Aufsatz im Hintergrund arbeitet und so die Diskettenarbeit ermöglicht (sonst nur Kassette). In der zweiten Betriebsart arbeitet man unter einem CP/M-kompatiblen Betriebssystem, bei der das Grundgerät nur als Terminal dient.

Das Betriebssystem nennt sich "MicroDos". Es ist zu CP/M Version 2.6 kompatibel und angeblich in Moskau entwickelt worden (wenn das mal nicht irgendwo geklaut wurde!). Diese Version ist so ein Zwischending von 2.2 und 3.0. Der Funktionsumfang des BDOS wurde bereits geringfügig erweitert und der CCP ist gegenüber 2.2 etwas komfortabler.

Durch diesen "Luxus" schrumpft allerdings der verfügbare Speicher auf zirka 50 kB TPA. Zwar sind alle Funktionen des Grundgerätes verfügbar (z. B. Farbgrafik), aber für manche Anwendungen reicht der TPA einfach nicht aus.

Ich versuche seit einiger Zeit, das Betriebssystem zu reassemblieren, um dann unnötige Sachen rauszuschmeißen. Inzwischen glaube ich aber, daß das eine Lebensaufgabe wird. Ein Kenner wäre eine echte Hilfe!

Nun noch kurz zu den Laufwerken. Standardmäßig benutze ich ein Format mit 780 kB, kann aber auch andere Formate lesen (nicht immer problemfrei). Am liebsten arbeite ich jedoch mit Laufwerk A. Unter MicroDos wird der RAM des Grundgerätes für die RAM-Floppy A: benutzt. Dort kann man mittels Modulen über mehrere MB (!) verfügen (bei mir sind meistens 1,3 MB gesteckt).

Ich denke, daß es eine recht ausführliche Vorstellung war. Wenn jemand mehr über das System wissen möchte, kann er sich jederzeit an mich wenden.

Jörg Linder
Küstriner Straße 68

15306 Seelow

Austausch von Grafiken DOS -> CP/M

Nachdem sich das WDR-Format (siehe Artikel von Hartmut Obermann im Info Nr. 25) leider oder glücklicherweise, je nach Standpunkt, nicht so recht durchsetzen konnte und durch die verschiedenen Standardformate der AMIGAS, ATARIs und PCs verdrängt worden ist, hat es im Club keinen ernsthaften Anlauf mehr gegeben, einen anderen Grafikstandard auf unseren Rechnern verfügbar zu machen. Die vielen extrem hochauflösenden Bildchen mit noch mehr Farben wie VGA mit True Color usw. kann man wohl von vornherein ausschließen und so bietet sich, besonders für kleinere schwarz/weiße Clip-Arts, das sogenannte PCX-Format an. Es ist weit verbreitet und basiert auf einer einfachen Lauflängencodierung, d.h. mehrere gleichartige Bytefolgen werden durch ein Byte und einen Wiederholungszähler ersetzt, und ist somit leicht zu decodieren. Die Idee und das Grundgerüst zu folgendendem Programm hatte ich mir schon vor längerer Zeit aus der C-Ecke im FIDO geholt, aber erst jetzt hat mich Hartmut darauf gebracht, daß man sowas doch gefälligst im Clubinfo bringen sollte. An Grafikbefehlen braucht man lediglich das Löschen des Bildschirms und das Setzen eines Punktes, sodaß das Programm wohl auf jedem grafikfähigen Rechner laufen sollte. In der Struktur PCXHEADER ist der Aufbau des Headers beschrieben und es dürfte kein großes Problem sein, die wichtigen Daten bei Bedarf auch mit einem BASIC-Programm zu lesen und das Programm umzusetzen. Wer zu faul zum Abtippen ist, kann das Programm nebst Hi-Tech-C (ist seit kurzem Public Domain, siehe ein anderer Artikel) von mir bekommen oder es aus der DISCOVERY downloaden.
Wenn ich den Redaktionsschluß jetzt nicht verpennt hätte und den Artikel als Nachtschicht schreiben müßte, hätte ich das Programm noch so erweitern können, daß man auch selber PCX-Files erstellen kann, aber so kann es wenigstens noch einen Fortsetzungsartikel geben und außerdem ist das Angebot von PCX-Bildchen aus dem DOS-Sektor wohl groß genug, daß Ihr mit dem Ansehen vorläufig genug zu tun habt :-)

```
/* ------------------------------------------------------------------------ */
/* dispcx.c    -   Read & Display a *.PCX graphics file in Microsoft's      */
/* Quick C in EGA and in Borland's Turbo C 2.0 and equivilents with         */
/* all supported BGI graphics modes (VGA/EGA/CGA/Herc).                     */
/* ------------------------------------------------------------------------ */
/* Original by                         :  M. CLYNES   07-24-89              */
/* Borland Compiler Enhancements by :  J. Braatz   02-16-92                 */
/* Adapted to Club-80 Terminal by     :  A. Schmid   06-07-93               */
/* ------------------------------------------------------------------------ */

#include <stdio.h>
#include <conio.h>
#include <grafmod2.h>

#define HOROFF   0  /* horizontaler Offset */
#define TOPOFF   0  /* vertikaler Offset   */

char inversfl='N';  /* Bild normalerweise nicht invertieren */

struct pcxheader{
        char manufact;       /*  0      Hersteller-ID              */
        char version;        /*  1      Versionsnummer             */
        char encode;         /*  2                                 */
        char bpp;            /*  3                                 */
        int xmin;            /*  4, 5   Bildgroesse                */
        int ymin;            /*  6, 7       "                      */
```



```
        int xmax;            /*  8, 9       "                       */
        int ymax;            /*  10,11      "                       */
        int hres;            /*  12,13  Aufloesung der Hardware     */
        int vres;            /*  14,15      "      "     "          */
        char colormap[48];   /*  16-63                              */
        char reserved;       /*  64     momentan reserviert         */
        char ebene;          /*  65     Zahl der Farbebenen         */
        int bpline;          /*  66,67  Byte pro Zeile je Ebene     */
        int paletinfo;       /*  68,69  Art des Bildes              */
        char blank[58];      /*  70-127 derzeit keine Informationen */
        } header;

/*------------------------------------------------------------------------*/
/*   gobal decoded data    char decodebuff[200];                          */
/*------------------------------------------------------------------------*/
/* Decode a line of .PCX                                                  */
/* This procedure reads one encoded line from the image file             */
/* 0 = valid data stored                                                  */
/* EOF = out of data in file                                              */

decodepcx(decodebuff, bytesline, fp)
char *decodebuff;                                  /* where to place data */
int bytesline;                                     /* # of bytes per line */
FILE *fp;                                          /*  image file handle  */
{
  int data;
  int cnt;

  while(bytesline > 0 ){
    cnt = 1;
    if(EOF == (data = getc(fp))) return(EOF);   /* retrive a data byte */
    if(0xc0 == (0xc0 & data)){
      cnt = 0x3f & data;                        /* get repeat count */
      if(EOF == (data = getc(fp))) return(EOF);  /* get real data */
      }

    while(cnt--){                         /* expand data into buffer */
      *decodebuff = data;
      ++decodebuff;
      --bytesline;
      }
    }
  return(0);                                     /* file read status */
  }
/* decodepcx()  end */


/*------------------------------------------------------------------------*/
/*                     read .PCX header into structure                    */
/*------------------------------------------------------------------------*/
readheader(fp)
FILE *fp;
{
  int count;
  char *loader;

  loader = (char *) &header;

  for(count = 0; count < 127; ++count){
     *loader = (char)fgetc(fp);
     ++loader;
     }
```

```
/* Note: For Borland Compiler users, the following will need to be      */
/* changed to outtext() if you want to see the header information on    */
/* Hercules monitors (with the old herc.bgi driver)                     */

  printf("Encoding type %d\n",header.encode);
  printf("Window X-min = %4d,  Y-min = %4d\n",header.xmin, header.ymin);
  printf("Window X-max = %4d,  Y-max = %4d\n",header.xmax, header.ymax);
  printf("Hres = %4d,  Vres = %4d\n",header.hres, header.vres);
  printf("# bytes per/scan line %4d   \n\n",header.bpline);
  printf("Bild (i)nvers oder (n)ormal einlesen ? ");
  inversfl=getch() & 0x5f;

  return(header.bpline);
}

/*----------------------------------------------------------------------*/
/*                        pixel line dump                               */
/*----------------------------------------------------------------------*/
/* DUMP A RASTER LINE TO DISPLAY                                        */
/* pixeldump((pointer buffer), (# of BYTES), (Y line on screen to dump)) */
/*----------------------------------------------------------------------*/

pixeldump(rdbuff, bytes, line)
char *rdbuff;
int bytes, line;
{
  unsigned x;
  unsigned char byte, loop;

  bytes = (bytes * 8) + HOROFF;

  if (header.ymax>YMAX-TOPOFF){ /* in X-Richtung komprimieren */
    for(x = HOROFF; x < (bytes>>1); ++rdbuff){
      if (inversfl=='I')
        byte = ~(*rdbuff);
      else
        byte = *rdbuff;
      for (loop = 0x80; loop > 0; loop >>= 1){
        if (loop & byte) plot(x, YMAX-line);
        loop >>= 1;
        if (loop & byte) plot(x, YMAX-line);
        ++x;
        }
      }
    }
  else{
    for(x = HOROFF; x < bytes; ++rdbuff){
      if (inversfl=='I')
        byte = ~(*rdbuff);
      else
        byte = *rdbuff;
      for (loop = 0x80; loop > 0; loop >>= 1){
        if (loop & byte) plot(x, YMAX-line);
        ++x;
        }
      }
    }
  }
/* pixeldump() End */
```



```
/*------------------------------ MAIN() ------------------------------*/

int main( argc, argv )
int argc;
char *argv[2];
{
  int line, status, bytesline;
  char decodebuff[200];
  FILE *fp;

  color=1;
  argc = argc;                          /* Nuke's Compiler warning */

  if((fp = fopen(argv[1],"rb")) == NULL){
    printf("ERROR: FILE '%s' CANNOT BE OPENED.\n",argv[2]);
    exit(0);
    }
  bytesline = readheader(fp);
  line = TOPOFF;
  status = 0;
  clsh();  /* Grafik loeschen   */
  hires(); /* Grafik einblenden */

  while(status != EOF){    /* do EXTRA READS to srink vertical */
    if (header.ymax>(YMAX-TOPOFF)*2){     /* sehr grosses Bild */
      status = decodepcx(decodebuff, bytesline, fp);
      pixeldump(decodebuff, bytesline, line);
      }
    if (header.ymax>YMAX-TOPOFF){          /* grosses Bild */
      status = decodepcx(decodebuff, bytesline, fp);
      pixeldump(decodebuff, bytesline, line);
      }
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    ++line;
    }

    getch();
    textmode(); /* normalen Textbildschiem einschalten */
  }
/* End of main() */
```

Alexander Schmid

Grafik des Genie IIIs unter Holte CP/M+

Volker Dose, Egbert Schröer, Juli 1993

Die Grafikmöglichkeiten des Genie IIIs unter Holte CP/M+ zu nutzen wird eine kleine Artikelserie beschreiben. Dies ist in den Programmiersprachen Turbo Pascal 3.0, ANSI C und Assembler realisiert worden. Beginnen möchten wir mit einem Assembler-Programm, das die Möglichkeit bietet Grafik aus dem Speicher auszulesen und als File zu schreiben und diese wieder einzulesen. Ausgiebig Gebrauch wird dabei von der USERF Funktion gemacht.

```
;*****************************************************************
;*                    H R G . C O M                              *
;*****************************************************************
;*                                                               *
;* Programm um HRG-Bilder des Genie IIIs von Diskette zu laden   *
;* und zu speichern.                                             *
;* Der Bildinhalt wird mit einer Systemroutine des Bios in den   *
;* HRG-Speicher des G IIIs übertragen.                           *
;*                                                               *
;*****************************************************************
;*      Volker Dose, Egbert Schröer     Februar-Juli 1993        *
;*****************************************************************

; Das Programm muß zunächst an den jeweiligen Monitor
; angepaßt werden !
; Die Anzahl der Scanzeilen und die Anzahl der angezeigten
; Zeilen muß hier eingetragen werden.

scanzeilen      equ     11
zeilenzahl      equ     25
maxy            equ     scanzeilen*zeilenzahl

;BIOS
;BDOS

bel     equ     07h
lf      equ     0ah
cr      equ     0dh
esc     equ     1bh
clrscr  equ     1ah             ;Bildschirm löschen

bdos    equ     0005h
wboot   equ     0000h

userf   equ     30

prtstr  equ     9
open    equ     15
close   equ     16
searchf equ     17
next    equ     18
write   equ     21
delete  equ     19
makef   equ     22
parse   equ     152
read    equ     20
multio  equ     44
```



```
adrdma  equ     26
grafik  equ     25
fnein   equ     01h
fnaus   equ     02h

fcb     equ     5ch
;--------------------------------
        org     100h            ;Start der TPA
;--------------------------------
;       Hauptprogramm
;--------------------------------
start   ld      de,hello        ;Begrüßung ausgegeben
        ld      c,prtstr        ;mittels BDOS-Call
        call    bdos            ;Auswahl Laden/Speichern anbieten
echo    ld      c,fnein         ;Funktionsnummer für Eingabe von
                                ;der Tastatur laden
        call    bdos            ;Zeichen von der Tastatur ins
                                ;A-Register holen
        sub     a,20h           ;in Großbuchstabe umwandeln
        ld      e,1             ;Zeichen in Parameterregister
                                ;bringen
        cp      'L'             ;Was will Sie/Er denn ?
        jr      nz,weiter
        ld      de,frage3
        ld      c,prtstr
        call    bdos
        ld      c,fnein         ;Welche HRG Seite ?
        call    bdos            ;
        ld      (hrgsei),a      ;speichern
        call    liesstr         ;Filenamen holen
        call    loadhrg

ende    call    graoff          ;Nun machen wir die Grafik
                                ;wieder aus
        jp      wboot           ;Rücksprung ins Betriebssystem
                                ;über Warmstart
weiter  cp      'S'
        jr      nz,ende         ;Was hat'er nu' gedrückt ?
        ld      de,frage3
        ld      c,prtstr
        call    bdos
        ld      c,fnein         ;Welche HRG Seite ?
        call    bdos            ;
        ld      (hrgsei),a      ;speichern
        call    liesstr         ;Filenamen holen
        call    savehrg

        jr      ende            ;Programm beenden

;----------------------------------------
;       Unterprogramme
;----------------------------------------

;----------------------------------------
;       Eingabe Filename
;----------------------------------------

liesstr ld      de,frage2
        ld      c,prtstr
```

```
        call    bdos
        ld      hl,fname        ;Zeiger auf Speicher für Name
lies    ld      c,fnein
        push    hl              ;Zeiger sichern
        call    bdos            ;Zeichen von Tastatur holen
        pop     hl              ;Zeiger restaurieren
        ld      (hl),a          ;Zeichen in Speicherbereich
                                ;ablegen
        inc     hl              ;auf nächsten freien Speicher-
                                ;platz zeigen
        cp      cr              ;Zeichen mit Endmarkierung ver-
                                ;gleichen
        jp      nz,lies         ;Lesen, bis Endmarkierung gelesen
                                ;wurde
        ld      hl,fname        ;Zunächst wird der Filename des
        ld      de,nambuf       ;Bildes gesichert
        ldir                    ;Speichertransfer
        ret                     ;Zurück zur aufrufenden Stelle

;----------------------------------------
;       Aufruf über USERF 30
;----------------------------------------
system: push    bc              ;zunächst BC sichern
        ld      ix,(wboot+1)    ;(IX) Start der BIOS Jump Table
        ld      bc,3*(userf-1)  ;Offset zum JP USERF
        add     ix,bc           ;dazu addieren
        pop     bc              ;USERF-Nr. wieder zurück
        jp      (ix)            ;Sprung nach USERF und zurück zum
                                ;Aufrufer von SYSTEM
;--------------------------------
;       L O A D H R G
;--------------------------------
loadhrg ld      bc,0            ;zunächst wird die Länge des
                                ;Filenames geholt
        ld      hl,nambuf       ;Start der Zeichenkette
loop    inc     c               ;C einen weiterzählen
        inc     hl              ;Zeiger auch
        ld      a,(hl)          ;Ist das Terminierungszeichen
                                ;erreicht ?
        cp      ':'             ;ist ein Drive angegeben?
                                ;das darf nicht sein !
        jp      z,error
        cp      00h             ;dann stimmt BC
        jr      nz,loop         ;Schleife durchlaufen bis Schluß
        ld      a,(hrgsei)      ;Welche Seite soll gespeichert
                                ;werden ?
        cp      '0'             ;ist es Null ?
        jr      z,goon          ;dann zunächst nur speichern
        cp      '1'             ;oder Seite 1 ?
        jp      nz,error        ;Falsche Eingabe der Seiten #

goon    ld      (seite),a       ;In Meldung unterbringen
        ld      de,schlu        ;Jetzt wird eine Meldung ausgegeben
        ld      c,prtstr        ;mittels BDOS-Call
        call    bdos
        ld      de,schlu2
        ld      c,prtstr
        call    bdos
```



```
        call    graon           ;Grafik anknippsen
aufdsk  ld      de,pfcb         ;PARSE Filename will einen Control-
                                ;block sehen
        ld      c,parse         ;der Filename ist jetzt aufbereitet
                                ;worden
        call    bdos            ;jetzt wird geguckt, ob das File
                                ;schon existiert
        ld      de,fcb          ;File Control Block
        ld      c,searchf       ;suche Eintrag
        call    bdos
        cp      0ffh            ;Eintrag gefunden = File existiert
                                ;bereits ?
        jp       z,keins        ;FILE NOT EXIST ausgeben
holes   ld      de,fcb          ;jetzt soll das File von Disk
                                ;gelesen werden
        ld      c,open          ;es wird zunächst geöffnet
        call    bdos
        cp      00h             ;ist alles gut gegangen ?
        jp      nz,nixis        ;wenn nicht Fehlermeldung ausgeben
        ld      (fcb+32),a      ;cr-feld des FCB auf 0
        ld      (fcb+12),a      ;ext-feld auf 00
        ld      e,128           ;Multisektor I/O mit 16 K Blöcken
        ld      c,multio        ;Bdos-Call
        call    bdos
        ld      de,buffer       ;Zieladresse ist der Buffer
        ld      c,adrdma        ;Set DMA Adress
        call    bdos
        ld      de,fcb          ;jetzt werden 16 KB gelesen
        ld      c,read
        call    bdos
        cp      00h             ;alles gut gegangen ?
        jp      nz,nixis
        ld      e,128           ;wieder 16 KB schreiben
        ld      c,multio
        call    bdos
        ld      hl,buffer
        ld      bc,4000h        ;16kB dazuaddieren
        add     hl,bc
        ex      de,hl           ;DMA Adresse nach DE
        ld      c,adrdma        ;set DAM Adresse
        call    bdos
        ld      de,fcb
        ld      c,read
        call    bdos
        cp      00h
        jp      nz,nixis
gutgut  ld      de,fcb          ;das File schließen
        ld      c,close
        call    bdos            ;das wars dann!!!
trans   ld      a,(seite)       ;jetzt wird das Bild in den
                                ;HRG-Speicher übertragen
        sub     '0'             ;ASCII => binär umwandeln
        add     0f0h            ;Quelle ist der Buffer
        ld      hl,parablock
        ld      de,buffer
        ld      c,33            ;Systemfunktion Bereich kopieren
        call    system
        ret                     ;Zurück zum Hauptprogramm
```

```
;---------------------------------
;        S A V E H R G
;---------------------------------
savehrg ld      bc,0            ;zunächst wird die Länge des
                                ;Filenames geholt
        ld      hl,nambuf       ;Start der Zeichenkette
loop1   inc     c               ;C einen weiterzählen
        inc     hl              ;Zeiger auch
        ld      a,(hl)          ;Ist das Terminierungszeichen
                                ;erreicht ?
        cp      ':'             ;ist ein Drive angegeben?->Fehler
        jp      z,error
        cp      00h             ;dann stimmt BC
        jr      nz,loop1        ;Schleife durchlaufen bis Schluß
        ld      a,(hrgsei)      ;Welche Seite soll gespeichert
                                ;werden ?
        cp      '0'             ;ist es Null ?
        jr      z,goon1         ;dann zunächst nur speichern
        cp      '1'             ;oder Seite 1 ?
        jp      nz,error        ;Falsche Eingabe der Seiten #
goon1   ld      (seite),a       ;In Hello-Meldung unterbringen
        ld      de,schlu        ;Meldung ausgegeben
        ld      c,prtstr        ;mittels BDOS-Call
        call    bdos
        ld      de,schlu2
        ld      c,prtstr
        call    bdos
        call    graon           ;Grafik anknippsen
trans1  ld      a,(seite)       ;jetzt wird das Bild in die
                                ;TPA übertragen
        sub     '0'             ;ASCII => binär
        rrca                    ;Quellseite ins obere Nibble schieben
        rrca
        rrca
        rrca
        add     0fh             ;Ziel ist der Buffer
        ld      hl,parablock
        ld      de,buffer
        ld      c,33            ;Systemfunktion Bereich kopieren
        call    system
aufdsk1 ld      de,pfcb         ;PARSE Filename will einen
                                ;Controlblock sehen
        ld      c,parse         ;der Filename ist aufbereitet
        call    bdos            ;jetzt wird gekuckt, ob das File
                                ;schon existiert
        ld      de,fcb          ;File Control Block
        ld      c,searchf       ;suche Eintrag
        call    bdos
        cp      0ffh            ;Eintrag gefunden = File
                                ;existiert bereits ?
        jr      z,keins1        ;nicht ? dann weiter im Text
        ld      de,frage1       ;Frage 1 ausgeben
        ld      c,prtstr
        call    bdos            ;
        ld      c,01h           ;hole Zeichen von der Konsole
        call    bdos
        cp      'n'             ;wenn anderer Filename soll,
                                ;dann Abbruch
        jp      z,wboot
```



```
loefil  ld      de,fcb          ;sonst File löschen
        ld      c,delete
        call    bdos            ;und neues File erzeugen
keins1  ld      de,fcb          ;jetzt das File erzeugen
        ld      c,makef         ;
        call    bdos
        cp      00h             ;ist alles gut gegangen ?
        jp      nz,nixis        ;wenn nicht Fehlermeldung ausgeben
        ld      e,128           ;Multisektor I/O mit 16 K Blöcken
        ld      c,multio        ;Bdos-Call
        call    bdos
        ld      de,buffer       ;Quelladresse ist der Buffer
        ld      c,adrdma        ;Set DMA Adress
        call    bdos
        ld      de,fcb          ;jetzt wird echt geschrieben
        ld      c,write
        call    bdos
        cp      00h             ;alles gut gegangen ?
        jp      nz,nixis
        ld      e,128           ;wieder 16 KB schreiben
        ld      c,multio
        call    bdos
        ld      hl,buffer
        ld      bc,4000h        ;16kB dazuaddieren
        add     hl,bc
        ex      de,hl           ;DMA Adresse nach DE
        ld      c,adrdma        ;set DAM Adresse
        call    bdos
        ld      de,fcb
        ld      c,write
        call    bdos
        cp      00h
        jp      nz,nixis
        ld      a,(fcb+32)
        ld      (0010h),a
        ld      de,fcb          ;das File schließen
        ld      c,close
        call    bdos            ;das wars dann!!!
        ret                     ;Rücksprung zum Hauptprogramm

;---------------------------------------------------------------
;       Knippst den gewählten HRG Bildschirm des Genie IIIs an
;---------------------------------------------------------------
graon:  ld      a,(hrgsei)      ;hier steht die angegebene
                                ;Seitennummer
        sub     '0'             ;ASCII -> binaer
        cp      1               ;ist Seite 1 gefragt?
        jr      z,sei1          ;dann diese Seite einschalten
        cp      0               ;ist Seite 0 gefragt?
        ret     nz              ;wenn nicht 0 oder 1 raus
        di                      ;jetzt wird Seite 0 eingeschaltet
        in      a,(0f9h)        ;bit 4 von Sysport 0 entscheidet,
        res     4,a             ;welche von beiden Seiten
        out     (0f9h),a        ;selektier werden kann
        jr      on
sei1    di                      ;Grafik-Seite 1 ist gewünscht
        in      a,(0f9h)
        set     4,a
        out     (0f9h),a
```

```
on      ei
        ld      a,1                     ;A<>0  => Grafik an !
        ld      c,grafik                ;C enthält die USERF Nummer
        call    system                  ;Ruft USERFunktion auf und RET
        ret                             ;zurück zum Aufrufer von GRAON

;------------------------------------------------------
;       Knippst den HRG Bildschirm des Genie IIIs aus
;------------------------------------------------------
graoff: ld      a,0                     ;A=0  => Grafik aus !
        ld      c,grafik                ;C enthält die USERF Nummer
        call    system                  ;Ruft USERFunktion auf und RET
        ret                             ;zurück ins Betriebssystem

;--------------------------------
;       Fehlermeldungen
;--------------------------------
nixis   ld      de,mist                 ;es ist ein Fehler aufgetreten !
        ld      c,prtstr                ;Fehlermeldung ausgeben
        call    bdos
        jp      wboot
;--------------------------------
keins   ld      de,nichtda              ;Fehlermeldung ausgeben
        ld      c,prtstr
        call    bdos
        jp      wboot
;--------------------------------
error   ld      a,'$'                   ;Zunächst einen Teil ausgeben
        ld      (schlu),a               ;String dort terminieren
        ld      de,hello                ;den dann ausgeben
        ld      c,prtstr
        call    bdos
        ld      de,falsch               ;Fehlermeldung ausgeben
        ld      c,prtstr
        call    bdos
        jp      wboot

;--------------------------------
;       Nachrichten an den User
;--------------------------------
nichtda defb    bel,'  Es ist kein File dieses Namens vorhanden.',cr,lf,'$'

falsch  defb    bel,'  Syntax: HRG-Seite 0 oder 1; filename.ext',lf,cr
        defb    '$'

hello   defb    clrscr,cr,lf,esc,'R','HRG.COM',esc,'S',' liest und schreibt'
        defb    ' HRG-Bilder im aktuellen Disk/User-Bereich.',lf,cr
        defb    '            Eingestellt sind 11 Scan-Zeilen und 25 Zeilen.',lf,cr
        defb    '            Nur lauffähig auf Genie IIIs mit Holte CP/M +!'
        defb    cr,lf,lf

wahl    defb    'Grafik ',esc,'R','l',esc,'S','aden oder',esc,'R','S',esc,'s'
        defb    'peichern ?',lf,cr
        defb    '$'

schlu   defb    'Das Bild mit dem Namen ',lf,cr

nambuf  defb    00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h,00h
        defb    '.'
```



```
        defb    '$'

schlu2  defb    cr,lf
        defb    'wird in die HRG Seite '
seite   defb    '0'
        defb    ' eingelesen. ',cr,lf,lf
        defb    '$'

mist    defb    bel,' Es ist ein Fehler aufgetreten. Abbruch !!','$'

frage1  defb    bel,'File dieses Namens existiert bereits. File löschen ? (J/N) '
        defb    '$'

frage2  defb    bel,cr,lf,'Filename: ',cr,lf
        defb    '$'

frage3  defb    bel,cr,lf,'Welche Seite ? ',cr,lf
        defb    '$'
;------------------------------------
parablock                               ;hier stehen die Parameter
                                        ;für die Systemroutine,
        defw    0000h                   ;hier Startpunkt X
        defw    0000h                   ;Startpunkt Y
        defw     640                    ;horizontale Kantenlänge
        defw    maxy+1                  ;vertikale Kantenlänge

pfcb    defw    nambuf                  ;Filename in ASCII-Format
        defw    fcb                     ;hier liegt der 'wahre' FCB

hrgsei  defs    1
fname   defs    12

buffer  equ     $                       ;Buffer für das Bild !

        end     start
```

Model 4 intern: Dem Interpreter aufs Bit geschaut

Naturgemäß wendet sich dieser Beitrag in erster Linie an Assembler-Programmierer und solche, die es werden wollen. Doch heiße Tips gibt es auch für Basic-Fans, zum Beispiel wie der List-Schutz funktioniert.

Zuerst: Viele Routinen aus dem Level2-Basic findet man auch hier wieder. Der Arithmetik-Teil wurde fast unverändert übernommen, nur kräftig gemischt und im Adreßbereich verschoben. Kleine Verbesserungen sind festzustellen, die große Schwachstelle, reiner 8080-Code, blieb. In Listing 1 sind die wichtigsten Unterprogramme aufgeführt. Wo WRA1 und WRA2 hingewandert sind, können sie aus Listing 2 ab 6664 (Hex) entnehmen. Sofern nicht anders vermerkt, sind alle Werte Hex-Zahlen.

Bei der Entwicklung von Ausdrücken benutzt Basic eine Tabelle mit Adressen einiger Arithmetik-Routinen; diese finden Sie in Listing 3. Generell ist WRA1 der am meisten benutzte Speicherplatz. Er hält immer die »laufende Variable«, das heißt, wenn Basic irgendein Ergebnis erwartet, dann muß es da stehen. Um welchen Typ es sich handelt, ist in (60BE) notiert.

Mit PEEK sollte man da ruhig einmal nachschauen; es ist nicht alles »Double-Precision, wo man es vermutet. Die trigonometrischen Funktionen werden immer in einfacher Genauigkeit ausgeführt, und FIX ist noch lange nicht Integer, SGL bleibt erhalten, es wird nur abgeschnitten.

Sollten Sie ein Sortier-Programm schreiben wollen, so vergessen Sie die Subtraktions-Routinen, die »Compares« (siehe Listing 1) sind deutlich schneller.

Der Schlüssel zu einem Interpreter ist natürlich die Adreßtabelle der Basic-Befehle. Sie wurde (Listing 4) mit einem Assembler in lesbare Form gebracht, ein Disassembler zeigt da nur sinnlosen Code. Das Prinzip ist einfach: Trifft Basic auf ein Token, subtrahiert es 81H, multipliziert das mit 2, addiert das Ergebnis auf die Anfangsadresse der Tabelle und schon ist die Adresse der Routine gefunden.

Die Tabelle hat zwei Basis-Adressen, nämlich (5757) und (57F1). Letztere gilt für Doppel-Tokens, deren erstes Byte immer FF ist. Ab 692B können Sie das (mit einem Disassembler) nachlesen, bei 68EB startet der »Execution Driver«.

Nicht alle Tokens sind in dieser Tabelle aufgeführt. Solche, die logisch zu einem Befehl gehören, wie TO zu FOR werden dort, solche die erst später in einem Ausdruck auftreten, werden in EVALEX aufgeprüft. Die Routine zur Entwicklung von Ausdrücken (EVALEX) startet bei 7000, einige weitere Interpreter-Routinen in Listing 5.

Etwas anders als im Level-2-Basic werden Variable abgelegt. Da ein Name anstatt zwei jetzt bis zu 40 signifikante Zeichen haben kann, wird eine Variable in VARTAB so eingetragen:

Byte 1: Typ (2, 3, 4 oder 8)
Byte 2: Erstes Zeichen des Namens (Großbuchstabe)
Byte 3: Zweites Zeichen des Namens (Großbuchstabe oder Ziffer)
Byte 4: Anzahl weitere Zeichen (Länge des Namens 2)
Byte 5 und folgende: die weiteren Zeichen des Namens, ASCCI-Code, aber Bit 7 gesetzt.

Hiernach ist also die Namenslänge nicht auf 40 begrenzt, wohl aber die Größe des Puffers, den Basic für die Behandlung des Namens zur Verfügung stellt (ab 5E4B).

Mit Byte 4 wird erreicht, daß Basic die Variablentabelle schnell scannen kann, der Test eines langen Namens kostet natürlich dennoch mehr Zeit als der eines kurzen.

Alle Ein-/Ausgabe-Operationen laufen ähnlich wie in CP/M über eine Funktionsnummer mit anschließendem »CALL BDOS«, allerdings mit folgenden Unterschieden:

Die Funktionsnummer wird im Register A, eine eventuelle Unterfunktion in B übergeben. Ein auszugebendes Zeichen muß immer in C stehen, die Rückgabe erfolgt ins Register B. Gegebenenfalls sind vor dem Aufruf weitere Register zu laden, auch stehen in weiteren Registern Rückgabeparameter bereit.

Der Aufruf erfolgt immer über RST 28. Das Z-Flag bedeutet, die Routine wurde (ohne Fehler) ausgeführt, NZ heißt Fehler oder »kein Erfolg«. Wird zum Beispiel die Ta-


```
Arithmetik-Routinen
===================
Bedeutung der Kürzel:
WRA1: Work Area1, WRA2: Work Area 2
INT: Integer, SP: Single Precision, DP: Double Precision
(HL)...: Bytes ab Adresse in HL, dto. (DE)...
TF : Typ-Flag, Inhalt von (60BE)
BCDE: hält einen SP-Operanden als Exp, MSB, LSB, LSB
---------------------------------------------------------------
2607: DP-Add.  : WRA1=WRA1+0.5
2617: SP-ADD.  : WRA1=WRA1+(HL)....
261C: SP-Sub.  : WRA1=(HL)...-WRA1
261F: SP-SUB.  : WRA1=BCDE-WRA1
2622: SP-ADD.  : WRA1=BCDE+WRA1
2746: LOG      : WRA1=LOG(WRA1)
278B: SP-Mult. : WRA1=BCDE*WRA1
27ED: SP-Div.  : WRA1=BCDE/WRA1
28C4: SGN-Test: Nur SP: Sgn(WRA1)->A, -1=neg, +1=pos., 0=0
28E6: ABS      : WRA1=ABS(WRA1)
28F3: NEG      : WRA1=-WRA1
28FB: SGN      : WRA1=SGN(WRA1)
2905: SGN-Test: wie 28C4, aber alle Typen lt. TF
2917: Push WRA: PUSH WRA1 -> Stack, POP mit POP BC, POP DE
2924: Get SP   : WRA1=(HL)...
2927: Get SP   : WRA1=BCDE
2932: Get SP   : BCDE=WRA1
2935: Get SP   : BCDE=(HL)...
293E: Transfer: WRA1->(HL)         4 Bytes
2941: Transfer: (DE)...->(HL)...   4 Bytes
2945: Transfer: (HL)...->(DE)...   TF Bytes
2946: Transfer: (DE)...->(HL)...   TF Bytes
2949: Transfer: (DE)...->(HL)...   A Bytes
294A: Transfer: (DE)...->(HL)...   B Bytes
2970: Transfer: WRA1 -> WRA2       TF Bytes
Die folgenden Compare-Routinen sind schneller als Sub.
2982: Compare  : SP : WRA1-BCDE    Flags valid
299C:            Wie vor, wenn Vorzeichen gleich
29AF: Compare  : INT: HL-DE        Flags valid
29C4: Compare  : DP : WRA1-WRA2    Flags valid
29ED: Compare  : DP : WRA2-WRA1    Flags valid
29F4: CINT     : WRA1=HL=CINT(WRA1), alle Typen lt. TF
2A53: MAKINT   : HL->WRA1, setzt TF=2, USR-RET
2A6A: CSNG     : WRA1(DP) -> WRA1(SP)
2A87: Convert  : WRA1(INT) -> WRA1(SP)
2A8A: Convert  : (HL)...-> WRA1(SP)
2A96: CDBL     : WRA1(alle Typen lt TF) -> WRA1(DP)
2AA9: Set TF8  : TF=8 (DP)
2AAC: Set TF4  : TF=4 (SP)
2AB0: Test TF  : IF TF=3 THEN exit to "Type mismatch error"
2AB4: TM-Error: Print "type mismatch error"
2AE2: FIX      : WRA1=FIX(WRA1)
2AF5: FRCINT   : WRA1=HL=INT(WRA1) (SP & DP)
2B1A:            wie vor, aber nur DP
2B93: Sub. INT : WRA1=DE-HL
2B9E: Add. INT : WRA1=DE+HL
2BBE: Mult.INT : WRA1=DE*HL
2CEB: Abs(INT) : WRA1=ABS(WRA1)
2C81: Sub.  DP : WRA1=WRA1-WRA2
2C88: Add.  DP : WRA1=WRA1+WRA2
2DB5: Mult. DP : WRA1=WRA1*WRA2
2DF0: REF      : Konstante 10 in DP
2E8C: Div.  DP : WRA1=WRA1/WRA2 (Siehe auch Entry 2E88)
2F16: ASCII    : Convert ASCII-String (TF) -> WRA1
                 (HL)=Begin of String, Delimeter 00 or ","
3331:          : Print "in", weiter bei 313D
313D: Print #  : Print (WRA1)=(HL), i.d.R. Zeilennummer
3147: Convert  : WRA1 -> ASCII, formatiert lt. Bits set in A
                 Bit muß auf 1, sonst unfomatiert
                 Bit 6  5  4  3  2
                     ,  *  $  +  trailing blanks


36D4: SQR      : WRA1=SQR(WRA1)
36ED: Exp.     : Basis in BCDE, Exp. in WRA1, Result->WRA1
378F: EXP      : WRA1=EXP(WRA1)
3876: RND(N)   : WRA1=RND(1..N)
389C: RND(0)   : WRA1=RND(0..1)
die folgenden trigonmetr. Funktionen ergeben immer SP
38EA: COS      : WRA1=COS(WRA1)
38F0: SIN      : WRA1=SIN(WRA1)
388D: TAN      : WRA1=TAN(WRA1)
39A2: ATN      : WRA1=ATN(WRA1)
```

Listing 1. Die Arithmetik-Routinen des Model 4, das alte Level 2-Basic kaum verändert

```
Systemvariable & Buffer      Übersicht
=========================================

5753-5856: Adressen der Basic-Routinen
5787-588A: Adressen für Anfangsbuchstaben A-Z in
           folgender Liste.
588B-5AC0: Basic-Worte alpabetisch sortiert.
5AC1-5AE1: Liste Operator Precedence
5AE2-5B08: Adressen Arith-Routinen
5B09     : NOP
5B0A-5DF3: Texte der Error Meldungen
5DF4-5E07: Adressen USR0-USR9 oder 6A45 (FC-Error)
5E08-5E74: Variable & Buffer
5E75-5FB4: Buffer für Tokenizing
5FB5-60B4: Input Buffer
60B5-62A4: Variable und Buffer
62A5-62A9: Message "in"
62AA-62B0: Message "ready"
62B1-62B6: Message "Break"
------------------------------------------------------------
Die wichtigsten Variablen
============================
5E0C: Code last error
5E0E: Druckerposition für LPOS
5E0F: Ausgabe-Flag. 00=CRT
5E11: Zeilenlänge Printer
5E12: Zeilenlänge CRT
5E17: String Data Pointer
5E19: Aktuelle Zeilennummer
5E1B: Head Pointer. Beginn Basic-Pgm
5E1D: Adresse aktueller Error Text
5E2A: Pointer durch Tokenized Buffer
5E4B-5E71: Variable in VARPTR
60B8: Pointer durch Video-Zeile (auch TAB)
60B9-60BB: RND Use
60BD: Flag locate(00)/create(FF) für Find Var Routine
60BE: Typ-Flag (2=INT. 3=STRING. 4=SGL. 8=DBL)
60C1: letzte Zeilennummer vor aktueller
60CF: HIMEM-Pointer
60D1: Adr. next free Entry to LSTPA
60D3-60F0: Literal String Pool Area. 10*3 Bytes
60F1-60F3: Dope Vektor aktueller String
60F6: Adresse aktuelles Token
60FC: Aktuelle DATA-Zeile
60FE: CALL-Flag
60FF: Read/Input Flag. 00=Input, else Read
6103: Auto Flag
6104: Zeilennummer in Eingabe
6106: Auto-Inkrement
6108: Encoded Statement Pointer
610A: Stack Pointer
610C: ON ERROR-Zeile
610E: EDIT-Zeile. lfd. Zeile in LIST
6110: Text-Pointer in Error-Zeile
6112: ON ERROR Adresse
6114: RESUME-Flag (00) else FF
6115: Adresse next Token (wenn Backspacing)
6117: CONT-Zeile
6119: CONT-Adresse
611B: Tail Pointer (Ende Basic-Pgm)
611D: Pointer einfache Variable
611F: Pointer Array-Variable
6121: DATA-Pointer
6123-613C Typ-Tabelle A-Z für DEFxxx
621F: Option Base (0 oder 1)
6220: Option Base Flag (Option Base+1)
6247: Prot Flag (>0: no LIST. no Direkt Mode)
6259: Trace Flag

Work Area 1        WRA1
----------------------------------------
Adr.  if INT     if SGL       if DBL
----------------------------------------
6264                          LSB
6265                          LSB
6266                          LSB
6267                          LSB
6268     LSB        LSB       LSB
6269     MSB        LSB       LSB
626A                MSB       MSB
626B                EXP       EXP

Work Area 2       WRA2
----------------------------------------
6272     LSB        LSB       LSB
6273     MSB        LSB       LSB
6274                MSB       LSB
6275                EXP       LSB
6276                          LSB
6277                          LSB
6278                          MSB
6279                          EXP

627A-6294: PRINT USING Buffer
```

Listing 2. Systemvariable, Texte und Puffer; Adressen in der Übersicht und im einzelnen

```
                  00100 ;ADRESS-TABELLE FUER ARIHT-ROUTINEN
                  00110 ;==================================
                  00120
5AE2              00130          ORG    5AE2H
5AE2              00140 BASEDP   EQU    $        ;BASIS-ADR. WENN DBL PRECIS.
5AE2 962A         00150          DEFW   2A96H    ;CDBL
5AE4 0000         00160          DEFW   0000H    ;NOT USED
5AE6 F429         00170          DEFW   29F4H    ;CINT
5AE8 B02A         00180          DEFW   2AB0H    ;TESTE TYP
5AEA 6A2A         00190          DEFW   2A6AH    ;CSNG
5AEC 882C         00200          DEFW   2C88H    ;ADD  (DBL)
5AEE 812C         00210          DEFW   2C81H    ;SUB  (DBL)
5AF0 B52D         00220          DEFW   2DB5H    ;MULT (DBL)
5AF2 8C2E         00230          DEFW   2E8CH    ;DIV. (DBL)
5AF4 ED29         00240          DEFW   29EDH    ;COMPARE (DBL)
5AF6              00250 BASSGL   EQU    $        ;BASIS ADR. SGL PRECIS.
5AF6 2226         00260          DEFW   2622H    ;ADD  (SGL)
5AF8 1F26         00270          DEFW   261FH    ;SUB  (SGL)
5AFA 8B27         00280          DEFW   278BH    ;MULT (SGL)
5AFC ED27         00290          DEFW   27EDH    ;DIV. (SGL)
5AFE 8229         00300          DEFW   2982H    ;COMPARE (SGL)
5B00              00310 BASINT   EQU    $        ;BASIS-ADRESS INTEGERS
5B00 9E2B         00320          DEFW   2B9EH    ;ADD
5B02 932B         00330          DEFW   2B93H    ;SUB
5B04 BE2B         00340          DEFW   2BBEH    ;MULT
5B06 6C71         00350          DEFW   716CH    ;DIV.
5B08 AF29         00360          DEFW   29AFH    ;COMPARE
5B0A              00380 ERROR    EQU    $        ;BEGINN TEXTE ERROR MELDUNGEN
0000              00390          END
00000 TOTAL ERRORS
34294  TEXT AREA BYTES LEFT


BASEDP 5AE2 00140
BASINT 5B00 00310
BASSGL 5AF6 00250
ERROR  5B0A 00380
```

Listing 3. Nach dieser Adreßtabelle werden Arithmetik-Routinen aufgerufen

statur abgefragt, heißt NZ »keine Taste gedrückt«, Z dann »Taste betätigt«, der Code steht in A. In Listing 6 sind einige Routinen praktisch demonstriert. Sie sind ein Teil des DOS, das hier aus Platzgründen nicht behandelt werden kann. Nur soviel: Auch jede Disk-Operation läuft über RST 28. Sie sollten nie eine Routine direkt aufrufen. Der Sinn dieser Organisation ist es, das System beliebig erweitern (und ändern) zu können, womit sich absolute Adressen verschieben, die Funktionsnummern aber bleiben. Den Drucker hätte ich fast vergessen, Funktion 6, Zeichen in C, oder direkt über Port F8. Die RST-Calls des Level2-Basics sind hier entfallen. Das kostet zwar Speicher, macht das System aber schneller. Aus RST 8 (Syntax-Check) wurde CALL 430E, RST 10 (Scanner) ist jetzt CALL 6943, RST 18 (HL-DE) wurde zu CALL 4308 und RST 20 (get vartyp) zu CALL 735E.

Interpreter mit Error-Meldungen knacken

Ein weiterer Weg, so einen Interpreter zu knacken, sind die Error-Meldungen, die hier auf vier Arten erzeugt werden. Meistens geschieht dies über JP in die Sprungliste (Listing 7). Mit einem Disassembler sieht man da nur Unmengen von »LD BC,nn«, ein beliebter Trick, den Microsoft häufig anwendet. Wenn Sie in der Adreßtabelle nur 3-Byte-Abstände vorfinden, geht das auf diese Art. Bei 2-Byte-Abständen steht da »OR AF«, womit A > 0 wird. Steigt man beim zweiten Byte ein (AF), heißt das »XOR A«, also A = 0. Zu den Error-Meldungen, Weg 2: "LD E,Code/JP 6343". 6343 ist die Adresse, auf die die vorher genannte Liste durchfällt. Weg 3: ein JP zur Adresse, wo ein »Weg 2« beginnt, und Weg 4: HL wird mit der Adresse des Fehlertextes geladen und dann wird 4660 (Print Text ab (HL)) gerufen. Als Hilfe dazu Listing 8.

Störend ist oft, daß bei Syntax-Fehlern sofort EDIT aufgerufen wird, womit leider alle Variablen gelöscht werden. Wollen Sie das vermeiden, poken Sie NOPs (00) auf 63F5-63F7. Da wir gerade beim Poken sind. Ist Ihr Drucker grafikfähig, Sie wollen also die Punkte nicht im Screenprint, dann reicht

"POKE &HOACE,&H18". Unter Programm-Kontrolle können Sie den Screendump auch aufrufen und zwar mit :"X=&HOABA:CALL X". Das sind zwar leider absolute Calls ins DOS, aber dafür habe ich keine Funktionsnummer gefunden. Für folgendes wird es wohl nie eine geben, die Sache mit dem Listschutz: Ein mit P-Option gesichertes Programm setzt das Flag in 6247 <>0. So ziemlich am Anfang der LIST-Routine wird das Flag geprüft. Jetzt poken geht nicht mehr, da die Routine, die Direkt-Mode testet, auch das Flag prüft. Die wird aber nicht immer aufgerufen, zum Beispiel nicht bei SYSTEM. Sie können also in aller Ruhe ein Programm nachladen (und dabei starten) in dem steht »COR A:LD« (6247),A:RET". Haben Sie das nicht, geht folgendes: Vor dem Laden des P-Programms poken Sie auf 5740 ein C9 (RET), und nach dem Laden auf 6247 eine Null, und Ihr versehentlich mit P gespeichertes Programm ist wieder sichtbar.

Konstanten schneller als Variablen

Bliebe noch zu erwähnen, warum hier Konstanten schneller sind als Variable. Sie werden direkt nach der Eingabe umgewandelt und stehen als Hex-Zahlen im Programmtext.

Der Encoder startet bei 6583, die Konstanten werden ab 6731 konvertiert. Aus "&Hxx" wird "OC nn", wobei nn für die beiden Bytes in LSB/MSB-Order steht. Auch eine Oktal-Konstante wird so abgelegt, nur steht anstatt 0C das Typ-Byte 0B. Eine Dezimal-Konstante wird ähnlich behandelt, wenn sie größer als 255 ist. Das Typ-Byte ist dann 1C. Bei Zahlen von 0 bis 9 wird 1 addiert (01H-0AH), das Typ-Byte entfällt dann, das higher Nibble wird aber 1, aus 0 - 9 wird 11H - 1AH. Aus Zahlen von 10 bis 255 wird OAH - FFH, wobei das Typ-Byte jetzt 0FH ist. Wenn Sie also beim Disassemblieren im Scanner (so ab 6966) oder im LIST auf diese Bytes stoßen, that's it. Da direkt nach der Eingabe konvertiert wird, muß da auch geprüft werden. Das ist der Grund, warum die Eingabe einer Zeile mit FC-Error quittiert werden kann.

Bliebe noch die obligatorische Frage zu klären, wo man in den Interpreter mit eigenen Routinen einsteigt. Hier ist die Sache ganz einfach, da man immer im RAM ist. Man »patcht« da, wo es für den jeweiligen Fall am günstigsten ist. Die Routine, die die Tastatur abfragt, startet bei 7BC8. Hauptschleife und Scanner hatten wir schon erwähnt. EVALEX auch, also: Wo bleiben die interessanten Listings für das Modell 4?

(P. Wollschläger/ml)

```
Adressen einiger Interpreter-Routinen
=====================================
39FB: FNDVAR    : Find Var.Adr. Var-Name pointed by HL.
                  Deli. 00 or ",". Adr. -> DE
410E: PRTCRT    : Output Device is CRT
4308: Compare   : HL-DE, Flags valid
430E: SNTEST    : Teste auf Zeichen, das dem Call folgt.
                  Syntax-Error wenn <>, Incr. HL
442E: TESTLET.  : Teste Zeichen (HL) auf Buchszaben, NC if
442F:           : wie vor, Zeichen in A
4660: PRTLIN    : Print Text ab (HL) bis 00
5740: NODIRT    : Verhindert Direkt-Mode wenn Prot-Flag (6247)
                  gesetzt (<>00)
5749: TESTPT    : Teste P-Mode (Listschutz)
6343: PRTERR    : Print Error-Text, Code in E
6521: NEWLIN    : Update Line Pointers
6564: FNDLIN    : Find Line Number
6943: NXTBYT    : Hole nächstes Zeichen aus dem Pgm-Text.
                  Skip Blanks u. einige Controls, behandle
                  einige Codes direkt.
```

Listing 5. Einige Interpreter-Routinen, weitere im Text

Listing 4. Der Schlüssel zum Interpreter: Adressen der Routinen je Befehl

```
                ;TABELLE MIT ADRESSEN DER BASIC-ROUTINEN LT. TOKEN
                ;===============================================

                ;FUER NICHT BELEGTE TOKENS STEHT '****'.
                ;ADR. 0000: RET
                ;ADR. 5329: PRINT "SYNTAX-ERROR"

5757                    ORG     5757H
5757 DE43               DEFW    43DEH   ;81: END
5759 0A68               DEFW    680AH   ;82: FOR
575B E244               DEFW    44E2H   ;83: NEXT
575D 156B               DEFW    6B15H   ;84: DATA
575F 6C6E               DEFW    6E6CH   ;85: INPUT
5761 F639               DEFW    39F6H   ;86: DIM
5763 476F               DEFW    6F47H   ;87: READ
5765 3C6B               DEFW    6B3CH   ;88: LET
5767 BA6A               DEFW    6ABAH   ;89: GOTO
5769 8F6A               DEFW    6A8FH   ;8A: RUN
576B 776C               DEFW    6C77H   ;8B: IF
576D 2043               DEFW    4320H   ;8C: RESTORE
576F A36A               DEFW    6AA3H   ;8D: GOSUB
5771 F56A               DEFW    6AF5H   ;8E: RETURN
5773 176B               DEFW    6B17H   ;8F: REM
5775 3A43               DEFW    433AH   ;90: STOP
5777 DB6C               DEFW    6CDBH   ;91: PRINT
5779 3944               DEFW    4439H   ;92: CLEAR
577B 4076               DEFW    7640H   ;93: LIST
577D 6862               DEFW    6268H   ;94: NEW
577F A26B               DEFW    6BA2H   ;95: ON
5781 F975               DEFW    75F9H   ;96: WAIT
5783 1D74               DEFW    741DH   ;97: DEF
5785 C578               DEFW    78C5H   ;98: POKE
5787 9743               DEFW    4397H   ;99: CONT
5789 2963               DEFW    6329H   ;9A: ****
578B 2963               DEFW    6329H   ;9B: ****
578D F375               DEFW    75F3H   ;9C: OUT
578F B56C               DEFW    6CB5H   ;9D: LPRINT
5791 3B76               DEFW    763BH   ;9E: LLIST
5793 D47C               DEFW    7CD4H   ;9F: CLS
5795 0000               DEFW    0000H   ;A0: ****
5797 2963               DEFW    6329H   ;A1: ****
5799 176B               DEFW    6B17H   ;A2: ELSE
579B AB43               DEFW    43ABH   ;A3: TRON
579D AC43               DEFW    43ACH   ;A4: TROFF
579F B143               DEFW    43B1H   ;A5: SWAP
57A1 F243               DEFW    43F2H   ;A6: ERASE
57A3 223D               DEFW    3D22H   ;A7: EDIT
57A5 3A6C               DEFW    6C3AH   ;A8: ERROR
57A7 EB6B               DEFW    6BEBH   ;A9: RESUME
57A9 7D78               DEFW    787DH   ;AA: DELETE
57AB 456C               DEFW    6C45H   ;AB: AUTO
57AD FE78               DEFW    78FEH   ;AC: RENUM
57AF F969               DEFW    69F9H   ;AD: DEFSTR
57B1 FC69               DEFW    69FCH   ;AE: DEFINT
57B3 FF69               DEFW    69FFH   ;AF: DEFSNG
57B5 026A               DEFW    6A02H   ;B0: DEFDBL
57B7 016E               DEFW    6E01H   ;B1: LINE
57B9 0000               DEFW    0000H   ;B2: ****
57BB 0000               DEFW    0000H   ;B3: ****
57BD 2B52               DEFW    522BH   ;B4: WHILE
57BF 4C52               DEFW    524CH   ;B5: WEND
57C1 BB52               DEFW    52BBH   ;B6: CALL
57C3 5C56               DEFW    565CH   ;B7: WRITE
57C5 156B               DEFW    6B15H   ;B8: COMMON
57C7 4E53               DEFW    534EH   ;B9: CHAIN
57C9 2E7A               DEFW    7A2EH   ;BA: OPTION
57CB 527D               DEFW    7D52H   ;BB: RANDOM
57CD 0000               DEFW    0000H   ;BC: ****
57CF A17D               DEFW    7DA1H   ;BD: SYSTEM
57D1 0000               DEFW    0000H   ;BE: ****
57D3 7250               DEFW    5072H   ;BF: OPEN
57D5 784D               DEFW    4D78H   ;C0: FIELD
57D7 7D4F               DEFW    4F7DH   ;C1: GET
57D9 7C4F               DEFW    4F7CH   ;C2. PUT
57DB 3B4D               DEFW    4D3BH   ;C3: CLOSE
57DD DC4B               DEFW    4BDCH   ;C4: LOAD
57DF 9A4C               DEFW    4C9AH   ;C5: MERGE
57E1 0000               DEFW    0000H   ;C6: ****
57E3 827E               DEFW    7E82H   ;C7: NAME
57E5 8F51               DEFW    518FH   ;C8: KILL
57E7 D54D               DEFW    4DD5H   ;C9: LSET
57E9 D44D               DEFW    4DD4H   ;CA: RSET
57EB 0000               DEFW    0000H   ;CB: ****
57ED 0000               DEFW    0000H   ;CD: SOUND

                ;JETZT FOLGEN 2-BYTES TOKENS.
                ;DAS ERSTE BYTE IM BASIC-TEXT IST IMMER FF.
                ;IN DER LISTE STEHT DAS ZWEITE.

57EF 7348               DEFW    4873H   ;81: LEFT$
57F1 A348               DEFW    48A3H   ;82: RIGHT$
57F3 AC48               DEFW    48ACH   ;83: MID$
57F5 FB23               DEFW    23FBH   ;84: SGN
57F7 F52A               DEFW    2AF5H   ;85: INT
57F9 E628               DEFW    28E6H   ;86: ABS
57FB D436               DEFW    36D4H   ;87: SQR
57FD 7633               DEFW    3376H   ;88: RND
57FF F038               DEFW    38F0H   ;89: SIN
5801 4627               DEFW    2746H   ;8A: LOG
5803 8F37               DEFW    378FH   ;8B: EXP
5805 EA38               DEFW    38EAH   ;8C: COS
5807 8D39               DEFW    398DH   ;8D: TAN
5809 A239               DEFW    39A2H   ;8E: ATN
580B 244A               DEFW    4A24H   ;8F: FRE
580D DC75               DEFW    75DCH   ;90: INP
580F C273               DEFW    73C2H   ;91: POS
5811 0548               DEFW    4805H   ;92: LEN
5813 D945               DEFW    45D9H   ;93: STR$
5815 CD48               DEFW    48CDH   ;94: VAL
5817 1148               DEFW    4811H   ;95: ASC
5819 2148               DEFW    4821H   ;96: CHR$
581B BB78               DEFW    78BBH   ;97: PEEK
581D 5848               DEFW    4858H   ;98: SPACE$
581F CF45               DEFW    45CFH   ;99: OCT$
5821 D445               DEFW    45D4H   ;9A: HEX$
5823 BD73               DEFW    73BDH   ;9B: LPOS
5825 F429               DEFW    29F4H   ;9C: CINT
5827 6A2A               DEFW    2A6AH   ;9D: CSNG
5829 962A               DEFW    2A96H   ;9E: CDBL
582B E22A               DEFW    2AE2H   ;9F: FIX
                ;TOKENS A0-A9 ADR. 0000H ****
5843                    ORG     5843H
5843 9B4A               DEFW    4A9BH   ;AA: CVI
5845 9E4A               DEFW    4A9EH   ;AB: CVS
5847 A14A               DEFW    4AA1H   ;AC: CVD
5849 E34E               DEFW    4EE3H   ;AD: EOF
584B 274F               DEFW    4F27H   ;AE: LOC
584D 2A4F               DEFW    4F2AH   ;AF: LOF
584F 824A               DEFW    4A82H   ;B0: MKI$
5851 854A               DEFW    4A85H   ;B1: MKS$
5853 884A               DEFW    4A88H   ;B2: MKD$
5855 987D               DEFW    7D98H   ;B3: ROW
0000                    END
00000 TOTAL ERRORS
31865 TEXT AREA BYTES LEFT
```

```
                 :DEMONSTRATION DOS-CALLS (RST 28H)
                 :==================================

                 :TASTAUR-ABFRAGE
                 :---------------
0000 3E08        SCANKI  LD      A.8             :TESTE KEYBOARD
0002 EF                  RST     28H             :CALL FUNCTION
0003 20FB                JR      NZ.SCANKI
                                                 :CHAR NOW IN A

0005 3E01        WAITKI  LD      A.1             :SCAN KB & WAIT
0007 EF                  RST     28H
                                                 :CHAR NOW IN A

0008 3E09        GETLIN  LD      A.9             :GET LINE
000A 0607                LD      B.7             :ALLOW MAX 7 CHAR
000C 0E00                LD      C.0             ;MAX NOW IN BC
000E 21B55F              LD      HL.5FB5H        ;USE BASICS BUFFER
0011 EF                  RST     28H
0012 3804                JR      C.BREAK         :IF BREAK TERMINATOR
0014 AF                  XOR     A               :A=0
0015 B8                  CP      B               :CHARS INPUT
0016 28F0                JR      Z.GETLIN        :IF NULLSTRING
0018             BREAK   EQU     $               :HANDLE BREAK HERE

                 :AUSGABE AUF VIDEO
                 :-----------------
0018 3E02        DSPCHR  LD      A.2             :DISP CHAR §·CURSOR
001A 0E58                LD      C.'X'           :CHAR TO DISP
001C EF                  RST     28H

7CD4                     ORG     7CD4H           :BASICS CLS-ROUTINE
7CD4 E5                  PUSH    HL
7CD5 0E1C                LD      C.1CH           :HOME CURSOR
7CD7 3E02                LD      A.2
7CD9 EF                  RST     28H
7CDA 0E1F                LD      C.1FH           :ERASE TO END OF FRAME
7CDC 3E02                LD      A.2
7CDE EF                  RST     28H
7CDF E1                  POP     HL
7CE0 AF                  XOR     A               :A=0
7CE1 32B860              LD      (60B8H).A       :TAB(0)
7CE4 C9                  RET

7CE5 3E0A        DSPMSG  LD      A.0AH           :DISP MESSAGE
7CE7 21ED7C              LD      HL.MESSG        :POINT TO TEXT
7CEA EF                  RST     28H
7CEB 1809                JR      POINT
7CED 54          MESSG   DEFM    'TEXT HERE'
                         DEFB    03H
                 :VIDEO CONTROL
                 :-------------
7CF6.3E0F        POINT   LD      A.0FH           :VIDEO CONTROL
7CF8 0601                LD      B,1             :FUNCTION POINT
7CFA 2603                LD      H,3             :ROW 3
7CFC 2E07                LD      L,7             :COL 7
7CFE EF                  RST     28H
                                                 ;CHAR (H),(L) NOW

7CFF 3E0F        PRTAT   LD      A.0FH
7D01 0602                LD      B.2             :FUNCTION PRINT A1
7D03 210703              LD      HL.0307H        :ROW 3, COL 7
7D06 0E58                LD      C,'X'           :CHAR TO PRINT
7D08 EF                  RST     28H

7D09 3E0F        MOVCUR  LD      A.0FH
7D0B 0603                LD      B,3             :FUNCTION MOVE CURSOR
7D0D 210703              LD      HL,0307H        ;ROW 3, COL 7
7D10 EF                  RST     28H

7D11 3E0F        FNDCUR  LD      A,0FH
7D13 0604                LD      B,4             :FIND CURSOR POSITION
7D15 EF                  RST     28H
                                                 :CURSOR POS. NOW IN H,L
0000                     END
00000 TOTAL ERRORS
33502  TEXT AREA BYTES LEFT
```

Listing 6. Jegliches I/O läuft über DOS-Calls. Hier einige praktische Beispiele

```
                 :                    ADRESSEN FEHLERMELDUNGEN
                 :                    ========================
                 :EIN JP ODER CALL ADR. FUEHRT ZUR AUSGABE DES
                 :ENTPRECHENDEN TEXTES. "LD BC.NN" SCHUETZT DIE
                 :"LD E.N"-BEFEHLE.

62FB                     ORG     62FBH
62FB 1E3D                LD      E.3DH     :DISK FULL
62FD 01                  DEFB    01
62FE 1E39                LD      E.39H     :DEVICE I/O-ERROR
6300 01                  DEFB    01
6301 1E36                LD      E.36H     :BAD FILE MODE
6303 01                  DEFB    01
6304 1E35                LD      E.35H     :FILE NOT FOUND
6306 01                  DEFB    01
6307 1E34                LD      E.34H     :BAD FILE NUMBER
6309 01                  DEFB    01
630A 1E33                LD      E.33H     :INTRENAL ERROR
630C 01                  DEFB    01
630D 1E3E                LD      E.3EH     :INPUT PAST END
630F 01                  DEFB    01
6310 1E37                LD      E.37H     :FILE ALREADY OPEN
6312 01                  DEFB    01
6313 1E40                LD      E.40H     :BAD FILE NAME
6315 01                  DEFB    01
6316 1E3F                LD      E.3FH     :BAD RECORD NUMBER
6318 01                  DEFB    01
6319 1E32                LD      E.32H     :FIELD OVERFLOW
631B 01                  DEFB    01H
631C 1E43                LD      E.43H     :TO MANY FILES
631E 01                  DEFB    01
631F 1E3A                LD      E.3AH     :FILE ALREADY EXISTS
6321 1820                JR      PRINT

6323 2AFC60              LD      HL.(60FCH)        :LAST DATA LINE
6326 22195E              LD      (5E19H).HL        :CURRENT LINE

6329 1E02                LD      E.02H     :SYNTAX ERROR
632B 01                  DEFB    01
632C 1E0B                LD      E.0BH     :DIVISION BY ZERO
632E 01                  DEFB    01
632F 1E01                LD      E.01H     :NEXT WITHOUT FOR
6331 01                  DEFB    01
6332 1E0A                LD      E.0AH     :REDIMED ARRAY
6334 01                  DEFB    01
6335 1E12                LD      E.12H     :UNDEFINED USER FUNCTION
6337 01                  DEFB    01
6338 1E14                LD      E.14H     :RESUME WITHOUT ERROR
633A 01                  DEFB    01
633B 1E06                LD      E.06      :OVERLOW
633D 01                  DEFB    01
633E 1E16                LD      E.16H     :MISSING OPERAND
6340 01                  DEFB    01
6341 1E0D                LD      E.0DH     :TYPE MISMATCH
6343             PRINT   EQU     $
0000                     END
00000 TOTAL ERRORS
33897  TEXT AREA BYTES LEFT

PRINT  6343 00055   00033
```

Listing 7. Entry-Points zu Fehlermeldungen: Demonstration eines häufig von Microsoft gebrauchten Tricks

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
            -----------------------------------------------
5B00 =>  9E 2B 93 2B BE 2B 6C 71 AF 29 00 4E 45 58 54 20    .+.+.+1q.).NEXT
5B10 =>  77 69 74 68 6F 75 74 20 46 4F 52 00 53 79 6E 74    without FOR.Synt
5B20 =>  61 78 20 65 72 72 6F 72 00 52 45 54 55 52 4E 20    ax error.RETURN
5B30 =>  77 69 74 68 6F 75 74 20 47 4F 53 55 42 00 4F 75    without GOSUB.Ou
5B40 =>  74 20 6F 66 20 44 41 54 41 00 49 6C 6C 65 67 61    t of DATA.Illega
5B50 =>  6C 20 66 75 6E 63 74 69 6F 6E 20 63 61 6C 6C 00    l function call.
5B60 =>  4F 76 65 72 66 6C 6F 77 00 4F 75 74 20 6F 66 20    Overflow.Out of
5B70 =>  6D 65 6D 6F 72 79 00 55 6E 64 65 66 69 6E 65 64    memory.Undefined
5B80 =>  20 6C 69 6E 65 20 6E 75 6D 62 65 72 00 53 75 62     line number.Sub
5B90 =>  73 63 72 69 70 74 20 6F 75 74 20 6F 66 20 72 61    script out of ra
5BA0 =>  6E 67 65 00 44 75 70 6C 69 63 61 74 65 20 44 65    nge.Duplicate De
5BB0 =>  66 69 6E 69 74 69 6F 6E 00 44 69 76 69 73 69 6F    finition.Divisio
5BC0 =>  6E 20 62 79 20 7A 65 72 6F 00 49 6C 6C 65 67 61    n by zero.Illega
5BD0 =>  6C 20 64 69 72 65 63 74 00 54 79 70 65 20 6D 69    l direct.Type mi
5BE0 =>  73 6D 61 74 63 68 00 4F 75 74 20 6F 66 20 73 74    smatch.Out of st
5BF0 =>  72 69 6E 67 20 73 70 61 63 65 00 53 74 72 69 6E    ring space.Strin
5C00 =>  67 20 74 6F 6F 20 6C 6F 6E 67 00 53 74 72 69 6E    g too long.Strin
5C10 =>  67 20 66 6F 72 6D 75 6C 61 20 74 6F 6F 20 63 6F    g formula too co
5C20 =>  6D 70 6C 65 78 00 43 61 6E 27 74 20 63 6F 6E 74    mplex.Can't cont
5C30 =>  69 6E 75 65 00 55 6E 64 65 66 69 6E 65 64 20 75    inue.Undefined u
5C40 =>  73 65 72 20 66 75 6E 63 74 69 6F 6E 00 4E 6F 20    ser function.No
5C50 =>  52 45 53 55 4D 45 00 52 45 53 55 4D 45 20 77 69    RESUME.RESUME wi
5C60 =>  74 68 6F 75 74 20 65 72 72 6F 72 00 55 6E 70 72    thout error.Unpr
5C70 =>  69 6E 74 61 62 6C 65 20 65 72 72 6F 72 00 4D 69    intable error.Mi
5C80 =>  73 73 69 6E 67 20 6F 70 65 72 61 6E 64 00 4C 69    ssing operand.Li
5C90 =>  6E 65 20 62 75 66 66 65 72 20 6F 76 65 72 66 6C    ne buffer overfl
5CA0 =>  6F 77 00 3F 00 3F 00 46 4F 52 20 57 69 74 68 6F    ow.?.?.FOR Witho
5CB0 =>  75 74 20 4E 45 58 54 00 3F 00 3F 00 57 48 49 4C    ut NEXT.?.?.WHIL
5CC0 =>  45 20 77 69 74 68 6F 75 74 20 57 45 4E 44 00 57    E without WEND.W
5CD0 =>  45 4E 44 20 77 69 74 68 6F 75 74 20 57 48 49 4C    END without WHIL
5CE0 =>  45 00 46 49 45 4C 44 20 6F 76 65 72 66 6C 6F 77    E.FIELD overflow
5CF0 =>  00 49 6E 74 65 72 6E 61 6C 20 65 72 72 6F 72 00    .Internal error.
```

*Listing 8.
Viele Fehlermeldungen werden nach direktem Zugriff auf diese Tabelle ausgegeben*

MSDOS - Trick

COMMAND /F

Der Aufruf des MSDOS Kommandointerpreters COMMAND.COM mit dem Parameter /F bewirkt, daß bei Zugriffsfehlern auf Disketten und Festplatten auf die Frage: (A)bbrechen, (W)iederholen, (U)ebergehen automatisch mit (U)ebergehen geantwortet wird.

Das ist nicht immer ganz ungefährlich, kann aber manchmal ganz nützlich sein.

Um wieder zur normalen Fehlerbehandlung zurückzukehren, kann man den Kommandointerpreter mit "EXIT" verlassen oder "COMMAND /D" aufrufen.

Microsoft Link8Ø und .SYM-Datei

Als ich vor einiger Zeit mehrfach und wiederholt ein längeres Assemblerprogramm übersetzen wollte, lernte ich den theoretischen Nutzen einer sogenannen SYMBOL-Datei schätzen. Hingewiesen wurde ich auf diese Möglichkeit durch die Beschreibung des Debugger-Programms SID. Doch wie sollte ich zu solch einer SYMBOL-Datei meines Programms kommen ?

Ich wußte zwar, daß das Assembler-Listing am Ende eine Auflistung der Symbole aufwies, doch wie sollte ich das in eine SYMBOL-Datei schreiben und hatte das dann auch das erforderliche Format für SID ? Mein Microsoft-Makroassembler M8Ø gab mir in seiner Beschreibung darauf keine Antwort. Erst eine PD-Beschreibung des Microsoft-Linkers L8Ø wies auf einen Schalter hin, der die Bildung einer SYMBOL-Datei veranlassen sollte.

Geschafft, dachte ich. Doch die Praxis zeigte mir anderes. Der /Y-switch des Linkers legte zwar die Datei .SYM an, der Inhalt hatte jedoch nichts mit Symbolen zu tun und war bestenfalls die Speicherkopie eines unbenutzten Programmabschnittes von L8Ø. Was nun ? Die mir von der Fa. Triumph-Adler gelieferten Unterlagen gaben nicht einmal diesen Schalter an.

Die Lösung fand sich auf dem letzten Clubtreffen. Hartmut Obermann bot mir die Originalbeschreibung des Microsoft-Linkers an. Doch da stand erstmal auch nichts anderes als in der PD-Beschreibung. Hmm...

Nach weiteren Dutzenden Versuchen mit variierender Schreibweise fand ich beim nochmaligen Studium der Originalbeschreibung die Lösung: Der Linker erstellt nur für GLOBALS eine .SYM-Datei. Aha ! Mit GLOBAL ist ein 'Public Symbol' gemeint und da diese ausdrücklich gekennzeichnet sein müssen, findet L8Ø z. B. in einem normal disassemblierten Programm kein solches Symbol für den Eintrag in die .SYM-Datei.

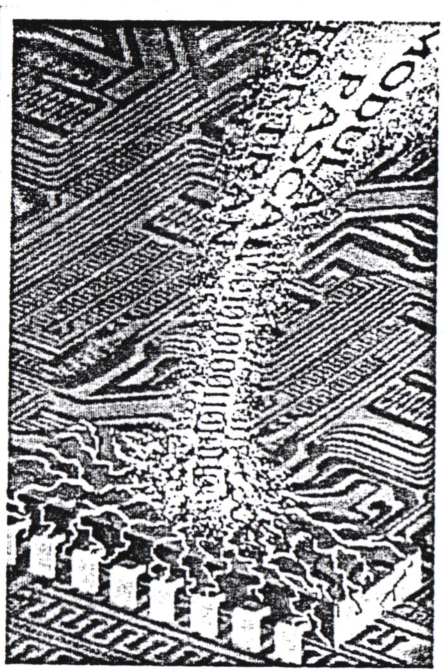
Abhilfe ? Der umständliche Weg wäre, alle LABELs im Programmvorspann als PUBLIC auszuweisen. Der leichtere Weg ist die Kurzkennzeichnung des Labels im Programm durch Verdoppelung des Doppelpunkts (LABEL::). Das läßt sich sogar automatisieren.

Einfacher gehts natürlich mit einem anderen Assemler, z. B. MAC oder RMAC, der die SYMBOL-Datei direkt erzeugt. Doch solch einen hatte ich zu diesem Zeitpunkt noch nicht.

Herzliche Grüße

Uwe Schoberth

(;-) Noch lebt meine CPU 8Ø85 !



Software-Technik und Compilerbau

oder:

Wie man einer CPU Programme mundgerecht zubereitet.

Bestimmt hat schon jeder von Ihnen oft das Wort »Compiler« gehört und verwendet, ehrfurchtsvoll davor den Hut gezogen oder neidisch auf Spezialisten, die mit diesem »Ding« umgehen konnten geschaut. Der Ausdruck stammt aus dem Englischen und wird laut Wörterbuch mit »Sammler, Zusammensteller« übersetzt. Als »Startup« für unsere, in diesem Heft beginnende Serie über Compilerbau (hört, hört!) wird dieser Beitrag grundsätzliche Einblicke in den Compilerbau liefern.

Ein in einer höheren Programmiersprache geschriebenes Programm ist nichts weiter als ein Text, also eine Folge von Zeichen. Betrachten wir dazu ein kleines Pascal Programm:

```
PROGRAM Beispiel;
VAR x: INTEGER;
BEGIN
  x:= 1;
END.
```

Mit diesem Programm werden folgende Operationen beschrieben:

1. Reserviere Speicherplatz für die Variable x.
2. Weise x den Wert 1 zu.
3. Stelle den Speicherplatz von x wieder zur Verfügung (am Programmende).

Der Prozessor eines Computers versteht aber die Anweisungen in der Form eines Pascal-Programmes nicht. Die sogenannte Maschinensprache eines Prozessors ist nichts weiter als eine Folge von Zahlen, die als Operationen (Anweisungen) interpretiert werden. Diese Maschinensprache ist von Prozessor zu Prozessor unterschiedlich. Wenn Sie also ein Programm in Maschinensprache für den Prozessor XYZ geschrieben haben und wollen dieses Programm auf einem anderen Computer mit dem Prozessor ABC laufen lassen, müßen Sie das Programm umschreiben bzw. in dessen Maschinensprache übersetzen.

Unser Beispielprogramm könnte in Maschinensprache so aussehen:

```
 1  10
16  227  2
10  228  1
37  227  228
9
```

Mit jeder Zeile legen wir einen Maschinenbefehl fest, der aus einem Operationscode besteht und evtl. Argumente oder Parameter enthält. Der Operationscode bezeichnet die auszuführende Aktion, während die Argumente weiterführende Angaben enthalten, die die Operation für ihre Ausführung benötigt.

Dieses Maschinenspracheprogramm ist schwierig zu lesen, denn man sieht ihm nicht an, was es eigentlich macht. Man ist deshalb schon sehr früh dazu übergegangen, die Operationscodes in eine symbolische Schreibweise (Mnemonics) zu überführen. Das sähe dann so aus:

```
PROG 10
  RES 227 2
  CONST 228 1
  STORE 227 228
  ENDPROG
```

Dies ist eine Möglichkeit für eine sogenannte Assemblersprache.

Um sich von den komplizierten und maschinenabhängigen Befehlen der Maschinen- oder Assemblersprache zu lösen, entwickelte man die höheren Programmiersprachen wie COBOL, FORTRAN, PASCAL, BASIC und ... und ... und .... Die Befehle und Anweisungen dieser Sprache sind der (englischen) Umgangssprache entlehnt und orientieren sich an den zu lösenden Problemen, nicht an den Gegebenheiten einer bestimmten Maschine oder eines Prozessors. Deshalb werden sie auch problemorientierte Sprache genannt.

Es wird übersetzt ...

Ein Programm in einer höheren Programmiersprache liegt, wie oben gesagt, als reine Zeichenfolge vor. Dieser Text muß nun in ein äquivalentes Maschinenprogramm übersetzt werden. Äquivalent heißt, daß das Maschinenprogramm das ausführen muß, was die Anweisungen des Programms in der höheren Sprache ausführen würden, wenn der Prozessor es direkt ausführen könnte.

Übersetzungen aus einer Hochsprache gestalten sich nicht so einfach wie die Übersetzung eines Assemblerprogramms in Maschinensprache, den dabei handelt es sich ja um eine einfache 1:1 Übersetzung: es brauchen nur die symbolischen Befehle in den entsprechenden Code übersetzt zu werden, wobei jedes Symbol genau einem Operationscode entspricht. Bei der Übersetzung von höheren Programmiersprachen in Maschinensprache handelt es sich um eine 1:n Übersetzung. Jede Anweisung wird in der Regel in mehrere Maschinenanweisungen übersetzt.

Analyse und Synthese

Jede höhere Programmiersprache besteht normalerweise aus Symbolen und reservierten Wörtern. In Pascal gibt es zum Beispiel die reservierten Wörter PROGRAM, VAR, PROCEDURE usw. Die erste Aufgabe des Compilers ist es, den Eingabetext Zeichen für Zeichen zu lesen und die Symbole der Sprache zu erkennen. Liest der Compiler z.B. die Zeichenfolge

V A R

so muß er erkennen, daß es sich dabei um das Schlüsselwort VAR handelt. Diese erste Phase nennt man die lexikalische Analyse, da hier der ursprüngliche Text (der ja alles mögliche sein kann) in die lexikalischen oder Sprachelemente der jeweiligen Sprache zerlegt wird.

Nach dieser Umwandlung in Symbole muß der Compiler feststellen, ob das nun aus Sprachelementen oder lexikalischen Symbolen bestehende Programm den Sprachregeln von PASCAL (bei einem Pascal Compiler) entspricht. Die Aufgabe wird diesmal etwas komplexer, denn der Compiler prüft die vorliegenden »Sätze« (Anweisungen) des Programms auf grammatikalische Fehler. Tatsächlich formuliert man die Sprachregeln einer Programmiersprache durch eine Grammatik, wie dies in einer natürlichen Sprache auch der Fall ist. Ein Compiler muß also etwas von Grammatik verstehen. Liegt z.B. der »Satz«

x:= 1

vor, muß der Compiler diesen als Zuweisung erkennen. Wichtig ist jedoch, daß der Compiler auch Verletzungen der Sprachregeln oder Syntax erkennt. Bei einem Satz wie

x = 1

muß sofort erkannt werden, daß es sich hierbei nicht um einen korrekten Satz handeln kann. Diese Phase der Compilierung nennt man die syntaktische Analyse.

Leider ist damit die Analysearbeit noch nicht erledigt. Es gibt Sätze, die syntaktisch korrekt, aber trotzdem fehlerhaft sind. Eine Zuweisung wie

x:= FALSE

ist gemäß der grammatikalischen Regeln oder der Syntax von PASCAL sicherlich korrekt. In dem oben beschriebenen Programm handelt es sich bei der Variablen x aber um eine ganzzahlige Variable (integer), an die wir hier einen Wahrheitswert zuweisen. Diese Datentypen sind jedoch unverträglich. hier liegt also ein Fehler vor. Auch die folgende Anweisung hätte in unserem Beispielprogramm keine Bedeutung:

IF m = 10 THEN x:= x + 1;

Da die Variable m nicht definiert ist, kann diese Anweisung nicht ausgeführt werden.

Der Compiler muß also prüfen, ob sich die benutzten Namen auf vereinbarte und damit bekannte Objekte beziehen. Dazu kommt bei der Programmiersprache Pasal, daß man vereinbarte Namen nicht unbedingt im ganzen Programm benutzen darf. Wenn Sie zum Beispiel Variablen in einem Unterprogramm vereinbaren, können Sie auf diese nicht außerhalb des Unterprogramms zugreifen. Man nennt dies auch den Gültigkeitsbereich von Variablen oder sonstigen vereinbarten Objekten.

Ob eine benutzte Variable in Ihrem Gültigkeitsbereich benutzt wird, stellt der Compiler in der Scope Analyse oder Gültigkeitsbereichsanalyse fest. Die Verträglichkeit von Datentypen wird während der Typenprüfung vom Compiler erkannt. Die letzten beiden Analysen faßt man zusammen zur semantischen Analyse. Die Semantik ist nämlich die Lehre von der Bedeutung einer Sprache. Hier werden »unsinnige« Konstruktionen erkannt, während bei der Syntaxprüfung nur die formalen Sprachregeln überprüft werden.

Bisher wurde noch keine einzige Zeile des Originalprogramms übersetzt! Die Prüfung auf Fehler ist in der Tat eine der aufwendigsten Aufgaben eines Compilers. Erst wenn keine Fehler im Sinne dieser Prüfung gefunden werden, wird in der Code Generierungsphase das Objektprogramm erzeugt. Dieses Objektprogramm kann ein Maschinenprogramm sein, das anschließend direkt vom Prozessor ausgeführt wird.

Bloß, für welche Maschinensprache soll man sich entscheiden? Wählen wir den 8088 Prozessor (MS DOS), haben Benutzer eines ATARI ST Probleme, weil dieser einen 68000 Prozessor besitzt. Was ist mit den CP>M Benutzern, deren Maschinen mit einem Z80 ausgerüstet sind? Man kann jedoch einen anderen Weg wählen.

Die Pascalprogramme werden nicht in Maschinensprache eines bestimmten Prozessors übersetzt, sondern in sogenannten P-Code.

P-Code

Bei dem P-Code handelt es sich um eine »künstliche« Maschinensprache für eine virtuelle Maschine. Der erste, von Professor Wirth an der ETH in Zürich entwickelte, Pascal Compiler benutzte auch P-Code als Zielsprache. Das hat den großen Vorteil, daß solche Compiler sehr leicht auf verschiedene Maschinen portiert werden können. Wie bringen wir aber jetzt ein solches P-Code Programm zum Laufen, da die Prozessoren ja nur Maschinensprache verstehen?

Dazu benötigen wir einen P-Code Interpreter, den wir auch im Rahmen dieser Serie vorstellen wollen. Ein solcher P-Code Interpreter ist wesentlich leichter zu schreiben, als einen Compiler an eine neue Maschine anzupassen. Wer auf Geschwindigkeit Wert legt, kann den Interpreter mit minimalem Aufwand in Assembler oder Maschinensprache erstellen. Oder Sie ersetzen den P-Code Interpreter durch einen Code-Generator, der aus dem P-Code die Maschinensprache Ihrer Maschine erzeugt. Dabei können noch Optimierungen durchgeführt werden, die das Objektprogramm besonders schnell oder besonders klein machen.

Überblick

Die Aufgaben und Phasen eines Compilers kann man folgendermaßen unterteilen:

Lexikalische Analyse
(Erkennen der »Worte« eines Programms)

Syntaxanalyse
(Erkennen von grammatikalischen Fehlern)

Semantikanalyse
(Prüfung der »Satzbedeutung«)

- Prüfung der Gültigkeitsbereiche (Scope)
- Prüfung der Typenverträglichkeit
  Code Generierung

Nach diesen mehr prinzipiellen Erläuterungen können Sie sich nun an das »Eingemachte« begeben und sich den ersten Teil der Compilerbau-Serie zu Gemüte führen.

(G. Geißmann)

I M P - ein DFÜ-Programm für CP/M

Nach verschiedenen Beiträgen zum Thema DFÜ in den letzten Clubinfos und der Einrichtung einer eigenen CLUB80-Ecke in der Discovery-Mailbox (vergl. Info 39, S. 8-12), soll heute ein Terminal-Programm aus der Public Domain vorgestellt werden, das auf CP/M-Rechnern immer noch im Einsatz ist.

Neben MEX (in seiner letzten Version 1.14 von 1985) und KERMIT (Version 4.05 von 1986) gehört IMP (Version 2.44 von 1986) zu den bekanntesten DFÜ-Programmen. IMP steht für "Improved Modem Program" = verbessertes Modem-Programm: Man kann damit also - unter Zuhilfenahme eines Modems oder Akustikkopplers - zwei Computer über die Telefonleitung miteinander verbinden. Dabei besteht die Verbesserung gegenüber älteren Programmen hauptsächlich darin, dass nicht nur Übertragungen mit 300 und 1200 Baud (= Bit pro Sekunde) möglich sind, sondern auch noch mit 2400 Baud und in dem damals neu eingeführten MNP-5-Verfahren, wobei der Modem bei auftretenden Übertragungsfehlern die Übertragung automatisch wiederholt oder mit der Übertragungsgeschwindigkeit auf 1200 oder 300 bps zurückgeht.

Ein anderes Merkmal, in dem sich IMP von früheren DFÜ-Programmen unterscheidet, ist der Wegfall lästiger Einstellmöglichkeiten für Mailbox-Parameter. Da sich "8N1" für Übertragungen mit 8 Datenbits, keiner ("none") Parität und 1 Stoppbit inzwischen allgemein durchgesetzt hat, kann man das ja fest einprogrammieren und braucht es nicht mehr vor jeder DFÜ-Verbindung einzeln neu einzustellen.

Entscheidend für die weltweite Verbreitung dieses beliebten Programms war jedoch, dass für alle gängigen CP/M-Rechner eine Anpassung erstellt wurde und die SIG/M in New Jersey (USA) diese in ihre Public Domain Kollektion aufnahm. Das gesamte Paket wurde auf drei Disketten angeboten und steht seither wie folgt zur Verfügung:

SIG/M Volume 272 Improved Modem Program (Volume 1 of 3)
by Irvin M. Hoff
released May 16, 1986

index	name		size	description
272.01	IMP244	.LBR	80K	Improved modem program
272.02	IMP-OVL	.LST	5K	Overlay list
272.03	I2A2-1	.AQM	9K	Apple //c with various cards
272.04	I2A3-2	.AQM	9K	Apple ///
272.05	I2AC-1	.AQM	8K	Actrix computer
272.06	I2AD-1	.AQM	8K	Advanced Digital Super Six
272.07	I2AL-1	.AQM	8K	Altos 8000/series 5
272.08	I2AM-2	.AQM	8K	Ampro "Little Board"
272.09	I2AP-7	.AQM	18K	Apple //e
272.10	I2B2-1	.AQM	8K	Big Board II
272.11	I2BW-2	.AQM	8K	Bondwell 12/14
272.12	I2C8-1	.AQM	11K	Commodore C128
272.13	I2CC-3	.AQM	10K	CCS 2719 and CCS 2830
272.14	I2CP-1	.AQM	8K	CompuPro interfacer 3/4
272.15	I2DM-1	.AQM	8K	Dynabye Monarch
272.16	I2DP-1	.AQM	8K	Datapoint 1560

SIG-M Library -CATALOG Volume Number-272, 16 Files cataloged.

* * *

SIG/M Volume 273 Improved Modem Program (Volume 2 of 3)
by Irvin M. Hoff
released May 16, 1986

index	name		size	description
273.01	I2DV-1	.AQM	8K	Datavue 3000
273.02	I2EA-1	.AQM	10K	Eagle II and III
273.03	I2EP-1	.AQM	8K	Epson QX-10
273.04	I2EQ-3	.AQM	10K	Insight Enterprises EQ-4
273.05	I2H8-4	.AQM	9K	Heath/Zenith 89
273.06	I2HZ-5	.AQM	9K	Heath/Zenith 100 series
273.07	I2KP-1	.AQM	8K	Kaypro with external modem
273.08	I2LO-1	.AQM	8K	Lobo MZX-80
273.09	I2MC-1	.AQM	10K	Molecular Series X
273.10	I2MD-1	.AQM	8K	Morrow MD3 (2 baud gen)
273.11	I2MDA-2	.AQM	13K	Morrow MD3
273.12	I2ME-1	.AQM	8K	Morrow MD5/MD11
273.13	I2MM-1	.AQM	8K	MicroMint SB180
273.14	I2MO-1	.AQM	8K	Morrow MD3 (older units)
273.15	I2MT-1	.AQM	8K	Memotech SDX-512
273.16	I2NA-2	.AQM	8K	North Star Advantage
273.17	I2NH-2	.AQM	8K	North Star Horizon - standard
273.18	I2NS-3	.AQM	9K	North Star Horizon
273.19	I2OS-1	.AQM	15K	Osborne OS-1
273.20	I2OV-2	.AQM	11K	Osborne Vixen
273.21	I2OX-1	.AQM	8K	Osborne Executive
273.22	I2PR-1	.AQM	6K	Penrill 1200-AD modem
273.23	I2R2-1	.AQM	8K	TRS-80 Model II/12
273.24	I2R4-1	.AQM	7K	TRS-80 Model 4 with Montezuma Micro
273.25	I2RS-1	.AQM	8K	TRS-80 Model III,IV,4P/4D,LNW80

SIG-M Library -CATALOG Volume Number-273, 25 Files cataloged.

* * *

SIG/M Volume 274 Improved Modem Program (Volume 3 of 3)
KMD (Xmodem Substitute)
by Irvin M. Hoff
released May 16, 1986

index	name		size	description
274.01	I2S2-1	.AQM	8K	Sanyo 1200/1250 overlay
274.02	I2SB-1	.AQM	8K	Intertec Super Brain
274.03	I2SS-1	.AQM	8K	CompuPro System Support 1
274.04	I2SY-2	.AQM	9K	Sanyo 1000/1100/1200/1250
274.05	I2TV-3	.AQM	10K	Televideo 802/803
274.06	I2US-1	.AQM	8K	US Robotics S-100 plug in modem
274.07	I2VB-1	.AQM	3K	Verbose modem (such as Avatex)
274.08	I2VI-1	.AQM	10K	Visual 1050

```
274.09  I2XE-1  .AQM    8K   Xerox 820
274.10  I2ZB-1  .AQM    8K   Zorba
274.11  KMD17   .LBR  152K   BBS download tool similar to XMODEM
SIG-M Library -CATALOG Volume Number-274, 11 Files cataloged.
```

Während es sich bei den vorgenannten AQM-Dateien um komprimierte Overlay-Dateien mit den systemspezifischen Teilen der entsprechenden Computer handelt, enthält die gepackte Datei IMP244.LBR auf der ersten der oben angegebenen Disketten folgende Einzelprogramme (die Dokumentationen natürlich in englischer Sprache):

	I2FUNC	.COM	4K	zur Änderung einiger Funktionstasten-Belegungen
	I2FUNC	.DOC	2K	Kurzdokumentation zu I2FUNC.COM
	I2LIB	.COM	2K	zur schnellen Änderung einiger Telefon-Nummern (bei automatisch wählenden Modems)
	I2LIB	.DOC	2K	Kurzdokumentation zu I2LIB.COM
	I2NM-2	.ASM	8K	zur Einstellung von SPRINT und MCI, oder zur Änderung der Telefonnummern oder Funktionstasten-Belegungen
	IMP	.DOC	38K	Anleitung zur Benutzung des IMP Terminal-Programms
	IMP	.INF	16K	Information über 1k-Blöcke und den Batch-Übertragungsmodus
	IMP-DISK	.NOT	2K	wie die Speicher-Grösse zur Protokollierung auf Diskette eingestellt werden kann
	IMP-OVL	.CUS	4K	über die Auswahl von Optionen in den Overlays
*	IMP-OVL	.LST	6K	Liste der gegenwärtig verfügbaren IMP Overlay-Dateien in Assembler-Quellcode
	IMP244	.BEL	2K	Information über die mögliche Abschaltung eines Klingelzeichens
*	IMP244	.COM	18K	das IMP.COM Grund-Programm; benötigt zur Anpassung an einen bestimmten Computer noch die entsprechende Overlay-Datei
	IMP244	.DTR	2K	Information über DTR zum Abbrechen der Telefon-Verbindung
	IMP244	.UPD	4K	berichtet, was an dieser IMP-Version neu ist
	MLOAD	.COM	4K	verbindet IMP.COM mit der entsprechenden Overlay-Datei, nachdem diese assembliert wurde

Nach dieser IMP-Version 2.44 kam 1987 auch noch eine Version 2.45 heraus. In wie weit die älteren Overlay-Dateien damit zusammenarbeiten, bzw. für welche Computer neue Overlays geschrieben wurden, entzieht sich allerdings unserer Kenntnis.

Wenn man nun ein fertig angepasstes Programm IMP.COM in seinen Rechner lädt, erhält man beispielsweise beim Commodore 128 folgende Bildschirm-Anzeige, und nach Eingabe von "m" auf drei Bildschirmen das nachfolgende Menu:


```
IMP v244 modem pgm (type M for Menu)
Copyright (c) 1985 Irvin M. Hoff
Commodore C128 Version
Modem speed is: 1200 bps

>>COMMAND:m

                    Single Letter Commands

 ?  - Display current settings
 ^  - Function key intercept character, then (0-9)
 M  - Display the menu
 E  - Terminal mode with echo
 L  - Terminal mode with local echo
 T  - Terminal mode
^Z  - Clears screen (command mode only)
 R  - Receive an 8-bit binary CP/M file
 S  - Send an 8-bit binary CP/M file

      COMMAND: R (or S) FILENAME.TYP
      R and S can use the following subcommands:
         B  - Batch transfer, can use wildcards (e.g., *.*)
         D  - Disconnect when done
         K  - Manual request for 1k transmissions
         Q  - Quiet mode (no messages to console)
         V  - View ASCII files on CRT during a file transfer
         X  - Inhibits auto 1k request to sender
         Z  - When done, disconnect, go to CP/M

      For copying text to disk use T (E or L) FILENAME.TYP

ÄmoreÜ                  Three Letter Commands

BYE - Disconnect, then return to CP/M
CAL - Dial number
CPM - Exit from this program to CP/M
DIR - List directory and space free (may specify drive)
DSC - Disconnect from the phone line
ERA - Erase file (may specify drive)
LOG - Change default drive/user no. (specify drive/user)
      and reset disks i.e., LOG A0: or LOG B: (user # unchanged)
SET - Set modem baud rate
SPD - Set file output speed in terminal mode
TBM - Toggle MODEM7/KMD batch mode selection
TCC - Toggle CRC/Checksum mode on receive
TLF - Toggle LF after CR in "L" or "T" mode for a disk file
TRB - Toggle rubout to backspace conversion

      The following are terminal text buffer commands:

DEL - Delete memory buffer and file
WRT - Write memory buffer to disk file

ÄmoreÜ             Local Commands while in Terminal Mode

ESC-E - Exit to command mode
ESC-F - Send file to remote system
```

ESC-L - Send log-on message
ESC-N - Disconnect from the phone line
ESC-P - Turn printer on (or off)
ESC-Z - Clears screen, terminal mode

ESC-R - Stop copy into buffer
ESC-Y - Start copy into buffer

Start and stop may be toggled as often as desired.
A ";" at start of line indicates buffer is copying.
X-off automatically used to stop input when writing full buffer to disk, X-on sent to resume.

>>COMMAND:?

Current Settings

Mode: CRC
KMD batch mode
Rub is backspace
Printer buffer is OFF
Modem speed is: 1200 bps
Terminal mode file buffer is inactive
Unused portion of buffer is 16384 bytes
Use ESC before local command in terminal mode
XOFF testing used in terminal mode file output
LF NOT sent after CR in "L" or "T" for a disk file

Char. delay (terminal file mode) is: 2 ms. per character
Line delay (terminal file mode) is: 200 ms. per CR character

>>COMMAND:

Nach Eingabe eines "?" nach COMMAND: erhält man also die derzeitigen Einstellungen des Programms ("Current Settings"). Da der C-128 höchstens 1200 Baud verkraftet, ist diese Übertragungsgeschwindigkeit voreingestellt; sie kann mit dem Befehl "set" aber auch noch auf 300 bps geändert werden.

Darunter wird angezeigt: "Terminal mode file buffer is inactive", was bedeutet, dass der Pufferspeicher zur Protokollierung ankommender Daten ausgeschaltet ("inactive") ist. Darunter wird der zur Verfügung stehende Speicher im Rechner mit 16 384 Bytes beziffert. Durch Eingabe des Befehls "ESC und Y" (jedoch erst im Terminal-Modus, wenn man eine Telefon-Verbindung aufgenommen hat) lässt sich der Pufferspeicher jederzeit einschalten, und mit "ESC und R" auch wieder aus. Auf dem Bildschirm ist dann an den ";" am Beginn jeder Zeile zu erkennen, ob der Puffer eingeschaltet ist.

Da der Inhalt des Pufferspeichers, sobald dieser voll ist, automatisch auf eine Diskette kopiert wird, muss man vor Aufnahme einer DFÜ-Verbindung auch noch eine leere Diskette bereitstellen und dem Rechner mitteilen, unter welchem Namen die angekommenen Daten dort abgespeichert werden sollen. Beim C-128 hat man dabei zwei Möglichkeiten: Besitzt man eine RAM-Disk (z. B. die REU 1750), dann kann man zunächst diese verwenden und mit M: ansprechen. Andernfalls wechselt man im Laufwerk A: die IMP-Programmdiskette (die nach Laden des Programms nicht mehr gebraucht wird) gegen eine frisch formatierte Diskette aus. Falls die angekommenen Daten jedoch nicht auf der Diskette in A: sondern in der RAM-Disk M: abgespeichert werden sollen, meldet man das dem Rechner mit dem Befehl:

A>>COMMAND:log m:

und kann gleich anschliessend nachprüfen:

M>>COMMAND:dir

worauf angezeigt wird:

Drive M:
++ FILE NOT FOUND! ++
508k bytes free on drive M:

Nun kann man seine Mailbox anwählen und nach Zustandekommen der Telefon-Verbindung das Programm in den Terminal-Modus umschalten. Das geschieht mit dem Befehl "t", und danach gibt man gleich auch noch den Dateinamen für das abzuspeichernde Protokoll ein, beispielsweise "discover.txt":

M>>COMMAND:t discover.txt

Jetzt wird also die Datenfernübertragung durchgeführt, und mit "ESC und Y" kann wie gesagt auch der Pufferspeicher zugeschaltet werden.

Am Ende der Übertragung, wenn die Telefon-Verbindung wieder abgebrochen ist, bleibt der Cursor auf dem Bildschirm zunächst irgendwo stehen. Nun ist es wichtig, durch Eingabe von "ESC und E" aus dem Terminal-Modus wieder in den Command-Modus zurück zu schalten. Es wird dann folgende Warnung angezeigt:

** File still open, use DEL, DIR, WRT, E, L or T **

... Bytes of buffer free

Mit Hilfe des Befehls "wrt" (wie write = schreiben) wird der restliche Puffer-Inhalt dann noch in die Protokoll-Datei geschrieben und diese Datei geschlossen:

M>>COMMAND:wrt

DISCOVER.TXT closed

M>>COMMAND:cpm

Mit dem Befehl "cpm" kehrt man aus IMP wieder zum CP/M-Betriebssystem zurück, und kann dann nach Laden einer Textverarbeitung wie WORDSTAR in Ruhe die Protokoll-Datei DISCOVER.TXT durchsehen.

(Günther W. Braun)

Internet und UUCP mit CP/M

Ein Bericht über die Installation von UUCP 2.1b auf CP/M Plus.

Das Internet

Das Internet ist eines der größten Computer-Netze, die es überhaupt gibt. Es handelt sich hierbei um einen internationalen Zusammenschluß von Computern mit dem Unix Betriebssystem, der europäische Teil dieses Netzes heißt Eunet. Internet benötigt neben dem Unix-Rechner auch eine Standleitung, was das Ganze teuer macht. Im Gegensatz zu FIDO oder ZERBERUS zum Beispiel kostet der Account in einer Internet-Box Geld. Die monatlich zu entrichtenden Beträge betragen im günstigsten Fall 10,- DM für Schüler bis zu mehreren hundert Mark für Firmen.

Internet gibt zum einen die Möglichkeit, persönliche Mails rund um die Welt zu schicken. Es gibt außerdem Gateways zu anderen Netzen wie ZERBERUS oder lokalen Netzen wie in Kiel das KIEL-NETZ. Ausserdem bietet Internet Zugriff auf mehrere hundert Newsgroups, vergleichbar mit den Brettern aus den anderen Netzen. Die meisten Mailboxen haben Software Archive und gegen einen erhöhten Monatsbeitrag kann man **'ftp','file transfer protocol'**, nutzen, und so online von anderen Rechnern Files downloaden. Es gibt meines Wissens ausser dem Monatsbeitrag keine Gebühren für übertragene Information. Bei der Kieler Box **'Toppoint'** wird jeder User angehalten, monatlich nicht mehr als 500 KByte private Mail zu versenden.

Ausserdem bieten Internet-Anschlüsse den Zugriff auf einen Unix-Rechner, man ist dort ganz normaler User und kann etwas in die Unix-Welt hineinschnuppern.

Der andere Grund für mein Interesse ist die Newsgroup **'comp.os.cpm'**. Von allen Brettern, die das Wort CPM in sich führen, die ich in anderen Boxen kennengelernt habe, ist hier am meisten los. Man trifft hier auf so bekannte Namen wie Jay Sage, Joe Wright oder David Goodenough. Natürlich werden hier auch meistens nur Fragen nach bestimmten Systemen gestellt oder hie und da mal ein Computer zum Verkauf angeboten, aber man behält das Gefühl daß man mit CP/M noch nicht ganz hinter dem Mond lebt.

UUCP

In der Unixwelt gibt es einen Standard, um Files, also auch Nachrichten oder Briefe von einem Unixsystem zu einem anderen zu übertragen. UUCP heißt **'Unix to Unix Copy'**. Zu diesem Verfahren gehören eine ganze Menge Files. SMAIL bereitet Files zum Transfer vor. Die mit einem Editor geschriebenen Files werden um z.B. den **'From: <username>'**-Eintrag sowie den Pfad zurück zu diesem System und einer fortlaufenden Nummer, um die Nachricht genau identifizieren zu können erweitert, und unter einem anderen Namen in einem Verzeichnis abgelegt, nennen wird es 'transferdrive'. Und es wird ein zweites File

erzeugt, welches beschreibt, was mit diesem File auf dem Zielsystem geschehen soll, also zum Beispiel 'Sende dieses File weiter an irgendwohin'.

Sollen Files übertragen werden, ruft man UUCP auf, es funktioniert ähnlich wie PIP, bloß daß auch UUCP die Files nicht sofort überträgt, sondern auch eine Art Batchfile erzeugt. Um diese Files dann wirklich zu übertragen wird UUCICO benötigt. Es stellt eine Verbindung zu einem anderen Rechner her, der nach dem login ein UUCICO im Slavemodus als Shell startet. Die beiden Rechner machen etwas 'handshake' um das Protokoll zum Übertragen zu klären und dann werden die Files in Paketen übertragen. Wenn der Slave nun auch Files für den Master hat, wechseln beide ihre Rollen und das ursprünglich angerufene System sendet nun seinerseits die Files blockweise. Ist die Übertragung beendet, wird die Leitung unterbrochen.

Wurden nun Files empfangen, so müssen sie aus dem oben erwähnten Format in die private Mailbox (File) einsortiert werden. Sie heisst üblicherweise <USERNAME>.MAI. Mit einem Mailreader können so die empfangenen Nachrichten mehr oder weniger komfortabel zu Hause bearbeitet werden.

Um UUCP nutzen zu können, muß man zunächst also erstmal einen normalen Account auf einer Box, die am Internet angeschlossen ist, erwerben. Dort kann man natürlich Online Mails und News verschicken und lesen. Die dort vorhandenen Mail- und News-Reader sind sehr komfortabel. Meistens handelt es sich um den News-Reader **NN** (No News - are good news !) und für private Mails ist meistens **ELM** vorhanden. Als User hat man natürlich eine email-Adresse, sie setzt sich zusammen aus Usernamen (bei mir **'v_p_dose'**) und einer Art Pfad inklusive Namen der Mailbox. Die Kieler Box heisst **'toppoint'**, die vollständige Adresse lautet bei mir:

```
'v_p_dose§tpki.toppoint.de'.
         !  !     !     !- Nationalität
         !  !     !- Name der Mailbox
         !  !- 'hostname' der Mailbox
         !- Kennung für Internet Adresse
```

Für UUCP benötigt man einen zweiten, unabhängigen Account. Auf der Toppoint heisst dieser Vorgang **'Point beantragen'**. Man erhält einen zweiten Usernamen, und nach dem Login wird keine normale Unix-Shell oder ein Menü-Programm gestartet, sondern UUCICO im Slave-Mode. Als zweiten Usernamen wählte ich das Wort **'cpmgut'**, die email-Adresse dieses neuen Points ist **'v_p_dose§cpmgut.toppoint.de'**.

UUCP und CP/M

Nun gibt es auf der J.Sage Disk #27 das File UUCP21.LBR. Der Author ist David Goodenough, er schrieb auch das geniale Terminalprogramm QTERM.COM! Das Paket besteht aus 9 Files, die es ermöglichen von einem CP/M-System aus Verbindung über UUCP zum Internet und auch zu einem einzelnen Unix-Rechner aufzunehmen.

Folgende Files werden hierzu benötigt:

- SMAIL.COM verarbeitet zu sendende private Mails
- RMAIL.COM verarbeitet mit UUCP empfangene private Mails

- PNEWS.COM — verarbeitet Mails, die in eine Newsgroup eingeordnet werden sollen
- RNEWS.COM — sortiert eingegangene News in einzelne Newsgroups

- UUCP.COM — ermöglicht File transfer zwischen Rechnern
- WAIT.COM — ermöglicht zeitgesteuertes Ablaufen von SCCICO.COM o.ä.

- CCICO.COM — aktiv, entspricht UUCICO im Master-Mode
- SCCICO.COM — passiv, entspricht UUCICO im Slave-Mode

- MAIL.COM — der Mail-Reader

außerdem sind enthalten:

- UUCP.DOC — Dokumentation zum Paket
- QTCHAT.DOC — Dokumentation zu Chatfiles bei QTERM, gilt auch für CCICO

Die Dokumentation zu UUCP ist zwar recht kurz, enthält aber sehr viel Informationen. Der Author ging aber davon aus, daß grundsätzliches Vorwissen um UUCP vorhanden sei, ich hatte z.T. erhebliche Schwierigkeiten.

Ich werde das DOC File nicht komplett übersetzen, sondern nur dort kommentieren, wo ich Schwierigkeiten hatte. Mein Wissen ist aus bitterer Erfahrung und aus diversen man-pages (Online-Handbuch) von Unix sowie einem File mit FAQs zum Thema UUCP gespeist.

INSTALLATION

Die Erstellung eines QTERM-ähnlichen Patches für CCICO, SCCICO und WAIT sollte kein großes Problem sein, es gibt dieses Overlay für sehr viele Rechner und für CP/M Plus gibt es eins, welches die AUX: Schnittstelle benutzt. Dies ist sehr praktisch zum Testen, wenn man zwei serielle Schnittstellen hat, und ein Terminal um die Programmausgaben zu beobachten. Ich habe diese Overlay disassembliert und die Terminal Control Codes für das Genie IIIs eingefügt. Bei Interesse sende ich es gerne jedem zu.

Da RMAIL, SMAIL, PNEWS und RNEWS Zeit und Datum des Empfangens bzw. des Sendens eintragen, benötigen sie, ebenso wie WAIT einen Overlay, um die aktuelle Uhrzeit samt Datum in einem bestimmten Format zu erhalten. Dieses Routine ist natürlich vom jeweiligen Rechner abhängig, obwohl für CP/M Plus ein kompaktibles OVL geschrieben werden kann. Ich habe aber eine BIOS-Routine des Holte-BIOS benutzt. Es dürfte nicht allzu schwierig sein, so eine Routine zu implementieren. Die Voraussetzung ist natürlich eine fortlaufende Uhrzeit auf dem Rechner, besser natürlich eine Hardware-Uhr.

Mit SETDRV.COM wird im jeweils letzten Sektor jedes relevanten Files eine User/Laufwerksbezeichnung eingetragen, in dem sich das CONFIG.SYS File befindet. Dies muß nicht dieselbe sein, in der die COM-Files sich befinden.

CONFIG.SYS

Am meisten Fehler machte ich bei der Anpassung von 'CONFIG.SYS'. Dieses File enthält mehrere User- und auch Systemspezifische Informationen. Es ist ein ganz normales ASCII File mit mehreren einzeiligen Zuweisungen.

'hostname' – Hier muss der Name des Systems eingetragen werden, er ist identisch mit dem Usernamen des Points, bei mir also:

'hostname   v_p_dose'

'user' – Dieser Eintrag spezifiziert den bzw. die User die auf dem 'host' angesprochen werden können. Sinnvollerweise steht hier der eigene Username plus der ausgeschriebene richtige Name. Ich habe bei mir die vollständige email-Adresse eingetragen, bin mir aber nicht ganz sicher, ob das so richtig ist. Zumindest PNEWS erzeugt im Briefkopf hinter dem Usernamen noch den String '§<hostname>.UUCP'. Das wäre auch die richtige Adresse, aber Post soll ja an meinen Point in der Mailbox gehen und nicht direkt an meinen Computer zuhause. SMAIL erzeugt aus meinem Eintrag aber die richtige (also die Point-) Adresse.

'user   v_p_dose§cpmgut.toppoint.de (Volker Dose)'

Wird in einer zu versendenden Mail kein 'From:' eingetragen, so wird von SMAIL und PNEWS der erste User in dem CONFIG.SYS-File als Absender eingetragen. Ausserdem benutzt MAIL diesen Namen, wenn es ohne Parameter gestartet wird. Dies funktioniert aber nicht, wenn die vollständige email Adresse eingetragen wurde.

'defsys' – Hier wird angegeben, welche Systeme mit CCICO angerufen werden sollen. Jedes System hat eine Kennung bestehend aus drei Buchstaben, diese Kennung wird auch als Fileextension benutzt. Es können mehrere solcher Buchstabenkombinationen angegeben werden, wird CCICO dann ohne die '-S' Option (nur ein System !) aufgerufen, werden alle in der Kette enthaltenen Systeme angerufen. (siehe 'remote'!)

'defsys   TPK'

'xferdrv' – gibt eine User/Laufwerksbezeichnung an, in die die temporären Files abgespeichert werden. Sinnvollerweise sollte man hier eine Ramdisk angeben, so vorhanden. Es kann die Usernummer mit angegeben werden. Der Doppelpunkt muß mit eingegeben werden.

'xferdrv   m7:'

'workdrv' – Auf diesem 'Drive' liegen sowohl die Scriptfiles als auch die Mailboxfiles. Diese Files sind von längerer Gültigkeit.

'workdrv   d7:'

'remote' – Diese Einträge bezeichnen die Systeme, die angesprochen werden können. Der erste Eintrag muß der eigene 'hostname' sein. Jeder Eintrag setzt sich aus einer drei Buchstaben langen Folge sowie dem kompletten 'hostname' zusammen. Es sind also mindestens zwei Einträge nötig. Nach dem Login sendet der Slave einer UUCP-Verbindung eine Kennung '^PShere=<hostname>' und der Master sendet '^PS<hostname>'. Bei mir sendet also die Mailbox '^PShere=tpki' woraufhin CCICO '^PScpmgut' sendet.

'remote   CPMcpmgut'
'remote   TPKtpki'

'alias' - Solche Aliasnamen werden von SMAIL und RMAIL eingesetzt. Allerdings nicht in der 'To:' Zeile einer Mail sondern nur in dem zur Mail gehörenden D-File bzw. X-File (siehe weiter unten)

```
alias        norbert        nobs§isys-hh.hanse.de
alias        jay            sage§ll.mit.edu
```

'$drv' - Hier wird die Laufwerksbezeichnung eingetragen,in dem der CCP das $$$.SUB File bei der Batchabarbeitung erzeugt.

```
'$drv        m:'
```

'smarthost' - Die Bedeutung dieses Eintrages ist mir noch unklar, meiner scheint aber zu funktionieren.

```
'smarthost   cpmgut.toppoint.de!v_p_dose'
```

'inethost' - Dies gibt einen Host an, der mit Adressen im Internet-Styl weiterleiten kann. Also die Internet-Mailbox.

```
'inethost    tpki.toppoint.de'
```

'progdrv' - Dies ist die Laufwerksbezeichnung, die angibt, wo sich die COM-Files befinden. Insbesondere WAIT braucht diese Angabe, außerdem wird sie bei der Batch-Abarbeitung gebraucht.

```
'progdrv     d7:'
```

'feed' - Dies ist das System, zu dem PNEWS die zu übertragenen Nachrichten verschickt. Also wieder die Internet-Mailbox.

```
'feed        tpki'
```

'newsgroup' - RNEWS sortiert eingegangene Nachrichten in verschieden Mailboxen. Das sind Files mit der Extension 'MAI'. In jedem Eintrag steht also der Filename exklusive Extension.

```
'newsgroup   comp.os.cpm        cpm'
'newsgroup   tp.sysinfo         tpsysinf'
'newsgroup   tp.hilfe           tphilfe'
'newsgroup   tp.forum           tpforum'
'newsgroup   tp.test            tptest'
'newsgroup   kiel.biete         ki-biete'
'newsgroup   kiel.suche         ki-suche'
'newsgroup   kiel.allgemein     ki-allge'
'newsgroup   comp.sys.tandy     tandy'
```


ERLÄUTERUNGEN ZU DEN EINZELNEN PROGRAMMEN

SMAIL.COM

SMAIL heißt 'SEND MAIL', dient also zum Verschicken von privaten Nachrichten. Jede Mail wird mit einem Editor wie z.B. ZDE geschrieben und enthält einen Kopf sowie die tatsächliche Nachricht. Der Kopf muss mindestens die Zieladresse (To:) enthalten, er kann aber noch ein Thema (Subject:) oder auch einen Pfad (Path:) und diverses anderes enthalten. Wird kein Absender (From:) angegeben, setzt SMAIL den ersten User aus CONFIG.SYS hier ein. Aus einer kleinen Nachricht, abgespeichert in TEST.OUT

```
--------------------------------------------------------------
To: nobs§isys-hh.hanse.de
Subject: Test Nachricht

Hallo Norbert.
Ist dies angekommen ?
--------------------------------------------------------------
```

erzeugt der Befehl 'SMAIL TEST.OUT' im 'xferdrv' vier Files.

- C--0A500.TPK
- C--0A501.TPK
- D-D59E1B.CPM
- D-D59E1C.CPM

Die C-Files (Command-Files) enthalten Befehle für CCICO. Sie sind bestimmt für das System mit der Abkürzung 'TPK', also 'tpki'.

Der Inhalt vom ersten C-File :

```
'S D.cpmguA0a500 D.cpmguA0a500 uucp - m7:D-d59e1b.CPM 644'
```

Das 'S' steht für 'Senden', die beiden Teilstrings die mit 'D.cp..' beginnen dienen zur Identifizierung und der String ab 'uucp' ist dann der eigentliche Befehl. Heißt : 'Unix to Unix Copy das File auf M7: mit dem Namen.' Die Bezeichnung 644 ist eine Oktal Zahl die die Art des Files angibt. Das erste C-File enthält also den Befehl zur Übertragung des ersten D-Files (Daten-File).

Der Inhalt des zweiten C-Files:

```
'S D.cpmguA0a501 X.cpmguA0a501 uucp - m7:D-d59e1c.CPM 644
```

Es enthält den Befehl zum Versenden des zweiten Datenfiles.

Das erste Datenfile enthält die zu versendende Nachricht :

```
From v_p_dose§cpmgut.toppoint.de  Tue Jul 27 15:42:34 1993
remote from cpmgut
Received: by cpmgut.UUCP (CP/M MAIL V2.1b)
id XX000000a5; Tue, 27 Jul 93 15:42:34 MSZ
To: nobs§isys-hh.hanse.de
Subject: Test Nachricht
From: cpmgut!v_p_dose§cpmgut.toppoint.de (Volker Dose)
Date: Tue, 27 Jul 93 15:42:34 MSZ
Message-Id: <XX000000a5§cpmgut.UUCP>

Hallo Norbert.
Ist dies angekommen ?
```

Der Absender ist korrekt eingetragen worden, es sei beachtet, daß sich neben der ersten Zeile mit einem 'From'-Eintrag ohne Doppelpunkt noch ein zweiter Eintrag weiter unten befindet, die durchaus verschieden sind.

Die abgeschickte Mail ist also von 'cpmgut.UUCP' empfangen worden. Sie wird weitergeleitet nach dem 'inethost', also in Richtung tpki. Wenn sie dort empfangen wird kommt ein weiterer 'Received'-Eintrag dazu, wenn Files über weite Entferungen zum Beispiel nach USA übertragen werden, kommen schnell vier bis fünf 'Empfänger' dazu, die die Nachricht weiterleiten (to forward) . SMAIL hat auch das Datum des Versendens eingetragen. Es ist aber beileibe noch nicht wirklich losgeschickt, dies besorgt erst ein CCICO-Lauf.

Der Eintrag <Message-Id:> sorgt für die Identifizierung dieser Mail. Es ist die 'OA5H'te übertragene Mail. Diese Ziffer wird in dem File 'SEQ.NUM' gespeichert und weitergezählt. SEQ.NUM wird auf 'progdrv' abgelegt.

Das zweite D-File wird auch auf den 'inethost' übertragen und enthält Befehle, die auf dem System ausgeführt werden.

```
U uucp cpmgut
Z
R uucp
F D.cpmguA0a500
I D.cpmguA0a500
C rmail nobs§isys-hh.hanse.de
```

Die Adresse in der letzten Zeile würde von SMAIL eingesetzt werden, wenn anstatt der ganzen Adresse nur der Alias 'norbert' eingetragen worden wäre.

Wird SMAIL mit der Option '-O' gestartet, so werden diese abgeschickten Mails in der Mailbox (File) 'OUTMAIL.MAI' abgespeichert. So können ausgehende Mails gesichert werden.

RMAIL.COM

RMAIL soll nach einem CCICO-Lauf eingegangene Post in die Mailbox <username>.MAI, in meinem Fall 'v_p_dose.mai' einsortieren. Ist Post an einen anderen User auf diesem System eingegangen, so wird eine andere Mailbox erzeugt (Ich habe eine Mail an 'v_p_§cpmgut.toppoint.de' erhalten, es wurde die Mailbox 'v_p_.mai' erzeugt).

Die eingegangenen Files sind, analog zu den ausgehenden, wieder zwei stück pro Mail, ein Datenfile mit einer oder mehreren Nachrichten, sowie ein Command-File, dieses ist mit 'X' bezeichnet, für 'Execution'. Einige Post, die an mich gerichtet war, enthielt in diesen X-Files den Befehl 'rmail cpmgut.toppoint.de!v_p_dose'. Dies wurde von RMAIL nicht als meine Adresse erkannt, sondern die Nachricht wurde in Richtung 'inethost', also zur toppoint hin, weitergeleitet. Aus dem D- und dem X-Files wurden, wie bei SMAIL beschrieben, zwei C-Files und zwei D-Files. Mir ist nicht klar, warum diese Mail an mich mit dieser Adresse geschickt wurde, andere Mails sind einwandfrei angekommen. Es muß also in dem X-File stehen: 'rmail v_p_dose'. Das muß ich bis jetzt leider kontrollieren und von Hand ändern, da die Post sonst verloren geht.

PNEWS.COM

PNEWS dient vzum Versenden von Nachrichten in 'Newsgroups', also in öffentliche Bretter. Die Nachricht muß ähnlich wie die private Mail aufgebaut sein, allerdings muß ein Kopfeintrag 'Newsgroups:' vorhanden sein. PNEWS erweitert Mails ähnlich wie SMAIL. Aus der Nachricht:

```
Newsgroups: comp.sys.cpm
Subject: UUCP with CP/M

Hello.
UUCP works !!!!!
```

---

werden zwei C-Files analog zum Beispiel bei SMAIL und zwei D-Files.

Das eine enthält :

```
Path: cpmgut!v_p_dose§cpmgut.toppoint.de
From: v_p_dose§cpmgut.toppoint.de§cpmgut.UUCP (Volker Dose)
Newsgroups: comp.sys.cpm
Subject: UUCP with CP/M
Date: Tue, 27 Jul 93 16:24:04 MSZ
Message-Id: <XX000000a6§cpmgut.UUCP
Hello.
UUCP works !!!!!
```

Der Absender ist hier eindeutig falsch! Wenn man in der Nachricht einen 'From:' Eintrag einfügt, mit der kompletten email Adresse, so bleiben die Einträge 'Path:' und 'From:' (oberhalb des Eintrages 'Newsgroups') leer, bzw. erscheint 'Path: cpmgut!' und 'From: §cpmgut.UUCP'. Unter dem Eintrag 'Subject:' erscheint dann aber der richtige 'From:'-Eintrag.

Auch PNEWS zählt im File SEQ.NUM die Anzahl der herausgehenden Mails.

Da die Frage der Adresse bisher nicht abschliessend gelöst ist, benutze ich ein Fußteil am Ende jeder Mail, in der meine richtige email Adresse angegeben ist. Mit dem Block-Einlesekommando von ZDE ist so etwas schnell eingefügt.

Meine 'signature':

```
---
Volker Dose                                  email: v_p_dose§cpmgut.toppoint.de
Dorfstrasse 10
D-24235 Brodersdorf
Tel.: 04343/1357                       CP/M is not dead, it just smells funny !
```

RNEWS.COM

RNEWS sortiert eingegangene Nachrichten in die verschiedenen Newsgroup-Mailboxen (siehe 'newsgroup'-Einträge im CONFIG.SYS). Die Internet-Mailbox darf die Nachrichten nicht komprimieren. Scheinbar können die äquivalenten Programme unter MS-DOS und Unix etwas mehr. Meine News wurden mir auch komprimiert zugesandt, mit ZPATCH entdeckte ich am Beginn eines jeden solchen Files den String '!# cunbatch'. Das ist eine Anweisung an RNEWS die Files zu entpacken. Der oder die Sysops der Mailbox können aber auch auf nicht komprimierte Übertragung umstellen. Mir ist noch nicht klar, wie man die Newsgroups bestimmt, die man erhalten will. Bei dem NN (siehe oben) gibt es ein File '.newsrc', in dem die gewünschten Bretter markiert werden. So etwas ähnliches muß auch für den Pointbetrieb existieren.

MAIL.COM

MAIL ist der Offline Reader sowohl für die eingegangenen privaten Mails als auch für die Newsgroups. Mit der Option '-u' kann die 'Mailbox' (File) angewählt werden. Ich habe mir einen Alias geschrieben, der den ZFILER mit der Option 'UUCP:*.MAI' aufruft. So kann ich die gewünschte Mailbox anwählen und mit 'ESC M' wird MAIL mit dem markierten File aufgerufen.

MAIL ist nicht besonders komfortabel, der CRR für das FIDO-Netz ist erheblich mächtiger. Insbesonders kann man nicht sehr einfach auf eine Mail antworten, man muß das File mit 's' abspeichern und dann den Editor aufrufen. Ich habe mir einen ZDE-Macro geschrieben, der 'quoting' erlaubt, in jede Zeile wird also ein '> TAB' eingefügt.

CCICO.COM

Dieses Programm leistet die Übertragung zum Zielsystem. CCICO entspricht in der Unix-Welt einem UUCICO im Master-Modus. SCCICO wäre UUCICO im Slave-Modus. Wird CCICO aufgerufen, sucht es ein Chatscript, mit der Extension des gewünschent Systems. Bei mir heißt dieses File 'SYSTEM.TPK'. Wenn dies gefunden wurde, wird zunächst einmal der String '+++ATHO' an das Modem gesendet, um eine eventuell zu bestehende Verbindung zu trennen. Dann wird das Scriptfile abgearbeitet. Es handelt sich um ein File ähnlich den Chatfiles, die QTERM verarbeiten kann. Dieses File soll die Internet-Box telefonisch anwählen und das Login durchführen. Meine Mailbox besitzt 7 Modemports, diese sieben Telefonnummern werden dreimal nacheinander gewählt. Wenn vom Modem entweder 'BUSY' oder 'NO CARRIER' gesandt wird, wird zur nächsten Telefonnummer weitergegangen. Sendet das Modem 'CONNECT' beginnt das Login.

Filename: SYSTEM.TPK

```
!: reset
!§ z z + 1
!# z > 3 'exit
!> Das Modem wird initialisiertörön
 .ATMOV1X4B0ör.OKörön.5.5
!> Die erste Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
 .ATDP0431549061ör..
!Ä - 10 .CONNECT.BUSY.NO CARRIER.
!Ä = CONNECT 'connect
!: nummer2
!> Die zweite Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
 .ATDP0431549062ör..
!Ä - 10 .CONNECT.BUSY.NO CARRIER.
!Ä = CONNECT 'connect
!: nummer3
!> Die dritte Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
 .ATDP04315490063ör..
!Ä - 10 .CONNECT.BUSY.NO CARRIER.
!Ä = CONNECT 'connect
!: nummer4
!> Die vierte Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
 .ATDP04315490064ör..
!Ä - 10 .CONNECT.BUSY.NO CARRIER.
!Ä = CONNECT 'connect
!: nummer5
!> Die fuenfte Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
 .ATDP04315490065ör..
!Ä - 10 .CONNECT.BUSY.NO CARRIER.
!Ä = CONNECT 'connect
!: nummer6
!> Die sechste Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
```

```
.ATDP0431549066ör..
!Ẍ - 10 .CONNECT.BUSY.NO CARRIER.
!Ẍ = CONNECT 'connect
!: nummer7
!> Die siebte Tel.-Nummer wird gewaehltörön
 .ATör.OKörön.5.5.
 .ATDP0431549067ör..
!Ẍ - 10 .CONNECT.BUSY.NO CARRIER.
!Ẍ = CONNECT 'connect
!Ẍ = BUSY 'reset
!Ẍ = NO CARRIER 'reset
!# a = a 'reset



!: exit
!> Das script ist beendetörön
 .ATH0ör.OKörön.5.5.
!Q

!: connect
!> Das login beginntö7örön
 ..Login:.                          ; Das <SPACE. in diesen drei
 .cpmgutör.Password:.               ; Zeilen muß natürlich entfernt
 .PASSWORDör..                      ; werden.
!d
```

In der Wählschleife dürfen keine leeren Zeilen stehen. Ist dieser Login-Vorgang beendet beginnt eine 'handshake'-Periode. Dies läßt sich sehr schön verfolgen, wenn CCICO mit dem Parameter '-X127' aufgerufen wird.

Jeder gesendete String beginnt mit einem '^P' und endet mit einer binären Null. Dieser **'initial handshake'** geht folgendermaßen ab:

called = das angerufene System
caller = das anrufende System, hier CCICO

called : **'Shere=<hostname>'**	'hostname' ist der UUCP Name des angerufenen Systems
caller : **'S<hostname>'**	das anrufende System sendet seinen Namen
called : **'ROK'**	das angerufene System sendet 'OK', wenn der <hostname> in Ordnung war.
called : **'P<protocols>'**	das angerufene System übermittelt, mit welchen Protokollen die nachfolgende Übertragung stattfinden kann. üblich sind das 'g' und das 'e' Protokoll. Das 'g' Protokoll ist Paketorientiertund Fehlerkorrigierend. Es wird von allen UUCP Installationen unterstützt. Bei der CP/M Version ist die Paketgröße auf 64 Byte begrenzt.
caller : **'U<protocol>'**	Das CP/M UUCP sendet 'Ug'.

Zum Abschluß des 'initial handshake' werden die Strings 'INITA', 'INITB' und 'INITC' gesendet und zurückgeschickt. Jetzt beginnt die eigentliche Übertragung mit dem gewählten Protokoll (hier 'g').

Der **Master** hat die Möglichkeit vier verschiednene Befehle zu senden. CCICO liest nacheinander alle C-Files auf dem 'xferdrv' (siehe SMAIL.COM). Auf so einen 'Sende'-Befehl reagiert der Slave mit 'SY', wenn er bereit zum Empfangen ist. Ist der Slave nicht bereit sendet er 'SN4', wenn der Transfer im Moment nicht durchgeführt werden kann oder 'SN2', wenn der Transfer niemals gelingen kann, falls zum Beispiel auf das Ziel-Directory nicht zugegriffen werden kann.

Ist der Transfer geglückt, sendet der Slave 'CY'. Der Master bearbeitet nun das nächste C-File. Sind diese alle abgearbeitet, so sendet CCICO ein 'H', womit er signalisiert, daß von seiner Seite aus die Verbindung getrennt werden kann. Der Slave sendet nun seinerseits 'HY', wenn er auch dieser Auffassung ist. Er kann aber auch 'HN' senden, dies bedeutet, daß der Slave jetzt Master werden will, um Files zum ursprünglichen 'caller' zu übertragen. Die Systeme tauschen dann die Rollen, es werden Kommandos wie in den C-Files beschrieben, ausgeführt. Sind nun alle Files von dem angerufenen System übertragen, sendet es wieder 'H' und CCICO antwortet mit 'HY. Dieses 'HY' wird geechot und die Übertragungsphase ist beendet.

Es folgt der 'final handshake'. Der Master sendet 'CLOSE', der Slave antwortet mit 'CLOSE' und zum Schluß sendet der 'caller' '000000', also sechs ASCII-Nullen, die so geechot werden. Dananch unterbricht der 'called' die Verbindung. CCICO sendet hiernach noch einmal '+++ATH0' an das Modem und kehrt dann zum CCP zurück.

UUCP.COM

UUCP nun dient zum übermitteln von Files von einem System zu einem anderen. UUCP erzeugt pro zu übertragendem File je ein C-File, welches den Befehl zum übertragen enthält, und je ein D-File, welches das zu übertragende File enthält. Das C-File wird von CCICO ausgeführt. Da Unix ein relativ 'sicheres' Betriebssystem ist, ist es meistens nicht möglich Files wirklich zu übertragen. Der Sender benötigt eine Schreiberlaubnis für das Directory, in das das File gesandt wird, die er in der Regel nicht hat. Beim nächsten Login erhält der User eine Mail mit dem Inhalt **'Request Denied'**. Es ist aber normalerweise möglich, in das Unterdirectory **'/usr/spool/uucppublic'** Files zu senden. Ebenso wird es kaum möglich sein, Files zu **'requesten'**, da das Login ja nicht mit dem 'normalen' Usernamen erfolgte, sondern mit dem Pointnamen.

SCCICO.COM

Dieses Programm ist der Gegenpart von CCICO. Es wird dann aufgerufen, wenn WAIT einen Anrufer (ein anderes System) erkannt hat und den Zugang zu dem Hostsystem erlaubt. Mangels einem zweiten Rechner habe ich das Programm bisher nicht ausprobieren können.

WAIT.COM

WAIT funktioniert beim mir nicht! Da es auch nur für SCCICO nötig ist, ist dies zu verschmerzen.

Soweit meine Erfahrungen mit der CP/M Version von UUCP. Abschliessend muß gesagt werden, daß es sich hierbei wirklich um ein großartiges Programmpaket handelt. Die übertragung mit CCICO zum Internet hin funktioniert ausgezeichnet.

Dieser Bericht ist kein Königsweg zum Installieren von UUCP 21b. Insbesondere meine Einschätzungen über die Bedeutung der Einträge 'user', 'inethost' und 'smarthost' sind wahrscheinlch falsch, mindestens korrekturbedürftig. Die Internetkundigen unter den Lesern werden gebeten, ihre Erkenntnisse hierzu beizusteuern.

Brodersdorf, 27.7.1993
Volker Dose

ERGÄNZUNGEN

Zum Thema WAIT.COM sei noch hinzugefügt, daß dieses Programm auf meinem Rechner (Genie IIIs mit 9.2 MHz und Z180 CPU) sofort abbricht. Der Timer-OVL, der auch für SMAIL usw. nötig ist, wird bei WAIT.COM gar nicht durchlaufen. Eventuell enthält WAIT.COM Illegals oder ist nur für CP/M 2.2 geeignet.

Ich verwende in diesem Artikel den Begriff 'Internet-Adresse'. Dies ist nicht ganz korrekt. Der String 'v_p_dose§cpmgut.toppoint.de' ist zunächst einmal mein vollständiger Username. Außerdem ist dies meine 'Usenet'-Adresse. Eine Internet-Adresse haben nur Rechner mit TCP/IP, ein bestimmtes Übertragungsprotokoll für Unix-Rechner. Solche Adressen sehen folgendermaßen aus : '193.174.3.<irgendwas>'.

Zu dem Problem mit dem richtigen Absender hier eine Antwort von Cornelius Kock (vicki§vicki.toppoint.de) aus Kiel:

Zitat Anfang

Die Adressierung im Use/Internet ist i.d.R. von der Form

user§kiste.subdomain.domain.topleveldomain

user§kiste ist klar. Domains sind Verwaltungsbereich eines Netzes, eine Subdomain ein Teilbereich einer Domain, topleveldomain ist ein Oberbereich. Oberbereiche sind z.B.

- de -- Deutschland
- ch -- Schweiz
- uk -- England
- com -- Commercial: Kommerzielles aus den USA
- mil -- Military : US-Army und aehnliches
- edu -- Educational : Amerikanische Unis.

Beispiel: Der Rechnername

methan.chemie.fu-berlin.de zerfaellt in

```
                      ^ de -- Deutschland
               ^ Freien Uni Berlin in
        ^ Fachbereichs Chemie der
^ der Rechner methan des
```

Domains sind also hierarchisch geordnet. Es gibt in .de eine Domain .toppoint, in welcher der Rechner cpmgut zu finden ist. Dein Rechnername im Usenet ist also cpmgut.toppoint.de, Dein (vollstaendiger) Username v_p_dose§cpmgut.toppoint.de. Die Domain .UUCP ist fuer diejenigen Rechner gedacht, die noch keiner Domain angehoeren. Dummerweise sind sie damit mailmaessig nur noch schwer erreichbar, weil Mail in eine Domain ueber bestimmte Knotenrechner weitergeleitet wird. Der Knoten fuer .toppoint.de ist die tpki. In anderen Worten: Irgendwo muss bei Dir noch eine Domainangabe rumliegen, welche dafuer sorgt, dass cpmgut sich nicht wie cpmgut.toppoint.de fuehlt.

Zitat Ende.

Ich habe jetzt eine Lösung gefunden, damit PNEWS und SMAIL die richtige Adresse in den 'From:'-String eintragen. Dazu mußte ich beide Programme patchen. In PNEWS findet sich der ein Codesegment, welches für den Eintrag des String '§cpmgut.UUCP' verantwortlich ist.

```
        ORG     01A96H
        DEFB    'From: %s§%s.UUCP',0Ah,00h
```

C-Kundige werden hierin einen String erkennen, der dem in einer 'printf'-Anweisung sehr ähnlich ist. Ich habe ihn geändert in:

```
        DEFB    'From: %s',0Ah,00h
```

Somit wird in News der richtige Absender eingetragen.

In SMAIL kann man ähnlich vorgehen. In dem 'From: '-Eintrag wird folgendes eingetragen, soweit man keinen 'From:'-Eintrag im Header definiert:

From: <hostname>!<user-ID>

bei mir also 'From: cpmgut!v_p_dose§cpmgut.toppoint.de (Volker Dose)'

Da die Anweisung im Programm nicht so einfach zu ändern war wie in PNEWS (hier stehen zwei '%s'), habe ich SMAIL um einen Record erweitert und in diesen dann den 'From:'-Header so eingetragen, wie er mir richtig erscheint, also ohne den vorgestellten <hostname>.

```
        ORG     2580H
        DEFB    'From: v_p_dose§cpmgut.toppoint.de (Volker Dose)',0Ah,00h
```

Außerdem müssen zwei Zeiger auf den neuen String gerichtet werden.

```
        ORG     0B2CH
        LD      HL,02580H       ; vorher 0227FH

        ORG     022BFH
        DEFW    2580H           ; vorher 0227FH
```

Diese Änderungen mögen brutal erscheinen, resultieren aber wahrscheinlich daraus, daß David Goodenough dieses Programm für wirkliche UUCP-Points erstellt hat. Ich will jedoch nur die Nachrichten und Mails von meiner UUCP-Box zuhause bearbeiten können (siehe Cornelius' Erläuterungen zu Domains und Adressen).

GEnie is proud to bring you the Internet RoundTable, with a staff of experts, hundreds of informational new files, and many new features:

-* Support for the Internet Mail Gateway
-* Mailing Lists & Newsgroup Digests
-* "Anonymous FTP" File Request Service
-* And much much more

The Internet RoundTable is a place for people who have an interest in the Internet, including electronic mail, usenet newsgroups, mailing lists & digests, anonymous FTP files, international wide area networking, and both beginning and advanced Internet uses. Introductory tutorials about the Internet can be found, as well as technical discussions of setting up Internetworks. Real-time Conferences are held on a regular basis, the schedule can be found in the Internet RT as one enters.

GEnie INTERNET-RT Page 1405
Internet RoundTable

1. Internet Bulletin Board
2. Internet Conference Room
3. Internet Library of Files
4. About the Internet RoundTable
5. RoundTable News (930504)
6. Feedback to the Internet Support Sysops
7. Internet Mail
8. Download Sysop's Treat File
9. Unix RoundTable
10. Search the Internet for a file
11. Request a file from the Internet

Enter #, <P>revious, or <H>elp?4

Welcome to the Internet RoundTable, a most excellent place on GEnie!
Like nearly all GEnie RoundTables the INTERNET-RT has several components. We'll show the menu and explain each item:

GEnie INTERNET-RT Page 1405
Internet RoundTable

1. Internet Bulletin Board
2. Internet Conference Room
3. Internet Library of Files
4. About the Internet RoundTable
5. RoundTable News (930408)
6.[*]Feedback to the Internet RT
7.[*]Internet Mail
8.[*]Download Sysop's Treat File
9. Unix RoundTable
10. Search the Internet for a file
11. Request a file from the Internet

===

1. Internet Bulletin Board

The Internet Bulletin Board is the primary discussion area. By selecting item

number 1 (or using the "A" "C" or "N" options in the Aladdin products) you will be placed in the Bulletin Board (BB). Within the BB you will find topics and information about the various kinds of Internet services, as well as support for GEnie's Internet Gateway, and a feedback area for Suggestions & Wish Lists.

2. Internet Conference Room

The Internet Conference Room is the place where Tutorials and other Real-time interactive events are held. Three times weekly, and at other times when announced as you MOVE to the Internet-RT, you will find the Sysops and fellow enthusiasts just chatting about the Internet. CHAT LINES on GEnie page 400 is also a great place to visit the INTERNET HELP DESKS on channel 4 -- these are held on Sunday and Thursday nights.

3. Internet Library KFiles

The Internet Library of Files contains hundreds of informational files and software relating to the Internet, such as the Internet Requests-for-Comments and every digested newsgroup. You'll also find programs that let you connect your computer to the Internet, or communicate with Internet machines. You'll want to check here frequently for new files (the "D" option in Aladdin), currently being added at about 20-30 files per day.

4. About the Internet RoundTable

About the Internet RoundTable is used to describe what the Internet RoundTable is all about.

5. RoundTable News (930408)

The RoundTable News item is used to give news of interest to all Internet RT participants. It will be changed on a frequent basis.

6.[*]Feedback to the Internet RT

This GEnie*Basic option allows you to send mail easily to the Internet RT to ask a question or give us your feedback. You can also send us mail at the special address, INTERNET$. The "$" is important.

7.[*]Internet Mail

By choosing item 7 you will be brought to GEnie's Internet Mail gateway on page 207. This is the place to sign up to send and receive Internet mail. The costs associated are minimal. There is no necessity to sign up for the gateway in order to receive the services here in the Internet RT.

8.[*]Download Sysop's Treat File

About twice a month a new file is made available by the Sysop for downloading as a FREE item. Selecting item 8 will present you with a description of the file, its size, and then offer you choices to Download or List.

9. Unix RoundTable

Many times persons wanting information about the Internet will also be interested in the Unix RoundTable on page 160. Selecting item 9 will bring you right to the Unix RT, or you can type "MOVE 160" or "UNIX".

10. Search the Internet for a file

11. Request a file from the Internet

The Sysops will search out a file for you on the Internet. In the best of cases the file is already here on GEnie in another RoundTable. If not, we'll get it and upload it for you. There are no extra charges beyond the standard GEnie Value prices for this service.

===

If you have any questions, please do not hesitate to ask us. We are here to serve with a friendly smile and a helping hand.

Welcome to the Internet RoundTable, a most excellent place on GEnie!

Your sysops are:

Andrew Finkenstadt	ANDY	Chief Sysop
Michael Nolan	MIKE.NOLAN	Assistant Sysop
Sarah Collier	SARAH	Administrative Assistant
Gary Smith	GARS	Library Manager
Janet McNeely	JANS	Bulletin Board Manager

Conference Schedule:
Tuesday May 4th, 1993: 10:00pm to 11:30pm Eastern Time
--> INTERNET Tutorial

Tuesday June 8th, 1993 6:00pm to Midnight Eastern Time
--> OPEN HOUSE! <--

HELP DESKS in GEnie CHAT LINES:
Thursdays at 10:00pm Eastern Time, Channel 4
Sundays at 8:00pm Eastern Time, Channel 4

GEnie INTERNET-RT Page 1405
Internet RoundTable

1. Internet Bulletin Board
2. Internet Conference Room
3. Internet Library of Files
4. About the Internet RoundTable
5. RoundTable News (930504)
6. Feedback to the Internet Support Sysops
7. Internet Mail
8. Download Sysop's Treat File
9. Unix RoundTable
10. Search the Internet for a file
11. Request a file from the Internet

Enter #, <P>revious, or <H>elp?

GEnie
Internet Software Library
Library: ALL Libraries

1. Description of this Library
2. Directory of Files
3. Search File Directory

```
 4.  Browse through Files
 5.  Upload a New File
 6.  Download a File
 7.  Delete a File You Own
 8.  Set Software Library
 9.  Save Current Software Library
10.  Instructions for Software Exchange
11.  Directory of New Files
12.  Join/Ignore Library Category

Enter # or <P>revious?

ALL Libraries.

No. File Name          Type Address  YYMMDD Bytes  Access Lib
---- ----------------- - ----------- ------ ------ ------ ---

645 RFC1373            X GARS        930504  19968   1 19
   Desc: rfc1373, Portable DUA's
644 RFC1372            X GARS        930504  11136   1 19
   Desc: rfc1372, Telnet Remote Flow Cnt.
643 RFC1371            X GARS        930504  18176   1 19
   Desc: rfc1371, Choosing Common IGP
642 RFC1370            X GARS        930504   4352   2 19
   Desc: rfc1370, OSPF Applicability
641 RFC1369            X GARS        930504  14080   1 19
   Desc: rfc1369, Ethernet Implementation
640 RFC1368            X GARS        930504  83968   1 19
   Desc: rfc1368, Managed Obj IEEE 802.3
639 RFC1367            X GARS        930504   4864   1 19
   Desc: rfc1367, Addr Management Sched.
638 RFC1366            X GARS        930504  17920   1 19
   Desc: rfc1366, Guide - Manage IP addr
637 RFC1365            X GARS        930504  12800   1 19
   Desc: rfc1365, IP addr ext proposal
636 RFC1364            X GARS        930504  32128   1 19
   Desc: rfc1364, BGP OSPF Interaction
635 PAS01040.Z         X GARS        930504  16792   1 25
   Desc: Pascal. Digest Vol 01, Issue 040
634 C++01040.Z         X GARS        930504  37000   1 25
   Desc: C++ Lang. Digest Vol 01, Issue 040
632 COH01069.Z         X GARS        930504   7800   1 25
   Desc: COHERENT Digest Vol 01, Issue 069
631 C-01069.Z          X GARS        930504  90764   1 25
   Desc: C Language Digest Vol 01, Issue 069
630 BSD01069.Z         X GARS        930504  28592   1 25
   Desc: 386BSD Digest Vol 01, Issue 069
629 UNIX-BBS.593       X GARS        930504  33384   1 4
   Desc: BBS Software for UNIX-like OS 5/93
628 PROLOG.593         X GARS        930504  13072   2 4
   Desc: Prolog Frequently Asked Questions
627 CHOOSENEWS15.CSH   X GARS        930504   8208   3 5
   Desc: ChooseNews II: Choose newsgroups
626 C++01039.Z         X GARS        930503  41216   1 25
   Desc: C++ Lang. Digest Vol 01, Issue 039
625 PAS01039.Z         X GARS        930503  16784   1 25
   Desc: Pascal. Digest Vol 01, Issue 039
624 SECURE01.068       X SARAH       930503  18600   1 25
   Desc: Security Digest Vol 01 #068 5-03-93
623 SECURE01.067       X SARAH       930503  46692   1 25
   Desc: Security Digest Vol 01 #067 5-02-93
622 LNX01068.Z         X GARS        930503  52332   1 25
   Desc: Linux Digest Vol 01, Issue 068
621 SECURE01.066       X SARAH       930503  24492   1 25
   Desc: Security Digest Vol 01 #066 5-01-93
620 COH01068.Z         X GARS        930503   5128   1 25
   Desc: COHERENT Digest Vol 01, Issue 068
619 C-01068.Z          X GARS        930503  36620   1 25
   Desc: C Language Digest Vol 01, Issue 068
617 G++_FAQ.593        X GARS        930503  64484   1 4
   Desc: FAQ for g++ and libg++, 01 May 1993
616 SRCECHEK.PL        X GARS        930503   3584   1 23
   Desc: see if posting contains any source
615 PAS01038.Z         X GARS        930503  22528   2 25
   Desc: Pascal. Digest Vol 01, Issue 038
614 C++01038.Z         X GARS        930503  86656   1 25
   Desc: C++ Lang. Digest Vol 01, Issue 038
613 LNX01067.Z         X GARS        930503 126976   1 25
   Desc: Linux Digest Vol 01, Issue 067
612 COH01067.Z         X GARS        930503  20352   1 25
   Desc: COHERENT Digest Vol 01, Issue 067
611 C-01067.Z          X GARS        930503  70912   1 25
   Desc: C Language Digest Vol 01, Issue 067
610 BSD01067.Z         X GARS        930503  44288   1 25
   Desc: 386BSD Digest Vol 01, Issue 067
609 TREK-ACT.593       X GARS        930503 239616   1 4
   Desc: Star Trek Actors' Other Roles FAQ
608 SOFTENG.493        X GARS        930503  95104   1 4
   Desc: Software-eng periodic postings 4/93
607 NUDE_FAQ.493       X GARS        930503  36736   1 4
   Desc: Nude Beaches Etc. FAQ: April 1993
606 MUD_FAQ.493        X GARS        930503  65152   2 4
   Desc: FAQ April 1993, MUDs and MUDding
605 FAQ.493            X GARS        930503  28416   1 4
   Desc: FAQ on FAQs, April 1993
604 CRYONICS.493       X GARS        930503  66816   1 4
   Desc: Cryonics FAQ, April 1993
603 COMPILER.FAQ.493   X GARS        930502 186240   1 4
   Desc: Catalog of Compilers, etc. Apr 1993
602 INET_SERVICES.LIST.493 X GARS    930502  25088   6 15
   Desc: Internet Services List, Apr 1993
601 NEWSTATS.493       X GARS        930502  32256   3 10
   Desc: Top 25 News Submitters Apr 1993
600 NEWSFEED.TAR.Z     X GARS        930502 131840   1 23
   Desc: newsfeed.tar.Z, net admin news tool
599 RFC1363            X GARS        930502  50304   1 19
   Desc: rfc1363, Proposed Flow Spec.
598 RFC1362            X GARS        930502  30336   1 19
   Desc: rfc1362, Novell IPX over WAN
597 RFC1361            X GARS        930502  23936   1 19
   Desc: rfc1361, Simple Network Time Prot
596 RFC1360            X GARS        930502  71936   1 19
   Desc: rfc1360, IAB Protocol Standards
595 U.DEU.Z            X SARAH       930502  72192   1 9
   Desc: Germany UUCPMAP Files *compressed*
594 SAVEMAP.NAWK.Z     X SARAH       930502   2048   1 9
   Desc: savemap.nawk.Z, comp.mail map saver
593 RFC1359            X GARS        930502  53504   1 19
   Desc: rfc1359, Connecting to Internet
592 MAPSTATS.Z         X SARAH       930502   4864   1 9
   Desc: mapstats.Z, UUCP stats utility
591 JAPANMAP.ZIP       X SARAH       930502  56576   2 9
   Desc: UUCP Maps for Japan
590 RFC1358            X GARS        930502  11392   1 19
   Desc: rfc1358, IAB Charter
589 RFC1357            X GARS        930502  25088   1 19
   Desc: rfc1357, E-mail Bibliographies
588 RFC1356            X GARS        930502  32128   1 19
   Desc: rfc1356, Multiprot X.25 & ISDN
587 RFC1355            X GARS        930502   8960   1 19
   Desc: rfc1355, Privacy and Accuracy
586 RFC1354            X GARS        930502  24960   1 19
   Desc: rfc1354, IP Forwarding Table
585 RFC1353            X GARS        930502  59648   1 19
   Desc: rfc1353, Managed Obj SNMP Parties
584 RFC1352            X GARS        930502  95744   1 19
   Desc: rfc1352, SNMP Security Protocols
583 RFC1351            X GARS        930502  80768   1 19
   Desc: rfc1351, SNMP Admin Model
582 RFC1350            X GARS        930502  24704   1 19
   Desc: rfc1350, TFTP-2 Protocol
581 RFC1349            X GARS        930502  68992   1 19
   Desc: rfc1349, Type of Service Discussion
580 RFC1348            X GARS        930502   6912   1 19
   Desc: rfc1348, DNS NSAP RR's
579 RFC1347            X GARS        930502  26624   1 19
   Desc: rfc1347, TCP & UDP Bigger Addresses
578 RFC1346            X GARS        930502  13184   1 19
   Desc: rfc1346, Resource Allocation
577 RFC1345            X GARS        930502 249856   1 19
   Desc: rfc1345, Character Mnemonics/Sets
576 RFC1344            X GARS        930502  25984   1 19
   Desc: rfc1344, Implications of MIME
575 RFC1343            X GARS        930502  29312   1 19
   Desc: rfc1343, User Agent Mail Config
574 RFC1342            X GARS        930502  15872   1 19
   Desc: rfc1342, Non-ASCII Text
573 RFC1341            X GARS        930502 211200   1 19
   Desc: rfc1341, MIME Mail Extensions
:
:
:
```

usw.

Queue# Item From Length Sent Subject
1 3707222 ANDY 125 93/05/06 Welcome to the Internet RT!

Item forwarded by R.REASERJR1 to DATAQUE.1

Item forwarded by R.REASERJR1 to CRAIG.S.THOM

Item 3707222 93/05/06 08:58

From: ANDY Andy Finkenstadt, Unix RT SysOp

To: ANDY Andy Finkenstadt, Unix RT SysOp

Sub: Welcome to the Internet RT!

Welcome to the Internet RoundTable, a most excellent place on GEnie!

Since you recently visited the Internet RoundTable for the first time, we'd like to take this opportunity to welcome you as a new member and offer our services to you. If you have any interest about the Internet, the INTERNET-RT is the place to come.

Like nearly all GEnie RoundTables the INTERNET-RT has several components. We'll show the menu and explain each item:

GEnie INTERNET-RT Page 1405
Internet RoundTable

1. Internet Bulletin Board
2. Internet Conference Room
3. Internet Library of Files
4. About the Internet RoundTable
5. RoundTable News (930408)
6.[*]Feedback to the Internet RT
7.[*]Internet Mail
8.[*]Download Sysop's Treat File
9. Unix RoundTable
10. Search the Internet for a file
11. Request a file from the Internet

==

1. Internet Bulletin Board

The Internet Bulletin Board is the primary discussion area. By selecting item number 1 (or using the "A" "C" or "N" options in the Aladdin products) you will be placed in the Bulletin Board (BB). Within the BB you will find topics and information about the various kinds of Internet services, as well as support for GEnie's Internet Gateway, and a feedback area for Suggestions & Wish Lists.

Be sure to introduce yourself in the "Welcome & Introduce Yourself" topic, in category 2 topic 2. It's great to meet new people online!

2. Internet Conference Room

The Internet Conference Room is the place where Tutorials and other Real-time interactive events are held. Three times weekly, and at other times when announced as you MOVE to the Internet-RT, you will find the Sysops and fellow enthusiasts just chatting about the Internet. CHAT LINES on GEnie page 400 is also a great place to visit the INTERNET HELP DESKS on channel 4 -- these are held on Sunday and Thursday nights.

3. Internet Library of Files

The Internet Library of Files contains hundreds of informational files and software relating to the Internet, such as the Internet Requests-for-Comments and every digested newsgroup. You'll also find programs that let you connect your computer to the Internet, or communicate with Internet machines. You'll want to check here frequently for new files (the "D" option in Aladdin), currently being added at about 20-30 files per day.

4. About the Internet RoundTable

About the Internet RoundTable is used to describe what the Internet RoundTable is all about.

5. RoundTable News

The RoundTable News item is used to give news of interest to all Internet RT participants. It will be changed on a frequent basis. The number in parenthesis indicates the date it was last changed.

6.[*]Feedback to the Internet RT

This GEnie*Basic option allows you to send mail easily to the Internet RT to ask a question or give us your feedback. You can also send us mail at the special address, INTERNET$. The "$" is important.

7.[*]Internet Mail

By choosing item 7 you will be brought to GEnie's Internet Mail gateway on page 207. This is the place to sign up to send and receive Internet mail. The costs associated are minimal. There is no necessity to sign up for the gateway in order to receive the services here in the Internet RT.

8.[*]Download Sysop's Treat File

About twice a month a new file is made available by the Sysop for downloading as a FREE item. Selecting item 8 will present you with a description of the file, its size, and then offer you choices to Download or List. Because of limitations in the GEnie software, you must download the Sysop Treat file via this option and not directly from the software library. Aladdin users will want to "Go online." Type "MOVE 1405;8" to get to the Sysop Treat directly.

9. Unix RoundTable

Many times persons wanting information about the Internet will also be interested in the Unix RoundTable on page 160. Selecting item 9 will bring you right to the Unix RT, or you can type "MOVE 160" or "UNIX".

10. Search the Internet for a file
11. Request a file from the Internet

[*] = Kostenloser Menüpunkt

GE-Mail auf Internet Access

The Sysops will search out a file for you on the Internet. In the best of cases the file is already here on GEnie in another RoundTable. If not, we'll get it and upload it for you. There are no extra charges beyond the standard GEnie Value prices for this service. You do not need to be registered with the Internet Gateway on page 207 in order to use these services.

==

If you have any questions, please do not hesitate to ask us. We are here to serve with a friendly smile and a helping hand.

Best regards,

Andrew Finkenstadt, Sarah Collier, Michael Nolan, Janet McNeely, Gary Smith
Internet RT Sysops

PS - Every month we will distribute a short article to you via electronic mail. If you'd prefer not to receive such mail, just write me a note and I'll remove you from the mailing list.

=END=

Press <RETURN>?

[*] = Kostenloser Menüpunkt

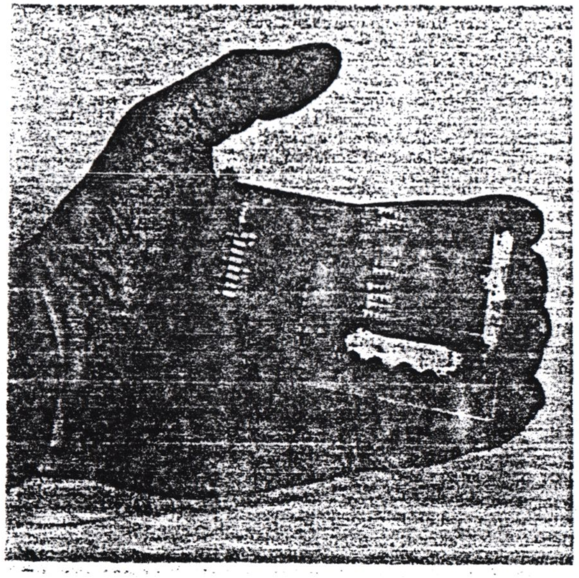
Floppy-Tester
Elektor Mai 1984

Floppy-Disk-Laufwerke haben sich schnell auch bei Heim-Computern als Massenspeicher durchgesetzt. Der im Vergleich zu einem Kassettenrekorder viel einfachere Zugriff auf gespeicherte Daten erfordert allerdings ein komplexes und sorgfältig konstruiertes Gerät. Es ist unvermeidlich, daß Fehler auftreten können, und dann sind es häufig auch noch recht merkwürdige. Sofort stellt sich die Frage: Liegt es am Laufwerk, oder irgendwo anders im System — sei es in der Hard- oder Software. Diese Schaltung ermöglicht ein schnelles und einfaches Überprüfen aller Laufwerksfunktionen.

Floppy-Tester

Prüfsimulator für Floppy-Disk-Laufwerke

Ein Floppy-Disk-Laufwerk dürfte eines der teuersten Stücke in der Hardware eines Heimcomputers sein: Nur der Computer selbst (die "Zentraleinheit") und der Drucker liegen in der gleichen Preisklasse. Und wenn einem das DOS-Betriebssystem immer wieder nur Fehlermeldungen ausgibt, ohne daß der Computer selbst bei der Fehlersuche helfen könnte, dann stellt sich die Frage: Was tun? Hat man zwei Laufwerke, tauscht man die beiden gegeneinander aus und kann so meist schon feststellen, ob der Fehler im Laufwerk liegt oder irgendwo anders im System. Damit ist man ein Stück weiter, aber noch nicht am Ende mit der Fehlersuche. Hat man dagegen nur ein Laufwerk, ist man meist schon "aufgeschmissen" bei der Frage: Liegt der Fehler im Laufwerk oder irgendwo sonst im System. In beiden Fällen hilft diese Schaltung weiter: Sie simuliert gegenüber dem Laufwerk alle Funktionen des Computers und zeigt gleichzeitig durch Leuchtdioden an, ob und in welchem Bereich Fehlfunktionen des Laufwerks vorliegen. Da die Funktionen des Simulators manuell gesteuert werden, kann man sehr schnell die gesamten elektronischen und mechanischen Funktionen des Laufwerks überprüfen.



Wer jetzt allerdings eine sehr aufwendige Schaltung erwartet, wird vom Schaltbild (angenehm) enttäuscht sein: In diesem Fall ist eine relativ einfache Schaltung die beste Lösung. Drei ICs und eine "Handvoll Drumherum" reichen aus. Mit den Schaltern S3 . . . S5 werden drei Drive-Select-Signale erzeugt. Unbedingt erforderlich ist davon nur eins, die anderen beiden erleichtern jedoch das Durchprüfen der Selektionslogik: Man kann einen entsprechenden Stecker fest mit der Schaltung verbinden und die normale Floppy-Kabel verwenden, ohne das oder die Floppys bereits beim Prüfen aus dem Gehäuse ausbauen zu müssen.

Nach dem Selektieren des Laufwerks wird mit S6 der (Antriebs-)Motor eingeschaltet. Wenn bei der eingelegten Diskette die Schreibschutzkerbe bedeckt ist, muß LED D3 aufleuchten. Dann legt man eine Diskette ohne Schreibschutz ein: D3 darf jetzt nicht aufleuchten.

Sofort nach Einschalten des Motors muß LED D1 aufleuchten. Nach Einlegen der Diskette beginnt D3 zu flackern: Das Indexloch wird signalisiert. Die Frequenz beträgt dabei 5 Hz (bei 5¼-Zoll-Laufwerken) oder 6 Hz (bei 8-Zoll-Laufwerken) bei soft-sektorierten Disketten und jeweils das Sechzehnfache bei hard-sektorierten Disketten. Wer dies genau nachmessen will, kann zwischen Pin 2 von N5 und Masse einen Frequenzzähler oder ein Oszilloskop anschließen. Ist diese Funktion nicht gegeben, untersucht man zuerst, ob sich irgendweiche "UFOs" (= Unbekannte Fremdobjekte) in den Photodetektor eingeschlichen haben.

Ist soweit alles in Ordnung, kommt der Stepper-Motor ins Spiel: Mit dem Umtaster S1, dem mit N1 und N2 aufgebauten Entprell-Flipflop und dem monostabilen Multivibrator MMV1 können einzelne Schritte des Stepper-Motors generiert werden. Durch S7 wird dabei bestimmt, ob der Kopf nach außen oder nach innen bewegt wird (S7 geschlossen = nach innen). D2 leuchtet auf, wenn Track Ø erreicht ist. Also geht man erst nach außen, bis D2 aufleuchtet, und dann die Gesamtzahl der Spuren nach innen. Häufig kann man dann hören, wenn etwas "schief läuft" (wenn das Gehör nicht gerade disco-geschädigt ist). Dann das Ganze nochmal nach außen bis Track Ø.

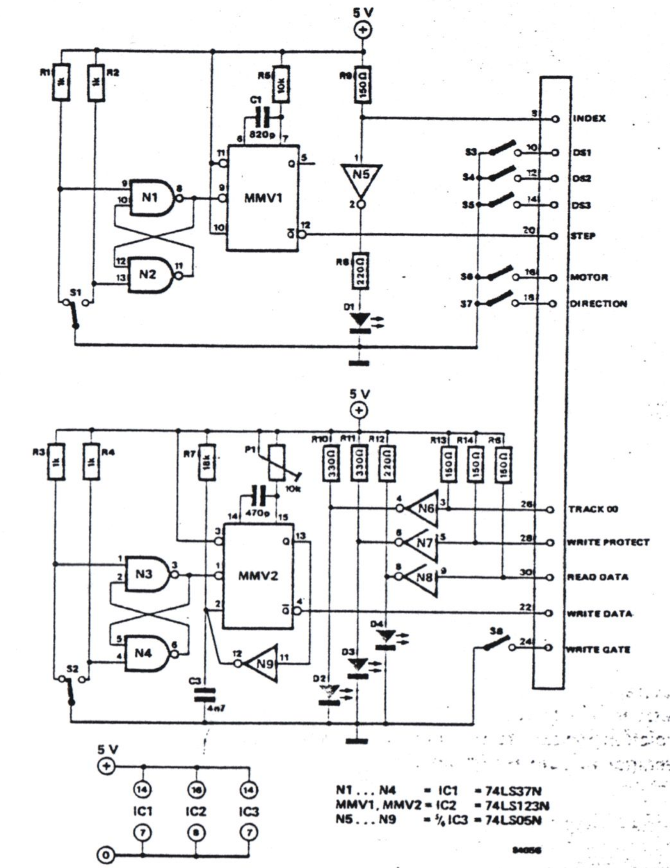
Nun geht es ums Schreiben und Lesen. Also steckt man zunächst eine leere Diskette (oder eine, die gelöscht werden kann) ins Laufwerk. S8 (Write Gate) wird geöffnet und dann S2 gedrückt. Solange man S2 festhält, werden "Daten" auf die Diskette gesetzt: MMV2 ist durch N9 zum Oszillator "umfunktioniert". Die Impulsfrequenz beträgt 125 kHz (= 8 µs), die Impulsweite liegt bei 500 ns. Mit P1 kann dieser Wert genau abgeglichen werden.

Nach dem "Laden" einer Spur wird S8 wieder geschlossen, und dadurch werden die "Daten" eingelesen. D4 muß jetzt leuchten (zumindest sporadisch, je nachdem, wie man S2 gedrückt hat). Zur genaueren Überprüfung sollte man beim Abspeichern S2 lange genug drücken und beim Lesen ein Oszilloskop zwischen Pin 8 von N8 und Masse anschließen.

Jetzt müßte der Fehler schon weitgehend eingekreist sein. Liegt er im Schreib-Lese-Teil, dann kann man mit MMV2 als Signalgenerator und dem Oszilloskop den Schreib- und danach den Leseteil der Floppy-Elektronik überprüfen.

Die im Schaltbild angegebene Belegung des Steckers gilt für 34-polige Standard- (Shugart-kompatible) Laufwerke. Bei 8-Zoll-Laufwerken (50-polige Steckverbinder) und manchen "Exoten" gilt eine andere Steckerbelegung! Bei der angegebenen Belegung und auch bei den "8-Zöllern" gilt: Alle ungeraden Stecker-Pins werden mit Masse verbunden (Twisted-Pair-GND). Nun können wir nur noch jedem einzelnen Leser wünschen, daß er sich diese Schaltung nie fädeln muß. Gut aufbewahren sollte man sie auf alle Fälle. ◼

1



Floppy-Tester
Elektor Mai 1984

Bild 1. Trotz geringen Aufwandes liefert die Testschaltung alle Signale, um ein Floppy-Laufwerk "auf Herz und Nieren" untersuchen zu können.

Club 80 Börse --

Uwe Schoberth, Petrus-Waldus-Str. 14, neu 75443 Oetisheim
Telefon 07041-7254

S u c h e

1. Original-Listing oder Assemblerdatei des CP/M-Bios für den Triumph Adler Rechner AlphaTronic P3.

2. Unterlagen zum Systemverschiebeprogramm MOVCPM, bzw. genauere Angaben über die Innereien des Programms.

Da ich des häufigeren auch abends nicht zu hause bin, ruft einfach ein paarmal an.

-- Club 80 Börse

Insider-Info

WINDOWS-Fehler

Gestern hat mich WINDOWS wieder einmal verblüfft. Plötzlich wollte es mit der Maus nicht mehr zusammenarbeiten, und das obwohl sie von DOS-Programmen einwandfrei erkannt wurde.

Dabei hatte ich nur mein internes Modem (ZyXEL 1496 B, Steckkarte) ausgebaut und dadurch war der Kommunikationsport COM 1 nicht mehr vorhanden. Da die Maus aber auf COM 2 läuft, dürfte das doch wohl nichts ausmachen!?

Weit gefehlt! Während sowohl der Maus-Treiber als auch alle anderen MSDOS-Programme sich von dem fehlenden COM 1-Port nicht irritieren ließen, verwürfelte WINDOWS sowohl die Portadressen als auch die Interrupts der Schnittstellen so, daß gar nichts mehr ging.

Erst nachdem ich die als COM 3 konfigurierte Schnittstelle auf COM 1 geändert hatte, klappte das "Mausen" wieder. WINDOWS verkraftet es einfach nicht, wenn COM 1 fehlt!

Meiner Meinung nach stimmt die Behauptung der MicroSoft-Verkäufer "WINDOWS ist absolut absturzsicher!" hundertprozentig - es stürzt laufend ab!

Hartmut Obermann

Neuigkeiten für CP/M

Langsam lassen sich immer mehr von Euch in der Discovery sehen und ich fände es eine gute Idee, wenn jeder nach Möglichkeit seine Artikel, bzw. eine Kurzfassung davon, in die Clubecke setzen würde. Dann können sich alle mit Modem die neuesten Artikel brühwarm abholen und alle anderen bekommen sie dann im Info nachgeliefert.

Hier eine gute Nachricht für alle Speicherplatzfetischisten und die, die sich mit einem 2400 Baud Modem durch die DFÜ quälen:

Es gibt einen neuen Packer für CP/M. Gerade habe ich einen neuen Schwung Public-Domain Software für ZCPR bekommen und da war auch dieses sehr interessante Teil dabei (läuft aber auch ohne ZCPR). Das Format ist dem DOS-LHA sehr ähnlich und man kann damit auch LZH-Files vom DOS auspacken (andersrum geht es allerdings nicht, aber das spielt wohl keine Rolle). Die Packungsdichte unter den CP/M-Packern ist damit momentan wohl am höchsten, das Beste an diesem Packer ist aber, daß man damit SELBSTAUSPACKENDE Archive erzeugen kann ! Dann bekommt man ein ganz normales COM-File und wenn man das auf dem Zielsystem aufruft, entpackt es sich ohne weiteres Zutun. Als PMAUTOAE.COM ist es in der Filearea in der DISCOVERY zu finden.

Helmut Jungkunz hat mir vor kurzem gesagt, daß er die Preise für ZCPR und Z3PLUS auf 70 Mark gesenkt hat. Bei diesem Preis dürfte sich die Fangemeinde wohl bald noch vergrößern denn einmal ist es eine ungemeine Arbeitserleichterung und im Preis inbegriffen ist eine kostenlose Versorgung mit Public Domain Software zum Z-System, die momentan über 40MB umfaßt. Nein, ich bekomme keine Prozente, aber ich arbeite selber mit dem Z3PLUS und ich würde mich mit Händen und Füßen wehren, wenn ich es wieder hergeben sollte.

Das nächste Highlight betrifft die Programmiersprache C. Der Hi-Tech C Compiler ist ab sofort Public Domain ! Keine Ente, ich habe die Files und die Freigabe hier. Hi-Tech C ist meines Wissens einer der ganz wenigen (der einzige ?) ANSI-C Compiler unter CP/M und wirklich erste Sahne. Sogar die Quelltexte der Bibliotheken sind dabei und ich habe schon Programme, die unter Turbo-C (DOS) erstellt wurden, problemlos compiliert. Gebt die frohe Botschaft doch bitte weiter, damit sich die Leute nicht weiter mit den anderen Compilern rumquälen müssen. Das komplette Handbuch liegt als File vor und ich kann jedem, der sich unter CP/M mit C beschäftigen will oder muß, nur zu diesem Compiler raten.

So gut wie alle Programme, über die ich hier so schreibe sind auf den PD-Floppies zum Z-System zu finden, aber wer das trotz allem nicht will, kann mir natürlich auch eine Floppy nebst Rückporto schicken.

Zum Schluß noch eine kurze Anmerkung zum Offline-Reader: Am Anfang hatte ich gedacht, daß das Protokoll zum Up- und zum Download identisch sein muß, aber das ist keineswegs so. Zum Upload hat sich ARC als praktisch erwiesen, da ich noch keine Mailbox erlebt habe, die das nicht automatisch erkennen und auspacken kann. Zum Download kann man jedes beliebige Format auswählen, für das man einen Entpacker hat (das stellt man dann in der Box fest ein)

Alexander Schmid

Neue Preisstrukuren

Egbert Schröer, Juli 93

Langsam hält die bei vielen Online Datenbankanbietern angebotene Preisstruktur einer monatlichen **subscriber fee** plus Online Nutzungsgebühr pro Zeiteinheit bei den kommerziellen Mailboxanbietern, bzw. "Zwittern" wie GEnie Einzug. Seit 1.Juli realisierte GEnie seine neue Preisstruktur. Betroffen davon sind bisher nur die USA und Kanada. Betroffen ? oder bevorzugt ? Was hat sich geändert ?

Bisher - und das gilt auch weiterhin unter anderem für Deutschland - zahlte man nur nach Online Zeit 18$ pro Stunde zwischen 18 und 8 Uhr bzw. an Wochenenden und 27$ während der Prime Time (Abb. 1)

Basic Connect Rate

Germany, Austria, Switzerland Rates

Baud	Prime $/Hour	Non-Prime $/Hour
up to 300	$27.00	$18.00
1200	$27.00	$18.00
2400	$27.00	$18.00

Abb.1

Die neue Preisstruktur - und da hebt sich GEnie erfreulich von anderen ab - bleibt einfach und übersichtlich wie bisher. Monatliche subscriber fee, Standard-Stundensatz und monatlichen Kredit an Conect Time machen die Sache auch für neue Nutzer übersichtlich (Abb.2).

GEnie Services [4]	U.S. (U.S.$)[5]	CANADA (CAN $)
Monthly Subscription Fee	$8.95/month	$10.95/month
Includes:	Up to 4 hours of standard connect time [1]	
Standard Hourly Connect Charge	$3.00/hour	$4.00/hour
GEnie Premium Services	Prices vary per individual service. These include: Charles Schwab Brokerage Services (not available in Canada), Dow Jones News/Retrieval (R), GE Mail to Fax, The Official Airline Guides Electronic Edition (R) Travel Service, QuikNews clipping service, Investment ANALYST(SM), and other database research services.	

Additional Charges (where applicable) [4]

Prime Time Surcharge [2]	$9.50/hour	$12.00/hour
9600 Baud Surcharge	$6.00/hour	$8.00/hour
Communications Surcharge:		
"800" Service [3]	$6.00/hour	--
Extended Network b	$2.00/hour	--
SprintNet	$2.00/hour	--
Datapac	--	$6.00/hour

[1] Your monthly subscription fee includes a credit during your first 4 hours of standard $3.00 an hour ($4.00 an hour CAN) connect time that month. Unused credit will not be transferred to the next month.

[2] Prime-time: 8 a.m. - 6 p.m. local time on weekdays only. The prime-time surcharge is in addition to the $3 an hour standard hourly connect charge. Prime-time surcharge is waived for selected holidays. Residents of Hawaii, Alaska, Indiana, Arizona and Puerto Rico, please verify hours with GEnie Client Services.

[3] "800" Service surcharge waived at 9600 baud.

```
[4] State/Local taxes and adminstrative charges may apply.

[5] International PDN subscribers billed at U.S. rates in U.S. Dollars.

    Prices subject to change without notice. Effective as of 7/1/93.
              Copyright (C) 1993, General Electric Company
```

Other notes:

1. There will no longer be additional charges or registration fees for Internet mail gateway use.
2. Club pricing options will be eliminated effective July 1.

Abb. 2

Was bedeutet das für die monatliche Rechnung ? Für die Mehrzahl der User wird sich wenig ändern, aber bei geschickter Planung kann man doch Geld sparen. Der Vorteil liegt auf der Hand: vor der Online Sitzung kommt die Planung, dann gezieltes Ansteuern der Roundtables oder Software Libraries spart bei 4 Stunden freier Connect Time echtes Geld. Der Pferdefuß ist auch offensichtlich - bei Nichtnutzung des Mailboxdienstes verfallen die 4 Stunden und die subscriber fee ist immer fällig. Das Offline Planen einer Mailbox Session wird durch eine überarbeitete ALADDIN Version erelichtert. ALADDIN ist ein komfortables Terminalprogramm, mit Offline Reader, LogOn Script Erstellung usw. (erhältlich als Download für IBM, Apple Mac und Atari). Beispielsweise dauert bei Voreinstellung des CP/M Roundtables mit der Vorgabe alle neuen Nachrichten und alle neue Files seit dem letzten LOGIN abzufragen die gesamte Connect Time bei 1200 baud knappe 30 Sekunden. Damit hat man die Nachrichten und Files noch nicht als Download, aber ALADDIN gibt jetzt die Möglichkeit die Nachrichten und Files anhand ihrer Themenzuordnung und Kurzbeschreibung (und/oder Autor) anzuwählen und anschließend mit dieser Vorauswahl automatisch erneut mit dem neuen Frageprofil einzusteigen und die Downloads - wiederum durch ALADDIN automatisiert - anzufordern. Diese Vorgehensweise sollte man auch als Deutschland Nutzer von GEnie jetzt schon üben. Bleibt abzuwarten, ob sich - vielleicht durch Druck der USER - dies Preispolitik für Deutschland ändert. Eins hat sich sichtbar für alle geändert: Die bisher mit <*> gekennzeichneten GEnie*Basic Services sind für den deutschen Nutzer nicht mehr sichtbar. Zur Erläuterung: in den Basic Services wird keine Connect Time berechnet (GEnie Mail, Billing Services,GEnie Lamp usw.). Da die Datex-P (oder sonstige Einwählknoten) Kosten von GEnie übernommen werden, kann man dort ohne seinen Geldbeutel arg zu strapazieren kostenlos Onine bleiben und von diesen "Plattformen" seine Mailbox-Exkursion starten. Neuen Nutzern aus Deutschland ist jetzt auf Anhieb nicht mehr ersichtlich ob nun die Connect Time fällig ist oder nicht. Wenn man allerdings ALADDIN mit der Option /CLOCK startet, wird die gebührenpflichtige Connect Time angezeigt.

Zugangsmöglichkeiten GEnie

Egbert Schröer, August 93

Als Normalnutzer haben die wenigsten einen DATEX-P Anschluß. Es gibt aber doch die Möglichkeit ohne postalischen DATEX-P Anschluß oder Einwählknoten mit NUI und NUA auf GEnie und andere kommerzielle Anbieter zuzugreifen.

Als erstes die wohl teuerste Variante: Zugang über die Wirtschaftsdatenbank Genios, dem deutschen Pendant zu GEnie mit Schwerpunkt Wirtschafts- und Presseinformationen, bietet ein Gateway zu GEnie und ist über Telefonleitung erreichbar.

Weiter Möglichkeiten:

1. Netze

SprintNet Alaska
SprintNet
DATAPAC Canada
C & C VAN
DATEX-J (Durchschalten zu DATEX-P möglich)

2. Mailboxen
einige Mailboxen bieten Gateways zu DATEX-P, z.B. die Box C.A.T.

TITLE: Communicating.
AUTHOR: Pournelle, Jerry
JOURNAL: Byte VOL.: v11 ISSUE: n2 PAGINATION: p291-292
PUBLICATION DATE: Feb., 1986
ARTICLE TYPE: column
ABSTRACT: The Little Board from Ampro Computers Inc., Mountain View, California, is good microcomputer which runs ZCPR, a public-domain operating system with better performance than CP-M 2.2. The price of the Little Board begins at $289. The MPI Printer from Micro Peripherals Inc., Salt Lake City, Utah, is a portable, fast, smart printer which performs an above average task for drafts. The MPI works with Zenith PCcompatibles. The Laser BDS 630-8 from BDS Corporation, Mountain View, California, is a printer which performs better than Hewlett-Packard Co.'s LaserJet. Other software and hardware are included.
OPERATING SYSTEM(S): ZCPR
DESCRIPTORS: Compilers; Evaluation; Hardware; Microcomputers; Performance; Printers; Software

TITLE: CP-M 2.2 Goes Public.
AUTHOR: Mitchell, B.; McKay, D.
JOURNAL: Dr. Dobbs Journal VOL.: v9 ISSUE: n11 PAGINATION: p48-54
PUBLICATION DATE: Nov., 1984
SPECIAL FEATURES: illustration; program
SOURCE FILE: CD File 275
ABSTRACT: A search path feature is provided in ZCPR, the Z80 replacement for the CP-M 2.2 Console Command Processor (CCP). Plu*Perfect Systems' public file approach enables the program to be used from more than one user number. A Z80 BDOS patch is the key to the public file approach and is listed. The PUBLIC supporting utility is also listed. Over ninety percent of the control over user numbers that are needed on a CP-M system are provided by public files combined with the ZCPR-type search path.
CAPTIONS: File Utility Program-Assembly Language (program)
OPERATING SYSTEM(S): CP-M
OPERATING PLATFORM(S): CP-M
ISSN: 1044-789X
DESCRIPTORS: CP-M; Files; File Activity; Opening-Closing Files; File Management; Tutorial; Programs; Disk-Tape File Utilities; Assembly Language

TITLE: The Public Domain of CP-M Software.
AUTHOR: Thompson, C.J.
JOURNAL: Softalk VOL.: v4 ISSUE: n3 PAGINATION: p104-109
PUBLICATION DATE: Nov., 1983
ARTICLE TYPE: evaluation
ABSTRACT: Apple computer users who own a CP-M card can find some useful utility programs at a minimal cost among the many programs in the public domain. Public domain programs are available through CP-M user groups and clubs. The utility programs described are: CAT. COM, CLEANUP. COM, COMPARE. COM, DISPLAY. COM, DU. COM, DUPUSR. COM, FINDBAD. COM, FMAP. COM, GOTO. ASM, GOTO. COM, SAP. COM, SD. COM, SD. DOC, TED. COM, TURNKEY. COM, UCAT. COM, UNERA. COM, ZCPR. COM, ZCPR. DOC, and ZCPR. HLP
OPERATING SYSTEM(S): CP-M
PRODUCT NAME(S): CAT. Com; CLEANUP. COM; COMPARE. COM; DISPLAY. COM; DU. COM; DUPUSR. COM; FINDBAD. COM; FMAP. COM; GOTO. ASM; GOTO. COM; SAP. COM; SD. DOC; TED. COM; UCAT. COM; UNERA. COM; ZCPR. COM; ZCPR. DOC; ZCRR. HLP
DESCRIPTORS: CP-M; Utility Programs; Public Software; Systems Software

ZCPR-3: The Manual - CP/ M-80 Super Consol Command

DATE ENTERED: 930315 STATUS IN FILE: Changed
Conn Richard
New York Zoetrope,U.S. Dec 85: Gazelle Bk.Services
24cm.viii,351. 15ill.6tabs.
BINDING: Paperback PRICE: L11.50 ISBN: 0918432596
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics
Computers (Software)

Literraturrecherche USA , England und Deutschland
USA und GB: 1985 bis 1993

1. USA und England

10/9/1 (Item 1 from file: 470)
01561682 2394976XX STATUS: Active entry
TITLE: Using CP-M Plus on the Amstrad PCW: Step-by-Step
AUTHOR: Campbell, John
PUBLISHER: Buttrwrth-Heinemann PUBLICATION DATE: 03/1992 (920301)
NO. OF PAGES: 256p.
LCCN: N/A
BINDING: pap. - $34.95
ISBN: 0-7506-0460-3
VOLUME(S): N/A
ORDER NO.: N/A
IMPRINT: N/A
STATUS IN FILE: New (92-04)
NOTE(S): Illustrated
SUBFILE: PB (Paperbound Books in Print); ST (Scientific and Technical Books and Serials in Print)

LIBRARY OF CONGRESS SUBJECT HEADINGS: CP-M PLUS (COMPUTER OPERATING
SYSTEM) (01011935)
PAPERBOUND BOOK SUBJECT HEADINGS: TECHNOLOGY- COMPUTERS AND COMPUTER
TECHNOLOGY (0000456X)

10/9/2 (Item 1 from file: 430)
02042705
CP/ M Plus on the Amstrad PCW
DATE ENTERED: 930408 STATUS IN FILE: Changed
Campbell John
SERIES: Step by Step S.
Butterworth-Heinemann Jan 92
22cm.256. Ill.
BINDING: Paperback PRICE: L14.95 ISBN: 0750604603
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics
Computers (Software)

10/9/3 (Item 2 from file: 430)
01970256
ZCPR-3: The Manual - CP/ M-80 Super Consol Command
DATE ENTERED: 930315 STATUS IN FILE: Changed
Conn Richard
New York Zoetrope,U.S. Dec 85: Gazelle Bk.Services
24cm.viii,351. 15ill.6tabs.
BINDING: Paperback PRICE: L11.50 ISBN: 0918432596
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics
Computers (Software)

10/9/4 (Item 3 from file: 430)
01456749
Personal Computer Care Manual: Diagnosing and Maintaining Your M. S.

-DOS, CP/ M or Macintosh System
DATE ENTERED: 920817 STATUS IN FILE: Changed
Morrison Chris
Stover Teresa S.
TAB Bks.,U.S. Jul 88: Wiley
M6.220. Ill.
BINDING: Paperback PRICE: L15.95 ISBN: 0830629912
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

10/9/5 (Item 4 from file: 430)
00639770
Small Mac User's Manual Release 1.2 for CP/ M
DATE ENTERED: 900718 STATUS IN FILE: New
Hendrix James E.
M & T Publishing,U.S. Apr 88: Computer Bkshops.
23cm.71.
BINDING: Paperback PRICE: L29.95 ISBN: 0934375054
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics
Computers (Software)

15/2/1 (Item 1 from file: 470)
01143505 1711143XX STATUS: Active entry
TITLE: PC Care Manual: Diagnosing & Maintaining Your MS-DOS CP-M or Macintosh System
AUTHOR: Morrison, Chris; Stover, Teresa S.
PUBLISHER: TAB Bks PUBLICATION DATE: 11/1987 (871101)
NO. OF PAGES: 224p.
LCCN: 87-026235; N/A
BINDING: Trade - $24.95; pap. - $16.95 o.p.
ISBN: 0-8306-0991-1; 0-8306-2991-2
VOLUME(S): N/A; N/A
ORDER NO.: 2991; N/A
IMPRINT: N/A; N/A
STATUS IN FILE: New (88-02)
NOTE(S): Illustrated
SUBFILE: PB (Paperbound Books in Print); ST (Scientific and Technical Books and Serials in Print)

LIBRARY OF CONGRESS SUBJECT HEADINGS: MACINTOSH (COMPUTER)
(01008067)
PAPERBOUND BOOK SUBJECT HEADINGS: TECHNOLOGY- COMPUTERS AND COMPUTER
TECHNOLOGY (0000456X)

15/2/2 (Item 2 from file: 470)
01097292 1621320XX STATUS: Active entry
TITLE: CP-M 3.0 Handbook for the C128
TRANSLATED TITLE: Trans. Title: CP-M 3.0 Anwender Hanbuch C128
AUTHOR: Huckstadt, Jurgen; Dorn, Susan-Editor; Prazak, Ludwig J.-Translator
PUBLISHER: Prog Peripherals PUBLICATION DATE: 05/1987 (870501)
EDITION: Orig. Ed. NO. OF PAGES: 250p.
LCCN: N/A
BINDING: pap. - $19.95
ISBN: 0-941689-00-X
VOLUME(S): N/A

ORDER NO.: N/A
IMPRINT: N/A
STATUS IN FILE: New (87-04)
SUBFILE: PB (Paperbound Books in Print); ST (Scientific and Technical Books and Serials in Print)

LIBRARY OF CONGRESS SUBJECT HEADINGS: C (COMPUTER PROGRAM LANGUAGE) (
01006617)
PAPERBOUND BOOK SUBJECT HEADINGS: TECHNOLOGY- COMPUTERS AND COMPUTER
TECHNOLOGY (0000456X)

15/2/3 (Item 3 from file: 470)
01015740 1227589XX STATUS: Active entry
TITLE: System Programmer's Manual for Montezuma Micro CP-M 2.2 (release 2.2x) for the TRS-80 Model 4
AUTHOR: Zuma, Monte; Overholt, Jesse B.
PUBLISHER: Montezuma Micro PUBLICATION DATE: 09/1985 (850901)
EDITION: Orig. Ed. NO. OF PAGES: 64p.
LCCN: N/A
BINDING: pap. text ed. - $39.00
ISBN: 0-928295-01-X
VOLUME(S): N/A
ORDER NO.: N/A
IMPRINT: N/A
STATUS IN FILE: New (85-09)
SUBFILE: PB (Paperbound Books in Print); ST (Scientific and Technical Books and Serials in Print)

PAPERBOUND BOOK SUBJECT HEADINGS: TECHNOLOGY- COMPUTERS AND COMPUTER
TECHNOLOGY (0000456X)

15/2/4 (Item 4 from file: 470)
01010723 1207972XX STATUS: Active entry
TITLE: Scientific & Engineering Applications with Personal Computers: A Software Approach with Examples for the Apple, IBM-PC & CP-M Based Microcomputer Systems
AUTHOR: Annino, Raymond; Driver, Richard
PUBLISHER: Wiley PUBLICATION DATE: 04/1986 (860401)
NO. OF PAGES: 577p.
LCCN: 85-020275
BINDING: Text ed. - $69.95
ISBN: 0-471-79978-5
VOLUME(S): N/A
ORDER NO.: N/A
IMPRINT: N/A
STATUS IN FILE: New (85-07)
SUBFILE: ST (Scientific and Technical Books and Serials in Print)

LIBRARY OF CONGRESS SUBJECT HEADINGS: SCIENCE-DATA PROCESSING (00533567);
ENGINEERING-DATA PROCESSING (00147102)

15/2/5 (Item 1 from file: 430)
02022100
Scientific and Engineering Applications with Personal Computers; A Software Approach with Examples for the Apple, I. B. M. P. C. and CP/ M Based Microprocessor Systems

DATE ENTERED: 930408 STATUS IN FILE: Changed
Annino Raymond
Driver Richard D.
Wiley Jun 86
M8.600. Ill.
BINDING: Cloth PRICE: L68.00 ISBN: 0471799785
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/6 (Item 2 from file: 430)
02014147
CP/ M Plus Handbook; Operator's and Programmer's Guide for the Amstrad CPC 6128 and PCW 8256
DATE ENTERED: 930408 STATUS IN FILE: Changed
Digital Research
Heinemann Jul 87
D8.514.
BINDING: Paperback PRICE: L20.00 ISBN: 0434903213
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/7 (Item 3 from file: 430)
02014146
Digital Research CP/ M Plus Manual
DATE ENTERED: 930408 STATUS IN FILE: Changed
Heinemann Jul 86
C4.500. Ill.
BINDING: Paperback PRICE: L25.00 ISBN: 0434903205
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/8 (Item 4 from file: 430)
01707625
CP/ M 80 System Programming
DATE ENTERED: 921207 STATUS IN FILE: Changed
Brookes Graham R. etc.
Blackwell Sci. Sep 85
M8.192. 95ill.
BINDING: Paperback PRICE: L10.95 ISBN: 0632012978
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics

15/2/9 (Item 5 from file: 430)
01606506
Using CP/ M
DATE ENTERED: 921116 STATUS IN FILE: Changed
Gosling P.E.
Macmillan Educ. Jan 85
M8.128.
BINDING: Paperback PRICE: L9.99 ISBN: 0333384032
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics

15/2/10 (Item 6 from file: 430)
01413587

Osborne/ McGraw-Hill CP/ M 86 Users' Guide
DATE ENTERED: 920817 STATUS IN FILE: Changed
Sachs Jonathan
A.Osborne,U.S. Mar 85: McGraw
M8.350. Ill.
BINDING: Paperback PRICE: L19.95 ISBN: 0078811430
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/11 (Item 7 from file: 430)
01407193
Datapro/ McGraw-Hill Guide to CP/ M Software
DATE ENTERED: 920817 STATUS IN FILE: Changed
McGraw Jan 85
D4.386. Ill.
BINDING: Paperback PRICE: L23.95 ISBN: 0070154082
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/12 (Item 8 from file: 430)
01271953
Introducing Amstrad CP/ M Assembly Language
DATE ENTERED: 920615 STATUS IN FILE: Changed
Sinclair Ian R.
Collins Apr 86
DISTRIBUTOR: BSP Professional
M8.
BINDING: Paperback PRICE: L9.95 ISBN: 0003833097
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics

15/2/13 (Item 9 from file: 430)
01267644
Disc Operating System Plus Reference Guide for Personal Computer DOS, M. S. -DOS and CP/ M Programmers
DATE ENTERED: 920515 STATUS IN FILE: Changed
Digital Research
Glentop P. Mar 87
D8.333.
BINDING: Spiral PRICE: L9.95 ISBN: 1851811478
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/14 (Item 10 from file: 430)
01174500
CP/ M 80 Programmer's Guide
DATE ENTERED: 920409 STATUS IN FILE: Changed
Morrell Barry
Whittle Peter
Macmillan Educ. Nov 85
M8.176. 48ill. 68d.
BINDING: Paperback PRICE: L10.95 ISBN: 0333395581
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics

15/2/15 (Item 11 from file: 430)
00890407
CP/ M 86 Assembly Language
DATE ENTERED: 910819 STATUS IN FILE: Changed
Lindsay Jon
Charles P.,U.S. Apr 85
DISTRIBUTOR: Prentice-Hall
M8.288.
BINDING: Paperback PRICE: L19.10 ISBN: 0893033901
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Mathematics

15/2/16 (Item 12 from file: 430)
00674561
Introduction to CP/ M Plus on Amstrad Computers
DATE ENTERED: 901015 STATUS IN FILE: Changed
McBride P.K.
Glentop Publrs. Jul 86
M8.176. 16ill.
BINDING: Paperback PRICE: L7.95 ISBN: 1851810544
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/17 (Item 13 from file: 430)
00674559
Choosing and Using CP/ M Business Software on the Amstrad PCW 8256/ 8512
DATE ENTERED: 901015 STATUS IN FILE: Changed
McBride P.K.
Glentop Publrs. Mar 87
D8.200.
BINDING: Paperback PRICE: L6.50 ISBN: 1851810528
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/18 (Item 14 from file: 430)
00539472
Using Amstrad CP/ M Business Software
DATE ENTERED: 891113 STATUS IN FILE: Changed
Sinclair Ian R.
Collins Aug 86
DISTRIBUTOR: BSP Professional
M8.160.
BINDING: Paperback PRICE: L8.95 ISBN: 0003833593
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: Commerce

15/2/19 (Item 15 from file: 430)
00521807
Amstrad CP/ M Plus
DATE ENTERED: 890814 STATUS IN FILE: Changed
Clarke Andrew R.M.
Lybbe David Powys-
M.M.L.Systems Jun 86: R.J.B.,54 Stanhope Rd.,Littlehampton BN17 6AQ
IC6.540. 2r.e.
BINDING: Cloth PRICE: L12.95 ISBN: 1869910052
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/20 (Item 16 from file: 430)
00439652
Introduction to CP/ M Assembly Language
DATE ENTERED: 881017 STATUS IN FILE: Changed
Lindsay Jon
Hayden Bk.Co.,U.S. Dec 86: Pitman
M8.176.
BINDING: Cloth PRICE: L15.95 ISBN: 0810452103
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/21 (Item 17 from file: 430)
00348510
Introduction to CP/ M
DATE ENTERED: 870715 STATUS IN FILE: New
Penfold R.A.
Bernard Babani Oct 86
D8.96. 10ill.
BINDING: Paperback PRICE: L2.95 ISBN: 0859341577
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/22 (Item 18 from file: 430)
00346531
Amstrad CP/ M Plus
DATE ENTERED: 870616 STATUS IN FILE: Changed
Clarke Andrew R.M.
Lybbe David Powys-
MML Systems Apr 86: RJB,54 Stanhope Rd, Littlehampton BN17 6AQ
M6.470.
BINDING: Cloth PRICE: L14.95 in ring-binder ISBN: 1869910001
PRINT STATUS: Out of Print

COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/23 (Item 19 from file: 430)
00343614
Introducing CP/ M on the Z-80, on the BBC Micro Second Processor
DATE ENTERED: 870616 STATUS IN FILE: Changed
Smith Bruce
Collins Jul 85
DISTRIBUTOR: BSP Professional
M8.160.
BINDING: Paperboards PRICE: L9.95 ISBN: 0003831086
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/24 (Item 20 from file: 430)
00336662
CP/ M Solutions
DATE ENTERED: 870518 STATUS IN FILE: Changed
Barbier Ken
Prentice-Hall Oct 85
M6.176.
BINDING: Paperback PRICE: L17.35 ISBN: 0131881787
PRINT STATUS: Out of Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/25 (Item 21 from file: 430)
00318062
Amstrad CP/ M Plus
DATE ENTERED: 861021 STATUS IN FILE: New
Clarke Andrew R.M.
Lybbe David Powys-
M.M.L.Systems Apr 86: R.J.B.,54 Stanhope Rd.,Littlehampton BN17 6AQ
M6.470.
BINDING: Cloth PRICE: L14.95 in ring-binder ISBN: 1869910001
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

15/2/26 (Item 22 from file: 430)
00244761
Osborne/ McGraw-Hill CP/ M-86 Users' Guide
DATE ENTERED: 861021 STATUS IN FILE: New
Sachs Jonathan
A.Osborne,U.S. Mar 85: McGraw
M8.350. Ill.
BINDING: Paperback PRICE: L23.75 ISBN: 0881341436
PRINT STATUS: In Print
COUNTRY OF PUBLICATION: England
SUBJECT HEADINGS: General

2. Deutschland

2.1. Literatur zum Z80 Prozessor

L4 ANSWER 1 OF 22 COPYRIGHT 1993
AN (93)911484329 BIBLIODATA

TI Mikrocomputer-Datensammlung: 8085/Z80,8086-80486, 8087-80387, XT/AT, PS/2, MS-DOS, DEBUG, MASM, CP/M, DDT, ASM, PASCAL, C.
AU Hannemann, Dieter; Niemietz, Arno
AS D. Hannemann ; A. Niemietz
PB Gelsenkirchen-Buer : Hannemann, 1992.
SO 1. Aufl. - 157 S. ; 21 cm. - ISBN 3-920088-30-1 kart. : DM 22.85.
CY Deutschland, Bundesrepublik
CC *28 Informatik, Kybernetik
37 Elektrotechnik
CT Datensammlung; Mikrocomputer
CTC 30
DN 93,A17,1132; 91,N49,0280

L4 ANSWER 2 OF 22 COPYRIGHT 1993
AN (89)891635467 BIBLIODATA

MAIN:
TI Mikroprozessor-Datenbuch: [Daten in englischer Sprache].
AU Hogenboom, Paul
AS zsgest. von Paul Hogenboom
PB Aachen : Elektor-Verl.
SO 22 cm.
PART:
TI 2. Die Periferie zur 180er Serie, Z80-Familie, 6800er Serie, 6502-Familie, 8086er Serie und Verwandte und andere
PB 1989.
SO 414 S. : zahlr. graph. Darst. - ISBN 3-921608-76-7 kart. : DM 39.80.
DN 89,A49,0699

L4 ANSWER 3 OF 22 COPYRIGHT 1993
AN (89)890463387 BIBLIODATA

MAIN:
TI Das Emuf-Sonderheft: Einplatinen-Mikrocomputer f*uer universelle Festprogramm-Anwendung.
PB M*uenchen : Franzis
SO 30 cm. - Literaturangaben.
PART:
TI 2. 6504-, 6502-, Z80-, 68008-, 8086- und Basic-Emuf, Telefon-Emuf, universelle Fernbedienung, Basic-Emuf mit LCD und Tasten, Emuf als Bordcomputer, Z80-Emuf mit Display und Tastatur, Multitasking mit dem Z80-Emuf
AU Hofer, Rudolf [Hrsg.]
AS [f*uer d. Text verantw.: Rudolf Hofer]
PB 1987.
SO 1. Aufl. - 98 S. : Ill., graph. Darst. - geh. : DM 19.50.
TS (Sonderheft / Franzis-Verlag GmbH ; Nr. 247)
DN 89,A13,0610

L4 ANSWER 4 OF 22 COPYRIGHT 1993
AN (89)890463360 BIBLIODATA

MAIN:
TI Mikro-Computer Schritt f*uer Schritt: das Sonderheft zum NDR-Klein-Computer aus der Fernsehserie "Mikroelektronik" des NDR.
AU Rohde, Ulrich [Hrsg.]
AS [f*uer d. Text verantw.: Ulrich Rohde]
PB M*uenchen : Franzis
SO 30 cm.

PART:
TI 2. Der Vollausbau des NDR-Computers: Z80 und CP/M: von der Hardware zum Betriebssystem ; 68008 und CP/M: vom Maschinenbefehl bis zum Floppy-Betrieb alles verst*aendlich.
PB 1985.
SO 129 S. : Ill., graph. Darst. - geh. : DM 28.00.
TS (Sonderheft / Franzis-Verlag GmbH ; Nr. 204)
DN 89,A13,0626

L4 ANSWER 5 OF 22 COPYRIGHT 1993
AN (89)890344841 BIBLIODATA
TI Lehrbuch der Mikroprozessortechnik: eine allgemeine Einf*uehrung in Hardware, Software und Anwendung am Beispiel des Z80-Systems.
AU Stewen, Ludolf
AS Ludolf Stewen
PB Heidelberg : H*uethig, 1989.
SO X, 326 S. : graph. Darst. ; 24 cm. - ISBN 3-7785-1323-0 kart. : DM 88.00.
DN 89,A40,0566; 89,N08,0269

L4 ANSWER 6 OF 22 COPYRIGHT 1993
AN (89)890124701 BIBLIODATA

MAIN:
TI Programmieren in Maschinensprache mit Z80
E. Schmitter.
AU Schmitter, Ernst-D.
PB Holzkirchen : Hofacker
SO 21 cm.
PART:
TI Bd. 2. Geeignet f*uer TRS-80, GENIE, SINCLAIR, SHARP
PB 1983.
SO 1. Aufl. - 100 S. : graph. Darst. - ISBN 3-88963-024-3 kart. : DM 29.80.
DN 89,A05,0780

L4 ANSWER 7 OF 22 COPYRIGHT 1993
AN (88)880052678 BIBLIODATA
TI Eine PASCAL-Maschine f*uer das Doppelprozessorsystem MC68000/Z80
AU Heubel, Thomas; Malowaniec, Krzysztof T.; Scherer, Edgar
AS T. Heubel ; K. Malowaniec ; E. Scherer. Rechenzentrum Univ. d. Saarlandes
PB Saarbr*uecken : RZ-UNI-SB, 1983.
SO 57, 10 Bl. : graph. Darst. ; 30 cm. - geh. (Pr. nicht mitget.).
TS (Bericht / Rechenzentrum der Universit*aet des Saarlandes Saarbr*uecken ; Nr. RZ 83,02)
DN 88,B05,0328

L4 ANSWER 8 OF 22 COPYRIGHT 1993
AN (88)871454203 BIBLIODATA
TI [Z-achtzig-Anwendungen]
Z80 Anwendungen
AU Coffron, James W.
AS James W. Coffron. [Dt. *Uebers.: Winfried Wolf]
PB Berkeley ; Paris ; D*uesseldorf : SYBEX, 1987.
SO 4. Aufl. - 290 S. : graph. Darst. ; 22 cm. - ISBN 3-88745-037-X Pp. : DM 48.00. - Engl. Ausg. u.d.T.: Coffron, James W.: Z eighty applications.
DN 88,A01,0494

L4 ANSWER 9 OF 22 COPYRIGHT 1993
AN (87)870302256 BIBLIODATA

MAIN:
TI [Z-achtzig-Programmierpraxis]
Z80-Programmierpraxis
PB Konstanz : Technisches Lehrinstitut Christiani
SO 30 cm.
PART:
TI Stichw*oerter
PB [1986 ?].
SO 7 S. - geh. (Pr. nicht mitget.).
DN 87,A11,0753

L4 ANSWER 10 OF 22 COPYRIGHT 1993
AN (87)870302248 BIBLIODATA

MAIN:
TI [Z-achtzig-Programmierpraxis]
Z80-Programmierpraxis
PB Konstanz : Technisches Lehrinstitut Christiani
SO 30 cm.
PART:
TI [Registerbl*aetter]
PB 1986.
SO [9] Bl. - (Pr. nicht mitget.).
DN 87,A11,0753

L4 ANSWER 11 OF 22 COPYRIGHT 1993
AN (87)860380920 BIBLIODATA
TI Z80-Maschinenprogramme [Z-achtzig-Maschinenprogramme] mit Sharp MZ 700 und MZ 800: e. Einf. mit vielen Beispielen.
AU Ostermann, Helmut
AS Helmut Ostermann
PB W*uerzburg : Vogel, 1986.
SO 2. Aufl. - 242 S. : graph. Darst., Noten ; 21 cm. - ISBN 3-8023-0830-1 kart. : DM 30.00.
TS (HC - mein Home-Computer)
DN 87,A25,0519; 86,N16,0178

L4 ANSWER 12 OF 22 COPYRIGHT 1993
AN (86)861122275 BIBLIODATA

MAIN:
TI [Z-achtzig-Programmierpraxis]
Z80-Programmierpraxis
PB Konstanz : Technisches Lehrinstitut Christiani
SO 30 cm.
PART:
TI 4
PB (1986).
SO 81 S. : graph. Darst. - kart. (Pr. nicht mitget.).
DN 86,A48,0555

L4 ANSWER 13 OF 22 COPYRIGHT 1993
AN (86)861122267 BIBLIODATA

MAIN:
TI [Z-achtzig-Programmierpraxis]
Z80-Programmierpraxis
PB Konstanz : Technisches Lehrinstitut Christiani
SO 30 cm.
PART:
TI 3
PB (1986).
SO 86 S. : graph. Darst. - kart. (Pr. nicht mitget.).
DN 86,A48,0555

L4 ANSWER 14 OF 22 COPYRIGHT 1993
AN (86)860841847 BIBLIODATA

MAIN:
TI [Z-achtzig-Programmierpraxis]
Z80-Programmierpraxis
PB Konstanz : Technisches Lehrinstitut Christiani
SO 30 cm.
PART:
TI 2.
PB 1986.
SO 1. A. - 76 S., S. 15 - 20 : graph. Darst. - kart. (Pr. nicht mitget.).
DN 86,A38,0689

L4 ANSWER 15 OF 22 COPYRIGHT 1993
AN (86)860841839 BIBLIODATA

MAIN:
TI [Z-achtzig-Programmierpraxis]
Z80-Programmierpraxis
PB Konstanz : Technisches Lehrinstitut Christiani
SO 30 cm.
PART:
TI 1.
PB 1985.
SO 1. A. - 56, 14 S. : Ill., graph. Darst. - kart. (Pr. nicht mitget.).
DN 86,A38,0689

L4 ANSWER 16 OF 22 COPYRIGHT 1993
AN (86)860583392 BIBLIODATA
TI Universelle Heizungssteuerung: d. Z80-EMUF steuert Ihre Heizung.
AU Lang, Egon

AS Egon Lang
PB M*uenchen : Franzis-Verlag, 1986.
SO 102 S. : zahlr. Ill. u. graph. Darst. ; 23 cm. - ISBN 3-7723-8211-8 kart. : DM 36.00.
TS (Franzis-Elektronik-Praxis)
DN 86,A38,0857; 86,N25,0202

L4 ANSWER 17 OF 22 COPYRIGHT 1993
AN (86)860169162 BIBLIODATA
TI Ein Z80-Basic-Interpreter [Z-achtzig-Basic-Interpreter] f*uer Mess-
und Steueraufgaben
AU Kiefer, Dieter
AS [Dieter Kiefer]
PB M*uenchen : IRT, 1979.
SO 21 S. ; 30 cm. - kart. (Pr. nicht mitget.).
TS (Technischer Bericht / Institut f*uer Rundfunktechnik ; Nr. [B]16)
DN 86,B09,0224

L4 ANSWER 18 OF 22 COPYRIGHT 1993
AN (85)850138558 BIBLIODATA
TI Z80-Maschinenprogramme [Z-achtzig-Maschinenprogramme] mit dem Sharp
MZ 700 und MZ 800: e. Einf. mit vielen Beispielen.
AU Ostermann, Helmut
AS Helmut Ostermann
PB W*uerzburg : Vogel, 1985.
SO 1. Aufl. - 242 S. : graph. Darst. ; 21 cm. - ISBN 3-8023-0830-1 kart.: DM 30.00.
TS (HC - mein Home-Computer)
DN 85,A25,0336; 85,N05,0182

L4 ANSWER 19 OF 22 COPYRIGHT 1993
AN (85)850102944 BIBLIODATA
TI Z eighty applications <dt.>
[Z-achtzig-Anwendungen]
Z80 Anwendungen
AU Coffron, James W.
AS James W. Coffron. [Dt. *Uebers.: Winfried Wolf]
PB Berkeley ; Paris ; D*uesseldorf : Sybex, 1984.
SO 1. Aufl. - 290 S. ; 21 cm. - ISBN 3-88745-037-X kart. : DM 48.00. -
Aus d. Engl. *uebers.
DN 85,A05,0571

L4 ANSWER 20 OF 22 COPYRIGHT 1993
AN (85)841024138 BIBLIODATA

MAIN:
TI Programmierung von Mikroprozessoren
AU Hannemann, Dieter
AS Dieter Hannemann
PB Essen : Girardet
SO 24 cm. - Literaturangaben.
PART:
TI 1. Die 8-bit-Mikroprozessoren: 8080, 8085, Z80, C800.
PB 1984.
SO 1. Aufl. - 48 S. : graph. Darst. - ISBN 3-7736-8302-2 kart. : DM 10.80.
DN 85,A10,0479; 84,N44,0134

L4 ANSWER 21 OF 22 COPYRIGHT 1993
AN (84)831021632 BIBLIODATA
TI Mikroprozessoren 6502, 6800, 8080, Z80, 9900: Grundlagen -
Programmierung - Vergleiche - *Uebungen.
AU Schumny, Harald

AS Harald Schumny
PB Braunschweig ; Wiesbaden : Vieweg, 1983.
SO VII, 240 S. : graph. Darst. ; 24 cm. - ISBN 3-528-04235-4 kart. : DM 48.00.
DN 84,A11,0466; 83,N40,0218

L4 ANSWER 22 OF 22 COPYRIGHT 1993
AN (84)831010207 BIBLIODATA
TI Assembler-Programmierung von Mikroprozessoren (8080, 8085, Z80) mit
dem ZX 81: mit 100 Programmbeispielen.
AU Kahlig, Peter
AS Peter Kahlig. Mit e. Geleitw. von Curt C. Christian
PB Braunschweig ; Wiesbaden : Vieweg, 1983.
SO VIII, 185 S. ; 23 cm. - ISBN 3-528-04238-9 kart. : DM 38.00.
TS (Programmieren von Mikrocomputern ; Bd. 8)
DN 84,A14,0488; 83,N40,0212

2.2. Literatur Wordstar 4.0

L7 ANSWER 1 OF 3 COPYRIGHT 1993
AN (89)881228508 BIBLIODATA
TI WordStar 4.0 kurz und b*uendig: mit MailMerge.
AU Langfelder, Christian
AS Christian Langfelder
PB W*uerzburg : Vogel, 1988.
SO 1. Aufl. - 184 S. : graph. Darst. ; 21 cm. - ISBN 3-8023-0219-2 Pp. : DM 35.00.
TS (Chip-Wissen)
DN 89,A01,0733; 88,N40,0216

L7 ANSWER 2 OF 3 COPYRIGHT 1993
AN (88)880878495 BIBLIODATA

MAIN:
TI IBM PC und Kompatible, Einf*uehrung in WordStar
AU Naiman, Arthur; Hofmann, Barbara M. S. [Bearb.]
AS Arthur Naiman. [*Ueberarb.: B. M. S. Hofmann]
PB D*uesseldorf ; San Francisco ; Paris ; London : SYBEX
SO 22 cm. - ISBN 3-88745-648-3.
PART:
TI [Hauptbd.]. Version 3.4
AS [Dt. *Uebers.: Bernd Pol]
PB 1987.
SO 1. Aufl. - 282 S. : Ill. - Pp. : DM 42.00, sfr 38.60, S 328.00. -
Orig.-Ausg. u.d.T.: Naiman, Arthur: Introduction to WordStar.
DN 88,A32,0521

L7 ANSWER 3 OF 3 COPYRIGHT 1993
AN (88)880878487 BIBLIODATA

MAIN:
TI IBM PC und Kompatible, Einf*uehrung in WordStar
AU Naiman, Arthur; Hofmann, Barbara M. S. [Bearb.]
AS Arthur Naiman. [*Ueberarb.: B. M. S. Hofmann]
PB D*uesseldorf ; San Francisco ; Paris ; London : SYBEX
SO 22 cm. - ISBN 3-88745-648-3.
PART:
TI Beil. Anhang f*uer WordStar 4.0
PB 1988.
SO 1. Aufl. - 109 S. - kart. : DM 42.00, sfr 38.60, S 328.00.
DN 88,A32,0521

2.3. ANSI - C

L8 ANSWER 1 OF 5 COPYRIGHT 1993
AN (93)921518153 BIBLIODATA
TI C f*uer Pascalprogrammierer
AU Rupp, Christine
AS Christine Rupp
PB Heidelberg : H*uethig, 1992.
SO 315 S. ; 24 cm. - ISBN 3-7785-2146-2 kart. : DM 49.00.
DN 93,A05,0704; 92,N47,0217

L8 ANSWER 2 OF 5 COPYRIGHT 1993
AN (93)920472745 BIBLIODATA
TI The C primer <dt.>
Das McGraw-Hill-C-Buch: [ANSI C und C++].
AU Hancock, Les; Krieger, Morris; Zamir, Saba
AS Les Hancock ; Morris Krieger ; Saba Zamir. [*Uebers.: Peter Dobrowolski. Hrsg. von Jay Renade]
PB London ; Hamburg [u.a.] : McGraw-Hill, 1993.
SO XVIII, 444 S. : graph. Darst. ; 24 cm. - ISBN 3-89028-393-4 Pp. : DM 75.00.
TS (McGraw-Hill computing)
DN 93,A16,0939; 92,N25,0260

L8 ANSWER 3 OF 5 COPYRIGHT 1993
AN (92)920658040 BIBLIODATA
TI Programmieren mit C: Ansi Standard.
AU Lowes, Martin; Paulik, Augustin
AS von Martin Lowes und Augustin Paulik
PB Stuttgart : Teubner, 1992.
SO 2., durchges. Aufl. - 272 S. : graph. Darst. ; 23 cm. - ISBN 3-519-12286-3 kart. : DM 32.00.
DN 92,A35,0777; 92,N23,0349

L8 ANSWER 4 OF 5 COPYRIGHT 1993
AN (91)901274712 BIBLIODATA
TI Programmieren mit C: ANSI Standard.
AU Lowes, Martin; Paulik, Augustin
AS von Martin Lowes und Augustin Paulik
PB Stuttgart : Teubner, 1990.
SO 272 S. : graph. Darst. ; 23 cm. - ISBN 3-519-02286-9 kart. : DM 32.00.
DN 91,A04,0688; 90,N39,0434

L8 ANSWER 5 OF 5 COPYRIGHT 1993
AN (90)891449981 BIBLIODATA
TI ANSI C: Programmieren portabler C-Programme im k*uenftigen Sprachstandard "ANSI C".
AU Herold, Helmut
AS Helmut Herold
PB M*uenchen : tewi, 1989.
SO XV, 424 S. : Ill. ; 22 cm. - ISBN 3-89362-040-0 Kunststoff : DM 79.00, sfr 72.70, S 616.00.
DN 90,A26,0519; 89,N40,0290

Quellen: Deutschland - STN File Bibliodata
USA/England - DIALOG File 470 und 430 (Books in Print)

Wichtige Tips zur Handhabung und Pflege Ihres

E L E K T R I K E R S

1.) Heranwachsende Elektriker dürfen nicht an Dosen hantieren , bei welchen mit Schlägen jeglicher Art zu rechnen ist.

2.) Ausgewachsene Elektriker sind 3x täglich , mindestens jedoch nach jedem Genuß von Kaffee, Tee, Suppe oder Kakao kräftig von innen mit Alkohol zu reinigen. (Merke : Nicht am falschen Ende sparen !)

3.) Streicheln Sie Ihren Elektriker niemals gegen den "Strich".

4.) Bei Mangelerscheinungen wie z. B. ruhigen Händen , ist sofort ein erfahrener Gastronom zu Rate zu ziehen.

5.) Sollten Sie einen schlafenden Elektriker finden , sollten Sie ihn nicht wecken , erschrecken oder gar einstecken.
Treffen Sie vielmehr dafür Sorge daß :
- er in seinem wohlverdienten Schlaf nicht gestört wird.
- er warm und weich zu liegen kommt.
(Lagertemperatur +18C - +22C)
- er nach dem Aufwachen eine warme Mahlzeit vorfindet.

6.) Im übrigen ist es ratsam :
- ihn nicht zu hetzen.
- ihn nicht zu reizen.
- ihn nicht von der Seite oder gar barsch anzureden.
- ihm immer zeigen , wie symphatisch er Ihnen ist.
- und natürlich ab und zu eine nicht zu kleine Aufmerksamkeit.

7.) Verwenden Sie nur ORIGINAL-ERSATZ-ELEKTRIKER.
Best.-Nr.: 0815-026 (Elektriker , Standard)
4711-026 (Elektriker , Bürobereich)
0000-026 (Elektriker , Vorbeiarbeiter)
2001-026 (Elektriker , High Tech)
007-026 (Elektriker , Lizenz zum Löten)

Impressum

1. Vorsitzender:	**Hartmut Obermann** Mozartring 23 Postfach 14 30	Tel.: 0 82 21/ 3 02 48 BTX: 0 82 21/ 3 02 48 FAX: 0 82 21/ 3 35 75
89304	**Günzburg**	
2. Vorsitzender:	**Gerald Schröder** Am Schützenplatz 14	Tel.: 0 41 05/ 26 02
21218	**Seevetal**	
Hardwarekoordinator:	**Andreas Magnus** Bismarckstraße 29	Tel.: 02 09/ 87 02 30
45879	**Gelsenkirchen**	
NewDOS-Diskothekar:	**Oliver Volz** Am Ochsenwald 37A	Tel.: 07 11/ 74 40 51
70565	**Stuttgart** (Rohrerhöhe)	
CP/M-Diskothekar:	**Fritz Chwolka** Saarstraße 34	Tel.: 0 24 64/ 89 20
52457	**Aldenhoven**	
C-128-Diskothekar:	**Günter W. Braun** Postfach 80 02 26	
81602	**München**	
Clubbücherei:	**Kurt Müller** Sophie-Scholl-Ring 3b	Tel.: 0 41 52/ 7 06 43
21502	**Geesthacht**	
Redaktion:	**Jens Neueder** Gschlachtenbretzingen Rudolf-Then-Straße 32	Tel.: 07 91/ 4 28 77 BTX: 07 91/ 44 47 22 FAX: 0 79 71/ 2 50 55
74544	**Michelbach / Bilz**	
Bankverbindung:	**Club 80** Postgiro Sonderkonto CLUB 80 Obermann H., 8870 Günzburg	**Postgiroamt Frankfurt** **BLZ: 500 100 60** **Kto.Nr.: 496 071 - 605**

Autoren: Die Redaktion bedankt sich bei den im **Inhaltsverzeichnis** genannten Autoren für die Mitarbeit an der Club-INFO. Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.

Schluß

Hallo Club 80'er,

wie Ihr gesehen habt, ist das aktuelle Info wieder recht umfangreich geworden. Die Auswertung Eurer Meinungen und das Aktualisieren Eurer Daten beanspruchte ganz schön viel Zeit, dafür sind wir aber auch mal wieder "UP TO DATE".

Eure Informationen aus dem Fragebogen sind ausgewertet, wir werden versuchen, soweit möglich, Euren Vorschlägen nachzukommen. Viele der gewünschten Aktivitäten lassen sich aber erst durch Eure Mithilfe realisieren. Ich hoffe, daß wir da an einem Strang ziehen.

Diesem Info liegt wieder ein Sonderheft "**Wissenswertes rund um den Z280**" bei. Ein "**Microcomputer-Software-Guide**" (37 Seiten) kann auf Wunsch bei der Redaktion angefordert werden, er wird dann mit dem nächsten Info mitgeliefert.

Auch in diesem Jahr möchten wir uns wieder auf der Hobbyelektronik in Stuttgart (07.11.93) treffen. Als Treffpunkt dient, wie bisher jedes Jahr, der Eingangsbereich nach der Kartenkontrolle am unteren Eingang. Hier hängt an der Wand das Clubemblem. Bisher war dort auch immer "Conrad electronic" gegenüber, man trifft sich gleich morgens zum Einlaßbeginn oder mittags, 12Uhr.

So, daß war es auch schon wieder für dieses INFO, denkt bitte an den Redaktionsschluß zum 1. Dezember.

Man sieht sich auf der Hobbyelektronic oder
bis zum nächsten mal Euer

Jens

lfd. Nr.	Nachname	Vorname	Straße		PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
1	Barendt	Harry	Hermann-Löns-Straße 7	GER	50181	Breedburg (Erft)	02272/7168	-	-	-
							-	-	-	-
2	Berndt-Jochum	Ilse	Stachelsgut 24	GER	51427	Bergisch Gladbach	02204/ 65254	-	-	-
	IBM, GENIE III, GENIE IIIs, SHARP Pocket 1600■ Scanner						02204/ 65254	-	-	-
3	Bernhardt	Helmut	Hafenstraße 7	GER	24262	Heikendorf	0431/ 241907	0431/ 245717	-	-
	IBM, Prof 180, CPU 280■ diverse PCs,Novellite-Netz,1496E,Mustek105+						0431/ 77578-20	-	2:242/262.26	-
4	Betz	Heinrich	St. Wolfgangstraße 13	GER	91353	Hausen	09191/ 31698	-	-	-
	IBM, Tandy M4p■						09191/ 611108	-	-	-
5	Brans	Jörg	Tieloh 55	GER	22307	Hamburg	040/ 6906531	-	-	-
	IBM■ CD-Rom, Streamer, Soundkarte						-	-	-	-
6	Braun	Günter W.	Postfach 80 02 26	GER	81602	München	-	-	-	-
	Commodore 128 D■ Akustikkoppler						-	-	-	-
7	Braun	Harald	Postfach 8011	GER	24154	Kiel	0431/35139	-	-	-
							-	-	-	-
8	Böckling	Ulrich	Juchaczstraße 61	GER	56203	Höhr-Grenzhausen	02624/ 4861	-	-	-
	IBM 386DX33,1040ST,VC20,C64,TRS80 M I,ZX81 ■ Modem, Videodat-, Videotextdecoder						02631/ 895168	-	-	-
9	Chwolka	Fritz	Saarstraße 34	GER	52457	Aldenhoven	02464/ 8920	-	-	-
	IBM 386+Co, Commodore, Apple, Z 280■ Modem 2.4						-	-	2:248/242:8	-
10	Dose	Volker	Dorfstraße 10	GER	24235	Brodersdorf	04343/ 1357	-	-	-
	GENIE IIIs mit Z180■ EPROMer, Modem						-	-	-	-
11	Halgasch	Gert	Großschönauer Straße 26	GER	02796	Jonsdorf	035844/ 636	-	-	-
	IBM 386DX■						-	-	-	-
12	Hartmann	Hans-Günther	Möwenstraße 9	GER	27804	Berne	04406/ 6911	04406/ 1071	-	-
	IBM 386SX, TANDY M4p, Z280-Kartenrechner■ Real Time Clock, Speed Up Kit 6,3MHz, 40MB-P1						0421/ 248-2419	-	2:240/300.24	-
13	Held	Manfred	Stirner Straße 22	GER	91785	Pleinfeld	09144/ 6563	09144/ 8514	-	-
	IBM■ Modem ZyXEL 1496+, CD-ROM						0911/ 219-2245	-	2:2400/10.10	-
14	Hermann	Klaus	Forchenstraße 8	GER	72124	Pliezhausen	07127/ 71945	-	-	Discovery
	IBM 386■ ET 4000, Modem						-	-	2:2407/70.740	07127/70107
15	Hürdler	Manfred	Niederhoferstraße 29	GER	97222	Rimpar	09365/ 4235	-	-	-
	Victor Sirius 1 (IBM), CPC 6128■						-	-	-	-
16	Johnen	Willi	Hansemannstraße 1	GER	52351	Düren	02421/ 501305	-	-	-
	GENIE IIIs■						02421/ 33064	-	-	-
17	Kauka	Dietmar	Straße des Friedens 37	GER	04552	Neukirchen (Borna/Leipz	03433/851019	-	-	-
							-	-	-	-
18	Kemmer	Jürgen	Dorfberg 7	GER	97232	Sulzdorf	09334/ 1050	-	-	-
	IBM 386, SB Pro 1.0, ZyXEL U1496E+■ DCF-77-Empfänger parallel, IR-Sender seriell						-	-	2:246/86.1	-
19	Kuhn	Eckehard	Im Dorf 14	GER	72636	Frickenhausen	07022/ 45417	-	-	-
	Atari ST 1040, TRS80 M I■						-	-	-	-
20	Linder	Jörg	Küstriner Str. 68	GER	15306	Seelow	03346/ 520	-	-	-
	KC 85/4 mit Floppy■						-	-	-	-
21	Lorenz	Walter	Mahräckerstraße 9	GER	60431	Frankfurt /Main	069/ 531656	-	-	-
	IBM 286/486, Z80-, HD 64180-Eigebau, Z280-T.R■ Scanner, Soundkarte, ET 4000						-	-	-	-
22	Magnus	Andreas	Bismarckstraße 29	GER	45879	Gelsenkirchen	0209/ 144029	-	-	-
	IBM 386Dx, GENIE IIIs■ Modem 2400						-	-	-	-
23	Mahlert	Herbert	Hohenbudbergerstraße 112 A	GER	47229	Duisburg	02065/ 47217	-	-	-
	IBM, GENIE I■ c't Videotext-Karte, Vobis Videodat-Decoder						02065/ 902592	-	-	-
24	Müller	Kurt	Sophie-Scholl-Ring 3B	GER	21502	Geesthacht	04152/ 70643	-	-	-
	IBM, Atari■ ATONCE 386, Pixelwounder, Scanner						040/ 89983403	-	-	-
25	Mössel	Franz	Schafferstraße 12	I	39012	Meran	0039-473/34178	-	-	-
	IBM, Joyce, Workemate-Bullet■ Modem						0039-471/980496	-	2:333/400	-

lfd. Nr.	Nachname	Vorname	Straße	PLZ Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
26	Neueder	Jens	Rudolf-Then-Straße 32	GER 74544 Michelbach / Bilz	0791/ 42877	-	791440000004722	CCWN
	IBM 286/386/486, Atari ST 1040, TRS80 MI■ Modem, Sound Galaxi NX, Scanner, Tape, Borsu				07971/ 250-50	07971/ 250-55	1001667	0715168434
27	Neumann	Christof	Zeitblomstraße 22/2	GER 89077 Ulm /Donau	0731/ 6022568	-	-	-
	IBM, Tandy MII, Tandy M4p■ Novell				0731/ 9749720	-	-	-
28	Obermann	Hartmut	Mozartring 23	GER 89312 Günzburg	08221/ 30248	08221/ 33575	0822130248	Brainstorm
	IBM486, Tandy M4p, Epson PX-8 ■ Modem, Scanner				-	-	2:241/7902.10	07300 ...
29	Retzlaff	Bernd	Kleiner Sand 98	GER 25436 Uetersen	04122/ 43551	-	-	-
	IBM 386, C64, GENIE I■				04103/ 605310	-	-	-
30	Rinio	Gerd	Rennbahnstraße 9	GER 22111 Hamburg	040/6552630	-	-	-
	IBM 486DX66/2, RTS 80, TRS 80 RS ■ Modem				-	-	-	-
31	Ruschinski	Claus	Pommernstraße 21	GER 45770 Marl	02365/ 34646	-	-	-
	IBM 386, TRS80 M I■ Highscreen-Scanner, CoProz IIT387				-	-	-	-
32	Schimmer	Jörg	Stettinerstraße 28	GER 60388 Frankfurt	06109/ 35336	-	-	-
	IBM 486, Schneider CPC■ Modem CSR2400				069/ 3800-2385	-	2:249/70.9	-
33	Schmid	Alexander	Entmannsdorf 5	GER 96317 Kronach /Gehülz	09261/ 53496	-	-	-
	GENIE IIs, GENIE IIIs, CPU 280■ Modem 2400,Prommer80,Ramdisk,Club80Terminal				-	-	2:2400/830	-
34	Schmitz	Rainer	Küferweg 12/1	GER 73099 Adelberg	07166/ 1397	-	-	-
	IBM,Portfolio,NCR Decision Mate V,Joyce,... ■ Modem 1200, Märklin Digital Interface 6050				07161/ 608-475	-	-	-
35	Schoberth	Uwe	Petrus-Waldus-Straße 14	GER 75443 Oetisheim	07041/ 7254	-	-	-
	Alphatronic P3■				0711/ 2011455	-	-	-
36	Scholz	Hans-Werner	Spitalstraße 54	GER 41334 Nettetal	02157/ 3613	-	-	-
	IBM 386, Prof 80, ITT 3030■ Prommer80				-	-	-	-
37	Schroers	Horst-Dieter	Breslauer Straße 9	GER 85622 Feldkirchen	089/ 9032615	-	-	-
	IBM■ Modem, Scanner				089/ 9043413	-	-	-
38	Schröder	Gerald	Am Schützenplatz 14	GER 21218 Seevetal	04105/ 2602	-	-	-
	IBM 386SX, Atari 1040STF, Z280■				-	-	-	-
39	Schröer	Egbert	Joachimstraße 18	GER 46284 Dorsten	02362/ 75311	-	-	-
	Portfolio, TRS80 MI, GENIE I, GENIE IIIs■				02362/ 49-9649	-	-	-
40	Schulte	Hartmut	Entenschnabel 8	GER 31311 Uetze	05173/ 1248	05173/ 24631	-	-
	IBM, Z280 u.a.■ Scanner				-	-	-	-
41	Sonnemann	Harald	In den Eckwiesen 9	GER 64405 Fischbachtal	06166/ 8512	-	-	-
	NDR-Klein ■ parallel/seriell, EPROMer				06151/ 92-1265	-	-	-
42	Stumpferl	Stefan	Hasenberglstraße 57	GER 80933 München	089/ 3138193	//0893144001//	-	-
	Amstrad CPC 6128+, GENIE IIIs ■ Modem, 20MB-Wechselplatte, PSG&PIO,SIO, ...				-	-	-	-
43	Sörensen	Rüdiger	Wiesbadener Str. 28B	GER 55252 Mainz-Kastel	06134/65342	-	-	-
					-	-	-	-
44	Tornow	Wilhelm	Elbblick 46	GER 21629 Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
	IBM 386DX, Atari Mega ST4, Tandy M4p■ Streamer,Soundblaster,Modem 2400,F.A.K.S. 910				-	-	-	-
45	Vogel	Michael	Weidenweg 15	GER 41515 Grefenbroich /Laach	02181/45112	-	-	-
					-	-	-	-
46	Volkmer	Richart	Am Spörkel 69	GER 44227 Dortmund	0231/ 752574	-	-	-
	IBM XT, Apple LC, CPC 6128, Osborne, TandyMII■ Modem 2400, ScanMan-Scanner				-	-	-	-
47	Volz	Oliver	Am Ochsenwald 37A	GER 70565 Stuttgart (Rohrerhöhe)	0711/ 744051	-	-	-
	IBM, GENIE IIs■ Modem Avantec				0711/ 685-3013	-	-	-
48	Werner	Heiko	Reichenberger Straße 5	GER 01129 Dresden	0351/ 4608612	-	-	-
	IBM 286/486■ Modem 2400, Scanner				-	-	-	-

Bitte überprüft Eure Daten und teilt uns Änderungen mit!

Beachtet die Umstellung der Postleitzahlen!

Wissenswertes
rund
um
den
Z280

Title: Z280, truly a microsystem in silicon
Author(s): Magill, J.
Author Affiliation: Zilog, Campbell, CA, USA
Conference Title: Midcon/87 Conference Record p.388-93
Publisher: Electron. Conventions Manage, Los Angeles, CA, USA
Publication Date: 1987 Country of Publication: USA vi+516 pp.
Conference Sponsor: IEEE; ERA
Conference Date: 15-17 Sept. 1987 Conference Location: Chicago, IL, USA
Availability: Western Periodicals Co., North Hollywood, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Product Review (R)
Abstract: Zilog's 'system on silicon' called the Z280 provides a 16-bit processor which runs Z80 code together with many other closely coupled advanced system components. Other major blocks include memory management, peripherals, memory refresh logic, cache memory, wait state generators, and a clock oscillator. The peripherals include just about everything else needed, apart from some RAM and ROM to put together a complete system; included are 4 DMA channels, 3 counter/timers, and a UART. Specialised glue logic ties all this together, making a closely coupled, high performance, low cost microsystem. To minimise pin count the Z280 communicates to the external world via a multiplexed address and data bus. (0 Refs)
Descriptors: CMOS integrated circuits; computer architecture; microprocessor chips; VLSI
Identifiers: timers; Zilog; counters; pin count minimisation; multiplexed bus; closely coupled system; microsystem in silicon; system on silicon; Z280; 16-bit processor; Z80 code; memory management; peripherals; memory refresh logic; cache memory; wait state generators; clock oscillator; DMA channels; UART; glue logic; low cost microsystem; multiplexed address and data bus; 16 bit
Class Codes: B1265F (Microprocessors and microcomputers); B2570D (CMOS integrated circuits); C5130 (Microprocessor chips)
Numerical Indexing: word length 1.6E+01 bit

TIS iServer. (Hardware Review) (one of three evaluations of fault-tolerant microcomputers in 'Servers that survive') (evaluation)
Lavin, Paul
PC User ISSUE: n164 PAGINATION: p39(1)
PUBLICATION DATE: July 31, 1991
ARTICLE TYPE: evaluation
AVAILABILITY: FULL TEXT ONLINE LINE COUNT: 00088
SPECIAL FEATURES: illustration; photograph; table
CAPTIONS: PC User verdict. (table)
SOURCE FILE: CD File 275
OPERATING PLATFORM(S): Intel 80486
PRODUCT NAME(S): TIS iServer (Microcomputer) - evaluation
COMPANY NAME(S): TIS - Products
DESCRIPTORS: Evaluation; Fault Tolerance; Microcomputers
SIC CODE: 3571
ISSN: 0263-5720

PC USER VERDICT

Products: TIS iServer
Description: Faul-tolerant Unix server.

Supplies by: TIS

Tel: (0628) 810909

Price: 13,995 pounds sterling

	Poor	Moderate	Good	Excellent
Characteristics	[unkeyable]			
Ease of use	[unkeyable]			
Documentation	[unkeyable]			
Engineering	[unkeyable]			
Performance	[unkeyable]			

* Expensive, but it comes from a pedigree open systems stable with vast-experience of multiuser systems. With TIS you get what you pay for.
TIS is a highly recommended Unix system Value Added Reseller (VAR), which found it couldn't get what it wanted for its Intel-based open systems customers off the shelf. The company commissioned Intel to produce a machine which it christened the iServer and is designed to provide the resources required by small Unix systems users.

The iServer is a short, broad box measuring 14x18x20 inches, reminding you of one of those valuables safes you sometimes find in hotel rooms. It arrives in your office strapped to a wooden pallet and it's better to get two people if you want to move it. The weight comes from the heavy-gauge steel construction and the UPS batteries.

The floppy disk and tape drives are hidden behind a door which isn't connected to the key lock on the top of the unit. The lock serves only to secure the system electrically. There aren't any alarms to tip the wink on an unauthorised intruder. The top fascia panel has LEDs for AC and DC power status, and LAN and serial I/O activity.

There are several strengths of iServers and I examined the top-of-the-line Model 425. It has a 25MHz 486DX chip which makes it considerably more powerful for a multiuser Unix installation than either the 325 or 333 models, which both use 386 CPUs. While the 386 machines get a 64Kb static RAM cache, the 425 has only the 8Kb internal cache in the 486.

There's no socket for an auxiliary floating-point co-processor, nor would you expect one on an Intel board. The motherboard is the same one that Intel puts into its SYP machines. I've used one of those Unisys/Intel boxes for a year with hardly a hiccup.

The standard memory configuration is 8Mb, but expansion to 32Mb is possible, using 8Mb expansion boards that slot into Intel's own 32-bit bus. This bus is compatible with 16 and 8-bit cards, but when you put a memory board in it, it speeds up to provide fast memory access.

There are four 32-bit slots, and if you fill them all with memory, you'll only have one 16-bit slot have one 16-bit slot and one 8-bit slot left after the inclusion of the SCSi and serial I/O controllers. That's not as dire as it sounds, given that the only other likely inhabitant of the iServer would be a 16-bit Ethernet card.

The 16-bit busmastering Adaptec 1540 SCSI controller, whichs sits in one of the 16-bit slots, can deliver 10Mbps DMA transfer. The maximum synchronous SCSI transfer rate is 5Mbps. A second disk controller can be

added for the protection and performance bonuses that duplexing can provide.

The serial I/O controller uses a 20MHz Z280 chip to drive up to four sub-controllers, each of which carries eight RS232 connectors. Eight ports are standard. The system also carries two additional serial ports labelled console, and a parallel printer port.

Standard mass storage ia a 5 1/4-inch floppy disk drive and a 150Mb QIC tape cartridge drive -- something of a standard in the Unix world.

The chassis will accommodate up to two full-height SCSI hard disks. TIS offers options of 120, 325 and 650Mb, although drives that hold 2Gb are available in that form factor. A 1.3Gb DAT drive is an option for unattended large volume backups. If you go for the maximum security configuration (two drives mirrored with duplex controllers), the maximum capacity is 650Mb from four 325Mb half-height drives.

TIS provided a series of measurements using the standard AIM III suite, which establish the 486 iServer in the same league as the Olivetti LSX5010, and the Apricot FTs. The iServer beat the Altos 100/486 on every measure. The tests used were the Multiuser, Random Disk Performance, Sequential Disk Performance and CPU Performance.

Providing a measure of protection from the slings and arrows of the power company is a 300W power supply with a UPS battery backup, which can keep a fully loaded system running for 10 vital minutes so that the file system can be secured. The battery takes 24 hours to recharge.

The TIS iServer runs an enhanced version of AT&T's System V Release 4 Unix. The disk and I/O drivers have been massaged for reliability and performance and the UPS integrated, so that the system will perform an orderly shutdown if the mains power goes away. Disk mirroring software is standard, as is security that complies with the NSCS Orange Book C2 standard.

TIS' software bundle includes TCP/IP, NFS, Xenix and BSD compatibility, X.Window (X11R4) and Open Look. The iServer supports Portable NetWare as well as the ISO/OSI protocol stack.

The TIS iServer is a husky little creation that's been tailored specifically for small Unix systems users. I can't fault it technically in either quality or specification, but in relation to the other grown-up PCs reviewed here, it's pretty expensive.

COPYRIGHT EMAP Business & Computer Publications (UK) 1991

Operating-system software. (includes a related article on accelerating system performance through GUI) (Software Review) (All-Star PC Part 4.) (evaluation)
Leibson, Steven H.
EDN VOL.: v35 ISSUE: n9 PAGINATION: p171(8)
PUBLICATION DATE: April 26, 1990
ARTICLE TYPE: evaluation
AVAILABILITY: FULL TEXT ONLINE LINE COUNT: 00278
SOURCE FILE: CD File 275
PRODUCT NAME(S): Open Desktop (Operating system); MS-DOS 4.01 (Operating system); OS/2 2.0 (Operating system); DESQview 386 (Utility program); Microsoft Windows-386 2.11 (Graphical user interface)

COMPANY NAME(S): Microsoft Corp. - products; Quarterdeck Office Systems - products; Perceptive Solutions Inc. - products; Santa Cruz Operation Inc. - products; Future Domain Corp. - products
DESCRIPTORS: Operating Systems; Software Selection; Evaluation; Comparison ; Graphical User Interface
SIC CODE: 3672; 7371; 7372; 3577
ISSN: 0012-7515
TICKER SYMBOL: MSFT
OPERATING-SYSTEM SOFTWARE

Operating systems get very little respect from users, probably because they get in your way more often than not. Cryptic commands and mysterious command-line switches make text-oriented operating systems difficult to usE. Yet these essential programs define how you interact with your computer. Microsoft's DOS is currently the most popular PC operating system, but it isn't the only player available for high-end Pcs like EDN's All-Star PC. Multitasking operating systems, including Unix and OS/2, are waiting in the on-deck circle.

Many people think DOS is getting long in the tooth. DOS and the PC were introduced together in 1981 and both have since evolved. DOS 1.0 mimicked Digital Research's CP/M operating system, which ran on the 8080 [microprocessor]. But DOS 2.0 gained a character all its own by incorporating hierarchical directories. The most recent release, DOS 4.01, overcomes many of the operating system's lingering limitations. It supports expanded memory, large hard disks, and a limited form of multitasking. The All-Star PC includes DOS 4.01 as one of its operating systems because of the many applications programs available for DOS-based systems.

DOS 4.01 allows the All-Star PC to use the entire 315M bytes of the Seagate Wren Runner hard-disk drive as one physical disk drive. Earlier versions of DOS could handle logical drives no larger than 32M bytes. DOS 3.3 split each of the All-Star PC's 315M-byte Wren Runners into 10 logical drives, a shortcoming that caused problems with some programs. DOS 4.01's solution is much cleaner.

Since DOS's introduction, the industry's definition of just what an OS should provide has evolved. In the heyday of CP/M and DOS 1.0, most PCs had either limited or no graphics hardware. Consequently, the operating systems for these machines provided very little support. Now however, graphics displays are integral to most PCs. Yet DOS still shuns graphics support. It clings to the past because its existing base of applications programs can't take advantage of new graphics services. DOS 4.01 provides a graphics-shell program that creates the illusion of a graphical user interface (GUI), but the applications programs that you activate from that shell must supply their own graphics drivers, like any DOS program. As a result, every DOS applications program looks and acts differently. There's nothing inherently wrong with differentiation, but a wide-open operating environment like DOS extracts a penalty in user inefficiency and in learning time.

Microsoft recognized years ago that DOS was falling short of contemporary operating-system expectancies, so it developed the Windows operating-system shell to add missing features. The earliest versions of Windows provided a standard set of graphics services that allowed applications programmers to produce software with a common look and feel. Unfortunately, Windows can only provide graphics services to programs that are specifically written to make use of them. No software shell can retrofit graphics on a program created ignorant of GUI concepts. But DOS programs still outnumber Windows applications programs by a substantial margin, so most existing PCs currently operate without the Windows GUI.

This situation may change however, as more 80386-and 80486-based PCs enter the workplace. Windows/386 version 2.11, the latest and most powerful version of the package, runs DOS applications. With the added hardware boost from the 80386-microprocessor] family, it also provides memory management and multitasking services. Thus Windows/386 provides a bridge between existing DOS applications and a GUI-based OS: DOS plus Windows.

Quarterdeck Office Systems recognized the limitations inherent in DOS and took a different approach to enhancing DOS's capabilities. The company's DESQview 386 operating-system shell for DOS provides multitasking and windowing services like Microsoft's Windows/386. QEMM (the Quarterdeck expanded memory manager for the 80386) is part of the DESQview 386 package; it provides memory-management services for DESQview. With QEMM, you can convert all memory above the first megabyte into so-called expanded memory (called block-switched memory in the days of CP/M). Many DOS programs can take advantage of expanded memory for large data structures, and the DESQview shell uses expanded memory to provide multitasking under DOS. Windows/386 performs similar feats with expanded memory. However, unlike Windows/386, DESQview doesn't bestow a graphics-interface standard upon application programs. Instead, its windows differ greatly in appearance from one to the next, depending on the user interface built into the underlying program.

EDN's All-Star PC runs both Windows/386 and DESQview 386, but it can only run one at a time because both programs want to manage the system's memory, an even that would lead to conflicts if both operated concurrently. To use both, you must keep two different CONFIG.SYS files on your hard disk and place the one you want in the root directory, then reboot your PC to activate the desired shell program. This procedure lacks any semblance of elegance, but it works.

Concurrency streamlines work

Windows/386 and DESQview 386 make PC users more efficienty by allowing many programs to run concurrently. In such an environment, PC users can jump from program to program instantly as warranted by the day's tasks and by those pesky interruptions everyone deals with. Both programs also allow processing to background tasks, so you can initiate a long processing job and then switch to another task while the background processing occurs. Far too many people state that most PC users do not need and cannot take advantage of concurrent-program operation. That's bunk! Concurrent-program operation allows you to leave one task to handle an interruption and later return to the same spot. Everyone deals with interruptions on a daily basis, so it's foolish to say that few can benefit from the advantages provided by programs like DESQview 386 or Windows/386.

Software compilation, for example, greatly benefits from background processing. Software developers have used background compilation on minicomputers and mainframe computers for decades. A multitasking Os allows you to jump from a debugger to an editor to a compiler and back to the debugger. Similarly, a hardware designer can jump from a schematic-capture package to a simulator to a pc-board-layout package to a thermal-analysis package and back again. Engineers can bery definitely benefit from concurrent-program operation and multitasking.

But DOs applications aren't really designed for multitasking environments and therefore lack features that could make them even more useful. For example, programs designed for a single-tasking environment aren't likely to contain facilities to communicate with other running programs. Only one program at a time can run under a single-tasking OS, so

it would be silly to include such capabilities. Yet real-time communications among programs can be quite useful, particularly for engineering applications.

For example, a schematic-capture program can transmit design changes to a bill-of-materials program to a program that calculates power dissipation to still another program that calculates the project's component cost. The electronic-product-development tools in the latest software release from Mentor Graphics Corp (Beaverton, OR) do exactly that. However these tools, collectively called Software Release 8.0, currently run under Unix and only on workstation-class computers. Without multitasking, you would need to run each program in turn to compute the various effects caused by the design changes. Because a collection of integrated application programs like Software Release 8.0 promises to provide a large jump in individual productivity, the PC market is girding for the jump to multitasking operating systems over the next two years.

Dueling operating systems

Unix and OS/2 are fighting for the lead as the PC multitasking operating system of choice. Most engineers have at least heard of Unix. Workstation and software vendors like Mentor made Unix the primary operating system for their technical applications. GUIs such as the Open Software Foundation's Motif metamorphose Unix from a hacker's delight into an OS that can be used by people with a wide range of computer skills. Intel claims, based on industry surveys, that almost 60% of Unix-based cokmputer systems shipped annually now incorporate 80286 and 80386 [microprocessor]s, and that many of those systems are Pcs. Minicomputers,

RISC-based products, and Motorola 68000-based computers constitute the remainder of the market. PCs are clearly a big part of the Unix world, although Unix currently represents only a small part of the PC's market.

For high-end PC's, OS/2 will be Unix's biggest competitor. OS/2 looks and operates a lot like DOS but provides additional capabilities such as multitasking. With the addition of the Presentation Manager GUI, OS/2 becomes a graphical OS that resembles the DOS/Windows combo. But OS/2 is strictly an OEM product for Microsoft; the company doesn't sell the operating system to end users, at least not yet. Currently, computer OEMs must use Microsoft's OS/2 development kit to buil a custom-tailored version of OS/2 for their specific hardware configurations. Although earlier versions of OS/2 (through version 1.2) run on both 80286-and 80386-based computers, Microsoft's OS/2 version 2.0, currently available only in development-kit form, runs only on computers than employment [microprocessor]s from the 80386 family.

In addition to multitasking, Unix and OS/2 provide another sorely needed feature: the ability to use more than 640K bytes of RAM for a program. This long-standing DOS-imposed limitation seemed inconsequential when DOS was first introduced. Compared to CP/M's 64k-byte limit, DOS's 640k-byte ceiling seemed vast. But software developers have made grand extensions to their products since 1981. They have attempted (and often succeeded) to take on tasks never thought possible on a PC. As a result, word processors have evolved into desktop-publishing packages, schematic-capture programs graple with pc boards of ever increasing complexity, and some PLD compilers now attempt to fit designs into multiple devices. All of these added capabilities demand more memory, so the 640k-byte barrier, once a liberating factor, has now become a constant stumbling block.

Both Unix and OS/2 circumvent DOS's 640k-byte limitation. By their inherent designs, they make use of the memory above the first megabyte as conventional (not extended or expanded) memory. Because Unix and OS/2 version 2.0 don't run on 8088- and 80286-based machines, they don't limit programs to the first megabyte of memory. In addition, both operating systems support virtual-memory operation, which permits applications programs to run as though there were more RAM in the system than is actually present. Virtual-operating systems load only the parts of a program that are required for immediate operations and leave unneeded sections of code on the system's disk until they are needed. Disk capacity becomes the new limiting factor for program size, and hard-disk drives are getting very big indeed.

Unix for the PC

EDN's All-Star PC runs two flavors of Unix--Xenix and Unix/386. Both are offered by the Santa Cruz Operation (SCO). Xenix is a compact version of Unix designed to operate on 80286- and 80386-based PCs. The stock version supports only two types of hard-disk controllers--the de facto standard Western Digital WD1003 controller for ST-506 hard-disk drives and Adaptec's AHA1540 SCSI-host-adapter family. The WD1003 doesn't work with SCSI-based disk drives, and, as discussed in Part 2, the Adaptec boards conflict with Quarterdeck's DOS-extension products. The solution, for Xenix, is to add a special Xenix driver for the IN-2000. Always Technology, the vendor that supplied the IN-2000 SCSI host adapter for the All-Star PC, finished a disk driver for SCO's Xenix just in time for this article.

SCO's Unix/386 is part of the company's Open Desktop package, which combines a version of Unix System V version 3.2, a PC version of Motif, and various standard networking and communications programs into one comprehensive product. Open Desktop is a very big operating system by PC standards. To run it, a PC needs at least 8M bytes of RAM and should have at least 100M bytes of hard-disk space available. Unlike Xenix, SCO's Open Desktop package runs strictly on PCs that employ [microprocessors] from the 80386 family.

The Open Desktop can share the hard disk with other operating systems, such as DOS and OS/2, through DOS's disk-partitioning scheme. However, you must use the respective operating systems' FDISK programs to activate a partition. The active partition determines which operating system will take over the next time the computer boots. SCO Xenix has a "DOS" command that is supposed to switch a system over to DOS, but this command didn't work on the All-Star PC. Steven Katz, the vice president of software engineering at Always Technology, surmised that Unix's DOS command failed because it didn't recognize the large DOS 4.01 disk partition as a valid DOS partition. DOS 4.01's disk partition differs from the partitions created by older DOS vintages.

A tenuous, 3-way OS marriage

Trying to get DOS, OS/2, and Unix/386 to coexist on a hard disk on the All-Star PC presented several problems. The first was getting the right disk drivers for all three operating systems. OS vendors write their disk drivers for the PC's lowest common denominator: the WD1003. Any deviation requires a hard-disk driver to link the operating system to the "foreign" controller. The All-Star PC's IN-2000 incorporates a BIOS ROM that provides its own DOS driver, so no extra software is needed. Always also supplied a loadable driver for Xenix because Xenix does not use the BIOS ROM's disk-access routines. Because OS/2 and Unix also won't make use of the IN-2000's BIOS ROM, they too require disk drivers. As the deadline for this

article approached, it appeared that Always Technology would not be able to supply OS/2 and Unix drivers for its board in time for this project.

Fortunately, Perceptive Solutions Inc (PSI) had just completed its first production run of Hyperstore 1600 disk controllers. The Hyperstore 1600 is a caching disk controller with interchangeable media adapters. It can operate ST-506-, ESDI-, and SCSI-based hard disks. The controller board accepts 4M bytes of dynamic RAM (DRAM) for its disk cache and employs a Zilog Z280 [microprocessor] for SCSI and cache control. An auxiliary memory board can boost the controller's cache RAM to 20M bytes. The Hyperstore 1600 operates in either a native SSP (standard-storage-protocol) mode or a WD1003 emulation mode. PSI says that its SSP mode is slightly faster than the WD1003 mode. But, any PC operating system can use the emulation mode without additional disk drivers. The major drawback to emulation is that the WD1003 register-set definition limits the hard-disk capacity to two 450M-byte drives. Fortunately, that limitation presented no problems for the All-Star PC's two 315M-byte drives.

Though installing the Hyperstore 1600 presented few problems, there was one tough nut that had to be cracked--a conflict between the Hyperstore 1600's integral floppy-disk controller and the Compaticard IV card from Microsolutions, which was already in the All-Star PC. You can't disable the Hyperstore 1600's floppy-disk controller (although you can move it to a secondary address, which also conflicts with the Compaticard IV). As a result, the final configuration gives control of the All-Star PC's two 5-1/4-in. floppy-disk drives to the Hyperstore 1600 and retains the Compaticard IV for the two 3-1/2-in. floppy-disk drives. In this configuration, the Hyperstore 1600 uses the primary floppy-disk-controller address, and the Compaticard IV occupies the secondary address.

Tape-drive control became a problem when the Hyperstore 1600 entered the mix. Because I would be switching between the IN-2000 and the Hyperstore 1600, the All-Star PC needed a new tape-backup scheme. PSI has not yet developed tape drivers for its SCSI controller, and the Novastor tape-utility program supplied with the IN-2000 only works with the IN-2000. The addition of a third SCSI host adapter, Future Domain's TMC-860, solved the problem. Future Domain introduced drivers for the Sytos tape-backup program late last year and supplied that software for the All-Star PC project.

Sytos from Sytron Corp (Westboro, MA) is a general-purpose tape-backup utility. The Future Domain drivers link Sytos and the TMC-860 to the All-Star PC's Exabyte EXB-8200 cartridge-tape subsystem. Packing three SCSI host adapters into one PC is hardly an efficient configuration. One SCSI controller should be able to control all of the All-Star PC's SCSI-based mass-storage devices, but the TMC-860 is a bit slow for the Wren Runner disk drives, the IN-2000 has a limited repertoire of OS drivers, and the Hyperstore 1600 lacks tape drivers. The vendors of these SCSI host adapters will undoubtedly solve most of these problems in the near future, but for now, the All-Star PC must rely on the services of several SCSI host adapters to accomplish all of the project's goals.

Using either the IN-2000 or the Hyperstore 1600, DOS 4.01 and Windows/386 loaded easily. However, while using the Hyperstore 1600 to load either SCO's Open Desktop or Microsoft's OS/2 version 2.0, the All-Star PC locked up shortly after beginning the installation. As it turned out, neither problem was related to the Hyperstore 1600. Unable to solve the Unix and OS/2 installation problems alone, I took the All-Star PC and Steven Katz to Colorado Springs to visit Ron Sartore at Cheetah's R&D labs.

Sartore solved the Unix/386 problem in less than a day. Early in the installation process, Unix/386 tries to gauge the size of a PC's memory.

When it tried to read the first byte of the 17th megabyte, the All-star PC's memory subsystem refused to complete the memory-bus handshake and the system froze. A new address-decoder PLD cured that problem, and we loaded SCO's Open Desktop without further trouble.

OS/2 version 2.0 also didn't like the All-Star PC's configuration and refused to load. However, in this case the PC's hardware wasn't freezing; the processor continued to execute code, but externally, the PC was dead and wouldn't load OS/2 from the floppy. The same problem appeared when we tried to load OS/2 version 2.0 on a Cheetah system using a genuine WD1003 disk controller, again exonerating the Hyperstore 1600. It was probably a bit of a reach to expect that this pre-release version of OS/2 would run on the All-Star PC's complicated configuration, but we made the attempt anyway. By then time had run out, so we took a fall-back position. OS/2 version 1.1 loaded without any problems.

High-end PCs based on Intel's 80386 [microprocessor] family give you many operating-system alternatives. For today's PC software, such as the engineering packages to be discussed in Part 5, DOS is the clear operating system of choice. However, the advent of multitasking, virtual-operating systems such as Unix/386 and OS/2 version 2.0 will allow software developers to create more powerful, interrelated sets of application programs for PCs in the years ahead.

COPYRIGHT Cahners Publishing Co. 1990

Adding DMA to other functions boosts speed. (direct memory access) (includes related article on DMA basics)

Leibson, Steven H.
EDN VOL.: v34 ISSUE: n22 PAGINATION: p95(9)
PUBLICATION DATE: Oct 26, 1989
AVAILABILITY: FULL TEXT ONLINE LINE COUNT: 00214
SPECIAL FEATURES: illustration; photograph; chart
CAPTIONS: (Two types of DMA transfers.); The combination of a DMAC and several system peripheral devices.
SOURCE FILE: CD File 275
DESCRIPTORS: Controllers; Technology; Data Transfer Rate; Direct Memory Access; Integrated circuits; Performance Improvement

Adding DMA to other functions boosts speed

You can't beat DMA for high-speed data transfers in a [microprocessor]- or [mu]C-based system. DMA controllers (DMACs) waste no bus cycles to fetch instructions; these hard-wired machines already know how to do the job. DMACs are really zero-instruction-set computers and offer superior performance to even the much-publicized reduced-instruction-set computers (RISCs), at least for transferring data.

Because they provide the highest possible data-transfer rates, DMACs quickly became an essential component in the [microprocessor]'s support-chip retinue. Today, however, DMACs appear to have been almost totally eclipsed by the 32-bit processor architectures introduced by semiconductor manufacturers during the last two or three years. In fact, many IC vendors currently offer no DMACs or 16-bit controllers to support their highly touted, 32-bit RISC and complex-instruction-set computer power-houses.

But the DMAC isn't dead yet. It has simply moved into other types of ICs and fused with other IC functions. These hybrid controllers can deliver performance superior to a multichip solution in a form that's easier for

you to use in a system.

One company, Siemens Components Inc, continues to advance the idea of discrete DMAC ICs. Siemens developed its 4-channel 82257 DMAC a few years ago, and it's still available for $30 (1000). However, Siemens has upgraded the 82257 design twice, and the latest product of that evolution is the 82258A advanced DMA (ADMA) IC.

The 82258A supports only 8- and 16-bit flow-through transfers, (see box, "DMA basics: flow-through and fly-by transfers"). Rather than build a 32-bit device, the 82258A's designers chose to use additional transistors so they could include other capabilities. For instance, one of the 82258A's four DMA channels supports a multiplexed mode that services as many as 32 devices without [microprocessor] intervention. The 82258A also sports features such as data comparison and matching (which terminates a transfer or triggers a change in transfer parameters when a match occurs), verification (which compares two data blocks without moving data), and translation (which uses a translation table to perform data conversion on the fly during transfers). A 10-MHz 82258A costs $80 (1000).

32-bit DMACs suit 32-bit [mu]Cs

The 82258A does not support 32-bit flow-through data transfers but can manage 32-bit fly-by transfers because its address incrementer can count by four. However, many processor-based systems that employ the latest crop of 32-bit [microprocessor]s require full 32-bit performance from a DMAC. Recognizing this need, Intel Corp currently offers two ICs that incorporate 8-channel, 32-bit DMACs: the 82370 and the 82380. These two devices are companions to the company's 80376 and 80386 [microprocessor]s. Both chips are 32-bit devices with equivalent features. The address and data buses of the 82370 match those of the 80376 [microprocessor] and the 82380's buses mate to the 80386 [microprocessor]. The 82370 (16 MHz) costs $45 (1000) and the 82380 (16 MHz) costs $70 (1000).

The 82370 and the 82380 are not simply DMACs. Their designers also gave them several system-level functions, such as a 20-level interrupt controller, reset logic, a programmable wait-state generator, four interval timers, and a dynamic RAM (DRAM) refresh controller (Fig 1). The combination of these components results in a device that offers more performance than you might extract from the individual ICs and saves you precious board space.

Integration breeds performance

Both the [microprocessor] and the DMAC share the 82370's and the 82380's wait-state generator. The generator lets you create six wait-state settings for your system: three for the memory space and three for I/O space. During an access cycle, the device being addressed drives two input pins that select one of the preset wait-state counts. The generator then postpones the cycle's completion by the requested number of counts. The combination of the wait-state generator and the DMAC on one chip eliminates the need to add any additional wait-state circuitry to your system design.

Intel recently introduced a similar device—the 82357 integrated system peripheral--as part of its Extended Industry Standard Architecture (EISA) chip set. (EISA is a specification for an enhanced version of IBM's PC/AT bus. It accommodates 32-bit expansion cards and multiprocessing.) Intel's 82357 incorporates a timer/counter, an interrupt controller, a bus arbiter, and a DRAM refresh controller, in addition to a DMAC, which supports EISA's three special burst-DMA modes. Intel only sells the 82357

with a companion 82358 EISA bus controller. The 2-chip set (25 MHz) costs $99 (1000).

Zilog offers a similar device for its 16-bit [microprocessor]s--the Z16C20 General Logic Unit. It can control peripheral devices and several types of system memory in addition to performing DMA transfers. The Z16C20 incorporates a 1-channel DMAC, DRAM and static-RAM controllers, a wait-state generator, an EPROM controller that speeds up EPROM accesses by as much as 25%, and two 16-bit timers. One of the timers can also serve as a watchdog timer. The Z16C20's DMAC operates only in the fly-by mode. The device costs $13.33 (100).

Merging [microprocessor]s and DMACs

Intel's 82370, 82380, and 82357 and Zilog's Z16C20 are part of a definite trend in DMAC IC design. Many chip designers are using the bounty of extra transistors, made available by the increasing circuit densities of advanced IC fabrication processes, for additional system functions rather than devoting all their IC's transistors to the DMA controller. To put it another way, many types of ICs now incorporate DMACs. In fact, as Table 1 illustrates, an increasing number of [microprocessor]s and [mu]Cs include DMACs as part of the processors' basic feature set.

Intel's recently introduced 80960CA [microprocessor] is an excellent example of a processor with an integrated DMAC. The 80960CA incorporates a 4-channel DMAC in addition to an interrupt controller, a bus controller, and an 80960 processor core. The [microprocessor]'s DMAC is so tightly woven into the fabric of the chip that it shares many of the IC's hardware resources with the processor core. These resources include the integer unit (for incrementing and decrementing registers) and the microcode ROM.

Through a variety of clever design features, the combined abilities of the 80960CA's processor core, DMAC, bus controller, and interrupt controller let the 80960CA achieve better system performance than if these functions were implemented with individual ICs. The interaction between the 80960CA's DMAC and bus controller demonstrates the real power of the combined on-chip functions.

As Fig 2 illustrates, two 128-bit data buses link the [microprocessor]'s DMAC, bus controller, data registers, and 1k-byte internal data RAM. Because the DMAC and the bus controller use these very wide buses, they can minimize the number of data transfers made inside the IC by packing the data into 128-bit words.

Move 16 bytes per bus cycle

Using only one internal-bus cycle, the 80960's DMAC can perform a 16-byte transfer between the [microprocessor]'s registers or on-chip RAM and the bus controller. In addition, because there are two of these wide data buses on the 80960CA, the [microprocessor]'s scheduling circuitry never has to turn one of these wide buses around between transfers. Packaging and pin limitations make 128-bit buses impractical as external [microprocessor] buses, and two such external buses are simply out of the question. But if critical functions are placed on the chip, as they are in the 80960CA, you can employ such a performance-boosting architecture for a 1-chip [microprosessor] design.

The 80960CA's bus controller uses a memory-region configuration table that lets you partition the chip's addresses space into 16 regions. For each region, you can specify

* the bus width (8-, 16-, or 32-bit)

* the byte ordering (big-or-little-endian)

* whether or not the chip's burst-mode feature should be enabled

* the number of wait states required

* whether or not to use address pipelining. The configuration table specifies how the external bus should operate for transactions initiated by both the processor core and the DMAC.

I/O requests from the 80960CA's processor core and from the DAMAC appear nearly identical to those from the bus controller. The [microprocessor]'s scheduling circuitry treats the DMAC as a true coprocessor and intermixes its requests with requests from the CPU in the chip's instruction pipeline. In addition, the processor's instruction set includes two DMa instructions ("set up DMA" and "update DMA") that facilitate construction of fast interrupt service routines. DMA instructions represent just one more example of the enhanced performance that results from building a DMAC into a [microprocessor].

A DMAC can transfer data faster than a CPU operating at the same clock rate because the DMAC is essentially a zero-instruction-set computer. While transferring data, a DMAC uses no bus bandwidth for fetching instructions. In addition, the transfer of bus control between a CPU and a DMAC requires less time than an interrupt context switch does. This situation gives DMA-based I/O transfers even more of a performance edge over interrupt-driven transfers. For the 80960CA, intermixing CPU and DMAC requests in the chip's instruction pipeline completely eliminates context-switching overhead.

However, the performance advantages of integrated, on-chip DMA don't just benefit new 32-bit [microprocessor]s. You can find integrated DMACs in newer versions of well-established 8- and 16-bit [microprocessor]s as well. For example, Zilog's Z80 processor architecture serves as the core processor for its Z280 [microprocessor] and for Hitachi's HD64180 (also available from Zilog as the Z180). Motorola's 68000 [microprocessor], a processor with a 16-bit data bus but 32-bit registers, is the core processor for the Signetics 68070 integrated [microprocessor]. The 64180, Z280, and 68070 [microprocessor]s all incorporate DMACs, as shown in Table 1.

[mu]Cs benefit from DMACs, too

Several 8-bit muCs also include integrated DMA facilities. Hitachi's HD647180X, a [mu]C verison of the company's HD64180 [microprocessor], replicates the [microprocessor] version's DMAC capabilities. Other 8-bit [mu]Cs such as Toshiba's TMP90C84X series and NEC's 78k2XX and 78K3XX [mu]Cs provide a form of DMA controlled by the CPU's hardware.

Block diagrams of the NEC and Toshiba [mu]Cs do not show functional blocks marked "DMA controller," but they have DMA capabilities nevertheless. Toshiba calls its DMA operations "micro DMA processing." NEC calls its DMA operations "macro service functions."

In both NEC's and Toshiba's [mu]C families, the DMA controller circuitry is merged into the processor core. Blocks of [muC registers hold source and destination addresses and a terminal count. After a program loads these registers with appropriate values and activates the DMA transfer, the special DMA circuitry in the processor core executes the transfer operation without further program intervention.

NEC has enhanced the macro service functions of its 78K2XX [mu]Cs, and its DMA operations can often replace quite a bit of software. the [mu]C's type C macro service function accepts interruprts from one of the chip's timers. For each interrupt, the function transfers one byte of data to an I/O port and transfers a new value to the interrupting timer. This enables you to create two tables, one containing data to be output and one containing timer values; the type C macro service function transfers the data using the intervals specified in the timer table.

NEC specifically developed this particular DMA mode to allow the [mu]C to control the ramp-up and rampd-down operations of stepper motors. However, this facility can come in handy when you need to transfer data at irregular intervals. Most processors can only do this through interrupt-service routines, which have relatively long and unpredictable latency times. The 78K2XX-[mu]C family's macro service function treats the job as a background task. In addition, the macro service function is invisible to the [mu]C's program when operating and has a short, fixed, latency time.

Intel's 8XC152 [mu]C family incorporates a 2-channel DMAC to support its high-speed, on-chip, multiprotocol communications controller. This controller can handle several network protocols including synchronous data link control and carrier-sense, multiple access with carrier detection, and it is targeted at high-speed LAN applications.

LAN communications generally involve high-speed transfer rates, and a DMAC helps ensure that the [mu]CC can move data from the LAN to memory quickly and efficiently. For example, Ethernet LAN-controller ICs have incorporated DMA capabilities for years. No other data-transfer technique can support the data rates required by high-speed LANs.

Peripheral chips get DMACs

AT&T recently introduced its T7115, a 32-channel HDLC controller designed for digital T1 and primary-rate Integrated Services Digital Network communications systems. It incorporates a DMAC to handle the demanding data rates required by these telecommunications applications. The T7115 costs $55 (1000).

DMACs make sense for other serial-I/O protocols, too. Most data-communications protocols employ inherently asynchronous processes that need DMA's low-latency characteristics. For example, Zilog offers two serial devices that incorporate DMACs: the $75 (100) Z16C30 universal serial controller (USC) and the $ 25 (1000) Z16C35 integrated serial communications controller (ISCC). The Z16C30 supports serial data rates up to 10M bps, making it a prime candidate for DMA service. It also incorporates a 32-byte FIFO buffer so that its integrated DMAC can move data from the USC to memory in 32-byte bursts. Burst operation eliminates much of the bus-exchange overhead that occurs between a CPU and DMAC. That overhead can bog down individual DMA transfers.

The Z16C35's two full-duplex serial channels operate at rates up to 4M bps, and its 4-channel DMAC can move data at 3.1M bytes/sec. With two serial channels, each running at 4M bps, the Z16C35 has data-transfer requirements almost as stringent as the Z16C30's.

One of the key benefits of the integrated-DMAC approach is its support for fly-by DMA transfers, which are the most efficient type. Because DMACs are integrated into serial devices, a fly-by DMA transfer involves only two system components (memory and the serial chip) rather than three (memory,

the serial chip, and the DMAC)). Elimination of the external DMAC also reduces or eliminates the need for additional glue logic.

Integrated DMA controllers require less board space, improve performance, and lower costs. As higher integration levels allow IC designers to redefine existing ICs and invent new devices, you will see an increasing use of DMA to move data in [microprocessor]- and [mu]C-based systems. For data movement, not even RISC [microprocessor]s can outperform DMACs.

COPYRIGHT Cahners Publishing Co. 1989

Nixdorf opts for 16-bit Z280 version of Z80 at low, Unix at high end for retail line. (Nixdorf Computer AG)

Computergram International ISSUE: n1237 PAGINATION: CGI08080007
PUBLICATION DATE: August 8, 1989 LINE COUNT: 00052
SOURCE FILE: CD File 275
PRODUCT NAME(S): Nixdorf POS 2000 (Computer system) - product specifications; Nixdorf POS 2000 10 (Computer system) - marketing; Nixdorf POS 2000 20 (Computer system) - marketing; Nixdorf POS 2000 30 (Computer system) - marketing; Nixdorf POS 2000 40 (Computer system) - marketing
COMPANY NAME(S): Nixdorf Computer AG - marketing
DESCRIPTORS: Marketing Strategy; POS Terminals; Retailing; UNIX
SIC CODE: 3571

NIXDORF OPTS FOR 16-BIT Z280 VERSION OF Z80 AT LOW,

UNIX AT HIGH END FOR RETAIL LINE

Nixdorf Computer Ltd's Retail Division, which launched its new range of point of sale terminals last week (CI No 1,230), looks to the new line to buy it a stronger market penetration of the department store, garage forecourt and hotel and leisure sectors, while retaining its hold on the food multiples, non-food multiples and cash receipting sectors. The technological key to further expansion lies with its new POS 2000 family. Nixdorf's reputation in retail has hitherto been built upon its modular range of terminals, the 8812 family which was launched in 1977 and which can be configured to suit different retail environments. To date, the division has contracts with such illustrious UK high street names as Bally Shoes, Currys, Dixons, Toys 'R' Us, Peacock's Stores and SavaCentre as well as Laura Ashley. In fact this last name was taken by Nixdorf as of great significance in terms of hype for the single European Market, since it has operated on a Euro-pean scale for many years. In an attempt to steal a march on companies such as ICL Data-checker (CI No 1,227) and British Coal's Compower (CI No 1,223) Nixdorf has moved towards Unix with its new POS 200 range.

Uniform designs

Each of the four products in the range is based on a uniform design which comprises a printer, keyboard and customer display, with the central processor housed in an electronic box under the printer. The low-end model, the 2000/10, which is available now, uses a Zilog Z280 16-bit processor with a main memory of 1Mb and is compatible with Nixdorf's 8812 range, offering an upgrade gateway to the POS 2000 family via the exchange of a central processor card. Such terminals are normally backed by a personal computer but the 2000/20 and 2000/30 terminals integrate front and back office applications through their AT-compatible architectures which drive POS 2000 peripherals. The /20 version is an 80286 single user MS-DOS systemc

enabling the user to hot key between the POS application and any back office personal computer application. The /30 80386 multiuser version runs Unix and supports further back office terminals and a second POS terminal. Both these products will ship by the end of the year. Finally, t/range's big boy is the 2000/40 which is designed to be both a POS terminal and controller. The /40 is built around Intel's 80386X chip, has up to 4Mb of memory and interfaces with RS232, RS485, as well as OCIA for scanners, SCSI for mass storage peripherals, an>andard Nixdorf interfaces. It can be equiped with a 3.5" floppy and 3.5" hard disk, s}porting operating systems such as Flexos and Unix. This model will be ready to ship some time in 1990. So how do these terminals fit in with Nixdorf's marketing drive? Well, the Retail Division is particularly targeting the /10 and /40 models at department stores which it believes will find the full screen capability this configuration affords useful for train-ing and for sales pushes; the /10 and /30 terminals offering open systems compatibility and connectivity to a variety of peripherals will, it is hoped wow, garage forecourt managers; while the /10 and /20 versions offered as stand-alones with the option of a special catering keyboard are intended to hook the hotel, and leisure sectors.

COPYRIGHT Apt Data Services 1989

Wissenswertes rund um den Z280

Title: Z280, truly a microsystem in silicon
Author(s): Magill, J.
Author Affiliation: Zilog, Campbell, CA, USA
Conference Title: Midcon/87 Conference Record p.388-93
Publisher: Electron. Conventions Manage, Los Angeles, CA, USA
Publication Date: 1987 Country of Publication: USA vi+516 pp.
Conference Sponsor: IEEE; ERA
Conference Date: 15-17 Sept. 1987 Conference Location: Chicago, IL, USA
Availability: Western Periodicals Co., North Hollywood, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Product Review (R)
Abstract: Zilog's 'system on silicon' called the Z280 provides a 16-bit processor which runs Z80 code together with many other closely coupled advanced system components. Other major blocks include memory management, peripherals, memory refresh logic, cache memory, wait state generators, and a clock oscillator. The peripherals include just about everything else needed, apart from some RAM and ROM to put together a complete system; included are 4 DMA channels, 3 counter/timers, and a UART. Specialised glue logic ties all this together, making a closely coupled, high performance, low cost microsystem. To minimise pin count the Z280 communicates to the external world via a multiplexed address and data bus. (0 Refs)
Descriptors: CMOS integrated circuits; computer architecture; microprocessor chips; VLSI
Identifiers: timers; Zilog; counters; pin count minimisation; multiplexed bus; closely coupled system; microsystem in silicon; system on silicon; Z280; 16-bit processor; Z80 code; memory management; peripherals; memory refresh logic; cache memory; wait state generators; clock oscillator; DMA channels; UART; glue logic; low cost microsystem; multiplexed address and data bus; 16 bit
Class Codes: B1265F (Microprocessors and microcomputers); B2570D (CMOS integrated circuits); C5130 (Microprocessor chips)
Numerical Indexing: word length 1.6E+01 bit

TIS iServer. (Hardware Review) (one of three evaluations of fault-tolerant microcomputers in 'Servers that survive') (evaluation)
Lavin, Paul
PC User ISSUE: n164 PAGINATION: p39(1)
PUBLICATION DATE: July 31, 1991
ARTICLE TYPE: evaluation
AVAILABILITY: FULL TEXT ONLINE LINE COUNT: 00088
SPECIAL FEATURES: illustration; photograph; table
CAPTIONS: PC User verdict. (table)
SOURCE FILE: CD File 275
OPERATING PLATFORM(S): Intel 80486
PRODUCT NAME(S): TIS iServer (Microcomputer) - evaluation
COMPANY NAME(S): TIS - Products
DESCRIPTORS: Evaluation; Fault Tolerance; Microcomputers
SIC CODE: 3571
ISSN: 0263-5720

PC USER VERDICT

Products: TIS iServer
Description: Faul-tolerant Unix server.

Supplies by: TIS

Tel: (0628) 810909

Price: 13,995 pounds sterling

	Poor	Moderate	Good	Excellent
Characteristics	[unkeyable]			
Ease of use	[unkeyable]			
Documentation	[unkeyable]			
Engineering	[unkeyable]			
Performance	[unkeyable]			

* Expensive, but it comes from a pedigree open systems stable with vast-experience of multiuser systems. With TIS you get what you pay for.

TIS is a highly recommended Unix system Value Added Reseller (VAR), which found it couldn't get what it wanted for its Intel-based open systems customers off the shelf. The company commissioned Intel to produce a machine which it christened the iServer and is designed to provide the resources required by small Unix systems users.

The iServer is a short, broad box measuring 14x18x20 inches, reminding you of one of those valuables safes you sometimes find in hotel rooms. It arrives in your office strapped to a wooden pallet and it's better to get two people if you want to move it. The weight comes from the heavy-gauge steel construction and the UPS batteries.

The floppy disk and tape drives are hidden behind a door which isn't connected to the key lock on the top of the unit. The lock serves only to secure the system electrically. There aren't any alarms to tip the wink on an unauthorised intruder. The top fascia panel has LEDs for AC and DC power status, and LAN and serial I/O activity.

There are several strengths of iServers and I examined the top-of-the-line Model 425. It has a 25MHz 486DX chip which makes it considerably more powerful for a multiuser Unix installation than either the 325 or 333 models, which both use 386 CPUs. While the 386 machines get a 64Kb static RAM cache, the 425 has only the 8Kb internal cache in the 486.

There's no socket for an auxiliary floating-point co-processor, nor would you expect one on an Intel board. The motherboard is the same one that Intel puts into its SYP machines. I've used one of those Unisys/Intel boxes for a year with hardly a hiccup.

The standard memory configuration is 8Mb, but expansion to 32Mb is possible, using 8Mb expansion boards that slot into Intel's own 32-bit bus. This bus is compatible with 16 and 8-bit cards, but when you put a memory board in it, it speeds up to provide fast memory access.

There are four 32-bit slots, and if you fill them all with memory, you'll only have one 16-bit slot have one 16-bit slot and one 8-bit slot left after the inclusion of the SCSi and serial I/O controllers. That's not as dire as it sounds, given that the only other likely inhabitant of the iServer would be a 16-bit Ethernet card.

The 16-bit busmastering Adaptec 1540 SCSI controller, whichs sits in one of the 16-bit slots, can deliver 10Mbps DMA transfer. The maximum synchronous SCSI transfer rate is 5Mbps. A second disk controller can be

added for the protection and performance bonuses that duplexing can provide.

The serial I/O controller uses a 20MHz Z280 chip to drive up to four sub-controllers, each of which carries eight RS232 connectors. Eight ports are standard. The system also carries two additional serial ports labelled console, and a parallel printer port.

Standard mass storage ia a 5 1/4-inch floppy disk drive and a 150Mb QIC tape cartridge drive -- something of a standard in the Unix world.

The chassis will accommodate up to two full-height SCSI hard disks. TIS offers options of 120, 325 and 650Mb, although drives that hold 2Gb are available in that form factor. A 1.3Gb DAT drive is an option for unattended large volume backups. If you go for the maximum security configuration (two drives mirrored with duplex controllers), the maximum capacity is 650Mb from four 325Mb half-height drives.

TIS provided a series of measurements using the standard AIM III suite, which establish the 486 iServer in the same league as the Olivetti LSX5010, and the Apricot FTs. The iServer beat the Altos 100/486 on every measure. The tests used were the Multiuser, Random Disk Performance, Sequential Disk Performance and CPU Performance.

Providing a measure of protection from the slings and arrows of the power company is a 300W power supply with a UPS battery backup, which can keep a fully loaded system running for 10 vital minutes so that the file system can be secured. The battery takes 24 hours to recharge.

The TIS iServer runs an enhanced version of AT&T's System V Release 4 Unix. The disk and I/O drivers have been massaged for reliability and performance and the UPS integrated, so that the system will perform an orderly shutdown if the mains power goes away. Disk mirroring software is standard, as is security that complies with the NSCS Orange Book C2 standard.

TIS' software bundle includes TCP/IP, NFS, Xenix and BSD compatibility, X.Window (X11R4) and Open Look. The iServer supports Portable NetWare as well as the ISO/OSI protocol stack.

The TIS iServer is a husky little creation that's been tailored specifically for small Unix systems users. I can't fault it technically in either quality or specification, but in relation to the other grown-up PCs reviewed here, it's pretty expensive.

COPYRIGHT EMAP Business & Computer Publications (UK) 1991

Operating-system software. (includes a related article on accelerating system performance through GUI) (Software Review) (All-Star PC Part 4.) (evaluation)
Leibson, Steven H.
EDN VOL.: v35 ISSUE: n9 PAGINATION: p171(8)
PUBLICATION DATE: April 26, 1990
ARTICLE TYPE: evaluation
AVAILABILITY: FULL TEXT ONLINE LINE COUNT: 00278
SOURCE FILE: CD File 275
PRODUCT NAME(S): Open Desktop (Operating system); MS-DOS 4.01 (Operating system); OS/2 2.0 (Operating system); DESQview 386 (Utility program); Microsoft Windows-386 2.11 (Graphical user interface)

COMPANY NAME(S): Microsoft Corp. - products; Quarterdeck Office Systems - products; Perceptive Solutions Inc. - products; Santa Cruz Operation Inc. - products; Future Domain Corp. - products
DESCRIPTORS: Operating Systems; Software Selection; Evaluation; Comparison ; Graphical User Interface
SIC CODE: 3672; 7371; 7372; 3577
ISSN: 0012-7515
TICKER SYMBOL: MSFT
OPERATING-SYSTEM SOFTWARE

Operating systems get very little respect from users, probably because they get in your way more often than not. Cryptic commands and mysterious command-line switches make text-oriented operating systems difficult to usE. Yet these essential programs define how you interact with your computer. Microsoft's DOS is currently the most popular PC operating system, but it isn't the only player available for high-end Pcs like EDN's All-Star PC. Multitasking operating systems, including Unix and OS/2, are waiting in the on-deck circle.

Many people think DOS is getting long in the tooth. DOS and the PC were introduced together in 1981 and both have since evolved. DOS 1.0 mimicked Digital Research's CP/M operating system, which ran on the 8080 [microprocessor]. But DOS 2.0 gained a character all its own by incorporating hierarchical directories. The most recent release, DOS 4.01, overcomes many of the operating system's lingering limitations. It supports expanded memory, large hard disks, and a limited form of multitasking. The All-Star PC includes DOS 4.01 as one of its operating systems because of the many applications programs available for DOS-based systems.

DOS 4.01 allows the All-Star PC to use the entire 315M bytes of the Seagate Wren Runner hard-disk drive as one physical disk drive. Earlier versions of DOS could handle logical drives no larger than 32M bytes. DOS 3.3 split each of the All-Star PC's 315M-byte Wren Runners into 10 logical drives, a shortcoming that caused problems with some programs. DOS 4.01's solution is much cleaner.

Since DOS's introduction, the industry's definition of just what an OS should provide has evolved. In the heyday of CP/M and DOS 1.0, most PCs had either limited or no graphics hardware. Consequently, the operating systems for these machines provided very little support. Now however, graphics displays are integral to most PCs. Yet DOS still shuns graphics support. It clings to the past because its existing base of applications programs can't take advantage of new graphics services. DOS 4.01 provides a graphics-shell program that creates the illusion of a graphical user interface (GUI), but the applications programs that you activate from that shell must supply their own graphics drivers, like any DOS program. As a result, every DOS applications program looks and acts differently. There's nothing inherently wrong with differentiation, but a wide-open operating environment like DOS extracts a penalty in user inefficiency and in learning time.

Microsoft recognized years ago that DOS was falling short of contemporary operating-system expectancies, so it developed the Windows operating-system shell to add missing features. The earliest versions of Windows provided a standard set of graphics services that allowed applications programmers to produce software with a common look and feel. Unfortunately, Windows can only provide graphics services to programs that are specifically written to make use of them. No software shell can retrofit graphics on a program created ignorant of GUI concepts. But DOS programs still outnumber Windows applications programs by a substantial margin, so most existing PCs currently operate without the Windows GUI.

This situation may change however, as more 80386-and 80486-based PCs enter the workplace. Windows/386 version 2.11, the latest and most powerful version of the package, runs DOS applications. With the added hardware boost from the 80386-microprocessor] family, it also provides memory management and multitasking services. Thus Windows/386 provides a bridge between existing DOS applications and a GUI-based OS: DOS plus Windows.

Quarterdeck Office Systems recognized the limitations inherent in DOS and took a different approach to enhancing DOS's capabilities. The company's DESQview 386 operating-system shell for DOS provides multitasking and windowing services like Microsoft's Windows/386. QEMM (the Quarterdeck expanded memory manager for the 80386) is part of the DESQview 386 package; it provides memory-management services for DESQview. With QEMM, you can convert all memory above the first megabyte into so-called expanded memory (called block-switched memory in the days of CP/M). Many DOS programs can take advantage of expanded memory for large data structures, and the DESQview shell uses expanded memory to provide multitasking under DOS. Windows/386 performs similar feats with expanded memory. However, unlike Windows/386, DESQview doesn't bestow a graphics-interface standard upon application programs. Instead, its windows differ greatly in appearance from one to the next, depending on the user interface built into the underlying program.

EDN's All-Star PC runs both Windows/386 and DESQview 386, but it can only run one at a time because both programs want to manage the system's memory, an even that would lead to conflicts if both operated concurrently. To use both, you must keep two different CONFIG.SYS files on your hard disk and place the one you want in the root directory, then reboot your PC to activate the desired shell program. This procedure lacks any semblance of elegance, but it works.

Concurrency streamlines work

Windows/386 and DESQview 386 make PC users more efficienty by allowing many programs to run concurrently. In such an environment, PC users can jump from program to program instantly as warranted by the day's tasks and by those pesky interruptions everyone deals with. Both programs also allow processing to background tasks, so you can initiate a long processing job and then switch to another task while the background processing occurs. Far too many people state that most PC users do not need and cannot take advantage of concurrent-program operation. That's bunk! Concurrent-program operation allows you to leave one task to handle an interruption and later return to the same spot. Everyone deals with interruptions on a daily basis, so it's foolish to say that few can benefit from the advantages provided by programs like DESQview 386 or Windows/386.

Software compilation, for example, greatly benefits from background processing. Software developers have used background compilation on minicomputers and mainframe computers for decades. A multitasking Os allows you to jump from a debugger to an editor to a compiler and back to the debugger. Similarly, a hardware designer can jump from a schematic-capture package to a simulator to a pc-board-layout package to a thermal-analysis package and back again. Engineers can bery definitely benefit from concurrent-program operation and multitasking.

But DOs applications aren't really designed for multitasking environments and therefore lack features that could make them even more useful. For example, programs designed for a single-tasking environment aren't likely to contain facilities to communicate with other running programs. Only one program at a time can run under a single-tasking OS, so

it would be silly to include such capabilities. Yet real-time communications among programs can be quite useful, particularly for engineering applications.

For example, a schematic-capture program can transmit design changes to a bill-of-materials program to a program that calculates power dissipation to still another program that calculates the project's component cost. The electronic-product-development tools in the latest software release from Mentor Graphics Corp (Beaverton, OR) do exactly that. However these tools, collectively called Software Release 8.0, currently run under Unix and only on workstation-class computers. Without multitasking, you would need to run each program in turn to compute the various effects caused by the design changes. Because a collection of integrated application programs like Software Release 8.0 promises to provide a large jump in individual productivity, the PC market is girding for the jump to multitasking operating systems over the next two years.

Dueling operating systems

Unix and OS/2 are fighting for the lead as the PC multitasking operating system of choice. Most engineers have at least heard of Unix. Workstation and software vendors like Mentor made Unix the primary operating system for their technical applications. GUIs such as the Open Software Foundation's Motif metamorphose Unix from a hacker's delight into an OS that can be used by people with a wide range of computer skills. Intel claims, based on industry surveys, that almost 60% of Unix-based cokmputer systems shipped annually now incorporate 80286 and 80386 [microprocessor]s, and that many of those systems are Pcs. Minicomputers,

RISC-based products, and Motorola 68000-based computers constitute the remainder of the market. PCs are clearly a big part of the Unix world, although Unix currently represents only a small part of the PC's market.

For high-end PC's, OS/2 will be Unix's biggest competitor. OS/2 looks and operates a lot like DOS but provides additional capabilities such as multitasking. With the addition of the Presentation Manager GUI, OS/2 becomes a graphical OS that resembles the DOS/Windows combo. But OS/2 is strictly an OEM product for Microsoft; the company doesn't sell the operating system to end users, at least not yet. Currently, computer OEMs must use Microsoft's OS/2 development kit to buil a custom-tailored version of OS/2 for their specific hardware configurations. Although earlier versions of OS/2 (through version 1.2) run on both 80286-and 80386-based computers, Microsoft's OS/2 version 2.0, currently available only in development-kit form, runs only on computers than employment [microprocessor]s from the 80386 family.

In addition to multitasking, Unix and OS/2 provide another sorely needed feature: the ability to use more than 640K bytes of RAM for a program. This long-standing DOS-imposed limitation seemed inconsequential when DOS was first introduced. Compared to CP/M's 64k-byte limit, DOS's 640k-byte ceiling seemed vast. But software developers have made grand extensions to their products since 1981. They have attempted (and often succeeded) to take on tasks never thought possible on a PC. As a result, word processors have evolved into desktop-publishing packages, schematic-capture programs graple with pc boards of ever increasing complexity, and some PLD compilers now attempt to fit designs into multiple devices. All of these added capabilities demand more memory, so the 640k-byte barrier, once a liberating factor, has now become a constant stumbling block.

Both Unix and OS/2 circumvent DOS's 640k-byte limitation. By their inherent designs, they make use of the memory above the first megabyte as conventional (not extended or expanded) memory. Because Unix and OS/2 version 2.0 don't run on 8088- and 80286-based machines, they don't limit programs to the first megabyte of memory. In addition, both operating systems support virtual-memory operation, which permits applications programs to run as though there were more RAM in the system than is actually present. Virtual-operating systems load only the parts of a program that are required for immediate operations and leave unneeded sections of code on the system's disk until they are needed. Disk capacity becomes the new limiting factor for program size, and hard-disk drives are getting very big indeed.

Unix for the PC

EDN's All-Star PC runs two flavors of Unix--Xenix and Unix/386. Both are offered by the Santa Cruz Operation (SCO). Xenix is a compact version of Unix designed to operate on 80286- and 80386-based PCs. The stock version supports only two types of hard-disk controllers--the de facto standard Western Digital WD1003 controller for ST-506 hard-disk drives and Adaptec's AHA1540 SCSI-host-adapter family. The WD1003 doesn't work with SCSI-based disk drives, and, as discussed in Part 2, the Adaptec boards conflict with Quarterdeck's DOS-extension products. The solution, for Xenix, is to add a special Xenix driver for the IN-2000. Always Technology, the vendor that supplied the IN-2000 SCSI host adapter for the All-Star PC, finished a disk driver for SCO's Xenix just in time for this article.

SCO's Unix/386 is part of the company's Open Desktop package, which combines a version of Unix System V version 3.2, a PC version of Motif, and various standard networking and communications programs into one comprehensive product. Open Desktop is a very big operating system by PC standards. To run it, a PC needs at least 8M bytes of RAM and should have at least 100M bytes of hard-disk space available. Unlike Xenix, SCO's Open Desktop package runs strictly on PCs that employ [microprocessors] from the 80386 family.

The Open Desktop can share the hard disk with other operating systems, such as DOS and OS/2, through DOS's disk-partitioning scheme. However, you must use the respective operating systems' FDISK programs to activate a partition. The active partition determines which operating system will take over the next time the computer boots. SCO Xenix has a "DOS" command that is supposed to switch a system over to DOS, but this command didn't work on the All-Star PC. Steven Katz, the vice president of software engineering at Always Technology, surmised that Unix's DOS command failed because it didn't recognize the large DOS 4.01 disk partition as a valid DOS partition. DOS 4.01's disk partition differs from the partitions created by older DOS vintages.

A tenuous, 3-way OS marriage

Trying to get DOS, OS/2, and Unix/386 to coexist on a hard disk on the All-Star PC presented several problems. The first was getting the right disk drivers for all three operating systems. OS vendors write their disk drivers for the PC's lowest common denominator: the WD1003. Any deviation requires a hard-disk driver to link the operating system to the "foreign" controller. The All-Star PC's IN-2000 incorporates a BIOS ROM that provides its own DOS driver, so no extra software is needed. Always also supplied a loadable driver for Xenix because Xenix does not use the BIOS ROM's disk-access routines. Because OS/2 and Unix also won't make use of the IN-2000's BIOS ROM, they too require disk drivers. As the deadline for this

article approached, it appeared that Always Technology would not be able to supply OS/2 and Unix drivers for its board in time for this project.

Fortunately, Perceptive Solutions Inc (PSI) had just completed its first production run of Hyperstore 1600 disk controllers. The Hyperstore 1600 is a caching disk controller with interchangeable media adapters. It can operate ST-506-, ESDI-, and SCSI-based hard disks. The controller board accepts 4M bytes of dynamic RAM (DRAM) for its disk cache and employs a Zilog Z280 [microprocessor] for SCSI and cache control. An auxiliary memory board can boost the controller's cache RAM to 20M bytes. The Hyperstore 1600 operates in either a native SSP (standard-storage-protocol) mode or a WD1003 emulation mode. PSI says that its SSP mode is slightly faster than the WD1003 mode. But, any PC operating system can use the emulation mode without additional disk drivers. The major drawback to emulation is that the WD1003 register-set definition limits the hard-disk capacity to two 450M-byte drives. Fortunately, that limitation presented no problems for the All-Star PC's two 315M-byte drives.

Though installing the Hyperstore 1600 presented few problems, there was one tough nut that had to be cracked--a conflict between the Hyperstore 1600's integral floppy-disk controller and the Compaticard IV card from Microsolutions, which was already in the All-Star PC. You can't disable the Hyperstore 1600's floppy-disk controller (although you can move it to a secondary address, which also conflicts with the Compaticard IV). As a result, the final configuration gives control of the All-Star PC's two 5-1/4-in. floppy-disk drives to the Hyperstore 1600 and retains the Compaticard IV for the two 3-1/2-in. floppy-disk drives. In this configuration, the Hyperstore 1600 uses the primary floppy-disk-controller address, and the Compaticard IV occupies the secondary address.

Tape-drive control became a problem when the Hyperstore 1600 entered the mix. Because I would be switching between the IN-2000 and the Hyperstore 1600, the All-Star PC needed a new tape-backup scheme. PSI has not yet developed tape drivers for its SCSI controller, and the Novastor tape-utility program supplied with the IN-2000 only works with the IN-2000. The addition of a third SCSI host adapter, Future Domain's TMC-860, solved the problem. Future Domain introduced drivers for the Sytos tape-backup program late last year and supplied that software for the All-Star PC project.

Sytos from Sytron Corp (Westboro, MA) is a general-purpose tape-backup utility. The Future Domain drivers link Sytos and the TMC-860 to the All-Star PC's Exabyte EXB-8200 cartridge-tape subsystem. Packing three SCSI host adapters into one PC is hardly an efficient configuration. One SCSI controller should be able to control all of the All-Star PC's SCSI-based mass-storage devices, but the TMC-860 is a bit slow for the Wren Runner disk drives, the IN-2000 has a limited repertoire of OS drivers, and the Hyperstore 1600 lacks tape drivers. The vendors of these SCSI host adapters will undoubtedly solve most of these problems in the near future, but for now, the All-Star PC must rely on the services of several SCSI host adapters to accomplish all of the project's goals.

Using either the IN-2000 or the Hyperstore 1600, DOS 4.01 and Windows/386 loaded easily. However, while using the Hyperstore 1600 to load either SCO's Open Desktop or Microsoft's OS/2 version 2.0, the All-Star PC locked up shortly after beginning the installation. As it turned out, neither problem was related to the Hyperstore 1600. Unable to solve the Unix and OS/2 installation problems alone, I took the All-Star PC and Steven Katz to Colorado Springs to visit Ron Sartore at Cheetah's R&D labs.

Sartore solved the Unix/386 problem in less than a day. Early in the installation process, Unix/386 tries to gauge the size of a PC's memory.

When it tried to read the first byte of the 17th megabyte, the All-star PC's memory subsystem refused to complete the memory-bus handshake and the system froze. A new address-decoder PLD cured that problem, and we loaded SCO's Open Desktop without further trouble.

OS/2 version 2.0 also didn't like the All-Star PC's configuration and refused to load. However, in this case the PC's hardware wasn't freezing; the processor continued to execute code, but externally, the PC was dead and wouldn't load OS/2 from the floppy. The same problem appeared when we tried to load OS/2 version 2.0 on a Cheetah system using a genuine WD1003 disk controller, again exonerating the Hyperstore 1600. It was probably a bit of a reach to expect that this pre-release version of OS/2 would run on the All-Star PC's complicated configuration, but we made the attempt anyway. By then time had run out, so we took a fall-back position. OS/2 version 1.1 loaded without any problems.

High-end PCs based on Intel's 80386 [microprocessor] family give you many operating-system alternatives. For today's PC software, such as the engineering packages to be discussed in Part 5, DOS is the clear operating system of choice. However, the advent of multitasking, virtual-operating systems such as Unix/386 and OS/2 version 2.0 will allow software developers to create more powerful, interrelated sets of application programs for PCs in the years ahead.

COPYRIGHT Cahners Publishing Co. 1990

Adding DMA to other functions boosts speed. (direct memory access) (includes related article on DMA basics)
Leibson, Steven H.
EDN VOL.: v34 ISSUE: n22 PAGINATION: p95(9)
PUBLICATION DATE: Oct 26, 1989
AVAILABILITY: FULL TEXT ONLINE LINE COUNT: 00214
SPECIAL FEATURES: illustration; photograph; chart
CAPTIONS: (Two types of DMA transfers.); The combination of a DMAC and several system peripheral devices.
SOURCE FILE: CD File 275
DESCRIPTORS: Controllers; Technology; Data Transfer Rate; Direct Memory Access; Integrated circuits; Performance Improvement

Adding DMA to other functions boosts speed

You can't beat DMA for high-speed data transfers in a [microprocessor]- or [mu]C-based system. DMA controllers (DMACs) waste no bus cycles to fetch instructions; these hard-wired machines already know how to do the job. DMACs are really zero-instruction-set computers and offer superior performance to even the much-publicized reduced-instruction-set computers (RISCs), at least for transferring data.

Because they provide the highest possible data-transfer rates, DMACs quickly became an essential component in the [microprocessor]'s support-chip retinue. Today, however, DMACs appear to have been almost totally eclipsed by the 32-bit processor architectures introduced by semiconductor manufacturers during the last two or three years. In fact, many IC vendors currently offer no DMACs or 16-bit controllers to support their highly touted, 32-bit RISC and complex-instruction-set computer power-houses.

But the DMAC isn't dead yet. It has simply moved into other types of ICs and fused with other IC functions. These hybrid controllers can deliver performance superior to a multichip solution in a form that's easier for

you to use in a system.

One company, Siemens Components Inc, continues to advance the idea of discrete DMAC ICs. Siemens developed its 4-channel 82257 DMAC a few years ago, and it's still available for $30 (1000). However, Siemens has upgraded the 82257 design twice, and the latest product of that evolution is the 82258A advanced DMA (ADMA) IC.

The 82258A supports only 8- and 16-bit flow-through transfers, (see box, "DMA basics: flow-through and fly-by transfers"). Rather than build a 32-bit device, the 82258A's designers chose to use additional transistors so they could include other capabilities. For instance, one of the 82258A's four DMA channels supports a multiplexed mode that services as many as 32 devices without [microprocessor] intervention. The 82258A also sports features such as data comparison and matching (which terminates a transfer or triggers a change in transfer parameters when a match occurs), verification (which compares two data blocks without moving data), and translation (which uses a translation table to perform data conversion on the fly during transfers). A 10-MHz 82258A costs $80 (1000).

32-bit DMACs suit 32-bit [mu]Cs

The 82258A does not support 32-bit flow-through data transfers but can manage 32-bit fly-by transfers because its address incrementer can count by four. However, many processor-based systems that employ the latest crop of 32-bit [microprocessor]s require full 32-bit performance from a DMAC. Recognizing this need, Intel Corp currently offers two ICs that incorporate 8-channel, 32-bit DMACs: the 82370 and the 82380. These two devices are companions to the company's 80376 and 80386 [microprocessor]s. Both chips are 32-bit devices with equivalent features. The address and data buses of the 82370 match those of the 80376 [microprocessor] and the 82380's buses mate to the 80386 [microprocessor]. The 82370 (16 MHz) costs $45 (1000) and the 82380 (16 MHz) costs $70 (1000).

The 82370 and the 82380 are not simply DMACs. Their designers also gave them several system-level functions, such as a 20-level interrupt controller, reset logic, a programmable wait-state generator, four interval timers, and a dynamic RAM (DRAM) refresh controller (Fig 1). The combination of these components results in a device that offers more performance than you might extract from the individual ICs and saves you precious board space.

Integration breeds performance

Both the [microprocessor] and the DMAC share the 82370's and the 82380's wait-state generator. The generator lets you create six wait-state settings for your system: three for the memory space and three for I/O space. During an access cycle, the device being addressed drives two input pins that select one of the preset wait-state counts. The generator then postpones the cycle's completion by the requested number of counts. The combination of the wait-state generator and the DMAC on one chip eliminates the need to add any additional wait-state circuitry to your system design.

Intel recently introduced a similar device--the 82357 integrated system peripheral--as part of its Extended Industry Standard Architecture (EISA) chip set. (EISA is a specification for an enhanced version of IBM's PC/AT bus. It accommodates 32-bit expansion cards and multiprocessing.) Intel's 82357 incorporates a timer/counter, an interrupt controller, a bus arbiter, and a DRAM refresh controller, in addition to a DMAC, which supports EISA's three special burst-DMA modes. Intel only sells the 82357

with a companion 82358 EISA bus controller. The 2-chip set (25 MHz) costs $99 (1000).

Zilog offers a similar device for its 16-bit [microprocessor]s--the Z16C20 General Logic Unit. It can control peripheral devices and several types of system memory in addition to performing DMA transfers. The Z16C20 incorporates a 1-channel DMAC, DRAM and static-RAM controllers, a wait-state generator, an EPROM controller that speeds up EPROM accesses by as much as 25%, and two 16-bit timers. One of the timers can also serve as a watchdog timer. The Z16C20's DMAC operates only in the fly-by mode. The device costs $13.33 (100).

Merging [microprocessor]s and DMACs

Intel's 82370, 82380, and 82357 and Zilog's Z16C20 are part of a definite trend in DMAC IC design. Many chip designers are using the bounty of extra transistors, made available by the increasing circuit densities of advanced IC fabrication processes, for additional system functions rather than devoting all their IC's transistors to the DMA controller. To put it another way, many types of ICs now incorporate DMACs. In fact, as Table 1 illustrates, an increasing number of [microprocessor]s and [mu]Cs include DMACs as part of the processors' basic feature set.

Intel's recently introduced 80960CA [microprocessor] is an excellent example of a processor with an integrated DMAC. The 80960CA incorporates a 4-channel DMAC in addition to an interrupt controller, a bus controller, and an 80960 processor core. The [microprocessor]'s DMAC is so tightly woven into the fabric of the chip that it shares many of the IC's hardware resources with the processor core. These resources include the integer unit (for incrementing and decrementing registers) and the microcode ROM.

Through a variety of clever design features, the combined abilities of the 80960CA's processor core, DMAC, bus controller, and interrupt controller let the 80960CA achieve better system performance than if these functions were implemented with individual ICs. The interaction between the 80960CA's DMAC and bus controller demonstrates the real power of the combined on-chip functions.

As Fig 2 illustrates, two 128-bit data buses link the [microprocessor]'s DMAC, bus controller, data registers, and 1k-byte internal data RAM. Because the DMAC and the bus controller use these very wide buses, they can minimize the number of data transfers made inside the IC by packing the data into 128-bit words.

Move 16 bytes per bus cycle

Using only one internal-bus cycle, the 80960's DMAC can perform a 16-byte transfer between the [microprocessor]'s registers or on-chip RAM and the bus controller. In addition, because there are two of these wide data buses on the 80960CA, the [microprocessor]'s scheduling circuitry never has to turn one of these wide buses around between transfers. Packaging and pin limitations make 128-bit buses impractical as external [microprocessor] buses, and two such external buses are simply out of the question. But if critical functions are placed on the chip, as they are in the 80960CA, you can employ such a performance-boosting architecture for a 1-chip [microprosessor] design.

The 80960CA's bus controller uses a memory-region configuration table that lets you partition the chip's addresses space into 16 regions. For each region, you can specify

* the bus width (8-, 16-, or 32-bit)

* the byte ordering (big-or-little-endian)

* whether or not the chip's burst-mode feature should be enabled

* the number of wait states required

* whether or not to use address pipelining. The configuration table specifies how the external bus should operate for transactions initiated by both the processor core and the DMAC.

I/O requests from the 80960CA's processor core and from the DAMAC appear nearly identical to those from the bus controller. The [microprocessor]'s scheduling circuitry treats the DMAC as a true coprocessor and intermixes its requests with requests from the CPU in the chip's instruction pipeline. In addition, the processor's instruction set includes two DMa instructions ("set up DMA" and "update DMA") that facilitate construction of fast interrupt service routines. DMA instructions represent just one more example of the enhanced performance that results from building a DMAC into a [microprocessor].

A DMAC can transfer data faster than a CPU operating at the same clock rate because the DMAC is essentially a zero-instruction-set computer. While transferring data, a DMAC uses no bus bandwidth for fetching instructions. In addition, the transfer of bus control between a CPU and a DMAC requires less time than an interrupt context switch does. This situation gives DMA-based I/O transfers even more of a performance edge over interrupt-driven transfers. For the 80960CA, intermixing CPU and DMAC requests in the chip's instruction pipeline completely eliminates context-switching overhead.

However, the performance advantages of integrated, on-chip DMA don't just benefit new 32-bit [microprocessor]s. You can find integrated DMACs in newer versions of well-established 8- and 16-bit [microprocessor]s as well. For example, Zilog's Z80 processor architecture serves as the core processor for its Z280 [microprocessor] and for Hitachi's HD64180 (also available from Zilog as the Z180). Motorola's 68000 [microprocessor], a processor with a 16-bit data bus but 32-bit registers, is the core processor for the Signetics 68070 integrated [microprocessor]. The 64180, Z280, and 68070 [microprocessor]s all incorporate DMACs, as shown in Table 1.

[mu]Cs benefit from DMACs, too

Several 8-bit muCs also include integrated DMA facilities. Hitachi's HD647180X, a [mu]C version of the company's HD64180 [microprocessor], replicates the [microprocessor] version's DMAC capabilities. Other 8-bit [mu]Cs such as Toshiba's TMP90C84X series and NEC's 78k2XX and 78K3XX [mu]Cs provide a form of DMA controlled by the CPU's hardware.

Block diagrams of the NEC and Toshiba [mu]Cs do not show functional blocks marked "DMA controller," but they have DMA capabilities nevertheless. Toshiba calls its DMA operations "micro DMA processing." NEC calls its DMA operations "macro service functions."

In both NEC's and Toshiba's [mu]C families, the DMA controller circuitry is merged into the processor core. Blocks of [muC registers hold source and destination addresses and a terminal count. After a program loads these registers with appropriate values and activates the DMA transfer, the special DMA circuitry in the processor core executes the transfer operation without further program intervention.

NEC has enhanced the macro service functions of its 78K2XX [mu]Cs, and its DMA operations can often replace quite a bit of software. the [mu]C's type C macro service function accepts interruprts from one of the chip's timers. For each interrupt, the function transfers one byte of data to an I/O port and transfers a new value to the interrupting timer. This enables you to create two tables, one containing data to be output and one containing timer values; the type C macro service function transfers the data using the intervals specified in the timer table.

NEC specifically developed this particular DMA mode to allow the [mu]C to control the ramp-up and rampd-down operations of stepper motors. However, this facility can come in handy when you need to transfer data at irregular intervals. Most processors can only do this through interrupt-service routines, which have relatively long and unpredictable latency times. The 78K2XX-[mu]C family's macro service function treats the job as a background task. In addition, the macro service function is invisible to the [mu]C's program when operating and has a short, fixed, latency time.

Intel's 8XC152 [mu]C family incorporates a 2-channel DMAC to support its high-speed, on-chip, multiprotocol communications controller. This controller can handle several network protocols including synchronous data link control and carrier-sense, multiple access with carrier detection, and it is targeted at high-speed LAN applications.

LAN communications generally involve high-speed transfer rates, and a DMAC helps ensure that the [mu]CC can move data from the LAN to memory quickly and efficiently. For example, Ethernet LAN-controller ICs have incorporated DMA capabilities for years. No other data-transfer technique can support the data rates required by high-speed LANs.

Peripheral chips get DMACs

AT&T recently introduced its T7115, a 32-channel HDLC controller designed for digital T1 and primary-rate Integrated Services Digital Network communications systems. It incorporates a DMAC to handle the demanding data rates required by these telecommunications applications. The T7115 costs $55 (1000).

DMACs make sense for other serial-I/O protocols, too. Most data-communications protocols employ inherently asynchronous processes that need DMA's low-latency characteristics. For example, Zilog offers two serial devices that incorporate DMACs: the $75 (100) Z16C30 universal serial controller (USC) and the $ 25 (1000) Z16C35 integrated serial communications controller (ISCC). The Z16C30 supports serial data rates up to 10M bps, making it a prime candidate for DMA service. It also incorporates a 32-byte FIFO buffer so that its integrated DMAC can move data from the USC to memory in 32-byte bursts. Burst operation eliminates much of the bus-exchange overhead that occurs between a CPU and DMAC. That overhead can bog down individual DMA transfers.

The Z16C35's two full-duplex serial channels operate at rates up to 4M bps, and its 4-channel DMAC can move data at 3.1M bytes/sec. With two serial channels, each running at 4M bps, the Z16C35 has data-transfer requirements almost as stringent as the Z16C30's.

One of the key benefits of the integrated-DMAC approach is its support for fly-by DMA transfers, which are the most efficient type. Because DMACs are integrated into serial devices, a fly-by DMA transfer involves only two system components (memory and the serial chip) rather than three (memory,

the serial chip, and the DMAC)). Elimination of the external DMAC also reduces or eliminates the need for additional glue logic.

Integrated DMA controllers require less board space, improve performance, and lower costs. As higher integration levels allow IC designers to redefine existing ICs and invent new devices, you will see an increasing use of DMA to move data in [microprocessor]- and [mu]C-based systems. For data movement, not even RISC [microprocessor]s can outperform DMACs.

COPYRIGHT Cahners Publishing Co. 1989

Nixdorf opts for 16-bit Z280 version of Z80 at low, Unix at high end for retail line. (Nixdorf Computer AG)

Computergram International ISSUE: n1237 PAGINATION: CGI08080007
PUBLICATION DATE: August 8, 1989 LINE COUNT: 00052
SOURCE FILE: CD File 275
PRODUCT NAME(S): Nixdorf POS 2000 (Computer system) - product specifications; Nixdorf POS 2000 10 (Computer system) - marketing; Nixdorf POS 2000 20 (Computer system) - marketing; Nixdorf POS 2000 30 (Computer system) - marketing; Nixdorf POS 2000 40 (Computer system) - marketing
COMPANY NAME(S): Nixdorf Computer AG - marketing
DESCRIPTORS: Marketing Strategy; POS Terminals; Retailing; UNIX
SIC CODE: 3571

NIXDORF OPTS FOR 16-BIT Z280 VERSION OF Z80 AT LOW,

UNIX AT HIGH END FOR RETAIL LINE

Nixdorf Computer Ltd's Retail Division, which launched its new range of point of sale terminals last week (CI No 1,230), looks to the new line to buy it a stronger market penetration of the department store, garage forecourt and hotel and leisure sectors, while retaining its hold on the food multiples, non-food multiples and cash receipting sectors. The technological key to further expansion lies with its new POS 2000 family. Nixdorf's reputation in retail has hitherto been built upon its modular range of terminals, the 8812 family which was launched in 1977 and which can be configured to suit different retail environments. To date, the division has contracts with such illustrious UK high street names as Bally Shoes, Currys, Dixons, Toys 'R' Us, Peacock's Stores and SavaCentre as well as Laura Ashley. In fact this last name was taken by Nixdorf as of great significance in terms of hype for the single European Market, since it has operated on a Euro-pean scale for many years. In an attempt to steal a march on companies such as ICL Data-checker (CI No 1,227) and British Coal's Compower (CI No 1,223) Nixdorf has moved towards Unix with its new POS 200 range.

Uniform designs

Each of the four products in the range is based on a uniform design which comprises a printer, keyboard and customer display, with the central processor housed in an electronic box under the printer. The low-end model, the 2000/10, which is available now, uses a Zilog Z280 16-bit processor with a main memory of 1Mb and is compatible with Nixdorf's 8812 range, offering an upgrade gateway to the POS 2000 family via the exchange of a central processor card. Such terminals are normally backed by a personal computer but the 2000/20 and 2000/30 terminals integrate front and back office applications through their AT-compatible architectures which drive POS 2000 peripherals. The /20 version is an 80286 single user MS-DOS systemc

enabling the user to hot key between the POS application and any back office personal computer application. The /30 80386 multiuser version runs Unix and supports further back office terminals and a second POS terminal. Both these products will ship by the end of the year. Finally, t/range's big boy is the 2000/40 which is designed to be both a POS terminal and controller. The /40 is built around Intel's 80386X chip, has up to 4Mb of memory and interfaces with RS232, RS485, as well as OCIA for scanners, SCSI for mass storage peripherals, an>andard Nixdorf interfaces. It can be equiped with a 3.5" floppy and 3.5" hard disk, s}porting operating systems such as Flexos and Unix. This model will be ready to ship some time in 1990. So how do these terminals fit in with Nixdorf's marketing drive? Well, the Retail Division is particularly targeting the /10 and /40 models at department stores which it believes will find the full screen capability this configuration affords useful for train-ing and for sales pushes; the /10 and /30 terminals offering open systems compatibility and connectivity to a variety of peripherals will, it is hoped wow, garage forecourt managers; while the /10 and /20 versions offered as stand-alones with the option of a special catering keyboard are intended to hook the hotel, and leisure sectors.

COPYRIGHT Apt Data Services 1989

HARDWARE * SOFTWARE
COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80
ERFAHRUNGSAUSTAUSCH

CLUBINFO

42. Ausgabe

Kontaktadresse:
Club 80
Hartmut Obermann
Mozartring 23
Postfach 1430
8870 Günzburg
Tel.: 08221/30248
BTX: 08221/30248
Anrufbeantworter &
FAX: 08221/33575

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Neues vom Vorstand 1
Neue FIDO-Nummer 2
BTX-Anschluß 2
Hartmut Obermann
Traueranzeige 2
Termine 3
Jens Neueder

Software

Grafik des GIIIs unter Holte CP/M+ 4 - 16
Volker Dose, Egbert Schröer
Das PCX-Format 17 - 22
Universelle Textersetzung 23 - 28
Alexander Schmid
Oberflächenmenü 29
Artikel aus ...
Compilerbau - Methoden und Werkzeuge 30 - 38
Gerald Schröder
ZCPR - eine Allzweckwaffe 39 - 42
Günther W. Braun

Club 80 Börse

Suche ... 43 - 44
Jörg Brans

Sonstiges

Deskjet News 45
Alexander Schmid
MODEM umsonst !!!??? 46
Hartmut Obermann

Die letzten Seiten

Impressum 47
Schluß 48
Redaktion
Mitgliederadressenliste am INFO-Ende
Geburtstagsliste am INFO-Ende
Infoform am INFO-Ende
Computermessen 94 am INFO-Ende

Neues vom Vorstand

Wie immer möchte ich mich am Anfang des Infos kurz zu Wort melden. Diesmal mit einem sehr, leider aber auch mit einem weniger erfreulichen Thema. Fangen wir mit letzterem an.

I. Südländertreffen/Jahreshauptversammlung des CLUB 80
Wie schon in den vergangenen Jahren, würde ich auch zum Jahresanfang 1994 gerne wieder ein Regional-Treffen Süd hier im Raum Ulm organisieren. Leider wird mir persönlich das diesmal nicht möglich sein. Da ich mit Wirkung vom 01.04.94 nach Karlsruhe versetzt werde und bis dahin beruflich sehr stark eingespannt bin (mal ganz abgesehen von den Umzugsvorbereitungen), sehe ich mich außerstande, die Organisation des Südländertreffens zu übernehmen. Fraglich ist ebenfalls noch, ob ich 1994 in der Lage sein werde eine Jahreshauptversammlung zu veranstalten.

Gesucht wird also ein CLUB 80-Mitglied, das willens und in der Lage ist, ein Regionaltreffen Süd und/oder eine Jahreshauptversammlung zu organisieren. Dabei kommt es vor allem darauf an, eine Unterkunft passender Größe (Unterbringungsmöglichkeit für max. 10 - 15 Teilnehmer) ausfindig zu machen, die noch dazu einen Nebenraum für die gleiche Anzahl Rechner zur Verfügung stellen kann. Das ganze natürlich zu einem vernünftigen Preis und in einer möglichst günstigen Verkehrslage.

"Das klingt schwerer als es in Wirlichkeit ist" sagt meine Frau, die in den letzten Jahren für diese Dinge zuständig war. Wie die Teilnehmer der letzten Treffen bestätigen können, war ich ja nur auf dem Papier der Organisator des Treffens und maximal für die Tagesordnung zuständig.

Sollte sich ein CLUB 80-Mitglied, oder seine bessere Hälfte, in der Lage sehen und noch dazu willens sein, ein Treffen zu organisieren, so möge er sich bitte bei mir melden. Wenn möglich noch vor dem nächsten Redaktionsschluß, damit im Info 43 eventuell schon Näheres veröffentlicht werden kann.

II. Clubinfo
In diesem Info erscheinen unter anderem ein paar Artikel, die auf Anregungen aus der Fragebogenaktion, früheren Infos und Gesprächen mit Mitgliedern entstanden sind. Stellvertretend genannt sei z.B. Alexander Schmid, der eine Fortsetzung des PCX-Programms für CP/M-Rechner und eine Lösung für das ASCII-Konvertierungsproblem zwischen verschiedenen Rechnertypen beisteuert. Oder Gerald Schröder, der seine Diplomarbeit etwas umgearbeitet hat und versucht, uns in die Geheimnise von Compilern einzuweihen.

Beiden möchte ich an dieser Stelle recht herzlich dafür danken, daß sie auf meine mehr oder weniger direkten "Arbeitsaufträge" so prompt reagieren und ganz wesentlich zum Informationsgehalt der Infos beitragen! Gedankt sei natürlich auch allen anderen, hier nicht namentlich genannten Infoschreibern, die sich die Zeit nehmen uns ihre EDV-Kenntnisse näher zu bringen.

In diesem Sinne wünsche ich euch allen ein frohes Weihnachtsfest und ein gesundes, gutes und erfolgreiches Jahr 1994,

Neue FIDO-Nummer

Wie schon im letzten Info berichtet, gibt es nach der Zwangsregionalisierung zwei FIDO-Netze. Das neue, zwangsregionalisierte wird als FIDO-Light bezeichnet, während die Konter-revolutionäre ihr Netz unter dem Namen FIDO-Classic betreiben.

Der SysOp der BRAINSTROM (ehemals 2:241/7902), die jahrelang meine Heimatbox war, hat sich nun leider dazu entschlossen, ganz aus dem FIDO-Netz auszutreten. Dadurch habe ich die Mailbox und damit auch die FIDO-Nummer gewechselt.

Ab sofort bin ich in der SYN-REL-Box (Tel.: 08282-4311) oder unter der FIDO-Nummer 2:241/7922.10 zu erreichen.

BTX-Anschluß

Durch die Probebestellung des kostenlosen 1&1-Modems ist der CLUB 80 jetzt auch per BTX zu erreichen. Leider ist eine Anmeldung im BTX nur unter dem Namen möglich, auf den auch der verwendete Telefonanschluß eingetragen ist. Aus diesem Grund ist der CLUB 80 unter Hartmut Obermann auf dem Anschluß **0822130248** zu erreichen.

Da ich allerdings BTX nicht regelmäßig nutze, kann es passieren, daß mich eine Nachricht längere Zeit nicht erreicht!

Wir trauern um einen guten
Freund und unser Clubmitglied

Heinrich Betz
* 17. 01. 1943 † 05. 12. 1993

91353 Hausen,
St. Wolfgangstraße 13

In Liebe und Dankbarkeit:
Elke Betz mit Sabine und Martin
Hilde Spangenberg, Mutter
Gerhard und Marie Scheunemann
Alfred und Christine Scheunemann, mit Kinder

Die Beerdigung fand am Dienstag, den 7. 12. 1993 auf dem Friedhof in Hausen statt.

Der Vorstand Club80

Termine... Termine... Termine... Termine... Termine

Graphic 1	Kassel	13.01.- 15.01.94
HTC Hamburger Computertage	Hamburg	20.01.- 23.01.94
Computerschau	Dortmund	02.02.- 06.02.94
Comdesign	Sinsheim	23.02.- 26.02.94
Cebit	Hannover	16.03.- 23.03.94

Redaktionsschluß für das nächste Clubinfo ist der

01. März 1994.

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

Beachtet die Umstellung der
Postleitzahlen!

Grafik des Genie IIIs unter Holte CP/M+

Volker Dose, Egbert Schröer, November 1993

Im ersten Beitrag zur kleinen Artikelserie über die Ansteuerung der Grafik des Genie IIIs unter CP/M wurde eine Möglichkeit in Assembler vorgestellt. Auch mit Hochsprachen wie Turbo Pascal ist dies möglich, wenn man dazu die BDOS-Funktion 50 nutzt. Die entsprechende Routine ist ebenso im Include GRAFIK.INC implementiert wie alle Funktionen zum Zeichnen/Löschen von Punkten, Linien etc.

```
(* INCLUDE-File GRAFIK.INC
   in das Programm einzufuegen mit

   ($I GRAFIK.INC)

  Es folgen einige Unterroutinen, mit denen die HRG des
  Genie IIIs angesprochen werden kann. Es sind folgende
  Funktionen, bzw Proceduren implementiert:

 Procedure  Gra_On (seite)
 Procedure  Gra_Off(seite)
 Procedure  Gra_Cls(farbe)

 Procedure  Set_Point(xKoor,yKoor,seite)
 Procedure  Res_Point(xKoor,yKoor,seite)
 Function   Ask_Point(xKoor,yKoor,seite)
 Procedure  Hrg_Line(xStart,yStart,xEnd,yEnd,farbe,seite)

 Procedure  Set_Point_Normiert(xkoor,ykoor,seite)
 Procedure  Res_Point_Normiert(xkoor,ykoor,seite)
 Procedure  Hrg_Line_Normiert(xStart,yStart,xEnd,yEnd,farbe,seite)
 Procedure  Hrg_Circle(xKoor,yKoor,radius,farbe,seite)

  Bei den nicht normierten Proceduren gilt

 xKoor, xStart und xEnd  :  INTEGER im Bereich von 0-639
 yKoor, yStart und yEnd  :  INTEGER im Bereich von 0-275
                         :  (25 Zeilen * 11 Scanlinien)
 seite                   :  BYTE im Bereich 0-1
 farbe                   :  Byte 0 = schwarz   1 = weiss

  Bei den normierten Proceduren gilt abweichend

 yKoor, yStart und yEnd  INTEGER im Bereich von 0-449


!  ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG  !
!    Es wird nicht ueberprueft, ob die uebergebenen Parameter !
!    im oben angegebenen Breich liegen. Eventuell kann der    !
!    Rechner bei einigen Proceduren abstuerzen.               !
!  ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG  !

*)
(* ------------------------------------------- *)
(* Direkter BIOS-Aufruf ueber BDOS-Funktion 50 *)
(* geklaut aus CPM-IBM.PAS aus der c't         *)
```

```
function UBIOS(fn,pa,pbc,pde,phl:integer):integer;
var biospb : record
                 func,a    : byte;
                 bc,de,hl  : integer;
                 end;
    result : integer;
begin
  with biospb do begin
    func:=fn; a:=pa;
    bc:=pbc; de:=pde; hl:=phl;
    end;

      result:=BDOS(50,addr(biospb));

    ubios:=result;
  end;

(* Schaltet eine Grafikseite ein *)

Procedure Gra_On (seite : byte);
var  sys0,sys1   :    byte;
begin
  sys0 := portÄ249Ü;
   if (seite = 0) then   portÄ249Ü := sys0 and 239
                  else   portÄ249Ü := sys0 or   16;
  sys1 := portÄ250Ü;
  portÄ250Ü := sys1 or 2;
end;

(* ------------------------------------------------------------ *)

(* Schaltet eine Grafikseite aus *)

Procedure Gra_Off (seite : byte);
var  nixwert   :    integer;
begin
nixwert:=ubios(30,0,(seite*256)+25,0,0);
end;
(* -------------------------------------------------------------*)

(* Fuellt die Grafikseite mit FARBE *)

procedure Gra_Cls(seite,farbe : byte);
var  nixwert   :    integer;
begin
nixwert := UBIOS(30,farbe,(seite*256)+26,0,0);
end;
(* ------------------------------------------------------------- *)

(* Setzt einen nicht normierten Punkt auf SEITE *)

Procedure Set_Point(xkoor,ykoor : integer; seite : byte);
var  nixwert   :    integer;
begin
nixwert := ubios(30,1,(seite*256)+27,xkoor,ykoor);
end;
(* ------------------------------------------------------------- *)
```



```
(* Loescht einen nicht normierten Punkt auf SEITE *)

Procedure Res_Point(xkoor,ykoor : integer; seite : byte);
var nixwert    :    integer;
begin
nixwert := UBIOS(30,0,(seite*256)+27,xkoor,ykoor);
end;
(* ------------------------------------------------------------------- *)

(* Gibt 1 zurueck wenn der nicht normierte Punkt gesetzt ist *)

Function Ask_Point(xkoor,ykoor : integer;seite : byte) :  integer;
begin
ask_point := UBIOS(30,0,(seite*256)+28,xkoor,ykoor);
end;
(* ------------------------------------------------------------------- *)

(* Zieht eine Linie auf SEITE mit FARBE *)

Procedure Hrg_Line(xstart,ystart,xend,yend : integer; farbe,seite : byte

var  parablock : arrayÄ0..4Ü of integer;
     nixwert,z :                integer;

begin
parablockÄ1Ü := xstart;
parablockÄ2Ü := ystart;
parablockÄ3Ü := xend;
parablockÄ4Ü := yend;
z := addr(parablockÄ1Ü);
nixwert := UBIOS(30,farbe,(seite*256)+29,0,z);
end;
(* ------------------------------------------------------------------- *)

(* Setzt normierten Punkt auf SEITE *)

Procedure Set_Point_Normiert(xkoor,ykoor : integer; seite : byte);
var  nixwert   :    integer;
begin
nixwert := ubios(30,17,(seite*256)+27,xkoor,ykoor);
end;
(* ------------------------------------------------------------------- *)

(* Loescht normierten Punkt auf SEITE *)

Procedure Res_Point_Normiert(xkoor,ykoor : integer; seite : byte);
var  nixwert   :    integer;
begin
nixwert := UBIOS(30,16,(seite*256)+27,xkoor,ykoor);
end;
(* ------------------------------------------------------------------- *)

(* Zeichnet normierte Linie mit FARBE auf SEITE *)

Procedure Hrg_Line_Normiert(xstart,ystart,
                      xend,yend : integer; farbe,seite :byte);
var  parablock : arrayÄ0..4Ü of integer;
     nixwert,z :                integer;
begin
```

```
parablockÄ1Ü := xstart;
parablockÄ2Ü := ystart;
parablockÄ3Ü := xend;
parablockÄ4Ü := yend;
z := addr(parablockÄ1Ü);
nixwert := UBIOS(30,16+farbe,(seite*256)+29,0,z);
end;
(* ------------------------------------------------------------ *)

(* Zeichnet einen Kreis auf SEITE mit FARBE *)

Procedure Hrg_Circle(xkoor,ykoor,radius : integer; farbe,seite : byte);
var parablock : arrayÄ1..3Ü of integer;
    nixwert,z :                integer;
begin
parablockÄ1Ü := xkoor;
parablockÄ2Ü := ykoor;
parablockÄ3Ü := radius;
z := addr(parablockÄ1Ü);
nixwert := UBIOS(30,16+farbe,(seite*256)+30,0,z);
end;
(* ------------------------------------------------------------ *)
```

Mit diesem Include File schafft man sich die Voraussetzungen zur weiteren Programmierung. Nicht implementiert ist eine Funktion, die eigentlich beim Genie IIIs unter Holte CP/M nie genutzt wurde - die Window-Funktion. Window Technik unter CP/M ist ja möglich (es sei nur an die CHIP Turbo Pascal Hefte erinnert), aber Speicherintensiv und mühselig zu programmieren. Als Genie IIIs Besitzer muß man sich darum nicht kümmern, denn eine Window Funktion ist im BIOS implemtiert. Wie das unter Turbo Pascal aussieht zeigt der Include File WINDOWG.BIB.

```
ä ============================================================ ü
ä               WINDOWG.BIB                                    ü
ä Bibliotheks-Modul fuer Windows unter Holte CP/M+             ü
ä zusaetzlich wird, wie bei WINDOW(7).BIB die Routine WINDOW.PAR ü
ä benoetigt. Es wurde versucht weitgehend die Syntax von       ü
ä WINDOW.BIB einzuhalten.                                      ü
ä ============================================================ ü
ä Egbert Schroeer u. Volker Dose Januar bis Juni 1993          ü
ä ============================================================ ü

var     anystring          :     stringÄ255Ü;
        buffer             :     arrayÄ0..2000Ü of byte;
        TempScr            :     arrayÄ0..2000Ü of byte;
        i,j                :     byte;
        a1,b1,a2,b2,
        WindowBreite,
        WindowHoehe        : integer;


(* Direkter BIOS-Aufruf ueber BDOS-Funktion 50 *)
(* wurde hier auch definiert wenn GRAFIK.INC   *)
(* nicht benötigt                              *)

function UBIOS(fn,pa,pbc,pde,phl:integer):integer;
var biospb : record
             func,a   : byte;
             bc,de,hl : integer;
             end;
    result : integer;
begin
  with biospb do begin
    func:=fn; a:=pa;
    bc:=pbc; de:=pde; hl:=phl;
    end;
  result:=0;
  case fn of
    2,3,7,13..15,17..19,24 : result:=BDOS(50,addr(biospb));
    9,16,20,22,25          : result:=BDOSHL(50,addr(biospb));
    else                     BDOS(50,addr(biospb));
    end;
  ubios:=result;
  end;

procedure Save_Screen(zeiger_buffer : integer);
var   nixwert : integer;
begin
nixwert := ubios(30,0,22,0,zeiger_buffer);
end;

procedure Restore_Screen(zeiger_buffer : integer);
var nixwert : integer;
begin
  nixwert := ubios(30,1,22,0,zeiger_buffer);
end;

procedure Reduzier_Window;

begin
  a1:=a1+1;
  b1:=b1-1;
  a2:=a2+1;
  b2:=b2-1;
end;

(*****************************************************************)
(*        OpenWindow - Oeffnen eines Fensters                   *)
(*****************************************************************)
ä Procedure  OpenWindow
  eroeffnet auf dem Bildschirm einen Bereich mit neu definierten
  Bildfenstergroessen. Alle Bildschirmausgaben nach Aufruf
  dieser Funktion beziehen sich nur noch auf die neuen Bildschirm-
  groessen.                                                     ü

procedure OpenWindow (Window_Number,top_line,bottom_line,
                      left_column,right_column : byte);

const   window_char  =     'F';   ä Die uebergebenen Parameter ü
        set_top      =     'I';   ä muessen mit 32 addiert     ü
        set_bottom   =     'J';   ä werden, steht so im HOLTE  ü
        set_left     =     'K';   ä Handbuch CPM3.DOC.         ü
        set_right    =     'L';   ä Durch die Definition der   ü
        escape       =     #27;   ä Konstanten innerhalb der   ü
                                  ä Prozedur kann sie einfach  ü
                                  ä 'so' mit $I eingebunden    ü
                                  ä werden                     ü
```

```
begin
  a1 := top_line;           ä zunaechst Daten sichern zur
                            Initialisierung des eigentlichen Fensters ü

  b1 := bottom_line;
  a2 := left_column;
  b2 := right_column;

  Cursor_Off;                   ä der stoert jetzt nur ü

  ä Breite und Hoehe des Fensters ermitteln          ü
  ä Davon 2 Zeichen fuer LinksOben, RechtsOben usw.  ü
  ä abziehen                                         ü

  WindowBreite := right_column - left_column-1;
  WindowHoehe  := bottom_line - top_line-1;

  ä Fenster fuer Rahmen initialisieren ü

  write(escape,window_char,chr(Window_Number+32));
  write(escape,set_top,chr(top_line+32));
  write(escape,set_bottom,chr(bottom_line+32));
  write(escape,set_left,chr(left_column+32));
  write(escape,set_right,chr(right_column+32));
  ClrScr;       ä Fenster sauber machen  ü

  ä Rahmen zeichnen          ü

  write(LinksOben);
  for i:=1 to WindowBreite do write(WaagrechtO);
  writeln(RechtsOben);
  for j := 1 to WindowHoehe do
    begin
      write(SenkrechtL);
      gotoxy(WindowBreite+2,j+1);
      writeln(SenkrechtR);
    end;
  write(LinksUnten);
  for i:=1 to WindowBreite do write(WaagrechtU);
  write(RechtsUnten);

ä Und nun die Fenstergroesse reduzieren um den
  Rahmen zu erhalten ü

  Reduzier_Window;
  write(escape,window_char,chr(Window_Number+32));
  write(escape,set_top,chr(a1+32));
  write(escape,set_bottom,chr(b1+32));
  write(escape,set_left,chr(a2+32));
  write(escape,set_right,chr(b2+32));
  ClrScr;
  Cursor_On;        ä den brauchen wir jetzt wieder ü
end;

(*****************************************************************)
(*     ExitWindows - stellt urspruenglichen Bildschirminhalt her *)
(*****************************************************************)

procedure ExitWindow;
const    window_char  =       'F';    ä Die uebergebenen Parameter ü
```



```
         set_top      =       'I';    ä muessen mit 32 addiert      ü
         set_bottom   =       'J';    ä werden, steht so im HOLTE   ü
         set_left     =       'K';    ä Handbuch CPM3.DOC.          ü
         set_right    =       'L';    ä Durch die Definition der    ü
         escape       =       #27;    ä Konstanten innerhalb der    ü
                                      ä Prozedur kann sie einfach   ü
                                      ä 'so' mit $I eingebunden     ü
                                      ä werden                      ü
begin
  write(escape,window_char,chr(0+32));
  write(escape,set_top,chr(0+32));
  write(escape,set_bottom,chr(24+32));
  write(escape,set_left,chr(0+32));
  write(escape,set_right,chr(80+32));
  clrscr; (* alles sauber hinterlassen *)
  Restore_Screen(addr(buffer)); (* urspr. Anzeige restaurieren *)
end;
```

Nun besitzen wir zwei Include Files, die es uns ermöglichen alle im Holte BIOS implementierten Grafik-Funktionen auszunutzen. Als erstes Beispiel das Programm GRAFIK.PAS, das einfach einige Aktionen mit der Grafik durchführt:

```
program grafik;

var    fn,pa,pbc,pde,phl    :          integer;
       xkoor,ykoor          :          integer;
       dummi                :          integer;
       sys0,sys1            :             byte;
       radius,i             :          integer;
       taste                :             char;
       hugo                 :          boolean;

ä$I c5:grafik.incü

procedure raumschiff(xkoor,ykoor : integer; farbe,seite : byte);
begin
  hrg_line(xkoor   ,ykoor,  xkoor+29,ykoor,  farbe,seite);
  hrg_line(xkoor+4 ,ykoor+1,xkoor+25,ykoor+1,farbe,seite);
  hrg_line(xkoor+4 ,ykoor-1,xkoor+25,ykoor-1,farbe,seite);
  hrg_line(xkoor+13,ykoor-2,xkoor+16,ykoor-2,farbe,seite);
  hrg_line(xkoor+11,ykoor+2,xkoor+12,ykoor+2,farbe,seite);
  hrg_line(xkoor+17,ykoor+2,xkoor+18,ykoor+2,farbe,seite);
  hrg_line(xkoor+12,ykoor+3,xkoor+12,ykoor+3,farbe,seite);
  hrg_line(xkoor+17,ykoor+3,xkoor+17,ykoor+3,farbe,seite);
  hrg_line(xkoor+13,ykoor+4,xkoor+16,ykoor+4,farbe,seite);
end;


begin                   (* Hauptprogramm  *)
clrscr;

writeln('Testprogramm um die HRG einzuschalten ');

gotoxy(1,2);  clreol;
writeln('Jetzt wird die HRG eingeschaltet !');
  gra_cls(0,0);
  gra_cls(1,0);
```

```
  gra_on(0);
repeat until keypressed;


gotoxy(1,2);  clreol;
writeln('Jetzt werden Punkte gesetzt ! Und zwar normiert !');
  set_point_normiert(200,100,0);
  set_point_normiert(201,102,0);
  set_point_normiert(202,100,0);
  set_point_normiert(203,102,0);
  set_point_normiert(204,100,0);
  set_point_normiert(205,102,0);
  set_point_normiert(206,100,0);
  set_point_normiert(207,102,0);
  set_point_normiert(208,100,0);
  set_point_normiert(209,102,0);
repeat until keypressed;


gotoxy(1,2);  clreol;
writeln('Jetzt werden sie wieder gelöscht !');
  res_point_normiert(200,100,0);
  res_point_normiert(201,102,0);
  res_point_normiert(202,100,0);
  res_point_normiert(203,102,0);
  res_point_normiert(204,100,0);
  res_point_normiert(205,102,0);
  res_point_normiert(206,100,0);
  res_point_normiert(207,102,0);
  res_point_normiert(208,100,0);
  res_point_normiert(209,102,0);
repeat until keypressed;

gotoxy(1,2);  clreol;
writeln('Jetzt werden nicht normierte Punkte gesetzt.');
  set_point(638,174,0);
  set_point(639,174,0);
  set_point(638,175,0);
  set_point(639,175,0);

randomize;

for i := 0 to 500 do begin
                       xkoor := random(639);
                       ykoor := random(275);
                       set_point(xkoor,ykoor,0);
                     end;

repeat until keypressed;

gotoxy(1,2);  clreol;
write('Der normierte Punkt (639,174) ist ');

if ask_point(639,174,0)=1 then writeln('gesetzt.')
                          else writeln('nicht gesetzt.');

repeat until keypressed;
```



```
gotoxy(1,2);  clreol;
writeln('Und jetzt wird mit SetPoint eine Linie gezogen !');

for xkoor := 200 to 250 do
  for ykoor := 300 to 350 do set_point_normiert(xkoor,ykoor,0);

repeat until keypressed;

gotoxy(1,2);  clreol;
writeln('Und jetzt Linien mit HrgLine !!        Schneller !!');
  hrg_line_normiert(0,225,639,225,1,0);
  hrg_line_normiert(315,0,315,449,1,0);

  for xkoor:= 200 to 300 do hrg_line_normiert(xkoor,
                                  110,xkoor,210,1,0);

repeat until keypressed;


gotoxy(1,2);  clreol;
write   ('Jetzt noch einige Kreise !!');
writeln ('  Sie werden immer normiert gezeichnet.');
  hrg_circle(50,50,25,1,0);
  hrg_circle(100,50,25,1,0);
  hrg_circle(150,50,25,1,0);
  hrg_circle(50,100,25,1,0);
  hrg_circle(100,100,25,1,0);
  hrg_circle(150,100,25,1,0);
  hrg_circle(320,220,188,1,0);
repeat until keypressed;


gotoxy(1,2);  clreol;
writeln('Jetzt wird Seite 1 eingeschaltet !!');
  gra_cls(1,0);
  gra_on(1);
repeat until keypressed;

gotoxy(1,2);  clreol;
writeln('Wir loeschen den Schirm mit weiss !!');
  gra_cls(1,1);
repeat until keypressed;

gotoxy(1,2);   clreol;
writeln('Es folgen einige Grafikspielereien.');

randomize;

for i := 0 to 4000 do
    begin
      xkoor := 200+random(239);
      ykoor := 100+random(249);
      res_point_normiert(xkoor,ykoor,1);
    end;


for ykoor := 130 to 320 do hrg_line_normiert(230,ykoor,
                                 410,ykoor,1,1);
```

```
radius := 1;
while  radius < 60 do begin
                          radius := radius +3;
                          hrg_circle(160,225,radius,0,1);
                          hrg_circle(480,225,radius,0,1);
                          hrg_circle(320,225,radius,0,1);
                        end;

radius := 64;
while radius >1 do      begin
                          radius := radius  -3;
                          hrg_circle(160,225,radius,1,1);
                          hrg_circle(480,225,radius,1,1);
                          hrg_circle(320,225,radius,1,1);
                        end;


for i:= 1 to 100 do     begin
                          radius := 2 + random(20);
                          xkoor := 23 + random(600);
                          ykoor := 23 + random(410);
                          hrg_circle(xkoor,ykoor,radius,0,1);
                        end;

gra_cls(1,0);

randomize;

for i := 0 to 200 do begin  xkoor := random(639);
                            ykoor := random(274);
                            set_point(xkoor,ykoor,1);
                            end;

xkoor := 200;
ykoor := 100;

for i := 1 to 25 do begin xkoor := xkoor + 15;
                          ykoor := ykoor +  5;
                          raumschiff(xkoor,ykoor,1,1);
                          raumschiff(xkoor,ykoor,0,1);
                          end;

for i:= 1 to 70 do begin  ykoor := ykoor - 3;
                          raumschiff(xkoor,ykoor,1,1);
                          raumschiff(xkoor,ykoor,0,1);
                          end;

raumschiff(xkoor,ykoor,1,1);

gotoxy(1,2);  clreol;
writeln('Raumschiff bewegen mit Ziffernblock, ENDE mit <x>.');

hugo := true;

while ( hugo = true ) do begin
                           repeat until keypressed;
                           read(kbd,taste);
                           case taste of
                           '4' : begin
```



```
                                 raumschiff(xkoor,ykoor,0,1);
                                 xkoor:=xkoor-10;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '6' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 xkoor := xkoor+10;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '8' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 ykoor := ykoor +5;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '2' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 ykoor := ykoor -5;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '7' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 ykoor := ykoor + 5;
                                 xkoor := xkoor -10;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '1' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 ykoor := ykoor - 5;
                                 xkoor := xkoor -10;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '9' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 ykoor := ykoor + 5;
                                 xkoor := xkoor +10;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           '3' : begin
                                 raumschiff(xkoor,ykoor,0,1);
                                 ykoor := ykoor - 5;
                                 xkoor := xkoor +10;
                                 raumschiff(xkoor,ykoor,1,1);
                               end;
                           'x' : hugo := false;
                          end;
                         end;
gra_off(1);
gotoxy(1,2);
clreol;
writeln('Das wars dann.');
end.
```

Die mächtige Window-Funktion kann man beispielsweise im MINITED.PAS einsetzen, einem kleinen Texteditor. Durch den Einsatz der Window Technik bekommt er gleich ein sehr professionelles Aussehen. Auf alle anderen Include-Files wollen wir hier aus Platzgründen nicht eingehen. Es sei da auf die im Programmtext genannte Originalliteratur verwiesen.

```
ä ============================================================= ü
ä Diese Version von MiniTed ist nur fuer Genie 3s unter Holte   ü
ä CP/M 3.0. Herr Holte hat ueber Bios Funktion 30 den Zugriff   ü
ä auf Fenstertechnik eingebaut.                                 ü
ä Um Verbreitung unter Genie 3s Benutzern wird ausdruecklich    ü
ä gebeten.                                                      ü
ä Cursor Position muesste noch gespeichert werden, um nach      ü
ä restore_screen den Cursor an die "richtige" Position zu       ü
ä bringen.                                                      ü
ä Copyright: Die urspruengliche Version von MiniTed wurde in
             CHIP Turbo Pascal Special Ausgabe 9 veroeffentlicht.
             Von mir stammt nur die Erweiterung um die Fenster-
             technik. Ansprung des Bios und erste Gehversuche
             mit Holte CP/M und Window stammen von Volker Dose.
             Egbert Schroeer
             Joachimstrasse 18
             4270 Dorsten                                       ü

PROGRAM MiniTed;

(*$I c5:CPM-80.BIB    *)  ä                                     ü
(*$I c5:WINDOW.PAR    *)  ä hier Modifikationen fuer Zeichensatzü
(*$I c5:WINDOWG.BIB   *)  ä Window fuer Genie 3s                ü
(*$I c5:WINDEFMI.INC  *)  ä Window Definitionen fuer TED        ü
(*$I c5:READCHAR.INC  *)  ä Angepasste Tastatureingabe          ü
(*$I c5:DYNSTR.BIB    *)  ä Dynamische Strings fuer Turbo-Pascalü
(*$I c5:TED-1.INC     *)  ä Deklarationen von TED               ü
(*$I c5:TED-2.INC     *)  ä Der erste Teile von EditText        ü
(*$I c5:CTRLQ.INC     *)  ä Das Ctrl-Q-Menue                    ü
(*$I c5:CTRLK.INC     *)  ä Das Ctrl-K-Menue                    ü
(*$I c5:TED-3.INC     *)  ä Der zweite Teil von EditText        ü

procedure Logo;
  begin
      save_screen(addr(buffer));
      ClrScr; Write  (´    MINITED    fuer Genie 3s´);
              Writeln(´  Version 1.0 - ´#152´ ES (Juni 1993) ´);
  end;

(* Hauptproramm *)

VAR T : TextListe;
    f : Text;
    X, Y, Z : INTEGER;
    c : char;
    s : stringÄ080Ü;

BEGIN (* MiniTed *)
  CBreak:=FALSE;
  IF ParamCount=0 THEN
    BEGIN
      Logo;
      SelWindow(1);
      ClrScr;
      Write(´ Editfile : ´); readln(s);
      assign(f,s);
      SelWindow(2);gotoxy(1,1); (* Textbereich oeffnen *)
      ClrScr; Cursor_On;
    end
```



```
    else assign(f,ParamStr(1));
    (*$I-*) reset(f); (*$I+*)
    IF IOResult=0 THEN BEGIN close(f);
                          LiesText(f,T) end ELSE NeuerText(T);
    X:=1; Y:=1; Z:=1; ClrScr; EditText(T,X,Y,Z,79,25,TRUE);
    SelWindow(1);
    gotoxy(1,1); Write(´Speichern ? (J/N) ´); clreol;
    repeat read(kbd,c); c:=upcase(c) until c in Ä´J´,´N´Ü;
    write(c);
    if c=´J´ then SchreibText(f,T);
    LoescheText(T); ClrScr;
    ExitWindow;
    Cursor_On
end. (* MiniTed *)
```

Im nächsten Teil werden wir ein Beispiel in der Sprache ANSI C vorstellen. Hier hat Alexander Schmidt - wie man an seinem PCX-Reader im letzten Info sehen konnte - einiges an Vorarbeit geleistet.

Das PCX-Format

Hier ist also nun die angedrohte Fortsetzung des Artikels über den PCX-Reader. Falls jemand eigene Versuche mit dem PCX-Format machen oder die Programm in andere Sprachen umsetzen will, möchte ich noch etwas näher auf die Codierung und auf den Header eingehen.

Die PCX-Codierung

Da es auch bei den DOSen Speicherplatz nicht zum Nulltarif gibt, speichert man die Bilder nicht 1:1 ab, sondern komprimiert sie. Die Codierung ist eine sog. Lauflängencodierung, oder Neudeutsch Run Length Coding, und funktioniert so, daß statt einer langen Folge identischer Bytes nur noch ein Zählbyte und ein Datenbyte geschrieben wird. Am besten arbeitet das Verfahren, wenn viele gleichartige Bytes hintereinander kommen, was bei Grafiken mit großen einfarbigen Flächen ja meist gegeben ist. Ein Beispiel:

45 45 45 45 45 45 45 45 07 00 00 00 00 00 00 00 00 00 00 00

Würde codiert ergeben

08 45 07 0B 00

Damit man nun Zähler und Datenbytes unterscheiden kann und nicht auch einzelnen Bytes ein Zählbyte spendieren muß, werden beim PCX-Format die Zähler mit gesetztem Bit 6 und 7 markiert, d.h. sie sind immer > C0h und man bekommt

C8 45 07 CB 00

Wer aufgepaßt hat merkt jetzt sicher, daß da was nicht stimmen kann, denn da die Daten beliebig sind, können auch ganz zufällig Bytes mit gesetztem Bit 6 und 7 auftauchen. Deutlicher wird das Problem bei der Bytefolge

FF FF FF FF FF FF FF F0 00 00 00 00 00 00

Nach obiger Anleitung codiert bekommt man

C7 FF F0 C6 00

Was nun ? Ist F0 jetzt ein Zähler, oder nicht ? Aber dann wäre C6 ein Datenbyte und alles käme durcheinander. In so einem Fall stellt man einfach doch noch ein Zählbyte voran und man erhält schließlich

C7 FF C1 F0 C6 00

Im Klartext bedeutet das also 7 mal FF, einmal F0 und 6 mal 00. Hier muß man also tatsächlich ein Byte zusätzlich spendieren, aber das braucht weniger Platz, als wenn man das grundsätzlich so machen und stur Zähler-Datenbyte schreiben würde. Beim Decodieren der Daten geht man nach folgendem Muster vor:

```
                 Byte lesen
                     !
                     !
       ist es kleiner als C0 ?
          !                 !
          ja               nein
          !                 !
     als Daten-      nächstes Byte als
    byte ablegen     Datenbyte lesen und
                     ablegen
```

Wenn also ein Zählbyte erkannt ist, wird das nächste Byte als Datenbyte interpretiert, egal welche Bits gesetzt sind. Nochmal ein paar Bytes zum Üben:

0F C8 FF C1 E0 80 85 C9 00

Die Reihenfolge ist Datenbyte, Zähler, Datenbyte, Zähler, Datenbyte, Datenbyte, Datenbyte, Zähler, Datenbyte und die decodierte Folge lautet

0F FF FF FF FF FF FF FF FF E0 80 85 00 00 00 00 00 00 00 00 00

Der PCX-Header

Der Header im PCX-Format ist immer genau 128 Bytes lang und hat noch genug Freiraum, daß auch zukünftige Erweiterungen darin Platz haben. Die genaue Bedeutung der einzelnen Bytes ist anhand der Struktur im Programm ersichtlich, hier möchte ich nur noch ein paar ergänzende Anmerkungen machen.
Mit PCX kann man auch farbige Bilder codieren, wobei die Farbebenen wie die Folien auf einem Overheadprojektor übereinandergelegt werden. Dann müssen natürlich auch die Informationen für die Farbpalette belegt werden und hier liegt auch ein großer Nachteil des PCX-Formats. Da z.B. EGA und VGA-Karten völlig andere Paletten haben, bekommt man mit Sicherheit einen ganz anderen Farbeindruck wenn man Bilder zwischen zwei solcher Karten austauscht. Na ja, aber das kann uns wohl ziemlich egal sein.

Der PCX-Maker

Hier nun endlich das versprochene Programm, mit dem man auch selber Grafiken im PCX-Format abspeichern kann. Um den Bereich zu markieren, der abgespeichert werden soll, wird dieser mit einem Rechteck eingerahmt, das man über die Tastatur verschieben kann. Mit Kleinbuchstaben kann man die linke bzw. untere Kante verschieben, mit Großbuchstaben die rechte und die obere Kante. Die Bele-

gung ist wie beim WordStar üblich, E nach oben, X nach unten, S nach links und D nach rechts. R, C, A und F machen größere Sprünge in die entsprechehde Richtung. An Grafikbefehlen braucht man die Abfrage eines Punktes und das Zeichnen eines Rechtecks, das man aber auch mit einem selbstgeschriebenen Unterprogramm machen kann. Als "Zeichenfarbe" nimmt man zweckmäßigerweise die zum Invertieren, sonst kommt der Bildschirminhalt ziemlich durcheinander.

```
/*******************************************************************/
/* PUTPCX.C                                                        */
/*                                                                 */
/* Gets pixel data from the screen and writes a PCX format file   */
/* Alexander Schmid 9/93                                           */
/*******************************************************************/

#include <stdio.h>
#include <conio.h>
#include <grafmod2.h>   /* computer specific graphic driver */

struct pcxheader{
       char manufact;     /*  0    Hersteller-ID            */
       char version;      /*  1    Versionsnummer           */
       char encode;       /*  2                             */
       char bpp;          /*  3                             */
       int xmin;          /*  4, 5 Bildgroesse              */
       int ymin;          /*  6, 7      "                   */
       int xmax;          /*  8, 9      "                   */
       int ymax;          /* 10,11      "                   */
       int hres;          /* 12,13 Aufloesung der Hardware  */
       int vres;          /* 14,15      "       "      "    */
       char colormap[48]; /* 16-63                          */
       char reserved;     /* 64    momentan reserviert      */
       char ebene;        /* 65    Zahl der Farbebenen      */
       int bpline;        /* 66,67 Byte pro Zeile je Ebene  */
       int paletinfo;     /* 68,69 Art des Bildes           */
       char blank[58];    /* derzeit keine Informationen    */
       } header;

/*-----------------------------------------------------------------------*/
/*                     write .PCX header into file                       */
/*-----------------------------------------------------------------------*/
writeheader(fp)
FILE *fp;
{
  int  count;
  char *loader;                 /* Zeiger auf ein character-array        */

  loader = (char*) &header;     /* LOADER zeigt auf den Header im Speicher */

  for(count = 0; count < 128; ++count){      /* 128 Bytes schreiben      */
    putc(*loader,fp);
    ++loader;                                /* Zeiger eins weiter       */
    }
}

/*-----------------------------------------------------------------------*/
/*                     read pixel from screen                            */
/*-----------------------------------------------------------------------*/
unsigned char readscreen(x,y)
```



```
int x,y;
{
  return point(x,YMAX-y);
}

/*-----------------------------------------------------------------------*/
/*                     mark area to write                                */
/*-----------------------------------------------------------------------*/
area(x1,y1,x2,y2)
int *x1,*y1,*x2,*y2;
{
  int xa1,ya1,xa2,ya2;
  char ch;

  hires();                                 /* Grafik einblenden       */
  color=2;                                 /* Punkt invertieren       */
  box(*x1,YMAX-*y1,*x2,YMAX-*y2);          /* Ausschnitt markieren    */
  while((ch=getch())!=0x0D){               /* auf ENTER warten        */
    xa1=*x1; ya1=*y1; xa2=*x2; ya2=*y2;    /* Werte merken            */
    switch(ch){
      case 'd': if(*x2<XMAX) ++(*x2);        break;   /* rechter Rand ->  */
      case 'f': if(*x2<=XMAX-10) (*x2)+=10;  break;   /* rechter Rand ->> */
      case 's': if(*x2>*x1+10) --(*x2);      break;   /* rechter Rand <-  */
      case 'a': if(*x2>*x1+20) (*x2)-=10;    break;   /* rechter Rand <<- */
      case 'x': if(*y2<YMAX) ++(*y2);        break;   /* unterer Rand V   */
      case 'c': if(*y2<YMAX-10) (*y2)+=10;   break;   /* unterer Rand VV  */
      case 'e': if(*y2>*y1+10) --(*y2);      break;   /* unterer Rand ^   */
      case 'r': if(*y2>*y1+20) (*y2)-=10;    break;   /* unterer Rand ^^  */

      case 'D': if(*x1<*x2-10) ++(*x1);      break;   /* linker Rand ->   */
      case 'F': if(*x1<*x2-20) (*x1)+=10;    break;   /* linker Rand ->>  */
      case 'S': if(*x1>0) --(*x1);           break;   /* linker Rand <-   */
      case 'A': if(*x1>=10) (*x1)-=10;       break;   /* linker Rand <<-  */
      case 'X': if(*y1<*y2-10) ++(*y1);      break;   /* oberer Rand V    */
      case 'C': if(*y1<*y2-20) (*y1)+=10;    break;   /* oberer Rand VV   */
      case 'E': if(*y1>0) --(*y1);           break;   /* oberer Rand ^    */
      case 'R': if(*y1>10) (*y1)-=10;        break;   /* oberer Rand ^^   */
      }
    box(xa1,YMAX-ya1,xa2,YMAX-ya2);   /* alten Rand loeschen      */
    box(*x1,YMAX-*y1,*x2,YMAX-*y2);   /* neuen Rand ziehen        */
    }
  box(*x1,YMAX-*y1,*x2,YMAX-*y2);     /* Markierung wieder loeschen */
  textmode();                         /* Grafik abschalten        */
  }

/*-----------------------------------------------------------------------*/
/*                     do run length encoding                            */
/*-----------------------------------------------------------------------*/
compress(arr1,bytes,fp)
unsigned char arr1[];
int bytes;
FILE *fp;
{
  unsigned char arr2[150],ch;
  int repeat,ausgabe=0,eingabe=0;

  while(eingabe<bytes){              /* Zahl der Eingangsbytes          */
    ch=arr1[eingabe];                /* Byte aus dem Eingangsarray holen */
    ++eingabe;                       /* Eingabezeiger eins weiter       */
    repeat=1;                        /* Wiederholungszaehler            */
```

```
    while((ch==arr1[eingabe]) && (repeat<0x3F)){  /* Byte wiederholt sich */
      ++repeat; ++eingabe;                        /* Zaehler erhoehen      */
      }
    if(repeat>1){                             /* wenn es Wiederholungen gab  */
      arr2[ausgabe]=(char)repeat + 0xC0;      /* Zaehlbyte zusammensetzen    */
      ++ausgabe;                              /* Ausgabezaehler erhoehen     */
      }
    else{
      if((ch & 0xC0) == 0xC0){          /* Datenbyte mit gesetztem Bit 6 und 7 */
        arr2[ausgabe]=0xC1;             /* Zaehlbyte = 1 schreiben             */
        ++ausgabe;                      /* Ausgabezaehler erhoehen             */
        }
      }
    arr2[ausgabe]=ch;                   /* Byte in Ausgabearray schreiben      */
    ++ausgabe;                          /* Ausgabezaehler erhoehen             */
    }
  for(eingabe=0; eingabe<ausgabe; ++eingabe)  /* Ausgabearray auf Floppy  */
    fputc(arr2[eingabe],fp);                  /* schreiben                */
  }

/*----------------------------------------------------------------------------*/
/*                        write .PCX data                                     */
/*----------------------------------------------------------------------------*/
writepcx(x1,y1,x2,y2,fp)
int x1,y1,x2,y2;
FILE *fp;
{
  int x,y,bits;
  unsigned char array[150],byte,count;

  hires();                               /* Grafik einblenden              */
  for(y=y1; y<=y2; ++y){                 /* vom oberen zum unteren Rand    */
    count=0;                             /* Zaehler fuer den Zeilenpuffer */
    for(x=x1; x<=x2; x+=8){              /* vom linken zum rechten Rand    */
      byte=0;
      for(bits=0; bits<8; ++bits){       /* Byte zusammenbasteln           */
        byte |= readscreen(x+bits,y)<<7-bits;
        }
      array[count]=byte;
      ++count;
      }
    compress(array,(x2-x1+1)/8,fp);      /* Zeile komprimieren und auf     */
    }                                    /* Floppy schreiben               */
  textmode();                            /* Grafik abschalten              */
  }

/*------------------------------ MAIN() --------------------------------------*/

main( argc, argv )
int argc;
char *argv[2];
{
  FILE *fp;
  int x1,y1,x2,y2,xoff=0,yoff=0;
  char ch;

  argc = argc;

  if((fp = fopen(argv[1],"rb")) != NULL){
    printf("ERROR: FILE '%s' DOES ALREADY EXIST.\n",argv[1]);
```



```
    exit(0);
    }

  if((fp = fopen(argv[1],"wb")) == NULL){
    printf("ERROR: CAN NOT OPEN FILE '%s' \n",argv[1]);
    exit(0);
    }

  x1=0; y1=0; x2=200; y2=200;
  do{
    area(&x1,&y1,&x2,&y2);
    printf("Die Koordinaten sind %d/%d %d/%d\n",x1,y1,x2,y2);
    }while((ch=getch())==0x0D);    /* Korrektur mit RETURN moeglich */

  printf("\nX-/Y-Offsets so lassen ? ");
  if(((ch=getch() & 0xDF) == 'J') || (ch=='Y') || (ch==0x0D)){
    header.xmin=x1;
    header.ymin=y1;
    header.xmax=x2;
    header.ymax=y2;
    }
  else{
    printf("\n\n  X-Offset: "); scanf("%d",&xoff);
    printf("  Y-Offset: "); scanf("%d",&yoff);
    header.xmin=xoff;
    header.ymin=yoff;
    header.xmax=x2-x1+xoff;
    header.ymax=y2-y1+yoff;
    }

  header.encode=1;
  header.bpp=1;
  header.hres=XMAX;
  header.vres=YMAX;
  header.ebene=1;
  header.bpline=(x2-x1+1)/8;

  writeheader(fp);
  writepcx(x1,y1,x2,y2,fp);
  fclose(fp);

  exit(0);
  }
/* End of main() */
```

Fehler im PCXSHOW

Leider bemerkt man sowas immer erst, wenn man den Artikel schon abgeschickt hat, aber es ist nur eine Zahl. In der Routine READ-HEADER steht in der FOR-Schleife <127, da muß ein <128 hin. Ansonsten habe ich das Programm noch so geändert, daß man selber festlegen kann, wohin das Bild geladen werden soll. So kann man aus mehreren kleinen Bildern ein großes zusammensetzen. In der Discovery findet sich natürlich die neueste Version.

Alexander Schmid

Universelle Textersetzung

Es hat ja schon viele Programme gegeben, um Textdateien zwischen MSDOSen und CP/M-Rechnern auszutauschen, und besonders WordStar-Texte sind bei DOSen-Besitzern immer wieder gerne gesehen :-) Nachteilig bei solchen Programmen ist aber, daß ihr Funktionsumfang vom Programmierer genau festgelegt ist und sie nur ganz bestimmte, meistens einzelne, Zeichen umcodieren können. So hat man zwangsläufig bald eine größere Sammlung solcher Utilities, über die man schnell den Überblick verliert. Viel universeller ist nun ein Programm, das durch eine Parameterdatei gesteuert wird und so mehr oder weniger beliebige Strings gegen beliebige andere Strings austauschen kann. Hartmut Obermann und ich hatten so ein Programm vor Jahren mal von MSDOS nach CP/M umgesetzt und er hat mich jetzt gebeten, das auch für die vielen neuen Mitglieder (nochmal ?) im Info vorzustellen.

Das Parameterfile ist ein ganz gewöhnliches Textfile mit folgendem Aufbau:

```
--- dies ist ein Kommentar
alter String
neuer String
--- dies ist der nächste Kommentar
dritter String
vierter String
```

Die Stringlänge ist auf 80 Zeichen begrenzt, aber wenn das wirklich nicht reicht, muß man halt auf zweimal ersetzen. Steuerzeichen werden mit einem vorangestellten '^' eingegeben, also z.B. als ^I und um auch die IBM-Sonderzeichen ohne große Verrenkungen zu erreichen, wird bei einem vorangestellten '_' 64 zum Zeichencode des folgenden Zeichens addiert, d.h. das IBM-ä wird durch _D repräsentiert. Damit die IBM-Umlaute und sonstige Sonderzeichen mit Codes über 128 nicht von vornherein kaputtgemacht werden, maskiert GLOBAL das 8. Bit nicht aus, das muß nach wie vor mit einem getrennten Programm erfolgen. Wenn man also einen WordStar-Text nach MSDOS umsetzen will, muß man erst das 8. Bit mit einem anderen Programm löschen und dann GLOBAL aufrufen.

Natürlich gibt es für den Sonderfall DOS <-> CP/M spezielle Programme, die das Maskieren und Umcodieren viel einfacher und in einem Rutsch machen, aber das war hier nur als Beispiel gedacht und es gibt ja auch unter CP/M noch zahlreiche Textverarbeitungen, die teilweise recht seltsame Formate haben und für die es keine speziellen Umsetzungsprogramme gibt, damit man die Texte mit "normalen" Textprogrammen weiterverarbeiten kann.

Eine andere mögliche Anwendung mit durchaus praktischer Bedeutung könnte die Ersetzung von (persönlichen) Kürzeln gegen Text, z.B. "mfg" gegen "Mit freundlichen Grüßen" oder "sgh" gegen "Sehr geehrte Herren" sein. Wenn man solche Phrasen nicht auf Funktionstasten legen kann, kann man sich so einiges an Tipparbeit sparen. Wenn man will, kann man so sogar ganze Briefköpfe erzeugen, indem man z.B. sowas schreibt:


```
--- Brief an Anton
AntonAdr
Herrn^M^JAnton Meier^M^JKleine Str. 12^M^J12345 Posemuckel^M^J
```

Rekursionen sollte man dabei tunlichst vermeiden (darum steht oben AntonAdr und nicht Anton), sonst erhält man zwar sehr interessante, aber sicher nicht die gewünschten Ergebnisse.

```
--- vorsicht !
Anton
Hallo Anton, wie geht's ?
```

Das mit den IBM-Umlauten habe ich erst nachträglich eingefügt und in der Discovery liegt noch eine ältere Version, aber die Änderung in der Prozedur "InterpretCC" ist minimal und kann sehr einfach durch Kopieren des vorherigen Blocks und etwas editieren gemacht werden. Wenn wir schon bei den Änderungen sind, wie ich gerade gemerkt habe, hat die Version in der Discovery auch noch einen Bug, der verhindert, daß das weiter oben mit dem Briefkopf richtig funktioniert. Der Bug ist auch in "InterpretCC", sodaß Ihr die Patches sehr leicht machen könnt, wenn Ihr Euch das File aus der Discovery holt.

Der Aufruf des Programms erfolgt mit

"GLOBAL eingabefile ausgabefile parameterdatei"

oder

"GLOBAL eingabefile ausgabefile parameterdatei I"

wobei im zweiten Fall noch die IBM-Umlaute in ASCII-Umlaute umgewandelt werden. Im Prinzip ist das überflüssig, weil man es mit einem Parameterfile genausogut machen kann, aber es ist nunmal im Programm drin. Das Ausgabefile darf noch nicht existieren, sonst wird das Programm mit einer Fehlermeldung abgebrochen.

```
PROGRAM Global_Replace;

(* mit Umsetzung der IBM-Umlaute in ASCII-Umlaute *)

(*$I bytefile.bib *)

CONST Version='Version 1.1.2';
      InputLength=80;

TYPE String80=STRING[80];
     String255=STRING[255];
     T_WorkFiles=ByteFile;   (* statt FILE OF BYTE *)
     T_DefsFile=TEXT;
     P_ReplDefs=^T_ReplDefs;
     T_ReplDefs=RECORD
                  OldString:String80;
                  NewString:String80;
```

```
                Next      :P_ReplDefs;
              END;

VAR InputLine,OutputLine:String255;
    InputFile,OutputFile:T_WorkFiles;
    ReplDefsFile        :T_DefsFile;
    ReplDefs            :P_ReplDefs;
    MaxReplString,i     :BYTE;
    IBM                 :BOOLEAN;

PROCEDURE Message;
BEGIN
  WRITELN('GLOBAL REPLACE (',Version,')');
  WRITELN;
  WRITELN('(c) Hans Wiederhold 1986');
  WRITELN
END;

PROCEDURE Oops;
BEGIN
  WRITELN('Aufruf mit "GLOBAL InputFile OutputFile ReplaceDefs [I]" ');
  WRITELN('Ein File mit dem Namen der Output-Datei darf nicht existieren.');
  WRITELN;
  WRITELN('Das ReplaceDefs-File hat folgendes Format:');
  WRITELN;
  WRITELN('--- Kommentar fuer den Benutzer, wird nicht interpretiert');
  WRITELN('zu ersetzender String');
  WRITELN('einzufuegender String');
  WRITELN('--- naechster Kommentar usw.');
  WRITELN;
  WRITELN('Control-Zeichen werden mit ^-Zeichen und ^ mit ^! eingegeben');
  WRITELN;
  WRITELN('Mit dem Parameter I werden IBM-Umlaute in ASCII-Umlaute umcodiert');
  WRITELN;
  HALT
END;

PROCEDURE FilesInit(VAR Input:T_WorkFiles; VAR Output:T_WorkFiles;
                    VAR ReplaceDefsFile:T_DefsFile);
VAR InputFileName,OutputFileName,DefsFileName:String80;
  PROCEDURE UpperCase(VAR WorkString:String80);
  VAR i:BYTE;
  BEGIN
    FOR i:=1 TO length(WorkString) DO
      WorkString[i]:=UpCase(WorkString[i])
  END;
BEGIN
  IF ParamCount<3 THEN Oops;
  IBM:=(ParamStr(4)='I');
  InputFileName:=ParamStr(1);
  (*V-*) UpperCase(InputFileName); (*V+*)
  OutputFileName:=ParamStr(2);
  (*V-*) UpperCase(OutputFileName); (*V+*)
  DefsFileName:=ParamStr(3);
  (*V-*) UpperCase(DefsFileName); (*V+*)

  ASSIGN(ReplaceDefsFile,DefsFileName);
  (*$I-*) RESET(ReplaceDefsFile); (*$I+*)
  IF (IOresult<>0) THEN Oops;
  AssignByteFile(Input,InputFileName);
```



```
  (*$I-*) ResetByteFile(Input); (*$I+*)
  IF (IOresultByteFile<>0) THEN Oops;
  AssignByteFile(Output,OutputFileName);
  (*$I-*) ResetByteFile(Output); (*$I+*)
  IF (IOresultByteFile=0) THEN Oops;
  RewriteByteFile(Output)
END;

PROCEDURE LoadReplacements(VAR ReplaceDefsFile:T_DefsFile;
                           VAR FirstDef:P_ReplDefs; VAR MaxIstLength:BYTE);
VAR LastDef,NewDef:P_ReplDefs;
    Dummy,Ist,Soll:String80;

  PROCEDURE InterpretCC(VAR WorkString:String80);
  VAR CCPos,CCh,i:BYTE;
  BEGIN
    i:=1;
    REPEAT
      CCPos:=POS('^',COPY(WorkString,i,LENGTH(WorkString)-i+1));
      IF CCPos>0 THEN
      BEGIN
        CCh:=ORD(WorkString[CCPos+i]);
        IF CCh=33 THEN CCh:=94 ELSE CCh:=(CCh AND 159);
        WorkString[CCPos+i-1]:=CHR(CCh);
        IF CCh=95 THEN i:=i+CCPos+1 ELSE i:=i+CCPos;
        DELETE(WorkString,i,1);
      END;
    UNTIL CCPos=0;
    REPEAT
      CCPos:=POS('_',COPY(WorkString,i,LENGTH(WorkString)-i+1));
      IF CCPos>0 THEN
      BEGIN
        CCh:=ORD(WorkString[CCPos+i]);
        IF CCh=33 THEN CCh:=95 ELSE CCh:=(CCh + 64);
        WorkString[CCPos+i-1]:=CHR(CCh);
        IF CCh=95 THEN i:=i+CCPos+1 ELSE i:=i+CCPos;
        DELETE(WorkString,i,1);
      END;
    UNTIL CCPos=0;
  END;

BEGIN
  Ist:=''; Soll:='';
  MaxIstLength:=0;
  FirstDef:=NIL;
  WHILE NOT EOF(ReplaceDefsFile) DO
  BEGIN
    READLN(ReplaceDefsFile,Dummy);  (* Kommentar *)
    IF (NOT EOF(ReplaceDefsFile)) THEN READLN(ReplaceDefsFile,Ist);
    IF (NOT EOF(ReplaceDefsFile)) THEN READLN(ReplaceDefsFile,Soll);
    IF ((Ist<>'') AND (Ist<>Soll)) THEN
    BEGIN
      interpretCC(Ist);
      interpretCC(Soll);
      NEW(NewDef);
      NewDef^.OldString:=Ist;
      NewDef^.NewString:=Soll;
      IF (FirstDef=NIL) THEN FirstDef:=NewDef ELSE LastDef^.Next:=NewDef;
      LastDef:=NewDef;
      LastDef^.Next:=NIL
```

```
      END;
      IF LENGTH(Ist)>MaxIstLength THEN MaxIstLength:=LENGTH(Ist)
    END
END;

PROCEDURE GetInputLine(VAR Input:String255; VAR InputFile:T_WorkFiles);
CONST InputLength=80;
VAR Ch:BYTE;
BEGIN
  WHILE ((LENGTH(Input)<InputLength) AND (NOT EofByteFile(InputFile))) DO
  BEGIN
    ReadByteFile(InputFile,Ch);
    IF IBM THEN
      Case Ch OF
        $15 : Ch:=$40; (* '@' *)
        $8E : Ch:=$5B; (* '[' *)
        $99 : Ch:=$5C; (* '\' *)
        $9A : Ch:=$5D; (* ']' *)
        $84 : Ch:=$7B; (* '{' *)
        $94 : Ch:=$7C; (* '|' *)
        $81 : Ch:=$7D; (* '}' *)
        $E1 : Ch:=$7E; (* '~' *)
    END;
    Input:=Concat(Input,CHR(Ch))
  END
END;

PROCEDURE PutOutputLine(VAR Output:String255; VAR OutputFile:T_WorkFiles);
VAR i,Ch:BYTE;
BEGIN
  FOR i:=1 TO LENGTH(Output) DO
  BEGIN
    Ch:=ORD(Output[i]);
    WriteByteFile(OutputFile,Ch)
  END
END;

PROCEDURE DoReplacements(VAR Input:String255; VAR Output:String255;
                         Replacements:P_ReplDefs);
VAR PattPos:BYTE;
BEGIN
  Output:=Input;
  WHILE Replacements<>NIL DO
  BEGIN
    REPEAT
      PattPos:=Pos(Replacements^.OldString,Output);
      IF ((PattPos>0) AND (PattPos<=(LENGTH(Output)-MaxReplString))) THEN
      BEGIN
        DELETE(Output,PattPos,LENGTH(Replacements^.OldString));
        INSERT(Replacements^.NewString,Output,PattPos)
      END;
    UNTIL ((PattPos=0) OR (PattPos>=(LENGTH(Output)-MaxReplString)));
    Replacements:=Replacements^.Next
  END;
  Input:=COPY(Output,LENGTH(Output)-MaxReplString+1,MaxReplString);
  IF NOT EofByteFile(InputFile) THEN
    DELETE(Output,LENGTH(Output)-MaxReplString+1,MaxReplString);
END;
```



```
PROCEDURE FilesClose(VAR Input:T_WorkFiles; VAR Output:T_WorkFiles;
                     VAR ReplaceDefs:T_DefsFile);
BEGIN
  CLOSE(ReplaceDefs);
  CloseByteFile(Input);
  CloseByteFile(Output);
END;

BEGIN
  Message;
  FilesInit(InputFile,OutputFile,ReplDefsFile);
  LoadReplacements(ReplDefsFile,ReplDefs,MaxReplString);
  InputLine:=''; OutputLine:='';
  WHILE NOT EofByteFile(InputFile) DO
  BEGIN
    GetInputLine(InputLine,InputFile);
    DoReplacements(InputLine,OutputLine,ReplDefs);
    PutOutputLine(OutputLine,OutputFile)
  END;
  FilesClose(InputFile,OutputFile,ReplDefsFile)
END.
```

Wer das Listing nicht abtippen will, kann es wie immer aus der Discovery downloaden (GLOBAL.LBR) oder von mir oder Hartmut bekommen.

Alexander Schmid

Oberflächenmenü

Menüdatei unter DOS selbst erstellen

Viele Leser kennen Menüoberflächen, mit denen u.a. einfach und bequem Dateien aufgerufen werden können. Ein speicherplatzsparendes und effektives Menü bieten Ihnen die nachstehenden Dateien.
Zur Erstellung des Oberflächenmenüs gehen Sie wie folgt vor:

1. Schreiben Sie - am besten über einen Editor oder ein Textprogramm - nachstehende Zeilen ab und speichern diese als ASCII-Text unter dem Namen "WEITER.DAT". Die nach INT 21 stehende Leerzeile muß als Leerzeile erfaßt werden.

```
A
MOV AH,07
INT 21
CMP AL,00
JNZ 010A
JMP 0100
MOV AH,4C
INT 21

R CX
10
N WEITER.COM
W
Q
```

2. Starten Sie den DOS-Debugger (ein im DOS-Betriebssystem enthaltenes Programm) mit "debug < weiter.dat" (ohne Anführungszeichen!) gefolgt von einem return. Bei fehlerfreier Arbeit sollte die Datei "WEITER.COM" erzeugt sein. Wenn WEITER.COM später fehlerfrei arbeitet, kann WEITER.DAT gelöscht werden.

3. Schreiben Sie - am besten über einen Editor oder ein Textprogramm - nachstehende Zeilen ab und speichern diese als ASCII-Text unter dem Namen "MENUE.BAT". Die in Großbuchstaben und kursiv geschriebenen Texte können Sie Ihren Verzeichnissen und Programmen entsprechend anpassen.
Die beim "errorlevel" genannten Zahlen entsprechen den Kleinbuchstaben laut ASCII-Tabelle. Diese wiederum korrespondieren in unserem kleinen Beispiel mit den im Textmenue "MASKE.TXT" eingerahmten Buchstaben. Wenn Sie später also "w" für WINDOWS aufrufen, entspricht dies dem errorlevel 119 = w. Durch die dann folgende Anweisung "... goto WINDOWS" wird in den Bereich ":WINDOWS" verzweigt. Dort wird dann die Datei "WIN" aufgerufen. Nach Ende von "WINDOWS" wird wieder zu "haupt" an den Anfang verwiesen und Sie können das nächste Programm aufrufen.
Die Verweise nach haupt1 (z.B. "if errorlevel 121 goto haupt1") wurden erstellt, um irrtümliche Tasteneingaben abzufangen. Wenn also anstelle von "w" ein "q" gedrückt wird, bricht das Programm nicht ab sondern verzweigt an den Anfang und wartet auf eine "richtige" Anweisung. Sie können die abzufangenden Tasteneingaben selbstverständlich erweitern ("if errorlevel .... goto haupt1").

```
@echo off
:haupt
c:
cd\
cls
type c:\menue\maske.txt
:haupt1
c:\menue\weiter.com
:abfrage1
if errorlevel 122 goto DOS
if errorlevel 121 goto haupt1
if errorlevel 120 goto haupt1
if errorlevel 119 goto WINDOWS
if errorlevel 118 goto haupt1
if errorlevel 117 goto haupt1
if errorlevel 116 goto WORD
if errorlevel 115 goto SMART
if errorlevel 114 goto haupt1
if errorlevel 113 goto haupt1
rem weitere errorlevel möglich
if errorlevel 105 goto haupt1
if errorlevel 104 goto HARVARD
if errorlevel 103 goto haupt1
if errorlevel 102 goto haupt1
if errorlevel 101 goto haupt1
if errorlevel 100 goto DBASE
if errorlevel 99 goto haupt1
if errorlevel 98 goto haupt
if errorlevel 97 goto haupt
goto haupt1
:DBASE
c:
cd\DBASE
DBASE.EXE
goto haupt
:WORD
c:
cd\WORD
WORD.EXE
goto haupt
:WINDOWS
c:
cd\WINDOWS
WIN
goto haupt
:HARVARD
c:
cd\HARVARD
HARVARD.EXE
goto haupt
:SMART
c:
cd\SMART
SMART D
goto haupt
:DOS
cd\
cls
```

w	indows	h	arvard
s	mart	d	Base
t	extprogramm	z	urück zu DOS

MASKE.TXT: So kann die Oberfläche Ihrer Menüdatei aussehen. Nach Eingabe eines der eingerahmten Buchstabens erscheint das entsprechende Programm. Sie können diese Textdatei natürlich auch ganz anders gestalten.

4. Schreiben Sie - am besten über einen Editor oder ein Textprogramm - u.a. Grafikkasten ab und speichern ihn als ASCII-Text unter dem Namen "MASKE.TXT".

5. Sie haben nun folgende Dateien geschrieben/erzeugt: **weiter.dat, weiter.com, menue.bat, maske.txt**; legen Sie diese vier Dateien in ein Unterverzeichnis C:\MENUE ("md MENUE").

6. Nehmen Sie sich nun Ihre Datei autoexec.bat vor und fügen dem PATH-Eintrag das Unterverzeichnis C:\MENUE an. (PATH=...;C:\MENUE;). Geben Sie schließlich in Ihrer autoexec.bat als letzte Zeile ein: C:\MENUE\MENUE.BAT. Fertig.

7. Starten Sie Ihren Rechner neu. Bei fehlerfreier Eingabe muß die unter Ziffer 4. beschriebene Maske erscheinen. Sie können nun Ihre gewünschte Datei aufrufen, indem Sie den entsprechenden Buchstaben eingeben. Nach Programmende geht es automatisch wieder in die Maske. Über "Z" wie "zurück zu DOS" gelangen Sie - wenn gewünscht - wieder in Ihr Hauptverzeichnis.

8. Passen Sie die Maske Ihren Bedürfnissen entsprechend an. Sie können über diese Maske natürlich auch Kopier- und Formatierbefehle automatisieren.

(erstellt nach Hinweis von Helmut Bruditz, MatAmtBw)

Compilerbau — Methoden und Werkzeuge

Gerald Schröder, e-mail: gschroed@dbis1.informatik.uni-hamburg.de

Oktober 1993

Der folgende Artikel ist ein (gekürzter) Teil eines Kapitels meiner Diplomarbeit über Syntaxerweiterungen in Programmiersprachen, die ich im August dieses Jahres fertiggestellt habe. Der Artikel knüpft an den Compilerbau-Artikel aus Info 41, S. 37+38, an, der die grundsätzliche Arbeitsweise eines Compilers beschreibt. Ich kenne zwar die dazugehörige Artikelserie nicht, vermute aber, daß dort beschrieben wird, wie ein Compiler *von Hand* gebaut wird, sprich: jede Funktion des Compilers wird in Pascal (oder einer anderen Programmiersprache) ausprogrammiert.

Dieser Artikel soll dagegen eine andere Möglichkeit zeigen: Ein Compiler kann (ganz oder teilweise) mit speziellen Sprachen *beschrieben* werden. Wenn so eine Beschreibung vorliegt, können *Werkzeuge* bzw. *Generatoren* (also bestimmte Programme) automatisch aus diesen Beschreibungen einen Compiler erzeugen (bzw. generieren). Vorteil dieser Methode ist, daß die Beschreibungen und Werkzeuge für die verschiedenen Teile eines Compilers angepaßt sind und eine Fehlerprüfung erlauben, während bei von Hand geschriebenen Compilern beliebige Fehler eingebaut werden können.

Nachteile gibt es natürlich auch: Wer kennt/lernt schon diese speziellen Sprachen? (Eine Programmiersprache kennt dagegen jeder.) Wer hat die Werkzeuge mal eben zur Hand? (Einen Compiler hat dagegen jeder.) Wer möchte schon auf die Tricks und die daraus resultierende Effizienz verzichten, die handgeschriebener Code gegenüber automatisch erzeugtem Code bietet?

Aus meiner Erfahrung kann ich darauf antworten: Ein handgeschriebener Compiler, der auch noch trickreich programmiert ist, strotzt vor Fehlern und ist nicht wartbar. Ich habe das bei einem Modula-2 Compiler von ca. 30.000 Zeilen mitgemacht und war kurz vor dem Mordanschlag (auf meine Vorgänger). Natürlich sehe ich ein, daß nicht jeder die Werkzeuge zur Hand hat (ich habe sie zu Hause auch nicht), aber das sollte uns nicht hindern, wenigstens die Methodik anzuwenden: Ein Compiler, der erst (formal) beschrieben und dann anhand dieser Beschreibungen programmiert wird, ist immer noch wartbarer als ein „gehackter" Compiler.

Last, but not least, sind solche Methoden und Werkzeuge auch nicht nur beim Compilerbau zu gebrauchen: Wer zum Beispiel ein Konvertierungsprogramm (Daten von einem Format in ein anderes übersetzen, also auch eine Art Compiler) geschrieben hat, weiß solche Beschreibungen auch zu schätzen. Natürlich gibt es weitere Anwendungen, aber das überlasse ich Eurer Phantasie.

Ach ja: Wer eine Programmieraufgabe sucht, kann sich ja mal daran versuchen, eines der beschriebenen Werzeuge zu implementieren. Hinweise dazu geben z. B. [Aho 88] oder [Waite 85]. Ich habe selber einen Scanner- und einen Parser-Generator geschrieben, und wenn ich es kann, kann es jeder. Als einführendes Werk in den Bau von *handgeschriebenen* Compilern kann ich [Wirth 86] empfehlen.

1 Compilermodell

Dieser Abschnitt erläutert ein Compilermodell, das den Aufbau eines Compilers beschreibt. Es handelt sich um ein klassisches Phasenmodell nach [Aho 88].

1.1 Grobstruktur eines Compilers

Ein Compiler hat die Aufgabe, einen Quelltext (*source text*) in einen maschinenabhängigen Zielkode (*target code*) zu übersetzen. Diese Aufgabe läßt sich in verschiedene Teilaufgaben zerlegen, die hintereinander ausgeführt werden können. Jede Teilaufgabe transformiert eine Repräsentation des Quelltextes in eine andere, wobei die letzte Repräsentation der Zielkode ist. Die Abarbeitung einer Teilaufgabe wird als Phase bezeichnet.

Dieses Modell beschreibt den *logischen* Aufbau eines Compilers. In einer Implementation können aus Effizienzgründen durchaus mehrere Phasen in einem Lauf (*pass*) zusammengefaßt und verschiedene Repräsentationen in einer Datenstruktur dargestellt werden.

Abbildung 1 zeigt die aus zwei Phasen und drei Repräsentationen bestehende Grobstruktur eines Compilers.

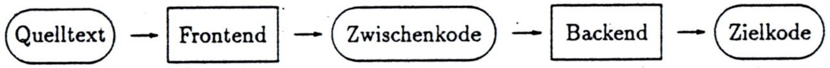


Abbildung 1: Grobstruktur eines Compilers

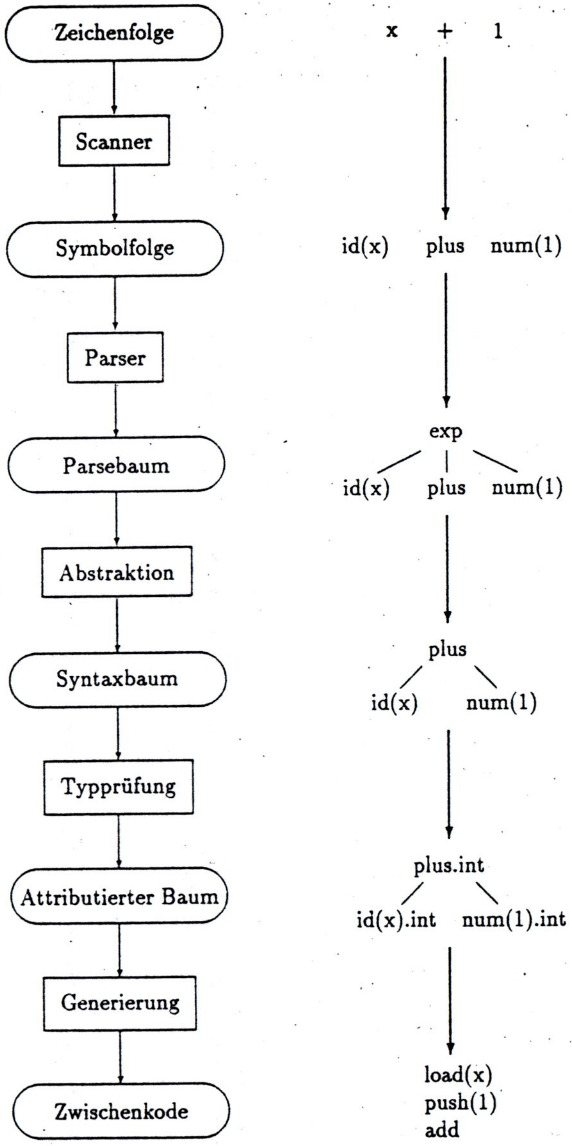
1. Das Frontend analysiert den Quelltext und transformiert ihn in einen maschinenunabhängigen Zwischenkode.
2. Das Backend synthetisiert aus dem Zwischenkode einen maschinenabhängigen Zielkode.

Ich betrachte im folgenden nur das Frontend, weil mich Backends einfach nicht interessieren (bzw. in meiner Diplomarbeit nicht das Thema waren).

1.2 Feinstruktur des Frontends eines Compilers

Das Frontend teilt sich in verschiedene Phasen auf, die im folgenden als Funktionen definiert werden. Abbildung 2 a) zeigt die Phasen und Repräsentationen des Frontends. Abbildung 2 b) enthält ein Beispiel für die Transformation der Datenstrukturen.

scan : *Characters* → *Symbols*
: Die lexikalische Analyse (Scanner) zerlegt den Quelltext, der als Zeichenfolge betrachtet wird, in eine Symbolfolge.



a) Phasen und Repräsentationen b) Beispiel

Abbildung 2: Feinstruktur eines Compiler-Frontends

parse : *Symbols* → *ParseTree*
Die syntaktische Analyse (Parser) analysiert die Symbolfolge auf syntaktische Konstrukte, die als Parsebaum repräsentiert werden.

abstract : *ParseTree* → *AST*, *AST* = *AbstractSyntaxTree*
Durch die Abstraktion wird der Parsebaum in einen abstrakten Syntaxbaum transformiert.

check : *AST* → *AST'*
Die statische semantische Analyse (Typprüfung) überprüft, ob die Konstrukte semantisch korrekt verwendet werden, und fügt semantische Informationen als Attribute in den Syntaxbaum ein.

generate : *AST'* → *IntermediateCode*
Die Zwischenkode-Erzeugung (Generierung) erzeugt aus dem attributierten Syntaxbaum einen Zwischenkode, der vom Backend weiterverarbeitet wird.

Mit den beschriebenen Funktionen und einem Quelltext (*source*) als Eingabe läßt sich der (logische) Aufbau des Compilers funktional folgendermaßen beschreiben:

$$backend(frontend(source)) = backend(generate(check(abstract(parse(scan(source))))))$$

Diese Beschreibungsform betont, daß die einzelnen Phasen (Funktionen) nur vom Ergebnis der vorherigen Phase abhängen.

2 Beschreibungstechniken und Generatoren

Eine Programmiersprache kann durch passende Beschreibungstechniken formal spezifiziert werden, was sich als vorteilhaft für die Kommunikation zwischen Sprachdesignern und Compilerbauern sowie Anwendungsprogrammierern erweist. Generatoren können aus bestimmten Beschreibungen automatisch entsprechende Teile eines Compilers erzeugen. Wenn die Beschreibung eines Frontends vorliegt und entsprechende Generatoren zur Verfügung stehen, kann die Sprache geändert oder erweitert werden, indem die Beschreibung geändert und ein neues Frontend generiert wird, was weitaus einfacher und weniger fehlerträchtig als die Änderung eines handgeschriebenen Compilers ist.

2.1 Lexikalische Analyse

Die Abbildung von Zeichenfolgen auf Symbolfolgen wird durch reguläre Ausdrücke spezifiziert. Jedem Symbol der Sprache wird ein regulärer Ausdruck zugeordnet, der die Repräsentation des Symbols als Zeichenkette beschreibt. Umgekehrt muß jeder Zeichenkette, die in einem Quelltext

auftauchen kann, ein Symbol der Sprache zugeordnet werden. Aus diesen Paaren (regulärer Ausdruck, Symbol) erzeugt ein Scanner-Generator einen endlichen Automaten, der die Abbildung realisiert.

Die folgende Tabelle zeigt einige Beispiele für reguläre Ausdrücke:

"begin"	Schlüsselwort: **begin**
":"	Schlüsselwort: Doppelpunkt
[0–9]+	ganze Zahl: mindestens eine Ziffer
[a–zA–Z][a–zA–Z0–9]*	Bezeichner: Buchstabe gefolgt von Buchstaben und Ziffern

Ein Scanner übernimmt weitere Aufgaben, die sich teilweise nicht durch reguläre Ausdrücke beschreiben lassen, beispielsweise das Entfernen von Leerzeichen, Zeilenumbrüchen und Zeilenvorschüben, Tabulatoren und (evtl. geschachtelten) Kommentaren, oder das Erkennen von speziellen Kommandos (*pragmas*), die den Compilerlauf beeinflußen, aber nicht zur Sprache gehören. Für diese Aufgaben müssen spezielle Beschreibungsformen eingeführt werden, worauf an dieser Stelle nicht näher eingegangen wird, da keine allgemein akzeptierte Beschreibungsform existiert.

Der bekannteste Scanner-Generator ist LEX, dessen Verwendbarkeit allerdings stark an das Betriebssystem UNIX und die Programmiersprache C gebunden ist.

2.2 Syntaktische Analyse

Die in einer Sprache erlaubten Symbolfolgen (die Sprachsyntax) werden durch kontextfreie Grammatiken beschrieben. Aus einer kontextfreien Grammatik erzeugt ein Parser-Generator einen Parser, der die eine Symbolfolge akzeptierenden Produktionen als Parsebaum darstellt.

Die grundlegende Form der kontextfreien Grammatiken beinhaltet folgende Elemente:

Terminal: Ein Symbol aus der Sprache, das vom Scanner erkannt wird, notiert als Zeichenkette oder Name eines im Scanner definierten regulären Ausdrucks.

Die folgende Tabelle zeigt einige Beispiele für Terminale:

"for"	das Schlüsselwort **for**
":"	ein Doppelpunkt, der auch als Schlüsselwort betrachtet wird
int	eine Zahl gemäß der Definition des Scanners
id	ein Bezeichner gemäß der Definition des Scanners

Nonterminal: Name einer Produktion. Beispiele: *ideG, valG.*

Produktion: Benannte Folgen von Terminalen und Nonterminalen, wobei auch leere Folgen möglich sind, die durch das leere Wort ϵ notiert werden. Jede Folge wird **Alternative** genannt.

Das folgende Beispiel zeigt eine Produktion mit vier Alternativen:

valG ::= "nil" | *int* | *id* | *valG* "+" *valG*

Der Name (Nonterminal) der Produktion lautet *valG*. Die akzeptierten Symbolfolgen (Alternativen) sind das Schlüsselwort nil, eine Zahl *int* oder ein Bezeichner *id* gemäß der Definition des Scanners sowie die Folge [*valG*, +, *valG*], wobei das Nonterminal *valG* rekursiv eine von der Produktion *valG* akzeptierte Symbolfolge beschreibt.

Kontextfreie Grammatiken können mehrdeutig sein, d. h. zu einer Symbolfolge lassen sich evtl. mehrere Parsebäume konstruieren, die diese Symbolfolge akzeptieren. Mit den verschiedenen Parsebäumen sind normalerweise auch verschiedene Semantiken verbunden, ein Programm sollte aber *eine* festgelegte Semantik besitzen. Deshalb schränken Parser-Generatoren die Klasse der erlaubten Grammatiken ein [Aho 88, Kap. 4] und/oder benötigen zusätzliche Informationen zur Auflösung von Mehrdeutigkeiten [Aho 88, Kap. 4.8]. Ein Generator kann also eine kontextfreie Grammatik als unzulässig zurückweisen. Verschiedene Generatoren können die gleiche Grammatik als zulässig oder unzulässig betrachten oder verschiedene Zusatzinformationen benötigen.

Zur Konstruktion von Parsebäumen sind zwei Verfahren möglich: Beim Top-Down-Verfahren wird der Parsebaum von der Wurzel (einer bestimmten Startproduktion) zu den Blättern (Symbolen) konstruiert. Bottom-Up-Parser konstruieren den Baum von den Blättern ausgehend zur Wurzel. Zu beiden Verfahren existieren Klassen von Grammatiken, die das Generieren eines deterministischen Parsers erlauben, der nach dem jeweiligen Verfahren den Parsebaum konstruiert. Bei der Entscheidung für eine Klasse von Grammatiken sind zwei sich widersprechende Aspekte zu bedenken: Die Menge der erlaubten Grammatiken ist bei Generatoren, die Bottom-Up-Parser z. B. aus LR-Grammatiken erzeugen, am größten. Nach dem Top-Down-Verfahren arbeitende Parser, die z. B. aus LL-Grammatiken erzeugt werden, sind (meistens) effizienter. Im Sinne der Allgemeinheit ist ein Generator für Bottom-Up-Parser zu empfehlen [Aho 88, S. 234].

Der bekannteste Parser-Generator ist YACC, dessen Verwendbarkeit allerdings stark an das Betriebssystem UNIX und die Programmiersprache C gebunden ist.

2.3 Abstraktion

Abstrakte Syntaxbäume werden durch Konstruktoren beschrieben, die als Parameter die Werte der Unterbäume bzw. Attribute des Knotens erhalten. Verschiedene Konstruktoren werden zu einer Sorte von Knoten zusammengefaßt.

Das folgende Beispiel zeigt Konstruktoren für abstrakte Syntaxbäume einer Sorte *val*:

val ::= *nilNode()*
| *intNode(int)*
| *idNode(id)*
| *addNode(val val)*

Ein Syntaxbaum der Sorte *val* ist entweder ein leerer Knoten *nilNode*, ein Knoten *intNode* mit einer ganzen Zahl *int* als Attribut, ein Knoten *idNode* mit einem Bezeichner *id* oder ein Knoten *addNode* mit zwei Unterbäumen der Sorte *val*.

Die Konstruktion eines abstrakten Syntaxbaums aus dem Parsebaum läßt sich durch semantische Aktionen in den Produktionen der kontextfreien Grammatik darstellen, wie das folgende Beispiel zeigt:

valG ::= "nil" => *nilNode()*
| *int* => *intNode(int)*
| *id* => *idNode(id)*
| $valG_1$ "+" $valG_2$ => *addNode*($valG_1$ $valG_2$)

Durch das Schlüsselwort nil wird ein leerer Knoten *nilNode* erzeugt. Eine Integer-Zahl erzeugt einen Knoten *intNode*, der eine vom Scanner gelieferte Repräsentation *int* des Zahlenwertes als Attribut enthält. Ein Knoten *idNode* wird durch einen Bezeichner *id* erzeugt. Die Addition zweier Werte wird durch den Knoten *addNode* dargestellt, der als Unterbäume die von der Produktion *valG* erzeugten Syntaxbäume enthält.

2.4 Semantische Analyse

2.4.1 Attributierte Grammatiken

Die Analyse der statischen Semantik läßt sich mit attributierten Grammatiken beschreiben [Aho 88, Kap. 5]. Jedem Knoten des Syntaxbaums werden weitere Attribute zugeordnet und Vorschriften zur Berechnung dieser Attribute angegeben.

Das folgende Beispiel zeigt eine attributierte Grammatik mit Typinformationen:

val.env ::= *nilNode()* => *val.type* := *undefined*
| *intNode(int)* => *val.type* := *integer*
| *idNode(id)* => *val.type* := *lookup(val.env id)*
| *addNode*(val_1 val_2) => val_1*.env* := *val.env*;
val_2*.env* := *val.env*;
if val_1*.type* = *integer*
andif val_2*.type* = *integer*
then *val.type* := *integer*
else *val.type* := *undefined*
endif

Jeder Knoten des Syntaxbaums der Sorte *val* erhält zwei Attribute: *env* ist eine Umgebung, in der jedem deklarierten Bezeichner ein Typ zugeordnet wird; *type* ist der Typ eines Knotens mit den möglichen Werten *undefined* und *integer*. Ein Knoten *nilNode* hat den Typ *undefined*, ein Knoten *intNode* den Typ *integer*. Der Typ eines Knotens *idNode* wird durch eine Funktion *lookup()* bestimmt, die den einem Bezeichner *id* in der Umgebung zugeordneten Typ zurückliefert. Der Typ des Knotens *addNode* ist *integer*, wenn beide Unterbäume den Typ *integer* haben, ansonsten

ist der Typ *undefined*. Außerdem könnten noch Fehlermeldungen erzeugt werden, was hier nicht gezeigt wird.

Die Attribute eines Knotens lassen sich in ererbte (*inherited*) und abgeleitete (*derived*) Attribute aufteilen. Ererbte Attribute eines Knotens erhalten ihren Wert durch einen Vorgängerknoten. Dieser Wert ist entweder der Wert eines ererbten Attributs des Vorgängerknotens oder der Wert eines abgeleiteten Attributs eines Knotens, der im Vorgängerknoten parallel zu diesem Knoten steht. Im Beispiel ist *val.env* ein ererbtes Attribut, dessen Wert im Knoten *addNode* an die Nachfolgerknoten weitergereicht wird. Die Werte der abgeleiteten Attribute werden durch die Nachfolgerknoten bestimmt. Im Beispiel ist *val.type* ein abgeleitetes Attribut.

Durch die Berechnungsvorschriften für die Attributwerte ergibt sich ein Abhängigkeitsgraph, der festlegt, wie oft bzw. in welcher Reihenfolge der Baum durchlaufen werden muß, um alle Attributwerte zu berechnen. Wünschenswert ist die Berechnung aller Attributwerte beim einmaligen Durchlauf durch den Baum, z. B. parallel zur syntaktischen Analyse. Dies wird durch durch links-attributierte Grammatiken sichergestellt [Aho 88, Kap. 5.4]. Bei links-attributierten Grammatiken hängt der Wert eines ererbten Attributs nur von den ererbten Attributen der Vorgängerknoten und den abgeleiteten Attributen der im Vorgängerknoten links von diesem Knoten stehenden Knoten ab. Im obigen Beispiel dürfte im Knoten *addNode* das ererbte Attribut $val_2.env$ von dem abgeleiteten Attribut $val_1.type$ abhängig sein, aber nicht umgekehrt $val_1.env$ von $val_2.type$.

Links-attributierte Grammatiken korrespondieren sowohl mit LL-Grammatiken und Generatoren für Top-Down-Parser als auch mit LR-Grammatiken und Generatoren für Bottom-Up-Parser.

2.4.2 Prädikate

Zur Analyse der statischen Semantik können Regeln formuliert werden, die festlegen, welche abstrakten Syntaxbäume wohlgeformt sind. Ein automatischer Beweiser (oder ein Prolog-Interpreter) kann mit Hilfe der Regeln die Wohlgeformtheit eines Syntaxbaums überprüfen.

Das folgende Beispiel zeigt Regeln für wohlgeformte Syntaxbäume:

[IntNode]

$$\frac{}{Env \vdash intNode(int) : Int}$$

[IdNode]

$$\frac{id \notin Env'}{(Env,\ id : Int,\ Env') \vdash idNode(id) : Int}$$

[AddNode]

$$\frac{Env \vdash val_1 : Int \quad Env \vdash val_2 : Int}{Env \vdash addNode(val_1\ val_2) : Int}$$

Für Syntaxbäume *v* der Sorte *val* wird ein Prädikat $v : Int$ definiert. Das Prädikat wird von jedem Knoten *intNode* erfüllt, wie die Regel [IntNode] angibt. Ein Knoten *idNode* erfüllt das Prädikat,

wenn in der Umgebung dem Bezeichner *id* der Typ *Int* zugeordnet ist und der Bezeichner im Rest der Umgebung *Env'* nicht redefiniert wird. Gemäß der Regel [AddNode] erfüllt ein Knoten *addNode* das Prädikat, wenn beide Unterbäume in der gleichen Umgebung *Env* das Prädikat erfüllen.

2.5 Zwischenkode-Erzeugung

Die Erzeugung des Zwischenkodes aus dem Syntaxbaum läßt sich wie die Typprüfung durch attributierte Grammatiken beschreiben [Aho 88, Kap. 8], wie das folgende Beispiel zeigt:

val.env	::=	*nilNode()*	=> *val.code := nop*
	|	*intNode(int)*	=> *val.code := push(int)*
	|	*idNode(id)*	=> *val.code := load(val.env id)*
	|	$addNode(val_1\ val_2)$	=> $val.code := seq(val_1.code\ val_2.code\ add)$

Jeder Knoten des Syntaxbaums erhält ein weiteres Attribut *code*, das die Werte *nop* (keine Operation), *push(int)* (lege die Zahl *int* auf dem Keller ab), *load(env id)* (lege den durch den Bezeichner *id* in der Umgebung *env* benannten Wert auf dem Keller ab), *add* (addiere die beiden obersten Elemente des Kellers und lege das Ergebnis auf dem Keller ab) und $seq(x_i)$ (führe die Operationen x_i nacheinander aus) annehmen kann. Der Knoten *nilNode* erzeugt den Befehl *nop* und der Knoten *intNode* den Befehl *push* mit dem Zahlenwert-Attribut als Argument. Der Knoten *idNode* führt dazu, daß ein durch den Bezeichner *id* benannter Wert auf dem Keller abgelegt wird. Der Knoten *addNode* erzeugt eine Befehlssequenz aus den Befehlen der beiden Unterbäume und dem Befehl *add*.

3 Schluß

Ich habe in diesem Artikel einige (formale) Beschreibungen für Teile eines Compilers vorgestellt. Auch ohne die entsprechenden Werkzeuge sind diese Beschreibungen sinnvoll, um einen Compiler methodisch und gut dokumentiert zu entwickeln. Mit den entsprechenden Werkzeugen wird die Entwicklung eines Compilers einfacher und sicherer. Der resultierende Compiler ist änderungsfreundlicher und wartbarer als sein handgeschriebenes Gegenstück.

Wenn Ihr Fragen, Anregungen, Wünsche für Folge-Artikel o.ä. habt, sagt mir Bescheid!

Literatur

[Aho 88] A. V. Aho, R. Sethi und J. D. Ullman: *Compilerbau*. Addison-Wesley 1988.

[Waite 85] W. M. Waite und G. Goos: *Compiler Construction*. Springer 1985.

[Wirth 86] N. Wirth: *Compilerbau*. Teubner 1986.

ZCPR - eine Allzweckwaffe?

Zwischen Benutzern des Z-Systems und Leuten, die damit noch keine nähere Bekanntschaft gemacht haben, gibt es immer wieder Diskussionen über seine Vor- und Nachteile. So auch in unserer Club-80 Ecke in der Discovery-Mailbox.

Fritz Chwolka stellte dort beispielsweise die provozierende Frage: "Welchen Sinn hat denn ein Z80-Rechner ohne ZCPR ?!" Und Alexander Schmid schrieb: "Wie mir Helmut Jungkunz vor kurzem gesagt hat, hat er die Preise für die selbstinstallierenden ZCPR-Pakete ("NZCOM" für CP/M 2.2 und "Z3PLUS" für CP/M 3) auf jeweils 70 Mark gesenkt. Für diesen Preis dürfte es aber wohl einmalig sein, denn im Preis inbegriffen ist eine kostenlose Versorgung mit Public Domain Software zum Z-System, die momentan über 50 MB umfasst. Nein, ich bekomme keine Prozente, aber ich arbeite selber mit dem Z3PLUS und ich würde mich mit Händen und Füssen wehren, wenn ich es wieder hergeben sollte."

Als langjähriger Besitzer eines Commodore 128 D, der - ausser im Urlaub - keinen Tag ungenutzt herumsteht, schaue ich mich natürlich ständig nach neuen Programmen um. Warum sollte neben Verbesserungen bei Anwender-Programmen nicht auch eine Betriebssystem-Verbesserung möglich sein? ZCPR steht ja für "Z Console Processor Replacement" und soll den standardmässig zum CP/M-System gehörenden "Console Command Processor" (CCP) ersetzen. Ursprünglich lag wohl der Gedanke zugrunde, dass das für den Intel 8080 Prozessor-Chip entwickelte CP/M-Betriebssystem eigentlich mehr könnte, wenn man es an die Zilog Z-80, -180 und -280 Prozessoren der moderneren CP/M-Maschinen anpassen würde, die seit etwa 1980 auf dem Markt erschienen.

Meine genaueren Nachforschungen ergaben dann auch, dass Leute, die von ZCPR so begeistert sind, hauptsächlich mit "aufgebohrten" modernen CP/M-Maschinen arbeiten. Fritz beispielsweise ist stolzer Besitzer eines kleinen Museums mit an die 30 verschiedenen CP/M-Computern, darunter auch den legendären Commodore PET 2001 mit Diskdrive, sowie einem C-128. Sein Arbeitsgerät ist jedoch ein Z-280 Rehdesign mit 60 MB Harddrive, IDE-Interface und Terminalkarte von Dipl.-Ing. Tilmann Reh, Siegen. Ebenso lüftete Alexander - einer der wenigen Profis, die privat noch unter CP/M programmieren - das Geheimnis seiner Anlage etwas, als er mir mitteilte: "Ich habe hier einen Z-280-Rechner mit 12,5 MHz und einer 80 MB Festplatte, und da kann ich fast beliebig viele Utilities zur ständigen Verfügung halten. Und wenn Programme wie der WordStar Overlays nachladen, zuckt bei mir auch beim schnellen Tippen nur kurz der Cursor. Auf einem 2 MHz Z-80 mit 140 KB pro Laufwerk kann man das wohl vergessen." - Wer jetzt übrigens noch rätselt, was Helmut Jungkunz wohl für einen Rechner einsetzt, dem kann schnell geholfen werden: ein CPU 280!

Die Leistungsmerkmale meines C-128 - Z-80 Prozessor mit 4 MHz Taktgeschwindigkeit - sind dagegen also bescheiden, wobei von den 128 KB Hauptspeicher nach Laden des Standard-CP/M-Plus nur noch 58 KB Arbeitsspeicher (TPA) zur Verfügung stehen. Und wo ausser dem eingebauten Laufwerk für zweiseitige 5 1/4"-Disketten (mit 300 - 400 KB) allenfalls weitere 5 1/4"- oder 3 1/2"-Laufwerke extern angeschlossen werden können, oder eine RAM-Disk mit 508 KB.

Derartige Unterschiede in der Hardware bedingen notwenigerweise aber auch Unterschiede in der Software. So benötigt CP/M 3 - obwohl es theoretisch auch für den alten 8080 Prozessor geeignet ist - in der Praxis mehr Speicherplatz und damit eben auch einen schnelleren Prozessor, als das ältere CP/M 2. Und ebenso ist auf den verschiedenen PC-DOSen ja auch nicht in allen Fällen das gleiche Betriebssystem im Einsatz.

Um nicht erst nach erfolgtem Kauf und Hereinfall zu wissen, dass eine eventuelle Anschaffung des ZCPR für meinen 128-er vielleicht nur hinausgeschmissenes Geld ist, nahm ich also die Herausforderung an und teilte Alexander meine Meinung mit. Nachstehend einige Auszüge aus meinem Mailbox-Dialog mit ihm:

```
GB> Es ist schon interessant, dass die ZCPR-Autoinstaller
GB> inzwischen von ehemals DM 150,-- auf DM 70,-- gefallen
GB> sind. Als gelernter Kaufmann wuerde ich das allerdings
GB> mangelnder Nachfrage zuschreiben. Und da die dazu passende
GB> P-D-Software keineswegs kostenlos ist - das Einschicken
GB> formatierter Disketten mit Rueckumschlag und Rueckporto
GB> finde ich sogar ziemlich laestig -, wuerde ich "Z3PLUS" z.
GB> Z. nicht einmal einfuehren, wenn das Install-Paket
GB> kostenlos waere: Meine bisherige Marken-Software (ausser
GB> "BDS-C" und "Wordstar 4.0") wird davon ja nicht besonders
GB> unterstuetzt. Und bei ZCPR-Programmen haette ich
GB> Kompatibilitaetsprobleme beim Dikettentausch mit Kollegen,
GB> die kein ZCPR haben! Aber vielleicht veroeffentlichst Du
GB> hier gelegentlich mal eine Liste der Software, mit der Du
GB> unter "Z3PLUS" arbeitest? Eventuell koennte man darunter
GB> etwas Interessantes entdecken?
```

Darauf reagierte Alexander zunächst zwar etwas sauer. Er (wie übrigens auch Fritz) ist immerhin Software-Verteiler der amerikanischen ZSIG-Organisation unter Helmut Jungkunz, der wiederum seit Frühjahr 1990 den Alleinvertrieb für Deutschland übernommen hat.

Wenn Diskussionspartner aber ehrlich auf der Suche nach dem richtigen Standpunkt und der Wahrheit sind, anstatt mit Gewalt rechthaben zu wollen, dann lässt sich auch ein Kompromiss finden.

```
GB> Wie mir scheint, ist ZCPR softwareseitig der Bereich,
GB> der mithilft, Deine leistungsfähige Hardware zu einer
GB> überdurchschnittlichen Z280-Anlage zu machen. Bei langsamen
GB> Rechnern mit geringer Speicherkapazität lohnt sich der
GB> Aufwand (der ZCPR-Installation) dagegen möglicherweise
GB> tatsächlich nicht; also beispielsweise kein CPC 664 ohne
GB> Speichererweiterung und zusätzlichem 5 1/4"-Laufwerk.
```

(Anmerkung: Der Schneider CPC 664 hatte einen Hauptspeicher von nur 64 KB für Betriebssystem und Anwenderprogramm zusammen, sowie ein Laufwerk für einseitige 3"-Disketten mit etwa 180 KB.)

AS: Dem kann ich nichts mehr hinzufuegen. Windows auf einem 8086er wuerde ja sicher auch keinen Spass machen ...

GB> Natürlich möchte ich Dich keineswegs drängeln, über Deine
GB> rund 150 ZCPR-Utilities jeweils einen Aufsatz zu schreiben,
GB> aber die wichtigsten dieser Programme wie Dein ZFILER
GB> würden mich schon näher interessieren.

AS: Der ZFILER selber bietet eigentlich recht wenige Funktionen, er ist mehr der Verwalter einer maechtigen Macrobibliothek. Das Hilfsmenue des ZF sieht so aus:

```
                  -- Filer Commands --
  A - Alpha Sort  C - Copy   D - Delete  J - Jump    T - Tag
  F - File Size   M - Move   P - Print   R - Rename  U - Untag
  W - Wild Tag    V - View   Y - Retag                   View
G - Group: Archive Copy Delete Fsize Move Print Reverse Tag Untag

  -- Misc --                   -- Cursor --
                        ^R  - Top of Screen      WS diamond
X - eXit ZFILER         ^C  - End of Screen
L - Login DU:/DIR:      ^T  - First File             ^E
S - disk Status         ^B  - Last File               ^
Z - ZCPR3 command     +/^F  - Next Screen       ^S <-+-> ^D
H - Help ZFILER       -/^A  - Prev Screen             v
E - refresh scrEen   SP/CR  - Next File              ^X
O - Option toggles      BS  - Prev File
```

Die eigentliche Funktion besteht nun darin, dass man auf dem Monitor ein sortiertes Directory hat und darin mit dem Cursor beliebig rumfahren kann. Hat man das File ausgewaehlt, kann man es mit den Grundfunktionen kopieren, verschieben, ausdrucken oder umbenennen. Wenn das nicht reicht, kann man den Filenamen mit einem entsprechenden Macro mit einem einzigen Tastendruck an externe Programme uebergeben, die dann denken, man haette das in der Kommandozeile eingetippt. Wenn man einen Befehl auf mehrere Files anwenden will, kann man diese markieren und dann werden die der Reihe nach bearbeitet. Das ist z.B. beim Crunchen von mehreren Files sehr praktisch.

Andere Annehmlichkeiten des ZCPR werden zwar auch durch externe Programme realisiert, aber die haengen so eng mit dem System zusammen, dass sie ohne ZCPR einfach keinen Sinn ergeben. Wenn man im ZCPR z.B. einen Filenamen eingibt, den es auf dem eingeloggten Laufwerk nicht gibt, wird erstmal der interne Suchpfad abgeklappert und wenn da auch nichts gefunden wird, wird ein Programm namens CMDRUN gestartet, das dann die weitere Fehlerbehandlung uebernehmen kann. Im Normalfall durchsucht dieses Programm ein Textfile namens ALIAS.CMD, das folgendermassen aufgebaut ist:

```
brief      cd wordstar;ws $1
```

Wenn man also BRIEF GUENTHER eintippt und es kein BRIEF.COM gibt, findet CMDRUN im ALIAS.CMD obige Zeile, loggt im Directory WORDSTAR ein und ruft den Wordstar mit WS GUENTHER auf. Das File kann mehr oder weniger beliebig lang werden und wenn man will, kann man sich fuer fast alles eine Abkuerzung schreiben.

Oder man macht das System auch fuer DOSler benutzbar, indem man

z. B. schreibt:

```
era=erase=del=delete      era $1
```

Egal, was man dann von den Befehlen links eintippt, es wird das Gewuenschte getan, naemlich das File geloescht.

Wenn man da nun einen langsamen Rechner mit langsamen Laufwerken hat, wartet man sich tot, bis man wieder beim CP/M-Prompt landet und wird das Z-System sicher umgehend wieder rauswerfen.

Das war also die instruktive Stellungnahme Alexander Schmids: Für Leute mit entsprechenden Rechnern sicher ein Anlass zum Nachdenken, ob sie jetzt in diesem Winter nicht die Einführung des ZCPR in Angriff nehmen sollten.

Für Commodore 128 User ergibt sich jedoch noch kein Handlungsbedarf. Dass andere C-128 Besitzer ebenso denken, beweist auch die Adressenliste von Helmut Jungkunz. Anfang 1993 hatten dort von knapp 100 eingetragenen ZCPR-Usern nur etwa 10 einen C-128. Für diesen Rechner wären einige andere Projekte von viel grösserem Interesse, beispielsweise:

8 MHz Karte: Die Firma Rossmöller Handshake GmbH in Meckenheim brachte bis 1990 eine "Turbo CP/M 128" Karte für DM 99,-- auf den Markt, womit der von Commodore eingebaute Prozessor Z-80 A durch einen Z-80 H ersetzt und die Rechengeschwindigkeit damit verdoppelt werden konnte. Wer Glück hat, kann sich vielleicht noch so ein Teil gebraucht beschaffen.

DCF-77 Zeit-Empfänger: Die Firma Conrad-Electronic in Hirschau brachte bis 1991 einen zündholzschachtel-grossen Atomuhr-Empfänger zum Preis von etwa DM 60,-- für den Userport des C-64 heraus (heute nur noch für Amiga?). Wenn jemand die dazugehörende Assembler-Software an das CP/M anpassen würde, wären auf dem C-128 automatische Zeiteinträge wie auf PCs möglich.

Festplatte: Verschiedene Firmen (wie die Scanntronik GmbH in Zorneding) liefern die amerikanische CMD-Harddisc für C-64 und C-128 mit 20 MB aufwärts und ab etwa 1100 Mark. Das Gerät wird von CP/M leider nicht optimal unterstützt und lässt sich bisher nur in Diskettengrössen partitionieren. Zum Einsatz dieser Festplatte wäre also eine Betriebssystem-Änderung durchzuführen.

Wie aus der Mitglieder-Umfrage im vergangenen Sommer ersichtlich wurde, besitzt über die Hälfte der Club-80 Mitglieder inzwischen (auch) einen PC und arbeitet mehr oder weniger häufig (nur) damit. Die Anwender von CP/M-Computern werden also immer weniger. Berücksichtigt man daneben aber auch die zusätzlichen Leser der Club-Informationen aus der Anwendergruppe "CP/M aktuell", so ist der Commodore 128 derzeit der noch meistbenutzte CP/M-Computer überhaupt. Und da anzunehmen ist, dass von diesen "Probe-Lesern" ab Januar 1994 doch einige ihren Beitrag auf das Konto des Club-80 überweisen, ist die Schlussfolgerung wohl erlaubt, dass wir hier eine Gruppe von C-128 Anwendern zusammenbekommen, unter denen vielleicht sogar ein ambitionierter System-Programmierer ist. Man darf gespannt sein! (GWB)

Tel.: 040 - 691 27 16

Jörg Brans - Tieloh 55, 22307 Hamburg

Club 80 Börse

Liebe Clubfreunde !

Da ich mich aus Zeitmangel nur noch mit MS-DOS beschäftigen kann, will ich mich daher von meinen Tandy Computern, Software usw. trennen. Vielleicht ist jemand im Club, der an dem folgendem Angebot Interesse hat:

Computer

Zwei Model 4P mit folgender Ausstattung:

1. Gerät 8 MHz (auf 2-4-6-8 per Software), 512K RAM, 2 x 40/80 Floppy - umschaltbar, Schnittstellen: P/S, Floppyanschluß extern

2. Gerät 6,3 MHz, 512K RAM, 2 x 40/80 Floppy - umschaltbar, Zeichensatz Tandy u. IBM (umschaltbar), Schnittstellen: P/S, Floppyanschluß extern

Peripherie

Gehäuse mit 2 Netzt. für 2 externe Laufwerke (eingebaut 1 x 3.5 Zoll - 720K Laufwerk)

Software (mit Lizenz)

Betriebssysteme : LSDOS 6.2 (deutsch) - LSDOS 6.3 (englisch)

Textverarbeitung: 1.) LeScript Vers. 2.0 (Spezialv. für deutsche Tastatur) mit zus. Druckertreiber für HP-Deskjet von Anitek
2.) Allwrite

Sprachen : 1.) Multi-Basic Vers. 1.00.01
2.) Bascom - Basic Compiler
3.) Pro-Create Editor/Assembler
4.) Assembly Language Development System (ALDS)

Integrierte Software : T/Maker mit folgenden 9 Funktionen:
* File Management
* Word Processing
* Spell Checking
* Speadsheet
* Database Management
* List Processing
* Data Transfer
* Graphics (Bar Charts)
* Programming

Utilities : Model 4 ToolBelt, Packer, DoubleDuty, Hyperdrive, Superdrive

Literatur : Mod 4 by Chris, Using Super Utility+ 3.x - Super Utility 4/4P
Job Control Language f. LDOS 5.3, TRSDOS 6.1 6.2 6.3
Programmierung des Z 80 v. Rodnay Zaks, ROM Listing Mod III
Vom Umgang mit CP/M, Multiplan deutsch, Schaltpläne Mod. 4/4P
Technical Reference Manual Mod. 4P, Micro 80 (letzte Jahrgänge)
Das DOS Buch (NEWDOS,GDOS,Color-DOS) f. TRS -80 I+III, Genie I,II,III,IIs + IIIs, Color Genie
Basic Handbuch v. ITT Microcomputer Software

Ich bin mir nicht sicher, ob ich an alles gedacht habe. Aber jetzt will ich zum Preis kommen.

Meine Preisvorstellung ist 700,- DM zzgl. Versandkosten

Selbstabholer wäre natürlich ideal. Wenn Interesse besteht, erhält der Abnehmer noch zusätzlich kostenlos einen Drucker DMP 430/DIN A3 (Druckkopf defekt). Außerdem gehört noch jede Menge PD-Software zum Angebot.

Wer sich dieses einmalige Angebot entgehen läßt hat selbst schuld. Also greift zum Telefon und wählt 040 - 691 27 16 !!

Gruß an alle die ich kenne

Jörg

Deskjet News

Der HP Deskjet duerfte inzwischen ja recht weit verbreitet sein und neben den diversen Nachfuellrezepten fuer die Tintenpatronen habe ich vor kurzem einen anderen sehr interessanten Tip im FIDO gelesen.

Wenn man im Ausdruck immer wieder weisse Streifen hat, kann das entweder an einer eingetrockneten Duese, an Luftblasen, die beim Nachfuellen entstanden sind, oder an einer Unterbrechung der Stromzufuehrung einer Duese bzw. einem durchgebrannten Heizelement in einer Duese liegen. Wenn Die Streifen nur von Zeit zu Zeit auftreten, liegt es wahrscheinlich an Luftblasen und es hilft oft, wenn man die Patrone mit der Seite flach auf einen Tisch klopft. Um aber festzustellen, ob eine Patrone wirklich defekt, oder vielleicht auch nur nicht richtig eingerastet ist, braucht man keine teuren Messgeraete, das sagt einem freundlicherweise der Selbsttest, auch wenn ich dazu im Handbuch nichts gefunden habe. Man drueckt dazu waehrend des Einschaltens die FONT-Taste und beobachtet die Dinge, die sich entwickeln. Ganz am oberen Rand, noch vor dem Treppchen, steht normalerweise ein "ID H" o.ae., wenn dort aber Zahlen vor dem ID stehen, z.B. "11 24 ID E", dann bekommen in diesem Beispiel die Duesen Nr. 11 und 24 keinen Saft. Wenn man Glueck hat, hat man nur mit seinen Marmeladenfingern auf die Kontakte gefasst und es reicht, wenn man sie mit einem spiritusgetraenkten Lappen reinigt. Wenn dann auch ein vorsichtiges Festdruecken nicht hilft, ist wohl ein Heizelement durchgebrannt und man kann die Patrone entsorgen.

Wenn das dann mal der Fall sein sollte, steht man allerdings vor dem Problem, einen neuen Druckkopf zu bekommen. Immer mehr Haendler scheinen naemlich der Meinung zu sein, dass es fuer sie doch wesentlich guenstiger ist, wenn sie die Nachfuellsets gleich selber zu einem Wucherpreis verkaufen. In den meisten Geschaeften bekommt man dann nur noch diese Sets oder die Patronen mit der doppelten Kapazitaet, die man aber praktisch nicht nachfuellen kann. Bei den "normalen" Patronen heisst es dann immer, dass es die nicht mehr gibt. Irrtum, nur steht vorne auf der Schachtel DeskJet 300J und DeskJet Portable. Ist aber die alte Patrone und hinten drauf steht auch die ganze Litanei an Druckern, wo die Patrone sonst noch reinpasst. Also, nicht verar***** lassen, die Patronen fuer den tragbaren DJ kaufen und munter weiter selber nachfuellen.

Alexander Schmid

--- FMail 0.94+
* Origin: * Discovery Club 80 - Ecke * LifeNet (77:7700/740)

MODEM umsonst!!!???

Schon im letzten Info habe ich kurz die wohl preiswerteste Möglichkeit angesprochen, zu einem postzugelassenen Modem zu kommen. Sicher habt ihr aber auch schon selbst die Werbung der 1&1 Telekommunikation GmbH gesehen, die in letzter Zeit des öfteren Computerzeitschriften beigelegt war.

Ein Modem zum Nulltarif, nämlich auf unbegrenzte Zeit kostenfrei ausgeliehen, wird da angeboten. Verbunden mit dem Leihmodem ist zudem eine kostenlose Eintragung als BTX-Teilnehmer, die normalerweise schon 65,-- DM kostet. Und obendrein gibt es ein, ebenfalls kostenloses, BTX-Decoder-Programm für PC´s (DOS oder WINDOWS), Atari oder Amiga.

Aber was bringt das Angebot dem Besitzer eines CP/M-Rechners, der nur auf der Suche nach einem billigen Modem für den Einstieg in die DFÜ ist? Leider nichts!

Das von 1&1 kostenfrei abgegebene Modem ist ein speziell für den BTX-Anschluß nach der "alten" V.23-Norm konzipiertes Gerät. Es arbeitet deshalb auch nur mit den, im restlichen DFÜ-Bereich (zum Glück) unüblichen, Übertragungsraten von 75 bps für den Sende- und 1200 bps für den Empfangsbetrieb. Diese Übertragungsgeschwindigkeit ist nicht, wie z.B. bei hayeskompatiblen Modems üblich, vom Rechner aus änderbar. Damit ist das Modem für DFÜ-Einsteiger aus dem CP/M-Lager praktisch wertlos.

Im Gegensatz dazu sind die ebenfalls zu einem recht günstigen, allerdings inzwischen fast marktüblichen Kaufpreis angebotenen 2400´er (129,--) und 14400´er (349.--) Modems der gleichen Firma durchaus auch für andere Zwecke als den BTX-Anschluß verwendbar. Auch in diesem Preis ist die BTX-Anschlußgebühr von 65,-- DM enthalten. Stellt sich nur die Frage, welchen Wert der BTX-Anschluß für einen CP/M-User hat, für dessen Rechner es keinen Software-Decoder gibt (oder sollte da etwas an mir vorbeigelaufen sein)?

Und so interessant ist BTX nun auch wieder nicht, daß man sich dafür extra einen PC zulegen müßte ;-)

Allways good Hacking,

Impressum

1. Vorsitzender:	**Hartmut Obermann** Mozartring 23 Postfach 14 30	Tel.: 0 82 21/ 3 02 48 BTX: 0 82 21/ 3 02 48 FAX: 0 82 21/ 3 35 75
89304	**Günzburg**	
2. Vorsitzender:	**Gerald Schröder** Am Schützenplatz 14	Tel.: 0 41 05/ 26 02
21218	**Seevetal**	
Hardwarekoordinator:	**Andreas Magnus** Bismarckstraße 29	Tel.: 02 09/ 87 02 30
45879	**Gelsenkirchen**	
NewDOS-Diskothekar:	**Oliver Volz** Am Ochsenwald 37A	Tel.: 07 11/ 74 40 51
70565	**Stuttgart** (Rohrerhöhe)	
CP/M-Diskothekar:	**Fritz Chwolka** Saarstraße 34	Tel.: 0 24 64/ 89 20
52457	**Aldenhoven**	
C-128-Diskothekar:	**Günther W. Braun** Postfach 80 02 26	
81602	**München**	
Clubbücherei:	**Kurt Müller** Sophie-Scholl-Ring 3b	Tel.: 0 41 52/ 7 06 43
21502	**Geesthacht**	
Redaktion:	**Jens Neueder** Gschlachtenbretzingen Rudolf-Then-Straße 32	Tel.: 07 91/ 4 28 77 BTX: 07 91/ 44 47 22 FAX: 0 79 71/ 2 50 55
74544	**Michelbach / Bilz**	
Bankverbindung:	**Club 80** Postgiro Sonderkonto CLUB 80 Obermann H., 8870 Günzburg	**Postgiroamt Frankfurt** **BLZ: 500 100 60** **Kto.Nr.: 496 071 - 605**
Autoren:	Die Redaktion bedankt sich bei den im **Inhaltsverzeichnis** genannten Autoren für die Mitarbeit an der Club-INFO. Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.	

Schluß

Hallo Club 80'er,

zum Jahreswechsel haltet Ihr das 42. Club-Info in den Händen. Ich möchte mich an dieser Stelle für Eure Mitarbeit an unserer Club-Info bedanken und für die Feiertage die besten Wünsche übermitteln.

Im Anhang des Info's ist diesmal dabei:
InfoForm - Info & Tips für Artikelschreiber
Messeliste - Übersicht diverser Computermessen
Geburtstagsliste - Aufstellung der Geburtstage der Clubmitglieder

Zu der Geburtstagsliste möchte ich noch anmerken, daß sie nicht vollständig ist, da nicht alle Clubmitglieder Ihr Geburtsdatum angegeben haben. Hervorzuheben sind vielleicht die Häufung der Geburtstage im April sowie die Tage **19.04.** und **31.05.** Obwohl wir eine kleine Gruppe sind, und es doch über 360 Möglichkeiten gibt einen Geburtstag zu feiern, haben sich zu beiden Tagen jeweils drei von uns entschieden gemeinsam Geburtstag zu feiern. Das dann noch ein genau 10-jähriger Altersunterschied hinzukommt ist sicher nur Zufall. Sicher sind diese zwei Termine besonders geeignet für Clubtreffen, da die "**Rundenzahler**" schon feststehen. ...soweit zum Mißbrauch von Statistiken.

Ich hoffe die Winter-/Schlechtwetterzeit ermöglicht Euch die Ausarbeitung neuer Artikel für unsere Club-Info. Viel Spaß beim Computern. Ich wünsch Euch nochmals ein Gutes Neues.
Bis zum nächsten mal Euer

Jens

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
26	Mössel	Franz	Schafferstraße 12	I 39012	Meran	0039-473/34178	-	-	-
	IBM, Joyce, Workemate-Bullet■ Modem					0039-471/980496	-	2:333/400	-
27	Neueder	Jens	Rudolf-Then-Straße 32	GER 74544	Michelbach / Bilz	0791/ 42877	-	791444722	CCWN
	IBM 286/386/486, Atari ST 1040, TRS80 MI■ Modem, Sound Galaxi NX, Scanner, Tape, Borsu					07971/ 250-50	07971/ 250-55	1001667	0715168434
28	Neumann	Christof	Zeitblomstraße 22/2	GER 89077	Ulm /Donau	0731/ 6022568	-	-	-
	IBM, Tandy MII, Tandy M4p■ Novell					0731/ 9749720	-	-	-
29	Obermann	Hartmut	Mozartring 23	GER 89312	Günzburg	08221/ 30248	08221/ 33575	0822130248	SYNREL
	IBM486, Tandy M4p, Epson PX-8 ■ Modem, Scanner					-	-	2:241/7922.10	08282/ 4311
30	Retzlaff	Bernd	Kleiner Sand 98	GER 25436	Uetersen	04122/ 43551	-	-	-
	IBM 386, C64, GENIE I■					04103/ 605310	-	-	-
31	Rinio	Gerd	Rennbahnstraße 9	GER 22111	Hamburg	040/6552630	-	-	-
	IBM 486DX66/2, RTS 80, TRS 80 RS ■ Modem					-	-	-	-
32	Ruschinski	Claus	Pommernstraße 21	GER 45770	Marl	02365/ 34646	-	-	-
	IBM 386, TRS80 M I■ Highscreen-Scanner, CoProz IIT387					-	-	-	-
33	Schimmer	Jörg	Stettinerstraße 28	GER 60388	Frankfurt	06109/ 35336	-	-	-
	IBM 486, Schneider CPC■ Modem CSR2400					069/ 3800-2385	-	2:249/70.9	-
34	Schmid	Alexander	Entmannsdorf 5	GER 96317	Kronach /Gehülz	09261/ 53496	-	-	-
	GENIE IIs, GENIE IIIs, CPU 280■ Modem 2400,Prommer80,Ramdisk,Club80Terminal					-	-	2:2400/830	-
35	Schmitz	Rainer	Küferweg 12/1	GER 73099	Adelberg	07166/ 1397	-	-	-
	IBM,Portfolio,NCR Decision Mate V,Joyce,... ■ Modem 1200, Märklin Digital Interface 6050					07161/ 608-475	-	-	-
36	Schoberth	Uwe	Petrus-Waldus-Straße 14	GER 75443	Oetisheim	07041/ 7254	-	-	-
	Alphatronic P3■					0711/ 89394500	0711/ 89394513	-	-
37	Scholz	Hans-Werner	Spitalstraße 54	GER 41334	Nettetal	02157/ 3613	-	-	-
	IBM 386, Prof 80, ITT 3030■ Prommer80					-	-	-	-
38	Schroers	Horst-Dieter	Breslauer Straße 9	GER 85622	Feldkirchen	089/ 9032615	089/ 9043413	-	-
	IBM■ Modem, Scanner					-	-	-	-
39	Schröder	Gerald	Arminiusstraße 2	GER 22525	Hamburg	040/ 8507131	-	-	-
	IBM 386SX, Atari 1040STF, Z280■					040/ 54715334	-	-	-
40	Schröer	Egbert	Joachimstraße 18	GER 46284	Dorsten	02362/ 75311	-	-	-
	Portfolio, TRS80 MI, GENIE I, GENIE IIIs■					02362/ 49-9649	-	-	-
41	Schulte	Hartmut	Entenschnabel 8	GER 31311	Uetze	05173/ 1248	05173/ 24631	-	-
	IBM, Z280 u.a.■ Scanner					-	-	-	-
42	Sonnemann	Harald	In den Eckwiesen 9	GER 64405	Fischbachtal	06166/ 8512	-	-	-
	NDR-Klein ■ parallel/seriell, EPROMer					06151/ 92-1265	-	-	-
43	Stumpferl	Stefan	Hasenbergstraße 57	GER 80933	München	089/ 3138193	//0893144001//	-	-
	Amstrad CPC 6128+, GENIE IIIs ■ Modem, 20MB-Wechselplatte, PSG&PIO,SIO, ...					-	-	-	-
44	Sörensen	Rüdiger	Wiesbadener Str. 28B	GER 55252	Mainz-Kastel	06134/65342	-	-	-
						-	-	-	-
45	Tornow	Wilhelm	Elbblick 46	GER 21629	Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
	IBM 386DX, Atari Mega ST4, Tandy M4p■ Streamer,Soundblaster,Modem 2400,F.A.K.S. 910					-	-	-	-
46	Vogl	Michael	Weidenweg 15	GER 41515	Grefenbroich /Laach	02181/45112	-	0218145112-0001	-
	IBM, Amstarad CPC 464/6128■ Modem Gigitek 2400, Scanner Dat F.CPC					-	-	-	-
47	Volkmer	Richart	Am Spörkel 69	GER 44227	Dortmund	0231/ 752574	-	-	-
	IBM XT, Apple LC, CPC 6128, Osborne, TandyMII■ Modem 2400, ScanMan-Scanner					-	-	-	-
48	Volz	Oliver	Am Ochsenwald 37A	GER 70565	Stuttgart (Rohrerhöhe)	0711/ 744051	-	-	-
	IBM, GENIE IIs■ Modem Avantec					0711/ 685-3013	-	-	-
49	Werner	Heiko	Reichenberger Straße 5	GER 01129	Dresden	0351/ 4608612	-	-	-
	IBM 286/486■ Modem 2400, Scanner					-	-	-	-

Adressen-Liste **CLUB80**

Stand vom 29-12-1993

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat ■ Tel. geschäftl. ■	Telefax privat ■ FAX geschäftl. ■	BTX ■ FIDO-Node ■	Mailboxname ■ Mailboxnummer ■
1	Barendt	Harry	Hermann-Löns-Straße 7	GER 50181	Breedburg (Erft)	02272/7168	-	-	-
						-	-	-	-
2	Berndt-Jochum	Ilse	Stachelsgut 24	GER 51427	Bergisch Gladbach	02204/ 65254	-	-	-
	IBM, GENIE III, GENIE IIIs, SHARP Pocket 1600■ Scanner					02204/ 65254	-	-	-
3	Bernhardt	Helmut	Hafenstraße 7	GER 24262	Heikendorf	0431/ 241907	0431/ 245717	-	-
	IBM, Prof 180, CPU 280■ diverse PCs,Novellite-Netz,1496E,Mustek105+					0431/ 77578-20	-	2:242/262.26	-
4	Bielenberg	Georg	Erikaweg 1	GER 24568	Kaltenkirchen	04191/ 3751	-	-	-
	Atari 260ST, C128, Schneider Joyce ■ Modem: LC2496 Digitech, Scanner: Supersc.III					04193/ 90430	-	-	-
5	Brans	Jörg	Tieloh 55	GER 22307	Hamburg	040/ 6906531	-	-	-
	IBM■ CD-Rom, Streamer, Soundkarte					-	-	-	-
6	Braun	Günter W.	Postfach 80 02 26	GER 81602	München	-	-	-	-
	Commodore 128 D■ Akustikkoppler					-	-	-	-
7	Braun	Harald	Postfach 8011	GER 24154	Kiel	0431/35139	-	-	-
						-	-	-	-
8	Böckling	Ulrich	Juchaczstraße 61	GER 56203	Höhr-Grenzhausen	02624/ 4861	-	-	-
	IBM 386DX33,1040ST,VC20,C64,TRS80 M I,ZX81 ■ Modem, Videodat-, Videotextdecoder					02631/ 895168	-	-	-
9	Chwolka	Fritz	Saarstraße 34	GER 52457	Aldenhoven	02464/ 8920	-	-	-
	IBM 386+Co, Commodore, Apple, Z 280■ Modem 2.4					-	-	2:248/242:8	-
10	Dose	Volker	Dorfstraße 10	GER 24235	Brodersdorf	04343/ 1357	-	-	-
	GENIE IIIs mit Z180■ EPROMer, Modem					-	-	-	-
11	Halgasch	Gert	Großschönauer Straße 26	GER 02796	Jonsdorf	035844/ 636	-	-	-
	IBM 386DX■					-	-	-	-
12	Hartmann	Hans-Günther	Möwenstraße 9	GER 27804	Berne	04406/ 6911	04406/ 1071	-	-
	IBM 386SX, TANDY M4p, Z280-Kartenrechner■ Real Time Clock, Speed Up Kit 6,3MHz, 40MB-P1					0421/ 248-2419	-	2:240/300.24	-
13	Hebecker	Ulrich	Büsnauer Straße 15	GER 70563	Stuttgart	0711/ 734800	-	-	-
	IBM 286+386, 128D, 1581, Kaypro484■					-	-	-	-
14	Held	Manfred	Stirner Straße 22	GER 91785	Pleinfeld	09144/ 6563	09144/ 8514	-	-
	IBM■ Modem ZyXEL 1496+, CD-ROM					0911/ 219-2245	-	2:2400/10.10	-
15	Hermann	Klaus	Forchenstraße 8	GER 72124	Pliezhausen	07127/ 71945	-	-	Discovery
	IBM 386■ ET 4000, Modem					-	-	2:2407/70.740	07127/70107
16	Hürdler	Manfred	Niederhoferstraße 29	GER 97222	Rimpar	09365/ 4235	-	-	-
	Victor Sirius 1 (IBM), CPC 6128■					-	-	-	-
17	Johnen	Willi	Hansemannstraße 1	GER 52351	Düren	02421/ 501305	-	-	-
	GENIE IIIs■					02421/ 33064	-	-	-
18	Kauka	Dietmar	Straße des Friedens 37	GER 04552	Neukirchen (Borna/Leipz	03433/851019	-	-	-
						-	-	-	-
19	Kemmer	Jürgen	Dorfberg 7	GER 97232	Sulzdorf	09334/ 1050	-	-	-
	IBM 386, UltraSound, ZyXEL U1496E+■ DCF-77-Empfänger parallel, IR-Sender seriell					-	-	2:247/2086.1	-
20	Kuhn	Eckehard	Im Dorf 14	GER 72636	Frickenhausen	07022/ 45417	-	-	-
	Atari ST 1040, TRS80 M I■					-	-	-	-
21	Linder	Jörg	Küstriner Str. 68	GER 15306	Seelow	03346/ 520	-	-	-
	KC 85/4 mit Floppy■					-	-	-	-
22	Lorenz	Walter	Mahräckerstraße 9	GER 60431	Frankfurt /Main	069/ 531656	-	-	-
	IBM 286/486, Z80-, HD 64180-Eigebau, Z280-T.R■ Scanner, Soundkarte, ET 4000					-	-	-	-
23	Magnus	Andreas	Bismarckstraße 29	GER 45879	Gelsenkirchen	0209/ 144029	-	-	-
	IBM 386Dx, GENIE IIIs■ Modem 2400					-	-	-	-
24	Mahlert	Herbert	Hohenbudbergerstraße 112 A	GER 47229	Duisburg	02065/ 47217	-	-	-
	IBM, GENIE I■ c't Videotext-Karte, Vobis Videodat-Decoder					02065/ 902592	-	-	-
25	Müller	Kurt	Sophie-Scholl-Ring 3B	GER 21502	Geesthacht	04152/ 70643	-	-	-
	IBM, Atari Mega ST4 ■ ATONCE 386,OverScan,Scanner,HBS640-T36,Modem					040/ 89983403	-	-	-

HARDWARE + SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

CLUBINFO

43. Ausgabe

Kontaktadresse:

Club 80

Hartmut Obermann

Wilhelm-Baur-Straße 8

Postfach ______

76135 Karlsruhe

Tel.: 0721/854068

BTX: 0721/______

Anrufbeantworter &

FAX: 0721/______

Inhaltsverzeichnis

	Autor & Seite
Clubinternes	
Neues vom Vorstand	1 Hartmut Obermann
Termine	3 Jens Neueder
Hardware	
VGA-Karte anstelle Hercules am Club-80-Terminal	5 - 14 Helmut Bernhardt
Club 80 Börse	
Biete ...	15 - 16 Andre Schut
Sonstiges	
Neue Sachen CP/M 80	17 - 18 Aera CPM.GER Hartmut Obermann
Mailbox-Führer '94	19 - 20 Günther W. Braun
Die letzten Seiten	
Impressum	21
Schluß	22 Redaktion
Mitgliederadressenliste	am INFO-Ende

Neues vom Vorstand

Hallo CLUB 80 - Freunde,

inzwischen dürfte wohl jeder seine Einladung zum Treffen nach Kronach bekommen haben. Es war den Verantwortlichen wichtig, die Einladung möglichst frühzeitig zu den Mitgliedern zu bringen, um diesen eine langfristige Planung zu ermöglichen. Ich glaube, das ist uns gelungen!

Leider ist bisher die erwartete Flut von Anmeldungen ausgeblieben. Aber was nicht ist, kann ja noch werden! Ich würde mich wirklich freuen, wenn möglichst viele von euch zu dem Treffen erscheinen und es damit zu einem vollen Erfolg machen würden.

Ein festes Programm für das Treffen gibt es noch nicht. Auf jeden Fall gibt es jede Menge interessante Gespräche zu Computerthemen, einige (teilweise recht exotische) Rechner und sicher viel Spaß. Auch mitreisende (Ehe-)Frauen werden sich auf keinen Fall langweilen!

In diesem Sinne wünsche ich mir eine rege Beteiligung und uns allen ein schönes Treffen in Kronach. Für all diejenigen, die ihre Einladung irgendwo unter einem Stapel Papier vergraben haben und nicht mehr wiederfinden, hier nochmals die wichtigsten Fakten:

Anreise für die Freaks :	12.05.94
für die restlichen Teilnehmer :	13.05.94
offizieller Teil des Treffens :	14.05.94 ab ca. 13:00 Uhr
Abreise:	15.05.94 nach dem Mittagessen

Ort des Treffens: Gasthaus "Erlabrück"
Telefon 09262/320

Anmeldung bis spätestens: 15. April 1994
Anmeldung bei: **Alexander Schmid**
Entmannsdorf 5
96317 Kronach/Gehülz
Tel. 09261 / 53496

Ansonsten gibt es nicht viel neues vom Vorstand. Positiv anzumerken ist, daß der CLUB 80 drei Neuzugänge zu verbuchen hat. Dabei handelt es sich um **Andreas Kleespies**, **Kurt Messerschmidt** und **Hans-Jürgen Walter**. Wenn die drei Neuen meinem Vorschlag folgen, können wir, vielleicht schon in diesem, spätestens aber im nächsten Info, eine kurze Vorstellung ihrer Person, Rechner und Interessen lesen.

Nicht ganz so positiv sieht es mit der Nutzung des CLUB 80-Bereichs in der DISCOVERY- Maibox aus. Irgendwie will der Informationsaustausch nicht so richtig in Gang kommen, was aber sicherlich auch and er relativ geringen Anzahl der mit Modem ausgerüsteten Clubmitglieder liegt. Ich habe mir schon überlegt, ob sich der CLUB 80 nicht ein oder zwei Leihmodems zulegen soll, um das Interesse an der Datenfernübertragung zu fördern. All die, die Interesse an einem solchen Service haben, möchte ich bitten, sich schon mal bei mir zu melden. Ich kann dann den Bedarf besser abschätzen und habe auf der Hauptvesammlung ein Argument für die Bewilligung des Geldes für die Anschaffung der Modems.

Daß ich im April umziehe, habe ich ja schon im letzten Info bekanntgemacht.
Hier meine neue Adresse, die ab der ersten Aprilwoche gültig ist:

Hartmut Obermann
Wilhelm-Baur-Str. 8
76135 Karlsruhe
Tel.: 0721/ 854068

Im nächsten Info erscheint dann meine komplette neue Adresse incl. Telefon- und FAX-Nummer.

Die Kontonummer des CLUB 80-Kontos ändert sich, da es sich um ein Postgiroamt handelt, nicht.

Das wäre für diesmal schon alles. Ich wünsche Euch viel Spaß bei der Lektüre des Infos und bei der Computerei,

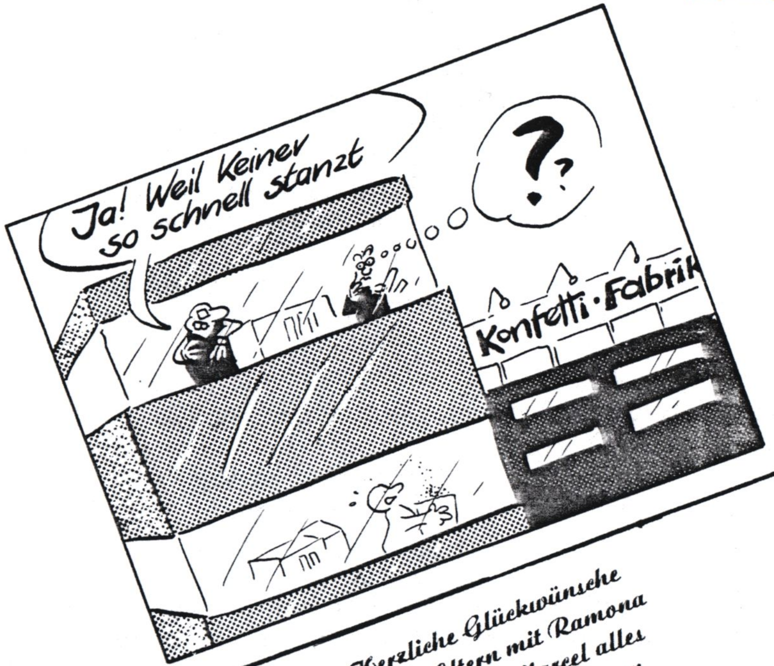
Termine... Termine... Termine... Termine... Termine

Mevocom	Schwerin	22.04.- 24.04.94
Das moderne Büro	Stuttgart	03.05.- 06.05.94
Computer '94	Hongkong	11.05.- 14.05.94
Infobase	Frankfurt /Main	17.05.- 19.05.94
Networks Expo	Düsseldorf	17.05.- 19.05.94
CAT	Stuttgart	17.05.- 20.05.94
Comdex Spring	Atlanta	23.05.- 26.05.94 (Georgia,USA)
Windows World	Atlanta	23.05.- 26.05.94 (Georgia,USA)
PC Expo	New York	Juni 94
Multimedia	London	07.06.- 09.06.94

Redaktionsschluß für das nächste Clubinfo ist der

01. Juni 1994.

Das nächste INFO wird allerdings erst Ende Juli erscheinen. Oben genannter Redaktionsschluß wird aber trotzdem beibehalten.



Herzliche Glückwünsche an die Eltern mit Ramona und Patrick Marcel alles Gute für seinen weiteren Lebensweg.
Der Vorstand

Wir freuen uns über die Geburt unseres Sohnes und Brüderchens

Patrick Marcel
* 28.Januar 1994
3860 g
54 cm

Die glücklichen Eltern und Schwester:
Oliver, Anita und Ramona Volz
Am Ochsenwald 37 A
70565 Stuttgart

VGA-Karte anstelle der Hercules-Karte am Club-80-Terminal

Helmut Bernhardt

So sehr viel was anderes als die Hercules-Karte ist die VGA-Karte auch nicht. Bei beiden Karten ist der Bildwiederholspeicher sowohl im Text- als auch im Grafik-Modus memory mapped und die CPU hat sich selbst darum zu kümmern, an welche Adressen die Zeichen bzw. die Grafik-Bytes zu verholen sind. Beide Karten sind völlig passiv und weisen keinerlei Videoprozessor-Funktionen auf.

VGA-Karten gibt es in einer Vielzahl von Erscheinungsformen, sie können einen 8- oder 16-Bit-Bus haben, sie können im Fall eines 16-Bit-Bus in 8-Bit-Slots des PC funktionieren oder dies auch unterlassen, sie können ihr BIOS in 8- oder 16-Bit-Breite zugänglich machen, sie können 256K oder auch bis 2MB Video-RAM enthalten, sie können maximal 256, 32K, 64K oder 16 mega Farben gleichzeitig auf dem Monitor darstellen, sie können eine einfache VGA-Karte oder eine dazu abwärtskompatible Video-Karte mit zusätzlichem Grafik-Prozessor (Windows-Accellerator) sein, sie können die Super-VGAModi VESA-kompatibel oder auch mit Wildwuchs-Einbindungen oder auch garnicht unterstützen, und es ist nicht abzusehen, welche Überraschungen noch zu erwarten sind bzw. bereits existieren.

Für den PC-Benutzer ist diese Vielfalt - abgesehen von Performanceunterschieden - relativ schnuppe, dort funktioniert die Karte fast immer und mit entsprechenden Treibern auch unter Windows mit ihren eventuell zusätzlichen Eigenschaften.

Wenn eine VGA-Karte im Club-80-Terminal laufen soll, dürfen einige Umstände nicht gegeben sein. Von den Abmessungen her darf es nicht eine der riesigen Lappen sein, die am ECB-Bus nicht mehr als Huckepack-Karte auf das Terminal paßt. Es darf auch keine 16-Bit-Karte, die im 8-Bit-Slot nicht läuft, sein, und sie muß registerkompatibel zur ursprünglichen VGA von IBM sein (was meistens der Fall ist).

Hier sollen die Grundlagen der Einbindung der VGA-Karte in das Betriebsprogramm des Club-80-Terminals herausgestellt werden.

Leider geht es nicht ganz ohne Hardware-Eingriffe, deshalb soll erst dieser Aspekt behandelt werden.

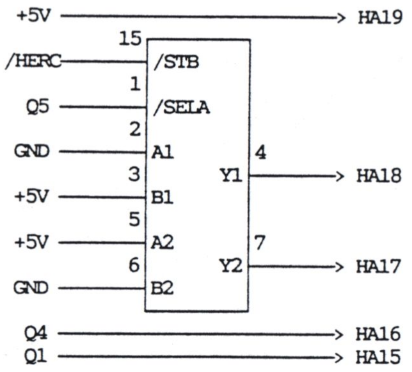
Hardware-Anpassung

Die Hercules-Karte belegt im PC zwei 32K-Video-Seiten an den Adressen B0000..B7FFF und B8000..BFFFF. Zur Erzeugung des Freigabesignals für den Speicherzugriff genügte es, eine low aktive Adreßleitung A18 auf die Hercules-Karte einwirken zu lassen, wenn die CPU ein low aktives /MERQ und einen High-Pegel auf A15 ausgab. Die an die Hercules-Karte gelangenden Adressen A16, A17 und A19 waren dauerhaft high und der Pegel von A15 (zur Hercules) wurde durch ein Bit eines Latch gesteuert.

Die VGA-Karte benutzt diese Speicherbereiche auch als Video-RAM in den Text-Modi. Im Grafik-Modus wird aber der Bereich A0000..AFFFF benutzt und das BIOS in einem ROM auf der VGA-Karte belegt den Bereich C0000..C7FFF. Natürlich läuft die darin enthaltene Software nicht auf dem Z80 des Terminals - es gibt aber eine Menge wichtiger Tabellen in diesem ROM und eine Reihe von Zeichensätzen (die für die Benutzung in den Textmodi erst in die Speicherebene 2 geladen werden müssen). Die Pegel der Adressen A15..A19, die für die Freigabe der einzelnen Bereiche maßgeblich sind, seien hier zusammengestellt - daraus ergibt sich dann auch die einfache Schaltung, die diese Bereiche ansteuern kann.

Adreß-bereich	A19	A18	A17	A16	A15 (zur VGA-Karte)	Speicherart
A0000..A7FFF	1	0	1	0	0	Grafik, erste Hälfte
A8000..AFFFF	1	0	1	0	1	Grafik, zweite Hälfte
B0000..B7FFF	1	0	1	1	0	Text, monochrom
B8000..BFFFF	1	0	1	1	1	Text, color
C0000..C7FFF	1	1	0	0	0	BIOS-ROM

Es ist zu erkennen, daß A19 für alle Bereiche high ist, Diese Leitung kann an +5V angeschlossen bleiben. A18 und A17 haben entweder das Muster 01 oder 10, wenn hier 00 oder 11 anliegt, wird die VGA nicht angesprochen. Damit läßt sich eine Freigabe der Karte herleiten, in die die bisherige Freigabe (/HERC = /MERQ * A15) einbezogen wird. Dafür ist ein zusätzliches IC 74HCT157 nötig, das in das Rasterfeld am oberen Ende der Terminal-Karte eingelötet und folgendermaßen beschaltet wird:



/HERC ist das bisherige Freigabe-Signal für die Hercules-Karte, dessen Verbindung zu HA18 (A18 des PC-Slot) getrennt wird und Q5, Q4 und Q1 sind Ausgänge des 74HCT-259-Latch (wobei Q1 bereits an HA15 und +5V an HA19 des PC-Slot liegen.

Wenn /HERC nicht low aktiv ist, sind alle Ausgänge des 74HCT157-Multiplexers high (oder low - so genau weiß ich das nicht aus dem Kopf), auf alle Fälle liegt dann nicht eines der Freigabemuster 01 oder 10 an HA18/HA17.

Nun aber zur Programmierung:

Im Laufe der Jahre hat sich die Video-Ausgabe beim PC von ursprünglich einer maximalen Auflösung von 640*200 Punkten im Monochrom-Modus auf heute 1024*768 Punkten mit gleichzeitig 256 aus 256k Farben oder auch 640*480 Punkten mit gleichzeitig 16 mega Farben entwickelt. Für eine Reihe von Auflösungen haben sich dafür Standards entwickelt,

andere werden nie benutzt bzw. von jedem Hersteller mit anderen Modus-Nummern bedacht. Hier werden deshalb nur die von IBM abgesegneten Modi 00h bis 13h behandelt; die folgenden Überlegungen gelten dann auch für alle mit IBM registerkompatiblen VGA-Karten.

Standard Video Modi der VGA

Modus		Auflösung	Farben	Zeichen-Box	Seiten	Adresse	Länge	Berechnung der Offsetadresse
0h	Text	40x25	16	8x8	8	B800	800	2*(40*y+x)
1h	Text	40x25	16	8x8	8	B800	800	2*(40*y+x)
2h	Text	80x25	16	8x16	8	B800	1000	2*(80*y+x)
3h	Text	80x25	16	8x16	8	B800	1000	2*(80*y+x)
4h	Grafik	320x200	4	8x8	1	B800	4000	8152*(y and 2)+y*40 + x div 4
5h	Grafik	320x200	4	8x8	1	B800	4000	8152*(y and 2)+y*40 + x div 4
6h	Grafik	640x200	2	8x8	1	B800	4000	8152*(y and 2)+y*40 + x div 4
7h	Text	80x25	mono	9x14	8	B000	1000	2*(80*y+x)
8h	reserviert							
9h	reserviert							
Ah	reserviert							
Bh	reserviert							
Ch	reserviert							
Dh	Grafik	320x200	16	8x8	8	A000	2000	y*40 + x div 8
Eh	Grafik	640x200	16	8x8	4	A000	4000	y*80 + x div 8
Fh	Grafik	640x350	mono	8x14	2	A000	8000	y*80 + x div 8
10h	Grafik	640x350	16	8x14	2	A000	8000	y*80 + x div 8
11h	Grafik	640x480	mono	8*16	1	A000	A000	y*80 + x div 8
12h	Grafik	640x480	16	8*16	1	A000	A000	y*80 + x div 8
13h	Grafik	320x200	256	8*8	1	A000	FFFF	320*y +x

in den Modi 0h bis 3h (und 7h bei der VGA) hat die EGA-Karte 14 und die VGA-Karte 16 Rasterzeilen pro Zeichen

Erweiterte Super-VGA-Modi für Tseng ET4000 Chip (und HiColor-DAC)

Diese Auflösungen können bei anderen Super-VGA-Karten mit anderen Modus-Nummern zu bedienen sein; erst der VESA-Standard hat hier nachträglich Einheitlichkeit gebracht, ältere Karten wußten davon noch nichts. Es ist auch nicht bekannt, wo für diese Modi in der Parameter-Table die Daten stehen.

29h	Grafik	800x600	16	8*16	1	A000	FFFF	
2Eh	Grafik	640x480	256	8*16	1	A000	FFFF	
	Grafik	640x480	32k	8*16	1	A000	FFFF	mit Sierra-DAC
30h	Grafik	800x600	256	8*16	1	A000	FFFF	
	Grafik	800x600	32k	8*16	1	A000	FFFF	mit Sierra-DAC
37h	Grafik	1024x768	16	8*16	1	A000	FFFF	
38h	Grafik	1024x768	256	8*16	1	A000	FFFF	
	Grafik	1024x768	32k	8*16	1	A000	FFFF	mit Sierra-DAC

Grundsätzlich gibt es zwei Strategien, wie dem Z80 des Terminals all die Informationen zur Initialisierung eines Video-Modus zur Verfügung gestellt werden:

1) Auf dem PC wird durch entsprechende Programme festgestellt, wo im BIOS-ROM diese Informationen vorliegen und der Z80 bedient sich aus dem BIOS-ROM selbst.

2) Alle Informationen zur Initialisierung der relevanten Modi werden auf dem PC ausgelesen und in ein EPROM gebrannt, das anstelle des BIOS-EPROM in die VGA-Karte gesteckt wird.

Für beide Methoden soll hier das Vorgehen im Einzelnen beschrieben werden. Die zweite Methode kann eventuell bedeuten, daß der Inhalt des EPROM noch byteweise auf zwei EPROMs aufgespalten werden muß, wenn eine 16-Bit-VGA vorliegt.

Registerinitialisierung

Mit welchen Werten der Registerdschungel einer VGA-Karte in den verschiedenen Modi zu initialisieren ist, weiß das BIOS der VGA-Karte aus einem Feld von Tabellen im BIOS-ROM. Beim Hochfahren des PC wird eine Initialisierungs-Routine im VGA-BIOS aufgerufen, die außer der Initialisierung des Modus 03h u.a. auch einen Save-Table-Pointer an der Adresse 0000h:04A8h ablegt, der (wie nicht schwer zu erraten) auf eine Save-Table zeigt. Diese Save-Table ist ein Feld von weiteren Pointern deren erster auf den Anfang der Tabellen mit den Registerinhalten für die verschiedenen Video-Modi zeigt. Behufs dieser Information kommt man an die Initialisierungs-Werte für die Register der VGA heran. Der Offset der Werte für einen der Video-Modi 00h bis 13h (bzw. 11h bei der EGA-Karte) geht aus diesen beiden Arrays hervor.

```
const VGAOfs: array[0..$13] of Word =
          ($5C0,$5C0,$600,$600, $100,$140,$180,$640,
           $200,$240,$280,$2C0, $300,$340,$380,$440,
           $480,$680,$6C0,$700);
const EGAOfs: array[0..16] of Word =
          ($4C0,$500,$540,$580, $100,$140,$180,$1C0,
           $200,$240,$280,$2C0, $300,$340,$380,$440, $480);
```

Aufbau der Parameter-Tabelle eines Video Modus, je Modus 64 Bytes

			IndexAdr	DataAdr	Index
0	Byte	Anzahl Bildschirm-Textspalten	BIOS-Variablen		
1	Byte	Anzahl Bildschirm-Textzeilen			
2	Byte	Zeichenhöhe in Rasterzeilen			
3	Word	Speichergröße pro Bildseite			
		Sequenzer Register	3C4h	3C5h	
5	Byte	Clocking Mode Reg.			1
6	Byte	Map Mask Reg.			2
7	Byte	Character Generator Select Reg.			3
8	Byte	Memory Mode Reg.			4
		Allgemeine Register			
9	Byte	Miscellaneous Output Reg.	3C2h (W),	3CCh (R)	
		CRTC Register	3B4h/3D4h	3B5h/3D5h	
10	Byte	Horizontal Total Reg.			0
11	Byte	Horizontal Display End Reg.			1
12	Byte	Start Horizontal Blanking Reg.			2
13	Byte	End Horizontal Blanking Reg.			3
14	Byte	Start Horizontal Retrace Reg.			4
15	Byte	End Horizontal Retrace Reg.			5
16	Byte	Vertical Total Reg.			6

17	Byte	Overflow Reg.	7
18	Byte	Preset Row Scan Reg.	8
19	Byte	Maximum Scan Line Reg.	9
20	Byte	Cursor Start Reg.	10
21	Byte	Cursor End Reg.	11
22	Byte	Start Address High Reg.	12
23	Byte	Start Address Low Reg.	13
24	Byte	Cursor Location High Reg.	14
25	Byte	Cursor Location Low Reg.	15
26	Byte	Vertical Retrace Start Reg.	16
27	Byte	Vertical Retrace End Reg.	17
28	Byte	Vertical Display End Reg.	18
29	Byte	Offset Reg.	19
30	Byte	Underline Location Reg.	20
31	Byte	Start Vertical Blanking Reg.	21
32	Byte	End Vertical Blanking Reg.	22
33	Byte	Mode Control Reg.	23
34	Byte	Line Compare Reg.	24
		Attribute Controller Register 3C0h 3C0h(W),3C1h(R)	
35	Byte	Palette Reg. 0	0
36	Byte	Palette Reg. 1	1
37	Byte	Palette Reg. 2	2
38	Byte	Palette Reg. 3	3
39	Byte	Palette Reg. 4	4
40	Byte	Palette Reg. 5	5
41	Byte	Palette Reg. 6	6
42	Byte	Palette Reg. 7	7
43	Byte	Palette Reg. 8	8
44	Byte	Palette Reg. 9	9
45	Byte	Palette Reg. 10	10
46	Byte	Palette Reg. 11	11
47	Byte	Palette Reg. 12	12
48	Byte	Palette Reg. 13	13
49	Byte	Palette Reg. 15	14
50	Byte	Palette Reg. 16	15
51	Byte	Mode Control Reg.	16
52	Byte	Overscan Reg.	17
53	Byte	Color Plane Enable Reg.	18
54	Byte	Horizontal Pel Panning Reg.	19
		Graphics Controller Register 3CEh 3CFh	
55	Byte	Set/Reset Reg.	0
56	Byte	Enable Set/Reset Reg.	1
57	Byte	Color Compare Reg.	2
58	Byte	Data Rotate Reg.	3
59	Byte	Read Map Select Reg.	4
60	Byte	Mode Reg.	5
61	Byte	Miscellaneous Reg.	6
62	Byte	Color Don't Care Reg.	7
63	Byte	Bit Mask Reg.	8

Mit SAVEROMP wird ein Z80-Source-Listing ROMTABLE.ASM erzeugt, das per DEFB die Tabellen der Initialisierungdaten für die relevanten Video-Modi enthält.

Eine andere Strategie besteht darin, die verschiedenen Video-Modi der VGA durch das BIOS einstellen zu lassen und im jeweiligen Modus die Inhalte der Register auszulesen und diese in der Struktur der ROM-Tabelle in eine Datei zu schreiben. Das Programm SAVEREGS erzeugt auf diese Art das Sourcelisting REGTABLE.ASM.

Soft-Zeichensätze laden

In den Text-Modi gilt es außerdem noch, daß im Zeichengenetator-RAM (Speicherebene 2) ein passender Zeichensatz geladen ist. Zeichensätze in den Auflösungen 8*8, 8*14, 8*16, 9*14 und 9*16 liegen im VGA-ROM gebrauchsbereit vor. Bei der Darstellung von 9*14 und 9*16 Rasterpunkten pro Zeichen wird durch die Hardware nach je 8 Rasterpunkten entsprechend der Information aus dem Zeichensatz ein neunter Punkt in der Farbe des Hintergrunds erzeugt.

Den Anfang eines Zeichensatzes im ROM kann man sich über eine Funktion des VGA-BIOS-Interrupt 10h ebenfalls vom BIOS holen. Damit sind auch die Zeichensätze für andere Benutzer als einen 80X86 zugänglich. Man kann entweder mit GETVPARM sich ein Listing ZEIGER.ASM erzeugen, das alle Pointer auf Tabellen und Zeichensätze im VGA-BIOS (übersetzt in die Sicht das Z80 im Terminal) enthält, oder sich mit dem Programm SAVEFONT ein Listing VGAFONTS.ASM erzeugen, das die Zeichensätze selbst enthält.

Laden der Color-Lookup-Table

Damit ist auf dem Bildschirm aber noch nicht viel zu sehen, denn die Werte in den 16 Palettenregistern ergeben noch nicht direkt die auf dem Monitor dargestellte Farbe. diese 8 Bit Werte stellen Zeiger in die Color Lookup Table (CLUT) dar. die CLUT ist ein Register-Array von 256 3-Byte-Registern worin jeweils nur die Bits 0..5 von Bedeutung sind, es sind dies die Werte der Farbintensitäten (0..63) für die Farbkomponenten R, G und B.

Der lange Weg zum Zustandekommen einer Farbe auf dem Monitor ist dieser: Im Attribut-Byte eines Zeichens im Video-RAM kodieren die Bits 0..3 die Farb-Nr. des Textzeichens und die Bits 4..7 die Farb-Nr. des Hintergrunds. Für beide Komponenten wird aus dem Palettenregister mit der Farb-Nr. der Inhalt des Registers als Zeiger in die CLUT verwendet. Die dort vorliegenden Werte für R, G und B werden schließlich über den Video-DAC analog gewandelt und zum Monitor geschickt.

Die Standard-Inhalte der CLUT kann man sich auch wieder durch ein kleines Programm (SAVECLUT) vom BIOS einstellen lassen, auslesen und als Assembler-Source-Listing (FARBEN.ASM) auf die Diskette schreiben lassen.

Für dieses Problem habe ich noch keine Alternativ-Lösung gefunden, um die an versteckter Stelle im VGA-BIOS irgendwo vorliegenden Werte aus dem BIOS-ROM heraus direkt zu benutzen. Wenn das BIOS-ROM der VGA selbst im Terminal benutzt werden soll, muß FARBEN.ASM in das Betriebsprogramm des Terminals gelinkt werden.

Damit hätte man dann schon mal die Informationen, mit welchen Inhalten die Register der VGA im jeweiligen Video-Modus zu versorgen sind. Dabei ist zu bedenken, daß die Register 0..6 des CRTC erst beschrieben werden können, wenn im Register 11h D7=0 gesetzt wird. Der CRTC liegt in den Monochrommodi auf der Portadresse 3B4h/3B5h und in den Farbmodi bei 3D4h/3D5h. Monochrom und Farbe wird durch D0 des Miscellaneous Output Registers vorgegeben. Man kann sich dieses Bit in der Tabelle (Byte 9) ansehen.

Während der Programmierung der Register, die für das Video-Timing zuständig sind, sollte im Reset Register des Sequenzers (Index 0) immer 03h stehen. Anschließend muß dort 00h eingetragen werden, um den Resetzustand der VGA zu beenden.

Um zu zeigen, was bei der Initialisierung der Register nacheinander zu erfolgen hat, sei das Progamm VINITEST und darin insbesondere die Procedure write_regs(mode:byte) ein Anhaltspunkt. Sie ist in Z80-Assembler umzusetzen und in das Terminal-Betriebsprogramm anstelle der derzeitigen Initialisierung der Hercules-Karte einzumassieren. Als grundsätzlicher Initialisierungs-Video-Modus sollte Modus 03h verwendet werden.

Die 3-Digit-Portadressen des PC reduzieren sich beim Club-80-Terminal auf die niederwertigsten beiden Digits (A8 und A9 werden bei Portzugriffen auf die Video-Karte immer high-gesteuert, so daß automatisch das höchstwertige Digit '3' ausgegeben wird). Das hier auch erfolgende Update der BIOS-Variablen NrOfColumns bis PageSize braucht nicht in Z80-Assembler übertragen zu werden.

Es gibt zusätzlich zu den oben behandelten Registern noch einige weitere Register, die hier kurz einzeln vorgestellt werden sollen:

Feature Control Register: Port 3BAh/3DAh (wr), 3CAh (rd, nur VGA)

Dieses Register wird im PC sowohl bei EGA- als auch bei VGA-Karten in allen Videomodi mit 00h geladen.

Input Status #0 Register: Port 3C2h (nur Lesen)

D0..D33: unbenutzt
D4: 0= Dip-Schalter geschlossen
1= Dip-Schalter offen
Lesen eines von 4 Dip-Schaltern, D3 und D2 im Miscellaneous Output Reg. geben den Schalter vor
D5:* Logischer Pegel von Pin 19 des Feature Connectors
D6:* Logischer Pegel von Pin 17 des Feature Connectors
D7: 0= Kein Vertical Retrace Interrupt IRQ2 aufgetreten
1= Vertical Retrace Interrupt IRQ2 aufgetreten
* nur bei EGA

Input Status #1 Register: Port 3BAh/3DAh (nur Lesen)

D0: 1= Hor/Vert Retrace aktiv
D1: 1= Lichtgriffelimpuls aufgetreten
D2: 0= Lichtgriffelschalter gedrückt
D3: 1= Vertical Retrace (nicht horizontal)
D4,D5: gibt log. Pegel zweier Video-Signale wieder welche beiden Signale, gibt das Color Plane Enable Reg. des Attribute Controller vor

Reset Register des Sequenzers: Port 3C4h Index(:=0), Port 3C5h Data

D0,D1: 0,0: VGA-Hardware ist aktiv
1,1: Reset-Zustand, keine Signale zum Monitor

Color Select Register des Attribute Controllers: Port 3C0h Index (:=20), 3C1h Data

D0..D3: Farbregister Adreßbit 4..7
ist in allen Videomodi mit 00h zu befüllen

CGA- und Hercules-compatible Register

Diese Register lassen sich nur erreichen, wenn die Karte ein Umschalten auf Register-Kompatibilität zu diesen Karten erlaubt. Das Umschalten erfolgt durch Ausgabe von D0=0 (für CGA) und D0,D1=0 (für Hercules) an das Mode Control Register des CRTC und Bedröhnen der für die jeweilige Karte im folgenden aufgeführten Register mit den für diese Karte gängigen Werten.

6845-Register: Port 3B4h/3D4h (Index), 3B5h/3D5h (Daten)
Mode Control Register: Port 3B8h/3D8h
Status Register: Port 3BAh/3DAh
Configuration Register: Port 3BFh (nur Hercules)
Color Select Register: Port 3D9h (nur CGA)
Diese Register haben die gleichen Funktionen wie bei der Hercules-Karte bzw. der CGA-Karte. Da sie für die Hercules-Karte im Beitrag zum Club-80-Terminal bereits beschrieben wurden, wird hier kein weiterer Platz verschwendet.

Spezielle Register, die für die tägliche Arbeit wichtig sind

die Standardwerte für diese Register stehen in der Parameter-Table und werden durch InitVGA eingetragen

Map Mask Register des Sequenzers: Port 3C4h Index(:=2), 3C5h Data

D0..D3: 1= Schreiben in die RAM-Ebene 0..3 freigegeben

Character Map Select Register des Sequenzers:
Port 3C4h Index(:=3), 3C5h Data
D5(nur VGA), D1, D0: Nr des Blocks des primären Zeichensatz
D6(nur VGA), D3, D2: Nr des Blocks des sekundären Zeichens.

Die Zeichensätze werden in den Textmodi in der Speicherebene 2 in aufeinanderfolgenden 8K-Bereichen gespeichert. Ob das darzustellende Zeichen aus dem primären oder sekundären Zeichensatz zu nehmen ist, entscheidet das Attributbit 3. Damit lassen sich 512 verschieden Zeichen gleichzeitig darstellen. Die Anzahl der Farben, in denen ein Zeichen dargestellt werden kann, reduziert sich in diesem Modus auf 8.

Mode Control Register des CRTC: Port 3B4h/3D4h Index(:=23),
3B54h/3D5h Data
D0: 0= CPU-A13 := Rasterzeilenadreßbit 0
1= CPU-A13 wirkt direkt auf das Video-RAM
D1: 0= CPU-A14 := Rasterzeilenadreßbit 1
1= CPU-A14 wirkt direkt auf das Video-RAM
wenn D0=0, verhält sich die VGA wie eine CGA
wenn D0 und D1 = 0, verhält sich die VGA wie eine Hercules

D2..D6: sollte man in Ruhe lassen
D7: 0= CRTC-Reset, alle Register werden mit 0 geladen

Das Miscellaneous Register des Graphics Controllers:
Port 3CEh Index(:=6), 3CFh Data
D0: 0= Textmodus, 1= Grafikmodus
D1: 0= Adreßbit 0 unverändert siehe [1] Seite 392
1= Adreßbit 0 substituiert
D3,D2: 00= A0000..BFFFF Video-RAM Adresse
01= A0000..AFFFF ""
10= B0000..B7FFF "
11= B8000..BFFFF "
(dadurch erübrigt sich leider nicht der Hardpatch)

Das Index Register des Attribute Controllers:
Port 3C0h (wr), 3C1h (rd)
Wie bei CRTC u.s.w. wird hier in Bit 0..4 vorgegeben, auf welches Datenregister (3C0h wr, 3C1h rd) zugegriffen werden soll. Zusätzlich steuert D5, ob die CPU (D5=0) oder die Video-Hardware (D5=1) auf die Register zugreifen kann. D5 ist nach Verändern der Register immer auf 1 zu setzen.

Durch Lesen des Input Status #1 Registers wird auf das Indexregister für Schreibzugriffe geschaltet. beim nächsten Schreibzugriff wird dann das Datenregister angesprochen.

Die Register des Video-DAC:

Zugriffe auf den Video-DAC sollten nur während des Vertical Retrace stattfinden (D0,D3 des Input Status #1 Reg. high).
PEL Adress Read Mode Register: Port 3C7h (wr) 3C8h (rd, Wert+1)
PEL Adress Write Mode Register: Port 3C8h (rd/wr)
diese beiden Register sind Index Register mit unterschiedlichen Adressen für Lesen oder Schreiben des eigentlichen PEL Data Registers. Beim Lesen des PEL Address Read Mode Registers erhält man einen um 1 erhöhten Wert gegenüber dem vorher eingeschriebenen Wert.
PEL Data Register: Port 3C9h (rd/wr)
Durch dreimaliges Lesen/Schreiben dieses Registers erhält/schreibt man nacheinander die 6Bit-Farbwerte für Rot, Grün und Blau. Danach wird automatisch die Adresse im Index Register erhöht, sodaß sofort die nächsten 3 Farbwerte gelesen/geschrieben werden können.

DAC State Register: Port 3C7h (rd)
D1,D0: 00= Lesemodus aktiv, 11= Schreibmodus aktiv

PEL Mask Register: Port 3C6h (rd/wr)
8Bit AND-Maske für Farbregister-Adresse; damit lassen sich einzelne Bits, die von den Palettenregistern kommen, desaktivieren.

Das Laden der Zeichensätze

Die Adresse der Zeichensätze im BIOS-ROM läßt sich das Programm GETVPARM durch das BIOS selbst geben und trägt sie (umgerechnet auf die Adressen im Terminal) in VGA.ASM ein.

In den Textmodi beinhhaltet Video-Speicher eine lineare Folge von Bytepaaren (Zeichen und Attribut), wobei eine lineare Zuordnung der Bildschirmposition zur Adresse im Video-RAM besteht. Aus den ersten beiden Einträgen in der Parameter-Tabelle eines Modus können Zeilenlänge und Anzahl Bildschirmzeilen entnommen werden. Die Größe des für eine Seite benötigten Video-RAMs ergibt sich aus

2 * Zeichen pro Zeile * Zeilen pro Seite

Literatur:

[1] Matthias Uphoff: Die Programmierung der EGA/VGA Grafikkarte; Addison-Wesley; ISBN 3-89319-274-3
[2] Michael Tischer: PC Intern 3.0; Data Becker Verlag 1992; ISBN 3-89011-591-8
[3] Arno Fritz: Regenbogen im Eigenbau - Programmierung der 256--Color-Modi des ET3000; mc 8/91, S.120
[4] Michael Schulte: Heißer Farb_Ofen - Wie man aus dem ET4000--Chipsatz das Beste herausholt; mc 12/91, S.140
[5] Dirk Meiners: Farbe bekennen - Auswahl der Farbpaletten bei VGA-Karten; c't 4/91, S.330
[6] Peer Meier: Schöne bunte Welt, Hardwarenahe Programmierung von VGA-Karten; c't 4/91, S.338 und c't 5/91, S.292
[7] Detlef Grell: Kleine Extras - Video-Modi und Optionen bei Super-VGAs einstellen; c't 1/92, S.156

Das Thema mit seinem ursprünglich gedachten Zweck wird nur wenige Leute interessiere; die Pascal-Sources beinhalten aber alles, was zur direkten VGA-Programmierung nötig ist, und sind deshalb eventuell auch für PC-Programmierer interessant.

Damit in diesem Info noch Platz für weitere Beiträge bleibt, habe ich darauf verzichtet, die 110 KB Source-Listings der PC-Programme und die damit erzeugten Assembler-Sources für den Terminal-Z80 hier abzudrucken. Wer daran Interesse hat, schickt mir eine leere Diskette und 2,--DM in Briefmarken und erhält dann im PC-Format die Programme.

Da ich momentan keinen festen Wohnsitz habe (aus meiner alten Wohnung bin ich ausgezogen und meine neue ist noch nicht fertig), bitte ich, mich bei der Arbeit zu kontaktieren.

Inst. für Schadenforschung, Preetzer Str. 75, 24143 Kiel
Tel 0431/77578-20, Fax 0431/77578-99

Anmerkung der Redaktion:
Oben genannte Sorce-Listings sind auch in der **Discovery** eingelagert. Alle File der Diskette sind unter **VGA_C80T.ZIP** zusammengepackt.
Gleichzeitig kann eine Kopie der Diskette auch von mir geordert werden. Bitte sendet mir eine leere PC-Format-Diskette. Ihr erhaltet Sie dann mit dem INFO 44 kopiert zurück.

Club 80 Börse — Club 80 Börse

ANGEBOT

Wegen Hobbyaufgabe komplett abzugeben:

Genie III A (keine Grafik) mit drei Laufwerken à 720 KB
mit drei Betriebssystemen (CPM, Newdos,G-Dos)

\+ Genie III A - teildemontiert - mit noch einem Laufwerk

\+ ca. 80 Disketten mit Programmen
viele Utilities und Kleinprogramme (meist G-Dos)

\+ Wordstar, Multiplan,Turbo-Pascal,Cobol

\+ folgende Originalprogamme mit Original-Dokumentation:
Textverarbeitung Genie-Text
mit Datenbank Midabas und Kalkulation,
Geschäftsprogamm
mit Finanzbuchhaltung -Uwe Markus-
Visicalc von Radio-Shack

\+ Bücher, Dokumentationen u.s.w. -teils Fotokopie-
z.B. Programmierung des Z-80 (Rodnay Zaks)
Das Dos-Buch (Hartmut Grosser)
Vom Umgang mit CP/M (Bernd Pol)
Betriebssystem-Handbücher G-Dos, Newdos, CP/M
User-Manual und Techn. Manual Genie IIIa

+(evtl.) Zeitschriften - mc und chip - (mehrere für Z-80 -
Computer iteressante Jahrgänge (1981-1988,
unvollständig)

NUR ZUSAMMEN gegen DM 50.– (Schutzgebühr)
bei Selbstabholung abzugeben!

Kontakt: André Schut, Sanderstr. 26, 12047 Berlin Tel. 030 / 6 91 78 61
(wochentags von ca. 13-18 Uhr Tel. 030 / 7 87 51 92
Fax 030 / 7 87 51 91)

„. . . dann machte ich eine Vollbremsung, da ein Fußgänger bei Rot auf die Straße sprang, die Hose runterließ und seinen Hintern zeigte!“

■

„Die Geburtsurkunde des Kfz ist beigefügt.“

■

„Ich kam von der Straße ab, wobei ein Baum meinem Kfz nicht ausweichen konnte.“

■

„Die Ehefrau des zertrümmerten Autos wurde nur leicht, der Mann glücklicherweise nicht verletzt.“

■

„Bei einer abendlichen Fahrt mit meiner Ehefrau habe ich Wildschaden erlitten!“

■

„Der Pkw kam von der Fahrbahn ab, als er eine Cassette wechseln wollte.“

„Es war die Zeit des Stoßverkehrs. Ich stieß auf das vor mir zum Stehen gekommene Fahrzeug.“

■

„Herr P. fuhr plötzlich rückwärts und fuhr mich um. Dann fuhr er wieder vorwärts und fragte, was passiert sei.“

■

„Vier Wochen nach dem Unfall wurde der Gehgips am rechten Arm wieder entfernt.“

■

„Obwohl der Versicherte einen Seitensprung machte, kam nicht er, sondern seine Frau zu Schaden . . .“

■

„Ich holte die Polizei und ließ mir meine Unschuld bestätigen.“

■

„Der Unfall passierte, weil ich von einem Auto unbekannter Nationalität geblendet wurde.“

■

„Ich fuhr vorne auf, während es gleichzeitig hinten bumste!“

■

„Frl. B. näherte sich der Ampel und sprang von Gelb auf Grün um.“

■

„Dann durchbrach ich mit meinem Lkw die Schallmauer an der Autobahn.“

Die folgende Nachricht entstammt der Area CPM.GER aus dem FIDO-Netzes.Sie ist zwar nicht mehr ganz frisch, dürfte aber trotzdem noch aktuell sein. Vielleicht gibt es unter Helmuts Angeboten das eine oder andere, was ihr brauchen könnt.

(53) Mon 20 Dec 93 22.47
By: Helmut Jungkunz
To: All
Re: Cp/m Compiler
St:

@TID: IMAIL 1.40G2+ AK00021
Hi,
es gibt ein paar nette neue Sachen unter CP/M 80:

HITECH-C Compiler	- jetzt Freeware !!! K&R und ANSI C !
LPascal Compiler	- demnaechst erhaeltlich bei mir soll voll Turbo Pascal kompatibel sein kommerzielles Produkt
DRACO Compiler	- Zwischending zwischen C und Pascal SHAREWARE glaube ich

Non-Compiler:

REDABAS	- der Ost-Clone des bekannten Datenbank-standards laesst sich patchen damit er .DBF Dateien liest und .CMD Dateien abarbeitet. ERASE (Clearscreen) und MODI COMM funktionieren nicht korrekt. MURKSWARE wuerd ich das mal nennen.
PCFILE	- Ist glaube ich jetzt FREEWARE. Database-Programm.
FILEBASE	- Freeform Database. User supported Softw.
NZCOM	- Auto-Installversion des weltberuehmten ZCPR. Pluesch fuers CP/M.
Z3PLUS	- Auto-Installversion ZCPR fuer CP/M Plus.
JUGGLER128	- C128 only - Diskformat-Genie, auch AMSTRAD 3.5-Zoll-Formate (keine 3-Zoll!!) lesbar auf 1581 !!

Informationen und ne Katalogdiskette voll gibt's bei mir gegen Einsendung einer formatierten Leerdiskette mit Rueckporto. C128-User bitte KAYPRO IV oder (Jugglerformat) MG1581 benutzen!

Schoene Methode ist, zwei Ruecketiketten beizufuegen, dann geht das Handling schneller. (Eins mit mir als Absender und eins mit Dir als Empfaenger).

Helmut Jungkunz, Zacherlstr. 14,85737 Ismaning,089/969374 18:30 - 21:00 oder Anrufbeantworter. Wenn ich nicht da bin, sagt mindestens Euren Namen und Eure Telefonnummer, ich rufe verlaesslich zurueck!

--- SuperQWK 1.17-2 (Reg)
 * Origin: Harry's GRAFFITI BOX Munich * V32B-V42B * (2:246/115)

@PATH: 246/115 56 76 55 241/7922

Buchbesprechung:

Mailbox-Führer '94

Vom Markt & Technik Verlag in Haar bei München erschien soeben eine Neuauflage des 1991 erstmals veröffentlichten Mailbox-Führers zum Preis von DM 29,80.

Besonders Neulinge in der DFÜ, die nächtelang auf Entdeckungsreisen durch das Telephonnetz sind, erleben ein böses Erwachen spätestens dann, wenn ihnen die TELEKOM dafür die Rechnung präsentiert. Dann macht sich eine Übersicht, die man in aller Ruhe "off-line" durchblättern kann, sehr schnell bezahlt; besonders, wenn in einem solchen Führer nicht nur steht, wo sich Mailboxen befinden, sondern auch, welche speziellen Angebote und Informationen dort anzutreffen sind.

Diesem Anspruch wird der vorliegende Führer auf insgesamt über 260 Seiten gerecht. Gegenüber der früheren Ausgabe wurden mehr als 50 deutschsprachige Mailboxen neu aufgenommen, wobei man Österreicher und Schweizer allerdings noch vermisst. Die Aufnahmekriterien sind: Zugriffsmöglichkeit für jedermann, 24 Stunden tägliche On-Line-Zeit und Betriebsbereitschaft seit mindestens einem Jahr. Leute mit Mailbox-Erfahrung wissen das zu schätzen, denn viele Boxen - insbesondere die kostenlosen - sind genau so schnell aus der Szene wieder verschwunden, wie sie vorher auftauchten. Den Betreiber verlässt die anfängliche Begeisterung, sobald er merkt, welche Arbeit für die Pflege einer Box täglich aufgewendet werden muss und welche Kosten entstehen, wenn man dafür nicht nur einen leistungsfähigen Computer abstellen muss, mit dem man dann nichts anderes mehr anfangen kann, sondern dieser auch ganz schön Strom braucht, wenn er Tag und Nacht in Betrieb ist, vom erforderlichen Telephonanschluss ganz zu schweigen.

Die Informationen werden im Mailbox-Führer sehr übersichtlich dargeboten: Für jede Box mindestens eine separate Seite. Ausser den jeweils möglichen Übertragungsgeschwindigkeiten wird z. B. auch angegeben, wie man als Neuer Zugang zu den verschiedenen Boxen bekommen kann und wie hoch die Mailbox-Gebühren gegebenenfalls sind.

Aus dem Inhaltsverzeichnis ist auch ersichtlich, dass in dem Führer eine ganze Reihe interessanter Themen rund um die DFÜ behandelt werden. Das "Lexikon der DFÜ" in Kapitel 4 ist allerdings etwas oberflächlich geraten. Aus dem selben Verlag stammt auch "Das grosse Computer-Lexikon", mit über 6000 Stichworten eines der umfangreichsten (und trotzdem preisgünstigsten) auf dem deutschen Markt. Bei gegenseitiger Abstimmung hätte man also leicht Überschneidungen vermeiden und echte DFÜ-Begriffe ausführlicher erläutern können. Weiteren Platz dazu hätte man durch Verzicht auf so Spässe wie den "Pangalaktischen Donnergurgler" gewonnen.

Für künftige Auflagen des Mailbox-Führers gibt es also noch Verbesserungsmöglichkeiten. Und wer seine Lieblings-Box im Führer nicht gefunden hat, kann seinem Sysop ja eine Kopie des Meldeformulars aus dem Anhang zusenden und darauf hoffen, dass die nächste Ausgabe weiter ergänzt und verbessert wird. (GWB)

INHALTSVERZEICHNIS

VORWORT

KAPITEL 1
1.1 Gewußt wie – der erste Mailbox-Kontakt 9
1.2 Akustikkoppler oder Modem – Entscheidungshilfen 12
1.3 Terminalprogramme 14
1.4 Tips zum Sparen von Telefongebühren 15
1.5 Errichten einer eigenen Mailbox 17

KAPITEL 2
2.1 Zugriff auf Mailboxen 19
2.2 Datex-J, Btx 20
2.3 Datex-P – die NUI als Schlüssel 25

3
3.1 Mailboxnetze 37
3.2 Konfigurieren von Modems 44
3.3 Die wichtigsten AT-Befehle im Überblick 49
3.4 Wichtige Standard-Mailboxbefehle 51

EL 4
4.1 Lexikon der DFÜ 55
4.2 Die RS232/V.24-Schnittstelle 71
4.3 Definition der Abkürzungen und Funktionen der RS232C 73
4.4 Verbindungsbeispiele 75

KAPITEL 5
5.1 Die Top 150 – deutsche Mailboxen 77
5.2 Datex-P-Mailboxen 249

ANHANG A
A.1 Kriterien für die Aufnahme von Mailboxen in den Mailboxführer 251

ANHANG B
B.1 Mailboxen – alphabetisch sortiert 253
B.2 Mailboxen – nach Vorwahlnummern sortiert 257
B.3 Mailboxen – über ISDN zu erreichen 261

Impressum

1. Vorsitzender:	**Hartmut Obermann** Wilhelm-Baur-Straße 8 Postfach	Tel.: 07 21/ 85 40 68 BTX: 07 21/ FAX: 07 21/
76135	**Karlsruhe**	
2. Vorsitzender:	**Gerald Schröder** Am Schützenplatz 14	Tel.: 0 41 05/ 26 02
21218	**Seevetal**	
Hardwarekoordinator:	**Andreas Magnus** Bismarckstraße 29	Tel.: 02 09/ 87 02 30
45879	**Gelsenkirchen**	
NewDOS-Diskothekar:	**Oliver Volz** Am Ochsenwald 37A	Tel.: 07 11/ 74 40 51
70565	**Stuttgart** (Rohrerhöhe)	
CP/M-Diskothekar:	**Fritz Chwolka** Saarstraße 34	Tel.: 0 24 64/ 89 20
52457	**Aldenhoven**	
C-128-Diskothekar:	**Günther W. Braun** Postfach 80 02 26	
81602	**München**	
Clubbücherei:	**Kurt Müller** Sophie-Scholl-Ring 3b	Tel.: 0 41 52/ 7 06 43
21502	**Geesthacht**	
Redaktion:	**Jens Neueder** Gschlachtenbretzingen Rudolf-Then-Straße 32	Tel.: 07 91/ 4 28 77 BTX: 07 91/ 44 47 22 FAX: 0 79 71/ 2 50 55
74544	**Michelbach / Bilz**	
Bankverbindung:	**Club 80** Postgiro Sonderkonto CLUB 80 Obermann H., 8870 Günzburg	**Postgiroamt Frankfurt** **BLZ: 500 100 60** **Kto.Nr.: 496 071 - 605**
Autoren:	**Die Redaktion bedankt sich bei den im Inhaltsverzeichnis genannten Autoren für die Mitarbeit an der Club-INFO. Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.**	

Schluß

Hallo Club 80'er,

diesmal ist unser INFO wieder etwas schmäler ausgefallen, auch kommt es später als geplant. Zuerst hatte ich noch etwas zugewartet, da doch noch einige "angedrohte" Artikel ausstehen. Danach kamen mir privat einige wichtige Termine dazwischen, so daß ich das INFO jetzt erst fertigstellen konnte. Ich hoffe, daß es Euch trotzdem gefällt.

Ich möchte hier auch wieder mal die Bitte nach Futter für das INFO aussprechen. Ihr habt doch sicher schon den einen oder anderen Artikel für mich bereit?

Als nächster wichtiger Termin in unserem Clubleben steht vom 12. bis 15.05.94 unsere Jahreshauptversammlung an. Sicher haben sich schon fast alle beim Alexander angemeldet. Wer sich jetzt noch kurzfristig dafür entscheiden kann sollte das Anmelden schleunigst nachholen. Ich persönlich freue mich schon auf das gesellige Wiedersehen. Sicher gibt es wieder viele neue Erfahrungen, die ausgetauscht werden möchten.

Bis zum Clubtreffen
oder zum nächsten INFO
Euer

Jens

Adressen-Liste CLUB80

Stand vom 14-04-1994
Seite 1

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat Tel. geschäftl.	Telefax privat FAX geschäftl.	BTX FIDO-Node	Mailboxname Mailboxnummer
1	Barendt	Harry	Hermann-Löns-Straße 7	GER 50181	Breedburg (Erft)	02272/7168	-	-	-
						-	-	-	-
2	Berndt-Jochum	Ilse	Stachelsgut 24	GER 51427	Bergisch Gladbach	02204/ 65254	-	-	-
	IBM, GENIE III, GENIE IIIs, SHARP Pocket 1600■ Scanner					02204/ 65254	-	-	-
3	Bernhardt	Helmut	Preetzer Straße 75	GER 24143	Kiel	-	-	-	-
	IBM, Prof 180, CPU 280■ diverse PCs,Novellite-Netz,1496E,Mustek105+					0431/ 77578-20	0431/ 77578-99	2:242/262.26	-
4	Bielenberg	Georg	Erikaweg 1	GER 24568	Kaltenkirchen	04191/ 3751	-	-	-
	Atari 260ST, C128, Schneider Joyce ■ Modem: LC2496 Digitech, Scanner: Supersc.III					04193/ 90430	-	-	-
5	Brans	Jörg	Tieloh 55	GER 22307	Hamburg	040/ 6906531	-	-	-
	IBM■ CD-Rom, Streamer, Soundkarte					-	-	-	-
6	Braun	Günther W.	Postfach 80 02 26	GER 81602	München	-	-	-	Discovery
	Commodore 128 D■ Akustikkoppler					-	-	-	07127/ 70107
7	Braun	Harald	Postfach 8011	GER 24154	Kiel	0431/35139	-	-	-
						-	-	-	-
8	Böckling	Ulrich	Juchaczstraße 61	GER 56203	Höhr-Grenzhausen	02624/ 4861	-	-	-
	IBM 386DX33,1040ST,VC20,C64,TRS80 M I,ZX81 ■ Modem, Videodat-, Videotextdecoder					02631/ 895168	-	-	-
9	Chwolka	Fritz	Saarstraße 34	GER 52457	Aldenhoven	02464/ 8920	-	-	-
	IBM 386+Co, Commodore, Apple, Z 280■ Modem 2.4					-	-	2:248/242:8	-
10	Dose	Volker	Dorfstraße 10	GER 24235	Brodersdorf	04343/ 1357	-	-	-
	GENIE IIIs mit Z180■ EPROMer, Modem					-	-	-	-
11	Halgasch	Gert	Großschönauer Straße 26	GER 02796	Jonsdorf	035844/ 636	-	-	-
	IBM 386DX■					-	-	-	-
12	Hartmann	Hans-Günther	Möwenstraße 9	GER 27804	Berne	04406/ 6911	04406/ 1071	-	-
	IBM 386SX, TANDY M4p, Z280-Kartenrechner■ Real Time Clock, Speed Up Kit 6,3MHz, 40MB-P1					0421/ 248-2419	-	2:240/300.24	-
13	Hebecker	Ulrich	Büsnauer Straße 15	GER 70563	Stuttgart	0711/ 734800	-	-	-
	IBM 286+386, 128D, 1581, Kaypro484■					-	-	-	-
14	Held	Manfred	Stirner Straße 22	GER 91785	Pleinfeld	09144/ 6563	09144/ 8514	-	-
	IBM■ Modem ZyXEL 1496+, CD-ROM					0911/ 219-2245	-	2:2400/10.10	-
15	Hermann	Klaus	Forchenstraße 8	GER 72124	Pliezhausen	07127/ 71945	-	-	Discovery
	IBM 386■ ET 4000, Modem					-	-	2:2407/70.740	07127/70107
16	Hürdler	Manfred	Niederhoferstraße 29	GER 97222	Rimpar	09365/ 4235	-	-	-
	Victor Sirius 1 (IBM), CPC 6128■					-	-	-	-
17	Johnen	Willi	Hansemannstraße 1	GER 52351	Düren	02421/ 501305	-	-	-
	GENIE IIIs■					02421/ 33064	-	-	-
18	Kauka	Dietmar	Straße des Friedens 37	GER 04552	Neukirchen (Borna/Leipz	03433/851019	-	-	-
						-	-	-	-
19	Kemmer	Jürgen	Dorfberg 7	GER 97232	Sulzdorf	09334/ 1050	-	-	-
	IBM 386, UltraSound, ZyXEL U1496E+■ DCF-77-Empfänger parallel, IR-Sender seriell					-	-	2:247/2086.1	-
20	Kleespies	Andreas	Mainzer-Land-Straße 765	GER 65934	Frankfurt /Main	069/ 387432	-	-	-
						-	-	-	-
21	Kuhn	Eckehard	Im Dorf 14	GER 72636	Frickenhausen	07022/ 45417	-	-	-
	Atari ST 1040, TRS80 M I■					-	-	-	-
22	Linder	Jörg	Küstriner Str. 68	GER 15306	Seelow	03346/ 520	-	-	-
	KC 85/4 mit Floppy■					-	-	-	-
23	Lorenz	Walter	Mahräckerstraße 9	GER 60431	Frankfurt /Main	069/ 531656	-	-	-
	IBM 286/486, Z80-, HD 64180-Eigebau, Z280-T.R■ Scanner, Soundkarte, ET 4000					-	-	-	-
24	Magnus	Andreas	Bismarckstraße 29	GER 45879	Gelsenkirchen	0209/ 144029	-	-	-
	IBM 386Dx, GENIE IIIs■ Modem 2400					-	-	-	-
25	Mahlert	Herbert	Hohenbudbergerstraße 112 A	GER 47229	Duisburg	02065/ 47217	-	-	-
	IBM, GENIE I■ c't Videotext-Karte, Vobis Videodat-Decoder					02065/ 902592	-	-	-

7445

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
26	Messerschmidt	Kurt	Hanns-Eisler-Straße 54	GER 10409	Berlin	-	-	-	-
						-	-	-	-
27	Müller	Kurt	Sophie-Scholl-Ring 3B	GER 21502	Geesthacht	04152/ 70643	-	-	-
	IBM, Atari Mega ST4 ■ ATONCE 386,OverScan,Scanner,HBS640-T36,Modem					040/ 89983403	-	-	-
28	Mössel	Franz	Schafferstraße 12	I 39012	Meran	0039-473/34178	-	-	-
	IBM, Joyce, Workemate-Bullet■ Modem					0039-471/980496	-	2:333/400	-
29	Neueder	Jens	Rudolf-Then-Straße 32	GER 74544	Michelbach / Bilz	0791/ 42877	-	791444722	CCWN
	IBM 286/386/486, Atari ST 1040, TRS80 MI■ Modem, Sound Galaxi NX, Scanner, Tape, Borsu					07971/ 250-50	07971/ 250-55	1001667	0715168434
30	Neumann	Christof	Zeitblomstraße 22/2	GER 89077	Ulm /Donau	0731/ 6022568	-	-	-
	IBM, Tandy MII, Tandy M4p■ Novell					0731/ 9749720	-	-	-
31	Obermann	Hartmut	Wilhelm-Baur-Straße 8	GER 76135	Karlsruhe	0721/ 854068	-	-	SYNREL
	IBM486, Tandy M4p, Epson PX-8 ■ Modem, Scanner					-	-	2:241/7922.10	08282/ 4311
32	Retzlaff	Bernd	Kleiner Sand 98	GER 25436	Uetersen	04122/ 43551	-	-	-
	IBM 386, C64, GENIE I■					04103/ 605310	-	-	-
33	Rinio	Gerd	Rennbahnstraße 9	GER 22111	Hamburg	040/6552630	-	-	-
	IBM 486DX66/2, RTS 80, TRS 80 RS ■ Modem					-	-	-	-
34	Ruschinski	Claus	Pommernstraße 21	GER 45770	Marl	02365/ 34646	-	-	-
	IBM 386, TRS80 M I■ Highscreen-Scanner, CoProz IIT387					-	-	-	-
35	Schimmer	Jörg	Stettinerstraße 28	GER 60388	Frankfurt	06109/ 35336	-	-	-
	IBM 486, Schneider CPC■ Modem CSR2400					069/ 3800-2385	-	2:249/70.9	-
36	Schmid	Alexander	Entmannsdorf 5	GER 96317	Kronach /Gehülz	09261/ 53496	-	-	-
	GENIE IIs, GENIE IIIs, CPU 280■ Modem 2400,Prommer80,Ramdisk,Club80Terminal					-	-	2:2400/830	-
37	Schmitz	Rainer	Küferweg 12/1	GER 73099	Adelberg	07166/ 1397	-	-	Discovery
	IBM,Portfolio,NCR Decision Mate V,Joyce,... ■ Modem 1200, Märklin Digital Interface 6050					07161/ 608-475	-	-	07127/ 70107
38	Schoberth	Uwe	Petrus-Waldus-Straße 14	GER 75443	Oetisheim	07041/ 7254	-	-	Discovery
	Alphatronic P3■					0711/ 89394500	0711/ 89394513	-	07127/ 70107
39	Scholz	Hans-Werner	Spitalstraße 54	GER 41334	Nettetal	02157/ 3613	-	-	-
	IBM 386, Prof 80, ITT 3030■ Prommer80					-	-	-	-
40	Schroers	Horst-Dieter	Breslauer Straße 9	GER 85622	Feldkirchen	089/ 9032615	089/ 9043413	-	-
	IBM■ Modem, Scanner					-	-	-	-
41	Schröder	Gerald	Arminiusstraße 2	GER 22525	Hamburg	040/ 8507131	-	-	-
	IBM 386SX, Atari 1040STF, Z280■					040/ 54715334	-	-	-
42	Schröer	Egbert	Joachimstraße 18	GER 46284	Dorsten	02362/ 75311	-	-	-
	Portfolio, TRS80 MI, GENIE I, GENIE IIIs■					02362/ 49-9649	-	-	-
43	Schulte	Hartmut	Entenschnabel 8	GER 31311	Uetze	05173/ 1248	05173/ 24631	-	-
	IBM, Z280 u.a.■ Scanner					-	-	-	-
44	Sonnemann	Harald	In den Eckwiesen 9	GER 64405	Fischbachtal	06166/ 8512	-	-	-
	NDR-Klein ■ parallel/seriell, EPROMer					06151/ 92-1265	-	-	-
45	Stumpferl	Stefan	Hasenberglstraße 57	GER 80933	München	089/ 3138193	//0893144001//	-	-
	Amstrad CPC 6128+, GENIE IIIs ■ Modem, 20MB-Wechselplatte, PSG&PIO,SIO, ...					-	-	-	-
46	Sörensen	Rüdiger	Wiesbadener Str. 28B	GER 55252	Mainz-Kastel	06134/65342	-	-	-
						-	-	-	-
47	Tornow	Wilhelm	Elbblick 46	GER 21629	Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
	IBM 386DX, Atari Mega ST4, Tandy M4p■ Streamer,Soundblaster,Modem 2400,F.A.K.S. 910					-	-	-	-
48	Vogl	Michael	Postfach 10 06 10	GER 41490	Grefenbroich /Laach	02181/45112	-	0218145112-0001	-
	IBM, Amstarad CPC 464/6128■ Modem Gigitek 2400, Scanner Dat F.CPC					-	-	-	-
49	Volkmer	Richart	Am Spörkel 69	GER 44227	Dortmund	0231/ 752574	-	-	-
	IBM XT, Apple LC, CPC 6128, Osborne, TandyMII■ Modem 2400, ScanMan-Scanner					-	-	-	-
50	Volz	Oliver	Am Ochsenwald 37A	GER 70565	Stuttgart (Rohrerhöhe)	0711/ 744051	-	-	-
	IBM, GENIE IIs■ Modem Avantec					0711/ 685-3013	-	-	-

Adressen-Liste CLUB80

Stand vom 14-04-1994

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat Tel. geschäftl.	Telefax privat FAX geschäftl.	BTX FIDO-Node	Mailboxname Mailboxnummer
51	Walter	Hans-Jürgen	Elssholzstraße 9	GER 10781	Berlin	- -	- -	- -	- -
52	Werner IBM 286/486▪ Modem 2400, Scanner	Heiko	Reichenberger Straße 5	GER 01129	Dresden	0351/ 4608612 -	- -	- -	- -

Bitte überprüft Eure Daten
und teilt uns Änderungen mit!

HARDWARE * SOFTWARE

COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80

ERFAHRUNGSAUSTAUSCH

CLUBINFO

44. Ausgabe

Kontaktadresse:

Club 80

Hartmut Obermann

Wilhelm-Baur-Straße 8

Postfach _____

76135 Karlsruhe

Tel.: 0721/85 40 68

BTX: 0721/______

Anrufbeantworter &

FAX: 0721/85 40 68

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Protokoll der Hauptversammlung vom 14.05.94	1 - 2
	Gerald Schröder
Neues vom Vorstand	3 - 4
	Hartmut Obermann
10 Jahre Club 80 -eine kleine Club-Historie	5 - 6
Termine	6
	Jens Neueder
Mitgliedvorstellung	7
	Michael Kalb

Software

Grafik des Genie IIIs unter Holte CP/M+	9 - 19
	Volker Dose, Egbert Schröer
Assembler-Tip	21
Textumsetzung die N + erste	22 - 28
Nachtrag und Erweiterung zum PCXPUT.C	29 - 33
Nachtrag und Erweiterung zum PCXSHOW.C	34 - 39
	Alexander Schmid

Club 80 Börse

Suche ...	20
	Michael Vogl

Sonstiges

Model-1 Emulator für Dosen	40 - 42
	Alexander Schmid

Die letzten Seiten

Impressum	43
Schluß	44
	Redaktion
Mitgliederadressenliste	am INFO-Ende
Fragebogen DFÜ	am INFO-Ende
Bilder vom Club-Treffen 94	am INFO-Ende

Protokoll der Hauptversammlung vom 14. Mai 1994

von Gerald Schröder

Vom 12. bis 15. Mai trafen sich die Aktiven des Club 80 und einige Gäste zur Hauptversammlung 1994, diesmal organisiert vom Alexander Schmid und daher in Kornach stattfindend. Anwesend waren vom Club 80 (teilweise traten die Mitglieder erst während des Treffens dem Club bei):

Helmut Bernhardt
Günther W. Braun
Fritz Chwolka
Manfred Held
Michael Kalb
Jürgen Kemmer
Jens Neueder
Hartmut Obermann
Jürgen Peters
Alexander Schmid
Rainer Schmitz
Uwe Schoberth
Gerald Schröder
Ulrich Staimer.

Als Gäste durften wir begrüßen:

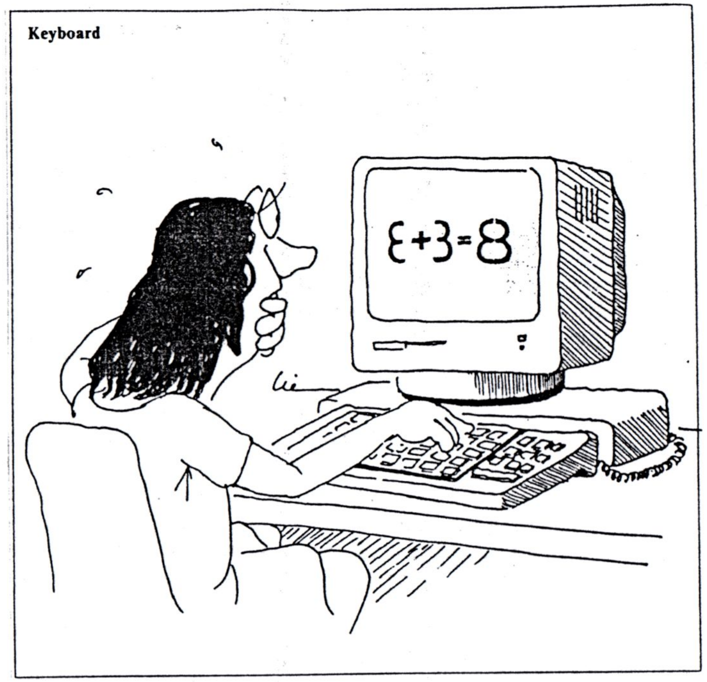
Helmut Jungkunz
Andreas Kleespies
Tilmann Reh
Claudio Romanazzi.

Es ließ sich kein TRS-80 oder kompatibler Rechner mehr blicken, dafür überwogen die CP/M-Kisten gegenüber den MS-DOSen, die nur wegen der Farbe und der Lautstärke auffielen.

Im (kurzen) offiziellen Teil wurden folgende Themen abgehandelt:

1. Die Clubbibliothek, aus der seit mindestens einem Jahr nichts mehr ausgeliehen wurde, wird von Kurt Müller zu Rainer Schmitz transferiert. Dieser wird sie ein Jahr lang hüten, eine aktuelle Bücherliste ins Info setzen und auf Ausleiher warten. Wenn niemand Bücher ausleiht, wird die Bibliothek aufgelöst und die Bücher werden meistbietend versteigert. Damit soll erreicht werden, daß die Bücher im Club verfügbar bleiben, da sie größtenteils nicht mehr zu kaufen sind, und trotzdem einer sinnvollen Nutzung zugeführt werden.

2. Hartmut machte den Vorschlag, einige Modems auf Clubkosten anzuschaffen und an die Mitglieder zu verleihen, damit diese mal in die weite Welt der freien Netze und Mailboxen schnuppern können. Vorher wird Hartmut herauszufinden versuchen, wie groß das Interesse der Mitglieder an der Ausleihe, an Schulungen und Installationshilfe ist. In diesem Zusammenhang hofft der Club, daß die Clubecke in der DISCOVERY-Mailbox von den DFÜlern auch mal schreibender- und nicht nur lesenderweise frequentiert wird.



3. Die Kassenprüfung von Rainer Schmitz und Uwe Schoberth förderte keine Unregelmäßigkeiten zu Tage, wofür wir uns wieder bei Jutta Obermann bedanken. Aufgrund des Kassenstands von 2459 DM und jährlichen Ausgaben von ca. 700 DM für das Info können wir weiterhin ohne Clubbeitrag auskommen, ausgenommen die Aufnahmegebühr.

4. Mangels neuer Bewerber mußte der alte Vorstand auf Clubkosten abgefüllt und dann wiedergewählt werden.

5. Jens Neueder klagte wieder über Artikelmangel. Daraufhin wurde nochmal bekräftigt, daß alle Themen interessant sind und abgedruckt werden, ausdrücklich auch Artikel zu IBM/MS-DOS. Allerdings wird der Club nicht seinen Namen und seine Ziele ändern, wie von Günther W. Braun vorgeschlagen, um nicht zu einem reinen MS-DOS-Club zu werden.

6. Weiterhin wollen wir wenigstens die Erinnerung an die alten Z80-Kisten wachhalten. Allerdings wurde angeregt, der inhaltlichen Verschiebung des Clubs durch ein neues Logo Rechnung zu tragen. Daher wird ein Wettbewerb zur Gestaltung eines neuen Club-Logos aufgelegt.

Gerald Schröder

Neues vom Vorstand

Liebe CLUB 80-Mitglieder,

wie Ihr schon dem Protokoll der Jahreshauptversammlung entnehmen konntet, gibt es diesmal einiges neues vom Vorstand zu berichten. Also frisch ans Werk!

1. DFÜ im CLUB 80

Leider ist die CLUB 80-Ecke in Klaus Hermanns DISCOVERY-Mailbox nicht sehr gut besucht. Da es recht frustrierend ist festzustellen, daß schon wieder mal keine neuen Mails eingegangen sind, finde selbst ich mich dort nur noch äußerst selten ein.

Zunächst verwunderte mich das etwas, da unter Betreibern von MSDOSen die Datenfernübertragung inzwischen fast schon zur Selbstverständlichkeit geworden ist. Nach einigen Gesprächen mit Besitzern von CP/M- und Z80-Maschinen wurde die Ursache für die mangelnde Beteiligung an dieser modernen Form der Kommunikation sichtbar.

Während es für MSDOSen sehr einfach ist, ein passendes Modem zu beschaffen, haben CP/Mler da teilweise erhebliche Probleme. Dies liegt vor allem daran, daß sich Grundfragen des Modembetriebs für CP/M- und Z80-Rechner nicht pauschal beantworten lassen. Dies wiederum hängt mit der oftmals sehr unterschiedlichen Hardware der genutzen Rechner zusammen.

Viele dieser Fragen ließen sich klären, wenn man mal ein Modem zum Testen zur Verfügung hätte. Aber wer möchte schon gerne ein paar Hunderter in den Kauf eines Modems investieren, welches hinterher mit dem eigenen Rechner garnicht oder nicht richtig zusammenarbeitet? Sicherlich niemand!

Genau hier möchte der CLUB 80 seinen Mitgliedern zur Seite stehen. Dazu sollen einige Modems angeschafft werden, die sich die Mitglieder dann für einen begrenzten Zeitraum und gegen Hinterlegung des Kaufpreises als Verrechnungsscheck zum Testen ausleihen können. Zusätzlich soll der Ausleiher Hilfe bei der Installation und seinen ersten Gehversuchen im DFÜ-Dschungel bekommen.

Um festzustellen, ob für einen solchen Service tatsächlich Bedarf besteht, muß ich mal wieder eine Fragebogenaktion starten. Da das Erscheinen dieses Heftes gerade in die Urlaubszeit fällt, wird als Einsendeschluß für den Fragebogen der 31. Oktober festgelegt. Das sollte eigentlich reichen, um allen Interessenten die Möglichkeit zu geben, sich zu melden. Bei entsprechendem Bedarf wird dann entschieden, welche und wieviele Geräte angeschafft werden.

Spätestens zu Beginn nächsten Jahres soll die Aktion, entsprechendes Interesse vorausgesetzt, dann anlaufen. Ich würde mich über eine rege Beteiligung freuen!

2. CLUB 80-Logo

Bei der Hauptversammlung wurde angeregt, dem CLUB 80 doch ein etwas moderneres Logo zu verpassen. Dabei sollte vor allem die Tatsache nicht zu kurz kommen, daß der CLUB 80 inzwischen nicht mehr von den Tandy TRS 80-Rechner, sondern von CP/M-Maschinen sehr unterschiedlicher Bauart dominiert wird.

Da schon das bisher verwendete Logo des CLUB 80 im Rahmen eines Ideenwettbewerbs entstand, wollen wir auch diesmal so verfahren. Hiermit sind also alle CLUB 80-Mitglieder aufgerufen, ihre Kreativität spielen zu lassen. Die Einsendungen sollten folgende Bedingungen erfüllen:
1. Ausführung in schwarz/weiß,
2. nicht größer als DIN A5,
3. geistiges Eigentum des Einsenders,

Eure Vorschläge für das neue CLUB 80-Logo solltet ihr an meine Adresse (findet ihr im Impressum) schicken. Einsendeschluß ist der 31. Dezember 1994. Eure Einsendungen werden alle im ersten nach dem Einsendeschluß erscheinenden Info veröffentlicht. Die Entscheidung, welches das neue CLUB 80-Logo wird, trifft die nächste Hauptversammlung.

Selbstverständlich soll sich das Mitmachen bei dem Ideenwettbewerb lohnen. Nach dem Motto "Ohne Preis keinen Fleiß", wird hiermit ein postzugelassenes 2400 Baud-Modem als Preis für den Gewinner des Wettbewerbs ausgesetzt. Dann zeichnet mal schön!

3. CLUB 80-Bibliothek

Die Büchersammlung des CLUB 80 ist leider eines unserer Sorgenkinder. Kurt Müller, der die Bibliothek seit einigen Jahren unter seinen Fittichen hat, klagt über zu geringe Ausleihraten. Im letzten Jahr wurde nicht ein einziges Buch entliehen.

An der Attraktivität der Bücher kann es eigentlich nicht liegen. Teilweise handelt es sich schon um richtige Raritäten, die auf dem freien Markt schon lange nicht mehr erhältlich und entsprechend gefragt sind.

Kurt Müller wird die Bibliothek in den nächsten Tagen an Rainer Schmitz abgeben. Dieser wird versuchen, das Interesse an der Bibliothek durch die eine oder andere Buchvorstellung nochmals anzuregen. Wenn nach dieser Aktion immer noch keine Bücher ausgeliehen werden, werden sie zur Versteigerung freigegeben.

Letzteres würde mir sehr leid tun, da damit eine Einrichtung, die der CLUB 80 für seine Mitglieder betreibt, unwiederbringlich verloren ginge.

4. Info-Beiträge

Ein anderes Sorgenkind des CLUB 80-Vorstands ist das Info, von dem Ihr inzwischen immerhin die vierundvierzigste Ausgabe in der Hand haltet. Vielleicht wird es statt eines Infos demnächst nur noch Vorstandsrundschreiben geben. Das wird dann der Fall sein, wenn die einzigen Beiträge, die zur Veröffentlichung kommen, die sind, in denen Jens oder ich Euch zu mehr Mitarbeit auffordern.

Wenn das Info stirbt, wird es keinen CLUB 80 mehr geben! Und wenn Ihr nicht aktiv am Info mitarbeitet, wird es bald sterben!

In der Hoffnung, demnächst mal ein bißchen was von Euch zu lesen,

10 Jahre Club 80 - eine kleine Club-Historie

von Gerald Schröder

Hartmut stellte im Vorfeld der diesjährigen Hauptversammlung fest, daß der Club 80 in diesem Jahr sein 10-jähriges Jubiläum feiert. Also wurde ich beauftragt, die Clubhistorie anhand der alten Infos und unseres teilweise lückenhaften Gedächtnisses zu rekonstruieren.

Der Club 80 wurde im März 1984 von Günther Wagner gegründet. Nach einigen Veröffentlichungen der Adresse in Zeitschriften wie der CHIP strömten die Mitglieder scharenweise dem Club zu. Mitglied Nr. 2 war Hartmut Obermann, Nr. 3 Manfred Held, als Nr. 4 traf Jens Neueder ein, die Nr. 9 bildete Gerald Schröder. Von dem ersten Dutzend Mitglieder (Stand Juli 84) sind nur noch diese vier im Club.

Anfang 1985 war der Club auf 45 Mitglieder gewachsen, der Höchststand mit 70 Mitgliedern war nach ca. 3 Jahren erreicht. Durch Karteileichen und mangels Clubbeitrag konnte dieser Mitgliederstand gehalten werden, bis Hartmut im letzten Jahr den FzdAM verschickte, der den "realen" Mitgliederstand von 51 Mitglieder offenbarte.

Das Clubinfo erblickte schon 1984 das Licht der Welt. Bis zur Nr. 6 übernahm Günther Wagner selbst die Redaktion, aber mit der wachsenden Anzahl von Artikeln, insbesondere von Hartmut Obermann, mußte ein hauptamtlicher Chefredakteur angeworben werden und fand sich mit Jens Neueder, der seit März 1985 (!), also seit 9 Jahren bzw. 38 Infos und dutzenden von Sonderinfos das Zusammenstellen des Infos, das Kopieren und das Versenden übernimmt!

An dieser Stelle sollten wir auch unsere inoffizielle Kassenwartin Jutta Obermann erwähnen, die schon so lange die Finanzen des Clubs in Ordnung hält, daß niemand mehr weiß, wann sie eigentlich damit angefangen hat.

Die Vorsitzenden des Clubs wechselten anfangs recht regelmäßig, in den letzten Jahren trat allerdings eine gewisse Beruhigung bei der Bildung des Vorstands ein. Den Anfang machte der Clubgründer Günther Wagner, der auf dem ersten Clubtreffen 1985 in Reinheim, bei dem 21 der 40 Mitglieder anwesend waren (!), für eine zweite Amtszeit bestätigt wurde. Beim Clubtreffen 1986 in Holzhausen übernahm Peter Stevens den Vorsitz, mußte aber mitten in seiner Amtszeit durch seinen Stellvertreter Hartmut Obermann abgelöst werden.

Hartmut wurde dann 1987 in Alsfeld von 16 Mitgliedern als 1. Vorsitzender eingesetzt. 1988 und 1989 trafen sich jeweils 21 Leute in Idstein, wobei die Youngsters Gerald Schröder und Alexander Schmid an die Spitze des Clubs gesetzt wurden. 1990 in Weilburg tauchte Fritz Chwolka auf, der sofort Mitglied, damit 18. Teilnehmer des Treffens und 1. Vorsitzender des Clubs wurde (es wird kolportiert, daß Fritz' Bereitschaft zur Übernahme des Vorsitzes durch Helmut Bernhardts Frage an Fritz "Darf ich Dir ein Bier ausgeben?" und der positiv ausfallenden Antwort erreicht wurde).

1991 fiel das Clubtreffen mangels Beteiligung aus, da aber Gerald Schröder schon seine Bahnkarte gekauft hatte und sich daher bitterlich bei Hartmut beschwerte, trafen sich diese beiden doch - und es tauchten noch 8 andere Mitglieder auf.

Im nächsten Jahr war der Tiefststand der Clubaktivitäten erreicht - statt Club- wurde nur ein Vorstandstreffen abgehalten. Fritz blieb mangels anderer Bewerber an der Spitze des Clubs.

1993 kamen in Burgau-Röfingen 9 Mitglieder zusammen und Hartmut übernahm wieder die Geschicke des Clubs. 1994 konnten wir wieder sensationelle 14 Teilnehmer plus 4 Gäste verzeichnen und Hartmut behielt seinen Posten.

Als Kuriosität am Rande sollten wir die Eintragung des Clubs ins Vereinsregister vermerken: Seit dem ersten Clubtreffen wurde diese Eintragung dauernd vorbereitet, durch Beschlüsse und mit ständig revidierten Satzungen untermauert, dann aber fallengelassen, nachdem feststand, daß zur Eintragung der gesamte Vorstand zu einem Ort und einer Zeit präsent sein müßte, was in endlicher Zeit nicht zu erwarten ist.

Außerdem wurde auf jedem Treffen die Öffnung des Clubs in Richtung anderer Rechner, insbesondere IBM/MS-DOS, diskutiert und verworfen, da dies unserer "Tradition" nicht entsprechen würde. Faktisch gehen zwar die meisten Mitglieder fremd, aber alle denken mehr oder weniger wehmütig an die guten alten Zeiten und fühlen sich dadurch zusammengehörig.

Gerald Schröder

__Redaktionsschluß__ für das nächste Clubinfo ist der

01. September 1994.

Hallo Club 80 Mitglieder

Ich war auf dem Cluptreffen in Erlabrück und war so davon begeistert das ich mich spontan im Clup anmeldete. Hartmut bat mich dann auch noch mich im Info etwas vorzustellen. Dieses werde ich sofort erledigen. Nun ich bin 31 Jahre jung, Verheiratet, 186 cm Groß, mein Markenzeichen ist ein Oberlippenbart und Dunkelbraune Haare. Mit einem Computer kam ich 1985 erstmals in kontackt und diesen habe ich auch heute noch. Es ist ein CPC 464 mit Disketten Laufwerk auf diesem spielt nun mein Großer Sohn Patrick (6) und auch der kleine Christjan (3) versucht hin und wieder mal sein Glück an Ihm. Mit CPM kam ich aber erst vor sechs jahren in behrürung als ich mir einen CPC 6128 zulegte. Etwas später hatte ich das große glück das Alexander Schmid nach Kronach/Gehülz zog (er Wohnt etwa 10 Autominuten von mir entfernt) und mich mit dem Z3Plus in behrührung brachte. Nach anfänlichen Schwierigkeiten die mir Alexander aber beseitigte läuft es nun ziehmlich ohne Schwierigkeiten. Es ist halt nicht einfach ohne Festplatten laufwerk es voll auszunutzen. Tja nun würde ich noch sagen das ich seit einem halben Jahr einen Joyce 9512 von Steimer Ulrich bekam auf dem ich nun herum hämmere. Ich bin natürlich immer an Soft. und Hardware für Joyce und CPC interesiert auch wenn jemand eine Bastelanleitung hat wie man ein Festplattenlaufwerk an den CPC oder Joyce anschliesen kann so würde mich dieses interesieren. Meine Configuration möchte ich doch nun auch noch nennen :

So da währe ein CPC 464 von 85 mit DD1 3'' und Robotron 5.25 + 6128 Rom umschaltbar! natürlich mit Grünmonitor.	Nun noch ein CPC 6128 mit Dobbertin 512k Ram erweiterung Star lc 24 10 Drucker Zwei 5.25 ein 3.5 Lautwerk Selbstbau Schnittstelle aus dem PC1 Modem Farbmonitor und Z3Plus System.	Tja jetzt wäre noch zu erwähnen der Joyce 9512 von Steimer Ulrich auf dem natürlich auch das Z3Plus läuft er hat auch noch zwei 5.25 Laufwerke dabei.

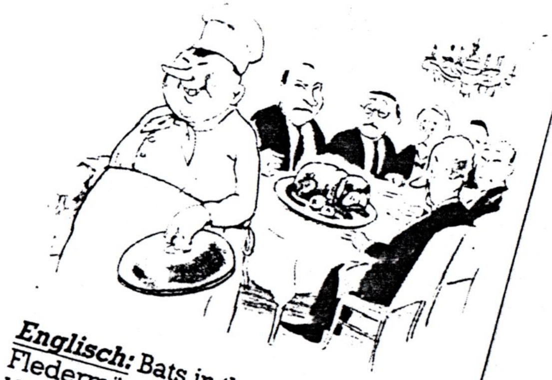
Wenn natürlich jemand von euch an Software für den CPC interesiert ist so kann er sich natürlich auch bei mir melden villeicht habe ich ja auch etwas interresantes für ihn.

Tja das solls fürs erstemal gewesen sein bis dann sagt euch

Kalb M.

Arabisch: Hammam bila mojje.
Eine Badeanstalt ohne Wasser.
Müßiges Geplapper.

Chinesisch: Kuang yen hsia tzu.
Ein Blinder mit strahlenden Augen.
Einer, der mehr scheint, als er ist.
Kou jou shang pu techeng hsi.
Du kannst kein Hundefleisch auf einem Bankett servieren.
Narren sind großen Unternehmungen nicht gewachsen.



Englisch: Bats in the belfry.
Fledermäuse im Glockenturm.
Verrückt.

Französisch: Se noyer dans un verre d'eau.
In einem Glas Wasser ertrinken.
Sich unnötige Schwierigkeiten machen.

Miriam Rech,
Glan-Münchweiler

Grafik des Genie IIIs unter Holte CP/M+

Volker Dose, Egbert Schröer, Juni-August 1994

1. Grundlagen

Der zweite Beitrag zur kleinen Artikelserie über die Ansteuerung der Grafik des Genie IIIs unter CP/M zeigte die Möglichkeiten mit Turbo Pascal. Hier sind nun Möglichkeiten zur Programmierung in ANSI-C.

Thomas Holte selbst hat viele Utilities in C programmiert, nutzte dazu aber den MIC Compiler von Rose. Dieser ist zwar für die unterschiedlichsten Rechner noch zu kaufen, allerdings läßt Herr Rose mit sich nicht über den Preis reden und so bleibt dieser C-Compiler mit ca. 400,- in der heutigen CP/M Welt unbeachtet; schade. Holte hat mit der Überlassung der Source Codes des Genie 3s für den Club 80 auch die C-Routinen zur Graphik-Ansteuerung zur Verfügung gestellt. Deshalb fällt die Entscheidung neue Wege mit dem jetzt PD-Compiler von Hitech zu gehen leicht, zumal dieser den ANSI Standard unterstützt.

Die erste Arbeit, die man nun erledigen muß, ist die Erstellung einer Library für den C-Compiler aus den Holte´schen MIC Routinen. Die eigentliche Graphikfunktionalität ist ja im BIOS implementiert. Dem C-Compiler wird halt nur in den Funktionen gesagt, welche BIOS-Funktion zu nutzen ist und wie. Dazu einige Beispiele:

```
/*****************************************************************************
* C L O S E P L * L I B P L O T 0 1 5 * T h o m a s   H o l t e * 8 6 0 4 0 1 *
*****************************************************************************
*
* Version 1.0 by Thomas Holte.
*
* Closes graphic screen.
*/

closepl ()
ä
  system  (25, 0);  /* disable output on graphic screen of phoenix Genie IIIs */
ü


/*****************************************************************************
*  P O I N T  *  L I B P L O T 0 3 5  * T h o m a s   H o l t e * 8 6 0 4 0 1 *
*****************************************************************************
*
* Version 1.0 by Thomas Holte.
*
* Plots point (x,y).
*/

point (x, y)
  int x, y;
ä
  system (27, 0x11, 0, y, x);    /* plots point on phoenix Genie IIIs */
ü


/*****************************************************************************
```



```
* L I N E * L I B P L O T 0 2 5 * T h o m a s   H o l t e * 8 6 0 4 0 1 *
*****************************************************************************
*
* Version 1.0 by Thomas Holte.
*
* Draws a line from (x1,y1) to (x2,y2).
*/

line (x1, y1, x2, y2)
  int x1, y1, x2, y2;
ä
  /* common line drawing routine */
/*
  int dx, dy, px, py;
  int x = x1;
  int y = y1;
  int p = 0;

  if (x2 >= x1)
  ä
    px = x2 - x1 + 1;
    dx = 1;
  ü
  else
  ä
    px = x1 - x2 + 1;
    dx = -1;
  ü

  if (y2 >= y1)
  ä
    py = y2 - y1 + 1;
    dy = 1;
  ü
  else
  ä
    py = y1 - y2 + 1;
    dy = -1;
  ü

  point (x, y); /* wurde ja in POINT.C definiert ! */

  if (py <= px)
  ä
    while (x != x2)
    ä
      if ((p += py) >= px)
      ä
        p -= px;
        y += dy;
      ü

      point (x += dx, y);
    ü
  ü
  else
  ä
    while (y != y2)
    ä
```

```
        if ((p += px) >= py)
        ä
          p -= py;
          x += dx;
        ü

        point (x, y += dy);
      ü
    ü
*/

  /* line drawing routine on phoenix Genie IIIs */
  system (29, 0x11, 0, &x1);
ü
/*====================================================================*/
```

Diese Library mit Namen HOLTE.LIB enthält die volle Grafikfunktionalität des Genie 3s - selbstverständlich mit den Windows Optionen - und ist auf einer der V0xx Disk´s (für Z3PLUS User) oder direkt bei uns erhältlich.

```
/*************************************************************************/
/*  Funktionsprototypen der Grafiklibrary für Genie IIIs  mit  Holte    */
/*  CP/M +.                                       HOLTE.LIB-File        */
/*************************************************************************/
/*      Volker Dose, Egbert Schroeer                                    */
/*      function window not included                                    */

void erase (void);                  /* loescht HRG Seite 0             */
void openpl (void);                 /* loescht HRG Seite 0             */
                                    /* und schaltet an                 */
void closepl (void);                /* schaltet HRG Seite 0 aus        */
void point (int x, int y);          /* setzt einen Punkt (639,449)     */

/* zieht Linie (links,unten,rechts,oben)       */
void line (int x1, int y1, int x2, int y2);

/* Kreisbogen (mittel_x,y,start_x,y,end_x,y   */
void arc (int x, int y, int x0, int y0, int x1, int y1);

void circle (int x, int y, int r);  /* Kreis (mittel_x,y,radius)       */
void move (int x, int y);           /* Grafikcursor nach (x,y)         */
void cont (int x, int y);           /* draw line from (X,Y) to (x,y)   */
void unpoint (int x, int y);        /* loesche einen Punkt (639,449)   */

void unline (int x1, int y1, int x2, int y2);       /* loescht eine Line  */

/* loescht einen Kreis mittelpunkt_x,y radius     */
void uncircle (int x, int y, int r);

/* loescht einen Kreisbogen mittelpunkt_y,y radius    */
void unarc (int x, int y, int x0, int y0, int x1, int y1);

void uncont (int x, int y);         /* undraw line from (X,Y) to (x,y)  */
/*====================================================================*/
```

Natürlich wird auch - wie bei der Programmierung unter Turbo Pascal - eine Möglichkeit gebraucht, um über den BIOS-Aufruf über BDOS-

Funktion 50 zu realisieren. BIOS3.C befindet sich in der HOLTE.LIB, BIOS3.H wird als Header File in das entsprechende Programm eingebunden werden (s.a. Disk V057). Zusätzlich sind die ´system()´ Funktionen der original Holte Library eingebunden, um die Spezialfunktionen des BIOS nutzen zu können.

```
/* BIOS-Aufruf fuer CP/M Plus ueber die BDOS-Funktion 50 */

/* CP/M-80 BIOS entry points */
#define BOOT    0           /* perform cold start initialization     */
#define WBOOT   1           /* perform warm start initialization     */
#define CONST   2           /* check for console input char ready    */
#define CONIN   3           /* read  console character in            */
#define CONOUT  4           /* write console character out           */
#define LIST    5           /* write list    character out           */
#define AUXOUT  6           /* write auxiliary output character      */
#define AUXIN   7           /* read  auxiliary input  character      */
#define HOME    8           /* move to track 00 on selected disk     */
#define SELDSK  9           /* select disk drive                     */
#define SETTRK  10          /* set track  number                     */
#define SETSEC  11          /* set sector number                     */
#define SETDMA  12          /* set DMA address                       */
#define READ    13          /* read  specifidd sector                */
#define WRITE   14          /* write specified sector                */
#define LISTST  15          /* return list status                    */
#define SECTRN  16          /* translate logical to physical sector */
#define CONOST  17          /* return output status of console       */
#define AUXIST  18          /* return input  status of aux. port     */
#define AUXOST  19          /* return output status of aux. port     */
#define DEVTBL  20          /* return address of char. I/O table     */
#define DEVINI  21          /* initialize char. I/O devices          */
#define DRVTBL  22          /* return address of disk drive table    */
#define MULTIO  23          /* set number of logically consecutive   */
                            /* sectors to be read or written         */
#define FLUSH   24          /* force physical buffer flushing for    */
                            /* user-supported deblocking             */
#define MOVE    25          /* memory to memory move                 */
#define TIME    26          /* time set/get signal                   */
#define SELMEM  27          /* select bank of memory                 */
#define SETBNK  28          /* specify bank for DMA operation        */
#define XMOVE   29          /* set bank when a buffer is in a bank   */
                            /* other than 0 or 1                     */
#define USERF   30          /* reserved for system implementor       */


#ifndef uchar
#define uchar unsigned char
#endif

/* function prototypes */
unsigned bios(uchar nummer, ...);
unsigned bioshl(uchar nummer, ...);
/*==================================================================*/
```

Um kein Plagiat zu begehen: An BIOS3.* war Alexander Schmid nicht ganz unbeteiligt.

2. Programmierung der Grafik

Die Voraussetzungen sind nun implementiert. Es folgt ein kleines Beispielprogramm, das einige Grafikspielereien erzeugt.

Das Programm GRATEST zeigt dabei alle Möglichkeiten der Grafik. Die Punkte werden nur normiert gesetzt bzw. gelöscht, das schafft Kompatibilität zwischen Rechnern mit verschieden Zeichensätzen (z.B. FONT14 und FONT12) und Kreise werden immer normiert gezeichnet.

```
/* GRATEST.C Beispielprogramm mit den
   derzeitigen Grafikmöglichkeiten */

#include <stdio.h>
#include <time.h>
#include <grafik.h>


int main(void);
int rand();


int zaehler, radius, x0, y0, x1, y1;
int max_x,max_y,max_int;

#define sub 0x1a


/* draws a box (lower_left to upper_right)   */
void box(int xb0, int yb0, int xb1, int yb1)
ä
        line(xb0,yb0,xb1,yb0);          /* links unten nach rechts unten */
        line(xb1,yb0,xb1,yb1);          /* rechts unten nach rechts oben */
        line(xb1,yb1,xb0,yb1);          /* rechts oben nach links oben   */
        line(xb0,yb1,xb0,yb0);          /* links oben nach links unten   */
ü


/* undraws a box ( lower_left to upper_right */
void unbox(int xb0, int yb0,int xb1, int yb1)
ä
        unline(xb0,yb0,xb1,yb0);        /* links unten nach rechts unten */
        unline(xb1,yb0,xb1,yb1);        /* rechts unten nach rechts oben */
        unline(xb1,yb1,xb0,yb1);        /* rechts oben nach links oben   */
        unline(xb0,yb1,xb0,yb0);        /* links oben nach links unten   */
ü


int main(void)
ä

        x0 = 0; y0 = 0; x1 = 100; y1 = 100;
        putch(sub);
        puts("Hallo Welt, wie geht's denn so ? ");


        openpl();
        line(x0,y0,x1,y1);
```



```
        line(100,100,620,400);
        circle(300,140,30);
        box(30,30,240,100);
        box(240,100,639,300);
        box(250,110,629,290);
        box(260,120,619,280);
        box(270,130,609,270);

        erase();
        for (radius =5; radius <200; radius +=2)
        circle(320,220,radius);

        for (radius = 5; radius < 200; radius +=2)
        uncircle(320,220,radius);


        move (330,280);
        srand (2204);
        for (zaehler = 0; zaehler <120; zaehler ++)
        ä
                x0 = rand() / 52;
                y0 = rand() / 80;
                        cont(x0,y0);
        ü

        move(330,280);
        srand(2204);
        for (zaehler = 0; zaehler <121; zaehler ++)
        ä
                x0 = rand() / 52;
                y0 = rand() / 80;
                        uncont(x0,y0);
        ü

        srand(2204);
        for (zaehler =0; zaehler < 52; zaehler ++)
        ä
                x0 = rand() / 52; x1 = rand() / 52;
                y0 = rand() / 80; y1 = rand() / 80;
                box(x0,y0,x1,y1);
        ü

        srand(2204);
        for (zaehler =0; zaehler < 52; zaehler ++)
        ä
                x0 = rand() / 52; x1 = rand() / 52;
                y0 = rand() / 80; y1 = rand() / 80;
                unbox(x0,y0,x1,y1);
        ü

        srand(2204);
        for (zaehler = 0; zaehler < 80; zaehler ++)
        ä
                x0 = rand() / 52; y0 = rand() / 80;
                radius = ( rand() / 1000 ) + 5;
                circle(x0, y0, radius);
        ü

        srand(2204);
```

```
        for (zaehler = 0; zaehler < 80; zaehler ++)
        ä
                x0 = rand() / 52; y0 = rand() / 80;
                radius = ( rand() / 1000 ) + 5;
                uncircle(x0, y0, radius);
        ü

        srand(2204);
        for (zaehler = 0; zaehler < 1500; zaehler ++)
        ä
                x0 = rand() / 52; y0 = rand() / 80;
                point(x0,y0);
        ü

        srand(2204);
        for (zaehler = 0; zaehler < 1500; zaehler ++)
        ä
                x0 = rand() / 52; y0 = rand() / 80;
                unpoint(x0,y0);
        ü

        puts("Fertig, Taste um zurueck ins Betriebssystem !!");
        getch();

        closepl();

ü
/*=================================================================*/
```

Jetzt sollten die Grundlagen klar sein, um ein komplexeres C-Programm vorzustellen, den PCX - Reader.

```
/* ---------------------------------------------------------------- */
/* dispcx.c   -   Read & Display a *.PCX graphics file in Microsoft's */
/* Quick C in EGA and in Borland's Turbo C 2.0 and equivilents with  */
/* all supported BGI graphics modes (VGA/EGA/CGA/Herc).              */
/* ---------------------------------------------------------------- */
/* Original by                          :  M. CLYNES   07-24-89     */
/* Borland Compiler Enhancements by     :  J. Braatz   02-16-92     */
/* Adapted to Club-80 Terminal by       :  A. Schmid   06-07-93     */
/* adapted to Genie IIIs by             :  E. Schroeer 18-11-93     */
/* ---------------------------------------------------------------- */
/* WARNING Only with LIBZ and HOLTE !!!                              */

#include <stdio.h>
#include <conio.h>
#include <grafik.h>

#define YMAX     449     /* Genie IIIs xmax=639,ymax=449 */
#define XMAX     639

char inversfl='N';
int  xoff=0,yoff=0;
int  eof_flag=0;

struct pcxheaderä
        char manufact;      /*  0    Hersteller-ID          */
        char version;       /*  1    Versionsnummer         */
        char encode;        /*  2                           */
```



```
        char bpp;              /*  3                            */
        int xmin;              /*  4, 5 Bildgroesse             */
        int ymin;              /*  6, 7      "                  */
        int xmax;              /*  8, 9      "                  */
        int ymax;              /* 10,11      "                  */
        int hres;              /* 12,13 Aufloesung der Hardware */
        int vres;              /* 14,15    "       "      "     */
        char colormapÄ48Ü;     /* 16-63                         */
        char reserved;         /* 64    momentan reserviert     */
        char ebene;            /* 65    Zahl der Farbebenen     */
        int bpline;            /* 66,67 Byte pro Zeile je Ebene */
        int paletinfo;         /* 68,69 Art des Bildes          */
        char blankÄ58Ü;        /* derzeit keine Informationen   */
        ü header;

/*=====================================================================*/
/*   gobal decoded data    char decodebuffÄ2000Ü;                      */
/*=====================================================================*/
/* Decode a line of .PCX                                               */
/* This procedure reads one encoded line from the image file          */
/* 0 = valid data stored                                               */
/* EOF = out of data in file                                           */

decodepcx(decodebuff, bytesline, fp)
char *decodebuff;                                 /* where to place data */
int bytesline;                                    /* # of bytes per line */
FILE *fp;                                         /*  image file handle  */
ä
  int data;
  int cnt;

  while(bytesline > 0 )ä
    cnt = 1;
    if(EOF == (data = getc(fp))) return(EOF);   /* retrive a data byte */
    if(0xc0 == (0xc0 & data))ä
      cnt = 0x3f & data;                         /* get repeat count */
      if(EOF == (data = getc(fp))) return(EOF);   /* get real data */
      ü

    while(cnt--)ä                          /* expand data into buffer */
      if(data==0x1A) eof_flag=1; else eof_flag=0;
      if(eof_flag && data==0x1A) data=0;
      *decodebuff = data;
      ++decodebuff;
      --bytesline;
      ü
    ü
  return(0);                                     /* file read status */
  ü
/* decodepcx()  end */

/*=====================================================================*/
/*                     read .PCX header into structure                */
/*=====================================================================*/
readheader(fp)
FILE *fp;
ä
  int count;
```

```
  char *loader;

  loader = (char *) &header;

  for(count = 0; count < 128; ++count)ä
     *loader = (char)fgetc(fp);
     ++loader;
     ü

/* Note: For Borland Compiler users, the following will need to be     */
/* changed to outtext() if you want to see the header information on   */
/* Hercules monitors (with the old herc.bgi driver)                     */

  printf("Encoding type %dÖn",header.encode);
  printf("Window X-min = %4d,  Y-min = %4dÖn",header.xmin, header.ymin);
  printf("Window X-max = %4d,  Y-max = %4dÖn",header.xmax, header.ymax);
  printf("Hres = %4d,  Vres = %4dÖn",header.hres, header.vres);
  printf("# bytes per/scan line %4d   ÖnÖn",header.bpline);
  printf("Bild (i)nvers oder (n)ormal einlesen ? ");
  inversfl=getch() & 0x5f;

  return(header.bpline);
ü

/*======================================================================*/
/*                        pixel line dump                               */
/*======================================================================*/
/* DUMP A RASTER LINE TO DISPLAY                                        */
/* pixeldump((pointer buffer), (# of BYTES), (Y line on screen to dump)) */
/*======================================================================*/

pixeldump(rdbuff, bytes, line)
char *rdbuff;
int line, bytes;

ä
  unsigned x;
  unsigned char byte, loop;

  bytes = (bytes * 8);

/* Sehr grosses Bild in X-Richtung komprimieren */

  if (header.xmax-header.xmin+xoff>XMAX*2)ä
    for(x = xoff; x < (bytes/2)+xoff; ++rdbuff)ä
      if (inversfl=='I')
        byte = ß(*rdbuff);
      else
        byte = *rdbuff;
      for (loop = 0x80; loop > 0; loop >>= 1)ä
        if (loop & byte) point(x, YMAX-line);
        loop >>= 1;
        if (loop & byte) point(x, YMAX-line);
        ++x;
        ü
      ü
    ü
  elseä
    for(x = xoff; x < bytes+xoff; ++rdbuff)ä
```



```
      if (inversfl=='I')
        byte = ß(*rdbuff);
      else
        byte = *rdbuff;
      for (loop = 0x80; loop > 0; loop >>= 1)ä
        if (loop & byte) point(x, YMAX-line);
        ++x;
        ü
      ü
    ü
  ü
/* pixeldump() End */


/*======================== MAIN() ==================================*/

int main( argc, argv )
int argc;
char *argvÄ20;
ä
  int line, status, bytesline;
  char decodebuffÄ2000,ch;
  FILE *fp;

  z3vinit();

  argc = argc;                          /* Nuke's Compiler warning   */

  cls();

  if((fp = fopen(argvÄ10,"rb")) == NULL)ä
    printf("ERROR: FILE '%s' CANNOT BE OPENED.Ön",argvÄ20);
    exit(0);
    ü
  bytesline = readheader(fp);
  printf("ÖnÖnOffset aus dem Header nehmen ?");
  ch=getch() & 0xDF;
  if((ch=='J') öö (ch=='Y') öö (ch==0x0D))ä
    xoff=header.xmin;
    yoff=header.ymin;
    ü
  elseä
    printf("Ön  X-Offset: "); scanf("%d",&xoff);
    printf("Ön  Y-Offset: "); scanf("%d",&yoff);
    ü
  line = yoff;
  status = 0;

  printf("ÖnÖnBildschirm vor dem Laden loeschen ? ");
  if((ch=getch() & 0xDF)=='J') erase();
  openpl();
  cls();

  while(status != EOF)ä     /* do EXTRA READS to srink vertical */
    if ((header.ymax-header.ymin+yoff)>YMAX*2)ä     /* sehr grosses Bild */
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
```

```
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);

      pixeldump(decodebuff, bytesline, line);
      ++line;
      ü
    else if (header.ymax-header.ymin+yoff>YMAX)ä     /* grosses Bild */
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      status = decodepcx(decodebuff, bytesline, fp);
      pixeldump(decodebuff, bytesline, line);
      ++line;
      ü
    else          /* Bild laesst sich mit Aufloesung darstellen */
      status = decodepcx(decodebuff, bytesline, fp);
      pixeldump(decodebuff, bytesline, line);
      ++line;
      pixeldump(decodebuff, bytesline, line);
      ++line;
    ü
    gotoxy(24,25);
    fprintf(stderr,"PRESS 'ENTER' to continue");
    getch();
    closepl();
  ü
/* End of main() */
```

Diese Beispiele sollten eigentlich ausreichend sein, um eigene Experimente zu starten. Gerüchten zufolge soll es ja noch einige andere Genie IIIs Besitzer im Club geben. Rafft euch mal auf und nehmt Kontakt auf.

Das Holte CP/M plus für den Genie IIIs Computer ist leider nie mit seinen hervorragenden Zusatzfunktionen programmtechnisch ausgenutzt worden. Der Rechner ist wohl zu einer Unzeit auf dem Markt erschienen, als schon die MSDOSe Welle über alles hereinbrach. Aus heutiger Sicht bedauerlich, aber so bleibt uns ein weites Betätigungsfeld - solange die Rechner lauffähig sind.

Juni-August '94

Egbert Schröer, Volker Dose

Club 80 Börse

Ich suche für den Amateurfunk ein Paket Radioprogamm für einen Amstrad CPC 464/6128 wenn möglich für CP/M 3.
Angebot an Michael Vogl Postfach 100610 41490 Grevenbroich
Tel./Datex-J +49218145112-1 Fido 2:2433/27.2
Z-Netz MICHAEL_VOGL@R-M-B-1.gun.de

Club 80 Börse ---

Datum : 11-07-94 22:32 (PRIVAT) (EMPFANGEN) (GESENDET)
Von : Alexander Schmid bei [77:7700/740]
An : Jens Neueder bei [77:7700/740]

Assembler-Tip

Wenn innerhalb von Programmen Pufferbereiche vorkommen, mache ich es während der Entwicklung eigentlich immer so, daß ich im COM-File einen Text zur Identifizierung einsetze, was dann z.B. so aussieht:

```
0100  C3 45 02 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0110  42 75 66 66 65 72 20 31  11 11 11 11 11 11 11 11  Buffer 1........
0120  11 11 11 11 11 11 11 11  11 11 11 11 11 11 11 11  ................
0130  11 11 11 11 11 11 11 11  11 11 11 11 11 11 11 11  ................
0140  11 11 11 11 11 11 11 11  11 11 11 11 11 11 11 11  ................
0150  42 75 66 66 65 72 20 32  22 22 22 22 22 22 22 22  Buffer 2""""""""
0160  22 22 22 22 22 22 22 22  22 22 22 22 22 22 22 22  """"""""""""""""
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

So sieht man nachher mit einem Debugger o.ä., ob man z.B. einen Stackbereich viel zu groß gemacht hat, oder ob man vielleicht einen anderen Puffer überschreibt. Im Quelltext steht dazu dann irgendwo:

```
buf1:  defb    'Buffer 1'      ; Markierungstext
       defs    56,11h          ; Rest des Puffers fuellen
buf2:  defb    'Buffer 2'
       defs    24,22h
```

Um aber nicht immer lange die Zeichen für die restlichen Füllbytes zählen zu müssen, habe ich mir folgendes kleines Macro gebastelt:

```
resbuf  macro   length,text,fill
        local   x1,x2,x3
x1:     defb    text            ; Text einsetzen
x2      equ     $-x1            ; Textlaenge
x3      equ     length-x2       ; verbleibende Bytes
        defs    x3,fill         ; Fuellbytes einsetzen
        endm
```

Man gibt nur die Gesamtgröße des Puffers, den Text und das Füllbyte an, der Rest geht automatisch. In einem Programm sieht das dann so aus:

```
        org     100h
        jp      start
        ds      13
buf1:   resbuf  40h,'Buffer 1',11h
buf2:   resbuf  20h,'Buffer 2',22h
         .       .
         .       .
         .       .
```

Viele ältere Assembler erlauben es nicht, das Füllbyte bei DEFS frei zu wählen, dann muß man das Macro und den Aufruf entsprechend ändern.

Alexander Schmid

Datum : 11-07-94 22:33 (PRIVAT) (EMPFANGEN) (GESENDET)
Von : Alexander Schmid bei [77:7700/740]
An : Jens Neueder bei [77:7700/740]

Textumsetzung die N + erste

Es gibt ja schon diverse Programme, um die Umlaute zwischen CP/M und DOS umzusetzen und ich hatte das auch schon mal in einem Programm zussammengefaÔt, aber das Programm war bis jetzt auf MSDOS-Umlaute beschrÕnkt. Als ich mal wieder bei Hartmut war, hatte er die Idee, das Programm auch f▪r Windows anzupassen und so kann man jetzt folgende Umwandlungen machen:

```
CP/M ASCII      --> MSDOS Text
CP/M ASCII      --> MSDOS Wordstar
MSDOS Text      --> CP/M ASCII
MSDOS Wordstar  --> CP/M ASCII
MSDOS Text      --> CP/M ae, ue, oe
MSDOS Wordstar  --> CP/M ae, ue, oe
MSDOS Text      --> Windows
Windows         --> MSDOS Text
CP/M ASCII      --> Windows
Windows         --> CP/M ASCII
```

Bei den CP/M-Texten wird zusÕtzlich das 7. Bit gel-scht, sodaÔ auch Wordstar-Texte vom CP/M ohne Probleme umgesetzt werden k-nnen. Wer das Listing nicht abtippen will, kann es in der Discovery per DFÉ holen, oder von mir auf Floppy bekommen.

```
(* Textumsetzungen zwischen CP/M und IBM           *)
(* A. Schmid, 5/93-6/94                            *)
(* compilierbar mit TP 3.0 - 6.0 unter CP/M und DOS *)

TYPE str15=STRING[15];

VAR infile,outfile,temp,mode:str15;
    b,c:CHAR;
    t,z:TEXT;

PROCEDURE show_mode;
BEGIN;
  WriteLn('Modi sind: 0 - CP/M ASCII    -> IBM Text');
  WriteLn('           1 - CP/M ASCII    -> IBM WS');
  WriteLn('           2 - IBM Text      -> CP/M ASCII');
  WriteLn('           3 - IBM WS        -> CP/M ASCII');
  WriteLn('           4 - IBM Text      -> CP/M ae,oe,ue');
  WriteLn('           5 - IBM WS        -> CP/M ae,ue,oe');
  WriteLn('           6 - IBM Text      -> Windows Text');
  WriteLn('           7 - Windows Text  -> IBM Text');
  WriteLn('           8 - CP/M ASCII    -> Windows Text');
  WriteLn('           9 - Windows Text  -> CP/M ASCII');
  WriteLn;
END;
```

```
PROCEDURE con0; (* CP/M ASCII file -> IBM text file *)
BEGIN
  WHILE NOT EOF(t) DO
    BEGIN
      Read(t,c);
        c:=Chr(Ord(c) AND $7F);
        IF c IN [#31,'@','[','\',']','{','|','}','~'] THEN
        BEGIN
          CASE c OF
            #31 :b:=#45;
            #64 :b:=#21;
            #91 :b:=#142;
            #92 :b:=#153;
            #93 :b:=#154;
            #123:b:=#132;
            #124:b:=#148;
            #125:b:=#129;
            #126:b:=#225;
          END;
          Write(z,b); Write('.');
        END
        ELSE
          Write(z,c);
    END;
END;
```

```
PROCEDURE con1; (* CP/M ASCII file -> IBM WS file *)
BEGIN
  WHILE NOT EOF(t) DO
    BEGIN
      Read(t,c);
        c:=Chr(Ord(c) AND $7F);
        IF c IN ['@','[','\',']','{','|','}','~'] THEN
        BEGIN
          Write(z,#27);
          CASE c OF
            #64 :b:=#21;
            #91 :b:=#142;
            #92 :b:=#153;
            #93 :b:=#154;
            #123:b:=#132;
            #124:b:=#148;
            #125:b:=#129;
            #126:b:=#225;
          END;
          Write('.');
          Write(z,b);
          Write(z,#28);
        END
        ELSE
          Write(z,c);
    END;
  END;
```

```
PROCEDURE con2; (* IBM text file -> CP/M ASCII file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
      CASE c OF
        #21 : BEGIN c:=#64; Write ('.'); END;
        #142: BEGIN c:='['; Write ('.'); END;
        #153: BEGIN c:='\'; Write ('.'); END;
        #154: BEGIN c:=']'; Write ('.'); END;
        #132: BEGIN c:='{'; Write ('.'); END;
        #148: BEGIN c:='|'; Write ('.'); END;
        #129: BEGIN c:='}'; Write ('.'); END;
        #225: BEGIN c:='~'; Write ('.'); END;
      END;
      c:=Chr(Ord(c) AND $7F);
      Write(z,c)
    END;
END;
```

```
PROCEDURE con3; (* IBM WS file -> CP/M ASCII file *)
BEGIN
  WHILE NOT EOF(t) DO
    BEGIN
      Read(t,c);
       IF (c=#27) OR (c=#155) THEN
          BEGIN
             Read(t,c);
               CASE c OF
                   #21 : b:=#64;
                   #142: b:='[';
                   #153: b:='\';
                   #154: b:=']';
                   #132: b:='{';
                   #148: b:='|';
                   #129: b:='}';
                   #225: b:='~'
               END;

             Write('.');
             Write(z,b);
             Read(t,c)
         END
        ELSE
          BEGIN
            c:=Chr(Ord(c) AND $7F);
            Write(z,c);
          END;
    END;
 END;
```

```
PROCEDURE con4; (* IBM text -> CP/M ae,oe,ue *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    CASE c OF
      #21 : BEGIN Write(z,#64);                   Write ('.'); END;
      #142: BEGIN Write(z,'A'); Write(z,'e'); Write ('.'); END;
      #153: BEGIN Write(z,'O'); Write(z,'e'); Write ('.'); END;
      #154: BEGIN Write(z,'U'); Write(z,'e'); Write ('.'); END;
      #132: BEGIN Write(z,'a'); Write(z,'e'); Write ('.'); END;
      #148: BEGIN Write(z,'o'); Write(z,'e'); Write ('.'); END;
      #129: BEGIN Write(z,'u'); Write(z,'e'); Write ('.'); END;
      #225: BEGIN Write(z,'s'); Write(z,'s'); Write ('.'); END;
    ELSE
      c:=Chr(Ord(c) AND $7F);
      Write(z,c);
    END;
  END;
END:
```

```
PROCEDURE con5; (* IBM WS -> CP/M ae,oe,ue *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
    IF (c=#27) OR (c=#155) THEN
    BEGIN
      Read(t,c);
      CASE c OF
        #21 : BEGIN Write(z,#64);                  END;
        #142: BEGIN Write(z,'A'); Write(z,'e'); END;
        #153: BEGIN Write(z,'O'); Write(z,'e'); END;
        #154: BEGIN Write(z,'U'); Write(z,'e'); END;
        #132: BEGIN Write(z,'a'); Write(z,'e'); END;
        #148: BEGIN Write(z,'o'); Write(z,'e'); END;
        #129: BEGIN Write(z,'u'); Write(z,'e'); END;
        #225: BEGIN Write(z,'s'); Write(z,'s'); END;
      END;
      Write('.');
      Read(t,c)
    END
    ELSE
    BEGIN
      c:=Chr(Ord(c) AND $7F);
      Write(z,c);
    END;
  END;
END;

PROCEDURE con6; (* IBM text file -> Windows text *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
      CASE c OF
        #21 : BEGIN c:=#$A7; Write ('.'); END;
        #142: BEGIN c:=#$C4; Write ('.'); END;
        #153: BEGIN c:=#$D6; Write ('.'); END;
        #154: BEGIN c:=#$DC; Write ('.'); END;
        #132: BEGIN c:=#$E4; Write ('.'); END;
        #148: BEGIN c:=#$F6; Write ('.'); END;
        #129: BEGIN c:=#$FC; Write ('.'); END;
        #225: BEGIN c:=#$DF; Write ('.'); END;
      END;
      Write(z,c)
    END;
END;

PROCEDURE con7; (* Windows text -> IBM text file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
      CASE c OF
        #$A7: BEGIN c:=#21 ; Write ('.'); END;
        #$C4: BEGIN c:=#142; Write ('.'); END;
        #$D6: BEGIN c:=#153; Write ('.'); END;
        #$DC: BEGIN c:=#154; Write ('.'); END;
        #$E4: BEGIN c:=#132; Write ('.'); END;
        #$F6: BEGIN c:=#148; Write ('.'); END;
        #$FC: BEGIN c:=#129; Write ('.'); END;
        #$DF: BEGIN c:=#225; Write ('.'); END;
      END;
      Write(z,c)
    END;
END;
```



```
PROCEDURE con8; (* CP/M ASCII file -> Windows text *)
BEGIN
  WHILE NOT EOF(t) DO
    BEGIN
      Read(t,c);
        c:=Chr(Ord(c) AND $7F);
        IF c IN [#31,'@','[','\',']','{','|','}','~'] THEN
        BEGIN
          CASE c OF
            (* #31 :b:=#45; *)
            #64 :b:=#$A7;
            #91 :b:=#$C4;
            #92 :b:=#$D6;
            #93 :b:=#$DC;
            #123:b:=#$E4;
            #124:b:=#$F6;
            #125:b:=#$FC;
            #126:b:=#$DF;
          END;
          Write(z,b); Write('.');
        END
        ELSE
          Write(z,c);
    END;
END;

PROCEDURE con9; (* Windows text -> CP/M ASCII file *)
BEGIN
  WHILE NOT EOF(t) DO
  BEGIN
    Read(t,c);
      CASE c OF
        #$A7: BEGIN c:=#64; Write ('.'); END;
        #$C4: BEGIN c:='['; Write ('.'); END;
        #$D6: BEGIN c:='\'; Write ('.'); END;
        #$DC: BEGIN c:=']'; Write ('.'); END;
        #$E4: BEGIN c:='{'; Write ('.'); END;
        #$F6: BEGIN c:='|'; Write ('.'); END;
        #$FC: BEGIN c:='}'; Write ('.'); END;
        #$DF: BEGIN c:='~'; Write ('.'); END;
      END;
      c:=Chr(Ord(c) AND $7F);
      Write(z,c)
    END;
END;

BEGIN
  IF ParamCount=0 THEN
  BEGIN
    WriteLn('IBM und CP/M Textumsetzer V1.1 A.Schmid 6/94');
    WriteLn;
    WriteLn('Dieses Programm setzt IBM Texte in CP/M Texte um und umgekehrt');
    WriteLn;
    WriteLn('Syntax: CONVERT infile outfile modus');
    WriteLn;
    show_mode;
    REPEAT
      Write('Name des Eingabefiles '); ReadLn(infile);
    UNTIL infile<>'';
    IF Pos('.',infile)=0 THEN
      temp:=infile+'.xxx'
     ELSE
      temp:=copy(infile,1,Pos('.',infile))+'xxx';
    Write('Name des Ausgabefiles (ENTER=',temp,') '); ReadLn(outfile);
    IF outfile='' then outfile:=temp;
```

```
    REPEAT
      Write('Modus '); ReadLn(mode);
    UNTIL (mode[1] in ['0'..'9']);
    WriteLn;
  END
  ELSE
  BEGIN
    infile:=ParamStr(1);
    outfile:='';
    mode:=ParamStr(2); mode:=mode[1];
    IF Not(mode[1] in ['0'..'9']) then
    BEGIN
      outfile:=ParamStr(2);
      mode:=ParamStr(3); mode:=mode[1];
    END;

    IF outfile='' THEN
    BEGIN
      mode:=ParamStr(2); mode:=mode[1];
      IF Pos('.',infile)=0 THEN
        outfile:=infile+'.xxx'
       ELSE
         outfile:=copy(infile,1,Pos('.',infile))+'xxx';
      WriteLn;
    END;

    IF (mode='') OR (mode>'9') THEN
    BEGIN
      show_mode;
      REPEAT
        Write('Modus : :',#8,#8);
        ReadLn(mode); mode:=mode[1];
      UNTIL mode[1] IN ['0'..'9'];
    END;
  END;

  writeln('infile=',infile,'  outfile=',outfile,'   Modus=',mode);writeln;
```



```
Assign(t,infile); (*$I-*) Reset(t); (*$I+*)
IF IOResult<>0 THEN
BEGIN
  WriteLn('Datei ',infile,' konnte nicht geoeffnet werden');
  HALT;
END;
temp:='$$$.$$$';
Assign(z,temp); Rewrite(z);

CASE mode[1] OF
  '0': con0;
  '1': con1;
  '2': con2;
  '3': con3;
  '4': con4;
  '5': con5;
  '6': con6;
  '7': con7;
  '8': con8;
  '9': con9;
END;

Close(t);
Close(z);

IF infile=outfile THEN
BEGIN
  Assign(t,infile); Erase(t);
END;

    Assign(t,outfile); (*$I-*) Reset(t); (*$I+*)
    WHILE IOResult=0 DO
    BEGIN
      WriteLn;
      Write('Ausgabedatei existiert schon, ueberschreiben (J/N) ? ');
      ReadLn(mode); mode:=UpCase(mode[1]);
      IF (mode<>'Y') AND (mode<>'J') THEN
      BEGIN
        Write('Neuer Name: ');
        ReadLn(outfile)
      END
      ELSE
        Erase(t);  (* altes outfile loeschen *)
      Assign(t,outfile); (*$I-*) Reset(t); (*$I+*)
    END;
    Assign(t,temp); Rename(t,outfile);
END.
```

Alexander Schmid

Datum : 11-07-94 22:34 (PRIVAT) (EMPFANGEN) (GESENDET)
Von : Alexander Schmid bei [77:7700/740]
An : Jens Neueder bei [77:7700/740]

Nachtrag und Erweiterung zum PCXPUT.C

Inzwischen habe ich es auch zu einer DOSe gebracht, wenn auch nur zu einem 386sx16, der normalerweise als reiner DFÉ-Knecht dient. So hatte ich aber die Gelegenheit, die mit PCXPUT erzeugten Bildchen auf der DOSe anzusehen und das Erstaunen war groÔ: geht nicht. Warum nicht ? Die DOSe wollte mehr Informationen, als PCXPUT in den Header geschrieben hat. Im einzelnen sind das die Hersteller-ID, die Versionsnummer und die Farbtabelle. PCXSHOW braucht die nicht und daher bin ich erst jetzt drauf gekommen. Manche Programme m-gen es auch nicht, wenn eine Zeile mitten im Byte aufh-rt und es ist sinnvoller, das Fenster in 8-Punkt-Schritten zu bewegen. Um nicht unn-tig Verwirrung zu stiften, kommt hier nochmal das ▪berarbeitete Listing

```
/**************************************************************/
/* PUTPCX.C                                                   */
/*                                                            */
/* Gets pixel data from the screen and writes a PCX format file */
/* Alexander Schmid 9/93 - 7/94                               */
/**************************************************************/

#include <stdio.h>
#include <conio.h>
#include <grafmod2.h>    /* computer specific graphic driver */

struct pcxheader{
       uchar manufact;     /*  0   Hersteller-ID              */
       uchar version;      /*  1   Versionsnummer             */
       uchar encode;       /*  2                              */
       uchar bpp;          /*  3                              */
       int xmin;           /*  4, 5 Bildgroesse               */
       int ymin;           /*  6, 7    "                      */
       int xmax;           /*  8, 9    "                      */
       int ymax;           /* 10,11    "                      */
       int hres;           /* 12,13 Aufloesung der Hardware   */
       int vres;           /* 14,15    "        "       "     */
```



```
       uchar colormap[48]; /* 16-63                          */
       uchar reserved;     /* 64   momentan reserviert       */
       uchar ebene;        /* 65   Zahl der Farbebenen       */
       int bpline;         /* 66,67 Byte pro Zeile je Ebene  */
       int paletinfo;      /* 68,69 Art des Bildes           */
       uchar blank[58];    /* derzeit keine Informationen    */
       } header;

       uchar colormap[48]={0,0,0,255,255,255,0,170,0,0,170,170,170,
                           0,0,170,0,170,170,170,0,170,170,170,85,85,
                           85,85,85,255,85,255,85,85,255,255,255,85,
                           85,255,85,255,255,255,85,255,255,255};

       int bytcnt=0;

/*==============================================================*/
/*                  write .PCX header into file                 */
/*==============================================================*/
writeheader(fp)
FILE *fp;
{
  int  count;
  char *loader;                  /* Zeiger auf ein character-array    */

  loader = (char*) &header;      /* LOADER zeigt auf den Header im Speicher */

  for(count = 0; count < 128; ++count){      /* 128 Bytes schreiben   */

    putc(*loader,fp);
    ++loader;                                /* Zeiger eins weiter    */
    }
}

/*==============================================================*/
/*                  read pixel from screen                      */
/*==============================================================*/
unsigned char readscreen(x,y)
int x,y;
{
  return point(x,YMAX-y);
}
```

```
/*========================================================================*/
/*                        mark area to write                               */
/*========================================================================*/
area(x1,y1,x2,y2)
int *x1,*y1,*x2,*y2;
{
  int xa1,ya1,xa2,ya2;
  char ch;

  hires();                                    /* Grafik einblenden    */
  color=2;                                    /* Punkt invertieren    */
  box(*x1,YMAX-*y1,*x2,YMAX-*y2);             /* Ausschnitt markieren */
  while((ch=getch())!=0x0D){                  /* auf ENTER warten     */
    xa1=*x1; ya1=*y1; xa2=*x2; ya2=*y2;       /* Werte merken         */
    switch(ch){
      case 'd': if(*x2<=XMAX-8) (*x2)+=8;  break;    /* rechter Rand -> */
      case 's': if(*x2>*x1+16) (*x2)-=8;   break;    /* rechter Rand <- */
      case 'x': if(*y2<=YMAX-8) (*y2)+=8;  break;    /* unterer Rand V  */
      case 'e': if(*y2>*y1+16) (*y2)-=8;   break;    /* unterer Rand ^  */
      case 'D': if(*x1<*x2-16) (*x1)+=8;   break;    /* linker Rand ->  */
      case 'S': if(*x1>=8) (*x1)-=8;       break;    /* linker Rand <-  */
      case 'X': if(*y1<*y2-16) (*y1)+=8;   break;    /* oberer Rand V   */
      case 'E': if(*y1>=8) (*y1)-=8;       break;    /* oberer Rand ^   */
      }
    box(xa1,YMAX-ya1,xa2,YMAX-ya2);  /* alten Rand loeschen         */
    box(*x1,YMAX-*y1,*x2,YMAX-*y2);  /* neuen Rand ziehen           */
    }
  box(*x1,YMAX-*y1,*x2,YMAX-*y2);    /* Markierung wieder loeschen */
  textmode();                        /* Grafik abschalten          */
}

/*========================================================================*/
/*                        do run length encoding                           */
/*========================================================================*/
compress(arr1,bytes,fp)
unsigned char arr1[];
int bytes;
FILE *fp;
{
  unsigned char arr2[150],ch;
  int repeat,ausgabe=0,eingabe=0;

  while(eingabe<bytes){                  /* Zahl der Eingangsbytes           */
    ch=arr1[eingabe];                    /* Byte aus dem Eingangsarray holen */
    ++eingabe;                           /* Eingabezeiger eins weiter        */
    repeat=1;                            /* Wiederholungszaehler             */
    while((ch==arr1[eingabe]) && (repeat<0x3F)){  /* Byte wiederholt sich */
      ++repeat; ++eingabe;                        /* Zaehler erhoehen      */
      }
    if(repeat>1){                          /* wenn es Wiederholungen gab */
      arr2[ausgabe]=(char)repeat + 0xC0;   /* Zaehlbyte zusammensetzen   */
      ++ausgabe;                           /* Ausgabezaehler erhoehen    */
      }
    else{
      if((ch & 0xC0) == 0xC0){       /* Datenbyte mit gesetztem Bit 6 und 7 */
        arr2[ausgabe]=0xC1;          /* Zaehlbyte = 1 schreiben             */
        ++ausgabe;                   /* Ausgabezaehler erhoehen             */
        }
      }
    arr2[ausgabe]=ch;                /* Byte in Ausgabearray schreiben      */
    ++ausgabe;                       /* Ausgabezaehler erhoehen             */
    }
  bytcnt += ausgabe;                 /* geschriebene Bytes zaehlen          */
  for(eingabe=0; eingabe<ausgabe; ++eingabe)  /* Ausgabearray auf Floppy    */
    fputc(arr2[eingabe],fp);                  /* schreiben                  */
}
```



```
/*========================================================================*/
/*                          write .PCX data                                */
/*========================================================================*/
writepcx(x1,y1,x2,y2,fp)
int x1,y1,x2,y2;
FILE *fp;
{
  int x,y,bits;
  unsigned char array[150],byte,count;

  hires();                                 /* Grafik einblenden               */
  for(y=y1; y<=y2; ++y){                   /* vom oberen zum unteren Rand     */
    count=0;                               /* Zaehler fuer den Zeilenpuffer   */
    for(x=x1; x<=x2; x+=8){                /* vom linken zum rechten Rand     */
      byte=0;
      for(bits=0; bits<8; ++bits){         /* Byte zusammenbasteln            */
        byte |= readscreen(x+bits,y)<<7-bits;
        }
      array[count]=byte;
      ++count;
      }
    compress(array,(x2-x1+1)/8,fp);        /* Zeile komprimieren und auf      */
    }                                      /* Floppy schreiben                */
  textmode();                              /* Grafik abschalten               */
  for(x=0; x<128-bytcnt%128; ++x)          /* Rest des letzten Sektors mit    */
    fputc(0x1a,fp);                        /* 1Ah fuellen                     */
}

/*========================== MAIN() ===================================*/

main( argc, argv )
int argc;
char *argv[];
{
  FILE *fp;
  int x1,y1,x2,y2,xoff=0,yoff=0;
  int a;
  char ch;

  argc = argc;

  if((fp = fopen(argv[1],"rb")) != NULL){
    printf("ERROR: FILE '%s' DOES ALREADY EXIST.\n",argv[1]);
    exit(0);
    }

  if((fp = fopen(argv[1],"wb")) == NULL){
    printf("ERROR: CAN NOT OPEN FILE '%s' \n",argv[1]);
    exit(0);
    }
  x1=0; y1=0; x2=199; y2=199;
  do{
    area(&x1,&y1,&x2,&y2);
    printf("Die Koordinaten sind %d/%d %d/%d\n",x1,y1,x2,y2);
    }while((ch=getch())==0x0D);     /* Korrektur mit RETURN moeglich */

  printf("\nX-/Y-Offsets so lassen ? ");
  if(((ch=getch() & 0xDF) == 'J') || (ch=='Y') || (ch==0x0D)){
    header.xmin=x1;
    header.ymin=y1;
    header.xmax=x2;
    header.ymax=y2;
    }
```

```
  else{
    printf("\n\n  X-Offset: "); scanf("%d",&xoff);
    printf("  Y-Offset: "); scanf("%d",&yoff);
    header.xmin=xoff;
    header.ymin=yoff;
    header.xmax=x2-x1+xoff;
    header.ymax=y2-y1+yoff;
    }
  header.manufact=10;
  header.version=5;
  header.encode=1;
  header.bpp=1;
  header.hres=XMAX;
  header.vres=YMAX;
  header.ebene=1;
  header.bpline=(x2-x1+1)/8;

  for(a=0; a<48; ++a)
    header.colormap[a]=colormap[a];

  writeheader(fp);
  writepcx(x1,y1,x2,y2,fp);
  fclose(fp);

  exit(0);
  }
/* End of main() */
```

Jetzt kann man seine Kreationen (hoffentlich) problemlos mit DOSen austauschen. Bei dem Programm, das ich auf der DOSe habe, w▪rde es auch ohne die Farbtabelle, also nur mit ID und Versionsnummer, gehen, aber die "fertigen" PCX-Bildchen haben auch immer einer Tabelle und sicher ist sicher. Wie die Werte in der Tabelle genau zustande gekommen sind, weiÔ ich nicht, aber alle meine schwarz/weiÔ Bilder haben dieselbe Tabelle, so falsch kann die also nicht sein. Die uchar in der Struktur bedeuten beim Hi-Tech-C unsigned char und ich habe die hier statt char eingesetzt, damit es mit dem Wert 255 keine Probleme gibt. Ihr k-nnt nat▪rlich auch char lassen und ausprobieren, ob es damit bei Euch geht. Um ganz korrekt zu sein, wird auch noch der Rest des letzten Sektors mit 1Ah aufgef▪llt.

Alexander Schmid

Datum : 11-07-94 22:36 (PRIVAT) (EMPFANGEN) (GESENDET)
Von : Alexander Schmid bei [77:7700/740]
An : Jens Neueder bei [77:7700/740]

Nachtrag und Erweiterung zum PCXSHOW.C

PCXSHOW war auch noch nicht ganz wasserdicht, es ist z.B. davon ausgegangen, daÔ der unbenutzte Rest des letzten Sektors mit 1A gef▪llt war und hat dabei nichtmal abgebrochen, sondern nur keinen Punkt gezeichnet. Abgebrochen wurde erst beim physikalischen Dateiende. PCXPUT hatte bis jetzt keine EOFs geschrieben und so war am unteren Rand der Bilder meistens ein Streifen mit M▪ll. Jetzt wird die Zeilenzahl, die ja im Header steht, ausgewertet und am echten Bildende abgebrochen. Auch wenn das PCX-File defekt ist, wird spõtestens am Dateiende abgebrochen.
Eine andere Erweiterung ist die Erkennung eines anderen Formats. Das DOS-Programm PV, mit dem man Bilder u.a. zwischen zahlreichen Formaten konvertieren kann, besteht darauf, PCX-Files mit 8 Bit pro Pixel zu schreiben, was dann z.B. so aussieht: FF 00 FF 01 FF 01. Das wõren hier 63 dunkle und danach 126 helle Punkte. Wahrscheinlich werden bei Farbbildern auch die anderen Bits benutzt, aber das alte PCXSHOW konnte damit nichts anfangen.

```
/* --------------------------------------------------------------- */
/* dispcx.c   -   Read & Display a *.PCX graphics file in Microsoft's */
/* Quick C in EGA and in Borland's Turbo C 2.0 and equivilents with   */
/* all supported BGI graphics modes (VGA/EGA/CGA/Herc).               */
/* --------------------------------------------------------------- */
/* Original by                        :  M. CLYNES  07-24-89           */
/* Borland Compiler Enhancements by   :  J. Braatz  02-16-92           */
/* Adapted to Club-80 Terminal by     :  A. Schmid  06-07-93 - 07-11-94 */
/* --------------------------------------------------------------- */

#include <stdio.h>
#include <conio.h>
#include <grafmod2.h>
char inversfl='N';
int  xoff=0,yoff=0;
int  linecnt;
```

```
struct pcxheader{
        char manufact;     /*  0    Hersteller-ID              */
        char version;      /*  1    Versionsnummer             */
        char encode;       /*  2                               */
        char bpp;          /*  3                               */
        int xmin;          /*  4, 5 Bildgroesse                */
        int ymin;          /*  6, 7      "                     */
        int xmax;          /*  8, 9      "                     */
        int ymax;          /* 10,11      "                     */
        int hres;          /* 12,13 Aufloesung der Hardware    */
        int vres;          /* 14,15     "         "     "      */
        char colormap[48]; /* 16-63                            */
        char reserved;     /* 64    momentan reserviert        */
        char ebene;        /* 65    Zahl der Farbebenen        */
        int bpline;        /* 66,67 Byte pro Zeile je Ebene    */
        int paletinfo;     /* 68,69 Art des Bildes             */
        char blank[58];    /* derzeit keine Informationen      */
        } header;

/*=======================================================================*/
/*   gobal decoded data   char decodebuff[200];                          */
/*=======================================================================*/
/* Decode a line of .PCX                                                 */
/* This procedure reads one encoded line from the image file            */
/* 0 = valid data stored                                                 */
/* EOF = out of data in file                                             */

decodepcx(decodebuff, bytesline, fp)
char *decodebuff;                                  /* where to place data */
int bytesline;                                     /* # of bytes per line */
FILE *fp;                                          /*  image file handle  */
{
  int data;
  int cnt;
  while(bytesline > 0 ){
    cnt = 1;
    if(EOF == (data = getc(fp))) return(EOF);   /* retrive a data byte */
    if(0xc0 == (0xc0 & data)){
      cnt = 0x3f & data;                         /* get repeat count */
      if(EOF == (data = getc(fp))) return(EOF);  /* get real data */
      }
```



```
    while(cnt--){                               /* expand data into buffer */
      *decodebuff = data;
      ++decodebuff;
      --bytesline;
      }
    }
  return(0);                                    /* file read status */
  }
/* decodepcx()  end */


/*======================================================================*/
/*                   read .PCX header into structure                    */
/*======================================================================*/
readheader(fp)
FILE *fp;
{
  int count;
  char *loader;

  loader = (char *) &header;

  for(count = 0; count < 128; ++count){
     *loader = (char)fgetc(fp);
     ++loader;
     }

/* Note: For Borland Compiler users, the following will need to be     */
/* changed to outtext() if you want to see the header information on   */
/* Hercules monitors (with the old herc.bgi driver)                     */

  printf("Encoding type %d\n",header.encode);
  printf("Window X-min = %4d,  Y-min = %4d\n",header.xmin, header.ymin);
  printf("Window X-max = %4d,  Y-max = %4d\n",header.xmax, header.ymax);
  printf("Hres = %4d,  Vres = %4d\n",header.hres, header.vres);
  printf("# bytes per/scan line %4d   \n\n",header.bpline/header.bpp);

  if(header.bpp!=1 && header.bpp!=8){
    printf("\n\nSorry ich kann nur Bilder mit 1 oder 8 Bit pro Pixel\n");
    printf("Dieses Bild hat %d Bit/Pixel\n",header.bpp);
    exit(0);
    }
```

```
  printf("Bild (i)nvers oder (n)ormal einlesen ? ");
  inversfl=getch() & 0x5f;

  return(header.bpline);
}

/*=======================================================================*/
/*                          pixel line dump                              */
/*=======================================================================*/
/* DUMP A RASTER LINE TO DISPLAY                                         */
/* pixeldump((pointer buffer), (# of BYTES), (Y line on screen to dump)) */
/*=======================================================================*/

pixeldump(rdbuff, bytes, line)
char *rdbuff;
int bytes, line;
{
  unsigned x;
  unsigned char byte, loop;
  int xxx;

  if(header.bpp==1) bytes = (bytes * 8);
  if (header.xmax-header.xmin+xoff>XMAX){    /* in X-Richtung komprimieren */
    for(x = xoff; x < (bytes/2)+xoff; ++rdbuff){
      if(header.bpp==1)
        byte = *rdbuff;
      else {
        byte=0;
        for (xxx=0; xxx<8; ++xxx) {
          byte |= *rdbuff << 7-xxx;
          ++rdbuff;
          }
        }
      if (inversfl=='I')
        byte = ~byte;

       for (loop = 0x80; loop > 0; loop >>= 1){
         if (loop & byte) plot(x, YMAX-line);
         loop >>= 1;
         if (loop & byte) plot(x, YMAX-line);
         ++x;
         }
       }
     }
   else{
     for(x = xoff; x < bytes+xoff; ++rdbuff){
       if(header.bpp==1)
         byte = *rdbuff;
       else {
         byte=0;
         for (xxx=0; xxx<8; ++xxx) {
           byte |= *rdbuff << 7-xxx;
           ++rdbuff;
           }
         }
       if (inversfl=='I')
         byte = ~byte;

       for (loop = 0x80; loop > 0; loop >>= 1){
         if (loop & byte) plot(x, YMAX-line);
         ++x;
         }
       }
     }
   }
 /* pixeldump() End */
```



```
/*======================== MAIN() ====================================*/

int main( argc, argv )
int argc;
char *argv[2];
{
  int line, status, bytesline;
  static char decodebuff[5000],ch;
  FILE *fp;

  argc = argc;                          /* Nuke's Compiler warning   */

  if((fp = fopen(argv[1],"rb")) == NULL){
    printf("ERROR: FILE '%s' CANNOT BE OPENED.\n",argv[2]);
    exit(0);
    }

  if (argv[2][0]=='1')
    color=1;              /* je nach Bild kann setzen oder */
  else
    color=2;              /* invertieren besser aussehen   */

  bytesline = readheader(fp);
  printf("\n\nOffset aus dem Header nehmen ?");
  ch=getch() & 0xDF;
  if((ch=='J') || (ch=='Y') || (ch==0x0D)){
    xoff=header.xmin;
    yoff=header.ymin;
    }
  else{
    printf("\n  X-Offset: "); scanf("%d",&xoff);
    printf("\n  Y-Offset: "); scanf("%d",&yoff);
    }
  line = yoff;
  status = 0;

  printf("\n\nBildschirm vor dem Laden loeschen ? ");
  if((ch=getch() & 0xDF)=='J') clsh();
  hires();
```

```
linecnt = header.ymax - header.ymin;
while(status != EOF && linecnt>0){     /* do EXTRA READS to srink vertical */
  if ((header.ymax-header.ymin+yoff)>YMAX*2){      /* sehr grosses Bild */
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    --linecnt;
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    --linecnt;
    }
  if (header.ymax-header.ymin+yoff>YMAX){          /* grosses Bild */
    status = decodepcx(decodebuff, bytesline, fp);
    pixeldump(decodebuff, bytesline, line);
    --linecnt;
    }
  status = decodepcx(decodebuff, bytesline, fp);
  pixeldump(decodebuff, bytesline, line);
  --linecnt;
  ++line;
  }
  color=2;
  grprint(1,1,"PRESS 'ENTER' to continue");
  getch();
  color=2;
  grprint(1,1,"PRESS 'ENTER' to continue");  /* erase text */
  textmode();
}
/* End of main() */
```

Ich hoffe, daß ich jetzt alle Schwachpunkte gefunden habe, aber soweit sieht's ganz gut aus.

Alexander Schmid

Datum : 24-08-94 22:36 (PRIVAT) (EMPFANGEN) (GESENDET)
Von : Alexander Schmid bei [77:7700/740]
An : Jens Neueder bei [77:7700/740]

Model-1 Emulator für DOSen

Wer denkt nicht hin und wieder wehmütig an die gute alte Zeit, als man stundenlang Hex-Dumps abgetippt hat und noch niemand an Windows und ähnliche Entgleisungen gedacht hat ? Aber jetzt ist Abhilfe in Sicht. In der Discovery liegt seit einiger Zeit ein Model-1 Emulator und was soll ich sagen, der funktioniert sogar. Das File heißt MODEL1_B.ZIP und enthält fast alles, was man braucht, um auf der DOSe einen richtigen Rechner zu emulieren. Aus Copyright-Gründen fehlen die ROMs und das DOS und das waren auch die größten Hürden, das Teil zum Laufen zu bringen. Ich hatte vor langer Zeit mal die original ROMs eines Model-1 ausgelesen, aber entweder ist da was schiefgegangen, oder ich habe die Reihenfolge vertauscht, jedenfalls lief er damit nicht. Dann habe ich die ROMs des Genie IIs genommen und damit funktionierts jetzt einwandfrei. In der Anleitung steht, daß die vier "Laufwerke" des Emulators, die als Files auf der Festplatte verwaltet werden, 80 Track SS SD sind. Das stimmt auch im Prinzip, aber die "Bootfloppy" muß unbedingt 40 Track SS SD haben, sonst steht man im Wald. Die PDRIVE-Parameter erlauben nur eine einzige Einstellung, mit der man auch die Originalfloppy formatieren muß, die dann als Muster dienen soll:

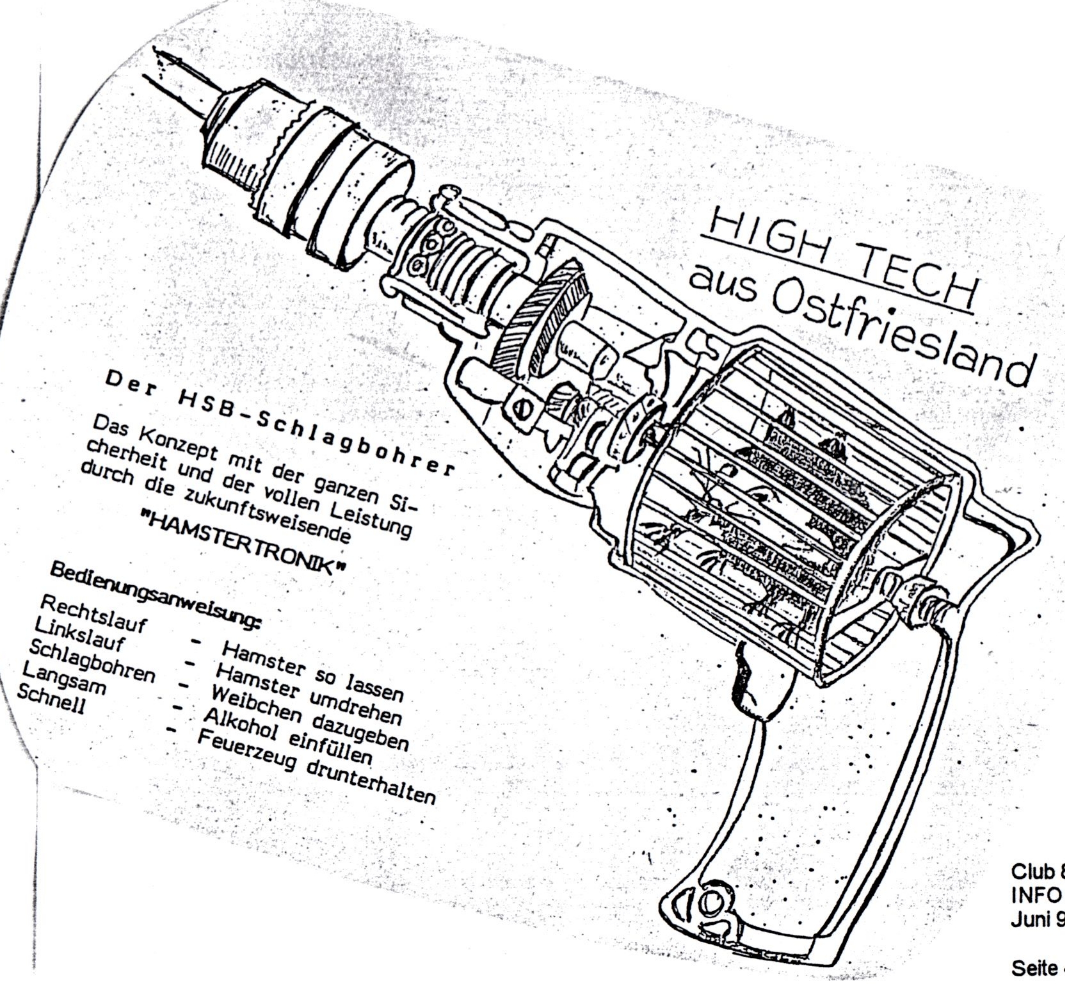
TI=AL TD=A SP=40 SEK=10 SWZ=0 EIB=2 SBIV=17 AEIV=2

Das sind die Namen vom GDOS, aber die Bedetung für's NewDOS dürfte klar sein. Die "Floppy-Files" sind eine exakte, sektorweise 1:1 Kopie, von der Bootspur bis zu HIT und GAT. Bei den anderen Laufwerken kann man problemlos SP=80 eintragen. Soweit ich es sehe, ist der Emulator 100% kompatibel, er stellt ja auch nur die "Hardware" zur Verfügung und die Software ist original.

Sogar der FORMAT-Befehl merkt nicht, daß er beschummelt wird und läßt schön brav das Sternchen rechts oben blinken und auch die Dreitasten-Befehle (123, DFG und JKL) funktionieren. Ärger machen nur Programme, die Illegals benutzen (z.B. Zeus), die laufen nicht und Verursachen einen Warmstart. Daneben bietet der Emulator einige Funktionen, die auf dem Model 1 so natürlich nicht möglich waren, z.B. auf Tastendruck einen Monitor, der u.a. am Bildschirmrand alle Register und einen Teil des Speichers anzeigt. Auf meinem 386sx16 ist die Arbeitsgeschwindigkeit nicht gerade berauschend, aber auf einem 486er erreicht man durchaus die Originalgeschwindigkeit und etwas mehr. Ein Problem stellt die Übertragung von Programmen dar, es gibt leider kein Tool, mit dem man einzelne Files direkt von der DOSe in den Emulator übertragen kann. Der Emulator unterstützt zwar die serielle Schnittstelle, aber wer hatte die schon für das Model 1 ? Hartmut Obermann und ich hatten angefangen, in Anlehnung an die Anleitung einen Weg über die parallele Schnittstelle auszutüfteln, aber das ist noch nicht lauffähig. Ich habe die Bootfloppy komplett über zwei Ecken mit Hilfe eines CP/M-Programms gelesen und als File zur DOSe überspielt, aber das dürfte für die meisten nicht in Frage kommen. Da muß also noch was getan werden. Mein Genie hat zwar eine serielle Schnittstelle, aber ich habe für NewDos leider
kein funktionsfähiges Terminalprogramm gefunden. Beim Kermit funktioniert zwar der Terminalmodus, aber der Filetransfer geht nicht. Hat vielleicht noch jemand ein Terminalprogramm für NewDos das XMODEM kann ? Wenn Bedarf besteht, kann ich mit meiner Methode aber ohne Probleme Floppies mit den wichtigsten Utilities und Programmen zusammenstellen. Am besten schickt Ihr mir dann eine NewDos-Floppy mit den Programmen und ich bastel Euch daraus Files, die der Emulator schluckt. Innerhalb des Emulators können die Files dann beliebig umkopiert werden und mit FORMAT kann man sich soviele Leerfloppies basteln, wie auf die Festplatte passen. Wäre nett, wenn Ihr die Floppies mit den o.g. Parametern (mit SP=80) formatieren würdet und auch ein paar 3 1/2" Floppies für die DOSen-Files dazu tut.

Wenn man sich für 25 Dollar registrieren läßt, bekommt man eine neuere Version des Emulators, die mehr Funktionen hat und schneller ist und wenn sich genug Interessenten finden, könnten wir auch versuchen, einen Clubrabatt rauszuhandeln. Wenn einer mit der Installation gar nicht zurechtkommt, kann er sich ruhig melden. Die ROMs werden natürlich nur gegen eine amtlich beglaubigte Kopie der original Kaufrechnung rausgegeben ;-)))

Alexander Schmid



Impressum

1. Vorsitzender:	**Hartmut Obermann** Wilhelm-Baur-Straße 8	Tel.: 07 21/ 85 40 68 BTX: 07 21/ FAX: 07 21/ 85 40 68
76135	Karlsruhe	
2. Vorsitzender:	**Gerald Schröder** Am Schützenplatz 14	Tel.: 0 41 05/ 26 02
21218	Seevetal	
Hardwarekoordinator:	**Andreas Magnus** Bismarckstraße 29	Tel.: 02 09/ 87 02 30
45879	Gelsenkirchen	
NewDOS-Diskothekar:	**Oliver Volz** Am Ochsenwald 37A	Tel.: 07 11/ 74 40 51
70565	Stuttgart (Rohrerhöhe)	
CP/M-Diskothekar:	**Fritz Chwolka** Saarstraße 34	Tel.: 0 24 64/ 89 20
52457	Aldenhoven	
C-128-Diskothekar:	**Günther W. Braun** Postfach 80 02 26	
81602	München	
Clubbücherei:	**Kurt Müller** Sophie-Scholl-Ring 3b	Tel.: 0 41 52/ 7 06 43
21502	Geesthacht	
Redaktion:	**Jens Neueder** Gschlachtenbretzingen Rudolf-Then-Straße 32	Tel.: 07 91/ 4 28 77 BTX: 07 91/ 44 47 22 FAX: 0 79 71/ 2 50 55
74544	Michelbach / Bilz	
Bankverbindung:	**Club 80** Postgiro Sonderkonto CLUB 80 Obermann H., 8870 Günzburg	**Postgiroamt Frankfurt** **BLZ: 500 100 60** **Kto.Nr.: 496 071 - 605**
Autoren:	Die Redaktion bedankt sich bei den im **Inhaltsverzeichnis** genannten Autoren für die Mitarbeit an der Club-INFO. Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.	

Schluß

Hallo Club 80'er,

nachdem ich meinen Urlaub hinter mir habe, ist wieder einmal INFO-Time. Die letzte Hauptversammlung und die Softwareartikel brachten einiges an Infomaterial, so daß wir diesmal doppelt so dick sind wie beim letzten INFO. Ich hoffe, Ihr habt damit mal wieder genügend zum Lesen. Ich fand zum Beispiel die Club-Historie recht spannend. Was in zehn Jahren so alles passiert! Prima wäre auch, wenn in der Discovery etwas mehr Aktivität herrschen würde (nicht nur zu gunsten meines Arbeitgebers, Ihr unterstützt indirekt auch mich!). Die Infoartikel könnt Ihr, wie Ihr sicher alle wißt, auch dort bunkern. Ich hole sie mir spätestens vor dem jeweiligen aktuellen INFO dort ab.

Nichts desto trotz möchte ich, wie immer an dieser Stelle, wieder die Bitte um Futter für das INFO loswerden. Nach der Sommerpause habt Ihr doch sicher schon den einen oder anderen Artikel für mich bereit?

Als Beilage im INFO findet Ihr diesmal einen Fragebogen **DFÜ im CLUB 80** und einige Impressionen von unserem 94ér Clubtreffen. Die Bilder hat uns Tilmann Reh zur Verfügung gestellt. Vielleicht fühlt sich der ein oder andere dadurch animiert, beim nächsten Mal mit dabei zu sein.

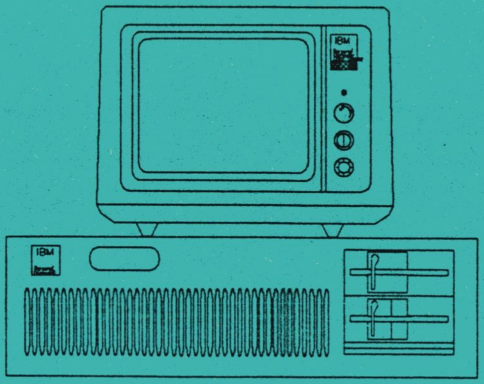
Bis zum nächsten INFO
grüßt Euch Euer

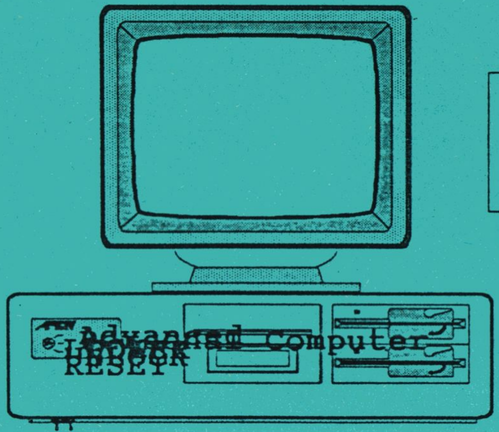
Clubtreffen '94

HARDWARE * SOFTWARE

COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80

ERFAHRUNGSAUSTAUSCH





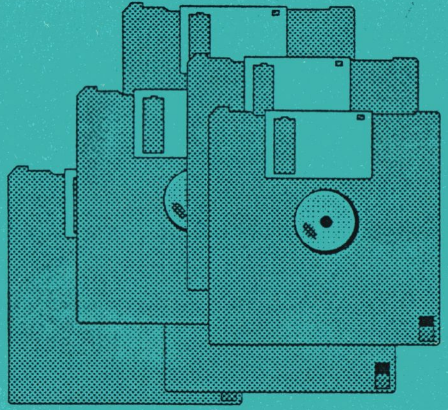
nicht vergessen !!!

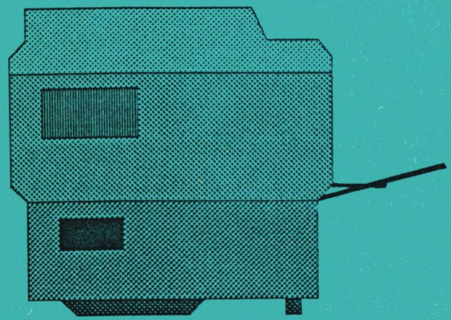
Hobbytronik Stuttgart

Sonntagstreff

27.11.1994

9.00Uhr oder 12.00Uhr
nach der Kartenkontrolle
beim unteren Eingang.





Kontaktadresse:

Club 80

Hartmut Obermann

Wilhelm-Baur-Straße 8

Postfach _____

76135 Karlsruhe

Tel.: 0721/85 40 6

BTX: 0721/_____

Anrufbeantworter

FAX: 0721/85 40 6

CLUBINFO

45. Ausgabe

Inhaltsverzeichnis

	Autor & Seite
Hardware	
Centronics-Schnittstellen autom. ...	1 - 2
Druckeranschluß für Z80-Systeme	3 - 4
Serielle Schnittstelle für 8085-System	5 - 6
Hardware-Oldies	
... aus INFO´s 1-10	
Der TRS-80 und die Außenwelt	7 - 8
Interface mit 8251 und 8255 für Z80-Rechner	9 - 12
Syncronisierte Taktumschaltung für VideoGenie	13 - 17
Billiger NF-Verstärker für den TRS-80	18
Umbau eines TRS-80 Modell1 16 auf 48k (64)	19 - 21
Ein sicheres Plätzchen	23 - 28
Ein sicheres Plätzchen (für TRS80)!	36
Supertape für TRS-80	29 - 36
BASICODE-2	37 - 39
BASICODE-Interface	40 - 41
Allerlei Nützliches für den TRS-80	42 - 43
Billiges CP/M von Bill Brewer	44 - 53
Die letzten Seiten	
Impressum	55
Schluß	56
Mitgliederadressenliste	Redaktion am INFO-Ende

Termine:

Multimedia World ´94	Frankfurt	29.09.- 01.10.1994
IVD Kongress ´94	Kassel	01.10.- 03.10.1994
Frankfurter Buchmesse	Frankfurt	05.10.- 10.10.1994
Document Imaging ´94	Berlin	10.10.- 13.10.1994
NETWORLD+ INTEROP ´94	Paris	24.10.- 28.10.1994
5. Elmshorner Computertage	Elmshorn	29.10.- 30.10.1994
Computer ´94	Köln	04.11.- 06.11.1994
LERNTEC	Karlsruhe	08.11.- 10.11.1994
TELMA	Magdeburg	10.11.- 13.11.1994
SIMO TCI	Madrid	08.11.- 13.11.1994
Networks Expo ´94	Frankfurt	22.11.- 24.11.1994
ComCoBit	Fulda	26.11.- 27.11.1994
Windows Solutions	Frankfurt	29.11.- 02.12.1994
5th CD-I World Conference	Düsseldorf	04.12.- 07.12.1994

Bitte nicht vergessen:

Der obligatorische **Hobbytronk-Sonntags-Treff** am **27.11.1994** in Stuttgart um **9.00Uhr** und/oder **12.00Uhr** am unteren Eingang (bisher war dort immer der Conradstand gegenüber)

Redaktionsschluß für das nächste INFO ist der:
01. Dezember 1994

Centronics-Schnittstellen automatisch umgeschaltet

Oft ergeben sich praktische Probleme, wenn zwei Mikroprozessorsysteme einen gemeinsamen Drucker nutzen sollen. Im folgenden Beitrag wird eine Schaltung beschrieben, die das automatische und vollelektronische Umschalten zwischen zwei Centronics-Schnittstellen ermöglicht. Die Centronics-Schnittstelle ist durch acht parallele Datensignale sowie einige Steuer- bzw. Handshake-Signale definiert. Die wichtigsten davon sind STROBE, BUSY, ACKNOWLEDGE und INPUT PRIME. Diese Steuerleitungen werden in der nachfolgenden Schaltung berücksichtigt.

Übertragungsprotokoll

Das Protokoll der Schnittstellen läßt sich anhand des Signal-Zeit-Diagramms in *Bild 1* erklären.

Die Bereitschaft, Daten aufzunehmen, zeigt der Drucker durch LOW-Signal auf der Steuerleitung BUSY an. Durch das kurze LOW-Setzen des STROBE-Eingangs des Druckers werden die anstehenden Daten übernommen. Der Drucker quittiert die Übernahme über den ACKNOWLEDGE-Ausgang. Ein LOW-Impuls am Eingang INPUT PRIME setzt den Drucker zurück.

Schaltung

Derjenige Rechner, welcher zuerst den Drucker belegen möchte, soll automatisch den Drucker zugeteilt bekommen. Jeder Rechner gibt seine Druckeranforderung durch ein logisches LOW auf der STROBE-Leitung bekannt.

Hat der akzeptierte Rechner seine Übertragung beendet, was durch das Fehlen von Datenzeichen während einer bestimmten Beobachtungszeit erkannt wird, so steht der Drucker wieder zur freien Verfügung. LEDs zeigen den jeweiligen Betriebszustand an.

Die Schaltung (*Bild 2*) zeigt den Fall, daß zwei Rechnersysteme mit einem gemeinsamen Drucker arbeiten. Den Kern des Umschalters bilden die drei Multiplexer IC1 bis IC3; diese schalten jeweils die acht Datenleitungen Data1 bis Data8 sowie die drei Steuerleitungen STROBE, INPUT PRIME und ACKNOWLEDGE. Die Auswahl der durchzuschaltenden Leitungen übernimmt das Flipflop IC4, das von den STROBE-Signalen der beiden Rechner gesteuert wird.

Für das Status-Signal BUSY ist die etwas aufwendigere Logik um IC5 zuständig. Wird der Drucker von keinem Rechner belegt, dann erhalten beide Rechner das Freigabesignal des Druckers (BUSY = LOW). Ist der Drucker bereit, dann wird das retriggerbare Monoflop IC7 freigegeben (Eingang B1 auf HIGH). Belegt nun einer der beiden Rechner den Drucker, indem dieser Daten und den STROBE-Impuls aussendet, geschieht folgendes:

Über das Flipflop IC4 wird der angeforderte Rechner zum Drucker durchgeschaltet und das Monoflop IC7 getriggert. Es bleibt bei der angegebenen Dimensionierung für rund 12 s aktiv.

Über die Logik IC5 erhält der jeweils andere Rechner ein aktives BUSY-Signal, was seine Datenübertragung suspendiert.

Solange der ausgewählte Rechner Daten aussendet, erhält IC7 ständig Triggerimpulse, und die Zuteilung bleibt erhalten. Erst wenn die Übertragung der Daten für länger als etwa 12 s aussetzt, kann der andere Rechner die Zuteilung seinerseits anfordern.

Die LEDs 2 und 3 zeigen an, welcher Rechner gerade zugeteilt ist. Ist die Belegung offen, leuchten beide LEDs. Die LED1 leuchtet, wenn der Drucker nicht frei verfügbar, also BUSY ist.

Die Versorgungsspannung von 5 V für die Schaltung läßt sich dem Drucker über den Anschluß 18 der Centronics-Schnittstelle entnehmen.

Dipl.-Ing. K. Baderschneider
Cand. el. T. Lindner

Literatur

[1] Hartmann, I.: Die Centronics-Druckerschnittstelle. Micro Extra (1982), H. 2, S. 13 bis 16.
[2] Siemens AG: Drucker PT 88/PT 89. Anwendungsbeschreibung. Firmenschrift (1983).

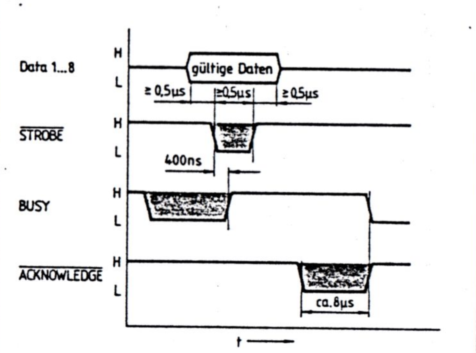


Bild 1. Zeitliche Zusammenhänge zwischen den Daten und den Steuersignalen bei der Centronics-Schnittstelle

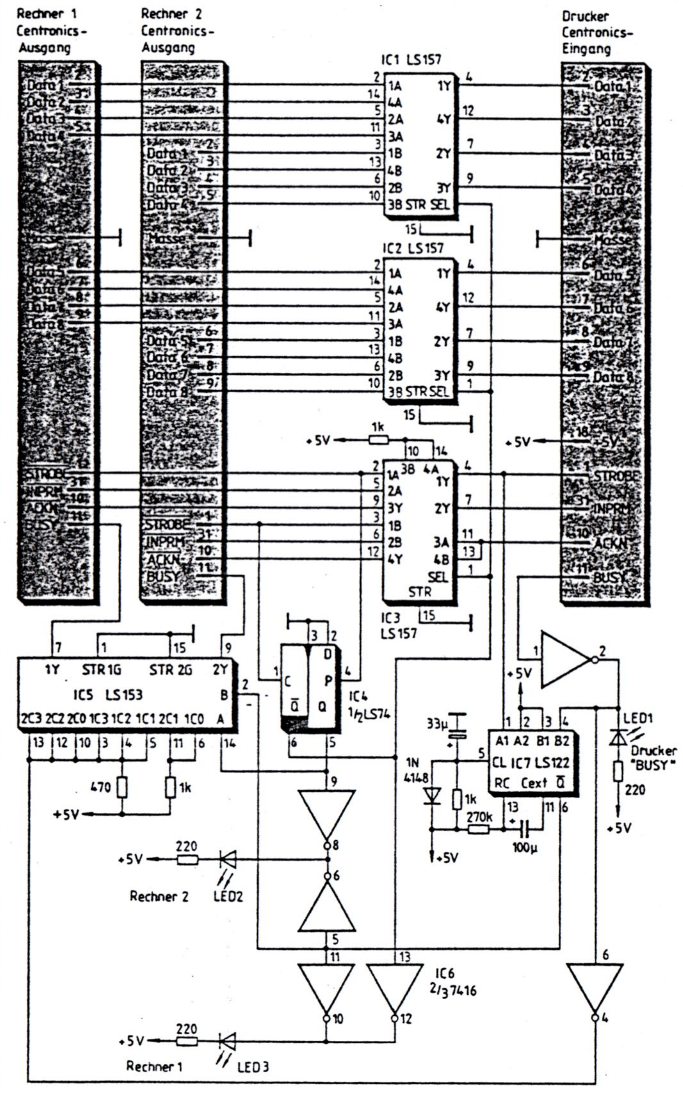


Bild 2. Schaltung eines Schnittstellen-Umschalters für zwei Rechner und einen Drucker

Druckeranschluß für Z80-Systeme

Der Beitrag beschreibt die Realisierung einer Centronics-Schnittstelle mit dem PIO-Baustein der Z80-Familie. An Zusatzhardware sind nur noch wenige Bauteile erforderlich. Die Treiberroutinen sind ausgelegt für die Ausgabe einzelner ASCII-Zeichen bzw. von Zeichenketten, die mit ETX (03H) abgeschlossen sind.

Die Centronics-Schnittstelle ist eine parallele Schnittstelle zum Ansteuern von Druckern. Die Daten werden mit 8 Bit parallel als ASCII-Zeichen übertragen. Die Schnittstelle arbeitet mit TTL-Pegeln. *Bild 1* zeigt, wie die Übertragung eines Zeichens mit Quittungssignalen gesteuert wird. Wenn die Daten auf den Datenleitungen gültig, d. h. eingeschwungen sind, wird durch ein $\overline{\text{DATA STROBE}}$ = Low dem Drucker angezeigt, daß ein Zeichen zur Übertragung bereitsteht. Das BUSY-Signal geht auf High und zeigt dem Rechner damit, daß das Zeichen gelesen wird. Ist das Zeichen übernommen, geht das BUSY-Signal wieder auf Low, und gleichzeitig wird durch ein $\overline{\text{ACKNOWLEDGE}}$ = Low dem Rechner angezeigt, daß das Zeichen übernommen wurde. Jetzt kann das nächste Zeichen übertragen werden.

Von den vier Betriebsarten, die der Z80-PIO-Baustein bietet, ist die Betriebsart 0, Byteausgabe mit Quittungsinterrupt, für die Centronics-Schnittstelle am besten geeignet, da in diesem Fall der zweite PIO-Kanal anderweitig verwendet werden kann. Das Einstellen der Betriebsart 0 zeigt *Bild 2*. Die zwei Quittungsleitungen des einen Kanals reichen aus, um mit den acht Datenleitungen zusammen die Schnittstelle zu realisieren. Das Zeitdiagramm für die Betriebsart 0 zeigt *Bild 3*.

Vergleicht man die Bilder 3 und 1, so sieht man, daß aus dem READY-Signal des PIO-Bausteins das $\overline{\text{DATA-STROBE}}$-Signal der Centronics-Schnittstelle werden muß. Das $\overline{\text{DATA-STROBE}}$-Signal muß allerdings zurückgehalten werden, solange das BUSY-Signal des Druckers auf High liegt. Das $\overline{\text{STROBE}}$-Signal für den PIO-Baustein stellt die Quittierung des Zeichens durch den Drucker dar und ist identisch mit dem $\overline{\text{ACKNOWLEDGE}}$ der Centronics-Schnittstelle. Bleibt nur noch eine Pufferung der Datenleitungen mit einem Treiberbaustein. Zur Erzeugung des $\overline{\text{DATA-STROBE}}$-Signals kann man die Schaltung nach *Bild 4* verwenden.

Der Baustein 74LS123 enthält zwei nachtriggerbare monostabile Kippstufen; davon wird in der Interfaceschaltung aber nur eine gebraucht, um einen $\overline{\text{DATA-STROBE}}$-Impuls von bestimmter Dauer zu erzeugen. Da für die Schaltung aber noch ein D-Flipflop erforderlich ist, wird die zweite monostabile Kippstufe als D-Flipflop eingesetzt. Dies ist möglich, wenn man eine Zeitkonstante T→∞ realisiert. Dann läßt sich das umfunktionierte Flipflop mit einer positiven Flanke am Triggereingang setzen (Q geht von 0 auf 1). Das Flipflop bleibt gesetzt, bis es über seinen $\overline{\text{R}}$-Eingang ($\overline{\text{R}}$ von 1 auf 0) wieder zurückgesetzt wird.

Die Zeitkonstante T→∞ läßt sich sehr einfach realisieren, indem man den Widerstand R_{ext} und den Kondensator C_{ext} wegläßt. Wird statt des 74LS123 der 74123 verwendet, ist ein Widerstand von etwa 10 kΩ zwischen den Anschlüssen J (7) und $\overline{\text{Q}}$ (12) erforderlich.

Damit das auf den $\overline{\text{R}}$-Eingang rückgeführte $\overline{\text{DATA-STROBE}}$-Signal keine Nachtriggerung über die interne

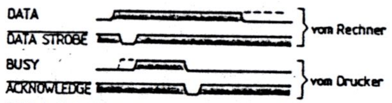


Bild 1. Zeitdiagramm der Centronics-Schnittstelle

```
PRGPIO  LD   A,0FH         ; Steuerwort für Betriebsart 0
        OUT  (PIOCHA+2),A
        LD   A,INTVEK      ; Interruptvektor
        OUT  (PIOCHA+2),A
        LD   A,87H         ; Interrupt freigeben
        OUT  (PIOCHA+2),A
```

Bild 2. So wird die Betriebsart 0 des PIO-Bausteins eingestellt. PIOCHA ist die Adresse des Kanals A

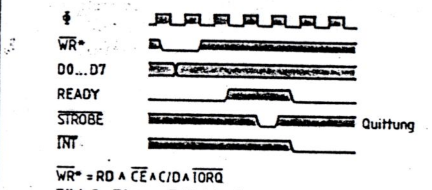


Bild 3. Dieser Zeitablauf ergibt sich in der PIO-Betriebsart 0

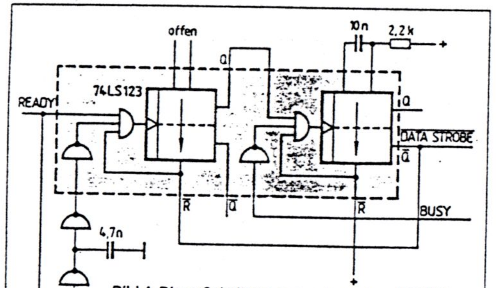


Bild 4. Diese Schaltung erzeugt aus dem READY-Signal des PIO-Bausteins $\overline{\text{DATA STROBE}}$ für die Centronics-Schnittstelle

Verbindung von $\overline{\text{R}}$ zu dem UND-Glied bewirken kann, wird das READY-Signal über eine Differenzierschaltung, auf den Eingang des Flipflops geführt. Dadurch ist der Triggereingang bereits gesperrt, wenn etwas später das $\overline{\text{DATA-STROBE}}$-Signal mit der positiven Flanke ansteht. *Bild 5* zeigt die Gesamtschaltung der Zusatzhardware.

Bevor ein ASCII-Zeichen an den Drucker ausgegeben wird, muß sichergestellt sein, daß der Drucker das vorhergehende ASCII-Zeichen angenommen, d. h. quittiert hat. Zu diesem Zweck wird ein Flag definiert, das sogenannte Druckerquittungsbit DQB. Das DQB ist einfach ein beliebiges Bit einer RAM-Adresse des zur Verfügung stehenden Speichers.

Ist das DQB nicht zurückgesetzt, wird es in einer Warteschleife so lange getestet, bis es durch einen Druckerinterrupt – der als Quittierung des empfangenen ASCII-Zeichens dient – zurückgesetzt wird. Beim Einschalten des Systems muß das DQB während der Initialisierung mit 0 vorbesetzt werden, damit das erste ASCII-Zeichen überhaupt ausgegeben werden kann.

Ist das DQB durch einen Druckerinterrupt bzw. durch die Initialisierung zurückgesetzt, wird es wieder gesetzt, und das Zeitglied zur Überwachung des Druckers wird angestoßen. Dann wird das ASCII-Zeichen an den Drucker ausgegeben.

Übernimmt der Drucker die zu druckenden ASCII-Zeichen in einen Puffer, so erfolgt der Druckvorgang erst mit Ausgabe des Zeichens CR (0DH) an den Drucker.

Die Routine zur Quittungsinterruptbehandlung wird aufgerufen, wenn der Drucker das empfangene ASCII-Zeichen quittiert. Hierbei werden das Zeitglied zur Druckerüberwachung und das DQB zurückgesetzt, danach wird der Interrupt wieder freigegeben.

Läuft das bei der Ausgabe eines ASCII-Zeichens an den Drucker angestoßene Zeitglied ab, so erzeugt es einen Interrupt, der zu einer Fehlermeldung und zu einem Ausstieg aus dem Druckerprogramm genutzt werden kann. Als Zeitglied kann z. B. ein Z80-CTC-Baustein (zwei Kanäle kaskadiert, der erste Kanal als Zeitgeber, der zweite Kanal als Zähler) benutzt werden. Die Laufzeit des Zeitglieds muß auf die Druckgeschwindigkeit des Druckers abgestimmt werden, d. h. sie sollte etwas größer sein, als für den Ausdruck einer Zeile benötigt wird.

Steht in dem vorhandenen Z80-System kein CTC-Baustein zur Verfügung, so kann für die Druckerüberwachung auch ein Schleifenzähler (Software) in die Warteschleife des DQB-Tests eingefügt werden. Wird ein bestimmter Zählerstand erreicht, verzweigt das Programm zu einer Fehlermeldung und steigt aus dem Druckerprogramm aus. Der Schleifenzähler muß vor jedem auszugebenden ASCII-Zeichen auf seinen Startwert gesetzt werden.

Die auszugebenden ASCII-Zeichen stehen in einem Speicherbereich (Puffer), der mit ETX (03H) abgeschlossen ist. Der Zeiger HL zeigt auf das erste zu druckende Zeichen im Puffer.

Die Pufferausgabe läuft nach folgendem Schema ab: Der Akku wird aus der Speicherstelle geladen, auf die HL zeigt. Ist das Zeichen ein ETX, so erfolgt ein Rücksprung ins Hauptprogramm, andernfalls wird die Routine zur Ausgabe eines ASCII-Zeichens an den Drucker aufgerufen und danach der Zeiger HL inkrementiert. Der Vorgang wiederholt sich so lange, bis ein ETX auftritt.

Manfred Scherer

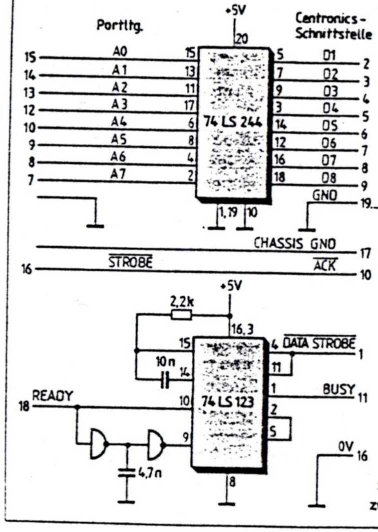


Bild 5. Nur wenige Zusatzbauteile sind neben dem PIO-Baustein erforderlich, um eine Centronics-Schnittstelle zu realisieren

Literatur

[1] Digitale Schaltungen MOS. Datenbuch der Siemens AG. Bereich Bauelemente. 1974/75.
[2] Z80-PIO Technical Manual. Zilog Incorporated. 1977.
[3] Z80-CTC Counter/Timer Circuit Technical Manual. Zilog Incorporated. 1977.
[4] Technical Manual Z80 Microprocessor Family. SGS-Ates. 1983.
[5] RX-80-Bedienungshandbuch. EPSON Deutschland GmbH. Düsseldorf.

Ing. (grad.) Karl-Alfred Knapp

Serielle Schnittstelle für 8085-System

Mikrocomputer werden in vielen Fällen über Datensichtgeräte bedient. Dabei genügt als Verbindung eine einfache Schnittstelle nach der V.24-Norm. Der Beitrag zeigt, wie man ein 8085-System um ein solches Interface erweitert.

1 Hardware

Einfache Datensichtgeräte (DSG) sind meist mit einer V.24-Schnittstelle ausgerüstet, die im Start-Stop-Betrieb und mit acht Datenbits pro Zeichen betrieben wird. Die Darstellung der Zeichen erfolgt nach dem ISO-7-Bit-Code [4]. Das achte Datenbit dient dabei als Paritätsbit, wobei nach gerader und ungerader Parität unterschieden wird (gerade Parität bedeutet: Anzahl der 1-Bits in den acht Datenbits ist gerade, das Paritätsbit ist entsprechend zu setzen).

Sollen Zeichen zwischen Mikrocomputer und Sichtgerät übertragen werden, dann muß ein E/A-Baustein die parallel auf dem Datenbus anliegende Information in einen seriellen Bitstrom umsetzen und umgekehrt. Diese Aufgabe kann der Baustein 8251A übernehmen [5]. *Bild 1* zeigt, wie der SIO-Chip (*Serial Input/Output*) an die CPU angeschlossen wird. Als Mikroprozessor dient der 8085. Für diesen Prozessor, wie auch für den 8080, umfaßt der E/A-Adreßraum die Adressen 0...255 (00H...FFH), die an den Adreßleitungen A0...A7 anliegen. Im folgenden wird angenommen, daß die E/A-Adressen 3EH und 3FH für die zu realisierende V.24-Schnittstelle verfügbar sind.

Die Bausteinauswahl (CS-*Chip Select*) erfolgt über die Adreßbits A1...A7 und das Steuersignal IO/$\overline{M}$ (IO/$\overline{M}$ = 1 bedeutet: E/A-Adresse auf Adreßbus) nach der Beziehung:

$$\overline{CS} = \overline{A_7}\,\overline{A_6}\,A_5\,A_4\,A_3\,A_2\,A_1\,IO/\overline{M}$$

Diese Beziehung kann leicht durch ein NAND-Gatter mit acht Eingängen realisiert werden.

Der Baudratengenerator hat die Aufgabe, die Übertragungsgeschwindigkeit für den seriellen Bitstrom auf den Leitungen T×D und R×D festzulegen. Zur Schaltungsvereinfachung wird ihm der Systemtakt des Mikrocomputers zugeleitet und entsprechend heruntergeteilt. In diesem Beispiel beträgt der Systemtakt 3072 kHz, er wird auf die in *Bild 2* angegebenen Frequenzen geteilt.

2 Software

Beim Baustein 8251A ist eine Reihe von Parametern programmierbar: z. B. die Betriebsart (asynchron oder synchron), die Anzahl der Bits pro Zeichen und die Paritätsprüfung (ja/nein oder gerade/ungerade).

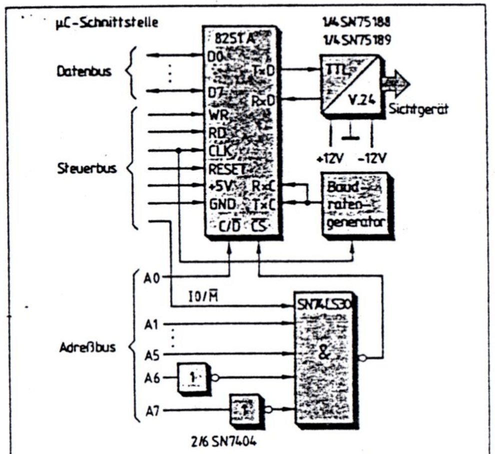


Bild 1. So wird der SIO-Baustein an den Mikrocomputer angeschlossen

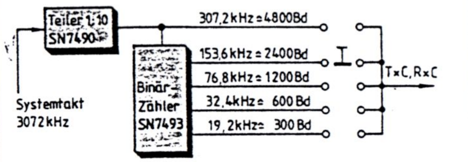


Bild 2. Zwei Zählerbausteine ergeben den Baudratengenerator

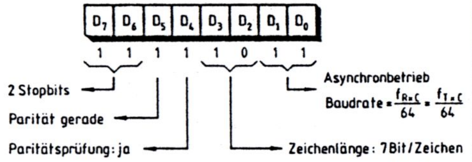


Bild 3. Im Mode-Register wird die Betriebsart des SIO-Bausteins festgelegt

Außerdem besteht jederzeit die Möglichkeit, seinen Zustand durch den Prozessor abzufragen, z. B. das Signal am Steueranschluß DSR (DSR = *Data Set Ready*, Betriebsbereitschaft).

Bevor man Zeichen empfangen oder senden kann, muß der 8251A zurückgesetzt und für ein bestimmtes Datenübertragungsverfahren programmiert werden. Das Rücksetzen übernimmt das Resetsignal des Mikroprozessors. Da das verwendete Sichtgerät nur die Betriebsart „asynchron" zuläßt, sind nach [5] zwei Steuerwörter (Steuerbytes) vom Prozessor an den Baustein zu übergeben. Anschließend können Zeichen gesendet und empfangen werden. Dazu übergibt der Prozessor dem SIO Zeichen in einem Byte über den Datenbus als Parallelinformation und holt Zeichen in Paralleldarstellung wieder ab.

Die Übergabe der beiden Steuerbytes muß in einer festen Reihenfolge geschehen. Zuerst ist das sogenannte Mode-Wort zu übertragen. Die Bedeutung der einzelnen Bits für den Asynchronbetrieb ist in *Bild 3* erläutert.

Das zweite Steuerwort erlaubt unter anderem die Freigabe und das Sperren von Sender und Empfänger und wählt verschiedene Hilfsfunktionen an. Dieses Steuerwort wird als Kommandowort bezeichnet (*Bild 4*).

Der SIO muß Steuerworte von den „wirklichen" Bildschirmdaten unterscheiden. Diese Unterscheidung wird durch ein spezielles Eingangssignal (C/$\overline{D}$) getroffen. Ist es log. 1, dann wird ein Steuerwort vom Datenbus an den SIO oder umgekehrt geschickt. C/$\overline{D}$ = 0 entspricht der Übergabe von Nutzdaten. Wie aus Bild 1 hervorgeht, ist dieser Eingang mit der Adreßleitung A0 verbunden, d. h. Daten und Steuerinformationen unterscheiden sich durch die Verwendung unterschiedlicher E/A-Adressen.

Aus der Beschaltung des $\overline{CS}$-Eingangs für die E/A-Adresse ergibt sich der Wert 3EH für Daten und der Wert 3FH für Steuerinformationen.

Damit wird der Baustein mit folgenden Befehlen auf seine Aufgabe vorbereitet:

```
MVI   A,0FBH   ;Mode-Wort in
OUT   3FH      ;8251A einschreiben
MVI   A,05H    ;Kommandowort
OUT   3FH      ;in 8251A einschreiben
```

Empfangen eines Zeichens:

```
CIN:  IN    3FH   ;Statuswort in Akku
      ANI   02H   ;D1 im Statusw. gesetzt?
      JZ    CIN   ;nein, nochmal erfragen
      IN    3EH   ;Empf. Zeichen in Akku
```

Anschließend ist die eigentliche Datenübertragung möglich. Hierbei muß allerdings beachtet werden, daß der 8251A nur dann ein neues Zeichen aufnehmen kann, wenn das Signal T×RDY (*Transmitter Ready*, Sender bereit) log. 1 führt. Dieses Signal läßt sich über das sogenannte Statuswort abfragen. Das gleiche gilt auch für das Empfangen von Zeichen. Hierbei ist das Signal R×RDY (*Receiver Ready*, Empfänger bereit) abzufragen. Es informiert den Prozessor, ob ein empfangenes Zeichen abgeholt werden kann.

Die möglichen Statusabfragen sind in *Bild 5* näher erläutert. Für einfache Anwendungen genügt die Auswertung von D1 und D0, das verdeutlichen folgende Programmzeilen.

Ausgabe des Zeichens ‚A' mit vorheriger Statusabfrage:

```
COUT: IN    3FH    ;Statuswort in Akku
      ANI   01H    ;D0 im Statuswort gesetzt?
      JZ    COUT   ;nein, nochmal erfragen
      MVI   A,41H  ;Zeichen ‚A' in Akku
      OUT   3EH    ;und zum 8251A übertragen
```

Ing. (grad.) Karl-Alfred Knapp

Literatur

[1] DIN 66020, Datenübertragung – Anforderungen an die Schnittstelle bei Übergabe bipolarer Datensignale.
[2] CCITT-Empfehlungen der V-Serie und der X-Serie, Band 2, Datenübertragung im Fernsprechnetz. R. v. Decker's Verlag, 4. Auflage, 1982.
[3] ELEKTRONIK-Arbeitsblatt Nr. 124. Die V.- und X.-Empfehlungen des CCITT. ELEKTRONIK 1979, H. 21, S. 87...90.
[4] DIN 66003, Informationsverarbeitung – 7-Bit-Code.
[5] Mikrocomputer-Bausteine, Band 3: Peripherie. Fa. Siemens.
[6] Experimentiercomputer ECB85, Bedienungsanleitung, Ausgabe 11.81. Fa. Siemens.

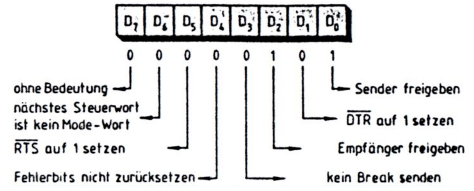


Bild 4. Weitere Steuerinformationen übergibt man dem SIO im Kommandowortregister

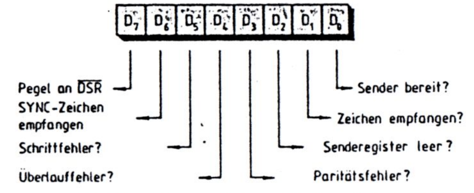


Bild 5. Auskunft über verschiedene Betriebszustände gibt das Statuswort des SIO-Bausteins

Wilfried Klein

Wenn man einen Computer häufig benutzt, so wird man mehr und mehr Anwendungen für ihn sehen. Und es kommt die Zeit, daß Tastatur und Bildschirm als Ein- und Ausgabe nicht mehr ausreichen. Sicherlich sucht man dann nach einer Möglichkeit, den Computer mit der Außenwelt korrespondieren zu lassen. Und man weiß, daß ein Computer dazu befähigt ist – man muß nur wissen, wie man eine dafür geeignete Schaltung und die Software entwirft.

Der TRS-80 und die Außenwelt

Um eine Kommunikation mit der Außenwelt zu ermöglichen. muß man sich um 2 Dinge kümmern: Hardware und Software. Der erste Teil ist die Software. Es gibt nun 2 Möglichkeiten. Eingabe-Ausgabe-Programme zu schreiben:

1) als Maschinen- oder Assembler-Routine
2) mit Benutzung der Basic-Befehle IN. OUT. PEEK + POKE

Beim TRS-80-Level I und bei anderen Mikrocomputern ist eine Benutzung der entsprechenden BASIC-Befehle nicht möglich: hier muß man eine Maschinensprache-Routine schreiben. während man beim Level II die BASIC-Befehle benutzen kann.

Die zusätzliche Hardware ist beim TRS-80 eine Notwendigkeit. denn weder der TRS-80 noch das Expansion-Interface haben Relais. Schalttransistoren oder Optokoppler eingebaut. um externe Geräte zu steuern. Und wenn man z. B. ein Relais an den vorhandenen Port anschließen würde. würde das zu steuernde Gerät jedesmal in Aktion treten. wenn man ein Programm von der Kassette in den Arbeitsspeicher laden würde.

Wichtig ist es. zunächst die Software zu definieren. die man für das jeweilige Projekt benötigt. und anschließend erfolgt die technische Realisierung. Es gibt 2 Möglichkeiten. eine Steuerung von externen Geräten vorzunehmen:

1) als „Memory-mapped"-System
2) als Port-System

Bei einem Memory-mapped-System hat die CPU die Möglichkeit. das zu steuernde Gerät wie einen Speicherplatz anzusprechen. Man sucht sich eine nicht benutzte Adresse und verwendet sie als Ansprechpartner für die CPU. Wenn man nun Daten an das externe Gerät senden will. muß man dieses Gerät über den 16-bit-Adressen-Bus adressieren. Um die Übertragung zu ermöglichen. ist es noch erforderlich. den Write-Eingang der CPU zu aktivieren. Ist es notwendig. Daten in den Computer zu leiten. muß man den Read-Eingang aktivieren.

Nehmen wir einmal an. das externe Gerät soll mit der Adresse 36863 (dezimal) entsprechend 8FFF (hex) aktiviert werden und das binäre Datenwort zum Einschalten dieses Gerätes soll 02 sein.
LD 8FFFH. 02H; schalte das Gerät ein
So sieht die Assembler-Routine aus. LD ist die Load-Instruktion (Laden). Mit anderen Worten:

Lade die Adresse 8FFF Hex mit der Binär-Information 02 Hex.

Durch die Assembler-Routine wird ein Objekt-Maschinen-Code erzeugt. der die CPU aktiviert. In Basic würde diese Routine so aussehen:
POKE 36863.2
In einem Port-System hat man nun die Möglichkeit. 1 von 256 8-bit-Ports zu wählen. die die CPU adressieren kann. Dieses hat den Vorteil. nur 8 von 16 bit des Adressenbus benutzen zu müssen.

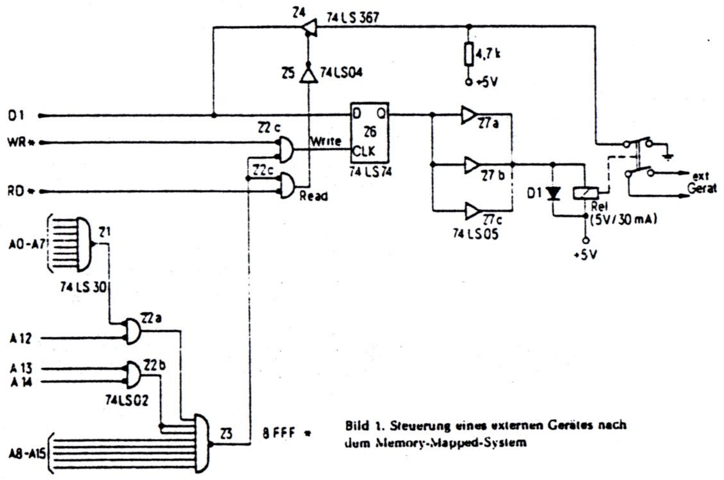


Bild 1. Steuerung eines externen Gerätes nach dem Memory-Mapped-System

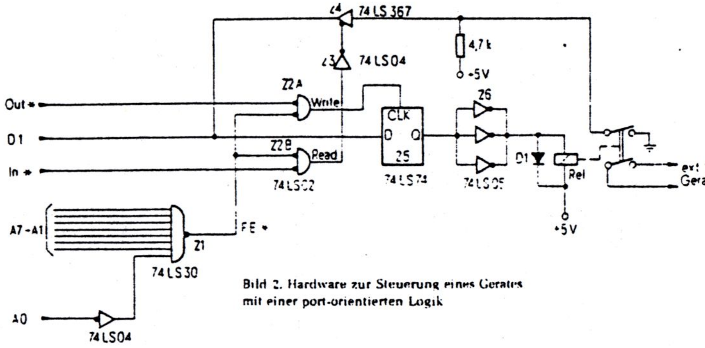


Bild 2. Hardware zur Steuerung eines Gerätes mit einer port-orientierten Logik

Wir wählen als Beispiel Port 255 (dezimal) entsprechend FF Hex. Die Assembler-Routine lautet:

OUT FF. 02 H; Einschalten des externen Gerätes

In Basic Level II würde es so aussehen:
OUT 255.2

Bei einem Port-System verwenden wir also die Instruktionen IN und OUT und nur 8 Adreßleitungen für den Bereich 0000...00FF.

Soll man nun das Memory-Map- oder das Port-System verwenden? Das bleibt eigentlich jedem selbst überlassen. Entscheidet man sich für Memory-Map. dann sollten die I/O-Adressen so hoch wie möglich liegen. um Störungen mit dem Arbeitsspeicher zu umgehen. Wenn man das Portsystem benutzt. kann man 255 Adressenleitungen verwenden. weil der Kassettenrecorder den Port FF benutzt. Von der Hardware-Seite her bietet sich das Port-System eher an.

Warum? Es ist notwendig. extern eine Adressen-Decodierung vorzunehmen. Und während man beim Port-System nur 1 von 256 Möglichkeiten bei acht Adressenleitungen decodieren muß. fällt beim Memory-Map-System die Decodierung von 1 aus 65 536 Möglichkeiten bei 16 Adressenleitungen an.

Das Memory-Mapped-System

Bild 1 zeigt die notwendige Hardware für ein Memory-Mapped-I/O-System. Die sechzehn Adressenleitungen werden hier mit NAND-Gattern decodiert. Mit einem Reed-Relais läßt sich ein externes Gerät steuern. Ein zweiter Relaiskontakt dient als Rückmeldung. ob auch tatsächlich ein Schaltvorgang stattfand.

Zusätzlich benötigt man eine selbständige 5-V-Stromversorgung. weil diese Schaltung nicht aus dem TRS-80 gespeist werden kann.

Ein kleines Beispielprogramm:

```
10  REM STEUERUNG VON EXTER-
    NEN GERÄTEN
20  CLS: PRINT „SOLL DAS GERÄT
    EINGESCHALTET WERDEN";
30  INPUT A$ : IF A$ = „NEIN" THEN
    GOTO 60
40  POKE 4092.2
50  B = 2 : A = PEEK (4092) : REM AN-
    FORDERUNG NR. 3
60  END
```

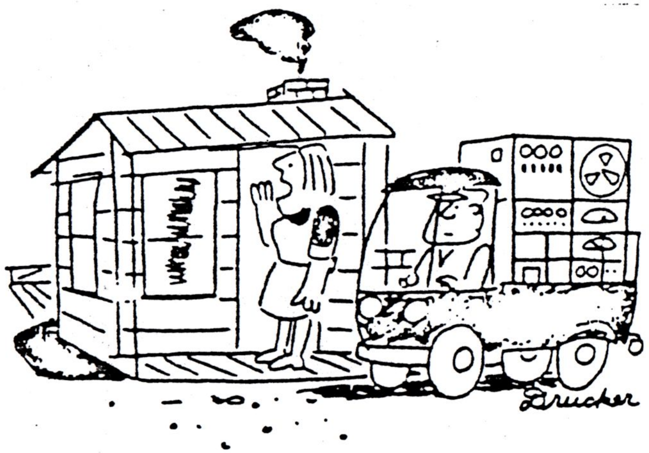
In Zeile 40 schreibt die CPU das Binärwort 2 in den Speicher unter der Adresse 4092 (binär: 1000 1111 1111 1111). Die ICs Z1. Z2a und Z3 dekodieren die Adresse für dieses 16-bit-Wort. welches auch als Signal am Ausgang von Z3 anliegt. Dieses Signal gelangt an die Eingänge von Z2c und Z2d. Die POKE-Instruktion bedingt. daß der Write-Eingang der CPU zur gleichen Zeit „Low" wird. wenn das 4092-Signal auf „low" geht. Dabei wird am Ausgang von Z2c ein „High"-Signal frei. welches als WRITE-Signal interpretiert wird und an den Takt-Eingang des Z6 eines Latches (Zwischenspeicher) gelangt. Das Datenwort „2". binär 0000 0010. gelangt als „High"-Signal über die Datenleitung D1 des Datenbus an den D-Eingang des Flipflops. Das Datenwort wird in Z6 abgespeichert. Das daraus resultierende „High"-Signal gelangt über den Ausgang Q von Z6 zu einem Relaistreiber Z7. dieser kann mit etwa 30 mA belastet werden. Während ein Kontakt das externe Gerät steuert. gibt der andere Kontakt ein Meldesignal über den Datenbus an den Computer zurück.

Das Port-System

Bild 2 zeigt das Schaltbild des Port-orientierten Systems. Wie man sieht. ändert sich hardwareseitig nicht viel. Im Basic-Betriebsprogramm ändern wir 2 Zeilen:

```
40  OUT 254.2
50  B = 2 : A = INP 254
```

Selbstverständlich ist es auf ähnliche Weise auch möglich. nicht ein externes Gerät zu steuern. sondern periphere Logikzustände abzufragen. Der Experimentierfreudigkeit sind also kaum Grenzen gesetzt.



„Charlie! Wo soll der neue Computer hin?"

H.O.

INTERFACE MIT 8251 UND 8255 FÜR Z80-RECHNER

Das Interface wurde entwickelt, um beliebige Geräte mit dem Computer verbinden zu können.
Es gibt bereits folgende Anwendungen:
Eprom - Programmierer
Joystick Anschluß
Morsezeichen - Decoder
Morsetrainer
Funkfernschreiben
Steuerung von el. Verbrauchern über Relais
Bei der Entwicklung wurde darauf Bedacht genommen, daß es nicht nur in Maschinensprache, sondern auch in Basic leicht programmiert werden kann.
Bei Verbindung der ser. Schnittstelle mit RS-232 od. V-24 ist zu beachten, daß der 8251 TTL - Signale verarbeitet. Sollte die Schnittstelle mit EIA-Pegeln (+- 3.....15 Volt) arbeiten, sind noch entsprechende Line-Driver nachzuschalten. (z.B.: 75189)
Literatur zur Programmierung dieser Bausteine gibt es bei Siemens: Die Beschreibungen der SAB 8080-Serie

SCHALTUNGSBESCHREIBUNG

Der Adreß - und Datenbus wird aus dem Computer über ein Kabel entnommen. Die Steckerbezeichnungen gelten für den KOMTEK. Bei anderen Geräten bitte im User-Manual nachsehen. Direkt am Eingang ins Interface sind alle Bus-Leitungen mit je 3,9 k nach Masse terminiert. Diese Widerstände habe ich im Schaltbild nicht eingezeichnet.
Alle Signale sind mit 74245 voll gepuffert, sodaß Rückwirkungen auf den Computer ausgeschlossen sind.
Das Signal I/O request ist mit einem 74121 um ca. 0.4 usec verzögert. Das ist notwendig bei Floppy-Betrieb, um die Timer-Interrupts vom Interface fernzuhalten.
Das Signal 'OSZ 2 MHZ' an Pin 45 ist in der Frequenz völlig unkritisch. Notfalls lässt sich auch die Leitung 'MI' verwenden. Diese Clock ist nur zur internen Steuerung des 8251 notwendig.
Die Adreß-Decodierung besorgt ein 74138. Die Ausgänge liegen an Lötstiften: sodaß eine leichte Änderung möglich ist.
Originalbelegung:

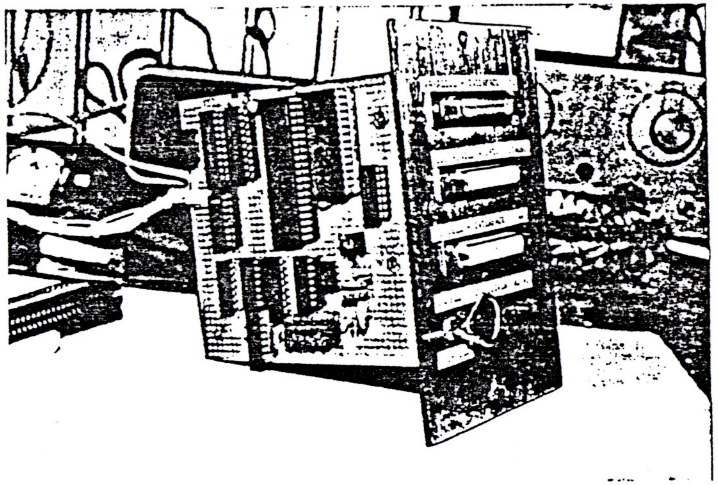
PIO 8255:	7C = Port A	
	7D = Port B	
	7E = Port C	
	7F = Control-Register (für Steuerung der Ports)	
SIO 8251:	64 = Datenregister	
	66 = Steuer- und Statusregister	
Timer :	60 = Steuerwort für Baudrate (1,2,4,8)	

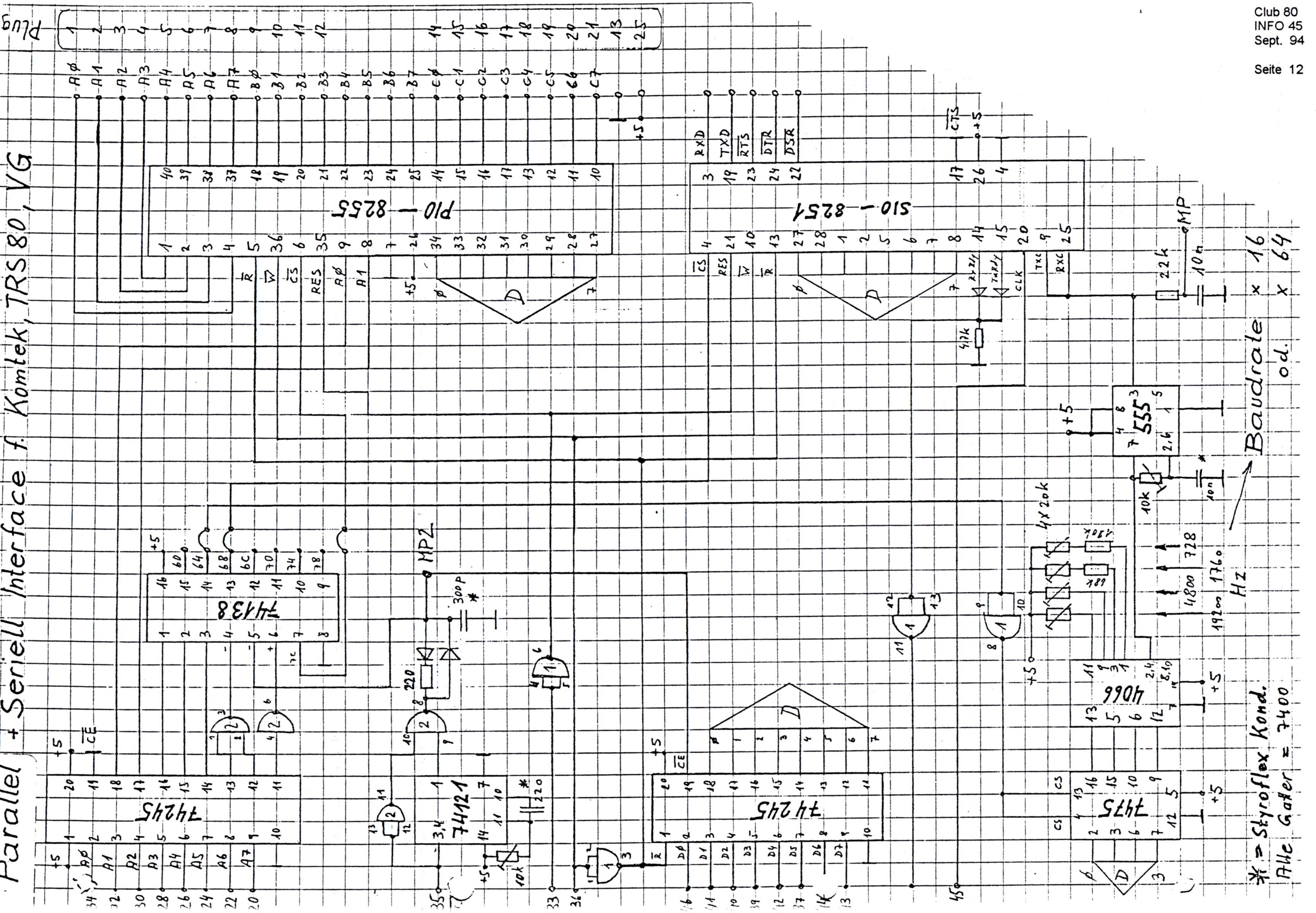
AUFBAU UND ABGLEICH

Die Verdrahtung habe ich auf einer Veroboard-Platte in der Größe einer Europa-Karte (160 X 100 mm) vorgenommen.
Es wurde die sogenannte Fädeltechnik angewandt, die für ein Einzelstück die schnellste Aufbauart darstellt.
Bei einer größeren Stückahl würde sich der Entwurf einer Platine durchaus lohnen. Es wurde allerdings noch kein Layout dafür erstellt.

Einstellung:
Der Timer kann mit einem OUT-Befehl auf 4 verschiedene Takt-Raten eingestellt werden. Diese ist am Punkt MP mit einem Freqenzzähler meßbar. Im Schaltbild sind die Werte für 45,110,300,1200 Baud angegeben. Durch Wahl der Potis und ggf der Vorwiderstände läßt sich jede beliebige Frequenz einstellen. Formel: Frequenz=Baudrate X 16 oder X64 oder X1 je nach Steuerwort im 8251.
Das Poti am NE555 bestimmt das Tastverhältnis. Es wird so eingestellt, daß bei der niedrigsten und auch bei der höchsten Taktrate ein sicheres Schalten erfolgt. (nicht sehr kritisch, aber etwas Einfluß auf die Frequenz)
Einstellung des Trimmers am 74121: Oszilloskop mit I/O req triggern. Signal am MP2 auf 0.4 usec Verzögerung mit dem Trimmpoti einstellen. (Fallende Flanke)
Ohne Oszi: Trimmer auf Mittelstellung

Walter Zwickel
Lengfelden 123
A5101-Bergheim




Parallel + Seriell Interface f. Komtek, TRS80, VG
PIO — 8255
SIO — 8251
74138
74245
74121
74245
4066
4175
555
Baudrate x 16 od. x 64
* = Styroflex Kond.
Alle Gatter = 7400

Syncronisierte Taktumschaltung für VideoGenie

Liebe Clubfreunde,

Im Clubinfo Nr. 5 hat Gerald Dreyer eine Schaltung veröffentlicht, mit der man die Taktfrequenz des VideoGenie verdoppeln kann. Nachteil dieser Schaltung ist die Tatsache, daß bei der Umschaltung zwischen einfacher und doppelter Taktfrequenz in den meisten Fällen der Rechner hängenbleibt. Außerdem kann man die Taktfrequenz nur mit einem Schalter, also hardwaremäßig, jedoch nicht softwaremäßig umschalten. Um diese Nachteile zu beseitigen, hat Manfred Held die unten abgebildete Schaltung entworfen.

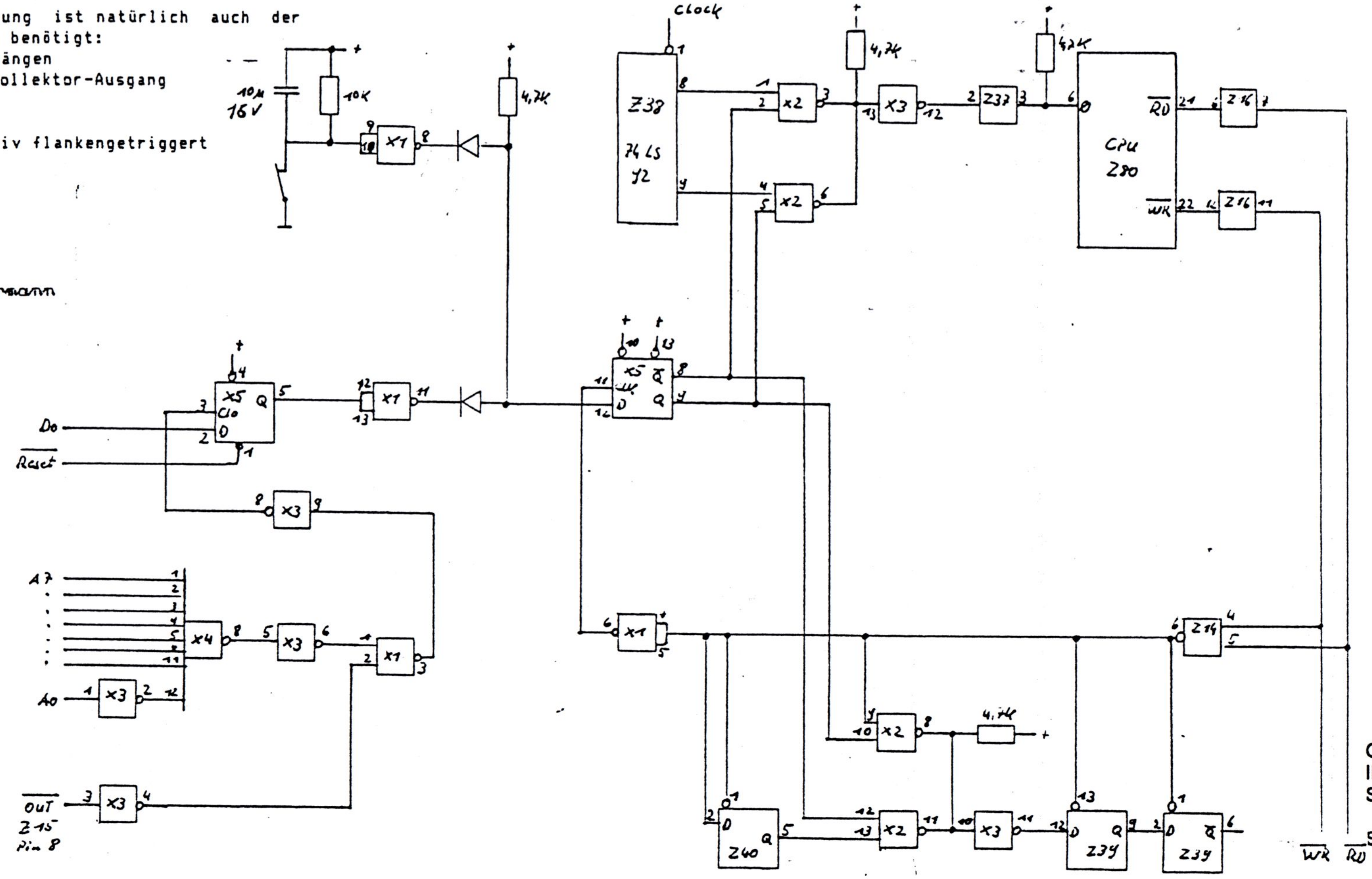
Die eigentliche Umschaltung der Taktfrequenz bewerkstelligen zwei NAND-Gatter (x2, rechts neben Z38). Die Umschaltung kann durch einen Schalter (linke obere Ecke) oder über den Port FEh (254d) (Decodierlogik linke untere Ecke) erfolgen. Um einen Systemabsturz wärend der Umschaltung zu verhindern, wird durch Z14 sichergestellt, daß der Umschaltvorgang erst erfolgt, wenn die CPU weder Schreib- noch Leseoperationen durchführt.

Durch die erweiterten Möglichkeiten der Schaltung ist natürlich auch der Bauteileaufwand gestiegen. Es werden folgende Teile benötigt:

IC x1 = 74LS00 = vierfach NAND-Gatter mit je 2 Eingängen
IC x2 = 74LS03 = vierfach NAND-Gatter mit offenem Kollektor-Ausgang
IC x3 = 74LS04 = sechsfach Inverter
IC x4 = 74LS30 = NAND-Gatter mit 8 Eingängen
IC x5 = 74LS74 = zweifach Speicher-Flip-Flop, positiv flankengetriggert
Alle IC's je 1 mal. Weiterhin:
2* 1N4148 = Siliziumdiode
3* 4700 Ω Widerstand
1* 10 kΩ Widerstand
1* 10 µF/16V Kondensator

Viel Spaß beim Basteln, wünscht euch Hartmut Obermann

Eine Operateuse aus Wien
dacht' beim Arbeiten stets nur an ‚Ihn'.
Ihr Computer, voll Frust,
verlor deshalb die Lust –
er war auch nur ein Mensch, wie es schien.



Club 80
INFO 45
Sept. 94


```
*****************************************************************************
*                          Ergaenzung und Aenderung                         *
*****************************************************************************
```

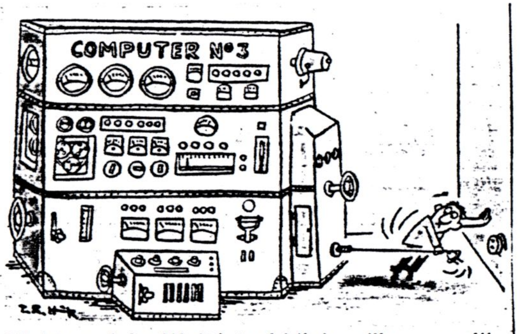
Synchronisierte Taktumschaltung fuer Videogenie

In der Clubinfo Nr. 8 ist der Schaltplan fuer eine Taktumschaltung fuer das Videogenie, der zwei Fehler enthaelt.

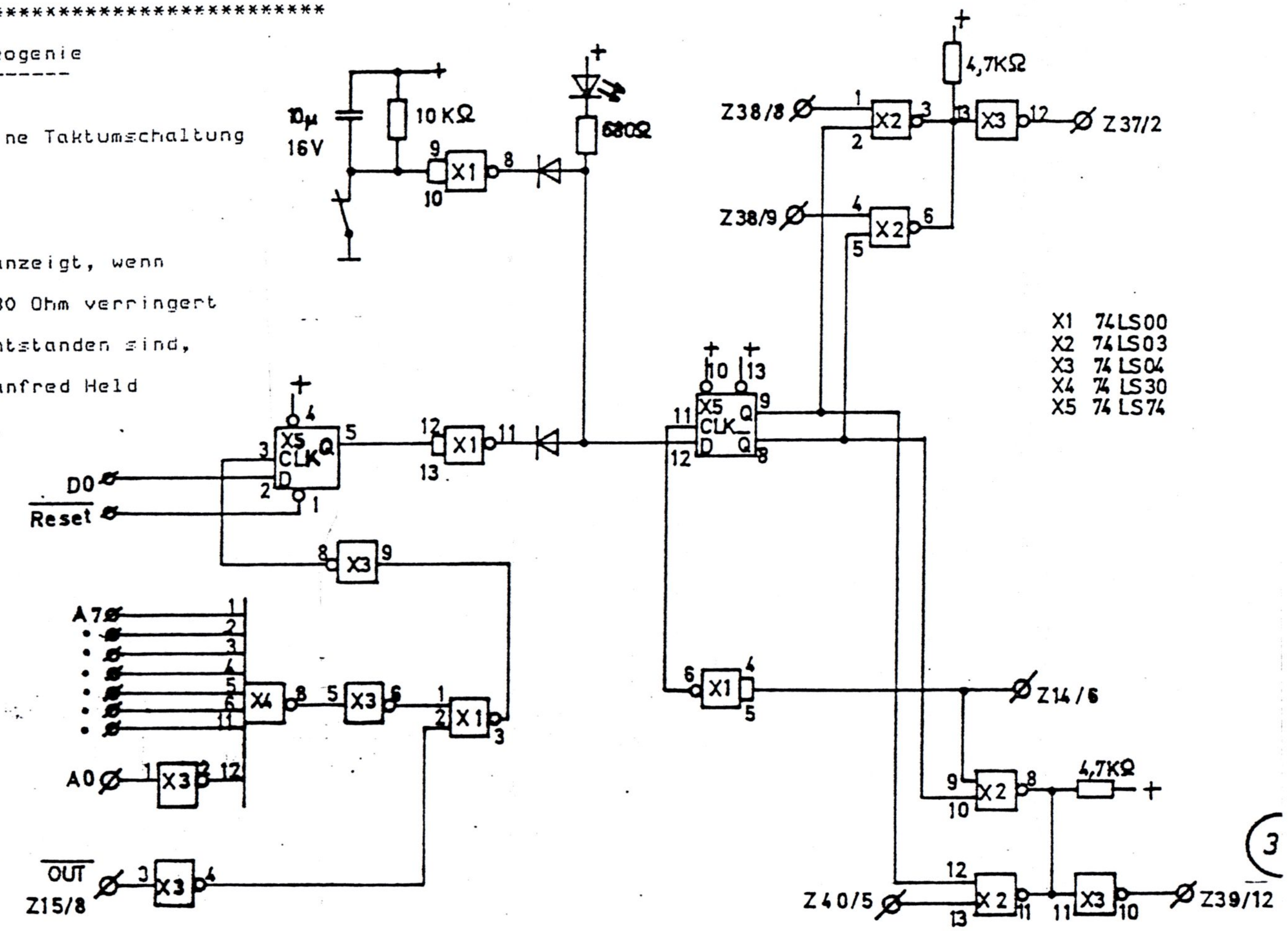
- Am Ic X3 ist Pin 10 und Pin 11 vertauscht.
- Am Ic X5 ist Pin 8 und Pin 9 vertauscht.

Als ergaenzung ist eine LED hinzugekommen, die anzeigt, wenn die CPU mit 3.55 MHz arbeitet.
Bedingt durch die LED muss der Widerstand auf 680 Ohm verringert werden.
Die Fehler, die beim Zeichnen des Schaltplans entstanden sind, bitte ich zu entschultignen.

Manfred Held



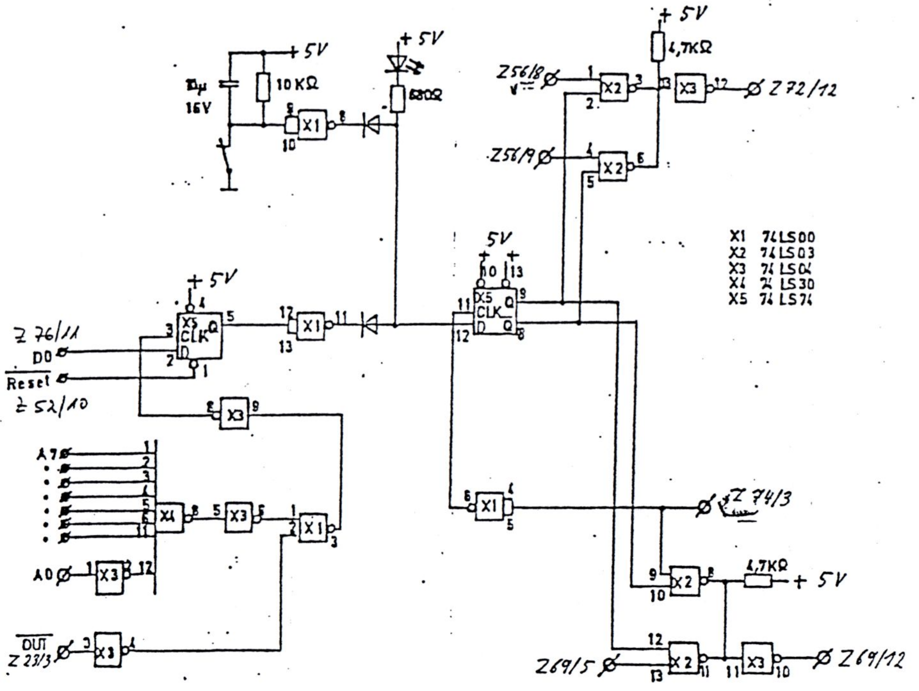
„Nichts auf der Welt ist wirklich vollkommen!"



```
*********************************************************************
*          Ergaenzung zur synchronisierten Taktumschaltung          *
*********************************************************************
```

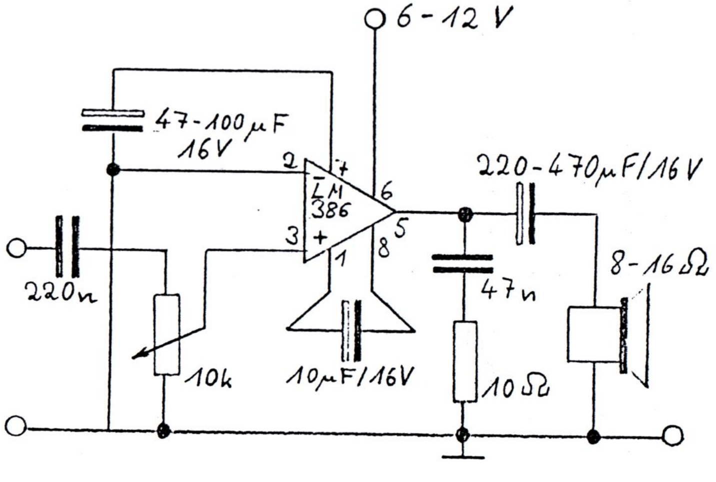
Ergaenzend zu den Artikeln ueber die synchromisierte Taktumschaltung in den Clubinfo's 8 und 9 moechte ich hier eine Aenderung vorstellen in der die Anschlusspunkte fuer den TRS-80 Modell 1 angegeben sind. Weiterhin moechte ich noch hinzufuegen das die Leiterbahnen zwischen Z 56/8 und Z 72/12 und zwischen Z 69/5 und Z 69/12 zu trennen sind.

Bernd Retzlaff



Billiger NF-Verstaerker fuer den TRS 80

Die nachfolgend aufgefuehrte Schaltung mit dem LM 386, einem NF-Verstaerker-IC kleiner Leistung, kann direkt an den Tapeausgang des TRS 80 oder eines anderen Computers geschaltet werden. Sie macht den Anschluss eines grossen Verstaerkers zur Tonwidergabe (z.B.: Musikprogr., Spiele) ueberfluessig. Der Ruhestromverbrauch liegt bei ca. 3 mA. Die Ausgangsleistung betraegt bei der angegebenen Beschaltung ca. 500 mW an einem 16 Ohm Lautsprecher. Es wird eine Versorungsspannung von 6 - 12 Volt benoetigt. Ich habe diese der Stromversorung des Monitors entnommen in dem die Schaltung auch untergebracht ist. Die Wiedergabequalitaet ist zwar nicht ueberragend aber ausreichend.



* Hartmut Obermann *

Umbau eines TRS-80 Modell 1 16K auf 48k (64k)

Nachdem ich im Sommer des letzten Jahres einen gebrauchten TRS-80 Modell 1 gekauft hatte, der mit 16 k RAM bestueckt war, stand ich vor dem Problem diesen Rechner auf 48 k RAM aufzuruesten.
Zu diesem Zeitpunkt hatte ich schon einen Genie-I mit 48-k (64-k) RAM, und ich dachte das es kein groesseres Problem sein sollte auch den Tandy auf die gewuenschte Speicherkapazitaet zu bringen. Aber erstens kommt es anders, und zweitens als man denkt.
Da ich die technischen Handbuecher fuer beide Geraete besitzte verglich ich erst einmal die Schaltunterlagen von beiden Rechnern. Auf den ersten Blick sahen auch beide Schaltplaene sehr aehnlich aus, aber gerade beim interessanten Teil, naemlich dem Adressdekoder fuer die RAM's und ROM's unterschieden sich beide Geraete erhaeblich. So musste ich feststellen das der Genie-Computer von seiner Grundauslegung her schon fuer den Umbau von 16-k auf 48-k mit den neueren 64-k-RAM Speicherchips ausgelegt war, waehrend der TRS-80 einen voellig anderen Adressdekoder hatte, der sich auf den ersten und zweiten Blick ueberhaupt nicht fuer diesen Umbau eignete.
Es musste also ein mehr oder minder neuer Adressdekoder eingebaut werden. Ich hatte zuerst vor, den Dekoder des Genie-I in den TRS-80 einzubauen. Aber auch dies erwies sich als nicht so einfach machbar, da sich beide Rechner in der Ansteuerung der ROM's und des Bildspeichers stark unterschieden.
Also was tun ?
Ich entschied mich dazu die ROM und Bildspeicheradressierung so zu lassen wie sie war und nur die Ansteurung der RAM's zu aendern.
Zuerst zum Adressdekoder Z21. Dieser Baustein zerlegt die unteren 32K, die die ROM's, Bildspeicher, Tastatur und die alten 16 K RAM enthalten in Bloecke zu je 4K. Die Ansteurung der alten RAM's konnte ja nun wegfallen, also machte ich folgende Ueberlegung, die neue RAM's sollen nicht angesprochen werden, wenn auf die unteren 16K zugegriffen wird, bei einer Adresse oberhalb von 3FFFH muss der RAM Bereich angesprochen werden.
Dies wird nun so erreicht: Wenn einer der ersten 4 Ausgaenge von Z21 LOW wird, so bedeutet dies, das auf die unteren 16K zugegriffen wird, also darf in diesem Fall der RAM Bereich nicht angesprochen werden.
Ich nahm also ein 4-fach-NAND und legte die Eingaenge dieses Chips auf die 4 Ausgaenge des Z21 (Pin 9 - 12), den Ausgang des NAND legte ich auf X3 Pin 15 von wo aus die RAM-Select Leitung abgeht (RAM *). Ist diese Leitung LOW, so werden die RAM-Bausteine selectiert, und zwar geht das Signal ueber X71 Pin 12 - Pin 5 auf Z67 Pin 15, den Select-Eingang eines Buffers ueber den MCAS geschaltet wird.
Somit war das Ziel der Adressierung schon erreicht. Wird eine Aresse der ROM's des Bildspeichers oder der Tastatur (0000H - 3FFFH) angesprochen, so wird der RAM-Bereich nicht selectiert, wird aber eine Adresse ueber 3FFFH (4000H - FFFFH) angesprochen so wird der RAM-Bereich selectiert.

Nun aber zum praktischen Teil des Umbau's. Als erstes mussten die alten RAM's (4116) gegen die neuen RAM's (4164) ausgetauscht werden.
Zuerst musste die Spannungsversorgung der RAM's geaendert werden.

Tabelle 1:

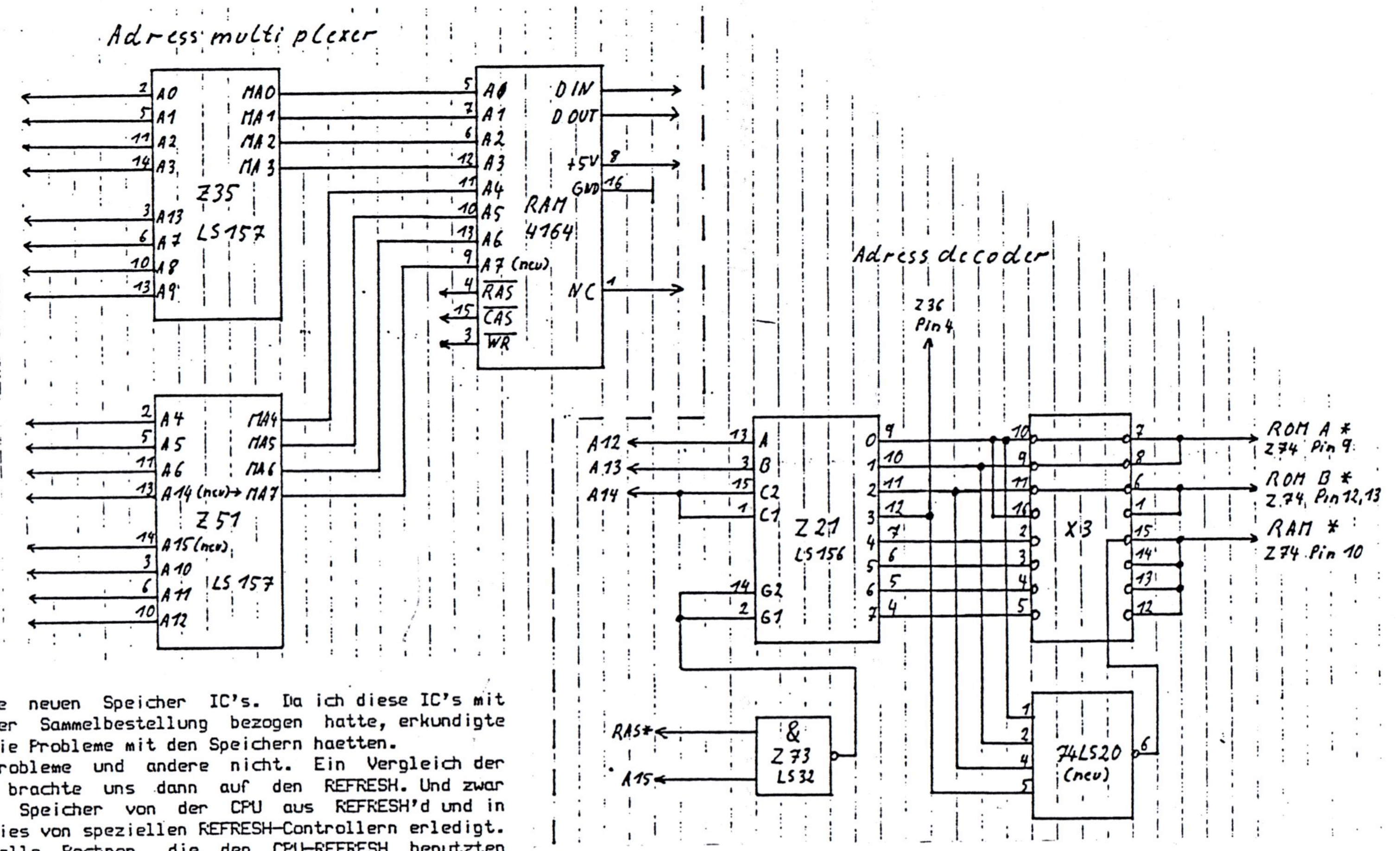
4116 -> :	Pin 1	—	-5 V
	Pin 8	—	+12 V
	Pin 9	—	+5 V
	Pin 16	—	GND
4164 -> :	Pin 1	—	N.C.
	Pin 8	—	+5 V
	Pin 9	—	A 7
	Pin 16	—	GND

Wie aus dieser Tabelle zu ersehen muss der Pin 1 freigemacht werden. Auf der Platine sind die einzelnen Pins der 8 Speicher IC's durch Leiterbahnen miteinander verbunden. Also wurde festgestellt von welchem IC aus die Leiterbahn Als naechstes musste aus dem Stromversorgungsanschluss Pin 9 der Adresseingang A 7 gemacht werden. Dies geschah aehnlich dem Umbau von Pin 1. Zuerst stellte ich fest, von wo aus die Versorgungsspannung aus dem Netzteil kam, dann trennte ich diese Leiterbahn auf, entfernte noch die Siebkondensatoren (100nF) die auf dieser Leiterbahn gegen Masse lagen und dann wurde der Pin 9 der Speicher IC's mit dem Adressmultiplexer Z51 Pin 2 verbunden. Damit der Adressmultiplexer auch die Adresse A 7 erzeugen kann muss er auch etwas modifiziert werden. Es muessen die Adressleitungen A 14 und A 15 auf die Pins 14 und 15 gelegt werden. Diese drei Anschluesse sind normalerweise nicht belegt.
Als letztes musste noch der Pin 8 der Speicher-IC's von der +12 V Versorgung getrennt, und mit der +5 V Versorgung verbunden werden. Dies war fast der schlimmste Teil des ganzen Umbaus, da an jedem einzelnen der 8 Speicher-IC's der Pin 8 von den Leiterbahnen getrennt werden musste und die neuen Anschluesse mit Schaltlitze auf der Platine nachverdrahtet werden musste.
Als letztes wurden dann die neuen Speicher-Bausteine in die Fassungen eingesetzt. Allerdings ist zu empfehlen, vor dem Einbau der IC's den Rechner schon einmal einzuschalten und die Spannungen an den IC-Sockeln zu pruefen, ob auch alle Aenderungen korrekt sind.
Dann koennen die neuen Speicherbausteine getestet werden. Also Rechner einschalten. Es sollte jetzt wie gewohnt 'MEM-SIZE' auf dem Bildschirm erscheinen. Nach dem Druecken der 'ENTER'-Taste dauert der Speicherausbautest Jetzt etwa dreimal solange wie bisher. Also keine Panik wenn es jetzt etwas laenger dauert bis 'READY' erscheint. Nachdem der Rechner sich mit 'READY' gemeldet hat, sollte man 'PRINT MEM' eingeben und die 'ENTER'-Taste druecken. Wenn alles richtig geklappt hat, sollte jetzt die Zahl '48340' auf dem Bildschirm erscheinen.

Jetzt noch ein paar Tips zu den Speicher-Bausteinen selbst :
Bei meinem ersten Umbau klappte alles wunderbar, aber nach ein paar Tagen als ich ein ziemlich grosses Programm im Speicher hatte fing mein Rechner ploetzlich an zu Spinnen. Einer der auftretenden Effekte war ein sporadisch auftredender Kaltstart ab Aresse 0000H ('MEM-SIZE'), ein weiterer Fehler war, dass sich der Rechner irgendwo im Programm aufhaengte und nur durch ausschalten und wiedereinschalten erneut zum Arbeiten zu bewegen war.
Diese und aehnliche Effekte traten aber nur auf, wenn die 'MEM-SIZE' Eingabe ueber 32767 lagen, also im neu dazugewonnenen Speicherbereich lag.

> Der Verstand ist unser größtes Vermögen, aber Armut schändet nicht.

> Bildung ist die Fähigkeit, sich alles anzuhören und dabei weder die Selbstbeherrschung zu verlieren noch das Selbstvertrauen.



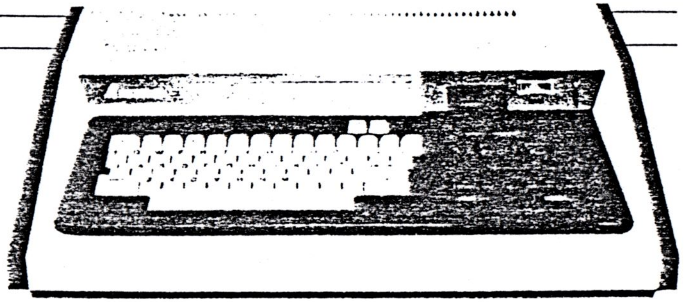
Also fiel der Verdacht auf die neuen Speicher IC's. Da ich diese IC's mit einigen Arbeitskollegen in einer Sammelbestellung bezogen hatte, erkundigte ich mich beide diesen, ob auch sie Probleme mit den Speichern haetten.
Und siehe da, einige hatten Probleme und andere nicht. Ein Vergleich der Hardware der einzelnen Rechner brachte uns dann auf den REFRESH. Und zwar wurden in einigen Rechnern die Speicher von der CPU aus REFRESH'd und in einigen anderen Rechner wurde dies von speziellen REFRESH-Controllern erledigt. Wir fanden also heraus, das alle Rechner, die den CPU-REFRESH benutzten Probleme mit den Speicher IC's hatten. Nach dem Durchsehen von diversen Datenblaettern fanden wir schliesslich heraus wo der Wurm zu suchen war.
Die Z80-CPU erzeugt einen 7 Bit REFRESH. Nun gibt es allerdings einige Hersteller die einen 8 Bit REFRESH fuer ihre 64K dymamischen RAM's brauchen Wenn man nun diese 8 Bit REFRESH IC's direkt an den Z80 anschliesst dann werden die Adressen 8000H -> FFFFH nicht refrescht, dass bedeutet das die Informationen, die auf diese Speicherplaetze geschrieben werden nach einer gewissen Zeit verschwinden.
Also Leute, aufpassen beim Kauf von 4164 IC's, fuer den TRS-80, und den Genie I muessen es IC's mit 7 Bit REFRESH sein.

Bernd Retzlaff

Computerurteil

Ein Mann besitzt zwei Uhren, die beide defekt sind. Die eine geht grundsätzlich 10 Minuten vor, das Laufwerk der anderen ist kaputt. Der Mann beschließt, das Elektronengehirn zu befragen.
Frage an den Computer:
»Welche Uhr soll ich behalten?«
Antwort des Computers: »DIE KAPUTTE UHR!«
Rückfrage: »Weshalb?«
Antwort des Computers: »DIE ERSTE UHR GEHT STÄNDIG FALSCH. DIE KAPUTTE UHR ZEIGT ZWEIMAL TÄGLICH DIE RICHTIGE ZEIT AN.«

Ein sicheres Plätzchen



Maschinenroutinen geschützt im GENIE untergebracht

Helmut Bernhardt

Die Sicherung von Maschinenprogrammen im oberen Adreßbereich durch Befehle wie HIMEM oder MEMSIZE gewährt lediglich Schutz gegen Überschreiben durch BASIC-Programme. Der Computer legt Maschinenprogramme beim Laden immer in den Speicherbereich, für den sie vorgesehen sind. Die Pointer werden dabei nicht berücksichtigt, so daß es normalerweise keinen Speicherbereich gibt, der wirklich 'sicher' ist. Mit einer kleinen Zusatzschaltung kann man jedoch ein 'sicheres Plätzchen' schaffen.

Auf der Suche nach diesem Platz im Adreßraum des TRS-80 oder GENIE I, II findet man schnell einige 'Lücken'. So beginnt der Adreßbereich der Tastatur bei 3800h und endet bei 38FFh. Da aber die Dekodierung des Freigabesignals für die Tastatur-Lesetreiber unvollständig ausgeführt ist, wird auch der Bereich von 3900h bis 3BFFh blockiert.

Das muß man nicht so hinnehmen. Durch geringfügige Eingriffe in den Rechner läßt sich die Tastatur vollständig dekodieren und gleichzeitig ein Freigabesignal für einen zusätzlichen Speicher-Baustein im Bereich 3900h bis 3BFFh gewinnen. Die dazu notwendige Hilfsschaltung zeigt Bild 1. Der zusätzliche Speicher besteht aus zwei ICs 2114 (1 K x 4 Bit); die weitere Dekodierung geschieht mit zwei TTL-ICs.

In den so gewonnenen 768 Byte RAM sind eigene Routinen ziemlich sicher, da man normalerweise nicht auf diesen Speicherbereich zugreifen kann. Programme mit eigener Treiberroutine könnten allerdings versuchen, die Tastatur im Bereich 3900h bis 3BFFh auszulesen. Solche Routinen würden laufend 'Eingaben' aus dem RAM erhalten, was zu Fehlfunktionen führt.

In solchen (seltenen) Fällen hilft dann nur noch der Einbau eines Zweifach-Umschalters, der entweder das unvollständig dekodierte oder das neu gewonnene Freigabesignal an die Tastatur-Lesetreiber legt und die Freigabe der RAMs bei Benutzung des 'alten' Freigabesignals für die Tastatur abschaltet. Programme, die diesen Schalter benötigen, lassen sich dann nicht mit dem zusätzlichen RAM verwenden. Das Betriebssystem NEWDOS-80 gehört glücklicherweise nicht zu dieser Gruppe von Software.

Eingriffe

Die vier ICs der Hilfsschaltung kann man auf einem Streifen Lochraster-Platine unterbringen. Die Versorgungsspannung (+5 V) ist jeweils an Pin 14 der TTL-ICs und an Pin 18 der RAM-Bausteine anzuschließen. Der Masseanschluß ist bei den RAMs Pin 9, bei den TTL-Bausteinen Pin 7. Zwischen Versorgungsspannung und Masse sollte man einen Kondensator (100 nF) schalten.

Beim Einbau der Karte in GENIE-Rechner kann man folgendermaßen vorgehen:

Zuerst ist das Gehäuse des Computers zu öffnen. Nachdem man die acht tiefer versenkten Schrauben in der Bodenwanne des Gerätes herausgedreht hat, läßt sich die obere Gehäuseschale abheben. Schraubt man noch die Tastatur ab, liegt links das CPU-Board, in der Mitte das Interface-Board und rechts das Netzteil sowie der Kassettenrecorder.

Die durchzutrennende Leitung (siehe Bild 1) ist die Leiterbahn, die vom hintersten IC (Z35, 74LS32) auf dem CPU-Board in der rechten Spalte nach 'hinten' verläuft (neben der 5-V- und Masse-Leitung). An die beiden Seiten der Trennstelle kann man den Ein- und Ausgang des 'OR'-Gatters der Hilfsschaltung anschließen. Das IC 'Z12' findet man an zweiter Stelle (von vorne) in der linken Spalte auf dem CPU-Board. Der Baustein 'Z15' (74LS32) liegt in der zweiten Spalte (von links gezählt) als zweites IC.

Die Numerierung der Verbindungsleitungen zum Interface-Board, an denen die meisten Signale für die Zusatz-Schaltung abzunehmen sind, erfolgt von vorne nach hinten (Nummern 1 bis 32).

Beim Anschließen der Erweiterung sollte man unbedingt darauf achten, daß die Datenleitungen nicht an den ROMs, sondern an den Lötpunkten der Stecker für die Verbindungsleitung zum Interface-Board abgenommen werden. Die an den ROM-Chips anliegenden Datenleitungen sind wegen der Freigabeschaltung der Treiber nur in Leserichtung verwendbar. An der Steckverbindung lassen sich außer $\overline{CS}$ und $\overline{MWR}$ alle für die RAMs nötigen Signale abnehmen. □

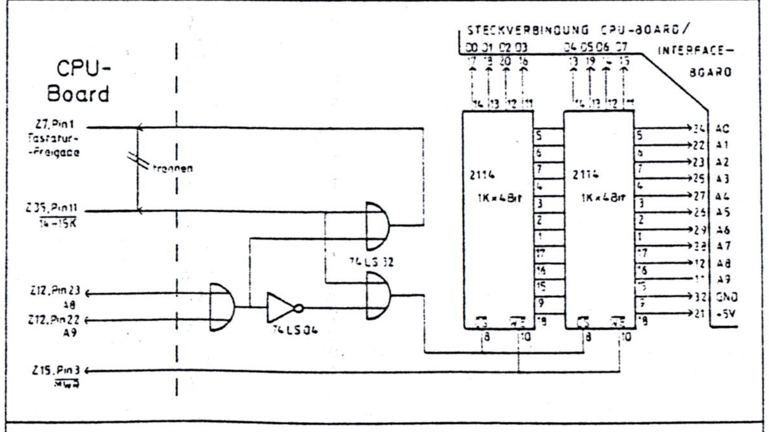


Bild 1. Die Zusatzschaltung für ein sicheres Plätzchen

Extra RAM oder Ein sicheres Plätzchen im TRS80 Modell

Artikel aus 80-U.S.Jurnal von Truman Krumholz
sehr frei übersetzt von Hartmut Obermann

In diesem Artikel möchte ich den Aufbau eines Memory-Boards beschreiben, das den Zwichenraum zwischen ROM und Tastatur ausfüllt. Dieses zusätzliche RAM ist besonders zur geschützten Unterbringung von Maschinenroutinen geeignet, da es nicht vom System bzw. DOS benutzt wird. Das Board benutzt die Adressen 3000H-377FH und stellt 1920 Bytes freien Speicherplatz zur Verfügung.

Das Board läßt sich schnell und problemlos auf einer Lochrasterplatine in Fädeltechnik aufbauen. Der Anschluß an das System erfolgt über den meist unbenutzten Screenprinteranschluß des Expansionsboards *1. Dazu benötigt man entweder ein selbstgestricktes Kabel mit passendem Stecker oder man lötet die Leitungen direkt auf. Als Stromversorgung werden +5 Volt bei 0.5 Ampere benötigt *2. Als Speicherchips werden zwei 2114s (1024*4 Bit Static RAM) verwendet.

Die Adressdecodierung wurde mit gängigen TTL-Bausteinen (7400 und 7410) aufgebaut. Alle Leitungen zwischen dem Board und dem Computer sind gebuffert (74LS367 und 8216) um Überlastung und Beeinflussung des Systems zu vermeiden. Die 8 "Pullup"-Wiederstände (je 1 kΩ) werden benötigt, da ansonsten die High-Signale auf dem Datenbus manchmal nicht die geforderten Werte erreichen.

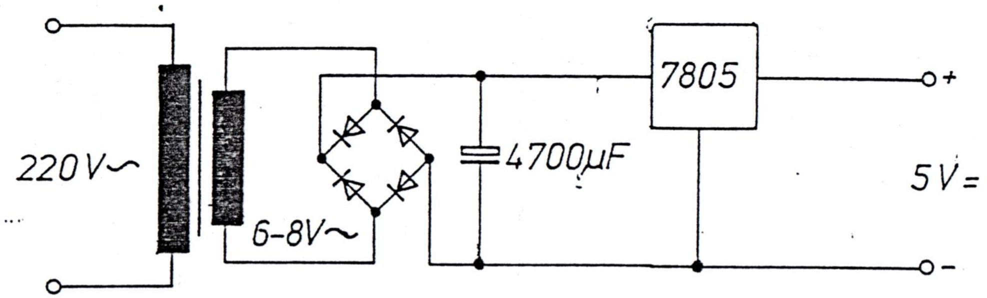
Das externe Powersupply *2 besteht aus einem 6-8 Volt Transformator (0.5-1 Ampere belastbar), einem Brückengleichrichter und einem 4700 µF Siebkondensator. Als Spannungsstabilisator dient ein IC 7805 Spannungskonstanter. Die Versorgungsspannung sollte an jedem Chip mit einem Keramikkondensator (0.1 µF) von Störungen frei gehalten werden.

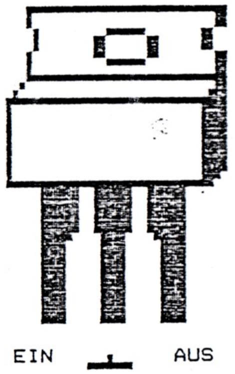
Nach dem Zusammenbau von Powersupply und Memory Board sollte als erstes eine genaue Sichtkontrolle der Schaltung erfolgen. Auf der Speicherplatine kann man außerdem mit einem Ohmmeter feststellen, ob die 5 Volt-Leitungen einen Kurzschluß haben. Erst nach diesen Kontrollen sollte man die Stromversorgung anschließen. Als letzte Überprüfung schaltet man die Versorungsspannung ein und überprüft alle Leitungen, die später zum Computer führen darauf, daß keine mehr als 5 Volt führt.

Nach dem endgültigen Anschluß der Erweiterung kann man das nun zusätzlich vorhandene RAM durch das unten stehende Programm auf seine Funktionsfähigkeit überprüfen.

*1 Natürlich ist auch der Anschluß über den geplanten ECB-Bus realisierbar
*2 Siehe Schaltbild Nr. 2

```
 00 'MEMORY-TESTROUTINE FUER DAS ZUSATZ-RAM 3000H-377FH
    Langsam aber einfach! Am besten mit ZBASIC oder ACCEL
    Compilieren!!!
 10 CLS
 20 FOR X = 12288 TO 14207
 30       FOR Y = 0 TO 255
 40                POKE X,Y
 50                IF PEEK (X) <> Y THEN GOSUB 100
 60       NEXT Y
 70 NEXT X
 80 END
 90 'FEHLERBEHANDLUNG
100 PRINT"SPEICHERFEHLER IN ZELLE ";X;" SOLLWERT ";Y;" ISTWERT ";
110 PRINT PEEK (X)
120 PRINT, "WEITER MIT <ENTER> !"
130 IF INKEY>CHR13) THEN GOTO 130 ELSE RETURN
```



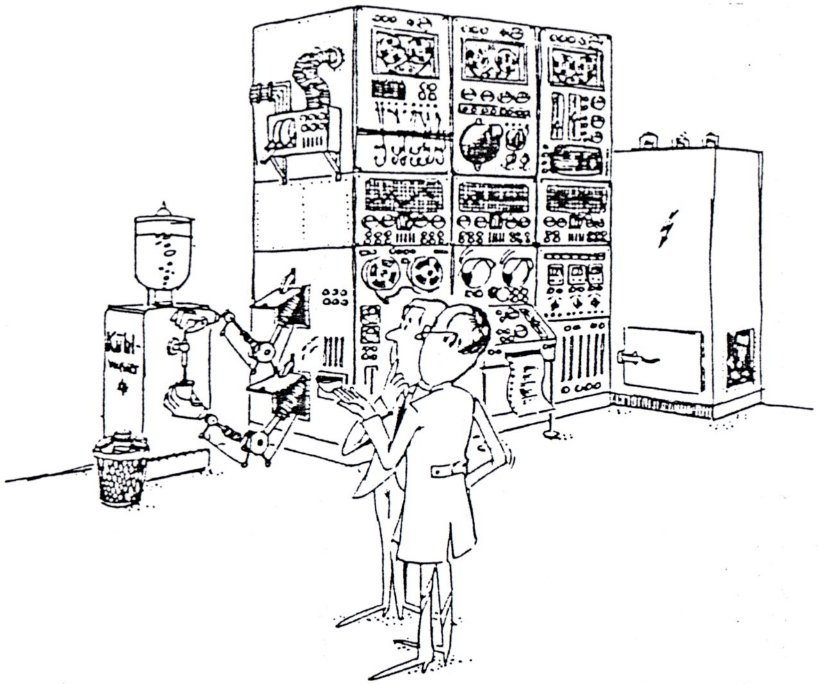


R E G L E R

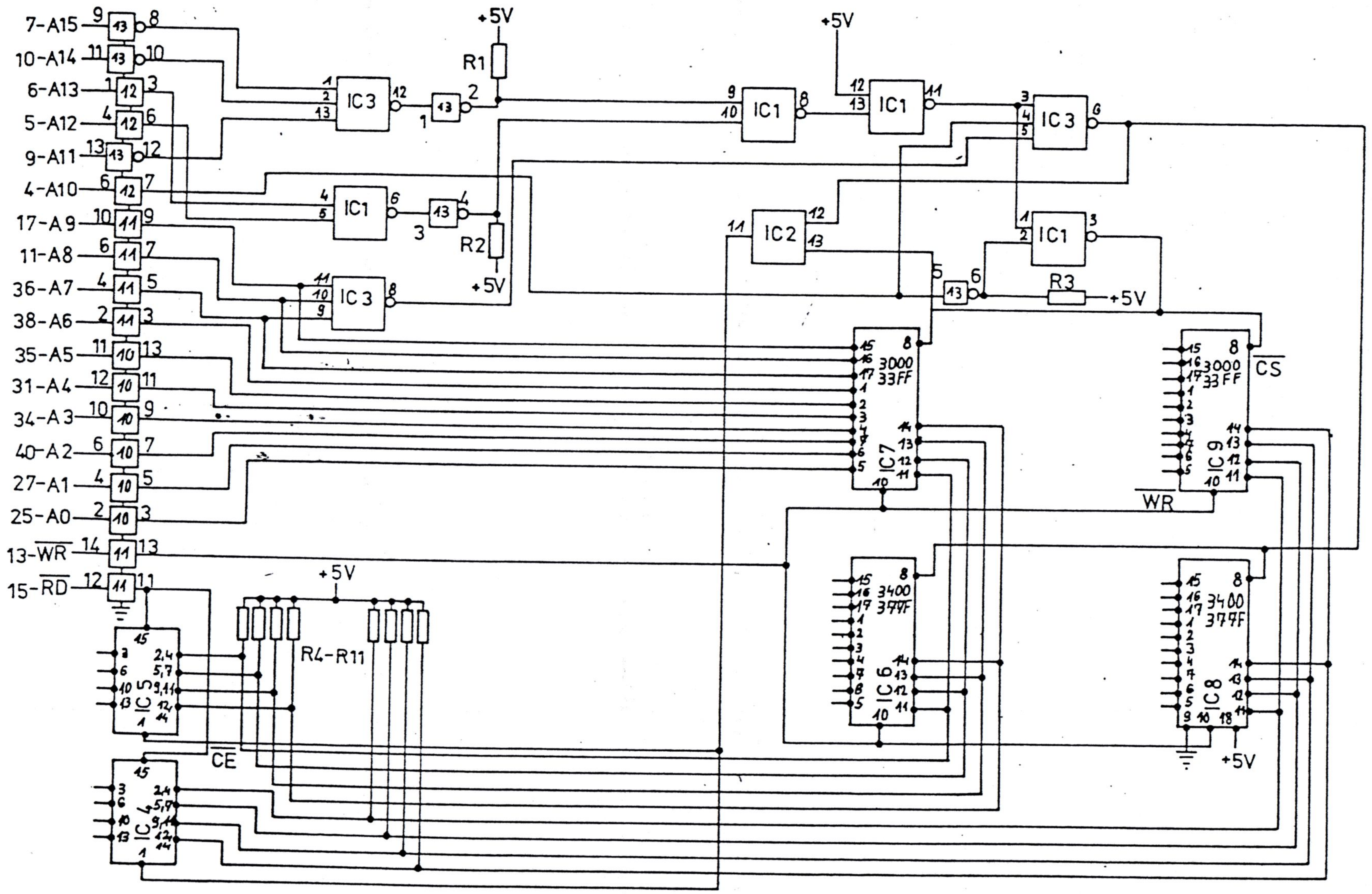
UA 78 05
UA 78 06
UA 78 08
UA 78 10
UA 78 12
UA 78 15
UA 78 18
UA 78 24

Die Originalzeichnung des Memory Board war wegen ihrer schlechten Qualität nicht zur Veröffentlichung geeignet. Ich bedanke mich vielmals bei Manfred Held, daß er eine reproduzierbare Zeichnung erstellte.
Die Zeichnung des Netzteils stammt von mir. Wie man feststellen kann, sind meine Fähigkeiten auf dem Gebiet des technischen Zeichnens nicht allzusehr ausgeprägt. Ich hoffe, man kann erkennen was gemeint ist.

Hartmut Obermann



»Donnerwetter! Das ist ja beinahe menschlich, nicht?«



Bauteile:

R1-R11 = 1KΩ
IC1 = 74LS00
IC2 = 74 LS08
IC3 = 74 LS10
IC4 = 8216
IC5 = 8216
IC6 = 2114
IC7 = 2114
IC8 = 2114
IC9 = 2114
IC10 = LS 367
IC11 = LS 367
IC12 = LS 367
IC13 = LS 368

zu IC 8216:

8216		
RD	CE	
0	0	Mem→Buss
1	0	Mem←Buss
0	1	off
1	1	off

SuperTape für TRS-80

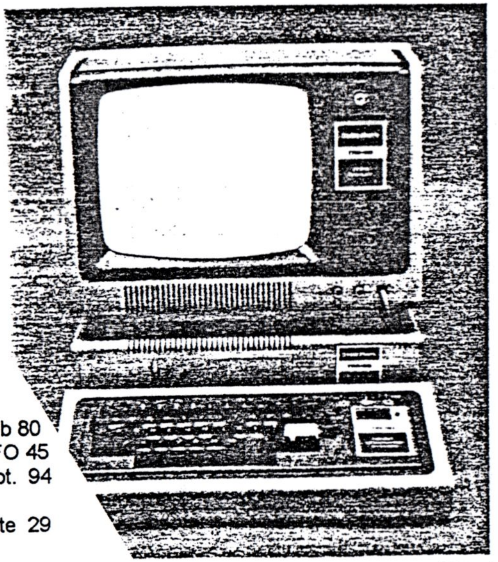
Andreas Burgwitz, Andreas Stiller

Endlich schließt sich auch der TRS-80 der Gemeinde der SuperTape-Rechner an. Zuvor galt es allerdings eine Hürde zu nehmen, die die Entwickler des mittlerweile recht betagten Rechners errichteten: Vom SuperTape-Eingangssignal läßt die Schaltung des Kassettenports nichts über, was mittels Software ausgewertet werden könnte. Die Lösung dieses Problems besteht aus einer kleinen Zusatzplatine, die dem TRS-80 SuperTape-Betrieb mit 7200 Baud ermöglicht.

Gerade die Anwender des TRS-80 werden ein schnelles und vor allem störsicheres Kassettenverfahren begrüßen; sind sie doch vom Standard-Verfahren des Rechners Schlimmes gewohnt. Und gerade dieses Verfahren bescherte die Probleme, die den Einsatz einer Zusatzschaltung nötig machen. Aber wie fast alles, hat auch dies seine guten Seiten: Mit dem so notwendig gewordenen zusätzlichen Interface kann der TRS-80 SuperTape-Kassetten mit 7200 Baud beschreiben und lesen. Zusätzlich bietet die Schaltung je ein frei verwendbares 'Ein-/Ausgabebit'.

Die SuperTape-Software entspricht im wesentlichen dem in Heft 1/85 vorgestellten Programm für das Colour Genie, und damit auch dem SuperTape für den ZX-Spectrum. Allerdings sind die SuperTape-Ein-/Ausgaberoutinen an den Tandy angepaßt. Da der TRS-80 auch mit 7200 Baud lesen und schreiben soll, mußte die Laderoutine völlig neu erstellt werden.

Die TRS-80 SuperTape-Lösung kann man auf Geräten mit und ohne Diskettenlaufwerk betreiben. Sinnvollerweise sollte der Tandy aber mit minimal 16 KByte RAM ausgebaut sein. Das abgedruckte Programm liegt im Speicher ab der Adresse B900h. Natürlich kann man das Programm auch an die obere Grenze des Speichers legen, wo es aber oftmals mit einem Monitor oder anderen Utilities kollidiert.



Vor dem Start von SuperTape sollte man diesen Speicherbereich gegen Überschreiben durch BASIC-Programme schützen. Dies geschieht durch die Beantwortung der Frage 'MEM SIZE ?' mit 47300. Allerdings sichert man SuperTape damit nicht gegen Maschinenprogramme, die im Speicher ab B900h liegen.

Input

Die Eingabe des Programms in den Rechner sollte nicht ohne die Hilfe eines Monitor-Programms geschehen, das das Abspeichern der eingegebenen Routinen als 'SYSTEM-File' erlaubt. Mehr Tipparbeit, aber wesentlich weniger Ärger bei Änderungen ergibt die Eingabe mit Hilfe eines Assemblers. Verfügt man über keins dieser Hilfsprogramme, sollte man SuperTape vom c't-Software-Service als SYSTEM-File beziehen.

Bits schieben

Wie das SuperTape-Verfahren funktioniert, ist in mehreren vorausgegangenen c't-Ausgaben ausführlich erläutert worden. Die SuperTape-Programme aller Z80-Rechner orientieren sich dabei an den Routinen für den Spectrum aus c't 6/84.

Für 7200 Baud wird allerdings dabei eine einigermaßen 'flotte' CPU vorausgesetzt, die der Tandy mit seinen 1,774 MHz Taktfrequenz offensichtlich nicht aufweisen kann. Vor allen Dingen bei der Laderoutine wird es zeitlich etwas zu 'eng'.

Die optimale Prüfzeit liegt für 7200 Baud bei etwa 2/3 einer ganzen Periode für die Übertragung einer Null: also 93 Mikrosekunden. Davon ist sogar noch die mittlere Flankenerkennungszeit abzuziehen, so daß dem Tandy also ungefähr 150 bis 160 Takte zur Verfügung stehen. Außerdem sollte man nach oben und nach unten noch etwas Spiel haben, um rein empirisch die beste Prüfzeitkonstante ermitteln zu können.

Kommunizieren mit SuperTape

Das SuperTape-Verfahren, in c't 4/84 vorgestellt und ausführlich beschrieben, macht Computer verschiedener Fabrikate datenträgerkompatibel. Dies bedeutet, daß die im SuperTape-Format beschriebenen Kassetten von jedem Computer gelesen werden können, für den es SuperTape-Routinen gibt. SuperTape funktioniert mit handelsüblichen Kassetten und Recordern, arbeitet aber um ein Vielfaches schneller als alle anderen Aufzeichnungsverfahren, die sonst bei Homecomputern verwendet werden. Für folgende Rechner hat c't bisher SuperTape-Routinen vorgestellt:

- [illegible] c't 4/84 u. 5/84
- VC-20 c't 5/84 u. 10/84
- C64 c't 5/84 u. 10/84
- Sinclair Spectrum c't 6/84 u. 9/84
- Apple II (und kompatible) c't 7/84
- 6809-Rechner (am Beispiel des Eurocom II) c't 8/84
- CP/M-Rechner c't 11/84
- Oric-1 und Atmos c't 12/84
- Colour Genie c't 1/85
- CBM 3000/4000/8000 c't 2/85
- Sharp PC1500 c't 3/85

Die meisten der Programme sind auf Datenträgern beim Heise-Software-Service (siehe Anzeigenteil) erhältlich. Unsere Leser sind eingeladen, SuperTape-Anpassungen für weitere Rechner (selbstverständlich gegen Honorar) in c't vorzustellen.

Die Spectrum-Routine benötigt jedoch bereits 175 Takte, selbst wenn gar kein Sprung mehr in eine Warteroutine erfolgt. Daß so ein Sprung nicht ausgeführt werden soll, müßte aber das Programm auch noch feststellen, da ja für 3600 Baud auf jeden Fall eine zusätzliche Warteschleife nötig ist. Für diese Abfrage wäre erst recht keine Zeit mehr vorhanden, so daß die gesamte Laderoutine einschließlich Synchronisation zweimal geschrieben werden müßte, einmal für 7200 Baud und einmal für 3600 Baud.

Das Spectrum-Programm birgt auch noch eine Gefahr für andere Rechner: Beim Einlesen des Ports erstreckt sich die Abfrage über den gesamten Datenbus. Eine eventuell 'floatende' Datenleitung verfälscht das Ergebnis völlig. Eine zusätzliche Ausmaskierung des einzulesenden Bits ist also vonnöten.

Über alle Register

Um 7200 Baud auch auf dem Tandy — sogar mit einiger Reserve — lauffähig zu bekommen, muß man fast alle Register der Z80-CPU 'ziehen'. Lediglich das Interrupt- und Refresh-, das IY- und das DE'-Register bleiben unberücksichtigt.

Das neue Interface legt die Signale vom Kassettenrecorder auf die Datenleitung D7. Das hat den Vorteil, daß hierbei bereits mit dem IN-Befehl das Bit abgefragt werden kann. Allerdings muß dafür das Register C die Portadresse enthalten, also IN A,(C). Die Abfrage JP M beziehungsweise JP P wertet dann unmittelbar das eingelesene Bit auf D7 aus. Damit läßt sich die Flankenabfrage innerhalb von nur 22 Takten durchführen, was der Lesesicherheit sehr zugute kommt.

Der eingelesene Wert wird als Carry-Flag in AF' zwischengespeichert. Falls der neu eingelesene Wert mit dem alten übereinstimmt, erkennt das Programm eine Null; bei Ungleichheit eine Eins. Bei einer Eins wird das Prüfsummenregister HL inkrementiert. Das entsprechend gesetzte Carry-Flag gelangt dann in bewährter Manier per Rotation ins E-Register, das so Bit für Bit bis zu dem gewünschten Byte aufgefüllt wird. Das Programm erkennt die 'Fülle' an dem Endflag, das am Anfang (Adresse BCC9h) gesetzt wird und dann nach acht Rotationen ins Carry gelangt.

Beim bitweisen Einlesen innerhalb der ersten Synchronisationsstufe findet — bedingt durch den Befehl SET E,0 — nur eine Rotation statt.

Der indirekte Sprung über das IX-Register ermöglicht eine einfache und schnelle Verzweigung (8 Takte) in eine Warteschleife, die der gewählten Baudrate entspricht. Bei 7200 Baud läßt sich so die Wartezeit im 'Viertakterhythmus' durch Einsprung in die NOP-Reihe von LDW36 bis LDSAM verändern. Der Benutzer von SuperTape sollte sowohl diese Einsprungadresse für 7200 Baud (in Adresse BC25h) als auch die Wartekonstante LCON36 für 3600 Baud etwas variieren, um die für seinen Recorder besten Werte zu finden.

Immerhin 22 Takte ließen sich gegenüber den Spectrum-Routinen dadurch sparen, daß für die Blocklade- und Vergleichsoperationen die gestrichenen Register herangezogen wurden. Die zeitkritische Situation am Ende der Synchronisation konnte entschärft werden, da die Routine gleich in die LDBYT-Routine 'überläuft' und somit bereits mit einem geladenen Byte zurückkehrt. Dieses eine Byte ist allerdings extra zu behandeln, bevor die eigentliche Lade- beziehungsweise Verify-Schleife beginnt.

Summa summarum kommt die Laderoutine einschließlich des JP (IX)-Befehls mit 139 Takten 'Eigenzeit' aus.

Gut ge'toggle't

Beim SAVE liegen die Dinge wesentlich einfacher. Da das Interface auch noch die Arbeit des 'toggelns' übernimmt, also bei jedem Aufruf den Portzustand ändert, wird das sonst nötige Toggle-Register frei. Dieses findet im Programm Verwendung als Flagregister, das das Baudratenflag schnell abrufbar speichert. Bei 3600 Baud (Bit 7 von D = 0) wird dann eine zusätzliche Wartezeit von 124 Takten eingeschoben.

Der Rahmen

Den Rahmen um die eigentlichen SuperTape-Routinen bildet ein menügesteuerter Programmteil, der die Anwendung von SuperTape komfortabel macht. Nach dem Laden von SuperTape meldet sich der TRS-80 mit:

```
SUPERTAPE FUER TRS-80. MODELL 1
(S)AVE (Q)UICK (L)OAD (V)ERIFY (E)XIT
COMMAND:
```

Durch die Eingabe eines in Klammern stehenden Buchstabens verzweigt das Programm zu der entsprechenden Routine und fordert die eventuell noch benötigten Angaben an. Gibt man zum Beispiel 'S' für Speichern mit 3600 Baud (oder Q — Speichern mit 7200 Baud) ein, fragt das Programm nach dem Namen des abzuspeichernden Programms:

```
NAME. . . . . .TYP
```

Allerdings sollte man bei 7200 Baud einen besseren Recorder verwenden als das Tandy-Gerät.

Nach der Eingabe des Namens gelangt man durch Betätigen der 'RETURN'-Taste zur TYP-Abfrage, zum Beispiel 'BAS' für BASIC-Programme oder 'HEX' für Speicherinhalte. Beim Typ 'HEX' fragt das Programm nach der Start- und Endadresse des Speicherbereichs. Diese Frage entfällt beim Typ 'BAS'; das Programm schreibt ein BASIC-Programm vom Anfang bis zum Ende (im Rechner durch Zeiger markiert) auf Band.

Bei der folgenden Frage 'TAPE READY ?' hat man noch die Möglichkeit, sofort wieder zum Menü zurückzukehren, indem man diese Frage mit 'N' beantwortet. Andernfalls schreibt das Programm die Daten auf Band.

Alle Funktionen, die man schon einmal ausgeführt hat, kann man im Menü durch die Eingabe von 'Shift Buchstabe' erneut aufrufen, wobei die zuvor eingegebenen Parameter übernommen und angezeigt werden.

Soll die Aufzeichnung sogleich überprüft werden, genügt die Eingabe von 'v' (Shift), da die Parameter ja dieselben wie beim Abspeichern sind. Nach der Frage 'TAPE READY ?' erscheint in der untersten Bildschirmzeile ein Sternchen, das zu 'flackern' beginnt, sobald Signale vom Band kommen. Während das Sternchen 'blinkt', kann man den Ladevorgang jederzeit durch die 'BREAK'-Taste abbrechen und zurück zum Menü gelangen. Als Abschluß der Verify-Routine listet das Programm die Daten des 'geladenen' Programms und gibt eine Lade-/Prüfmeldung neben dem Sternchen aus:

- #• Fehlerfrei
- B• Blockfehler
- >• Fehler bei Verify
- F• Prüfsummenfehler

Beim Load-Befehl 'L' fragt das Programm nach der Eingabe des Namens noch zusätzlich nach einer neuen Startadresse. Somit kann man ein Programm an beliebiger Stelle im Speicher ablegen. Bei der Namenseingabe erlaubt das Programm die Verwendung von 'Jokern'; das Zeichen '•' steht für einen beliebig langen Namen mit beliebigen Buchstaben. Die einzige Ausnahme hiervon bilden BASIC-Programme: Sie sollten immer mit der Typ-Angabe 'BAS' geladen werden, da nur dann die Zeiger für den späteren Start des BASIC-Programms richtig gesetzt werden. Die Zeiger 'stehen' zwar richtig, wenn man ein mit dem DISK-BASIC erstelltes Programm wieder in diesem BASIC starten beziehungsweise listen will; sie stimmen aber nicht, wenn ein ROM-BASIC-Programm unter DISK-BASIC laufen soll (oder umgekehrt).

Nach dem erfolgreichen Laden eines BASIC-Programms verzweigt das SuperTape-Programm in das BASIC: Hat man SuperTape aus dem ROM-BASIC gestartet, springt das Programm auch in dieses BASIC. Arbeitet man mit Disketten und hat SuperTape nicht aus dem DISK-BASIC aufgerufen, erscheint kurz die Fehlermeldung 'KEIN BASIC' und anschließend das Menü. Soll ein BASIC-Programm nach dem Laden ausgeführt werden, ist folgendermaßen vorzugehen:

Vom DOS aus mit dem Befehl 'LOAD SUTRS/CMD' SuperTape laden.

Anschließend muß man BASIC aufrufen.

Im BASIC den Befehl 'SYSTEM' eingeben und auf die Meldung '?•' mit '/47360' antworten — SuperTape wird gestartet.

Einen direkten Rücksprung ins BASIC oder DOS kann man durch die Eingabe von 'E' im SuperTape-Programm erreichen. Arbeitet man mit Disketten und hatte vor dem Aufruf von SuperTape BASIC geladen, führt 'E' direkt ins DISK-BASIC. Ohne geladenes BASIC gelangt man durch die Eingabe von 'E' in die Befehlsebene vom DOS, beim Arbeiten im ROM-BASIC erfolgt ein BASIC-Warmstart.

Die Unterscheidung zwischen diesen drei 'Betriebssystemen' erfolgt selbsttätig durch das Programm. Dazu wertet es ein Byte aus dem 'Keyboard DCB' (Zeiger auf die Tastatur-Treiberroutine) aus.

Beim TRS-80, Modell 1 mit dem 'Level 2 ROM-BASIC' steht in dieser Speicherzelle (4017h) der Wert 03 beim aktivierten ROM-BASIC, beim Betrieb des TRS-80 unter NEW-DOS 80 erwartet das Pro-

gramm das Byte 43h in der Speicherzelle. Ein Sprung in das DISK-BASIC erfolgt dann, wenn die zuvor genannten Werte nicht erkannt werden konnten.

Interface

Die Adreßdekodierung des Interfaces besteht aus IC1 und IC2. Die ausgewerteten Adressen schalten entweder ein Flipflop (IC3a) oder erlauben Ein-/Ausgaben über den Puffer IC4. Das Flipflop IC3a schaltet bei jedem Zugriff auf den Port 7Fh (127d) um, unabhängig von den ausgegebenen Daten. Über die drei Inverter (IC5a...IC5c) und das Filter (R2, R3, C1, C2) gelangt das Ausgangssignal des Flipflops an den Kassettenrecorder.

Die Datenleitungen D0, D1, D6 und D7 des TRS-80 liegen über den Puffer IC4 an IC3a und IC3b. Der logische Pegel der Datenleitung D0 gelangt über die Brücke J1 an den 'Clear'-Eingang von IC3a. Legt man diesen Eingang kurzfristig auf logisch 0, schaltet der Ausgang von IC3a auf logisch 1. Diese Funktion benötigt man aber nur dann, wenn mehr als zwei Rechner mit einer 'SuperTape-Ringleitung' verbunden werden sollen. Im Normalfall kann man die Brücke 1 offenlassen und an der Leiste X3 den Impuls von D0 abgreifen.

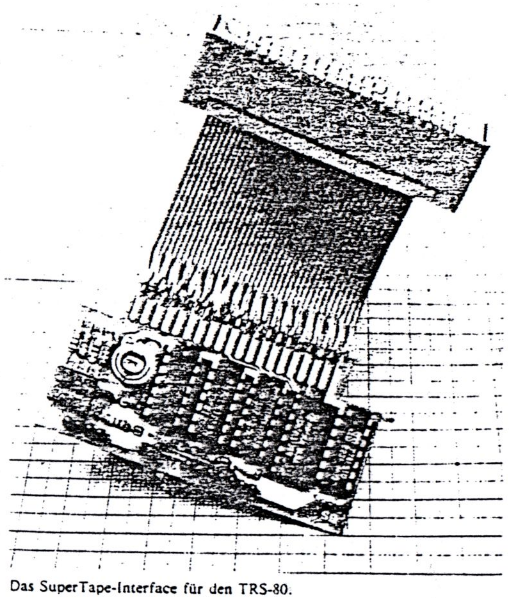
Mit dem Datenbit D1 kann man das Flipflop IC3b steuern. So setzt zum Beispiel der BASIC-Befehl 'OUT 126,2' den Ausgang von IC3b auf logisch 1. Schließt man an diesen Ausgang eine Treiberstufe und ein Relais an, kann man den Kassettenrecorder programmgesteuert ein- und ausschalten.

Das Signal vom Kassettenrecorder gelangt über die Triggerschaltung (IC5d, IC5e) an die Datenleitung D7. Der Pegel dieser Leitung entspricht nach einer Portabfrage dem Eingangssignal am Kassetten-Eingang. Die Leitung D6 liegt über den Puffer an der Anschlußleiste X3, Pin 2. Hier kann man ein Eingangssignal mit TTL-Pegel in den Rechner einlesen.

Aufbau

Da die Leiterplatte sehr eng bestückt wird, sollte man beim Aufbau besonders sorgfältig vorgehen. Der Anschluß des Interfaces an den Rechner geschieht über die Steckerleiste X4. Die Platine ist so ausgelegt, daß man eine 40polige Steckerleiste direkt anlöten kann. Soll das Interface an den 'Interface-Bus' (J3) des Tandy-Expansion-Interface angeschlossen werden, hat diese Lösung allerdings den Nachteil, daß die Platine 'in der Luft schwebt'. Beim direkten Anschluß an den TRS-80-Bus kann man der Leiterplatte durch zwei Distanzrollen festen Halt geben.

Die wohl bessere Lösung besteht darin, die Platine über ein Stück Flachbandkabel und einen 'Preßstecker' mit dem Rechner zu verbinden. Allerdings muß das Kabel so kurz wie möglich sein, da sonst der Tandy recht 'eigenwillig' wird.



Das SuperTape-Interface für den TRS-80.

Test

Nach dem Bestücken sollte man die Platine sehr sorgfältig auf Kurzschlüsse und Unterbrechungen untersuchen. Anschließend kann man die Karte auf den Bus des Rechners oder des Expansion-Boards stecken und den Computer einschalten: Das System sollte sich wie gewohnt verhalten. Andernfalls ist der Computer sofort abzuschalten und das Interface erneut zu überprüfen.

Tritt dieser Fall nicht ein, können die weiteren Tests des Interfaces mit BASIC-Befehlen und einem Voltmeter durchgeführt werden. Dazu sollte man die Brücke J1 schließen und ein Voltmeter an Pin 2, 4 oder 6 von IC5 anschließen. Nach jeder Ausführung des BASIC-Befehls 'OUT 127,0' muß der Pegel an IC5 wechseln; von circa 0 Volt auf etwa 5 Volt und umgekehrt. Für den nächsten Test sollte die Spannung an IC5 etwa 0 Volt betragen. Mit der Ausführung der Anweisung 'OUT 126,0' muß an IC5 der Pegel auf rund 5 Volt wechseln (Reset-Funktion).

Anschließend ist das Voltmeter an Pin 5 von der Steckerleiste X3 anzuschließen. Der Befehl 'OUT 126,2' bewirkt, daß der Ausgang von IC3b auf logisch 1 (etwa 5 Volt) springt. Mit der Anweisung 'OUT 126,0' ergibt sich eine Spannung von rund 0 Volt an dem Ausgang dieses ICs.

Für die Überprüfung des Kassetten-Eingangs gibt man folgendes Programm ein:

```
10 A = INP(126)
20 PRINT A;
30 GOTO 10
```

Nach dem Start des Programms wird ein Wert angezeigt, der sich ändern muß, wenn man P1 an den rechten oder linken Anschlag dreht. Die richtige Einstellung von P1 liegt an dem Punkt, wo die Anzeige zwischen den beiden Werten springt. □

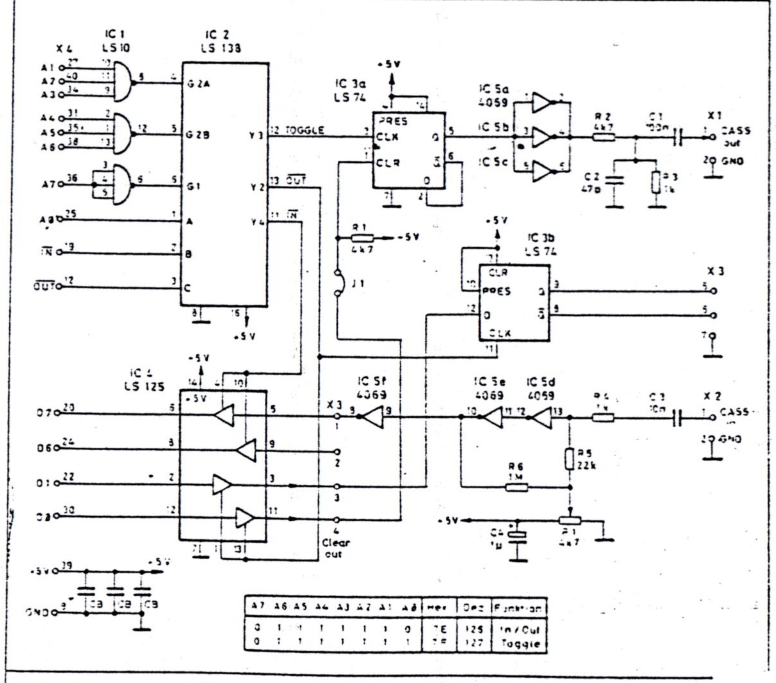


Bild 1. Schaltplan des TRS-80-Interfaces.

Stückliste

Widerstände		IC2	74LS138
R1,2	4k7	IC3	74LS74
R3,4	1k	IC4	74LS125
R5	22k	IC5	CD 4069
R6	1M	**Sonstiges**	
P1	4k7 Trimmer, liegend	X1, X2, J1	Pfostenleiste, 2polig
Kondensatoren		X3	Pfostenleiste, 7polig
C1	100n	X4	40poliger 'Card-Edge'-Stecker, 2reihig, Raster 2,54 mm
C2	47p		
C3	10n		
C4	1μ, Tantal		
CB	3 Stützkondensatoren, 100n, ker.		
Halbleiter			
IC1	74LS10		

Brückenstecker für Pfostenleiste; IC-Fassungen: 4 x 14polig, 1 x 16polig; Platine 'TRS-80 SuperTape'.

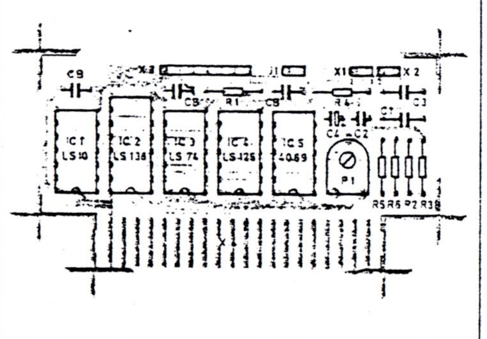


Bild 2. Der Bestückungsplan. Da die ICs sehr dicht beieinander liegen, müssen 'anreihbare' Fassungen verwendet werden.

```
8900                    ORG     08900H
0033            DISPL   EQU     00033H
0049            POTAS   EQU     00049H
0066            READY   EQU     00066H
01C9            CLS     EQU     001C9H
401A            CURSYM  EQU     0401AH
40A4            HEADP   EQU     040A4H
40F9            TAILP   EQU     040F9H
40FB            VARANF  EQU     040FBH
40FD            VAREND  EQU     040FDH
3FF0            BILDP1  EQU     03FF0H
3FF1            BILDP2  EQU     03FF1H
3840            ZTAST   EQU     03840H
0008            LCON36  EQU     08H
007E            PORTIN  EQU     07EH
007F            PORTOU  EQU     07FH
06CC            ROMBAS  EQU     06CCH
06CC            DOSBAS  EQU     06CCH
402D            DOS     EQU     402DH
FFFF            STACK   EQU     0FFFFH
4017            ECHEC   EQU     04017H
                ;
                ;
                ;*****************************************************
                ;SUPERTAPE TANDY TRS-80, MODELL I
                ;MENUEAUSWAHL
                ;
                ;*****************************************************
                ;
                ;
                ;
8900 31FFFF     START:  LD      SP,STACK
8903 CD2989     START1: CALL    CLSRE           ;CLEAR BILDSCHIRM
8906 215D80     START2: LD      HL,TEXT1        ;TEXT
8909 CDEF89             CALL    PRTEXT          ;ZUM
890C CD4900     TAST:   CALL    POTAS           ;BILDSCHIRM
890F 4F                 LD      C,A
8910 E65F               AND     5FH
8912 FE53               CP      053H
8914 2868               JR      Z,SAVE          ;'S'
8916 FE51               CP      51H             ;'Q'
8918 2864               JR      Z,SAVE
891A FE4C               CP      04CH
891C CA898A             JP      Z,LOAD          ;'L'
891F FE56               CP      056H
8921 CAF789             JP      Z,VERI          ;'V'
8924 FE45               CP      045H            ;'E'
8926 CA1980             JP      Z,BLARM         ;ZUM BASIC ODER DOS
8929 18E1               JR      TAST            ;SONST ZUR
                ;                               ;TATATURABFRAGE
                ;*****************************************************
                ;BILDSCHIRM LOESCHEN UND CURSOR SETZEN
                ;*****************************************************
892B CDC901     CLSRE:  CALL    CLS             ;CLEAR   SCREEN
892E 3E67               LD      A,067H          ;NEUES CURSOR-
8930 321A40             LD      (CURSYM),A      ;SYMBOL
8933 3E0E               LD      A,00EH          ;CURSOR
8935 CD3300             CALL    DISPL           ;EIN
8938 C9                 RET
                ;
                ;*****************************************************
                ;TASTATURABFRAGE FUER LD NAME/TYP
                ;*****************************************************
8939 7E         RONIT:  LD      A,(HL)
893A CB69               BIT     5,C             ;GROSS ODER KLEIN?
893C 200C               JR      NZ,RDCONT
893E CD4900             CALL    POTAS           ;TASTATURABFRAGE
8941 FE0D               CP      00DH            ;RETURN ?
8943 280C               JR      Z,RETUR         ;WENN JA SPRUNG
8945 FE08               CP      008H            ;CURSOR -
8947 2812               JR      Z,CURS          ;WENN JA SPRUNG
8949 77                 LD      (HL),A          ;ZEICHEN ABLEGEN
894A CD3300     RDCONT: CALL    DISPL           ;UND ZUM BILD
894D 23                 INC     HL
894E 10E9               DJNZ    RONIT           ;WEITER BIS B=0
8950 C9                 RET
                ;*****************************************************
                ; RETURN WURDE GEDRUECKT
                ;*****************************************************
8951 3E20       RETUR:  LD      A,020H          ;FUELLE REST MIT
8953 77                 LD      (HL),A          ;SPACE AUF
8954 CD3300             CALL    DISPL
8957 23                 INC     HL
8958 10F7               DJNZ    RETUR
895A C9                 RET
                ;*****************************************************
                ; CURSOR ZURUECKSETZEN
                ;*****************************************************
895B 2B         CURS:   DEC     HL              ;
895C 2B                 DEC     HL              ;CURSOR
895D 04                 INC     B               ;EIN ZEICHEN
895E 04                 INC     B               ;ZURUECKSETZEN
895F 3E19               LD      A,019H
8961 18E7               JR      RDCONT
                ;*****************************************************
                ;LADE HEXADRESSE NACH HL
                ;*****************************************************
8963 CD988A     HEXADR: CALL    ASIBYT          ;LAEDT 4STELLIGE
8966 57                 LD      H,A             ;HEXADRESSE UEBER
8967 CD988A             CALL    ASIBYT          ;ASCII/HEX-WANDLER
896A 6F                 LD      L,A             ;VON TASTATUR
896B C9                 RET                     ;NACH HL
                ;*****************************************************
                ;LADE NAME UND TYP
                ;*****************************************************
896C 212B8D     NAITY:  LD      HL,SUNAME       ;NAME UND TYP
896F 060C               LD      B,00CH          ;WERDEN IN DEN
8971 CD3989             CALL    RONIT           ;SU-PUFFER
```

```
B974 3E2E               LD      A,02EH          ;GELADEN
B976 77                 LD      (HL),A
B977 23                 INC     HL
B978 0603               LD      B,003H
B97A CD39B9             CALL    RDNIT
B97D C9                 RET
                ;*********************************************
                ;SAVE
                ;*********************************************
B97E F5         SAVE:   PUSH    AF              ;RETTE BEFEHL
B97F 2182BD             LD      HL,TEXT2        ;TEXT
B982 CDEFB9             CALL    PRTEXT          ;ZUM BILD
B985 CD6CB9             CALL    NAITY           ;LADE BLOCKPARAMETER
B988 3A38BD             LD      A,(SVTYP)
B98B FE42               CP      042H            ;WENN TYP=B(BASIC)
B98D 2832               JR      Z,TIBAS         ;HOLE BASIC-POINTER
B98F CB69               BIT     5,C             ;ALTE WERTE?
B991 203E               JR      NZ,SAVEA
B993 21C9BD             LD      HL,TEXT3        ;
B996 CDEFB9             CALL    PRTEXT          ;LADE STARTADRESSE
B999 CD63B9             CALL    HEXADR          ;IN SV-PUFFER
B99C 223CBD             LD      (SVSTA),HL      ;
B99F E5                 PUSH    HL              ;
B9A0 3E20               LD      A,020H          ;EIN SPACE ZUM
B9A2 CD3300             CALL    DISPL           ;BILD
B9A5 CD63B9             CALL    HEXADR          ;LADE ENDADRESSE
B9A8 C1                 POP     BC              ;UND
B9A9 ED42       DTLEN:  SBC     HL,BC           ;ERRECHNE LAENGE
B9AB 223EBD             LD      (SVLEN),HL      ;FUER PUFFER
B9AE 3E00       STFLA:  LD      A,000H          ;
B9B0 213BBD             LD      HL,SVFLAG       ;BAUDRATE SETZEN
B9B3 77                 LD      (HL),A          ;FLAG 00 SETZEN
B9B4 F1                 POP     AF              ;HOLE BEFEHL
B9B5 FE51               CP      51H
B9B7 2002               JR      NZ,NOQ          ;NEIN: BAUDRATE 0
B9B9 3680               LD      (HL),80H        ;SONST 80H
B9BB CDDBB9     NOQ     CALL    TAPERY          ;TAPE READY?
B9BE C345BB             JP      SVSUP           ;DANN SAVEN
                ;*********************************************
                ; TYP BASIC
                ;*********************************************
B9C1 2AA440     TIBAS:  LD      HL,(HEADP)      ;FALLS TYP =B
B9C4 223CBD             LD      (SVSTA),HL      ;WERDEN ADRESSEN
B9C7 E5                 PUSH    HL              ;FUER START/LAENGE
B9C8 ED4BF940           LD      BC,(TAILP)      ;VOM HEAD/TAILPOINTER
B9CC C5                 PUSH    BC              ;ABGELEITET UND
B9CD E1                 POP     HL              ;AUTOMATISCH IN DEN
B9CE C1                 POP     BC              ;SV-PUFFER GESETZT
B9CF 18D8               JR      DTLEN
                ;
B9D1 213DBD     SAVEA:  LD      HL,SVSTA+1
B9D4 0602               LD      B,02H
B9D6 CD04BB             CALL    LILP2
B9D9 18D3               JR      STFLA
                ;*********************************************
                ;TAPE READY ?
                ;*********************************************
B9DB 2106BD     TAPERY: LD      HL,TEXT4        ;HOLE TEXTADRESSE
B9DE CDEFB9             CALL    PRTEXT          ;TEXT ZUM BILD
B9E1 CD4900             CALL    ROTAS           ;TASTATURABFRAGE
B9E4 FE4E               CP      04EH            ;'N' NEIN: BREAK
B9E6 CA00B9             JP      Z,START         ;DANN ZUM MENUE
B9E9 3E0F               LD      A,00FH          ;SONST RET
B9EB CD3300             CALL    DISPL
B9EE C9                 RET
                ;*********************************************
                ;MENUEGENERATOR (HL) ZUM BILD
                ;*********************************************
B9EF 7E         PRTEXT: LD      A,(HL)          ;PRINT WAS IN (HL)
B9F0 FE23               CP      023H            ;STEHT ZUM BILD
B9F2 C8                 RET     Z               ;BIS ENDMARKE
B9F3 CD3300             CALL    DISPL           ;ERREICHT
B9F6 23                 INC     HL
B9F7 18F6               JR      PRTEXT
                ;*********************************************
                ;VERIFY
                ;*********************************************
B9F9 21E6BD     VERI:   LD      HL,TEXT5        ;TEXT
B9FC CDEFB9             CALL    PRTEXT          ;ZUM BILD
B9FF CD6CB9             CALL    NAITY           ;HOLE BLOCKPARAMETER
BA02 3E02       VECONT: LD      A,002H          ;SETZE 02 IN
BA04 322ABD             LD      (ZFLAG),A       ;ZFLAG
BA07 1833               JR      TPRDY           ;TAPE READY
                ;
                ;*********************************************
                ;LADEAUFRUF LOAD
                ;*********************************************
BA09 21FEBD     LOAD:   LD      HL,TEXT6        ;TEST 'LOAD' USW.
BA0C CDEFB9             CALL    PRTEXT          ;ZUM BILD
BA0F CD6CB9             CALL    NAITY           ;NAME + TYP EINGEBEN
BA12 3A38BD             LD      A,(SVTYP)
BA15 FE42               CP      42H             ;'B'?
BA17 200D               JR      NZ,LQHEX
BA19 2AA440             LD      HL,(40A4H)
BA1C 223CBD             LD      (SVSTA),HL
BA1F 212ABD             LD      HL,ZFLAG
BA22 CBD6               SET     2,(HL)
BA24 1816               JR      TPRDY
                ;
BA26 2115BE     LQHEX:  LD      HL,TEXT7        ;TEXT
BA29 CDEFB9             CALL    PRTEXT          ;'NEUE STARTADRESSE'
BA2C CB69               BIT     5,C             ;ALTE WERTE?
BA2E 2832               JR      NZ,LOADA
BA30 CD4900             CALL    ROTAS           ;TASTATURABFRAGE
BA33 FE4A               CP      04AH            ;JA?
BA35 2818               JR      Z,NADR          ;DANN ZUR EINGABE
BA37 3E00       LDRDY:  LD      A,000H          ;ZFLAG AUF NULL
BA39 322ABD             LD      (ZFLAG),A       ;

BA3C CDDBB9     TPRDY:  CALL    TAPERY          ; TAPE READY?
BA3F 3E2A               LD      A,02AH          ;STERNCHEN
BA41 32F13F             LD      (BILDP2),A      ;SETZEN
BA44 CDDBBB             CALL    LDSUP           ;LADEN
BA47 3229BD             LD      (ERROR),A       ;FEHLERMELDUNG RETTEN
BA4A CDE2BA             CALL    LIIPU           ;LD-PUFFER AUSGEBEN
BA4D 182B               JR      LDCHK           ;ZUM LADECHECK
                ;-----------NEUE STARTADRESSE EINGEBEN
BA4F 2126BE     NADR:   LD      HL,TEXT8        ;TEXT 'ADRESSE'
BA52 CDEFB9             CALL    PRTEXT          ;ZUM BILD
BA55 CD63B9             CALL    HEXADR          ;LADE ADRESSE
BA58 223CBD             LD      (SVSTA),HL      ;IN SV-PUFFER
BA5B 3E04               LD      A,004H          ;SETZE FLAG 04
BA5D 322ABD             LD      (ZFLAG),A       ;IN ZFLAG
BA60 18DA               JR      TPRDY           ;UND WEITER
                ;-----------LADEAUFRUF "L" OHNE BLOCKEINGABE
BA62 3A2ABD     LOADA:  LD      A,(ZFLAG)
BA65 CB57               BIT     2,A
BA67 280A               JR      Z,LADR
BA69 213DBD             LD      HL,SVSTA+1
BA6C 0601               LD      B,001
BA6E CD04BB             CALL    LILP2
BA71 18C9               JR      TPRDY
BA73 3E2A       LADR:   LD      A,02AH
BA75 CD3300             CALL    DISPL
BA78 18BD               JR      LDRDY
                ;*********************************************
                ;LDCHECK:AUSWERTUNG LOAD/VERIFY
                ;*********************************************
BA7A 3A29BD     LDCHK:  LD      A,(ERROR)       ;FEHLERMELDUNG
BA7D 32F03F             LD      (BILDP1),A      ;ABLEGEN
BA80 FE42               CP      042H            ;'B' ? =BLOCKFEHLER
BA82 28B8               JR      Z,TPRDY         ;ZURUECK ZUR AUFNAHME
BA84 FE23               CP      023H            ;'#' LADE ZFLAG
BA86 2803               JR      Z,LDOK
BA88 C306B9     RUECK:  JP      START2          ;ZUM MENUE
                ;
BA8B 3A2ABD     LDOK:   LD      A,(ZFLAG)       ;LADE ZFLAG
BA8E FE02               CP      002H            ;VERIFY?
BA90 28F6               JR      Z,RUECK         ;DANN ZUM MENUE
BA92 3A51BD             LD      A,(LDTYP)       ;TYP LADEN
BA95 FE42               CP      042H            ;B(BASIC) ?
BA97 20EF               JR      NZ,RUECK        ;JA ? DANN WEITER
BA99 187B               JR      STBAS           ;UND BASIC-POINTER
                ;                               ;SETZEN
                ;*********************************************
                ;ASCII/HEX WANDLER
                ;*********************************************
BA9B CDA8BA     ASIBYT: CALL    ASINIB          ;
BA9E 07                 RLCA                    ;ROUTINE WANDELT
BA9F 07                 RLCA                    ;ZWEI ASCII-ZEICHEN
BAA0 07                 RLCA                    ;IN EINE ZWEISTELLIGE
BAA1 07                 RLCA                    ;HEX-ZAHL
BAA2 4F                 LD      C,A             ;
BAA3 CDA8BA             CALL    ASINIB          ;
BAA6 B1                 OR      C               ;ERGEBNIS IN A
BAA7 C9                 RET
                ;
BAA8 CD4900     ASINIB: CALL    ROTAS           ;EIN ZEICHEN WIRD
BAAB F5                 PUSH    AF              ;VON DER TASTATUR
BAAC CD3300             CALL    DISPL           ;GELESEN UND ALS
BAAF F1                 POP     AF              ;EIN HEX-NIBBLE
BAB0 D630               SUB     030H            ;INTERPRETIERT
BAB2 D8                 RET     C
BAB3 C6E9               ADD     A,0E9H
BAB5 D8                 RET     C
BAB6 C606               ADD     A,006H
BAB8 F2BEBA             JP      P,ASICON
BABB C607               ADD     A,007H
BABD D8                 RET     C
BABE C60A       ASICON: ADD     A,00AH
BAC0 B7                 OR      A
BAC1 C9                 RET
                ;
BAC2 7E         BYTIAS: LD      A,(HL)          ;
BAC3 F5                 PUSH    AF              ;ROUTINE WANDELT
BAC4 D9                 EXX                     ;ZWEISTELLIGE
BAC5 0F                 RRCA                    ;HEXZAHL
BAC6 0F                 RRCA                    ;IN ZWEI ASCII-
BAC7 0F                 RRCA                    ;ZEICHEN UND
BAC8 0F                 RRCA                    ;GIBT DIESE
BAC9 E60F               AND     00FH            ;ZUM BILD
BACB CDD6BA             CALL    NIBIAS          ;AUS
BACE F1                 POP     AF
BACF E60F               AND     00FH
BAD1 CDD6BA             CALL    NIBIAS
BAD4 D9                 EXX
BAD5 C9                 RET
                ;
BAD6 C690       NIBIAS: ADD     A,090H          ;BERECHNET ASCII
BAD8 27                 DAA                     ;AUS NIBBLE UND
BAD9 CE40               ADC     A,040H          ;BRINGT DAS ZEICHEN
BADB 27                 DAA                     ;AUF DEN BILDSCHIRM
BADC F5                 PUSH    AF
BADD CD3300             CALL    DISPL
BAE0 F1                 POP     AF
BAE1 C9                 RET
                ;*********************************************
                ;LIST PUFFER
                ;*********************************************
BAE2 3E0D       LIIPU:  LD      A,00DH          ;CR
BAE4 CD3300             CALL    DISPL           ;ZUM BILD
BAE7 2144BD             LD      HL,LDNAME       ;LADE ADR AUS LD-PUFFER
BAEA 0610               LD      B,010H          ;LAENGE 10H FUER
BAEC 7E         LILP1:  LD      A,(HL)          ;NAME UND TYP
BAED CD3300             CALL    DISPL           ;ZUM BILD
BAF0 23                 INC     HL              ;WEITER
BAF1 10F9               DJNZ    LILP1           ;BIS B=0

BAF3 2A55BD             LD      HL,(LDSTA)      ;LADE STARTADR
BAF6 ED4B57BD           LD      BC,(LDLEN)      ;UND LAENGE
BAFA ED4A               ADC     HL,BC           ;ADDIEREN
BAFC 2259BD             LD      (LDFREI),HL     ;END-ADR ABLEGEN
BAFF 2156BD             LD      HL,LDSTH        ;ZEIGE AUF ADRESSEN
BB02 0603               LD      B,003H          ;
BB04 3E20       LILP2:  LD      A,020H          ;AUSGABESCHLEIFE
BB06 CD3300             CALL    DISPL           ;FUER STARTADRESSE
BB09 CDC2BA             CALL    BYTIAS          ;LAENGE
BB0C 2B                 DEC     HL              ;UND ENDADRESSE
BB0D CDC2BA             CALL    BYTIAS
BB10 23                 INC     HL
BB11 23                 INC     HL
BB12 23                 INC     HL
BB13 10EF               DJNZ    LILP2
BB15 C9                 RET
                ;*********************************************
                ; SET BASIC-POINTER
                ;*********************************************
BB16 3A1740     STBAS:  LD      A,(ECHEC)       ;TEST OB BASIC
BB19 FE43               CP      43H             ;GELADEN IST
BB1B 2817               JR      Z,NOBAS         ;Z=NEIN
BB1D 2131BE             LD      HL,TEXT9        ;TEXT 'START BASIC'
BB20 CDEFB9             CALL    PRTEXT          ;AUSGEBEN
BB23 0129BB             LD      BC,WARM
BB26 C3AE19             JP      19AEH
BB29 2A55BD     WARM:   LD      HL,(LDSTA)
BB2C ED5B57BD           LD      DE,(LDLEN)
BB30 19                 ADD     HL,DE
BB31 C3772C             JP      2C77H
                ;
                ;KEIN BASIC GELADEN
BB34 213FBE     NOBAS:  LD      HL,TEXT10       ;TEXT 'KEIN BASIC'
BB37 CDEFB9             CALL    PRTEXT          ;AUSGEBEN
BB3A 010000             LD      BC,0000
BB3D 0B         NOBALO: DEC     BC
BB3E 78                 LD      A,B
BB3F B1                 OR      C
BB40 20FB               JR      NZ,NOBALO
BB42 C300B9             JP      START
                ;*********************************************
                ;SVSUP
                ;*********************************************
BB45 F3         SVSUP:  DI                      ;INTERRUPTS ABSCHALTEN
BB46 212BBD             LD      HL,SVNAME       ;SENDET PARAMETERBLOCK
BB49 011900             LD      BC,00019H       ;
BB4C 3E2A               LD      A,02AH          ;
BB4E 1600               LD      D,0             ;FUER 3600 BAUD
BB50 CD66BB             CALL    SVBLO           ;SENDET DATENBLOCK
BB53 2A3CBD             LD      HL,(SVSTA)      ;
BB56 ED4B3EBD           LD      BC,(SVLEN)      ;
BB5A 3A3BBD             LD      A,(SVFLAG)      ;HOLE BAUDRATE
BB5D 57                 LD      D,A             ;NACH   D
BB5E 3EC5               LD      A,0C5H
BB60 CD66BB             CALL    SVBLO
BB63 C306B9             JP      START2
                ;*********************************************
                ;SVBLO
                ;*********************************************
BB66 F5         SVBLO:  PUSH    AF              ;RETTE ZUM STACK
BB67 0B                 DEC     BC              ;DECR. BLOCKLNGE
BB68 C5                 PUSH    BC              ;UND RETTE ZUM STACK
BB69 0640               LD      B,040H          ;SENDE 64 SYNC-ZEICHEN
BB6B 1E16       SVLO1:  LD      E,016H          ;LADE  E MIT SYNC-BYTE
BB6D A7                 DEFB    0A7H            ;WARTE 13 TAKTE
BB6E A7                 DEFB    0A7H
BB6F D8                 DEFB    0D8H
BB70 CDA2BB             CALL    SVBYT
BB73 10F6               DJNZ    SVLO1
BB75 C1                 POP     BC              ;HOLE LAENGE VOM STACK
BB76 F1                 POP     AF              ;HOLE SYNC-ABSCHLUSS
BB77 5F                 LD      E,A             ;SENDE SYNC-ABSCHLUSS
BB78 CDA2BB             CALL    SVBYT
BB7B DD210000           LD      IX,00000H       ;INIT PRUEFSUMME
BB7F A6                 DEFB    0A6H            ;WARTE 12 TAKTE
BB80 D8                 DEFB    0D8H
BB81 5E         SVLO2:  LD      E,(HL)          ;SENDE BYTE
BB82 CDA2BB             CALL    SVBYT           ;
BB85 EDA1               CPI                     ;INC HL,DEC BC
BB87 EA81BB             JP      PE,SVLO2        ;SCHLEIFE BIS BC=0
BB8A 5E                 LD      E,(HL)          ;SENDE LETZTES BYTE
BB8B CDA2BB             CALL    SVBYT           ;
BB8E C0                 DEFB    0C0H            ;WARTE 5 TAKTE
BB8F DDE5               PUSH    IX              ;SENDE PRUEFSUMME
BB91 C1                 POP     BC              ;
BB92 59                 LD      E,C
BB93 CDA2BB             CALL    SVBYT
BB96 E5                 DEFB    0E5H            ;WARTE 29 TAKTE
BB97 E1                 DEFB    0E1H
BB98 A7                 DEFB    0A7H
BB99 A7                 DEFB    0A7H
BB9A 58                 LD      E,B
BB9B CDA2BB             CALL    SVBYT
BB9E E3                 DEFB    0E3H            ;WARTE 50 TAKTE UND
BB9F E3                 DEFB    0E3H            ;SENDE SCHLUSSBYTE
BBA0 A6                 DEFB    0A6H
BBA1 D8                 DEFB    0D8H
                ;*********************************************
                ;SVBYTE: SAVE BYTE
                ;*********************************************
BBA2 37         SVBYT:  SCF                     ;SETZE ENDFLAG
BBA3 CB1B       SVBIT:  RR      E               ;AUSGABEBIT IN CARRY
BBA5 3827               JR      C,SVEINS        ;BIT =1? SPRUNG
BBA7 D37F               OUT     (07FH),A        ;UND AUSGEBEN
BBA9 A6                 DEFB    0A6H            ;WARTE  14 TAKTE
BBAA A6                 DEFB    0A6H            ;
BBAB CB7A       SVCONT: BIT     7,D

BBAD CCD3BB             CALL    Z,SVWAIT        ;WARTE FUER 3600 BAUD
BBB0 E3                 DEFB    0E3H            ;WARTE WEITERE 80 TAKTE
BBB1 E3                 DEFB    0E3H
BBB2 E3                 DEFB    0E3H
BBB3 E3                 DEFB    0E3H
BBB4 A7                 DEFB    0A7H
BBB5 D37F               OUT     (07FH),A        ;AUSGABE ZUM PORT
BBB7 CB7A               BIT     7,D
BBB9 CCD3BB             CALL    Z,SVWAIT        ;WARTE FUER 3600BD
BBBC D8                 DEFB    0D8H            ;WARTE 10 TAKTE
BBBD D8                 DEFB    0D8H
BBBE 1D                 DEC     E               ;8 BITS AUSGEGEBEN?
BBBF C8                 RET     Z               ;RETURN FALLS JA
BBC0 1C                 INC     E               ;SONST RESTAURIERE E
BBC1 3A4038             LD      A,(ZTAST)       ;WARTE 40 TAKTE
BBC4 FE04               CP      004H            ;UND TESTE BREAK
BBC6 CA00B9             JP      Z,START
BBC9 C8                 DEFB    0C8H
BBCA C8                 DEFB    0C8H
BBCB AF                 XOR     A               ;LOESCHE CARRY
BBCC 18D5               JR      SVBIT           ;WEITER
                ;
BBCE DD23       SVEINS: INC     IX              ;INKR. PRUEFSUMME
BBD0 C3ABBB             JP      SVCONT
                ;*********************************************
                ; SVWAIT WARTET :17 TAKTE FUER 3600 BAUD
                ;*********************************************
BBD3 3E06       SVWAIT: LD      A,006           ;WARTE 98 TAKTE
BBD5 3D         WTCON:  DEC     A
BBD6 20FD               JR      NZ,WTCON
BBD8 A7                 DEFB    0A7H            ;WEITERE 9 TAKTE
BBD9 D8                 DEFB    0D8H
BBDA C9                 RET
                ;*********************************************
                ;LDSUP:LADEROUTINE
                ;*********************************************
BBDB 215CBD     LDSUP:  LD      HL,LDEND        ;
BBDE 0619               LD      B,019H
BBE0 3E20               LD      A,020H
BBE2 77         CLCON:  LD      (HL),A          ;LOESCHE LD-PUFFER
BBE3 2B                 DEC     HL
BBE4 10FC               DJNZ    CLCON
BBE6 23                 INC     HL              ;HL=LDNAM
BBE7 011900             LD      BC,00019H       ;BLOCKLAENGE 19H
BBEA 3E2A               LD      A,02AH
BBEC DD2111BC           LD      IX,LDWAIT       ;BAUDRATE FUER 3600BD
BBF0 CD38BC             CALL    LDBLOC          ;LADE BLOCKPARAMETER
BBF3 C0                 RET     NZ              ;RET BEI FEHLER
BBF4 1144BD             LD      DE,LDNAME       ;VERGLEICHE NAMEN
BBF7 212BBD             LD      HL,SVNAME       ;
BBFA 0610               LD      B,010H
BBFC 7E         CPCON:  LD      A,(HL)
BBFD FE2A               CP      02AH            ;FALLS * SKIP REST
BBFF 280A               JR      Z,SKNAM
BC01 FE3F               CP      03FH            ;FALLS ? SKIP CHAR
BC03 2805               JR      Z,SKCHA
BC05 1A                 LD      A,(DE)
BC06 BE                 CP      (HL)
BC07 3E42               LD      A,042H
BC09 C0                 RET     NZ              ;FALSCH? DANN RET
BC0A 23         SKCHA:  INC     HL
BC0B 13         SKNAM:  INC     DE
BC0C 3E04               LD      A,004H          ;AB B=4 MIT
BC0E B8                 CP      B               ;SV-TYP VERGLEICHEN
BC0F 2003               JR      NZ,SKCON
BC11 2138BD             LD      HL,SVTYP
BC14 10E6       SKCON:  DJNZ    CPCON
BC16 2A55BD             LD      HL,(LDSTA)      ;PARAMETER FUER DATEN-
BC19 ED4B57BD           LD      BC,(LDLEN)      ;BLOCK SETZEN
BC1D 3A54BD             LD      A,(LDFLAG)
BC20 17                 RLA
BC21 3004               JR      NC,IXALT
BC23 DD21D0BC           LD      IX,LDW24        ;BAUDRATE 7200BD
BC27 3A2ABD     IXALT   LD      A,(ZFLAG)       ;
BC2A CB57               BIT     2,A             ;NEUE STARTADRESSE
BC2C 2803               JR      Z,SKADR
BC2E 2A3CBD             LD      HL,(SVSTA)
BC31 CB4F       SKADR:  BIT     1,A             ;LADEN ODER VERIFY
BC33 3EC5               LD      A,0C5H
BC35 C26DBC             JP      NZ,LDVER
                ;*********************************************
                ;LDBLOC: LAEDT BLOCK
                ;*********************************************
BC38 167E       LDBLOC  LD      D,PORTIN
BC3A D9                 EXX
BC3B CDBEBC             CALL    LDSYN           ;SYNCHRONISIERT
BC3E D9                 EXX
BC3F 77                 LD      (HL),A          ;BYTE NACH (HL)
BC40 EDA1               CPI                     ;INC HL UND DEC BC
BC42 E250BC             JP      PO,LTEST        ;TEST BEI ENDE
BC45 D9         LDNEXT: EXX
BC46 CDC9BC             CALL    LDBYT           ;NAECHSTES BYTE
BC49 D9                 EXX
BC4A 77                 LD      (HL),A          ;NACH (HL)
BC4B EDA1               CPI                     ;UND SCHLEIFE BIS
BC4D EA45BC             JP      PE,LDNEXT       ;ENDE
BC50 4A         LTEST:  LD      C,D             ;SETZE PORTADRESSE
BC51 CDC9BC             CALL    LDBYT           ;LADE LO-PRUEFBYTE
BC54 00                 NOP                     ;WARTE 12 TAKTE
BC55 00                 NOP
BC56 00                 NOP
BC57 47         TEST:   LD      B,A
BC58 E5                 DEFB    0E5H            ;WARTE 25 TAKTE
BC59 00                 DEFB    00H
BC5A E1                 DEFB    0E1H
BC5B CDC9BC             CALL    LDBYT           ;LADE HI-PRUEFBYTE
BC5E 48                 LD      C,B             ;GELADENE
```

```
BC5F 47                 LD      B,A             ;PRUEFSUMME NACH BC
BC60 D9                 EXX
BC61 E5                 PUSH    HL
BC62 D9                 EXX                     ;BESTIMMTE
BC63 E1                 POP     HL              ;PRUEFSUMME NACH HL
                ;**************************************************
                ;LDTES: PRUEFSUMMENTEST
                ;**************************************************
BC64 AF                 XOR     A
BC65 ED42               SBC     HL,BC           ;VERGLEICHE HL MIT BC
BC67 3E23               LD      A,023H          ;LADE AKKU MIT '#'
BC69 C8                 RET     Z               ;RET FALLS OK
BC6A 3E46               LD      A,046H          ;SONST 'F'
BC6C C9                 RET                     ;UND RETURN
                ;**************************************************
                ;LDVER : VERIFY
                ;**************************************************
BC6D 167E       LDVER:  LD      D,PORTIN        ;PORTADRESSE
BC6F D9                 EXX
BC70 CD8EBC             CALL    LDSYN           ;SYNCHRONISIERT
BC73 D9                 EXX
BC74 EDA1               CPI                     ;VERGLEICH
BC76 2013               JR      NZ,ERRVE        ;ERROR FALLS <>
BC78 E285BC             JP      PO,VTEST        ;SPRUNG FALLS ENDE
BC7B D9         VERNXT: EXX
BC7C CDC9BC             CALL    LDBYT           ;NAECHST. BYTE
BC7F D9                 EXX
BC80 EDA1               CPI                     ;VERGLEICH
BC82 EA7BBC             JP      PE,VERNXT       ;BEI PE WEITER
BC85 4A         VTEST:  LD      C,D             ;SONST LADE PRUEF-
BC86 CDC9BC             CALL    LDBYT           ;SUMME
BC89 18CC               JR      TEST
BC8B 3E3E       ERRVE:  LD      A,03EH          ;FEHLER BEI VERIFY
BC8D C9                 RET
                ;**************************************************
                ;LDSYNC: SYNCHRONISATION
                ;**************************************************
BC8E 1E00       LDSYN:  LD      E,000H          ;INIT LADEBYTE
BC90 0E7E               LD      C,PORTIN        ;PORTADRESSE
BC92 F3                 DI                      ;INTERRUPTS AUS
BC93 57                 LD      D,A             ;C=ABSCHLUSSMARKE
                ;WARTE AUF EIN SYCHA
BC94 00         SYNEU1: DEFB    00H             ;WARTE 4 TAKTE
BC95 A6         LDNXB:  DEFB    0A6H            ;WARTE  7 TAKTE
BC96 C399BC     SYNEU2: JP      SYWEI           ;WARTE  10 TAKTE
BC99 CBC3       SYWEI:  SET     0,E             ;LADE NUR EIN BIT
BC9B CDCBBC             CALL    LDBIT
BC9E FE16               CP      016H            ;SYCHA ERKANNT?
BCA0 5F                 LD      E,A             ;
BCA1 20F2               JR      NZ,LDNXB        ;NEIN ? DANN NEXT
                ;TEST AUF WEITERE SYCHA
BCA3 A7                 DEFB    0A7H            ;WARTE 9 TAKTE
BCA4 D8                 DEFB    0D8H            ;
BCA5 060A               LD      B,00AH          ;ZAEHLER AUF 10
BCA7 A6         SYCON:  DEFB    0A6H            ;WARTE 7 TAKTE
BCA8 CDC9BC             CALL    LDBYT           ;LADE NAECHSTES BYTE
BCAB FE16               CP      016H            ;VERGL. MIT SYCHA
BCAD 20E5               JR      NZ,SYNEU1       ;UNGLEICH ? SYNC NEU
BCAF A6                 DEFB    0A6H            ;WARTE 7TAKTE
BCB0 10F5               DJNZ    SYCON           ;WEITER BIS 10*SYCHA
                ;UEBERLESEN FOLGENDER SYCHA
BCB2 1804               JR      SYEIN
BCB4 A6         SYIGN:  DEFB    0A6H            ;WARTE 22 TAKTE
BCB5 D8                 DEFB    0D8H
BCB6 D8                 DEFB    0D8H
BCB7 D8                 DEFB    0D8H
BCB8 CDC9BC     SYEIN:  CALL    LDBYT           ;LADE NAECHSTES BYTE
BCBB FE16               CP      016H            ;VERGL.MIT SYCHA
BCBD 28F5               JR      Z,SYIGN         ;JA, DANN WEITER LADEN
                ;TEST AUF ABSCHLUSSMARKE
BCBF BA                 CP      D               ;ABSCHLUSS RICHTIG?
BCC0 20D4               JR      NZ,SYNEU2       ;NEIN SYNCHR. NEU
BCC2 210000             LD      HL,00000H       ;INIT PRUEFSUME
BCC5 A6                 DEFB    0A6H            ;WARTE 23 TAKTE
BCC6 A6                 DEFB    0A6H
BCC7 A7                 DEFB    0A7H
BCC8 D8                 DEFB    0D8H
                ;**************************************************
                ;LDBYT: LAEDT EIN BYTE/BIT
                ;**************************************************
BCC9 1E80       LDBYT:  LD      E,080H          ;SETZE FLAG FUER 8 BITS
BCCB DDE9       LDBIT:  JP      (IX)            ;WARTE
BCCD 00         LOW36:  NOP                     ;VARIABLE
BCCE 00         LOW32:  NOP                     ;EINSRUNGS-
BCCF 00         LOW28:  NOP                     ;MOEGLICHKEITEN
BCD0 00         LOW24:  NOP                     ;FUER
BCD1 00         LOW20:  NOP                     ;IX
BCD2 00         LOW16:  NOP
BCD3 00         LOW12:  NOP
BCD4 00         LOW08:  NOP
BCD5 00         LOW04:  NOP
BCD6 ED78       LDSAM:  IN      A,(C)           ;SAMPLE-WERT
BCD8 FAFEBC             JP      M,HIGH
BCDB ED78       LOW:    IN      A,(C)           ;WARTE AUF STEIGENDE
BCDD F2DBBC             JP      P,LOW           ;FLANKE
BCE0 AF                 XOR     A               ;LOESCHE CARRY
BCE1 08                 EX      AF,AF'          ;NACH AF' UND HOLE
BCE2 00                 NOP                     ;ALTEN CARRY
BCE3 3001               JR      NC,NULL1        ;C=0 DANN 0
BCE5 23                 INC     HL              ;SONST INC PRUEFSUMME
BCE6 7B         NULL1:  LD      A,E             ;ROTIERE CARRY IN E
BCE7 1F                 RRA
BCE8 D8                 RET     C               ;RET BEI ENDFLAG
BCE9 5F                 LD      E,A             ;SPEICHER BITS
BCEA 3AF13F             LD      A,(BILDP2)
BCED EE01               XOR     001H            ;STERNCHEN BLINKT
BCEF 32F13F             LD      (BILDP2),A      ;
BCF2 3A4038     BRCHC:  LD      A,(ITAST)       ;ZEICHEN VON TAST.
BCF5 FE04               CP      004H            ;BREAK?
BCF7 00                 NOP
BCF8 C2CBBC             JP      NZ,LDBIT        ;LADE NAECHSTES BIT
BCFB C30089             JP      START
BCFE ED78       HIGH:   IN      A,(C)           ;WARTE AUF FALLENDE
BD00 FAFEBC             JP      M,HIGH          ;FLANKE
BD03 37                 SCF                     ;SETZE CARRY
BD04 08                 EX      AF,AF'          ;NACH AF' UND HOLE ALTES
BD05 3F                 CCF                     ;CARRY UND INVERTIERE
BD06 3001               JR      NC,NULL2        ;C=0 ERGIBT BIT=0
BD08 23                 INC     HL              ;INC PRUEFSUMME
BD09 7B         NULL2:  LD      A,E             ;ROTIERE CARRY IN E
BD0A 1F                 RRA
BD0B D8                 RET     C               ;RETURN BEI ENDFLAG
BD0C 5F                 LD      E,A
BD0D E5                 DEFB    0E5H            ;WARTE 21 TAKTE
BD0E E1                 DEFB    0E1H
BD0F 18E1               JR      BRCHC
                ;**************************************************
                ;LDWAIT: WARTE FUER 3600 BAUD
                ;**************************************************
BD11 3E08       LDWAIT: LD      A,LCON36                ;WARTE 172 TAKTE
BD13 3D         LOWCON: DEC     A
BD14 20FD               JR      NZ,LOWCON
BD16 C3D6BC             JP      LDSAM
                ;**************************************************
                ;EXIT NACH DOS, DOS-BASIC ODER ROM-BASIC
                ;**************************************************
BD19 3A1740     BWARM:  LD      A,(ECHEC)
BD1C FE03               CP      03
BD1E CACC06             JP      Z,ROMBAS        ;INS ROM-BASIC SPRINGEN
BD21 FE43               CP      43H
BD23 CA2D40             JP      Z,DOS           ;INS DOS SPRINGEN
BD26 C3CC06             JP      DOSBAS          ;INS DOS-BASIC SPRINGEN
                ;**************************************************
                ;DER PUFFERBEREICH
                ;**************************************************
0001            ERROR:  DEFS    1
0001            ZFLAG:  DEFS    1
000C            SVNAME: DEFS    12
0001            SVPUN:  DEFS    1
0003            SVTYP:  DEFS    3
0001            SVFLAG: DEFS    1
0002            SVSTA:  DEFS    2
0002            SVLEN:  DEFS    2
0004            SVFREI: DEFS    4
000C            LDNAME: DEFS    12
0001            LDPUN:  DEFS    1
0003            LDTYP:  DEFS    3
0001            LDFLAG: DEFS    1
0002            LDSTA:  DEFS    2
BD56            LDSTH   EQU     LDSTA+1
0002            LDLEN:  DEFS    2
0004            LDFREI: DEFS    4
BD5C            LDEND   EQU     LDFREI+3
                ;
BD5D 0D         TEXT1:  DEFB    00DH
BD5E 53                 DEFM    'SUPERTAPE FUER TRS-80, MODELL 1'
BD7D 0D                 DEFB    00DH
BD7E 0D                 DEFB    00DH
BD7F 28                 DEFM    '(S)AVE (Q)UICK  (L)OAD  (V)ERIFY (E)XIT'
BDA6 0D                 DEFB    00DH
BDA7 0D                 DEFB    00DH
BDA8 43                 DEFM    'COMMAND:'
BDB0 0E                 DEFB    00EH
BDB1 23                 DEFB    '#'
BDB2 0D         TEXT2:  DEFB    00DH
BDB3 53                 DEFM    'SAVE'
BDB7 0D                 DEFB    00DH
BDB8 4E                 DEFM    'NAME........TYP'
BDC7 0D                 DEFB    00DH
BDC8 23                 DEFB    '#'
BDC9 0D         TEXT3:  DEFB    00DH
BDCA 53                 DEFM    'START ENDE'
BDD4 0D                 DEFB    00DH
BDD5 23                 DEFB    '#'
BDD6 0D         TEXT4:  DEFB    00DH
BDD7 54                 DEFM    'TAPE READY ? #'
BDE5 0D                 DEFB    00DH
BDE6 56         TEXT5:  DEFM    'VERIFY'
BDEC 0D                 DEFB    00DH
BDED 4E                 DEFM    'NAME........TYP'
BDFC 0D                 DEFB    00DH
BDFD 23                 DEFB    '#'
BDFE 0D         TEXT6:  DEFB    00DH
BDFF 4C                 DEFM    'LOAD'
BE03 0D                 DEFB    00DH
BE04 4E                 DEFM    'NAME........TYP'
BE13 0D                 DEFB    00DH
BE14 23                 DEFB    '#
BE15 0D         TEXT7:  DEFB    00DH
BE16 4E                 DEFM    'NEUE ADRESSE ? #'
BE26 0D         TEXT8:  DEFB    00DH
BE27 41                 DEFM    'ADRESSE
BE2E 0D                 DEFB    00DH
BE2F 23                 DEFB    #'
BE30 0D                 DEFB    00DH
BE31 0D         TEXT9:  DEFB    00DH
BE32 42                 DEFM    'BASIC-START
BE3D 0D                 DEFB    00DH
BE3E 23                 DEFB    #
BE3F 0D         TEXT10: DEFB    00DH
BE40 4B                 DEFM    'KEIN BASIC'
BE4A 0D                 DEFB    00DH
BE4B 23                 DEFB    #
8900                    END     START
```


Thema: SUPERTAPE

Das im Info Nr. 7 (Seite 44-50) vorgestellte Bandaufzeichnungsverfahren SUPERTAPE ist sicher für einige Mitglieder des CLUB 80 so interessant, daß sie sich entschlossen haben, das Programm abzutippen und die Platine nachzubauen. Dazu zwei Tips:

1. Das Programm wird vom Heise-Verlag, der auch die Zeitschrift c't herausgibt für 7.-- DM auf Kassette verschickt! Man kann sich also die Mühe des Eintippens und Fehlersuchens sparen. Wer zudem noch etwas Zeit hat, kann ja warten bis das Programm in der Clubbibliothek auftaucht (was sicher nicht allzu lange dauert).
2. Die Platine für die benötigte Zusatzschaltung wird für unter 20,-- DM von verschiedenen Inserenten in der c't angeboten. Die Schaltung kann also auch von den in der Hardware nicht so erfahrenen Clubmitgliedern aufgebaut werden.

Einige Diskuser werden sich natürlich fragen, was man mit einem Tapeaufzeichnungsverfahren, sei es auch noch so schnell, soll. Die Antwort ist relativ einfach. Mit SUPERTAPE erhält man die Möglichkeit, Programme und Daten mit anderen (supertapefähigen) Computern auszutauschen! Weiterhin kann man sich damit vielleicht manche Backupdiskette sparen.

Ein sicheres Plätzchen (für TRS80)!

Im letzten Clubinfo wurde eine Schaltung veröffentlicht, mit der sich die Besitzer von VideoGenie-Computern ein sicheres Plätzchen für Maschinenroutinen in ihrer Maschine sichern können. Die Schaltung ist für TRS80-User nur mit erheblichen Änderungen brauchbar und somit mit Vorsicht zu genießen. Wer sich trotzdem in seinen TRS80 etwas mehr RAM einbauen möchte, der möge noch etwas Geduld haben. Ich bin z.Z dabei eine Schaltung aus einer älteren amerikanischen Zeitschrift (80 US) auszutesten, die speziell für TRS80-Computer 1920 Bytes zusätzlichen Speicher bringt. Die Schaltung und der zugehörige Artikel werden dann im nächsten Info veröffentlicht (wenn alles funktioniert!).

Hartmut Obermann

Bei den meisten heutigen Hobby-Computern wird die Programmiersprache BASIC verwendet. Das heißt aber (leider) nicht, daß ein BASIC-Programm zwischen zwei verschiedenen Computertypen direkt oder über Kassette ausgewechselt werden kann. Die BASIC-Befehle mögen dann wohl das gleiche bedeuten, aber die interne Verarbeitung dieser Befehle und die Speicherung auf der Kassette ist häufig ganz anders. Nichts geht mehr . . . ? Doch: Mit Hilfe von BASICODE, einer Art universeller, aus BASIC abgeleiteter Computersprache, ist es möglich, BASIC-Programme zwischen verschiedenen Computertypen auszutauschen.

BASICODE-2

Endlich ein "allgemeinverständliches" BASIC

Es ist ungefähr zwei Jahre her, daß die Hobbyskoop-Redaktion von Radio Hilversum in den Niederlanden auf die Idee kam, einen universellen Kode auszuarbeiten, mit dem man BASIC-Programme zwischen verschiedenen Computern austauschen kann. Denn daß BASIC nicht gleich BASIC ist, merkt man schnell, wenn man zum Beispiel ein Junior-Programm auf dem ZX-81 laufen lassen will. Obwohl es ein Standard-BASIC gibt, verwendet jeder Computer-Hersteller seinen eigenen "Dialekt" mit seinen eigenen (meist durchaus sinnvollen) Erweiterungen. Der Unterschied zwischen ostfriesischem Platt und Ur-Bajuwarisch ist nichts dagegen . . . Dazu kommen Probleme mit Speicheradressen und der internen Programmabarbeitung und als eines der wesentlichsten Probleme das der Übertragung. Bei den meisten Hobbycomputern wird der Kassettenrekorder als externer Massenspeicher verwendet. Die Speichermethodik (das Datenformat), die Übertragungsgeschwindigkeit und die gebrauchten Frequenzen sind auch wieder bei jedem Computertyp anders. Da half bisher nur eins: sich hinzusetzen, das Programm umzuschreiben und dann mühsam Zeichen für Zeichen in den eigenen Computer einzutippen.
Der BASICODE-Standard ist ein festgelegter Audio-Kode, mit dem BASIC-Programme auf der Kassette gespeichert werden können. Dabei ist es im Prinzip unwichtig, ob diese Programme vom Computer oder vom Radio oder vom Fernseher angeliefert werden. Denn daß diese drei "Lieferanten" in Frage kommen, wissen aufmerksame Elektor-Leser aus dem September-Heft. Durch die Standardisierung sind Abspeichern und Wiedereinlesen unabhängig vom Computertyp. Die einzelnen Festlegungen, auf die wir unten noch eingehen, betreffen:

- Übertragungsformat, -geschwindigkeit, -frequenzen
- zulässiger BASIC-Befehlssatz
- Speicheraufteilung
- Variablenbenennung
- Schirmformat

Das neue, universellere BASICODE-2 enthält neben anderen Verbesserungen vor allem eine Anzahl Standard-Unterprogramme. Dieses BASICODE-2 werden wir hier besprechen.

BASICODE auf dem Band

Beim BASICODE werden die beiden Frequenzen 1200 Hz und 2400 Hz verwendet. Eine volle Periode der 1200 Hz bedeutet eine logische 0, zwei volle Perioden der 2400 Hz eine logische 1.
Die einzelnen Bytes werden mit einer Geschwindigkeit von 1200 Baud gesendet beziehungsweise empfangen, wobei die Bytes folgendermaßen aufgebaut sind (siehe auch Bild 1):

- 1 Startbit (logisch 0)
- 8 Datenbits, beginnend mit dem LSB
- 2 Stoppbits (logisch 1).

Das BASIC-Programm wird Zeichen für Zeichen kodiert, also so, wie es beim Kommando LIST zu sehen ist. Von den internen Op-Codes des Computers wird kein Gebrauch gemacht: Alle Zeichen werden im ASCII-Kode dargestellt. Jedem BASIC-Befehl muß eine Leertaste folgen, jede BASIC-Zeile muß mit CR (Wagenrücklauf, Hex 8D) abgeschlossen werden. Das MSB jedes ASCII-Zeichens muß logisch 1 gemacht werden.
Ein vollständiges Programm auf der Kassette besteht aus folgenden Teilen:

- Ein "leader", bestehend aus einem 5 Sekunden langen Ton von 2400 Hz
- dem ASCII-Zeichen "start text" (Hex 82)
- dem BASIC-Programm in ASCII-Kode
- dem ASCII-Zeichen "end of text" (HEX 83)
- einer "checksum" (Prüfsumme)
- einem "trailer", bestehend aus einem 5 Sekunden dauernden Ton von 2400 Hz.

Die Prüfsumme ermöglicht es dem Anwender zu kontrollieren, ob das Programm fehlerlos eingelesen wurde. Sie wird durch die bitweise Exklusiv-ODER-Verknüpfung aller ihr vorangehenden Bytes einschließlich des "start text"-Zeichens gewonnen. Daher hat sie selber auch die Länge von 1 Byte (8 bit).

Das BASICODE-2-Protokoll

Allgemeine Absprachen

Es dürfen nur die BASIC-Befehle eingesetzt werden, die alle Computer verwenden. Diese Befehle sind in Tabelle 1 aufgelistet, wir kommen noch darauf zurück. Eine bestimmte Anzahl Programmzeilen ist für vorgegebene Unterprogramme reserviert. Diese Routinen ermöglichen Programmabläufe, die im Standard-BASIC nicht ohne weiteres möglich sind. Die Arbeitsweise dieser Unterprogramme wird noch beschrieben. Sie werden nicht mit dem Programm mitgesendet, sondern müssen im BASICODE-Übersetzungsprogramm enthalten sein oder vor dem Programmablauf gesondert eingegeben werden.
Das Schirmformat ist festgelegt auf 24 Zeilen zu je 40 Zeichen. Da manche Computer noch weniger Zeilen beziehungsweise Zeichen abbilden, wird empfohlen, nicht mehr als 16 Zeilen für ein Schirmbild zu verwenden und die Zeilen nicht länger als nötig zu machen. Ferner darf eine Programmzeile einschließlich Zeilennummer, Leerzeichen und CR maximal 60 Zeichen lang sein.

Programm-Aufbau

Bei den Zeilennummern wird bei BASICODE-2 folgende Einteilung eingehalten:
0-999: Standard-Unterprogramme. Diese Routinen sind speziell für den betreffenden Computer entworfen und werden mit dem Übersetzungsprogramm oder gesondert eingelesen.
1000: Erste Zeile des Hauptprogramms. Es muß folgende Form eingehalten werden:
1000 A = (Wert): GOTO 20: REM Programmname. (Wert) ist die maximale Anzahl Zeichen, die von allen Strings zusammen benötigt werden. Durch den Sprung nach Zeile 20 reservieren die Computer, bei denen dies notwendig ist, entsprechenden Speicherraum für die Strings.
1010: Erste frei zur Verfügung stehende Programmzeile.
1010-32767: Platz für die eigenen Programme. Eine bestimmte Systematik in diesem Programm ist zwar nicht Pflicht, wird aber von den BASICODE-Entwicklern empfohlen:
1000-19999: Hauptprogramm.
20000-24999: Unterprogramme mit Befehlen, die im BASICODE-2 nicht zulässig sind.
25000-29999: DATA-Statements.
30000-32767: REM-Statements. Diese können eine Beschreibung des Programms, Literaturhinweise und als letztes Namen und Anschrift des Programmverfassers enthalten.
Weiter wird empfohlen, die Zeilennummern in 10er-Schritten zu setzen. Über die Unterprogramme in den Reihen 20000-24999 muß man anmerken, daß sie natürlich nach Möglichkeit zu vermeiden sind. Wenn dies nicht geht, muß man wenigstens eine gute Beschreibung mitliefern, was in diesen Unterprogrammen geschieht.

Standard-Routinen in BASICODE-2

Wie diese Unterprogramme jeweils aussehen, ist stark vom jeweiligen Computertyp abhängig, so daß wir hier nur eine allgemeine Beschreibung geben.
GOSUB 100: Dieses Unterprogramm löscht den Schirm und setzt den Cursor auf Position 0,0 (links oben auf dem Schirm).
GOSUB 110: Setzt den Cursor auf eine bestimmte Stelle des Schirms. Die gewünschte Position wird durch die Variablen HO und VE festgelegt. VE gibt die Nummer der Bildschirmzeile an (die oberste Reihe hat die Nummer 0) und HO den Platz in der Reihe (außen links ist 0). Da das Schirmformat bei BASICODE-2 24 Reihen zu je 40 Zeichen ist, darf VE nicht größer als 23 und HO nicht größer als 39 werden. Die Variablen VE und HO dürfen durch den Aufruf dieses Unterprogramms ihren Wert nicht ändern.
GOSUB 120: Die Position des Cursors auf dem Schirm wird durch die Variablen VE und HO gespeichert. Hierbei ist VE = 0 die oberste Reihe und HO = 0 die erste Position in der Reihe (ganz links). Dieses Unterprogramm kann zusammen mit dem vorigen dazu eingesetzt werden, die Position des Cursors per Programm beliebig zu verändern (zum Beispiel bei selbstgestrickten Bildschirmspielen).
GOSUB 200: Prüft, ob eine Taste gedrückt ist, und setzt den zu dieser Taste gehörigen Inhalt in die Variable IN$. Ist in diesem Moment keine Taste gedrückt, ist IN$ leer. Im Prinzip kann man hier auch Steuerzeichen einlesen, aber dabei muß man vorsichtig sein, da diese bei manchen Computern eine unterschiedliche Bedeutung und Wirkung haben! Eine Ausnahme dabei ist RETURN. Das ist bei allen Heimcomputern der ASCII-Code 13.
GOSUB 210: Dieses Unterprogramm wartet, bis eine Taste gedrückt wird, und setzt dann den zugehörigen Inhalt in IN$. Vorsicht beim Programmieren, sonst wird das die berühmte Endlos-Warteschleife.
GOSUB 250: Erzeugt bei den Computern, bei denen dies möglich ist, einen Piepser. Tonhöhe und Tondauer sind hierbei nicht festgelegt.
GOSUB 260: Eine Zufallszahl zwischen 0 und 1 wird erzeugt und in die Variable RV eingeschrieben.
GOSUB 270: Der Variablenspeicher wird überprüft, und es wird festgestellt, wieviel Speicherplatz noch zur Verfügung steht. Die Variableninhalte werden dabei nicht verändert. Die Anzahl der freien Bytes wird in die Variable FR eingelesen.
GOSUB 300: Der Wert der Variablen SR wird als String in SR$ eingelesen. Der String darf kein Leerzeichen am Anfang oder Ende der Zahl enthalten (im Gegensatz zum Statement STR$, dieses Statement darf bei BASICODE-2 nicht verwendet werden).
GOSUB 310: Dieses Unterprogramm erzeugt einen String SR$, der folgendermaßen aufgebaut ist:
Der Wert von SR$ ist gleich dem Inhalt der Variablen SR und immer in Festkomma-

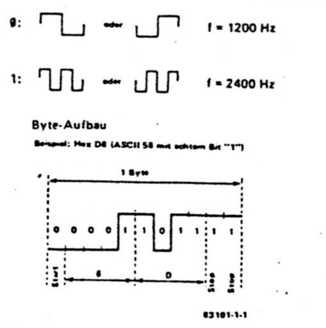


Bild 1. Der Aufbau des Übertragungsformates bei BASICODE. Achtung: Die Übertragung beginnt mit dem LSB (least significant bit = niederwertigstes Bit)!

BASICODE-2
Elektor Oktober 1983

Notation. Die Gesamtlänge von SR$ beträgt CT Zeichen, die Anzahl der Zeichen nach dem Dezimalkomma wird durch CN festgelegt. Wenn die Zahl nicht in das vorgegebene Format paßt, besteht SR$ aus CT Sternen. Wenn nötig, wird SR entsprechend gerundet. CT, CN und SR werden durch den Zugriff auf dieses Unterprogramm nicht verändert. Ein Beispiel: CT = 7, CN = 3 und SR = 0,6666 ergibt in SR$ " 0,667".
GOSUB 350: Printed SR$, schließt aber die Zeile noch nicht ab. Dadurch ist es möglich, verschiedene Strings hintereinander in eine Zeile zu schreiben.
GOSUB 360: Abschluß einer Schreibzeile mit einem Wagenrücklauf- und Neue-Zeile-Kommando.

Variable

Um die Auswechselbarkeit der Programme zu garantieren, sind auch die Variablen bestimmten Festlegungen unterworfen.

- Numerische Variablen sind immer reell und mit "single precision".
- Der Name einer Variablen darf höchstens aus zwei Zeichen bestehen, wobei das erste Zeichen ein Buchstabe sein muß. Das zweite Zeichen kann, wenn nötig, aus einem Buchstaben oder einer Ziffer bestehen. Bei String-Variablen steht noch ein $ nach dem Namen. Kleinbuchstaben dürfen in Variablennamen nicht verwendet werden.
- Bei logischen Variablen wird lediglich auf wahr oder unwahr geprüft. Eventuell vom Computer erzeugte Werte, zum Beispiel +1 für wahr und 0 für unwahr, dürfen nicht verwendet werden.
- Man darf nicht davon ausgehen, daß eine Variable durch den Programmstart automatisch auf 0 gesetzt wird.
- Stringvariablen dürfen maximal 255 Zeichen lang sein.
- Namen von Variablen dürfen nicht mit dem Buchstaben O beginnen, dieser ist reserviert durch die Standard-Unterprogramme.
- Weiter sind folgende Variablennamen nicht zulässig: AS, AT, FN, GR, IF, PI, ST, TI, TI$ und TO.

Für den Datenaustausch zwischen dem BASIC-Programm und den Standard-Unterprogrammen werden die Variablen HO, VE, FR, SR, CN, CT, RV, IN$ und SR$ verwendet.

BASIC-Absprachen

Tabelle 1 gibt eine Übersicht über alle BASIC-Kommandos und -Operatoren, die verwendet werden dürfen. Auch in diesem Punkt sind strenge Absprachen nötig, da es verschiedene BASIC-Varianten gibt. D Bedeutung der Befehle stimmt mit der offiziellen BASIC-Version überein, so da wir hierauf an dieser Stelle nicht eingehen Einige spezielle Punkte müssen wir bei d BASIC-Befehlen noch besprechen. So da: man nach einem GOSUB oder GOTO keir Variablennamen verwenden. Die Form: A = 1000 : GOTO A darf also nicht verwe det werden. Nach IF muß immer ein THEN folgen. Also: IF ... THEN A = 5 IF ... THEN 1000 oder IF ... THEN GOSUB 20000. Die Form IF ... THEN .. ELSE ist verboten. Kommentare oder mehrere Variablen nach einer INPUT-Anweisung sind nicht zugelassen. Die Forr INPUT "Der Wert = "; A$ ist verboten. Eine Zeilennummer nach RUN ist nicht zulässig. Beim TAB-Befehl muß man berücksichtigen, daß der eine Computer bei Null und der andere bei Eins zu zählen beginnt.

Tabelle 1. Zulässige BASIC-Befehle und -Operatoren.

ABS	DIM	INPUT	NOT	RETURN	STOP
AND	END	LEFT$	ON	RIGHT$	TAB
ASC	FOR	LEN	OR	RUN	TAN
ATN	GOSUB	LET	PRINT	SGN	THEN
CHR$	GOTO	LOG	READ	SIN	TO
COS	INT	MID$	REM	SQR	VAL
DATA	IF	NEXT	RESTORE	STEP	
+	↑	<>			
−	=	<=			
*	<	>=			
/	>				

Die Praxis

So, das waren die wichtigsten BASICODE Absprachen. Um damit zu arbeiten, brauc man jetzt noch das Übersetzungs-Program und die Standard-Unterprogramme. Dazı müssen wir für die "Allgemeinheit" an die Stelle passen: Denn hier gilt (hoffentlich z letzten Mal), daß halt jeder Computer etw anders ist. Glückliche Junior-Computer-Besitzer: Sie finden das Übersetzungsprc gramm und die Standard-Unterprogramm sowohl für den erweiterten Junior mit Mikrosoft-Basic als auch für den DOS-Junior an anderer Stelle in diesem Heft. Und wenn sie die Sendung "Know-how-Special" empfangen können, bekommen sie das auch noch vom Fernsehen auf Kassette geliefert (siehe Elektor im September-Heft). Und die Besitzer der vielen anderen Computer? Wenn alles gut gegangen ist, ist inzwischen das BASICODE Buch Nummer 2 erschienen – allerdings i: niederländischer Sprache. Neben einer au: führlichen Beschreibung von BASICODE-. enthält es auch die Übersetzungsprogramm für diverse Computertypen. Sobald das Buch der Redaktion vorliegt, werden wi: unsere Leser noch genauer informieren.
Müssen nun diejenigen, denen weder Rad Hilversum noch ein Fernsehprogramm noc dieses Elektorheft ein Übersetzungsprogramm liefert, etwa aufgeben? Nun, es gib schließlich viele, auf einen bestimmten Computertyp spezialisierte Computerclut bei denen man einmal anfragen kann.
Natürlich stellt sich jetzt mancher Leser di Frage: Kann ich denn, wo ich als Hamburç schon den WDR nicht empfangen kann, wenigstens Hilversum II empfangen? Leid können wir da auch keine eindeutige Ant wort geben, weil dies von verschiedenen "funktechnischen" Bedingungen abhängt Es kann vorkommen, daß ein Elektor-Lese in Bayern einen guten Empfang hat, wäh rend ein Leser im viel näher gelegenen Hessen leer ausgeht. Aber was spricht dagegen, durch Leserbriefe an Fernseh-Intendanten dazu beizutragen, daß Know how-Special in alle 3ten Fernseh-Programr. kommt?

BASICODE-Interface
Elektor Mai 1984

BASICODE-Interface

Für TRS-80 und Video-Genie

Um Basicode-Programme von der Kassette in den TRS-80 oder den Video-Genie laden zu können, benötigt man ein spezielles Interface, mit dem die FSK-Signale in Rechteckimpulse umgewandelt werden.
Die Schaltung (Bild 1) besteht aus einem Impulsaufbereiter-Teil (A1, A2 und "Umgebung"), einem Adreßdekoder (N1 ... N9 und N17) und einem Indikator (A3, D3). Sie wird an der Erweiterungsanschlußleiste des Computers angeschlossen, deren Belegung aus dem jeweiligen Handbuch hervorgeht.

Die Schaltung

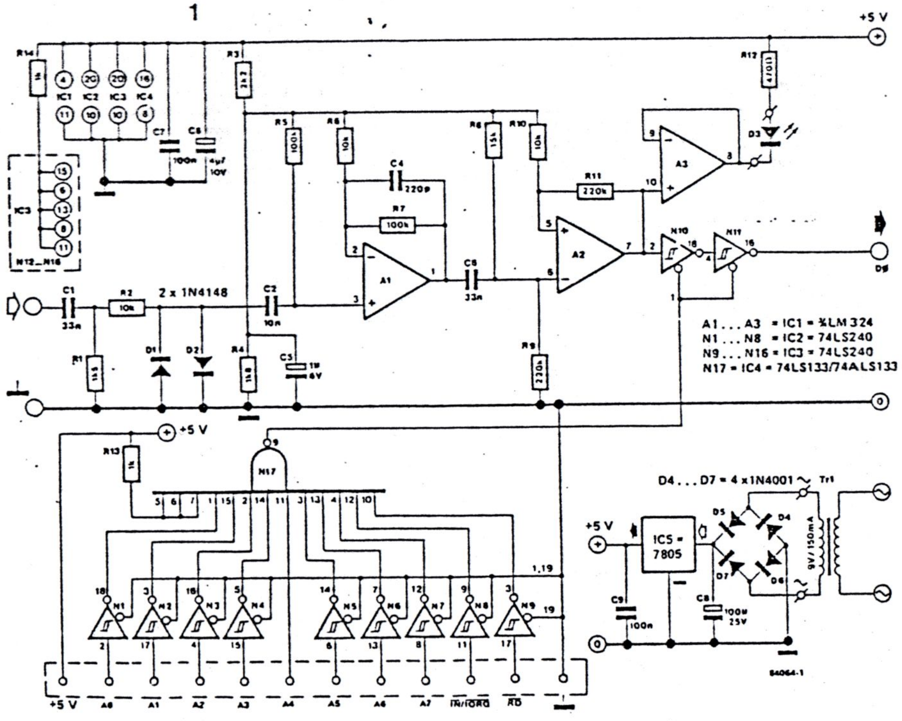
Wenn der Ausgang von N17 auf "0" ist, wird das Signal auf die Datenbuslinie D0 durchgegeben. Das Übersetzungsprogramm — es muß vorher eingegeben beziehungsweise über den normalen Kassettenanschluß eingelesen werden — sorgt hier auch für die Umwandlung der seriellen Eingangssignale in "normale Bytes" und deren Abspeicherung. Der Adreßdekoder wird durch das Übersetzungsprogramm so angesteuert, daß beim Einlesen N10 und N11 freigegeben sind.
Die Dioden D1 und D2 begrenzen das Signal vom Kassettenrekorder. Mit A1 ist eine Verstärkerstufe aufgebaut, mit A2 ein Schmitt-Trigger. Der OP A3 dient als Treiber für die Indikator-LED. Ist kein Eingangssignal vorhanden, sorgt R9 dafür, daß der Ausgang von A2 auf "1" liegt. Die LED D3 leuchtet dann nicht. Im Signalweg liegen die beiden invertierenden Treiber N10 und N11, sie werden vom Ausgang des Adreßdekoders gesteuert.

Aufbau

Auf der Platine (Bild 2) finden — mit Ausnahme des Trafos — auch noch die Bauteile für ein kleines Netzteil Platz. Allerdings braucht man diese nur, wenn das

Bild 1. Die Schaltung des BASICODE-Interface für TRS-80 und Video-Genie.

1



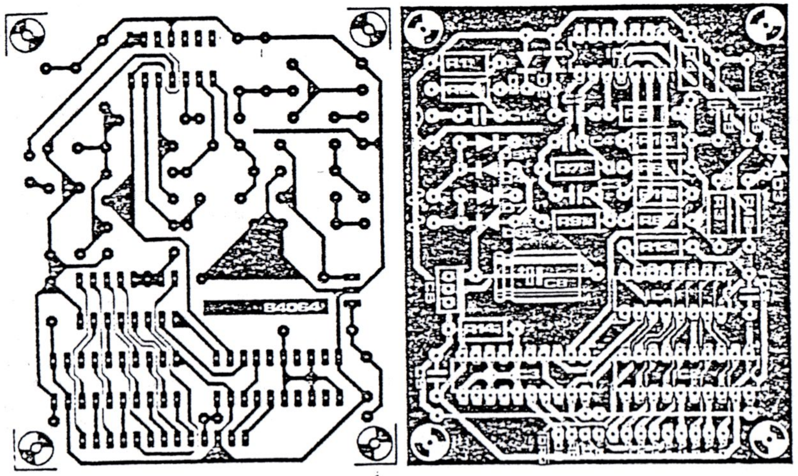


Bild 2. Mit der Platine fällt der Aufbau der Interface-Schaltung leicht.

Netzteil des Computers bereits "anderweitig" zusätzlich belastet ist: Die Schaltung des Interface benötigt lediglich 50 mA. In den meisten Fällen kann man also den 5-V-Anschluß auf der Erweiterungs-Anschlußleiste als "Kraftwerk" verwenden.

Auf der Platine gibt es eine Leiterbahn, die "im Nichts endet". Das ist *kein* Fehler, diese Leiterbahn dient als einfache Abschirmung des Eingangs gegen die direkt daneben anliegende Netzteil-Wechselspannung.

Abschließend noch ein Hinweis: Über das BASICODE-Verfahren ist ausführlich im Oktober-Heft 1983 berichtet worden.

Neues von BASICODE

Über das BASICODE-Verfahren haben wir ausführlich in Elektor vom Oktober 1983 berichtet. Das BASICODE-2-Buch (in Niederländisch und Englisch) von Radio Hilversum liegt der Redaktion inzwischen vor. Es enthält die Anpassungen und Übersetzungsprogramme für folgende Computer beziehungsweise Betriebssysteme:
Apple 2, BBC-Computer Modell A und B, Commodore (Pet und VC), CP/M-Systeme, DAI-Computer, Exidy Sorcerer, OSI-IP-Computer, Philips P-2000, Sharp MZ80 A und MZ80 K, SWTPC-6800, TRS-80 (I und III) und Video-Genie. Weitere Anpassungen sind bei Radio Hilversum — genauer: bei der Hobbyskop-Redaktion — in Vorbereitung. Außerdem soll in Kürze bei einem deutschen Verlag ein Sonderheft über BASICODE — ebenfalls mit Anpassungen — erscheinen, hier wissen wir aber noch nicht, für welche Computer (Rubrik Büchermarkt beachten).

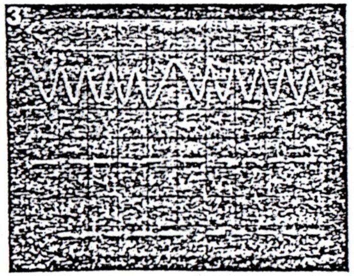


Bild 3. Auf dem Foto erkennt man oben das Eingangssignal und unten das Ausgangssignal der Interface-Schaltung.

Stückliste

Widerstände:
R1 = 1k5
R2,R6,R10 = 10 k
R3 = 2k2
R4 = 1k8
R5,R7 = 100 k
R8 = 15 k
R9,R11 = 220 k
R12 = 470 Ω
R13,R14 = 1 k

Kondensatoren:
C1,C3 = 33 n
C2 = 10 n
C4 = 220 p
C5 = 1 µ/6 V
C6 = 4µ7/10 V
C7,C9 = 100 n
C8 = 100 µ/25 V

Halbleiter:
D1,D2 = 1N4148
D3 = LED
D4,D5,D6,D7 = 1N4001
IC1 = LM 324
IC2,IC3 = 74LS240
IC4 = 74LS133 (oder 74ALS133)
IC5 = 7805

außerdem:
Tr1 = Netztrafo 9 V/150 mA sek.
Platine 84064

Abschließend nochmals die Adresse von Radio Hilversum:
NOS-Hobbyscoop
postbus 1200
1200 BE Hilversum
Nederland

Jan Lichtermann

Allerlei Nützliches für den TRS-80

TRS-80-Besitzer müssen manche Kleinigkeit bei ihrem Computer in Kauf nehmen, die auf die Dauer ärgerlich ist. In den meisten Fällen kann man mit wenig Aufwand Abhilfe schaffen. Die im folgenden angesprochenen Tips und Kniffe haben sich in der Praxis vielfältig bewährt.

Die Dauer eines Tastendrucks wird festgestellt

Die INKEY-Funktion hat den Nachteil, daß nach einmaligem Tastendruck die Dauer der Betätigung nicht ermittelt werden kann. Möchte man mit einer Taste, z. B. in einem Grafikprogramm, eine Dauerfunktion erreichen, so kann man die Tastatur mit der PEEK-Funktion direkt abfragen und in einer Programmschleife einen Befehl so lange ausführen, wie die Taste betätigt wird. Die *Tabelle* gibt Aufschluß darüber, welches Argument zur Abfrage einer bestimmten Taste erforderlich ist. Wird die Taste betätigt, so erhält man den zugehörigen Wert als Resultat der Abfrage. Ist keine Taste betätigt, ergibt die PEEK-Abfrage den Wert 0. *Bild* 1 verdeutlich das Prinzip der Tastaturabfrage durch PEEK.

Soll aus einem bestimmten Grund, etwa zur Erhöhung der Aufmerksamkeit, eine Stringvariable durch einen blinkenden Cursor eingelesen werden, so empfiehlt sich die Nachbildung des INPUT-Befehls durch die INKEY-Funktion. Durch die Wahl der Schleifenlänge kann man die Blinkfrequenz des Cursors beeinflussen. *Bild* 2 zeigt, wie das Einlesen eines Strings mit der INKEY-Funktion bewerkstelligt wird.

Ausgabe im technischen Format

Obwohl das TRS-80-Basic die komfortable PRINT-USING-Anweisung beinhaltet, ergeben sich bei der Ausgabe von Zahlen manchmal Schwierigkeiten. Von der Technik her schätzt man das Arbeiten mit den Kürzeln piko (10^{-12}), nano (10^{-9}), Mega (10^{6}), Tera (10^{9}), Giga (10^{12}). Von Kürzel zu Kürzel unterscheidet sich der Exponent um jeweils 3. Eine automatische Anpassung wäre deshalb in vielen Fällen wünschenswert. Das in *Bild* 3 dargestellte Unterprogramm erledigt das. Übergeben wird die auszugebende Zahl Z, ebenfalls verwendet werden die Variablen ZZ und V. Letzteres kann eine Ganzzahlvariable sein, lediglich ZZ und Z müssen vom selben Typ sein, also entweder von einfacher oder doppelter Genauigkeit.

Tastenprellen und Ladeprobleme

Vielen mag das Tastaturprellen beim TRS-80 schon Kopfzerbrechen bereitet haben. Sind die Tastenkontakte verschmutzt, so werden Buchstaben zum Teil doppelt geschrieben. Tandy empfiehlt dagegen das Reinigen der Tastaturkontakte oder die Verwendung eines kleinen Maschinensprachprogramms KBFIX. Dieses Programm konnte ich von Tandy kostenlos erhalten, und es arbeitet zu meiner vollen Zufriedenheit. Ist das Programm geladen, wird das Tastaturprellen wirkungsvoll per Software eliminiert.

Ein anderes beinahe noch kritischeres Problem ist das Laden der Programme vom Kassettenrecorder. Dabei ist die Lautstärke am Recorder peinlich genau

```
10 A=PEEK(14337):IF A=2 THEN PRINT"A";:GOTO 10
20 IF A=4 THEN PRINT"B";:GOTO 10
30 IF A=8 THEN PRINT"C";:GOTO 10
40 GOTO10
```

Bild 1. Das Programm gibt so lange einen der Buchstaben A, B oder C aus, wie die entsprechende Taste gedrückt ist

Tabelle zur Tastaturabfrage

Zeichen	PEEK()	Wert
@	14337	1
A	14337	2
B	14337	4
C	14337	8
D	14337	16
E	14337	32
F	14337	64
G	14337	128
H	14338	1
I	14338	2
J	14338	4
K	14338	8
L	14338	16
M	14338	32
N	14338	64
O	14338	128
P	14340	1
Q	14340	2
R	14340	4
S	14340	8
T	14340	16
U	14340	32
V	14340	64
W	14340	128
X	14344	1
Y	14344	2
Z	14344	4
frei	14344	8
frei	14344	16
frei	14344	32
frei	14344	64
frei	14344	128
0	14352	1
1 !	14352	2
2 "	14352	4
3 #	14352	8
4 $	14352	16
5 %	14352	32
6 &	14352	64
7 '	14352	128
8 (	14368	1
9)	14368	2
: *	14368	4
; +	14368	8
, <	14368	16
− =	14368	32
. >	14368	64
/ ?	14368	128
ENTER	14400	1
CLEAR	14400	2
BREAK	14400	4
↑	14400	8
↓	14400	16
←	14400	32
→	14400	64
SPACE	14400	128
SHIFT	14464	1
frei	14464	2
frei	14464	4
frei	14464	8
frei	14464	16
frei	14464	32
frei	14464	64
frei	14464	128

```
10 B$=""
20 PRINT CHR$(14);:FOR N=1TO50:A$=INKEY$:IF A$=""THEN NEXT ELSE GOSUB 100:NEXT
30 PRINT CHR$(15);:FOR N=1TO50:A$=INKEY$:IF A$=""THEN NEXT ELSE GOSUB 100:NEXT.
40 GOTO 20
100 IF ASC(A$)=13 THEN PRINT CHR$(14);A$;:GOTO 130
110 IF ASC(A$)=8 AND LEN(B$)>0 THEN B$=LEFT$(B$,LEN(B$)-1):PRINTA$;:RETURN
120 IF ASC(A$)<>8 THEN B$=B$+A$:PRINTA$;
125 RETURN
130 REM STRING STEHT ALS B$ BEREIT, NACH BEARBEITUNG MIT RETURN ZURUECKKEHREN !
```

Bild 2. Das Programm liest einen String ein, dabei wird ein blinkender Cursor erzeugt – Enter schließt den String ab

```
60000 IF Z=0 THEN V=1:GOTO60010 ELSE ZZ=ABS(Z):V=FIX(LOG(ZZ)/LOG(10))-FIX(LOG
      (ZZ)/LOG(10)/3)*3+2:IF ZZ<1 THEN V=V+2
60010 PRINTUSING STRING$(V,"#")+".###↑↑↑↑";Z:RETURN
```

Bild 3. Das Unterprogramm gibt die Zahl Z im technischen Format aus

einzustellen, da sonst immer wieder Ladefehler auftreten. Auch ist die Lautstärke je nach Bandsorte verschieden zu wählen. Um so erfreuter war ich, als ich las, daß Tandy eine Zusatzplatine einbaut. Der kostenlose Einbau (er erforderte zwar einige Schreibarbeit meinerseits, da die Firma Tandy nicht sofort reagierte) hat sich durchaus gelohnt. Ladefehler treten jetzt nur noch in Ausnahmefällen auf.

Weitere Hardwareverbesserungen einfachster Art sind der zusätzliche Einbau zweier Tastschalter rechts neben der Tastatur. In der Klappe zur Erweiterung des TRS-80 befindet sich ein Taster, der beim Schieflaufen des Ladevorgangs bzw. eines Maschinenspracheprogramms betätigt werden kann und einen nichtmaskierbaren Interrupt am Prozessor auslöst. Die Betätigung dieses Tasters ist recht umständlich: Man muß jedesmal die Klappe öffnen und ihn auf der Rückseite mühsam ertasten. Deshalb kann man einen einfachen Taster parallel dazu legen und neben der Tastatur montieren. Dies empfiehlt sich sehr für perimente mit Maschinensprache.

Der Prozessor kann damit jederzeit ohne Programmverlust unterbrochen werden. Bekannt ist jedem TRS-80-Anwender die nach dem Einschalten des TRS-80 erscheinende Frage: MEMORY SIZE?, die zur Reservierung eines vor Basic geschützten Arbeitsbereichs für Maschinensprache dient. Ausgelöst wird dieser Vorgang hardwaremäßig nach dem Einschalten durch den sogenannten Power-On-Reset. Arbeitet man viel mit Maschinensprache, so sind manchmal Umreservierungen nötig. Dies würde ein Aus- und Einschalten der Anlage erforderlich machen, wobei die Anlage einige Sekunden ausgeschaltet bleiben muß. Hier kann man einen weiteren Taster einbauen, der diesen Power-On-Reset simuliert. Dadurch braucht man die Anlage nicht an der Rückseite ein- und auszuschalten.

Betrachten wir einmal den Schaltungsausschnitt in *Bild 4*: Wird das Gerät eingeschaltet, so ist C42 entladen; am Eingang von Z53 liegt also L-Pegel an, damit auch am Ausgang von Z52. Dies veranlaßt den Prozessor dazu, mit der Befehlsausführung bei Adresse 0000 zu beginnen, sobald der Reset-Eingang wieder H-Pegel erhält. Dies geschieht dadurch, daß C42 über R47 langsam aufgeladen wird. Im Betriebszustand ist C42 also geladen. Legt man jetzt einen Taster parallel zu C42, kann man ihn auch während des Betriebs entladen und damit den Einschaltvorgang simulieren. Dazu muß man C42 auf der Platine suchen und eine entsprechende Leitung anlöten. Ich verwende hierzu eine abgeschirmte Leitung, damit im Videoteil keine wilden Schwingungen ausgelöst werden. Doch Achtung! Versehentliches Betätigen dieses Tasters löscht ein Basic-Programm im Speicher. Auch verliert man durch einen solchen Eingriff die Garantie des Herstellers.

Der Speicher wird erweitert

Bekanntlich kann ein 4-KByte-Modell auf 16 K erweitert werden. Dazu werden die Speicherchips ausgetauscht. Zusätzlich müssen zwei Shunts ausgetauscht werden. Führt man diese Arbeiten selbst aus, kann man eine Menge Geld sparen! Benötigt werden acht Chips vom Typ 4116 (450 ns oder schneller). Es handelt sich um dynamische NMOS-RAMs. Beim Einbau sind unbedingt die Handhabungsregeln von MOS-Bausteinen zu beachten, d. h. auf jeden Fall ist statische Aufladung zu vermeiden. Man entfernt die alten Speicherchips Z13 bis Z20 und setzt unter Beachtung der Richtung (Kerbe) die neuen ICs ein. Dann wechselt man die DIP-Shunts A3 und A71 aus. Das folgende gilt nur für die D-Version des TRS-80 (letzter Buchstabe der Nr. an Unterseite des Gehäuses) und „BASIC LEVEL II": Bei A3 sind sämtliche Kontakte zu überbrücken, bei A71 sind Brücken gemäß *Bild 5* herzustellen. Möchte man keine Shunts kaufen, so kann man die Verbindung mit Schaltdraht herstellen. Korrekte Verbindungen sind die erste Voraussetzung für den Betrieb, sonst läuft der Computer nicht mehr. Hat man eine andere Platinenversion oder kein Level-2-Basic, so sind die Brücken anders herzustellen! Die Funktion des eingebauten neuen Speichers läßt sich durch die Eingabe von PRINT MEM grob überprüfen: Bei leerem Arbeitsspeicher muß die Zahl 15572 erscheinen.

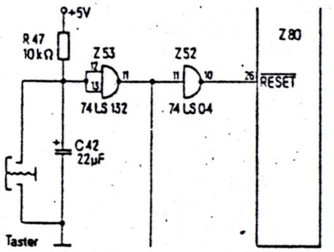


Bild 4. So wird der Power-On-Reset beim TRS-80 erzeugt

Literatur

[1] TRS-80 micro computer technical reference handbook. Radio Shack.

[2] Stübs, Martin: Programmieren mit TRS-80. Ing. W. Hofacker GmbH.

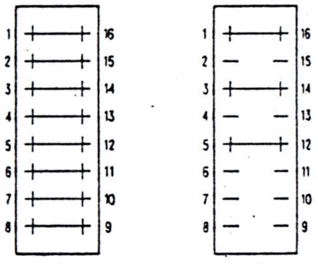


Bild 5. Erweitert man den Speicher des TRS-80 von 4 KByte auf 16 KByte, dann müssen bei der D-Version (letzter Buchstabe der Nummer auf der Unterseite des Gehäuses) die eingezeichneten Verbindungen hergestellt werden

Billiges CP/M von Bill Brewer

übersetzt von G. Schröder

Das Modell III kann in seinem ursprünglichen Zustand keine CP/M-Programme laufen lassen. Aber dieser Bericht zeigt Ihnen, wie Sie eine einfache Modifikation mit zwei Chips für fünf Dollar bauen können.

Das ROM belegt im Modell III die Adressen 0000-37FF hexadezimal. Außerdem liegen die Tastaturadressen und der Bildschirmspeicher in 37FF-3FFF. Nur der Bereich von 4000 bis FFFF kann im Modell III als Lese/Schreib-RAM benutzt werden. Die Aufteilung des Speichers wird in Bild 1 gezeigt.

Das ROM liegt im Modell III leider genau dort, wo auch die Programme unter CP/M stehen würden. CP/M würde Daten an der gleichen Stelle speichern, an der das Modell III-ROM ab 0000 liegt, und CP/M lädt alle Programme, die unter ihm laufen, in den Speicher ab 100 hexadezimal. CP/M und alle Programme, die unter ihm laufen, brauchen einen durchgehenden Speicher von 0000 bis zum Ende des Speichers, wo das Betriebssystem steht.

Abgesehen von dieser Speicherabweichung ist die Modell III Hardware durchaus dazu geeignet, leicht modifizierte 8080 oder Z80 Betriebssysteme laufen zu lassen; auch CP/M.

Diese Hardware-Begrenzung im Modell III wird durch nur sehr wenige der Schaltkreise dieses Computers verursacht. Wenn man sich diese Schaltkreise ansieht, wird rasch ein schneller und leichter Weg sichtbar, um diese Hindernisse aufzuheben.

Die Modell III CPU adressiert Signale über 16 Drähte in seinem Adressbus. Ein unmodifiziertes Modell III führt seine beiden höchstwertigen Adressdrähte in die eine Hälfte eines 74LS139 Multiplexers (U58), der einen Decoder Schaltkreis darstellt. Dieser Decoder gibt Signale aus, um eine der vier 16K-Speicher- bänke einzuschalten (Bild 2).

Wenn beide der höchstwertigen Adressdrähte (A14 und A15) auf einem niedrigen logischen Level sind, schaltet der Decoder die 16K-Speicherbank ein, die das Modell III ROM sowie die Tastatur und den Bildschirmspeicher enthält. Abhängig von den binären Werten dieser beiden Adressdrähte wird eine der vier 16K-Speicherbänke im Modell III Adressenraum ausgewählt (Tabelle 1).

Ein zweiter Decoder, der die untere Hälfte desselben Multiplexers benutzt, produziert Einschalt-Signale in der 16K-Bank, die das ROM enthält. Indem er Adressdraht A14 und den 0000-3FFF-Einschalter (ROMENABLEX) benutzt, schaltet dieser Decoder entweder den 8K ROM-Chip ein, der das meiste des Modell III Basic enthält, oder eine andere Gruppe von Schaltkreisen mit einem 4K-ROM, einem 2K-ROM, der Tastatur und dem Bildschirmspeicher. Dieser Segment-Decoder wählt diese verschiedenen Routinen zur Benutzung in ihrem jeweiligen Speicherplatz wie in Tabelle 2 gezeigt.

SIXTEEN KILOBYTE ADDRESS BANK SELECTED BY MRQ*, A16, AND A15

MRQ*	A16	A15	BANK SELECTED
HIGH	EITHER	EITHER	NONE
LOW	LOW	LOW	0000-3FFF
LOW	LOW	HIGH	4000-7FFF
LOW	HIGH	LOW	8000-BFFF
LOW	HIGH	HIGH	C000-FFFF

Table 1. 16K Bank Selection by U58

EIGHT KILOBYTE DEVICE SELECT BY ROM ENABLE* AND A14

ROM ENABLE*	A14	DEVICE SELECTED
HIGH	EITHER	NONE
LOW	LOW	8 KILOBYTE PROM
LOW	HIGH	PROM, KBD, VIDEO

Table 2. Segment Selection in Bank Containing ROM

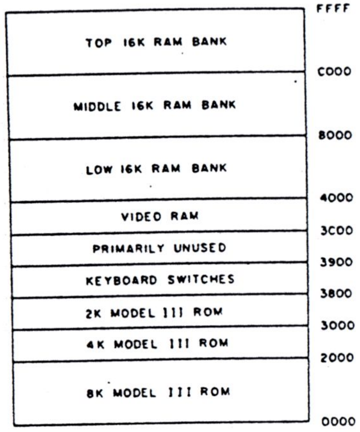


Fig. 1. Model III Memory Map

Eine Hardware-Lösung

Die einfachste Lösung, um das Modell III zu einer CP/M-fähigen Maschine zu machen, ist, das ROM aus dem Adressenbereich auszuschalten, den die meisten Programme benutzen, die nicht für das Modell III geschrieben wurden, und dafür das RAM einzu- schalten.
Das sieht zunächst sehr einfach aus. Man kann dies erreichen, indem man die Leiterbahnen, die im Bild 2 mit 0000-3FFF und C000-FFFF bezeichnet sind, vertauscht. Somit wäre das ROM, die Tastatur und der Bildschirm in C000-FFFF und die 16K-RAM-Bank der Bank 3 würde von 0000 bis 3FFF adressiert werden.
Dieser einfache Ansatz bringt aber zwei Nachteile mit sich. Erstens können keine Programme mehr laufen, die für das Modell III geschrieben wurden, denn ROM, Tastatur und Bildschirm würden ständig aus dem Speicherbereich herausgeschaltet werden, in dem sie laufen müssen.
Zweitens würde der Prozessor - ohne entsprechende Änderungen seines Kontroll-Schaltkreises - bei jedem RESET versuchen, ein Programm an der Adresse 0000 im RAM auszuführen; es gibt keine Möglichkeit, ein Programm in das RAM zu laden, bevor der Prozessor beginnt, von ihm Instruktionen zu holen.
Sie können die ursprünglichen Fähigkeiten bewahren, wenn Sie elektronische Schalter benutzen. Elektronische Schalter können die Adressierung so ändern, daß 0000 die Startadresse des ROMs ist, wenn Programme unter TRSDOS etc. laufen und daß 0000 die Startadresse des RAMs ist, wenn Sie Programme unter CP/M laufen lassen.
Elektronische Schaltungen achten auch auf das Problem mit dem Laden des Programms: Mit einem elektronischen Schalter können Sie Programme mit Routinen aus dem ROM laden, bevor der Schalter die ROM- und RAM-Adressen tauscht. Somit versorgt ein elektronischer Schalter das Programm mit einer dauernden Kontrolle durch den Prozessor.

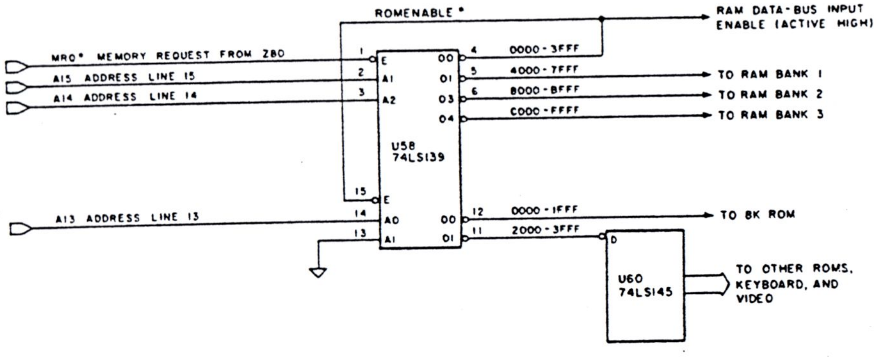


Fig. 2. Model III Address Decoder Circuit

Ein elektronischer Schalter

Ein Weg, um die elektronische Speicher-Umschaltung zu erreichen, ist, die Adressensignale des Z80 zu bearbeiten, bevor sie den Decoder erreichen. Während sich der Speicher im Standard Modell III-Modus befindet, kann der Z80 Instruktionen im Adressenbereich von 4000 bis BFFF ausführen, um die Signale von A15 und A14 zu invertieren, wenn beide WAHR oder beide FALSCH sind. Somit wird der Decoder genarrt, denn er denkt, daß eine vom Prozessor gelieferte Adresse im Bereich 0000-3FFF in Wirklichkeit im Bereich C000-FFFF liegt, und der Prozessor hat benutzbares RAM bei 0000.
Es gibt einige kommerzielle Modifikationen, die den Speicher in dieser Weise umschalten. Diese Methode hat einen Vorteil und einen Nachteil. Das Vorbearbeiten der Adressensignale ist mechanisch bequem; es kann durch eine einzelne Huckepack-Platine im Z80-Sockel erreicht werden, welche natürlich wieder den Z80 in einem Sockel aufnehmen muß.
Der Vorteil ist, daß der Computer nicht physisch verändert werden muß; die Huckepack-Platine ist keine unwiederrufliche Veränderung, denn der Z80 ist einer der wenigen gesockelten Chips im Modell III.
Der Nachteil ist, daß dieses Vorbearbeiten der Signale natürlich eine Verzögerung vor dem Erreichen des Decoders und der RAM-Bänke bewirkt, wodurch RAM-Operationen unzuverlässig werden. Diese Verzögerung ist im Modell III kritischer als im Modell I, denn die Speicher-Zyklus-Zeit (memory-cycle time) ist im Modell III um annähernd 20% kürzer als im Modell I.

Ein neuer Bank-Wahl-Decoder

Eine bessere Lösung stellt der Ersatz des Multiplexer-Schaltkreises (U58) durch einen neuen Decoder-Schaltkreis dar, wobei dieser die Adresszuweisungen für ROM und RAM umschalten kann und trotzdem nicht langsamer im Ansprechen der Adressen ist als das Originalteil.
Der neue Decoder, der den 16K Bank-Wahl Decoder in der oberen Hälfte von U58 ersetzt, besteht aus zwei anwählbaren Decodern in einem 74LS156 (Demultiplexer mit offenem Collector). Einer der neuen Decoder kann das ROM mit Tastatur und Bildschirm von 0000-3FFF und die RAM-Bank 3 von C000-FFFF einschalten (wie der Original Modell III Decoder). Der andere neue Decoder schaltet das ROM + Tastatur-Matrix + Bildschirmspeicher in C000-FFFF und RAM-Bank 3 in 0000-3FFF ein. Dieser Schaltkreis, um den 74LS156 Demultiplexer (genannt U58-1) herum aufgebaut, wird in Bild 3 gezeigt.
Es ist jederzeit immer nur einer der Decoder aktiv. Ein aktiv-HIGH Signal (REMAP), extern an die Decoder herangeführt, bestimmt, welcher der beiden Decoder die Speicherbank-Auswahl vornimmt. Weil die aktiv-LOW Ausgänge der Decoder als verdrahtete OR-Glieder (wired OR) geschaltet sind und der nicht angewählte Decoder HIGH (inaktive) Ausgänge für alle hat, kann nur der gewählte Decoder ein aktiv-LOW Signal über die Bank-Wahl-Anschlüsse senden.
Wenn REMAP logisch LOW (inaktiv) ist, wird der Decoder, der in Bild 4 oben gezeigt ist, eingeschaltet; er gibt Schaltsignale aus, die mit der Original Speicheraufteilung des Modell III übereinstimmen. Ist REMAP logisch HIGH (aktiv), wird der untere Decoder eingeschaltet; er gibt Schaltsignale aus, die das ROM mit Tastatur und Bildschirm in die obersten 16K des Speichers verlegen. RAM-Bank 3 wird in die untersten 16K des Speichers gelegt.

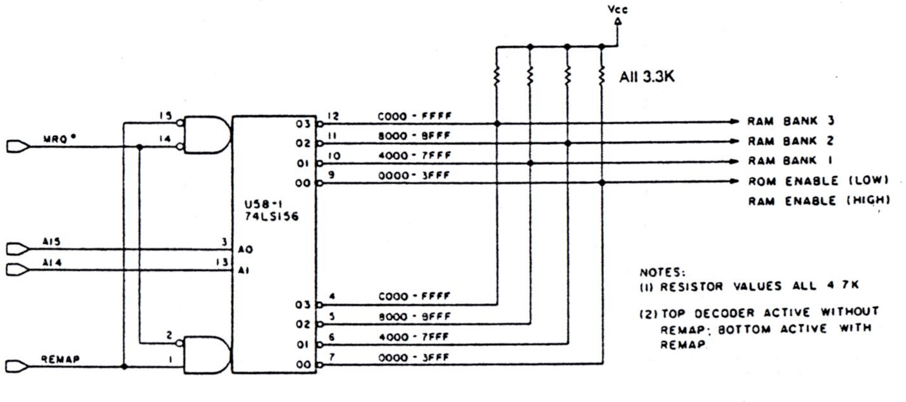


Fig. 3. Bank Decoder with External Request Line

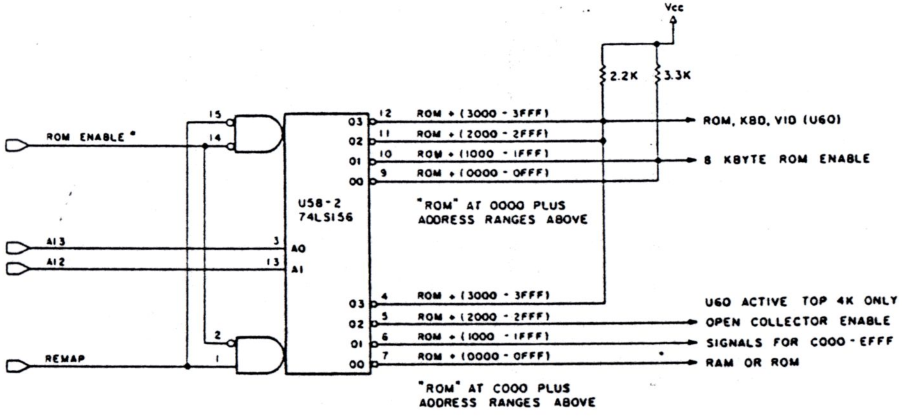


Fig. 4. Decoder for Bank Containing ROM, Keyboard, and Video

Treffen der Roboter

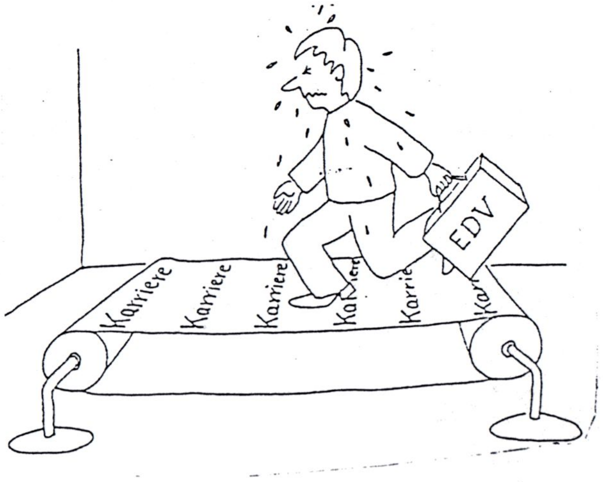
Chicago – Eine Baufirma schrieb einer Fabrik für elektronische Geräte:
»Unser Elektronengehirn hat ausgerechnet, daß sich die Kosten des von Ihnen geplanten Umbaus auf 125 000 Dollar belaufen.«
Die Firma erhielt umgehend die Antwort ihres Kunden:
»Unser Elektronengehirn hält Ihren Kostenanschlag für viel zu hoch. Wir schlagen daher sobald wie möglich ein Treffen der beiden Roboter in unserem Büro vor.«

Der zusätzliche Decoder

Ein zusätzlicher Decoder muß außerhalb von U58-1 benutzt werden, um die 8K Segment-Wahl Funktionen zu übernehmen, die vorher der untere Teil von U58 durchführte. Wie in Bild 4 gezeigt, besorgt der obere Decoder in einem zweiten 74LS156 Chip (genannt U58-2) die 8K Schaltungen für die unter Normalaufteilung zum ROM gehörenden 16K Speicher-Segmente. Der andere Decoder versorgt vier 4K Schalter im Adressbereich C000-DFFF unter der Neuaufteilung des Speichers.
Die Segment-Decoder werden folgendermaßen durch REMAP angesprochen. Ist REMAP LOW, wird das ROM mit Tastatur und Bildschirm wie im Original-Decoder geschaltet. Ist REMAP HIGH, produziert der untere Decoder vier Einschaltsignale in übereinstimmung mit Adressleitung A12. Eine dieser beiden Möglichkeiten ist immer aktiv, wenn der Prozessor C000-FFFF adressiert.
Der Draht, der "ROM + (3000-3FFF)" genannt wurde, schaltet in aktivem Zustand U60 ein, um das 4K Segment anzuwählen, welches das 2K ROM, die Tastaturmatrix und den Bildschirmspeicher enthält. Die drei zusätzlichen Signale sind Ersatz, und sie sind für die drei anderen 4K Segmente in C000-FFFF aktiv. Sie können zusammen benutzt werden, um 12K RAM außer der Reihe einzuschalten (womit wir insgesamt 60K hätten). Die Original 8K- und 4K-ROMs, die nutzlos sind, wenn sie von 0000 weggeschaltet sind, tauchen nicht mehr in der Speicheraufteilung auf. Dieser Raum ist frei für eine Speichererweiterung.
Der Ersatz von U58 durch die beiden 744LS156 Demultiplexer- Chips versorgt das Modell III mit seiner Original-Speicher- aufteilung und einer von CP/M nutzbaren Speicheraufteilung (siehe Bild 5). Außerdem erzeugen die neuen Decoder keine längere Verzögerung als die Decoder, die sie in dem jeweiligen Speicherzustand ersetzen. Statt Unzuverlässigkeit im Speicherzugriff bewirken sie sogar eine Speichererweiterung.

Das Signal, das den Speicher umorganisiert (REMAP)

Das REMAP-Signal, das zwischen den beiden Speicher Konfigurationen umschaltet, könnte von einem Schalter außerhalb des Modell III-Gehäuses ausgehen. Aber der Schalter kann nur betätigt werden, wenn der Z80 ein Programm ausführt, das in den Speicher 4000-BFFF geladen wurde (oder im ROM ist). Sonst "vergißt" der Programmzähler, welche Instruktionen er gerade ausführte.
Diese Einschränkung erfordert also, daß Sie ein REMAP nur dann ausführen, wenn die Kontrolle bei einem Programm liegt, das in den richtigen Bereich geladen wurde.
Das Signal kann von einem unbenutzten Ausgabe-Port des Modell III Logic-Boards versorgt werden. Der Ausgabebaustein U98, normalerweise benutzt, um Kassettenrekorder-Levels und ein Video Zeichengröße Signal zu senden, beinhaltet ein unbenutztes D-Flip-Flop.
Indem der unbenutzte Eingang dieses Bausteins mit Bit 0 des Data-Bus verbunden wird, kann die Ausgabe des freien D-Flip-Flop durch das niederwertigste Bit des Port EC kontrolliert werden. Ist dieses Bit LOW (inaktiv), geht der Ausgang des Flip-Flop (REMAP*) auf LOW, und die Speicher- aufteilung des Modell III ist für TRSDOS richtig. Ist dieses Bit HIGH (aktiv), wird der Remap-Befehl ausgegeben, welcher den Speicher für CP/M umorganisiert.
Das RESET* Signal von U98 ist normalerweise an Vcc (+ 5 Volt) gebunden. Die Information der beinhalteten Flip-Flops wird nicht unbedingt gelöscht, wenn Sie den Computer einschalten; auf jeden Fall wird es nie gelöscht, wenn Sie Reset von der Tastatur aus eingeben.
Um sicher zu gehen, daß der Computer mit dem ROM an 0000 (Startadresse des Prozessors) startet, muß das RESET* Signal zu U98 von der Unterstützungseinheit des Prozessors kommen und zwar genau von U57 Pin 13.



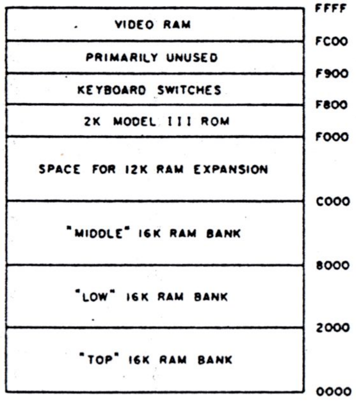


Fig. 5. CP/M-Compatible Model III Memory Map.

Physische Modifikationen

Die Decoder im Modell III zu ersetzen ist weitaus komplizierter als das Aufstecken einer Huckepack-Platine in den Z80-Sockel, obwohl an der CPU keine größeren Änderungen vorgenommen werden. Der neue Decoder-Schaltkreis wird auf einer separaten Platine plaziert, welche in einen Sockel gesteckt wird, der wiederum den Platz des ursprünglichen Decoders (U58) auf der Hauptplatine einnimmt (Fotos 1a und 1b).

Teile des Decoders, bestehend aus IC58-1, IC58-2, zwei 16-Pin-Sockeln, verschiedenen Widerständen und einem DIP-Stecker (der in den obern erwähnten, U58 ersetzenden DIP- Sockel gesteckt wird) werden auf einer passenden Lochraster- platine aufgebaut. Die Sockel für die Chips sind auf der Komponenten-Seite der Platine eingesteckt und die Verdrahtung wird auf der Lötseite mit ummanteltem Draht (Wrap-Draht) vorgenommen. Der DIP-Stecker ist auf die Platinenbahnen auf der Lötseite der Platine gelötet. Die Fotos 2a und 2b zeigen die Komponenten- und die Lötseite der Platine.

Die schwierigste Änderung ist das Entfernen des Original-Decoder-Chips (U58) und die Installation eines DIP-Sockels. Das Abschneiden jedes Drahtes des Chips und das Entfernen je eines Drahtes mit einem nicht zu stark erhitzten Lötkolben ist am wenigsten riskant. Ein Sockel, der den Decoder-Board-Stecker aufnehmen kann, wird dann anstelle des IC58 eingelötet. Dies kann ein konventioneller IC-Sockel sein, doch ein aus einzelnen Pins (IC-Kontakte Meterware) individuell aufgebauter Sockel ist erheblich sicherer und besser.

Die Reset-Leitung des Ausgabe-Latch (U98) muß auf dem Locig-Board gekappt werden. Mit isoliertem Draht, der direkt an die IC-Pins gelötet wird, werden folgende Verbindungen hergestellt:

- -Data-Bus Leitung 0 von U100, Pin 14, zu dem Ersatz Port EC Latch Input bei U98, Pin 4.
- -Port EC Latch Output von U100, Pin 5, zu einer Klemme auf der Decoder-Platte, welche REMAP an U58-1 und U58-2 weiterleitet.
- -/RESET (RESET*) von U57, Pin 13, zum Port EC Latch Reset bei U98, Pin 1.
- -Adressleitung U12 von U60, Pin 13, zur Decoder-Platte.

Ich habe die oben beschriebene Modifikation über hundert Stunden lang getestet und es ist kein Fehler aufgetreten. Die 8K und 4K Modell III ROMs sind ausgeschaltet und drei 4K Speichererweiterung Einschaltsignale funktionieren wie beschrieben.

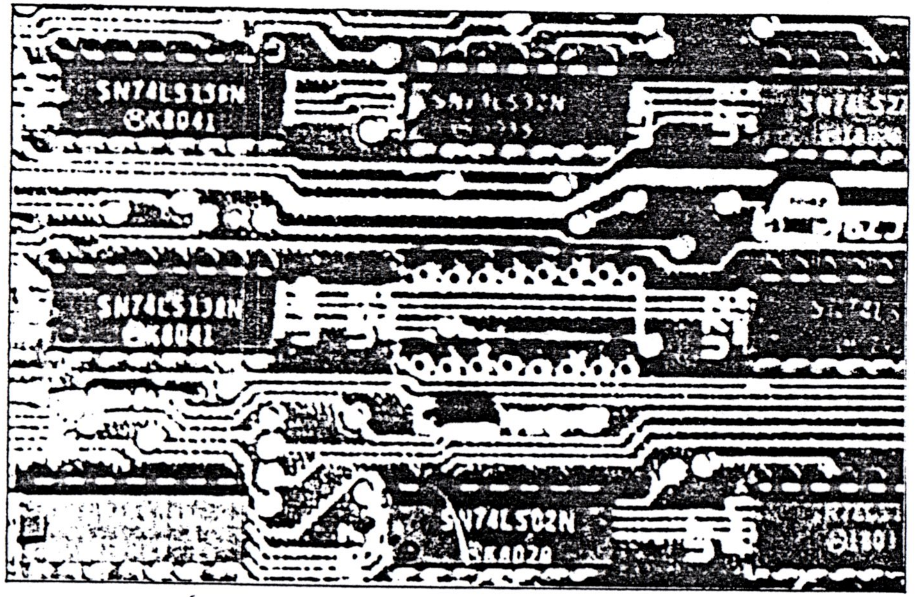


Photo 1a. Socket for New Decoder

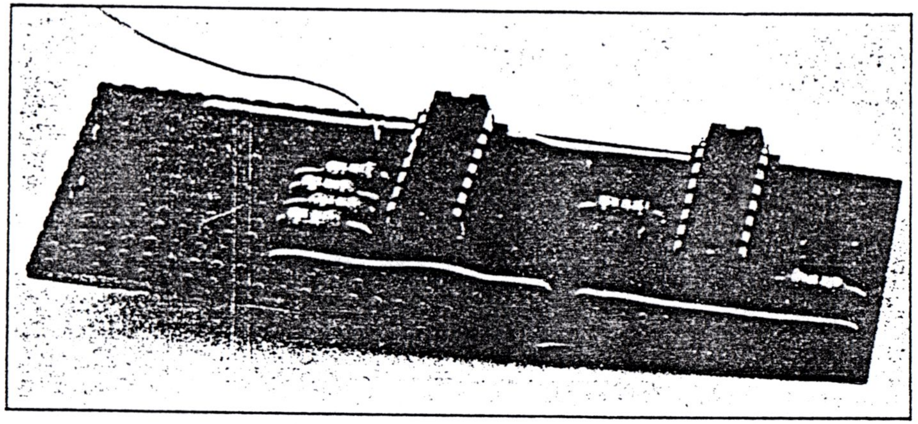


Photo 2a. Component Side of Decoder Board

Das Assemblerlisting für den Bootstrap Loader ist als "Paperware-Kopie" von Hartmut Obermann erhältlich

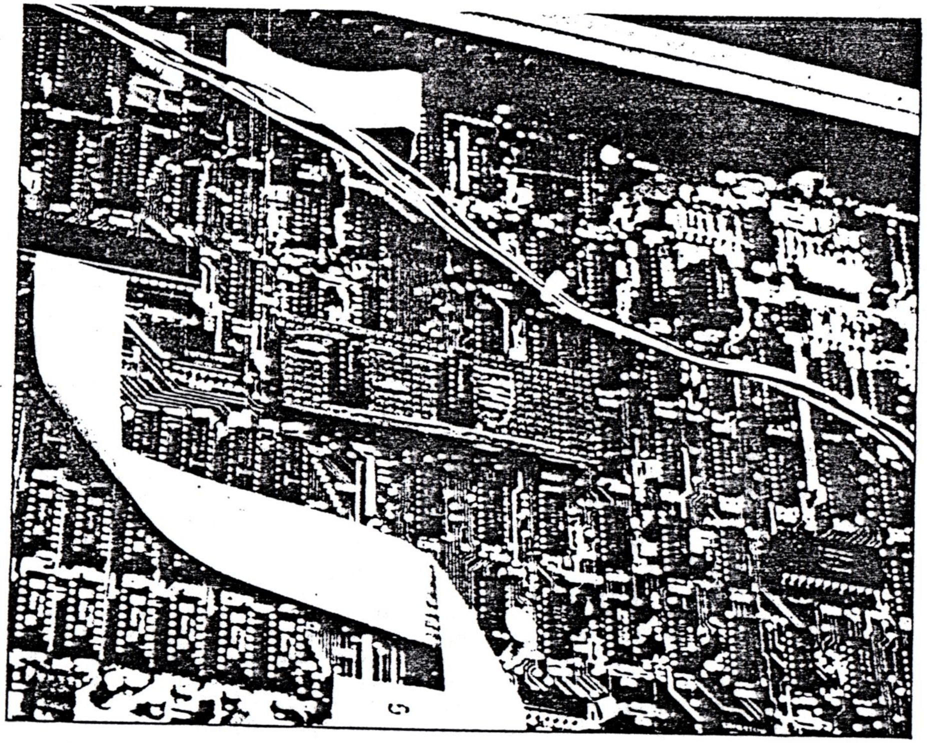


Photo 1b. New Decoder Installed

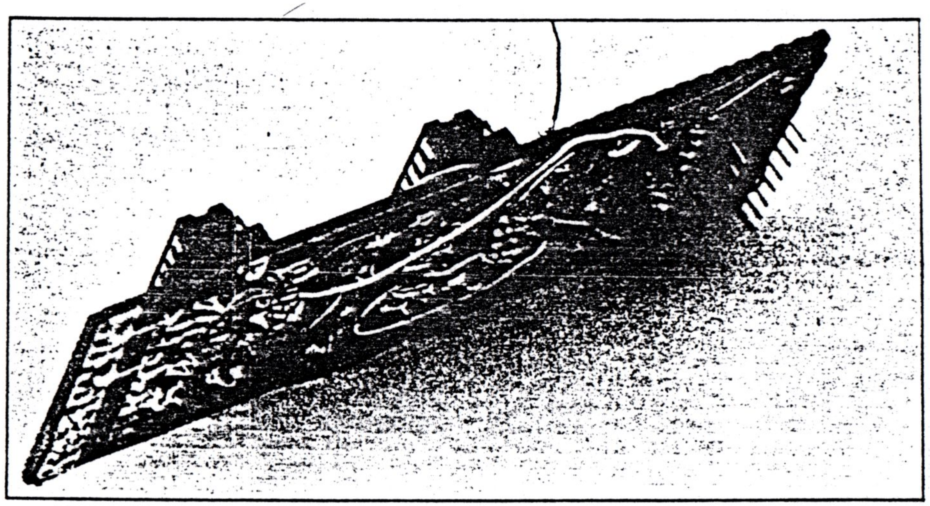


Photo 2b. Solder Side of Controller Board

Vorläufiges Design zur Speichererweiterung

Es wurden keine Tests mit einem Modell III durchgeführt, das zusätzlichen Speicher hatte. Die Speichererweiterung ist eine kostenlose Nebenerscheinung der Wahl unserer Chips für die Speicherneuaufteilungsmodifikation.
Speicherweiterungen können in derselben Technik ausgeführt werden wie die beschriebene Modifikation zur Neuaufteilung des Speichers. Stecken Sie eine Platine, die aus 8 DIP-Steckern und 16 Sockeln konstruiert ist, in die Sockel für eine der 16K RAM Banken. Verbinden Sie 8 der Sockel parallel mit entsprechenden Headern und stecken Sie die 4116 RAM Chips in sie. Sie funktionieren genau wie bei einem nagelneuen Computer.
Verbinden Sie alle Pins bis auf das CAS*-Signal (CAS* - Column Adress Strobe pin) der acht übrigen Sockel (für die acht zusätzlichen 4116s) parallel mit entsprechenden Steckern. Eine CAS-Leitung, die mit allen CAS* Pins der neuen RAM Chips verbunden ist, kann hergestellt werden, indem die drei 4K Einschaltleitungen des U58-2 als wired OR verdrahtet werden. Um das neue RAM Bank-Wahlsignal CAS$* zu erhalten, wird das daraus resultierende Signal mit dem Model III CAS* über ein OR-Glied verbunden (siehe Bild 6).
Nur weiteres Konstruieren und Austesten kann sicherstellen, daß diese Speichererweiterungsmodifikation über längere Zeiträume brauchbar ist. Auf jeden Fall sollte die Stromversorgung, die mit einer 50-Hertz Leitung arbeiten kann, die neuen RAMs mitversorgen können. Außerdem dürfte in diesem gut durchlüfteten Teil des Gehäuses kein Hitzestau auftreten.
Äußerste Vorsicht ist die einzige Garantie für einen Erfolg bei der Arbeit mit der CPU-Platine. Alle Verbindungskabel und Verbindungen, die in der Platine stecken, sind sehr dünn und sehr zerbrechlich.
Versuchen Sie sich nicht an dieser Modifikation, wenn Sie ein Neuling im Arbeiten mit Lötzinn sind.

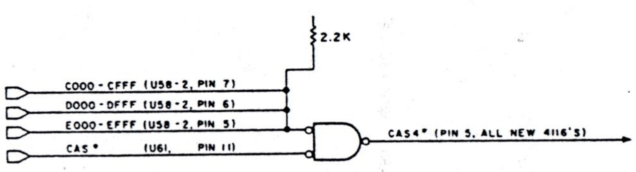


Fig. 6. Preliminary Logic for a New RAM Bank-Select Signal

Software Kontrolle

Die Software, die die Speicherumstellung schaltet, ist einfach. Um den Speicher zu re-organisieren, schreiben Sie die folgenden Instruktionen in Ihr Bootstrap Loader Programm:

```
LD    A,9
OUT   (0ECH),A
```

Dementsprechend benutzen Sie folgende Zeilen, um den Originalzustand Ihres Computers wiederherzustellen:

```
LD    A,8
OUT   (0ECH),A
```

Das am wenigsten signifikante Bit des Ausgabe-Bytes schaltet den Speicher um. Andere Bits in diesem Byte betreffen Video Zeilenlänge und Zeichensatz, Video Warte-Status, Kassettenmotor Schaltung und externe I/O Bus Operationen. Sind diese Bits aus (off), ist das Modell III in folgendem Zustand:
64 Zeichen/Zeile, Japanischen Kana Spezialzeichensatz, kein Video Warte Status und Kassettenmotor sowie I/O Bus sind ausgeschaltet. Mit Bit 3 an wird der griechische Zeichensatz angewählt.
Die meisten mit kommerziellen Modifikationen gebrauchten CP/M-Versionen können leicht an diese Modifikation angepaßt werden. Durchsuchen Sie nur den Bootstrap Loader und das Basic Input/Output System (BIOS, das maschinenabhängige Modul in CP/M) nach den Speicherumstellungs-Routinen, die für die jeweilige Modifikation gelten und ersetzen Sie diese durch die oben angeführten.
Als Alternative können Sie ein neues BOOT (s. unten) und BIOS für die Standard CP/M-Version schreiben. Der unten gelistete Bootstraploader kann in ein vorhandenen BIOS eingebaut werden.
Ein Bootstrap Loader für CP/M oder jedes andere Programm, das Sie auf der Diskette plaziert haben, ist im Listing gezeigt. Der Loader muß auf Track 0 , Sektor 1 einer Diskette im Modell III Standard Format von 18 256-Byte Sektoren per Track stehen. Das zu ladende Programm muß auf Track 0, Sektor 2 beginnen.
Die Speicheradressen, in die das Programm geladen wird, sind immer am Anfang des Programms angegeben und dort können Sie diese leicht verändern, um das Programm anzupassen. Wenn Ihre Diskette eingelegt ist, bringt das Modell III ROM bei einem RESET den Loader in das RAM bei 4300 hexadezimal und übergibt ihm die Kontrolle. Er lädt dann Ihr Programm und dieses Programm gibt Ihnen Möglichkeiten, die Sie vorher nicht hatten.

Bild 1: Modell III Speicher Aufbau
Bild 2: Modell III Adressen Decoder Schaltkreis
Tabelle 1: 16K Bank-Wahl beim U58
Tabelle 2: Segment-Wahl in ROM-Bank
Bild 3: Bank Decoder mit externer Wahlleitung
Bild 4: Decoder für Bank mit ROM, Tastatur und Bildschirm
Bild 5: CP/M-kompatibler Modell III Speicher Aufbau
Bild 6: Vorläufige Logik für ein neues RAM Bank-Wahl Signal
Foto 1a: Sockel für den neuen Decoder
Foto 1b: Neuer Decoder installiert
Foto 2a: Komponenten Seite des Decoder Boards
Foto 2b: Lötseite des Controller (Decoder) Boards

Impressum

1. Vorsitzender: Hartmut Obermann
Wilhelm-Baur-Straße 8
76135 Karlsruhe
Tel.: 07 21/ 85 40 68
BTX: 07 21/
FAX: 07 21/ 85 40 68

2. Vorsitzender: Gerald Schröder
Am Schützenplatz 14
21218 Seevetal
Tel.: 0 41 05/ 26 02

Hardwarekoordinator: Andreas Magnus
Bismarckstraße 29
45879 Gelsenkirchen
Tel.: 02 09/ 87 02 30

NewDOS-Diskothekar: Oliver Volz
Am Ochsenwald 37A
70565 Stuttgart (Rohrerhöhe)
Tel.: 07 11/ 74 40 51

CP/M-Diskothekar: Fritz Chwolka
Saarstraße 34
52457 Aldenhoven
Tel.: 0 24 64/ 89 20

C-128-Diskothekar: Günther W. Braun
Postfach 80 02 26
81602 München

Clubbücherei: Kurt Müller
Sophie-Scholl-Ring 3b
21502 Geesthacht
Tel.: 0 41 52/ 7 06 43

Redaktion: Jens Neueder
Gschlachtenbretzingen
Rudolf-Then-Straße 32
74544 Michelbach/Bilz
Tel.: 07 91/ 4 28 77
BTX: 07 91/ 44 47 22
FAX: 0 79 71/ 2 50 55

Bankverbindung: Club 80
Postgiro Sonderkonto CLUB 80
Obermann H., 8870 Günzburg
Postgiroamt Frankfurt
BLZ: 500 100 60
Kto.Nr.: 496 071 - 605

Autoren: Die Redaktion bedankt sich bei den im **Inhaltsverzeichnis** genannten Autoren für die Mitarbeit an der Club-INFO.
Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.

Schluß

Hallo Club 80'er,

wenn schon mal alles schief geht, dann auch beim INFO. Ihr haltet heute INFO 44 und 45 in den Händen, da bei der Produktion des 44. INFO´s der Kopierer sich für eine Reparatur (natürlich mit notwendiger Ersatzteilbestellung) entschlossen hatte. Nach erfolgreicher Reparatur war dann auch schon der nächste Info-Termin erreicht und so gibt es heute mal **"Zwei auf Einmal"**.

Da ich wenig Material für dieses INFO hatte, habe ich mich entschlossen, zum Hardwarethema passend, einmal die Hardwarebeiträge der ersten zehn INFO´s hier nochmals zu veröffentlichen. Vielleicht sind ein paar neue/alte brauchbare Tips für Euch dabei. Wenn nicht, ist es sicher einmal wieder schön in der Hardwarenostalgie der ersten Jahre zu schwelgen. Bei den heutigen **"Multi-Power-User-Chips"** ist man meist recht schnell nur noch als **Softi** und/oder **Power-User** der vorhandenen Software gefordert.

Nachdem es mit dem September-Redaktionsschluß nicht klappen konnte, wünsch ich mir für den Dezember natürlich rege Aktivität bei den Artikelschreibern, denen ich hierbei mal wieder meinen Dank für die Mitarbeit im Club 80 aussprechen möchte. Denkt bitte alle auch an den **DFÜ-Fragebogen**, der jetzt leider knapp terminiert ist.

Viel Spaß beim Schmökern in der Hardwarevergangenheit.
Bis zum nächsten INFO grüßt Euch Euer

Jens

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
1	Barendt	Harry	Hermann-Löns-Straße 7	GER 50181	Breedburg (Erft)	02272/ 7168	-	-	-
						-	-	-	-
2	Berndt-Jochum	Ilse	Stachelsgut 24	GER 51427	Bergisch Gladbach	02204/ 65254	-	-	-
	IBM, GENIE III, GENIE IIIs, SHARP Pocket 160■ Scanner					02204/ 65254	-	-	-
3	Bernhardt	Helmut	Preetzer Straße 75	GER 24143	Kiel	-	-	-	-
	IBM, Prof 180, CPU 280■ diverse PCs,Novellite-Netz,1496E,Mustek105+					0431/ 77578-20	0431/ 77578-99	2:242/262.26	-
4	Bielenberg	Georg	Erikaweg 1	GER 24568	Kaltenkirchen	04191/ 3751	-	-	-
	Atari 260ST, C128, Schneider Joyce■ Modem: LC2496 Digitech, Scanner: Supersc.III					04193/ 90430	-	-	-
5	Brans	Jörg	Tieloh 55	GER 22307	Hamburg	040/ 6906531	-	-	-
	IBM■ CD-Rom, Streamer, Soundkarte					-	-	-	-
6	Braun	Günther W.	Postfach 80 02 26	GER 81602	München	-	-	-	Discovery
	Commodore 128 D■ Akustikkoppler					-	-	-	07127/ 70107
7	Braun	Harald	Postfach 8011	GER 24154	Kiel	0431/35139	-	-	-
						-	-	-	-
8	Böckling	Ulrich	Juchaczstraße 61	GER 56203	Höhr-Grenzhausen	02624/ 4861	-	-	-
	IBM 386DX33,1040ST,VC20,C64,TRS80 M I,ZX81■ Modem, Videodat-, Videotextdecoder					02631/ 895168	-	-	-
9	Chwolka	Fritz	Saarstraße 34	GER 52457	Aldenhoven	02464/ 8920	-	-	-
	IBM 386+Co, Commodore, Apple, Z 280■ Modem 2.4					-	-	2:248/242:8	-
10	Dose	Volker	Dorfstraße 10	GER 24235	Brodersdorf	04343/ 1357	-	-	-
	GENIE IIIs mit Z180■ EPROMer, Modem					-	-	-	-
11	Halgasch	Gert	Großschönauer Straße 26	GER 02796	Jonsdorf	035844/ 636	-	-	-
	IBM 386DX■					-	-	-	-
12	Hartmann	Hans-Günther	Möwenstraße 9	GER 27804	Berne	04406/ 6911	04406/ 1071	-	-
	IBM 386SX, TANDY M4p, Z280-Kartenrechner■ Real Time Clock, Speed Up Kit 6,3MHz, 40MB-P					0421/ 248-2419	-	2:240/300.24	-
13	Hebecker	Ulrich	Büsnauer Straße 15	GER 70563	Stuttgart	0711/ 734800	-	-	-
	IBM 286+386, 128D, 1581, Kaypro484■					-	-	-	-
14	Held	Manfred	Stirner Straße 22	GER 91785	Pleinfeld	09144/ 6563	09144/ 8514	-	-
	IBM■ Modem ZyXEL 1496+, CD-ROM					0911/ 219-2245	-	2:2400/10.10	-
15	Hermann	Klaus	Forchenstraße 8	GER 72124	Pliezhausen	07127/ 71945	-	-	Discovery
	IBM 386■ ET 4000, Modem					-	-	2:2407/70.740	07127/70107
16	Hürdler	Manfred	Niederhoferstraße 29	GER 97222	Rimpar	09365/ 4235	-	-	-
	Victor Sirius 1 (IBM), CPC 6128■					-	-	-	-
17	Johnen	Willi	Hansemannstraße 1	GER 52351	Düren	02421/ 501305	-	-	-
	GENIE IIIs■					02421/ 33064	-	-	-
18	Kalb	Michael	Königstraße 10	GER 96242	Sonnefeld	09562/ 6387	-	-	-
	Schneider CPC, Joyce■ Modem, Drucker					-	-	-	-
19	Kauka	Dietmar	Straße des Friedens 37	GER 04552	Neukirchen (Borna/Leipz	03433/851019	-	-	-
						-	-	-	-
20	Kemmer	Jürgen	Dorfberg 7	GER 97232	Sulzdorf	09334/ 1050	-	-	-
	IBM 386, UltraSound, ZyXEL U1496E■ DCF-77-Empfänger parallel, IR-Sender seriell					-	-	2:247/2086.1	-
21	Kleespies	Andreas	Mainzer-Land-Straße 765	GER 65934	Frankfurt /Main	069/ 387432	-	-	-
						-	-	-	-
22	Kuhn	Eckehard	Im Dorf 14	GER 72636	Frickenhausen	07022/ 45417	-	-	-
	Atari ST 1040, TRS80 M I■					-	-	-	-
23	Linder	Jörg	Küstriner Str. 68	GER 15306	Seelow	03346/ 520	-	-	-
	KC 85/4 mit Floppy■					-	-	-	-
24	Lorenz	Walter	Mahräckerstraße 9	GER 60431	Frankfurt /Main	069/ 531656	-	-	-
	IBM 286/486, Z80-, HD 64180-Eigebau, Z280-T.■ Scanner, Soundkarte, ET 4000					-	-	-	-
25	Magnus	Andreas	Bismarckstraße 29	GER 45879	Gelsenkirchen	0209/ 144029	-	-	-
	IBM 386Dx, GENIE IIIs■ Modem 2400					-	-	-	-

Adressen-Liste CLUB80

Stand vom 05-10-1994
Seite 2

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat Tel. geschäftl.	Telefax privat FAX geschäftl.	BTX FIDO-Node	Mailboxname Mailboxnummer
26	Mahlert	Herbert	Hohenbudbergerstraße 112 A	GER 47229	Duisburg	02065/ 47217 02065/ 902592	- -	- -	- -
	IBM, GENIE I■ c't Videotext-Karte, Vobis Videodat-Decoder								
27	Messerschmidt	Kurt	Hanns-Eisler-Straße 54	GER 10409	Berlin	- -	- -	- -	- -
28	Müller	Kurt	Sophie-Scholl-Ring 3B	GER 21502	Geesthacht	04152/ 70643 040/ 89983403	- -	- -	- -
	IBM, Atari Mega ST4■ ATONCE 386,OverScan,Scanner,HBS640-T36,Modem								
29	Mössel	Franz	Schafferstraße 12	I 39012	Meran	0039-473/34178 0039-471/980496	- -	- 2:333/400	- -
	IBM, Joyce, Workemate-Bullet■ Modem								
30	Neueder	Jens	Rudolf-Then-Straße 32	GER 74544	Michelbach / Bilz	0791/ 42877 07971/ 250-50	- 07971/ 250-55	791444722 1001667	CCWN 0715168434
	IBM 286/386/486, Atari ST 1040, TRS80 MI■ Modem, Sound Galaxi NX, Scanner, Tape, Borsu								
31	Neumann	Christof	Zeitblomstraße 22/2	GER 89077	Ulm /Donau	0731/ 6022568 0731/ 9749720	- -	- -	- -
	IBM, Tandy MII, Tandy M4p■ Novell								
32	Nitschke	Stefan	Germenenstraße 5	GER 75045	Walzbachtal 1	07203 /452 -	- -	- -	- -
	CPU 280■ NEC 7220 1024*1024 Grafikkarte								
33	Obermann	Hartmut	Wilhelm-Baur-Straße 8	GER 76135	Karlsruhe	0721/ 854068 -	- -	- 2:241/7922.10	SYNREL 08282/ 4311
	IBM486, Tandy M4p, Epson PX-8■ Modem, Scanner								
34	Peters	Jürgen	Heukoppel 14	GER 22179	Hamburg	040/ 6412371 -	- -	- -	- -
	IBM, Z280, MZ821, SHARP■ Modem, Soundkarte, Video, Drucker								
35	Retzlaff	Bernd	Kleiner Sand 98	GER 25436	Uetersen	04122/ 43551 04103/ 605310	- -	- -	- -
	IBM 386, C64, GENIE I■								
36	Rinio	Gerd	Rennbahnstraße 9	GER 22111	Hamburg	040/6552630 -	- -	- -	- -
	IBM 486DX66/2, RTS 80, TRS 80 RS■ Modem								
37	Ruschinski	Claus	Pommernstraße 21	GER 45770	Marl	02365/ 34646 -	- -	- -	- -
	IBM 386, TRS80 M I■ Highscreen-Scanner, CoProz IIT387								
38	Schimmer	Jörg	Stettinerstraße 28	GER 60388	Frankfurt	06109/ 35336 069/ 3800-2385	- -	- 2:249/70.9	- -
	IBM 486, Schneider CPC■ Modem CSR2400								
39	Schmid	Alexander	Entmannsdorf 5	GER 96317	Kronach /Gehülz	09261/ 53496 -	- -	- 2:2400/830	- -
	GENIE IIs, GENIE IIIs, CPU 280■ Modem 2400,Prommer80,Ramdisk,Club80Terminal								
40	Schmitz	Rainer	Küferweg 12/1	GER 73099	Adelberg	07166/ 1397 07161/ 608-475	- -	- -	Discovery 07127/ 70107
	IBM,Portfolio,NCR Decision Mate V,Joyce,...■ Modem 1200, Märklin Digital Interface 6050								
41	Schoberth	Uwe	Petrus-Waldus-Straße 14	GER 75443	Oetisheim	07041/ 7254 0711/ 89394500	- 0711/ 89394513	- -	Discovery 07127/ 70107
	Alphatronic P3■								
42	Scholz	Hans-Werner	Spitalstraße 54	GER 41334	Nettetal	02157/ 3613 -	- -	- -	- -
	IBM 386, Prof 80, ITT 3030■ Prommer80								
43	Schroers	Horst-Dieter	Breslauer Straße 9	GER 85622	Feldkirchen	089/ 9032615 -	089/ 9043413 -	- -	- -
	IBM■ Modem, Scanner								
44	Schröder	Gerald	Arminiusstraße 2	GER 22525	Hamburg	040/ 8507131 040/ 54715334	- -	- -	- -
	IBM 386SX, Atari 1040STF, Z280■								
45	Schröer	Egbert	Joachimstraße 18	GER 46284	Dorsten	02362/ 75311 02362/ 49-9649	- -	- -	- -
	Portfolio, TRS80 MI, GENIE I, GENIE IIIs■								
46	Schulte	Hartmut	Entenschnabel 8	GER 31311	Uetze	05173/ 1248 -	05173/ 24631 -	- -	- -
	IBM, Z280 u.a.■ Scanner								
47	Sonnemann	Harald	In den Eckwiesen 9	GER 64405	Fischbachtal	06166/ 8512 06151/ 92-1265	- -	- -	- -
	NDR-Klein■ parallel/seriell, EPROMer								
48	Staimer	Ulrich	Dekan-Mayer-Straße 15	GER 86199	Augsburg	0821/ 993104 0821/ 2574-107	- -	- -	- -
	IBM (Siemens), C128, Reh Z280■ Video, Panasonic-Drucker								
49	Stumpferl	Stefan	Hasenberglstraße 57	GER 80933	München	089/ 3138193 -	//0893144001// -	- -	- -
	Amstrad CPC 6128+, GENIE IIIs■ Modem, 20MB-Wechselplatte, PSG&PIO,SIO, ...								
50	Sörensen	Rüdiger	Wiesbadener Str. 28B	GER 55252	Mainz-Kastel	06134/65342 -	- -	- -	- -

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
51	Tornow	Wilhelm	Elbblick 46	GER 21629	Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
	IBM 386DX, Atari Mega ST4, Tandy M4p. Streamer,Soundblaster,Modem 2400,F.A.K.S. 91					-	-	-	-
52	Vogl	Michael	Postfach 10 06 10	GER 41490	Grevenbroich /Laach	02181/45112	-	0218145112-0001	-
	IBM, Amstard CPC 464/6128. Modem Digitek 2400, Scanner Dart F.CPC					-	-	2:2433/27.2	-
53	Volkmer	Richart	Am Spörkel 69	GER 44227	Dortmund	0231/ 752574	-	-	-
	IBM XT, Apple LC, CPC 6128, Osborne, TandyMI. Modem 2400, ScanMan-Scanner					-	-	-	-
54	Volz	Oliver	Am Ochsenwald 37A	GER 70565	Stuttgart (Rohrerhöhe)	0711/ 744051	-	-	-
	IBM, GENIE IIs. Modem Avantec					0711/ 685-3013	-	-	-
55	Walter	Hans-Jürgen	Elssholzstraße 9	GER 10781	Berlin	-	-	-	-
						-	-	-	-
56	Werner	Heiko	Reichenberger Straße 5	GER 01129	Dresden	0351/ 4608612	-	-	-
	IBM 286/486. Modem 2400, Scanner					-	-	-	-

HARDWARE + SOFTWARE

COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80

ERFAHRUNGSAUSTAUSCH

Merry Christmas

Happy New Year

CLUBINFO

46. Ausgabe

Kontaktadresse:
Club 80
Hartmut Obermann
Wilhelm-Baur-Straße 8
Postfach ____
76135 Karlsruhe
Tel.: 0721/85 40 68
BTX: 0721/____
Anrufbeantworter &
FAX: 0721/85 40 68

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Neues vom Vorstand 1 - 2
Einladung zum Clubtreffen 95 3
Hartmut Obermann
Termine, INFO´s 4
Redaktion
Mitgliedvorstellung 5
Georg Bielenberg

Software

Mau Mau 7 - 14
Willi Johnen

Hardware

Bad News for IBM and Apple: C 128 15 - 22
Georg Bielenberg
Rantanplan's TAE-Info's 24 - 29
Rantanplan

Club80 Börse

Angebot - 30
Klaus Hermann

Sonstiges

Wie die verschiedenen Programmierer ihre
Fahrräder bauen - 23
Weizenprotokoll 31 - 36
Hartmut Obermann

Die letzten Seiten

Impressum 37
Schluß 38
Redaktion

Mitgliederadressenliste am INFO-Ende
Anfahrtsliste Clubtreffen 95 am INFO-Ende
Anmeldeformular Clubtreffen 95 am INFO-Ende

Neues vom Vorstand

Liebe CLUB 80-Mitglieder,

diesmal habe ich einige erfreuliche Neuigkeiten zu berichten. Los geht's!

1. CLUB 80-Treffen 1995

Wie in diesem Jahr, so hat sich auch für 1995 Alexander Schmid als Organisator für das CLUB 80-Treffen zur Verfügung gestellt, wofür ich mich jetzt schon bei ihm bedanken möchte! Alle weiteren Informationen zum Treffen wie Einladung, Anfahrskizze und Anmeldeformular findet Ihr in diesem Info.

2. CLUB 80-Logo - Terminverlängerung

Wegen dem, durch den defekten Kopierer verursachten, verspäteten Erscheinen des Infos 44, wird der Einsendeschluß für die Vorschläge zum neuen CLUB 80-Logo verlängert. Der neue Einsendeschluß ist der **01.03.1995.**

Nochmals zur Erinnerung:

Die Einsendungen sollten folgende Bedingungen erfüllen:

1. Ausführung in schwarz/weiß,
2. nicht größer als DIN A5,
3. geistiges Eigentum des Einsenders.

Nach dem Motto "Ohne Preis keinen Fleiß", wird hiermit ein postzugelassenes 2400 bps- Modem (wenn die Preise weiter so fallen, eventuell sogar ein 14400´er) als Preis für den Gewinner des Wettbewerbs ausgesetzt!

3. DFÜ im CLUB 80

Die Verspätung des 44´er Infos hat noch einen Termin über den Haufen geworfen, nämlich den Einsendeschluß für den "DFÜ im CLUB 80"-Fragebogen. Um die DFÜ-Interessierten im CLUB 80 nicht noch länger auf die Folter zu spannen, habe ich zunächst einmal ein Modem bestellt.

Es handelt sich dabei um ein postzugelassenes 14400 bps-Modem der Firma CREATIX, welches unter anderem von der Firma 1&1 in Montabaur vertreiben wird. Das Gerät kostet inklusive aller benötigten Anschlußkabel und einiger Software, die aber für CP/M-Rechner nicht nutzbar ist, zur Zeit ca. 220 DM.

Ab Anfang Januar steht das Modem allen CLUB 80-Mitgliedern zum Einstieg in die DFÜ zur Verfügung. Interessenten wenden sich am besten telefonisch (0721/854068 ab 17 bis 21 Uhr) direkt an mich. Die Warteliste wird natürlich in der Reihenfolge des Eingangs der Anfragen abgearbeitet.

Um die Warteliste nicht zu lang werden zu lassen, wird die Ausleihzeit zunächst auf drei Wochen beschränkt. Sollte es einen unerwartet großen Ansturm und damit unzumutbare Wartezeiten geben, wird noch ein zweites Modem beschafft.

Bei dem nicht unbeträchtlichen Wert, den das Modem darstellt, dürfte klar sein, daß vom jeweiligen Entleiher eine Kaution (220,-- DM) hinterlegt werden muß. Dies geschieht am besten und einfachsten in Form eines Euroschecks, welcher mir zugesandt wird. Der Scheck wird natürlich nur bei Verlust des Modems eingelöst und vernichtet bzw. zurückgeschickt, sobald das Modem wieder wohlbehalten bei mir eingetroffen ist.

4. Modell 3/4 - Emulator

Und noch etwas können Interessenten bei mir zum Testen anfordern, nämlich einen Emulator für die Tandy TRS 80 Modelle 3 und 4.

Im Info 44 informierte uns Alexander Schmid über einen "Modell 1 - Emulator für Dosen". Als ich, vor dem Erscheinen des Infos, Hans-Günther Hartmann von dem Emulator erzählte, klärte er mich auf, daß es auch einen Modell 3/4 - Emulator geben würde. Kurze Zeit später ließ Hans-Günther mir ein paar Kopien der Zeitschrift "Computer News 80" zukommen, in der für den Emulator geworben wird. Da mir das Programm sehr interessant und nicht zu teuer erschien, habe ich es für den CLUB 80 erworben.

Das Programm funktioniert nach erstem Augenschein sehr gut, allerdings gilt die gleiche Einschränkung wie beim von Alexander beschriebenen Modell 1-Emulator. Ohne die ROM´s und ohne die passenden Betriebssysteme und Programme kann man mit dem Emulator nur wenig anfangen. Aber da läßt sich sicher Abhilfe schaffen!

Wer sich den Emulator für MS-DOS ausleihen möchte, schreibt mir einfach einen kurzen Brief. Dieser sollte neben der Bestätigung, daß keine Kopie von Programm und Handbuch angefertigt wird, das gewünschte Diskettenformat (5¼/360k oder 3½/720k) sowie fünf Mark in Briefmarken für Porto und Verpackung enthalten. Die Anfragen werden in der Reihenfolge des Eingangs abgearbeitet. Die maximale Ausleihzeit beträgt auch hier drei Wochen.

5. CLUB 80 - Mitgliedsbeitrag

Da das neue Jahr ins Haus steht, ist auch das Thema CLUB 80 - Mitgliedsbeitrag wieder mal aktuell. Dazu folgendes:

- ob und in welcher Höhe ab 1995 wieder ein Mitgliedsbeitrag erhoben wird, entscheidet die Hauptversammlung während des CLUB 80-Treffens 1995
- bis diese Entscheidung gefallen ist, sollte niemand einen Mitgliedsbeitrag überweisen.

6. last but not least

Ich hoffe, daß die neuen, attraktiven Angebote des CLUB 80 von möglichst vielen Leuten genutzt werden und würde mich freuen, wenn der Club dadurch wieder mit etwas mehr Leben gefüllt würde.

Bei allen, die sich im Jahr 1994 aktiv am Clubleben beteiligt haben, möchte ich mich hiermit recht herzlich bedanken. Ganz besonderen Dank verdienen Jens Neueder für seine Arbeit am CLUB 80-Info und Alexander Schmid für die Organisation des CLUB 80-Treffens!

Ich wünsche allen CLUB 80-Mitgliedern ein frohes Weihnachtsfest, einen guten Rutsch ins neue Jahr und ein gesundes, glückliches und erfolgreiches Jahr 1995,

Einladung zum CLUB 80-Treffen 1995

Nach dem durchschlagenden Erfolg des von Alexander Schmid organisierten 94´er Treffens war für mich klar, wer das CLUB 80-Treffen 95 organisieren würde und wo es stattfinden würde. Blieb nur noch die Frage, ob sich Alexander nochmal würde breitschlagen lassen. Zum Glück zierte er sich nicht lange und erklärte sich ohne Zögern bereit, nochmals diese für den CLUB 80 sehr wichtige Arbeit zu übernehmen.

Da wir mit dem Hotel, in dem das 94´er Treffen stattfand, nicht ganz zufrieden waren, haben wir uns nach einem geeigneteren Treffpunkt umgesehen und sind fündig geworden. Das 95´er Treffen findet im Grenzgasthof, Allee 37, 96465 Neustadt/Fürth am Berg (Tel.: 09568/3096, FAX: 09568/7595) statt. Eine genaue Anfahrtbeschreibung wird durch Alexander erstellt und ist ebenfalls in diesem Info zu finden.

Als Zeitpunkt für das Treffen haben wir diesmal das Osterwochenende 14.-17. April 1995 ausgesucht. Die Entscheidung für diesen Termin ist uns nicht leicht gefallen, da Ostern ja ein Fest ist, welches man normalerweise im Familienrahmen feiert. Da sich aber die Gegend rund um Kronach durchaus auch für einen Osterurlaub mit der Familie eignet, hoffen wir trotzdem auf rege Teilnahme. Und selbstverständlich wird im Bedarfsfall ein Programm für die Familienangehörigen angeboten.

Ein festes Programm für das Treffen gibt es zu einem so frühen Zeitpunkt noch nicht, aber der grobe Rahmen steht:

- Anreise am Freitag den 14. ,nachmittags oder Samstag 15., vormittags
- am Samstag Nachmittag findet der offizielle Teil des Treffens, die Hauptversammlung statt
- am Sonntag weitere Programmpunkte (wie z.B. DFÜ-Seminar, Vorträge usw.)
- Abreise am Sonntag Abend oder Montag (wer länger bleiben will, kann das natürlich tun)

Um das Treffen möglichst interessant zu gestalten, sind wir natürlich auf die Mitarbeit aller CLUB 80-Mitglieder angewiesen! Aus diesem Grunde bitte ich um Vorschläge für die Ausgestaltung der "weiteren Programmpunkte" am Sonntag. Ob Vortrag, Vorführung oder Seminar, alles ist erlaubt und erwünscht! Und dabei muß es sich durchaus nicht gleich um eine semiprofessionelle Präsentation handeln, schließlich geht es um die Sache und nicht um die Art der Darstellung.

Auch was die Tagesordnung der Jahreshauptversammlung angeht, bin ich stark auf eure Mitarbeit angewiesen! Wer Punkte hierfür hat, möge mir diese bitte zukommen lassen. Obwohl auch hier das Motto "je früher desto besser" gilt, werden Vorschläge für die Tagesordnung noch bis Samstag , 14.04.95 12:00 Uhr angenommen.

Ich würde mich sehr freuen, wenn auch das CLUB 80-Treffen 1995 von möglichst vielen Mitgliedern dazu genützt würde, sich persönlich kennenzulernen und Erfahrungen auszutauschen.

In diesem Sinne wünsche ich uns allen eine gute Anreise und ein interessantes Treffen 1995,

Termine:

Club80-Clubtreffen Neustadt, Fürth am Berg 14.04.95 - 17.04.1995

Redaktionsschluß für das nächste **INFO** ist der:
01. März 1995

Achtung:

Bitte überprüft Eure Adress- & Sonstige -Daten:
Wir möchten zum Jahresanfang unsere **Adress- und Info-Datenbank** mal wieder aktualisieren. Nachdem sich sicher irgend etwas bei Euch, Eurem Rechnerequipment geändert hat, solltet Ihr uns diese Änderungen bitte mitteilen.

Hallo liebe Club80-Mitglieder,

nachdem ich schon fast ein Jahr Mitglied im Club bin, wird es wohl Zeit, mich vorzustellen. Ich bin 44 Jahre alt, 1,72 groβ und 84 Kilo schwer, verheiratet und habe (natürlich zusammen mit meiner Frau) zwei Kinder. Mein Wohnort Kaltenkirchen liegt etwa 30 km nördlich von Hamburg und hat ungefähr 16.000 Einwohner. Ich habe einen kaufmännischen Beruf. Zum Club80 bin ich durch Günther W. Braun gekommen, der mir ein Exemplar der Clubinfo überlieβ. Meinen Computer (Commodore 128) habe ich seit 1988. Hier bin ich auf das Betriebssystem CP/M+ gestosssen. Nachdem ich mich damit beschäftigt hatte, war ich erstaunt, was mit diesen Oldtimern schon alles möglich war. Ich bin auf dem Gebiet der Hardwarebasteleien so gut wie gar nicht bewandert, im Bereich der Software interessieren mich besonders kaufmännische Anwendungen. Seit kurzem verfüge ich auch über ZCPR, aber bis ich mir alles angeschaut habe, werden sicher noch einige Monate vergehen.

Da ich ein eher mitleidiger Mensch bin, kann ich keinen alten Computer in den Schrott wandern sehen. Ich habe deshalb einige Exemplare gesammelt und zum Leidwesen meiner Frau in unserer Wohnung abgestellt. Teilweise habe ich dazu dann noch entsprechende Literatur erworben. Wenn jemand also Übertragungen vom einen auf den anderen Rechner wünscht (Schneider 3"/3,5"<>Commodore 3,5"/5,25/"<>MS-DOS 3,5"/5,25"<>Atari 3,5" oder andere Kombinationen) würde ich es mal versuchen. Zum Schluβ noch eine Aufstellung der vorhandenen Rechner (Leider ist noch kein so richtig alter Z80-Rechner dabei):

Schneider	CPC464 (nur mit Datasette), Terminalprogramm vorhanden
Schneider	Joyce mit 3" + 3,5" Laufwerk
Commodore	128 D mit 5,25" + 3,5" Laufwerk
Commodore	64 mit 5,25" Laufwerk
Commodore	C16 mit Datasette
Commodore	+4 mit Datasette
Commodore	CBM 8032SK mit 1MB Doppelfloppy (leider ohne Handbücher) 32K RAM, 80-Zeichen Bildschirm

Texas Instruments TI 99/4A mit Cassettenrecorder

Atari 260 ST mit 2 x 3,5" Laufwerken (1 Schreib/Lesekopf und 2 Schreib/Leseköpfe)

XT (CPU NEC V20) mit 3,5" + 5,25" Laufwerk und 32MB Festplatte

So, damit möchte ich es vorerst bewenden lassen, grüβe alle Clubinfoleser und wünsche unserer Clubinfo noch mindestens 100 Ausgaben.

Georg Bielenberg

Hallo Ihr Genie III,s-Freaks!

Gibt es eigentlich noch solche ? Ich bin ein eingefleischter GENIE III,s-Fan und werde es auch nicht seinlassen.Obwohl ich schon viele Möglichkeiten hatte auszusteigen, bleibe ich bei meinem GENIE IIIs.
Schade ist nur, daß über diesen Computer und dessen Sprache vielzuwenig im CLUB-Info berichtet wird. Ich meine einfach,es gibt bestimmt noch viele GENIE IIIs-Besitzer die hiermit arbeiten. Auf mein vor einem Jahr im CLUBINFO verbreitetes < TEXT 90 > habe ich noch nie Antwort erhalten. Obwohl dieses Programm schon sehr interessant war. Ich arbeite mit diesem < TEXT 90 > (überarbeitete Version) schon 10 Jahre, bearbeite damit meine gesamte Buchführung und die Finanzamtverwaltung meiner Ehefrau sowie die Lagerverwaltung der Firma wo ich arbeite mit 10.000 Artikeln.
Und das sage und schreibe nur mit einer Festplatte von 5 MB und 2 LW.

Ich weiß auch nicht,ob sich schon mal jemand intensiv mit dem RDL/BASIC des GENIE IIIs befaßt hat. Jedenfalls ich habe es schon.Trotz großer Probleme beim compilieren, bin ich zufrieden.

Ein Kompliment dem CLUB80-Mitglied Volker Dose aus Brodersdorf,der mir im Jahre 1993 mal geholfen hat, komplizierte Compilierarbeiten mit von ihm zur Verfügung gestellten Spezialprogrammen zu erledigen. Einiges hiervon konnte ich gut gebrauchen.Schade Volker,daß Du nicht mehr mit dem G-DOS arbeitest, ich kann mich nicht mehr davon trennen.Vielleicht liegt es daran,daß die Programierung hiermit kinderleicht ist.

In meiner Freizeit habe ich unter anderem eine Spielesammlung programiert zu dem ich heute meine neue Erungenschaft, nämlich das < MAU-MAU >-Spiel vorstellen möchte. Dieses Spiel hat Selbststart aus einem SPIELESAMMLUNG-MENü heraus. Es läuft nur im RDL-BASIC, da hier hochauflösende Graphik verarbeitet wurde.

Wer sich die Mühe macht, dieses MAU-MAU-Spiel abzutippen, wird bestimmt seine Freude nachher beim Spielen haben. Nach richtigem abschreiben ist das Spiel ohne Fehler.

Nun viel Spass

Euer Willi Johnen
52351 Düren
Hansemannstr.1


```
10 '#)) Programmname:  < MAUMAU01/SPL >                    <<#
20 '#)) Spiel: MAUMAU-Kartenspiel                          <<#
30 '#)) Stand: 07.11.1994            18.00 UHR             <<#
40 '#)) Bildschirmformat: 80x24                            <<#
50 '#)) Mit hochauflösende Graphik                         <<#
60 CMD"80":CLS:HCLS:HONO:GOSUB3760
70 HBIG25,155,2,6,5,"MAU-MAU":HBIG35,138,2,1,2,"EIN NOSTALGIE-KARTENSPIEL FüR JUNG UND ALT":HBIG65,80,1,1,2,"COPYRIGHT BY WILL
I JOHNEN  *  5.11.1994":HBIG125,55,1,1,2,"52351 DüREN - HANSEMANNSTR. 1"
80 PRINT§1615,"Zum Spielbeginn bitte die Taste  < ENTER >  drücken";:LINEINPUTA$:CU=1
90 GOSUB3760:PRINT§185,"S P I E L A N L E I T U N G":PRINT§265,STRING$(27,140)
100 PRINT§403,"1. Jeder (Computer, Spieler) bekommt 4-10 Karten, wobei die Karten des":PRINT§486,"Computers für den Spieler un
sichtbar sind."
110 PRINT§563,"2. Wer zuerst alle Karten ablegen konnte, hat gewonnen."
120 PRINT§643,"3. Wenn eine SIEBEN abgelegt wurde,muß der Gegner 2 Karten ziehen, darf":PRINT§726,"danach aber wieder eine Kar
te ablegen"
130 PRINT§803,"4. Wenn eine ACHT oder ein AS abgelegt wurde,darf der betreffende Spie-":PRINT§886,"ler noch eine Karte ablegen
."
140 PRINT§963,"5. Wenn ein BUBE abgelegt wurde, darf der betreffende Spieler die Farbe":PRINT§1046,"bestimmen, mit der weiterg
espielt werden soll."
150 PRINT§1123,"6. Wie üblich gilt: Farbe auf Farbe, Zeichen auf Zeichen."
160 PRINT§1203,"7. Wenn Du nicht bedienen kannnst, dann gebe < Z > ein,wenn Du dann im-":PRINT§1286,"mer noch nicht ablegen ka
nnst,mußt Du wieder < Z > eingeben,der Com-":PRINT§1366,"puter übernimmt dann das Spiel"
170 PRINT§1531,"Zum Spielbeginn bitte die Taste  < ENTER >  drücken";:LINEINPUTA$:CU=1:GOSUB3760
180 CLEAR10000:DEFSTRE-F:DEFINTA-D:DIMEA(32),EB(32),EC(20),ED(20)
190 FA=STRING$(76,140):FB=STRING$(75,32):CU=1
200 '#---------------- Initialisierung ---------------------#
210 FORA=1TO32:READEA(A):NEXTA
220 PRINT§170," ";:INPUT"Wieviele Karten soll jeder bekommen < 4 oder 10 >";CV
230 IFCV<4ORCV>10THEN3900ELSEAC=CV
240 GOSUB3760:FORA=1TOCV
250 B=RND(32):IFEA(B)=""THEN250
260 EC(A)=EA(B):EA(B)="":NEXTA
270 FORA=1TOCV
280 B=RND(32):IFEA(B)=""THEN280
290 ED(A)=EA(B):EA(B)="":NEXTA
300 N=5
310 B=RND(32):IFEA(B)=""THEN310
320 ER=EA(B):EA(B)="":EQ=LEFT$(ER,2):EF=ER
330 GOSUB700:GOSUB2240
340 '#---------- Bildschirmausgabe --------------------------#
350 BV=0:EG="":EX="":EY="":EZ="":AX=0:GOSUB3860:PRINT§162,"Deine Karten:";:PRINT§242,FA;
360 FORA=1TO20:IFED(A)=""THEN380
370 EG=EG+ED(A)+"  "+CHR$(149)+"  "
380 NEXTA
390 IFLEN(EG)<=75THENEX=EG:GOTO420
400 IFLEN(EG)>75THENEX=LEFT$(EG,75):AX=(LEN(EG)-75):EY=RIGHT$(EG,AX):GOTO420
410 IFLEN(EG)>149THENEX=LEFT$(EG,75):EZ=MID$(EG,76,75):AX=(LEN(EG)-149):EY=RIGHT$(EG,AX)
L109
420 PRINT§322,EX;:PRINT§402,FA;:IFLEN(EG)>75THEN430ELSE450
430 PRINT§482,EY;:PRINT§562,FA;:IFLEN(EG)>149THEN440ELSE450
440 PRINT§642,EZ;:PRINT§722,FA;
450 PRINT§801," ";:IFN=1THENPRINT"<< Ich habe nur noch eine Karte auf der Hand >>          ";ELSEPRINT"<< Ich habe noch";N;"Ka
rten auf der Hand >>                         ";
460 PRINT§208,"Oben liegt: ";:PRINTER+"   (";EQ;")";:EK=EQ
470 '#------------ Inputroutine ----------------------------#
480 GOSUB3870:PRINT§1762,"Was spielst Du aus ":PRINT§1782,"";:INPUTEF:GOSUB3880:PRINT§801," ";
490 IFLEFT$(EF,1)="Z"ANDAD=0THENAD=1:GOSUB2550:GOTO0350
500 IFLEFT$(EF,1)="Z"ANDRND(2)=1THENPRINTTAB(3)"Klappt wohl immer noch nicht, wie?":EF=ER:AD=0:GOTO0680ELSEIFLEFT$(EF,1)="Z"THE
NPRINTTAB(3)"HAHA, Deine Gewinnchance ist gleich NULL !!":EF=ER:AD=0:GOTO0680ELSEZX=0
510 IFMID$(EF,3,1)<>" "THENEF=LEFT$(EF,2)+" "+RIGHT$(EF,LEN(EF)-2)
520 IFMID$(EF,4,1)=" "THENEF=LEFT$(EF,2)+RIGHT$(EF,LEN(EF)-3):GOTO0520
530 IFLEFT$(EF,1)=" "THENEF=RIGHT$(EF,LEN(EF)-1):GOTO0530
540 IFRIGHT$(EF,1)=" "THENEF=LEFT$(EF,LEN(EF)-1):GOTO0540
550 '#------------- Fehlererkennung ------------------------#
560 FORA=1TO20
570 IFED(A)=""THEN590
```

```
580 IFEF=ED(A)THEN610
590 NEXTA
600 PRINTTAB(3)"Du mogelst !!   ";EF;" hast Du garnicht auf der Hand":GOTO480
610 EL=RIGHT$(EF,LEN(EF)-3):EM=LEFT$(EF,2)
620 IFEL="BUBE"THEN650
630 EN=RIGHT$(ER,LEN(ER)-3):EO=EQ
640 IFEL<>ENANDEM<>EOTHENPRINTTAB(3)"Du hirnlose Ratte !    ";EF;" paßt überhaupt nicht !!":GOTO480
650 EQ=EM:GOSUB2740:ER=EF:IFEL="BUBE"THENGOSUB770:GOSUB1270:GOTO2240
660 IFEL="SIEBEN"THENGOSUB1000:GOSUB1270:GOTO2240
670 IFEL="ACHT"OREL="AS"THENGOSUB1130:GOTO2240
680 GOSUB2660:GOSUB1270:GOTO2240
690 '#----------- Anfangstest ------------------------------#
700 EB(1)=ER:EN=RIGHT$(ER,LEN(ER)-3):EO=EQ
710 PRINT§82,FA;:PRINT§162,"Die erste Karte ist ";ER;:PRINT§242,FA;:GOSUB3840
720 PRINT§801," ";:IFEN="BUBE"THENGOSUB1960:RETURN
730 IFEN="SIEBEN"THENGOSUB1000:GOSUB1270:RETURN
740 IFEN="ACHT"OREN="AS"THENGOSUB3240:IFCT=1THENRETURNELSEGOSUB1270:RETURN
750 RETURN
760 '#------- Spieler darf sich eine Farbe wünschen ----------#
770 PRINT§1842,"Welche Farbe wünschst Du Dir ":PRINT§1872,"";:INPUTEQ:GOSUB3760:PRINT§801," ";
780 IFEQ=ESTHENPRINTTAB(3)"Du spielst unüberlegt, die Farbe <";ES;" > lag doch schon !":GOSUB3840
790 IFAC-1<4ANDRND(2)=1THENPRINTTAB(3)"Soso, ";EQ;" hast Du":GOSUB3840
800 IFEQ<>"KR"ANDEQ<>"PK"ANDEQ<>"HZ"ANDEQ<>"KA"THENPRINTTAB(3)"Kille, Kille, Du kleiner Scherzkeks":GOTO770
810 FORA=1TO20:IFED(A)=""THEN830
820 IFED(A)=EFTHENED(A)="":GOTO840
830 NEXTA
840 FORA=1TO32:IFEB(A)=""THENEB(A)=EF:GOTO860
850 NEXTA
860 AC=0:FORA=1TO20:IFED(A)<>""THENAC=AC+1
870 NEXTA
880 IFAC>2ORRND(3)<>2THENER=EF:RETURN
890 FORA=1TO20:IFEC(A)=""THEN910
900 IFRIGHT$(EC(A),LEN(EC(A))-3)="BUBE"THEN920
910 NEXTA:ER=EF:RETURN
920 EH(1)="HZ":EH(2)="KA":EH(3)="PK":EH(4)="KR"
930 C=RND(4):IFEH(C)=EQ$THEN930
940 PRINTTAB(3)"Ich lege den ";EC(A):PRINT" und wünsche mir die Farbe ";EH(C):GOSUB3840
950 ER=EC(A):EQ=EH(C):EC(A)=""
960 FORA=1TO32
970 IFEB(A)=""THENEB(A)=ER:GOTO350
980 NEXTA
990 '#--------- Spieler hat eine SIEBEN gelegt -------------#
1000 AO=RND(5):ONAOGOSUB3630,3640,3650,3660,3670
1010 PRINTTAB(3)"Ich muß 2 Karten ziehen":GOSUB3840:AE=0
1020 B=RND(32):IFEA(B)=""THEN1020
1030 FORA=1TO20:IFEC(A)=""THENEC(A)=EA(B):EA(B)="":GOTO1050
1040 NEXTA
1050 AE=AE+1:IFAE<2THEN1020
1060 FORA=1TO32:IFEB(A)=""THENEB(A)=EF:GOTO1080
1070 NEXTA
1080 FORA=1TO20:IFED(A)=""THEN1100
1090 IFED(A)=EFTHENED(A)="":GOTO1110
1100 NEXTA
1110 ER=EF:EQ=LEFT$(ER,2):RETURN
1120 '#--------------- Spieler legt AS oder ACHT -----------#
1130 AF=1:PRINTTAB(3)"Du darfst nochmal .....";:GOSUB3840:GOSUB3760:AO=RND(7)
1140 IFAO=1THENPRINTTAB(3)"Glaube aber nicht,daß Du den Sieg schon in der Tasche hast !";
1150 IFAO=2THENPRINTTAB(3)"Da bin ich aber gespannt, was jetzt kommt ...";
1160 IFAO=3ANDN<=3THENPRINTTAB(3)"Na, hast Du noch eine SIEBEN ?";
1170 IFAO=4THENPRINTTAB(3)"Wer wagt, gewinnt !";
1180 IFAO=5THENPRINTTAB(3)"Frohlocke nicht zu früh..";
1190 IFAO=7THENPRINTTAB(3)"Na, geht Dir schon die Puste aus ?";:GOSUB3840
1200 FORA=1TO31:IFEB(A)=""THENEB(A)=EF:GOTO1220
1210 NEXTA
```



```
1220 FORA=1TO20:IFED(A)=""THEN1240
1230 IFED(A)=EFTHENED(A)="":GOTO1250
1240 NEXTA
1250 ER=EF:EQ=LEFT$(ER,2):RETURN
1260 '#------- Computer berechnet seinen Zug ---------------#
1270 GOSUB3760:PRINT§801," ";:EL=RIGHT$(ER,LEN(ER)-3):EM=EQ
1280 FORA=1TO20:IFEC(A)=""THEN1300
1290 IFRIGHT$(EC(A),LEN(EC(A))-3)<>"BUBE"AND(RIGHT$(EC(A),LEN(EC(A))-3)=EL)+(LEFT$(EC(A),2)=EM)<0THEN1320
1300 NEXTA
1310 GOTO1840
1320 FORA=1TO20:IFEC(A)=""THEN1410
1330 IFRIGHT$(EC(A),LEN(EC(A))-3)<>"ACHT"ANDRIGHT$(EC(A),LEN(EC(A))-3)<>"AS"THEN1410
1340 IFLEFT$(EC(A),2)<>EQANDRIGHT$(EC(A),LEN(EC(A))-3)<>ELTHEN1410
1350 PRINTTAB(3)"Ich lege ";EC(A):ER=EC(A):EC(A)="":GOSUB3840:BV=0
1360 FORB=1TO32
1370 IFEB(B)=""THENEB(B)=ER:GOTO1390
1380 NEXTB
1390 EQ=LEFT$(ER,2):GOSUB3240
1400 IFCT=1THENRETURNELSEGOSUB1270:RETURN
1410 NEXTA
1420 FORA=1TO20
1430 IFEC(A)=""THEN1630
1440 IFRIGHT$(EC(A),LEN(EC(A))-3)<>"SIEBEN"THEN1630
1450 IFLEFT$(EC(A),2)<>EQANDRIGHT$(EC(A),LEN(EC(A))-3)<>ELTHEN1630
1460 PRINTTAB(3)"Ich lege die ";EC(A);" und Du mußt 2 Karten ziehen ...!":BV=0
1470 AO=RND(5)
1480 IFAO=1THENPRINTTAB(3)"Na, damit hast Du wohl auch nicht gerechnet ..."
1490 IFAO=2THENPRINTTAB(3)"Das hast Du nun davon !"
1500 IFAO=3THENPRINTTAB(3)"Das ist Pech für Dich !"
1510 IFAO=4THENPRINTTAB(3)"Das würde mir aber stinken ..."
1520 IFAO=5THENPRINTTAB(3)"HAHA, das geschieht Dir recht ...":GOSUB3840
1530 ER=EC(A):EC(A)="":EQ=LEFT$(ER,2)
1540 FORDA=1TO32
1550 IFEB(DA)=""THENEB(DA)=ER:GOTO1570
1560 NEXTDA
1570 BN=0:PRINTTAB(3)"Du hast gezogen: ";
1580 FORC=1TO20
1590 IFED(C)<>""THENNEXTC
1600 B=RND(32):IFEA(B)=""THEN1600
1610 ED(C)=EA(B):EA(B)="":PRINTED(C)+"   ";
1620 BN=BN+1:IFBN>1THENFORAB=1TO1000:NEXTAB:RETURNELSE1580
1630 NEXTA
1640 EU="":AG=0
1650 FORA=1TO20
1660 IFEC(A)=""THEN1710
1670 IFLEFT$(EC(A),2)<>EQANDRIGHT$(EC(A),LEN(EC(A))-3)<>ELTHEN1710
1680 EU=RIGHT$(EC(A),LEN(EC(A))-3)
1690 IFEU="NEUN"THENAE=1ELSEIFEU="DAME"THENAE=2ELSEIFEU="KÖNIG"THENAE=3ELSEIFEU="ZEHN"THENAE=4
1700 IFAE>AGTHENAG=AE
1710 NEXTA
1720 IFAG=1THENEU="NEUN"ELSEIFAG=2THENEU="DAME"ELSEIFAG=3THENEU="KÖNIG"ELSEIFAG=4THENEU="ZEHN"
1730 FORA=1TO20
1740 IFEC(A)=""THEN1760
1750 IFRIGHT$(EC(A),LEN(EC(A))-3)=EUANDLEFT$(EC(A),2)=EQTHEN1780ELSEIFRIGHT$(EC(A),LEN(EC(A))-3)=EUANDEU=ELTHEN1780
1760 NEXTA
1770 IFAG>1THENAG=AG-1:GOTO1720ELSE1840
1780 ER=EC(A):EQ=LEFT$(EC(A),2):EC(A)=""
1790 FORB=1TO32
1800 IFEB(B)=""THENEB(B)=ER:GOTO1820
1810 NEXTB
1820 PRINTTAB(3)"Ich lege ";ER:GOSUB3840:BV=0:RETURN
1830 '#------------- Besonderes Suchen --------------------#
1840 AC=0
1850 FORA=1TO20
1860 IFEC(A)=""THEN1890
```

```
1870 AC=AC+1
1880 IFRIGHT$(EC(A),LEN(EC(A))-3)="BUBE"THEN1920
1890 NEXTA
1900 IFAC>0THEN2170ELSERETURN
1910 BV=0
1920 PRINTTAB(3)"Ich lege den ";EC(A):ER=EC(A):EC(A)="":AC=0
1930 FORA=1TO32
1940 IFEB(A)=""THENEB(A)=ER:GOTO1960
1950 NEXTA
1960 BA=0:BZ=0:BK=0:BR=0
1970 FORA=1TO20
1980 IFEC(A)=""THEN2020
1990 AC=AC+1
2000 IFRIGHT$(EC(A),LEN(EC(A))-3)="BUBE"THEN2020
2010 IFLEFT$(EC(A),2)="KA"THENBA=BA+1ELSEIFLEFT$(EC(A),2)="HZ"THENBZ=BZ+1ELSEIFLEFT$(EC(A),2)="PK"THENBK=BK+1ELSEIFLEFT$(EC(A)
,2)="KR"THENBR=BR+1
2020 NEXTA
2030 IFBR=0ANDBK=0ANDBA=0ANDBZ=0THEN2060
2040 IFAC=0THEN2060
2050 IFAC<2ANDRND(3)<>2THEN2100ELSEIFAC>1THEN2100
2060 CR=RND(4)
2070 IFCR=1THENEI="KA"ELSEIFCR=2THENEI="HZ"ELSEIFCR=3THENEI="PK"ELSEIFCR=4THENEI="KR"
2080 IFEI=EQTHEN2060
2090 GOTO2150
2100 D=0
2110 IFBA>DTHEND=BA:EI="KA"
2120 IFBZ>DTHEND=BZ:EI="HZ"
2130 IFBK>DTHEND=BK:EI="PK"
2140 IFBR>DTHEND=BR:EI="KR"
2150 PRINTTAB(3)"und ich wünsche mir die Farbe ";EI;:GOSUB3840:EQ=EI:RETURN
2160 '#--------------- Computer zieht eine Karte -------------#
2170 IFBV=1THENPRINTTAB(3)"Ich kann nicht":GOSUB3840:GOSUB3690:GOSUB3840:RETURN
2180 PRINTTAB(3)"Ich ziehe eine Karte ...":GOSUB3840
2190 B=RND(32):IFEA(B)=""THEN2190
2200 FORA=1TO20
2210 IFEC(A)=""THENEC(A)=EA(B):EA(B)="":BV=1:GOSUB1270:RETURN
2220 NEXTA
2230 '#---------------- Endroutinen ---------------------------#
2240 AF=0
2250 FORA=1TO20
2260 IFEC(A)<>""THEN2290
2270 NEXTA
2280 GOTO3020
2290 FORA=1TO20
2300 IFED(A)<>""THEN2330
2310 NEXTA
2320 GOTO2790
2330 FORA=2TO32
2340 IFEB(A)=EB(A-1)THENEB(A)=""
2350 NEXTA:AI=0
2360 FORA=1TO32
2370 IFEA(A)<>""THENAI=AI+1
2380 NEXTA
2390 IFAI>3THEN2490
2400 PRINTTAB(3)"Ich drehe den Spieß um ...":GOSUB3840
2410 FORA=1TO32:IFEA(A)=""THEN2440
2420 FORB=1TO32:IFEB(B)=""THENEB(B)=EA(A):EA(A)="":GOTO2440ELSE2430
2430 NEXTB
2440 NEXTA
2450 FORA=1TO32
2460 IFEB(A)=ERTHEN2480
2470 EA(A)=EB(A):EB(A)=""
2480 NEXTA
2490 N=0
2500 FORA=1TO20
```



```
2510 IFEC(A)<>""THENN=N+1
2520 NEXTA
2530 GOTO0350
2540 '#------------ Spieler zieht eine Karte ---------------#
2550 IFAF=0ANDRND(2)=1THENPRINTTAB(3)"Ich habe bessere Karten...!":GOTO2570ELSEIFAF=0THENPRINTTAB(3)"Tja, gegen mich hast Du k
eine Chance !":GOTO2570
2560 IFAF=1THENAF=0
2570 A=RND(4):IFA=1THENPRINTTAB(3)"Tja, erst denken, dann legen !":GOTO2600ELSEIFA=3THENPRINTTAB(3)"Jetzt sind Dir die ";EQ" K
arten wohl ausgegangen, wie?":GOTO2600ELSEIFA=4THENPRINTTAB(3)"Hätte mich auch gewundert, wenn das so weitergegangen wäre !":G
OTO260
2580 IFN<4THENPRINTTAB(3)"Ich würde mich etwas beeilen, ich habe nur noch";N;"Karten auf der Hand":GOTO2600
2590 PRINTTAB(3)"Naja, Du hast ja noch etwas Zeit"
2600 GOSUB3840
2610 B=RND(32):IFEA(B)=""THEN2610
2620 FORA=1TO20
2630 IFED(A)=""THENED(A)=EA(B):EA(B)="":PRINTTAB(3)"Du hast ";ED(A);" gezogen":GOSUB3840:RETURN
2640 NEXTA
2650 '#-------------------- Löschroutine -------------------#
2660 ER=EF
2670 FORA=1TO20
2680 IFED(A)=""THEN2700
2690 IFED(A)=EFTHENED(A)="":GOTO2710
2700 NEXTA
2710 FORA=1TO32
2720 IFEB(A)=""THENEB(A)=EF:RETURN
2730 NEXTA
2740 AE=0:FORA=1TO20
2750 IFED(A)<>""THENAE=AE+1
2760 NEXTA
2770 IFAE<2THENGOTO2790
2780 RETURN
2790 D=0:FORA=1TO20
2800 IFEC(A)=""THEN2870
2810 ET=RIGHT$(EC(A),LEN(EC(A))-3)
2820 IFET="ZEHN"THEND=D+10
2830 IFET="BUBE"THEND=D+2
2840 IFET="DAME"THEND=D+3
2850 IFET="KÖNIG"THEND=D+4
2860 IFET="AS"THEND=D+11
2870 NEXTA
2880 GOSUB3760:PRINT§425,"I C H  G R A T U L I E R E  !!";:PRINT§584,"Du hast mit";D;"Punkten gewonnen."
2890 PRINT§811,"";:IFD=0THENPRINT§725,"Mit 0 Punkten zu gewinnen ist aber etwas schwachsinnig ...";:GOTO2980
2900 IFD<=4THENPRINT§730,"Das war aber ganz schön knapp ...";:GOTO2980
2910 IFD<=10THENPRINT§730,"Du hast recht gut gespielt.";:GOTO2980
2920 IFD<=15THENPRINT§740,"Du warst gut !;":GOTO2980
2930 IFD<=20THENPRINT§725,"Prima gespielt! Ich bin direkt neidisch.";:GOTO2980
2940 IFD<=25THENPRINT§725,"Du hast ganz hervorragend gespielt !!";:GOTO2980
2950 IFD<=30THENPRINT§723,"Ich habe hoch verloren.Ich glaube ich habe einen sehr schwachen Tag gehabt.";:GOTO2980
2960 IFD>30THENPRINT"Du bist Spitze,die Europameisterschaft ist Dir sicher"
2970 '#------------------ Spielende -----------------------#
2980 GOSUB3850:PRINT§1070,"Neues Spiel  < J/N >";
2990 EA=INKEY$:IFEA=""THEN2990
3000 IFEA="N"OREA="n"THEN3010ELSERUN
3010 FORDX=1TO5:CLS:PRINT§340,"Auf Wiedersehen .....":GOSUB3840:CLS:FORDQ=1TO500:NEXTDQ:NEXTDX:RUN"SPIELMSA/BAS"
3020 D=0:FORA=1TO20
3030 IFED(A)=""THEN3100
3040 ET=RIGHT$(ED(A),LEN(ED(A))-3)
3050 IFET="ZEHN"THEND=D+10
3060 IFET="BUBE"THEND=D+2
3070 IFET="DAME"THEND=D+3
3080 IFET="KÖNIG"THEND=D+4
3090 IFET="AS"THEND=D+11
3100 NEXTA
3110 GOSUB3760:PRINT§403,"HAHA, wie nicht anders zu erwarten, habe ich Dich mit";D;"Augen geschlagen.";
3120 IFD>=30PRINT§563,"Tja, ich bin ja auch von Willi Johnen programmiert worden ... ,da hast Du na-";:PRINT§643,"türlich kein
e Chance !!";:GOTO3180
```

```
3130 IFD)=20PRINT§563,"Bin schon ein toller Hecht, nicht wahr ...?";:GOTO3180
3140 IFD)=15PRINT§563,"Wenn Du noch übst, klappt es vielleicht nochmal ...!";:GOTO3180
3150 IFD)=10PRINT§563,"Für einen blutigen Anfänger warst Du garnicht schlecht ...!";:GOTO3180
3160 IFD)=5THENPRINT§563,"Na ja, so schlecht warst Du garnicht";:GOTO3180
3170 IFD)=0THENPRINT§563,"Fairerweise muß ich Dir sagen,daß die Entscheidung recht knapp war ...!";
3180 GOSUB3840
3190 PRINT:A0=RND(5):IFA0=1THENEA="uninteressantes"ELSEIFA0=2THENEA="beknacktes"ELSEIFA0=3THENEA="stinklangweiliges"ELSEIFA0=4
THENEA="frustierendes"ELSEEA="blödes"
3200 PRINT§803,"Ich ringe noch mit mir,ob ich noch so ein ";EA;" Spiel mit Dir machen soll ...":GOSUB3840
3210 IFRND(3)(3THENPRINT§963,"Na gut, ich bin ja nicht so ... , obwohl Du es ja eigentlich nicht verdient";:PRINT§1043,"hast .
.. , doch ich will Dir eine Chance geben !":GOSUB3840:GOSUB3760:RUN
3220 PRINT§963,"Also, wenn ich ehrlich bin, habe ich keinen Bock mehr und bevor ich mich hier";:PRINT§1043,"zu Tode langweile,
mache ich lieber den Laden dicht ...";:GOSUB3840:PRINT§1240,"M A C H   E S   G U T ...";:GOSUB3840:CLS:CLEAR10:RUN"SPIELMSA
/BAS"
3230 '#---------------------- UP -    ------------------------#
3240 EP=RIGHT$(ER,LEN(ER)-3)
3250 AH=0:FORA=1TO20:IFEC(A)=""THEN3270
3260 IFRIGHT$(EC(A),LEN(EC(A))-3)=EPTHENAH=AH+1
3270 NEXTA
3280 IFAH=0THEN3540
3290 FORA=1TO20:IFEC(A)=""THEN3360
3300 IFRIGHT$(EC(A),LEN(EC(A))-3)()EPTHEN3360
3310 EV=LEFT$(EC(A),2)
3320 FORB=1TO20:IFEC(B)=""THEN3350
3330 IFRIGHT$(EC(B),LEN(EC(B))-3)()"SIEBEN"ORLEFT$(EC(B),2)()EVTHEN3350
3340 GOTO3370
3350 NEXTB
3360 NEXTA:GOTO3540
3370 FORA=1TO20:IFEC(A)=""THEN3440
3380 IFRIGHT$(EC(A),LEN(EC(A))-3)()EPTHEN3440
3390 IFLEFT$(EC(A),2)=EVTHEN3440
3400 ER=EC(A):EQ=LEFT$(ER,2):EC(A)=""
3410 FORB=1TO32:IFEB(A)=""THENEB(A)=ER:GOTO3430
3420 NEXTB
3430 PRINTTAB(3)"Ich lege";ER
3440 NEXTA
3450 FORA=1TO20:IFEC(A)=""THEN3530
3460 IFRIGHT$(EC(A),LEN(EC(A))-3)()EPTHEN3530
3470 ER=EC(A):EQ=LEFT$(ER,2):EC(A)=""
3480 FORB=1TO32:IFEB(B)=""THENEB(B)=ER:GOTO3500
3490 NEXTB
3500 PRINTTAB(3)"Ich lege ";ER
3510 FORA=1TO20:IFEC(A)=""THEN3530
3520 IFRIGHT$(EC(A),LEN(EC(A))-3)="SIEBEN"ANDLEFT$(EC(A),2)=EVTHENGOSUB1460:CT=1:RETURN
3530 NEXTA
3540 FORA=1TO20:IFEC(A)=""THEN3580
3550 IFRIGHT$(EC(A),LEN(EC(A))-3)()"SIEBEN"THEN3580
3560 IFLEFT$(EC(A),2)()EQTHEN3580
3570 GOSUB1460:CT=1:RETURN
3580 NEXTA:CT=0:RETURN
3590 '#------------------ Datafils ---------------------------#
3600 DATA KA SIEBEN,KA ACHT,KA NEUN,KA ZEHN,KA BUBE,KA DAME,KA KÖNIG,KA AS,HZ SIEBEN,HZ ACHT,HZ NEUN,HZ ZEHN,HZ BUBE,HZ DAME,H
Z KÖNIG,HZ AS,PK SIEBEN,PK ACHT,PK NEUN,PK ZEHN,PK BUBE,PK DAME,PK KÖNIG,PK AS
3610 DATA KR SIEBEN,KR ACHT,KR NEUN,KR ZEHN,KR BUBE,KR DAME,KR KÖNIG,KR AS
3620 '#------------------- Bemerkungen -----------------------#
3630 PRINTTAB(3)"Du altes Arsch....und das jetzt!";:GOTO3680
3640 PRINTTAB(3)"Verdammter Mist !";:GOTO3680
3650 PRINTTAB(3)"Na warte,das zahle ich Dir heim";:GOTO3680
3660 PRINTTAB(3)"Ach Du Scheibenkleister !";:GOTO3680
3670 PRINTTAB(3)"Das findest Du wohl witzig, was ??";:GOTO3680
3680 GOSUB3840:PRINT§801," ";:RETURN
3690 A0=RND(6):IFA0=1THENPRINT"So ein Ärger !"
3700 IFA0=2THENPRINT"Verdammter Mist und das gerade jetzt ...!"
3710 IFA0=3THENPRINT"Himmel, Arsch und Zwirn !!"
3720 IFA0=4THENPRINT"Lach nicht, das kann Dir auch passieren !!"
```



```
3730 IFA0=5THENPRINT"Da kann man doch aus der Haut fahren !!"
3740 IFA0=6THEN PRINT"Jetzt reicht es mir aber ...!":RETURN
3750 '#------------ UP - Startbild ------------------------#
3760 CLS:HCLS:'HLOAD"BLOCK/RZS"
3770 HBIG0,242,1,1,0,"MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU":HBIG0,234,1,1,0,"-":HBIG627,234,1,1,0,"-":HBIG0,226,1,1
,0,"M":HBIG627,226,1,1,0,"M":HBIG0,218,1,1,0,"A":HBIG627,218,1,1,0,"A":HBIG0,210,1,1,0,"U":HBIG627,210,1,1,0,"U":HBIG0,202,1,1
,0,"-"
3780 HBIG627,202,1,1,0,"-":HBIG0,194,1,1,0,"M":HBIG627,194,1,1,0,"M":HBIG0,186,1,1,0,"A":HBIG627,186,1,1,0,"A":HBIG0,178,1,1,0
,"U":HBIG627,178,1,1,0,"U":HBIG0,170,1,1,0,"-":HBIG627,170,1,1,0,"-":HBIG0,162,1,1,0,"M":HBIG627,162,1,1,0,"M":HBIG0,154,1,1,0
,"U"
3790 HBIG627,154,1,1,0,"U":HBIG0,146,1,1,0,"-":HBIG627,146,1,1,0,"-":HBIG0,138,1,1,0,"M":HBIG627,138,1,1,0,"M":HBIG0,130,1,1,0
,"A":HBIG627,130,1,1,0,"A":HBIG0,122,1,1,0,"U":HBIG627,122,1,1,0,"U":HBIG0,114,1,1,0,"-":HBIG627,114,1,1,0,"-":HBIG0,106,1,1,0
,"M"
3800 HBIG627,106,1,1,0,"M":HBIG0,98,1,1,0,"A":HBIG627,98,1,1,0,"A":HBIG0,90,1,1,0,"U":HBIG627,90,1,1,0,"U":HBIG0,82,1,1,0,"-":
HBIG627,82,1,1,0,"-":HBIG0,74,1,1,0,"M":HBIG627,74,1,1,0,"M":HBIG0,66,1,1,0,"A":HBIG627,66,1,1,0,"A":HBIG0,58,1,1,0,"U"
3810 HBIG627,58,1,1,0,"U":HBIG0,50,1,1,0,"-":HBIG627,50,1,1,0,"-":HBIG0,42,1,1,0,"M":HBIG627,42,1,1,0,"M":HBIG0,34,1,1,0,"A":H
BIG627,34,1,1,0,"A":HBIG0,26,1,1,0,"-":HBIG627,26,1,1,0,"-":HBIG0,18,1,1,0,"M":HBIG627,18,1,1,0,"M":HBIG0,10,1,1,0,"A"
3820 HBIG627,10,1,1,0,"A":HBIG0,2,1,1,0,"U-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-MAU-M":RETURN
3830 '#------------ UP - Bildschirmroutinen ----------------#
3840 FORAB=1TO1000:NEXTAB:RETURN
3850 PRINT§242,FA;:RETURN
3860 PRINT§162,FB;:RETURN
3870 PRINT§1842,FB;:RETURN
3880 PRINT§802,FB;:PRINT§882,FB;:PRINT§962,FB;:PRINT§1042,FB;:RETURN
3890 '#----------- Neustart --------------------------------#
3900 PRINT§825,"D U   S C H E R Z B O L D !";:GOSUB3840:GOSUB3760:GOTO220
```

"Bad News for IBM and Apple"

Mit dieser Schreckensmeldung für die Konkurrenz erfolgte 1985 die Vorstellung des Commodore 128 PC. Der weitere Text der Anzeige (übersetzt und etwas gekürzt):

Der Commodore 128 PC verfügt über den gleichen Speicher wie der IBM PCjr. und Apple IIc.
Er hat die Möglichkeit der 80-Zeichen Farbdarstellung, der IBM-PCjr. nicht.
Der Speicher ist auf 512 K erweiterbar, der des Apple IIc nicht.
Er ist der einzige Personal-Computer, für den es schon vor seiner Einführung mehr als 6.000 Programme gibt, weil er zum C 64 kompatibel ist."

Die Euphorie begründete sich auf den Einbau einer zusätzlichen Z80 CPU, mit der CP/M möglich wurde.

Den C128 gab es in drei Ausführungen:

In der ältesten ähnelte der C128 äußerlich dem Amiga, er verfügte über einen separaten 10er Tastenblock. Der gesamte Rechner war im gleichen Gehäuse wie die Tastatur untergebracht. Ein Diskettenlaufwerk mußte extra angeschlossen werden.

Das Nachfolgemodell sah bereits wie ein PC aus; in einem Kunststoffgehäuse befand sich der Rechner und ein 5,25" Diskettenlaufwerk mit 340 K. Das Gehäuse hatte einen Griff, die Tastatur konnte in einer Aussparung des Gehäuses untergebracht werden. Das Gerät war also transportabel. Allerdings war ein Netzanschluß und ein separater Monitor notwendig. Damit wurden der Mobilität Grenzen gesetzt.

In der letzten Ausführung wurde das Gehäuse aus Blech hergestellt. Der Tragegriff und die Einbuchtung für die Tastatur entfielen.

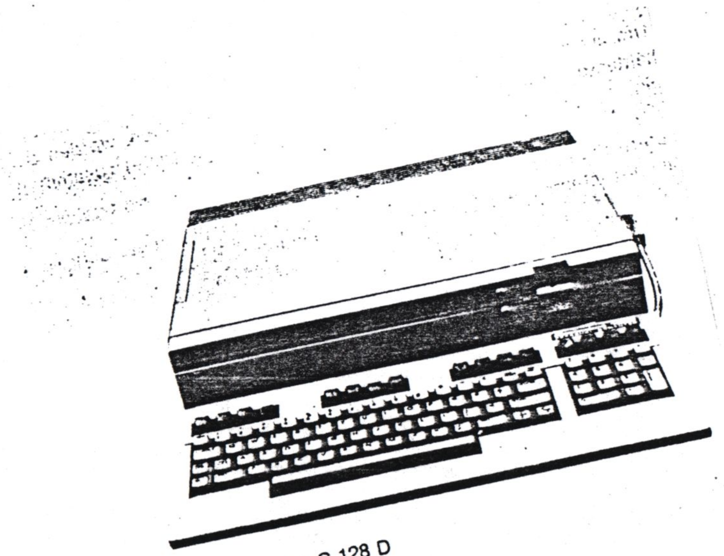


Abb. 3.1: Tastatur des C 128 D

Anschlüsse:

RGB-Ausgang

Anschlußkompatibel zu IBM Schwarzweiß und Farbmonitoren
80-Zeichen Darstellung, 8 Farben in je zwei Intensitätsstufen

Datasettenanschluß

Anschluß eines Commodore-Datenrecorders

Serieller Bus

Anschluß von Diskettenlaufwerken, Druckern und Plottern.
Der serielle Bus ist bei den Floppies durchgeschleift. Wenn also mehrere Floppies angeschlossen werden sollen, wird die nächste einfach an einen entsprechenden Anschluß der Vorfloppy eingesteckt. Der Drucker ist in der Regel dann das letzte Gerät. Jedes Gerät hat eine Adresse (z.B. Floppies 8 - 11, Drucker 4 und 5). Die Datenübertragung an die seriell angeschlossenen Geräte erfolgt unter Angabe der Gerätenummer und gelangt dann an die richtige Adresse. Man könnte fast schon von einer frühen Form des SCSI - Busses sprechen.

Videoausgang

Zum Anschluß eines Monitors oder Farbfernsehgeräts das ein FBAS-Signal verarbeiten kann. Unter CP/M ist hier nur 40-Zeichendarstellung möglich. Nach Erreichen der rechten Bildschirmbegrenzung scrollt das Bild dann nach links. Es sind also auch 80-Zeichen pro Zeile möglich, allerdings nicht gleichzeitig darstellbar.

HF - Antennenausgang

Dient zum Anschluß eines (Farb-) Fernsehers als Monitorersatz. Auch hier ist nur 40-Zeichendarstellung möglich.

Audio-Eingang

Externe Audio Signale können mit Hilfe des eingebauten Basic verarbeitet werden.

VOL	Lautstärke
ENVELOPE	Definition von Tonhüllkurven zur Nachahmung verschiedener Instrumente
FILTER	Zum Ausfiltern verschiedener Frequenzen
TEMPO	Bestimmt das Spieltempo

Abb. 3.1: Tastatur des C 128

User-Port

Eine bidirektionale, programmierbare 8-Bit-Schnittstelle mit zwei Handshakeleitungen zur Steuerung des Ein/Ausgabeprozesses. Hier kann sowohl eine serielle RS-232-Schnittstelle, als auch eine parallele Schnittstelle zum Anschluß eines Druckers konfiguriert werden. Leider stimmt der Signalpegel für die RS-232 nicht mit der Norm überein. Es gibt Pegelwandler, die den Betrieb dann problemlos ermöglichen. CP/M unterstützt den Userport als parallele Schnittstelle zum Anschluß eines Druckers.

Expansion-Port

Es stehen sämtliche Daten- und Adreßleitungen, sowie einige Steuerleitungen zur Verfügung. Der komplette Systembus ist herausgeführt. Anschluß von Programm- und Speichererweiterungsmodulen oder Interfaces für die Meß- und Regeltechnik.

Joystickports

Für zwei Joysticks oder vier Paddles, außerdem können Maus und Lightpen angeschlossen werden.

CPUs

8502

kompatibel zur 6502 des 64ers, wird im C64 und C128 Modus eingesetzt. Taktfrequenz 1 MHz im C64-Modus und 2 MHz im C128 Modus. Die Taktfrequenz läßt sich mit Hilfe eines Basic-Befehls (SLOW/FAST) umschalten. Das ist notwendig, weil mit 2MHz keine Ausgabe des 40-Zeichen/Grafik Bildschirms möglich ist.

Z80

Übernimmt die Steuerung des Rechners wenn beim Einschalten die CP/M-Startdiskette in die Floppy eingelegt ist. Taktfrequenz 4 MHz, bei Peripheriezugriffen nur 2MHz, weil die gleiche Peripherie benutzt wird, auf die auch der 8502 zugreift. Unter CP/M ergibt sich damit eine durchschnittliche Verarbeitungsgeschwindigkeit von 2,5 MHz.

Zubehör:

Diskettenlaufwerk

5,25" mit 170 K (einseitig)	Typ 1541 GCR
5,25" mit 340 K (zweiseitig)	Typ 1571 MFM/GCR
3,5 " mit 800 K (zweiseitig)	Typ 1581 MFM

Die Floppies verfügen alle über eine eigene CPU, eigenes RAM und ein eigenes Betriebssystem, sind also "intelligent".

RAM-Erweiterungen mit 128, 256 und 512 K (aufrüstbar bis 2MB)

werden Expansionsport eingesteckt und sind dann unter Basic ansprechbar. Große Bedeutung unter GEOS und CP/M als zusätzliche Laufwerke.

Scanner

eine damals relativ preiswerte Lösung war der Aufsatz des Scanners (mit 4 Graustufen) auf den Druckkopf des Druckers. Die Scanvorlage wurde dann in den Drucker eingespannt und Zeile für Zeile abgetastet.

BTX-Modul

Wurde von Fa. Siemens für den C64 hergestellt. Mit diesem Modul war BTX schon sehr früh und mit einer ausgezeichneten Darstellung möglich. Das Modul bot einige Zusatzfunktionen wie:

- Speichern von BTX-Seiten im RAM (20K) des Moduls oder auf Diskette
- Anlegen und Ausführung von Makrodateien
- ASCII-Terminal-Modus für Datenverkehr mit Datex-P
- Laden von Telesoftware

Heute werden allerdings überwiegend Software-Decoder verwendet.

SCSI Festplatten

relativ neu entwickelt

Laufwerke mit 1,6 und 3,2 MB

relativ neu entwickelt

Betriebssysteme

C 64 - Modus (Speichergröße 64 K)

Ist vollständig C 64 kompatibel. Basic V 2.0 ist im ROM vorhanden und steht sofort nach dem Einschalten zur Verfügung. Hier laufen alle Programme, die für den C 64 geschrieben wurden. Übrigens gab es schon für den C 64 ein Steckmodul für den Expansionsport, mit dem CP/M 2.2 zur Verfügung stand. Sogar eine 80-Zeichen-Karte war erhältlich.

C 128 - Modus (Speichergröße 128 K)

Hier steht Basic V 7.0 nach dem Einschalten zur Verfügung. Für die Speichererweiterung stehen zusätzliche Befehle bereit, mit denen Inhalte der Speicherbänke (bis zu 640 K mit RAM-Erweiterung) ausgelesen, eingelesen oder ausgetauscht werden können. Es ist sogar der Betrieb zweier Bildschirme (40-Zeichen/Grafik und 80-Zeichen) gleichzeitig möglich.
Für den C128- Modus gab es schon professionelle Programme (Textverarbeitung, Datenbank, Finanzbuchhaltung, Anwendungen für Gewerbetreibende), allerdings schlief die Weiterentwicklung dann doch mangels Nachfrage und der Konkurrenz der 16-Bit Rechner ein.

GEOS (Graphic Environment Operating System)

wird von Diskette geladen und bietet eine, mit der Maus zu bedienende, grafische Benutzeroberfläche, inzwischen wird das BS auch auf EPROM ausgeliefert (freier Steckplatz ist im Rechner vorhanden) und kann dann mit Tastendruck sofort gestartet werden. Unter GEOS gibt es Hunderte von Anwendungen (Textverarbeitung, Datenverwaltung, Tabellenkalkulation, Grafik, DTP, DFÜ), deren Daten einfach untereinander auszutauschen sind. Ähnlich wie unter Windows erfolgt auch der Ausdruck der mit GEOS erstellten Dokumente als Grafik. Hier lassen sich also alle möglichen Schriftarten verwenden. GEOS gibt es bereits seit 1983 für den C 64 und wird immer noch weiterentwickelt, es gibt einen GEOS-User-Club und mehrere Regionalclubs.
Auch für den PC ist GEOS verfügbar, hier nennt es sich Geoworks.

CP/M (59 K TPA)

CP/M - Entwickler für den C 128 war Von Ertwine. Es wurde CP/M + (CP/M 3.0) eingesetzt. CP/M+ kann im Gegensatz zu CP/M 2.2 mehr als 64 KByte Speicher verwalten. Erreicht wird dies durch sogenanntes "Bankswitching", dem Umschalten zwischen verschiedenen Speicherbänken. Dabei befindet sich ein Teil des Arbeitsspeichers, der Common-Bereich, immer im logischen Adreßfeld der CPU, während die jeweiligen Bänke nach Bedarf wechselweise zugeschaltet werden.
Für neu hinzugekommene Geräte wurden die Versionen immer wieder erneuert. In der letzten Version werden alle Laufwerke (5,25" und 3,5", sowie die 512 K RAM-Erweiterung als Laufwerk M:) unterstützt. Mit dem Laufwerk M: ist dann ein einigermaßen schnelles Arbeiten möglich. Wenn das Anwendungsprogramm in diesen Bereich geladen wird, geht das Nachladen einzelner Programmteile blitzschnell. Unter CP/M können die Laufwerke IBM, Osborne und Kaypro-Disketten lesen, beschreiben und formatieren.
Nachdem Markt und Technik die Programme Wordstar 3.0, dBase II V 2.41, Multiplan V 1.06, MS-Basic und Turbo-Pascal 3.0 auch für den C 128 verkaufte, war der leider nur kurze Weg zum Profi-Computer frei.

Das 5,25" (Typ 1571) Diskettenlaufwerk kann unter CP/M von Haus aus folgende Diskettenformate verarbeiten:
Epson QX10 IBM-8 Kaypro II KayPro IV Osborne

Falls eine dieser Disketten verwendet wird, überprüft das System, um welches Format es sich handelt. Das Format wird dann in einer Box in der linken, unteren Bildschirmecke angezeigt. Mit Hilfe des Zusatzprogramms JUGGLER.COM können zusätzlich drei fremde Diskformate im BIOS installiert werden. Die Bearbeitung dieser Formate ist dann ebenfalls problemlos möglich. JUGGLER bietet eine Auswahl von etwa 200 verschiedenen Formaten.

Unter CP/M wird ein ADM31-Terminal emuliert. Der C 128 bietet acht Bildschirmfarben in jeweils zwei Helligkeitsstufen. Die Darstellung kann, getrennt nach Hintergrund- und Zeichenfarbe, jederzeit durch Tastendruck geändert werden und gilt dann bis zum Neustarten des Systems. Darüberhinaus können nahezu sämtliche Tasten mit

- Zeichenfolgen
- Einzelzeichen
- Farbzuordnungen
- Hex-Werten und
- Sonderfunktionen

belegt werden. Diese Einstellungen können für die Einschaltdauer des Computers aktiviert werden oder permanent in der Datei CP/M+.SYS abgespeichert werden, womit sie dann nach jedem Booten zur Verfügung stehen.

Hier eine Auflistung der beim C 128 vorhandenen Befehle, soweit die Befehle unter CP/M 2.2 auch vorhanden sind, habe ich auf eine Erläuterung verzichtet. Allerdings ist mir der Befehlsumfang von CP/M 2.2 nicht genau bekannt.

CCP.COM
CPM+.SYS

COPYSYS.COM
hat beim C128 keine Bedeutung. Zur Übertragung der Systemdateien wird PIP verwendet.

DATE.COM
Datum und Uhrzeit stellen und abfragen

DEVICE:COM
dient dazu, die momentane Zuordnung von physikalischen zu logischen Geräten zu zeigen oder zu ändern. Außerdem lassen sich die Parameter der seriellen Schnittstelle (RS-232) bestimmen und die Bildschirmgröße festsetzen

DIR.COM
DUMP.COM
ED.COM
ERASE.COM

FORMAT.COM
formatiert eine Diskette

GENCOM.COM
erzeugt RSX-Erweiterungen in COM.Dateien

GET.COM
liest Eingaben aus Datei anstatt von der Konsole

HELP.COM
zum Abruf von Informationen über die gebräuchlichsten Befehl und Kommandodateien. Die Informationen sind in der Datei HELP.HLP abgelegt und lassen sich ändern oder ergänzen

HEXCOM.COM
erzeugt aus INTEL-Hex-File eine COM.Datei

INITDIR.COM
bereitet Directory für die Aufnahme von Timestamps vor

KEYFIG.COM
individuelle Tastaturbelegung

LIB.COM
erstellen und editieren von Programmbibliotheken

LINK.COM
Verbindung und Umwandlung von Modulen in COM-Datei

MAC.COM
8080-Macro-Assembler

PATCH.COM
Änderung von Betriebssystem und COM-Dateien

PIP.COM

PROFILE.SUB
Submit-Datei, wird automatisch nach dem Booten ausgeführt

PUT.COM
Umleitung von Drucker/Bildschirmausgabe in Datei

RENAME.COM

RMAC.COM
Macroassembler für ASM-Quelldateien

SAVE.COM

SET.COM
Setzen von Labels, Timestamps, Passwort, RO,

SETDEF.COM
Angabe eines Suchpfades für SUB und COM Dateien

SETUP.COM
Auswahl von Zeichensatz (ASCII/German) und Drucker (parallel, seriell)

SHOW.COM
zum Abruf von Informationen über die angeschlossenen Laufwerke und Diskettencharakteristika

SID.COM
Debugger zum Laden, Ändern und Testen von Programmen

SUBMIT.COM
arbeitet Kommandos aus einer Submit-Datei automatisch ab. Wenn eine Datei namens PROFILE.SUB vorhanden ist, wird diese Datei sofort nach Systemstart automatisch ausgeführt (ähnlich AUTOEXEC.BAT bei MS-DOS)

TYPE.COM

USER
dient zum Umschalten auf einen der 16 möglichen Userbereiche. Mit Hilfe von USER können auf einer Diskette bis zu 16 verschiedene Unterverzeichnisse angelegt werden, die allerdings nicht verschachtelt werden können

XREF.COM
Erzeugung einer Cross-Reference-Datei

Soweit zur Vorstellung des Commodore 128, den ich aufgrund seiner C64-Kompatibilität für eine gelungene Maschine halte. Ich würde mich freuen, wenn auch einmal die älteren Rechner (mit Bild) beschrieben würden. Als Neuling kenne ich die nur vom Lesen.

Georg Bielenberg

Wie die verschiedenen Programmierer ihre Fahrraeder bauen:

ADA-Programmierer bauen ein viereckiges Rad und passen alle Strassen an.

ALGOL-Programmierer weigern sich, Raeder zu bauen, weil es sie irgendwohin bringen koennte.

APL-Programmierer schweben in hoeheren Sphaeren, sie brauchen keine Raeder.

ASSEMBLER-Programmierer bauen tausende von Raedern, keine 2 passen auf eine Achse.

BASIC-Programmierer bauen nur ein Rad, aber finden keine Achse dazu.

COBOL-Programmierer bauen TAUSEND-RAEDRIGE-TRANSPORT-MODULE und verbieten das Gehen.

FUNKTIONAL-Programmierer rufen eine Funktion HOLZ auf und hoffen, ein Rad zu bekommen.

FORTH-Programmierer bauen Raeder und vergessen, wo sie sie gestapelt haben.

FORTRAN-Programmierer werden wahnsinnig bei der Suche nach Raedern, die mit "I" beginnen.

LOGO-Programmierer bauen kleine rote Autos.

LISP-Programmierer (bauen Raeder mit (Raedern mit/Raedern mit (Raedern mit (dem was LISP-Programmierer bauen))))).

MICRO-Programmierer wissen nicht, dass Raeder existieren.

PASCAL-Programmierer erklaeren das Gehen zur Tugend.

PL/I-Programmierer setzen ein Team ein, um eine Raederfabrik zu entwerfen, die Raeder der falschen Groesse bauen wird.

RPG-Programmierer haben ein Rad, schade das es viereckig ist.

SYSTEM-Analytiker sind viel zu beschaeftigt, Raeder zu suchen, um eins zu bauen.

TEXTVERARBEITUNGS-Benutzer bauen Raeder.

TOS-Benutzer bleiben wo sie sind, sie sind das Warten gewohnt.

UNIX hat unter irgendeiner Schale sicher irgendwo ein Verzeichnis von Raedern.

Rantanplan's gesammelte Werke der TAE-Anschlusstechnik

Wer hat zu Hause nicht schon oft mit krampfverzerrtem Gesicht vor seiner neuen TAE-Dose gesessen und den P*stminister verflucht, weil der Netzabschluss der Telekom Geheimnisse verbirgt, die gar arglistig und nervzerreibend sind. Dank diesem Werk gehoeren solche Beinahe-Herzattaken der Vergangenheit an. Nicht genug, dass man nun eine Dreier-TAE-Dose hat, aber wie zum Geier schliesse ich denn jetzt den NON-FTZ-Anrufbeantworter, das US-Robotics-Modem und das amerikanische Einhandtelefon denn nun an? Alles kein Problem, weiter unten wird alles erklaert.

Wer keine TAE-Dosen hat, aber auf die neue Technik umsteigen will, sollte sich immer nur TAE-Dosen der Kodierung NFF besorgen. Die kann man am mueheloseten beschalten. Auch wenn die P*st kommt und eine fuer den neuen Telefonanschluss setzten muss, unbedingt drauf bestehen, dass eine NFF-Dose gesetzt wird !!! Die Post versucht immer die NFN kodierte Dose an den Mann/die Frau zu bringen.

Wer keine TAE-Stecker hat, kann diese in den oertlichen Elektrobastelladen kaufen, die Kodierung ist in diesem Fall egal, wir DECODIEREN sowieso alles. Natuerlich koennen auch fertige Schnuere gekauft werden, aber Achtung, keine verschweissten. Das hemmt die Loetwut.

Und nun zur Theorie:

TAE-(Telekommunikationsanschlusseinheit)-Dosen gibt es in verschiedenen Ausfuehrungen. Ihre Funktionen werden durch die Buchstaben -N- oder -F- unterschieden. -N- steht in diesem Fall fuer NICHTFERNSPRECHEN und -F- fuer FERNSPRECHEN.

```
-F- Kodiert werden alle TELEFONE der neuen Anschlusstechnik
-N- Kodiert werden alle TELEFAXE
                        ANRUFBEANTWORTER
                        BTX-BOXEN
                        POSTMODEMS der neuen Anschlusstechnik
```

Kodiert heisst: -F- Geraete lassen sich nur in -F- Dosen stecken, und -N-Geraete nur in -N- Dosen.

Frueher gab es auch noch die -D- kodierte Dose. Diese ist aber schon seit langem wieder weg vom Fenster, sie wurde durch die -N- Dose ersetzt.
Auch ISDN-Telefone werden mit TAE-Dosen betrieben, aber das sind Sonderformen mit mehreren Steckkontakten.

Die erste Aktion

Um der schrecklichen Kodierung ein Schnippchen zu schlagen, sollte man gleich ein scharfes Messer auspacken und alle TAE-Stecker decodieren.
Dazu wird die nach aussen weitest abstehenden Fuehrungsnasen (eine rechts, eine links) entfernt.
Damit kann man jetzt alle Dienste in allen Dosenarten betreiben, goil was !

Und nun zur Praxis:

Fangen wir mit der primitivsten TAE an:

```
                    TAE F-Kodiert:
                    --------------

        AMT
        REIN
        |  |         ^  ^
        V  V         |  |
        O  O  O  O  O  O         Das O stellt den Schraub-
        1  2  3  4  5  6         anschluss dar !!!

      _________________
      |       __      |          |
      |     F|  |     |          V  Eingang der Amtsleitung
      |      |  |     |
      |      |__|     |          ^
      |_______________|          |  Weiterfuehrung der Amtsltg
```

Die Amtsleitung wird an Punkt 1 und 2 angeklemmt. Es kann nur ein Telefon, oder nur ein Fax, oder nur ein Modem, oder nur ein Anrufbeantworter gesteckt werden. Wird das Telefon rausgezogen, wird die Amtsleitung auf Punkt 5 und 6 weitergeschaltet. Hat man nun weitere Telefondosen hintereinander geschaltet, wird immer wenn ein Telefon gesteckt wird, die naechste Dose totgelegt. Um dieses zu umgehen, einfach die Draehte die auf 5 und 6 liegen, auf 1 und 2 zur Amtsleitung dazuklemmen. Hat man noch eine Klingel, dann kann man die auf 2 und 3 anschliessen. Fuer Freax die an einer Nebenstellenanlage dranhaengen, ist noch Punkt 4 interessant. Da kommt der Erddraht dran, damit man ein Amt bekommt, oder Gespraeche weitervermitteln kann.

```
                    TAE FF-Kodiert:
                    ---------------

        AMT         AMT
        REIN        RAUS
        |  |         ^  ^
        V  V         |  |
        O  O  O  O  O  O         Das O stellt den Schraub-
        1  2  3  4  5  6         anschluss dar !!!

      _________________
      |  __      __   |          Das obere Anschlussfeld ist
      | F|  |   F|  | |          fuer den linken Steckschlitz
      |  |  |    |  | |          Das untere Anschlussfeld ist
      |  |__|    |__| |          fuer den rechten Steckschlitz
      |_______________|

        6  5  4  3  2  1
        O  O  O  O  O  O
                     ^  ^
                     |  |
                     AMT
                    WIEDER
                     REIN
```


Anschlussmoeglichkeit fuer maximal:

- zwei Amtsleitung / zwei Telefone
- eine Amtsleitung / ein Fax / ein Telefon
- eine Amtsleitung / ein Anrufbeantworter / ein Telefon
- eine Amtsleitung / ein Modem / ein Telefon

Anstatt eines Telefons, kann natuerlich auch jede andere Kombination von zwei Geraeten gewaehlt werden.

Anschluss MODEM / TELEFON:

Dazu muessen wir uns erst einmal den Stecker naeher betrachten:
Die Steckeransicht wird hier so dargestellt, als wenn man von hinten auf das Gehaeuse schaut. Das Steckelement ist also vorne.

```
              ----
      3       ][        4
      2       ][        6
      1       ][        5
              ----
```

Um nun ein Modem anzuschliessen, dass hier in Deutschland bei jeder FTZ-Pruefung durchfaellt, aber sonst ueberall in der Welt eingesetzt werden darf, ist es erforderlich die beiden Modemdraehte (zu 95% immer gruen und rot) an dem TAE-Stecker auf Punkt 1 und 2 anzuloeten. Damit funktioniert das Modem auf dem linken Steckschliz. Aber das Telefon soll ja auch noch angeklemmt werden. Um das zu realisieren, muss im TAE-Stecker des Modems zusaetzlich noch eine Bruecke von 1 nach 5 und 2 nach 6 geloetet werden. Dadurch wird die Amtsleitung beim Stecken des Modems weiter durchgeschaltet

An der TAE-Dose wird nun oben am Punkt 5 und 6 mit Drahtbruecken eine Verbindung auf die untere Seite der Dose auf Punkt 1 und 2 angeklemmt. Dadurch wird der rechte Steckschlitz der TAE-Dose mit der Amtsleitung verbunden. Dort wir das Telefon reingesteckt. Telefone mit TAE-Stecker funktionieren sofort. Hat man noch Telefone mit VDO-Stecker (Flachstecker) oder mit ADO-Stecker (weisser Klumpen), muss man sich einen TAE-Stecker loeten. Dazu werden die Draehte folgendermassen beschaltet:
1 - weiss
2 - braun
3 - gruen (nicht zwingend erforderlich)
4 - gelb (nicht zwingend erforderlich)

Bei amerikanischen und japanischen Telefonen gilt:
1 - rot
2 - gruen

Anschlusshinweise beim Stecken der Geraete:
Grundsaetzlich gilt: Alle Zusatzgeraete (Fax, Anrufbea, Modem) werden VOR das Telefon angeschlossen, ergo ist das Telefon immer das letzte Geraet und braucht daher nicht im Stecker durchgebrueckt zu werden (Garantie).

Das Modem wird an der TAE-FF links gesteckt, das Telefon rechts.

Anschlussmoeglichkeit fuer maximal:

- zwei Amtsleitung / zwei Telefone / ein Modem
- zwei Amtsleitung / zwei Telefone / ein Anrufbea
- zwei Amtsleitung / zwei Telefone / ein Fax
- eine Amtsleitung / ein Fax / ein Anrufbea / ein Telefon
- eine Amtsleitung / ein Modem / ein Anrufbea / ein Telefon
- eine Amtsleitung / ein Modem / ein Fax / ein Telefon

Anstatt eines Telefons, kann natuerlich auch jede andere Kombination von zwei oder sogar drei Geraeten gewaehlt werden.

Der Unterschied dieser Dose zur TAE-FF besteht nur darin, dass das linke Anschlussfeld ohne umstaendliche Bruecken einzubauen schon direkt mit dem mittleren Anschlussfeld verbunden ist. Das bedeutet. LINKS und MITTE stellen in sich eine TAE-FF dar. Die RECHTE Seite ist eine normale TAE-F Dose. Durch die Bruecke von oben 5 & 6 nach unten 1 & 2 wird aus diesen "zwei" Dosen eine und man kann nun drei (!!!) Geraete gleichzeitig anschliessen. Die Kabelloetereien und Verdrahtungen wurden oben ja schon ausgiebig erlaeutert. Ich glaube hier ist das nicht mehr erforderlich.

Wer nun einen Doppelanschluss hat, kann sich seine zwei Rufnummern nun auf eine Dose schalten und an den linken Anschluss einen Anrufbea, FAX oder Modem anschliessen.

TAE NFN-Kodiert:

```
AMT
REIN
|  |           ^  ^
|  |           |  |
v  v           |  |
O  O  O  O  O  O
1  2  3  4  5  6
 _____________________
|  __    __    __    |
|N|  |  F|  |  N|  | |
| |  |   |  |   |  | |
| |__|   |__|   |__| |
|____________________|
```

Das O stellt den Schraubanschluss dar !!!

Das obere Anschlussfeld ist fuer den linken, mittleren und rechten Steckschlitz

Anschlussmoeglichkeit fuer:

- eine Amtsleitung / ein Fax / ein Anrufbea / ein Telefon
- eine Amtsleitung / ein Modem / ein Anrufbea / ein Telefon
- eine Amtsleitung / ein Modem / ein Fax / ein Telefon

Anstatt eines Telefons, kann natuerlich auch jede andere Kombination von zwei oder sogar drei Geraeten gewaehlt werden.

Diese Dose bietet die Moeglichkeit wie die NFF-Dose, kann aber nicht aufgetrennt werden um zum Beispiel zwei Amtsleitungen mit zwei Telefonen anschliessen zu koennen.

Anschluss FAX / TELEFON

Der Anschluss der Dose erfolgt wie oben, also Amtsleitung bei Punkt 1 & 2 rein, bei 5 & 6 wieder raus, und unten bei 1 & 2 wieder rein. Ein Fax hat immer mindestens 4 Draehte. Zwei Draehte fuer in den Fax, und zwei wieder zurueck in die Dose. Hat der Fax einen alten ADO-8 Stecker, dann umloeten auf TAE-Stecker. Die vier oder fuenf Kabel kommen auf 1,2,5,6,(4). Durch probieren muss getestet werden, wie es zusammenpasst. Es gibt kein allgemeingueltiges Rezept dafuer. Ein eventuell fuenfter Draht ist fuer die Erde bei Nebenstellenanlagen. Eindeutige Zuordungen der Farben der Draehte kann ich hier nicht machen, ist unterschiedlich. Bei Fax die von Haus aus mit TAE-Steckern ausgeliefert werden, brauch nix geaendert zu werden, der funktioniert immer.

Der Fax wird links in die TAE gesteckt und das Telefon rechts.

Anschluss ANRUFBEANTWORTER / TELEFON

Bei einem Anrufbeantworter mit FTZ-Nummer und TAE-Stecker stimmt die Steckerbelegung immer, einfach links stecken, fertig. Hat der Anrufbeantworter einen alten ADO-8 Stecker, dann umloeten auf TAE-Stecker. Die vier Kabel (weiss, gruen, braun, gelb) kommen auf 1,2,5,6. Durch probieren muss getestet werden, wie es zusammenpasst. Es gibt kein allgemeingueltiges Rezept dafuer.

Bei einem NON-FTZ-Geraet werden zwei Draehte (zu 95% rot und gruen) auf den TAE-Stecker Punkt 1 & 2 geloetet und zusaetzlich noch mit Bruecken von 1 nach 5 und 2 nach 6 verbunden.

Bei einem alten FTZ-Anrufbeantworter (duerfen neuerdings nicht mehr angeschlossen werden, obwohl FTZ-Nummer), werden zwei Draehte (zu 80% weiss und braun) auf den TAE-Stecker Punkt 1 & 2 geloetet und zusaetzlich noch mit Bruecken von 1 nach 5 und 2 nach 6 verbunden.

Achtung: Es gibt vereinzelt Anrufbeantworter aelterer Bauweise, die funktionieren nicht ohne gestecktes Telefon !!!!

TAE NFF-Kodiert:

```
AMT         AMT
REIN        RAUS
|  |        ^  ^
|  |        |  |
v  v        |  |
O  O  O  O  O  O
1  2  3  4  5  6
 ____________________
|  __    __    __    |
|N|  |  F|  |  F|  | |
| |  |   |  |   |  | |
| |__|   |__|   |__| |
|____________________|

6  5  4  3  2  1
O  O  O  O  O  O
            ^  ^
            |  |
            |  |
            AMT
        WIEDER REIN
```

Das O stellt den Schraubanschluss dar !!!

Das obere Anschlussfeld ist fuer den linken und mittleren Steckschlitz
Das untere Anschlussfeld ist fuer den rechten Steckschlitz

Weitere Anschlusstips

Es soll ja die Oberfreaks geben, die wollen sich an eine Amtsleitung ein Telefon, ein Fax, einen Gebuehrenzaehler, ein Modem, einen Anrufbeantworter und eine BTX-Box anschliessen. Diese Kombinationen sind recht einfach zu verkabeln, wenn man folgendes einhaelt:

- Jedes Gearat muss die Amtsleitung nach hinten weiterschalten, ausser das Telefon. Alle Faxgeraete und neuen Anrufbeantworter koennen das, Bei Modems und aelteren Anrufbeas muss halt die Bruecke von 1>5 + 2>6 in den TAE-Stecker reingemacht werden.

- Gebuehrenzaehler mit TAE-Stecker werden genauso wie ein Fax angeschlossen (gleiche Steckerbelegung) und sind IMMER als erstes Geraet anzuschliessen. Dies ist notendig, damit er von allen Geraeten die Gebuehren zaehlen kann und die stoerenden Zaehlimpulse vor den anderen Geraeten ausfiltern kann.

- Wer Btx hat, schliesst nun die BTX-Box an. Ich hab leider keinen Plan wie das geht, mir ist noch keine BTX-Box vor die Augen gekommen.

- als nextes kommt wahlweise der Fax oder der Anrufbea. Kommt darauf an was oeffters benutzt wird. Was mehr benutzt wird, kommt zuerst.

- danach das Modem. Dann aber aufpassen, das keiner die Wahltaste am Fax drueckt, das koennte Probleme bringen.

- Und zu allerletzt das gute alte Telephone !!

Uebrigens: Hinter der ersten POST-TAE darf man soviel private TAE-Dosen schalten wie man will, vorausgesetzt man geht an der Post-TAE am Punkt 5 & 6 raus.

Zusammengestellt von RANTANPLAN, User der

BABUBO
DARMSTADT

24h Online ++49-6151-595240 oder
++49-6151-595200

CLUB 80 BÖRSE

Aus den alten TRS-80 Zeiten habe ich noch folgende Unterlagen und Software:

- Club 80 Info, ERST-Ausgabe b. Ausgabe 10, gebunden
- Club 80 Info, Ausgabe 11-17, gebunden
- Club 80 Info, ab Ausgabe 18, ungebunden
- Artikel aus der Zeitschrift '80 MICRO', gebunden
- Programmlistings, Artikel, Programmdokus, gebunden
- Programmhandbuch 'Utility Pack' u. 'Best of Model 4'
- TRS 80, Editor/Assembler Handbuch
- TRS DOS/LS DOS 6.2 Handbuch m. Systemfile-Listing
- TRS 80, DiskBasic 1.1, TRSDOS 2.0/2.1 Handbuch (7.7.78)!
- TRS-80 Disk & Other Mysteries
- Montezuma-CPM Handbuch

- ein Packen Disketten mit TRS-80 und CPM-Software

Die Sachen gebe ich kostenlos bzw. gegen Erstattung der Versandkosten ab.

Klaus Hermann
Tel.: 07127/71945
Box: 07127/70107
FIDO: 2:246/1307.740
oder Klaus_Hermann@Discovery.lifenet.org

CLUB 80 BÖRSE

Hallo Bastler !

Ich habe da ein Problem:

Kuerzlich verabschiedete ich mich bei einem CHAT mit den Worten: 'Leg Dich wieder hin, ich werd noch mir noch'n Weizen reintun und dann auch schlafen gehen...' Darauf der andere: 'Mensch, auf'n Weizen koennt ich jetzt auch...'

Also begab ich mich am darauffolgenden Tag in meinen Bastelkeller und fing an...

Nun ist es FAST fertig !!! Das WEIZENBIERUEBERTRAGUNGS-PROTOKOLL und die dazugehoerige Hardware:

Beschreibung der Hardware: sieht aus wie ne alte Schuhschachtel (Marke mit den Streifen...) Oben ein Trichter drin, an der Seite ein Zapfhahn und hinten eine Buchse fuer 'nen Kaltgeraetestecker und eine serielle Schnittstelle.

Beschreibung der Software: Protokoll aehnelt sehr dem Z-Modem- Protokoll: Je nach 'Guete der Leitung' wird die Uebertragungsmenge gesteigert...

Vorgang der Uebertragung: Bei beiden Stellen ist ein Rechner mit zwei seriellen Schnittstellen notwendig: eine fuer's Modem, die andere fuer die Hardware. Bier in den Trichter, Uebertragung starten, fertig. Auf der Gegenseite: Uebertragung starten und nach Beendigung das Bier ueber den Zapfhahn entnehmen.

Leider hatte ich dann ein Problem: Mit Weizen klappte die Sache teilweise genauso gut wie die Uebertragung dieses Textes. (Beispiel siehe 2 Zeilen hoeher...)

Manchmal ging es allerdings fuerchterlich schief...

Daraufhin habe ich spezielle Uebertragungsprotokolle entwickelt.
1. helles Weizen Kristall (KWh-Protokoll)
2. helles Weizen Hefe (HWh-Protokoll)
3. dunkles Weizen Kristall (dWk-Protokoll)
4. dunkles Weizen Hefe (dWh-Protokoll)

(3. konnte leider mangels Material nicht getestet werden...)

Mit normalen Bier (Pils oder so) gab es allerdings fuerchterliche Probleme:
Bei normalem Bier kam - egal, welche Marke - nur Wasser bei der Gegenstelle an. Ausnahme: BOELKSTOFF (hier kam teilweise alles richtig rueber, aber auch leider manchmal Milch an...)

Ich suche jetzt ganz dringend einen Programmierer, der mir ein Protokoll zur Pils- und Altbier-Uebertragung schreibt. Nach dem vielen Testen bin ich einfach zu erschoepft, um dies selbst zu machen...

In der Hoffnung auf baldige Antwort.

Prost Prian

Und dann kam diese Antwort...

Weizenprotokoll II

Offensichtlich herrschen immer noch reichlich Unklarheiten, was das Uebertragen von alkoholischen Getraenken angeht. Daher an dieser Stelle ein kurzer Abriss der technischen Anforderungen und Moeglichkeiten.

Zunaechst gibt es zwei grundsaetzlich verschiedene Arten der Alkoholuebertragung, diese sind gemaess der CCITA (Committee consultativ international de transmission du alcohol) Empfehlungen B.21 bis B.23bier unterteilt nach dem Volumenanteil des Alkohols:

B.21
- Tafelweine und Duennbiere bis 3.5 %
- (darunter fallen auch die sogenannten 'alkoholfreien' Biere, die
- allerdings besser im Halbschluckspecht-Betrieb nach B.08pust
- uebertragen werden sollten)

B.22
- Qualitaetsweine und Pilsener bis 5.3 %
- (hierunter fallen auch der sogenannte 'Boelkstoff' und Altbiere,
- sowie Schaumweine)

B.23
- Alt- und Weizenbiere, sowie Champagner

B.23bier
- Fuer den Uebertragungsbereich bis 21 %
- (geeignet fuer alle Bierarten und Weine, jedoch aus technischen
- Gruenden nicht fuer Schaumweine)

Ausserdem duerfte noch die Empfehlung A.22 interessant sein, die fuer Uebertragungen ab 22 % in allen Alkoholklassen gedacht ist, sowie A.22T fuer technische Alkohole und A.22S fuer Spiritus und Haarwasser.

Uns interessiert in diesem Zusammenhang also die Empfehlung B.23bier, da wir dort die meisten Moeglichkeiten haben, das Projekt Weizenbieruebertragung zu realisieren. B.23bier sieht grundsaetzlich die Uebertragung im Vollschluckspechtverfahren vor, bei dem beide an der Uebertragung beteiligten Seiten gleichzeitig einschenken und trinken koennen.

Wichtig ist vor allem eine korrekte Einstellung der Uebertragungsparameter, bei den derzeit erhaeltlichen BiMoDems (Bier-Modulator-Demodulator) sind leider meist nur zwei Einstellungen moeglich:

8N1

Es werden gleichzeitig acht Glaeser uebertragen, die Empfangsglaeser muessen geneigt sein, und zwar um 1 mal 15 Grad, zwecks Verhinderung von ueberhoehter Schaumbildung.

7E2

Es koennen nur sieben Glaeser zur Zeit uebertragen werden, die Glaeser duerfen gerade gehalten werden (E = even), oder um 2 * mal 15 = 30 Grad geneigt, was gerade bei Weizenbier empfehlenswert ist.

Die direkte Uebertragung ueber das Telefonnetz ist aber nur bei kleineren Entfernungen zwischen den Gegenstellen sinnvoll, da es bei laengeren Leitungen sehr oft vorkommt, dass das Bier unterwegs schal oder gar sauer wird. Deshalb hat die DBP (Doppelbock-Bier Pils) bereits im Jahre 1982 ein fassorientiertes Uebertragungsnetz in Betrieb genommen, das als BAREX bekannt ist. BAREX heisst 'barrel exchange' und bezeichnet die Art der Uebertragung. Es gibt zwei Arten von Diensten, die angeboten werden:

- BAREX-L ist das leitungsgebundene Verfahren, bei dem eine feste Leitung zwischen den Gegenstellen geschaltet wird, was sehr hohe Uebertragungsgeschwindigkeiten ermoeglicht und das Durchsickern an den Umschaltpunkten minimiert.
- BAREX-F fuellt die uebergebenen Faesser zunaechst mittels eines speziellen Rechners, dem BAD (bottle assembler/disassembler) in einzelne Flaschen ab, die dann in Kisten an einen BAD im Ortsbereich des Empfaengers uebertragen werden. Dort werden die Flaschen entkorkt und wieder zu kompletten Faessern zusammengekippt. Das einwandfreie Abfuellen und Ausleeren der Flaschen wird durch ein Schecksummenverfahren garantiert, d.h. die mit dem Umfuellen Beschaeftigten erhalten derart hohe Schecks, dass sie wirklich nicht mehr darauf angewiesen sind, fremde Flaschen zu trinken. Stoerungen, wie sie in der Einfuehrungsphase zu verzeichnen waren (Man erinnere sich nur an die gefuerchtete Fehlermeldung: 'BAREX-F: Voruebergehende Truebung im Glas'), gehoeren seit Einfuehrung von BIGFON der Vergangenheit an.
- BIGFON ist das Biere-integrierende Glasfasel-Ortsnetz und loest bis zum Uebergang auf ISDN (Internationales Sauf-Dienst-Netz) die herkoemmlichen Kupferleitungen ab.

Nach diesem kurzen Ueberblick ueber die netzabhaengige Seite wenden wir uns noch kurz der praktischen Seite zu.

Wir setzen voraus, dass hardwareseitig mindestens ein AT (Altbier-Technik) mit 640kB Hauptspeicher (kB = kleine Biere, 1 kB entspricht 0.2 Liter, mithin hat ein AT ein FASSungsvermoegen von rund 128 Litern). Besser ist natuerlich ein groesserer Hauptspeicher, dieser sollte dann aber unter EMS arbeiten koennen (EMS: europaeischer Malzbier Standard), insbesondere wenn auch Kinder Zugang zum Rechner haben (Kenntnis von Trinkername und Fasswort vorausgesetzt). Empfehlenswert sind 2 MB RAB (RAB: Random access barrel = Fass mit wahlfreiem Zugriff).

Ausserdem muss der Rechner vom ZZF (Zentralamt fuer Zulassungen im Fasswesen) zugelassen sein. Das erkennt man in der Regel an einem kleinem Metallschild an der Schnittstelle (Zapfhahn), auf dem die zugelassenen Biersorten vermerkt sind. Diese Schilder hat wohl jeder schon einmal an der Schnittstelle eines oeffentlichen Terminals (kurz Kneipe genannt) bemerkt:

'Erdinger Weissbier', 'Diebels Alt', etc...

Bei nicht zugelassenen Rechnern, so wird jedenfalls von Seiten der DBP immer wieder argumentiert, besteht die akute Gefahr, dass Verunreinigungen wie Milch oder gar Wasser das Netz beeintraechtigen.

Sind diese Voraussetzungen erfuellt, benoetigt man nur noch ein passendes Programm zur Uebertragung. Empfehlenswert sind dabei alle Produkte, die eine gesicherte Uebertragung ermoeglichen. Auf dem MS-GOS-Sektor (Multispirituosen-Glas-Operationssystem) sind das zum Beispiel PROFOMM (PROfessionelles Flaschen-Oeffnungs- und Misch-Modul) und TELHICKS. Beide unterstuetzen unter anderem Protokolle wie G-Modem (glasweises Uebertragen mit zyklischem Schaumcheck), F-Modem (flaschenweises Uebertragen mit Pfand) und F-Modem box (kasten-orientiertes Flaschenuebertragungssystem mit wahlweise 12 oder 24 Flaschen).

Wer trotz dieser sehr empfehlenswerten und preiswerten Produkte lieber auf selbst Programmiertes zurueckgreifen will, sollte sich sehr genau mit dem Befuellsystem seines Rechners auskennen. Mit schlichten Hicks und Hocks, wie sie in BASIC (beginners alcoholic stuff instruction code) moeglich sind, kommt man meist nicht weit. Grundsaetzlich gilt fuer alle Hochsprachen: Zunaechst wird mit ASSIGN eine Biersorte einem Glas zugewiesen.Danach (!) kann die Schnittstelle mit FILL geoeffnet werden, sofern es sich um den ersten Zugriff handelt. Zum Nachschenken oeffnet man selbstverstaendlich mit REFILL. Man sollte natuerlich nicht vergessen, nach dem Einschenken die Schnittstelle wieder zu schliessen, bevor es zu einem Dropout kommt. Bei groesseren gesellschaftlichen Anlaessen ist die Anschaffung eines Streamers ratsam, da nur damit die Bierstroeme halbwegs vernuenftig gesichert werden koennen. Das staendige Flaschenwechseln wird einem sonst schnell zuviel, selbst wenn man die modernen 3.5-Liter Flaschen verwendet.

Der Vollstaendigkeit halber hier ein kurzes Programm in Pascal, welches die grundsaetzliche Vorgehensweise veranschaulicht. Dies ist kein Demoprogramm, da das hiesige Ordnungsamt keine Genehmigung dafuer erteilt hat!

PROGRAM Weizenbier;

```
CONST   Weizen = 0 ;
        Hefeweizen = 1 ;
        Export = 2 ;
        Pils = 3 ;
        Alt = 4 ;
        Bock = 5 ;
        Urbock = 6 ;
        Maibock = 7 ;
        Festbock = 8 ;
        Keinbock = 9 ;
        Doppelbock =10 ;
        nuechtern = 0 ;

TYPE    ccm = Fluessigkeit ;
        kleines_Bier  = ARRAY [0..200] OF ccm ;
        grosses_Bier  = ARRAY [201..500] OF ccm ;
        Stiefel  = ARRAY [501..4500] OF ccm ;
        Korn  = ARRAY [0..20] OF ccm ;

VAR     leer        : BOOLEAN ;
        voll        : BOOLEAN ;
        promille    : INTEGER ;
        luett       : kleines_Bier ;
        halber      : grosses_Bier ;
        und_luett : korn ;
        bestellt    : integer ;

FUNCTION pusten : integer ;
EXTERNAL ;

FUNCTION received : boolean ;
EXTERNAL ;

FUNCTION Bier_fertig (welches:integer) : boolean ;
EXTERNAL ;

PROCEDURE einschenken (welches:integer) ;
EXTERNAL ;

PROCEDURE warten (wielange : geduldig) ;
EXTERNAL ;

PROCEDURE austrinken (welches:integer) ;
EXTERNAL ;
```



```
BEGIN
        REPEAT
        write('Welche Sorte (0..10): ') ;
        readln(bestellt)
        UNTIL bestellt IN [Weizen..Doppelbock] ;
        REPEAT
        REPEAT warten UNTIL received ;
        REPEAT
                einschenken ;
                warten
                UNTIL Bier_fertig(bestellt) ;
                austrinken(bestellt)
        UNTIL pusten >= 0.8
END.
```

Die extern deklarierten Prozeduren stammen aus dem Weizenbiertreiber hefeBASE IV+ von Erston Drink. Natuerlich muss das Programm, um wirklich einwandfrei zu funktionieren, noch um die ueblichen Fehlerroutinen erweitert werden. Dazu mag die folgende Liste der Fehlermeldungen hilfreich sein:

Code	Bedeutung
0	glass empty
1	glass broken
2	glass not ready
3	too many glasses
4	too much vine
5	too much song
6	too toot...
7	no carrier
8	no carreer
9	out of bottles
A	put ice into glass A:
B	straw bend
C	shake well before use
D	shake, rhythm & roll
E	synthicks error after glass nnn
F	out of jokes

Mach(t)s guat

Impressum

1. Vorsitzender	Hartmut Obermann Wilhelm-Baur-Straße 8 76135 Karlsruhe	Tel.: 0721/85 40 68 Fax: 0721/85 40 68
2. Vorsitzender	Gerald Schröder Am Schützenplatz 14 21218 Seevetal	Tel.: 04105 /26 02
Hardewarekoordinator	Andreas Magnus Bismarckstraße 29 45879 Gelsenkirchen	Tel.: 0209 /87 02 30
NewDOS-Diskothekar	Oliver Volz Am Ochsenwald 37A 70565 Stuttgart (Rohrerhöhe)	Tel.: 0711/74 40 51
CP/M-Diskothekar	Fritz Chwolka Saarstraße 34 52457 Aldenhoven	Tel.: 02464 /89 20
C-128-Diskothekar	Günther W. Braun Postfach 80 02 26 81602 München	
Clubbücherei	Kurt Müller Sophie-Scholl-Ring 3b 21502 Geesthacht	Tel.: 04152 /7 06 43
Redaktion	Jens Neueder Rudolf-Then Straße 32 74544 Michelbach /Bilz	Tel.: 0791 /4 28 77 FAX:07971 /2 50 55

Bankverbindung des CLUB 80

Postgirokonto Sonderkonto CLUB 80
Obermann H., 89312 Günzburg
Konto Nr. 496 071-605
Postgiroamt Frankfurt, BLZ 500 100 60

Autoren — Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.
Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.

Die Redaktion.

Schluß

Hallo Club 80'er,

wieder einmal haben wir ein weiteres Jahr zu Ende gebracht. Wieder einmal der Dank an die, die dem Club 80 so verbunden waren, um etwas Arbeit zum Nutzen Aller zu investieren.

Üblicherweise ist es dann auch an der Zeit Resümee zu ziehen. Hierbei fällt leider auf, daß die Aktivität im Club 80 allgemeinen sehr gering ist:

- in der Discovery finden sich ab und zu nur drei/vier Clubmitglieder per Modem ein
- zur Hobbytronik 94 waren auch nur drei/vier Clubmitglieder da
- im Info sind meist auch immer die gleichen Autoren zu lesen (Wenn wir die nicht hätten! Sie ermöglichen uns immerhin die vierteljährliche Erstellung des Club-Info's)

Bei fast 60 Clubmitgliedern sollte meiner Meinung nach doch etwas mehr los sein, als nur ca. 10%-ige Aktivität! Die wenigen Autoren stellen jedenfalls fest, daß Sie zu Ihren Artikeln fast keinen Feedback erhalten, dies ist der Motivation derselben sicher nicht gerade förderlich.
Hier stellt sich nun die Frage, ob der **80'er-Computer** nicht doch schon ein fosiles Überbleibsel ist? In der **i-kompatiblen** PC-Schwemme geht er doch jedenfalls unter!
Und nun hier die Frage:

Was machen denn die anderen 90% im Club so mit Ihrem Rechner?

Meldet Euch doch mal, es ist sicher interessant, was die Mehrheit so mit Ihrem Computer treibt. Vielleicht finden sich auf diesem Weg neue Gemeinsamkeiten. Sicher ist es nicht sinnvoll, daß manch Einer mit irgendeinem Problem still in seinem Kämmerchen vor sich hinbröselt, wo der Clubkamerad die Lösung vielleicht schon auf dem Tisch hat. Abhilfe schafft hier nur die Kommunikation untereinander, als Sprachrohr diene uns hierzu die INFO.

Mit der Hoffnung auf viele neue Beiträge, den besten Wünschen für die Feiertage und den Jahreswechsel verbleibt Euch Euer

Jens

PS.: Bitte denkt an die rechtzeitige Anmeldung zum Clubtreffen.

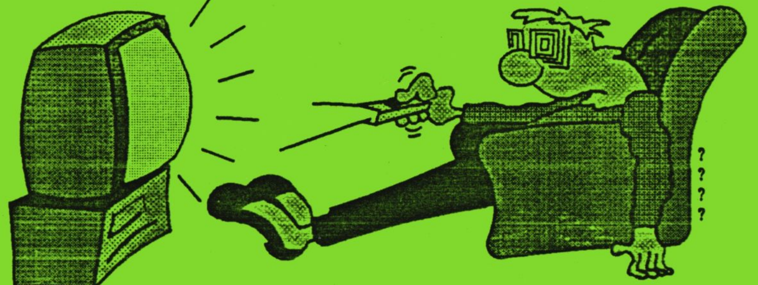
Adressen-Liste CLUB80

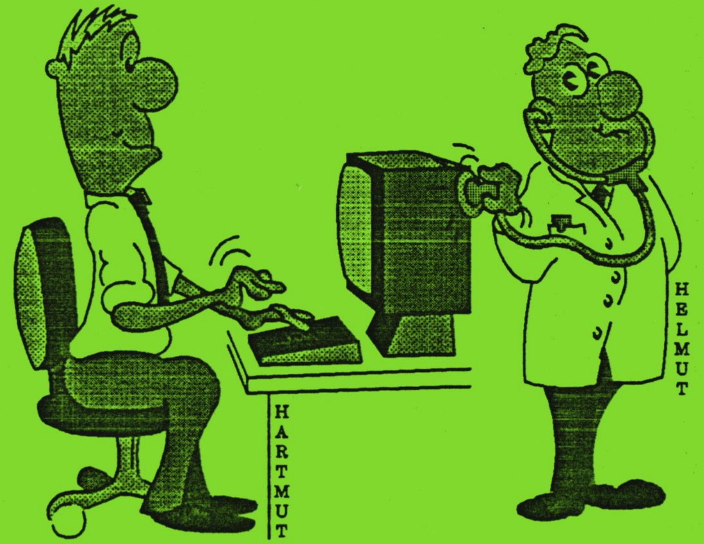
Stand vom 30-12-1994

lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
1	Barendt	Harry	Hermann-Löns-Straße 7	GER 50181	Bedburg (Erft)	02272/ 7168	02272/ 7168	-	-
	IBM 386DX40, CoProz■					-	-	2:2433/554.12	-
2	Berndt-Jochum	Ilse	Stachelsgut 24	GER 51427	Bergisch Gladbach	02204/ 65254	-	-	-
	IBM, GENIE III, GENIE IIIs, SHARP Pocket 160■ Scanner					02204/ 65254	-	-	-
3	Bernhardt	Helmut	Preetzer Straße 75	GER 24143	Kiel	-	-	-	-
	IBM, Prof 180, CPU 280■ diverse PCs,Novellite-Netz,1496E,Mustek105+					0431/ 77578-20	0431/ 77578-99	2:242/262.26	-
4	Bielenberg	Georg	Erikaweg 1	GER 24568	Kaltenkirchen	04191/ 3751	-	-	-
	Atari 260ST, C128, Schneider Joyce■ Modem: LC2496 Digitech, Scanner: Supersc.III					04193/ 90430	-	-	-
5	Brans	Jörg	Tieloh 55	GER 22307	Hamburg	040/ 6906531	-	-	-
	IBM■ CD-Rom, Streamer, Soundkarte					-	-	-	-
6	Braun	Günther W.	Postfach 80 02 26	GER 81602	München	-	-	-	Discovery
	Commodore 128 D■ Akustikkoppler					-	-	-	07127/ 70107
7	Braun	Harald	Postfach 8011	GER 24154	Kiel	0431/35139	-	-	-
						-	-	-	-
8	Böckling	Ulrich	Juchaczstraße 61	GER 56203	Höhr-Grenzhausen	02624/ 4861	-	-	-
	IBM 386DX33,1040ST,VC20,C64,TRS80 M I,ZX81■ Modem, Videodat-, Videotextdecoder					02631/ 895168	-	-	-
9	Chwolka	Fritz	Saarstraße 34	GER 52457	Aldenhoven	02464/ 8920	-	-	-
	IBM 386+Co, Commodore, Apple, Z 280■ Modem 2.4					-	-	2:248/242:8	-
10	Dose	Volker	Dorfstraße 10	GER 24235	Brodersdorf	04343/ 1357	-	-	-
	GENIE IIIs mit Z180■ EPROMer, Modem					-	-	-	-
11	Halgasch	Gert	Großschönauer Straße 26	GER 02796	Jonsdorf	035844/ 636	-	-	-
	IBM 386DX■					-	-	-	-
12	Hartmann	Hans-Günther	Möwenstraße 9	GER 27804	Berne	04406/ 6911	04406/ 1071	-	-
	IBM 386SX, TANDY M4p, Z280-Kartenrechner■ Real Time Clock, Speed Up Kit 6,3MHz, 40MB-P					0421/ 248-2232		2:2426/3050.24	-
13	Hebecker	Ulrich	Büsnauer Straße 15	GER 70563	Stuttgart	0711/ 734800	-	-	-
	IBM 286+386, 128D, 1581, Kaypro484■					-	-	-	-
14	Held	Manfred	Stirner Straße 22	GER 91785	Pleinfeld	09144/ 6563	09144/ 8514	-	-
	IBM■ Modem ZyXEL 1496+, CD-ROM					0911/ 219-2245	-	2:2400/10.10	-
15	Hermann	Klaus	Forchenstraße 8	GER 72124	Pliezhausen	07127/ 71945	-	-	Discovery
	IBM 386■ ET 4000, Modem					-	-	2:2407/70.740	07127/70107
16	Hürdler	Manfred	Niederhoferstraße 29	GER 97222	Rimpar	09365/ 4235	-	-	-
	Victor Sirius 1 (IBM), CPC 6128■					-	-	-	-
17	Johnen	Willi	Hansemannstraße 1	GER 52351	Düren	02421/ 501305	-	-	-
	GENIE IIIs■					02421/ 33064	-	-	-
18	Kalb	Michael	Königstraße 10	GER 96242	Sonnefeld	09562/ 6387	-	-	-
	Schneider CPC, Joyce■ Modem, Drucker					-	-	-	-
19	Kauka	Dietmar	Straße des Friedens 37	GER 04552	Neukirchen (Borna/Leipz	03433/851019	-	-	-
						-	-	-	-
20	Kemmer	Jürgen	Dorfberg 7	GER 97232	Sulzdorf	09334/ 1050	-	-	-
	IBM 386, UltraSound, ZyXEL U1496E■ DCF-77-Empfänger parallel, IR-Sender seriell					-	-	2:247/2086.1	-
21	Kleespies	Andreas	Mainzer-Land-Straße 765	GER 65934	Frankfurt /Main	069/ 387432	-	-	-
						-	-	-	-
22	Kuhn	Eckehard	Im Dorf 14	GER 72636	Frickenhausen	07022/ 45417	-	-	-
	Atari ST 1040, TRS80 M I■					-	-	-	-
23	Linder	Jörg	Küstriner Str. 68	GER 15306	Seelow	03346/ 520	-	-	-
	KC 85/4 mit Floppy■					-	-	2:2480/115.11	-
24	Lorenz	Walter	Mahräckerstraße 9	GER 60431	Frankfurt /Main	069/ 531656	-	-	-
	IBM 286/486, Z80-, HD 64180-Eigebau, Z280-T.■ Scanner, Soundkarte, ET 4000					-	-	-	-
25	Magnus	Andreas	Bismarckstraße 29	GER 45879	Gelsenkirchen	0209/ 144029	-	-	-
	IBM 386Dx, GENIE IIIs■ Modem 2400					-	-	-	-

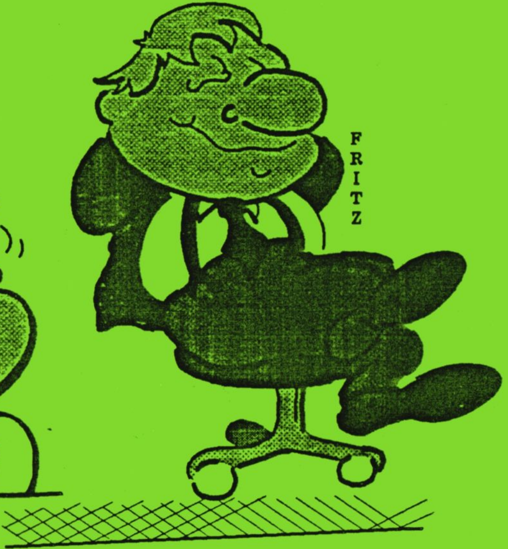
lfd. Nr.	Nachname	Vorname	Straße	PLZ	Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
26	Mahlert	Herbert	Hohenbudbergerstraße 112 A	GER 47229	Duisburg	02065/ 47217	-	-	-
	IBM, GENIE I■ c't Videotext-Karte, Vobis Videodat-Decoder					02065/ 902592	-	-	-
27	Messerschmidt	Kurt	Hanns-Eisler-Straße 54	GER 10409	Berlin	-	-	-	-
						-	-	-	-
28	Müller	Kurt	Sophie-Scholl-Ring 3B	GER 21502	Geesthacht	04152/ 70643	-	-	-
	IBM, Atari Mega ST4■ ATONCE 386,OverScan,Scanner,HBS640-T36,Modem					040/ 89983403	-	-	-
29	Mössel	Franz	Schafferstraße 12	I 39012	Meran	0039-473/34178	-	-	-
	IBM, Joyce, Workemate-Bullet■ Modem					0039-471/980496	-	2:333/400	-
30	Neueder	Jens	Rudolf-Then-Straße 32	GER 74544	Michelbach / Bilz	0791/ 42877	-	791444722	CCWN
	IBM 286/386/486, Atari ST 1040, TRS80 MI■ Modem, Sound Galaxi NX, Scanner, Tape, Borsu					07971/ 250-50	07971/ 250-55	1001667	0715168434
31	Neumann	Christof	Zeitblomstraße 22/2	GER 89077	Ulm /Donau	0731/ 6022568	-	-	-
	IBM, Tandy MII, Tandy M4p■ Novell					0731/ 9749720	-	-	-
32	Nitschke	Stefan	Germenenstraße 5	GER 75045	Walzbachtal 1	07203 /452	-	-	-
	CPU 280■ NEC 7220 1024*1024 Grafikkarte					-	-	-	-
33	Obermann	Hartmut	Wilhelm-Baur-Straße 8	GER 76135	Karlsruhe	0721/ 854068	-	-	SYNREL
	IBM486, Tandy M4p, Epson PX-8■ Modem, Scanner					-	-	2:2476/538.10	08282/ 4311
34	Peters	Jürgen	Heukoppel 14	GER 22179	Hamburg	040/ 6412371	-	-	-
	IBM, Z280, MZ821, SHARP■ Modem, Soundkarte, Video, Drucker					-	-	-	-
35	Retzlaff	Bernd	Kleiner Sand 98	GER 25436	Uetersen	04122/ 43551	-	-	-
	IBM 386, C64, GENIE I■					04103/ 605310	-	-	-
36	Rinio	Gerd	Rennbahnstraße 9	GER 22111	Hamburg	040/6552630	-	-	-
	IBM 486DX66/2, RTS 80, TRS 80 RS■ Modem					-	-	-	-
37	Ruschinski	Claus	Pommernstraße 21	GER 45770	Marl	02365/ 34646	-	-	-
	IBM 386, TRS80 M I■ Highscreen-Scanner, CoProz IIT387					-	-	-	-
38	Schimmer	Jörg	Stettinerstraße 28	GER 60388	Frankfurt	06109/ 35336	-	-	-
	IBM 486, Schneider CPC■ Modem CSR2400					069/ 3800-2385	-	2:244/1107.9	-
39	Schmid	Alexander	Entmannsdorf 5	GER 96317	Kronach /Gehülz	09261/ 53496	-	-	-
	GENIE IIs, GENIE IIIs, CPU 280■ Modem 14400,Prommer80,Ramdisk,Club80Terminal					-	-	2:2490/4015.2	-
40	Schmitz	Rainer	Küferweg 12/1	GER 73099	Adelberg	07166/ 1397	-	-	Discovery
	IBM,Portfolio,NCR Decision Mate V,Joyce,...■ Modem 1200, Märklin Digital Interface 6050					07161/ 608-475	-	-	07127/ 70107
41	Schoberth	Uwe	Petrus-Waldus-Straße 14	GER 75443	Oetisheim	07041/ 7254	-	-	Discovery
	Alphatronic P3■					0711/ 89394500	0711/ 89394513	-	07127/ 70107
42	Scholz	Hans-Werner	Spitalstraße 54	GER 41334	Nettetal	02157/ 3613	-	-	-
	IBM 386, Prof 80, ITT 3030■ Prommer80					-	-	-	-
43	Schroers	Horst-Dieter	Breslauer Straße 9	GER 85622	Feldkirchen	089/ 9032615	089/ 9043413	-	-
	IBM■ Modem, Scanner					-	-	-	-
44	Schröder	Gerald	Arminiusstraße 2	GER 22525	Hamburg	040/ 8507131	-	-	-
	IBM 386SX, Atari 1040STF, Z280■					040/ 54715334	-	-	-
45	Schröer	Egbert	Joachimstraße 18	GER 46284	Dorsten	02362/ 75311	-	-	-
	Portfolio, TRS80 MI, GENIE I, GENIE IIIs■					02362/ 49-9649	-	-	-
46	Schulte	Hartmut	Entenschnabel 8	GER 31311	Uetze	05173/ 1248	05173/ 24631	-	-
	IBM, Z280 u.a.■ Scanner					-	-	-	-
47	Sonnemann	Harald	In den Eckwiesen 9	GER 64405	Fischbachtal	06166/ 8512	-	-	-
	NDR-Klein■ parallel/seriell, EPROMer					06151/ 92-1265	-	-	-
48	Staimer	Ulrich	Dekan-Mayer-Straße 15	GER 86199	Augsburg	0821/ 993104	-	-	-
	IBM (Siemens), C128, Reh Z280■ Video, Panasonic-Drucker					0821/ 2574-107	-	-	-
49	Stumpferl	Stefan	Hasenberglstraße 57	GER 80933	München	089/ 3138193	//0893144001//	-	-
	Amstrad CPC 6128+, GENIE IIIs■ Modem, 20MB-Wechselplatte, PSG&PIO,SIO, ...					-	-	-	-
50	Sörensen	Rüdiger	Wiesbadener Str. 28B	GER 55252	Mainz-Kastel	06134/65342	-	-	-
						-	-	-	-

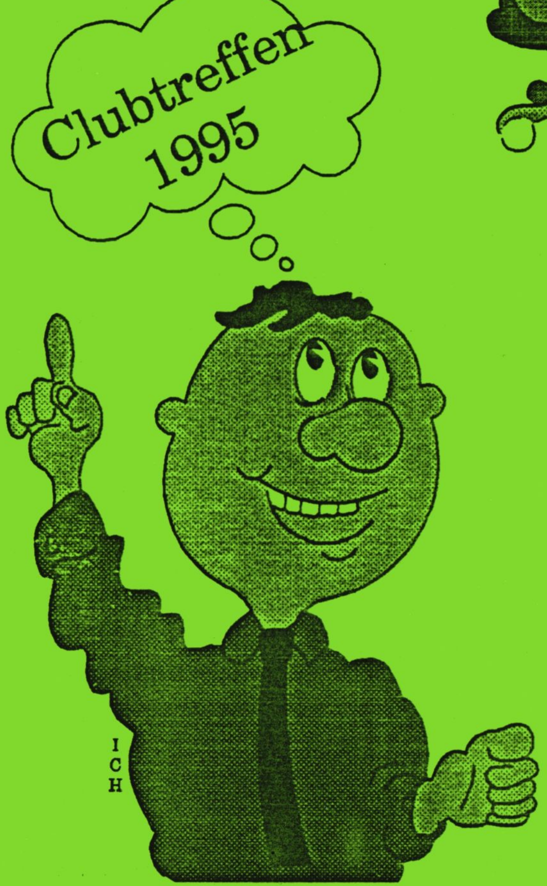
lfd. Nr.	Nachname	Vorname	Straße	PLZ Ort	Telefon privat / Tel. geschäftl.	Telefax privat / FAX geschäftl.	BTX / FIDO-Node	Mailboxname / Mailboxnummer
51	Tornow	Wilhelm	Elbblick 46	GER 21629 Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
	IBM 386DX, Atari Mega ST4, Tandy M4p■ Streamer,Soundblaster,Modem 2400,F.A.K.S. 91				-	-	-	-
52	Vogl	Michael	Postfach 10 06 10	GER 41490 Grevenbroich /Laach	02181/45112	-	0218145112-0001	-
	IBM, Amstard CPC 464/6128■ Modem Digitek 2400, Scanner Dart F.CPC				-	-	2:2433/554.2	-
53	Volkmer	Richart	Am Spörkel 69	GER 44227 Dortmund	0231/ 752574	-	-	-
	IBM XT, Apple LC, CPC 6128, Osborne, TandyMI■ Modem 2400, ScanMan-Scanner				-	-	-	-
54	Volz	Oliver	Am Ochsenwald 37A	GER 70565 Stuttgart (Rohrerhöhe)	0711/ 744051	-	-	-
	IBM, GENIE IIs■ Modem Avantec				0711/ 685-3013	-	-	-
55	Walter	Hans-Jürgen	Elssholzstraße 9	GER 10781 Berlin	-	-	-	-
					-	-	-	-
56	Werner	Heiko	Reichenberger Straße 5	GER 01129 Dresden	0351/ 4608612	-	-	-
	IBM 286/486■ Modem 2400, Scanner				-	-	-	-

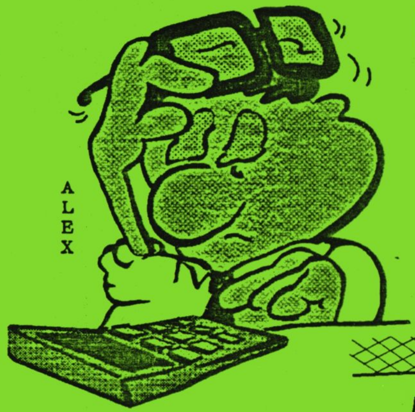




Hardwarespezialisten







Soft-
Ware-
Profies

CLUBINFO

47. Ausgabe

Kontaktadresse:

Club 80

Hartmut Obermann

Wilhelm-Baur-Straße 8

76135 Karlsruhe

Tel.: 0721/85 40 68

Anrufbeantworter &

FAX: 0721/85 40 68

Inhaltsverzeichnis

	Autor & Seite
Clubinternes	
Neues vom Vorstand	1 - 2 Hartmut Obermann
Mitgliedvorstellung	3 - 4 Andreas Kleespies
Termine, INFO´s	4 Redaktion
Neuzugänge	5 Hartmut Obermann
Sonstiges	
CP/M-Mailbox	- 6 Helmut Jungkunz
Gebührenimpulsfilter	7 - 11 Alexander Schmid
Z 380-CPU	12 Jörg Schimmer
Tips und Tricks zu MS-DOS	13 - 14 Georg Bielenberg
Die letzten Seiten	
Impressum	15
Schluß	16 Redaktion
Mitgliederadressenliste	am INFO-Ende
Anfahrtsliste Clubtreffen 95	am INFO-Ende

Neues vom Vorstand

Liebe CLUB 80-Mitglieder,

eigentlich bin ich etwas enttäuscht über die (ausgebliebenen) Reaktionen auf mein Vorwort zum Info 46. Da versucht man für den Club und vor allem für seine Mitglieder etwas zu tun, und was ist das Ergebnis? Richtig geraten - annähernd null! :-(((

1. CLUB 80-Treffen 1995

Die Anmeldungen zum Treffen liefen sehr zähflüssig ein, auch wenn sich inzwischen abzeichnet, daß die Veranstaltung auch in diesem Jahr gut besucht und ein voller Erfolg sein wird.

Dazu werden unter anderem beitragen:
- Rainer Schmitz mit dem Anschluß von Modems an CP/M-Rechner (ev. Stecker und Kabel mitbringen) sowie der Anpassung der Terminalsoftware an die Rechner
- Fritz Chwolka mit einer unter OS/2 installierten Mailbox und einer CP/M-Shareware-CD
- Helmut Bernhard mit seinem Telefonnetzsimulator, damit echtes DFÜ-Feeling aufkommt, ohne das es gleich viel Geld kostet!

Wer sich jetzt doch noch entscheiden will zu kommen, sollte sich schleunigst anmelden. Anmeldeschluß ist der 15. März 1995!

2. CLUB 80-Logo

Leider sind bisher nur zwei Vorschäge für ein neues CLUB 80-Logo eingegangen. Aber was solls, eines davon wird ab dem Treffen die CLUB 80-Infos schmücken. Die Einsender der beiden Vorschläge dürfen sich jetzt schon freuen, schließlich stehen ihre Chancen auf den Gewinn des Preises (Modem) 50 zu 50!

3. DFÜ im CLUB 80

Die "DFÜ im CLUB 80"-Fragebogenaktion war der Schlag ins Wasser schlechthin! Nicht ein einziger Fragebogen wurde zurückgeschickt. Wer das als absolutes Desinteresse an der Datenfernübertragung wertet, scheint recht zu haben. Für das zu Testzwecken angeschaffte Modem hat sich nämlich bisher auch kein Interessent gefunden. Niemand scheint den kostenlosen Service des CLUB 80 annehmen und das Modem ausleihen zu wollen. Und das, obwohl man damit sicher den kostengünstigsten und gefahrlosesten Einstieg in diesen Bereich der Computerei machen könnte.

Sollten sich bis zum Clubtreffen keine Interessenten melden, die das Modem ausleihen und damit den Einstieg in die DFÜ-Szene wagen wollen, wird das Gerät als Preis für den CLUB 80-Logo-Wettbewerb genutzt (was die beiden Einsender freuen dürfte!).

4. Modell 3/4 - Emulator

Auch die angeschafften Emulatoren fanden keinerlei Zuspruch. Wäre es jetzt Sommer, würde ich davon ausgehen, daß sich gerade alle Mitglieder die Sonne auf den Pelz brennen lassen und die Computerei erst wieder im Herbst/Winter als Hobby an Bedeutung gewinnt. Aber **wir haben** Winter! Wenn die Aktivitäten jetzt schon um Null tendieren, wie soll das dann erst im Sommer werden!? :-(((

5. Neue Männer braucht ... der CLUB 80!

Ihr könnt euch vielleicht vorstellen, daß die oben beschriebenen Reaktionen auf unsere Bemühungen, den CLUB 80 wieder zu etwas mehr Leben zu verhelfen, nicht gerade ermutigend sind. Zwischenzeitlich war ich schon fast soweit, das Handtuch zu werfen und mich für eine Wiederwahl nicht zur Verfügung zu stellen. Sollte sich jemand finden, der den Posten als 1. Vorsitzender des CLUB 80 übernehmen will und der beim Treffen von den anwesenden Mitgliedern gewählt wird, bin ich gerne bereit, meinen Platz zu räumen.

Ebenso geht es dem 2. Vorsitzender Gerald Schröder und dem Hardwarekoordinator Andreas Magnus. Beide sind zwar bereit, ihr (zur Zeit recht passiv ausgeübtes) Amt weiterhin so gut es geht auszufüllen, wollen aber ihren Platz räumen, falls sich ein anderer Kandidat für ihre Posten findet.

Oliver Volz (NewDOS-Diskothekar), Günther W. Braun (C-128-Diskothekar) und Rainer Schmitz (Clubbücherei) würden dagegen ihre Aufgabe, so sie wiedergewählt werden, gerne weiterhin wahrnehmen.

Nicht anders steht es mit unserem Info-Redakteur Jens Neueder, zu dem es aber meiner Meinung nach eh keine Alternative gibt!

Sollte also jemand die Zeit und die Lust haben, einen der vakanten Posten innerhalb des CLUB 80-Vorstandes zu besetzen und auszufüllen, dann sollte er sich entweder auf dem Treffen in Fürth am Berg einfinden und sich dort persönlich zur Wahl stellen, oder er soll mir seine Kandidatur (am besten mit ein paar begleitenden und erklärenden Worten) per Post zukommen lassen.

So, das war´s mal wieder von meiner Seite.

Ich würde mich sehr freuen, wenn sich die Mitgleider des CLUB 80 auch mal wieder etwas mehr von ihrer aktiven Seite zeigen würden. Ansonsten werden wir wohl demnächst dem (bisher nur telefonisch gemachten) Vorschlag von Günther W. Braun folgen müssen, das CLUB 80-Info nur noch zweimal im Jahr erscheinen zu lassen. Und wenn wir erst mal soweit sind, dann dürfte das Ende des CLUB 80 sehr nah sein!

Vorstellung

Nein nein, nicht ich stelle mich hier vor, mich kennen ja eh alle (so sollten es wenigstens sein). Vielmehr hat Andreas Kleespies mir vor einiger (langer) Zeit mal eine Mail zukommen lassen, die eigentlich schon längst im Info hätte erscheinen sollen. Leider ist die Datei irgendwann mal unbemerkt im Datennirvana entschwunden und erst beim Zurückschreiben eines Backups (wozu so ein Totalabsturz doch gut sein kann) vor kurzem wieder aufgetaucht. Ich hoffe, ihr, und vor allem Andreas, verzeiht mir, H.O.!

Hallo Leute !

Hier schon mal vorab meine Vorstellung als neuer Mitmacher im Club 80.

Name : Andreas Kleespies
Strasse : Mainzer Landstrasse 765
Ort : 65934 Frankfurt M.
Tel. : 069/387432
e-mail : kleesp@informatik.uni-frankfurt.de, sowie sporadisch hier in der Discovery.

Beruflich studiere ich Informatik und bin zwecks Broterwerb Aushilfsoperator an einer IBM AS/400. Tja, wie kommt man dazu sich mit Steinzeitrechnern zu beschäftigen, wo doch die lieben Mitstudis erst ab 32 bit Busbreite leuchtende Augen bekommen ?

Warum wird zur Steuerung des G-Kat im Auto kein AT eingesetzt, sondern u.a. eine 1802 CPU (mit 8 Bit), im "Bordcomputer" des Kfz ein MK 3870 tickert und demnaechst vielleicht ein Z80 in der Kaffeemaschine ? Sobald man von einem Rechner als universelles Hochleistungswerkzeug abrückt finden sich oft Pfennigs-Käferchen auf der Platine. Man kann mit weniger (mehr als auspacken) Aufwand (vgl. die Stangenlösung für DOS) seine eigenen Anwendungen selbst bauen und programmieren, Änderungen an der Hardware vornehmen (wenn man bereit ist, am Anfang genug Frust zu investieren). Eine willkommene Abwechslung zu dem sonstigen Theoriewust. (Soviel zu Lust und Frust)

Rechnertechnisch habe ich hier einen Schneider Joyce, Kaypro II/4, NDR, MC-CPM und einen 286 er Schlepptop rumstehen. Falls jemand Infos zu den Geräten braucht (oder hat), ran an den Rechner, Telefon, Flaschenpost, Brieftaube. Ich bin für Austausch sehr zu haben. Stellt sich nur noch die Frage was man mit soviel Zeug anstellt: DFÜ , DTP (Micro Design), Hardwareerweiterungen wie Belichtersteuerung für die Dunkelkammer, Protokolle der Intensität alkoholischer Gärung (klingt interessant, wie?), im Winter Schneewarnung (Nebenjob #2: Schneeräumer), usw... Zur Zeit arbeite ich daran, meinen MC-Rechner zum Coprozessor für den Joyce umzubauen (Multiprocessing mal anders), um Docs auszudrucken, Disketten formatieren, Dateien per LYNC umzupumpen und vielleicht später mal meinen privaten DFÜ-Anrufentgegennehmer zu schnitzen.

Also hoffe ich mal, dass ich entsprechend die Zeit finden werde, um mich am Clubleben zu beteiligen und daß der nächste Artikel nicht so lange auf sich warten läßt. Ich bitte den geneigten Leser, keine üeberhöhten Erwartungen an die fachliche Qualität zu stellen, dennoch wird Computer in jedem Computer dritten Wort Computer das Wort Computer vorkommen (minimale Forderung an einen Artikel, vgl. Treffen). <= gekonnte Ueberleitung

Das Treffen fand ich sehr interessant, um die Leute mal kennen zu lernen (einige kannte ich bis dato nur per Telefon). Die Vielfalt der mitgebrachten Rechner ist einmalig und ein Punkt, der sehr fuer den Club 80 spricht. Erfahrungsaustausch, Tips und Hilfe beim Troubleshooting steigern die Bereitschaft dafür ein paar hundert Kilometer zu fahren. Die Frage welchen Namen das Kind haben soll ist dagegen zweitrangig.

So long erstmal   Andreas

(Ich hoffe bis zum nächsten mal den Artikel über Dual-Slope Wandler fertig zu haben. Der Leser wird danach mehr zur praktischen Verwendung von Kodierungstheorien wissen als mancher Informatiker - Prof.)

Hallo Andreas,

wie war das doch noch gleich mit dem Artikel über den Dual-Slope Wandler? Wolltest Du den bis zum Erscheinen des nächsten Infos fertig haben oder bis zum nächsten Clubtreffen? In beiden Fällen wird es jetzt langsam Zeit! ;-)))

Always good hacking, H.O.!

Termine:

Club80-Clubtreffen
Club80-Jahreshauptversammlung
MEVOCOM
Multimedia Tage
Electronic Entertainment EXPO
COMDEX/Spring '95

Neustadt, Fürth am Berg
Berlin
Münster
Los Angeles
Atlanta, Georgia, USA

14.04. - 17.04.95
15.04.95
21.04. - 23.04.95
28.04. - 30.04.95
11.05. - 13.05.95
23.05. - 26.05.95

Redaktionsschluß für das nächste INFO ist der:
01. Juli 1995

Adressänderung **Alexander Schmid**

neu: Barbarastraße 64
97074 Würzburg
Tel.: (0931) 78 33 71
Fido: 2:2490/5230.10

!!! NEUZUGÄNGE !!!

Der CLUB 80 hat drei Neuzugänge zu verbuchen, die ich hiermit recht herzlich in unserer Mitte begrüßen möchte. Es handelt sich um

Günter Bornmann

Ulrich Kracker

Paul-Josef Messerig

Während von Paul-Josef wenigstens schon das Anmeldeformular vorliegt, wissen wir von Günter und Ulrich bisher leider recht wenig. Auf jeden Fall sollten alle drei sich, ihre Computer und ihre Interessen demnächst hier im Info vorstellen. Denn nur, wenn man die Interessen und Probleme der anderen Mitglieder kennt, kann man sich Rat und Hilfe holen oder geben! (Klingt zwar wie das "Wort zum Sonntag", ist aber nun mal so.)

In diesem Sinne

always good hacking,

CP/M-Mailbox

die folgende Mail aus der FIDO-Netz-Area "CPM.GER" zeigt, daß auch für Z80- und CP/M-User im Bereich DFÜ was zu holen ist.

Always good hacking, H.O.!

```
Empfaenger : /FIDO/CPM.GER (All)
Absender   : Helmut Jungkunz @ 2:2480/115.11(GRAFFITI BOX Munich, Muenchen)
Betreff    : CP/M BBS lebt! ZNODE 51!
Datum      : Sa 09.07.94, 18:37  (erhalten: 14.07.94) Groesse : 1542 Bytes
--------------------------------------------------------------------------
All,

endlich habe ich es geschafft! Mein ZNODE 51 lebt und kann (vorlaeufig
abends) mal angewaehlt werden.  Einige Files sind schon eingehaengt und per
ZMODEM downloadbar.  Der Rest wird schon.

Geboten wird alle Art von Unterstuetzung von:
CP/M-80    CP/M-86  CP/M68K  und Umsteigern von CP/M auf DOS Infos, Files,
Tools.  Ueber 200 MB Zeugs!  (Komplette SIG/M, CPMUG, C-UG und die gesamte
ZCPR P.D. Software zum gratis-download bereit! Freiwillige Spenden (auch
Disketten und Briefmarken willkommen!

Geplant ist auch ein spezieller Suche/Biete Bereich fuer die oben
erwaehnten Bereiche!  Uebrigens, speziell wegen diverser Terminals
Startmeldung ohne ANSI und ohne IBM-Zeichen.  Persoenliche Konfiguration
moeglich.

--------------------------------------------------------------------------
Name ...............: Helmut Jungkunz Strasse ...........: Zacherlstr. 14
PLZ - Ort ..........: 85737 Ismaning BBS-Phone-number ....: 089/961 45 75
Baud: 300-14400 FAX-phone-number ....:
Online-time in MEZ ..: abends zw. 18 Uhr und 6 Uhr
PVT-Node? ...........: Ja/Point         *falls "point"-status*
Voice-Nummer ........: 089/969374       *nur fuer dringende Rueckfragen*
Voice erreichbar ....: 18:30-20:00
Business Phone Number ..............: 089/961 46 33  CP/M Bestellungen
Erreichbar? ........................: 24 Std Q
--------------------------------------------------------------------------

---Ciao, Helmut Jungkunz
---GoldED 2.40 * Origin: ZNODE #51 - CP/M and ZCPR Support (2:2480/115.11)
```

Gebührenimpulsfilter

Was ist dieser sagenumwobene Gebührenimpuls eigentlich, der anscheinend nur dazu dient, die DFÜ zu stören ? Es handelt sich da-bei um einen Impuls mit 16kHz, der am Anfang jeder neuen Gebühren-einheit gesendet wird und dazu dient, Gebührenzähler, oder besser gesagt Gebührenanzeiger, zu Hause zu steuern. Dieser Impuls ist aber nur eine Kopie des internen Impulses bei der Telekom und mit einem Filter kann man leider keine Gebühren sparen. Wer gar keinen Zähler zu Hause hat, hat den Impuls normalerweise auch nicht beantragt (kostet einmalig 65 Mark, außer man beantragt ihn gleich zusammen mit dem Anschluß, und dann 1 Mark im Monat), aber es ist auch schon vorgekommen, daß man ihn trotzdem auf der Leitung hat. Wer zu Hause einen Zähler hat, hat ihn aber mit Absicht und wird ihn also auch nicht abbestellen und dann kann es zu Problemen bei der DFÜ kommen (muß aber nicht !). Gute Modems lassen sich durch den Impuls gar nicht beeindrucken, bei weniger guten stockt die Verbindung kurz und die noch weniger guten legen teilweise sogar auf. Ich schreibe hier absichtlich nichts von zugelassen oder nicht, das hat damit gar nichts zu tun. Ein zugelassenes Modem wurde nur daraufhin untersucht, daß es die Anlagen der Telekom nicht stört, ob es auch ordentlich funktioniert interessiert dabei keinen. Wer also nun merkt, daß sein Modem im Takt des Gebührenimpulses Probleme macht, der kann sich entweder einen Filter bei der Telekom holen (die alten Gebührenzähler hatten einen eingebauten Filter, die neueren aber nicht mehr), oder weiterlesen und zum Lötkolben greifen. Der Anschluß der Schaltungen an das Netz der Telekom ist natürlich nicht erlaubt, aber vielleicht kann man damit der Ursache der Probleme auf die Spur kommen und sich dann einen zugelassenes Filter holen, oder auch nicht.

Hier also ein paar Schaltungen, die immer mal wieder durch die DFÜ-Welt geistern. Ich habe sie noch nicht ausprobiert, aber sie sollen angeblich recht gut funktionieren. Die Kondensatoren müssen eine ausreichende Spannungsfestigkeit haben, 100V sollte das Minimum sein.


```
           -----------                       -----------
o---x-------! 100 Ohm !-----------x-----! 100 Ohm !-------x---o
    !      -----------            !      -----------      !
    !                             !                       !
    !           !!                !          !!           !
    -----------!!-----x-----------------------!!------------
      100nF !!       !            !          !! 100 nF
                     !            !
                  -------       -----
                  ! 56  !       ----- 200 nF (**)
                  ! Ohm ! (*)     !
                  -------         !
 (vom Telefon)       !            !                (zum Modem)
o--------------------x------------x---------------------------o

     (*): Rechnerisch 50 Ohm, 56 funktioniert bei mir prima
    (**): Falls nicht da, 2*100 parallel
```

```
a o------- Spule von 28mH ------x-------------------o a
                                !
Anschluß               Kondensator 56nF        Modem
                                !
b o-----------------------------x-------------------o b
```

```
a o--------- 130 Ohm ---------x------- 130 Ohm -----------o  a
                              !
                              !
Anschluß                    ----- 47 nF                 Modem
                            -----
                              !
                              !
b o---------------------------x---------------------------o  b
```

```
         24mH                (22mH geht auch)
a o---x---XXXX---x----o a
      !          !
     ---        ---
     --- 33nF   --- 33nF
      !          !
b o---x----------x----o b
```

Zum Schluß noch etwas Theorie, die bei der letzten Schaltung mit dabei war.

VERGLEICH DER SCHALTUNGEN

THEORETISCHE ERWARTUNGEN (habe ich mit einem Simulationsprogramm ermittelt; in den Erklärungen können kleine Schnitzer sein (ist nicht genau meine Fachrichtung) im Prinzip stimmts aber so) Der GI wird mit 16kHz von der Telekom-Vermittlungsstelle zum User-Anschluß geschickt. Der muß VOR dem Modem rausgefiltert werden.

IDEALFALL:

Maximale Dämfung nur bei 16 kHz, bei allen anderen Frequenzen Dämpfung =0 insbesondere von 100Hz..4kHz, da wollen wir ja übertragen. Phasengang nicht dargestellt, wäre hier zuviel

```
dB 0+--------------+  +---
    !              !  !
    !              !__!
-100+--+--+--+-...+--+--+-
    1  2  3  4    15 16 17kHz
```

RC-Filter (Widerstände & Kondensatoren) sind zwar schneller mal so aufgebaut, haben aber gegenüber LC-Filtern (Spulen & Kondensatoren) zwei Nachteile:

1. eine weniger steile Filterkennlinie und
2. beeinflussen im interessierenden Bereich viel stärker den Phasengang.

RC-TIEFPASS

```
dB 0+--_____
    !       -----____
    !                ----
-100+--+--+--+-...+--+--+-
    1  2  3  4    15 16 17kHz
```

Nur wenig steile Kennlinie => erfordert starken Kompromiß zwischen möglichst geringer Dämpung der gebrauchten Frequenzen und hoher Dämpfung der 16 kHz

RC-SPERRFILTER

```
dB 0+-----______
      !          ---__   __-
      !               - -
  -100+--+--+--+-...+--+--+-
      1  2  3  4    15 16 17kHz
```

Etwas besser (steiler), aber noch zu schlecht; und beide RC-Version beeinflussen ganz erheblich den Phasengang und die Gruppenlaufzeit im Übertragungsbereich von 100Hz..4kHz !

Eine Umdimensionierung, Verschiebung der Grenzfrequenz /Sperrfrequenz der RC-Filter ändert an diesen Verhältnissen prinzipiell nichts, bringt keinen Durchbruch.

LC-SPERRFILTER (obige Schaltung)

```
                  __-_
dB 0+---------      !
      !              !
      !               !
  -100+--+--+--+-...+--+--+-
      1  2  3  4    15 16 17kHz
```

Fast keine Dämpung im Übertragungsbereich, sehr hohe Dämpfung bei 16 kHz, keine Beeinflussung des Phasenganges im uns interessierenden Übertragungsbereich. Bei 16kHz dann gewaltig, juckt uns aber nicht mehr

Alexander Schmid

Z 380-CPU

Und gleich noch eine Mail aus der FIDO-Area "CPM.GER". Sie stammt übrigens von einem unserer Mitglieder, von denen übrigens mehrere über FIDO erreichbar und einige auch in der CPM.GER "aktiv" sind.

Always good hacking, H.O.!

```
Empfaenger  : /FIDO/CPM.GER (Alle)
Absender    : Joerg Schimmer @ 2:244/1107.9 (RLL Mailbox, Frankfurt/M)
Betreff     : Z380 mit 32 Bit
Datum       : Fr 19.08.94, 22:13  (erhalten: 23.08.94)
Groesse     : 921 Bytes
-----------------------------------------------------------------------
```

Hallo Alle,

ich bin zwar mit CP/M nicht mehr in Kontakt, ausser in dieser Area hier, aber kurz mal eine Neuigkeit.

Zilog hat nach 17 Jahren Z80 eine neue CPU rausgebracht - den Z380. Vom Z80 wurden letztes Jahr von Zilog immerhin noch 30 Mill. Stück verkauft ohne die Lizenznehmer zu berücksichtigen.

Die Features des neuen Chips:

* 32Bit Adressbus
* 16Bit Datenbus
* Verwaltung bis zu 4 GByte Hauptspeicher :-)
* 5 und 3.3 V Version erhaeltlich
* voll statische CPU
* 4x 32Bit-Registersaetze
* jeweils ein Satz dienst als Arbeitsbereich, die anderen 3 bleiben im Hintergrund.
* Jeder Satz hat ein 8Bit Akkumulator, Flagregister + 6x 8Bit Universalreg.
* max Takt 18Mhz
* volle Abwaertskompatibilitaet zum Z80

Ich freu mich schon auf das 1. 32Bit CP/M !

Gruss,   Joerg

---timEd/2 *Origin:vom Schneider-CPC zu OS/2 bei --> (2:244/1107.9)

Tips und Tricks zu MS-DOS

Beim Lesen der Zeitschrift Computer-Flohmarkt stieβ ich auf ganz ungewöhnliche Möglichkeiten, die MS-DOS bietet. Entfacht wurde die Diskussion vermutlich durch einen User, der mit DEL PRN seinen Drucker zum Verschwinden brachte. Auf diesen Bericht hin, entspann sich ein Erfahrungsaustausch, den ich hier wiedergebe:

Das mit der Explosion des Druckers ist mir vor einem halben Jahr auch passiert. Aber versuchs mal mit UNDELETE PRN. Bei mir konnte ich noch einige Teile des Druckers retten, aber meinen Einzelblatteinzug konnte ich vergessen, der war 100%ig zerstört.

Das mit DEL PRN ist mir auch schon einmal passiert. Nachdem der Rauch verzogen war, erzielte ich jedoch mit UNDELETE PRN gute Ergebnisse.

Um Deinen Drucker wieder zu reparieren, muβt Du nur folgendes eingeben: COPY CON PRN, dann RESTORE PRINTER, dann F6 drücken. Danach müβte alles wieder in Ordnung sein. Wahrscheinlich hast Du vorher probiert, das BIOS des Druckers zu kopieren und danach zu löschen. Damit hat sich dann eine Widerstandsbank ausgeschaltet bzw. ist explodiert. Mit dem RESTORE-Befehl wird auf eine neue, unverbrauchte Bank geschaltet.

Im Falle Deines TRUCKERS bin ich auch ein biβchen ratlos. Nur ein guter Rat von mir: Du solltest eben aufpassen ob Du ein "T" oder ein "D" in die Tastatur drückst. Tja so schnell hat man einen Trucker auf dem Tisch sitzen. Vielleicht versuchst Du mal folgendes: Fertige eine Kopie der Driver.sys und der autoexec.bat an. Diese drückst Du dem Trucker in die Hand, vielleicht fährt er damit davon. Übrigens habe ich herausgefunden, warum die Drucker explodieren. Nach der Eingabe DEL PRN wird im Knall-Register der Wert "PENG" übergeben. Dieser Wert wird ca. 512 Microsekunden lang im Druckerpuffer gespeichert. Das läβt sich nachprüfen, indem man schnell ein Multifunktionsdisplay mit dem A-Ausgang an den Masse-Anschluβ des Puffers und mit dem X-Eingang an den Minuspol der Wechselspannungsquelle (220V) anschraubt. Man hat dann noch Zeit den Schaltvorgang der 640 Relais des druckerinternen Transportschutzes zu beobachten, dann sollte man sich schleunigst Deckung suchen, um nicht von dem Elektronenblitz des rotglühend-rotierenden Postscript-Interpreters überrascht zu werden, der schlieβlich die 256 Kilotonnen Druckerschwärze zur Explosion bringt. Dies ist natürlich nicht allgemein auf alle Druckertypen anwendbar, deshalb kann ich für die gefundene Erklärung keine Garantie abgeben.

Was soll fehlerhaft sein an DEL PRN? Der Compi hat nur das gemacht, was Du von ihm verlangt hast.Er hat den Drucker gelöscht, indem er ihn explodieren lieβ. Gab es denn keine Sicherheitsabfrage? Ich empfehle Dir, Dich auf dem Markt einmal nach UNDELETE Progs umzusehen.Damit kannst Du gelöschte Drucker wiederherstellen wie eine Datei. Ich hab mir eine Betriebssystemfunktion zunutze gemacht: XCOPY PRN PRN, das ergibt ein genaues Duplikat eines vorhandenen Druckers und damit steige ich in den Versandhandel ein. Wo ich die Verpackung für die vielen Drucker hernehme? Ich gebe ein: ARJ PRN. Da wird der Drucker verpackt.

Wenn ich nun von Dir so einen gepackten Drucker bekomme, muβ ich den dann mit X oder E entpacken? Noch eine Frage: Kann man wohl mit EDIT PRN die Gehäusefarbe wechseln oder wie würde ein Drucker funktionieren, der mit einem Online-Packer behandelt wurde?

Du willst noch mehr sensationelle DOS-Funktionen? Wie wäre es mit COPY *.* PRN als Backup-Medium? Da wird aber das Zurücklesen zu einer elenden Tipperei.

Falls Du keinen Scanner hast, probier doch einfach mal COPY PRN>DATEI.GIF. Du kannst damit sogar in Farbe scannen, wenn Du eine farbige Vorlage in den Drucker legst. Das funktioniert aber nur, wenn Du einen Farbdrucker hast. Wenn dein Bildschirm dreckig ist, gibst Du einfach etwas Putzmittel ins Laufwerk und tippst CLS<A:

Und wenn ich den Drucker reinigen möchte, gebe ich einfach etwas Putzmittel auf die Mattscheibe des Monitors und drücke die Taste "DRUCK", oder wie?

In der nächsten Info werde ich euch berichten, wie man mit einem kleinem Bohrer seine Hardware kostengünstig aufrüsten kann.

Georg Bielenberg

Impressum

1. Vorsitzender	Hartmut Obermann	Tel.: 0721/85 40 68
	Wilhelm-Baur-Straße 8	Fax: 0721/85 40 68
76135	Karlsruhe	
2. Vorsitzender	Gerald Schröder	Tel.: 04105 /26 02
	Am Schützenplatz 14	
21218	Seevetal	
Hardewarekoordinator	Andreas Magnus	Tel.: 0209 /87 02 30
	Bismarckstraße 29	
45879	Gelsenkirchen	
NewDOS-Diskothekar	Oliver Volz	Tel.: 0711/74 40 51
	Am Ochsenwald 37A	
70565	Stuttgart (Rohrerhöhe)	
CP/M-Diskothekar	Fritz Chwolka	Tel.: 02464 /89 20
	Saarstraße 34	
52457	Aldenhoven	
C-128-Diskothekar	Günther W. Braun	
	Postfach 80 02 26	
81602	München	
Clubbücherei	Rainer Schmitz	Tel.: 07166 /13 97
	Küferweg 12/1	
73099	Adelberg	
Redaktion	Jens Neueder	Tel.: 0791 /4 28 77
	Rudolf-Then Straße 32	FAX:07971 /2 50 55
74544	Michelbach /Bilz	

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H., 76135 Karlsruhe
Konto Nr. 496 071-605
Postgiroamt Frankfurt, BLZ 500 100 60

Autoren Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.
Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Schluß

Hallo Club 80'er,

es ist, wie alle drei Monate, so auch jetzt im März, wieder einmal INFO-Zeit. Zum Glück war es diesmal nicht viel, sodaß ich es wenigstens noch vor dem Clubtreffen schaffen konnte. Diesen Monat hatte ich wirklich alles Mögliche -bloß keine Zeit. Berufliche und private Termine wechselten sich ab. Dazwischen gab es dann auch noch ein paar Ausfalltage zum Zwecke der Nachsynchronisation. Mein persönlicher Vierenscanner war wohl mal nicht ganz **Up to date**. Somit hat sich der von Hartmut anviesierte INFO-Erscheinungstermin total überholt.

Wobei wir nun auch schon bei dem nächsten, unserem wichtigsten Jahrestermin, dem **Clubtreffen 95** wären:
Wir würden uns freuen, wenn alle Clubmitglieder einmal am Clubtreffen teilnehmen könnten. Aus diesem Grund und als Terminerinnerung, liegen diesem Info nochmals die Anreisehinweise bei. Vielleicht kann sich der ein oder andere doch noch zu einem Kurzbesuch entschließen. Der Samstag, an dem auch die Jahreshauptversammlung stattfinden soll, ist sicher ein guter Zeitpunkt dafür. Vielleicht läßt sich auch kurzfristig noch etwas mit Zimmern managen. Dazu wäre aber vorher eine Rücksprache mit Alexander (ob sich da auf die Schnelle noch was organisieren läßt) sinnvoll.

Bis zum Clubtreffen oder zum nächsten Info
die besten Wünsche Euer

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

1996

1995

CLUBINFO

48. 49. 50. Ausgabe

Kontaktadresse:
Club 80
Fritz Chwolka
Saarstrasse 34
5173 Aldenhoven
Tel.: 02464/8920

Inhaltsverzeichnis

Autor & Seite

Clubinternes

Neues vom Vorstand 1
Hartmut Obermann

Protokoll Hauptversammlung 95 1 - 2
Alexander Schmid

Termine, INFO´s 2
Redaktion

Vorstellungen 3 - 4
Staimer Ulrich, Günter Bornmann

Soft & Hard

8-Zoll-Schrumpfkur 5
Alexander Schmid

MS-DOS-Start-UP 6 - 9
Heiko Werner

Dampf für die serielle Schnittstelle 10
Inhouse-Modem für RS-232 10
Die aktuellen Baustellen 11 - 11
Andreas Kleespies

Sonstiges

Zu verkaufen: 9
Jörg Brans

Ich suche.. - 12
Wolfgang Berndt

Netiquette, -Knigge für Datenreisende 13
Martin Goldmann

Die Kleincomputer (KC) aus Mühlhausen 14 - 15
Bericht von den Z-Festen 15 - 16
Star Trek - Die nächste Version 16 - 17
Jörg Linder & Mailboxen

Die letzten Seiten

Impressum 18
Schluß 18
Redaktion

Mitgliederadressenliste am INFO-Ende
AT Kommandos am INFO-Ende

Neues vom Vorstand

Liebe CLUB 80-Mitglieder,
zwischen dem letzten und diesem Info ist mal wieder eine (viel zu) lange Zeit vergangen. Normalerweise würde ich jetzt wieder einen Rüffel an die ach so passiven Mitglieder verteilen, diesmal geht der Anschiss aber vor allem an mich selbst.

Bedingt durch mehrere Lehrgänge und Dienstreisen, aber auch durch einigen privaten Streß mit meiner Mutter und Schwiegermutter, die beide ein künstliches Hüftgelenk bekommen bzw. gerade bekommen haben, habe ich den CLUB 80 etwas in den Hintergrund geschoben und bin meinen Verpflichtungen nur unzureichend nachgekommen.

Und selbst jetzt wo ich mich, angetrieben durch einen mahnenden Anruf von Jens Neueder, endlich dazu aufgerafft habe, ein Vonvort zum neuen Info zu schreiben, ist der Club nur Nebensache. Ich. bin nämlich gerade mal wieder dabei meine Sachen zu packen und aus dienstlichen Gründen für drei Wochen meine Familie zu verlassen.

Ganz abgesehen von diesen privaten und dienstlichen Gründen gibt es aber noch einen Grund, warum der CLUB 80 immer weiter in den Hintergrund meiner Aktivitäten gedrängt wird. Mein Model 4p steht inzwischen nämlich seit über zwei Jahren unbenutzt auf dem Schrank herum, und selbst beim letzten Treffen wurde es noch nicht einmal ausgepackt und in Betrieb genommen. Seinen Platz auf meinem Schreibtisch hat ja schon seit Jahren ein (und teilweise mehrere) PC eingenommen, und seit ich nun auch in Karlsruhe wieder ein angemeldetes EDVBüro betreibe, bleibt endgültig keine Zeit mehr für den früher so extensiv genutzten CP/M-Schlepptop.

Und wenn die Computerei so langsam vom Hobby zum Beruf wird, braucht man natürlich wieder etwas, was man nur so als Hobby betreibt. Für mich ist das mein Motorrad und die damit gefahrenen Touren durch halb Europa. Und natürlich hat auch dieses Hobby inzwischen einen größeren Stellenwert als der CLUB 80 - so leid es mir tut.

Vielleicht wäre das ja anders, wenn sich im CLUB 80 "etwas tun würde", aber dem ist leider nicht so. Jens berichtet wieder von einem akuten Mangel an Artikeln fur das neue Info und auch sonst tut sich herzlich wenig.

Genug gejammert und auf die Tränendrüse gedrückt!

Ich wünsche euch allen viel Spaß bei der Lektüre des aktuellen Infos,
einen guten Rutsch in das Jahr 1996 sowie geruhsame Feiertage.

Protokoll der Hauptversammlung des Club 80 am 15.April 1995

Protokollführer: Rainer Schmitz

anwesend waren: Jens Neueder
Horst Dieter Schroers
Alexander Schmid
Jürgen Kemmer
Fritz Chwolka
Helmut Bernhardt
Andreas Kleespies
Michael Kalb
Manfred Held
Rainer Schmitz
Hartmut Obermann

1. Vorstandswahl:

H.O. erklärt sich zu einer weiteren Amtszeit bereit.

Oliver Volz (NEWDOS-Archivar) ist bereit, seinen Posten zur Verfügung zu stellen, wenn sich ein Kandidat findet. Da dies nicht der Fall war, behält er seinen Posten auch weiterhin.

Andreas Kleespieß erklärte sich bereit, dem Hardware-Koordinator Andreas Magnus zu unterstützen. Andreas klärt die genauen Modalitäten telefonisch mit A. Magnus, das Ergebniss erfahren wir dann aus einem der nächsten Infos.

Da sich auch für den Posten, des stellvertretenden Vorsitzenden kein anderer Kanidat fand, wurde auch Gerald Schröder in seinem Amt bestätigt.

Der Rest des Vorstandes bleibt ebenfalls unverändert.

2. Mitgliedsbeitrag:

Es wurde festgestellt, daß der Bargeldbestand für etwa 7 Jahre ausreicht, und daher bis auf weiteres auf einen Mitgliedsbeitrag verzichtet werden kann.

3. Neues Clublogo:

Der Wettbewerb für ein neues CLUB 80-Logo brachte zwei Vorschläge, die, wie beim letzten Treffen festgelegt, heuer zur Begutachtung vorlagen.

Zur Abstimmung für das neue Clublogo standen zwei neue Vorschläge sowie das alte Logo.

Die Abstimmung ergab:

5 für das alte Logo
3 für das Bielenberg-Logo
3 für das LMder-Logo

Daraus folgt, daß das alte Logo beibehalten wird. Das als Preis ausgesetzte Modem bleibt Clubeigentum

4. Kassenprüfung:

Es wurde beschlossen, Jutta Obermann auf Rechnung der Clubkasse zum Essen einzuladen, um ihr für die einwandfreie Buchführung der Clubkasse zu danken.

5. Organisation der Clubtreffen

(Regional-)Treffen sollen weiterhin stattfinden, wenn ein Organisator gefunden wird. Alternativ wurde angeregt, am geplanten 8-Bit-Netz teilzunehmen. Fritz Chwolka kümmert sich um dieses Netz.

6. Investitionen

Der Club beschließt den Kauf eines Telefonsimulators von Helmut Bernhardt für 25 DM um Anpassungen von Modemprogrammen auf CP/M-Rechnern zu erleichtern. Das Gerät befindet sich z.Z. bei Rainer Schmitz und kann dort ausgeliehen werden.

Termine 96.. Termine 96.. Termine 96.. Termine 96.. Termine

Inland		
13.01.-21.01.	Stuttgart	CMT Internationale Ausstellung für Caravan, Motor, Touristik
18.01.-21.01.	Hamburg	HCT Hamburger Computertage
01.02.-07.02.	Nürnberg	Internationale Spielwarenmesse mit Fachmesse Modellbau, Hobby und Basteln
06.02.-09.02.	Hamburg	ONLINE Europäische Congressmesse für Technische Kommunikation
14.03.-20.03.	Hannover	CeBIT Hannover Welt-Centrum Büro- Information- Telekommunikation
17.04.-21.04.	Dortmund	COMPUTER-SCHAU Ausstellung für Computer, Software und Zubehör
17.04.-21.04.	Dortmund	Hobby-tronic Ausstellung für Funk- und Hobby-Elektronik
07.05.-10.05.	Stuttgart	telematika Internationale Messe für Telekommunikation
11.06.-14.06.	Stuttgart	CAT Internationale Fachmesse für Computer in Planung, Konstruktion und Fertigung
26.06.-28.06.	München	ELTEC Fachmesse für Elektrotechnik
18.09.-23.09	Köln	photokina Weltmesse Bild - Ton - Professional Media
12.11.-15.11	München	electronica Internationale Fachmesse

Ausland		
24.03.-26.03.	London	ECTS Europäische Computer Fachmesse
16.04.-18.04.	Stockholm	PC Wold Expo
Mai	Minsk	Automation Internationale Fachmesse für Automation, Elektrotechnik, Computer, Meßtechnik, Präzisionsgeräte
04.06.-07.06	Oslo	E & A Ausstellung für Elektronik und Automation
04.06.-07.06	Turin	INTELCOM Ausstellung für Telekommunikation
10.06.-14.06.	Birmingham	ElectroTech Internationale Ausstellung für Elektrotechnik und Elektronik
08.09.-10.09.	London	ECTS Europäische Computer-Fachmesse
22.10.-25.10.	Warschau	CONTROLA Meßtechnik, Prüftechnik, Computer
05.11.-10.11.	Madrid	SIMO TCI Internationale Fachmesse für Datenverarbeitung, Multimedia und Kommunikation

Redaktionsschluß für das nächste Clubinfo ist der

01. März 1996.

Staimer Ulrich
Dekan-Mayer-Straße 15
86199 Augsburg
0821/993104 (nur selten erreichbar, bitte von 20Uhr bis 23.30Uhr öfters mal probieren)

Vorstellung

Hallo Ihr Benutzer des besseren Betriebssystems! Nachdem fast ein Jahr meiner Mitwirkung (?) beim CLUB 80 vorüber ist habe ich mich zur Vorstellung in der Clubzeitung überreden lassen. Auf den CLUB 80 wurde ich durch Alexander Schmid aufmerksam. Alexander half und hilft mir bei manchen Schwierigkeiten mit meiner Z280 CPU (Handmade bei Tilmann Reh). Angefangen hat bei mir das Geldausgeben jedoch nicht erst mit dem Z280-Rechner. Der genaue Zeitpunkt meiner Leidenschaft hat ihren Ursprung irgendwo in den Jahren 1983/84. Mein Traum war damals ein IBM-Junior-Pc (Kostenpunkt damals so um die 5.500.-DM). Sehr viel bescheidener kam dann ein C64 mit Datasette her (Anschluß am Fernseher, mit Datasette und einem Spiel rund 850,-DM). Neidisch blickte ich auf die ganzen anderen Kisten, die - teilweise rund 2 bis 3 mal so teuer - mit CP/M arbeiteten. Als dann der Commodore C128 mit eingebauter Floppy rauskam schlug ich zu - mit Beziehung und dadurch Personalrabatt ergatterte ich ein Ausstellungsstück - 1.450,-DM billig. Und das war dann der Anfang von der Dauerebbe in meinem Geldbeutel. Denn Mann brauchte dazu ja dann unbedingt eine zweite Floppy (5-1/4 Zoll), eine dritte Floppy (3-1/2 Zoll), eine vierte Floppy, eine fünfte Floppy.... achja und irgendwann dazwischen einen Monitor, dbaseII, Multiplan, und und und. Und dann lernte ich Helmut Jungkunz kennen, wurde Mitglied in der SCUG München, und irgendwann wurde einem die Kiste natürlich zu langsam, als Diskjockey war man auch immer am suchen - und immerhin und außerdem und irgendwie .... Da war dann die Zeit reif für die Z280-CPU. Und dann geschahen ganz seltsame Dinge. Bekannte und Verwandte usw., alle hatten plötzlich Z80 Kompis zu verschenken, sehr günstig abzugeben usw. Und außerdem muß ja unbedingt doch mal so eine DOSe her. Seither steh ich (da noch zu haben) regelmäßig mit meiner Mutter im Clinch, deren Lieblingssatz ist "Verschenk doch das Zeug! Du kannst doch eh nur einen Computer auf einmal bedienen!!"

Worauf ich natürlich heftig widersprechen MUSS! Denn wer trennt sich schon gerne von einem Commodore C128 mit Monitor, 1x5-1/4Zoll-Lw, 4x3-1/2Zoll-LW, 512KB-RAM, einem mc-Selbstbau Z80 mit 2x8Zoll-LW und Siemens-Drucker PT80, einem MTX-512, Leider ohne Zubehör, einem Altos-380, MP/M Betriebssystem, einer Reh-Z280-CPU mit allem drum und dran, 5 (fünf) IBM PS/2 XT's mit 8086 Prozessor, (VIER SEHR GÜNSTIG ABZUGEBEN), 3 IBM PS/2 386slc20MHz, einem Siemens 486sx25 (wird derzeit - wenn Geld vorhanden - hochgerüstet), einem Compaq Deskpro 386sn20 (wird derzeit - wenn Geld vorhanden - umgebaut), einem HP-Monochrom-Terminal (9600baud), zwei HP_Farb-Terminals (9600baud), sowie für - fast - alle Rechner einen Laserdrucker OP105A von Seikosha und einen Neunnadler Panasonic KX-P1081.

Mal ehrlich: WER kann sich von solch einer tollen Sammlung schon trennen?

Und ganz ehrlich: Wenns schnell gehen soll - ran an den nächstbesten Z80-Compi und WS3 oder WS4 oder ZDE gestartet und los geht's (so wie jetzt gerade). Allen Usern virenfreie Compis wünscht

Uli

PS: Hätte ich fast vergessen: meine Erkennungszeichen sind Bauch und Hosenträger.

Hallo Clubmitglieder!

Zunächst einmal möchte ich sagen, daß ich aus der Commodore-Ecke komme und meine Erfahrungen mit Z80-CP/M Computern mit dem C128 gemacht habe (dßaseII-Anwendung und etwas Assembler). Meine allerersten Aktivitäten habe ich allerdings mit einem Sinclair Spectrum gemacht, wovon ich noch zwei Exemplare besitze aber (bis auf weiteres) nicht mehr benutze. Ansonsten habe ich C128 und C64 plus Peripherie (auch die modernere) mehrfach. Da man ja Commodore Computer nicht mehr im Geschäft bekommt, braucht man schließlich Reserven. Weitere Einzelheiten möchte ich hier nicht ausbreiten, wer aber Fragen zum Thema Commodore hat, kann mich gerne ansprechen,

Mein Interesse am Club 80 erwächst hauptsächlich aus dem Wunsch, Kontakt zu halten zu dem zwangsläufig schrumpfenden Kreis der 8-Bit-Computer Anwender und der Erwartung, so an Informationen zu kommen, die um die eine oder andere Problemecke herumhelfen. Der PC-Anwender, der mal eben ein kleines Utility braucht, kann in einem Berg von Software suchen. Die 8-Bit-Computer werden kommerziell nicht mehr bedient und da kann ein kleiner Tip von einem "Leidensgenossen" die entscheidende Hilfe bringen.

Damit wäre ich auch schon bei meinem Hauptproblem. Als berufstätiger Familienvater kann ich nur stundenweise und unregelmäßig meinem Hobby nachgehen. Umfangreichere Programmentwicklungen sind daher für mich nicht realisierbar. Trotzdem, wenn jemand Aktivitäten auf dem Sektor Hardware oder maschinennahe Software hat oder plant, kann er mich gerne ansprechen.

In diesem Zusammenhang noch kurz meine Interessenschwerpunkte: Im Prinzip komme ich von der elektronischen Hardware her und habe mich lange nur mit analoger Elektronik befaßt. Als ich ins Berufsleben eintrat, war gerade die Phase wo die elektronischen Steuerungen von analog auf Microcontroller umgestellt worden. Nach einiger Zeit fing ich dann an, mich auch privat mit Computern zu befassen und der 64er bzw. 128er erwies sich dabei als ideale "Bastelkiste". Beruflich arbeite ich heute mit PC, unter Windows und Netzwerkumgebung, aber als reiner Anwender (Word, Excel, etc.), Da bei den PCs im Schnitt alle 2-3 Jahre eine neue Hardwaregeneration rauskommt, habe ich gar nicht erst versucht, da die Nase tiefer rein zustecken - ein Grund mehr, den 8-Bit-Computern die Treue zu halten. Mit DFÜ habe ich mich übrigens noch nicht befaßt, sieht man davon ab, daß ich seit 4 Jahren BTX benutze.

So weit fürs erste. Das diesjährige Clubtreffen kam mir noch etwas früh, das nächste werde ich mal einplanen.

Viele Grüße

Günter

Günter Bornmann

8-Zoll-Schrumpfkur

Kleine Scheiben für große Formate

So wie mir dürfte es wohl so manchem CP/M-Freak mit einer älteren Maschine ergangen sein: die vorhandene Floppyschnittstelle für die 8-Zoll-Laufwerke wurde mangels Hardware einfach ignoriert, der 50-polige Steckverbinder blieb unbelegt. Die recht hohe Kapazität der "großen" Formate erschien mir aber verlockend, zumal sie im BIOS bereits eingebunden waren. 1,2 MB auf einer -inzwischen recht billigen- Diskette sind durchaus geeignet, das Nachdenken hinsichtlich einer Nutzung anzuregen...

Das Projekt bekam neuen Schwung, als ich bei einem bekannten Münchner Elektronikschrottladen zwei unbenutzte TEAC-Laufwerke im Laptop-Flunder-Design entdeckte, deren Preisschilder "Nimm mich mit!" riefen. Ich verließ den Laden mit einer etwas schwereren Tasche und einem etwas leichteren Geldbeutel. Zuhause angekommen, wurde der Fang zwecks Verheiratung mit meinem Wavemate Super Bullet näher untersucht. Alles wie gehabt: 34-polige Pfostenleiste und ein paar Jumper für Drive-Selects und Ready etc. Kein Problem also für Django mit dem schnellen Lötkolben!

Ich hatte die Inbetriebnahme als Laufwerk C: und D: ins Auge gefaßt, also als drittes und viertes DS-DD-Laufwerk mit 700k Kapazität. Der Anschluß verlief reibungslos, das Lesen der Disketten auch, nur das Formatieren nicht: Sektorfehler noch und nöcher! Also doch: Sonderangebot, Umtausch ausgeschlossen, Entsorgung bezahlt der Käufer??

Wie gut, wenn man das Inhaltsverzeichnis der c't im Computer und fast alle Hefte im Schrank hat. Ein kurzer Suchlauf zum Thema "Floppy" förderte die Information zu Tage, daß es früher, in der mittleren Computer-Jungsteinzeit HD-Laufwerke ohne HD-Loch-Abtast-Schalter gab, die bei DD einen L-Pegel an Pin 2 zwecks Formatumschaltung erwarten. Her mit dem Lötkolben! Und siehe da, das Laufwerk formatiert auf einmal ordentlich -bis die Sektorfehler bei Tracks größer 70 wieder auftauchten. So'n Schiet... Spaßeshalber steckte ich eine HD-Diskette in das Laufwerk, und es klappte! Ich hatte also einen Pyrrhussieg errungen: billige Laufwerke, aber teure Disketten. Immerhin war der Beweis erbracht, daß ich billig an HD-Laufwerke gekommen war.

Leider wollte auch mein PC wenig von diesen Laufwerken wissen, da er an Pin 2 einen Logikpegel erwartete, der ihm die eingelegte Floppysorte anzeigt. Da besann ich mich eines Hauptplatinen-Anschlusses im eleganten Tausendfüßlerdesign, der meinen Wavemate zierte (zu was anderem war er bisher nicht zu gebrauchen). Ihn hatte ich immer nur mit "Drive E: not ready" und Disketten im LP-Format (wer alte Computer sammelt, muß auch alte Tonträger kennen!) assoziiert. Meine technischen Unterlagen für den Wavemate taten mir aber kund, daß das erfolgreiche Bestehen der Anschlußprüfung zwar keine Prinzessin, dafür aber das 1.2-MB-Königreich als Belohnung bringt. Nun denn, es sei!

Diese Herausforderung mußte angenommen werden. Die besagten c't-Artikel förderten eine Anschlußbelegung der diversen Floppygrößen zu Tage, so daß ich einen Adapter von 50-polig (8 Zoll) auf 34-polig (3.5 Zoll) entwickeln konnte. Nach Fertigstellung des Adapter gab es wieder eine Enttäuschung: die Motoren der Floppies liefen nur an, wenn ich eine Diskette einlegte. Da 8-Zoll-Laufwerke offensichtlich kein Motor-On Signal haben, hatte ich das Motor-On Signal des HD-Laufwerks auf das
Head-Load Signal gelegt. Nun gab es aber ein Problem: Head-Load wird nur aktiv, wenn das Laufwerk "Ready" ist, und dieses Signal wird aktiv, wenn der Motor mit Nenndrehzahl läuft. Dies konnte ich leicht mit einer Brücke zwischen Masse und Motor On nachprüfen. Der letzte Schritt zum perfekten Zusammenspiel war also das Einlöten einer Schottky-Diode (Germanium tuts auch, entscheidend ist eine Durchlaßspannung von wenigen 100 mV, um den L-Pegel sicher unter 0,5 V zu bringen) zwischen Motor On (5 Zoll, hier Kathode) und Head Load (8 Zoll, hier Anode). Jetzt schnurrt die alte Säge!

Wer ähnliches vorhat, findet im Anhang die Belegung des Adapters. Da die meisten 8-Zoll-Laufwerke nur 77 Spuren haben, kann der versierte BIOS-Hacker noch drei Spuren und einige K an Kapazität herausholen. Wenn mir jemand dann sagt, wie's geht, probier ich es auch. Ich hatte nur keine Lust zu diesem Hack, es hat für mich was von "mit der Stange im Nebel herumstochern", und 1,2 MB auf inzwischen spottbilligen Scheiben (momentan bei Porst: zehn formatierte HD-Scheiben 3.5 Zoll für 6,35 - das zahlt man ja bald für eine 360k 3-Zoll-Scheibe!!!) mit "nur" 77 Spuren reichen ja auch.

Hallo liebe Club80 Mitglieder,

schon zweimal habe ich mich als Z80 User abgemeldet. Aus Verzweiflung mit niemandem Kontakt bekommen zu haben. Um mich herum waren alle auf PC's umgestiegen. So bin ich trotz gemunkel und geunke einer der Langweiler, der es nur mit einem PC tut.
"Noch ?" gehöre ich auch deshalb zu den 90%, die sich nicht melden. Meine Hardwarebastelleien haben 1990 aufgehört und meine Programmierkenntnisse sind mager. Als weiterer Minuspunkt waren die Kosten der Telekom. Nach anfänglichen geglückten Kontakten mit Mailboxen, habe ich meiner Frau das Alibi für tägliche Anrufe verschafft. Nun nach zwei Jahren Kampf ist meine Telefonrechnung wieder als akzeptabel einzustufen.
Ihr seht, ich passe eigentlich gar nicht in den Club. Da ich die Clubzeitung gut finde (mein Dankeschön an die Schreiber), möchte ich mich heute vorstellen und meine Daten etwas aktualisieren. Ich bin 35 Jahre alt und habe zwei Kinder die mich immer weniger an den PC lassen. Nun meine Ausstattung ist nicht ganz übel. Ich arbeite mit einem 486'er mit 8 MB Speicher, habe fünf kleine Festplatten angeschlossen, einem Flachbettscanner, einem Faxmodem mit 14400 Baud, 500 MB Streamer, Graphiktablett , eine Soundblasterkarte und ein SyQuest Laufwerk . Dann habe ich noch einen 486 Notebook.

Nun da die Software immer anspruchsvoller (an Speicher !! - nicht Leistung) wird, möchte ich mein Startbild vorrtellen . So sehen meine Autoexec.BAT und Config.SYS mit MS-DOS 6.2 aus (Ich habe so etwas auch in DRDOS 6.0):

```
@ECHO OFF
if "%WINFAX%"=='ja" goto WINFAX
if "%streambk%"=="ja'' goto STREAMBK
if"%DBASE%"=="ja" goto DBASE
if "%DOS%"=="ja" goto DOS
if "%HIMEM%"=='ja" goto HIMEM
if "%SPIELE%"=="ja" goto SPIEL
if "%MAXSOUND%"=="ja" goto MAXSOUND
goto end
```

```
:WINFAX
PROMPT [$D] [$T]$_[WINFAX] $p$g
PATH C:\DOS;C:\WINDOWS;C:\DATAFAX
SET TEMP=G:\temp
C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
C:\DOS\SMARTDRV.EXE /X
F:\MOUSE\mouse.exe
cd windows
ren autostar.grp autastar.001
ren autostar.fax autostar.grp
cd windows
ren autostar.grp autostar.fax
ren autostar.001 autostar.grp
goto end

:STREAMBK
PROMPT [$D] [$T]$_[STREANIßK] $p$g
PATH F:\SyQuest;C:\DOS;F:\TMOUSE;F:\MG;F:\BACKIT4;F:\MENU
C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
F:\BACKIT4\BKSCHED.COM
SET BACKLT=F:\BACKIT4
SET BKOUT=F:\REPORT
SET TEMP=G:\temp
F:\MOUSE\mouse.exe
bk4
goto end

:DBASE
PROMPT [$D] [$T]$_[dBASE] $p$g
PATH F:\SyQuest;C:\DOS;F:\TMOUSE;F:\MG;C:\DBASE
C:\DOS\SHARE.EXE /L:500 /F:5100
SET TEMP=G:\temp
C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
F:\MOUSE\mouse.exe
REM MEMMAKER dbase
goto end

:HIMEM
PROMPT [$D] [$T]$_[WIN] $p$g
PATH F:\SyQuest;C:\DOS;F:\MG;C:\WINDOWS;F:\ODAPI;F:\IDAPI
C:\DOS\SHARE.EXE /L:500 /F:5100
SET TEMP=G:\temp
C:\DOS\KEYB GR,437,C:\DOSWYBOARD.SYS
C:\DOS\SMARTDRV.EXE /X 1024 128
F:\MOUSE\mouse.exe
win :
goto end
```

```
:DOS
PROMPT [$D] [$T]$_[MSDOS 6.0] $p$g
PATH F:\SyQuest;C:\DOS;F:\TMOUSE;F:\MG;F:\MENU;F:\BACKIT4
SET LMOUSE=F:\MOUSE
C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
SET TEMP=G:\temp
F:\BACKIT4\BKSCHED.COM
SET BACKIT=F:\BACKIT4
SET BKOUT=F:\REPORT
doskey
F:\MOUSE\mouse.exe
prog.bat
goto end

:SPIEL
PROMPT [$D] [$T]$_[SPIELEMS] %p$g
PATH F:\SyQuest;C:\DOS;F:\TMOUSE;F:\MG;F:\MENU
SET SOUND=C:\SBPRO
SET BLASTER=A220 I2 D1 T4
SET MIDI=SYNTH:1 MAP:E
C:\SBPRO\SBPSET /P /Q
SET LMOUSE=F:\MOUSE
C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
SET TEMP=G:\temp
F:\MOUSE\mouse.exe
f:\menu\spiele
goto end

:MAXSOUND
PROMPT [$D] [$T]$_[SPIELOEMS] $p$g
PATH F:\SyOuest;C:\DOS;F:\TMOUSE;F:\MG;F:\MENU
SET SOUND=C:\SBPRO
SET BLASTER=A220 I2 D1 T4
SET MIDI=SYNTH:1 MAP:E
C:\SBPRO\SBPSET /P /Q
SET LMOUSE=F:\MOUSE
LH /L:2,17184 C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS
SET TEMP=G:\temp
LH /L:0;2,26944 /S F:\MOUSE\mouse.exe
f:\menu\spiele
goto end

:end
```

Config.sys

```
[MENU]
menuitem=winfax, Windows mit Fax starten
menuitem=himem, nur Windows
menuitem=stmbk, Streamer Backit4
menuitem=dbase, dßase IV starten
menuitem=dos, nur DOS laden
menuitem=spiele, Maximaler Speicher mit Sound
menuitem=maxsound. Ohne EMS mit Sound
menudefault=winfax,3
menucolor=14,1

[winfax]
DEVICE=C:\DOS\HIMEM.SYS
DEVIC E=C:\DO S\EMM386. EXE NOEMS X=d400-d800 X= D700-D7FF WIN=D000-D3FF
DEVICE=C:\DOS\SETVER.EXE
DOS=HIGH
STACKS=32,128
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
COUNTRY=049,437, C:\DO S\COUNTRY.SYS
SET WINFAX=ja

[himem]
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS X=d400-d800 X=D700-D7FF WIN=D000-D3FF
DEVICEHIGH=F:\SYQUEST\IDE\SQATDRVR.SYS
DEVICEHIGH=C:\MSCSI.SYS
DEVICEH IGX=C:\DOS\SETVER. EXE
DOS=HIGH
STACKS=32,128
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
SET HIMEM=ja

[stmbk]
DEVICE-C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS X=d400-d800 X=D700-D7FF WIN=D000-D3FF
DEVICE=F:\SYQUEST\IDE\SQATDRVR.SYS
DEVICE=C:\DOS\SETVER.EXE
DOS-=HIGH
STACKS=32, 128
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
COUNTRY=049,437, C:\DOS\COUNTRY.SYS
DEVICE=F:\BACKIT4\FPASPI.SYS /RO /WO
SET STREAMBK=ja
```

```
[dbase]
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS X=d40Q-d800 X=D700-D7FF WIN=D000-D3FF
DEVICE=F:\SYQUEST\IDE\SQATDRVR.SYS
DEVICE=C:\DOS\SETVER. EXE
DOS=HIGH
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
STACKS=32,128
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
SET DBASE=ja

[dos]
DEVICE=C:\DOS\H1MEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS X=d400-d800 X=D700-D7FF WIN=D000-03FF
DEVICE=F:\SYQUEST\IDE\SQATDRVR.SYS
DEVICE=C:\DOS\SETVER.EXE
DOS=High
DEVICE-C:\DOS\DBLSPACE.SYS /MOVE
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
DEVICE=F:\BACKIT4\FPASPI.SYS /RO /WO
DEVICE=C:\DOS\ANSI.SYS
SET DOS=ja

[spiele]
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM X=d400-d800 X=D700-D7FF
DEVICE=F:\SYQUEST\IDE\SQATDRVR.SYS
DEVICE=C :\DOS\SETVER.EXE
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
DOS-High
SET SPIELE=ja

[maxsound]
DE VICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS HIGHSCAN X=D900-DEFF
DEVICEHIGH /L:1,18720=F:\SYQUEST\IDE\SQATDRVR.SYS
DEVICEHIGH /L:2,12240=C:\DOS\SETVER.EXE
DEVICEHIGH /L:3,39392=C:\DOS\DBLSPACE.SYS /MOVE
BUFFERS=15,0
FILES-8
DOS=UMB
LASTDRIVE=L
FCBS=4,0
COUNTRY=049,437,C:\DOS\COUNTRY.SYS
DOS=High
SET MAXSOUND=ja
```

Zur Erläuterung:

Ich stand vor folgendem Problem: Der Speicher sollte effektiv verwaltet werden und mein Faxmodem automatisch ankommende Fax entgegen nehmen. Dazu habe ich eine Stand By Automatik und ein Faxstar gekauft. Über mein Anfangsmenü habe ich mit "menudefault=winfax,3“ den Pfad vorgegeben der nach 3 Sekunden automatisch Faxsoftware startet. Die Umbenennung der Autostartgruppe lädt automatisch die entsprechenden Treiber. Über ein Zählmodul, welches ich in den Rechner eingebaut habe (ein bißchen Lötkolben ist doch noch), lasse ich die Anzahl der erfolgten Rechnerstarts erfassen. Dient mir einmal zur Kontrolle wie oft meine Kinder heimlich den PC angemacht haben (ich kann's zwar nicht ändern) und ich brauche nicht immer den Rechner starten, um nach zu sehen ob ein Fax angekommenen ist.

Da Windows keine Treiber benötigt, habe ich hier Sound - und andere Treiber heraus genommen. Meiner Erfahrung nach, ist es nicht so günstig, den Path unnotig zu verlängern. Besser ist es über eine extra geschriebene bat Datei die entsprechenden Treiber zu laden. Z.B. CD ROM - wird sich hoffentlich nicht durchsetzen - benötige ich kaum, also kann ich den Einsatz im Anfangsmenü planen.

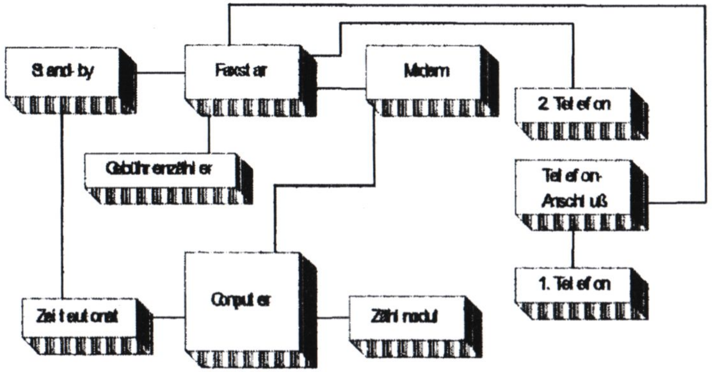
Nun ich weiß nicht ob meine Beispiele uninteressant sind, aber ich könnte noch Hinweise brauchen. Ich stehe nach wie vor, vor folgendem Problem. Wenn jemand mir Daten mit einem Modem senden möchte, ist der Faxstar zwar in der Lage die 1100 Hz von 1300 Hz zu trennen, aber die Software nicht. Ich würde gern z.B. auch diesen Brief per Datensendung an einem bestimmten festgelegtem Tag senden bzw. auch empfangen wollen. Die Software meldet mir nur, daß der Anschluß (durch Fax bzw. DFÜ) belegt ist. Ich würde mich freuen wenn jemand helfen könnte.

Nun zur Frage was ich alles mit meinem Rechner mache. Ich bin Geschichtsfreak. Ich sammle seit meiner Kindheit Geschichtsdaten und baue meine eigene Datenbank auf. Ich habe immerhin zwei Datenbanken mit etwa 80 MB Daten angesammelt. Die Ansammlung bewegt sich mittlerweile bei etwa 100000 Daten. Nachdem ich auf Arbeit einen eigenen Rechner habe, nutze ich meinen PC tatsächlich nur noch für mein Hobby. Ab und zu programmiere ich in Turbo Pascal für Windows 1.5 kleinere Routinen (dll) - mehr Formeln aus der Technik (zur Weiterverwendung in Tabellenkalkulationen).

Ich habe etwa 1988 mit einem ZX81 angefangen. Nun habe ich noch einen defekten ZX Spectrum eigentlich schade. Gibt es jemanden der mir diesen reparieren kann ? (Bildschirm springt nicht auf das Prompt und bleibt schwarz wie beim Reset).

So ich habe erst einmal einen Anfang gemacht. Ich möchte alle Clubmitglieder grüßen - und wünsche noch viele Ausgaben der Clubinfo.

Heiko Werner



PS: Seit heute habe ich eine neue Telefonnummer
0351/8494074

Zu verkaufen:

1.) Tandy Model 4P mit folgender Ausstattung:
512 KRAM, 8 Mhz, 2 Floppy 80/40 Track
Anschluß für externe Laufwerke — Prs. 200,- DM

2.) Tandy Illlodel 4P mit folgender Aussfaftung:
512 KRAM, 6,3 Mhz, 2 Floppy 80/40 Track
IBM - und Tandyzeichensatz
Anschluß für externe Laufwerke — Prs. 250,- DM

3.) Diskstation (mit 2 Netzteilen)
Für 2 Laufwerke - z.Zt. 1 3,5 Laufwerk — Prs. 100,- DM

4.) Diverse Orginal Software z.B.:
LeScript Vers. 2.0
(Spezialvers. für deutsche Tastatur) mit zus. Druckertreiber für HP Deskjet
Allwrite- Textverarbeitung
Multi Basic Vers. 1.00.01
Bascom - Basic Compiler
Pro-Create Editor/Assembler
Assembly Language Development System (ALDS)
T/Maker - Integriertes Paket
Diverse Utilities (Model 4 ToolBelt, Packer, DoubleDuty, Hyperdrive, Superdrive)
Preise Software nach Vereinbarung !

Jörg Brans
Tel.: 040-691 27 16
Tieloh 55, 22307 Hmb.

<u>**Dampf für die serielle Schnittstelle**</u>

(oder: wie komme ich an den RESET-Knopf?)

Die beschriebene Schaltung ist seit etwa einem halben Jahr bei mir in Betrieb und überträgt auf etwa 60 Metern Entfernung Daten mit bis zu 19200 Baud. Sicherer jedoch mit 9600 Baud. Ab 19200 Baud können die Op's nicht mehr und das Signal verschleift sich. Diese Geschwindigkeiten sind aber nur für DOSen und Z280-Rechner erreichbar. Als Kabel habe ich ein stinknormales, vieradriges Telefonkabel verwendet, daß z.T. auch noch direkt neben Netzleitungen verläuft. Nennenswerte Einstreuungen habe ich nicht festgestellt. Die Abschirmung braucht nur bei Übertragungsproblemen und dann nur einseitig an (E) angeschlossen zu werden.

Zur Funktionsweise: Auf der Sendeseite werden die Signale mit zwei Op's (als invertierende Treiber geschaltet) verstärkt. Es wird nicht nur das Signal sondern auch das Bezugspotential dazu übertragen. Dadurch werden auch Potentialunterschiede zwischen verschiedenen Spannungsversorgungen aufgefangen. Auf der Empfängerseite ist ein Op als Differenzverstärker geschaltet. Die jeweiligen Ausgänge sind hochohmig, ein nennenswerter Strom fließt nur bei der Übertragung von Daten. Handshake gibt's hardwaremäßig nicht, also muß mit softwarigem Protokoll gearbeitet werden.

Als einziges Problemchen bleibt die Betätigung des Resetschalters, wenn sich der Remote-Rechner aufgehängt hat. Dem Einsender der ersten brauchbaren Lösung sei ein Schinkenbrot ausgelobt, (Nein Rainer, das Telefonklingeln fällt aus, dazu rufen zu viele Leute an!)

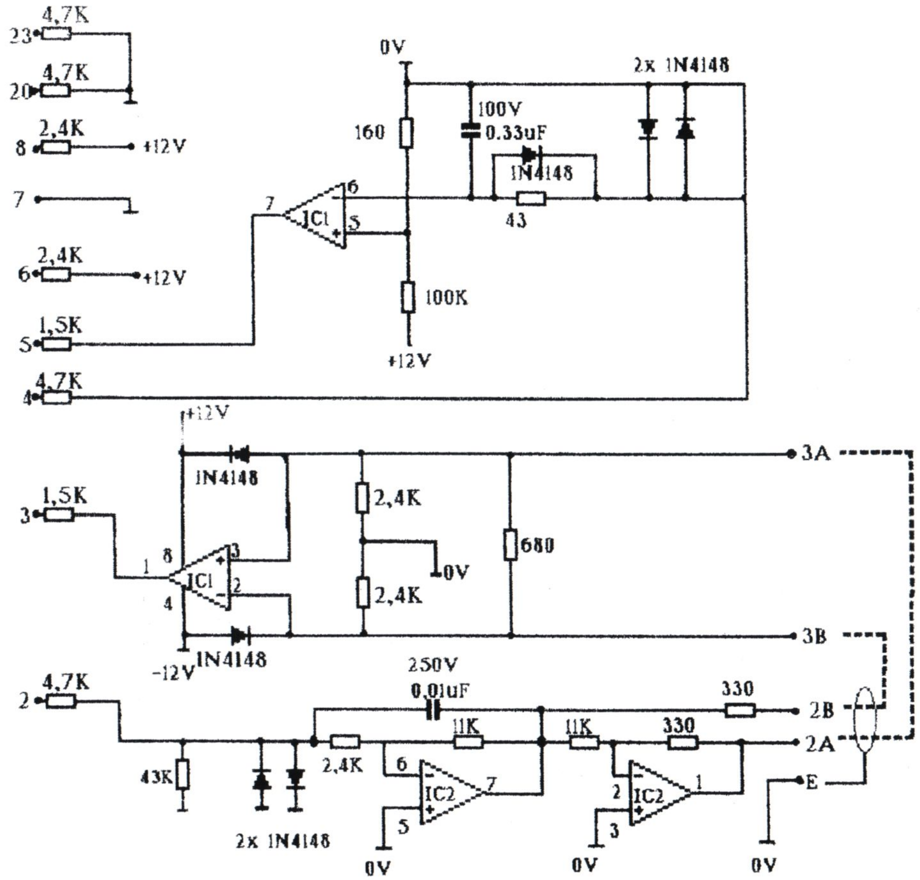
Ein Platinenlayout (einseitig) für ECB-Systeme habe ich schon halbfertig, das gibt's aber nur, wenn sich ein paar Leute melden, sonst lohnt sich der Aufwand nicht.

Ich habe den Schaltplan mehrfach durchgesehen und hoffe er ist jetzt fehlerfrei. Trotzdem ist jeder seines Glückes Schmied.

<u>**Achtung!**</u>
<u>**Dieser "Linecoupler" darf nur an eine gleiche Gegenstation angeschlossen werden, NICHT an den Telefonanschluß der POST!!**</u>

P.S: Logisch, daß zwei davon gebraucht werden!

Inhouse-Modem für RS-232



IC 1.2: MC1458
(2-fach Ops)

IC2: PIN8 +12V
PIN9 -12V

Es werden immer wechselseitig A und A, B und B miteinander von Sender und Emfanger verbunden.

Viel Spaß beim Löten

Andreas

Die aktuellen Baustellen

Hervorragende Eigenschaft eines Hardwarekoordinators scheint es zu sein, daß er derjenige ist, der das Oszi und den halboffenen Rechner auf's Clubtreffen schleppt. Hartmut fragte denn auch an, ob ich mich nicht als Hardwarekoordinator um das Clubleben bemühen wolle. Ich erklärte mich dazu bereit, zusammen mit Andreas Magnus diese Funktion fortzuführen, zum einen, weil es weniger Arbeitsaufwand bedeutet, zum anderen kann man sich des gegenseitigen Einmannfeedbacks sicher sein (besser als nix). Andreas Magnus fand die Idee des geteilten Leids auch nicht schlecht, also "schau'n wir mal", wie Franz sagen würde. Seitdem ist auch schon die eine oder andere Telefoneinheit durchgerasselt. Wir sind auch bereit die Telefondrähte zum G1ühen zu bringen, sollte es dereinst was zum Koordinieren geben.
Also bin ich schon ` mal auf dem Clubtreffen mit dem Hut 'rumgegangen, zu schauen womit sich der "harte Kern" beschäftigt. (so viele Artikel für's Info)
Das Ergebnis will ich hier nicht nur wiedergeben, ich fände es gut wenn IHR diese Seite zur Einrichtung macht. Eine Basis zum Hilfe suchen und bieten, vielleicht sogar zum spontanen Drei-Nasen-Arbeitstreffen. Frau/Mann muß ja nicht alle Schwierigkeiten gehabt haben und fast immer sind schon Erfahrungen da.

Name:	Rechner:	Projekte:
Fritz Chwolka	Genie 1,2,3,3s Kaypro II/4 TRS 80 I,II,IV Kontron PSI 80 Talung TPC 2000 Epson PX-8, Z280 IMS 8000+5000 Alphatronic P4/+P2	HD an Genie 3s mit OMPI.
Michael Kalb	CPC 6128, mit S10 aus PCI, Anpassung an Qterm und LYNC. Schneider PC1512 div. 3,5"+5,25" LWs STAR LC 24 10 Drucker Joyce PCW 9512 CPU 280 von T. Reh Prommer 80	Prommer 80, CPC RS23L, Verbindung der Rechner mit CLubnet von Rainer Schmitz
Jürgen Kemmer	486er, Zyxel-Modem Ultrasound, SB Pro 1.0 Drucker Epson LX800 DCF-Receiver an LPT	Portlerung von Turbo Vicion nach L1NUX Einbindung der Hardware nach LINUX IR-Sender an COM3
Andreas Kleespies	PX-8, MC-CP/M, Gen i e3s Knypro II/4, 286er, Prommer 80, Scanner (Joyce) Inhousemodem	HDD an Genie 3s m, OMTI div, A/D,D/A Wandler Inhouse-Modem
Rainer Kulhanek	3XSpectravideo 32B 2X ZX81, Bondwell 2, Joyce, 2* XT, 286, 386 9-Nadler NL1O, CD-ROM Modem 2400, 14400 TA-Typenraddrucker	CP/M-BIOS vom ROM laden, DFÜ verbessern, Bücherverwaltung unter dBase, ganz kleine Hardwarebasteleien
Jörg Linder	KC 85/4 (m,Floppy) CPU 280 486	HDD an KC Betriebssystem des KC erweitern, Modemanschluß an KC
Hartmut Obermann	Tandy Model 4P, 486er PC und Notebook, 386 PC, ZYXEL 19200er Modem, HP 4c Laserdrucker	Aufbau und Umbau von PCs jede Menge PC-Baugruppen vorhanden
ALexander Schmid	Genie 2s, 2* Genie 3s Triumph Adler P3, REH-CPU 280, 386er, Amstrad Notepad NC-IOO, 14400er Modem, Drucker (DeskJet 500), statische Ramdisk aus c't	Installation von UZI (Unix for Zilog) auf der CPU 280
Rainer Schmitz	PX-B, Joyce, Eurocom 2V7 PC, Bondwell 2, 3* NCR Decision MateV, NCR Pentop Atari Portfolio, Force VME Board, 2* Modem 1200,2400 Scanner (PC), Drucker (Epson LQ100) TNC2DC (Packet Radio) Wavemate Super Bullet	Wavemate ins Gehäuse einbauen (2X HD ,2X DD FDD) PC- Modemkarte an Z80-Bus anpassen, (wie Hercules-Karte aus Clubinfo)

Was mir sonst noch so einfällt:

Die Firma eMedia "verramscht" z.Z. Platinen aus alten Tagen: ECBBoards wie Prommer 80, 1MB Ramdisks (nach Kayser; weiß jemand was zu den Timingproblemen??), Pal Brenner für 30 DM. (GAL wäre interessanter) Auch Einplatinenrechner wie CEPAC 80, (wo gibt's noch NSC800 bzw. NSC810 ??); EPAC 09 (mit 6809) in gleicher Preislage zu haben, (Also falls nochmal jemand anfangen möchte mit Monitoreprom und Assembler: Der Winter ist lang). Näheres dazu steht in der aktuellen c't, ziemlich am Ende

Das war's erstmal von meiner Seite

Have fun
Andreas KLeesples

Ich suche einen TRS-80-Computer und zwar nur die Tastatur mit dem darunter sitzenden Z-80-Gerät. Ein Interface sowie Bildschirm, Netzgeräte und Laufwerke besitze ich noch (Model 1, Baujahr etwa 1979).

Folgende Geräte könnte ich kostenlos abgeben:

⇒ Defekte TRS-80-Tastaturen mit Computer. Zwei davon haben eine zusätzliche separate Zahleneingabe.

⇒ Dann habe ich noch ca. 20 nicht benutzte 5 Zoll Shugart-Laufwerke (Spitzenqualität ich benutzte diese Geräte seit Jahren, Ausfälle unbekannt.) Die Geräte sind für DS,DD 40 Tracks, also die klassisvhen 360-KB-Typen, natürlich auch für PC's geeignet, aber so etwas will ja heute kein Mensch mehr.

⇒ Auch drei funktionstüchtige Karten für hochauflösende Graphik kann ich abgeben. Vielleicht kennt sich jemand damit aus.

Schließlich habe ich noch 2 TRS-80-Computer (also Tastatur mit.... und Interface), aber diese beiden Geräte arbeiten nicht mit den oben genannten Laufwerken zusammen. Ich habe festgestellt, daß sie mit 35 Tracks-Laufwerken SS zusammenarbeiten, und ich weiß nicht so recht, was ich mit den Dingern machen soll. Wenn sie jemand will, bekommt er sie. Die Geräte besitzen zwischen TRS-80 und Interface einen Puffer (Buffer), der auf dem kurzen Zwischenkabel sitzt, außerdem muß man noch so eine Art Diodenstecker (von Tonhandgeräten her bekannt) zusätzlich benutzen. Es funktioniert!
Dazu gehören noch serielle Schnittstellen-Bausteine.

Na wie wär's, beißt jemand an?

Wolfgang Berndt
Friedberger Straße 92c
61169 Friedberg
Tel.: (06031) 2963

Netiquette
- der Knigge für Datenreisende

Auch das noch, Benimmregeln für das Datennetz! Sie helfen jedoch, Geld zu sparen und die Online-Foren interessant zu halten.

⇒ **Fassen Sie sich kurz**
Überlegen Sie bei Nachrichten und Mitteilungen, die Sie in das Usenet, in Compuserve-Foren oder in andere Dienste stellen wollen, vorher:
- ⇒ ob das Thema nicht schon behandelt und diskutiert wurde,
- ⇒ wie Sie Ihre Gedanken möglichst kurz zusammenfassen,
- ⇒ ob Ihre Aussagen wirklich genug Relevanz haben, um sie einmal rund um die Welt zu schicken.

⇒ **Vermeiden Sie Redundanz**
Wenn Sie auf einen Beitrag antworten, schreiben Sie nichts in Ihre Antwort, was bereits ein anderer Teilnehmer gesagt hat. Denn nichts ist so lästig wie in 10 oder 20 Antworten auf einen Beitrag immer wieder die gleichen Argumente zu lesen.

⇒ **Quoten Sie sparsam**
Die meisten Mail-Programme und Newsreader haben eine Quote-Funktion, die automatisch den Originaltext einfügt und diesen mit spitzen Klammern markiert. Machen Sie von dieser Funkrion Gebrauch und zitieren Sie in ihrer Antwort die Passagen, auf die Sie sich beziehen. Das gilt für eine Mail genauso wie für einen Netzbeitrag. Denn nur so weiß der Leser auch noch einige Tage nachdem er die Ursprungsnachricht losgeschickt hat, worauf Sie eigentlich antworten. Allerdings wäre es übertrieben, einen Beitrag komplett zu quoten und darin einige knappe Antworten zu verstecken.

⇒ **Aussagekräftiger Betreff**
Wie bei einer Zeitung die Schlagzeile, so entscheidet der Betreff Ihrer Nachricht darüber, ob sie gelesen wird oder nicht. Eine Nachricht mit dem Titel "Verkaufe Computer" hat sicher nicht den Zuspruch wie „verk. 486 DX2/66 für DM 1.000“. In den meisten deutschen bzw. deutschsprachigen Brettern und Foren hat sich übrigens eingebürgert, für Verkaufe ein „V:“ und für Suche ein „S:“ vor den jeweils gehandelten Gegenstand zu setzen.

⇒ **Keine kommerzielle Werbung**
Gerade im nichtkommerziellen Usenet, aber auch in Compuserve und anderen Diensten, reagieren Sysops wie User allergisch auf kommerzielle Werbung. Verzichten Sie darauf!

⇒ **Antworten per Mail senden**
Eine deutliche Entlastung für das Usenet bringt die Methode, Antworten auf öffentliche Beiträge per E-Mail zu senden. Denn in diesem Fall wird eine Antwort nur einmal über das Internet geschickt und nicht als Anhängsel des Originalbeitrages um die ganze Welt transportiert. Schreiben Sie deshalb an den Schluß Ihres Beitrags, daß Sie um die Zusendung von Antworten per E-Mail bitten, und schicken Sie selbst Forum-Antworten per Mail.

⇒ **Keine doppelten Nachrichten**
Schreiben Sie eine Nachricht nur in ein Brett. Wenn Sie meinen, daß in mehreren Foren ein Interesse an Ihrem Text bestehen könnte, schreiben Sie in diese Foren allenfalls einen kurzen Querverweis auf das Original. Das genügt in der Regel.

⇒ **Menschen lesen Ihre Texte**
Auch wenn der Computer, auf dem Sie eine Nachricht einhacken, kühl und unmenschlich wirkt, denken Sie immer daran, daß am anderen Terminal ein Mensch sitzt, der Ihre Texte liest. Da die elektronische Nachrichtenübertragung nur auf Text basiert, können Sie keine Gesten oder Gesichtsausdrücke mit übermitteln. Dadurch wirkt der Inhalt Ihrer Nachrichten oft anders, als es gemeint war.

Auf den Netzen entwickeln sich viele erbitterte Auseinandersetzungen aus Mißverständnissen. Daher:
- ⇒ verzichten Sie tunlichst auf persönliche Angriffe,
- ⇒ beschränken Sie sich auf Fakten und Argumente in Ihren Messages,
- ⇒ beachten Sie, daß Meinungsäußerungen oder scherzhaft gemeinte Anspielungen leicht mißverstanden werden können,
- ⇒ verwenden Sie Emoticons (siehe Kasten auf Seite 5).

Wenn Sie ein persönliches Problem mit einem anderen Netzteilnehmer haben, tragen Sie diesen Konflikt nicht via E-Mail und schon gar nicht in öffentlichen Foren aus. Ein Anruf: oder ein kurzes Treffen wirkt hier oft Wunder!

⇒ **Watch your language**
Internet und Compuserve sind keine geschlossenen Gesellschaften. Jeder kann Ihre Nachrichten in den öffentlichen Brettern lesen. Achten Sie daher darauf, daß Sie niemanden in diesen Foren bloßstellen und dessen Ruf schädigen.

⇒ **Kurze Signatur verwenden**
Die Signatur ist quasi die Visitenkarte eines Internet-Teilnehmers. Sie sollte unter jeder Nachricht und Mail stehen, die Sie über das Netz schicken. Generell sollte die Signatur nicht länger als zwei bis vier Zeilen sein. Manche User haben ihre digitale Unterschrift zur Kunstform erhoben und glänzen mit 15 Zeilen langen Signaturen, die die komplette Adresse des Netzteilnehmers sowie Unmengen schlauer Zitate berühmter Personen enthalten. CompuserveTeilnehmer können übrigens ganz auf eine Signatur verzichten.

MARTIN GOLDMANN

Die Kleincomputer (KC) aus Mühlhausen

Ein wenig Geschichte...
Ursprünglich war die KC-Serie als Heimcomputer konzipiert. Daher melden sich auch noch einige KC 85/2 nach dem Einschalten mit "HC 900", wobei HC für Homecomputer steht. Doch aufgrund des Bauelemente-Mangels (speziell Speicherbausteine waren in der DDR Goldstaub) wurde von oberster Stelle entschieden, diese Computer nur für gesellschaftliche Bedarfsträger zu produzieren. (Gesellschaftliche Bedarfsträger = volkseigene Betriebe, öffentliche Einrichtungen usw.)

Der erste Computer aus Mühlhausen wurde 1985 auf der Leipziger Messe als KC 85/2 vorgestellt. An die 85 wurde "/2" angehangen, weil zur gleichen Zeit von Robotron aus Sömmerda der KC 85/1 präsentiert wurde. Beide Geräte sind aber untereinander NICHT kompatibel.

Wenig später wurde der recht spärlich ausgestattete KC 85/2 weiterentwickelt. Es entstand der KC 85/3. Auch wenn es inzwischen 1987 geworden war, wurde er unter diesem Namen verkauft. Die Bildungsministerin der DDR hatte inzwischen die Notwendigkeit der Informatikausbildung erkannt. Als Arbeitsmittel wurde dazu der KC 85/3 auserkoren, weshalb dieser in relativ großer Stückzahl produziert wurde.

Krönender Abschluß der Mühlhäuser Produktionsreihe war der KC 85/4. Er ging 1989 in die Serienproduktion und kam auch offiziell in den Handel. Wäre nicht die Wende dazwischengekommen, so gäbe es wesentlich mehr Geräte dieser Art.

Als ein weiteres Produkt sollte unbedingt noch der KC-compact erwähnt werden. Er hat zwar mit der übrigen KC-Serie nichts zu tun, ist aber zum Schneider CPC kompatibel. Der Nachbau ist so perfekt, daß er auch den selben ROM-Bug hat wie der CPC. Von diesen Computern gibt es aber nur wenige, da sie erst ab Mitte 1989 produziert wurden.

Ein wenig Technik...
Alle KCs haben äußerlich den gleichen Aufbau. Das Gehäuse ist ca. 385*270*77 mm groß und beinhaltet neben der Stromversorgung die Hauptplatine und zwei Modulschächte. Standardmäßig wurde der KC mit einer abgesetzten Tastatur ausgeliefert, die aber nur wenig komfortabel ist. Normalerweise schließt man einen Fernseher über den HF-Ausgang an, doch ein besseres Bild liefert der rückseitige RGB-Ausgang.

Im Innern des KC verrichtet ein U 880 mit 1,75 MHz sein Werk. Dieser Prozessor ist zum Z80 voll kompatibel. Aufgrund der Tatsache, daß der Prozessor die farbige Vollgrafik allein bewerkstelligen muß, ist er in mancher Hinsicht etwas zu langsam. Erfreulich ist hingegen der modulare Aufbau des Systems. Durch eine geschickte Modulverwaltung und die Anschlußmöglichkeit von Erweiterungsaufsätzen kann man das System nahezu beliebig erweitern.

Der 64 kByte große Adreßraum wird in 16 kByte-Blöcken verwaltet. Der KC 85/2 ist werkseitig mit 16 kByte RAM Arbeitsspeicher, 16 kByte RAM Bildspeicher und 8 kByte ROM ausgestattet. Den KC 85/3 hat man um einem internen BASIC-Interpreter erweitert, so daß dieser auf 16 kByte ROM kommt. Im KC 85/4 sind 64 kByte RAM Arbeitsspeicher, 64 kByte RAM Bildspeicher und 20 kByte ROM enthalten.

Es gab über 35 der erwähnten Modulen, wovon jedoch nicht alle in den Handel kamen. Diese können grundsätzlich in drei Gruppen eingeteilt werden: ROM-Module (auch mit eingebrannter Software), RAM-Module und Ein-/Ausgabe-Module. Die Module werden mittels einer PIO verwaltet, die wie eine MMU eingesetzt wird. Dadurch ist es z. B. theoretisch möglich, bis zu 16 MB Speicher zu adressieren.

Als Erweiterungsaufsätze waren ein Busdriver und eine Floppy-Basis erhältlich. In den Busdriver können vier weitere Module gesteckt werden. Wesentlich interessanter ist die Floppy-Basis. In dieser ist ein kompletter Z80-Rechner mit 4 MHz enthalten. An die Basis lassen sich bis zu 4 Laufwerke mit je 780 kByte Kapazität anschließen.

Glückliche Besitzer der Floppystation können zwei Betriebssysteme "fahren". Zum einen wird das KC-eigene System um die Möglichkeit des Floppyzugriffs erweitert und zum anderen kann man unter einem System namens "MicroDOS" arbeiten.

Ein wenig Softwaretechnisches...
Das MicroDOS ist zu CP/M kompatibel. Es meldet die Version 2.6 zurück, die recht ungewöhnlich ist. Im Grunde genommen wurde nur ein CP/M 2.2 um einige Funktionen erweitert (z. B. SCB), so daß es irgendwo zwischen 2.2 und 3.0 liegt. Dadurch bleibt auch nur noch ca. 50 kByte TPA. Doch im Zuge der Einbindung des GIDE-Interface von Tilmann Reh werden unsere Softwarespezialisten ein leistungsfähigeres System programmieren (müssen).

Interessant ist auch das Zusammenspiel zwischen den beiden Rechnern. Der eigentliche KC dient unter MicroDOS nur noch als Terminal, während die Floppy-Basis die eigentliche Arbeit verrichtet. Der gesamte Datenaustausch wird über einen Koppel-RAM abgewickelt. Dieser ist ein 1 kByte groß und liegt im Adreßraum der Floppy-Basis sowie im I/O-Adreßbereich des KC-Grundgerätes.

Vorteil dieses Aufbaus ist die Nutzbarkeit aller Möglichkeiten des Grundgerätes unter MicroDOS. So kann man z. B. die farbige Vollgrafik nutzen, was jedoch die Programme inkompatibel zu anderen CP/M-Rechnern macht. Desweiteren werden alle gesteckten RAM-Module erkannt und zu einer RAM-Floppy zusammengefaßt. Diese kann durchaus mehrere MB groß sein.

Ein paar Preise...
Sicherlich sind auch die Preise dieser Technik interessant.

Da es die Typen KC 85/2 und 85/3 nicht für die Bevölkerung gab, sind Preise hier rein spekulativ. Computer wurden in der DDR nämlich an Betriebe zu höheren Preisen verkauft als an die Bevölkerung. (Ein PC 1715 mit Grünmonitor und 9-Nadel-Drucker kostete einen Betrieb ca. 20.000 bis 25.000 Mark. Zur Erinnerung: Das war auch nur ein Z80-Rechner mit CP/M 2.2.)

Als der KC 85/4 erstmals in den Handel kam, kostete er ca. 4.500 Mark. Das entsprach etwas mehr als 4 Monatseinkommen. Für ein 16 kByte RAM-Modul mußte man mehr als 750 Mark berappen. Doch trotz dieser horrenden Preise waren Computer und Zubehörteile nur sehr schwer zu bekommen. Digitale elektronische Bauelemente waren Mangelware, denn sie wurden in das NSW (= nichtsozialistische Wirtschaftsgebiet = Westen) exportiert.

Ein kleiner Blick in die Zukunft...
Nachdem es den Hersteller unseres KCs nicht mehr gibt, müssen wir uns halt gegenseitig helfen. Dazu wurde 1992 der KC-Club gegründet. Der damalige Gründer hat aber nach ca. einem Jahr aufgegeben und es ist ziemlich still um unseren Club geworden. Also habe ich kurzerhand die Initiative ergriffen. Seit Oktober 1994 gibt es wieder einen KC-Club und eine Zeitschrift.

Inzwischen ist unsere Zahl auf knapp 70 Mitglieder gewachsen, wovon ein großer Teil auch recht aktiv ist. Im April 1995 hatten wir unser erstes Clubtreffen. Es war ein voller Erfolg - 30 Mitglieder waren erschienen. Das nächste Treffen ist auch schon in Vorbereitung.

Unsere derzeitigen Projekte sind: Anschluß des GIDE-Interfaces von Tilmann Reh; Anschluß einer komfortableren Tastatur; Entwicklung eines Sound-Modules, Beschleunigung des KCs; Entwicklung eines neuen MicroDOS. Dies sind jedoch nur die größeren Projekte. Viele Mitglieder basteln oder programmieren kleinere Sachen, die ich hier nicht alle aufzählen kann.

Ich hoffe, daß dieser Einblick in die "KC-Welt" einigermaßen interessant war.

Bye, bye
Jörg Linder

Bericht von den Z-Festen

Die Aktivitäten rund um den Z80 und das CP/M-Betriebssystem waren in diesem Frühjahr kaum zu überbieten. Nachdem ich die Clubtreffen des KC-Clubs und des Club-80 gut hinter mich gebracht hatte, konnte ich die beiden Z-Feste natürlich nicht auslassen.

Z-Fest East
Am 15. Juni 1995 gab es sozusagen eine Weltpremiere: das erste Z-Fest in Ostdeutschland, genauer gesagt in Ilmenau. Wir hatten einen Raum der Technischen Universität Ilmenau bekommen. (Wie es dazu kam, möchte ich nicht erläutern, denn das wäre eine Geschichte für sich. Turbulent ist auf jeden Fall das treffende Wort.) Um 16.30 Uhr war der Raum frei und ich wollte gerade meine Anlage aus dem Auto holen, als ich jemanden sah, der ein "Werbeplakat" für das Z-Fest anklebte - Jay Sage.
Obwohl dieser Donnerstag kein Feiertag war, kamen immerhin 8 Leute zu unserem Treffen (einschließlich Jay und meiner Wenigkeit). Erstaunlicherweise waren auch alle pünktlich, was aufgrund der Verkehrssituation in Ilmenau und Umgebung nicht selbstverständlich ist.
Wir kamen dann auch gleich ins Gespräch. Jay interessierte sich sehr für den KC und die Rechentechnik aus der DDR. Als er aber die Preise hörte, verschlug es ihm fast die Sprache. Auch die Alltagsgeschichten, die man damals erleben durfte, waren für ihn teilweise unfaßbar. Umso erfreulicher fand er es, daß unser Club noch so aktiv ist.
Gegen 19.00 Uhr verließ Jay unsere kleine Runde, weil er zum Abendessen verabredet war. Wir machten derweil weiter. Um 21.00 Uhr war aber auch bei uns Schluß - schließlich mußten einige am nächsten Tag wieder zur Arbeit. Zusammenfassend kann man sagen, daß es ein interessantes Treffen war, das man wiederholen sollte. Dann allerdings zu einem günstigeren Zeitpunkt, so daß mehr Leute kommen (können).
An dieser Stelle möchte ich Jay Sage, Prof. Uhlmann und Helmut Jungkunz danken, die uns dieses Treffen durch ihren persönlichen Einsatz und mit ihrer modernen Kommunikationstechnik ermöglicht haben.

Z-Fest West
Dieses Treffen ist schon als traditionell zu bezeichnen, denn es war immerhin das fünfte Mal und davon das dritte Mal in Güglingen. Aufgrund meiner langen Anfahrt (zur Erinnerung: Güglingen liegt in der Nähe von Heilbronn) bin ich schon am Freitag, den 16. Juni gefahren. Die Familie Vogelmann, in deren Haus das Ganze stattfand, war sichtlich überrascht, als sie mich sahen. Trotzdem wurde ich gleich sehr herzlich begrüßt.
Am Samstag ging es dann aber richtig los. Nach und nach trafen jede Menge Leute ein. Insgesamt kamen wir auf die beachtliche Zahl von 15 Leuten aus 3 Ländern (Bayern bleibt mal unberücksichtigt). Ich glaube, daß man uns als den "harten Kern" der CP/M-Gemeinde in Deutschland bezeichnen kann.

Es wurde ausgiebigst gefachsimpelt, gezeigt und gestaunt. Unter anderem waren zu sehen:
- ein C 128 mit zwei Festplatten und einem CD-ROM-Laufwerk
- drei CPU280 in den unterschiedlichsten Ausbauvarianten (zwei davon wurden via Kabel und Programm gekoppelt)
- ein KAYPRO IV
- der lauffähige (!) Prototyp des G-IDE-Interfaces von Tilmann Reh
- natürlich meine KC-Anlage

Über letztere hat Helmut Jungkunz nicht schlecht gestaunt. Er war eigentlich ein wenig enttäuscht, daß sie ein so kompaktes und elegantes Design hat. Insgeheim hatte er einen großen unförmigen Blechkasten erwartet. ("Nun hat Jay ein ganz falsches Bild über Eure Computertechnik von Robotron.")

Als absoluten Höhepunkt des Abends präsentierte uns Helmut das Video von seiner Trenton-Reise. Es war wirklich faszinierend, was er auf zweieinhalb Stunden Kassette gebannt hat. Der Besuch im Bostoner Computermuseum ließ die Augen ebenso leuchten wie der Flohmarkt beim Treffen in Trenton. Auch der aufgezeichnete Vortrag von Hal Bower über sein Netzwerksystem unter CP/M (!!!) war absolute Spitze.

Dieses Video kann man übrigens von Helmut für nur 20,00 DM zuzügl. Versandkosten bekommen. Nähere Informationen bei mir oder Helmut Jungkunz, Zacherlstraße 14, 85737 Ismaning, Tel. (0 89) 9 61 46 33.

In den ersten Stunden des Sonntags suchten wir unser Hotel auf, um ein wenig zu schlafen (um 8.30 Uhr gab's schließlich Frühstück). Einigermaßen fit machten wir dann dort weiter, wo wir am Vortag aufgehört hatten.

Helmut nutzte die Gelegenheit, mich und meine KC-Anlage auf Video zu verewigen. Einen kleinen Beitrag möchte er zusammenstellen und der SCUG vorführen. Anschließend packte ich alles zusammen, denn am Vormittag wollte ich heimwärts fahren.

Natürlich kam es anders als geplant. Die neueste Software aus den USA mußte noch schnell kopiert werden, noch schnell Mittag essen, noch schnell ein paar Adressen ausgetauscht, noch schnell... Obwohl alles schnell ging, war es dann schon fast 16.00 Uhr, als ich den Heimweg antrat.

Alles in allem war es ein abwechslungsreiches, interessantes, informatives, nutzbringendes, ... Treffen, an dem ich auf alle Fälle wieder teilnehmen werde - vorausgesetzt, es findet wieder statt.

Zu guter letzt noch ein riesengroßes Dankeschön an Familie Vogelmann für die Gastfreundschaft und an Uwe Herzog und Roderich Vogelmann für die Organisation und an alle Anwesenden für ihre Anwesenheit. :-)

Star Trek - Die nächste Version

(Autor unbekannt, Übernahme aus SCUG-Info 1/95)

Picard: Mr. LaForge, war der Versuch, einen Schwachpunkt bei den Borg zu finden, erfolgreich? Und Mr. Data, konnten Sie eine Verbindung zu deren Kommandoebenen aufbauen?

Geordi: Ja, Captain. Wir fanden die Lösung in den Computerarchiven des späten zwanzigsten Jahrhunderts.

Geordi drückt eine Taste, und auf dem Monitor erscheint ein Logo.

Riker (etwas verwirrt): Was in aller Welt ist ein "Microsoft"?

Data: Lassen Sie mich Ihnen das erklären. Wir schicken dieses Programm, das - aus welchen Gründen auch immer - "Windows" heißt, durch die Kommandoebenen der Borg. Einmal in ihrer Kommandoeinheit installiert, fängt es an, unaufhaltsam Systemressourcen zu belegen.

Picard: Aber die Borg können sich anpassen. Werden sie nicht einfach ihr Verarbeitungssystem ändern, um den erhöhten Speicherbedarf auszugleichen?

Data: Das ist korrekt, Captain. Doch wenn Windows das merkt, erstellt es eine neue Version von sich selbst - ein sogenanntes "Upgrade". Und die Belegungsrate wächst bei jeder Version exponentiell. Die Borg werden nicht in der Lage sein, sich schnell genug anzupassen. Wahrscheinlich wird ihre ganze Rechenkapazität zum Erliegen kommen, so daß sie ihre normalen Aktivitäten nicht mehr fortsetzen können.

Picard: Ausgezeichnete Arbeit! Das ist sogar noch besser als die Idee mit der unlösbaren Rechenaufgabe.

... Fünf Stunden später...

Data: Captain, wir haben Windows erfolgreich in der Haupt-Kommandoebene installiert, und wie erwartet hat es sofort 85 % der Ressourcen belegt. Bisher haben wir noch keinerlei Informationen über das Upgrade.

Geordi: Unsere Scanner registrieren eine enorme Erhöhung der Speicher- und CPU-Kapazitäten, aber keinerlei Anzeichen vom Upgrade, um diese auszugleichen.

Picard: Mr. Data, durchsuchen Sie noch einmal die Geschichtsdatenbanken! Vielleicht haben wir etwas übersehen.

Data: Sir, ich glaube, ich habe den Grund für das Versagen des Upgrades. Anscheinend haben die Borg diesen Teil unseres Planes wirkungslos gemacht. Sie haben die "Registrierung" von Windows nicht vorgenommen.

Riker: Captain, wir haben keine Wahl. Bitte um Erlaubnis, die Notfallprozedur 3F.."

Geordi (aufgeregt): Warten Sie, Captain! Gerade habe ich entdeckt, daß ihre CPU-Kapazitäten auf 0 % abgesackt sind.

Picard: Mr. Data, was zeigen die Scanner jetzt an?

Data: Offensichtlich haben die Borg eine internes Windows-Modul namens "Solitär" gefunden. Das hat ihre gesamte Rechenzeit auf einen Schlag verbraucht.

Picard: Lassen Sie und warten und beobachten Sie, wie lange Solitär ihre Funktionen beeinflußt.

... zwei Stunden vergehen...

Riker: Geordi, wie ist der Status bei den Borg?

Geordi: Wie wir erwartet haben, versuchen die Borg ihre Speicher- und Rechenkapazität durch Modifikationen am System zu erhöhen. Aber jedesmal, wenn sie es geschafft haben, Ressourcen frei zu machen, übertrage ich neue Windows-Module aus dem sogenannten "Microsoft Fun-Pack".
Picard: Wieviel Zeit bringt uns das?
Data: Ausgehend von der derzeitigen Effizienz der Borg bleiben uns etwa noch sechs Stunden.
Geordi: Captain, ein drittes Schiff nähert sich unserem Sektor.
Picard: Identifizierung?
Geordi: Es hat Kennzeichen, die dem "Microsoft"-Logo sehr ähnlich sind.
... über die Lautsprecher der Brücke...
Gates: Hier spricht Admiral Bill Gates vom Microsoft-Flaggschiff "Monopoly". Wir haben eine positive Anzeige dafür, daß in diesem Sektor nichtregistrierte Software verwendet wird. Überschreiben Sie daher alle Ihre Vermögenswerte, und wir regeln diese Angelegenheit friedlich. Sie haben zehn Sekunden, unserem Vorschlag zuzustimmen.
Data: Das fremde Schiff hat seine vorderen Torpedorohre geöffnet und Tausende von humanoid geformten Objekten ausgestoßen.
Picard: Auf den Schirm, vergrößern!
Riker: Mein Gott, Captain! Diese Menschen fliegen genau auf das Borg-Schiff zu - ohne Raumanzüge. Wie können sie nur im Weltraum überleben?
Data: Ich glaube nicht, daß es Menschen sind, Sir. Wenn Sie genauer hinsehen, werden Sie bemerken, daß sie etwas bei sich haben, was man im 20. Jahrhundert als Aktenkoffer bezeichnet hat. Außerdem tragen sie "Armani"-Anzüge.
Riker und Picard (entsetzt): RECHTSANWÄLTE!!!
Geordi: Das kann nicht sein. Alle Rechtsanwälte wurden 2017 während des Großen Erwachens in Castor-Behälter gepackt und direkt in die Sonne geschossen.
Data: Das ist korrekt, aber anscheinend haben einige überlebt.
Riker: Sie haben das Borg-Schiff umzingelt und bedecken es mit weißen Papieren.
Data: Ich glaube, diese Dokumente nannte man damals "Einstweilige Verfügung" - eine sehr effiziente Arbeitsmethode.
Riker: Sie reißen die Borg in Stücke!
Picard: Schalten Sie den Bildschirm ab. Ich kann das nicht mitansehen. So ein Ende haben nicht einmal die Borg verdient.
ENDE

Impressum

1. Vorsitzender	Hartmut Obermann	Tel.: 0721/85 40 68
	Wilhelm-Baur-Straße 8	Fax: 0721/85 40 68
76135	Karlsruhe	
2. Vorsitzender	Gerald Schröder	Tel.: 04105 /26 02
	Am Schützenplatz 14	
21218	Seevetal	
Hardewarekoordinator	Andreas Magnus	Tel.: 0209 /87 02 30
	Bismarckstraße 29	
45879	Gelsenkirchen	
NewDOS-Diskothekar	Oliver Volz	Tel.: 0711/74 40 51
	Am Ochsenwald 37A	
70565	Stuttgart (Rohrerhöhe)	
CP/M-Diskothekar	Fritz Chwolka	Tel.: 02464 /89 20
	Saarstraße 34	
52457	Aldenhoven	
C-128-Diskothekar	Günther W. Braun	
	Postfach 80 02 26	
81602	München	
Clubbücherei	Rainer Schmitz	Tel.: 07166 /13 97
	Küferweg 12/1	
73099	Adelberg	
Redaktion	Jens Neueder	Tel.: 0791 /4 28 77
	Rudolf-Then Straße 32	FAX:07971 /2 50 55
74544	Michelbach /Bilz	

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H., 76135 Karlsruhe
Konto Nr. 496 071-605
Postgiroamt Frankfurt, BLZ 500 100 60

Autoren Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.
Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.

Schluß

Hallo Club 80'er,

nun endlich ist wieder einmal ein INFO fertig -sogar ein **Tripelheft**.

Dies liegt aber nicht an der Menge der Artikel, sondern am Zeitlimit. Zum ersten Termin (INFO 48) währe nur ein 6-blättriges INFO möglich gewesen. Die obligatorischen Zugaben der Redaktion abgezogen blieb nicht mehr viel übrig. Beim letzten Clubtreffen, welches auch schon etwas länger her ist, waren alle noch guten Mutes, daß es besser klappen würde.
Für Ende Oktober hatte ich dann ein **Doppelheft** fertig, welches Euch dann noch vor der Hobbytronik erreichen sollte. Dies fiel aber leider auch aus, weil dann der Kopierer streikte.
Zwischenzeitlich habe ich, nachdem ich meine Hardwareausstattung wieder etwas erweitert habe, das komplette INFO in digitalisierter Form unter Word vorliegen (und diesen Schluß neu geschrieben). Ab jetzt wird nur noch ausgedruckt. Das entsprechende INFO-File lege ich auch in Kürze in die DISCOVERY.

Hierzu ein Hinweis für zukünftige Artikel, die nicht auf Datenträger sind:
Bitte verwendet **weißes** Papier und eine **große Schrift** (12'er). Dann habe ich am wenigsten Probleme mit der Texterkennung. Bilder und Grafiken werden in digitalisierter Form eingefügt.

So, das war's auch schon wieder von meiner Seite aus. Der nächste Redaktionstermin ist der **01.03.1996**. Ich hoffe, daß Ihr mir bis dahin doch ein paar Artikel bereitstellen könnt.

Habt viel Spaß mit dem Info.
Ich verbleibe mit den besten Wünschen Euer

Mitgliederliste CLUB 80

Vorname	Nachname	Straße	Land	PLZ	Ort	Telefon privat	Telefax privat	Telefon geschäftlich	Telefax geschäftlich
Harry	Barendt	Hermann-Löns-Straße 7	GER	50181	Bedburg (Erft)	02272/ 7168	02272/ 7168	-	-
Wolfgang	Berndt	Friedberger Straße 92c	GER	61169	Friedberg	06031/ 2963	-	-	-
Ilse	Berndt-Jochum	Stachelsgut 24	GER	51427	Bergisch Gladbach	02204/ 65254	-	02204/ 65254	-
Helmut	Bernhardt	Preetzer Straße 75	GER	24143	Kiel	-	-	0431/ 77578-20	0431/ 77578-99
Georg	Bielenberg	Erikaweg 1	GER	24568	Kaltenkirchen	04191/ 3751	-	04193/ 90430	-
Ulrich	Böckling	Juchaczstraße 61	GER	56203	Höhr-Grenzhausen	02624/ 4861	-	02631/ 895168	-
Günter	Bornmann	Pfarrer Maybaum Weg 81	GER	51061	Köln	0221 / 638319	-	0221 / 9037567	-
Jörg	Brans	Tieloh 55	GER	22307	Hamburg	040/ 6906531	-	-	-
Günther W.	Braun	Postfach 80 02 26	GER	81602	München	-	-	-	-
Fritz	Chwolka	Saarstraße 34	GER	52457	Aldenhoven	02464/ 8920	-	-	-
Volker	Dose	Dorfstraße 10	GER	24235	Brodersdorf	04343/ 1357	-	-	-
Gert	Halgasch	Großschönauer Straße 26	GER	02796	Jonsdorf	035844/ 636	-	-	-
Hans-Günther	Hartmann	Möwenstraße 9	GER	27804	Berne	04406/ 6911	04406/ 1071	0421/ 248-2232	
Ulrich	Hebecker	Büsnauer Straße 15	GER	70563	Stuttgart	0711/ 734800	-	-	-
Manfred	Held	Stirner Straße 22	GER	91785	Pleinfeld	09144/ 92340	09144/ 92343	0911/ 219-5914	-
Klaus	Hermann	Forchenstraße 8	GER	72124	Pliezhausen	07127/ 71945	-	-	-
Manfred	Hürdler	Niederhoferstraße 29	GER	97222	Rimpar	09365/ 4235	-	-	-
Willi	Johnen	Hansemannstraße 1	GER	52351	Düren	02421/ 501305	-	02421/ 33064	-
Michael	Kalb	Königstraße 10	GER	96242	Sonnefeld	09562/ 6387	09562/ 6570	-	-
Dietmar	Kauka	Straße des Friedens 37	GER	04552	Neukirchen (Borna/Leipzig)	03433/851019	-	-	-
Jürgen	Kemmer	Dorfberg 7	GER	97232	Sulzdorf	09334/ 1050	-	-	-
Andreas	Kleespies	Mainzer-Land-Straße 765	GER	65934	Frankfurt /Main	069/ 387432	-	-	-
Ulrich	Kracker	Schwenden 3	GER	87452	Altusried	08373 / 8644	-	0831 / 5611132	0831 / 5611144
Eckehard	Kuhn	Im Dorf 14	GER	72636	Frickenhausen	07022/ 45417	-	-	-
Rainer	Kulhanek	Südendstraße 6	GER	82362	Weilheim	0881 / 5132	-	089 / 17908214	-
Jörg	Linder	Küstriner Str. 68	GER	15306	Seelow	03346/ 520	-	-	-
Walter	Lorenz	Mahräckerstraße 9	GER	60431	Frankfurt /Main	069/ 531656	-	-	-
Andreas	Magnus	Bismarckstraße 29	GER	45879	Gelsenkirchen	0209/ 144029	-	-	-
Herbert	Mahlert	Hohenbudbergerstraße 112 A	GER	47229	Duisburg	02065/ 47217	-	02065/ 902592	-
Paul-Josef	Messerig	Heyratherstraße 9	GER	54552	Dreis-Brück	06595 / 768	-	-	-
Kurt	Messerschmidt	Hanns-Eisler-Straße 54	GER	10409	Berlin	-	-	-	-
Franz	Mössel	Schafferstraße 12	I	39012	Meran	0039-473/34178	-	0039-471/980496	-
Kurt	Müller	Sophie-Scholl-Ring 3B	GER	21502	Geesthacht	04152/ 70643	-	040/ 89983403	-
Jens	Neueder	Rudolf-Then-Straße 32	GER	74544	Michelbach / Bilz	0791/ 42877	-	07971/ 250-50	07971/ 250-55
Christof	Neumann	Zeitblomstraße 22/2	GER	89077	Ulm /Donau	0731/ 6022568	-	0731/ 9749720	-
Stefan	Nitschke	Germenenstraße 5	GER	75045	Walzbachtal 1	07203 /452	-	-	-
Hartmut	Obermann	Wilhelm-Baur-Straße 8	GER	76135	Karlsruhe	0721/ 854068	-	-	-
Jürgen	Peters	Heukoppel 14	GER	22179	Hamburg	040/ 6412371	-	-	-
Bernd	Retzlaff	Kleiner Sand 98	GER	25436	Uetersen	04122/ 43551	-	04103/ 605310	-
Gerd	Rinio	Rennbahnstraße 9	GER	22111	Hamburg	040/6552630	-	-	-
Claus	Ruschinski	Pommernstraße 21	GER	45770	Marl	02365/ 34646	-	-	-
Jörg	Schimmer	Stettinerstraße 28	GER	60388	Frankfurt	06109/ 35336	06109/ 35336	069/ 3800-2385	-
C.	Schinzel	Berliner-Platz 2	GER	70734	Fellbach	0711/ 5782655	-	-	-
Alexander	Schmid	Barbarastraße 64	GER	97074	Würzburg	0931/ 783371	-	-	-

Mitgliederliste CLUB 80

Vorname	Nachname	Straße	Land	PLZ	Ort	Telefon privat	Telefax privat	Telefon geschäftlich	Telefax geschäftlich
Rainer	Schmitz	Küferweg 12/1	GER	73099	Adelberg	07166/ 1397	-	07161/ 608-475	-
Uwe	Schoberth	Petrus-Waldus-Straße 14	GER	75443	Oetisheim	07041/ 7254	-	0711/ 89394500	0711/ 89394513
Hans-Werner	Scholz	Spitalstraße 54	GER	41334	Nettetal	02157/ 3613	-	-	-
Gerald	Schröder	Arminiusstraße 2	GER	22525	Hamburg	040/ 8507131	-	040/ 54715334	-
Egbert	Schröer	Joachimstraße 18	GER	46284	Dorsten	02362/ 75311	-	02362/ 49-9649	-
Horst-Dieter	Schroers	Breslauer Straße 9	GER	85622	Feldkirchen	089/ 9032615	089/ 9043413	-	-
Hartmut	Schulte	Entenschnabel 8	GER	31311	Uetze	05173/ 1248	05173/ 24631	-	-
Harald	Sonnemann	In den Eckwiesen 9	GER	64405	Fischbachtal	06166/ 8512	-	06151/ 92-1265	-
Rüdiger	Sörensen	Wiesbadener Str. 28B	GER	55252	Mainz-Kastel	06134/65342	-	-	-
Ulrich	Staimer	Dekan-Mayer-Straße 15	GER	86199	Augsburg	0821/ 993104	-	0821/ 2574-107	-
Stefan	Stumpferl	Hasenberglstraße 57	GER	80933	München	089/ 3138193	//0893144001//	-	-
Wilhelm	Tornow	Elbblick 46	GER	21629	Neu Wulmstorf	040/ 7007280	040/ 7003854	-	-
Michael	Vogl	Postfach 10 06 10	GER	41490	Grevenbroich /Laach	02181/45112	-	-	-
Richart	Volkmer	Am Spörkel 69	GER	44227	Dortmund	0231/ 752574	-	-	-
Oliver	Volz	Am Ochsenwald 37A	GER	70565	Stuttgart (Rohrerhöhe)	0711/ 744051	-	0711/ 685-3013	-
Hans-Jürgen	Walter	Elssholzstraße 9	GER	10781	Berlin	-	-	-	-
Heiko	Werner	Reichenberger Straße 5	GER	01129	Dresden	0351/ 8494074	-	-	-

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ GENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

das ist das letzte, das ist das allerletzte CLUB 80 INFO

CLUBINFO

52. Ausgabe

Kontaktadresse:
Club 80
Hartmut Obermann
Wilhelm-Baur-Straße 8
76135 Karlsruhe
Tel.: 0721/ 85 40 68

Inhaltsverzeichnis

	Autor & Seite
Clubinternes	
....zum letzten Club 80 - Info	1
Abwicklung des Club 80	1
Dank und Verabschiedung	2
	Hartmut Obermann
Protokoll der Hauptversammlung	2 - 3
	Alexander Schmid
Hardstiftung	3
Mittelmeer-Anämie - Thalassämie	4 - 5
Förderverein für krebskranke Kinder e. V. Karlsruhe	5
	Hartmut Obermann
„Alte“ Computer in DFÜ-Netzen	6 - 8
	Rainer Kulhanek, Stefan Sommer
Die letzten Seiten	
Impressum	9
Schluß	9
	Redaktion
Bücherliste CLUB 80	am INFO-Ende

Vorwort zum letzten CLUB 80 - Info

Wie schon im letzten Info angedeutet, hat sich auf dem vom 28.02. bis 02.03. abgehaltenen Treffen des CLUB 80 dessen Auflösung ergeben.

Nicht etwa, daß die Anwesenden sich diese Entscheidung leicht gemacht hätten, ganz im Gegenteil. Zunächst wurde nochmals über eine Weiterführung und die Gründe für den Niedergang des CLUB 80 diskutiert. Dann entspann ein konstruktiver Disput um die Auflösungsmodalitäten für den CLUB 80. Schließlich soll auch dabei, wie schon zu „Lebzeiten" des Clubs, alles mit rechten Dingen zugehen.

Abwicklung des CLUB 80

Wie man dem Protokoll entnehmen kann, werden die Sachwerte des CLUB 80 unterschiedlich behandelt.

Der Kopierer, der vor Jahren angeschafft wurde, um eine kostengünstige Möglichkeit für die Vervielfältigung des Clubinfos zu haben, ist abgewirtschaftet. Er war schon bei der Anschaffung ein Gebrauchtgerät und wurde im Laufe der Jahre nicht besser, weswegen er dem Clubredakteur Jens Neueder zur Entsorgung überlassen wird.

Das 14400´er Modem, welches der CLUB 80 angeschafft hat, um seinen Mitglieder eine kostengünstige Möglichkeit für den Einstieg in die Datenfernübertragung zu ermöglichen, wird versteigert. Das Mindestgebot ist 30,– DM (+ Porto- und Verpackungskosten). Die Angebote sind bis 8 Wochen nach dem Erscheinungstermin dieses Infos schriftlich an mich zu richten (Hartmut Obermann, Wilhelm-Baur-Str. 8, 76135 Karlsruhe). Das höchste Gebot erhält den Zuschlag. Der erzielte Betrag wird dem Barvermögen zugeführt.

Der Telefonnetzsimulator (Anschlußmöglichkeit für zwei Modems zu Testzwecken) wird zu gleichen Bedingungen versteigert wie das Modem. Ein Mindestgebot gibt es hier allerdings nicht. Angebote bitte an mich.

Auch die Bücher aus der CLUB 80-Bücherei, die sich unter der Obhut von Rainer Schmitz befinden (Adresse im Impressum), werden zu den oben genannten Bedingungen versteigert. Ein Mindestgebot gibt es für die Bücher nicht. Die Angebote sind an Rainer Schmitz zu schicken. Bücher, die nicht ersteigert werden, gehen nach Ablauf der 8-Wochenfrist in den Besitz von Rainer Schmitz über.

Die CLUB 80-Programmbibliothek für NewDOS befindet sich bei Oliver Volz (Adresse im Impressum). Ob die Disketten überhaupt noch lesbar sind, läßt sich nicht sagen. Wer sich für einzelne Programme interessiert, kann diese bis 8 Wochen nach dem Erscheinungstermin dieses Infos bei Oliver anfordern. Kopien von den Programmen werden nicht mehr erstellt, sondern es werden die Originaldisketten verschickt. Wer zuerst bestellt, bekommt die entsprechende(n) Diskette(n). Die Unkosten (Porto und Verpackung) müssen vom Besteller im voraus entrichtet werden. Disketten, die nicht angefordert werden, gehen nach Ablauf der 8-Wochenfrist in den Besitz von Oliver Volz über.

Die CLUB 80-Programmbibliothek für CP/M befindet sich bei Fritz Chwolka. Da die Programme vorwiegend auf seinen privaten Disketten gespeichert sind, verbleiben diese natürlich bei ihm.
Das Barvermögen, inklusive der durch die Versteigerung des Modems, des Telefonnetzsimulators und der Bücher eingehenden Beträge, wird satzungsgemäß zu gleichen Teilen an drei gemeinnützige Institutionen verteilt. Die Vorstellung der Thallassämie-Hilfe, der Hardtstiftung und des Fördervereins für krebskranke Kinder e.V. findet ihr hier im Info.

Nachdem das Geld an die drei Institutionen überwiesen und das Konto aufgelöst wurde, wird die ordnungsgemäße Abwicklung durch Klaus Hermann und Alexander Schmid überprüft. Zu diesem Zweck werde ich beiden die Überweisungen und die letzten Kontoauszüge zusenden. Die Unterlagen werden hier bei mir für fünf Jahre gelagert und sind auch für andere CLUB 80-Mitglieder nach Absprache einsehbar. Nach Ablauf dieser Zeit werden die Unterlagen vernichtet.

Dank und Verabschiedung

Ich möchte es nicht versäumen, mich bei den Teilnehmern des Treffens recht herzlich für die „aktive Sterbehilfe" für den CLUB 80 zu bedanken. Nach dem Motto „besser ein Ende mit Schrecken als ein Schrecken ohne Ende" haben wir meiner Meinung nach das Beste für den CLUB 80 getan, was wir tun konnten.

Natürlich sei auch all jenen Dank, die sich in den letzten dreizehn Jahren für den CLUB 80 und seine Mitglieder aktiv eingesetzt haben. Bibliothekaren, Diskothekaren, Hardwarekoordinatoren und Vorsitzenden, allen sei für ihr Engagement und ihre Arbeit gedankt!

Nicht vergessen möchte ich an dieser Stelle eine Spezies, die in letzter Zeit leider immer seltener geworden ist, nämlich denjenigen, die Beiträge für unser Clubinfo geschrieben haben. Ohne sie hätte es den CLUB 80 nie gegeben, und weil sie praktisch ausgestorben sind, hat auch er das Zeitliche gesegnet.

Ganz besonders gilt mein Dank Jens Neueder, der seit dem zweiten Jahr des Bestehens des CLUB 80 die Clubredaktion übernahm und damit für 46 Infos und eine ganze Reihe von Sonderheften verantwortlich zeichnet. Aber wo keine Beiträge mehr sind, da ist auch nichts zu veröffentlichen und so ist dieses 52. Info leider das letzte, welches er für uns herausgeben wird.

Damit möchte ich mich von allen Mitgliedern verabschieden. Vielleicht „trifft" man sich einmal irgendwo im Internet, oder wo auch immer man sich heute und in Zukunft Informationen über „alte" Computer beschafft. Und natürlich freue ich mich immer noch über jeden, der mich mal anruft oder besucht!

In diesem Sinne always good hacking,

Protokoll der Hauptversammlung des Club 80 am 1.März 1997

Protokollführer: Alexander Schmid

Vom 28.2. bis 2.3.97 traf sich der letzte, klägliche Rest des Club 80 in Gaggenau. Der Ansturm hielt sich diesmal mit 7 Mitgliedern und einem Gast in sehr bescheidenen Grenzen, was der Stimmung aber keinen Abbruch tat. Es wurde jedoch allgemein ein immer geringer werdendes Interesse am Club beklagt und nachdem fast alle zu Fremdrechnern abgewandert sind und niemand mehr etwas zum Info und zum sonstigen Clubleben beiträgt, wurde schließlich und endlich beschlossen, den Club aufzulösen.

I. Feststellung der Beschlußfähigkeit

Anwesend waren:
Jens Neueder
Alexander Schmid
Andreas Kleespies
Hartmut Obermann
Rainer Kulhanek
Klaus Hermann
Kurt Müller
und als Gast Stefan Sommer

I. Kassenprüfung

Die Kassenprüfung durch Klaus Hermann und Alexander Schmid ergab keine Beanstandungen.

II. Entlastung des Vorstandes

Der Vorstand wurde mit 5 Ja-Stimmen bei 2 Enthaltungen entlastet.

III. Auflösung

Aufgrund der Vorgespräche wurde unter den anwesenden Mitgliedern die Frage nach der Auflösung des Club 80 gestellt.

Die Abstimmung ergab 7 Ja-Stimmen.

<u>Der Club 80 ist hiermit aufgelöst</u>.

IV. Feststellung der vorhandenen Sachwerte

- Kopierer (zur Zeit defekt): wird Jens Neueder zur Entsorgung überlassen,
  6 Ja-Stimmen, 1 Enthaltung.
- Modem 14400: wird zur Versteigerung im Clubinfo angeboten, Mindestgebot DM 30.- + Porto- und Verpackungskosten.
- Telefonnetzsimulator, wird zur Versteigerung im Clubinfo angeboten, kein Mindestgebot.
- Club-80-Programmbibliothek: Ankündigung im Clubinfo, wer Interesse hat kann Disketten/Programme anfordern, 8 Wochen nach Erscheinen des Infos Verbleib beim Diskothekar bzw. Entsorgung.
- CP/M-Programmbibliothek, Disketten sind Eigentum von Fritz Chwolka.
- Bücher, Liste wird ins Info gesetzt, Versteigerung gegen Gebot + Porto- und Verpackungskosten, 8 Wochen nach Erscheinen des Infos Verbleib bei Rainer Schmitz.

V. Vorhandenes Kapital

Es wurde beschlossen, das, nach Abschluß aller Auslagen, vorhandene Barvermögen entsprechend der Clubsatzung an folgende gemeinnützige Institutionen als Spende zu überweisen:

- Thalassämie-Hilfe
- Hardtstiftung
- Förderverein für Krebskranke Kinder e.V.

Macht's gut und danke für den Fisch !

HARDTSTIF*tunG*

Die HARDTSTIF*tunG* in Karlsruhe-Neureut ist eine sozialpädagogische Einrichtung für Mädchen, Schwangere, Mutter und Kind in Not- und Konfliktsituationen.

Ziel der sozialpädagogischen und psychologischen Arbeit ist es, die jungen Mädchen und Frauen zu einer selbständigen und eigenverantwortlichen Lebensführung zu befähigen, wozu auch eine Schul- und Berufsausbildung gehören, welche ebenfalls in der Einrichtung, aber auch außerhalb, absolviert werden können.

Zum gesamten Hilfeangebot gehören neben der Stammeinrichtung in Karlsruhe-Neureut noch Außenwohngruppen in Karlsruhe, sowie für besondere intensivpädagogische Maßnahmen ein erlebnispädagogisches Pilotprojekt im Südtiroler-Alpenbereich dazu.

Info für Betroffene und Interessierte:

HARDTSTIF*tunG*
Verein mit Körperschaftsrechten
Neureuter Hauptstraße 2
76149 Karlsruhe
Tel.: 0721/7082-0
FAX: 0721/7082-24
Spendenkonten:
Postscheck Karlsruhe 148 71-757 (BLZ 660 100 75)
Sparkasse Karlsruhe 9 018 003 (BLZ 660 501 01)
Evang. Kreditgenossenschaft Karlsruhe 0 506 001 (BLZ 660 608 00)

MITTELMEER - ANÄMIE
eine Erbkrankheit

1. Was ist THALASSÄMIE?

Die Thalassämie (=Mittelmeer-Anämie) ist eine Erkrankung des blutbildenden Systems. Sie gehört zu einer genetisch seltenen Störung der Hömoglobisynthese, die überwiegend bei den Einwohnern der Mittelmeerländer angetroffen wird. Genauer gesagt handelt es sich um eine Krankheit des Blutfarbstoffes, des Hämoglobins. Das Hämoglobin befindet sich in den roten Blutkörperchen, den Erythrozyten. Es ist eine kompliziert gebaute Eiweißverbindung, die die Aufgabe hat, den Sauerstoff von der Lunge über das Blutgefäßsystem zum Gewebe zu transportieren.

Bei der Mittelmeer-Anämie (Anämie = Blutarmut) ist nun durch einen vererbbaren Defekt die Bildung des normalen Blutfarbstoffes gestört und an seiner Stelle werden unreife und fehlerhafte Blutfarbstoffe gebildet, die zu einem verfrühten Absterben der Blutkörperchen führen.

Beim Zusammentreffen zweier krankhafter Erbanlagen kommt es zum tatsächlichen Auftreten von Krankheitserscheinungen unterschiedlicher Schweregrade. Man unterscheidet zwischen der Thalassämie minor, die fast immer symptomlos verläuft und der Thalassämie major. Letztere verläuft ungleich schwerer. Die im peripheren Blut zirkulierenden Erythrozyten haben eine verkürzte Lebensdauer. Sie beträgt bei an Mittelmeeranämie erkrankten Kindern 30 bis 40 Tage (bei gesunden Menschen ca. 130 Tage), der Sauerstoff wird nicht mehr transportiert, die Kinder werden teilnahmslos, ersticken innerlich und sterben. Das Krankheitsbild beginnt im zweiten Lebensjahr mit starker Blässe und Milzvergrößerung. Die körperliche Entwicklung ist beeinträchtigt und es kommt zu einer Lebervergrößerung.

2. Wie wird THALASSÄMIE behandelt?

Die Therapie besteht sehr oft in der Entfernung der Milz sowie in häufigen Bluttransfusionen. Beim Zerfall der roten Blutkörperchen wird Eisen freigesetzt. Dieser Prozeß schreitet mit der Zahl der Transfusionen fort und würde ohne medikamentöse Behandlung zum Tode der Patienten führen. Aus diesem Grund muß dem Körper Desferal oder L 1 zugeführt werden, Medikamente, die die Eisenablagerungen aus den betroffenen Organen schwemmen. Die Zuführung des Desferals erfolgt mit Hilfe einer Desferalpumpe, deren Hauptbestandteil ein Präzisionsuhrwerk ist, mit dessen Hilfe das Desferal gleichmäßig in die Blutbahn gedrückt wird. Diesen Apparat, der etwa so groß ist wie eine Zigarrenkiste, können die Kinder bei sich tragen, so daß sie während der Zuführung des Medikaments - Desferal muß über eine Dauer von zwölf Stunden gleichmäßig intravenös zugeführt werden - nicht ans Bett gebunden sind. L 1 wird oral eingenommen, ist aber für viele Familien noch unerschwinglich.

Vollständig heilbar ist Thalassämie nur durch eine Knochenmarktransplantation. Dazu wird einem völlig gesunden Familienangehörigen, in der Regel dem Bruder oder der Schwester eines kranken Kindes, Knochenmark entnommen und dem Patienten zugeführt. Aussicht auf Erfolg besteht bei einer derartigen, äußerst aufwendigen Operation besonders, wenn das kranke Kind noch sehr jung ist. Außerdem müssen Spender und Empfänger in etwa das gleiche Alter haben. Die Heilungschancen stehen heute bei einer Knochenmarktransplantation etwa 4 : 1.

Ein an Thalassämie erkranktes Kind, dem keine Hilfe zuteil wird, hat eine Lebenserwartung von ca. drei Jahren; ein erwachsener Mensch, der zwar nicht operiert werden kann, jedoch seinem Körper regelmäßig Desferal zuführt und Bluttransfusionen über sich ergehen läßt, kann bis zu 30 Jahre alt werden.

3. Die besondere Situation auf Sardinien

Sardinien ist eine arme Insel. Der Anteil an den Fahrtkosten zu den häufigen Bluttransfusionen in das meist weit entfernte, nächste Krankenhaus bedeutet für viele Familien im Hinblick auf die geringen Durchschnittseinkommen ein schwere Belastung.

Zwar sind auf Sardinien die wichtigsten medizinischen Einrichtungen vorhanden, aber längst nicht in der von uns gewohnten und erwarteten Dichte. Die Entfernungen zum nächsten Krankenhaus, dazu die schlechten Verkehrsverbindungen lassen einen Arztbesuch leicht zu einer Tagestour werden.

Durch den hohen Blutbedarf werden überdurchschnittlich viele Blutkonserven benötigt. Träger dieser Krankheit (ca. 20% der Gesamtbevölkerung) scheiden von vornherein als Spender aus.

4. Was hat die Bundeswehr mit der THALASSÄMIEHILFE zu tun?

Die Arbeitsgemeinschaft THALASSÄMIEHILFE ist heute ein Zusammenschluß von mehreren Bundeswehrdienststellen in Deutschland und auf Sardinien sowie ehrenamtlichen, zivilen Helfern, deren Arbeit durch das Kommando 1. Luftwaffendivision koordiniert wird. Sie haben sich das Ziel gesetzt, in größtmöglichem Umfang Blut für Transfusionen bereitzustellen, in Not geratene Familien zu unterstützen und den Kontakt zur Bevölkerung auf Sardinien, aber auch zu Ärzten und Presse zu halten.

Familien, die sich hilfesuchend an die THALASSÄMIEHILFE wenden, können materielle Hilfe, aber auch Unterstützung z. B. bei Behördengängen erhalten.

Neben der direkten Hilfe auf Sardinien sind die Helfer in Deutschland gefordert, mit Phantasie und Engagement die finanziellen Mittel zu sammeln, die diesen kranken Sardischen jungen Menschen das Leben etwas leichter machen können.

Info für Interessierte:

Thallassämie-Hilfe
Koordinationsbüro Deutschland
Rintheimer Querallee 11
76027 Karlsruhe
Tel.: 0721/60221 App. 259/263
FAX: 0721/606436
Spendenkonto:
Sparkasse Karlsruhe 10 420 180 (BLZ 660 501 01)

Förderverein für krebskranke Kinder e.V. Karlsruhe

besonders förderungswürdiger gemeinnütziger Verein

Der Förderverein für krebskranke Kinder e.V. Karlsruhe bittet um Ihr Interesse:

Der Förderverein wurde im November 1988 gegründet, um wirkungsvoll, konkret und rechtlich verbindlich betroffene Familien helfen zu können.

Die Mitglieder des Vereins sind betroffene Eltern, professionelle Helfer (Lehrer, Sozialarbeiter, Ärzte, Krankenschwestern) und zunehmend auch interessierte Bürger.

Auf Bundesebene sind wir Mitglied im Dachverband der Förderkreise, der Deutschen Leukämie-Forschungshilfe und auf Landesebene bei der Arbeitsgemeinschaft baden-württembergischer Förderkreise unter dem Vorsitz von Carl Herzog von Württemberg.

Die zur Verfügung stehenden Gelder setzen sich aus Mitgliedsbeiträgen, gerichtlichen Zuwendungen, Spenden sowie aus den Erlösen vielfältiger Veranstaltungen, die zugunsten des Vereins stattfinden, zusammen.

Die Arbeit und Ziele des Fördervereins umfassen folgende Schwerpunkte:

- die Unterstützung des Elternstammtisches und Organisation von Veranstaltungen mit krankheitsbezogenen Themen
- Aufzeigen von Möglichkeiten psychosozialer und sozialrechtlicher Betreuung
- Auseinandersetzung mit der Krankheit, Umgang mit der Angst
- Hilfestellung bei Schul- und Familienproblemen
- persönliche Betreuung in Form der Selbsthilfe
- Unterstützung bei finanziellen und beruflichen Problemen
- Zuschuß bei Kuren, Fahrtkosten, Nachhilfeunterricht, Haushaltshilfen, Kleidern und Spielsachen (falls kein anderer Träger zuständig ist)
- Unterhalt einer Elternwohnung gegenüber der Kinderklinik zur kostenlosen Nutzung für Betroffene während der Behandlung des Kindes.

In der Kinderklinik wurden vom Verein ein Aufenthaltsraum für Eltern eingerichtet und der Ausbau und die Verglasung des Balkons der onkologischen Station übernommen.

Beim Bau der Nachsorgeklinik Tannheim durch die Arbeitsgemeinschaft baden-württembergischer Förderkreise hat sich der Verein finanziell beteiligt.

Diesen Ausgaben stehen - durch streng ehrenamtliche Tätigkeiten - sehr geringe Verwaltungskosten gegenüber.

Auch künftig werden die betroffenen Familien und deren Förderung in allen Bereichen im Mittelpunkt der Arbeit stehen.
Dafür wünschen wir uns Ihr Verständnis und Ihre Unterstützung.

Info für Betroffene und Interessierte:

Margareta Höfele
Hubertusallee 21
76135 Karlsruhe
Tel: 0721/861005
Spendenkonto: Sparkasse Karlsruhe 9 436 791 (BLZ 660 501 01)

„Alte“ Computer in DFÜ-Netzen

Auf meine Bitte hin haben Rainer Kulhanek und Stefan Sommer noch während des Treffens ein paar Informationen zusammengestellt, wo im Wirrwarr der DFÜ Informationen zu „alten“ Computern zu finden sind. Wie man sieht, gibt es da einiges!

8-Bit-Net: (Fido-kompatibel)

Bullet_BBS, 49-7351-827162
Eremite-BBS, 49-8851-5710
CHiLL_ouT_Z0Ne!, 49-821-2290526
TransVision, 49-511-834572
CCS-Base, 49-8801-2453 (führt Helmut Jungkunz' Znode 51 weiter)
Die Znode 51 ist jene „berühmte“ Mailbox, die neben einer unheimlichen Menge an CP/M - Software wie z.B. der gesamten SIG/M-Sammlung und den Dateien zu Jay Sages Z-System auch für Rechner wie den Robtron-Rechnern (KC-Serie) und vieles andere zu Z80-Systemen anbietet.

Beim 8-Bit-Net sollte man darauf hinweisen, daß ein größerer Teil der User zu den Kids gehört. D.h. Sprache und Verhalten sind ein wenig gewöhnungsbedürftig. Nichts desto trotz oder gerade deswegen ist es schon bemerkenswert, daß gerade diese Altersgruppe sich um Rechner mit dem Z80 kümmert!

CPC/Eumel/GothicNet: (Fido-kompatibel)

Kittis Box, 49-234-682887

Eine Informationsquelle für Freunde des Z80-Prozessors und „alter“ Rechner sind natürlich die einschlägigen Bretter der verschiedenen Netze wie z.B. im FIDO-Netz: Comp_old.ger, cpmtech, Ti_echo, Trsmod_134.

Aber auch im Internet, dem vielgerühmten, gibt es entsprechende Bretter. Die Namen sind mir gerade nicht geläufig, aber unter alt.comp.os.cpm etc. müßte man auf jeden Fall fündig werden.

```
CCCCCC  PPPPPP    CCCCCC  NNN     NN  EEEEEE  TTTTTTTT
CC       PP   PP  CC       NNNN    NN  EE          TT
CC       PPPPPP   CC       NN NN   NN  EEEEE       TT
CC       PP       CC       NN  NN NN   EE          TT
 CCCCCC  PP        CCCCCC  NN    NNNN  EEEEEE      TT
```

Das Netzwerk für Amstrad/Schneider 8-Bit-(Z80-)Computer und kompatible.

Das CPCNet ist ein Fido-kompatibles Netzwerk (kann aber auch im ZConnect Format bezogen werden) über die guten alten Schneider/ Amstrad 8-Bit-(Z80-)Computer CPC (464, 664, 6128, 464+, 6128+), GX4000, PCW (8256, 8512, 9256, 9512, 9512+), die kompatiblen KCCs aus der ehemaligen DDR und die diversen CPC-Emulatoren (z.B. für PC, Amiga, Acorn, Unix, u.a.). Es soll eine Verbindung und einen Erfahrungsaustausch zwischen den wenigen noch verbliebenen Nutzern der Amstrad/Schneider 8-Bit-Rechner ermöglichen. Dazu wurden 5 Areas eingerichtet:

CPCNET.SOFTWARE: Softwareprobleme, Fragen, Tips (von Spielen, über Textverarbeitungen, bis zu Branchenlösungen)
CPCNET.HARDWARE: Hardwareprobleme, Fragen, Tips und Basteleien
CPCNET.CPM: Probleme, Fragen, Tips und Tricks zum -Control Programm for Microcomputers” (V2.x bis 3.x)
CPCNET.CODER: Programmierfragen, Tips und Tricks (von Z80-Assembler, über Locomotiv Basic, bis zu Turbo Pascal)
CPCNET.CHAT: Alles was nicht in die anderen 4 Areas paßt ist hier zu Hause (unteranderem werden hier regelmäßig FAQs und andere Infos gepostet)

In diesen Areas können sich die Amstrad-8-Bit-User-treffen” und Mails austauschen. Technisch ist zu sagen, dass das CPCNet ein Unternet des EumelNet ist, welches den Vorteil hatte, daß nicht extra eine neue Netzstruktur aufgezogen werden mußte. Außerdem bietet es den CPCNet-Lesern auch noch die Möglichkeit, ohne Aufwand (auf Wunsch) am EumelNet teilzunehmen. Für nähere Infos zum CPCNet (und entsprechend auch zum EumelNet) stehen die folgenden Möglichkeiten zur Verfügung. Bei „Kittis Box“ kann man per Fido-Filerequest unter MagicNamen EUMELINFO die Datei EUMELINF.LZH bekommen. In dieser Datei sind noch nähere Informationen und ein Antrag enthalten.

Thorsten Franke (CPCNet-HQ)
Kittis Box (02 34) 68 28 87
FidoNet: 2:2448/53
CPCNet: 13:490/2
EMail: Botho@kitti.ftg.donut.de

Zum gleichen Thema hier noch eine Mail aus dem FIDO-Netz:

Empfaenger : /FIDO/COMP_OLD.GER (All)
Absender : Christoph Bruder @ 2:2480/3135 (CHiLL ouT ZoNe BbS, Augsburg)
Betreff : cooles 8Bit-Net (CPC/C64 bis jetzt..)!!!!
Datum : Di 04.03.97, 20:11 (erhalten: 10.03.97)
Groesse : 5723 Bytes

Hi All!

=== Cut ===

```
-----------------------------------------------------------------------
        888888   8888888        888       888    88            888
       888  888  888   88  888  888       8888   88            888
       888  888  888   88       888       88888  88   888888   888
        888888   8888888   888  88888     888 88 88  888   88  88888
       888  888  888   88  888  888       888  8888  88888888  888
       888  888  888   88  888  888   88  888   888  888       888   88
        888888   8888888   888   8888888  888    88   8888888   8888888
v1.0
-----------------------------------------------------------------------
```

Hallole und Moin! :-)

Vor knapp einem halben Jahr wurde von einigen 8Bit Computerfans das 8BitNet
neu aufgezogen.

Anfangs war zwar ein reines CPC-Netz geplant, aber da ein systemuebergrei-fendes Netz für viele User und Boxbetreiber interessanter ist kam uns eben die Idee zum 8BitNet.

"8BitNet? für was soll das denn gut sein?"

Viele Leute die heute hauptsächlich mit ihrem PC arbeiten haben sich von
ihrem 8Bit Computer nicht trennen können, da er ja jahrelang seinen Dienst getan hat. Da er meist der erste Computer war, hängt man trotz HiTech an ihm. Die erste Liebe wird halt nie vergessen! ;-)

Es gibt sie immer noch, die Freaks die auf den alten Kisten auf Teufel-komm-raus coden und mit ihren Demos auf den 8Bittern den typischen Visual Basic "DOSen-Coder" aus der Fassung bringen. Es wird zwar keine (bzw. kaum) kommerzielle Software mehr für unsere 8Bitter produziert, aber PD, Shareware, Diskmags und vor allem Demos werden immer noch von den Freaks programmiert. Dies war bisher jedoch nur einer kleinen Gruppe von enthusiastischen Scenern vorbehalten und das möchten wir mit dem 8BitNet ändern.

So bieten wir neben Nachrichtenbrettern für die verschiedenen 8Bitter auch welche für Systemuebergreifenden Themen an. Ebenso befindet sich ein Filenetz für die oben genannte Software im Aufbau.

Nachdem beim ersten Start des 8BitNet viel zu viele Echos angelegt wurden und in den einzelnen Nachrichtenbrettern kaum Nachrichten kamen, haben wir jetzt deutlich weniger Bretter angelegt, dafür ist das Nachrichtenaufkommen jetzt wesentlich besser. Wenn die Bretter vom Nachrichtenumfang zu "groß" werden, werden diese auch wieder aufgesplittet.

Bis jetzt ist das Netz ziemlich CPC-lastig. Es gibt aber auch schon einen C64 Node. Willkommen sind natürlich auch die Nutzer von anderen 8Bit Systemen(Atari, Sinclair, Commodore, ... und alle, die mir jetzt nicht einfallen).

Es gibt es folgende Echos (Stand 08/96)

```
-------------------------------------------------------------
*** Allgemein *************************
8BIT.SMALLTALK.GER   Ueber alles Moegliche (deutsch)
8BIT.SMALLTALK       about everything (english)
*** CPC *******************************
8BIT.CPC.GER         Ueber alles Moegliche (deutsch)
8BIT.CPC             about everything (english)
*** C64 *******************************
8BIT.C64.GER         Ueber alles Moegliche (deutsch)
8BIT.C64             about everything (english)
-------------------------------------------------------------
```

Es existieren folgende Fileareas (Stand 08/96)

```
-------------------------------------------------------------
*** Allgemein *************************
8BIT.NODELIST        8BitNet Nodelisten
*** CPC *******************************
8BIT.DEMO.CPC        Demo's
8BIT.DISKMAG.CPC     Diskmag's
8BIT.EMUDISK.CPC     Emulator Diskimages
8BIT.GAME.CPC        Game's
8BIT.MISC.CPC        Misc CPC
8BIT.PC.CPC          PC related CPC
8BIT.ROM.CPC         ROM's
8BIT.SOUND.CPC       Sound's
8BIT.SOUND128.CPC    128k Sound's
8BIT.SOUND64.CPC     64k Sound's
8BIT.UTILITIE.CPC    Utilitie's
-------------------------------------------------------------
```

Sobald wir noch einen C64-User haben, werden auch die entsprechenden C64
Bretter angelegt.

Solltest Du z.B. ein Fanzine für einen 8Bitter herausbringen und Dich aktiv am Netzgeschehen beteiligen, kannst du Dir nicht nur Artikel für Dein Fanzine per Netmail schicken lassen, sondern auch tüchtig Werbung für Dein Mag machen! ;-))

Leider hat es ein so spezielles Netz wie unseres etwas schwer sich zu
etablieren. Deshalb suchen wir neben HUBs, HOSTs und NODEs vor allem User, die hier schreiben. Ihr könnt hier Fragen stellen, Infos verteilen oder euch einfach nur unterhalten.

Also, wenn Du noch an 8Bit Computern interessiert bist, dann melde dich!

Solltest Du noch Fragen zum 8BitNet haben, dann setze Dich bitte mit einem der folgenden Nodes in Verbindung:

Name	Fido	BBS	Modem
Stefan Kuhs	2:2487/9017	+49-DOWN	V32b
Wolfgang Noisternig	2:241/1115 2:241/1116	+49-511-834572 +49-511-836575	ZyX iSDN

Ciao,
Stefan Kuhs

Bye,
Wolfgang Noisternig

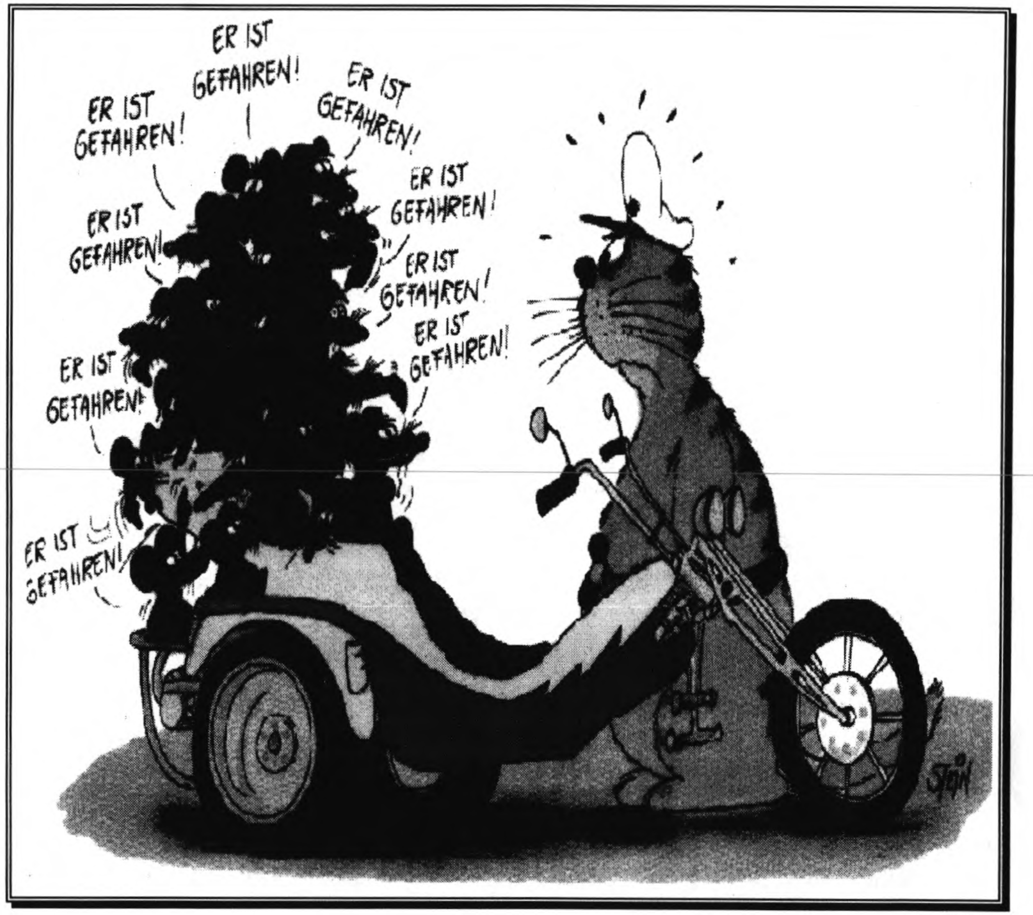
=== Cut ===

könnt bei mir auch Point/Node werden..

Cee Ya!

Christoph

– FMail/386 1.02
* Origin: oOo GEt PhUNkY StAPh iN Chill OUt ZONE! NOhW! oOo (2:2480/3135)



Impressum

1. Vorsitzender	**Hartmut Obermann** Wilhelm-Baur-Straße 8 76135 **Karlsruhe**	Tel.: 0721/85 40 68 Fax: 0721/85 40 68
2. Vorsitzender	**Gerald Schröder** Am Schützenplatz 14 21218 **Seevetal**	Tel.: 04105 /26 02
Hardewarekoordinator	**Andreas Magnus** Bismarckstraße 29 45879 **Gelsenkirchen**	Tel.: 0209 /87 02 30
NewDOS-Diskothekar	**Oliver Volz** Am Ochsenwald 37A 70565 **Stuttgart** (Rohrerhöhe)	Tel.: 0711/74 40 51
CP/M-Diskothekar	**Fritz Chwolka** Saarstraße 34 52457 **Aldenhoven**	Tel.: 02464 /89 20
C-128-Diskothekar	**Günther W. Braun** Postfach 80 02 26 81602 **München**	
Clubbücherei	**Rainer Schmitz** Küferweg 12/1 73099 **Adelberg**	Tel.: 07166 /13 97
Redaktion	**Jens Neueder** Rudolf-Then Straße 32 74544 **Michelbach /Bilz**	Tel.: 0791 /4 28 77 FAX:07971 /2 50 55

Bankverbindung des **CLUB 80**

Postgirokonto Sonderkonto CLUB 80
Obermann H., 76135 Karlsruhe
Konto Nr. 496 071-605
Postgiroamt Frankfurt, **BLZ 500 100 60**

Autoren — Die Redaktion bedankt sich bei den im **INHALTSVERZEICHNIS** genannten Autoren für die Mitarbeit an der Club-INFO. Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.

Schluß

Hallo Club 80'er,

nun ist es endgültig vorbei mit dem CLUB 80 !

Mit einem lachenden und einem weinenden Auge haben wir den CLUB 80 an der wahrlich **letzten** Jahreshauptversammlung (...ist leider auch schon eine Weile her, aber ich hatte bisher keine Zeit mich um das INFO zu kümmern) zu Grabe getragen. In Erinnerung an 12 Jahre Clubtätigkeit fiel es uns nicht ganz leicht, diesen Entschluß zu fassen. Er ist aber, zurückblickend auf die letzten beiden Jahre, mit Sicherheit ein vernünftiger Entschluß.

Die PC-Zeiten haben sich doch enorm gewandelt. Selbst Firmen, die zur damaligen Zeit Marktführer waren, sind heute nicht einmal mehr auf der CeBIT vertreten. Der PC-Kompatible und das Betriebsystem mit den vielen Fensterchen haben allem den Rang abgelaufen. Obwohl doch zu mancher Zeit schon besseres und/oder innovativeres geboten wurde. Hier sei einfach bloß mal an den 68000 erinnert, der dem damaligen XT doch in vielem voraus war.

Ich möchte hier dieses letzte INFO auf alle Fälle noch einmal nutzen, um den Club 80´ern zu danken, die durch ihre aktive Arbeit an und in dem CLUB-INFO einen wesentlichen Beitrag zum Clubleben geleistet haben. Gerne habe ich mit Euch zusammen dieses INFO gestaltet.

Vielleicht sieht man sich mal oder hört voneinander. Um mit den Worten meines Arbeitgebers zu sprechen:

Es wäre nett, wenn wir in Verbindung bleiben.

Bis dahin die besten Wünsche für Euch.
Euer

J. Neueder

Liste der CLUB 80 - Bücher

Nr.	Gew.	Seiten	ISBN	Aufl.	Jahr	Titel	Autor	Verlag	Thema	Kommentar
2	250	300	0-0000-00-000	1	1982	Taschenlexikon fuer alle Personal-Comp.	Michael Scharfenberger	Markt & Technik	Allgemein	Recht brauchbar, da die ganz alten Datenschleudern aus der goldenen Computeraera ausfuehrlich behandelt werden.
3	400	300	3-8023-0703-8	2	1983	Computersp. & Knobeleien progr. in BASIC	Ruedeger Baumann	Vogel-Verlag	Hochsprache	Der Titel spricht fuer sich. Sehr gut aufgemacht.
4	100	66	3-923220-05-7	1		Mein Homecomputer - Eine Verbraucherfibel	./.	Vogel-Verlag	Allgemein	Einige Tips & Erklaerungen rund um den Computer
5	250	148	3-921682-93-2	1	1982	Programmieren mit dem ZX81	E. Floegel	W. Hofacker Verlag	Hochsprache	Wie benutze ich ein Spielzeug zum Programmieren. Das Buch enthaelt eine bunte Sammlung an BASIC-Progr. zu den unterschiedlichsten Themen.
6	150	125	0-907080-88-X	1	1983	Games for your TRS-80	./.	Virgin Books, USA	Hochsprache	Einige Spiele in BASIC, Englischkenntnisse erforderlich!
7	220	142	0-918398-18-5	3	1979	Introduction to TRS-80 Graphics	Don Inman	Dilithium Press	Hochsprache	Es wird sehr gut auf die Technik der Grafikprogr. eingegangen. Die Vorgehensweise & Technik wird anhand guter Beispiele erklaert. Englischkenntn. erfordl.
8	480	196	0-916688-19-4	1	1980	More BASIC Computer Games	David H. Ahl	Creativ Computing Press	Hochsprache	Spiele satt, aber in Englisch!
9	300	104	3-7723-7421-2	1	1984	Dateien, Listen & Verzeichnisse	Rudolf Busch	Franzis' Verlag	Hochsprache	Eine Software-Sammlung in BASIC zum Thema Dateiverwaltung
10	350	128	3-7723-7441-7	1	1984	Matrix-Operationen	Rudolf Busch	Franzis' Verlag	Hochsprache	Eine Software-Sammlung in BASIC zum Thema Matrix-Mathematik
11	390	267	0-918398-27-4	7	1980	TRS-80 Programms	Tom Rugg & Phil Feldmann	Delithium Press Beaverton	Hochsprache	Diverse Programme in BASIC mit Erlaeuterungen (Spiele etc.) Englischkenntnisse erforderlich!
12	350	110	3-88965-001-5	1	1983	Programme & Tricks fuer Genie I & II	Clemens Becher, Frank Seger	./.	Hochsprache	Jede Menge Listings in BASIC (Spiele etc.)
13	250	70	3-7723-7531-6	1	1984	Alles ueber Peek & Poke	Heiko Requardt	Franzis' Verlag	Hochsprache	Eine Software-Sammlung zum Thema Tastatur,
16	500	89	0-0000-00-000	4	1985	Das Modem Sonderheft	./.	Franzi' Verlag	Listing	Das Modem-Sonderheft mit Grundlagen & Schaltungen.
17	200	66	0-0000-00-000	1	1984	Genie Data	Fa. TCS Computer	TCS Computer GmbH	Listing	Eine Art Club-Zeitung fuer Besitzer eines Genie 1/2/3 Computers
17	200	58	0-0000-00-000	1	1985	Genie Data	Fa. TCS Computer	TCS Computer GmbH	Listing	Anwenderzeitung fuer Genie-Besitzer
17	175	54	0-0000-00-000	1	1984	Genie Data	Fa. TCS Computer	TCS Computer GmbH	Listing	Anwenderzeitung fuer Genie-Besitzer
17	125	42	0-0000-00-000	1	1985	Genie Data	Fa. TCS Computer	TCS Computer GmbH	Listing	Anwenderzeitung fuer Genie-Besitzer
17	75	24	0-0000-00-000	1	1984	Genie Data	Fa. TCS Computer	cc Computer Studio GmbH	Listing	Anwenderzeitung fuer Genie-Besitzer
17	200	58	0-0000-00-000	1	1984	Genie Data	Fa. TCS Computer	TCS Computer GmbH	Listing	Anwenderzeitung fuer Genie-Besitzer
17	150	58	0-0000-00-000	1	1985	Genie Data	Fa. TCS Computer	TCS Computer GmbH	Listing	Anwenderzeitung fuer Genie-Besitzer
17	200	62	0-0000-00-000	1	1984	Genie Data	TCS Computer GmbH	TCS Computer GmbH	Listing	Eine Art Club-Zeitung fuer Besitzer eines Genie 1/2/3 Computers
18	1000	430	0-936200-25-1	1	1983	BASIC Disk-I/O faster & better	Lewis Rosenfelder	IJG Inc. USA	Know-How	Englischkenntnisse erforderlich!
19	650	288	0-936200-03-0	1	1981	BASIC Faster & Better & o. Mysteries	Lewis Rosenfelder	IJG Inc. USA	Hochsprache	Englischkenntnisse erforderlich!
20	800	335	0-936200-02-2	1	1982	The Cutom TRS-80 & other Mysteries	Dennis Bathory	IJG Inc. USA	Hardware	Wer gern an seinem TRS-80 bastelt liegt mit diesem Buch genau richtig. Englischkenntnisse erforderlich!
22	650	300	0-936200-07-3	1	1982	TRSDOS 2.3 Encoded & other Mysteries	James Lee Farvour	IJG Inc. USA	Assembler	Das gesamte Betriebssystem offengelegt! Englischkenntnisse erforderlich!
23	350	124	3-88623-007-4	1	1982	TRS-80 Disk & andere Geheimnisse	H. C. Pennington	Interface Age, Muenschen	Know-How	Endlich einmal etwas ueber den TRS-80 in deutsch!
24	490	306	0-0000-000-00	1	1981	TRS-80 Data File Programming M-1/3	Radio Shack	John Wiley & Sons, Inc.	Hochsprache	Englischkenntnisse erforderlich!
25	260	224	0-0000-000-00	1	1981	Programming Technics for L-II BASIC	William Barden, Jr.	Radio Shack	Hochsprache	Englischkenntnisse erforderlich! Wer mit dem Enlischen zurechtkommt, findet eine gute Grundlageneinfuehrung in die Programmierung des TRS-80 vor.
27	550	351	0-471-06466-1	2	1980	TRS-80 Level-II BASIC	Robert L. Albrecht	John Wiley & Sons Inc., USA	Hochsprache	Das Handbuch zum Level-II BASIC.
28	800	342	0-07-018492-5	1	1980	TRS-80 Programming Style	Thomas Dwyer	Radio Shack	Hochsprache	Englischkenntnisse erforderlich
29	600	56	3-921682-56-8	1	1979	The First Book of 80-US	80-US	Ing. W Hofacker Verlag	Allgemein	Ein Re-Print einiger Ausgaben der US-Zeitschrift 80-US
30	380	236	0-88006-062-X	1	1983	The Rest of 80 ./.	Wayne	Green Books	Assembler	Ein Software-Querschnitt rund um den TRS-80. Englischkenntnisse erforderlich!

Liste der CLUB 80 - Bücher (2)

Nr.	Gew.	Seiten	ISBN	Aufl.	Jahr	Titel	Autor	Verlag	Thema	Kommentar
31	380	144	3-7723-6671-6	1	1979	Z80-Applikationsbuch	Michael Klein	Franzis-Verlag	Assembler	Eine Einfuehrung in die Programmier- & Interfacetechnik zum Z80
32	450	424	0-000-0000-00	1	1980	Z80-Assembler Handbuch	Elcomp	Ing. W. Hofacker	Assembler	Einer Auflistung aller Z80-Befehle mit Erlaeuterungen.
33	800	606	3-88745-006-X	1	1980	Programmierung des Z80	Rodney Zaks	Sybex-Verlag	Assembler	Die Bibel zum Z80-Prozessor
34	400	299	3-8023-0751-8	1	1984	Daten - Disketten - Dateien	Hans-Joachim Sacht	Vogel-Verlag	Hochsprache	Effektive Dateienverwaltung mit MBASIC
35	300	185	3-921682-64-9	1	1981	Anwenderprogramme fuer TRS-80	M. Stuebs	W. Hofacker Verlag	Hochsprache	Eine Sammlung von Geschaeftsprogr, Utilities in BASIC fuer den TRS-80.
36	275	202	3-921682-45-2	2	1981	Programmieren mit dem TRS-80	M. Stuebs	W. Hofacker Verlag	Hochsprache	Zweiter Band mit gleichem Inhalt.
36	275	202	3-921682-45-2	2	1981	Programmieren mit dem TRS-80	M. Stuebs	W. Hofacker Verlag	Hochsprache	Einfuehrung in den TRS-80 u. seine Programmierung nebst einigen BASIC Programmen.
38	330	222	3-88707-021-6	1	1984	Textsystem in BASIC & Z80-Assembler	Gerd Scheller	W.-D. Luther Verlag	Assembler	Wie No. 36. aber deutlich verbesserte Version des TESY-Textsystems. Auf einem TRS-80 geschrieben.
39	250	145	3-620-00122-7	1	1985	Textsystem in BASIC & Z80-Assembler	Wolfgang Weber	W.-D. Luther Verlag	Hochsprache	
40	400	145	3-7643-1552-0	1	1984	Einfache Peripheriegeraete im Selbstbau	Owen Bishop	Birkhaeuser Verlag	Hardware	Es werden einige Hardware-Projekte wie z.B. eine Kaffemschinen-Steuerung, EPROM Karte etc beschrieben. Durchaus anregen.
41	450	285	3-8023-0771-2	1	1984	Start in die Computergrafik	Fred Wagenknecht	Vogel-Verlag	Hochsprache	Viele kurze Progr., die aufzeigen, wie man mit der TRS-80 Kloetzchengrafik Grafik erzeugen kann mit guten Erlaeuterungen.
42	460	180	0-0000-00-0	1	1976	Level-II BASIC Reference Manual	Radio Shack	Radio Shack	Hochsprache	
43	350	222	3-473-44010-8	1	1984	BASICODE - Die komp. Computersprache	M. Wiegand, M. & H. Fillinger	Elektor-Verlag	Hochsprache	Die kompatible Computersprache, 14 Prog. fuer ueber 25 Homecomputer. Enthaelt viele BASIC-Listings (allg. Themen)
44	150	112	3-7723-1691-3	1	1983	Kleiner BASIC-Wortschatz	Rufdolf Busch	Franzis' Verlag	Hochsprache	Eine Kommentierte Auflistung von BASIC-Befehlen mit Beispielen.
45	460	183	0-916688-07-0	1	1980	BASIC Computer Games	David H. Ahl	Creativ Computing Press	Hochsprache	Spiele satt, aber in englisch!
46	400	350	0-8306-1205-X	1	1980	PASCAL	David L. Heisermann	Tab Books Inc.	Hochsprache	Eine Einfuehrung in PASCAL und wie man diese Programmiersprache richtig anwendet. Viele Listings. Englischkenntnisse erforderlich!
47	275	221	0-000000-00-0	1	1979	TRS-80 Assembly-Language Programming	William Barden, Jr.	Radio Shack	Assembler	Text in Englisch!
49	430	192	0-931988-54-3	1	1981	Some Common BASIC Programms	Lon Poole, Mary Borchers, Karl Koessel	McGraw-Hill	Hochsprache	Enthaelt etliche interessante Programme
51	700	288	3-89090-086-0	2	1986	Der Einstieg in C	Paul M. Chirlian	Markt & Technik	Hochsprache	Soll eine Einweisung in C sein. Es gibt besseres zu diesem Thema.
52	500	303	3-486-20258-8	6	1986	PASCAL und PASCAL-Systeme	Rudolf Herschel	Oldenburg-Verlag	Hochsprache	Beschreibung der Programmierspr. PASCAL
54	600	443	3-528-14443-2	2	1986	Mikrocomputer Aufbaukurs	E. Kaier, E. Rudolfs	Vieweg Verlag	Hochsprache	Das Buch informiert durch ausfuehrliche Beispiele grundlegend ueber die Anwendungsmoeglichkeiten die Turbo Pascal unter CP/M, MSDOS und MSX-DOS bietet.
55	300	295	3-519-10100-9	2	1986	Programmierung in Modula-2	Dal Cin, Lutz Risse	Teubner Studienskripten	Hochsprache	Eine Einfuehrung in das modulare Progr. mit MODULA-2. Hervorragender Inhalt! Eignet sich zum Einarbeiten in das Konzept der Sparache.
56	1000	682	3-89090-219-7	1	1987	Das Z80-Buch	Eberhard Zehender	Markt & Technik	Assembler	Die Z80-Bibel! Es werden hier querbeet alle fuer den Z80 relevanten Gebiete in Assembler programmiert, sogar Gleitkommaroutinen!
57	600	321	3-89090-087-9	3	1985	CP/M - Anatomie eines Betriebssystems	Jack D. Dennon	Markt & Technik	Assembler	Ein Wissen-Torpedo mit Informationen zum Thema CP/M. Sehr brauchbar.
58	190	111	3-620-00113-8	1	1985	Listing-Verwaltung in MBASIC	H.-J. Rothe	W.-D. Luther Verlag	Hochsprache	Ein kompl. BASIC-Progr. zum Verwalten von Zeitschriften-Artikeln aller Art. Ausfuehrliche Erkl. mit Beispielausdrucke
59	120	91	3-88707-040-2	1	1984	dBASE-II im Ueberblick	Thomas LAuer	W.-D. Luther Verlag	Know-How	

Liste der CLUB 80 - Bücher (3)

Nr.	Gew.	Seiten	ISBN	Aufl.	Jahr	Titel	Autor	Verlag	Thema	Kommentar
61	150	58	3-528-04263-X	1	1984	PASCAL - Datenstrukt. & prakt. Anwendg.	D. Herrmann	Vieweg Verlag	Hochsprache	Die Programmierung von div. Algorithmen Sehr ausfuehrlich m. Flussdiagramm und Ergebnisausdrucken.
62	150	80	3-7724-0367-0	1	1979	BASIC - Grundlagen & Beispiele	Guenter Abeldt	Frech-Verlag Stuttgart	Hochsprache	Es werden mathematische Probleme behandelt. Empfehlenswerter Inhalt!
63	330	224	39-620-001324	1	1985	Program. in MBASIC unter CP/M	Wilfried Plieth W.-D.	Luther Verlag	Hochsprache	Tips & Trick zum optimaleren Programmieren in BASIC unter CP/M. Viele Beispiel-Listings!
64	120	190	3-88707-022-4	1	1984	RPNL - Eine FORTH-aehnliche Sprache	Gustav Wostrack W.-D.	Luther Verlag	Assembler	Der Compiler ist in Z80-Assembler abgedruckt. Laeuft in der abgedruckte Form unter CP/M.
66	290	136	3-922120-18-0	1	1982	Personal Computer Lexikon	Computer Persoenlich	Markt & Technik	Allgemein	Wie der Titel es schon sagt. Der Inhalt erscheint mir hinreichend vollstaendig. Ein
67	475	245	3-8032-0660-0	2	1980	Mein Computer 2^10 Fragen & Antworten	Rich Didday	Vogel-Verlag	Allgemein	Viele Fragen mit Antworten in Form eines bebilderten Computerlexikons.
68	350	295	3-921682-35-5	4	1979	Z8, Z80, Z8000 Mikrocomputer-Technik	H.-P. Blomeyer-Bartenstein	W. Hofacker Verlag	Assembler	Es werden der Z8, Z80 & Z8000 behandelt
69	275	270	3-925074-10-4	2	1984	Das DOS-Buch	Hartmut Grosser	Roeckrath Verlag	Assembler	Wer dieses Buch nicht kennt ...Eine fundierte Erklaerung von NewDOS80, GDOS, und Colour-DOS fuer den TRS-80
70	250	199	3-7029-0142-6	1	1979	Tischcomputer - BASIC fuer Anfaenger	Juilius Schaerf	R. Oldenbourg Verlag	Hochsprache	Ein sehr gutes Uebungsbuch fuer Anfaenger, die sich fundiert in BASIC einarbeiten wollen. Empfehlenswert !
71	225	167	3-7029-0167-1	2	1981	BASIC-Uebungen fuer Anfaenger	Julius Schaerf, Tobias Schlaepfer	R. Oldenbourg Verlag	Hochsprache	Ein gutes Buch zum Einarbeiten ins Programmieren mit BASIC. Allein die vielen Beispiele sind ein Blick wert.
72	120	55	0-0000-000-00	1	1981	Disk Interfacing Guide (TRS-80 M-1)	William Barden, Jr.	80-Northwest Publishing	Assembler	Eine Auslassung ueber das Diskinterface des TRS-80.
73	200	110	3-88963-024-3	1	1983	Progr. in Assembl. mit Z80, Bd 2	Dr. E. Schmitter	W. Hofacker Verlag	Assembler	Eine kleine Befehlsreferenz mit Beispielen
74	200	223	3-349-00231-5	1	1986	Der Personal-Computer 1715	Prof. Dr.-Ing. Rolf Zeth	Verlag Die Wirtschaft	Know-How	Eine Art Bedienerhandbuch zum Robotron 1715
75	300	272	3-327-00361-0	2	1988	Praktische Mikrocomputertechnik	Manfred Kramer	Militaerverlag der DDR	Assembler	Das Buch zeigt ausfuehrlich den Bau von Rechnerhardware fuer den Z80. Mit ausfuehrlicher Beschreibung und Listings.
76	140	112	3-327-00553-2	1	1988	Mikrorechner in der Amateurme×technik	A. Bogatz	Militaerverlag d. DDR	Hardware	Interface-Schaltungen in TTL (DDR-Symbole, deshalb gewoehnungsbeduerftig!)
77	300	224	3-341-00567-6	1	1988	CP/M in der Praxis	Joachim Bernert	Verlag Technik der DDR	Know-How	Beschreibung von CP/M und einigen Anwendungen mit Beschreibung.
78	150	65	0-0000-00-000	1	1988	radio fernsehen elektronik nr.11	./.	VEB Verlag Technik Berlin	Allgemein	Nr. 11, Elektronikzeitschrift
79	100	47	0-0000-00-000	1	1988	Funkamateur 11/88	./.	Zeitschrift der GST	Allgemein	
80	100	50	0-0000-00-000	1	1988	Funkamateur 6/88	./.	Zeitschrift der GST	Allgemein	
81	300	240	3-341-00703-2	1	1989	W. Grafik	Wilfried Grafik	VEB-Verlag Technik, Berlin	Anwendung	Es wird der Umgang mit DBase III beschrieben.
82	250	224	3-341-00567-6	2	1990	CP/M in der Praxis	Joachim Bernert	VEB Verlag Technik, Berlin	Anwendung	Beschreibt den Umgang mit CP/M und seiner Kommandos.
83	700	300	0-936200-07-3	1	1979	TRSDOS 2.3 Decoded & other Mysteries	James Lee Farvour	IJG Inc., USA	Assembler	Es wird das Betriebsystem TRSDOS 2.3 in seinen SYS-Modulen beschrieben. Englischkenntnisse erforderlich!
84	250	158	3-88707-004-6	1	1982	BASIC im Buero	W. Bruemmer, D. Hemmerling, H. Fritsch	Luther-Verlag	Hochsprache	BASIC-Programme mit ausfuehrlicher Programmbschreibung
85	340	180	0-936200-00-6	1	1979	TRS-80 Disk & other Mysteries	H. C. Pennington	IJG Inc. USA	Hardware	Englischkenntnisse erforderlich!
86	123	129	0-000000-00-0	1	1984	TRS-80 & Video-Genie ROM-Listing	Luidger Roeckrath	./.	Assembler	Das komplette ROM-Listing fuer den TRS-80
87	500	402	3-921682-62-2	2	1984	Programmieren in Maschinenspr. mit Z80	P. Immerz	Ing. W. Hofacker	Assembler	Der Titel sagt wohl Alles!?
88	350	128	3-341-00524-2	1	1988	Programmieren mit C	Matthias Clauss	VEB Verlag Technik, Berlin	Hochsprache	Das Buch beschreibt ausfuehrlich die Anwendung der Sprach C. Gut aufgemacht.
89	123	240	3-341-00707-5	2	1989	MS-DOS	J. Huebener	VEB Verlag Technik, Berlin	Know-How	Wer mit PC's arbeitet und Details zu MS-DOS sucht liegt hier richtig.

HARDWARE * SOFTWARE
COMPUTER-CLUB 80
TANDY-/ GENIE- ANWENDER
ERFAHRUNGSAUSTAUSCH

SONDERINFO

FLOPPY-CONTROLLER

1771
179X
279X

Die Programmierung der Floppy-Controller
1771, 179x, 279x

von Gerald Schröder

In den ersten TRS-80- und Genie-Systemen wurde der Floppy-Controller WD-1771 eingesetzt (WD=Western Digital, kann auch SAB=Siemens sein). Es folgte der 1791, der auch doppelte Schreibdichte verarbeiten konnte. Die letzten Systeme (Genie IIs und IIIs) setzten den 2791 ein, den vorläufigen Schlußpunkt dieser Serie. Auch in CP/M-Rechnern dürften diese Controller zu finden sein. Übrigens sind der 1791 und der 2791 Mitglieder einer Familie (179x bzw. 279x genannt) von mehreren Controllern, die sich nur in Kleinigkeiten unterscheiden. Auf die Unterschiede werde ich noch eingehen, auch darauf, daß es zwischenzeitlich eine Serie 179x-02 gab. Allerdings habe ich keine Unterlagen über den Ur-Großvater 1761 und einen 1781, so daß ich mich über die beiden ausschweigen muß.

Der Otto-Normal-Benuter des Newdos (oder GDOS oder CP/M) macht sich nun wenig Gedanken über die Programmierung dieser Teile. Sobald jemand (wie ich) ein CP/M-BIOS selber schreiben muß, sollte er sich damit beschäftigen. Ich habe das dadurch umgangen, daß ich einfach aus dem GDOS und einem anderen CP/M-BIOS die Floppy-Routinen abgeschrieben habe. Aber ich stellte mir die Frage: Was kann das Ding eigentlich alles? Und somit habe ich Grossers "DOS-Buch" genommen und mir Teile vom Datenblatt des 179x übersetzt. Nachdem der Artikel schon fertig war, bekam ich von Helmut noch die Datenblätter zu den Controllern 1771, 179x-02, (SAB-)279x und (WD-)279x-02 (wobei die beiden letzten sich nach den mir vorliegenden Daten scheinbar nicht unterscheiden). Dadurch ist der Artikel noch etwas gewachsen.

Das Ergebnis ist eine Anleitung zum Programmieren bzw. Benutzen dieser Floppy-Controller. Der Aufbau und Anschluß (also der Hardware-Krempel) ist mir dabei egal. Ich werde versuchen, alle Möglichkeiten zu beschreiben, die sich bieten, wenn auch einige durch die Hardware bei einigen Rechnern verbaut wurden (vor allem der DMA-Zugriff). Aber das kann ja geändert werden.

Da der Artikel ziemlich lang geraten ist, will ich kurz den Aufbau erklären. In Kapitel 1 wird gezeigt, wie man an den Controller überhaupt rankommt (unabhängig vom verwendeten Computer) und was er so leistet. In diesem Kapitel dürfen sich auch die Hardware-Freaks mal angesprochen fühlen, einige der Möglichkeiten der Controller besser auszunutzen.

Kapitel 2 liefert eine Anleitung zum Programmieren des Controllers und sagt, wie er darauf reagiert. Allerdings werde ich mich dabei auf den Endpunkt der Fahnenstange, den 2791, beschränken. Die Unterschiede zu den anderen werden angemerkt oder in Kapitel 4 beschrieben. Für den Profi sind eigentlich nur die Tabellen interessant, die sich jeder zum Programmieren auf's Laufwerk legen sollte (s. Anhang).

Die Formatierung einer Diskette ist in Kapitel 3 das Thema. Das mag trivial erscheinen (gebe ich einfach "FORMAT" ein), ist es aber nicht, sobald wir uns genau ansehen, was zu passieren hat und was neben den "normalen" Formaten noch möglich ist.

Kapitel 4 bringt dann die Unterschiede zwischen zwischen den einzelnen Controllern. Dort beschreibe ich auch, was sie nicht können. Evtl. solltet Ihr Euch den Teil für Euren Controller zuerst durchlesen, damit Ihr in Kapitel 2 schon wißt, ob Ihr das für Euren Controller überhaupt gebrauchen könnt.

Wie die Programmierung bei den TRS-80s und ihren Kompatiblen konkret aussieht, sage ich nicht. Weil das sehr gut im Grosser steht (Kapitel 1). Wer das in natura sehen will, sollte sich im Grosser im Kapitel 3 die Seiten 24 bis 31 bzw. im SYS0 4630h-47E2h ansehen.

(

1. Die Verbindung zum Prozessor

Gelegentlich muß ich hier näher auf die Hardware eingehen, aber das werde ich kurz halten. Für den Programmierer ist das nicht so wichtig. Interessant dagegen ist, daß
1. die Daten parallel übertragen werden (8 Bits),
2. der Controller den DMA-Betrieb unterstützt,
3. zur Ansteuerung des Controllers 4 Register zur Verfügung stehen,
4. nach Beendigung jedes Kommandos ein Interrupt ausgelöst wird,
5. der Controller einiges nicht macht bzw. kann.

Der erste Punkt dürfte kaum weiter überraschen, nur wird damit ein "Handshaking" nötig, d.h. der Controller muß über irgendeine Extra-Leitung sagen, wann er Daten holen oder abgeben möchte. Die einfache (und beim TRS-80 samt Nachfolgern genutzte) Möglichkeit ist, ständig den Status abzufragen (wie, dazu weiter unten mehr) und sobald das Bit 1 (DRQ=Data Request) gesetzt wird, die Daten abzuholen bzw. zu übergeben. Dieses Verfahren wird "Polling" genannt.

Mir gefällt die zweite Möglichkeit besser: Ein DMA-Controller (DMA=Direct Memory Access) übernimmt die Übertragung direkt aus dem bzw. in das RAM, während der Prozessor sich langweilt oder andere Sachen macht (Wurzel aus 2 auf 10 Stellen genau berechnen oder solchen Quatsch). Der Floppy-Controller verfügt dazu über eine Extra-Leitung, sinningerweise auch DRQ genannt, die dasselbe leistet wie das DRQ-Bit im Statusbyte, aber direkt an einen DMA-Controller angeschlossen werden kann. Damit hätten wir dann drei Prozessoren (bzw. einen Hauptprozessor und zwei Controller) im System, was den TRS-80-Leuten wohl zu aufwendig war und schätzungsweise nur bei entsprechend aufwendiger Programmierung drumherum was bringt (ansonsten "langweilt" sich der Proz eh' nur rum). Aber die Möglichkeit ist vorhanden, wenn a) sich ein DMA-Controller im System befindet oder b) der HD-64180 als Haupt-Prozessor benutzt wird (der hat nämlich schon zwei DMA-Controller "on chip", werden also kostenlos mitgeliefert).

So, damit hätten wir Punkt zwei auch abgehakt. Kommen wir zu Punkt drei, den Registern. Davon haben wir 4 zur Verfügung, was bedeutet, daß nur 4 I/O-Ports oder (Memory-Mapped-) Speicheradressen dafür verbraten werden. Zur Adressierung werden nur 2 Bit benötigt (Adressen 00, 01, 10, 11). Wo diese 4 Adressen nun wirklich liegen, bleibt dabei dem Hardware-Entwickler überlassen. Die Bedeutung der Register:
00 Status- und Kommando-Register
01 Track-Register
10 Sektor-Register
11 Daten-Register

Die letzten drei Register dürften leicht zu verstehen sein: Das Track-Register nimmt eine Track-Nummer von 0 bis 244 auf, das Sektor-Register eine Sektor-Nummer von 0 bis 244. Beim Schreiben wird dem Controller mitgeteilt, auf welchem Track das Laufwerk gerade steht bzw. welcher Sektor als nächstes bearbeitet werden soll. Beim Lesen werden die aktuellen Werte ausgegeben. Bis jetzt werden nur Track-Nummern bis 80 und Sektor-Nummern bis ca. 40 benutzt. Die Laufwerke können also noch etwas leistungsfähiger werden, als sie es heute schon sind. 244 (F4h) ist eine Grenze, die durch die Format-Prozedur bestimmt wird (s. Kapitel 3). Das Daten-Register dient zum Übertragen der Daten zwischen Prozessor (oder DMA-Controller) und Floppy-Controller (u.a. ist auch der nächste zu suchende Track ein Datum, s. das Seek-Kommando). Nur das erste Register (Kommando und Status) ist etwas "Besonderes": Wenn etwas in dieses Register geschrieben wird, nimmt der Floppy-Controller das Byte als Kommando entgegen (sowas wie "Schreibe

(

(

Sektor" usw.), beim Lesen gibt er aber seinen momentanen Status zurück, wobei die Bedeutung der einzelnen Bits stark davon abhängt, welches Kommando vorher gegeben wurde. Dieses Register hat also zentrale Bedeutung für den gesamten Betrieb und jedesmal, wenn im folgenden etwas von "Kommandos geben" oder "Status abfragen" steht, ist dieses Register im Spiel.

Kommen wir zu Punkt vier. Sobald der Floppy-Controller ein Kommando erhalten hat (durch Schreiben eines Bytes in Register 00), fängt er an zu werkeln; einzige Ausnahme: beim Kommando "Force Interrupt" hört er auf, das vorherige Kommando auszuführen. Nachdem das Kommando beendet ist (ob zur allgemeinen Zufriedenheit oder mit einem Fehler, ist dabei egal) wird ein Interrupt ausgelöst; dafür gibt es wie für DRQ auch eine eigene Leitung (hier: zum Prozessor oder einem Interrupt-Controller), genannt INTRQ (Interrupt Request). Diese Möglichkeit wird im TRS-80 auch nicht genutzt, kann aber eingestellt werden. Was für einen Sinn könnte das Ding haben? Nun, wie schon gesagt, kann der Proz weiterarbeiten, während der Floppy-Controller ein Kommando ausführt, z.B. einen Sektor an den DMA-Controller weitergibt. Aber der Proz sollte irgendwann darüber informiert werden, daß das Kommando ausgeführt wurde und ob Fehler aufgetreten sind. Dann wird ein Interrupt ausgelöst und der Proz kann sich ansehen, was inzwischen passiert ist, während er sich mit der Wurzel aus 2 herumgeschlagen hat.

Auf zum letzten Punkt: Was leistet der Floppy-Controller nicht? Das hängt nun ganz von dem Controller ab. Zum Beispiel kann der 1771 nur SD (=Single Density, einfache Schreibdichte) lesen oder schreiben und kann mit DD (=Double Density, doppelte Schreibdichte) nichts anfangen. Die Nachfolger können teilweise zwischen SD und DD umschalten, was aber über eine Extra-Logik (und Extra-Ports oder -Adressen) dem Controller mitgeteilt werden muß. Wie dies passiert? Das weiß nur der jeweilige Hardware-Entwickler.

Außerdem kennt der 1771 keine DS-Disketten (=Double Sided, doppelseitig), während die Nachfolger teilweise überprüfen können, ob ein Track auf Seite 1 steht und einige sogar die Seite 1 des Laufwerks direkt anwählen können. Ansonsten muß dies eine Extra-Logik übernehmen, die wie bei der Anwahl von SD/DD natürlich wieder Ports/Adressen verbrät.

Eine Komplikation gibt es noch: Auf dem Markt sind heutzutage 8-, 5.25- und 3.5-Zoll-Laufwerke. Früher wurde einfach zwischen 8 Zoll und 5.25 Zoll unterschieden und die 3.5-Zoll-Drives zu den 5.25-Zöllern gesteckt. Für den Controller gibt es nur einen Unterschied: Die 8-Zoll-Laufwerke haben eine höhere Übertragungsrate (KBit/Sekunde) als die kleineren Laufwerke. Deshalb muß bei einigen Controllern der Takt zwischen 2 Mhz (8") und 1 Mhz (5.25") umgeschaltet werden, einige verlangen außerdem noch eine Benachrichtigung darüber, was nun anliegt, während einige mit 2 Mhz Takt zufrieden sind und ihn intern durch zwei teilen, wenn sie mit 5.25" arbeiten sollen. Der Grund für die Unterschiede ist darin zu suchen, daß je nach Controller verschiedene ZusatzChips (Data Separator und Write Precompensation) nötig sind oder deren Funktionen direkt in den Controller übernommen wurden. Aber nun kommt der Hammer: Irgendwann fiel einem Hersteller ein, daß mit der höheren Übertragungsrate auch auf 5.25"- und 3.5"-Laufwerken mehr gespeichert werden kann. Und schon hatten wir 8-Zoll-Laufwerke in der Größe 5.25" und 3.5", die heute bis 1.2 MB bzw. 1.44 MB aufnehmen können. Wenn im folgenden von 5.25" die Rede ist, bezeichnet dies einfach die niedrigere Übertragungsrate (Kapazitäten bis 720 KB sind typisch dafür, sowohl bei 5.25"- als auch bei 3.5"-Laufwerken).

(

Welches Diskettenlaufwerk soll der Controller nun ansprechen, wenn mehrere vorhanden sind? Dem Floppy-Controller ist das vollkommen schnurzpiepe. Auf den ersten Blick mag das etwas verwundern, aber so können theoretisch (vom FDC her) beliebig viele Laufwerke angeschlossen werden oder mehrere Disketten-Controller, z.B. für jedes Laufwerk einer, in dem System rumhängen. Im TRS-80 waren drei Laufwerke maximal vorgesehen, in den Nachfolgern wurde diese Zahl teilweise auf vier erhöht. Ich finde diese Einschränkung etwas kurzsichtig. Auf jeden Fall geht dafür bestimmt ein Extra-Port oder eine Memory-Mapped-I/O-Adresse verloren.

2. Die Kommandos für den Controller und seine Reaktion

Ein Kommando sollte nur an den Controller gegeben werden, wenn er nicht beschäftigt ist (Not Busy, Status-Bit 0 = 0). Einzige Ausnahme ist das Kommando "Force Interrupt", das immer gegeben werden kann. Welche Kommandos kann der Floppy-Controller nun ausführen? Entsprechend den Datenblättern der Floppy-Controller lassen sich die Kommandos in vier Klassen einteilen:

- I. Kopf-Positionierung: Diese Kommandos setzen den Schreib-/Lesekopf des Laufwerks auf verschiedene Tracks.
- II. Sektor-Zugriffe: Hier werden einzelne Sektoren gelesen oder geschrieben.
- III. Adress- und Track-Zugriffe: Diese Kommandos dienen zum Lesen einer Adresse (ID-Feld) und dem Lesen oder Schreiben eines Tracks.
- IV. Force-Interrupt: Hier gibt es nur das gleichnamige Kommando, das meist zum Unterbrechen anderer Befehle dient.

Im folgenden werden die einzelnen Kommandos der verschiedenen Klassen kurz erklärt. Eine Zusammenfassung in Tabellenform befindet sich im Anhang.

Wenn Zeiten angegeben werden (z.B. wie lange der Kopf zum Positionieren braucht), beziehen sich diese auf 8"! Bei 5.25" (bzw. bei der niedrigen Übertragungsrate) müssen die Zeiten jeweils <u>verdoppelt</u> werden! Hier gleich die erste Rechen-Übung: Nach jedem Kommando (bzw. dem Schreiben in das Kommando-Register) muß eine bestimmte Zeit gewartet werden, bevor der Status zum ersten Mal ausgelesen werden darf (sonst gibt's nur Müll). Das Status-Bit 0 ist nach 12 (SD) bzw. 6 (DD) Mikrosekunden OK. Die restlichen 7 Bits können frühestens nach 28 (SD) bzw. 14 (DD) Mikrosekunden (sinnvoll) abgefragt werden. Wie gesagt, verdoppeln sich im 5.25"-Modus die Zeiten.

Außerdem werden die beim 2791 möglichen Kommandos beschrieben. Unterschiede zu anderen Controllern werden in Kapitel 4 abgehandelt.

2.1 Klasse I: Positionieren des Schreib-/Lese-Kopfes

Alle Kommandos dieser Klasse dienen dazu, den Schreib-/Lesekopf über einem bestimmten Track der Diskette zu positionieren. Von der Step-Rate hängt es ab, wie schnell der Kopf bewegt wird; das Laden des Kopfes an die Disketten-Oberfläche (zum Schreiben oder Lesen) kann von diesen Kommandos gesteuert werden, wird aber ansonsten beim Schreiben oder Lesen automatisch vorgenommen.

Die <u>Step-Rate r1/r0</u> hängt vom angeschlossenen Laufwerk-Typ ab. Die meisten Laufwerke sind so gut, daß sie eine Step-Rate von 3 Millisekunden vertragen (also r1/r0 = 00).

Die Flags Head Load und Verify sollten zusammen betrachtet werden. Das Flag h legt fest, ob der Kopf am Anfang des Kommandos auf die Oberfläche gesenkt wird; v gibt an, daß am Ende (nach einer Kunstpause von 15 ms) verglichen werden soll, ob der richtige Track erreicht wurde; gleichzeitig wird auch hier der Kopf geladen (sonst könnte die Track-Nr. nicht gelesen werden). Die Möglichkeiten:

h	v	Wirkung
0	0	der Kopf wird abgehoben, wenn er vorher geladen war
1	0	der Kopf wird am Anfang geladen
0	1	der Kopf wird am Ende geladen und der Vergleich durchgeführt
1	1	der Kopf wird am Anfang geladen und der Vergleich durchgeführt

Leider ist das nur die Theorie, denn in der Praxis scheren sich die Laufwerke kaum um das h-Flag, sondern laden den Kopf, wenn die Diskette eingelegt oder die Klappe geschlossen wurde, oder wenn der Motor anspringt (TRS-80 und Kumpanen) oder was-weiß-ich. Und das Verify geht in die Hose, wenn die logischen (in den Sektoren vermerkten) Track-Nummern nicht den physikalischen entsprechen (z.B. die Diskette beginnt mit dem logischen Track 1, der physikalisch trotzdem meist Track 0 heißt).

Der Kopf wird (theoretisch) abgehoben, wenn a) ein Kommando mit h=0 und v=0 kommt oder b) während 15 Disk-Umdrehungen nach dem letzten Kommando kein neues gegeben wird.

Zum Verify (v=1): Der Controller versucht 5 Disk-Umdrehungen lang, das ID-Feld eines Sektors zu lesen. Wenn er es schafft (d.h. eine korrekte CRC (Checksumme) in einem ID-Feld findet), wird die gefundene Track-Nr. mit der aus dem Track-Register verglichen. Stimmen sie überein, wird das Kommando ohne Fehler beendet. Stimmen sie nicht überein oder findet er 5 Umdrehungen lang kein ordentliches ID-Feld, wird ein "Seek Error" im Status-Register angezeigt und das Kommando beendet. Übrigens ist das Beenden eines Kommandos immer mit einem Interrupt verbunden, was ich ab jetzt nicht mehr wiederholen werde.

Bei den Step-Kommandos gibt es noch das Update-Flag. Wenn es gesetzt ist, wird das Track-Register beim Steppen in- oder dekrementiert. Ansonsten bleibt es unverändert.

2.1.a Restore (=Seek Track 0)

Dieses Kommando bewirkt, daß der Kopf über Track 0 positioniert und das Track-Register auf 0 gesetzt wird. Wenn das Laufwerk nach 255 Steps immer noch nicht Track 0 erreicht hat, bricht der Controller die Sache mit der Fehlermeldung "Seek Error" ab (physikalisch Track 0, ganz außen).

2.1.b Seek

Im Track-Register muß sich die aktuelle Track-Nr. und im *Daten-Register* die gewünschte Track-Nr. befinden. Der Controller steppt dann automatisch solange, bis die gewünschte Track-Nr. erreicht ist. Achtung: Die logische Track-Nr. muß nicht der physikalischen entsprechen! Dann gibt es eine "falsche" Fehlermeldung, wenn "Verify" gewählt wurde!

2.1.c Step

Dieses Kommando führt einen Step in die Richtung aus, in die beim letzten Mal gesteppt wurde.

(

2.1.d Step In

Der Kopf wird in in Richtung auf Track 80 zu bewegt, und zwar um einen Schritt.

2.1.e Step Out

Der Kopf wird in Richtung auf Track 0 zu bewegt, wieder um einen Schritt.

2.2 Klasse II: Schreiben und Lesen eines Sektors

Bei beiden Kommandos wird zuerst der Kopf an die Diskettenoberfläche geladen. Wenn das Delay-Flag e gesetzt ist, folgt eine Warteschleife von 15 ms, bevor auf die Bestätigung durch das Laufwerk gewartet wird (e sollte gesetzt sein, warum auch immer).

Dann sucht der Controller 5 Disk-Umdrehungen lang nach dem richtigen Sektor. Dazu liest er die ID-Felder, vergleicht die Track-Nr. mit dem Track-Register, die Sektor-Nr. mit dem Sektor-Register und prüft die CRC-Bytes. Wenn das Side-Compare-Flag c gesetzt ist, vergleicht der Controller auch die Seitennummer im ID-Feld mit dem Side-Flag s. Wurde nicht der richtige Sektor gefunden, endet das Kommando mit einem "Record Not Found Error".

Nun wird der Sektor gelesen oder geschrieben. Sollte das Multiple-Record-Flag m gesetzt sein, inkrementiert der Controller das Sektor-Register und fängt von vorne an (ohne das Laden des Kopfes). Beendet wird das Kommando erst dann, wenn a) das Kommando "Force Interrupt" gegeben wird oder b) die Sektor-Nummer zu groß wird und der Controller mit einem "Record Not Found Error" abbricht.

2.2.a Read Sector

Nach der CRC-Prüfsumme im ID-Feld muß innerhalb der nächsten 30 Bytes (Single Density) bzw. 43 Bytes (Double Density) die Daten-Adreßmarke (Data Address Mark) folgen, sonst wird der Fehler "Record Not Found" ausgegeben (s. Kapitel 3: Formatierung).

Wenn ein Datenbyte abholbereit ist, setzt der Controller das DRQ-Bit (im Status-Byte) bzw. die DRQ-Leitung. Der Prozessor (oder der DMA-Controller) muß nun das Datum aus dem Daten-Register lesen. Tut er das nicht, bevor das nächste Byte abholbereit ist, setzt der Controller die Fehlermeldung "Lost Data".

Am Ende des Sektors steht noch eine Prüfsumme (CRC), die der Controller überprüft. Stellt sich ein Fehler heraus, setzt er die Fehlermeldung "CRC Error". Zuletzt gibt der Controller in Bit 5 des Status-Bytes bekannt, welche Adreßmarke er gefunden hat: 1 bedeutet "Deleted Data Mark", 0 "Data Mark".

2.2.b Write Sector

Wenn der richtige Sektor gefunden wurde, überspringt der Controller zuerst 11 Bytes (Single Density) bzw. 22 Bytes (Double Density). In dieser Zeit muß der erste DRQ vom Prozessor oder DMA-Controller beantwortet worden sein, sonst bricht der Controller mit der Fehlermeldung "Lost Data" ab. Ansonsten schreibt der Controller jetzt 6 bzw. 12 Null-Bytes und dann die Adreßmarke (Bit 0=1: Deleted, Bit 0=0: Data Mark). (s. Kapitel 3).

(

Nun werden die Daten-Bytes geschrieben. Wenn ein DRQ nicht rechtzeitg vom Prozessor (DMA-Controller) wahrgenommen wird, schreibt der Controller ein Null-Byte und setzt die Fehlermeldung "Lost Data", bricht aber nicht ab!

Am Ende des Sektors wird die vom Controller intern errechnete Prüfsumme (CRC, 2 Bytes) auf die Diskette geschrieben, gefolgt von einem FEh.

2.3 Klasse III: Adresse lesen, Track lesen oder schreiben

2.3.a Read Address

Der Controller sucht sich das nächste ID-Feld, liest es und gibt es an den Prozessor aus. Es handelt sich um sechs Bytes, die der Proz sich nach dem jeweiligen DRQ abholen muß. Die Bedeutung der Bytes:

Byte Nr.	Bedeutung
1	Track-Nr.
2	Seiten-Nr. (nur Bit 0)
3	Sektor-Nr.
4	Sektor-Länge (nur Bits 1/0: 00=128, 01=256, 10=512, 11=1024 Byte)
5	CRC 1 (Prüfsumme)
6	CRC 2 "

Die Prüfsumme wird auch vom Controller überprüft und er setzt die Fehlermeldung "CRC Error", wenn sie nicht ok ist. Außerdem wird die Track-Nr. auf jeden Fall in das Sektor-Register geschrieben. Die kann das Programm mit der "echten" im Track-Register vergleichen.

2.3.b Read Track

Ein kompletter Track (von Indexloch bis Indexloch) wird eingelesen, mit allem Drum und Dran (Gaps usw., s. Kapitel 3). Es wird keine Prüfung der CRCs vorgenommen. Beim Lesen können Fehler auftreten! Dieses Kommando eignet sich nicht zum Kopieren einer Diskette, sondern nur zum Analysieren.

2.3.c Write Track

Ein kompletter Track (von Indexloch bis Indexloch) wird geschrieben. Wenn das erste DRQ nicht rechtzeitig beantwortet wird, endet das Kommando sofort mit einem "Lost Data Error". Wenn später ein Byte nicht rechtzeitig angeliefert wird, schreibt der Controller ein Null-Byte und setzt den Fehler "Lost Data". Adreßmarken und CRCs werden vom Controller auf spezielle Kommandos im Datenstrom hin geschrieben. Näheres dazu im Kapitel 3 "Formatierung einer Diskette".

2.4 Klasse IV: Force Interrupt

Dieses Kommando unterbricht normalerweise andere Kommandos und erzeugt einen Interrupt, wobei der Zeitpunkt von den Bits 3-0 abhängt: Bit 3=Interrupt sofort, Bit 2=beim Index-Loch, Bit 1=Übergang von Ready auf Not-Ready, Bit 0=Übergang von Not-Ready auf Ready. Es können auch mehrere gesetzt werden. Ist keins gesetzt, wird kein Interrupt erzeugt. Wenn Bit 3 gesetzt war (Kommando D8h), muß danach das Kommando D0h (kein Interrupt) folgen, sonst gibt es beim nächsten Kommando Probleme!

(

Dieses D8h ist gut zu gebrauchen, wenn der Controller im Interrupt-Betrieb läuft und nun ein Kommando abgebrochen werden soll: Mit D8h wird die Interrupt-Routine sofort angesprungen und man erfährt dann, was nicht geklappt hat. Dann muß aber ein Kommando der Interrupt-Routine D0h sein.

Das Status-Byte wird nicht geändert, wenn vorher ein Kommando ausgeführt wurde (vorher Busy=1). Nur Busy wird gelöscht. Wenn der Controller vorher nichts zu tun hatte, wird das Status-Byte neu gesetzt, wobei die Bits die Bedeutung wie bei den Kommandos der Klasse I haben (s. Tabelle 3a).

Nach einem Force Interrupt muß 8 Mikrosekunden (DD) bzw. 16 Mikrosekunden (SD) gewartet werden, bevor das nächste Kommando gegeben werden darf, sonst gibt es Ärger! Nicht vergessen: Bei 5.25" verdoppeln sich die Werte!

3. Die Formatierung einer Diskette

Ich möchte vorausschicken, daß ich meine Erkenntnisse nur durch Lesen und Nachdenken erworben, aber nicht am "lebenden Objekt" überprüft habe. Ich kann also keine Garantie für die folgenden Informationen übernehmen. Leider sind die Angaben, wie die Formatierung zu geschehen hat, in den Datenblättern etwas dürftig und teilweise widersprüchlich. Ich habe deshalb hier etwas meine Phantasie spielen lassen.

Die Einteilung einer Diskette beginnt bei den Tracks, die jeweils einen "Kreis" auf der Diskettenoberfläche bilden. Normalerweise können heutige Laufwerke bis zu 40 oder 80 Tracks bearbeiten. Nun kann ein Track aber sowohl auf der Vorder- als auch auf der Rückseite einer Diskette stehen, wenn das Laufwerk zwei Köpfe hat. Deshalb ist noch eine Seiten-Kennung (0 oder 1) nötig.

Jeder Track umfaßt (meistens) mehrere Sektoren. Jeder Sektor bekommt eine Nummer und muß eine bestimmte Länge haben. Der 2791 unterstützt Sektor-Längen von 128, 256, 512 oder 1024 Bytes, d.h. soviele Daten-Bytes können in einem Sektor gespeichert werden. Der "echte" Sektor auf der Diskette enthält außer den Daten eine Menge Zusatzinformationen, so daß er sehr viel länger ist; in den Datenblättern wird er in zwei Felder geteilt: das ID-Feld (ID=Identify), das alle Infos über Track-Nr, Sektor-Nr., Seiten-Nr. und Anzahl der Datenbytes enthält, sowie das Datenfeld, das aus den Datenbytes besteht. Dazu kommen bei jedem Feld Prüfsummen und Adreßmarken sowie einige GAPs (Lücken).

Etwas sollte hier noch einmal klargestellt werden: Dem Controller ist es ziemlich egal, wie eine Diskette aussieht, solange sie einigen kleinen Konventionen genügt, damit er die enthaltenen Bytes richtig interpretiert. Es ist also keinesfalls nötig, daß eine Diskette homogen ist, wie es uns die Angabe "80/DS/DD mit Sektorlänge 1024" weismachen will. Es kann sein, daß nebeneinanderliegende Tracks verschiedene Schreibdichten aufweisen, mal ein Track ausgelassen wurde, auf den Tracks Sektoren mit unterschiedlicher Länge stehen usw. So ist es denkbar, daß der Track Nr. 12 (physikalisch) die Nummer 120 trägt und nur einen Sektor enthält, der 128 Byte lang ist, die Nummer 150 und eine Kennung für die Rückseite hat, obwohl er auf der Vorderseite steht.

Zur Formatierung einer Diskette wird das Kommando "Write Track" benutzt. Der Controller erwartet dann, daß ihm ein kompletter Track, Byte für Byte, übergeben wird, so daß er ihn auf die Diskette schreiben kann. Allerdings schreibt er nicht jedes Byte so auf die Diskette, wie es übergeben wird, sondern einige spezielle Bytes erkennt er als Befehle und schreibt etwas anderes. Diese Bytes dürfen deshalb nicht als Daten oder Sektor-Nummer o.ä. benutzt werden, weil der Controller sonst durcheinander kommen würde. Die gefährlichen" Bytes tragen die Nummern F5h bis FEh.

Diese Daten sind dem Controller zu übergeben:
(modifizierte Backus-Naur-Form)

```
Track ::= GAP4(Density)         (Anfangslücke, abhängig von der Schreibdichte)
          FCh                   (Index-Markierung)
          GAP1(Density)         (Lücke nach Index-Markierung)
          {Sektor}              (die Sektoren)
          {Füllbyte(Density)}   (Auffüllen bis zum Trackende)

Sektor::= GAP3(Density)         (Lücke am Anfang)
          FEh                   (ID-Adreß-Marke)
          TrackNr               (die Track-Nummer)
          SeitenNr              (die Seiten-Nummer)
          SektorNr              (die Sektor-Nummer)
          SektorLänge           (die Kennung der Sektor-Länge)
          F7h                   (CRC erzeugen)
          GAP2(Density)         (Lücke zwischen ID- und Daten-Feld)
          Adreßmarke            (Adreßmarke)
          {Data}128*2**SektorLänge
                                (soviele Daten-Bytes, wie bei SektorLänge
                                 angegeben)
          F7h                   (CRC erzeugen)
          [GAP1(Density)]       (Lücke am Ende des Sektors, kann wegfallen)

Density       ::= SD | DD       (einfache/doppelte Schreibdichte)
Füllbyte(SD)  ::= FFh           (Füllbyte bei Single Density)
Füllbyte(DD)  ::= 4Eh           (Füllbyte bei Double Density)
TrackNr       ::= 00h | 01h | ... | F4h | FFh
SeitenNr      ::= 00h | 01h
SektorNr      ::= 00h | 01h | ... | F4h | FFh
SektorLänge   ::= 00h | 01h | 02h | 03h
             (für 128,  256,  512, 1024  Bytes/Sektor)
Adreßmarke    ::= DeletedDM | DataMark
DeletedDM     ::= F8h
DataMark      ::= FBh
Data          ::= 00h | 01h | ... | F4h | FFh  (meist wird E5h benutzt)

GAP4(SD)      ::= {Füllbyte}16                (Lücke bei Single Density)
GAP4(DD)      ::= {Füllbyte}16 [{F6h}3]       (Lücke bei Double Density)
GAP3(SD)      ::= {Füllbyte}10 {00h}4         (usw.)
GAP3(DD)      ::= {Füllbyte}24 {00h}8  {F5h}3 (oder 16 Füllbytes?)
GAP2(SD)      ::= {Füllbyte}11 {00h}6         (genau 6 Nullen!)
GAP2(DD)      ::= {Füllbyte}22 {00h}12 {F5h}3 (genau 12 Nullen und 3xF5h!)
GAP1(SD)      ::= {Füllbyte}16
GAP1(DD)      ::= {Füllbyte}32                (oder 16?)
```

Legende:

```
  x ::= y      x wird ersetzt durch y
  x(z)         x ist abhängig von z
  xxh          Byte xx (in Hexadezimal) einsetzen
  (qqq)        qqq ist ein Kommentar
  x | y        x oder y
  [x]          setze x keinmal oder einmal ein
  {x}          setze x keinmal, einmal oder beliebig oft ein
  {x}Wert      setze x hier "Wert"-mal ein, z.B.
               {00h}5 = setze 00h fünfmal ein
```

Interessant sind vor allem die "GAP"s. Das sind Lücken, die an strategisch wichtigen Stellen eingebaut werden müssen. Warum? Darüber schweigen sich die Datenblätter aus. Es scheint so, als ob dort die Synchronisation stattfindet. Der dafür "verschwendete" Platz sollte möglichst klein gehalten werden, damit viele Sektoren auf den Track passen, bei gleichzeitig möglichst schnellem und fehlerfreien Zugriff. Die Datenblätter weisen die oben angegebenen Werte als Minimalwerte aus. Es kann also damit herumexperimentiert werden. Z.B. besteht GAP4 in einem Beispiel in einem Datenblatt (für DD) aus 80x4Eh, 12x00h und 3xF6h und die Index-Markierung fehlt oft ganz. Auf jeden Fall sollten immer die 3xF5h (GAP3 und GAP2) geschrieben werden, denn die scheinen immens wichtig zu sein (warum auch immer); ebenso die 6/12 Nullen in GAP2 (s. auch Read/Write Sector!).

Im Grosser steht auf den Seiten 1-17 und 1-18, welches Format das Newdos 80 benutzt. Grosser sagt dort, daß die Größe der GAPs von der der Anzahl und Größe der Sektoren abhängt. In einem alten Format-Programm von TCS habe ich jedenfalls wilde Zahlen gefunden, die teilweise auch unter den als Minimum angegebenen Werten lagen und gerade für GAP2, wo sie exakt gelten sollten (nach dem Datenblatt), mit steigender Sektorgröße in die Höhe schossen (1024 Bytes/Sektor in DD: 88x4Eh, 48x00h, 3xF5h).

An Sektoren können soviele erzeugt werden, wie eben auf den Track passen. Wie kann festgestellt werden, ob der Track zu lang war? Ganz einfach: Dann erzeugt der Controller beim "Write Track" keine DRQs mehr, obwohl noch Daten zu übertragen sind. Zu kurz darf die Sache dagegen immer sein. Der Rest des Tracks sollte dann mit den Füllbytes aufgefüllt werden (solange der Controller noch mit DRQs nach Futter verlangt), damit es keine Probleme beim späteren Lesen gibt (da steht sonst ja nur Müll).

Jeder Sektor enthält die Angaben, auf welchem Track und welcher Seite er steht. Es besteht also die Möglichkeit, hier Mist zu bauen, indem zum Beispiel jeder Sektor eine andere Tracknummer bekommt usw. Das stört den Controller wenig, nur beim Arbeiten muß man dann später ziemlich aufpassen. Die Sektorlänge ist auf die vier Größen (binär) 00, 01, 10 und 11 für die Sektorlängen 128, 256, 512 und 1024 beschränkt. Es müssen auf jeden Fall (bei "Data") soviele Datenbytes geschrieben werden, wie der Sektor gemäß "SektorLänge" lang sein soll (Vorsicht! Keine Bytes von F5h bis FEh; die dürfen erst bei "Write Sector" auftauchen!). Deshalb diese seltsame Rechenformel ("**" soll "hoch" heißen).

Da die Bytes F5h bis FEh besondere Bedeutung haben, sind sie als Track-, Sektor- und Seitennummer sowie als Sektorlängenkennung und Datum (im Datenfeld) tabu! Hier eine Tabelle, die zusammenfaßt, was der Controller mit welchen Bytes macht:

Byte	Bedeutung bei Single Density	Double Density	
00-F4, FF	00-F4 bzw. FF schreiben		
F5	nicht erlaubt	A1* schreiben, CRC-Init	
F6	nicht erlaubt	C2* schreiben	
F7	2 CRC-Bytes erzeugen		
F8-FB	schreiben, CRC-Init	schreiben	(Data Address Mark)
FC	schreiben (")	"	(Index Address Mark)
FD	" (")	"	(frei)
FE	" , CRC-Init	schreiben	(ID Address Mark)

Die Bytes 00h-F4h und FFh werden einfach auf die Diskette geschrieben, in DD auch noch die Bytes F8h-FEh. Mit F8h-FBh sowie FEh (SD) bzw. nur mit F5h (DD) wird vom Controller der Prüfsummenzähler gesetzt, der dann beim Empfang von F7h zwei CRC-Bytes (Prüfsumme, Cyclic Redundancy Check) auf die Diskette schreibt. Die Bytes A1* und C2* (nur in DD) sind nicht die normalen Bytes dieser Art. Scheinbar kann der Controller an ihnen feststellen, wo er sich gerade befindet, weil hier das Taktsignal verändert wurde (Synchronisation).

4. Die anderen Controller

Erstmal eine Übersicht über alle mir (von den Datenblättern her) bekannten Controller und ihre Eigenschaften:

Familie -->		179x-01				179x-02						279x(-02)			
Eigenschaft	1771	91	92	93	94	91	92	93	94	95	97	91	93	95	97
Double Density		x		x		x		x		x	x	x	x	x	x
inv. Datenbus	x	x	x			x	x			x		x		x	
PLL-Daten-Sep.												x	x	x	x
Write Precomp.		x	x	x	x	x	x	x	x	x	x	x	x	x	x
Side Compare		x	x	x	x	x	x	x	x	x	x	x	x	x	x
Side Select										x	x			x	x
2 Mhz -> 1 Mhz												x	x		

Fangen wir bei dem 1771 an. Vorher gab es scheinbar noch einen 1761, über den ich aber keine Unterlagen habe.

1771

Die Zeit, bis der Kopf sich auf die Diskette gesetzt hat, wird hier mit 10 ms (statt 15) angegeben. Die Step-Raten betragen 6/6/10/20 ms (alles bei 8"). Der Controller hat zwar einen internen Daten-Separator (der die vom Laufwerk gelieferten Daten entschlüsseln kann), aber es wird ein externer empfohlen.

Bei den Kommandos ergeben sich folgende Unterschiede:
 Read Sector 1 0 0 m b e 0 0
 Write Sector 1 0 1 m b e $a_1 a_0$
  b=1: Sektor-Länge 128/256/512/1024 Bytes (wie normal bei allen späteren)
  b=0: Sektor-Länge von 16 bis 4096 Bytes, 01=16, 02=32 usw., 00=4096, die Längen F7h bis FEh sind nicht erlaubt
  a_1 und a_0 geben Daten-Adreßmarke an:
   0 0 = FBh, normal
   0 1 = FAh, Benutzer-definiert
   1 0 = F9h, Benutzer-definiert
   1 1 = F8h, deleted
   (alle späteren Controller kennen nur FBh und F8h)
  e=1: 10 ms Delay
 Read Track 1 1 1 0 0 1 0 s*
 Write Track 1 1 1 1 0 1 0 0
  s*=0: Synchronisation zu den Adreßmarken durchführen
  (es fehlt Flag e)

Beim Lesen eines Sektors muß die Daten-Adreßmarke spätestens 28 Bytes nach dem Ende des ID-Feldes gefunden werden. Im Statusregister zeigen die Bits 5 und 6 die gefundene Adreßmarke an (Bit 5=a_0, Bit 6=a_1).

(

Beim Formatieren muß GAP2 17 Bytes lang sein, wovon mindestens die letzten 6 Bytes 00h sein müssen. 00h sind statt FFh als Füllbytes erlaubt. Vor jeder Adreßmarke muß mindestens ein Nullbyte stehen. Empfohlen wird, daß jedes GAP mindestens 17 Bytes lang ist und mindestens 6 Nullen am Ende enthält. Die Index-Adreßmarke kann (wie bei allen anderen Controllern auch) fehlen.

Über den direkten Nachfolger des 1771, den 1781, habe ich wieder keine Daten.

179x-01

Die Nachfolger-Serie 179x-01 besteht aus 4 Controllern (1791/2/3/4). Zwei davon können Double Density schreiben und lesen. Zwei sind mit einem "wahren" Datenbus ausgerüstet (normal ist scheinbar der invertierte, bei dem ein 0-Signal eine logische 1 anzeigt). Alle Controller sind hier bereits mit einer "Write Precompensation" ausgerüstet (was immer das auch sein mag). Sie benötigen aber alle einen externen Daten-Separator.

Die Kommandos und Reaktionen entsprechen denen des oben beschriebenen 2791. Auch hier kann die Index-Markierung beim Formatieren weggelassen werden, aber in SD darf nur FFh als Füllbyte benutzt werden (in DD sowieso nur 4Eh). Ab hier kennen die Controller übrigens doppelseitige Disketten und liefern (zumindest) einen automatischen Vergleich, ob der Sektor auf der richtigen Seite steht. Aber die Controller selbst können dem Laufwerk (noch) nicht mitteilen, welche Seite zu benutzen ist.

179x-02

Diese Neuauflage der ersten Serie führt zwei neue Controller ein (1795/7). Die beiden können dem Laufwerk mitteilen, welche Seite angewählt wurde. Außerdem prüfen sie automatisch, ob auch die im ID-Feld eingetragene Seitennummer der angewählten Seite entspricht. Dieser Vergleich kann nicht abgeschaltet werden! Außerdem können sie die Bedeutung des Sektorlängen-Bytes etwas verändern. Übrigens ist bei diesen beiden die Null als Füllbyte in SD wieder erlaubt.

Folgende Kommandos ändern sich somit:

```
 Read Sector    1 0 0 m L e U 0
 Write Sector   1 0 1 m L e U a0
 Read Address   1 1 0 0 0 e U 0
 Read Track     1 1 1 0 0 e U 0
 Write Track    1 1 1 1 0 e U 0
  L=0: Sektor-Längen=256/512/1024/128 (was immer das auch soll)
  L=1: Sektor-Längen=128/256/512/1024 (normal)
  U=0: Seite 0 angewählt (und mit Seiten-Bit abgeglichen)
  U=1: Seite 1 angewählt (und mit Seiten-Bit abgeglichen)
```

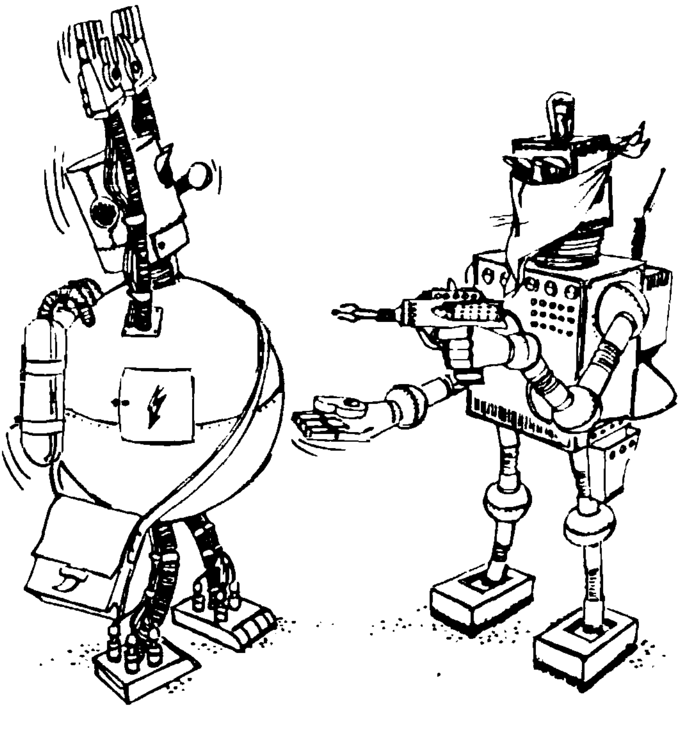
Kommen wir zum Schlußpunkt, der Serie

279x (-02)

Ich habe zwei Datenblätter vorliegen (eins von Western Digital und eins von Siemens) und mir scheint, daß die Siemens SAB-279x genau den WD-279x-02 entsprechen, nur daß Western Digital die Dinger noch in einem anderen Gehäuse anbietet (diese Quadrat-Form, "quad pack"). Auf jeden Fall haben die Dinger endlich einen (richtigen) Daten-Separator eingebaut, der "Phase Lock Loop Data Separator" genannt wird. Bei den anderen muß das Ding extra aufgebaut werden (bzw. es gibt sonst z.B. den WD-9216 dafür).

Bis auf eine Ausnahme scheinen die Dinger sonst der 179x-02-Familie zu entsprechen: Während der 2795 und der 2797 wie ihre Vorgänger eine Seiten-Auswahl ermöglichen, können der 2791 und der 2793 intern den angelegten 2 Mhz-Takt auf 1 Mhz herunterteilen. Vorher mußte immer eine externe Logik dies besorgen, wenn zwischen 8" und 5.25" umgeschaltet werden sollte. Hier ist dies dann nicht nötig (wohl aber noch bei den 2795/7). Außerdem müssen alle vier Controller darüber informiert werden, ob 8" oder 5.25" gewünscht wird, denn sonst arbeitet der Daten-Separator nicht richtig. Warum das nicht gleich alles zusammengepackt wurde, entzieht sich meiner Kenntnis. So sind jedenfalls beide Zweige (1/3 und 5/7) nichts Halbes und nichts Ganzes. Vielleicht gibt es ja inzwischen die Serie 379x. Darüber weiß ich aber nichts.

Literatur:
Hartmut Grosser: Das DOS-Buch, Aachen 1985, bes. Anhang A und Kapitel 1
Western Digital:
FD 1771-01 Floppy Disk Formatter/Controller
FD 179X-01 Floppy Disk Formatter/Controller Family
FD 179X-02 Floppy Disk Formatter/Controller Family
WD 279X-02 Floppy Disk Formatter/Controller Family
WD 9216-00/9216-01 Floppy Disk Data Separator - FDDS
Siemens:
SAB 279X Floppy Disk Formatter/Controller Family (1983)



»GELD HER ODER BATTERIE!«

(

Anhang: Tabellen

Tabelle 1: Kommando-Übersicht

Klasse	Kommando-Name	7	6	5	4	3	2	1	0	Bemerkung
		Bits und Flags								
I	Restore	0	0	0	0	h	v	r1	r0	= Seek Track 0
I	Seek	0	0	0	1	h	v	r1	r0	Suche Track
I	Step	0	0	1	u	h	v	r1	r0	Kopf bewegen
I	Step In	0	1	0	u	h	v	r1	r0	Kopf Richtung 80
I	Step Out	0	1	1	u	h	v	r1	r0	Kopf Richtung 00
II	Read Sector	1	0	0	m	s	e	c	0	Sektor lesen
II	Write Sector	1	0	1	m	s	e	c	a	Sektor schreiben
III	Read Address	1	1	0	0	0	e	0	0	ID lesen
III	Read Track	1	1	1	0	0	e	0	0	Track lesen
III	Write Track	1	1	1	1	0	e	0	0	Track schreiben
IV	Force Interrupt	1	1	0	1	I3	I2	I1	I0	Abbrechen

Tabelle 2: Flags

Flag	Bit	Name	wenn gesetzt (=1):
Klasse I:			
u	4	Update	Track-Reg. aktualisieren
h	3	Head Load	Kopf an Disk-Oberfläche laden
v	2	Verify	Track-Nr. vergleichen
r1,0	1,0	Step Rate	5.25", 3.5": 00=6, 01=12, 10=20, 11=30 ms 8": jeweils die Hälfte (3/6/10/15)
Klassen II&III:			
m	4	Multiple Record	mehrere Sektoren lesen/schreiben
s	3	Side Select	Seite 1 (nur mit c zusammen)
e	2	Delay	15 ms (8") bzw. 30 ms warten
c	1	Side Compare	Seite s überprüfen (sonst ist s egal)
a	0	Data Address Mark	Deleted Data Mark (F8h), sonst Data Mark (FBh)
Klasse IV:			
i3	3	Immediate	sofortiger Interrupt
i2	2	Index	Interrupt bei Index-Loch
i1	1	Not-Ready	Interrupt bei Übergang Ready -> Not Ready
i0	0	Ready	Interrupt bei Übergang Not Ready -> Ready
i0-i3=0000: kein Interrupt			

Tabelle 3a: Status-Byte bei Kommandos der Klasse I

Bit	Name	Bedeutung, wenn gesetzt (=1)
7	Not Ready	das Laufwerk ist nicht bereit
6	Protected	die Diskette ist schreibgeschützt
5	Head Loaded	der Kopf ist geladen
4	Seek Error	der Kopf steht nicht über dem richtigen Track
3	CRC Error	die CRC-Prüfsumme ist falsch
2	Track 00	der Kopf befindet sich über Track 0
1	Index	das Index-Loch wurde überquert
0	Busy	ein Kommando wird gerade abgearbeitet

Tabelle 3b: Status-Byte bei Kommandos der Klassen II und III

Bit	Name	Bedeutung, wenn gesetzt (=1)
7	Not Ready	das Laufwerk ist nicht bereit
6	Protected	Write: die Diskette ist schreibgeschützt sonst: 0
5	Record Type Write Fault	Read Sector: Adreßmarke ist "Deleted" Write : es ist ein Fehler aufgetreten sonst : 0
4	Record Not Found	Read/Write Sector: keinen Sektor mit dieser Track-, Sektor- und (evtl.) Seiten-Nr. gefunden Read Address : keinen Sektor gefunden sonst : 0
3	CRC Error	Read/Write Sector, Read Address: wenn Bit 4=1: Fehler im ID-Feld wenn Bit 4=0: Fehler im Daten-Feld sonst: 0
2	Lost Data	der Prozessor hat auf ein DRQ nicht rechtzeitig reagiert
1	Data Request	der Prozessor kann ein Datum schreiben/lesen
0	Busy	ein Kommando wird gerade abgearbeitet

(

(

Zugriff auf Disketten-Controller beim TRS-80 und Kompatiblen

Nachdem der Artikel über die Disketten-Controller 1771 bis 2797 fertigestellt war, mußte ich mir von Helmut und Hartmut vorhalten lassen, daß viele TRS-80-Besitzer solange nichts damit anfangen können, wie der Zugriff nicht klarer wird. Also habe ich mir den Grosser geschnappt und werde das Kapitel 1 hier noch einmal nacherzählen. Es geht um die Adressen und Ports, über die der Disketten-Controller und die Laufwerke gesteuert werden.

Der Zugriff erfolgt bei den meisten Geräten über eine Memory-Map, d.h. an bestimmten Speicherstellen im Speicher befindet sich kein RAM, sondern bei einer Schreib- oder Lese-Operation wird die Disketten-Einheit angesprochen. Diese Adressen reichen (für den Disketten-Zugriff) von 37E0h bis 37FFh. Für den Zugriff werden wie beim normalen Speicher der Assembler-Befehl "LD" oder die Basic-Befehle "PEEK" und "POKE" benutzt. Allerdings sind die Hochsprachen (Basic, Turbo-Pascal) für diesen direkten Zugriff einfach zu langsam. Hier muß innerhalb von Mikrosekunden reagiert werden, was nur mit Assembler geht.

Unter Newdos ist die Memory-Map immer vorhanden und zugreifbar, nur das Genie IIs und IIIs schalten sie manchmal aus und dann befindet sich dort normales RAM. Schwierig wird's unter CP/M, wo dort immer RAM sein muß. Dazu unten mehr.

Ohne Memory-Map arbeiten dagegen das Model III und das Model IV(p). Sie verwenden Ports für den Zugriff. Der Vorteil ist, daß sich dort, wo sonst die Memory-Map liegt, jetzt freies RAM befinden kann und Probleme unter CP/M umgangen werden. Auch dazu unten mehr.

Die vom Controller benutzten Register sind (unter Newdos) einfach zu erreichen: Status-/Kommando-Register = (37ECh), Track-Register = (37EDh), Sektor-Register = (37EEh), Daten-Register = (37EFh).

Zu beachten ist allerdings, daß beim Lesen nicht unbedingt der Wert herauskommt, der vorher geschrieben wurde, sondern einer, den der Controller bestimmt. Z.B. werden in das Kommando-Register Kommandos geschrieben, aber beim Lesen bekommt man nicht das zuletzt gesetzte Kommando, sondern den momentanen Status des Controllers. Außerdem enthält das Track-Register nur die Nummer des Tracks, über dem der Kopf des angewählten Laufwerks steht. Wenn das Laufwerk gewechselt wird, muß das Programm dem Controller mitteilen, über welchem Track sich der Kopf des neuen Laufwerks befindet, oder ihn mit dem Kommando "Restore" über Track 00 positionieren (s. Beispiel unten).

Es gilt jetzt noch einige Steuerungsaufgaben zu lösen, die nicht vom Controller vorgenommen werden. Wichtig ist vor allem die Auswahl des Laufwerks, mit dem der Controller arbeiten soll. Dafür ist die Adresse 37E0h zuständig. Allerdings kann auf diese Adresse zum Laufwerk-Anwählen nur geschrieben werden. Wenn dabei Bit 0 gesetzt ist, wird Laufwerk 0 angewählt (merkt man daran, daß der Motor anläuft und die Laufwerkslampe leuchtet). Bit 1 gilt für Laufwerk 1, Bit 2 für Laufwerk 2. Bei Bit 3 scheiden sich die Geister: Im TRS-80 M.I und den Genies I und II wird dieses Bit zur Auswahl der Disketten-Seite benutzt (wenn es sich um doppelseitige Laufwerke handelt und die Hardware dies unterstützt). Im Genie III und IIIs übernimmt Bit 4 diese Aufgabe, während mit Bit 3 Laufwerk 3 angewählt wird. Im Genie IIs ist beides möglich.

Wenn sich im System ein Controller befindet, der mit beiden Schreibdichten arbeiten kann (meist 1791), wird diese über 37ECh ausgewählt. Damit der Controller diese Auswahl nicht als Kommando auffaßt, muß sie sich von allen möglichen Kommandos unterscheiden. Es werden deshalb alle Bits außer Bit 0 gesetzt; die Auswahl SD oder DD erfolgt über Bit 0.

(

Die letzte Unterscheidung betrifft die Auswahl zwischen 8-Zoll- und 5.25-Zoll-Laufwerken. Sie wird über 37EEh vorgenommen, also parallel zum Sektor-Register. Dazu muß ein Wert >=80h in diese Speicherstelle geschrieben werden (was bedeutet, daß keine Sektoren mit Nummern größer als 127 mehr benutzt werden können). Mit LD A,80h und LD (37EEh),A werden 5.25" angewählt, mit C0h dagegen 8". Achtung! Die IBM-HD-Laufwerke mit 1.2 MB in der Größe 5.25 Zoll zählen als 8-Zoll-Laufwerke! Ebenso die HD-Laufwerke in der Größe 3.5 Zoll mit 1.44 MB.

Noch ein kleines Schmankerl, das aber m.W. im TRS-80 selten benutzt wird: Beim Auslesen der Adresse 37E0h (Drive-Auswahl) haben die Bits 6 und 7 eine besondere Bedeutung. Wenn Bit 7 gesetzt ist, kam der letzte Interrupt (Ansprung über 0038h) von der eingebauten "Uhr", einem Taktgeber, der alle 25 ms einen Interrupt erzeugt (wird teilweise bei der Tastatur-Entprellung und der System-Uhr benutzt). Wenn Bit 6 gesetzt ist, kam der letzte Interrupt vom Floppy-Controller, d.h. ein Interrupt-gesteuerter Betrieb ist möglich, wird aber im Newdos nicht benutzt. Dort werden beim Disketten-Zugriff die Interrupts ausgeschaltet, womit die System-"Uhr" dann aus dem Takt kommt und ein paar (Milli-) Sekunden verliert.

Die Reihenfolge beim (ersten) Zugriff auf ein Laufwerk müßte so aussehen:
- Controller rücksetzen (altes Kommando abbrechen, Force Interrupt)
  LD A,0D0h
  LD (37ECh),A
- Auswahl der Laufwerksgröße über 37EEh (80h=5.25", C0h=8")
- Auswahl der Schreibdichte über 37ECh (FEh=SD, FFh=DD)
- Anwahl des Laufwerks und der Seite über 37E0h
  (Bit 0-2/3: Laufwerk; Bit 3/4: Seite)
- Warten, bis der Motor läuft (Warteschleife)
- Restore-Kommando an Controller (FDC) über 37ECh (00h schreiben)
- Warten, bis FDC nicht mehr Busy (Lesen von 37ECh, bis Bit 0=0)
- Track-Nr. ins Datenregister (37EFh) schreiben
- Seek-Kommando an Controller (37ECh, 10h schreiben)
- Warten, bis FDC nicht mehr Busy
- Sektor-Nr. ins Sektor-Register (37EEh) schreiben
- Kommando an Controller (z.B. Read Sector=80h, Write Sector=A0h)
  - auf DRQs warten (Lesen 37ECh, Bit 1 gesetzt = DRQ)
  - jeweils ein Byte aus dem Daten-Register (37EFh) holen oder hineinschreiben
  - Ende bei Motor Off (Bit 7 von 37ECh gesetzt) oder
  - Ende bei "nicht mehr Busy" (Bit 0 von 37ECh gelöscht)
  - evtl. zwischendurch nochmal Select (37E0h schreiben)
    (damit der Motor weiterläuft)
- Auswertung der Fehler (37ECh lesen, je nach Kommando andere Bedeutung)

<u>Zu den Models III und IV(p)</u>

Hier gibt es keine Memory-Map, sondern der Zugriff erfolgt über Ports (Befehle IN/OUT in Assembler oder Basic). Die Controller-Register liegen bei F0h bis F3h (Kommando-Register usw.). Die Auswahl des Laufwerks und der Zugriffs-Parameter erfolgt über Port F4h. Die Bits 0 bis 3 wählen eins von 4 Laufwerken aus, Bit 4 dient zur Seiten-Anwahl, Bit 7 zur Schreibdichte-Auswahl (SD/DD). Neu sind Bit 5, mit dem eine Write-Precompensation ein- und ausgeschaltet werden kann, und Bit 6, mit dem die CPU scheinbar zum Warten auf den Controller veranlaßt wird. Genaues weiß ich aber leider dazu nicht. Allerdings wird beim Model IV zumindest unter CP/M (Montezuma) ein Interrupt-gesteuerter Diskettenzugriff durchgeführt.

Memory-Maps unter CP/M

Im TRS-80 ist der Speicher so organisiert, daß RAM erst ab 4000h existiert und somit die Programme nie mit der Memory-Map bei 37E0h-37FFh (für Disk, 3800h-38FFh für Tastatur, 3C00h bis 3FFFh für Bildschirm) in Konflikt kommen. Unter CP/M stehen aber die Programme ab 100h durchgängig im Speicher, auch im Bereich 3xxxh. Also wurde dort RAM eingesetzt und die Memory-Map entweder in einen anderen Speicher-Bereich verschoben (nach "oben", bei Fxxxh), oder nur eingeschaltet, wenn ein Programm auf diese Memory-Map zugreifen wollte.

Im ersten Fall gibt es keine Probleme: Die Memory-Map ist ja immer da, nur nicht am gewohnten Platz. Im zweiten Fall kann es passieren, daß ein Programm bei 3xxxh steht (oder dort wichtige Daten wie den Stack o.ä. hat) und nun die Memory-Map einschaltet. Dann "verschwindet" aber das dort vorher befindliche RAM, bis die Memory-Map wieder ausgeschaltet wird. Also darf das Programm sich nicht in diesem Bereich befinden, dort keinen Stack haben und nicht auf Daten zugreifen, die dort abgelegt wurden oder abgelegt werden sollen. Wenn zum Beispiel ein Sektor nach 3700h eingelesen werden soll, muß das Programm (meist das BIOS) ihn erst irgendwo anders ablegen, nach dem Lesen dann die Memory-Map ausschalten und ihn an seinen Bestimmungsort (hier 3700h) verschieben.

Solange das BIOS alle diese Zugriffe erledigt, gibt es keine Probleme, denn das BIOS steht oben im Speicher, meist über E000h, und hat alle Stacks und Daten dort gelagert. Sobald aber direkt von einem normalen Programm auf die Memory-Map zugegriffen werden soll, muß folgendes sichergestellt werden:

1. Das Programm muß komplett vor oder hinter der Memory-Map stehen.
2. Der Stack darf nicht im Memory-Map-Bereich liegen.
3. Es dürfen keine für den Zugriff wichtigen Daten (z.B. die Tracknummer) im RAM-Bereich "neben" der Memory-Map stehen.
4. Wenn die Memory-Map eingeschaltet ist, sollte das BIOS nicht dazwischenfunken, also: Keine Interrupts, keine BIOS-Aufrufe!
5. Nach dem Zugriff muß die Memory-Map wieder ausgeschaltet und evtl. Interrupts eingeschaltet werden.

Der Punkt "Keine BIOS-Aufrufe" ist nicht zu unterschätzen, denn meist schaltet das BIOS die Memory-Map wieder aus, wenn es seine Arbeit beendet hat. Während das Programm "denkt", sie wäre noch eingeschaltet. Dann kommt nur noch Daten-Müll zustande.

Dieser Bereich läßt sich auch ziemlich gut anhand des Beispielprogramms nachvollziehen, das irgendwo in diesem Info steht.

Gerald Schröder

Lit.: Hartmut Grosser: Das DOS-Buch, Röckrath, Aachen 1985
besonders Kapitel 1
Diverse Listings, u.a.:
Röckrath : ROM-Listing des TRS-80
TCS : BIOS für Speedmaster
Montezuma: BIOS für Model IVp
Schmidtke: BIOS für Genie I/II u.a.
myself : BIOS für Genie IIs

(

Das FDC-Demo-Programm

Wieder muß ich zugeben, daß Helmut und Hartmut als treibende Kräfte hinter diesem Artikel standen; jaja, ich werde von den beiden zu den übelsten Machenschaften getrieben. Eigentlich sollte das Ding nur kurz demonstrieren, wie der FDC (Floppy-Disk-Controller) programmiert werden kann. Im TRS-80-Disk-Guide gibt es dazu ein herrlich kurzes Programm und Manfred Held schickte mir ebenfalls ein sehr schönes kurzes Exemplar. Aber es überkam mich wieder mal. Also: Für die Experten sind nur die Seiten 1, 3 und die erste Hälfte von Seite 4 interessant. Der Rest sind Umschaltungen und Fehler-Ausgaben, die der Profi von Hand erledigt bzw. an irgendwelchen gesetzten oder ungesetzten Bits erkennt. Dem Assembler-Neuling empfehle ich, das komplette Ding abzutippen.

Allerdings sind dabei einige Fallen zu beachten, denn ich habe das Ding auf einem Genie IIs unter CP/M erstellt. Unter Newdos dürfte es trotzdem genauso laufen. Ich werde die Unterschiede unten zusammenfassen. Erstmal gehe ich auf die CP/M-Version ein.

Der Zugriff auf den Controller erfolgt hier über die alte TRS-80-Memory-Map, die beim Genie IIs eingeblendet werden kann. Wie das bei anderen TRS-80-Kompatiblen funktioniert, weiß ich nicht. Auf jeden Fall müssen die Routinen "MemMapOn" und "MemMapOff" geändert werden (im Zweifelsfall einfach ein "RET"). Damit verbunden ist auch das Ein- und Ausschalten der Interrupts. "DisableInt" und "EnableInt" können auch nur nur aus "DI" und "EI" (& "RET") bestehen. Im Zweifelsfall lieber ein "DI" zuiel, dafür aber kein "EI".

Das Programm läuft total unter Debugger-Kontrolle. Ihr könnt es auch so starten, aber dabei kommt nicht viel heraus (außer, daß es hoffentlich nicht abstürzt). Also folgender Aufruf: "DDTZ FDCTEST.COM" (so heißt das Programm bei mir). Dann mit "B103" einen Breakpoint bei 103h setzen und mit "G100" das Programm starten. Evtl. auftretende Fehler werden angezeigt, ansonsten sollte die Meldung "Es trat kein Fehler auf." erscheinen und die Ausgabe aller Register-Inhalte. Mit "D6000" (und mehrmals <ENTER>) kann nun der geladene Sektor betrachtet werden. Evtl. vor dem Programmstart mit "Z6000,7000,00" diesen Bereich löschen, damit sicher ist, daß auch wirklich etwas geladen wurde.

Wenn andere Parameter (Laufwerk, Track, Sektor, Buffer, Kommandos) gewünscht werden: Einfach die EQUates ändern, neu assemblieren und testen. Oder direkt per DDTZ und Kommando "A"ssemble eines der diversen "LD A,.." patchen. Alle Parameter werden nur am Anfang (erste Seite) benutzt und jeder taucht nur einmal auf. Ein Wort zum Assemblieren: Nachdem der Text mit Wordstar oder so bearbeitet wurde, Macro-80 mit "M80 =fdctest" aufrufen. Wenn er "NO FATAL ERRORS" meldet, kann gelinkt werden mit "L80 fdctest,fdctest/n/e". Wenn alles geklappt hat, steht der COM-File auf der Disk (schneller geht's mit 'ner Ramdisk).

Das Programm eignet sich nicht nur zum Lesen eines Sektors. Es ist vor allem zum Experimentieren gedacht. Also immer kräftig Breakpoints setzen und die Register belauern. Und alles mögliche variieren, z.B. andere Kommandos geben, Warte-Schleifen ändern, Status-Ausgaben genauer überprüfen usw. Außerdem kann das Ding erweitert werden, z.B. zu einem Formatier- oder Kopier-Programm, direkt auf Hardware-Ebene, so daß auch "kopiergeschützte" Disks problemlos angegangen werden können.

Zum Programm im einzelnen

Zuerst wird auf einen eigenen Stack umgeschaltet, denn es ist nicht sicher, ob der vorherige nicht in dem Bereich liegt, wo später (evtl.) die Memory-Map eingeblendet wird. Dann wird dem Controller befohlen, ein evtl. laufendes Kommando zu unterbrechen, und danach ein Laufwerk angewählt (DiskStart). Hier kommen die ersten Feinheiten: Wenn nicht lange genug gewartet wird, kann der Motor nicht starten und nichts klappt mehr. Nehmt

dazu einfach mal die Warteschleife raus. Übrigens läßt sich schlecht testen, ob auch wirklich eine Diskette im Laufwerk liegt. Eine Möglichkeit: Man frage das Index-Bit (Bit 1 im Status-Register, 37ECh) ab. Es müßte fünfmal in der Sekunde von 0 auf 1 und wieder zurück springen, wenn sich eine Diskette im Laufwerk dreht (300 U/min, falls ich ich mich nicht irre). Sonst bleibt es immer 1. Ich habe auf die Abfrage verzichtet. (Nachzulesen im Grosser, Seite 3-32, 47ECh-4808h: TSTDSK).

Nun führt der Controller ein Restore aus, d.h. er sucht Track 0. Das ist nötig, weil er nicht wissen kann, wo der Kopf des Laufwerks vorher gestanden hat und die im Track-Register stehende Track-Nummer höchstwahrscheinlich für ein anderes Laufwerk (bzw. dessen Kopf) gilt. Nach diesem Kommando stimmt das Track-Register mit der aktuellen Kopf-Position überein (beide sind 00). In dieser Version (Kommando 04h) liest der FDC auch noch von der Diskette einen Sektor-Header und sieht nach, ob dort wirklich die (Track-)-Nummer 0 eingetragen ist. Wenn das Kommando 00h benutzt wird, prüft er das nicht nach, sondern verläßt sich auf das vom Laufwerk gelieferte Signal "Track00", das aber auch kommt, wenn keine Diskette im Laufwerk liegt! Die kleine Warte-Schleife nach dem Ausgeben des Kommandos an den Controller ist nötig, weil der erstmal die Status-Bits aktualisieren muß. Wenn die Schleife fehlt, wird vom ersten "LD A,(fdccmd)" der <u>vor</u> dem Kommando gültige Status gelesen, was wahrscheinich Quatsch ergibt. Nach der Schleife wartet das Programm, bis das Busy-Bit gelöscht wird (Kommando abgearbeitet) oder das Laufwerk sich abmeldet. Die Abfrage des Busy-Bit allein reicht nicht, denn der Controller macht 5 Disk-Umdrehungen lang Leseversuche, aber wenn keine Diskette im Laufwerk liegt, kommen keine Index-Impulse! Dann wartet er ewig, daß die erste Umdrehung beendet ist... (S. auch c't-Artikel zum WD-2797). Eigentlich müßte hier (und bei den anderen Routinen auch) ein "Force Interrupt" am Ende dafür sorgen, daß das letzte Kommando wirklich abgebrochen wurde. Habe ich wieder drauf verzichtet.

Nun fährt das Laufwerk den richtigen Track an. Dazu erhält der FDC den gewünschten Track im <u>Daten</u>-Register (im Track-Register steht der <u>aktuelle</u> Track, noch 00, der nicht überschrieben werden darf!). Dann kommt das Seek-Kommando ("Suche Track"). Wieder überprüft der FDC am Ende, ob die Track-Nummer auch in einem Sektor-Header vermerkt ist. Wenn das nicht passieren soll, muß 10h statt 14h genommen werden (weitere Möglichkeiten: s. FDC-Artikel).

Wenn bei einem dieser Kommandos ein Fehler auftrat, wird dieser angezeigt und das Programm abgebrochen. Der Freak setzt hier Breakpoints und versucht, allein aus dem in A stehenden Fehler-Status herauszubekommen, was falsch gelaufen ist. Sollte jeder zum Test mal versuchen (z.B. nach den CALLs oder bei dem "AND 98h" oder evtl. schon nach dem ersten "LD A,(fdccmd)"). Allerdings kann dann nicht fortgesetzt werden, denn das Laufwerk schaltet sich meistens automatisch ab. Außerdem ist Vorsicht bei Breakpoints geboten, die beim "x-ten" Durchlauf erst wirksam werden (deaktivierte Breakpoints im DDTZ; ich habe keine Ahnung, wie man sie ganz entfernen kann). DDTZ fragt da mal "kurz" die Tastatur ab; das dauert zu lange und evtl. wird die Memory-Map wieder abgeschaltet! "Trace" würde ich lieber gleich vergessen (zu langsam). Schließlich geht es hier um "Echtzeit"-Operationen.

Kommen wir zum wichtigsten Teil: Dem Lesen des Sektors. Zuerst wird die Sektor-Nummer gesetzt und dann das Read-Kommando ausgegeben. Hier ist die Warteschleife danach besonders wichtig, denn sonst kommt der Status vom "Seek"-Kommando durch. Es dauert relativ lange, bis das erste DRQ erzeugt wird, das anzeigt, das ein Byte abgeholt werden muß. Aber nun geht es Schlag auf Schlag. Jedes DRQ muß sofort mit dem Lesen des Daten-Registers (37EFh) beantwortet werden, sonst kommt ein "Lost Data Error". Und hier heißt "sofort" wirklich <u>SOFORT</u>. Deshalb: Keine Interrupts, möglichst nur mit Registern und schnellen Kommandos arbeiten. Bei meinem 9.2-Mhz-Genie-IIs klappt die vorgestellte Routine, aber einige eingefügte NOPs bringen das Ganze zum Zusammenbruch. Ich hoffe, daß mit den angemerkten Zeilen auch die alten 1.77-Mhz-Maschinen mithalten können.

(

Hier beginnt nun das große Spielfeld: Statt einem einfachen "Read Sector" können mit einem gesetzten Multiple-Bit (Bit 4, +10h) alle auf einem Track befindlichen Sektoren (ab der gewählten Sektor-Nummer) eingelesen werden, ohne Änderung der Routine, ebenso ein ganzer Track (Kommando E0h) oder einfach ein Sektor-Header (Read Address, Kommando C0h). Zum Schreiben muß nicht viel verändert werden: Die beiden Zeilen bei "ReadData" werden einfach vertauscht, fertig! Allerdings sollte bei "ReadEnde" dann ein "AND FCh" folgen, denn hier gelten andere Fehler-Meldungen (die Fehler-Routine stimmt dann nicht mehr ganz). Dann kann natürlich auch ein ganzer Track geschrieben werden (Write Track, Kommando F0h) und schon ist das Format-Programm (fast) fertig!

Zurück zu dem vorliegenden Programm. Wenn kein Fehler aufgetreten auf, kommt die entsprechende Meldung und ab geht's in den Debugger, damit Ihr Euch den gelesenen Kram ansehen könnt. Falls irgendwo ein Fehler aufgetreten ist, wird dieser angezeigt und ebenfalls der Debugger angesprungen. Übrigens können die Fehler-Routinen auch angesprungen werden, wenn Ihr einfach mal wissen wollt, wie an einer Stelle der Status aussieht, z.B. bei "StepLoop" nach dem x-ten "LD A,(fdccmd)" ein Sprung zu "StepError" und schon steht's auf dem Bildschirm, Phosphor für Phosphor, grün auf schwarz bzw. sehr-hellgrün auf hellgrün (Gruß an Arnulf!).

Womit wir beim Thema wären: Wie steht's mit <u>Newdos</u>?

Das ORG sollte 5200h sein; dann dürfte es keine Probleme geben. Die Labels sollten ohne ":" eingegeben werden (jedenfalls unter ZEUS). Vielleicht sind einige Namen auch zu lang. Müßt Ihr mal probieren. Am Anfang sollte statt JP 0000 ein JP 440Dh stehen, dann könnt Ihr das Ding einfach als CMD-File starten und DEBUG meldet sich nach dem Lesen des Sektors automatisch. Oder Ihr setzt die letzte Zeile auf "END 440dh", womit DEBUG sofort nach dem Laden des Programms angesprungen wird.

Ansonsten ist der meiste Kram meiner Meinung nach unkritisch (habe ich aber nicht unter Newdos getestet!). Die Memory-Map muß jedenfalls nicht eingeschaltet werden (ersatzlos streichen). Die Routine "Print" besteht unter Newdos nur aus einer Zeile: "JP 4467h" (gibt den String (HL) bis zu einem 03 oder 0dh aus). Da Newdos aber nur 0ah haben möchte, um in die nächste Zeile zu kommen, müssen ab "KeinFehler" alle 0dh rausgeworfen werden (sonst kommen statt Fehlermeldungen nur Leerzeilen).

<u>Andere Rechner</u>

Für alle Rechner mit einem Controller der Familie 1771 bis 2797 müßte das Programm gültig sein, wenn auch evtl. andere Adressen benutzt werden. Wer (sinnvollerweise) seinen Controller über Ports anspricht, kann von mir eine Port-Version bekommen. Finde ich persönlich viel eleganter, aber der TRS-80 macht's nicht (von sich aus, läßt sich ändern, s. Model III und IV sowie meinen Port-Umbau).

Allerdings fehlt hier etwas ganz: Die Auswahl von 8-/5.25-Zoll-Laufwerken und einfacher bzw. doppelter Schreibdichte. Das dürfte überall anders gelöst sein und ist zum "Spielen" nicht so wichtig. Die Seiten-Auswahl kann einfach beim Drive-Select mit Bit 3 bzw. 4 (Genie III(s), einige IIs) erledigt werden, ist aber bei anderen Rechnern vielleicht anders gelöst, beim 2797 macht es z.B. der Controller selbst.

Gerald Schröder

```
;          ReadSector: Liest einen Sektor von der Diskette
;          28.2.89 von Gerald Schröder, verbessert: 3.3.89
;          für FDC-Controller 1771/179x/279x
;          auf Genie IIs (TRS-80-kompatibel) unter CP/M
;          erweiterungsfähig in alle Richtungen
;          Vielen Dank an Manfred Held und Hartmut Obermann.

           .Z80                           ;für Macro-80 unter CP/M
           CSEG                           ;für Macro-80 (absolutes Segment)
;(Newdos: ORG 5200h)

;          Memory-Mapped I/O-Adressen (TRS-80 u. Kompatible):
fdcsel     EQU      37e0h                 ;Laufwerk-Auswahl
fdccmd     EQU      37ech                 ;Kommando- und Status-Register
fdctrk     EQU      37edh                 ;Track-Register
fdcsec     EQU      37eeh                 ;Sektor-Register
fdcdat     EQU      37efh                 ;Daten-Register

;          Parameter
Drive      EQU      02                    ;Laufwerk B:
Track      EQU      03                    ;Track 3
Sector     EQU      04                    ;Sektor 4
RestCMD    EQU      04                    ;Restore-Kommando mit Verify
SeekCMD    EQU      14h                   ;Seek-Kommando mit Verify
ReadCMD    EQU      80h                   ;Read-Sector-Kommando ohne alles
ForceI     EQU      0D0h                  ;Force-Interrupt-Kommando
Buffer     EQU      6000h                 ;hier steht nachher der Sektor

Debugger:                       ;hier sollte ein Breakpoint hin!
           JP       0                     ;falls vergessen: Warm Boot
                                          ;(Newdos: JP 4400h ; Debug)
; ******************* Anfang Hauptprogramm *******************
Start:
           LD       SP,Stack  ;eigenen Stack benutzen
           CALL     MemMapOn  ;evtl. Memory-Map einschalten

           LD       A,ForceI  ;vorher laufendes Kommando
           LD       (fdccmd),A            ;abbrechen

           LD       A,Drive               ;Laufwerk und Seite wählen
           CALL     DiskStart ;Motor anlaufen lassen und Select-Maske
                                          ;merken
           LD       A,RestCMD ;Restore-Kommando
           CALL     Step                  ;ausführen
           JP       NZ,StepError

           LD       A,Track               ;Track-Nummer
           LD       (fdcdat),A            ;in das Daten-Register (nicht Track-R.!)
           LD       A,SeekCMD ;Seek-Kommando
           CALL     Step                  ;ausführen
           JP       NZ,StepError

           LD       A,Sector
           LD       (fdcsec),A            ;Sektor auswählen
           LD       A,ReadCMD ;Read Sector
           CALL     ReadSector            ;Sektor lesen
           JP       NZ,ReadSError         ;falls Lesefehler

           CALL     MemMapOff ;Memory Map ausschalten
           LD       HL,KeinFehler         ;Meldung kein Fehler
           CALL     Print                 ;anzeigen
           CALL     CrLf                  ;CR,LF ausgeben
           JP       Debugger  ;ab in den Debugger oder so
```

(

```
; **************  Ende Hauptprogramm - Anfang Unterprogramme  ******

;         MemMapOn: Memory-Map bei 3700h einblenden
;         nur nötig, wenn unter CP/M benutzt und keine Port-I/O
;         hier: für Genie IIs mit Port-Umbau
;         ---------------------------------------------------
sysport   EQU       0feh
MemMapOn:
          CALL      DisableInt          ;Interrupts lieber aus
          IN        A,(sysport)         ;alten Zustand des Ports
          LD        (PortAlt),A         ;merken
          AND       0eeh                ;Port I/O aus, Memory Map ein
          OUT       (sysport),A
          RET

;         MemMapOff: Memory-Map ausblenden
;         nur nötig, wenn unter CP/M benutzt und keine Port-I/O
;         hier: für Genie IIs mit Port-Umbau
;         ---------------------------------------------------
MemMapOff:
          LD        A,00                ;alter Zustand
PortAlt   EQU       $-1
          OUT       (sysport),A
          CALL      EnableInt ;hier dürfen sie wieder an
          RET

;         DisableInt: schaltet Interrupts aus
;         wenn mehrmals aufgerufen: merken, damit erst beim letzten
;         EnableInt diese wieder angeschaltet werden
DisableInt:
          LD        A,(AnzahlAufrufe)
          INC       A
          LD        (AnzahlAufrufe),A
          DI
          RET

;         EnableInt: schaltet Interrupts aus, wenn Aufruf 0 erreicht
;         Achtung! Bei TRS-80 (-Kompatiblen) unter CP/M mit DDTZ darf
;         hier nie ein EI erfolgen, denn der TRS-80 liefert alle 25 ms
;         einen INT0, der auf 0038h geht, was gleichzeitig Breakpoint
;         für DDTZ ist (RST 38h). Somit wird DDTZ aufgerufen, als ob
;         ein Breakpoint erreicht wäre.
EnableInt:
          LD        A,(AnzahlAufrufe)
          DEC       A
          LD        (AnzahlAufrufe),A
          RET       NZ                  ;noch nicht 0: zurück
;         EI                            ;Achtung! Nicht bei TRS-80-CP/M-DDTZ
          RET

AnzahlAufrufe:      DB        0

; ******************  FDC-Unterprogramme  ********************

;         DiskStart: Motor des Laufwerks anlaufen lassen und
;                    Seite auswählen, Select-Maske retten
;         IN: A=Laufwerk/Seite
;             Bit 0: Lw. 0, Bit 1: Lw. 1, Bit 2: Lw. 2
;             Bit 3: Genie IIs/IIIs: Lw. 3; TRS-80 u.a.: Seite 1
;             Bit 4: Genie IIs/IIIs: Seite 1
;         OUT: / (hier könnte überprüft werden, ob das Laufwerk überhaupt
;                existiert und ob eine Diskette eingelegt ist)
;         benutzte Register: A
;         ---------------------------------------------
```

(

```
DiskStart:
        LD      (SelMask),A         ;Select-Maske merken
        CALL    DriveSel  ;Motor anwerfen
;evtl. prüfen, ob Lw. vorhanden und Diskette eingelegt
        RET

;       DriveSel: Läßt Motor des Laufwerks weiterlaufen
;                 vorher muß DiskStart zum Setzen der Select-Maske
;                   aufgerufen werden
;       IN: /
;       OUT: /
;       benutzte Register: A
;       --------------------------------------------------------
DriveSel:
        LD      A,0                 ;Select-Maske
SelMask EQU     $-1
        LD      (fdcsel),A          ;Motor laufen lasse
        RET

;       Step: Kommando der Klasse I ausführen
;       IN: A=Kommando
;       OUT: Flag NZ = Fehler, Fehler-Maske in A
;       benutzte Register: A
;       ----------------------------------------
Step:
        LD      (fdccmd),A          ;Kommando ausgeben
        CALL    Wait                ;Warten
        CALL    DriveSel  ;Motor weiterlaufen lassen
StepLoop:
        LD      A,(fdccmd)          ;Status
        BIT     0,A                 ;noch Busy?
        JR      Z,StepEnde          ;nein, fertig
        BIT     7,A                 ;Motor Off?
        JR      Z,StepLoop          ;nö, warten
        LD      A,ForceI  ;ja, Kommando abbrechen
        LD      (fdccmd),A
        LD      A,(fdccmd)          ;und Fehler lesen
StepEnde:
        AND     9Ch                 ;Motor Off, Seek Error oder CRC-Error?
        RET                         ;wenn ja: NZ


;       ReadSector: liest einen Sektor von der Diskette in Buffer
;       IN: A=Kommando
;       OUT: Flag NZ=Fehler; Fehler-Maske in A
;
;       Vorsicht! Diese Routine ist unheimlich empfindlich gegenüber
;       Verzögerungen. Ein Befehl mehr oder weniger kann über den
;       Erfolg oder Mißerfolg entscheiden!
;       ------------------------------------------------------------
ReadSector:
        PUSH    AF
        CALL    DisableInt          ;Interrupt aus
        POP     AF
        LD      HL,fdccmd ;Adresse Kommando-/Status-Reg.
        LD      DE,fdcdat ;Adresse Daten-Register
        LD      BC,Buffer ;Anfang des Buffers
        LD      (HL),A              ;Kommando an den Controller
        CALL    Wait                ;warten, bis der das kapiert hat
        CALL    DriveSel  ;Motor laufen lassen
```

```
ReadLoop:
        BIT     1,(HL)          ;DRQ? (nur bei langsamen Rechnern)
        JR      NZ,ReadData     ;ja
        BIT     0,(HL)          ;noch Busy?
        JR      Z,ReadEnde      ;nein, Ende
        BIT     1,(HL)          ;DRQ? (langsame Rechner)
        JR      NZ,ReadData     ;ja
        BIT     7,(HL)          ;Motor off?
        JR      NZ,ReadBreak    ;ja, Fehler
        BIT     1,(HL)          ;DRQ?
        JR      Z,ReadLoop      ;nein, warten
ReadData:
        LD      A,(DE)          ;ja, Datum lesen
        LD      (BC),A          ;und ablegen
        INC     BC              ;Buffer-Adresse weiter
        JR      ReadLoop ;nächstes Datum
ReadBreak:
        LD      A,ForceI ;Kommando abbrechen
        LD      (HL),A
        LD      A,(HL)          ;Fehlerstatus lesen
ReadEnde:
        LD      A,(HL)
        AND     9ch             ;Fehler aufgetreten? (Not Ready, Record
                                ;Not Found, CRC-Error, Lost Data)
                                ;wenn ja: NZ
        PUSH    AF
        CALL    EnableInt ;evtl. Interrupts einschalten
        POP     AF
        RET


;       Wait: wartet einige Mikrosekunden oder so
;       IN: /
;       OUT: /
;       benutzte Register: A
;       ---------------------------------------
Time    EQU     20
Wait:   LD      A,Time
Wait1:  DEC     A
        JR      NZ,Wait1
        RET

;********** Ende FDC-Routinen - Anfang Fehler-Ausgabe ***********

;       Fehler-Routinen
;       Alle Routinen erwarten eine Fehler-Maske in A.
;       Zuerst wird immer auf den Normalbetrieb zurückgeschaltet.
;       Dann wird ein Zeiger auf eine Tabelle mit Zeigern auf Fehler-
;       meldungen nach HL geladen. Je nach FDC-Kommando sehen die
;       Tabelle und die Fehlermeldungen anders aus.
;       Die Routine 'LookForError' wertet die Fehler-Maske aus und
;       läßt über die Routine 'ShowFault' Fehler-Meldungen ausgeben.
;       Je nachdem, nach welchem FDC-Kommando der Fehler aufgetreten
;       ist, sind die verschiedenen Routinen anzuspringen.
;       Momentan wird danach der Debugger angesprungen. Alternativ ist
;       auch ein RETurn möglich

;       StepError: Bei Fehlern nach Kommandos der Klasse I oder IV
;                 (Step oder Force Interrupt)
;       mögliche Fehler sind: Not Ready, Seek-Error, CRC-Error
;       außerdem können die Meldungen Protected, Head Loaded, Track 0,
;       und Index auftreten (dies sind keine Fehler!)
;       IN: A=Fehlermaske
;       OUT: /
;       ------------------------------------------------------------
```

```
StepError:
        PUSH    AF              ;Fehlermaske retten
        CALL    MemMapOff       ;nur unter CP/M
        POP     AF              ;Fehlermaske zurück
        LD      HL,StepTable    ;Tabelle mit Meldungen
        JR      LookForError    ;Fehler anzeigen


;       ReadSError: Bei Fehlern nach dem Read-Sector-Kommando
;       mögliche Fehler sind: Not Ready, Record Not Found, CRC-Error,
;        Lost Data
;       außerdem kann die Meldung Deleted Data Mark auftreten (dies ist
;       kein Fehler!)
;       IN: A=Fehlermaske
;       OUT:
;       ------------------------------------------------------------
ReadSError:
        PUSH    AF              ;Fehlermaske retten
        CALL    MemMapOff       ;nur unter CP/M
        POP     AF              ;Fehlermaske zurück
        LD      HL,ReadSTable   ;Tabelle mit Meldungen
        JR      LookForError    ;Fehler anzeigen


;       LookForError: Auswertung einer Fehlermaske und Ausgabe von Fehler-
;                     meldungen
;       IN: A=Fehlermaske
;           HL=Zeiger auf Tabelle mit Zeigern auf Fehlermeldungen
;       OUT: /
;       ------------------------------------------------------------------
LookForError:
        LD      B,8             ;# Bits
ErrorLoop:
        RRCA                    ;Bit unten herausrotieren
        CALL    C,ShowFault     ;Bit gesetzt: Fehler anzeigen
        INC     HL              ;nächste Meldung
        INC     HL
        DJNZ    ErrorLoop       ;für alle 8 Bits

        CALL    CrLf            ;neue Zeile
        JP      Debugger        ;Abgang


;       ShowFault: Zeigt eine Fehlermeldung
;       IN: A=Fehlermaske, B=Fehler-Zähler
;           HL=Zeiger auf Zeiger auf Fehlermeldung
;       OUT: /
;       ----------------------------------------
ShowFault:
        PUSH    AF              ;retten: Status-Bits
        PUSH    BC              ;        Zähler
        PUSH    HL              ;        Tabellen-Zeiger
        LD      A,(HL)          ;LSB vom Zeiger auf Fehlermeldung
        INC     HL
        LD      H,(HL)          ;MSB
        LD      L,A             ;nach HL
        CALL    Print           ;Fehlermeldung (HL) anzeigen
        POP     HL              ;zurück: Tabellen-Zeiger
        POP     BC              ;        Zähler
        POP     AF              ;        Fehler-Maske
        RET
```

```
;       Print: Zeigt Fehler-Meldung an
;       hier: für CP/M über BDOS-Funktion Nr. 9 (PrintString)
;       IN: HL=Zeiger auf Meldung
;       OUT: /
;       ---------------------------------------------------------
Print:  EX      DE,HL           ;Zeiger nach DE
        LD      C,9             ;BDOS-Fkt. PrintString
        JP      0005            ;BDOS aufrufen

;       CrLf: Springt in die nächste Zeile
;       unter CP/M wird der String CR/LF ausgegeben
;       unter Newdos kann ein CR ausgegeben werden
;       ------------------------------------------------
CrLf:   LD      HL,CRLFString
        JR      Print

CrLfString:
        DB      0dh,0ah,'$'


;       Tabellen mit Zeigern auf Fehlermeldungen für verschiedene
;       Kommandos
;       Zu beachten: immer 8 Einträge
;       -------------------------------------------------------------

StepTable:
        DW      Busy            ;Bit 0 (dürfte nie auftreten)
        DW      Index           ;Bit 1
        DW      Track0          ;Bit 2
        DW      CRCError        ;Bit 3
        DW      SeekError       ;Bit 4
        DW      HeadLoaded      ;Bit 5
        DW      Protected       ;Bit 6
        DW      NotReady        ;Bit 7

ReadSTable:
        DW      Busy            ;dürfte nie auftreten
        DW      DRQ             ;ebenso
        DW      LostData
        DW      CRCError
        DW      RecNotFnd
        DW      Deleted         ;kein Fehler
        DW      Nothing         ;kann nicht auftreten
        DW      NotReady

;       hier können weitere Tabellen angefügt werden
;       für: Write Sector, Read Track, Write Track, Read Address


;       Fehler-Meldungen (bzw. überhaupt alle Meldungen)
;       -----------------------------------------------------
Ende    EQU     '$'     ;Markierung für CP/M, bei Newdos: 03
                        ;außerdem bei Newdos: alle 0dh entfernen!
KeinFehler:
        DB      0dh,0ah,'Es trat kein Fehler auf.',Ende
Busy:   DB      0dh,0ah,'Der FDC ist beschäftigt (kein Fehler).',Ende
Index:  DB      0dh,0ah,'Das Index-Loch wurde überquert (kein Fehler).',Ende
Track0: DB      0dh,0ah,'Der Kopf steht über Track 0 (kein Fehler).',Ende
CRCError:
        DB      0dh,0ah,'Eine Prüfsumme ist falsch!',Ende
SeekError:
        DB      0dh,0ah,'Der Track hat eine falsche Nummer.',Ende
```

»Verdammt noch mal, sag nicht immer PAPA zu mir!«

```
HeadLoaded:
        DB      0dh,0ah,'Der Kopf ist geladen (kein Fehler).',Ende
Protected:
        DB      0dh,0ah,'Die Diskette ist schreibgeschützt (hier '
        DB      'kein Fehler).',Ende
NotReady:
        DB      0dh,0ah,'Das Laufwerk ist nicht bereit!',Ende
DRQ:    DB      0dh,0ah,'Der FDC hat ein Data Request erzeugt (kein Fehler)
        DB      Ende
LostData:
        DB      0dh,0ah,'Beim Übertragen sind Daten verlorengegangen!',Ende
RecNotFnd:
        DB      0dh,0ah,'Der Sektor wurde nicht gefunden!',Ende
Deleted:
        DB      0dh,0ah,'Die Adreßmarke ist * Deleted * (kein Fehler).',Ende
Nothing:
        DB      0dh,0ah,'Dieser Fehler kann/darf nicht auftreten.'
        DB      0dh,0ah,'Mad Programmer Error.',Ende

        DS      30
Stack:
        END     Start           ;Newdos: END 440Dh ;Enter Debug
```

Ein Controller muß den Speicher steuern

Wer ein Floppy-Disk-Laufwerk — das Prinzip ist übrigens bei Festplatten-Laufwerken dasselbe — an seinen Computer anschließen will, benötigt dazu einen Controller, der auf den jeweiligen Rechnertyp zugeschnitten sein muß. Der Controller kann Bestandteil von Rechner oder Laufwerk sein, aber auch die Form einer Steckkarte haben, die ins Computergehäuse oder eine Erweiterungsbox paßt.

Die Information befindet sich bitseriell als magnetisches Abbild des verwendeten Codes in konzentrischen Spuren auf den Datenträgeroberflächen. Das Umsetzen elektrische Größe in magnetische Größe und umgekehrt erfolgt durch einen Schreib-/Lesekopf. Zum Auffinden des Speicherorts wird dieser vor einem Schreib- oder Lesevorgang zur gewünschten Spur positioniert. Durch die Rotation des Datenträgers bewegt sich während einer Umdrehung eine gesamte Spur am Kopf vorbei.

Da jedoch sowohl beim Lesen als auch beim Schreiben üblicherweise nur ein Teil einer Spur, dieser aber gezielt, transferiert werden soll, ist ein Verfahren notwendig, um Informationen innerhalb einer Spur genau adressieren zu können.

Adressen und Daten sind durch Polynome gesichert

Aus diesem Grund wird jede »jungfräuliche« Diskette oder Platte vor der erstmaligen Verwendung zur Datenspeicherung »formatiert«. Hierbei teilt man durch das vollständige Beschreiben aller Spuren den Speicherplatz auf dem Datenträger bei einem Synchronisationspunkt beginnend ein und macht ihn adressierbar.

Eine Spur wird in »Sektoren« unterteilt. Diese bilden die kleinste adressierbare Einheit und bestehen üblicherweise aus einem Datenfeld, für die zu speichernden Nutzdaten auch noch aus einem Adreßfeld. In diesem sind die Spuradresse, die Kopfadresse (bei mehreren Datenoberflächen pro Laufwerk), die Sektornummern sowie möglicherweise weitere Kennungen (zum Beispiel über die Länge des Datenfeldes) eingetragen. Die Adreßfelder der Sektoren werden nach dem Formatieren nicht mehr verändert. Sie dienen bei allen späteren Transfers zum Auffinden des Speicherplatzes.

Nach der Positionierung auf die gewünschte Spur liest und interpretiert der Controller alle den Schreib-/Lesekopf passierenden Adreßfelder. Sind Spurnummer und Kopfnummer identisch mit der gewünschten Vorgabe, so ist richtig positioniert. Stimmen dann auch noch die Sektornummern des Adreßfeldes mit dem gewünschten Wert überein, so ist richtig adressiert und der entsprechende Sektor befindet sich unter dem Schreib-/Lesekopf. Es wird kurz darauf das zugehörige Datenfeld entweder überschrieben oder gelesen.

Adreßfeld und Datenfeld eines Sektors grenzen nicht bündig aneinander, sondern es befinden sich Lücken zwischen ihnen. Regelmäßige Muster in diesen Lücken werden zur Synchronisation eines »Phase Locked Oszillator« (PLO) verwendet. Dieser erzeugt aus den gelesenen und in digitale Signale umgewandelten Lese-Rohdaten den »Lesetakt«, mit dessen Hilfe diese Rohdaten decodiert werden. Um aus dem ankommenden Bitstrom den richtigen Bytestrom zu erzeugen, werden sowohl Adreßfelder als auch Datenfelder von vereinbarten Mustern, den sogenannten Adreßmarken, eingeleitet, die unter anderem zur Bytesynchronisation dienen.

Sowohl Adreß- als auch Datenfeld sind durch ein Polynom gesichert, dessen Summe jeweils unmittelbar im Anschluß an die Nutzinformation in diesen Feldern aufgezeichnet wird. Beim Lesen wird die Summe nach derselben Vorschrift wie beim Schreiben gebildet. Sind die aufgezeichnete und die beim Lesen gebildete Summe identisch, so ist die Wahrscheinlichkeit sehr hoch, daß die Daten korrekt gelesen wurden. Für die Auswertung des Polynoms sowie für das Einleiten der weiteren Aktionen benötigt der Controller Zeit, die ihm durch die Lücken zur Verfügung gestellt werden.

Bei Schreibtransfer muß der Kopf von Lesen auf Schreiben, beziehungsweise umgekehrt, umgeschaltet werden. Die hierdurch entstehenden Stoßstellen mit eventuell aufgezeichneten Störimpulsen werden ebenfalls in die Lücken gelegt. Dies geht so vor sich, daß der Controller nach Auswertung des Adreßfeldes an einem definierten Ort der folgenden Lücke den Schreibstrom einschaltet, Synchronisierbytes und Adreßmarke des Datenfeldes selbst erzeugt und aufzeichnet, anschließend die Nettodaten transferiert und auf den Datenträger schreibt (wobei jedes Byte in die Bildung des Sicherungspolynoms einbezogen wird), die Summe des Polynoms hinzufügt und schließlich in der folgenden Lücke den Schreibstrom abschaltet.

Sieben Aufgaben müssen mindestens erfüllt werden

Aus den bisher skizzierten Vorgängen bei Informationstransfers von und zu rotierenden Speichern lassen sich folgende Aufgaben ableiten, die ein Controller unbedingt erfüllen muß:

— Erzeugen aller Formatdaten (Adreßfelder, Lücken, Datenfelder, Adreßmarken, Summe des Sicherungspolynoms) und deren bündige Aufzeichnung beim Formatieren des Datenträgers;

— Parallel-Seriell-Umsetzen der Nutzdaten beim Schreiben;

— Seriell-Parallel-Umsetzung der Nutzdaten beim Lesen mit vorheriger Bytesynchronisierung;

— Erzeugen der Summe von Sicherungspolynomen und Erkennen von Lesefehlern;

— Aufsuchen und Identifizieren des gewünschten Speicherorts durch Positionieren des Schreib-/Lesekopfes auf die richtige Spur und Identifikation des Sektors durch Auswerten des Adreßfeldes;

– zeitrichtiges Umschalten des Schreib-/Lesekopfes bei Schreibtransfer mit dem Hinzufügen einiger Formatdaten (Synchronisierbytes, Adreßmarke, Summe des Sicherungspolynoms) zu den Nutzdaten;
– Ausfiltern der Nettodaten aus den Formatdaten und ihre Weiterleitung beim Lesen.

Neben diesen Basisaufgaben verfügt ein Controller im allgemeinen über weitere Funktionen. Deren Art und Umfang hängen in einem starken Maße von der Schnittstelle und Art des Laufwerkes und des Gastrechners sowie vom gebotenen Komfort und Leistungsumfang des Controllers ab.

Die gängige Schnittstelle ist recht einfach

Die gängige Schnittstelle für Floppy-Disk-Laufwerke ist recht einfach gestaltet und legt die meisten Steuerungsaufgaben in die Hand des Controllers. Dieser wählt den Kopf an, falls mehrere vorhanden sind, steuert die Positionierung, indem er ein Richtungssignal und zeitlich richtig dosierte Impulse für einen Schrittmotor ausgibt und so weiter. Auch die Codierung und Decodierung übernimmt er. Bei »doppelter Aufzeichnungsdichte« (MFM) führt er gegebenenfalls eine bitmusterabhängige Vorkompensation der Schreibimpulse durch.

Bei Festplattenlaufwerken mit floppyähnlicher Schnittstelle sind die Aufgaben grundsätzlich ähnlich wie bei den Floppy-Disk-Laufwerken, nur werden die einzuhaltenden Zeitbedingungen wegen der merklich höheren Datenrate spürbar kritischer. Das gilt für die Auswertung der Adreßfelder und die zeitrichtige Umschaltung des Schreib-/Lesekopfes ebenso wie für den Datenkanal (PLO-Dimensionierung, Vorkompensation, Adreßmarkenerkennung mit Bytesynchronisierung, Transfergeschwindigkeit und so weiter).

Eine merkliche Entlastung des Controllers ist dann gegeben, wenn Laufwerke mit integriertem PLO verwendet werden. Laufwerke mit einer mikroprozessorähnlichen Schnittstelle (Disk-Bus-, ANSI-Schnittstelle) übernehmen meist sowohl die Codierung und Vorkompensation als auch die Lesetakterzeugung (PLO) und Decodierung selbst.

Der Datenkanal ist jedoch nach wie vor bitseriell, das heißt Bytesynchronisierung, Seriell-Parallel- und Parallel-Seriell-Umsetzung und so weiter ist nach wie vor Aufgabe des Controllers. Bei diesem Schnittstellentyp übernimmt das Laufwerk über einen Datenbus Kommandos und liefert über ihn Statusinformationen. Nach Erhalt der Zieldaten steuert es die Positionierung und Kopfauswahl selbst.

Bei Festplatten dürfen im Interesse einer niedrigen Ausschußquote bei der Produktion die Datenträger bereits bei der Auslieferung über einige feste Defektstellen verfügen. Sinnvollerweise schließt man diese Stellen beim Betrieb vom Zugriff aus. Dies kann per Systemsoftware (Eintrag in Buchführung) ohne oder mit Unterstützung des Controllers (zum Beispiel hardwaremäßiges Kennzeichnen des Defektes) erfolgen. Besser ist es, wenn der für die Systemsoftware sichtbare Externspeicherbereich durchgehend fehlerfrei erscheint. Dies läßt sich durch Zuweisung eines Ersatzbereiches für einen Defektbereich erreichen. Komfortable Controller stellen hierfür Hilfsmittel zur Verfügung (Formatieren von Ersatzsektoren, Ersatzspuren) und greifen im Idealfall ohne Belastung der Systemsoftware selbständig auf den Ersatzbereich statt auf den Defektbereich zu.

Für Fehler, die während des Betriebes dynamisch entstehen, ist es von großer Bedeutung, ob der Controller das Datenfeld mit einem fehlererkennenden (CRC-) oder fehlerkorrigierenden (ECC-)Code absichert. Während bei CRC nur der Nachweis möglich ist, daß ein Fehler aufgetaucht ist, besteht bei ECC die Möglichkeit, den Fehler zu identifizieren und zu korrigieren. Je nach verwendetem Code lassen sich 4 bis 11 zusammenhängende fehlerhafte Bits wieder richtig rekonstruieren.

Bei den heute gängigen Floppy-Formaten ist CRC üblich, während bei Festplatten der Mehraufwand für ECC in vielen Fällen aufgebracht wird.

Zur Entkoppelung der Zeitbedingungen zwischen Laufwerks- und Rechnerschnittstelle befindet sich in der Regel zwischen diesen ein Puffer, in dem die zu lesenden beziehungsweise zu schreibenden Daten zwischengespeichert werden. Bei dem verbreiteten einfachen Sektorpuffer ist die Pufferorganisation relativ einfach, und es werden an die rechnerseitige Schnittstelle keine besonderen Geschwindigkeitsanforderungen gestellt.

Allerdings läßt sich hiermit die vom Laufwerk her mögliche Datenrate auch bei einer schnellen Rechnerschnittstelle nicht voll ausnutzen (bestenfalls halbe Transferrate), da Transfers zwischen Laufwerk und Puffer nicht simultan zu Transfers zwischen Puffer und Rechner stattfinden können. Hingegen ist bei einem Wechselpuffer die Nutzung der vollen Datenrate möglich, wobei jedoch dessen Organisation nicht einfach ist und entsprechend hohe Anforderungen an die einzuhaltenden Zeitbedingungen der steuernden Einheit (Controller oder Adapter) gestellt werden.

Ein Großteil der Funktionen auf einem Chip

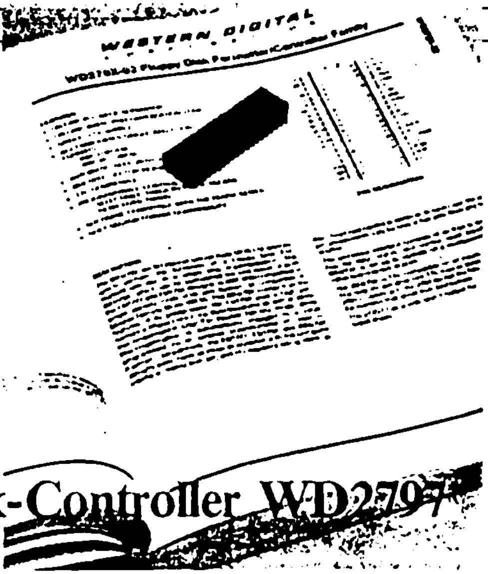
Ein FIFO als Pufferspeicher führt zu einem sehr guten Zeitverhalten, jedoch verlangt es von der Rechnerschnittstelle, daß die mittlere Datenrate gleich der Laufwerksdatenrate ist und daß zeitliche Störungen an dieser Schnittstelle ein gewisses Maß nicht überschreiten. Dieser Puffertyp macht, abgesehen von speziellen Lösungen, die Fehlerkorrektur über ECC nicht oder nur sehr umständlich möglich.

Das Zeitverhalten bei Transfers ist dann nicht unwesentlich, wenn Floppy-Disks als Back-up-Medium verwendet werden. Ein einfacher Sektorpuffer halbiert mindestens die mögliche Datenrate, wodurch die Zeit für den Back-up-Vorgang merklich ansteigt.

Ein Großteil der Funktionen eines Floppy-Disk-Controllers ist heute als LSI-Chip erhältlich. Allerdings müssen PLO, Vorkompensation, Puffer und ähnliches extern realisiert werden. Die angebotenen Controllerbaugruppen (meist mit dem LSI-Chip) unterscheiden sich bezüglich des Umfanges an Vorverarbeitung, der Leistungsfähigkeit und des Komforts.

Controller für Festplattenlaufwerke sind heute noch grundsätzlich als eine oder mehrere Baugruppe(n) ausgeführt. Bei einigen Typen werden zusätzlich auch Floppy-Disk-Laufwerke oder Magnetband-Kassettenlaufwerke gesteuert. Dies ergibt kompakte und wirtschaftliche Lösungen.

Armin Schwarz



Rolf Keller

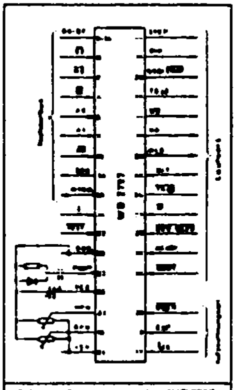


Bild 1. Pinbelegung des WD2797, geordnet nach Funktionsgruppen

Der Floppy-Disk-Controller-Chip WD2797 von Western Digital übertrumpft andere Bausteine ähnlicher Art durch seine integrierten Schaltungsteile für Datenseparation und Schreib-Vorkompensation. Damit werden einige Bauelemente bei der äußeren Beschaltung eingespart, und man kommt mit der 'unberechenbaren' Analogtechnik kaum noch in Berührung. Der 2797 wird zum Beispiel im c't 86 und im c't 68000 eingesetzt.

Floppy-Disk-Controller WD2797

Der 2797 ist ein Mitglied der Bausteinfamilie 279x, die ihrerseits eine (softwarekompatible) Weiterentwicklung der verbreiteten Familie 179x ist. Behandelt wird hier nur der Typ 2797, die anderen Typen unterscheiden sich davon geringfügig. Zur Klärung dieser Unterschiede sollte man das Datenblatt zu Rate ziehen, wenn es um die letzten Feinheiten geht.

Alle Typen der Familie 279x können 8"- und 5,25"-Laufwerke (und dazu kompatible Mikrofloppy-Laufwerke) in Single und Double Density ansteuern. Dabei können alle verbreiteten IBM-ähnlichen Datenformate benutzt werden.

Beschaltung

Bild 1 zeigt die Pinbelegung des 2797, geordnet nach Funktionsgruppen. Die auf dem Chip integrierten Analogschaltungen für Datenseparation und Schreib-Vorkompensation benötigen einige Bauteile für den Abgleich (Bild 1 links unten). Bei der Aktivierung des Eingangs $\overline{\text{TEST}}$ kann der Abgleich ohne Umstände in der betriebsbereiten Schaltung erfolgen.

Die Laufwerksignale können in der Regel direkt mit den Pins der ICs verbunden werden, die die gleiche Signalbezeichnung tragen. Besondere Beachtung verdienen die Signale HLT (Head Load Timing = Kopfaufsetzzeit) und READY. HLT teilt dem 2797 mit, daß die Beruhigungszeit nach dem Aufsetzen des Kopfes abgelaufen ist. Die meisten Laufwerke stellen dieses Signal nicht zur Verfügung; man muß es dann mit einem Zeitglied erzeugen. Das Signal READY soll die Lese-/Schreib-Bereitschaft des Laufwerks anzeigen, das heißt Diskette eingelegt, Tür geschlossen und Solldrehzahl erreicht. Meistens liefern 5,25"-Laufwerke dieses Signal nicht. Vielfach sieht man Schaltungen, die die Signale HLT und READY nicht unter allen Bedingungen (zum Beispiel beim Umschalten zwischen zwei Laufwerken) korrekt erzeugen. Dieser Fehler bleibt oft unerkannt, da der 2797 einen Lese-/Schreib-Vorgang erst nach fünf Plattenumdrehungen abbricht. Diese Zeit ist gewöhnlich länger als die Hochlaufzeit des Motors und die Beruhigungszeit beim Kopfaufsetzen, so daß am Ende meist alles gutgeht. Tatsächlich funktioniert es daher sogar, wenn man HLT und READY fest auf High-Pegel legt, aber empfehlenswert ist dieses Verfahren nicht.

Der Eingang $\overline{\text{DDEN}}$ des ICs dient zum Umschalten auf Double-Density-Betrieb, der Eingang ENP zum Einschalten der Schreib-Vorkompensation. Mit dem Eingang 5/8 kann man zwischen 5,25"- und 8"-Laufwerken wählen, muß aber gleichzeitig den Takt Φ umschalten (1 MHz bei 5,25", 2 MHz bei 8"). $\overline{\text{MR}}$ ist der Reset-Eingang des Bausteins.

Die fünf adressierbaren Register des Bausteins werden wie folgt angesprochen:

$\overline{\text{CS}}$	A1	A0	$\overline{\text{RE}}$	$\overline{\text{WE}}$
0	0	0	Status	Befehl
0	0	1	Spur	Spur
0	1	0	Sektor	Sektor
0	1	1	Daten	Daten

Der Ausgang DRQ (Data Request = Datenanforderung) kann bei reinem Polling-Betrieb unbenutzt bleiben, ansonsten wird er mit einem DMA-Baustein oder über eine Zusatzlogik mit dem $\overline{\text{WAIT}}$-Eingang der CPU verbunden. Voll DMA-fähig ist der 2797 allerdings nicht: Ihm fehlt dazu ein DMA-Acknowledge-Eingang. Alternativ kann man den Ausgang DRQ auch als Interrupt-Ausgang (Daten-Interrupts) betrachten.

Der Ausgang INTRQ (Interrupt Request = Interrupt-Anforderung) kann beim Polling ebenso unbenutzt bleiben. Sein Hauptzweck ist es, beim DMA-Betrieb das Befehlsende zu melden. Im Fehlerfall würde der DMA-Baustein sonst 'ewig' auf Daten warten.

Leider 'kennt' der 2797 nur ein einziges Laufwerk, das heißt, er hat nur für dieses eine 'Drive' Anschlußpins und interne Register. Um mehrere Laufwerke anschließen zu können, ist ein zusätzlicher Ausgabeport nötig, mit dem diese selektiert werden können. Mit den übrigen Portbits kann man dann gleich die Signale $\overline{\text{DDEN}}$, ENP, 5/8 und am besten auch $\overline{\text{MR}}$ per Software steuern.

Programmierung

Die möglichen Befehle (Tabelle 1) sind in vier Typen eingeteilt. Der Befehl 'Interrupt' nimmt eine Sonderstellung ein. Befehle der Typen 1 bis 3 erteilt man einfach durch Schreiben des entsprechenden Codes in das Befehlsregister. Dies darf aber nur dann geschehen, wenn der 2797 gerade nichts zu tun hat (BUSY-Bit im Statusregister = 0). Die Spur- und Sektorregister müssen vor der Ausgabe des Befehls geladen werden, sofern die Werte für die Ausführung des Befehls benötigt werden. Falls man mit mehreren Laufwerken arbeitet, muß man beim Wechsel auf ein anderes Laufwerk Spur- und Sektorregister entsprechend umladen, da der 2797 diese Werte nur für ein Laufwerk speichern kann.

Die Befehlsbearbeitung beginnt mit dem Setzen des Statusregisters. Der Baustein setzt das BUSY-Bit, löscht die Fehlerbits und aktualisiert die übrigen Bits gemäß Tabelle 2 beziehungsweise 3. Dabei ist wichtig, daß die Statusbits 0 und 7

Bit								Typ	Befehl	Aktion
7	6	5	4	3	2	1	0			
0	0	0	0	K	P	R1	R0	1	Nullstellung	Positionieren auf Spur 0
0	0	0	1	K	P	R1	R0	1	Suchen	Positionieren auf Spur x: Datenregister = x, Spurregister = aktuelle Spur
0	0	1	A	K	P	R1	R0	1	Spur weiter	Positionieren auf Nachbarspur: Richtung wie beim letzten Positionierbefehl
0	1	0	A	K	P	R1	R0	1	Spur vor	Positionieren auf nächsthöhere Spur
0	1	1	A	K	P	R1	R0	1	Spur zurück	Positionieren auf nächstniedrigere Spur
1	0	0	M	L	W	S	0	2	Sektor lesen	Sektor x der aktuellen Spur lesen; Sektorregister = x Kopf muß gemäß Spurregister positioniert sein
1	0	1	M	L	W	S	AM	2	Sektor schreiben	Sektor x der aktuellen Spur schreiben; Kopf muß gemäß Spurregister positioniert sein
1	1	0	0	0	W	S	0	3	Adreßfeld lesen	Nächstes vorbeilaufendes Sektoradreßfeld der aktuellen Spur lesen
1	1	1	0	0	W	S	0	3	Spur lesen	Gesamte aktuelle Spur transparent lesen
1	1	1	1	0	W	S	0	3	Spur formatieren	Gesamte aktuelle Spur formatieren
1	1	0	1	I3	I2	I1	I0	4	Interrupt	siehe Text

Bezeichnungen zu den Befehlen

K	(Kopf) 1: Zu Beginn des Befehls den Kopf aufsetzen 0: Zu Beginn des Befehls den Kopf abheben
P	(Prüfung) 1: Nach Erreichen der Zielspur wird automatisch durch Lesen der Adreßfelder auf der Platte geprüft, ob es sich wirklich um die gewünschte Spur handelt, deren Nummer im Spurregister steht. 0: Keine Prüfung
A	(Aktualisieren) 1: Das Spurregister wird bei Positioniervorgängen automatisch aktualisiert (weitergezählt). 0: Spurregister bleibt unverändert
M	(Mehrere Sektoren) 1: Beginnend mit dem vorgegebenen Sektor werden automatisch Sektoren mit den nachfolgenden Nummern so lange bearbeitet, bis das Spurende erreicht ist. 0: Nur der eine vorgegebene Sektor wird bearbeitet.
W	(Wartezeit) 1: Nach Aktivierung des Ausgangs HLD (Kopf aufsetzen) läuft eine Wartezeit (15 ms bei 8", 30 ms bei 5,25") ab, bevor endlich der Eingang HLT (Kopf ist aufgesetzt?) abgefragt wird. 0: Keine Wartezeit
S	(Seitenwahl) 1: Seite 1 anwählen 0: Seite 0 anwählen
AM	(Adreßmarken) 1: Adreßmarke $F8 schreiben (deleted data address mark) 0: Adreßmarke $FB schreiben (normal data address mark), dies ist bei den meisten Betriebssystemen der Normalfall
L	(Länge eines Sektors) L = 1 bei IBM Format Kodierung im Adreßfeld: 00 / 01 / 10 / 11 L = 0: 256 / 512 / 1024 / 128 L = 1: 128 / 256 / 512 / 1024
R	(Schrittrate für Spurwechsel) Die angegebenen Zeiten gelten für 5,25", bei 8" halbieren sich die Zeiten. R1 R0: 0 0 6 ms; 0 1 12 ms; 1 0 20 ms; 1 1 30 ms
I	(Interruptmaske) I0: Wechsel nicht READY — READY I1: Wechsel READY — nicht READY I2: Index-Impuls I3: sofortiger Interrupt

Tabelle 1. Übersicht über die Befehle des FDC 2797

zu jedem Zeitpunkt dieselbe Bedeutung haben; bei den Bits 1 bis 6 ist das nicht der Fall.

Nach der Ausgabe eines Befehls muß man mit dem Auslesen des Statuswortes etwas warten (max. 56 μs): Der Chip 'denkt' nicht so schnell. Ansonsten kann aber das Statusregister zu jedem Zeitpunkt gelesen werden.

Ist die Ausführung eines Befehls beendet, so wird das BUSY-Bit wieder gelöscht. Trat während der Ausführung des Befehls ein Fehler auf, ist das jeweilige Fehlerbit im Statusregister gesetzt.

Datentransfer

Weil der 2797 nur ein oder zwei Datenbytes intern speichern kann, muß er jedes Byte von beziehungsweise zur Diskette synchron zu deren Drehzahl abgenommen beziehungsweise geliefert bekommen. Dies geschieht durch Lesen/Schreiben des Datenregisters. Der 2797 fordert dazu pro Byte einmal auf, indem er das DRQ-Bit (Data Request = Datenanforderung) im Statusregister (Tabelle 3) setzt. Der Ausgangspin DRQ zeigt diese Anforderung ebenfalls an. Die DRQ-Anforderung wird beim Zugriff auf das Datenregister automatisch wieder gelöscht. Die maximal zulässige Reaktionszeit beträgt beim Schreiben 11,5 μs (8" DD), 23,5 μs (8" SD/5,25" DD) oder 47 μs (5,25" SD). Um hier mithalten zu können, ist ein schnelles Programm (Polling unter Interrupt-Sperre) oder Zusatzhardware (zum Beispiel DMA mit Hilfe des DRQ-Pins) erforderlich. Wird die Zeit 'verpaßt', so ist Datenverlust die Folge. Der 2797 setzt dann das Fehlerbit 2 im Statusregister (Tabelle 3).

Interrupts

Nach der Ausführung jedes Befehls (mit Ausnahme eines 'Interrupt'-Befehls mit I3 = 0) aktiviert der 2797 seinen INTRQ-Ausgang. Dabei ist es gleichgültig, ob der Befehl erfolgreich ausgeführt wurde oder nicht. Eine Interrupt-Sperre ist nicht möglich.

Zusätzlich kann man noch mit Hilfe eines 'Interrupt'-Befehls weitere Interrupt-Bedingungen vorgeben (Tabelle 1). Der Befehl '$D1' bewirkt zum Beispiel, daß dann ein Interrupt erzeugt wird, wenn der Pegel am READY-Eingang von high auf low wechselt. 'I3 = 1' löst sofort einen Interrupt aus. Damit kann man den Durchlauf der Interrupt-Routine erzwingen, wenn das programmtechnisch erforderlich ist.

Alle Interrupts werden quittiert durch Lesen des Statusregisters oder durch Schreiben in das Befehlsregister. Ein durch 'I3 = 1' ausgelöster Interrupt kann jedoch nur durch den 'Interrupt'-Befehl '$D0' quittiert werden.

Eine zweite Wirkung des 'Interrupt'-Befehls ist es, daß er die Ausführung jedes anderen Befehls unmittelbar abbricht. Dies ist möglich, weil der 'Interrupt'-Befehl der einzige Befehl ist, der ausgegeben werden darf, während ein anderer in Bearbeitung ist (BUSY = 1).

Um es ganz kompliziert zu machen, beeinflußt der 'Interrupt'-Befehl auch noch das Statusregister. War gerade ein anderer Befehl in Bearbeitung, so wird nur das BUSY-Bit auf null gesetzt, die übrigen Bits bleiben erhalten. War aber der 2797 gerade nicht 'beschäftigt', so wechselt das Statusregister in den Zustand, den es nach der Ausführung eines Befehls vom Typ 1 hat (Tabelle 2).

Verwendung der Befehle

Vor jedem Datenzugriff muß der Kopf des Laufwerks auf die gewünschte Spur gestellt werden. Dazu sind die Befehle des Typs 1 (Positionierbefehle) bestimmt. Für einen 'normalen' Disk-Treiber benötigt man davon nur die Befehle 'Nullstellung' und 'Suchen'.

Vor der Ausgabe des Befehls 'Suchen' muß das Spurregister die Nummer der Spur enthalten, auf der der Kopf gerade steht. Nach dem Einschalten und nach Fehlern ist das nicht unbedingt der Fall. Man gibt dann zunächst den Befehl 'Nullstellung'. Dabei werden so lange Schritte in Richtung Spur

Bit	
7	Laufwerk nicht READY Das Bit zeigt den invertierten Zustand des READY-Eingangs.
6	Diskette schreibgeschützt Das Bit zeigt den invertierten Zustand des WPRT-Eingangs.
5	Kopf aufgesetzt Das Bit zeigt an, daß die Signale HLD und HLT auf High-Pegel liegen.
4	Positionierfehler Das Bit zeigt an, daß nach der Ausführung eines Befehls mit gesetztem P-Bit der Kopf nicht auf der richtigen Spur steht.
3	Prüfsummenfehler Das Bit zeigt an, daß bei der Prüfung nach einem Befehl mit gesetztem P-Bit ein Sektor-Adreßfeld mit verfälschter Information gefunden wurde.
2	Spur 00 Das Bit zeigt den invertierten Zustand des TR00-Eingangs.
1	Index Impuls Das Bit zeigt den invertierten Zustand des IP-Eingangs.
0	Baustein BUSY Das Bit zeigt an, daß der Baustein gerade mit der Ausführung eines Befehls beschäftigt ist.

Tabelle 2. Statusregister nach Befehlen vom Typ I

Bit	
7	Laufwerk nicht READY Das Bit zeigt den invertierten Zustand des READY-Eingangs.
6	Diskette schreibgeschützt Nur bei den Befehlen 'Sektor schreiben' und 'Spur formatieren', sonst 0.
5	Art der Adreßmarke Nur beim Befehl 'Sektor lesen', sonst 0. Das Bit zeigt an, welche Daten-Adreßmarke gelesen wurde. 0: Adreßmarke SFB (normal data address mark); 1: Adreßmarke SF8 (deleted data address mark).
4	Sektor nicht gefunden Nur bei den Befehlen 'Sektor lesen', 'Sektor schreiben' und 'Adreßfeld lesen', sonst 0. Das Bit zeigt an, daß der Baustein kein Sektor-Adreßfeld mit passender Sektor-, Spur- und Seitennummer einschließlich korrekter Prüfsumme gefunden hat.
3	Prüfsummen-Fehler Nur bei den Befehlen 'Sektor lesen', 'Sektor schreiben' und 'Adreßfeld lesen', sonst 0. Das Bit zeigt an, daß der Baustein anhand der Prüfsumme einen Fehler auf der Platte erkannt hat. Ist Bit 4 ebenfalls gesetzt, so lag der Fehler in einem Sektor-Adreßfeld, andernfalls in einem Datenfeld.
2	Datenverlust Das Bit zeigt an, daß der Baustein nach einer Anforderung über das Statusbit 1 nicht rechtzeitig (schritthaltend mit der Umdrehung der Diskette) durch Lesen bzw. Schreiben des Datenregisters bedient wurde.
1	Datenanforderung Das Bit fordert zum Lesen des nächsten Datenbytes (nach Lesebefehlen) aus dem Datenregister bzw. zum Schreiben des nächsten Bytes (nach Schreibbefehlen) in das Datenregister auf.
0	Baustein BUSY Das Bit zeigt an, daß der Baustein gerade mit der Ausführung eines Befehls beschäftigt ist.

Tabelle 3. Statusregister nach Befehlen von Typ 2 und 3

00 ausgeführt, bis das Signal $\overline{TR00}$ (vom Endschalter des Laufwerks) das Erreichen von Spur 00 meldet. Damit ist die aktuelle Position des Kopfes bekannt; der 2797 selbst setzt sein Spurregister auf 00.

Vor der Ausgabe des Befehls 'Suchen' lädt man die Nummer der zu suchenden Spur in das Datenregister. Der Baustein vergleicht sie mit der Nummer der momentanen Spur im Spurregister und führt so viele Schritte in der richtigen Richtung aus, bis die gewünschte Spur erreicht ist. Damit steht im Spurregister dann auch die Nummer der neuen, nunmehr aktuellen Spur.

Jedes Laufwerk verträgt nur eine bestimmte Schrittrate und kommt 'aus dem Tritt', wenn man sie überschreitet. Deshalb müssen die Bits R1 und R0 bei jedem Positionierbefehl korrekt gesetzt werden.

Nach einem Positionierbefehl folgt sinnvollerweise ein Schreib- oder Lesebefehl, für den der Kopf aufgesetzt sein muß. Um die Beruhigungszeit beim Aufsetzen zu sparen, kann man den Kopf mit Hilfe des 'K-Bits' schon zu Beginn des Suchens aufsetzen. Das Setzen des 'P-Bits' ist meist nicht sinnvoll, da die Ausführung der Prüfung Zeit kostet. Bei den Befehlen des Typs 2 wird außerdem die Spurnummer automatisch kontrolliert.

Datenzugriff

Das Schreiben und Lesen eines Sektors wird durch die Befehle des Typs 2 ermöglicht. Zuvor muß der Kopf auf die gewünschte Spur gestellt worden sein. Vor der Befehlsausgabe lädt man das Sektorregister mit der gewünschten Sektornummer. Das Spurregister steht meist wegen des vorangegangenen Positionierbefehls schon richtig. Zu Beginn wählt der 2797 über den Ausgang SSO (Side Select Output = Seitenwahl) die durch das 'S-Bit' vorgegebene Plattenseite an und setzt den Kopf auf. Dann liest er die Spur (auch bei 'Sektor schreiben') so lange, bis ein Sektoradreßfeld erscheint, das die richtige Spur-, Sektor- und Seitennummer einschließlich korrekter Prüfsumme (CRC) enthält. Falls der Controller innerhalb von fünf Plattenumdrehungen kein passendes Sektoradreßfeld findet, bricht er ab und setzt das Fehlerbit 4 (zusätzlich auch Bit 3) im Statusregister. Dieser Fehler kann durch schlechte Positionierung verursacht worden sein. Ein 'defensives' Programm wird deshalb 'Nullstellung' und 'Suchen' befehlen und dann einen neuen Versuch starten.

Im Normalfall wird ein passendes Sektoradreßfeld gefunden, und der 2797 wartet auf den Beginn des folgenden zugehörigen Datenfeldes, um direkt mit dem Schreiben beziehungsweise Lesen zu beginnen (DRQ-Anforderung). Die Länge des Sektors (Anzahl der Datenbytes) ist beim Formatieren der Diskette festgelegt worden; der Baustein findet diese Angabe (Tabelle 1, L-Bit) im Sektoradreßfeld.

Adreßfeld lesen

Der Befehl 'Adreßfeld lesen' liest das nächste in der momentanen Spur am Kopf vorbeilaufende Sektoradreßfeld und liefert seinen Inhalt (6 Byte: Spur-, Seiten-, Sektornummer, Sektorlänge, Prüfsumme davon) als Daten ab. Im Normalfall kann man damit nicht allzuviel anfangen. Eine Anwendung wäre zum Beispiel ein schnelles 1:1-Kopierprogramm, das jeweils eine komplette Spur in einer einzigen Plattenumdrehung liest beziehungsweise schreibt. Der Befehl macht es dann möglich, unmittelbar an der augenblicklichen Stelle mitten in der Spur zu beginnen. Bei den folgenden Befehlen ('Sektor lesen/schreiben') fängt man dann nicht mit Sektor 1 an, sondern mit demjenigen, der gerade vorbeiläuft.

Spur lesen

Der Befehl 'Spur lesen' ermöglicht es, eine komplette Spur in einem Stück transparent (d.h. einschließlich aller Adreßmarken, Datenbytes, Prüfsummen, Synchronisationsbytes und Füllbytes) zu lesen. Die Auswertung des Gelesenen ist aber recht kompliziert, insbesondere dadurch, daß einige Bytes 'angeschnitten' sein können. Der Befehl wird im normalen Betrieb nicht benutzt. Er kann Anwendung finden bei Hardware-Tests, beim Überprüfen von Formatierprogrammen oder zur Untersuchung fremder Aufzeichnungsarten.

Spur formatieren

Der Befehl 'Spur formatieren' formatiert die gesamte Spur, auf der der Kopf gerade steht, indem er sie vollkommen neu mit leeren Sektoren beschreibt. Ein vorheriges Kontroll-Lesen, wie etwa bei 'Sektor schreiben', findet nicht statt. Da man normalerweise die gesamte Platte formatiert und damit löscht, ist das nicht gefährlich. Es ist sogar erwünscht, denn nur so kann man eine fabrikneue oder fehlerhaft beschriebene Diskette 'auf Vordermann bringen'. Man beginnt dann mit Spur 00 und geht schrittweise mit Hilfe des 'Spur vor'-Befehls weiter, so daß Positionierfehler nicht zu befürchten sind.

Im Gegensatz zu anderen Bausteinen (zum Beispiel µPD 765) ist das Formatieren beim 2797 ziemlich programmaufwendig. Man muß dem Chip als Daten jeweils den kompletten Spurinhalt übergeben. Um diese Daten zusammenstellen zu können, bedarf es einer genauen Kenntnis der Datenstruktur auf einer Diskette. Es handelt sich dabei um mehrere tausend Bytes (siehe auch unter 'Spur lesen'), nicht etwa nur um Anzahl und Länge der Sektoren. Anders gesagt: Der Baustein 'weiß' gar nicht, wie man eine Platte formatiert. Es wird ihm per Programm mitgeteilt. Der Vorteil dieser Methode ist es, daß der Programmierer freie Hand hat und auch 'ausgefallenere' Formate erzeugen kann.

Probleme ...

Beim Arbeiten mit dem 2797 fielen zwei Probleme auf. Wie schon erwähnt, versucht der Baustein im Fehlerfall fünf Plattenumdrehungen lang, den gewünschten Sektor aufzufinden. Leider mißt er aber diese fünf Umdrehungen nicht zeitlich ab, sondern zählt einfach fünf Indexpulse (Eingang $\overline{IP}$). Falls nun keine Diskette eingelegt ist, das Signal READY aber trotzdem 'high' ist (Tür kann geschlossen sein, Motor kann laufen), so wartet der 2797 'ewig'. Ein ihm anzulastender Fehler ist das strenggenommen nicht, es stört nur bei einigen 'Sparschaltungen'.

Ein echter Fehler zeigt sich dagegen bei den Befehlen 'Sektor lesen', 'Sektor schreiben' und 'Spur lesen'. Dabei darf man nach der Befehlsausgabe zunächst nur die Bits 0 (BUSY) und 1 (Datenanforderung) abfragen. Die übrigen Bits zeigen zu Beginn Fehler an, die nicht aufgetreten sind. Erst nach der Datenanforderung für das allererste Datenbyte (Statusbit 1) oder nach Abbruch des Befehls (Statusbit 0) sollte man die übrigen Statusbits abfragen. Dieser Fehler wirkt sich allerdings nur aus, wenn man den Datentransfer durch Polling abwickelt. □

Copyright Redaktion

HARDWARE * SOFTWARE

COMPUTER-
TANDY-/ GENIE- ANWENDER
CLUB 80

ERFAHRUNGSAUSTAUSCH

Ausgabe Juni 89

Club 80 Sonderinfo

Genie/TRS-Benutzeroberfläche

GEGRAS

Auf vielen Computern gibt es Systeme, welche eine Benutzeroberfläche für die DOS-Funktionen darstellen. Der Benutzer soll damit von vielen Tastatureingaben befreit werden und es soll auch die triste DOS-Kommandoebene aufgelockert werden.

Bisher gab es eine solche Benutzeroberfläche für die GENIE/TRS Rechner noch nicht.

Die Schwierigkeit, solche Systeme zu erstellen, ist der enorme Speicherplatzverbrauch, den solche Systeme haben. Desweiteren wird eine hochauflösende Grafik gebraucht.

Ein weiterer, nicht zu unterschätzender Faktor ist die Geschwindigkeit des Rechners. Rechner mit grafischen Benutzeroberflächen haben Taktfrequenzen von 8 Mhz und höher. Da wird es auf den GENIE/TRS-Rechnern mit ihren 1.75 Mhz doch schon langsam. Wegen dieser, im Vergleich langsamen, Geschwindigkeit, wird empfohlen, eine Geschwindigkeitserhöhung auf 3.5 Mhz vorzunehmen. Wie dieses geschieht, stand in einer älteren Clubinfo. Eine Platine kann bei mir gegen Selbstkostenpreis bezogen werden.

Als Name für die Benutzeroberfläche wurde GEGRAS gewählt. GEGRAS steht für GENIE GRAPHIK SYSTEM.

GEGRAS läuft unter dem Treiberprogramm HRGPACK(r), welches leider aus Copyrightgründen nicht mitgeliefert werden kann.

Es empfiehlt sich vor dem Arbeiten mit GEGRAS, dieses Handbuch, welches als Sonderinfo erscheint, erst einmal durchzulesen.

Für Anregungen, die sich auf künftige Änderungen und/oder Verbesserungen und Vorschläge beziehen, bin ich immer dankbar.

Sicher gibt es auch noch einige Fehler in dem System, doch bitte ich, diese mir zu verzeihen, da ich dieses System in meiner Freizeit erstellt habe.

1.0 Installation

Da GEGRAS nicht mit dem HRGPACK und auch nicht mit einem Betriebssystem ausgeliefert wird (Copyrightgründe), muß es noch installiert werden.

Bei den folgenden Anweisungen wird von zwei Diskettenlaufwerken ausgegangen.

Wie geht die Installation nun vor sich?

Als erstes nimmt man sich eine leere Diskette zur Hand und formatiert diese.

Unter NEWDOS(r) wird dieses mit

FORMAT 0 oder FORMAT 1

gemacht. Anschließend wird das Betriebssystem auf diese Diskette kopiert.

COPY 0,1 oder
COPY 0,1,,NFMT,CBF

Natürlich kann man beides auch in einem kompletten Gang erledigen:

COPY 0,1,,FMT,CBF

Es gibt auch noch andere Möglichkeiten. Hier wählt jeder Benutzer am Besten seinen eigenen Weg.

Als Betriebssystem wird NEWDOS 80/2 vorgeschlagen, da nicht alle Funktionen unter anderen Systemen funktionieren. Deshalb beziehen sich alle folgenden Angaben auf dieses Betriebssystem.

Hat man nun eine Systemdiskette erstellt, werden alle benötigten Dateien der HRGPACK(r) Diskette auf diese kopiert. Wie, daß steht in den einschlägigen Handbüchern.

Jetzt werden alle Dateien von der GEGRAS-Diskette auf diese Diskette mit

COPY 1,0,,NFMT,CBF,USR,NDMW o.ä.

kopiert und es wird das AUTO-Kommando auf

AUTO DO DO

gesetzt.

Ist man noch im Besitz der Tabellenkalkulation VISICALC(r), so kann diese (VC/CMD) auch auf die neue GEGRAS Diskette kopiert werden.

Für einen passenden Diskettenaufkleber gibt man folgende Befehlsequenz ein:

BASIC,RUN "DISKL/BAS"

Es wird dann ein Diskettenaufkleber auf Endlosaufkleber gedruckt.

Damit wäre die Installations abgeschlossen.

Die Diskette muß nun nach einem RESET booten und es muß sich nach einiger Zeit das Titelbild zeigen:

GECAD

(c) 1990

Frank Schoof

Ist dieses nicht der Fall, so bitte alle Schritte noch einmal durchgehen. Oft kann schon eine falsche Taste den ganzen Installtionsvorgang zunichte machen.

2.0 GEGRAS Funktionen und Erläuterungen.

Wurde das SYSTEM ordnungsgemäß gebootet, so erscheint das Titelbild und nach einiger Zeit, die das System zum vorbereiten benötigt, das folgende Bild.

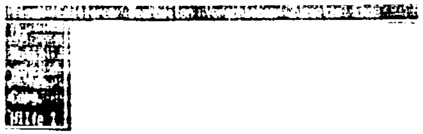
Mit den (PFEILTASTEN) kann nun der Cursor bewegt werden und mit der (LEERTASTE) kann ein Punkt angewählt werden.

Die Funktionen der Menüpunkte sollen nun erklärt werden.

2.1 TISCH

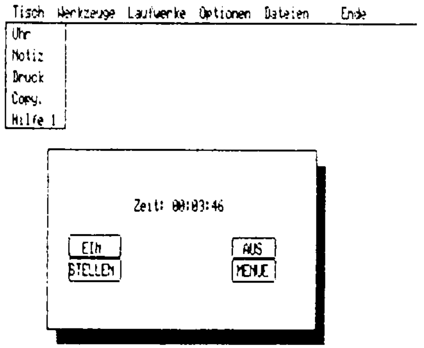
Stellt Utensilien zur Verfügung. Dazu gehört eine Uhr, eine Notizfunktion, eine Notizdruckfunktion, eine Copyrightfunktion und eine Hilfsfunktion.

Nach dem 'anklicken' des Menüpunkts TISCH erscheint auf dem Monitor:



2.1.1 UHR

'Klickt' man diesen Menüpunkt an, so wird folgendes Bild auf dem Monitor sichtbar.



Mit dem Cursor kann nun die Uhr ausgeschaltet, eingeschaltet ode gestellt werden.

Soll die Uhr gestellt werden, so wird einfach der Fensterpunkt 'STELLEN' angefahren und dann 'angeklickt'. Es erscheint dann der Eingabecursor und es kann die neue Uhrzeit eingegeben werden.

Abgeschloßen wird diese Eingabe mit der (ENTER) Taste.

Wird die Uhr eingeschaltet, so erscheint sie in der rechte untern Ecke.

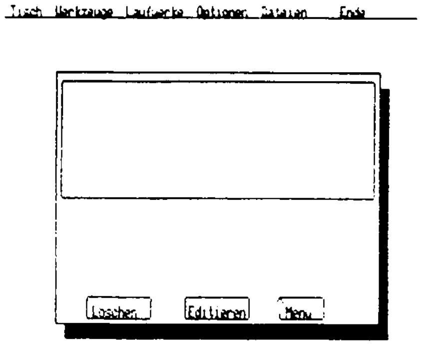
Hat man die Uhr gestellt, ein- oder ausgeschaltet, so wird der Fensterpunkt 'MENÜ' mit der (LEERTASTE) 'angeklickt' und das Uhrenfenster verschwindet wieder.

Die Uhr bleibt bei jedem Diskettenzugriff stehen.

2.1.2 NOTIZ

Es kann hiermit eine 200 Zeichen lange Notiz erstellt, editiert oder gelöscht werden.

Abgeschloßen wird die Notizeingabe damit, daß die (ENTER) Taste solange gedrückt wird, bis der Textcursor verschwindet.



GEGRAS

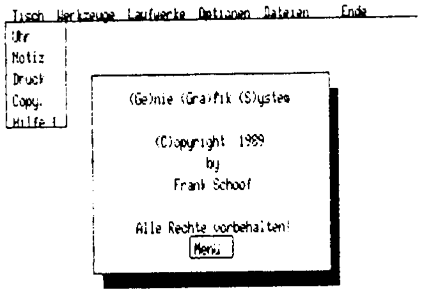
2.1.3. DRUCK

Ermöglicht es, die Notiz auszudrucken.

Nach dem Ausdruck kehrt das Programm zurück in die Menüebene.

2.1.4 COPY.

Zeigt das Copyright des Programmes an.



2.1.6. HILFE 1

Es werden diese Seiten als kurze Textseiten auf dem Bildschirm angezeigt.

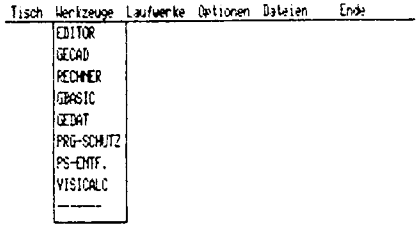
Mit den (PFEILTASTEN) kann zwischen den Seiten geblättert werden.

Welche Taste welche Funktion hat, steht immer in der untersten Zeile.

2.2 WERKZEUGE

Da unter GEGRAS nicht immer alle Werkzeuge wie eine Tabellenkalkulation oder ein CAD-Programm wie GECAD im Speicher gehalten werden können, gibt es diesen nützlichen Menüpunkt.

Wird er angewählt, so erscheint auf dem Monitor:



Derzeit kann also zwischen acht Werkzeugen gewählt werden. Hierzu gehören:

EDITOR	-->	Ein ASCII-Editor.
GECAD	-->	Ein CAD-Programm.
RECHNER	-->	Ein Taschenrechner.
GBASIC	-->	Ein SUPERHRG(r) Umsetzer.
GEDAT	-->	Eine Datenbank.
PRG-SCHUTZ	-->	Schützt ein BASIC Programm.
PS-ENTF.	-->	Entfernt diesen Schutz.
VISICALC	-->	Startet die Tabellenkalkulation VISICALC(r).

2.3 LAUFWERKE

Hiermit kann man auf DOS-Funktionen wie DIR, FORMAT und COPY zurückgreifen.

2.3.1 INHALT 0

Listet die Dateien im Laufwerk 0 auf. Entspricht DIR 0 im DOS.

2.3.2 INHALT 1

Listet die Dateien im Laufwerk 1 auf. Entspricht DIR 1 im DOS.

2.3.3 FORMAT 0

Formatiert die Diskette in Laufwerk 0. Entspricht FORMAT 0 im DOS. VORSICHT !!!

2.3.4 FORMAT 1

Formatiert die Diskette in Laufwerk 1. Entspricht FORMAT 1 im DOS. VORSICHT !!!

2.3.5 CLEAR 0

Löscht alle USR-Dateien im Laufwerk 0. Entspricht dem DOS-Befehl PURGE 0,USR.

2.3.6 CLEAR 1

Löscht alle USR-Dateien im Laufwerk 1. Entspricht dem DOS-Befehl PURGE 1,USR.

2.3.7 KOPIE 0:1

Kopiert die Dateien der Diskette in Laufwerk 0 auf die Diskette im Laufwerk 1.

Unter NEWDOS(r) ist dieses der Befehl COPY 0,1,,NFMT,CBF.

2.3.8 KOPIE 1:0

Kopiert die Dateien der Diskette in Laufwerk 1 auf die Diskette im Laufwerk 0.

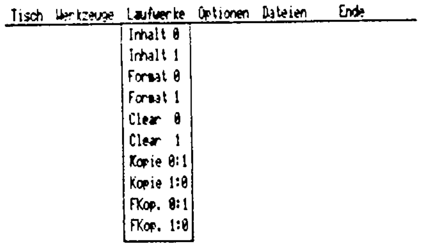
Unter NEWDOS(r) ist dieses der Befehl COPY 1,0,,NFMT,CBF.

2.3.9 FKOPIE 0:1

Wie 2.3.7 nur, daß die Diskette in Laufwerk 1 formatiert wird.

2.3.10 FKOPIE 1:0

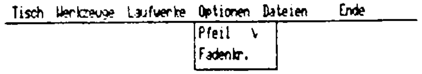
Wie 2.3.8 nur, daß die Diskette in Laufwerk 0 formatiert wird.



2.4 OPTIONEN

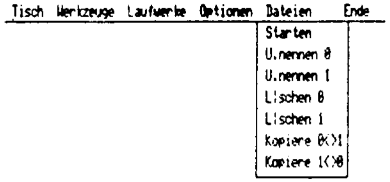
Hier kann eingestellt werden, ob der Cursor als Pfeil ode Fadenkreuz erscheint.

Hinter der aktuellen Einstellung erscheint ein Haken.



2.5 DATEIEN

Nach dem Anwählen dieses Feldes erscheint dieses Bild:



Wie zu sehen ist, können hier Programme umbenennen, löschen oder kopieren.

2.5.1 STARTEN

Mit diesem Menüpunkt können Maschine werden. Diese müssen als Endung /CMD haben.

Es eignen sich jedoch nicht alle Programme zum Starten. Hier hilft nur ausprobieren.

2.5.2 U.NENNEN 0

Hiermit können Dateien auf der Diskette in Laufwerk 0 umbenannt werden.

Die Dateien werden mit dem Pfeil ausgewählt. Sind die Dateien ausgewählt, so wird der Cursor ganz nach oben gefahren und dann die (LEERTASTE) gedrückt.

Nun erscheint der alte Name und die Aufforderung, den neuen Name einzugeben. Abgeschloßen wird die Eingabe mit der (ENTER) Taste.

2.5.3 U.NENNEN 1

Wie 2.5.2, nur daß sich alles auf die Diskette in Laufwerk 1 bezieht.

2.5.4 Löschen 0

Hiermit können einzelne Dateien der Diskette in Laufwerk 0 gelöscht werden.

Die Auswahl erfolgt wie bei 2.5.2 und 2.5.3 !

2.5.5 Löschen 1

Wie 2.5.4 nur bezogen auf Laufwerk 1.

2.5.6 Kopiere 0 1

Mit dieser Funktion können einzelne oder mehrere Dateien von der Diskette in Laufwerk 0 auf die Diskette in Laufwerk 1 kopiert werden.

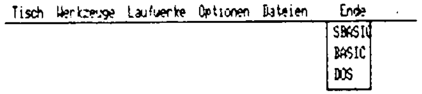
Die Auswahl erfolgt wie bei 2.5.2!

2.5.7 Kopiere 1 0

Wie 2.5.6, nur auf Laufwerk 1 als Original- und Laufwerk 0 als Kopiediskette.

2.6 ENDE

An dieser Stelle kann gewählt werden, ob das System zurück ins normale DISKBASIC(r) springen soll, ob zurück ins DOS gesprungen werden soll, oder ob mit dem HRGPACK(r) weitergearbeitet werden soll.



2.6.1 SBASIC

Kehrt nach einer Sicherheitsabfrage zurück in das HRGPACK(r).

2.6.2 BASIC

Kehrt in das normale DISKBASIC zurück.

2.6.3 DOS

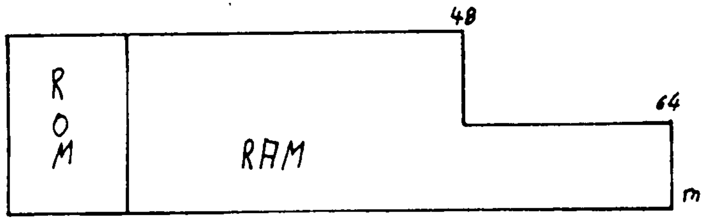
Nach einer Sicherheitsabfrage befindet sich der Benutzer wieder in der DOS-Ebene.

GEGRAS

3.0 SYSTEMPROGRAMME

Wie aus der großen Menge an Zusatzprogrammen ersichtlich ist, verbraucht GEGRAS eine Menge Speicherplatz.

Und da die GENIE/TRS-Modelle in der Regel nur 64(48) KB RAM



haben, ist es leider unmöglich, alle diese Unterprogramme immer im Speicher zu haben.

Deshalb wurde ein Art Overlay Technik gewählt. Es werden dabei die Zusatzprogramme, sofern sie nicht zu groß sind (GECAD), in einen bestimmten Zeilennummernbereich geladen und nach dem Gebrauch wieder gelöscht.

Größere Programme werden per CHAIN-Datei gestartet.

Damit der Programmierer sich selbst Zusatzprogramme erstellen kann, hier einige Vereinbarungen, an die sich bitte die Programmierer halten um es auch anderen leicht zu machen.

Zeilen 32000–39000: Raum für Unterprogramme
Zeilen 40000–41000: Dürfen nicht überschrieben werden.

Die Unterprogramme werden als normale BASIC-Programme gespeichert und dürfen keine Maschinenspracheteile enthalten.

Angesprungen werden diese Unterprogramme oder auch Module mit GOTO 32140 !!!

Verlassen werden können die Module mit GOTO 41000 !!!

4.0 GRAFISCHE ELEMENTE

Für die grafische Gestaltung von GEGRAS-Programmen und Modulen stehen folgende Routinen zur Verfügung:

Zeile 170 :	Hauptmenüzeile
Zeile 3320:	Fenster
Zeile 3540:	Kasten mit runden Kanten

5.0 MODULE

Zur besseren Übersicht sind nachfolgend alle Module aufgelistet.

FMODULMA

```
32140 GOSUB170
32150 X=80:Y=70:X1=260:Y1=110:GOSUB3310
32160 PRINT@465,"Maskenname:";:LINEINPUTNA$
32170 IFNA$=""THENPRINT"Hier Ruecksprung"
32180 GOSUB170
32200 FORI=0TO15:FORO=0TO63:S$(I,O)=" ":NEXT:NEXT
32210 GOSUB170
32220 :
32230 X=0:Y=1
32240 T$=""
32250 P=ASC(S$(Y,X)):PRINT@X+64*Y,"<";
32260 PRINT@X+Y*64,"<";
32270 T$=INKEY$:IFT$=""THEN32270
32280 PRINT@X+Y*64,CHR$(P);
32290 S$(Y,X)=CHR$(P)
32300 PRINT@64*Y+X,CHR$(P);
32310 T=ASC(T$)
32320 IFT=91THENIFY>1THENY=Y-1:GOTO32250
32330 IFT>31THENIFT<127THENPRINT@X+Y*64,CHR$(T);:S$(Y,X)=CHR$(T):X=X+1:GOTO32250
32340 IFT=8THENIFX>0THENX=X-1
32350 IFT=9THENIFX<63THENX=X+1
32360 IFT=10THENIFY<15THENY=Y+1
32370 IFT=13THENIFY<15THENY=Y+1:X=0
32380 IFT=31THEN32200
32390 IFT=26THEN32410
32400 GOTO32250
32410 FORY=1TO14
32420 FOR X=0 TO 63
32430 ZE=ASC(S$(Y,X))
32440 IFZE>32THENIFZE<>ASC(".")THENIFZE<>ASC("*")THENIFZE<>58THENZ$=Z$+CHR$(ZE)
32450 IFZE=58THENSA$(Z)=Z$:Z$="":SS(Z,1)=X-LEN(SA$(Z)):SS(Z,2)=Y:Z=Z+1
32460 IFZE=ASC(".")THENLE=LE+1
32470 IFZE=ASC("*")THENLA(Z-1)=LE:LE=0
32480 NEXT:NEXT
32490 OPEN"O",#1,NA$+"/fld"
32500 PRINT#1,Z
32510 FORI=0TOZ
32520 PRINT#1,SA$(I):LPRINT SA$(I)
32530 NEXT
32540 FORI=0TOZ
32550 PRINT#1,LA(I)
32560 NEXT
32570 FORI=0TOZ
32580 PRINT#1,SS(I,1)
32600 PRINT#1,SS(I,2)
32610 NEXT
32620 CLOSE
32630 GOTO41000
```

FMODULU

```
32140 X=40:Y=80:X1=290:Y1=180
32150 GOSUB3310
32160 PRINT@596,"Zeit: ";MID$(TIME$,10,LEN(TIME$));
32170 PRINT@715," EIN ";
32180 PRINT@740," AUS ";
32190 PRINT@777," STELLEN ";
32200 PRINT@803," MENUE ";:GO=0
32210 X=60:Y=128:X1=108:Y1=140:GOSUB3540
32220 X=60:Y=141:X1=108:Y1=153:GOSUB3540
32230 X=212:Y=129:X1=252:Y1=140:GOSUB3540
32240 X=212:Y=141:X1=252:Y1=153:GOSUB 3540
32250 GOSUB80
32260 IFGO=1THEN32280
32270 GOTO32250
32280 IFX>211THENIFX<253ANDY>140ANDY<154THENGO=0:GOTO 41000
32290 GO=0
32300 IFX>211THENIFX<253ANDY>128ANDY<141THENPOKE16709,0
32310 IFX>59THENIFX<109ANDY>127ANDY<141THENPOKE16709,1
32320 IFX>59THENIFX<109ANDY>140ANDY<154THEN32340
32330 GOTO32250
32340 PRINT@602,"";:LINEINPUTTI$
32350 D$="TIME "+TI$
32360 CMDD$
32370 GOTO32250
```

FMODULE

```
32140 CLEAR6700
32150 CLH
32160 DEFINTA-Z
32170 CLS
32180 CLH
32190 LINK(0,179,382,179,1)
32200 LINK(0,179,0,191,1)
32210 LINK(0,191,382,191,1)
32220 LINK(382,191,382,179,1)
32230 LINK(0,0,0,178,1)
32240 LINK(382,0,382,178,1)
32250 DIMZE$(100)
32260 Z=0
32270 ZM=0
32280 FORI=0TO100:ZE$(I)=STRING$(64," "):NEXT
32290 IFZ<0THENZ=0
32300 FORI=0TO14
32310 PRINT@I*64,ZE$(I+Z);
32320 NEXT
32330 ZE=PEEK(15360+X+Y*64):PRINT@X+Y*64,">";
32340 PRINT@960,"X: ";X;
32350 PRINT@970,"Y: ";Y+Z;
32360 T$=INKEY$
32370 IFY+Z>ZMTHENZM=Z+Y
32380 IFT$=""THEN32360
32390 POKE15360+X+Y*64,ZE
```

```
32400 IFT$=CHR$(10)THENY=Y+1
32410 IFT$=CHR$(13)THENY=Y+1:X=0
32420 IFT$=CHR$(91)THENY=Y-1
32430 IFT$=CHR$(8)THENX=X-1
32440 IFT$=CHR$(9)THENX=X+1
32450 IFY=15THENY=14:Z=Z+1:GOTO32290
32460 IFY=-1THENY=0:Z=Z-1:GOTO32290
32470 IFT$=CHR$(91)THEN32490
32480 IFT$>CHR$(31)ANDT$<CHR$(125)THENPRINT@X+Y*64,T$;:MID$(ZE$(Z+Y),X+1,1)=T$:X=X+1
32490 IFT$=CHR$(31)THENX=0:Y=0
32500 IFT$=CHR$(25)THENZE$(Z+Y)=STRING$(64,32):GOTO32290
32510 IFT$=CHR$(27)THEN32550
32520 IFT$=CHR$(26)THEN32640
32530 IFT$=CHR$(24)THEN#PRINT2:GOTO41000
32540 GOTO32330
32550 PRINT@980,"";:LINEINPUT"Sichern Datei: ";DA$
32560 PRINT@980,"                              ";
32570 OPEN"O",#1,DA$
32580 FORI=0TOZM
32590 PRINT#1,ZE$(I)
32600 NEXT
32610 CLOSE
32620 CLS
32630 GOTO32330
32640 PRINT@980,"";:LINEINPUT"Laden Datei: ";DA$
32650 PRINT@980,"                                  ";
32660 OPEN"I",#1,DA$
32670 Z=0
32680 ZE$(Z)="":LINEINPUT#1,ZE$(Z)
32690 Z=Z+1
32700 IFNOTEOF(1)THEN32680
32710 CLOSE
32720 CLS
32730 GOTO32300
```

PMODULC

```
32140 XS=X:YS=Y:IFLW=0THENDI$="DIR 0"
32150 IFLW=1THENDI$="DIR 1"
32160 CLS:CLH
32170 CMDDI$
32180 X=10:Y=1
32190 AD=15360

32200 NA$=""
32210 FORI=1TO8
32220 NA$=NA$+CHR$(PEEK(AD+Y*64+X+I))
32230 NEXT
32240 X=20
32250 FORI=1TO8
32260 DA$=DA$+CHR$(PEEK(AD+Y*64+X+I))
32270 NEXT
32280 X=0:Y=3
32290 FORI=0TO11
32300 DD$(ZD)=DD$(ZD)+CHR$(PEEK(AD+X+I+Y*64))
32310 NEXT
32320 IFMID$(DD$(ZD),1,3)="   "THEN32370
32330 ZD=ZD+1
32340 X=X+15
32350 IFX>52THENX=0:Y=Y+1
32360 IFY<15THEN32290
32370 CLS:CLH
32380 FORI=0TOZD
32390 CO=INSTR(DD$(I)," ")
32400 DD$(I)=MID$(DD$(I),1,CO-1)
32410 IFLW=1THENDD$(I)=DD$(I)+":1"ELSEDD$(I)=DD$(I)+":"
32420 NEXT
32430 GOSUB170
32440 CMDDI$
32450 FORI=0TO64:DD(I)=0:NEXT
32460 X=62:Y=10:X1=118:Y1=22:GOSUB3540
32470 X=122:X1=178:GOSUB3540
32480 X=0:Y=34:X1=76:Y1=44
32490 FORI=0TOZD/4
32500 FORO=1TO4
32510 X1=X+78:GOSUB3540
32520 X=X+88
32530 NEXT
32540 X=0:Y=Y+12:Y1=Y+10
32550 NEXT
32560 X=XS:Y=YS:GOSUB80:XS=X:YS=Y
32570 XP=FIX(X/88)
32580 YP=FIX(Y/10)-4
32590 IFGO=0THEN32560
32600 IFY<34THEN32690
32610 IFDD(YP*4+XP)=0THENDD(YP*4+XP)=1ELSEDD(YP*4+XP)=0
32620 IFY<10THENGOTO41000
32630 Y=34+(12*YP)
32640 X=0+(88*XP)
32650 FORI=Y+1TOY+9STEP1
32660 LINK(X+2,I,X+77,I,-1)
32670 NEXT:X=XS:Y=YS
32680 GOTO32560
32690 FORI=0TOZD
32700 IFDD(I)=1THENIFKE=0THENCO$="COPY "+DD$(I)+" TO :1":CMDCO$
32710 IFDD(I)=1THENIFKE=1THENCO$="COPY "+DD$(I)+" TO :0":CMDCO$
32720 NEXT
32730 GOTO41000
```

PMODULPS

```
32140 GOSUB170
32150 X=50:Y=20:X1=300:Y1=140:GOSUB3310
32160 PRINT@211,"Programmschutz V 1.0";
32170 X=98:Y=32:X1=254:Y1=50:GOSUB3540
32180 PRINT@458,"Originaldatei: ";
32190 PRINT@586,"Schutzdatei  : ";
32200 LINK(150,92,221,92,1)
32210 LINK(150,166,221,116,1)
32220 PRINT@473,"";:LINEINPUTDA$
32230 PRINT@601,"";:LINEINPUTSD$
32240 IFSD$=DA$THEN32140
32250 PRINT@458,STRING$(30,32);
32260 PRINT@586,STRING$(30,32);
32270 OPEN"R",#1,DA$
```



```
32280 LA=LOF(1)
32290 CLOSE#1
32300 LA=LA*245
32310 PRINT@459,"0";
32320 PRINT@466,"50";
32330 PRINT@474,"100";:PRINT@482,"150";:PRINT@490,"200 %";
32340 FORI=63TO263STEP10:LINK(I,90,I,98,-1):NEXT
32341 X=56:Y=74:X1=294:Y1=128:GOSUB3540
32350 OPEN"I",#1,DA$:OPEN"O",#2,SD$
32360 LINEINPUT#1,A$
32370 LE=LE+LEN(A$)
32380 PR=(LE*100)/LA
32390 IFP1=PRTHEN32410
32400 LINK(63+PR,100,63+PR,110,1)
32410 P1=PR
32420 A$=A$+":'"+CHR$(28)+CHR$(31)+STRING$(2,11)
32430 PRINT#2,A$
32440 LI=LI+LEN(A$)
32450 P2=(LI*100)/LA:IFP2=P3THEN32470
32460 FORL2=112TO122STEP2:LINK(63+P2,L2,63+P2,L2,1):NEXT
32470 P3=P2
32480 IFNOTEOF(1)THEN32360
32490 P2=0:P3=0:SD$="":DA$="":LA=0:LE=0:PR=0:L2=0
32500 CLOSE:GOTO41000
```

PMODULR

```
32000 Z4$=Z1$:Z1$=Z2$:MID$(Z1$,15,1)="."
32010 RETURN
32020 Z1=VAL(Z1$)
32030 Z2=VAL(Z4$)
32040 IFRO=1THENZ3=Z1*Z2
32050 IFRO=2THENZ3=Z2/Z1
32060 IFRO=3THENZ3=Z1+Z2
32070 IFRO=4THENZ3=Z2-Z1
32080 Z1$=STR$(Z3)
32090 RO=0
32100 Z3$=Z1$:Z1=0:Z2=0:Z3=0:Z4$=Z1$
32110 IFINSTR(Z3$,".")>0THENZ1$=Z2$:MID$(Z1$,15-LEN(Z3$),LEN(Z3$))=Z3$:GOTO32360
32120 IFINSTR(Z3$,".")=0THENZ1$=Z2$:MID$(Z1$,15-LEN(Z3$)-1,LEN(Z3$)+1)=Z3$+".":GOTO32360
32130 GOTO32360
32140 GOSUB170
32150 GO=0:X=160:Y=50:X1=360:Y1=170:GOSUB3320
32160 PRINT@480,"7   8   9   -    %    @PI ";
32170 PRINT@608,"4   5   6   +   STO    [ ";
32180 PRINT@736,"1   2   3   /   RCL   CLE";
32190 PRINT@864,"0   .   +-  *   CE/C";
32200 FORX=185TO265STEP24
32210 FORY=80TO170STEP24
32220 X1=X+20:Y1=Y+13:GOSUB3540
32230 NEXT
32240 NEXT
32250 X=285:Y=152:X1=320:Y1=165:GOSUB3540
32260 X=195:Y=55:X1=330:Y1=73:GOSUB3540
32270 FORX=285TO325STEP35
32280 FORY=80TO150STEP24
32290 X1=X+25:Y1=Y+13:GOSUB3540
32300 NEXT:NEXT
32310 Z1$=STRING$(15,32)
32320 PU=0
```

```
32  2$=Z1$
32  1=14:Z2=Z1
32  MID$(Z1$,14,2)=" ."
32  PRINT@356,Z1$;
32  $=INKEY$:IF T$="" THEN 32370
32  A=ASC(T$)
32  IFTA=67ORTA=99THENZS$=""
3   IF T$=CHR$(13) THEN 41000
3   IFTA=42THENRO=1:GOSUB32000
3   IFTA=47THENRO=2:GOSUB32000
3   IFTA=43THENRO=3:GOSUB32000
3   IFTA=45THENRO=4:GOSUB32000
3   FTA=61THEN32020
3   FTA=57THENZ3$="9"
3   FTA=56THENZ3$="8"
3   IFTA=55THENZ3$="7"
3   FTA=54THENZ3$="6"
3   IFTA=53THENZ3$="5"
3   IFTA=82ORTA=114THENZ1$=ZS$:GOTO32360
32  IFTA=91THENZ3=VAL(Z1$)*VAL(Z1$):Z3$=STR$(Z3)
32  IFTA=91THEN32110
32  IFTA=52THENZ3$="4"
3   FTA=51THENZ3$="3"
3   IFTA=50THENZ3$="2"
3   FTA=49THENZ3$="1"
3   IFTA=48THENZ3$="0"
32  FTA=46THENIFPU=0THENZ3$=".":MID$(Z1$,15,1)=" ":PU=1
3   FTA=31THEN32310
3   FTA=37THENZ3=VAL(Z1$)/100:Z1$=Z2$:Z3$=STR$(Z3):MID$(Z1$,15-LEN(Z3$),LEN(Z3$))=Z3$:GOTO32360
3   FTA=64THENZ1$=Z2$:Z3$="3.14159":MID$(Z1$,15-LEN(Z3$),LEN(Z3$))=Z3$:GOTO32360
3   FTA=83ORTA=115THENZS$=Z1$
3   FZ3$=""THENIFGO=0THEN32370
32  0=0
3   F X<160 OR X>360 OR Y<50 OR Y>170 THEN GOSUB 170:GOTO 34000
3   ORI=1TO13
3   ID$(Z1$,I,1)=MID$(Z1$,I+1,1)
3   EXT
3   FZ1>0THENMID$(Z1$,Z1+1-LEN(Z3$),LEN(Z3$))=Z3$
3   3$=""
32  OTO32360
34  OTO41000
```

PMODULG

```
32  RINT#2,"40000 K=1"
32  RINT#2,"40010 XS=X:YS=Y"
32  RINT#2,"40020 #LINE(X,Y)-(X1,Y),RESET:#LINE(X1,Y)-(X1,Y1),RESET:#LINE(X1,Y1)-(X,Y1),RESET:#LINE(X,Y1)-(X,Y),RESET"
3   RINT#2,"40030 #LINE(X1+9,Y+9)-(X1+9,Y1+9),RESET:#LINE(X1+9,Y1+9)-(X+9,Y1+9),RESET"
3   RINT#2,"40040 FOR X=X1+1 TO X1+9:#LINE(X,Y+9)-(X,Y1+9),POINT:NEXT:X=XS:Y=YS"
32  RINT#2,"40050 FOR Y=Y1+1 TO Y1+9:#LINE(X+9,Y)-(X1,Y),POINT:NEXT:X=XS:Y=YS:RETURN"
32  ETURN
3
3
3
3
3
3
3
32  OSUB170
3   =60:Y=50:X1=300:Y1=170:GOSUB3310
```



```
32160 PRINT@335,"GBASICdateiname: ............";
32170 PRINT@463,"BASICdateiname : ............";
32190 PRINT@590,"Bitte geben Sie die gewuenschten";
32200 PRINT@654,"Dateinamen ein und druecken Sie ";
32210 PRINT@718,"dann <NEW LINE>. Moechten Sie in";
32220 PRINT@782,"das Menue, so geben Sie bitte **";
32230 PRINT@846,"ein !!!";
32240 PRINT@352,"";:LINEINPUTGB$
32250 IFGB$="**"THEN41000
32260 PRINT@480,"";:LINEINPUTBA$
32270 IFBA$="**"THEN41000
32280 GOSUB170
32290 X=60:Y=50:X1=350:Y1=170:GOSUB3310
32300 PRINT@334,GB$+":";
32310 PRINT@526,BA$+":";
32320 X=86:Y=70:X1=312:Y1=90:GOSUB3540
32330 Y=108:Y1=128:GOSUB3540
32340 PRINT@780,"Druecken einer <TASTE> bricht Vorgang ab!";
32350 OPEN"I",#1,GB$
32360 OPEN"O",#2,BA$
32370 LINEINPUT#1,Z$
32371 TA$=INKEY$:IFTA$>""THEN41000
32380 L1=L1+LEN(Z$)
32390 FORI=87TO87+L1/100
32400 LINK(I,71,I,89,1)
32410 NEXT
32420 GOSUB33000
32430 FORI=87TO87+L2/95
32440 LINK(I,109,I,127,1)
32450 NEXT
32460 IFNOTEOF(1)THEN32370
32461 GOSUB32000
32470 CLOSE
32480 GOTO41000
33000 CO=INSTR(Z$," ")
33010 NU$=MID$(Z$,1,CO-1)
33020 T$=MID$(Z$,CO+1,LEN(Z$))
33030 CO=INSTR(T$,"BOX")
33040 IFCO>0THENGOSUB33160
33041 CO=INSTR(T$,"PLOT")
33042 IFCO>0THENGOSUB34600
33050 CO=INSTR(T$,"WINDOW")
33060 IFCO>0THENGOSUB33390
33070 CO=INSTR(T$,"SCREEN")
33080 IFCO>0THENGOSUB33510
33090 CO=INSTR(T$,"INVERS")
33100 IFCO>0THENGOSUB33690
33101 CO=INSTR(T$,"WHILE")
33102 IFCO>0THENGOSUB35000
33103 CO=INSTR(T$,"WEND")
33104 IFCO>0THENGOSUB35060
33105 CO=INSTR(T$,"WAIT")
33106 IFCO>0THENGOSUB34000
33107 CO=INSTR(T$,"AT")
33108 IFCO>0THENGOSUB34500
33110 IFZE$>""THENZ$=ZE$ELSEZ$=T$
33120 ZE$=NU$+" "+Z$
33121 L2=L2+LEN(ZE$)
33130 PRINT#2,ZE$
33140 ZE$=""
```

```
33141 Z$=""
33150 RETURN
33160 Z1=ASC(MID$(T$,4,1))
33170 IFZ1=32THENZ1$=MID$(T$,5,LEN(T$))
33180 IFZ1<>32THENZ1$=MID$(T$,4,LEN(T$))
33190 C1=INSTR(Z1$,",")
33200 C2=INSTR(MID$(Z1$,C1+1,LEN(Z1$)),",")
33210 C3=INSTR(MID$(Z1$,C1+C2+1,LEN(Z1$)),",")
33220 C4=INSTR(MID$(Z1$,C1+C2+C3+1,LEN(Z1$)),",")
33230 T1$=MID$(Z1$,1,C1-1)
33240 T2$=MID$(Z1$,C1+1,C2-1)
33250 T3$=MID$(Z1$,C2+1+C1,(C1+C2+C3)-(C1+C2)-1)
33260 T4$=MID$(Z1$,C1+C2+C3+1,(C1+C2+C3+C4)-(C1+C2+C3)-1)
33270 T5$=MID$(Z1$,C1+C2+C3+C4+1,LEN(T$))
33280 IFT5$="0"THENGOSUB33330
33290 IFT5$="1"THENGOSUB33350
33300 IFT5$="2"THENGOSUB33370
33310 ZE$="X="+T1$+":Y="+T2$+":X1="+T3$+":Y1="+T4$+":"+ZE$
33320 RETURN
33330 ZE$="#LINE(X,Y)-(X1,Y),SET:#LINE(X1,Y)-(X1,Y1),SET:#LINE(X1,Y1)-(X,Y1),SET:#LINE(X,Y1)-(X,Y),SET"
33340 RETURN
33350 ZE$="#LINE(X,Y)-(X1,Y),RESET:#LINE(X1,Y)-(X1,Y1),RESET:#LINE(X1,Y1)-(X,Y1),RESET:#LINE(X,Y1)-(X,Y),RESET"
33360 RETURN
33370 ZE$="#LINE(X,Y)-(X1,Y),POINT:#LINE(X1,Y)-(X1,Y1),POINT:#LINE(X1,Y1)-(X,Y1),POINT:#LINE(X,Y1)-(X,Y),POINT"
33380 RETURN
33390 Z1=ASC(MID$(T$,7,1))
33400 IFZ1=32THENZ1$=MID$(T$,8,LEN(T$))
33410 IFZ1<>32THENZ1$=MID$(T$,7,LEN(T$))
33420 C1=INSTR(Z1$,",")
33430 C2=INSTR(MID$(Z1$,C1+1,LEN(Z1$)),",")
33440 C3=INSTR(MID$(Z1$,C1+C2+1,LEN(Z1$)),",")
33450 T1$=MID$(Z1$,1,C1-1)
33460 T2$=MID$(Z1$,C1+1,C2-1)
33470 T3$=MID$(Z1$,C2+1+C1,(C1+C2+C3)-(C1+C2)-1)
33480 T4$=MID$(Z1$,C1+C2+C3+1,LEN(Z1$))
33490 ZE$="X="+T1$+":Y="+T2$+":X1="+T3$+":Y1="+T4$+":"+"GOSUB 40000"
33500 RETURN
33510 Z1=ASC(MID$(T$,7,1))
33520 IFZ1=32THENZ1$=MID$(T$,8,LEN(T$))
33530 IFZ1<>32THENZ1$=MID$(T$,7,LEN(T$))
33540 C1=INSTR(T$,",")
33550 IFC1<>0THEN33560ELSE33650
33560 V1=VAL(MID$(Z1$,1,1))
33570 V2=VAL(MID$(Z1$,3,1))
33580 IFV1=1THENIFV2=0THENZ$="#PRINT 0"
33590 IFV1=1THENIFV2=1THENZ$="#PRINT 0:FOR I=15360 TO 16383:POKE I,191:NEXT"
33600 IFV1=2THENIFV2=0THENZ$="#PRINT 2"
33610 IFV1=2THENIFV2=1THENZ$="#PRINT 2:#NOT"
33620 IFV1=3THENIFV2=0THENZ$="#PRINT 4"
33630 IFV1=3THENIFV2=1THENZ$="#PRINT 4:#NOT"
33640 GOTO33670
33650 ZE$=T$
33660 GOTO33680
33670 ZE$=Z$
33680 RETURN
33690 Z1=ASC(MID$(T$,7,1))
33700 IFZ1=32THENZ1$=MID$(T$,8,1)
33710 IFZ1<>32THENZ1$=MID$(T$,7,1)
33720 IFZ1$="0"THENZE$="FOR I=0 TO 255:#LET"+CHR$(34)+"G:I:NEXT"+CHR$(34)
```



```
33730 IFZ1$="1"THENZE$="FOR I=0 TO 255:#LET"+CHR$(34)+"G:I:NEXT"+CHR$(34)
33740 RETURN
34000 Z1=ASC(MID$(T$,5,1))
34010 IFZ1=32THENZ1$=MID$(T$,6,LEN(T$))
34020 IFZ1<>32THENZ1$=MID$(T$,5,LEN(T$))
34030 Z1=VAL(Z1$)*1000
34040 ZE$="FOR I=1 TO "+STR$(Z1)+":NEXT"
34050 RETURN
34500 Z1=ASC(MID$(T$,3,1))
34510 IFZ1=32THENZ1$=MID$(T$,4,LEN(T$))
34520 IFZ1<>32THENZ1$=MID$(T$,3,LEN(T$))
34530 C1=INSTR(Z1$,",")
34540 T1$=MID$(Z1$,1,C1-1)
34550 T2$=MID$(Z1$,C1+1,LEN(Z1$))
34560 T2=VAL(T2$)*64+VAL(T1$)
34570 ZE$="PRINT @"+STR$(T2)+","+STRING$(2,34)+";"
34580 RETURN
34600 Z1=ASC(MID$(T$,5,1))
34610 IFZ1=32THENZ1$=MID$(T$,6,LEN(T$))
34620 IFZ1<>32THENZ1$=MID$(T$,5,LEN(T$))
34630 C1=INSTR(Z1$,",")
34640 T1$=MID$(Z1$,1,C1-1)
34650 T2$=MID$(Z1$,C1+1,LEN(Z1$))
34660 ZE$="#SET("+T1$+","+T2$+")"
34680 RETURN
35000 Z1=ASC(MID$(T$,6,1))
35010 IFZ1=32THENWI$(WZ)=MID$(T$,7,LEN(T$))
35020 IFZ1<>32THENWI$(WZ)=MID$(T$,6,LEN(T$))
35030 N1$(WZ)=NU$:WZ=WZ+1
35040 ZE$=":"
35050 RETURN
35060 WZ=WZ-1
35070 ZE$="IF "+WI$(WZ)+" THEN GOTO "+N1$(WZ)
35075 WI$=""
35080 RETURN
```

PMODULBE

```
32140 ZE=2:ZP=0:ZZ=0:ZD=0:Z1=0:Z2=0:X1=0:Y1=0:X=0:Y=0:HH=0:T1=0
32150 GOSUB32510:GOSUB33110
32160 GOSUB32650
32170 GOSUB32770
32180 GOTO41000
32190 LSETA1$=A$(0)
32200 LSETA2$=A$(1)
32210 LSETA3$=A$(2)
32220 LSETA4$=A$(3)
32230 LSETA5$=A$(4)
32240 LSETA6$=A$(5)
32250 LSETA7$=A$(6)
32260 LSETA8$=A$(7)
32270 LSETA9$=A$(8)
32271 IF ZE=1 THEN ZE=2
32280 PUT#DN,ZE
32290 RETURN
32300 GET#DN,ZE
32310 A$(0)=A1$:A$(1)=A2$:A$(2)=A3$:A$(3)=A4$:A$(4)=A5$:A$(5)=A6$:A$(6)=A7$:A$(7)=A8$:A$(8)=A9$
32400 RETURN
32510 OPEN"I",#2,DM$
32520 INPUT#2,ZD
```

```
32530 FORI=0TOZD:LINEINPUT#2,FE$(I):NEXT
32560 FORI=0TOZD:INPUT#2,LA(I):NEXT
32590 FORI=0TOZD:INPUT#2,SS(I,1):INPUT#2,SS(I,2):NEXT
32630 CLOSE#2
32640 RETURN
32650 ZP=ZE:FORI=1TOLOF(DN)
32660 ZE=I
32670 GOSUB32300
32680 FORQ=0TOZD-1:IN$(I)=IN$(I)+A$(Q):NEXT
32690 NEXT
32700 ZE=ZP
32710 RETURN
32770 GOSUB170
32780 X=30:Y=50:X1=290:Y1=160:GOSUB3310
32785 X=36:X1=80
32787 FORY=55TO135STEP24
32788 Y1=Y+17:GOSUB3540:NEXT
32800 PRINT@327,"Eingabe";
32801 PRINT@455,"Dritat";
32802 PRINT@503,"Brief";
32803 PRINT@711,"--------";
32810 FORY=55TO103STEP24:Y1=Y+17
32820 FORX=100TO250STEP64
32830 X1=X+52:GOSUB3540
32840 NEXT:NEXT
32841 KY=0
32850 PRINT@338,"Suchen";
32860 PRINT@466,"L";CHR$(124);"sche";
32870 PRINT@476," Zeige";
32880 PRINT@487,"Drucken";
32890 X=100:Y=113:X1=280:Y1=133:GOSUB3540
32900 PRINT@569,"Men";CHR$(125);
32910 PRINT@349,"Sehen";
32920 PRINT@359,CHR$(91);"ndern";
32930 X=100:Y=115
32940 GOSUB980:IFGO=0THEN32940
32950 IFX>99THENIFX<281THENIFY>112THENIFY<134THEN33050
32960 IFX>35THENIFX<89THENIFY>54THENIFY<73THENGOSUB33490
32970 IFY>54THENIFY<73THEN32980ELSE33010
32980 IFX>99THENIFX<153THENGOSUB33900
32990 IFX>163THENIFX<217THENGOSUB34220
33000 IFX>227THENIFX<281THENGOSUB33700
33010 IFY>78THENIFY<97THEN33020ELSE33040
33020 IFX>99THENIFX<153THENGOSUB35000
33021 IFX>163THENIFX<217THENGOSUB33560
33030 IFX>227THENIFX<281THENDR=1:GOSUB34220
33040 IFX>35THENIFX<89THENIFY>78THENIFY<97THENLP=1:GOSUB33490
33041 IFX>35THENIFX<89THENIFY>102THENIFY<121THENGOSUB35150
33049 GOTO32770
33050 RETURN
33110 OPEN"R",#DN,DA$
33120 IFZD>3THEN33140
33130 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$:GOTO33270
33140 IFZD>4THEN33160
33150 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$:GOTO33270
33160 IFZD>5THEN33180
33170 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$,LA(4)ASA5$:GOTO33270
33180 IFZD>6THEN33200
33190 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$,LA(4)ASA5$,LA(5)ASA6$:GOTO33270
33200 IFZD>7THEN33220
33210 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$,LA(4)ASA5$,LA(5)ASA6$,LA(6)ASA7$:GOTO33270
```



```
33220 IFZD>8THEN33240
33230 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$,LA(4)ASA5$,LA(5)ASA6$,LA(6)ASA7$,LA(7)ASA8$:GOTO33270
33240 IFZD>9THEN33260
33250 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$,LA(4)ASA5$,LA(5)ASA6$,LA(6)ASA7$,LA(7)ASA8$,LA(8)ASA9$:GOTO33270
33260 FIELDDN,LA(0)ASA1$,LA(1)ASA2$,LA(2)ASA3$,LA(3)ASA4$,LA(4)ASA5$,LA(5)ASA6$,LA(6)ASA7$,LA(7)ASA8$,LA(8)ASA9$
33270 LA=LOF(DN):ZE=LA-1:IFZE=0THENZE=1
33300 ZE=ZE+1
33310 RETURN
33320 ZE=LOF(DN)+1
33321 GOSUB170
33330 FORI=0TOZD-1:PRINT@SS(I,1)+SS(I,2)*64,FE$(I);":";STRING$(LA(I),46);:NEXT
33360 FORI=0TOZD-1
33370 PRINT@SS(I,1)+SS(I,2)*64+LEN(FE$(I))+1,"";:LINEINPUTF1$(I)
33380 IFF1$(I)="*ENDE*"THEN33480
33390 NEXT
33400 PRINT@10+838,"Eingaben richtig (J/N) ?";
33410 T$=INKEY$:IFT$=""THEN33410
33420 IFT$="n"ORT$="N"THEN33320
33430 T$=""
33440 GOSUB34090
33441 ZE=LOF(DN)+1
33450 FORI=0TOZD-1:A$(I)=F1$(I):NEXT
33480 RETURN
33490 GOSUB33320
33500 IFF1$(I)="*ENDE*"THENLP=0:GOTO33550
33510 IFOK=1THENOK=0:RETURN
33520 GOSUB32190
33530 IN$(ZE)="":FORI=0TOZD-1:IN$(ZE)=IN$(ZE)+A$(I):NEXT
33541 IFLP=1THEN33490
33550 RETURN
33560 ZP=ZE:GOSUB170:ZE=2
33570 PRINT@0,STRING$(64,32);
33580 PRINT@0,"  < <- >  R";CHR$(125);"ckw";CHR$(123);"rts   < -> > Vorw";CHR$(123);"rts      <LEER> Men";CHR$(125);
33590 LINK(0,9,383,9,1)
33600 T2=PEEK(14440):IFT2=0 THEN 33600
33610 IFT2=32 THENIFZE>2THENZE=ZE-1
33620 IFT2=64HTNEIFZE<LOF(DN)THENZE=ZE+1
33630 IFT2=128 THEN 33690
33650 PRINT@58,"#";ZE;
33660 LINK(0,9,383,9,1)
33670 GOSUB32300
33671 FORI=0TOZD-1:PRINT@SS(I,1)+SS(I,2)*64,FE$(I);":";A$(I);:NEXT
33680 GOTO33600
33690 ZE=ZP:RETURN
33700 GOSUB170:ZP=ZE:X=100:Y=70:X1=200:Y1=90:GOSUB3540:PRINT@401,"Nummer:";:INPUTZE:GOSUB170:GOSUB32300
33701 PRINT@58,"#":ZE;
33710 FORI=0TOZD-1
33720 PRINT@2+(I+2)*64,A$(I);
33730 NEXT
33740 FORI=0TOZD-1
33750 PRINT@34+(2+I)*64,">";
33760 NEXT
33767 IFKK=1THENZE=ZP:RETURN
33770 FORI=0TOZD-1
33780 PRINT@35+I*64,;:LINEINPUTF1$(I)
33790 IFF1$(I)=""THENF1$(I)=A$(I)
33800 NEXT
33810 FORI=0TOZD-1
33820 A$(I)=F1$(I)
33830 NEXT
33840 GOSUB32190
```

```
33850 GOSUB32770
33860 ZE=ZP:RETURN
33870 CLS
33880 CLOSE
33890 END
33900 GOSUB170
33910 ZP=ZE
33920 X=60:Y=60:X1=290:Y1=92:GOSUB3310
33930 PRINT@395,"Suchbegriff:";:LINEINPUTSU$:PRINT@395,STRING$(26,32);
33931 PRINT@395,"Drucken (J/N): ";:LINEINPUTDR$
33932 IFDR$="j"ORDR$="J"THENDR=1
33940 IFSU$=""THENRETURN
33944 KK=1
33950 P=0
33955 ZZ=ZE
33956 HH=1
33960 :
33970 CO=INSTR(IN$(HH),SU$)
33980 IFCO>0THEN33990
33981 HH=HH+1
33989 IFHH<=LOF(DN)THEN33960ELSE34080
33990 Z1=HH:Z2=HH
34055 GOSUB170:GOSUB34251
34056 PRINT@58,"#";ZE;
34070 IFT$=CHR$(13)THENHH=HH+1:GOTO34120
34071 IFKK=1THENKK=0:ZE=ZZ:RETURN
34080 FORP=1TO1000:NEXT
34090 FORI=1TOZE
34100 IFIN$(I)=F1$(1)THEN34130
34110 NEXT
34120 ZE=ZZ:RETURN
34130 GOSUB170
34140 PRINT@2+128,"Es besteht schon ein Eintrag mit diesem";
34150 PRINT@2+192,"                    Namen !!!           ";
34160 PRINT@2+320,"Soll Eintrag trotzdem uebernommen werden ?";
34170 PRINT@2+384,"(Tasten (J)(j) oder (N)(n) druecken !!!)";
34180 T$=""
34190 T$=INKEY$:IFT$=""THEN34190
34200 IFT$="N"ORT$="n"THENOK=1ELSEOK=0
34210 RETURN
34220 GOSUB170
34230 X=90:Y=90:X1=250:Y1=112:GOSUB3310
34240 PRINT@528,"Nummer:";:INPUTNU$
34241 CO=INSTR(1,NU$,"-")
34242 IFCO>0THENZ1=VAL(MID$(NU$,1,CO-1)):Z2=VAL(MID$(NU$,CO+1,LEN(NU$)))
34243 IFCO=0THENZ1=VAL(NUL$):Z2=Z1
34250 ZP=ZE
34251 FORKL=Z1TOZ2
34260 ZE=KL
34270 GOSUB32300
34280 GOSUB170
34281 PRINT@58,"#";ZE;
34290 FORQ=0TOZD-1
34300 PRINT@SS(Q,1)+SS(Q,2)*64,FE$(Q);": ";A$(Q);
34310 IFDR=1THENLPRINTFE$(Q);": ";A$(Q)
34320 NEXT
34321 NEXT
34330 DR=0
34340 T$=""
34350 T$=INKEY$:IFT$=""THEN34350
34360 ZE=ZP
34370 RETURN
34380 RESUME32170
35000 GOSUB170
35010 ZP=ZE
35020 X=60:Y=60:X1=290:Y1=92:GOSUB3310
35030 PRINT@395,"Suchbegriff: ";:LINEINPUTSU$
35040 IFSU$=""THENZE=ZP:RETURN
35050 FORHH=1TOLOF(DN)
35060 CO=INSTR(IN$(HH),SU$)
35070 IFCO>0THENZQ=HH:GOTO35090
35080 NEXT:ZE=ZP:RETURN
35090 ZE=ZQ
35100 GOSUB32300
35110 FORI=0TO8:A$(I)="Leereintrag":NEXT
35120 GOSUB32190
35130 GOTO35200
35140 ZE=ZP:RETURN
35150 ZP=ZE:GOSUB170:X=60:Y=60:X1=290:Y1=92:GOSUB3310
35160 RRINT@395,"Suchbegriff: ";:LINEINPUTSU$
35170 IFSU$=""THENZE=ZP:RETURN
35180 FORHH=1TOLOF(DN)
35190 CO=INSTR(IN$(HH),SU$)
35200 IFCO>0THENZQ=HH:GOTO35220
35210 NEXT:ZE=ZP:RETURN
35220 ZE=ZQ:GOSUB170
35230 GOSUB32300
35240 FORI=0TOZD-1:PRINT@SS(I,1)-1+SS(1,2)*64,I;" ";FE$(I);":";A$(I);:NEXT
35250 PRINT@0,STRING$(63,32);
35260 PRINT@0,"Reihenfolge f";CHR$(125);"r Brief(n/n/): ";:LINK(0,9,383,9,1):LINE INPUT SU$
35270 IFSU$=""THENZE=ZP:RETURN
35280 OPEN"O",#1,"ADRESSE/DAT"
35281 Z1=1
35290 FORI=1TOLEN(SU$)
35300 M$=MID$(SU$,I,1)
35310 IFM$="/"THENA$=MID$(SU$,Z1,I-1):Z1=I+1
35311 IFM$<>"/"THEN35340
35320 NU=VAL(A$)
35330 PRINT#1,A$(NU)
35340 NEXT
35350 CLOSE#1
35360 ZE=ZP
35370 RETURN
```

FMODULSE

```
32140 GOSUB170
32150 X=50:Y=20:X1=300:Y1=140:GOSUB3310
32160 PRINT@211,"Schutzentfernung V 1.0";
32170 X=98:Y=32:X1=254:Y1=50:GOSUB3540
32180 PRINT@458,"Schutzdatei  : ";
32190 PRINT@586,"Originaldatei:";
32200 LINK(150,92,221,92,1)
32210 LINK(150,116,221,116,1)
32220 PRINT@473,"";:LINEINPUTDA$
32230 PRINT@601,"";:LINEINPUTSD$
32240 IFSD$=DA$THEN32140
32250 PRINT@458,STRING$(30,32);
32260 PRINT@586,STRING$(30,32);
32270 OPEN"R",#1,DA$
32280 LA=LOF(1)
32290 CLOSE#1
32300 LA=LA*245:B$=CHR$(28)+CHR$(31)+STRING$(2,11)
32310 PRINT@458,"0";
32320 PRINT@466,"50";
32330 PRINT@474,"100";:PRINT@482,"150";:PRINT@490,"200 %";
32340 FORI=63TO263STEP10:#LINE(I,90)-(I,98),POINT:NEXT
32341 X=56:Y=74:X1=294:Y1=128:GOSUB3540
32350 OPEN"I",#1,DA$:OPEN"O",#2,SD$
32360 LINEINPUT#1,A$
32370 LE=LE+LEN(A$)
32380 PR=(LE*100)/LA
32390 IFP1=PRTHEN32410
32400 LINK(63+PR,100,64+PR,110,1)
32410 P1=PR
32420 CO=INSTR(A$,B$):IF CO>0 THEN A$=MID$(A$,1,CO-3)
32430 PRINT#2,A$
32440 LI=LI+LEN(A$)
32450 P2=(LI*100)/LA:IFP2=P3THEN32470
32460 FORL2=112TO122STEP2:LINK(53+P2,L2,53+P2,L2,1):NEXT
32470 P3=P2
32480 IFNOTEOF(1)THEN32360
32490 P2=0:P3=0:SD$="":DA$="":LA=0:LE=0:PR=0:L2=0
32500 CLOSE:GOTO41000
```

FMODULBR

```
32140 CLS:CLH:X=0:Y=0:ZQ=0
32150 DEFINTA-Z
32160 CLS
32170 DIMZE$(70),B1$(50),C$(50)
32180 Z=0:ZM=0
32190 FORI=0TO70:ZE$(I)=STRING$(64,32):NEXT
32191 PRINT:PRINT:PRINT"<B>rief oder <T>ext erstellen ?";
32192 T$=INKEY$:IF T$=""THEN 32192
32193 IF T$="T" OR T$="t" THEN 32200
32199 GOSUB33670
32200 IFZ<0THENZ=0
32220 FORI=0TO14:PRINT@I*64,ZE$(I+Z);:NEXT
32230 ZE=PEEK(15360+X+Y*64):PRINT@X+Y*64,">";
32240 PRINT@960,"X: ";X;:PRINT@970,"Y: ";Y+Z;
32250 T$=INKEY$
32260 PT=PEEK(14440)
32270 IFY+Z>ZMTHENZM=Z+Y
32280 IFPT=96THEN33630
32290 IFT$=""THEN32250
32300 POKE15360+X+Y*64,ZE
32310 IFT$=CHR$(10)THENY=-NOTY
32320 IFT$=CHR$(13)THENY=-NOTY:X=0
32330 IFT$=CHR$(91)THENY=NOT-Y
32340 IFT$=CHR$(8)THENX=NOT-X
32350 IFT$=CHR$(9)THENX=-NOTX
32360 IFX>63THENX=0:Y=-NOTY
32370 IFX<0THENX=63:Y=NOT-Y
32380 IFY=15THENY=14:Z=-NOTZ:GOTO32200
32390 IFY=-1THENY=0:Z=NOT-Z:GOTO32200
32400 IFT$=CHR$(91)THEN32420
32410 IFT$>CHR$(31)THENIFT$<CHR$(125)THENPRINT@X+Y*64,T$;:MID$(ZE$(Z+Y),X+1,1)=T$:X=-NOTX
32420 IFT$=CHR$(31)THENX=0:Y=0
32430 IFT$=CHR$(25)THENZE$(Z+Y)=STRING$(64,32):GOTO32200
```

```
32440 IFT$=CHR$(27)THEN32480
32450 IFT$=CHR$(26)THEN32570
32460 IFT$=CHR$(24)THENGOTO32680
32470 GOTO32230
32480 PRINT@980,STRING$(20," ");
32490 PRINT@980,"Speicher Datei: ";:LINEINPUTDA$
32500 PRINT@980,"                         ";
32510 IFDA$=""THEN32560
32520 OPEN"O",#1,DA$
32530 PRINT#1,ZM
32540 FORI=0TOZM:PRINT#1,ZE$(I):NEXT
32550 CLOSE#1
32560 GOTO32230
32570 PRINT@980,STRING$(20," ");
32580 PRINT@980,"Lade Datei: ";:LINEINPUTDA$
32590 PRINT@980,"                            ";
32600 IFDA$=""THEN32670
32610 OPEN"I",#1,DA$
32620 INPUT#1,AZ:Z=0
32630 ZE$(Z)="":LINEINPUT#1,ZE$(Z)
32640 Z=-NOTZ
32650 IFZ<AZTHEN32630
32660 CLOSE#1
32670 ZM=Z:GOTO32220
32680 GOTO33060
32690 Z2=1:ZZ=1
32700 CC=INSTR(1,A$,"  ")
32710 IFCC>0THENA$=MID$(A$,1,CC-1)
32720 FORI=1TOLEN(A$)
32730 IFMID$(A$,I,1)=" "THENB1$(ZZ)=" ":C$(ZZ)=MID$(A$,Z2,I-Z2):Z2=I+1:ZZ=-NOTZZ
32740 NEXT
32750 C$(ZZ)=MID$(A$,Z2,LEN(A$))
32760 RETURN
32770 :
32780 LZ=0:LL=0:LS=0:LG=0:AL=0:FORI=1TOZZ
32790 W$=C$(I):FORL=1TOLEN(W$):IFMID$(W$,L,1)<CHR$(32)THENLQ=LQ+1
32800 NEXT
32810 LG=LG+LEN(C$(I))
32820 NEXT
32830 LZ=ZZ-1:ML=ML+LQ+(LQ)/2:LQ=0:LL=ML-LG
32840 IFLZ<1THENLZ=1
32850 AL=INT(LL/LZ)
32860 FORI=1TOZZ-1:B1$(I)=STRING$(AL," "):NEXT
32870 LG=0:FORI=1TOZZ:LG=LG+LEN(B1$(I))+LEN(C$(I)):NEXT
32880 IFLG<MLTHEN32890ELSE32910
32890 DI=ML-LG
32900 FORI=1TODI:B1$(I)=B1$(I)+" ":NEXT
32910 D$=""
32920 FORI=1TOZZ
32930   D$=D$+C$(I)+B1$(I):NEXT
32940 RETURN
32950 :
32960 LS=LEN(A$)
32970 HM=INT(ML/2):HS=INT(LS/2)
32980 D$=STRING$(64," "):MID$(D$,HM-HS,LEN(A$))=A$
32990 RETURN
33000 D$=STRING$(ML," ")
33010 LSETD$=A$
33020 RETURN
33030 D$=STRING$(ML," ")
33040 RSETD$=A$
33050 RETURN
33060 CLS
33070 FORK=0TOZM
33080 A$=ZE$(K)
33090 C0=INSTR(1,A$,"@BL")
33100 C1=INSTR(1,A$,"@ZE")
33110 C2=INSTR(1,A$,"@LB")
33120 C3=INSTR(1,A$,"@RB")
33130 C4=INSTR(1,A$,"@NO")
33140 IFC0>0THENA$=MID$(A$,4,LEN(A$)):OK=1
33150 IFC1>0THENA$=MID$(A$,4,LEN(A$)):OK=2
33160 IFC2>0THENA$=MID$(A$,4,LEN(A$)):OK=3
33170 IFC3>0THENA$=MID$(A$,4,LEN(A$)):OK=4
33180 IFC4>0THENA$=MID$(A$,4,LEN(A$)):OK=0
33190 ML=64
33200 GOSUB33410
33210 GOSUB32690
33220 IFOK=1THENGOSUB32770
33230 IFOK=2THENGOSUB32960
33240 IFOK=3THENGOSUB33000
33250 IFOK=4THENGOSUB33030
33260 IFOK=0THEND$=A$
33270 ZE$(K)=D$
33280 NEXT
33290 PRINT@0,STRING$(64,"*");
33300 A$="DRUCKEN DES BRIEFES":GOSUB32960
33310 PRINT@64,D$;
33320 PRINT@128,STRING$(64,"*");
33330 PRINT:PRINT
33340 PRINT"Soll dieser Brief gedruckt werden (J/N)? ";:
33350 T$=INKEY$:IFT$=""THEN33350
33360 IFT$="N"ORT$="n"THENGOTO33430
33361 LPRINTCHR$(27);"A";CHR$(8);CHR$(27);"2";
33370 FORI=0TOZM
33380 D$=ZE$(I)
33381 GOSUB34000
33400 NEXT:GOTO41000
33410 C0=INSTR(1,A$,"@FE")
33420 C1=INSTR(1,A$,"@FA")
33430 C2=INSTR(1,A$,"@UE")
33440 C3=INSTR(1,A$,"@UA")
33450 C4=INSTR(1,A$,"@BE")
33460 C5=INSTR(1,A$,"@BA")
33470 C6=INSTR(1,A$,"@HE")
33480 C7=INSTR(1,A$,"@HA")
33490 C8=INSTR(1,A$,"@TE")
33500 C9=INSTR(1,A$,"@TA")
33510 IFC0>0THENMID$(A$,C0,3)=CHR$(27)+"E"+CHR$(0)
33520 IFC1>0THENMID$(A$,C1,3)=CHR$(27)+"F"+CHR$(0)
33530 IFC4>0THENMID$(A$,C4,3)=CHR$(14)+CHR$(0)+CHR$(0)
33540 IFC5>0THENMID$(A$,C5,3)=CHR$(20)+CHR$(0)+CHR$(0)
33550 IFC2>0THENMID$(A$,C2,3)=CHR$(27)+"-"+CHR$(1)
33560 IFC3>0THENMID$(A$,C3,3)=CHR$(27)+"-"+CHR$(0)
33570 IFC6>0THENMID$(A$,C6,3)=CHR$(27)+"S"+CHR$(0)
33580 IFC7>0THENMID$(A$,C7,3)=CHR$(27)+"T"+CHR$(0)
33590 IFC8>0THENMID$(A$,C8,3)=CHR$(27)+"S"+CHR$(1)
33600 IFC9>0THENMID$(A$,C9,3)=CHR$(27)+"T"+CHR$(0)
33610 RETURN
33620 GOTO32220
33630 FORI=Y+ZTOZM
33640 ZE$(I)=ZE$(I+1)
33650 NEXT
33660 ZM=I:ZE$(I)=STRING$(64,32):GOTO32220
33670 OPEN"I",#1,"ABSEND/DAT"
33680 LINEINPUT#1,AN$:LINEINPUT#1,OT$:LINEINPUT#1,ST$:LINEINPUT#1,WO$:LINEINPUT#1,TE$
33690 CLOSE#1
33700 OPEN"I",#1,"ADRESSE/DAT"
33710 LINEINPUT#1,NA$
33720 LINEINPUT#1,NN$
33730 C0=INSTR(1,NA$,"  ")
33740 IFC0>0THENNA$=MID$(NA$,1,C0-1)
33750 LINEINPUT#1,SE$
33760 LINEINPUT#1,WT$
33770 LINEINPUT#1,TT$
33780 CLOSE#1
33790 MID$(ZE$(0),1,LEN(AN$))=AN$
33800 MID$(ZE$(0),40,LEN(OT$))=OT$
33810 MID$(ZE$(1),1,LEN(ST$))=ST$
33820 MID$(ZE$(2),1,LEN(WO$))=WO$
33830 MID$(ZE$(3),1,LEN(TE$))=TE$
33840 MID$(ZE$(9),1,16)="Herrn/Frau/Firma"
33850 MID$(ZE$(10),1,LEN(NA$))=NA$:MID$(ZE$(10),LEN(NA$)+2,LEN(NN$))=NN$
33860 MID$(ZE$(11),1,LEN(SE$))=SE$
33870 MID$(ZE$(13),1,LEN(WT$))=WT$:MID$(ZE$(13),6,LEN(TT$))=TT$:ZM=14:Y=14
33880 RETURN
33900 :
34000 A$=D$:CLS
34001 C0=INSTR(1,A$,"@LM")
34010 C1=INSTR(1,A$,"@LQ")
34020 C2=INSTR(1,A$,"@CS")
34030 C3=INSTR(1,A$,"@MI")
34040 C4=INSTR(1,A$,"@CM")
34050 C5=INSTR(1,A$,"@P1")
34060 C6=INSTR(1,A$,"@M1")
34070 C7=INSTR(1,A$,"@01")
34080 C8=INSTR(1,A$,"@OE")
34090 C9=INSTR(1,A$,"@FS")
34100 D0=INSTR(1,A$,"@SP")
34110 D1=INSTR(1,A$,"@MC")
34120 D2=INSTR(1,A$,"@IP")
34130 D3=INSTR(1,A$,"@TR")
34140 D4=INSTR(1,A$,"@ST")
34150 D5=INSTR(1,A$,"@AN")
34160 D6=INSTR(1,A$,"@BA")
34170 D7=INSTR(1,A$,"@BB")
34180 D8=INSTR(1,A$,"@BO")
34190 D9=INSTR(1,A$,"@BW")
34200 E0=INSTR(1,A$,"@FF")
34210 E1=INSTR(1,A$,"@IS")
34220 E2=INSTR(1,A$,"@IH")
34230 E3=INSTR(1,A$,"@M2")
34240 E4=INSTR(1,A$,"@SC")
34250 E5=INSTR(1,A$,"@SH")
34260 E6=INSTR(1,A$,"@SC")
34270 E7=INSTR(1,A$,"@GR")
34280 E8=INSTR(1,A$,"@LE")
34290 E9=INSTR(1,A$,"@HS")
34291 F0=INSTR(1,A$,"@IE")
34292 F1=INSTR(1,A$,"@IA")
34300 LA=LEN(D$)
34320 IFC0>0THENF$="LMES/FON"
```

```
34330 IFC1>0THENF$="LQ/FON"
[illegible]
34440 IFD2>0THENF$="IPL/FON"
34450 IFD3>0THENF$="TR"
34460 IFD4>0THENF$="STR"
34470 IFD5>0THENF$="ANT"
34480 IFD6>0THENF$="BALL"
34490 IFD7>0THENF$="BB"
34500 IFD8>0THENF$="BO"
34510 IFD9>0THENF$="BW"
34520 IFE0>0THENF$="FF"
34530 IFE1>0THENF$="ISE"
34540 IFE2>0THENF$="ISHADED"
34550 IFE3>0THENF$="MB2"
34560 IFE4>0THENF$="SCOMP"
34570 IFE5>0THENF$="SHADOW"
34580 IFE6>0THENF$="STENCIL"
34590 IFE7>0THENF$="GR"
34600 IFE8>0THENF$="LED"
34610 IFE9>0THENF$="SCRIPT"
34611 IF F0>0 THEN IN=1
34612 IFF1>0THEN IN=0
34620 IFF$=""THENLPRINTD$:RETURN
34621 IFE$=F$THENFI=1ELSEFI=0
34630 ST=1:ONERRORGOTO35500
34631 IFFI=1THEN34670
34640 PRINT"*";:OPEN"I",1,F$
34650 LINEINPUT#1,W$:B=ASC(MID$(W$,2,1)):Z=ASC(MID$(W$,1,1))/B
34655 FI=1
34656 E$=F$
34662 CLOSE
34670 T$=D$
34671 PRINT@128,"Zeile: ";ZQ;:ZQ=ZQ+1:IFT$=""THENRETURN
34672 CO=INSTR(1,T$,"@"):IFCO>0THENT$=MID$(T$,4,LEN(T$))
34675 IFT$=""THENT$=" "
34680 LA=LEN(T$)*B:LO=INT(LA/256):HI=ABS(INT(LA-(LO*256)))
34690 OPEN"R",1,F$,"FF",B*Z
34700 FORK=0TOZ-1
34710 LPRINTCHR$(27);"Y";:OUT253,HI:OUT253,LO
34720 FORN=1TOLEN(T$)
34730 A=ASC(MID$(T$,N))-31:GOSUB35000:NEXT:LPRINTCHR$(10);:NEXT
34740 CLOSE:ONERRORGOTO35500:RETURN
35000 IFA=1THENFORP=1TOB:OUT253,0:NEXT:RETURN
35010 GET1,A,,(B*Z)A$;
35020 FORQ=1TOB:A=ASC(MID$(A$,Q+K*B))
35030 IF IN=1 THEN A=255-A
35031 OUT253,A:NEXT
35040 RETURN
35500 END
36000 :
36500 END
38000 :
```

PMODULTH

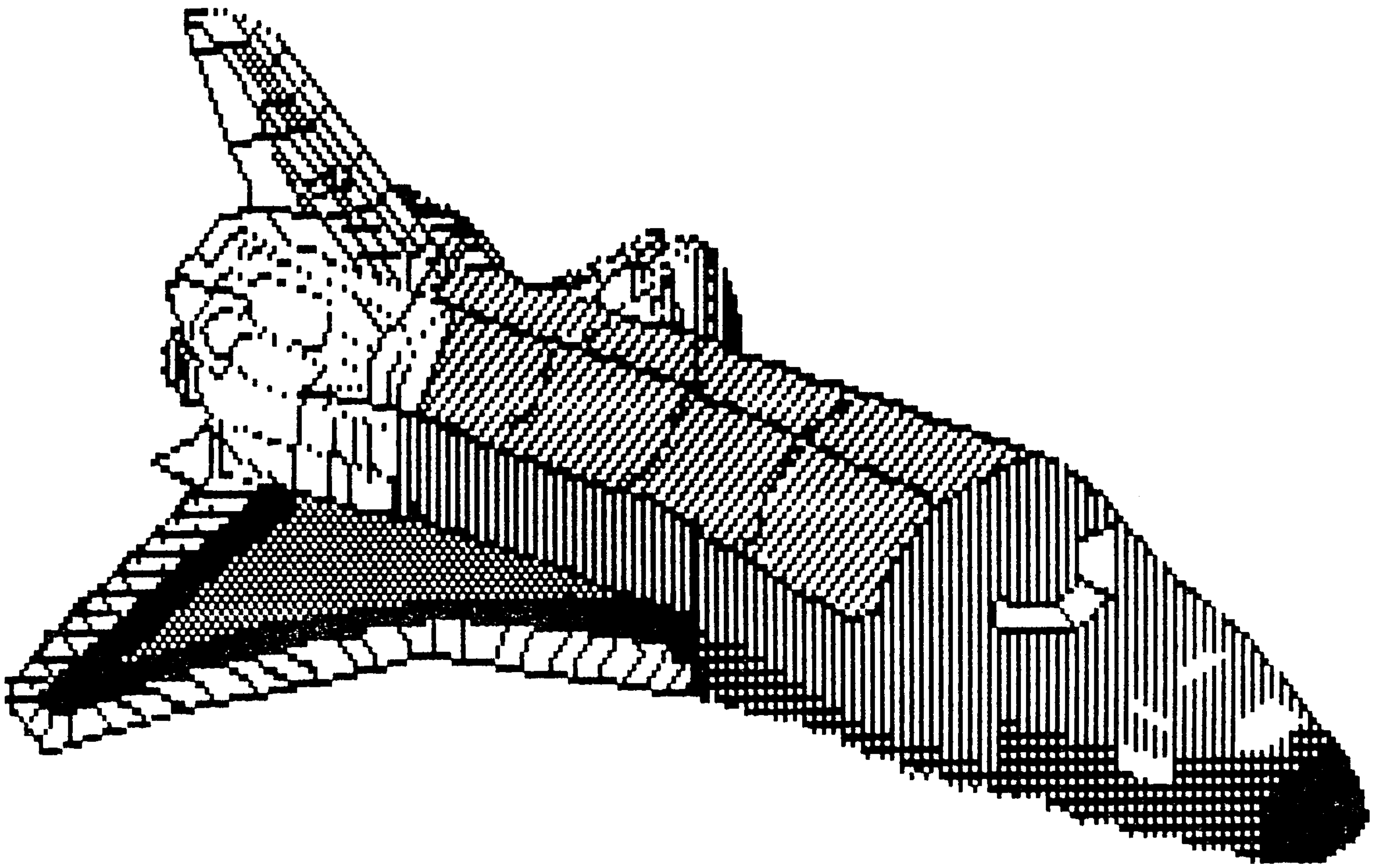
```
32000 :
32140 CLH
32150 CLS
32160 PRINT@20,"Hilfstext Nr. 1"
32170 PRINT
32180 PRINT" Tasten f";CHR$(125);"r Pfeilsteuerung:"
32190 PRINT
32200 PRINT" Pfeil links   -> Pfeil links Taste"
32210 PRINT" Pfeil rechts  -> Pfeil rechts Taste"
32220 PRINT" Pfeil runter  -> Pfeil runter Taste"
32230 PRINT" Pfeil rauf    -> Pfeil rauf Taste"
32240 PRINT" Men";CHR$(125);"punkt anw";CHR$(123);"hlen  -> Leertaste"
32250 PRINT" Eingabe abschlie";CHR$(126);"en  -> NEW LINE/ENTER"
32260 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
32270 T2=PEEK(14440):IF T2=0THEN 32270
32280 IF T2=128 THEN 36000
32290 GOSUB 35500
32300 PRINT@20,"Hilfstext Nr. 2"
32310 PRINT
32320 PRINT" TISCH:"
32330 PRINT
32340 PRINT "Nach anw";CHR$(123);"hlen dieses Balkenpunktes, k!nnen Sie"
32350 PRINT "zwischen den foldenden Men)punkten w";CHR$(123);"hlen:"
32360 PRINT "UHR"
32370 PRINT "NOTIZ"
32380 PRINT "DRUCK"
32390 PRINT "COPY"
32400 PRINT "HILFE 1"
32410 PRINT
32420 PRINT "Mit dem Pfeil k!nnen Sie nun einen dieser Punkte anfahren"
32430 PRINT "und anschlie";CHR$(126);"end mit der Leertaste anw";CHR$(123);"hlen."
32440 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
32450 T2=PEEK(14440):IF T2=0THEN 32450
32460 IF T2=128 THEN 36000
32461 IF T2=32 THEN 32150
32470 GOSUB 35500
32480 PRINT@20,"Hilfstext Nr. 3"
32490 PRINT
32500 PRINT" UHR:"
32510 PRINT
32520 PRINT" Mit diesem Men)punkt k!nnen Sie die interne Uhr Ihres"
32530 PRINT" Computer stellen und auch bestimmen, ob sie in der"
32540 PRINT" rechten unteren Bildschirmecke angezeigt werden soll."
32550 PRINT" Ist dieses der Fall, so l";CHR$(123);"uft diese Uhr"
32560 PRINT" immer in der rechten unteren Ecke. Lediglich bei den"
32570 PRINT" Werkzeugen wie z. B. GECAD oder dem EDITOR wird die"
32580 PRINT" Zeit nicht angezeigt."
32590 PRINT
32600 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
32610 T2=PEEK(14440):IF T2=0THEN 32610
32620 IF T2=128 THEN 36000
32621 IF T2=32 THEN 32290
32630 GOSUB 35500
32640 PRINT@20,"Hilfstext Nr. 4"
32650 PRINT
32660 PRINT" NOTIZ:"
32670 PRINT
32680 PRINT" Wenn Sie diesen Men)punkt anw";CHR$(123);"hlen, so haben"
32690 PRINT" Sie die M!glichkeit, eine Notiz zu erstellen, eine alte zu"
```

```
32700 PRINT" l!schen oder eine alte zu editieren. Die Notiz wird auf"
32710 PRINT" Dauer gespeichert und ist bei jedem GEGRAS Aufruf verf)gbar."
32720 PRINT" In dem Notizfenster kann der Cursor mit den Pfeiltasten be-"
32730 PRINT" wegt werden."
32740 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
32750 T2=PEEK(14440):IF T2=0THEN 32750
32751 IF T2=32 THEN 32470
32760 CLS
32770 PRINT@20,"Hilfstext Nr. 5"
32780 PRINT
32790 PRINT" DRUCK:"
32800 PRINT
32810 PRINT" Wenn Sie diesen Men)punkt anw";CHR$(123);"hlen, so wird die"
32820 PRINT" Notiz, die sich gegenw";CHR$(123);"rtig im Speicher befindet,"
32830 PRINT" auf dem Drucker )ber die Centronics-Schnittstelle ausge-"
32840 PRINT" geben."
32850 PRINT
32860 PRINT" COPY:
32870 PRINT
32880 PRINT" Damit Sie wissen, da";CHR$(126);" dieses Programm von mir"
32890 PRINT" erstellt wurde, wird mit diesem Men)punkt eine Copyright-"
32900 PRINT" meldung ausgegeben."
32920 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
32930 T2=PEEK(14440):IF T2=0THEN 32930
32940 IF T2=32THEN32630
32950 IFT2=128THEN36000
32960 CLS
32970 PRINT@20,"Hilfstext Nr. 6"
32980 PRINT
32990 PRINT" HILFE 1:"
33000 PRINT
33010 PRINT" Sie erhalten mit diesem Men)punkt diesen Hilfstext !"
33020 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
33030 T2=PEEK(14440):IF T2=0THEN 33030
33040 IF T2=128 THEN 36000
33050 IF T2=32 THEN 32760
33060 CLS
33070 PRINT@20,"Hilfstext Nr. 7"
33080 PRINT
33090 PRINT" WERKZEUGE:
33100 PRINT
33110 PRINT" Hiermit k!nnen Sie verschiedene Zusatzprogramme, die wir von"
33120 PRINT" nun an Werkzeuge nennen wollen, starten. "
33130 PRINT" Zur Zeit stehen Ihnen unter GEGRAS folgende Werkzeuge zur"
33140 PRINT" Verf)gung:"
33150 PRINT
33160 PRINT" GECAD
33170 PRINT" GEKALK
33180 PRINT" EDITOR
33190 PRINT" Programmschutz
33200 PRINT" Programmschutzentferner"
33210 PRINT" RECHNER"
33220 PRINT@970,"  <Pfeiltasten = Weiter>  <Leertaste = Men";CHR$(125);">";
33230 T2=PEEK(14440):IF T2=0THEN 33230
33240 IFT2=32 THEN 32960
33250 IFT2=128 THEN 36000
33260 CLS
33270 PRINT@20,"Hilfstext Nr. 8"
33280 PRINT
33290 PRINT" GECAD:
```

```
33300 PRINT
33310 PRINT" Dieses ist ein (C)omputer (A)ided (D)esign Programm. Sie"
33320 PRINT" k!nnen mit diesem technische Zeichnungen erstellen."
33330 PRINT
33340 PRINT" GEKALK:
33350 PRINT
33360 PRINT" Ist eine Tabellenkalkulation. Ist noch nicht o.k."
33380 PRINT
33390 PRINT" RECHNER:
33400 PRINT
33410 PRINT" Ist ein Taschenrechner, welcher auf dem Bildschirm erscheint."
33420 PRINT" Verlassen: Cursor au";CHR$(126);"erhalb des Fensters, dann SPACE."
33450 PRINT@970," <Pfeiltasten = Weiter> <Leertaste = Men";CHR$(125);">";
33460 T2=PEEK(14440):IFT2=0THEN33460
33470 IFT2=128 THEN 36000
33480 IFT2=32 THEN 33060
33490 CLS
33500 PRINT@20,"Hilfstext Nr. 9"
33510 PRINT
33520 PRINT" Programmschutz:"
33530 PRINT
33540 PRINT" Ein Programm (BASIC), welches als ASCII-Datei vorliegt, wird"
33550 PRINT" mit einem Schutz versehen, welcher das Listen des Programmes"
33560 PRINT" unm!glich macht."
33570 PRINT
33580 PRINT" Programmschutz entfernen:"
33590 PRINT
33600 PRINT" Entfernt des zuvor beschriebenen Schutz aus den Programmen."
33610 PRINT
33620 PRINT" EDITOR:"
33630 PRINT
33640 PRINT" Dient zum Erstellen von ASCII-Dateien."
33650 PRINT@970," <Pfeiltasten = Weiter> <Leertaste = Men";CHR$(125);">";
33660 T2=PEEK(14440):IFT2=0THEN33660
33670 IFT2=32 THEN33260
33680 CLS
33690 PRINT@20,"Hilfstext Nr. 10"
33700 PRINT
33710 PRINT" LAUFWERKE:"
33720 PRINT
33730 PRINT" Dieser Balkenmen)punkt gibt Ihnen die M!glichkeit, auf der"
33740 PRINT" GEGRAS Ebene, DOS Befehle auszuf)hren, allerdings fast ohne"
33750 PRINT" Tastatureingaben wie DIR :0 oder FORMAT :0. Die Funktions-"
33760 PRINT" weise ist dieselbe wie bei den DOS-Befehlen."
33770 PRINT
33780 PRINT" OPTIONEN:"
33790 PRINT
33800 PRINT" Hier k!nnen Sie einstellen, ob der Cursor als Pfeil oder als"
33810 PRINT" Fadenkreuz dargestellt werden soll."
33830 PRINT@970," <Pfeiltasten = Weiter> <Leertaste = Men";CHR$(125);">";
33840 T2=PEEK(14440):IFT2=0THEN33840
33850 IFT2=128 THEN36000
33860 IFT2=32 THEN 33490
33870 CLS
33880 PRINT@20,"Hilfstext Nr. 11"
33890 PRINT
33900 PRINT" DATEIEN:"
33910 PRINT
33920 PRINT" Hiermit k!nnen Sie Dateien umbenennen oder auch Disketten-"
33930 PRINT" namen und -daten ver";CHR$(123);"ndern. "
```



```
33940 PRINT
33950 PRINT" Es k!nnen auch mehrere Dateien ausgew";CHR$(123);"hlt werden."
33960 PRINT" Dazu werden diese einfach 'angeklickt'. Verlassen kann die"
33970 PRINT" Anzeige mit einer Pfeilpositionierung ganz oben auf dem Bild-"
33980 PRINT" schirm."
33990 PRINT
34000 PRINT" ENDE:"
34010 PRINT
34020 PRINT" Hier haben Sie M!glichkeiten, GEGRAS zu verlassen."
34030 PRINT@970," <Pfeiltasten = Weiter> <Leertaste = Men";CHR$(125);">";
34040 T2=PEEK(14440):IFT2=0THEN34040
34050 IFT2=32 THEN 33680
35500 CLS:RETURN
36000 GRY:CLS:GOTO41000
```



GECAD

G E C A D V 2.0

GECAD ist ein Teilprogramm des GEGRAS Systems, welches in diesem Sonderheft vorgestellt wird. Einige Teile wie die Bilder können leider nicht abgedruckt werden, da die Eingabe zu viel Zeit und auch die Länge zu groß wär.

Viele der erhältlichen Programme für diese Rechner benutzen meistens die Blockgrafik mit den 128 x 48 Punkten. Diese Grafikauflösung kann jedoch auf die Dauer nicht befriedigen.

Eine bessere Alternative bietet die HRG1B(A) Karte der Firma

RB-Electronic. Sie stellt immerhin eine Grafikauflösung von 384 x 192 Punkten zur Verfügung. Das sind 73 728 Punkte im Gegensatz zu 6 144 Punkten in der Blockgrafik. Sicher, diese Auflösung kommt lange nicht an die der echten CAD-Systeme mit 1024 x 1024 Punkten oder höher nicht heran, jedoch läßt sich schon einiges mit dieser Auflösung erreichen. Die Software ist hierbei der entscheidende Faktor.

Dieses soll jedoch nicht heißen, daß GECAD(c) das Beste am Markt sei, jedoch hat es einige Vorzüge.

Alle Funktionen sind über die Tastatur erreichbar. Ein Joystick ist empfehlenswert, jedoch nicht Bedingung.

Das Programm ist in BASIC geschrieben und läuft unter dem HRG-PACK(c). Es existiert auch eine ~~SUPERHRG~~-Treiber Version, jedoch mit viel weniger Funktionen. Da BASIC nicht zu den schnellsten Sprachen auf den Computern gehört, ist eine Geschwindigkeitserhöhung der internen Taktrate von 1.77 Mhz auf 3.54 Mhz unabdingbar. Wie dieses geht, stand in einer älteren Club-Info. Auf Wunsch liefere ich auch eine Platine(-vorlage) zum Selbstkostenpreis.

Natürlich läuft das Programm auch mit den normalen 1.77 Mhz, jedoch wird die Geschwindigkeit auf die Dauer doch zu langsam.

Ein Abdruck des Listings folgt nach dieser Anleitung.

Für die Clubmitglieder steht das Programm auf einer 80 Track DD/DS und auf einer 40 Track DD/DS Diskette zur Verfügung.

Für ein vernünftiges Arbeiten empfiehlt sich die 80 Track Disk sowie zwei Laufwerke, wovon eines mit 80 Track und eines mit 40 Track voll ausreicht.

GECAD

1.0 Tastenbelegung

In der Zeichenebene sind folgende Tasten belegt:

1 - 5						
!						
Dollar	Sternchen	(	)	'	Balkenkreuz	
Alpha	Minus	Plus	Gleich	Kleiner als		
;	Größer als		.	,		
W	E	R	T	Y	U	I
O	P	A	S	D	F	G
H	J	K	L	Z	X	C
V	B	N	M			

Wie man sieht eine beachtliche Anzahl von Funktionen. All diese Funktionen sind dann erreichbar, wenn der Cursor auf dem Bildschirm erscheint.

Bei den Buchstabentasten ist es egal, ob diese mit oder ohne SHIFT gedrückt wurden.

1.1 Steuertastenbelegung

Natürlich kann der Cursor auch bewegt werden, da sonst solch ein Programm ja witzlos wäre.

Zur Steuerung stehen die vier Pfeiltasten sowie die SHIFT-Taste zur Verfügung.

Pfeil links	---> Cursor um einen Bildpunkt nach links
Pfeil rechts	---> Cursor um einen Bildpunkt nach rechts
Pfeil runter	---> Cursor um einen Bildpunkt nach unten
Pfeil rauf	---> Cursor um einen Bildpunkt nach oben
SHIFT und	
Pfeil links	---> Cursor um SW Bildpunkte nach links
Pfeil rechts	---> Cursor um SW Bildpunkte nach rechts
Pfeil runter	---> Cursor um SW Bildpunkte nach unten
Pfeil rauf	---> Cursor um SW Bildpunkte nach oben

SW steht hier für die Schrittweite (siehe unter Taste (W)).

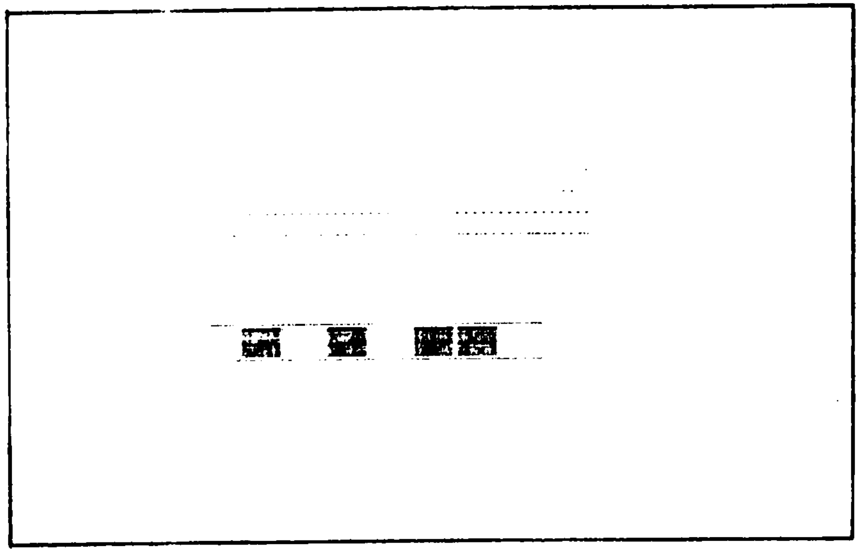
Mit der Leertaste wird eine Funktion gesetzt.

2.0 Die (!)-Funktion

Da unter dem HRG-PACK(c) die Möglichkeit besteht, die Linienform zu ändern, wurde die Möglichkeit auch in GECAD implementiert.

Der Nachteil dieser Funktion ist der, daß noch 10 GRANUEL auf der Diskette in Laufwerk 0 frei sein müssen, da das aktuelle Bild dort zwischen gespeichert wird.

Ist die (!)-Funktion angewählt worden, so wird das aktuelle Bild zwischengespeichert (unter SPEICHERN/GRF) und es erscheint folgendes Bild auf dem Monitor:



Wie zu sehen ist, werden die Linienformen auch gezeigt. Es braucht also nicht "blind" editiert werden.

Mit den Tasten (PLUS) und (MINUS) kann die zu editierende Liniennummer ausgewählt werden.
Ist dieses geschehen, so kann der Cursor mit den Tasten (PFEIL LINKS) und (PFEIL RECHTS) bewegt werden.

Soll ein Punkt gesetzt werden, so wird die Taste (S) gedrückt. Das (S) steht hierbei für SETZEN. Also ist es nur logisch, daß die Taste (L) für LÖSCHEN eines Punktes steht.
Wie die Linie dann aussieht sieht man jederzeit im oberen Bildschirmteil.

Zusätzlich kann man die gesetzen Punkte noch mit der Taste (CLEAR) alle löschen und mit der Taste (I) invertieren.

Ist die Linie oder die Linien fertig, so genügt ein Druck auf die LEERTASTE und das Bild wird wieder eingeladen und kann weiter bearbeitet werden.

2.1 Die (DOLLAR)-Funktion

Mit der (DOLLAR)-Funktion können Sie eine Linie von einem Punkt (X,Y) zu einem anderen Punkt (X1,Y1) zeichnen. Die Besonderheit besteht jedoch darin, daß diese Linie alle Formen, die ihr im Wege stehen umgeht.
D. h. die Linie wird nicht durch ein Objekt gezeichnet, sondern um es herum.

Um dieses auszuprobieren, zeichnet man am Besten erst einmal mit der (B)-Funktion ein Kasten auf dem Bildschirm. Dann positioniert man den Cursor links neben den Kasten (in etwa auf Mitte), drückt dann die (DOLLAR)-Taste und fährt anschließend den Cursor auf die rechte Seite der Box (in etwas mittig). Nun drückt man die (LEERTASTE) und schon wird eine Linie um den Kasten gezeichnet.

Sicher ein interessanter Effekt.

2.2 Die (BALKENKREUZ)-Funktion

Mit dem GEGRAS(c) Programm BIBEDI/CMD, welches auf der GEGRAS(c) Diskette sich befindet, können kleine Teilzeichnungen wie z. B. Stühle, Tische, Fenster, usw. erstellt werden.

Damit diese Module auch in der Zeichnung verwendet werden können, gibt es diese Funktion.

Nachdem der Cursor dort plaziert worden ist, wo sich später das gewünschte Modul befinden soll, wird die (BALKENKREUZ)-Taste gedrückt. Daraufhin wird nach dem Namen des Moduls gefragt. Es kann nun der Name des gewünschten Moduls eingeben werden. Abgeschlossen wird die Eingabe mit der (ENTER)-Taste.

Das Modul wird dann geladen und eingezeichnet.

Als Endung müssen die Module die Dateikennung /BIN haben!

Wird anstelle eines Namens nur die (ENTER)-Taste gedrückt, so gelangt man ohne Laden eines Moduls zurück in den Zeichenmodi.

2.3 Die (PLUS)-Funktion

Oftmals kann die Anzeige der Koordinaten und des Zeichenmodi in der rechten Ecke störren. Auch beeinträchtigt sie die Geschwindigkeit ein bischen.

Mit der (PLUS)-Funktion kann gewählt werden, ob die Anzeige angezeigt werden soll oder nicht.

Nach dem Drücken der (PLUS)-Taste wird gefragt, ob eine Anzeige gewünscht wird oder nicht. Beantwortet wird diese Frage durch Drücken der Tasten (J/j) für JA oder (N/n) für NEIN.

Wurde NEIN gewählt, so erlischt die Anzeige und wird erst dann wieder eingeschaltet, wenn diese Funktion wieder aufgerufen und die Taste (J/j) gedrückt wird.

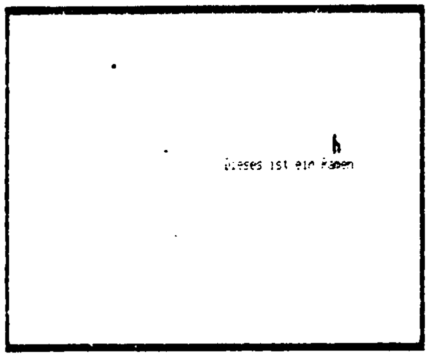
2.4 Die (-)-Funktion

Unter GECAD(c) besteht die Möglichkeit, einen ausgefüllten Rahmen zeichnen zu lassen.

Dieser Rahmen hat jedoch nicht immer dieselbe Breite. Sie läßt sich einstellen.

Nach der Anwahl mit der (-) Taste kann die Breite mit den Tasten (PFEIL LINKS) und (PFEIL RECHTS) verändert werden.

Ist die gewünschte Breite erreicht worden, so wird die Eingabe mit der (LEERTASTE) abgeschlossen.



2.5 Die (=)-Funktion

Wird die (=) Taste gedrückt, so wird der Rahmen mit der zuvor eingestellten Rahmenbreite gezeichnet.

2.6 Die (,)-Funktion

Es wird die Form des Cursors auf die eines großen X verändert.

2.7 Die (<)-Funktion

Es wird die Form des Cursors auf die eines Kreuzes verändert.

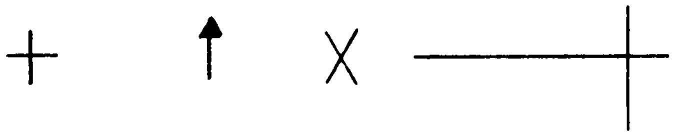
2.8 Die (>)-Funktion

Es wird die Form des Cursors auf die eines Fadenkreuzes verändert.

2.9 Die (.)-Funktion

Es wird die Form des Cursors auf die eines Pfeiles verändert.

Die Cursorformen:

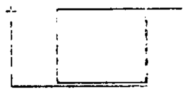


2.10 Die (W)-Funktion

Mit dieser Funktion kann die Schrittweite (SW), mit der der Cursor bei betätigen der SHIFT-Taste in Verbingung mit einer Pfeiltaste bewegt wird, verändert werden.

Nach dem Drücken der (W) Taste wird in der rechten Hälfte des Bildschirmes der aktuelle Schrittweite angezeigt. Diese kann mit der (PFEIL LINKS) und der (PFEIL RECHTS) Taste verändert werden.
Abgeschlossen wird die Eingabe mit der (LEERTASTE).

2.11 Die (E)-Funktion



Sollten die Marker eingeschaltet sein, so kann man sich die späteren Verbindungen anzeigen lassen, ohne daß die Zeichnung dementsprechend verändert wird.

2.12 Die (R)-Funktion

Oftmals muß bei einer Zeichnung auf dem Papier einiges an bestimmten Stellen weg radiert werden. Da man jedoch nicht mit dem Radiergummi etwas von der Bildschirmzeichnung entfernen kann, gibt es die (R)adierfunktion.

Zuerst wird der Cursor auf die linke obere Ecke des zu radierenden rechteckigen Bereichs gefahren. Dann wird die (R) Taste gedrückt und die rechte untere Ecke des Bereichs wird festgelegt. Nun noch die (LEERTASTE) drücken und der zuvor festgelegte Bereich wird gelöscht.

2.13 Die (T)-Funktion

Hiermit kann ein beliebiger Text auf den Bildschirm gebracht werden.
Wurde der Standardzeichensatz ausgewählt, so wird jeder Tastendruck, wie auf einer Schreibmaschine, auf den Bildschirm gebracht. Abgeschlossen wird die Eingabe mit der (ENTER) Taste.

Ist jedoch ein DOTWRITER(c) Zeichensatz gewählt worden, so wird nach dem Text gefragt und, ob ein Zeilenvorschub erfolgen soll. Nach der Eingabe und der Entscheidung, ob mit oder ohne Zeilenvorschub, wird der Text dann in dem entsprechenden Zeichensatz auf dem Bildschirm dargestellt.

Die Standardeinstellung ist der Zeichensatz LET1/CHR!

2.14 Die (Y)-Funktion

Hiermit kann eingestellt werden, ob Marker erlaubt sind oder nicht.

Wird die (E/e) Taste gedrückt, so sind sie erlaubt. Beim Drücken der (A/a) Taste sind sie nicht erlaubt.

Als Marker werden hier Punkte bezeichnet, dessen, wenn Marker erlaubt sind, Position jeweils beim Drücken der (LEERTASTE) gespichert wird.
Später können diese Punkte dann verbunden werden (durch Linien).

Es sind bis zu 100 Punkte(Marker) erlaubt. Die Linientypen sind auch hierfür geeignet.

2.15 Die (U)-Funktion

Diese Funktion erlaubt das Ausführen von DOS-Befehlen sowie das Laden, Speichern und Mischen von Zeichnungen.

Nach dem Drücken der (U) Taste erscheint auf der rechten Bildschirmhälfte das Wort "Befehl:" zusammen mit dem Textcursor. Das Programm erwartet jetzt eine Eingabe. Wird

einfach nur die (ENTER) Taste gedrückt, so gelangt man zurück in die Zeichenebene.

Wird der Name einer Zeichnung, z. B. SPACE/BLD:0 (ENTER) eingeben, so kann man mit den Tasten (L/l,S/s,M/m) wählen, ob das Bild in den Speicher geladen wird, ob es aus dem Speicher auf Diskette gespeichert wird, oder ob das im Speicher befindliche Bild mit dem angegebenen von der Diskette im Speicher gemischt wird.

Sollte die (ENTER) Taste gedrückt werden, so gelangt man in die Zeichenebene ohne eine Wirkung.

Anstelle des Namen kann jedoch auch ein Befehl eingegeben werden. An Befehlen stehen zur Verfügung:

```
%0 DIR 0          &1 FORMAT 1          !0 FREE 0
%1 DIR 1          &2 FORMAT 2          !1 FREE 1
%2 DIR 2          &3 FORMAT 3          !2 FREE 2
%3 DIR 3                               !3 FREE 3
                                       !! FREE
01 COPY 0,1,,NFMT,CBF,SUR
10 COPY 1,0,,NFMT,CBF,USR


! -->> Balkenkreuz


% -->> Sternchen
& -->> Dollar
```

2.16 Die (I)-Funktion

Invertiert einen bestimmten Bereich des Bildschirmes.

Als erstes wird der Cursor auf die linke obere Ecke des zu invertierenden Bereichs gefahren, dann wird die (I) Taste gedrück und es wird die rechte untere Ecke des Bereichs festgelegt. Dann wird die (LEERTASTE) gedrückt und der festgelegte Bereich wird invertiert, d. h. alle gesetzten Punkte werden gelöscht und alle nicht gesetzten Punkte werden gesetzt.

2.17 Die (O)-Funktion

Alle Marker werden gelöscht und der Markerzähler wird auf Null gesetzt.

2.18 Die (P)-Funktion

Nach dem Drücken dieser Taste wird eine Spraydose simuliert.

In einem Umkreis von 10 x 10 Punkte werden zufällig Punkte gesetzt.

Verlassen werden kann dieser Modus mit der (LEERTASTE).

2.19 Die (A)-Funktion

Es kann durch den Aufruf dieser Funktion die Zeichnung auf einem EPSON(r) kompatibelen Drucker ausgegeben werden. Dieses kann in normaler Größe (1:1) oder in doppelter Größe (2:1) geschehen.

Wird die Taste (N/n) gedrückt, so wird die Zeichnung normal ausgedruckt. Beim Drücken der Taste (G/g) wird die Zeichnung doppelt so groß ausgedruckt.

Es wird kein DRUCKER-RESET an den Drucker gesendet, da manche Drucker diesen Code nicht akzeptieren.

2.20 Die (S)-Funktion

Mit der (S)-Funktion hat man die Möglichkeit, einen bestimmten rechteckigen Bereich der Zeichnung zu spiegeln.

Dieses kann entweder über die X-Achse oder aber auch über die Y-Achse geschehen.

Als erstes wird die linke obere Ecke des zu spiegelnden Bereichs festgelegt. Dann wird die (S) Taste gedrückt und anschließend wird die rechte untere Ecke des Bereichs festgelegt. Nun wird die (LEERTASTE) kurz gedrückt und es wird der Cursor auf den Bereich gefahren, in welchem das Spiegelbild abgelegt werden soll. Diese Positionierung wird dann mit der (LEERTASTE) abgeschlossen.

Nun kann mit den Tasten (1 und 3) gewählt werden, ob der Bereich horizontal oder vertikal gespiegelt wird.

TASTE 1 -> Y-Achse
TASTE 3 -> X-Achse

2.21 Die (D)-Funktion

Die (D)-Funktion ist dazu da, ein Dreieck zeichne zu können.

Dort, wo sich der Cursor im Moment befindet, ist die erste Ecke des Dreiecks. Nun wird die (D) Taste gedrückt und es wird die zweite Ecke des Dreiecks festgelegt. Zum guten Schluß noch die ditte Ecke festlegen.

Die Ecken werden übrigens mit der (LEERTASTE) festgelegt.

2.22 Die (F)-Funktion

Verzerren eines bestimmten Bildschirmbereiches ist mit dieser Funktion möglich.

Es wird dabei nach SINUS verzerrt.

Die aktuelle Cursorposition ist die linke obere Ecke des zu verzerrenden Bereichs. Wenn die Position o. k. ist, dann wird die (F) Taste gedrückt und die rechte untere Ecke festgelegt. Dann wird die (LEERTASTE) gedrückt.

Jetzt wird der Cursor auf die linke obere Ecke des Bereichs gefahren, wo sich später der verzerrte Bereich befinen soll. Nun noch die (LEERTASTE) drücken.

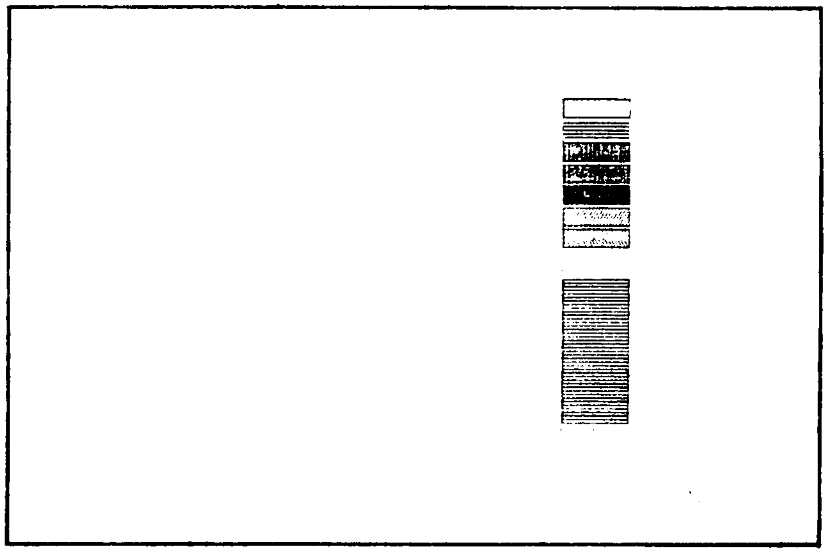
2.23 Die (G)-Funktion

Dient zum Füllen von Bereichen. Gefüllt wird von dort an, wo der Cursor steht.

Achtung!!! Sollte der zu füllende Bereich nicht vollkommen geschlossen sein, so wird der gesamte Bildschirm gefüllt !!!

Ist der Vollbildschirm gewählt worden, so fragt das Programm nach dem Drücken der (G) Taste nach der Nummer (0-6) des Musters, mit dem gefüllt werden soll. Hierzu braucht einfach nur eine Taste von 0 bis 6 gedrückt werden. Der Bereich wird danach mit dem gewählten Muster gefüllt.

Ist kein Vollbildschirm gewählt worden, so erscheint auf der rechten Seite:



GECAD

Nun kann mit den (PFEIL RAUF) und (PFEIL RUNTER) Tasten der Mustercursor gesteuert werden. In dem unteren Block sieht man, wie das Muster etwas größer aussieht.

Ist das passende Muster gefunden worden, so genügt ein Druck auf die (LEERTASTE) und es wird dort, wo sich der Cursor befindet in dem gewählten Muster gefüllt.

2.24 Die (H)-Funktion

Oftmals kann es vorkommen, daß nicht ein voller Kreis bzw. eine volle Ellipse gezeichnet werden soll, sondern nur ein Halbkreis oder ein n-ter Kreis.

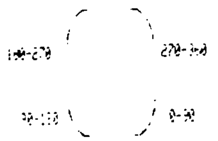
Hier hilft diese Funktion weiter.

Dort, wo sicher der Cursor befindet, ist der Mittelpunkt des Kreises. Nun wird die die (H) Taste gedrückt und es wird dann

mit den Cursortasten der X- und Y-Radius gewählt. Dann wird die (LEERTASTE) gedrückt.

Es erscheint dann in der rechten Ecke die Anzeige WA und WE. WA steht für den Anfangswinkel und WE für den Endwinkel. Diese Werte können nun mit den Pfeiltasten (+SHIFT= größere Schrittweite) geändert werden. Stehen die Werte für WA und WE schon fest, so kann einfach die (ENTER) Taste gedrückt werden. Die Werte können dann per Tastatur eingeben werden. Abgeschlossen werden die Eingaben mit der (ENTER) Taste.

Sind die gewählten Werte o. k. dann einfach die (LEERTASTE) drücken und der Kreisausschnitt wird gezeichnet.



2.25 Die (J)-Funktion

Sind alle Marker gekennzeichnet (LEERTASTE), so können diese durch Drücken der (J) Taste gezeichnet werden.

Voraussetzung dafür ist allerdings, daß die Marker mit der (Y)-Funktion eingeschaltet sind.
Der Markerzähler wird dabei nicht zurück gesetzt !!!

GECAD

2.26 Die (K)-Funktion

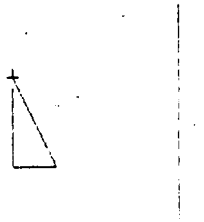
Sollen mehrere Linienzüge nacheinander gezeichnet werden, so ist dieses mit der (L)-Funktion sehr umständlich.

Besser geeignet dafür ist die (K)-Funktion, mit der Linien aneinander gezeichnet werden können.

Ausgangsposition hierfür ist die aktuelle Cursorposition. Wird nun die (K) Taste gedrückt, so kann der Cursor frei bewegt werden. Bei jedem Drücken der (LEERTASTE) wird dann eine Linie von dem zuvor festgelegten Punkt gezeichnet.

Verlassen werden kann dieser Modus mit der (ENTER) Taste.

2.27 Die (L)-Funktion



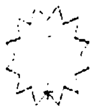
Zeichnet eine Linie von der Ausgangsposition zur Cursorposition.

Dazu wird als erstes der Cursor auf der Ausgangsposition positioniert. Dann wird die (L) Taste gedrückt und anschließend wird der Cursor auf die Endposition gefahren. Jetzt nur noch die (LEERTASTE) drücken. Fertig !

2.28 Die (*)-Funktion

Hiermit lassen sich Zahnrad ähnliche Gebilde (Kreise) zeichnen.

Als erstes wird der Cursor auf den Mittelpunkt des Zahnrades gefahren. Dann wird die (*) Taste gedrückt und anschließend wird mit dem Cursor der Innenkreis (wird später nicht gezeichnet) festgelegt. Es folgt die (LEERTASTE) und das Festlegen des Außenkreises (wird später nicht gezeichnet). Nun noch die (LEERTAST) drücken und das Zahnrad wird gezeichnet.



2.29 Die (Z)-Funktion

Es kann mit dieser Funktion eingestellt werden, ob ein Standardzeichensatz oder ein Dotwriter(r)-Zeichensatz verwendet werden soll.

Nach dem die (Z) Taste gedrückt worden ist, wird nach dem Zeichensatz gefragt.

Mit der (S) Taste wird der STANDARD-Zeichensatz und mit der (D) Taste wird der DOTWRITER(r)-Zeichensatz bestimmt.

Wird die (ENTER) Taste gedrückt, so gelangt man wieder in den Zeichenmodus.

Wurde die (S) oder (D) Taste gedrückt, so wird anschließend nach dem Namen des Zeichensatzes gefragt.

Bei DOTWRITER(r) Zeichensätzen wird der Zeichensatz bei der Textausgabe von der Diskette gelesen. Bei den STANDARD Zeichensätzen wird der Zeichensatz in den Speicher geladen.

2.30 Die (X)-Funktion

Diese Funktion stellt eine Lupe von der Größe 128 x 48 Punkten zur Verfügung. Diese Lupe kann über den ganzen Bildschirm geschoben werden.

Wird die (X)-Taste gedrückt, so erscheint der Cursor zusammen mit einem 128 x 48 Punkte großem Block. Dieser kann nun über die zu vergrößernde Fläche gefahren werden.

Ist die Position o. k., dann wird die (LEERTASTE) gedrückt und der Bereich wird vergrößert.

Es erscheint dann in der linken Ecke ein Cursor, der wie der hochauflösende Cursor, mit den (PFEILTASTEN/SHIFT) bewegt werden kann.

Wird die (LEERTASTE) gedrückt, so wird der Punkt bei der Cursorposition invertiert.

Desweiteren stehen noch folgende Tasten mit einer Funktion zur Verfügung:

(S)	-->	Setzen des Punktes an der Cursorposition
(L)	-->	Löschen des Punktes an der Cursorposition
(C)	-->	Löschen des Bildschirmes
(K)	-->	Invertieren von 0,0 bis Cursorposition
(I)	-->	Laden einer ZOOM-Datei
(O)	-->	Speichern einer ZOOM-Datei
(R)	-->	Radieren
(M)	-->	Linie zeichnen
(H)	-->	Lupenbereich ausdrucken

Mit der (ENTER) Taste wird der Lupenbereich zurück in die Zeichnung geschrieben.

Wird die (H) Taste gedrückt, so erfolgt ein Ausdruck des Bildschirmes, welcher z. B. so aussehen kann:

2.31 Die (C)-Funktion

Hiermit lassen Kreise als auch Ellipsen zeichnen.

Die Cursorposition ist der Mittelpunkt des Kreises bzw. der Ellipse.

Wird dann die (C) Taste gedrückt, dann kann der Radius in X-Richung als auch in Y-Richtung festgelegt werden.

Durch Drücken der (LEERTASTE) wird der Kreis bzw. die Ellipse gezeichnet.

2.32 Die (V)-Funktion

Mit dieser Funktion kann gewählt werden, ob im VOLLBILD- oder im NICHTVOLLBILD-Modus gearbeitet wird.

Wird die (V) Taste gedrückt, so wird gefragt, ob der Vollbildmodus gewählt werden soll oder nicht.

Mit der (J/j) Taste wird der Vollbildmodus eingeschaltet und mit der (N/n) Taste wird der Vollbildmodus ausgeschaltet.

In dem Vollbildmodus stehet dem Benutzer die volle grafische Auflösung der HRGIB(A) zur Verfügung, also 384 x 192 Punkte.

Im Gegensatz zu dieser Auflösung stehen dem Benutzer im Nichtvollbildmodus nur 320 x 192 Punkte zur Verfügung.

Diese Modi wurden gewählt, damit rechts neben dem Arbeitsfeld auch Grafik für Menüs angezeigt werden.

Im Nichtvollbildmodus wird eine Begrenzungslinie vom Programm gezogen, die mit dem Cursor nicht überschritten werden kann.

2.33 Die (B)-Funktion

Ein Kasten kann mit dieser Funktion gezeichnet werden.

Die aktuelle Cursorposition gibt die linke obere Ecke des Kastens an. Wird dann die (B) Taste gedrückt, so kann anschließend die rechte untere Ecke des Kastens festgelegt werden.

Dann wird die (LEERTASTE) gedrückt und es wird ein Kasten gezeichnet.

2.34 Die (N)-Funktion

Mit dieser Taste werden die zwei Hilsseiten auf dem Monitor angezeigt.

Alle Tastenbelegungen werden angezeigt.

2.35 Die (M)-Funktion

In vielen Fällen kann es vorkommen, daß ein bestimmter Bereich einer Zeichnung verschoben werden soll.

Hierzu kann man sicher der (M)-Funktion bedienen.

Die aktuelle Cursorposition stellt die linke obere Ecke des zu verschiebenden Bereichs dar. Ist diese Ecke in Ordnung, so wird die (M) Taste gedrückt und anschließend wird die rechte untere Ecke des zu verschiebenden Bereichs festgelegt.

Ist dieses geschehen, so wird die (LEERTASTE) gedrückt und die linke obere Ecke gewählt, so der festgelegte Block hin kopiert werden soll.

Es gelten hier die Modi 1-4.

2.36 Die (@)-Funktion

Dient zum Löschen des gesamten Bildschirmes.

Achtung !!!! Die gesamte Zeichnung geht verloren !!!!

Nach Drücken der (@) Taste wird gefragt, ob der Bildschirm gelöscht werden soll oder nicht. Dieses kann mit den Tasten (J/j) oder (N/n) beantwortet werden.

2.37 Die (:)-Funktion

Legt fest, ob die Zeichengröße normal oder doppelt so groß sein soll.

Mit der (N/n) Taste wählt man die normale Zeichengröße und mit der (D/d) Taste die doppelte Zeichengröße. Weiteres hierzu in den DOTWRITER(r) und HRGPACK(r) Handbüchern.

2.38 Die (1)-Funktion

Wählt den SETZEN-Modus aus. Die Punkte einer Figur werden gesetzt. Bei der (M)-Funktion bedeutet die (1)-Funktion FLAG 0. Modusanzeige ändert sich auf (1).

2.39 Die (2)-Funktion

Wählt den LÖSCH-Modus aus. Die Punkte einer Figur werden gelöscht. Bei der (M)-Funktion bedeutet die (2)-Funktion FLAG1.

2.40 Die (3)-Funktion

Wählt den XOR-Modus aus. Die Punke einer Figur werden invertiert. Bei der (M)-Funktion bedeutet die (3)-Funktion FLAG2.

GECAD

2.41 Die (4)-Funktion

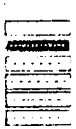
Setzt den Modus 4 für die (M)-Funktion und andere.

Entspricht FLAG 3.

2.42 Die (5)-Funktion

Bei nicht gewähltem Vollbildschirm kann in der rechten Hälfte die Lininart per (PFEIL RAUF) und (PFEIL RUNTER) Tasten eingestellt werden. Abgeschlossen wird diese Wahl mit der (LEERTASTE).

Ist der Vollbildschirm eingeschaltet, so erscheint keine Auswahlgrafik, sondern eine Textabfrage.



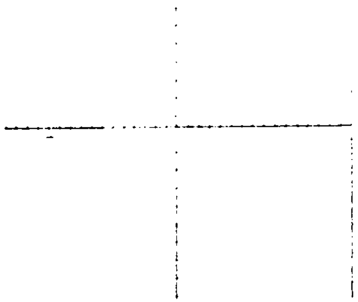
2.43 Die (()-Funktion

Schaltet das eingeblendete Koordinatenkreuz aus.

Die Punkte, wo sich das Kreuz befinden, werden gelöscht.

2.44 Die ())-Funktion

Zeichnet ein Koordinatenkreuz auf dem Bildschirm.



GECAD

2.45 Die (')-Funktion

Ist der Vollbildschirm eingeschaltet, so wird der gesamte Bildschirm invertiert.

2.46 Die (&)-Funktion

Kehrt nach Sicherheitsabfrage zurück zu GEGRAS.

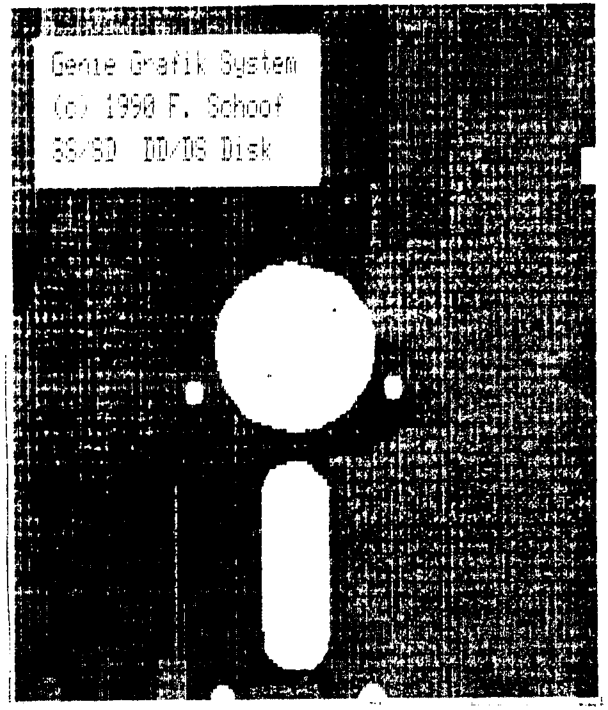
3.1 Verschiedenes

Dieses waren alle Funktione, die GECAD zur Zeit beherrscht. Für Anregungen bin ich jederzeit dankbar.

Es lassen sich übrigens Bilder von dem SUPERHRG(r) und dem GRAPE(r) laden.

Die Druckerausgaben beziehen sich auf EPSON(r) und werden zum größten Teil über den Port 253/FE&H ausgegeben. TRS80(r) Benutzer können dieses leicht ändern.

Nun noch viel Spaß mit GECAD.



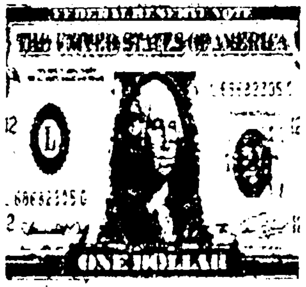
GECAD

(c) 1990

Frank Schoof

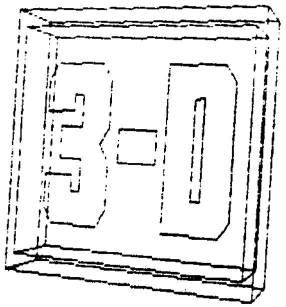
```
10 CLEAR2000
20 GRY
30 GRLE("TITEL1")
40 GRY
50 DIMBZ$(3)
60 FORI=0TO3:BZ$(I)=STRING$(255,32):NEXT
70 GRY
80 ONERRORGOTO3150
90 CLS
100 CMD"load hard/cim"
110 DEFUSR=-15746
120 SELECT(0)
130 LDF("let2/chr")
140 SKIP(0,0)
150 CLH
160 ZS=0
170 DB=6
180 RB=1:DIMXF(100,1),W$(40)
190 X=160:Y=90:KO=0:R1=1:R=R1:MO=1
200 AN=1:MA=0:PU=0
210 PI=4*ATN(1)
220 RM=PI/180
230 SW=10
240 FK=1
250 GOSUB5490
260 GOTO250
270 GOC(320,0,FF):IFVB=0THENIFFF=0THENLINK(320,0,320,191,1)
280 IFFK=0THEN310
290 IFVB=0THENLINK(0,Y,319,Y,-1)ELSELINK(0,Y,383,Y,-1)
300 LINK(X,0,X,191,-1)
310 IFAN=0THEN370
320 PRINT@54,"X: ";X;
330 PRINT@118,"Y: ";Y;
340 PRINT@182,"Modus: ";MO;
350 PRINT@246,"Marker:";MA;
360 PRINT@502,"Pkte:";PU;
370 IFKO=1THENBOX(X1-R,Y1-R1,X1+R,Y1+R1,-1):GOTO430
380 IFKO=2THENLINK(X,Y,X1,Y1,-1):GOTO430
390 IFKO=3THENBOX(X,Y,X1,Y1,-1):GOTO430
400 IFKO=4THENLINK(X,Y,X1,Y1,-1):GOTO430
410 IFKO=5THENBOX(X1-ABS(X-X1),Y1-ABS(Y-Y1),X1+ABS(X-X1),Y1+ABS(Y-Y1),-1):GOTO430
420 IFKO=6THENLINK(X,Y,X1,Y1,-1):LINK(X1,Y1,X2,Y2,-1):LINK(X,Y,X2,Y2,-1):GOTO430
430 IFFK=1THENLINK(X-5,Y,X+5,Y,-1):LINK(X,Y-5,X,Y+5,-1):DOT(X,Y,-1)
440 IFFK=3THENLINK(X,Y,X,Y+8,-1):LINK(X-1,Y+1,X-3,Y+3,-1):LINK(X+1,Y+1,X+3,Y+3,-1)
450 IFFK=2THENLINK(X,Y,X-5,Y-5,-1):LINK(X,Y,X+5,Y-5,-1):LINK(X,Y,X-5,Y+5,-1):LINK(X,Y,X+5,Y+5,-1)
460 PRINT@512,STRING$(40,32);
470 RETURN
480 GOSUB270
490 T$="":T$=INKEY$:IFT$=""THEN490
500 GOSUB270
510 T=ASC(T$)
520 T2=PEEK(14440)
530 IFT2=8THENY=NOT-Y
540 IFT2=16THENY=-NOTY
550 IFT2=32THENX=NOT-Y
560 IFT2=64THENX=-NOTX
570 IFT=27THENY=Y-SW
580 IFT=26THENY=Y+SW
590 IFT=25THENX=X+SW
600 IFT=24THENX=X-SW
610 IFT=40THENX=NOT-X:Y=NOT-Y
620 IFT=72THENX=-NOTX:Y=NOT-Y
630 IFT=48THENX=NOT-X:Y=-NOTY
640 IFT=80THENX=-NOTX:Y=-NOTY
650 IFX<0THENX=0
660 IFVB=0THENIFX>319THENX=319
670 IFVB=1THENIFX>383THENX=382
680 IFY<0THENY=0
690 IFY>191THENY=0
700 :
710 IFT$="L"ORT$="l"THENGOSUB1310
720 IFT$="K"ORT$="k"THENLM=1:GOSUB1310
730 IFT$="c"ORT$="C"THENGOSUB1460
740 IFT$="B"ORT$="b"THENGOSUB1540
750 IFT$="T"ORT$="t"THENGOSUB1700
760 IFT$="R"ORT$="r"THENGOSUB1770
770 IFT$="!"THENGOSUB6970
780 IFT$="S"THENGOSUB7730
790 IFT$="="THENGOSUB4040
800 IFT$="-"THENGOSUB3940
810 IFT$="$"THENGOSUB5150
820 IFT$="I"ORT$="i"THENGOSUB1830
830 IFT$="H"ORT$="h"THENGOSUB4620
840 IFT$="F"ORT$="f"THENGOSUB6190
850 IFT$="d"ORT$="D"THENGOSUB2630
860 IFT$="A"ORT$="a"THENGOSUB3740
870 IFT$="Y"ORT$="y"THENGOSUB4180
880 IFT$="J"ORT$="j"THENGOSUB4230
```

```
890 IFT$="O"ORT$="o"THENPU=0
900 IFT$="#"THENGOSUB4910
910 IFT$="N"ORT$="n"THENGOSUB4380
920 IFT$="E"ORT$="e"THENGOSUB4300
930 IFT$=")"THENGOSUB4140
940 IFT$="("THENGOSUB5530
950 IFT$="'"THENIFVB=1THENINVERT
960 IFT$="+"THENGOSUB3890
970 IFT$="1"THENMO=1
980 IFT$="2"THENMO=2
990 IFT$="3"THENMO=3
1000 IFT$="4"THENMO=4
1010 IFT$="V"ORT$="v"THENGOSUB3830
1020 IFT$=" "THENIFMA=1THENXF(PU,0)=X:XF(PU,1)=Y:PU=PU+1
1030 IFT$="M"ORT$="m"THENGOSUB1890
1040 IFT$="s"ORT$="S"THENGOSUB2010
1050 IFT$="U"ORT$="u"THENGOSUB2260
1060 IFT$="G"ORT$="g"THENGOSUB5570
1070 IFT$="*"THENGOSUB2880
1080 IFT$="X"ORT$="x"THEN3160
1090 IFT$="<"THENFK=1
1100 IFT$=">"THENFK=0
1110 IFT$=","THENFK=2
1120 IFT$="."THENFK=3
1130 IFT$="z"ORT$="Z"THENGOSUB5930
1140 IFT$=":"THENGOSUB6080
1150 IFT$="@"THEN1160ELSE1200
1160 PRINT@566,"Bildschirm";:PRINT@630,"loeschen";:PRINT@694,"(J/N)";
1170 T$=INKEY$:IFT$=""THEN1170
1180 IFT$="J"ORT$="j"THENCLS:CLH:IFVB=0THENLINK(320,0,320,191,1)
1190 GOSUB3720
1200 IFT$="W"ORT$="w"THEN1210ELSE1290
1210 PRINT@566,"Schritt-";:PRINT@630,"weite:";SW;
1220 T$=INKEY$:IFT$=""THEN1220
1230 IFT$=" "THEN1280
1240 T=ASC(T$)
1250 IFT=8THENSW=NOT-SW
1260 IFT=9THENSW=-NOTSW
1270 GOTO1210
1280 GOSUB3720
1290 IFT$="P"ORT$="p"THENGOSUB2530
1300 RETURN
1310 X1=X:Y1=Y:KO=2
1320 GOSUB480
1330 IFAN=0THEN1350
1340 PRINT@310,"X1: ";X1;:PRINT@374,"Y1: ";Y1;
1350 IFT$=" "THEN1380
1360 IFT$=CHR$(13)THENLM=0:GOTO1390
1370 GOTO1320
1380 IFMO=1THENLINK(X,Y,X1,Y1,1)
1390 IFMO=2THENLINK(X,Y,X1,Y1,0)
1400 IFMO=3THENLINK(X,Y,X1,Y1,-1)
1410 IFMO=0THENLINK(X,Y,X1,Y1,LT)
1420 GOTO1440
1430 GOTO1320
1440 KO=0:CLS:IFLM=1THEN1310
1450 RETURN
1460 X1=X:Y1=Y:KO=1
1470 GOSUB490
1480 R=ABS(X-X1):R1=ABS(Y-Y1):IFAN=0THEN1490ELSEPRINT@310,"R1: ";R;:PRINT@374,"RY: ";R1;
1490 IFT$=" "THEN1500ELSE1470
1500 IFMO=1THENPOLYG(X1,Y1,R,R1,4,0,1)
1510 IFMO=2THENPOLYG(X1,Y1,R,R1,4,0,0)
1520 IFMO=3THENPOLYG(X1,Y1,R,R1,4,0,-1)
1530 KO=0:R=0:CLS:RETURN
1540 X1=X:Y1=Y:KO=3
1550 GOSUB490
1560 IFAN=0THEN1580
1570 PRINT@310,"X1: ";X1;:PRINT@374,"Y1: ";Y1;
1580 IFT$=" "THEN1600
1590 GOTO1550
1600 IFMO=1THEN1610ELSE1630
1610 BOX(X,Y,X1,Y1,1)
1620 GOTO1680
1630 IFMO=2THEN1640ELSE1660
1640 BOX(X,Y,X1,Y1,0)
1650 GOTO1680
1660 BOX(X,Y,X1,Y1,-1)
1670 KO=0
1680 KO=0:CLS
1690 RETURN
1700 IFZS=1THEN5830
1710 T$=INKEY$:IFT$=""THEN1710
1720 IFT$=CHR$(13)THENRETURN
1730 SKIP(INT(X/6),INT(Y/12)):PRT(T$)
1740 IFVB=0THENIFX<314THENX=X+DB
1750 IFVB=1THENIFX<377THENX=X+DB
1760 GOTO1710
1770 X1=X:Y1=Y:KO=3
```



```
1780 GOSUB480
1790 IFT$=" "THEN1800ELSE1780
1800 FBOX(X,Y,X1,Y1,0)
1810 KO=0
1820 RETURN
1830 X1=X:Y1=Y:KO=3
1840 GOSUB480
1850 IFT$=" "THEN1860ELSE1840
1860 FBOX(X,Y,X1,Y1,-1)
1870 ST=0:KO=0
1880 RETURN
1890 X1=X:Y1=Y:KO=3
1900 GOSUB480
1910 IFT$=" "THEN1920ELSE1900
1920 X2=X:Y2=Y
1930 KO=0
1940 GOSUB480
1950 IFT$=" "THEN1960ELSE1940
1960 X3=X:Y3=Y
1970 Y4=Y3:X4=X3
1980 COPY(X1,Y1,X2,Y2,X3-X1,Y3-Y1,MO-1)
1990 KO=0
2000 RETURN
2010 X1=X:Y1=Y:KO=3
2020 GOSUB480
2030 IFT$=" "THEN2040ELSE2020
2040 X2=X:Y2=Y
2050 KO=0
2060 GOSUB480
2070 IFT$=" "THEN2080ELSE2060
2080 X3=X:Y3=Y
2090 XZ=X1:YZ=Y1
2100 X=X1:Y=Y1
2110 T$=INKEY$:IFT$=""THEN2110
2120 IFT$=" "THENRETURN
2130 IFT$<"1"ORT$>"4"THEN2110
2140 ST=VAL(T$)
2150 IFY1<Y2THENSU=1
2160 IFY1>Y2THENSU=-1
2170 IFX1>X2THENSI=-1
2180 IFX1<X2THENSI=1
2190 IFST=1ORST=2THEN2200ELSE2220
2200 COPY(X1,Y1,X2,Y1,(X3-X1),(Y3+YZ)-(4*(Y1/2)),1)
2210 Y1=Y1+SU:IFABS(Y1-Y2)>0THEN2200
2220 IFST=3ORST=4THEN2230ELSE2250
2230 COPY(X1,Y1,X1,Y2,(X3+XZ)-(4*(X1/2)),(Y3-YZ),1)
2240 X1=X1+SI:IFABS(X1-X2)<>0THEN2230
2250 RETURN
2260 PRINT@310,"Befehl: ";:PRINT@374,"";:LINEINPUTDA$
2270 IFDA$="M"THENOUT1,1:CLS:GOTO2520
2280 IFLEN(DA$)>2THEN2440
2290 OUT0,0
2300 IFDA$="*0"THENCMD"dir 0":GOTO2260
2310 IFDA$="*1"THENCMD"dir 1":GOTO2260
2320 IFDA$="*2"THENCMD"dir 2":GOTO2260
2330 IFDA$="*3"THENCMD"dir 3":GOTO2260
2340 IFDA$="$1"THENCMD"format 1":GOTO2260
2350 IFDA$="$2"THENCMD"format 2":GOTO2260
2360 IFDA$="$3"THENCMD"format 3":GOTO2260
2370 IFDA$="#1"THENCMD"free 1":GOTO2260
2380 IFDA$="#2"THENCMD"free 2":GOTO2260
2390 IFDA$="#3"THENCMD"free 3":GOTO2260
2400 IFDA$="#0"THENCMD"free 0":GOTO2260
2410 IFDA$="##"THENCMD"free":GOTO2260
2420 IFDA$="01"THENCMD"copy 0,1,,nfat,cbf,usr":GOTO2260
2430 IFDA$="10"THENCMD"copy 1,0,,nfat,cbf,usr":GOTO2260
2440 PRINT@310,"              ";:PRINT@374,"              ";
2450 PRINT@310,"<L>ade";:PRINT@374,"<S>peicher";:PRINT@438,"<M>ische";
2460 T$=INKEY$:IFT$=""THEN2460
2470 IFT$="l"ORT$="L"THENGRLD(DA$)
2480 IFT$="S"ORT$="s"THENGRSV(DA$)
2490 IFT$="M"ORT$="m"THENGRMR(DA$)
2500 OUT1,1
2510 CLS
2520 RETURN
2530 GOSUB480
2540 IFT$=" "THEN2620
2550 FORP=1TO11
2560 IFMO=1THENDOT(X+RND(10),Y+RND(10),1)
2570 IFMO=2THENDOT(X+RND(10),Y+RND(10),0)
2580 IFMO=3THENDOT(X+RND(10),Y+RND(10),-1)
2590 NEXT
2600 IFVB=0THENLINK(320,0,320,191,1)
2610 GOTO2530
2620 RETURN
2630 X1=X:Y1=Y:KO=4:X2=X:Y2=Y:X3=X:Y3=Y
2640 GOSUB480
2650 IFAN=0THEN2670
2660 PRINT@310,"X1: ";X1;:PRINT@374,"Y1: ";Y1;
```

```
2670 IFT$=" "THEN2680ELSE2640
2680 X2=X:Y2=Y
2690 GOSUB480
2700 IFAN=0THEN2720
2710 PRINT@630,"X2: ";X2;:PRINT@694,"Y2: ";Y2;
2720 IFT$=" "THEN2730ELSE2690
2730 X3=X:Y3=Y
2740 KO=0
2750 IFMO=1THEN2760ELSE2800
2760 LINK(X1,Y1,X2,Y2,1)
2770 LINK(X2,Y2,X3,Y3,1)
2780 LINK(X3,Y3,X1,Y1,1)
2790 GOTO2870
2800 IFMO=2THEN2810ELSE2840
2810 LINK(X1,Y1,X2,Y2,0)
2820 LINK(X2,Y2,X3,Y3,0)
2830 LINK(X3,Y3,X1,Y1,0):GOTO2870
2840 LINK(X1,Y1,X2,Y2,-1)
2850 LINK(X2,Y2,X3,Y3,-1)
2860 LINK(X3,Y3,X1,Y1,-1)
2870 CLS:RETURN
2880 X1=X:Y1=Y:KO=5
2890 GOSUB480
2900 IFT$=" "THEN2910ELSE2890
2910 X2=X:Y2=Y:KO=6
2920 X3=X:Y3=Y
2930 GOSUB480
2940 IFT$=" "THEN2950ELSE2930
2950 X3=X:Y3=Y
2960 POKE16405,0
2970 X5=X1+SIN(0)*ABS(X2-X1)
2980 Y5=Y1+COS(0)*ABS(Y2-Y1)
2990 KO=6
3000 FORW=0TO360*BMSTEP30*BM
3010 X=X1+SIN(W)*ABS(X2-X1)
3020 Y=Y1+COS(W)*ABS(Y2-Y1)
3030 IFMO=1THENLINK(X,Y,X5,Y5,1)
3040 IFMO=2THENLINK(X,Y,X5,Y5,0)
3050 IFMO=3THENLINK(X,Y,X5,Y5,-1)
3060 X4=X1+SIN(W+(15*BM))*ABS(X3-X1)
3070 Y4=Y1+COS(W+(15*BM))*ABS(Y3-Y1)
3080 IFMO=1THENLINK(X,Y,X4,Y4,1)
3090 IFMO=2THENLINK(X,Y,X4,Y4,0)
3100 IFMO=3THENLINK(X,Y,X4,Y4,-1)
3110 X5=X4:Y5=Y4
3120 NEXT
3130 POKE16405,1
3140 KO=0:RETURN
3150 GRY:RESUMENEXT
3160 KO=3:X1=X+128:Y1=Y+48:ZX=0:ZY=0
3170 GOSUB480
3180 IFT$=" "THEN3190ELSE3160
3190 XN=0:YN=0
3200 X1=X1-1:Y1=Y1-1
3210 CLS
3220 GRN
3230 XFER2(X,Y,0)
3240 ZZ=0
3250 FORI1=YTOY1
3260 QDOT(X1,I1,F)
3270 IFF=-1THENSET(127,ZZ)ELSERESET(127,ZZ)
3280 ZZ=-NOTZZ
3290 NEXT
3300 ZX=0:ZY=0
3310 FA=POINT(ZX,ZY)
3320 RESET(ZX,ZY)
3330 T$=INKEY$
3340 SET(ZX,ZY)
3350 :
3360 IFT$=""THEN3320
3370 IFFA=0THENRESET(ZX,ZY)
3380 T2=ASC(T$)
3390 IFFA=-1THENSET(ZX,ZY)
3400 IFT2=27THENZY=ZY-2
3410 IFT2=26THENZY=ZY+2
3420 IFT2=25THENZX=ZX+2
3430 IFT2=24THENZX=ZX-2
3440 IFT$="C"ORT$="c"THENCLS
3450 IFT$="s"ORT$="S"THENSET(ZX,ZY)
3460 IFT$="L"ORT$="l"THENRESET(ZX,ZY)
3470 IFT$="O"ORT$="o"THENGOSUB6290
3480 IFT$="I"ORT$="i"THENGOSUB6470
3490 IFT$="R"ORT$="r"THENGOSUB6600
3500 IFT$="M"ORT$="m"THENGOSUB6780
3510 IFT$="H"ORT$="h"THENGOSUB7530
3520 IFT$=" "THENIFFA=-1THENRESET(ZX,ZY)ELSESET(ZX,ZY)
3530 IFT$=CHR$(13)THEN3650
3540 IFT$="K"ORT$="k"THEN3550ELSE3560
3550 BFBOX(0,0,ZX,ZY,-1)
3560 IFT$=CHR$(10)THENIFZY<48THENZY=-NOTZY
```

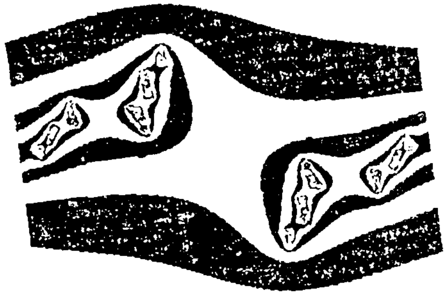


```
3570 IFT$=CHR$(91)THENIFZY>-1THENZY=NOT-ZY
3580 IFT$=CHR$(8)THENIFZX>-1THENZX=NOT-ZX
3590 IFT$=CHR$(9)THENIFZX<128THENZX=-NOTZX
3600 IFZX>127THENZX=0:ZY=-NOTZY
3610 IFZX<0THENZX=127:ZY=NOT-ZY
3620 IFZY<0THENZY=0:ZY=47
3630 IFZY>47THENZY=0:ZX=0
3640 GOTO3310
3650 XFER2(X,Y,2)
3660 CLS
3670 GRY:KO=0
3680 GOTO250
3690 T$=INKEY$:IFT$=""THEN3690
3700 PRINTASC(T$);
3710 END
3720 CLS
3730 RETURN
3740 PRINT@566,"Drucken";:PRINT@630,"<N>ormal";:PRINT@694,"<G>ross";
3750 T$=INKEY$:IFT$=""THEN3750
3760 IFT$=" "THENCLS:RETURN
3770 LPRINTCHR$(27);"A";CHR$(8);CHR$(27);"2";
3780 IFVB=0THENLINK(320,0,320,191,0)
3790 IFT$="N"ORT$="n"THENGR=USR(1)
3800 IFT$="G"ORT$="g"THENGR=USR(0)
3810 T$=CHR$(255)
3820 CLS:RETURN
3830 PRINT@566,"Vollbild?";:PRINT@630,"(J/N)";
3840 TT$=INKEY$:IFTT$=""THEN3840
3850 IFTT$="J"ORTT$="j"THENVB=1ELSEVB=0
3860 PRINT@566,"          ";:PRINT@630,"     ";
3870 QDOT(320,0,FF):IFFF=-1THENIFVB=1THENLINK(320,0,320,191,-1)
3880 RETURN
3890 PRINT@566,"Anzeige?";:PRINT@630,"(J/N)";
3900 TT$=INKEY$:IFTT$=""THEN3900
3910 IFTT$="J"ORTT$="j"THENAN=1ELSEAN=0
3920 CLS
3930 RETURN
3940 IFVB=0THENMX=319ELSEMX=383
3950 PRINT@566,"R-Breite:";:PRINT@630,RB;
3960 T$=INKEY$:IFT$=""THEN3960
3970 IFT$=CHR$(32)THEN4030
3980 IFT$=CHR$(8)THENRB=NOT-RB
3990 IFT$=CHR$(9)THENRB=-NOTRB
4000 IFRB<1THENRB=1
4010 IFRB>20THENRB=20
4020 GOTO3950
4030 CLS:RETURN
4040 IFVB=0THENMX=319ELSEMX=383
4050 FORY1=0TORB
4060 LINK(0,Y1,MX,Y1,1)
4070 LINK(0,191-Y1,MX,191-Y1,1)
4080 NEXT
4090 FORX1=0TORB
4100 LINK(X1,0,X1,191,1)
4110 LINK(MX-X1,0,MX-X1,191,1)
4120 NEXT
4130 RETURN
4140 IFVB=0THENMX=319ELSEMX=383
4150 XAX(0,96,MX,10,1)
4160 YAX(INT(MX/2),0,191,10,1)
4170 RETURN
4180 PRINT@566,"Marker";:PRINT@630,"(E)in(A)us";
4190 TT$=INKEY$:IFTT$=""THEN4190
4200 IFTT$="E"ORTT$="e"THENMA=1ELSEMA=0
4210 CLS
4220 RETURN
4230 IFPU=0THENRETURN
4240 FORI=1TOPU-1
4250 IFMO=1THENLINK(XF(I-1,0),XF(I-1,1),XF(I,0),XF(I,1),1)
4260 IFMO=2THENLINK(XF(I-1,0),XF(I-1,1),XF(I,0),XF(I,1),0)
4270 IFMO=3THENLINK(XF(I-1,0),XF(I-1,1),XF(I,0),XF(I,1),-1)
4280 NEXT
4290 RETURN
4300 IFPU=0THENRETURN
4310 FORJ=1TO2
4320 FORI=1TOPU-1
4330 LINK(XF(I-1,0),XF(I-1,1),XF(I,0),XF(I,1),-1)
4340 NEXT
4350 FORP=1TO150:NEXT
4360 NEXT
4370 RETURN
4380 OUT0,0
4390 CLS
4400 PRINT"<W> Schrittweite   <E> Marker zeigen   <R> Radierer"
4410 PRINT"<T> Text           <Y> Marker Ein/Aus  <U> Unterkommando"
4420 PRINT"<I> Invers Aussch. <O> Marker loeschen <P> Spraydose"
4430 PRINT"<A> Ausdruck       <S> Spiegeln        <D> Dreieck"
4440 PRINT"<F> Verzerren      <G> F}llen          <K> Zeichnen"
4450 PRINT"<J> Marker zeichn. <L> Linie           <*> Zahnrad "
4460 PRINT"<X> Lupe           <C> Kreis           <V> Vollbild"
```

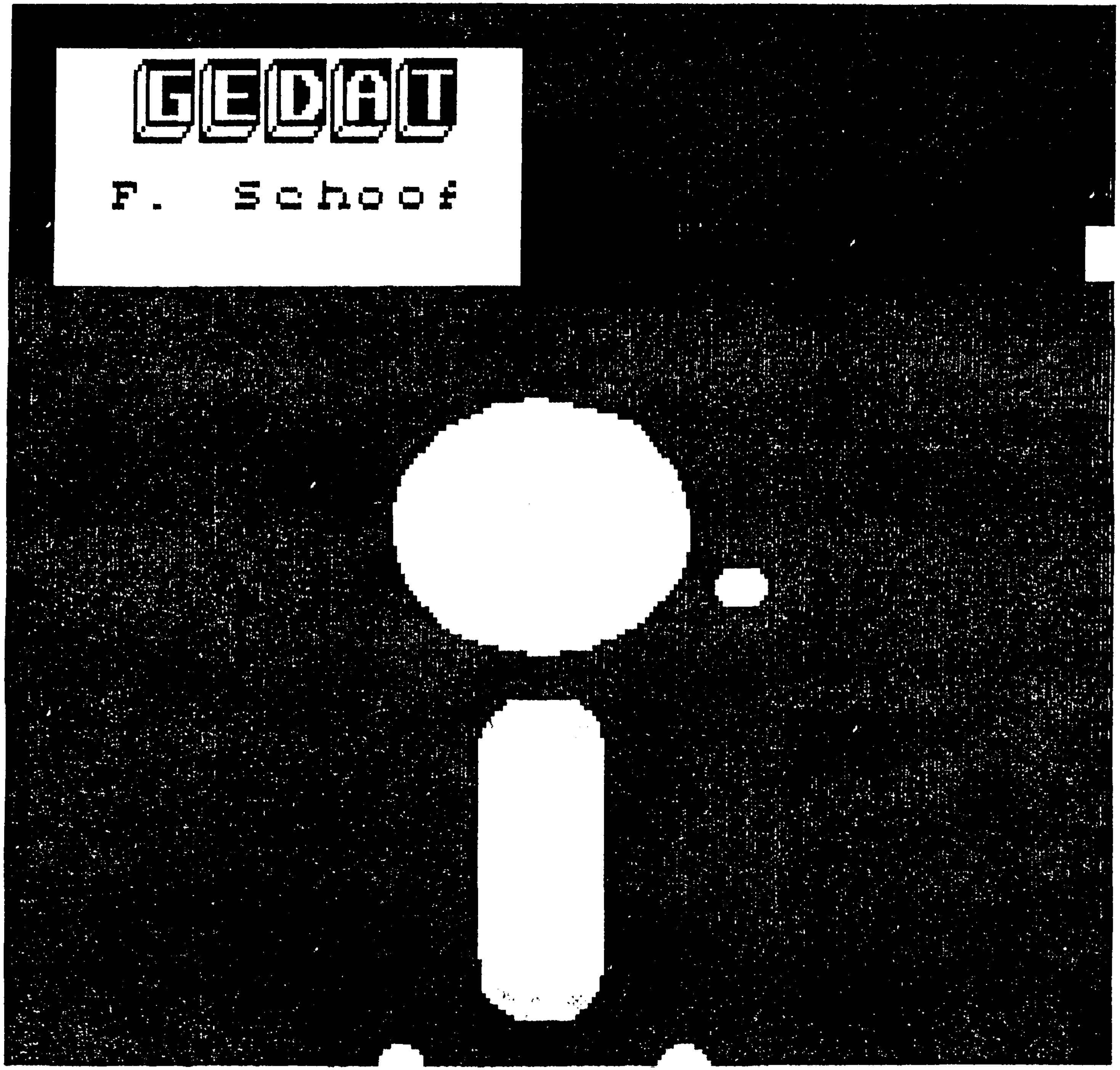
```
4470 PRINT"<B> Kasten         <M> Kopieren        <=> Rahmen"
4480 PRINT"<-> Rahmenstaerke  <1> Setzmodus       <2> Loeschmodus"
4490 PRINT"<3> Inversmodus    <'> Invert Ganz     < > Setze marke"
4500 PRINT"<@> Bild loeschen  <+> Anzeige Ein/Aus <<> Querkreuz"
4510 PRINT"<>> Vollkreuz      <,> Normal Kreuz    <.> Pfeil
4520 PRINT"<)> Koordinatensystem ein  <(> Koordiantensystem aus";
4530 PRINT:PRINT"          Bitte eine Taste druecken!";
4540 TT$=INKEY$:IFTT$=""THEN4540
4550 CLS
4560 PRINT"<H> Kreisausschnitt   <:> Zeichengr|sse   <Z> Zeichensatz"
4570 PRINT"<N> Hilfsfunktion     <$> Minirouter      <#> Baustein"
4580 PRINT"<!> Linieneditor"
4590 PRINT:PRINT:PRINT"      <Bitte eine Taste dr}cken !!!>"
4600 T$=INKEY$:IFT$=""THEN4600
4610 CLS:OUT1,1:RETURN
4620 X1=X:Y1=Y:KO=1
4630 GOSUB480
4640 R=ABS(X-X1):R1=ABS(Y-Y1):IFAN=0THEN1490ELSEPRINT@310,"R1: ";R;:PRINT@374,"RY: ";R1;
4650 IFT$=" "THEN4660ELSE4630
4660 CLS
4670 :
4680 TT$=INKEY$:IFTT$=""THEN4680
4690 IFTT$=CHR$(8)THENWE=-NOTWE
4700 IFTT$=CHR$(9)THENWE=NOT-WE
4710 IFTT$=CHR$(25)THENWE=WE-5
4720 IFTT$=CHR$(91)THENWA=NOT-WA
4730 IFTT$=CHR$(10)THENWA=-NOTWA
4740 IFTT$=CHR$(27)THENWA=WA-5
4750 IFTT$=CHR$(26)THENWA=WA+5
4760 IFTT$=CHR$(24)THENWE=WE+5
4770 IFTT$=CHR$(13)THEN4780ELSE4800
4780 PRINT@566,"WA:";:INPUT"";WA
4790 PRINT@630,"WE:";:INPUT"";WE
4800 IFTT$=" "THEN4850
4810 PRINT@566,"WA:";WA;
4820 PRINT@630,"WE:";WE;
4830 GOTO4680
4840 RETURN
4850 IFMO=1THENCIRCLE(X1,Y1,R,R1,WE+WA,WA,1)
4860 IFMO=2THENCIRCLE(X1,Y1,R,R1,WE+WA,WA,0)
4870 IFMO=3THENCIRCLE(X1,Y1,R,R1,WE+WA,WA,-1)
4880 T$=""
4890 CLS
4900 KO=0:RETURN
4910 PRINT@566,"Baustein:";
4920 X5=X:Y5=Y
4930 PRINT@630,"";:LINEINPUTDA$:DA$=DA$+"/BIN"
4940 IFDA$="/BIN"THENCLS:RETURN
4950 GOSUB4980
4960 CLS
4970 RETURN
4980 Z=1
4990 OPEN"I",#1,DA$
5000 FORI=0TO47
5010 FORO=0TO15
5020 INPUT#1,I$
5030 IFI$=STRING$(8,"0")THENX5=X5+8:GOTO5090
5040 FORL=1TOLEN(I$)
5050 B$=MID$(I$,L,1)
5060 IFB$="1"THENDOT(X5,Y1,1):X5=-NOTX5
5070 IFB$="0"THENX5=-NOTX5
5080 NEXT
5090 NEXT
5100 Y5=Y5+1
5110 X5=X
5120 NEXT
5130 CLOSE
5140 RETURN
5150 SP=SW:XS=X:YS=Y
5160 X1=X:Y1=Y
5170 KO=2
5180 GOSUB480
5190 IFT$=" "THEN5200ELSE5180
5200 X1=X:Y1=Y:X=XS:Y=YS
5210 GOSUB5240
5220 GOSUB5450
5230 SW=SP:KO=0:RETURN
5240 IFY<Y1THENSW=1ELSESW=-1
5250 QDOT(X+2,Y,F)
5260 IFX>=X1THEN5330
5270 IFF=0THENX=X+1:DOT(X,Y,1):GOTO5250
5280 QDOT(X,Y+(SW*2),F)
5290 QDOT(X+2,Y,F1)
5300 IFF=0THENIFF1=-1THENY=Y+1*SW:DOT(X,Y,1):GOTO5280
5310 Y=Y+1*SW:X=NOT-X
5320 IFX<X1THEN5240
5330 IFY<Y1THEN5350
5340 IFY>Y1THEN5420
5350 QDOT(X,Y+SW*2,F)
```

```
5360 QDOT(X+2,Y,F1)
5370 IFF1=0THENIFY>Y1THEN5390
5380 IFF1=-1THENIFF=0THENY=Y+1*SW:DOT(X,Y,1):GOTO5350
5390 IFX<X1THENY=Y+SW*1:X=X-1:GOTO5240
5400 IFX>=X1THEN5420
5410 GOTO5440
5420 QDOT(X,Y-(SW*2),F)
5430 IFF=0THENY=Y+1*SW:DOT(X,Y,1):GOTO5420
5440 RETURN
5450 IFY<Y1THEN5470
5460 IFY>Y1THEN5500
5470 QDOT(X,Y+2,F)
5480 IFF=0THENIFY<Y1THENY=-NOTY:DOT(X,Y,1):GOTO5470
5490 RETURN
5500 QDOT(X,Y-2,F)
5510 IFF=0THENIFY>Y1THENY=NOT-Y:DOT(X,Y,1):GOTO5500
5520 RETURN
5530 IFVB=0THENMX=319ELSEMX=383
5540 XAX(0,96,MX,10,0)
5550 YAX(INT(MX/2),0,191,10,0)
5560 RETURN
5570 CLS
5580 IFVB=0THEN5600ELSE5750
5590 SBOX(321,100,383,180,MU)
5600 FORI=0TO6
5610 SBOX(321,I*12,383,I*12+10,I)
5620 NEXT
5630 FBOX(321,MU*12,383,MU*12+10,-1)
5640 T$=INKEY$:IFT$=""THEN5640
5650 FBOX(321,MU*12,383,MU*12+10,-1)
5660 IFT$=CHR$(91)THENIFMU>0THENMU=NOT-MU
5670 IFT$=CHR$(10)THENIFMU<6THENMU=-NOTMU
5680 IFT$=CHR$(32)THEN5710
5690 SBOX(321,100,383,180,MU)
5700 GOTO5630
5710 CLS
5720 FBOX(321,0,383,191,0)
5730 FILL(X,Y,MU)
5740 CLS:RETURN
5750 PRINT@566,"Muster:";
5760 PRINT@630,"(0-6) ?";
5770 T$=INKEY$:IFT$=""THEN5770
5780 IFT$<"0"ORT$>"6"THENCLS:RETURN
5790 MU=VAL(T$)
5800 FILL(X,Y,MU)
5810 CLS:RETURN
5820 END
5830 CLS
5840 GRN
5850 PRINT@512,"Text: ";:LINEINPUTT$
5860 SKIP(X,Y)
5870 PRINT@640,"Zeilenvorschub (J/N) ?":
5880 TT$=INKEY$:IFTT$=""THEN5880
5890 IF TT$="J" OR TT$="j" THEN PRTLN(T$) ELSE PRT(T$)
5900 CLS
5910 GRY
5920 RETURN
5930 CLS:PRINT@566,"Zeichenstz";
5940 PRINT@630,"laden:";
5950 PRINT@694,"<S>tandard";
5960 PRINT@758,"<D>olur.";
5970 T$=INKEY$:IFT$=""THEN5970
5980 IF T$="S" OR T$="s" THEN ZS=0:SELECT(0)
5990 IF T$="D" OR T$="d" THEN ZS=1:SELECT(1)
6000 CLS
6010 PRINT@566,"Name des";
6020 PRINT@630,"Fonts:";
6030 PRINT@694,"";:LINEINPUTZS$
6040 IFZS$=""THENCLS:RETURN
6050 IFZS=0THENLDF(ZS$)
6060 IF ZS=1 THEN LDD(ZS$)
6070 CLS:RETURN
6080 CLS
6090 PRINT@566,"Zeichen-";
6100 PRINT@630,"groesse:";
6110 PRINT@694,"<N>ormal";
6120 PRINT@758,"<D>oppelt";
6130 T$=INKEY$:IFT$=""THEN6130
6140 IFT$="D"ORT$="d"THENDB=12ELSEDB=6
6150 IF DB=6 THEN IF ZS=0 THEN PTYPE(0)
6160 IF DB=12 THEN IF ZS=0 THEN PTYPE(4)
6170 CLS
6180 RETURN
6190 X1=X:Y1=Y:KO=3
6200 GOSUB480
6210 IFT$=" "THEN6220ELSE6200
6220 X2=X:Y2=Y:KO=0
6230 GOSUB480
6240 IFT$=" "THEN6250ELSE6230
6250 FORXX=X1TOX2
```

```
6260 COPY(XX,Y1,XX,Y2,X-X1,(Y-Y1)+SIN(XX/8)*3,MO-1)
6270 NEXT
6280 RETURN
6290 GRN:FOR I=15360 TO 16363 STEP 254
6300 FORO=0TO254
6310 MID$(BZ$(BZ),O+1,1)=CHR$(PEEK(I+O))
6320 POKEI+O,32
6330 NEXT
6340 BZ=BZ+1
6350 NEXT
6360 PRINT@960,STRING$(63,32);
6370 BZ=0
6380 PRINT@960,"";:LINEINPUT"Name des ZOOM-Datei: ";ZD$
6390 CLS
6400 OPEN"O",#1,ZD$
6410 FORI=0TO3
6420 PRINT#1,BZ$(I);
6430 PRINTBZ$(I);
6440 NEXT
6450 CLOSE
6460 RETURN
6470 GRN:PRINT@960,STRING$(63,32);
6480 PRINT@960,"":LINEINPUT"Name der ZOOM-Datei: ";ZD$
6490 CLS
6500 BZ=15360
6510 OPEN"I",#1,ZD$
6520 LINEINPUT#1,A$
6530 FORI=1TOLEN(A$)
6540 POKEBZ,ASC(MID$(A$,I,1)):BZ=BZ+1
6550 NEXT
6560 IFNOTEOF(1)THEN6520
6570 CLOSE
6580 BZ=0
6590 RETURN
6600 XX=ZX:YY=ZY
6610 BBOX(XX,YY,ZX,ZY,-1)
6620 T2=PEEK(14440)
6630 IFT2=0THEN6620
6640 BBOX(XX,YY,ZX,ZY,-1)
6650 IFT2=8THENZY=NOT-ZY
6660 IFT2=16THENZY=-NOTZY
6670 IFT2=32THENZX=NOT-ZX
6680 IFT2=64THENZX=-NOTZX
6690 IFT2=128THEN6750
6700 IFZX<0THENZX=127
6710 IFZX>127THENZX=0
6720 IFZY<0THENZY=47
6730 IFZY>47THENZY=0
6740 GOTO6610
6750 BFBOX(XX,YY,ZX,ZY,0)
6760 T2=0:POKE14440,0
6770 RETURN
6780 XX=ZX:YY=ZY
6790 BLINK(XX,YY,ZX,ZY,-1)
6800 T2=PEEK(14440)
6810 IFT2=0THEN6800
6820 BLINK(XX,YY,ZX,ZY,-1)
6830 IFT2=8THENZY=NOT-ZY
6840 IFT2=16THENZY=-NOTZY
6850 IFT2=32THENZX=NOT-ZX
6860 IFT2=64THENZX=-NOTZX
6870 IFT2=128THEN6930
6880 IFZX<0THENZX=127
6890 IFZX>127THENZX=0
6900 IFZY<0THENZY=47
6910 IFZY>47THENZY=0
6920 GOTO6790
6930 T2=0
6940 POKE14440,0
6950 BLINK(XX,YY,ZX,ZY,1)
6960 RETURN
6970 CLS
6980 GRY
6990 GRSV("SPEICHER/GRF:0")
7000 CLH
7010 LN=2
7020 GRID(20,120,40,20,8,1,1)
7030 FORI=0TO4:PRINT@I*64,I+2;:NEXT
7040 X=22
7050 Y=102
7060 SW=40
7070 SX=36
7080 SY=16
7090 FBOX(X,Y,X+SX,Y+SY,-1)
7100 T$=INKEY$:IFT$=""THEN7100
7110 FBOX(X,Y,X+SX,Y+SY,-1)
7120 IFT$=CHR$(8)THENIFX>22THENX=X-40:PS=NOT-PS
7130 IFT$=CHR$(9)THENIFX<301THENX=X+40:PS=-NOTPS
7140 IFT$=CHR$(115)ORT$=CHR$(83)THENIFPZ(PS)=0THENFBOX(X,Y,X+SX,Y+SY,1):DZ=DZ+2[PS:PZ(PS)=1
```



```
7150 IFPZ(PS)=1THENIFT$=CHR$(76)ORT$=CHR$(108)THENFBOX(X,Y,X+SX,Y+SY,0):DZ=DZ-2[PS:PZ(PS)=0
7160 IFT$=CHR$(31)THENGOSUB7270
7170 IFT$=" "THEN7490
7180 IFT$=CHR$(73)ORT$=CHR$(105)THENGOSUB7320
7190 IFT$="+"THENIFLN<7THENLN=-NOTLN
7200 IFT$="-"THENIFLN>2THENLN=NOT-LN
7210 PRINT@708,"Dezimal: ";USING"###";DZ;:PRINT@748,"Liniennr.: ";USING"#";LN;
7220 POKE-513-(LN-2),DZ
7230 FORI=2TO6
7240 LINK(20,(I-2)*12+2,383,(I-2)*12+2,I)
7250 NEXT
7260 GOTO7090
7270 :
7280 GRID(20,120,40,20,8,1,1)
7290 FORI=0TO7:PZ(I)=0:NEXT
7300 DZ=0
7310 RETURN
7320 FORI=0TO7
7330 IFPZ(I)=0THENPZ(I)=1ELSEPZ(I)=0
7340 NEXT
7350 X=22
7360 Y=102
7370 SX=36
7380 SY=16
7390 DZ=0
7400 FORI=0TO7
7410 IFPZ(I)=0THENFBOX(X,Y,X+SX,Y+SY,0)
7420 IFPZ(I)=1THENFBOX(X,Y,X+SX,Y+SY,1)
7430 IFPZ(I)=1THENDZ=DZ+2[I
7440 X=X+SW
7450 NEXT
7460 X=22
7470 PS=0
7480 RETURN
7490 GRN
7500 GRLD("SPEICHER/GRF:0")
7510 GRY
7520 CLS:RETURN
7530 LPRINTCHR$(27);"A";CHR$(4);CHR$(27);"2";
7540 XS=X:YS=Y
7550 FORY=0TO47
7560 LPRINTCHR$(9);CHR$(9);CHR$(27);"Z";:OUT253,0:OUT253,4
7570 FORX=0TO127
7580 FA=POINT(X,Y)
7590 IFFA=-1THENGOSUB7650
7600 IFFA=0THENGOSUB7690
7610 NEXT
7620 OUT&HFD,10
7630 NEXT
7640 X=XS:Y=YS:RETURN
7650 FORI=1TO8
7660 OUT253,127
7670 NEXT
7680 RETURN
7690 FORI=1TO8
7700 OUT253,0
7710 NEXT
7720 RETURN
7730 MO=0:CLS
7740 IFVB=1THEN7860
7750 FORI=2TO7
7760 LINK(325,I*10,376,I*10,I)
7770 BOX(321,(I*10)-4,380,(I*10)+4,1)
7780 NEXT
7790 FBOX(321,(LT*10)-4,380,(LT*10)+4,-I)
7800 T2=PEEK(14440):IFT2=0THEN7800
7810 FBOX(321,(LT*10)-4,380,(LT*10)+4,-1)
7820 IFT2=8THENIFLT>2THENLT=LT-1
7830 IFT2=16THENIFLT<7THENLT=LT+1
7840 IFT2=128THENFBOX(321,0,383,191,0):RETURN
7850 GOTO7790
7860 PRINT@54,"Linien-";
7870 PRINT@118,"type: ";
7880 PRINT@181,LT;
7890 T2=PEEK(14440):IFT2=0THEN7890
7900 IFT2=8THENIFLT>2THENLT=LT-1
7910 IFT2=16THENIFLT<7THENLT=LT+1
7920 IFT2=128THENCLS:RETURN
7930 GOTO7880
7940 '
7950 '  ******************************
7960 '  *                            *
7970 '  *         G E C A D          *
7980 '  *                            *
7990 '  *         (c) 1990:          *
8000 '  *                            *
8010 '  *      Frank   Schoof        *
8020 '  *                            *
8030 '  ******************************
```


GEDAT
F. Schoof

GEDAT ist ein Datenbankprogramm, welches unter **GEGRAS** läuft. Es können nicht nur Adressen verwaltet werden, sondern alle Arten von Daten.

Diese Daten können z. B. Adressen, Listen, Lexika, uva. sein. Für diese Daten kann eine spezielle Maske erstellt werden.

GEDAT ist in **BASIC** geschrieben und benötigt für das schnelle Arbeiten die erhöhte Taktfrequenz von 3.5 Mhz.

1.0 START

Gestartet wird **GEDAT** von **GEGRAS** aus unter dem Menüpunkt **WERKZEUGE** . Es wird dann geladen und anschließend gestartet.

Es zeigt sich dann folgendes Bild:

2.0 TISCH

Dieser Menüpunkt ist der Gleiche wie unter **GEGRAS** . Die Untermenüpunkte haben die selbe Funktion wie bei **GEGRAS** . Bei Unklarheiten hilft das **GEGRAS** Handbuch weiter.

3.0 EDITIEREN

Mit diesem Menüpunkt hat man die Möglichkeit, eine Maske für eine Datei zu erstellen.

Eine Maske stellt den Bildschirm für die spätere Ausgabe der Datei dar.

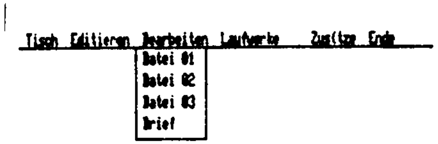
Wie eine Maske aussehen kann, und wie der Maskeneditor bedient wird, ist im Kapitel **MASKEN** nachzulesen.

4.0 BEARBEITEN

Unter **GEDAT** könne bis zu drei Dateien offen sein. So kann z. B. als erste Datei eine Adressdatei, als zweite eine Modellliste und als dritte eine Kochrezeptliste offen sein.

Desweiteren kann noch ein Brief bearbeitet werden. Hierzu unter **BRIEF** mehr.

Nach dem Anwählen dieses Menüpunktes erscheint auf dem Monitor:



Es weren bei den zu bearbeitenden Dateien die Namen gespeichert und beim nächsten Aufrauf wieder angezeigt.

Auch wenn die Arbeit abgebrochen wird, werden die Namen gesichert.

4.1 DATEI 1

Wählt die erste Datei an. Es wird anschließend nach dem Masken/Dateinamen gefragt.

Ist dieser o. k., dann wird die entsprechende Datei zur Bearbeitung freigegeben.

4.2 DATEI 2

Wie 4.1 nur hier für Datei 2.

4.3 DATEI 3

Wie 4.2 und 4.1 nur hier für Datei 3.

4.4 BRIEF

Wird dieser Menüpunkt angewählt, so kann ein Brief erstellt werden.

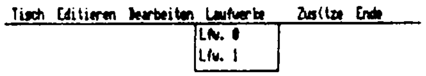
In diesem Brief könne DOTWRITER(r) Zeichensätze verwendet werden.

Mehr hierzu unter **BRIEF** .

5.0 LAUFWERKE

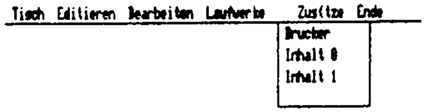
An dieser Stelle wird eingestellt, in welchem Laufwerk sich die Diskette mit den Dateien befindet.

Ist dieses das Laufwerk 0, so wird einfach der Menüpunkt Lfw.0 'angeklickt'. Ansonsten der Menüpunkt Lfw.1 .



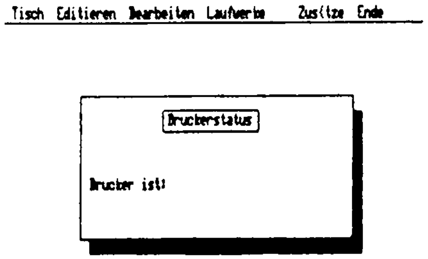
6.0 ZUSÄTZE

Wurde dieser Menüpunkt gewählt, so erscheint auf dem Monitor dieses Bild:



6.1 DRUCKER

Es wird der aktuelle Druckerstatus angezeigt, wie im folgenden Bild zu sehen ist.



6.2 Inhalt 0

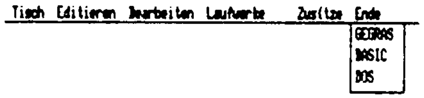
Der Inhalt der Diskette in Laufwerk 0 wird auf dem Bildschirm ausgegeben.

6.3 Inhalt 1

Der Inhalt der Diskette in Laufwerk 1 wird auf dem Bildschirm augegeben.

7.0 ENDE

An dieser Stelle kann GEDAT verlassen werden, ins BASIC oder ins DOS gesprungen werden.



Kommen wir nun zu den einzelnen, speziellen Dingen von GEDAT , wie die Masken.

MASKEN

für die Datenverarbeitung ist bei **GEDAT** immer eine Maske nötig.

Solch eine Maske zeigt dem Benutzer, wo er seine Eingabe tätigen kann und zeigt ihm auch, wo er seine Daten finden kann.

Eine Maske für eine Adressdatei kann z. B. so aussehen:

```
Name:                     Vorname:
Straße:
PLZ:          Wohnort:
Telefon:

Bemerkung:
```

Wird diese Maske für die Eingabe benutzt, so kann sie so aussehen:

```
Name:..............       Vorname:.................
Straße:..........................................
PLZ:......    Wohnort:...........................
Telefon:...............

Bemerkung:.......................................
```

Überall, wo jetzt die Punkte stehen, kann später eine Eingabe stehen.

Ist eine Eingabe erfolgt, so sieht die fertige Datenmaske dann z. B. so aus:

```
Name:Franz                Vorname:August
Straße:Tellerstraße 22
PLZ:D-4312    Wohnort:Meinenfeld 22
Telefon:02134/23 11 22

Bemerkung:Computertrefforganisator im Club
```

Ein solche Maske hat denselben Namen, den später die Datei haben soll. Als Dateiendung hat eine Maske /FLD. Eine Datei hingegen hat die Endung /BAK.

Erstellt wird eine Maske unter **GEDAT** mit dem Menüpunkt **EDITIEREN** .

Wurde dieser Menüpunkt aufgerufen, erscheint auf dem Bildschirm die Aufforderung, den Masken-/Dateinamen einzugeben. Abgeschlossen wird die Eingabe mit der (ENTER) Taste.

Ist der Name korrekt, so erscheint nach kurzer Wartezeit ein leerer Bildschirm und in der oberen linken Ecke blinkt der Eingabecursor, wie auf dem nachfolenden Ausdruck zu sehen ist.

Es kann nun dort ein Text plaziert werden, wo er später auch erscheinen soll.

Das Ganze funktioniert wie ein ASCII-Editor, nur, daß der Bereich hier auf einen Bildschirm begrenzt ist.

Damit das Programm später auch erkennt, wo eine Eingabe erscheine soll, müssen diese Felder mit einem Punkt(.) markiert werden. Das Ende muß mit einem Sternchen markiert werden.

Das Programm erkennt, wann es Text vorfindet, und welche Felder zu dem Text gehören.

Zum besseren Verständnis hier ein Beispiel:

BEISPIEL

Zuerst wird eine Maske erstellt, die so aussieht:

```
Name:
Vorname:
Straße:
PLZ:            Wohnort:
Telefon:
```

Der Cursor steht nun hinter **TELEFON:** . Nun wird er auf das N von **NAME:** bewegt. An dieser Stelle wird überlegt, wie lang die einzelnen Felder für sein sollen. Die Gesamtlänge darf 255 Zeichen nicht überschreiten !!!

In unserem Beispiel sollen einmal folgende Werte gelten:

```
Name     = 10
Vorname  = 10
Straße   = 15
PLZ      = 06
Wohnort  = 20
Telefon  = 10
           --
           71
```

Diese Werte werden nun in die Maske per Tastatur eingetragen.

GEDAT

Name:..........(STERNCHEN)
Vorname:..........(STERNCHEN)
Straße:................(STERNCHEN)
PLZ:......(STERNCHEN)
Wohnort:....................(STERNCHEN)
Telefon:..........(STERNCHEN)

Vor den Punkten muß immer ein Doppelpunkt(:) stehen !!!

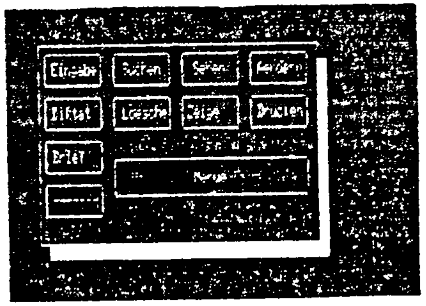
Mit der (CLEAR) Taste wird die Maske gelöscht und auch der Bildschirm. Der Cursor erscheint dann wieder in der linken oberen Ecke.

Mit der (SHIFT + PFEIL LINKS) Taste wird die Maske abgespeichert und GEDAT wird aufgerufen.

GEDAT

DATEN BEARBEITEN

Ist eine Datei zum Bearbeiten frei gegeben (Menüpunkt Bearbeiten), so erscheint auf dem Monitor:



1.0 EINGABE

Mit diesem Fenstermenüpunkt können Daten in eine zu der Datei passende Maske eingegeben werden.

Jedes Feld muß mit der (ENTER) Taste beschloßen werden.

Sind alle Felder eingegeben, wird nachgefragt, ob alle Eingaben in Ordnung sind. Diese Frage kann mit den Tasten (J/j .. N/n) beantwortet werden. Wird die Frage mit N/n beantwortet, so wird der Bildschirm gelöscht und es können die Eingaben von neuem getätigt werden.

Die Eingaben werden anschließend gespeichert, der interne Zähler wird erhöht und es erscheint wieder das Menüfenster.

2.0 DIKTAT

Funktioniert wie die **EINGABE** , nur, daß hierbei nach der Eingbe nicht in das Menüfenster gesprungen wird.

Ist die Eingabe o. k., dann wird sie gesichert und es kann die nächste Eingabe erfolgen.

Zurück zum Menüfenster kommt man mit der Eingabe (STERNCHEN)ENDE(STERNCHEN).

3.0 BRIEF

Mit diesem Menüfensterpunkt kann die Anschriftreihenfolge für den Brief festgelegt werden.

Als erstes muß ein Begriff zum Suchen eingegeben werden. Wurde der Begriff gefunden, so kann die Reihenfolge festgelegt werden.

GEDAT

Wie sicher schon bemerkt wurde, sind die Menüpunkte sehr kurz erklärt worden. Ich bin der Meinung, daß bei einem Programm das Probieren über Studieren (langweiliger Handbücher) geht.

Das Beste ist es, wenn man die Menüpunkte ausprobiert. Frei nach dem Motto:

M A C H E,
S E H E,
S T A U N E ! ! !

BRIEF

Nach dem Laden dieses Unterprogrammes wird gefragt, ob ein Brief geschrieben werden soll, oder ob nur ein Text geschrieben werden soll.

Ist der Brief gewählt worden, so wird der Absender von der Datei ABSEND/DAT geladen. Noch steht dort meine Adresse.

Wurde eine Anschriftenreihenfolge eingegeben, so wird die Anschrift geladen.

Dann kann der Brief eingegeben werden.

In dem Brief können Steuerzeichen verwendet werden. Als Erkennungszeichen dient das (@)-Zeichen.

Es gibt zwei Steuerzeichen. Da wären erst einmal die **Formatsteuerzeichen** , welche die Textformatierung steuern, und dann gibt es noch die **Schriftsteuerzeichen**.

An Formatsteuerzeichen gibt es:

@BL	--->	BLOCKSATZ
@ZE	--->	ZENTRIERT
@LB	--->	LINKSBÜNDG
@RB	--->	RECHTSBÜNDIG
@NO	--->	AUFHEBUNG der vorhergehenden!

Auf der nächsten Seite sind die Schriftsteuerzeichen zu finden.

Voraussetzung ist allerdings das Vorhandensein der entsprechenden DOTWRITER(r)-Zeichensätze.

GEDAT

Abgeschlossen wird die Eingab mit einem '/'-Zeichen.

Beispiel: 2/1/3/4/

Diese Reihenfolgen werden abgespeichert und bei dem BRIEF wieder geladen.

4.0 SUCHEN

An dieser Stelle hat man die Möglichkeit, eine Begriff suchen zu lassen und, wenn er gefunden wurde, den passenden Datensatz anzuzeigen und evtl. zu drucken.

Der Suchbegriff wird in der ganzen Datei gesucht.

5.0 LOESCHEN

Es kann ein bestimmter Datensatz gelöscht werden.

Gelöscht wird der Datensatz, in dem der Begriff auftaucht, den man eingegeben hat.

6.0 SEHEN

Es können bestimmte Daten angesehen werden.

Dazu wird die Datensatznummer (>1) von dem Benutzer benötigt.

Der passende Datensatz wird dann angezeigt. Mit der (LEERTASTE) gelangt man zurück zum Menüfenster.

7.0 ZEIGEN

Mit dieser Funktion kann zwischen allen Datensätzen geblättert werden.

In der obersten Zeile stehen die Tasten, mit denen geblättert werden kann.

8.0 AENDERN

Datensätze, die nicht mehr dem aktuellen Stand entsprechen, können hiermit geändert werden.

9.0 DRUCKEN

Bestimmte Datensätze lassen sich hiermit ausdrucken.

Zeichensatztabelle

Die folgende Tabelle zeigt die Steuercodes für den GEDAT-Brief!

Steuerzeichen:	Zeichensatz:
@FE	FETT ein
@FA	FETT aus
@UE	UNTERSTREICHEN ein
@UA	UNTERSTREICHEN aus
@BE	BREIT ein
@BA	BREIT aus
@HE	HERVORHEBEN ein
@HA	HERVORHEBEN aus
@TE	HOCHSTELLEN ein
@TA	HOCHSTELLEN aus
@LM	LMES/FON
@LQ	LQ/FON
@CS	CS
@MI	MID
@CM	CM/FON
@P1	PL1
@M1	MB1
@O1	OE1
@OE	OE
@FS	FSC/FON
@SP	SPL
@MC	MC1
@IP	IPL/FON
@TR	TR
@ST	STR
@AN	ANT
@BQ	BALL
@BB	BB
@BO	BO
@BW	BW
@FF	FF
@IS	ISE
@IH	ISHADED
@M2	MB2
@SC	SCOMP
@SH	SHADOW
@SC	STENCIL
@GR	GREEK
@LE	LED
@HS	SCRIPT
@IE	INVERS EIN
@IA	INVERS AUS

Die Steuercodes dürfen allerdings nur **ZEILENWEISE** verwendet werden !!!

Mit der (CLEAR)-Taste springt der Cursor in die linke obere Ecke des Bildschirmes.

Mit den (PFEIL)-Tasten kann der Cursor bewegt werden. Beim Verlassen des Bildschirmes in vertikaler Richtung verschiebt sich der Bildschirmtext entsprechend.

Zusammen mit der (SHIFT)-Taste haben die (PFEIL)-Tasten noch weitere Funktionen.

Einmal zum Sichern des Briefes, dann zum Laden eines Briefes, zum Löschen einer Zeile und zum Ausdrucken des Briefes.

Beim Ausdrucken wird noch einmal gefragt, ob der Brief/Text gedruckt werden soll.
Ist dieses der Fall, so kann der Ausdruck eine Weile (BASIC) dauern.

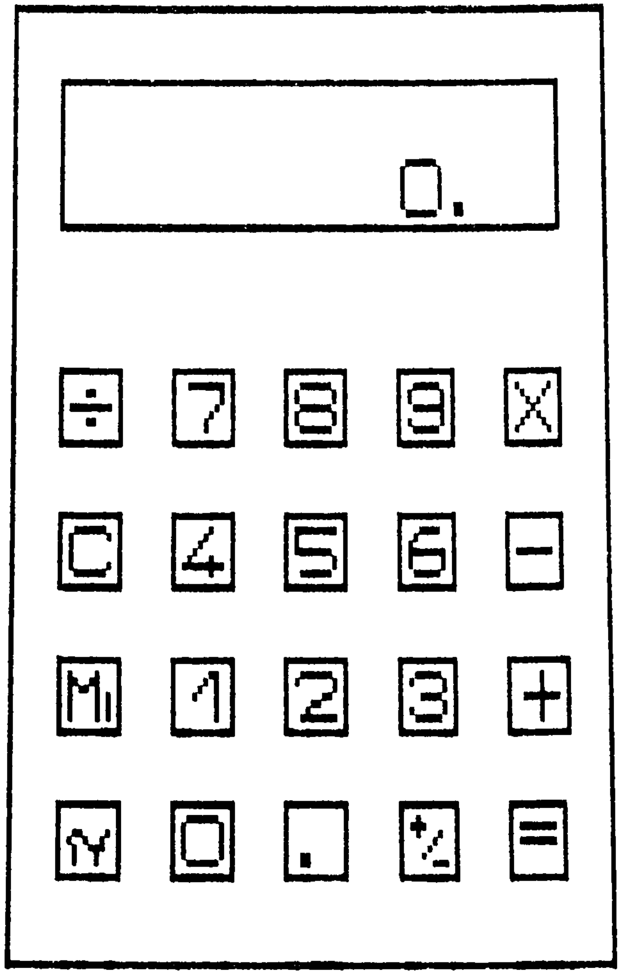
Dieses ist für das Datenbankprogramm alles, was es von mir zu sagen gibt. Also ran ans Gerät, die Finger geschmeidig gemacht und probieren !

Ach ja, hier noch das Listing von **GEDAT** .

```
10 CLEAR5000:UH=1:DEFINTA-Z:DI=1:ONERRORGOTO3480:POKE16708,0:X=190:Y=10:XS=X:YS=Y:GRY:CLH:GOSUB170:GO=0:LW$=":0"
20 DIMSA$(40),LA(60),SS(30,2),S$(15,63),DQ$(8)
21 FORI=1TO4:DQ$(I+2)=" ":NEXT
30 OPEN"I",#1,"NOTIZ/DAT":FORI=0TO4:LINEINPUT#1,NO$(I):NEXT:CLOSE#1
31 OPEN"I",#1,"MASK/DAT":FORI=1TO3:LINEINPUT#1,DQ$(1+2):NEXT:CLOSE#1
50 GOSUB80
60 IFGO=1THENGOSUB970:GO=0
70 GOTO50
80 GOSUB200
90 GO=0:T2=PEEK(14440):IFT2=0THEN90
91 GOSUB200
100 IFT2=8THENIFY>0THENY=Y-6
110 IFT2=16THENIFY<191THENY=Y+6
120 IFT2=32THENIFX>0THENX=X-6
130 IFT2=64THENIFX<383THENX=X+6
140 IFT2=40THENX=X-6:Y=Y-6
141 IFT2=72THENX=X+6:Y=Y-6
142 IFT2=48THENX=X-6:Y=Y+6
143 IFT2=80THENX=X+6:Y=Y+6
150 IFT2=128THENGO=1
160 RETURN
170 CLH:CLS
180 PRINT@0," Tisch  Editieren  Bearbeiten  Laufwerke     Zus";CHR$(123);"tze  Ende";
190 LINK(0,9,383,9,1):RETURN
200 IFCU=0THENLINK(X,Y,X,Y+8,-1):LINK(X-1,Y+1,X-3,Y+3,-1):LINK(X+1,Y+1,X+3,Y+3,-1):RETURN
210 IFCU=1THENLINK(X/5,Y,X+5,Y,-1):LINK(X,Y-5,X,Y+5,-1)
220 RETURN
440 IFBA=0THENCLEAR:RUN"MASTER"
450 IFBA=1THENNEW:END
750 GOSUB170
760 X=70:Y=50:X1=320:Y1=130:GOSUB3310
770 PRINT@345,"Druckerstatus";
790 X=146:Y=58:X1=234:Y1=70:GOSUB3540
791 FORO=1TO10:T$="":NEXT
800 PRINT@525,"Drucker ist: ";
810 IN=INP(253)
820 IFPEEK(14440)=128THEN900
830 IFIN=63THENPRINT"ON LINE      ";
840 IFIN=47THENPRINT"OFF LINE     ";
850 IFIN=191THENPRINT"AUSGESCHALTET";
860 GOTO800
900 GOSUB170:RETURN
970 :
990 XW=X:YW=Y:IFX>-1THENIFX<44THEN1080
1000 IFX>43THENIFX<101THENGOSUB1290
1010 IFX>108THENIFX<176THENGOSUB1420
1020 IFX>177THENIFX<229THENGOSUB1690
1030 IFX>228THENIFX<299THENGOSUB1830
1040 IFX>299THENIFX<337THENGOSUB2000
1050 :
1060 GO=0
1070 GOTO50
1080 GOSUB170
1090 LINK(0,10,0,68,1)
1100 PRINT@64," Uhr  ";
1110 PRINT@128," Notiz ";
1120 PRINT@192," Druck ";
1130 PRINT@256," Copy. ";
```



```
1140 PRINT@320," Hilfe I";
1150 LINK(0,68,56,68,1)
1160 LINK(56,68,56,10,1)
1170 X=XS:Y=YS
1180 GO=0:GOSUB80
1190 IFGO=1THEN1210ELSE1180
1210 IFX>-1THENIFX<57THEN1220ELSE1050
1220 IFY>9THENIFY<21THEN2130
1230 IFY>21THENIFY<36THENGOSUB2370:GOTO1050
1240 IFY>36THENIFY<45THENGOSUB2670:GOTO1050
1250 IFY>45THENIFY<57THENGOSUB2710:GOTO1050
1260 IFY>57THENIFY<69THENDD$="PMODULTH":GOTO40000
1270 GOTO1180
1290 GOSUB170
1300 PRINT@71," Maske ";
1330 LINK(45,10,45,25,1)
1340 LINK(45,25,113,25,1)
1350 LINK(113,25,113,10,1)
1360 GO=0:GOSUB80
1370 IFGO=0THEN1360
1380 IFX>45THENIFX<111ANDY>9ANDY<23THENDD$="PMODULMA":GOTO40000
1410 GOTO1050
1420 GOSUB170
1430 PRINT@82," Datei 01";
1440 PRINT@146," Datei 02";
1450 PRINT@210," Datei 03 ";
1460 PRINT@274," Brief ";
1530 LINK(108,10,108,60,1)
1540 LINK(108,60,172,60,1)
1550 LINK(172,60,172,10,1)
1560 GO=0:GOSUB80:IFGO=0THEN1560
1570 GO=0
1580 IFX>109THENIFX<171THEN1590ELSE1680
1590 IFY>9THENIFY<23THENLW=0:GOSUB3420:GOTO1050
1600 IFY>22THENIFY<35THENLW=1:GOSUB3420:GOTO1050
1610 IFY>34THENIFY<47THENLW=2:GOSUB3420:GOTO1050
1620 IFY>46THENIFY<59THENDD$="PMODULBR":GOTO40000
1680 GOTO1050
1690 GOSUB170
1700 PRINT@93," Lfw. 0 ";
1710 PRINT@157," Lfw. 1";
1720 LINK(176,10,176,36,1)
1730 LINK(176,36,253,36,1)
1740 LINK(253,36,253,10,1)
1770 GO=0:GOSUB80:IFGO=0THEN1770
1780 IFX>177THENIFX<235THEN1790ELSE1820
1790 IFY>9THENIFY<23THENLW$=":0"
1800 IFY>22THENIFY<35THENLW$=":1"
1810 GOSUB170
1820 GOTO1050
1830 GOSUB170
1840 PRINT@106," Drucker ";
1850 PRINT@170," Inhalt 0 ";
1860 PRINT@234," Inhalt 1 ";
1890 LINK(252,10,252,56,1)
1900 LINK(252,56,336,56,1)
1910 LINK(336,56,336,10,1)
1920 GOSUB80:IFGO=0THEN1920
```



RECHNER

```
1930 IFX>231THENIFX<297THEN1940ELSE1920
1940 IFY>9THENIFY<23THEN750
1950 IFY>22THENIFY<35THENLW=0:GOTO3590
1960 IFY>34THENIFY<47THENLW=1:GOTO3590
1990 GOTO1050
2000 GOSUB170
2010 PRINT@117," GEGRAS ";
2020 PRINT@181," BASIC ";
2030 PRINT@245," DOS ";
2040 LINK(320,10,320,48,1)
2050 LINK(320,48,370,48,1)
2060 LINK(370,48,370,10,1)
2070 GOSUB80:IFGO=0THEN2070
2080 IFX>301THENIFX<347THEN2090ELSE2120
2090 IFY>9THENIFY<23THENBA=0:GOTO440
2100 IFY>21THENIFY<35THENBA=1:GOTO440
2110 IFY>33THENIFY<47THENNEW
2120 GOTO1050
2130 DD$="PMODULU"
2140 GOTO40000
2370 GOSUB170:ZY=1:X=50:X1=350:Y=40:Y1=180:GOSUB3310
2380 X=55:Y=45:X1=345:Y1=110:GOSUB3540
2390 PRINT@910,"Loeschen";
2400 PRINT@925,"Editieren";
2410 PRINT@940,"Menue";
2420 X=79:X1=139:Y=165:Y1=177:GOSUB3540
2430 X=171:Y=165:X1=231:Y1=177:GOSUB3540
2440 X=259:Y=165:X1=299:Y1=177:GOSUB3540
2450 FORI=0TO4:PRINT@266+(I*64),NO$(I);:NEXT
2460 IFLO=1THENLO=0:GOTO2530
2470 GOSUB80:IFGO=0THEN2470
2480 GO=0
2490 IFX>258THENIFX<300ANDY>164ANDY<178THENGOSUB170:RETURN
2500 IFX>78THENIFX<140ANDY>164ANDY<178THENFORI=0TO4:NO$(I)=STRING$(46,32):NEXT:GOTO2450
2510 IFX>170THENIFX<232ANDY>164ANDY<178THEN2530
2520 GOTO2470
2670 GOSUB170:X=50:Y=50:X1=260:Y1=80:GOSUB3310
2680 PRINT@330," Notizen werden gedruckt !";
2681 LPRINT:LPRINT"          ";CHR$(14);"N O T I Z";CHR$(20):LPRINT:LPRINT"Datum: ";MID$(TIME$,4,2);".";MID$(TIME$,1,2);".19";MID$(T
IME$,7,2):LPRINT"Zeit : ";MID$(TIME$,10,8):LPRINT
2682 FORI=0TO4:LPRINTNO$(I):NEXT:LPRINT:LPRINTSTRING$(40,"-")
2690 FORI=1TO1000:NEXT
2700 GO=0:GOSUB170:RETURN
2710 X=80:X1=300:Y=40:Y1=150:GOSUB3310
2720 PRINT@275,"(Ge)nie (Gra)fik (S)ystem";
2730 PRINT@407,"(C)opyright  1989";
2740 PRINT@538,"Frank Schoof";
2750 PRINT@479,"by";
2760 PRINT@660,"Alle Rechte vorbehalten!";:GO=0
2770 PRINT@733,"Menue";
2780 X=170:Y=130:X1=210:Y1=142:GOSUB3540
2790 GOSUB80
2800 IFGO=0THEN2790
2810 IFX>169THENIFX<211ANDY>129ANDY<143THENGO=0:GOSUB170:RETURN
2820 GOTO2790
2980 X=140:Y=115:X1=235:Y1=130:GOSUB3540
2990 GOTO1050
3310 K=1
```



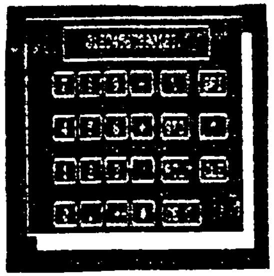
```
3320 XS=X:YS=Y
3340 BOX(X,Y,X1,Y1,1)
3360 LINK(X1+9,Y+9,X1+9,Y1+9,1)
3370 LINK(X1+9,Y1+9,X+9,Y1+9,1)
3380 FORX=X1+1TOX1+9:LINK(X,Y+9,X,Y1+9,-1):NEXT
3390 X=XS:Y=YS
3400 FORY=Y1+1TOY1+9:LINK(X+9,Y,X1,Y,-1):NEXT
3410 X=XS:Y=YS:RETURN
3420 GOSUB170
3430 IFLW=0THENDN=3
3440 IFLW=1THENDN=4
3450 IFLW=2THENDN=5
3460 CLOSE
3470 X=100:Y=60:X1=290:Y1=90:GOSUB3310
3471 GOTO5000
3480 GRY:RESUMENEXT
3490 IFPEEK(16710)=1THENRETURN
3500 FORI=0TO255:&LET"G":I:&LET"K":I:NEXT
3510 POKE16710,1:RETURN
3540 LINK(X,Y,X1,Y,-1)
3550 LINK(X1,Y,X1,Y1,-1):LINK(X1,Y1,X,Y1,-1):LINK(X,Y1,X,Y,-1)
3580 RETURN
3590 GOSUB170
3600 IFLW=0THENCMD"dir 0"ELSECMD"dir 1"
3601 T$=""
3610 T$=INKEY$:IFT$=""THEN3610
3620 GOTO1050
4160 GOSUB170
4500 RETURN
5000 DA$="":GOSUB4500:PRINT@402," Masken/Dateiname:";DD$(DN);:PRINT@420,"";:LINEINPUTDA$
5001 IFDA$=""THENDA$=DD$(DN)
5002 DD$(DN)=DA$
5003 OPEN"O",#1,"MASK/DAT":FORI=1TO3:PRINT#1,DD$(2+I):NEXT:CLOSE#1
5010 DM$=DA$+"/FLD"+LW$
5020 DA$=DA$+"/BAK"+LW$
5030 DD$="PMODULBE"
5040 GOTO40000
40000 MERGEDD$
40010 GOTO32140
41000 CMD"F",DELETE32000-36000
41010 CLEAR:RUN
42001 CMD"copy,0,1,,nfmt,ndmw"
42002 CMD"route,do"
50000 ' ************************
50010 ' *                      *
50020 ' *     GEDAT V 2.0      *
50030 ' *                      *
50040 ' *   (c)  F. Schoof     *
50050 ' *                      *
50060 ' ************************
```

RECHNER

Ein weiteres Unterprogramm unter **GEGRAS** ist der Rechner.

Er wird unter dem Menüpunkt **WERKZEUGE** mit **RECHNER** aufgerufen.

Es erscheint dann auf dem Monitor:



Mit den Zifferntasten können nun die Zahlen eingegeben werden. Die Eingabe funktioniert wie bei einem normalen Taschenrechner.

Tastenbelegung:

0	->	0
1	->	1
2	->	2
3	->	3
4	->	4
5	->	5
6	->	6
7	->	7
8	->	8
9	->	9
+	->	+
-	->	-
/	->	/
STERNCHEN	->	STERNCHEN (Multiplizieren)
@	->	PI
=	->	=
CLEAR	->	Löschen
S	->	STO (Speichern)
R	->	RCL (Laden)
%	->	%
ENTER	->	MENÜ

Als Beispiel:

Taste:	Bildschirm:
2	2
+	
22	22
=	24
@	3.14159
+	
1	1
=	4.14159

Und hier noch
einige diverse Dinge!

DIVERSES

Weitere Zusätze von GEGRAS sind die Programme LKOPIE, HKOPIE und MATHE/LIB.

Mit dem Programm LKOPIE/BAS kann eine DIN A4 Kopie des HRG-Speichers erstellt werden.

Beispiele dafür sind die Titelbilder.

Das Programm HKOPIE/BAS stellt eine Kopie (1 Nadel=1 Punkt) des HRG-Speichers her.

MATHE/LIB ist eine Mathematiklibary unter BASIC.

d.


```
10 '
20 ' **************************
30 ' *                        *
40 ' *     LKOPIE/BAS         *
50 ' *                        *
60 ' *  (c)   F. Schoof       *
70 ' *                        *
80 ' **************************
90 '
1000 DEFINTA-Z
1010 CLS
1020 LPRINT CHR$(27);"A";CHR$(2);CHR$(27);"2";
1030 FOR X=383 TO 0 STEP -1
1040      LPRINT CHR$(27);"Z";:OUT 253,192:OUT 253,6
1050      FOR Y=0 TO 191
1060              QDOT(X,Y,FA)
1070              IF FA=0 THEN GOSUB 1130
1080              IF FA=-1 THEN GOSUB 1170
1090      NEXT
1100      OUT 253,10
1110 NEXT
1120 END
1130 FOR I=0 TO 8
1140    OUT 253,0
1150 NEXT
1160 RETURN
1170 FOR I=0 TO 8
1180    OUT 253,3
1190 NEXT
1200 RETURN
```

```
1000 '
1010 ' ******************
1020 ' *                *
1030 ' *   HKOPIE/BAS   *
1040 ' *                *
1050 ' * (c) F. Schoof  *
1060 ' *                *
1070 ' ******************
1080 '
1090 DEFINT A-Z
1100 CLS
1110 LPRINT CHR$(27);"A";CHR$(1);CHR$(27);"2";
1120 FOR Y=0 TO 191
1130      LPRINT CHR$(27);"Z";CHR$(128);:OUT 253,1
1140      FOR X=0 TO 383
1150              QDOT(X,Y,FA)
1160              IF FA=0 THEN OUT 253,0
1170              IF FA=-1 THEN OUT 253,1
1180     NEXT
1190    OUT 253,10
1200 NEXT
1210 END
```

```
32000 '
32010 ' **************************
32020 ' *                        *
32030 ' *     Mathe-Libary       *
32040 ' *                        *
32050 ' *       (c) 1990         *
32060 ' *                        *
32070 ' *     Frank   Schoof     *
32080 ' *                        *
32090 ' **************************
32100 '
32140 ' Arcus Cosinus
32150 DEFFNACS(WW)=(3.14/2-ATN(WW/SQR(1-WW*WW)))*180/3.14
32160 ' Arcus Sinus
32170 DEFFNASN(WW)=ATN(WW/SQR(1-WW*WW))
32180 ' Addiere XX und YY
32190 DEFFNADD(XX,YY)=XX+YY
32200 ' Subtrahiere YY von XX
32210 DEFFNSUB(XX,YY)=XX-YY
32220 ' erhoehe XX um 1
32230 DEFFNINC(XX)=XX+1
32240 ' Verringere XX um 1
32250 DEFFNDEC(XX)=XX-1
32260 ' Dividiere XX durch YY
32270 DEFFNDIV(XX,YY)=XX/YY
32280 ' Multipliziere XX mit YY
32290 DEFFNMUL(XX,YY)=XX*YY
32300 ' Berechnet naechste Zahl
32310 DEFFNCEL(XX)=-INT(-XX)
32320 ' Negiert Vorzeichen
32330 DEFFNUMK(XX)=-XX
32340 ' Nachkommaanteil
32350 DEFFNPFRAC(XX)=XX-INT(XX)
32360 ' Negativer Nachkommaanteil
32370 DEFFNNFRAC(XX)=XX-INT(XX)-1
32380 ' Logarithmus XX zur Basis NN
32390 DEFFNLG(XX,NN)=LOG(XX)/LOG(NN)
32400 ' Reset der Division von YY durch XX
32410 DEFFNMOD(YY,XX)=YY-INT(YY/XX)*XX
32420 ' Differenz zwischen XX und YY
32430 DEFFNDFF(XX,YY)=ABS(XX-YY)
32440 ' Berechnet XX hoch YY
32450 DEFFNHOH(XX,YY)=XX[YY
32460 ' PI
32470 DEFFNPI=4*ATN(1)
32480 RETURN
```

HARDWARE * SOFTWARE

COMPUTER-CLUB 80

TANDY-/ BENIE- ANWENDER

ERFAHRUNGSAUSTAUSCH

SONDERINFO

HD 64180

Interne Ports

MMU

DMAC

Serielle Schnittstellen

Timer

Die internen Ports des HD64180

Helmut Bernhardt

01

Sämtliche gegenüber dem Z80 zusätzlich im HD64180 zur Verfügung gestellten Leistungen sind über Portadressen zu progammieren. Für diese internen Ports der CPU wird ein I/O-Adreßbereich von 3FH Ports verbraucht.

Wie prinzipiell auch schon beim Z80 kann mit dem HD64180 ein I/O--Adreßraum von 64K-Ports differenziert werden. Beim Z80 werden dafür ausschließlich die Befehle IN A,(C) und OUT (C),A sinnvoll sein. Bei diesen Befehlen werden die Inhalte der Register C (A0-A7) und B (A8-A15) als Adressen auf den Bus gelegt. Bei den normalen IN- und OUT-Befehlen mit Angabe einer 8Bit-Portadresse im Befehlsoperanden wird der Inhalt des Accu auf A8-A15 des Adreßbus gelegt. Diese Befehle sind in Systemen, deren I/O-Adreßdecoding auf 16Bit-Adressen aufbaut, nicht verwendbar.

Die internen Ports des HD64180 liegen in der Zero-Page (A8-A15 sind low) des 64K-I/O-Adreßraums. Um diese internen Ports zu lesen und zu schreiben, hat der HD64180 spezielle Befehle, die automatisch A8-A15 low ausgeben, wenn auf die interne Ports zugegriffen wird.

Traditionsbedingt decodieren aber sämtliche I/O-mapped Peripherie--Geräte für Z80-Systeme nur A0-A7 und beschränken den I/O-Adreßraum damit auf 256 Ports. Diese Baugruppen fühlen sich bei Zugriffen auf die internen Ports ebenfalls angesprochen, wenn ihre 8Bit-Adresse mit der Zero-Page-Adresse des angesprochenen internen Ports übereinstimmt. Beim Lesen eines internen Ports werden deshalb die Datentreiber des HD64180-Chips zur Umwelt geschlossen, so daß keine Kollisionen von internen und externen Daten möglich sind. Beim Schreiben gelangen aber die Daten, die für die internen Ports bestimmt sind, auch nach außen und werden von einem auf gleicher Adresse liegenden externen Port auch vereinnahmt. Verrückt spielende Peripherie-Geräte wären die Folge, wenn nicht noch eine zusätzliche Sicherung im HD64180 eingebaut wäre.

Der zusammenhängende Adreßbereich von 3FH Zero-Page-Ports für die Register des HD64180 beginnt default bei 00H. Durch Ändern der Bits 7 und 6 des ICR-Registers (3FH) lassen sich die 3FH Ports innerhalb der Zero-Page verschieben.

D7	D6	Adreßbereich der internen Ports
0	0	00 - 3F (nach RESET)
0	1	40 - 7F
1	0	80 - BF
1	1	C0 - FF

Die im folgenden benutzte Angabe von Portadressen geht davon aus, daß D6 und D7 von ICR beide low sind und die internen Ports bei 00-3F in der Zero-Page liegen. Die relativen Adressen und Funktionen sind in der Tabelle aus dem Prof180-Handbuch zu ersehen. Mit dieser Tabelle alleine ist man aber noch lange nicht in der Lage, die internen Ports sinnvoll zu programmieren. Vielleicht helfen die folgenden Ausführungen da weiter:

Zunächst seien die Angaben zum ICR-Register (I/O Control Register, Port 3FH) vervollständigt. D5 (IOSTP) = 0 ermöglicht den normalen Betrieb der HD64180-internen Peripherie-Ports (Einstellung nach RESET). Wenn IOSTP auf 1 gesetzt wird, sind die Funktionen von ASCI, CSI/O und PRT (serielle Schnittstellen und Reload-Timer) unterbunden.

02

Damit ein bei 9,216MHz Taktrate tickender HD64180 auch mit langsamer Peripherie laufen kann, gibt es nicht nur den WAIT-Pin, über den die Peripherie die CPU um Geduld bitten kann; die CPU (und auch die internen DMA-Kanäle) können auch selbständig WAIT-Zyklen einlegen, wobei zwischen memory-mapped und I/O-mapped Zugriffen unterschieden wird. Mit dem oberen Nibble von Port 32H (DCNTL) lassen sich unterschiedliche Anzahlen WAITs einstellen:

D7	D6	Anzahl WAITs bei memory mapped Adressierung
0	0	0
0	1	1
1	0	2
1	1	3 (nach RESET)

		Anzahl WAITs bei	
D5	D4	I/O-Zugriffen	/INT0-Acknowledge
0	0	1	2
0	1	2	4
1	0	3	5
1	1	4	6 (nach RESET)

Bei Zugriff auf die internen Ports werden unabhängig von D5 und D4 keine WAITs eingefügt. Nur für die Datenregister von ASCI, CSI/O und PRT werden je nach Zustand bis zu 4 WAITs für die Synchronisation mit der CPU eingefügt.

Die Bits 0-3 von DCNTL dienen der Steuerung des DMA-Betriebs und werden dort behandelt.

RCR, Refresh Control Rgister, Port 36H

Gegenüber dem Hidden Refresh des Z80 während jeden M1-Zyklus' gibt der HD64180 jeweils nach einer bestimmten Anzahl von Taktzyklen eine 8Bit--Refreshadresse aus. Durch die 8Bit-Adresse können 256K-Bit- und 1M--Bit-Speicher-Chips ohne zusätzliche Refresh-Logik verwendet werden. Ein Refresh erfolgt asynchron zur CPU-Arbeit nach einer über RCR, D0 und D1 eingestellten Anzahl von Taktzyklen:

D1	D0	Anzahl Takte zwischen zwei Refresh-Zyklen
0	0	10 (nach RESET)
0	1	20
1	0	40
1	1	80

Mit D7=0 an RCR wird der Refresh-Controller abgeschaltet (für statische RAMs kann das Refreshing entfallen. D7=1 (nach RESET) aktiviert den Refresh-Controller.

Mit D6 läßt sich das Refresh Timing den Möglichkeiten des Speichers anpassen:

D6 = 0 : Refresh ist 2 Taktzyklen lang
D6 = 1 : Refresh ist 3 Taktzyklen lang (nach RESET voreingestellt)

D2-D5 im RCR-Register haben keine Funktion.

ICT, Interrupt/Trap Register, Port 34H

D7 (Trap) wird immer dann von der CPU auf 1 gesetzt, wenn diese auf einen illegalen Befehl gestoßen ist (z.B. undefinierte Z80-Befehle). Durch einen OUT0-Befehl kann D7 wieder auf 0 gesetzt werden (nicht aber auf 1). Nach RESET ist D7=0.

Ein illegaler OP-Code löst den für den HD64180 höchst priorisierten Trap-Interrupt aus. Die Service-Routine kann dann aus dem Zustand von D6 (UFO) erkennen, ob der Fehler im 2ten oder 3ten Byte des OP-Codes auftrat.

D6 erlaubt die Korrektur des beim Trap auf den Stack geretteten Program-Counters. Bei D6=0 zeigt der gerettete PC auf das zweite Byte des als falsch erkannten Befehls, bei D6=1 zeigt der PC auf das 3 Byte des faschen Befehls. D6 kann nur gelesen werden.

D0, D1 und D2 können zur Maskierung der externen Interrupts /INT0, /INT1 und /INT2 verwendet werden. Wenn das entsprechende Bit auf 0 gesetzt ist, ist der Interrupt gesperrt; auf 1 gesetzt wird der entsprechende Interrupt freigegeben. Nach RESET ist nur /INT0 freigegeben (D0=1) und /INT1 und /INT2 sind gesperrt (D1,D2=0).

IL, Interrupt Vector Low Register, Port 33H

Die externen Interrupts /INT0 im Mode 2 sowie /INT1 und /INT2 grundsätzlich und alle internen Interrupts werden von Service-Routinen bedient, deren Startadressen in einer Tabelle abgelegt sind. Die Technik ist die des IM2 beim Z80. Im I-Register der CPU werden die oberen 8 Bit (A8-A15) der Adresse der Vektor-Tabelle abgelegt. Die unteren 8 Bit der Adresse (Pointer in die Vektor-Tabelle) liefert das Interrupt-auslösende Peripherie-Gerät beim Interrupt-Acknowledge über den Datenbus. Die CPU ignoriert davon D0, um eine gerade Adresse zu erzeugen, und bildet aus den beiden Hälften einen Pointer in die Vektor-Tabelle. Die dort in der Tabelle abgelegte Adresse ist die Startadresse der Service-Routine.

Beim HD64180 benötigt nur der /INT0, der dem /INT des Z80 im Mode 2 entspricht, einen extern gelieferten Adreßanteil (A1-A7 über D1-D7 des Datenbus), um in der Vektor-Tabelle die Adresse der Service-Routine zu finden. Damit können abwechselnd mehrere INT-Qellen über nur einen Interrupt-Eingang der CPU angeschlossen werden.

Die Adressen der übrigen Interrupt-Quellen des HD64180 sind (mit gewissen Freiheiten) festgelegt.

So stellen die ersten beiden Bytes der Tabelle (Adresse mit A0-A7 = 0) die Adresse mit der Service-Routine für /INT1 dar. Die nächsten beiden Bytes liefern den Vektor für /INT2. Die Reihenfolge der Vektoren ist durch die Interrupt-Quellen festgelegt; es läßt sich aber die relative Lage der Gruppe von Vektoren (32 Bytes) innerhalb der 256-Bytes-Tabelle in 32-Bytes-Schritten verschieben. Wo diese Adressengruppe relativ in der Tabelle liegt, wird durch D7-D5 des IL-Registers vorgegeben (nach RESET sind diese low und die Gruppe liegt am Anfang der Vektor-Tabelle).

Wenn D7-D5 alle low sind, liegen die Vektoren für /INT1, /INT2 und die internen Interrupts ab Adresse XX00H in der Tabelle; wenn D7-D5 das Bitmuster 001 haben, liegen diese Vektoren ab XX20H in der Tabelle; u.s.w.

Die Bits 0-4 werden von der internen Interrupt-Quelle selbst erzeugt und geben die relative Adresse innerhalb der 32Byte-Gruppe in der Tabelle vor. Die Tabelle 2.7.2 des HD64180-Handbuchs (die hier ohne freundliche Genehmigung von Hitachi wiedergegeben ist) gibt die Reihenfolge der Einträge für die Interrupt-Quellen und die damit verknüpfte Prioritäts-Hirarchie vor.

Table 2.7.2 Interrupt Source and Lower Vector

Interrupt-Source	Priority	IL b7	IL b6	IL b5	Fixed Code b4	b3	b2	b1	b0
/INT1	highest	.	.	.	0	0	0	0	0
/INT2		.	.	.	0	0	0	1	0
PRT channel 0		.	.	.	0	0	1	0	0
PRT channel 1		.	.	.	0	0	1	1	0
DMA channel 0		.	.	.	0	1	0	0	0
DMA channel 1		.	.	.	0	1	0	1	0
CSI/O		.	.	.	0	1	1	0	0
ASCI channel 0		.	.	.	0	1	1	1	0
ASCI channel 1	lowest	.	.	.	1	0	0	0	0

Außer den maskierten Interrupts gibt es noch den oben erwähnten Trap, der eine Verzweigung zur logischen Adresse 0000H zur Folge hat. Wenn diese logische Adresse mit der physikalischen Adresse 00000H identisch ist (Banking), entspricht dies dem /RESET-Vektor. Durch Prüfen von D7 in Port 32H (ICT) kann erkannt werden, ob ein Trap oder ein RESET vorliegt.

Ein /NMI bewirkt wie beim Z80 einen CALL 66H, wobei auch hier die logische Adresse innerhalb der eingestellten Bank und nicht unbedingt die absolute Adresse 00066H relevant ist.

Der /INT0 kennt wie der /INT des Z80 noch die Modi 0 und 1.

Mode 0 : Beim Acknowledge wird ein 1Byte-Befehl vom Datenbus gelesen und ausgeführt (RST XX). Dieses Byte muß die Interrupt-Quelle liefern.

Mode 1 : Bei einem Interrupt am /INT0-Pin erfolgt ein RST 38H. Die Adresse 0038H liegt in der per MMU eingestellten Bank (logische Adresse).

Weitere Steuerungen der internen Interrupts werden bei der Beschreibung der jeweiligen Funktionsgruppen gezeigt. Dazu aber erst im nächsten Info.

Table 2.5.1 Internal I/O Register Address Map (1)

	Register	Mnemonic	Address	
			Binary	Hexadecimal
ASCI	ASCI Control Register A Ch 0	CNTLA0	XX000000	00H
	ASCI Control Register A Ch 1	CNTLA1	XX000001	01H
	ASCI Control Register B Ch 0	CNTLB0	XX000010	02H
	ASCI Control Register B Ch 1	CNTLB1	XX000011	03H
	ASCI Status Register Ch 0	STAT0	XX000100	04H
	ASCI Status Register Ch 1	STAT1	XX000101	05H
	ASCI Transmit Data Register Ch 0	TDR0	XX000110	06H
	ASCI Transmit Data Register Ch 1	TDR1	XX000111	07H
	ASCI Receive Data Register Ch 0	RDR0	XX001000	08H
	ASCI Receive Data Register Ch 1	RDR1	XX001001	09H
CSI/O	CSI/O Control Register	CNTR	XX001010	0AH
	CSI/O Transmit/Receive Data Register	TRDR	XX001011	0BH
Timer	Timer Data Register Ch 0L	TMDR0L	XX001100	0CH
	Timer Data Register Ch 0H	TMDR0H	XX001101	0DH
	Reload Register Ch 0L	RLDR0L	XX001110	0EH
	Reload Register Ch 0H	RLDR0H	XX001111	0FH
	Timer Control Register	TCR	XX010000	10H
	Reserved		XX010001	11H
			⁞	⁞
			XX010011	13H
	Timer Data Register Ch 1L	TMDR1L	XX010100	14H
	Timer Data Register Ch 1H	TMDR1H	XX010101	15H
	Reload Register Ch 1L	RLDR1L	XX010110	16H
	Reload Register Ch 1H	RLDR1H	XX010111	17H
Others	Free Running Counter	FRC	XX011000	18H
	Reserved		XX011001	19H
			⁞	⁞
			XX011111	1FH
DMA	DMA Source Address Register Ch 0L	SAR0L	XX100000	20H
	DMA Source Address Register Ch 0H	SAR0H	XX100001	21H
	DMA Source Address Register Ch 0B	SAR0B	XX100010	22H
	DMA Destination Address Register Ch 0L	DAR0L	XX100011	23H
	DMA Destination Address Register Ch 0H	DAR0H	XX100100	24H
	DMA Destination Address Register Ch 0B	DAR0B	XX100101	25H
	DMA Byte Count Register Ch 0L	BCR0L	XX100110	26H
	DMA Byte Count Register Ch 0H	BCR0H	XX100111	27H
	DMA Memory Address Register Ch 1L	MAR1L	XX101000	28H
	DMA Memory Address Register Ch 1H	MAR1H	XX101001	29H
	DMA Memory Address Register Ch 1B	MAR1B	XX101010	2AH
	DMA I/O Address Register Ch 1L	IAR1L	XX101011	2BH
	DMA I/O Address Register Ch 1H	IAR1H	XX101100	2CH
	Reserved		XX101101	2DH
	DMA Byte Count Register Ch 1L	BCR1L	XX101110	2EH
	DMA Byte Count Register Ch 1H	BCR1H	XX101111	2FH
	DMA Status Register	DSTAT	XX110000	30H
	DMA Mode Register	DMODE	XX110001	31H
	DMA/WAIT Control Register	DCNTL	XX110010	32H
INT	IL Register (Interrupt Vector Low Register)	IL	XX110011	33H
	INT/TRAP Control Register	ITC	XX110100	34H
	Reserved		XX110101	35H

	Register	Mnemonic	Address	
			Binary	Hexadecimal
Refresh	Refresh Control Register	RCR	XX110110	36H
	Reserved		XX110111	37H
MMU	MMU Common Base Register	CBR	XX111000	38H
	MMU Bank Base Register	BBR	XX111001	39H
	MMU Common/Bank Area Register	CBAR	XX111010	3AH
I/O	Reserved		XX111011	3BH
			⁞	⁞
			XX111110	3EH
	I/O Control Register	ICR	XX111111	3FH

Die seriellen Schnittstellen (ASCI und CSI/O) des HD64180
von Gerald Schröder

Der HD64180 stellt drei serielle Schnittstellen zur Verfügung. Bei zweien handelt es sich um "Asynchronous Serial Communication Interfaces" (wie RS-232/V24) und die dritte heißt "Clocked Serial Input/Output".

Asynchronous Serial Communication Interfaces

Aufgabe dieser beiden Schnittstellen ist die serielle Verbindung zu einem Modem, Terminal, (RS-232-)Drucker o.ä. Zusätzlich gibt es einen "Multiprozessor-Modus", mit dem sich leicht ein System mit mehreren HD64180s aufbauen ließe, wobei die ASCIs dann als (Verbindung zum) System-Bus dienen würden.

Jeder Kanal belegt 5 Ports: einen zum Lesen und einen zum Schreiben der Daten, einen für den Status (Zustand) des Kanals, zwei für die Einstellung des Kanals (Baudrate usw.) und einige Zusatz-Informationen. Die Kanäle unterscheiden sich in einigen Features, was ich aber jeweils anmerken werde.

Die einzelnen Ports (Kanal 0 und Kanal 1):

Name	Port	Bezeichnung
CNTLA0	0	Control Register A Channel 0
CNTLB0	2	Control Register B Channel 0
STAT0	4	Status Register Channel 0
TDR0	6	Transmit Data Register Channel 0
RDR0	8	Receive Data Register Channel 0
CNTLA1	1	Control Register A Channel 1
CNTLB1	3	Control Register B Channel 1
STAT1	5	Status Register Channel 1
TDR1	7	Transmit Data Register Channel 1
RDR1	9	Receive Data Register Channel 1

Die einzelnen Bits der Ports:

CNTLA (Control A)

7	6	5	4	3	2	1	0
MPE	RE	TE	Kanal 0: RTS0* Kanal 1: CKA1D	Read: MPBR Write: EFR	MOD2	MOD1	MOD0

MPE = Multiprocessor Mode Enable
RE = Receiver Enable
TE = Transmitter Enable
RTS0* = Request To Send Channel 0 (aktiv LOW, nur Kanal 0!)
CKA1D = Asynchronous Clock Channel 1 Disable (nur Kanal 1!)
MPBR = Multiprocessor Bit Receive (nur beim Lesen!)
EFR = Error Flag Reset (nur beim Schreiben!)
MODx = ASCII Data Format Mode

Anmerkung: Bit 4 (D4) hat bei Kanal 0 eine andere Bedeutung als bei Kanal 1. Bit 3 (D3) bedeutet beim Schreiben etwas ganz anderes als beim Lesen.

CNTLB (Control B)

7	6	5	4	3	2	1	0
MPBT	MP	Read: CTS* Write: PS	PEO	DR	SS2	SS1	SS0

MPBT = Multiprocessor Bit Transmit
MP = Multiprocessor Mode
CTS = Clear To Send (nur beim Lesen!)
PS = Prescale (nur beim Schreiben!)
PEO = Parity Even/Odd
DR = Divide Ratio
SSx = Source/Speed Select

STAT (Status)

7	6	5	4	3	2	1	0
RDRF	OVRN	PE	FE	RIE	Kanal 0: DCD0* Kanal 1: CTS1E	TDRE	TIE

RDRF = Receive Data Register Full (nur lesbar)
OVRN = Overrun Error (nur lesbar)
PE = Parity Error (nur lesbar)
FE = Framing Error (nur lesbar)
RIE = Receive Interrupt Enable
DCD0* = Data Carrier Detect (aktiv LOW, nur lesbar, nur Kanal 0!)
CTS1E = Clear To Send Channel 1 Enable (nur Kanal 1!)
TDRE = Transmit Data Register Empty (nur lesbar)
TIE = Transmit Interrupt Enable

Einstellung der Kanäle

Über CNTLB wird die Baudrate, der Multiprozessor-Mode und die Parität festgelegt. D0 bis D3 und D5 sind für die Baudrate zuständig. Sie legen fest, durch welche Werte die Taktfrequenz geteilt wird.

D5	1.Teiler	D3	2.Teiler	D2	D1	D0	3.Teiler
0	10	0	16	0	0	0	1
1	30	1	64	0	0	1	2
				0	1	0	4
				0	1	1	8
				1	0	0	16
				1	0	1	32
				1	1	0	64
				1	1	1	externer Takt!

Der externe Takt wird von den Pins CKA0 (gemultiplext mit DREQ0*) bzw. CKA1 (gemultiplext mit TEND0*) abgenommen, wobei sich die Baudrate aus dem externen Takt geteilt durch 16 oder 64 (abhängig von D3) ergibt. Dies ist der Reset-Zustand. Falls die Baudrate vom System-Takt abgeleitet wird, wird der resultierende Takt über CKA0/1 ausgegeben!

Zwei Rechenbeispiele (bei einem Systemtakt von 9.216 Mhz):
19200 Baud: 19200 = 9.216.000 DIV 30 DIV 16 DIV 1 => D5-D0 = 1x0000
300 Baud: 300 = 9.216.000 DIV 30 DIV 64 DIV 16 => D5-D0 = 1x1100

D4 ist für die Art der Parität zuständig, falls Paritäts-Prüfung gewünscht wird: D4=0: gerade Parität (EVEN); D4=1: ungerade Parität (ODD).

Zu guter Letzt D6 und D7, die für den Multiprozessor-Modus zuständig sind: Wenn D6=1, dann wird statt des Paritäts-Bits ein Multiprozessor-Bit gesendet/empfangen. Wenn D7=1, wird beim Senden das Multiprozessor-Bit auf 1 gesetzt. Das könnte z.B. bedeuten: "Achtung, an alle anderen Prozessoren, es kommt was für alle!"

Weitere Einstellungen erfolgen über CNTLA. Mit D2 bis D0 wird das Datenformat festgelegt:

Bit	bedeutet
D2=0	7-Bit-Daten
=1	8-Bit-Daten
D1=0	keine Parität
=1	Parität
D0=0	1 Stop-Bit
=1	2 Stop-Bits

Wenn D3 mit 0 beschrieben wird, werden alle Fehler-Meldungen für diesen Kanal (im Register STAT0/1) gelöscht. Mit D4 kann bei Kanal 0 der Pin RTS0* gesetzt/gelöscht werden (Request To Send = Gegenseite darf senden). Bei Kanal 1 hingegen erfolgt hier die Auswahl, ob der Pin CKA1/TEND0* als Takteingang (CKA1, D4=0) oder als Kennung für das Ende eines Blocktransfers (TEND0*, D4=1) benutzt wird.

Mit D5 wird der Sender ein/ausgeschaltet, mit D6 der Empfänger. D7 hat nur im Multiprozessor-Modus eine Funktion. Wenn D7=1, werden nur Daten mit gesetztem MPB-Bit (s. D7 in CNTLB) angenommen, z.B. im Modus: "Empfange nur Daten für alle Prozessoren!". Bei D7=0 werden alle Daten angenommen. So ist folgende Kommunikation denkbar: Der Master-Prozessor sendet an alle Prozessoren (durch gesetztes MPB-Bit gekennzeichnet), welche beiden Prozessoren sich unterhalten dürfen. Diese schalten dann D7 auf 0 und wickeln ihre Kommunikation ohne MPB-Bit ab. Dann schalten sie D7 auf 1 und der Master-Prozessor wird benachrichtigt, die Sache wieder zu übernehmen.

Zu guter Letzt noch die Einstellungen, die über das Status-Register STAT vorgenommen werden. D0 legt fest, ob ein Interrupt erfolgt, wenn der Kanal sendebereit ist. Wenn D3=1, wird auch ein Interrupt ausgelöst, wenn ein Zeichen empfangen wurde oder ein Fehler auftrat. Bei Kanal 1 kann über D2 angewählt werden, ob der Pin CTS1*/RXS als CTS* (D2=1, Clear To Send = HD64180 darf senden) oder als Daten-Empfangs-Pin für CSI/O (D2=0) dient.

Senden/Empfangen

Der Status des Kanals kann über STAT abgefragt werden. D1=1 bedeutet, daß Daten gesendet werden dürfen. Wenn dann ein Byte in das Sende-Register TDR geschrieben wird, geht D1 auf 0, bis das Zeichen gesendet wurde.

D7=1 bedeutet, daß ein Zeichen empfangen wurde und jetzt aus dem Empfangs-Register RDR gelesen werden kann oder daß beim Empfangen ein Fehler aufgetreten ist. Nach dem Lesen des Bytes aus RDR wird D7 gelöscht. Folgende Fehler werden erkannt:

D6 Overrun (letztes Byte wurde nicht schnell genug aus RDR geholt)
D5 Parity (das Paritäts-Bit des Zeichens war falsch)
D4 Framing (das/die Stop-Bits stimmten nicht)

Falls ein Fehler auftrat, muß dieser gelöscht werden, indem D3 von CNTLA mit 0 beschrieben wird. Achtung: dabei werden dann auch die anderen Einstellungen von CNTLA geändert!

Über D2 aus STAT kann bei Kanal 0 noch der Zustand des Signals "Data Carrier Detect" abgefragt werden. D2 wird auf 1 gesetzt, wenn das Signal am Pin DCD0* irgendwann mal HIGH war oder noch ist. D2 wird nach dem Lesen auf 0 gesetzt, wenn inzwischen DCD0*=LOW ist. Solange D2=1, wird nichts empfangen. Diese Verbindung ist nur bei bestimmten Modems sinnvoll.

Über D3 aus CNTLA kann das Multiprozessor-Bit des letzten empfangenen Bytes abgefragt werden. D5 aus CNTLB gibt Auskunft über den Zustand des Pins CTS* Für Kanal 0 gilt, daß der Kanal erst sendebereit wird (STAT0, D1=1), wenn der Pin CTS0*=0. CTS1* kann nur abgefragt werden und hat nur einen Einfluß auf die Sendebereitschaft, wenn wenn D2 aus STAT1 auf 1 gesetzt wurde (CTS1E = CTS-Enable Kanal 1).

Clocked Serial Input/Output

Diese Schnittstelle dient zur Verbindung mehrerer Prozessoren, wobei die anderen Prozessoren nicht unbedingt HD64180s sein müssen. Aber sie müssen über eine gemeinsame (synchronisierte) Taktquelle verfügen und die Sende-/Empfangs-Operation muß in beiden Prozessoren gleichzeitig beginnen!

Der HD64180 kann nicht gleichzeitig empfangen und senden. Es gibt nur ein Sende-/Empfangsregister: TRDR (Transmit/Receive Data Register), Port 0Bh. Es sind auch nur wenige Einstellungen möglich, so daß nur ein Kontroll-Register nötig ist: CNTR (Control), Port 0Ah. Der Aufbau dieses Registers:

CNTR (Control)

7	6	5	4	3	2	1	0
EF	EIE	RE	TE	-	SS2	SS1	SS0

EF = End Flag (nur lesbar)
EIE = End Interrupt Enable
RE = Receive Enable
TE = Transmit Enable
SSx = Speed Select

Einstellungen

Es kann nur die Baudrate eingestellt werden. Wenn die Baudrate neu gesetzt wird, sollte nicht gleichzeitig ein Sende-/Empfangs-Zyklus (s.u.) gestartet werden. Der System-Takt wird durch den folgenden Wert geteilt (hier für 9.216 Mhz):

SS2	SS1	SS0	Teiler	ergibt Baudrate
0	0	0	20	460800
0	0	1	40	230400
0	1	0	80	115200
0	1	1	160	57600
1	0	0	320	28800
1	0	1	640	14400
1	1	0	1280	7200
1	1	1	externer Takt! (kleiner als 460800)	

Bei 111 wird der externe Takt wird über den Pin CKS eingelesen; ansonsten wird der resultierende Takt hier ausgegeben.

Empfangen/Senden

Zuerst muß abgefragt werden, ob der Sender gerade aktiv ist. Erst wenn D7 aus CNTR gesetzt ist, kann die Operation beginnen. Zum Senden wird das Byte in TRDR geschrieben und dann D4 in CNTR gesetzt. Gleichzeitig wird die Baudrate (wieder) gesetzt. Sie sollte aber nicht verändert werden, da es sonst Sende-Probleme geben kann. Die Operation ist erst beendet, wenn D7 wieder auf 1 geht. Falls D6 gesetzt wurde, wird dann ein Interrupt ausgelöst.

Beim Empfangen wird D5 gesetzt (zusammen mit der Baudrate). Nun beginnt der Empfangs-Vorgang, der immer gleich lang dauert. Wenn D7 wieder auf 1 geht, kann das Byte aus TRDR gelesen werden. Wenn vor dem Empfangen D6 gesetzt wurde, wird ein Interrupt ausgelöst, sobald D7 auf 1 springt. Es ist zu beachten, daß der Empfangs-Pin (RXS) mit CTS* für Kanal 1 gemultiplext ist. Vor dem Empfang muß also über D2 von STAT1 die Funktion "Pin RXS/CTS1* als RXS" ausgewählt werden.

Der Zähler TMDR darf zwischendurch ausgelesen werden, aber nur in der Reihenfolge Low-Byte und dann High-Byte. Sobald TMDR den Zählerstand 0000 erreicht, wird der in RLDR stehende Wert in TMDR geladen und es wird wieder mit dem Zählen angefangen. Außerdem wird TIF in TCR gesetzt. Falls vorher TIE gesetzt wurde, wird auch ein Interrupt ausgelöst.

TIF wird erst gelöscht, wenn zuerst (!) TCR und dann TMDR (in der Reihenfolge Low-/High-Byte) gelesen werden. Ansonsten bleibt es gesetzt, was bei der Rückkehr von einer Interrupt-Routine sofort einen neuen Interrupt verursacht!

Die Timer des HD64180

von Gerald Schröder

Der HD64180 verfügt über zwei programmierbare 16-Bit-Zähler und einen nicht-programmierbaren 8-Bit-Zähler. Der 8-Bit-Zähler wird alle 10 Takte dekrementiert und kann nur gelesen werden, und zwar über Port 18h. Das Beschreiben dieses Ports kann Fehler im Timing des HD64180 verursachen.

Die 16-Bit-Zähler werden alle 20 Takte dekrementiert. Einer der 16-Bit-Zähler kann über den Pin TOUT an einen anderen Chip melden, daß er den Zählerstand 0000 erreicht hat. Beide Zähler können einen Interrupt auslösen, wenn sie 0000 erreichen.

Die Ports

Port	Abk.	Name
0Ch	TMDR0L	Timer Data Register Channel 0 Low
0Dh	TMDR0H	Timer Data Register Channel 0 High
0Eh	RLDR0L	Reload Data Register Channel 0 Low
0Fh	RLDR0H	Reload Data Register Channel 0 High
10h	TCR	Timer Control Register
14h	TMDR1L	Timer Data Register Channel 1 Low
15h	TMDR1H	Timer Data Register Channel 1 High
16h	RLDR1L	Reload Data Register Channel 1 Low
17h	RLDR1H	Reload Data Register Channel 1 High

TCR (Control)

0	1	2	3	4	5	6	7
TDE0	TDE1	TOC0	TOC1	TIE0	TIE1	TIF0	TIF1

TDE0/1 Timer Decrement Enable Channel 0/1
TOC0,1 Timer Output Control (beide für Kanal 1!)
TIE0/1 Timer Interrupt Enable Channel 0/1
TIF0/1 Timer Interrupt Flag Channel 0/1

Programmierung

Zuerst sollte der Timer gestoppt werden indem TDE in TCR auf 0 gesetzt wird. Dann das gewünschte Zählintervall in RLDR und (!) TMDR eintragen; die Eintragung in TMDR ist nötig, weil der Zähler sonst bei dem alten Wert startet. Es handelt sich hier um 16-Bit-Werte, so daß bei einem Takt von 9,216 Mhz Warteschleifen von ca. 2,2 Microsekunden bis ca. 0,142 Sekunden realisiert werden können; bzw. der Zähler erreicht 460800 (Intervall 0001) bis 7,03125 (Intervall FFFFh) Mal in der Sekunde den Stand 0000.

Der Zähler fängt erst an zu zählen, wenn TDE in TCR auf 1 gesetzt wird. Gleichzeitig wird über TIE ausgewählt, ob ein Interrupt ausgelöst werden soll, wenn TMDR den Zählerstand 0000 erreicht. Für Kanal 1 kann über TOC noch ausgewählt werden, welche Reaktion am Pin TOUT (Timer Output) beim Erreichen von 0000 erfolgt (Achtung: TOUT ist mit Adreßbit A18 gemultiplext):

TOC0	TOC1	Reaktion TOUT
0	0	keine (Pin als A18)
0	1	TOUT wird getoggelt (0 -> 1, 1 -> 0)
1	0	TOUT auf 0
1	1	TOUT auf 1

Die zusätzlichen Befehle des HD64180

Helmut Bernhardt

Wer in seinem Computer den Z80 durch einen HD64180 ersetzt hat, wird sicher schon ein plötzliches Booten oder einen Absturz beim Aufruf von bestimmten Programmen bemerkt haben, die vorher mit dem Z80 einwandfrei zusammengearbeitet haben. Zu solchen Programmen zählen REZILOG unter CP/M oder ZEUS unter NEWDOS/80. Der Fehler liegt nicht am Prozessor sondern am Programm. Der Programmierer hat dort Befehle benutzt, die von Zilog nicht dokumentiert sind aber trotzdem auf dem Z80 zur Verfügung stehen. Es sind dies die Illegals mit Doppelbyte-Op--codes.

Hitachi hat nun andere Vorstellungen entwickelt, wie diese freien Opcodes verwendet werden können, und eine Reihe dieser Codes den im Anhang zusammengestellten Befehlen zugeordnet. Gegenüber den Illegals des Z80 sind das sinnvolle Ergänzungen, die der Architektur des HD64180 entgegenkommen und u.a. eine elegante Programmierung der internen Ports der CPU ermöglichen. Die ebenfalls hinzugekommene Multiplikation ist leider nur auf 8Bit-Faktoren beschränkt geblieben.

Der wichtigste Befehl ist OTIMR, der eine Anzahl von Bytes, die in B vorgegeben wird, ab einer Adresse im Speicher, auf die HL zeigt, an aufeinander folgende interne Zeropage-Ports ausgibt, wobei die Adresse des ersten Ports in C vorzugeben ist. Damit kann z.B. der DMA-Controller schnell mit Parametern für eine Übertragung versorgt werden.

In mc 1/87, S.74 wurde gezeigt, wie die Illegals durch normale Befehle ersetzt werden können. Danach kann man von Hand entsprechende Programme auf normale Befehle umstellen. Was der Normalverbraucher zu Fuß macht, läßt der Experte die CPU zur Laufzeit des Illegal-verseuchten Programms selbst machen. Eventuell schreibt Gerald dazu nochmal was.

Bei Erreichen eines unbekannten Opcodes wird der höchstpriorisierte TRAP-Interrupt ausgeführt, der einem CALL 0000H gleichkommt. Er ist mit einem RESET vergleichbar, nur daß kein richtiges /RESET-Signal erfolgt, das die gesamte Hardware zurücksetzt. Dadurch wird auch kein Floppy-Controller in den Einschaltzustand zurückgesetzt und auch kein ausgeblendetes EPROM wieder eingeschaltet.

Bei 0000H liegt dann RAM vor. Dort kann bei der Initialisierung ein Sprung auf die TRAP-Service-Routine gepatcht werden. Diese Routine kann mit dem Stack und den Flags im TRAP-Register herausfinden, ob ein TRAP die Ursache des Aufrufs war, oder ob die Adresse 0000H durch Software aufgerufen wurde; bei einem Trap kann sie ermitteln, welches Byte im Programm den TRAP auslöste und dem entsprechend alternative normale Befehle abarbeiten, und anschließend wieder die Kontrolle an das unterbrochene Programm zurückgeben.

Wenn eine solche TRAP-Service-Routine nicht vorliegt, bedeutet der CALL 0000H einen Warmstart unter CP/M oder einen Absturz unter NEWDOS/80, weil die BOOT-Routine im ROM nicht den Floppy-Controller auf Single Density zurücksetzt (deshalb funktioniert beim GENIE auch der Befehl BOOT oder ein JP 0000H nicht, wenn mit einem Doubler gearbeitet wird; der TRS 80 decodiert die Adresse 0000H und erzeugt daraus ein /RESET-Signal).

Natürlich funktioniert der TRAP-Interrupt nur bei den Z80-Illegals, für die der HD64180 keine neuen Befehle eingeführt hat. Wenn ein illegaler Opcode auftaucht, der einem HD64180-Befehl entspricht, so wird dieser Befehl nach Hitachi's Vorstellung ordnungsgemäß ausgeführt. Der Erfolg ist dann aber nicht der, den sich der Illegal--Programmierer einst gedacht hat.

Zusätzliche Befehle des HD64180

Mutiplikation

ED 4C	MULT	BC	;BC := B * C	17 Taktzyklen
ED 5C	MULT	DE	;DE := D * E	
ED 6C	MULT	HL	;HL := H * L	
ED 4C	MULT	SP	;SP := SPhigh * SPlow	

Logisches AND zum Setzen der Flags ohne die Register zu verändern

ED 04	TEST	B	; A and B	7 Taktzyklen
ED 0C	TEST	C	; A and C	
ED 14	TEST	D	; A and D	
ED 1C	TEST	E	; A and E	
ED 24	TEST	H	; A and H	
ED 2C	TEST	L	; A and L	
ED 3C	TEST	A	; A and A	
ED 34	TEST	(HL)	; A and (HL)	10 Taktzyklen
ED 64 nn	TEST	nn	; A and nn	9 Taktzyklen
ED 74 pp	TESTIO	pp	; (C) and pp	12 Taktzyklen

Zeropage I/O-Befehle

ED 00 pp	INO	B,(pp)	; B := (pp)	12 Taktzyklen
ED 08 pp	INO	C,(pp)	; C := (pp)	
ED 10 pp	INO	D,(pp)	; D := (pp)	
ED 18 pp	INO	E,(pp)	; E := (pp)	
ED 20 pp	INO	H,(pp)	; H := (pp)	
ED 28 pp	INO	L,(pp)	; L := (pp)	
ED 30 pp	INO	F,(pp)	; S,Z,P <- (pp)	
ED 38 pp	INO	A,(pp)	; A := (pp)	
ED 01 pp	OUTO	(pp),B	; (pp) := B	13 Taktzyklen
ED 09 pp	OUTO	(pp),C	; (pp) := C	
ED 11 pp	OUTO	(pp),D	; (pp) := D	
ED 19 pp	OUTO	(pp),E	; (pp) := E	
ED 21 pp	OUTO	(pp),H	; (pp) := H	
ED 29 pp	OUTO	(pp),L	; (pp) := L	
ED 39 pp	OUTO	(pp),A	; (pp) := A	

ED 83	OTIM	; (C):=(HL), HL:=HL+1, C:=C+1, B:=B-1	14 Taktzyklen
ED 93	OTIMR	; (C)..(C+B) := (HL)..(HL+B), B:=0 ; oder: (C):=(HL), C:=C+1, HL:=HL+1, B:=B-1 ; wiederhole bis B=0	B*16 Taktzyklen
ED 8B	OTDM	; (C):=(HL), HL:=HL-1, C:=C-1, B:=B-1	14 Taktzyklen
ED 9B	OTDMR	; (C)..(C-B) := (HL)..(HL-B), B:=0 ; oder: (C):=(HL), C:=C-1, HL:=HL-1, B:=B-1	B*16 Taktzyklen
ED 76	SLP	; Sleep: Einstellen aller Aktivitäten incl. ; RAM-Refresh, bis /RESET oder /INT eintritt	

Soppsches

Neuer Dreitasten-Befehl ";./"
Die Library vergrößern
Mehrere Sysfiles gleichzeitig
Neuer DOS-Befehl OUT port#, xx
SYS-Files und wie man sie macht
Die Library-Befehle des G-DOS
Die Records handhaben
BASIC frei im RAM verlegen
PUT TO adresse - ein neuer BASIC-Befehl (s.o.)
Die Memory Size automatisch setzen
Kaffee kochen ab sofort gestattet (BEL bei Fehlern!)
BEL ohne Nachladen autom. bei BOOT (siehe oben?)
DSMBLR ein wenig komfortabler
Mehr über DDE
LPRINT CHR$ irgendwas
Und es geht doch: LPRINT CHR$(10)
ColourGenie liest NEWDOS und umgekehrt!
Wie ist GENIETEXT aufgebaut
Die Systemoptik ausgetrickst!
ITEM: PRINT @
Convert SCRIPSIT → ASCII

Neuer Dreitastenbefehl ",./"

Das mehrfach erwähnte Problem, innerhalb des residenten Teils von SYS0/SYS eine freie Stelle für weitergehende Features zu finden, ist bisher nur unbefriedigend gelöst: Für das heute vorgestellte Programm mußte vorerst der DOS-Eingabepuffer herhalten. In dessen oberem Teil gehen nun 11 Bytes verloren, so daß sich die maximale Befehlslänge von 80 Zeichen (inkl. NEW LINE, also 79 Schriftzeichen) auf gut eine Bildschirmzeile verringert. Das könnte alle halbe Jahre problematisch werden. Trotzdem, wer eine bessere Stelle in SYS0 weiß, bitte weitersagen!

Das Programm, das durch die Zaps in diesem Beitrag aufgerufen wird, soll bei mir mal wieder den Memory-Banking-Adapter EG 64 MBA unterstützen - ich weiß, Ihr habt's geahnt. Dieses Programm stelle ich aber nicht vor, sondern nur eine Methode, wie man es aufrufen kann. Folglich könnt Ihr auf diese Weise ein beliebiges Maschinensprache-File starten, meinetwegen Space Invaders.

Eine Memory-Banking-Utility oder ein entsprechendes Programm Eurer Wahl ist ihrer/seiner Natur nach ein Bestandteil des Betriebssystems. Es sollte deshalb möglich sein, es jederzeit, und sei es mitten in einem BASIC-Programm, aufzurufen, ähnlich DEBUG, um anschließend nahtlos fortfahren zu können. Das ist genau die Anforderung, die an ein SYS-File gestellt wird. So ist es nur logisch, hierfür eine SYS-Datei zu kreieren, die nach Möglichkeit auch im Bereich 4D00-51FFh arbeitet, so daß der Anwenderspeicher frei bleibt.

G-DOS/NEWDOS kann 32 Systemdateien ansprechen. Zwei davon heißen GDOS/SYS (BOOT/SYS) und INHALT/SYS (DIR/SYS). Die übrigen sind durchnumeriert von SYS0 bis SYS29. Sie alle existieren bereits auf der Systemdiskette. Also keine Chance mehr? Schaut man sie sich der Reihe nach mit DDE bzw. SUPERZAP/CMD an, stellt man fest, daß einige (je nach DOS verschiedene) zwar eingerichtet sind, aber nichts als Nullen enthalten (FFh bei TRSDOS, soviel ich weiß). Sie werden vom System auch nicht aufgerufen. Diese Files sind frei verfügbar und können für das hier besprochene Anliegen herhalten!

Ein SYS-File zu erstellen, ist überhaupt kein Problem. Man kann unter dem Namen z. B. SYS26/SYS ein eigenes Maschinenprogramm assemblieren. Es wird dann genau den Diskettenplatz belegen, der für das bisher funktionslose SYS-File reserviert war. Oder man kopiert ein fertiges Programm mit COPY,filename/typ:dr,SYS26/SYS:0. Das Resultat ist dasselbe. Dabei spielt es auch keine Rolle, welchen Ladebereich das Programm belegt. Bei meinen Tests stellte sich allerdings eine Bedingung heraus, deren Gründe ich noch nicht durchschaue: Wenn das File länger als der vom SYS-File belegte Platz auf der Diskette ist und deshalb Extensions kreiert werden, wird die Datei beim Aufruf nicht gefunden. Daher muß zuvor sichergestellt sein, daß die Anzahl der benötigten Sektoren höchstens gleich ist.

Zur Sache: Wenn von der Floppy ein INT ankommt, landet seine Bearbeitung schließlich bei 45F2h. Die Befehle, die dort nach dem Retten der Register (PUSH) stehen, werden zunächst abgearbeitet, bis auf den letzten vor dem Restaurieren der Register (POP): CALL 45BEh wird in CALL 435Dh umgezapt. Ab 435Dh bis zum Ende des Eingabepuffers steht nun eine kleine Routine, die die Speicherstelle 3820h abfragt. Dort stehen bestimmte Werte, wenn eine der Satzzeichentasten gedrückt wurde. Waren es Komma, Punkt und Querstrich gleichzeitig, heißt dieser Wert 0Dh bzw. 208d. Trifft diese Bedingung nicht zu, wird endlich 45BEh angesprungen.

Andernfalls wird es jetzt interessant: Ein Dreitastenbefehl, da er z. B. den Bildschirm sauber läßt, ist so wertvoll, daß damit auch längere Routinen aufrufbar sein sollten. Deshalb die Notwendigkeit eines eigenen

SYS-Files, denn in SYS0/SYS ist es dafür zu eng. Eine Systemdatei wird mit RST 28h aufgerufen. Dieser Befehl braucht einen Parameter im Akku (Register A). Ihn zu bestimmen, ist ein bißchen kompliziert:

In binärer Schreibweise muß er das Format

<u>uuubbsss</u>

haben. Dabei bedeutet sss+2 den relativen Sektor im Directory, in dem der Eintrag des SYS-Files steht. Das hieße beim Sektor 2 (gezählt ab 0) z. B. sss=0. Diese drei Bits können maximal die Zahl 7 darstellen. Inkl. 0 sind das 8 ansprechbare Sektoren. bb bedeutet die laufende Nr. des Fileeintrags in diesem Sektor, ebenfalls gezählt ab 0. Diese beiden Bits ermöglichen nach dem obigen Strickmuster die Adressierung der vier oberen Dateien dieses Sektors. So errechnet sich die Möglichkeit, insg. 32 Files mit RST 28h zu laden. Und uuu kann zur Übergabe eines Parameters genutzt werden. Lt. NEWDOS-Handbuch muß uuu auf jeden Fall > 0 sein. Es scheinen noch mehr Bedingungen daran zu hängen, so daß man sicherheitshalber immer das Bit 7 setzen sollte. Dieses Berechnungsschema für den Parameter im Akku steht im deutschen Handbuch auf S. 168.

Hier nun das konkrete Fallbeispiel, wieder für SYS26/SYS: Diese Datei steht im 6. relativen Sektor des Inhaltsverzeichnisses (inkl. 0. Sektor). 6-2=4, also muß sss 4 lauten bzw. 100 binär. Es ist, die 0 wieder eingeschlossen, der dritte Eintrag in diesen Sektor. bb heißt deshalb 11 binär (=3). Soll ohne weitere Parameter das Bit 7 gesetzt sein, ergibt sich ein uuu-Code von 100b. uuu, bb und sss addieren sich ihrem Stellenwert entsprechend zu 10011100b, also 9Ch bzw. 156d. Demnach brauchen wir, um SYS26/SYS aufzurufen, die Befehlssequenz LD A,09CH - RST 28H.

Uff! Nachdem Euch die Lektüre dieser Erläuterung zunächst abgeschreckt haben dürfte, macht Euch bitte die Mühe, es ein paarmal zu probieren. Spätestens nach der fünften Berechnung schafft Ihr es bereits im Kopf!

Werden nun die Tasten <,./> gleichzeitig gedrückt, wird SYS26/SYS geladen und angesprungen. Was dieses File leisten soll, ist Euch überlassen. Es könnte z. B. ein hübscher Bildschirm in einen anderen Speicherbereich gerettet oder auf die Floppy ausgelesen werden. Der Phantasie sind nur die Grenzen Eurer Kreativität gesetzt.

Auf jeden Fall muß das Programm mit RET abschließen. Und zwar deshalb: An 4600h (s. Assemblerlisting) steht ein CALL nach 435Dh. Bei erfolgreicher Tastaturabfrage wird der RST-Befehl angesprungen, der ebenfalls für die CPU ein CALL ist, also den Stack um zwei Bytes nach unten verlängert. Bei der Bearbeitung des Befehls RST 28h wird allerdings der Stackpointer zweimal inkrementiert, so daß wir uns wieder in der ersten CALL-Ebene befinden. Daher genügt das simple RET, mit dem das Programm schließlich am Ende der INT-Serviceroutine weitermacht, von wo wir ursprünglich gestartet sind.

Das Assemblerlisting zeigt den Programmablauf der beiden Zaps in SYS0/SYS. Die Dumps darunter sind die Hex-Codes, die in das File gezapt werden müssen. Es genügen wie immer die unterstrichenen Codes. Die ersten 4 Stellen der linken Spalte bezeichnen den relativen Sektor des Files, die beiden folgenden Hex-Ziffern vor dem Doppelpunkt verweisen auf das relative Byte dieses Sektors, mit dem die angezeigte Zeile beginnt.

Der fortgeschrittene Leser wird bemerken, daß das Byte 4 des Sektors 0Eh nicht zum Programm gehört und dennoch umgezapt wird. Es hat mit der Record-Organisation zu tun. Auch die ersten vier Bytes ab E7h im selben Sektor erscheinen nicht im Listing. Dieser Artikel wäre endgültig unverdauliche Kost, wenn ich jetzt auch noch dieses erklären wollte. Nehmt es hin und zapt!

Da ich gerade Eure Aufmerksamkeit habe (danke!) noch kurz zu einer anderen Sache: Seit einiger Zeit erstelle ich mit Hilfe von DSMBLR, DEBUG, SUPERZAP, ziemlich viel Bier, EDTASM, DDE usw. einen Assembler-Quellcode von SYS0/SYS. Im Gegensatz zu den ziemlich abstrakten Hex-Codes, die uns z. B. SUPERZAP an den Kopf wirft, ist diese Source direkt lesbar. Mit ihrer Hilfe hoffe ich, zumindest in groben Zügen die Alchimistenküche dieser Datei zu durchschauen. Die Source ist bereits recht reichlich kommentiert, Befehlscodes sind von reinen Datencodes (hoffentlich vollständig) unterschieden.

Wer Interesse an diesem Quelltext hat, kann mir zu diesem Zweck eine formatierte Diskette (G-DOS oder NEWDOS) schicken. Für das Porto und einen speziellen "Floppy Disk Shipper" (Versandtasche für Disketten) fügt bitte DM 2,- in Briefmarken bei (bitte gängige Werte). Um mir Herumprobiererei zu ersparen, legt auch bitte einen Zettel mit den PDRIVE-Parametern dazu. Zwischen 80/DS/DD und 40/SS/DD nimmt meine Disco alle Formate an.

Und hier das Listing:

```
SYS0/ZAP                      SOFTSOPP-Software      00:02 23 Jun 84 Seite 1

               00100     TITLE   SYS0/ZAP
               00110
4600           00120     ORG     4600H          ;Ende der INT-Routine
4600 CD5D43    00130     CALL    435DH          ;Ergänzung aufrufen
               00140
435D           00150     ORG     435DH          ;Ende Eingabepuffer
435D 3A2038    00160     LD      A,(3820H)      ;Tastatur Satzzeichen
4360 FED0      00170     CP      0D0H           ;<,./> gedrückt?
4362 C2BE45    00180     JP      NZ,45BEH       ;sonst normal weiter
4365 3E9C      00190     LD      A,9CH          ;Code für SYS26/SYS
4367 EF        00200     RST     28H            ;laden und bearbeiten
               00210
0000           00220     END
00000 mal gepennt
```

Bemerkung: Wer EDTASM plus hat und in der Kopfzeile seiner Listings immer noch Reklame für Microsoft macht, hat mit denen wohl einen Vertrag. Es ist ganz einfach, diesen Text den eigenen Bedürfnissen anzupassen (s. o.).

Die Sektordumps auf der nächsten Seite sind mit DDE erstellt. Mit SUPERZAP wird SYS0/SYS mit "DFS" aufgerufen, anschließend kann man die beiden relativen Sektoren, wie oben beschrieben, anzeigen und ändern.

Sekt. Ø2h, alt

```
000200:  79DA 7D04 C358 043A 6943 EE20 E668 3ECB  y.ü..X.:iC. .h>.
000210:  CCDD 49C8 21BE 4536 C9E5 2136 407E B73E  ..I.!.E6..!6§ß.>
000220:  0020 0332 8045 AFF6 0018 09B6 2004 3D32  . .2.E...... .=2
000230:  7345 3600 2E36 0101 3816 FF0A 5FAE 73A3  sE6..6..8..._.s.
000240:  200D 7AC6 0857 2CCB 01F2 4A45 AF18 0F5F   .z..W,...JE..._
000250:  140F 30FC C5D5 222A 45CD 0B04 E1C1 B757  ..0..."*E......W
000260:  2818 7CFE FF32 7345 3E00 2006 0AA5 28EE  (.ö..2sE>. ...(.
000270:  3EFF 3237 453E 0232 8045 CDBF 45E1 3600  >.27E>.2.E..E.6.
000280:  FE1F 281D C9E6 DFD6 41FE 1F79 3815 FE20  ..(.....A..y8..
000290:  C021 7F38 7E23 A60F 79D0 21B4 457E EEC9  .!.8ß#..y.!.Eß..
0002A0:  77AF C9EE 20C9 FE61 D8FE 7FD0 D620 C900  w... ..a..... ..
0002B0:  2169 437E E66C 2026 3A01 38FE D018 053E  !iCß.1 &:.8....>
0002C0:  E30E 04EF 3A10 38FE 0E18 0E3A BE45 D6C9  ....:.8....:.E..
0002D0:  CA0D 44F1 C1D1 E118 223A 0238 FE1C 7AC9  ..D.....":.8..z.
0002E0:  3EA5 EFF5 E5D5 C53A E037 07CD 5301 00FC  >......:.7..S...
0002F0:  4540 DC10 46CD BE45 C1D1 E1F1 FBC9 F53E  E§..F..E.......>
```

Sekt.. Ø2h, neu

```
000200:  79DA 7D04 C358 043A 6943 EE20 E668 3ECB  y.ü..X.:iC. .h>.
000210:  CCDD 49C8 21BE 4536 C9E5 2136 407E B73E  ..I.!.E6..!6§ß.>
000220:  0020 0332 8045 AFF6 0018 09B6 2004 3D32  . .2.E...... .=2
000230:  7345 3600 2E36 0101 3816 FF0A 5FAE 73A3  sE6..6..8..._.s.
000240:  200D 7AC6 0857 2CCB 01F2 4A45 AF18 0F5F   .z..W,...JE..._
000250:  140F 30FC C5D5 222A 45CD 0B04 E1C1 B757  ..0..."*E......W
000260:  2818 7CFE FF32 7345 3E00 2006 0AA5 28EE  (.ö..2sE>. ...(.
000270:  3EFF 3237 453E 0232 8045 CDBF 45E1 3600  >.27E>.2.E..E.6.
000280:  FE1F 281D C9E6 DFD6 41FE 1F79 3815 FE20  ..(.....A..y8..
000290:  C021 7F38 7E23 A60F 79D0 21B4 457E EEC9  .!.8ß#..y.!.Eß..
0002A0:  77AF C9EE 20C9 FE61 D8FE 7FD0 D620 C900  w... ..a..... ..
0002B0:  2169 437E E66C 2026 3A01 38FE D018 053E  !iCß.1 &:.8....>
0002C0:  E30E 04EF 3A10 38FE 0E18 0E3A BE45 D6C9  ....:.8....:.E..
0002D0:  CA0D 44F1 C1D1 E118 223A 0238 FE1C 7AC9  ..D.....":.8..z.
0002E0:  3EA5 EFF5 E5D5 C53A E037 07CD 5301 00FC  >......:.7..S...
0002F0:  4540 DC10 46CD 5D43 C1D1 E1F1 FBC9 F53E  E§..F.üC.......>
```

Sekt. ØEh, alt

```
000E00:  0000 0001 F1E6 5000 0000 0000 0000 0000  ......P.........
000E10:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E20:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E30:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E40:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E50:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E60:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E70:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E80:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E90:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EA0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EB0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EC0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000ED0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EE0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EF0:  0000 0000 0000 0104 2840 4800 0202 004D  ........(§H....M
```

Sekt. ØEh, neu

```
000E00:  0000 0001 E2E6 5000 0000 0000 0000 0000  ......P.........
000E10:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E20:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E30:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E40:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E50:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E60:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E70:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E80:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000E90:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EA0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EB0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EC0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000ED0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
000EE0:  0000 0000 0000 0001 0D5D 433A 2038 FED0  .........üC: 8..
000EF0:  C2BE 453E 9CEF 0104 2840 4800 0202 004D  ..E>....(§H....M
```

Arnulf Sopp, Tel. Ø451-791926

Die Library vergrößern

Die "Bibliothek" (engl. library) des G-DOS, H-DOS und NEWDOS/80 steht in SYS1, verteilt auf zwei Sektoren. In G-DOS 2.1 für das Genie 3 und in H-DOS für die Modelle 1 und 2 ist sie wegen der zusätzlichen Befehle nahezu randvoll. Gleichwohl ist am Ende von SYS1 noch etwas Platz. Der Haken ist aber, daß die Library eben nicht dort, sondern einen Sektor früher zuende ist. Es folgt die Bearbeitungsroutine für den LIB-Befehl nahezu unmittelbar.

Um für weitere Befehle Platz zu schaffen, gibt es ein paar Möglichkeiten. Z. B. ist in G-/H-DOS der Befehl LIB überflüssig, weil das Befehlsverzeichnis auch mit ? ausgegeben werden kann. Dasselbe gilt für mehrere andere Befehle. Aber eine Erweiterung der Library mit einer Verkürzung erkaufen? Dabei ergäbe sich auch das Problem der Kompatibilität meines H-DOS mit G-DOS. Also änderte ich zunächst nur solche Befehlswörter ab, die es nur in H-DOS gibt: INIT heißt jetzt INI, BANK? wurde zu B?. Der neue Befehl ID, mit dem PDRIVE-Parameter von Disketten automatisch erkannt werden, hat nun Platz.

Gut und schön, es soll aber nicht der letzte Mögliche gewesen sein. Mit einer etwas komplizierteren Manipulation gab es noch einmal Platz für einen weiteren Befehl, der genau ein Zeichen lang sein darf. Nach diesem Zeichen müssen noch drei weitere Bytes vorhanden sein für einen Code, der im Register C an das aufgerufene SYS-File übergeben wird, den Requestcode, der in den Akku kommt und ein Schlußbyte, das anzeigt, ob dieser Befehl die Angabe von zusätzlichen Operanden/Parametern erlaubt. Diese insgesamt vier Bytes wurden auf folgende Weise frei:

Die oben erwähnte LIB-Bearbeitungsroutine beginnt mit dem Befehl LD HL,4F58. Damit wird HL als Zeiger mit dem Beginn der Library geladen. Dieser Befehl kann ohne weiteres an eine andere Stelle verschoben werden. Im Prinzip ginge das auch mit den folgenden Maschinencodes, aber aus Gründen, die zu erläutern hier zu weit führen würde, hätte das einen Rattenschwanz von zusätzlichen Änderungen zur Folge. Also nur drei Bytes Gewinn, aber das reicht immerhin für einen neuen Befehl (hat jemand eine Idee, was der bewirken sollte?).

Der LIB-Befehl wird nun auf das Ende von SYS1 umgeleitet. An der Stelle 51E2 wird zuerst der erwähnte Ladebefehl für den Zeiger nachgeholt und sodann an die nächste Stelle in der alten Routine gesprungen. Um dies aber zu ermöglichen, mußte der letzte Record von SYS1 bis ans Ende des letzten Sektors verlängert werden. Das Zählbyte im Record-Header wurde entsprechend geändert, der EOF-Code rutschte ganz nach hinten.

Das war für die Theoretiker, hier nun die Praxis: Auf der letzten Seite dieses Beitrags stehen drei Sektordumps, in denen die modifizierten Bytes unterstrichen sind. Im 0. Sektor wurde der Sprungvektor für den LIB-Befehl auf die neue Adresse gezapt. Im 3. Sektor sind die drei Bytes gekennzeichnet, wo zuvor der Ladebefehl stand. Sie sind nun frei. Im letzten Sektor lautet das Record-Zählbyte jetzt EA, weil noch EAh = 234d Bytes folgen. Am Ende des Records stehen zwei NOPs dort, wo zuvor das EOF war. Es folgen der HL-Ladebefehl und der Sprungbefehl in die alte Routine, schließlich das neue EOF.

Wer gerne an seinem DOS herumzapt, sollte sich unbedingt auf die unterstrichenen Bytes beschränken. Es gibt in H-DOS noch eine ganze Reihe weiterer abgeänderter Bytes, auch in den hier abgebildeten Sektoren. Bitte unbedingt ignorieren, denn diese Änderungen beziehen sich auf wieder andere Stellen im DOS, ohne die eine Änderung das System zuverlässig zum Absturz bringt. Eltern haften für ihre Kinder!

Wem erzähle ich das? Mir ist inzwischen bekannt, daß ich nicht der einzige bin, der DOS-Erweiterungen ausheckt (warum laßt ihr anderen im Info kaum von euch hören?). Wem's in der Library zu eng wird, der kann nun etwas dagegen tun. Das gilt besonders für Benutzer des Genie 3, die in dieser Beziehung ganz schön angemeiert sind.

Arnulf Sopp

```
DRV 00   0102 004D FE23 CA8A 4DFE 4328 73FE 63CA ...M.#..M.C(s.c.
0   10   304E FE83 CA55 51FE A3CA 2A4F FEC3 283F ON...UQ...*O..(?
0H  20   0D28 3A0D CAE2 510D CA32 4D0D CAF3 500D .(:...Q..2M...P.
    30   2852 0DCA 1251 0D28 430D CA34 4E0D 280C (R...Q.(C..4N.(.
DRS 40   0D0D CA68 510D 283C 3E2A B7C9 21DD 51C3 ...hQ.(<>*..!.Q.
143050   6744 D901 01E3 11D3 49C5 D5D9 EFF1 C9CD gD......I.......
596H60   4351 0100 00EB 216A 43CB 76CB F628 04ED CQ....!¡C.v..(..
    70   4B9D 43C5 ED73 9D43 EBC3 354E F1F1 180F K.C..s.C..5N....
    80   AF37 180B 216A 437E E62F 772B CBAE AFF3 .7..!¡Cß./w+....
    90   216B 4336 002B 462B 4E1E 0BF5 CB50 2023 !kC6.+F+N....P.#
    A0   F1F5 3802 280A FE38 2806 1E04 CB69 2013 ..8.(..8(....i..
    B0   CBB6 CB70 205E 3A6C 43CB 7728 0BCB 6920 ...p.^:lC.w(..i.
FRS C0   071E 0C16 EB7A 4BEF CB78 2014 31E0 41CB .....zK..x..1.A.
0   D0   6F21 B045 2213 433E C328 023E C932 1243 o!.E".C>.(.>.2.C
0H  E0   21CD 51CB 7828 07ED 7B9B 4321 C851 FB3E !.Q.x(..ä.C!.Q.>
    F0   1ECD 3300 CB69 CC67 4421 6A43 CBEE 0108 ..3..i.gD!¡C....

DRV 00   0049 802A 0049 4E46 4F81 FF00 0102 0050 .I.*.INFO......P
0   10   4A4B 4C80 A510 4B49 4C4C 8045 904C 4388 JKL...KILL.E.LC.
0H  20   E500 4C46 81FE 004C 4942 82E3 004C 4953 ..LF...LIB...LIS
    30   5485 F088 4C4F 4144 80A4 504C 5754 81F9 T...LOAD..PLWT..
DRS 40   004E 81E4 B04E 4446 C028 0050 4155 5345 .N...NDF.(.PAUSE
143350   88EB 0050 4483 E900 504F 5254 82FF 0050 ...PD...PORT...P
599H60   5249 4E54 86F0 8850 524F 5486 E900 5055 RINT...PROT...PU
    70   5247 4589 E900 5280 2300 5381 E900 5354 RGE...R.#.S...ST
    80   4D54 89EB 0055 4852 82E5 0056 2B84 E500 MT...UHR...V+...
    90   5632 3480 FA00 5A86 FF00 5A45 4954 8AE9 V24...Z...ZEIT..
    A0   0026 83E5 0021 83EB 8A3B 86E3 002F 85E3 .&...!...;.../..
    B0   003F 82E3 003E C048 004D 3E82 EBB0 4444 .?...>.H.M>...DD
FRS C0   45C0 F100 494E 4980 BA00 423F 80DF 002A E...INI...B?...*
3   D0   809C 004F 5554 87FF 0049 4480 F800 0000 ...OUT...ID.....
3H  E0   0000 0E40 0608 7ECB 7F23 2005 CDB7 5110 ...§..ß..#....Q.
    F0   F523 237E B7CA B551 0DCC B551 28E4 CDAD .##ß...Q...Q(...

DRV 00   5118 E1F3 CD43 5121 6A43 7EE6 C020 333A Q....CQ!¡Cß...3:
0   10   01EA 0051 2240 F5ED 739B 43CB FEFB 3E0D ...Q"§..s.C...>.
0H  20   CD33 00C3 8A4D 216A 43CB 7ECA 444D ED7B .3...M!¡C.ß.DM.ä
    30   9B43 3E0E CD33 00F1 B747 3E0F CC33 0078 .C>..3...G>..3.x
DRS 40   3222 40F3 CBBE AF08 FDE1 DDE1 F1C1 D1E1 2"§.............
143450   D9C1 D1E1 08FB C9F1 E5D5 C508 D9E5 D5C5 ................
59AH60   F5DD E5FD E5D9 08F5 C9CD 6851 F57E D603 ..........hQ.ß..
    70   2802 D60A 2801 23F1 C911 8044 D506 20CD (...(.#....D....
    80   7251 D106 00C9 7EFE 2A20 0412 1323 05E5 rQ....ß.*....#..
    90   7ED6 30FE 0ACD A151 3016 7ED6 2EFE 0DCD ß.0....Q0.ß.....
    A0   A151 3806 3E03 12F1 AFC9 7E12 1323 10EA .Q8.>.....ß..#..
    B0   F601 E17E C9D8 7ED6 41FE 1FD8 D620 FE1F ...ß..ß.A.......
FRS C0   C93E 20CD B751 10F9 C93E 0DD5 F5CD 3300 .>...Q...>....3.
4   D0   F1D1 C943 4D44 4A4F 4254 4F00 074D 202D ...CMDJOBTO..M.-
4H  E0   204B 6F6D 6D2C 2068 6175 2072 6569 6E3A .Komm,.hau.rein:
    F0   2003 0000 0000 2158 4FC3 D250 0202 004D ......!XO..P...M
```

Mehrere SYS-Files gleichzeitig

Einige Systemdateien von NEWDOS, G-DOS, H-DOS, TRSDOS usw. arbeiten mit anderen zusammen. So werden beispielsweise von der G-DOS/H-DOS-Funktion DDE in SYS15/SYS weitere SYS-Files aufgerufen, um Sektoren zu laden, zu schreiben, das Inhaltsverzeichnis zu sichten usw.. Da sich diese Dateien nicht überlagern dürfen, hilft sich DDE mit dem sicherlich einfachsten Ausweg: Der Ladebereich von SYS15/SYS beginnt bei 5200.

Es ist kaum anzunehmen, daß man mitten in einem gerade laufenden Programm mit DDE eine Datei modifizieren will. Deshalb ist es unproblematisch, daß SYS15/SYS den unteren Teil des Anwenderspeichers zuschaufelt. Soll jedoch eine Systemdatei im Hintergrund jederzeit, also auch mitten in einem Programm abrufbar sein, ohne anderen Systemdateien den Platz wegzunehmen, muß man sich etwas einfallen lassen.

Bei meinem H-DOS war diese Notwendigkeit gegeben. Darin spielt der EG 64 MBA eine entscheidende Rolle. Er wird von mehreren SYS-Files angesteuert. Da in einigen von ihnen der Platz sehr beschränkt war, sollte ein gemeinsames Unterprogramm für alle da sein. Ich entschied mich für SYS24/SYS, eine freie Systemdatei (die beim Genie 3 übrigens belegt ist). Für den Ladebereich bleibt nicht mehr viel Auswahl: Nur noch der DOS-Eingabepuffer 4318-4367 und der Sektorpuffer 4200-42FF stellen unterhalb 5200 noch eine nennenswerte Lücke dar.

Wenn man wegen der Mutterschaft der Vorsicht über die Porzellankiste alle 5 Sektoren von SYS24/SYS erhalten will, entsteht nun ein gravierendes Problem: In den Sektorpuffer wird alles geladen, auch Records, die mit dem Code 05 (s. Sektordumps) als das gekennzeichnet sind, was in BASIC REM heißt. Das bedeutet, daß der letztenendes aktive Record, der direkt im Sektorpuffer laufen soll, von allem Nachfolgenden verschüttet würde. Folglich darf nichts nachfolgen. Dieser Record muß im unwiderruflich letzten Sektor stehen. Alles Davorliegende muß entweder "REM" (05) heißen, also für den Programmablauf belanglos sein oder in einen anderen Speicherbereich laden, z. B. den Input Buffer.

Es gibt eine weitere Besonderheit, die zu beachten ist: Sollte von einer solchen SYS-Datei aus eine andere aufgerufen werden, so muß die Kontrolle endgültig an diese übergehen, denn ihre Sektoren werden selbstverständlich ebenfalls über 4200 ff. an ihren Bestimmungsort geladen, wobei sie unseren Spezialfile in die ewigen/Jagdgründe schicken. Kurz: Während unser SYS24/SYS arbeitet, ist jegliche Disk-I/O ganz einfach verboten.

Um SYS24/SYS aufzurufen, gibt es zwei Möglichkeiten. Auf jeden Fall muß der Akku mit xA geladen sein, wobei x eine ungerade Zahl >= 3 ist. Warum, das will ich jetzt nicht erklären, weil es zu weit führen würde. Der Leser wolle sich bitte mit dem Aufruf von SYS-Dateien in der einschlägigen Literatur vertraut machen, z. B. im Clubinfo des Genie/TRS80 User Club Bremerhaven. Die eine Möglichkeit besteht darin, direkt mit dem Befehl RST 28H SYS24/SYS anzuspringen. Das hat dieselbe Wirkung wie ein GOTO in BASIC. Oder man callt eine der vielen Stellen im DOS-Kern 4000-4CFF, wo ein RST 28H steht. Das kommt einem GOSUB gleich, so daß nach Erledigung an der alten Stelle fortgefahren werden kann.

Nanu!? Angeblich ist ein RST doch ein Unterprogrammaufruf wie ein CALL! Der Schlüssel zu dieser Merkwürdigkeit liegt in der Bearbeitungsroutine von RST 28. An 4BC2 wird nämlich der Stackpointer zweimal inkrementiert, wodurch die RET-Adresse vom Stack "verschwindet" (richtiger: Ein RET würde die falsche, die darüberliegende Adresse finden).

Mein MBA-Problem, das auf diese Weise gelöst werden konnte, ist nur ein Beispiel. Es wäre interessant, in einem der kommenden Infos weitere

Anwendungen dieser Technik aus den Reihen des Clubs zu finden.

Noch etwas zu den Sektordumps: Da die Sektoren 0-3 völlig identisch sind, sind nur pars pro toto der Sektor 0 und als einziger Programmmsektor der Sektor 4 wiedergegeben. Das erste Byte jedes Sektors bedeutet die Gattung seines Inhalts. Sektor 0 ist Füllwerk und deshalb mit 05 gekennzeichnet. In Sektor 4 steht ein Maschinenprogramm, das mit dem Code 01 beginnt.

Arnulf Sopp

SYS24/SYS, Sekt. Ø

```
000000: 05FE 0000 0000 0000 0000 0000 0000 0000 ................
000010: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000020: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000030: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000040: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000050: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000060: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000070: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000080: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000090: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0000A0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0000B0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0000C0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0000D0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0000E0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
0000F0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
```

SYS24/SYS, Sekt. 4

```
000400: 01E2 0042 CD54 4428 0B7E E6DF FE4A 2877 ...B.TD(.ß...J(w
000410: FE4E 2804 3E2F 182B CD5D 42DB DF21 5348 .N(.>/.+.üB..!SH
000420: 2208 453E 3A32 D345 2110 3822 D445 AF18 ".E>:2.E!.8".E..
000430: 13E5 D5C5 F3FE 9A28 4EFE BA28 C7FE DA28 .......(N..(...(
000440: D73E 37B7 C1D1 E1FB C9FE FA20 E43E 0BD3 .>7........ .>..
000450: DF3D D3DF 21FF 3536 007E B73E 08C0 3EFD .=..!.56.ß.>..>.
000460: EF06 043E 0ED3 DF3D 10FB 3DD3 DF01 0037 ...>...=..=....7
000470: 6169 5159 EDB0 707E B728 05F1 3E08 18C3 aiQY..pß.(..>..
000480: 1C01 FF08 EDB0 C9CD 5D42 21B5 4211 7500 ........üB!.B.u.
000490: D501 2B00 EDB0 2185 0022 0845 AF32 FF2F ..+...!..".E.2./
0004A0: 32FF 35DB DF3E 08D3 DF3E 0FD3 DFE1 3ECD 2.5..>...>....>.
0004B0: 32D3 4522 D445 AF18 8B3A 2038 FED0 2806 2.E".E...: 8..(.
0004C0: 3A10 38FE E0C0 F61C EF79 FE07 C253 480E :.8......y...SH.
0004D0: B041 3E01 D3FF 10FE 413C D3FF 10FE 0D41 .A>.....A<.....A
0004E0: 10F0 F1C9 0202 4542 0000 0000 0000 0000 ......EB........
0004F0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Neuer DOS-Befehl: OUT port#,xx,yy,...

Wenn man sich sein DOS näher ansieht, wird man je nach Version feststellen, daß möglicherweise nicht jede Funktion der Library auch wirklich lauffähig ist. Bei meinem G-DOS 2.1b (das ich inzwischen nach all' den Zaps frech 2.1c nenne) ist beispielsweise der Befehl V24 wirkungslos. Er soll eigentlich die V24-Schnittstelle, sofern eingebaut, initialisieren. Gibt man diesen Befehl ein, erscheint jedoch die Meldung, daß diese Funktion noch nicht implementiert sei. Der Speicherbereich, wo diese Meldung steht und die Routine, die sie anzeigt, stehen deshalb für Sinnvolleres zur Verfügung.

Eine der großen Stärken des Z80 ist seine Fähigkeit, 256 Ports anzusprechen. In BASIC ist das sehr einfach durch den OUT-Befehl. Vom DOS aus geht es aber nicht (Geduld, gleich geht es!). Wer immer in BASIC arbeitet, kann jetzt weiterblättern, falls er nicht sein DOS für den Wiederverkauf aufwerten will. In Maschinensprache ist die Ausgabe einer Größe auf einen Port zwar genauso simpel, es gibt dafür aber keinen DOS-Befehl. So war das hier vorgestellte Programm kein Problem, nur die Frage, wie man diesen Befehl implementieren könnte.

In SYS1/SYS sind alle DOS-Befehle gespeichert. Das System checkt bei einer Eingabe, ob einer dieser Befehle eingetippt wurde. Ist das der Fall, wird die entsprechende Routine geladen und abgearbeitet. Dabei wird bei dieser Befehlstabelle nicht überprüft, ob sie die Original-Apparat-Befehle oder die TCS-Verballhornungen enthält. Man kann daher problemlos z. B. "LOAD" in "LADE" umzappen. Mit etwas mehr Aufwand wäre sogar "SCHLÜRFE DIR 'REIN" möglich. So änderte ich kurzerhand "V24" in "OUT" um (oberer Sektordump).

Damit allein ist es allerdings nicht getan, denn der OUT-Befehl soll bitteschön auch befolgt werden, und dazu braucht es eine Bearbeitungsroutine. Nach dem Befehl V24 wird nach 519E in SYS29/SYS verzweigt, wo die Routine zur Anzeige der Denkste-Meldung steht. Die wiederum steht ab 515E. Also ist der ganze Bereich dazwischen verfügbar. Um Platz zu sparen, wird deshalb der Sprungbefehl am Anfang von SYS29/SYS auf 515E umgezapt (mittlerer Sektordump). Ab 515E folgt nun die Bearbeitungsroutine (unterer Dump). Und die geht so:

Gemäß der Befehlssyntax wird als erste Hexzahl die Portnummer erwartet (s. Überschrift). Sie wird eingelesen und dem Register C übergeben, mit dem man unabhängig von der Zahl einen Port indirekt adressieren kann. Sodann kommen die Werte, die auf diesen Port ausgegeben werden sollen. Es ist möglich, gleichzeitig mehrere Werte einzugeben. Die Obergrenze ist durch das DOS gegeben: Der Eingabepuffer faßt maximal 80 Zeichen inkl. NEW LINE am Ende.

Das Befehlswort OUT, die Portnummer und die einzelnen Werte zur Ausgabe werden wie im DOS üblich wahlweise durch Komma oder Blank getrennt. In Zeile ~~280~~ erfolgt deshalb ein CALL nach 4454, wo diese Trennzeichen erkannt werden. Falls in diesem Unterprogramm ein NEW LINE festgestellt wurde, falls also der Befehl zuende ist, steht das Zero-Flag auf 1. In diesem Falle wird die Bearbeitung beendet. Ansonsten wird aus den eingegebenen Hexzahlen der entsprechende binäre Code gebastelt und auf den Port ausgegeben. Sollte ein anderes Zeichen als eine Hexziffer gefunden werden, erfolgt die Fehlermeldung "falsche Parameter".

Sowohl die Rückkehr beim Ende der Befehlsbearbeitung als auch die Fehlerbehandlung haben einen kleinen Haken: Im Unterprogramm GETCHR befinden wir uns bereits in der zweiten Unterprogrammebene. Deshalb muß der Stack mit zwei POP-Befehlen auf den alten Stand gebracht werden, bevor das Programm in die DOS-Befehlseingabe oder die Fehleranzeige zurückspringt. Das ist alles.

* Bei NEWDOS 4C5D

```
                 00100 ;* * * * * * * * * * * * * * * * * *
                 00110 ;*                                  *
                 00120 ;*       OUT port#,xx,yy,...        *
                 00130 ;*      (C) '84 by St. Arnulf       *
                 00140 ;*                                  *
                 00150 ;* * * * * * * * * * * * * * * * * *
                 00160
4D1D             00170          ORG     4D1DH    ;Operand des Sprungbefehls
4D1D 5E51        00180          DEFW    OUT      ;nach dort springen
                 00190
515E             00200          ORG     515EH    ;hier OUT-Routine
515E CD6951      00210 OUT      CALL    GETVAL   ;Hexzahl einlesen
5161 4F          00220          LD      C,A      ;Portnummer in C laden
5162 CD6951      00230 LOOP     CALL    GETVAL   ;Hexzahl einlesen
5165 ED79        00240          OUT     (C),A    ;auf Port ausgeben
5167 18F9        00250          JR      LOOP     ;bis zum bitteren Ende
5169 CD7651      00260 GETVAL   CALL    GETCHR   ;eine Hexziffer einlesen
516C 17          00270          RLA              ;Stellenwert korrigieren,
516D 17          00280          RLA              ;d. h. in linkes Nibble
516E 17          00290          RLA              ;schieben
516F 17          00300          RLA              ;(4 Bits nach links)
5170 57          00310          LD      D,A      ;linke Hexziffer merken
5171 CD7651      00320          CALL    GETCHR   ;nächste Hexziffer einlesen
5174 B2          00330          OR      D        ;LSN mit MSN vereinigen
5175 C9          00340          RET              ;jetzt der korrekte Wert im Akk
5176 CD5444      00350 GETCHR   CALL    4454H*   ;Trennzeichen und CR erkennen
5179 2817        00360          JR      Z,EXIT   ;Ende bei NEW LINE
517B 7E          00370          LD      A,(HL)   ;Hexziffer laden
517C FE30        00380          CP      '0'      ;Dezimalziffer?
517E 3816        00390          JR      C,ERROR  ;falls ASCII < Dezimalziffer
5180 FE3A        00400          CP      ':'      ;> ASCII "9" ?
5182 380A        00410          JR      C,RETURN;falls korrekte Dezimalziffer
5184 FE41        00420          CP      'A'      ;Hexziffer > ASCII "9" ?
5186 380E        00430          JR      C,ERROR  ;falls < ASCII "A"
5188 FE47        00440          CP      'G'      ;> ASCII "F" ?
518A 300A        00450          JR      NC,ERROR;falls falsche Eingabe
518C D607        00460          SUB     7        ;falls Alpha-Hexziffer
518E E60F        00470 RETURN   AND     0FH      ;ASCII nach binär umwandeln
5190 23          00480          INC     HL       ;nächste Bildschirmstelle
5191 C9          00490          RET              ;erledigt
5192 F1          00500 EXIT     POP     AF       ;Stack korrigieren
5193 F1          00510          POP     AF       ;(2. CALL-Ebene)
5194 AF          00520          XOR     A        ;Z-Flag für "kein Fehler"
5195 C9          00530          RET              ;ins Betriebssystem
5196 F1          00540 ERROR    POP     AF       ;Stack korr. (s. o.)
5197 F1          00550          POP     AF
5198 3E2F        00560          LD      A,2FH    ;Fehlercode "falsche Parameter"
519A B7          00570          OR      A        ;Z-Flag rücksetzen
519B C9          00580          RET              ;ins Betriebssystem
002F             00590          END              ;wohlverdient
00000 mal gepennt
33470  Zeichen verfügbar

ERROR  5196 00540   00390 00430 00450
EXIT   5192 00500   00360
GETCHR 5176 00350   00260 00320
GETVAL 5169 00260   00210 00230
LOOP   5162 00230   00250
OUT    515E 00210   00180
RETURN 518E 00470   00410
```

Arnulf Sopp
Tel. 0451-791926

SYS-Files, und wie man sie macht

In der folgenden Tabelle sind alle SYS-Files von G-DOS 2.1b verzeichnet mit ihren Aufgaben und ihrer Länge. Die Länge deshalb, weil man sie beim Benutzen der freien SYS-Dateien kennen sollte. In den beiden letzten Spalten steht in binär und sedezimal, mit welchem Wert der Akku geladen sein muß, wenn man sie mit RST 28h aufrufen will. Manche Files erwarten außer dem Code in A weitere Parameter; Vorsicht ist also geboten!

File	Aufgabe	Sekt.	Akku bin.	hex.[1]
GDOS	DOS booten	5	xxx00000	g0
INHALT	Inhaltsverzeichnis	30	xxx00001	g1
SYS0	DOS-Kern, bis 4CFFh resident	15	xxx00010	g2
SYS1	DOS-Befehle interpretieren	5	xxx00011	g3
SYS2	File-Handling	5	xxx00100	g4
SYS3	dto., JKL, DEBUG usw.	5	xxx00101	g5
SYS4	DOS-Fehlermeldungen	5	xxx00110	g6
SYS5	DEBUG	5	xxx00111	g7
SYS6	NDF, COPY, APPEND, PD, S	35	xxx01000	g8
SYS7	UHR, DATUM, AUTO, ATTRIB usw.	5	xxx01001	g9
SYS8	I (DIR), FREE	5	xxx01010	gA
SYS9	B2, BOOT, Chaining-Kommandos	5	xxx01011	gB
SYS10	BASIC-Befehle GET und PUT	5	xxx01100	gC
SYS11	BASIC-Befehl RENUM (Teil)	5	xxx01101	gD
SYS12	BASIC-Befehl REF	5	xxx01110	gE
SYS13	BASIC-Fehlermeldungen, RENUM	5	xxx01111	gF
SYS14	CLEAR, CREATE, E, LIST, @, DR	5	xxx10000	u0
SYS15	FORM, V24	4	xxx10001	u1
SYS16	Hauptteil von PD	5	xxx10010	u2
SYS17	Hauptteil von S, AIK	5	xxx10011	u3
SYS18	direkte BASIC-Kommandos	5	xxx10100	u4
SYS19	versch. BASIC-Befehle	5	xxx10101	u5
SYS20	dto.	5	xxx10110	u6
SYS21	BASIC-Befehl CMD"O"	5	xxx10111	u7
SYS22	frei	5	xxx11000	u8
SYS23	LWT[2])	5	xxx11001	u9
SYS24	frei	5	xxx11010	uA
SYS25	frei	10	xxx11011	uB
SYS26	frei	5	xxx11100	uC
SYS27	frei	5	xxx11101	uD
SYS28	FORM (Druckercodes)	5	xxx11110	uE
SYS29	INFO	5	xxx11111	uF

[1]) x = 1 oder 0, g = gerade Hex-Ziffer, u = ungerade Hex-Ziffer

[2]) Diese Tabelle gilt für G-DOS 2.1b. Dort ist der Befehl LWT nicht mehr implementiert. Der Laufwerkstest funktioniert aber mit mindestens 3 Blanks und NEW LINE.

Die Aufgaben der SYS-Files sind teilweise der deutschen Anleitung zu NEWDOS-80 entnommen, soweit sie dort verzeichnet sind. Den Rest habe ich selber herausgetüftelt. Da G-DOS und NEWDOS nicht völlig identisch sind, und da ich entgegen anderslautenden Gerüchten nicht unfehlbar bin, sind Irrtümer in der Spalte "Aufgabe" möglich.

Wie SYS-Files aufgerufen werden, ist im Prinzip in dem Artikel "Neuer Dreitastenbefehl <,./>" erklärt. Die obige Tabelle erleichtert hoffentlich die Berechnung des Parameters im Akku. Der geneigte Leser muß sich nun nicht mehr mit einzelnen Bits herumschlagen, es genügt, zu wissen, worin sich eine gerade von einer ungeraden einstelligen Zahl unterscheidet. Darüber hinaus möchte ich diesmal erklären, wie ein eigenes SYS-File geschrieben werden kann. Dazu sind ein paar Vorkenntnisse nötig, deshalb wieder als hors d'oeuvre ein wenig Theorie:

Was später im Hauptspeicher stehen soll, liegt Byte an Byte auf der Diskette wie die einzelnen Töne einer Melodie auf einer Schallplatte. Es ist jedoch in einzelne Portionen, sog. Records unterteilt. Am Anfang eines jeden Records steht ein Byte, das die Art der folgenden Codes kennzeichnet. Bei einem Maschinenprogramm hat es den Wert 01. Das zweite Byte ist die Anzahl der Bytes, die dieser Record enthält. Dabei steht 00 für 256 dez.. Die relativen Bytes 3 und 4 des Records enthalten die Ladeadresse des ersten Codes (Byte 5), und zwar in der gewohnten Reihenfolge LSB-MSB. Die beiden Bytes der Adresse werden im Byte 2 übrigens mitgezählt. Die Abb. 1 zeigt ein beliebiges Beispiel. Die Bytes der Record-Organisation sind unterstrichen.

In einem nicht belegten SYS-File (22, 24, 25, 26 und 27) enthalten alle Records 256 Bytes, also ist der Zähler (rel. Byte 2) immer 00. Bei eigenen SYS-Files (wofür man diese freien Dateien ausnutzen muß) sollte man dieses Schema beibehalten.

Am Ende einer Datei schließlich steht die Einsprungsadresse des Programms. Sie wird mit dem Kenncode 02 eingeleitet. Es folgt auch hier die Anzahl der folgenden Bytes. Da es sich nur um diese Adresse handelt, lautet der Wert dieses Bytes ebenfalls 02 (2 Bytes für LSB und MSB der Adresse, s. Abb. 2a).

Die freien SYS-Files belegen 5 bzw. 10 Sektoren. Eine eigene SYS-Routine mag zwar kürzer sein, aber im Hinblick auf spätere Erweiterungen wäre es verschwenderisch, auf den Rest zu verzichten. Und das macht die Geschichte leider etwas kompliziert, wie wir später sehen werden. Grundsätzlich ist es aber ohne weiteres möglich, unter dem Namen z. B. SYS26/SYS mit EDTASM ein Maschinenprogramm zu assemblieren, das sich (außer der Länge) in nichts von einer Apparat-Systemdatei unterscheidet. Man kann sogar mit POKE-Befehlen ein Maschinenprogramm von BASIC aus zusammenschustern und mit CMD"DUMP,..." sein SYS-File kreieren.

Besser als diese Partisanen-Heimarbeit ist die Methode der sektorweisen Kopie. Dabei bleiben die Lage und Länge sowie alle anderen Eigenschaften der freien SYS-Datei erhalten. EDTASM erzeugt beispielsweise das File SYS26/CMD. Mit SUPERZAP und seinem Befehl DFS kriegt man raus, welche Sektoren dieses Programm belegt. Es sind später für den Befehl CDS die Quellsektoren. Auf die gleiche Weise findet man die Zielsektoren von SYS26/SYS.

Wenn SYS26/CMD auf einer ziemlich vollen Diskette nicht mehr zusammenhängend aufgezeichnet wurde (vorher mit SUPERZAP feststellen!), kann man leider bei der Frage SECTOR COUNT nicht mehr alles auf einen Rutsch erledigen. Wer Sorge hat, bei der Kopie auf Raten Fehler zu machen, kann für SYS26/CMD eine frische Diskette nehmen.

Zuletzt ist von Fall zu Fall noch etwas Kosmetik notwendig. Die Record-Grenzen liegen bei SYS26/SYS nicht notwendigerweise an denselben Stellen wie bei SYS26/CMD. Sie lassen sich einfach umzappen, wie in Abb. 2a (SYS26/CMD) und 2b (SYS26/SYS) gezeigt. Die zu modifizierenden Codes sind unterstrichen. Außerdem enthalten die CMD-Sektoren vielleicht noch Müll des Assemblers (Abb. 2a), den man aus optischen Gründen auf 00 zappen kann (Abb. 2b). Die Einsprungsadresse wird ebenfalls mit Nullen überschrieben, denn am Ende einer jeden SYS-Datei steht sie bereits: 4D00h. Das ist das Bytemuster 02 02 00 4D im letzten Sektor (der hier

nicht abgebildet ist, um nicht eine neue Infoseite zu beginnen). In Abb. 2a stehen diese Codes mitten im Sektor.

Auch bei diesem Beitrag habe ich das Gefühl, Euch eher abgeschreckt als ermutigt zu haben. Wenn Ihr Euer DOS-Original im Panzerschrank laßt und nur mit Kopien arbeitet, darf aber gerne alles schiefgehen. Und es wird einiges schiefgehen, das walte Murphy. Nach zwei Stunden Training, zwei Litern Schweiß und ebensoviel Bier habt Ihr es aber im Griff. Merke: Die teure Disco geht nur mit Hardware von der Art eines Vorschlaghammers kaputt.

```
                ↓    1 2  3 4  5 6  7 8  9 A  B
DRV 00   010B 0C40 C3C2 4BC3 0946 C3F2 4501 0B2D  ...§..K..F..E..-
0   10   40C3 0044 3E43 EFC3 DB4A 0121 3E40 0000  §..D>C...J.!>§..
OH  20   0032 1E08 530A 0500 52FF FF00 0000 0000  .2..S...R.......
    30   0000 00F5 07DC E847 F1C9 0000 0001 1F63  .......G........
DRS 40   407A CD68 407B F50F 0F0F 0FCD 7140 F1E6  §...§.........§..
5   50   0FC6 9027 CE40 2777 23C9 3B3B 1700 0112  ...'.§'.#.;;....
5H  60   0843 0000 1123 0323 0A02 0000 C3B0 4501  .C...#.#......E.
    70   0000 0143 6843 A540 0000 0000 0000 5A11  ...C.C.§.......Z.
    80   2303 230A 0200 0011 02FF 0100 0100 0000  #.#.............
    90   00FF 00FF 0100 0100 0000 00FF 00FF 0100  ................
    A0   0100 0000 00FF 0071 4300 0000 0004 0100  ........C.......
    B0   0100 0100 000D 0D01 30B2 4300 0000 FFFF  ........0.C.....
FRS C0   FC4C 0000 00FF FF00 0000 0000 FFFF 0000  .L..............
0   D0   0000 00FF FF00 0080 2800 0042 0000 FF00  ........(..B....
OH  E0   0000 0040 0000 00FF FF01 0000 443E 23EF  ...§........D>#.
    F0   0000 3E63 EFC8 F53E 26EF C309 463E 65EF  ..>.....>&...F>..
```

Abb. 1

```
DRV 00   0162 FC4D 3063 2C30 642C 3065 2C30 662C  ...MO..O.,O.,O.,
0   10   3067 2C30 6820 2D20 456E 6465 206D 6974  O.,O..-.E.......
OH  20   203C 4E45 5720 4C49 4E45 3E03 527D 636B  .<NEW.LINE>.R...
    30   6B65 6872 6164 7265 7373 653F 2028 4865  ..........?.(H.
DRS 40   783B 2066 7D72 2052 4554 206E 7572 203C  .;.....RET.....<
161 50   4E45 5720 4C49 4E45 3E20 6569 6E67 6562  NEW.LINE>.......
A1H 60   656E 2903 0202 004D 462C 3061 2C30 622C  ..)....MF,O.,O.,
    70   3063 2C30 642C 3065 2C30 662C 3067 2C30  O.,O.,O.,O.,O.,O
    80   6820 2D20 456E 6465 206D 6974 203C 4E45  ..-.E........<NE
    90   5720 4C49 4E45 3E27 0DB0 B1B2 B3B0 2009  W.LINE>'........
    A0   4445 4642 0930 3348 0DB0 B1B2 B4B0 2054  DEFB.03H.......T
    B0   4558 5433 0944 4546 4D09 2752 7D63 6B6B  EXT3.DEFM.'R....
FRS C0   6568 7261 6472 6573 7365 3F20 2848 6578  ..........?.(H..
1   D0   3B20 667D 7220 5245 5420 6E75 7220 3C4E  ;......RET.....<N
1H  E0   4557 204C 494E 453E 2065 696E 6765 6265  EW.LINE>........
    F0   6E29 270D B0B1 B2B5 B020 0944 4546 4209  .)'........DEFB.
```

Abb. 2a

```
DRV 00   3063 0100 FE4D 2C30 642C 3065 2C30 662C  O....M.O..O.,O.,
0   10   3067 2C30 6820 2D20 456E 6465 206D 6974  O.,O..-.E.......
OH  20   203C 4E45 5720 4C49 4E45 3E03 527D 636B  .<NEW.LINE>.R...
    30   6B65 6872 6164 7265 7373 653F 2028 4865  ..........?.(H.
DRS 40   783B 2066 7D72 2052 4554 206E 7572 203C  .;.....RET.....<
135150   4E45 5720 4C49 4E45 3E20 6569 6E67 6562  NEW.LINE>.......
547H60   656E 2903 0000 0000 0000 0000 0000 0000  ..).............
    70   0000 0000 0000 0000 0000 0000 0000 0000  ................
    80   0000 0000 0000 0000 0000 0000 0000 0000  ................
    90   0000 0000 0000 0000 0000 0000 0000 0000  ................
    A0   0000 0000 0000 0000 0000 0000 0000 0000  ................
    B0   0000 0000 0000 0000 0000 0000 0000 0000  ................
FRS C0   0000 0000 0000 0000 0000 0000 0000 0000  ................
1   D0   0000 0000 0000 0000 0000 0000 0000 0000  ................
1H  E0   0000 0000 0000 0000 0000 0000 0000 0000  ................
    F0   0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Abb. 2b

Arnulf Sopp, Tel. 0451-791926

Die Library-Befehle des G-DOS 2.x

In meinem Artikel "SYS-Files und wie man sie macht" (Info 9/84) äußere ich noch skeptisch, daß in der Rubrik "Aufgaben" meiner Tabelle evtl. Fehler sein könnten. Das liegt daran, daß ich verschiedene Systemdateien disassembliert und mir auf die Programme vorsichtig einen Reim gemacht hatte. Die Tabelle war zwar korrekt, aber jetzt kann ich eine lückenlose abliefern. Sie ist diesmal (nicht ganz konsequent) alphabetisch in der Reihenfolge der LIB-Bildschirmausgabe geordnet. Die SYS-Dateien sind dezimal angegeben, die Registerinhalte von A und C (s. u.) sedezimal. Es ist meine persönliche Library nach ein paar Zaps, die nicht mehr überall mit dem Original-DOS übereinstimmt:

LIB-CMD	SYS	A	C	LIB-CMD	SYS	A	C	LIB-CMD	SYS	A	C
0	14	F0	04	§	14	F0	01	AIK	17	53	00
APPEND	6	68	40	ATTRIB	7	E9	05	AUTO	7	E9	04
B2	9	EB	06	BL	3	E5	01	BOOT	9	EB	0A
BREAK	3	E5	05	CLS	1	E3	09	CONT	9	EB	45
COPY	6	48	40	CREATE	14	F0	02	DATUM	7	E9	0B
DIR	8	2A	00	DISK	29	FF	03	DO	9	EB	43
DR	28	FE	02	DUMP	7	E9	07	E	14	F0	07
FORM	28	FE	08	FREE	8	4A	00	HIMEM	7	E9	02
I	8	2A	00	INFO	29	FF	01	JKL	3	A5	00
KILL	3	45	00	LC	3	E5	08	LF	28	FE	01
LIB	1	E3	02	LIST	14	F0	05	LOAD	2	50	24
LWT	23	F9	01	N	2	E4	01	NDF	6	28	40
PAUSE	9	EB	08	PD	7	E9	03	PORT	29	FF	02
PRINT	14	F0	06	PROT	7	E9	06	PURGE	7	E9	08
R	1	23	00	S	7	E9	01	STMT	9	EB	09
UHR	3	E5	02	V+	3	E5	04	OUT	29	FF	07
Z	29	FF	06	ZEIT	7	E9	0A	&	3	E5	03
!	9	EB	03	;	1	E3	06	/	1	E3	05
?	1	E3	02	>	6	48	40	M>	9	EB	02
DDE	15	F1	40	123	5	27	00	567	26	FC	00
,./	26	DC	00								

Die angegebenen SYS-Files sind diejenigen, bei denen eingesprungen wird. Es gibt durchaus Befehle, die mehrere Systemfiles durchlaufen. Man kann das daran erkennen, daß in der betreffenden Routine ein RST 28h vorkommt. Der jeweilige Akkuinhalt verrät dann, wo es hingeht. Die obige Tabelle und die erstgenannte helfen dabei, das Ziel herauszufinden.

Diese neue Tabelle ist das Resultat einer Spielerei: Zapt einmal irgendeinen Befehl um; nennt meinetwegen BOOT einfach KAHN. Wenn Ihr nachher in der Befehlsebene KAHN eintippt, wird das System neu gebootet. Auf dieser Basis konnte ich auch meinen neuen Befehl OUT (auch in diesem Info?) implementieren, der nun den sinnlosen Befehl V24 ersetzt. Bei diesem Zufallsfund blieb es natürlich nicht, sondern die Untersuchungen in SYS1/SYS, wo die DOS-Befehle erkannt werden, zeitigten weitere Ergebnisse:

Hinter jedem Befehlswort stehen drei weitere Bytes. Bei dem ersten ist immer das Bit 7 gesetzt. Danach folgt der DOS-Request-Code, der in den Akku geladen wird, um mit RST 28h in das zuständige SYS-File springen zu können. Das dritte Byte ist zumeist 00. Ein paarmal ist dies nicht der Fall. Bisher bin ich noch nicht dahintergekommen, welche Bedeutung das hat. Die beiden abgedruckten Sektordumps aus SYS1/SYS zeigen die Befehlswörter und diese drei Bytes für jeden Befehl.

Befehlswort, Akkuinhalt und (meistens) 00 als Ende-Markierung (?) waren leicht zu identifizieren. Das erste Schlußbyte aber hat eine wichtige Bedeutung: In vielen Systemdateien steht am Anfang eine Art Hühnerleiter, wo das Register C wiederholt dekrementiert wird. Ist es bei 00 angekommen, wird in das zuständige Segment des Programms gesprungen.

Dieses Byte gelangt aber nicht unverändert durch die Routine, die den Befehl erkennt, sondern das Bit 7 wird zurückgesetzt. So wird aus 82 beispielsweise 02. Nach zweimaligem Dekrementieren ist die Zero-Bedingung erfüllt und der Befehl JP Z,xxxx wird ausgeführt. In einigen SYS-Dateien genügt als Zeiger auf die betreffende Routine allerdings auch der Akku. Dies scheint z. B. in SYS6/SYS der Fall zu sein, wo ich diese Hühnerleiter nicht fand.

Was kann man nun mit diesen Informationen anfangen? Mein DEBUG kommt nicht nur, wenn ich gleichzeitig <123> drücke, sondern auch, wenn ich die Ziffern nacheinander eingebe. Dazu war es nur nötig, das zweite Byte, das in den Akku kommt, von 00 auf 87 zu zappen. Eigene DOS-Befehle, die nicht, wie OUT, einen alten Befehl ersetzen, lassen sich ebenfalls leicht einschummeln. Man muß nur in dem Bereich hinter dem letzten Befehl zuerst das Wort, dann irgendetwas >= 80, dann den korrekten Akku-Wert für das zuständige SYS-File und schließlich 00 in die Library einzappen.

Freie SYS-Dateien gibt's genug. Sie sollen nicht bloß Platz auf der Diskette kosten sondern etwas leisten. Es ist kein Problem, wie man sieht. In einer "NEWDOS8052" genannten DOS-Verschönerung, die angeblich copyrightswidrig in den Kreisen des Clubs kursieren soll, ist PRINT in LLIST umbenannt, um eine Analogie zu BASIC zu schaffen. Wer dergleichen sinnvoll findet, kann das nach der beschriebenen Methode ebenfalls machen und noch einiges mehr. Viel Spaß dabei!

Arnulf Sopp*

Beispiel: BOOT: A=EB für SYS9/SYS, C=0A (8A-80=0A)

```
000200: D54C 20A9 CB59 2802 0102 004F E3E5 79E6 .L ..Y(....O..y.   SYS1/SYS, rel. Skt. 2
000210: 0728 0EE5 21BC 5123 2323 3D20 FACD 2A4F .(..!.Q###= ..*O
000220: E179 01D3 49C5 CB7F C806 0021 0042 CB77 .y..I......!.B.w
000230: CA24 44C3 2044 D5C5 011C 091A FE3A 280A .$D. D.......:(.
000240: FE2F 3806 281B 0D13 10F1 2323 E5EB 0600 ./8.(.....##....
000250: 0954 5D2B 1313 13ED B8E1 0E03 EDB8 3E2F .Tü+..........>/
000260: 12C1 D1C9 3084 F000 4081 F000 4149 4B80 ....0...§...AIK.
000270: 5300 4150 5045 4E44 C068 0041 5454 5249 S.APPEND.h.ATTRI
000280: 4285 E988 4155 544F 84E9 0042 3286 EB00 B...AUTO...B2...
000290: 424C 81E5 0042 4F4F 548A EB10 4252 4541 BL...BOOT...BREA
0002A0: 4B85 E500 434C 5389 E310 434F 4E54 C5EB K...CLS...CONT..
0002B0: 0043 4F50 59C0 4800 4352 4541 5445 82F0 .COPY.H.CREATE..
0002C0: 4044 4154 554D 8BE9 0044 4952 802A 0044 §DATUM...DIR.*.D
0002D0: 4953 4B83 FF00 444F C3EB 8A44 5282 FE00 ISK...DO...DR...
0002E0: 4455 4D50 87E9 C845 87F0 0046 4F52 4D88 DUMP...E...FORM.
0002F0: FE00 4652 4545 804A 0048 494D 454D 82E9 ..FREE.J.HIMEM..

000300: 0049 802A 0049 4E46 4F81 FF00 0102 0050 .I.*.INFO......P   SYS1/SYS, rel. Skt. 3
000310: 4A4B 4C80 A510 4B49 4C4C 8045 904C 4388 JKL...KILL.E.LC.
000320: E500 4C46 81FE 004C 4942 82E3 004C 4953 ..LF...LIB...LIS
000330: 5485 F088 4C4F 4144 80A4 504C 5754 81F9 T...LOAD..PLWT..
000340: 004E 81E4 B04E 4446 C028 0050 4155 5345 .N...NDF.(.PAUSE
000350: 88EB 0050 4483 E900 504F 5254 82FF 0050 ...PD...PORT...P
000360: 5249 4E54 86F0 8850 524F 5486 E900 5055 RINT...PROT...PU
000370: 5247 4589 E900 5280 2300 5381 E900 5354 RGE...R.#.S...ST
000380: 4D54 89EB 0055 4852 82E5 0056 2B84 E500 MT...UHR...V+...
000390: 4F55 5487 FF00 5A86 FF00 5A45 4954 8AE9 OUT...Z...ZEIT..
0003A0: 0026 83E5 0021 83EB 8A3B 86E3 002F 85E3 .&...!...;.../..
0003B0: 003F 82E3 003E C048 004D 3E82 EBB0 4444 .?...>.H.M>...DD
0003C0: 45C0 F100 3132 3380 8700 3536 3780 FC00 E...123...567...
0003D0: 2C2E 2F80 DC00 0000 0000 0000 0000 0021 ,./............!
0003E0: 584F 0E40 0608 7ECB 7F23 2005 CDB7 5110 XO.§..ß..# ...Q.
0003F0: F523 237E B7CA B551 0DCC B551 28E4 CDAD .##ß...Q...Q(...
```

Die Records handhaben

In den beiden Beiträgen "Neuer Dreitastenbefehl ..." und "SYS-Files ..." war bereits von der Record-Organisation die Rede. Mich läßt das dumpfe Gefühl nicht los, daß ich mich dort etwas zu global, nur für den Experten verständlich ausdrückte. Da meine literarischen Absonderungen letztenendes für die Praxis auch des weniger geübten Infolesers etwas bringen sollen, möchte ich einen weiteren Beitrag zu diesem Thema nachschieben. Wer bei seiner Zapperei möglichst wenige Fehler machen will, muß mit Records einfach umgehen können.

Es mag auf Anhieb paradox klingen, daß ich als Beispiel eine Datei der Systemdiskette nehme, die als einzige eben nicht in Records gegliedert ist: Habt Ihr schon einmal versucht, GDOS/SYS (BOOT/SYS) zu disassemblieren? Disassembler erwarten als allererstes Byte des Files den Code 01. Er signalisiert ein Maschinenprogramm bzw. ein Datenfeld, das ähnlich einem Programm einem ganz bestimmten Speicherbereich zugeordnet ist. In GDOS/SYS findet sich aber nur der Maschinencode dieses Urladers, nichts von Adressen, kein Kenncode 01. Das liegt daran, daß es DOS erst laden muß und selbst vom Microsoft-ROM geladen wird.

Für unser Problem ist zunächst nur der erste Sektor von GDOS, der eigentliche Urlader interessant und hiervon auch nur die ersten 238 Bytes. Der Rest des Sektors enthält einen Copyright-Vermerk und der Rest des Files Daten zur späteren Verwendung.

Zur Wiederholung: Das erste Byte eines Maschinenprogramms lautet immer 01. Das zweite hält die Anzahl der zu diesem Record, dieser Portion des Programms gehörigen Bytes. Dabei steht 00 für 100h (256d). Das dritte und vierte Byte (die bei der Anzahl bereits mitgezählt werden) stellen in der Folge LSB-MSB die Ladeadresse des ersten zum Maschinencode gehörenden Bytes (des fünften) dar. Erst jetzt folgt das eigentliche Programm. Am Ende des Files schließlich finden wir mit dem Kenncode 02 die Einsprungsadresse des Programms. Nach 02 folgt wieder ein Bytezähler. Er lautet ebenfalls 02, weil nur noch die Adresse (2 Bytes) folgt.

Um z. B. für DSMBLR oder DISASSEM aus GDOS ein lesbares File zu machen, müssen wir für Record-Codes sorgen. Es wäre schade, die ersten vier Bytes dafür zu überschreiben. Davor ist dem File aber kein Sektor mehr zugeordnet, den man dafür verwenden könnte. Die Lösung ist simpel: Wir "borgen" uns ein unbenutztes File, das mindestens zwei Sektoren Platz bietet. Die bereits früher erwähnten freien SYS-Dateien eignen sich hervorragend. Hier kommt willkürlich SYS22/SYS zur Anwendung.

In dessen zweiten Sektor kopieren wir den ersten Sektor von GDOS. Das geht gut mit DEBUG oder SUPERZAP. Die letzten vier Bytes des ersten SYS22-Sektors halten jetzt für die Record-Organisation her. Es beginnt mit 01. Weshalb es ausgerechnet mit FE weitergeht, erkläre ich später. Danach folgt die Ladeadresse 4200h, mit dem LSB beginnend.

Die 252 Nullen davor würde der Disassembler aber ebenfalls nicht verzeihen, deshalb werden sie mit entsprechenden Record-Codes zu dem erklärt, was in BASIC REM heißt: Der Code 05 bezeichnet Bereiche auf der Diskette, die Kommentare enthalten und ansonsten ignoriert werden sollen. Es muß auch hier ein Bytezähler folgen, damit DOS "weiß", ab wo es wieder interessant wird. In unserem Beispiel lautet er FAh (250d Bytes bis zum nächsten Record-Header).

Erinnern wir uns: Die letzte Dump-Zeile von GDOS, das ist jetzt der zweite Sektor von SYS22, enthält nur noch den Copyright-Vermerk. Seine

letzten vier Buchstaben (s. Abb.) sind ohnehin hochgestapelt, also können sie für die Einsprungsadresse dienen. Sie wird, wie gesagt, mit der Bytefolge 02 02 lb mt dargestellt, wobei in diesem Fall konkret 00 42 für 4200h einzusetzen ist. Zwischen dem Kenncode 02 und dem letzten Zählbyte im ersten Sektor liegen 254 Bytes, und so erklärt sich der Zähler FE dort.

Alles klar? Alles klar.

GDOS/SYS, Sektor Ø

```
000000: 00FE 30F3 21EC 3736 FF36 D023 3600 2336 ..0.!.76.6.#6.#6
000010: 0011 0501 D931 E041 21FF 51CD 5242 FE20 .....1.A!.Q.RB.
000020: 4730 2957 CD52 424F CD52 425F 1012 CD52 G0)W.RBO.RB_...R
000030: 4257 0D0D 2CCC 5542 7E12 130D 20F6 18DB BW....UBB... ...
000040: 10F9 CD52 4257 1AFE A513 D5C8 21E5 42C3 ...RBW......!.B.
000050: C342 2C7E C0D9 060A 21E1 3736 01D5 C57B .B,B....!.76...ä
000060: D612 3803 5F36 0921 EC37 CDCE 42ED 53EE ..8._6.!.7..B.S.
000070: 3736 1BCD CE42 3688 11EF 3701 0051 CDD7 76...B6...7..Q..
000080: 427E E683 E281 421A 0203 CB4E C287 42CB BB....B....N..B.
000090: 4EC2 8742 CB4E 20EF CB46 2808 CB4E 20E7 N..B.N ..F(..N .
0000A0: CB7E 28E6 7E36 D0C1 D1E6 FC20 0C1C 7BD6 .B(.B6..... ..ä.
0000B0: 2420 0314 1E00 D97E C9CD D742 360B 1098 $ .....B...B6...
0000C0: 21DD 427E FE03 28FB 23CD 3300 1BF5 CDD7 !.BB..(.#.3.....
0000D0: 42CB 4620 FC7E C93E 063D 20FD C91C 1F52 B.F .B.>.= ....R
0000E0: 4553 4554 031C 1F47 2D44 4F53 3F03 0000 ESET...G-DOS?...
0000F0: 4027 3832 2F38 3420 5443 532F 536F 7070 §'82/84 TCS/Sopp
```

SYS22/SYS, Sektor Ø

```
000000: 05FA 2020 2020 2020 2020 2020 2020 2020 ..
000010: 2020 2020 2020 2020 2020 2020 2020 2020
000020: 4465 7220 436F 6465 2030 3520 6265 2D20 Der Code 05 be-
000030: 7A65 6963 686E 6574 2065 696E 2020 2020 zeichnet ein
000040: 4665 6C64 2C20 6461 7320 766F 6D20 2020 Feld, das vom
000050: 444F 5320 6967 6E6F 7269 6572 7420 2020 DOS ignoriert
000060: 7765 7264 656E 2073 6F6C 6C2E 2046 4120 werden soll. FA
000070: 6461 6869 6E74 6572 2069 7374 2064 6173 dahinter ist das
000080: 5A7B 686C 6279 7465 2C20 6461 6D69 7420 Zählbyte, damit
000090: 6465 7220 666F 6C67 656E 6465 2020 2020 der folgende
0000A0: 4B65 6E6E 636F 6465 2030 3120 2873 2E20 Kenncode 01 (s.
0000B0: 752E 2920 6765 6675 6E64 656E 2020 2020 u.) gefunden
0000C0: 7769 7264 2E20 2020 2020 2020 2020 2020 wird.
0000D0: 2020 2020 2020 2020 2020 2020 2020 2020
0000E0: 2020 2020 2020 2020 2020 2020 2020 2020
0000F0: 2020 2020 2020 2020 2020 2020 01FE 0042                 ...B
```

SYS22/SYS, Sektor 1

```
000100: 00FE 30F3 21EC 3736 FF36 D023 3600 2336 ..0.!.76.6.#6.#6
000110: 0011 0501 D931 E041 21FF 51CD 5242 FE20 .....1.A!.Q.RB.
000120: 4730 2957 CD52 424F CD52 425F 1012 CD52 G0)W.RBO.RB_...R
000130: 4257 0D0D 2CCC 5542 7E12 130D 20F6 18DB BW...,UBB... ...
000140: 10F9 CD52 4257 1AFE A513 D5C8 21E5 42C3 ...RBW......!.B.
000150: C342 2C7E C0D9 060A 21E1 3736 01D5 C57B .B,B....!.76...ä
000160: D612 3803 5F36 0921 EC37 CDCE 42ED 53EE ..8._6.!.7..B.S.
000170: 3736 1BCD CE42 3688 11EF 3701 0051 CDD7 76...B6...7..Q..
000180: 427E E683 E281 421A 0203 CB4E C287 42CB BB....B....N..B.
000190: 4EC2 8742 CB4E 20EF CB46 2808 CB4E 20E7 N..B.N ..F(..N .
0001A0: CB7E 28E6 7E36 D0C1 D1E6 FC20 0C1C 7BD6 .B(.B6..... ..ä.
0001B0: 2420 0314 1E00 D97E C9CD D742 360B 1098 $ .....B...B6...
0001C0: 21DD 427E FE03 28FB 23CD 3300 1BF5 CDD7 !.BB..(.#.3.....
0001D0: 42CB 4620 FC7E C93E 063D 20FD C91C 1F52 B.F .B.>.= ....R
0001E0: 4553 4554 031C 1F47 2D44 4F53 3F03 0000 ESET...G-DOS?...
0001F0: 4027 3832 2F38 3420 5443 532F 0202 0042 §'82/84 TCS/...B
```

Arnulf Sopp, Tel. 0451-791926

BASIC FREI IM RAM VERLAGERN

Üblicherweise werden Maschinenprogramme geschützt, indem man nach dem Einschalten die MEMSIZE-Frage mit der entsprechenden Adresse beantwortet. Wozu der Umstand? Das kann ein Programm auch selbst erledigen. Doch davon mehr im nächsten Artikel.

Hier soll das genaue Gegenteil besprochen werden: BASIC kann einem Programm nichts anhaben, das unterhalb residiert, denn BASIC wächst immer nur nach oben. Der Programmtext (PST, program statement table) und die Variablenliste (VLT, variable list table) bekommen vom Betriebssystem einen Platz zugewiesen, der die Untergrenze bestimmt, von wo sie sich nach den Wünschen des Benutzers nach oben ausdehnen.

Und damit ist für den aufmerksamen Leser auch schon alles gesagt: Man braucht nur die Zeiger auf die jeweils zuzuweisenden Plätze zu "verbiegen", um sich unterhalb von BASIC beliebig viel Platz zu schaffen, wo man ein Maschinenprogramm einschummeln kann. Doch zum Verständnis sind einige Informationen nötig:

Je nach dem, ob man Level 2 oder ein DOS fährt, liegen gewisse wichtige Adressen an unterschiedlicher Stelle. Die Beschaffenheit der Peripherie kann unser Tandy oder Genie nicht ahnen, deshalb gibt es einen reservierten Speicherbereich im RAM, der z. B. von einer angeschlossenen Floppy umgeschrieben wird. Da dieser Bereich u. a. der Kommunikation mit der Peripherie dient, heißt er "communications region". Dort finden sich an festgelegter Stelle auch unsere Zeiger auf BASIC.

Für den unteren BASIC-Bereich, den wir verlagern wollen, sind nur fünf Zeiger von Belang, und zwar der auf den Anfang der PST, den der Liste einfacher Variabler, den Anfang der Liste dimensionierter Variabler, ihr Ende und den Pointer auf das zuletzt gelesene DATA-Statement (der anfangs auf eine Stelle vor der PST zeigt). Die String-Variablen sind ganz oben gespeichert, so daß wir uns hier nicht um sie kümmern müssen. Wo diese Pointers stehen, geht aus dem anhängenden Assembler-Listing hervor.

Dieses Assembler-Programm ist für sich allein völlig witzlos. Wer möchte schon gerne ein Programm laden und fahren, bis endlich sein Zielprogramm drankommt? Sinn der Sache ist, es zum Bestandteil einer eigenen Routine zu machen. Dabei sollte man beachten, daß das kleine Programm nach dem Start überflüssig geworden ist und nicht mehr geschützt zu werden braucht. Daher sollte zuerst das eigene Programm kommen, dann die drei Nullen ab "ENDE", dann das Segment ab "ANFANG". Der Befehl "JP BASIC" ist dann natürlich durch einen Sprung ins eigene Zielprogramm zu ersetzen (das seinerseits nach BASIC springen kann).

Es muß unbedingt beachtet werden, daß das Programm nur von BASIC aus (mit SYSTEM bzw. mit CMD"FILENAME") geladen werden darf. Unter DOS stehen an der Stelle der Zeiger völlig andere Werte, die alles aus dem Lot bringen würden.

Arnulf Sopp. Tel. 0451-791926

```
                  00100 ;**********************************************************
                  00110 ;*                                                        *
                  00120 ;*          BASIC RELOZIEREN                              *
                  00130 ;*                                                        *
                  00140 ;**********************************************************
                  00150 ;(C) 1984 by A. Sopp, Wakenitzstr. 8, 2400 Lübeck 1
                  00160
                  00170
                  00180 ;SYMBOLVEREINBARUNGEN:
40A4              00190 BASPTR  EQU     40A4H   ;START BASIC-PROGRAMMTEXT
40F9              00200 VARPTR  EQU     40F9H   ;START EINFACHE VARIABLE
40FB              00210 ARRANF  EQU     40FBH   ;START DIMENS. VARIABLE
40FD              00220 ARREND  EQU     40FDH   ;DTO. ENDE
40FF              00230 DATPTR  EQU     40FFH   ;ZEIGER AUF LETZTES DATA-STMT.
1A19              00240 BASIC   EQU     1A19H   ;BASIC-WARMSTARTADRESSE
                  00250
                  00260 ;BASIC VERSCHIEBEN:
                  00270
6A45              00280         ORG     6A45H           ;FUER LEVEL 2: 42E8H
                  00290
6A45 215D6A       00300 ANFANG  LD      HL,ENDE         ;NEUER DATA-ZEIGER
6A48 22FF40       00310         LD      (DATPTR),HL     ;VERBIEGEN
6A4B 23           00320         INC     HL              ;EINE STELLE HOEHER
6A4C 22A440       00330         LD      (BASPTR),HL     ;NEUER BASIC-ANFANG
6A4F 23           00340         INC     HL              ;ZWEI STELLEN HOEHER
6A50 23           00350         INC     HL
6A51 22F940       00360         LD      (VARPTR),HL     ;NEUER ZEIGER EINF. VAR.
6A54 22FB40       00370         LD      (ARRANF),HL     ;DTO. ANFANG DIMENS. VAR.
6A57 22FD40       00380         LD      (ARREND),HL     ;DTO. ENDE DTO.
6A5A C3191A       00390         JP      BASIC           ;WARMSTART
6A5D 00           00400 ENDE    DEFB    0               ;BASIC BEGINNT MIT 3 X 0
6A5E 00           00410         DEFB    0               ;2. 0
6A5F 00           00420         DEFB    0               ;3. 0
                  00430
6A45              00440         END     ANFANG
00000 TOTAL ERRORS
34013  TEXT AREA BYTES LEFT


ANFANG 6A45 00300   00440
ARRANF 40FB 00210   00370
ARREND 40FD 00220   00380
BASIC  1A19 00240   00390
BASPTR 40A4 00190   00330
DATPTR 40FF 00230   00310
ENDE   6A5D 00400   00300
VARPTR 40F9 00200   00360
```

P U T T O a d r e s s e - ein neuer BASIC-Befehl

Nein, es ist durchaus nicht mein Lieblingsthema, BASIC im RAM auf- und abzuschubsen. Hat man aber einmal über ein derartiges Thema nachgedacht, um ein paar Seiten für das Clubinfo daraus zu machen, kommen in der Folge ständig neue Ideen zumgleichen Problemkreis.

Heute soll der BASIC-Programmierer etwas davon haben: Der neue Befehl PUT TO geht noch ein Stück weiter als die bisherigen Vorschläge. Er verschiebt ebenfalls BASIC, aber im Speicher befindliche Programme bleiben erhalten. Sie rutschen kurzerhand an die neue Stelle mit und tun dort so, als sei nichts gewesen. In BASIC merkt der User nur dann etwas, wenn er mit PRINT MEM fragt, wieviel Platz er noch im RAM hat.

Die Befehlssyntax lautet wie in der Überschrift. Dabei können die Blanks entfallen, die Adresse muß dezimal eingeben werden. Mit dieser Adresse ist die BASIC-Untergrenze gemeint. Es ist die Stelle, auf die der DATA-Zeiger deutet, bevor RUN eingegeben wurde. In diesem Vor-RUN-Zustand befindet sich übrigens auch das Programm nach der Verschiebung mit PUT TO: Variable sind gelöscht, der DATA-Zeiger weist unmittelbar vor den Programmtext.

Die einzugebende Adresse muß ohne den zuvor besprochenen EG 64 MBA mindestens 17128 betragen, nach oben sind nur die physikalischen Grenzen des RAM gesetzt. Es obliegt dem Anwender, darauf zu achten, daß von dieser Adresse an nach oben für das Programm und später hinzukommende Variable genügend Platz ist. Diese Eigenverantwortung des Users macht deutlich, daß gewisse Grundkenntnisse über die Speicherverwaltung von BASIC nötig sind wie etwa auch bei den Befehlen VARPTR und USR.

In diesem Zusammenhang ist auch der Grund zu sehen, weshalb ich einerseits eine BASIC-Utility vorstelle, andererseits aber nur ein Assembler-Listing anbiete. Der BASIC-Programmierer ohne Maschinensprachekenntnisse wird den Befehl PUT TO wohl nie brauchen, der Z80-Freak hat aber einen Assembler, mit dem er PUT TO seinen eigenen Bedürfnissen anpassen kann (z. B. ORG verlegen).

Das Programm läuft nicht unter Disk-BASIC, weil der PUT-Befehl zweckentfremdet wird. Mit einer kleinen Änderung ist aber auch das möglich: Die Befehle

```
        RST     08H
        DEFB    TO
```

werden ersetzt durch

```
        DEC     HL          ;als Zeiger für RST 10H
        RST     10H         ;nächstes Zeichen laden
        CP      TO          ;TO-Token?
        JP      NZ,627BH    ;PUT-Routine Disk-BASIC
        INC     HL          ;war in RST 8 enthalten
```

(Die Adresse 627BH kann von DOS zu DOS variieren. Schaut euch in 4183/4184H den genauen Vektor an und ermittelt seinen dezimalen Wert mit PRINT PEEK(&h4183)+PEEK(&h4184)*256.)

Und was das Ganze soll? Das Himem wird für tausend kleine Popelprogramme gebraucht; man setzt pausenlos die Memsize für alles Mögliche. Mit PUT TO wird stattdessen die BASIC-Untergrenze bestimmt, so daß

man z. B. ganz oben einen Monitor fahren kann und unten die Tasten entprellt oder (nein, es ist wirklich nicht mein Lieblingsthema) für EDTASM Platz hat. Irgendwo lümmelt sich dann BASIC beliebig dazwischen.

Auch dies ist ein solches Popelprogramm, denn man wird es recht selten brauchen. Immerhin, es kann im Gegensatz zum Meisten dieser Art mitten in der BASIC-Arbeit nachgeladen werden. So war für mich an der ganzen Geschichte hauptsächlich die Frage interessant, wie man überhaupt einen neuen BASIC-Befehl implementiert. Es gibt einige Möglichkeiten:

Zum gleichen Zweck hätte auch das Befehlswort MOVE BASIC dienen können. Dabei würde zunächst ein Syntax Error erkannt werden. Im Verlauf der Fehlerbehandlungsroutine wird auch ins RAM verzweigt, das wir für diesen Zweck umprogrammieren können. Da steht jetzt ein Jump zu unserer neuen Verb Action-Routine. Dort wird der Stack nach der Herkunft des Fehlers durchpflügt. Kommt er aus der Variable Assignment-Routine, checken wir, ob das fehlerhafte Statement genau MOVE BASIC oder MOVEBASIC und nicht anders lautete. Ist das der Fall, ist es kein Fehler und der Spaß geht los.

Zu umständlich? Einverstanden. Da der BASIC-Verschiebebefehl wohl hauptsächlich nicht programmiert, sondern als direktes Statement eingetippt wird, vereinfacht sich die Sache: Der hierfür zuständige DOS-Exit ist 41B2H (JP 6033H, je nach DOS). Wir verbiegen ihn auf unsere Routine und können den Stack Stack sein lassen. Jetzt wird wieder untersucht, ob die folgenden Zeichen genau M-O-V-E-(-)B-A-S-I-C hießen.

Immer noch zu umständlich? Wieder einverstanden. Die einfachste Lösung ist immer, einen Befehl zu "mißbrauchen", den es bereits gibt, der ein Token hat. Wird ein Token angetroffen, springt der Interpreter in die zuständige Verb Action-Routine. So ziemlich jede dieser Routinen verzweigt, z. T. mehrmals, in das freie RAM, wo wir Vektoren auf unsere Ergänzung richten können. Nur ist leider bis zum Sprung ins RAM häufig schon allerhand passiert, das wir wieder einnorden müssen. Deshalb ist eigentlich nur eine einzige Methode wirklich empfehlenswert:

Die reinen Disk-BASIC-Befehle, die unter Level 2 zu einem ?L3 Error oder ?SN Error führen, springen das RAM sofort an. Noch ist alles im Lot, und unser Programm kann kurz und bündig bleiben. Deshalb die zahllosen BASIC-Erweiterungen, die mit NAME aufgerufen werden. Wenn dem Befehl (hier PUT) noch ein weiteres Zeichen oder Token (hier TO) folgt, kann gleich am Anfang festgestellt werden, ob alles seine Richtigkeit hat. Spätere Syntaxchecks übernimmt wieder der Interpreter. Wenn z. B. PUT TO 20000? eingegeben wurde, wird der Befehl zunächst korrekt ausgeführt. Erst dann erkennt Mr. Microsoft das Fragezeichen und versteht Bahnhof: ?SN Error. Wir haben ihn ja nur ergänzt, nicht verändert.

Mit EDTASM ist die Eingabe des Programms am bequemsten. Wer diesen Assembler nicht hat, muß ihn entweder für sehr viel Geld kaufen oder gewisse Mitglieder unseres Clubs ansprechen, die angeblich jedes Copyright mißachten und Programme weitergeben. Einem böswilligen Gerücht zufolge soll es kein Mitglied geben, das es nicht tut. Muß ich deutlicher werden?

Arnulf Sopp, Wakenitzstr. 8, 2400 Lübeck 1, Tel. 0451-791926

```
                 ;***                     PUT TO adresse                  ***
                 ;***                                                     ***
                 ;*** Ein neuer BASIC-Befehl zum freien Verschieben       ***
                 ;*** von BASIC inkl. Programmtext im RAM. Variable       ***
                 ;*** werden dabei gelöscht.                              ***

                 ;(C) 1984 by Arnulf Sopp, Wakenitzstr. 8, D-2400 Lübeck 1


                 ;Symbolvereinbarungen:
4182             PUT      EQU     4182H       ;Adresse PUT-Verb Action-Routine
06CC             BASIC    EQU     06CCH       ;BASIC-Warmstartadresse
1E5A             ASCINT   EQU     1E5AH       ;ASCII-String -> DE als Integer
40FF             DATPTR   EQU     40FFH       ;DATA-Pointer
40A4             PSTPTR   EQU     40A4H       ;Pointer auf BASIC-Textanfang
40F9             VLTPTR   EQU     40F9H       ;dto. auf Anfang Variablenliste
40FB             ARRANF   EQU     40FBH       ;dto. Anf. dimens. Var.
40FD             ARREND   EQU     40FDH       ;dto. Ende (Anfang freies RAM)
00BD             TO       EQU     0BDH        ;TO-Token

                 ;PUT-Vektor verbiegen:
4182                      ORG     PUT
4182 C30080               JP      PUTTO               ;neue Verb Action-Routine

                 ;neue Verb Action-Routine:
8000                      ORG     8000H               ;oder je nach RAM-Größe
8000 CF          PUTTO    RST     08H                 ;Syntax Error, falls
8001 BD                   DEFB    TO                  ;nicht TO angetroffen
8002 CD5A1E               CALL    ASCINT              ;Adresse in DE einlesen
8005 2AFF40               LD      HL,(DATPTR)         ;für die Ermittlung
8008 ED4BF940             LD      BC,(VLTPTR)         ;der Programmlänge
800C 0B                   DEC     BC                  ;auf die 0 davor
800D DF                   RST     18H                 ;rauf oder runter?
800E C8                   RET     Z                   ;falls dieselbe Adresse
800F E5                   PUSH    HL                  ;HL und BC vertauschen
8010 C5                   PUSH    BC
8011 E1                   POP     HL
8012 C1                   POP     BC
8013 D5                   PUSH    DE                  ;neue Adresse für später
8014 380B                 JR      C,UPLOAD            ;Carry, falls rauf
8016 C5                   PUSH    BC                  ;als Quelle für LDIR
8017 ED42                 SBC     HL,BC               ;Programmlänge
8019 E5                   PUSH    HL                  ;als Zähler für LDIR
801A C1                   POP     BC                  ;nach BC laden
801B 03                   INC     BC                  ;Bytezähler korrig.
801C E1                   POP     HL
801D EDB0                 LDIR                        ;BASIC umschaufeln
801F 180C                 JR      CNGPTR              ;dort weiter
8021 E5          UPLOAD   PUSH    HL                  ;spätere Quelle für LDDR
8022 B7                   OR      A                   ;Carry löschen
8023 ED42                 SBC     HL,BC               ;wie oben
8025 E5                   PUSH    HL
8026 C1                   POP     BC
8027 03                   INC     BC                  ;Bytezähler korrig.
8028 19                   ADD     HL,DE               ;ans Ende des Programms
8029 EB                   EX      DE,HL               ;Ziel DE laden
802A E1                   POP     HL                  ;Quelle
802B EDB8                 LDDR                        ;wie oben, von hinten

                 ;BASIC-Zeiger nachstellen:
802D E1          CNGPTR   POP     HL                  ;neue Adresse
802E 22FF40               LD      (DATPTR),HL         ;DATA-Pointer laden
8031 23                   INC     HL                  ;eins höher
8032 22A440               LD      (PSTPTR),HL         ;PST-Pointer laden

```

```
                00750 ;Zeilenzeiger im Programmtext nachstellen:
8035 E5         00760           PUSH    HL              ;erster Zeilenpointer
8036 23         00770           INC     HL              ;insges. 4X erhöhen
8037 D1         00780 LOOP      POP     DE
8038 23         00790           INC     HL              ;auf 1. Programmstatement
8039 23         00800           INC     HL              ;erhöhen (die nächste 0
803A 23         00810           INC     HL              ;muß EOL sein.)
803B AF         00820           XOR     A               ;A=0 für CPIR
803C 47         00830           LD      B,A             ;dto. BC (=256!)
803D 4F         00840           LD      C,A
803E EDB1       00850           CPIR                    ;0 als EOL aufsuchen
8040 EB         00860           EX      DE,HL           ;Register umordnen
8041 73         00870           LD      (HL),E          ;Zeilenpointer korri-
8042 23         00880           INC     HL              ;gieren
8043 72         00890           LD      (HL),D
8044 EB         00900           EX      DE,HL           ;alte Registerordnung
8045 E5         00910           PUSH    HL              ;neuer Zeilenpointer
8046 7E         00920           LD      A,(HL)          ;auf EOL prüfen
8047 23         00930           INC     HL              ;eine zweite 0?
8048 B6         00940           OR      (HL)
8049 20ED       00950           JR      NZ,LOOP         ;nein, weiter
                00960
                00970 ;Variablenpointers laden und Ende:
804B 23         00980           INC     HL              ;auf neuen VLTPTR erhöhen
804C 22F940     00990           LD      (VLTPTR),HL
804F 22FB40     01000           LD      (ARRANF),HL
8052 22FD40     01010           LD      (ARREND),HL
8055 E1         01020           POP     HL              ;Stack korrigieren
8056 C3CC06     01030           JP      BASIC
                01040
06CC            01050           END     BASIC
00000 TOTAL ERRORS
32283  TEXT AREA BYTES LEFT


ARRANF 40FB 00260   01000
ARREND 40FD 00270   01010
ASCINT 1E5A 00220   00380
BASIC  06CC 00210   01030 01050
CNGPTR 802D 00700   00570
DATPTR 40FF 00230   00390 00710
LOOP   8037 00780   00950
PSTPTR 40A4 00240   00730
PUT    4182 00200   00310
PUTTO  8000 00360   00320
TO     00BD 00280   00370
UPLOAD 8021 00530   00490
VLTPTR 40F9 00250   00400 00990
```

ML

Die MEMORY SIZE automatisch

In einem meiner früheren Artikel war beiläufig davon die Rede, daß ein Maschinenprogramm sich selbsttätig vor BASIC schützen kann, indem es selber die obere Speichergrenze für BASIC setzt. Dies ist sogar sehr einfach, wie wir sehen werden, denn auch der Microsoft-Interpreter kocht nur mit Wasser. Zum Verständnis der Materie möchte ich jedoch zuvor erklären, woher BASIC "weiß", bis wohin es sich ausbreiten darf.

Das RAM unseres Computers ist an der Speicherstelle FFFFh (65535d) zuende. Dies gilt jedenfalls für 64KB-Systeme. Wenn die Memory Size nach dem Einschalten nicht definiert wurde (nur <ENTER> gedrückt), können alle BASIC-Bestandteile (Programmtext, numerische Variable, Stringvariable) bis dorthin anwachsen.

Anderfalls wird die Obergrenze an der Stelle 40B1/40B2h (16561/16562d) in der Communications Region abgelegt. Bei jedem Befehl, der BASIC anwachsen läßt (Einfügen einer Programmzeile, Zuweisung einer Variablen), wird zunächst geprüft, ob dafür bis zur Obergrenze noch genügend Platz ist. Falls nicht, wird ein "?OM Error" ausgegeben. Dies geschieht übrigens auch, wenn der Stack nach unten bis in die VLT (s. früheren Artikel) anwächst, wenn beispielsweise ein GOSUB oder eine FOR-NEXT-Schleife aktiv ist.

Die beiden Bytes, die die Memory Size halten, lassen sich ohne weiteres verändern, sogar mit POKE von BASIC aus. Man braucht daher nicht den Computer aus- und wieder einzuschalten (oder mit SYSTEM /0 die Einschaltroutine anzuspringen), wenn man während der Arbeit ein zu schützendes Programm nachladen will. Hier möchte ich als bekannt voraussetzen, wie man eine Integerzahl wie die Speichergröße in zwei Bytes aufteilt, um sie nach 40B1/40B2h zu laden.

Damit ist allerdings noch nicht alles getan. Hierzu ein Blick auf den oberen Speicherbereich: Ganz oben liegt ggf. der geschützte Bereich. Direkt darunter schließt sich die String Area an, der Bereich, wo Zeichenketten gespeichert werden, die nicht in der Form V$="..." im Programmtext stehen. Unterhalb beginnt der Stack, der sich nach unten vergrößert. Der Bereich zwischen der Stack-Untergrenze und der VLT ist frei.

Der String Space umfaßt nach dem Einschalten zunächst 50 Bytes. Mit CLEAR kann er nach Bedarf vergrößert werden. Seine Untergrenze (Memory Size minus 50 bzw. CLEAR-Argument) wird ebenfalls in der Communications Region abgelegt, und zwar an der Stelle 40A0/40A1h (16544/16545d). Wird diese Grenze durch Zuweisung von Zeichenketten unterschritten, wird ein "?OS Error" ausgegeben.

Beim Verändern der Memory Size muß nun zuletzt noch der Stack den neuen Werten angepaßt werden, denn er würde sonst unser zu schützendes Maschinenprogramm oder die Zeichenketten zuschaufeln. Hierzu wird der Stack Pointer (Stapelzeiger) auf den Wert der String-Untergrenze gesetzt. Daraus geht logischerweise hervor, daß der Stack im Augenblick der Veränderung nichts enthalten darf (z. B. GOSUB-Rücksprungadresse), denn der Zeiger deutet danach gewissermaßen in die falsche Richtung.

Leider kann man den Stack Pointer nicht mit einem BASIC-Befehl verbiegen. Ein kleines Maschinenprogramm, dessen entscheidender Bestandteil die Befehlsfolge

```
LD HL,(40A0H)
LD SP,HL
```

ist, kann aber durchaus über DATA eingelesen und angesprungen werden. Näher möchte ich darauf hier nicht eingehen, denn dieser Artikel ist ohnehin eher für den Assembler-Programmierer gedacht.

Das ist bereits alles. Abschließend wäre noch zu bemerken, daß das Programmsegment, das die Memory Size setzt, an den unteren Adressen des

Programme stehen sollte. Es ist nach der Initialisierung nämlich überflüssig geworden und braucht somit nicht vor BASIC geschützt zu werden. Das untenstehende Assembler-Programm gibt ein Beispiel. Das Hauptprogramm selbst löscht nur den Bildschirm (CALL 01C9H hätte genügt), denn es geht hier nur darum, die selbsttätige Manipulation der Memory Size zu demonstrieren.

```
                00100 ;PROGRAMM SETZT SEINE MEMORY SIZE SELBSTTAETIG
                00110
                00120 ;AUFGEWAERMT BY ARNULF SOPP. TEL. 0451-791926
                00130
                00140 ;SYMBOLVEREINBARUNGEN:
0033            00150 OUTACC   EQU     0033H    ;AKKUINHALT -> BILDSCHIRM
40B1            00160 MEMSIZ   EQU     40B1H    ;HIER IST DIE MEM SIZE ABGELEGT
40A0            00170 STRING   EQU     40A0H    ;DTO. STRING SPACE-UNTERGRENZE
001C            00180 HOME     EQU     1CH      ;VIDEO-STEUERCODE "HOME CURSOR"
001F            00190 CLEAR    EQU     1FH      ;DTO. "CLEAR TO END OF FRAME"
                00200
                00210
7000            00220          ORG     7000H             ;ODER WO AUCH IMMER
                00230
                00240 ;MEM SIZE, STRING SPACE, STACK EINNORDEN:
7000 210E70     00250 INIT     LD      HL,ANFANG-2       ;AB HIER GESCHUETZT
7003 22B140     00260          LD      (MEMSIZ),HL       ;-> COMMUN. REGION
7006 113200     00270          LD      DE,50             ;FUER "CLEAR 50"-AEQUIV.
7009 B7         00280          OR      A                 ;CARRY = 0 WEGEN SBC
700A ED52       00290          SBC     HL,DE             ;HL = MEM SIZE - 50
700C 22A040     00300          LD      (STRING),HL       ;STRING SPACE BEGRENZEN
700F F9         00310          LD      SP,HL             ;NEUER STACK-ANFANG
                00320
                00330 ;HAUPTPROGRAMM (WAHRLICH EINE EDV-OFFENBARUNG!):
7010 3E1C       00340 ANFANG   LD      A,HOME            ;CURSOR NACH LINKS OBEN
7012 CD3300     00350          CALL    OUTACC            ;AKKU -> BILDSCHIRM
7015 3E1F       00360          LD      A,CLEAR           ;ALLES DAHINTER LOESCHEN
7017 CD3300     00370          CALL    OUTACC            ;WIRD'S BALD?
701A F7         00380          RST     30H               ;UNTER DOS "LOAD DEBUG",
                00390                                    ;UNTER L2 BASIC-WARMSTART
                00400                                    ;VON HINTEN DURCH DAS
                00410                                    ;KNIE INS AUGE
                00420
7000            00430          END     INIT              ;START AB "INIT" MIT "/"
00000 TOTAL ERRORS
33940  TEXT AREA BYTES LEFT


ANFANG 7010 00340   00250
CLEAR  001F 00190   00360
HOME   001C 00180   00340
INIT   7000 00250   00430
MEMSIZ 40B1 00160   00260
OUTACC 0033 00150   00350 00370
STRING 40A0 00170   00300
```

Arnulf Sopp, Wakenitzstr. 3, D-2400 Lübeck 1, Tel.:0451-791926

Kaffeekochen ab sofort gestattet.

Dies sind eigentlich zwei Artikel. Ich will mich aber kurz- und beide zusammenfassen. Der eine handelt von einem Problem und seiner Lösung, der andere von dem wesentlich kniffligeren Problem, das nämliche zu lösen und wiederum dessen Lösung. Oder so ähnlich.

Jedenfalls haben Wolfgang Frey und ich vor kurzem ein bißchen Assembler trainiert. Wir schusterten eine kleine Routine zusammen, die bei jedem Tastendruck einen kurzen Rülpser auf den Lautsprecher ausgab. Derlei geht mit der Zeit freilich auf die Nerven. Um aber nicht die Zeit mit Spielkram vergeudet zu haben, zapte ich diese Routine jeweils ein wenig abgewandelt in SYS4/SYS und SYS13/SYS ein. Nun piept es nur noch, wenn ein DOS- oder BASIC-Fehler auftritt. Aber wirklich sinnvoll ist es in SYS6/SYS. Dort wird kopiert und formatiert.

Es dauert schon einige Minuten, bis z. B. die Systemdiskette inkl. Formatierung kopiert ist. Während dieser Zeit geht der Hacker normalerweise zum Briefkasten oder Kaffee kochen. Tritt nun aber ein Fehler auf, der mit der Eingabe "<A>bbruch, <W>iederholung, <F>ortfahren" beantwortet werden muß (G-DOS; in NEWDOS erscheint dergl. in Englisch), dann ist der Hacker woanders. Ein Rufsignal wie oben beschrieben wäre hochwillkommen.

Leider ist SYS6/SYS nahezu bis auf das letzte Bit voll. Eine Lärmroutine dieser Art hat ohne Änderungen keinen Platz mehr. Deshalb habe ich eine besonders lange Fehlermeldung abgekürzt: Aus "Schlechte Parameter oder Konflikt mit Pdrivedaten" wurde "PD/Par.!". Der so freigewordene Platz faßt die Krawallroutine leicht und läßt sogar noch ein paar Bytes Platz für schlechte Zeiten.

Dieses Pfeifprogramm ist auf der übernächsten Seite gelistet. Es geht von folgender Überlegung aus: Die Meldung "<A>bbruch ..." wird in SYS6/SYS an der Speicherstelle 58EBh mit dem Befehl CALL 4467h angezeigt. Stattdessen steht nun hier ein CALL nach 6E5Eh. Das ist im Bereich der genannten nunmehr abgekürzten Fehlermeldung. Zunächst wird die Anzeige nachgeholt, die im ersten Zap (oberer Sektordump, geänderte Adresse unterstrichen) unterdrückt worden ist. Sodann wird ein Tatütata aus zwei Tönen generiert. Dabei werden abwechselnd ein positives und ein negatives Signal auf den Cassettenport FFh gelegt. Die frequenzbestimmenden Verzögerungen zwischen zwei Signalen verhalten sich zur Tonlänge je nach Ton umgekehrt (oder wie drückt man das mathematisch korrekt aus?).

Nach jedem Tatü wird die Tastatur abgefragt. Wurde keine Taste gedrückt, ist der User wohl noch nicht zugegen. Dann lärmt es eben weiter. Andernfalls passiert dasselbe wie gewohnt: Nach irgendeinem Tastendruck und anschließend der Eingabe von A, W oder F tut der Computer, was er nicht lassen kann. Es mutet vielleicht befremdlich an, daß das Unterprogramm KRLOOP nicht mit RET abgeschlossen ist. Es ist! Am Ende der Routine ab 002Bh, die die Tastatur befragt, steht ein RETurnbefehl.

Die Hexcodes, die dieses Programm darstellen, sind in der zweiten Spalte des Assemblerlistings zu sehen. Sie finden sich wieder im unteren Sektordump. Dabei ist zu beachten, daß die ersten 9 unterstrichenen Codes die gekürzte Fehlermeldung sind. In der nächsten Zeile beginnt das eigentliche Programm.

Die zweite Geschichte handelt von der aufregenden Suche nach einem Caller, die schließlich in einem wahren Showdown endete. Die mehrfach zitierte Fehlermeldung, bei der es piepen sollte, wird über 4467h angezeigt. Dort steht der Befehl JP 4BA6h. Das Registerpaar Hl muß zu diesem Zweck als Zeiger mit der Adresse des Textes geladen werden. Das erste Byte dieser Meldung steht in 5A88h. So lag es nahe, mit SUPERZAP nach der Bytefolge 21-88-5A (LD HL,5A88H) zu suchen. Es gibt sie aber nicht. Da

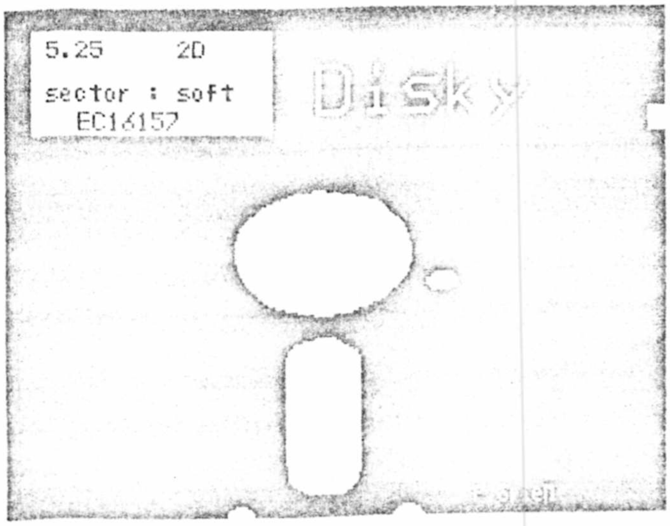
sind zu viele Möglichkeiten, HL mit einer davor oder dahinter liegenden Adresse zu laden, deshalb schrieb ich anstelle einer endlosen Suche eine kleine Routine, die mir die Arbeit abnehmen sollte:

Zunächst werden die benutzten Register in einen eigenen Puffer geschrieben, denn ein PUSH hätte die Ermittlung des Callers erschwert. Anschließend wird die RET-Adresse (Caller +3) vom Stack gepopt (und sofort wieder draufgepusht) und über die DOS-Routine 4063h angezeigt. Ein Blank dahinter sorgt für übersichtlichkeit. Wegen der Scrollerei während der DOS-Arbeit setzte ich vorsichtshalber die erste Anzeigestelle auf 3F00h. So können Zeilenvorschübe die angezeigte Adresse nicht zum Verschwinden bringen.

Das Resultat zeigt die Bildschirm-Hardcopy unter dem Assembler-Listing. Mitten im Bildschirm, durch Zeilenvorschübe des DOS auseinandergezogen, finden sich mehrere Hexzahlen. Die letzte vor der "erzwungenen Beendigung der Funktion" mit dem Kommando A lautet 58EEh. Demzufolge wurde die mehrfach erwähnte Fehlermeldung von 58EBh aus aufgerufen. Dort liegt jetzt als Kuckucksei der erste Zap, der den zweiten, das Krachprogramm, in Gang bringt. So einfach ist das. Oder so schwierig, je nach dem.

Den zweiten Teil dieses Beitrags bringe ich nicht, um die spannende Geschichte einer Verfolgungsjagd nach einer Adresse zu schildern. Vielmehr stellen sich derartige Probleme ständig bei der Zapperei. Dies ist ein Weg, sie zu lösen.

Arnulf Sopp, Tel. 0451-791926



```
000B00: 2195 59CB 7EC2 1A52 F6C0 CD09 44C3 C858 !.Y.ß..R....D..X
000B10: CDDA 57C0 DD34 0A20 03DD 340B AFC9 AFF6 ..W..4. ..4.....
000B20: 00C8 21E0 59CD 6744 CDDD 56CD A058 CD81 ..!.Y.gD..V..X..
000B30: 58F6 FFC9 CD67 443E 0DC3 BD58 7EFE 2023 X....gD>...Xß. #
000B40: 3804 FE80 3802 3E20 CDBD 5810 EFC9 3E20 8...8.> ..X...>
000B50: CDBD 5810 F9C9 7DC6 056F 0608 CDB3 587E ..X...ü..o....Xß
000B60: FE20 0603 3E2F C4BD 587E FE20 23C4 BD58 . ..>/..Xß. #..X
000B70: 10F7 C9D5 F5CD 3300 F1D1 C9CD 6744 3A95 ......3.....gD:.
000B80: 59CB 7FC2 4952 2184 5ACD E458 FE01 D03E Y...IR!.Z..X...>
000B90: 39C3 1A52 CD67 4421 B35A C5E5 7EB7 2320 9..R.gD!.Z..ß.#
000BA0: FBCD 5E6E CD2B 57E1 0EFF E534 0C35 28F4 ..^n.+W....4.5(.
000BB0: BE23 20F7 CDBD 0102 0059 58CD 8158 79B7 .# ......YX..Xy.
000BC0: E1C1 C901 0004 2133 59C5 4E23 4623 EB3E ......!3Y.N#F#.>
000BD0: 2F3C 0938 FCED 42C1 EBFE 3020 040C 0D28 /<.8,.B...0 ...(
000BE0: 040C CDBD 5810 E27B C630 C3BD 58F0 DB18 ....X..ä.0..X...
000BF0: FC9C FFF6 FF00 FF6F 4259 0007 0001 0000 .......oBY......

002100: 736B 6574 7465 2077 6972 6420 6B6F 7069 skette wird kopi
002110: 6572 740D 5369 6E64 2053 7973 7465 6D20 ert.Sind System
002120: 756E 6420 0320 6964 656E 7469 7363 683F und . identisch?
002130: 0344 6973 6B65 7474 6520 666F 726D 6174 .Diskette format
002140: 6965 7265 6E3F 0350 442F 5061 722E 210D ieren?.PD/Par.!.
002150: CD67 44C5 01C0 60C5 CD71 6EC1 7948 4728 .gD...`..qn.yHG(
002160: F6C1 C9C5 413E 01D3 FF10 FE41 3CD3 FF10 ....A>.....A<...
002170: FEC1 10EF C32B 0000 0000 0046 6F72 6D61 .....+.....Forma
002180: 7466 6568 6C65 7220 4672 6F6E 7473 6569 tfehler Frontsei
002190: 7465 2076 6F01 02A3 6E6E 2053 7075 7220 te vo...nn Spur
0021A0: 0346 726F 6E74 2052 7D63 6BCD C06E C8CD .Front Rück..n..
0021B0: D94C D8C8 2BC9 7EFE 0DC8 CDD9 4CD0 C31A .L..+.ß.....L...
0021C0: 527E FE3A 2001 237E D630 FE0A D0CD E76E Rß.: .#ß.0.....n
0021D0: CD76 4720 E97B 37C9 CDF1 6E18 03CD 0F6F .vG .ä7...n....o
0021E0: 7AB7 7BC8 C318 52E5 CD14 6F7E D641 FE08 z.ä...R...oß.A..
0021F0: 300D E106 01E5 CD16 6F7E FE48 2320 E5CB 0.......oß.H# ..
```

*6E55

(C) '84 by W. Frey & A. Sopp

```
6E5E            00100          ORG     6E5EH
6E5E CD6744     00110          CALL    4467H       ;Meldung anzeigen
6E61 C5         00120          PUSH    BC
6E62 01C060     00130          LD      BC,60C0H    ;2 Zähler: 60 und C0
6E65 C5         00140 KRACH    PUSH    BC          ;retten
6E66 CD716E     00150          CALL    KRLOOP      ;1 Schwingung erzeugen
6E69 C1         00160          POP     BC          ;zurückholen
6E6A 79         00170          LD      A,C         ;B und C vertauschen
6E6B 48         00180          LD      C,B
6E6C 47         00190          LD      B,A
6E6D 28F6       00200          JR      Z,KRACH     ;falls keine Taste gedr.
6E6F C1         00210          POP     BC          ;T. gedr.: Register rest.
6E70 C9         00220          RET                 ;weiter mit Fehlerbeh.
6E71 C5         00230 KRLOOP   PUSH    BC          ;1. Zähler retten
6E72 41         00240          LD      B,C         ;2. Zähler laden
6E73 3E01       00250          LD      A,1         ;posit. Signal
6E75 D3FF       00260          OUT     (0FFH),A    ;auf Krawallport
6E77 10FE       00270          DJNZ    $           ;Warteschleife
6E79 41         00280          LD      B,C         ;2. Zähler erneuern
6E7A 3C         00290          INC     A           ;Akku = 2, neg. Signal
6E7B D3FF       00300          OUT     (0FFH),A    ;auf Lärmport
6E7D 10FE       00310          DJNZ    $           ;Warteschleife
6E7F C1         00320          POP     BC          ;1. Zähler restaur.
6E80 10EF       00330          DJNZ    KRLOOP      ;bis 1 Schwingg. zuende
6E82 C32B00     00340          JP      002BH       ;Tast. abfr. (dort RET)
0000            00350          END
00000 mal gepennt
34328  Zeichen verfügbar
```

```
8000                   ORG     8000H           ;beliebige Adresse
8000 210980     START  LD      HL,DEVIAT       ;Adr. d. Umleitung laden
8003 226844            LD      (4468H),HL      ;Sprungbefehl verbiegen
8006 C32D40            JP      402DH           ;retour ins DOS
8009 223880     DEVIAT LD      (HLBUFF),HL     ;Register retten
800C ED533A80          LD      (DEBUFF),DE
8010 323C80            LD      (ABUFF),A
8013 2A3680            LD      HL,(VDBUFF)     ;Bildschirmstelle laden
8016 D1                POP     DE              ;Caller ermitteln
8017 D5                PUSH    DE              ;RET-Adresse restaurieren
8018 CD6340            CALL    4063H           ;DE in Hex anzeigen
801B 3E20              LD      A,' '           ;Blank zwischen den
801D 77                LD      (HL),A          ;  Adressen anzeigen
801E 23                INC     HL              ;nächste Bildschirmstelle
801F 7C                LD      A,H             ;Screen zuende, d. h.
8020 FE40              CP      40H             ;  HL >= 4000h?
8022 2002              JR      NZ,GOON         ;keine Panik, falls nein
8024 263F              LD      H,3FH           ;sonst wieder 3F00h
8026 223680     GOON   LD      (VDBUFF),HL     ;neue Stelle merken
8029 2A3880            LD      HL,(HLBUFF)     ;Register restaurieren
802C ED5B3A80          LD      DE,(DEBUFF)
8030 3A3C80            LD      A,(ABUFF)
8033 C3A64B            JP      4BA6H           ;jetzt Meldung anzeigen
8036 003F       VDBUFF DEFW    3F00H           ;hält Bildschirmadresse
0002            HLBUFF DEFS    2               ;Puffer für Register
0002            DEBUFF DEFS    2
0001            ABUFF  DEFS    1
8000                   END     START           ;Einsprungadresse
00000 mal gepennt
34106  Zeichen verfügbar
```

G TCS- DOS 2.1c

4DF5

6639

2.1b – mod.
1984 durch
Arnulf Sopp

```
Komm, hau 55C7 55CC 55D8

Komm, hau rein:ndf 1

Diskette wird formatiert 6810

" NEW LINE ", wenn Zieldiskett693E 6944 erk Nr.  1
                                          58EE

Diskette hat Daten

Zieldiskettenname, -datum: G-DOS FC     SOFTSOPP

<A>bbruch, <W>iederholung, <F>ortfahren

A                                         4DF5


Erzwungene Beendigung der Funktion

Komm, hau rein:
```

(C) '84 by A. Sopp

BEL ohne Nachladen gleich bei BOOT

Im letzten Artikel zu diesem Thema stellte ich ein Programm vor, das den ASCII-Code 07 nutzbar macht, den unsere Maschinen normalerweise ignorieren. Er wird BEL genannt und führt bei der Ausgabe "auf den Bildschirm" (mit PRINT oder einem entsprechenden Maschinenbefehl) zu einem akustischen Signal.

Wie bereits angedroht, folgt hier eine Version, die Bestandteil von SYS0/SYS ist und daher gleich beim Booten aktiviert wird. Sie residiert im Adreßbereich des L2-ROMs an einer Stelle, die nur während der IPL-Sequenz unmittelbar nach dem Einschalten benötigt wird. Dieser Speicherbereich ist daher Sekundenbruchteile nach dem Druck auf den Knopf frei. Da man im ROM aber bekanntlich nicht schreiben kann, muß der EG 64 MBA angeschlossen sein, der das dort liegende RAM zugänglich macht.

Zum Programm selbst muß hier nicht mehr viel gesagt werden, das war im vorigen Beitrag hoffentlich ausführlich genug. Auch hier wird an der Stelle CPBEL zunächst geprüft, ob ASCII 07 anliegt und gepfiffen, falls ja. Der Unterschied besteht in der Einsprungstelle: Im Videotreiber wird an der Stelle 0506 DE mit 0480 geladen und dieser Wert als RET-Adresse auf den Stack gepusht. Hier steht nun aber der CALL nach CPBEL. Daher muß in Zeile 670 der Befehl LD DE,0480H ausgeführt werden, falls im Akku nicht 07 stand. Diesen notwendigen Befehl habe ich ja mit meinem CALL einfach übertüncht.

Dieses Programm ist demnach ein Eingriff mitten im Interpreter, der ohne Memory Banking nicht denkbar wäre. Der Vorteil liegt darin, daß der Anwenderspeicher oberhalb des ROM-Adreßbereichs nicht tangiert wird. Die einfachere Routine vom letzten Mal besetzt nun einmal leider das Himem.

Zum Verständnis des Zaps ist er im Listing komplett wiedergegeben, nicht nur der BEL-Bestandteil. Im ersten Teil wird zerst geprüft, ob der Linkspfeil gedrückt wurde. Mit ihm kann man nämlich verhindern, daß die ganzen Änderungen aktiv werden. Also ein abschaltbarer Zap, sozusagen. Ohne Linkspfeil wird zunächst das ROM auf das RAM kopiert, das RAM im I/O-Adreßbereich wird auf 00 gesetzt. Ab Zeile 340 wird nun das Kukkucksei namens NEWCOD in den Interpreter gelegt. Die Routinen, die es anspringen sollen (Videotreiber und INT-Service-Routine) werden entsprechend verbogen. Übrigens ist die INT-Ergänzung für das Erkennen der Dreitastenbefehle für das Banking und den Spooler (s. frühere Infos) gut. Zum genaueren Verständnis bitte ich den Leser, diese Routinen an 0506 (mitten im Videotreiber) und 45D3 (mitten in der INT-Bearbeitung) selbst zu disassemblieren. Das würde hier zu weit führen.

Die beiden Sektordumps zeigen den Zap in den letzten beiden Sektoren von SYS0/SYS. Wie üblich sind die unterstrichenen Codes zu ändern. Dieser neue Programmteil wird als Unterprogramm aufgerufen aus 4F2A. Deshalb muß dort der Befehl LD A,(3840H) ersetzt werden durch CALL 50A8H. Dieser zusätzliche Zap steht im Sektor 0C von SYS0/SYS an den Bytes 43/44/45H. Hier sind die Codes CD-A8-50 einzuzappen. Dieser Zap ist nicht als Sektordump abgebildet, um für 3 Bytes das Info nicht unnötig dick zu machen.

Mir ist klar, daß kaum jemand von Euch den MBA hat. Mit meinen Beiträgen möchte ich Euch deshalb für das Ding interessieren, denn man kann sein Geld kaum besser anlegen.

Arnulf Sopp, Tel. 0451-791926

```
0072        00010 DESTIN  EQU   0072H            ;dort NEWCOD ablegen
50A8        00100         ORG   50A8H            ;am Ende von SYS07SYS
50A8 3A4038 00110         LD    A,(3840H)        ;Tastatur abfragen
50AB CB6F   00120         BIT   5,A              ;Linkspfeil gedrückt?
50AD C0     00130         RET   NZ               ;nichts veränd., falls ja
50AE E5     00140         PUSH  HL               ;benutzte Register retten
50AF F5     00150         PUSH  AF
50B0 F3     00160         DI                     ;bloß keine Störungen!
50B1 0604   00170         LD    B,04H            ;4 Codes auf MBA ausgeben
50B3 3E0E   00180         LD    A,0EH            ;1. Code
50B5 D3DF   00190 BANK    OUT   (0DFH),A         ;auf Banking-Port
50B7 3D     00200         DEC   A                ;nächster Code
50B8 10FB   00210         DJNZ  BANK             ;usw.
50BA 3D     00220         DEC   A                ;0A aussparen
50BB D3DF   00230         OUT   (0DFH),A         ;09 ausgeben
50BD 010036 00240         LD    BC,3600H         ;Zähler für L2/4-ROM
50C0 61     00250         LD    H,C              ;H=00
50C1 69     00260         LD    L,C              ;HL=0000
50C2 51     00270         LD    D,C              ;D=00
50C3 59     00280         LD    E,C              ;DE=0000
50C4 EDB0   00290         LDIR                   ;ROM auf RAM kopieren
50C6 70     00300         LD    (HL),B           ;(3600)=00
50C7 1C     00310         INC   E                ;Ziel DE=3601
50C8 01FF09 00320         LD    BC,09FFH         ;Zähler für I/O-Bereich
50CB EDB0   00330         LDIR                   ;dort Nullen einschreiben
50CD 21FB50 00340         LD    HL,NEWCOD        ;Anfang Zap für "ROM"
50D0 117200 00350         LD    DE,DESTIN        ;Ziel des Zaps
50D3 D5     00360         PUSH  DE               ;brauchen wir noch
50D4 012D00 00370         LD    BC,FINITO-NEWCOD         ;Länge des Zaps
50D7 EDB0   00380         LDIR                   ;Zap übertragen
50D9 3EC3   00390         LD    A,0C3H           ;JP-Opcode
50DB 320605 00400         LD    (0506H),A        ;Videotreiber verwanzen
50DE 218200 00410         LD    HL,CPBEL-OFFSET  ;0506: JP 0082
50E1 220705 00420         LD    (0507H),HL       ;Sprungadresse ablegen
50E4 DBDF   00430         IN    A,(0DFH)         ;reset MBA
50E6 3E08   00440         LD    A,08H            ;read RAM 0000-2FFF
50E8 D3DF   00450         OUT   (0DFH),A         ;auf Banking-Port
50EA 3E0F   00460         LD    A,0FH            ;reset nicht mit R-Taste
50EC D3DF   00470         OUT   (0DFH),A         ;auch ausgeben
50EE E1     00480         POP   HL               ;HL=0072
50EF 3ECD   00490         LD    A,0CDH           ;CALL-Opcode
50F1 32D345 00500         LD    (45D3H),A        ;in INT-Service-Routine
50F4 22D445 00510         LD    (45D4H),HL       ;45D3: CALL 0072
50F7 F1     00520         POP   AF               ;Register restaurieren
50F8 E1     00530         POP   HL
50F9 FB     00540         EI                     ;INT wieder zulassen
50FA C9     00550         RET                    ;und zuende booten
5089        00560 OFFSET  EQU   $-DESTIN         ;Subtrahend f. Relokation
50FB 3A2038 00570 NEWCOD  LD    A,(3820H)        ;Tastaturabfrage
50FE FED0   00580         CP    0D0H             ;<,./> gedrückt?
5100 2806   00590         JR    Z,RST28          ;verarbeiten, falls ja
5102 3A1038 00600         LD    A,(3810H)        ;Tastaturabfrage
5105 FEE0   00610         CP    0E0H             ;<567> gedrückt?
5107 C0     00620         RET   NZ               ;norm. weiter, falls nein
5108 F61C   00630 RST28   OR    1CH              ;Akku für RST 28 vorber.
510A EF     00640         RST   28H              ;und SYS26/SYS laden
510B FE07   00650 CPBEL   CP    07H              ;BEL-Code?
510D 2806   00660         JR    Z,BEL            ;falls ja
510F 118004 00670         LD    DE,0480H         ;überschriebener Befehl
5112 C30905 00680         JP    0509H            ;dahinter weiter
5115 0EB0   00690 BEL     LD    C,0B0H           ;Anfangswert f. Schleifen
5117 3E01   00700 BEEP    LD    A,01H            ;positiver Impuls
5119 D3FF   00710         OUT   (0FFH),A         ;auf Port ausgeben
511B 10FE   00720         DJNZ  $                ;etwas warten
511D 41     00730         LD    B,C              ;Schleifenzähler erneuern
511E 3C     00740         INC   A                ;A=2, negativer Impuls
511F D3FF   00750         OUT   (0FFH),A         ;ausgeben
5121 10FE   00760         DJNZ  $                ;ein wenig trödeln
5123 0D     00770         DEC   C                ;Zähler erniedrigen
5124 41     00780         LD    B,C              ;und laden
5125 10F0   00790         DJNZ  BEEP             ;bis Ton zuende
5127 C9     00800         RET                    ;ins Betriebssystem
5128        00810 FINITO  EQU   $
0000        00820         END
00000 mal gepennt
32739 Zeichen verfügbar                       C. '84 by Sopp
```

```
000D00: BF01 00E8 4F80 BCBC 2054 4353 2D20 BFC2  ...O   TCS- .
000D10: BFC2 BFC2 BFC2 8B8C 8CB4 C400 0000 0000  . . .   ......
000D20: 0000 A09E 81C1 8020 8080 BF20 C020 2020  ..  .     .
000D30: BE83 838D C431 3938 34C2 6475 7263 680A    .1984.durch.
000D40: AFBC BC9F C6BF BCBC 9F20 20AF BCBC 9FC2   .     .
000D50: AFBC BC9F C3B8 BFBC BC20 A894 C208 BCBF   .     ..
000D60: BC20 C120 AFB0 B09C 2020 C241 726E 756C   .       .Arnul
000D70: 6620 536F 7070 CE0D 0000 0000 0000 0000  f Sopp..........
000D80: 4441 5455 4D3F 2028 5454 2E4D 4D2E 4A4A  DATUM? (TT.MM.JJ
000D90: 2920 035A 4549 543F 2020 2848 483A 4D4D  ) .ZEIT?  (HH:MM
000DA0: 3A53 5329 2003 5454 2E4D 4D2E 4A4A 2020  :SS) .TT.MM.JJ
000DB0: 4848 3A4D 4D3A 5353 0D01 1F01 0C53 6400  HH:MM:SS.....Sd.
000DC0: 1800 3C00 3C3A 4038 CB6F C0E5 F5F3 0604  ..<.<:§8.o......
000DD0: 3E0E D3DF 3D10 FB3D D3DF 0100 3661 6951  >...=..=....6aiQ
000DE0: 59ED B070 1C01 FF09 EDB0 21FB 5011 7200  Y. p..... !.P.r.
000DF0: D501 2D00 EDB0 3EC3 3206 0521 8200 2207  ..-.. >.2..! .".

000E00: 05DB DF01 F7E6 503E 08D3 DF3E 0FD3 DFE1  ......P>...>....
000E10: 3ECD 32D3 4522 D445 F1E1 FBC9 3A20 38FE  >.2.E".E....: 8.
000E20: D028 063A 1038 FEE0 C0F6 1CEF FE07 2806  .(.:.8........(.
000E30: 1180 04C3 0905 0EB0 3E01 D3FF 10FE 413C  . .....  >.....A<
000E40: D3FF 10FE 0D41 10F0 C900 0000 0000 0000  .....A..........
000E50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000E60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000E70: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000E80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000E90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000EA0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000EB0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000EC0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000ED0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000EE0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000EF0: 0000 0000 0000 0000 0000 0000 0202 004D  ...............M
```

DSMBLR/CMD - ein wenig komfortabler

Es ist nicht besonders sinnvoll, Utilities hier im Info vorzustellen, denn bei unseren Tauschaktivitäten (legal oder nicht) hat eh' jeder früher oder später alles. Aber interessante Änderungen in Dienstprogrammen sind das Papier schon wert.

DSMBLR/CMD ist ein Disassembler, der (u. a.) EDTASM-kompatiblen Sourcecode erstellt. Damit ist es möglich, fremde Maschinenprogramme praktisch beliebig zu verändern (sofern man sich in die Programme eingedacht hat und ihren Ablauf versteht, damit man keine wichtigen Funktionen ruiniert).

Wenn die DSMBLR-Source von EDTASM "verstanden" werden soll, muß sie einen Header von 6 Zeichen und Zeilennummern haben. Zu diesem Zweck wird DSMBLR mit der Syntax DSMBLR,(HEADER,NUMBER) aufgerufen. Das ist ein bißchen umständlich. Ein Job, der mit DO,DIS aufgerufen wird und dieses Bandwurmkommando enthält, ist bequemer. Noch simpler ist die Lösung, die in den beiden Listings auf der nächsten Seite vorgestellt ist: Ein kleines Programm wird mitten in den DOS-Befehlspuffer geladen. Es enthält zunächst das Aufrufkommando und anschließend zwei Befehle, die den Aufruf bewerkstelligen (Zz. 5 und 6). Jetzt wird nur noch DIS eingegeben.

Der String in Zz. 2 und 3 kürzt die Optionen HEADER und NUMBER ab, was erlaubt ist. Außerdem enthält er weitere Parameter, auf die ich aber nicht weiter eingehen will. Sie sind hier nicht wichtig. Da solche Strings mit einer Länge von 18 Zeichen dargestellt werden, hat DSMBLR aus dieser Zeichenkette zwei Zeilen gemacht.

Die beiden Listings zeigen, wie DSMBLR arbeitet. Nach zwei Kopfzeilen, die auf jeder Seite einer Disassembly wiederholt werden, folgt der Code. Auf einfache Weise kann Daten- und Befehlscode unterscheidbar gemacht werden. Leider müssen aber die Pseudo-Ops DW (kommt hier nicht vor) und DB mit dem Editor/Assembler je nach dem in DEFB, DEFW oder DEFM verwandelt werden, damit die Syntax für EDTASM paßt. Um nahezu beliebig wursteln zu können, werden alle Adressen in Labelform wiedergegeben.

An den Kopfzeilen ist zwar nicht viel auszusetzen, aber auf deutsch lesen sie sich leichter. Das zweite Linefeed vor der ersten Programmzeile stört die optische Verbindung der Spaltenüberschriften und der Spalten selbst. Zusätzlich kann man, wenn man mag, eine persönliche Kopfzeile wie im unteren Listing entwerfen. Kleiner Schönheitsfehler: So wie diese Änderung sieht dann auch das Hello des Disassemblers aus, denn es wird aus eben diesem String erzeugt. Na und? Die kleine Routine ist übrigens kein "Fremdprogramm". Ich habe dieses Wort in die Zeile aufgenommen, weil ich von eigenen Programmen die Source habe und in diesen Fällen nicht auf DSMBLR angewiesen bin. Dies ist ja nur eine Demonstration.

Die notwendigen Änderungen, die im relativen Sektor 14h (20d) durchgeführt werden, gehen aus den beiden Hexdumps hervor (alt oben, neu unten). Das letzte Byte ist das P von "PAGE" bzw. das S von "Seite". Die ersten Bytes des nächsten Sektors müssen entsprechend geändert werden ("AGE" mit "eite" überschreiben). Eine zusätzliche Änderung ist sinnvoll, wenn der Computer/Drucker kleine Umlaute und ß hat: Im relativen Sektor 0Eh (14d) des Files wird das relative Byte 17h von 7B auf 7F gezapt. Jetzt werden diese Codes ausgedruckt und nicht durch Punkte ersetzt.

Arnulf Sopp, Tel. 0451-791926

```
MISOSYS Disassembler - Disk Version 3.0  DIS/CMD        PAGE  00001

ADDR CONTENTS LINE# LABEL  INSTRUCTION        ASCII


                00001        ORG   4318H
4318            00002 M4318  DB    'DSMBLR (C,H,L=62,N'
432A            00003        DB    ',S=32)'
4330            00004        DB    0DH
4331 211843     00005 M4331  LD    HL,M4318       !.C
4334 C30544     00006        JP    M4405          C.D
                00007 M4405  EQU   4405H
                00008 MFFFF  EQU   0FFFFH
                00009        END   M4331
```

C. 1984 A. Sopp

```
SOFTSOPP Software  -  Fremdprogramm:      DIS/CMD        Seite 00001

Adr. Inhalt     Zeile Symbol Befehl             ASCII

                00001        ORG   4318H
4318            00002 M4318  DB    'DSMBLR (C,H,L=62,N'
432A            00003        DB    ',S=32)'
4330            00004        DB    0DH
4331 211843     00005 M4331  LD    HL,M4318       !.C
4334 C30544     00006        JP    M4405          C.D
                00007 M4405  EQU   4405H
                00008 MFFFF  EQU   0FFFFH
                00009        END   M4331
```

Tel. 0451-791926

______ wichtig, ... optional

```
001400: 2061 6E64 2045 6E74 6572 203C 482C 4C3E  and Enter <H,L>
001410: 0320 2020 2020 203A 2053 7461 7274 3D78 .      : Start=x
001420: 7878 782C 2045 6E64 0100 D867 3D78 7878 xxx, End...g=xxx
001430: 782C 2054 7261 6E73 6665 723D 7878 7878 x, Transfer=xxxx
001440: 0D50 726F 6772 616D 2077 696C 6C20 6F76 .Program will ov
001450: 6572 7772 6974 6520 4469 7361 7373 656D erwrite Disassem
001460: 626C 6572 0D52 6561 6479 2070 7269 6E74 bler.Ready print
001470: 6572 2061 6E64 2065 6E74 6572 2074 6974 er and enter tit
001480: 6C65 0D41 4444 5220 434F 4E54 454E 5453 le.ADDR CONTENTS
001490: 204C 494E 4523 204C 4142 454C 2020 494E  LINE# LABEL  IN
0014A0: 5354 5255 4354 494F 4E20 2020 2020 2041 STRUCTION      A
0014B0: 5343 4949 0D50 726F 6365 7373 696E 6720 SCII.Processing 
0014C0: 7061 7373 2031 0D0A 4D49 534F 5359 5320 pass 1..MISOSYS 
0014D0: 4469 7361 7373 656D 626C 6572 202D 2044 Disassembler - D
0014E0: 6973 6B20 5665 7273 696F 6E20 332E 3003 isk Version 3.0.
0014F0: 2020 2020 2020 2020 2020 2020 2020 2050                P

001400: 2061 6E64 2045 6E74 6572 203C 482C 4C3E  and Enter <H,L>
001410: 0320 2020 2020 203A 2053 7461 7274 3D78 .      : Start=x
001420: 7878 782C 2045 6E64 0100 D867 3D78 7878 xxx, End...g=xxx
001430: 782C 2054 7261 6E73 6665 723D 7878 7878 x, Transfer=xxxx
001440: 0D50 726F 6772 616D 2077 696C 6C20 6F76 .Program will ov
001450: 6572 7772 6974 6520 4469 7361 7373 656D erwrite Disassem
001460: 626C 6572 0D52 6561 6479 2070 7269 6E74 bler.Ready print
001470: 6572 2061 6E64 2065 6E74 6572 2074 6974 er and enter tit
001480: 6C65 0D41 6472 2E20 496E 6861 6C74 2020 le.Adr. Inhalt  
001490: 205A 6569 6C65 2053 796D 626F 6C20 4265  Zeile Symbol Be
0014A0: 6665 686C 2020 2020 2020 2020 2020 2041 fehl           A
0014B0: 5343 4949 0D50 726F 6365 7373 696E 6720 SCII.Processing 
0014C0: 7061 7373 2031 0D0A 534F 4654 534F 5050 pass 1..SOFTSOPP
0014D0: 2053 6F66 7477 6172 6520 202D 2020 4672  Software  -  Fr
0014E0: 656D 6470 726F 6772 616D 6D3A 2020 2003 emdprogramm:   .
0014F0: 2020 2020 2020 2020 2020 2020 2020 2053
```

1

Mehr über DDE

Die G-DOS-Anleitung ist notorischerweise ein Groschenheft. Wer nicht total computerkrank ist und nach und nach sein DOS selber auseinanderpflückt, ahnt nichts von dem, was noch drinsteckt.

DDE steht in SYS15/SYS und ist eine Utility, die die Inspektion einer Diskette erlaubt, ähnlich SUPERZAP von NEWDOS-80. Dabei können lt. Anleitung nur Dateien aufgerufen werden, keine Sektoren "an sich", was jedoch SUPERZAP erlaubt. In der Anzeige eines Sektors steht links auch nur die relative Sektornummer des gerade geladenen Files, nicht die absolute Sektornummer der Diskette. Sagt die Anleitung.

Da in SYS15/SYS noch reichlich Platz ist, wollte ich DDE ein wenig aufmotzen (Druckerausgabe, Anzeige der absoluten Sektornummer usw.). Der erste Schritt dazu ist immer eine Disassembly, um zu sehen, wo was geschieht. Beim Lesen des Quelltextes staunte ich nicht schlecht:

DDE residiert von 5200-54FF. Zusätzlich reserviert es 256 Bytes ab 5500 für einen systeminternen I/O-Puffer. Weshalb nicht einfach der DOS-Puffer an 4200 hergenommen wird, bleibt dunkel. Das bedeutet, daß Programme im unteren Anwenderbereich gnadenlos zugeschaufelt werden. Unser G-DOS-Manual (welch Wort für das!) weiß nichts davon.

Es gibt zusätzliche Funktionen, die dort ebenfalls nicht erläutert sind. Bei der Frage nach dem Dateinamen wird mit # die zuletzt angezeigte Datei "vergessen", aber ihr relativer Sektor nicht. Jetzt steht dieser relative Sektor, aber von der ganzen Diskette, auf dem Display. Wurde nach # eine Laufwerksnummer angegeben, wird dieses Laufwerk angewählt. Nur mit NEW LINE wird Lw. 1 geschaltet, weil 0D (NEW LINE) für die Nummer gehalten und mit 03 UND-verknüpft wird.

Beispiel: Wurde zuletzt der relative Sektor 04 des Files PROG/BAS untersucht, so steht da nach #2 der relative Diskettensektor 04 von Laufwerk 2 auf dem Bildschirm. Das ist Sektor 04 von GDOS/SYS.

In diesem Modus können alle bekannten Funktionen abgerufen werden. Mit Shift ;/+ kommt man normalerweise ans Ende einer Datei. In diesem Modus wird ebenfalls ein sehr hoher Sektor angezeigt, nämlich immer 018F, der aber natürlich nichts mehr mit GDOS/SYS zu tun hat. Das Ende der Diskette ist es auch nicht, denn mit 80/DS/DD habe ich 2880 (0B40) Sektoren. Diese Sektornummer ist an der Adresse 54EA in zwei Bytes niedergelegt, das sind die relativen Bytes 02 und 03 des rel. Sektors 03 von SYS15/SYS. Klar, daß ich sie sofort in 40 und 0B umzappte. Bisher läuft damit alles normal. Nichts spricht dagegen, auch bei anderen Spurenzahlen und Dichten diese Möglichkeit auszunutzen. Man denke aber bitte an die Reihenfolge LSB-MSB.

Eine weitere Funktion ist der Anleitung nicht bekannt: Mit * anstelle eines Dateinamens wird der zuletzt bearbeitete Sektor angezeigt. Dabei darf gerne zwischen zwei DDE-Aufrufen allerhand andere EDV gelaufen sein. Bedingung ist allerdings, daß der untere Bereich des Anwender-RAMs nicht verändert wurde. DDE legt dort nämlich die Sektoradressen und noch einiges auf Eis. Übrigens macht das BASIC genauso; der Befehl BASIC * ruft BASIC/CMD aus dem DOS auf und beläßt alle alten Zeiger und den Programmtext.

Eine dritte Besonderheit ist nicht gerade aufregend und bietet keine nennenswerte Bereicherung der DDE-Routine. Gleichwohl gehört das in die Anleitung: Beim S-Kommando, mit dem man einen bestimmten Sektor des Files aufrufen kann, wird jede Eingabe < ASCII 31h ("1") als Anwahl des Sektors 0 verstanden. Zusätzlich wird NEW LINE überflüssig, wenn die Eingabe < ASCII 21h ist. Um also in den ersten Sektor einer Datei zu

kommen, kann man statt Shift -/= auch z. B. S und Blank eingeben.

Ein neues Manual ist von der Fa. TCS angekündigt. Es sei vollständiger, heißt es. Aber der Käufer des alten Heftchens bleibt angeschmiert, denn der Kaufvertrag des G-DOS scheint bisher beinhaltet zu haben, daß der Käufer auf eine vernünftige Anleitung verzichtet. Auf Deutsch: Jeder kann das neue Manual bekommen - gegen Cash.

Bekanntlich verbraucht jede Datei auf einer Diskette mindestens eine Einheit, also fünf Sektoren. Das bedeutet, daß alle Sektoren, die beim Schreiben eines Files zum Vielfachen von fünf Sektoren noch fehlen, für alle Zeiten verloren sind (jedenfalls ohne drastische Änderung des Systems). So ist es unverständlich, daß ausgerechnet eine SYS-Datei, nämlich SYS15/SYS, nur vier Sektoren belegt.

Man könnte sich damit abfinden, denn sogar bei 40/SS/SD hat eine Platte noch immer 720 Sektoren. Systemdateien haben jedoch einen unschätzbaren Wert, denn sie können mit dem Befehl RST 28h in eigenen Programmen ohne Tastatureingabe aufgerufen werden, ohne sich der diffizilen Handhabung der Disk-I/O bedienen zu müssen. So ist es um den fünften Sektor in SYS15/SYS besonders schade.

Aber dagegen ist ein Kräutlein gewachsen. Leider ist dabei diesmal die Zapperei nicht nur auf die Zieldatei SYS15/SYS beschränkt. Gleichzeitig müssen zwei Bytes in INHALT/SYS geändert werden. Es handelt sich um die Bytes 42 und 54 (beide hex) im relativen Sektor 03 von INHALT/SYS. Das Letztere gibt schlicht die Anzahl der vom File SYS15/SYS belegten Sektoren wieder. So wird aus 04 eben 05. Das Erstere ist das EOF-Byte. Es besagt, im wievielten Byte des letzten Sektors der Datei das File zuende ist. Es zeigt also auf das erste nicht mehr zur Datei gehörige Byte. Da in meiner Modifikation der fünfte Sektor bis zum letzten Bit ausgenutzt wird, muß das EOF-Byte 00 lauten, das entspricht 256.

SYS15/SYS enthält nur die Routine zur Bearbeitung des DOS-Befehls DDE. Es liegt daher nahe, die Änderungen nicht ausgerechnet mit DDE durchzuführen. SUPERZAP ist ohnehin besser (wenngleich wesentlich unbequemer). Mit DDE geht es dann, wenn man die Zaps im Laufwerk 1 oder höher einbringt. Es empfiehlt sich, zunächst den Dateieintrag im Inhaltsverzeichnis zu ändern (oberer Sektordump). Dabei bleibt DDE voll funktionsfähig und kann notfalls sich selbst bearbeiten (nicht übel, wenn man nur ein Laufwerk und kein SUPERZAP hat).

Die Änderungen im Sektor 03 von SYS15/SYS sind kein Problem; dergleichen wurde schon in diversen Infos vorgestellt. Wer schon gelegentlich zapte, wird sich wundern, daß es selbst mit DDE auch im (überhaupt nicht belegten) Sektor 04 keine Kunst ist, obwohl das EOF im Sektor 03 bereits definiert ist. Das liegt eben daran, daß das soeben bereits verwanzte Inhaltsverzeichnis DDE glauben macht, es habe 5 Sektoren. Man kommt deshalb mit dem gewohnten Druck auf die ";"-Taste in den nachfolgenden Sektor. Der Rest ist Handwerk.

Im ersten Teil dieses Beitrags steht zu lesen, daß SYS15/SYS den Platz von 5200-55FF beansprucht (inkl. Sektorpuffer). Um nicht ohne Not bei jedem Laden des Files noch mehr Speicherplatz zu besetzen, sollte man die durch den zusätzlichen Sektor gewonnenen Records als das definieren, was in BASIC REM heißt. Der Record-Header 05 (im mittleren und unteren Sektordump unterstrichen) sorgt dafür. So wird weiterer Platz erst beansprucht, wenn später entsprechende Zaps hinzukommen. Da noch der DOS-Eingabepuffer mit seinen 80 Zeichen frei ist, sollte man eigene Routinen in SYS15/SYS zunächst dorthin legen. Der Sektorpuffer ab 4200 wird von DDE ebenfalls nicht gebraucht, so daß auch dort Platz ist. Und schließlich ist der ganze Overlay-Bereich 4D00-51FF frei, wenn die neue Routine keine weiteren DOS-Moduln nachladen soll.

Auf jeden Fall ist jetzt in SYS15/SYS Platz, und um den wäre es schade gewesen. Versteht dies bitte nicht als Preisausschreiben, aber wer hat eine Idee, was man in diesen Raum legen könnte? Ihr müßt hierzu nicht gleich die fertigen Maschinenprogramme abliefern, so daß auch die BASIC-Spezialisten unter Euch über Bedarfslücken im DOS nachdenken können.

Arnulf Sopp

INHALT/SYS, rel. Skt. 03

```
000300: 5D20 0000 0049 4E48 414C 5420 2053 5953 Ü ...INHALT  SYS
000310: A71D F9E5 1E00 3005 FFFF FFFF FFFF FFFF ......0.........
000320: 5F20 0000 0053 5953 3720 2020 2053 5953 _ ...SYS7    SYS
000330: 5678 1234 0500 2E80 FFFF FFFF FFFF FFFF Vx.4...........
000340: 5F20 0000 0053 5953 3135 2020 2053 5953 _ ...SYS15   SYS
000350: 5678 1234 0500 2E40 FFFF FFFF FFFF FFFF Vx.4...§........
000360: 5F20 0000 0053 5953 3233 2020 2053 5953 _ ...SYS23   SYS
000370: 5678 1234 0500 2D80 FFFF FFFF FFFF FFFF Vx.4..-.........
000380: 1020 00F4 0047 4553 4348 4252 494B 5046 . ...GESCHBRIKPF
000390: 9642 9642 0500 0160 FFFF FFFF FFFF FFFF .B.B...`........
0003A0: 1020 0000 0052 4F55 5445 2020 2041 534D . ...ROUTE   ASM
0003B0: 9642 9642 1100 0183 FFFF FFFF FFFF FFFF .B.B............
0003C0: 1020 00FC 0041 5254 494B 454C 3220 2020 . ...ARTIKEL2
0003D0: 9642 9642 1100 3883 FFFF FFFF FFFF FFFF .B.B..8.........
0003E0: 0020 0017 004C 5052 494E 5420 2043 4D44 . ...LPRINT  CMD
0003F0: 9642 9642 0100 3F20 FFFF FFFF FFFF FFFF .B.B..? ........
```

SYS15/SYS, rel. Skt. 03

```
000300: 0000 400B FFFF FFFF FFFF FFFF FFFF FFFF ..§.............
000310: FFFF FFFF FFFF 0500 0000 0000 0000 0000 ................
000320: 496E 2064 6965 7365 6D20 4265 2D20 2020 In diesem Be-
000330: 7265 6963 6820 6465 7320 4669 6C65 7320 reich des Files
000340: 5359 5331 352F 5359 5320 7374 6568 7420 SYS15/SYS steht
000350: 6B65 696E 207A 7520 6C61 6465 6E64 6572 kein zu ladender
000360: 4D61 7363 6869 6E65 6E63 6F64 652E 2020 Maschinencode.
000370: 4B65 6E6E 746C 6963 6820 6475 7263 6820 Kenntlich durch
000380: 6465 6E20 5265 636F 7264 2D48 6561 2D20 den Record-Hea-
000390: 6465 7220 3035 2069 6D20 4279 7465 2020 der 05 im Byte
0003A0: 3136 682C 2068 616E 6465 6C74 2065 7320 16h, handelt es
0003B0: 7369 6368 206C 6564 6967 6C69 6368 2020 sich lediglich
0003C0: 756D 2065 696E 656E 204C 7D63 6B65 6E2D um einen Lücken-
0003D0: 667D 6C6C 6572 2C20 756D 2066 7D6E 6620 füller, um fünf
0003E0: 5365 6B74 6F72 656E 2069 6D20 4869 6E2D Sektoren im Hin-
0003F0: 626C 6963 6B20 6175 6620 7370 7B2D 2020 blick auf spä-
```

SYS15/SYS, rel. Skt. 04 (neu)

```
000400: 2846 6F72 7473 2E20 6175 7320 6465 6D20 (Forts. aus dem
000410: 766F 7269 6765 6E20 05E2 5365 6B74 2E29 vorigen ..Sekt.)
000420: 2020 2020 2020 2020 2020 2020 2020 2020
000430: 7465 7265 2056 6572 7765 6E64 756E 6720 tere Verwendung
000440: 7A75 2062 6573 6574 7A65 6E2E 2020 2020 zu besetzen.
000450: 4869 6572 2068 6162 656E 206E 6F63 6820 Hier haben noch
000460: 7265 6368 7420 756D 6661 6E67 7265 692D recht umfangrei-
000470: 6368 6520 5072 6F67 7261 6D6D 6520 2020 che Programme
000480: 506C 6174 7A2E 2020 2020 2020 2020 2020 Platz.
000490: 2020 2020 2020 2020 2020 2020 2020 2020
0004A0: 2020 2020 2020 2020 2020 2020 2020 2020
0004B0: 2020 2020 2020 2020 2020 2020 2020 2020
0004C0: 2020 2020 2020 2020 2020 2020 2020 2020
0004D0: 2020 2020 2020 2020 2020 2020 2020 2020
0004E0: 2020 2020 2020 2020 2020 2020 2020 2020
0004F0: 2020 2020 2020 2020 2020 2020 0202 0052             ...R
```

LPRINT CHR$(irgendwas)

Der Druckertreiber im Microsoft-ROM ist auf der nächsten Seite aufgelistet und kommentiert. Wie man sieht, bezieht sich der größte Aufwand darauf, im Drucker-DCB den Zeilenzähler auf dem laufenden zu halten und bei senkrechter Tabulation lauter Zeilenvorschübe auszugeben. Das sind Features, die modernere Drucker beherrschen. Doppelt genäht hält besser, also sei's drum. Aber da ist eine andere Eigenschaft, die sich durchaus störend auswirkt. Welchen Nutzen sie hat, ist mir nicht klar:

Bekanntlich kann ein NUL-Code (ASCII 0) nicht mit LPRINT ausgedruckt werden. Man muß sich mit OUT 253,0 behelfen (beim TRS-80: POKE 14312,0). Bei gewöhnlichen Texten, Listings usw. spielt das keine Rolle, denn NUL kommt da nicht vor. Will man an seinen intelligenten Drucker jedoch die Null ausgeben, weil sie etwa Bestandteil eines Steuercodes ist, ist es immer wieder ärgerlich, den zusammenhängenden LPRINT-Befehl durch ein OUT-Statement unterbrechen zu müssen.

Eine entsprechende Änderung des Druckertreibers ist sehr einfach. Die Treiberadresse steht im Drucker-DCB an der Stelle 4026/4027h (16421/16422d). Bei einem Druckerbefehl wird dort die Adresse ausgelesen und die Routine angesprungen, die an dieser Adresse steht. Das ist normalerweise 05D8h im ROM. Wir können dort aber eine beliebige andere Adresse einschreiben. An dieser neuen Adresse (im abgedruckten Beispiel 7000h) wird zunächst der Akku wieder mit dem Zeichen geladen, das im ROM zuvor in das Register C übernommen wurde. Der nächste Schritt besteht lediglich darin, beliebige Teile des Originaltreibers einfach zu überspringen.

Wenn wir nur den Ausdruck von NUL ermöglichen wollen, springen wir an die Stelle NOZERO (Listing des Originaltreibers) zurück. Der Zeilenzähler bleibt dann in Betrieb. Wenn die Prüfung des Zeichens auf vertical tab nicht erwünscht ist (weil es nur Zeit kostet, denn der Drucker kann das auch), kann man nach NOVT springen. Soll auch nicht auf form feed getestet werden (dto.), geht es bei NOFF weiter. Diese Version zeigt das Listing unten.

Beim Austesten hat sich herausgestellt, daß LPRINT und LLIST durch diese Manipulation in keiner Weise beeinträchtigt werden. Die JKL-Option (DOS) bleibt ebenfalls unverändert intakt. Wozu also überhaupt dieser Aufwand im ROM? Ältere Drucker mögen einen Zeilenzähler im Speicher des Computers nötig gehabt haben. Es ist auch durchaus denkbar, daß bestimmte Textverarbeitungsprogramme ihn benutzen. Das NUL-Verbot mag auch einmal sinnvoll gewesen sein. Wer aber im Normalbetrieb einfach nur LLISTen und LPRINTen möchte, ist mit der vorgestellten Treiber-Umleitung gut bedient:

G-DOS 2.1.6

```
7000            00100         ORG    7000H           ;oder wo auch immer
7000 210770     00110 START   LD     HL,DRIVER       ;neuer Anfang d. Treibers
7003 222640     00120         LD     (4026H),HL      ;in den DCB laden
7006 C9         00130         RET                    ;-> DOS oder BASIC
                00140
7007 79         00150 DRIVER  LD     A,C             ;Zeichen in Akku laden
7008 C3B405     00160         JP     05B4H           ;ausdrucken "as is"
                00170
7000            00180         END    START           ;Einsprung dort
```

Arnulf Sopp, Tel. 0451-791926

```
                 00100 ;***   Der Druckertreiber im ROM    ***
                 00110
                 00120 ;CALL 003BH druckt immer:
003B             00130          ORG     003BH
003B 112540      00140          LD      DE,4025H         ;Adr. des Printer-DCB
003E 18DB        00150          JR      M001B
                 00160
                 00170 ;CALL 001BH druckt, wenn DE mit DCB geladen:
001B             00180          ORG     001BH
001B C5          00190 M001B:   PUSH    BC               ;retten, wird verändert
001C 0602        00200          LD      B,02H            ;Bit 1 des Device-Typs
001E 1826        00210          JR      M0046
                 00220
                 00230 ;CALL 0046H druckt, wenn DE und B bereits geladen:
0046             00240          ORG     0046H
0046 C3C203      00250 M0046:   JP      M03C2            ;nur Hühnerleiterfunktion
                 00260
                 00270 ;dasselbe, denn 0049H enthält nur den Sprungbefehl:
03C2             00280          ORG     03C2H
03C2 E5          00290 M03C2:   PUSH    HL               ;Register retten
03C3 DDE5        00300          PUSH    IX
03C5 D5          00310          PUSH    DE               ;DCB-Adresse
03C6 DDE1        00320          POP     IX               ;nach IX laden
03C8 D5          00330          PUSH    DE               ;Adr. retten (wozu?)
03C9 21DD03      00340          LD      HL,M03DD         ;RET-Adresse
03CC E5          00350          PUSH    HL               ;auf den Stack
03CD 4F          00360          LD      C,A              ;zu druckendes Zeichen
03CE 1A          00370          LD      A,(DE)           ;Devic-Typ
03CF A0          00380          AND     B                ;Bit 1 maskieren
03D0 B8          00390          CP      B                ;Bit vorhanden?
03D1 C23340      00400          JP      NZ,4033H         ;DOS-Exit, falls nein
03D4 FE02        00410          CP      02H              ;Flags setzen (wozu?)
03D6 DD6E01      00420          LD      L,(IX+01H)       ;HL mit der Treiber-
03D9 DD6602      00430          LD      H,(IX+02H)       ;adresse laden
03DC E9          00440          JP      (HL)             ;den Treiber anspringen
03DD D1          00450 M03DD:   POP     DE               ;nach RET: Reg. restaur.
03DE DDE1        00460          POP     IX
03E0 E1          00470          POP     HL
03E1 C1          00480          POP     BC
03E2 C9          00490          RET                      ;Druckvorgang beendet
                 00500
                 00510 ;Hier wird gedruckt und der Zeilenzähler verwaltet:
058D             00520          ORG     058DH
058D 79          00530          LD      A,C              ;zu druckendes Zeichen
058E B7          00540          OR      A                ;= NUL (0) ?
058F 2840        00550          JR      Z,M05D1          ;nichts tun, falls ja
0591 FE0B        00560 NOZERO   CP      0BH              ;vertical tab?
0593 280A        00570          JR      Z,M059F          ;falls ja
0595 FE0C        00580 NOVT     CP      0CH              ;form feed?
0597 201B        00590          JR      NZ,NOFF          ;falls nein
0599 AF          00600          XOR     A                ;form feed: A <- 0
059A DDB603      00610          OR      (IX+03H)         ;A <- Zeilen/Seite
059D 2815        00620          JR      Z,NOFF           ;falls 0 Z/S
059F DD7E03      00630 M059F:   LD      A,(IX+03H)       ;Zeilenzahl wieder laden
05A2 DD9604      00640          SUB     (IX+04H)         ;abzgl. bisher gedr. Z.
05A5 47          00650          LD      B,A              ;B <- Zähler f. restl. Z.
05A6 CDD105      00660 M05A6:   CALL    M05D1            ;Drucker bereit?
05A9 20FB        00670          JR      NZ,M05A6         ;abwarten, bis bereit
05AB 3E0A        00680          LD      A,0AH            ;line feed
05AD 00          00690          NOP                      ;TRS-80: LD (37E8H),A
05AE D3FD        00700          OUT     (0FDH),A         ;line feed drucken
05B0 10F4        00710          DJNZ    M05A6            ;bis Seite voll
05B2 1818        00720          JR      M05CC            ;dort weiter
05B4 F5          00730 NOFF     PUSH    AF               ;retten
05B5 CDD105      00740 M05B5:   CALL    M05D1            ;Drucker bereit?
05B8 20FB        00750          JR      NZ,M05B5         ;falls nein
05BA F1          00760          POP     AF               ;zu druckendes Zeichen
05BB 00          00770          NOP                      ;TRS-80: s. o.
05BC D3FD        00780          OUT     (0FDH),A         ;Zeichen drucken
05BE FE0D        00790          CP      0DH              ;war es carriage return?
05C0 C0          00800          RET     NZ               ;zurück, falls nein
05C1 DD3404      00810          INC     (IX+04H)         ;ja, Zeilenzähler erhöhen
05C4 DD7E04      00820          LD      A,(IX+04H)       ;wieviele Zeilen bisher?
05C7 DDBE03      00830          CP      (IX+03H)         ;Seite schon voll?
05CA 79          00840          LD      A,C              ;Zeichen zurückholen
05CB C0          00850          RET     NZ               ;falls S. noch nicht voll
05CC DD360400    00860 M05CC:   LD      (IX+04H),00H     ;voll: Zähl. auf 0 setzen
05D0 C9          00870          RET                      ;und zurück
                 00880
                 00890 ;Druckerbereitschaft feststellen:
05D1 00          00900 M05D1:   NOP                      ;TRS-80: LD A,(37E8H)
05D2 DBFD        00910          IN      A,(0FDH)         ;Druckerstatus laden
05D4 E6F0        00920          AND     0F0H             ;linkes Nibble maskieren
05D6 FE30        00930          CP      30H              ;bereit, wenn Bits 4&5=1
05D8 C9          00940          RET                      ;mit Statusflags zurück
0000             00950          END
00000 mal gepennt
32487 Zeichen verfügbar
```

```
00010           TITL    '*** U m l a u t ***
00020           SBTL    'Umlautreiber für MULTIDOS, (c) 1984 by Ralf Folkerts
00030           COMM    '*****************  Umlaut  **** (c) 1984 by: *
00040           COMM    '* Ralf Folkerts*****************
00050           ;Umlauttreiber fuer MULTIDOS und NEWDOS
00060           ;Copyright (c) 06/84 by
00070           ;Ralf Folkerts
00080           ;Nutzhorner Strasse 9
00090           ;2875  Bookholzberg
00100           ;********
00110           ;DIE TITL UND SBTL BEI ASSMEBLERN <>ZEUS
00120           ;BITTE WEGLASSEN. EBENSO COMM.
00130           ;ALLE LABEL UND HEXZAHLEN GROSS.
00140           ;FALLS DER TREIBER NICHT UNTER DOS LAUFEN
00150           ;SOLL, SIND DIE ZEILEN
00160           ;330 UND 240 WEGZULASSEN.
00170           ;FALLS UNTER NEWDOS LAUFT, KOENNEN DIE
00180           ;ZEILEN 310 UND 320 WEGFALLEN.
00190           ;**********
00200           ;
00210           ;NACH DER AKTIVIERUNG VERSCHIEBT DER TREIBER
00220           ;SICH SELBSTSTAENDIG AN TOPMEM. DANN GIBT ER
00230           ;EINE FERTIG - MELDUNG AUS. NUN KANN DIE UM-
00240           ;LAUTBILDUNG DURCH DRUECKEN DER <CLEAR> TASTE
00250           ;EINGELEITET WERDEN. WENN MAN NACH DER <CLEAR>
00260           ;TASTE EINE ANDERE TASTE DRUECKT, ERGEBEN SICH
00270           ;FOLGENDE FUNKTIONEN:
00280           ;<A> : AE;  <O> : OE;  <U> : UE;  <S> : SZ
00290           ;MIT <SHIFT> ENTSPRECHEND.
00300           ;<CLEAR> : EINGETLICHE <CLEAR> FUNKTION
00310           ;JEDE ANDERE TASTE: WIE NORMAL
00320           ;
00330           ;**********.
00340           ORG     5200h               ;Overlay
00350 start     LD      HL,end              ;Letzte Zeile
00360           LD      BC,byte-begin       ;Anzahl der Bytes
00370           LD      DE,(topmem)         ;Letzte Adresse
00380           LDDR                        ;Verschibe Prgm.
00390           EX      DE,HL               ;Hole neue Adresse
00400           LD      (topmem),HL         ;Schreibe neuen Topmem
00410           LD      (basic),HL          ;Letzte Adr. f. BASIC
00420           INC     HL                  ;Counter + 1
00430           PUSH    HL                  ;Sichere Adresse
00440           LD      HL,(kidcb)          ;Hole alte Treiber Adr.
00450           POP     IX                  ;Hole HL in IX
00460           LD      (IX+2),H            ;Speichere LSB
00470           LD      (IX+1),L            ;Speichere MSB
00480           LD      (kidcb),IX          ;Neuen DCB Start
00490           LD      (IX+10),H
00500           LD      (IX+9),L
00510           LD      A,20h               ;Code fuer LC ein
00520           LD      (kidcb+2),A         ;Schalte LC ein
00530           LD      HL,bereit           ;Hole Text, bereit
00540           CALL    vod                 ;Ausgabe auf Schirm
00550           JP      dos                 ;Zurueck zum DOS
00560 ;
00570 ;         Bis hier Relocater und Init. Ab jetzt
00580 ;         folgt eigentlicher Treiber
00590 ;
00600 begin     CALL    0                   ;Hier Dummy f. DCB Adr.
00610           OR      A                   ;FLAG's setzen
00620           RET     Z                   ;Wenn '0', zurueck
```

```
        CP      1fh             ;Ist es <CLEAR> ?
        RET     NZ              ;Wenn nein, zurueck
recall  CALL    0               ;Treiber call
        OR      A               ;FLAG's
        JR      Z,recall        ;Bis Taste
        CP      'a'             ;Ist es a
        LD      C,7bh           ;Code f. ae
        JR      Z,ende
        CP      'o'             ;Ist es o
        LD      C,7ch           ;Code f. oe
        JR      Z,ende
        CP      'u'             ;Ist es u
        LD      C,7dh           ;Code f. ue
        JR      Z,ende
        CP      's'             ;Ist es s
        LD      C,7eh           ;Code f. sz
        JR      Z,ende
        CP      'A'             ;Ist es 'A' ?
        LD      C,5bh           ;Code f. Ae
        JR      Z,ende
        CP      'O'             ;Ist es 'O'
        LD      C,5ch           ;Code f. Oe
        JR      Z,ende
        CP      'U'             ;Ist es 'U'
        LD      C,5dh           ;Code f. Ue
        JR      Z,ende
        CP      1fh             ;Ist es <CLEAR>
        LD      C,1fh           ;Code f. <CLEAR>
        JR      Z,ende
        LD      C,A             ;Code der Taste
ende    LD      A,C             ;Code in AKKU
end     RET                     ;Zurueck
byte    DB      0               ;Dummy f. relocat.
;
;       Ab hier Definitionen
;
topmem  EQU     4049h
basic   EQU     40b1h
kbdcb   EQU     4016h
vod     EQU     4467h
das     EQU     402dh
bereit  DM      0ah,'Der Umlauttreiber, (c) by Ralf Folkerts ist bereit'
,0ah,0dh
        END     start
```

ΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨΨ

Nochmal zum Thema 'AkustikKoppler':
Heinrich Thönnißen braucht bis

<u>spätestens 09.01.1985</u>

die Bestellungen. Es bleibt dann noch genügend Zeit, die genaue Anzahl im Januar-Info zu veröffentlichen. Jeder kann sich dann den genauen Preis ausrechnen.

<u>Stichtag: ----->> 09.01.1985 !!!!</u>

Und es geht doch: LPRINT CHR$(10)

Nachdem O. Stark und ich mit blindem Gottvertrauen behauptet hatten, mit LPRINT sei 0Ah auf den Drucker zu kriegen (womit wir uns bei den Selbstdenkern unter Euch gründlich blamiert haben dürften), und nachdem ich in einem weiteren Beitrag (der frühestens im November-Info steht) Euch zur Suche nach dem casus knaxus animierte, habe ich ihn num doch selber gefunden:

Der Druckertreiber ist wirklich vollkommen unschuldig. Er hat zwar seine Tücken, weil er bei ASCII 00 nur den Druckerstatus abfragt und aus einer vertikalen Tabulation und einem Blattvorschub lauter einzelne Zeilenvorschübe macht, aber dagegen ist das Kraut aus meinem Beitrag "LPRINT CHR$(irgendwas)" gewachsen. ASCII 0A kommt dort aber gar nicht erst an!

Eine BASIC-Zeile wie

```
10 LPRINT CHR$(10)
```

ist für den Interpreter eine relativ harte Nuß. Ohne mir jetzt die Mühe machen zu wollen, das zu überprüfen, schätze ich, daß zur Bearbeitung mindestens an die 30 Unterprogramme durchlaufen werden müssen. Dazu gehört auch das UP an 039C, das an der Ausgabe eines Zeichens auf den Drucker beteiligt ist. In dessen Verlauf findet sich auch diese Befehlsfolge:

```
03A6  FE0A   CP   0AH        ;auf ASCII 0A prüfen
03A8  2003   JR   NZ,03ADH   ;dort weiter, falls anderer Wert
03AA  3E0D   LD   A,0DH      ;0A durch 0D ersetzen!
```

Bei der Suche nach dieser Gemeinheit erwies sich mal wieder der EG 64 MBA als lohnende Investition. Bei Sprungbefehlen das Ziel zu finden, ist simpel. Das Gegenteil erweist sich aber im ROM, wo man normalerweise nicht mit einem Monitor einen Breakpoint setzen oder auf andere Weise ein Kuckucksei legen kann, als Fleißarbeit. Wo der Interpreter aber soft vorliegt, läßt sich locker ein F7 (RST 30h, um DEBUG aufzurufen) einschreiben. Das DEBUG-Display zeigt nun u. a. den Stackpointer SP an. Er verweist auf die RET-Adresse und damit auf den CALLer. Dieser wird nun seinerseits mit einem F7 verwanzt usw., bis der Übeltäter gefunden ist. Das dauerte in diesem Falle keine drei Minuten.

Ohne Disco geht es nicht minder einfach. Statt RST 30h wird dann eben JP MONADR eingeschrieben, ein Sprungbefehl zu einem zuvor geladenen Monitor. Diese Adresse muß allerdings zuvor ermittelt werden: Man setzt an einer beliebigen Stelle im RAM einen Breakpoint. Mit Reset verläßt man nun den Monitor und schaut sich (mit PEEK oder wie auch immer) die drei Bytes ab Breakpoint an. Dort steht bei allen mir bekannten Monitoren ein CALL an die Adresse, wo Breakpoints bearbeitet werden. Das erste Byte dürfte daher CD lauten. Die beiden folgenden Bytes bilden die Adresse, die bei unserem künstlichen Breakpoint mit JP angesprungen wird. In der Anzeige erscheint dann SP mit seinem verräterischen Inhalt.

Natürlich soll dieser Fund nicht folgenlos bleiben. Wer seinen Microsoft-Interpreter durch drei EPROMs ersetzt, kann an den Stellen 03AA und 03AB zwei NOPs einschreiben und ist damit diese Sorge los. Bequemer geht es mit dem MBA (bitte nicht hauen, er wird in diesem Artikel nicht wieder erwähnt), denn der Interpreter liegt auf Wunsch ohnehin soft vor.

Arnulf Sopp

FDE - Sektoren

Der Rest des Direktorys ist belegt mit den FDE-Sektoren. Jeder Sektor enthält maximal 8 Einträge zu je 32 Bytes. Im Normalfall sind jedem Programm 32 Bytes zugeordnet.
Die verschiedenen Bits des 1. Bytes enthalten Informationen wie Ersteintrag oder Folgeeintrag, FDE frei oder belegt, usw.
Wird ein Eintrag durck "KILL" gelöscht, so wird nur das 4. Bit des 1. Bytes auf 0 gesetzt. So könnte also ein versehentlich gelöschter Eintrag mit dem COLZAP gerettet werden.
Byte 6 - 13 enthält den linksbündigen Namen, falls notwendig aufgefüllt mit Leerzeichen.
Byte 14 - 16 gibt den linksbündigen Filetyp (CMD, BAS, ..) an.
Die letzten beiden Bytes vor der "FF-Reihe" dienen zur Berechnung der relativen Sektornummer, an der der Eintrag beginnt.
Das erste dieser beiden Bytes gibt die dezimale Nummer des LUMPS an, die mal 15 gerechnet werden muß.
Die linken 4 Bits des nächsten Bytes können nur den Wert 0/1 2/3 oder 4/5 annehmen. Dieser Wert steht für den 0-ten, 5-ten oder den 10-ten Sektor des LUMPS, der nun zu dem vorigen Ergebnis zu addieren ist.

Beispiel: 2. Eintrag des 1. FDE-Sektors (SINGLEST/BAS)

Die beiden Bytes sind 2C 40
2C hex ist 44 dez. 44 * 15 = 660
4 bedeutet den 10. Sektor → 660 + 10 = 670
Das Programm SINGLEST/BAS ist somit ab dem relativen Sektor Nummer 670 zu finden.

```
0000 5E083300004E435731393833204A484C
0010 607F1FB205000000FFFFFFFFFFFFFFFF
0020 102000DE0053494E474C455354424153    Name und Filetyp
0030 000000000003002C40FFFFFFFFFFFFFF
0040 00200000004550524F4D332020202020
0050 0000000000002D002F21FFFFFFFFFFFF
0060 00000000000000000000000000000000
0070 00000000000000000000000000000000
0080 00000000000000000000000000000000    8 Einträge zu je
0090 00000000000000000000000000000000    32 Bytes
00A0 00000000000000000000000000000000
00B0 00000000000000000000000000000000
00C0 00000000000000000000000000000000
00D0 00000000000000000000000000000000
00E0 00000000000000000000000000000000
00F0 00000000000000000000000000000000
Drive 0, DRS 0362, TRK 21, SEC 02, PROT
0000 ^.8..NCW1983 JHL`. ....
0020 . . ..SINGLESTBAS.......,$
0040 . ...EPROM3        .......-./!
0060 ................................
0080 ................................
00A0 ................................
00C0 ................................
00E0 ................................
```

Natürlich sind die von mir gemachten Angaben nicht vollständig. Für ein anfängliches Verstehen und Arbeiten dürfte das aber mehr als genug sein. In Bälde gibt es sicher auch für das Colourgenie-Floppy entsprechende Literatur, so daß ein Weiterstudium möglich ist.

Zum Schluß möchte ich noch auf eine Möglichkeit hinweisen, wie Disketten eines anderen Systemes lesbar werden, (z.B. NEWDOS 80).
Da das Colour-DOS keine direkte Beeinflussung der sogenannten PDRIVE-Daten hat, sind normalerweise Fremddisketten nicht lesbar. In den EPROMs des Betriebssystemes sind diese Daten allerdings fest eingebrannt. Nach einem Start werden sie dann an den Arbeitsbereich ab der HEX-Adresse 5A71 (für das 1. Laufwerk) geladen. Um nun eine Anpassung durchführen zu können, müssen die neuen Daten dahin gepoked werden.

Aktuelle Laufwerksinformationen des 1. LW

5A71	18	Track/Dir	neu 11
5A72	30		
5A73	53		
5A74	28	40 Spuren/Disk	
5A75	12	Sektoren/Track	
5A76	03		neu 02
5A77	00		
5A78	03		
5A79	05	Sektoren/Granule	
5A7A	03	Granules/Dir	neu 02

Mit nur drei pokes ist es nun möglich, Disketten des NEWDOS 80 zu lesen (POKE&H5A71,11 - POKE&H5A76,2 - POKE&H5A7A,2). Prinzipiell kann somit das Laufwerk auf jedes andere DISK-System angepaßt werden.
Die Veränderungen, um für dieses System Disketten schreiben zu können, habe ich leider noch nicht gefunden.
Für mich hat allerdings das Lesen von Disketten anderer Systeme bisher immer genügt. Denn zum Weiterbearbeiten sind die Programme dann ja auf die eigene Diskette zu saven.

Und nun umgekehrt! Auch das NEWDOS 80 kann Colourgenie-Disketten lesen, wenn die PDRIVE-Einstellung verändert wird. Der Arbeitsgang läuft folgendermaßen ab:

```
Eingabe:  PDRIVE,0,1

          Laufwerk     |Laufwerk
          mit System-  |mit Colour-
          diskette     |diskette

          PDRIVE,0,1,TD=E,TC=40,SPT=18,
                     TRS=0,GPL=3,DDSL=24,
                     DDGA=3
          PDRIVE,0,A
```

Wenn nur ein Laufwerk vorhanden ist, muß natürlich die Systemdiskette gegen die Colourgeniediskette nach Anweisung ausgetauscht werden. Diese Angaben hat mein Freund Christian Röhrs auf seinem TRS 80 herausgefunden.
Und nun wünsche ich allen, die das ausprobieren wollen viel Erfolg beim Diskettentausch.

Dieter Gerblinger
Alpenkorpsstrasse 23
8102 Mittenwald
Tel.: 08823 - 2017

Wie ist Genietext aufgebaut?

Antwort auf die Frage von K.J.Mühlenbein warum Genietext die rechte Randbegrenzung nicht mehr mitmacht, wenn ein sog. "umgewandeltes Zeichen" (z.B.: #a) verwendet wird.

Genietext folgt einem bestimmten mathematischen Algorithmus, um den Randausgleich zu berechnen. Es zählt wieviel Wörter in eine Zeile passen und dann fügt es "Blanks" zwischen den Wörtern ein, bis die Zeile die endgültige Länge erreicht hat. Erst wenn Genietext den Randausgleich einer Zeile errechnet hat, geht er zum Drucken über.

Jetzt kommt aber der große Fehler von Genietext! Beim Berechnen des Randausgleichs, werden die Sonderzeichen ($#) mitgezählt. Beim Drucken werden diese Zeichen aber nicht ausgedruckt. Damit schiebt sich jede Zeile jeweils um ein Zeichen nach links. Der Randausgleich funktioniert nicht mehr. Um es zu verändern, müßte man mit Maschinensprache an das Programm ran. Meine Version von Genietext wandelt die Sonderzeichen beim Drucken in Blanks um. Das ist auch ein lästiger Effekt.

Die Tatsache, daß Genietext selbstbootend ist, hat bestimmt viele von uns abgeschrekt, sich an das Programm ranzumachen. Jetzt kommt aber der Clou!!! Genietext bassiert auf NewDos80. Bootet mal eine NewDos80 Diskette. Gebt dann <DIR $U> ein und tut dann die Genietext Diskette ins Laufwerk ein. Ihr seht jetzt das Directory von Genietext. Leider erscheint nicht alles auf das Directory, was in Wirklichkeit drauf ist. (Kleiner Trick von Zender!) Es handelt sich um die Programme BASIC/CMD, SYS8/CMD und SYS9/CMD.

Wer die Inhalte der anderen Sys-Files genau wissen will, dem rate ich vorher Ramsys zu laden, da Genietext alle überflüßige Files gekillt hat. Ich habe es getan, und bin zu folgende überraschende Ergebnisse gekommen:

1) BASIC/CMD ist nicht das Basic, sondern das eigentliche Genietextprogramm. Man kann es aber leider nur auf der Genietextdiskette zum Laufen bringen.

2) SYS8/CMD ist das Basic. Wozu braucht aber Genietext Basic? Das wird im nächsten Punkt erklärt.

3) SYS5/SYS ist ein Basic-Programm!!! Es wird von Genietext aufgerufen, wenn man das Kommando 'U' eingibt. Die Unterkommandos werden also vom Basic aus bedient. (Leicht zu kopieren und zu verändern!). Wie werden aber die Änderungen durchgeführt? Sie werden mit Superzap gemacht.

4) SYS9/CMD ist Superzap, und wird vom Basic aus aufgerufen.

Die Änderungen sind leicht durchzuführen. Wenn jemand weiterkommt, bitte im Info bescheidgeben.

(© Alfonso Sanz)

Die Systemoptik ausgetrickst!

System- oder programminterne (was wohl im Prinzip dasselbe ist) Bedingungen erschweren von Fall zu Fall den Versuch, auf dem Bildschirm oder der Hardcopy ein optisch ansprechendes Erscheinungsbild zu erzeugen. So ist beispielsweise ein Diskname (ebenso das Datum) nicht mit dazwischenliegenden Blanks (Leerstellen) einzugeben. Und EDTASM, der wohl weitestverbreitete Editor-Assembler, läßt nicht die Eingabe von Steuercodes für den Drucker zu. Und im File SYS28/SYS, in das die Zeichen für den FORM-Befehl (G-DOS) eingegeben werden, sinddie Codes 03 und 0Dh nicht zulässig, denn sie werden als Endzeichen interpretiert.

Häufig hilft die CTRL-Taste, mit der man Codes unter ASCII 32 (Blank) erzeugen kann. Wird sie gleichzeitig mit einer Zeichentaste gedrückt, steht im I/O-Puffer der um (zumeist) 64 verminderte Wert. Aber genau gleichzeitig schafft man es nie, so daß in der Regel zunächst ein LF (line feed = Zeilenvorschub) ausgegeben wird.

Wenn nichts mehr hilft, kann man zumeist problemlos mitten in die fertig vorliegende Datei (Programm, Daten, Text, Quelltext usw.) hineinschreiben. Es versteht sich, daß für die später beabsichtigten Steuerzeichen für den Bildschirm oder den Drucker Platz reserviert werden muß. Man kann bei der Eingabe an der betreffenden Stelle ein beliebiges, möglichst leicht wiederzufindendes Zeichen eingeben. Anschließend wird mit einem Monitorprogramm oder durch schlichtes PEEKen diese Stelle aufgesucht und der endgültige Code eingeschrieben.

Die Lösung ist für das o. g. FORM-Problem am einfachsten, deshalb dies zuerst: Für die meisten Drucker ist das höchstwertige, das Bit 7 des übertragenen Codes erst wichtig, wenn der Rest 20h (32d) überschreitet, wenn also der Gesamtwert mindestens ASCII A0h (160d) beträgt. So kann 0Dh leicht als 8Dh und eine 3 als 83h eingetippt werden, ohne daß für den Drucker ein Unterschied bestünde.

Etwas umständlicher, im Prinzip aber ebenso einfach geht es, in einem BASIC- oder EDTASM-Quelltext die Anweisungen zur Druckerformatierung unterzubringen. Im Folgenden sollen für BASIC und Assembler je ein Fallbeispiel gezeigt werden:

1. Problem: REM-Statements sollen hervorgehoben werden.

Das Befehlswort REM kann auch durch das Hochkomma <'> ersetzt werden, was hier geschieht, damit bei LLIST nur ein Zeichen übermittelt wird. Und dies soll zunächst zum Verschwinden gebracht werden. Hierzu dient der DEL-Code (delete, ASCII 7Fh = 127d) für den Drucker. Er bewirkt, daß das vorangegangene Zeichen getilgt wird und nicht zum Ausdruck kommt. Sodann soll die REM-Zeile unterstrichen erscheinen. Dies geschieht bei meinem Drucker mit der Zeichenfolge <ESC> <-> ASCII 1. Insgesamt müssen also vier Dummy-Zeichen zwischen dem <'> und dem REM-Text erscheinen.

Um die Unterstreichung wieder zu löschen, wird normalerweise <ESC> <-> NUL (ASCII 0) eingegeben. Da jedoch die Null als Zeilenende interpretiert wird, was hier fatale Folgen hätte, kann man auf die Neuinitialisierung (bei meinem Gerät <ESC> <§>) ausweichen. Auch hierfür sind nach dem REM-Text zwei Dummy-Zeichen vorzusehen.

Die Abb. 1 und 2 zeigen denselben BASIC-Text, mit Dummy-Zeichen und nach dem Ersetzen durch die beabsichtigten Codes. Ich habe es mir leichtgemacht, indem ich DEL mit D andeutete, ESC mit $ usw. So konnte ich die betreffenden Speicherstellen leicht wiederfinden. Wie, das kommt später.

Selbstredend muß man darauf achten, daß solche Manipulationen nur da erlaubt sind, wo sie keinen Syntax-Error verursachen können, also nur nach REM oder zwischen <">. Daß man mit der gleichen Methode auch die Zeilennummern in eine phantasievolle Reihenfolge verbiegen und sonst noch allerhand anstellen kann, sei hier nur erwähnt. Das Bildschirmlisting mag kraus aussehen - das darf hier nicht stören, wenn es darum geht, "für die Akten" eine ansprechende Hardcopy zu erzeugen.

2. Problem: Gutaussehender Header einer EDTASM-Source

Es ist gern geübter Brauch, den Kopf eines Assembler-Quelltextes mit <*> vom eigentlichen Programmtext optisch zu trennen. Weit mehr ist möglich, und zwar mit der gleichen Methode, die oben für BASIC-Texte beschrieben ist. Die Abb. 3 und 4 zeigen das "Vorher-Nachher". Für dieses Beispiel habe ich den Header einer Maschinenroutine gewählt, die ich vor längerer Zeit entwickelte, um Maschinenprogramme mit dem SYSTEM-Befehl auch mit sedezimaler Adresse starten zu können (natürlich nur für Level 2 interessant, denn Disk-BASIC kann das auch).

Hier ist zu beachten, daß EDTASM beim H- bzw. A/LP-Befehl das Bit 7 des übertragenen Codes mißachtet. Es ist daher leider nicht möglich, die Graphikzeichen >80h bzw. >A0h auf den Drucker zu bringen, ohne zuvor EDTASM umzukrempeln. Aber was anstandslos funktioniert, ist genug:

In unserem Beispiel werden die Semikola <;> gelöscht, die dieselbe Funktion wie REM in BASIC haben, Breitschrift wird eingeschaltet und der Programmname wird unterstrichen. Hier habe ich der Demonstration halber ein wenig zuviel des Guten getan, denn man stelle sich vor, jemand tippt eine Kommentarzeile ohne Semikolon ab!

Wie schon beim BASIC-Beispiel hat auch hier die Neuinitialisierung, um die Unterstreichung zu löschen, die Folge eines LF. Je nach Druckertyp kann mit der hier beschriebenen Methode wohl auch dieser Schönheitsfehler beseitigt werden.

Bei Manipulationen dieser Art gilt es natürlich zunächst, den Text aufzufinden, bevor man ihn verändern kann. Auf BASIC-Programmtexte weist ein Zeiger in 40A4/40A5h (16548/16549d). Man findet die Stelle mit der Befehlsfolge

```
PRINT PEEK(16548)+256*PEEK(16549)
```

Mit einer FOR-NEXT-Schleife, einem Monitorprogramm oder wie auch immer kann man nun das Programm nach den vorbereiteten Dummy-Zeichen durchsuchen und die so aufgespürten Speicherstellen neu beschreiben. Die dergestalt veränderten Programmzeilen lassen sich ohne weiteres edieren (EDIT), solange die Codes <80h (128d) sind.

Bei EDTASM wird es insofern etwas schwieriger, als derartige Utilities gerne für den eigenen Bedarf verändert werden. Bei der mir vorliegenden Disk-Version beginnt der Quelltext bei 7700h (30464d). In Level-2-Versionen läßt er sich mit einem Monitor oder mit der BASIC-Befehlsfolge

```
FOR I%=17129 TO 65535: PRINT CHR$(PEEK(I%));: NEXT
```

auffinden. Irgendwann erscheint auf dem Bildschirm etwas, das man als den Anfang seiner EDTASM-Source wiedererkennt. Dann kann man mit <BREAK> und PRINT I% feststellen, wo gerade gesucht wurde. Mit einer kleineren FOR-NEXT-Schleife wird dann die genaue Stelle gefunden. Hierbei ist zu beachten, daß die Level-2-Version von EDTASM ziemlich tief residiert. Es ist daher riskant, programmgesteuert zu suchen. BASIC könnte EDTASM zuschaufeln. Direkte Befehle ohne Zeilennummern sind unbedingt vorzuziehen. Die Zählvariablen (zumal von Integertyp V%) für FOR-NEXT allein reichen im RAM noch nicht bis zu EDTASM hinauf. Die Variable sollte aber immer denselben Namen haben, damit nicht mehrere Variable zusammen so viel RAM besetzen, daß EDTASM die weiße Fahne schwenkt.

Das für diesen Artikel verwendete Textverarbeitungsprogramm TSCRIPS hat sich bisher leider meinen Versuchen widersetzt, dergleichen auch mit ihm zu veranstalten. Z. B. wollte ich meinen Briefkopf mit Graphikelementen verschönern, leider bisher ohne Erfolg. Wer TSCRIPS intimer kennt als ich und mir helfen möchte, es zu verändern, findet meine Adresse in Abb. 3/4. Vielen Dank!

```
10 'D$-1Variablenzuordnung:$§
20 A=5: B$="Testprogramm": C1=&H42E9
```

Abb. 1

```
10 Variablenzuordnung:

20 A=5: B$="Testprogramm": C1=&H42E9
```

Abb. 2

```
00100 :DB************************
00110
00120 :DB     $-1S  Y  S  H  E  X$S
00130
00140 :D Eingabe der Startadresse von System-Format-Pro=
00150      grammen wahlweise dezimal oder hexadezimal
00160
00170 Dezimale Eingabe wie gewohnt mit       "/ddddd",
00180 Hexeingabe stattdessen mit             ":hhhh".
00190
00200 :DB************************
00210 ;DC 1983 by A. Sopp, Wakenitzstr. 8, 2400 Lübeck 1
```

Abb. 3

```
00100 ************************
00110
00120         S   Y   S   H   E   X

00130
00140  Eingabe der Startadresse von System-Format-Pro=
00150     grammen wahlweise dezimal oder hexadezimal
00160
00170 Dezimale Eingabe wie gewohnt mit       "/ddddd",
00180 Hexeingabe stattdessen mit             ":hhhh".
00190
00200 ************************
00210 C 1983 by A. Sopp, Wakenitzstr. 8, 2400 Lübeck 1
```

Abb. 4

Arnulf Sopp

Convert

Das Programm 'CONVERT' dient der Umwandlung von SCRIPSIT/SUSCRIP Text Files in ASCII Files. Hierdurch kann man z.B. mit dem Textprogramm ein BASIC Programm schreiben, mit dem Programm convert in ASCII Format umwandeln und danach als BASIC Programm laden. Nach einer kleinen Modifikation (vertauschen zweier Zeilen) ist auch der umgekehrte Weg möglich. Es kann dann ein BASIC Programm 'REconvertIERT' und mit dem Textprogramm editiert werden. Das Programm ist in der vorliegenden Form für das DOS 'MULTIDOS' geschrieben, kann jedoch nach einer kleinen Änderung auch unter NEWDOS 'gefahren' werden. Auch eine Anpassung an andere Textprogramme ist möglich.

Funktion:
In den Zeilen 120 - 440 werden die beiden Filespecs 'geholt'. Hierzu werden die Filenamen, die in Klammern (mit beliebig vielen Leerzeichen von Befehl 'CONV' getrennt, jedoch OHNE Leerzeichen in der Klammer) stehen zuerst in die Zwischenspeicher 'QFILE' und 'ZFILE' geschoben, und jeweils mit einem 0D abgeschlossen.

In den Zeilen 460 - 510 werden die File DCB's gebildet. Dies geschieht dadurch, daß die Filenamen in 'QFILE' und 'ZFILE' stehen, durch einen Aufruf der DOS Routine ab 441CH in die Speicher 'QCONV' und 'ZCONV' verschoben werden.

Dann werden in den Zeilen 540 - 680 die Files geöffnet. Hierbei ist zu beachten, daß bei NEWDOS in einer Zeile 555 'LD B,0' stehen muß, da Multidos die LRL aus der Directory entnimmt, während er bei NEWDOS angegeben werden muß.

In den Zeilen ab 700 werden dann die einzelnen Quellfile Sectoren in den Zwischenspeicher 'QDAT' gelesen, dann in den Speicher 'ZDAT' kopiert, wobei auch die Konvertierung geschieht. Der Speicher 'ZDAT' wird dann in den Zielfile Sector geschrieben.

Benutzung:
Quellfile (zu konvertierender SCRIPSIT File) = QFILE
Zielfile (ASCII File, Ergebnis der Konvertierung) = ZFILE

Aufruf mit: 'CONV (QFILE=ZFILE) <NEW LINE>
Zu beachten ist, daß in der Klammer kein Leerzeichen stehen darf, während zwischen Klammer und 'CONV' beliebig viele Blanks stehen dürfen. Wenn keine Klammer vorhanden ist, oder die Filespecs falsch in der Klammer stehen meldet sich daß System mit 'DOS ERROR = 2F UNKNOWN ERROR' (dies ist die Meldung 'BAD PARAMETER(S)' bei NEWDOS).

Um das Programm auch umgekehrt laufen zu lassen (ASCII in SCRIPSIT) sind die Zeilen 810 und 830 einfach zu vertauschen. Dies Programm kann man dann 'RECONV' nennen. Zu beachten ist natürlich, daß ein Basic Programm als ASCII File gespeichert werden muß.

Anpassung an andere Textprogramme:
Hierzu ist in der Zeile 810 das 8DH in das ASCII Zeichen zu verändern, welches das verwendete Textprogramm als 'CR' verwendet (bei TRSTEXT ist dies z.B....).

Da sich das Programm in den Reservierten DOS Bereich ab 5200H lädt, ist ein setzen von HIMEM oder TOPMEM nich nötig.

```
                00010 ;         CONV/OBJ
                00020 ;         KONVERTIERT SCRISIT TEST FILES IN ASCII FORMAT
                00030 ;         03/84 BY
                00040 ;         RALF FOLKERTS
                00050 ;         NUTZHORNER STRASSE 9
                00060 ;         2875  BOOKHOLZBERG
                00070 ;         *****
                00080 ;         START
                00090 ;         *****
5200            00100           ORG     5200H
                00110 ;         ***     5200 IST START VON DOS OVERLAY AREA
5200 211843     00120 START     LD      HL,4318H
                00130 ;         ***     4318H IST START DES 'BEFEHLSSPEICHERS'
5203 7E         00140 SEARCH    LD      A,(HL)          ;HL AUF BEFEHLSPEICHER
5204 FE0D       00150           CP      0DH             ;IST ES CR ?
5206 CA3255     00160           JP      Z,PAERR         ;WENN JA, PARAMETER ERROR
5209 FE28       00170           CP      '('             ;IST ES '(' ?
520B 23         00180           INC     HL              ;ERHOEHE ZAEHLER
520C C20352     00190           JP      NZ,SEARCH       ;WIEDERH. BIS '('
520F 11C252     00200           LD      DE,QFILE        ;ADDR. FUER QUELLE IN DE
5212 7E         00210 GETQ      LD      A,(HL)          ;AKKU AUS HL
5213 FE0D       00220           CP      0DH             ;IST ES CR ?
5215 CA3255     00230           JP      Z,PAERR         ;WENN CR, DANN PA. ERR.
5218 FE3D       00240           CP      '='             ;IST ES '=' ?
521A CA2352     00250           JP      Z,GETZ          ;WENN JA, DANN ZIELFILE
521D 12         00260           LD      (DE),A          ;ABSPEICHERN FUER QUELLE
521E 13         00270           INC     DE              ;POINTER <= POINTER + 1
521F 23         00280           INC     HL              ;---'--- <= ---'--- ' '
5220 C31252     00290           JP      GETQ            ;WIEDERHOLE BIS GES.QFILE
5223 3E0D       00300 GETZ      LD      A,0DH           ;CR IN AKKU
5225 12         00310           LD      (DE),A          ;CR AN FILESPEC
                00320 ;         ***     JETZT IST QUELLFILE FERTIG
5226 11DA52     00330           LD      DE,ZFILE        ;ADDR. FUER ZIEL IN DE
5229 23         00340 GETZ2     INC     HL              ;POINTER <= POINTER + 1
522A 7E         00350           LD      A,(HL)          ;HOLE NAECHSTES ZEICHEN
522B FE29       00360           CP      ')'             ;IST ES ')'
522D CA3A52     00370           JP      Z,REFILE        ;WENN JA, WEITER
5230 FE0D       00380           CP      0DH             ;IST ES CR ?
5232 CA3A52     00390           JP      Z,REFILE        ;WENN JA, WEITER
5235 12         00400           LD      (DE),A          ;ABSPEICHERN FUER ZIEL
5236 13         00410           INC     DE              ;POINTER <= POINTER + 1
5237 C32952     00420           JP      GETZ2           ;WIEDERHOLE BIS ZIEL
523A 3E0D       00430 REFILE    LD      A,0DH           ;CR IN AKKU
523C 12         00440           LD      (DE),A          ;CR AN FILESPEC.
                00450 ;         ***     JETZT IST ZIELFILE FERTIG
523D 21C252     00460           LD      HL,QFILE        ;QUELL ADDR. IN HL
5240 11F252     00470           LD      DE,QCONV        ;QUELL CONV. ADDR. IN DE
5243 CD1C44     00480           CALL    441CH           ;CONVERTIERE
5246 21DA52     00490           LD      HL,ZFILE        ;ZIEL ADDR. IN HL
5249 111253     00500           LD      DE,ZCONV        ;ZIEL CONV. ADDR. IN DE
524C CD1C44     00510           CALL    441CH           ;CONVERTIERE
                00520 ;         ***     DIE KONVERTIERTEN ADDRESSEN STEHEN JETZT
                00530 ;         ***     IN QCONV UND ZCONV.
524F 11F252     00540           LD      DE,QCONV        ;QUELL FILE IN DE
5252 213253     00550           LD      HL,QDAT         ;DATENSPEICHER
```

```
5255 CD2444           CALL    4424H           ;OEFFNE FILE
              ;       ***     A C H T U N G
              ;       ***     BEI ANDEREN DOS'ES MUSS LRL ANGEGEBEN
              ;       ***     WERDEN (MULTIDOS ERMITTELT SIE SELBST)
5258 FE00             CP      0               ;FEHLERCODE IN AKKU ?
525A C23A55           JP      NZ,DOSERR       ;WENN JA, ABBRUCH
              ;       ***     QUELLFILE IST JETZT OFFEN
525D 111253           LD      DE,ZCONV        ;ZIEL FILE IN DE
5260 213254           LD      HL,ZDAT         ;DATENSPEICHER
5263 0600             LD      B,0             ;LRL VON 256
5265 CD2044           CALL    4420H           ;OPEN / INIT FILE
5268 FE00             CP      0               ;FEHLERCODE IN AKKU ?
526A C23A55           JP      NZ,DOSERR       ;WENN JA, ABBRUCH
              ;       ***     ZIELFILE IST JETZT GEOFFNET
526D 11F252   GETSEC  LD      DE,QCONV        ;QUELLFILE
5270 CD3644           CALL    4436H           ;READ
5273 FE00             CP      0               ;FEHLERCODE IN AKKU ?
5275 CA8052           JP      Z,NOERR         ;WENN NEIN, WEITER
5278 FE1D             CP      1DH             ;EOF ?
527A CAA952           JP      Z,CLOSE         ;WENN JA, CLOSE
527D C33A55           JP      DOSERR          ;SONST ABBRUCH
5280 0600     NOERR   LD      B,0             ;ZAEHLER AUF 0
5282 113254           LD      DE,ZDAT         ;ZIELDATENADDR. IN DE
5285 213253           LD      HL,QDAT         ;QUELLDATENADDR.IN DE
5288 7E       CONV    LD      A,(HL)          ;LADE ZEICHEN
5289 FE8D             CP      8DH             ;IST ES 8DH ?
528B C29052           JP      NZ,NOCHNG       ;WENN NEIN, NO CHANGE
528E 3E0D             LD      A,0DH           ;ASCII CR
5290 12       NOCHNG  LD      (DE),A          ;SPEICHERN AUF ZIEL
5291 13               INC     DE              ;ZIEL + 1
5292 23               INC     HL              ;QUELLE + 1
5293 04               INC     B               ;COUNTER + 1
5294 78               LD      A,B             ;B IN AKKU
5295 B7               OR      A               ;FALGS SETZEN
5296 FE00             CP      0               ;IST ER 0 (=256) ?
5298 C28852           JP      NZ,CONV         ;WENN NEIN, WEITER
529B 111253           LD      DE,ZCONV        ;ZIELFILE IN DE
529E CD3C44           CALL    443CH           ;WRITE UND VERIFY
52A1 FE00             CP      0               ;FEHLERCODE IN AKKU ?
52A3 C23A55           JP      NZ,DOSERR       ;WENN JA, ABBRUCH
52A6 C36D52           JP      GETSEC          ;SONST NAECHSTEN SEKTOR
52A9 11F252   CLOSE   LD      DE,QCONV        ;QUELLFILE IN DE
52AC CD2844           CALL    4428H           ;CLOSE QUELLFILE
52AF FE00             CP      0               ;FEHLERCODE IN AKKU ?
52B1 C23A55           JP      NZ,DOSERR       ;WENN JA, ABBRUCH
52B4 111253           LD      DE,ZCONV        ;ZIELFILE IN DE
52B7 CD2844           CALL    4428H           ;CLOSE ZIELFILE
52BA FE00             CP      0               ;FEHLERCODE IN AKKU ?
52BC C23A55           JP      NZ,DOSERR       ;WENN JA, ABBRUCH
52BF C34055           JP      END
52C2          QFILE   EQU     $               ;START QUELL TAB.
52DA                  ORG     $+24            ;LAENGE = 23 BYTES
52DA          ZFILE   EQU     $               ;START ZIEL TAB.
52F2                  ORG     $+24            ;LAENGE = 23 BYTES
52F2          QCONV   EQU     $               ;CONVERTIERTE QUELLE
```

```
5312         01110         ORG    $+20H     ;LAENGE
5312         01120 ZCONV   EQU    $         ;CONVERTIERTE ZFILE
5332         01130         ORG    $+20H     ;LAENGE
5332         01140 QDAT    EQU    $         ;START QUELL DAT.
5432         01150         ORG    $+256     ;LAENGE
5432         01160 ZDAT    EQU    $         ;START ZIEL DAT.
5532         01170         ORG    $+256     ;LAENGE
5532 3E2F    01180 PAERR   LD     A,2FH     ;FEHLERCODE
5534 CD0944  01190         CALL   4409H     ;AUSGABE VON ERRORCODE
5537 C32D40  01200         JP     402DH     ;ZUM DOS
553A CD0944  01210 DOSERR  CALL   4409H     ;DISPLAY ERROR
553D C32D40  01220         JP     402DH     ;ZUM DOS
5540 C32D40  01230 END     JP     402DH     ;ZUM DOS
5200         01240         END    START
00000 TOTAL ERRORS

CL[illegible]E  52A9
CO[illegible]   5288
D[illegible]ERR 553A
END      5540
GETQ     5212
GETSEC   526D
GETZ     5223
GETZ2    5229
NOCHNG   5290
NOERR    5280
PAERR    5532
QCONV    52F2
QDAT     5332
QFILE    52C2
REFILE   523A
SEARCH   5203
START    5200
ZCONV    5312
ZDAT     5432
ZF[illegible]    52DA
```

(

SERIELL mit PIP

Hallo Clubmitglieder !

Vor einiger Zeit traf sich der Club auf der Jahreshauptversammlung , zumindest war eine anberaumt worden.
Viele von Euch kennen den Ablauf eines Clubtreffens und auch diesmal verlief das Treffen wie gewohnt. Andreas hatte seinen Genie 3s inclusive Lötkolben und Oskar dabei , diverse REH-280 Rechner im Siemens-Gehäuse waren gekommen , Hartmuds M4P und Christopher mit seinem TANDY Modell II inclusive drei 8"-Drives waren dabei und viel Zeit zum fachsimpeln.
Meine Probleme fingen mit dem TANDY Modell II an, denn Christopher wollte ein Terminalprogramm auf dem TANDY haben um Dateien vom Modell 4P auf das Modell II zu übertragen.
Nun , der Genie 3s von Andreas wollte partut nicht mit den 8"-Drives umgehen, und ansonsten war auf dem TANDY Modell II nichts DFUe-mäßiges vorhanden.

Außer ... PIP.

Tja , mit PIP, da ging doch auch was mit der seriellen Verstöpselung. Wenn man genau weiß wies geht ist es einfach , aber ich möchte hier auch erzählen wie man es nicht machen sollte.
Man nehme nicht auf der einen Seite einen KAYPRO II mit TERM und auf der anderen Seite einen TANDY MII und PIP. Beide verstehen sich (gleiche Baudrate vorausgesetz) prächtig, aber wenn der TANDY speichert sendet das Terminalprogramm auf dem Kaypro fleißig weiter, wodurch natürlich einige BITs fehlen und das übertragene Programm kaum laufen dürfte.
Mir ging es so und eigendlich hätte ich dies wissen müßen, aber da ich den KAYPRO auf dem Clubtreffen erst bekam war ich natürlich ganz neugierig auf diesen Rechner und beachtete so eine wichtige Voraussetzung für die serielle Übertragung binärer Daten nicht, welche im normalen Terminalmodus selten gebraucht wird.
Es muß ein Handshake her. Als ich mir dies alles wieder klar gemacht hatte war schon einiges an kohlensäurehaltigem Gebräu die Kehle hinunter, alkohohlfrei versteht sich und ein Vormittag hiermit verbracht.
PIP gehört zur Standardsoftware eines CP/M - Rechners und kann auch die seriellen Schnittstellen bedienen, vorausgesetzt PIP (für CP/M2.2) ist hierfür gepatcht. PIP für CP/M 3.x macht dies ohne Änderungen, da die Devicetreiber AUXIN und AUXOUT die serielle Schnittstelle bedienen und mit dem Programm DEVICE die Parameter der Schnittstellen eingestellt werden können.
Leider ist bei der seriellen Datenübertragung das Controll-Z die Dateiendmarke und in einer binären Datei ist oft ein Controll-Z als Bildschirmsteuerzeichen oder sonstiges mitten im Programmcode enthalten.
Also geht es nicht so einfach mit der Übertragung einer binären Datei, welche schon vorzeitig durch Controll-Z beendet wird, aber es gibt da Abhilfe.

Die binäre Datei wird in eine HEX-Datei umgewandelt.

Mit dem Programm UNLOAD wird aus einer binären Datei eine Filename.HEX Datei erstellt.

(

Die HEX-Datei besteht nur aus hexadezimalen Zahlen und enthält so keine bei der serielle Übertragung störenden Zeichen.
Nach der korrekten Übertragung auf den Zielrechner wird die HEX-Datei mit LOAD, HEXCOM, DDT(Z) oder dergleichen in eine COM-Datei umgewandelt. LOAD und DDT wird dem CP/M2.x mitgeliefert und HEXCOM ist dem CP/M3.x beigefügt.

Die Syntax dieser Programme ist einfach:

Befehl	Eingabe	Ergebniss
UNLOAD	Filename.COM ----->	Filename.HEX
LOAD	Filename.HEX ----->	Filename.COM
HEXCOM	Filename.HEX ----->	Filename.COM

DDT(Z) und SID(Z) laden auch HEX-Dateien und speichern diese Dateien dann als COM-Datei ab. Bei DDT muß mit SAVE die Datei abgespeichert werden, während SID den Befehl W(rite) mitbringt.
Ihr seht es gibt einige Möglichkeiten eine HEX/COM - Datei zu bearbeiten.
Diese HEX-Datei ist also Voraussetzung für unsere problemlose serielle Datenübertragung. Nachdem dies nun bekannt ist und auf beiden Rechnern PIP zur Verfügung steht sucht man sich ein NULL-MODEM Kabel und meistens klappts. Falls beide Rechner die gleiche Belegung der RS232 Schnittstelle haben, sollte eine Probeübertragung klappen, ansonsten muß man an einem der Stecker wahrscheinlich PIN 2(TXD) und PIN 3(RXD) tauschen.
Hier noch mal zur Erinnnerung - zwei Rechner sind vorhanden, gleiche Baudzahl eingestellt, Kabel ist richtig konfiguriert und nun sollte ein ASCII-Text zum Schnittstellentest übertragen werden. Glücklicherweise war das PIP des TANDY MII schon entsprechend an die serielle Schnittstelle angepaßt und mein Rechner ist ein CP/M3 Gerät, und dessen PIP kann die serielle Schnittstelle von Haus aus bedienen.
Die Änderung an PIP unter CP/M 2.x beschreibe ich am Ende es Artikels.

Testen der seriellen Verbindung:

```
SENDER:       PIP PUN:=b:Filename[EB]
EMPFÄNGER:    PIP b:Filename:=RDR:[EB]
```

Bedeutung der Optionen E und B:
PIP schickt mit der OPTION "E" die Datei auch auf das angeschlossene Display, was aber nur bei Textdateien einen Sinn ergibt und hier nur zum Testen der ASCII-Übertragung gebraucht wird.
Mit der Option "B" arbeitet PIP im BLOCK-Modus und die übertragene Datei wird in Blöcken zu je 256 Bytes gesendet. Die Daten werden blockweise in den Puffer gelesen und zum Zielgerät übertragen, bevor ein neuer Block gelesen werden kann. Das Blockende-Zeichen ist der ASCII-Code 19, d.h. CTRL-S oder auch X-OFF.

Der Block-Modus ist also ein X-ON/X-OFF Protokoll und verhindert, daß beim Speichern der Daten welche verloren gehen, bzw. nicht empfangen werden.
Also, bei Datenübertragung mit PIP über die RS232 Schnittstelle nur im BLOCK-Modus arbeiten.
Falls der oben gezeigt "Test" der Verbindung fehlerfrei funktioniert hat, kann jetzt die eigendliche Datei im HEX-Format übertragen , und die HEX-Datei anschließend auf dem Zielrechner mit HEXCOM oder LOAD in eine COM-Datei umgewandelt werden.
Ein kleines Problem haben wir aber nur kurz angesprochen und wollen das hier nachholen.
PIP, unter CP/M3, ist für AUXIN/AUXOUT-Device generiert worden. Jeder CP/M 3.x (CP/M+) Rechner kann mit PIP für CP/M 3.0 die serielle Schnittstelle bedienen.
Unter CP/M 2.x haben wir leider nicht diese Standardisierung und PIP muß von der Lage der seriellen Schnittstelle informiert werden.
Im allgemeinen wird der serielle Baustein über IN/OUT-PORTS angesprochen und diese sind hoffentlich in den Handbüchern zum Rechner beschrieben.
PIP muß also wissen an welcher Portadresse der serielle Baustein angesprochen wird und wann der Baustein dazu in der Lage ist. Hierfür ist in PIP im Bereich von 10AH bis 1FFH Platz zur Verfügung gestellt worden.
Es wurden zwei Labels vorgesehen, und zwar INP:(INPUT) an der Stelle 103H. PIP springt, um Daten vom zu übernehmen, die Adresse 103H per CALL-Befehl an. Dann wird in 110H eine kleine Routine durchgeführt, welche das eingetroffene Zeichen an der Stelle 109H ablegt und mit RET nach PIP zurückspringt. Das Zerro-Bit (BIT 8) muß auf Null gesetzt sein.
Ähnliches geschieht auch beim Senden der Daten. Hier wird das Label OUT:(OUTPUT) an der Adresse 106H angesprungen , eine kleine Routine auf 120H angesprungen , welche das zu sendende Zeichen aus Register C liest und an die serielle Schnittstelle weitergibt.
Natürlich müssen die Baudraten der gekoppelten Rechner stimmen, aber da es die verschiedensten seriellen Bausteine gibt kann hier nur empfohlen werden, über ein SETUP Programm oder dergleichen die Baudrate vorher einzustellen. Meist sind Programme vorhanden, mit denen der Druckeranschluß, die Laufwerke und die serielle Schnittstelle konfiguriert werden können. Falls nichts dergleichen machbar ist muß man einen Fachmann fragen und ist in diesem Artikel nicht weiter ausgeführt.

Hier noch eine kleine Routine mit welcher PIP gepatcht werden kann. Damit es für alle nützlich ist auch in 8080 Code:
Die hier genutzten Portadressen sind für den TATUNG TPC2000 gültig.

Prozessor 8080:

```
IN:
    103H     JMP 110H   ; Sprung zur Eingabeunterroutine
    106H     JMP 120H   ; Sprung zur Ausgabeunterroutine
    110H     IN  1DH    ; Port 1DH liefert das Statuswort
    112H     ANI 2      ; Testen ob Zeichen eingetroffen
    114H     JZ  110H   ; Schleife bis Zeichen da ist
    117H     IN  1CH    ; Port 1CH liefert das Zeichen von
                        ; der seriellen Schnittstelle
```

(

```
    119H      ANI 7FH    ; höchstes Bit ausblenden
    11BH      STA 109H   ; Empfangenes Zeichen in 109H ablegen
    11EH      RET        ; und nach PIP zurück

OUT:
    120H      IN  1DH    ; liefert Statuswort
    122H      ANI 1      ; testet ob serieller Baustein bereit
                         ; ist zum senden
    124H      JZ  120H   ; Schleife bis senden möglich
    127H      MOV A,C    ; Auszugebendes Zeichen aus C in A
    128H      OUT 1DH    ; Zeichen über Port  1DH ausgeben
    12AH      RET        ; Sprung aus Sendeunterroutine  zu PIP
```

Prozessor Z80:

```
 IN:
    103H      JR  110H       ; Sprung zur Eingabeunterroutine
    106H      JR  120H       ; Sprung zur Ausgabeunterroutine
    110H      IN  A,(1DH)    ; Port 1DH liefert das Statuswort
    112H      AND 2          ; Testen ob Zeichen eingetroffen
    114H      JR  Z,110H     ; Schleife bis Zeichen da ist
    117H      IN  A,(1CH)    ; Port 1CH  liefert das Zeichen von
                             ; der seriellen Schnittstelle
    119H      AND 7FH        ; höchstes Bit ausblenden
    11BH      LD  (109H),A   ; Empfangenes Zeichen in 109H ablegen
    11EH      RET            ; und nach PIP zurück

 OUT:
    120H      IN  A,(1DH)    ; liefert Statuswort
    122H      AND 1          ; testet ob serieller Baustein bereit
                             ; ist zum senden
    124H      JR  Z,120H     ; Schleife bis senden möglich
    127H      LD  A,C        ; Auszugebendes Zeichen aus C in A
    128H      OUT (1DH),A    ; Zeichen über Port 1DH ausgeben
    12AH      RET            ; Sprung aus Sendeunterroutine wieder
                             ; zu PIP
```

Mit dieser kleinen Anleitung müßte jeder sein PIP für CP/M2.2 patchen können um die seriellen Schnittstellen anzusprechen. Etwas Geduld braucht man hierzu, vor allem da die Konfiguration der seriellen Schnittstelle, bzw. der Stecker doch unterschiedlich sein können.
Unterschieden wird hier zwischen DTE (Data Terminal Equipment) oder deutsch DEE (Daten-Endeinrichtung/en) also Rechner; und DCE (Data Communication Equipment) beziehungsweise DÜE (Daten-Übertragungseinrichtung/en) also Modem.

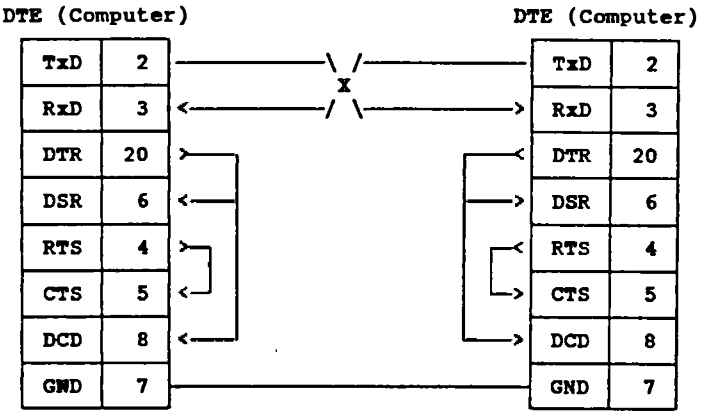
Modem und Computer passen direkt zusammen, da hier Sende- auf Empfangsleitung geschaltet ist. Manchmal ist aber auch der Computer (am Tatung kann dies gejumpert werden) als DCE konfiguriert.
Bei zwei Rechnern, welche als DTE beschaltet sind sollte das nachfolgende Nullmodem für eine funktionierende Verbindung der beiden Rechner sorgen.

In der Regel sind als RS232 Stecker 25-polige D-SUB Typen eingebaut, oder oft bei IBM und kompatiblen platzsparende 9polige D-SUB .
Da ein XON/XOFF - Protokoll genutzt wird reicht ein NULL-Modemkabel aus und braucht nur aus drei Adern zu bestehen, eventuell als vierte Ader eine Abschirmung. Die PINS 4,5,6,8,20 werden am Stecker miteinander verbunden.
Falls jemand Probleme mit der seriellen Übertragung bei seinem Rechner hat stehe ich gerne mit RAT & TAT beiseite. Für folgende Rechner sind bei mir funktionfähige Terminalprogramme zu erhalten: TATUNG TPC2000 - EPSON PX-8 - GENIE 3s - Morrow MD3 - KAYPRO II und Generic CP/M+ Versionen der gängigen Terminalprogramme.

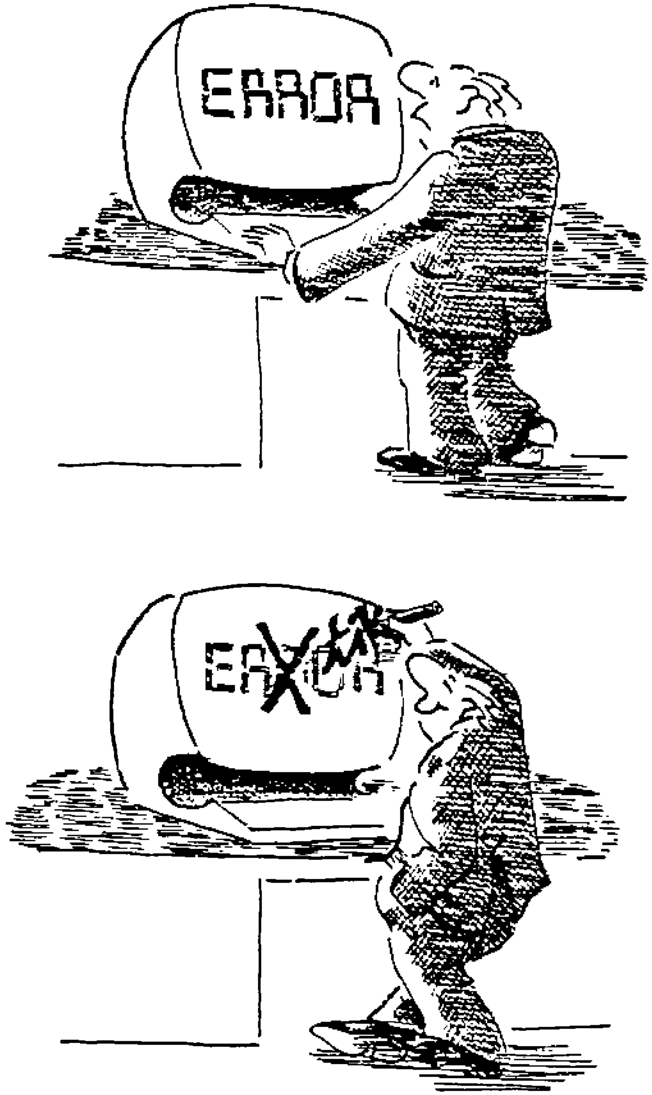
/ Meine E-Mail Adresse: fritz chwolka@solaris.hamm.sub.org /

oder Postadresse laut INFO.

Belegung des Nullmodemkabels bei 25 D-Sub.



Anmerkung: Dieser Artikel wurde mit Wordstar 4 geschrieben und auf einem Panasonic KX-P1124 mit Einzelblatteinzug der Firma Orbitronic ausgedruckt.



© Dietmar Große

Noch serieller mit PIP

Im Info Nr. 35 hat "uns Fritz" geschrieben, wie man im Notfall auch ohne kompatible Laufwerke, bzw. wenn keine Informationen über das Diskettenformat vorliegen, Dateien mit Hilfe von PIP von einem Rechner zum anderen übertragen kann. Dazu noch zwei kleine Anmerkungen/Ergänzungen:

Wenn man PIP als Parameter für die Übertragung zusätzlich noch ein "H" (Hex) mitgibt, schaltet es in den Hex-Modus und überprüft, ob mit der übertragenen Datei (in der Z80-Welt üblicherweise im sog. Intel-Hex-Format) alles stimmt. Diesen Service hat man nur mit "B" (Block) alleine nicht und kann verfälschte Zeichen höchstens daran erkennen, daß der HEX-COM-Wandler (HEXCOM, LOAD o.ä.) hinterher meckert, oder das Programm nicht läuft. Im Hex-Modus stellt PIP anhand der Prüfsumme in jeder Zeile fest, ob möglicherweise ein Fehler aufgetreten ist und meldet dies dann sofort. Wenn man nur beim sendenden Rechner ein E (Echo) als Parameter angibt, wird dieser weiter gebremst, was dem empfangenden Rechner etwas mehr Zeit gibt, die ankommenden Zeichen zu verarbeiten. Normalerweise dürfte das nur bei einem relativ langsamen Rechner auf der einen und einem schnelleren Rechner auf der anderen Seite notwendig sein, aber aus Erfahrung würde ich sagen, daß man das immer so machen sollte. Auch ist es nicht unbedingt gesagt, daß es mit einer kleineren Baudrate immer besser geht, ich habe es schon erlebt, daß es mit 1200 Baud einwandfrei ging, mit 300 Baud aber nur Schrott angekommen ist. Warum ? Keine Ahnung, aber wenn alle Geheimnisse gelöst wären, wäre es ja langweilig. Außerdem sollte man beim sendenden Rechner die Schnittstelle tunlichst auf das XON/XOFF-Protokoll einstellen, sonst funktioniert der Blockmodus nicht. PIP sendet zwar XOFF- und XON-Zeichen, aber die Übertragung wirklich anhalten muß der Schnittstellentreiber bzw. das BIOS. Wenn sich der empfangende Rechner nach der Übertragung nicht mehr zurückmeldet, kann man ihm mit "PIP PUN:=EOF:" bzw. "PIP OUT:=EOF:" noch ein explizites Dateiende zukommen lassen.

Was in Fritz' Artikel vielleicht auch nicht ganz klar rausgekommen ist und viel Zeit und Nerven kosten kann: Der Test mit

PIP PUN:=filename[EB] und PIP filename=RDR:[EB]

(hinter dem Filenamen darf, entgegen der Angabe von Fritz, übrigens <u>KEIN</u> Doppelpunkt kommen, sonst gibt's eine Fehlermeldung !!)

funktioniert nur, wenn das CP/M das IO-Byte unterstützt ! Es gibt Implementationen, die das Byte einfach nicht benutzen und dann wundert man sich, daß nichts geht. Ausprobieren kann man das mit "PIP LST:=CON:", mit dem alle Eingaben von der Tastatur auf den Drucker umgelenkt werden sollten. Da LST, genauso wie EOF weiter oben, ein logisches Gerät ist, muß hier zur Unterscheidung von einem Filenamen nun tatsächlich ein Doppelpunkt dahinter stehen. Wenn man aber ein komfortabes BIOS hat, das u.a. bei einem leeren Empfangspuffer brav selber auf ein neues Zeichen wartet und der Testtext so fehlerfrei angekommen ist, ist man schon fertig und braucht die ganze Patcherei garnicht. Dann kann man mit PUN: bzw. RDR: arbeiten und im Prinzip sofort loslegen und beim CP/M Plus, wo man in beiden Fällen nur über AUX: geht, muß man sowieso nicht fummeln und kann auch gleich anfangen.

Wenn es aber doch sein muß, sei es weil das mit dem IO-Byte nicht funktioniert, oder dauernd Zeichen ankommen, ohne daß der andere Rechner

(

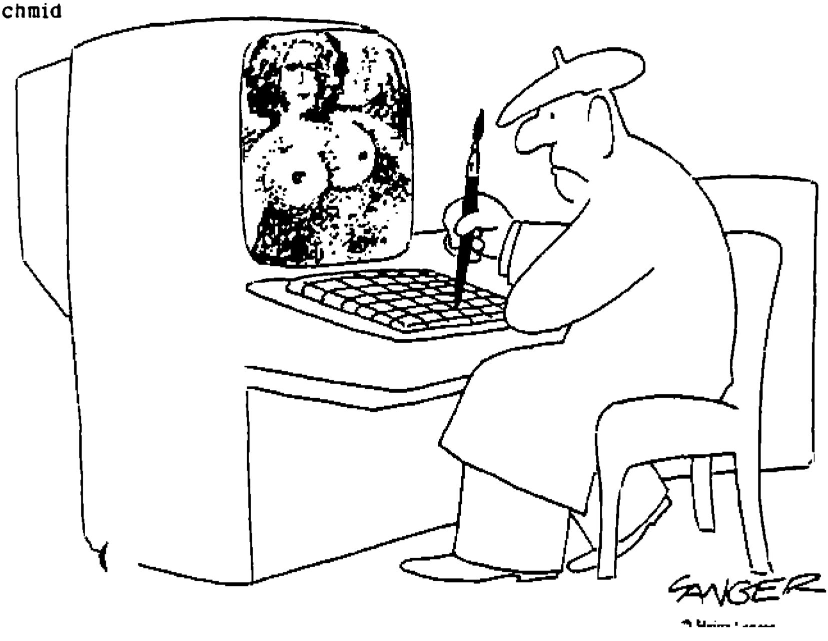
sendet, und man PIP schließlich so gepatcht hat, daß es die Ports direkt anspricht, muß die Ein- und Ausgabe folgendermaßen aussehen: (PUN: und RDR: gehen nach wie vor über das Betriebssystem, da helfen alle Patches nichts !)

PIP OUT:=filename[EBH] und PIP filename=INP:[BH]

Wenn diese Hürde dann schließlich genommen ist und die Übertragung klappt, sollte man so schnell wie möglich LYNC rüberziehen. Das Programm besteht aus dem eigentlichen COM-File (LYNC.COM), einem Installer (LYNC-INST.COM) und einer Treiber-Bibliothek (IO.DRV). Am besten konfiguriert man das Programm schon vorher, dann braucht man nur ein File zu übertragen. Das File ist auch als Hex-File so klein, daß es zusammen mit PIP in den Arbeitsspeicher paßt, sodaß im Block-Modus erst nach Ende der Übertragung auf die (langsame) Floppy zugegriffen werden muß und es so keine Probleme mit dem Protokoll geben kann.

Warum aber nun ausgerechnet LYNC ? Ganz einfach, es ist verhältnismäßig klein und so ziemlich das einfachste und sicherste Programm für diesen Job, bei dem man eigentlich nichts falsch machen kann. Der Installer arbeitet menügeführt und man muß nur die Ports der seriellen Schnittstelle, die man ja eventuell schon von den Patches in PIP kennt, per Hand eingeben; für viele Rechner (u.a. TRS-80 Model I und IV) gibt es einen fertigen Menüpunkt, den man nur anzuwählen braucht. Danach kann man von einem der beiden Rechnern beliebig Files senden und empfangen, ohne daß am anderen Rechner jemand sitzen muß. Man kann sich z.B. sogar das Directory des anderen Rechners ferngesteuert ansehen und dann mit SEND und FETCH loslegen. Das ist viel bequemer als mit Kermit oder sonstigen Terminalprogrammen, wo man immer noch einen Helfer braucht. Die Übertragung wird durch ein LYNC-eigenes Protokoll gesichert und nichtmal das mutwillige Unterbrechen der Leitung bringt es aus dem Tritt. Nachdem die Verbindung wieder steht, geht es sofort weiter. Außerdem überprüft sich LYNC beim Starten selber, sodaß man dadurch nochmal eine Kontrolle hat, ob die Übertragung des Hex-Files fehlerfrei war.

Alexander Schmid



.Hardware

Das Club 80 Netzwerk

oder: Poor Man's LAN

von: Rainer Schmitz, Göppingen
Stand: 11. August 1991

Warum?

Jeder kennt wohl die Situation, wenn man auf einen CP/M-Rechner trifft, dessen Diskettenformat man (beziehungsweise die eigene Kiste!) nicht verarbeiten kann und es steht ein Datentransfer an. Sofern beide Rechner über eine serielle Schnittstelle und passende Modemprogramme (LYNC, ZMP, QTERM, MEX, KERMIT oder ähnliche) verfügen, kann man den Datentransfer zwar langsam, aber sicher über ein Nullmodem-Kabel durchführen. Schwierig wird die Sache aber, wenn mehrere Rechner miteinander in Verbindung treten sollen, wie es bei umfangreichen "Gebietsrechenzentren" à la Fritz Chwolka oder Helmut Jungkunz oder bei Clubtreffen (auch ausgefallenen...) vorkommen kann. Man steht dann vor der Alternative, entweder bei den Hobbykollegen mit dem Formatmanager (hallo Z280-Besitzer!) Schlange zu stehen, um die Datenpumpe anlaufen zu lassen oder mit einem möglichst langen Nullmodemkabel und einigen Adaptern und "Geschlechtswechslern" zu versuchen, die RS-232-"Norm" zu umgehen und die Kommunikation zu beginnen. Dabei stört mich, daß man den "Stolperdraht" regelmäßig umstecken muß, um mit einem anderen Comp zu kommunizieren. Kurz, beide Lösungen können nicht als optimal betrachtet werden.

Die Super-Luxus-Lösung des Problems wäre natürlich ein LAN (Local Area Network) in PC-Manier, dem steht allerdings entgegen, daß kaum jemand über ein paar überzählige Ethernet-Karten und ein passendes CP/M-NetBIOS verfügt, außerdem ist die zum Betrieb nötige Kabelage nebst Steckverbindern nicht gerade billig. Zusätzlich stellt sich die Frage, ob ein LAN nicht unnötiger "Overkill" für die oben angesprochene Aufgabenstellung ist.
Der Wunschzettel, den ich daraufhin im Geiste aufstellte, sah folgendermaßen aus:

- kein Umbau am Rechner selbst erforderlich
- keine besondere Hard- und Software nötig, speziell kein "Rumfummeln" am Betriebssystem, möglichst Verwendung vorhandener Modemprogramme
- niedriger Materialaufwand, damit niedriger Geldaufwand
- einfach nachzubauen, auch für Nichtelektroniker
- störungssicher und robust

Wie?

Einen brauchbaren Lösungsansatz fand ich im Leitartikel von Herwig Feichtinger in der mc Heft 7/1986 : dort beschrieb er ein einfaches Netzwerk, das in der mc-Redaktion eingesetzt wurde, um Texte zwischen den Redakteuren auszutauschen, da diese es leid waren, dauernd Disketten hin und her zu tragen. Es basierte darauf, daß die seriellen Schnittstellen in einer Art "verdrahtetem Oder" miteinander verbunden waren. Der Materialaufwand war extrem gering (drei Widerstände, eine Diode), als Verbindungskabel war Koaxkabel vorgeschlagen. Ich habe diese Schaltung aufgebaut und ausprobiert und dabei festgestellt, daß die "handelsüblichen" Modemprogramme damit bei Datentransfer nicht liefen: die Ursache war schnell gefunden. Wenn man sich die Schaltung (Bild 1) anschaut, wird man feststellen, daß der Sender sein eigenes Echo mithört! Das Modemprogramm erwartet aber bei einem Datentransfer nicht irgendetwas, sondern ein entsprechendes Quittungszeichen, mit dem das empfangende Programm den erfolgreichen Empfang eines Datenblocks kundtut (oder auch nicht). Beim Terminalbetrieb stört das Echo nicht weiter, da man dann gleich am Schirm sieht, was man eingetippt hat, sofern man nicht im Terminalmodus "Local Echo" eingestellt hat, denn dann hat man den "Jägermeister-Effekt": man sieht alles doppelt!

Die Schaltung funktioniert so: der Widerstand R3 in der Masseleitung soll ganz einfach Ausgleichsströme zwischen den Rechnern (unterschiedliche Massepotentiale) auf ein ungefährliches Maß begrenzen. Durch den Widerstand R2 in der RXD-Leitung wird der Empfängereingang hochohmiger gemacht, so daß mehr Eingänge parallelgeschaltet werden können. Der Senderausgang TXD zieht bei Ruhepegel den Anschluß Netzwerk+ über den Widerstand R1 unter die Empfängerschaltschwelle, bei aktivem Pegel wird R1 durch D1 überbrückt. Die Abhilfe des Problems beim Datentransfer erforderte kurzes Nachdenken und noch ein paar Bauteile der billigen Natur. In Bild 2 ist zu erkennen, wie's gemacht wird: der Transistor T1 schließt ganz einfach den Empfängereingang kurz, wenn der zugehörige Senderausgang aktiv wird. Angesteuert wird er mit R4 von TXD aus. D2 verhindert, daß der Transistor durch die negative Ausgangsspannung von TXD bei Ruhepegel zerstört wird.

Bei geschicktem Aufbau ist es möglich, die gesamte Schaltung in eine Steckerhaube für einen Submin-D-Stecker (so die offizielle Bezeichnung) einzubauen. Damit der Kontakt zum restlichen Netzwerk hergestellt werden kann, sollte jeder Anschluß mit zwei parallelgeschalteten Cinch-Buchsen (auch als RCA-Buchsen bekannt) versehen werden. Zum Anschluß gehört natürlich auch das passende Verbindungs[illegible]l: ich empfehle aus eigener Erfahrung ganz normales NF-

Kabel, und zwar die einadrige abgeschirmte Sorte. Bei den niedrigen Kabelpreisen halte ich eine Lange von drei Metern pro Station für angemessen, das reicht in den meisten Fällen (sprich: Clubtreffen). Der Innenleiter (Seele) kommt an den Netzwerk + Anschluß, die Abschirmung an den Netzwerk - Anschluß, das entspricht jeweils Stift und Massehülse beim Cinch-Stecher. Den Stecker am freien Kabelende steckt man dann in die freie Netzwerk-Buchse des nächsten Rechners.

Versuche ergaben noch bei ca. 30 m Kabellänge einwandfreie übertragungen bei 9600 Bd, was ja wohl ausreicht. Ich sehe aber keinen Grund, warum es bei den höheren Baudraten nicht auch gehen soll! Das Oszilloskop zeigte jedenfalls bei 19200 Bd und ca. 250 m Kabel (!) am freien Ende noch schöne saubere Flanken. Ach ja, noch was: wenn der Rechner auf Hardware-Handshake Wert legt, sollte man ihm den Gefallen tun und RTS mit CTS verbinden (Pins 4 und 5), dasselbe gilt für die Pins 6,8 und 20 (das Dreieck rund um Masse, Pin 7). Diese Pinbelegung gilt aber nur für 25-polige Verbinder nach Norm! Bei meiner Kiste (Joyce) kann ich aber wählen, ob ich Handshake haben will oder nicht (via SETSIO). Besitzer anderer Rechner sollten lieber ins Handbuch schauen (so vorhanden!). Gerade bei Normbelegungen sind die Hersteller ja besonders einfallsreich...

Eine weitere Frage wäre noch in einem Feldversuch zu klären: nämlich die, wieviele Stationen das Netz verträgt. Nach meinen Berechnungen sollten bis zu sechs Stationen möglich sein, das hängt aber von verschiedenen Randbedingungen ab. Der in den meisten Schnittstellenschaltungen verwendete 1488 (75488) kann durchschnittlich 10 mA Ausgangsstrom liefern. Da jeder Eingang eines Empfängers 1489 ca. 1.2 mA Eingangsstrom oberhalb der Schaltschwelle benötigt, kann man sich schnell ausrechnen, wann Schluß ist. In der Realität gibt es aber Streuungen der Bauteilewerte, die diese Grenze nach oben oder unten verschieben können, und das probiert man wohl am besten aus. Bei der Originalschaltung aus der mc war bei drei Stationen am Netz Schluß, wie man durch Berechnung der Ströme im Netzwerk leicht nachweisen kann! Daher habe ich auch einige Werte gegenüber der Ursprungsschaltung verändert, ohne daß bei den Versuchen Probleme auftraten.

Womit?

Wie weiter oben schon angedeutet wurde, war angestrebt, das Netzwerk mit Standardsoftware zu betreiben, um den bei Software-Neuentwicklungen üblichen Zyklus Editieren - Assemblieren - Systemabsturz zu umgehen (ich gebe zu, mir fehlte dazu die Motivation und Erfahrung). Bei den ersten Versuchen setzte ich (oh Schande!!) mangels eines weiteren CP/M-Rechners einen kompandebilen Computer (mit MeSsy-DOS) mit Procomm ein, dazu meinen "Taschenknecht"

Portfolio. Alle diese Rechner verstanden sich unter XMODEM prächtig! Auf CP/M-Seite setzte ich ZMP ein, das ich inzwischen auch auf meinen neuen Rechner, einen Wavemate Super Bullet, angepaßt habe (falls jemand den Quelltext für den Overlay braucht, bei mir melden!).

Einen ersten Härtetest bestand das Netzwerk bein Treffen des Club 80 am 9. bis 12. Mai in Leipheim (nicht Leibheim, Gerald!). Wie bereits im letzten Clubinfo vermerkt, funktionierte das Ganze mit ca. 20 m Kabel recht gut. Es wurde hauptsächlich ZMP und LYNC zur Kommunikation eingesetzt, wobei man mit LYNC sogar den anderen Rechner "fernsteuern" kann, also Laufwerke wechseln, Directories anzeigen und dergleichen mehr (nur Disketten wechseln muß man von Hand!). Auf diese Weise gelangten einige Programmpakete aus der PD von einem zum anderen Rechner, und das über nur zwei Drähte! Mit 9600 Bd geht der Filetransfer sogar recht flott, so daß eigentlich bei einem der nächsten Treffen ein Rechner mit einer mit dem Vereinsarchiv gefüllten Platte und LYNC in der Ecke stehen und als "Fileserver" dienen könnte. Denkbar wäre auch ein Mailboxprogramm, mit dem man sich unterhalten könnte. Dabei muß man sich aber bewußt sein, daß dann nur Kommunikation mit dem Fileserver oder der Mailbox möglich ist, nicht aber mit anderen Rechnern. Das Netzwerk funktioniert ähnlich wie CB-Funk, wenn einer spricht, ist halt der Kanal belegt und die anderen müssen warten. Wenn zwei gleichzeitig sprechen, kommt Müll raus, aber das ist der Preis der Einfachheit. Besonders lustige Effekte kann man beobachten, wenn man mehr als zwei Stationen unter LYNC betreibt: da das Programm versucht, bei jeder Aktion eine Verbindung herzustellen, und jedesmal zwei antworten, sind bald alle drei damit beschäftigt, sich gegenseitig "die Hand zu schütteln". Die Konsequenz der Eigenheiten dieses Netzwerks ist daher: Disziplin der Benutzer! Bevor man die Leitung belegt, sollte man im Terminal-Modus nachschauen, ob gerade Ruhe herrscht, und erst dann loslegen. Wenn dann doch zwei gleichzeitig senden, passiert außer einer versauten Übertragung nichts weiter, das heißt, ihr müßt in diesem Fall nicht mit kleinen Atompilzen, die eurem Rechner entsteigen, rechnen! (Das gilt aber nur, wenn ihr bei der Hardware keinen Mist gebaut habt, auch 220 V sind von der Netzwerk-Teilnahme ausgeschlossen!)

Zusammenfassung (für Diagonalleser)

In diesem Artikel habe ich ein Netzwerk für den Filetransfer "nach Holzfällerart" beschrieben, das für unsere Clubtreffen und den heimischen Gebrauch gedacht war. Seine Einfachheit wird durch ein paar Funktionseinschränkungen erkauft, diese stellten sich aber in der Praxis als tolerabel heraus, besonders bei Berücksichtigung des Aufwands. Hinzu kommt, daß das Netzwerk unabhängig vom Betriebssystem funktioniert, also auch für PC's, Atari's, Amiga's und andere geeignet ist, sofern diese über eine serielle Schnittstelle nach RS 232 (V-24) mit TXD und RXD verfügen. Ob es zu einer Kommunikation kommt, hängt dann nur noch von der (Modem-)Software ab.
Soweit mein Bauvorschlag, seht zu, was ihr draus macht, beim nächsten Clubtreffen will ich jede Menge CP/M-Rechner mit Netzwerkanschluß sehen!

Ausblick (auch das noch!)

Seit ich mit diesem Artikel angefangen habe, erhielt ich einige faszinierende Informationen aus der Welt des Amateurfunks: dort ist im UKW-Band ein drahtloses Netzwerk namens "Packet Radio" entstanden. Es handelt sich dabei um ein modifiziertes X.25-Protokoll à la ISDN. Dadurch ist es möglich, auf einem Funkkanal mehrere Verbindungen gleichzeitig zwischen verschiedenen Stationen aufzubauen. Speziell die Hardware dürfte für uns interessant sein: der typische TNC (das ist das Modem zum Anschluß eines Computers an ein Funkgerät) ist mit einer Z80 CPU und SIO (wegen dem synchronen X.25-Protokoll) aufgebaut. Ein Bekannter besorgt mir Schaltpläne und ein Eprom mit der Betriebssoftware, so daß man sich die Sache mal anschauen kann. ZMP enthalt ja ein Packet Radio Overlay namens ZMYAPP, das erleichtert die Sache gewaltig. Soweit also die neuesten Infos zum Thema Ausbau des Netzwerks.

Bild 1

```
                12kΩ R1
TXD O-------+-----[_____]--------+------O Netzwerk +
            |                    |        (Seele)
            |     |\ | 1N4148    |
            +-----|->|-----------+
                  |/ |    D1     |
                                 |
RXD O-------------[_____]--------+
               1.8kΩ R2
Gnd O-------------[_____]---------------O Netzwerk -
                                          (Schirm)
                 22Ω R3
```

Netzwerkankopplung nach mc 7/1986

Bild 2

```
                 330kΩ R1
TXD O---+---+-----[_____]---------------O Netzwerk +
        |   |                    |        (Seele)
        L   |     |\ | 1N4148    |
 47kΩ | |   +-----|->|-----------+
  R4  |_|         |/ |    D1     |
        T                        |
        |                        |
RXD O--(------+---[_____]--------+
        |     |
        |     |    5.6kΩ
        |     |     R2
        |     | C
        |  B /
        +---|      BC 546 B
1N4148 _|_  \| E     T1
  D2   /|\   |
Gnd O---+-----+---[_____]---------------O Netzwerk -
                                          (Schirm)
                 22Ω R3
```

Erweiterte Netzwerkankopplung

Z E U S

© Copyright
1983

**Cosmopolitan
Electronics
Corporation**

5700 Plymouth Road
Ann Arbor, MI 48105

INTRODUCTION

Cosmopolitan Electronics Corporation distributes all software on an "as-is" basis without warranty. Cosmopolitan Electronics Corporation shall not be liable or responsible to the purchaser with respect to liability, loss, or damage caused or alleged to be caused directly or Indirectly by the use of this software, which Includes but is not limited to any Interruption of service, loss of business, anticipatory profits, or consequential damage resulting from the operation or use of this software.

ZEUS is copyrighted with all rights reserved. Copying, duplicating, selling, or any unauthorized distributing of this product is expressly forbidden. In accepting this product, the purchaser recognizes and accepts this agreement.

Throughout this manual references are made to trademarked products. The (tm) symbol is used once here to serve throughout the manual.

DOSPLUS (tm)	Micro Systems Software.
LDOS (tm)	Logical Systems, Inc.
NEWDOS/80 (tm)	Apparat, Inc.
TRSDOS (tm)	Radio Shack, a division of Tandy Corp.
Z80 (tm)	Zilog, Inc.

ZEUS
Copyright (c) 1983 by Cosmopolitan Electronics Corporation.

ZEUS
Copyright (c) 1983 by Cosmopolitan Electronics Corporation.

This Manual
Copyright (c) 1983 by Cosmopolitan Electronics Corporation.

ACKNOWLEDGEMENT

We would like to thank Rowland Archer, Johnn Audritsh, Mark Feldman, Andrew Gariepy, Phil Manfield, Mark Robinson, and Dave Welsh for taking the time to beta test and provide constructive input for ZEUS.

ZEUS COMMANDS

A Assemble.

B Calculator.

C Global change.

D Delete line(s).

E Edit line.

F Find label.

G LPrint format.

H LPrint line (s).

I Insert.

J LPrint raw data.

K Kill file.

L Load file.

M Move/Duplicate line(s).

N Reset buffer.

O Object/Operand reference.

P Print line (s).

Q Exit.

R Reference.

S Save file.

T Print label table.

U Usage.

X Recovery on reentry.

Print, as used in the following text, means output to the device in the video DCB (401DH).

Lprint, as used in the following text, means output to the device in the printer DCB (4025H).

INTRODUCTION

The "ZEUS" files on the *ZEUS* diskette are:

ZEUS/CMD - Editor/Assembler.

ZCON/CMD - DOS configuration program and default forms.

Z80E/ASM - Source code of 794 Z80 intructions in numeric order.

ZEUS is executed by entering:

    ZEUS

At the DOS command level.

ZEUS/CMD will work "as is" on DBLDOS, DOSPLUS, MULTIDOS, NEWDOS, ULTRADOS, Z'DOS, and Model III TRSDOS version 1.3. However, ZEUS/CMD must be modified to work with Model I TRSDOS, Model III TRSDOS 1.0, 1.1, 1.2, VTOS and LDOS.

The ZEUS diskette is a read only, 35 cylinder, single sided diskette, with the directory on cylinder/lump 17. (The DOS is described in Section 4).

A sophisticated DOS (DOSPLUS, LDOS, MULTIDOS, NEWDOS/80, & Z'DOS) can be configured to read the ZEUS diskette directly. However, if Model III TRSDOS is used, then "CONVERT" the files from the ZEUS diskette to the Model III TRSDOS diskette.

If a sophisticated DOS will be used in drive zero, copy the three files to this DOS.

Model I users of TRSDOS, NEWDOS/2.1, VTOS, ULTRADOS, and DBLDOS will be confronted with the "address mark" syndrom. To avoid grinding away at the ZEUS diskette ten times, boot the ZEUS diskette in drive 0 and place the ancient DOS into drive one. Use ZEUS copy function, "$", to copy the files using the following syntax:

With two drives

$ ZEUS/CMD 1

or with one drive

$ $ZEUS/CMD   {swap as directed. NOT every prompt requires a swap!}

Use the appropriate procedure to copy the other two files to the ancient DOS. After all three files have been copied, the user must reset the directory address marks on the ancient DOS by keying in:

SETDIR :d (0)   {d = drive ancient DOS will be in}

ZEUS is now on the DOS to be used in drive zero.

INTRODUCTION

ZEUS has its own forms to explicitly control the printout of source text. ZEUS forms must be active to utilize all of the assemblers printout functions.

To load the forms code, enter

    ZEUS F

At the DOS comand level. Or configure ZEUS/CMD to automatically load the forms code.

The configuration program, ZCON/CMD, will modify ZEUS/CMD to execute properly with all of the popular operating systems (DOS), alter the default ZEUS forms parameters, and configure ZEUS/CMD to automatically load the ZEUS forms code upon entry.

ZCON/CMD is accessed by entering:

    ZCON

In responce to the DOS command prompt.

ZCON/CMD is menu driven, and self explanatory.

The default condition of ZEUS is:

DOS selection = 2.

FORMS parameters
- printed lines per page = 60.
- number of charaacters per line = 80
- total number of printed lines per page = 66
- auto forms = N

FILENAMES

ZEUS will automatically append "/ASM" to all user prompted source filenames, and "/CMD to all object code filenames which do not include an extention. Since, the GET instruction is not user prompted, an extention, if part of the GET file, must be in the filespec of the GET instruction. To access a user prompted file without an extention, append a "/" immediately behind the filename.

EXAMPLES:

User input	Pilespec given to DOS.
DOGGY	DOGGY/ASM
KOOL.MAN	KOOL/ASM.MAN
COLUMNS:2	COLUMNS/ASM:2
TABLES.MONKEY:0	TABLES/ASM.MONKEY:0
MOVIE/INC	MOVIE/INC
VIDEO/SOR.CAR	VIDEO/SOR.CAR
KILLER/GET:2	KILLER/GET:2
OVERMAKE/SRC.FUNNY:1	OVERMAKE/SRC.FUNNY:1
SEXY/	SEXY
MUCHO/.GIRL	MUCHO.GIRL
W0MAN/:3	WOMAN:3
COLDHEAT/.SUN:3	COLDHEAT.SUN:3

ZEUS I/O

Throughout this manual, the words print and Lprint are used to designate ZEUS output.

Print is used whenever output is directed to the device in the video DCB at 401DH.

Lprint is used whenever output is directed to the device in the printer DCB at 4025H.

ZEUS uses the *KI device at 401EH for all keyboard input, but scans the 3800H page of RAM to implement PAUSE, control keys, and to distinguish a shift "ENTER" from a non-shift "ENTER".

Disk I/O is performed by using the ROM calls at 0013H and 001BH for byte I/O, and the DOS vectors at 4436H and 4439H for sector I/O. ZEUS uses 256 as the logical record length for all files.

GLOSSARY

ASCII format - Alphanumeric source code stored without line numbers.

Current line - The last printed text buffer line.

EDTASN format - Source code stored with a header byte, X'D3', followed by a six character name, alphanumeric text with each line preceded by a five character line number with the high bit set, and a terminator byte, X'lA'.

LABEL - A string used to symbolicly represent a value.

Mnemonic - An assembly language instruction.

Object code - Binary code directly executable by the microprocessor.

Operator - A single symbol describing an operation to be performed.

Psuedo-op - Special orders given by the programmer to the assembler, whichdo not generate object code.

Relative address - The value for the beginning of the current instruction .

Source code - The symbolic representation of LABELS and mnemonics, which the assembler translates into specific object code.

Text buffer - The RAM resident storage area for source code.

REFERENCE MANUAL NOTATION

< >	Carats enclose a key to be pressed. The "<" and ">" are not keyed in. i.e. <ENTER> = Press the "ENTER" key. i.e. <BREAK> = Press the "BREAK" key. i.e. <N> = Press the "N" key.
{ }	Braces enclose manual comments. The braces "{and the enclosed comment}" are not keyed in. i.e. {will be shown...}
{ }	Brackets enclose optional parameters. The brackets "[" and "]" are not keyed in. i.e. [H]
...	Ellipses represent the repetition of an optional parameter, i.e. [,EXP][,EXP][,...]
RELNUM	RELNUM is a parameter which may be 1) a number which represents a relative line in the text buffer, or 2) a LABEL which indirectly represents a relative line in the text buffer where LABEL is in the label field.
#	The pound sign is a RELNUM parameter for line #1.
*	The asterisk is a RELNUM parameter for the highest line.
•	The period is a RELNUM parameter for the current line.
<	Less than modifier for RELNUM. (One number less) i.e. <354. {parameter = 353}

ZEUS SYNTAX

ZEUS FUNCTIONAL COMMANDS

SINGLE KEYSTROKE COMMANDS

Single keystroke commands, must be the first keystroke after the prompt, ">" is printed.

<UP-ARROW>	Print next lower line,if exists, otherwise print line #1 (first line).
<DOWN-ARROW>	Print next higher line, if exists, otherwise null.
<ENTER>	Print page from current line.
<SHIFT><ENTER>	Print page to current line, first press. Subsequent presses, print page to current line less 14.
<SHIFT><UP-ARROW>	Print line #1 (first line).
<SHIFT><RIGHT-ARROW>	Print the highest line.
. {period}	Print current line.

PAUSE COMMAND

<SPACE>	The <SPACE> will pause printing. After a pause, single step with <SPACE>, continue with <ENTER>, or abort with <BREAK>•

CURSOR MOVES

<RIGHT-ARROW>	Move the cursor to the next TAB position.
<SHIFT><SPACE>	Move the cursor to the next TAB position.
<LEFT-ARROW>	Backspace one position and remove character at the new cursor position.
<SHIFT><LEFT-ARROW>	Erase all characters previously keyed in, and position cursor to the first input position.

INTERPRET/ABORT

The "ENTER" key is used to signify completion of an user input command. All commands are terminated with <ENTER>.

The "BREAK" key will abort any user command, and return the user to the command prompt ">".

ZEUS SYNTAX

In order for the assembler to translate source code into object code, a specific format is required when entering text. Fields are used to distinguish the various zones in a source text line for the symbolic representation of a user"s program, and user comments.

The LABEL field, optionally contains a label with .a value equivalent to the relative address of, or by, the instruction which follows.

The OPCODE field contains a Z80 opcode, or a ZEUS psuedo-op.

The OPERAND field contains operands. An operand is one or two groups of characters separated by a comma.

The INSTRUCTION field contains a opcode and operand consecution - mnemonic, which will be translated into object code by the assembler.

The COMMENT field, optional, contains user remarks.

SOURCE TEXT LINE:

```
        <--------------------FIELDS-------------------- >
        <--------SYMBOLIC CODE---------- >
                   <----INSTRUCTION----->
LINE# [LABEL] OPCODE [OPERAND(S)]             [;COMMENT]
```

LINE NUMBERS - The line numbers printed, indicate the relative line in the text buffer.

LABELS - Labels are a string of characters with every character and character case significant. To distinguish a label from a numeric constant, the first character in a label must be a @,A thru Z, [, \,], ^, _, ', a thru z, {, |, }, and ~. The balance of the label can also have 0 thru 9, :, =, >, or ?.

COMMENT - A comment begins with a semicolon, and overrides any field.

During assembly printing, two additional zones are added to the left of the line numbers.

```
                      <--------------------FIELDS--------------------->
                      <--------SYMBOLIC CODE---------- >
                                 <----INSTRUCTION----->
ADDR OBJCODE LINE# [LABEL] OPCODE [OPERAND(S)]               [;COMMENT]
```

ADDR - The addr zone contains a hexadecimal number which is either 1) the relative address of the current instruction, 2) the label value for the EQU and DEFL psuedo-ops, 3) the length of the space for the DEFS psuedo-op, or 4) the transfer address for the END psuedo-op.

OBJCODE - The objcode zone contains the hexadecimal object code translated from the mnemonics.

ZEUS SYNTAX

CONSTANT - A constant is a numeric representation of a value. To distinguish a numeric constant from a label, the first character must be numeric or the constant must be enclosed in single quotes. Decimal constants use the numbers 0 thru 9, and optionally are suffixed with the letter D. Hexadecimal constants use the numbers 0 thru 9, the letters A thru F, and are suffixed with the letter H. ASCII constants use the entire ASCII set and are enclosed in single quotes. The single character dollar sign, "$", is the value of the relative address for the current instruction in which it appears.

EXPRESSION - An expression is a group of characters consisting of two or more constants and/or labels separated by an operator. Expressions are evaluated in a left to right order, maintaining an interim result after processing a constant and/or label, ignoring algebraic hierarchy, and not recognizing parentheses as a precedence modifier.

The nine ZEUS operators are:

1. + ADDITION. Adds the value following the operator to the interim result and establishes the sum as the new interim result.

2. - SUBTRACTION. Subtracts the value following the operator from the interim result and establishes the difference as the new interim result.

3. * MULTIPLICATION. Multiplies the value following the operator to the interim result and establishes the product as the new interim result.

4. / DIVISION. Divides the value following the operator into the interim result and establishes the integer portion of the quotient as the new interim result.

5. ! Logical OR. Logical OR between the value following the operator and the interim result establishing a new interim result.

6. # Logical XOR. Logical XOR between the value following the operator and the interim result establishing a new interim result.

7. & Logical AND. Logical AND between the value following the operator and the interim result establishing a new interim result.

8. % MODULO. Divides the value following the operator into the interim result and establishes the integer remainder as the new interim result.

9. < SHIFT. Shifts the interim result the number of bits of the value following the operator establishing a new interim result. If the value following the operator is positive, then the shift is left, otherwise the shift is right.

ZEUS SYNTAX

PSUEDO-OPS

1. COMM - Comment.
2. DEFB or DB-Define byte(s).
3. DEFL or DL-Define label.
4. DEFM or DM-Define string(s) and/or byte(s). (define message)
5. DEFS or DS-Define space.
6. DEFW or DW-Define word(s).
7. END - Terminate assembly.
8. ENIF - Delimit conditional assembly.
9. ERR - Abort assembly.
10. EQU - Set a label to a value. (equate)
11. GET - Include source from disk file.
12. IF - Begin conditional assembly.
13. LIST - Control printing.
14. ORG - Establish relative address. (origin)
15. PAGE - Lprint linefeeds to top of form.
16. SBTL - Establish a subtitle for Lprint.
17. TITL - Establish the title for Lprint.

Psuedo-ops are entered in the instruction field of a source text line.

ZEUS PSUEDO-OP DETAIL

COMM - Generate an object file comment block at the beginning of the file. COMM is primarily used to put a message on the first sector of a object file

NOTE: A COMM instruction is only recognized if no object code has been generated.

Syntax:

```
        00017          COMM 'STRING
```

STRING is a string of characters 1 to 63 characters in length, and must be preceded by a single quote. To create a comment greater than 63 characters, add an additional line with a COMM instruction immediately behind the line with the first COMM instruction. The assembler will append these STRINGs into one comment.

EXAMPLES:

```
00003          COMM 'I     can't  stop  loving  you.
```

Generates the comment "I   can't stop loving you."

```
00008          COMM ****************          My       **     best      *
00009          COMM '* program! ****************
```

Generates a comment block which, when viewed with a "ZAP" utility, would appear as:

```
.N**************
*      My      *
*     Best     *
*   Program!   *
****************
```

The comment block on a disk sector is preceded with two bytes. The first byte will be a X'05', and the second byte will be the length of the comment.

The layout of this comment is as follows:

1. 14 *'S   {adjust for the twobytes preceding comment.}
2. *, 6 Spaces, My, 6 spaces, *
3. *, 5 Spaces, Best, 5 spaces, *
4. *, 3 Spaces, Program!, 3 Spaces, *
5. 16 *'S

DEPB or DB - Generate a byte(s) at the relative address. DEFB is primarily used to create single byte lookup tables.

Syntax:

```
        01333          DEFB BVAL[,BVAL][,BVAL][,...]
```

BVAL is a constant, expression, or label with a value of 0 to 255. For each byte generated, the relative address is incremented by one.

EXAMPLES:

```
00422          DEFB     45

00017          DEFB     46H,63H,73H,74H

00324          DEFB     78,CAT,'U'      {CAT             is definedelsewhere}
```

During assembly printing, only the first byte value is printed.

```
8923 4 E          00324          DEFB 78,CAT,'U'
```

DEFL or DL - Establishes a value for a label which only can be redefined with another DEFL psuedo-op. DEFL is primarily used to perform a complex assembly function. An example of a complex assembly function is shown at the end of ENIF.

Syntax:

```
00389 LABEL DEFL WVAL
```

WVAL is a constant, expression, or label. If WVAL is a label or an expression with a label, and the label has not been established, then LABEL will have value of zero on the first pass. If WVAL is LABEL then LABEL will be established as the value of the relative address.

EXAMPLES:

```
00120 TOPCO   DEFL     78BAH            {TOPCO = 78BAH}
00121 MAKE    DEFL     MAKE             {MAKE = relative address}
00122 KITTEN  DEFL     TOPCO-80H        {KITTEN = 783AH}
00123          DEFB    KITTEN&0FFH      {byte value * 3AH}
00124 KITTEN  DEFL     TOPCO+80H        {KITTEN = 793AH}
00125          DB      KITTEN<-8        {byte value = 79H}
```

DEFM or **DM** - Generate a string(s), and/or byte(s) at the relative address. DEFM is primarily used to insert messages into the object code.

Syntax:

```
00457          DEFM 'STRING *[,BVAL] [,'STRING'] [,...]
```

although DEFM contains DEFB in its repertoire, DEFM can generate object code from a string of ASCII characters enclosed in single quotes.

EXAMPLES:

```
00317           DEFM   'This is a message.'

00237           DEFM   'This is a message with a terminator.',13

01283           DEFM   28,31,'ERROR','1'+80H,10,13
```

During assembly printing, only the first byte value is printed.

```
AB77 1C          01283          DEFM 28,31,'ERROR','!'+80H,10,13
```

DEFS or DS - Increments the relative address by a user defined value. DEFS is primarily used to retain the contents of a block of memory. The special case, DEFS 0, is primarily used to easily mark the end of a table for reference purpses.

Syntax:

```
00017          DEFS WVAL
```

WVAL is a constant; expression with labels, if any, previously defined; or previously defined label with a value between 0 and 65535 inclusive. A DEFS psuedo-op will generate a new loading address for the object file, which when loaded, will skip over WVAL bytes of RAM. Therefore, DEFS preserves WVAL bytes of RAM starting at the relative address.

EXAMPLES:

```
00056          DEFS 256         {this will leave 256 bytes of RAM
                                untouched when the object program is
                                loaded into RAM.}

00330 PICBEG DB        'a','b','c','d','e','f'
00331 PICEND DEFS      0
     ... More source text
00677          LD      B,PICEND-PICBEG {loads B with the length of
                                      the table, 6.}
```

DEFW or DW - Generate a word(s) at the relative address. DEFW is primarily used to create an address table.

Syntax:

```
        00017          DEFW WVAL[,WVAL] [,WVAL] [,...]
```

WVAL is a constant, expression, or label with a value of 0 to 65535 inclusive. For each word generated, the relative address is incremented by two.

EXAMPLES:

```
00188          DEFW TEST          {TEST is defined elsewhere}

00376          DEFW    4488H,65123,3,12
```

During assembly printing, only the first word value is printed.

```
1228  8844      00376          DEFW 4488H,651233,3,12
```

END - Terminate assembly. END is used to direct the assembler to ignore any source text which follows. The END psuedo-op in a GET file will terminate assembly of the GET file and resume assembly from the source text which has the GET instruction.

Syntax:

```
        00017          END      [TRANSFER]
```

TRANSFER is the address which the DOS program execution routine will transfer control (hopefully) over to the user program. This address has also been designated as the program execution point. TRANSFER can be a constant, expression (unusual), or a previously defined label.

EXAMPLES:

```
03122          END                 {transfer address 0000H}

02883          END     402DH        {transfer  address402DH}


00001          ORG     7000H
00002  START LD         A,(HL)      {START = 7000H}
      ... More source text
02893         RET
02894          END     START        {transfer  address7000H}
```

ENIF and IF - Conditional assembly psuedo-ops.
Conditional assembly provides the programmer with an efficient tool to create programs for a variety of conditions. Conditional assembly source text is structured as follows:

```
00017          IF      EXP
          ... Conditional source text
00043          ENIF
```

If EXP is evaluated to a non zero value, then the conditional source text is assembled. If exp is evaluated to a zero value, then the conditional source text is not assembled,

or

```
00017          IF      NOT,EXP
          ... Conditional source text
00043          ENIF
```

If EXP is evaluated to a zero value, then the conditional source text is assembled. If exp is evaluated to a non zero value, then the conditional source text is not assembled.

During an assembly printing, the IF and ENIF text lines, regardless of the condition, are not printed. If the conditional source code is not assembled, then the lines which contain the conditional source text are also not printed.

Typically, the programmer will incorporate at the beginning of the source text, the status for conditional assembly

```
00002 MOD1 DEFL       1
00003 MOD3    DEFL    1-MOD1       {MOD3 = 0}
```

Then have subsequent code test this status

```
00789          IF      MOD1
00790          LD      A,(37ECH)
00791          ENIF
00792          IF      MOD3
00793          IN      A,(0F0H)
00794          ENIF
```

EXAMPLE:

```
00001          ORG     7000H
00002 DOG     DEFL     DOG-$
00003          IF      NOT,DOG
00004          GET     EQU/ASM
00005          ENIF
```

The GET instruction on line 00003 will only be executed on the first pass. The file EQU/ASM must only have EQU'sl.

ERR - Abort assembly and display message. The ERR psuedo-op is usually in a conditional block of source text, set to assemble if an undesirable situation exists. ERR may also be used to quickly get the relative address in a program.

Syntax:

```
        00017           ERR      'STRING
```

If the line containing ERR is assembled, then the relative address is printed followed by STRING. STRING must be preceded by a single quote.

EXAMPLE:

```
00001          ORG      7000H
      ... More source text
01782          RET
01783          ERR      'is the first free byte.
01784          END      START
```

If line 01782 would assemble to relative address 8A21H, then an assembly would produce

8A22 is the first free byte.

during the first pass.

EQU - Explicitly establish the value for a label. The EQU psuedo-op is primarily used to symbolically make references to constants.

Syntax:

```
        00017 LABEL EQU         WVAL
```

WVAL is a constant; expression with labels, if any, previously defined; or previously defined label with a value between 0 and 65535 inclusive.

EXAMPLE:

```
00002 KEYBRD  EQU      2BH
00003 VIDEO   EQU      33H
00004 PRINT   EQU      3BH
00005 LINEIN  EQU      40H
00006 KBWAIT  EQU      49H
      ... More source text
00345 VLINE   LD       A,(HL)
00346         CALL     VIDEO
00347         INC      HL
00348         CP       0DH
00349         JR       NZ,VLINE
00350         RET
```

ZEUS SYNTAX

GET - Include source text from disk file. The GET psuedo-op is used to assemble large amounts of source text, by reading the disk file as source text instead of the resident text buffer.

Syntax:

```
00017           GET FILESPEC
```

If FILESPEC is not found assembly is aborted.

An END instruction is not necessary in a GET file. If an END instruction is found in a GET file or the end of file is encountered, then assembly resumes in the source text which has the GET instruction. 592 Bytes of RAM are dynamically allocated for each concurrent opened GET file. The limit of nesting GET files is dependent on the amount of RAM space available.

During assembly printing, the text line with the GET instruction will not be printed.

LIST - Control assembly printing. The LIST psuedo-op is primarily used during program development to print only pertinent parts of the assembled code.

Syntax:

```
00017           LIST    OFF
```

LIST OFF will disable assembly printing, starting with this line,

or

Syntax:

```
00033           LIST    ON
```

LIST ON will enable assembly printing with the following line, provided the N option is not entered under the A command.

ZEUS SYNTAX

ORG - Establish a new value for relative address. The ORG pseudo-op is used to direct the assembler where the object code will be.

Syntax:

```
        00001         ORG WVAL
```

WVAL is a constant; expression with labels, if any, previously defined; or previously defined label between the values of 0 to 65535 inclusive.

PAGE, SBTL, and TITL - Assembly Lprinting psuedo-ops. These psuedo-ops dress up the assembly Lprinting by placing a title and subtitle on each page. The PAGE psuedo-op generates linefeeds to the top of,the next page.

NOTE: These three psuedo-ops require ZEUS forms, and are not recognized in a GET file.

Syntax:

```
        00023          TITL     'STRINGTITLE
        00024          SBTL     'STRINGSUBTITLE
        00876         PAGE
```

TITL is used to place a title, STRING TITLE, on the first line on each page of an assembly Lprinting. Only the first title is Lprinted, subsequent titles are ignored. The subtitle, SBTL STRING, if exists, is Lprinted on the line after title. New subtitles may be inserted in the source text, but will only appear at the next page break.

EXAMPLE:

```
    00023          SBTL    'Screen formatting routines.
         ... More source text
    00317          SBTL    'The DOG is MAN's best friend.
    00318         PAGE
```

The original subtitle, "Sreen formatting routines.", will be printed immediately after each title, if exists. When the assembler processes lines 00317 and 00318, PAGE will generate linefeeds to the next top of form, then establish the subtitle, "The DOG is MAN's best friend.", as the new subtitle, printing this subtitle on the new page.

During assembly printing, text lines with one of these three psuedo-ops will not be printed.

ZEUS COMMAND DETAIL

ZEUS COMMANDS

A Assemble.

B Calculator.

C Global change.

D Delete line(s).

E Edit line.

F Find label.

G LPrint format.

H LPrint line(s).

I Insert.

J LPrint raw data.

K Kill file.

L Load file.

M Move/Duplicate line(s).

N Reset buffer.

O Object/Operand reference

P Print line(s).

Q Exit.

R Reference.

S Save file.

T Print label table.

U Usage.

X Recovery on reentry.

Print, as used in the following text, means output to the device in the video DCB (401DH).

Lprint, as used in the following text, means output to the device in the printer DCB (4025H).

ZEUS COMMAND DETAIL

Assemble source code.

Syntax A[E][H][N][0][Q][T[nn]]<ENTER>

E - Wait on error. The E option will call assembly to be suspended whenever an error occurs. After the error message and the offending line is printed, assembly can be resumed by pressing either <SPACE> or <ENTER>.

H - LPrint link to Print. The H option will link the assembly printing to the lineprinter.

N - Printing off (does not suppress label table). The N option will inhibit assembly printing for non-erroneous lines, and overrides the LIST ON instruction.

0 - Generate object deck. The 0 option directs the assembler to generate loading instructions as well as object code to a user selected filespec. If the 0 option is entered, then the prompt

Filespec:

will be printed. The assembler will automatically append /CMD to the filename if none was entered, then search for the filespec.
If the filespec is found, then the query

Overwrite file?

will be printed. Press <Y> to overwrite file, <N> to return to the filespec prompt, or <BREAK> to abort assembly.
If the filespec is not found, then the query

Create file?

will be printed. Press <Y> to create a new file, <N> to return to the filespec prompt, or <BREAK> to abort assembly.

Q - Quit after first pass. The Q option will terminate assembly if an error is detected during the first pass. After the assembler makes one pass (builds label table), the assembler will not make this first pass again unless Q is entered as an assembly option, a disk I/O error occurs, or one of the following commands is invoked.

C, D, E, I, L, M, N, or Q.

The psuedo ops IF, DEFS, DS, EQU, ORG, and END will generate first pass and second pass errors if the corresponding expressions cannot be evaluated (undefined label, or expression error). The totalerrors will include the first pass as well as the second pass errors, even if the same error is indicated on both passes. If the A command is executed again, and the first pass is not processed, then the total errors will be the results of the second pass only. A repeated label in the label field, redefinition error, will result in first pass errors only.

ZEUS COMMAND DETAIL

T - Print label table. The T option will sort the labels in ascending alphanumeric order, then print the labels and their corresponding hexadecimal values up to nn characters wide.

NOTE: Options may be 1) in any order, 2) repeated, 3) separated by one or more spaces, 4) separated by one or more commas, 5) separated by any combination of spaces and commas, or 6) may not be separated at all.

EXAMPLE:

ANO {is equivalent to.}
{1} AON {2} A,O,N {3} A,N,O {4} A O N
{5} A ,O ,N {6} A, ,O,,,,N,O,O,N N N etc.

EXAMPLE:

A,N,H<ENTER> {print and Lprint error(s) and error line (s), if any.}

EXAMPLE:

AT80H<ENTER> {print and Lprint source code and label table using 80 columns for table.}

EXAMPLE:

AQE<ENTER> {pause on error line(s), if any, and terminate assembly after the first pass if any errors occured.}

EXAMPLE:

A N<ENTER> {print error(s) and error line(s), if any, then print the total errors for the processed source code.}

The assembler operates in three phases. During the first pass/phase the assembler creates the label table, and establishes the values for DEFS, DS, END, EQU, and ORG. If there are no first pass errors, or Q option not entered, the assembler sets a first phase complete status byte and continues with the second pass/phase. During the second pass/phase the assembler will output in accordance to the selected options, then print the total errors. After the second phase is complete, the assembler sets a second phase complete status byte. Additional A commands with the second phase status byte set, will proceed with phase three/second pass. Phase three is instant assembly on all processed source code in the text buffer.

This characteristic is unique with **ZEUS**, and encourages the user to assemble with "AN", to check for assembly errors, then proceed with "ANO" for the generation of an object deck, if there are no ertors. The maximum speed is obtained if the user has maximized the amount of code in the resident text buffer.

ZEUS COMMAND DETAIL

B - Command mode calculator.

Syntax Bexpl[operator exp2][operator exp3][...]<ENTER>

Operator can be:
+ ADD.
- SUBTRACT (also unary)•
* MULTIPLY.
/ DIVIDE.
! OR.
XOR.
& AND.
% MODULO.
< SHIFT LEFT.
<- SHIFT RIGHT.

The B command will calculate expressions in the range of 0 to 65535 decimal and/or 0000H to 0FFFFH hexadecimal, returning an integer result in the range of 00000 to 65535 decimal and 0000H to FFFFH hexadecimal.

EXAMPLES: {The <ENTER> is left out for clarity.)

B 38H	B 99	B 8C9EH
00056 0038H	00099 0063H	35998 8C9EH
B 56+5	B 56H+5	B 0A0H+23H
00061 003DH	00091 005BH	00195 00C3H
B 37-3	B 500-45H	B 0FFH-255
00034 0022H	00431 01AFH	00000 0000H
B 67D*33	B 0AH*0AH	B 233*56H
02211 08A3H	00100 0064H	20038 4E46H
B 78/9	B 0FE9H/23H	B 3C00H/4
00008 0008H	00116 0074H	03840 0F00H
B 5!25	B 37H123	B 0F132H!89DDH
00029 001DH	00055 0037H	63999 F9FFH
B 0F23H#1A44H	B 89#19	B 99D#78H
05479 1567H	00074 004AH	00027 001BH
B 5&-5	B 23&5/3	B 89!8&444/2
00001 0001H	00001 0001H	00012 000CH
B 89%7+23*6	B 89H%7	B 12345%2
00168 00A8H	00004 0004H	00001 0001H
B 89<1	B 0FFFFH<8	B 0FFFFH<16
00178 00B2H	65280 FF00H	00000 0000H
B 89<-l	B 0FFFFH<-8	B 566H/23<-3
00044 002CH	00255 00FFH	00007 0007H

If the assembler has processed phase one, and the label table is intact, then the labels created by the assembler can be used as expressions.

EXAMPLES:

```
B WORD-BYTE                  {from the file Z80E/ASM).
01548 060CH

B OFFSET*3
00090 005AH

B DISPLACEMENT<8/BYTE+WORD
01797 0705H
```

Calculations are performed on a left to right basis. If an expression is missing or an operator is used improperly, then

Expression?

is printed, and the B command terminates. If a label is used as an expression, and this label is not found in the label table, then

Undefined labell

is printed, and the B command terminates.

EXAMPLES:

```
B 75-33*
Expression?

B UNCLE+3
Undefined label!
```

If the resulting calculation exceeds 65535 decimal, FFFFH hexadecimal, then the results will be a modulo 65536 answer.

EXAMPLES:

```
B300*512       {300*512=153600. 153600%65536=22528.}
22528 5800H

B65200+500
00164 00A4H
```

ZEUS COMMAND DETAIL

C - Global change.

Syntax C[OPT]STR1,[STR2][,RELNUM1][,RELNUM2]<ENTER>

The C command will change STR1 to STR2 from RELNUM1 to RELNUM2 inclusive. Single quoted text and comments are untouched, unless OPT is used as specified below. IF STR2 is null, the STR1 will be deleted.

OPT can either be a single quote, or a semicolon. If OPT is a single quote, then changes will only take place in single quoted text. If OPT is a semicolon, then changes will only take place in comment text.

If a change in a given line would result in the line exceeding 127 characters in length, then the message

Out of Memory!

will be printed and the C command aborted without changing the line.

The C command will print the line before the change occurs and again after the change has successfully taken place. The <SPACE> may be used to pause the C command, and the <BREAK> key to terminate the C command.

If RELNUM1 is not specified, then line one is used. If RELNUM2 is not specified and RELNUM1 is specified, then RELNUM2 will default to RELNUM1 (single line change). If neither is specified, then the changes will take place over the entire text.

EXAMPLES:

C'aborted,terminated	{change "aborted" to "terminated", in single quoted text, throughout the text buffer}
C DOG,,4,37	{delete "DOG", from line 4 to 37 inclusive}
C CAT,	{delete "CAT" throughout the text buffer}
C MAN,BOY,.,*	{change "MAN* to "BOY", from the current line to the end of the text buffer}
C ABC , DEF , LOG ,<PERK	{change "ABC" to "DEF", from the line which has the label "LOG" in the label field, to one line less than the line which has "PERK" in the label field}

NOTE: The C command processes each changed line through the syntax check routine. If a change is made which would result in an erroneous line, the E command will be invoked permitting corrections to the offending line. Although the C command continues after the line is deemed acceptable, the outcome may not be the desired results. The C command can be terminated immediately after the "force" edit, by pressing the <ENTER> and <BREAK> keys simultaneously. The C command can be directed to bypass the syntax checking by using an explanation mark for OPT. (User beware!)

ZEUS COMMAND DETAIL

D - Delete specific line(s) from the text buffer.

```
Syntax      DRELNUM1,[RELNUM2]<ENTER>
```

The D command will remove lines from RELNUM1 to RELNUM2 inclusive. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUM1.

EXAMPLE: {current line * 37}

```
D,.             {delete line 37}
D               {delete line 37}
D,              {delete line 37}
D.              {delete line 37}
D.,             {delete line 37}
D37             {delete line 37}
D#,.            {delete lines 1 to 37 inclusive}
D.,*            {delete lines 37 to the end of the text
                buffer}

D43,.                       {results in}
.  <— parameter error {RELNUM2 cannot be less than
                RELNUM1}
```

EXAMPLE:

```
D 43,22         {results in}
22 <— parameter error
```

EXAMPLE:

```
00075          RET
00076  LOOP    LD      A,{HL)
00077          INC     HL
00078          OR      A
00079          RET     Z
00080          DJNZ    LOOP
00081  DOG     LD      HL,MOKE
00082          CALL    BOTTLE
```

Lines 76 to 80 may be deleted by one of the following.

```
1) D76,80
2) D76,<81
3) D76,<DOG
4) D<77,80
5) D<77,<81
6) D<77,<DOG
7) DLOOP,80
8) DLOOP,<81
9)     DLOOP,<DOG
```

E - Edit specific text line.

Syntax ERELNUM<ENTER>

The E command will permit the user to modify the contents of a text line. If RELNUM is not specified, then the current line is used. When the E command is invoked, the line will be printed with a small transparent rectangle cursor at the first position, and a block at the end of the line. Concurrently the <DOWN-ARROW> and <CLEAR> keys become function keys, and the user will be in the overstrike mode. In the overstrike mode, any non-control key will replace the character at the cursor position.

In the insert mode, indicated by the small rectangle cursor becoming a large rectangle cursor, any non-control character will be inserted at the cursor position, and the balance of the line moved right one position. The insert mode will drop the 128th character if the number of characters in a line would exceed 127.

Control keys which do not overstrike or insert.

Edit functions:

The edit functions are implemented by pressing one of the function keys (<CLEAR> or <DOWN-ARROW>) with one appendage, and, while keeping this key held down, press one of the four keys listed below:

A	abort changes and start editing again,
D	delete character at cursor.
H	delete all characters from cursor to end of 1 ine.
I	shift between overstrike and insert.

Cursor moves:

<RIGHT-ARROW>	move right 1 character position (to tab use <SHIFT><SPACE>).
<LEFT-ARROW>	move left 1 character position.
<SHIFT><RIGHT-ARROW>	move to end of line.
<SHIFT><LEFT-ARROW>	move to beginning of line.

Other control keys:

<BREAK>	abort EDIT and leave the line without any changes.
<ENTER>	process the modified line through the syntax check routine. If no error is found, replace the edited line with the modified line, and exit the E command. If an error is found, then the E command will disable <BREAK>, place the edit function in the overstrike mode, and position the cursor at the first character in the potentially erroneous symbolic field.

F - Print line with the specified label.

Syntax FLABEL<ENTER>

The F command will search from line one for LABEL in the label fields, and print the line if found. The LABEL is mandatory with the F command. If LABEL is null, then

Label not specified!

will be printed. If the LABEL does not exist in the label field, then

Label or line not found,

will be printed.

G - Set printer values.

Syntax GL,W,P<ENTER>
or
Syntax G<ENTER>

The G command sets up the parallel lineprinter characteristics for ZEUS forms. The parameters are: L = the number of printed text lines per page, W = the maximum number of characters printed before ZEUS forms will send a carriage return, and P = the number of physical lines per page.

G <ENTER> will print the current forms settings.

EXAMPLES:

G 50,80,52<ENTER> {Lprint 50 text lines, of 80 characters maximum on a 52 line page. (11" for QUICK PRINTER)}

G 80,132,88<ENTER> {Lprint 80 text lines, of 132 characters maximum on a 88 line page.}

G 60,80,6<ENTER> {Lprint 60 text lines, of 80 characters maximum on a 66 (see NOTE!) line page, (normal default for most parallel printers)}
NOTE: If P is less than L, then P is added to L.

H - LPrint line(s).

Syntax HRELNUM1,RELNUM2<ENTER>

The H command will print and Lprint RELNUM1 to RELNUM2 inclusive. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUMl. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUMl.

The **H** commands **links** the lineprinter to the video monitor.

I - Insert lines.

Syntax IRELNUM<ENTER>

The I command is used to insert or add source code into the text buffer immediately behind RELNUM. If RELNUM is not specified then the current line is used. IO<ENTER> is used to insert a line ahead of the text buffer (new line one).

The I command will print, a five digit line number one unit higher than RELNUM followed by a space, and wait for the user to key in text. After a text line is keyed in and terminated with <ENTER>, the line is processed through the syntax check routine. If no error is found, then the line is inserted into the text buffer. If an error is found, the I command is suspended, <BREAK> disabled, the E command invoked, with the edit function in the overstrike mode, and the cursor positioned at the first character in the potentially erroneous symbolic field. When subsequent modifications correct the erroneous line, then the I command will resume.

The I command continues to print a line number one unit higher than the previously entered line and await user input until <BREAK>, which will return the user to the command mode.

ZEUS COMMAND DETAIL

J - Raw data to Lprint.

Syntax J<ENTER> {Software top of form.}
Syntax Jdata<ENTER> {will send data **directly** to printer.}

EXAMPLES:

J27,69 J 1BH,45H J 27,'E' J 1BH,'E'

J 'Z','e','u','s',13<ENTER> {will Lprint "Zeus".}

J <ENTER> {will determine how many software lines remain to top of form, and generate linefeeds to match.}

K - Kill filespec.

Syntax Kfilespec<ENTER> {auto "/ASM".}

L - Load source file.

Syntax L[OPTl][filename][0PT2]<ENTER> {auto "/ASM".}

If 0PT1 =#, then load a ASCII format file.
If 0PT1 =*, then load a EDTASM format file.
If 0PT1 =%, then load a EDTASM format file without header.

0PT2 is used in conjunction with 0PT1 to tell ZEUS to skip over OPT2 number of lines before loading source code into the text buffer.

If text already in buffer, then

Append to buffer?

will be printed. Respond <Y>, <N>, or <BREAK>. If the filename was not entered in the command prompt, then

Filespec:

will be printed. Enter filespec or <BREAK>.

EXAMPLE:

L*SUPER 3000<ENTER>

This command directs ZEUS to load the EDTASM format file SUPER/ASM, starting with the 3001th source text line.

M - Move/duplicate text lines.

Syntax M[OPT]RELNUM1,RELNUM2,RELNUM3<ENTER>

The M command will move RELNUMl to RELNUM2 inclusive behind RELNUM3. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUM1. RELNUM3 must be greater than RELNUM2 or less than RELNUMl. If RELNUM3 is not specified then RELNUM3 defaults to RELNUM2 (error).

If OPT = 1, then the lines in the range of RELNUM1 to RELNUM2 inclusive are duplicated behind RELNUM3.

EXAMPLE:

M!,,< {duplicates current line at current line.}

N - New text buffer.

Syntax N<CLEAR><ENTER>

Resets all pointers. {Recoverable via "X"}

O - Opcode/operand reference.

Syntax 0[OPT]STR1<ENTER>

The O command will search from line one for the first occurrence of STR1. STR1 can be any combination of labels, tabs, opcodes, or operands.

If OPT = #, then the O command will print all lines with STR1.

EXAMPLE:

O LD{TAB}A,(HL)<ENTER> {locates LD A,(HL).}

If STR1 is located, then subsequent 0<ENTER>'s will find the next occurrence of STR1 until

Text end.

is printed, at which time the reference search will restart at line one.

If STR1 is located, then 0#<ENTER> will print all remaining line numbers with STR1.

P - Print line(s)

Syntax PRELNUM1,RELNUM2<ENTER>

The P command will print text lines from RELNUM1 to RELNUM2 inclusive to the video monitor. If RELNUM1 is not specified, then the current line is used. If RELNUM2 is not specified, then RELNUM2 defaults to RELNUM1. If RELNUM2 is specified, then RELNUM2 must be equal to or greater than RELNUMl.

Q - Exit **ZEUS**.

Syntax Q<CLEAR><ENTER>

R - REFERENCE

Syntax R[OPT]STR1<ENTER>

The R command will search from line one for the first occurrence of STR1. STR1 can be ASCII text and not opcodes and/or operands. The 0 command is used to find opcodes and/or operands.

If OPT = #, then the R command will print all lines with STR1.

EXAMPLE:

R DOGGY<ENTER> {locates DOGGY.}

If STR1 is located, then subsequent R<ENTER>'s will find the next occurrence of STR1 until

Text end.

is printed, at which time the reference search will restart at line one.

If STR1 is located, then R#<ENTER> will print all **remaining** line numbers with STR1.

S - Save filespec.

Syntax S[OPTl][filename]<ENTER> {auto "/ASM"}

If OPT1 =#, then save is in ASCII format.
If 0PT1 =*, then save is in EDTASM format.
If OPT1 =%, then save in EDTASM format without header.

T - Sorted label table printing.

Syntax T[nn]<ENTER>
nn = 1 to 252

The T command will sort the label table in ascending alphanumeric order, then print its contents in the format:

LABEL 0000 LABEL 0000 LABEL 0000 LABEL 0000 LABEL 0000

nn is the width of device to receive the label table. If nn = 0 or nn is null, then the device width will default to 64.

Once an assemly has gone through the first phase, the label table is completed. However, a disk I/O error, or the completion of one of the following commands C, D, E, I, L, M, N,or Q will reset the label table.

U - Memory usage.
Syntax U<ENTER>

The U command will print the memory used by the resident source text; and, if the assembler has completed phase one, the memory used by the label table, and GET files. In addition, the amount of free memory is printed.

EXAMPLE: {File Z80E/ASM with ZEUS forms.}

```
U<ENTER>

09466 source       {source text usage.}
00045 table        {memory for lables in label table.}
00000 GET          {memory GET files used.}
25119 free         {unused.}
```

X - Recover source text on reentry to ZEUS.

Syntax <X>

If "X" is the first key pressed upon entry to ZEUS, then

nnnnn text lines.

will be printed, with nnnnn being the number of recovered lines.

If the "X" key, inadvertenly, was not the first key pressed, exit and try again.

ZEUS

ZEUS is an small disk operating system, with only one utility. This utility, $/CMD (for file copying), is customized to work with *ZEUS* onlyl

Although *ZEUS* is supplied single density diskette, *ZEUS* will communicate with double density media in a machine equipped with a double density disk controller. *ZEUS* has automatic hardware recognition, as well as automatic density recognition.

ZEUS has five single keystroke commands.

0 = DIR :0
1 = DIR :1
2 = DIR :2
3 = DIR :3
<ENTER> = repeat last DOS input.

These single keystroke commands only work if they are the first key pressed after receiving the "0K" prompt, or after <BREAK> is pressed.

LIBRARY COMMANDS

ZEUS has 24 library commands. These commands are listed below with a brief description. All commands are terminated with <ENTER>.

APPEND filespec1 to filespec2 - add filespec1 to the end of filespec2.

AUTO doscmd - automatic, invincible, operation of a command or executable file on powerup/reboot.

BOOT - reboot system.

CLEAR - zero memory from 5200H to TOPMEM.

CLOCK (ON) or CLOCK (OFF) - enable/disable time printing in the upper right-hand corner of the video.

CONFIG[[:][drn] (param[,param])] - set default power-up drive attributes.

Param may be:

STEP=nn	where nn is the stepping speed of drive drn.
DENSITY=q	where q is the default density of drive drn.
SIDES=ss	where ss is the number of sides on drive drn.

CONFIG<ENTER> echo's system disk's configuration.
CONFIG (7)<ENTER> echo's RAM's current configuration.
CONFIG (X)<ENTER> writes RAM's configuration onto the system diskette and removes all SKIP's.

DATE[mm/dd/yy] - set date to mm/dd/yy. DATE<ENTER> prints time and

DIR[[:]d][(opt][,...])] - print directory of diskette in drive d. All visible files are displayed.

°Pt may be: A, I, K, P, or S.
A - print directory with file attributes.
I - include invisible files in directory printing.
K - include killed files in directory printing.
P - link printeer to printer.
S - include system files in directory printing.

DUMP filespec (START=X'ssss',END=X'eeee'[,TRA=X'tttt'][,CIM][,TITLE]) Transfer the contents of memory from ssss to eeee inclusive to disk. This command will insert the extension "/CMD" if none is specified, ssss, eeee and tttt are 4 digit hexadecimal addresses, eeee must be greater than ssss. tttt is optional, and specifies the entry point for execution of the file. CIM specifies directs transfer to diskette sectors without 'load marks'. TITLE will place a title block at the beginning of the file's contents.

FORMS[(param][,param1][,param2)] - set printout parameters.

Where param:

I intialize line counter and character counter to zero.
W=www www=l to 254 (the maximum width of print line in characters). If www = 255 infinite width.
p=ppp PPP=1 to 255 (the page length in print lines).
T=ttt ttt=l to 255 (the printed TEXT in print lines).
S=sss sss=l to 127 (the blank spaces between blocks of printed TEXT).
X Writes the current W, P, and T values to the system diskette for default parameters on subsequent power-ups.
N=nn nn = nulls after linefeed.
L send linefeed after carriage return.
C directs printer output to RS-232-C (Model III only).

FREE - print the number of available file spaces, free granules, and kilobytes of disk space on all mounted diskettes.

KILL filespec - remove filespec form the active roll.

LIB - print *ZEUS* library commands.

LIST filespec - print filespec to the video monitor.

LOAD filespec - load filespec from diskette into RAM and return to OK prompt. The file is not executed.

PATCH filespec (REC=nn, [BYTE=yy]) b1[;b2][;b3][...] - modify the contents of a diskette file, nn specifies the physical record in the filespec. The first record is 0. yy is the relative byte in the physical record nn. yy can be 0 to 255 decimal (0 to FF hexadecimal). The physical record, nn, will automatically advance if b2 up causes relative byte, yy, to exceed 255 decimal.

PRINT filespec - lprint the filespec to the printer.

RENAME #filespecl [to] filespec2 - change the name of filespecl to filespec2. If '#' is specified, then the renamed file's date will be changed to the current RAM date. Filespec2 will retain the protection level, password, and directory attributes of filespecl. Filespec2 cannot already exist on the diskette.

RESTOR filespec:d - recover a file that was inadvertantly killed. The drive number is mandatory.

SETCOM[([BAUD=r][,s][,W0RD=1][,STOP=b][,p][,t][,DTR][,RTS][,w])]
Set RS-232-C parameters. (MODEL III only)

r = baud rate. s * ODD or EVEN parity.
1 = word length. b = stop bits.
p = PD or PE - parity disabled or parityenabled.
t = TD or TE - transmit disabled or transmit enabled.
DTR = DTR set DTR high, otherwise low.
RTS = RTS set RTS high, otherwise liw.
w = WAIT or NOWAIT.

SETDIR[[:]d] (type) - alter the directory cylinder Data Address Marks on a single density diskette. (MODEL I only)

type = 0 for MODEL I
type = N for MODDEL III

SKIP[[:]d] - read a 40 track diskette in 80 track disk drive "d". If "d" is not specified remove all SKIPS.

TIME[hh:mm:ss] - set time to hh:mm:ss. TIME<ENTER> prints time and date currently stored in RAM.

TOPMEM ddddd, TOPMEM Hnnnn, or TOPMEM nnnnH - set TOPMEM.

ddddd = a decimal number from 28671 to 65535 inclusive, nnnn = a hexadecimal number from 6FFF to FFFF inclusive. To display TOPMEM, key TOPMEM <ENTER>•

VERIFY (ON) or VERIFY (OFF) - reread sectors for parity errors.

$/CMD - Duplicate a single file.

This utility will copy a single file from one diskette to another.

If the filespec for the destination diskette is the same as the source diskette, then the filename does not have to be entered twice.

    $ TEST/ASM:2 1

If the destination diskette's drivespec is the same as the first, then the user will be prompted to swap diskettes.

    $ TOKEN/ASM:2 TO PACK/ASM:2

Whenever drive zero is specified, a "$" must precede the filespec, and the user will be double prompted to swap diskettes.

$ $ZEUS:0 0

The user will receive the message:

Press "ENTER" when a ZEUS system diskette is in drive 0.

or

Insert *ZEUS* <ENTER>•

This message is printed even .if the *ZEUS* diskete is in drive zero.

Hardware Spezial

> 256K RAM fuer Z80

> 8255-Parallelport

> AmigaSidecar

> Der GIIIs die 4 unbekannten Wesen

> Doubler

> EG64MBA

> Ersetzen des Z80 durch den HD64180

> EXP1 192K

> FCSwitch

> GDP-64 Grafikkarte auslesen

> Gepufferter ECB-Bus fuer Genie1 und 2

> HD64180 als Coprozessor im Z80-System

> Hercules Genie

> Komtek Zeichensatz

> Konzept Terminal

> RAM bei 3000h-37DFh

> Sicheres Plaetzchen

> SpeedUp

> Ver-16-fachung des Sonder-ROM

> Volldecodierter Sonder-ROM

> Z180-GIIIs

256K RAM für Z80-Systeme

Helmut Bernhardt

"64K-Byte RAM sind die obere Grenze für den vom Z80 adressierbaren Speicher". Diese Behauptung ist nur dann richtig, wenn die direkte Adressierung gemeint ist. Durch Banking d.h. Umschalten eines Teilbereichs des 64K- Adreßraumes auf parallele Speicher läßt sich der vom Z80 nutzbare Speicher beliebig ausbauen; 1M-Byte-RAM- Karten werden heute für fast alle Z80-Systeme angeboten.

Gegenüber solchen Karten hat die im folgenden beschriebene Speichererweiterung den Vorteil, daß sie für wenig Geld zu realisieren ist und weder an einen bestimmten Busstandard noch gar einen speziellen Computer gebunden ist. Einzige Voraussetzung ist ein Z80-Prozessor.

Der Nachteil des Projektes besteht darin, daß man sich auf dem CPU-Board ein bißchen auskennen muß und daß es ohne Lötkolben doch nicht geht.

Das Grundprinzip der Speichererweiterung ist ein Ersetzen der vorhandenen 4164-RAMs (oder auch 4116-RAMs) durch die pinkompatiblen 41256-Typen und eine mit wenigen Standard-ICs aufgebaute Banking-Logik, mit der ein gezielter Zugriff auf die vielen K-Byte möglich ist.

Die Banking-Logik

Vom Z80 aus gesehen ist der direkt adressierbare Speicherraum von 64K-Byte in zwei Blocks von 32K-Byte Größe unterteilt. Die obere Hälfte (Adressen 8000H-FFFFH) ist dabei nicht umschaltbar und deshalb vom Z80 immer erreichbar (Common-Bereich). Interrupt-Service-Routinen, der Stack und alle Routinen, auf die ständig zugegriffen werden muß, müssen hier liegen.

Die untere Hälfte ist in mehreren parallelen 32K-Blocks vorhanden. Welcher dieser Blocks im Adreßraum 0000H- 7FFFH für den Z80 jeweils verfügbar sein soll, läßt sich durch Ausgabe der Nummer des Blocks (Bank Nr.) an einen Port (Latch) einstellen. Der Block ist dann solange als Speicher verfügbar, bis eine andere Bank Nr. ausgegeben wird. Durch RESET* wird beim Einschalten des Computers automatisch immer die Bank 0 eingestellt, um bei der Systeminitialisierung definierte Verhältnisse vorzugeben.

Von der Aufteilung der 256K (512K, 1024K) RAM her gesehen sind die Verhältnisse etwas anders zu sehen. Die unteren 64K sind nach dem Einschalten des Computers diejenigen, auf die der Z80 zugreifen kann. Alle anderen 32K-Blocks können anstelle des ursprünglichen 32K-Blocks in die untere Hälfte des Adreßraumes des Z80 gelegt werden, wenn eine der Bank Nummern 1-7 (15, 31) ausgegeben wird.

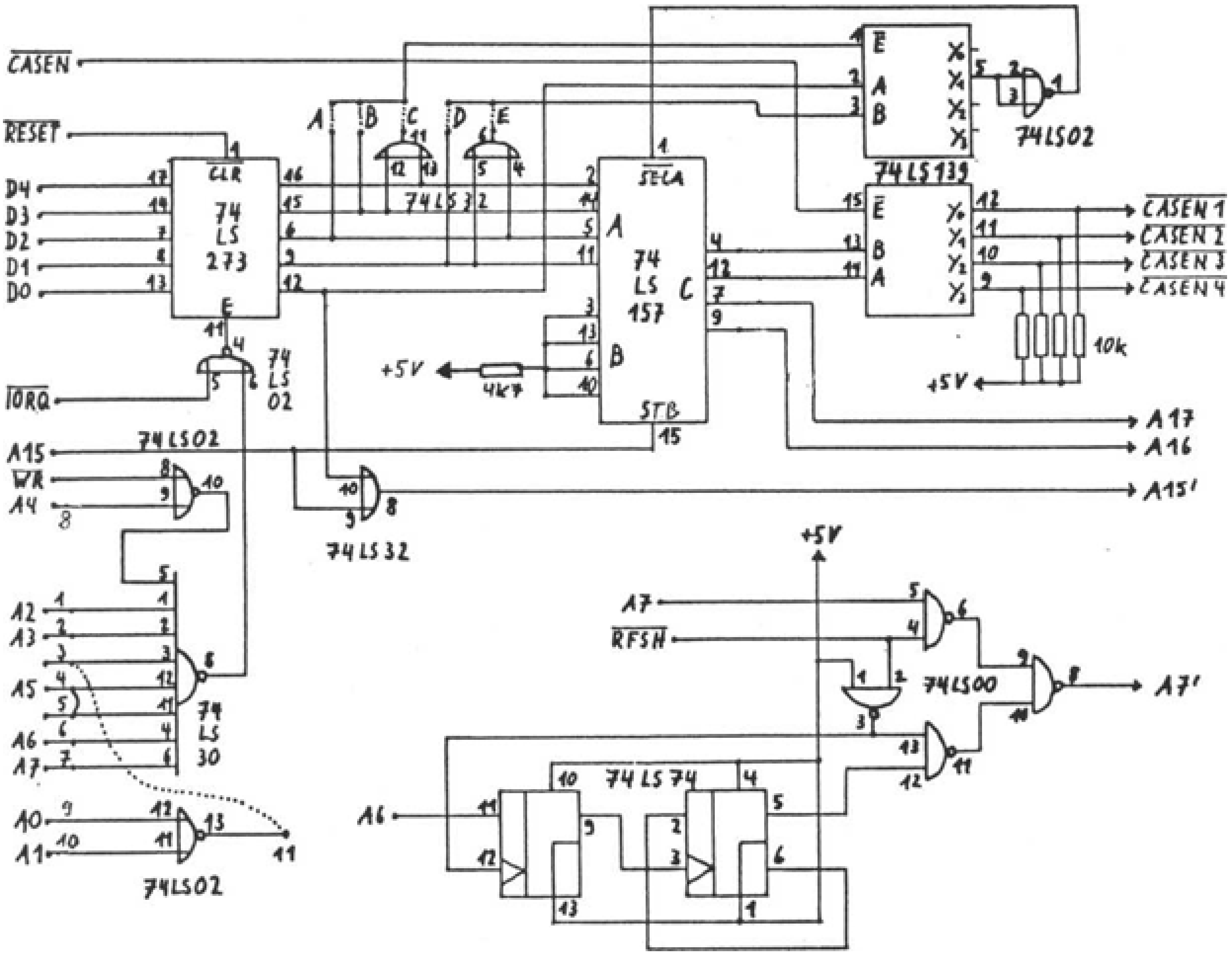
Das wird dadurch erreicht, daß durch die Latchausgänge Q1 und Q2 die Adressen A16 und A17 vorgegeben werden, die einen der vier 64K-Blocks innerhalb eines 256K-Blocks anwählen, Q3 und Q4 codieren einen der bis zu vier 256K-Blocks und Q0 und A15 des Systembus werden dafür herangezogen, die Adresse A15' für den Speicher zu liefern, um dort die untere oder obere Hälfte eines 64K-Blocks zu adressieren,

Die Herleitung von A15' über OR-Verknüpfung von A15 und Q0 sorgt dafür, daß immer dann, wenn A15 high ist, die obere Hälfte des 64K-Blocks angewählt wird (Common-Bereich), während dann, wenn A15 low ist, der Pegel des Latchausgangs Q0 den Pegel von A15' bestimmt und so zur richtigen Anwahl der eingestellten Bank beiträgt.

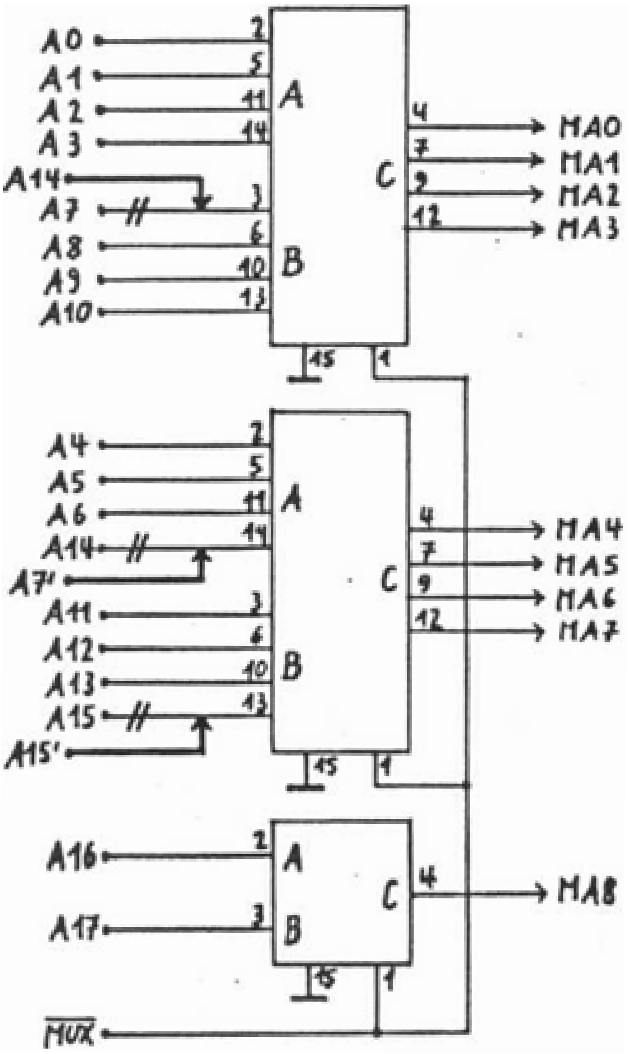
Da nun aber bei einer Adresse im Common-Bereich (A15=1) nicht nur die obere Hälfte irgendeines durch Q1 bis Q4 codierten 64K-Blocks im RAM sondern ausschließlich die obere Hälfte des unteren 64K-Blocks angewählt werden soll, wird durch den Pegel von A15 am STB*-Eingang des 74LS157-Multiplexers (Abb.1) Low-Pegel ausgegeben.

Der Rest der Schaltung erfüllt die Aufgabe, zu erkennen, ob Bank Nr.1, die den gleichen 32K-Block im RAM wie der Common-Bereich belegen würde, eingestellt ist, und daraufhin die höchste verfügbare Bank zu adressieren. Das geschieht dadurch, daß bei eingestellter Bank Nr.1 das Bitmuster an den Latchausgängen zu einem High-Pegel am Select-Eingang des Multiplexers führt, der dann die Pegel der B-Eingänge, die über 4k7 an +5V gelegt sind, durchschaltet. Diese Sicherung des Common-Bereichs führt dazu, daß unter den Bank-Nummern 1 und 7 (15, 31) der gleiche 32K-Block adressiert wird. Die Bank 7 (bei 256K), 15 (bei 512K) bzw. 31 (bei 1024K) sollte also nicht benutzt werden.

1) Schaltung des Banking-Boards und Generierung einer 8-Bit Refresh- Adresse



2) Änderung und Erweiterung des Adreßmultiplexers für die RAM- Adressen am hypothetischen Beispiel eines Computers, der schon von 4116- RAMs auf 4164-RAMs umgerüstet worden ist. Die Belegung der A-Eingänge mit den niederen Adressen ist nicht zwingend (siehe mc-CP/M-Computer). Von einem zusätzlich nötigen 74LS157 wird nur 1/4 benötigt.



Mehr als 256K RAM lassen sich dann realisieren, wenn im Computer mehrere Reihen mit 8 Sockeln für RAMs vorhanden sind, oder indem die RAM-ICs übereinander gelötet werden, wobei dann Pin 15 (CAS*) des aufgelöteten ICs hochgebogen werden muß. Durch die Drahtbrücken A-E muß die Schaltung dem Speicherausbau angepaßt werden. Je nach Speichergröße sind folgende Drahtbrücken zu legen:

256K	1	Reihe RAMs	Brücken	A	und	D
512K	2	Reihen RAMs	"	B	"	E
1024K	4	Reihen RAMs	"	C	"	E

Diese Drahtbrücken sorgen dafür, daß je nach Speicherausbau der durch Bank Nr.1 angewählte 32K-Block identisch ist mit dem durch die höchste mögliche Bank-Nummer freizugebenden Block.

Bei mehr als 256K RAM müssen die einzelnen 256K-Blocks durch die Signale CASEN1* bis CASEN4* (bzw. deren Verknüpfung mit CAS* über OR-Gatter) freigegeben werden. Bei nur 256K RAM braucht die CAS*-Freigabe, wie sie vorher für 64K vorgelegen hat, nicht geändert zu werden.

Wenn 512K RAM realisiert worden sind, muß auf dem Banking-Board die Leiterbahn zu Pin13 des 74LS139 durchtrennt werden, an diesen Pin ist statt dessen GND zu legen, um zu verhindern, daß bei Ausgabe der Bank-Nr.1 nicht die nicht vorhandene Bank31 anstelle der Bank15 eingestellt wird.

Das Refreshing

Ein weiteres Problem stellt neben der Adressierung innerhalb der 41256-RAMs noch das Refreshing dar. Gegenüber den meisten 4164-Typen mit 128 Refresh-Zyklen (nur einige Ausnahmen) benötigen 41256-RAMs grundsätzlich 256 Refresh-Zyklen. Der Z80 liefert bei seinem Hidden Refresh aber nur eine 7-Bit-Refresh-Adresse.

Dieses Problem ist aber von ***Michael Hungershausen (c't 9/84, S.22***) schon gelöst worden und soll deshalb hier nicht weiter abgehandelt werden. Der Vollständigkeit halber ist in **Abb.1** eine der beiden vorgestellten Varianten für die Herleitung einer 8-Bit-Refresh-Afresse wiedergegeben.

Der Adreßmultiplexer

Ebenfalls in Verbindung mit dem Refreshing ist folgendes Problem zu sehen. Durch den Zyklus von 128 Refresh- Adressen ist es nötig, daß beim Adreß-Multiplex die Adressen A0 bis A6 zusammen durchgeschaltet werden. Anstelle von A7 kann aber auch irgendeine höhere Adresse zusammen mit A0-A6 durchgeschaltet werden. Bei Computern, in denen ursprünglich 4116-RAMs verwendet wurden, und die dann spater auf 4164-RAMs umgerüstet wurden, muß sogar davon ausgegangen werden, daß dies der Fall ist, da für die interne Adressierung von 4116- RAMs die Adressen A0-A6 und A7-A13 abwechselnd durchgeschaltet werden.

Es muß bei den Adreßmultiplexern also ermittelt werden, ob A7 mit A0-A6 zusammen durchgeschaltet wird. Wenn dies nicht der Fall ist, muß ermittelt werden, wo A7 anliegt und wo eine höhere Adresse mit A0-A6 zusammen durchgeschaltet wird. Beide Signale sind von den Pins der Multiplexer abzutrennen; an den Pin, wo vorher die höhere Adresse anlag, ist A7' (Abb.1) zu legen und die dort abgetrennte Adreßleitung ist dahin zu führen, wo vorher A7 anlag.

Außerdem muß durch ein huckepack-aufgelötetes 74LS157 (auf einen der schon vorhandenen Multiplexer; bei 74LS157 können die Pins 1,8,15 und 16 direkt angelötet werden) noch das Multiplexen von A16 und A17 möglich gemacht werden. Der Ausgang MA8 (Abb.2) wird (gegebenenfalls über einen 33-Ohm-Widerstand) an die Pins 1 der RAMs geführt.

Anstelle von A15 muß das Signal A15' (Abb.1) an die Adreßmultiplexer geführt werden.

Bei Computern mit 4116-RAMs müssen alle Speicherchips entfernt werden, und bei Vorhandensein getrennter Daten-Treiber für verschiedene 16K- Blöcke muß auch die Freigabe der Treiber nicht benutzter RAM-Reihen durch Anlegen von +5V über einen 4K7-Widerstand an den E*-Eingang der Treiber verhindert werden.

Da diese Speicherchips mit -5V an Pin 1, +12V an Pin 8 und +5V an Pin 9 versorgt werden, müssen diese Leitungen von den Versorgungsspannungen abgetrennt werden, Die Versorgungsspannung +5V wird an die Pins 8 gelegt. Bei den die Pins 1 und die Pins 9 verbindenden Leitungen werden alle Abblockkondensatoren entfernt. An die Pins 9 wird MA7 und an die Pins 1 wird MA8 der Adreßmultiplexer gelegt.

Das CAS*-Signa1 an Pin 15 der 4116-RAMs ist ein mit einem 16K-Freigabesignal (meistens durch OR, 74LS32) verknüpftes CAS*. Anstelle des 16K-Freigabesignals muß ein low aktives (bei OR-Verknüpfung) Freigabesignal, das den gesamten als RAM vorgesehenen Adreßraum des Z80 umfaßt (nur ROM und memory-mapped I/O ausgeblendet) oder bei Speichererweiterung auf mehr als 256K eines der Signale CASEN1* - CASEN4* verwendet werden.

Banking der oberen 32K ist nur dann sinnvoll, wenn es sich um einen BASIC-im-ROM-Computer, der nicht CP/M-fähig ist, handelt, und läßt sich durch Invertieren von A15 (durch z.B. 74LS04) für die Schaltung in **Abb. 1** erreichen.

Einbau des Banking-Boards

Die Wahl der Portadresse für das Latch zum Einstellen der Bank Nr. wurde flexibel gehalten. Die Portadresse kann sich aus maximal 7, minimal 5 high aktiven und maximal 3, minimal 1 low aktiven Adreßbits zusammensetzen.

Low aktive Adreßleitungen werden an die Punkte 8 bis 10 angeschlossen. Der Punkt 8 muß auf alle Fälle belegt werden. Wenn dann nur noch eine zusätzliche Adreßleitung low sein soll, muß diese an beide Punkte (9 und 10) gelegt werden. Wenn die Punkte 9 und 10 belegt werden, muß Punkt 11 mit einem der Punkte 1-7 verbunden werden. Die high aktiven Adreßleitungen werden an die restlichen Punkte (der Punkte 1-7) gelegt. Nicht belegte Punkte (der Punkte 1-7) werden mit belegten Punkten verbunden. Selbstverständlich müssen A6 und A7 auch noch an die dafür vorgesehenen Punkte (n und m) neben dem 74LS00 angeschlossen werden.

Die Adreßdekodierung sei am Beispiel der Portadresse ECH genauer beschrieben: ECH hat folgendes Bitmuster:

A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	1	1	0	0

Es müssen dann A0, A1 und A4 als low aktive Adreßleitungen an die Punkte 8, 9 und 10 gelegt werden und wegen der Belegung der Punkte 9 und 10 muß Punkt 11 mit z.B. Punkt 3 verbunden werden. Die Adreßleitungen A2, A3, A5, A6 und A7 werden dann z.B. an die Punkte 1, 2, 4, 6 und 7 angeschlossen. Der unbenutzte Punkt 5 muß dann mit z.B. Punkt 4 verbunden werden.

Das Signal CASEN* ist das CAS*-Freigabe-Signal für einen mit 4164-RAMs bestückten Speicher. Wenn vor dem Umrüsten auf 41256-RAMs noch 4116-RAMs vorhanden waren, muß ein solches Signal noch hergeleitet werden. BASEN* muß bei allen Adressen, die weder ROM noch memory-mapped I/O ansprechen, low aktiv sein. Bei TRS 80 und Kompatiblen ist dies ein Signal 16-64K*, das durch Invertieren des Signals 0-16K* und - leider nicht überall durchgeführt- OR-Verknüpfung mit MERQ* hergeleitet wird.

Die Freigabe des Datenbus-Transceivers

für die RAMs und eventuell auch ROMs kann in einigen Computern größere Schwierigkeiten bereiten. Ara einfachsten gestaltet sich die Lösung dieses Problems, wenn nur 8 Stück 4164-RAMs vorhanden sind und der zugehörige Datenbus-Transceiver ausschließlich die RAMs bedient, In diesem Fall kann die Freigabe des Transceivers so, wie sie war, belassen werden.

Auch dann, wenn vorher mehrere Reihen 4116-RAMs vorhanden waren und später durch Umrüsten auf 4164-RAMs in einer Reihe und Freilassen aller anderen Sockel die Freigabe eines für alle RAMs gemeinsamen Datenbus- Transceivers auf die 4164-RAMs angepaßt wurde, braucht bei Verwendung nun wieder aller Sockel die Freigabe des Transceivers nicht mehr geändert zu werden.

Bei Vorhandensein je eines Transceivers für eine Reihe RAMs (ursprünglich 4116) und nun Bestückung von 2 oder 4 Reihen mit 41256-RAMs müssen zur Freigabe der Transceiver die Signale CASEN1* bis CASEN4* des Banking-Boards verwendet werden.

Wenn für ROM und RAM bisher ein gemeinsamer Datenbus-Transceiver bestanden hat, so kann dessen Freigabe nur bei bisheriger Bestückung mit 4164-RAMs beibehalten werden. In Fällen, wo für ROM und einen 16K-RAM-Block (z.B. ältere GENIE-Modelle) ein gemeinsamer Datenbus-Treiber besteht, kann nur diese eine Reihe RAM-Sockel verwendet werden und die Freigabe muß die vollen 64K abzüglich von Adreßbereichen für memory-mapped I/O umfassen. Wenn in solchen Fällen eine Umrüstung auf 4164-RAMs schon durchgeführt worden ist, ist die Freigabe des Treibers bereits den Anforderungen für den Betrieb von 41256-RAMs mit Banking gerecht.

Anschluß des Banking-Boards

Signal	Anschlußpunkt Banking-Board	TRS 80 IC	TRS 80 Typ	TRS 80 Pin	GENIE I/II IC	GENIE I/II Typ	GENIE I/II Pin	Komtek 1 IC	Komtek 1 Typ	Komtek 1 Pin	mc-CP/M-Computer IC	mc-CP/M-Computer Typ	mc-CP/M-Computer Pin
D0	g	76	367	11	19	367	6	42	2732	9		Z80	14
D1	f	·	·	13	·	·	10	·	·	10		·	15
D2	e	75	·	13	·	·	4	·	·	11		·	12
D3	d	·	·	5	·	·	13	·	·	13		·	8
D4	c	·	·	7	·	·	11	·	·	14		·	7
A0	8	55	·	11	17	·	9	·	·	8		·	30
A1	9	·	·	13	·	·	7	·	·	7		·	31
A2	1	22	·	11	·	·	5	·	·	6		·	32
A3	2	·	·	13	·	·	3	·	·	5		·	33
A4	10	39	·	7	4	·	9	·	·	4		·	34
A5	4,5	·	·	9	·	·	7	·	·	3		·	35
A6	6,m	·	·	5	·	·	11	·	·	2		·	36
A7	7,n	·	·	11	·	·	5	·	·	1		·	37
A15	h	38	·	9	6	·	9	33	Z80	5		·	5
RESET* (SYSRES*)	a	37	02	1	5	Z80	26 #1	·	·	26 #1		·	26
WR* (OUT*)	l	23	32	3	16	367	11	·	·	22		·	22
IORQ* (OUT*)	k	·	·	·	·	·	9	·	·	20		·	20
CASEN*	b	74	00	10	36	04	2 #2	43′	04	2 #2		32	1 #7
RFSH*	o	40	Z80	28	16	367	3	33	Z80	28		Z80	28
A7′	p	51	157	14	24	157	14 #3	30	157	5 #4		157	13 #8
A15′	q	35	·	6	23	·	6 #5	·	·	6 #5		·	14 #8
A16	r	51′	·	2 #6	24′	·	2 #6	34′	·	2 #6		·	3 #6
A17	s	·	·	3 #6	·	·	3 #6	·	·	3 #6			2 #6
CASEN1*	w	-----------			-----------			54	32	12	-----------		
CASEN2*	v	-----------			-----------			2	2	1	-----------		
CASEN3*	u	-----------			-----------			-----------			-----------		
CASEN4*	t	-----------			-----------			-----------			-----------		
+5V	j	52	157	16	23	157	16	30	157	16		Z80	11
GND	i	·	·	8	·	·	8	·	2	8		·	29

#1 Anstelle des reinen RESET*-Signals ist eine AND-Verknüpfung von RESET* und NMI * (Z80, Pin17) zweckmäßiger, um bei jedem Booten Bank 0 einzustellen.

#2 Anstelle dieses invertierten 0-16K*-Signals ist es sicherer, dieses Signal über ein OR-Gatter 74LS32 mit MERQ* (GENIE: IC16, 74LS367, Pin 5/ Komtek 1: IC33, Z80, Pin 19) zu verknüpfen.

#3 Das beim Umrüsten auf 4164-RAMs an diesen Pin gelegte Signal ist an Pin 13 von Z24, 74LS157 zu legen. Die dorthin führende Leiterbahn ist zu durchtrennen.

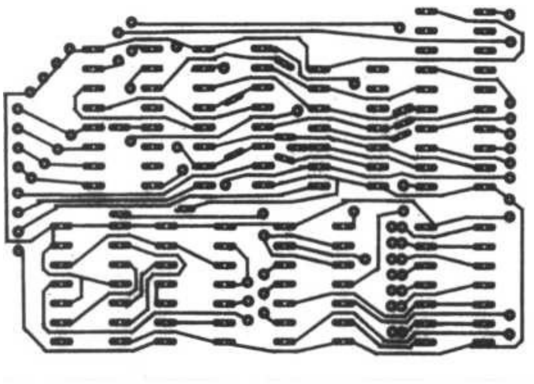
#4 Das beim Umrüsten auf 4164-RAMs an diesen Pin gelegte Signal ist an Pin 3 von IC 34, 74LS157 zu legen. Die dorthin führende Leiterbahn ist zu durchtrennen.

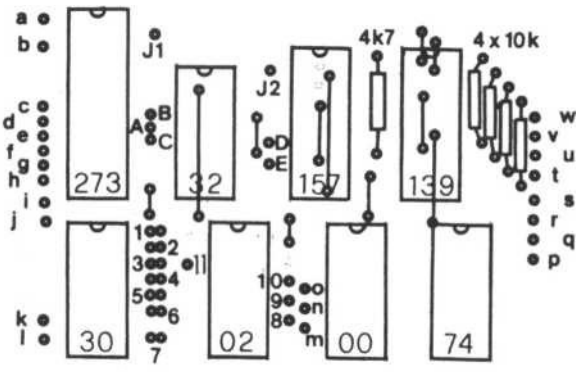
#5 Die an diesen Pin führende Leiterbahn ist zu durch- trennen.

#6 Dieses IC 74LS157 wird mit den Pins 1,8, 15 und 16 huckepack auf einen anderen 74LS157 Multiplexer auf gelötet. Die Numerierung bezieht sich auf das IC darunter.

#7 74LS32 auf Höhe des EPROMs

#8 74LS157 beim u1-Kondensator





3) Layout, Bestückung, Drahtbrücken, Jumper und externe Anschlüsse

Hier die Grafik abgetippt, wo das ? steht bitte kontrollieren. Im Original ist hier eine Leerstelle. ***

Signal	Anschlußpunkt Banking-Board	TRS 80 IC	TRS 80 Typ	TRS 80 Pin	Genie I/II IC	Genie I/II Typ	Genie I/II Pin	Komtek 1 IC	Komtek 1 Typ	Komtek 1 Pin	mc-CP/M-Computer IC	mc-CP/M-Computer Typ	mc-CP/M-Computer Pin
D0	g	76	367	11	19	367	6	42	2732	9		Z80	14
D1	f	"	"	13	"	"	10	"	"	10		"	15
D2	e	75	"	13	"	"	4	"	"	11		"	12
D3	d	"	"	5	"	"	13	"	"	13		"	8
D4	c	"	"	7	"	"	11	"	"	14		"	7
A0	8	55	"	11	17	"	9	"	"	8		"	30
A1	9	"	"	13	"	"	7	"	"	7		"	31
A2	1	22	"	11	"	"	5	"	"	6		"	32
A3	2	"	"	13	"	"	3	"	"	5		"	33
A4	10	39	"	7	4	"	9	"	"	4		"	34
A5	4,5	"	"	9	"	"	7	"	"	3		"	35
A6	6,m	"	"	5	"	"	11	"	"	2		"	36
A7	7,n	"	"	11	"	"	5	"	"	1		"	37
A15	h	38	"	9	6	"	9	33	Z80	5		"	5
RESET*(SYSRES*)	a	37	02	1	5	Z80	26 #1	"	"	26 #1		"	26
WR* (OUT*)	l	23	32	3	16	367	11	"	"	22		"	22
IORQ* (OUT*)	k	"	"	"	"	"	9	"	"	20		"	20
CASEN*	b	74	00	10	36	04	2 #2	43′	04	2 #2		32	1 #7
RFSH*	o	40	Z80	28	16	367	3	33	Z80	28		Z80	28
A7′	p	51	157	14	24	157	14 #3	30	157	5 #4		157	13 #8
A15′	q	35	"	6	23	"	6 #5	"	"	6 #5		"	14 #8
A16′	r	51′	"	2 #6	24′	"	2 #6	34′	"	2 #6		"	3 #6
A17′	s	"	"	3 #6	"	"	3 #6	"	"	3 #6		?	2 #6 ***
CASSEN1*	w	------------			------------			54	32	12	-----------		
CASSEN2*	v	------------			------------			2	2	1	-----------		
CASSEN3*	u	------------			------------			------------			-----------		
CASSEN4*	t	------------			------------			------------			-----------		
+5V	j	52	157	16	23	157	16	30	157	16		Z80	11
GND	i	"	"	8	"	"	8	"	2	8		"	29

Parallel-I/O-Adapter mit dem PPI8255

Helmut Berhardt

TRS-80, GENIE und Komtek haben weder für IEC-Bus, RS232C noch sonst irgendwelche standartisierten Schnittstellen Anschlüsse vorbereitet. Das einzige (außer Stecker für Drucker und Floppy), was man vorfindet, ist der herausgeführte Systembus. Leider gibt es keinerlei Interfaces, die auf den Systembus aufgesteckt werden und dann Standard-Schnittstellen bereitstellen. Es ist auch nicht zu erwarten, daß irgendein Hersteller für unsere wenig bedeutenden Geräte sowas je anbieten wird. Wir müssen uns da schon selber helfen.

Andererseits ist der Systembus für Bastler viel inreressanter als ein IEC-Bus oder User-Port. Es stehen dabei alle Möglichkeiten offen, irgendwelche Schnittstellen selbst zu bauen und über den Systembusstecker anzuschließen. Man ist nicht auf die vom Hersteller vorgegebenen Möglichkeiten beschränkt.

Eine einfache Parallel-I/O-Schnittstelle mit 24 Leitungen läßt sich mit dem 8255 aufbauen. Neben dem 8255 werden noch 2 weitere ICs zur Portdecodierung benötigt. Die in Abb.1 vorgeschlagene Schaltung benutzt die Portadressen 98H-9BH. Sie ist so einfach, daß sich die Herstellung eines Layouts nicht lohnt. Die Schaltung läßt sich auf einem 5 cm * 8 cm Stück Lochrasterplatine aufbauen.

Der Anschluß an den Systembus des Computers ist in Tabelle 1 wiedergegeben. Für den TRS-80 ergeben sich dabei gewisse Schwierigkeiten. Die Versorgungsspannung von +5V liegt nicht auf dem Systembus-Stecker. Sie muß an geeigneter Stelle auf dem CPU-Board entnommen werden. Außerdem ist das Signal IORQ* nicht vorhanden. Anstelle von IORQ* muss dann auch an Pin 11 des 74LS02 das Signal A2 gelegt werden. Anstelle der nicht verfügbaren Signale RD* und WR* werden die Signale IN* und OUT* an die Pins 5 und 36 des 8255 gelegt.

Der 8255 hat gegenüber dem für den Z80 üblichen PIO-Baustein den Nachteil, daß er keinen IM2-Interrupt liefern kann und daß die Leitungen eines Ports nicht unabhängig voneinander als Input oder Output definiert werden können. Dafür stehen aber in der einfachsten Betriebsart (Mode 0) drei 8-Bit-Ports zur Verfügung. Beim Port C lassen sich zusätzlich noch oberes und unteres Nibble unabhängig voneinander als Input oder Output festlegen.

Ein echtes Handshaking ist mit einzelnen Leitungen des Port C für die Ports A und B ebenfalls möglich, wenn die Betriebsarten Mode 1 oder Mode 2 programmiert werden. Dann muß noch ein zusätzliches IC 74LS05 eingesetzt weden, das die Interrupt-Ausgänge des 8255, PC0 und PC3 invertiert und auf Open-Collector Ausgänge bringt. Außerdem kann darüber auch ein externes, low aktives Strobe-Signal an den INT-Eingangder CPU geführt werden, wenn das Handshaking nicht durch den 8255 erledigt werden soll.

Tabelle 1: Anschuß an den Systembus

	Pin des Systembus-Steckers		
Signal	**GENIE**	**Komtek**	**TRS-80**
D0	15	46	30
D1	14	41	22
D2	12	40	32
D3	16	39	26
D4	20	42	18
D5	11	37	28
D6	18	44	24
D7	17	43	20
A0	10	34	25
A1	7	32	27
A2	9	30	40

Tabelle 2: Pinbelegung des 40poligen I/O-Steckers

Signal	**Pin**	**Pin**	**Signal**
PA4	39	40	PA3
PA5	37	38	PA2
PA6	35	36	PA1
PA7	33	34	PA0
GND	31	32	GND
-	29	30	-
-	27	28	-
-	25	26	-
-	23	24	-
-	21	22	ext.STROBE*
-	19	20	-
PC7	17	18	PC0

A3	8	28	34
A4	6	26	31
A5	5	24	35
A6	4	22	38
A7	3	20	36
IORQ*	38	35	40 #2 (A2)
RD*	41	38	19 #2 (IN*)
WR*	40	36	12 #2 (OUT*)
RESET*	46 #1	33 #1	2 (SYSRES*)
INT*	31	10	21
+5V	19	17,18	39 #3
GND	1,2,	1,2,	8,29,37
	49,50	49,50	

PC6	15	16	PC1
PC5	13	14	PC2
PC4	11	12	PC3
+5V	9	10	+5V
PB7	7	8	-
PB6	5	6	PB1
PB5	3	4	PB2
PB4	1	2	PB3

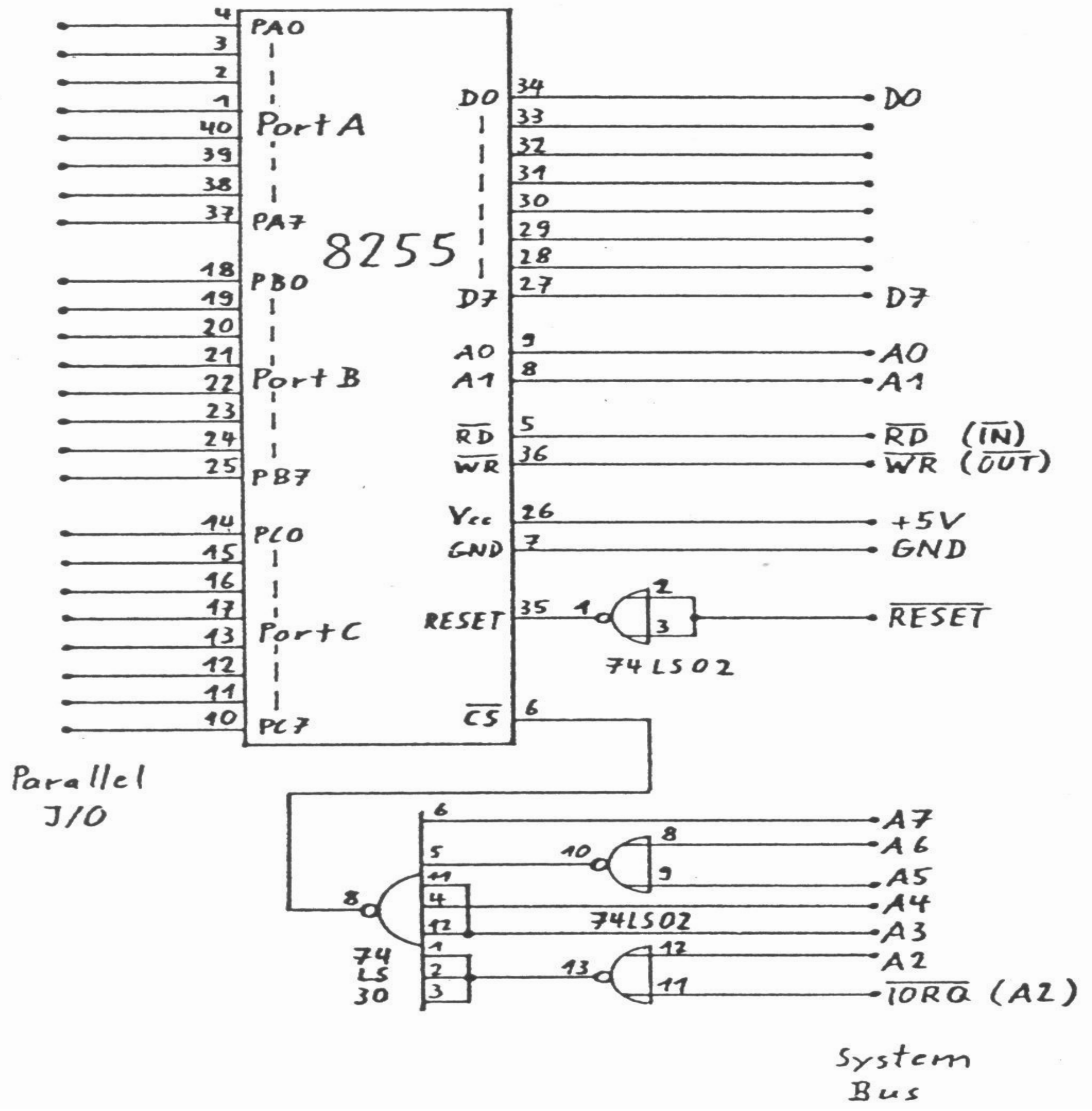
Anmerkungen zu Tabelle 1

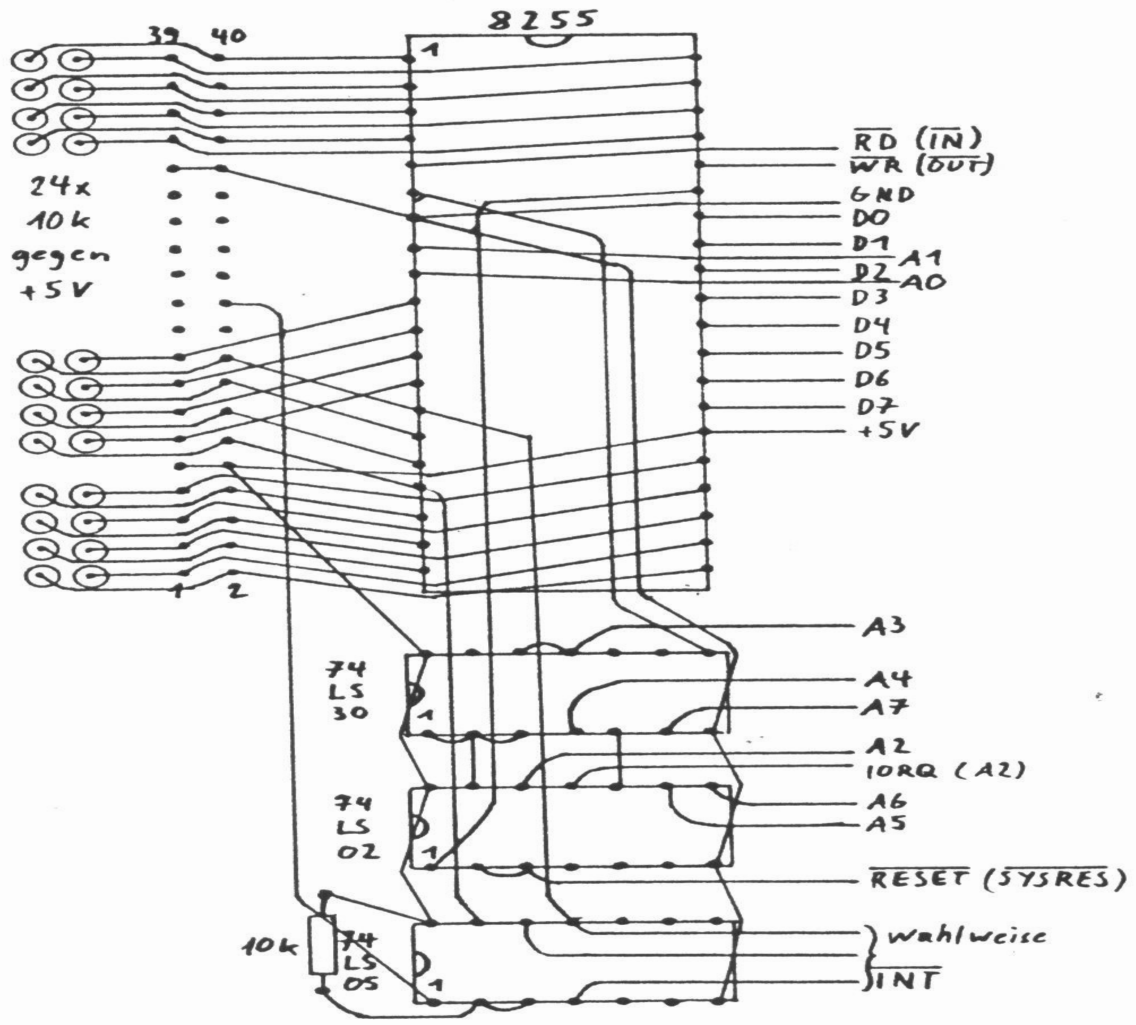
#1 anstelle von RESET* sollte ein durch AND-Verknüpfung aus RESET* und NMI* gebildetes SYSRES* benutzt werden

#2 anstelle der nicht vorhandenen Signale IORQ*, RD* und WR* müssen A2, IN* und OUT* benutzt werden

#3 Die Versorgungsspannung ist nur am Expansion-Interface und nicht am Grundgerät zu entnehmen; bei Anschluß an das Grundgerät kann sie an geeigneter Stelle auf dem CPU-Board abgenommen werden oder auch vom angeschlossenen Peripheriegerät geliefert werden, zu dem auch Masse-Verbindung bestehen muß

Abb.1: Schaltung und Verdrahtung des I/O-Boards


8255
Port A
PA0
PA7
Port B
PB0
PB7
Port C
PC0
PC7
D0
D7
A0
A1
RD
WR
Vcc
GND
RESET
CS
RD (IN)
WR (OUT)
+5V
GND
RESET
74LS02
74 LS 30
A7
A6
A5
A4
A3
A2
IORQ (A2)
Parallel I/O
System Bus


8255
39 40
24x 10k gegen +5V
RD (IN)
WR (OUT)
GND
D0
D1
D2
A1
A0
D3
D4
D5
D6
D7
+5V
A3
A4
A7
A2
IORQ (A2)
A6
A5
RESET (SYSRES)
wahlweise
INT
74 LS 30
74 LS 02
74 LS 05
10k

Sidecar mit 4 Rädern
Amiga's Sidecar wird ein selbständiger PC

Helmut Bernhardt, Christof Ueberschaar

Eine der aufwendigsten Varianten, einen PC-XT zu betreiben, besteht darin, anstelle einer Tastatur und, eines Bildschirms mit Video-Karte einen kompletten 68000-Computer für solche Kleinigkeiten einzusetzen. Dieser vornehme Stil wird beim Gespann Amiga 1000 und dem dazugehörenden Sidecar gewahrt. Das Sidecar selbst ist fast ein vollständiger PC-XT mit 256K RAM und Sockeln zum Aufrüsten auf 512K.

Es drängt sich dem Sidecar-Besitzer geradezu auf, zu überlegen, ob man diesen Teil des Gespanns nicht selbständig betreiben kann und dadurch die Amiga für andere Anwendungen freihält. Dabei gilt es, nur eine Video-Karte und einen Tastatur-Anschluß nachzurüsten und das Interface zur Amiga (auf das Sidecar- Motherboard aufgesteckte Platine mit Anschluß zur Amiga, eine reiche Quelle für auszuschlachtende ICs) zu entfernen.

Dem Sidecar ein eigenes Video-Interface zur Verfügung zu stellen, ist das geringste Problem. Für wenig Geld läßt sich eine gebrauchte Hercules-Karte und ein TTL-Monitor auftreiben. Um beim Betrieb einer eigenen Video-Karte im Sidecar nicht die sonst über die Pins 11 und 13 von U26, 74HCT367 ausgelesenen Video-Statusbits denen der Video-Karte entgegenwirken zu lassen, müssen die Schalter 1 und 2 des 4fach-DIL-Schalters SW2 auf ON gestellt werden. Damit gibt dann das Custom-IC U40 nicht mehr die beiden Treiber von U26 frei, die bei Zugriff auf Port 3BAH (wenn Schalter 1 von SW2 auf OFF steht) bzw. auf Port 3DAH (wenn Schalter 2 von SW2 auf OFF steht) die Ausgänge eines CD4020-Zählers als Statusbits auf DO und D3 des Datenbus legt, um der CPU das Vorhandensein einer entsprechenden Video-Karte vorzutäuschen.

Der Betriebssoftware ist die Art des verwendeten Video-Adapters über die Stellung der Schalter 5 und 6 des 8fach-DIL-Schalters SW1 bekanntzugeben:

Schalterstellungen an SW1 Schalter 5	Schalter 6	Art des Video-Adapters
OFF	OFF	MGA (monochrom), MDA (Hercules)
OFF	ON	CGA 40*25
ON	OFF	CGA 80*25
ON	ON	nicht festgelegt

Beim Einsatz von EGA- und VGA-Karten spielt die Schalterstellung keine Rolle. Wie Schalter 5 und 6 haben auch alle anderen Schalter von SW1 die gleiche Bedeutung wie in jedem anderen IBM-kompatiblen PC-XT.

In diesem Zustand (noch ohne Tastatur) bootet jede Version von MS-(PC-)DOS bis zur ersten erwarteten Tastatureingabe. Außer dem ohne Amiga-Interface nicht vorhandenen Tastatur-Anschluß ist das Sidecar offensichtlich vollkommen PC-XT- kompatibel.

Aus Schaltplänen eines anderen PC-Kompatiblen geht hervor, wie der Anschluß der inzwischen angeschafften Tastatur zu erfolgen hat. Danach sollen die seriell /parallel-gewandelten Scan-Codes der Tastatur über den als Eingang konfigurierten Port A eines 8255-Parallelport-Bausteins eingelesen werden. Die Steuerung der Tastatur wird über D6 und D7 des als Ausgang eingestellten Ports B des 8255 durchgeführt. Die anderen Bits des Ports B erledigen weitere Aufgaben wie Sound-Ausgabe, Rücksetzen von Flags bei Parity-Fehlern im Speicher oder Fehlern bei Erweiterungskarten und

Schaltern des DIL8-Schalters zur Vorgabe der System-Konfiguration zum Auslesen über die untere Hälfte des Ports C des 8255. Die obere Hälfte des Ports C dient zum Auslesen der Fehler-Flags und des jeweiligen Pegels am Lautsprecher- Eingang.

Einen 8255 sucht man im Sidecar aber vergebens. Andererseits müssen bis auf Parity-Check (die Sidecar-Hardware sieht das nicht vor) und Tastatur-Anschluß (was durch das Amiga-Interface bereitgestellt wird) alle anderen Funktionen des 8255 auch im Sidecar auf andere Art realisiert sein, denn die entspechenden Steuerungen funktionieren dort einwandfrei. Auf der Suche nach Fragmenten des teilweise nachgebildeten 8255 stößt man zunächst auf U25, 74HCT174. Mit diesem Latch werden 5 Ausgabe-Bits des Ports B des 8255 bereitgestellt. Die Bits 2, 4 und 6 bleiben dort aber unberücksichtigt.

Der Port C des 8255 wird durch U20, 74HCT257 (Auslesen des 8fach-DIL-Schalters SW1 über D0-D3) und U26, 74HCT367, Pins 1-10 (Auslesen von Fehler-Flags über D4-D7) dargestellt. Port A, über den die Scan-Codes der Tastatur einzulesen sind, fehlt gänzlich und eine Möglichkeit, die über die verfügbaren Bits von Port B eingestellte Systemkonfiguration zurückzulesen, ist auf dem Motherboard auch nicht zu finden. Diese Funktionseinheiten werden offensichtlich vom Amiga- Interface bereitgestellt und müssen bei dessen Wegfall nachgerüstet werden.

Die in Abb.1.1 gezeigte Schaltung stellt den mit einem 74LS245 nachempfundenen Port A des 8255 dar, an den ein 74LS322-Seriell/Parallel-Wandler angeschlossen ist. Das über D6 und D7 von Port B abgewickelte Handshak ing zwischen Tastatur und Computer wird mit einigen Flip Flops und Open Collector Treibern realisiert. Das Freigabe-Signal zum Auslesen des Ports A und auch die Freigabe- Signale zum Lesen und Schreiben des Ports B werden mit der in Abb.1.2 wieder- gegebenen Schaltung erzeugt.

Der Port B läßt sich gemäß Abb.1.3 auch rücklesbar gestalten. Um den Verdrahtungsaufwand zwischen Zusatzschaltung und Motherboard in Grenzen zu halten, wird der Port B zusätzlich zu der in U25 auf dem Motherboard schon vorhandenen Form auf unserem Tastatur-Interface nochmals durch einen 74LS174 - nun aber auch unter Berücksichtigung von D6 - nachgebildet. Davon werden aber nur D6 und D7 für das Handshaking verwendet. Die übrigen Bits werden nur zum Zurücklesen über den Treiber 74LS367 zur Verfügung gestellt (Abb.1.3).

Diese Schaltungselemente lassen sich auf einer kleinen Platine unterbringen, die anstelle des Amiga-Interface auf die Steckerleiste J3 aufzusetzen ist. Dafür muß am Tastatur-Interface von der Lötseite eine 34polige Pfostenbuchse angelötet werden. Bei Beschaffungsschwierigkeiten leistet auch eine VG64-a,b- Buchse nach Ablängen auf 34 Kontakte diesen Dienst.

Außer den auf J3 zugänglichen Signalen (+5V, GND, Daten und Adressen) müssen noch einige Signale von J4 und von zwei Punkten auf dem Motherboard zugeführt werden. Die Anschlußpunkte auf dem Platinchen sind aus Abb.2.3 zu ersehen, sie müssen mit folgenden Punkten auf dem Motherboard verdrahtet

Signal von/zu Tastatur-Interface	Anschlußpunkt auf dem Motherboard
IRQ1	J4, Pin28
PCLK	U27, 74LS74, Pin3
/PB7	J4, Pin17
/RESET	U25, 74HCT174, Pin1
/IOWC	J4, Pin8
/IORC	J4, Pin7
AEN	J4, Pin9

Der einfachen Schaltung entsprechend ist das Layout einseitig mit einer entsprechenden Anzahl Drahtbrücken <Abb.2.2) ausgeführt. Die Drahtbrücken müssen vor dem Einlöten der ICs bzw. der IC-Sockel (nur anreihbare Typen verwendbar) in dünnem Kupfer-Lackdraht oder Wrapdraht gezogen werden, um bei der teilweise dichten Lage Kurzschlüsse zu vermeiden.

Neben dem Widerstands-Netzwerk sind die Anschlüsse für die Tastatur zu finden. Die Belegung der üblichen 5poligen Diodenstecker von PC-Tastaturen geht aus Abb.3 hervor.

Ganz ohne Eingriffe auf dem Sidecar-Motherboard geht es leider nicht. Pin 9 von U27, 74LS74, der ebenfalls den Tastatur-Interrupt IRQ1 bedient, muß durchtrennt werden .

Der jetzt vollständige PC ist 100proz. IBM-kompatiblen (mit leider "nur" 512K RAM). Auch ein V20-Prozessor läuft problemlos mit dem vorhandenen Sidecar-BIOS und auch mit jedem anderen BIOS.

Leider bleiben nach Einstecken einer Video-Karte nur noch 2 Slots frei. Dabei mag es tröstlich sein, daß im Sidecar eine Floppy-Controller-Karte nicht mehr nötig ist, weil auf dem Motherboard schon ein vollständiges Floppy-Interface vorhanden ist.

Für den Anschluß von Fremdlaufwerken - maximal 2 Laufwerke sind möglich - muß aber das an Pin 16 des ansonsten voll Shugart-kompatiblen 34poligen Floppy- Steckers fehlende Signal /MOTOR ON nachgerüstet werden. Dieses kann man sich aber einfach aus den Signalen DS0 und DS1 erzeugen, indem man 2 Dioden AA118 von Pin 16 des Steckers in Durchlaßrichtung zu den Pins 10 und 12 des Steckers lötet. Eine sinnvolle Lösung ist die Verwendung von umschaltbaren 80/40-Track-5 1/4—Zoll Laufwerken, auf denen mit neueren Versionen von MS- (PC-)DOS sowohl 360K- als auch 720K-Disketten (das entspricht der in PC-User-Kreisen vermeintlich nur auf 3 1/2-Zoll Laufwerken möglichen Kapazität) verarbeitet werden können.

Belegung der Pfostenleisten J3 und J4

Stecker J3				Stecker J4			
+5V	1 o	o 2	+5V	+5V	1 o	o 2	+5V
		o 4	nc	/MEMR	3 o	o 4	/MEMW
D0	5 o	o 6	D1			o 6	nc
D2	7 o	o 8	D3	/IORC	7 o	o 8	/IOWC
D4	9 o	o 10	D5	AEN	9 o	o 10	/DACK0
D6	11 o	o 12	D7	GND	11 o	o 12	CLK
A0	13 o	o 14	A1	GND	13 o	o 14	IRQ3
A2	15 o	o 16	A3	IRQ7	15 o	o 16	IRQ4
A4	17 o	o 18	A5	/PB7	17 o	o 18	U29,Pin11
A6	19 o	o 20	A7	I/OCHRDY	19 o	o 20	/RESET IN
A8	21 o	o 22	A9	/WR	21 o	o 22	U40,Pin13
A10	23 o	o 24	A11	U40,Pin11	23 o	o 24	nc
A12	25 o	o 26	A13	nc	25 o	o 26	nc
A14	27 o	o 28	A15	U40,Pin12	27 o	o 28	IRQ1
A16	29 o	o 30	A17	SW2,4	29 o	o 30	SW2,1
A18	31 o	o 32	A19	SW2,3	31 o	o 32	SW2,2
GND	33 o	o 34	GND	GND	33 o	o 34	GND

Stückliste

Z1	74LS245	R1	2K2
Z2	74LS322	RN1	Widerstandsnetzwerk
Z3	74LS74		7*4K7 oder 4*4K7
Z4	74LS74	C1	100nF Keramik
Z5	7407	J3	34polige Buchsenleiste
Z6	74LS367		(lötseitig bestücken)
Z7	74LS174	Drahtbrücken (isolierter Draht)	
Z8	74LS32	Leiterkärtchen	
Z9	74LS138		

Beschriftung der Abbildungen

Abb.1.1: Nachgebildeter Port A eines 8255 mit Seriell/Parallel - -Wandler und Handshaking zwischen Computer und Tastatur

Abb.1.2: Erzeugen der Freigabe-

Abb.1.3: Der auslesbare Port B des 8255

Abb.2.1: Das Layout des Tastatur-Inter- -face, bestückungsseitig gesehen

Abb.2.2: beim einseitigen Layout sind etliche Drahtbrücken nötig

Abb.2.3: Bestückungsplan

Abb.3: Signalbelegung der 5poligen Diodenbuchse am PC für den Anschluß der Tastatur

Die Abbildungen 2.1 bis 2.3 geben die Größe des Boards direkt wieder. Die Originalzeichnungen sind in doppelter Größe vorhanden. Das in Abb. 2.2 und 2.3 hinterlegte Layout befindet sich auf einem Extrablatt, das unter die auf Pergamentpapier gezeichnete Bestückung gelegt wurde.

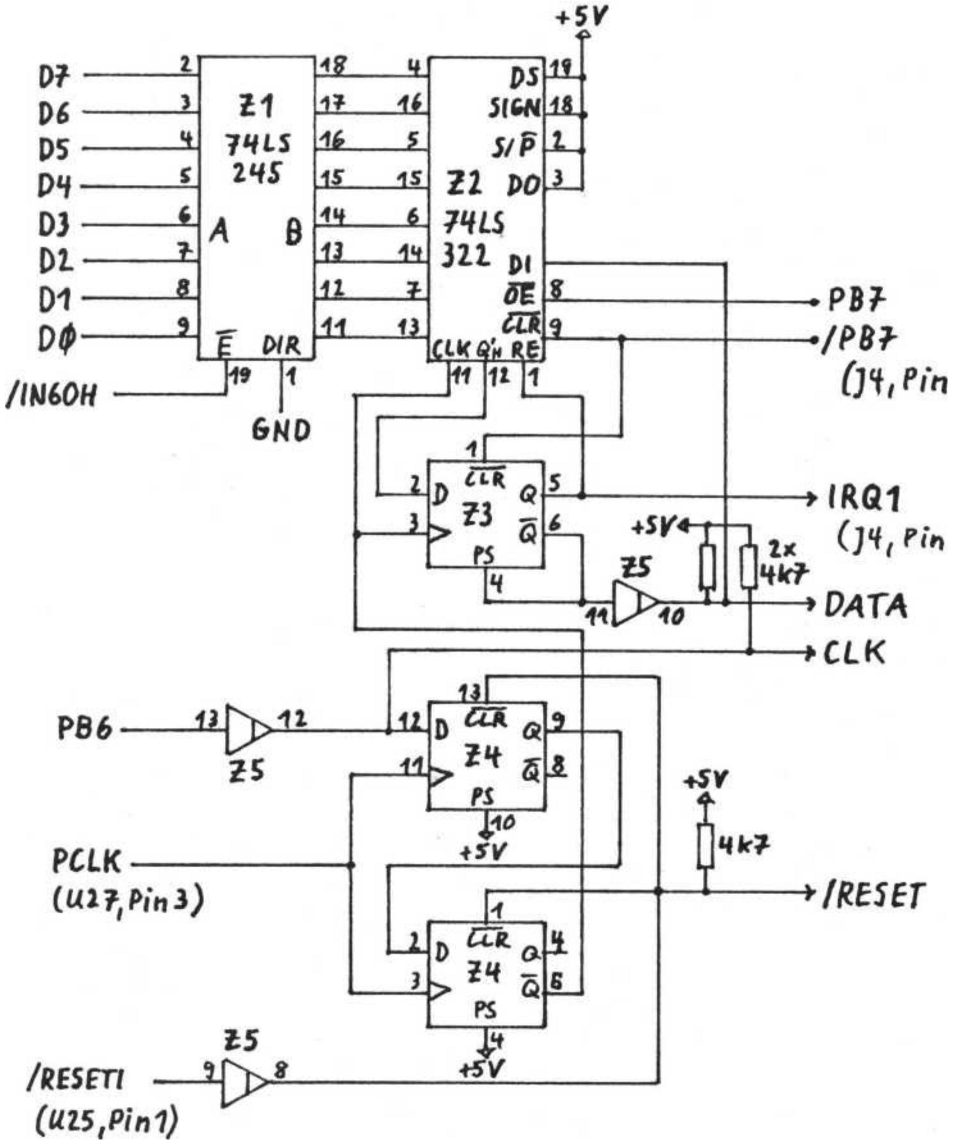

+5V
D7
D6
D5
D4
D3
D2
D1
DØ
/IN60H
Z1
74LS
245
A
B
E
DIR
GND
Z2
74LS
322
DS
SIGN
S/P
DO
DI
OE
CLR
CLK
Q'H
RE
PB7
/PB7
(J4, Pin
IRQ1
(J4, Pin
D
CLR
Z3
Q
PS
+5V
Z5
2x
4k7
DATA
CLK
PB6
Z5
Z4
PCLK
(U27, Pin 3)
+5V
4k7
/RESET
/RESETI
(U25, Pin1)

Abb. 1.1

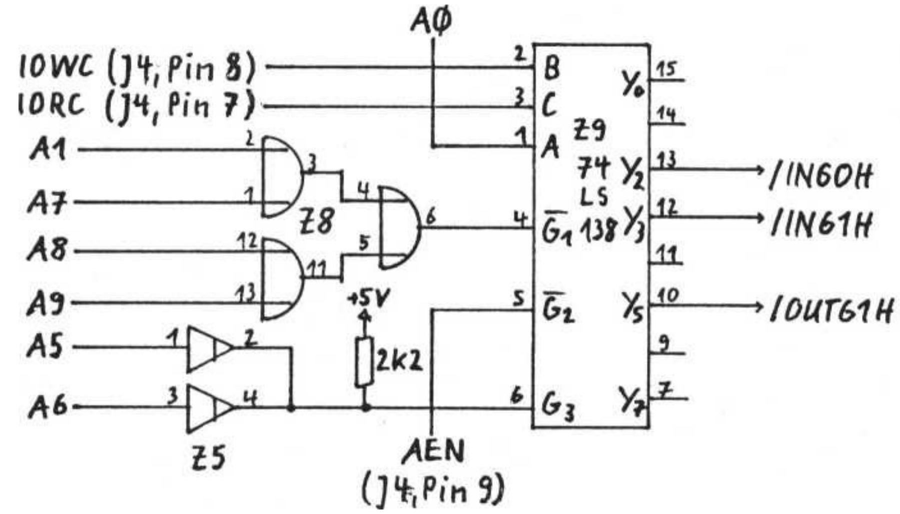

AØ
IOWC (J4, Pin 8)
IORC (J4, Pin 7)
A1
A7
A8
A9
A5
A6
Z8
+5V
2k2
Z5
AEN
(J4, Pin 9)
Z9
74
LS
138
B
C
A
G1
G2
G3
Y0
Y2
Y3
Y5
Y7
/IN60H
/IN61H
/OUT61H

Abb. 1.2

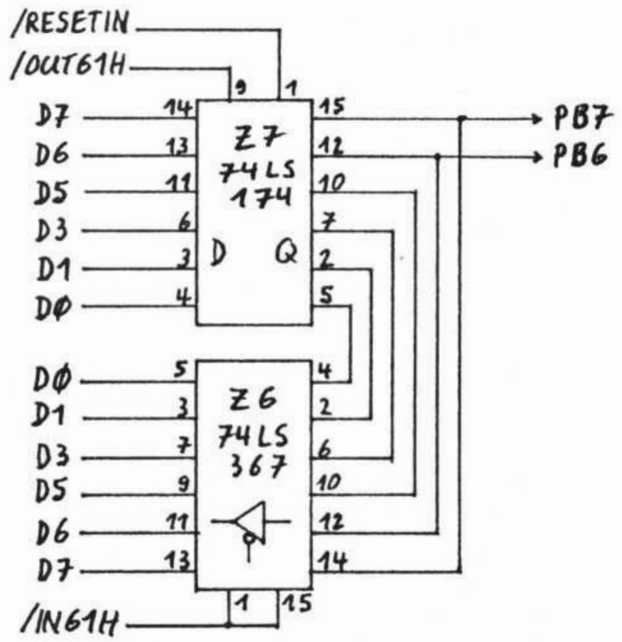

/RESETIN
/OUT61H
Z7
74LS
174
D
Q
PB7
PB6
D7
D6
D5
D3
D1
DØ
Z6
74LS
367
/IN61H

Abb. 1.3

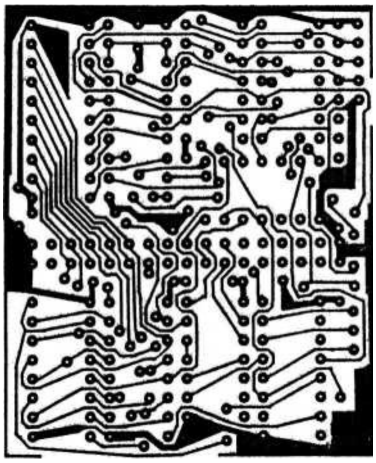


Abb. 2.1

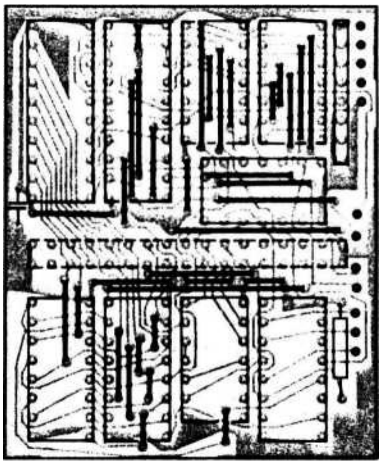


Abb. 2.2

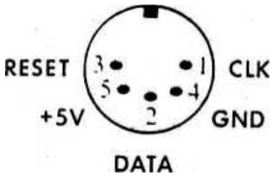


Abb. 3

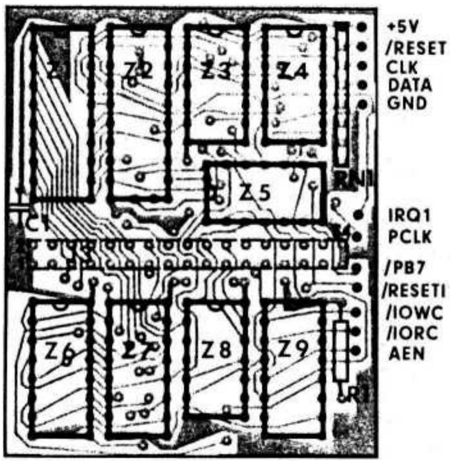


Abb. 2.3

Dein GIIIs, die 4 unbekannten Wesen

A: Hardware-Erweiterung für 1 MB RAM

Die 128 Refreshzyklen des Z80 lassen sich auf 256 Zyklen erweitern, was die Verwendung von 41256-RAMs ermöglicht (Abb. 3). Dadurch kann jeder Z80-Computer auf mindestens das vierfache Speichervolumen aufgebohrt werden, indem man die vorhandenen 4164-RAMs durch 41256-Typen ersetzt. Die beiden zusätzlich nötigen Adressen kann ein I/O-adressiertes Latch liefern (Abb. 1). Damit kann man zwischen vier Blocks mit je 64 kB umschalten.

Dabei würden dann aber der Stack, Interrupt-Serviceroutinen und alles an Software abgekniffen werden, was für eine einwandfreie Funktion nötig ist. Der Z80 würde sich in der neuen Bank, wo noch kein Programm enthalten ist, verlaufen. Deshalb muß ein Teilbereich des Speichers auch beim Umschalten der Bank erhalten bleiben, wo der Z80 alles Lebenswichtige vorfindet.

Im Genie III s ist das mit vier Banks in Form von vier Reihen 4164-RAMs so gelöst, daß mit dem Zustand von Bit 0 des Ports F9h festgelegt wird, ob der Adreßbereich 0000h-3FFFh oder der Bereich E000h-FFFFh nicht mit umgeschaltet wird (Common-Bereich):

```
Port F9h, Bit 0:      0: Common = 0000h-3FFFh
                      1: Common = E000h-FFFFh
```

Die Auswahl unter den vier Banks geschieht mit den Bits 6 und 7 desselben Ports:

```
Port F9h, Bits 6, 7:  00: Bank 0
                      01: Bank 1
                      10: Bank 2
                      11: Bank 3
```

Unabhängig von der eingeschalteten Bank wird bei Zugriff auf den als Common definierten Bereich immer auf den RAM-Block 0 zurückgeschaltet. Das wird durch entsprechende Steuerung der CAS*-Freigabesignale für die vier RAM-Reihen erreicht (Abb. 4).

Zur Vorgabe der Adressen A16 und A17 für 41256-RAMs anstelle der ursprünglichen 4164-RAMs wurde ein unter der Portadresse F3h decodiertes Latch 74LS273 herangezogen, dessen Ausgänge Q0 und Q1 diese Adressen liefern. Damit läßt sich innerhalb eines 256-kB-Blocks (8 Stück 41256-RAMs) vorgeben, welches Viertel der RAM-Bausteine ansprechbar sein soll. Die RAM-Reihe selbst wird weiterhin über Port F9h, Bits 6 und 7 selektiert. Das Zusammenspiel beider Ports bei der Adressierung jeweils eines Blocks zu 64 kB veranschaulicht die Abb. 6.

Das Umschalten auf die RAM-Reihe 0 beim Zugriff auf den als Common definierten Adreßbereich (Port F9h, Bit 0) leistet eine entsprechende Schaltung im GIIIs. Die Common-Bedingung wird auf Wunsch (s. u.) auch bei der Umschaltung innerhalb der Viertel der 41256-Chips beachtet: Die Selektion des unteren Viertels einer Reihe 41256-RAMs wird Uber die Steuerung des Pins 1 (Select) des 74LS157-Multiplexers in der abgebildeten Schaltung erreicht. Das wäre prinzipiell durch die AND-Verknüpfung der beiden low aktiven Signale COML* und COMH* (U84, 74LS00, Pins 12 und 13) möglich. Wenn eines dieser Signale low ist, wird Uber low am SEL-Eingang des Multiplexers GND auf die Ausgänge durchgeschaltet und nicht die Bits Q0 und Q1 des Latch. COML* und COMH* werden low, also aktiv, wenn der unten bzw. oben im 64-kB-Adreßraum als Common definierte Speicherbereich angesprochen wird.

Hier wurde aber zusätzlich die Möglichkeit geschaffen, über Q7 des Latch (Port F3h, Bit 7) die Steuerung des SEL-Eingangs des Multiplexers

zu unterbinden. Wenn D7 des Ports F3h high ausgegeben ist (Bit im Output gesetzt), wird die automatische Umschaltung auf das untere Viertel einer 256K-RAM-Reihe unterbunden. Dann ist nur noch die ursprüngliche Steuerung der CAS*-Freigabesignale vom Zugriff auf den festgelegten Common-Bereich belangt (Selektion einer RAM-Reihe, aber nicht des unteren Viertels).

Das Verhalten des Computers läßt sich mit Q7=1 aus dem Normalverhalten (nur 1 aktiver Common-Bereich, 16 Banks) in einen zweiten Zustand umschalten: Es gibt dann 6 Blocks zu je 256 kB (Port F3h, Bits 0 und 1) mit je einem eigenen Common-Bereich an der durch Port F9h, Bit 0 definierten Adresse. Jeder dieser Blocks kann unabhängig hiervon je nach Port F9, Bits 6 und 7 eine der vier RAM-Reihen benutzen. Das entspricht der parallelen Existenz von vier Computern mit je vier Banks zu 64 kB.

In anderen Worten: Wahlweise stellt die hier vorgestellte Schaltung einem Computer 1 MB RAM zur Verfügung, oder vier voneinander unabhängige Computer verwalten je 256 kB RAM. Die zweite Variante entspricht dem Originalzustand des GIIIs - mal vier. Das Nebeneinander der nunmehr 16 Blocks zu je 64 kB ist in Abb. 5 b und c dargestellt.

Da das verwendete Latch 74LS273 aber 8 Bits verfügbar hat, sollte damit weiteres Tune-Up verwirklicht werden.

Der Bildwiederholspeicher und der Zeichengenerator sind 6116LP15-RAMs. Diese Einheiten können unter Verwendung von 6264LP15-Speichern ebenfalls vervierfacht werden, wenn deren zusätzliche Adreßleitungen A11 und A12 mit den Ausgängen 02 und 03 des Latch beschaltet werden. Über D2 und D3 am Port F3h läßt sich einer von vier Bildwiederholspeichern mit eigenem Zeichengenerator-RAM selektieren.

Und schließlich sollte auch noch der HRG-Speicher von 64 auf 256 kB durch Austausch der 4164-RAMs gegen 41256-Typen auf acht Pages aufgebohrt werden. Die Selektion eines Viertels des neuen Speichers sollte über D4 und D5 am Port F3h erfolgen. Innerhalb des adressierten Viertels könnte dann weiterhin wie bisher eine von zwei Pages zu 32 kB gewählt werden. Die alte Software würde dadurch in ihrer Funktionsfähigkeit nicht beeinträchtigt werden.

Aus nicht verständlichen Gründen funktionierte das Umschalten bei einem Versuch nicht, Q4 und Q5 als Adressen A16 und A17 über einen Multiplexer 74LS157 (auch 74HC157) auf die neuen 41256-RAMs des HRG-Speichers zu legen. Sein Pin 1 wurde durch das gleiche Signal gesteuert wie die beiden schon existierenden Multiplexer. Evtl, kann die Verwendung eines S- anstatt des LS- bzw. HC-Typs das Problem lösen. Vielleicht kann uns jemand mit einem Tip weiterhelfen?

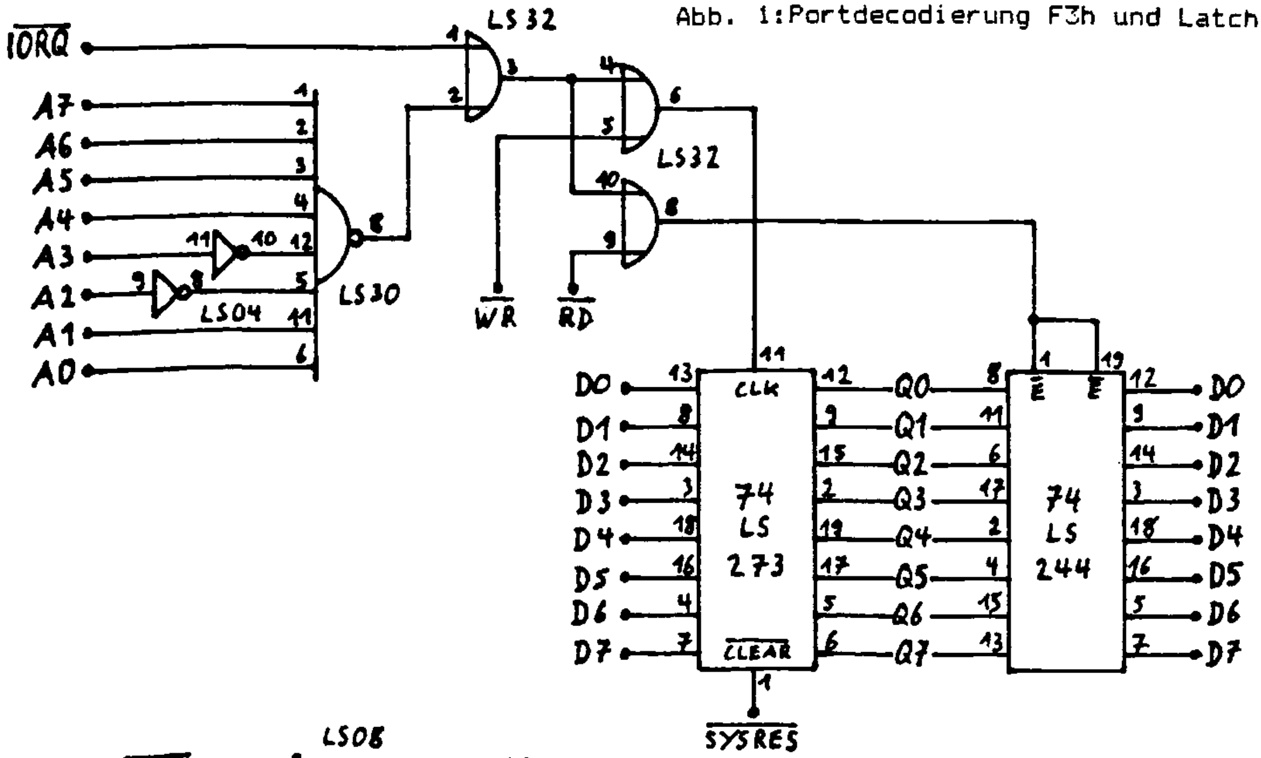
Auch eine andere Funktion, die nach Schaltplan einwandfrei arbeiten sollte, versagte den Dienst. Durch NAND-Verknüpfung von Q6 des Latch und MWR* sollte erreicht werden, daß bei high gesetztem Bit 6 im Port F3h beim Lesen des Speichers auf Computer 0 geschaltet wird, und nur beim Schreiben der RAM-Speicher des mit DO und D1 festgelegten Computers angesprochen wird. Damit wäre es besonders einfach geworden, den Speicherinhalt von Computer Nr. 0 auf seine drei Nachbarn zu übertragen, um sie mit Initial-Information zu laden. Das hätte der Fähigkeit des EG 64 MBA entsprochen, mit einem einfachen LDIR den ROM-Inhalt auf das parallele RAM zu kopieren. Auch dieses Feature funktioniert wider Erwarten nicht.

Im Layout wurden trotzdem die Schaltungen dieser nicht funktionierenden Elemente beibehalten. Evtl, gelingt es noch, diese Features später zum Laufen zu kriegen. Ansonsten steht einer anderweitigen Verwendung der Latch-Ausgänge Q4, Q5 und Q6 nichts im Wege. Die im folgenden beschriebene Software kommt ohne diese Funktionen aus. Die nicht funktionsfähigen Teile der Schaltung haben insb. auch keinen negativen Effekt auf sie, so daß sie im Hinblick auf späteres Gelingen durchaus bereits bestückt wer-

den können.

Die drei Gruppen zu je zwei Bits am Port F3h sind voneinander unabhängig ansprechbar. So kann z.B. über D2 und D3 zwischen mehreren Bildschirmen und Zeichensätzen gewählt werden, D4 und D5 können im Graphikspeicher einen Spooler mit luxuriöser Kapazität zur Verfügung stellen, während D0 und D1 ständig low bleiben, also gewissermaßen im Computer Nr. 0 gearbeitet wird. Und Computer Nr. 3 kann eine Graphik ändern, die Computer Nr. 1 erstellte usw..

Abb. 1:Portdecodierung F3h und Latch



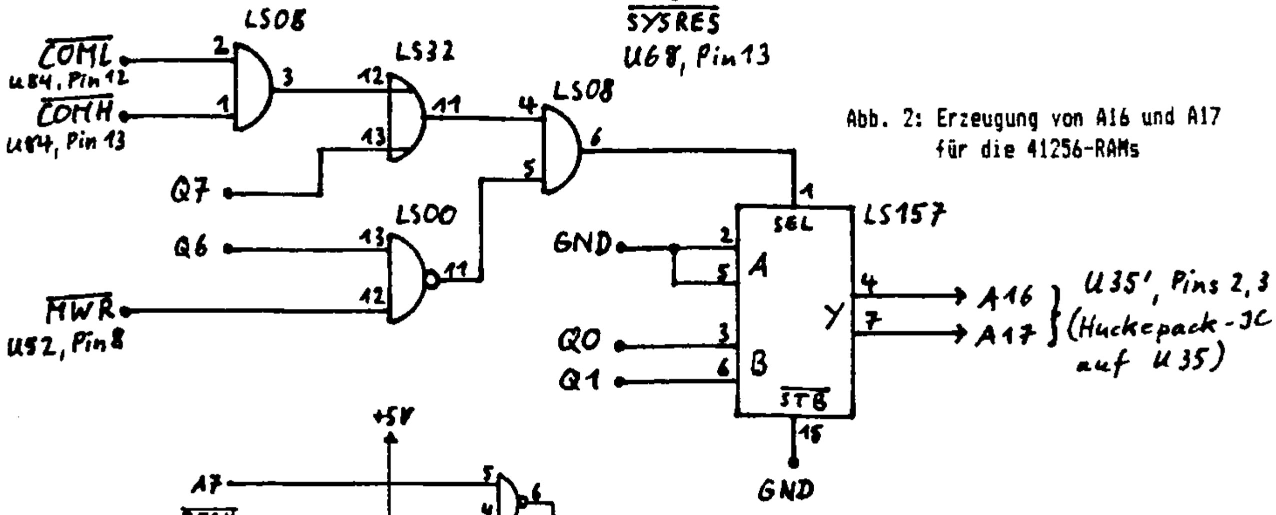


Abb. 2: Erzeugung von A16 und A17 für die 41256-RAMs

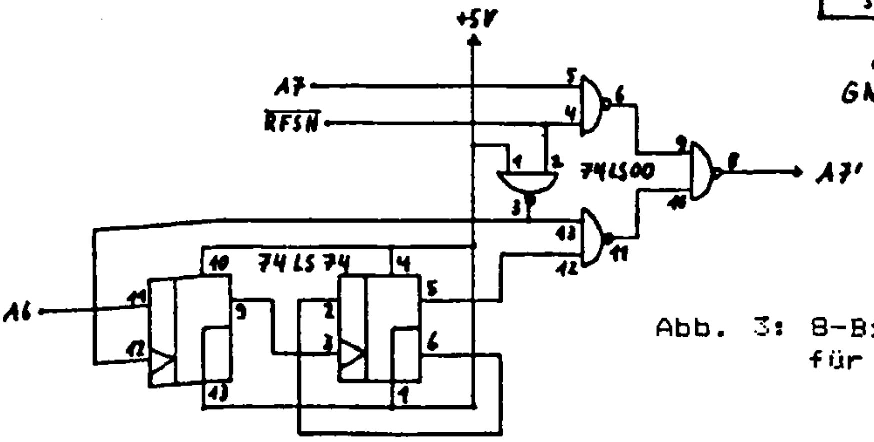


Abb. 3: 8-Bit-Refreshgenerator für die 41256-RAMs

Abb. 4: CAS*-Freigabesignale der 4 64-kB-Blocks

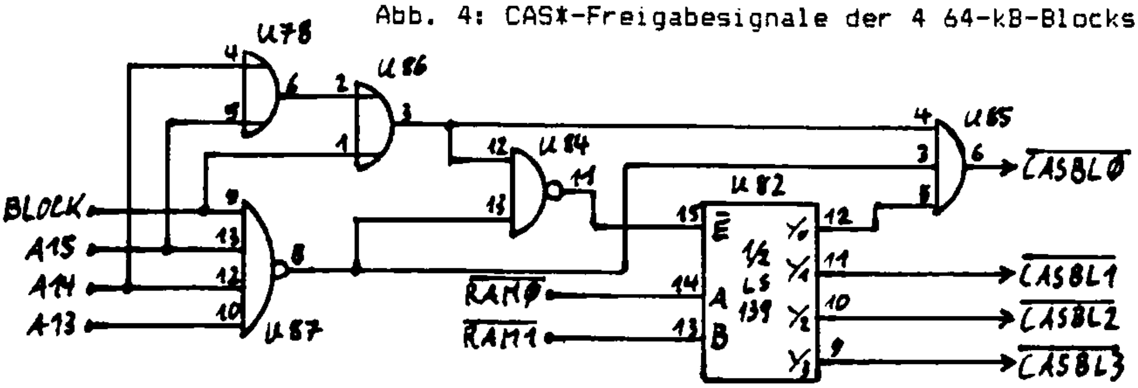
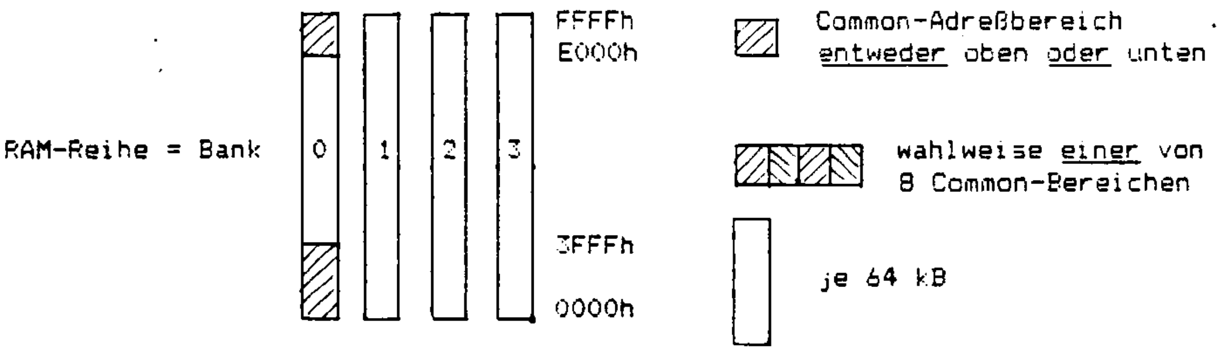
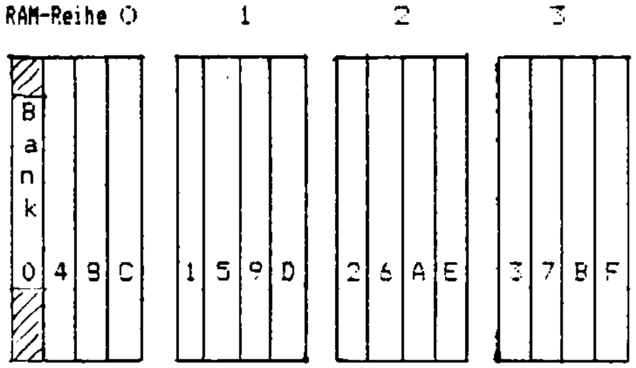
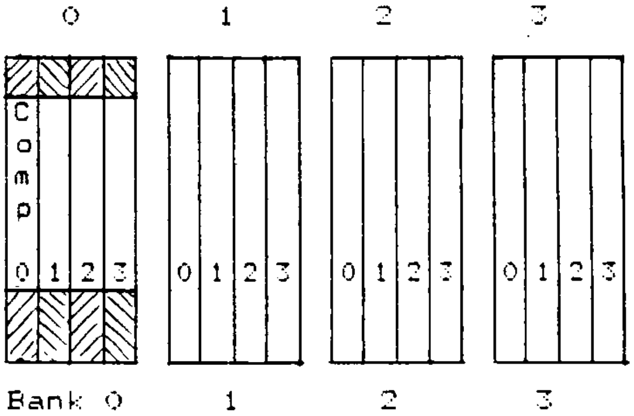


Abb.5 : Die Speicherkonfiguration des Genie III s



a) im Originalzustand





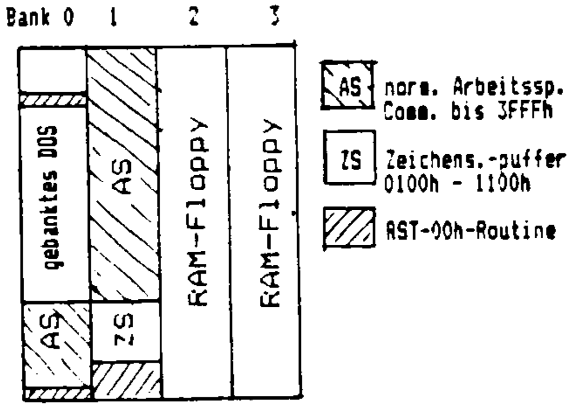
b) bei 1 Computer mit 1 MB RAM

c) bei 4 Computern mit je 256 kB RAM

Abb.6 : Die Adressierung der 16 64-kB-Blocks

Port F9h	....	00xxxxxx	01xxxxxx	10xxxxxx	11xxxxxx
Port F3h	xxxxxx00	0	1	2	3
	xxxxxx01	4	5	6	7
	xxxxxx10	8	9	A	B
	xxxxxx11	C	D	E	F

Abb.7 : Die Speicherbelegung pro Computer (auch im Originalzustand)



B: Die Software zur Umschaltung zwischen vier Rechnern

Während dieser Text entsteht, lauert im selben Gehäuse, aber dennoch in einem zweiten Computer ZEUS mit dem Quelltext des Programms, das nun vorgestellt werden soll. So kann während der Arbeit mal eben hinübergeschaltet werden, um vergessene Einzelheiten ins Gedächtnis zurückzurufen. Und damit ist auch schon gesagt, was das Ganze soll. Wir sozial Schwachen, die sich nur einen Rechner leisten können, brauchen auf den Komfort mehrerer Maschinen nun nicht mehr zu verzichten.

Da gibt es Blocks, RAM-Reihen, Banks, Adreßräume usw.. Um die Verwirrung in Grenzen zu halten, schlagen wir folgende Sprachregelung vor: Ein Block ist eine bestimmte Datenmenge, die gleichzeitig bearbeitet wird. Eine RAM-Reihe sind acht Speicherchips 41256, die über Port F9h, Bits 6 und 7 gleichzeitig selektiert sind. Funktional gesehen ist es sinnvoll, sie als Banks zu bezeichnen. Die Speicherviertel innerhalb einer Reihe von 41256-RAMs, um die es in diesem Beitrag geht, sollen hier Computer genannt werden, denn das Programm verwirklicht das Konzept von vier Rechnern, die einander auf Knopfdruck abwechseln.

Dieser Knopfdruck sollte nicht aus Versehen erfolgen können. Deshalb wird mit der Kombination von Shift-CTRL und einer Taste von 0-3 im Ziffernblock umgeschaltet. Dazwischen liegen rd. 35 cm, so daß Zufälle ausgeschlossen sind. Gleichwohl wäre ein solches Versehen nicht tragisch, denn eine Umschaltung ist sowohl mit, wie auch ohne Absicht erlaubt und ungefährlich. Alle Speicherinhalte bleiben bestehen, sogar die restliche Blinkzeit eines Cursors wird auf die Nanosekunde genau beim Zurückschalten nachgeholt. Der Tastendruck wird während der Interrupt-Bearbeitung erkannt. Hierzu wird ein CALL nach 45BEh (Erkennen der Dreitastenbefehle) an der Stelle 4601h umgeleitet (Z. 490 im Listing).

Das alte Lied: Wohin mit Erweiterungen des Betriebssystems? Diesmal mußte ein Speicherbereich dran glauben, der von TCS bei der Modifikation des Microsoft-Interpreters offenbar vergessen wurde. Ab 0000h wird weder nach dem Einschalten noch bei einem Reset gelesen, denn dann ist das Boot-EPROM aktiviert. In diesem Speicher-Niemandsland wird vom Programm erkannt, ob Shift-CTRL-Ziffer gedrückt wurde. Dazu stehen aber nur 8 Bytes bis zum Beginn der RST-08h-Routine zur Verfügung. Deshalb verabschiedet sich das neue RST 00h an der Stelle 0007h kurzerhand aus der Bank 0 und macht in Bank 1 weiter, wo das Betriebssystem erst wieder bei 0100h (Zeichensatz-Puffer) Platz beansprucht.

Diese Umschaltung geschieht mit dem Bit 0 von Port F9h. Da die normale Arbeitsbank Bank 1 ist, wobei Common unten in Bank 0 liegt, ist der Trick simpel: Das Common-Bit wird auf 1 gesetzt, so daß die CPU, ohne es zu merken, nun auch unten in Bank 1 liest, denn: Common, d. h. der Bereich, wo immer in Bank 0 gearbeitet wird, liegt jetzt ab E000h. Im Listing hat dieses Programmsegment das Label rst00 (Z. 664). Es liegt sowohl in Bank 0 als auch in Bank 1 bereit, so daß die Umschaltung aus beiden Banks auf die gleiche Weise geschieht. Es wird einfach D0 des Ports F9h invertiert.

Beim Umschalten aus Bank 0 (Common unten) wird die scheinbar nahtlose Fortsetzung in Bank 1 gefunden (Label bank1). Beim Zurückschalten aber muß rechtzeitig vor dem Erreichen der anschließenden RST-08h-Routine zur Fortsetzung der INT-Serviceroutine gesprungen werden. Deswegen wurde zuvor in Bank 1 HL mit 45BEh geladen, der Adresse, wo es weitergehen soll (s. o.). In Z. 495 wurde zu diesem Zweck der Befehl JP (HL) (Code E9h) in Bank 0 ans Ende dieses Segments gepatcht.

Nachdem HL die Adresse der Fortsetzung enthält, wird die Tastatur in 3880h und 38C0h nach den Tasten Shift und CTRL sowie nach den Ziffern 0-3 im Dezimalblock befragt. Wenn diese Tastenkombination nicht gedrückt wurde, will der User offenbar nicht den Computer wechseln. Es erfolgt ein

Rücksprung nach 0000h, wo wieder die alte Bank-Konfiguration eingestellt wird. Es ist, als wäre nichts geschehen.

Wurde aber Shift-CTRL-Ziffer eingegeben, wird zunächst der Ziffern-code in einen Zähler für die gewünschte Computer-Nr. umgerechnet und nach B geladen. Da sich die vier Computer außer der CPU natürlich auch alle Ports teilen müssen, werden die Zustände der beiden einzig wirklich lebenswichtigen, FAh und F9h, zwischengepuffert, jener sofort, dieser etwas später. Bei dieser Gelegenheit wird auf FAh das Bit 5 gesetzt, um einen Schreibschutz für den Adreßbereich des ehemaligen TRS-80-ROMs zu erreichen. Es könnte ja sein, daß ein Programm ihn als Sektorpuffer o. ä. benutzt.

In der Folge bekommt die Umschaltroutine auch ihren eigenen Stack. Das muß sein, weil er z. B. bei BASIC im Himem liegt, wo ja jetzt die Bank 0 selektiert ist; der Stack wäre unerreichbar! Der alte Stapelzeiger wird gepuffert. Je nach dem kurz zuvor aufgesetzten Computer-Zähler wird so oft zum Ausgangswert -15h (=+EBh) 15h addiert, bis die Nr. des zuständigen Computers im Akku steht.

Die Zahl 15h ergibt sich folgendermaßen: Beispielsweise bei der Selektion von RAM 1, Bildschirm und Zeichensatz Nr. 1 und Graphikspeicher Nr. 1 liegt am Port F3h der Wert xx01.0101b an (xx sind die beiden Steuerbits, die keinem eigenen Computer zugeordnet sind). Abgesehen von den Bits 6 und 7 ist das die Zahl 15h. Um einen Computer höher zu gehen, wird deshalb 15h addiert. Der so gewonnene Output für den Port F3h wird an die Adresse mit dem Label newcomp gepatcht.

Jetzt müssen alle Register gerettet werden, weil die benachbarten Computer nicht jeder eine eigene CPU haben. Die INT-Routine hat die Hälfte dieser Arbeit schon erledigt, so daß nur noch die Indexregister und der alternative Registersatz gepushed zu werden brauchen.

Nun kann umgeschaltet werden. Das Bit 7 wird zuvor gesetzt, damit bei Common-Einstellung nicht in die Bank 0 des Computers Nr. 0 zurückgeschaltet wird. Modifikationen, die der User oder das System dort vorgenommen hat, sollen ja nicht für alle vier Computer gelten.

Nach dem Umschalten auf einen Nachbarcomputer wird alles wieder restauriert. Es beginnt mit dem Output auf den Port F9h, der aber zunächst noch in E zwischengelagert wird. Zur Restauration des Bildschirmformats (auch den Video-Controller gibt es nur einmal) müssen nämlich noch im oberen Common-Bereich, der ja z. Zt. noch selektiert ist, die CRTC-Parameter aus 37F0h-37FFh ausgelesen und auf den Controller ausgegeben werden.

Diese Video-Codes liegen nun wiederum im unteren Common. Also hilft nur eine Umschaltung auf Bank 0, die beide Common-Bereiche gleichzeitig zugänglich macht. Daher der Umweg Uber das Unterprogramm vidpar, das die komplette Bank 0 verwalten kann. Zur Erinnerung: Der Arbeitsspeicher ist die Bank 1, daher diese Umstände, vidpar macht sich die Arbeit leicht, indem es ein fertiges Unterprogramm des Betriebssystems ab 3551h aufruft, das die Codes an den Video-Controller ausgibt. Zuletzt schaltet es wieder auf Bank 1, Common oben und kehrt zurück.

Jetzt werden auch der Port FAh, die CPU-Arbeitsregister und der Stackpointer restauriert. Wohlgemerkt sind es die Registerinhalte, die der Nachbarcomputer irgendwann zuvor für sich selber auf den eigenen Stack der Umschaltroutine gelegt hat. Was soeben gerettet wurde, liegt im alten Computer sicher. Mit dem abschließenden Sprung nach 0000h (Label rst00) wird Common wieder nach unten verlegt und die INT-Routine - diesmal diejenige des neuen Computers - zuende bearbeitet.

Außer der Umschaltung mit dem komplizierten Tastendruck geht es auch durch Aufruf des Programms mit seinem Namen. Um dabei nicht jedesmal die

Floppy rotieren zu lassen, ist es hier eine Systemdatei, so daß sie gleich beim Booten in den Speicher geladen wird. Sie erledigt verschiedene Aufgaben:

Bevor die Umschaltroutine arbeiten kann, sind etliche Initialisierungen notwendig. Hauptsächlich muß natürlich der Computer Nr. 0 auf die drei Nachbarn kopiert werden, damit die CPU nach dem Umschalten sofort sinnvollen Code lesen kann. Bei Testläufen hat es sich erwiesen, daß nur vom Computer 0 aus die Kopie stattfinden darf. Es könnte ja sein, daß der User ein wenig damit herumexperimentieren möchte oder sein digital bereits ebenso verseuchtes Kind nicht weiß, daß Mama oder Papa schon den Rechner wechselte. Deshalb wird bald nach dem Einsprung (Label test0) überprüft, ob die Bits 0 und 1 unseres neuen Systembytes 2 (zuständig für den Hauptspeicher) beide auf 0 stehen. Ist das nicht der Fall, wird einfach nur der derzeit selektierte Computer angezeigt. Diese Anzeige kann auch mit einem dem Aufruf folgenden Fragezeichen erhalten werden.

Ein weiteres Argument kann dem Befehl folgen: Wenn eine Ziffer von 0-4 eingegeben wurde, wird der so angewählte Computer selektiert. Allerdings nur dann, wenn zuvor die Initialisierung stattgefunden hat. Das merkt das Programm daran, ob an der Stelle 0000h in der Bank 1 bereits der Befehl DI steht. Ist das nicht der Fall, wird mit einer Fehlermeldung das Unternehmen abgebrochen. Wenn aber alles klar ist, ergibt die eingegebene Ziffer bereits die gewünschte Computer-Nr.. Es kann daher in Bank 1 an der Stelle switch mitten in die RST-00h-Routine gesprungen werden, wobei die Ermittlung des Computers und - sehr wichtig! - die Tastaturabfrage übersprungen werden.

Zusätzlich gibt es folgende Argumente: Wie in G-DOS üblich, kann das Ein- und Ausschalten des Features der vier Computer mit J (für "ja") und N (für "nein") erfolgen. J wird auch angenommen, wenn kein Argument folgt. N kann in jedem der vier Computer eingegeben werden. Dann bleibt immer dasselbe Viertel der 41256-RAMs selektiert, um in diesem Computer beispielsweise eine unverfälschte INT-Routine zu Testzwecken zu haben. Dennoch kann mit SYS,BK,n (n = Nr. des Computers) umgeschaltet werden; nur die Tastenkombination Shift-CTRL-n ist deaktiviert. Und dies gilt auch nur für den gerade selektierten Computer. Mit R (für "recover") werden die vier Computer reaktiviert, wenn ein zwischenzeitliches Reset oder ein SYS,BK,N es abschaltete. Auch hier muß zuvor bereits die Initialisierung erfolgt sein, sonst wird ein Fehler gemeldet.

Falls die Initialisierung der Nachbarcomputer gewünscht wird und erlaubt ist, wird’s jetzt hektisch: Im Verlauf der nächsten Mikrosekunden werden der Stackpointer, das Systembyte 0 (Port F9h) und das Systembyte 1 (Port FAh) als Initialwerte in die RST-00h-Routine gepatcht. Deren kürzerer Teil, der in Bank 0 liegen soll, wird nach 0000h verschoben. Das bereits erwähnte Byte E9h für JP (HL) kommt dahinter. Durch verschiedene Manipulationen der beiden Systembytes 0 und 1 werden Bildschirme aus- und wieder angeknipst, expandiert und wieder komprimiert, Tastaturen abgewürgt, Zeichensätze freigegeben, Common-Bereiche umdefiniert und dergleichen mehr. Es würde zu weit führen, das hier im einzelnen zu erläutern. Die Kommentare im Listing protokollieren es jeweils. Eine Hilfe zum Verständnis kann auch die Tabelle am Ende dieses Abschnitts geben.

Die Kopie des Hauptspeichers kann nicht auf einen Schlag geschehen. Das liegt daran, daß jeweils ein Common-Bereich erforderlich ist, der immer in der Bank 0 arbeitet. Das Feature, das mit Q6 des Latches ermöglicht werden sollte, funktionierte ja leider nicht. Soll also beispielsweise im Common-Adreßraum gerade die Bank 2 kopiert werden, ist sie einfach nicht da. Daher wird das Unterprogramm COPY, das später erklärt werden soll, sowohl in das obere als auch in das untere Common kopiert. Die beiden Routinen bearbeiten dann jeweils nur eine Speicherhälfte von viermal 32 kB. Mit dem oben residierenden Kopierprogramm wird gleich VIDPAR mitverschoben.

Zuerst wird der Bildschirm vervierfacht. Dazu muß er zunächst in einen Puffer im unteren Adreßraum der Bank 1 kopiert werden. Er findet Platz hinter dem Zeichensatzpuffer ab 1100h. Beim Schreiben des gewählten Zeichensatzes in die drei benachbarten Zeichengeneratoren steht das Adreß-LSB des gerade bearbeiteten Zeichens im Bildschirm an der Stelle 3C00h. Um den dadurch veränderten Bildschirm nicht umständlich restaurieren zu müssen, erfolgt zuerst die Kopie des Zeichensatzes, erst anschließend wird der Bildschirminhalt aus dem Puffer in den Bildwiederholspeicher geladen. Das Zeichen von 3C00h wurde im Puffer ja nicht verändert. Die Auswahl des gerade aktuellen Bildschirms bzw. Zeichensatzes geschieht Über die Bits 2 und 3 des Ports F3h.

Um die beiden Kopierroutinen aufrufen zu können, ist ab Z. 576 wieder einiges Banking erforderlich. Auch hier mag der Verweis auf die Kommentare und die Tabelle ausreichen. Die Routinen selbst arbeiten folgendermaßen :

Am Label copy wird ein Zähler mit 32 aufgesetzt. Die Datenmenge, für die der erforderliche Puffer ausreicht, beträgt 1 kB. 32 kB pro Bank übernimmt eins der beiden Kopierprogramme. Gleich anschließend wird dieser Zähler gerettet und derjenige für die Banks geladen. Jetzt wird aus jeder Bank des Computers Nr. 0 ein Block zu 1 kB in den Puffer geholt. Schließlich wird in der inneren Schleife, welche die drei Nachbarcomputer abzählt, der Pufferinhalt auf die Nachbarn kopiert. Um die Adresse des zu bearbeitenden 1-kB-Blocks zu aktualisieren, genügt es, ihr MSB um 4 zu erhöhen (ab Z. 654), denn 0400h = 1 kB.

Um die Adressen beim Blocktransferbefehl LDIR nicht zu zerstören, müssen sie fortlaufend gerettet und wieder restauriert werden. Neben den LDIR-Befehlen selbst sind es diese PUSHes und POPs, die beim Kopieren die meiste Zeit verbrauchen. Insgesamt läuft das Programm deshalb etwa dreieinhalb Sekunden, bei immerhin einem Megabyte eine erträgliche Zeit. Um möglichst schnelle 8-Bit-Befehle anwenden zu können, werden nur die MSBs geladen. Die LSBs betragen immer 00h, so daß sie nicht berücksichtigt werden müssen. Diese - im Computer-Maßstab - extrem langwierige Prozedur läuft jedoch nur nach dem Aufruf des Programms ab. Nach der Initialisierung geschieht das Umschalten zwischen den Computern hundertmal schneller als ein Blinzeln.

Nach den Kopiervorgängen bleibt nur noch übrig, die Systembytes 0 und 1 sowie den Stackpointer zu restaurieren. Mit RET wird ins DOS zurückgekehrt.

Wie gesagt, müssen sich die vier Computer die gesamte Hardware teilen, lediglich die Speicherbausteine haben sie für sich. Der User muß deshalb wissen, daß gewisse Einschränkungen in Kauf zu nehmen sind. So ist es beispielsweise nicht möglich, mit der seriellen Schnittstelle auf Computer Nr. 0 eine Mailbox anzurufen und zwischendurch damit einen seriellen Drucker zu bedienen. Die Schnittstelle wird beim Umschalten nicht berücksichtigt. Dasselbe gilt für die Floppies, die PIO und evtl, angeschlossene Peripherie auf dem Systembus. Dennoch können alle Bauteile von allen Computern benutzt werden. Es darf eben nur nicht umgeschaltet werden, wenn sich z. B. gerade die Floppy dreht.

In der hier vorgestellten Form ist das Programm Teil der Systemdatei SYS25/SYS. Dies erschien sinnvoll, weil es ganz entschieden ins Betriebssystem gehört und notfalls auch unter Mini-DOS mitten in einem Programmablauf abrufbar sein sollte. Da die Library des Genie III s schon voll ist, wird es mit dem SYS-Befehl (hier SYS,BK für "BANK") aktiviert, der an dieser Stelle bereits vorgestellt wurde. So erklärt sich das Vorgeplänkel am Beginn des Listings, wo zum alten Inhalt von SYS25/SYS (Bedienung der F-Tasten) der Test auf den neuen Requestcode DBh für SYS,BK eingefügt wurde. Was zwischen den Zeilen 27 und 405 steht, ist mit LIST OFF ausgespart, denn die F-Tasten sind hier nicht unser Thema. Dies

gilt auch für die Label-Definitionen bis Z. 22.

Es gibt keine Einschränkungen des Speicherplatzes. Nach dem Laden steht SYS25/SYS zunächst im DOS-Overlay-Bereich ab 4D00h, der für diese Zwecke vorgesehen und reserviert ist. Die Bereiche, die die RST-00h-Routine und das Unterprogramm vidpar belegen, sind völlig frei und werden weder vom Betriebssystem noch von den Autoren bekannten Anwenderprogrammen beansprucht. Für das RAM-Floppy-Programm MEMDISK/CMD gilt sogar, daß es vor dem Aufruf bereits aktiviert sein darf. Es wird nicht Uberschrieben, auch nicht die Dateien, die in der RAM-Floppy stehen. Alles wird einfach in die Nachbarcomputer hinüberkopiert und kann auch von dort aus benutzt werden.

Wie sich herausstellte, kann aber leider nicht unter BASIC CMD'SYS,BK<,J>' eingegeben werden, um BASIC/CMD solo oder gar mit bereits geladenem Programm in die drei Nachbarcomputer hinüberzukopieren. Vielleicht gelingt es in der Folge, das ebenfalls noch zu ermöglichen. Einstweilen muß es unter DOS geschehen, wonach dann in jedem der vier Computer BASIC/CMD gesondert gestartet werden muß. Auf jeden Fall aber kann nach erfolgter Initialisierung mitten im Lauf eines BASIC-Programms auf einen Nachbarcomputer umgeschaltet werden. Mit anderen Programmen sind bisher diese Schwierigkeiten nicht aufgetreten. Im Zweifelsfall hilft immer die Methode, die eben für BASIC beschrieben wurde.

Eine weitere Besonderheit muß beim Betrieb mit BASIC beachtet werden. In der RST-00h-Routine kehrt der Akku mit einem Wert >0 zurück (Output für den Port F9h). Das Z-Flag ist rückgesetzt, d. h. die NZ-Bedingung ist gegeben. BASIC versteht das als DOS-Fehler, gibt eine entsprechende Meldung aus und sagt ‘READY’. Deshalb kann man zwar ohne weiteres den Befehl CMD"SYS,BK,n' (n ist die Computer-Nr.) über die Tastatur eingeben, wenn es aber innerhalb eines Programms geschieht, wird dieses wegen des angeblich auf getretenen Fehlers abgebrochen. Das wäre mit geringen Modifikationen zu beheben gewesen, die aber Speicherplatz im kostbareren oberen Adreßraum gekostet hätten. So wurde darauf verzichtet, denn eine Umschaltung, die ein BASIC-Programm selbsttätig steuert, ist praxisfern.

Das Verändern von SYS-Files und das Implementieren eines neuen Library-Befehls (auch in der Zweit-Lib mit SYS) wurde hier schon mehrfach gezeigt. Es soll nicht erneut erklärt werden. Wer sich scheut, auf diese Weise ins Betriebssystem einzugreifen, kann das Programm in der Zeile 421 mit ORG 5200h beginnen und ein CMD-File daraus machen. Dann muß das END-Statement die Startadresse sidekck haben. Ansonsten ändert sich dabei nichts.

Zuletzt sollen noch diejenigen Datenbits der hier erwähnten Ports erläutert werden, die für das Programm von Belang sind:

	Bit 7	6	5	4	3	2	1	0
F3h	Common-	.	=HRG-Nummer=		=Bildsch.- u. =		=Hauptsp.-Nr.=	
	Defin.	.	.	.	=Zeichens.-Nr.=		.	.
F9h	=Orig.-Banks=		.	.	.	.	Zeichen-	Common-
	.	.	.	.	.	.	satz	Defin.
FAh	.	.	Schreib-	Video	.	.	.	mm-I/O
	.	.	schutz	ab 3800h	.	.	.	an/aus

Sollte es mit der Software zur Verwirklichung von 4 Computern in einem Schwierigkeiten geben, kann Arnulf Sopp weiterhelfen.

```
4D00                00023          ORG     4d00h
4D00    FEFB        00024   sys25  CP      0fbh         ;Lib-Befehl F# ?
4D02    2822        00025          JR      Z,fkey       ;falls ja
4004    FEDB        00026          CP      0dbh         ;Lib-Befehls SYS,BK ?
4D06    CA2A50      00027          JP      Z,sidekck    ;falls ja
                    00405
                    00406
                    00407   ;***************************************************
                    00408   ;
                    00009   ;           Sidekick für das Genie IIIs
                    00010   ;           4 Computer in einem Gehäuse
                    00411   ;
                    00412   ;       Hardware: 1 MB RAM, 8 kB Bildschirm,
                    00413   ;       16 kB Zeichensatz, 256 kB Graphik
                    00414   ;
                    00415   ;       (C) 1986 by Arnulf Sopp
                    00416   ;       starring:   Helmut Bernhardt
                    00417   ;
                    00418   ;***************************************************
                    00419
                    00420   ;Test, ob vier Computer bereits initialisiert sind
5015    DBF9        00421   tstinit IN     A,(0f9h)        ;Systembyte 0
5017    57          00422          LD      D,A             ;retten
5018    F601        00423          OR      01h             ;unten Bank 1 selektieren
501A    F3          00424          DI                      ;vorsichtshalber
501B    D3F9        00425          OUT     (0f9h),A
501D    3A0000      00426          LD      A,(0000h)       ;Anf. der RST-00-Routine
5020    FEF3        00427          CP      0f3h            ;steht dort bereits DI ?
5022    7A          00428          LD      A,D             ;Systembyte 0
5023    D3F9        00429          OUT     (0f9h),A        ;restaurieren
5025    C8          00430          RET     Z               ;o. k., falls schon DI
5026    E1          00431          POP     HL              ;RET-Adresse vom Stack
5027    C3224D      00432   jperr  JP      error           ;raus mit Fehlerbeding.
                    00433
                    00434   ;Tests, ob Befehl mit Argument eingegeben wurde
502A    DBF3        00435   sidekck IN     A,(0f3h)            ;Systembyte 2
502C    E603        00436          AND     03h             ;nur Hauptspeicher-Bits
502E    4F          00437          LD      C,A             ;retten
502F    CDD54C      00438          CALL    4cd5h           ;folgt etwas dem Befehl?
5032    2841        00439          JR      Z,test0         ;falls nein
5034    7E          00440          LD      A,(HL)          ;was folgt?
5035    FE52        00441          CP      'R'             ;Recover 4 Computer?
5037    200A        00442          JR      NZ,norecov      ;falls nein
5039    CD1550      00443          CALL    tstinit         ;schon initialisiert?
503C    3EC9        00444          LD      A,0c9h          ;RET-Befehl
503E    329150      00445          LD      (retrecv),A     ;dort patchen
5041    1838        00446          JR      init            ;dort weiter
5043    FE4A        00447   norecov CP     'J'             ;'ja' für initialisieren?
5045    282E        00448          JR      Z,test0         ;falls ja
5047    FE4E        00449          CP      'N'             ;'nein' für abschalten?
5049    2007        00450          JR      NZ,notoff       ;falls nicht
504B    21BE45      00451          LD      HL,45beh        ;alte Adr. in   INT-Routine
504E    220146      00452          LD      (4601h),HL      ;dort restaurieren
5051    C9          00453          RET                     ;dann Ende
5052    FE3F        00454   notoff CP      '?'             ;Status-Abfrage?
5054    2819        00455          JR      Z,dspcmp1       ;falls ja
5056    FE30        00456          CP      'O'             ;Computer-Nr.eingegeben?
5058    38CD        00457          JR      C,jperr         ;falls falsche  Eingabe
505A    FE34        00458          CP      '4'             ;nur Comp. Nr.  0-3
505C    30C9        00459          JR      NC,jperr        ;falls falsche  Eingabe
                    00460
                    00461   ;neuen Computer selektieren, falls schon initialisiert
505E    E603        00462          AND     03h             ;ASCII -> binär 0-3
5060    3C          00463          INC     A               ;für Zähler B:  1-4
```

```
5061   47                LD      B,A              ;Computerzähler aufsetzen
5062   CD1550            CALL    tstinit          ;schon initialisiert?
5065   E1                POP     HL               ;RET-Adresse
5066   DBF9              IN      A,(0f9h)         ;Systembyte 0
5068   F601              OR      01h              ;Common oben, Bk. 1 unten
506A   D3F9              OUT     (0f9h),A         ;Systembyte neu ausgeben
506C   C31E00            JP      switch-rst00     ;dort Computer umschalten

                 ;aktuelle Computer-Nr. anzeigen
506F   79        dspcmp1 LD      A,C              ;Computer-Nr.
5070   F630      dspcmp2 OR      30h              ;binär -> ASCII
5072   C33300            JP      0033h            ;anzeigen und raus

                 ;falls Comp. Nr. 0, dann initialisieren, sonst anzeigen
5075   79        test0   LD      A,C              ;Computer-Nr.
5076   B7                OR      A                ;Nr. 0 selektiert?
5077   20F7              JR      NZ,dspcmp2       ;nur anzeigen, falls nein
5079   D3F3              OUT     (0f3h),A         ;auch Video Nr. 0 usw.

                 ;INT-Patch in SYS0, RST-00-Routine für Bank 0 verschieben
507B   ED73D751          init    LD  (spbuf1),SP ;Initialstack
507F   216B51            LD      HL,rst00         ;RST-00h-Routine
5082   010700            LD      BC,bank1-rst00  ;Länge der Rout, in Bk. 0
5085   50                LD      D,B              ;DE <- Zieladresse 0000
5086   58                LD      E,B
5087   F3                DI                       ;DISABLE INTERUPT
5088   ED530146          LD      (4601h),DE       ;CALL 0000 statt 45BE
508C   EDB0              LDIR                     ;RST-00-Routine ->  Bank 0

                 ;Befehl zum Weitersprung schließt RST-00-R. in Bank 0 ab
508E   3EE9              LD      A,0e9h           ;Befehl JP (HL)
5090   12                LD      (DE),A           ;dahinter patchen (0007h)

                 ;Rout. zur Kopie der ob. Speicherhälfte nach 3800 laden
5091   DBFA      retrecv IN      A,(0fah)         ;Systembyte 1
5093   F5                PUSH    AF           ;   retten
5094   32C851            LD      (fabuf),A        ;Patch in RST 00, Bank 1
5097   F611              OR      11h              ;Großbildsch. , mm-I/O aus
5099   D3FA              OUT     (0fah),A         ;Systembyte 1 neu ausgeb.
509B   08                EX      AF,AF’           ;nochmals retten
509C   213751            LD      HL,copy          ;Kopierprg. f. ob. Hälfte
509F   110038            LD      DE,combot        ;im unt. Common-Bereich
50A2   0E34              LD      C,rst00-copy     ;Länge der Routine
50A4   EDB0              LDIR                     ;nach 3800 übertragen

                 ;Routine zur Restauration der Bildschirmparameter und
                 ;zur Kopie der unteren Speicherhälfte nach F600 laden
50A6   DBF9              IN      A,(0f9h)         ;Systembyte 0
50A8   F5                PUSH    AF               ;retten
50A9   F601              OR      01h              ;Common oben, Bank 1 unt.
50AB   F5                PUSH    AF               ;nochmals retten
50AC   32BD51            LD      (f9buf),A        ;Patch in RST 00, Bank 1
50AF   D3F9              OUT     (0f9h),A         ;geänd. Systemb. ausgeben
50B1   212E51            LD      HL,vidpar        ;CRTC-Restaur., Kopierr.
50B4   1100F6            LD      DE,comtop        ;im oberen Common-Bereich
50B7   0E3D              LD      C,rst00-vidpar  ;Länge der Routine
50B9   EDB0              LDIR                     ;nach F600 kopieren

                 ;RST-00-Routine für Bank 1 in diese Bank nach 0000 laden
50BB   50                LD      D,B              ; DE <- 0000h, wie oben
50BC   58                LD      E,B              ;(Zieladr. RST-00-Rout.)
50BD   0E70              LD      C,endrst00-rst00    ;Länge RST-00-R.
50BF   EDB0              LDIR                     ;RST-00-Routine -> Bank 1
```

```
                    00528   ; memory-mapped I/O einschalten, Zeichen-RAM freigeben
50C1    F602        00529           OR      02h             ;Zeichensatz freigeben
50C3    D3F9        00530           OUT     (0f9h),A
50C5    08          00531           EX      AF,AF’          ;geändertes Systembyte 1
50C6    F5          00532           PUSH    AF              ;wird noch gebraucht
50C7    E6FE        00533           AND     0feh            ;mm-I/O einblenden
50C9    D3FA        00534           OUT     (0fah),A        ;Systembyte 1 neu ausgeb.
                    00535
                    00536   ;Großbildschirm ab 3800 in Bank 1 ab 1100 zwischenspeich.
50CB    210038      00537           LD      HL,3800h        ;Anfang Bildschirm
50CE    110011      00538           LD      DE,1100h        ;Anfang Puffer
50D1    0608        00539           LD      B,08h           ;BC <- Länge 2 kB
50D3    EDB0        00540           LDIR                    ;Bildschirm puffern
                    00541
                    00542   ;Zeichensatz in 3 neue Zeichengeneratoren kopieren
                    00543   ;(3 obere Viertel der beiden 6264)
50D5    0603        00544           LD      B,03h           ;3 Bildsch. u. Zeichens.
50D7    C5          00545   vidloop PUSH    BC              ;Bildschirmzähler retten
50D8    78          00546           LD      A,B             ;Bildschirm-Nr.
50D9    07          00547           RLCA                    ;Bit 0-1 -> 2-3
50DA    07          00548           RLCA
50DB    D3F3        00549           OUT     (0f3h),A        ;Bildschirmseite selekt.
50DD    2601        00550           LD      H,01h           ;HL <- Beginn Zeichensatz
50DF    AF          00551           XOR     A               ;ab ASCII 00
50E0    32003C      00552   chrlop1 LD      (3c00h),A       ;LSB Zeichenadresse
50E3    08          00553           EX      AF,AF’          ;retten
50E4    1684        00554           LD      D,84h           ;MSB in D
50E6    0610        00555           LD      B,10h           ;16 Bytes pro Zeichen
50E8    7E          00556   chrlop2 LD      A,(HL)          ;1 Byte aus dem Puffer
50E9    12          00557           LD      (DE),A          ;ins Zeichensatz-RAM
50EA    23          00558           INC     HL              ;auf nächstes Byte
50EB    7A          00559           LD      A,D             ;Adreß-MSB
50EC    C608        00560           ADD     A,08            ;um 2 kB erhöhen
50EE    57          00561           LD      D,A             ;neues MSB
50EF    10F7        00562           DJNZ    chrlop2         ;bis 1 Zeichen eingelesen
50F1    08          00563           EX      AF,AF’          ;LSB Zeichenadresse
50F2    3C          00564           INC     A               ;nächster ASCII-Wert
50F3    20EB        00565           JR      NZ,chrlop1      ;bis 256 Zeichen eingel.
                    00566
                    00567   ;Bildschirm aus dem Puffer in 3 neue Video-Speicher kop.
                    00568   ;(3 obere Viertel des einzelnen 6264)
50F5    1638        00569           LD      D,38h           ;Bildschirmadresse
50F7    0608        00570           LD      B,08h           ;BC <- Bildschirmlänge
50F9    EDB0        00571           LDIR                    ;Bi1dschirmseite füllen
50FB    C1          00572           POP     BC              ;Bildschirmzähler
50FC    10D9        00573           DJNZ    vidloop         ;bis 4 Bildschirme fertig
                    00574
                    00575   ;Vorber. d. Hauptspeicherkopien, Aufruf d. Kopierroutinen
50FE    F1          00576           POP     AF              ;Systembyte 1
50FF    D3FA        00577           OUT     (0fah),A        ;mm-I/0 ausblenden
5101    F1          00578           POP     AF              ;Systembyte 0
5102    D3F9        00579           OUT     (0f9h),A        ;Common oben, Bank 1 unt.
5104    60          00580           LD      H,B             ;untere Speicherhälfte
5105    16F7        00581           LD      D,comtop+256/256    ;DE <- Pufferadr.
5107    ED732551    00582           LD      (spbuf2),SP     ;Stackpointer retten
510B    3100F7      00583           LD      SP,comtop+256   ;eig. Stack im ob. Common
510E    CD09F6      00584           CALL    copy+offset1    ;unteren Speicher kopier.
5111    212D40      00585           LD      HL,402dh        ;DOS-Ready als RET-Adr.
5114    E5          00586           PUSH    HL              ;auf den Initial-Stack
5115    E6FE        00587           AND     0feh            ;Common unten
5117    D3F9        00588           OUT     (0f9h),A        ;Systembyte 0 neu ausgeb.
5119    210080      00589           LD      HL,8000h        ;obere Speicherhälfte
511C    1630        00590           LD      D,30h           ;DE <- Pufferadresse
511E    310039      00591           LD      SP,combot+256   ;Stack im unteren Common
```

```
5121    CD0038    00592           CALL    copy-offset2     ;oberen Speicher kopieren
5124    310000    00593           LD      SP,0000h         ;Stackptr. restaurieren
5125              00594   spbuf2  EQU     $-2              ;(variabel)
5127    F1        00595           POP     AF               ;altes Systembyte 0
5128    D3F9      00596           OUT     (0f9h),A         ;restaurieren
512A    F1        00597           POP     AF               ;dto. Systembyte 1
512B    D3FA      00598           OUT     (0fah),A
512D    C9        00599           RET                      ;zurück i. Betriebssystem
                  00600
                  00601   ;Bildschirm-Parameter restaurieren (Arbeitsadresse F600)
F600              00602   comtop  EQU     0f600h           ;im oberen  Common-Bereich
A4D2              00603   offset1 EQU     comtop-$         ;für Kopie  d. unt. Hälfte
512E    D3F9      00604   vidpar  OUT     (0f9h),A         ;Bank 0 selektieren
5130    CD5135    00605           CALL    3551h            ;Videoparameter setzen
5133    7B        00606           LD      A,E              ;Systembyte 0
5134    D3F9      00607           OUT     (0f9h),A         ;restaurieren
5136    C9        00608           RET
                  00609
                  00610   ;Speicher von Computer Nr. 0 auf Nachbarcomputer kopieren
                  00611   ;(Arb.-adr. 3800 f. ob. Hälfte o. hinter vidpar f. unten)
3800              00612   combot  EQU     3800h            ;im unteren Common-Ber.
1937              00613   offset2 EQU     $-combot         ;für Kopie  der ob. Hälfte
                  00614
                  00615   ;32 Blocks zu je 1 kB in 3 Computer mit je 4 Banks kop.
5137    0620      00616   copy    LD      B,32             ;32 Blocks zu 1 kB kop.
5139    C5        00617   coplop1 PUSH    BC               ;Blockzähler retten
                  00618
                  00619   ;1 Block zu je 1 kB vor Kopie im Puffer Zwischenspeichern
513A    0604      00620           LD  B,4                  ;4 Banks
513C    C5        00621   coplop2 PUSH    BC               ;Bankzähler retten
513D    E5        00622           PUSH    HL               ;dto. Zeiger
513E    D5        00623           PUSH    DE
513F    0604      00624           LD      B,04h            ;0400 = 1 kB
5141    EDB0      00625           LDIR                     ;Block puffern
5143    E1        00626           POP     HL               ;Quelle und Ziel tauschen
5144    D1        00627           POP     DE
                  00628
                  00629   ;1 Block zu je 1 kB in 1 Bank von 3 Computern kopieren
5145    0603      00630           LD      B,3              ;Zähler 3 Nachbarcomputer
5147    C5        00631   coplop3 PUSH    BC               ;Computerzähler retten
5148    78        00632           LD      A,B              ;lfd. Computer-Nr.
5149    D3F3      00633           OUT     (0f3h),A         ;Computer selektieren
514B    0604      00634           LD      B,04h            ;1 kB
514D    E5        00635           PUSH    HL               ;Zeiger retten
514E    D5        00636           PUSH    DE
514F    EDB0      00637           LDIR                     ;1 Block in Nachbarcomp.
5151    D1        00638           POP     DE               ;Zeiger restaurieren
5152    E1        00639           POP     HL
5153    C1        00640           POP     BC               ;dto. Computerzähler
5154    10F1      00641           DJNZ    coplop3          ;bis 1 Bl. in 4 C., 1 Bk.
                  00642
                  00643   ;dasselbe in der nächsthöheren Bank
5156    AF        00644           XOR     A                ;Computer Nr. 0 selekt.
5157    D3F3      00645           OUT     (0f3h),A
5159    DBF9      00646           IN      A,(0f9h)         ;Systembyte 0
515B    C640      00647           ADD     A,40h            ;1 Bank höher
515D    D3F9      00648           OUT     (0f9h),A         ;selektieren
515F    EB        00649           EX      DE,HL            ;Zeiger rücktauschen
5160    C1        00650           POP     BC               ;Bankzähler
5161    10D9      00651           DJNZ    coplop2          ;b. 1 Bl. in 3 C., 4 Bks.
                  00652
                  00653   ;dasselbe für den nächsten Block der Speicherhälfte
5163    24        00654           INC     H                ;auf nächsten Block
5164    24        00655           INC     H                ;(= 0400 addieren)
```

```
5165    24              INC     H
5166    24              INC     H
5167    C1              POP     BC              ;Blockzähler
5168    10CF            DJNZ coplop1            ;bis 1 Memory-Hälfte kop.
516A    C9              RET                     ;zum Aufrufer

                ;RST-00h-Rout., Segment in Bank 0 u. 1 (Arbeitsadr. 0000)
                ;von Bank 0 auf Bank 1 und umgekehrt schalten
516B    F3      rst00   DI                      ;keine Störungen
516C    DBF9            IN      A,(0f9h)        ;Systembyte 0
516E    EE01            XOR     01h             ;Bank 0 <-> 1
5170    D3F9            OUT     (0f9h),A
                ;in Bank 0 an der Adresse 0007 folgt hier JP (HL)

                ;Fortsetzung in Bank 1 ab 0007
                ;INT-Forts. sichern, prüfen, ob Sh.-CTRL-Ziffer gedrückt
5172    21BE45  bank1   LD      HL,45beh        ;Forts, d. INT-Routine
5175    3A8038          LD      A,(3880h)       ;Tastatur Steuerzeichen
5178    FE03            CP      03h             ;Shift-CTRL gedrückt?
517A    20EF            JR      NZ,rst00        ;falls nein
517C    3AC038          LD      A,(38c0h)       ;Tastatur Ziffernblock
517F    E60F            AND     0fh             ;0  -   3   gedrückt?
5181    28E8            JR      Z,rst00         ;falls nein

                ;Tastendruck erfolgt, Zähler für Computer-Nr. aufsetzen
5183    0600            LD      B,00h           ;Anfangsw. Computer-Nr.
5185    04      getcomp INC     B               ;B hochzählen
5186    0F              RRCA                    ;Tastenbit ins Cy
5187    30FC            JR      NC,getcomp      ;falls noch nicht gefund.

                ; Systembytes, Stack und Adresse der Programmforts, retten
                ;(ab hier auch Bearbeitung mit Befehl SYS,BK,n)
5189    DBFA    switch  IN      A,(0fah)        ;Systembyte 1
518B    E6DF            AND     0dfh            ;Schreibschutz aufheben
518D    D3FA            OUT     (0fah),A        ;Systembyte neu ausgeben
518F    F620            OR      20h             ;Schreibschutz 0000-2FFF
5191    325D00          LD      (fabuf-rst00)   ;Systembyte 1 retten
5194    ED736C00        LD      (spbuf1-rst00),SP   ;SP retten
5198    3100F7          LD      SP,comtop+256   ;eigener Stack im Common
519B    E5              PUSH    HL              ;Fortsetzungsadresse
519C    DBF9            IN      A,(0f9h)        ;Systembyte 0
519E    325200          LD      (f9buf-rst00),A ;retten

                ;Nachbarcomp. errechnen (RAM, Video, Zeichens., HRG-RAM)
51A1    3EEB            LD      A,0ebh          ; -15,  Anfw. Comp. & Video
51A3    C615    getall  ADD     A,15h           ;Cmp., Vid. u. Grph. ab 0
51A5    10FC            DJNZ    getall          ;bis neuer Comp, errechn.
51A7    324A00          LD      (newcomp-rst00),A   ;dort patchen

                ;alle Alternativregister des bisherigen Computers retten
51AA    08              EX      AF,AF’
51AB    D9              EXX
51AC    DDE5            PUSH    IX
51AE    FDE5            PUSH    IY
51B0    F5              PUSH    AF
51B1    E5              PUSH    HL
51B2    D5              PUSH    DE
51B3    C5              PUSH    BC

                ;neuen Computer selektieren, dessen Videoparameter setzen
51B4    3E00            LD      A,00h           ;Output für Port F3
51B5            newcomp EQU     $-1             ;je nach Tastendruck
5186    E63F            AND     3fh             ;Steuerbits ausmaskieren
51B8    F680            OR      80h             ;Common nur comp.-intern
```

```
51BA    D3F3      00720           OUT     (0f3h),A        ;Nachbarcomputer    selekt.
51BC    1E00      00721           LD      E,00h           ;Systembyte 0 (kommt In
51BD              00722   f9buf   EQU     $-1             ;vidpar in den Akku)
51BE    21F037    00723           LD      HL,37f0h        ;dort CRTC-Parameter
51C1    0610      00724           LD      B,10h           ;16 Codes
51C3    AF        00725           XOR     A               ;ab CRTC-Register 0
51C4    CD00F6    00726           CALL    vidpar+offset1  ;Video-Attribute    restaur.
                  00727
                  00728   ;Systembytes, Register usw. restaur. und Programm forts.
51C7    3E00      00729           LD      A,00h           ;Systembyte 1
51C8              00730   fabuf   EQU     $-1             ;je nach dem
51C9    D3FA      00731           OUT     (0fah),A        ;restaurieren
51CB    C1        00732           POP     BC              ;alle Register restaur.
51CC    D1        00733           POP     DE
51CD    E1        00734           POP     HL
51CE    F1        00735           POP     AF
51CF    FDE1      00736           POP     IY
51D1    DDE1      00737           POP     IX
51D3    D9        00738           EXX                     ;in den Alternativsatz
51D4    08        00739           EX      AF,AF'
51D5    E1        00740           POP     HL              ;Fortsetzungsadresse
51D6    310000    00741           LD      SP,0000h        ;SP restaurieren
51D7              00742   spbuf1  EQU     $-2             ;(variabel)
51D9    1890      00743           JR      rst00           ;umschalten und zurück
                  00744   endrst00
                  00745
4D00              00746           END     sys25               ;dort Einsprung
00000 Fehler
```

```
10  REM: Testprogramm, beschränkt sich auf das Umschalten des Bildschirms.
20  REM: Die vier Bildschirme wechseln einander ab. Zuvor muß das   Initiali-
30  REM: sierungsprogramm mit SYS,BK<,J> vom DOS aus aufgerufen worden sein.
40  REM:
50  A=INP(&HF3):REM Zustand des neuen Systembytes 2
60  A=AAND&HF3:REM  Video-Bits  ausmaskieren
70  B=B+&H04:REM    Video-Bits  hochschalten
80  B=BAND&H0C:REM  höhere Bits nicht belangen
90  OUTSHF3,A+B:REM auf nächsten Bildschirm wechseln
100 F0RI=0T0100:NEXT:REM ein bißchen warten
110 G0TO60:REM  dasselbe Spielchen von vorne
```

Dasselbe in Assembler:

```
5200    00001     ORG             5200h
5200    DBF3      00002   Start   IN      A,(0f3h)    ;Systembyte 2
5202    C604      00003           loop    ADD A,04h   ;nächste Bildschirmseite
5204    E60F      00004           AND     0fh         ;nur bis Bit 3
5206    D3F3      00005           OUT     (0f3h),A    ;neu ausgeben
5208    08        00006           EX      AF, AF'     ;retten
5209    010080    00007           LD      BC,8000h    ;Zähler für Verzögerung
520C    0B        00008   delay   DEC     BC          ;herunterzählen
520D    78        00009           LD      A,B         ;prüfen, ob Zähler
520E    B1        00010           OR      C           ;schon abgelaufen
520F    20FB      00011           JR      NZ,delay    ;falls noch nicht
5211    08        00012           EX      AF,AF'      ;Systembyte 2
5212    18EE      00013           JR      loop        ;und von vorne
5200              00014   END     Start               ;dort Einsprung

00000 Fehler
```

C: Einbau des Erweiterungsboards und Anschluß der damit erzeugten / dafür benötigten Signale

Da die meisten der für diese Schaltung benötigten Signale direkt vom Z80 geliefert werden, war es zweckmäßig, ein Board zu entwerfen, das anstelle des Z80 in dessen Sockel gesteckt wird und den Prozessor selbst aufnimmt. Damit sind dann nur noch zehn Signale über freie Verdrahtung mit dem CPU-Board zu verbinden. Diese Leitungen sind an einen zweireihigen Pfostenstecker geführt. Die Signale COML* und COMH* werden von den Pins 12 und 13 von U84, 74LS00 abgenommen und in beliebiger Anordnung an die Pins 3 und 4 des Steckers geführt. MWR* an Pin 5 des Steckers ist von Pin 8 von U52 (74LS32) zu beziehen, SYSRES* an Pin 6 des Steckers kommt von Pin 13 von U68, 74LS74.

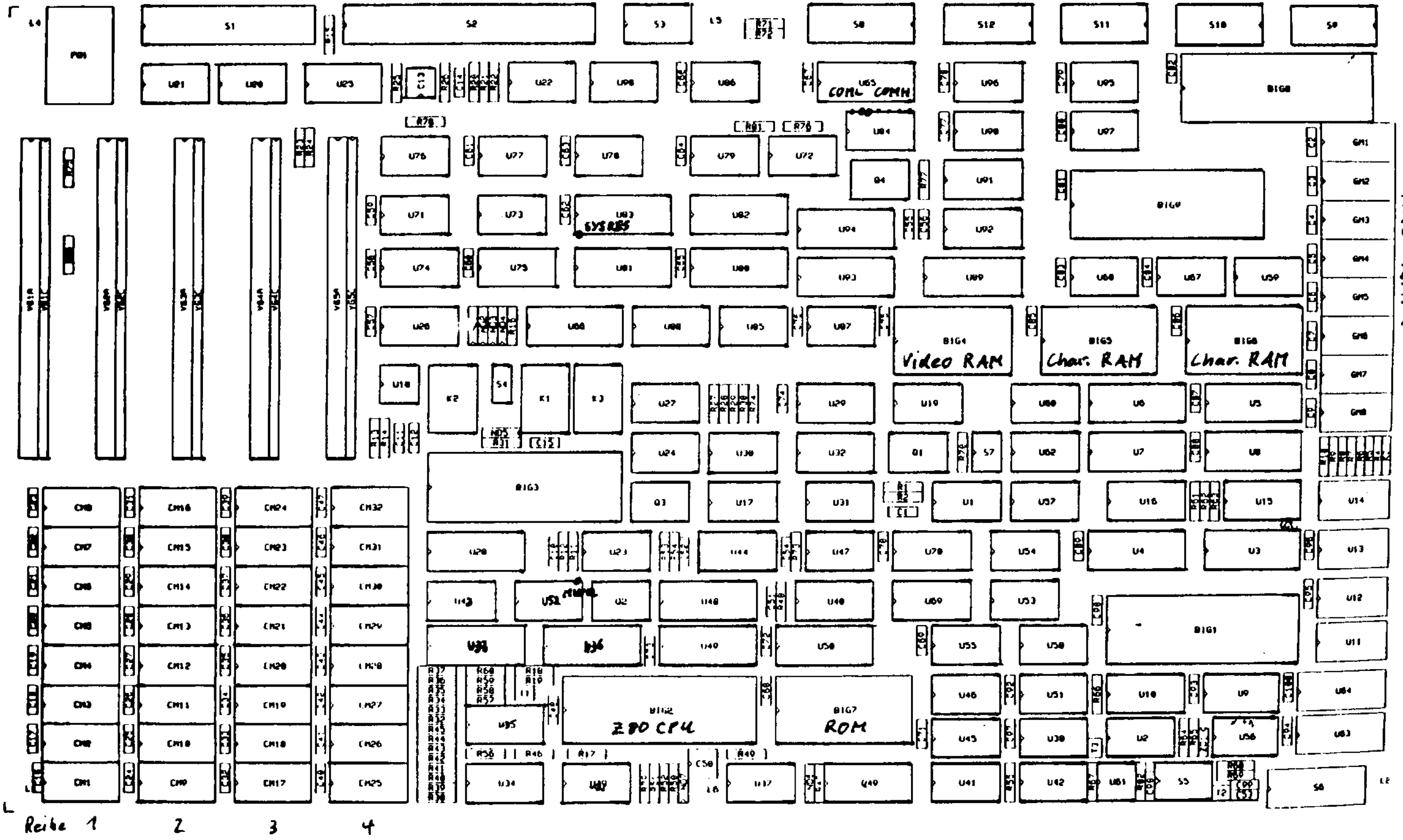
Die Adressen A16 und A17 für die dynamischen RAMs (für den Arbeitsspeicher an den Pins 1 und 2 und für den HRG-Speicher an den Pins 9 und 10 des Steckers S2) können nicht direkt an die RAMs gelegt werden. Sie müssen gemultiplext an ihre jeweiligen Pins 1 geführt werden. Dafür ist zunächst die Computerplatine auszubauen und die Pins 1 aller 32 (Arbeitsspeicher) sowie 8 RAMs (Graphik) untereinander zu verbinden.

Dann werden auf U35 und U15 (beide 74S157) je ein 74S157-IC mit den Pins 1, 8, 15 und 16 huckepack aufgelötet. Dafür muß das untere IC aus der Fassung genommen werden, um es nicht versehentlich in derselben festzulöten. Selbstverständlich müssen beim Auf einander löten beide ICs die gleiche Orientierung haben (Kerbe auf der gleichen Seite). Die nicht verbundenen Pins des oberen ICs werden um ca. 45* hochgebogen, um keinen Kontakt mit dem unteren zu haben.

An die Pins 13 und 14 des auf U35 angebrachten 74S157 (hier genügt auch ein 74LS157) werden zwei isolierte Litzen gelötet, die an die Pins 1 und 2 des Steckers geführt werden. Zwei Litzen, die an die Pins 13 und 14 des auf U15 gesetzten ICs gelötet sind, werden an die Pins 9 und 10 des Steckers geführt. In beiden Fällen spielt die Anordnung der beiden Leitungen keine Rolle. Die Pins 12 der beiden Huckepack-ICs werden je über einen 33-Ω-Widerstand mit dem Pin 1 eines der benachbarten 41256-RAMs verbunden. Dafür muß das jeweilige RAM-IC aus der Fassung genommen und der Draht vorsichtig mit wenig Lötzinn an das obere dicke Ende des Pins 1 gelötet werden. Es ist ebenso möglich, bei herausgenommener Mutterplatine den Widerstand um deren Rand herum auf die Lötseite zu führen. Es kann zweckmäßig sein, den in die Leitung eingefügten 33-Ω-Widerstand mit einem Stück Schrumpf schlauch davor zu bewahren, Kontakt mit irgendwelchen IC-Pins zu bekommen.

Die zur Vervierfachung von Video-RAM und Zeichengenerator (BIG4, BIG5 und BIG6) benötigten 6264LP15-ICs passen nicht mehr in die Fassungen der 6116LP15-RAMs. Hier hilft folgende Vorgehensweise: Die 6116-RAMs werden aus den Fassungen genommen. Die drei Kondensatoren an den Stirnseiten der Sockel werden ausgelötet und auf dem Rücken der 6264-RAMs mit kurzen Drahtstücken zwischen die Pins 14 und 28 gelötet. Die Pins 26 und 28 jedes 6264-Bausteins werden miteinander verbunden. Das IC, das in den Sockel von BIG4 gesteckt werden soll, muß an der Stirnseite (mit Kerbe) so weit beschliffen werden, daß es neben U87 mit den überstehenden Pins 1, 2, 27 und 28 in den Sockel paßt. Die beiden anderen 6264-ICs passen ohne zusätzliche Maßnahmen mit den links überstehenden vier Pins in die Sockel. Vor den Einstecken werden aber die Pins 23 aller drei RAMs um 45* abgebogen.

Die Pins 2 aller drei Speicher werden miteinander und mit dem Pin 8 des Steckers S2 des 4XG3s-Boards verbunden. Auch die hochgebogenen Pins 23 werden untereinander und mit dem Pin 7 des Steckers S2 verbunden. An die Pins 27 der RAMs werden folgende Signale gelegt (die vorher an die Pins 21 der 6116-Bausteine gelangt waren):



Die Mutterplatine des Genie III s

WIKO AG 21-Mar-84 B302MU
WIKO60

D: Eine nützliche Anwendung für das Bit Q6

In Abschnitt A dieses Artikels müssen wir leider zugeben, daß die vereinfachte und beschleunigte Kopie des Computers Nr. 0 auf seine drei Nachbarn mit Hilfe eines gesetzten Bit 6 im Port F3h leider nicht funktionierte. Auf dieses Feature kann auch verzichtet werden. Es würde die Kopierzeit von ohnehin nur dreieinhalb Sekunden um vielleicht eine halbe Sekunde verringern können. Da man nur einmal, höchstens zweimal am Tag diese Initialisierung vornimmt, ist die Wartezeit nicht so wichtig. Um Q6 aber nicht zu verschenken, haben wir es bis zum Aufspüren des Fehlers einer anderen interessanten Verwendung zugeführt:

Das Boot-EPROM des Genie III s ist ohne Treiber direkt mit der CPU verbunden. Daher ist es grundsätzlich möglich, Daten nicht nur aus dem EPROM zu lesen, sondern es auch damit zu beschreiben. Es gibt dazwischen keinen unidirektionalen Treiber, der diese Datenrichtung verbieten würde. Nun kann freilich in einem ROM nicht schreibenderweise etwas geändert werden, denn es ist eben ein Read-Only-Memory. Es ist jedoch möglich, ein statisches RAM parallel dazu zu stecken. Dazu muß lediglich das WR*-Signal der CPU auf den WE*-Pin des RAMs gelegt werden. Eine einfache Logik trifft die Auswahl zwischen dem EPROM und dem statischen RAM. Die Abb. 10 zeigt die Beschaltung der nunmehr zwei Speicherbausteine anstelle des bisherigen EPROMs, nämlich dieses selbst und ein statisches RAM des Typs 6264.

Ein möglichst kurzes Flachbandkabel sitzt mit einem 28-poligen Quetschstecker im EPROM-Sockel. An seinem anderen Ende ist eine Lochrasterplatine entweder direkt aufgelötet oder mit einer Pfostensteckerverbindung befestigt. Fast alle Signale aus dem Sockel werden auf die korrespondierenden Pins der beiden Bausteine einfach durchgeschleift. Nur das Signal ROMEN* (ROM-enable) muß auf die in der Abbildung beschriebene Weise mit Bit 6 des Ports F3h verknüpft werden, um eine Auswahl zwischen dem EPROM und dem neuen statischen RAM treffen zu können. Außerdem wird MWR* von S2 des Bankers, Pin 5 dem Pin 27 (WE*) des RAMs zugeführt.

Q6 wird an Pin 5 des 74LS274 auf der Lötseite der Banker-Platine abgegriffen und auf einen freien Pin von S2 geführt, so daß sich die Verbindung mit der RAM/EPROM-Bank vereinfacht. Es ist nicht erforderlich, die weiterführende Leiterbahn zum 74LS00 zu durchtrennen und gegen GND zu legen, denn das Kopierfeature funktioniert vorerst sowieso nicht. Es kann also nichts passieren, und nach der Korrektur des Fehlers ist weniger Restaurationsarbeit erforderlich (falls man überhaupt noch restaurieren will, nachdem man die Vorzüge dieser ROM/RAM-Bank erkannt hat).

Der Platz zwischen dem EPROM-Sockel und der Zusatzplatine für den Banker ist so eng, daß zwar das EPROM noch eben dazwischen paßte, der neue Quetschstecker jedoch nicht mehr ohne weiteres. Um die Zusatzplatine dennoch fest und kontaktsicher in den Sockel der CPU stecken zu können, empfiehlt es sich, das Deckelteil des Steckers nach dem Anquetschen des Kabels wieder zu entfernen und die hochstehenden Spitzen der Kontaktklemmen abzuschleifen. Diese werden anschließend mit Isolierband beklebt, damit sie nicht die Lötseite der Zusatzplatine berühren können.

Wer den Banker nicht, wohl aber das parallele RAM neben dem EPROM einbauen möchte, kann sich diese Mühe sparen. Dann steht allerdings selbstredend nicht das Bit 6 des Ports F3h zur Verfügung. Ein Umschalter, der anstelle von Q6 wahlweise GND oder über einen Widerstand von 10 kΩ

+5V an die zugehörigen Eingänge des NAND-Gatters legt, erfüllt denselben Zweck.

Um beim Einschalten oder nach einem Reset auf jeden Fall das Boot-EPROM zu erreichen, bekommt es den Zustand 0 des Bits 6 zugeordnet. Wenn Q6 aber 1 lautet, ist das RAM selektiert.

Der zugängliche Adreßraum dieser neu geschaffenen EPROM/RAM-Bank ist 12 kB groß (0000-2FFFh). Das Original-EPROM des Genie III s (2732) belegt davon nur 4 kB (0000-0FFFh). Es kann einfach durch einen 2764 ersetzt werden, in dessen doppelt großem Speicherplatz interessante Features untergebracht werden können. Der 28-polige Sockel führt auch an den vier bisher nicht benutzten zusätzlichen Pins die dafür erforderlichen Signale. Das statische RAM 6264 belegt von vornherein den doppelten Adreßraum von 8 kB (0000-1FFFh).

Bei 1 MB Hauptspeicher grenzt die vergleichsweise bescheidene Zugabe von 8 kB an Memory-Raffgier, aber bei dem Konzept von vier Computern existiert ohne relativ aufwendiges Banking kein gemeinsamer Speicher, der zur Übergabe von Daten geeignet wäre. Dies ist in der EPROM/RAM-Bank der Fall. Wird nämlich Bit 2 des Ports FAh low gesetzt, dann ist ursprünglich auf jeden Fall im Adreßraum 0-12 kB das EPROM ansprechbar. Mit dieser Modifikation ist es nun, je nach Q6, wahlweise das EPROM oder das RAM. So können die vier Computer dort bei Selektion des RAMs auf bequeme Weise miteinander kommunizieren.

Eine andere Anwendung ist für Tüftler interessant, die das Boot-EPROM gerne nach ihrem eigenen Geschmack ändern möchten. Wenn in seinem Adreßraum auch ein RAM liegt, kann eine Versuchsversion des neuen EPROM-Inhalts dort probegefahren werden, ohne gleich ein EPROM dafür zu opfern. Hier empfiehlt es sich, das Flachkabel nach außen zu verlegen, um beim Einstecken eines neuen EPROMs nicht jedesmal den Computer zerlegen zu müssen. Nun ist es zwar nicht möglich, einfach durch Aufruf mit dem Programmnamen den EPROM-Code ins parallele RAM zu laden und dort zu starten. Das anschließend gelistete Programm besorgt das aber.

Die Datei muß zuvor zwischengepuffert werden, um sie anschließend nach dem Umschalten auf die EPROM/RAM-Bank und der Selektion des RAMs dort ablegen zu können. Die Programm-Laderoutine des G-DOS (Einsprung an 4419h) verfrachtet aber Programme immer gleich an ihre Bestimmungsadresse, was in diesem Falle mit der Fehlermeldung 'Ladeversuch auf ROM-Speicherplatz' quittiert würde. Deshalb muß diese Routine (in Grossers DOS-Buch LOADP genannt) zuvor durch einen Patch geändert werden:

An der Stelle 4C4Dh in SYS0 wird das MSB der Ladeadresse eines Records gelesen. In dem hier vorgestellten Programm wird dort eine Umleitung (Label devia) gezapt, die feststellt, wie dieses MSB lautet. Wenn die Adresse unterhalb 3000h liegt, muß es sich um den Code für das neue EPROM handeln, den wir zunächst puffern wollen. Deshalb wird dann in Z. 41 das Bit 7 des MSB gesetzt, so daß z. B. aus der Adresse 0500h die Adresse 8500h wird. So steht nach dem Laden das Programm ab 8000h im Speicher.

Im ersten Segment der Laderoutine muß nun noch der Patch in SYS0 rückgängig gemacht werden. Schließlich sorgt noch eine Sicherung dafür, daß ein fehlerhaft geladenes Programm nicht die Testbedingungen verfälscht: Bei einem aufgetretenen Fehler ist die NZ-Bedingung erfüllt. In diesem Fall erfolgt einfach unverrichteter Dinge eine Rückkehr ins DOS (Z. 18).

Wenn aber alles geklappt hat, wird zunächst in der EPROM/RAM-Bank das RAM selektiert, indem das Bit 6 des neuen Systembytes 2 (Port F3h) high gesetzt wird. Danach wird über das Systembyte 1 die Bank zugeschaltet. Jetzt ist von 0000-1FFFh das statische RAM zugänglich. Die Zeiger für die Quelle (Puffer ab 8000h), das Ziel (RAM ab 0000h) und der Zähler für die Programmlänge (2000h) werden aufgesetzt und der Code an seinen Bestimmungsort geladen.

Die DOS-Laderoutine legt die Einsprungsadresse eines Programms in 4403h ab. Von dort wird sie nach HL geladen, so daß mit JP (HL) das Programm angesprungen werden kann. Für ein Boot-EPROM wird diese Adresse immer 0000h sein, aber auf diese Weise kann auch ein beliebiges anderes

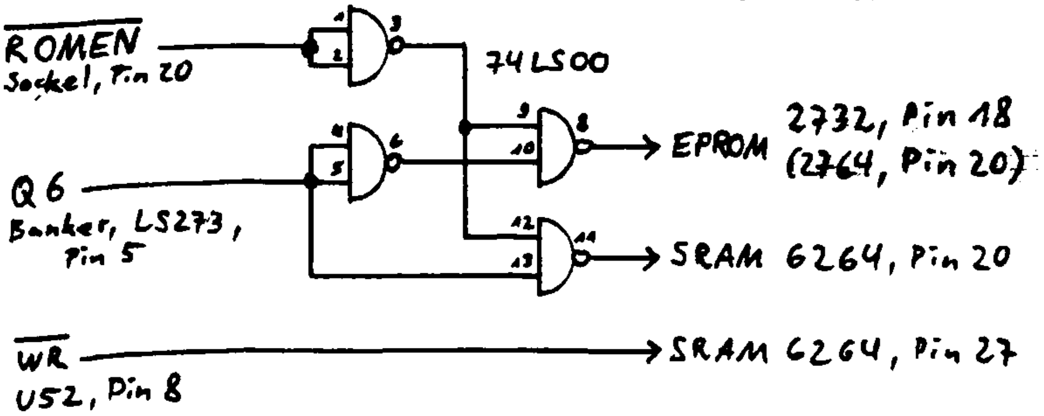
Programm in der neuen Bank abgearbeitet werden.

Beim Einsprung in das Boot-Programm wird geprüft, ob die F1-Taste gedrückt ist, um anstelle eines Boot im Monitor zu bleiben. Das Laden des Programms dauert je nach Größe lange genug, um nach dem Aufruf der Laderoutine rechtzeitig diese Taste zu erreichen. So können sowohl die Bootfähigkeit des neuen EPROM-Inhalts als auch der vielleicht modifizierte Monitor getestet werden.

Der Loader benutzt hier für den neuen EPROM-Inhalt den Filenamen BOOTEPRH/CMD. Das ist natürlich willkürlich. Der Name kann beliebig dem eigenen Geschmack angepaßt werden. Es muß auch nicht unbedingt ein Boot-Programm sein. Diese Routine lädt, was sie erwischt, und springt sie dann an.

Falls Schwierigkeiten beim Einbau oder Betrieb des Statt-ROM-RAMs auftreten (auch solo ohne den Banker), können beide Autoren weiterhelfen.

Abb. 10: Schaltung für die EPROM/RAM-Bank



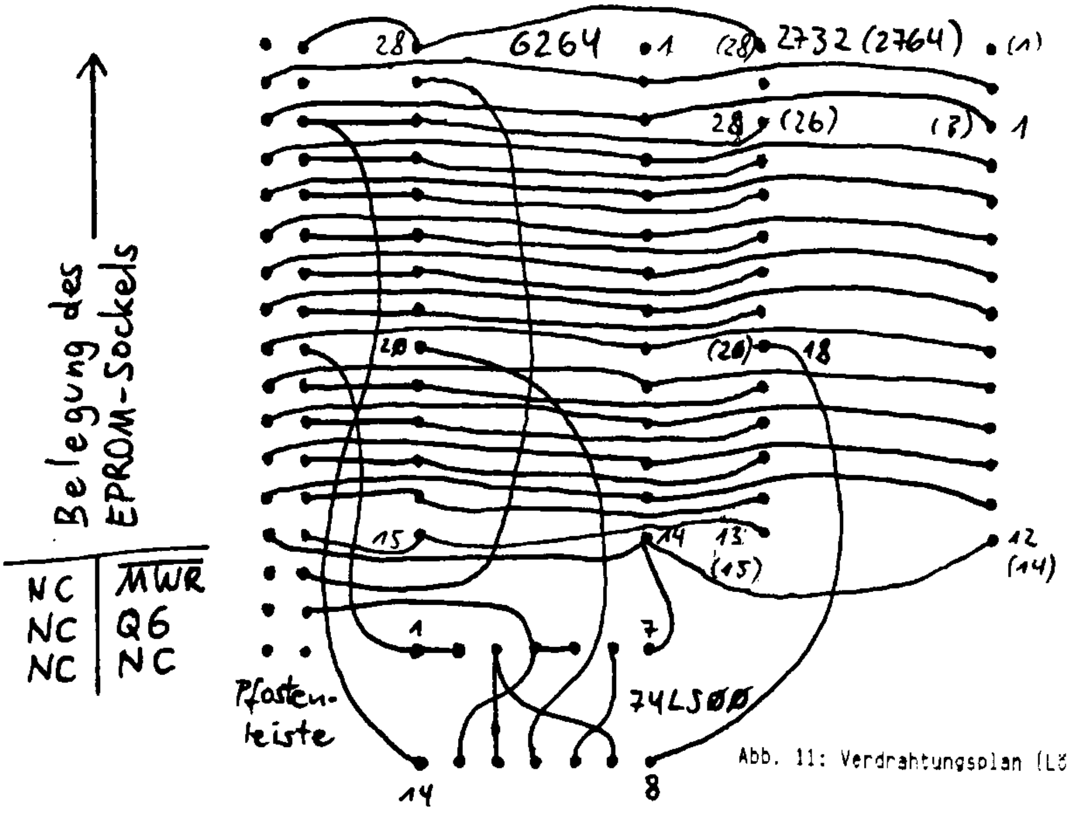


Abb. 11: Verdrahtungsplan (Lötseite)

```
                  ;*******************************************************
                  ; Routine zum Laden von Programmen, die im statischen RAM
                  ; neben dem Boot-EPROM (0000-1FFF) laufen sollen
                  ;
                  ;   (C) 1986 by Arnulf Sopp
                  ;*******************************************************

3000              ORG     3000h

                  ;Umleitung im UP LOADP (Programm laden) in SYS0 setzen,
                  ;Programm BOOTEPRH/CMD nach 8000 verschoben laden
3000    212C30    start   LD      HL,devia    ;Umleitung zur Modifikat.
3003    224E4C            LD      (4c4eh),HL  ;in LOADP patchen
3006    213530            LD      HL,doscall  ;Zeiger auf DOS-Befehl
3009    CD1944            CALL    4419h       ;diesen ausführen
300C    21654C            LD      HL,4c65h    ;alten Code in LOADP
300F    224E4C            LD      (4c4eh),HL  ;restaurieren
3012    C0                RET     NZ          ;Ende, falls Fehler

                  ; EPROM/RAM-Bank und dort wiederum das RAM selektieren
3013    F3                DI                  ;es wird brenzlig
3014    DBF3              IN      A,(0f3h)    ;Systembyte 2
3016    F640              OR      40h         ;RAM statt EPROM
3018    D3F3              OUT     (0f3h),A    ;selektieren
301A    3EC0              LD      A,0c0h      ;auf RAM/EPROM-Bank
301C    D3FA              OUT     (0fah),A    ;Bank freigeben

                  ;Prg. von 8000 nach 0000 im Boot-RAM laden und anspringen
301E    210080            LD      HL,8000h    ;Puffer des Boot-Progr.
3021    55                LD      D,L         ;DE <- 0000
3022    5D                LD      E,L     ;( = Zieladresse)
3023    010020            LD      BC,2000h    ;Programmlänge
3026    EDB0              LDIR                ;Programm ins Boot-RAM
3028    2A0344            LD      HL,(4403h)  ;Puffer der Startadresse
302B    E9                JP      (HL)        ;Programm anspringen

                  ;Patch in LOADP: Code bis 3000 soll nach 8000 ff. kommen
302C    CD654C    devia   CALL    4c65h       ;MSB Ladeadresse holen
302F    FE30              CP      30h         ;zum Boot-Progr. gehörig?
3031    D0                RET     NC          ;norm. weiter, falls nein
3032    F680              OR      80h         ;ab 8000 laden
3034    C9                RET                 ;normal weiter

                  ;DOS-Befehl zum Laden des neuen Boot-Programms
3035    4C        doscall DM      'LOAD,BOOTEPRH/CMD',0dh ;DOS-Bef., ENTER

3000                      END     start       ;dort Einsprung

00000 Fehler

devia   302C    doscall  3035   start   3000
```

Helmut Bernhardt Arnulf Sopp

<u>Anmerkung der Redaktion:</u>
Diese PDF wurde von Helmut Bernhardt, Arnulf Sopp und Fritz Chwolka aus den Resten der Altpapiersammlung wieder zurechtgebastelt. Mai 2015

Selbstbau Doubler für EXP1

Helmut Bernhardt

Nachdem sich bei mir nun schon der zweite WD1771 Single Density Floppycontroller verabschiedet hat und der Chip wohl auch bald nicht mehr zu haben ist, entschloß ich mich, mir selbst einen Doubler zu bauen, der ohne diesen Käfer auskommt. Es ist auch nicht einzusehen, warum auf einem Doubler zwei Floppycontroller sitzen müssen, wo doch der WD1791 alles das, was der WD1771 kann (nämlich beim Booten die erste Spur der Diskette in Single Density und gelegentlich auch mal ganze Disketten in SD zu lesen oder zu schreiben, wenn ein unterausgerüsteter Zeitgenosse den digitalen Informationsaustausch nur in diesem Format leisten kann) nebenbei auch erledigen könnte.

Letzteres Argument soll nun aber auch gegenstandslos werden, denn den hier beschriebenen Doubler kann sich leicht jeder selbst bauen und auch finanzieren. Grundlage ist der WD2793 Floppycontroller, der schon aufgrund seiner Bescheidenheit bei der Spannungsversorgung (ausschließlich + 5V und GND) mein Wohlwollen verdient hat. Außerdem enthält er auf dem Chip schon einen PLL-Datenseparator und eine Schreibvorkompensation. Dadurch können die meisten ICs auf den Doublern von TCS und dem davon abgekupferten von RB entfallen, die nur diesen Job leisten.

Der WD2793 kann intern den an Pin 24 angelegten Takt von 2MHz (für den Betrieb von 8" Floppies) auf 1MHz (5" Floppies) runterteilen, wenn an den Pins 17 und 25 Low-Pegel angelegt wird, über Pin 37 läßt sich vorgeben, ob in Single Density (high) oder Double Density (low) gelesen und geschrieben werden soll.

Durch die 5 zusätzlichen ICs links neben dem WD2793 auf dem Doubler Board lassen sich diese Features programmieren. Zwei Flip Flops geben über ihre Q-Ausgänge die Pegel für die Pins 17, 25 und 37 vor. Nach einem RESET ist der Floppycontroller auf Minifloppy und Single Density eingestellt. Durch Ausgabe folgender Werte an das Contollregister mit der memory mapped Adresse 37ECH bzw. das Sektorregister 37EEH lassen sich die Betriebszustände folgendermaßen einstellen:

37ECH:	1111 1xx0 bin (FFH)	Single Density
	1111 1xx1 bin (FEH)	Double Density
37EEH	1000 0xx0 bin (80H)	Mini Floppy (5 1/4")
	1100 0xx0 bin (C0H)	Maxi Floppy (8")

Diese Daten haben im Controll- bzw. Sektorregister keine Wirkung. Aus den Daten und der Schreibadresse werden nur extern die Triggersignale für die Flip Flops decodiert, wobei für die Density DØ und für die Drive-Größe D6 den einzustellenden Pegel vorgibt.

Diese Schaltung ist in Abb.1 gezeigt. Das einseitige Layout und die Anordnung der Drahtbrücken und Bauteile ist aus den folgenden Abbildungen zu ersehen. Die zur Position des WD2793 leicht versetzte Dil40-Anordnung von Lötpunkten wird von der Lötseite mit einem entsprechenden Stecker bestückt, der in den Sockel des WD1771 auf dem EXP1 gesteckt wird. Zur Zugentlastung dieses Steckers sollten an ca. 10 der durchragenden Pins auf der Bestückungsseite kurze Drahtstücke angelötet werden. Das verhindert ein eventuelles Abreißen der Leiterbahnen beim Herausziehen des Doublers aus dem 1771-Sockel .

Beim Einsatz dieses Doublers muß der invertierende Datentreiber 74LS640 neben dem 1771-Sockel durch den nichtinvertierenden pinkompatiblen Typ 74LS245 ersetzt werden, da der WD2793 keine invertierenden Daten-Ein-/Ausgänge (wie WD1771 und WD1791) hat. Die Entscheidung fiel für den WD2793, der sich nur in seinen nichtinvertierenden Dateneingängen vom WD2791 unterscheidet, wegen un-

überwindlicher Beschaffungsschwierigkeiten beim WD2791. Das bedeutet aber leider auch, daß dieser Doubler bei anderen Floppycontroller-Boards nicht verwendet werden kann, da dort nicht einfach ein Treiber ausgewechselt werden kann.

Auch ein weiterer Nachteil dieses Doublers soll noch erwähnt werden. Für die einwandfreie Funktion von Datenseparator und Schreibvorkompensation sind einige Trimmungen an den Potentiometern P1 und P2 und dem Trimmkondensator C1 vorzunehmen. Dazu ist ein Oszilloskop nötig. Wenn das nicht verfügbar ist, kann im Notfall auch mit Probieren getrimmt werden. Dann werden P1, P2 und C1 auf Mittelstellung gedreht und durch wiederholte Bootversuche und ständiges Verändern der Einstellung von P2 und C1 die günstigste Einstellung für das Lesen einer Diskette gefunden. Mit P1 muß dann noch die Schreibvorkompensation geregelt werden, was durch Schreiben auf Diskette und anschließendes Lesen geprüft werden kann. Die Einstellungen sind nicht sehr kritisch, so daß auch dieses Vorgehen möglich ist.

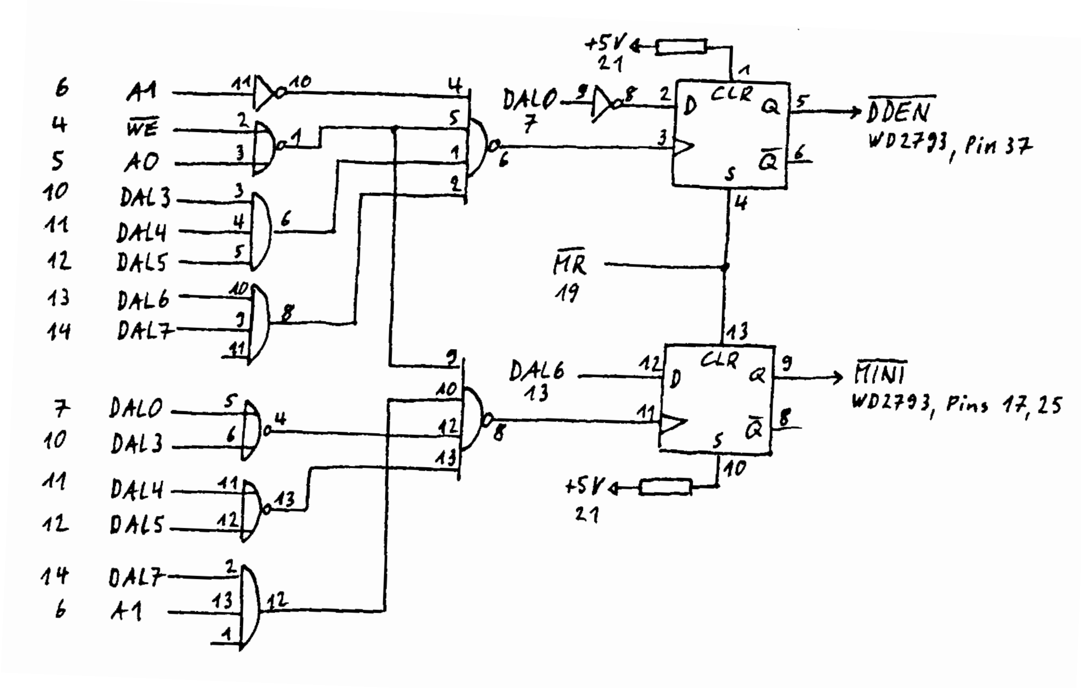
Die im Datenblatt vorgegebene Einstellung mit Hilfe eines Oszilloscops läuft folgendermaßen ab:

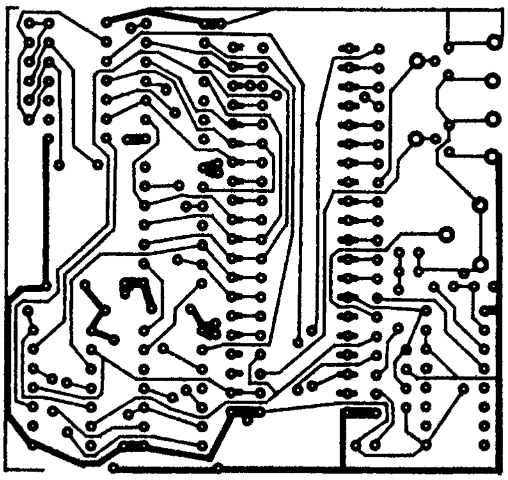
Der Computer wird mit gedrückter <BREAK> Taste eingeschaltet, um im Level II BASIC zu landen. Es wird der Befehl POKE 14316,255 (auf DD schalten) ausgeführt. Dann werden mit einem Jumper die beiden Stifte "TEST" kurzgeschlossen (Pin 21 des WD2793 an GND). Mit P1 kann dann die Breite der Pulse an Pin 31 (WD) unter Oscarkontrolle auf einen Wert zwischen 100 und 300ns eingestellt werden.

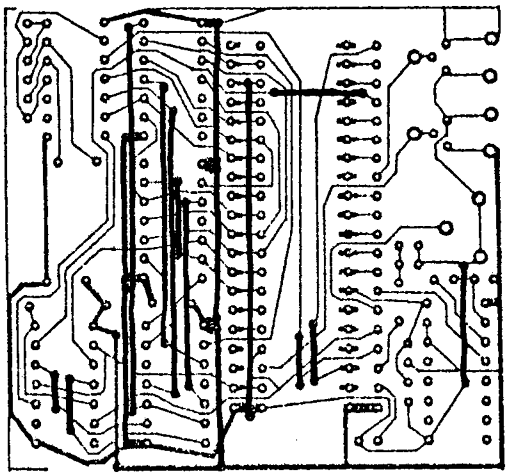
Mit P2 wird dann die Pulsbreite an Pin 29 (TG43) auf 60-65ns eingestellt. Mit CI wird an Pin 16 (DIRC) eine Frequenz von 250kHz eingestellt.

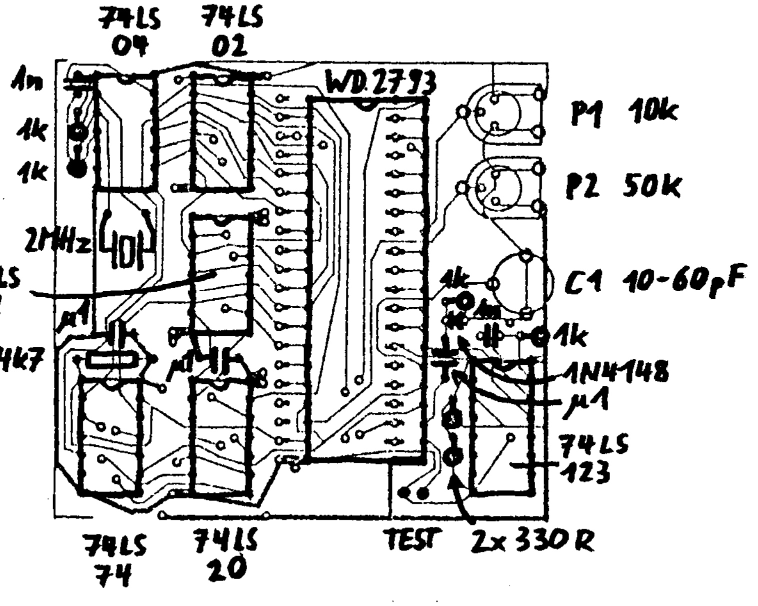
Um die Hardware-Voraussetzungen, auch 8"-Floppies betreiben zu können, zu nutzen, muß die Anordnung der Signale auf dem 34poligen Shugart-Bus für Minifloppies auf die für den 50poligen Bus für 8"-Floppies umsortiert werden. Bei älteren 8"-Drives muß diesen zusätzlich noch das Signal TG43 (WD2793, Pin29) über einen Open Collector Inverter zugeführt werden. Neuere 8"-Drives benötigen dieses Signal nicht mehr.

Wer sich diesen Doubler aufbauen will, kann für 10,-DM (ind. Rückporto) von mir ein geätztes, ungebohrtes Board bekommen und darf sich mit der Beschaffung der Bauteile, dem Bohren, Löten und Trimmen selbst, abmühen.








74LS 04
74LS 02
WD 2793
1n
1k
1k
2MHz
74LS 11
µ1
4k7
µ1
P1 10k
P2 50k
C1 10-60pF
1k
1n
1k
1N4148
µ1
74LS 123
74LS 74
74LS 20
TEST
2x 330 R

N3: Minifloppy-Stecker

Belegung:			Funktion:	
01,02:	GND	/READY,/MINI	GND:	Signalmasse
03,04:	GND	/HDLD	/DS0-3:	Laufwerkauswahl
05,06:	GND	/DS3	/HDLD:	Kopf laden
07,08:	GND	/INDEX	/MOTOR ON:	Motor ein
09,10:	GND	/DS0	/DIR:	Spurwechsel-Richtung
11,12:	GND	/DS1	/STEP:	Spurwechsel-Impuls
13,14:	GND	/DS2	/WR DATA:	Schreibdaten
15,16:	GND	/MOTOR ON	/WR GATE:	Schreiben ein
17,18:	GND	/DIR	/SIDE:	Kopfauswahl
19,20:	GND	/STEP	/MINI:	Mini/Maxi-Umschaltung
21,22:	GND	/WR DATA		
23,24:	GND	/WR GATE	/READY:	Laufwerk bereit
25,26:	GND	/TRACK 0	/INDEX:	Indexloch-Impuls
27,28:	GND	/WR PT	/TRACK 0:	Kopf auf Spur 0
29,30:	GND	/RD DATA	/WR PT:	Schreibschutz ein
31,32:	GND	/SIDE	/RD DATA:	Lesedaten
33,34:	GND	/READY,/DCHG	/DCHG:	Diskettenwechsel

N4: Maxifloppy-Stecker

Belegung:			Funktion:	
01,02:	GND	/LOW CUR	GND:	Signalmasse
03,04:	GND	/MOTOR OFF		
05,06:	GND	--	/DS0-3:	Laufwerkauswahl
07,08:	GND	--	/HDLD:	Kopf laden
09,10:	GND	/TWO SIDED	/MOTOR OFF:	Motor aus
11,12:	GND	/DCHG	/DIR:	Spurwechsel-Richtung
13,14:	GND	/SIDE	/STEP:	Spurwechsel-Impuls
15,16:	GND	--	/WR DATA:	Schreibdaten
17,18:	GND	/HDLD	/WR GATE:	Schreiben ein
19,20:	GND	/INDEX	/SIDE:	Kopfauswahl
21,22:	GND	/READY	/LOW CUR:	Schreibstrom niedrig
23,24:	GND	--		
25,26:	GND	/DS0		
27,28:	GND	/DS1	/READY:	Laufwerk bereit
29,30:	GND	/DS2	/INDEX:	Indexloch-Impuls
31,32:	GND	/DS3	/TRACK 0:	Kopf auf Spur 0
33,34:	GND	/DIR	/WR PT:	Schreibschutz ein
35,36:	GND	/STEP	/RD DATA:	Lesedaten
37,38:	GND	/WR DATA	/DCHG:	Diskettenwechsel
39,40:	GND	/WR GATE	/TWO SIDED:	Zweiseitige Diskette
41,42:	GND	/TRACK 0		
43,44:	GND	/WR PT		
45,46:	GND	/RD DATA		
47,48:	GND	--		
49,50:	GND	--	-- = frei	

EG 64 MBA

Die Funktion des EG-64 MBA's besteht darin, die in den unteren 16 K Adressraum liegenden ROM- und I/O-Bereiche "auszublenden" um stattdessen den Zugriff auf den dort vorhandenen RAM zu gestatten.

Die separate Umschaltbarkeit einzelner Blöcke (anstatt der gesamten unteren 16 K) ermöglicht ihre unabhängige Verwendung durch verschiedene Programmteile bzw. für verschiedene Aufgaben .

Hierbei wurde die ursprüngliche Speicheraufteilung des GENIE zugrundegelegt, wie aus der nachfolgenden Tabelle zu entnehmen ist.

Die Steuerung des MBA erfolgt über den Port 223 (bzw. DF hex.), das heißt, über OUT 223 ,F (bzw. entspr. Maschinenbefehle). Jede Umschaltung kann unabhängig von den anderen vorgenommen werden, und zwar nach folgender Tabelle*:

Adressraum	normale Belegung		<>	Umschaltg. a. RAM	
	F =	Funktion		F =	Funktion
0000 - 2FFF	0 1	read ROM write ROM	<> <>	8 9	read RAM write RAM
3000 - 35FF	2 3	read L-4 write L-4	<> <>	10 11	read RAM write RAM
3600 - 37FF	4	r/w I/O (Floppy)	<>	12	r/w RAM
3800 - 3BFF	5	r/w Tastatur	<>	13	r/w RAM
3C00 - 3FFF	6	r/w Bildschirm	<>	14	r/w RAM
RESET des MBA	7	per RESET-Knopf	<>	15	nicht per RESET-Knopf

Adressraum in hexadezimaler Notation
L-4 ist der Zusatzrom (falls vorhanden)
I/O ist nur bei Anschluß von FC-Laufwerken belegt
(TRS-80-Druckeradresse, Floppycontroller, Uhr)

Die Ausgangskonfiguration beim Einschalten des GENIE ist die gleiche, wie sie ohne MBA vorliegt: alle ROM's sowie Tastatur, Bildschirm und I/O sind eingeschaltet (linke Seite d. Tabelle).

Dieser Grundzustand kann auf zwei Wegen schnell wieder erreicht werden:
1. Durch Lesen vom Port 223 (z.B.: X=INP(223))
2. Durch Druck auf RESET

Dies erspart das einzelne Zurückschalten aller Blöcke.

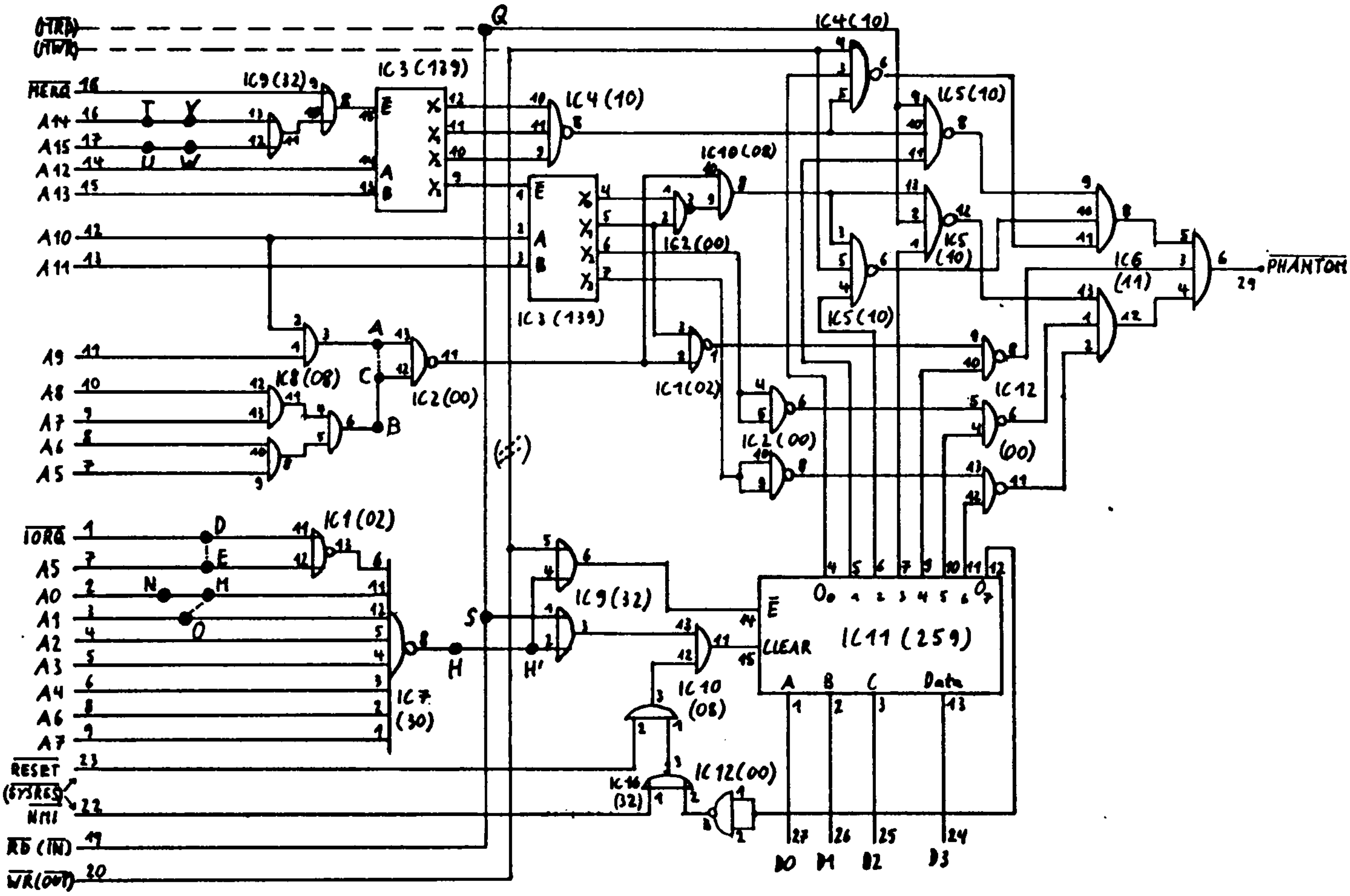
Funktion 7/15 dient einem besonderen Zweck:

Will man ein Rücksetzen der Speicherumschaltung bei einem RESET verhindern (um z.B. mit einem modifizierten Interpreter weiterzuarbeiten), so kann man dies durch ein OUT 223,15 erreichen und mit OUT 223,7 oder X=INP(223) wieder rückgängig machen.

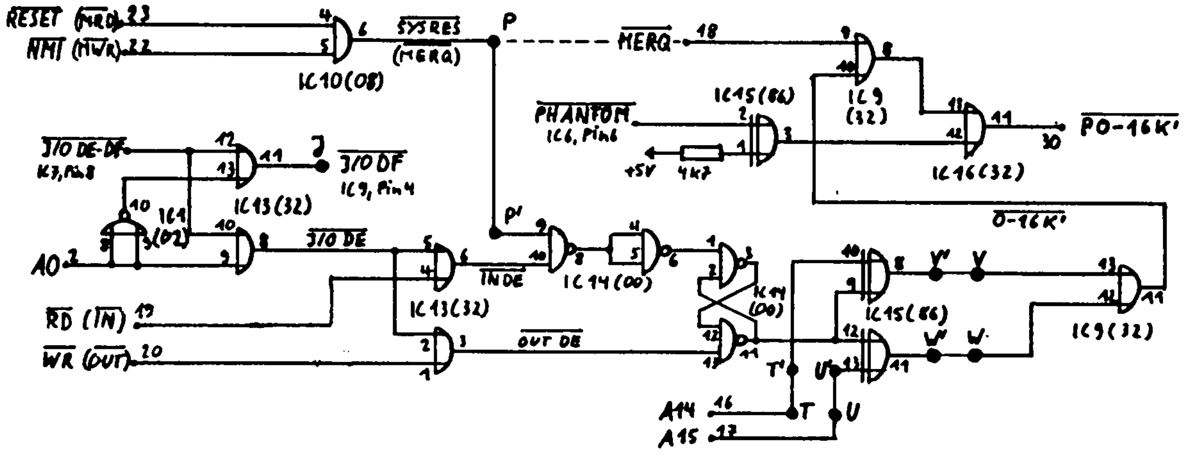
Für den Maschinenspracheprogrammierer ist eine etwas andere Betrachtungsweise nützlich:

Die unteren 3 Bits des Ports DF dienen der Funktionsauswahl, sie selektieren quasi den Schalter, der "umgelegt" werden soll (0 - 7).

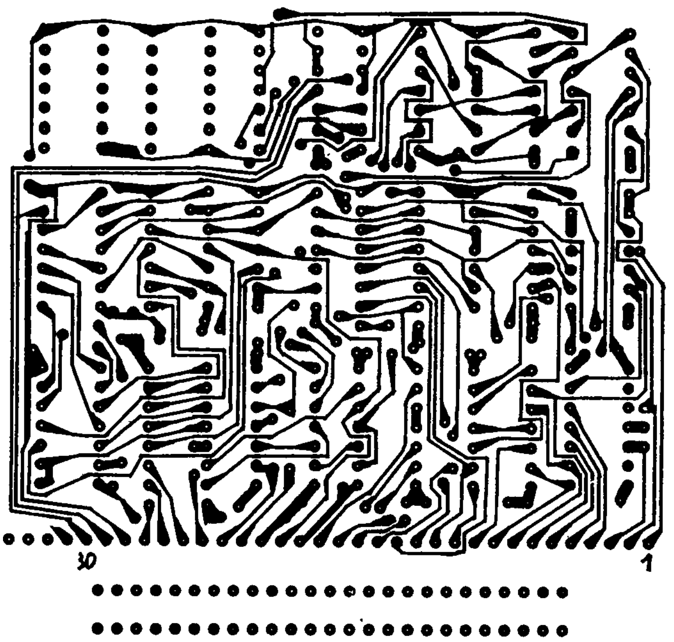
Das vierte Bit ist der eigentliche Schalter; ist es gesetzt, so wird auf RAM-Zugriff geschaltet, ist es rückgesetzt, kann der entsprechende ROM- bzw. I/O-Block erreicht werden.

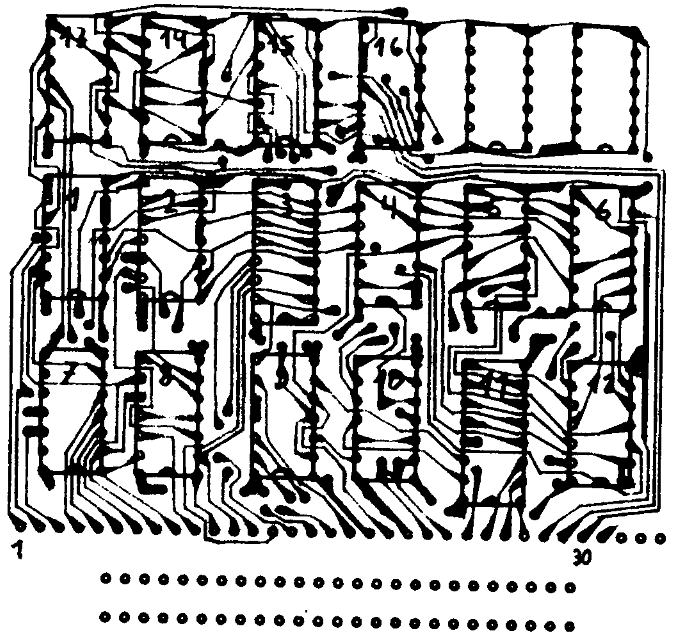


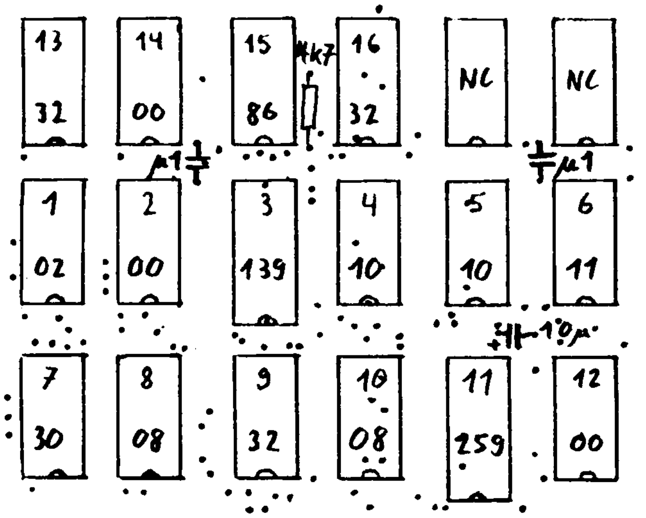
(ĪORQ)
(HWR)
Q
MERQ 18
A14 16
A15 17
A12 14
A13 15
A10 12
A11 13
A9 11
A8 10
A7 9
A6 8
A5 7
IC9(32)
IC3(139)
IC4(10)
IC5(10)
IC6 (11)
IC10(08)
IC2(00)
IC1(02)
IC8(08)
IC2(00)
IC12 (00)
PHANTOM
29
IORQ 1
A5 7
A0 2
A1 3
A2 4
A3 5
A4 6
A6 8
A7 9
IC1(02)
IC7 (30)
IC9(32)
IC10 (08)
IC11(259)
CLEAR
Data
RESET 23
(SYSRES)
NMI 22
RD (IN) 19
WR(OUT) 20
IC16 (32)
IC12(00)
D0 D1 D2 D3

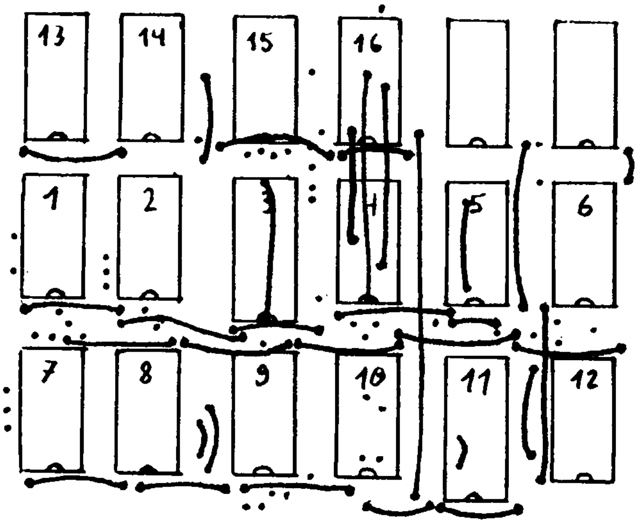


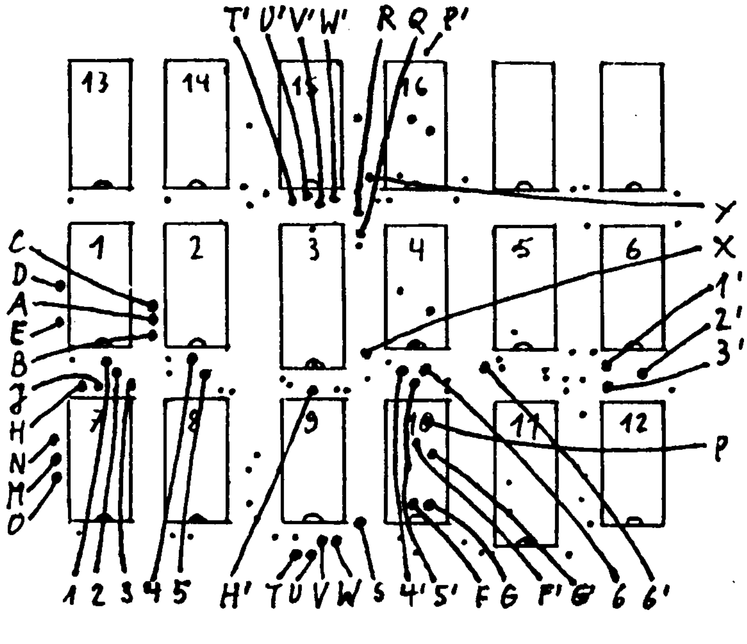
RESET (MRQ) 23
NMI (HWR) 22
IC10(08)
SYSRES
(MERQ)
P
MERQ 18
IC9 (32)
PHANTOM
IC6, Pin6
IC15(86)
+5V
4k7
IC16(32)
30
PO-16K'
0-16K'
J/O DE-DF
IC3, Pin8
J/O DF
IC9, Pin4
IC13(32)
IC1 (02)
J/O DE
A0 2
INDE
IC13(32)
IC14(00)
RD (IN) 19
WR (OUT) 20
OUT DE
IC14 (00)
IC15(86)
IC9(32)
A14 16
A15 17











3,4) Layout

5) Bestückungsplan

6) Drahtbrücken, die für alle Computer und Funktionsvarianten gelegt werden müssen

7) Anschlußpunkte für die Drahtbrücken zur Auswahl der Funktionen

- **512K RAM Adressierung mit flexibler MMU**
- **2 serielle Schnittstellen**
- **2 DMA Kanäle**
- **2 16-Bit-Timer**
- **Erweiterter Befehlssatz**
- **Höhere Geschwindigkeit**

durch Ersetzen des Z80 durch den HD64180

Der Hersteller der HD64180-CPU benötigte zur Darstellung aller Leistungen dieses Z80-aufwärtskompatiblen Prozessors in einem 'Datenblatt' ein 206 Seiten starkes Buch. Deshalb soll hier gar nicht erst der Versuch gemacht werden, die Vorteile dieses Chips gegenüber seinem Vorgänger Z80 zusammenzufassen. Hier soll nur dem Bastler eine Möglichkeit gezeigt werden, wie dieser Prozessor im alten Z80-Microcomputer die Regie übernehmen soll.

Für die Unterbringung des HD64180 anstelle des Z80 im Computer wird eine kleine Adapterplatine benötigt, an die von der Lötseite ein 40poliger Dil Stecker angelötet wird und auf der die neue CPU und 5 weitere ICs untergebracht werden. Diese Platine wird in den Sockel des Z80 gesteckt und liefert über den Dil Stecker alle Z80-äquivalenten Signale des HD64180 an den Computer.
Auf der Platine sind schon zwei Treiber-ICs enthalten, die bei den Signalen der CPU-internen seriellen Schnittstellen die Pegelwandlungen +12V/-12V => GND/+5V und umgekehrt vornehmen. Der Sendetreiber muß über zwei zusätzliche Leitungen +12V und -12V zugeführt bekommen. Die Signale der seriellen Schnittstellen können vom Stecker CNA direkt abgenommen werden und z.B. an ein Modem geführt werden.

Auf der Adapterplatine wird der Takt für den Prozessor aus einem internen Oszillator und einem externen Quarz hergeleitet. Am Pin 64 des HD64180 wird die durch 2 geteilte Quarzfrequenz, die dann dem Systemtakt entspricht, bereitgestellt und kann bei Bedarf mit der Drahtbrücke J1 über den Pin 6 des Z80-Sockels an den Computer geliefert werden. Dann muß aber die bisherige Z80-Systemtaktquelle davon abgetrennt werden. Es muß geprüft werden, ob der Takt für den Z80 anderweitig als feste Frequenz verwendet wird. Dann muß dort weiterhin die Taktfrequenz für den Z80 hingeführt werden. Wenn aber der Takt für die Synchronisation mit der CPU benötigt wird, muß dieser vom Pin 64 des HD64180 entnommen werden.
Außerdem werden mit 3 weiteren ICs aus den zusätzlichen Adreßleitungen A16 bis A18 Freigabesignale für die Ansteuerung von 2 x 8 Stück 41256-RAMs und die an die Pins 1 dieser RAMs zu legende gemultiplexte RAM-Adresse MA8 erzeugt. Je nach Funktionsweise des bisherigen Adreßmultiplexers für vorher 4164-RAMs wird für die Steuerung des Adreßmultiplexers nun MUX oder MUX* vom Stecker CNB verwendet. Für die Wahl zwischen diesen Signalen muß geprüft werden, ob das bisherige MUX-Signal low oder high aktiv war, entsprechend ist dann das low aktive MUX* oder das high aktive MUX zu verwenden. Das ist deshalb von Bedeutung, weil sonst beim Refresh die oberen Adressen an die RAMs durchgeschaltet werden, und dann das Refreshing nicht stattfinden kann.

Das Signal RAS* von CNB kann direkt anstelle des bisherigen RAS*-Signals an die RAMs gelegt werden (eventuell über 33 Ohm). Bei Bestückung des Computers mit nur 8 Stück 41256-RAMs wird nur das Signal CAS0* von CNB benötigt. Dieses Signal wird nicht direkt an die RAMs geführt sondern an den Eingang des Gatters gelegt, mit dem die Verknüpfung mit dem RAM-Freigabesignal erfolgt. Diese Stelle in der Schaltung herauszufinden, ist die einzige Anforderung an den Bastler, der sich die Vorteile

des HD64180 für seinen Computer zunutze machen will. Hier muß auch geprüft werden, ob das bisherige CAS-Signal high oder low aktiv an diese Stelle gelangte. Wenn hier ein high aktives CAS nötig ist, muß CAS0* noch über einen Inverter geführt werden.
Wenn die bisherige Herleitung dieser Signale aus dem Systemtakt für den Z80 oder über Gatterlaufzeiten aus dem MERQ*-Signal erfolgte, kann diese weiterbenutzt werden. Bei einer Änderung des Systemtaktes beim Umrüsten kann bei größeren Unterschieden zwischen altem und neuen Takt eine Herleitung von RAS*, MUX bzw. MUX* und CAS* aus MERQ* über Gatterlaufzeiten eventuell nicht mehr funktionieren. Hier muß dann probiert werden, wieviele Verzögerungsgatter zwischen den einzelnen Signalen günstig sind.

Wenn das RAS*-, MUX- und CAS*-Timing bisher aus dem Systemtakt für den Z80 (nicht aus einem dazu vielfachen Muttertakt) hergeleitet wurde, kann diese Schaltung weiterbenutzt werden, indem die bisherige Taktquelle unschädlich gemacht wird und statt dessen der Takt vom Pin 64 des HD64180 über Pin 6 des 280-Sockels ins System geführt wird.
Eventuelle Treiber des ursprünglichen Systemtaktes zwischen ursprünglicher Taktquelle und Z80 müssen überbrückt werden, um auch Bauteile zu erreichen, die mit dem Takt vor diesem Treiber versorgt werden.

Dann werden nicht die Signale RAS*, MUX bzw. MUX* und CAS* des Adapterboards benutzt. Hier wird dann statt dessen das Signal MUX bzw. MUX* (was immer es dort ist) der bisherigen Schaltung an den Freigabe-Eingang des 74LS125- Treibers für A16 auf dem Adapterboard gelegt, der bisher mit diesem Pin verbundene Pin des 74LS04 muß dann aus der Fassung gebogen werden.
Wenn eine weitere Reihe von 8 Sockeln für RAMs vorhanden ist, oder wenn man das Risiko auf sich nehmen will, 8 weitere 41256-RAMs auf die anderen 8 Stück huckepack zu löten, dann wird der Pin 15 dieser zusätzlichen 8 RAMs mit CAS1* von CNB direkt beschältet. Beim Huckepack-Verfahren dürfen die Pins 15 der beiden übereinander sitzenden ICs also nicht miteinander verbunden werden.
Es sei noch angemerkt, daß die meisten 4164-RAMs mit einer 7-Bit-Refreshadresse auskommen, während 41256-RAMs grundsätzlich eine 8-Bit-Refreshadresse benötigen. Es muß also gewährleistet sein, daß A0-A7 beim Multiplexen zusammen durchgeschaltet werden. Bei älteren Geräten ist es wahrscheinlich, daß anstelle von A7 eine höhere Adresse zusammen mit A0-A6 durchgeschaltet wird. In diesem Fall sind A7 und die falsch angeschlossene höhere Adresse an den Multiplexer-Eingängen zu vertauschen.

Ein praktisches Beispiel

Nach diesem allgemeinen philosophischen Prolog, der eigentlich nur dem versierten Bastler prinzipiell zu berücksichtigende Punkte aufzeigt und wenig zum Wie sagt, soll nun an einem praktischen Beispiel gezeigt werden, was zu tun ist.

Das Beispiel ist ein GENIE IIs, dessen Besonderheit darin besteht, daß dort eine Umschaltung des Taktes zwischen 1,77 und 5,3 (8) MHz möglich ist. Wenn die 'LSP'-Taste gedrückt ist, das EPROM eingeschaltet ist und/oder, wenn durch die Software ein Bit in einem Steuerlatch auf 0 gesetzt wird (nach RESET ist dieses Bit immer 0), läuft der Computer nur mit 1,77 MHz. Ein Lesen des EPROMs ist bei höheren Taktraten nicht möglich. Diese Takterzeugung und -umschaltung kann für die HD64180-CPU nicht verwendet werden, mit diesem Prozessor muß man auf eine Taktumschaltung verzichten. Wegen der nach einem RESET selbst auferlegten 4 WAITs dieser CPU ist es aber durchaus möglich, ein EPROM auch bei sehr hoher Taktrate noch zu lesen. Nach Abschalten des EPROMs können auch die WAITs zurückgenommen werden. Der GENIE IIs läuft so mit ausschließlich 9,216 MHz ohne Probleme. Es muß allerdings darauf geachtet werden, daß vor dem Reaktivieren des EPROMs die WAITs wieder eingefügt werden.

Dafür wurde einfach das Adapterboard für die Takterzeugung und -umschaltung entfernt und das 74S74-IC dieses Boards in den Sockel (U18) gesteckt, aus dem das Board entnommen wurde. Der Pin 6 dieses ICs ist aus der Fassung zu biegen. Bei U19 ,74LS125 wurden die Pins 8 und 9 aus der Fassung gebogen und diese Pins des Sockels miteinander verbunden. Damit ist der Treiber zwischen alter (abgehängter) Taktquelle und 280, Pin6 überbrückt. Auf dem HD64180-Adapterboard werden an den dafür vorgesehenen Stellen ein 18,432MHz-Quartz und die beiden 22pF- Kondensatoren bestückt. Der Pin 64 des HD64180 und der Pin 6 des Steckers für den Z80-Sockel werden durch einen Draht miteinander verbunden.

Beim GENIE IIs ist außerdem zu berücksichtigen, daß durch entsprechende Bits im Systemsteuerlatch zwischen RAM und memory-mapped I/O-Baugruppen innerhalb der unteren 16K umgeschaltet werden kann. Die Decodierung der Freigabesignale dieser Baugruppen berücksichtigt natürlich nur die Z80-Adressen bis A15 und nicht die HD64180- Adressen A16 bis A18. Entsprechend würden diese Baugruppen auch in den entsprechenden Adreßbereichen anderer Banks (physikalische Adressen 10000H-7FFFFH) angesprochen werden. Um dies zu verhindern, muß in der Adreßdecoder-Baugruppe auf der I/O-Karte anstelle von A15 ein durch OR- Verknüpfung aus 0-64K* und A15 hergeleitetes 0-32K* Signal verwendet werden, das an den aus der Fassung gebogenen Pin 5 von U13, 74LS138 gelegt wird (siehe Abbildung).

Aus geometrischen Gründen läßt sich die Standardversion des Layouts des Adapterboards im GENIE IIs nicht verwenden. Ein für diesen Computer geeignetes Layout ist ebenfalls gezeigt, bei dem aber wegen ungünstigerer Anordnung der Bauteile zueinander sehr viel mehr Drahtbrücken und freie Verdrahtungen nötig sind.

Beim GENIE IIs wird das RAS*-, MUX- und CAS*-Timing aus dem Systemtakt hergeleitet und kann deshalb beibehalten werden. MUX von Pin 1 von U6, 74S157 kann für die Umschaltung von A16 und A17 auf dem HD64180-Adapterboard verwendet werden. Dafür wird durch freie Verdrahtung von Pin 1 von U6 eine Verbindung mit MUX* von CNB des Adapterboards hergestellt und der Pin 5 des 74LS04 auf dem Board aus der Fassung gebogen.

Auf dem speziell für den GENIE IIs gebauten Adapterboard muß die Erzeugung des 0-64K*-Signals noch in freier Verdrahtung nachgearbeitet werden. Dazu stehen die OR-Gatter des 74LS32 (Pins 1,2,3 und 13,12,11) noch zur Verfügung. Das Signal kann über den unbenutzten Pin des Steckers CNB dem I/O-Board zugeführt werden.

Die Sonderausführung des Adapterboards für den GENIE IIs entspricht prinzipiell auch der angegebenen Schaltung. Jedoch wurden teilweise andere Gatter innerhalb der ICs benutzt, so daß die Pin-Nummern in der Schaltung nicht überall zutreffen.

Wieviele WAITs der einzelne Computer beim HD64180 und höherer Taktrate beibehalten muß, ist im Einzelfall zu untersuchen. Es ist durchaus nicht in jedem Computer ein Takt von 9,216MHz möglich, auch wenn alle 4 WAITs bei Memory-Zugriffen beibehalten werden.

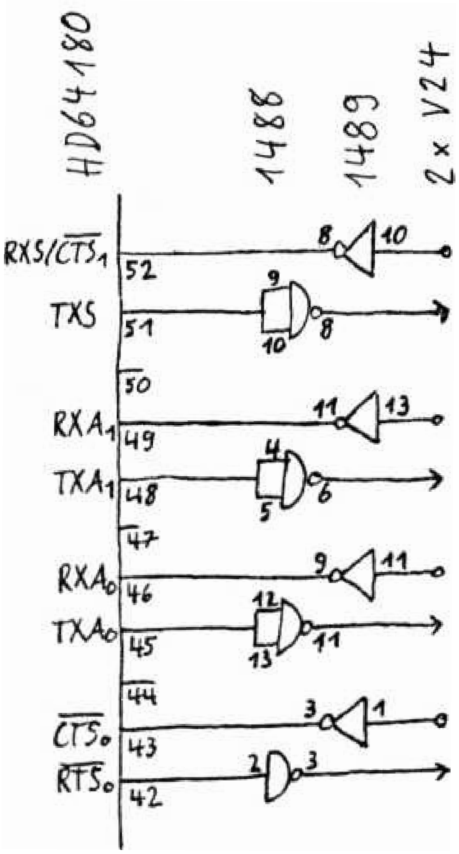
<u>Und wie geht es in meinem Computer?</u>

Auf diese Frage hin muß ich nun doch wieder auf die allgemeingültigen Bemerkungen zurückkommen. Ich habe weder das Geld, noch die Zeit, noch überhaupt die Lust, diesen Hardware-Patch in allen mir zugänglichen Computern auszuprobieren, um dann für alle Geräte einfach nachzuvollziehende Lötanleitungen vorzulegen.

Ein bißchen Abenteuer und Eigenleistung und dadurch auch mehr Spaß an der Sache soll denen bleiben, die sich an dieses Projekt heranwagen. Als Unterstützung kann ich lediglich eine ungebohrte geätzte Platine (einer der beiden vorgestellten Varianten) für 10,-DM zur Verfügung stellen und eventuell fernmündlich oder -schriftlich helfen, soweit ich vom Stand meiner zwangsläufig begrenzten Hobbyistenkenntnisse und der Versorgung mit Schaltplänen der verschiedenen Geräte dazu in der Lage bin.

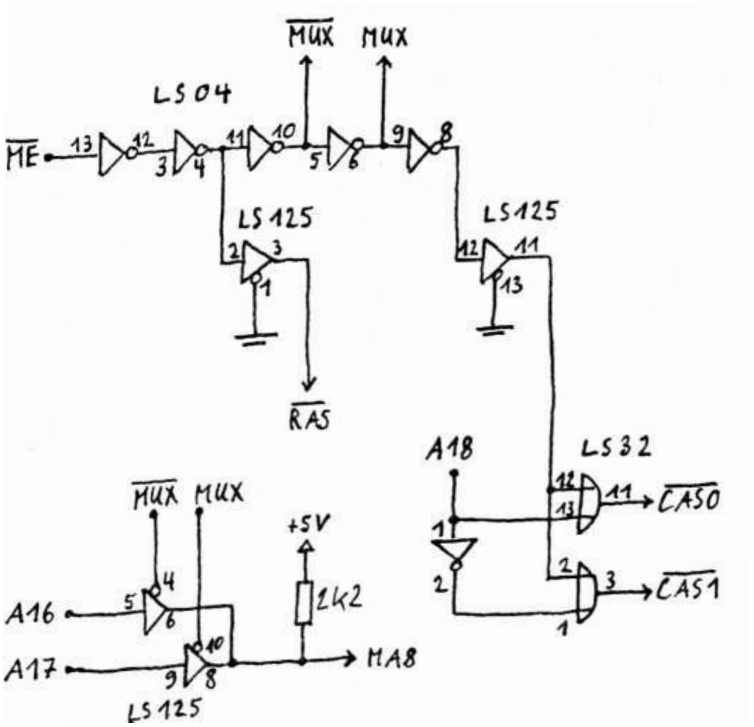
Über Einbauanleitungen des HD64180-Adapters in spezielle Computer (und dadurch indirekt über Erfolgsmeldungen) im Info würde ich mich sehr freuen.

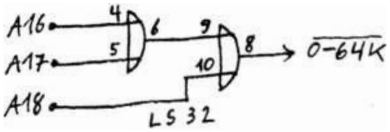
Helmut Bernhardt

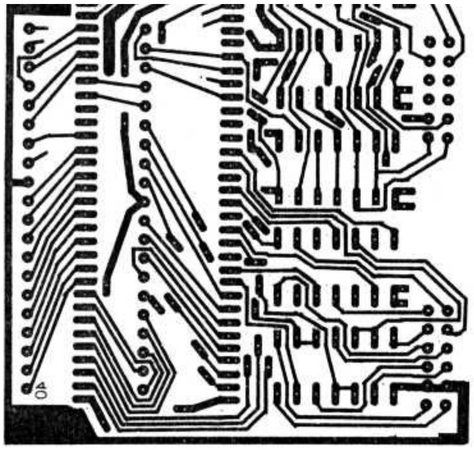


Beschaltung der Pins der seriellen Schnittstellen

Schaltung zur Ansteuerung von 512K RAM auf dem Board

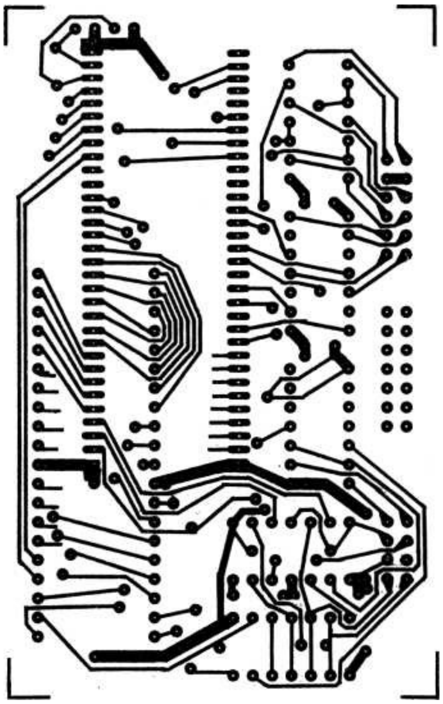


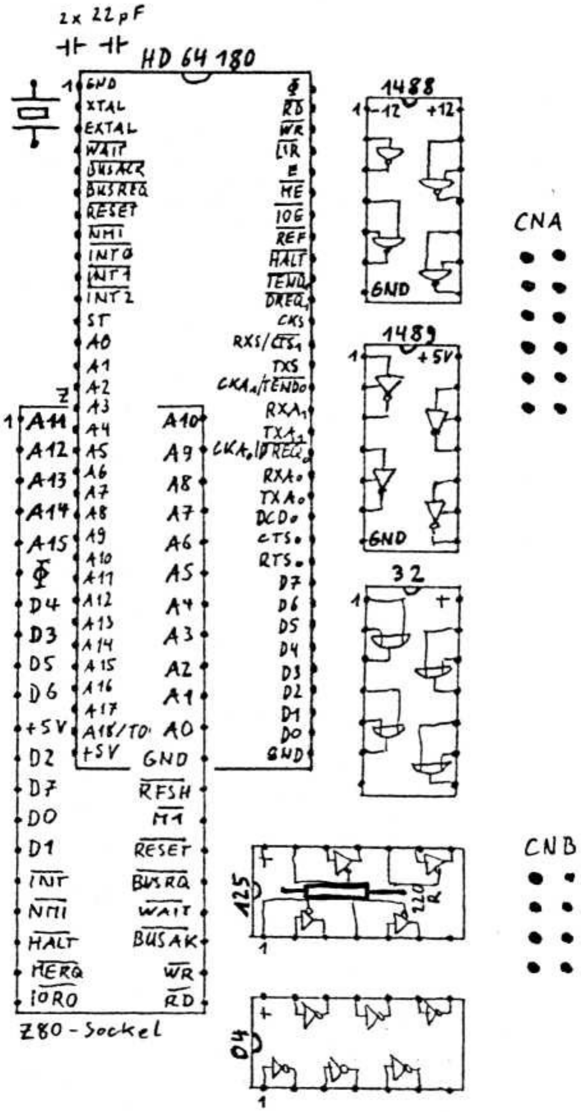


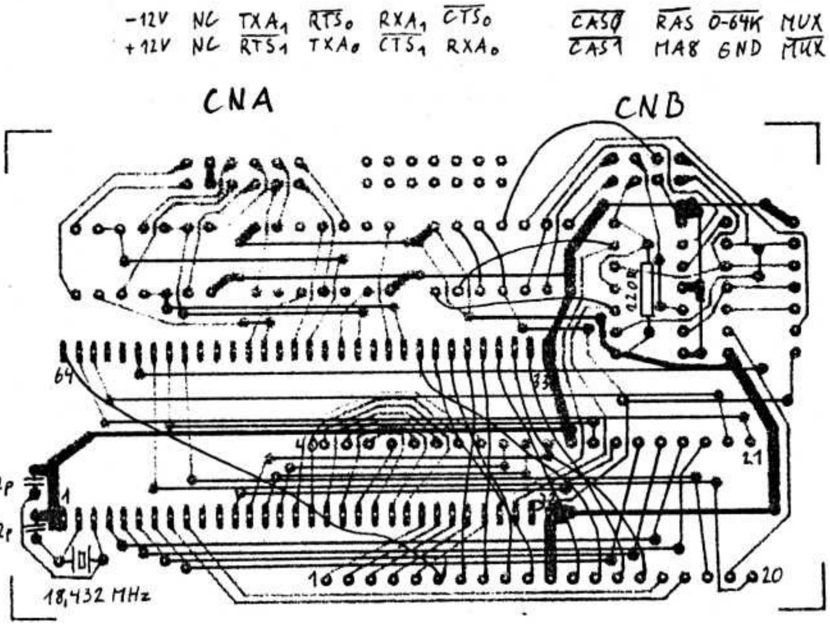


B

Sonderversion für GENIE IIs





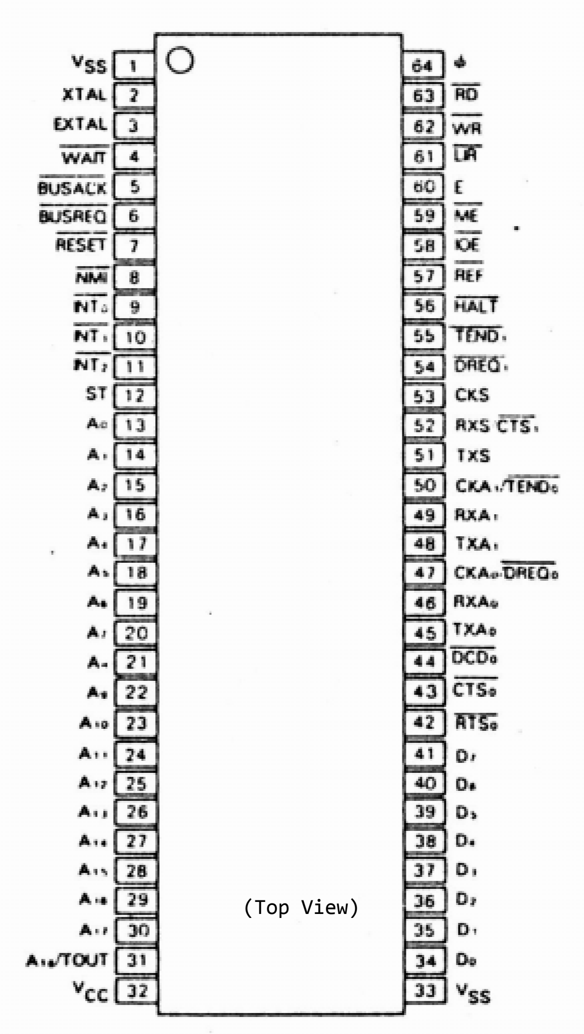


Direkt zu verbindende Pins

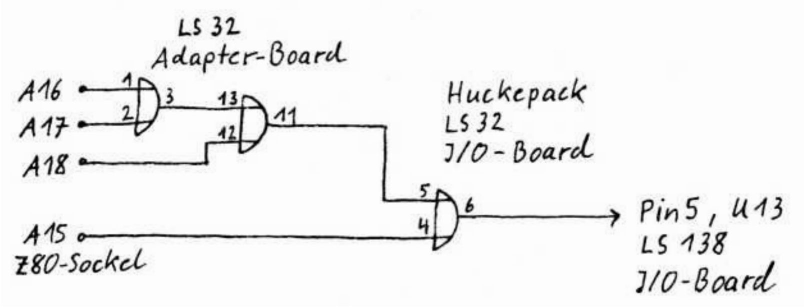
HD64180 Pin	Signal	Z80 Pin
13	A0	30
14	A1	31
15	A2	32
16	A3	33
17	A4	34
18	A5	35
19	A6	36
20	A7	37
21	A8	38
22	A9	39
23	A10	40
24	A11	1
25	A12	2
26	A13	3
27	A14	4
28	A15	5
34	D0	14
35	D1	15
36	D2	12
37	D3	8
38	D4	7
39	D5	9
40	D6	10
41	D7	13
6	BUSREQ*	25
5	BUSACK*	23
7	RESET*	26
8	NMI*	17
9	INT0* = INT*	16
59	ME* = MERQ*	19
58	IOE* = IORQ*	20
63	RD*	21
62	WR*	22
57	REF* = RFSH*	28
12	ST* = M1*	27
56	HALT*	18
4	WAIT*	24
64	PHI	6 #
32	+5V	11
1,33	GND	29

CLOCK ist beim Z80 ein Eingang und beim HD64180 ein Ausgang

Pinout der HD64180-CPU



(Top View)



Herleiten von 0-64K* auf dem Adapterboard der Sonderversion und Verknüpfen mit A15 zur Erzeugung des Signals 0-32K* für den Pin 5 von U13 auf dem I/O-Board

192 K RAM im Genie I und II

H.Bernhardt, C.Ueberschaar

Einer der wesentlichen Nachteile von 8 Bit-Mikroprozessoren gegenüber 16 Bit-Prozessoren ist die Beschränkung auf nur 16 Adreßleitungen. Der Prozessor kann damit 2^{16} = 65536 = 64 K Speicherstellen von einander unterscheiden. Von diesen 64 K werden beim Genie 16 K für ROM, Sonder-ROM, Video-RAM, Tastatur und memory mapped decodierte Peripheriegeräte benutzt. Bei angeschlossener Floppy Disk und geladenem DOS und Disk-BASIC werden je nach DOS nochmals annähernd 16 K RAM belegt. Die verbleibenden 32 K Speicherraum reichen für die meisten Anwendungen aus.

Wenn allerdings Aufgaben bewältigt werden sollen, bei denen große Datenmengen schnell gesammelt werden müssen (z.B. Messungen), miteinader verrechnet und z.B. in Form von Graphiken ausgegeben werden müssen, dann werden 32 K schnell zu knapp.

Den prinzipiellen Weg zu mehr Speicher beim Genie I und II hat der Hersteller bei der Änderung von 16 K RAM auf 48 K RAM im Grundgerät (83er Modell) aufgezeigt. Durch Ersetzen der 4116 (16 K x 1 Bit) RAM-Chips durch 4164 (64 K x 1 Bit) RAM-Chips und einige dafür nötige Änderungen, ist der gesamte vom Z80-Prozessor adressierbare Speicherbereich bereits im Grundgerät auf dem CPU-Board realisiert. Die Steckplätze für 32 K RAM im Expansionsinterface werden nicht mehr gebraucht.

Was auf dem CPU-Board möglich ist, ist aber auch im Expansionsinterface durchzuführen, indem 4116er Chips durch 4164er RAMs ersetzt werden.

Die dadurch vorhandenen 192 K RAM lassen sich allerdings nicht durch Umschalten zwischen den drei 64 K-Blöcken benutzen. Beim Umschalten vom Hauptblock in einen anderen Block wären Betriebssystem und Stack nicht mehr verfügbar. Das Umschalten darf nur einen kleinen Teil im oberen Adreßraum umfassen.

In der im Folgenden beschriebenen Lösung wurden dafür die oberen 4 K (oder 8 K; durch Software umschaltbar) also der Adreßraum F000 H bis FFFFH (bzw. E000 H bis FFFF H beim 8 K-Banking) gewählt. Das bedeutet, daß der gesamte Adreßraum von 64 K in 60 K (56 K) Grundspeicher und einen 4 K (8 K)-Bereich, der durch Software steuerbar überall in den zusätzlichen Speicher hineingelegt werden kann, unterteilt ist.

Die Schaltung

Die softwaregesteuerte Umschaltung der 4 K (8 K)-Blöcke, die bei einer Ausgabe einer Adresse in diesem Speicherbereich jeweils angewählt werden, erfolgt über ein 8 Bit-Latch mit ENABLE und CLEAR 74 116, das durch die Portadresse F 0 H decodiert ist (Abb.1). Durch einen OUT-Befehl läßt sich in das Latch ein 8 Bit-Wort eintragen, das dann ständig an den Ausgängen anliegt und zur Selektion eines von maximal 256 4K (8 K)-Blöcken herangezogen werden kann. Der Eingang CLEAR wird an die RESET-Leitung des Systems angeschlossen und gewährleitet dadurch, daß beim Einschalten des Systems immer alle Bits des Latch auf Low gesetzt sind.

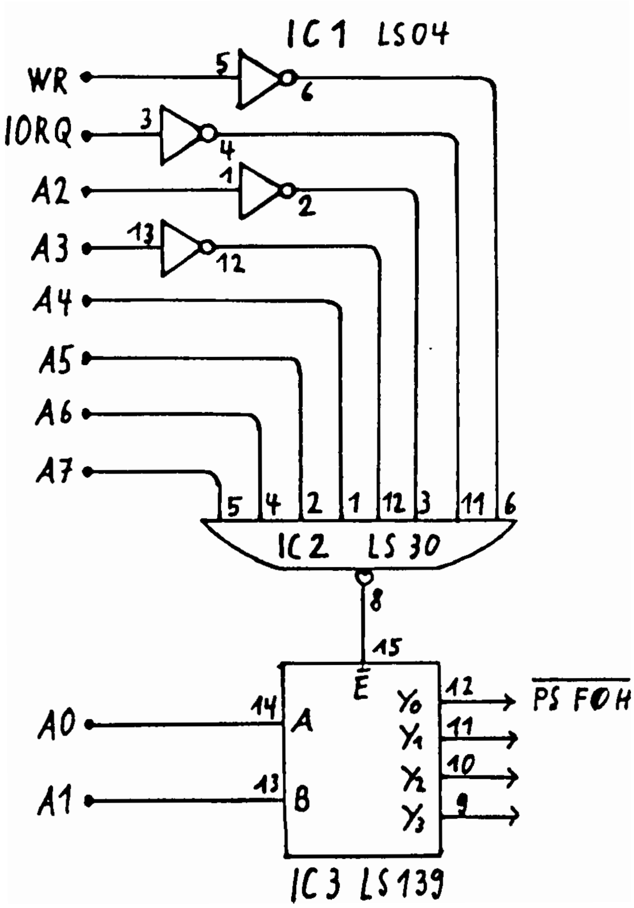


Abb. 1 Decodieren der Portadresse F0H (F1H, F2H, F3H)

Das 8 Bit-Wort im Latch wird benutzt, um einerseits einen der 3 im Gerät vorhandenen 64 K-Blöcke anzuwählen und um andererseits im angewählten Block einen bestimmten 4 K (8 K)-Bereich zu decodieren.

Die niederwertigen 4 (3) Bits im Latch werden anstelle der Adreßleitungen A 12 (A 13) bis A 15 an die Adreßmultiplexer der RAMs gelegt und adressieren damit einen 4 K (8 K)-Bereich im angewählten 64 K-Block.

Die höherwertigen 4 (5) Bits werden zur Auswahl eines von maximal 16 (32) Blöcken mit je 64 K benutzt. Auf diese Art lassen sich mit dem 8 Bit-Wort im Latch insgesamt 1 M Byte (2 M Byte) Speicher verwalten. Hier wird aber nur von drei 64 K-Blöcken Gebrauch gemacht, weil für diese nach Ersetzen der 4116er RAMs durch 4164er RAMs und einige geringfügige Änderungen die nötigen Steckplätze im System vorhanden sind. Die Schaltung liefert aber die Auswahlimpulse für insgesamt 32 solcher 64 K-Blöcke. Mit geeigneten Speicherkarten läßt sich das System auf 2 M Byte Speicher erweitern.

Der Blockauswahlimpuls, der gewonnen wird, wenn alle 4 (5) höherwertigen Bits des Latch 0 sind und der Prozessor eine Adresse oberhalb F000 H (E000 H) ausgibt, muß den RAM-Block auf dem CPU-Board anwählen. Dieser Block muß aber auch angewählt werden, wenn eine Adresse unterhalb F000 H (E000 H) ausgegeben wird. Die Schaltung muß also erkennen, ob eine Adresse unterhalb des umschaltbaren Bereichs anliegt, und dann die normalen Adreßleitungen A12 (A 13) bis A 15 an die Adreßmultipexer der RAMs durchschalten, oder ob eine Adresse im umschaltbaren Bereich anliegt, und dann die niederwertigen 4 (3) Bits des Latch als Adresse an die Multiplexer der RAMs anlegen.

Diese Aufgabe wird durch ein 4 Input NAND 74 LS 20 (IC 10), dessen Ausgang immer dann low ist, wenn A 12 (A13) bis A 15 alle gleichzeitig high sind, und einen Multiplexer 74 LS 157 (IC 15), dessen SELECT Eingang je nach Zustand des NAND entweder die unteren Bits des Latch oder die System-Adreßleitungen A 12 bis A 15 durchschaltet.

Wenn die an das NAND-Gatter angelegten Adreßleitungen alle high sind (also der obere, umschaltbare Adreßbereich angewählt wird, dann werden vom Multiplexer die Ausgänge des Latch durchgeschaltet, wenn auch nur eine dieser Adreßleitungen low ist, werden die normalen Adreßleitungen vom Multiplexer durchgeschaltet (Abb. 2).

Die Umschaltung zwischen Banking von 4 K und von 8 K erfolgt über das Bit 0 eines anderen Latch. Ein zweiter Baustein 74 116 wird dazu in zwei 4 Bit-Latches unterteilt. Beide Hälften werden durch unterschiedliche Portadressen Pins 11 und 10 von IC 3) decodiert. Die Hälfte, die durch die Portadresse F2 H decodiert wird, dient als Software-Schalter. Der Bit 0-Ausgang wird an die SELECT-Eingänge von 3 weiteren 74 LS 157 Multiplexern gelegt und kann so die Durchschaltung von je einem von 2 Eingängen zum Ausgang des Multiplexers steuern.

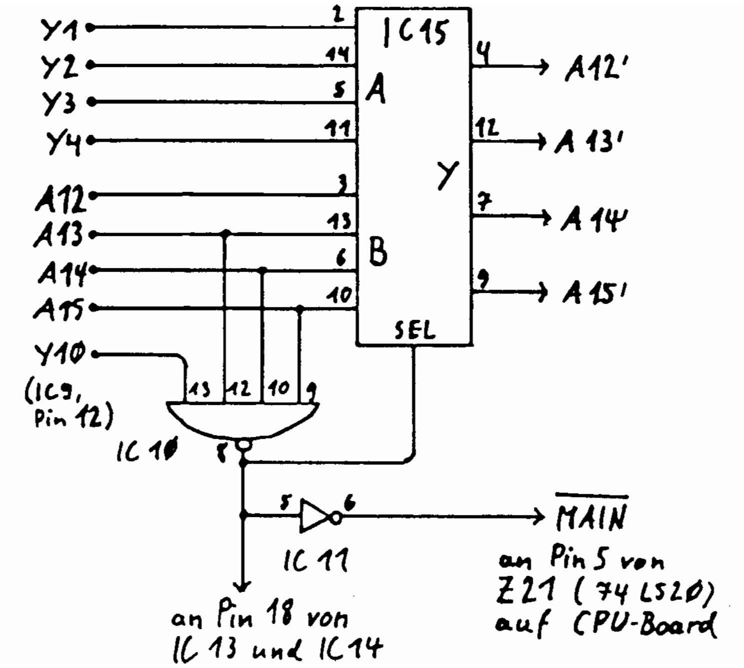


Abb. 2

Multiplexern der Adressen A 12 - A 15 und der unteren Bits des Latch. Wenn A 12 (A 13) bis A 15 alle gleichzeitig high sind, werden die unteren Bits des Latch durchgeschaltet .

An den A-Eingängen der beiden Multiplexer IC 7 und IC 8 liegen die 8 Bit des Latch F0 H an. Diese 8 Bit werden dann direkt durchgeschaltet, wenn das Bit 0 des Steuerlatch F2 H low ist (4 K-Banking). An den B-Eingängen der Multilexer liegen ebenfalls die Bits 0 bis 6 des Latch F0 H an, nur, daß jedes Bit um eine Position zum höheren Stellenwert verschoben ist. Wenn das Bit 0 des Steuerlatch F2 H high ist (8 K-Banking), erscheint das Wort des Latch an den Ausgängen der Multilexer arithmetisch links geshiftet. Das Bit 7 des Latch F0 H liegt dann am Ausgang 1 des dritten Multipexers IC 9 an. Am niederwertigsten Ausgang von IC 7 liegt dann der Pegel der Adreßleitung A 12 an (Abb.3).

Die Adreßleitungen A 12 und A 13 werden durch den dritten 74 LS 157 (IC 9) ebenfalls gemultiplext, so daß beim 4 K-Banking A 12 an den vierten Eingang des 4 Inputs NAND Gatters (IC 9) anliegt, während beim 8 K-Banking A 13 an diesen Eingang gelegt wird und A 12 an den B-Eingang des Multiplexers, um bei Zugriff auf den Grundspeicherbereich 0 - 56 K als niederwertigstes Bit neben den 8 Bit des Latch durchgeschaltet zu werden. Die Daten an den Ausgängen der Multiplexer IC 7 bis IC 9 beim 4 K-Banking und beim 8 K-Banking werden in Tabelle 1 dargestellt.

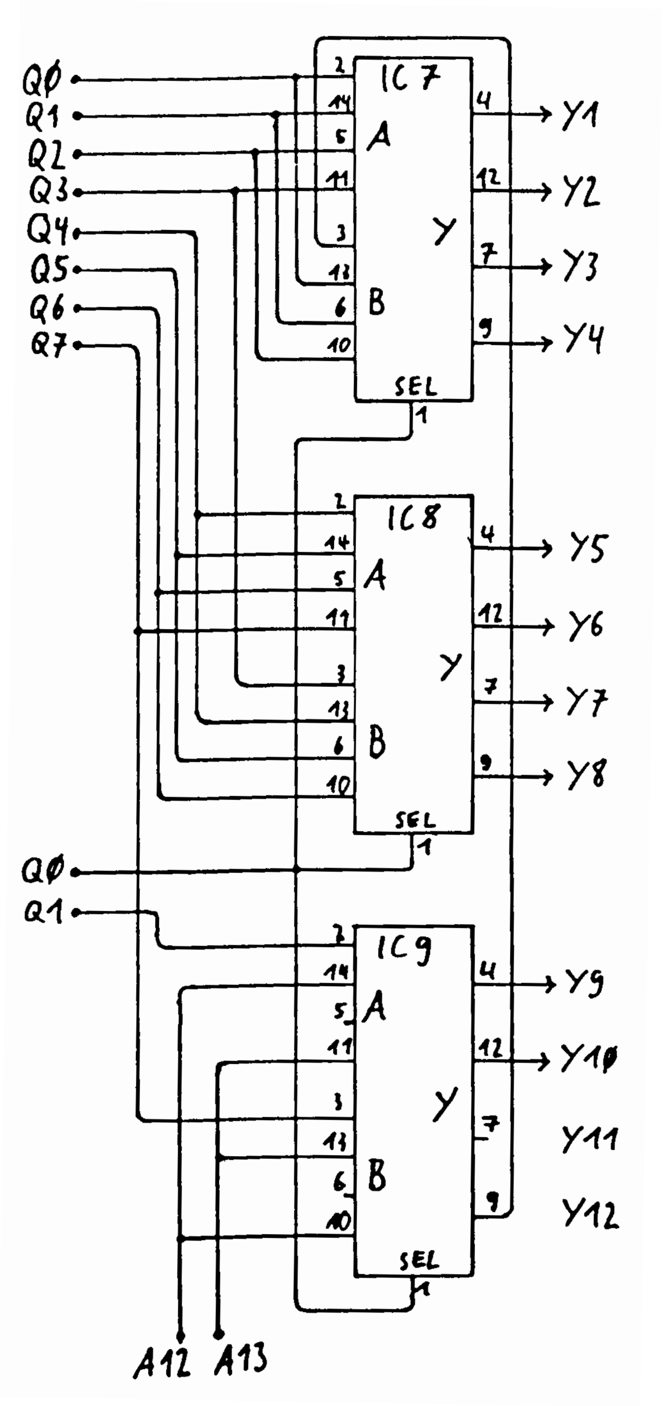


Abb. 3

Einstellen von 4K- oder 8K- -Banking über Bit 0 des Steuerlatch F2H (Signal Q0). Die Bits des Latch werden bei 8K-Banking arithmetisch links geschiftet durchgeschaltet. A 12 wird als LSB durchgeschaltet.

In beiden Fällen werden die Multiplexerausgänge Y 1 bis Y 4 (am IC 7) zusammen mit den System-Adreßleitungen A 12 bis A 15 an den Multiplexer IC 15 gelegt. Durch Selektion des NAND-Gatters IC 10 wird eine dieser beiden Gruppen dort durchgeschaltet und ergibt die Adreßleitungen A 12' bis A 15'.

Zwingend ist die Verwendung der Signale A 12' bis A 15' nur im 64 K-Block auf dem CPU-Board, weil hier auch der nicht umschaltbare Hauptadreßraum bis 60 (56) K liegt. Alle anderen 64 K-Blöcke können auch direkt mit den

Multiplexer Ausgang	Durchgeschaltetes Signal	
	4K Banking	8K Banking
IC7 Y1 Pin 4	Latch FØH, Bit Ø	A12
Y2 " 12	" " " 1	Latch FØH, Bit Ø
Y3 " 7	" " " 2	" " " 1
Y4 " 9	" " " 3	" " " 2
IC8 Y5 Pin 4	" " " 4	" " " 3
Y6 " 12	" " " 5	" " " 4
Y7 " 7	" " " 6	" " " 5
Y8 " 9	" " " 7	" " " 6
IC9 Y9 Pin 4	Latch F2H, Bit 1	" " " 7
Y10 " 12	A12	A13
Y11 " 7	——	——
Y12 " 9	A13	A12

Tabelle 1 Von den Multiplexern (IC 7 bis IC 9) durchqe-schaltete Signale für 4K- und 8K-Banking

Ausgängen Y 1 bis Y 4 des Multiplexers IC 7 als Adresse A 12 bis A 15 beschaltet werden, weil diese Speicher je nach eingestellter Bank im Latch F0 H nur bei Ausgabe von Systemadressen oberhalb F000 H (E000 H) angesprochen werden.

Die Multiplexer Ausgänge Y 5 bis Y 8 werden an zwei 74 LS 154 (1 aus 16 Decoder) gelegt und ergeben die Block-Auswahlimpulse BS 0 bis BS 31, die den jeweiligen 64 K-RAM-Block freigeben.

Y 9 gibt jeweils einen der beiden 1 aus 16 Decoder frei. Beim 4 K-Banking muß diese Auswahl durch Setzen von Bit 1 im Steuerlatch F2 H (low: 0-1 M

Byte Bereich; high: 1-2 M Byte Bereich) erfolgen. Die prinzipielle Freigabe beider 1 aus 16 Decoder erfolgt durch das 4 Input NAND (IC 10), wenn A 12 bis A 15 (A 13 bis A 15) alle high sind.

Y 11 wird nicht benutzt. Y 10 und Y 12 steuern die Schaltung von A 12 oder A 13 an den Eingang des 4 Input NAND Gatters IC 10 und an den Eingang B 1 des Multiplexers IC 7.

Der Grundspeicherbereich, der 64 K-Block auf dem CPU-Board, der bei Low auf den Ausgängen des Multiplexers IC 8 (Y 5 bis Y 8) decodiert wird, hat dabei mehrere Funktionen. Der Adreßbereich 16 - 60 (56) K wird vom Prozessor direkt adressiert. Andererseits ist er auch in den Banking-Zugriff eingeschlossen, so daß jeder 4 K (8 K)-Bereich durch Ausgabe der Bank Nummern 0 - 15 (0 - 7) in das Latch F0 H in den oberen Banking-Adreßraum F000 H (E000 H) bis FFFF H gelegt werden kann. Das kann mit dem ohnehin in diesem oberen Adreßraum liegenden

Bereich und mit dem Bereich 0 - 16 K, der parallel zum ROM liegt, gemacht werden. Wenn allerdings Teile des Betriebssystems, des BASIC oder des Anwenderprogramms in den oberen Adreßraum gelegt werden, in den z.B. Meßdaten eingetragen werden, so überschreiben diese Daten die jeweiligen Programme.

Dieses Problem kann z.B. dadurch gelöst werden, daß entsprechende Bank Nummern nicht benutzt werden (4 — 14 beim 4 K-Banking und 2-6 beim 8 K-Banking). Sicherer ist aber eine Kontrollschaltung, die immer dann, wenn eine verbotene Bank Nummer an die RAMs durchgeschaltet wird, sofort einen RESET am Latch F0 H auslöst und darin die Bank Nummer 0 einstellt, die in jedem Fall erlaubt ist.

In Tabelle 2 sind die Bitkombinationen an den Ausgängen des Multiplexers IC 7 angegeben, die beim Banking im 64 K-Block auf dem CPU-Board erlaubt sind, und diejenigen, die verboten sind.

Es ist zu erkennen, daß für die erlaubten Bank-Nummern folgende Bedingungen für die Bitkombination gelten:

1) Y 4, Y 3, Y 2 und Y 1 müssen alle gleichzeitig high sein oder
2) Y 4 und Y 3 müssen beide gleichzeit low sein und
3) Y 5 bis Y 8 müssen alle low sein (64 K-Block Nr. 0 decodiert), weil in anderen Blöcken diese Kombinationen erlaubt sind.

4K Banking

	Bank Nr.	Y4	Y3	Y2	Y1
0-16K	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	0
	3	0	0	1	1
16-60K	4	0	1	0	0
	⋮	⋮	⋮	⋮	⋮
	14	1	1	1	0
60-64K	15	1	1	1	1

8K Banking

	Bank Nr.	Y4	Y3	Y2	Y1	
0-16K	0	0	0	0	X	erlaubt
	1	0	0	1	X	
16-56K	2	0	1	0	X	nicht erlaubt
	⋮	⋮	⋮	⋮	⋮	
	6	1	1	0	X	
56-64K	7	1	1	1	X	erlaubt

Tabelle 2 Bitkombinationen (an den Ausgängen des Multiplexers IC 7) der möglichen Bank-Nummern für den RAM-Block auf dem CPU-Board

Die "Sicherung" muß also erkennen, wenn Bedingung 3 und eine der beiden anderen Bedingungen gleichzeitig erfüllt sind. Die in Abb. 4 gezeigte einfache Schaltung erkennt diese Bitkombinationen und gibt am letzten NAND-Gatter ein Low-Signal aus, wenn eine verbotene Bank-Nummer vorliegt. Dieses low-Singal wird zum Löschen des Latch F0H benutzt. Da aber auch durch RESET das Latch gelöscht werden soll, ist eine AND-Verknüpfung (2 NAND Gatter in IC 4) von RESET und dem Signal der "Sicherung" nötig. An die beiden durch F1H und F2H decodierten Latches kann das RESET Signal direkt angeschlossen werden.

Diese Schaltung hat einen kleinen Schönheitsfehler. Beim 8 K-Banking wird auch der Adreßbereich E000 H bis EFFF H, der als untere Hälfte des oberen 8 K-Adreßbereichs durchaus erlaubt ist, als verboten erkannt (Y 1 = A 12 = 0 ; damit sind Y 1 bis Y 4 nicht alle gleichzeitig high). Diese 4 K Byte werden dabei zugunsten erhöhter Sicherheit verschenkt.

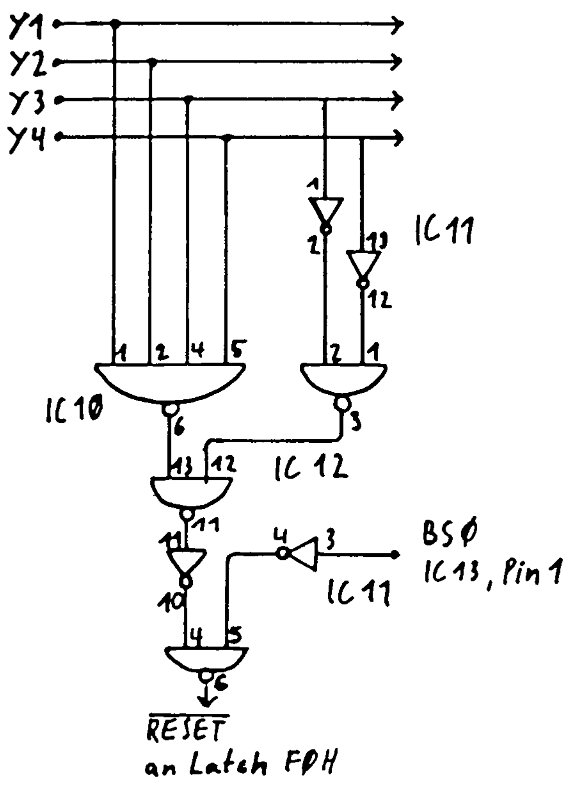


Abb.4

Sicherungsschaltung zur Verhinderung verbotener Bank-Nummern

Vereinfachte Schaltung für beschränkte Anwendung

In vielen Fällen wird ein Umschalten von 4 K- auf 8 K-Banking durch Software gar nicht gefordert sein. Es genügt hier eine einmalige in Hardware vorgegebene Einstellung auf entweder 4 K- oder 8 K-Banking.

Weiterhin kann davon ausgegangen werden, daß 1 M Byte Speicherplatz ausreichen.

Wahrscheinlich kann auch die weiter unten noch beschriebene Bankumschaltung des Adreßbereichs des Sonder-ROM (3000 H - 36FF H) entfallen.

Es kann dann auf die ICs 6,7,8,9 und 14 der Banking-Schaltung verzichtet werden. Anstelle der ICs 7 bis 9 können die Eingänge A (für 4 K-Banking) oder B (für 8 K-Banking) direkt per Drahtbrücken in der Banking Karte an die Ausgänge Y geschaltet werden.
Der Ausgang Y 9 ist an GND zu schalten.

Bei Verzicht auf IC 14 (Verwaltung des Bereiches 1 - 2 M Byte) kann auch das Steuerlatch F2 H (IC 6) entfallen. Damit entfällt aber auch die Möglichkeit, mit dem Latch F1 H eine Umschaltung des Adreßbereichs des Sonder-ROM zu verwalten.

Auch in dieser abgemagerten Version kann die Schaltung beim 83er Modell ohne herkömmliches Expansionsinterface mit Speichersteckplätzen den Zugriff auf die 16 K RAM, die parallel zum ROM liegen und deshalb bisher nicht verfügbar waren, ermöglichen.

Notwendige Änderungen im Genie

Beim Genie Mikrocomputer (82er Modell) bestehen im Grundgerät 16 K RAM und im Expansionsinterface 32 K RAM in Form von 8 und 16 Stück 4116 Chips (16 K x 1 Bit). Beim 83er Modell sind die 8 Chips auf dem CPU-Board durch 4164 (64 K x 1 Bit) Chips ersetzt worden. Der Unterschied zwischen beiden Typen besteht darin, daß die 4116er RAMs die drei VersorgungsSpannungen + 5 V (V_{CC}), + 12 V (V_{DD}) und - 5 V (V_{BB}) benötigen, während die 4164er Chips nur noch + 5 V benötigen (Abb.5). Um in den gleichen Sockeln 4164er RAMs benutzten zu können, müssen die drei Versorgungsspannungszuführungen zu den RAM-Blocks abgetrennt werden. + 5 V ist nun an die Pins 8 der RAMs anzulegen, während an die Pins 9 (bisher beim 4116 + 5 V) A 7 anzulegen ist. Die Bezeichnung der Adreßleitungs-Pins im Pinout der beiden RAM-Typen ist nicht zu verwechseln mit den Adreßleitungen des Systembus A 0 bis A 7. An die Pins A 0 bis A 7 der RAMs werden die durch Multiplex nacheinander durchgeschalteten Gruppen A 0 bis A 7 und A 8 bis A 15 des Systembus gegeben.

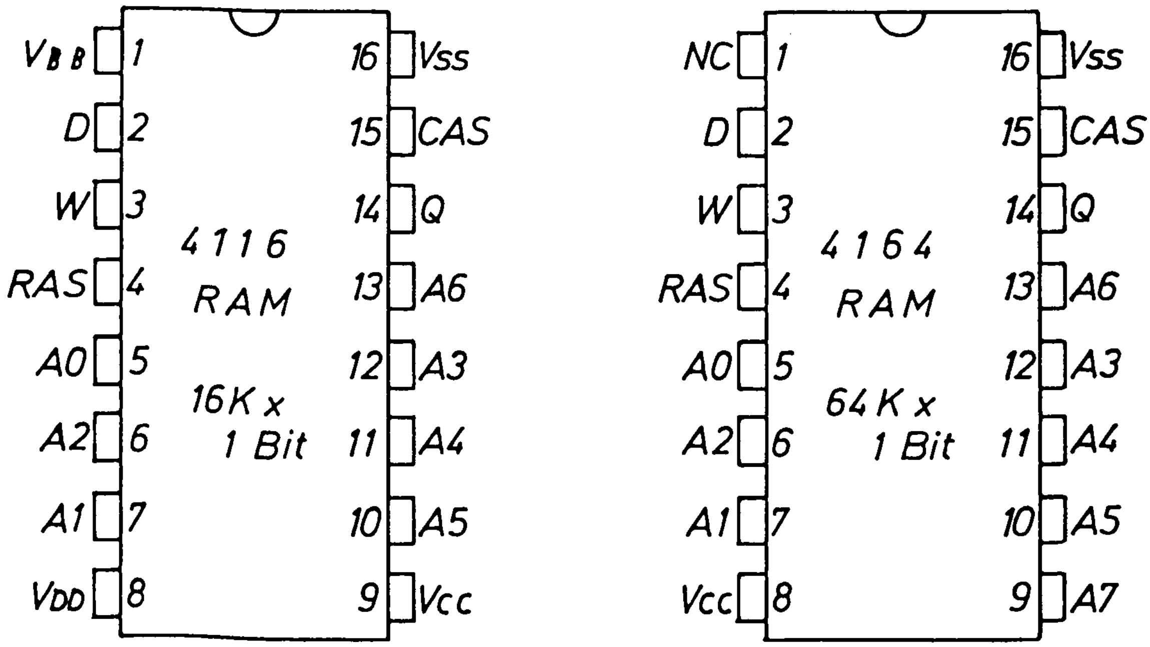


Abb. 5 Pinout der RAM-Typen 4116 und 4164

Gegenüber der Belegung der Multiplexer 74 LS 157 bei Benutzung von 4116 RAMs, wo A 0 bis A 6 und A 7 bis A 13 als Gruppen nacheinander durchgeschaltet werden, muß bei Benutzung von mehr als nur einem 64 K-Block wegen des Refreshings unbedingt A 7 mit A 0 bis A 6 zusammen durchgeschaltet werden, während A 8 bis A 13 zusammen mit den dazukommenden Adressen A 14 und A 15 durchgeschaltet werden müssen. Die nötigen Änderungen an den Multiplexern im Grundgerät und im Expansionsinterfache sind in Abb.6 und Abb. 7 gezeigt.

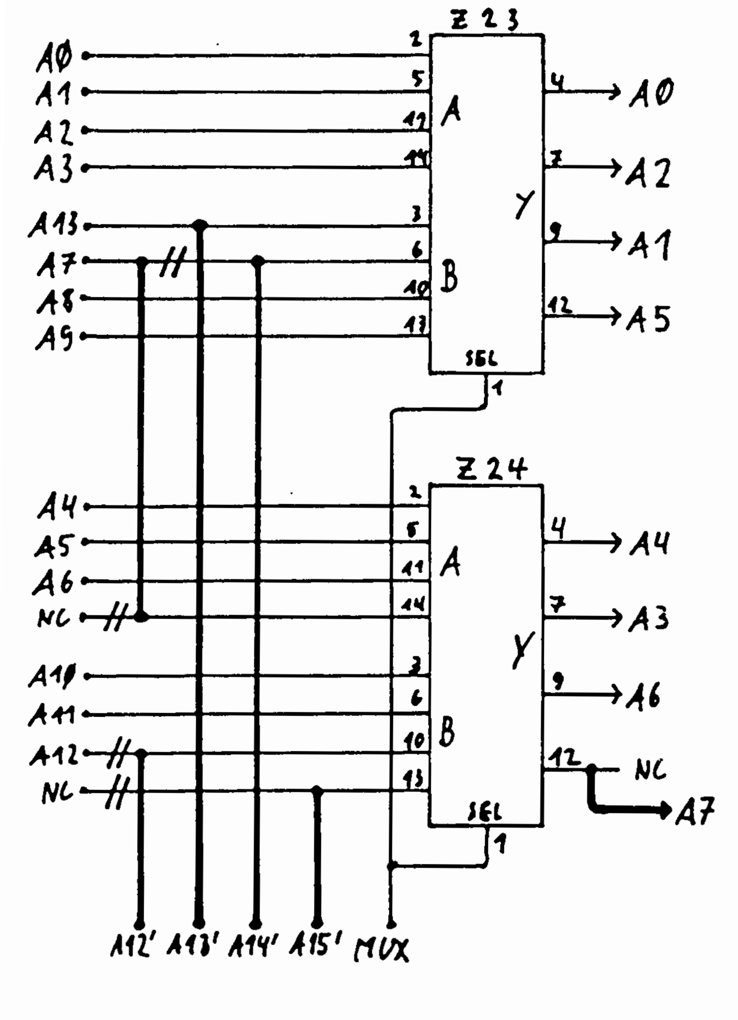


Abb. 6

Änderung der Beschaltung der Adreß-multipexer des RAM-Blocks auf dem CPU-Board

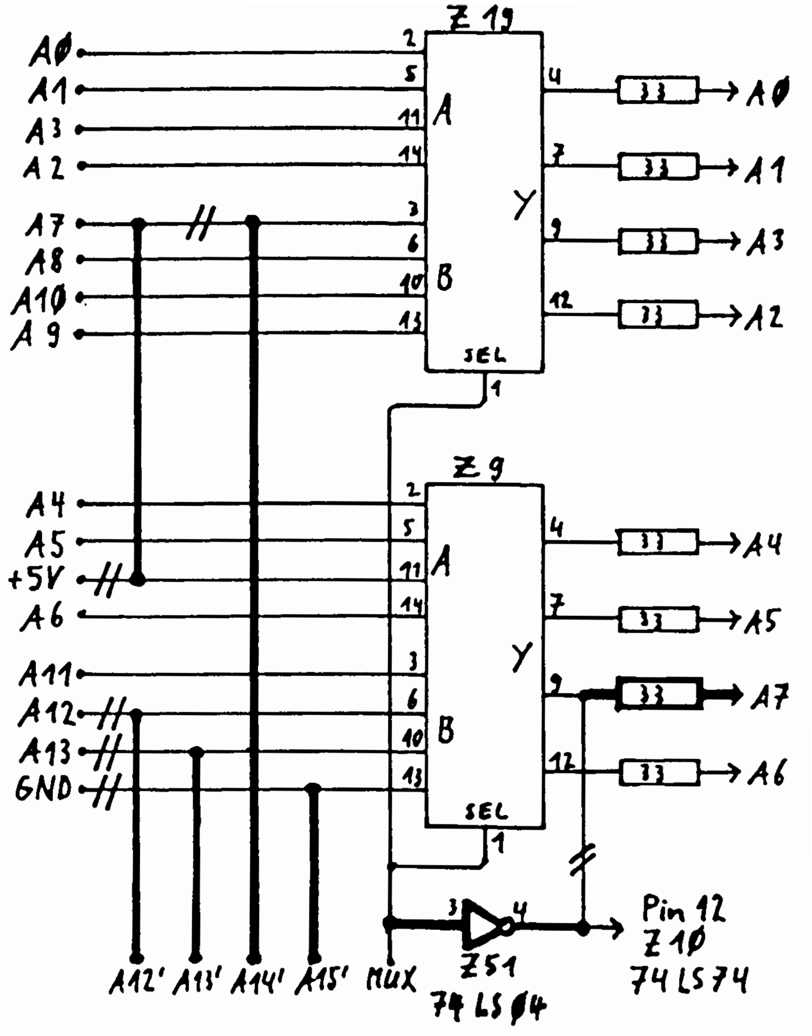


Abb. 7

Änderung der Beschaltung der Adreß-multiplexer der RAM-Blocks im Expansions-Interface

Anstelle der Systembus-Adressen A 12 bis A 15 sind zumindest auf dem CPU-Board die Signale A 12' bis A 15' der Banking-Schaltung zu benutzen. Auf dem Board im Expansionsinterface und auf eventuell dazukonmenden weiteren Speicherkarten können auch die Signale Y 1 bis Y4 von IC 7 anstelle von A 12' bis A 15' verwendet werden.

Anstelle der bisherigen Blockauswahlinpulse 32 K und 48 K (Pins 6 und 7 von Z 29, 74 LS 139) sind im Expansionsinterface die 64 K-Blockauswahlimpulse BS 1 und BS 2 (Pins 2 und 3 von IC 13, 74 LS 154) der Banking-Schaltung an Pins 1 und 2 von Z 32 (74 LS 00) und Pins 10 und 12 von Z 22 (74 LS 32) zu legen. Die Pins 9 und 10 von Z 37 (74 Ls 20), an denen auch die Signale 32 K und 48 K lagen, müssen beide mit dem Signal von Pin 8 des IC 12 74 LS 00) der Banking-Karte beschaltet werden. Die entsprechenden Verbindungen zwischen Z 29 und Z 37 und zwischen Z 37 und (Z 22 und Z 32) müssen getrennt werden.

Die Auswahl des 64 K Blocks auf dem CPU Board und auch der ROMs soll unter mehreren Voraussetzungen möglich sein:

1) Wenn eine Adresse oberhalb von F000 H (E000 H beim 8 K-Banking) auf dem Systembus liegt und der 64 K-Block 0 eingestellt ist (und beim 4 K Banking Bit 1 im Steuerlatch F2 H low ist):
   Dann liegt an Pin 1 des 1 aus 16 Decoders IC 13 (74 LS 154) Low-Pegel an (BS 0= 0).

2) Wenn eine Adresse unterhalb F000 H) vom Prozessor ausgegeben wird:
   Dann ist der über IC 11 invertierte Ausgang Pin 8 des 4 Imput NAND Gatters IC 10 low aktiv.

Diese beiden Signale werden an die Pins 1 und 2 sowie 4 und 5 von Z 21 (74 LS 20) auf dem CPU-Board gelegt. Die bisherigen Anschlüsse 0 - 16 K und 16 - 32 K (Pins 12 und 11 vom Z 25, 74 LS 139 im 82er Modell) oder 0 - 16 K und dessen Inversion (über Z 36, 74 LS 0 4) 16 - 64 K (Pin 12 vom Z 25 und Pin 2 von Z 36, 74 LS 04 im 83er Modell) werden abgetrennt.

Die NAND Verknüpfung der Signale BS0(IC 13, Pin 1) und MAIN = 0-60 (56) K (IC 11, Pin 8) über IC 12, Ausgang Pin 8 wird mit dem in der Banking Schaltung in der zweiten Hälfte des IC 3 (74 LS 139) aus A 14 und A 15 gewonnenen Signale 0-16 K im IC 4 NAND-verknüpft und ergibt das Freigabesignal CASEN für den CAS-Treiber Z 37 (74 LS 367) Pin 15 auf dem CPU-Board (Abb.8).

Der 64 K-RAM-Block auf dem CPU-Board ist also nur dann verfügbar, wenn entweder eine Adresse im Bereich 16 - 60 (56) K ausgegeben wird, oder wenn eine Adresse oberhalb von F000 H) ausgegeben wird und gleichzeitig eine Bank 0 ins Latch F0H eingetragen ist und beim 4 K Banking Bit 1 im Steuerlatch F2 H low ist. Die Gesamtschaltung ist in Abb. 9 dargestellt. Sie läßt sich bequem auf einer Europakarte unterbringen.

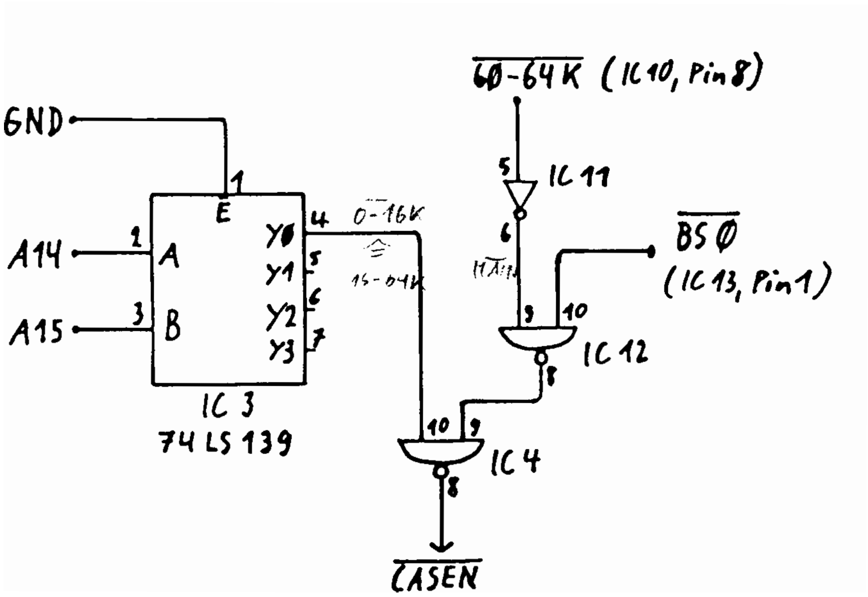


Abb. 8 Erzeugung des CAS-Treiber-Freigabe-signals CASEN für den 64K-Block auf dem CPU-Board

Anstelle eines Layouts, das entweder doppelseitig sein müßte oder viele Drahtbrücken aufweisen müßte, soll hier nur eine Wiedergabe einer günstigen Anordnung der ICs und deren Verdrahtung (durch Drahtverbindungen auf der Rückseite einer Lochrasterplatine oder ein Teil-Layout für ein einseitiges Board mit entsprechenden Drahtbrücken auf der Bestückungsseite) gezeigt werden (Abb. 10). Danach kann sicher jeder je nach den zur Verfügung stehenden technischen Mitteln die Karte selbst aufbauen.

Zweckmäßig ist ein Anschluß an den S-100-Bus-Stecker auf der Lötseite des Boards im Expansionsinterface (Achtung ! Die Belegung der Kontakte entspricht nicht der des 50poligen Platinenrandsteckers auf der Rückseite des Grundgerätes. An den Pins 1,3 und 5 liegen z.B. die Spannungen + 16 V, - 16 V und + 8 V an, mit denen man sehr viel Schaden anrichten kann). Die Versorgungsspannung + 5 V muß an geeigneter Stelle auf dem Board selbst abgegriffen werden.

Anwendung

Die Schaltung ist so konstruiert, daß volle Konpatibilität zum TRS 80, Modell 1 erhalten bleibt. Alle Programne laufen so, wie ohne diese Hardware-Erweiterung. Der Anwender, der nichts davon weiß, merkt die Änderung nicht, es sei denn, er gibt Daten über die Ports F0 H und F2 H aus.

Abb. 9 Prinzipschaltbild der Bankingkarte

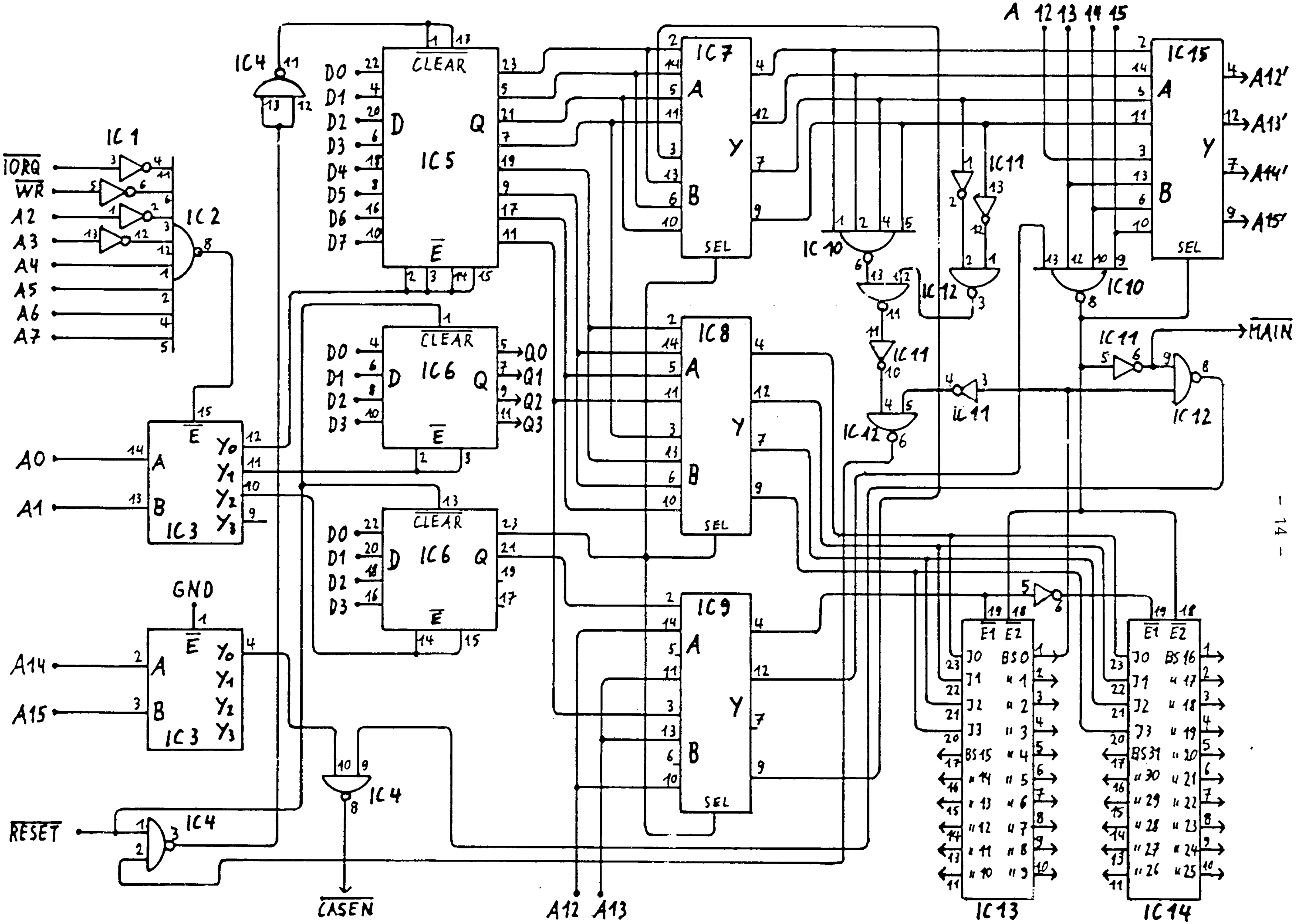
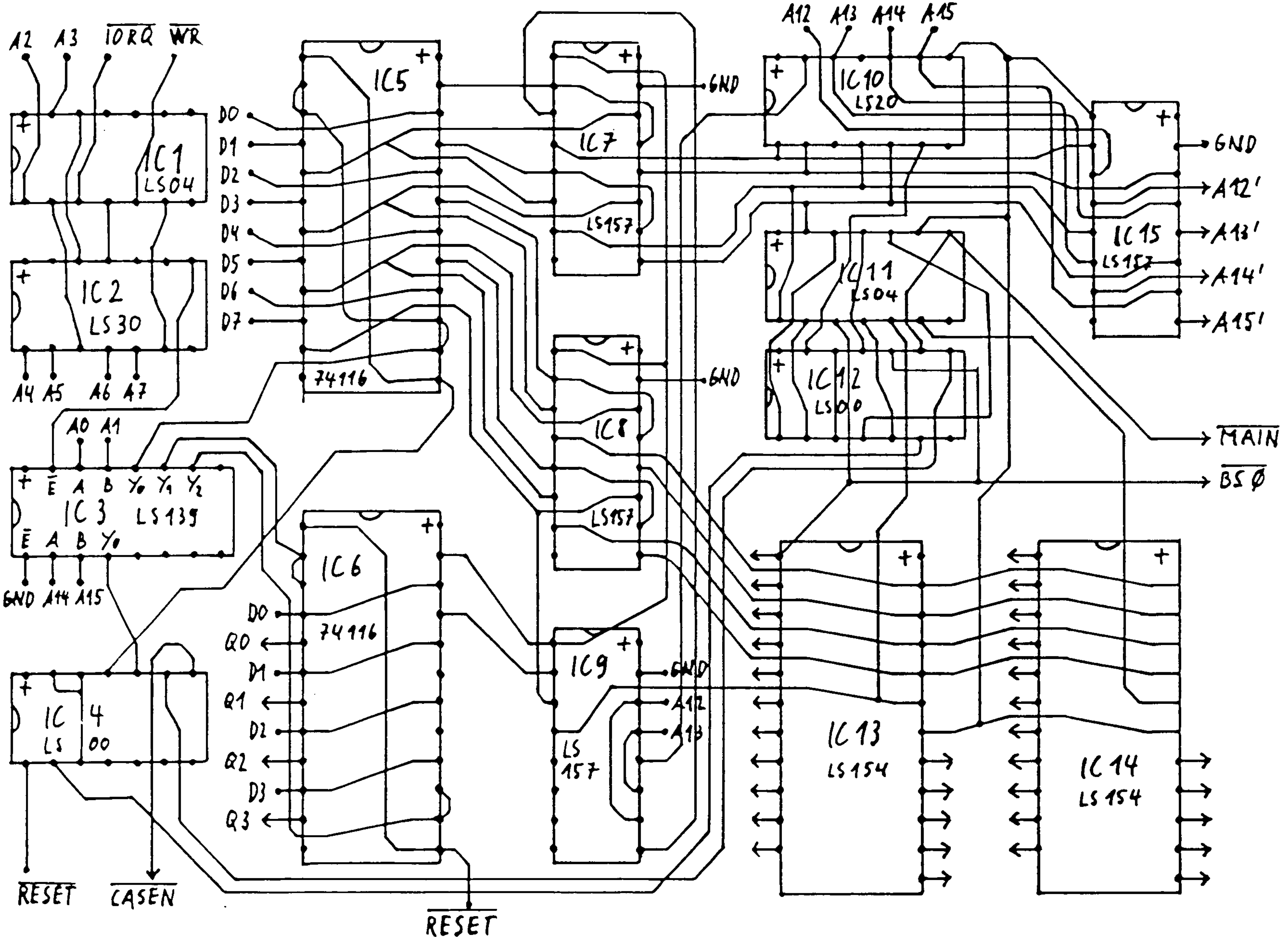


Abb . 10 Bestückungs- und Verdrahtunqsplan



Allerdings wird der erweiterte Speicher vom DOS und BASIC nicht automatisch mitbenutzt. Um den vollen Speicherplatz vom BASIC aus zu nutzen, können in den einzelnen Banken entweder Maschinenprogramme gespeichert sein, die nach Einschalten der entsprechenden Bank aufgerufen werden können, oder BASIC kann in den Banken Daten durch POKE-Befehle ablegen und durch PEEK-Befehle wieder lesen.

Die Speicherung von Variablen oder gar BASIC-Programmtext in diesen Bereichen wird nicht unterstützt. Vor dem Benutzen der Banken muß zu Anfang HIMEM bzw. MEMORY SIZE auf einen Wert unter 60 (56) K gesetzt werden, um keine Stringvariablen oder eventuell auch numerische Variablen durch Bankumschaltung verschwinden zu lassen.

Banking des Adreßbereiches des Sonder-ROMs

Nachdem nun die Verwaltung von bis zu 2 Megabyte dynamischer RAMs realisiert wurde, soll noch kurz erwähnt werden, daß von IC 6 (74 116) nur eine Hälfte als Steuerlatch (Portadresse F2 H) benutzt wird. Die andere Hälfte mit ebenfalls einem 4 Bit-Latch wurde durch die Portadresse F1 H decodiert. Hier stehen noch einmal 4 Bit zur Verfügung, die an einen 1 aus 16 Decoder angeschlossen werden können und dann 16 Freigabe-Signale erzeugen können.

Diese Möglichkeit erlaubt es prinzipiell, den Speicherbereich des Sonder-ROM 3000 H bis 36FF H zu vervielfachen, indem das Freigabe Signal für das ROM zur Freigabe des 1 aus 16 Decoders benutzt wird. Der Decoder liefert die 16 Freigabesignale für 2716 EPROMs und/oder pinkompatible 6116 CMOS RAMs.

Durch die Beschaltung des Sonder-ROM im Genie ist die Verwendung aller Signale an den Pins dieses Sockels aber nicht möglich.

1) Der WE-Eingang, Pin 21, der beim 6116 RAM für Speicherschreibeoperationen auf low gehen muß, ist über einen Widerstand an + 5 V angeschlossen.

2) Der CE -Eingang ist direkt an GND angeschlossen.

3) Am OE -Eingang liegt das 12-14K-Signal (von Pin 5, Z 22, 74 LS 156), das auch bei Adressen im Bereich 3700 H - 37FF H einen Zugriff auf das ROM ermöglicht.

4) Das Ausblenden des Bereiches 3700 H bis 37FFH erfolgt nicht durch die OE -Freigabe sondern durch die Freigabe der Speicherlesetreiber. Dies wird durch eine im Schaltplan des Service Manuals nicht dokumentierte Schaltung (Abb. 11) erreicht.

5) Die Datenleitungen sind über unindirektionale Speicherlesetreiber angeschlossen, so daß die Verwendung von RAMs nicht ohne Änderungen möglich ist.

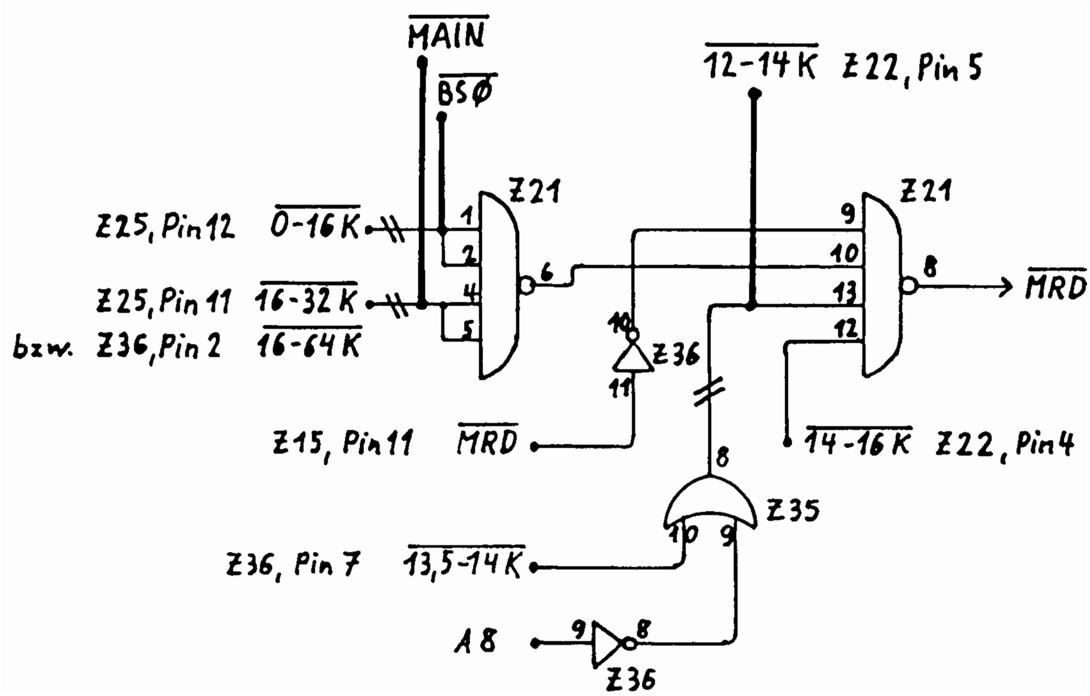


Abb.11 Ausblenden des Speicherbereiches 3700 H — 37FF H im Sonder—ROM über die Freigabe des Speicherlesetreibers (im Service Manual nicht dokumentiert)

Zur Benutzung von RAMs dürfen die Datenleitungen nicht über die unidirektionalen Speicherlesetreiber an das Speicherchip geführt werden. Anstelle dieser bisherigen Anschlüsse an den Sonder-ROM-Sockel (Pins 9-11, 13-17) müssen die Signale D 0bis D 7 des bidirektionalen Systemdatenbus verwendet werden (Abb. 12).

An die Anschlüsse OE , Pin 20 und WE , Pin 21 müssen sie Signale MRD (Z 36, Pin 11) und MWR (Z 33 Pin 3) gelegt werden.

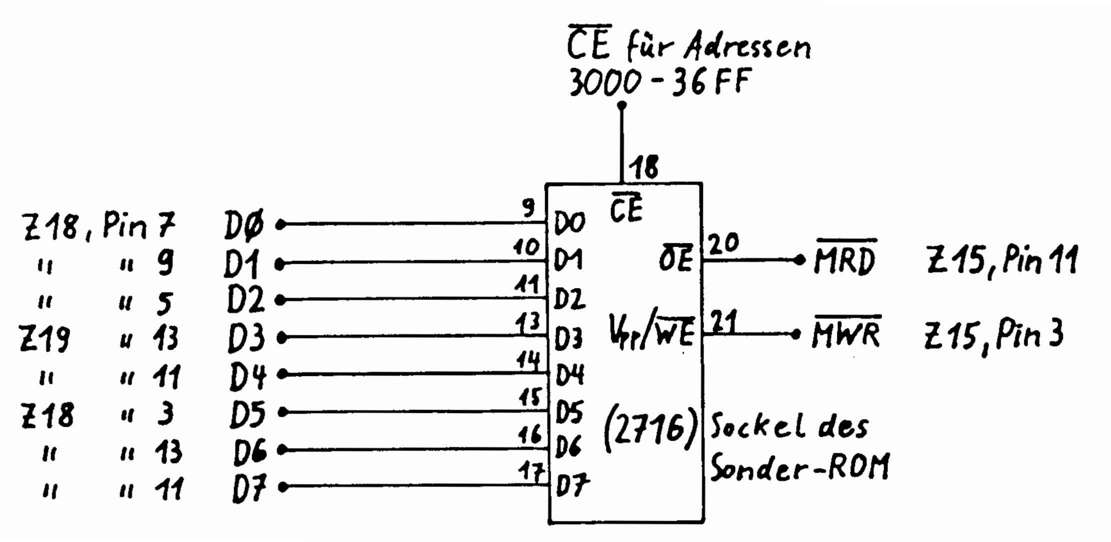


Abb.12 Änderungen am Sockel des Sonder-ROM für das Banking von wahlweise EPROMS und/oder RAMs im Adreßbereich 3000 H - 36FF H

Die Freigabe von CE, Pin 18 muß neu decodiert werden, da auf das Ausblenden des Bereiches 3700H-37FFH durch die Freischaltung der Speicherlesetreiber verzichtet werden soll. Eine einfache Erzeugung dieses Signals ist in Abb 13. gezeigt. Der dazu auf dem CPU-Board zusätzlich nötige Baustein 74 LS 08 läßt sich direkt neben Z 25 (74 LS 139) in das dort vorhandene freie Feld einsetzen.

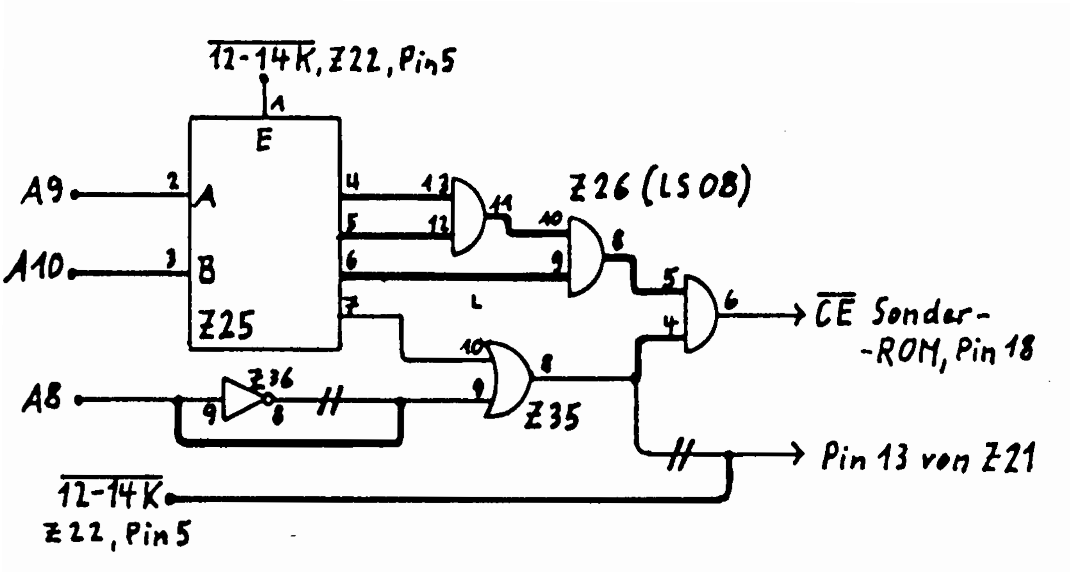


Abb.13 Neue Decodierung des Sonder-ROM-Adreßbereiches 3000 H - 36FF H und des Bereiches der memory— mapped-Divices 37XX H

Nun lassen sich alle Signale des Sonder-ROM-Sockel an eine Karte führen, auf der bis zu 16 Stück 2716 ERROMs und/oder 6116 RAMs stecken. Das nach Abb. 13 gewonnene Freigabesignal wird an die Freigabe-Eingänge des 1 aus 16 Decoders gelegt. Die Eingänge 10 bis I 3 werden mit den Ausgängen Q 0 bis Q 3 des Latch F1 H verbunden. Die Ausgänge des Decoders schalten dann das jeweils angewählte Speicherchip frei (Abb. 14)

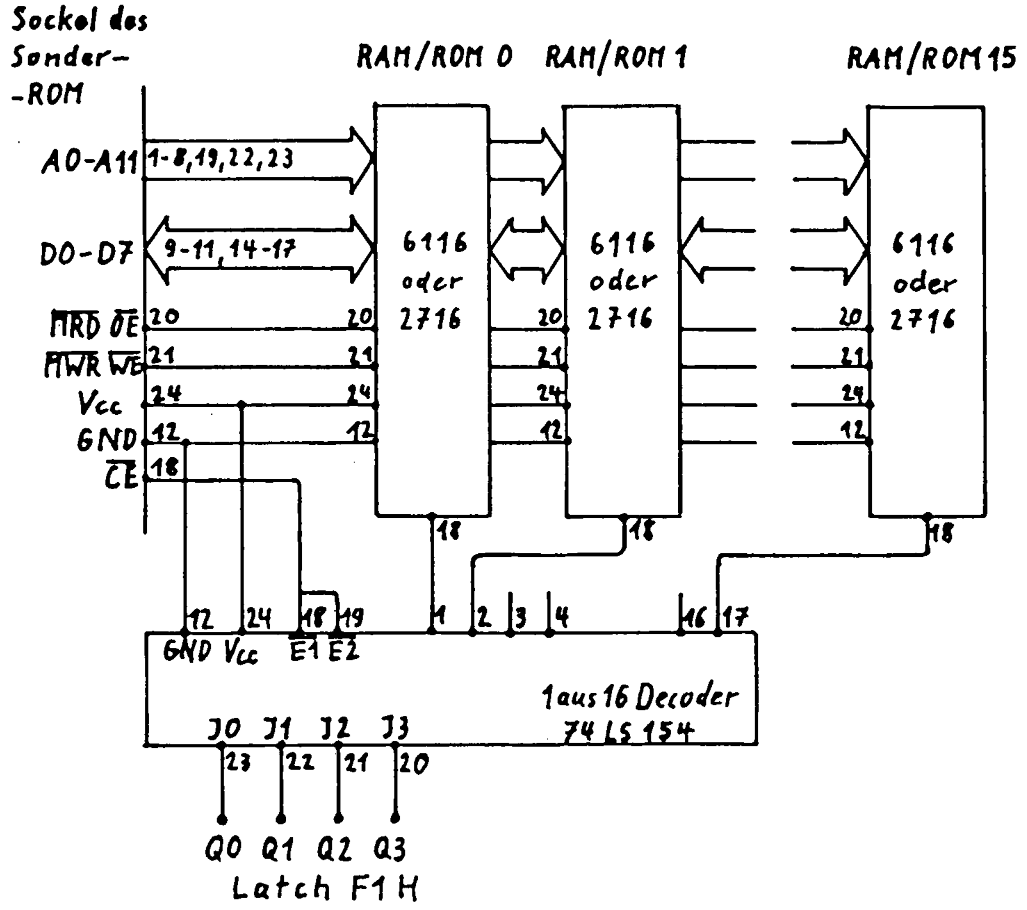


Abb. 14 Ver—16—fachung des Sonder—ROM—Speicher bereiches

Auf diese Weise läßt sich der Speicherbereich 3000 H bis 36FFH versechszehnfachen. Das ist besonders interessant, weil dadurch ein Austesten von Maschinenspracheprogrammen im RAM und späteres Benutzen im EPROM bei gleichen Speicheradressen möglich wird. Es läßt sich dadurch ein Gesamtspeicherraum von 28 K in Form von EPROM-residenten Maschinenprogrammen, Debuggern, Utilities, BASIC-Erweiterungen usw. bereitstellen, die weder bei Gebrauch erst geladen werden müssen noch den Hauptarbeitsspeicher belegen.

Wenn der 2716 Sonder-ROM aus dem Genie durch Pin 1 des 1 aus 16 Decoders 74 LS 154 freigegeben wird, so ist gewährleistet, daß beim Einschalten des Gerätes (durch RESET wird in das Latch 0 eingetragen) immer dieser Baustein freigegeben ist und damit genauso verfügbar ist wie im Urzustand.

Eleganter Adreßdecoder für GENIE 1

Helmut Bernhardt

Als ich bei der Fehlersuche in meinem GENIE 1 durch die zahlreichen selbstgeschnitzten Erweiterungen und das damit verbundene Strippengewirr selbst nicht mehr durchblickte, keimte der Gedanke, das alles sehr viel übersichtlicher und völlig neu zu gestalten. Als Ergebnis kam ein kleines Board heraus, das in den Sockel des Z80 zu stecken ist und den Z80 selbst mit aufnimmt. Zusätzlich wird auf dem Board ein 27128-EPROM (das anstelle der dann überflüssigen Original-ROMs das Level-II-Basic enthält), ein 74LS273-Latch, ein PAL16L8 (das ein Freigabesignal für das 74LS273 und einige zusätzliche Freigabesignale erzeugt) und ein Adreßdecoder-PAL20L8 (das die Freigabesignale für alle memory mapped Systemkomponenten erzeugt) untergebracht.

Die Ausgänge des Latch 74LS273 haben folgende System-Steuerfunktionen:

Q0 = 0 : 0000H-2FFFH ist beim Lesen ROM und beim Schreiben RAM
= 1 : 0000H-2FFFH ist beim Lesen und Schreiben RAM

Q1 = 0 : 37E0H-37EFH ist Floppy, Drucker, INT-Status, Drive-Select
= 1 : 37E0H-37EFH ist RAM; Floppy u.s.w. sind über die Ports E0H- EFH erreichbar

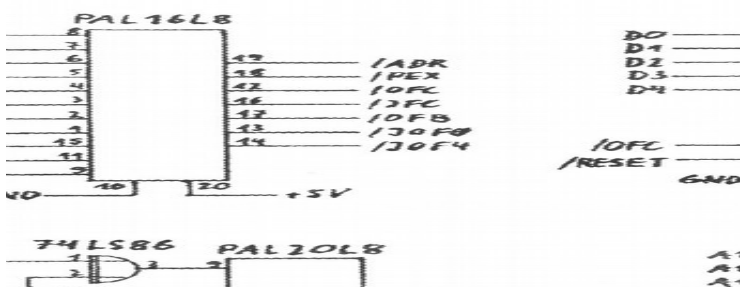
Q2 = 0 : 3800H-38FFH ist beim Lesen die Tastatur und beim Schreiben RAM
= 1 : 3800H-38FFH ist beim Lesen und Schreiben RAM

Q3 = 0 : 3C00H-3FFFH ist Video-RAM
= 1 : 3C00H-3FFFH ist RAM
Q4 = 0 : die memory mapped Systemkomponenten liegen im Bereich 0000H-3FFFH
= 1 : die memory mapped Systemkomponenten liegen im Bereich C000H-FFFFH

Unabhängig vom Zustand von Q0-Q4 ist dauerhaft bei 3000H-37DFH, 37F0H- 37FFH und 3900H-3BFFH RAM eingeblendet. Anstelle eines Sonder-ROM war mir RAM bei 3000H wichtiger.

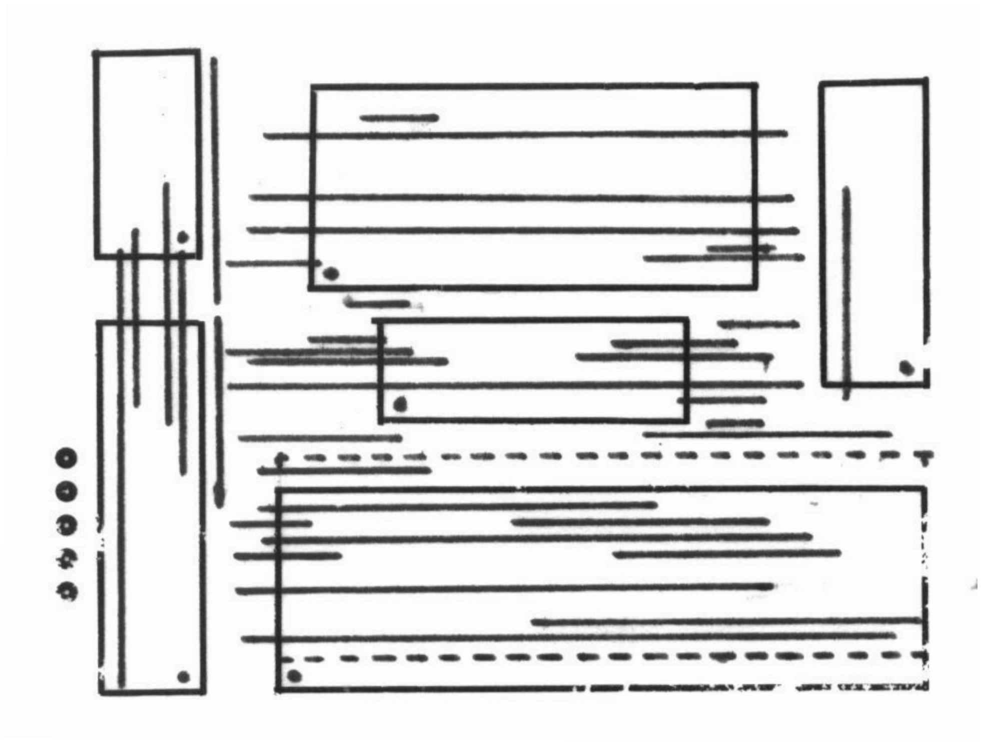
Das Board erzeugt die Freigabe-Signale /ROM, /RAM, /FLO, /KB und /VID und das Steuersignal /MRD für die Lesetreiber des RAMs. Diese Signale sind an die entsprechenden Punkte auf dem CPU-Board zu führen; die ursprünglichen Signale sind zu durchtrennen.

Schaltplan



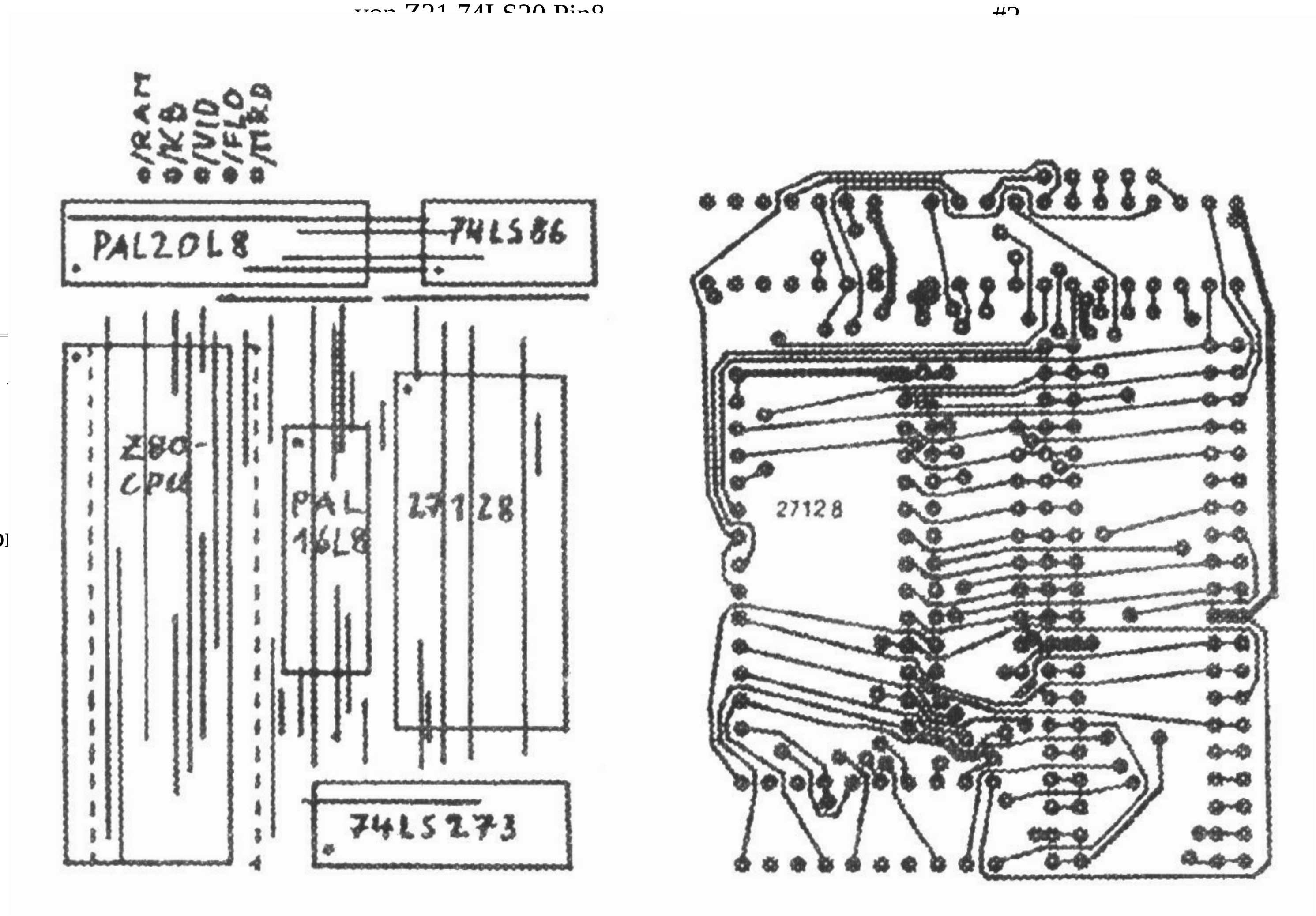
Layout

Drahtbrücken



Frei zum CPU-Board und zum RB-EXP1 zu verdrahtende Signale

Signal	Pin des 20L8	zu verbinden mit	zu durchtrennende Leitung	
/RAM	22	Z37,74LS367,Pin15	von Z36,74LS04,Pin2	#1
/MRD	17	Z9,74LS367,Pin1	von Z21,74LS20,Pin8	#2
/KB	20	Z7,74LS368,Pin1		
/VID	19	Interboard-Conn. Pin10		
/FLO	18	RB-EXP1, Z18, 74LS15,Pins2,14		



#1 bei Aufrüstung auf 64K (256K) RAM besteht diese Verbindung als freie

#2 bei Entfernen von Z21,74LS20 entfällt das Durchtrennen

#3 Z13,74LS30 auf dem EXP1 kann entfernt werden anstelle von /MWR vo
geführt werden, damit der Betrieb über die Ports E0H-EFH möglich ist

Das Freigabesignal /ROM ist auf dem Huckepack-Board direkt an die Pins 20 und 22 des EPROMs geführt.

Die Erzeugung dieser Freigabesignale ist aus dem Assembler-Listing für das PAL20L8 zu ersehen. Die Adreßeingänge A14 und A15 dieses PALs sind nicht die Adressen A14 und A15 der CPU. Diese Signale werden aus 2 XOR-Gattern (74LS86) aus den Adressen A14 und A15 der CPU erzeugt. Mit Q4 des Latch läßt sich damit steuern, ob A14 und A15 invertiert oder direkt an das PAL gelangen.

PAL-Assembler-Listing für das Adreßdecoder PAL20L8

PAL20L8 GENIE 1 Adreßdecoder mit mm-I/O-Switching
04.09.88 H. Bernhardt

A13 A11 MERQ RD A8 A9 A10 Q3 A15 Q1 PEX GND
ADR A14 MRD Q0 ROM FLO VID KB Q2 RAM A12 VCC

/ROM = /Q0 */MERQ */RD */A15 */A14 */A13
+ /Q0 */MERQ */RD */A15 */A14 * A13 */A12
/FLO = /Q1 */MERQ */A15 */A14 *A13 *A12 */A11 *A10 *A9 *A8 */ADR + Q1 */PEX
/KB = /Q2 */MERQ */RD */A15 */A14 *A13 *A12 *A11 */A10 */A9 */A8
/VID = /Q3 */MERQ */A15 */A14 *A13 *A12 *A11 *A10
/RAM = /MERQ *ROM *FLO *KB *VID
/MRD = /MERQ */RD *ROM *FLO *KB *VID

Um bei den zur Verfügung stehenden Pins des PAL20L8 auch das Signal /FLO voll ausdecodieren zu können (A4-A15), mußte eine Gruppe von Adressen (A4-A7) schon zusammengefaßt (/ADR) und an einen Pin des PALs geführt werden. Auch für das alternativ von einer Portadresse herzuleitende /FLO war ein vorheriges Ausdecodieren von /IORQ, A4-A7 nötig (/PEX). Diese Signale werden vom PAL16L8 erzeugt, an das für das Erzeugen von /OUTFC (Freigabe des Steuer-Latch) ohnehin schon die Adressen A0-A7 und /IORQ gelangen.

PAL-Assembler-Listing für das I/O-Decoder-PAL16L8
PAL16L8
GENIE 1 Portdecoder
04.09.88 H.Bernhardt
A7 A6 A5 A4 A3 A2 A1 A0 WR GND
RD OFC IOF0 IOF4 IORQ IFC OFB PEX ADR VCC

/OFC	= /IORQ */WR *A7 *A6 *A5 *A4 *A3 *A2 */A1 */A0	;Pin12
/IFC	= /IORQ */RD *A7 *A6 *A5 *A4 *A3 *A2 */A1 */A0	;Pin16
/OFB	= /IORQ */WR *A7 *A6 *A5 *A4 *A3 */A2 *A1 *A0	;Pin17
/IOF0	= /IORQ *A7 *A6 *A5 *A4 */A3 */A2	;Pin13
/IOF4	= /IORQ *A7 *A6 *A5 *A4 */A3 *A2	;Pin14
/PEX	= /IORQ *A7 *A6 *A5 */A4	;Pin18
/ADR	= A7 *A6 *A5 */A4	;Pin19

Da das PAL16L8 noch einige freie Ausgangs-Pins hatte, wurden noch einige weitere Port-Freigabe-Signale decodiert.

Funktion der Port-Freigabesignale des PAL16L8

/OFC (/OUT FCH)	wird intern zur Freigabe des Sytem-Steuerlatch benutzt
/IFC (/IN FCH)	steht für die Freigabe eines über Port FCH zu lesenden Treibers zur Verfügung
/OFB (/OUT FBH)	Freigabe-Signal, um über Port FBH ein Latch zu be schreiben; kann für die Freigabe des 74LS273 (Pin11) auf dem 256K-Banker benutzt werden (A0-A5 brauchen dann nicht mehr zum Banker geführt zu werden, A6 und A7 sind für das Erzeugen einer 8Bit-Refreshadresse dort aber weiterhin nötig; das 74LS30 auf dem Banker kann entfallen)

/IOF0 und /IOF4; (/In/OUT F0H-F3H; /IN/OUT F4H-F7H)	Freigabe-Signale für z.B. Z80-Peripherie-Bausteine mit 4-Portadressen (PIO, SIO, CTC, DMAC, PPI8255...)

Die Verwendung eines 27128-EPROMs (wovon es auch 150ns-Versionen gibt) erlaubt bedeutend höhere Taktraten als die ursprünglichen ROMs. Wenn dann auch noch das Timing der Signale /RAS, /MUX und /CAS anders gelöst wird, spielen auch die RAMs mit. 5,3MHz sind auch ohne große Eingriffe möglich.

Da das ROM nun innerhalb der Treiber der CPU-Signale liegt, muß zur Steuerung der Daten-Lesetreiber auch das Signal /ROM herangezogen werden. Diese Treiber dürfen nicht freigegeben werden, wenn das ROM gelesen wird.

Um dies zu erreichen, wird Pin9 von IC1,74LS14 durchgekniffen und hochgebogen. Auf IC1 wird ein 74LS08 mit den Pins 7 und 14 huckepack aufgelötet. Alle anderen Pins des 74LS08 werden hochgebogen. Der abgekniffenen und hochgebogene Pin9 des IC1 wird mit dem seitlich gebogenen Pin8 des 74LS08 verlötet. Pin10 des 74LS08 wird mit Pin22 des 27128-EPROMs und Pin9 des 74LS08 mit Pin3 von IC14,74LS00 verbunden (Abbildung).

Zur Freigabe der Speicherlese-Treiber erzeugt das PAL20L8 das Signal /MRD, das an Pin15 von IC9,74LS367 zu legen ist. Durch Entfernen von IC21,74LS20 wird das bisherige Signal entfernt.

Außer IC21 sind auch IC22,74LS156 und IC25,74LS139 überflüssig geworden. Diese ICs können ebenfalls entfernt werden. Und natürlich können auch die alten ROMs rausgeschmissen werden.

Beim Aufbau des Huckepack-Boards sind folgende Punkte zu beachten:

Vor dem Einlöten irgendwelcher Bauteile sind zunächst sämtliche Drahtbrücken zu legen. Als erstes sind dann der Sockel für den Z80 und der Stecker für das Anflanschen des Boards in den Z80-Sockel auf dem CPU-Board in geeigneter Reihenfolge einzulöten. Der Stecker wird auf der Lötseite und der Sockel auf der Bestückungsseite eingebaut. Dabei erschwert das zuerst eingebaute Teil den Einbau des anderen. Was zuerst angelötet wird, sollte eine gründliche Besichtigung der Teile ergeben.

Es sollten alle ICs gesockelt werden. Sowohl beim EPROM als auch bei den PALs kann irgendwann der Wunsch nach geändertem Inhalt aufkommen. Und für Testzwecke bei der Fehlersuche ist es auch bei Standard-ICs zweckmäßig, wenn man diese auswechseln kann oder einfach nur einzelne Pins aus der Fassung biegen kann.

Ungebohrte Boards, mit Level-II befüllte 27128-EPROMs und gebrannte PALs kann ich zum Selbstkostenpreis zur Verfügung stellen.

Auslesen des Graphikspeiches

GDP64-Karte des NDR-Klein-Computers im TRS80 und GENIE

Helmut Bernhardt, Jörg Seelmann-Eggebert

Zwischen der in den Computern TRS80 und GENIE wohl am weitesten verbreiteten HRG 1B und der GDP64 des NDR-Klein-Computers bestehen folgende gravierenden Unterschiede:

Feature	**HRG 1B**	**GDP64**
Bildformat	192x384	256x512
Mischbareit des Video-signals mit dem des Text-Bildschirms	ja	nein
Geschwindigkeit der Gra-phikausgabe	langsam	sehr schnell
Auslesbarkeit des Graphik-speichers	ja	nein
Anzahl Bildschirmseiten	1	4
Programmierbarkeit	umständlich	komfortabel
Anschluß an den Computer	beide frei verdrahtet	
BASIC-Treiber vorhanden	ja	ja
Verschleiß an Portadressen	128	32

Die wesentlichsten Nachteile der GDP64 bestehen in der nicht vorhandenen Auslesbareit des Grafik-Speichers, wodurch ein Abspeichern von Bildern auf Diskette ein Ausdrucken der Graphik ein Spiegeln und Invertieren des Bildes und die Implementation eines Graphik-Cursors nicht möglich sind.

Die nicht vorhandene Mischbarkeit des Videsignals mit dem des normalen Textbildschirms kann dadurch umgangen werden, daß entweder ein zweiter Monitor benutzt wird oder eine einfache softgesteuerte (und/oder handgesteuerte) Umschaltung des Videosignals für den Monitor dazugestrickt wird. Ein möglicher Schaltungsvorschlag wird zum Schluß noch vorgestellt.

Zunächst aber soll das Hauptproblem, das Auslesen des Graphikspeichers gelöst werden. Dazu sei vorweg bemerkt, daß wegen des kritischen Timings des EF9396-Graphikprozessors nicht garantiert werden kann, daß jeder Chip dabei funktioniert. Mit einem Pulldown-Widerstand kann der abfallenden Flanke des entsprechenden Signals zwar etwas Dampf gemacht werden, ob das bei dem jeweiligen Exemplar des 9366 dann ausreicht, muß probiert werden. Durch Verringern des Widerstandswertes läßt sich wahrscheinlich auch noch was rausholen.

Für das Auslesen des Graphik-Speichers der GDP64 benötigt man 2 zusätzliche ICs und einen 2k2-Widerstand. Hauptbetandteil ist ein 74LS374-Latch, in das der EF9366 den Inhalt der ihm benannten 8 Bits des Bildwiederholspeichers einträgt und aus dem sich die CPU dieses Datum abholen kann. Außerdem sind noch zwei OR-Gatter zur Ansteuerung des Latch nötig.

Und was muß gemacht werden?

Nicht viel. Bei dem zusätzlichen 74L374 werden außer den Pins 10 und 20 alle Pins um 45 Grad hochgebogen. Dann wird IC 20,74LS245 der GDP64-Karte aus aus der Fassung gezogen und das 74LS374 mit den Pins 10 und 20 auf den 74LS245 an dessen Pins 10 und 20 angelötet.

Dann wird IC 19, 74LS273 aus der Fassung genommen und auf dem Tisch so neben den eben gebauten Doppeldecker gelegt, wie die ICs 19 und 20 nebeneinander auf der Karte gesteckt

haben. Mit möglichst kurzen isolierten Drahtstücken werden dann alle in der folgenden Tabelle aufgeführten Verbindungen zwischen Huckepack-LS374 und 74LS273 hergesellt. Beim 74LS273 sollten die Drahtenden möglichst weit oben am Pin angelötet werden, damit die Pins nachher auch wieder in die Fassung passen. Danach kann das Verbundsystem wieder in die Fassungen gesteckt werden.

Nun werden die Verbindungen des Doppeldecker-74L374 mit den Pins des 74LS166 entsprechend der rechten Hälfte der Tabelle hergestellt, wobei dann der 74LS166 aus der Fassung zu ziehen und locker auf diese zu stellen ist (damit sich die Drahtlängen richtig bemessen lassen und andererseits beim Anlöten keine dauerhafte Verbindung zwischen IC und Fassung hergestellt wird; das entspräche dann auch nicht mehr dem Sinn einer Fassung).

Verbindungen zwischen dem Huckepack-LS374 und IC 19 sowie IC 4

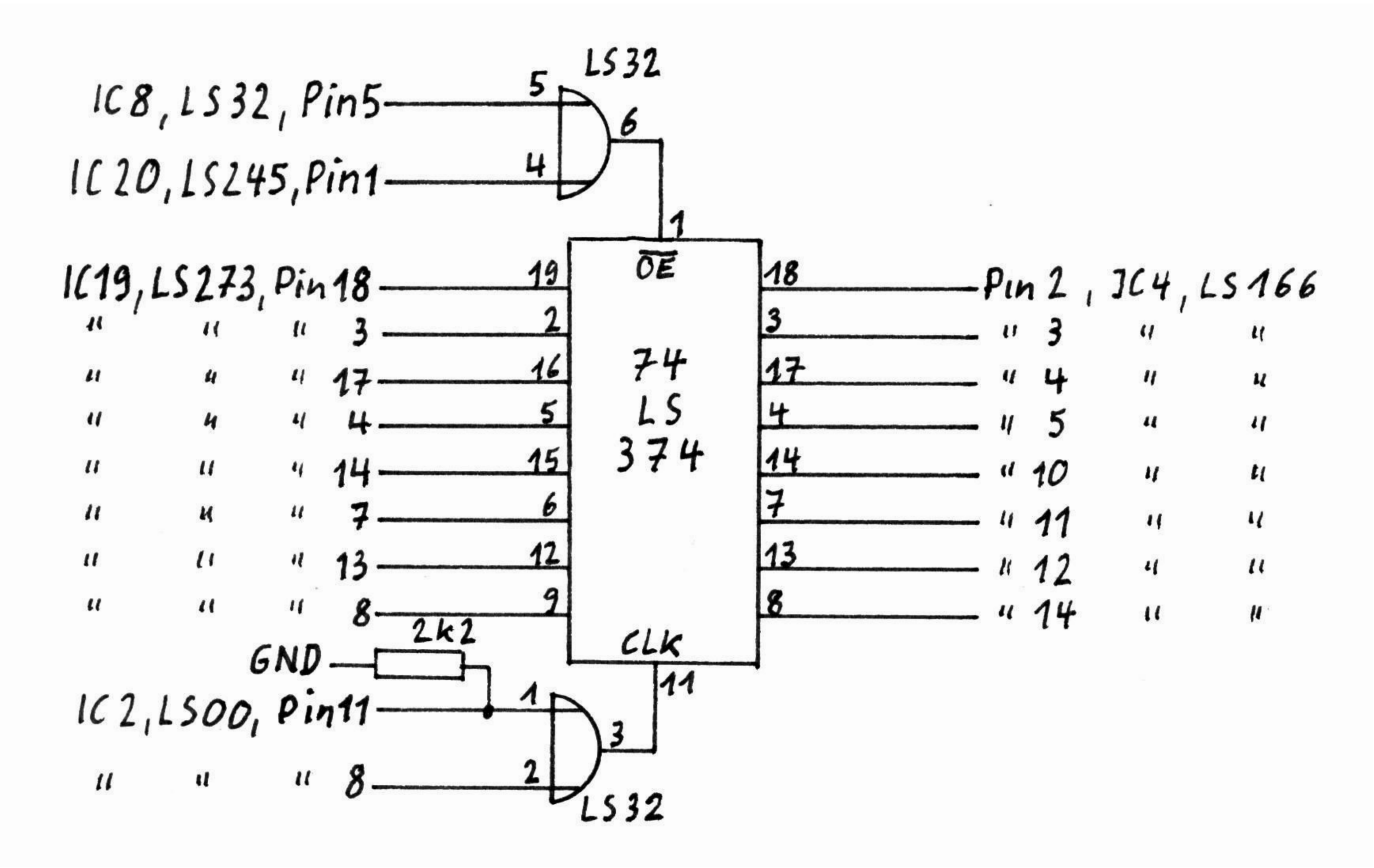
Signal	**IC19, LS273** Pin	**Huckepack LS374** Pin	Pin	**IC4, LS166** Pin
D1	3 ------------- 2	3 -------------- 3		
D3	4 ------------- 5	4 -------------- 5		
D5	7 ------------- 6	7 ------------ 11		
D7	8 ------------- 9	8 ------------ 14		
D6	13 ----------- 12	13 ------------ 12		
D4	14 ----------- 15	14 ------------ 10		
D2	17 ----------- 16	17 -------------- 4		
D0	18 ----------- 19	18 -------------- 2		

Nun wird ein zusätzliches 74LS32 genommen und dessen Pins 1-4, 6 und 8-13 ebenfal1 um 45 Grad hochgebogen. Mit den Pins 5, 7 und 1 wird es auf dem aus der Fassung genommenen IC8, 74LS32 an dessen Pins 5, 7 und 14 angelötet, Nun kann auch dieser Doppeldecker wieder in seine Fassung gesteckt werden. Eventuell müssen die abgespreitzten Pins der beiden Huckepack-ICs noch etwas zurechtgebogen werden, damit sie sich nicht gegenseitig berühren und auch keinen Kontakt mit ihren Träger-ICs haben. Dann werden folgende Verbindungen hergestellt:

Huckepack-LS32 Verbinden mit

Pin 1	IC2, 74LS00, Pin1
Pin 2	IC2, 74LS00, Pin8
Pin 3	Huckepack-LS374, Pin11
Pin 4	IC20, 74LS245, Pin1
Pin 6	Huckepack-LS374, Pin1

Und schließlich werden noch die Pins 1 und 7 des IC2 74LS00 über einen 2k2-Widerstand verbunden (Pulldown des MFREE-Signals des EF9366).



Damit ist alles getan, um die dem EF9366 vorgegebene Speicherstelle auslesen zu können.

Wenn danach das Auslesen des HRG-Speichers nicht funktioniert, ist das nicht verwunderlich, weil man zunächst etwas falsch gemacht hat; zumindest ich (He1mut) bin das so gewohnt. Dieser Fehler kann aber durch Vergleich des Strippengewirrs mit der Einbauanleitung schnell gefunden werden.

Möglicherweise treten dann beim Auslesen immer noch Fehler auf. Die Ursache liegt dann beim oben beschriebenen Timing des EF9366 mit seinem MFREE-Signal. Man kann dann noch mit einem kleineren Widerstand als 2k2 Versuche machen oder eventuell einen anderen EF9366 ausprobieren. Letztere Variante kann natürlich recht kostspielig werden, bis man ein funktionierndes Exemplar gefunden hat. A1ternativ kann man sich damit abfinden, daß es nicht funktioniert und alle Umbauten wieder abreißen (weniger erstrebenswert).

Und was bringt die GDP64 gegenüber der HRG 1B mehr?

An Software-Unterstützung gibt es dank der Bemühungen von Christof Ueberschaar ein 4K-Byte großes BASIC-Treiberprogramm, das ins HIMEM geladen wird und nach Einbinden in das Disk-BASIC sehr komfortable Graphik-Befehle bereitstellt. Gegenüber der Variante des Treibers für die HRG 1B wo die Graphik-Befehle durch ein # eingeleitet werden und daher der BASIC-Interpreter bei jedem Befehl erst prüfen muß, ob ein # vorliegt, ist hier der CMD-Vektor verbogen worden. Es wird dann nur noch bei Auftauchen des CMD-Tokens im Programmtext eine zusätzliche Abfrage gemacht, ob das nach CMD obligatatorische (") folgt und nur dann, wenn das nicht der Fall ist, weiter untersucht, ob das Byte ein gültiger Graphik-Befehl ist, der dann abgearbeitet wird.

Diese Philosophie hat zwar den Nachteil, daß man anstelle des einfachen # jedesmal 3 Zeichen eintippen muß, bevor der eigentliche Graphik-Befehl kommt, andererseits wird dadurch nicht die Abarbeitung der normalen BASIC- Befehle verlangsamt womit ein Teil des Geschwindigeitsvorteils der GDP64 verschenkt wäre.

Außerdem ist die Punktadressierung der GDP64 bedeutend einfacher als die der HRG 1B. Es werden dem EF9366 einfach die X- und Y-Koordinaten des Punktes als 16-Bit-Wörter in entsprechende Register geschrieben und dann einer der Befehle "Punkt setzen", "Punkt löschen" oder "Punkt abfragen" ausgeführt. Bei der HRG 1B bedeutet das ein umständliches Berechnen

der Speicheradresse und der entsprechenden Bits im Byte.

Noch gravierender ist die Eleganz beim Zeichnen von Linien über Vektorbefehle. Hier werden die Punktkoordinaten (X1/Y1) für den Ausgangspunkt einer Linie in die entsprechenden Register eingetragen und in zwei weitere Register werden die Werte Delta X und Delta Y eigetragen. Es sind dies die Differenzen X2-X1 und Y2-Y1 (Koordinaten von Anfangs- und Endpunkt einer Linie). Diese Differenzen müssen stets positive Zahlen sein. Das Vorzeichen (Vektorrichtung) wird durch entsprechende Bits in den Vektorbefehlen vorgegeben. Es läßt sich auch noch festlegen, ob die Linie durchgezogen, gepunktet oder punktstrichliert ausgegeben werden soll oder ob die Linie überhaupt gezeichnet werden soll oder eine vorher hier gezeichnete Linie wieder gelöscht werden soll.

Wenn die Line dann mit einer Geschwindigkeit von 1,5 Mio Punkten pro Sekunde gezeichnet ist, stehen automatisch in den X- und Y-Koordinatenregistern die Koordinaten X2 und Y2 der eben gezeichneten Linie, so daß von deren Endpunkt durch Vorgabe neuer Werte für Delta X und Delta Y gleich ein weiterer Vektor gezeichnet werden kann, ohne daß die Endpunktskoordinaten der vorherigen Linie noch als Anfangskoordinaten der neuen Linie ausgegeben werden müssen.

Dann hat der EF9366 noch einen eingebauten Zeichengenerator (der aber leider nur die amerikanischen Zeichen kennt:), mit dem ein komfortables Einbinden von Texten in die Graphik möglich ist. Außer Aufrecht- und Schrägschrift sowie in Breite und Höhe voneinander unabhängig in 16 Stufen in der Größe variierbaren Zeichen läßt sich vorgeben, ob normal von links nach rechts, auf dem Kopf stehend von rechts nach links oder um 90 Grad gedreht von unten nach oben oder von oben nach unten geschrieben werden soll. Für die Textausgabe ist vor der Ausgabe der Textzeichen neben dem Festlegen dieser Modi auch noch der Fußpunkt des ersten Zeichens in die Koordinatenregister einzutragen. Nach der Ausgabe eines Zeichens erhöht sich der Pointer auf den Fußpunkt des dahinter auszugebenden Zeichens, wobei die oben beschriebenen Modi natürlich berücksichtigt werden.

Diese Bemerkungen sollen keinesfalls als erschöpfende Beschreibung der Leistung der GDP64 zu werten sein. Für die Programmierung des EF9366 kommt man ohne Datenblatt nicht aus. Es sollte hier nur ein ungefährer Eindruck vermittelt werden, wofür der Aufwand überhaupt getrieben wurde, bei der unsicheren Erolgsaussicht, den Hardware-Patch überhaupt anzufangen.

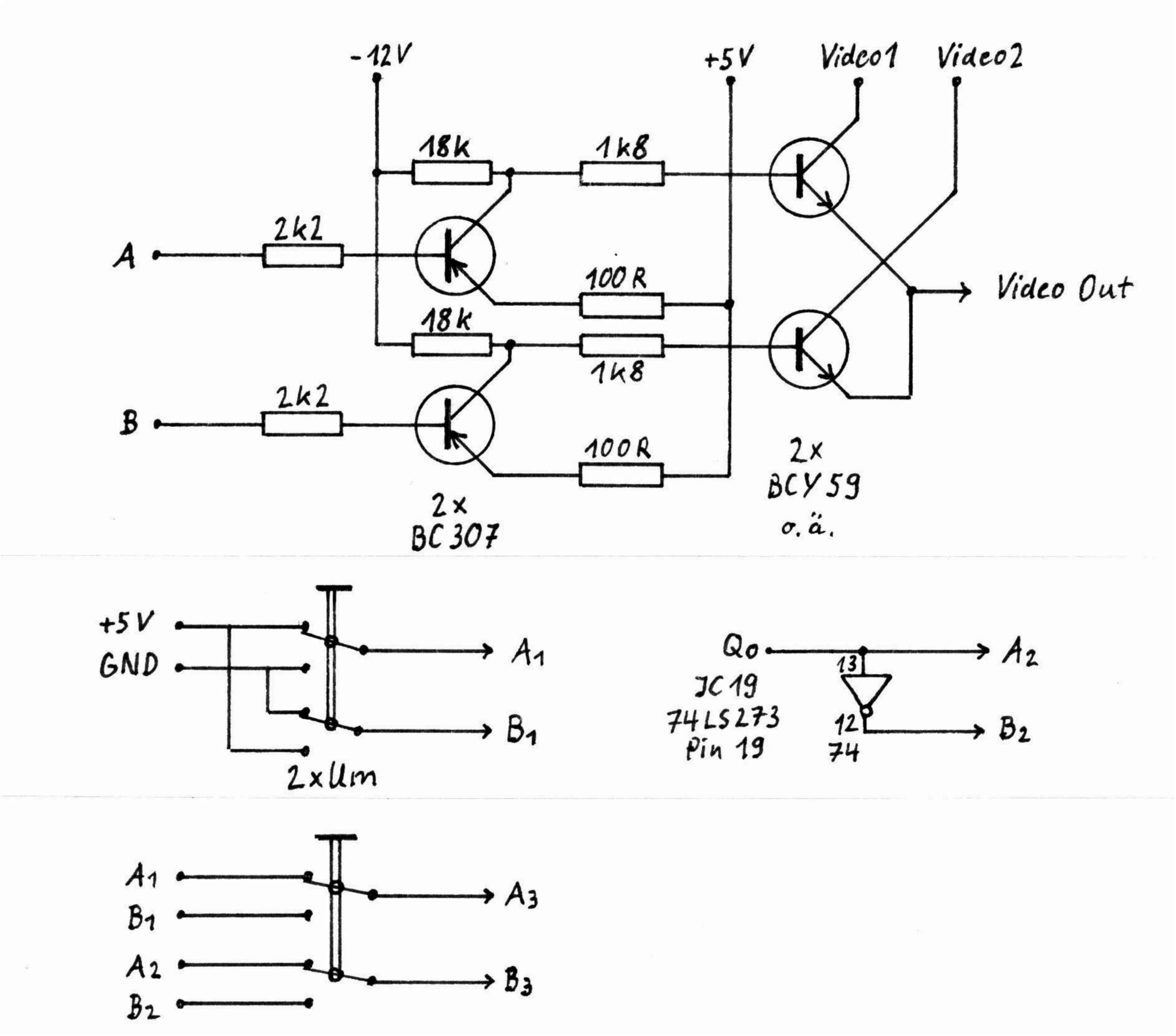
Das Umschalten der Video-Signale

Da die Video-Signale des Text/Bauklötzchen-Graphik-Interface des Computers und der GDP64-Karte nicht direkt miteinander mischbar sind, muß mit einer Umschaltung zwischen den Signalen gearbeitet werden. Für Experimentierzecke reicht dazu ein einfacher Umschalter, mit dem abwechselnd eines der beiden Video-Signale durchgeschaltet werden kann. Die Bezugsmassen der beiden Signale sind miteinander und mit der Abschirmung des Kabels zum Monitor zu verbinden.

Als Dauerlösung ist aber eine Softumschaltung wesent1ich eleganter. Dafür kann das Bit 0 des unter Port 60H decodierten Latch (IC19, LS273) mit dem auch die HRG-Speicherbanks angewählt werden, mißbraucht werden. Es ist dann noch ein Inverter nötig, den IC9, 7404, Pins 13 und 12 liefert. Dabei ist zu beachten, daß mit einem OUT-Befehl an Port 60H dann sowohl die für Auslesen und Schreiben einzustellenden Banks als auch die Seletion des Video-Signals gesteuert werden. Da dieses Latch sich nicht auslesen läßt, muß über dessen Inhalt im RAM Buch geführt werden, damit beim Ändern der Bankanwahl nicht das Video-Signal umgeschaltet wird und umgekehrt. Außerdem sollte der Pin 1 des IC 19, 74LS273 nicht mit +5V sondern mit RESET* oder einem aus RESET* und NMI* erzeugten SYSRES* verbunden werden, damit durch Löschen des Latch beim Einschalten bzw. Drücken des RESET-Knopfes, wodurch ein NMI* erzeugt wird) ein definierter Zustand vorliegt.

In Abb. 2 ist oben eine prinzipielle Schaltung zur Umscha1tung der Video-Sinale (Video1 und Video2) gezeigt, deren Steuersignale A und B entgegengesetzte logische Pegel haben müssen. Wenn anstelle von -12V nur -5V zur Verfügung stehen, kann auch -5V verwendet werden. Wenn anstelle der beiden 18k-Widerstände solche von 7k5 genommen werden.

Unten in Abb.2 sind zwei Möglichkeiten für die Beschaltung der Steuereingänge A und B gezeigt, womit sich entweder durch Hand- oder durch Softumschaltung das gewünschte Video-Signal selektieren läßt. Und schließlich ist noch die Umschaltung der Umschaltungsvarianten per Hand gezeigt, wofür dann noch ein 2fach-Umschalter nöig ist.



Gepufferter DMA- und IM2-fähiger ECB-Bus für GENIE I und II

Helmut Bernhardt

Während bei Verzicht auf DMA- und IM2-Fähigkeit die Steuerung des Datenbuffers zwischen Computer und externem ECB-Bus relativ einfach zu realisieren ist (der Buffer darf nur in Richtung CPU treiben, wenn Daten von Baugruppen gelesen werden, die nicht auf der CPU-Seite des Buffers - also auf dem ECB-Bus - sitzen), sind bei der Einbindung dieser Features einige zusätzliche Aspekte zu berücksichtigen.

Der Interrupt-Mode2 der Z80-CPU setzt voraus, daß der Interrupt-Lieferant beim Interrupt-Acknowledge (M1* und IORQ* beide low) ein 8Bit-Wort auf den Datenbus legt, das die CPU als LSB einer Adresse in der Interrupt-Vektor-Tabelle interpretiert. Dieses Byte muß von einer Interrupt-Quelle vom ECB-Bus durch den Datenbuffer auch zur CPU gelangen. Der Buffer muß beim Interrupt-Acknowledge also auch in Richtung CPU treiben.

Bei übernahme der Daten-, Adreß- und Steuerbusse durch einen DMA-Controller (DMAC) auf dem ECB-Bus muß dieser auch Zugriff auf RAM, EPROM und alle CPU-seitig des Buffers liegenden Baugruppen haben. Für Adreß- und Steuerleitungen gestaltet sich diese Richtungssteuerung recht einfach. Hier genügt es, wenn durch das low aktive CPU-Signal BUSAK* (nur dann darf der DMAC arbeiten) die Buffer dieser Signale in Richtung zur CPU treiben, so daß der DMAC dort entsprechend adressieren kann. Für die Steuerung des Datenbuffers sind noch weitere Umstände zu berücksichtigen.

Wenn die CPU den Bus kontrolliert, kann beim Schreiben der Buffer immer in Richtung ECB-Bus treiben. Beim Lesen darf er nur dann nicht in Richtung CPU treiben, wenn interne Baugruppen (CPU-seitig des Buffers) adressiert werden. Sonst würde ein von außen gelesenes FFH gegen ein von der intern adressierten Baugruppe ausgegebenes Datenwort arbeiten. Abgesehen davon, daß die gegeneinander arbeitenden Komponenten sich irgendwann gegenseitig zerstören, würde die CPU Fehlinformationen lesen.

Wenn der DMAC <45> den Bus steuert, ist Lesen und Schreiben für die Treiberrichtung des Buffers das ganze Gegenteil. Wenn der DMAC Daten einer CPU-seitigen Systemkomponente liest, wobei er dann auch die Leitung RD* low zieht, muß der Buffer in Richtung ECB-Bus treiben. Das darf er aber nur dann, wenn er auf CPU-seitige Komponenten zugreift. Wenn er von einer Baugruppe auf dem ECB-Bus liest, muß der Buffer in Richtung CPU treiben, um dann nicht auf dem ECB-Bus zwei Datenwörter gegeneinander arbeiten zu lassen.

Für die Treiberrichtung des Buffers muß also berücksichtigt werden, ob der DMAC oder die CPU den Bus kontrolliert, ob die adressierte Baugruppe auf dem ECB-Bus oder CPU-seitig des Buffers liegt und ob gelesen oder geschrieben wird.

In Tabelle 1 sind die für verschiedene Betriebszustände aus obigen überlegungen abgeleiteten Treiberrichungen (von der CPU aus gesehen) zusammengestellt. Dabei ist zu berücksichtigen, daß zusätzlich auch noch beim Interrupt-Acknowledge die Richtung "rein" sein muß, weil der GENIE keine eigenen vektorisierten INT-Quellen hat. Diese Tabelle ist Grundlage der Schaltung in Abb.1. Die Richtungssteuerung soll ausschließlich aus Signalen, die von CPU-seitigen Baugruppen zu beziehen sind, erfolgen. Nur dann ist auch gewährleistet, daß auf dem ECB-Bus jede handelsübliche Karte funktioniert. Kommerzielle ECB-Bus-Karten legen kein aktives Signal auf den Bus, das in die Buffersteuerung einbezogen werden kann.

Tabelle 1: Treiberrichtung des ECB-Bus Databuffers

Bus-Steuerung durch	interne Baugruppe adressiert		externe Baugruppe adressiert	
	Read	Write	Read	Write
CPU	raus	raus	rein	raus
DMAC	raus	rein	rein	rein

Der Aufwand dafür ist aber nicht unerheblich und der Anwender dieser Bussteuerung muß sein System genau kennen, um die nötigen Steuersignale an die Schaltung legen zu können. Einen Anhaltspunkt dafür, welche Freigabesignale vorkommen können, mag die Tabelle 2 geben. Darin sind die prinzipiell vorhandenen Baugruppen und einige häufig benutzte Hardware-Erweiterungen und die Punkte, wo deren Freigabesignale abzugreifen sind, zusammengestellt. Diese Punkte sind mit jeweils einem der Punkte A bis H auf der Eingangskarte zu verbinden.

Tabelle 2: Interne Freigabesignale

CPU-Board	Adresse		IC	Typ	Pin
RAM	4000-FFFF		37	367	15
	(3900-3BFF)	<7>			
ROM1	0000-0FFF		10	3001	20
ROM2	1000-1FFF		11	3002	20
ROM3	2000-2FFF		12	3003	20
ROM4	3000-36FF(37DF)	<7>	13	2716	20
Tastatur	3800-3BFF(38FF)	<7>	7	368	1
Video-RAM	3C00-3FFF		35	32	3
Video-Board					
Cassette 1	Port FF		20	32	11
Cassette 2	Port FE	bleibt unberücksichtigt #1			
Expansion-Interface	EG3014				
RAM1	8000-BFFF		37	20	10
RAM2	C000-FFFF		37	20	9
Floppy	37E0-37FF		29	139	12
Drucker	37E8 u.Port FD		31	156	9
RS 232 EG3020	Ports F8, F9		37	20	10
EXP1 von RB	37E0-37FF		18	155	2,14
	Port FD		14	32	3
RB.V24	Ports 80-88		Z80-CTC		16
			Z80-DART		35
HRG1B von RB	Ports 00-7F		?	155	2,14
Schmidtke 80Z.Karte	B000-BFFF		?	245	19
	Ports D0, D1		?	138	15
EG64MBA	Port DF	bleibt unberücksichtigt #1			
EG64MBA+ <6>	Ports DE, DF	"		"	"
256K-Banker <10>	Port EC	"		"	"
(u. ED mit	dieser Karte)	"		"	"
CP/M-Banker <6>	Ports 50-5F	"		"	"

#1 diese Baugruppen werden nicht gelesen und brauchen deshalb nicht in die Buffersteuerung miteinbezogen zu werden, das IN A,(0DFH) des EG64MBA dient nicht der Datenaquisition, es setzt nur das 74LS259-Latch zurück, ein Datenwort wird nicht ausgegeben

Im Normalfall liegen alle memory-mapped (mm) Baugruppen CPU-seitig des Treibers, so daß dann einzig MERQ* (IC16, 74LS367, Pin5) an die Schaltung geführt werden muß. Wenn aber auch nur eine mm Baugruppe auf dem ECB-Bus steckt, müssen alle anderen mm Freigabesignale einzeln an die Schaltung geführt werden.

Durch Vorschalten weiterer AND-Gatter vor die Eingänge des 74LS30-ICs (z.B. 74LS08, 74LS11, 74LS21) kann die Anzahl der verwendbaren Eingänge für Freigabesignale beliebig erhöht werden (Abb.10).

Die DMA-Fähigkeit der ECB-Busbuffer alleine reicht nicht aus. Es muß durch Verbinden der Systembus-Leitung BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS* auch dafür gesorgt werden, daß beim DMA-Zugriff die Buffer der CPU-Signale in den hochohmigen Zustand versetzt werden. Das kann entweder auf der Eingangskarte durch Querverdrahtung der entsprechenden Pins von CN2 bzw. CN3 oder durch feste Verdrahtung auf dem CPU-Board gemäß Abb.11 erreicht werden.

Da für die Richtungssteuerung des Datenbuffers ohnehin einige ICs nötig waren, und der ECB-Bus auch IM2-fähig ist, ließ sich mit etwas Mehraufwand auch gleich eine Vektorisierung der Interrupts des 25ms-Timers und des Floppy-Controllers realisieren. Dafür ist Voraussetzung, daß auf dem ECB-Bus auf einer Karte ein freier Z80-CTC-Kanal (2) vorhanden ist, über den durch Software das gemeinsame INT*-Signal des Systems umgeleitet werden kann. Der CTC-Kanal ist dann im Zähler-Modus mit der Zählkonstanten 1 und Triggern durch eine negative Flanke zu programmieren.

Die direkte Verbindung des INT*-Signals (im Schaltplan INTEXP*) von der Floppy-Baugruppe zur CPU ist dafür aufzutrennen. Das Signal INTEXP* ist stattdessen an den entsprechend bezeichneten Punkt der Schaltung und an den freien Triggereingang des CTC-Kanals zu legen. Die INT*-Leitung des ECB-Bus ist an den Punkt INTECB* zu legen und der Ausgang INTCPU* ist mit dem INT-Eingang der CPU (Z80, Pin16) zu verbinden.

Damit ist gewährleistet, daß nach dem Einschalten oder Drücken des RESET-Knopfes immer der Interrupt vom FDC-Board direkt zur CPU durchgeschaltet wird und das Betriebssystem im IM1 arbeiten kann. Wenn allerdings andere Baugruppen auf dem ECB-Bus in Echtzeit im IM2 betrieben werden sollen, müssen auch die systemeigenen Interrupts dieses Spiel mitmachen können und das geht nur dann, wenn ihnen der CTC-Kanal beim INT-Acknowledge einen Vektor zur Verfügung stellt. Die Schaltung leistet es, daß ein LD A,xxxx xx1xB; OUT (0EDH),A das INT* Signal des FDC-Boards über den CTC umleitet.

Die Software muß dann aber Rücksicht auf diesen Umstand nehmen. In der INT-Vektor-Tabelle darf nicht einfach 4012H stehen (INT-Vektor des DOS), da die Serviceroutine des DOS nicht mit RETI sondern mit RET endet und der Z80-CTC dann nicht weiß, wann er seinen IEO-Ausgang wieder auf High zurücknehmen soll. Es muß folgender Umweg eingeschlagen werden:

```
INTAB  DEFW   SERV1   ;Anfang INT-Tabelle
       DEFW   SERV2   ;nächster Vektor
       DEFW   .....   ;u.s.w.
       ....   .....
       DEFW   CTCN    ;Vektor für Serviceroutine des INT vom CTC-Kanal
       ....           ;N, über den die System-Interrupts laufen
       DEFW   LETZT   ;Ende der Vektor-Tabelle
       ....   .....
CTCN   CALL   4012H   ;Serviceroutine des DOS
       EI
       RETI
```

Außerdem bleibt es dem Anwender offen, während des IM2-Betriebes anderer Baugruppen den 25ms INT verhungern zu lassen. Wenn beim Lable CTCN einfach ein EI, RETI steht, wird 37E0H nicht mehr gelesen und daraufhin erfolgt kein weiterer Interrupt des Timers mehr. Eine Reinitialisierung des Timer-INTs geschieht durch Umprogrammieren des Vektors auf ein anderes Lable, das wieder den CALL 4012H enthält, und einmaliges Lesen von 37E0H.

Durch Zurücksetzen von D1 in Port EDH wird wieder der INT des FDC-Boards direkt an die CPU durchgeschaltet und alle Interrupts des ECB-Bus werden unterbunden.

Mit der für die Interrupt-Steuerung benötigten Portdecodierung und mit dem Latch wurden noch einige zusätzliche Funktionen realisiert. Zunächst wurden auch die Portfreigabesignale OUTECH*, OUTEEH* und OUTEFH* hergeleitet. Davon kann OUTECH* zur Freigabe des 74LS273 Latch auf dem 256K-Banker <10> benutzt werden. Dieses Signal wird direkt an den Pin11 des 74LS273 geführt, wobei dann aber die bisher dahin führende Leitung zu durchtrennen ist. Damit ist dann das 74LS30 IC auf dem Banker überflüssig und auch die Leitungen A0 bis A5 sowie IORQ* und WR* brauchen nicht mehr dorthin geführt zu werden (A6 und A7 werden dort nicht nur zur Portdecodierung sondern auch noch zur Erzeugung der 8Bit-Refreshadresse benötigt und müssen deshalb weiterhin angeschlossen bleiben).

Mit Bit 0 in Port EDH wurde ein softgesteuertes Invertieren des zum 256K-Banker führenden Signals A15 realisiert. Damit läßt sich mit D0=0 an Port EDH (Einschaltzustand) vorgeben, daß die unteren 32K Adreßbereich des Z80 gebankt werden können, während D0=1 an Port EDH das Banking der oberen 32K festlegt.

Und schließlich kann mit D2=1 an Port EDH ein gemeinsames Invertieren der Adressen A14 und A15 erreicht werden. Die dabei gewonnenen Signale A14' und A15' können anstelle der Signale A14 und A15 an die Decoderschaltung für die 16K (ROMs und mm I/O), IC25, 74LS139, Pins 14 und 13 geführt werden und gestatten dadurch ein Verlegen dieser Komponenten an das obere Speicherende 48-64K. In diesem Bereich funktionieren dann zwar die ROMs nicht mehr, wenn man aber dann noch ein 74LS32 IC spendiert und mit dem Signal ED,3 (D3 von Port EDH) ein programmierbares Abschalten der ROMs einbaut (Abb.10), hat man auch noch die Funktionen des Omikron Mappers (kompatibler Selbstbau in <9>) für den Betrieb von CP/M realisiert. Dieser leistet (wenn auch mit einem anderen OUT-Befehl) ebenfalls das Einstellen der Systemkonfiguration:

0000H-F7DFH	RAM <7> #1
F7E0H-F7FFH	Floppy, Drucker
F800H-FBFFH	Tastatur <7>
FC00H-FFFFH	Video-RAM

#1 der Omikron Mapper für den TRS 80 leistet selbst eine vollständige Decodierung des Bereichs des Sonder-ROMs des GENIE

Ein Patchen der CP/M Systemdiskette ist sehr einfach. Auf den Systemspuren läßt sich relativ einfach die Sprungleiste des Loader Bios finden, die aus einer Reihe von Jumps besteht (C9 nn nn C9 nn ...). Daran schließen sich die Disk Parameter Tabellen mit sehr vielen 00H an und unmittelbar danach stehen die Befehle LD A,40H; OUT (50H),A. Diese beiden Befehle sind durch die Befehle LD A,xxxx 11xxB; OUT (0EDH),A zu ersetzen. Damit läuft das CP/M des Omikron Mappers dann auch ohne denselben.

Wenn diese Eingriffe auf dem CPU-Board gemacht werden, ist der Computer aber von der ECB-Bus-Eingangskarte abhängig, weil von dieser dann lebenswichtige

Signale geliefert werden. Der ECB-Bus braucht selbst nicht angeschlossen zu sein.

Von dieser Abhängigkeit kann man sich durch einen Mehrfach- (oder mehrere Wenigfach-) Umschalter lösen. Abb.9 zeigt, wie man damit entweder die ursprünglichen Signale des CPU-Boards oder die steuerbaren Signale dieser Schaltung an die entsprechenden Stellen des CPU-Boards führt, und damit das Gerät entweder wie bislang ohne ECB-Eingangskarte oder mit derselben und allen ihren Mehrleistungen laufen lassen kann.

Um den Umfang der (zusätzlich zu denen des Systembus) an die Eingangskarte geführten Leitungen etwas zu reduzieren, bietet es sich an, IC7, 74LS30 auf der Eingangskarte nicht zu bestücken und dieses IC im Computer an geeigneter Stelle huckepack mit den Pins 7 und 14 auf ein anderes Dil14 IC zu löten und die internen Freigabesignale fest dorthin zu verlegen (Pins 1-6, 11-12). Dann braucht nur noch das Signal INTERN von Pin8 des 74LS30 an die Eingangskarte (CN2, Pin25c) geführt zu werden.

Wenn die zusätzlichen Features der Eingangskarte voll genutzt werden sollen, können alle benötigten Signale (SYSRES* A15* A14' A15' ED,3 OUTECH*) über freie Verdrahtung an unbelegte Pins von CN2 gelötet werden. Im Layout sind nur die Sinale INTCPU* INTEXP* INTECB* und INTERN über CN2 geführt. Dadurch können alle Verbindungen mit dem Computer (nicht nur der Systembus) über CN2 laufen.

Wer diese Variante des ECB-Bus bauen möchte, kann bei mir für 10 DM (incl. Rückporto) eine geätzte aber ungebohrte Karte im Europaformat bekommen. Sollte irgendjemand die Karte am Komtek 1 betreiben wollen, kann ich entsprechende Informationen zur Verfügung stellen, die eine Nutzung auch dort ermöglichen. Der Anschluß an den TRS 80 ist nicht ohne Sondermaßnahmen möglich, weil die internen Datentreiber IM2-Betrieb verhindern. Ein Hardware-Patch, der dagegen hilft, ist in (8) beschrieben. Eine Herleitung der Signale IORQ* MERQ* RD* und WR* aus den daraus gewonnenen Signalen IN* OUT* MWR* und MRD* des TRS 80 Systembus mit Gattern erzeugt Signale mit nicht ganz richtigem Timing, die eventuell Schwierigkeiten bereiten können; hier ist das Abgreifen der originalenSignale eine bessere Lösung.

Aufbau und Anschließen der Karte

Entgegen der üblichen Technik, solche Karten doppelseitig geätzt zu entwerfen, wurde hier ein einseitiges Layout mit einer unverschämten Menge Drahtbrücken bevorzugt, was das Durchkontaktieren an IC-Sockel-Pins vermeidet. Damit wird beim Aufbau aber viel Fummelei und beim Testen viel Ärger erspart.

Der Stecker CN1 (VG64, a,c, Stifte, gewinkelt) wird auf den ECB-Bus gesteckt. Für den Anschluß an den Systembus wurden 2 Varianten vorgesehen. Durch Einbau einer VG64 a,c-Buchsenleiste bei CN2 von der Lötseite (Abb.8) muß durch einen entsprechenden Drahtverhau zum Gegenstück für den 50poligen Platinenrandstecker des CPU-Boards die Anordnung der Signale umsortiert werden. Wer allerdings ein ästhetisches 50poliges Flachkabel bevorzugt, kann CN2 unbestückt lassen und stattdessen in die nicht angeschlossenen Lötaugen von CN3 von der Lötseite einen geeigneten 50poligen Stecker einlöten. Dann wird aber der zum Umsortieren der Signale nötige Drahtverhau nur auf die Karte zwischen CN2 und CN3 verlagert.

Außer den reinen Bus-Signalen des Computers müssen bei Bedarf auch noch die aktiven Steuersignale der Eingangskarte an den Computer zurückgeführt werden und die im Computer abzugreifenden Feigabesignale an die Eingangskarte geführt werden. Dafür sind am oberen Platinenrand entsprechende Lötpunkte vor-

handen. Die wichtigeren Signale sind aber auch an Lötpunkte von CN2 geführt, womit dann alle Signale über CN2 (und bei entsprechender Querverdrahtung auch CN3) mit dem Computer verbunden werden können. Dafür reicht dann aber der 50polige Platinenrandstecker des CPU-Boards mit seinen 3 freien Pins nicht mehr aus. Da muß dann noch eine zusätzliche fliegende Steckverbindung herhalten, die diese zusätzlichen Signale übergibt.

Seckerbelegungen

CN1 (ECB-Bus)

Reihe a	NR.	Reihe c
GND	32	GND
BUSAK*	31	RESET*
A9	30	MERQ*
A13	29	CLOCK
RFSH*	28	A15
IORQ*	27	A12
#1	26	#1
#1	25	HALT*
#1	24	RD*
#1	23	#1
-12V #2	22	WR*
#1	21	INT*
M1*	20	NMI
#1	19	#1
A14	18	A10
#1	17	A11
#1	16	IEO
-5V	15	#1
#1	14	D1
+12V	13	#1
#1	12	A14
BUSRQ*	11	IEI
Wait*	10	#1
A6	9	A7
A5	8	A8
A4	7	A1
A2	6	A3
D4	5	A0
D3	4	D2
D6	3	D7
D5	2	D0
+5V	1	+5V

CN2 (vom Systembus)

Reihe a	Nr.	Reihe c
GND	32	GND
BUSAK*	31	RESET*
CLOCK	30	nc
MERQ*	29	RFSH*
IORQ*	28	RD*
M1*	27	WR*
HALT*	26	nc
nc	25	INTERN
nc	24	INTEXP*
NMI*	23	INTECB*
A11	22	A8
A10	21	A14
A13	20	A9
A12	19	A15
nc	18	nc
nc	17	nc
nc	16	nc
nc	15	nc
A7	14	A6
A5	13	A4
A3	12	A2
A1	11	A0
BUSRQ*	10	nc
nc	9	WAIT*
nc	8	nc
D5	7	D0
D6	6	D7
D3	5	D2
D4	4	D1
nc	3	INTCPU*
nc	2	nc
(+5V)	1	(+5V)

Stückliste

ICs:
4 x 74LS245
1 x 74LS174
1 x 74LS139
1 x 74LS125
1 x 74LS86
1 x 74LS32
1 x 74LS30
1 x 74LS11
1 x 74LS08
1 x 74LS00

Widerstände:
2 x 10k

Sonstiges:
1 x VG64 a,c Stifte gewinket
1 x VG64 a,c Buchsen gerade
1 x Leiterplatte
jede Menge Draht
viel Geduld

#1 diese Pins sind zum Teil nicht belegt oder führen Signale, die hier nicht benutzt werden
#2 die Lage von -12V wird nicht einheitlich gehandhabt, diese Spannung kann auch woanders liegen (muß bei jeder gekauften Karte geprüft werden)

Literatur

<1> Verbindend - Der ECB-Bus / Johannes Assenbaum / c't 12/85, 60-61

<2> Z80 CTC - Einsatz und Programmierung / Rolf Keller / c't 4/85, 92-94

<3> Interrupt-Priority-Encoder und Vektorgenerator für IM2 mit dem Z80 CTC / H. Bernhardt / Info des GENIE/TRS 80- User-Club Bremerhaven 12/85, 10-11

<4> Direkter Speicherzugriff; Schneller als die CPU - So arbeiten DMA-Controller-ICs / Rolf Keller / c't 8/85, 80-83

<5> Z80 DMA - Schnell, aber Kompliziert / Rolf Keller / c't 12/85, 76-79

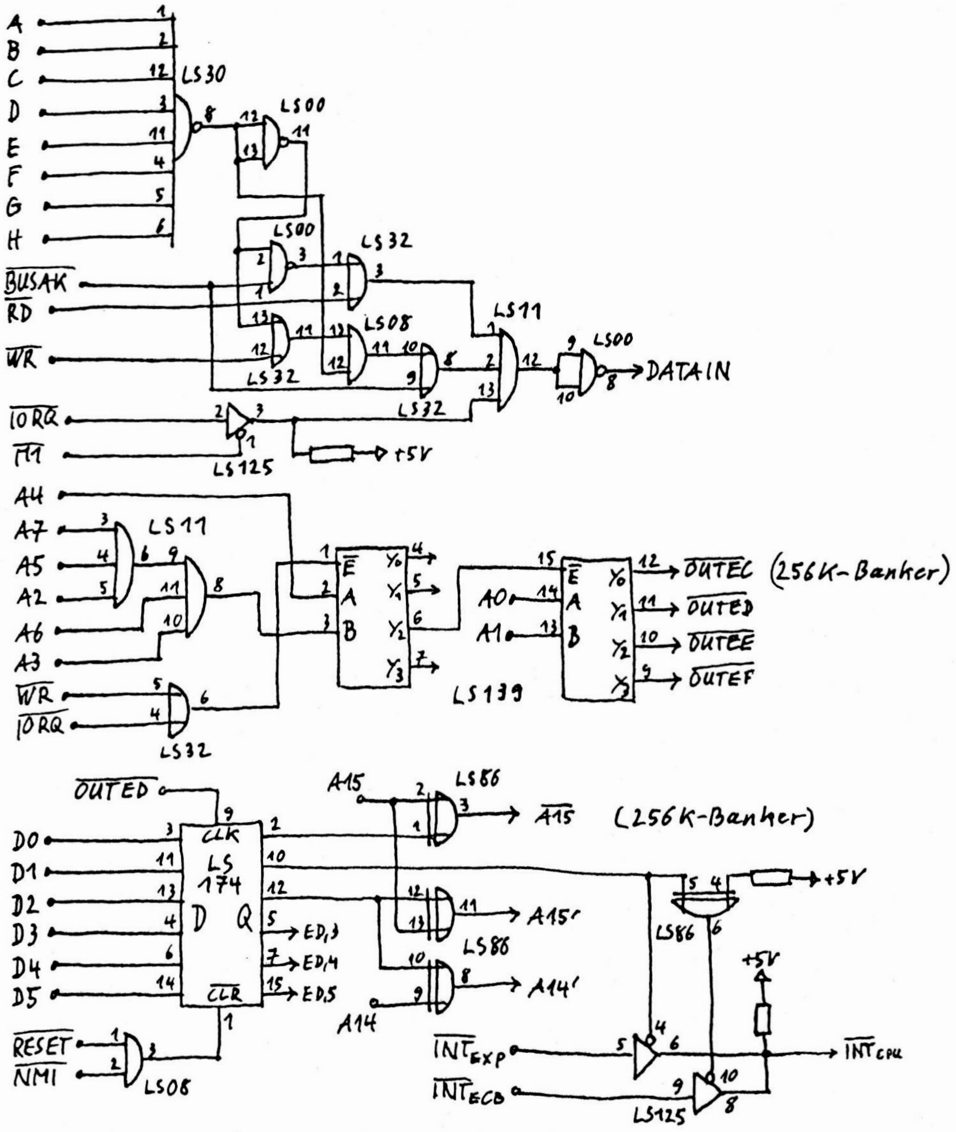
<6> Hardware-Umschaltung für den Betrieb von CP/M bei den Computern TRS 80, GENIE I und II und Komtek 1 / H. Bernhardt / Info des GENIE/TRS 80-User-Club Bremerhaven, Hardware-Sonderheft 11/85, 1-5

<7> GENIE I und II: Voll decodierter Sonder-ROM; RAM im Bereich 3900H bis 3BFFFH / H. Bernhardt / Info des GENIE/TRS 80-User-Club Bremerhaven 11/-85,....?

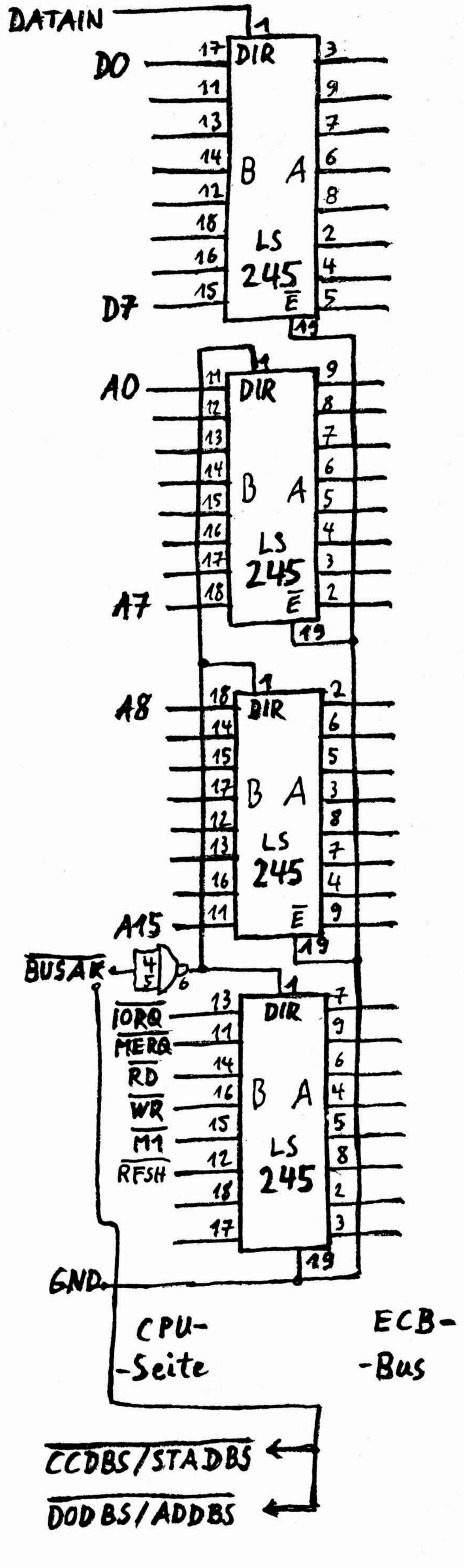
<8> Interrupt Your 80 - Without a hardware mod, your 80 is immune to rude interruptions / Douglas C. Fisher / 80 micro, January 1983, 258-266

<9> CP/M Hardware für TRS-80, GENIE I und II und Komtek 1 / H. Bernhardt, c.Ueberschaar, Info des TRS 80/GENIE-User-Clubs Bremerhaven

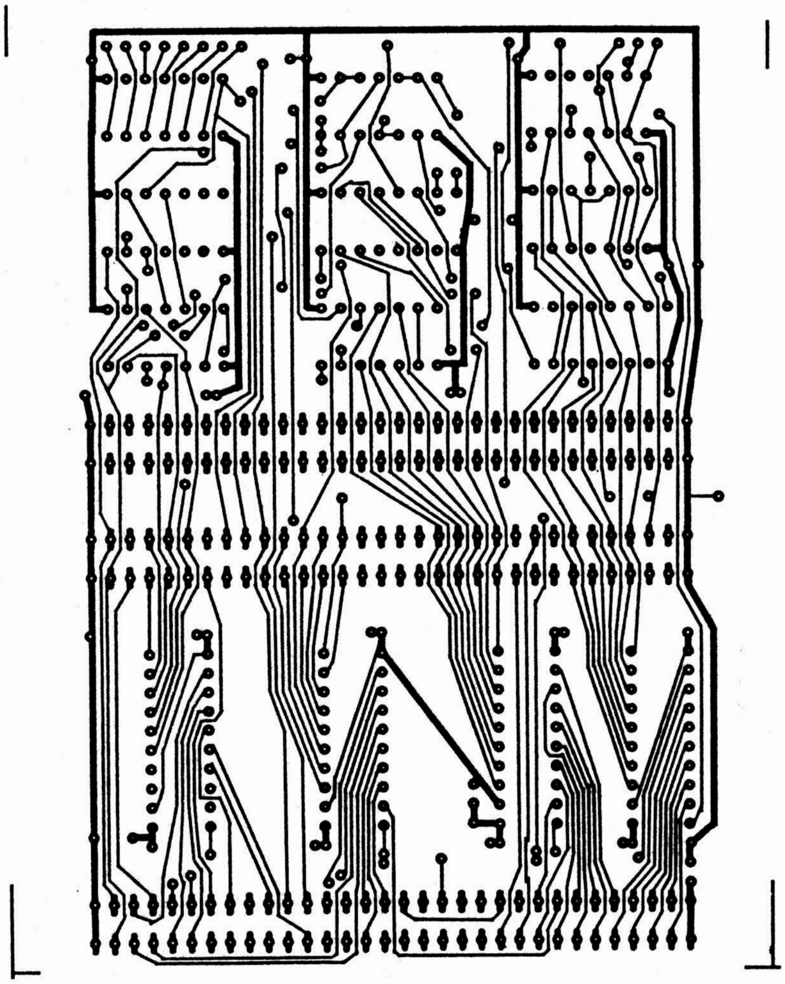
<10> 256K- (1M-) RAM für Z80-Systeme / H. Bernhardt / Info des Club 80 ,Nr.14,



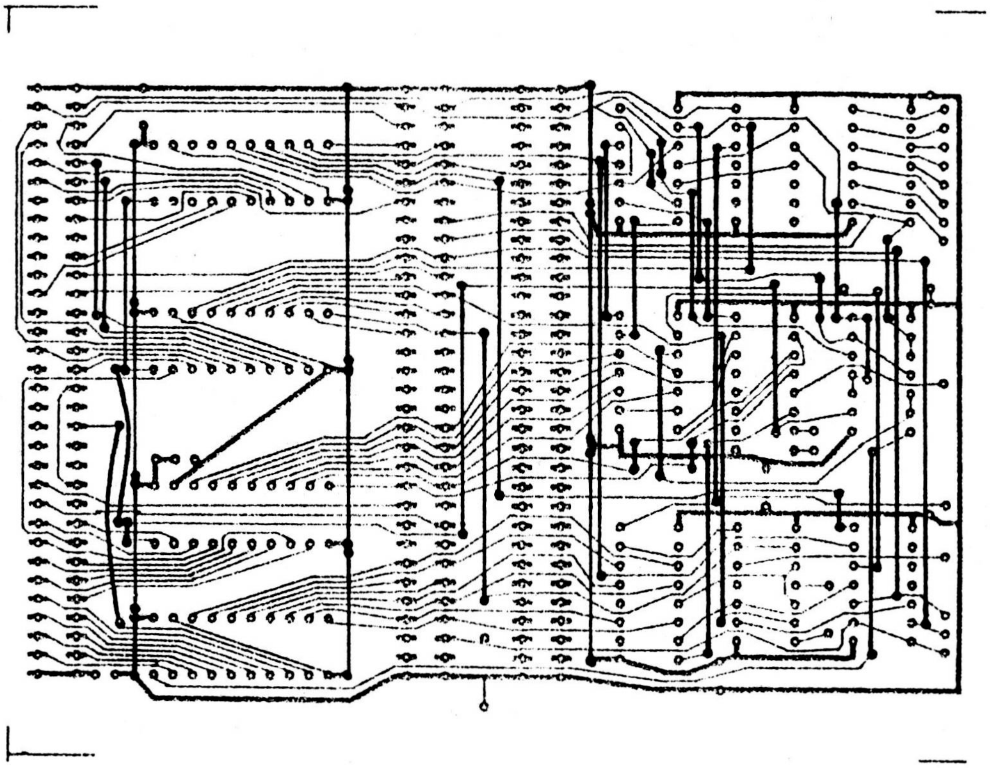
1) Steuerung des Databuffers, Portdecodierung, programmierbares Invertieren von A15 sowie A14 UND A15, Umschalten der Interrupts



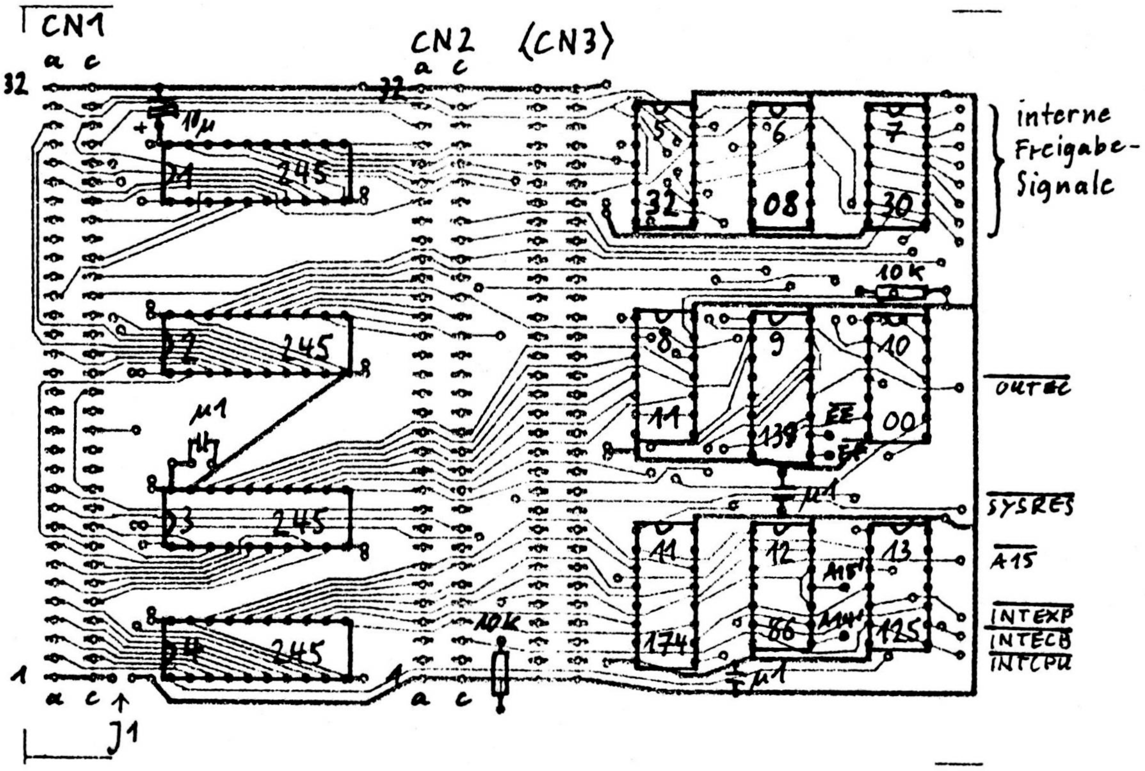
2) Pufferung des ECB-Bus



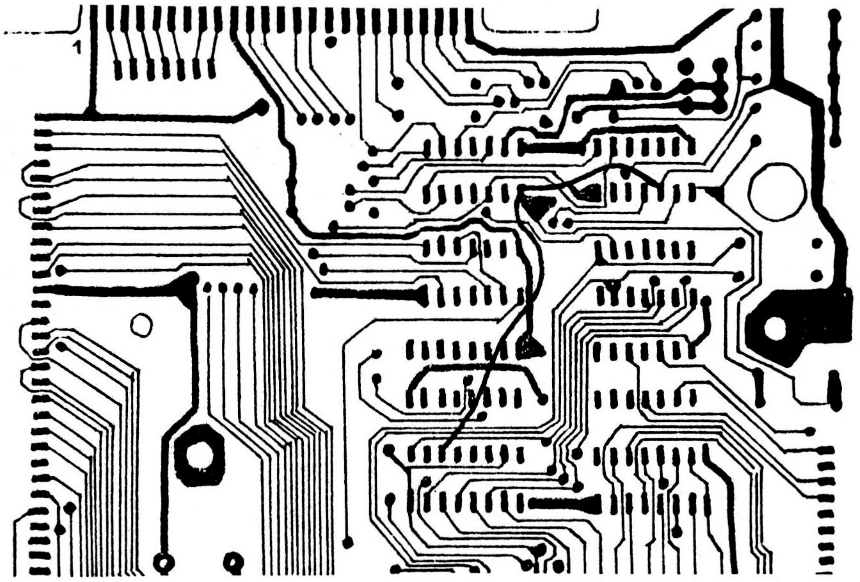
3) Layout der Eingangskarte



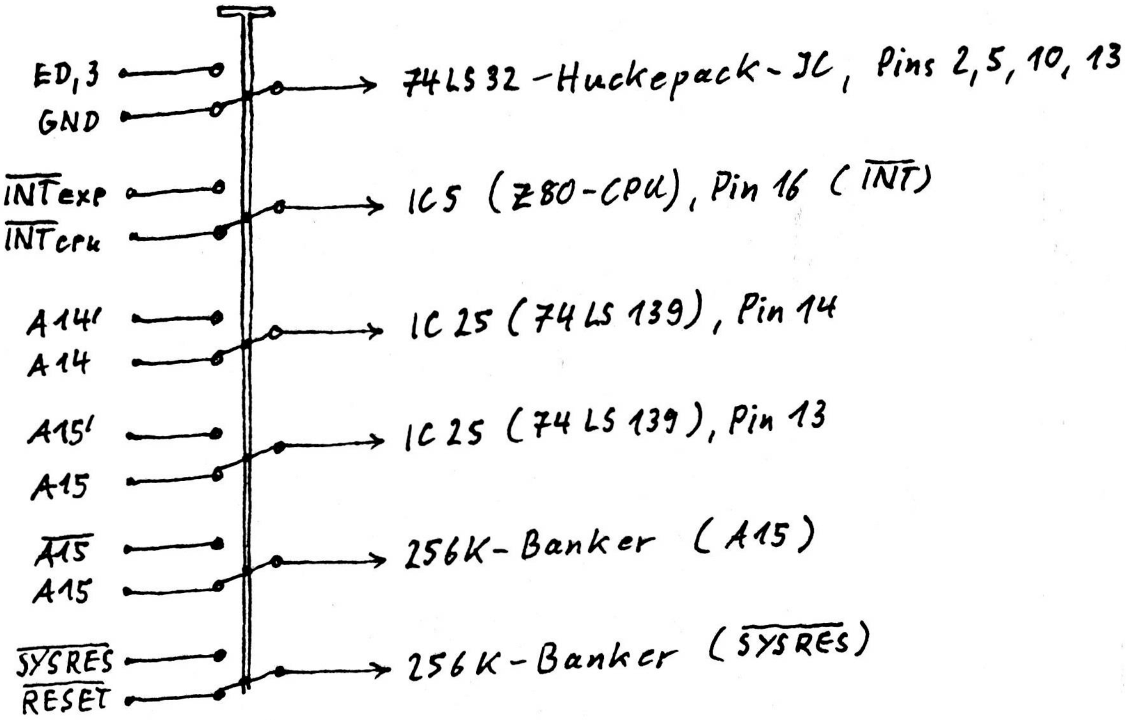
4) Drahtbrückenplan, die Drahtbrücken sind vor der Bestückung aller anderen Bauteile zu legen



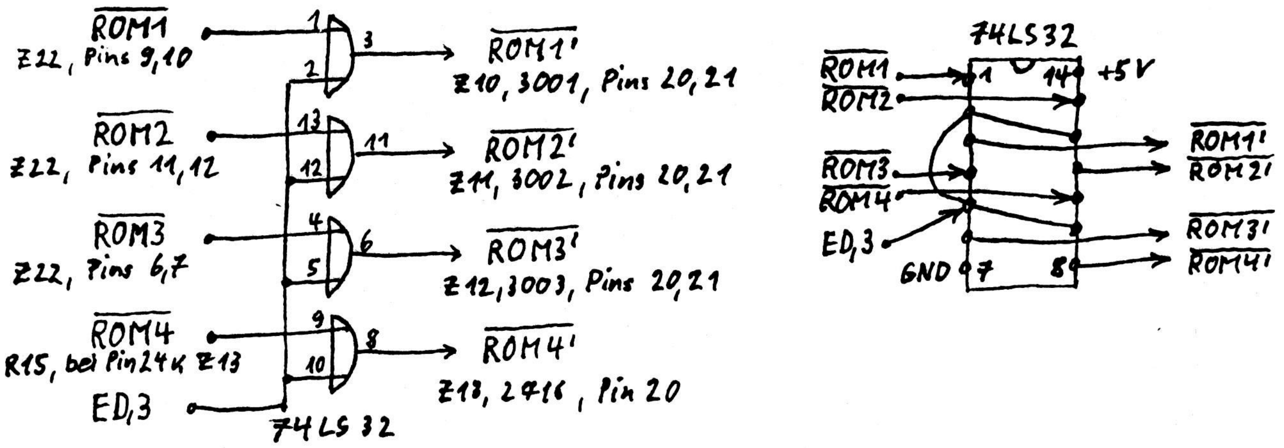
5) Bestückungsplan, J1 ist nur dann zu legen, wenn im GENIE ein stärkeres Netzteil eingebaut wurde, das den ECB-Bus mitversorgen soll



8) Verbinden von BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS*



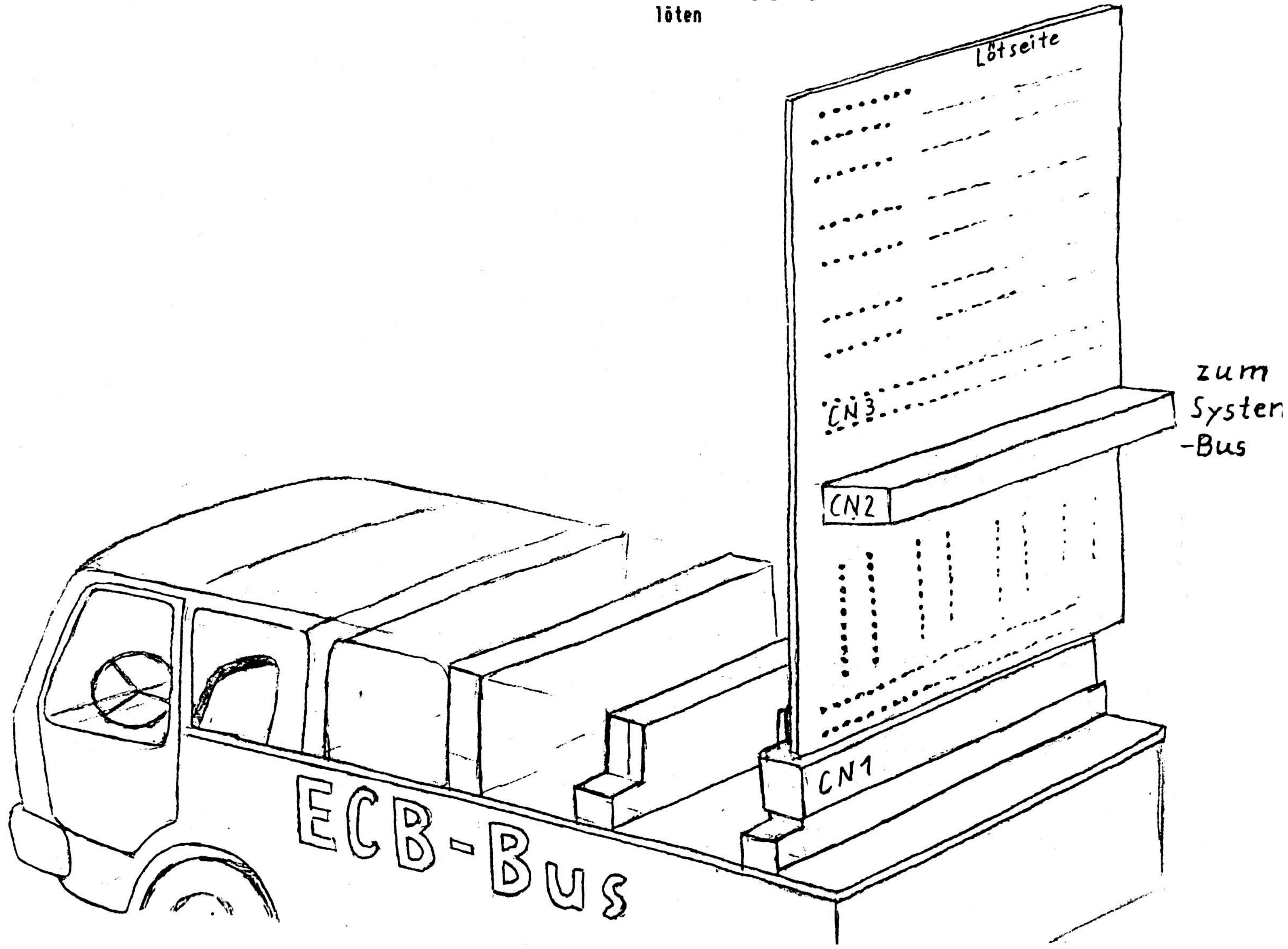
9) Umschalten zwischen originalen GENIE-Signalen und steuerbaren Signalen der Eingangkarte



6) Beschaltung des Huckepack 74LS32 auf dem CPU-Board zum Ausblenden der ROMs

ROM1*	Z22, 74LS156, Pins 9,10	ROM1'*	Z10, 3001, Pins 20,21
ROM2*	" " " 11,12	ROM2'*	Z11, 3002, Pins 20,21
ROM3*	" " " 6,7	ROM3'*	Z12, 3003, Pins 20,21
ROM4*	R15, Seite zu Pin24 von Z13 (ROM4)	ROM4'*	Z13, 2716, Pin 20

diese Signale sind an die aus der Fassung gebogenen Pins zu löten



7) Verbinden des ECB-Bus mit dem Systembus des Computers über die Eingangskarte

----- ECB-BUS Belegung (CONITEC-Norm) -----

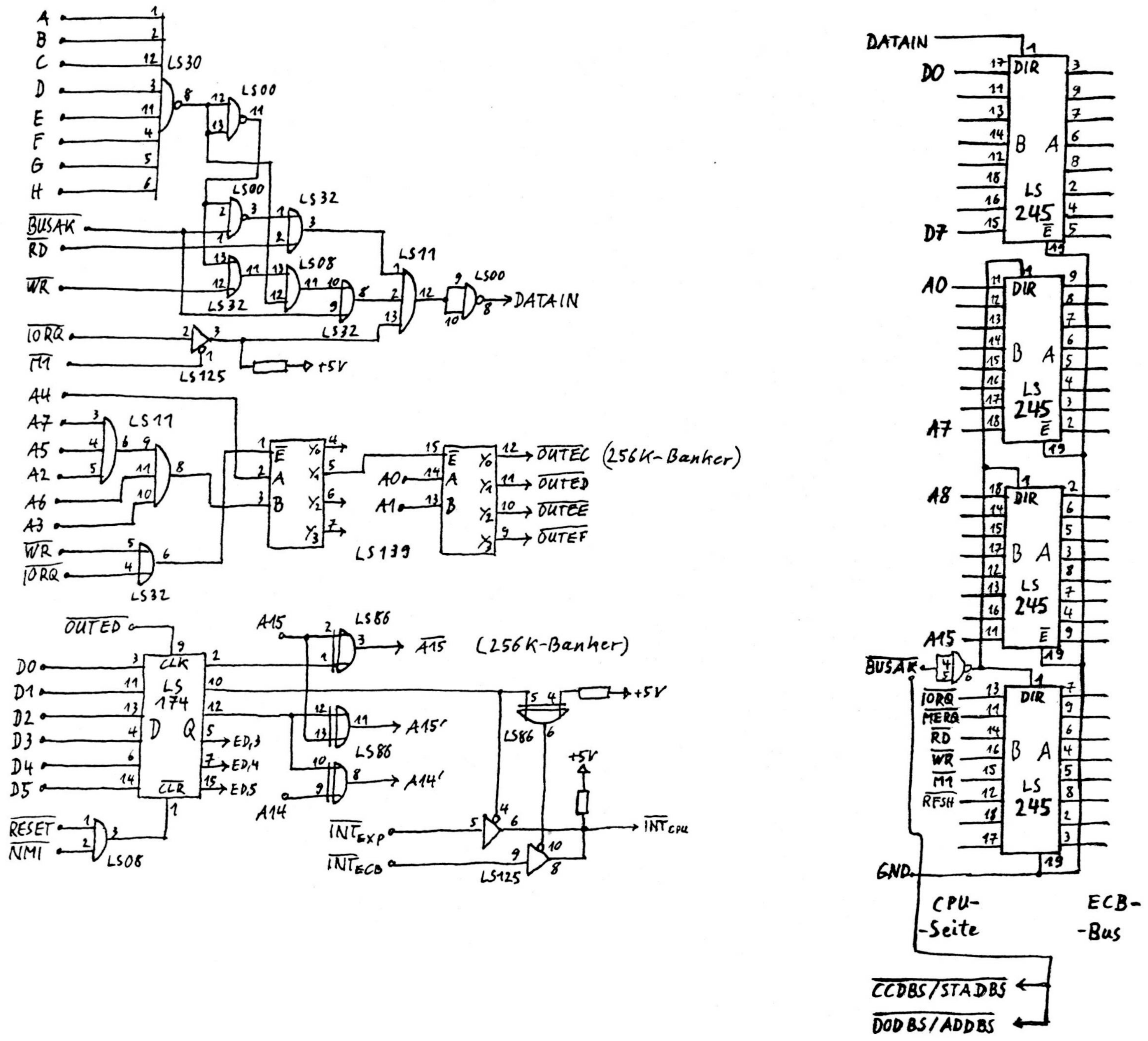
Reihe	a	b	c	Bedeutung:	
1:	+5V	xxx	+5V	+/-5V/12V:	Betriebsspannungen
2:	D5	xxx	D0	UBAT:	Akku (+3.6 Volt)
3:	D6	xxx	D7	GND:	gemeinsame Masse
4:	D3	xxx	D2		
5:	D4	xxx	A0	D0-D7:	8-Bit-Datenbus
6:	A2	xxx	A3	A0-A15:	16-Bit-Adressbus
7:	A4	xxx	A1	A16-A19:	20-Bit-Adressen
8:	A5	xxx	A8	A19/MS:	MASTER/SLAVE-Signal
9:	A6	xxx	A7		
10:	/WAIT	xxx	xxx	/RD:	Lesen
11:	/BUSRQ	xxx	...IEI	/WR:	Schreiben
12:	A18 !	xxx	: A19/MS !	/MREQ:	Speicherzugriff
13:	+12V	xxx	: xxx	/IORQ:	Ein/Ausgabe
14:	-12V !	xxx	: D1		
15:	-5V	xxx	: -	/NMI:	Vorrang-Interrupt
16:	-	xxx	:..IEO	/INT:	normaler Interrupt
17:	A17 !	xxx	A11	IEI:	INT-Ketteneingang
18:	A14	xxx	A10	IEO:	INT-Kettenausgang
19:	-	xxx	A16 !		
20:	/M1	xxx	/NMI	/BUSRQ:	DMA-Busanforderung
21:	xxx	xxx	/INT	/BUSAK:	DMA-Busfreigabe
22:	-	xxx	/WR	/BAI:	DMA-Ketteneingang
23:	.../BAI !	xxx	-	/BAO:	DMA-Kettenausgang
24:	: UBAT	xxx	/RD	/RDY:	DMA-Handshake
25:	:../BAO !	xxx	/HALT		
26:	/RDY !	xxx	/PCL	/RESET:	Rücksetz-Eingang
27:	/IORQ	xxx	A12	/PCL:	Initialisieren
28:	/RFSH	xxx	A15	/WAIT:	CPU-Wartezyklus
29:	A13	xxx	CLOCK	/HALT:	CPU gestoppt
30:	A9	xxx	/MREQ	/M1:	1. Maschinenzyklus
31:	/BUSAK	xxx	/RESET	/RFSH:	Auffrischzyklus
32:	GND	GND	GND	CLOCK:	CPU-Takt (4-8 MHz)

xxx = reserviert für Erweiterungen ... = Kettenleitungen (Daisy Chain)
\- = zur freien Verwendung ! = abweichend von der KONTRON-Norm

----- ECB-Busnormen anderer Hersteller auf einen Blick -----

	CONITEC	E-LAB	ELZET	J&K	KONTRON	MARFLOW	miro	o+r
c10	xxx	A21	--	A16	MBS0	PHI/2	--	A16
a12	A18	A18	A18	/BAI	/BAI	A18	A18	/BAI
c12	A19	A19	A19	A17	MBS1	A19	A19	A17
c13	xxx	-12V	--	A18	MBS2	-12V	A20	A18
a14	-12V	/MMUDIS	-12V	A19	MBS3	xxx	--	A19
a16	--	2PHI	2PHI	2PHI	2PHI	nPHI	2PHI	--
a17	A17	A17	A17	/BAO	/BAO	A17	A17	/BAO
c19	A16	A16	A16	--	MBS5	A16	A16	--
a21	xxx	BDCL	--	--	MBS6	xxx	A22	--
a22	--	/RDY	/BOOT	/RDY	MBS7	xxx	A23	--
a23	/BAI	/BAI	/BAI	/PF	DPR	/BAI	/BAI	--
c23	--	A20	/VBLANK	/SEL	MBS4	xxx	A21	--
a25	/BAO	/BAO	/BAO	--	nPHI	/BAO	/BAO	--
a26	/RDY	MEMD	--	/DESLCT	WREN	xxx	/RDY	/DESLCT

Obige Tabelle ohne Gewähr. Achtung! Die unterschiedliche Belegung der -12V-Leitungen kann bei einigen Kombinationen von Karten und Bussen verschiedener Hersteller zu starker Rauchentwicklung bei der Inbetriebnahme führen! Deshalb -12V vorsichtshalber immer vorher abklemmen.


LS30
LS00
LS32
LS08
LS11
LS125
LS139
LS174
LS86
LS245
DATAIN
BUSAK
RD
WR
IORQ
M1
+5V
OUTEC (256K-Banker)
OUTED
OUTEE
OUTEF
A15 (256K-Banker)
A15'
A14'
ED,3
ED,4
ED,5
RESET
NMI
INT EXP
INT ECB
INT CPU
MERQ
RFSH
GND
CPU-
-Seite
ECB-
-Bus
CCDBS/STADBS
DODBS/ADDBS

Natuerlich kann man mit entsprechenden Aenderungen in jedem Z80-Computer den Z80 gegen einen HD64180 austauschen und hat damit die Leistungsfaehigkeit dieser CPU zur Verfuegung. Man kann aber auch den Z80 beibehalten und zusaetzlich einen HD64180 im System betreiben, wobei sich beide CPUs in der Kontrolle des Computers abwechseln koennen.

By Helmut Bernhardt & Arnulf Sopp

HD64180 als Coprozessor im Z80-System

Helmut Bernhardt

Natürlich kann man mit entsprechenden Änderungen in jedem Z80-Computer den Z80 gegen einen HD64180 austauschen und hat damit die Leistungsfähigkeit dieser CPU zur Verfügung. Man kann aber auch den Z80 beibehalten und zusätzlich einen HD64180 im System betreiben, wobei sich beide CPUs in der Kontrolle des Computers abwechseln können.

Für den Z80 sieht dies so aus, daß er entweder als CPU den Computer steuert oder ganz abgeschaltet ist. Der HD kennt aber zwei verschiedene aktive Zustände. Solange der Z80 Regie führt, muß der HD sich mit 256K RAM und einem EPROM begnügen und hat keinen Zugriff auf den Computer. Wenn der Z80 sich abschaltet und dem HD das System übergibt, kann dieser auf den Computer über die untere Hälfte seines Adreßraumes zugreifen. Sein EPROM ist dann abgeschaltet.

Selbstverständlich muß der HD, wenn er den Computer übernimmt, wissen, was er zu tun hat. Es müssen in der Software beider CPUs Vereinbarungen getroffen werden, die eine Parameterübergabe beim Regierungswechsel ermöglichen. Wenn der HD die Regie an den Z80 übergibt, ist eine solche Parameterübergabe nicht nötig, weil der Z80 seine Arbeit dort wieder aufnimmt, wo er vorher aufgehört hat.

Unter diesen Voraussetzungen ist es möglich, daß der Z80 sich ganz verabschiedet und den HD64180 stattdessen als CPU arbeiten läßt. Andererseits kann der Z80 dem HD ein Programm und einen Satz Daten in den Speicher schreiben, die der HD sich in seine nur ihm zugänglichen oberen 256K RAM kopiert, sich dort hin bankt und die Kontrolle des Computers wieder dem Z80 überläßt. Nun kann er sich in Ruhe mit dem Number-Crunching beschäftigen, bis der Z80 ihn zur Übergabe der Rechenergebnisse auffordert und ihm eventuell neue Aufgaben in den Speicher schreibt.

Eine weitere Beschäftigung des HD kann die Ausgabe von Nachrichten über eine seiner seriellen Schnittstellen sein, ohne daß der währenddessen den Computer steuernde Z80 sich darum kümmern muß, er kann inzwischen Wichtigers verrichten.

Weiterhin kann der HD mit seinen eigenen 256K RAM als intelligente RAM-Floppy mit eigenem Prozessor und eigenen Treiberroutinen mißbraucht werden, wobei dann einer der HD-internen DMA-Controller wirkungsvoll eingesetzt werden kann, um Daten vom/zum Sectorbuffer zu übertragen.

Das Umschalten zwischen den CPUs erfolgt über ein Flip Flop, das nach dem Einschalten des Computers (RESET*) so eingestellt ist, daß der Z80 aktiv ist und die Bustreiber zwischen HD64180 und Computer im hochohmigen Zustand sind. In dieser Kofiguration findet der HD in den unteren 256K seines physikalischen Adreßraumes von 512K sein "Boot-EPROM", das eine Größe von 8 bis 64K-Byte haben kann und innerhalb der 256K gespiegelt ist. In den oberen 256K des HD liegt RAM-Speicher vor, der dem HD immer zur Verfügung steht. Im EPROM muß eine Routine vorhanden sein, die zunächst alles Nötige an Intelligenz ins RAM kopiert und dann den HD in das RAM umbankt. Nachdem der volle logische Adreßraum von 64K in das RAM gebankt wurde, braucht der HD zunächst nur zu warten, bis der Z80 ihm den Computer übergibt. Das kann er im einfachsten Fall durch Untersuchen der Beschreibbarkeit eines Bytes in den unteren 256K oder nach einem geringfügigen Hardware-Patch über einen Interrupt an seinem INT2-Eingang feststellen.

Daraufhin hat er an vereinbarter Stelle im Speicher einen Vektor zu lesen und an die durch den Vektor bezeichnete Stelle zu springen. Das dort stehende Programm entscheidet dann darüber, welche Funktion ihm im folgenden zugedacht ist.

Das Umschalten zwischen den CPUs erfolgt immer durch die gerade aktive CPU mit einem OUT-Befehl an Port FBH, wobei das ausgegebene Datum belanglos ist. Wenn zunächst der Z80 diesen OUT-Befehl ausführt, wird sein BUSRQ*-Eingang low gezogen und die internen Treiber der Z80-Bussignale gehen in den hochohmigen Zustand. Gleichzeitig werden die Bustreiber des HD-Boards eingeschaltet, so daß der HD nun den Systembus des Computers kontolliert. Ein Zugriff auf Speicher und memory mapped I/O des Computers ist aber nur möglich, wenn A18 des HD low ist. Wenn dies nicht der Fall ist, bleibt der Databuffer des Boards hochohmig, so daß der HD dann ohne Störungen seine nur ihm zugänglichen 256K RAM in der oberen Hälfte seines Adreßraumes lesen und beschreiben kann.

Der HD kann nun ebenfalls die Buskontrolle durch einen OUT-Befehl an Port FBH wieder dem Z80 übergeben. Dieser Befehl hat aber nur Wirkung, wenn der HD den Computer kontrolliert. Wenn der Z80 aktiv ist, kann der HD sich durch diesen OUT-Befehl nicht die Kontolle über den Computer erzwingen. Es kann nur der jeweils den Computer kontrollierende Prozessor die Kontrolle an den anderen übergeben.

Der Teil des Coprozessor-Boards, der das Umschalten zwischen den CPUs bewirkt und die Steuerung des BUSRQ*-Eingangs des Z80 und die Freigabe der Bustreiber und des EPROMs des HD-Boards ausführt, ist in Abb.1 gezeigt. Andere wesentliche Schaltungselemente sind in Abb.2 wiedergegeben. Die banalen Dinge wie Beschaltung der RAMs, des EPROMs, der V24-Treiber und der Bustreiber (außer deren Steuereingänge) sind nicht dargestellt.

Die Signale der 50poligen Steckerleiste sind mit den entsprechend bezeichneten Pins des Z80-Sockels bzw. bei einem DMA-fähigen Bus alternativ mit den entsprechenden Pins des Bus zu verbinden. Wenn im Computer durch eine MMU Adressen A16 und A17 erzeugt werden, so kann im einfachsten Fall auf die Verwendung von A16 und A17 des HD verzichtet werden, wobei sich der HD auch der MMU des Computers bedienen muß und seine eigene MMU unwirksam bleibt (außer bei der Steuerung von A18 und der Speicheradressierung in den oberen 256K RAM).

Eine andere Möglichkeit besteht darin, entsprechend der gerade aktiven Betriebsart über einen Multiplexer entweder A16 und A17 der MMU des Computers oder A16 und A17 des HD durchzuschalten. Wenn die MMU des Computers auch ein 8Bit-Refresh für 256K-Bit-RAMs erzeugt, müssen ebenfalls A7 der MMU des Computers und des HD gemultiplext werden. Auch

andere durch die MMU des Computers beeinflußte Adressen sind zu multiplexen (A15, A14...?).

Je nach aktiver CPU muß in den Computer entweder der ursprüngliche Takt oder der Takt des HD64180 (Pin 64) eingespeist werden. Es muß also im Computer ein steuerbarer Treiber vorgesehen werden, der den ursprünglichen Systemtakt abschalten kann, wenn der HD das System übernimmt, denn dann wird der HD-Takt über den Bustreiber an den Computer geliefert.

Wenn die RAM-Steuersignale RAS*, MUX* und CAS* des Computers aus dem Systemtakt und aus Z80-Signalen hergeleitet werden, sind hier keine Änderungen nötig. Wenn allerdings dieses Timing aus einem höheren Takt (aus dem eventuell auch der CPU-Takt hergeleitet wird) oder über Gatterlaufzeiten erreicht wird, muß eine andere Schaltung dafür verwendet werden. Eine Möglichkeit ist die auf dem HD-Board verwendete Schaltung (Abb.2), die auch im Computer selbst anstelle der vorhandenen eingesetzt werden kann. Bei Computern mit softumschaltbarer Taktrate sind meistens Schaltungen vorhanden, die so beibehalten werden können. Hier braucht dann nur ein Umschalten zwischen ursprünglichem Systemtakt und Takt des HD realisiert zu werden.

Abb. 1: Bussteuerung der Coprozessor-Karte

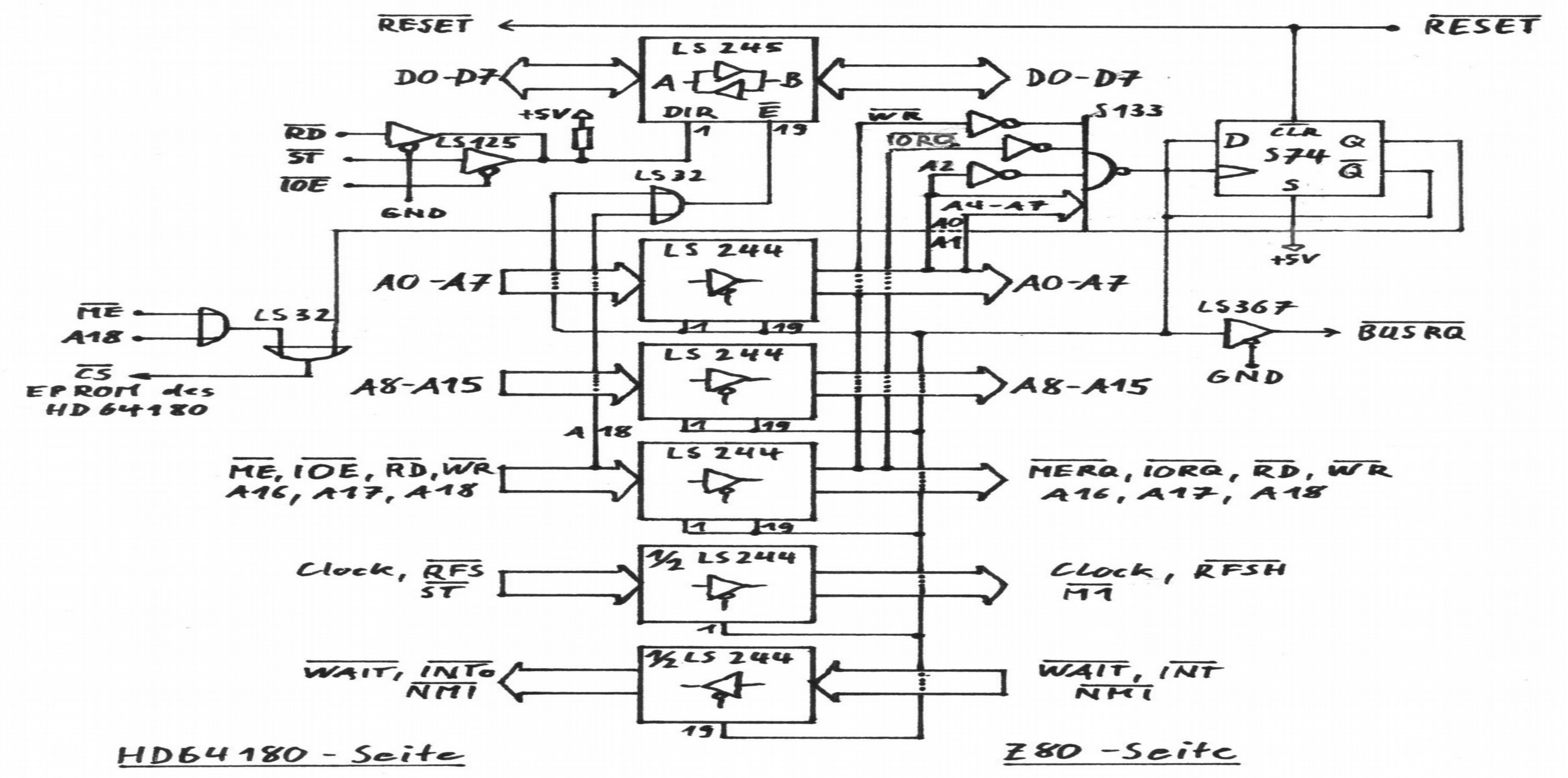


Abb. 2: Steuerung der RAMs der Coprozessor-Karte

Abb. 3: Layout des HD64180-Boards (Ansicht von der Bestückungsseite)

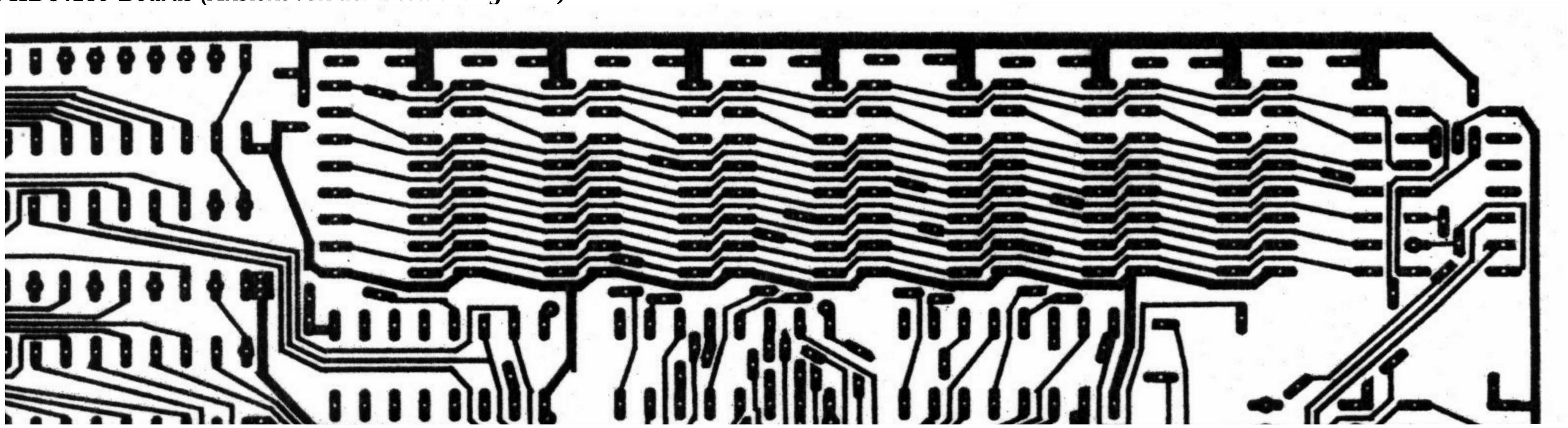


Abb. 4: Drahtbrücken Bestückungsseite

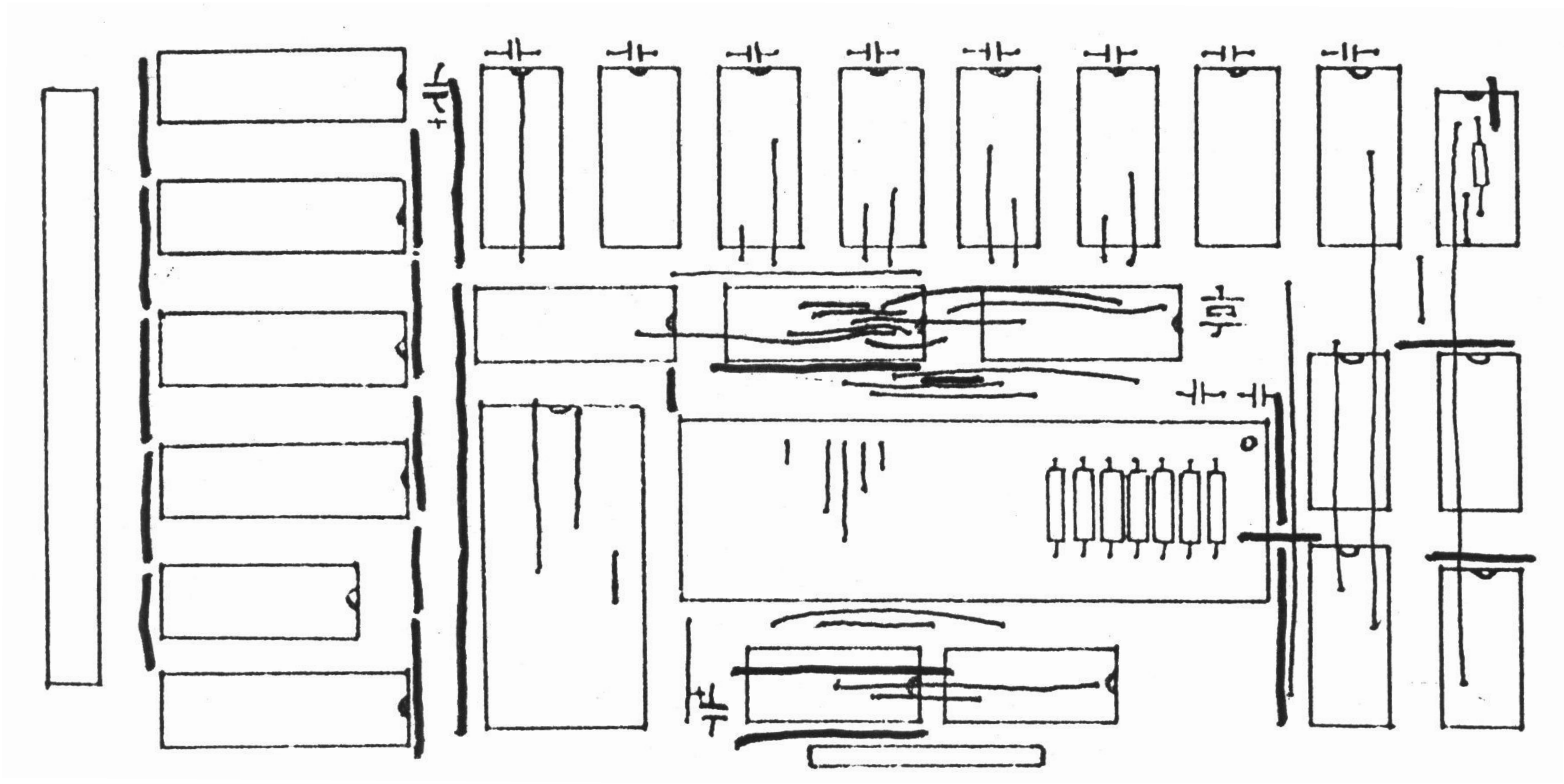


Abb. 5: Bestückungsplan

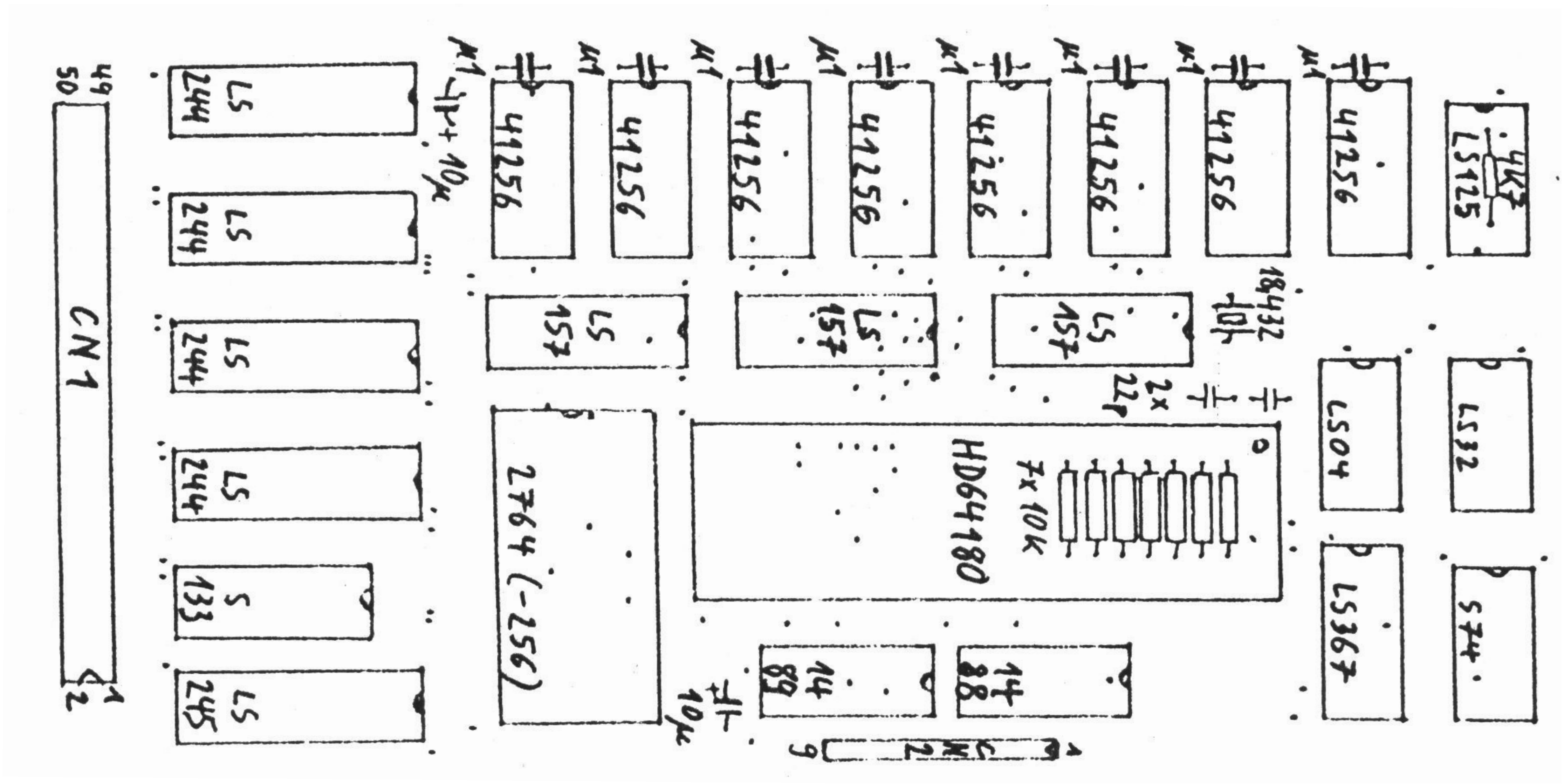
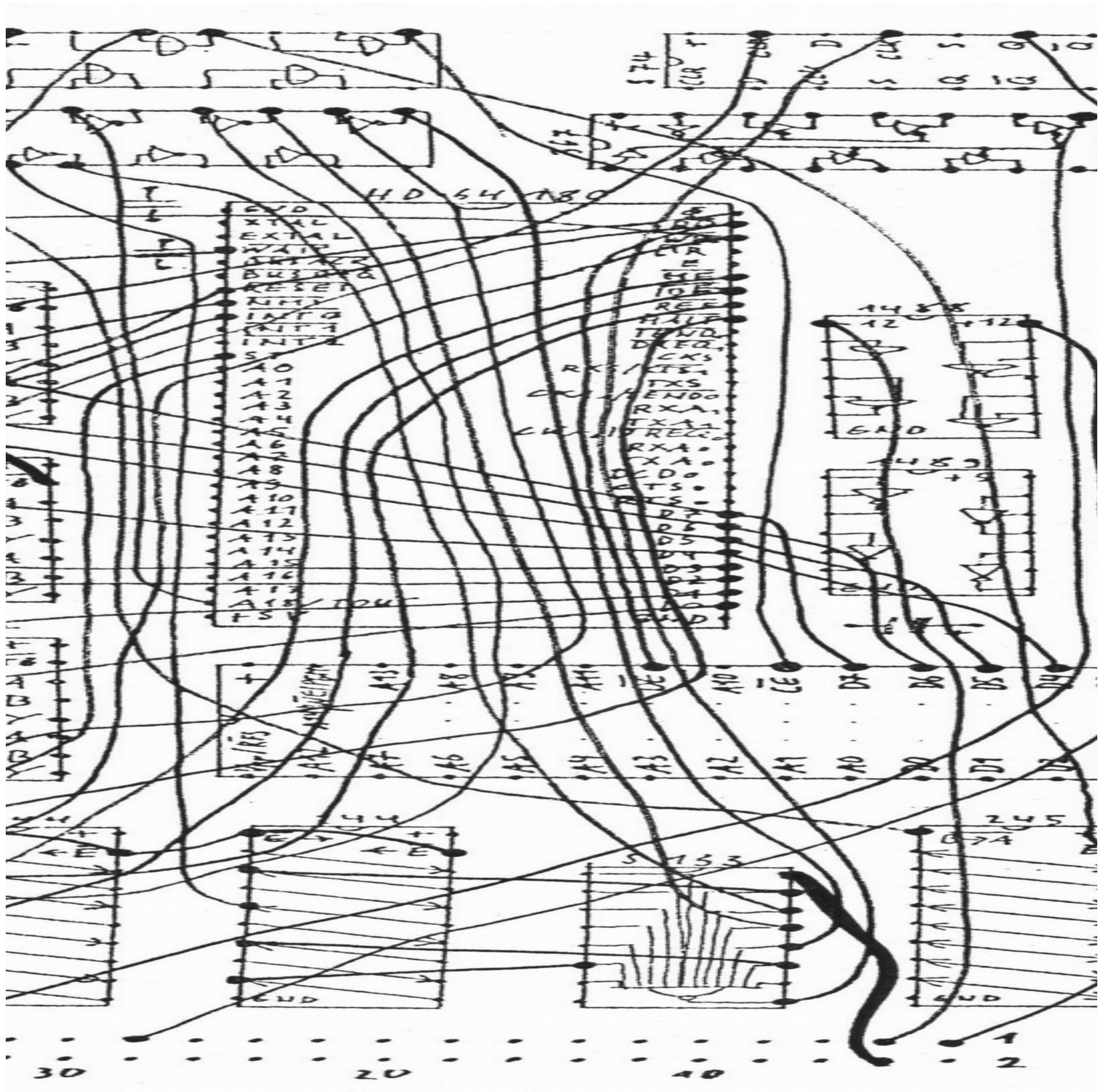


Abb. 6: Drahtbrücken auf der Lötseite (Ansicht von der Bestückungsseite)



Steckerbelegung von CN1
Verbindung zum Z80

Signal	Pin	Pin	Signal
+12V	1	2	+ 5V
-12V	3	4	+ 5V
D3	5	6	D4
D2	7	8	D5
D1	9	10	D6
D0	11	12	D7
A4	13	14	A5
A6	15	16	A7
A0	17	18	Al
A2	19	20	A3
A14	21	22	Al5
Al0	23	24	All
A8	25	26	A9
BUSRQ	27	28	nc
A13	29	30	Al2
A16	31	32	A17
A18	33	34	RD*
IORQ*	35	36	nc out
RESET*	37	38	BUSRQ*
NERO*	39	40	WR*
HALT*	41	42	nc in
NMI*	43	44	Ml*
INT*	45	46	RFSH*
WATT*	47	48	Clock
GND	49	50	GND

Steckerbelegung von CN2
serielle Schnittstellen

Pin	Signal
1	RTS1*
2	TXA0
3	TXA1
4	RTS0*
5	GND
6	CTS0*
7	CTS1*
8	RXA0
9	RXA1

Aufbau der Coprozessor-Karte

Zunächst werden alle Bohrungen mit einem 0,8mm-Bohrer angelegt. Nur die Anschlüsse für die Stecker CN1 und CN2 werden in 1mm Weite gebohrt. Nun können die bestückungsseitigen Drahtbrücken eingesetzt werden. Dafür sollte allgemein dünner Fädeldraht benutzt werden. Nur die im Drahtbrückenplan dicker eingezeichneten Brücken sollen mit Schaltdraht realisiert werden. Anschließend können dann Widerstände, Kondensatoren, der Qartz, die IC-Sockel und die Pfostenstecker eingesetzt werden.

Die abschließende, zeitraubendste Arbeit ist schließlich die freie Verdrahtung mit Fädeldraht auf der Lötseite gemäß Verdrahtungsplan. Auch hier sollen die dick eingezeichneten Verbindungen (für +5V und GND) mit Schaltdraht gelegt werden.

Der Anschluß der Karte kann allgemein mit den entsprechend bezeichneten Pins von CN1 an den Sockel der 280-CPU erfolgen. A16 und A17 können mit entsprechenden Signalen eventuell vorhandener Banking-Schaltungen gemultiplext werden, wobei dann BUSRQ* als Steuersignal verwendet wird.

Die Versorgungsspannungen +12V und -12V für den Schnittstellentreiber des HD64180 sind nur dann anzulegen, wenn die seriellen Schnittstellen des HD aufgebaut werden. Wenn diese nicht benötigt werden, brauchen die ICs 1488 und 1489 sowie der Stecker CN2 nicht eingesetzt zu werden.

Je nach Computer ist dann noch eine mehr oder weniger aufwendige Umschaltung des Systemtaktes und eventuell eine andere Erzeugung der RAM-Steuersignale RAS*, MUX* und CAS* nötig. Wenn außer RAM auch noch andere Baugruppen memory-mapped sind, müssen in die Freigabesignale von RAM und diesen Baugruppen auch noch A16 17 eingehen, um bei Adressierung durch den HD diese Baugruppen innerhalb der möglichen 256K nicht 4mal vorliegen zu haben. Bei Computern mit nur 64K RAM bleiben A16 und A17 unberücksichtigt.

Durch Ersetzen der 4164-RAMs durch 41256-RAMs und Einsetzen eines 8Bit-RefreshGenerators läßt sich der Speicher des Computers selbst auf 256K erweitern, ohne daß eine zusätzliche MMU nötig ist. Dann kann aber nur der HD diesen Speicher voll adressieren; der Z80 hat weiterhin nur auf 64K Speicher Zugriff. Dabei wären dann A16 und A17 des HD mit GND unter Kontolle von BUSRQ* zu multiplexen.

Wenn der HD die Kontrolle über den Computer hat und in seinen oberen 256K adressiert, ist der Datentreiber des HD64180-Boards gesperrt. Die Adressen und Steuersignale werden aber zum Computer durchgeschaltet, um dort ein Refreshing der RAMs aufrecht zu erhalten. Bei Schreibzugriffen auf die oberen 256K des HD erfolgen also auch Schreibzugriffe auf die entsprechende Adresse in den unteren 256K, nur daß dort dann nicht die Daten des HD sondern zufällige Werte des offenen Datenbus ankommen. Um dies zu verhindern, muß A18 so in die Herleitung aller memory-mapped Baugruppen eingreifen, daß ein Zugriff auf diese Baugruppen nur dann verhindert wird, wenn das Coprozessor-Board angeschlossen ist, der HD den Computer kontrolliert und auf seine oberen 256K RAM zugreift. Wenn das Board nicht angeschlossen ist oder wenn der HD bei Kontrolle des Computers in den unteren 256K des Computers adressiert (A18 ist low) müssen diese Baugruppen freigegeben werden, wenn sie adressiert werden. Dies leistet ein Treiber eines 74LS125-ICs und ein Widerstand. Das dabei gewonnene Signal MMEN* ist im Computer an entsprechenden Stellen in die Herleitung solcher Freigabesignale mit einzubeziehen (siehe unten).

Abb. 7: 8-Bit-Refresh Generator

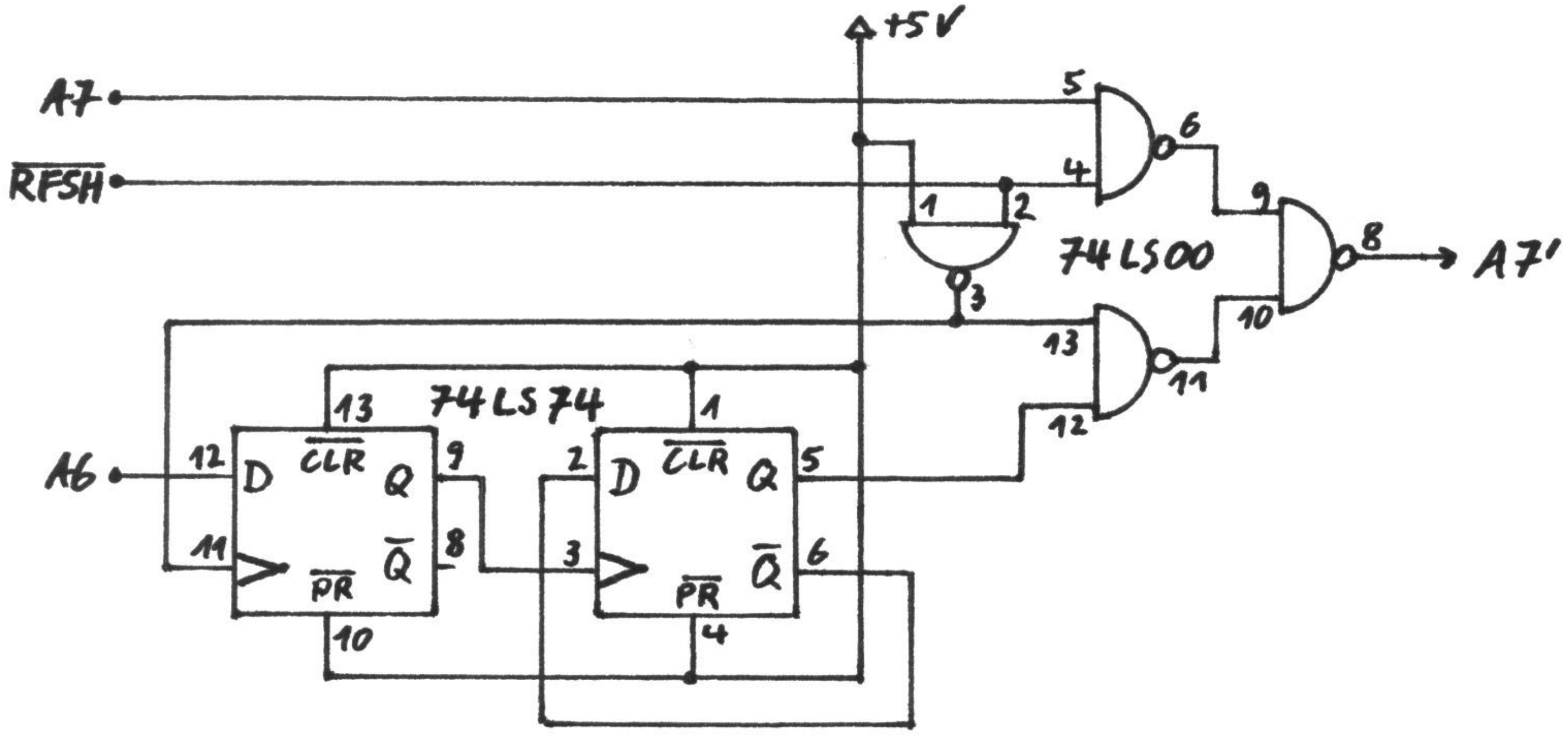


Abb. 8: Erzeugen der Adresse MA8 für 41256-RAMs in Computern ohne Banker

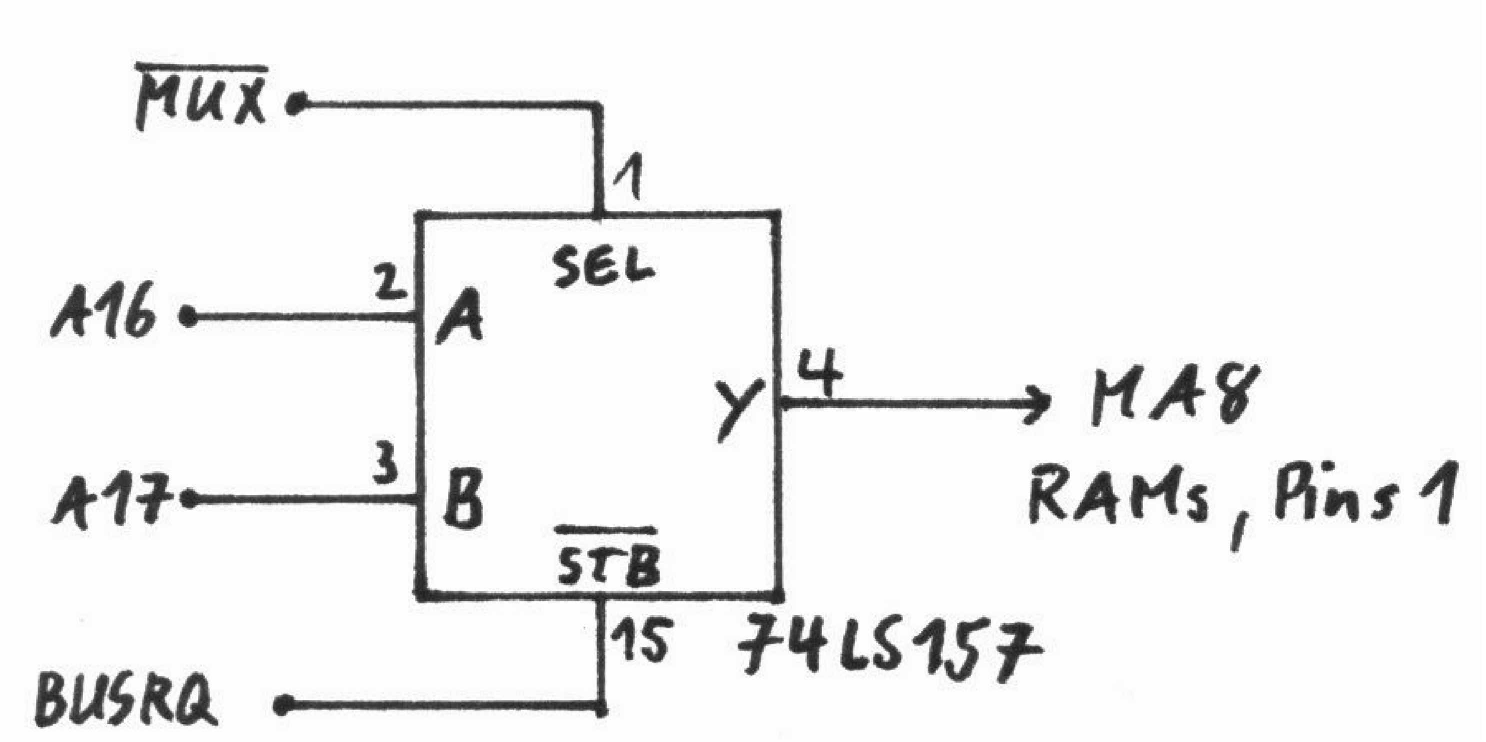


Abb. 9: Erzeugen eines Int2-Interupts bei Out FBh des Z80

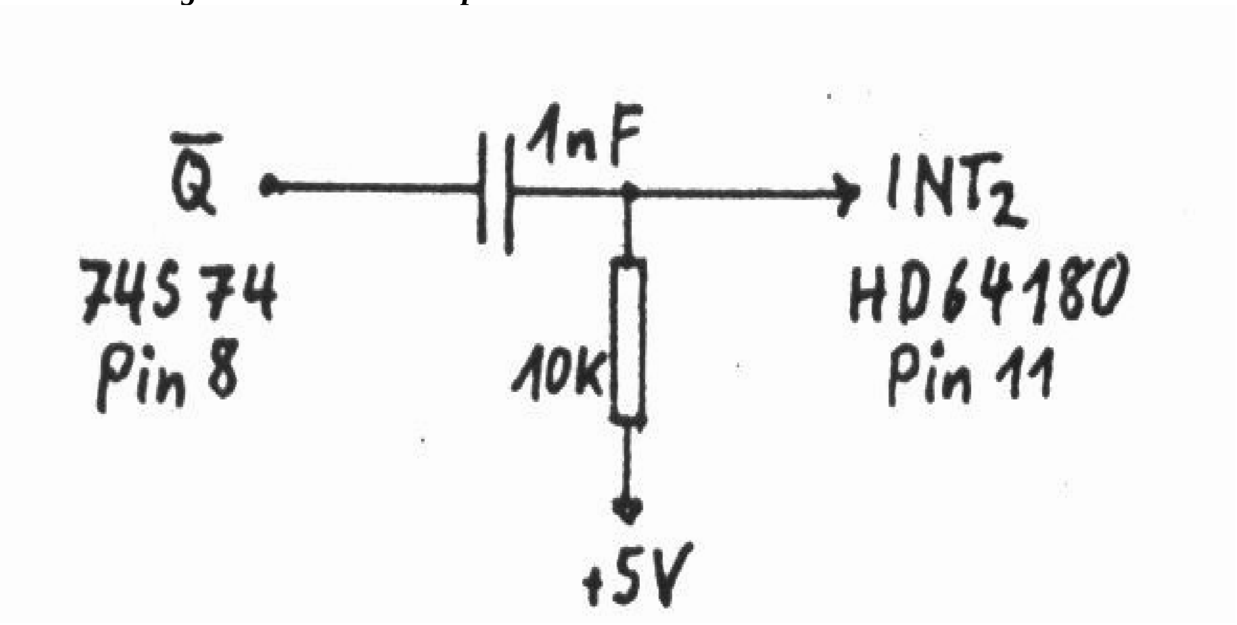
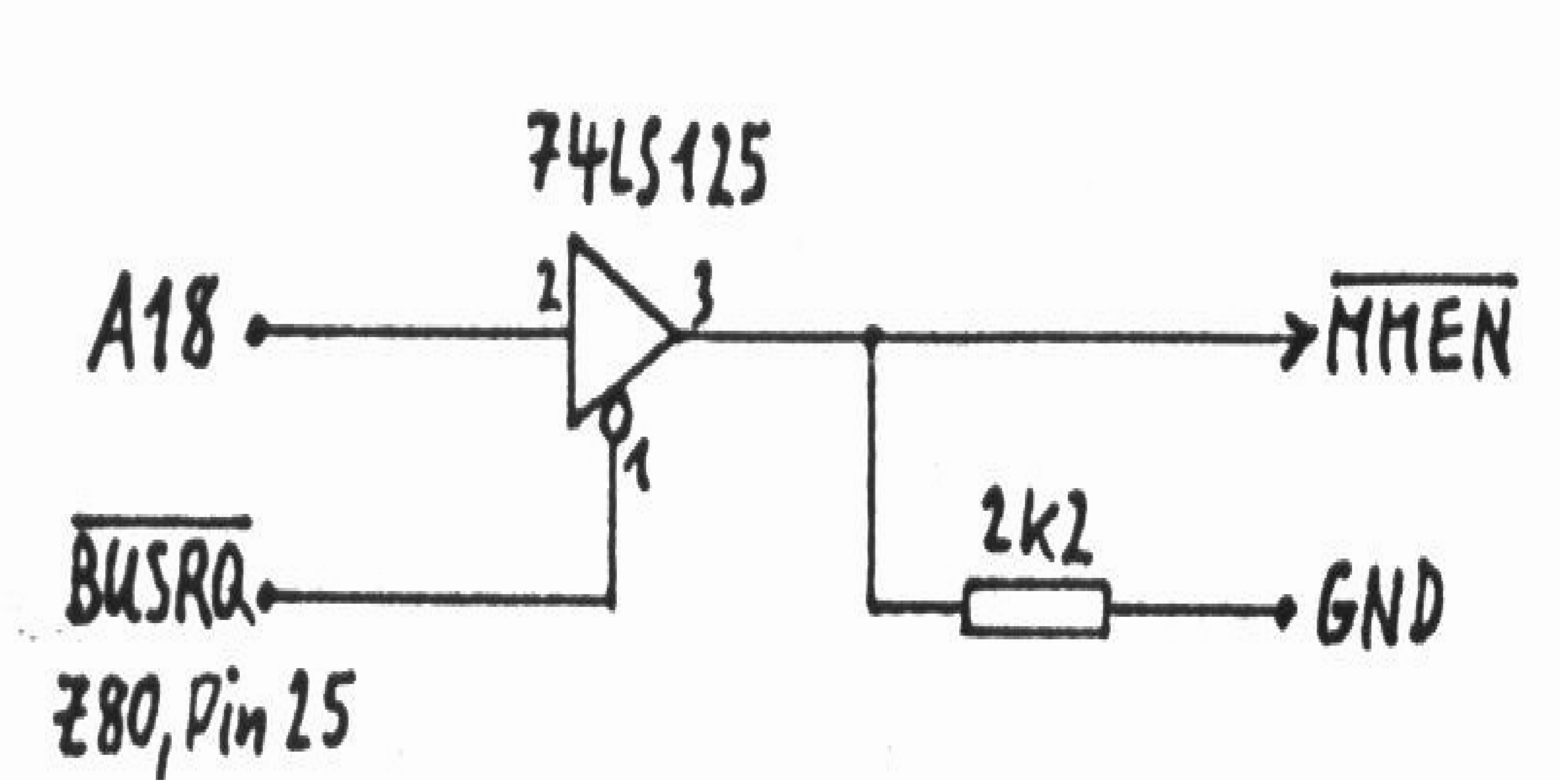


Abb. 10: Herleiten eines Steuersignals für die Freigbe von memory-mapped Baugruppen

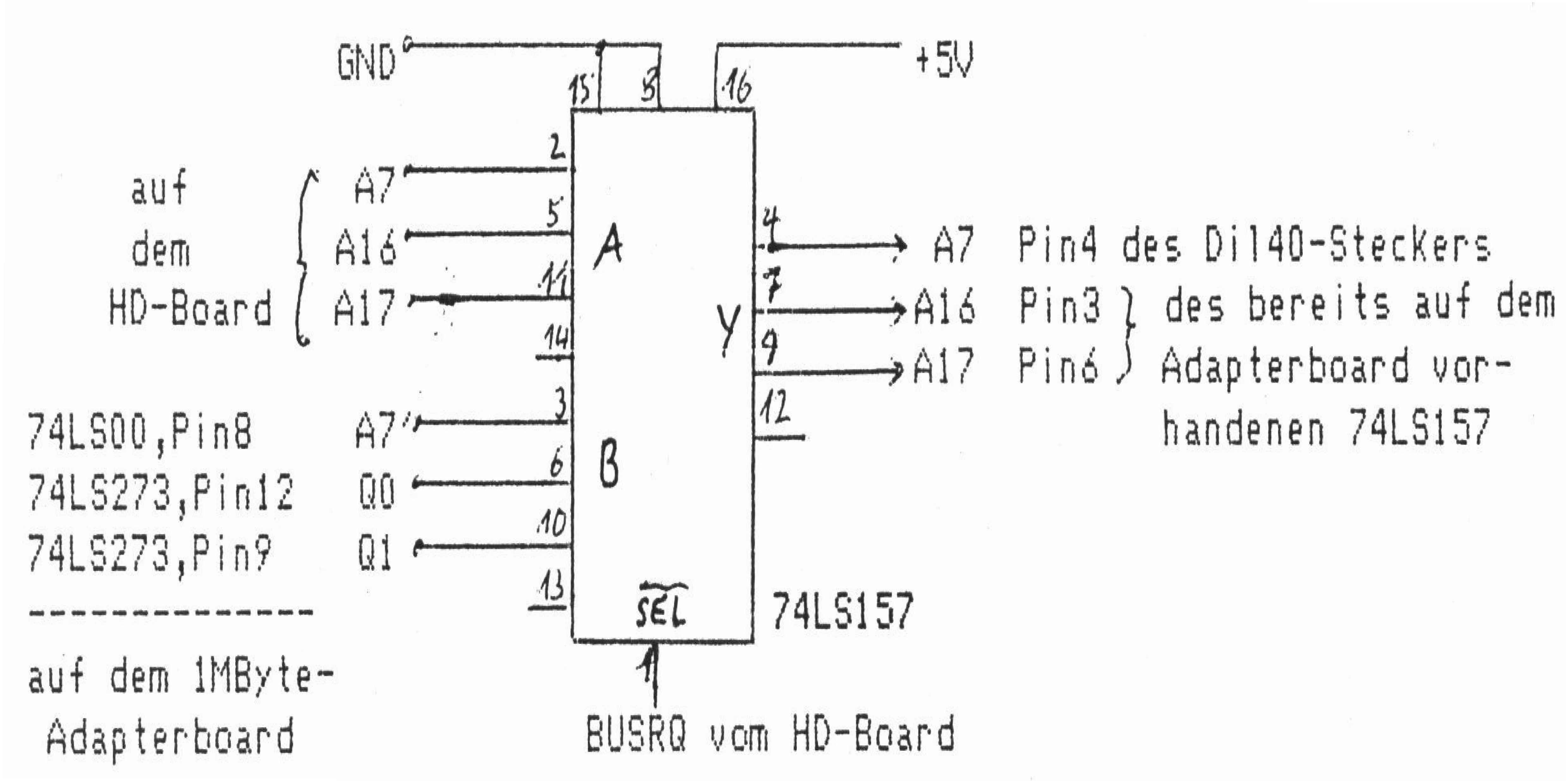


Änderungen im Computer

1) GENIE IIIs mit Erweiterung auf 1M Byte gemäß "Dein GIIIs, die 4 unbekannten Wesen"; Sonderheft des Club 80 Infos, 1986

Auf dem Adapterboard für die 1M Byte-Erweiterung wird das freie Rasterfeld für ein Dil-14-IC auf ein Dil-16-Feld erweitert, wobei dann GND vom Pin 7 auf den neu hinzugekommenen Pin 8 verlegt wird. Hier wird ein 74LS157-IC eingesetzt und gemäß folgender Abbildung beschaltet.

Abb. 11: Multiplex der höheren Adressen von HD64180 und 1M-Erweiterung



Die vorhandenen Verbindungen der Signale zu den B-Eingängen dieses zusätzlichen LS157 mit den Zielpunkten der '-Ausgänge sind zu durchtrennen.

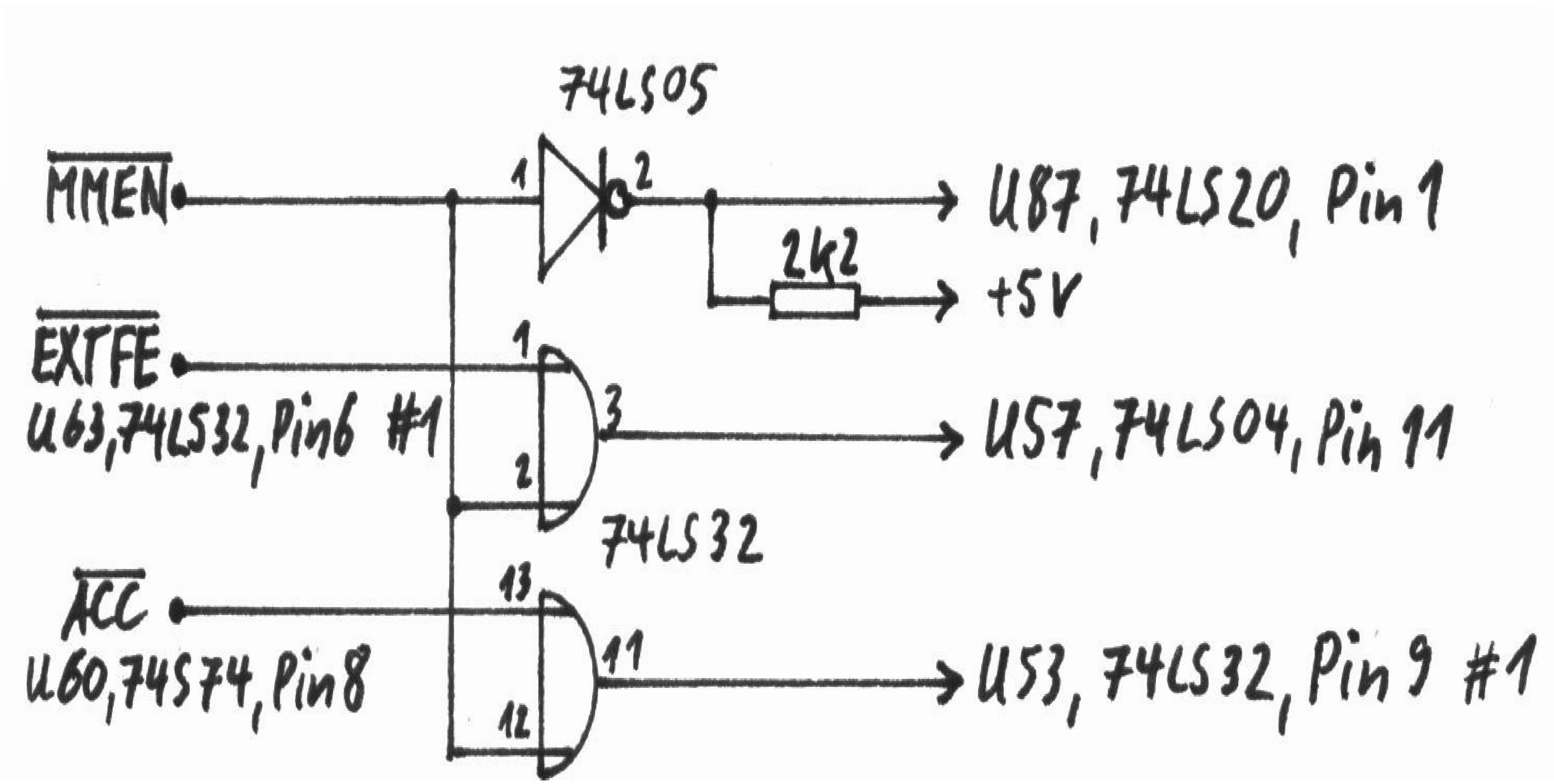
Durch diese Änderung wird nur das über DO und D1 von Port F3H bewirkte Banking durch die MMU des HD64180 ersetzt, wenn dieser die Regie übernimmt. Das ursprüngliche Banking von nur 256K des GIIIs bleibt auch bei Kontrolle des Computers durch den HD bestehen. Einschränkend für die flexible MMU des HD bleibt hier aber auch die in Hardware durch D6 und D7 von Port F9H einzustellende Common-Bedingung wirksam. Die Programmierung der MMU des HD sollte diesen Voraussetzungen angepaßt werden Common entweder bei 0-16K oder bei 56-64K).

Das Umschalten des Systemtaktes kann durch Zwischenschalten eines Treibers mit Freigabe-Eingang (z.B. 74LS125) zwischen GIIIs-Taktquelle, U44, 74LS161, Pin12 und U39, 74L5125, Pin6 erfolgen. Die Freigabe des Treibers erfolgt durch BUSRQ, Pin27 von CN1 des HD-Boards. An Pin5 von U39 wird außerdem auch das Clock-Signal der Steckerleiste CN1 des HD-Boards gelegt. Durch die Steuerung des Treibers des GIIIs-Taktes und der Bustreiber des HD-Boards ist gewährleistet, daß nur einer der beiden Takte dem Computer zugeführt wird.

Abb. 12: Taktumschaltung zwischen System-Clock und HD-Clock

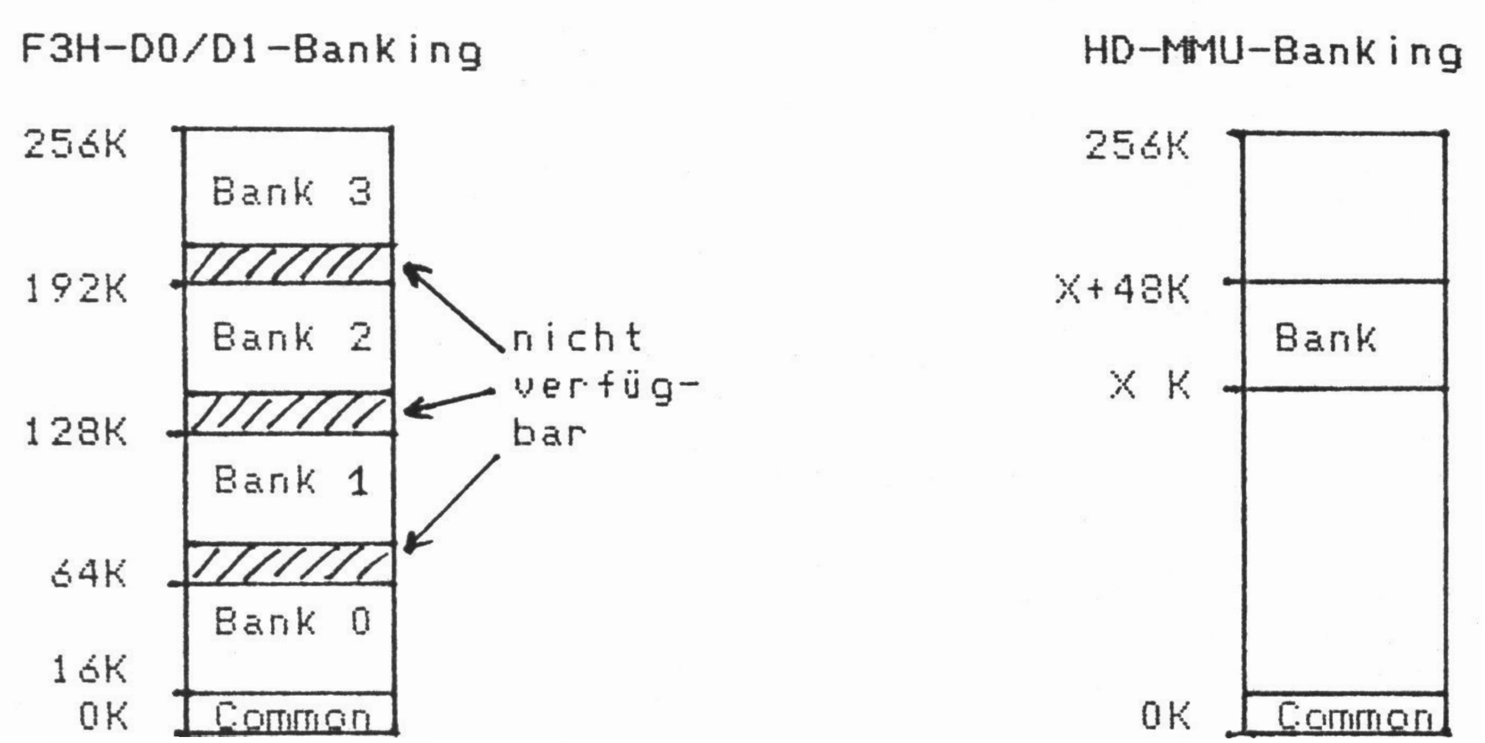
Um den Zugriff auf memory-mapped Baugruppen bei Adressierung der oberen 256K des HD zu verhindern, ist das aus A18 und BUSRQ* hergeleitete Steuersignal MMEN* folgendermaßen zu verwenden. Die dabei einbezogenen Pins (U62, 74LS32, Pin6 und U53, 74LS32, Pin9) sind aus der Fassung zu biegen.

Abb. 13: Eingriff in die Erzeugung der Freigabesignale der memory-mapped Baugruppen des GIIIs



Durch die Hardware des GIIIs ist vorgegeben, daß innerhalb der ursprünglichen 256K je nach Lage von Common (0-16K oder 56-64K) diese Bereiche in den oberen 3 Banks nicht verfügbar waren. In den bei der 1MByte-Erweiterung dazugekommenen 3 x 256K war durch Bit 7 in Port F3H steuerbar, ob bei logischer Adressierung innerhalb des als Common festgelegten Adreßbereichs ein Zugriff auf die absolute 64K-Bank 0 oder auf die relative Bank 0 innerhalb des durch D0 und D1 in Port F3H eingestellten 256K-Blocks zugegriffen wird. Die Unterschiede der beiden Banking-Speicheraufteilungen sind folgende:

Abb. 14: Speichereinteilung beim Banking mit Z80 über D0, D1 von Port F3H und HD64180-MMU



Das HD-Banking ist der in Hardware vorgegebenen Common/Bank-Einteilung der logischen 64K des GIIIs anzupassen.

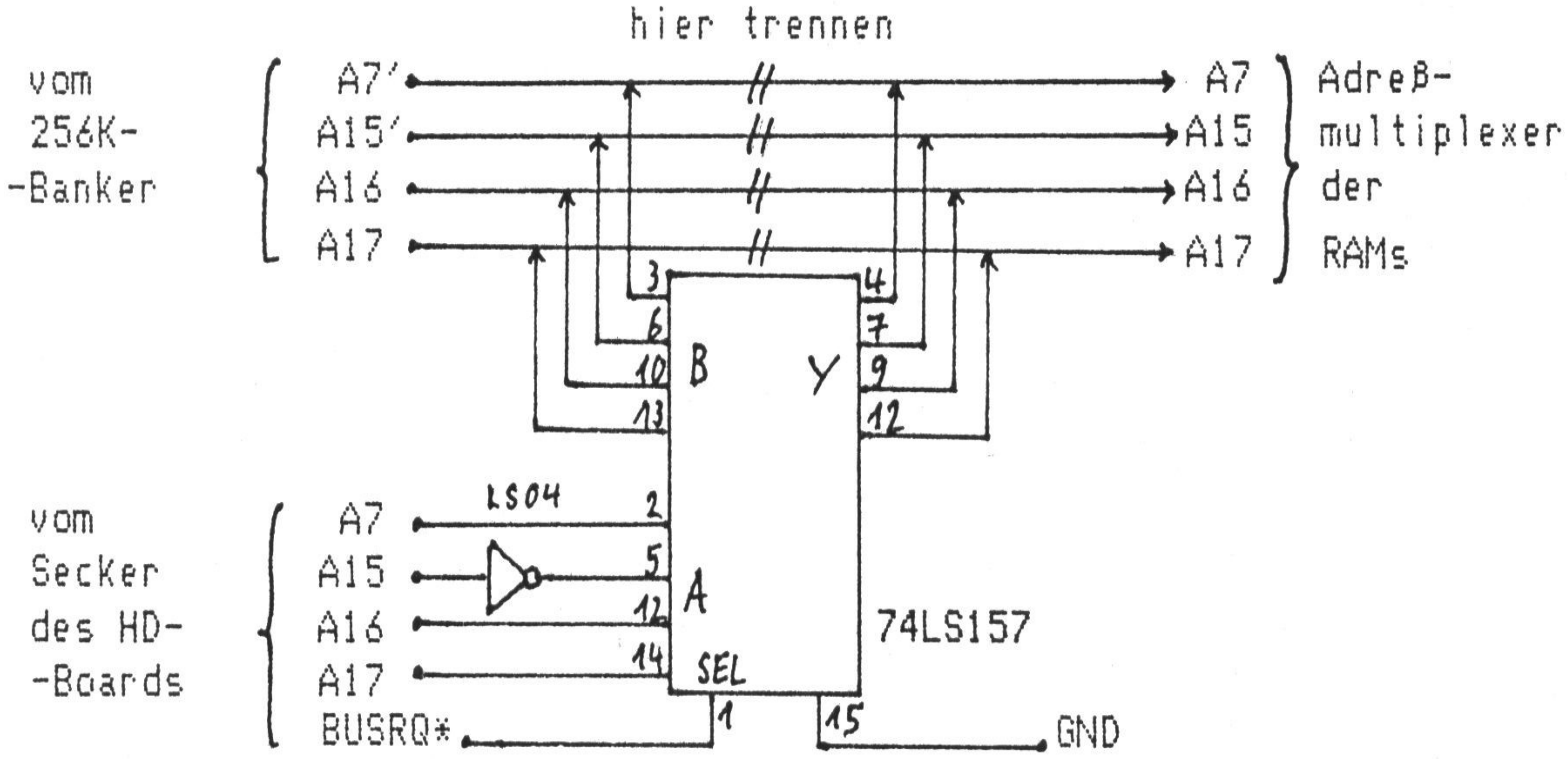
Die MMU des HD ist dafür folgendermaßen zu programmieren:

```
1) Common bei 0-16K:
CBAR : 44H      Common 0 logisch von 0000H bis 3FFFH
                Common 1 logisch von 4000H bis FFFFH
                Bank Area ist nicht definiert
BBR  : 00H      hat keine Bedeutung
CBR  : 00H, 10H, 20H, 30H
                für Banks 0-3 analog zur Speichereineinteilung
                beim Banking mit D0 und D1 in Port F3H des Z80
2) Common bei 56-64K:
CBAR : E0H      Common 0 ist nicht definiert
                Common 1 logisch von E000H bis FFFFH
                Bank Area logisch von 0000H bis DFFFH
CBR : 00H       hat keine Bedeutung
BBR : 00H, 10H, 20H, 30H
                für Banks 0-3 analog zur Speichereinteilung
                beim Banking mit D0 und D1 in Port F3H des Z80
```

2) TRS80, Model I, GENIE I und II und Komtek 1 mit 256K (1M)-Banker gemäß Infos der TRS80/GENIE-User-Clubs, 1986

Zum Umschalten zwischen den Adressen A7, A16 und A17 des Bankers unter Z80-Kontrolle oder der MMU des HD64180, wenn dieser den Computer kontrolliert, ist ein 74LS157-IC nötig, das zwischen Banker und RAM-Adreßmultiplexer geschaltet wird. Dieses IC wird folgendermaßen beschaltet:

Abb. 15: Multiplex der höheren Adressen des HD64180 und des 256K-Bankers



Durch Huckepacklöten auf ein anderes 74LS157-IC mit den Pins 8 und 16 entweder auf dem CPU-Board oder auf dem Banker ist auch die Versorgungsspannung für dieses IC vorhanden. Um unter 280- und HD-Kontrolle beim Banking gleiche physikalische Speicher anzusprechen, sollte A15 des HD-Boards über einen 74LS04-Inverter angeschlossen werden, wenn dies zum Banking der oberen 32K für den Banker des Z80 auch gemacht wurde.

Die Decoderschaltungen für die Freigabe von RAM und ROM sowie memory-mapped I/O im TRS80, GENIE und Komtek 1 berücksichtigen nur die Adressen bis A15. Da ROM und memory-

mapped I/O aber unter HD-Kontrolle nur dann freigegeben werden sollen, wenn auch A16, A17 und A18 low sind, müssen diese Signale in der Decoderschaltung noch berücksichtigt werden.

Dafür werden die Signale A16 und A17 von den Ausgängen des zusätzlichen 74LS157 und das aus A18 und BUSRQ* hergeleitete MMEN* verwendet.

#1 die direkte Verbindung zu Z37, Pin15 wird wieder entfernt Eingriff in die Decoderschaltung bei GENIE I und II

Abb. 16: Eingriff in die Decoderschaltung bei Genie I und II

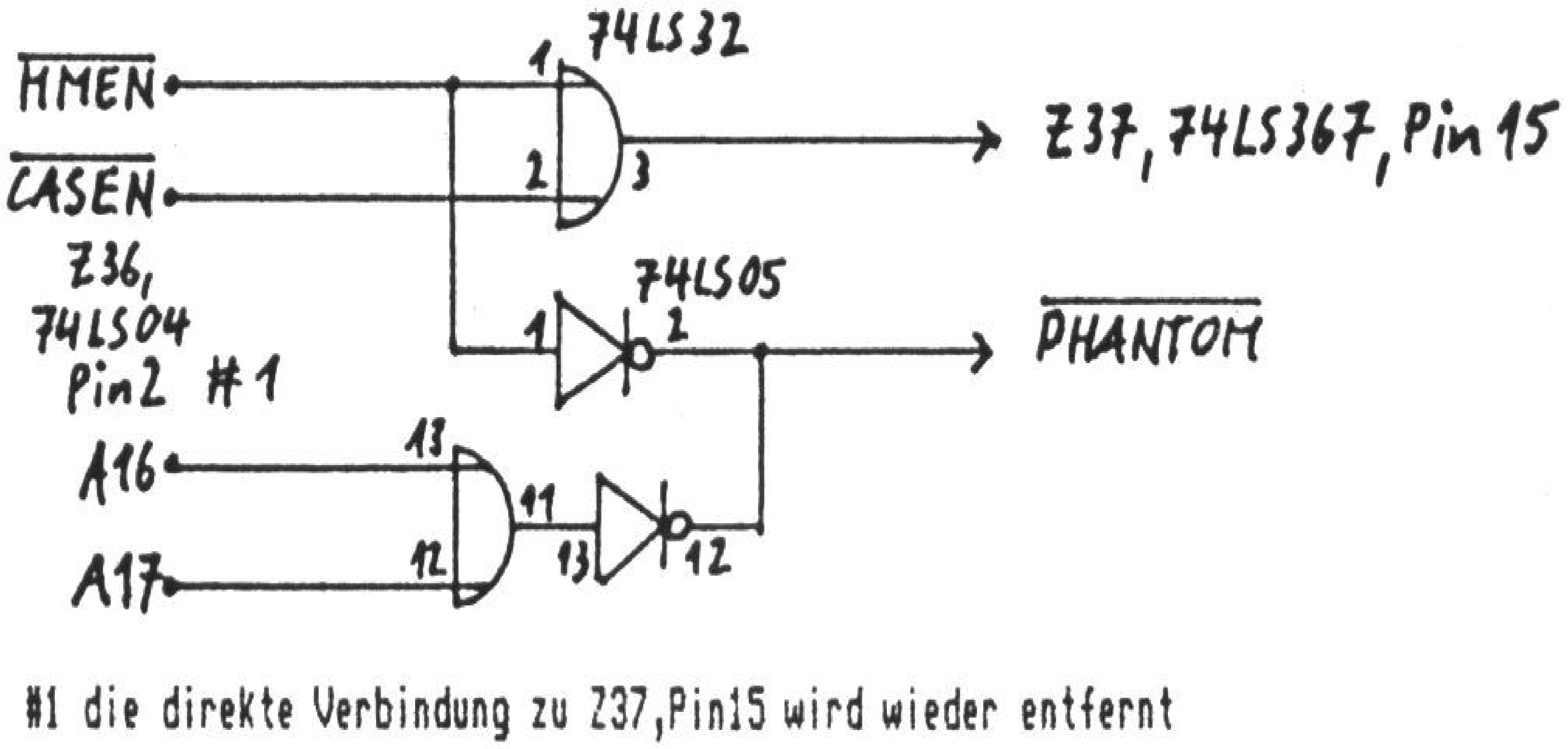
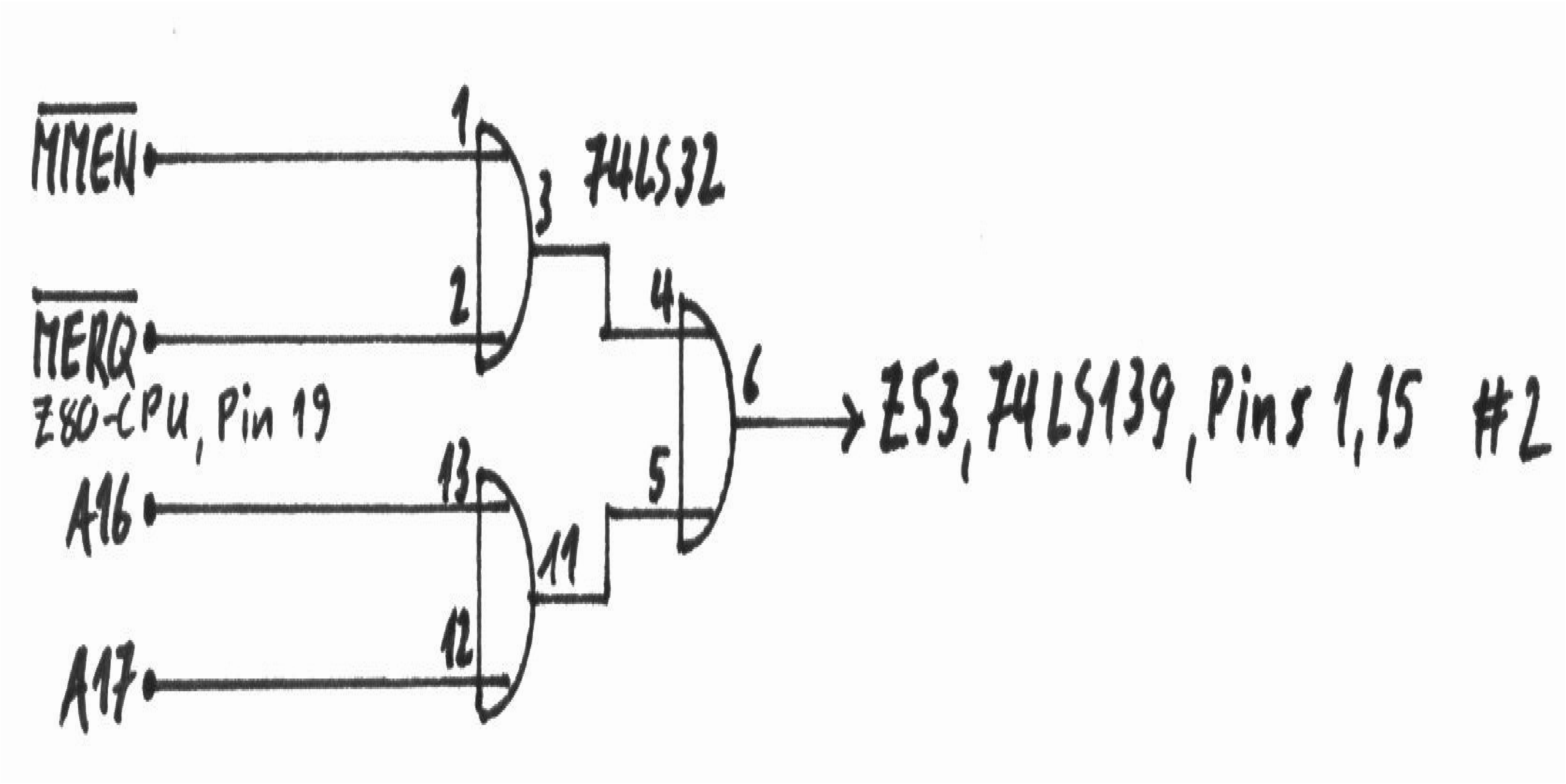
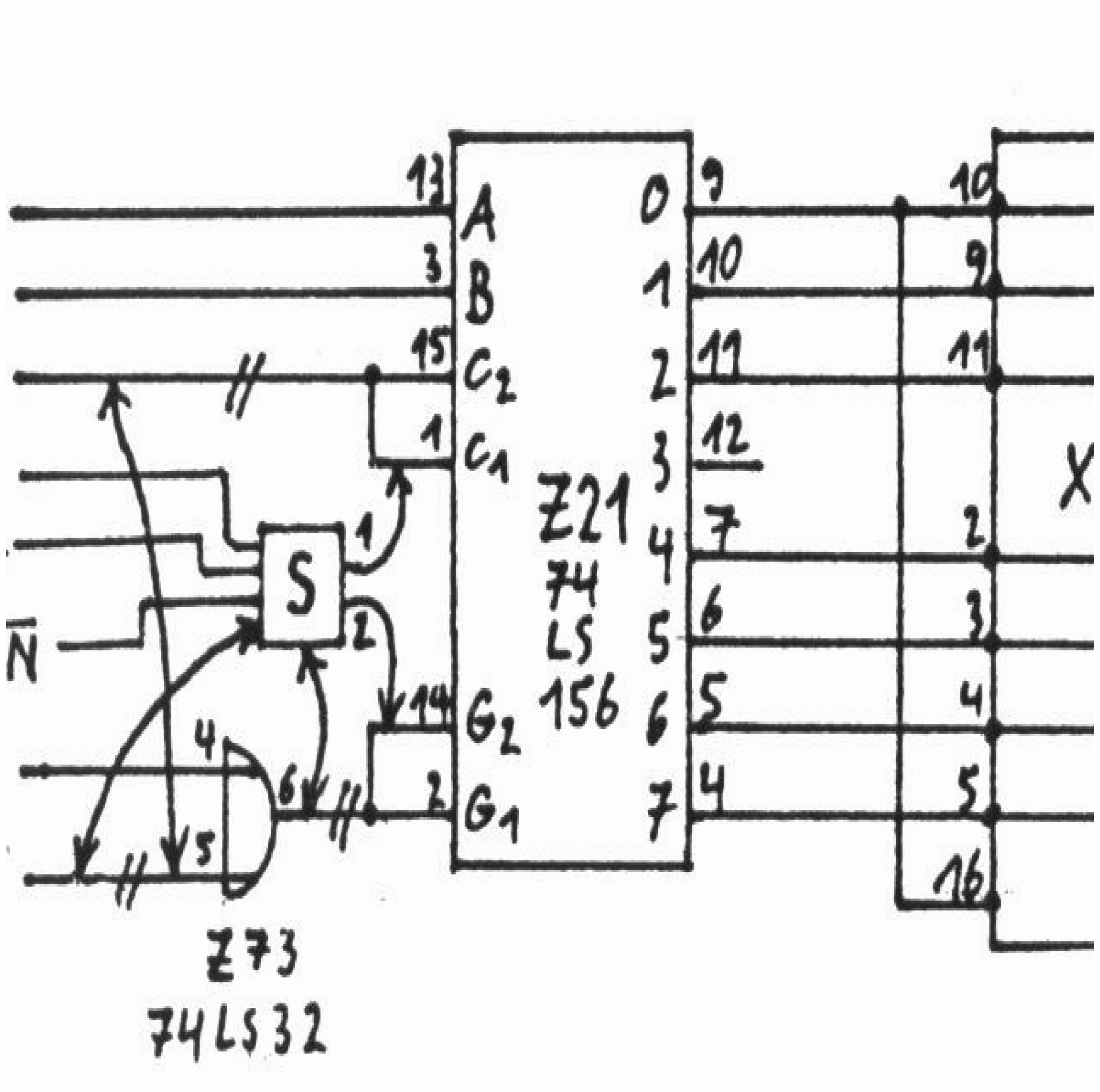


Abb. 17: Eingriff in die Decoderschaltung beim Komtek 1
#2 das bisher an diese Pins gelangende CAS wird abgetrennt*



Im TRS80 ist eine etwas aufwendigere Methode nötig. Zum Teil sind die in der folgenden Abbildung dargestellten Änderungen schon beim Umrüsten auf 64 bzw. 256K RAM auf dem CPU-Board durchgeführt worden.

Abb. 18: Änderungen am Adreßmultiplexer des TRS80
***S** links im Bild ist die hierauf folgende Schaltung*

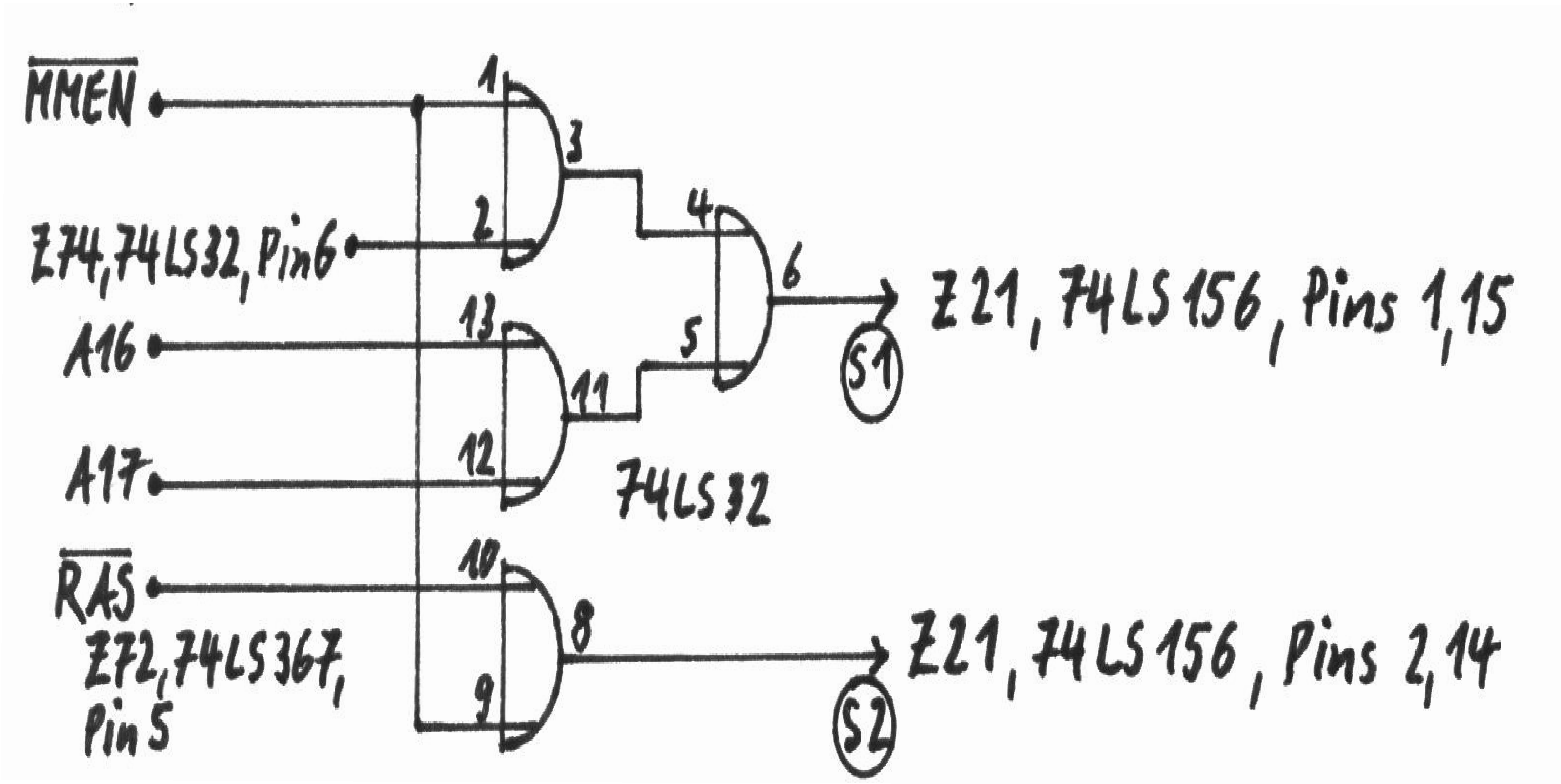


Anstelle von A14 wird der Ausgang S1 an die Pins 1 und 15 von Z21 gelegt.

Anstelle von Pin 6 von Z73 wird der Ausgang S2 an die Pins 2 und 14 von Z21 gelegt.

Anstelle von RAS* wird A14 an Pin 5 von Z73 gelegt.

Im Jumperfeld X3 werden alle Brücken außer 1-16 gelegt. Zusätzlich wird mit einem an den Pins 7 und 14 an geeigneter Stelle huckepack-gelöteten 74LS32 folgende Zusatzschaltung realisiert. ***Abb. 19: Zusatzschaltung S für den Adreßdecoder***

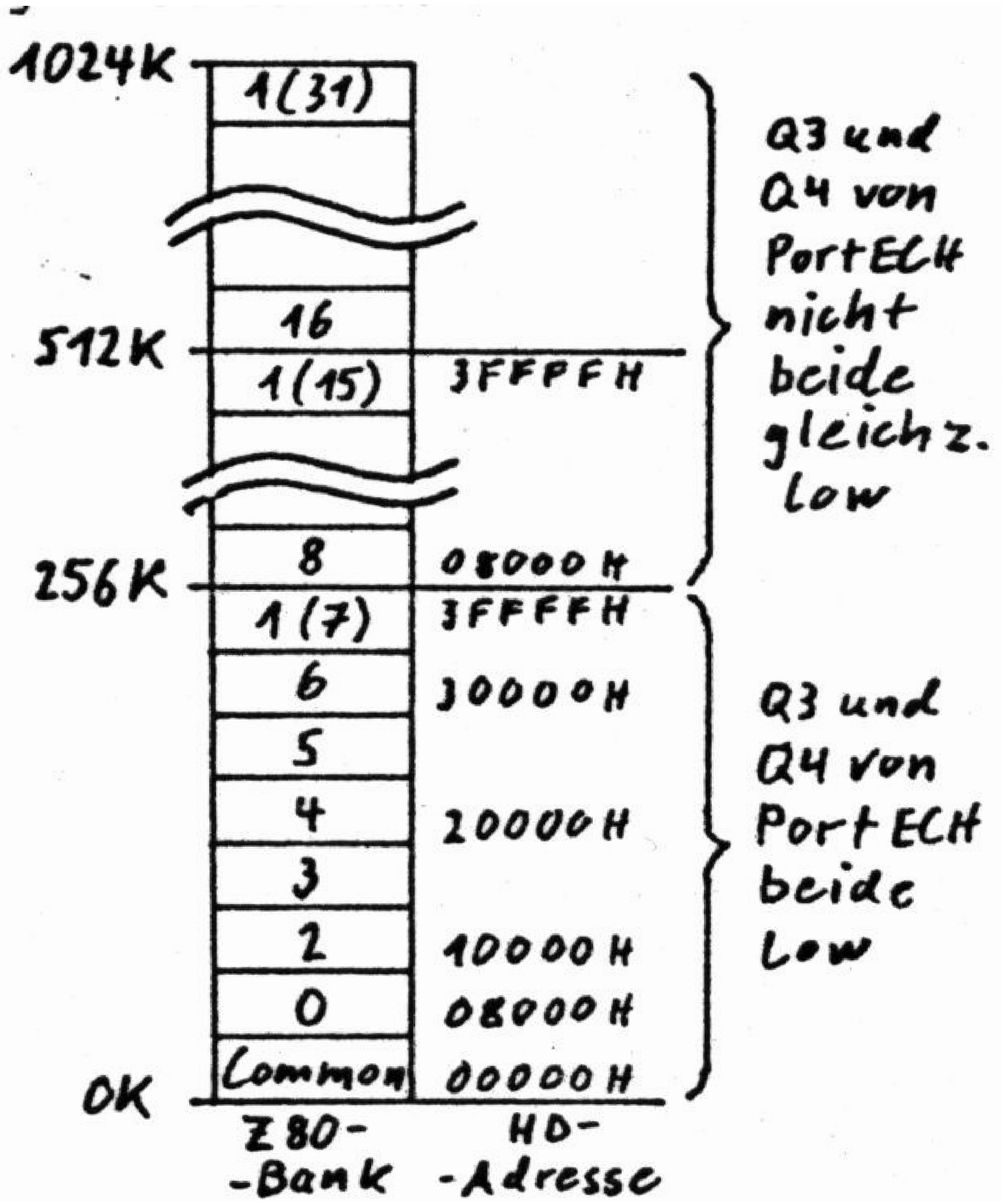


Da der HD im Computer nur 256K adressieren kann, muß ein Zugriff auf darüber hinausgehende Speichergrößen unter Zuhilfenahme des Bankers erfolgen, wobei dan die Bits 0, 1 des Ports ECh, mit denen per Banking innerhalb von 256K adressiert wird, wirkungslos sind, weil die Adressen A15, A16 und A17 des HD dann durchgeschaltet werden. Es muß dann das Banking des HD dem des Bankers angepaßt werden: Common 0 bei 00000H-07FFFH und Common 1 bei 08000H-0FFFFA (für Bank 0). Banking mit der MMU des HD muß dann durch Verschieben von Common 1 um jeweils 32K erfolgen.

D0, D1 und D2 in Port ECA haben für den HD64180 keine Bedeutung; wenn aber später der Z80 wieder den Computer übergeben kriegt, werden diese Bits des Port ECH aktiv. Das kann dann unangenehme Folgen haben, wenn bei mehr als 256K im Computer der HD mit D3 (und D4) in Port ECA auf den Gesamtspeicher zugreift und dabei auch D0, D1 und D2 verändert. Da Port ECH nicht ausgelesen werden kann, muß eine Vereinbarung zwischen den CPUs getroffen werden, daß z.B. bei Übergabe des Computers immer Bank 0 eingestellt sein muß.

Für die Übertragung bisheriger Banking-Software auf den Betrieb der MMU des HD gilt folgende Speichereinteilung und Entsprechung von Banks. Das ist zu berücksichtigen, wenn HD und Z80 abwechselnd auf Daten und Programme in den Banks zugreifen sollen.

Abb. 20: Speicheraufteilung beim Banking mit 256K(1M)-Banker



Die Schaltung des Bankers gibt vor, daß die Bank 1 immer mit der höchsten verfügbaren Bank identisch ist

256K RAM : Bank1 = Bank7
512K RAM : Bank1 = Bank15
1024K RAM : Bank1 = Bank31

Diese jeweils höchste Bank sollte also nicht benutzt werden, um Konflikte mit Bank 1 zu vermeiden.

Speicheraufteilung beim Banking mit 256K(1M)-Banker

Für den HD besteht der Computer nur aus 256K Speicher, innerhalb derer er durch seine MMU oder per DMA adressieren kann. Bei logischen Adressen unterhalb von 8000H wird immer der erste 32K Speicherblock bzw. ROM oder Memory-mapped I/O adressier.Bei logischen Adressen ab 8000H wird die durch die MMU (Common-1-Lage)vorgegebene Bank innerhalb des vom Z80 durch Q3 und Q4 von Port ECh adressierten 256K-Blocks angewählt.

Eine entsprechend Bank 1 ses Bankers mögliche Adreßverschiebung für die MMU des HD findet nicht statt. Bei logischen Adressen oberhalb 7FFFh adreßieren Q3 und Q4 des Port FCh und die physikalische Lage von Common 1 direkt den Speicher.

Der Systemtakt läßt sich ähnlich, wie für den Genie IIIs gezeigt, umschalten, wobei der bisherige Systewmtakt (Punkt A in der bei der Beschreibung für den GIIIs gezeigten Schaltung) an den Eingang des Treibers gelegt wird und der Ausgang anstelle des bisherigen Taktes hier direkt an Pin 6 des Z80 geführt wird. Der Punkt A ist zu finden bei:

TRS80, M1	Z72, 74LS367, Pin 11
Genie I, II	Z37, 74LS367, Pin 3
Komtek 1	Z18, 74LS74, Pin 9

Eventuell muß an Pin 6 des Z80 noch ein Pullup-Widerstand von 4K7 gelegt werden, wenn ein solcher nach dem Abtrennen des direkten Systemtaktes nicht mehr Verbindung mit Pin 6 des Z80 hat.

Die MMU des HD ist folgendermaßen zu programmieren:

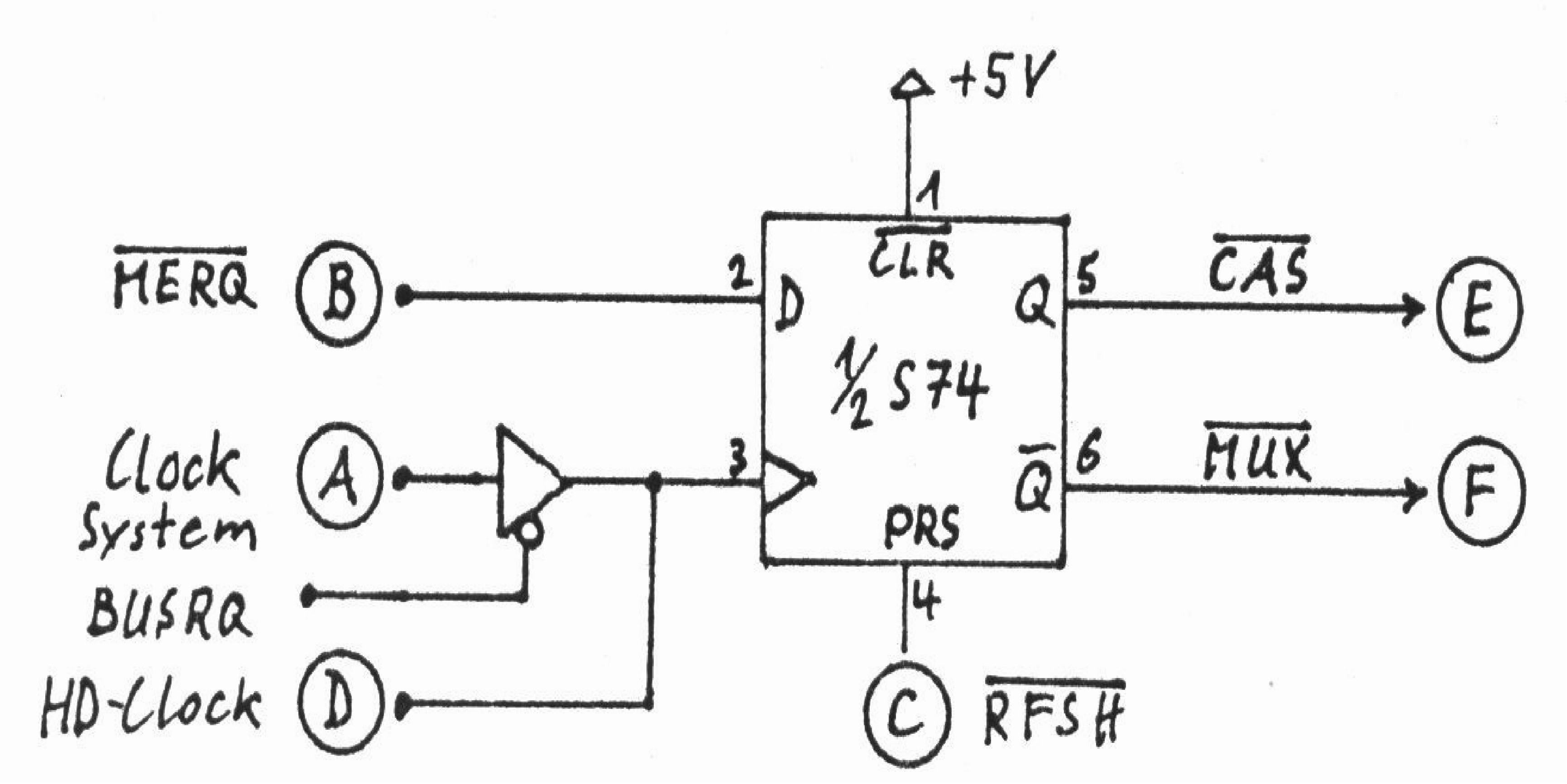
```
CBAR : 88H      Common 0 logisch bei 0000H-7FFFH
                Common 1 logisch bei 8000H-FFFFH
                Bank Area nicht definiert
BBR : 00H       hat keine Bedeutung
```

```
CBR : 00H, 08H, 10H, 18H, 20H, 28H, 30H
                entsprechend den Banks (für Port ECH)
                0  2  3  4  5  6  1(7,15,31)
```

Zur Anwahl der Banks 8-14 bei 512K bzw. 8-30 bei 1024K sind diese Einträge in das CBR-Register in Kombination mit den Bits (3 und 4) des Ports ECH zu wählen. Zur Anwahl der Bank 1 sollten diese Bits immer high gesetzt werden.

Da bei TRS80 Model 1, GENIE I und II und Komtek 1 die RAM-Steuersignale nicht aus dem Systemtakt selbst sondern aus einem mehrfachen des Systemtaktes hergeleitet werden, muß diese Schaltung durch eine solche ersetzt werden, die den Systemtakt selbst zur Steuerung benutzt. Es muß dabei der an den Z80, Pin 6 gelangende Takt genommen werden, damit die RAM-Steuerung abwechselnd mit dem Z80-Takt oder dem HD64180-Takt synchron läuft. Dafür ist ein 74S74-IC und ein Gatter eines 74LS125-ICs nötig, das die Taktumschaltung (zusammen mit den Bustreibern des HD-Boards) erledigt.

Abb. 20: Erzeugen der RAM-Steuersignale MUX* und CAS* für unterschiedliche Systemtakte



Die in dieser Schaltung bezeichneten Punkte A bis F sind mit folgenden Punkten auf dem CPU-Board zu verbinden:

		TRS80 Model 1			GENIE I u. II			Komtek 1		
Punkt	Signal	IC	Typ	Pin	IC	Typ	Pin	IC	Typ	Pin
A	Z80CLK	72	367	11	37	367	3	18	74	9
B	RAS*	72	367	5	16	367	5	11	05	8
C	RFSH*	40	Z80	28	16	367	3	33	Z80	28
D	HDClock	40	280	6#1	5	Z80	6#1	33	Z80	6#1
E	CAS*	72	367	10#2	37	367	14#2	54	32	2#2
F	MUX*	72	367	2#2	37	367	4#2	30	157	1#2

#1 Die direkte Verbindung der bisherigen Taktquelle (Punkt A) mit Pin 6 des Z80 ist zu durchtrennen, auf den Systembus wird das Signal von Punkt D gelegt.
#2 Die bisher zu diesem Punkt führende Leitung wird durchtrennt.

Die Grundintelligenz der HD 64180-Coprozessorkarte

Arnulf Sopp

Der HD 64180 kann seine erweiterten Fähigkeiten dem Computer nur auf Anforderung des Z80 zur Verfügung stellen. Er muß daher per Software in die Lage versetzt werden, eine solche Anforderung zur Kenntnis zu nehmen und zu interpretieren. Auf Helmut Bernhardts HD-Board ist ein EPROM vorgesehen, das dieses Programm enthält. Sein Inhalt ist im Listing 1 wiedergegeben.

Der HD liest nicht nur nach Reset, sondern auch nach einem TRAP-Interrupt an der Stelle logisch 0000h. Darauf wird hier nicht weiter eingegangen, weil Gerald Schröder darüber bereits schrieb. In diesem Falle verzweigt der HD in eine Fehlerbehandlungsroutine, die den unbekannten Befehl imitiert und dann zum alten Programm zurückkehrt. Diese Routine habe ich fast wörtlich von Geralds Programm übernommen, das in der 17. Ausgabe des Clubinfos abgedruckt ist. Deshalb sei hier nur auf den geringen Unterschied zu Geralds Routine hingewiesen: Die Interrupts müssen nicht mit DI maskiert werden, weil sich die Speicherkonfiguration während der Trap-Behandlung nicht ändert. Ein INT dürfte getrost eintreten. In diesem EPROM entfällt natürlich auch das Banking, das für das Genie 2s im Info beschrieben wurde. Ein TRAP kann im EPROM nicht auftreten, weil keine Illegals verwendet sind. Die Routine liegt daher nur zur Bearbeitung aus dem RAM bereit.

Wurde 0000h nach einem Reset gelesen, finden einige Initialisierungen statt. Für den leistungsfähigen HD-Befehl OTIMR (gib ein Datum aus (HL) auf Port (C) aus, erhöhe C und HL, erniedrige B, wiederhole dies, bis B=0 ist) wird HL als Zeiger auf einige Initialisierungsdaten (Label initab) aufgesetzt. Diese Daten konfigurieren im wesentlichen die MMU (Memory Management Unit) des HD, um aus dem Adreßraum von insgesamt 512 kB folgende "Fenster" der CPU zugänglich zu machen:

Common 0:	nicht definiert
Bank-Area:	0.0000-0.EFFFh
Common 1:	7.F000-7.FFFFh

In dieser Tabelle von Daten sind weitere Informationen für den HD enthalten, die aber für das Verständnis des Folgenden nicht von Bedeutung sind. Soweit sie sich nicht aus den Kommentaren des Listings erklären, möchte ich auf das Datenbuch verweisen.

Die CPU hat nun insgesamt 64 kB Speicher verfügbar, deren physikalische Lage oben dargestellt ist. Nach dem Befehl OTIMR wird HL nun auf das letzte Byte einer zweiten Tabelle gestellt (tabend). Sie enthält Daten für den DMAC (Direct Memory Access Controller). Er kann unabhängig von der Konfiguration der MMU Daten aus jedem beliebigen Teil des Gesamtspeichers in jeden beliebigen anderen transferieren. Die Werte vom Label srclog an besagen in diesem Falle, daß die Daten ab dma0 bis zum Label prt in den physikalischen Speicherbereich ab 7.F000h übertragen werden sollen. Die beiden Bytes davor in der Tabelle bestimmen die Art des DMA-Zugriffs und führen schließlich zum Start des DMA.

Die Routine zur Programmierung des DMA-Controllers (ab Label dma0) ist so gestaltet, daß der Programmierer nicht weiter über die einigermaßen komplizierten Port-Ausgaben bescheidwissen muß. Analog zum Befehl LDIR lädt er lediglich HL mit der Quelladresse, DE mit der Zieladresse (beide innerhalb je eines beliebigen physikalischen 64-kB-Blocks) und BC als Bytezähler. Zusätzlich muß nur noch im Akku angegeben werden, in welchem physikalischen 64-kB-Block des Gesamtspeichers von 512 kB Quelle und Ziel zu suchen sind. Die Bits 0-2 tragen diese Information für die Quelle, Bit 4-6 für das Ziel. Ich habe diese Bits bewußt ausgewählt, weil dann der Akku Quelle und Ziel sozusagen im Klartext, nämlich in je einer Hex-Ziffer enthält. Je nach dem, welche Vorarbeit bei der Programmierung des DMAC schon geleistet ist, kann auch an späterer Stelle eingesprungen werden.

Der EPROM-Inhalt von srclog bis prt-1 ist nun ins obere RAM übertragen. Er enthält auch eine Interrupt-Vektortabelle für den Fall des IM 2. Der Z80 kann nämlich mit einer geringfügigen Hardware-Änderung mit der Ansprache des Ports FBh einen Interrupt am HD-Pin INT2 auslösen, wodurch der HD die Buskontrolle übernimmt. Ebenso ist es denkbar, daß der HD im Interrupt-Betrieb für den Z80 mit irgendwelcher Peripherie kommuniziert. Das EPROM ist auf diese Fälle vorbereitet. Da die Vektoren von Fall zu Fall programmiert werden müssen, sind sie im EPROM alle zu FFFFh gesetzt, so daß sie entweder umgebrannt oder später soft aufgesetzt werden können.

Nur der Vektor für einen Interrupt auf den Pin INT2 des HD ist auf intz80 gesetzt, weil er anstelle des Polling die Busübergabe signalisieren könnte. Die geschähe nun freilich asynchron. Ein Job könnte gerade in Arbeit sein. Daher gibt es das Flag rdyflag, aus dem der HD ablesen kann, ob er die neue Aufgabe bereits übernehmen kann. Wenn nicht, fährt er erst mit dem alten Job fort. Wenn er später wieder bei waitz80 landet und pollt, findet er sofort RAM und tut nun verspätet, was von ihm verlangt wird. Im ausschließlichen Polling-Betrieb merkt der HD nichts von der Z80-Ansprache bevor er nach waitz80 zurückkehrt.

Schließlich wird noch der gesamte Inhalt des EPROMs nach 6.0000-7.FFFF übertragen, damit der HD später auch vom Z80-RAM aus bequemen Zugriff auf Routinen hat, die im EPROM niedergelegt sind. Ohne diese Kopie ginge das nicht, weil das Z80-RAM und das EPROM niemals gleichzeitig erreichbar sind. Solche Routinen können nachträglich ins EPROM gebrannt werden. Ihre Funktion muß nichts mit dem HD zu tun haben.

Da ab physikalisch 7.F000h bereits Common 1 definiert ist, sind die per DMA nach oben geladenen Programmteile sofort verfügbar. Deshalb können bereits die Wertetabelle und das DMA-Unterprogramm im oberen RAM benutzt werden, um das EPROM nach oben zu kopieren.

Damit hat sich das EPROM nun selber überflüssig gemacht. Dennoch bleibt es eingeblendet. Der HD liest nun fortgesetzt an der Speicherstelle 0000h und prüft, ob dort der Opcode 00h für

NOP steht. Wenn ja, handelt es sich an dieser Adresse immer noch um sein EPROM. Wenn ein beliebiger anderer Code gelesen wird, liegt dort unten nun der bisherige Z80-Speicher vor. Dies ist das Signal, daß der Z80 soeben die Kontrolle über seinen Systembus an den HD abgab. Kein Betriebssystem wird bei 0000h ausgerechnet mit einem NOP beginnen, so daß Zweifel über EPROM oder Z80-Memory nicht bestehen können. Und wenn doch, sollte der User ohnehin zuvor für den HD einen JP (Opcode C3h) zu einer Trap-Routine dort gepatcht haben. Auf Computer, die in ihrem unteren Adreßraum ein nicht schreibbares ROM haben, wird an späterer Stelle eingegangen.

Die Adresse 0000h für das Polling resultiert aus der Forderung, daß das Board mit seinem EPROM möglichst in jedem Z80-Rechner ohne Änderungen benutzbar sein sollte. In den TRS-80-kompatiblen Computern mit NEWDOS o. ä. gäbe es viele Möglichkeiten, auf eine andere Adresse auszuweichen. Jedoch schon unter CP/M fangen die Restriktionen an. Ab 0000h beansprucht CP/M den Speicher für sich. Dort steht ein JP-Befehl für den Warmstart mit CTRL-C. Unter CP/M muß deshalb eine Trap-Routine prüfen, ob es sich tatsächlich um einen TRAP handelt und gegf. den überschriebenen CP/M-Sprung nachholen. Dies gilt für viele Computer analog. Nur solche, die NEWDOS/G-DOS/TRSDOS fahren und ein Boot-EPROM besitzen, können sicher sein, daß 0000h nur bei einem TRAP angesprungen wird.

Die beiden Mikroprozessoren müssen einander mitteilen können, was anliegt. Eine Möglichkeit bestünde darin, daß der HD bei einer Ansprache durch den Z80 immer an dieselbe vereinbarte Speicherstelle springt und einfach tut, was da steht. Für diejenigen unter uns, die auch noch im Tiefschlaf vom Hacken träumen, wäre dies zweifellos die flexibelste Lösung. Ein bequemerer Weg, der sogar unter BASIC beschritten werden kann, wurde hier gewählt: An der Stelle 0003h für TRS-80 usw.: kommt später) wird der Stackpointer abgelegt. Alles Weitere, was zur Kommunikation der beiden Mikroprozessoren erforderlich ist, muß nun über den Stack abgewickelt werden. Der Stackpoin-ter des HD wird dazu aus 0003/4h geladen.

Hier gibt es folgende Vereinbarung: Zuunterst im Stack kann AF gePOPt werden. Im Akku steht dann ein Signalbyte, das dem HD verrät, welche von fünf möglichen Anforderungen der Z80 an ihn stellt (was in welcher Reihenfolge dazu auf den Stack gelegt werden muß, ist in den Listings reichlich kommentiert und teilweise tabellarisch zusammengefaßt):

1. Die endgültige Übergabe des Systembus an den HD. Wenn der Akku =00h ist, will sich der Z80 endgültig verabschieden. Der HD übernimmt nun seine Arbeit bis zum nächsten Reset oder Ausschalten. In diesem Falle wird in der nächsthöheren Ebene auf dem Stack eine Adresse erwartet, wo der HD ein paar Bytes zum Umbanken des Common- 1 -Bereiches ablegen kann. Bankt er nämlich noch oben im alten Common 1 dieses einfach weg, findet er an der entsprechenden Adresse im Bereich 0.xxxxh sonst etwas, bloß nicht viel Gescheites. Eine Ebene höher im Stack steht schließlich die Adresse, an der der HD seine Arbeit beginnen kann. Das wäre für NEWDOS/G-DOS z. B. im einfachsten Fall 402Dh, der DOS-Ready-Einsprung. Falls zusätzlich Registerinhalte aus dem Stack übergeben werden sollen, beginnt das Programm für den HD eben mit ein paar POPs.
2. Erledigung eines vorübergehenden Jobs für den Z80. Irgendein Akku-Inhalt zwischen 03h und FEh signalisiert dem HD, daß jetzt eine Einzelaufgabe für den Z80 übernommen werden soll, wonach der HD auf eine erneute Ansprache des Z80 zu warten hat. Nun erwartet der HD in der nächsten StackEbene die Adresse, wo er ins Job-Programm einspringen soll. Um nicht auch noch Informationen über die Anfangs- und Endadresse des zu bearbeitenden Programms übergeben zu müssen, wird kurzerhand der ganze Speicherinhalt von 0000-EFFFh per DMA nach oben geholt. Die Bank-Area, bisher bei 0.0000- 0.EFFFh, wird nach 7.0000-7.EFFFh verlegt. Das Z80-Programm wird nun als nterprogramm behandelt (es muß mit RET enden). Wenn es abgearbeitet ist, kehrt der HD wieder in die Warteschleife waitz80 zurück. Es besteht die Vereinbarung, daß ein solches Programm Rechenergebnisse nur im Bereich ab F800h ablegen darf, und zwar höchstens 2 kB (bis logisch FFFFh). Weiterhin ist klar, daß es ab F000h, wo der HD waltet, nichts mehr verloren hat. Es darf aber nahezu beliebig herumbanken, denn an der Rückkehradresse waitz80 werden die lebenswichtigen Systemkonfigurationen restauriert.
3. Die Abholung von Rechenergebnissen. Wenn ein solcher vorübergehender Job Ergebnisse ab F800h niedergelegt hat, könnte ein weiterer Job so programmiert sein, daß er sie von dort per DMA in den unteren Speicher transferiert. Das wäre für diese simple Aufgabe aber zu aufwendig. Deshalb gibt es die dritte Möglichkeit, daß der Akku genau =01h ist. In diesem Falle "weiß" der HD ohne weitere programmierte Absprache, daß er jetzt 2 kB Daten ab 7.F800 in einen Speicherbereich übertragen soll, der ihm eine Ebene höher im Stack mitgeteilt wird.
   Größere Datenmengen, etwa eine Graphik von 32 kB Umfang oder sehr lange Tabellen, können gleichwohl vom HD errechnet werden. Sie müssen allerdings dann wirklich in einem gesonderten Job-Programm abgeholt werden. Ihr Puffer muß weit genug unterhalb logisch F000h oder jedenfalls bei entsprechender Umsicht unterhalb physikalisch 7.F000h beginnen.
4. Die Kopie des Speichers 0.0000-0.FFFF nach 1.0000-3.FFFF. Das EPROM bietet die Möglichkeit, das untere Viertel der 41256-RAMs in die drei oberen Viertel zu kopieren. Zu diesem Zweck wird der Akku mit FFh geladen und auf den Stack gelegt. Weitere Informationen für den HD sind nicht erforderlich. Nun liegen vier identische Speicherbereiche vor, zwischen denen hin- und hergeschaltet werden kann, weil in der Nachbarbank sofort die passende Fortsetzung gefunden wird. Lediglich dem Stack ist dabei Aufmerksamkeit zu widmen, falls er nicht in einem Common-Bereich liegt und für alle Speicherviertel gültig ist. Das Unterprogramm clone kann von überall her aufgerufen werden, um beispielsweise das Konzept von vier Computern in einem Gehäuse mit der DMA-Fähigkeit des HD komfortabler zu verwirklichen.
5. Daß der Stack vom HD mit POPs und RETs gelesen wird, ist für den Z80 völlig belanglos. Sein Stackpointer geht bei der Busübergabe mit der ganzen CPU in den Ruhestand. Wenn der HD die Buskontrolle zurückgibt, zeigt der SP des Z80 an dieselbe Stelle wie vor dem OUT (FBh),A. Der User muß lediglich daran denken, den vorausgegangenen PUSHes nun die entsprechenden POPs folgen zu lassen, damit z. B. eine RET-Adresse wiedergefunden werden kann. Daß der Stack nicht ausgerechnet im Bereich ab F000h liegen darf, versteht sich. Dort hat der HD sein privates RAM, aus dem er höchstens POPpen könnte, was er selber gePUSHt hätte. Unter BASIC muß daher beispielsweise das Himem begrenzt werden.

   Bei der endgültigen Übergabe der Buskontrolle an den HD ist es so, als wäre im Rechner nie ein Z80 eingebaut gewesen. Man braucht sich um ihn nicht mehr zu kümmern. Allerdings darf der HD dann auch niemals einen Output auf den Port FBh geben, weil er damit die Kontrolle an den Z80 zurückgäbe. Prinzipiell ist das natürlich auch möglich, erfordert aber besondere Umsicht. Da der Z80 genau da weiterarbeiten würde, wo er selber mit einem OUT (FBh),A die Kontrolle abgab, sollte man an der Stelle dahinter z. B. einen Jump nach DOS-Ready oder ähnlich unkritischen Code unterbringen. Es muß auch beachtet werden, daß sich der HD zuvor wieder Common 1 in den oberen Speicher legt, damit er sich dort in sein Schneckenhaus zurückziehen kann.

   In dem unter 2. genannten Fall muß der Z80 selbstredend ein Programm übergeben, das sozusagen mit dem Gehirn des HD denkt. Wenn Common 0, die Bank-Area oder Common 1 dabei manipuliert wird, kann einiges schiefgehen. Der Programmierer kann mit der erforderlichen Vorsicht gleichwohl ohne weiteres über den EPROM-Inhalt und den gesamten Speicher ab 4.0000h verfügen sowie über die Annehmlichkeiten des DMAC, der seriellen Schnittstellen des HD usw. Es ist sogar möglich, nach diesem Muster einen BASIC-Job zu übergeben (wer wollte ernsthaft eine Wurzel in Maschinensprache ziehen?). Da der gesamte Speicher von 0.0005-0.EFFF nach oben verholt wird, darf selbstverständlich auch BASIC/CMD oder auch nur Level 2 dort am Werk sein.

Es dürfen auch Illegals im Programm enthalten sein, denn Geralds Trap-Routine liegt bereit. Dieser Fall ist jedoch nicht vorgesehen. Dazu müßte an der Stelle logisch 0000h ein JP zur Trap-Routine gesetzt werden. Es wäre aber nicht besonders klug, weil die Umschreibung eines Illegals mit zwei, drei Befehlen um Größenordnungen schneller geht als ein Trap.

Im Falle der Übergabe von Rechenergebnissen muß von der Z80-Seite aus nichts weiter bedacht werden. Es versteht sich lediglich, daß von der auf dem Stack abgelegten Adresse an wirklich mindestens 2 kB RAM frei sein müssen. In beiden letztgenannten Fällen knipst der HD im unteren Adreßraum mit OUT (FBh),A sein EPROM wieder an, sobald er das zu bearbeitende Programm nach oben oder die Resultate eines Programms nach unten verfrachtet hat. Der Z80 hat im unteren RAM nun wieder das Sagen.

In allen Fällen muß der Programmierer wissen, daß der HD noch längst nicht auf die Hardware eingestellt ist, in der er die Arbeit übernehmen soll. Bei den Tandy-Kompatiblen ist z. B. der Befehl für den Interrupt-Modus IM 1 ins HD-Programm zu schreiben. Da bei der Busübernahme DI durchgeführt wurde, müssen die Interrupts auch bei Bedarf gesondert wieder enabled werden. Überhaupt kann der Z80 in keinerlei Hinsicht von sich auf andere schließen (Registerinhalte, INT-Behandlung), so daß vor der Busübergabe auf der Z80-Seite sehr sorgfältig an alles Wesentliche gedacht werden muß.

Um das EPROM im Interesse einer allgemeinen Verwendbarkeit auch nicht auf das Genie 3s mit Helmuts Megabyte-Banker maßzuschneidern, darf der Z80 den HD 64180 nur ansprechen, wenn er gerde im unteren Viertel der 256er RAM-Chips arbeitet. Das ist der "Computer Nr. 0", wie wir ihn in dem Sonderheft "Dein G3s, die 4 unbekannten Wesen" genannt haben. Bei Computern mit 64 kB RAM entfällt diese Bedingung.

Dies sind die Möglichkeiten, die der HD sich mit seinem EPROM selber schafft. Wenn er jedoch endgültig oder vorübergehend die Kontrolle über den Systembus übernimmt, muß ihm der Z80 zuvor Quartier gemacht haben. Wegen der möglicherweise auftretenden TRAP-Interrupts sollte an 0000h in eine Error-Trap-Routine verzweigt werden. Wo die Trap-Routine dann steht, ist gleichgültig. Es muß auch nicht die komplette Routine sein, wie sie bereits im EPROM programmiert ist. Da sie nach dem DMA bei den Initialisierungen nach 7.F100 übertragen wurde, wo sie noch immer steht, braucht der Z80-Beitrag zum TRAP nur aus dem entsprechenden Banking und einem Sprung nach dort oben zu bestehen.

Die älteren TRS-80-kompatiblen Rechner lesen von 0000-2FFFh in einem ROM und haben erst ab 4000h freies RAM. Wenn diese Computer einen Banker eingebaut haben, der im unteren Adreßraum RAM einblenden kann, gilt alles bisher Gesagte unverändert. Im RAM muß dann eine Kopie des ROMs vorliegen, die an 0000h aber den Trap-Sprung enthält. Aber ohne Banker entfällt die Möglichkeit, unter der Regie des HD Programme mit Illegals zu fahren, da wegen der nicht änderbaren RST-00h-Routine ein TRAP nicht möglich ist. Solche Programme würden zu einem Boot führen. Am Anfang des Listings muß dann auch spbuff auf eine RAM-Adresse gesetzt werden, z. B. 4200h (DOS-Sektorpuffer).

Im Folgenden soll nun ein Beispiel gegeben werden, wie der Z80 lästige Pflichten loswerden kann, um sich mittlerweile wichtigeren Dingen zu widmen. Das Beispiel besteht hier nur aus dem Löschen des Bildschirms. Eine läppische Sache, die diese digitale Orgie mit einem zweiten Mikroprozessor gar nicht lohnt. Es geht jedoch nur darum, zu zeigen, wie beide CPUs miteinander kommunizieren können. Das Programm ist im Listing 2 wiedergegeben. Der Z80 übergibt im Stack zuerst die Einsprungsadresse des Programms namens job, dann das Signalbyte, aus dem der HD entnehmen soll, daß es nur eine vorübergehende Aufgabe ist. Schließlich wird der Stackpointer an 0003/4h abgelegt.

Jetzt hat der Z80 die Hände frei, um Denkwürdiges zu leisten, während der HD inzwischen 2 kB Pufferraum mit Leerzeichen vollmalt. Wenn der Z80 der Meinung ist, es sei nun an der Zeit, den Bildschirm wirklich zu löschen, holt er sich die 2 kB Blanks vom HD ab. Das geschieht analog zur obigen Prozedur: Auf den Stack kommt zunächst die Pufferadresse, hier der Bildschirm, dann das Signalbyte 01h, damit der HD seinen Puffer hinunterkopiert. Es wäre wesentlich simpler gegangen. Das Job-Programm hätte selbständig per DMA auf den Bildschirm zugreifen können, um ihn um Größenordnungen schneller als gewohnt zu löschen. Wie der User mit den vielfältigen Möglichkeiten jongliert, die ihm das EPROM offenläßt, bleibt ihm schließlich selbst überlassen.

Als weiteres Beispiel wird gern. Listing 3 die Buskontrolle endgültig an den HD 64180 übergeben. Das Signalbyte am unteren Ende des Stack ist 00h. Darüber liegt die Adresse, wo der HD sein kleines Banking-Programm switch ablegen soll. Darüber schließlich findet der HD den Einsprung in seine künftigen Pflichten. Dort wird zunächst aus ihm sozusagen ein Z80 gemacht, damit er mit G-DOS klarkommt. Anwenderprogramme, die er später bearbeiten wird, können mit dem ganzen Spektrum seiner Leistungsfähigkeit arbeiten.

Es gibt eine fünfte Möglichkeit, wie sich der Z80 des HD bedienen kann. Sie unterscheidet sich von den bisher Genannten so grundlegend, daß sie erst jetzt besprochen werden soll. Nachdem die Methode Stack-orientierter Kommunikation vorn Leser verdaut wurde, ist nun hoffentlich nicht mehr mit Konfusion zu rechnen.

6. Der Vektor-Task. Bisher mußte das Byte (SP+0) als LSB eines gePUSHten 16-Bit-Worts nicht beachtet werden. Es beginnt, eine Rolle zu spielen, wenn der Inhalt von (SP+1) (gePUSHtes MSB) 02h beträgt. Dies ist das Signalbyte, um eine beliebige Routine zu bearbeiten, deren Adresse man nicht einmal kennen muß. Das Register C enthält nach POP BC die laufende Nummer der gewünschten Routine, von denen es 128 geben darf. Wird beispielsweise gewünscht, daß das Programm Nr. 9 (ab 0 gezählt) läuft, dann wird ein 16Bit-Register vor der Busübergabe mit 0209h geladen und gePUSHt. 02h ist das Signalbyte für diese Art der Ansprache des HD durch den Z80.

Wo dieses Programm zu suchen ist, steht in einer Tabelle. Die Zahl im Register C wird zunächst verdoppelt, weil jeder Vektor (Zeiger) eine Zwei-Byte-Adresse enthält. Nachdem das MSB auf 00h gesetzt ist, steht nun in BC der Summand zur Basisadresse einer Vektortabelle. BC plus Basis ist dann der Zeiger auf die Adresse, an der das gewünschte Programm steht. Das ist dasselbe Prinzip, das bei Interrupts des Modus IM 2 angewendet wird. Ein solches Programm ist in seiner Flexibilität etwa der endgültigen Busübergabe vergleichbar, kehrt aber immer nach waitz80 zurück. Listing 7 zeigt ein Anwendungsbeispiel.

Mit Listing 4 soll gezeigt werden, wie die Trap-Routine des EPROMs vom Z80-RAM aus aufgerufen werden kann. Das Programm wurde mit dem Genie 3s erstellt, dürfte aber wohl auf allen TRS-80-kompatiblen Computern laufen, die kein ROM an den unteren Adressen haben. Andernfalls muß nach einem anderen Speicherabschnitt für die Routine trap Ausschau gehalten werden.

Er darf auf keinen Fall die logische Adresse F000h überschreiten.

Die Routine liest die MMU-Register BBR und CBAR aus und puffert ihren Inhalt, damit er später wieder restauriert werden kann. Dann wird Common 1 wie bei Reset definiert (ab 7.F000h), damit die Trap-Routine unter der logischen Adresse F100h gefunden wird. Da (z. B. unter BASIC) der Stack im Himem liegen könnte, darf bei der Bankerei nichts gePUSHt werden; alles wird über Speicherladebefehle abgewickelt.

Dieses Programm stellt die Voraussetzung, daß nur im User-RAM ein TRAP auftreten wird. Das DOS, das auch in seiner eigenen Bank werkelt, benutzt keine Illegals, so daß von dort nichts zu befürchten ist. Der Anwender muß sich eben damit abfinden, daß er bei der Arbeit in einer anderen Bank nur die dokumentierten Befehle verwenden darf. Die ziemlich lendenlahme TRAP-Prozedur legt ohnehin nahe, Illegals durch andere Befehle zu ersetzen.

Es fällt auch auf, daß das TRAP-Bit nicht gelesen wird. Das Programm setzt einfach voraus, daß ein RST 00h nur bei Trap auftritt, denn bei Reset wird ja im Boot-EPROM (neuerdings auch der Coprozessorkarte) gelesen. Auch bei ROM-Computern mit nachträglich eingebautem Banker ist das so. Ein Reset-Taster existiert dort nicht, und nach dem Einschalten ist sowieso das ROM selektiert. Der angebliche RESET dieser Maschinen ist in Wirklichkeit ein NMI und springt bei 0066h ein. Dort steht nach wie vor der alte Code.

Aber unter CP/M wäre wegen des RST 00h bei Warmstart mit CTRL-C die Prüfung auf TRAP unumgänglich. CP/M stellt ebenfalls die Bedingung, unbedingt die Interrupts zu maskieren, weil es im Himem arbeitet, wo nach dem Umschalten nun eine andere Bank vorliegt. Die oben beschriebene Akrobatik zum Ermitteln des EI/DI-Zustandes braucht dabei aber nicht stattzufinden. Es geht auch so, wie im Listing 5 beschrieben. Der intern belegte Port 34h enthält nämlich nicht nur die Bits UFO und TRAP (Bits 6 und 7), sondern auch ITEO-2 (Bits 0, 1, 2). Wenn diese Bits 0 lauten, werden Interrupts an den korrespondierenden INT-Eingängen der CPU ignoriert. Der Input von 34h kann also an den Ausgang geladen, dann manipuliert und chließlich am Ausgang wieder restauriert werden.

Der HD 64180 bietet die Möglichkeit, eine programmierbare Anzahl von WAIT-Zyklen für Speicher- und I/O-Operationen getrennt zu generieren. Man möchte natürlich seine Maschine so schnell wie möglich arbeiten lassen, also so wenige WAITs einfügen, daß das System noch eben zuverlässig arbeitet. Im Listing 6 wird gezeigt, wie man diese Grenze herausfinden kann. Ein Job-Programm verändert in der EPROM-Tabelle initab das Byte für den Port 32h. Es bestimmt die Anzahl der WAITs, und zwar für den Speicher in den Bits 6-7 und für Portansprachen in den Bits 4-5. Nach der Bearbeitung des Jobs wird die Buskontrolle endgültig an den HD 64180 übergeben. Wenn das System mit diesem Programm abstürzt oder eine I/O-Operation nicht das erwartete Ergebnis bringt, war der Akku zu sparsam beladen.

Durch wiederholten Aufruf dieses Programms mit variiertem Wert für den Akku wird experimentell bestimmt, wie schnell höchstens mit der eigenen Hardware gearbeitet werden kann. Der so gefundene Wert sollte dann immer bei der ersten Ansprache des HD nach einem Reset gern. dem Muster in Listing 6 an den Anfang der Tabelle initab gesetzt werden. Mit korrekt geladenem Akku kann dies auch als Programm zur endgültigen Busübergabe genommen werden. Abgesehen vom Signalbyte für den HD und den Adressen, die im Stack übergeben werden, braucht dem Stack dabei keine weitere Aufmerksamkeit gewidmet zu werden, weil der Stackpointer beim Sprung nach DOS-Ready neu geladen wird.

Alle sieben Listings enthalten bereits die Mnemonics einiger zusätzlicher Befehle, die der HD 64180 kennt. Da ZEUS sie jedoch noch nicht kennt, stehen sie wie Kommentare hinter einem Semikolon. Um ZEUS begreiflich zu machen, was auf die Diskette assembliert werden soll, sind die Hex-Entsprechungen dieser Befehle darunter immer als DBs programmiert.

Kein Programm wird jemals fertig. Der Inhalt dieses EPROMs wird sich noch ändern, besonders in den ersten Monaten der Erprobung in der Praxis. Das Listing 1 ist deshalb vielleicht (oder eher wahrscheinlich) zum Zeitpunkt seiner Veröffentlichung bereits überholt. Wer das EPROM für seine eigene Coprozessorkarte haben möchte, erhält die jeweils neueste Version. Ein aktualisiertes Listing wird nicht mitgeliefert. Sein Inhalt kann mit einem Job-Programm in den unteren Speicher kopiert und dort leicht disassembliert werden.

Von den möglichen 64 kB des EPROMs ist mit bisher gerade einem Kilobyte nur ein Bruchteil für die Grundintelligenz der Coprozessorkarte verbraucht. Viele nützliche Routinen, z. B. ein Spooler mit enormem Pufferplatz oder eine intelligente RAM-Floppy, finden hier noch Platz. Zu diesem Zweck sind bestimmte Adressen mit FFh-Bytes freigehalten worden. FFh kann bei einem bereits gebrannten EPROM nachgebrannt werden. So sind beispielsweise noch sämtliche Restart-Adressen frei (außer RST 00h, versteht sich). Auch Füllbytes, die nachfolgende Routinen oder Tabellen auf "gerade" Adressen schieben sollen, lauten FFh und können noch sinnvoll programmiert werden, ebenso die Tabelle der Vektor-Utilities. Dasselbe gilt für den NMI-Einsprung an 0066h, obwohl die Karte nicht für einen NMI im EPROM (sehr wohl im RAM) vorgesehen ist. Aber irgendein Hard-Freak unter euch könnte sich das nachrüsten wollen.

```
; Listing 1: EPROM-Inhalt der Coprozessorkarte

  *******************************************************
  Boot-EPROM für die Coprozessorkarte
  mit dem HD 64180

  für das Genie 3s und die meisten anderen Z80-Rechner

  von Helmut Bernhardt

  (C) 1987 Arnulf Sopp

  die Trap-Routine entspricht weitestgehend der Vorlage
  von Gerald Schröder     (C) 1987
;************************************** ***************

;Puffer für den Stackpointer des Z80 bei Übergabe der Buskontr. an den HD
```

```
                ;die Adresse gilt für Computer mit einem Boot-EPROM und RAM ab 0000
0003            spbuff    EQU     0003h           ;für ROM-Computer: auf freie RAM-Adresse ändern

0000                      ORG     0000h           ;physikalisch 0.0000

                ;RST 00: nach Reset die MMU des HD 64180 initialisieren usw.
0000 00                   NOP                     ;wegen des Pollings an logisch 0000
0001 ED5E                 IM      2               ;wegen mögl. Ansprache des Z80 über INT2
0003 214501               LD      HL,initab+2     ;Tabelle der internen Port-Initialwerte
0006 1833                 JR      gorst00         ;dort weiter

                ;weitere RST-Routinen können nachgerüstet werden
0008 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;Platzhalter für RST 08
0010 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 10
0018 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 18
0020 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 20
0028 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 28
0030 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 30
0038 FF                   DB      Offh,Offh,Offh                  ;dto. RST 38 (JP genügt)

                ;Fortsetzung der RST 00-Routine (Reset oder Trag)
0038 01340C     gorst00   LD      BC,0c34h                ;12 Werte ab Port 34h
                ;         OTIMR                           ;Initialwerte für interne
003E ED                   DB      0edh,93h                ;Ports aufsetzen (s. o.)

                ;Warteprogramm und EPROM-Inhalt per DMA ins Himem transferieren
0040 219800               LD      HL,tabend       ;Ende der Wertetabelle für den DMAC
0043 CD8700               CALL    dma2            ;per DMA Himem-Programm hochtransferieren
0046 68                   LD      L,B             ;HL (- 0000 (Anfang EPROM, H=00, BC=0000)
0047 50                   LD      D,B             ;DE (- 0000 (LSW logische Zieladresse)
0048 58                   LD      E,8
0049 3E60                 LD      A,60h           ;Quellber. 0.xxxx, Zielber. 6.xxxx (MSB)
0048 CD6900               CALL    dma0            ;EPROM-Inhalt nach 6.0000-6.FFFF übertr.
004E C33CF0               JP      inidone+offs    ;Warteroutine anspringen

                ;Platzhalter bis zum NMI-Einsprung an 0066
0051 FFFF                 DW      Offffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh
0061 FF                   DB      Offh,Offh,Offh,Offh,Offh

                ;Einsprung für NMI (nicht vorgesehen, kann aber nachgerüstet werden)
0066 FF         nmi       DB      Offh,Offh,Offh  ;3 Bytes, können mit JP überschr. werden

F000            himem     EQU     0f000h          :Common 1, Arbeitsadresse des Programms
EF97            offs      EQU     himem-$         ;Offset zur Arbeitsadresse des Programms

                ;der folgende Teil des EPROM-Inhalts arbeitet im Speicher ab 7.F000

                ;Unterprogr. für DMA: HL = LSW Quelle, DE = LSW Ziel, BC = Bytezähler,
                ;A Bit 0-2 = MSB der Quelle, A Bit 4-6 = MSB des Ziels
0069 222BF0     dma0      LD      (srclog+offs),HL        ;Quellbereich   LSW     x.HL
006C ED532EF0             LD      (dstlog+offs),DE        ;Zielbereich    LSW     x.DE
0070 ED4331F0             LD      (dmacnt+offs),BC        ;Bytezähler für DMA aufsetzen
0074 F5                   PUSH    AF                      ;phys. Quell- und Ziel-MSB retten
0075 E60F                 AND     0fh                     ;Quellbits maskieren
0077 322DF0               LD      (srcphys+offs),A        ;Quellbereich   phys.   A.HL
007A F1                   POP     AF                      ;MSB des Quell- und Zielblocks
0078 OF                   RRCA                            ;in die unteren Bits rotieren
007C OF                   RRCA
007D OF                   RRCA
007E OF                   RRCA
007F E60F                 AND     Ofh                     ;Zielbits maskieren
0081 3230F0               LD      (dstphys+offs),A        ;Zielbereich    phys.   A.DE
                ;Einsprung in DMA, falls die Tabelle der Adressen vorbereitet ist 0084
0084 2132F0     dma1      LD      HL,tabend+offs          ;Beginn der DMA-Wertetabelle
                ;Bank-unabhängiger Teil des Programms, kann von überallher gecallt werden
0087 012708     dma2      LD      BC,0827h                ;8 Werte ab Port 27 abwärts
                ;         OTDMR                           ;DMAC aufsetzen
008A ED                   DB      0edh,9bh                ;(s. o.)
008C 013102               LD      BC,0231h                ;2 Werte ab Port 31 abwärts
                ;         OTDMR                           ;Rest aufsetzen u. DMA abschießen
008F ED                   DB      0edh,9bh                ;(s. o.)
```

```
0091 C9                   RET

          ;Init.-Werte des DMAC zur Kopie des Himem-Programms ins RAM ab 7.F000 0092
0092 41                   DB      41h     ;30.    DSTAT:  Kanal 0, Zustand 'scharf'
0093 02                   DB      02h     ;31,    DMODE:  Speicher zu Speicher, Burst-Modus
0094 6900       srclog    DW      dma0    :20/1, SAROL/H: Quelladresse    LSW     dma0
0096 00         srcphys   DB      00h     ;22,    SAROB:  MSB     0.xxxx
0097 00F0       dstlog    DW      himem   ;23/4, DAROL/H: Zieladresse     LSW     F000
0099 07         dstphys   DB      07h     ;25,    DAROB:  MSB     7.xxxx
009A 0403       dmacnt    DW prt-dma0     ;26/7, BCROL/H: Bytezähler für die Programmlänge
0098            tabend    EQU     S-1     ;Ende der Tab.. womit der Zeiger HL geladen wird

          ;nach der Initialis. auf Ansprache des Z80 warten, sie interpretieren
009C 00         rdyflag   DB      00h             ;Flag für noch nicht beendeten Job
009D 21DAF0     waitz80   LD      HL,initab+offs  ;Tabelle der internen Port-Initialwerte
00A0 01320E               LD      BC,0e32h        ;14 Werte ab Port 32h
          ;               OTIMR                   ;Initialwerte für interne Ports aufsetzen
00A3 ED                   DB      0edh,93h        ;(s. o.)
00A5 3EF3       inidone   LD      A,inttab+offs/256       ;MSB der Interrupt-Vektortabelle
00A7 ED47                 LD      I,A             ;ins Interrupt-Vektorregister
00A9 D3FB       copdone   OUT     (0fbh),A        ;falls der HD noch die Buskontrolle hat
00AB 3100F1               LD      SP,trap+offs    ;Stack im freien Bereich
00AE 210000               LD      HL,0000h        ;Prüfadresse (im Z80-RAM unkritisch)
0081 AF           XOR     A                       ;A (- 00, Flag für beendeten Job
0082 3233F0               LD      (rdyflag+offs),A        ;Bereitsch. des HD dort vermerken
0085 FB                   EI                      ;für evtl. Z80-Interrupt
0086 7E         polloop   LD      A,(HL)          ;Testbyte laden
0067 B7                   OR      A               ;immer noch 00? (EPROM noch eingeblendet)
0088 28FC                 JR      Z,polloop       ;falls noch EPROM

          ;der Z80 gab soeben die Kontrolle an den HD ab (auch Entry für Z80-INT)
008A F3         intz80 DI ;wegen unklaren Interrupt-Modus usw.
0088 2A33F0               LD      HL,(rdyflag+offs)       ;Flag für anhängigen Job
008E 7E                   LD      A,(HL)          ;Flag laden
00BF 67                   OR      A               ;ist z. Zt. noch ein Job in Arbeit?
0000 2802                 JR      Z,nojob         ;falls nicht mehr
0002 ED4D                 RETI                    ;falls ja (nur bei INT vom Z80)
00C4 ED780300   nojob     LD      SP,(spbuff)     ;Stackpointer des Z80 übernehmen

          ;vereinbarte Vorgaben für den Stack des Z80:
          ; (SP+0): evtl. Wegweiser in die Utility-Vektortabelle
          ; (SP+1): Signalbyte für die Art der übergebenen Aufgabe
          ;
          ;bei endgültiger Übernahme:
          ; (SP+2), (SP+3): Platz für den Befehlsstring zum Banking
          ; (SP+4), (SP+5): Startadresse (Entrypoint) für den HD 64180
          : evtl. darüber : Registerinhalte, die gePOPt werden können
          ;
          ;bei vorübergehendem Job:
          ; (SP+2), (SP+3): Startadresse (Entrypoint) für den HD 64180
          ;
          ;bei Übergabe von Rechenergebnissen:
          ; (SP+2), (SP+3): Pufferadresse im Z80-RAM
          ;
          ;bei Speicherkopie von 0.0000-0.FFFF nach 1.0000-3.FFFF im Z80-RAM
          ;und bei Abruf einer programmierten Utility aus der Vektortabelle:
          ; keine weiteren Daten
          ;
          ;Vereinbarung für das Signalbyte (SP+0) (Register B nach POP BC)
          ; B:00:   endgültige Übernahme des Systembusses durch den HD 64180
          ; B=01:   2 kB Ergebnisse eines früheren Jobs aus 7.F800 übergeben
          ; B=02:   eine Utility aus der Vektortabelle aufrufen
          ; B=FF:   0.0000-0.FFFF in die 64-kB-Blocks bis 3.FFFF kopieren
          ; FF>8>02: einen sonstigen vorübergehenden Job vom Z80 übernehmen

0008 C1                   POP     BC              ;unterste Stack-Ebene lesen
00C9 78                   LD      A,B             ;Task-Signalbyte vom Z80-Stack
00CA 77                   LD      (HL),A          ;Flag für anhängenden Job setzen
00CB 3C                   INC     A               ;A=FF, das RAM 0.0000-0.FFFF kopieren?
00CC 2840                 JR      Z,copy          ;falls ja
00CE 3D                   DEC     A               ;A=00, Kontrolle endgültig übernehmen?
```

```
00CF 200F                JR      NZ,job          ;>00: Job übernehmen o. Ergebn. übergeben

               ;der Z80 hat im Stack signalisiert, daß er die Kontrolle ganz abgibt
00D1 2173F0              LD      HL,switch+offs  ;Befehlsfolge zum Umschalten von Common 1
00D4 D1                  POP     DE              ;Zieladr. des Umschaltbefehls a. d. Stack
00D5 D5                  PUSH    DE              ;noch einmal auf den Stack als RET-Adr.
0006 010400              LD      BC,0004h        ;Länge der beiden Befehle
00D9 EDBO                LDIR                    ;übertragen
0008 C9                  RET                     ;banken und Arbeit des Z80 übernehmen

               switch
               ;         OUT0    (38h),A         ;A=00, Common 1 ab 0.0000 einstellen
00DC ED                  DB      0edh,39h,38h    ;Hex-Entsprechung des Befehls
00DF C9                  RET                     ;an die vom Z80 vorgegebene Adr. springen

               ;der Z80 hat im Stack signalisiert, daß er für den HD 64180 einen Job hat
               ;er muß mit RET aufhören (wird wie ein Unterprogramm behandelt)
00E0 E1        job       POP     HL              ;Einsprungsadresse für den Job
00E1 3100F1              LD      SP,trap+offs    ;Stack wieder im eigenen RAM
00E4 E5                  PUSH    HL              ;Entry des Jobs auf den eigenen Stack
00E5 2134F0              LD      HL,waitz80+offs ;Rücksprungadresse
00E8 E3                  EX      (SP),HL         ;als RET-Adresse a. d. Steck, HL <- Entry
00E9 E5                  PUSH    HL              ;Programm-Entry retten
00EA 3D                  DEC     A               :A=01, Ergebnisse hinunterladen?
00E8 2816                JR      Z,putres1       ;falls ja
00ED 3D                  DEC     A               ;A=02, eine Utility aus d. Vektortabelle?
00EE 2823                JR      Z,vectask       ;falls ja
00F0 210000              LD      HL,0000h        ;Quelladresse für DMA x.0000 (LSW)
00F3 54                  LD      D,H             ;Zieladresse dto.
00F4 5D                  LD      E,L
00F5 0100F0              LD      BC,himem        ;Größe des zu übertrag. Speicherblocks
00F8 3E70                LD      A,70h           ;Quelle phys. 0.xxxx, Ziel 7.xxxx (MSB)
               ;         OUT0    (39h),A         ;Bank-Area ebenfalls ab 7.0000
00FA ED                  DB      0edh,39h,39h    ;(s. 0.)
00FD CD00F0    dmaresl   CALL    dma0+offs       ;Speicher hoch- oder runterkopieren
0100 D3FB      vectret   OUT     (0fbh),A        ;dem Z80 die Buskontrolle zurückgeben
0102 C9                  RET                     ;evtl. Programm anspr., dann nach waitz80

               ;der Z80 hat signalisiert, daß er 2 kB Resultate aus 7.F800 abholen will 0103 D1
               putresl   POP     DE              ;für Ergebnisse: DE = Zielbereich
0104 2100F8              LD      HL,0f800h       ;Puffer für Job-Ergebnisse (Quellbereich)
0107 010008              LD      BC,0800h        ;Länge des Puffers immer 2 kB
010A 3E07                LD      A,07h           ;Quellbereich 7.xxxx, Zielbereich 0.xxxx
010C 18EF                JR      dmaresl         ;Ergebnisse übertragen, zurück zu waitz80

               ;der Z80 hat signalisiert, 0.0000-0.FFFF nach 1.0000-3.FFFF zu kopieren
010E CDBAF0    COPY      CALL    clone+offs      ;den Speicher kopieren
0111 1896                JR      copdone         ;zurück in die Warteschleife

               ;der Z80 ruft einen Task auf, dessen Adresse in der Vektortabelle steht
0113 2197F0    vectask   LD      HL,vectret+offs ;Rückkehradresse für diesen Fall
0116 E3                  EX      (SP),HL         ;auf den Stack, Nonsense-Adresse löschen
0117 47                  LD      B,A             ;8 als MSB des Vektorsummanden =00
0118 CB01                RLC     C               ;LSB verdoppeln wegen 2 Bytes pro Vektor
011A 2100F2              LD      HL,vectab+offs  ;Beginn der Vektortabelle
011D 09                  ADD     HL,BC           ;Zeiger auf die Task-Adresse stellen
011E 7E                  LD      A,(HL)          ;das LSB der Adresse laden
011E 23                  INC     HL              ;auf das MSB stellen
0120 66                  LD      H,(HL)          ;dieses laden
0121 6F                  LD      L,A             ;HL - Adresse der Utility
0122 E9                  JP      (HL)            ;sie anspringen

               ;UP zur Speicherkopie, das auch unabhängig von A=FF gecallt werden kann
               ;je nach gewünschter Blockgröße, Startbank, Zahl der Banks, Adressen usw.
               ;Einsprung mit unterschiedl. Registerinhalten an verschiedenen Stellen
0123 010000    clone     LD      BC,0000h        ;Bytezähler 0000 : 1.0000 (64 kB)
0126 60                  LD      H,B             ;Quelladresse LSW 0000
0127 69                  LD      L,C
0128 50                  LD      D,8             ;Zieladresse LSW 0000
0129 59                  LD      E,C
012A 3E10                LD      A,10h           ;Quelle phys. 0.0000, Ziel phys. 1.0000
```

```
0120 ED43CDF0             LD      (cloncnt+offs),BC       ;den Zähler aufsetzen
0130 0603                 LD      B,03h           ;3 Banks zu je 64 kB
0132 C5         coploop   PUSH    BC              ;Schleifenzähler retten
0133 E5                   PUSH    HL              ;dto. LSW Quelladresse
0134 F5                   PUSH    AF              ;dto. MSB Quell- und Zieladresse
0135 010000               LD      BC,0000h        ;Bytezähler 0000 (:1.0000)
0136            cloncnt   EQU     $-2             ;kann auf andere Größe geändert werden
0138 CD00F0               CALL    dma0+offs       ;eine Bank kopieren
0138 F1                   POP     AF              ;MSB Quell- und Zieladresse
013C C610                 ADD     A,10h           ;der nächste phys. 64-kB-Block
013E E1                   POP     HL              ;LSW Quelladresse
013F C1                   POP     BC              ;Schleifenzähler
0140 10F0                 DJNZ    coploop         ;bis 3 64-kB-Blocks kopiert sind
0142 C9                   RET

                ;Initialwerte der internen Ports 32-3F (nach RESET erst ab 34 gelesen)
0143 F0         initab    DB      0f0h            ;32, DCNTL: 3 WAITs für Memory und I/O
0144 00                   DB      00h             ;33, IL: interne INTs ab 00 in der Tabelle
0145 07                   DB      07h             ;34, ITC: alle ext. Interrupts zugelassen
0146 FF                   DB      0ffh            ;35: belanglos
0147 83                   DB      83h             ;36, RCR: Refreshes möglichst zeitsparend
0148 FF                   DB      0ffh            ;37: belanglos
0149 70                   DB      70h             ;38, CBR: Common 1 ab 7.x000
014A 00                   DB      00h             ;39, BBR: Bank-Area ab  0.x000
014B F0                   DB      0f0h            ;3A, CBAR: Common 0 nicht definiert
                                                  ;Bits 0-3: Bank-Area: 0.0000 - 0.EFFF
                                                  ;Bits 4-7: Common 1 : 7.F000 - 7.FFFF
014C FF                   DB      0ffh            ;3B:    belanglos
014D FF                   DB      0ffh            ;3C:
014E FF                   DB      0ffh            ;3D:
014F FF                   DB      0ffh            ;3E:
0150 20                   DB      20h             ;3F, ICR: interne Ports ab 00, IOSTOP-Modus

                ;Füllbytes, um trap0 im Himem auf die "gerade' Adresse F100 zu bekommen
                ;hier liegt auch der HD-eigene Stack (max. 12 Ebenen!)
0151    FFFF              DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0161    FFFF              DW      0ffffh,0ffffh,0ffffh,0ffffh

                ;Fehlerbehandlung bzw. Illegal-Simulation nach einem TRAP-Interrupt
0169 F5         trap      PUSH    AF              ;Akku und Flags retten
                ;         IN0     A,(34h)         ;TRAP-Bit und UF0-Bit einlesen
016A ED                   DB      0edh,38h,34h    ;(s. o.)
016D E67F                 AND     7fh             ;TRAP-Bit zurücksetzen
                ;         OUT0    (34h),A         ;TRAP löschen
016F ED                   DB      0edh,39h,34h    ;(s. o.)
0172 3211F1               LD      (ufo+offs),A    ;UFO retten
0175 F1                   POP     AF              ;Akku vom Stack holen
0176 E3                   EX      (SP),HL         ;HL mit Trapadresse tauschen
0177 F5                   PUSH    AF              ;Akku wieder retten
0178 C5                   PUSH    BC              ;retten
0179 3E00                 LD      A,00h           ;UFO (Dummy-Operand)
017A            ufo       EQU     $-1             ;je nach dem
0178 CB77                 BIT     6,A             ;UF0-Bit gesetzt?
017D 2801                 JR      Z,nosub         ;nein, Opcode direkt davor
017F 28                   DEC     HL              ;sonst Zeiger -1
0180 2B         nosub     DEC     HL              ;Zeiger auf illegalen 0pcode
0181 7E                   LD      A,(HL)          ;Opcode laden
0182 47                   LD      B,A             ;und retten
0183 FECB                 CP      0cbh            ;CBxx? (SLIA bzw. SLL)
0185 23                   INC     HL              ;Zeiger weiterstellen
0186 7E                   LD      A,(HL)          ;2. Byte laden
0187 200D                 JR      NZ,ddfd         ;falls 0perand mit IX/IY
0189 D620                 SUB     20h             ;Akku korrigieren
0186 3228F1               LD      (cbxx+offs),A   ;in den Programmtext laden
018E C1                   POP     BC              ;Register restaurieren
018F F1                   POP     AF
0190 23                   INC     HL              ;PC neu setzen
0191 E3                   EX      (SP),HL         ;als RET-Adresse auf den Stack
0192 37                   SCF                     ;Cy - 1, Bit 0 bei SLIA/SLL immer 11
0193 CB10                 RL      B               ;rotieren (8 ist Dummy-Operand)
0194            cbxx      EOU     S-1             ;0perand für Register B-A
```

```
0195 C9                      RET

0196 FECB        ddfd        CP      0cbh               ;ein ähnlicher Befehl für IX/IY?
0198 2827                    JR      Z,ddfdcb           ;falls ja
019A 3252F1                  LD      (substit+offs),A         ;nächsten 0pc. in d. Programmtext
019D FE26                    CP      26h                ;LD HX/HY,const ?
019F 2807                    JR      Z,ldcon            ;falls ja
01A1 FEZE                    CP      2eh                ;LD LX/LY,const ?
01A3 2803                    JR      Z,ldcon            ;falls ja
01A5 AF                      XOR     A                  ;keine Konstante: NOP
01A6 1802                    JR      gotrap             ;dort weiter
01A8 23          ldcon       INC     HL                 :Zeiger auf die Konstante
01A9 7E                      LD      A,(HL)             ;diese laden
01AA 3253F1      gotrap      LD      (const+offs),A     ;evtl. Konstante in den Programmtext
01AD 78                      LD      A,B                ;DD/FD zurück
01AE 324FF1                  LD      (ddfd1+offs),A     ;in den Programmtext setzen
0181 3255F1                  LD      (ddfd2+offs),A     ;dort auch
0164 C1                      POP     BC                 ;Register restaurieren
0185 F1                      POP     AF
0186 23                      INC     HL                 ;PC neu setzen
0167 E3                      EX      (SP),HL            ;als RET-Adresse auf den Stack
0188 DDE5        ddfd1       PUSH    IX                 ;bzw. IY
018A E3                      EX      (SP),HL            ;HL - IX/IY
0168 7F          substit     LD      A,A                ;Operation ausführen (hier Dummy)
016C 00          const       DB      00h                ;evtl. Konstante
01BD E3                      EX      (SP),HL            ;neues IX/IY zurück
018E DDE1        ddfd2       POP     IX                 ;bzw. IY
01C0 C9                      RET

01C1 78          ddfdcb      LD      A,B                ;DD/FD zurück
01C2 32A2F1                  LD      (ddfd3+offs),A     ;in den Programmtext laden
0105 32A7F1                  LD      (ddfd4+offs),A     ;dort auch
01C8 23                      INC     HL                 ;Zeiger auf den Offset
01C9 7E                      LD      A,(HL)             ;.d" aus (IX/IY+d) laden
01CA 32A4F1                  LD      (dspl1+offs),A     ;in den Programmtext laden
01CD 32A9F1                  LD      (dspl2+offs),A     ;dort auch
0100 23                      INC     HL                 ;Zeiger auf Operation
0101 7E                      LD      A,(HL)             ;Befehl laden
0102 E607                    AND     07h                ;Bits 0-2 maskieren
01D4 3C                      INC     A                  ;+1
0105 47                      LD      B,A                ;als Zähler
0106 3E3E                    LD      A,3eh              ;0pcode 46-7E erzeugen
01D8 C608        opclop1     ADD     A,08h              ;(46/4E/56/5E/66/6E/7E
010A 10FC                    DJNZ    opclop1            ;= LD B/C/D/E/H/L/A,(IX+d) )
010C 32A8F1                  LD      (load+offs),A      ;in den Programmtext
01DF 7E                      LD      A,(HL)             ;Befehl noch einmal laden
01E0 E6F8                    AND     0f8h               ;Bits 0-2 löschen
01E2 0F                      RRCA                       ;und herausrotieren
01E3 0F                      RRCA
01E4 0F                      RRCA
01E5 3C                      INC     A                  ;+1
01E6 47                      LD      B,A                ;als Zähler
01E7 3EFE                    LD      A,0feh             ;(06/0E/16/1E/26/2E/3E
01E9 C608        opclop2     ADD     A,08h              ;= RLC/RRC/RL/RR/SLA/SRA/SRL/RES/SET ...)
01EB 48                      LD      C,B                ;NOP erzeugen (für reta)
01EC FE36                    CP      36h                ;SLIA/SLL ?
01EE 200B                    JR      NZ,noslia          ;falls nein
01F0 0637                    LD      B,37h              ;SCF erzeugen
01F2 7E                      LD      A,(HL)             ;Befehl restaurieren
01F3 FE36                    CP      36h                ;SLIA/SLL (IX/IY+d) ?
01F5 2002                    JR      NZ,nopro           ;falls nein, kein Problem
01F7 0EC9                    LD      C,0c9h             ;Opcode RET für reta
01F9 3E16        nopro       LD      A,16h              ;RL erzeugen (statt SLIA/SLL)
01F8 32A5F1      noslia      LD      (opera+offs),A     ;in den Programmtext
01FE 78                      LD      A,B                ;SCF oder NOP
01FF 32A1F1                  LD      (cflag+offs),A
0202 79                      LD      A,C                ;RET oder N0P
0203 32A6F1                  LD      (reta+offs),A
0206 C1                      POP     BC                 ;Register restaurieren
0207 F1                      POP     AF
0208 23                      INC     HL                 ;PC korrigieren
```

```
0209 E3                    EX      (SP),HL         ;als RET-Adresse auf den Stack

020A 00           cflag    NOP                     ;oder SCF
0208 00           ddfd3    NOP                     ;DD oder FD für IX/IY
020C C8                    DB      0cbh            ;immer C8
020D 00           dsp11    NOP                     ;Displacement d bei (IX/IY+d)
020E 00           opera    NOP                     ;Operationsanweisung
020F 00           reta     NOP                     ;RET bei SLIA/SLL
0210 00           ddfd4    NOP                     ;DD/FD für IX/IY
0211 7F           load     LD      A,A             ;Ladebefehl (hier Dummy)
0212 00           dsp12    NOP                     ;s. dspll
0213 C9                    RET                     ;Simulation des Befehls beendet

                  ;Füllbytes für die "gerade" Adresse F200 der Utility-Vektortabelle
                  ;mit freundlicher Empfehlung:
0214 28                    DM      '(R) #0000 - (C) 1987 by The HACKT0RY '
0239 48                    DM      'Helmut Bernhardt, Arnulf Sopp, Gerald Schröder'
0267 FFFF                  DW      0ffffh

                  ;bei F200 die Tabelle der Einsprungsadressen bei vektorisiertem Z80-Task
                  ;zunächst alle auf FF (können nachgebrannt oder im RAM programm. werden)
0269 FFFF         vectab   DW      0ffffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh
0279 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0289 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0299 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02A9 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0289 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02C9 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0209 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02E9 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02F9 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0309 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0319 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0329 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0339 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0349 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0359 FFFF           DW     0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh

                  ;dahinter die Vektortabelle für Interrupts des Typs 2
                  :im EPR0M ist nur der Einsprung für einen INT durch den Z80 initialisiert
0369 FFFF         inttab   DW      0ffffh  ;für INTs auf den INT1-Pin (nicht init.)
0368 51F0                  DW      intz80+offs     ;dto. INT2-Pin (INT durch den Z80)
                  ;weiter für die internen Interrupts (nicht initialisiert)
                  ;(nur bis hierher muß das EPROM gebrannt werden, solange keine weiteren
                  ;Interrupt-Vektoren aufgesetzt werden; es folgen vorerst nur FF-Bytes) 036D
FFFF              prt      DW      0ffffh,0ffffh   ;dto. PRT Kanal 0 und 1
0371 FFFF                  DW      0ffffh,0ffffh   ;dto. DMA Kanal 0 und 1
0375 FFFF                  DW      0ffffh          ;dto. CSI/0
0377 FFFF                  DW      0ffffh,0ffffh   ;dto. ASCI Kanal 0 und 1
                  ;dasselbe für die übrigen externen Interrupts (nicht weiter gelistet)
0378              end      EQU     S               ;Hilfslabel für Programmende und -lange

                  ;Hilfslabels zur Kontrolle v. Unterprogramm- u. Tabellenadressen im Himem
F000              dmaabs   EQU     dma0+offs       ;Arbeitsadresse des Unterprogramms dma0
F028              dmatabs  EQU     srclog+offs     ;dto. Adressentabelle des DMAC
F0BA              clonabs  EQU     clone+offs      ;dto. Unterprogramm clone0
F00A              initabs  EQU     initab+offs     ;dto. Tabelle der int. Port-Initialwerte
F100              trapabs  EQU     trap+offs       ;dto. Trap-Routine
F200              vectabs  EQU     vectab+offs     ;dto. Utility-Vektortabelle
F300              inttabs  EQU     inttab+offs     ;dto. INT-Vektortabelle

0000                       END
00000 Fehler
cbxx    0194    cflag   020A    clonabs F0BA    cloncnt 0136    clone   0123    const   01BC
copdone 00A9    coploop 0132    copy    010E    ddfd    0196    ddfd1   0188    ddfd2   016E
ddfd3   0206    ddfd4   0210    ddfdcb  01C1    dma0    0069    dmal    0084    dma2    0087
dmaabs  F000    dmacnt  009A    dmaresl 00FD    dmatabs F026    dspl1   020D    dsp12   0212
dstlog  0097    dstphys 0099    end     0378    gorst00 0038    gotrap  01AA    himem   F000
inidone 00A5    initab  0143    initabs F0DA    inttab  0369    inttabs F300    intz80  00BA
job     00E0    ldcon   01A8    load    0211    nmi     0066    nojob   00C4    nopro   01F9
noslia  01FB    nosub   0180    offs    EF97    opclop1 0108    opclop2 01E9    opera   020E
```

```
polloop 0066    prt     036D    putresl 0103    rdyflag 009C    reta    020F    spbuff  0003
srclog  0094    srcphys 0096    substit 0168    switch  000C    tabend  0098    trag    0169
trapabs F100    ufo     017A    vectab  0269    vectabs F200    vectask 0113    vectret 0100
waitz80 009D
                ; Listing 2: Der Z80 gibt dem HD 64180 einen vorübergehenden Job
0003            spbuff    EOU     0003h   ;bei TRS-80 u. ä.: stattdessen RAM-Adresse wählen
5200                      ORG     5200h
5200 211E52     start     LD      HL,job          ;Zeiger auf das Programm für den HD
5203 E5                   PUSH    HL              ;auf dem Steck ablegen
5204 7C                   LD      A,H             ;A - irgendetwas zwischen 02 und FE
5205 F5                   PUSH    AF              ;Signalbyte für 'bitte Job übernehmen!'
5206 ED730300             LD      (spbuff),SP     ;dem HD den Stackpointer mitteilen
520A D3F6                 OUT     (0fbh),A        ;die Buskontrolle an den HD übergeben
                ;jetzt ist der Z80 für einige Mikrosekunden im Schlummerzustand
                ;der HD holt sich soeben den Job nach oben
                ;danach:
520C F1                   POP     AF              ;Stack wieder bereinigen
520D F1                   POP     AF              ;(es waren zwei Ebenen)
                ;         CALL    WICHTIG         ;inzwischen etwas Wesentlicheres tun
520E 210038               LD      HL,3800h        ;Bildschiradresse (sichtbar ab 3C00)
5211 E5                   PUSH    HL              ;als Pufferadresse für den HD
5212 3E01                 LD      A,01h           ;Signalbyte 'bitte Ergebnisse abliefern!'
5214 F5                   PUSH    AF              :für den HD auf den Stack
5215 ED730300             LD      (spbuff),SP     ;wie oben
5219 D3FB                 OUT     (0fbh),A        ;Buskontrolle wieder an den HD übergeben
                ;wieder ist für einen Moment Funkstille, während der der HD die Codefolge
                ;20h, 20h. 20h, ... (Leerzeichen) in den Bildschirmspeicher lädt
                :der Bildschirm ist jetzt gelöscht
                ;danach:
5216 F1                   POP     AF              ;dasselbe wie oben
521C F1                   POP     AF
521D C9                   RET                     ;oder was auch immer
521E 2100F3     job       LD      HL,0f300h       ;Datenpuffer für Rechenergebnisse
5221 1101F3               LD      DE,0f301h       ;die nächste Stelle
5224 01FF07               LD      BC,07ffh        ;Länge 2 k8
5227 3620                 LD      (HL),' '        ;am Pufferanfang ein Leerzeichen ablegen
5229 EDB0                 LDIR                    ;den ganzen Puffer mit Blanks füllen
5226 C9                   RET                     ;in die Warteschleife zurückkehren
5200                      END     start
00000 Fehler
job  521E       spbuff 0003     start 5200
                ; Listing 3: Der Z80 gibt die Buskontrolle endgültig an den HD ab
0003            spbuff    EOU     0003h           ;TRS-80 u. ä.: stattdessen RAM-Adresse wählen!
4200                      ORG     4200h           ;im Sektorpuffer ist Platz genug
4200 211042     start     LD      HL,entry        ;Ansprungsadresse für den HD 64180
4203 E5                   PUSH    HL              ;für den HD auf den Stack
4204 211A42               LD      HL,bank         ;Platz für das Banking-Programm des HD
4207 E5                   PUSH    HL              ;dto.
4208 AF                   XOR     A               ;A - 00, Signal für endgültige Übernahme
4209 F5                   PUSH    AF              ;dto.
420A ED730300             LD      (spbuff),SP     ;dem HD verraten, wo der Stack ist
420E D3FB                 OUT     (0fbh),A        ;und den Bus endgültig übergeben
4210 ED56       entry     IM      1               ;für die Interrupt-Bearbeitung von G-D0S
4212 3E01                 LD      A,01h           ;nur noch Interrupts am Pin INT0 zulassen
                ;         OUT0    (34h),A         ;Ausgabe auf den INT/TRAP-Port
4214 ED                   DB      0edh,39h,34h    ;Hex-Entsprechung des Befehls
```

```
4217 C32D40               JP      402dh           ;DOS-Entry, dort steht z. B. auch EI
                bank                              ;Platz für die Banking-Routine des HD
4200                      END     start
00000 Fehler
bank    421A    entry 4210      spbuff 0003     start 4200
                ; Listing 4: Einsprung in d. Trap-Routine beim Genie 3s unter G-D0S
0000                      ORG     0000h           ;RST-00-Routine
0000 C30039               JP      trap            ;dort weitermachen
3900                      ORG     3900h           ;hier ist z. B. freies RAM
3900 322739     trap      LD      (accbuf1),A     ;Akku retten (Stack unklar, kein PUSH!)
3903 ED5F                 LD      A,R             ;Interrupt-Enable-Flipflop ins P/V-Flag
3905 F3                   DI                      ;Interrupts maskieren
3906 3EF3                 LD      A,0f3h          ;auf Verdacht: DI
3908 E20D39               JP      P0,intok        ;falls die Interrupts disabled waren
3908 3EFB                 LD      A,0fbh          ;sonst den Befehl EI laden
390D 324439     intok     LD      (eidi),A        ;und vor den RET-Befehl patchen
                ;         IN0     A,(38h)         ;den Inhalt des CBR lesen
3910 ED                   DB      0edh,38h,38h    ;(Hex-Entsprechung des Befehls)
3913 323939               LD      (cbrbuff),A     ;und p,uffern
3916 3E70                 LD      A,70h           ;Common 1 des HD ab 7.x000
                ;         OUT0    (38h),a         ;auf CBR ausgeben
3918 ED                   DB      0edh,39h,38h    ;(s. o.)
                ;         IN0     A,(3ah)         ;den Inhalt des CBAR lesen
391B ED                   DB      0edh,38h,3ah    ;(s. o.)
391E 323E39               LD      (cbarbuf),A     ;und p,uffern
3921 F6F0                 OR      0f0h            ;Common 1 des HD ab 7.F000
                ;         OUT0    (3ah),A         ;MMU neu konfigurieren
3923 ED                   DB      0edh,39h,3ah    ;(s. o.)
3926 3E00                 LD      A,00            ;Akku restaurieren (00 ist Dummy)
3927            accbufl   EQU     $-1             ;dort ist der Akku-Inhalt gepuffert
3928 ED733339             LD      (spbuff),SP     ;Stackpointer retten
392C 3100F1               LD      SP,0f100h       ;Stack im Common-1-Bereich
392F CDD0F0               CALL    0f100h          ;an 7F100 steht die Trap-Routine
3932 310000               LD      SP,0000h        ;den alten Stack restaur. (Dummy-Adresse)
3933            spbuff    EQU     $-2             ;je nach dem
3935 324339               LD      (accbuf2),A     ;Akku und Flags retten (ohne Stack, s. o)
3938 3E00                 LD      A,00h           ;alter Inhalt des CBR (00 ist Dummy)
3939            cbrbuff   EQU     $-1             ;je nach dem
                ;         OUT0    (38h),A         ;den alten Registerinhalt restaurieren
393A ED                   DB      0edh,39h,38h    ;(s. o.)
393D 3E00                 LD      A,00h           ;dasselbe für das CBAR
393E            cbarbuf   EQU     1-1
                ;         OUT0    (3ah),A
393F ED                   DB      0edh,39h,3ah
3942 3E00                 LD      A,00h           ;Akku und Flags restaurieren (s. o.)
3943            accbuf2   EQU     $-1
3944 FB         eidi      EI      ;oder DI, je nach altem Zustand
3945 C9                   RET     ;das war's
0000                      END
00000 Fehler
accbufl 3927    accbuf2 3943    cbarbuf 393E    cbrbuff 3939    eidi    3944    intok   390D
spbuff 3933     trap    3900
                ; Listing 5: möglicher TRAP-Einsprung unter CP/M u. a.
0000                      ORG     0000h
0000 C30300               JP      trap
                ;wo trap im RAM liegen kann, muß je nach CP/M-Version entschieden werden
                ;         ORG     irgendwo
                trap
                ;         IN0     A,(34h)         ;INT/TRAP-Register lesen
0003 ED                   DB      0edh,38h,34h    ;(Hex-Entsprechung des Befehls)
0006 CB7F                 BIT     7,A             ;war das ein TRAP-Interrupt?
                ;         JP      NZ,warmstart    ;falls nein (Sprungadresse je nach dem)
0008 E67F                 AND     7fh             ;TRAP-Bit löschen
```

```
000A 321300     00015           LD      (portbuf),A     ;den Input von Port 34h puffern
000D AF         00016           XOR     A               ;A <- 00, auch die ITE-Bits =0 (wie DI)
                00017 ;         OUT0    (34h),A         ;TRAP resetten und Interrupts disablen
000E ED         00018           DB      0edh,39h,34h    ;(s. o.)
                00019
                00020 ;hier folgt die Trap-Routine (oder CALL nach 7.F100), und so endet sie:
                00021
0011 F5         00022 exitrap   PUSH    AF              ;Akku und Flags retten
0012 3E00       00023           LD      A,00h           ;alter Inhalt von Port 34 (Dummy-Operand)
0013            00024 portbuf   EQU     $-1             ;Operanden-Byte des Befehls = Port-Inhalt
                00025 ;         OUT0    (34h),A         ;Inhalt des Ports restaurieren
0014 ED         00026           DB      0edh,39h,34h    ;(s. o.)
0017 F1         00027           POP     AF              ;Akku und Flags restaurieren
0018 C9         00028           RET
                00029
0000            00030   END
00000 Fehler
exitrap 0011    portbuf 0013    trap    0003
                00001 ; Listing 6: Testprogramm für maximale Arbeitsgeschwindigkeit
                00002
0003            00003 spbuff    EQU     0003h           ;TRS-80 u. ä.: auf freie RAM-Adresse ändern!
                00004
4200            00005   ORG     4200h                   ;das paßt in den Sektorpuffer
                00006
4200 211042     00007 start     LD      HL,entry        :Ansprungadresse für den HD 64180
4203 E5         00008           PUSH    HL              ;für den HD auf den Stack
4204 211F42     00009           LD      HL,bank         ;Platz für das Banking-Programm des HD
4207 E5         00010           PUSH    HL              ;dto.
4208 AF         00011           XOR     A               ;A - 00. Signal für endgültige Übernahme
4209 F5         00012           PUSH    AF              ;dto.
420A ED730300   00013           LD      (spbuff),SP     ;dem HD verraten, wo der Stack jetzt ist
420E D3FB       00014           OUT     (Ofbh),A        ;und den Bus endgültig übergeben
                00015
                00016 ;Einsprung bei der endgültigen Busübernahme durch den HD 64180
4210 ED56       00017 entry     IM      1               ;für die Interrupt-Bearbeitung von G-DOS
4212 3E01       00018           LD      A,01h           ;nur noch Interrupts am Pin INT0 zulassen
                00019 ;         0UT0    (34h),A         ;Ausgabe auf den INT/TRAP-Port
4214 ED         00020           DB      0edh,39h,34h    ;(Hex-Entsprechung des Befehls)
4217 3E00       00021           LD      A,00h           ;oder anderer Wert zwischen Ox und Fx
                00022 ;         OUT0    (32h),A         ;Zahl der WAIT-Zyklen auf DCNTL ausgeben
4219 ED         00023           DB      0edh,39h,32h    ;(s. o.)
421C C32040     00024           JP      402dh           ;DOS-Entry, dort steht z. B. auch EI
                00025
                00026 bank                              ;Platz für die Banking-Routine des HD
                00027
4200            00028           END     start
00000 Fehler
bank    421F    entry 4210      spbuff 0003     start 4200
                00001 ; Listing 7: Der vektororientierte Ansprung eines Programms
                00002
                00003 ;Voraussetzung: Zuvor wurden von einem Job-Programm einige Routinen ir
                00004 ;gendwo ab 4.0000 abgelegt, um den Z80-Speicher davon zu entlasten.
                00005 ;Sie können auch im EPR0M fertig vorliegen, müssen dann aber im RAM mit
                00006 ;OUT (0FBh),A beginnen.
                00007 ;Ihre Entrypoints, an denen z. B. zuerst das evtl. notwendige Banking
                00008 ;stattfinden könnte, sind in der Tabelle vectab (7.F200) der Reihe nach
                00009 ;abgelegt worden.
                00010 ;Hier soll nun das Programm Nr. 4 etwas berechnen. Später holt das Progr.
                00011 ;Nr. 2 die Ergebnisse in den Z80-Speicher, die nicht in den für Resultate
                00012 ;reservierten Raum ab 7.F800 gepaßt hätten.
                00013
0003            00014 spbuff    EQU     0003h           ;TRS-80 u. ä.: auf freie RAM-Adresse ändern!
                00015
5200            00016           ORG     5200h
5200 210402     00017 start     LD      HL,0204h        ;Signalbyte 02, Programm Nr. 4 (ab Nr. 0)
5203 E5         00018           PUSH    HL              ;auf den Stack für den HD 64180
5206 ED730300   00019           LD      (spbuff),SP     ;Stackpointer für den HD ablegen
5208 D3FB       00020           OUT     (0fbh),A        ;die Buskontrolle abgeben
                00021
                00022 ;Pause, bis der HD im Prg. Nr. 4 die Kontrolle über den Bus zurückgibt
                00023
```

```
                00024 ;         CALL    WICHTIG         ;dann z. B. Bedeutendes vollbringen ...
                00025
520A D1         00026           POP     DE              ;altes Signalbyte (Registerpaar egal)
520B 1E02       00027           LD      E,02h           ;Programm Nr. 2 (MSB ist noch richtig)
520D E5         00028           PUSH    HL              ;Anforderung für den HD auf den Stack
520E D3FB       00029           OUT     (0fbh),A        ;wie oben (korrekter SP noch in 7FFE)
                00030
                00031 ;wieder ein paar Millisekunden Pause ..., dann geht es irgendwie weiter.
                00032
5200            00033           END     start
00000 Fehler
spbuff 0003     start 5200
```

Es lebe der Standard!

Hercules Graphic Adapter meets Trash 80

Helmut Bernhardt

Genau besehen hat es auch Vorteile, daß der Industriestandard einen so niedrigen Standard hat. Insbesondere die Hercules-Karte, die dort schon nicht unbedingt als schlechteste Video-Karte gilt, ist schon fast zum TRS80 kompatibel. Ja, nicht ganz zu dem, aber doch schon zum GenieIIIs-Video-Interface. Wie im GIIIs gibt es dort eine 2K große Text-Video-Seite (80x25 Zeichen) und ebenso memory mapped. Zusätzlich bietet die Hercules aber auch noch einen 2K großen Attribut-Speicher, womit sich für jedes Zeichen auf dem Bildschirm einzeln Attribute setzen lassen. Und schließlich wird auf der Hercules wie im GenieIIIs mit 2 HRG-Pages mit je 32K Größe (ebenfalls bei beiden memory mapped) Grafik dargestellt. Bei beiden Geräten wird alles von einem CRTC6845 gesteuert.

Die maßgeblichste Inkompatibilität ist die Tatsache, daß bei der Hercules Text- und Attribut-Seicher ineinander verschoben sind. Dort werden innerhalb eines lineren 4K-Bereichs alle geradzahlien Adressen für Textzeichen und alle ungeradzahligen Adressen für die dazugehörigen Attribute verwendet. Weniger störend sind die 10Bit-Portadressen des Industrie-Standards, mit denen die internen Register der Karte angesprochen werden. A8 und A9 kann man der Karte bei Z80-Portzugriffen einfach vorgaukeln.

Folgende Tabelle stellt die Ähnlichkeiten und Abweichungen zwischen Hercules und GenieIIIs-Video-Interface zusammen:

Funktion	**Hercules**	**GenieIIIs**
Text-Speicher	B0000-B0FFF (mit A=0)	3800(3C00)-3FFF
Attribut-Speicher	B0000-B0FFF (mit A=1	nicht vorhanden
HRG-Seite 0	B0000-B7FFF	8000-FFFF
HRG-Seite 1	B0000-BBFFF	8000-FFFF
CRTC6845-Ports	3B4H/3B5H	F6H/F7H

Diese Ähnlichkeit ließ mich vor längerer Zeit schon mal darüber brüten, wie man die Hercules anstelle des dort recht lausigen Video-Interface in TRS80 und Genie-Rechnern einsetzen könnte. Zunächst dachte ich an einen Adapter, der mir die nötige Adreßübersetzung macht und die Verbindung zwischen Genie-CPU-Board und Hercules herstellt. Dann dachte ich daran, daß sich dieser Aufwand für das ebenfalls recht lausige Genie-CPU-Board nicht lohnt. Inzwichen ist um diesen reinen Adapter ein ganzes CPU-Baord gewachsen, das noch einiges mehr bietet als das Genie-CPU-Board und dabei auf die Maße einer Europa-Karte geschrumpft ist. Das Board enthält all das, was mir bislang lieb geworden ist:

[1] H64180-CPU
[2] 2 serielle Schnittstellen
[3] 1MB RAM bestehend aus 8 mal 511000-Megachips
[4] auf 1MB erweiterte MMU des HD64180
[5] 27128-EPROM mit Level II BASIC und Initialisierungen
[6] ECB-Bus mit Buspufferung und IM2- und DMA-fähiger Steuerung
[7] Interface zur Hercules
[8] komfortables Switching für memory mapped I/O

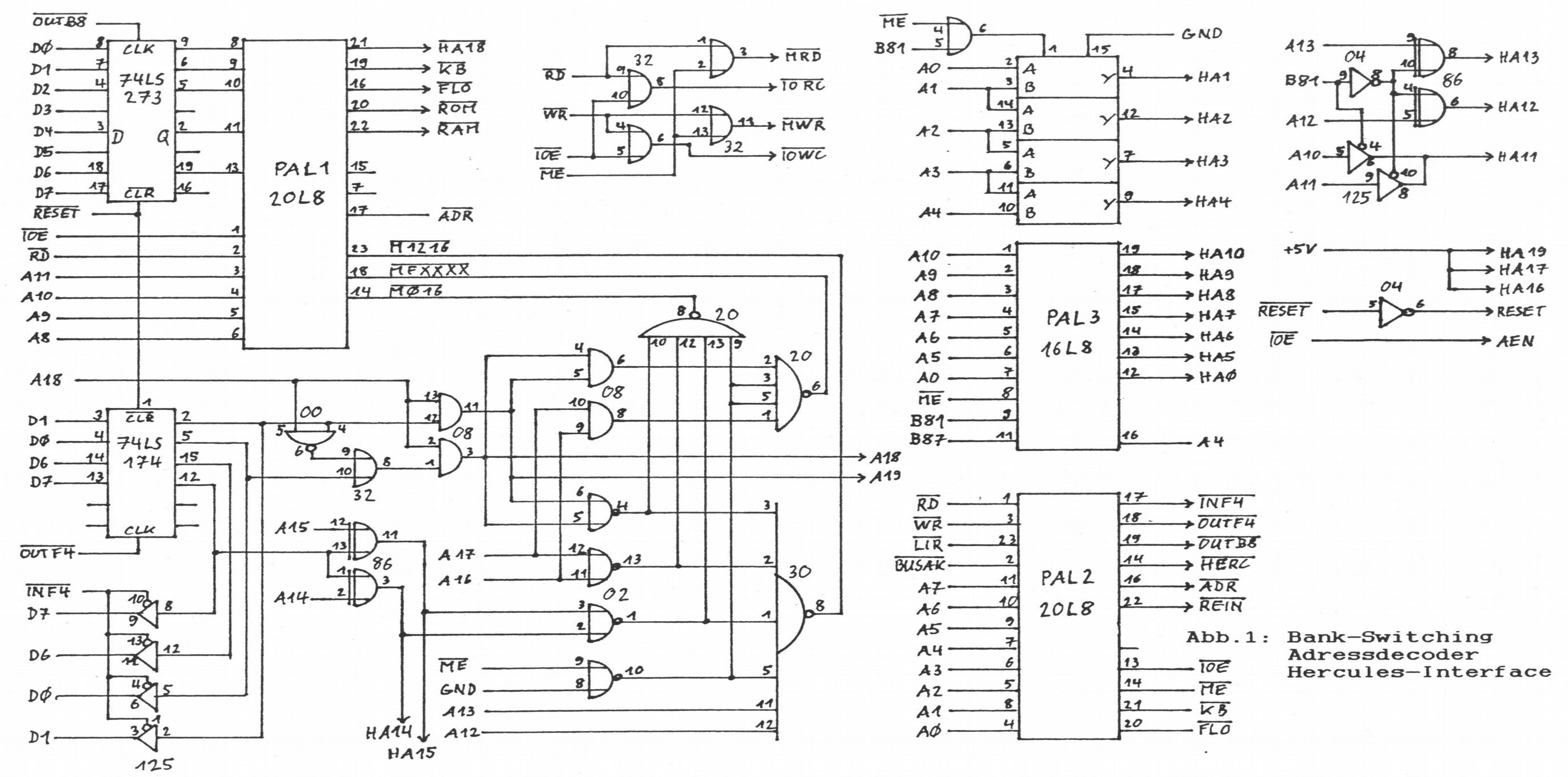


Abb.1: Bank-Switching Adressdecoder Hercules-Interface

Bank-Switching, Adressdecoder, Hercules-Interface)

Damit steht mir ein voll Genie1-kompatibler (nur Spiele lassen sich ohne Bauklötzchengrafik nicht so richtig spielen) Computer zur Verfügung. Die Hercules stellt mir dafür den Standard-64x16-Bildchirm, eine 80-Zeichen-Karte und zwei hochauflösende Grafikseiten mit je 720x348 Punkten zur Verfügung, und das alles für 10,-DM.

Die eigentlichen Kosten fallen bei der Beschaffung der RAMs an. Zugegeben - 256K hätten's beim Genie auch getan, aber das Teil-Layout mit CPU, RAMs, EPROM, RAM-Steuerung und Treibern für die V24-Schnittstellen brauchte ich nur aus der Schublade zu ziehen und beim Rest habe ich mir dann auch keine große Mühe mehr gemacht. So ist

Der Aufbau der Karte

zu einem richtigen kleinen Abenteuer geworden. Das Layout beschränkt sich auf einseitge Leiterbahnführung und versucht, dort so viele Verbindungen unterzubringen, wie irgend möglich. Nur im hinteren Teil der Karte (der aus der Schublade), war es möglich, alles mit Drahtbrücken zu erledigen - bis auf das EPROM, das muß auch von Hand verdrahtet werden, dort sind nur D0-7, +5V und GND angeschlossen. Was von Hand zu verdrahten ist, kann aus der Verdrahtungsliste ersehen werden.

Wer nun immer noch glaubt, daß er dieses Projekt durchstehen kann, und von mir zum Selbstkostenpreis ein geätztes, ungebohrtes Board bekommen hat, muß zunächst mal 1-2 Stunden mit der Bohrmaschine zubringen. Die 3 Stecker für VG64-Leiste, für die Pfostenfeld-Leiste zum Aufstecken der Hercules-Karte und die Pfostenfeld-Leiste für die seriellen Schnittstellen müssen 1mm-Bohrungen erhalten, alle anderen Bohrungen sind in 0,8mm niederzubringen.

Wenn man sich davon wieder erholt hat, legt man sämtliche Drahtbrücken (Drahtbrücken- und Bestückunpsplan) und fängt dann erst an, Kleinteile und IC-Sockel einzulöten. Da die Drahtbrücken teilweise recht eng liegen, sollte grundsätzlich dünner Kuper-Lackdraht verwendet werden. Bei Drahtbrücken, die +5V und GND führen sollen (solche Brücken enden meistens an etwas breiteren Leiterbahnen), ist etwas dickerer isolierter Schaltdraht vorzuziehen.

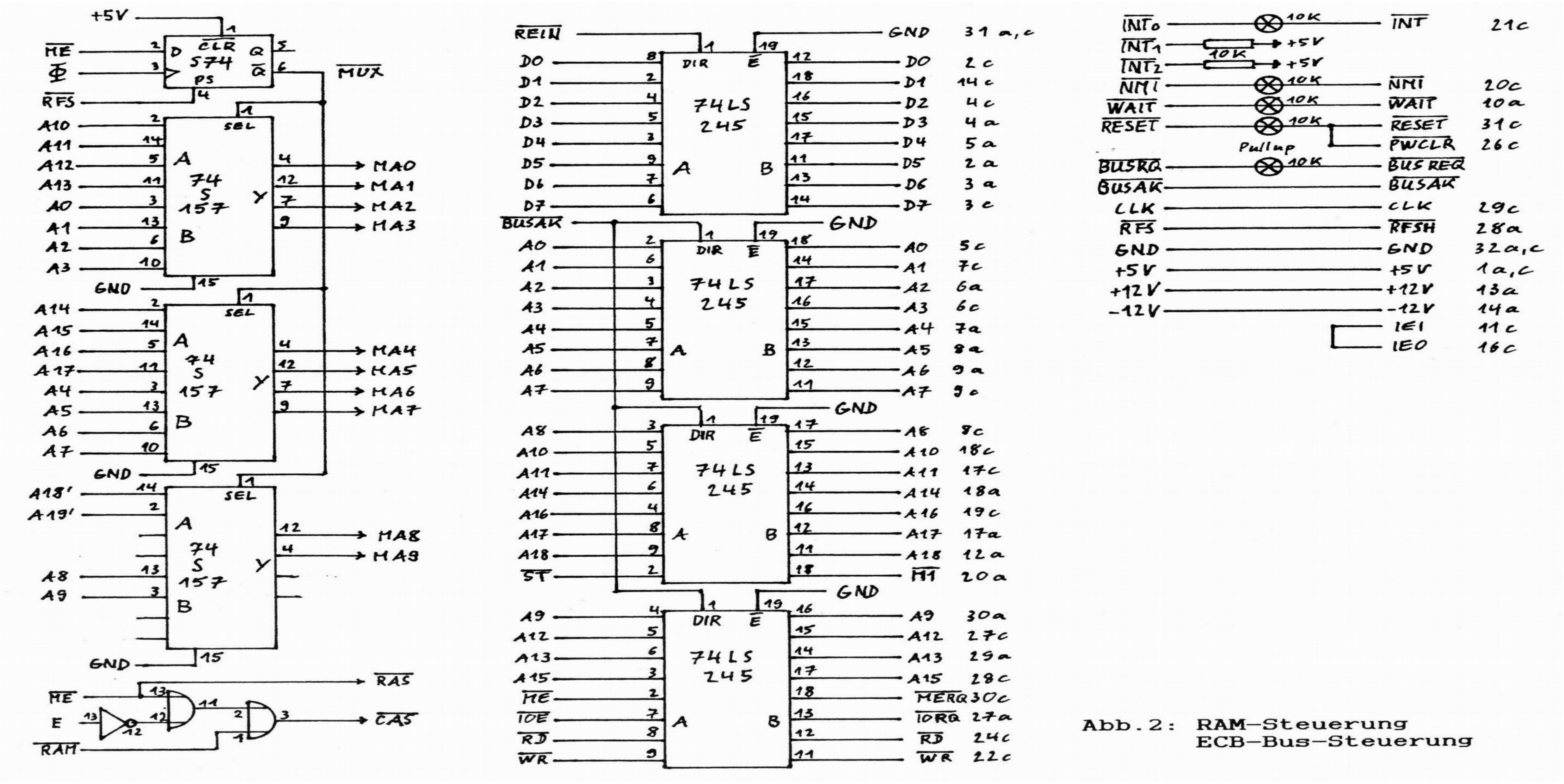
Es sei noch gesagt, daß im dichter besiedelten vorderen Bereich (beim Busstecker) in der ersten Reihe ein IC nicht zu bestücken ist. Weil der Platz halt da war, habe ich da vorsichthalber mal

die Lötaugen hinterlegt, damit es spätere Bastler bei Erweiterungen einfacher haben.

Auf der Busplatine oder frei improvisiert auf dem CPU-Board müssen die CPU-Signale /BUSRQ, /RESET, /WAIT, /MMI und /INT0 mit 10kOhm an +5V gelegt werden. /INT1 und /INT2 müssen direkt am HD64180 über 10k an +5V gelegt werden. Wenn diese Eingänge nicht genutzt werden, genügt ein gemeinsamer Widerstand.

Die 62polige Steckleiste zum Anschluß der Hercules ist für eine spezielle, besonders kleine Karte gedacht, die direkt neben dem für PC-Karten üblichen Platinenrand-Stecker zwei Reihen Durchkontaktierungen aufweist, die freundlicherweise im RM2,54 Rastermaß angelegt sind und eine Weite von 1 mm haben. Wenn hier von der Bestückungseite eine zweireihige Buchsenleiste eingelötet wird, und auf dem CPU-Board die Pfostenfeld-Leiste mit etwa um 1cm verlängerten Beinchen eingelötet wird, können beide Karten direkt aufeinander gesteckt werden.

Der Fernseher darf leider nicht der alte vom Genie1 bleiben. Die Hercules benötigt einen TTL-Monitor (ohne Gehäuse und Netzteil bekommt man sowas im Elektronik-Ramsch-Versandhandel für ca. 40,-DM). Andererseits gibt es auch TTL/BAS-Wandler, denen man aber ein aus "Video" und "Intensity" zusammengeführtes Signal anstelle des reinen Video-Signals zuführen sollte, um nicht auf die Helligkeitssteuerung der Hercules verzichten zu müssen.



RAM-Steuerung, ECB-Bus-Steuerung

Programmierung der Hercules-Karte

Interne Ports (die Portdressen sind die auf die 8Bit-Decodierung durch das Interface reduzierten Adresen; im PC müssen zusätzlich A8 und A9 high sein)

B4H Adressregister des CRTC6845
B5H Datenregister des CRTC6845
B8H Display-Betriebsart-Steuerkanal
B9H Display-Status-Kanal
BBH Light-Pen Flip Flop zurücksetzen

BCH   Drucker-Datenort
BDH   Drucker-Statusport
BEH   Drucker-Steuerkanal
BFH   Konfigurations-Schalter

Für die Programmierung des CRTC6845 ist ein Datenblatt erforderlich. Durch Trial-and-Error Methoden soll man es schaffen können, den Monitor dauerhaft zu dissablen. Es sei nur soviel gesagt, daß der CRTC eine ganze Menge Datenregister besitzt, wovon nur die ersten 16 relevant sind. Sie haben alle die Adresse B5H. Auf welches dieser Register jeweils über Port B5H zugegrifen werden kann, bestimmt der Eintrag in das Adressregister B4H. Die Initialisierungsroutine im ROM enthält an der Adrese 3000H 5 Tabellen mit je 16 Daten für die ersten 16 Datenregister des CRTC und eine Routine, die dem CRTC diese Daten ein- verleibt. Wenn HL auf den Anfang der jeweiligen Tabelle zeigt. Dort mag sich der Softi informieren, der lieber noch exotischere Bildschirm-Formate einstellen möchte. Die grundsätzliche Bedeutung der CRTC-Register ist aus mc 1/88 S.113 zu entnehmen.

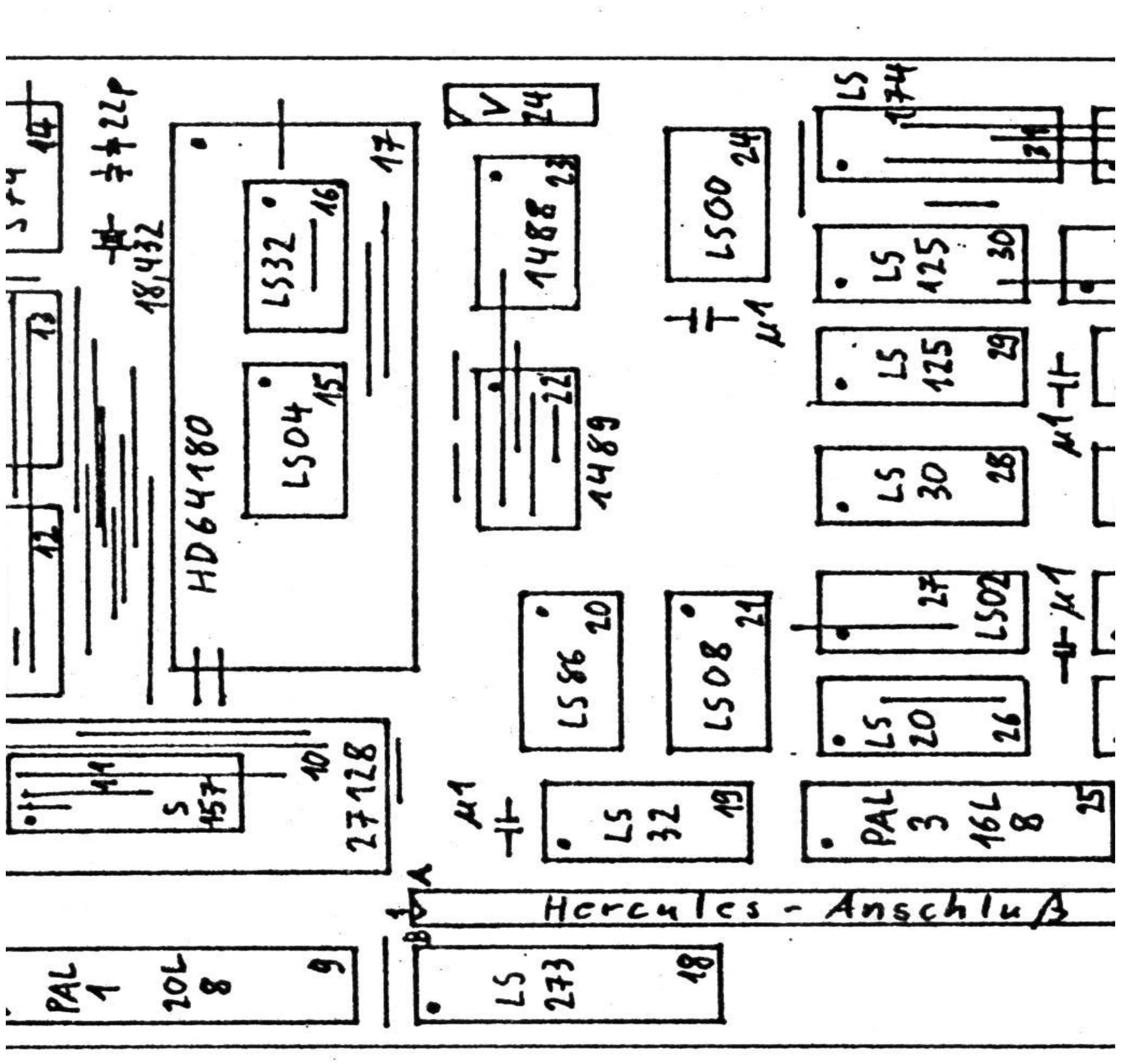
Der Display-Betriebsart-Steuerkanal (Port B8H) der Hercules reagiert nur auf die Bits 1, 3, 5 und 7. Die geradzahligen Datenbits haben dort keine Funktion. Da die Einstellungen dieses Registers für die Adreßübersetzung des Interface zur Hercules von Bedeutung sind, wurde dieser Port auf dem CPU-Board nachgebildet und um die geradzahligen Bits bereichert. Setzen und Zurücksetzen einzelner Bits dieses Ports durch Auslesen, Bit verändern und Zurückschreiben ist also nicht möglich. Man muß dafür im RAM über den Zustand des Ports Buch führen.

In der folgenden Tabelle sind die Funktionen der einzelnen Bits des erweiterten Ports B8H zusammengestellt. Die urgsprünglichen Funktionen des Ports auf der Hercules, die nach wie vor gelten, sind mit # markiert.

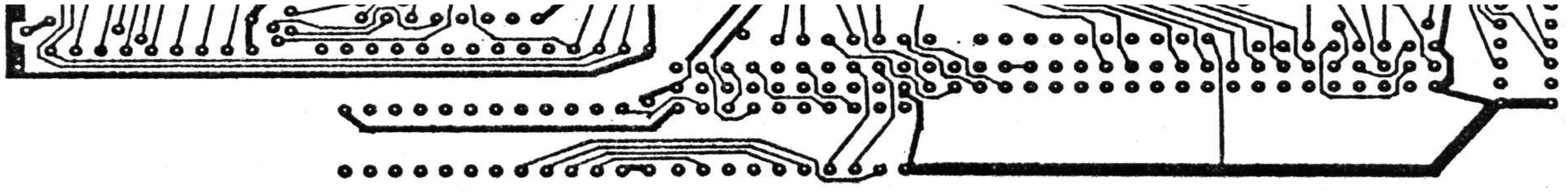
Display-Mode-Control-Port, B8H

D0	=0:	1K Video-RAM bei 03C00-03FFF (64x16), Keyboard bei 3800-38FF	
	=1:	2K Video-RAM bei 03800-03FFF (80x25), Keyboard disabled	
D1	=0:	Text-Video-Mode enabled, HRG disabled	#
	=1:	HRG enabled, Text-Video disabled	#
D2	=0:	ROM bei 00000-036FF (nur bei /RD, bei /WR RAM)	
	=1:	ROM disabled, bei 00000-036FF liegt RAM	
D3	=0:	die Hercules gibt kein Video-Signal an den Monitor	#
	=1:	das Video-Signal der Karte gelangt zum Monitor	#
D4	=0:	Floppy, Drucker u.s.w. bei 037EX memory mapped	
	=1:	037EX ist RAM, Floppy u.s.w. über Ports EXH	
D5	=0:	D7=1 des Attribut-Bytes bedeutet "heller Hintergrund"	#
	=1:	D7=1 des Attribut-Bytes bedeutet "Blinken"	#
D6	=0:	Text- bzw HRG-RAM sind freigegeben	
	=1:	Text- bzw HRG-RAM gesperrt, dort RAM eingblendet	
D7	=0:	HRG: Page 0 selected	#
		Text: 2K Textspeicher eingestellt	
	=1:	HRG: Page 1 selected	#

Text: 2K Attributspeicher eingstellt



Bestückungs- und Drahtbrücken-Plan



Teil-Layout (von der Bestückungsseite gesehen)

Die Einteilung des 1MB-Speichers

Der HD64180 (in der hier verwendeten DIL64-Form) kann nur 512K Speicher adressieren. Um den vollen 1MB-Speicher zu nutzen, muß durch zwei weitere Bits eines Ports (D0 und D1 von Port F4H) ein Banking von 256-K-Blocks dem Banking der MMU des HD64180 überlagert werden. Die Selektion eines der 4 256K-Blocks erfolgt mit A8 des HD64180 und D0/D1 von Port F4H. Aus diesen 3 Signalen werden die Adressen A18 und A19 für das RAM hergeleitet.

Port F4H		**CPU**	**RAM**		**256K-**
D1	**DO**	**A18**	**A19**	**A18**	**-Block**
X	X	0	0	0	0
0	0	1	0	1	1
0	1	1	0	1	1
1	0	1	1	0	2
1	1	1	1	0	3

Wenn A18 der CPU low ist, wird immer der unterste 256K-Block angewählt. Wenn A18 high ist, wird einer der 3 anderen Blocks selektiert. Welcher djeser 3 Blocks selektiert wird, hängt von den Zutänden von D0 und D1 des Ports F4H ab. Diese Vorgaben gelten sowohl für die MMU als auch für den DMA-Controller.

Nach einem RESET ist die MMU des HD64180 immer so eingetellt, daß der logische 64K-Z80-Adreßraum in den unteren 64K des unteren 256K-Blocks liegt. Um die Standard-Software (Newdos/80, CP/M) zu fahren, braucht die MMU nicht umrogrammiert zu werden. Die Decodierung von RAM und memory mapped I/O schließt auch die Adressen A16-A19 mit ein, so daß diese Baugruppen im Physikalischen Adressraum 00000H-03FFFH liegen. Nur der HRG-Speicher der Hercules-Karte mit 2mal 32K wäre bei 08000H eingeblendet etwas lästig und wurde deshalb an das obere Speicherende bei 0F0000H verbannt. Für die MMU und den DMAC des HD64180 sind dies die Adressen 70000H-7FFFFH, wobei D0 und D1 von Port F4H gesetzt sein müssen, damit der HRG-Speicher adressiert wird.

Neben der Grundeinstellung (D7 an Port F4H low, Zustand nach RESET) gibt es noch eine andere Lage für die memory mapped Baugruppen. Wenn 7=1 an Port F4H ausgegeben wird, liegen ROM, Floppy, Drucker, Tastatur und Text-Video-RAM bei 0C000H-0FFFFH und die untersten 16K sind RAM.

Und schließlich läßt sich innerhalb dieser 16K (wo auch immer die memory mapped Baugruppen eingeblendet sind) sehr komfortabel umkonfigurieren, darüber gibt die Tabelle der Bedeutung der Bits des erweiterten Ports B8H und die Map der Speipcheraufteilung Auskunft. In allen Bereichen, wo die jeweilige memory mapped Baugruppe ausgeblendet ist, liegt dann automatisch RAM vor.

Der Text-Bildwiederholseicher

Die Hercules-Karte stellt im Text-Modus einen 2K großen Bildwiederholspeicher für Text und einen 2K großen Attribut-Speicher zur Verfügung. Diese Speicher sind ineinander verschoben; innerhalb eines linearen 4K-Bereichs stellen die geraden Adressen den Text-Speicher und die ungeraden Adressen den Attribut-Speicher dar.

Video-Attribute waren aber damals, als der TRS-80 geboren wurde, noch unbezahlbarer Luxus. Deshalb scheint der Attribut-Speicher der Hercules der Kompatibilität zum TRS-80 im Wege zu stehen. Der TRS-80 braucht einen zusammenhängenden Bildwiederhol-Speicher von 1K Größe (und manche jüngeren Brüder können auch mit 2K Textspeicher umgehen).

Dieses unlösbare Problem läßt sich dadurch lösen, daß beim Text-Modus die Aressen A0-A10 der CPU (2K-Bereich) auf die Adressen A1-A11 der Hercules-Karte gelegt werden. Diese Umsetzung darf aber nur bei Zugriff auf den Text- und den Attribut-Speicher erolgen. Bei HRG-Betrieb und bei Zugriff auf die Ports der Karte müssen alle Adressen direkt durchgeschaltet werden. Die für das Selektieren von Text- oder Attribut-Speicher zuständige Adresse A0 (zur Hercules) wird im Text-Modus von D7 des erweiterten Ports B8H und bei Zugriff auf Hercules-Ports und HRG-Speicher von A0 der CPU bedient. Diese Adressteuerung wird von PAL3 und einem 74LS157-Multiplexer erledigt.

Die damit auf 2K reinen Text-Speicher (und nach Umschalten 2K reinen Attriut-Speicher) beschränkte Bildschirmseite ist aber für den TRS-80, der nur 1K bei 3C00H dafür vorsieht, immer noch zu groß. Deshalb kann mit D0=0 an Port B8H (Einstellung nach der Initialisierung im ROM) die Freigabe nur der oberen Hälfte des Text-Speichers im Bereich C000-3FFF vorgegeben werden. Für das TRS-80-Format 64x16 Zeichen muß dann aber auch der CRTC-6845 entsprechend programmiert werden. Dies wird ebenfalls durch die Initialisierung im ROM erledigt. Das Betriebssystem findet dann beim anschließenden Booten eine voll TRS-80-kompatible Maschine vor.

Wenn D0=1 an Port B8H ausgegeben wird, ist der volle 2K-Text-Speicher im Bereich 3800-3FFF verfügbar. Um eine vernünftige Darstellung auf dem Bildschirm zu haben, muß dann aber auch der CRTC6845 entsprechend initialisiert werden. Jm 80x25-Modus ist die Tastatur dann aber nicht mehr zugänglich. Für das Aulesen der Tastatur muß dann immer erst in den 64x16-Modus zurückgeschaltet werden und nach dem Auslesen der Tastatur für weitere Bildschirm-Ausgaben wieder der 80x25-Modus eingestellt werden. Wenn dabei die Register des CRTC-6845 in Ruhe gelassen werden und nur mit D0 von Port B8H geschaltet wird, bleibt die Darstellung auf dem Bildschirm unbeeinträchtigt. Die Umschaltung betrifft dann nur die Zugriffmöglichkeiten der CPU auf den Bildwiederholspeicher. Das für die Darstellung des Bildpeichers auf dem Monitor verantwortliche Auslesen des 6845 wird dadurch nicht belangt.

Die Initialisierungs-Routine im ROM füllt zunächst den gesamten Attribut-Speicher mit dem Standard-Wert 7. Das bedeutet, daß die Zeichen auf dunklem Hintergrund mit normaler Helligkeit dargestellt werden, wenn der Attribut-Speicher bei der Textausgabe nicht berücksichtigt wird.

Es sind allerdings dies die Zeichen entsprechend Big Blue's Verständnis von ASCII-Code. Die Zeichen mit den Codes 5BH-5FH und 7BH-7FH sind grundsätzlich die amerikanischen Zeichen (geschweifte und eckige Klamnern u.s.w.). Den deutschen Umlauten und auch allen anderen nationalen Sonderzeichen sind feste ASCII-Codes mit gesetztem Bit7 zugeordnet. Das hat den Nachteil, daß bei allen von Standard-Programmen unter MEWDOS und CP/M ausgegebenen Texten anstelle der deutschen Umlaute die entsprechenden amerikanischen Zeichen erscheinen. Ein Vorteil besteht aber auch darin, daß amerikanische Sonderzeichen und sämtliche nationalen Umlaute gleichzeitig dargestellt werden können. Für Soft-Freaks besteht hier ein weites Betätigungsfeld in der Anpassung von Tastatur-Treibern und Standard-Programmen.

Die Bildchirm-Attribute

Wenn mit D1=1 und D7=1 an Port B8h die Text-Betriebsart und der Attributspeicher eingeschaltet werden, kann im Adressbereich des Text-Speichers ein Attribut-Byte eingetragen werden, das die Darstellung des ASCII-Zeichens auf gleicher Adresse im Text-Speicher beeinflußt. Die Attribut-Bits wirken nur auf das eine Zeichen und nicht auf den gesamten Bildschirm. Wenn mit Attributen gearbeitet werden soll, muß für jedes Zeichen im Text-Speicher auch ein Byte im Attribut-Speicher eingetragen werden. Beim Scrollen muß auch der Attribut-Speicher genauo wie der Text-Speicher verschoben werden.

D7: die Wirkung von D7 eines Attribut-Bytes hängt davon ab, welcher Wert für D5 an Port B8H ausgegeben wurde.

Port B8H, D5=0:

D7(Attribut)	=0:	Zeichen erscheint auf dunklem Hintergrund
	=1:	Zeichen erscheint vor halbhellem Hintergrund

Port B8H, D5=1:

D7(Attribut)	=0:	kein blinkendes Zeichen
	=1:	Zeichen wird blinkend dargestellt
D3	=0:	normale Helligkeit der Darstellung
	=1:	Highlighting

D0-D2, D4-D6

D6	D5	D4	D2	D1	D0	Bedeutung
0	0	0	0	0	0	Zeichen nicht dargestellt
0	0	0	0	0	1	unterstrichen dargestellt
0	0	0	1	1	1	normale Darstellung
1	1	1	0	0	0	invere Darstellung

Die HRG der Hercules-Karte

Mit 720x348 Punkten Auflösung und 2 Seiten läßt sich schon sehr schöne Grafik auf die Mattscheibe bringen. Diese Auflösung benötigt natürlich auch Speicherraum für die beiden 32K-Pages. Diese Bereiche analog zum GenieIIIs einfach in den logischen Adreßraum der nach RESET eingestellten (unteren physikalischen) 64K zu legen, erscheint etwas unzweckmäig zu sein. Es müßte dann eine Hälfte des ohnehin nur 64K großen, direkt erreichbaren Arbeitspeichers ausgeschaltet werden. Für Grafik-Anwendungen müßte man sich festlegen, in welcher Hälfte der 64K die Grafikroutinen laufen sollen.

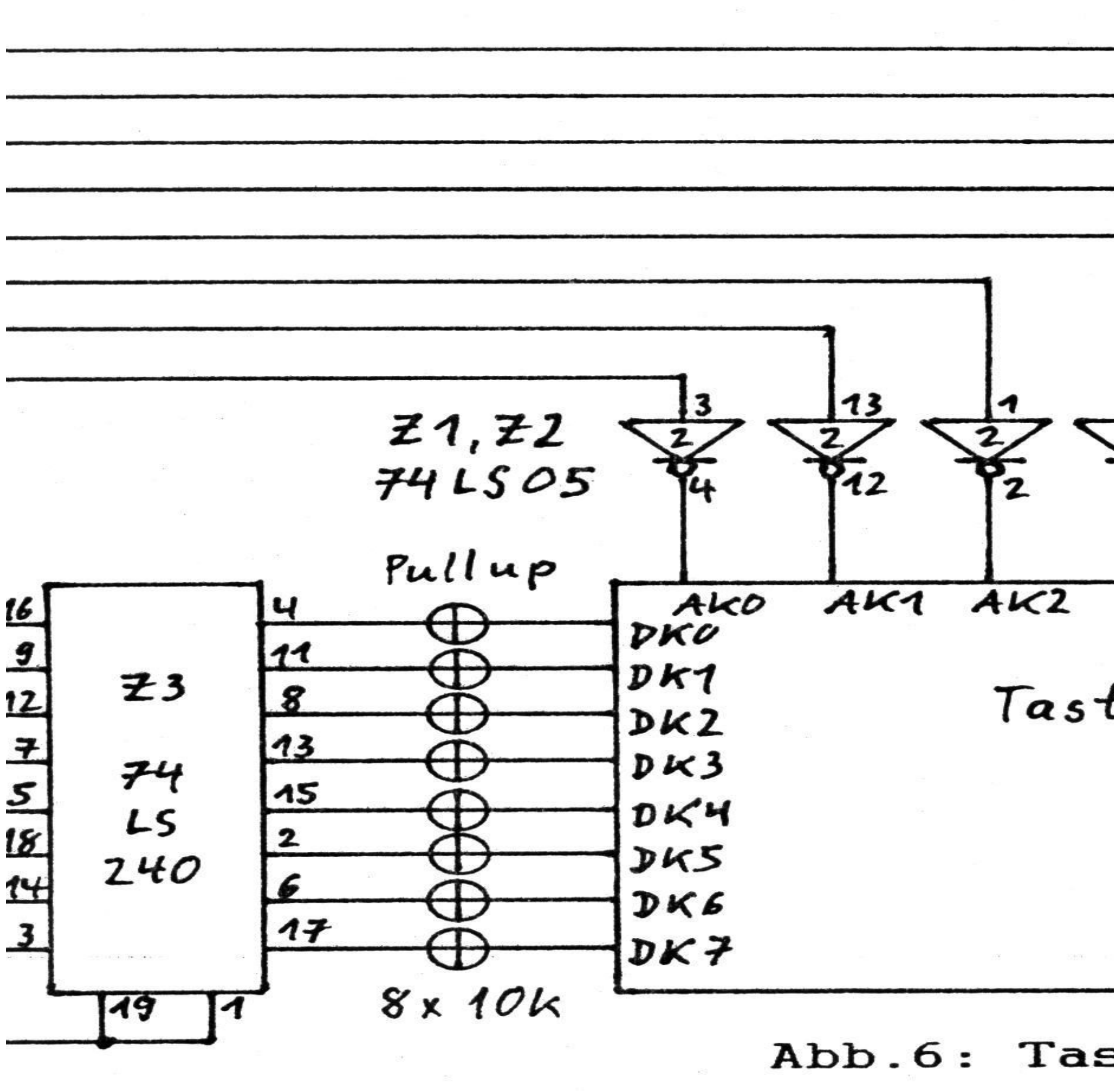
Hier ist der HRG-Speicher ans obere Ende des physikalischen 1MB-Speichers verbannt worden. Das Besudeln des Grafik-Seichers erfolgt im einfachsten Fall durch blockweise Übertragung per DMA. Wer aber trotzdem gerne dirrekt in das HRG-RAM schreiben möchte, kann sich dieses durch die in die untere oder obere Hälfte seines 64K-Arbeitsspeichers einblenden.

Nach einem RESET ist die Text-Betriebsart eingstellt. Mit D1=1 an Port B8H läßt sich auf die HRG-Betriebsart "Half-Mode" umstellen, wenn in Port BFH D0=1 eingetragen ist. Dabei ist die untere Hälfte (Page 0) erreichbar. Um auch die Page 1 erreichen zu können, muß auch D1=1 an Port BFH ausgegeben werden.

Während der Grafik-Betriebsart ist der Text-Bildspeicher nicht zugänglich. Da dieser aber physikalicher Bestandteil der Page 0 der HRG ist, wird der Inhalt des Textspeichers bei Grafikausgaben an Page 0 überschrieben. Gleiches gilt für den Attribut-Seicher.

Die Tastatur

Dieses Gerät erleidet hier die größten Einbußen in der Präsenz. So ist der Bereich, in dem die Tastatur ausgelesen werden kann, auf den dafür nötigen Platz von 256 Adressen bechränkt. In der 64x16-Text-Betriebsart (D0=0 an Port B8H, die einzige Einstellung, in der die Tastatur erreichbar ist) stehen bei 3900H-3BFFH ein paar Byte frei verfügbaren Speichers bereit. Auch der Speicher parallel zur Tastatur kann bei eingeblendeter Tastatur beschrieben werden; Auslesen läßt er sich nur, wenn auf Grafik-Betriebsart umgestellt wird.



Beim Arbeiten im 80x25-Textmodus ist die Tastatur nicht erreichbar. Dann liegt ab 3800H Bildwiederhol-Speicher vor. Um die Tastatur zwichendurch auszulesen, muß mit D0=0 an Port B8H vorübergehend auf 1K Video-RAM umgeschaltet werden. Da das aber nicht den Zugriff des CRTC6845 auf den Bildspeicher beeinflußt, zeigt sich das Umschalten nicht in der Darstellung auf dem Bildschirm.

Wenn es auch Wahnsinn ist, eine Tastatur memory-mapped als offene Matrix an einen Computer anzuschließen, weil bei höheren Taktraten und längeren Anschlußleitungen Probleme zu erwarten sind, kommt man bei einem TRS-80-kompatiblen Rechner um diesen Unsinn nicht herum, wenn auch der Betrieb von OLDDOS/80 und der darunter laufenden Programme möglich sein soll.

Ein einfaches Interface (Abb.7) kann auf einem Stück Lochrasterplatine in Fädeltechnik aufgebaut werden und mit einer VG64-Steckerleiste an den ECB-Bus gesteckt werden. Die Tastatur kann daran über ein nicht zu langes, 16poliges Kabel angeschlossen werden.

Die Freigabe des ROMs

Wie es sich für einen Trash-80-komatiblen Computer gehört, liegen nach dem Einchalt-RESET die 12K Löffel-II-BASIC bei 0000H-2FFFH im ROM rum. Zusätzlich wurde hier auch der Bereich 3000H-36FFH als ROM decodiert. Hier sind die nötigen Initialisierungen der HD64180-CPU und der Hercules-Karte untergebracht. Der Einsprung bei 3050H muß im Level-II-ROM anstelle des ersten Sprungbefehls nach RST 00H gepatcht werden. Die Initialisierung endet mit einem Sprung an das dort vorher vorhandene Sprungziel 0674H.

Wenn D7 an Port F4H high ausgegeben wird, verschiebt sich die Lage des ROM-Inhalt von 0000-36FF nach C000-F6FF. Das ist zwar wenig sinnvoll/ weil der Stoff auf dieser Adresslage nicht läuft, ließ sich aber wegen des für CP/M nötigen Verschiebens des memory mapped I/O nach F7E0-FFFF nicht verhindern. Wenn das ROM aber ausgeschaltet wird (D2 an Port B8H high), stört es dort nicht mehr.

Das Abschalten des ROMs über D2 an Port B8H läßt sich natürlich auch in der ursrünglichen Lage bei 00000H durchführen. Da das ROM ohnehin nur beim Lesen freigegeben wird und beim Schreiben RAM eingeblendet ist (auch wenn das ROM nicht abgeschaltet ist) kann das Kopieren des ROM-Inhaltes in das RAM durch ein einfaches LDIR auf sich selbst (HL = DE = 0) erfolgen. Das wird in der Initialisierung vorsichtshalber auch schon mal gemacht.

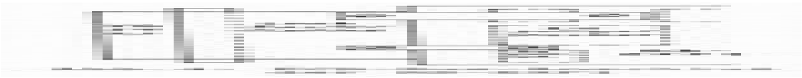
Floppy, Drucker und der ganze Rest

Damit die Abwärtskompatibilität zum TRS-80 beibehalten wird, liegen diese Komponenten memory mapped bei 37EXH. Allerdings wurde die Option eingebaut, anstelle dieser Memory-Adressen auch die entprechenden Ports EOH-EFH für das Ansprechen dieser Einheiten benutzen zu können. Dafür wird mit D4=1 an Port B8H auf Betrieb über Ports umgeschaltet.

Wer von seinem alten Genie oder TRS80 her noch einen RB-EXP1 Controller besitzt, hat damit ein weiterhin brauchbares Gerät vorliegen. Diesem Board muß nur noch ein VG64-Stecker spendiert werden. Dieser Stecker wird auf einen 2cm breiten Strefen Lochraster-Platine gelötet, der wiederum mit zwei schmalen Streifen Pertinax oder Sonstigem Kunststoff mit dem RB-EXP1 verschraubt wird (Abb.6). Auf der Lötseite werden dann die ECB-Signale des Steckers mit den entsprechenden Punkten der beiden Anschlußstecker des EXP1 verbunden.

Anschluß des RB-EXP1 Controllers an den ECB-Bus

ECB-Bus Pin	**Signal**	**RB-EXP1**	
2c	D0	Pin 9	24poliger
14c	D1	Pin 10	Stecker
4c	D2	Pin 11	
4a	D3	Pin 13	
5a	D4	P3n 14	
2a	D5	Pin 15	
3a	D6	Pin 16	
3c	D7	Pin 17	
5c	A0	Pin 8	
7c	A1	Pin 7	
6a	A2	Pin 6	
6c	A3	Pin 5	
1a	+5V	Pin 24	
32a	GND	Pin 12	
13a	+12V	Pin 11	16poliger



15a	-5V	Pin 1	Stecker
21c	/INT	Pin 4	
31c	/ RESET	Pin 5	

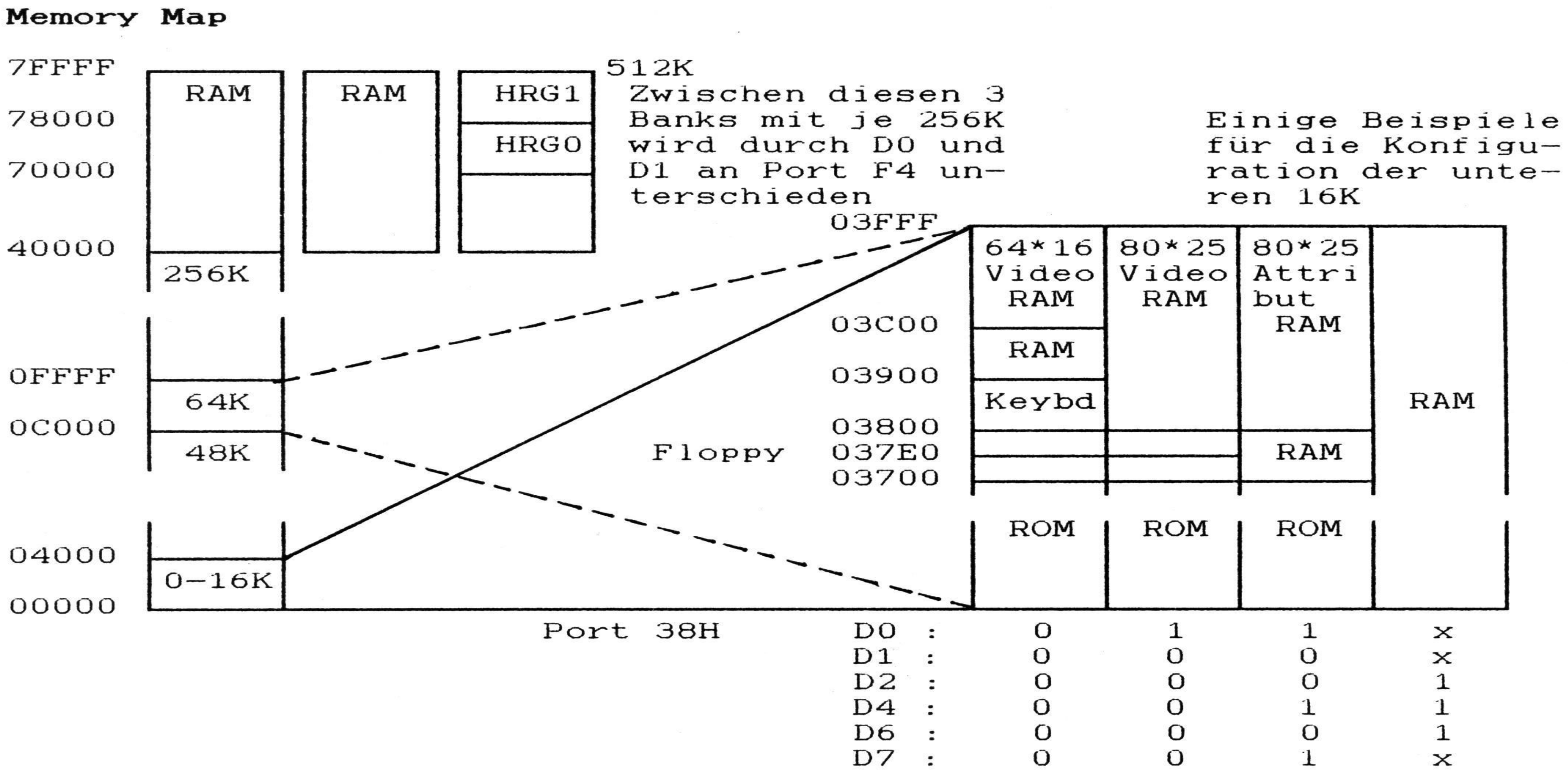
z.B. 23a /FLO Pins 2,14 74LS155

Da das Freigabe-Signal für den Controller bereits auf dem CPU-Board erzeugt wird und auch schon die Umschaltmöglichkeiten beinhaltet, sollte lieber dieses Signal verwendet werden als das auf dem EXP1 hergeleitete Signal /37EX. Dafür wird das Signal /FLO von PAL1 auf dem CPU-Board über eine nicht benutzte ECB-Leitung an den Controller geführt und dort an die Pins 2 und 14 des 74LS155 gelegt. Die beiden 74LS30 beim 16poligen Stecker, die vorher das Signal /37EX und die Drucker-Freigabe über Port FDH geliefert haben, können enternt werden. Die Verwendung des Signals /FLO erübrigt auch die Zuführung der Signale, die in obiger Tabelle nicht angeführt sind und bei Verwendung des EXP1 im Genie oder TR-S80 zusätzlich angeschlossen werden mußten.

RAM-Speicher satt

Grundsätzlich besteht der gesamte physikalische Adreßraum von 1MB aus RAM-Speicher. Nur dort, wo irgendwelche memory mapped I/O-Baugruppen eingeblendet werden, ist der RAM-Speicher nicht erreichbar. Durch Abschalten der mm I/O-Baugruppen wird dort dann automatisch der entsprechende RAM-Bereich eingeblendet.

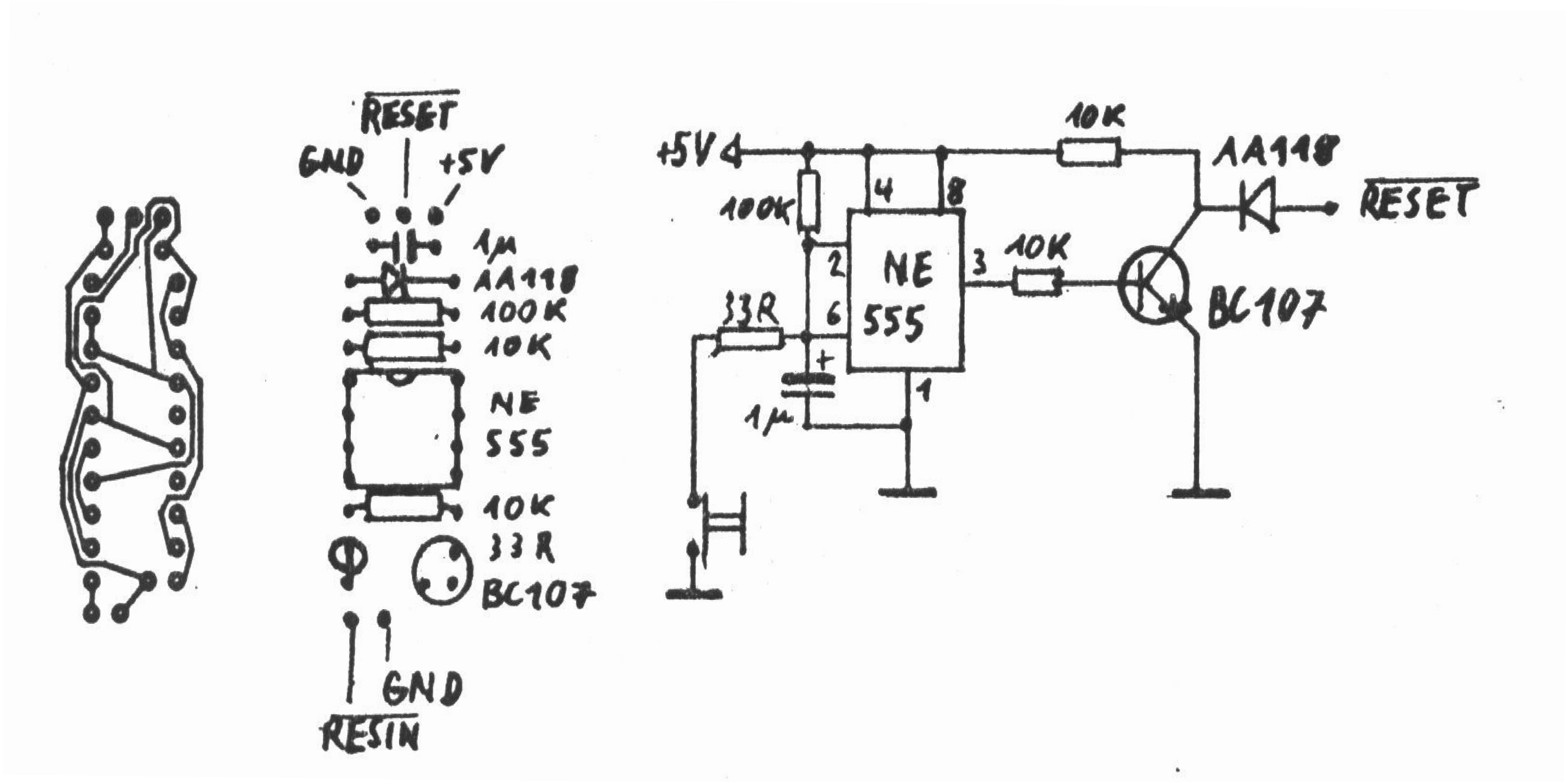
Im Bereich des ROMs (00000-003FF) und der Tastatur (03800-038FF), wenn diese im 64x16-Textmodus erreichbar ist, kann der RAM-Speicher auch beschrieben werden, ohne daß ROM bzw. Tastatur ausgeschaltet werden müssen. Nur zum Lesen dieser RAM-Bereiche müssen diese Einheiten ausgeblendet werden.



Ein entscheidender Nachteil des Industrie-Standards sei zum Schluß noch erwähnt. Um das vom Genie her bekannte Flickern bei Ausgaben auf den Bildschirm zu verhindern, lassen modernere Hercules-Karten einen Zugriff auf das Video-RAM nur zu, wenn der Elektronenstrahl vom rechten zum linken Bildrand zurückläuft und dabei dunkel gesteuert ist. Die CPU wird während der Zeit des Strahlrücklaufs mit einem Wait (/IO_CH_RDY, Pin A10 des PC-Slots) ruhiggehalten. Offensichtlich ist das Wait-Timing bei HD64180 anders als beim 80286 im PC-AT, denn die Hercules macht im Genie nur bis 5MHz Systemtakt mit; bei höheren Takten erscheint nur noch Müll auf dem Fernseher.

Es gibt aber auch Hercules-Karten, die sich wie ein normales Genie-Video-Interface ohne Rücksicht auf Flickern beschreiben lassen, Eventuell läßt sich damit dann der für das CPU-Board als machbar zu erwartende Takt von 9,216 MHz fahren. Wahrscheinlich muß dann aber ein Wait aus dem Freigabesignal /HA18 für die Speicher der Hercules-Karte erzeugt werden, das den Zugriff der CPU um 1 oder 2 Takte verlängert. Da ich eine solche Karte aber nicht habe, will ich darüber auch nicht weiter nachdenken.

Das System verfügt noch nicht über eine Reset-Schaltung. Diese kann im einfachsten Fall durch Kurzschließen von /RESET (ECB-Bus, 31c) mit GND über einen Taster bereitgestellt werden. Da aber während der Zeit, in der der Taster gedrückt ist, kein Refresh der dynamischen RAMs erolgt, gehen bei einem Reset Daten verloren. Eine einfache Schaltung mit einem 555-Timer und einem Taster erzeugt sowohl beim Einschalten als auch beim Betätigen des Tasters ein getimtes /RESET-Signal, das ein längeres Ausbleiben des Refresh verhindert (Abb.5).

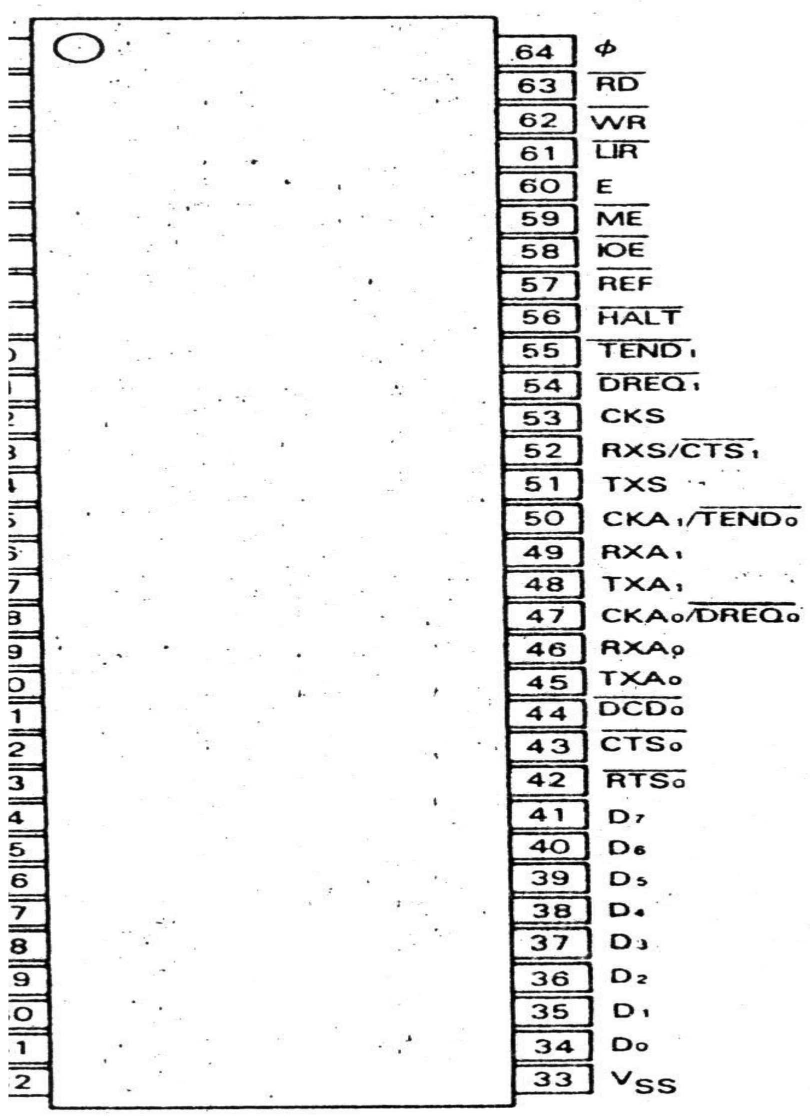


Freie Verdrahtung auf der Lötseite Signal IC,Typ,Pin IC,Typ,Pin ...

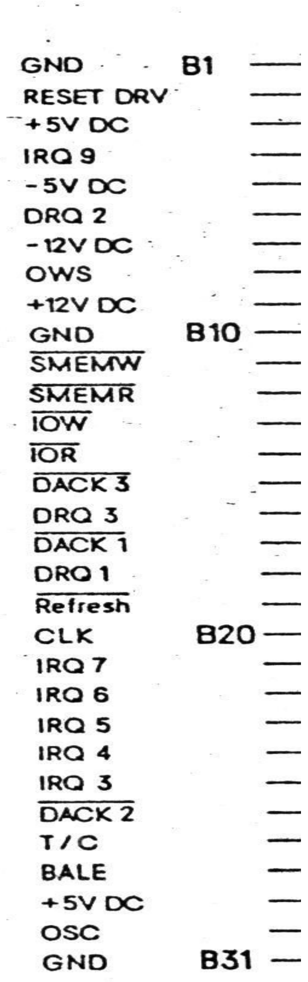
D0 18,273,8 10,27128,11
D1 18,273,7 10,27128,12
D2 18,273,4 10,27128,13; PC-Slot,A7
D3 18,273,13 10,27128,15
D4 18,273,8 10,27128,16; PC-Slot,A5
D5 18,273,14 10,27128,17
D6 18,273,17 10,27128,18
D7 18,273,18 10,27128,19
D0 17,64180,34 30,125,5
D1 17,64180,35 30,125,2
D2 17,64180,36 38,245,4
D3 17,64180,37 38,245,5
D4 17,64180,38 38,245,3
D5 17,64180,39 38,245,9
D6 17,64180,40 30,125,11
D7 17,64180,41 31,174,13
/HALT 17,64180,56 VG64-Leiste,25c
/REF 17,64180,57 VG64-Leiste,28a
/IOE 1,PAL1,1 17,64180,58; 19,32,10; PC-Slot,A10; 33,245,7; 37,PAL2,13
/ME 16,32,4 17,64180,59; 19,32,13; 33,245,2; 37,PAL2,14
/LIR 17,64180,61 37,PAL2,23

/WR   17,64180,62   19,32,12;   33,245,9;   37,PAL2,3
/RD   1,PAL1,2   17,64180,63;   19,32,1;   33,245,8;   37,PAL2,1
CLK   17,64180,64   VG64-Leiste,29c
/WAIT   17,64180,4   VG64-Leiste,10,a;   PC-Slot,A10
/BUSAK   17,64180,5   37,PAL2,2
/BUSRQ   17,64180,6   VG64-Leiste,11a
/RESET   18,273,1   17,64180,7;   15,04,5;   31,174,1;   VG64-Leiste,31c
/NMI   17,64180,8   VG64-Leiste,20c
/INT0   17,64180,9   VG64-Leiste,21c
/ST   17,64180,12   35,245,2
A0   17,64180,13   10,27128,10;   25,PAL3,7;   32,157,2;   38,245,2
A1   17,64180,14   10,27128,9;   32,157,14/3;   36,245,6
A2   17,64180,15   10,27128,8;   32,157,13/5;   36,245,3
A3   17,64180,16   10,27128,7;   32,157,11/6;   36,245,4
A4   17,64180,17   10,27128,6;   25,PAL3,16;   32,157,10;   35,245,5
A5   17,64180,18   10,27128,5;   25,PAL3,6;   36,245,7
A6   17,64180,19   10,27128,4;   25,PAL3,5;   36,245,8
A7   17,64180,20   10,27128,3;   25,PAL3,4;   36,245,9
A8   1,PAL1,6   10,27,128,24;   17,64180,21;   25,PAL3,3;   35,245,3
A9   1,PAL1,5   10,27,128,25;   17,64180,22;   25,PAL3,2;   35,345,4
A10   1,PAL1,4   10,27,128,21;   17,64180,23;   25,PAL3,1;   35,345,5
A11   1,PAL1,3   10,27,128,23;   17,64180,24;   35,345,7
A12   17,64180,25;   10,27128,2;   20,86,5;   28,30,12;   33,245,5
A13   17,64180,26;   10,27128,27;   20,86,9;   20,30,11;   33,245,6
A14   17,64180,27;   20,86,2;   35,245,6
A15   17,64180,28;   20,86,12;   33,245,3
A16   17,64180,29;   12,157,5;   27,02,11;   35,245,4
A17   17,64180,30;   12,157,11;   27,02,12;   35,245,8
A18   17,64180,31;   24,00,5;   21,08,13/2;   35,245,9
A18'   21,08,4;   11,157,14
A19'   21,08,5;   11,157,2 B81   18,273,6;   25,PAL3,9;   29,125,4;   16,32,5;   15,04,9
/B81   18,273,16;   25,PAL3,11
Q0   30,125,5;   16,32,10
Q1   30,125,2;   21,08,12
X   16,32,8;   21,08,1
Y   24,00,6;   15,32,9
/M016   26,20,8;   1,PAL1,14
/M1216   28,30,8;   1,PAL1,22
MFxxxx   26,20,6;   1,PAL1,18
/OUTB8   37,PAL2,19;   9,273,11
/OUTF4   37,PAL2,18;   31,174,9
/INF4   37,PAL2,17;   30,125,10
/ADR   37,PAL2,15;   1,PAL1,17
/ROM   1,PAL1,20;   10,27128,20;   10,27128,22
/FLO   1,PAL1,16;   37,PAL2,19;   VG64-Leiste,z.B.23c
/KB   1,PAL1,19;   37,PAL2,20;   VG64-Leiste,z.B.23a
/RAM   1,PAL1,22;   16,32,1
/HA18   1,PAL1,21;   PC-Slot,A13
HA15   20,86,11;   PC-Slot,A16
HA14   20,86,3;   PC-Slot,A17
HA11   29,125,6;   PC-Slot,A20
RESET   15,04,6;   PC-Slot,B2
SEL   16,32,6;   32,157,1
+5V   22,1489,14;   20,86,14;   17,64180,32;   10,27128,28;   4,RAM,9
+5V   1,PAL1,24;   2,RAM,9
+5V   25,PAL3,20;   PC-Slot,A12/A14/A15/B29
GND   10,27128,14;   20,86,7;   26,20,7
GND   17,64180,33;10,04,7;   V24-Stecker,9/10

+12V 23,1488,14; VG64-Leiste,13a
-12V 23,1488,1; VG64-Leiste,14a



out der HD64180–CPU



;**Initialisierung der Hercules-Karte und der HD64180-CPU**
;**im Sonder-ROM-Bereich ab 3000H**
;
;Diese Routinen werden aus dem ROM heraus angesprungen
;und übergeben zum Schluß wieder an die TRS-80 Boot-
;Routine. Dafür muß das ROM ab 0000H folgendermaßen
;geändert werden:

;vorher			nachher		
0000	DI		0000	DI	
0001	XOR	A	0001	XOR	A
0002	JP	0674h	0002	JP	3050h

;

```
;Die Initialisierung schließt mit JP 0674H ab.
;
attrib      EQU     07h     ;D7=1 nicht blinkend bzw
                            ;normaler Hintergrund
                            ;D6-D4=0,  D2-D0=1
                            ;normale Darstellung
mode        EQU     0       ;D7=0   HRG-Page 0
                            ;D5=1   Blinken erlaubt
                            ;D3=1   Video-Signal aktiv
                            ;D0=0   Text-Betriebsart
config      EQU     3       ;D0=1   HRG erlaubt
                            ;D1=1   HRG-Page 2 enabled
;
;
            Org     3000h   ;Anfang Sonder-ROM
;
;Initialisierungdaten für den CRTC6845
;64*16-TRS80-Standard-Einstellung
tab         DEFB    6eh,40h,50h,0ah,14h,06h,10h,12h
            DEFB    02h,0eh,29h,09h,04h,00h,00h,00h
;80*25-Modus
            DEFB    6eh,50h,58h,0ch,1fh,02h,19h,1bh
            DEFB    02h,09h,29h,09h,00h,00h,00h,00h
;64*24-Modus
            DEFB    6eh,40h,50h,0ah,1bh,09h,18h,19h
            DEFB    02h,0ah,29h,09h,02h,00h,00h,00h
```

```
;64*32 interlaced
        DEFB    6eh,40h,50h,0ah,13h,02h,10h,11h
        DEFB    03h,0fh,2fh,0fh,00h,00h,00h,00h
;92*22-Modus
        DEFB    70h,5ch,61h,08h,1dh,08h,16h,19h
        DEFB    02h,0bh,29h,09h,00h,00h,00h,00h
;
        LD      SP,8000
        LD      A,1
        OUT     (0f4h),a        ;obere 256K einstellen
        LD      A,81h           ;Attribut-Speicher mit
        OUT     (0b8h),A        ;Standard-Wert füllen
        LD      HL,3800h        ;Video-RAM ab 3800h
        LD      A,attrib
        LD      (HL),A
        LD      DE,3001h
        LD      BC,07ffh        ;2K Speicher
        PUSH    HL
        PUSH    DE
        PUSH    BC
        LDIR
        POP     BC
        POP     DE
        POP     HL
        LD      A,1             ;Umschalten auf Text
        OUT     (0b8h),A        ;-Speicher
```

```
        LD      A,' '           ;mit Blanks füllen
        LD      (HL),A
        LDIR
        LD      HL,0            ;ROM-Inhalt ins RAM
        LD      DE,0            ;kopieren
        LD      BC,3700h
        LDID
;
;HD64180 initialisieren
        LD      A,10h           ;WAITs: Memory 0, I/O 2
        DB      0edh,39h,32h    ;OUT= (DCNTL),A
        LD      A,83h           ;Refresh-Control
        DB      0edh,39h,36h    ;optimieren
        LD      A,44h           ;Common 0-32K, Bank 32-64K
        DB      0edh,39h,3ah    ;einstellen
        XOR     A               ;Common als Bank benutzen
        DB      0edh,39h,3ah    ;Bank bei 08000-0FFFF
;
;CRTC6845 für 64*16 Zeichen initialisieren
        LD      HL,tab          ;HL => CRTC-Parameter
        CALL    setcrtc         ;raus an CRTC
;
;Video-Signal einschalten
        LD      A,08h           ;Video-Signal aktiv
        OUT     (0b8h),A        ;Video bei 3C00-3fff
                                ;Text-Modus eingesch.
```

```
                                        ;Blinken gesperrt
          JP          0674h             ;und weiter im TRS80-
                                        ;Boot-Gebaren
;
;CRTC-6845 initialisieren
setcrtc   XOR         A                 ;für Video-Mode-Umschal-
          OUT         (0bfh),A          ;tung nötig
          LD          B,16              ;16 Bytes ausgeben
          LD          C,0b5h            ;CRTC-Daten-Port
          XOR         A                 ;CRTC-Register-Nr ab 0
crtclp    OUT         (0b4h),A          ;1. Regsiter
          INC         A                 ;A => nächstes Register
          OUTI                          ;raus das Zeugs
          JR          NZ,crtclp
          LD          A,config          ;HRG voll freigeben
          OUT         (0bfh),A
          RET
          END
```

Assembler-Listing für PAL1

```
PAL20L8
Helmut Bernhardt 22.01.89
Erzeugen der mm -Freigabesignale
ioe rd a10 a11 a9 a8 nc b80 b81 b82 b84 gnd
b86 m016 nc flo adr mfxxxx kb rom ha18 ram m1216 vcc

/ha18 = /b86 */b81 */b80 */m1216 *a11 *a10
      + /b86 */b81 * b80 */m1216 *a11
      + /b86 * b81 */mfxxxx

/kb = /b86 */b80 */rd */m1216 *a11 */a10 */a9 */a8

/flo = /b84 */m1216 */a11 *a10 *a9 *a8 */adr
      + b84 */ioe */adr
```

```
/rom = /b82 */m016 *m1216 */rd
         + /b82 */m1216 */a11 */a10 */rd
         + /b82 */m1216 */a11 * a10 *a9 */rd
         + /b82 */m1216 */a11 * a10 *a9 */a8 */rd

/ram = ioe *ha18 *kb *flo *rom
```

Assembler-Listing für PAL2

```
PAL20L8
Helmut Bernhardt 24.01.89
I/O-Decoder und Datentreiber-Steuerung
rd busak wr a0 a2 a3 a4 a1 a5 a6 a7 gnd
ioe me adr herc inf4 outf4 outb8 flo kb rein lir vcc

/inf4 = /ioe /rd *a7 *a6 *a5 *a4 */a3 *a2 */a1 */a0

/outf4 = /ioe */wr *a7 *a6 *a5 *a4 */a3 *a2 */a1 */a0

/outb8 = /ioe */wr *a7 */a6 *a5 *a4 *a3 */a2 */a1 */a0

/adr = a7 *a6 *a5 */a4

/herc = /ioe *a7 */a6 *a5 *a4

/rein = busak */ioe */lir
       + /ioe */rd *inf4 *herc
       + /busak *rd
       + /flo */rd
       + /kb */rd
```

Assembler-Listing für PAL3

```
PAL16L8
Helmut Bernhardt 23.01.89
Adreßübersetzung für die Hercules-Karte
a10 a9 a8 a7 a6 a5 a0 me b81 gnd
b87 ha0 ha5 ha6ha7 a4 ha8 ha9 ha10 vcc

/ha0 = /me * /b81 * /87
       + /me * b81 * /a0
       * /me * /a0

/ha5 = /me * /b81 * /a4
       + /me * b81 * /a5
       + me * /a5

/ha6 = /me * /b81 * /a5
       + /me * b81 * /a6
       + me * a6

/ha7 = /me * /b81 * /a6
       + /me * b81 * /a7
       + me * /a7

/ha8 = /me * /b81 * /a7
       + /me * b81 * /a8
```

/ha10 = /me * /b81 * /a9
+ /me * b81 * /a10

Pinbelegung des V24-Steckers CN2

Signal	Pin	Pin	Signal
/CTS1	1	2	TXS1
/RTS0	3	4	RXA0
/CTS0	5	6	RXA1
TXA0	7	8	TXA1
GND	9	10	GND

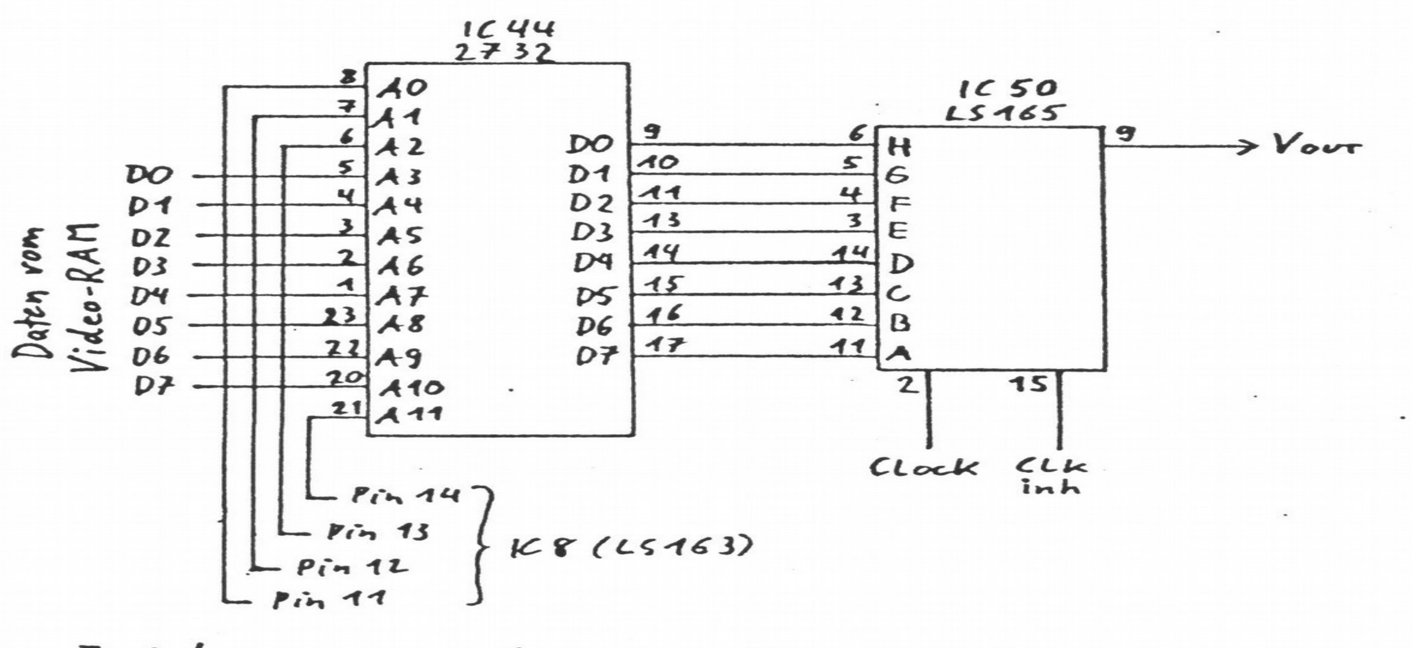
Komtek 1: Verbesserter Zeichengenerator

Wer seinen Komtek 1 nach den im Hardware-Sonderheft Nov.85 beschriebenen Umbauten als relativ komfortable Schreibmaschine nutzt, kann sich eigentlich nur noch an den amerikanischen Sonderzeichen, die zudem auch noch falsch sind, anstelle der deutschen Umlaute auf dem Bildschirm stören. Dieser Mißstand wird nun behoben. Voraussetzung ist, daß man jemanden kennt, der ein 2732 eprommen kann.

Zeichengenerator im Komtek 1

Bei der für den Komtek 1 wegen mangelnder Hardware-Unterlagen üblichen Wühlarbeit zum Durchschauen der Schaltung stellte sich die in Abb.1 wiedergegebene Beschaltung des 2732-Zeichengenerator-EPROMs heraus. Weitere Erkenntnisse über die Organisation (Zuordnung von EPROM-Adressen und -Daten zu Bildschirmpositionen) konnten durch Herausnehmen des EPROMs, Hochbiegen und An-Masse-Legen einzelner Pins und wieder Hineinstecken des EPROMs in den Sockel, und Interpretation der veränderten Bildschirmdarstellung erhalten werden. Letzte Unsicherheiten ließen sich durch Auslesen des Inhaltes des Zeichengerators mit einem EPROMer ausräumen (für diesen Dienst an der Digitaltechnik sei Arnulf Sopp herzlichst gedankt).

Wie sich die verschiedenen Beobachtungen und die Untersuchung des Hexdumps aus dem EPROM zu den folgenden Vorstellungen zusammenfügen, möchte ich mir hier darzustellen ersparen (nicht nur, weil es wohl niemanden interessiert, sondern hauptsächlich, weil ich dazu keine Lust habe). Wichtig ist nur das Ergebnis, und das rechtfertigt den Schreibaufwand.



Zeichengenerator im Komtek 1

Danach ist eine Druckposition für ein Zeichen auf dem Bildschirm in 7 waagerechte und 16 senkrechte Punkte aufgeteilt (bei TRS80 und GENIE 6x12). Der Inhalt einer waagerechten Dotreihe entspricht von links nach rechts den Bits D6 bis D0 im jeweiligen Byte im Zeichengenerator. Die EPROM-Adreßeingänge A0 bis A2 adressieren dabei eine von 8 Dotreihen (8 aufeinanderfolgende Bytes im EPROM) in entweder der oberen oder der unteren Hälfte einer Druckposition während durch den Adreßeingang A11 vorgegeben ist, welche Hälfte angesteuert wird.

An die Adreßeingänge A3 bis A10 sind die Daten D0 bis D7 des Video-RAMs geführt, Ensprechend dem ASCII-Code 00-FF liegen also hintereinander in den unteren 2K des EPROM-Adreßbereichs 256 Gruppen von je 8 Bytes, die dem Bildschirminhalt der oberen Hälfte einer Druckposition entsprechen. Analog sind die Informationen für die untere Hälfte einer Druckposition mit einem Offset von 2048 Adressen in den oberen 2K des EPROMs abgelegt.

Zur Darstellung von Zeichen mit ASCII-Codes bis 7FH wird dabei nur die obere Hälfte einer Druckposition genutzt. Die meisten Zeichen werden innerhalb einer 5x7-Matrix dargestellt. Die untere der 8 waagerechten und 2 seitliche senkrechte Dotreihen werden nicht genutzt. Nur bei Unterlängen wird auch die 8. waagerechte Reihe mitbenutzt. Da diese aber für eine echte Unterlänge nicht ausreicht, werden Zeichen wie "g" und "p" einfach hochgeschoben.

Die unteren Hälften der Druckpositionen werden nur bei den Pixelgraphikzeichen (ASCII-Codes größer 7FH) benutzt. Da die Teilung der 16 wagerechten Dotzeilen durch 3 und die der 7 senkrechten durch 2 keine ganzzahligen Werte ergibt, mußte hingenommen werden, daß die Blockgraphik entsprechend ungleichmäßig aussieht. Die Blockgraphik war aber leztendlich nicht Gegenstand der Kritik sondern der Satz der Zeichen mit ASCII-Codes bis 7FH.

Es ist nicht einzusehen, warum diese Zeichen nicht auch die untere Hälfte der Druckposition mitbenutzen sollen. Es muß nicht sein, daß ein Buchstabe an einem Graphikzeichen in der Zeile darüber festklebt. Dafür müssen nur alle Zeichen um 2 Dotzeilen in der Matrix nach unten geschoben werden, d.h. im EPROM müssen alle Bytes in eine um 2 höhere Adresse verschoben werden.

Innerhalb der 8 Bytes für ein Zeichen wird das erste Byte in die Adresse des 3. Bytes, das 2. Byte in die Adresse des 4. Bytes u.s.w. verschoben. Die freigewordenen ersten beiden Bytes werden mit 00H aufgefüllt. Das 7. und 8. Byte werden aber nicht einfach um zwei Adressen verschoben, sondern um 2048-8 Adressen, damit sie nicht in die ersten beiden Bytes des nächsten Zeichens sondern in die oberen 2 Dotzeilen der unteren Hälfte der Druckposition des gleichen Zeichens gelangen. Diese Prozedur wird allerdings nur mit dem Adreßbereich 0-1K (und 2-3K) durchgeführt, weil die in den anderer 2K des EPROMs untergebrachten Informationen die der Blockgrahik-Zeichen sind.

Damit sind die Zeichen zwar alle von der Zeile darüber abgerückt worden, hübscher sind sie dadurch noch nicht geworden. Deshalb müssen die Zeichen für die Umlaute, die Zeichen mit Unterlängen und alle sonstigen unschönen Gebilde neu gestaltet werden. Das ist aber eine enorme Arbeit, die zu verrichten bei Bestehen eines brauchbaren Zeichensatzes (z.B. der des GENIE) nicht ganz einzusehen ist. Der Zeichensatz im GENIE ist aber etwas anders organisiert. Erstens werden die Grafikzeichen nicht über den Zeichengenerator sondern per Hardware erzeugt, weshalb man dort mit einem 2716-EPROM auskommt. Zweitens wird ein Zeichen durch 16 aufeinanderfolgende Bytes im EPROM dargestellt, wovon nur die ersten 12 Bytes (entsprechend 12

Dotreihen pro Druckposition) benutzt werden. Und drittens ist die Zuordnung der Bits eines Bytes im Zeichengenerator zu den Dots einer Dotreihe genau umgekehrt, d.h. die niederwertigen Bits representieren die linken Dots in der Reihe, was beim Komtek ungekehrt ist.

Da mir bekannt war, daß Arnulf Sopp schon einen verbesserten Zeichensatz für den GENIE erstellt hat, der alle Zeichen um 2 Dots herunterrückt und die Großbuchstaben Ä, Ö und Ü auch tatsächlich als Großbuchstaben erscheinen läßt, und ich nicht nur zu bequem bin sondern auch Respekt vor den Fähigkeiten anderer, die das bedeutend besser können, habe, schmarotzte ich einfach von Arnulfs Arbeit und bat ihn, mir seinen Zeichensatz in den des Komtek einzumassieren. Das ist ihm mit der beiliegenden eleganten Soppftware auch hervorragend gelungen.

Nur einen Haken hat das ganze Projekt. Arnulfs Zeichensatz sieht für die ASCII- Codes 00-1F nicht die für den GENIE üblichen fernöstlichen Gebilde sondern die diesen Codes entsprechenden Hexzahlen vor. Die werden, wenn im GENIE eine ent- sprechende Änderung vorgenommem wird, dann auch im Bildschirm dargestellt, wenn eine Zahl zwischen 00 und 1F in den Bildwiederholseicher gepoked wird.

Beim Komtek ist eine entsprechende Änderung nicht möglich, bzw. ich habe sie noch nicht herausgefunden. Deshalb erscheinen beim Arbeiten im Level-II-Basic, das anstelle der ASCII-Codes 40-5F die Codes 00-1F an den Bildwiederholspeicher ausgibt, dort anstelle der Großbuchstaben die Arnu1f'schen übereinander gestellten Hexzahlen auf dem Bildschirm. Im DOS und Disk-Basic ist dies nicht mehr der Fall.

Um im Level-II-Basic auch erkennbare Zeichen auf dem Bildschirm zu haben, müßten deshalb anstelle der ansonsten sehr informativen Hexzahlen für die ASCII-Codes 00-1F lieber die Zeichen mit den ASCII-Codes 40-5F verwendet werden. Für Bastler, die wegen häuiger Hardware-Eingriffe gelegentlich auf ihre Floppy-Laufwerke verzichten müssen und dann für Testzwecke auf Level-II-Basic angewiesen sind, kann das doch von Bedeutung sein.

Helmut Bernhardt

Und hier Arnulfs Soppftwärechen

```
5200                    ORG            5200h

5200    2100A0   start  LD     HL,0a000h   ;ab hier Komtek-Zeichensatz
5203    110080          LD     DE,8000h    ;ab hier GENIE-Zeichensatz
5206    06B0            LD     B,80h       ;128 Zeichen mit 16 Bytes
5208    C5       loop1  PUSH   BC          ;Zähler retten
5209    0608            LD     B,8         ;Zähler 8 Bytes(1/2 Zeile)
520B    CD2452   loop2  CALL   modify      ;Komtek-Zeichens. verändern
520E    10FB            DJNZ   loop2       ;8 von 16 Bytes
5210    01F807          LD     BC,07f8h    ;untere Hälfte d. Zeichens.
5213    09              ADD    HL,BC       ;Adresse untere Hälfte
5214    04              INC    B           ;B <- Zähler 8 Bytes
5215    CD2452   loop3  CALL   modify      ;untere Hälfte modifizieren
4218    10FB            DJNZ   loop3       ;8 Bytes der unteren Hälfte
521A    B7              OR     A           ;Cy löschen
```

```
521B  010008          LD    BC,0800h   ;jetzt 2K tiefer
521E  ED42            SBC   HL,BC      ;neue Komtek-Adresse
5220  C1              POP   BC         ;Zähler 128*16
5221  10E5            DJNZ  loop1      ;bis 128 Zeichen fertig
5223  C9              RET              ;ins Betriebssystem

5224  C5      modify  PUSH  BC         ;Bytezähler retten
5225  0608            LD    B,8        ;8 Bit pro Byte
5227  1A              LD    A,(DE)     ;GENIE-Byte
5228  07      loop4   RLCA             ;zur Spiegelung
5229  CB19            RR    C          ;Byte rotieren
522B  10FB            DJNZ  loop4      ;bis ein Byte gespiegelt
522D  71              LD    (HL),C     ;neues Komtek-Zeichen
522E  23              INC   HL         ;nächste Komtek-Stelle
522F  13              INC   DE         ;nächste GENIE-Stelle
5230  C1              POP   BC         ;Bytezähler restaurieren
5231  C9              RET              ;ins Hauptprogramm

5200          END     start            ;dort Einsprung
```

Konzept zu Club-80 Terminal auf Basis der Hercules-Karte des PC

Club 80 Projekt:

Bau einer Terminal-Karte auf Basis der PC-Hercules-Karte

Der GENIE 1 mit Hercules-Karte als Video-Interface ist ja ganz witzig aber für die meisten Club-Mitglieder uninteressant, weil die Strippenzieherei einfach zu aufwendig ist und die Festlegung auf die GENIE-Kompatibilität zu beengend ist.

Um das günstige Preis/Leistungs-Verhältnis der Hercules allgemeiner zugänglich zu machen, wurde angeregt, als Projekt im Club 80 eine Terminal-Karte zu bauen, die zusammen mit der Hercules in jedem Computer als Konsole eingesetzt werden kann. Das Terminal soll mit Z80-Power und entsprechender Betriebssoftware fast ein eigenständiger Text-/Grafik-Computer sein, der in Arbeitsteilung mit der CPU Video-Ausgaben erledigt.

Da eine EGA- oder VGA-Karte ähnlich wie die Hercules-Karte funktioniert (erweiterter I/O-Adreßbereich, anderer Memory-Adreßbereich), ist das Prinzip auch auf diese Karten anwendbar. Es bestehen dort aber bei den über die IBM-Vorgaben hinausgehenden Features unterschiedliche Lösungen, die nicht ausreichend dokumentiert sind. Außerdem benötigen diese Karten Monitore, deren Preise weit über den Aktionsradius des Bastlers hinausgehen, so daß wir uns zunächst auf die Hercules-Karte beschränken sollten. Wenn wir das im Griff haben, können wir immer noch überlegen, ob wir nach den Sternen greifen wollen.

Von Andreas Magnus habe ich als letzte Meldung vor längerer Zeit erfahren, daß er auch schon einen entsprechenden Artikel ins Info gesetzt hat; ich weiß allerdings nicht, was er dort schon angekündigt hat. Deshalb möge man mir nachsehen, wenn einige Aussagen hier zum zweiten mal erscheinen. Da inzwischen schon etwas festere Vorstellungen bestehen, ist dieser Beitrag aber unumgänglich.

Die Kommunikation der Terminal-Karte mit der CPU soll über einen bidirektionalen Parallelport (Datenaustausch) und einen Statusport (Handshaking) erfolgen. Zwischen CPU und Terminal werden Befehle der CPU an das Terminal vereinbart, die das Terminal zur selbständigen Bearbeitung der Hercules anregen. Die bisherige Vorstellung sieht dabei im Textmodus die Emulation eines Televideo 950 Terminals und im Grafikmodus die eines Tektronix 4014 Grafikterminals vor.

Während für die Hardware des Terminals schon feste Vorstellungen bestehen, ist für die Betriebssoftware noch alles offen. Zur Erstellung eines Betriebsprogramms für das Terminal sind alle Club-Mitglieder herzlich eingeladen. Der Umfang der zu erstellenden Software ist so groß, daß jedes Gramm Gehirnschmalz im Club gefragt ist.

Durch das Herumwerkeln mehrerer Programmierer ist die Funktion eines Software-Koordinators nötig (Freiwillige bitte vortreten !). Dieser Job bedeutet, einen Überblick über die Gesamtarbeit zu gewinnen, diese in Module zu unterteilen und die Zusammenarbeit der einzelnen Module durch Festlegen von gemeinsam genutzten Datenfeldern und zu überneh-

menden und zu übergebenden Parametern der einzelnen Module (in den CPU-Registern) zu gewährleisten. Diese Vorgaben sind von den Programmierern der einzelnen Module einzuhalten. Wer dazu Vorstellungen hat, möge bitte nicht durch bescheidene Zurückhaltung versuchen, einen guten Eindruck zu machen.

Auch wenn jemand zu Teilaspekten der Betriebssoftware Ideen hat und die Programmierung dieses Parts übernehmen möchte (oder auch nur seine Idee einbringen möchte), kann er einen wertvollen Beitrag zu dem Projekt leisten.

Als Bestechung in Naturalien bekommt jeder, der sich an dem Projekt beteiligt, von mir eine geätzte Terminalkarte geschenkt, die er allerdings selbst und mit eigener Finanzierung der Bauteile aufbauen muß. Hercules-Karten kann ich auch günstig beibringen (zur Zeit für 40,-DM). Wer den Job des Software-Koordinators übernehmen will, bekommt auch die Hercules geschenkt (wenn der Club mit seinem gewaltigen Konto das nicht übernehmen will, stifte ich die Karte).

Um das ganze Projekt etwas konkreter darzustellen, will ich im folgenden die bestehenden Vorstellungen zur Hardware und Software, wie Andreas Magnus und ich uns das vorstellen, breittreten. Wie gesagt, davon ist noch nichts endgültig. Insbesondere bei der Software-Vorstellung kann noch alles ganz anders gestaltet werden. Das Folgende ist also nur als Diskussionsgrundlage zu werten.

So könnte die Hardware aussehen:

Das Terminal ist ein Z80-Computer mit 32K Arbeitsspeicher und 32K Hercules-Video-RAM. Nach dem Einschalten besteht der Arbeitsspeicher beim Lesen aus EPROM und beim Schreiben aus RAM. Durch ein LDIR wird der EPROM-Inhalt bei der Initialisierung ins RAM kopiert. Danach wird das EPROM abgeschaltet und beim Lesen und Schreiben in den unteren 32K erreicht der Z80 das RAM.

Die oberen 32K des Z80 sind dem Video-RAM der Hercules vorbehalten. Hier können die beiden 32K HRG-Pages der Hercules bzw. das Text-/Attribut-Video-RAM eingeblendet werden. Im Textmodus bietet die Hercules-Karte 8 linear hintereinander liegende Seiten mit je 4K Text-/Attribut-Speicher.

Im I/O-Adreßraum des Terminal-Z80 sind die Hercules-internen Ports, das Interface zum Host-Computer, ein Latch zur Einstellung der Konfiguration des Terminals und ein Tastatur-Interface unterzubringen. Zur Vereinfachung der Hardware scheinen folgende Portadressen die sinnvollsten zu sein:

B0H-BFH	interne Ports der Hercules-Karte (siehe Artikel zur Hercules im GENIE)
00H	Status-Register des Interface zum Host-Computer
01H	Daten-Register des Interface zum Host-Computer
02H	Konfigurations-Latch
03H	Tastatur

Konzept zu Club-80 Terminal auf Basis der Hercules-Karte des PC

Das Konfigurations-Latch unterstützt bislang folgende Befehle:

```
  OUT  (02H),0  Einschalten des Text-/Attribut-Speichers im Textmodus
                bzw. der HRG-Seite 0 im Grafik-Modus
  OUT  (02H),1  Einschalten der HRG-Seite 1; im Text-Modus ist der
                Text-Speicher nicht verfügbar
  OUT  (02H),2  im Bereich 0-32K ist bei Leseoperationen das EPROM
                eingeblendet
  OUT  (02H),3  sowohl beim Lesen als auch beim Schreiben liegt im Be-
                reich 0-32K RAM vor
  OUT  (02H),5  und danach
  OUT  (02H),4  Zurücksetzen des Interface der PC-XT-Tastatur
```

Die Ausgabe der Werte 6-15 kann für weitere Konfigurations-Einstellungen benutzt werden. D0 gibt dabei jeweils den Pegel einer entsprechenden Steuerleitung wieder (Q3-Q7 des 74LS259-Latch) und D1-D3 geben vor, welche Steuerleitung geschaltet werden soll.

Die Portadressen, unter denen das Terminal für den Host-Computer erreichbar sein soll, lassen sich über Jumper in Schritten von 4 beliebig einstellen. Die Basisadresse X+0 ist das Status-Register. D0 des Status-Registers zeigt an, ob das Terminal ein weiteres Byte annehmen kann:

```
  D0=0 :  das Terminal kann ein weiteres Byte annehmen
    =1 :  es darf noch kein weiteres Byte an das Terminal ausgegeben
          werden
```

D7 gibt an, ob das Terminal bei der Ausgabe von Daten an den Host ein neues Byte bereitgestellt hat:

```
  D7=0 :  es steht kein neues Byte des Terminals bereit
    =1 :  das Terminal hat ein neues Byte ausgegeben
```

Je nach Zustand der Bits 0 und 7 des Status-Ports kann die CPU an den Daten-Port X+1 ein weiters Byte ausgeben bzw. von dort ein weiteres Byte abholen.

Für den Z80 des Terminals liegen die Adressen der Ports des Interface fest (00H = Status-Register; 01H = Daten-Register). Durch Lesen des Satus-Registers kann der Terminal-Z80 feststellen, ob er Daten entgegennehmen kann/muß bzw. ausgeben kann/muß:

```
  D0=0 :  es kann/muß ein weiteres Byte an den Host ausgegeben werden
          (der Host hat das letzte Byte abgeholt)
    =1 :  es darf noch kein weiteres Byte an den Host ausgegeben wer-
          den
  D7=0 :  es liegt kein neues Byte vom Host vor
    =1 :  der Host hat ein neues Byte ausgegeben, das abzuholen ist
```

Zur Vereinfachung der Kommunikation zwischen Host und Terminal wird am Terminal-Z80 ein Interrupt erzeugt, wenn der Host ein neues Datum ausgegeben hat.

Als Tastatur kann wahlweise eine 8Bit-Parallel-ASCII-Tastatur oder eine serielle PC-XT-Tastatur verwendet werden. Beide Interfaces sind bei der

bislang bestehenden Prototypen-Karte vorhanden und können durch Jumper selektiert werden. Wenn von der Tastatur ein Byte gekommen ist, wird am Terminal-Z80 ein /NMI ausgelöst.

Und das sollte die Software unterstützen:

In allen vorgesehenen Betriebsmodi des Terminals sind Umschaltbefehle möglich, die in einen jeweils anderen Modus umschalten oder sonstige Spezialaufgaben erledigen:

1) Einschalten des TVI 950 Textmodus (Grundeinstellung)
2) Einschalten des Tektronix 4014 Grafikmodus
3) Umschalten zwischen Local Mode und OnLine-Betrieb
4) Übernehmen eines Maschinenprogramms vom Host;
   Der Befehl enthält als Parameter Ladeadresse, Länge und Call-Adr.
5) Umschalten zwischen Video- und Drucker-Ausgabe
6) Übernahme eines Initialisierungsstrings und der Bit-Grafik-Einleitung für den jeweils an der Hercules angeschlossenen Drucker
7) Hardcopy des Bildschirmes (sowohl Text als auch Grafik, was gerade eingestellt ist)

Die ersten 3 Punkte machen schon ein recht ansehnliches Terminal aus. Die Übernahme und optionale Ausführung von Maschinenprogrammen hat hauptsächlich in der Entwicklungsphase der Software ihre Bedeutung. Außerdem erlaubt sie auch später noch den Mißbrauch der Z80-Power des Terminals für Anwenderzwecke.

Der Punkt 5 erschlägt mit dem Terminal auch gleich eine Drucker-Schnittstelle. Die Punkte 6 und 7 drängen sich bei der gemeinsamen Lage von Video- und Drucker-Interface geradezu auf.

Als Grundstock an Treiberroutinen müssen die Initialisierung des Textbildschirms und des Grafikbildschirms sowie ein Tastatur-Treiber vorhanden sein. Für die beiden Video-Modi muß je ein Interpreter für die im jeweiligen Modus gültigen Control-Codes und Escape-Sequenzen existieren, der auch immer die oben angeführten globalen Umschaltbefehle mit berücksichtigt.

RAM bei 0000H-37DFH anstelle des Sonder-ROMs im GENIE

Helmut Bernhardt

Der Sonder-ROM des GENIE hat durchaus seine angenehmen Seiten; es gibt auch kaum GENIE-Besitzer, die ihn endgültig aussondern (eine ebenfalls mögliche Interpretation seines Namens). Aber, wenn man ehrlich zu sich selbst ist, muß man rückwirkend feststellen, daß man ihn so gut wie nie benutzt hat. Manchmal fragt man sich, ob es nicht besser wäre, diesen kostbaren Adreßraum als RAM zur Verfügung zu haben. Aber warum nicht das Eine tun ohne das Andere zu lassen.

Eine kleine Umbaumaßnahme führt dazu, daß im Grundzustand der Sonder-ROM verfügbar ist und auf Tastendruck (ohne Treiberroutine) auf RAM umgeschaltet wird. Es wurde dabei die Kombination der Tasten Y und Z ausgewählt, die hinreichend weit auseinander liegen, um ein ungewolltes Umschalten auszuschließen und wegen ihrer Lage in der Tastaturmatrix den Hardwareaufwand gering halten.

Dazu muß leider zunächst ein Teil der Arbeit, die dafür aufgebracht wurde, den Sonder-ROM voll zu decodieren, um im Bereich 3000H-37DFH RAM zur Verfügung zu stellen, wieder zunichte gemacht werden. D.h., das dabei eingesetzte IC 74LS08 wird wieder entfernt und durch ein 74LS11 ersetzt, und mit den gleichen Signalen wie im Schaltplan zu ersehen ist, versorgt. Auch das Signal SYSRES* (aus RESET* und NMI* durch AND-Verknüpfung gebildet) wird hier wiederhergestellt und auch gleich verwertet.

Zusätzlich zu den Signalen, die das 74LS08 schon erhielt, wird jetzt auch noch das das Signal 3000-37DF* an Pin 9 des 74LS11 gelegt. Dieses Signal wird an Pin 11 eines zusätzlichen 74LS32 erhalten und über ein weiteres OR-Gatter dieses ICs mit dem Ausgang eines Flip Flops verknüpft. An Pin 8 dieses Gatters kann nur dann Low-Pegel erscheinen, wenn das Signal 3000-37DF* low ist und wenn der Ausgang des Flip Flops (Pin 6 eines zusätzlichen 74LS00) low ist.

Dieser Ausgang ist aber allgemein high, sodaß das Signal 3000-37DF* nicht durchgeschaltet wird. Der Pegel an Pin 6 des Flip Flops wird bei jedem RESET (Einschalten des Computers) und jedem NMI (Drücken des RESET-Knopfes) auf high eingestellt. In diesem Zustand kann also bei Adressen zwischen 3000H und 37DFH nie Low-Pegel an Pin 9 des 74LS11 ankommen.

Erst wenn von Pin 6 des zusätzlich einzubauenden 74LS32 ein Low-Impuls ausgegeben wird und das Flip Flop umkippt, gelangt jedesmal, wenn eine Adesse zwischen 3000H und 37DFH auf dem Bus liegt, das low aktive Signal 3000-37DF* an den Pin 9 des 74LS11.

Der Ausgang dieses AND-Gatters (Pin 8) ist immer dann low, wenn einer seiner Eingänge low ist, also wenn Adressen in den Bereich 16-64K, 3900H-3BFFH und nun auch 3000H-37DFH auf dem Bus liegen. Der Pin 8 steuert die Freigabe des RAM-Speichers. Immer, wenn dort LOW-Pegel ausgegeben wird, erfolgt ein Zugriff auf das RAM.

Genauso, wie der Pin 6 des Flip Flops über das OR-Gatter (Pins 10, 9, 8) regelt, ob das Signal 3000-37DF* eine RAM-Freigabe bewirkt, steuert der Pin 8 des Flip Flops, ob durch dieses Signal der Sonder-ROM freigeschaltet wird. Da die Pins 6 und 8 immer entgegengesetzte logische Pegel haben, können keine Konflikte zwischen RAM und Sonder-ROM entstehen.

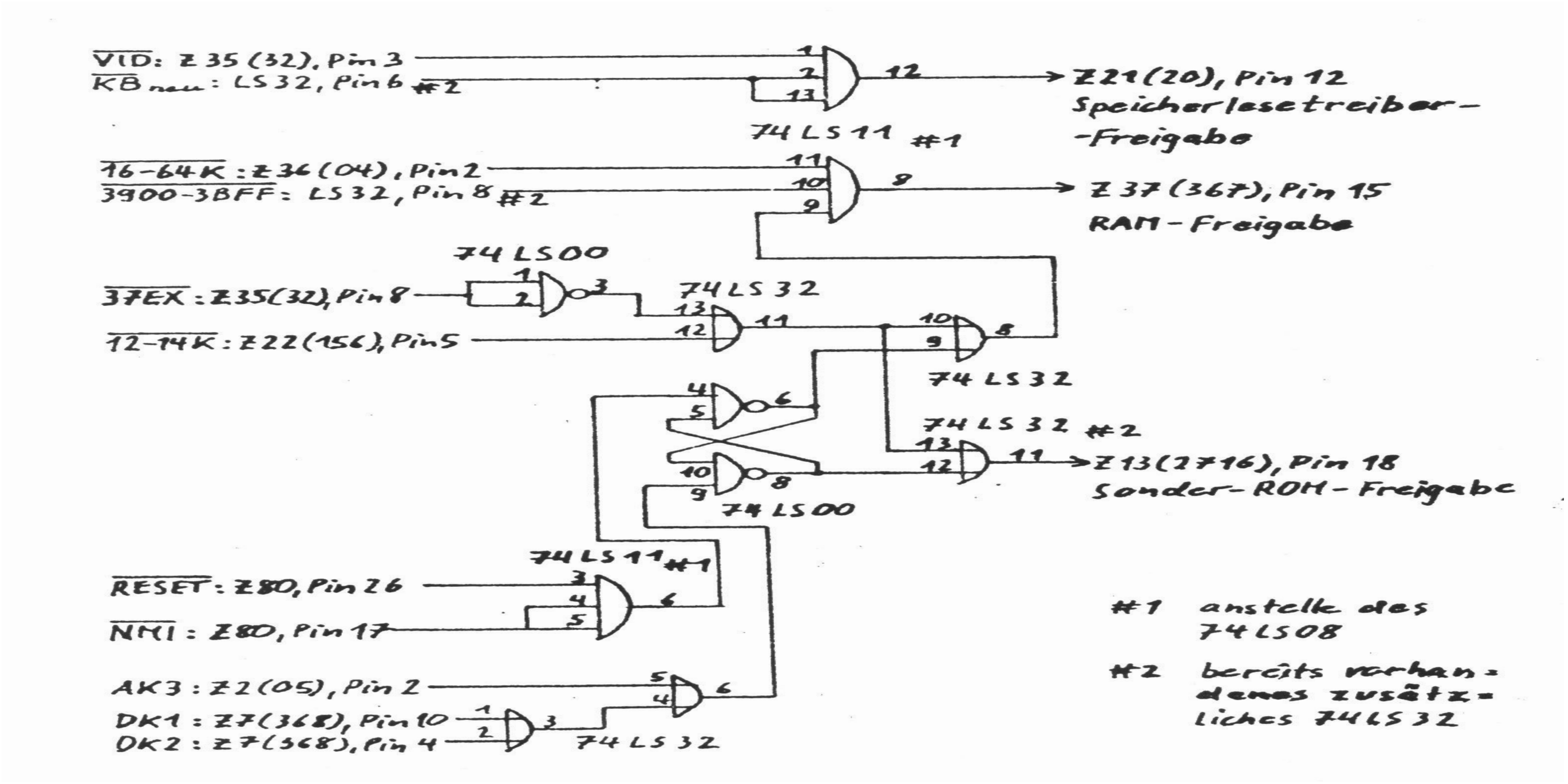
Das Umkippen des Flip Flops (kein Sonder-ROM mehr verfügbar, dafür RAM in dessen Adreßbereich) wird durch OR-Verknüfung von Tastatur-Signalen erreicht. Dafür muß die Tastaturadreßleitung AK3 low aktiv sein und die Tastatur-Datenleitungen DK1 und DK2 müssen ebenfalls low sein. AK3 wird immer dann low, wenn die Tastaturadresse 3808H gelesen wird. Wenn in diesem Moment die Tasten Y und Z gedrückt sind, werden dadurch DK1 und DK2 auf low gezogen das dann an Pin 6 der beiden hintereinander geschalteten OR-Gatter ausgegebene Low-Signal schaltet das Flip Flop um.

Das Umschalten ist nur dann möglich, wenn eine Tastatur-Treiberroutine aktiv ist (NEWDOS/80 READY; Befehlseingabe; Komm, laß knacken u.s.w.), oder wenn ein Programm die Adressen 3808H (oder jede andere Adresse im Bereich 3800H-38FFH mit A3 = 1) liest.

Umbauanleitung

Die Umrüstung baut darauf auf, daß der Computer mit 4164-RAMs bestückt ist und daß die Erweiterung zur vollen Decodierung des Sonder-ROMs und zur Bereitstellung von RAM im Bereich 3900H-3BFFH schon durchgeführt wurden.

Das bei diesem Umbau mit eingesetzte IC 74LS08 wird entfernt und durch ein 74LS11 ersetzt. Die Leitungen, die vorher an das 74LS08 führten, werden nun gemäß Schaltplan an das 74LS11 gelegt.

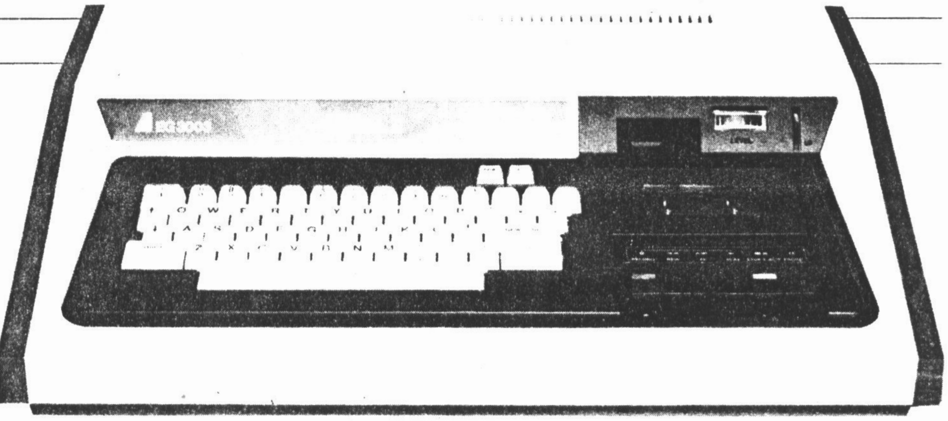


Außerdem sind noch ein weiteres 74LS32 und ein 74LS00 nötig. Da in der näheren Umgebung des Schlachtfeldes keine geeigneten DIL14-ICs mehr vorhanden sind, auf die diese ICs huckepack-gelötet werden können, müssen DIL16-ICs als Fundament herhalten (z.B. Z8 und Z9; 74LS368 und 74LS367). Dabei muß das DIL14-Huckepack-IC mit seinen Pins 7 und 14 etwas Spagat üben, um mit den Füßen an die Pins 8 und 16 des DIL16-ICs darunter zu gelangen. Es gilt auch hier wieder "Die Pins 1-6 und 8-13 werden hochgebogen". Das Freigabesignal für den Sonder-ROM wird über das noch freie OR-Gatter des 74LS32 (Pins 13, 12, 11), das bei der letzten Umbaumaßnahme schon eingesetzt wurde, erhalten. Die Leiterbahn, die zu Pin 18 des Sonder-ROM führt, muß durchtrennt werden.

Wenn die an die Pins 4 und 9 des zusätzlichen 74LS04 führenden Leitungen vertauscht werden, liegt von vornherein RAM vor und die Tasten Y und Z schalten auf Sonder-ROM um.

Bei eingebautem "Ver-16-facher des Sonder-ROM-Bereichs" ist dieses Projekt weder nötig noch möglich.

Ein sicheres Plätzchen



Maschinenroutinen geschützt im GENIE untergebracht

Helmut Bernhardt

Die Sicherung von Maschinenprogrammen im oberen Adreßbereich durch Befehle wie HIMEM oder MEMSIZE gewährt lediglich Schutz gegen Überschreiben durch BASIC-Programme. Der Computer legt Maschinenprogramme beim Laden immer in den Speicherbereich, für den sie vorgesehen sind. Die Pointer werden dabei nicht berücksichtigt, so daß es normalerweise keinen Speicherbereich gibt, der wirklich 'sicher' ist. Mit einer kleinen Zusatzschaltung kann man jedoch ein 'sicheres Plätzchen' schaffen.

Auf der Suche nach diesem Platz im Adreßraum des TRS-80 oder GENIE I, II findet man schnell einige 'Lücken'. So beginnt der Adreßbereich der Tastatur bei 3800h und endet bei 38FFh. Da aber die Dekodierung des Freigabesignals für die Tastatur-Lesetreiber unvollständig ausgeführt ist, wird auch der Bereich von 3900h bis 3BFFh blockiert.

Das muß man nicht so hinnehmen. Durch geringfügige Eingriffe in den Rechner läßt sich die Tastatur vollständig dekodieren und gleichzeitig ein Freigabesignal für einen zusätzlichen Speicher-Baustein im Bereich 3900h bis 3BFFh gewinnen. Die dazu notwendige Hilfsschaltung zeigt Bild 1. Der zusätzliche Speicher besteht aus zwei ICs 2114 (1 K x 4 Bit); die weitere Dekodierung geschieht mit zwei TTL-ICs.

In den so gewonnenen 768 Byte RAM sind eigene Routinen ziemlich sicher, da man normalerweise nicht auf diesen Speicherbereich zugreifen kann. Programme mit eigener Treiberroutine könnten allerdings versuchen, die Tastatur im Bereich 3900h bis 3BFFh auszulesen. Solche Routinen würden laufend 'Eingaben' aus dem RAM erhalten, was zu Fehlfunktionen führt.

In solchen (seltenen) Fällen hilft dann nur noch der Einbau eines Zweifach-Umschalters, der entweder das unvollständig dekodierte oder das neu gewonnene Freigabesignal an die Tastatur-Lesetreiber legt und die Freigabe der RAMs bei Benutzung des 'alten' Freigabesignals für die Tastatur abschaltet. Programme, die diesen Schalter benötigen, lassen sich dann nicht mit dem zusätzlichen RAM verwenden. Das Betriebssystem NEWDOS-80 gehört glücklicherweise nicht zu dieser Gruppe von Software.

Eingriffe

Die vier ICs der Hilfsschaltung kann man auf einem Streifen Lochraster-Platine unterbringen. Die Versorgungsspannung (+5 V) ist jeweils an Pin 14 der TTL-ICs und an Pin 18 der RAM-Bausteine anzuschließen. Der Masseanschluß ist bei den RAMs Pin 9, bei den TTL-Bausteinen Pin 7. Zwischen Versorgungsspannung und Masse sollte man einen Kondensator (100 nF) schalten.

Beim Einbau der Karte in GENIE-Rechner kann man folgendermaßen vorgehen:

Zuerst ist das Gehäuse des Computers zu öffnen. Nachdem man die acht tiefer versenkten Schrauben in der Bodenwanne des Gerätes herausgedreht hat, läßt sich die obere Gehäuseschale abheben. Schraubt man noch die Tastatur ab, liegt links das CPU-Board, in der Mitte das Interface-Board und rechts das Netzteil sowie der Kassettenrecorder.

Die durchzutrennende Leitung (siehe Bild 1) ist die Leiterbahn, die vom hintersten IC (Z35, 74LS32) auf dem CPU-Board in der rechten Spalte nach 'hinten' verläuft (neben der 5-V- und Masse-Leitung). An die beiden Seiten der Trennstelle kann man den Ein- und Ausgang des 'OR'-Gatters der Hilfsschaltung anschließen. Das IC 'Z12' findet man an zweiter Stelle (von vorne) in der linken Spalte auf dem CPU-Board. Der Baustein 'Z15' (74LS32) liegt in der zweiten Spalte (von links gezählt) als zweites IC.

Die Numerierung der Verbindungsleitungen zum Interface-Board, an denen die meisten Signale für die Zusatz-Schaltung abzunehmen sind, erfolgt von vorne nach hinten (Nummern 1 bis 32).

Beim Anschließen der Erweiterung sollte man unbedingt darauf achten, daß die Datenleitungen nicht an den ROMs, sondern an den Lötpunkten der Stecker für die Verbindungsleitung zum Interface-Board abgenommen werden. Die an den ROM-Chips anliegenden Datenleitungen sind wegen der Freigabeschaltung der Treiber nur in Leserichtung verwendbar. An der Steckverbindung lassen sich außer $\overline{CS}$ und $\overline{MWR}$ alle für die RAMs nötigen Signale abnehmen. □

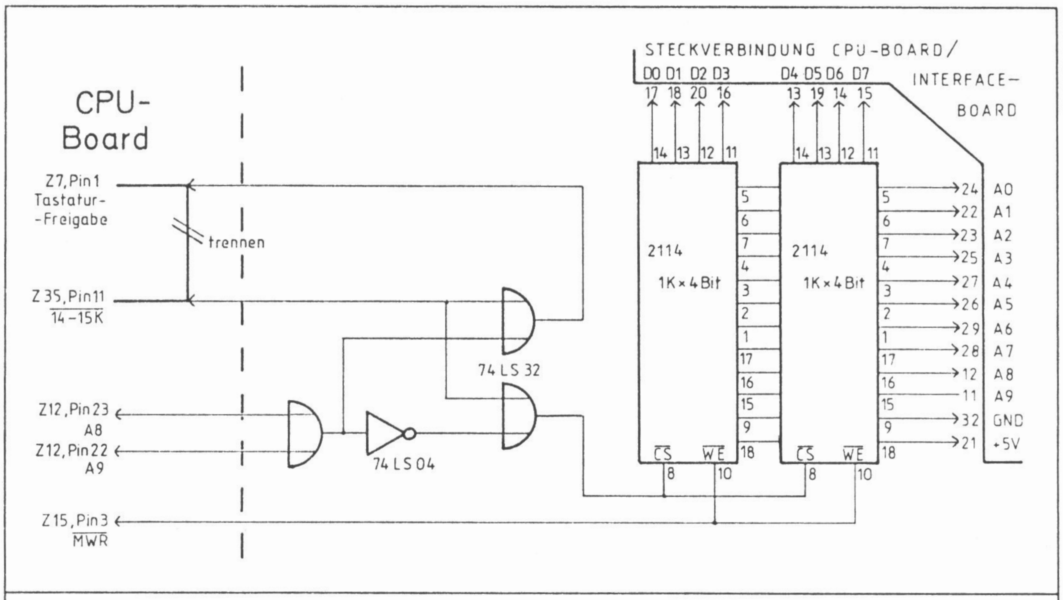


Bild 1. Die Zusatzschaltung für ein sicheres Plätzchen

Noch'ne Geschwindigkeitserhöhung

Helmut Bernhardt

In den letzten Jahren wurden verschiedene Möglichkeiten für Hardwarepatches zur Geschwindigkeitserhöhung bei TRS80 und GENIE vorgestellt. Allen Vorschlägen gemeinsam war, daß dies bis maximal 3,55 MHz (100% Speedup) möglich war. Voraussetzung war natürlich auch, daß 64K RAM auf dem CPU-Board vorhanden waren und daß ein Z80-A eingesetzt wird. Unterschiedlich war allerdings der Aufwand, der dafür getrieben wurde. Ein ehemaliges Mitglied war sogar der Meinung, daß der Erwerb eines Platinenschnippels mit 5 Allerwelts-ICs für den Gegenwert von 30 großen Bieren ein big Deal war, andere begnügten sich mit einfachem Umverdrahten am Frequenzteiler.

Wie auch immer vorgegangen wurde, bei 3,55 MHz war Schluß. Wenn man diese Grenze überwinden will, muß man vom vorgegebenen Prinzip der Erzeugung der RAM-Steuersignale RAS*, MUX* und CAS* aus dem System-Muttertakt, der in seinem Wert wegen der Verwendung beim Video-Timing festgelegt ist, abrücken. Dieses Prinzip erzeugt aus dem Muttertakt und den Z80-Signalen RD* und WR* diese Signale über zwei Flip Flops. Wenn man aber am Frequenzteiler für den Z80 einen nicht so weit heruntergeteilten Takt abgreift und nicht gleichzeitig das Timing der RAM-Steuersignale ändert, läuft irgendwann beides nicht mehr synchron. Das Prinzip, wie beim TRS80 und GENIE diese Schaltungen aussehen und wie dort zur Geschwindigkeitserhöhung eingegriffen werden kann, ist in Abb.1 zu sehen.

Eine andere Art der Erzeugung der Signale RAS*, MUX* und CAS* hat uns TCS im GENIE IIs vorgeführt. Dort ist eine soft- (und/oder) hardgesteuerte Umschaltung zwischen 1,77 und 5,3 (6,0 oder 8,0) MHz möglich. Dafür konnte bei so stark voneinander abweichenden Geschwindigkeiten natürlich nicht mehr ein konstantes Timing der RAM-Steuersignale verwendet werden. Die RAM-Steuersignale werden dort aus dem jeweils eingestellten CPU-Takt und dem Z80-Signal MERQ* hergeleitet. Wegen der dabei möglichen hohen Geschwindigkeiten wird dafür ein 74S74 Flip Flop eingesetzt.

Diese Variante läßt sich natürlich auch im TRS80 oder GENIE I oder II verwenden (Abb.2). Da hierbei sowohl CPU-Takt als auch Steuerung der RAMs aus dem gleichen Takt abgeleitet werden, kann dieser beliebig gewählt werden (solange die Hardware noch mitspielt). Ein solcher Takt kann entweder am Teiler 74LS92 auf dem CPU-Board abgegriffen werden oder über eine einfache Oszillator-Schaltung zusätzlich erzeugt werden.

Außer den bekannten Problemen mit RAMs im Expansions-Interface und einer flinkeren Variante der Z80-CPU spielt irgendwann auch die Zugriffszeit der ROMs eine Rolle. In beiden Computern stecken Exemplare, deren Zugriffszeiten so ungünstig wie nur möglich sind. Es gibt aber heute schon 32K-EPROMs (27256) mit 150ns Zugriffszeit und 16K-EPROMs (27128) mit 170ns Zugriffszeit. Wenn man sich den ROM-Inhalt in ein solches IC umfüllt, sollten bedeutend höhere Taktraten möglich sein. Bei Verwendung eines 27256-EPROMs ist durch Umschalten des Pegels am Adreßeingang A14 auch ein Wechsel zwischen Original Level 2 und einer aufgebohrten eigenen Schöpfung möglich.

Da diese EPROMs mit ihren 28 Pins aber nicht mehr in einen der bestehenden Sockel passen und beim TRS80 auch nicht pinkompatibel zu den dort verwendeten ROMs sind, müssen ein paar Leitungen durch die Luft dorthin verdrahtet werden. Außerdem müssen die Freigabe-Signale der bisherigen ROMs über AND-Verknüpfung zu einem gemeinsamen Freigabe-Signal für das neue EPROM zusammengefaßt werden.

Beim Umändern auf die in Abb.2 dargestellte Variante braucht nur das 74S74 Flip Flop zusätzlich eingessetzt zu werden. Die Treiber aller Signale sind auch im ursprünglichen Zustand schon vorhanden und können weiterhin benutzt werden. Die folgende Tabelle gibt für TRS80 und GENIE I und -II die Punkte an, wo das zusätzliche Flip Flop anzuschließen ist.

Punkt	74S74	Signal	TRS80			GENIE I/II		
	Pin		IC	Typ	Pin	IC	Typ	Pin
A	-	Clock	37	367	2 #1	72	367	12 #1
B	3	CPU-Clock	37	367	3	72	367	11
C	2	RAS*	16	367	5	72	367	5
D	6	MUX*	37	367	4 #2	72	367	2 #2
E	5	PCAS*	37	367	14 #2	72	367	10 #2
F	4	RFSH*	16	367	3	40	Z80	28

#1 hier kann entweder ein externer Oszillator oder eine aus dem Muttertakt über den 74LS92 heruntergeteilte Frequenz angelegt werden

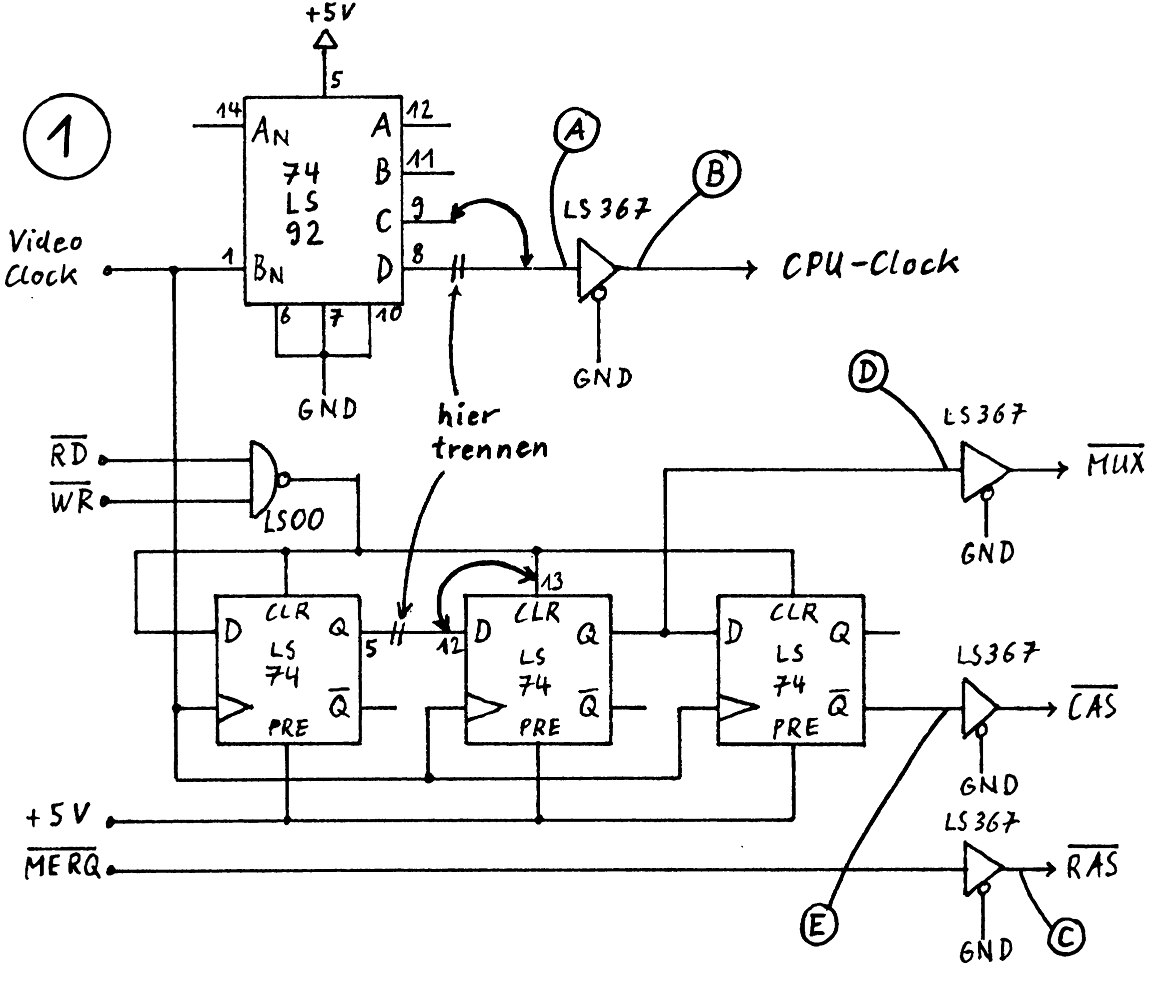
#2 die bisher zu diesem Pin führende Leitung wird durchtrennt

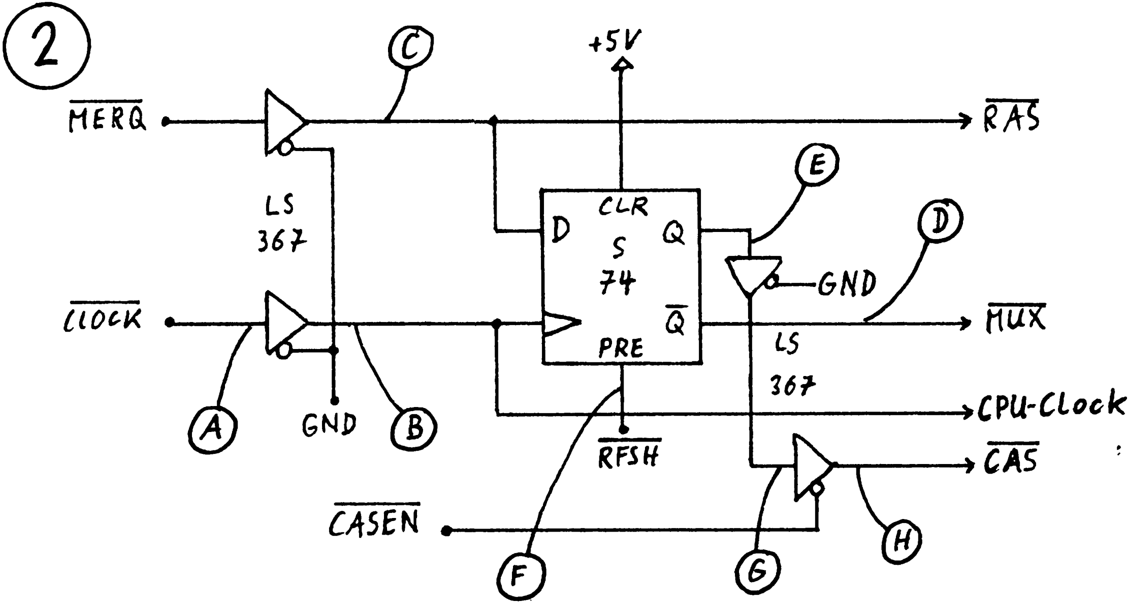
Zum Einsatz einer HD64180-CPU ist es nun nur noch ein kleiner Schritt. Im Unterschied zum Z80 wird der HD64180 an den Pins 2 und 3 direkt mit einem Quartz beschaltet, der die doppelte Frequenz des beabsichtigten Systemtaktes liefert. Der HD64180 halbiert diese Frequenz intern und gibt dann diesen Systemtakt über Pin 64 aus. Der Systemtakt kann dann über Pin 6 der Z80-Fassung dem Computer zugeführt werden. Dabei muß natürlich die Verbindung zwischen Treiber und Pin 6 der Z80-Fassung durchtrennt werden, damit nicht alter und neuer Takt gegeneinander arbeiten. Der neue Takt braucht nur noch an das 74S74 Flip Flop und - soweit vorgesehen - auf den externen Systembus geführt zu werden.

Im Zusammenhang mit dem HD64180 läßt sich natürlich auch noch der Speicher auf 256K oder 512K RAM erweitern. Die dabei anfallenden Änderungen gehören aber nicht mehr zum Thema und werden nur anhand von Abb.3 kurz skizziert.

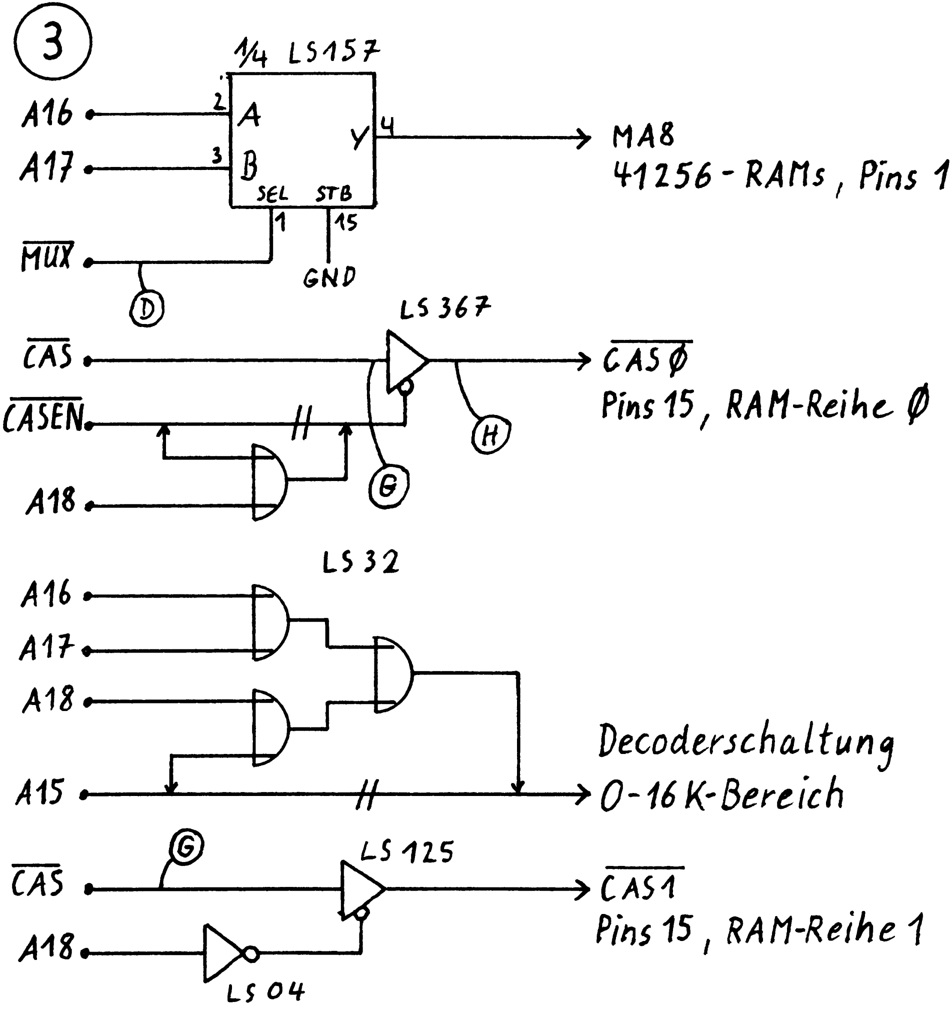
Obige Überlegungen sind für mich bislang nur Theorie. Für mich besteht im Moment nicht die Dringlichkeit, das in die Tat umzusetzen und andere Dinge dadurch zu verschieben. Ich kann also keinen Erfolg bei diesen Umbauten garantieren und auch nicht sagen, welcher Computer sich wie weit scheuchen läßt.

Vielleicht ist ja irgendjemand von euch etwas mutiger und sieht dem Projekt auch einen höheren Stellenwert ab. Es würde mich freuen, wenn dieser Denkanstoß irgendwo eine Hilfe sein kann. Die beschrieben Maßnahmen sind aber auch Vorausetzung, um in diesen Computern einen HD64180 fahren zu können und deshalb auch für mich wieder interessant (worin die Ursache dieses Beitrags begraben liegt).


1
+5V
74
LS
92
Video
Clock
CPU-Clock
LS 367
GND
hier
trennen
RD
WR
LS00
CLR
LS
74
PRE
MUX
CAS
RAS
+5V
MERQ



2
MERQ
CLOCK
C
+5V
RAS
LS
367
CLR
D
S
74
Q
Q
PRE
E
GND
D
MUX
LS
367
CPU-Clock
GND
A
B
RFSH
CAS
CASEN
F
G
H


3
1/4 LS157
A16
A17
2
3
A
B
Y
4
SEL
STB
1
15
MUX
GND
D
MA8
41256-RAMs, Pins 1
LS367
CAS
CASEN
A18
G
H
CAS Ø
Pins 15, RAM-Reihe Ø
LS32
A16
A17
A18
A15
Decoderschaltung
0-16K-Bereich
CAS
A18
G
LS125
LS04
CAS1
Pins 15, RAM-Reihe 1

Ver-16-fachung des Sonder-ROM-Bereichs und Nutzung durch das Betriebssystem

Helmut Bernhardt (Hardware); Arnulf Sopp (Software)

Die übliche Technik, das Betriebssysten zu tunen, führt in der Regel dazu, daß durch Herunterrutschen des HIMEM der Anwender-RAM-Speicher immer kümmerlicher wird, wobei anspruchsvollere Programme zwar eine Fülle zusätzlicher Features des Betriebssystems zur Verfügung haben (ob sie wollen oder nicht) und dafür eventuell zu wenig Speicherplatz für sich und ihre Daten/ Variablen haben.

Zwei Möglichkeiten, wie zusätzlicher Speicherplatz geschaffen werden kann, der nicht auf Kosten des Anwenders geht, sind in 1) und 2) beschrieben. Während die in 1) beschriebene Methode (vollständige Decodierung der Tastatur 3800H- 38FFH und Bereitstellen von RAM im Bereich 3900H- 3BFFH) mit allen Hard- und Software-Produkten zusammen funktioniert, ist die in 2) beschriebene Erweiterung an das Vorhandensein eines herkömmlichen Expansion-Interface EG 3014 mit Sockeln für 2 x 8 dynamische RAMs gebunden und die Vervielfältigung des Sonder ROM Bereichs ist in das ganze Projekt nur als Abfallprodukt mit eingebunden.

Hier soll nun beschrieben werden, wie man die Vervielfältigung des Sonder-ROM-Bereichs und gleichzeitig dessen vollständige Decodierung, so daß auch der PUNCH Befehl des GENIE-Monitors (Maschinen Programme auf Cassette schreiben) aus 3) funktioniert, durchführen kann, ohne das aufwendige Banking-Board zur Verwaltung zusätzlicher dynamischer RAMs aufbauen zu müssen.

Zunächst grob qualitativ das Prinzip der Schaltung:
In ein portdecodiertes Latch (74116) wird ein 4-Bit-Wort als Nummer der einzuschaltenden Speicherbank eingetragen. Dieses 4-Bit-Wort liegt anschließend am Latch-Ausgang ständig an, bis ein anderes Wort eingetragen wird. Die Latchausgänge werden an einen 1-aus-16-Decoder (74154) gelegt, der dann je nach Bitmuster an seinen 4 Eingangs-Pins den entsprechenden Ausgangspin low aktiv werden läßt, wenn seine beiden E*-Eingänge low aktiv sind. Wenn man an diese E*-Eingänge das volldecodierte Freigabesignal 3000H-37DFH* für das Sonder-ROM legt, können die 16 Ausgänge des 74154 als Freigabesignal für insgesamt 16 Stück 2716-EPROMs benutzt werden.

Will man anstelle von EPROMs auch RAMs benutzen, kann man nicht einfach die Signale vom Sockel des Sonder-ROMs an die 16 Speicherbausteine (2716-EPROMs oder 6116-RAMs) führen.

1) werden die am Sonder-ROM-Sockel anliegenden Datenleitungen über unidirektionle Treiber nur in Leserichtung mit dem Systemdatenbus verbunden, weshalb ein Schreiben in RAMs so nicht möglich ist und

2) liegt an Pin 21 nicht das für RAMs nötige Signal WR* bzw. MWR* an. Dieser Pin ist einfach an +5V angeschlossen.

3) ist an den CE*-Pin 18 Masse angelegt und ein nicht vollständig decodiertes Freigabesignal 12-14K* (=3000H-37FFH*) ist an den Pin 20 OE* geführt, der eigentlich ein RD*- oder MRD*-Signal verdient hätte. Kollisionen mit den I/O-Adressen 37E0H-37FFH werden dadurch verhindert, daß durch das Signal 3600H-37FFH* an Pin 13 von Z21 (74LS20) die Freigabe des Speicherlesetreibers gesperrt ist. In diesem großzügigen Abwürgen des Sonder-ROMs liegt auch die Ursache des nicht funktionierenden PUNCH-Befehls.

Um diesem übel abzuhelfen, sind folgende Eingriffe nötig:

1) Die Datenleitungen zum Sockel des Sonder-ROMs müssen durchtrennt werden.

2) Die Datenpins des Sonder-ROMs werden direkt mit den Ausgängen der Datenbustransceiver der CPU verbunden.

3)Die Datenleitung D7 die über den Sonder-ROM-Sockel an den RAM-Baustein Z34 führt, muß zwischen RAM und Sonder-ROM durchtrennt werden. D7 für Z34 muß stattdessen vom ROM Z12 an das RAM geführt werden.

4) Die Leitungen zu den Pins 18 und 19 des Sonder-ROM-Sockels (GND und +5V) werden durchtrennt.

5) An Pin 18 des Sockels wird MRD* und an Pin 21 wird MWR* gelegt.

6) Das Signal 3600H-37FFH* wird von der Steuerung des Speicherlesetreibers abgetrennt. Stattdessen wird dort 12-14K* angelegt, so daß der Speicherlesetreiber für den gesamten Bereich 3000H- 37FFH gesperrt wird.

7) Durch ein Flachkabel und 2 24-polige Quetschstecker und ein in dieses Kabel eingefügtes Adapter-Board wird die 16-fach-Sonder-ROM/RAM-Karte mit dem wie oben beschriebenen geänderten Sonder-ROM-Sockel verbunden. Das Adapter-Board bringt nicht nur die Signale CE* und MRD* auf die richtigen Pins sondern leistet auch die volle Decodierung auf den Bereich 3000H- 37DFH und stellt einen bidirektionalen Datenbustransceiver bereit, so daß auch RAMs benutzt werden können.

Die Wahl eines 74116 Latches bietet den Vorteil, daß durch AND-Verknüpfung der Signale RESET* und NMI* (ergibt das beim TRS 80 bereits vorhandene Signal SYSRES*) und Anlegen an den CLEAR-Eingang des Latch nach dem Einschalten oder Drücken des "RESET"-Knopfes immer die Bank 0 eingestellt ist. Wenn der bisherige Sonder-ROM auf der Ver-16-fachungs-Karte durch den Ausgang 0 des 1-aus-16-Decoders 74154 freigegeben wird, verhält sich der Computer wie vorher auch. Wenn keine andere Bank eingeschaltet wird, liegt im Adreßbereich 3000H-37DFH immer der Sonder-ROM vor.

Arbeitsanleitung

1) Änderungen am Sonder-ROM-Sockel auf dem CPU-Board

Auf der Lötseite werden die Leitungen zu den Pins 18 und 21 des Sonder-ROMs durchtrennt. An Pin 18 wird das Signal MRD* von Z15 (74LS32), Pin 11 und an Pin 21 das Signal MWR* von Z15 (74LS32), Pin 3 gelegt.

Die Verbindung zwischen Pin 14 des RAMs Z34 mit Pin 17 des Sonder-ROMs wird durchtrennt. Der Pin 14 des RAMs wird stattdessen mit dem Pin 17 des ROMs 3003 verbunden.

Auf der Bestückungsseite werden (von der Mitte des CPU-Boards aus gezählt) die ersten 7 Leitungen zwischen ROM 3 (3003) und dem Sonder-ROM-Sockel durchtrennt. Die 8. Leitung bleibt bestehen und die 9. Leitung wird ebenfalls durchtrennt. Dann werden mit isolierten Drähten auf der Lötseite folgende Verbindungen hergestellt:

Signal	Pin des Sockels	Verbindung zu Pin von	IC	Typ
D0	9	7	Z18	74LS367
D1	10	9	"	"
D2	11	5	"	"
D3	13	13	Z19	"
D4	14	11	"	"
D5	15	3	Z18	"
D6	16	13	"	"
D7	17	11	"	"

2) Änderung an der Speicherlesefreigabe auf dem CPU-Board

Die Leitung zu Pin 13 von Z21 (74LS20) wird durchtrennt. Der Pin 13 von Z21 wird mit Pin 5 von Z22 (74LS156) verbunden. (Abb.1)

3) Anschluß und Bestückung der Ver-16-fachungs-Karte

Auf dieser Karte werden ausschließlich 24-Pin-IC-Sockel eingesetzt. An der im Layout beschrifteten Position wird der 1-aus-16-Decoder 74154 eingesetzt. An eine der beiden Positionen daneben wird der Stecker des Flachkabels vom Sonder-ROM-Sockel auf dem CPU-Board eingesetzt. Dieser Stecker muß (außer Pin 18) mit allen seinen Pins mit den entsprechenden Pins aller Speicher-Sockel verbunden werden.

Da innerhalb einer Reihe diese Verbindungen per Layout schon bestehen, muß durch Drahtbrücken bzw. freie Verdrahtungen auf der Lötseite nur noch eine Querverbindung zwischen den entsprechenden Pins der 3 Reihen hergestellt werden.

Die nicht angeschlossenen Pins 18 aller Sockel werden mit je einem der Ausgänge 0-9, A-F des 1-aus-16-Decoders durch freie Verdrahtung auf der Lötseite verbunden. Der Pin 18 des Sockels, an den das Flachkabel führt, wird wird an die E*-Eingänge des Decoders gelegt. Die Eingänge I0- I3 des Decoders werden mit den Ausgängen Q0-Q3 des Latch 74116 verbunden.

Das Adapter-Board wird entspechend Abb.8 in das Flachkabel eingefügt. Dafür wird das Flachkabel an geeigneter Stelle durchgeschnitten, die einzelnen Leitungen auf eine Länge von ca. 2 cm voneinander getrennt und dann die Enden auf einer Länge von ca. 3 mm abisoliert und im Fall von Litzen schlank verdrillt und dünn eingezinnt.

Wenn die Ver-16-fachungs-Karte unter dem CPU-Board liegen soll und deshalb das Flachkabel nach Einstecken in den Sonder-ROM-Sockel vom CPU-Board wegführen soll, dann muß das Adapter-Board so orientiert sein, daß die Bestückungsseite zur gleichen Seite weist, wie die Stecker des Kabels. Die beiden in Kabelrichtung liegenden ICs (74LS00 und 74LS30) liegen dann links außen im Computer.

Im Gehäuse des Computers muß ein entsprechendere Pfosten auf der Bodenwanne abggesägt werden, damit das Board in das Gerät paßt.

4)Aufbau und Anschluß des Latch-Boards

Fast alle Signale für dieses kleine Board können vom Sockel des Sonder-ROMs (nicht von der darüber angeschlossenen Speicherkarte) abgenommen werden:

Signal	Anschlußpunkt Latch-Board	Pin Nr. am Sonder-ROM
+5V	1	24
WR*	2	#
IORQ*	3	#
A0	4	8
A1	5	7
A2	6	6
A3	7	5
A4	8	4
A5	9	3
A6	10	2
A7	11	1
SYSRES*	12	#

D0	13	9
D1	14	10
D2	15	11
D3	16	13
GND	17	12

siehe Text

Die Signale WR* und IORQ* werden an den Pins 11 und 9 von Z16 (74LS367) abgenommen und an die Latch-Board-Anschlüsse 2 und 3 gelegt.

Das Signal SYSRES* für den Anschluß 12 des Latch-Boards muß auf dem CPU-Board noch gebildet werden. Dafür muß in die Position Z26 (neben Z25, 74LS139) noch ein 74LS08 eingesetzt werden. Da diese Position für ein DIL-16-IC vorbereitet ist, muß je nach Einbau noch eine Verbindung mit +5V oder GND (am Lötauge außerhalb des ICs) hergestellt werden. Selbstverständlich muß das 74LS08 in gleicher Orientierung (Kerbe nach links) wie alle anderen ICs eingebaut werden.
Nun wird Pin 1 des 74LS08 mit Pin 26 des Z80 und Pin 2 mit Pin 17 des Z80 verbunden. Der Pin 3 des 74LS08 wird an den Anschluß 12 des Latch-Boards geführt.

Von den Anschlüssen 18 bis 21 des Latch-Boards werden Verbindungen zu den Pins 20 bis 23 des 1-aus-16-Decoders auf der Ver-16-fachungs-Karte hergestellt.

Latch-Board Anschluß	Latch-Board Signal	Ver-16-fachungs-Karte Pin des 74154	Ver-16-fachungs-Karte Signal
18	Q3	23	I3
19	Q2	22	I2
20	Q1	21	I1
21	Q0	20	I0

Für eventuelle spätere Service-Arbeiten ist es sinnvoll, wenn einzelne Boards nicht über Drähte direkt miteinander verlötet werden. Deshalb sollten an den vom Layout neben dem Decoder vorgesehenen Stellen Lötnägel eingesetzt werden und die Drähte mit passenden Buchsen aufgesteckt werden.

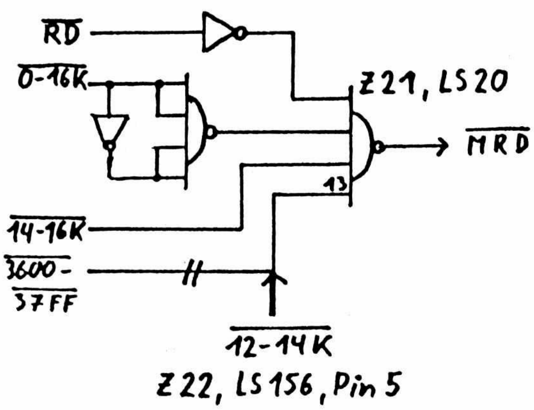


Abb.1:Freigabe des Speicher-lese- Treibers auf dem CPU- Board für den Betrieb der Ver-16-fach-ungs-Karte

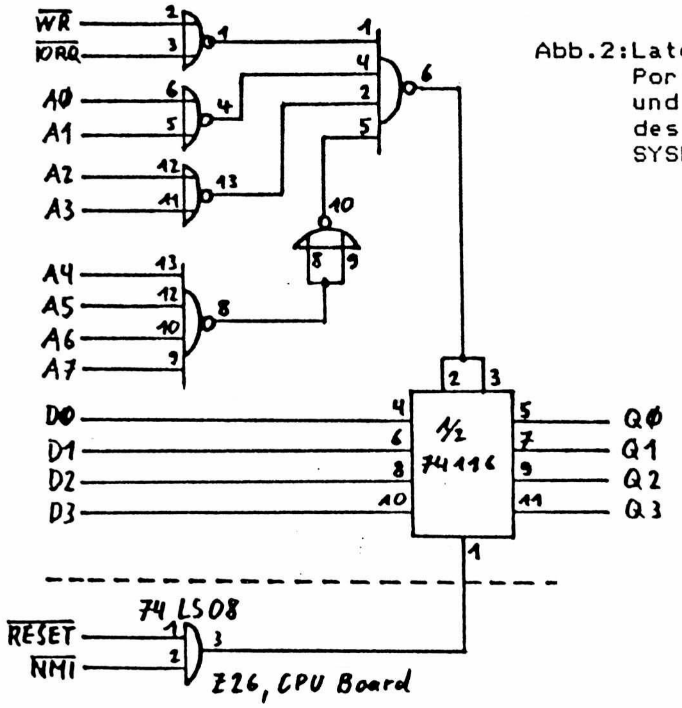


Abb.2:Latch-Board mit Portdecodierung und Erzeugung des Signals SYSRES*

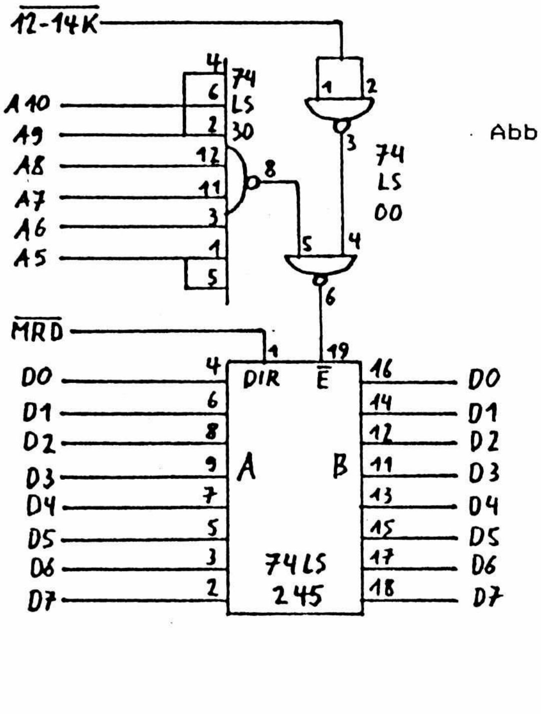


Abb.3:Adapter-Board mit vollständiger Adreßdecodierung und Datenbus-Transceiver

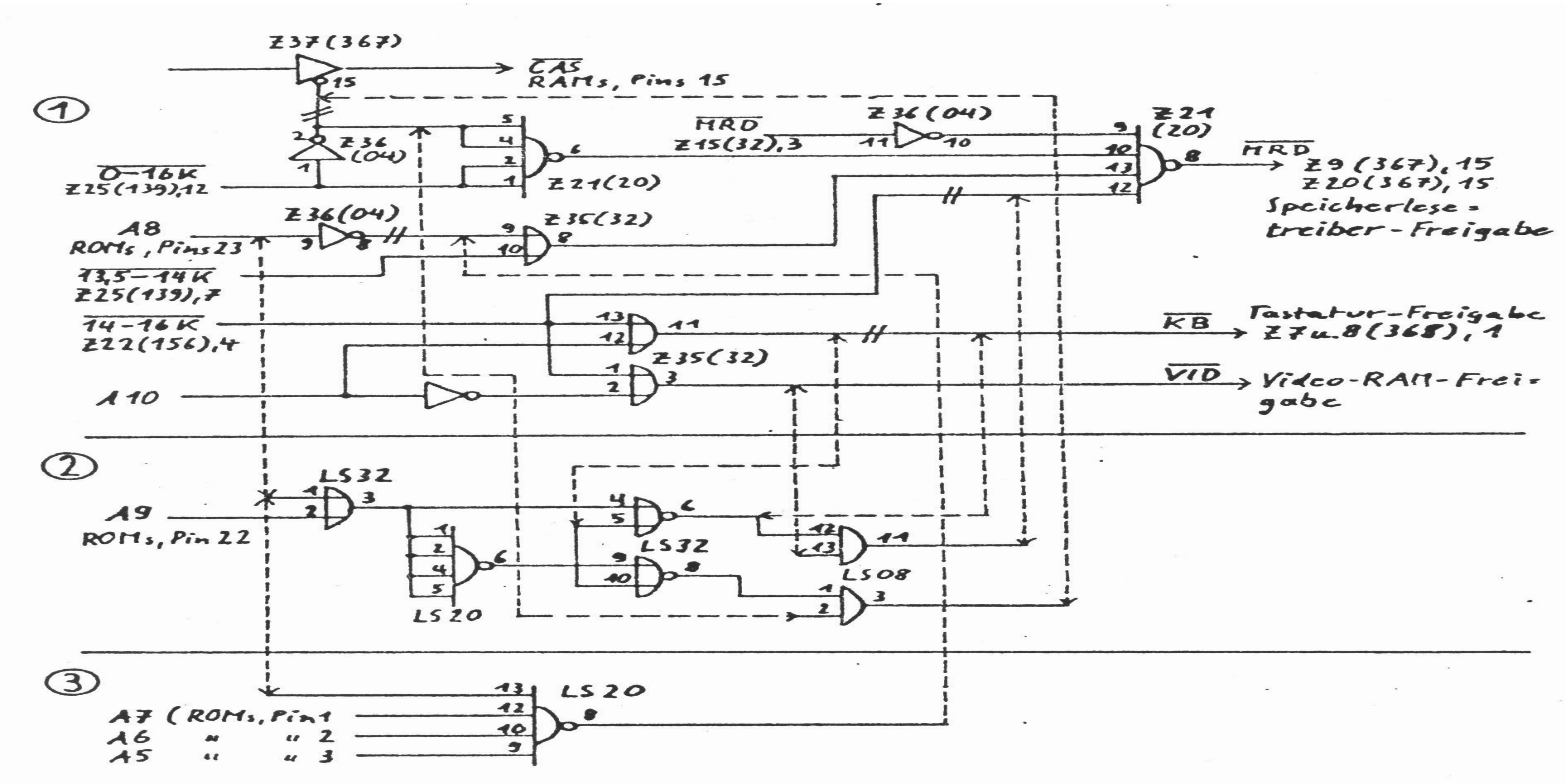
GENIE I und II: Volldecodierter Sonder-ROM; RAM im Bereich 3900H bis 3BFFH

Helmut Bernhardt

Der Begriff Kompatibilität, den man meistens im Zusammenhang mit IBM-PC-Nachbauten liest, spielte auch beim TRS-80-Nachbau GENIE eine wichtige Rolle. So wurde gegenüber den ersten "Video-GENIEs" später zusätzlich zu Port FDH auch die Adresse 37E8H für den Betrieb des Druckers eingerichtet. Leider ist auch viel Unsinn vom TRS-80 mit übernommen worden, so z.B. die Decodierung der Tastatur. Obwohl die Tastatur prinzipiell mit 8 Speicheradressen auskäme (oder sogar mit nur einer Portadresse), braucht sie wegen mangelnder Decodierung den gesamten Bereich 3800H-38FFH. Außerdem werden dabei auch noch die 3/4K Adreßraum 3900H bis 3BFFH verschenkt, ohne daß die Tastatur in der vorliegenden Decodierung dies erzwingt. Ein anderer TRS-80-Nachbau, der Komtek 1, hat die Tastatur auf den Bereich 3800H-38FFH beschränkt, ohne dabei (zumindest in Sachen Tastatur) die Kompatibilität zum TRS-80 einzubüßen. Mit dem Bereich 3900H-3BFFH wird aber dort auch nichts Sinnvolles gemacht.

Die Lücke innerhalb der unteren 16K zwischen ROM und Drucker-Floppy-Bereich, der Adreßraum 3000H bis 37DFH, wird im GENIE teilweise als Sonder-ROM genutzt. Durch mangelhafte Decodierung ist der Sonder-ROM im Bereich 3700H bis 37DFH nicht erreichbar.

Der Teil 1 der Schaltung zeigt die Bestandteile des CPU-Boards des GENIE, die für diese Mißstände verantwortlich sind. Die Abweichungen von den Darstellungen im Service-Manual sind keine Fehler sondern Richtigstellungen. Außerdem ist hier schon der Zustand nach Umrüsten auf 4164-RAMs zugrundegelegt, der für die Schaffung von RAM bei 3900H bis 3BFFH Vorraussetzung ist.



Die Freigabe des Tastatur-Lesetreibers erfolgt bei allen Adressen 3800H bis 38FFH. Die im Teil 2 dargestellte Schaltungsänderung beschränkt diese Freigabe auf die Adressen 3800H bis38FFH und bewirkt außerdem, daß die Freigabe des Speicherlesetreibers und des RAM auch im Adreßraum 3900H bis 3BFFH erfolgt, d.h. hier liegt RAM vor.

Teil 3 zeigt die für das vollständige Decodieren des Sonder-ROMs nötige Änderung. Nach dieser Änderung sind alle Befehle des Sonder-ROM (auch P) verfügbar. Wenn am GENIE ein EG64MBA oder Abkömmling davon betrieben wird, ist dadurch der Bereich 3900H bis 3BFFH nicht doppelt vorhanden. Bei Abschalten der Tastatur und Einblenden von RAM wird dann nur

noch zusätzlich der Bereich 3800H bis 38FFH als RAM verfügbar.

Wenn das "Sichere Plätzchen" nach c't 5/85 eingebaut ist, ist auch die o.a. Bereitstellung von RAM oberhalb der Tastatur nicht mehr nötig. Die sehr viel aufwendigere Schaltung aus c't hat den Vorteil, daß sie bei Betrieb des EG64MBA tatsächlich den Bereich 3900H bis 3BFFH zusätzlich zur Verfügung stellt. Der im Grundzustand oberhalb der Tastatur liegende RAM ist dort nicht der gleiche, der bei Abschalten der Tastatur verfügbar wird.

Umbauanleitung

In den Teilen 1-3 des Schaltplans sind alle ICs des CPU-Boards mit Nummer nach Bestückungsplan, Typ, und Pin bezeichnet. Auch die Herkunft sowie die Ziele der verwndeten Signale sind angegeben. Die in den Teilen 2 und 3 zusätzlich benötigten Gatter können durch 3 weitere ICs (74LS08, 20, 32) zur Verfügung gestellt werden. Diese Gatter haben im Schaltplan nur eine Typenbezeichnung.

Bei diesen zusätzlich einzubauenden ICs werden die Pins 1-6 sowie 8-13 hochgebogen. Mit den Pins 7 und 14 werden sie in geeigneter Anordnung an den Pins 7 und 14 der ICs Z35, Z36 und Z21 angelötet und durch Verbindung der bezeichneten Pins untereinander und mit den entsprechenden Pins des CPU-Boards mit isolierten Drähten beschaltet. Die Orientierung der Huckepack-ICs muß die gleiche sein wie die der ICs darunter.

Außerdem sind folgende Leiterbahnen zu durchtrennen bzw. Verbindungen zu legen:

1. Verbindung von Pin 15 von Z37(367) mit Pins 4 und 5 von Z21(20);
   Pin 2 von Z36(04) ist nun direkt mit den Pins 4 und 5 von Z21(20) zu verbinden
2. Verbindung von Pin 2 von Z36(04) mit Pin 15 von Z37(367);
   diese Verbindung liegt nach dem Umrüsten auf 64K als Drahtbrücke vor
3. Verbindung von Pin 4 von Z22(156) mit Pin 12 von Z21(20)
4. Verbindung von Pin 11 von Z35(32) mit den Pins 1 von Z7 und Z8(368)
5. Verbindung von Pin 9 von Z35(32) mit Pin 8 von Z36(04)

Die Maßnahmen 1 bis 4 sind nötig, um den zusätzlichen RAM im Bereich 3900H bis 3BFFH zu erhalten, während 5 für die vollständige Decodierung des Sonder-ROM durchgeführt werden muß.

Eines der nicht benutzten AND-Gatter des zusätzlichen 74LS00 läßt sich zur Erzeugung eines SYSRES*-Signals aus RESET* (Z80-CPU, Pin 26) und NMI* (Z80-CPU, Pin 17) benutzen. Dieses Signal wird bei vielen Hardware-Erweiterungen benötigt. (Beim EXP1 von RB-Elektronik mit Doubler sollte dieses Signal anstelle des vorgeschlagenen RESET* verwendet werden.)

Dein Genie 3s, die 5 unbekannten Wesen

Nun soll auch ein Z180-Prozessor Chef im System sein können

Helmut Bernhardt, Arnulf Sopp

Vorbemerkung: Im Folgenden ist viel von "wir" die Rede. Dies aus Gründen der Sprachökonomie. Jeder von uns machte aber in Wirklichkeit das, was er am besten konnte: Helmut die Hardware, Arnulf die Software. Das Pflichtenheft für beides entstand unter Mitwirkung etlicher Gebinde Hopfenbrause.

* * * * * * * * *

Die Hardware

Vor längerer Zeit hatten wir unter ähnlichem Titel mal eine Erweiterung für den Genie 3s beschrieben (1), mit der sämtliche Speicher (User-RAM, Video-RAM, HRG-RAM und Zeichengenerator-RAM) auf die vierfache Größe erweitert werden konnten. Außerdem fiel dabei noch ein SRAM ab, das parallel zum Boot-EPROM arbeitete. Dafür haben wir in den Sockel der Z80-CPU ein Adapter-Board gesteckt, das den Z80 aufnahm und einen schreib- und lesbaren Port bereitstellte, mit dessen Bits die Umschaltung der Speicherbanks gesteuert wurde.

Über dieses primitive Banking kann z. B. ein Genie 2s-Anwender, der auf eine HD64180-CPU umgerüstet hat (2), nur milde lächeln, wenn er seinen Computer mit 9,2 MHz (gegenüber 7,2 MHz des G3s) fährt. Es galt, dieses Manko beim G3s auszubügeln. Dafür haben wir ein neues Adapter-Board gebaut, auf dem nun kein Z80 (auch kein HD64180), sondern ein Z180 steckt. Der Z180 hat (wie auch der neuere HD64180-PLCC) 20 Adreßleitungen und kann damit über seine MMU und die beiden DMA-Kanäle 1 Megabyte RAM direkt adressieren.

Um diese CPU so, wie sie gedacht ist, direkt zu nutzen, hätte das riesige Betriebssystem G-DOS 2.4 (oder Calva-DOS 2.4, die auf 1 Megabyte angepaßte Version) vollkommen auf HD64180-MMU-Banking umgeschrieben werden müssen. In über 64 kB SYS- und anderen systemeigenen Files hätten alle OUT-Befehle an die System-Steuerports F9h und FAh gesucht und auf entsprechende Z180-MMU-Befehle geändert werden müssen. Außer viel Arbeit hätte das auch die Kompatibilität mit anderen Computern dieser und ähnlicher Bauart gekostet. Da entschieden wir uns dafür, dem Calva-DOS sein Banking über die Systemsteuerports zunächst mal zu erhalten. Die Ports F9h, FAh und FBh (früher F3h) auf dem Adapter-Board haben ihre volle Funktionsfähigkeit behalten.

Nun ist aber der Einsatz eines HD64180 bzw. Z180 witzlos, wenn dessen MMU und DMAC nicht benutzt werden können. Wir haben beide Möglichkeiten eingerichtet: Nach dem Einschalt-Reset sind alle Systemport-Bits auf 0 gesetzt, und ein voll G3s-kompatibler Computer incl. Erweiterung auf vierfache Speichergröße liegt vor. Durch Setzen des früher anders belegten Bits D6 in Port FBh kann aber umgeschaltet werden in einen lupenreinen Z180-Computer; alle Banking- und Common-Definitions-Bits der Systemports haben dann keine Funktion mehr. Das RAM wird ausschließlich durch die Adreßleitungen der CPU adressiert.

Nur das Einblenden von memory mapped I/O-Baugruppen in die dafür vorgesehenen physikalischen Adreßbereiche erfolgt, wie im G3s-Modus, durch die entsprechenden Systemport-Bits. Gleichwohl sind deren logische Adressen variabel. Der G3s dekodiert bei ihrer Selektion nur die Adreßbits 0-15. Wie

der Z180 derweil sein physikalisches Megabyte in bis zu drei logische Adreßbereiche unterteilt, indem er mit A12-19 jongliert, da redet ihm der G3s nicht drein.

Um die Ausmaße eines Adapter-Boards mit diesen Umschaltmöglichkeiten (s. Foto) in Grenzen zu halten, mußten wir zwei PALs einsetzen. PAL 1 erzeugt lediglich die Port-Freigabesignale /OUTFB und /INFB (und noch einige weitere Freigabesignale - s. PAL-Listing), mit denen das Latch 74LS273 und der Lesetreiber 74LS244 angesteuert werden, die den zusätzlichen Port FBh realisieren.

Die eigentliche Steuerung der RAM-Adressierung leistet PAL 2. Durch die Ausgänge /CAS0-/CAS3 wird eine der vier Reihen von 41256er RAM-Chips selektiert. A16 und A17 wählen darin einen 64 kB-Block an. Für die Adressierung im Z180-Modus gelangen die Adressen A16-A19 an das PAL. Im G3s-Modus erfolgt das Banking durch die Bits D6 und D7 von Port F9h sowie D0 und D1 von Port FBh. D7 von Port FBh entscheidet, ob bei logischer Adressierung im eingestellten Common-Bereich (0000h-3FFFh oder E000h-FFFFh) nur innerhalb eines 256 kB-Bereichs oder innerhalb des gesamten 1 MB-Speichers der unterste 64 kB-Block adressiert werden soll.

Der Zugriff auf den eingestellten Common-Bereich wird durch das high aktive Signal COMLH (U84, 74LS00, Pin 11 auf dem CPU-Board) angezeigt. Dieses Signal wird ebenfalls an das PAL geführt (wie auch D7 von Port FBh, das die Auswirkung von COMLH steuert). Durch D6 von Port FBh erfolgt schließlich die Steuerung, ob der G3s-Modus oder der Z180-Modus für die Adressierung des RAMs zuständig ist. Das PAL-Assemblerlisting veranschaulicht alle diese Steuerfunktionen.

Damit die Signale /CAS0-/CAS3 anstelle der bisherigen Signale /CAS0-/CAS3 an die RAMs gelangen, ist U43, 74LS125 zu entfernen. Die Signale /CAS0-/CAS3 des Adapter-Boards werden den RAMs über die Pins 3, 6, 11 und 8 der Fassung von U43 zugeführt. Die an diesen Pins liegenden Pull up-Widerstände R57-R60 sollten ausgelötet werden. Die Versorgung mit +5V und GND kann von den Pins 14 und 7 derselben Fassung erfolgen. Und schließlich liefert ihr Pin 9 auch noch das Signal /CAS für den Adapter.

Alle Signale, die dem Adapter-Board vom CPU-Board zugeführt werden müssen, werden über den Stecker CN2 geführt; ebenso die Signale, die das Adapter-Board dem CPU-Board liefert. Nur die ursprünglichen Z80-Pin-Signale laufen über den Sockel des Z80.

Der Einsatz eines Z180 bringt aber auch neue Probleme mit sich. Beim Z80 wird der Systemtakt extern erzeugt und der CPU über Pin 6 eingespeist. Beim HD64180 und beim Z180 wird der Systemtakt von der CPU geliefert. Dieser Takt wird intern aus einem von außen zugeführten, doppelt so hohen Takt hergeleitet.

Der G3s hat die Möglichkeit (leider auch die Notwendigkeit), zwischen einer hohen Frequenz und den normalen 1,77 MHz des TRS-80 umzuschalten. Diese Umschaltung muß bereits mit dem doppelten Muttertakt erfolgen. Es muß also zwischen 18,432 und 3,55 MHz umgeschaltet werden. Dadurch kann die CPU nicht mehr mit einem normalen Quarz an den Pins XT und EXT beschaltet werden. Es müssen externe Oszillatorschaltungen benutzt werden. Der jeweils selektierte Takt wird über einen Treiber (74LS125) an den Pin EXT der CPU geschaltet. Die Auswahl des jeweiligen Taktes wird durch die Signale LO (U41, 74LS74, Pin 9) und HI (U41, 74LS74, Pin 8) auf dem CPU-Board gesteuert.

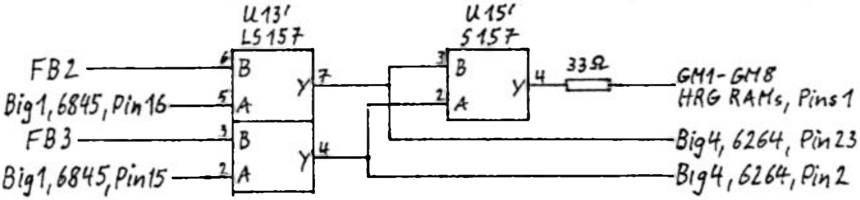
Die bisherige Taktquelle im System wird durch Entfernen von U44, 74LS161 abgeschaltet. Damit der nun von der CPU gelieferte Systemtakt auch auf dem CPU-Board verfügbar wird, müssen die Pins 5 und 6 von U39, 74LS125 aus der Fassung gebogen werden und die Pins 5 und 6 der Fassung miteinander verbunden werden. Die Widerstände R51, R52 und R79 sollten entfernt werden.

Die für das ursprüngliche 256 kB-Banking des G3s verantwortlichen Bits 6 und 7 des Ports F9h (hier als Signale F96 und F97 bezeichnet) sind von den Pins 14 und 13 von U88, 74LS139 zu beziehen.

Die vom Port FBh auf dem Adapter-Board zur Vervierfachung von Video-RAM, Zeichensatz-RAM und HRG-RAM gelieferten Bits 2-5 (FB2-FB5) werden in dieser Reihenfolge an folgende IC-Pins der Hauptplatine geführt:

U13', 74LS157, Pin 6
U13', 74LS157, Pin 3
BIG5 und BIG6, 6264, Pins 23 (diese Pins aus der Fassung biegen)
BIG5 und BIG6, 6264, Pins 2 (diese Pins sind bereits außerhalb der Fassung)

U13' ist ein mit den Pins 1, 8, 15 und 16 huckepack auf U13, 74LS157 gelötetes IC 74LS157. Die weitere Beschaltung dieses ICs gibt folgende Abbildung wieder:



U15' ist ebenfalls ein auf U15, 74LS157 mit den Pins 1, 8, 15 und 16 huckepack gelötetes weiteres IC 74S157.

Die Pins 15 und 16 von BIG1, 6845 sind aus der Fassung zu biegen und mit U13' zu verdrahten.

Schließlich sind noch die Adressen A16 und A17 von CN2 den RAMs gemultiplext zuzuführen. Das Multiplexen besorgt ein auf U34, 74S157 wiederum mit den Pins 1, 8, 15 und 16 huckepack gelöteter 74S157 (U34'), der an den Pins 13 und 14 mit A16 und A17 zu versorgen ist. Der Multiplex-Ausgang Pin 12 wird über 33 Ω mit den Pins 1 aller RAMs (32x 41256) verbunden.

Zur Vereinfachung des bis hierhin Gesagten sei das alles noch einmal in einer Tabelle wiedergegeben:

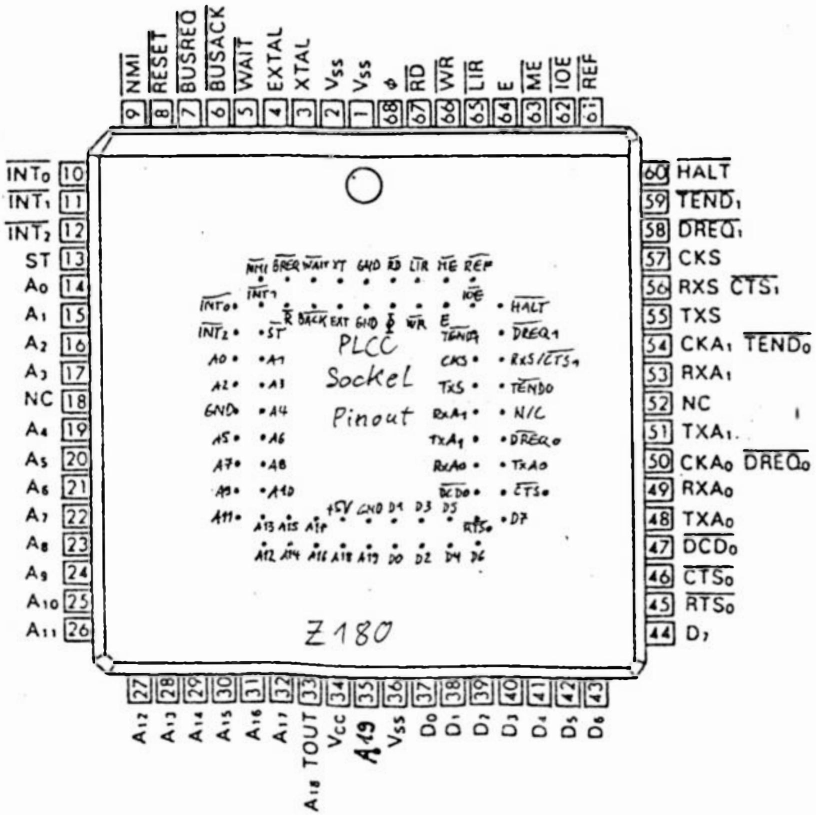
Verbindungen von CN2 zum CPU-Board

Stecker CN2		CPU-Board				
Pin	Signal	IC	Typ	Pin		
1	+5V	U43	LS125	14	*	#. huckepack gelötete
2	GND	U43	LS125	7	*	ICs 74LS(S)157
3	/HI	U41	LS74	8		(s. Text und Abbil-
4	/LO	U41	LS74	9		dungen
5	FB3	U13'	LS157	3	#	
6	FB2	U13'	LS157	6	#	
7	FB5	BIG5/BIG6	6264	2		
8	FB4	BIG5/BIG6	6264	23	(aus der Fassung biegen)	
9	F96	U88	LS139	14		
10	F97	U88	LS139	13		* U43 entfernen und
11	/CAS0	U43	LS125	3	*	Sockel auslöten,
12	COMLH	U88	LS139	15		Signale von CN2 an
13	/CAS2	U43	LS125	11	*	die Lötpunkte der
14	/CAS3	U43	LS125	8	*	entsprechenden Pins

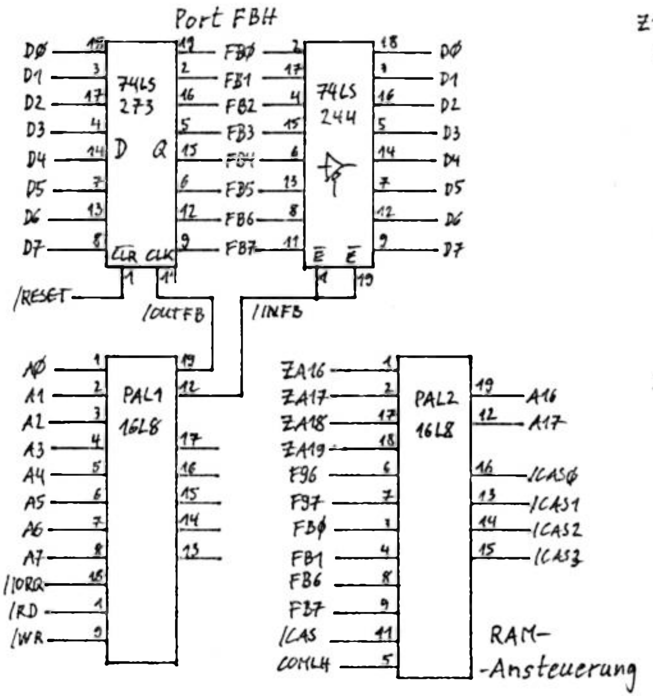
15	/CAS	U43	LS125	9 •	des Sockels von U43
16	/CAS1	U43	LS125	6 •	anlöten
17	A16	U34'	S157	13 #	
18	A17	U34'	S157	14 #	

Außerdem sind auf dem CPU-Board noch folgende (z. T. beim Einbau des 4x G3s-Z80-Adapters schon durchgeführten) Änderungen nötig:

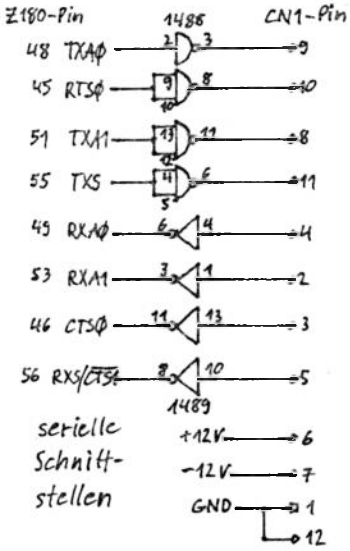
Die aus den Fassungen gebogenen Pins 1 von U16, U15 und U15' (alle 74S157) sind mit Pin 5 von U57, 74S04 zu verbinden. Die Pins 27 von BIG5 und BIG6 (6264), die beide außerhalb der Fassung hängen, werden mit Pin 1 von U5, 74LS244 verbunden. Pin 27 von BIG4, 6264 wird mit Pin 3 von U54, 74LS32 verlötet.



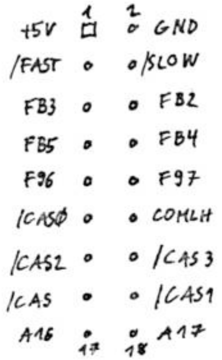
Pinout des Z180 und seines Sockels



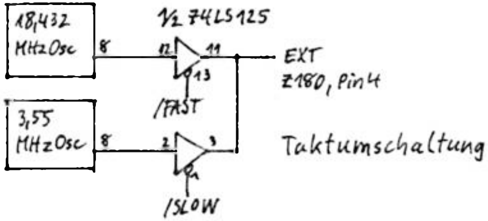
RAM- -Ansteuerung



serielle Schnittstellen



CN2 - Belegung



Taktumschaltung

Z180 Pin	Signal	Z80-Adapter Pin
14	A0	30
15	A1	31
16	A2	32
17	A3	33
18	A4	34
19	A5	35
20	A6	36
21	A7	37
22	A8	38
23	A9	39
24	A10	40
29	A14	4
30	A15	5
40	D3	8
42	D5	9
43	D6	10
10	/INT0	16
9	/NMI	17
66	/WR	21
62	/IORQ	20
7	/BUSRQ	25
6	/BUSAK	23
8	/RESET	26
65	/M1	27
5	/WAIT	24
61	/RFSH	28
68	CLK	6

von Hand zu verdrahten zwischen Z180-CPU und Z80-Adapter-Stecker

Die Listings der beiden PALs

```
PAL16L8
10.03.90    H. Bernhardt
Portdecodierung im GIIIS mit Z180 fuer zusaetzl. Systemports

A0 A1 A2 A3 A4 A5 A6 A7 WR GND
RD OUTFB OUTFC OUTF3 IODC INF3 INFC IORQ INFB VCC

/INFB  = /IORQ * /RD * A7 * A6 * A5 * A4 * A3 */A2 * A1 * A0
/OUTFB = /IORQ * /WR * A7 * A6 * A5 * A4 * A3 */A2 * A1 * A0
/INFC  = /IORQ * /RD * A7 * A6 * A5 * A4 * A3 * A2 */A1 */A0
/OUTFC = /IORQ * /WR * A7 * A6 * A5 * A4 * A3 * A2 */A1 */A0
/INF3  = /IORQ * /RD * A7 * A6 * A5 * A4 */A3 */A2 * A1 * A0
/OUTF3 = /IORQ * /WR * A7 * A6 * A5 * A4 */A3 */A2 * A1 * A0
/IODC  = /IORQ       * A7 * A6 */A5 * A4 * A3 * A2
```

```
PAL16L8
10.03.90     H. Bernhardt
RAM-Steuerung im GIIIs mit Z180

ZA16 ZA17 FB0 FB1 COMLH F96 F97 FB6 FB7 GND
CAS A17 CAS1 CAS2 CAS3 CAS0 ZA18 ZA19 A16 VCC

/A16    = /FB6 * /F96
        + /FB6 * COMLH
        +  FB6 * /ZA16
/A17    = /FB6 * /F97
        + /FB6 * COMLH
        +  FB6 * /ZA17
/CAS0   = /FB6 * /CAS * /FB1  * /FB0
        + /FB6 * /CAS * COMLH * /FB7
        +  FB6 * /CAS * /ZA19 * /ZA18
/CAS1   = /FB6 * /CAS * /FB1  *  FB0 * /COMLH
        + /FB6 * /CAS * /FB1  *  FB0 * FB7
        +  FB6 * /CAS * /ZA19 *  ZA18
/CAS2   = /FB6 * /CAS *  FB1  * /FB0 * /COMLH
        + /FB6 * /CAS *  FB1  * /FB0 * FB7
        +  FB6 * /CAS *  ZA19 * /ZA18
/CAS3   = /FB6 * /CAS *  FB1  *  FB0 * /COMLH
        + /FB6 * /CAS *  FB1  *  FB0 * FB7
        +  FB6 * /CAS *  ZA19 *  ZA18
```

Die Software

Der Z180 verleitet mit seiner interessanten MMU (Memory Management Unit) und seinem DMAC (Direct Memory Access Controller) geradezu zu Spielereien. Der Leser sei versichert, daß wir ihnen auch ausgiebig frönten. So ist es z. B. kein Problem, den Bildschirm oder irgendeinen anderen physikalischen Speicherabschnitt logisch gleich dreimal an verschiedenen Adressen im 64 kB-Fenster der CPU vorliegen zu haben. Hier soll aber ganz bierernst nur vorgestellt werden, was an Änderungen im DOS und Unterstützungen für das DOS programmiert wurde, um die Vorteile der neuen CPU nutzen zu können.

Zum Verständnis dessen, was sich nun eigentlich zum Besseren ändert, möchten wir außer den im Hardware-Teil schon genannten Veröffentlichungen noch auf ein Sonderheft des Club-80 (3) und einen Info-Artikel (4) hinweisen, die zum Thema HD64180 erschienen sind. Diese CPU ist softwaremäßig identisch mit dem Z180. Eine erneute Beschreibung der im wesentlichen gleichen Eigenschaften erübrigt sich an dieser Stelle.

Es sind vier Programme, die für den Z180 geändert bzw. neu erstellt wurden:

Patches in zwei verschiedenen Teilen von SYS0/SYS konfigurieren den Z180, um seine Arbeitsgeschwindigkeit und seine MMU auf optimale Werte einzustellen und um seinen DMAC dem System verfügbar zu machen.

SYS26/SYS ist zuständig, um die SYS-Files des DOS in Bank 0 des Speichers zu puffern. Es wird nun nur noch angesprungen, wenn nicht der Rechtspfeil gedrückt wurde (was bedeuten würde, daß das DOS nicht gepuffert werden soll). In SYS26 steht nun der DMA-Treiber aus dem neuen SYS0 bereits zur Verfügung, was alle Blockladevorgänge beschleunigt. Die Pufferung der DOS-Moduln benötigt jetzt nicht mehr das aufwendige Banking, das bisher erforderlich war, denn der DMAC kann von überall her den vollen Adreßbereich des Z180 überstreichen.

Eine Änderung in der SYS-Laderoutine für den DOS-Kern, ebenfalls in SYS26 untergebracht, lädt nun die SYS-Programme bei einem DOS-Request mit RST 28h per DMA entsprechend schneller und weniger speicheraufwendig. Daß im Zuge dessen viel Platz in SYS26 frei wurde, der edleren Zwecken zur Verfügung gestellt werden kann, sei nebenbei erwähnt. Zudem ist der besondere Sektorpuffer 3A00h-3AFFh für die SYS-Moduln jetzt nicht mehr notwendig - freier Speicher für andere Anwendungen.

Dann entstand ein Anwenderfile (bei uns TRAP/CMD geheißen), das hauptsächlich Gerald Schröders Illegal-Trap-Programm in Bank 0 und Bank 1 an die logische Adresse 0000h patcht. Hier springt der Z180 nämlich hin, wenn er einen Befehl antrifft, den er nicht kennt (für nähere Erläuterungen möchten wir auf Geralds Artikel im Info Nr. 17 verweisen). Außerdem wird hier die Delay-Routine an 0060h verlangsamt, denn der Z180 arbeitet zu schnell, um mit dem alten Verzögerungsprogramm noch die korrekten Wartezeiten zu erzielen.

Und schließlich zimmerten wir ein Utility-Programm namens SYSTEST/CMD für den DOS-Tüftler (Listing 5). Wenn ein SYS-File umgeschrieben wurde und nach Murphy's Gesetz noch Fehler enthalten muß, kann es zunächst in der Bank 0 gepuffert und (notfalls bei offenem Floppy-Türchen) von dort aus getestet werden. Damit erübrigt sich gleichzeitig die Notwendigkeit, vor dem Test erneut zu booten, um die Datei in die Bank 0 zu kriegen, von wo sie beim gebankten DOS aufgerufen wird.

Alle Änderungen in SYS0 und SYS26 sowie die neuen Files TRAP/CMD und SYSTEST/CMD sind in den Listings wiedergegeben. Weite Teile, die sich in den beiden SYS-Programmen nicht geändert haben, sind mit LIST OFF ausgespart worden, um den Rahmen dieses Sonderhefts nicht zu sprengen. Dasselbe gilt

für TRAP/CMD, soweit es Geralds bereits bekanntes Trap-Programm betrifft. Damit hat der Leser natürlich die Schwierigkeit, daß unklar bleibt, wo die Patches in den Files liegen sollen, und was sich in der Folge sonst noch ändert. Sorry - da können wir nur anbieten, eine Systemdiskette bei Arnulf anzufordern. Die Listings sollen lediglich den Umgang mit den neuen Möglichkeiten des Z180 demonstrieren und die sich ergebenden Vorteile besingen.

Es werden auch kleinere Teile der Listings unverständlich bleiben, weil sie sich auf nicht gelistete Programmabschnitte beziehen. Es handelt sich nämlich nicht um besondere Schöpfungen für dieses Sonderinfo, sondern um die Sources für das geänderte DOS. Wenn etwa Labels auftauchen, die sonst nirgends wiederzufinden sind, einfach darüber hinweglesen!

Die genannten Programme sind auch noch längst nicht der Weisheit letzter Schluß. So ist es beispielsweise wünschenswert, einige Konfigurationen des Z180 (WAIT-Zyklen, Refreshes, MMU, vielleicht noch mehr) bereits im Boot-EPROM erledigen zu lassen. So weit sind wir jedoch selbst noch nicht. Gegf. wird man in einem kommenden Clubinfo Weiteres dazu vorfinden.

Soweit Befehle enthalten sind, die zwar der Z180, nicht aber der Z80 kennt, sind sie mit ihren Hex-Entsprechungen als DB-(DEFB-)Statements eingefügt. ZEUS kennt sie nämlich bedauerlicherweise auch nicht. In einem Kommentar erscheint dann jeweils der Befehl in Z180-Assembler.

Die Programme sind essentiell für das Betriebssystem. Ohne Betriebssystem kein Computer. Daher wird unbedingt empfohlen, alles zunächst mit einer Kopie der Systemdiskette auszuprobieren (für SYS29 kann aber auch SYSTEST gefahren werden!). Erst, wenn sich die Kopie als funktionstüchtig erwiesen hat, hat die alte Systemdisk ausgedient!

Nun zu den Programmen im einzelnen:

SYS0/SYS ist die Seele des G3s. Es ist in seinem ersten Teil eine Fortsetzung von GDOS/SYS. Hier ist die erste Modifikation für den Z180 untergebracht: Die Anzahl der WAIT-Zyklen für Speicher- und Portzugriffe wird eingestellt, und zwar auf den schnellstmöglichen Wert. Sollte er sich im Einzelfall als Überforderung des Systems herausstellen, kann der Wert 00h für den Akku mit DDE auf einen anderen Betrag gepatcht werden (XOR A hätte den Akku ebenfalls gelöscht, jedoch ohne die Möglichkeit einer einfachen Änderung). Danach wird die MMU des Z180 auf den Normalzustand des G3s eingerichtet. So kann, etwa für einen DMA (s. u.), die Z180-Speicherkonfiguration in aller Regel geschaltet werden, ohne daß sich etwas am Banking-Zustand ändert.

Die zusätzlichen Befehle (s. Listing 1) werden einfach in die (zuvor per Disassembly gewonnene) Source eingefügt, das Programm neu assembliert. Es besteht aus zwei Sektoren, die in die ersten beiden Sektoren von SYS0/SYS gepatcht werden (das kann z. B. SUPERZAP).

Das Listing 2 zeigt ebenfalls einen Patch in SYS0. Anfangs wird getestet, ob der Rechtspfeil während des Bootens gedrückt wurde. Ist das nicht der Fall, möchte der User das DOS in Bank 0 des Computers puffern, um es später ohne die Laufwerke verfügbar zu haben. Dieser Test war bisher in SYS26 untergebracht. Es erschien uns überflüssig, SYS26 anzuspringen, wenn es bei gedrücktem Rechtspfeil ohnehin gleich wieder verlassen wird.

Dann enthält das Listing einen Treiber für den DMAC des Z180. Er erwartet beim Einsprung in HL die Anfangsadresse des zu transferierenden Blocks, in DE die Zieladresse, in BC den Bytezähler. Bis hierher entspricht der Treiber dem Z80-Befehl LDIR. Neu ist der Inhalt des Akkus, der in seinen beiden Nibbles die Adressen A16-19 des Quell- und des Zielbereichs enthält. So kann ein DMA auf das volle Megabyte des Z180-Adreßraums zugreifen.

Die internen DMAC-Register SAR0B (Kanal 0, Quelle, Bits 16-19) und DAR0B (Kanal 0, Ziel, Bits 16-19) werden nur in ihren unteren 4 Bits beachtet. So war es zur Laufzeit- und Platzersparnis möglich, auf das Löschen der oberen 4 Bits zu verzichten: Der Befehl RRD (Zeile 94) rotiert die hier relevanten Bits 4-7 ins untere Nibble von DAR0B, der folgende Befehl LD (srcphys),A packt einfach den kompletten Akku mit seinem unwichtig gewordenen oberen Nibble nach SAR0B.

In Zeile 102 wird die G3s-MMU deaktiviert, indem in Port FBh das Bit 6 gesetzt wird (s. Hardware-Teil). Nur so kann der DMA-Treiber über das ganze Megabyte verfügen. Ansonsten könnte nur das 64 kB-Fenster, das gerade vom G3s bearbeitet wurde, vom DMA überstrichen werden. Also ist beim Einsprung in die DMA-Routine unbedingt darauf zu achten, daß die MMU-Konfiguration des Z180 mit der des G3s zumindest in wichtigen Speicherabschnitten übereinstimmt: Der Treiber muß sich nach dem Umschalten selber wiederfinden können, der Stack muß ebenfalls an derselben logischen Adresse vorliegen. Erst nach dem Rückschaltbefehl (Zeile 111) ist Entwarnung gegeben.

Bei der Rückkehr aus dem Programm enthalten alle Register ihren alten Wert, der gerettet wurde. Im Gegensatz dazu gibt der Befehl LDIR die Quell- und Zielregister um den Betrag des Bytezählers erhöht zurück, BC selbst ist dann 0000h. Dieser Service hätte mit in den Treiber eingebaut werden können, der verfügbare Platz reichte jedoch nicht. Zudem haben beide Möglichkeiten ihre Vor- und Nachteile.

An dieser Portion von SYS0 ist nur der filerelative Sektor 10h interessant. Das ist der diskrelative Sektor 15h. Die Source gem. Listing 2 wird assembliert. Von der so entstandenen CMD-Datei muß nun (mit SUPERZAP oder wie auch immer) der erste Sektor nach SYS0 kopiert werden, und zwar in den eben bezeichneten Sektor. Wichtig ist, daß dieser Sektor des CMD-Programms genau am relativen Byte FFh aufhört, also ganz unten rechts. Danach, im nächsten Sektor, folgt der Rest, der mit einem neuen Record-Header beginnt. Bis dorthin können NOPs bzw. DW 0000h- oder DB 00h-Statements eingefügt werden. Um das Puzzle zu erleichtern, folgt hier ein Sektor-Dump, der den relativen Sektor 10h von SYS0 wiedergibt:

```
file: SYS0/SYS      drv: 0    frs: 0010h       drs: 0015h
00: 018C 0032 3E40 D3F9 FBC1 D1E1 C9E5 D5C5     2>@
10: 3E01 F3D3 F9C3 0AF0 3A07 43F6 3032 8232 >         : C 02 2
20: E60F FE04 2011 3AF8 42CB 5F28 0A3A 4038         : B  ( :@8
30: E640 3EFC CC02 4421 7F32 1180 4401 0B00  @>    D! 2   D
40: EDB0 1180 4421 0042 CD24 4428 0BFE 1820      D!  B $D(
50: 2B3E 0132 0743 180E EBCD 284C 201E 1162 +> 2 C      (L    b
60: 32D5 1198 31E9 2100 3011 AD4E 0100 023E 2    1 ! 0   N    >
70: 10CD BB35 E1F1 32F9 42C3 6744 F5CD C901     5  2 B gD
80: 3E46 EF4F 564C 782F 5359 533A 300D 0170 >F OVLx/SYS:0   p
90: BB35 22F0 35ED 53F3 35ED 43F6 35E5 21F5  5" 5 S 5 C 5 !
A0: 3577 ED67 32F2 3521 F735 C501 2708 F5DB 5w g2 5! 5   '
B0: FBF5 F640 F3D3 FBED 9B01 3102 ED9B F1D3    @        1
C0: FBF1 C1E1 C940 0200 0000 0000 0000 0000     @
D0: 0000 0000 0000 0000 0000 0000 0000 0000
E0: 0000 0000 0000 0000 0000 0000 0000 0000
F0: 0000 0000 0000 0000 0000 0000 0000 0000
```

Es ist ausreichend, das im Hex- und im ASCII-Teil unterstrichene Zählbyte des Record-Headers (rel. Sektorbyte 8Fh) genau so zu zappen und den Rest auszunullen. Dann dürfte nichts mehr schiefgehen. Voraussetzung ist, daß das Zählbyte auch an dieser Stelle steht. Ansonsten muß es eben auf andere Weise angepaßt werden, damit das letzte Datenbyte des Sektors an der sektorrelativen Stelle FFh zu liegen kommt.

Listing 3 behandelt ein paar Änderungen in SYS26. Dort wird, wie schon erwähnt, das Puffern des DOS im Speicher veranlaßt. Gleichzeitig wird der DOS-Kern gepatcht: Soll ein SYS-File mit RST 28h geladen werden, wird nicht

mehr die Floppy angeworfen. Statt dessen wird das Programm aus der Bank 0 geholt.

Dafür dient ein besonderes Minimal-DIR: Pro SYS-Modul enthält es zwei Bytes. Das erste ist das MSB der Pufferadresse in Bank 0. Da das LSB immer 00h ist, da also jedes SYS-File am Anfang eines 256-Byte-Blocks beginnt, genügt das MSB. Das zweite Byte gibt die Länge des Files in Sektoren an. Gepuffert werden nur die Programme SYS1-29. GDOS/SYS, INHALT/SYS und SYS0/SYS haben hier keine Bedeutung und bleiben deshalb auf der Floppy. Ihr Platz im Mini-DIR ist mit Nonsense (GDOS) bzw. mit Nullen (INHALT und SYS0) gefüllt.

Eine spezielle Laderoutine für SYS-Programme liest dieses DIR und transferiert dann die Sektoren ursprünglich nach 3A00h, einen besonderen Sektorpuffer. Von dort wird das File sektorweise in den Overlay-Speicher des DOS geladen und schließlich angesprungen.

Hier die Änderungen, die die Vorteile des Z180 ausnutzen: Der DMAC benötigt kein Banking, um Daten zu transportieren. Die Unterprogramme, die das DOS in den Speicher laden und es später beim Betrieb von dort wieder abholen, konnten deshalb wesentlich verkürzt (und damit beschleunigt) werden.

Das UP putsys puffert die SYS-Files. Dazu wird zunächst der Teil von INHALT/SYS ab 5200h in den Speicher geladen, der SYS-Programme enthält. Der Platzbedarf von 0800h Bytes zerstört dabei nach Reset den Inhalt des Anwenderspeichers. Daher wird (ab Label boot) zunächst dieser Bereich in die Bank 1 gerettet, um später restauriert zu werden. Dann wird das besondere SYS-DIR im Block 3900h eingerichtet. Die ersten Bytes des Blocks müssen ausgenullt werden, damit bei einem DOS-Request nicht versucht wird, INHALT oder SYS0 aus der Bank 0 zu holen; die Nullen sind Signalbytes für das Verbot. Anschließend wird ab 4100h in Bank 0 das restliche DOS abgelegt.

Mit der Restauration des Anwenderspeichers und dem Transfer des SYS-DIR an seine endgültige Adresse 4000h in Bank 0 endet dieser Teil von SYS26.

Der Patch im DOS-Kern, der SYS-Dateien aus der Bank 0 holt, ist anschließend gelistet. Der verfügbare Speicherplatz im residenten Teil von SYS0 (ab 4BE1h) reicht dazu nicht aus. Ein notwendiges Unterprogramm macht deshalb an der Stelle 3738h weiter (was auch bisher schon so war). Dieses UP kann nun per DMA auf aufwendiges Banking verzichten, das zuvor nötig war. Immerhin 13 Bytes wurden frei, die Arbeitsgeschwindigkeit stieg enorm.

Das nächste Programm, das mit Listing 4 hier vorgestellt werden soll, ist in dieser Form ein reines Anwender-File. Wegen seiner Wichtigkeit erscheint es sinnvoll, es später einer SYS-Datei einzuverleiben, vielleicht gleich SYS26. Es nimmt ein paar Patches im BASIC-Interpreter vor, die dem Z180 gerecht werden.

Seine Arbeitsgeschwindigkeit ist deutlich höher als die des Z80, weil viele Befehle weniger Taktzyklen beanspruchen. Dadurch beschleunigt sich die Delay-Routine an 0060h bedenklich. Wartezeiten, die dort z. B. für den Floppy-Controller abgetrödelt werden, sind nun zu kurz. Eine geringfügige Änderung schafft Abhilfe, so daß die Verzögerungen fast genau mit den alten Zeiten identisch werden:

In der alten Version wird mit einem JR NZ-Befehl in die Schleife zurückverzweigt, wenn der Zähler BC noch nicht abgelaufen ist. Die neue Variante durchläuft zuerst den Befehl RET Z, und zwar jedesmal. Hat der Zähler noch etwas Stoff, geht es mit bedingungslosem JR zurück in die Schleife. Der zusätzliche Return-Befehl, der bei jedem Schleifendurchlauf gelesen werden muß, kostet die erwünschten Taktzyklen zur Verlangsamung.

Dies alles trifft freilich nur bei der langsameren Taktfrequenz von 1,77 MHz zu, die über Port FAh, Bit 6 gewählt werden kann. Alle kritischen

Verzögerungsschleifen des DOS werden jedoch, soweit uns bekannt, erst nach dieser Taktumschaltung durchlaufen. Wo nicht, ist die verkürzte Laufzeit durch den High-Speed-Takt des Z180-Boards immerhin noch erträglich zu schnell.

Direkt an die Delay-Schleife schließt sich der NMI-Service an. Der TRS-80 benutzt ihn als Reset-Ersatz. Beim G3s ist der NMI unbenutzt (hardwaremäßig übrigens auch nicht zugänglich - die freien Tasten links oben und unten in der Haupttastatur können mit dem NMI-Pin der CPU verdrahtet werden). Unsere Service-Routine läßt einfach nur die Interrupts wieder zu. Damit können z. B. die Dreitastenbefehle aktiviert werden, wenn man etwa jederzeit mit dem Debugger den Speicher und die Register im Auge haben möchte.

Hauptbestandteil dieses Programms ist jedoch Gerald Schröders Error-Trap-Routine. Dorthin verzweigt der Z180, wenn ein ihm unbekannter Opcode angetroffen wird. Auch hier möchten wir wieder auf schon erschienene Erklärungen verweisen, um Wiederholungen zu vermeiden (Info Nr. 17).

Beim Einsprung am Label start wird der Beginn der Trap-Routine, nämlich ein Sprung nach 1650h, nach 0000h ff. gezapt. Vor dem eigentlichen JP findet sich ein NOP. Hier kann der User bei einem Trap-Interrupt auch etwas anderes als diesen Sprung veranlassen. Mit F7h (RST 30h) an dieser Stelle wird z. B. in den Debugger gesprungen. Besonders in der Experimentierphase, wenn der Z180 noch neu ist, wird man das gelegentlich wollen.

Bei 1650h sind die BASIC-Befehle im Klartext abgelegt. Das Trap-Programm überschreibt sie. Wer gerne in BASIC arbeitet, kann so natürlich nicht verfahren. Eine andere Adresse muß her. In einem kommenden Info wird die Lösung des Problems zu lesen sein.

In Bank 1 soll ebenfalls Geralds Trap-Programm verfügbar sein. Dort ist ab 0000h Platz dafür, ein JP erübrigt sich. Diese neue Ladeadresse erfordert die Relokation aller festen Adressen im Trap-Programm. Das beginnt mit Zeile 188 im Listing. Der Vorgang ist etwas verzwickt. Wir möchten aber darauf verzichten, ihn über die Kommentare im Listing hinaus zu erläutern. Dies kann kein Lehrbuch über Z80-Assembler sein. Alle kritischen Adressen wurden mit einem Label versehen, addr1-addr15. addr1 ist im Listing sichtbar, die übrigen nicht. Diese Labels finden sich in der Tabelle addrtab wieder.

Nur ein Detail soll beleuchtet werden, weil der Leser u. U. nicht gleich RAMDISK (s. u.) über JP 4405h abfahren möchte: Beim Sprung nach 4405h wird über den Requestcode 63h nach SYS1 verzweigt. Mit diesem Code im Akku (ACHTUNG: mit etlichen anderen nicht!) wird als erstes der Stack neu eingerichtet, weil dabei eine Rückkehr zum Aufrufer nicht stattfinden soll. Für den CALL dma und die PUSHes und POPs im DMA-Treiber genügt ein vorübergehender Stack an beliebiger Stelle. Er wird hier (Zeile 199) kurz und schnell unter die Trap-Routine gelegt.

Bei einem anderen Aussprung aus TRAP/CMD, ganz besonders mit RET, muß zuvor der Original-Stack unbedingt gerettet und nachher restauriert werden. Das geht mit dem Strickmuster, nach dem auch die Zeilen 192 und 189/190 (in dieser Reihenfolge!) funktionieren. Das hätten wir unter Dreingabe von ein paar Bytes und Mikrosekunden natürlich auch hier so machen können, aber dieser schlitzohrigere Weg war halt gar zu verlockend.

Nachdem nun die Trap-Routine auch nach Bank 1 transferiert ist, wird zuletzt die RAM-Disk des G3s aktiviert. Dieser Programmbestandteil wäre verzichtbar. Da wir aber immer mit der RAM-Disk arbeiten, kann das getrost gleich mit erledigt werden. Wenn nein: Unbedingt die Aussprungsbedingungen beachten (Stack, s. o.). Übrigens ist der ursprüngliche Name des Programms MEMDISK/CMD. Hier handelt es sich um eine geänderte Version.

SYSTEST transferiert ein CMD-Programm, das als SYS-File des DOS dienen soll, in die Bank 0, von wo die SYS-Dateien mit RST 28h aufgerufen werden.

Prinzipiell geht das mit jeder Datei, in der Regel wird es sich jedoch um neue Versionen eines echten DOS-Moduls handeln.

Es beginnt gem. Listing 5 mit der Prüfung, ob das DOS überhaupt schon in Bank 0 gepuffert ist. Das ist die Voraussetzung, weil sonst die neue Datei nicht mit RST 28h erreichbar wäre. Sollte das noch nicht geschehen sein, wird SYS26 (s. d.) mit FCh im Akku zuerst aufgerufen, um das DOS zu banken. Zum Test dient ein Lückenfüller-NOP, das bei der geänderten SYS-Laderoutine an der Stelle 4C0Bh stehen muß.

Das eigentliche Arbeitsprogramm klärt nun zunächst, was für ein CMD-File in welches SYS-Programm kopiert werden soll. Diese Angaben können jedoch schon beim Aufruf gemacht werden, denn die Befehlssyntax ist

SYSTEST<,FILESPEC<,SYSNR>>.

Wenn der zweite oder beide Befehlsparameter weggelassen werden, wird nach ihnen gefragt. Dabei wird FILESPEC in der üblichen Weise, z. B. auch mit Laufwerksnummer, eingegeben. SYSNR ist nur eine Dezimalzahl für SYSx/SYS.

In der Bank 0 befinden sich nur die Files SYS1-29. Das Programm prüft daher auch, ob mit SYSNR eine zulässige Datei benannt wurde. Außerdem darf die Testdatei höchstens so lang sein wie die Zieldatei in Bank 0. Sonst würde das dahinter liegende SYS-File überschrieben werden. In diesem Falle, wie überhaupt bei jedem denkbaren Fehler, bricht das Programm mit einer DOS-Fehlermeldung ab. Dies geschieht mit einem Sprung nach DOSERR (4409h), wo der Stack neu eingerichtet wird. Daher durfte getrost ignoriert werden, ob gerade ein PUSH oder CALL den Stack verbogen hat.

Auch hier sollen allzu weitschweifige Erklärungen des Programms vermieden werden. Die Kommentare mögen genügen. Für die ausgiebigen Aufrufe etlicher DOS-Routinen möchten wir auf Grossers "DOS-Buch" (5) verweisen, ohne das wir es übrigens wohl auch nicht geschafft hätten. Es sei nur darauf hingewiesen, daß wieder der DMA benutzt werden konnte, und daß ein komfortabler Befehl des Z180, MLT ww (3), das Umrechnen einer Dezimal- in eine Binärzahl erleichterte (Zeile 31/32).

In der kommenden Zeit liegt eine interessante Aufgabe in der weiteren Umgestaltung des Systems auf den Z180 hin. Wir würden uns freuen, wenn auch die Nachbauer des Z180-Boards gelegentlich mit Software im Club-Info vertreten wären. Vorschlag: Auch und gerade die RAM-Disk kann von der neuen CPU profitieren.

Literatur:

(1) Bernhardt, Sopp, Sonder-Info *"Dein GIIIs, die 4 unbekannten Wesen"*, Michelbach/Bilz 1986

(2) Bernhardt, *"512k-RAM-Adressierung mit flexibler MMU"*, Club-Info 18, Michelbach/Bilz 1987

(3) Bernhardt, Schröder, Sonder-Info *"HD64180"*, Gschlachtenbretzingen 1989

(4) Schröder, *"Z80, HD64180 und Illegals"*, Club-Info 17, Michelbach/Bilz 1987

(5) Grosser, *"Das DOS-Buch"*, Aachen 1985

```
               00001 ;Listing 1:      Patch in SYS0/SYS, rel. Sektor 00h:
               00002 ;SYS0/SYS        konfiguriert den Z180 für G-DOS
               00003 ;FRS 00h         (hier nicht interessierende Teile sind mit LIST OFF
               00004 ;DRS 05h         ausgespart - erkennbar an den springenden Zeilennummern)
               00035
4D10  3E00     00036 m4d10   LD       A,00h            ;A (- 00: Memory: 0 WAITs, I/O: 1 WAIT
4D12  ED       00037         DB       0edh,39h,32h     ;OUT0 (32h),A
4D15  3E83     00038         LD       A,83h            ;RFSH alle 80 Zyklen 2 Zyklen lang
4D17  ED       00039         DB       0edh,39h,36h     ;OUT0 (36h),A
4D1A  21624E   00040         LD       HL,mmutab        ;Tabelle der MMU-Werte
4D1D  013803   00041         LD       BC,endtab-mmutab*256+38h      ;3 Werte ab Port 38
4D20  ED       00042         DB       0edh,93h         ;OTIMR (Werte ausgeben)
               00208
               00209 ;Tabelle zum Einrichten der MMU des Z180
4E62  10       00210 mmutab  DB       10h              ;Common 1 ab    1.E000
4E63  10       00211         DB       10h              ;Bank-Area ab   1.4000
4E64  E4       00212         DB       0e4h             ;hierfür die logischen Adressen
               00213 endtab

00000 Fehler
```

```
               00001 ;Listing 2:      Patch in SYS0/SYS im relativen Sektor 10h:
               00002 ;SYS0/SYS        stellt einen DMA-Treiber zur Verfügung
               00003 ;FRS 10h         (hier nicht interessierende Teile sind mit LIST OFF
               00004 ;DRS 15h         ausgespart - erkennbar an den springenden Zeilennummern)
               00005
               00037 ;Test auf Rechtspfeil: DOS banken oder nicht? (früher in SYS26/SYS)
3229  3A4038   00038         LD       A,(m3840)        ;Tastatur Steuerzeichen
322C  E640     00039         AND      40h              ;Rechtspfeil gedrückt? (nein: DOS banken)
322E  3EFC     00040         LD       A,0fch           ;Requestcode für SYS26/SYS
3230  CC0244   00041         CALL     Z,m4402          ;mit RST 28h SYS26 aufrufen (DOS banken)
3233  217F32   00042 m322c   LD       HL,m3275         ;Filename 'OVL4/SYS'
               00081
               00082 ;Unterprogr. für DMA: HL = LSW Quelle, DE = LSW Ziel, BC = Bytezähler,
               00083 ;A Bit 0-3 = MSB der Quelle, A Bit 4-7 = MSB des Ziels
               00084 ;(MSB = most significant byte, A16-19, obere 4 Bits der Adresse,
               00085 ; LSW = less       "      word, A0-15, untere 16  "    "     "    )
               00086
35BB           00087         ORG      35bbh
35BB  22F035   00088 dma     LD       (srclog),HL      ;Quellbereich   log. LSW       x.HL
35BE  ED53F335 00089         LD       (dstlog),DE      ;Zielbereich    log. LSW       x.DE
35C2  ED43F635 00090         LD       (bytcnt),BC      ;Bytezähler                    0.BC
35C6  E5       00091         PUSH     HL
35C7  21F535   00092         LD       HL,dstphys       ;Zielbereich    phys. MSB      Ah.DE
35CA  77       00093         LD       (HL),A           ;MSB setzen (A16-19: Bit 4-7)
35CB  ED67     00094         RRD                       ;Nibbles rotieren (A16-19: Bit 0-3)
35CD  32F235   00095         LD       (srcphys),A      ;Quellbereich   phys. MSB      Al.HL
35D0  21F735   00096         LD       HL,tabend        ;Beginn der DMA-Wertetabelle
35D3  C5       00097         PUSH     BC
35D4  012708   00098         LD       BC,0827h         ;8 Werte ab Port 27 abwärts
35D7  F5       00099         PUSH     AF
35D8  DBFB     00100         IN       A,(0fbh)         ;Systembyte 2
35DA  F5       00101         PUSH     AF               ;retten
35DB  F640     00102         OR       40h              ;MMU des Z180 freigeben
35DD  F3       00103         DI
35DE  D3FB     00104         OUT      (0fbh),A
               00105 ;       OTDMR                     ;DMAC aufsetzen
35E0  ED       00106         DB       0edh,9bh         ;(Hex-Entsprechung des Befehls)
35E2  013102   00107         LD       BC,0231h         ;2 Werte ab Port 31 abwärts
               00108 ;       OTDMR                     ;Rest aufsetzen u. DMA abschießen
35E5  ED       00109         DB       0edh,9bh
35E7  F1       00110         POP      AF               ;altes Systembyte 2
35E8  D3FB     00111         OUT      (0fbh),A
```

```
35EA  F1        00112             POP     AF
35EB  C1        00113             POP     BC
35EC  E1        00114             POP     HL
35ED  C9        00115             RET
                00116
                00117 ;Wertetabelle für den DMA-Controller (fallende Reihenfolge für OTDMR)
35EE  40        00118             DB      40h      ;30,  DSTAT:   Kanal 0, Zustand 'scharf'
35EF  02        00119             DB      02h      ;31,  DMODE:   Speicher zu Speicher, Burst-Modus
35F0  0000      00120 srclog  DW      0000h    ;20/1, SAROL/H: Quelladresse     LSW     x.####
35F2  00        00121 srcphys DB      00h      ;22,  SAROB:                     MSB     #.xxxx
35F3  0000      00122 dstlog  DW      0000h    ;23/4, DAROL/H: Zieladresse      LSW     x.####
35F5  00        00123 dstphys DB      00h      ;25,  DAROB:                     MSB     #.xxxx
35F6  0000      00124 bytcnt  DW      0000h    ;26/7, BCROL/H: Bytezähler für die Programmlänge
35F7            00125 tabend  EQU     $-1      ;Ende der Tab., womit der Zeiger HL geladen wird

00000 Fehler


                00001 ;Listing 3:     Patch in SYS26/SYS:
                00002 ;SYS26/SYS      neue Routine zum Puffern und Laden von SYS-Files
                00003 ;               in Bank 0 des User-RAMs
                00004 ;               (hier nicht interessierende Teile sind mit LIST OFF
                00005 ;               ausgespart - erkennbar an den springenden Zeilennummern)
                00006
35BB            00012 dma     EQU     35bbh    ;Adresse des DMA-Treibers
                00331
                00332 ;UP zum Übertragen von SYS-Files in die Bank 0
                00333 ;       E:      HL = Adresse RAM-DIR-Eintrag
                00334 ;               D  = Länge des Files in Sektoren
                00335 ;               E  = Adreß-MSB des Files in der Bank 0
                00336
4F1C  E5        00337 putsys  PUSH    HL               ;39xx, DIR-Eintrag des akt. SYS-Files
4F1D  D5        00338         PUSH    DE               ;E = MSB des SYS-Files, D = Länge in Skt.
4F1E  C5        00339         PUSH    BC
4F1F  F5        00340         PUSH    AF
4F20  73        00341         LD      (HL),E           ;MSB der Zieladresse ins SYS-DIR
4F21  23        00342         INC     HL               ;nächste Stelle
4F22  72        00343         LD      (HL),D           ;Zählbyte eintragen
4F23  210042    00344         LD      HL,m4200         ;Sektorpuffer
4F26  010001    00345         LD      BC,0100h         ;Länge 1 Skt.
4F29  53        00346         LD      D,E              ;DE (- Zieladresse (MSB) des Sektors
4F2A  59        00347         LD      E,C              ;(LSB immer =00)
4F2B  3E01      00348         LD      A,01h            ;von Bank 1 nach Bank 0
4F2D  CDBB35    00349         CALL    dma              ;Block übertragen
4F30  F1        00350         POP     AF
4F31  C1        00351         POP     BC
4F32  D1        00352         POP     DE
4F33  E1        00353         POP     HL
4F34  FB        00354         EI
4F35  C9        00355         RET
                00356
                00357 ;Einsprung bei Boot: RAM-Inhalt retten, SYS-Files in der Bank 0 puffern
4F36  210052    00358 boot    LD      HL,m5200         ;Beginn des zu überschreibenden Bereichs
4F39  110008    00359         LD      DE,0800h         ;Zwischenpuffer dafür in Bank 1
4F3C  42        00360         LD      B,D              ;BC (- 0800, zu überschreibender Bereich
4F3D  4B        00361         LD      C,E
4F3E  E5        00362         PUSH    HL
4F3F  D5        00363         PUSH    DE
4F40  3E11      00364         LD      A,11h            ;von Bank 1 nach Bank 1
4F42  CDBB35    00365         CALL    dma              ;RAM-Inhalt retten
4F45  ED73914E  00366         LD      (m4f10),SP       ;alten Stack retten
4F49  CDCB4E    00367         CALL    m4f55            ;DIR-Sektoren mit SYS-Entries laden
4F4C  210039    00368         LD      HL,m3900         ;vorläufiger SYS-DIR-Puffer
4F4F  110139    00369         LD      DE,m3900+1       ;zum Ausnullen vorbereiten
```

```
4F52  7B                  LD    A,E                  ;A <- 01, Zähler für 1 Durchgang
4F53  32074F              LD    (filelen),A          ;dort ablegen
4F56  0E06                LD    C,06h                ;Zähler 6 Bytes (B = 00 seit m4f55)
4F58  75                  LD    (HL),L               ;1. Byte des SYS-DIR-Puffers ausnullen
4F59  EDB0                LDIR                       ;Rest ausnullen (hier schneller als DMA)
4F5B  1E41                LD    E,41h                ;ab Page 4100 die SYS-Files puffern
4FA0  E1                  POP   HL                   ;Adresse des gepufferten RAM-Inhalts
4FA1  D1                  POP   DE                   ;seine frühere Ladeadresse
4FA2  44                  LD    B,H                  ;BC <- 0800h, Blocklänge (C=00 seit LDIR)
4FA3  3E11                LD    A,11h                ;von Bank 1 nach Bank 1
4FA5  CDBB35              CALL  dma                  ;RAM-Inhalt ab 5200h restaurieren
4FA8  210039              LD    HL,m3900             ;Puffer für SYS-DIR
4FAB  110040              LD    DE,m4000             ;endgültige Adresse dafür in Bank 0
4FAE  0601                LD    B,01h                ;Länge 256 Bytes (C = 00)
4FB0  AF                  XOR   A                    ;A <- 00: Bank 0 -> Bank 0, 'kein Fehler'
4FB1  C3BB35              JP    dma                  ;Block übertragen und zurück ins DOS

                    ;Patch im DOS-Kern: Einsprung bei DOS-Request mit RST 28h
03D3                offs4    EQU   $-4be1h           ;Abstand zur Ladeadresse
4FB4  3E40          m4be1    LD    A,m4000/256       ;DIR des gebankten DOS (Adreß-MSB)
4FB6  323D37              LD    (addrmsb-offs3),A       ;in Laderoutine patchen
4FB9  CD3837              CALL  m3738-offs3          ;in den Sektorpuffer übertragen
4FBC  2642                LD    H,m4200/256          ;Sektorpuffer (MSB)
4FBE  3A1743              LD    A,(4317h)            ;SYS-Kenner
4FC1  07                  RLCA                       ;*2 (wegen 2 Bytes pro Eintrag)
4FC2  6F                  LD    L,A                  ;als LSB des Zeigers laden
4FC3  7E                  LD    A,(HL)               ;Adreß-MSB des SYS-Files in Bank 0
4FC4  323D37              LD    (addrmsb-offs3),A       ;in die Laderoutine patchen
4FC7  23                  INC   HL                   ;auf Längenbyte stellen
4FC8  7E                  LD    A,(HL)               ;laden
4FC9  B7                  OR    A                    ;= 00? (dann INHALT/SYS oder SYS0/SYS)
4FCA  2815                JR    Z,m4c0e              ;Fehler, falls ja (nicht vorhanden)
4FCC  213837              LD    HL,m3738-offs3       ;modifizierte Sektor-Laderoutine
4FCF  226D4C              LD    (4c6dh),HL           ;in SYS0 patchen
4FD2  CD2B4C              CALL  4c2bh                ;SYS-File laden
4FD5  221E4C              LD    (4c1eh),HL           ;Startadresse des SYS-Files dort patchen
4FD8  213644              LD    HL,4436h             ;alte Sektor-Laderoutine
4FDB  226D4C              LD    (4c6dh),HL           ;restaurieren
4FDE  00                  NOP                        ;Füller bis zur alten Fortsetzung
4FE1                m4c0e    EQU   $+2

                    ;neue Laderoutine für SYS-Files
18A7                offs3    EQU   $-3738h
4FDF  E5            m3738    PUSH  HL
4FE0  D5                  PUSH  DE
4FE1  C5                  PUSH  BC
4FE2  210000              LD    HL,0000h             ;Quellblock in Bank 0
4FE4                addrmsb  EQU   $-1               ;MSB wird jeweils gepatcht
4FE5  7C                  LD    A,H                  ;Adreß-MSB
4FE6  3C                  INC   A                    ;100h Bytes weiter
4FE7  323D37              LD    (addrmsb-offs3),A       ;MSB für nächsten Block patchen
4FEA  3E10                LD    A,10h                ;aus Bank 0 nach Bank 1
4FEC  110042              LD    DE,m4200             ;Sektorpuffer
4FEF  010001              LD    BC,0100h             ;Sektorlänge
4FF2  CDBB35              CALL  dma                  ;Block übertragen
4FF5  C1                  POP   BC
4FF6  D1                  POP   DE
4FF7  E1                  POP   HL
4FF8  AF                  XOR   A                    ;Flag 'kein Fehler'
4FF9  C9                  RET

00000 Fehler
```

```
              ;Listing 4:      Einige Änderungen für den Z180 im BASIC-Interpreter:
              ;TRAP/CMD        hauptsächlich Delay- und Illegal-Trap-Routine
              ;                (Gerald Schröders TRAP-Routine ist mit LIST OFF ausge-
              ;                spart - nachzulesen im Clubinfo Nr. 17. Nur ihr G3s-
              ;                spezifischer Anfang ist gelistet.)

0000          target1 EQU      0000h              ;Zieladresse der TRAP-Routine in Bank 1
1650          target0 EQU      1650h              ;dto. Bank 0, vorläufig im BASIC-Bereich
35BB          dma     EQU      35bbh              ;Adresse des DMA-Treibers

5200                  ORG      5200h

              ;TRAP-Einsprung für Adresse 0000
5200 00       lowmem  NOP                         ;Platz für evtl. F7 (RST 30, Debugger)
5201 C35016           JP       target0            ;Einsprung bei TRAP-Interrupt

              ;Einsprung der Warteschleife und NMI-Routine ab Adresse 0060
5204 0B       delay   DEC      BC                 ;(alter Teil der Delay-Routine
5205 78               LD       A,B                ;bleibt unverändert)
5206 B1               OR       C
5207 C8               RET      Z                  ;neu: wird immer bearbeitet (Verzögerung)
5208 18FA             JR       delay              ;jetzt Laufzeit etwa wie mit Z80

520A FB       nmi     EI                          ;NMI: Interrupts zulassen
520B C9               RET

              ;Fehlerbehandlung bzw. Illegal-Simulation nach einem TRAP-Interrupt
              ;vorläufige Version: kein anderer RST 00h außer TRAP berücksichtigt!
3BBC          offs    EQU      $-target0          ;Abstand zur Arbeitsadresse
520C F3       trap    DI                          ;Interrupts aus
520D F5               PUSH     AF                 ;Akku und Flags retten
520E DBFA             IN       A,(0fah)           ;Systembyte 1
5210 32FA16           LD       (fabuff-offs),A    ;puffern
5211          addr1   EQU      $-2
5213 E6DF             AND      0dfh               ;evtl. Schreibschutz aufheben
5215 D3FA             OUT      (0fah),A           ;Systembyte neu ausgeben

52C7 DBFA     start   IN       A,(0fah)           ;Systembyte 1
52C9 E6DF             AND      0dfh               ;evtl. Schreibschutz aufheben
52CB F3               DI
52CC D3FA             OUT      (0fah),A           ;neu ausgeben
52CE 2A0052           LD       HL,(lowmem)        ;Code für Adresse 0000
52D1 220000           LD       (0000h),HL         ;am Zielort ablegen
52D4 2A0252           LD       HL,(lowmem+2)      ;Fortsetzung (insgesamt 4 Bytes)
52D7 220200           LD       (0002h),HL
52DA 210452           LD       HL,delay           ;neue Verzögerungsroutine
52DD 116000           LD       DE,0060h           ;Zieladresse
52E0 010800           LD       BC,trap-delay      ;Länge
52E3 EDB0             LDIR                        ;(wenige Bytes: LDIR schneller als DMA)
52E5 115016           LD       DE,target0         ;Zieladresse (HL bereits auf Quelle trap)
52E8 01BB00           LD       BC,start-trap      ;Länge
52EB 3E01             LD       A,01h              ;von Bank 1 nach Bank 0
52ED CDBB35           CALL     dma                ;Block übertragen (schneller als LDIR)
52F0 060F             LD       B,endtab-addrtab/2     ;Anzahl zu relozierender Adressen
52F2 311E53   addrlop LD       SP,addrtab         ;Stack auf Adressentabelle setzen
52F3          spbuff  EQU      $-2
52F5 E1               POP      HL                 ;eine Adresse abholen
52F6 ED73F352         LD       (spbuff),SP        ;Tabelle als Stack retten
52FA F9               LD       SP,HL              ;dortige Adresse = Stack
52FB E1               POP      HL                 ;dortiges Word abholen
52FC ED52             SBC      HL,DE              ;auf neue Basisadresse umrechnen
52FE E5               PUSH     HL                 ;neues Word ablegen
52FF 10F1             DJNZ     addrlop            ;bis alle Adressen bearbeitet sind
5301 210C52           LD       HL,trap            ;Adresse der TRAP-Routine
```

```
5304  F9                      LD      SP,HL               ;unterhalb neuer Stack
5305  110000                  LD      DE,target1          ;Zieladresse in Bank 1
5308  01BB00                  LD      BC,start-trap       ;Länge der TRAP-Routine
530B  3E11                    LD      A,11h               ;von Bank 1 nach Bank 1
530D  CDBB35                  CALL    dma                 ;TRAP-Routine nach Bank 1 verladen
5310  211653                  LD      HL,ramdisk          ;Aufruf der RAM-Disk (kann entfallen)
5313  C30544                  JP      4405h               ;als DOS-Befehl ausführen und ENDE

5316  52              ramdisk DM      'RAMDISK',0dh       ;Name des RAM-Disk-Programms, CR

                      ;Tabelle der zu relozierenden Adressen in der TRAP-Routine
531E  1152            addrtab DW      addr1,addr2,addr3,addr4,addr5,addr6,addr7,addr8
532E  7352                    DW      addr9,addr10,addr11,addr12,addr13,addr14,addr15
                      endtab

00000 Fehler


                      ;Listing 5:     Ein geändertes SYS-File soll vor dem endgültigen Gebrauch
                      ;SYSTEST/CMD    in Bank 0 gepuffert und von dort aus getestet werden.

6000                          ORG     6000h

35BB                  dma     EQU     35bbh               ;DMA-Treiber

6000  3A0B4C          start   LD      A,(4c0bh)           ;bei gepuffertem DOS steht dort 00
6003  B7                      OR      A                   ;ist das DOS in Bank 0 schon gepuffert?
6004  2807                    JR      Z,go_baby           ;alles klar, falls ja
6006  E5                      PUSH    HL                  ;Befehlszeiger retten
6007  3EFC                    LD      A,0fch              ;Requestcode für SYS26: DOS erst bunkern
6009  CD0244                  CALL    4402h               ;dort steht RST 28h für den DOS-Request
600C  E1                      POP     HL                  ;Befehlszeiger
600D  CDD54C          go_baby CALL    4cd5h               ;HL auf nächsten Befehlsparameter stellen
6010  2006                    JR      NZ,makefcb          ;falls etwas folgt
6012  11AF60                  LD      DE,filespc          ;Text "Name ..."
6015  CD9560                  CALL    getinpt             ;ausgeben, Dateinamen einlesen
6018  11E060          makefcb LD      DE,fcb              ;FCB-Adresse
601B  CD1C44                  CALL    441ch               ;Dateinamen in den FCB übertragen
601E  CD2044                  CALL    4420h               ;FCB eröffnen
6021  206D                    JR      NZ,errexit          ;falls Fehler aufgetreten
6023  CDD54C                  CALL    4cd5h               ;HL auf nächsten Befehlsparameter stellen
6026  2006                    JR      NZ,getsys           ;falls etwas folgt
6028  11C960                  LD      DE,sysnmbr          ;Text "SYS-File ..."
602B  CD9560                  CALL    getinpt             ;ausgeben, SYS-Nr. einlesen
602E  CD9F60          getsys  CALL    getciph             ;1. Ziffer binär umwandeln
6031  78                      LD      A,B                 ;Binärziffer
6032  280B                    JR      Z,putsys            ;falls Eingabe beendet (SYS1-9)
6034  0E0A                    LD      C,0ah               ;Faktor 10d für zweistellige Zahlen
                      ;       MLT     BC                  ;BC = B * C: B (- 00, C (- 10*B
6036  ED                      DB      0edh,4ch            ;Hex-Entsprechung des Befehls
6038  CD9F60                  CALL    getciph             ;2. Ziffer einlesen (muß letzte sein)
603B  2053                    JR      NZ,errexit          ;falls Eingabe jetzt noch nicht beendet
603D  78                      LD      A,B                 ;neue Ziffer (binär)
603E  81                      ADD     A,C                 ;+ 10mal 1. Ziffer
603F  3C              putsys  INC     A                   ;+2, weil GDOS und INHALT keine Nr. haben
6040  3C                      INC     A
6041  FE03                    CP      03h                 ;SYS1? ("kleinste" erlaubte SYS-Datei)
6043  3849                    JR      C,illnmbr           ;Fehler: Eingabe von SYS0
6045  FE20                    CP      1fh+1               ;SYS29? ("höchste" erlaubte SYS-Datei)
6047  3045                    JR      NC,illnmbr          ;falls Quatsch wie etwa SYS4711
6049  07                      RLCA                        ;*2, weil 2 Bytes pro Eintrag im SYS-DIR
604A  08                      EX      AF,AF'              ;retten
604B  210040                  LD      HL,4000h            ;dort in Bank 0 das SYS-DIR
604E  110080                  LD      DE,8000h            ;Puffer dafür in Bank 1
```

```
6051  D5                      PUSH  DE               ;für später
6052  45                      LD    B,L              ;BC (- 0040h = 64d, Länge des SYS-DIR
6053  4C                      LD    C,H
6054  3E10                    LD    A,10h            ;von Bank 0 nach Bank 1
6056  CDBB35                  CALL  dma              ;SYS-DIR in den Puffer übertragen
6059  08                      EX    AF,AF'           ;aufbereitete SYS-Nr.
605A  5F                      LD    E,A              ;DE (- Adresse des Eintrags im SYS-DIR
605B  1A                      LD    A,(DE)           ;MSB der Adresse des SYS-Files in Bank 0
605C  328560                  LD    (sysaddr),A      ;dort ablegen
605F  13                      INC   DE               ;auf Längenbyte (Blockzähler) stellen
6060  1A                      LD    A,(DE)           ;laden
6061  5F                      LD    E,A              ;DE (- Länge des SYS-Files (LSB)
6062  55                      LD    D,L              ;MSB = 00 (max. 35 Sektoren bei SYS6)
6063  2AEC60                  LD    HL,(fcb+0ch)     ;2. u. 1. EOF-Byte, gibt Länge in Skt. an
6066  7D                      LD    A,L              ;Zähler für zu ladende Skt. (LSB genügt)
6067  3C                      INC   A                ;+1, weil ab 0 gezählt, Cy lösch. wg. SBC
6068  47                      LD    B,A              ;B als Zähler aufsetzen
6069  328860                  LD    (bytecnt),A      ;als DMA-Bytezähler (MSB) patchen
606C  ED52                    SBC   HL,DE            ;beide Filelängen vergleichen
606E  3E3B                    LD    A,3bh            ;Fehlercode 'unzureich. Speicherplatz'
6070  301E                    JR    NC,errexit       ;falls Testfile zu lang
6072  E1                      POP   HL               ;Puffer ab 8000h in Bank 1
6073  11E060                  LD    DE,fcb           ;Zeiger auf den FCB
6076  22E360        ldloop    LD    (fcb+03h),HL     ;Pufferadresse in den FCB schreiben
6079  24                      INC   H                ;Pufferzeiger 256 Bytes weiterstellen
607A  CD3644                  CALL  4436h            ;einen Sektor einlesen
607D  2011                    JR    NZ,errexit       ;falls ein Fehler auftrat
607F  10F5                    DJNZ  ldloop           ;bis alle Sektoren eingelesen sind
6081  2680                    LD    H,8000h/256      ;HL (- Puffer 8000h in Bank 1 (LSB = 00)
6083  110000                  LD    DE,0000h         ;Zieladresse in Bank 0 (LSB immer 00)
6085                sysaddr   EQU   $-1              ;MSB der SYS-Adresse in Bank 0
6086  010000                  LD    BC,0000h         ;Bytezähler (LSB immer 00)
6088                bytecnt   EQU   $-1              ;MSB wird gepatcht
6089  3E01                    LD    A,01h            ;von Bank 1 nach Bank 0
608B  C3BB35                  JP    dma              ;neues SYS-File speichern und ENDE

608E  3E29          illnmbr   LD    A,29h            ;Fehlercode 'unzul. log. Dateinummer'
6090  E67F          errexit   AND   7fh              ;NZ-Bedingung (Fehler), keine Rückkehr
6092  C30944                  JP    4409h            ;Fehlermeldung ausgeben und ENDE

6095  EB            getinpt   EX    DE,HL            ;HL (- Zeiger auf Text
6096  CD6744                  CALL  4467h            ;Text anzeigen
6099  0620                    LD    B,20h            ;max. 32d Zeichen (sehr reichlich)
609B  EB                      EX    DE,HL            ;HL (- DOS-Eingabepuffer für Eingabe
609C  C34000                  JP    0040h            ;Eingabe einlesen und zurück

609F  7E            getciph   LD    A,(HL)           ;Zeichen laden
60A0  FE30                    CP    '0'              ;kleinste erlaubte Ziffer?
60A2  38EA                    JR    C,illnmbr        ;Fehler, falls kleiner
60A4  FE3A                    CP    '9'+1            ;größte erlaubte Ziffer?
60A6  30E6                    JR    NC,illnmbr       ;Fehler, falls größer
60A8  E60F                    AND   0fh              :ASCII -) binär
60AA  47                      LD    B,A              ;Eingabe binär puffern
60AB  23                      INC   HL               ;Befehlszeiger nachstellen
60AC  C3D54C                  JP    4cd5h            ;nächsten Befehlsparam. holen und zurück

                    ;verschiedene Texte und Puffer
60AF  4E            filespc   DM    'Name des Testprogramms? ',0eh,03h        ;Text, Curs., EOS
60C9  53            sysnmbr   DM    'SYS-File-Nr.',09h,'(1-29)? ',0eh,03h     ;dto. mit TAB
                    fcb                                                       ;Puffer für FCB

6000                          END   start

00000 Fehler
```

Anhang

Probleme mit den Illegals

Geralds Trap-Programm ist eine wertvolle Hilfe, dennoch kein Allheilmittel. Von den vielen denkbaren Illegals, die im Code-Muster des Z80 stecken, sind längst nicht alle bereits entschlüsselt und dokumentiert. Die Trap-Routine müßte daher bei jeder neuen Entdeckung aktualisiert werden.

Und der Z180 benutzt selber für seine neuen Befehle solche ehemals weißen Stellen in der Code-Matrix des Z80. Glücklicherweise beschieden sich seine Designer mit solchen Codes, die mit dem Switch-Byte EDh beginnen, so daß ein notwendiges Debugging verdächtiger Programme in Grenzen bleibt. Nämlich auch Illegals mit EDh werden schon irgendetwas beim Z80 bewirken, was auch immer. Die Illegal-Trap-Routine würde in diesem Falle überhaupt nicht angesprungen, denn es wäre für den Z180 ja ein sinnvoller Befehl.

Einer dieser beiden Gründe wird es wohl sein, weshalb nach der Umrüstung des G3s auf den Z180 ausgerechnet mit ZEUS/CMD zahlreiche Abstürze auftraten. Aber vielleicht wurde auch Geralds Programm falsch abgetippt. Egal - es mußte etwas geschehen, denn mit ZEUS steht und fällt die Soft-Seite des ganzen Unternehmens:

DSMBLR ist ein Disassembler, der gottlob keine Illegals akzeptiert, wie es z. B. ZEUS durchaus tut. Wenn man daher ein Programm disassembliert, das Illegals enthält, zuckt DSMBLR ratlos die Achseln und setzt DM-(DEFM-)Statements, wo er in seinem Vorrat keinen Z80-Befehl findet (DSMBLR und ZEUS behandeln DB und DM gleich, soweit es sich um einzelne Bytes handelt).

So haben wir ZEUS ohne jegliches Screening disassembliert und nach DMs gesucht. An vier Stellen wurden wir fündig (Listing 6). In einem gesonderten File, mit dem wir seit jeher ZEUS patchen (ZAPZEUS/CMD, Listing 7), wurden Simulationen der Illegals programmiert, in ZEUS selbst CALLs nach dort gepatcht.

ZAPZEUS (für diesen Zweck genügen die gelisteten Teile) kann so abgetippt und auf Papier assembliert werden. Die Bestandteile bis Zeile 46 werden, etwa mit Hilfe von FED/CMD, mitten in ZEUS gezapt. Was dahinter liegt, muß im letzten Sektor von ZEUS angehängt werden. Der Ende-String (02-02-yy-xx) wird als neuer Record-Header (01-nn-yy-xx) geändert, das Anhängsel-Programm bringt seinen eigenen Ende-String mit. Dafür ist es wichtig, daß seine Einsprungsadresse (END addr) die von ZEUS sein muß (s. Listing).

Ob damit alles erledigt ist, muß die zukünftige Praxis zeigen. Ohne ein Screening-File für DSMBLR werden nämlich Texte und Tabellen wie Programmcode disassembliert. So ergeben sich Verschiebungen im "Programm"-Ablauf, die etwa ein Illegal-Switch-Byte wie einen harmlosen Bestandteil eines Sprungbefehls (Adresse, Distanz) erscheinen lassen können. Da das aber nur auftreten kann, wenn ein Illegal direkt an einen reinen Datenblock grenzt, darf man mit Recht auf Erfolg hoffen.

Auch das Gegenteil wäre eine mögliche Fehlerquelle: Eine Tabelle, die ohne Screening ja nicht als solche behandelt wird, enthält ein verdächtiges Byte. Der Zufall will es, daß der Inhalt der Tabelle wie sinnvoller Programmcode aussieht. Ein dort gepatchter CALL würde die Tabelle verfälschen. Wir können nur raten, in solchen Fällen die betreffenden Stellen ganz besonders sorgfältig zu untersuchen ...

... und unbedingt das Ergebnis zur Veröffentlichung an Jens-80 Neueder zu schicken!

```
                00001 ;Listing 6:        Illegals in ZEUS/CMD
                00002 ;ZEUS/CMD          (hier nicht interessierende Teile sind mit LIST OFF
                00003 ;                  ausgespart - erkennbar an den springenden Zeilennummern)
                00004
52E7  DD        00305          DM       0ddh            ;DD 7D:
52E8  7D        00306          LD       A,L             ;LD A,LX
52E9  FE2B      00307          CP       '+'
                00308
5B09  CB        00694          DM       0cbh            ;CB 31 06 01:
5B0A  310601    00695          LD       SP,m0106        ;SLIA C - LD B,01
                00696
5B45  CB        00731          DM       0cbh            ;CB 31 06 01:
5B46  310601    00732          LD       SP,m0106        ;SLIA C - LD B,01
                00733
6AE9  CB        00700          DM       0cbh            ;CB 37:
6AEA  37        00701          SCF                      ;SLIA A
                00702

00000 Fehler
```

```
                00001 ;Listing 7:        Illegal-Trap direkt in ZEUS/CMD
                00002 ;ZAPZEUS/CMD       (hier nicht interessierende Teile sind mit LIST OFF
                00003 ;                  ausgespart - erkennbar an den springenden Zeilennummern)
                00034
52E7            00035          ORG      52e7h           ;dort Illegal LD A,LX
52E7  CD2830    00036          CALL     ld_a_lx         ;simulieren
                00037
5B09            00038          ORG      5b09h           ;dort Illegal SLIA C
5B09  CD3330    00039          CALL     slia_c          ;simulieren
                00040
5B45            00041          ORG      5b45h           ;dort Illegal SLIA C
5B45  CD3330    00042          CALL     slia_c          ;simulieren
                00043
6AE9            00044          ORG      6ae9h           ;dort Illegal SLIA A
6AE9  37        00045          SCF                      ;Cy für Bit 0 setzen
6AEA  17        00046          RLA                      ;rotieren
                00076
3000            00077          ORG      3000h           ;wo es nicht stört
                00106
3028  DDE5      00107 ld_a_lx  PUSH     IX              ;auf den Stack
302A  E3        00108          EX       (SP),HL         ;HL retten, HL <- IX
302B  7D        00109          LD       A,L             ;A <- LX
302C  E3        00110          EX       (SP),HL         ;HL und IX restaurieren
302D  DDE1      00111          POP      IX
302F  FE2B      00112          CP       2bh             ;überschriebenen Befehl nachholen
3031  1805      00113          JR       return          ;zurück zur Fortsetzung
                00114
3033  CB21      00115 slia_c   SLA      C               ;C shiften
3035  0C        00116          INC      C               ;Bit 0 setzen
3036  0601      00117          LD       B,01h           ;überschriebenen Befehl nachholen
                00118
3038  E3        00119 return   EX       (SP),HL         ;HL <- RET-Adresse
3039  23        00120          INC      HL              ;ein Byte später
303A  E3        00121          EX       (SP),HL         ;zurücktauschen
303B  C9        00122          RET                      ;an die neue Adresse zurückkehren
                00123
5312            00124          END      5312h           ;hier Einsprung in ZEUS

00000 Fehler
```

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
BILDSCHIRM	BILDSCHIRMAUSGABE mit INVERSEN ZEICHEN	?	12	79
ECB-BUS	ECB-ADAPTER-PLATINE	?	15	65
INTERFACE	Relais-Interface	?	9	47
PROGRAMMIERG	STRUKTURIERTE PROGRAMMIERUNG	?	7	17
CODE	BASICODE-INTERFACE	?(aus ELEKTOR)	3	?
CODE	BASICODE—2	?(aus Elektor)	3	?
ECB-BUS	VERBINDEND	ASSENBAUM	11	39
CHIP	NANOHENRY BLOCKIEREN DEN FORTSCHRITT	aus c' t	19	70
TASTATUR	UNIVERSALGENIE	aus c’t	19	57
COMPUTER	SHARP 3541	aus Comp.Pers.	26	47
DFUe	Bewaehrtes DFUe-Protokoll	aus Comp.Pers.	27	51
HRG	Traumlandschaften aus dem Computer	aus Comp.Pers.	27	41
HRG	Wie funktioniert ein Vektorgenerator	aus Comp.Pers.	27	31
SCHNITTSTELL	Die Parallelschnittstelle	aus Comp.Pers.	27	49
CHIP	DER LEIDENSWEG DER CMOS-CHIPS	aus c't	19	67
SCHALTALGEBR	KLEINES LOGIK-LEXIKON	aus c't	19	41
PASCAL	Echte Progranne neiden PASCAL	aus DATAMON7/83	23	91
DBASE	dBASE II, III und III+ Ueberblick	aus DOS 5'88	26	67
SORTIEREN	TRS80: Sortieren	aus mc	23	12
TEXTSYSTEM	KLEINES TEXTPROGRAMM	aus mc	23	10
DISK	Sicherer Platz fuer Daten? (Disketten)	aus test 9/87	22	73
CHIP	ERSTER MEGABIT-CHIP IN SERIE	aus Zeitschrift	18	18
UTILITY	PAGELIST/CMD	aus Zeitschrift	1B	21
DRUCKZEICHEN	Zeichengenerator fuer Nadeldrücker	aut nc	23	18
HARDCOPY	TRS80 erstellt HARDCOPY	aut nc	23	17
GRAFIK	Auslesen des Grafikspreichers	BERNH./SEELMANN	20	65
BANKING	256K-RAM fuer Z80-SYSTEME	BERNHARDT	14	91
BILDSCHIRM	Bessere Darst.v.Text u.HRG auf d.Bildsch	BERNHARDT	26	37
BUS	PROF 180x und mc68000-ECB	BERNHARDT	25	57
DECODER	Eleganter Adressdecoder fuer GENIE I	BERNHARDT	28	73
DRUCKER	STAR NL 10 abwechselnd am CP/M-R. + PC	BERNHARDT	28	72
ECB-BUS	Anm. zum CLUB80-ECB-Bus-Projekt	BERNHARDT	23	73
ECB-BUS	Gepufferter DMA- u. IM2-faehiger ECB	BERNHARDT	23	74
ECB-BUS	Prof 180x und mc-68OOO-ECB	BERNHARDT	25	57
ERROR	"PRUEFZAHLFEHLER BEIM LESEN"	BERNHARDT	22	71
GRAFIK	Es lebe der Standard!	BERNHARDT	28	78
LAUFWERK	EXP1Floppy-Controller	BERNHARDT	26	38
PASCAL	APFELMAENNCHEN auf der GDP64	BERNHARDT	21	37
RAM	512K-RAM-ADRESSEN m. FLEXIBLER MMU	BERNHARDT	18	67
SPEICHER	Sicheres, Ein -, Plaetzchen	BERNHARDT	7	43
TRS-80	TRS-80 Japan-Modell aufbohren	BERNHARDT	25	60
VIDEO	Das 80-Zeichen-Karte-Problem u.d.Loesung	BERNHARDT	22	63
COMPUTER	Der fluesternde Portable	BETZ	28	67
SCHALTSTATUS	WACHHUND, Der-, b.unkontr.Schaltzustand	BETZ	20	63
Z80	ARBEITSTIER MIT GESUNDEM HERZEN	BETZ	21	52
BILDSCHIRM	BIN ICH STRAHLENGEFAEHRDET?	BOIKAT, Ute	20	109
CP/M	Billiges CP/M	BREWER	9	53
UTILITY	EINGABEROUTINE MIT KOMFORT	BULLING	7	13

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	VERFASSER [1]	Info	Seite
TAPE	SUPERTAPE FUER TRS 80	BURGWITZ/STILLR	7	44
UMLAUTE	Umlaute auf der Tastatur des GENIE I	DRECHSLER	28	69
UTILITY	Startprogramm f. BASIC-Programme	DRECHSLER	28	21
FREQUENZ	Taktfrequenz-Verdopplung GENIE I/II	DREYER	5	?
BUS	ECB-BUS-Projekt	DROWAELD. &OBERM	15	65
ECB-BUS	DIE EINGANGSPLATINE	DROWAELDER	15	73
LAUFWERK	Floppydisk-Controllerkarte	DROWAELDER	18	49
TASTATUR	Tastatur- und Drucker-Platine	DROWAELDER	20	71
DFUe	Datenuebertragung per Fernsehbild	Edde & Heribert	12	62
DFUe	DATENUEBERTRAGUNG p.FERNSEHBILD/WDR-CLUB	EDDE u.HERIBERT	12	62
TELEFON	TELEFONBUCH MIT PFIFF	FRISCH	2	?
GRUNDLAGEN	Digitaltechnik, Grundlagen der -	gesammelt: KUHN		
ZEICHENSATZ	Geheimnisse eines Zeichengenerators	GIESE	28	27
PROGRAMMIERG	KEINE MEINUNG ZUR STRUKTUR?	GRELL	7	28
MATHEMATIK	WIE der NAT.LOG.die RECHENZEIT VERKUERZT	GUAGLIANO	14	77
POKEPEEK	POKE+PEEK-ECKE	HANS -?-	4	?
AMATEURFUNK	Die Betriebsart PACKET-Radio (PR)	HARTMANN, H.-G.	28	111
Z81	422 NEUE Z80-BEFEHLE	HASSELBERG (MC)	4	?
ASSEMBLER	KLEINSCHRIFT fuer den EDTASM+	HEIDENREICH	20	37
BILDFORMAT	FUTTER fuer GENIE IIIs CRTC	HEIDENREICH	20	22
DOS	R-BEFEHL MSDOS-LIKE (ein Editor f. DOS)	HEIDENREICH	20	44
DOS	Moeglichkeiten und Grenzen des MS-DOS	HEIDENREICH	27	19
DRUCKER	Noch ein Druckerpatch	HELD	23	71
PASCAL	GENIE, CP/M und 80 ZEICHEN und HRG ???	HELD	21	43
UTILITY	SHELL/CMD	HERERMANN	24	34
GRAFIK	ERGAENZUNG ZU GRAFIK-SHORTY	HERMANN	19	32
KOPIERSCHUTZ	6.3 - KOPIERSCHUTZ	HERMANN	21	55
M4/4p	AUFRUF DER SUPERVISOR-CALLS in BASIC	HERMANN	14	62
MODEL 4	BOOT-ROM-VERSION und DATUM M4	HERMANN	19	42
PATCH	GROSZBUCHSTABENMODUS AB SYSTEMSTART (M4)	HERMANN	21	30
PROGRAMMIERG	EDIT-BEFEHLE	HERMANN	14	4
UTILITY	TRACE-ON	HOEH (COMP.PERS	4	?
GRAFIK	TRS80 Ml/3 SCHNELLE GRAFIK IN BASIC	HORNUNG	2	?
RECHT	URHEBERRECHTSSCHUTZ fuer PROGRAMME	KAHLEN	10	55
MATHEMATIK	SPLINE-INTERPOLATION (KUBISCH)	KASPER	16	19
MATHEMATIK	PRIMZAHLBERECHNUNGEN (3 PROGRAMME)	KASPER	5	?
MONITOR	MONITOR IM GENIE	KASPER	6	23
TRS-81	TRS-80, Der - und die Aussenwelt	KLEIN, WILFRIED	4	?
DRUCKZEICHEN	AM BILDSCHIRM DEFINIERT/EDITOR f.DRUCKER	KLUGE	9	19
TEXTSYSTEM	UMDEFINIERUNG v.SCRIPSIT-KOMMANDOTASTEN	KONRAD	8	14
ASSEMBLER	SCHUETZEN VON ASSEMBLERPROGRAMMEN	KONRAD J.	5	?
TASMON	ERWEITERUNG FUER TASMON (NEWDOS-BEFEHLE)	KONRAD J.	4	25
UTILITY	AUSGABEUMSCHALTUNG IN BASIC	KONRAD J.	6	24
ASSEMBLER	HALBAUTOMATISCHE FESTLEGUNG VON ORG	KONRAD, J.	9	3
PASCAL	PASCAL-PROBLEME	KONRAD, J.	9	73
PROGRAMMIERG	BASIC- LAUFENDE PROGRAMME ERHEITERN	KRENZKE	16	18
INTERFACE	Interface-Adapter 8255, Programmierbarer	KUHN	13	47
PORT	PORTBELEGUNG von COMPUTERKONFIGURATIONEN	KUHN	21	44

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
PORT	Portbelegung	KUHN	21	44
SPURANZAHL	80/4O-TRACKUMSCHALTUNG	KUHN	19	49
UTILITY	ALLERLEI NUETZLICHES FUER DEN TRS 80	LICHTERMANN /MC	4	?
DRUCKER	SPEED WRITING (fuer EPSON MX-DRUCKER)	LINDSEY	12	65
DRUCKER	Speed Writing (erhoeht Druckgeschwind.)	LINDSEY	12	65
BERICHT	Testbericht KX 1083	MAND	23	89
AUFRUESTEN	Modernisierung statt Neukauf	MATTHAEI/H.COMP	5	?
ASCII	Wogegen selbst ZEUS vergeblich kaempft	MUEHLENBEIN	25	5
ASSEMBLER	MEIN ERSTES ASSEMBLERPROGRAMM	MUEHLENBEIN	14	9
ASSEMBLER	MEIN DRITTES ASSEMBLERPROGRAMM	MUEHLENBEIN	16	33
BANKING	Bangen um Banken	MUEHLENBEIN	25	7
COMPILER	BASCOM-KURZANLEITUNG	MUEHLENBEIN	13	38
DRUCKER	...WEIL KUERZE SO WUERZIG IST...	MUEHLENBEIN	20	35
GENAUIGKEIT	MATHEMATISCHER UEBERFLUSS	MUEHLENBEIN	20	17
LOGIK	STENO-LOGIK	MUEHLENBEIN	16	24
LOGIK	BOOLE XOR ABEL	MUEHLENBEIN	17	16
MATH.STATIST	BIVARANA (2fache Varianzanalyse)	MUEHLENBEIN	19	21
MATHEMATIK	GOLDENER FIBONACCI	MUEHLENBEIN	16	21
MATHEMATIK	UMWANDLUNG! ROEMISCHE IN ARABISCHE ZAHL	MUEHLENBEIN	16	7
MATHEMATIK	DAS CARTESISCHE BLATT	MUEHLENBEIN	18	5
MATHEMATIK	CRAMER UND DIE DETERMINANTEN	MUEHLENBEIN	19	13
MATHEMATIK	DER SCHNELLSTE ZAHNARZT (WURZELZIEHEN)	MUEHLENBEIN	22	11
MATHEMATIK	Polynonberechnung	MUEHLENBEIN	24	13
MATHEMATIK	Ableitg.d.Negationsregel f.Binaerzahlen	MUEHLENBEIN	25	9
MATHEMATIK	Ein letzter 10Zeiler (Determinantenber.)	MUEHLENBEIN	28	10
MATHEMATIK	PASCAL St GALTON	MUEHLENBEIN	28	17
MATHEMATIK	Sitzverteilung nach d'Hondt	MUEHLENBEIN	2B	11
MATHE-SPIEL	WAS IST MIT DER PRIMZAHL '37' LOS?	MUEHLENBEIN	11	23
MUSIK	MEIN ZWEITES ASSEMBLERPROGRAMM	MUEHLENBEIN	15	15
MUSIK	SCHWERE KAVALLERIE ("DAS CLUBLIED")	MUEHLENBEIN	22	17
PASCAL	Vergleich BASIC nit TURBO-PASCAL	MUEHLENBEIN	28	39
ROM-ROUTINEN	CALLERLEI	MUEHLENBEIN	20	27
SEMANTIK	TRAU - SCHAU, WEM! Ein Bitchen Englisch	MUEHLENBEIN	18	47
SORTIEREN	IN VISICALC SORTIEREN	MUEHLENBEIN	20	93
SORTIEREN	Ordnung musz sein !	MUEHLENBEIN	28	13
SPRACHE	DER-DIE-DAS MACHT MIR SPASZ	MUEHLENBEIN	19	75
TEXTSYSTEM	KEINE ANGST VOR... (TRENNEN IN TSCRIPS)	MUEHLENBEIN	20	61
TEXTSYSTEM	Was TSCRIPS nicht kann	MUEHLENBEIN	23	20
TEXTSYSTEM	TSCRIPS retten	MUEHLENBEIN	2B	3B
TONERZEUGUNG	DA CAPO... (WIE MAN TOENE ERZEUGT)	MUEHLENBEIN	22	13
UTILITY	FILES WIE SAND AM MEER...	MUEHLENBEIN	11	7
UTILITY	DIRECTORY LESEN UND KOMMENTARE SPEICHERN	MUEHLENBEIN	13	15
VISICALC	Korrigieren und Steuern innerhalb VC	MUEHLENBEIN	23	24
VISICALC	SORTICALC	MUEHLENBEIN	26	11
BILDSCHIRM	DAS PROJEKT (FERNSEHER –> MONITOR)	MUELLER, K.	18	61
COMPILER	RPNL - DIE ANDERE ART ZU PROGRAMMIEREN	MUELLER, KURT	15	57
COMPILER	RPNL-BESCHREIBUNG	MUELLER, KURT	16	37
COMPILER	RPNL	Mueller, Kurt	26	31

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
FREQUENZ	Taktumschaltung	MUELLER, Kurt	16	45
HRG	HRG-1b-Karte	MUELLER, KURT	14	89
SPIEL	Lottozahlen	MUELLER, Kurt	24	19
TAKT	TAKTUMSCHALTUNG - DIE ALLERLETZTE LOESG.	MUELLER, KURT	14	45
TEXTSYSTEM	DUMP (TSCRIPS retten)	MUELLER, Kurt	26	18
TASTATUR	UMLAUTE IM NEWDOS? ABER NATUERLICH!	NEUEDEER	10	41
ECB-BUS	ECB-BUS-TRS80-PROJEKT	NEUEDER	13	39
ECB-BUS	ECB-BUS-PROJEKT	NEUEDER	14	101
ECB-BUS	ECB-BUS-GRUNDEINHEITSAUFBAU	NEUEDER	15	71
HRG	HRG vom BASIC aus (DEMO)	NEUEDER	10	31
HRG	HRG-PIXELS aus dem BASIC!	NEUEDER	10	23
HRG	HRG: 384x192 oder 192x192	NEUEDER	13	14
JOYSTICK	Joysticks fuer den TRS-80 M1	NEUEDER	5	?
UTILITY	FILEVERGLEICH (s.auch Progr.-Bibliothek)	NEUEDER	10	43
BASIC	EXTENDED BASIC(RESTOREn u. LPRINT alles)	OBERMANN	11	33
BUS	ECB-BUS-Projekt	OBERMANN	9	42
COMPILER	CALL mit BASCOM	OBERMANN	12	33
COMPUTER	UNTERSCHIEDE (zw. MODEL 1 und 3 bzw. 4)	OBERMANN	14	55
COMPUTER	MODEL 4p ROM-IMAGE-LOADER	OBERMANN	15	43
COMPUTER	TANDY<—> SCHNEIDER	OBERMANN	17	49
ECB-BUS	ECB-BUS fuer MODEL 3/4!???	OBERMANN	15	84
FLOPPY	FLOPPY-TRICKS	OBERMANN	13	41
FREQUENZ	Taktumschaltung fuer VIDEO-GENIE	OBERMANN	8	25
GRAFIK	HRG-BILDUEBERTRAGUNG OHNE HRG (WDR-FILES	OBERMANN	15	11
HRG	HRG-BILDUEBERTRAGUNG per DFUe	OBERMANN	14	24
HRG	UMKEHR!	OBERMANN	20	58
HRG	HRG-Bilduebertragung	OBERMANN	25	31
LAUFWERK	Floppy-Tricks	OBERMANN	14	87
LAUFWERK	80/40-Trackumschaltung Mitsubishi	OBERMANN	26	39
LOGIK	LOGISCHE OPERATIONEN IN BASIC	OBERMANN	15	9
LOGIK	LOGISCHE OPERATIONEN 2	OBERMANN	16	9
M4/4p	DAS BOOT-ROM des 4p	OBERMANN	14	63
M4/4p	HRG fuer MODEL 3/4/4p	OBERMANN	14	67
M4/4p	Atari-Bilder auf den 4p	OBERMANN	26	21
MAUS	Die Maus am TRS-80 !	OBERMANN	22	60
MEMDISK	RESET-FESTE MEMDISK M4-CP/M	OBERMANN	19	43
MODEL 5	TURBO TIP	OBERMANN	21	31
PASCAL	TURBO PATCH	OBERMANN	21	32
PASCAL	TURBO PATCH (FEHLERKORREKTUR)	OBERMANN	21	32
PASCAL	TURBO TIP	OBERMANN	21	31
PASCAL	TURBO-TIP	OBERMANN	21	31
PASCAL	Vergleichstest (versch.PASCAL-Versionen)	OBERMANN	23	33
PASSWORT	PASSWORT-BYPASS BEI TRSDOS 6.2	OBERMANN	19	42
PDRIVE	PDRIVE-IDENTIFIKATION fuer NEWDOS 80	OBERMANN	10	17
RAM	EXTRA RAM ("Ein sicheres Plaetzchen...")	OBERMANN	9	63
RESTORE	BASIC-PATCH FUER RESTORE N	OBERMANN	5	?
SENSOR	Digitales Temperatur-Zeit-Uhrenmodul	Obermann	16	41
SPIEL	JOYSTICK AN MODEL 4/4p	OBERMANN	19	50

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	VERFASSER [1]	Info	Seite
SPIEL	TAPE-GAMES AUF MODEL III/4/4p	OBERMANN	20	57
TASTATUR	KNOEPFE	OBERMANN	14	57
TEXTSYSTEM	SCRIPSIT AUF DEM ALTEN GENIE	OBERMANN	14	43
UHR	UHRUNTEN fuer MODEL 4p	OBERMANN	15	47
UTILITY	DDE St LWT	OBERMANN	10	19
UTILITY	LPRINT ALLES einfacher	OBERMANN	8	3
UTILITY	ZAP IM SUPERZAP/Zappen im Zeichenbereich	OBERMANN	8	13
ZAPPEN	EIN BYTE UND SEINE FOLGEN	OBERMANN	10	21
FORMATIERUNG	40 TRACKS? WARUM NICHT 42?	OBSCHERNINGKAT	18	11
M4/4p	Das neueste DOS fuer M4/4p	OBSCHERNINGKAT	24	29
VERTRETUNGEN	TRACE od.DIE GESCHICHTE EINER ZEITSCHR.	OBSCHERNINGKAT	18	75
PROGRAMMIERG	LOKALE VARIABLE BEIM TRS 80	PAULO	7	7
TONERZEUGUNG	TONGENERATOR FUER TRS 80	PLUMHOFF	B	I9
ADAPTER	Testadapter	REGGE	9	45
FREQUENZ	Synchronisierte Taktumschaltung	RETZLAFF	10	49
LAUFWERK	LWT fuer GENIE I und TRS-80 MODELL I	RETZLAFF	13	40
TRS-82	Umbau eines TRS-80 Mod.I 16K auf 48K	RETZLAFF	10	45
GRAFIK	SCHNELL BEWEGTE GRAFIK AUF DEM TRS80	ROECKRATH	2	?
TRS 80	DER BEKNACKTE TRS 80	ROECKRATH	6	33
HARDCOPY	HARDCOPY AUCH OHNE INTERRUPT	RYCHLIK	20	15
PROGRAMMIERG	PLANVOLL PROGRAMMIEREN	SACHSE	7	23
SPIEL	ADVEN-80	SCARGILL	11	9
BILDSCHIRM	NOCH EIN BILDSCHIRMSCHONER	SCHMID, A.	22	26
BILDSCHIRM	SCHWIMMENDE BILDER	SCHMID, A.	22	66
LAUFWERK	Ein klebriges Problem	SCHMID, A.	28	71
RAMDISK	Die RAMDISK und du 11.Gebot	SCHMID, A.	25	54
RAMDISK	Die RAMDISK und das elfte Gebot	SCHMID, A.	25	56
RAMDISK	Darf's ein bisserl mehr sein?	SCHMID, A.	26	46
SPRACHE	Welche Sprache fuer wen ?	SCHMID, A.	27	59
TASTATUR	ES GEHT NOCH SCHNELLER (TAST.ABSCHALTEN)	SCHMID, A.	22	32
TEXTSYSTEM	Wie sag' ich's meinem WORDSTAR ?	SCHMID, A.	24	11
UHR	Es war einmal ein Datum	SCHMID, A.	26	29
ZEITMESSUNG	ETWAS fuer LEUTE, die NIE ZEIT haben	SCHMID, A.	22	31
DRUCKER	DIE DRUCKERABFRAGE	SCHMITZ	18	12
ERFAHRUNG	PC STATT GENIE? (Ueber weitere PC's)	SCHMITZ	20	99
HAENDLER	Reingefallen alt Rheintec	SCHMITZ	23	90
TEXTSYSTEM	WORDSTAR auf allen Laufwerken	SCHOBER	28	35
ASSEMBLER	"INPUT"	SCHROEDER	11	5
ASSEMBLER	EINIGE ASSEMBLER-ROUTINEN	SCHROEDER	14	17
ASSEMBLER	ASSEMBLER CONTRA LISP	SCHROEDER	17	19
ASSEMBLER	DIE WERKZEUGE DES ASSEMBL.PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	WERKZEUGE DES ASSEMBLER-PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	LMOFFSET with DISABLE DOS	SCHROEDER	9	9
BANKING	SYSCOPY1 und RAMDISK f.HELMUT's BANKER	SCHROEDER	16	32
BERICHT	GENIE IIs - Erfahrungsbericht	SCHROEDER	9	67
BILDSCHIRM	DIE 80-ZEICHEN-KARTE	SCHROEDER	12	63
BILDSCHIRM	80-ZEICHENKARTE X80, SEDIT und TSCRIPS	SCHROEDER	13	17
BILDSCHIRM	EIN NEUER 80ZEICHEN-TREIBER	SCHROEDER	19	77

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
GRAFIK	GRAFIK-STANDARD: ALLEIN GEHT'S AUCH	SCHROEDER	19	11
LITERATUR	BUECHERLISTE: DIE ERSTEN KURZKRITIKEN	SCHROEDER	21	67
MATHE/GRAFIK	RECHNEN (ZEICHNEN) MIT DEM COMPUTER...	SCHROEDER	10	53
PASCAL	KURZKRITIKEN und BUECHERLISTE	SCHROEDER	21	71
PLOTTER	KREIS OHNE TRIGONOMETRIE	SCHROEDER	15	8
PROZESSOR	Z80, HD64180 UND ILLEGALS	SCHROEDER	17	27
SPIEL	DAS PIRATE ADVENTURE	SCHROEDER	11	15
TEXTSYSTEM	AENDERUNGEN IN TSCRIPS und NAME/CMD	SCHROEDER	13	16
TEXTSYSTEM	FREIRAEUME IN TSCRIPS	SCHROEDER	14	47
TEXTSYSTEM	SCRIPSIT-TABELLEN	SCHROEDER	14	45
TEXTSYSTEM	TEXTUMLEITUNG fuer TSCRIPS	SCHROEDER	14	49
TEXTSYSTEM	TSCRIPS-VERSCHLUESSELUNG	SCHROEDER	15	29
GRAFIK	EINF.GRAFIKPROGRAMM: PLOT/BAS	SCHROEDER /ARPS	14	29
ASSEMBLER	Assembler unter CP/M	SCHROEDER, G.	25	49
BUS	Freiheit fuer den BUS	SCHROEDER, G.	25	55
SCHNITTSTELL	Entfernung der Kassettenschnittstelle	SCHROEDER, G.	26	40
SCHNITTSTELL	Memory mapped I/O <==> Port I/O bei GIIs	SCHROEDER, G.	26	42
VIDEO	BO-Zeichen-Karte	SCHROEDER, G.	12	63
Z80	INTERFACE mit Kassetten-Schnittstelle	SCHROEDER, G.	26	40
GRAFIK	LINIENMODELLE IN 3 DIMENSIONEN	SCHROEDER/ ARPS	14	35
MATHEMATIK	Fraktale Geometrie	SCHROEER	27	15
BANKING	Nochmaliges Banken	SOERENSEN	28	47
INTERRUPT	Unterbrechung gestattet ?	SOERENSEN	28	52
PASCAL	PASCAL-VORSTELLUNG TEIL 2	SOERENSEN	22	40
PASCAL	PASCAL fuer INTERESSIERTE	SOERENSEN, R.	20	12
ASCII	ASCII-TABELLE IMMER ZUR HAND	SOPP	19	19
ASSEMBLER	ZEUS MIT 64 UND 80 ZEICHEN	SOPP	18	36
ASSEMBLER	Was tun bei zerschossener Source ?	SOPP	24	25
ASSEMBLER	Bedingt assemblieren mit ZEUS	SOPP	25	24
ASSEMBLER	Bedingte Assembly - notfalle ohne IF	SOPP	25	28
ASSEMBLER	WER HAT ANGST VOR ASSEMBLER?	SOPP	5	?
ASSEMBLER	LPRINT ALLES einfacher! (Teil 3)	SOPP	9	11
BASIC	EIN MYSTERIOESER BASIC-BEFEHLs IsA	SOPP	5	?
BILDSCHIRM	MEHR FERNSEHEN fuer's GELD (f.GENIE IIIs	SOPP	14	21
BILDSCHIRM	FUTTER fuer den CRTC	SOPP	19	40
BILDSCHIRM	NOCH MEHR FERNSEHN fuers GELD	SOPP	19	39
BILDSCHIRM	EIN SCHONER fuer die BILDROEHRE	SOPP	20	20
BILDSCHIRM	EIN WEITERER BILDSCHIRMSCHONER	SOPP	22	28
BILDSCHIRM	VIDHEX - HEXANZEIGE d.BILDSCHIRMS m.HRB	SOPP	9	13
DRUCKEN	LPRINT MIT MODERNEN GENIES	SOPP	18	19
DRUCKER	VON 8 AUF 24 PINS (= DRUCKNADELN)	SOPP	22	20
DRUCKERGRAF.	GRAPHIK-JKL, TSCRIPS UND DER NEC Pi	SOPP	16	27
ECHTZEITUHR	ACH DU LIEBE ZEIT!	SOPP	15	32
ECHTZEITUHR	REAL TIME BLACK BOX?	SOPP	15	53
FILEVERGLCH	VERGLEICH ZWEIER DISK-DATEIEN	SOPP	22	53
FILEVERGLCH	VERGLEICH ZWEIER PROGRAMME! noch'n TOOL	SOPP	22	57
FILEVERGLCH	Vergleich zweier Disk-Dateien	SOPP	25	20
GRAFIK	GRAFIK-CLS fuer das GENIE IIIs	SOPP	13	35

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
GRAFIK	STRINGANZEIGE MIT VOLLDAMPF	SOPP	6	41
GRAFIK-JKL	GRAFIK-JKL fuer das GENIE IIIs	SOPP	13	32
HARDCOPY	FLOPCOPY AUCH OHNE INTERRUPT	SOPP	21	14
HRG	HRG 1b PROGRAMMIEREN, BESSER ERKLAERT	SOPP	10	25
HRG	HRG UND BASIC NETTO	SOPP	10	27
HRG	HRG-SPEICHER LOESCHEN, DEN -	SOPP	10	29
HRG	GENIE IIIs, DIE HRG DES -	SOPP	12	21
HRG	HRG-HARDCOPY fuer das G IIIs	SOPP	13	25
HRG	HRG, dasz die Schwarte kracht !	SOPP	24	40
JKL	JKL - ABER DRUCKERSCHONEND!	SOPP	12	13
JOYSTICK	Joystick-Anschluss	SOPP	6	29
LAUFWERK	Ordnung muss sein !	SOPP	28	33
LIBRARY	WAS TUN BEI VOLLER LIBRARY?	SOPP	12	15
LOGIK	LOGISCHES ZU LOGIK	SOPP	16	14
MATHEMATIK	Bin.-Umwandlg.HEX/DEZ;kuerzere E-Routine	SOPP	24	23
MATHEMATIK	Binaeruwaandlung, letzter (?) Teil	SOPP	25	12
MAUS	Maus mit sieben Beinen	SOPP	23	69
MAUS	Die Pratze laeszt das Mausen nicht	SOPP	25	18
PORT	OUT,PORT#,xx,yy,...	SOPP	11	29
PORT	VERBESSERT: IO.PORT#<,xx,yy,.. >	SOPP	12	11
RAMDISK	MINI-RAM-FLOPPY im "SICHEREN PLAETZCHEN"	SOPP	10	37
RECORDORGAN.	ZU GERALDs ARTIKEL ueber LMOFFSET	SOPP	10	3
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES	SOPP	13	23
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES (NACHTRAG)	SOPP	14	53
SPEICHER	CALL UM DIE ECKE (MEM-BEREICHE ERREICHEN	SOPP	15	49
SYSTEM	Quatsch in OVL4/SYS	SOPP	24	27
SYSTEM	Schon Mieder SYS14/SYS	SOPP	24	20
TEXT-GRAFIK	(TELEFON) - EIN UD-ZEICHEN auf d.DRUCKER	SOPP	16	29
TEXTSYSTEM	TIPS fuer SCRIPS	SOPP	13	19
TEXTSYSTEM	NEUES ZUR PROPORTIONALSCHRIFT	SOPP	18	15
TEXTSYSTEM	§ CONTRA .	SOPP	19	27
TEXTSYSTEM	TSCRIPS & DIE &-CODES	SOPP	19	16
TEXTSYSTEM	VON DER WIEGE BIS ZUR BAHRE (ADR.AUTOMAT	SOPP	19	33
TEXTSYSTEM	KEINE ANGST VOR PROPORTIONALSCHRIFT	SOPP	18	13
TEXTSYSTEM	TSCRIPS UND DIE 24 NADELN	SOPP	22	23
UHR	WENN DIE UHR MAL STOERT	SOPP	14	39
UHR	THE TIMES, THEY ARE A-CHANGING	SOPP	22	35
UTILITY	LPRINT ALLES	SOPP	6	31
ZEICHENSATZ	NEUE ZEICHENSAETZE fuer das GENIE IIIs	SOPP	12	28
ASSEMBLER	SCHOENER LISTEN MIT ZEUS	SOPP	18	41
COMPUTER	DEIN GIIIs... (NACHTRAG)	SOPP/BERNHARDT	17	54
TEXTSYSTEM	DAS TSCRIPS-DRAMA - EIN LEHRSTUECK	SOPP/MUHLENBEIN	21	56
BUS	Indikatorkarte fuer ECB-BUS	SPIESS	19	45
BUS	Analog / Digital - Umsetzer	STOBER	24	37
DRUCKER	Funktionsweise des Drucker-Puffers	TRAPP	6	18
MODEM	Akustikkoppler	TRAPP	4	?
SPRACHE	COBOL - EINE EINFUEHRUNG	TRAPP	5	?
UTILITY	BILDSCHIRMAUSDRUCK BEIM TRS80+VIDEOGENIE	UEBERSCHAAR		

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
LITERATUR	WEITERE KURZKRITIKEN	VERSCH.	22	DIV
SPIEL	ADVENTURE GEHEIMAGENT XP-05	WAGNER, ALEX.	3	?
KASSETTE	CASSETTEVERZEICHNIS ERSTELLEN	WAGNER, G.	10	33
M4/4p	AUFGESTIEGEN?	WAGNER, G.	14	59
PROGRAMMIERG	DIRECTORY LESEN	WAGNER, G.	9	4
PROGRAMMIERG	EDITOR	WAGNER, G.	9	5
PROMMER	EPROM-Programmiergeraet	WOLF (aus mc)	5	?
MATHEMATIK	Was man weiss - schon inner wissen wollt	WOLF, Thomas	26	15
ASSEMBLER	BASIC RUFT ASSEMBLER	WOLLSCHLAEGER	21	11
ASSEMBLER	ZBO-Prograamtricks in Assembler	WOLLSCHLAEGER	27	39
COMPUTER	Modell 4	WOLLSCHLAEGER	12	69
MODEL 6	KLEINE ZIPPERLEIN RASCH KURIERT	WOLLSCHLAEGER	12	57
PROGRAMMIERG	TRICKS MIT STRINGS	WOLLSCHLAEGER	4	45
PROGRAMMIERG	PROGRAMMIEREN HEISST m. DATEIEN ARBEITEN	WOLLSCHLAEGER	7	29
SICHERHEIT	RAM-DISK KOSTENLOS	WOLLSCHLAEGER	2	?
CHIP	BEI SIEMENS HEISST ES: CHIP CHIP HURRA!	ZEITSCHRIFT	18	76
HAENDLER	TANDY SCHLIESST COMPUTER-CENTER	ZEITSCHRIFT	18	75
IC	Hardware-Tips: Neue IC's	ZEITSCHRIFT	17	38
LOGIK	LOGIKSIMULATION mit BASIC	ZEITSCHRIFT	16	11
PROGRAMMIERG	DEM INTERPRETER auf» BIT GESCHAUT	ZEITSCHRIFT	12	37
PROGRAMMIERG	DIE RICHTIGE DIMENSION	Zeitschrift	18	24
PROZESSOR	TAV's Z80-TUNING	ZEITSCHRIFT	17	33
UTILITY	SMARTSCREEN, SMARTKEY und SMARTPRINT	ZEITSCHRIFT	12	45
ASSEMBLER	TIP FUER SCHNELLERES ARBEITEN MIT EDTASM	ZWICKEL	6	67
COMPUTER	WIE PHOENIX AUS DER ASCHE (4 COMP.in 1)		17	9
COMPUTER	CP/MAC-eine BRUECKE zw.G-/NEWDOS u.CP/M		18	44
COMPUTER	32 BIT zum NIEDRIGPREIS (TANDY 4000)		22	79
CONTROLLER	FLOPPYCONTROLLER-KOMPATIBILITAET		19	47
CP/M	CP/MAC - NA UND?		19	78
CP/M	WAS HABT IHR GEGEN CP/M ?		21	53
CP/M	GENIE, CP/M UND 80 ZEICHEN UND HRG???		21	43
CP/M	CMD ==> COM ohne CPMac		21	23
CP/M	Wie baue ich mir ein CP/M ?		21	25
CP/M	CP/M intern - kurz und kompakt		23	49
CP/M	Grafik unter CP/M		25	35
CP/M	Z-CP/M - eine deutliche Verbesserung		25	37
CP/M	ZCPR33 - Installation auf M4...		25	45
CP/M	BDOS-Replacement - was'n du?		26	27
CP/M	CP/M-Warmstart ohne Nachladen		26	25
CP/M	PROMMER 80 unter G-DOS		28	58
CP/M	Fensterin mit CP/M		28	64
CP/M	Backen und Banken		28	44
CP/M	ZCPR33 - die Vierte		28	53
CPU	Z80-Hochintegrations-CPUs		23	88
DATEI	INDEX-DATEIEN		19	15
DATEN	VIDEODAT 300		13	51
DATEN	DATENVERLUSTE		19	80
DATENBANK	dBASE II - eine Einfuehrung		21	33

STICHWORT [2]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [1]	Info	Seite
DATENBANK	BDBS - BASIC DATENBANK SYSTEM (MODEL 4)		22	37
DATENBANK	dBASE II		23	65
DATENBANK	Datenbanken richtig aufgebaut		26	73
DATENBANK	Dateiprogramm		28	22
DATENBANK	Teamwork		28	37
DDE	DDE fuer das GENIE IIIs		12	17
DEF.FUNCTION	BASIC-PROGRAMMIERBARE FUNKTIONEN		19	26
DENKAUFGABEN	MATHEMATIK fuer FEINSCHMECKER		20	101
DFUe	Datenuebertragung / M4p als Terminal		23	30
DISASSEMBLER	Kleine Hexen bevorzugt		23	27
DISASSEMBLER	Kleine Hexen ohne Aufwand		25	17
DISASSEMBLER	Kurzanleitung fuer SCREENER/CMD		25	15
DISASSEMBLER	Screening mit DSMBLR/CMD		25	14
DOS	EXTENDED NEWDOS 80		10	5
DOS	NEWDOS/80+10 VERSION 2.4 (26.10.85)		11	25
DOS	DOS - (FAST) OHNE FLOPPY		14	69
DOS	DIR/SYS ODER INHALT/SYS?		15	21
DOS	FILTER UNTER TRSDOS 6.x		18	29
DOS	SUPERDOS		2	?
DOS	MDOS im INFO???????		20	11
DOS	LS-DOS 6.3 - KOPIERSCHUTZ		21	55
DOS	Der Teufe) steckt oft im Detail		25	30
DOS	Ein kleiner Fehler im Z80-DOS		27	14
DOS	Microcomputer-Betriebssysteme		28	103
DOS-FEHLER	FEHLER IM NEWDOS-BEFEHL 'BOOT1		11	21
DOS-Fehler	Ein Fehler in SYS14/SYS		24	19
DOUBLER	SELBSTBAU-DOUBLER fuer EXP1		17	45
DRUCKEN	DREISPALTIG DRUCKEN		21	15
DRUCKEN	EIN FEHLER IM SYS28/SYS		22	33
DRUCKEN	Druckausgabe f. COBOL-Programme		23	32
DRUCKEN	Druckausgabe mit PRBUF		24	12
HAENDLER	TANDY-VERTAGSHAENDLER		19	79
RECHT	COMPUTERKRIMINALITAET		19	81
SPRACHE	COBOL-EINFUEHRUNG		8	5
VIDEO	VIDEODAT-SYSTEM		1B	57
VERFASSER [1]	**Der hochgestellte Index gibt die Reihenfolge der Sortierung an**			
STICHWORT [2]				

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	VERFASSER [2]	Info	Seite
ADAPTER	Testadapter	REGGE	9	45
AMATEURFUNK	Die Betriebsart PACKET-Radio (PR)	HARTMANN, H.-G.	28	111
ASCII	Wogegen selbst ZEUS vergeblich kaempft	MUEHLENBEIN	25	5
ASCII	ASCII-TABELLE IMMER ZUR HAND	SOPP	19	19
ASSEMBLER	KLEINSCHRIFT fuer den EDTASM+	HEIDENREICH	20	37
ASSEMBLER	SCHUETZEN VON ASSEMBLERPROGRAMMEN	KONRAD J.	5	?
ASSEMBLER	HALBAUTOMATISCHE FESTLEGUNG VON ORG	KONRAD, J.	9	3
ASSEMBLER	MEIN ERSTES ASSEMBLERPROGRAMM	MUEHLENBEIN	14	9
ASSEMBLER	MEIN DRITTES ASSEMBLERPROGRAMM	MUEHLENBEIN	16	33
ASSEMBLER	"INPUT"	SCHROEDER	11	5
ASSEMBLER	EINIGE ASSEMBLER-ROUTINEN	SCHROEDER	14	17
ASSEMBLER	ASSEMBLER CONTRA LISP	SCHROEDER	17	19
ASSEMBLER	DIE WERKZEUGE DES ASSEMBL.PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	WERKZEUGE DES ASSEMBLER-PROGRAMMIERERS	SCHROEDER	21	9
ASSEMBLER	LMOFFSET with DISABLE DOS	SCHROEDER	9	9
ASSEMBLER	Assembler unter CP/M	SCHROEDER, G.	25	49
ASSEMBLER	ZEUS MIT 64 UND 80 ZEICHEN	SOPP	18	36
ASSEMBLER	Was tun bei zerschossener Source ?	SOPP	24	25
ASSEMBLER	Bedingt assemblieren mit ZEUS	SOPP	25	24
ASSEMBLER	Bedingte Assembly - notfalle ohne IF	SOPP	25	28
ASSEMBLER	WER HAT ANGST VOR ASSEMBLER?	SOPP	5	?
ASSEMBLER	LPRINT ALLES einfacher! (Teil 3)	SOPP	9	11
ASSEMBLER	SCHOENER LISTEN MIT ZEUS	SOPP	18	41
ASSEMBLER	BASIC RUFT ASSEMBLER	WOLLSCHLAEGER	21	11
ASSEMBLER	ZBO-Prograamtricks in Assembler	WOLLSCHLAEGER	27	39
ASSEMBLER	TIP FUER SCHNELLERES ARBEITEN MIT EDTASM	ZWICKEL	6	67
AUFRUESTEN	Modernisierung statt Neukauf	MATTHAEI/H.COMP	5	?
BANKING	256K-RAM fuer Z80-SYSTEME	BERNHARDT	14	91
BANKING	Bangen um Banken	MUEHLENBEIN	25	7
BANKING	SYSCOPY1 und RAMDISK f.HELMUT's BANKER	SCHROEDER	16	32
BANKING	Nochmaliges Banken	SOERENSEN	28	47
BASIC	EXTENDED BASIC(RESTOREn u. LPRINT alles)	OBERMANN	11	33
BASIC	EIN MYSTERIOESER BASIC-BEFEHLs IsA	SOPP	5	?
BERICHT	Testbericht KX 1083	MAND	23	89
BERICHT	GENIE IIs - Erfahrungsbericht	SCHROEDER	9	67
BILDFORMAT	FUTTER fuer GENIE IIIs CRTC	HEIDENREICH	20	22
BILDSCHIRM	BILDSCHIRMAUSGABE mit INVERSEN ZEICHEN	?	12	79
BILDSCHIRM	Bessere Darst.v.Text u.HRG auf d.Bildsch	BERNHARDT	26	37
BILDSCHIRM	BIN ICH STRAHLENGEFAEHRDET?	BOIKAT, Ute	20	109
BILDSCHIRM	DAS PROJEKT (FERNSEHER –> MONITOR)	MUELLER, K.	18	61
BILDSCHIRM	NOCH EIN BILDSCHIRMSCHONER	SCHMID, A.	22	26
BILDSCHIRM	SCHWIMMENDE BILDER	SCHMID, A.	22	66
BILDSCHIRM	DIE 80-ZEICHEN-KARTE	SCHROEDER	12	63
BILDSCHIRM	80-ZEICHENKARTE X80, SEDIT und TSCRIPS	SCHROEDER	13	17
BILDSCHIRM	EIN NEUER 80ZEICHEN-TREIBER	SCHROEDER	19	77
BILDSCHIRM	MEHR FERNSEHEN fuer's GELD (f.GENIE IIIs	SOPP	14	21
BILDSCHIRM	FUTTER fuer den CRTC	SOPP	19	40
BILDSCHIRM	NOCH MEHR FERNSEHN fuers GELD	SOPP	19	39

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
BILDSCHIRM	EIN SCHONER fuer die BILDROEHRE	SOPP	20	20
BILDSCHIRM	EIN WEITERER BILDSCHIRMSCHONER	SOPP	22	28
BILDSCHIRM	VIDHEX - HEXANZEIGE d.BILDSCHIRMS m.HRB	SOPP	9	13
BUS	PROF 180x und mc68000-ECB	BERNHARDT	25	57
BUS	ECB-BUS-Projekt	DROWAELD. &OBERM	15	65
BUS	ECB-BUS-Projekt	OBERMANN	9	42
BUS	Freiheit fuer den BUS	SCHROEDER, G.	25	55
BUS	Indikatorkarte fuer ECB-BUS	SPIESS	19	45
BUS	Analog / Digital - Umsetzer	STOBER	24	37
CHIP	NANOHENRY BLOCKIEREN DEN FORTSCHRITT	aus c' t	19	70
CHIP	DER LEIDENSWEG DER CMOS-CHIPS	aus c't	19	67
CHIP	ERSTER MEGABIT-CHIP IN SERIE	aus Zeitschrift	18	18
CHIP	BEI SIEMENS HEISST ES: CHIP CHIP HURRA!	ZEITSCHRIFT	18	76
CODE	BASICODE-INTERFACE	?(aus ELEKTOR)	3	?
CODE	BASICODE–2	?(aus Elektor)	3	?
COMPILER	BASCOM-KURZANLEITUNG	MUEHLENBEIN	13	38
COMPILER	RPNL - DIE ANDERE ART ZU PROGRAMMIEREN	MUELLER, KURT	15	57
COMPILER	RPNL-BESCHREIBUNG	MUELLER, KURT	16	37
COMPILER	RPNL	Mueller, Kurt	26	31
COMPILER	CALL mit BASCOM	OBERMANN	12	33
COMPUTER	SHARP 3541	aus Comp.Pers.	26	47
COMPUTER	Der fluesternde Portable	BETZ	28	67
COMPUTER	UNTERSCHIEDE (zw. MODEL 1 und 3 bzw. 4)	OBERMANN	14	55
COMPUTER	MODEL 4p ROM-IMAGE-LOADER	OBERMANN	15	43
COMPUTER	TANDY<–> SCHNEIDER	OBERMANN	17	49
COMPUTER	DEIN GIIIs... (NACHTRAG)	SOPP/BERNHARDT	17	54
COMPUTER	Modell 4	WOLLSCHLAEGER	12	69
COMPUTER	WIE PHOENIX AUS DER ASCHE (4 COMP.in 1)		17	9
COMPUTER	CP/MAC-eine BRUECKE zw.G-/NEWDOS u.CP/M		18	44
COMPUTER	32 BIT zum NIEDRIGPREIS (TANDY 4000)		22	79
CONTROLLER	FLOPPYCONTROLLER-KOMPATIBILITAET		19	47
CP/M	Billiges CP/M	BREWER	9	53
CP/M	CP/MAC - NA UND?		19	78
CP/M	WAS HABT IHR GEGEN CP/M ?		21	53
CP/M	GENIE, CP/M UND 80 ZEICHEN UND HRG???		21	43
CP/M	CMD ==> COM ohne CPMac		21	23
CP/M	Wie baue ich mir ein CP/M ?		21	25
CP/M	CP/M intern - kurz und kompakt		23	49
CP/M	Grafik unter CP/M		25	35
CP/M	Z-CP/M - eine deutliche Verbesserung		25	37
CP/M	ZCPR33 - Installation auf M4...		25	45
CP/M	BDOS-Replacement - was'n du?		26	27
CP/M	CP/M-Warmstart ohne Nachladen		26	25
CP/M	PROMMER 80 unter G-DOS		28	58
CP/M	Fensterin mit CP/M		28	64
CP/M	Backen und Banken		28	44
CP/M	ZCPR33 - die Vierte		28	53
CPU	Z80-Hochintegrations-CPUs		23	88

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
DATEI	INDEX-DATEIEN		19	15
DATEN	VIDEODAT 300		13	51
DATEN	DATENVERLUSTE		19	80
DATENBANK	dBASE II - eine Einfuehrung		21	33
DATENBANK	BDBS - BASIC DATENBANK SYSTEM (MODEL 4)		22	37
DATENBANK	dBASE II		23	65
DATENBANK	Datenbanken richtig aufgebaut		26	73
DATENBANK	Dateiprogramm		28	22
DATENBANK	Teamwork		28	37
DBASE	dBASE II, III und III+ Ueberblick	aus DOS 5'88	26	67
DDE	DDE fuer das GENIE IIIs		12	17
DECODER	Eleganter Adressdecoder fuer GENIE I	BERNHARDT	28	73
DEF.FUNCTION	BASIC-PROGRAMMIERBARE FUNKTIONEN		19	26
DENKAUFGABEN	MATHEMATIK fuer FEINSCHMECKER		20	101
DFUe	Bewaehrtes DFUe-Protokoll	aus Comp.Pers.	27	51
DFUe	Datenuebertragung per Fernsehbild	Edde & Heribert	12	62
DFUe	DATENUEBERTRAGUNG p.FERNSEHBILD/WDR-CLUB	EDDE u.HERIBERT	12	62
DFUe	Datenuebertragung / M4p als Terminal		23	30
DISASSEMBLER	Kleine Hexen bevorzugt		23	27
DISASSEMBLER	Kleine Hexen ohne Aufwand		25	17
DISASSEMBLER	Kurzanleitung fuer SCREENER/CMD		25	15
DISASSEMBLER	Screening mit DSMBLR/CMD		25	14
DISK	Sicherer Platz fuer Daten? (Disketten)	aus test 9/87	22	73
DOS	R-BEFEHL MSDOS-LIKE (ein Editor f. DOS)	HEIDENREICH	20	44
DOS	Moeglichkeiten und Grenzen des MS-DOS	HEIDENREICH	27	19
DOS	EXTENDED NEWDOS 80		10	5
DOS	NEWDOS/80+10 VERSION 2.4 (26.10.85)		11	25
DOS	DOS - (FAST) OHNE FLOPPY		14	69
DOS	DIR/SYS ODER INHALT/SYS?		15	21
DOS	FILTER UNTER TRSDOS 6.x		18	29
DOS	SUPERDOS		2	?
DOS	MDOS im INFO???????		20	11
DOS	LS-DOS 6.3 - KOPIERSCHUTZ		21	55
DOS	Der Teufe) steckt oft im Detail		25	30
DOS	Ein kleiner Fehler im Z80-DOS		27	14
DOS	Microcomputer-Betriebssysteme		28	103
DOS-FEHLER	FEHLER IM NEWDOS-BEFEHL 'BOOT1		11	21
DOS-Fehler	Ein Fehler in SYS14/SYS		24	19
DOUBLER	SELBSTBAU-DOUBLER fuer EXP1		17	45
DRUCKEN	LPRINT MIT MODERNEN GENIES	SOPP	18	19
DRUCKEN	DREISPALTIG DRUCKEN		21	15
DRUCKEN	EIN FEHLER IM SYS28/SYS		22	33
DRUCKEN	Druckausgabe f. COBOL-Programme		23	32
DRUCKEN	Druckausgabe mit PRBUF		24	12
DRUCKER	STAR NL 10 abwechselnd am CP/M-R. + PC	BERNHARDT	28	72
DRUCKER	Noch ein Druckerpatch	HELD	23	71
DRUCKER	SPEED WRITING (fuer EPSON MX-DRUCKER)	LINDSEY	12	65
DRUCKER	Speed Writing (erhoeht Druckgeschwind.)	LINDSEY	12	65

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
DRUCKER	...WEIL KUERZE SO WUERZIG IST...	MUEHLENBEIN	20	35
DRUCKER	DIE DRUCKERABFRAGE	SCHMITZ	18	12
DRUCKER	VON 8 AUF 24 PINS (= DRUCKNADELN)	SOPP	22	20
DRUCKER	Funktionsweise des Drucker-Puffers	TRAPP	6	18
DRUCKERGRAF.	GRAPHIK-JKL, TSCRIPS UND DER NEC Pi	SOPP	16	27
DRUCKZEICHEN	Zeichengenerator fuer Nadeldrücker	aut nc	23	18
DRUCKZEICHEN	AM BILDSCHIRM DEFINIERT/EDITOR f.DRUCKER	KLUGE	9	19
ECB-BUS	ECB-ADAPTER-PLATINE	?	15	65
ECB-BUS	VERBINDEND	ASSENBAUM	11	39
ECB-BUS	Anm. zum CLUB80-ECB-Bus-Projekt	BERNHARDT	23	73
ECB-BUS	Gepufferter DMA- u. IM2-faehiger ECB	BERNHARDT	23	74
ECB-BUS	Prof 180x und mc-68OOO-ECB	BERNHARDT	25	57
ECB-BUS	DIE EINGANGSPLATINE	DROWAELDER	15	73
ECB-BUS	ECB-BUS-TRS80-PROJEKT	NEUEDER	13	39
ECB-BUS	ECB-BUS-PROJEKT	NEUEDER	14	101
ECB-BUS	ECB-BUS-GRUNDEINHEITSAUFBAU	NEUEDER	15	71
ECB-BUS	ECB-BUS fuer MODEL 3/4!???	OBERMANN	15	84
ECHTZEITUHR	ACH DU LIEBE ZEIT!	SOPP	15	32
ECHTZEITUHR	REAL TIME BLACK BOX?	SOPP	15	53
ERFAHRUNG	PC STATT GENIE? (Ueber weitere PC's)	SCHMITZ	20	99
ERROR	"PRUEFZAHLFEHLER BEIM LESEN"	BERNHARDT	22	71
FILEVERGLCH	VERGLEICH ZWEIER DISK-DATEIEN	SOPP	22	53
FILEVERGLCH	VERGLEICH ZWEIER PROGRAMME! noch'n TOOL	SOPP	22	57
FILEVERGLCH	Vergleich zweier Disk-Dateien	SOPP	25	20
FLOPPY	FLOPPY-TRICKS	OBERMANN	13	41
FORMATIERUNG	40 TRACKS? WARUM NICHT 42?	OBSCHERNINGKAT	18	11
FREQUENZ	Taktfrequenz-Verdopplung GENIE I/II	DREYER	5	?
FREQUENZ	Taktumschaltung	MUELLER, Kurt	16	45
FREQUENZ	Taktumschaltung fuer VIDEO-GENIE	OBERMANN	8	25
FREQUENZ	Synchronisierte Taktumschaltung	RETZLAFF	10	49
GENAUIGKEIT	MATHEMATISCHER UEBERFLUSS	MUEHLENBEIN	20	17
GRAFIK	Auslesen des Grafikspreichers	BERNH./SEELMANN	20	65
GRAFIK	Es lebe der Standard!	BERNHARDT	28	78
GRAFIK	ERGAENZUNG ZU GRAFIK-SHORTY	HERMANN	19	32
GRAFIK	TRS80 Ml/3 SCHNELLE GRAFIK IN BASIC	HORNUNG	2	?
GRAFIK	HRG-BILDUEBERTRAGUNG OHNE HRG (WDR-FILES	OBERMANN	15	11
GRAFIK	SCHNELL BEWEGTE GRAFIK AUF DEM TRS80	ROECKRATH	2	?
GRAFIK	GRAFIK-STANDARD: ALLEIN GEHT'S AUCH	SCHROEDER	19	11
GRAFIK	EINF.GRAFIKPROGRAMM: PLOT/BAS	SCHROEDER /ARPS	14	29
GRAFIK	LINIENMODELLE IN 3 DIMENSIONEN	SCHROEDER/ ARPS	14	35
GRAFIK	GRAFIK-CLS fuer das GENIE IIIs	SOPP	13	35
GRAFIK	STRINGANZEIGE MIT VOLLDAMPF	SOPP	6	41
GRAFIK-JKL	GRAFIK-JKL fuer das GENIE IIIs	SOPP	13	32
GRUNDLAGEN	Digitaltechnik, Grundlagen der -	gesammelt: KUHN		
HAENDLER	Reingefallen alt Rheintec	SCHMITZ	23	90
HAENDLER	TANDY SCHLIESST COMPUTER-CENTER	ZEITSCHRIFT	18	75
HAENDLER	TANDY-VERTAGSHAENDLER		19	79
HARDCOPY	TRS80 erstellt HARDCOPY	aut nc	23	17

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
HARDCOPY	HARDCOPY AUCH OHNE INTERRUPT	RYCHLIK	20	15
HARDCOPY	FLOPCOPY AUCH OHNE INTERRUPT	SOPP	21	14
HRG	Traumlandschaften aus dem Computer	aus Comp.Pers.	27	41
HRG	Wie funktioniert ein Vektorgenerator	aus Comp.Pers.	27	31
HRG	HRG-1b-Karte	MUELLER, KURT	14	89
HRG	HRG vom BASIC aus (DEMO)	NEUEDER	10	31
HRG	HRG-PIXELS aus dem BASIC!	NEUEDER	10	23
HRG	HRG: 384x192 oder 192x192	NEUEDER	13	14
HRG	HRG-BILDUEBERTRAGUNG per DFUe	OBERMANN	14	24
HRG	UMKEHR!	OBERMANN	20	58
HRG	HRG-Bilduebertragung	OBERMANN	25	31
HRG	HRG 1b PROGRAMMIEREN, BESSER ERKLAERT	SOPP	10	25
HRG	HRG UND BASIC NETTO	SOPP	10	27
HRG	HRG-SPEICHER LOESCHEN, DEN -	SOPP	10	29
HRG	GENIE IIIs, DIE HRG DES -	SOPP	12	21
HRG	HRG-HARDCOPY fuer das G IIIs	SOPP	13	25
HRG	HRG, dasz die Schwarte kracht !	SOPP	24	40
IC	Hardware-Tips: Neue IC's	ZEITSCHRIFT	17	38
INTERFACE	Relais-Interface	?	9	47
INTERFACE	Interface-Adapter 8255, Programmierbarer	KUHN	13	47
INTERRUPT	Unterbrechung gestattet ?	SOERENSEN	28	52
JKL	JKL - ABER DRUCKERSCHONEND!	SOPP	12	13
JOYSTICK	Joysticks fuer den TRS-80 M1	NEUEDER	5	?
JOYSTICK	Joystick-Anschluss	SOPP	6	29
KASSETTE	CASSETTEVERZEICHNIS ERSTELLEN	WAGNER, G.	10	33
KOPIERSCHUTZ	6.3 - KOPIERSCHUTZ	HERMANN	21	55
LAUFWERK	EXP1Floppy-Controller	BERNHARDT	26	38
LAUFWERK	Floppydisk-Controllerkarte	DROWAELDER	18	49
LAUFWERK	Floppy-Tricks	OBERMANN	14	87
LAUFWERK	80/40-Trackumschaltung Mitsubishi	OBERMANN	26	39
LAUFWERK	LWT fuer GENIE I und TRS-80 MODELL I	RETZLAFF	13	40
LAUFWERK	Ein klebriges Problem	SCHMID, A.	28	71
LAUFWERK	Ordnung muss sein !	SOPP	28	33
LIBRARY	WAS TUN BEI VOLLER LIBRARY?	SOPP	12	15
LITERATUR	BUECHERLISTE: DIE ERSTEN KURZKRITIKEN	SCHROEDER	21	67
LITERATUR	WEITERE KURZKRITIKEN	VERSCH.	22	DIV
LOGIK	STENO-LOGIK	MUEHLENBEIN	16	24
LOGIK	BOOLE XOR ABEL	MUEHLENBEIN	17	16
LOGIK	LOGISCHE OPERATIONEN IN BASIC	OBERMANN	15	9
LOGIK	LOGISCHE OPERATIONEN 2	OBERMANN	16	9
LOGIK	LOGISCHES ZU LOGIK	SOPP	16	14
LOGIK	LOGIKSIMULATION mit BASIC	ZEITSCHRIFT	16	11
M4/4p	AUFRUF DER SUPERVISOR-CALLS in BASIC	HERMANN	14	62
M4/4p	DAS BOOT-ROM des 4p	OBERMANN	14	63
M4/4p	HRG fuer MODEL 3/4/4p	OBERMANN	14	67
M4/4p	Atari-Bilder auf den 4p	OBERMANN	26	21
M4/4p	Das neueste DOS fuer M4/4p	OBSCHERNINGKAT	24	29
M4/4p	AUFGESTIEGEN?	WAGNER, G.	14	59

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
MATH.STATIST	BIVARANA (2fache Varianzanalyse)	MUEHLENBEIN	19	21
MATHE/GRAFIK	RECHNEN (ZEICHNEN) MIT DEM COMPUTER...	SCHROEDER	10	53
MATHEMATIK	WIE der NAT.LOG.die RECHENZEIT VERKUERZT	GUAGLIANO	14	77
MATHEMATIK	SPLINE-INTERPOLATION (KUBISCH)	KASPER	16	19
MATHEMATIK	PRIMZAHLBERECHNUNGEN (3 PROGRAMME)	KASPER	5	?
MATHEMATIK	GOLDENER FIBONACCI	MUEHLENBEIN	16	21
MATHEMATIK	UMWANDLUNG! ROEMISCHE IN ARABISCHE ZAHL	MUEHLENBEIN	16	7
MATHEMATIK	DAS CARTESISCHE BLATT	MUEHLENBEIN	18	5
MATHEMATIK	CRAMER UND DIE DETERMINANTEN	MUEHLENBEIN	19	13
MATHEMATIK	DER SCHNELLSTE ZAHNARZT (WURZELZIEHEN)	MUEHLENBEIN	22	11
MATHEMATIK	Polynonberechnung	MUEHLENBEIN	24	13
MATHEMATIK	Ableitg.d.Negationsregel f.Binaerzahlen	MUEHLENBEIN	25	9
MATHEMATIK	Ein letzter 10Zeiler (Determinantenber.)	MUEHLENBEIN	28	10
MATHEMATIK	PASCAL St GALTON	MUEHLENBEIN	28	17
MATHEMATIK	Sitzverteilung nach d'Hondt	MUEHLENBEIN	2B	11
MATHEMATIK	Fraktale Geometrie	SCHROEER	27	15
MATHEMATIK	Bin.-Umwandlg.HEX/DEZ;kuerzere E-Routine	SOPP	24	23
MATHEMATIK	Binaeruwaandlung, letzter (?) Teil	SOPP	25	12
MATHEMATIK	Was man weiss - schon inner wissen wollt	WOLF, Thomas	26	15
MATHE-SPIEL	WAS IST MIT DER PRIMZAHL '37' LOS?	MUEHLENBEIN	11	23
MAUS	Die Maus am TRS-80 !	OBERMANN	22	60
MAUS	Maus mit sieben Beinen	SOPP	23	69
MAUS	Die Pratze laeszt das Mausen nicht	SOPP	25	18
MEMDISK	RESET-FESTE MEMDISK M4-CP/M	OBERMANN	19	43
MODEL 4	BOOT-ROM-VERSION und DATUM M4	HERMANN	19	42
MODEL 5	TURBO TIP	OBERMANN	21	31
MODEL 6	KLEINE ZIPPERLEIN RASCH KURIERT	WOLLSCHLAEGER	12	57
MODEM	Akustikkoppler	TRAPP	4	?
MONITOR	MONITOR IM GENIE	KASPER	6	23
MUSIK	MEIN ZWEITES ASSEMBLERPROGRAMM	MUEHLENBEIN	15	15
MUSIK	SCHWERE KAVALLERIE ("DAS CLUBLIED")	MUEHLENBEIN	22	17
PASCAL	Echte Progranne neiden PASCAL	aus DATAMON7/83	23	91
PASCAL	APFELMAENNCHEN auf der GDP64	BERNHARDT	21	37
PASCAL	GENIE, CP/M und 80 ZEICHEN und HRG ???	HELD	21	43
PASCAL	PASCAL-PROBLEME	KONRAD, J.	9	73
PASCAL	Vergleich BASIC nit TURBO-PASCAL	MUEHLENBEIN	28	39
PASCAL	TURBO PATCH	OBERMANN	21	32
PASCAL	TURBO PATCH (FEHLERKORREKTUR)	OBERMANN	21	32
PASCAL	TURBO TIP	OBERMANN	21	31
PASCAL	TURBO-TIP	OBERMANN	21	31
PASCAL	Vergleichstest (versch.PASCAL-Versionen)	OBERMANN	23	33
PASCAL	KURZKRITIKEN und BUECHERLISTE	SCHROEDER	21	71
PASCAL	PASCAL-VORSTELLUNG TEIL 2	SOERENSEN	22	40
PASCAL	PASCAL fuer INTERESSIERTE	SOERENSEN, R.	20	12
PASSWORT	PASSWORT-BYPASS BEI TRSDOS 6.2	OBERMANN	19	42
PATCH	GROSZBUCHSTABENMODUS AB SYSTEMSTART (M4)	HERMANN	21	30
PDRIVE	PDRIVE-IDENTIFIKATION fuer NEWDOS 80	OBERMANN	10	17
PLOTTER	KREIS OHNE TRIGONOMETRIE	SCHROEDER	15	8

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
POKEPEEK	POKE+PEEK-ECKE	HANS -?-	4	?
PORT	PORTBELEGUNG von COMPUTERKONFIGURATIONEN	KUHN	21	44
PORT	Portbelegung	KUHN	21	44
PORT	OUT,PORT#,xx,yy,...	SOPP	11	29
PORT	VERBESSERT: IO.PORT#<,xx,yy,.. >	SOPP	12	11
PROGRAMMIERG	STRUKTURIERTE PROGRAMMIERUNG	?	7	17
PROGRAMMIERG	KEINE MEINUNG ZUR STRUKTUR?	GRELL	7	28
PROGRAMMIERG	EDIT-BEFEHLE	HERMANN	14	4
PROGRAMMIERG	BASIC- LAUFENDE PROGRAMME ERHEITERN	KRENZKE	16	18
PROGRAMMIERG	LOKALE VARIABLE BEIM TRS 80	PAULO	7	7
PROGRAMMIERG	PLANVOLL PROGRAMMIEREN	SACHSE	7	23
PROGRAMMIERG	DIRECTORY LESEN	WAGNER, G.	9	4
PROGRAMMIERG	EDITOR	WAGNER, G.	9	5
PROGRAMMIERG	TRICKS MIT STRINGS	WOLLSCHLAEGER	4	45
PROGRAMMIERG	PROGRAMMIEREN HEISST m. DATEIEN ARBEITEN	WOLLSCHLAEGER	7	29
PROGRAMMIERG	DEM INTERPRETER auf» BIT GESCHAUT	ZEITSCHRIFT	12	37
PROGRAMMIERG	DIE RICHTIGE DIMENSION	Zeitschrift	18	24
PROMMER	EPROM-Programmiergeraet	WOLF (aus mc)	5	?
PROZESSOR	Z80, HD64180 UND ILLEGALS	SCHROEDER	17	27
PROZESSOR	TAV's Z80-TUNING	ZEITSCHRIFT	17	33
RAM	512K-RAM-ADRESSEN m. FLEXIBLER MMU	BERNHARDT	18	67
RAM	EXTRA RAM ("Ein sicheres Plaetzchen...")	OBERMANN	9	63
RAMDISK	Die RAMDISK und du 11.Gebot	SCHMID, A.	25	54
RAMDISK	Die RAMDISK und das elfte Gebot	SCHMID, A.	25	56
RAMDISK	Darf's ein bisserl mehr sein?	SCHMID, A.	26	46
RAMDISK	MINI-RAM-FLOPPY im "SICHEREN PLAETZCHEN"	SOPP	10	37
RECHT	URHEBERRECHTSSCHUTZ fuer PROGRAMME	KAHLEN	10	55
RECHT	COMPUTERKRIMINALITAET		19	81
RECORDORGAN.	ZU GERALDs ARTIKEL ueber LMOFFSET	SOPP	10	3
RESTORE	BASIC-PATCH FUER RESTORE N	OBERMANN	5	?
ROM-ROUTINEN	CALLERLEI	MUEHLENBEIN	20	27
SCHALTALGEBR	KLEINES LOGIK-LEXIKON	aus c't	19	41
SCHALTSTATUS	WACHHUND, Der-, b.unkontr.Schaltzustand	BETZ	20	63
SCHNITTSTELL	Die Parallelschnittstelle	aus Comp.Pers.	27	49
SCHNITTSTELL	Entfernung der Kassettenschnittstelle	SCHROEDER, G.	26	40
SCHNITTSTELL	Memory mapped I/O <==> Port I/O bei GIIs	SCHROEDER, G.	26	42
SEMANTIK	TRAU - SCHAU, WEM! Ein Bitchen Englisch	MUEHLENBEIN	18	47
SENSOR	Digitales Temperatur-Zeit-Uhrenmodul	Obermann	16	41
SICHERHEIT	RAM-DISK KOSTENLOS	WOLLSCHLAEGER	2	?
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES	SOPP	13	23
SONDERTASTEN	SONDERTASTEN DIVERSER GENIES (NACHTRAG)	SOPP	14	53
SORTIEREN	TRS80: Sortieren	aus mc	23	12
SORTIEREN	IN VISICALC SORTIEREN	MUEHLENBEIN	20	93
SORTIEREN	Ordnung musz sein !	MUEHLENBEIN	28	13
SPEICHER	Sicheres, Ein -, Plaetzchen	BERNHARDT	7	43
SPEICHER	CALL UM DIE ECKE (MEM-BEREICHE ERREICHEN	SOPP	15	49
SPIEL	Lottozahlen	MUELLER, Kurt	24	19
SPIEL	JOYSTICK AN MODEL 4/4p	OBERMANN	19	50

STICHWORT [1]	TITEL aus den Club-80 Infos 31/12/1989	*VERFASSER* [2]	Info	Seite
SPIEL	TAPE-GAMES AUF MODEL III/4/4p	OBERMANN	20	57
SPIEL	ADVEN-80	SCARGILL	11	9
SPIEL	DAS PIRATE ADVENTURE	SCHROEDER	11	15
SPIEL	ADVENTURE GEHEIMAGENT XP-05	WAGNER, ALEX.	3	?
SPRACHE	DER-DIE-DAS MACHT MIR SPASZ	MUEHLENBEIN	19	75
SPRACHE	Welche Sprache fuer wen ?	SCHMID, A.	27	59
SPRACHE	COBOL - EINE EINFUEHRUNG	TRAPP	5	?
SPRACHE	COBOL-EINFUEHRUNG		8	5
SPURANZAHL	80/4O-TRACKUMSCHALTUNG	KUHN	19	49
SYSTEM	Quatsch in OVL4/SYS	SOPP	24	27
SYSTEM	Schon Mieder SYS14/SYS	SOPP	24	20
TAKT	TAKTUMSCHALTUNG - DIE ALLERLETZTE LOESG.	MUELLER, KURT	14	45
TAPE	SUPERTAPE FUER TRS 80	BURGWITZ/STILLR	7	44
TASMON	ERWEITERUNG FUER TASMON (NEWDOS-BEFEHLE)	KONRAD J.	4	25
TASTATUR	UNIVERSALGENIE	aus c’t	19	57
TASTATUR	Tastatur- und Drucker-Platine	DROWAELDER	20	71
TASTATUR	UMLAUTE IM NEWDOS? ABER NATUERLICH!	NEUEDEER	10	41
TASTATUR	KNOEPFE	OBERMANN	14	57
TASTATUR	ES GEHT NOCH SCHNELLER (TAST.ABSCHALTEN)	SCHMID, A.	22	32
TELEFON	TELEFONBUCH MIT PFIFF	FRISCH	2	?
TEXT-GRAFIK	(TELEFON) - EIN UD-ZEICHEN auf d.DRUCKER	SOPP	16	29
TEXTSYSTEM	KLEINES TEXTPROGRAMM	aus mc	23	10
TEXTSYSTEM	UMDEFINIERUNG v.SCRIPSIT-KOMMANDOTASTEN	KONRAD	8	14
TEXTSYSTEM	KEINE ANGST VOR... (TRENNEN IN TSCRIPS)	MUEHLENBEIN	20	61
TEXTSYSTEM	Was TSCRIPS nicht kann	MUEHLENBEIN	23	20
TEXTSYSTEM	TSCRIPS retten	MUEHLENBEIN	2B	3B
TEXTSYSTEM	DUMP (TSCRIPS retten)	MUELLER, Kurt	26	18
TEXTSYSTEM	SCRIPSIT AUF DEM ALTEN GENIE	OBERMANN	14	43
TEXTSYSTEM	Wie sag' ich's meinem WORDSTAR ?	SCHMID, A.	24	11
TEXTSYSTEM	WORDSTAR auf allen Laufwerken	SCHOBER	28	35
TEXTSYSTEM	AENDERUNGEN IN TSCRIPS und NAME/CMD	SCHROEDER	13	16
TEXTSYSTEM	FREIRAEUME IN TSCRIPS	SCHROEDER	14	47
TEXTSYSTEM	SCRIPSIT-TABELLEN	SCHROEDER	14	45
TEXTSYSTEM	TEXTUMLEITUNG fuer TSCRIPS	SCHROEDER	14	49
TEXTSYSTEM	TSCRIPS-VERSCHLUESSELUNG	SCHROEDER	15	29
TEXTSYSTEM	TIPS fuer SCRIPS	SOPP	13	19
TEXTSYSTEM	NEUES ZUR PROPORTIONALSCHRIFT	SOPP	18	15
TEXTSYSTEM	§ CONTRA .	SOPP	19	27
TEXTSYSTEM	TSCRIPS & DIE &-CODES	SOPP	19	16
TEXTSYSTEM	VON DER WIEGE BIS ZUR BAHRE (ADR.AUTOMAT	SOPP	19	33
TEXTSYSTEM	KEINE ANGST VOR PROPORTIONALSCHRIFT	SOPP	18	13
TEXTSYSTEM	TSCRIPS UND DIE 24 NADELN	SOPP	22	23
TEXTSYSTEM	DAS TSCRIPS-DRAMA - EIN LEHRSTUECK	SOPP/MUHLENBEIN	21	56
TONERZEUGUNG	DA CAPO... (WIE MAN TOENE ERZEUGT)	MUEHLENBEIN	22	13
TONERZEUGUNG	TONGENERATOR FUER TRS 80	PLUMHOFF	B	I9
TRS 80	DER BEKNACKTE TRS 80	ROECKRATH	6	33
TRS-80	TRS-80 Japan-Modell aufbohren	BERNHARDT	25	60
TRS-81	TRS-80, Der - und die Aussenwelt	KLEIN, WILFRIED	4	?

STICHWORT [1]	**TITEL aus den Club-80 Infos 31/12/1989**	***VERFASSER*** [2]	**Info**	**Seite**
TRS-82	Umbau eines TRS-80 Mod.I 16K auf 48K	RETZLAFF	10	45
UHR	UHRUNTEN fuer MODEL 4p	OBERMANN	15	47
UHR	Es war einmal ein Datum	SCHMID, A.	26	29
UHR	WENN DIE UHR MAL STOERT	SOPP	14	39
UHR	THE TIMES, THEY ARE A-CHANGING	SOPP	22	35
UMLAUTE	Umlaute auf der Tastatur des GENIE I	DRECHSLER	28	69
UTILITY	PAGELIST/CMD	aus Zeitschrift	1B	21
UTILITY	EINGABEROUTINE MIT KOMFORT	BULLING	7	13
UTILITY	Startprogramm f. BASIC-Programme	DRECHSLER	28	21
UTILITY	SHELL/CMD	HERERMANN	24	34
UTILITY	TRACE-ON	HOEH (COMP.PERS	4	?
UTILITY	AUSGABEUMSCHALTUNG IN BASIC	KONRAD J.	6	24
UTILITY	ALLERLEI NUETZLICHES FUER DEN TRS 80	LICHTERMANN /MC	4	?
UTILITY	FILES WIE SAND AM MEER...	MUEHLENBEIN	11	7
UTILITY	DIRECTORY LESEN UND KOMMENTARE SPEICHERN	MUEHLENBEIN	13	15
UTILITY	FILEVERGLEICH (s.auch Progr.-Bibliothek)	NEUEDER	10	43
UTILITY	DDE St LWT	OBERMANN	10	19
UTILITY	LPRINT ALLES einfacher	OBERMANN	8	3
UTILITY	ZAP IM SUPERZAP/Zappen im Zeichenbereich	OBERMANN	8	13
UTILITY	LPRINT ALLES	SOPP	6	31
UTILITY	BILDSCHIRMAUSDRUCK BEIM TRS80+VIDEOGENIE	UEBERSCHAAR		
UTILITY	SMARTSCREEN, SMARTKEY und SMARTPRINT	ZEITSCHRIFT	12	45
VERTRETUNGEN	TRACE od.DIE GESCHICHTE EINER ZEITSCHR.	OBSCHERNINGKAT	18	75
VIDEO	Das 80-Zeichen-Karte-Problem u.d.Loesung	BERNHARDT	22	63
VIDEO	BO-Zeichen-Karte	SCHROEDER, G.	12	63
VIDEO	VIDEODAT-SYSTEM		1B	57
VISICALC	Korrigieren und Steuern innerhalb VC	MUEHLENBEIN	23	24
VISICALC	SORTICALC	MUEHLENBEIN	26	11
Z80	ARBEITSTIER MIT GESUNDEM HERZEN	BETZ	21	52
Z80	INTERFACE mit Kassetten-Schnittstelle	SCHROEDER, G.	26	40
Z81	422 NEUE Z80-BEFEHLE	HASSELBERG (MC)	4	?
ZAPPEN	EIN BYTE UND SEINE FOLGEN	OBERMANN	10	21
ZEICHENSATZ	Geheimnisse eines Zeichengenerators	GIESE	28	27
ZEICHENSATZ	NEUE ZEICHENSAETZE fuer das GENIE IIIs	SOPP	12	28
ZEITMESSUNG	ETWAS fuer LEUTE, die NIE ZEIT haben	SCHMID, A.	22	31
STICHWORT [1]	**Der hochgestellte Index gibt die Reihenfolge der Sortierung an**			
VERFASSER [2]				

Veroeffentlichungen Z80-, HD64180-, Z180-Computer

alle Artikel, wenn nicht anders angegeben, von Helmut Bernhardt

2015/06		**Erratum 256K Banker**
1984-Genie-Data-6	S.33-46	192K RAM im GENIE (& Christof Ueberschaar)
1985		Amiga Sidecar mit 4 Raedern (& Christof Ueberschaar)
TRS80 und Genie User Club Bremerhaven		
1985-BHV-11	S.15-19	Voll decodierter Sonder-ROM
1985-BHV-11		RAM bei 3000H-37DFH
1985-BHV-11		**Hardware-Sonderheft**
	S.1-5	Wie aus dem Komtek ein Computer wird
	S.6-10	Hardware Umschaltung fuer CP/M (EG 64 MBA+)
	S.13-16	Ver16fachung des Sonder-ROM
	S.17-22	...und dazu etwas Software (Arnulf Sopp)
1985-BHV-12	S.10-11	Ein Hilferuf (IM2 mit Z80-CTC)
1985-BHV-12	S.29	Fehlermeldungen
1986-BHV-01	S.10-12	Verbesserter Zeichensatz fuer Komtek 1
1986-BHV-02	S.11-17	CP/M Hardware (& Christof Ueberschaar)
1986-BHV-03	S.5-10	256K RAM fuer Z80 Systeme
1986-BHV-04	S.13-15	Parallel I/O-Adapter mit 8255
1986-BHV-06	S.17-19	1 MByte im Genie III
1986-BHV-06	S.24-29	ECB-Bus beim Genie I und II
1986-BHV-11	S.15-16	256K fuer Z80-Systeme - und wie es weitergeht (Gerald Schroeder, Alexander Schmid)
1986-BHV-12	S.7-10	Selbstbau Doubler fuer EXP1
1987-BHV-02	S.26-31	Adapterboard fuer HD64180 im GIIIs hard
1987-BHV-02	S.26-31	Adapterboard fuer HD64180 im GIIIs soft (Arnulf Sopp)
1987-BHV-02	S.31-36	HD64180 statt Z80 im GIIs
1987-BHV-03	S.13-18	Auslesen der GDP64 Grafikkarte (& Joerg Seelmann Eggebert)
1987-BHV-03	S.19-21	Noch'ne Geschwindigkeitserhoehung
1988-BHV-06	S.17-19	1 MB im G III
Club 80		
1986-Club80-14	S.91-99	256K fuer Z80-Systeme
1986-Club80-??		Dein GIIIs - die 4 unbekannten Wesen (& Arnulf Sopp)
1986-Club80-17	S.8-15	Wie Phoenix aus der Asche (Gerald Schroeder)
1986-Club80-17	S.27-32	Z80, HD64180 und Illegals im ROM-Modus (Gerald Schroeder)
1987-Club80-18	S.67-70	HD64180 statt Z80 im GIIs
1987-Club80-19	S.49	Dobler mit WD2793
1987-Club80-20	S.65-70	Auslesen der GDP64 Grafikkarte (& Joerg Seelmann Eggebert)
1987-Club80-21	S.37-42	Apfelmaennchen auf der GDP64
1987-Club80-21	S.53-54	Was habt Ihr gegen CP/M?
1987-Club80-22	S.71	Pruefzahlfehler beim Lesen
1987-Club80-22	S.63-66	Das 80-Zeichenkarten Problem
1988-Club80-23	S.73	Anmerkungen zum Club80-ECB-Bus
1988-Club80-23	S.74-84	Gepufferter IM2- und DMA-faehiger ECB-Bus
1988-Club80-24	S.40-42	HRG, dass die Schwarte kracht (Arnulf Sopp)
1988-Club80-25	S.57-59	Prof180 und mc68000-ECB
1988-Club80-25	S.60-68	TRS80 Japan Modell aufbohren

1989-Club80-26	S.37	Bessere Darstellung v. Text u. HRG
1989-Club80-26	S.38	Schluss mit EXP1-FDC Problemen
1989-Club80-26	S.72	Star NL10 abwechselnd an CP/M und PC
1989-Club80-28	S.73-77	Eleganter Adressdecoder fuer Genie I
1989-Club80-28	S.78-99	Es lebe der Standard (Hercules-Genie)
????-Club80-??		Dein Genie 3s - die 5 unbekannten Wesen (& Arnulf Sopp)
		HD64180-Sonderinfo
	S.1	HD64180-01-Interne Register
	S.7-9	HD64180-07-MMU
	S.10-14	HD64180-10-DMAC
	S.15-19	HD64180-15-serielle Schnittstellen (Gerald Schroeder)
	S.20	HD64180-20-Timer (Gerald Schroeder)
	S.21-22	HD64180-21-zusaetzliche Befehle
1990-Club80-30	S.30	Nachtrag zum Banker (Alexander Schmid)
1990-Club80-31	S.33	1MB-Banker mit 1MB (Alexander Schmid)
1990-Club80-31	S.17-22	Harddisk Einbindung ins CP/M (& Peter Petersen)
1990-Club80-31	S.23-24	Die clubeigene Grafikkarte (Andreas Magnus)
1990-Club80-31	S.25-32	Club80 Hercules Terminal (Konzept)
1990-Club80-32	S.46	Megabit-RAMs einfach erfrischt
1990-Club80-32	S.47	2MB RAM Erweiterung fuer Prof180 x
1991-Club80-??	S.25-34	Club80 Hercules Terminal 1.0
1991-Club80-34	S.9-12	Textkonvertierung IBM-PC nach CP/M
1991-Club80-34	S.39	NEWDOS, CP/M, MSDOS, UNIX
1991-Club80-35	S.41-42	Anmerkungen zum Club 80 Terminal (Alexander Schmid)
1994-Club80-43	S.5-14	VGA-Karte anstelle der Hercules-Karte im Club 80 Terminal

256K RAM für Z80-Systeme... ... und wie es weitergeht

Teil 1: Die Software

Gerald Schröder

SYSCOPY1 und RAMDISK für Helmut Bernhardt's Banker

Ihr habt ja alle hoffentlich aufmerksam den Artikel über Helmuts Banker und das erste Programm dafür, RAMSYS von Arnulf gelesen. Der Banker hat mich sofort fasziniert und so mußte er her. Leider stellt sich ein Problem: Auf dem Genie IIIs gibt es SYSCOPY und MEMDISK, die aber mit dem Banker nicht laufen. SYSCOPY leistet ähnliches wie RAMSYS, nur benötigt es absolut keine Systemdiskette mehr. Alle SYS-Files lassen sich aus dem Speicher laden. MEMDISK versorgt den Benutzer mit einer 128K-Disk als höchstes Laufwerk (vorhandene Laufwerke +1). Diese Ramdisk hat durch ihre Geschwindigkeit unheimliche Vorteile gegenüber den lahmen Diskettenlaufwerken.

Ich habe mir also erlaubt, auch ohne TCS oder Phoenix um Erlaubnis zu bitten, diese beiden Programme zu disassemblieren und auf Helmuts Banker umzustricken. Das ist nicht so problematisch, wenn man viel Zeit aufbringt und auf einige Disketten verzichten kann, die man beim Probieren zerschießt. Die Produkte heißen bei mir SYSCPOY1 und RAMDISK.

Im Gegensatz zu RAMSYS arbeitet SYSCOPY mit einem kleinen Directory, in das die SYS-Files beim Lesen eingetragen werden. Es dürfte mit allen Newdos/Gdos Versionen laufen, solange SYS0 nicht stark verändert wurde, was kaum zu befürchten ist. Die SYS-Files liegen dann in Bank 1 und 2, wobei in Bank 2 je nach Länge/Anzahl der Files noch mehr oder minder viel Platz ist, den man später problemlos nutzen kann. In Bank 1 stehen auch die Service-Routinen. Nach dem Aufruf lädt SYSCOPY erst einige Sekunden seine SYS-Files und dann kann man die System-Diskette entfernen. Beim Kopieren wird zwar noch danach gefragt, aber es kann auch eine Pappscheibe in Drive 0 liegen, wenn die als Diskette anerkannt wird. Alle SYS-Files werden aus den Banks geladen.

RAMDISK benutzt einigen Platz in Bank 2, den SYSCOPY frei läßt. Als RAM-Disk dienen die Banks ab 3 aufwärts. Beim Aufruf kann man die RAM-Disk in 64K-Schritten (also jeweils 2 Banks) von 64K-896K einstellen (falls Euer Banker soweit bestückt ist). Wenn schon vorher einmal RAMDISK gestartet worden war (auch nach einem Reset), bleibt der alte RAM-Disk Inhalt erhalten, wenn man nicht ausdrücklich eine Neuformatierung wünscht. Die RAM-

Disk wird wie ein Laufwerk angesprochen (beispielsweise läuft auch SUPERZAP!). Vor dem ersten Start muß man eine kleine Änderung am Source-Code vornehmen und das Programm neu assemblieren, weil jedes DOS (Newdos, Gdos, Hdos) einen anderen Initialisierungswert benötigt. Das dürfte aber nicht zu schwer sein.

Teil 2: das Opfer

Alexander Schmid

Wie das Schicksal so will, ist mir in einer langweiligen Stunde Helmut Bernhardt's Artikel wieder in die Hände gefallen. Interessant klingt das ja schon, aber wer repariert mir meinen Kasten hinterher? Solche oder ähnliche Fragen werden Euch damals wahrscheinlich auch gequält haben, aber keine Angst, es ist nur halb so schlimm, wie es aussieht.

Die Sache hat mir keine Ruhe mehr gelassen und so hatte ich einige Briefe später schließlich die kleine Platine in der Hand vor mir und wußte auch, wie ich sie ins GIIs einbauen muß. Die Fassungen und ICs besorgt und los ging's. In ein bis zwei Nachmittagen ist es leicht zu schaffen, je nach Ausdauer. Wenn man schon mal gelötet hat, dürfte es kein Problem geben. Die Anleitung von Helmut ist wirklich ausführlich und genau. Einschalten und... es läuft (fast). Zwei der ICs waren zu langsam für meine 8MHz (was Euch egal sein kann) und mußten gegen schnellere Typen ausgetauscht werden. Ansonsten hatte ich bis jetzt keinerlei Probleme. Der Kasten läuft sicher und zuverlässig wie eh und je.

So einfach kommt man für ca. DM 120 und etwas Eigeninitiative zu einem Genie oder TRS80 mit 256K RAM. Als Krönung hat Gerald Schröder, wie oben beschrieben, zwei Treiber auf die Beine gestellt, die es in sich haben. Wenn ich ohne SYSCOPY arbeite, tut mir das Systemlaufwerk mit seinem Geklapper richtig leid, und für das schnelle Abspeichern von Zwischenversionen ist eine Ramdisk einfach ideal. Sogar Compilerläufe machen damit fast Spaß.

Wenn jetzt noch einer sagt, das ist nichts für ihn, der weiß nicht was er perpaßt. Bei Geräten mit einem Laufwerk kann man dann z.B. auch reine Datendisketten verarbeiten, weil das DOS ja aus dem RAM geladen wird. Vor allem bei kleinen Formaten wird so eine Menge Platz frei und beim Kopieren entfällt der (meist vergebliche Wunsch: 'hoffentlich hat der ein DOS auf der Scheibe'.

Mit einer zweiten Reihe RAMs (huckepack auf den anderen) habe ich inzwischen 512K und eine entsprechende große Ramdisk zur Verfügung. Wer will, kann bis max. 1 MB realisieren.

Das wär's eigentlich von der 'User-Seite'. Wer noch irgendwelche Fragen hat, soll sich ruhig melden, sonst

hätten wir den Artikel ja nicht schreiben brauchen. Die Porgamme SYSCOPY und RAMDISK stehen jedem Mutigen frei zur Verfügung.

Die Adressen von Helmut und mir stehen im Info, die von Gerald ist: ....

1986-Club80 Nr.17 S.8-15

Wie Phönix aus der Asche

Vier Computer in einem Schrotthaufen

Gerald Schröder

Die Beilage von Arnulf und Helmut im letzten Info hat Euch sicherlich geschockt, denn schließlich verfügt nicht jedermann über einen IIIs, den er aufmotzen kann. Mir fehlt zwar das Kleingeld für einen IIIs, aber deshalb gönne ich den IIIs-Besitzern diesen Triumpf noch lange nicht. Deshalb habe ich kurzerhand Arnulfs Idee abgekupfert und auf den schon in der Überschrift erwähnten TRS80 angewandt. Wer sich jetzt extra einen IIIs gekauft hat, um mit Arnulf gleich ziehen zu können, ist selbst schuld.

Wie sieht das in der Praxis aus? Ganz einfach, Ihr belagert einfach den schon mehrmals erwähnten Helmut, bis er Euch einen Banker einbaut. Dann tippt Ihr das folgende Programm ab oder laßt es Euch vom nächsten Diskothekar schicken. Wenn Ihr dann <Shift><Down-Arrow><Ziffer> betätigt, wobei Ziffer := 0 bis 3, könnt Ihr zwischen 4 Computern hin- und herschalten. In jedem Computer kann ein Programm laufen, solange dieses Programm

1. die Interrupts abgeschaltet läßt
2. keine fiesen Nebeneinbauten benutzt.

Punkt 2 bezieht sich auf Graphikkarten o.ä. Damit wir uns richtig verstehen: benutzen darf das Progamm sie schon, solange beim Sprung zu einem anderen Computer die Sache wieder ausgeschaltet ist. Der weiß nämlich überhaupt nichts davon.

Gerettet wird der Speicher ab 3000h aufwärts. Darunter sollte ein festes ROM sein, das gefälligst nicht geändert wird.

Einige interne Notierungen:

- die Umschaltung ist in die normale Interrupt-Routine des Gdos/Newdos-SYS0 eingebaut.
- alle Routinen liegen in Bank 1
- die Banks 1, 2 und 3 enthalten die "unteren" Hälften der Computer 1-3, jeweils von B000-FFFF
- die Banks 0 und 4-6 enthalten die "oberen" Hälften der Computer 0-3
- die "untere" Hälfte von Computer 0 wird jeweils mit der des momentan aktiven Computers getauscht.
- es werden jeweils nur die "unteren" Hälften vertauscht (weil das Common nun mal eine untere Bank ist). während die "oberen" Hälften nur umgeschaltet werden müssen
- d.h.: wenn der Computer 1 auf 3 umgeschaltet wird, wird zuerst Computer 0 selektiert, bevor Computer 3 endgültig ans Rohr kommt
- deshalb spielt sich auf dem Bildschirm bei einer solchen Umschaltung kurz ein Wirrwarr ab, das aber nur mühsam zu umgehen wäre
- das Vertauschen läft folgendermaßen ab:
  - (von Bank 1 aus) 7E00-7FFF nach Bank 1 retten
    - Tausch-Routinen nach 7E00 (ins Common) übertragen
    - Tausch-Routinen starten
      - gerettetes 7E00 ff. mit dem aus sel. Bank tauschen
      - Common 3000-7DFF mit sel. Bank tauschen
      - (dabei Buffer 7E00-7FFF benutzt)
      - zurück nach Bank 1
    - geretteten Bereich 7E00 ff. ins Common
    - stareten des Computers
- um unerwünschte Nebeneffekte (wie Druckerausgabe) auszuschließen, wird der I/O-Bereich 3700-37FF nicht gerettet
- das Programm ist lange nicht so elegant wie Arnulfs Lösung, aber es tut seinen Dienst.

So, das war's schon, besorgt Euch 256kB und los geht's mit SIDEKICK

Gerald Schröder

Hardware: Helmut
Literatur: Arnulfs Listing

```
;               SIDEKICK für TRS80-Kompatible mit Helmuts Banker
;               im Dez. 1986 von Gerald Schröder
;               nach einer Idee von Arnulf Sopp


        ORG     3300h       ;Stack ab hier abwärts!

start   LD      (spbuf),SP  ;SP retten

        LD      HL,retpro   ;Rückkehr-Adresse (bei 1.
        USH     HL          ;Aufruf Comp. 1-3)
        LD      B,4         ;Stack vorbereiten
loop1   PUSH    AF
        DJNZ    loop1
        LD      (rettsp),SP ;retten für Comp. 1-3

        LD      A,1         ;Bank
        DI
        OUT     (0ech),A    ;einschalten
        LD      HL,schalt   ;Umschalt-Routine
        LD      DE,schalt-diff    ;in Bank 1
        LD      BC,ende-schalt
        LDIR
        XOR     A
        OUT     (0ehc),A    ;wieder Bank 0

        LD      HL,umleit   ;Umleitung in Interrupt-
        LD      DE,umleit-diff1  ;Routine anlegen
        LD      BC,ende1-umleit
        LDIR

        LD      HL,zurück   ;Rücksprung
        LD      DE,zurück-diff2  ;in SYS0 anlegen
        LD      BC,ende2-zurück
        LDIR

        LD      SP,start    ;SP neu setzen

        LD      B,3         ;drei Computer
loop2   CALL    copy        ;in die Banks
        DJNZ    loop2

        EI
        LD      SP,0000           ;SP zurück
sobuf   EQU     $-2

retpro          RET               ;ab
```

```
;               kopiert oberen und unteren Teil in Banks

        PUSH  BC         ;Zähler retten
        LD    A,B        ;als Bank-Nr.
        OUT   (0ech),A   ;für unteren Teil
        LD    DE,8000h+3000h   ;Ziel = 0b000h
        LD    D,50h      ;Länge = 5000h
        LD    C,E
        LD    H,30h      ;Anfang 3000h
        LD    L,E
        LDIR             ;(3000-7fff)=>(b000-ffff)

        ADD   A,3        ;für oberen Teil
        EX    AF,AF'           ;retten
        LD    D,60h      ;Buffer: 6000h-6100h

 loop3  INC   B          ;ein Sektor
        XOR   A          ;Bank 0
        OUT   (0ech,A          ;ein
        LDIR             ;Sektor aus 0 in Buffer
        DEC   D          ;Sektor-Zeiger auf 6000h
        LD    A,H        ;Quelle- als Zielzeiger
        LD    H,D
        LD    D,A
        DEC   D
        INC   B
        EX    AF,AF'           ;Bank-Nr. zurück
        OUT   (0ech),A   ;anwählen
        EX    AF,AF'           ;wieder rettten
        LDIR             ;Aus Buffer in Bank
        DEC   H          ;Buffer auf 6000h
        LD    A,D        ;und wieder tauschen
        LD    D,H
        LD    H,A
        OR    L          ;schon über ffffh hinaus?
        JR    NZ,loop3   ;nein
        OUT   (0ech),A   ;ja, Bank 0 ein
        POP   BC         ;Zähler zurück
        RET



 ;              Umleitung in der Interrupt-Routine

 diff1  EQU   $-45f6h
 umleit       LD    A,1        ;Bank 1
        OUT   (0ech),A   ;ein
        JP    schalt-diff;und springen
 ende1  EQU   $
```

```
;           Rückkehr-Routine

diff2 EQU   $-4053h

zurück      OUT   (0ech),A    ;Bank x für oben an
      JP    45fdh       ;weiter
ende2 EQU   $



;           Schaltroutine in Bank 1
;           Ansprung von Interrupt-Routine aus

diff  EQU   $-3600h
schalt      LD    HL,3880h    ;Tast.-Zeile
      BIT   0,(HL)            ;SHIFT?
      JR    Z,goon3           ;nein, ab
      SRL   L
      BIT   4,(HL)            ;Down-Arrow?
      JR    Z,goon3           ;nein, ab
      SRL   L
      SRL   L           ;Tast.-zeile 3810h
      LD    A,(HL)
      AND   0fh         ;Taste 0 bis 32
      JR    Z,goon3           ;nein, ab
      LD    L,A         ;retten
      XOR   A           ;Zähler 0
goon1 INC   A           ;+1
      SRL   L           ;solange
      JR    NC,goon1    ;bis Taste erreicht
      DEC   A           ;Zähler korrigieren

      LD    (rettsp)    ;SP retten

      LD    DE,goon4-diff     ;Rückkehradresse
      JZ    spsuch            ;neuen SP berechnen

goon4 PUSH  IX          ;Register dieses COMP.
      PUSH  IY          ;retten
      EX    AF,AF'
      EXX
      PUSH  AF
      PUSH  BC
      PUSH  DE
      PUSH  HL
      LD    SP,ende-diff+150 ;zeitweilig neu

      EX    AF,AF'            ;Comp.-Nr. zurück
      CP    0           ;schon eingeschaltet?
momba EQU   $-1
      JR    Z,abgang    ;ja
      LD    (banku-diff),A   ;dort unterer Teil
```

```
      OR    A           ;Comp. 0?
      JR    Z,goon2          ;ja, oben = Bank 0
      ADD   A,3         ;sonst oben = u +3
goon2 LD    (banko-diff),A   ;Bank oberer Teil
      LD    A,(momba-diff)   ;Comp. 0 eingeschaltet?
      OR    A
      CALL  NZ,tausch-diff   ;nein, Comp. 0 holen
      LD    A,0         ;Bank unterer Teil
banku EQU   $-1
      LD    (momba-diff),A   ;sichern: mom. Comp.
      OR    A           ;falls nicht 0
      CALL  NZ,tausch-diff   ;Computer holen
abgang      LD    DE,goon5-diff
      JR    spsuch
goon5 LD    B,12        ;SP korrigieren
ab1   DEC   SP
      DJNZ  ab1
                             ;Register des
      POP   HL          ;Computers zurück
      POP   DE
      POP   BC
      POP   AF
      EXX
      EX    AF,AF'
      POP   IY
      POP   IX
      LD    SP,(rettsp);SP restaurieren
goon3 LA    A,(37ech)   ;urspr, Int.-rout.
      LD    A,(37e0h)
      RLCA
      LD    A,0         ;Bank-Nr. für oberen Teil
banko EQU   $-1
      JP    zurück-diff2     ;nach SYS0 für Abgang
;           Unterprogramm: Stack-Pos. für geg. Comp A
;           berechnen
spsuch      LD    L,A        ;Nr. nach L
      LD    H,0
      LD    SP,ende.diff+20  ;Stack-Offset
      LD    B,5         ;+32*Comp.-Nr.
sps1  SLA   L
      DJNZ  sps1
      ADD   HL,SP
      LD    SP,HL       ;SP in Bank 1 setzen
      EX    DE,HL       ;Rückspr.adr. nach HL
      JP    (HL)        ;und ab
```

```
;          Unterprogramm Tausch:
;          vertauscht unteren Teil von Computer 0 mit dem
;          eines der drei anderen Computer

tausch     LD    (bankx1-diff),A  ;gew. Bank-Nr. sichern
      LD    (bankx2-diff),A
      LD    (bankx3-diff),A

      LD    HL,anf-diff3     ;Bereich Tausch-Rout
      PUSH  HL          ;retten
      LD    DE,0a000    ;Buffer für Original
      LD    BC,0200h
      LDIR              ;Orig.-Code retten

      POP   DE          ;7e00h
      PUSH  DE
      LD    HL,anf-diff;Routinen zum Tauschen
      LD    BC,ende-anf
      LDIR              ;in die untere Hälfte
                        ;bei 7e00h
      LD    (stack-d;if),SP  SP retten
      LD    SP,7f00h    ;auch nach unten
      LD    HL,0fe00h   ;7e00h aus Bank x
      LD    DE,7f00h    ;Buffer
      LD    BC,9f00h    ;MSB (Buffer in Bank)
      CAL   anf-diff3   ;Aufruf Tausch-Rout.
      LD    SP,0000          ;SP zurück
stack EQU   $-2
      POP   DE          ;Fe00 zurück
      LD    H,9fh       ;von 9f00 aus
      INC   B           ;BC = 0200h
      LDIR              ;7300-7fff neu füllen
      RET


diff3 EQU   $-7e00h
anf   PUSH  BC          ;9f00 (x-1) auf Stack
      PUSH  DE          ;7e00 (z)    "    "
      LD    B,1
      LD    A,2         ;2 Durchläufe

;7e00-7fff tauschen
loopp EX    AF,AF'           ;Zähler retten
      LD    A,0
bankx1     EQU   $-1
      OUT   (0ech),A
      POP   DE          ;retten
      POP   AF          ;(x-1) zurück
      PUSH  HL          ;Y retten
      PUSH  AF          ;(x-1) retten
      PUSH  DE          ;z retten
```

```
        LDIR
        INC   B
        LD    A,1
        OUT   (0ech),A
        POP   HL             ;z zurück
        POP   DE             ;(x-1) zurück
        PUSH  HL             ;z retten
        LDIR                 ;y -> z, Bank 1 -> Buffer
        INC   B
        LD    A,1
        OUT   (0ech),A
        POP   HL             ;z zurück
        POP   DE             ;(x-1) zurück
        LDIR                 ;y -> z, Bank 1 -> Buffer
        INC   B              ;(Original-Code)
        LD    A,0
bankx2        EQU   $-1
        OUT   (0ech),A
        POP   HL             ;z zurück
        POP   AF             ;x zurück
        POP   DE             ;y zurück
        PUSH  AF             ;x retten
        PUSH  HL             ;z retten
        LDIR                 ;z -> y, Buffer -> Bankx
        INC   B              ; (Orig in Bank x)
        LD    H,D            ;(y+1)
        EX    AF,AF'
        DEC   A
        JR    NZ,loopp
;Stack: (x+1),z
        POP   DE             ;Buffer z
        POP   AF             ;(x+1) weg
        LD    H,30h          ;Anfang mit unten (3000h)
        LD    A,0
bankx3        EQU   $-1
        OUT   (0ech),A       ;Bank x ein
loop    PUSH  DE             ;"Buffer z" retten
        PUSH  HL             ;"von y" retten
        LDIR                 ;1. Sektor in Buffer
        POP   DE             ;"nach y" zurück
        LD    H,D            ;"von z" (aus Bank)
        SET   7,H            ;deshalb immer >= 8000h
        PUSH  HL             ;retten
        INC   B              ;ein Sektor
        LDIR                 ;2. aus Bank nach unten
        POP   DE             ;"nach x" (Bank)
        POP   HL             ;"vonZ" (Buffer)
```

```
        INC   B           ;ein Sektor
        LDIR              ;aus Buffer in Bank
        INC   B           ;ein Sektor
        DEC   H           ;Buffer bleibt
        LD    A,D         ;"von y"
        AND   7fh         ;immer unter 8000h
        LD    D,H         ;Buffer in de
        LD    H,A         ;"von y" in HL
        CP    37h         ;I/O-Bereich?
        JR    NZ,lop            ;nein, weiter
        INC   H           ;ja, überspringen
lop     CP    7eh         ;=fertig?
        JR    NZ,loop           ;weiter wenn <> 7e00
        LD    A,1         ;Bank1
        OUT   (0ech),A    ;wieder ein
        RET               ;fertig
ende    EQU   $-1
rettsp        EQU   4058h
        END   start
```

Z80, HD64180 und Illegals im ROM-Modus

Das Herz unseres kleinen Haufens Silikonmasse besteht bekanntermaßen aus einem Z80. Dieser Z80 kann über 1100 Befehle verarbeiten, von denen aber nur 694 von den Erbauern vorgesehen waren. Der Rest ist also Abfallprodukt.

Nun gibt es einen neuen Prozessor, den HD64180, der Z80-kompatibel ist, aber einige Sachen mehr kann (512Kb RAM verwalten. DMA, Schnittstelle...). Nur haben die Erbauer dieses Dings leider vergessen, die Befehle des Z80 zu berücksichtigen, die nicht offiziell dokumentiert sind. Das kann böse Probleme geben, denn wenn der neue Proz auf einen ihm unbekannten Befehl trifft, macht er einen Brutalo-Reset nach 0000h, was in unserem Fall jedes Programm ins Jenseits befördert.

Es wäre dies kein Problem, wenn niemand die "Illegals" bzw. die nicht dokumentierten Befehle benutzen würde, aber viele Programmierer tun es doch. Prominentes Beispiel der Editor/Assembler ZEUS (3 Illegals).

Ihr könnt nun

1. den neuen Prozessor vergessen (sehr schwer!),
2. alle Programme mit Illegals vergessen (nicht ganz so schwer),
3. alle Illegals durch normale Befehle ersetzen (schwer) oder
4. dem HD64180 die unbekannten Befehle beibringen (fast leicht).

Punkt 3 erschien mir brauchbar, bis sich ZEUS nach den Änderungen so merkwürdig benahm. Leider sind die Ersatz-Befehle für die Illegals nämlich einen kleinen Tick länger als die Illegals und das gibt big problems.

Also auf zu Punkt 4! Vorraussetzung: Veränderliches RAM/ROM bei 0000h: bei jedem CP/M-fähigen Rechner der Fall.

Außerdem habe ich in meinem kleinen Programm noch einiges ausgenutzt: Arnulf hett seggt[1], daß RST 00h nie gebraucht wird. Also benutze man RST 00h für etwas anderes. In diesem Fall bot sich ein Banking an, das Helmuts alten Banker simuliert. Verfahrensweise: Der Speicher von 8000h-ffffh wird als Common 1 definiert und je nach gewünschter Bank wird der Offset von Common 1 neu gesetzt. Bank 0 ist die normale Bank, also sind dann Interrupts erlaubt. Ansonsten werden die Interrupts gesperrt. Die Bank-Nr. wird vor dem RST 00h in A geladen. Mein kleiner IIs kennt ein Parallel-RAM. D.h. Neben Tastatur. Bildschirm etc., die ja memory-mapped-I/O sind, gibt es einen RAM-Bereich von 3400h-3fffh. Je nach Wunsch kann ich nun entweder Ein-/Ausgaben vornehmen oder in diesem Extra-RAM einige zusätzliche Programme laufen lassen. Wiederum müßte jeder TRS80-CP/M-Rechner irgendetwas in der Richtung haben.

Nun versteht Ihr auch, warum ich in der Überschrift von "ROM-Modus" sprach. Der ganze Kram klappt bei CP/M in dieser Form nicht. Dafür andere Lösungen zu suchen, überlasse ich den CP/M-Freaks.

Zu guter Letzt kann ich auch das ROM/RAM überschreiben (s. wiederum CP/M-fähige Rechner). Da sich in dem ROM allerhand Müll angesammelt hat kann man gewisse Teile ohne Skrupel killen, was ich auch getan habe.

Mit diesen ganzen Einschränkungen wird das Programm zwar sehr auf eine Maschine festgelegt, aber wer wird sich schon einen HD64180 anschaffen? Wohl doch nur Leute, die sowieso am Satteln sind und bei deren Drahtansammlungen nicht viel vom TRS80 übrig geblieben ist.

Noch kurz zum eigentlichen Programm: Der erste Teil ist nur der Ansprung der Routine von 0C00h aus. Der zweite Teil ist ein Ausgang aus der Routine, bei dem die Interrupts ausgeschaltet bleiben. Die anderen Ein/Aus-Routinen stellt OVL3/SYS aus dem Gdos 2.4 zur Verfügung. Der dritte Teil ist derjenige, der beim Aufruf die Routine ins Parallel-RAM bringt. Danach folgt die eigentliche Routine, die im gegenwärtigen Zustand noch nicht mal 100h Bytes lang ist. Sie berücksichtigt alle Illegals, die in einen Artikel der "data welt" in Mai/Juni 1986 erwähnt wurden. Die HD64180-spezifischen Befehle IN0/OUT0 wurden über DBs realisiert, weil mein alter ZEUS sie natürlich nicht kennt.

Der Programmierstil ist leider sehr "unschön", wie mein Prof sagen würde, vor allem aufgrund der vielen Patches direkt im Programmablauf, aber es tut seinen Dienst.

Gerald Schröder Hardware: Helmut Bernhardt Literatur: "TAV’s Z80 Tuning" aus "data welt" 5 und 6 ’86 Z80-Reference-Manual

```
            00001 ;     Error-Trap für Genie IIs mit HD64180
            00002 ;     Weihnachten ’86 by Gerald Schröder
            00003
            00004
      06A0  00005 pramon     EQU   06a0h       ;schaltet Par.-RAM ein
      06AB  00006 pramoff    EQU   06abh       ;schaltet Par.-RAM aus
            00007
            00008
            00009 ;
            00010
            00011 ; 0000: Ansprung Error-Trap und RST 00h
            00012
0000        00013       ORG   0
0000  CDA006     00014       CALL  pramon           ;Par.-RAM ein
0003  C30034     00015       JP    routin-offset    ;anspringen
            00016
            00017
            00018 ;
            00019
            00020 ;     neuer Ausgang: Par.-RAM aus und Interrupts
            00021 ;     bleiben disabled
            00022
06D2        00023       ORG   06d2h
06D2  DBFE  00024 prama IN    A,(0feh)
06D4  CB87  00025       RES   0,A
06D6  D3FE  00026       OUT   (0feh),A
06D8  F1    00027       POP   AF
06D9  C9    00028       RET
```

```
;

;     Initialisierungsroutine

5200              ORG   5200H
5200  3E80  start LD    A,80h      ;Common 1 ab 8000h
5202  ED          DB    0edh,39h,3ah     ;setzen (OUT0 (3ah),A)

5205  CDA006            CALL  pramon           ;Par.-RAM ein
5208  211652            LD    HL,routin  ;Routine
520B  110034            LD    DE,routin-offset ;dorthin
520E  01C900            LD    BC,ende-routin   ;übertragen
5211  EDB0        LDIR
5213  C3AB06            JP    pramoff    ;Par.-RAM aus und ab


;

;     Routine im Par.-RAM für Error-Trap und RST 00h
;     bei Error: fehlende Codes simulieren
;     bei RST 00h: Bank A einschalten

1E16        offset      EQU   $-3400h
5216  F5    routin      PUSH  AF
5217  ED          DB    0edh,38h,34h     ;IN0 A,(34h)
521A  CB7F        BIT   7,A        ;TRAP gesetzt?
521C  2010        JR    NZ,illopc  ;ja, illegaler Opcode

;******************************
;* Banking; Bank A einschalten *
;******************************

521E  F1          POP   AF
521F  F5          PUSH  AF         ;retten
5220  07          RLCA
5221  07          RLCA             ;korr. Offset erzeugen
5222  07          RLCA             ;(Basis Common 1 neu)
5223  ED          DB    0edh,39h,38h     ;OUT0 (38h),A
5226  B7          OR    A          ;Bank 0?
5227  C2D206            JP    NZ,prama   ;nein, keine Interrupts
522A  F1          POP   AF
522B  C3AB06            JP    pramoff    ;Interrupts ein



;********************************
;* illegaler Opcode: simulieren  *
;********************************

522E  CBBF  illopc      RES   7,A        ;TRAP löschen
5230  ED          DB    0edh,39h,34h     ;OUT0 (34h),A
5233  322534            LD    (ufo-offset),A   ;UFO retten
5236  F1          POP   AF         ;A vom Stack
5237  E3          EX    (SP),HL    ;HL mit Err.-Adr. tausch
5238  F5          PUSH  AF         ;A wieder retten
5239  C5          PUSH  BC         ;BC als Hilfsregister
523A  3E00        LD    A,0        ;UFO
523B        ufo   EQU   $-1
523C  CB77        BIT   6,A        ;gesetzt?
523E  2801        JR    Z,nosub    ;nein, Opc. direkt davor
5240  2B          DEC   HL
5241  2B    nosub DEC   HL         ;Zeiger auf ill. Opc.
5242  7E          LD    A,(HL)           ;laden
5243  47          LD    B,A        ;und retten
5244  FECB        CP    0cbh       ;CBxx? (SLIA)
5246  23          INC   HL         ;Zeiger weiter
5247  7E          LD    A,(HL)           ;2. Byte laden
5248  200F        JR    NZ,ddfd    ;nein, Oper. mit IX/Y

;**********************************
;* illegaler Opcode; CBxx = SLIA x *
;**********************************

```

```
524A  D620              SUB   20h         ;korrigieren
524C  323F34                  LD    (CBxx-offset),A ;in den Programm-Text

524F  C1                POP   BC          ;Register zurück
5250  F1                POP   AF
5251  23                INC   HL          ;PC neu setzen
5252  E3                EX    (SP),HL     ;und HL zurück
5253  37                SCF               ;für invertiert!!!
5254  CB10              RL    B           ;und rotieren
5255               CBxx EQU   $-1         ;10-17 für Register B-A
5256  C3AB06            ab    JP    pramoff     ;weiter im Programm


                   ;***************************************
                   ;* illegaler Opcode: erstes Byte DD/FD *
                   ;***************************************

5259  FECB         ddfd CP    0cbh        ;danach CBh?
525B  2828              JR    Z,ddfdcb    ;ja
525D  326834                  LD    (ersatz-offset),A      ;in den Text
5260  FE26              CP    26h         ;"LD hX/Y, konstante"?
5262  2807              JR    Z,ldkon          ;ja
5264  FE2E              CP    2eh         ;"LD IX/Y, konstante"?
5266  2803              JR    Z,ldkon          ;ja

5268  AF                XOR   A           ;keine Konstante; NOP
5269  1802              JR    weiter

                   ;     Konstante nach MSB/LSB von IX/Y

526B  23           ldkonINC   HL          ; Zeiger auf Konstante
526C  7E                LD    A,(HL)           ;diese laden

526D  326934            weiter     LD    (konst-offset),A       ;u. in den Text
5270  78                LD    A,B              ;DD/FD zurück
5271  326534                  LD    (ddfd1-offset),A ;setzen
5274  326B34                  LD    (ddfd2-offset),A

5277  C1                POP   BC          ;alle Register zurück
5278  F1                POP   AF
5279  23                INC   HL          ;PC neu
527A  E3                EX    (SP),HL     ;HL zurück

527B  DDE5         ddfd1PUSH  IX          ;bzw. IY
527D  E3                EX    (SP),HL     ;nach HL
527E  7F           ersatz LD  A,A         ;Operation ausführen
527F  00           konst NOP              ;evtl. Konstante
5280  E3                EX    (SP),HL     ;neues IX/Y zurück
5281  DDE1         ddfd2 POP        IX          ;bzw. IY
5283  18D1              JR    ab


                   ;*************************************
                   ;* DD/FD gefolgt von CB               *
                   ;* RLC/RRC/RL/RR/SLA/SRA/(SLIA)/SRL, *
                   ;* (IX/Y+d),B/C/D/E/H/L/A        *
                   ;*************************************

5285  78           ddfdcb     LD    A,B         ;DD/FD zurück
5286  32C034                  LD    (ddfd3-offset),A ;setzen
5289  32C534                  LD    (ddfd4-offset),A
528C  23                INC   HL          ;Zeiger auf Offset
528D  7E                LD    A,(HL)           ;"d" aus (IX/Y+d) laden
528E  32C234                  LD    (offs1-offset),A ;setzen
5291  32C734                  LD    (offs2-offset),A
5294  23                INC   HL          ;Zeiger auf Operation

                   ;     Lade-Befehl errechnen und in den Text

5295  7E                LD    A,(HL)           ;Befehl laden
5296  E607              AND   7           ;nur die Bits 0-2
5298  3C                INC   A           ;+1
5299  47                LD    B,A         ;als Zähler
529A  3E3E              LD    A,3eh       ;Opcode 46h-7eh erzeugen
```

```
529C  C608         loop1 ADD        A,8                    ;(46/4e/56/5e/66/6e/7e)
529E  10FC               DJNZ  loop1      ;=LD B/C/D/E/H/L/A, (IX..)
52A0  32C634             LD    (lade-offset),A ;setzen

             ;     Rotations-/Schiebe-Befehl erzeugen

52A3  7E                 LD    A,(HL)
52A4  E6F8               AND   0f8h       ;die Bits 0-2 löschen
52A6  0F                 RRCA             ;und raus rotieren
52A7  0F                 RRCA
52A8  0F                 RRCA
52A9  3C                 INC   A          ;+1
52AA  47                 LD    B,A        ;als Zähler
52AB  3EFE               LD    A,0feh           ;6/e/16/le/26/2e/3e=RLC,
52AD  C608         loop2 ADD        A,8                    ;RRC,RL,RR,SLA,SRA,SRL
52AF  10FC               DJNZ  loop2      ;(u. RES/SET x)

52B1  48                 LD    C,B        ;NOP erzeugen (für reta)

52B2  FE36               CP    36h        ;SLIA?
52B4  200B               JR    NZ,noSLIA  ;nein

             ;     SLIA (IX/Y+d) oder SLIA (IX/Y+d),r

52B6  0637               LD    B,37h      ;SCF erzeugen
52B8  7E                 LD    A,(HL)           ;SLIA (IX/Y+d)?
52B9  FE36               CP    36h
52BB  2002               JR    NZ,nopro   ;nein, kein Problem
52BD  0EC9               LD    C,0c9h           ;RET für reta erzeugen

52BF  3E16         nopro LD    A,16h      ;RL erzeugen (statt SLIA)

52C1  32C334       noSLIA     LD    (opera-offset),A ;setzen
52C4  78                 LD    A,B        ;SCF oder NOP
52C5  32BF34             LD    cflag-offset),A  ;setzen
52C8  79                 LD    A,C        ;RET oder NOP
52C9  32C434             LD    (reta-offset),A ;setzen

52CC  C1                 POP   BC         ;Register zurück
52CD  F1                 POP   AF
52CE  23                 INC   HL         ;PC korrigieren
52CF  E3                 EX    (SP),HL          ;und HL zurück

52D0  CDBF34             CALL  cflag-offset     ;Operation ausführen
52D3  1881               JR    ab

             ;     Unterprogramm; Operation simulieren

52D5  00           cflag NOP              ;oder SCF

52D6  00           ddfd3      NOP               ;DD o. FD für IX/Y
52D7  CB                 DB    0cbh       ;immer CBh
52D8  00           offs1 NOP              ;Offset d (IX/Y+d)
52D9  00           opera NOP              ;Operationsanweisung

52DA  00           reta  NOP              ;RETurn bei SLIA (IX/Y+d)

52DB  00           ddfd4 NOP              ;DD/FD für IX/Y
52DC  7F           lade  LD    A,A        ;Ladebefehl
52DD  00           offs2 NOP              ;Offset (s. offs1)
52DE  C9                 RET

52DF               ende  EQU   $


5200                     END   start
```

[1] hett seggt = plattdeutsch oder auch umgangssprachlicher Ausdruck für "hat gesagt"

Fehlermeldung

Im Novemberheft ist auf Seite 16 ein noch fehlerhafter Schaltplan abgebildet. Die beiden Gatter im Teil 2 des Plans mit der Bezeichnung "LS32", die aber als NOR- Gatter gezeichnet sind, sollen tatsächlich OR-Gatter (74LS32) sein.

In meinem Dreizeiler "RAM bei 3000H-37DFH anstelle des Sonder-ROMs im GENIE" habe ich folgenden Fehler zu bereuen: im vorletzten Absatz habe ich gelogen, indem ich ein 74LS00 als 74LS04 bezeichnete.

Im Hardware-Sonderheft Nov.85 ist auf Seite 10 eine ältere Version (in der noch Fehler stecken) des Auszugs aus dem Schaltplan aus dem Komtek 1 abgedruckt. Da die richtige Schaltung auf Seite 5 an der richtigen Stelle steht, bitte ich, die untere Hälfte der Seite schwarz überzustreichen.

und noch ein Statement:

von verschiedenen Seiten ist mir zugetragen worden, daß so umfangreiche Hardware- Projekte wohl niemand anpackt. Ich habe die Sachen auch nicht deshalb zu Papier gebracht, damit Peter Spieß ein bißchen mehr Arbeit mit der Auflage eines zusätzlichen Heftes hat, dessen Papier wegen bedruckter Rückseiten nicht mal als Schmierpapier taugt, sondern dafür, daß einigen interessierten Leuten zu etwas meht Komfort am Computer verholfen wird. Meine Hilfe soll auch nicht damit enden, dasß ich das Prinzipielle beschreibe. Abgebildete Layouts sollen nicht nur zeigen, wie man sich das jeweilige Board selbst ätzen kann, sondern auch ein Hinweis dafür sein, daß ich das Layout habe und danach zum Selbstkostenpreis (je nach Größe bis zu 10,-DM) weitere Boards ätzen kann. Außerdem helfe ich auch sonst gerne, wo dies technisch möglich ist.

Helmut Bernhardt

Was habt Ihr gegen CP/M ?

Jedesmal, wenn sich einer der TRS-80/GENIE-Päpste zu CP/M äußert, sträuben sich bei mir die Nackenhaare. Da kommen dann Äußerungen zu Papier, die dem unvoreingenommenen Info-Leser gleich die Vorstellung aufzwingen, daß man von CP/M lieber die Finger läßt, wenn ein Herr H. aus E. oder ein Herr S. aus L. damit nicht arbeiten wollen oder vorgeben, damit Schwierigkeiten zu haben. Die gleichen Herren bedienen sich aber gerne der CP/M-Programme, die unter NEWSOS oder GDOS nicht verfügbar sind. Und gerade die Vielzahl guter Programme, die unter CP/M laufen, ist eines der Hauptargumente für dieses Betriebssystem.

Ein weitere Pluspunkt für CP/M ist seine Hardware-Unabhängigkeit. Voraussetzung auf der Hardwareseite ist lediglich, daß als CPU ein 8080, Z80, HD64180 oder Z280 vorliegt und daß ein Umschalten zwischen ROM und RAM (ab 0000H) möglich ist. Nach dem Abschalten des Boot-ROMs muß ein zusammenhängender RAM-Speicher von 0000H an vorliegen. Wie Peripherie angeschlossen ist, spielt keine Rolle. Diese Hardware-Komponenten werden entsprechend im BDOS berücksichtigt. Das BDOS ist der Teil des Betriebssystems, der elementare I/O-Operationen mit der Hardware erledigt (z.B. Zeichen von der Tastatur holen, Zeichen auf dem Bildschirm ausgeben, Sektor von Diskette lesen oder auf Diskette schreiben u.s.w.). Da durch Anpassung im BIOS CP/M auf jeder Hardware-Kombination zum Laufen zu kriegen ist, ist es völlig egal, ob der Bildschirm memory mapped CPU-Adreßraum verschlingt oder als seriell angeschlossenes Terminal bereitsteht.

Diese Vielfalt an möglichen Computerarchitekturen ließ in der Blütezeit der 8Bit-Rechner (und auch heute noch) hervorragende Maschinen entstehen, die sich in der Hardware zwar alle unterschieden, auf denen aber alle CP/M-Programme liefen.

Die neuere Enrwicklung durch Einstieg eines führenden Großrechner-Herstellers in den Microcomputer-Markt mit einem Möchtegern-16Bit-Prozessor und einem dem CP/M ähnlichen Betriebssystem führte aber ungewollt in eine Sackgasse. Das DOS dieser Geräte sieht zwar auch die Abwicklung der I/O-Operationen über Betriebssystem-Aufrufe vor, was aber wegen der erbärmlichen Geschwindigkeit der Bildschirmausgabe nicht eingehalten worden ist. Die meisten Programme bedienen die Peripherie direkt (ähnlich wie wir das von allen besseren Programmen unter NEWDOS/GDOS kennen).

Dieses Verhalten der Software-Anbieter führte dazu, daß nur der Hardware- Hersteller sein Gerät verkaufen konnte, der es 100% kompatibel zum Vorbild baute. Wenn diese Voraussetzung nicht erfüllt war, liefen viele Programme nicht einwandfrei, derentwegen der Anwender sich das Gerät gekauft hatte. Mit solchen Geräten ist nicht viel zu verdienen. Der Begriff "Industrie- Standard" ist nur eine positive Bezeichnung dieser Misere. Dahinter verbirgt sich ein in der Hardware völlig festgefahrenes System, das mit dem TRS80 in dieser Hinsicht durchaus vergleichbar ist.

Aber zurück zum TRS80 und seinen Nachfolgern. Die fest vorgegebene und recht verbaute Hardware und die direkt darauf zugreifenden Programme ließen wenig Spielraum für Verbesserungen, ohne daß das Gerät die TRS80-Kompatibilität einbüßt. Um CP/M fahren zu können, war es nötig, daß Floppy, Drucker, Tastatur und Bildwiederholspeicher nicht mitten im Adreßraum der CPU liegen. Für NEWDOS/80 und GDOS war das aber zwingend. Es wurden Banker entwickelt, die zwischen ROM und memory mapped I/O und RAM umschalten oder nur zwischen ROM und RAM umschalten und memory mapped I/0 an das obere Speicherende verschieben. Mit teilweise riesigem Aufwand wurde die gleichzeitige TRS8O-Kompatibilität und CP/M-Fähigkeit erreicht. In einem Testbericht der CHIP zum GIIIs liest sich das so: "Mußte es denn ausgerechnet eine 8Bit-Maschine sein, in die soviel Arbeit und Ideenreichtum gesteckt wurde ?"

Dabei kann die Architektur eines CP/M-Rechners sehr einfach ausfallen, wenn man auf die TRS-80-Kompatibilität verzichtet (siehe mc-CP/M-Computer). Es ist also nicht verwunderlich, daß bei dem dafür nötigen Aufwand und den damit verbundenen Einschränkungen heute niemand mehr TRS-80-kompatible Computer anbietet.

Aber nicht nur die geringen Anforderungen an die Hardware und die damit verbundene Gerätevielfalt und das riesige Software-Angebot sprechen für CP/M. Auch das Programmieren in Maschinensprache ist unter CP/M sehr viel einfacher. Alle Betriebssystem-Aufrufe und Hardware-Zugriffe erfolgen über CALL 0005H, wobei im Register C immer die gewünschte Funktion ausgegeben wird und in anderen Registern je nach Funktion die Parameter übergeben/übernommen werden. Der Programmierer braucht sich über die Hardware-Architektur überhaupt keibe

Gedanken zu machen.

Daß nun ausgerechnet unsere Software-Päpste um CP/M einen großen Bogen machen, liegt wahrscheinlich daran, daß sie die enorme Arbeit, die investiert werden muß, um mit NEWDOS/GDOS umgehen zu können, nicht umsonst geleistet haben wollen, und auch nicht bei CP/M noch einmal von vorne anfangen wollen. Es kostet tatsächlich einige Überwindung, anstelle von ZEUS nun ein Maschinenprogramm mit Wordstar zu schreiben und mit einem der zahlreichen Assembler zu übersetzen. Man muß gleich mehrere Handbücher wälzen.

Etwas fadenscheinig kommen mir die Argumente wie miese Leistungsfähigkeit und mangelnder Bedienugskomfort vor. Zugegeben, es ist schon etwas lästig, wenn man sich immer erst erinnern muß, in welchen Drive nun die Diskette steckt, auf der die gewünschte Datei vorliegt. NRWDOS sucht mir das selbst heraus. Andererseits ärgert mich bei NEWDOS das dort übliche DIR-Chaos. Ein neu auf die Diskette geschriebener File verschwindet irgendwo mittenzwischen den schon auf der Scheibe vorhandenen Files. Dagegen wird von CP/M das Directory so ausgegeben, wie die Einträge nacheinander erfolgt sind. Und dann gibt es da noch diese herrlichen Wildcards, mit denen eine selektive Anzeige des Directory möglich ist.

Die vorgeworfene geringe Leistungsfähigkeit von CP/M ist eher als Vorteil zu sehen. Dadurch, daß CP/M nur auf seine Betriebssystem-Funktionen beschränkt ist, nimmt es wenig Speicherplatz weg und braucht auch nicht mit Overlay Technik zu arbeiten. Alle über die Betriebssystem-Funktionen hinausgehenden Leistungen müssen entsprechende Programme liefern, und dafür gibt es sehr gute Utilities. Es ist mir völlig gleichgültig, ob ein Disketten-Kopieren mit einem Betriebssystem-Befehl oder mit einem Programm durchgeführt wird, ausschlaggebend ist der Komfort bei der ganzen Aktion.

Und schließlich habe ich als CP/M-User Anschluß an all die vielen Z80-Rechner, die in den letzten 10 Jahren auf den Markt gebracht wurden. Was bei mir läuft, ist auch dort verwendbar und umgekehrt.

Prüfzahlfehler beim Lesen

Diese Horrormeldung stelle man sich vor, wenn man nach dem Überarbeiten eines 3 Files umfassenden Assemblerprogramms, das beim Assemblieren 7 KByte Objektcode ergab, einen der Source-Files laden will. Wenn sich dieses Erlebnis dann auch noch bei den anderen beiden Files einstellt, reicht das dann erstmal für den Rest des Tages. Wer etwas härter im Nehmen ist, macht dann noch einige zaghafte Versuche mit SUPERZAP, das bei etwas Glück mit seiner Vermutung, daß einige oder alle Daten des defekten Sektors im Puffer enthalten sind, Recht hat. Leider weigert es sich, diese nach eventuell nötigen Korrekturen wieder auf die Diskette zurückzuschreiben. Der Umfang des Schadens läßt sich zwar feststellen aber nicht beheben.

So ganz wollte ich mich aber nicht mit der Vorstellung anfreunden, 32 DIN A4 Seiten gut kommentierten Programms neu einzutippen. Also erst mal versuchen, ob sich der Rest des Files auf eine Diskette kopieren läßt. Wenn dann da einige Sektoren mit E5H gefüllt sind, würde ich da mit SUPERZAP schon was reinbringen, was ich hinterher mit ZEUS wieder in sinnvollen Text verwandeln könnte. Hauptsache, es tritt kein CRC-Fehler mehr auf. Beim Kopieren mit COPY,0,1 traten dann auch die erwarteten Meldungen auf, die ich mit einem (F)ortfahren beantwortet habe. Insgesamt 5 Sektoren hat das DOS beanstandet. Ich sollte offensichtlich 1,25 KByte von E5H in von ZEUS verwertbares Material umzappen. Na gut, an die Arbeit, mit SUPERZAP den ersten beim Kopieren beanstandeten diskrelativen Sektor geholt und - eitel Freude - es war keine einzige E5H zu sehen sondern mein Originaltext, der ohne CRC-Fehler geladen wurde.

Von den anderen 4 Sektoren war dann noch einer geringfügig zu korrigieren und nach einer halben Stunde waren meine Source-Files wieder heil. Das mag für manchen von Euch der Schnee von gestern sein. Wenn man aber unter der Vorstellung lebt, bei solchen Katastrophen nur das Handtuch werfen zu können, dann mag diese Erfahrung recht hilfreich sein. Trotzdem hoffe ich, daß möglichst wenige von Euch darauf zurückgreifen müssen.

Helmut Bernhardt

1989 Club80-10 Nr.28 S.72

Star NL10 abwechselnd am CP/M-Rechner und am PC

Helmut Bernhardt

Eigentlich sollte es dem Drucker doch egal sein, ob jenseits des Centronics-Eingangs ein CP/M-Rechner oder ein Big Blue Nachbau Ursache für zu beschmutzendes Papier ist. Das ist es ihm auch und dadurch hat man die Schwierigkeit mit den deutschen Umlauten. Wenn der Drucker nicht der gleichen Kaste angehört wie der Computer. Der Drucker geht davon aus, daß es den ASCII-Code gibt und der für ihn genauso verbindlich wie für alle anderen ist. Schließlich ist der ASCII-Code ja ein allgemeiner Standard.

Das war vor einigen Jahren mal ein einheitlicher Standard. Damals genügten 7 Bit, um damit 128 Zeichen zu definieren. Mit 128 Zeichen waren alle amerikanischen Zeichen und eine Reihe Steuerzeichen zu codieren. Um der Vielfalt der nationalen Sonderzeichen in der übrigen Welt Rechnung zu tragen, hatten Drucker ein Mäuseklavier, auf dem man beispielsweide einstellen konnte, daß anstelle der amerikanischen eckigen und geschweiften Klammern unter den gleichen Codes (5BH-5FH, 7BH-7FH) dann die entspechenden deutschen Umlaute zu Papier gebracht werden. Andere Einstellungen machten den Drucker auch für Engländer, Franzosen, Spanier und noch manch anderes Völkchen nutzbar.

Bei der Einführung des PC hat der Urheber desselben einen neuen 8 Bit ASCII-Code zum Standard erhoben. Der IBM Zeichensatz Nr. 2 wird heute von allen Druckern und natürlich auch von allen PC-Videokarten unterstützt. Diesem Standard haben sich auch andere Computer (z.B. Atari ST) angeschlossen. Der 8 Bit ASCII-Code besteht zur Hälfte (Codes 00H-7FH) aus dem alten 7 Bit ASCII-Code. Sämtliche von IBM als berücksichtigenswert erachteten nationalen Sonderzeichen wurden Codes mit gestztem Bit 7 zugeteilt.

Da Big Blue aber auch so manchen Unsinn wie die Spielkarten-Farben, eine Teilmenge der TRS80-Grafik-Bauklötzchen und eine Menge von Linie-Grafikzeichen (für die Darstellung der furchtbar wichtigen Rahmen bei Pulldown- und Popup-Menüs) als wichtig erachtet hat, sind natürlich etliche Zeichen unter den Tisch gefallen. Anstelle des deutschen "Eßzett" muß man halt ein griechisches "Beta" nehmen.

Mein jüngst erworbener Drucker Star NL10 hatte das Glück, in einer Zeit herausgekommen zu sein, als nicht nur PCs sondern auch noch CP/M-Computer eine Rolle spielten. Entsprechend war Star so umsichtig, den Drucker mit verschiedenen Betriebsprogrammen und Zeichensätzen (im 27256-EPROM) herauszubringen.

Wenn man nun einen NL10 mit PC-kompatiblem Betriebsprogramm-EPROM hat und jemanden kennt, der den gleichen Drucker mit dem CP/M-kompatiblen Betriebsprogramm hat, und jemanden kennt, der 27256-EPROMs lesen und 27512- EPROMs brennen kann (falls man das nicht selbst kann), dann kann man sich die Inhalte der beiden 27256-EPROMs in ein 27512-EPROM füllen (lassen) und über A15 des EPROMs (Pin 1) durch Anlegen von +5V oder GND das jeweilige Betriebsprogramm für den jeweiligen Rechner einstellen. Das Umschalten darf natürlich nicht bei eingeschaltetem Drucker erfolgen. Wegen des sonst beim Absturz des Druckers erzeugten Lärms hat man sich aber schnell an diese Maßnahme gewöhnt.

Die Clubeigene Grafikkarte

Andreas Magnus

Nachdem im vorletzten Info mein Artikel erschienen ist, habe ich mit einer Flut von Zuschriften gerechnet. Dem war leider nicht so. Dem konnte ich entnehmen, daß Ihr alle sehr schreibfaul seid, oder kein Interesse an Projekten habt. Ich will hoffen, daß es der erste Grund ist. Aber immerhin habe ich ja zwei Zuschriften bekommen. Aber vielleicht waren Euch ja die Angaben zu dürftig. Jetzt sieht es anders aus. Auf dem Clubtreffen habe ich mich mit Helmut über eine Grafikkarte unterhalten. Wir sind zu dem Entschluß gekommen, es so einfach wie möglich zu machen und vor allem das Rad nicht neu zu erfinden. Man kann nicht widerlegen, daß auch IBM seine Vorteile hat, auch wenn es nur der ist, dafür gesorgt zu haben, 'billige' Grafikkarten auf den Markt zu schmeißen. Und diesen Vorteil wollen wir nutzen.

Aber nun im einzelnen. Wir haben beschlossen, einen Grafikadapter zu entwickeln. Dieser hat die Aufgabe, eine Herkules- oder EGA-Karte über Ports an den Rechner anzuschließen. Der Adapter wird 'intelligent', d.h. er bekommt eine eigene CPU. Alles, was zur Ansteuerung der Grafikkarte notwendig ist, erledigt ein Z80, genauso wie die Kommunikation mit dem Host-Rechner.

Die Kommunikation mit der Karte erfolgt über zwei Ports, einer lesend, der andere schreibend, die mit ein paar Bauteilen leicht in jeden Rechner einzubauen sind. Über diese Ports werden Daten und Befehle ausgetauscht. Die Adapterkarte enthält eine CPU, 32Kb RAM, 32Kb ROM sowie die Anschaltung der Grafikkarte. Es wird also nicht zu viel. Für ganz eilige: Die genaue Schaltung ist noch nicht fertig, weil Helmut und ich noch ein genaues Konzept ausarbeiten müssen, aber das ist jetzt auch noch nicht ganz so schlimm, weil noch ein paar andere Dinge festzulegen sind, die aber die Software angehen. Wer aber trotzdem mit Rat und Tat zur Seite stehen will, wende sich bitte an Helmut oder mich, auch was die Zoffware angeht.

Aber nun zur Karte selber. Wir haben uns zwar schon ein paar Gedanken gemacht, was das Teil können muß, aber wie schon gesagt, es ist noch in der Entwicklung, und außerdem sollt Ihr Eure Meinung auch dazu beitragen. Die Liste erhebt keinen Anspruch auf Vollständigkeit, sondern ist ein erster Überblick. Wenn Ihr eigene Ideen habt, laßt sie mich wissen, auch wenn Ihr nicht wißt, wie sie zu bewerkstelligen sind, denn darüber machen wir uns dann später Gedanken.

- Programm Download (Eigene Programme, die die Karte erledigt)
- Punkt setzen
- Punkt löschen
- Punkt testen
- Linie ziehen mit Farbangabe
- Viereck zeichnen mit Farbangabe
- Fläche ausfüllen mit Farbangabe
- Kreis zeichnen
- Schirm invertieren
- Bereich invertieren
- Fläche invertieren
- Bereichsfarbe ändern
- Bereich spiegeln
- Bereich drehen
- Bereich verschieben
- Objekt verschieben
- ASCII anzeigen
- Zeichensatz laden
- Terminalfunktion
- Letzten Befehl wiederholen
- Letzten Befehl zurücknehmen
- HRG auslesen (zum Abspeichern)
- HRG einschreiben (von Disk)
- HRG mischen (MERGE)

- beliebigen Cursor erzeugen
- Cursor bewegen (plazieren)
- ....

Das ist das, was mir auf die Schnelle so einfiel. Ich will es erstmal dabei belassen, um mir über den genauen Aufbau gedanken zu machen. Vielleicht gibt es im nächsten INFO schon Schaltungen oder Programme dazu. Aber wie gesagt, Eure Mitarbeit und Eure Anregungen sind ausdrücklich erwünscht.

Bis bald, Euer Hardware-Koordinator.

Megabit-RAMs einfach erfrischt

Helmut Bernhardt

Im letzten Info hat Alexander eine aufwendige Erweiterung der Erweiterung der 7Bit-Refreshadresse des Z80 auf 8Bit beim 256K-Banker und nun 9 Bit für Megabit-RAMs vorgestellt. Einerseits wäre ich ziemlich stolz gewesen, sowas selbt ersonnen zu haben, andererseits bin ich aber viel zu faul, die bisherige aus der c't übernommene Schaltung zu durchblicken und dann noch auf 9 Bit zu erweitern - es geht nämlich auch ohne.

Wie aus Datenblättern und den beiden üppigen Knowledge-Anhäufungen zu DRAMs in den Heften 10 und 11 aus c't 1990 von Johannes Assenbaum zu entnehmen ist, gibt es für alle 1M-Bit-DRAMs die Möglichkeit, das Refreshing durch den /CAS-before-/RAS-Refresh zu erledigen. Bei 256K-Bit- -DRAMs wird das nur bei einigen Herstellern unterstützt - hier sollte man in entsprechenden Schaltungen diese Technik nicht anwenden, weil nicht gesichert ist, ob die darin eingesetzten Chips den Sport mitmachen.

Beim /CAS-before-/RAS-Refresh muß nur ein low aktives /CAS an die RAMs gelegt werden, 'before' ein /RAS kommt. Dann wird im RAM mit einem internen Zähler eine Columnadresse erzeugt und die damit adressierten Speicherzellen werden refreshed. Ein externer Memorycontroller (bei uns z.B. der Z80) braucht sich dann nicht mehr um eine von außen zugeführte Refreshadresse zu kümmern. Damit ist es auch nicht mehr nötig, daß in der Breite der von den RAMs verwendeten Refreshadresse die nierigen Adressen der CPU zusammen beim /RAS durch die Adreßmultiplexr durchgeschaltet sein müssen. Die CPU-Adressen können in wilder Folge an die A- und B-Eingänge der Multiplexer gelegt werden - wie es für das Layout gerade am besten kommt.

Und wie erzeuge ich einen /CAS-before-/RAS-Refresh? Ich leite wie immer ein aus der Adreßdecodierung für die RAM-Bank und Timing (relativ zu /RAS und MUX) zusammengesetztes /CAS her und lege es über ein AND-Gatter mit /RFSH bzw. /RFS beim HD64180 zusammen und verabreiche den RAMs das Produkt dieser geringen Mühe an den /CAS-Eingängen.

Beim Refresh der CPU erscheint das /RFSH-Signal sehr viel früher als das als /RAS verwendete /MERQ. Damit ist die Bedingung der RAMS für den /CAS-before-/RAS-Reresh erfüllt. Bei normalen RAM-Zugriffen wird das herkömmliche /CAS, das zeitlich versetzt nach /RAS und MUX aktiv wird, zu den RAMs durchgeschaltet.

Wer mehr dazu wissen möchte, kann an anderer Stelle in diesem Info eine Anwendung in Form einer 2MB-RAM-Floppy finden, wo diese Refreshart den Schaltungsaufwand wesentlich reduziert hat.

Leserbrief zu

Newdos, CPM, MS-DOS, UNIX

von Gerald Schröder
in1990-Club80-32

Gerald, grundsätzlich sprichst Du mir aus der Seele. Ähnlich ketzerische Äußerungen zu MS-DOS und PC und ihre Verwandtschaft zu NEWDOS und Trash-80 habe ich - glaube ich - vor längerer Zeit auch schon mal im Info verbreitet. Dazu stehe ich auch heute voll und ganz. Hier will ich aber mal versuchen, die Sicht des Anwenders dagegen zu stellen und ein paar positive Seiten der beiden Anachronismen hervorheben.

Gehen wir aber zuerst mal gute 10 Jahre zurück: Damals hat ein kleiner Bauchladen Namens Radio Shack einen Billigst-Computer herausgebracht und in den USA über die Tandy-Kette verkauft. Wegen des niedrigen Preises war das Gerät ein Riesenerfolg und wurde in großen Stückzahlen verkauft. Daraufhin schossen etliche Software-Firmen aus dem Boden und brachten gute Betriebs-Systeme und gute Programme auf den Markt. Dieses für damalige Verhältnisse riesige Software-Angebot hätte es bestimmt nicht gegeben, wenn der TRS-80 nicht aufgrund seines niedrigen Preises eine solche Verbreitung gefunden hätte und dadurch für die Softfirmen ein entsprechender Markt bestanden hätte.

Big Blue's PC hat sich zunächst natürlich nicht des günstigen Preises (der kam erst sehr viel später durch Taiwan-Konkurrenz) sonder wegen der Machthörigkeit der ganzen Branche (auch der Fachpresse) verbreitet, obwohl die Erfinder des Gerätes sich alle Mühe gegeben hatten, ein möglichst primitives System zu schaffen, das man durchaus billiger verkaufen könnte. Das Vorhandensein von DMA-Controller und Timer war nicht etwa eine großzügige Ausstattung zugunsten von mehr Performance sondern für den Refresh der dynamischen RAMs nötig, weil die CPU das nicht wie ein Z80 selbst erledigt. An diesem Murks orientieren sich auch heute noch alle Neuschöpfungen des Industrie-Standards, so daß Software für den PC-XT auch noch auf einem EISA-486er-AT läuft.

Bei diesem Softwaremarkt stürzten sich alle Softfirmen auf den großen Kuchen und niemand hatte mehr etwas für andere Systeme zu bieten. Die Fülle der Anwenderprogramme führte dazu, daß die Programme nicht mehr nur die eigentlich bezweckten Funktionen erfüllen mußten, um gegenüber der Konkurrenz bestehen zu können; sie mußten auch durch originelle äußerliche Gestaltung das Interesse des Käufers auf sich lenken. Die 'Benutzeroberfläche' war erfunden.

Sowas läßt sich aber nur programmieren, wenn man die volle Kontrolle des Bildschirms hat. Ein unter CP/M übliches Terminal ist dagegen ein Datengrab, das kaum Kontrolle darüber zuläßt, was wo im Bildschirm steht. So erfolgen die Bildschirmausgaben bei den meisten Programmen unter CP/M zeilenweise. Wer unter CP/M mit DU.COM gearbeitet hat, sehnt sich nach SUPERZAP von NEWDOS/80 oder DEBUG zurück, wo das alles schon sehr viel eleganter durch Bewegen des Cursors über den ganzen Bildschirm möglich war und das auch nur wegen der festgelegten Hardware und des memory mapped Bildwiederhol-Speiches, den die CPU selbst verwalten muß bzw. manipulieren kann. Fenstertechnik ist nur möglich, wenn Bereiche des Bildwiederhol-Speiches ausgelagert werden können, und nach Schließen des Fensters wieder restauriert werden können.

Programme unter CP/M, die ansatzweise diesen Komfort zu bieten haben, müssen installiert werden. Die dabei durchzustehenden Fragen bringen auch einen alten Hasen gelegentlich aus dem Tritt, wenn er nach dem ersten Durchgang merkt, daß seine Annahme, er müsse die Daten in HEX eingeben, falsch war, und er nochmal wieder die entsprechenden Informationen zu seinem Terminal im Handbuch nachschlagen muß.

Auf dem PC bedeutet installieren im schlimmsten Fall, der Software bekanntzugeben, was für eine der 3 denkbaren Videokarten denn nun im Gerät steckt; das kriegt noch jeder hin, weil er sich beim Kauf des PCs wegen der sehr unterschiedlichen Preise darüber Gedanken gemacht haben muß. Meistens bemerkt das Installtions Progamm die Existenz der jeweiligen Videokarte aber selbst und belästigt den Anwender garnicht erst mit entsprechenden Fragen.

Ähnlich verhält es sich mit der Tastatur: Wie beim TRS-80 erhält die CPU nur Informationen von der Tastatur, welche Taste gedrückt bzw. losgelassen wird. In beiden Fällen muß die CPU sich aus der Identität der gedrückten Tasten selbst ein ASCCI-Zeichen basteln oder sich sonst einen Reim daraus machen. Und da man mit TSR-

Programmen beim PC selbst in die Auswertung der Tastatur- Scancodes eingreifen kann, ist ein kleiner Geschmack von Oligo-Tasking möglich ("Multi-Tasking" wäre etwas zu hoch gestochen.

Ein weiterer Schachzug der Softwarefirmen war, in die Programme eine jederzeit zugängliche Online-Hilfe einzubauen, die ein ständiges Blättern im Handbuch (und schließlich das Handbuch selbst) unnötig machte. Jeder konnte recht schnell mit einem Programm umgehen. Natürlich kostet das Bereitstellen von Hilfstexten und auch die üppige Austattung der Programme mit Menüs Speicherplatz.

Dem Anwender sind Kriterien wie Portabilität und Hardware- Unabhängigkeit ziemlich schnuppe; er will möglichst schnell sein Programm beherrschen können und vom Programm eine geschmackvolle Bildschirmgestaltung geboten bekommen. Er kann ohnehin nicht beurteilen, mit welchem Aufwand an Gehirnschmalz die einzelnen Funktionen des Programms realisiert worden sind. Wenn er ein neues Programm vor sich hat, dort seine gewohnte SAA-Oberfläche wiederfindet und auch gleich ein paar minimale Funktionen bewirken kann, ist das Erfolgerlebnis größer, als wenn er nach vielen Schiffbrüchen endlich in der Lage ist, durch kryptische Befehle gewisse Funktionen auszulösen.

Helmut Bernhardt

Nachtrag zum Banker

Alexander Schmid

Sollte es wahr sein, daß bis jetzt außer Arnulf Sopp und mir noch niemand Probleme mit dem Banker von Helmut Bernhardt hatte? Keine unmotivierten Abstürze oder Aufhänger? Na gut. Vielleicht hängt das Auftreten auch etwas mit der Taktgeschindigkeit zusammen und die ist bei uns halt etwas höher. Trotzdem hat die Schaltung einen echten Fehler, den ich hier mit freundlicher Genehmigung von Helmut breittreten darf.

Nach über einem Jahr Suchen und Ärgern habe ich ihn endlich gefunden. Der Käfer besteht darin, daß am 74LS74 auf dem Banker die Pins 11 und 12 vertauscht sind, was auf einen Druckfehler in einer alten c't zurückgeht, aus der Helmut einen Teil der Schaltung übernommen hat. Da vergleicht man immer wieder das Layout mit dem Schaltplan, wundert sich, daß alles stimmt und trotzdem nichts geht und dann sowas. Diese Verwechselung hat die unangenehme Folge, daß der Refresh für die dynamischen RAMs nicht richtig funktioniert und die Bits unter Umständen nach Lust und Laune umkippen. Beim TRS80/ Genie unter NEWDOS und GDOS macht sich das im Normalfall nicht bemerkbar, weil der Prozessor dort durch die Interrupts kreuz und quer durch's RAM springt und so schon alleine durch die Speicherzugriffe während der Interruptroutinen für eine ausreichende Erfrischung sorgt. Wenn die Interrupts aber abgeschaltet sind, was z.B. bei mir unter CP/M der Fall war, kann es Ärger geben, wenn z.B. eine Schleife zu kurz wird.

Klarheit kann man sich mit folgendem Programm schaffen:

```
      ORG   6000h      ;oder sonstwo

start DI               ;keinen Mogelrefresh machen
      LD    A,8        ;Zähler
      LD    B,A
      LD    C,A
Loop  INC   C
      JR    NZ,loop          ;innere Schleife
      INC   B
      JR    NZ,loop
      INC   A
      JR    NZ,loop          ;äußere Schleife
      JP    4400h      ;ins DOS
      END start
```

Durch die kurzen Schleifen bleibt der Speicherzugriff auf wenige Adressen beschränkt und der ganze Rest kann nur durch die Logik auf dem Banker behandelt werden. Wenn sich der Banker nach 1-2 Wochen noch nicht wieder gemeldet hat, könnt Ihr davon ausgehen, daß im restlichen RAM nur noch Schrott steht, womit sich auch das DOS verabschiedet hat. Nach dem Neubooten dürfte dann überall nur noch wirres Zeug stehen, was Ihr mit Debug o.ä. überprüfen könnt. Wenn alles stimmt, sollte nach ungefähr zwei Minuten wieder das DOS-Ready erscheinen und der Rest des Speicherinhalts unverändert sein. Wem das zu lange dauert, der kann A mit einem größeren Wert laden, wobei es aber auch wieder nicht zu schnell gehen darf, weil die RAMs, wenn sie einem guten Tag haben, durchaus ein paar Sekunden stabil sein können. Die paar Minuten sollte das doch wert sein. Zur Überprüfung könnt Ihr vorher noch irgendein Programm in die RAMdisk kopieren, die ja denselben Speicher benutzt. Wenn das Programm nachher noch läuft, ist mit ziemlicher Sicherheit alles in Ordnung. Wenn nicht, solltet Ihr schnellstens den Lötkolben heizen.

192 K RAM im Genie I und II

H. Bernhardt, C. Ueberschaar

Einer der wesentlichen Nachteile von 8 Bit-Mikroprozessoren gegenüber 16 Bit-Prozessoren ist die Beschränkung auf nur 16 Adreßleitungen. Der Prozessor kann damit 2^{16} = 65536 = 64K Speicherstellen von einander unterscheiden. Von diesen 64K werden beim Genie 16 K für ROM, Sonder-ROM, Video-RAM, Tastatur und memory mapped decodierte Peripheriegeräte benutzt. Bei angeschlossener Floppy Disk und geladenem DOS und Disk-BASIC werden je nach DOS nochmals annähernd 16K RAM belegt. Die verbleibenden 32K Speicherraum reichen für die meisten Anwendungen aus.

Wenn allerdings Aufgaben bewältigt werden sollen, bei denen große Datenmengen schnell gesammelt werden müssen (z.B. Messungen), miteinader verrechnet und z.B. in Form von Graphiken ausgegeben werden müssen, dann werden 32 K schnell zu knapp.

Den prinzipiellen Weg zu mehr Speicher beim Genie I und II hat der Hersteller bei der Änderung von 16K RAM auf 48K RAM im Grundgerät (83er Modell) aufgezeigt. Durch Ersetzen der 4116 (16 K x 1 Bit) RAM-Chips durch 4164 (64 K x 1 Bit) RAM-Chips und einige dafür nötige Änderungen, ist der gesamte vom Z80-Prozessor adressierbare Speicherbereich bereits im Grundgerät auf dem CPU-Board realisiert. Die Steckplätze für 32K RAM im Expansionsinterface werden nicht mehr gebraucht.

Was auf dem CPU-Board möglich ist, ist aber auch im Expansionsinterface durchzuführen, indem 4116er Chips durch 4164er RAMs ersetzt werden.

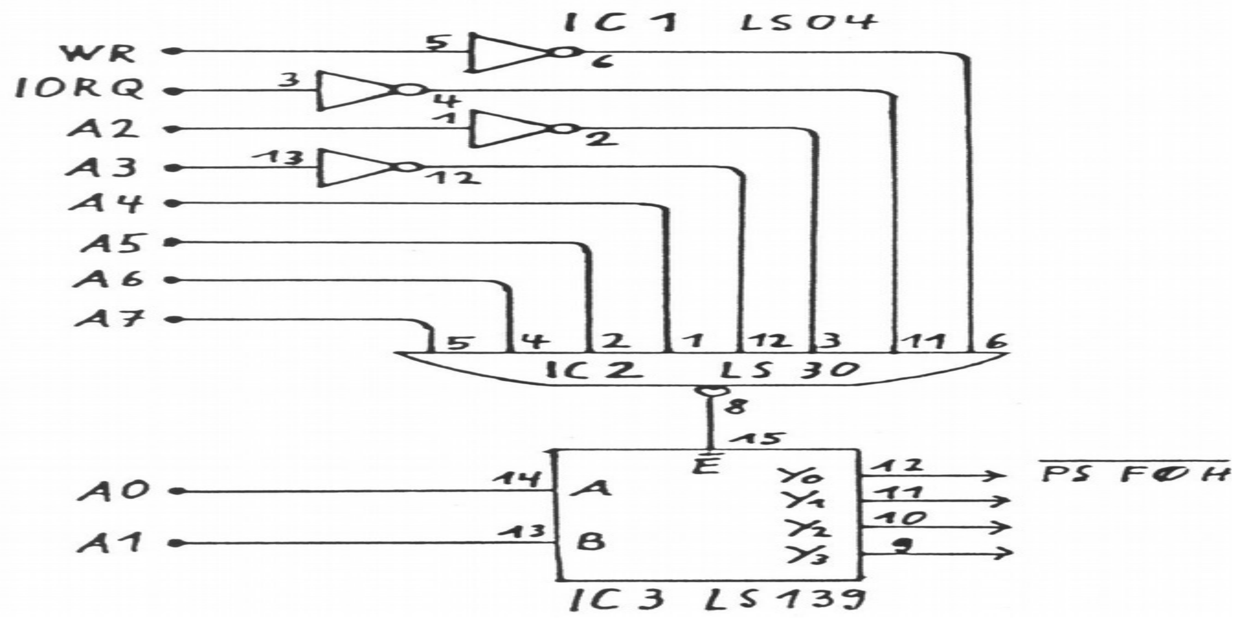
Die dadurch vorhandenen 192K RAM lassen sich allerdings nicht durch Umschalten zwischen den drei 64K-Blöcken benutzen. Beim Umschalten vom Hauptblock in einen anderen Block wären Betriebssystem und Stack nicht mehr verfügbar. Das Umschalten darf nur einen kleinen Teil im oberen Adreßraum umfassen.

In der im Folgenden beschriebenen Lösung wurden dafür die oberen 4K (oder 8K; durch Software umschaltbar) also der Adreßraum F000H bis FFFFH (bzw. E000H bis FFFFH beim 8K-Banking) gewählt. Das bedeutet, daß der gesamte Adreßraum von 64K in 60K (56K) Grundspeicher und einen 4K (8K)-Bereich, der durch Software steuerbar überall in den zusätzlichen Speicher hineingelegt werden kann, unterteilt ist.

Die Schaltung

Die softwaregesteuerte Umschaltung der 4K (8K)-Blöcke, die bei einer Ausgabe einer Adresse in diesem Speicherbereich jeweils angewählt werden, erfolgt über ein 8Bit-Latch mit ENABLE und CLEAR 74116, das durch die Portadresse F0H decodiert ist (Abb.1). Durch einen OUT-Befehl läßt sich in das Latch ein 8Bit-Wort eintragen, das dann ständig an den Ausgängen anliegt und zur Selektion eines von maximal 256 4K(8K)-Blöcken herangezogen werden kann. Der Eingang CLEAR wird an die RESET-Leitung des Systems angeschlossen und gewährleitet dadurch, daß beim Einschalten des Systems immer alle Bits des Latch auf Low gesetzt sind.

Abb. 1: Decodieren der Portadresse F0H (F1H, F2H, F3H)



Das 8Bit-Wort im Latch wird benutzt, um einerseits einen der 3 im Gerät vorhandenen 64K-Blöcke anzuwählen und um andererseits im angewählten Block einen bestimmten 4K (8K)-Bereich zu decodieren.

Die niederwertigen 4(3) Bits im Latch werden anstelle der Adreßleitungen A12 (A13) bis A15 an die Adreßmultiplexer der RAMs gelegt und adressieren damit einen 4K(8K)-Bereich im angewählten 64K-Block.

Die höherwertigen 4(5) Bits werden zur Aus-wahl eines von maximal 16(32) Blöcken mit je 64K benutzt. Auf diese Art lassen sich mit dem 8Bit-Wort im Latch insgesamt 1M Byte (2M Byte) Speicher verwalten. Hier wird aber nur von drei 64K-Blöcken Gebrauch gemacht, weil für diese nach Ersetzen der 4116er RAMs durch 4164er RAMs und einige geringfügige Ände-rungen die nötigen Steckplätze im System vorhanden sind. Die Schaltung liefert aber die Auswahlimpulse für insgesamt 32 solcher 64K- Blöcke. Mit geeigneten Speicherkarten läßt sich das System auf 2M Byte Speicher erweitern.

Der Blockauswahlimpuls, der gewonnen wird, wenn alle 4(5) höherwertigen Bits des Latch 0 sind und der Prozessor eine Adresse oberhalb F000H(E000H) ausgibt, muß den RAM-Block auf dem CPU-Board anwählen. Dieser Block muß aber auch angewählt werden, wenn eine Adresse unterhalb F000H(E000H) ausgegeben wird. Die Schaltung muß also erkennen, ob eine Adresse unterhalb des umschaltbaren Bereichs anliegt, und dann die normalen Adreßleitungen A12 (A13) bis A15 an die Adreßmultipexer der RAMs durchschalten, oder ob eine Adresse im umschaltbaren Bereich anliegt, und dann die niederwertigen 4(3) Bits des Latch als Adresse an die Multiplexer der RAMs anlegen.

Diese Aufgabe wird durch ein 4 Input-NAND 74LS 20(IC10), dessen Ausgang immer dann low ist, wenn A12 (A13) bis A15 alle gleichzeitig high sind, und einen Multiplexer 74LS157

(IC15), dessen SELECT Eingang je nach Zustand des NAND entweder die unteren Bits des Latch oder die System-Adreßleitungen A12 bis A15 durchschaltet.

Wenn die an das NAND-Gatter angelegten Adreßleitungen alle high sind (also der obere, umschaltbare Adreßbereich angewählt wird, dann werden vom Multiplexer die Ausgänge des Latch durchgeschaltet, wenn auch nur eine dieser Adreßleitungen low ist, werden die normalen Adreßleitungen vom Multiplexer durchgeschaltet (Abb. 2).

Die Umschaltung zwischen Banking von 4K und von 8K erfolgt über das Bit 0 eines anderen Latch. Ein zweiter Baustein 74116 wird dazu in zwei 4Bit-Latches unterteilt. Beide Hälften werden durch unterschiedliche Portadressen Pins 11 und 10 von IC3) decodiert. Die Hälfte, die durch die Portadresse F2H decodiert wird, dient als Software-Schalter. Der Bit0-Ausgang wird an die SELECT-Eingänge von 3 weiteren 74LS157 Multiplexern gelegt und kann so die Durchschaltung von je einem von 2 Eingängen zum Ausgang des Multiplexers steuern.

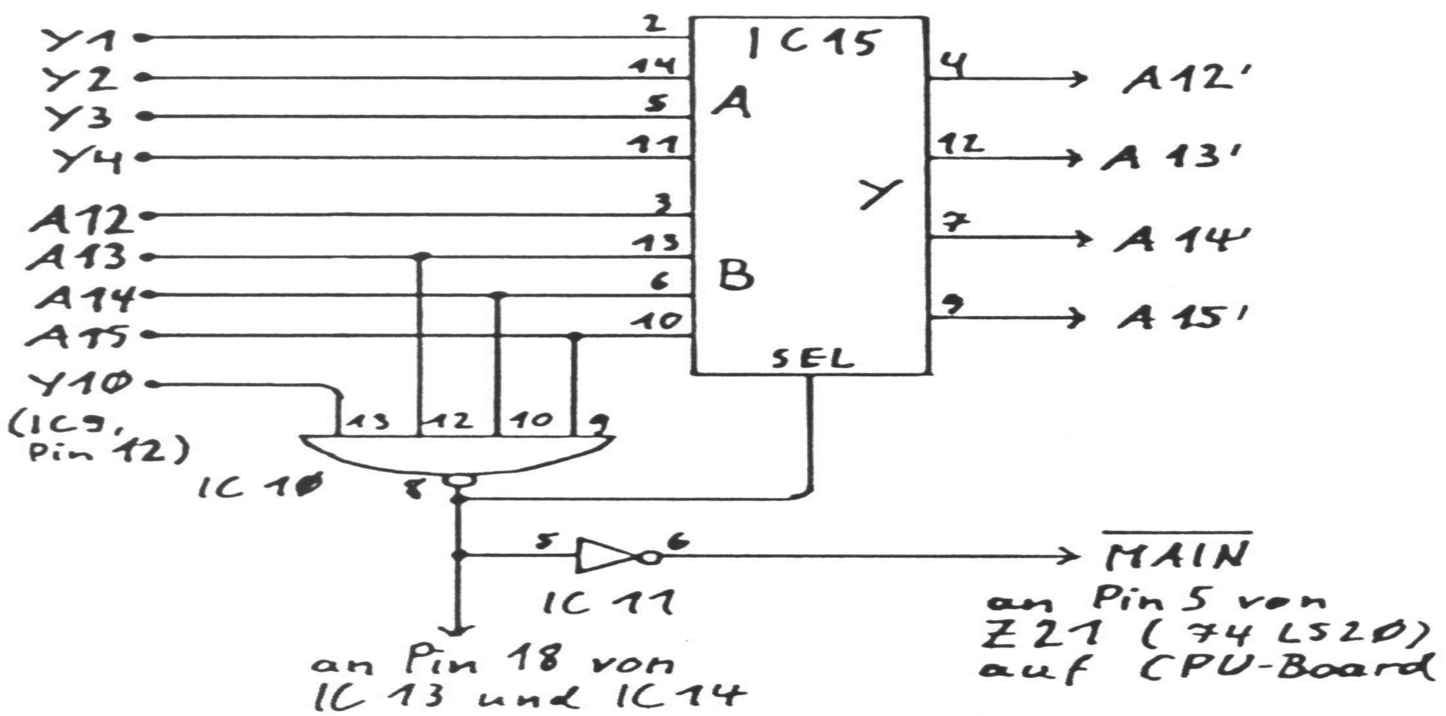


Abb. 2: Multiplexern der Adressen A12-A15 und der unteren Bits des Latch. Wenn A12 (A13) bis A15 alle gleichzeitig high sind, werden die unteren Bits des Latch durch-geschaltet.

An den A-Eingängen der beiden Multiplexer IC7 und IC8 liegen die 8 Bit des Latch F0H an. Diese 8 Bit werden dann direkt durchgeschaltet, wenn das Bit 0 des Steuerlatch F2H low ist (4K-Banking). An den B-Ein-gängen der Multilexer liegen ebenfalls die Bits 0 bis 6 des Latch F0H an, nur, daß jedes Bit um eine Position zum höheren Stellenwert verschoben ist. Wenn das Bit 0 des Steuerlatch F2H high ist (8K-Banking), erscheint das Wort des Latch an den Ausgängen der Multilexer arithmetisch links geshiftet.

Das Bit 7 des Latch F0H liegt dann am Ausgang 1 des dritten Multipexers IC9 an. Am niederwertigsten Ausgang von IC7 liegt dann der Pegel der Adreßleitung A12 an (Abb.3).

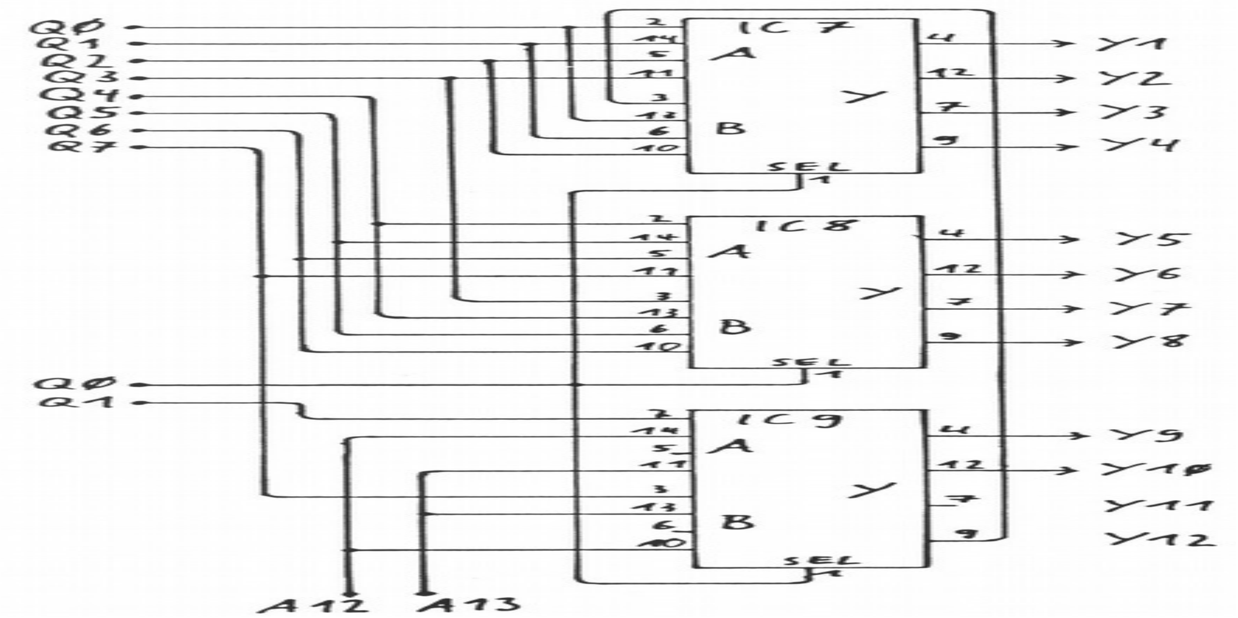


Abb. 3: Einstellen von 4K- oder 8K-Ban- king über Bit 0 des Steuerlatch F2H (Signal Q0). Die Bits des Latch werden bei 8K-Banking arithmetisch links geschiftet durchgeschaltet. A12 wird als LSB durch-geschaltet.

Die Adreßleitungen A12 undA 13 werden durch den dritten 74LS157 (IC9) ebenfalls gemul-tiplext, so daß beim 4K-Banking A12 an den vierten Eingang des 4 Inputs NAND Gatters (IC9) anliegt, während beim 8K-Banking A13 an diesen Eingang gelegt wird und A12 an den B-Eingang des Multiplexers, um bei Zugriff auf den Grundspeicherbereich 0-56K als nieder-wertigstes Bit neben den 8 Bit des Latch durchgeschaltet zu werden. Die Daten an den Ausgängen der Multiplexer IC 7 bis IC9 beim 4K-Banking und beim 8K-Banking werden in Tabelle 1 dargestellt.

In beiden Fällen werden die Multiplexer-ausgänge Y1 bis Y4 (am IC7) zusammen mit den System-Adreßleitungen A12 bis A15 an den Multiplexer IC15 gelegt. Durch Selektion des NAND-Gatters IC10 wird eine dieser beiden Gruppen dort durchgeschaltet und ergibt die Adreßleitungen A12' bis A15'.

Zwingend ist die Verwendung der Signale A12' bis A15' nur im 64K-Block auf dem CPU-Board, weil hier auch der nicht umschaltbare Haupt-adreßraum bis 60 (56)k liegt. Alle anderen 64K-Blöcke können auch direkt mit den Ausgängen Y1 bis Y4 des Multiplexers IC7 als Adresse A12 bis A15 beschaltet werden, weil diese Speicher je nach eingestellter Bank im Latch F0H nur bei Ausgabe von System-adressen oberhalb F000H (E000H) ange- sprochen werden.

Die Multiplexer Ausgänge Y5 bis Y8 werden an zwei 74LS154 (1 aus 16 Decoder) gelegt und ergeben die Block-Auswahlimpulse BS0 bis BS31, die den jeweiligen 64K-RAM-Block frei-

geben.

Y9 gibt jeweils einen der beiden 1 aus 16 Decoder frei. Beim 4K-Banking muß diese Auswahl durch Setzen von Bit 1 im Steuerlatch F2H (low: 0-1M Byte Bereich; high: 1-2M Byte Bereich) erfolgen. Die prinzipielle Freigabe beider 1 aus 16 Decoder erfolgt durch das 4 Input NAND (IC10), wenn A12 bis A15 (A13 bis A15) alle high sind.

Y11 wird nicht benutzt. Y10 und Y12 steuern die Schaltung von A12 oder A13 an den Eingang des 4 Input NAND Gatters IC10 und an den Eingang B1 des Multiplexers IC7. Der Grundspeicherbereich, der 64K-Block auf dem CPU-Board, der bei Low auf den Ausgängen des Multiplexers IC8 (Y5 bis Y8) decodiert wird, hat dabei mehrere Funktionen:

Tabelle 1: Von den Multiplexern (IC7 bis IC9) durchgeschaltete Signale für 4K- und 8K-Banking

Multiplexer Ausgang			Durchgeschaltetes Signal 4K Banking	8K Banking
IC7	Y1	Pin 4	Latch F0H Bit 0	A12
	Y2	" 12	" " " 1	Latch F0H Bit 0
	Y3	" 7	" " " 2	" " " 1
	Y4	" 9	" " " 3	" " " 2
IC8	Y5	Pin 4	" " " 4	" " " 3
	Y5	" 12	" " " 5	" " " 4
	Y5	" 4	" " " 6	" " " 5
	Y5	" 4	" " " 7	" " " 6
IC9	Y9	" 3	Latch F2H Bit 1	" " " 7
	Y10	" 12	A12	A13
	Y11	" 7	-------	-------
	Y12	" 9	A13	A12

Der Adreßbereich 16-60 (56)K wird vom Prozessor direkt adressiert. Andererseits ist er auch in den Banking-Zugriff eingeschlossen, so daß jeder 4K (8K)-Bereich durch Ausgabe der Bank Nummern 0-15 (0-7) in das Latch F0H in den oberen Banking-Adreßraum F000H (E000H) bis FFFFH gelegt werden kann. Das kann mit dem ohnehin in diesem oberen Adreßraum liegenden Bereich und mit dem Bereich 0-16K, der parallel zum ROM liegt, gemacht werden. Wenn allerdings Teile des Betriebssystems, des BASIC oder des Anwenderprogramms in den oberen Adreßraum gelegt werden, in den z.B. Meßdaten eingetragen werden, so überschreiben diese Daten die jeweiligen Programme.

Dieses Problem kann z.B. dadurch gelöst werden, daß entsprechende Bank Nummern nicht benutzt werden (4—14 beim 4K-Banking und 2-6 beim 8K-Banking). Sicherer ist aber eine Kontrollschaltung, die immer dann, wenn eine verbotene Bank Nummer an die RAMs durchgeschaltet wird, sofort einen RESET am Latch F0H auslöst und darin die Bank Nummer 0 einstellt, die in jedem Fall erlaubt ist.

In Tabelle 2 sind die Bitkombinationen an den Ausgängen des Multiplexers IC7 angegeben, die beim Banking im 64K-Block auf dem CPU-Board erlaubt sind, und diejenigen, die verboten sind.

Es ist zu erkennen, daß für die erlaubten Bank-Nummern folgende Bedingungen für die Bitkombination gelten:

1. Y4, Y3, Y2 und Y1 müssen alle gleichzeitig high sein oder
2. Y4 und Y3 müssen beide gleichzeit low sein und
3. Y5 bis Y8 müssen alle low sein (64K-Block Nr. 0 decodiert), weil in anderen Blöcken diese Kombinationen erlaubt sind.

Tabelle 2: Bitkombinationen (an den Ausgängen des Multiplexers IC7) der möglichen Bank-Nummern für den RAM-Block auf dem CPU-Board

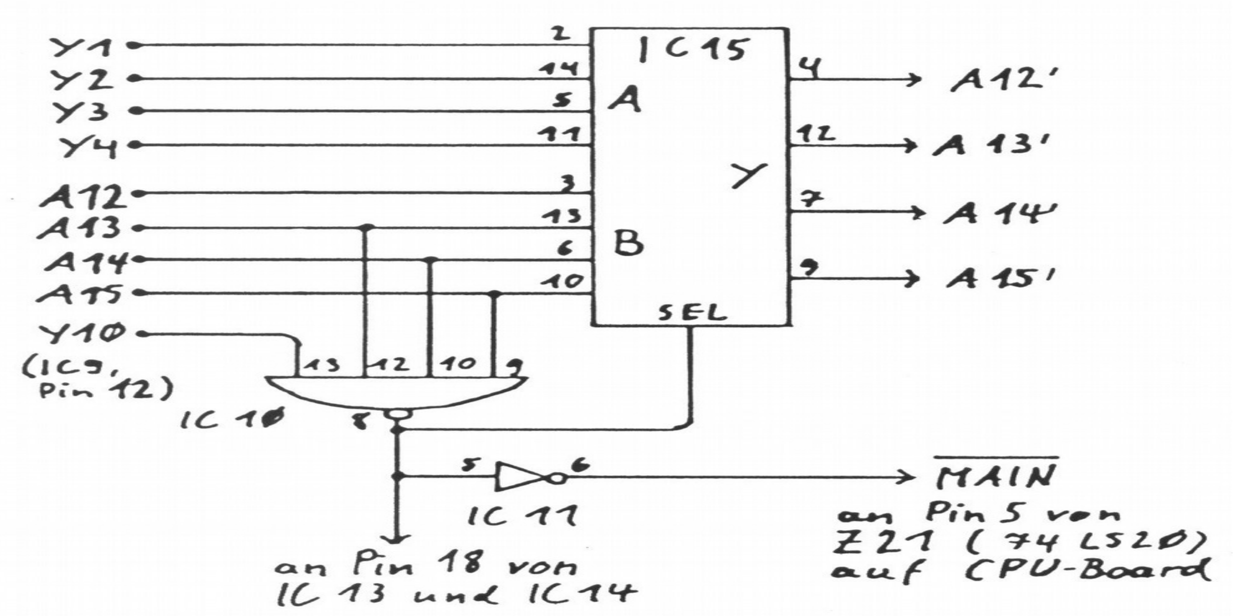
4K Banking Bank Nr.	Bit Kombination Y4	Y3	Y2	Y1	8K Banking Bank Nr.	Bit Kombination Y4	Y3	Y2	Y1	
0-16K					0-16K					
0	0	0	0	0	0	0	0	0	X	\
1	0	0	0	1	1	0	0	1	X	\ erlaubt
2	0	0	1	0						/
3	0	0	1	1						/
16-60K					16-56K					
4	0	1	0	0	2	0	1	1	X	\
|	|	|	|	|	|	|	|	|	|	nicht erlaubt
14	1	1	1	0	6	1	1	0	X	/
60-64K					56-64K					
15	1	1	1	1	7	1	1	1	X	erlaubt

Die "Sicherung" muß also erkennen, wenn Bedingung 3 und eine der beiden anderen Bedingungen gleichzeitig erfüllt sind. Die in Abb. 4 gezeigte einfache Schaltung erkennt diese Bitkombinationen und gibt am letzten NAND-Gatter ein Low-Signal aus, wenn eine verbotene Bank-Nummer vorliegt. Dieses Low-Singal wird zum Löschen des Latch F0H benutzt. Da aber auch durch RESET* das Latch gelöscht werden soll, ist eine AND-Verknüpfung (2-NAND-Gatter in IC4) von RESET und dem Signal der "Sicherung" nötig. An die beiden durch F1H und F2H decodierten Latches kann das RESET* Signal direkt angeschlossen werden.

Diese Schaltung hat einen kleinen Schönheitsfehler. Beim 8K-Banking wird auch der Adreß-bereich E000H bis EFFFH, der als untere Hälfte des oberen 8K-Adreßbereichs durchaus erlaubt ist,

als verboten erkannt (Y1 = A12 = 0) damit sind Y1 bis Y4 nicht alle gleichzeitig high). Diese 4K Byte werden dabei zugunsten erhöhter Sicherheit verschenkt.

Abb.4: Sicherungsschaltung zur Verhinderung verbotener Bank-Nummern. Ver- einfachte Schaltung für beschränkte Anwendung



In vielen Fällen wird ein Umschalten von 4K- auf 8K-Banking durch Software gar nicht gefordert sein. Es genügt hier eine einmalige in Hardware vorgegebene Einstellung auf entweder 4K- oder 8K-Banking.

Weiterhin kann davon ausgegangen werden, daß 1M Byte Speicherplatz ausreichen. Wahrscheinlich kann auch die weiter unten noch beschriebene Bankumschaltung des Adreß-bereichs des Sonder-ROM (3000H - 36FFH) entfallen.

Es kann dann auf die ICs 6,7,8,9 und 14 der Banking-Schaltung verzichtet werden. Anstelle der ICs 7 bis 9 können die Eingänge A (für 4K-Banking) oder B (für 8K-Banking) direkt per Drahtbrücken in der Banking Karte an die Ausgänge Y geschaltet werden.

Der Ausgang Y9 ist an GND zu schalten.

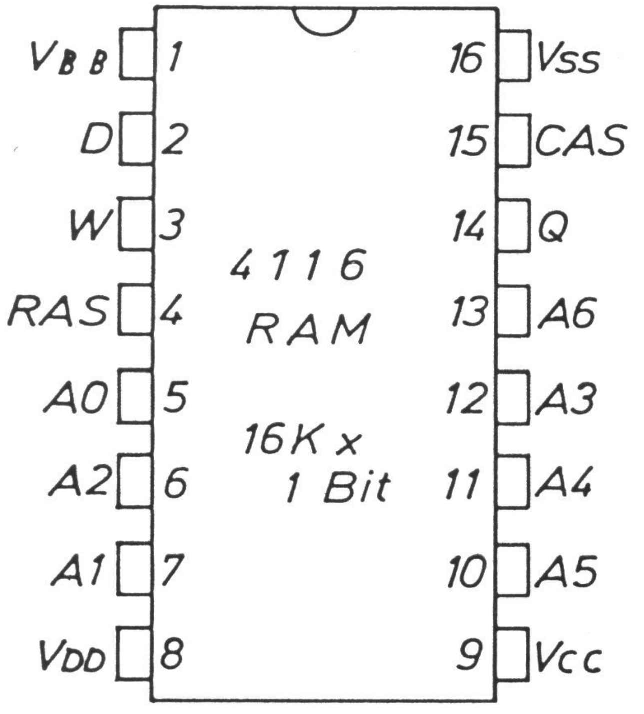
Bei Verzicht auf IC14 (Verwaltung des Bereiches 1-2M Byte) kann auch das Steuerlatch F2H (IC6) entfallen. Damit entfällt aber auch die Möglichkeit, mit dem Latch F1H eine Umschaltung des Adreßbereichs des Sonder-ROM zu verwalten.

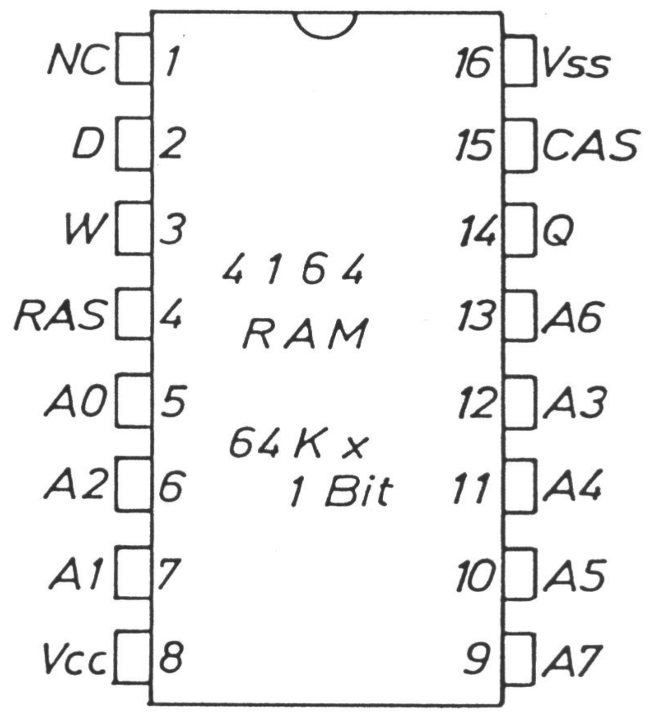
Auch in dieser abgemagerten Version kann die Schaltung beim 83er Modell ohne herkömmliches Expansionsinterface mit Speichersteckplätzen den Zugriff auf die 16K RAM, die parallel zum ROM liegen und deshalb bisher nicht verfügbar waren, ermöglichen.

Notwendige Änderungen im Genie

Beim Genie Mikrocomputer (82er Modell) bestehen im Grundgerät 16K RAM und im Expansionsinterface 32K RAM in Form von 8 und 16 Stück 4116 Chips (16K x 1 Bit). Beim 83er Modell sind die 8 Chips auf dem CPU-Board durch 4164 (64K x 1 Bit) Chips ersetzt worden. Der Unterschied zwischen beiden Typen besteht darin, daß die 4116er RAMs die drei Versorgungs-Spannungen +5V (Vcc), +12V (Vdd) und -5V (VBB) benötigen, während die 4164er Chips nur noch +5V benötigen -ZAbb.5). Um in den gleichen Sockeln 4164er RAMs benutzten zu können, müssen die drei Versorgungsspannungs-Zuführungen zu den RAM-Blocks abgetrennt werden. +5V ist nun an die Pins 8 der RAMs anzulegen, während an die Pins 9 (bisher beim 4116 +5V) A7 anzulegen ist. Die Bezeichnung der Adreßleitungs-Pins im Pinout der beiden RAM-Typen ist nicht zu verwechseln mit den Adreßleitungen des Systembus A0 bis A7. An die Pins A0 bis A7 der RAMs werden die durch Multiplex nacheinander durchgeschalteten Gruppen A0 bis A7 und A8 bis A15 des Systembus gegeben.

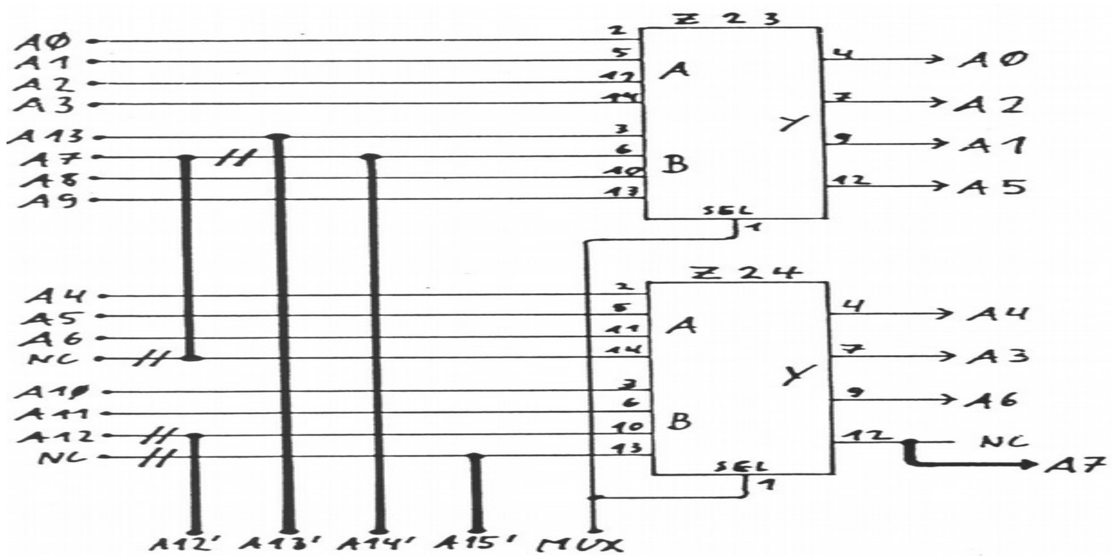
Abb. 5: Pinouts der RAM- Typen 4116 und 4164



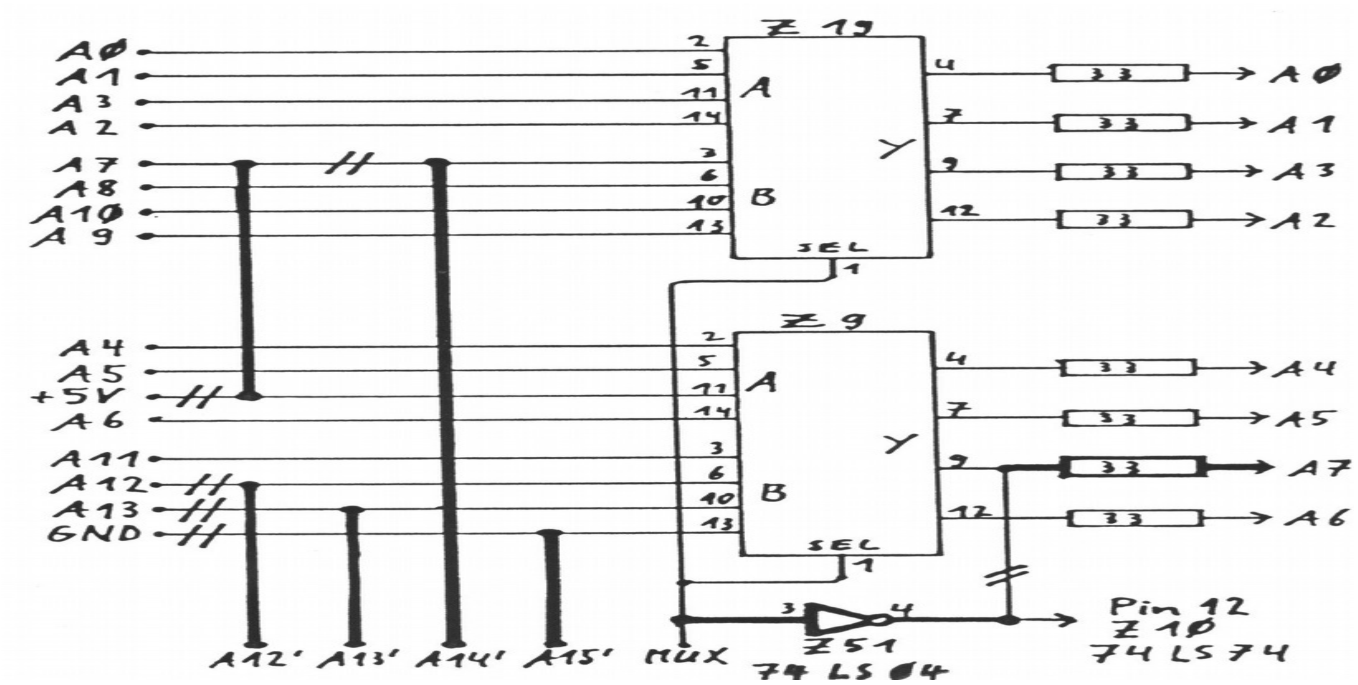


Gegenüber der Belegung der Multi-plexer 74LS157 bei Benutzung von 4116 RAMs, wo A0 bis A6 und A7 bis A13 als Gruppen nacheinander durchgeschaltet werden, muß bei Benutzung von mehr als nur einem 64K-Block wegen des Refreshings unbedingt A7 mit A0 bis A6 zusammen durchgeschaltet werden, während A8 bis A13 zusammen mit den dazukommenden Adressen A14 und A15 durchge-schaltet werden müssen. Die nötigen Änderungen an den Multiplexern im Grundgerät und im Expansionsinterfache sind in Abb. 6 und Abb. 7 gezeigt.

Abb. 6: Änderung der Beschaltung der Adreßmultipexer des RAM-Blocks auf dem CPU-Board



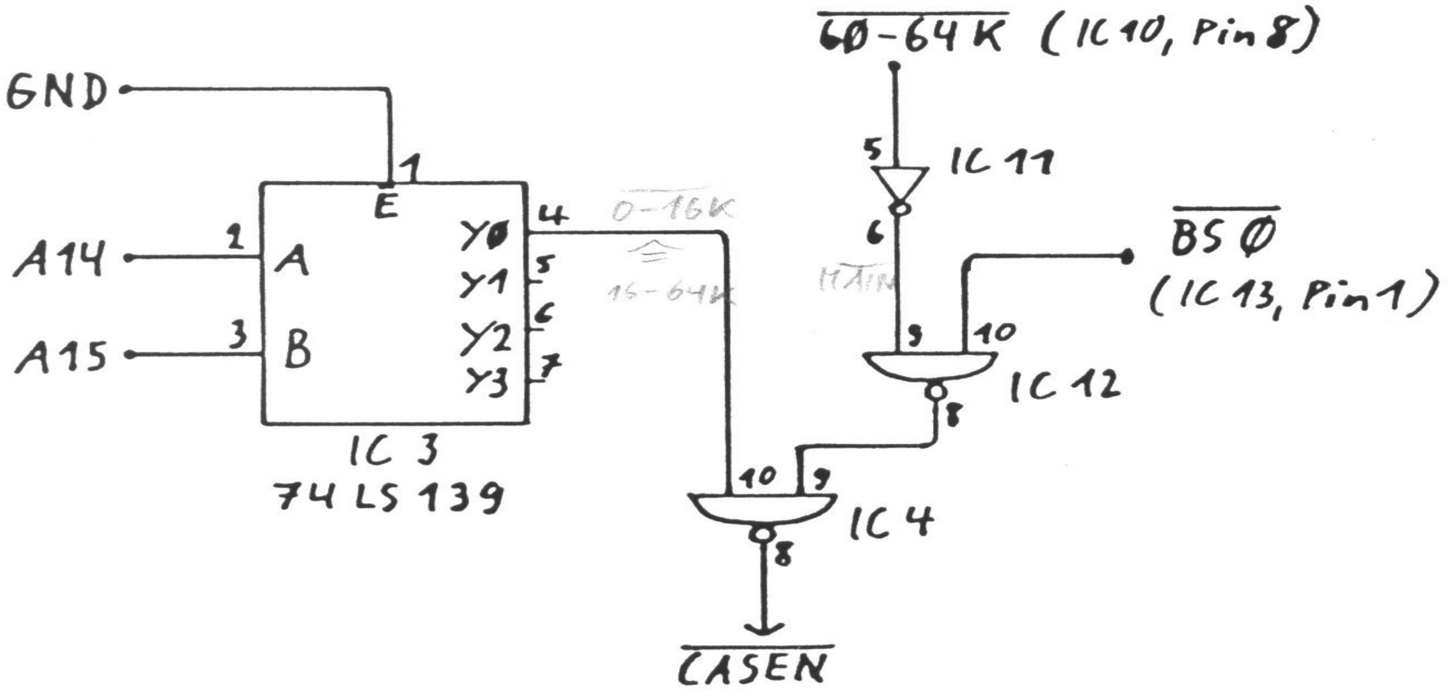
Anstelle der Systembus-Adressen A12 bis A15 sind zumindest auf dem CPU-Board die Signale A12' bis A15' der Banking-Schaltung zu benutzen. Auf dem Board im Expansionsinterface und auf eventuell dazukonmenden weiteren Speicherkarten können auch die Signale Y1 bis Y4 von IC7 anstelle von A12' bis A15' verwendet werden.



Anstelle der bisherigen Blockauswahlinpulse 32K und 48K (Pins 6 und 7 von Z29, 74LS139) sind im Expansionsinterface die 64K-Blockauswahlimpulse BS1 und BS2 (Pins 2 und 3 von IC13, 74LS154) der Banking-Schaltung an Pins 1 und 2 von Z32 (74LS00) und Pins 10 und 12 von Z22 (74LS32) zu legen. Die Pins 9 und 10 von Z37 (74LS20), an denen auch die Signale 32K und 48K lagen, müssen beide mit dem Signal von Pin 8 des IC12 74LS00) der Banking-Karte beschaltet werden. Die entsprechenden Verbindungen zwischen Z29 und Z37 und zwischen Z37 und (Z22 und Z32) müssen getrennt werden.

Die Auswahl des 64 K Blocks auf dem CPU Board und auch der ROMs soll unter mehreren Voraussetzungen möglich sein:

- Wenn eine Adresse oberhalb von F000H (E000H beim 8K-Banking) auf dem Systembus liegt und der 64K-Block 0 eingestellt ist (und beim 4K-Banking Bit 1 im Steuerlatch F2H low ist):
  Dann liegt an Pin 1 des 1 aus 16 Decoders IC13 (74LS154) Low-Pegel an (BS0 = 0).
- Wenn eine Adresse unterhalb F000H) vom Prozessor ausgegeben wird: Dann ist der über IC11 invertierte Ausgang Pin 8 des 4 Input NAND-Gatters IC10 low aktiv.



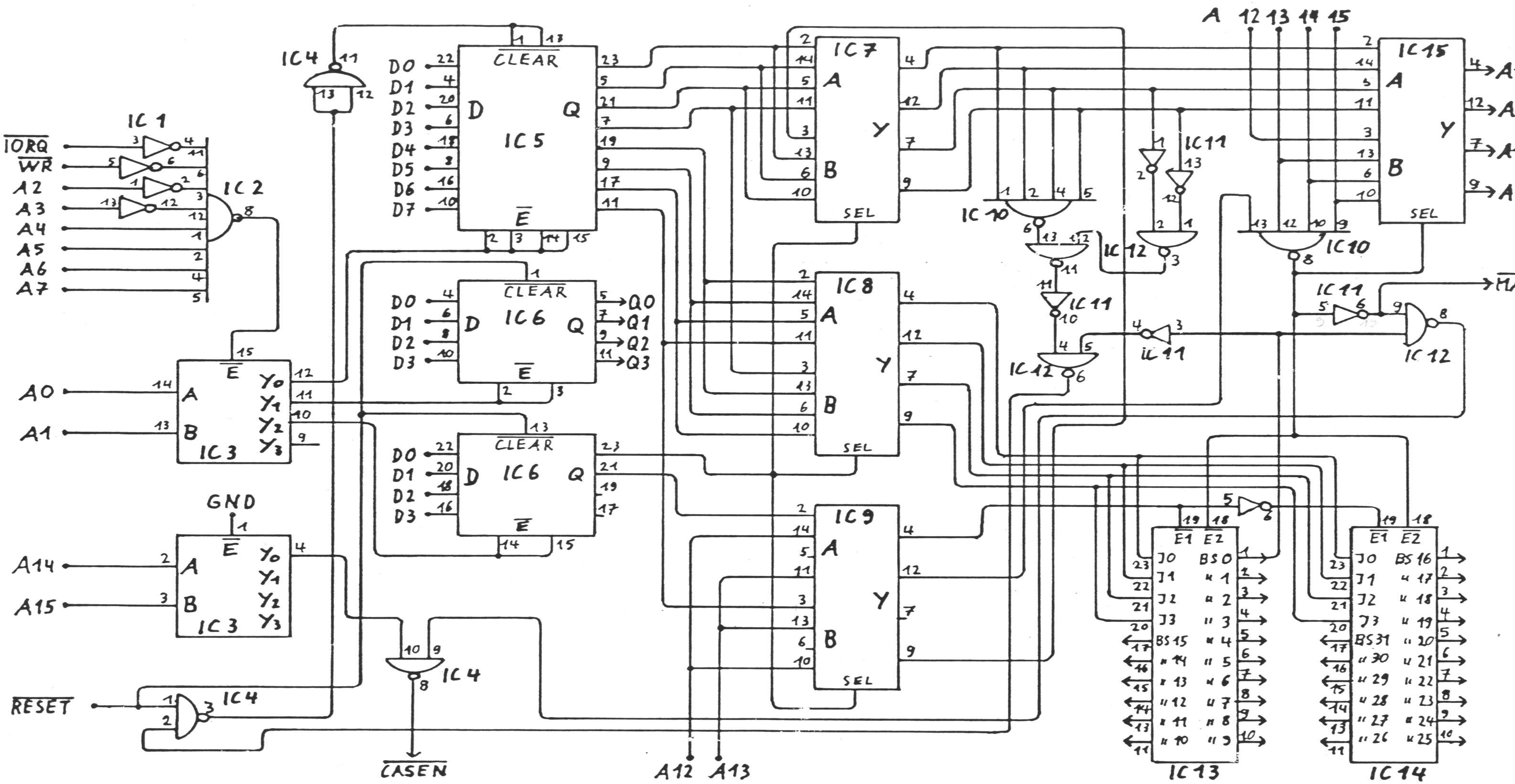
Diese beiden Signale werden an die Pins 1 und 2 sowie 4 und 5 von Z21 (74LS20) auf dem CPU-Board gelegt. Die bisherigen Anschlüsse 0-16K* und 16-32K* (Pins 12 und 11 vom Z25, 74LS139 im 82er Modell) oder 0-16K* und dessen Inversion (über Z36, 74LS04) 16-64K* (Pin 12 von Z25 und Pin 2 von Z36, 74LS04 im 83er Modell) werden abgetrennt.

Die NAND-Verknüpfung der Signale BS0(IC13, Pin 1) und MAIN = 0-60 (56)K (IC 11, Pin 8) über IC12, Ausgang Pin 8 wird mit dem in der Banking-Schaltung in der zweiten Hälfte des IC3 (74LS139) aus A14 und A15 gewonnenen Signale 0-16K* im IC4 NAND-verknüpft und ergibt das Freigabesignal CASEN* für den CAS-Treiber Z37 (74LS367) Pin 15 auf dem CPU-Board (Abb.8).

Abb. 8: Erzeugung des CAS-Trei-ber-Freigeabesignals CASEN* für den 64K-Block auf dem CPU- Board

Der 64 K-RAM-Block auf dem CPU- Board ist also nur dann verfügbar, wenn entweder eine Adresse im Be-reich 16-60(56)K ausgegeben wird, oder wenn eine Adresse oberhalb von F000H) ausgegeben wird und gleich-zeitig eine Bank 0 ins Latch F0H eingetragen ist und beim 4K Banking Bit 1 im Steuerlatch F2 H low ist. Die Gesamtschaltung ist in Abb. 9 dar- gestellt. Sie läßt sich bequem auf einer Europakarte unterbringen.

Abb. 9: Prinzipschaltbild der Bankingkarte



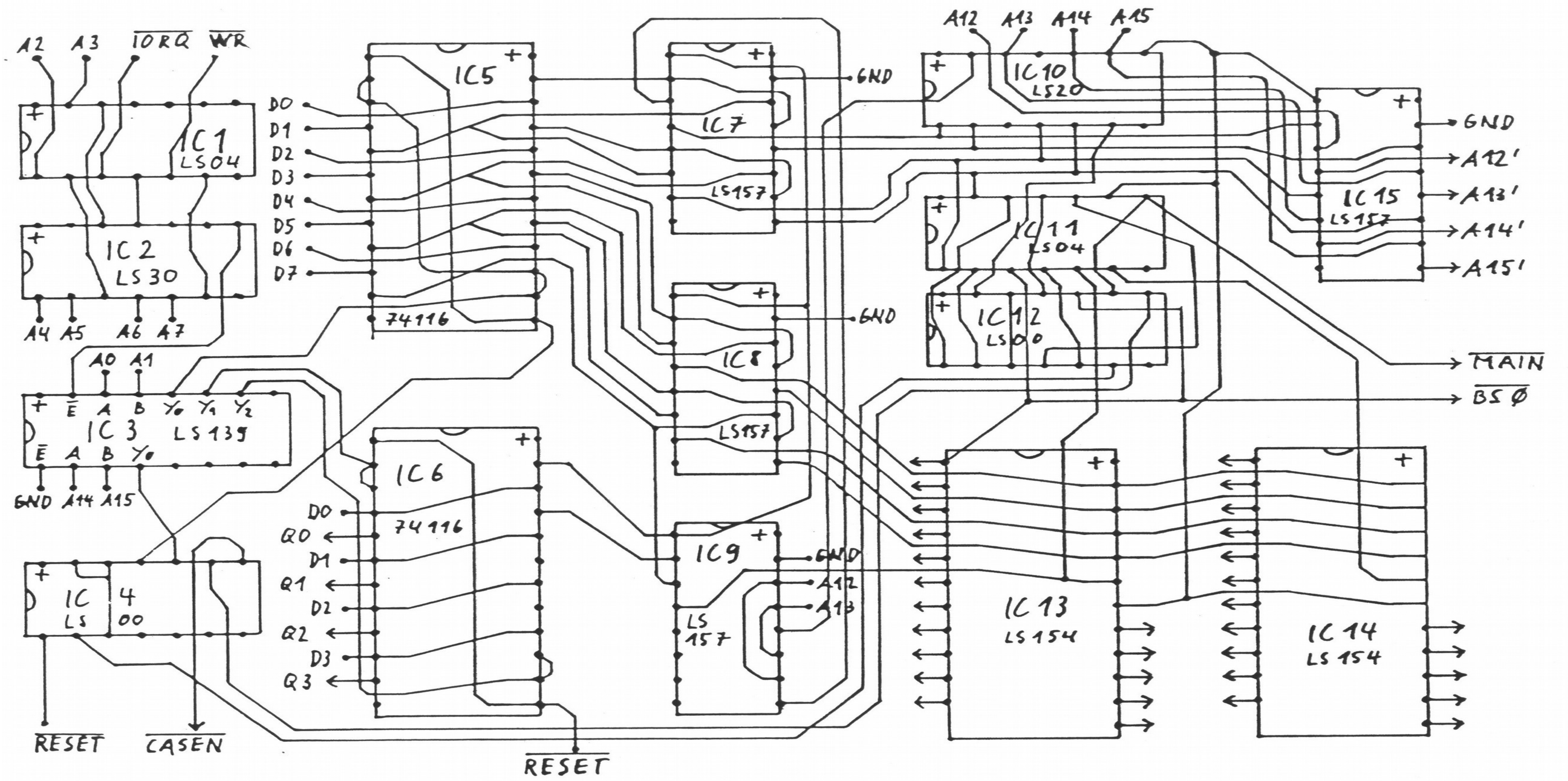
Anstelle eines Layouts, das entweder doppelseitig sein müßte oder viele Drahtbrücken aufweisen müßte, soll hier nur eine Wiedergabe einer günstigen Anordnung der ICs und deren Verdrahtung (durch Drahtverbindungen auf der Rückseite einer Lochrasterplatine oder ein Teil-Layout für ein einseitiges Board mit entsprechenden Drahtbrücken auf der Bestückungsseite) gezeigt werden (Abb. 10). Danach kann sicher jeder je nach den zur Verfügung stehenden technischen Mitteln die Karte selbst aufbauen.

Zweckmäßig ist ein Anschluß an den S-100-Bus-Stecker auf der Lötseite des Boards im Expansionsinterface (Achtung! Die Belegung der Kontakte entspricht nicht der des 50poligen Platinenrandsteckers auf der Rückseite des Grundgerätes. An den Pins 1,3 und 5 liegen z.B. die Spannungen +16V, -16V und +8V an, mit denen man sehr viel Schaden anrichten kann). Die Versorgungsspannung +5V muß an geeigneter Stelle auf dem Board selbst abgegriffen werden.

Anwendung

Die Schaltung ist so konstruiert, daß volle Kompatibilität zum TRS 80, Modell 1 erhalten bleibt. Alle Programme laufen so, wie ohne diese Hardware-Erweiterung. Der Anwender, der nichts davon weiß, merkt die Änderung nicht, es sei denn, er gibt Daten über die Ports F0H und F2H aus.

Abb. 10: Bestückungs- und Verdrahtunqsplan



Allerdings wird der erweiterte Speicher vom DOS und BASIC nicht automatisch mitbenutzt. Um den vollen Speicherplatz vom BASIC aus zu nutzen, können in den einzelnen Banken entweder Maschinenprogramme gespeichert sein, die nach Einschalten der entsprechenden Bank aufgerufen werden können, oder BASIC kann in den Banken Daten durch POKE-Befehle ablegen und durch PEEK-Befehle wieder lesen.

Die Speicherung von Variablen oder gar BASIC-Programmtext in diesen Bereichen wird nicht unterstützt. Vor dem Benutzen der Banken muß zu Anfang HIMEM bzw. MEMORY SIZE auf einen Wert unter 60 (56)K gesetzt werden, um keine Stringvariablen oder eventuell auch numerische Variablen durch Bankumschaltung verschwinden zu lassen.

Banking des Adreßbereiches des Sonder-ROMs

Nachdem nun die Verwaltung von bis zu 2 Megabyte dynamischer RAMs realisiert wurde, soll noch kurz erwähnt werden, daß von IC6 (74116) nur eine Hälfte als Steuerlatch (Portadresse F2H) benutzt wird. Die andere Hälfte mit ebenfalls einem 4 Bit-Latch wurde durch die Portadresse F1H decodiert. Hier stehen noch einmal 4 Bit zur Verfügung, die an einen 1 aus 16 Decoder angeschlossen werden können und dann 16 Freigabe-Signale erzeugen können.

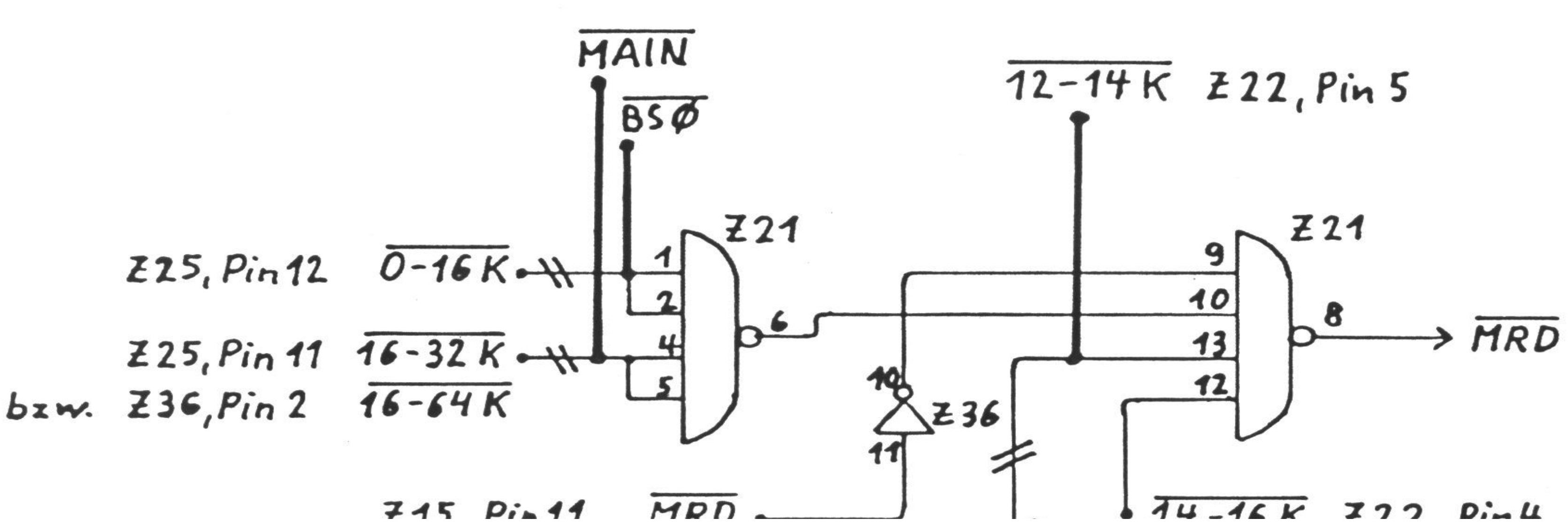
Diese Möglichkeit erlaubt es prinzipiell, den Speicherbereich des Sonder-ROM 3000H bis 36FFH zu vervielfachen, indem das Freigabe Signal für das ROM zur Freigabe des 1 aus 16 Decoders benutzt wird. Der Decoder liefert die 16 Freigabesignale für 2716 EPROMs und/oder pinkompatible 6116 CMOS RAMs.

Durch die Beschaltung des Sonder-ROM im Genie ist die Verwendung aller Signale an den Pins dieses Sockels aber nicht möglich.

1. Der WE*-Eingang, Pin 21, der beim 6116 RAM für Speicherschreibeoperationen auf low gehen muß, ist über einen Widerstand an +5V angeschlossen.
2. Der CE*-Eingang ist direkt an GND angeschlossen.
3. Am OE*-Eingang liegt das 12-14K*-Signal (von Pin5, Z22, 74LS156), das auch bei Adressen im Bereich 3700H - 37FFH einen Zugriff auf das ROM ermöglicht.
4. Das Ausblenden des Bereiches 3700H bis 37FFH erfolgt nicht durch die OE*-Freigabe sondern durch die Freigabe der Speicherlesetreiber. Dies wird durch eine im Schaltplan des Service Manuals nicht dokumentierte Schaltung (Abb. 11) erreicht.

5. Die Datenleitungen sind über unindirektionale Speicherlesetreiber angeschlossen, so daß die Verwendung von RAMs nicht ohne Änderungen möglich ist.

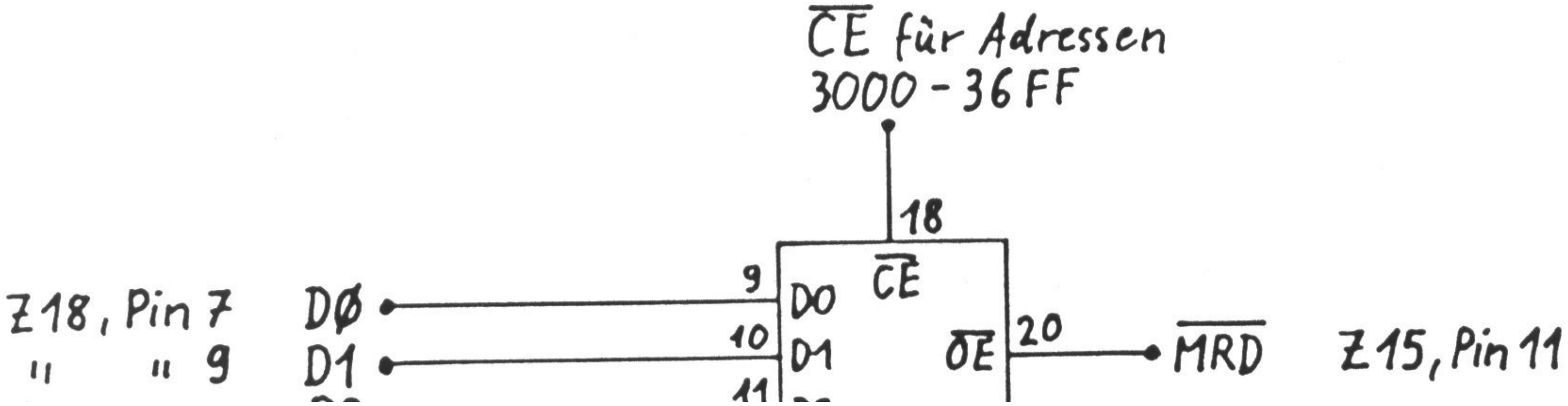
Abb.11: Ausblenden des Speicherbereiches 3700H-37FFH im Sonder-ROM über die Freigabe des Speicher-lesetreibers (im Service Manual nicht dokumentiert)



Zur Benutzung von RAMs dürfen die Datenleitungen nicht über die unidirektionalen Speicherlesetreiber an das Speicherchip geführt werden. Anstelle dieser bisherigen Anschlüsse an den Sonder-ROM-Sockel (Pins 9-11, 13-17) müssen die Signale D0 bis D7 des bidirektionalen Systemdatenbus verwendet werden (Abb.12).

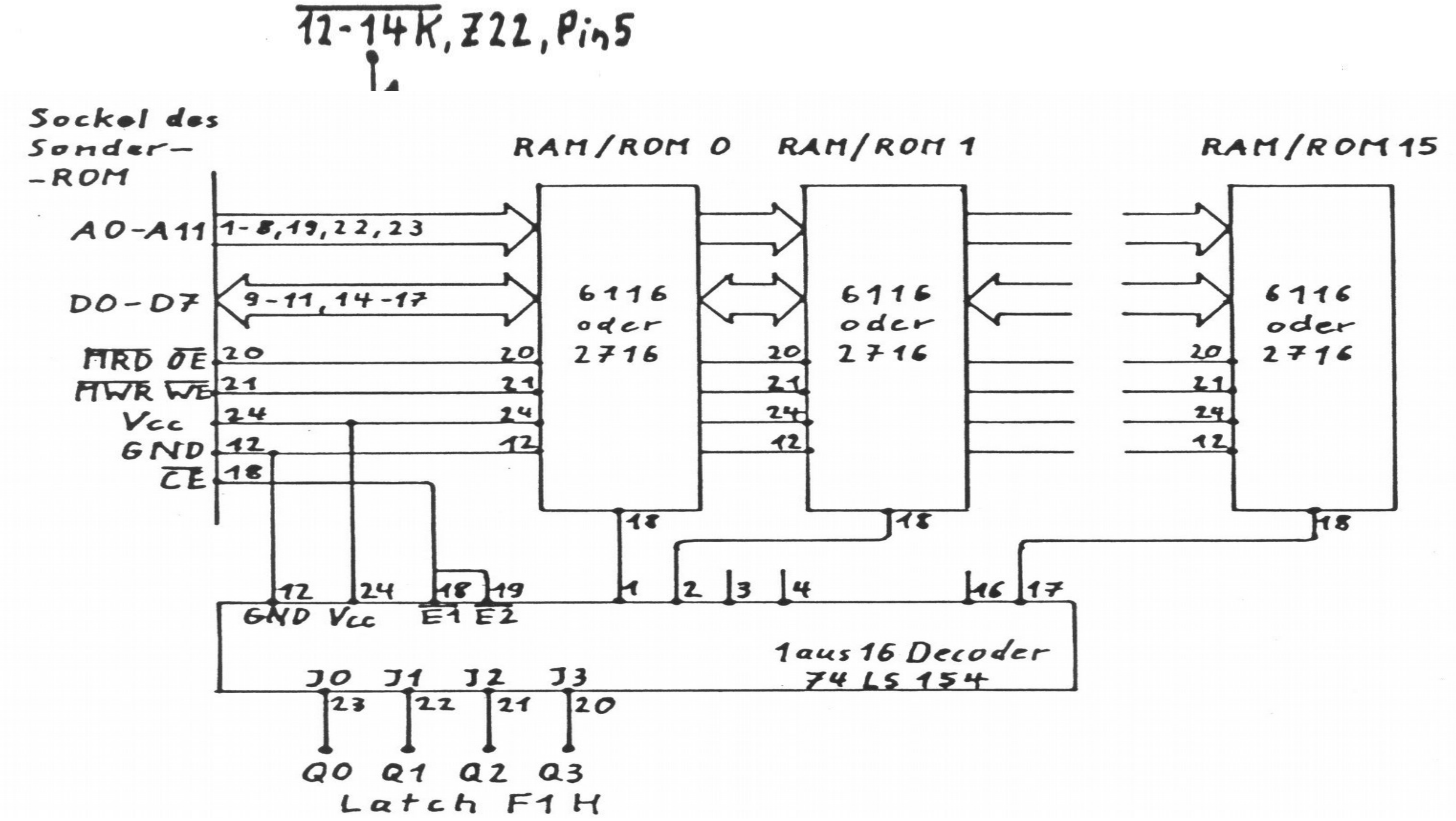
An die Anschlüsse OE*, Pin20 und WE*, Pin 21 müssen sie Signale MRD* (Z36, Pin 11) und MWR (Z33 Pin 3) gelegt werden.

Abb.12: Änderungen am Sockel des Sonder-ROM für das Banking von wahlweise EPROMS und/ oder RAMs im Adreß-bereich 3000H-36FFH



Die Freigabe von CE*, Pin 18 muß neu decodiert werden, da auf das Ausblenden des Bereiches 3700H-37FFH durch die Freischaltung der Speicherlesetreiber verzichtet werden soll. Eine ein- fache Erzeugung dieses Signals ist in Abb 13. gezeigt. Der dazu auf dem CPU-Board zusätzlich nötige Baustein 74LS08 läßt sich direkt neben Z25 (74LS139) in das dort vorhandene freie Feld einsetzen.

Abb.13: Neue Decodie- rung des Sonder-ROM-Adreßbereiches 3000H- 36FFH und des Bereiches der memory-mapped Divi- ces 37XXH



Nun lassen sich alle Signale des Sonder-ROM-Sockel an eine Karte führen, auf der bis zu 16 Stück 2716 ERROMs und/oder 6116 RAMs stecken. Das nach Abb. 13 gewonnene Freigabesignal wird an die Freigabe-Eingänge des 1 aus 16 Decoders gelegt. Die Eingänge 10 bis 13 werden mit den Ausgängen Q0 bis Q3 des Latch F1H verbunden. Die Ausgänge des Decoders schalten dann das jeweils angewählte Speicherchip frei (Abb. 14)

Abb. 14: Ver-16-fachung des Bereichs des Sonder-ROM

Auf diese Weise läßt sich der Speicherbereich 3000H bis 36FFH versechszehnfachen. Das ist besonders interessant, weil dadurch ein Austesten von Maschinenspra-che-Programmen im RAM und späteres Benutzen im EPROM bei gleichen Speicheradressen möglich wird. Es läßt sich dadurch ein Gesamtspeicherraum von 28K in Form von EPROM-residenten Maschinenprogrammen, Debuggern, Uti- lities, BASIC-Erweiterungen usw. bereitstellen, die weder bei Gebrauch erst geladen werden müssen noch den Hauptarbeitsspeicher belegen.

Wenn der 2716 Sonder-ROM aus dem Genie durch Pin 1 des 1 aus 16 Decoders 74LS154 freigegeben wird, so ist gewährleistet, daß beim Einschalten des Gerätes (durch RESET wird in das Latch 0 eingetragen) immer dieser Baustein freigegeben ist und damit genauso verfügbar ist wie im Urzustand.

WHV Hardware Sonderheft 11/85
Nachbesserung nach WHV 12/85 S.29 durchgeführt

RAM bei 0000H-37DFH anstelle des Sonder-ROMs im GENIE

Helmut Bernhardt

Der Sonder-ROM des GENIE hat durchaus seine angenehmen Seiten; es gibt auch kaum GENIE-Besitzer, die ihn endgültig aussondern (eine ebenfalls mögliche Interpretation seines Namens). Aber, wenn man ehrlich zu sich selbst ist, muß man rückwirkend feststellen, daß man ihn so gut wie nie benutzt hat. Manchmal fragt man sich, ob es nicht besser wäre, diesen kostbaren Adreßraum als RAM zur Verfügung zu haben. Aber warum nicht das Eine tun ohne das Andere zu lassen.

Eine kleine Umbaumaßnahme führt dazu, daß im Grundzustand der Sonder-ROM verfügbar ist und auf Tastendruck (ohne Treiberroutine) auf RAM umgeschaltet wird. Es wurde dabei die Kombination der Tasten Y und Z ausgewählt, die hinreichend weit auseinander liegen, um ein ungewolltes Umschalten auszuschließen und wegen ihrer Lage in der Tastaturmatrix den Hardwareaufwand gering halten.

Dazu muß leider zunächst ein Teil der Arbeit, die dafür aufgebracht wurde, den Sonder-ROM voll zu decodieren, um im Bereich 3000H-37DFH RAM zur Verfügung zu stellen, wieder zunichte gemacht werden. D.h., das dabei eingesetzte IC 74LS08 wird wieder entfernt und durch ein 74LS11 ersetzt, und mit den gleichen Signalen wie im Schaltplan zu ersehen ist, versorgt. Auch das Signal SYSRES* (aus RESET* und NMI* durch AND-Verknüpfung gebildet) wird hier wiederhergestellt und auch gleich verwertet.

Zusätzlich zu den Signalen, die das 74LS08 schon erhielt, wird jetzt auch noch das das Signal 3000-37DF* an Pin 9 des 74LS11 gelegt. Dieses Signal wird an Pin 11 eines zusätzlichen 74LS32 erhalten und über ein weiteres OR-Gatter dieses ICs mit dem Ausgang eines Flip Flops verknüpft. An Pin 8 dieses Gatters kann nur dann Low-Pegel erscheinen, wenn das Signal 3000-37DF* low ist und wenn der Ausgang des Flip Flops (Pin 6 eines zusätzlichen 74LS00) low ist.

Dieser Ausgang ist aber allgemein high, sodaß das Signal 3000-37DF* nicht durchgeschaltet wird. Der Pegel an Pin 6 des Flip Flops wird bei jedem RESET (Einschalten des Computers) und jedem NMI (Drücken des RESET-Knopfes) auf high eingestellt. In diesem Zustand kann also bei Adressen zwischen 3000H und 37DFH nie Low-Pegel an Pin 9 des 74LS11 ankommen.

Erst wenn von Pin 6 des zusätzlich einzubauenden 74LS32 ein Low-Impuls ausgegeben wird und das Flip Flop umkippt, gelangt jedesmal, wenn eine Adesse zwischen 3000H und 37DFH auf dem Bus liegt, das low aktive Signal 3000-37DF* an den Pin 9 des 74LS11.

Der Ausgang dieses AND-Gatters (Pin 8) ist immer dann low, wenn einer seiner Eingänge low ist, also wenn Adressen in den Bereich 16-64K, 3900H-3BFFH und nun auch 3000H-37DFH auf dem Bus liegen. Der Pin 8 steuert die Freigabe des RAM-Speichers. Immer, wenn dort LOW-Pegel ausgegeben wird, erfolgt ein Zugriff auf das RAM.

Genauso, wie der Pin 6 des Flip Flops über das OR-Gatter (Pins 10, 9, 8) regelt, ob das Signal 3000-37DF* eine RAM-Freigabe bewirkt, steuert der Pin 8 des Flip Flops, ob durch dieses Signal der Sonder-ROM freigeschaltet wird. Da die Pins 6 und 8 immer entgegengesetzte logische Pegel haben, können keine Konflikte zwischen RAM und Sonder-ROM entstehen.

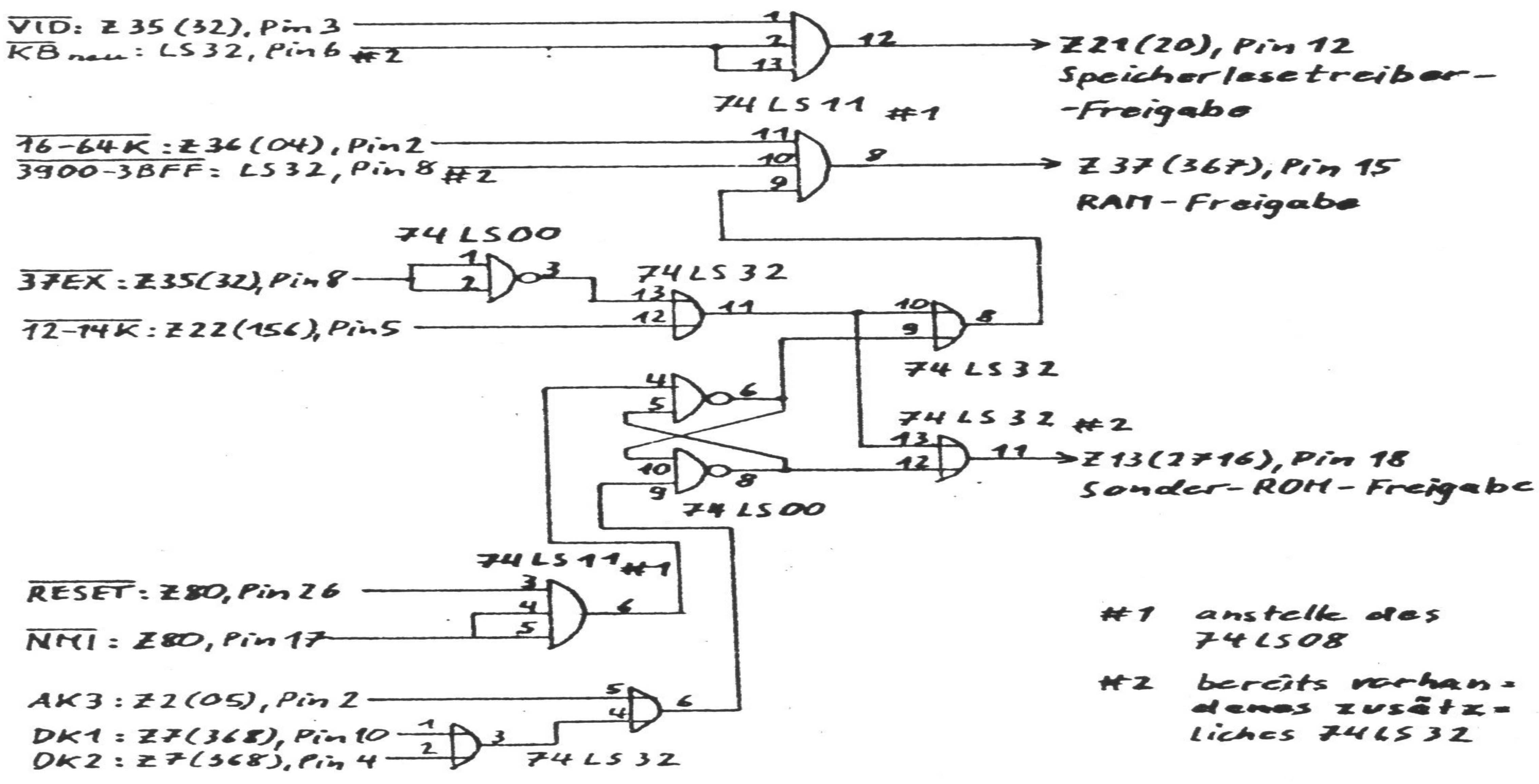
Das Umkippen des Flip Flops (kein Sonder-ROM mehr verfügbar, dafür RAM in dessen Adreßbereich) wird durch OR-Verknüfung von Tastatur-Signalen erreicht. Dafür muß die Tastaturadreßleitung AK3 low aktiv sein und die Tastatur-Datenleitungen DK1 und DK2 müssen ebenfalls low sein. AK3 wird immer dann low, wenn die Tastaturadresse 3808H gelesen wird. Wenn in diesem Moment die Tasten Y und Z gedrückt sind, werden dadurch DK1 und DK2 auf low gezogen das dann an Pin 6 der beiden hintereinander geschalteten OR-Gatter ausgegebene Low-Signal schaltet das Flip Flop um.

Das Umschalten ist nur dann möglich, wenn eine Tastatur-Treiberroutine aktiv ist (NEWDOS/80 READY; Befehlseingabe; Komm, laß knacken u.s.w.), oder wenn ein Programm die Adressen 3808H (oder jede andere Adresse im Bereich 3800H-38FFH mit A3 = 1) liest.

Umbauanleitung

Die Umrüstung baut darauf auf, daß der Computer mit 4164-RAMs bestückt ist und daß die Erweiterung zur vollen Decodierung des Sonder-ROMs und zur Bereitstellung von RAM im Bereich 3900H-3BFFH schon durchgeführt wurden.

Das bei diesem Umbau mit eingesetzte IC 74LS08 wird entfernt und durch ein 74LS11 ersetzt. Die Leitungen, die vorher an das 74LS08 führten, werden nun gemäß Schaltplan an das 74LS11 gelegt.



Außerdem sind noch ein weiteres 74LS32 und ein 74LS00 nötig. Da in der näheren Umgebung des Schlachtfeldes keine geeigneten DIL14-ICs mehr vorhanden sind, auf die diese ICs huckepack-gelötet werden können, müssen DIL16-ICs als Fundament herhalten (z.B. Z8 und Z9; 74LS368 und 74LS367). Dabei muß das DIL14-Huckepack-IC mit seinen Pins 7 und 14 etwas Spagat üben, um mit den Füßen an die Pins 8 und 16 des DIL16-ICs darunter zu gelangen. Es gilt auch hier wieder "Die Pins 1-6 und 8-13 werden hochgebogen". Das Freigabesignal für den Sonder-ROM wird über das noch freie OR-Gatter des 74LS32 (Pins 13, 12, 11), das bei der letzten Umbaumaßnahme schon eingesetzt wurde, erhalten. Die Leiterbahn, die zu Pin 18 des Sonder-ROM führt, muß durchtrennt werden.

Wenn die an die Pins 4 und 9 des zusätzlichen 74LS00 (nicht 74LS04, wie hier ursprünglich angegeben) führenden Leitungen vertauscht werden, liegt von vornherein RAM vor und die Tasten Y und Z schalten auf Sonder-ROM um.

Bei eingebautem "Ver-16-facher des Sonder-ROM-Bereichs" ist dieses Projekt weder nötig noch möglich.

Hardware-Umschaltung für den Betrieb von CP/M

bei den Computern TRS80, GENIE I, II und Komtek 1

Helmut Bernhardt

Der Einsatz des Betriebssystems CP/M ist auf TRS80 und kompatiblen Computern nicht ohne weiteres möglich, weil im Adressbereich 0000H bis 3FFFH leider kein RAM vorliegt. CP/M fordert aber, daß nach dem Booten des Betriebssystems, wofür natürlich ab Adresse 0000H ein ROM mit einer Boot-Routine vorliegen muss, durch Software ab Adresse 0000H bis zur Obergrenze des Speicherausbaus RAM eingeschaltet werden kann, weil das BDOS die ersten 256 Bytes nutzt und die Anwenderprogramme ab Adresse 0100H im Speicher liegen.

Eine für den TRS80 angepaßte Version, bei der diese unteren Grenzen nach 4000H verschoben sind, fordert natürlich, daß auch die Anwenderprogramme um 16K im Speicher verschoben liegen und dort auch funktionieren. Solche angepassten Anwenderprogramme soll es sogar geben; für die Vielzahl der unter CP/M laufenden Programme entfällt aber die Nutzung unter diesem Not-CP/M.

Es gibt mehrere Möglichkeiten (die auch teilweise in verschiedenen Aufrüst-Sätzen für die verschiedenen Rechner realisiert und angeboten werden), wie die Umschaltung auf RAM im Bereich 0000H bis 3FFFH erreicht werden kann. Voraussetzung für alle diese Varianten ist allerdings das Vorhandensein von 64K RAM in Form von 4164-ICs auf dem CPU-Board, wie dies in den letzten Modellen des EACA-GENIE schon der Fall war. Diese Speichererweiterung läßt sich in allen o.a. Computern realisieren.

Bei einem unter dem Namen BIGMEM für den TRS80 angebotenen Aufrüstsatz erfolgt die Umschaltung auf CP/M-Fähigkeit durch den Befehl OUT 50H,40H, bei dem der Adressbereich 0000H bis 3FFFH mit dem Bereich C000H bis FFFFH ausgetauscht wird, d.h. ROM, Drucker, Tastatur und Video-RAM liegen am oberen Speicherende, und unterhalb der Adresse 4000H ist ausschließlich RAM vorhanden. Außerdem ist der ROM auch noch ausgeschaltet, sodaß der gesamte Bereich 0000H - F7DFH als RAM verfügbar ist und alle memory mapped I/O-Adressen darüber liegen. Es besteht auch die Möglichkeit, mit dem Befehl OUT 50H,00H nur den ROM gegen RAM auszutauschen und die memory mapped I/O-Adressen an ihrem alten Platz zu belassen. In dieser Konfiguration läßt sich ein ins RAM kopierter ROM-Inhalt beliebig ändern. Nachteilig ist bei BIGMEM, daß ein Zurückschalten aus einer umgeschalteten Konfiguration nur durch RESET* und NMI* möglich ist.

Eine andere Variante ist der EG64MBA für den GENIE. Damit lassen sich durch Ausgabe verschiedener Daten an den Port DFH einzelne Bereiche innerhalb der unteren 16K gegen RAM austauschen und durch einen anderen Portbefehl auch wieder rückgängig machen. Die Tabelle 1* zeigt die durch verschiedene Daten an den Port DFH erreichte Funktion.

Tabelle 1: Funktionen des EG64MBA

Adressraum	**normale Belegung**		**Umschaltg. auf RAM**	
	F=	Funktion	F=	Funktion
0000-2FFF	0	read ROM	8	read RAM
	1	write ROM	9	write RAM
3000-35FF	2	read L-4	10	read RAM
	3	write L-4	11	write RAM
3600-37FF	4	r/w I/O (Floppy)	12	r/w RAM
3800-3BFF	5	r/w Tastatur	13	r/w RAM
3C00-3FFF	6	r/w Bildschirm	14	r/w RAM
RESET des MBA	7	per RESET-Knopf	15	nicht per Reset-Knopf

Lesen von Port DFH stellt die ursprüngliche Einstellung wieder her.
Adressraum in hexadezimaler Notation

L-4 ist der Zusatz-ROM (falls vorhanden)
I/O ist nur bei Anschluß von FC-Laufwerken belegt
(TRS-80-Druckeradresse, Floppycontroller, Uhr)

Ein für Anwender nicht unmaßgeblicher Vorteil gegenüber dem BIGMEM besteht darin, daß der EG64MBA einfach auf den Systembus-Stecker hinten am Gerät gesteckt wird (und den Bus auch wieder herausführt), während beim BIGMEM ein erheblicher Verdrahtungsaufwand notwendig ist, bei dem man auch Fehler machen kann.

Beide Aufrüstsätze zeichnen sich dadurch aus, daß dafür angepasste CP/M- Versionen angeboten werden, und daß für die Aufrüstsätze unverschämte Preise gefordert werden. Der Hardware-Freak vermißt auch eine Dokumentation der Schaltung und die Beschriftung der ICs.

Für diejenigen, denen die Einsicht in solche Investitionen fehlt und die stattdessen lieber ein bißchen mehr Arbeit mit dem Lötkolben übernehmen, sei im folgenden eine Schaltung beschrieben, die Features beider Schaltungen vereinigt:

- aufsteckbar auf den Systembus bei allen 3 Computern
- sämtliche Umschaltfunktionen des EG64MBA
- die Verschiebung der unteren 16K an das obere Speicherende, wobei die EG64MBA-Funktionen dann dort wirken

Zusätzlich bestehen noch folgende Möglichkeiten:

- Unter Verzicht auf die ICs 13-15 sind nur die Features des EG64MBA verfügbar.
- Mit Jumpern läßt sich festlegen, ob der Sonder-ROM-Bereich bis 35FFH oder 37DFH reichen und der I/O-Umschaltbereich bei 3600H oder 37E0H beginnen soll (die zweite Variante ist bei GENIE oder Komtek nur dann interessant, wenn der Sonder-ROM voll decodiert ist).
- Lesen des Ports DEH macht die BIGMEM-Funktion wieder rückgängig, ohne daß ein RESET* oder NMI* erfolgen muß.
- In einem Rangierfeld läßt sich das Board an alle drei Computer anpassen.
- Durch etwas zusätzlichen Verdrahtungsaufwand kann das Board auch am Bus des TRS80 betrieben werden, an dem die Signale IORQ*, MERQ*, RD* und WR* nicht vorhanden sind und stattdessen die Signale IN*, OUT*, MRD* und MRW* vorliegen und anstelle von RESET* und NMI* das daraus abgeleitete Signal SYSRES* bereitsteht.

In der einfachen Variante (nur EG64MBA-Features, nur ICs 1-12, 16) kann das Board beim GENIE wie der originale EG64MBA aufgesteckt werden und funktioniert ohne Änderungen auf dem CPU-Board. In der voll ausgebauten Version muß bei allen Rechnern auf dem CPU-Board anstelle des ursprünglichen Signals 0-16K* das Signal des Boards P0-16K'* verwendet werden. Wenn die dafür nötige Änderung durchgeführt wird, funktioniert der Rechner nur noch mit aufgestecktem Board. Mit einem Umschalter (zwischen den beiden Signalen) kann man sich aus dieser Misere herausmogeln.

Um das Signal P0-16K'* vom Board an das CPU-Board zu führen, kann bei GENIE und Komtek 1 ein unbenutzter Pin des Systembus-Steckers verwendet werden. Beim TRS80 ist eine zusätzliche Leitung nötig.

Funktionsprinzip des EG64MBA und dieses Board

Der GENIE weist auf dem Systembus-Stecker einen Eingang PHANTOM* auf. Wenn dieser Pin auf low gezogen wird, ist der sonst durch MERQ* freigegebene Adressdecoder IC 24 (74LS139) gesperrt. Das bei diesem IC an Pin 12 ausgegebene Signal 0-16K*, von dem alle anderen Freigabesignale in diesem Adressbereich abgeleitet werden, kann dann nicht mehr low aktiv werden, während die Freigabe des CAS*-Signals (durch das invertierte Signal 0-16K* (=16-64K*) dann auch im Adressbereich 0-16K gewährleistet ist.

Der EG64MBA übernimmt die Steuerung des PHANTOM*-Signals in Abhängigkeit von der jeweils auf dem Adressbus liegenden Speicheradresse und vom per OUT DFH,X eingestellten Zustand der Latch-Ausgänge.

Als Latch wurde hier (und wahrscheinlich auch im originalen EG64MBA) ein 74LS259 verwendet, über dessen Eingang A, B und C eines von 8 Ein-Bit-Latches selektiert wird, an dessen Ausgang anschießend der Pegel anliegt, der am Eingang DATA angelegen hat, als das letzte mal der Eingang E* low wurde.

Über einen low aktiven Impuls am Eingang CLEAR lassen sich alle 8 Latches auf low setzen.

An E* wurde das low aktive Freigabesignal OUT DFH* gelegt, während an CLEAR ein aus RESET*, IN DFH* und einem vom Zustand des Ausgangs D7 gesteuertem NMI* abgeleitetem Signal gelegt ist. Die Selektion des jeweiligen Latches wird durch Anlegen der Datenleitungen D0, D1 und D2 an die Eingänge A, B und C erreicht, während D3 an DATA gelegt den Zustand des durch E* freigegebenen Latch-Ausgangs bestimmt.

So läßt sich durch Ausgabe des Datenwortes, bei dem nur das untere Nibble signifikant ist, an den Port DFH ein durch die Pegel von D0 bis D2 adressiertes Latch mit dem Zustand von D3 laden.

Durch Lesen des Ports DFH, RESET* oder einen nicht per OUT (DFH),0FH gesperrten NMI* können alle Ausgänge auf low gesetzt werden.

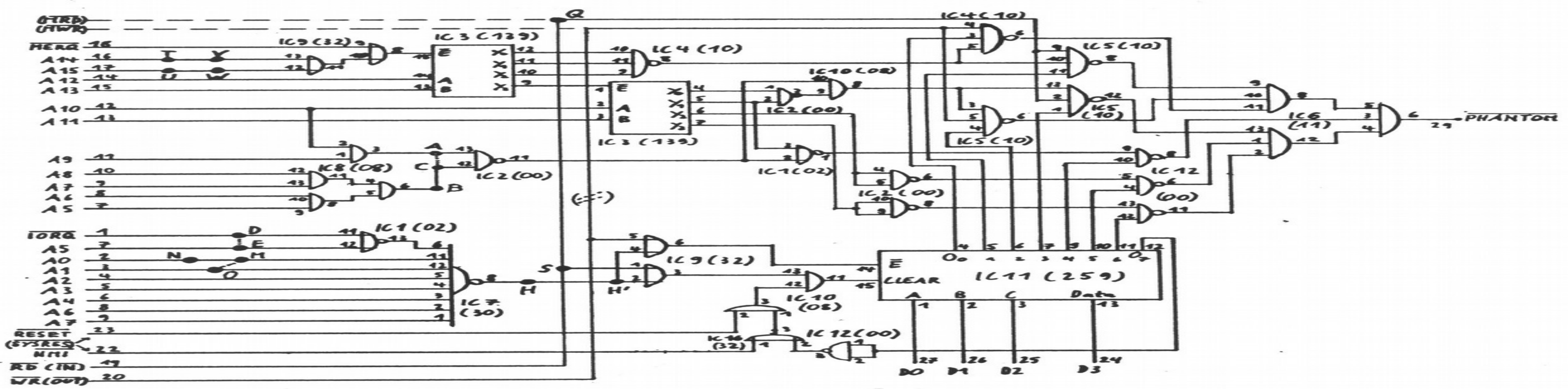
Diese Ausgänge werden mit den jeweiligen high-aktiven Speicherbereichs- Freigabesignalen und bei den beiden ROM-Bereichen zusätzlich noch jeweils einmal mit RD* oder WR* NAND-verknüpft. Sie ergeben dann am Ausgang des jeweiligen NAND-Gatters ein low-aktives Signal, wenn eine in diesem Bereich liegende Speicheradresse auf dem Bus liegt, wenn der steuernde Latch-Ausgang high gesetzt ist und bei den ROM-Bereichen das Signal RD* bzw. WR* nicht aktiv ist.

Über AND-Verknüpfung aller NAND-Ausgänge wird das Signal PHANTOM* erhalten, das immer dann low wird, wenn ein NAND-Ausgang low wird.

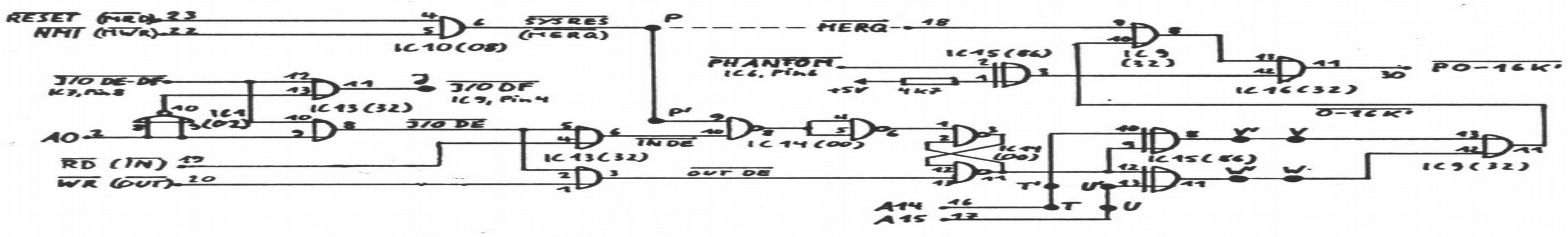
Die zusätzliche BIGMEM-Funktion dieses Boards wird durch Invertieren der Adressen A14 und A15 zu A14' und A15' und der Herleitung eines 0-16K'*-Signals daraus erhalten. Ob die Adressen invertiert werden oder nicht, wird durch den Ausgang eines Flip Flop gesteuert, dessen Zustand wiederum durch die decodierten Portsignale IN DEH*, OUT DEH*, RESET* und NMI* bestimmt wird.

Durch RESET*, NMI* und IN DEH wird das Flip Flop so eingestellt, daß die Adressen über die XOR-Gatter nicht invertiert werden. OUT DEH* kippt das Flip Flop um, so daß die Adressen invertiert werden.

Um den beim TRS80 und Komtek 1 gegebenen Voraussetzungen (kein PHANTOM*-Eingang auf dem Bus) Rechnung zu tragen, wurden nun auch beim GENIE nicht die Signale A14' und A15' an den Adressdecoder geführt, der das Signal 0-16K* herleitet. Es wurde auf dem Board ein Signal 0-16K'* aus A14', A15' und MREQ* hergeleitet und mit dem Signal PHANTOM* verknüpft. Dieses dabei erhaltene Signal P0-16K'* muß nun auf dem CPU-Board anstelle des urspünglichen Signals 0-16K* verwendet werden.



1) Schaltplan für die Grundfunktion (nur EG64MBA)



2) Schaltplan für die Erweiterung (zusätzlich BIGMEM)

Auswahl der Funktionen des Boards über Drahtbrücken und Jumper

Grundversion, nur EG64MBA-Funktionen:
(die ICs 13 bis 15 brauchen nicht bestückt zu werden)

- Punkte neben IC 7: N mit M verbinden

- Punkte unterhalb IC 9: T mit V und U mit W verbinden
- H (über IC 7) mit H' über IC 9 verbinden

Erweiterte Version, EG64MBA- und BIGMEM-Funktionen
(ICs 13 bis 15 müssen bestückt werden)

- Punkte neben IC 7: M mit O verbinden
- Punkte unterhalb IC 9: folgende Verbindungen mit den Punkten unter IC 15 herstellen: T mit T', U mit U', V mit V', W mit W'
- Anstelle von H (über IC 7) wird J (über IC 7) mit H' verbunden
- Punkt X (neben IC 3) und Y (neben IC 15) miteinander verbinden

Allgemeine Drahtbrücken und Jumper für alle Rechner

Zwischen den unteren beiden IC-Reihen müssen folgende Punkte verbunden werden:
1 mit 1', 2 mit 2', 3 mit 3', 4 mit 4', 5 mit 5', 6 mit 6'

Zwischen IC 1 und IC 2 kann durch Wahl der Verbindung A-C oder B-C die Einteilung der Größe der umschaltbaren Bereiche des Sonder-ROMs und des Drucker/Floppy-Bereichs eingestellt werden:

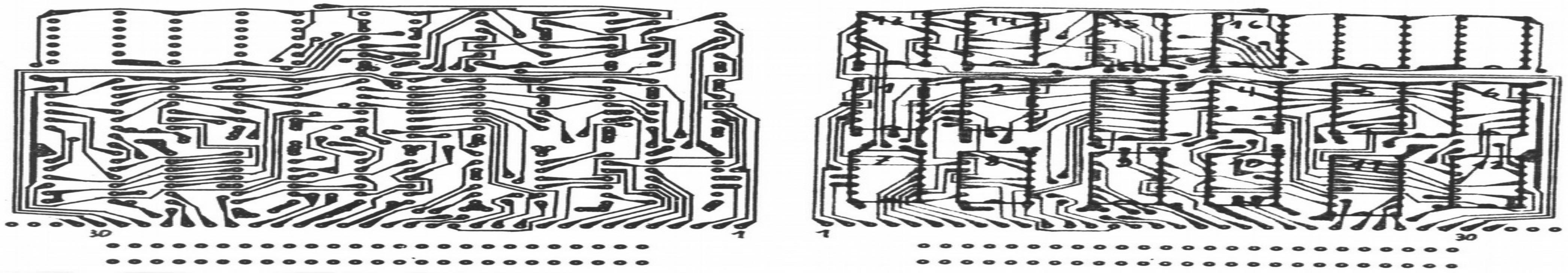
Verbdg. A-C:	Sonder-ROM	3000H - 35FFH
	Drucker/Floppy	3600H - 37FFH
Verbdg. B-C:	Sonder-ROM	3000H - 37DFH
	Drucker/Floppy	37E0H - 37FFH

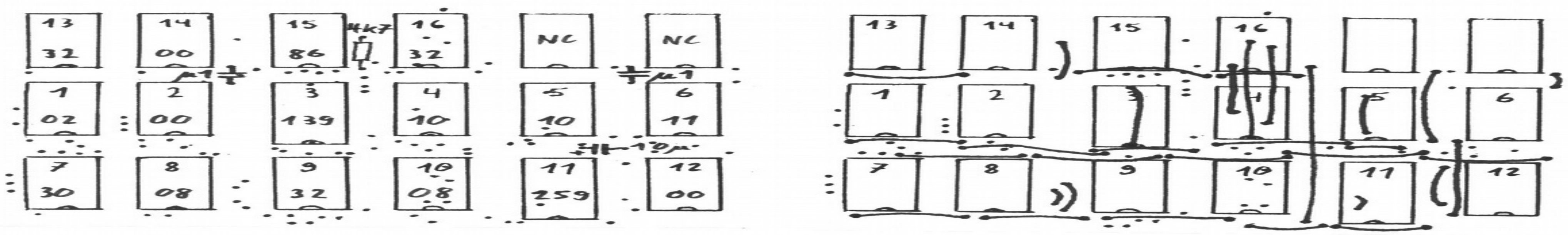
Die Variante B-C hat bei GENIE und Komtek nur dann Sinn, wenn der Sonder-ROM nachträglich voll decodiert wurde und beim Komtek die Druckerschnittstelle (8255 auf den Adressen 36X0H - 36X3H) geändert wurde. Anstelle einer festen Einstellung durch Jumper kann auch ein Umschalter eingesetzt werden.

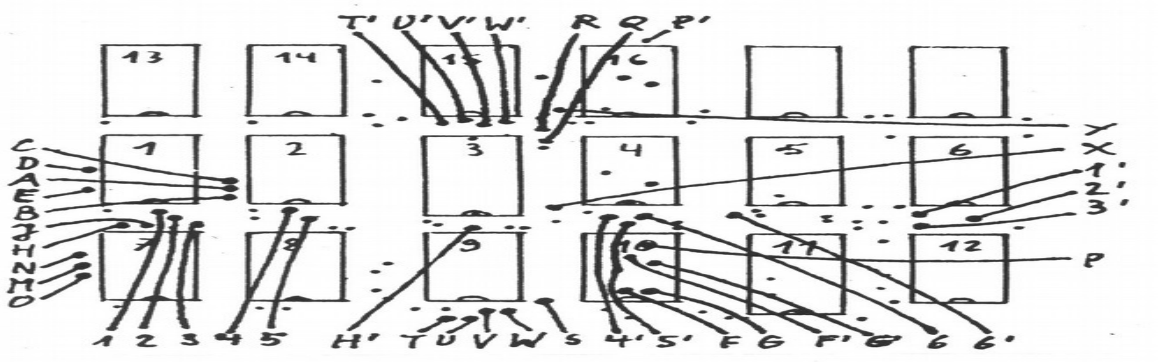
Bei GENIE und Komtek werden außerdem noch die Verbindungen F-F', G-G', S-O, Q-R, P-P' hergestellt; die Punkte D und E werden nicht miteinander verbunden.

Änderungen für den Anschluß an den TRS80

1. Die Drahtbrücken S-Q und Q-R werden nicht gelegt.
2. Die Leiterbahn von Pin 5 von IC 9 zu Pin 4 von IC 4 wird durchtrennt.
3. Die Drahtbrücken F-F', G-G', und P-P' werden nicht gelegt.
4. Die Punkte D und E neben IC 1 werden miteinander verbunden.
5. Die Punkte S und R werden miteinander verbunden.
6. Anstelle von RD* und WR* werden IN* und OUT* an die Anschlußleiste des Boards geführt.
7. Das Signal MRD* des Systembus wird an Pin 2 von IC 5 und Pin 4 von IC 10 geführt.
8. Das Signal MWR* wird an Pin 4 von IC 4 und Pin 5 von IC 10 gelegt.
9. Pin 6 von IC 10 wird mit Pin 9 von IC 9 verbunden.
10. Das Signal SYSRES* des Systembus wird an die Anschlußpunkte RESET* und NMI* und Punkt P' oberhalb von IC 16 geführt. Damit besteht dann nicht die Möglichkeit, das Zurückschalten in den Ausgangszustand durch NMI* per OUT (DFH),0FH zu unterbinden.
    Wenn dies gewünscht wird, müssen die Signale RESET* und NMI* anstelle von SYSRES* über Drahtverbindungen von den Pins 26 und 17 des Z80 an die entsprechenden Punkte auf der Anschlußleiste geführt werden. SYSRES* muß muß weiterhin an Punkt P' geführt werden (das berührt nicht die unter den Punkten 3), 7) und 9) getroffenen Aussagen).







3,4) Layout

5) Bestückungsplan

6) Drahtbrücken, die für alle Computer und Funktionsvarianten gelegt werden müssen

7) Anschlußpunkte für die Drahtbrücken zur Auswahl der Funktionen

Berücksichtigung bestimmter Systemkomponenten

Bei Vorhandensein eines Original-Expansion-Interface bei TRS80 und GENIE (EG 3014) müssen (in der voll ausgebauten Version dieses Switchboards) zum Expasion- Interface die Signale A14' und A15' anstelle von A14 und A15 geführt werden. Das Gleiche gilt auch für GENIE-Floppy-Laufwerke, die über den 50poligen Systembus angeschlossen sind (Controller im Laufwerksgehäuse).

Außerdem hat das Signal PHANTOM* bei der Adreßdecodierung des Drucker/Floppy- Bereichs bei diesen Interfaces keinen Einfluß. Um auch hier ein Ausblenden des I/O-Bereichs 37E0h-37FFH zu ermöglichen, müssen anstelle der Signale MREQ* beim GENIE bzw. MWR* und MRD* beim TRS80 die mit D7 (Pin 9 des 74LS259) OR-verknüpften Signale an das Expansion-Interface bzw. den Floppy-Controller im Laufwerksgehäuse weitergeleitet werden.

Dafür können die noch freien OR-Gatter in IC 16, 74LS32, Pins 4, 5 und 6 sowie Pins 10, 9 und 8 verwendet werden. Die Änderung muß über freie Verdrahtung auf der Lötseite erfolgen.

Bei Verwendung eines EXP1-Interface von RB-Elektronik in allen 3 Rechnern oder beim Komtek-Floppy-Interface (dessen RAMs wegen der 4164-RAMs auf dem CPU-Board nicht bestückt sein dürfen) sind keine zusätzlichen Maßnahmen erforderlich.

Änderungen auf dem CPU-Board

Um im Bedarfsfall auch ohne dieses Board arbeiten zu können, ist es sinnvoll, zwischen den Signalen 0-16K* vom CPU-Board und P0-16K'* auf diesem Board umschalten zu können. Dafür wird an den Mittelanschluß eines Umschalters die abgetrennte Leitung gelegt, an die vorher das Signal 0-16K* gelangte, während an die beiden äußeren Anschlüsse die Signale 0-16K* und P0-16K'* gelegt werden.

Beim GENIE liegt 0-16K* an Pin 12 von Z 25 (74LS139) und beim Komtek 1 an Pin 4 von IC 53 (74LS139).

Der TRS80 kommt dagegen ohne ein Signal 0-16K* aus. Erst bei der Umrüstung auf 64K RAM auf dem CPU-Board wird ein solches Signal erzeugt. Damit TRS80-Anwender dieses Board auch nutzen können, wird zum Schluß noch kurz der Umbau auf 64K auf dem CPU-Board beschrieben.

Umrüsten des TRS80 auf 64K RAM auf dem CPU-Board

1. Die Pins 1, 8 und 9 aller RAMs (Z14-Z20) von den Versorgungsspannungen +12V, -5V und +5V abtrennen.
2. +5V an die Pins 8 aller RAMs legen.
3. Alle Kondensatoren die an einer der die Pins 1 oder 9 verbindenden Leitungen liegt, auslöten.
4. A14 (Z38, 74LS367, Pin 11) an Pin 14 von Z51, 74LS157 legen.
5. A15 (Z38, 74LS367, Pin 9) an Pin 13 von Z51, 74LS157 legen.
6. Pin 12 von Z51, 74LS157 mit den Pins 9 der RAMs verbinden.
7. Die Leiterbahn von/zu folgenden Pins durchtrennen:
   Z21, 74LS156, Pins 2 und 14 sowie 1 und 15; Z73, 74LS32, Pin 5
8. Folgende Pins miteinander verbinden:
   Pin 5 von Z73, 74LS32 mit Pin 11 von Z38, 74LS367: A15 Pin 6 von Z73, 74LS32 mit Pins 1 und 15 von Z21, 74LS156: 0-16K* Pin 14 von Z68, 74LS367 mit Pins 2 und 14 von Z21, 74LS156: RAS*
9. Im Jumperfeld X3 müssen folgende Positionen miteinander verbunden sein: 2-15, 3-14, 4-13, 5-12, 6-11, 7-10, 8-9
10. Im Jumperfeld X71 müssen nur die Brücken 1-16, 3-14 und 5-12 gesetzt sein.
11. Im TRS80 Expasion-Interface muß die Verbindung zwischen Pin 6 von Z28, 74LS00 und Pin 19 von Z31, 74LS244 durchtrennt werden.

Belegung der 30poligen Anschlußleiste

1	IRQ*(NC)	11	A9	21	+5V
2	A0	12	A10	22	NMI*(SYSRES*)
3	A1	13	A11	23	RESET*(SYSRES*)
4	A2	14	A12	24	D3
5	A3	15	A13	25	D2
6	A4	16	A14	26	D1
7	A5	17	A15	27	D0
8	A6	18	MREQ*/NC)	28	GND
9	A7	19	RD*(IN*)	29	PHANTOM*
10	A8	20	WR*(OUT*)	30	P0-16K'*

Stückliste

1 Platine
1 50-(bei TRS80 40-) poliger Platinenrandstecker als Apple-Slot im Handel
1 Gegenstück zum Apple-Slot (beidseitig zu ätzen)
1 10µ/16V Elco
2 µ1 Keramik Kondensatoren
1 74LS159
1 74LS139
1 74LS86
1 74LS32
1 74LS11
2 74LS10
2 74LS08

1 73LS02
3 74LS00

Ver-16-fachung des Sonder-ROM Bereichs und Nutzung durch das Betriebssystem

Helmut Bernhardt (Hardware); Arnulf Sopp (Software)

Die übliche Technik, das Betriebssystem zu tunen, führt in der Regel dazu, daß durch Herunter-rutschen des HIMEM der Anwender-Speicher immer kümmerlicher wird, wobei anspruchsvollere Programme zwar eine Fülle zusätzlicher Featutes des Betriebssystems zur Verfügung haben (ob sie wollen oder nicht) und dafür eventuell zu wenig Speicherplatz für sich und ihre Daten/ Variablen haben.

Zwei Möglichkeiten, wie zusätzlicher Speicherplatz geschaffen werden kann, der nicht auf Kosten des Anwenders geht, sind in 1) und 2) beschrieben. Während die 1) beschriebene Methode (vollständige Decodierung der Tastatur 3800H-3BFFH und Bereitstellen von RAM im Bereich 3900H-3BFFH) mit allen Hard- und Software- Produkten zusammen funktioniert, ist die in 2) beschriebene Erweiterung an das Vorhandensein eines herkömmlichen Expansion-Interface EG 3024 mit Sockeln für 2 x 8 dynamische RAMs gebunden und die Vervielfachung des Sonder-ROM Bereichs ist in das ganze Projekt nur als Abfallprodukt mit eingebunden.

Hier soll nun beschrieben werden, wie man die Vervielfältigung des Sonder-ROM Bereichs und gleichzeitig dessen vollständige Decodierung, so daß auch der PUNCH Befehl des GENIE-Monitors (Maschinen-Programme auf Cassette schreiben) aus 3) funktioniert, durchführen kann, ohne das aufwendige Banking-Board zur Verwaltung zusätzlicher dynamischer RAMs aufbauen zu müssen.

Zunächst grob qualitativ das Prinzip der Schaltung:

In ein Portdecodiertes Latch (74116) wird ein 4-Bit-Wort als Nummer der einzuschaltenden Speicherbank eingetragen. Dieses 4-Bit-Wort liegt anschließend am Ausgang des Latch ständig an, bis ein anderes Wort eingetragen wird. Die Latch-Ausgänge werden an einen 1-aus-16-Decoder (74154) gelegt, der dann je nach Bitmuster an seinen 4 Eingangs-Pins den entsprechenden Ausgangs-Pin low-aktiv werden läßt, wenn seine beiden E*-Eingänge low aktiv sind. Wenn man an die E*-Eingänge das volldecodierte Freigabesignal 3000H-37DFH* für das Sonder-ROM legt, können die 16 Ausgänge des 74154 als Freigabesignale für insgesamt 16 Stück 2716-EPROMs benutzt werden.

Will man anstelle von EPROMs auch RAMS benutzen, kann man nicht einfach die Signale vom Sockel des Sonder-ROM an die 16 Speicherbausteine (2716-EPROMs oder 6116-RAMs) führen.

1. werden die am Sonder-ROM Sockel anliegenden Datenleitungen über unidirektionale Treiber nur in Leserichtung mit dem Systemdatenbus verbunden, weshalb ein Schreiben in RAMs so nicht möglich ist und
2. liegt an Pin 21 nicht das für RAMs nötige Signal WR* bzw. MWR* an. Dieser Pin ist einfach an +5V angeschlossen.
3. ist an den CE*-Pin 18 Masse angelegt, und ein nicht vollständig decodiertes Freigabesignal 12-14K* (=3000-E7FFH*) ist an den Pin 20 OE* geführt, der eigentlich ein RD* oder MDR*-Signal verdient hätte. Kollisionen mit den I/O-Adressen 37E0H-37FFH werden dadurch verhindert, daß durch das Signal 3600-37FF* an Pin 13 von Z21 (74LS20) die Freigabe des Speicherlesetreibers gesperrt ist. In diesem großzügigen Abwürgen des Sonder-ROM liegt auch die Ursache des nicht funktionierenden PUNCH-Befehls.

Um diesem Übel abzuhelfen, sind folgende Eingriffe nötig:

1. Die Datenleitungen zum Sockel des Sonder-ROM müssen durchtrennt werden. Die Datenpins des Sonder-ROM werden direkt mit den Ausgängen der Datenbustransceiver der CPU verbunden.
2. Die Datenleitung D7 die über den Sonder-ROM Sockel an den RAM-Baustein Z34 führt, muß zwischen RAM und Sonder-ROM durchtrennt werden. D7 für Z34 muß stattdessen vom ROM Z12 an das RAM geführt werden.
3. Die Leitungen zu den Pins 18 und 19 des Sonder-ROM Sockels (GND und +5V) werden durchtrennt.
4. An Pin 18 des Sockels wird MRD* und an Pin 21 MWR* gelegt.
5. Das Signal 3600H-37FFH wird von der Steuerung des Speicherlesetreibers abgetrennt. Stattdessen wird dort 12-14K* angelegt, sodaß der Speicherlesetreiber für den gesamten Bereich 3000H-37FFH gesperrt wird.
6. Durch ein Flachkabel und 2 24-polige Quetschstecker und ein in dieses Kabel eingefügtes Adapter-Board wird die 16-fach-Sonder-ROM Karte mit dem wie oben beschriebenen geänderten Sonder-ROM Sockel verbunden. Das Adapter-Board bringt nicht nur die Signale CE* und MRD* auf die richtigen Pins, sondern leistet auch die volle Decodierung auf den Bereich 3000H-37DFH und stellt einen bidirektionalen Datenbus-Treiber bereit, so daß auch RAMs benutzt werden können. (Abb.3).

Die Wahl eines 74116 Latches bietet den Vorteil, daß durch AND-Verknüpfung der Signale RESET* und NMI* (ergibt das beim TRS-80 bereits vorhandene Signal SYSRES*) und anlegen an den CLEAR-Eingang des Latch nach dem Einschalten oder Drücken des RESET-Knopfes immer die Bank 0 eingestellt ist. Wenn der bisherige Sonder-ROM auf der Ver-16-fachungs Karte durch den Ausgang 0 des 1-aus-16 Decoders 74154 freigegeben wird, verhält sich der Computer wie vorher auch. Wenn keine andere Bank eingeschaltet wird, liegt im Adreßbereich 3000H-37DFH immer der Sonder-ROM vor.

Arbeitsanleitung

1) Änderungen am Sonder-ROM Sockel auf dem CPU-Board

- Auf der Lötseite werden die Leitungen zu den Pins 18 und 21 des Sonder-ROM durchtrennt. An Pin 18 wird das Signal MRD* von Z15 (74LS32), Pin 11 und an Pin 21 das Signal MWR* von Z15 (74LS32), Pin 3 gelegt.
- Die Verbindung von Pin 14 des RAM Z34 mit Pin 17 des Sonder-ROM wird durchtrennt. Der Pin 14 des RAM wird stattdessen mit dem Pin 17 des ROM 3003 verbunden.
- Auf der Bestückungsseite werden (von der Mitte des CPU-Boards gezählt) die ersten 7 Leitungen zwischn ROM 3 (3003) und dem Sonder-ROM Sockel durchtrennt. Die 8. Leitung bleibt bestehen und die 9. Leitung wird ebenfalls durchtrennt. Dann werden mit isolierten Dähten auf der Lötseite folgende Verbindungen hergestellt.

Signal	Pin des	Verbindung zu		
	Sockels	Pin	IC	Typ
D0	9	7	Z18	74LS367
D1	10	9	"	"
D2	11	5	"	"
D3	13	13	Z19	"
D4	14	11	"	"
D5	15	3	Z18	"
D6	16	13	"	"
D7	17	11	"	"

2) Änderungen an der Speicherlesefreigabe auf dem CPU-Board

- Die Leitung zu Pin 13 von Z21 (74LD32) wird durchtrennt. Der Pin 13 von Z21 wird mit Pin 5 von Z22 (74LS156) verbunden (Abb.1).

3) Anschluß und Bestückung der Ver-16-fachungs Karte

- Auf dieser Karte werden ausschließlich 24-Pin IC-Sockel eingesetzt. An der im Layout beschrifteten Position wird der 1-aus-16-Decoder74154 eingesetzt. An eine der beiden Positionen daneben wird der Stecker des Flachkabels vom Sonder-ROM Sockel auf dem CPU-Board eingesetzt. Dieser Stecker muß (außer Pin 18) mit allen seinen Pins mit den entsprechenden Pins aller Speicher-Sockel verbunden werden.
- Da innerhalb einer Reihe diese Verbindungen per Layout schon bestehen, muß muß durch Drahtbrücken bzw. freie Verdrahtungen auf der Lötseite nur noch eine Querverbindung zwischen den entsprechenden Pins der 3 Reihen hergestellt werden.
- Die nicht angeschlossenen Pins 18 aller Sockel werden mit je einem der Ausgänge 0-9, A-F des 1-aus-16-Decoders durch freie Verdrahtung auf der Lötseite verbunden. Der Pin 18 des Sockels, an den das Flachkabel führt, wird an den E*-Eingang des Decoders gelegt. Die Eingänge 10-13 des Decoders werden mit den Ausgängen Q0-Q3 des Latch 74116 verbunden.
- Das Adapter-Board wird entsprechend Abb.7 u. 8 in das Flachkabel eingefügt. Dafür wird das Flachkabel an geeigneter Stelle durchgeschnitten, die Leitungen auf eine Länge von 2 cm voneinander getrennt und dann die Enden auf eine Länge von ca 3 mm abisoliert und die Enden schlank verdrillt und dünn eingezinnt.
- Wenn die Ver-16-fachungs Karte unter dem CPU-Board liegen soll und deshalb das Flachkabel nach einstecken in den Sonder-ROM Sockel vom CPU-Board wegführen soll, muß das Adapter-Board so orientiert werden, daß die Bestückungsseite zur gleichen Seite weist, wie die Stecker des Kabels. Die beiden in Kabelrichtung liegenden ICs (74LS00 und 74LS30) liegen dann links außen im Computer.
- Im Gehäuse des Computers muß ein entsprechender Pfosten der Bodenwanne abgesägt werden, damit das Board in das Gerät paßt.

4) Aufbau und Anschluß des Latch-Boards

- Fast alle Signale für dieses kleine Board können vom Sockel des Sonder-ROM (nicht von der darüber angeschlossenen Speicherkarte) abgenommen werden:
-

Signal	Anschlußpunkt Latch-Board	Pin Nr. am Sonder-ROM	
+5V	1	24	
WR*	2		#
IORQ*	3		#
A0	4	8	
A1	5	7	
A2	6	6	
A3	7	5	
A4	8	4	
A5	9	3	
A6	10	2	
A7	11	1	
SYSRES*	12		#
D0	13	9	
D1	14	10	
D2	15	11	
D3	16	13	
GND	17	12	

siehe Text

- Die Signale WR* und IORQ* werden an den Pins 11 und 9 von Z16 (74LS367) abgenommen und an die Latch-Board Anschlüsse 2 und 3 gelegt.
- Das Signal SYSRES* für den Anschluß 12 des Latch-Boards muß auf dem CPU-Board noch gebildet werden. Dafür muß in die Position Z26 (neben Z25, 74LS139) noch ein 74LS08 eingesetzt werden. Da diese Position für ein DIL16-IC vorbereitet ist, muß je nach Einbau noch eine Verbindung mit +5V oder GND (am Lötauge außerhalb des ICs hergestellt werden. Selbstverständlich muß das 74LS08 in gleicher Orientierung (Kerbe nach links) wie alle anderen ICs eingebaut werden.
- Nun wird Pin 1 des 74LS86 mit Pin 26 der Z80-CPU und Pin 2 mit Pin 17 der Z80-CPU verbunden. Der Pin 3 des 74LS08 wird an den Anschluß 12 des Latch-Boards geführt.
- Von den Anschlüssen 18 bis 21 des Latch-Boards werden Verbindungen zu den Pins 20 bis 23 des 1-aus-16-Decoders auf der Ver-16-fachungs Karte hergestellt.
-

Latch-Board		**Ver-16-fachungs Karte**	
Anschluß	**Signal**	**Pin des 74154**	**Signal<**
18	Q3	23	13
19	Q2	22	12
20	Q1	21	11
21	Q0	20	10

- Für eventuelle spätere Service-Arbeiten ist es sinnvoll, wenn einzelne Boards nicht über Drähte direkt miteinander verlötet werden. Deshalb sollten an den vom Layout neben dem Decoder vorgesehenen Stellen Lötnägel eingesetzt werden.

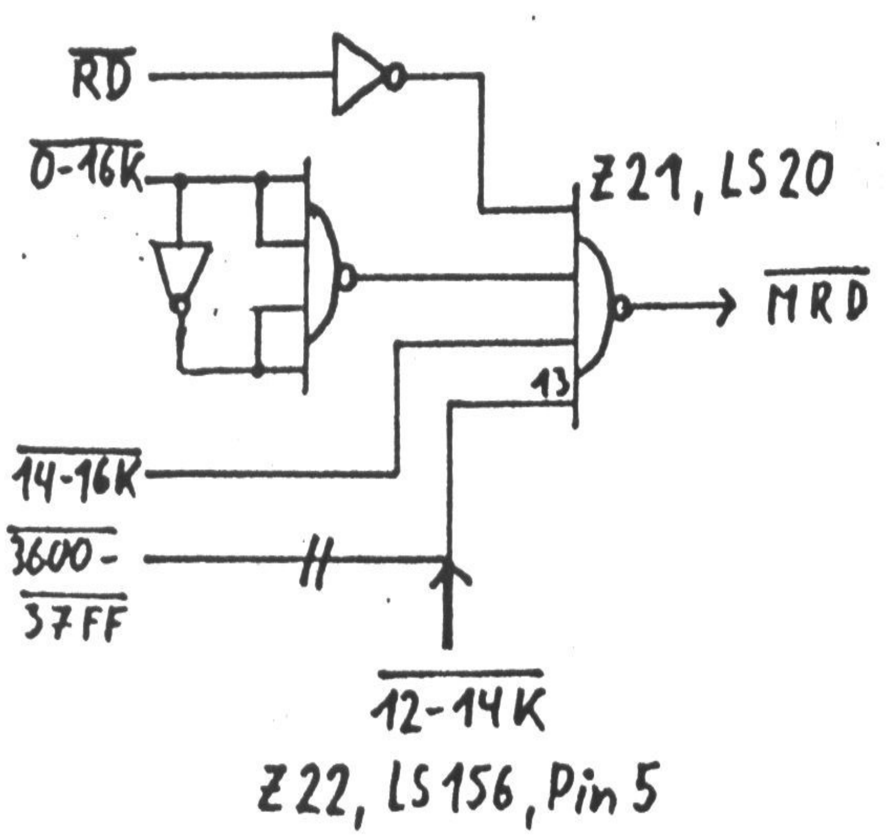


Abb.1: Freigabe des Speicher-lese- Treibers auf dem CPU- Board für den Be-trieb der Ver-16-fach--ungs-Karte

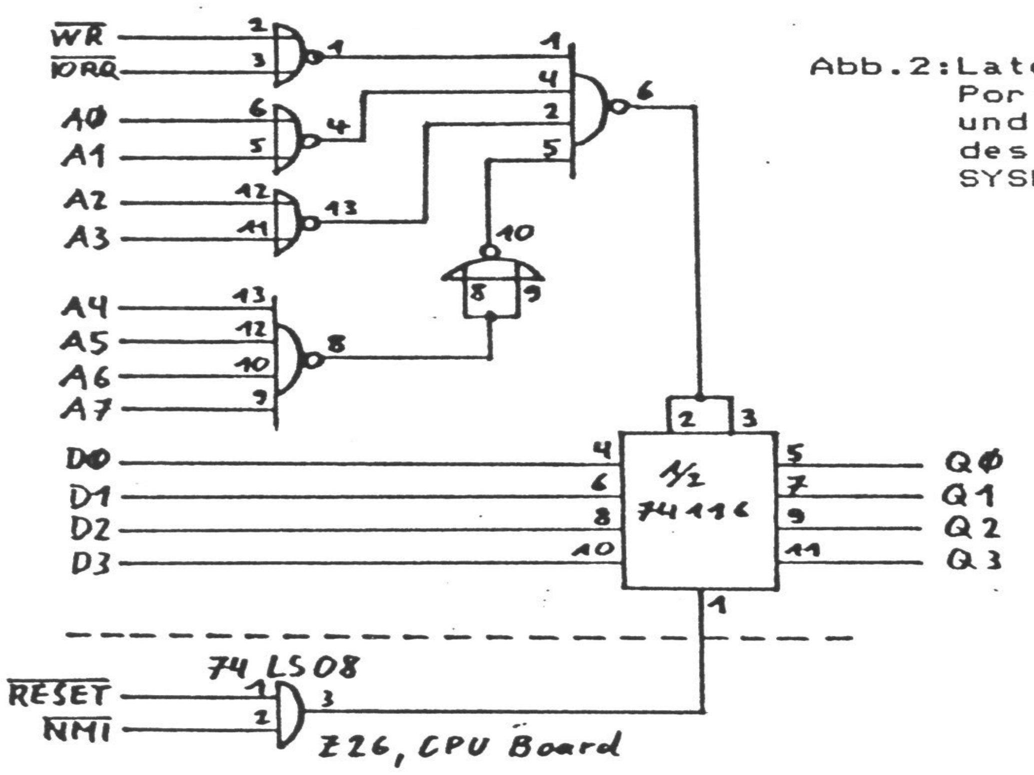


Abb.2: Latch-Board mit Portdecodierung und Erzeugung des Signals SYSRES*

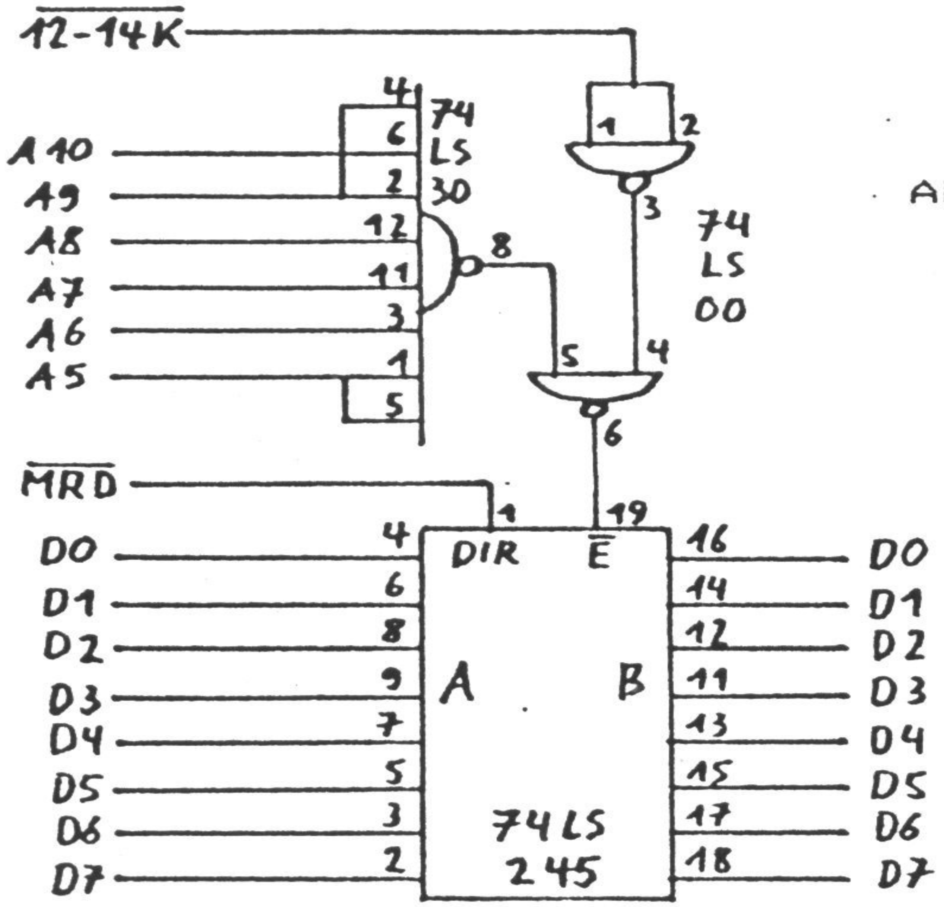


Abb.3: Adapter-Board mit vollständiger Adreßdecodierung und Datenbus-Transceiver

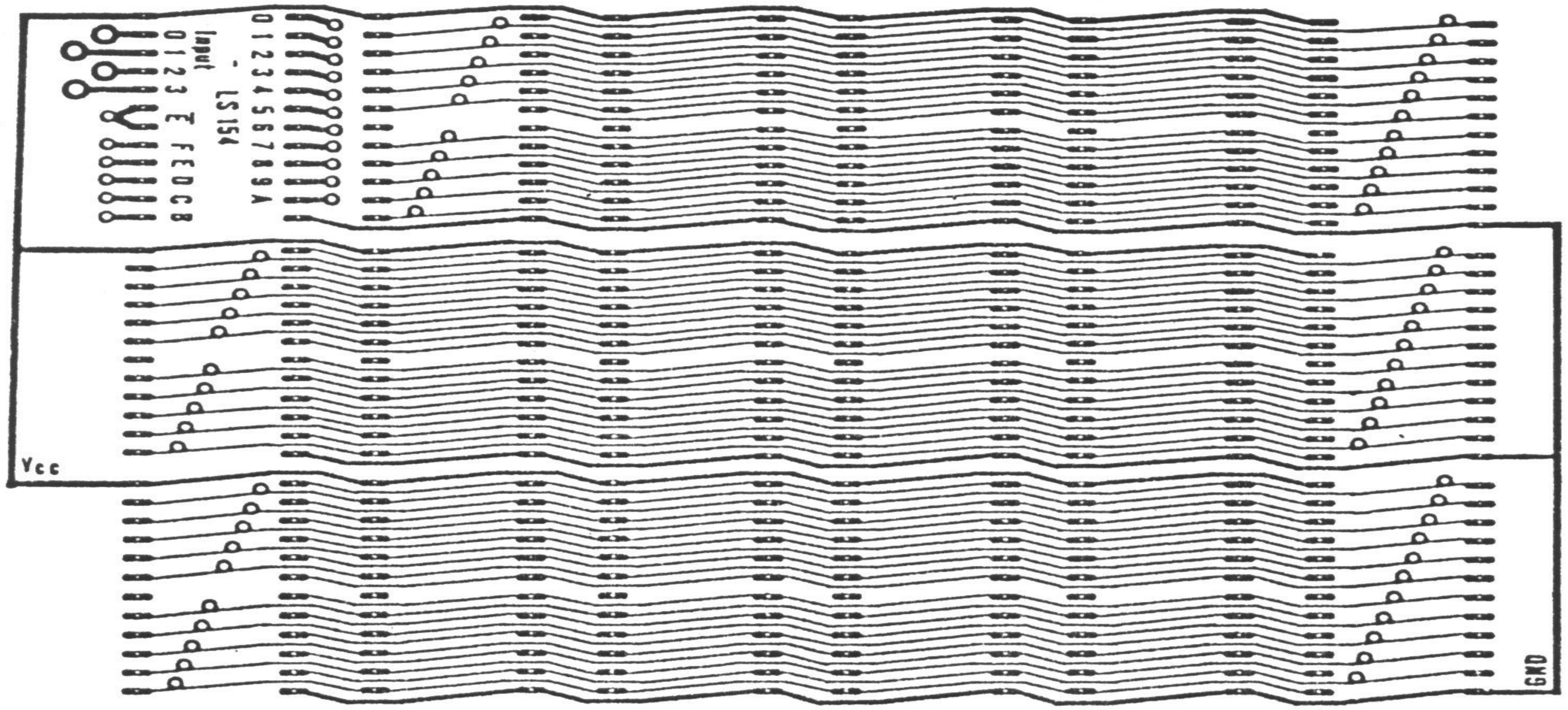


Abb.4:Layout des Ver-16-fachungs-Boards
(Ansicht Lötseite)

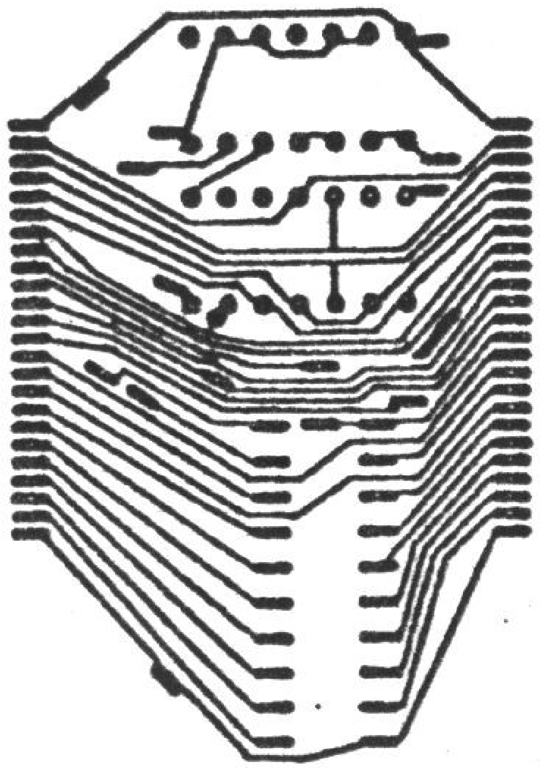


Abb.5:Layout des Adapter-Boards
(Ansicht Lötseite)

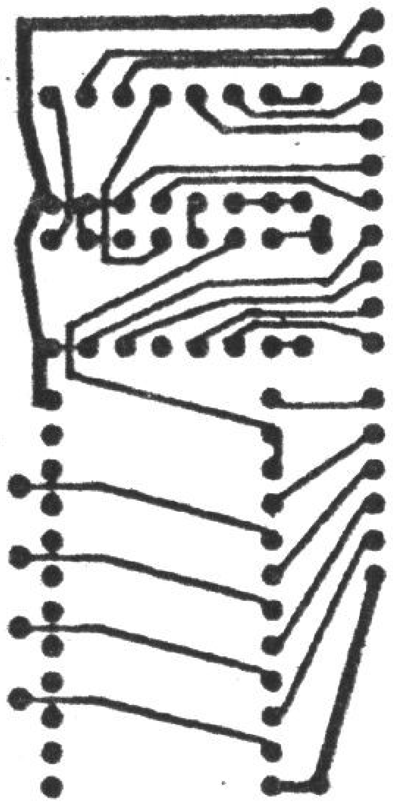


Abb.6:Layout des Latch-Boards
(Ansicht Lötseite)

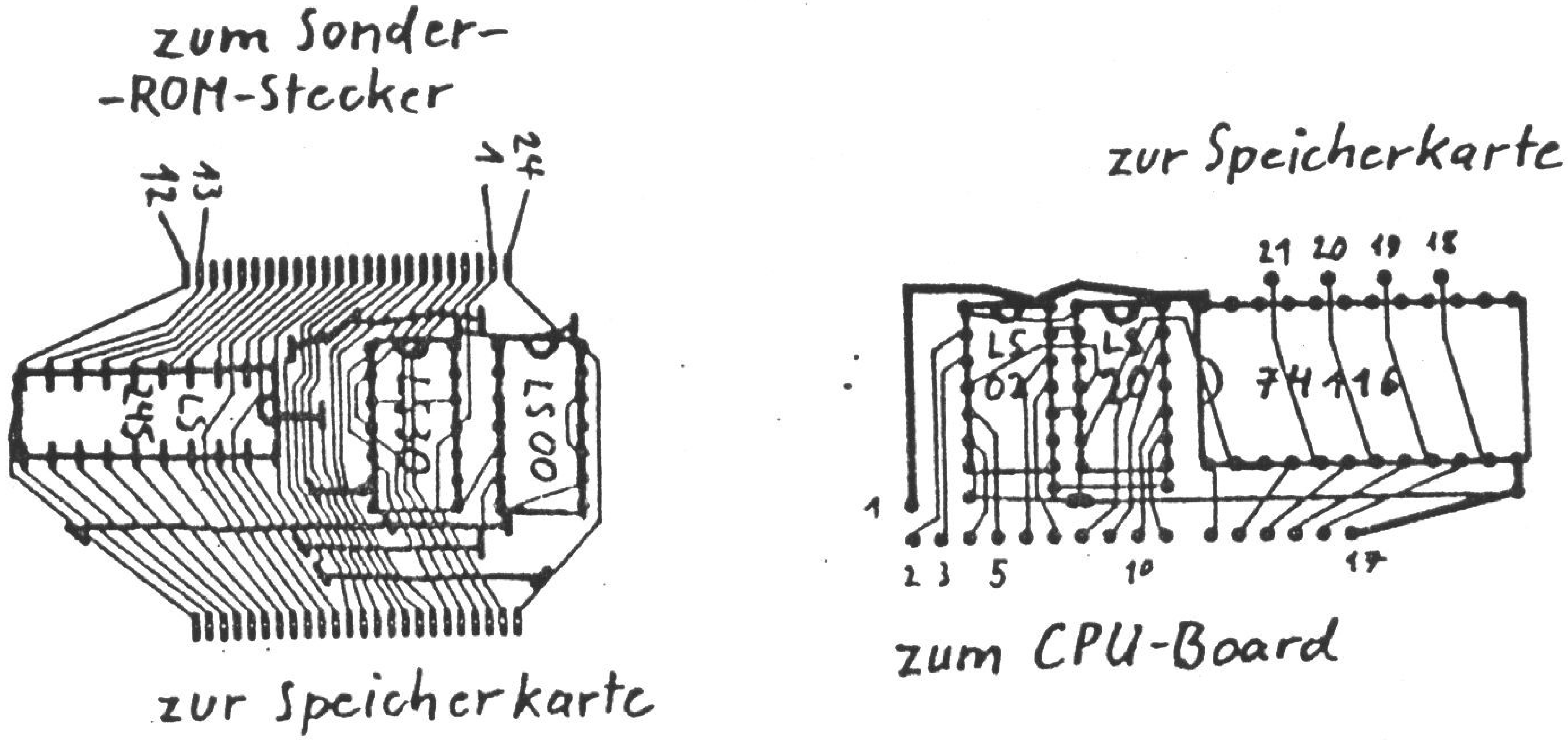


Abb.7:Bestückung und Anschlüsse des Adapter-Boards und des Latch-Boards

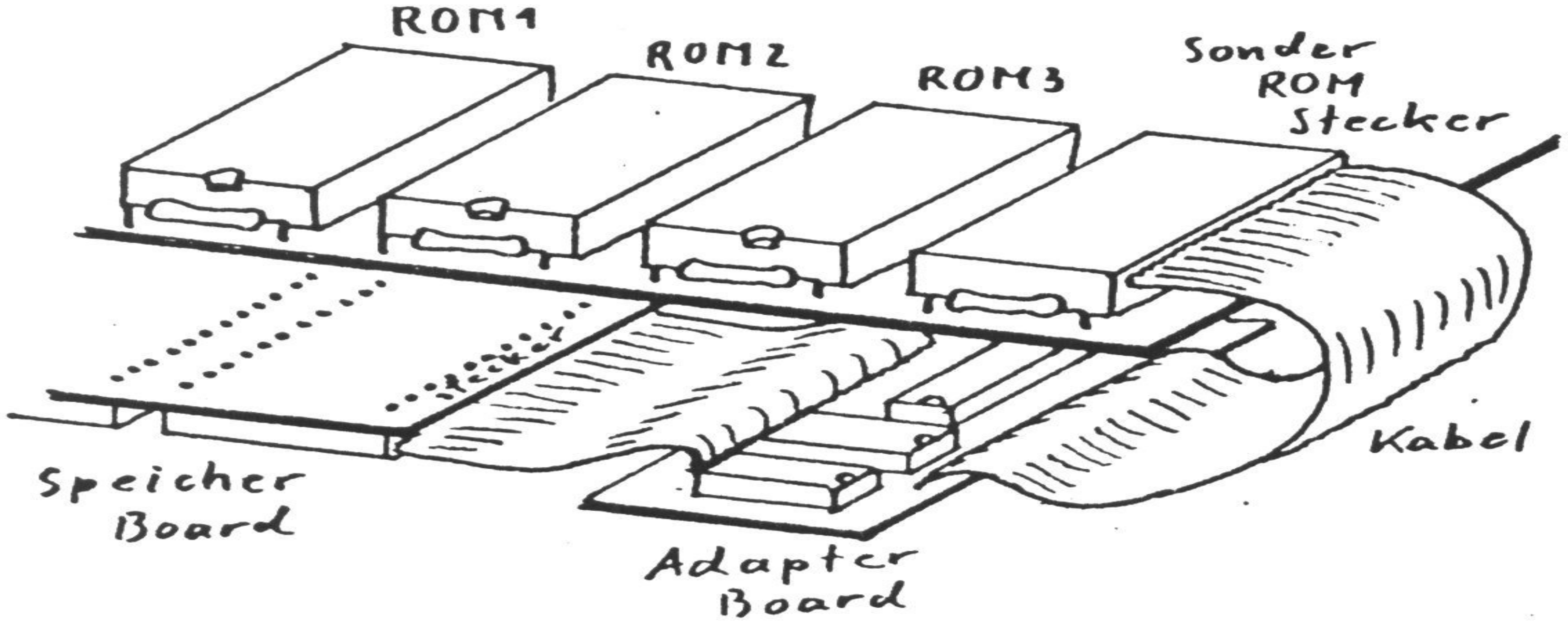


Abb.8:Orientierung beim Zusammenbau von CPU-Board, Adapter-Board und Ver--16-fachungs-Board

... und dazu etwas Software

Arnulf Sopp

Wie nützlich ein Umbau am Computer ist, zeigt sich natürlich erst, wenn er etwas zu tun bekommt. Das folgende Programm ist nur ein Beispiel, was man mit dem Versechzehnfacher anfangen kann. Jedes andere Maschinenprogramm, sogar BASIC-, Pascal- oder sonstige höhere Programmtexte können dort untergebracht werden (1,2), natürlich auch Datenbytes für Tabellen und dergleichen. Damit ist das leidige Himem-Thema von Tisch; nahezu jede popelige BASIC-Verschönerung (eingerückte Zeilennummern und ähnlich überflüssiges) drängelt sich dort oben und verschleißt den Platz für wirklich Wichtiges.

Dieses Programm ist ein Treiber für die HRG 1b von RB-Elektronik. Er lädt selbsttätig BASIC/CMD und arbeitet anschließend im Hintergrund mit BASIC zusammen. Sein Ladebereich geht von 3000-32E4. Dahinter sind noch ein paar Bytes als Datenpuffer reserviert. Um beim Laden des Treibers nicht den DOS-Fehler "Ladeversuch auf ROM-Speicherplatz" serviert zu bekommen, muß dort natürlich RAM vorhanden sein. Der User gibt deshalb zuvor auf den Port F0 die Nummer eines Steckplatzes auf dem Versechzehnfacher aus, wo ein statisches RAM sitzt. Unter H-DOS geht das mit OUT F0,xx (7). Ansonsten kann man zunächst BASIC starten und dann OUT 240,xx anschließend CMD"S=HRG" eingeben (sofern man dieses Programm unter dem Namen HRG/CMD auf Disk stehen hat).

A propos H-DOS: Da der Befehl CMD > (mehr darüber weiter unten) das Parallel-RAM auf der CPU-Platine benutzt, müssen Besitzer des EG 64 MBA mit diesem DOS zuvor INI.N eingeben. Damit werden zwar die interessantesten Features von H-DOS disabled, aber noch immer hat der User alle G-DOS-Möglichkeiten und noch ein bißchen mehr. Auf allen Maschinen, mit denen Banking nicht möglich ist, funktioniert dieser Befehl ganz einfach nicht, ohne aber Schaden anzurichten oder zu einem Fehler zu führen.

Der Treiber stellt eine Reihe von Kommandos an die HRG zur Verfügung. Sie alle beginnen mit CMD (1). Um dem Interpreter anzuzeigen, daß es sich nicht um das gewohnte CMD handelt, folgt unmittelbar darauf nicht eine Variable oder ein Ausdruck zwischen Gänsefüßchen, sondern ein zweites BASIC-Token:

```
CMD +           : HRG einschalten
CMD -           : HRG ausschalten
CMD CLS         : HRG-Speicher löschen
CMD SET         : einen HRG-Punkt setzen
CMD RESET       : einen HRG-Punkt löschen
CMD POINT       : testen, ob ein HRG-Punkt gesetzt ist
CMD <           : HRG invertieren (weiß <-> schwarz)
CMD LPRINT      : HRG-Inhalt auf Drucker ausgeben (Hardcopy)
CMD ASC         : ASCII-Bildschirm in Hex anzeigen, bis neue Taste gedrückt
CMD >           : Bildschirm gegen Pufferinhalt austauschen
CMD LINE        : noch nicht programmiert, ergibt aber keinen Fehler
```

Nach der üblichen BASIC-Syntax können CMD und das zweite Token direkt oder durch Blank getrennt hintereinander stehen. Diese Befehle können im Direktmodus und als Bestandteil eines Programms eingegeben werden.

Die drei erstgenannten HRG-Befehle bedürfen keiner näheren Erläuterung. CMD SET, CMD RESET und CMD POINT haben eine ganz ähnliche Syntax wie die gewohnten Befehle zur Ansteuerung der Genie-Pixelgraphik. Im Argument sind im Unterschied dazu jedoch Koordinaten im Bereich 0-383 für X und 0-191 für Y zulässig, entsprechend der feineren HRG-Matrix. Die Koordinaten können wie in Level 2 oder Disk-BASIC auch durch Variable dargestellt werden.

CMD < macht aus jedem Punkt im HRG-Speicher sein Komplement. Gesetzte Punkte werden dunkel, nicht gesetzte werden eingeschaltet. Es entsteht eine negative Darstellung. Der normale ASCII-Bildschirm bleibt hiervon unberührt.

Bei einem Bildschirmausdruck der HRG mit CMD LPRINT wird der Drucker zunächst in den Einschaltzustand versetzt (Reset aller Druckparameter). Danach wird der linke Rand auf die 9. Stelle gesetzt, um das Bild zu zentrieren. Der Zeilenabstand wird mit 6/72" so eingestellt, daß die Druckzeilen lückenlos untereinander stehen. Nach getaner Arbeit werden diese Steuercodes alle wieder gelöscht; der Drucker ist wieder im Einschaltzustand. Diese Codes gelten für den Star Gemini-10X. Für andere Printer müssen sie entsprechend geändert werden. Von den vergleichbaren Epson-Typen ist mir bekannt, daß für die Randeinstellung lediglich in ESC M 09 aus dem "M" ein "1" gemacht zu werden braucht. Die übrigen Codes sind identisch.

Der Befehl CMD ASC ist für vielerlei Verwendungen interessant. Mit ihm hat der Programmierer sozusagen immer eine ASCII-Tabelle auf dem Bildschirm. Die ASCII-Zeichen werden als zweistellige Hexzahlen dargestellt (4). So werden gleiche Zeichen mit verschiedenen Codes unterscheidbar, um nur eine Anwendung zu nennen: Die beiden Blanks mit den Codes 32 und 128 sowie die Graphikblocks 128-191 und 192-255 sind als Hexzahlen eindeutig zu identifizieren. Bei der Ausführung von CMD ASC werden die 32er Blanks übrigens nicht mit umgewandelt, sondern als Leerzeichen belassen, um den Bildschirm übersichtlich zu halten.

Dabei wird der normale ASCII-Bildschirm natürlich gelöscht, sonst würde er stören. Der Treiber rettet ihn in den HRG-Speicher, so daß -er nach dem Druck auf irgendeine Taste wieder restauriert werden kann. Ein besonderer Puffer im RAM ist dafür nicht erforderlich.

Nach diesem Schema geht auch der Befehl CMD > vor. Zunächst wird der Bildschirm in die HRG gerettet. Von ihr sind nämlich nur 6 Bits pro Byte sichtbar, die beiden höchstwertigen Bits

sind unbenutzt und können den Bildschirm aufnehmen. Dabei wird der sichtbare Inhalt des HRG-Speichers nicht beeinträchtigt. Alle Graphiken bleiben also erhalten. Im weiteren Verlauf der Ausführung dieses Befehls wird nun der Inhalt der HRG, also Graphik und ASCII, in einen Puffer gerettet, und der Inhalt dieses Puffers geht dafür in die HRG.

Die beiden höchstwertigen Bits des neuen HRG-Inhalts werden in den ASCII-Bildschirm übertragen. Es ist demnach ein Austausch Bildschirm gegen Puffer. In diesem Treiber liegt der Puffer im Adreßbereich ab 0000. Das ist natürlich nur mit dem EG 64 MBA oder einem anderen Banker realisierbar.

Entsprechende Änderungen, um stattdessen das gute alte Himem dafür zu benutzen, sind kein Problem.

Der Befehl CMD LINE soll eines fernen Tages, wenn der Autor mal wieder Lust hat, an dem Treiber weiterzuarbeiten, Linien, Rechtecke usw. ziehen. Der Leser möge mir nachsehen, daß die Computerei mein Hobby ist, bei dem ich gnadenlos dem Lustprinzip folge (6).

An diesem fernen Tage werden wohl noch ein paar andere Modifikationen folgen, die es ermöglichen. den HRG-Treiber fest in ein EPROM zu brennen. Das ist mit dieser Version nicht möglich, denn das Programm vollzieht Schreibzugriffe in seinem eigenen Adreßraum. Dazu gehören z. B. die Datenpuffer an seinem Ende, aber auch variable Sprungdistanzen mitten im Programm (Selbstmodifikation). Der Vorteil dieser Variante ist freilich, daß der Treiber mit den Bedürfnissen jederzeit mitwachsen kann. Immerhin sind noch eineinviertel kB frei!

Auf die Programmlogik möchte ich an dieser Stelle nicht eingehen. Die siebeneinhalb Seiten Listing wären halbwegs erschöpfend wohl nicht mit weniger als 15 Seiten Test erklärbar. Insbesondere diejenigen Features, die kein anderer mir bekannter HRG-Treiber zur Verfügung stellt, würden einige Seiten füllen: Z. B. die Umwandlung des ASCII-Bildschirms in die Hexdarstellung und das Puffern desselben im HRG-Speicher. Der Maschinensprache-Freak mag sich dabei amüsieren, sich in dem sehr ausführlich kommentierten Listing selber zurechtzufinden.

Zu diesem Thema sind an gleicher Stelle auch bereits ein paar Beiträge erschienen, die die Arbeit deutlich erleichtern werden (5 u.a.).

Nur einige wenige Kommentare, ohne die sich sogar der Autor kaum noch im Programm zurechtfindet, sollen angefügt sein: In BASIC/CMD steht nach dem Laden (also noch vor dem Ansprung!) an 67DD der CMD-Vektor (JP 57FF). Er wird durch JP cmd (3074) ersetzt. Andere Treiber benutzen kein bereits vorhandenes Token und verlangsamen damit den ganzen Computer: es muß bei jeder Eingabe geprüft werden, ob ein vorangestelltes Doppelkreuz o. dergl. ein Syntaxfehler oder die Einleitung eines HRG-Befehls ist. An der Stelle 64F3 wird HL als Zeiger auf das BASIC-Helle geladen. Mit der Änderung des LSB von 5C auf 5E werden die beiden ersten Codes, die den Bildschirm löschen, übergangen. So bleibt die oberste Titelzeile erhalten, die beim Label hello steht.

Beim späteren Programmablauf wird der Treiber nur dann angesprungen, wenn das CMD-Token angetroffen wurde. Nun prüft er, was folgt. In Disk-BASIC ist es immer ein Anführungszeichen oder ein Variablenname. Diese Zeichen liegen im Bereich bis 7F. Der Vergleich mit 80 beim Label cmd gibt deshalb Aufschluß, ob zur Ansteuerung der HRG ein weiteres Token folgt. Falls nein, geht es in der alten CMD-Routine weiter, falls ja, kommt der Treiber jetzt so richtig in Fahrt.

Anwender, die mit EDTASM/CMD oder ASM/CMD arbeiten. müssen auf die für diese Assembler zulässige Syntax achten; Labels dürfen höchstens 6 Zeichen lang sein, Kleinbuchstaben sind nur in den Kommentaren erlaubt, DB, DW und DM müssen als DEFB, DEFW und DEFM ausgeschrieben werden.

Bei diesen Statements dürfen auch nicht mehrere Argumente in derselben Zeile stehen, schon gar nicht DEFM und DEFB gemischt. Je eine eigene Zeile muß dafür eingerichtet werden.

Dieser HRG-Treiber ist, wie gesagt, lediglich eine Demonstration dafür, was mit Helmut Bernhardts Versechzehnfacher so alles anzufangen ist. Die Platine funktioniert selbstverständlich ohne HRG, auch ohne DOS ebenso gut wie mit. Vielleicht fällt einem von euch für eins der nächsten Infos eine sinnvolle Verwendung für Level 2 ein?

Literatur:

1) "PUT TO - ein neuer BASIC-Befehl", 4/1984
2) "Bank Selection mit dem Genie I", 5/1985
3) "Neuer DOS-Befehl: OUT port#,xx,yy,...", 11/1984
4) "VIDHEX - Hexanzeige des Bildschirms mit der HRG", 9/1985
5) "Die HRG 1b programmieren", 11/1984

(alle von A. Sopp im Club-Info des Genie/TRS80-User-Club Bremerhaven, Ausgabe jeweils angegeben).

6) R. Goscinny, M. Uderzo: "Asterix und der Kupferkessel", Dargaud S. A. 1969


```
                ;
                ;
                ; BASIC-Treiber für die HRG 1b von RB-Elektronik
                ;
                ; Demo-Programm zum Speicher-Versechzehnfacher
                ; im Adreßraum 3000-370F von Helmut Bernhardt
                ;
                ;       (C)     Arnulf Sopp
                ;
                ;

3000                    ORG     3000h           ;im Sonder-ROM-Bereich
```

```
                 ;DOS-Eingabepuffer retten
3000    11E432   finit   LD      DE,cmdbuf       ;Puffer für Aufrufbefehl
3003    014800           LD      BC,0048h        ;ca. Lng. DOS-Eing.-puff.
3006    E5               PUSH    HL              ;Aufrufbefehl retten,
3007    D5               PUSH    DE              ;weil 4419 den Puffer-
3008    C5               PUSH    BC              ;Inhalt verändert
3009    EDB0             LDIR                    ;zwischenspeichern

                 ;Initialisierungen in BASIC/CMD
3008    212C30           LD      HL,command      ;BASIC-Ladekommando
300E    CD1944           CALL    4419h           ;BASIC/CMD laden
3011    217430           LD      HL,cmd          ;neuer CMD-Vektor
3014    22DD67           LD      (67ddh),HL      ;in BASIC/CMD laden
3017    3E5E             LD      A,5eh           ;LSB des BASIC-Logo
3019    32F464           LD      (64f4h),A       ;CLS verhindern
301C    213830           LD      HL,hello        ;HRG-Logo
301F    CD6744           CALL    4467h           ;anzeigen
3022    C1               POP     BC              ;Länge Eingabepuffer
3023    E1               POP     HL              ;Aufrufbefehl
3024    D1               POP     DE              ;Adresse Eingabepuffer
3025    D5               PUSH    DE              ;brauchen wir gleich
3026    EDB0             LDIR                    ;Befehl restaurieren
3028    E1               POP     HL              ;Befehlserweit. oder CR
3029    C3BE66           JP      66beh           ;BASIC-Kaltstart

                 ;DOS-Befehl, um BASIC in den Speicher zu laden
302C    6C       command DM      'load,basic/cmd',0dh

                 ;neue 1. Zeile des BASIC-Hello
3038    1C       hello   DM      1ch,1fh,'H R G  1 b - Utility von       '
3056    54               DM      'The HACKTORY Arnulf Sopp für',Oah,0dh

                 ;BASIC-Erweiterung für HR-Graphikbefehle
3074    FE80     cmd     CP      80h             ;Graphik-CMD-Befehl?
3076    DAFF57           JP      C,57ffh         ;norm. weiter, falls nein

                 ;Es ist ein HRG-Befehl. Erkennen und ausführen:
3079    FED4             CP      0d4h            ;Bildschirm austauschen?
307B    2029             JR      NZ,CLS          ;falls nein

                 ;CMD >: Bildschirm gegen Pufferinhalt austauschen
307D    E5               PUSH    HL              ;Befehlszeiger
307E    CD5332           CALL    vidsav          ;Bildschirm retten
3081    F3               DI                      ;bloß keine Störungen!
3082    3E08             LD      A,8             ;RD RAM 0000-2FFF
3084    D3DF             OUT     (0dfh),A        ;auf Banking-Port ausg.
3086    3C               INC     A               ;WR RAM 0000-2FFF
3087    D3DF             OUT     (0dfh),A        ;dto.
3089    65               LD      H,L             ;HL <- 0000, Pufferadr.
308A    55               LD      D,L             ;DE <- 00xx, HRG-Adresse
3088    50               LD      E,L             ;DE <- 0000
308C    0E30             LD      C,30h           ;Konst. HRG-MSB
308E    CD4B32   swaplop CALL    HRGadr          ;HRG-Stelle adressieren
3091    DB04             IN      A,(4)           ;HRG-Byte laden
3093    46               LD      B,(HL)          ;Pufferbyte holen
3094    77               LD      (HL),A          ;HRG-Byte puffern
3095    78               LD      A.B             ;Pufferbyte
3096    0305             OUT     (5),A           ;auf HRG ausgeben
3098    23               INC     HL              ;Pufferzeiger erhöhen
3099    7A               LD      A,D             ;überprüfen, ob
309A    89               CP      C               ;HkG-Bereich überschr.?
3098    38F1             JR      C,swaplop       ;falls nein
309D    DBDF             IN      A,(0dfh)        ;Banks rücksetzen
309F    FBI              EI                      ;INTs wieder zulassen
30A0    CD5532           CALL    savrest         ;Bildsch.       restaurieren
30A3    E1               POP     HL              ;Befehlszeiger
30A4    1831             JR      exit1           ;fertig

                 ;CMD CLS: HRG-Speicher löschen
30A6    FE84     CLS     CP      84h             ;CLS?
```

```
30A8    2004            JR      NZ,INV          ;falls nein
30AA    1E00    clear   LD      E,0             ;Flag für HRG-CLS
30AC    1806            JR      clsinv          ;dort weiter

                ;CMD <: HRG invertieren (positiv -> negativ)
30AE    FED6    INV     CP      Od6h            ;invertieren?
3080    2027            JR      NZ,LINE         ;falls nein
3082    1EFF            LD      E,0ffh          ;Flag für Inversion
3084    010330  clsinv  LD      BC,3003h        ;8 = MSB 12kB, C = Port 3
3087    C5      msloop  PUSH    BC              ;retten
30B8    05              DEC                     ;B um 1 zu hoch
30B9    ED41            OUT     (C),B           ;MSB HRG-Adresse
3088    0600            LD      B,0             ;B <- 0 für HRG-LSB
30BD    0D              DEC     C               ;Port 2 für LSB
308E    ED41    lsloop  OUT     (C),B           ;LSB HRG-Adresse
3000    DB04            IN      A,(4)           ;HRG-Byte holen
30C2    F5              PUSH    AF              ;retten
30C3    E6C0            AND     0c0h            ;obere Bits isolieren
3005    57              LD      D,A             ;und retten
3006    F1              POP     AF              ;HRG-Byte
3007    2F              CPL                     ;invertieren
3008    E63F            AND     3fh             ;obere Bits ausmaskieren
30CA    CB7B            BIT     7,E             ;CLS oder INV?
30CC    2001            JR      NZ,outa         ;falls INV
30CE    AF              XOR     A               ;sonst löschen
30CF    82      outa    OR      D               ;obere Bits hinzufügen
3000    0305            OUT     (5),A           ;Blank od. Kompl. ausgeb.
3002    10EA            DJNZ    lsloop          ;LSB: 00, FF, FE ... 01
3004    C1              POF     BC              ;MSB und Port für MSB
3005    10E0            DJNZ    msloop          ;MSB-1: 2F ... 00

                ;Rückkehr für mehrere Unterprogramme
3007    23      exit1   INC     HL              ;Befehlszeiger nachst.
3008    C9              RET                     ;erledigt

                ;CMD LINE:      (bisher nicht programmiert)
3009    FE9C    LINE    CP      9ch             ;LINE?
3008    28FA            JR      Z,exit1         ;vorl. nicht implement.

                ;CMD 4: HRG-Speicher in den Bildschirm einblenden
3000    FECD            CP      Ocdh            ;HRG einschalten?
30DF    2004            JR      NZ,HRGoff       ;falls nein
30E1    0301            OUT     (1),A           ;einschalten
30E3    18F2            JR      exit1           ;fertig

                ;CMD -  HRG-Speicher aus dem Bildschirm ausblenden
30E5    FECE    HRGoff  CP      Oceh            ;HRG ausschalten?
30E7    2004            JR      NZ,LPRINT       ;falls nein
30E9    0300            OUT     (0),A           ;ausschalten
30EB    18EA            JR      exit1           ;fertig
30ED    FEAF
30EF    2066    ;CMD LPRINT: Hardcopy des HRG-Speichers
                LPRINT    CP      0afh            ;LPRINT (Hardcopy)?
                          JR      NZ,ASC          ;falls nein

                ;LPRINT;  Drucker initial.: Reset, 6/72" Zeilenabstand
30F1    E5                PUSH    HL              ;Befehlszeiger retten
30F2    210632            LD      HL,prestor+1    ;Druckerinitialisierung
30F5    0609              LD      B,9             ;mit 9 Codes
30F7    CDFC31            CALL    lprint          ;Ausgabe auf Drucker

                ;HRG-Zeiger laden, 16 (Doppel-)Zeilen vorbereiten
30FA    50                LD      D,B             ;DE <- 00xx
30FB    58                LD      E,B             ;DE <- 0000, Start HRG
30FC    010610            LD      BC,1006h        ;16 Zeilen, Konst. 6

                ;1 Doppelzeile zu je 6 Dotzeilen drucken
30FF    C5      scrnlop   PUSH    BC              ;Zähler retten
3100    D5                PUSH    DE              ;dto. HRG-Zeiger
3101    0602              LD      B,2             ;2 Halbzeilen/Zeile
3103    C5      linelop PUSH      BC              ;wird verändert
```

```
3104    D5               PUSH    DE              ;dto. HRG-Zeiger
3105    21DF32           LD      HL,lninit       ;Druckerinit. für 1 Zeile
310B    0605             LD      B,5             ;mit 5 Codes
310A    CDFC31           CALL    lprint          ;Ausgabe
3100    21E432           LD      HL,buffer       ;Puffer für Druckercodes
3110    0640             LD      B,40h           ;64 Bytes/Zeile

               ;1 Halbzeile drucken
3112    C5       hlinlop PUSH    BC              ;Zähler retten
3113    D5               PUSH    DE              ;dto. HRG-Zeiger
3114    78               LD      A,E             ;LSB des HRG-Zeigers
3115    D302             OUT     (2),A           ;auf HRG ausgeben
3117    41               LD      B,C             ;6 Bytes senkrecht

               :1 Byte drucken
3118    C5       bytelop PUSH    BC              ;wird verändert
3119    7A               LD      A,D             ;MSB des HRG-Zeigers
311A    0303             OUT     (3);A           ;auf HRG ausgeben
311C    DB04             IN      A,(4)           ;Dotzeile aus HRG
311E    41               LD      B,C             ;6 Dots/Stelle
311F    E5               PUSH    HL              ;retten

               ;6mal 1 Bit errechnen
3120    0F       bitlop  RRCA                    ;Cy <- HRG-Bit
3121    CB16             RL      (HL)            ;nächstes Pufferbit <- Cy
3123    CBB6             RES     6,(HL)          ;nur untere Bits
3125    23               INC     HL              ;nächstes Pufferbyte
3126    10F8             DJNZ    bitlop          ;bis 6 Dots gepuffert

               ;6 Bits fertig - 6 Bytes vervollständigen
3128    14               INC     D               ;MSB auf nächste Dotzeile
3129    14               INC     D               ;= um 1 kB erhöhen)
312A    14               INC     D
3120    14               INC     D
312C    E1               POP     HL              ;Pufferzeiger
312D    C1               POP     BC              ;Zähler
312E    10E8             DJNZ    bytelop         ;bis 1 Stelle im Puffer

               ;6 Bytes ausdrucken und weiter mit Halbzeile
3130    E5               PUSH    HL              ;wird verändert
3131    41               LD      B.0             ;6 Dotspalten
3132    CDFC31           CALL    lprint          ;ausgeben
3135    E1               POP     HL              ;Pufferzeiger
3136    D1               POP     DE              ;alter HRG-Zeiger
3137    13               INC     DE              ;nächstes HRG-Byte
3138    C1               POP     BC              ;Halbzeilenzähler
3139    1007             DJNZ    hlinlop         ;bis 1 Halbzeile gedruckt

               ;Ganzzeile vervollständigen
3138    D1               POP     DE              ;HRG-Zeiger
31330   3E18             LD      A.18h           ;6 Dotzeilen (6 kB)
313E    B2               ADD     A,D             ;zum MSB addieren
313F    57               LD      D,A             ;neu laden
3140    C1               POP     BC              ;Zeilenzähler
3141    1000             DJNZ    linelop         ;bis 12 Dotzeilen gedr.

               ;weiter mit der nächsten Zeile
3143    D1               POP     DE              ;HRG-Zeiger
3144    214000           LD      HL,0040h        ;Summand 64
3147    19               ADD     HL,DE           ;LSB um eine Zeile erhöh.
3148    EB               EX      DE,HL           ;Zeiger <- neuer Wert
3149    C1               POP     BC              ;Zeilenzähler
314A    1083             DJNZ    scrnlop         ;bis Bildschirm gedruckt

                 ;fertig - Drucker wieder in Normalstatus versetzen
314C    210532           LD      HL,prestor      ;Dr. reinitialisieren
314F    0604             LD      B,4             ;mit 4 Codes
3151    CDFC31           CALL    lprint          ;Ausgabe
3154    E1               POP     HL              ;Befehlszeiger
3155    23       exit2   INC     HL              ;Befehlszeiger aktualis.
3156    C9               RET                     ;und raus
```

```
                ;CMD ASC: Bildschirmanzeige in Hex umwandeln
3157    FEF6    ASC       CP      0f6h            ;ASC?
3159    2026              JR      NZ,PSR          ;falls nein
3158    ES                PUSH    HL              ;Befehlszeiger retten
315C    CDAA30            CALL    clear           ;HRG-Speicher löschen

                ;Bildschirm retten
315E    D301              OUT     (1),A           ;HRG einschalten
3161    CD5332            CALL    vidsav          ;Bildschirm retten

                ;Bildschirm von ASCII- in Hexanzeige ändern
3164    263C              LD      H,3ch           ;Bildschirmanfang
3166    E5      hexdisp   PUSH    HL              ;dto.
3167    4E                LD      C,(HL)          ;Bildschirmzeichen
3168    3E20              LD      A,' '           ;Blank
316A    B9                CP      C               ;Blank? (nicht verändern)
316B    77                LD      (HL),A          ;diese Stelle löschen
316C    C40832            CALL    NZ,byte         ;kein Bl., 1 Byte umwand.

                ;Zeichen umgewandelt oder Blank überspr.; nächst. Zeichen
316E    E1                POP     HL              ;Bildschirmzeiger
3170    23                INC     HL              ;nächste Stelle
3171    C874              BIT     6,H             ;Bildsch. überschritten?
3173    28F1              JR      Z,hexdisp       ;nein, nächstes Byte
3175    CD4900
                ;Nach Anzeige auf Tastendruck warten, dann Bildschirm restaur.
                          CALL    0049h           ;auf Taste warten
3178    4D                LD      C,L             ;C <- 00, Bildsch. rest.
3179    CD5532            CALL    savrest         ;UP abarbeiten
317C    0300              OUT     (0),A           ;HRG ausschalten
317E    E1                POP     HL              ;Befehlszeiger
317F    1804              JR      exit2           ;und raus

                ;CMD POINT, CMD SET oder CMD RESET
3181    CDA131  PSR       CALL    argum           ;Ausdruck auswerten
3184    FE82              CP      82h             ;RESET?
3186    2908              JR      Z,reset
3188    FE83              CP      83h             ;SET?
318A    280A              JR      Z,set
318C    FEC6              CP      0c6h            ;POINT?
318E    2800              JR      Z,point
3190    1808              JR      PSRexit         ;falls Fehler

                  ;CMD RESET
3192    78        reset   LD      A,B             ;HRG-Byte laden
3193    A9                XOR     C               ;Bit rücksetzen
3194    1802              JR      out             ;Ausgabe und zurück

                  ;CMD SET
3196    78        set     LD      A,8
3197    81                OR      C               ;Bit setzen
3198    D305      out     OUT     (5),A           ;auf HRG ausgeben

                  ;gemeinsamer Rücksprung für CMD RESET und SET
319A    C38C01    PSRexit JP      018ch           ;-> BASIC

                  ;CMD POINT - Flag für Dot ja oder nein in (CMDBUF)
319D    78        point   LD      A,B
319E    C39201            JP      0192h           ;weiter wie Level 2

                  ;CMD POINT, SET oder RESET - Argument ausrechnen
3141    F5        argum   PUSH    AF              ;Token retten
3142    23                INC     HL              ;Befehlszeiger nachstell.
3143    CF                RST     08h             ;auf '1' prüfen
3144    28                DB      '1'             ;sonst Syntaxfehler
3145    CD461E            CALL    1e46h           ;DE <- Abszisse
3148    E5                PUSH    HL              ;Befehlszeiger retten
3149    217F01            LD      HL,017fh        ;Maximalwert für X
31AC    DF                RST     18h             ;Vergleich mit Eingabe
314D    E1        fcterr  POP     HL              ;Befehlszeiger restaur.
```

```
314E    04441E  00297           JP      C,1e4ah         ;Funktionsf., falls mehr
3181    D5      00298           PUSH    DE              ;Abszisse retten
3182    CF      00299           RST     08h             ;auf ',' prüfen
3183    2C      00300           DB      ','             ;sonst Syntaxfehler
3184    CD461E  00301           CALL    1e46h           ;DE <- Ordinate
3187    E5      00302           PUSH    HL              ;Befehlszeiger retten
3188    218F00  00303           LD      HL,00bfh        ;Maximalwert für Y
3188    DF      00304           RST     18h             ;Vergleich mit Eingabe
31BC    E1      00305           POP     HL              ;Befehlszeiger restaur.
3I80    38EE    00306           JR      C,fcterr        ;Fehler, falls höher
318e    22E432  00307           LD      (cmdbuf),HL     ;Befehlszeiger retten
31C2    EB      00308           EX      DE,HL           ;zur Vereinf. d. Folgend.
31C3    E3      00309           EX      (SP),HL         ;X- <-> Y-Koordinate
3104    3E06    00310           LD      A,06h           ;wegen 6 Dots/Byte
3106    CD7944  00311           CALL    4479h           ;HL/A=HL Rest A
31C9    D1      00312           POP     DE              ;Ordinate holen
31CA    F5      00313           PUSH    AF              ;Bit-Nr. retten
3108    E5      00314           PUSH    HL              ;dto. X-Koordinate
31CC    EB      00315           EX      DE,HL           ;HL <- Y-Koordinate
31CD    3E0C    00316           LD      A,0ch           ;wegen  12 Dotzeil./Stelle
31CF    CD7944  00317           CALL    4479h           ;HL/A=HL Rest A
3102    F5      00318           PUSH    AF              ;Rest retten
3103    3E40    00319           LD      A,40h           ;64 Stellen/Zeile
3105    CD7644  00320           CALL    4476h           ;A*HL=AHL
3108    44      00321           LD      B,H             ;HL nach BC retten
31D9    4D      00322           LD      C,L
31D4    F1      00323           POP     AF              ;obigen Rest holen
3108    210004  00324           LD      HL,0400h        ;Faktor 1kB
310E    CD7644  00325           CALL    4476h           ;A*HL=AHL
31E1    09      00326           ADD     HL,BC           ;Zwischensumme
31E2    C1      00327           POP     BC              ;X-Koordinate
31E3    09      00328           ADD     HL,BC           ;Endsumme = HRG-Adresse
31E4    7D      00329           LD      A,L             ;LSB
31E5    0302    00330           OUT     (2),A           ;auf HRG ausgeben
31E7    7C      00331           LD      A,H             ;MSB
31E8    D303    00332           OUT     (03h),A         ;dto.
31EA    C1      00333           POP     BC              ;B <- Bit-Nr.
310B    04      00334           INC     B               ;B <- mindestens 1
31EC    3E80    00335           LD      A,80h           ;Anfangswert für A
31EE    07      00336   doublop RLCA                    ;fortgesetzt A*2
31EF    10FD    00337           DJNZ    doublop         ;bis richtiges Bit in A
31F1    E63F    00338           AND     3fh             ;nur die ersten 6 Bits
31F3    4F      00339           LD      C,A             ;Akku retten
31F4    DB04    00340           IN      A,(4)           ;Inh.   HRG-Speicherstelle
31F6    47      00341           LD      B,A             ;HRG-Byte retten
31F7    F1      00342           POP     AF              ;Token restaurieren
31F8    24E432  00343           LD      HL,(cmdbuf)     ;Befehlszeiger restaur.
31FB    C9      00344           RET                     ;erledigt
                00345
                00346 ;Zeichenfolge ab (HL) auf Drucker ausgeben
31FC    CDD105  00347 lprint    CALL    05d1h           ;Drucker bereit?
31FF    20FB    00348           JR      NZ,lprint       ;falls nein
3201    7E      00349           LD      A,(HL)          ;zu druckendes Zeichen
3202    D3FD    00350           OUT     (0fdh),A        ;auf Drucker ausgeben
3204    23      00351           INC     HL              ;nächste Stelle
3205    10F5    00352           DJNZ    lprint          ;nächstes Zeichen
3207    C9      00353           RET                     ;zurück
                00354
                00355 ;UP Hexanzeige: 1 Byte ändern
320E    7C      00356   byte    LD      A,H             ;MSB der Videoadresse
3209    E603    00357           AND     03              ;Adr. Vid. -> Adr. HRG
320B    57      00358           LD      D,A             ;neues MSB
320C    SD      00359           LD      E,L             ;HRG-MSB wie Video-MSB
320D    79      00360           LD      A,C             ;Videozeichen
320E    E6F0    00361           AND     0f0h            ;oberes Nibble
3210    0F      00362           RRCA                    ;ins untere schieben
3211    0F      00363           RRCA
3212    0F      00364           RRCA
3213    0F      00365           RRCA
3214    21B032  00366           LD      HL,chrtab-5     ;vor Tab. f. Hexzeich,
3217    E5      00367           PUSH    HL              ;brauchen wir noch
```

```
3218    CD1F32  00368           CALL    nibble          ;oberes Nibble anzeigen
321B    79      00369           LD      A,C             ;alter Code
321C    E60F    00370           AND     0fh             ;unteres Nibble
321E    E1      00371           POP     HL              ;Tabellenzeiger
                00372
                00373 ;einzelnes Halbbyte in die HRG laden
321F    47      00374   nibble  LD      B,A             ;als Zähler i. d. Tabelle
3220    3A3932  00375           LD      A,(displc)      ;Sprungdistanz
3223    EE02    00376           XOR     output-displc-1 ;umschalten
3225    323932  00377           LD      (displc),A      ;neu laden
3228    04      00378           INC     B               ;wegen DE = Tabelle -5
3229    23      00379 seekchr   INC     HL              ;Zeiger nachstellen
322A    23      00380           INC     HL              ;über 5 Stellen, weil
322B    23      00381           INC     HL              ;5 Codes pro Zeichen
322C    23      00382           INC     HL
322D    23      00383           INC     HL
322E    10F9    00384           DJNZ    seekchr         ;bis Code gefunden
3230    0605    00385           LD      B,5             ;5 Dotzeilen pro Zeichen
3232    C5      00386 nibloop   PUSH    BC              ;Zähler retten
3233    CD4B32  00387           CALL    HRGadr          ;HRG-Stelle adressieren
3236    18      00388           DEC     DE              ;HRG-Zeiger korrigieren
3237    7E      00389           LD      A,(HL)          ;Dotzeile laden
3238    1B00    00390           JR      $+2             ;variable Sprungdistanz
3239            00391 displc    EQU     $-1             ;hier Distanzbyte
3236    07      00392           RLCA                    ;lower Nibble verschieben
3238    07      00393           RLCA                    ;um 2 Dots
323C    4F      00394 output    LD      C,A             ;Dotzeile retten
323D    DB04    00395           IN      A,(4)           ;HRG-Byte mit Videocode
323F    B1      00396           OR      C               ;mit Dotzeile verknüpfen
3240    D305    00397           OUT     (5),A           ;Dotzeile in HFG laden
3242    14      00398           INC     D               ;im MSB um 1 kB erhöhen
3243    14      00399           INC     D               ;für nächste Dotzeile
3244    14      00400           INC     D
3245    14      00401           INC     D
3246    23      00402           INC     HL              ;nächster Code für Ziffer
3248    10E8    00404           DJNZ    nibloop         ;bis Nibble angezeigt
324A    C9      00405           RET                     ;zurück
3247    C1      00403           POP     BC              ;Zähler restaurieren
3248 10E8       00404           DJNZ    nibloop         ;bis Nibble angezeigt
324A C9         00405           RET                     ;zurück
                00406
                00407   ;UP, um die HRG-Adresse auszugeben
324B    7B      00408   HRGadr  LD      A,E             ;LSB der HRG-Adresse
324C    D302    00409           OUT     (2),A           ;auf Port ausgeben
324E    7A      00410           LD      A,D             ;MSB
324F    D303    00411           OUT     (3),A           ;dto.
3251    13      00412           INC     DE              ;nächste HRG-Stelle
3252    C9      00413           RET
                00414
                00415   ;UP, um Bildschirm zu retten oder zu restaurieren
3253    0EC0    00416   vidsav  LD      C,0c0h          ;Konstante
3255    210030  00417   savrest LD      HL,3c00h        ;Bildschirmadresse
3258    55      00418           LD      D,L             ;DE <- 00xx, HRG-Adresse
3259    50      00419           LD      E,L             ;DE <- 0000
325A    0604    00420   vidsav1 LD      B,4             ;4*2 Bits/Byte
325C    CD4B32  00421   vidsav2 CALL    HRGadr          ;HRG-Stelle adressieren
325F    CB79    00422           BIT     7,C             ;Bildschirm retten?
3261    2812    00423           JR      Z,restor        ;falls nein
3263    7E      00424           LD      A,(HL)          ;Bildschirmzeichen
3264    A1      00425           AND     C               ;nur oberste 2 Bits
3265    C5      00426           PUSH    BC              ;C retten
3266    4F      00427           LD      C,A             ;Akku retten
3267    DB04    00428           IN      A,(4)           ;HRG-Byte holen
3269    E63F    00429           AND     3fh             ;oberste Bits ausmaskier.
326B    B1      00430           OR      C               ;Videobits zufügen
326C    C1      00431           POP     BC              ;C restaurieren
326D    D305    00432           OUT     (5),A           ;auf HRG ausgeben
326F    CB06    00433           RLC     (HL)            ;Zeichen 2 Bits aufrücken
3271    CB06    00434           RLC     (HL)
3273    1808    00435           JR      gosave          ;dort weiter
3275    OB04    00436   restor  IN      A,(4)           ;HFG-Byte holen
```

```
3277    07      00437           RLCA                    ;2 oberste Bits
3278    CB16    00438           RL      (HL)            ;in den Bildsch. laden
327A    07      00439           RLCA
327B    CB16    00440           RL      (HL)
327D    1000    00441   gosave  DJNZ    vidsav2         ;bis 1 Byte fertig
327F    23      00442           INC     HL              ;nächste Videostelle
3280    CB74    00443           BIT     6,H             ;Bildsch. überschritten?
3282    C0      00444           RET     NZ              ;falls ja
3283    1805    00445           JR      vidsav1         ;weiter, falls nein
                00446
                00447   ;Codetabelle für HRG-Zeichensatz
3285    02      0044B   chrtab  DB      2,5,5,5,2       ;0
328A    04      00449           DB      4,6,5,4,4       ;1
328F    03      00450           DB      3,4,2,1,7       ;2
3294    07      00451           DB      7,4,2,4,7       ;3
3299    04      00452           DB      4,5,7,4,4       ;4
329E    07      00453           DB      7,1,3,4,3       :5
3243    06      00454           DB      6,1,7,5,2       ;6
3248    07      00455           DB      7,4,2,1,1       ;7
32AD    02      00456           DB      2,5,2,5,2       :8
32B2    02      00457           DB      2,5,7,2,1       ;9
32B7    02      00458           OB      2,5,7,5,5       ;A
32BC    03      00459           DB      3,5,3,5,3       ;B
3201    06      00460           DB      6,1,1,1,6       ;C
32C6    03      00461           DB      3,5,5,5,3       ;D
32CB    07      00462           OB      7,1,3,1,7       ;E
32D0    07      00463           DB      7,1,3,1,1       ;F
                00464
                00465   ;versch. Druckercodes, Puffer für Dotspalten usw.
32D5    0D      00466   prestor DB      0dh,0dh,1bh,'§'         ;Drucker reinitialisieren
32D9    1B      00467           DB      1bh,'M',09h             ;li. Rand auf 9. Stelle
32DC    18      00468           DB      1bh,'A',06h             ;Zeilenabstand 6/72"
32DF    0D      00469   lninit  DB      0dh,1bh,'K',80h,01h     ;Dotgr. einf. D.
32E4    0000    00470   buffer  DW      0000h,0000h,0000h       ;Puffer f. HRG
32EA            00471   cmdbuf  EQU     $                       ;Puffer für Befehlszeiger
                00472
3000            00473                   END     finit
00000 Fehler
ASC     3157    CLS     30A6    HRGadr  3248    HRGoff  30E5    INV     30AE
LINE    30D9    LPRINT  30ED    PSR     3181    PSRexit 319A    argum   31A1
bitlop  3120    buffer  32E4    byte    3208    bytelop 3118    chrtab  3285
clear   30AA    clsinv  3084    cmd     3074    cmdbuf  32EA    command 302C
displc  3239    doublop 31EE    exit1   30D7    exit2   3155    fcterr  31AD
gosave  327D    hello   3038    hexdisp 3166    hlinlop 3112    finit   3000
linelop 3103    lninit  32DF    lprint  31FC    lsloop  308E    msloop  3087
nibble  321F    nibloop 3232    out     3198    outa    30CF    output  323C
point   319D    prestor 32D5    reset   3192    restor  3275    savrest 3255
scrnlop 30FF    seekchr 3229    set     3196    swaplop 308E    vidsav  3253
vidsavl 325A    vidsav2 325C
```

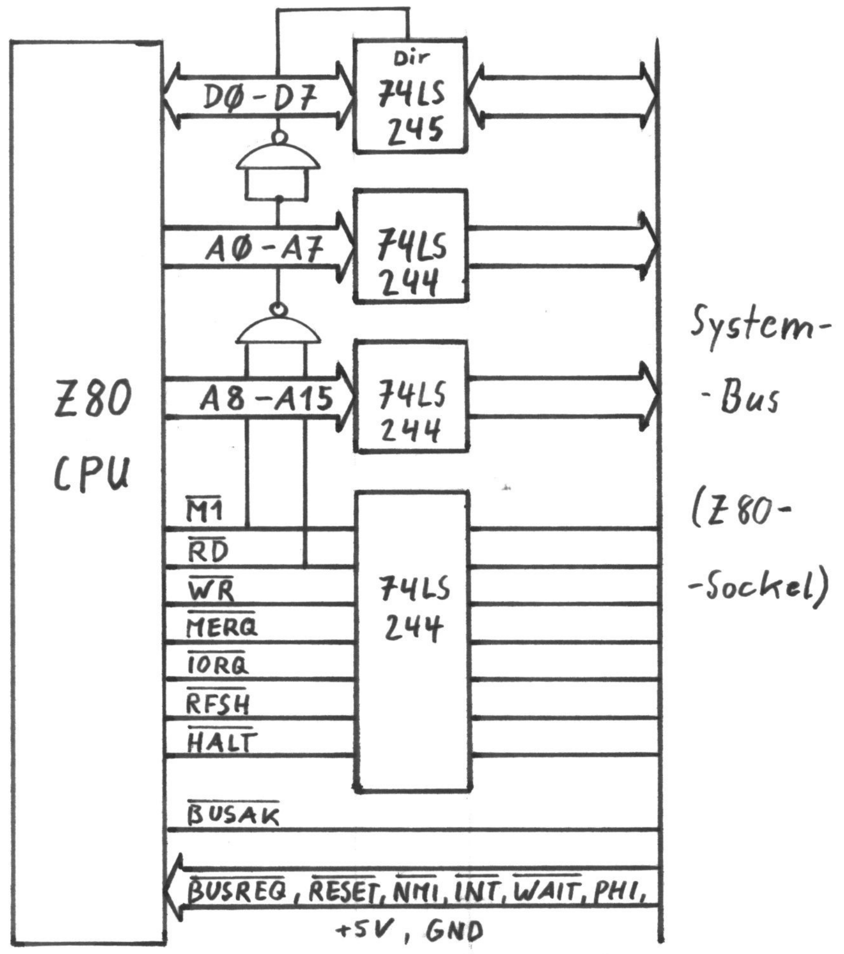
Wie aus dem Komtek 1 ein Computer wird

Bus-Puffer in Z80-Billig-Computern

Helmut Bernhardt

Erfreulicherweise gibt es eine Vielzahl von sehr billigen Computern, die vielen Interessierten den Einstieg in die Anwendung von Computern für die Lösung verschiedenster Aufgaben ermöglichen. Die günstigen Preise solcher Geräte sind hauptsächlich auf ein sparsames Konzept in der Hardware zurückzuführen. Ein Computer wird für einen bestimmten Ausbau der Hardware und Peripherie entworfen, wobei dann auf alles überflüssige verzichtet wird.

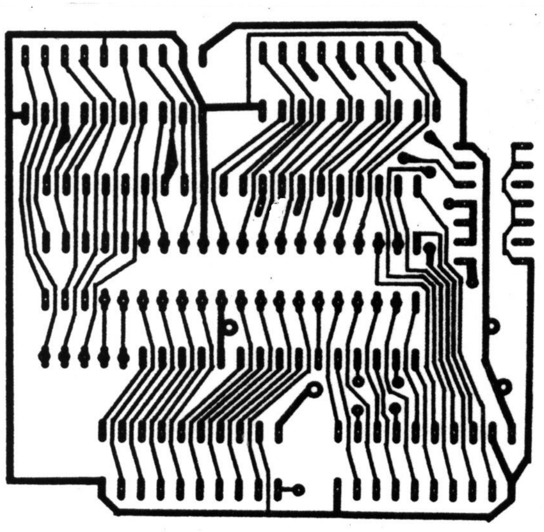
Der Appetit kommt erst beim Essen, d.h., wer sich einen Billig-Computer zugelegt hat, verspürt bald den Wunsch, diese oder jene Erweiterung anzuschließen. Bei entsprechendem Bedarf finden sich auch schnell Hersteller für solche Erweiterungen. Wenn die Menge der Erweiterungen über das ursprüngliche Konzept der Hardware hinausgeht, treten oft Schwierigkeiten auf. Eine dieser Schwierigkeiten kann die mangelnde Treiberleistung der aktiven CPU-Signale sein. Eine Lösung dieses Problems für Computer mit einer Z80-CPU, bei denen ca. 10,- DM für Treiber-ICs eingespart wurden, sei hier vorgestellt.

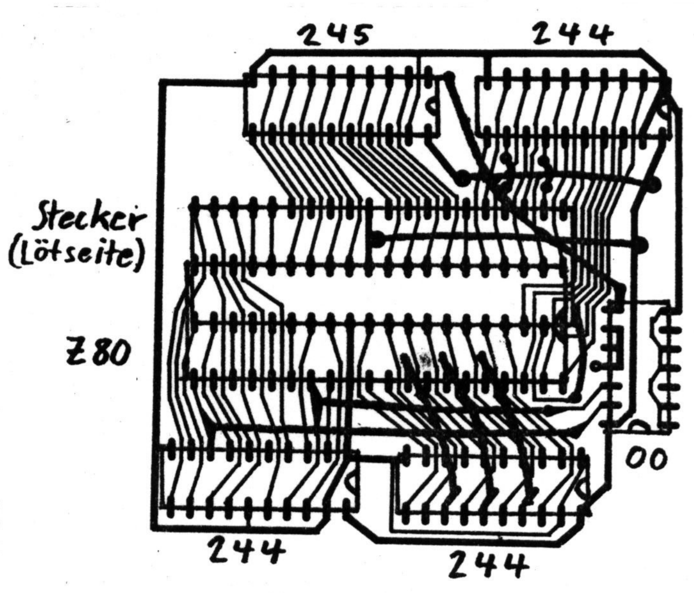


Prinzip-Schaltbild

Um zwischen den Z80 und das CPU-Board Treiber schalten zu können, ohne dabei alle Leiterbahnen zu durchtrennen und ohne über ein wildes Drahtgewirr Treiber mit dazwischen zu schalten, ist es am günstigsten, den Z80 mit den zusätzlichen Treiber ICs auf einem zusätzlichen Board unterzubringen und dieses Board in den Sockel des Z80 zu stecken. Das CPU-Board erhält damit ein 1. OG aufgesetzt, was eventuell zu Platzschwierigkeiten führt, dafür aber das Aussteigen des Computers bei etwas mehr Hardware unterbindet. Das Konzept des in Abb. 2 als Layout dargestellten Boards ist auf Bastler ausgerichtet, die sich das Board selbst ätzen wollen. Soweit dies möglich ist, sind alle Leiterbahnen auf einer Seite des Boards gebracht worden, so daß nur noch wenige Drahtbrücken auf der Bestückungsseite nötig sind.

Trotz der einfachen Ausführung ist über die AND-Verknüpfung von RD* und M1* zur Richtungssteuerung des Datenbus-Transceivers auch IM2-Fähigkeit gewährleistet.



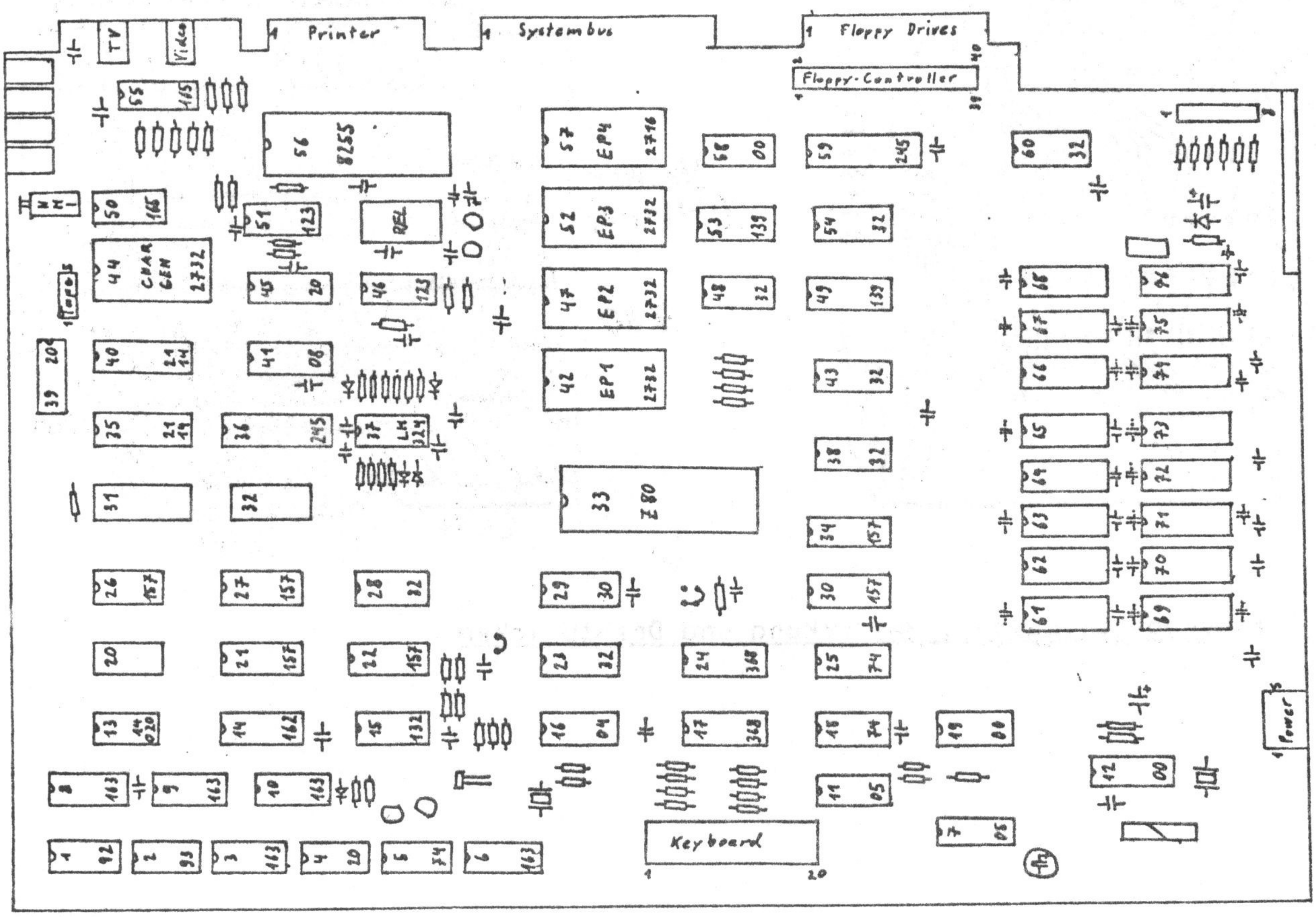


Layout *(von der Lötseite gesehen),* ***Drahtbrücken und Bestückung*** *(von der Bestückungsseite gesehen)*

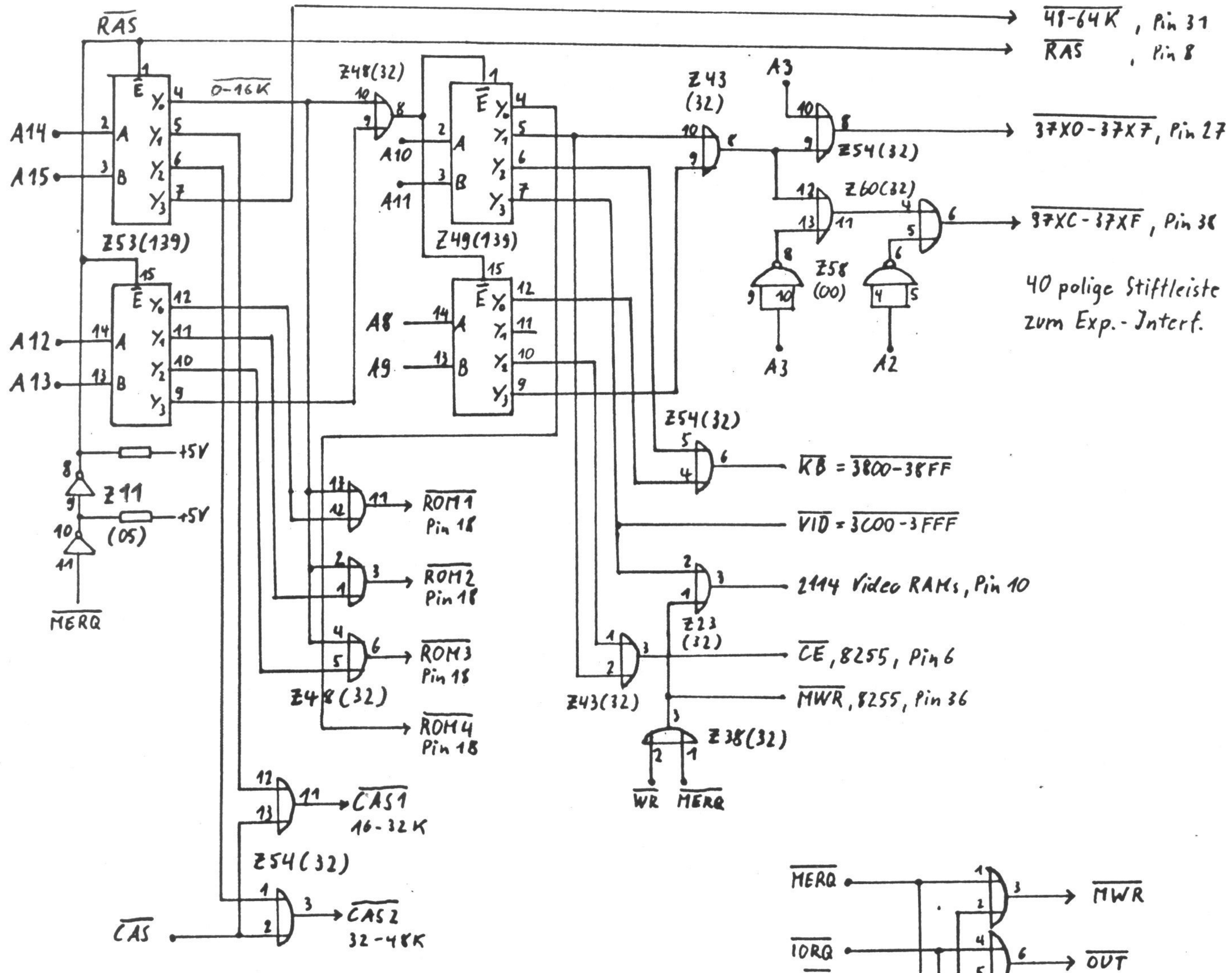
Durch Abtrennen der Masseleitungen von Pins 1 und 19 der 74LS244-ICs und von Pin 19 des 74LS245 und Anlegen des über ein freies NAND-Gatter des 74LS00 invertierten BUSAK* Signals läßt sich auch DMA-Fähigkeit herstellen. Da in Billig-Computern DMA-Betrieb nicht vorgesehen ist, wurde zur Vereinfachung des Layouts (weniger Drahtbrücken) darauf verzichtet.

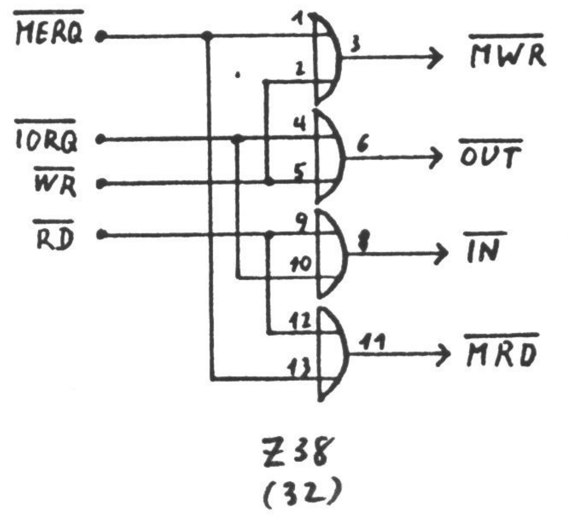
Mit einem Finanzaufwand von knapp 20,- DM läßt sich damit so manchem überlasteten Schneider-CPC, ZX81, Komtek 1 u.s.w. wieder die volle Funktionsfähigkeit zurückgeben.

Bestückungsplan des Motherboards des Komtek 1

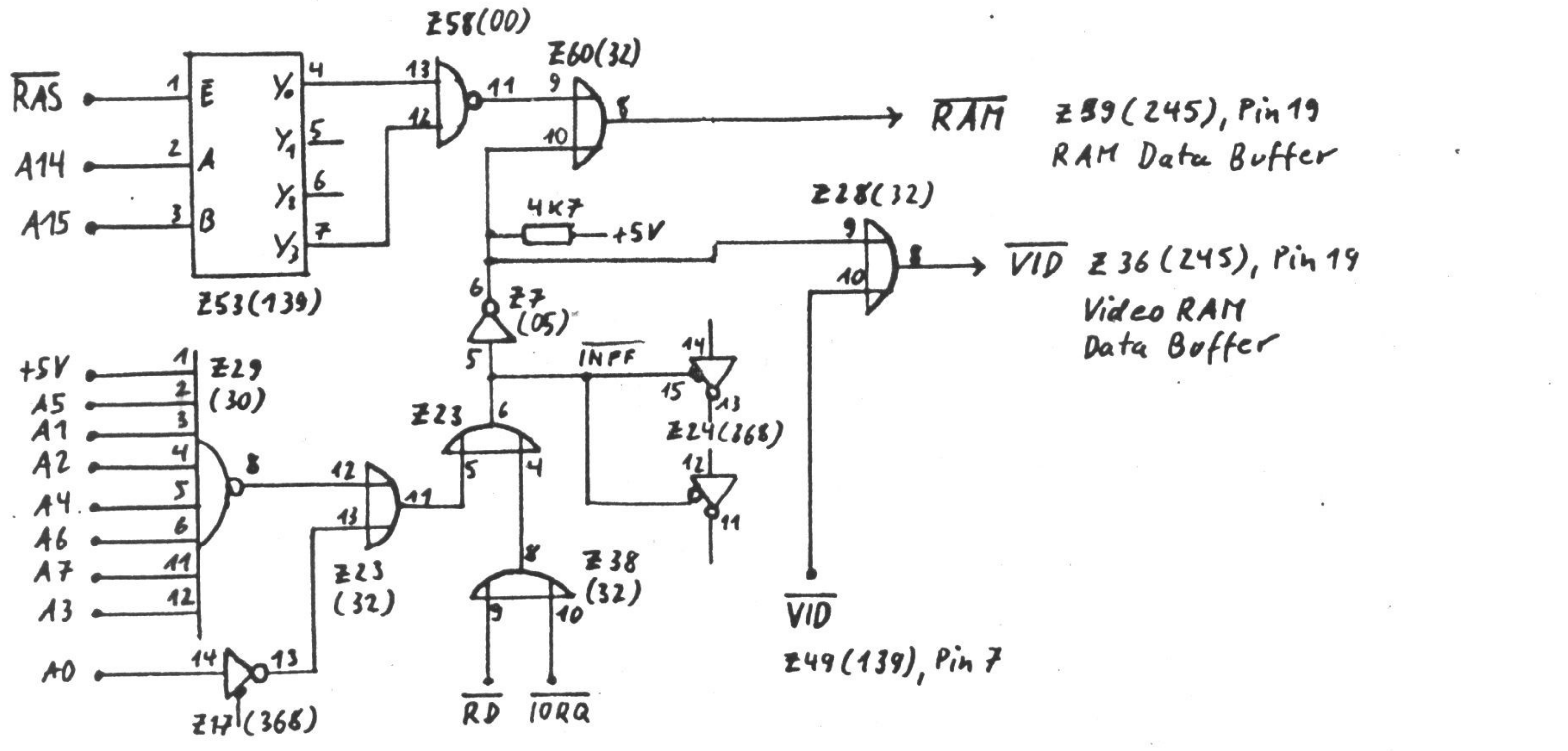


Auszüge aus der Schaltung des Motherboards





2) Auszüge aus dem Schaltplan



Spätere Anmerkung 2015, Helmut
Vor ein paar Tagen fand ich im Keller noch ein Exemplar dieses Bustreiber Boards und musste feststellen, dass gegenüber dem hier abgebildeten Layout und Bestückungs- und Drahtbrücken Plan auf dem Board noch einige Leiterbahnen getrennt und einige zusätzliche Drahtbrücken gelegt wurden. Die Dokumentation war fertig, bevor das Board geprüft und korrigiert wurde. Hauptsächlich lagen Fehler im Anschluss von +5V und GND vor.

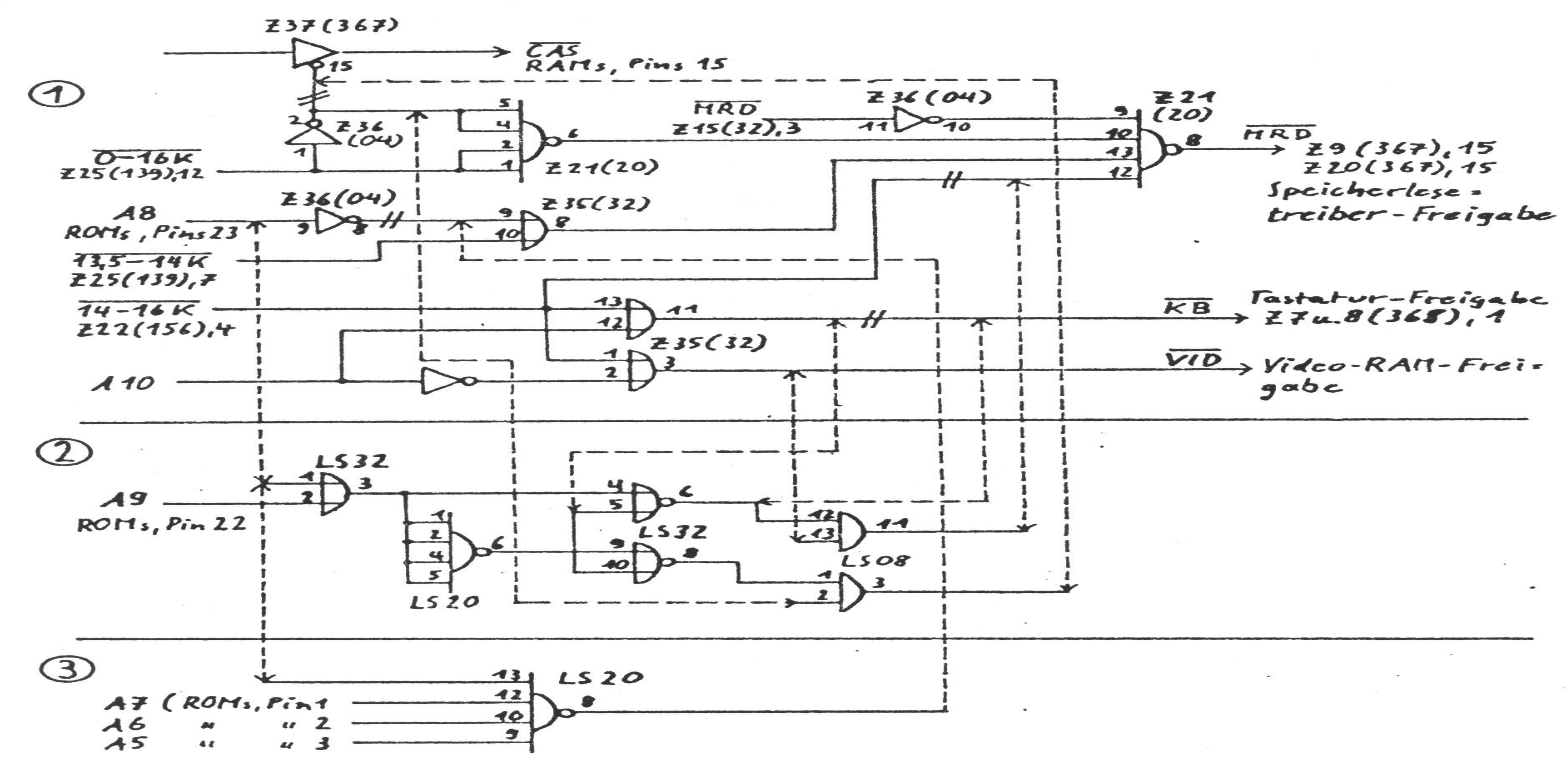
GENIE I und II: Volldecodierter Sonder-ROM; RAM im Bereich 3900H bis 3BFFH

Helmut Bernhardt

Der Begriff Kompatibilität, den man meistens im Zusammenhang mit IBM-PC-Nachbauten liest, spielte auch beim TRS-80-Nachbau GENIE eine wichtige Rolle. So wurde gegenüber den ersten "Video-GENIEs" später zusätzlich zu Port FDH auch die Adresse 37E8H für den Betrieb des Druckers eingerichtet. Leider ist auch viel Unsinn vom TRS-80 mit übernommen worden, so z.B. die Decodierung der Tastatur. Obwohl die Tastatur prinzipiell mit 8 Speicheradressen auskäme (oder sogar mit nur einer Portadresse), braucht sie wegen mangelnder Decodierung den gesamten Bereich 3800H-38FFH. Außerdem werden dabei auch noch die 3/4K Adreßraum 3900H bis 3BFFH verschenkt, ohne daß die Tastatur in der vorliegenden Decodierung dies erzwingt. Ein anderer TRS-80-Nachbau, der Komtek 1, hat die Tastatur auf den Bereich 3800H-38FFH beschränkt, ohne dabei (zumindest in Sachen Tastatur) die Kompatibilität zum TRS-80 einzubüßen. Mit dem Bereich 3900H-3BFFH wird aber dort auch nichts Sinnvolles gemacht.

Die Lücke innerhalb der unteren 16K zwischen ROM und Drucker-Floppy-Bereich, der Adreßraum 3000H bis 37DFH, wird im GENIE teilweise als Sonder-ROM genutzt. Durch mangelhafte Decodierung ist der Sonder-ROM im Bereich 3700H bis 37DFH nicht erreichbar.

Der Teil 1 der Schaltung zeigt die Bestandteile des CPU-Boards des GENIE, die für diese Mißstände verantwortlich sind. Die Abweichungen von den Darstellungen im Service-Manual sind keine Fehler sondern Richtigstellungen. Außerdem ist hier schon der Zustand nach Umrüsten auf 4164-RAMs zugrundegelegt, der für die Schaffung von RAM bei 3900H bis 3BFFH Vorraussetzung ist.



Die Freigabe des Tastatur-Lesetreibers erfolgt bei allen Adressen 3800H bis 38FFH. Die im Teil 2 dargestellte Schaltungsänderung beschränkt diese Freigabe auf die Adressen 3800H bis38FFH und bewirkt außerdem, daß die Freigabe des Speicherlesetreibers und des RAM auch im Adreßraum 3900H bis 3BFFH erfolgt, d.h. hier liegt RAM vor.

Teil 3 zeigt die für das vollständige Decodieren des Sonder-ROMs nötige Änderung. Nach dieser Änderung sind alle Befehle des Sonder-ROM (auch P) verfügbar. Wenn am GENIE ein EG64MBA oder Abkömmling davon betrieben wird, ist dadurch der Bereich 3900H bis 3BFFH nicht doppelt vorhanden. Bei Abschalten der Tastatur und Einblenden von RAM wird dann nur

noch zusätzlich der Bereich 3800H bis 38FFH als RAM verfügbar.

Wenn das "Sichere Plätzchen" nach c't 5/85 eingebaut ist, ist auch die o.a. Bereitstellung von RAM oberhalb der Tastatur nicht mehr nötig. Die sehr viel aufwendigere Schaltung aus c't hat den Vorteil, daß sie bei Betrieb des EG64MBA tatsächlich den Bereich 3900H bis 3BFFH zusätzlich zur Verfügung stellt. Der im Grundzustand oberhalb der Tastatur liegende RAM ist dort nicht der gleiche, der bei Abschalten der Tastatur verfügbar wird.

Umbauanleitung

In den Teilen 1-3 des Schaltplans sind alle ICs des CPU-Boards mit Nummer nach Bestückungsplan, Typ, und Pin bezeichnet. Auch die Herkunft sowie die Ziele der verwendeten Signale sind angegeben. Die in den Teilen 2 und 3 zusätzlich benötigten Gatter können durch 3 weitere ICs (74LS08, 20, 32) zur Verfügung gestellt werden. Diese Gatter haben im Schaltplan nur eine Typenbezeichnung.

Bei diesen zusätzlich einzubauenden ICs werden die Pins 1-6 sowie 8-13 hochgebogen. Mit den Pins 7 und 14 werden sie in geeigneter Anordnung an den Pins 7 und 14 der ICs Z35, Z36 und Z21 angelötet und durch Verbindung der bezeichneten Pins untereinander und mit den entsprechenden Pins des CPU-Boards mit isolierten Drähten beschaltet. Die Orientierung der Huckepack-ICs muß die gleiche sein wie die der ICs darunter.

Außerdem sind folgende Leiterbahnen zu durchtrennen bzw. Verbindungen zu legen:

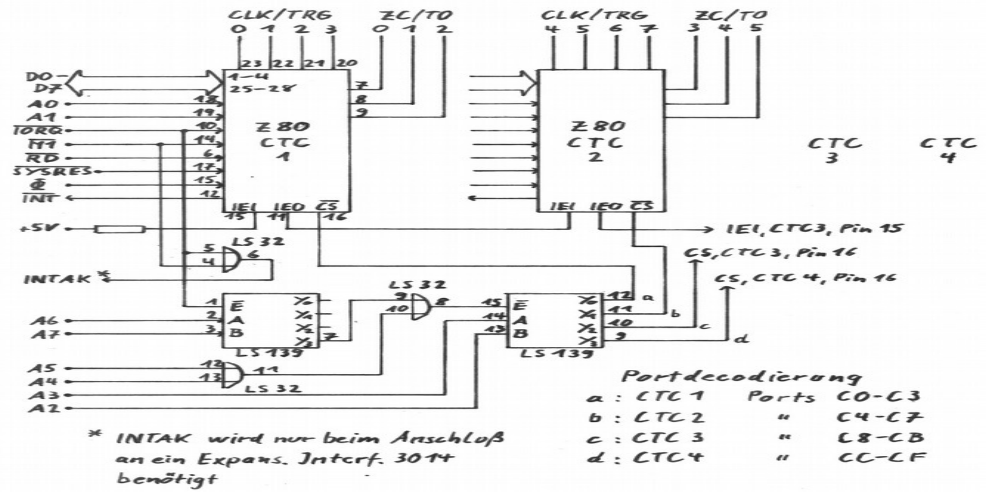
1. Verbindung von Pin 15 von Z37(367) mit Pins 4 und 5 von Z21(20);
   Pin 2 von Z36(04) ist nun direkt mit den Pins 4 und 5 von Z21(20) zu verbinden
2. Verbindung von Pin 2 von Z36(04) mit Pin 15 von Z37(367);
   diese Verbindung liegt nach dem Umrüsten auf 64K als Drahtbrücke vor
3. Verbindung von Pin 4 von Z22(156) mit Pin 12 von Z21(20)
4. Verbindung von Pin 11 von Z35(32) mit den Pins 1 von Z7 und Z8(368)
5. Verbindung von Pin 9 von Z35(32) mit Pin 8 von Z36(04)

Die Maßnahmen 1 bis 4 sind nötig, um den zusätzlichen RAM im Bereich 3900H bis 3BFFH zu erhalten, während 5 für die vollständige Decodierung des Sonder-ROM durchgeführt werden muß.

Eines der nicht benutzten AND-Gatter des zusätzlichen 74LS00 läßt sich zur Erzeugung eines SYSRES*-Signals aus RESET* (Z80-CPU, Pin 26) und NMI* (Z80-CPU, Pin 17) benutzen. Dieses Signal wird bei vielen Hardware-Erweiterungen benötigt. (Beim EXP1 von RB-Elektronik mit Doubler sollte dieses Signal anstelle des vorgeschlagenen RESET* verwendet werden.)

Hilferuf!
Interrupt-Priority-Encoder und Vektorgenerator für IM2 mit dem Z80-CTC

Wer hat Erfahrung damit, wer kann mir helfen?



Seit nunmehr drei Jahren probiere ich mit verschiedenen Schaltungsvorschlägen zu dem Thema meinen GENIE 1 IM2-fähig zu machen. Das hätte den Vorteil, daß bei mehreren angeschlossenen Hardware-Erweiterungen, die über Interrupt von der CPU bedient werden, das Herumsuchen des Z80 nach dem Störenfried nicht mehr nötig ist, der Timer nicht mehr ruhig gehalten werden muß, wenn ein anderer INT-Lieferant aktiv ist, und ein echter Real-Time-Betrieb mit der Peripherie laufen kann.

Die geeignetste Methode scheint mir über den Z80-CTC zu laufen. Die Interrupt-Signale der verschiedenen Geräte gehen an je einen CTC Zählereingang. Das Zeitkonstanten-Register wird mit 1 geladen, so daß bei jedem Interrupt von außen dieser an die CPU weitergegeben wird. Gleichzeitig kann der CTC einen Vektor liefern und die verschiedenen Kanäle eines CTC sind in ihrer Priorität gestaffelt.

Durch Hintereinander-Schalten mehrerer CTCs in der IEI-IEO-Daisy-Chain lassen sich so biliebig viele Interrupt-Quellen anschließen. Außerdem ist solch ein Priority-Encoder sehr flexibel. Es läßt sich programmieren, ob positive oder negative Flanken gezählt werden sollen. Der IM2-Vektor läßt sich frei programmieren. Durch Umprogrammieren lassen sich einzelne Interrupt-Quellen abwürgen. Und last but not least läßt sich ein CTC-Kanal auch noch dafür verwenden, wofür er gedacht ist.

Ich habe mir ein Board auf Lochraster-Platine aufgebaut und versuchsweise den Timer-Interrupt an den Zählereingang des CTC gelegt und zwischen Timer und CPU-INT-Eingang einen Soft-Ein/Aus-Schalter gelegt. Der Soft-Schalter funktioniert auch so wie geplant.

Offensichtlich bin ich aber in Software-Entwicklung zu unterbelichtet, um dem CTC in geeigneter Weise mitzuteilen, was ich von ihm erwarte. Auch der sehr informative Artikel in der c't 4/85, S.92, der mit 3 Worten diese Nutzung des CTC erwähnt, konnte nicht dazu beitragen, diesen Mißstand bei mir zu ändern.

Wer hat auf diesem Gebiet schon mal experimentiert? Wer kann mir sagen, wie ich den CTC zu behandeln habe, um mein Ziel doch noch zu erreichen? Oder habe ich einen Hardware-Fehler eingebaut?

Helmut Bernhardt

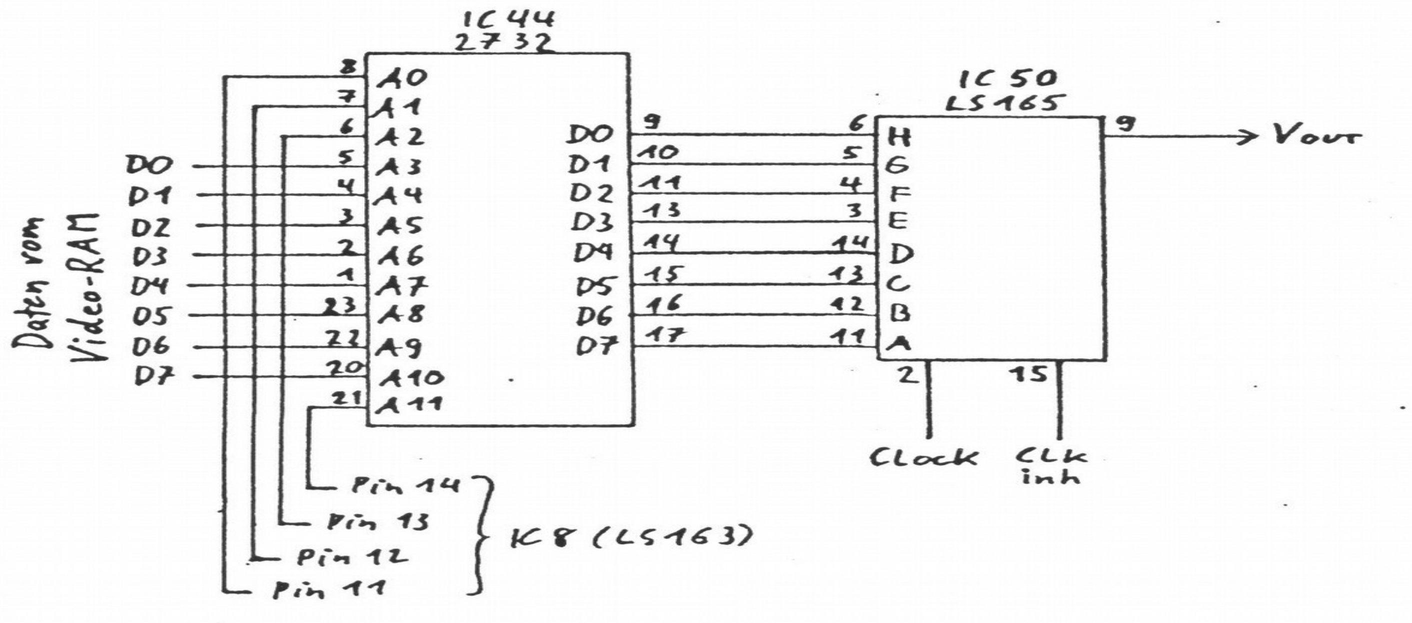
Komtek 1: Verbesserter Zeichengenerator

Helmut Bernhardt

Wer seinen Komtek 1 nach den im Hardware-Sonderheft Nov.85 beschriebenen Umbauten als relativ komfortable Schreibmaschine nutzt, kann sich eigentlich nur noch an den amerikanischen Sonderzeichen, die zudem auch noch falsch sind, anstelle der deutschen Umlaute auf dem Bildschirm stören. Dieser Mißstand wird nun behoben. Voraussetzung ist, daß man jemanden kennt, der ein 2732 eprommen kann.

Zeichengenerator im Komtek 1

Bei der für den Komtek 1 wegen mangelnder Hardware-Unterlagen üblichen Wühlarbeit zum Durchschauen der Schaltung stellte sich die in Abb.1 wiedergegebene Beschaltung des 2732-Zeichengenerator-EPROMs heraus. Weitere Erkenntnisse über die Organisation (Zuordnung von EPROM-Adressen und -Daten zu Bildschirmpositionen) konnten durch Herausnehmen des EPROMs, Hochbiegen und An-Masse-Legen einzelner Pins und wieder Hineinstecken des EPROMs in den Sockel, und Interpretation der veränderten Bildschirmdarstellung erhalten werden. Letzte Unsicherheiten ließen sich durch Auslesen des Inhaltes des Zeichengerators mit einem EPROMer ausräumen (für diesen Dienst an der Digitaltechnik sei Arnulf Sopp herzlichst gedankt).



Zeichengenerator im Komtek 1

Wie sich die verschiedenen Beobachtungen und die Untersuchung des Hexdumps aus dem EPROM zu den folgenden Vorstellungen zusammenfügen, möchte ich mir hier darzustellen ersparen (nicht nur, weil es wohl niemanden interessiert, sondern hauptsächlich, weil ich dazu keine Lust habe). Wichtig ist nur das Ergebnis, und das rechtfertigt den Schreibaufwand.

Danach ist eine Druckposition für ein Zeichen auf dem Bildschirm in 7 waagerechte und 16 senkrechte Punkte aufgeteilt (bei TRS80 und GENIE 6x12). Der Inhalt einer waagerechten Dotreihe entspricht von links nach rechts den Bits D6 bis D0 im jeweiligen Byte im Zeichengenerator. Die EPROM-Adreßeingänge A0 bis A2 adressieren dabei eine von 8 Dotreihen (8 aufeinanderfolgende Bytes im EPROM) in entweder der oberen oder der unteren Hälfte einer Druckposition während durch den Adreßeingang A11 vorgegeben ist, welche Hälfte angesteuert wird.

An die Adreßeingänge A3 bis A10 sind die Daten D0 bis D7 des Video-RAMs geführt, Ensprechend dem ASCII-Code 00-FF liegen also hintereinander in den unteren 2K des EPROM-Adreßbereichs 256 Gruppen von je 8 Bytes, die dem Bildschirminhalt der oberen Hälfte einer Druckposition entsprechen. Analog sind die Informationen für die untere Hälfte einer Druckposition mit einem Offset von 2048 Adressen in den oberen 2K des EPROMs abgelegt.

Zur Darstellung von Zeichen mit ASCII-Codes bis 7FH wird dabei nur die obere Hälfte einer Druckposition genutzt. Die meisten Zeichen werden innerhalb einer 5x7-Matrix dargestellt. Die untere der 8 waagerechten und 2 seitliche senkrechte Dotreihen werden nicht genutzt. Nur bei Unterlängen wird auch die 8. waagerechte Reihe mitbenutzt. Da diese aber für eine echte Unterlänge nicht ausreicht, werden Zeichen wie "g" und "p" einfach hochgeschoben.

Die unteren Hälften der Druckpositionen werden nur bei den Pixelgraphikzeichen (ASCII-Codes größer 7FH) benutzt. Da die Teilung der 16 wagerechten Dotzeilen durch 3 und die der 7 senkrechten durch 2 keine ganzzahligen Werte ergibt, mußte hingenommen werden, daß die Blockgraphik entsprechend ungleichmäßig aussieht. Die Blockgraphik war aber leztendlich nicht Gegenstand der Kritik sondern der Satz der Zeichen mit ASCII-Codes bis 7FH.

Es ist nicht einzusehen, warum diese Zeichen nicht auch die untere Hälfte der Druckposition mitbenutzen sollen. Es muß nicht sein, daß ein Buchstabe an einem Graphikzeichen in der Zeile darüber festklebt. Dafür müssen nur alle Zeichen um 2 Dotzeilen in der Matrix nach unten geschoben werden, d.h. im EPROM müssen alle Bytes in eine um 2 höhere Adresse verschoben werden.

Innerhalb der 8 Bytes für ein Zeichen wird das erste Byte in die Adresse des 3. Bytes, das 2. Byte in die Adresse des 4. Bytes u.s.w. verschoben. Die freigewordenen ersten beiden Bytes werden mit 00H aufgefüllt. Das 7. und 8. Byte werden aber nicht einfach um zwei Adressen verschoben, sondern um 2048-8 Adressen, damit sie nicht in die ersten beiden Bytes des nächsten Zeichens sondern in die oberen 2 Dotzeilen der unteren Hälfte der Druckposition des gleichen Zeichens gelangen. Diese Prozedur wird allerdings nur mit dem Adreßbereich 0-1K (und 2-3K) durchgeführt, weil die in den anderer 2K des EPROMs untergebrachten Informationen die der Blockgrahik-Zeichen sind.

Damit sind die Zeichen zwar alle von der Zeile darüber abgerückt worden, hübscher sind sie dadurch noch nicht geworden. Deshalb müssen die Zeichen für die Umlaute, die Zeichen mit Unterlängen und alle sonstigen unschönen Gebilde neu gestaltet werden. Das ist aber eine enorme Arbeit, die zu verrichten bei Bestehen eines brauchbaren Zeichensatzes (z.B. der des GENIE) nicht ganz einzusehen ist. Der Zeichensatz im GENIE ist aber etwas anders organisiert. Erstens werden die Grafikzeichen nicht über den Zeichengenerator sondern per Hardware erzeugt, weshalb man dort mit einem 2716-EPROM auskommt. Zweitens wird ein Zeichen durch 16 aufeinanderfolgende Bytes im EPROM dargestellt, wovon nur die ersten 12 Bytes (entsprechend 12 Dotreihen pro Druckposition) benutzt werden. Und drittens ist die Zuordnung der Bits eines Bytes im Zeichengenerator zu den Dots einer Dotreihe genau umgekehrt, d.h. die niederwertigen Bits representieren die linken Dots in der Reihe, was beim Komtek ungekehrt ist.

Da mir bekannt war, daß Arnulf Sopp schon einen verbesserten Zeichensatz für den GENIE erstellt hat, der alle Zeichen um 2 Dots herunterrückt und die Großbuchstaben Ä, Ö und Ü auch tatsächlich als Großbuchstaben erscheinen läßt, und ich nicht nur zu bequem bin sondern auch Respekt vor den Fähigkeiten anderer, die das bedeutend besser können, habe, schmarotzte ich einfach von Arnulfs Arbeit und bat ihn, mir seinen Zeichensatz in den des Komtek einzumassieren. Das ist ihm mit der beiliegenden eleganten Soppftware auch hervorragend gelungen.

Nur einen Haken hat das ganze Projekt. Arnulfs Zeichensatz sieht für die ASCII- Codes 00-1F nicht die für den GENIE üblichen fernöstlichen Gebilde sondern die diesen Codes entsprechenden Hexzahlen vor. Die werden, wenn im GENIE eine ent- sprechende Änderung vorgenommem wird, dann auch im Bildschirm dargestellt, wenn eine Zahl zwischen 00 und 1F in den Bildwiederholseicher gepoked wird.

Beim Komtek ist eine entsprechende Änderung nicht möglich, bzw. ich habe sie noch nicht herausgefunden. Deshalb erscheinen beim Arbeiten im Level-II-Basic, das anstelle der ASCII-Codes 40-5F die Codes 00-1F an den Bildwiederholspeicher ausgibt, dort anstelle der Großbuchstaben die Arnu1f'schen übereinander gestellten Hexzahlen auf dem Bildschirm. Im DOS und Disk-Basic ist dies nicht mehr der Fall.

Um im Level-II-Basic auch erkennbare Zeichen auf dem Bildschirm zu haben, müßten deshalb anstelle der ansonsten sehr informativen Hexzahlen für die ASCII-Codes 00-1F lieber die Zeichen mit den ASCII-Codes 40-5F verwendet werden. Für Bastler, die wegen häuiger Hardware-Eingriffe gelegentlich auf ihre Floppy-Laufwerke verzichten müssen und dann für Testzwecke auf Level-II-Basic angewiesen sind, kann das doch von Bedeutung sein.

Helmut Bernhardt

Und hier Arnulfs Soppftwärechen

5200		0001	ORG		5200h	
		0002				
5200	2100A0	0003	start	LD	HL,0a000h	;ab hier Komtek-Zeichensatz
5203	110080	0004		LD	DE,8000h	;ab hier GENIE-Zeichensatz
5206	06B0	0005		LD	B,80h	;128 Zeichen mit 16 Bytes
5208	C5	0006	loop1	PUSH	BC	;Zähler retten
5209	0608	0007		LD	B,8	;Zähler 8 Bytes(1/2 Zeile)
520B	CD2452	0008	loop2	CALL	modify	;Komtek-Zeichens. verändern
520E	10FB	0009		DJNZ	loop2	;8 von 16 Bytes
5210	01F807	0010		LD	BC,07f8h	;untere Hälfte d. Zeichens.
5213	09	0011		ADD	HL,BC	;Adresse untere Hälfte
5214	04	0012		INC	B	;B <- Zähler 8 Bytes
5215	CD2452	0013	loop3	CALL	modify	;untere Hälfte modifizieren
4218	10FB	0014		DJNZ	loop3	;8 Bytes der unteren Hälfte

```
521A    B7                        OR      A            ;Cy löschen
521B    010008                    LD      BC,0800h     ;jetzt 2K tiefer
521E    ED42                      SBC     HL,BC        ;neue Komtek-Adresse
5220    C1                        POP     BC           ;Zähler 128*16
5221    10E5                      DJNZ    loop1        ;bis 128 Zeichen fertig
5223    C9                        RET                  ;ins Betriebssystem

5224    C5              modify    PUSH    BC           ;Bytezähler retten
5225    0608                      LD      B,8          ;8 Bit pro Byte
5227    1A                        LD      A,(DE)       ;GENIE-Byte
5228    07              loop4     RLCA                 ;zur Spiegelung
5229    CB19                      RR      C            ;Byte rotieren
522B    10FB                      DJNZ    loop4        ;bis ein Byte gespiegelt
522D    71                        LD      (HL),C       ;neues Komtek-Zeichen
522E    23                        INC     HL           ;nächste Komtek-Stelle
522F    13                        INC     DE           ;nächste GENIE-Stelle
5230    C1                        POP     BC           ;Bytezähler restaurieren
5231    C9                        RET                  ;ins Hauptprogramm

5200                    END       start                ;dort Einsprung
```

CP/M Hardware für TRS 8O, GENIE und Komtek 1

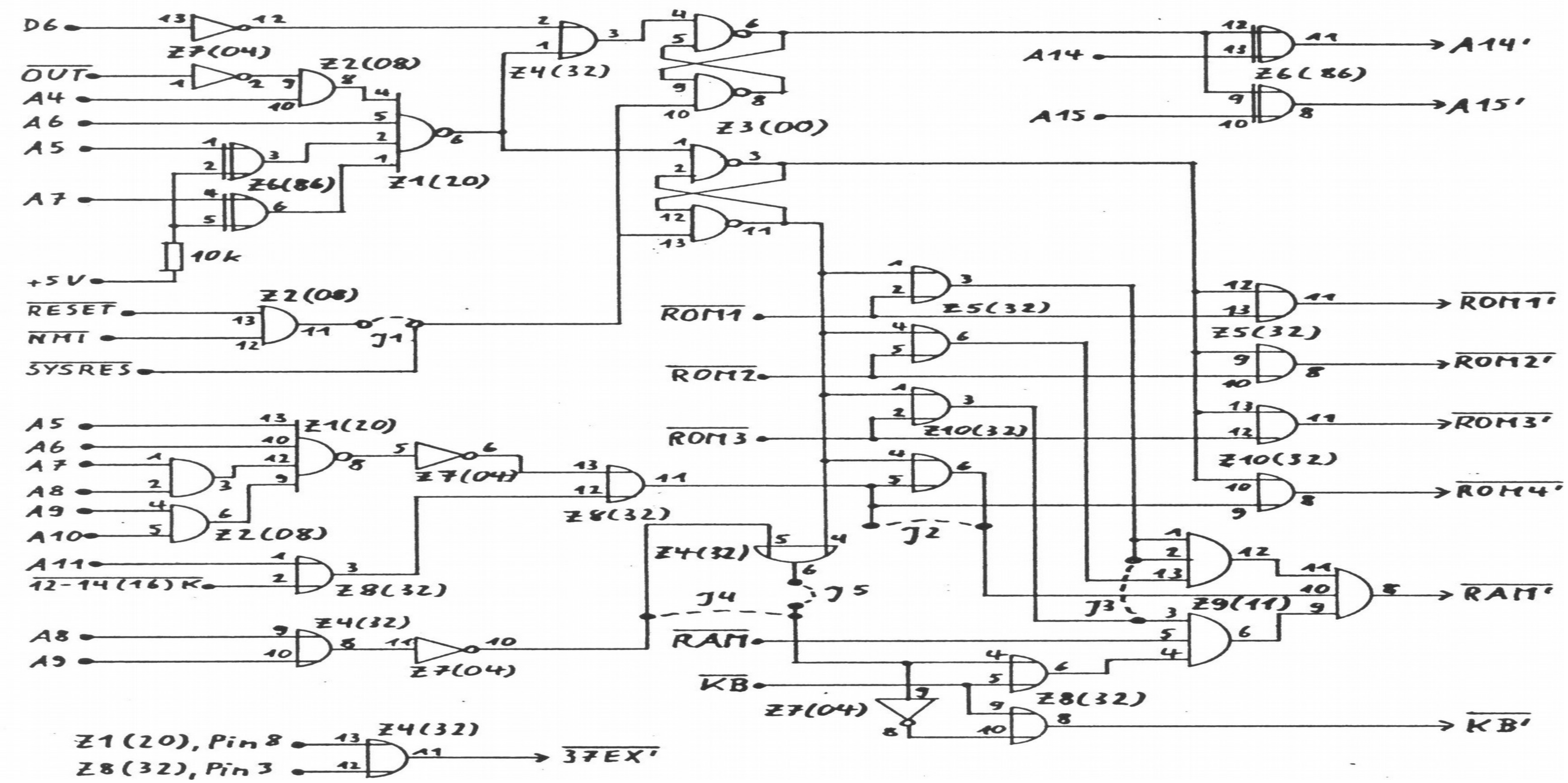
H. Bernhardt, C. Ueberschaar

Um den Betrieb unter CP/M zu ermöglichen, sind folgende Anforderungen an die Hardware zu stellen. Nach dem Booten durch die Level II ROMs müssen die ROMs per Software abgeschaltet und durch RAM ersetzt werden können. Der Zugriff auf Bildwiederholspeicher, Tastatur, FDC und Drucker muß auf den oberen Adreßraum (direkt unter FFFFH) verlegt werden. Der gesamte übrige Adressraum muß als RAM zur Verfügung stehen.

Dieses Problem ist re1ativ einfach zu lösen, vorausgesetzt daß die 4116 RAMs auf dem CPU-Board durch 4164 RAMs ersetzt sind und die dafür nötigen Änderungen vorgenommen wurden (wie dies beim GENIE ab Modell 1983 schon der Fall war oder durch Speichererweiterung auf "64K RAM" für ältere Modelle nachrüstbar war). Um diese Erweiterung auch bei TRS 80 machen zu können, ist zusätzlich auf dem CPU-Board die Änderung gemäß Abb. 7 durchzuführen. Bei allen Geräten müssen die eventuell im Expansion Interface vorhandenen 4116 RAMs entfernt werden.

Durch Invertieren der Adressen A14 und A15 für die Decoderschaltung kann erreicht werden, daß ROMs und I/O von 0-16K nach 48-64K verlegt werden. Durch Umschalten der ROM-Freigabesignale an die RAM-Freigabe läßt sich dann zusätzlich noch der Adreßbereich C000H bis F7DFH als RAM gewinnen. Diese Aufgabe übernimmt die in Abb. 1 dargestellte Schaltung.

Abb. 1: Schaltung des CP/M-Boards



Kernstück dieser Schaltung sind zwei aus NAND-Gattern gebildete D Flip Flops, die bei Einschalten (RESET*) oder Drücken der "RESET"-Taste (NMI*) immer so eingestellt werden, daß die ROMs freigegeben werden und zusammen mit dem meomry-mapped I/O im Adressbereich 0-16K* liegen (TRS 80 Modus).

Durch Ausgabe einer Portadresse von 50H bis 5FH wird über die Decoderschaltung (IC1, 74LS20, Pin 6) immer das obere Flip Flop umgekippt, so daß die ROMs abgeschaltet werden und in ihren Adreßbereich dann RAM Speicher vorliegt. Wenn gleichzeitig das Datenbit 0 high ist, wird auch das andere Flipp Flop umgekippt und dadurch per Invertieren von A14 und A15 für die Decoderschaltung des CPU-Boards der I/O-Bereich an das obere Speicherende gelegt.

In dieser für CP/M nötigen Konfiguration liegt außerdem noch der im TRS 80 nicht verfügbare Adreßbereich zwischen ROM und I/O, der im GENIE als Sonder-ROM genutzt wird, als RAM vor. Um beim TRS 80 diesen Bereich (3000H-37DFH) im TRS 80 Modus auch als RAM nutzen zu können, darf IC10 (74LS32) nicht bestückt werden und stattdessen müssen die Jumper J2 und J3 gelegt werden (IC10 kann beim TRS 80 deswegen entfallen, weil nur zwei ROMs vorhanden sind.)

Beim GENIE ist der Bereich 3700H-37DFH (bzw. F700H-F7DFH im CP/M-Modus) wegen der unvollständigen Decodierung der Freigabe der Speicherlesetreiber auf dem CPU-Board nicht verfügbar. Das läßt sich dadurch beheben, daß das Signal 37EXH von Pin 11, IC4 (74LS32) auf dem CP/M-Board an Pin 12 von Z21 (74L20) auf dem CPU-Board angeschlossen wird und die bisher an diesen Pin führende Leitung durchtrennt wird. Durch freie Verdrahtung auf der Lötseite des CP/M-Boards müssen außerdem noch folgende Verbindungen hergestellt werden:

IC1 (74LS20), Pin 8 mit IC4 (74LS32), Pin 13

IC8 (74LS32), Pin 12 mit IC4 (74LS32), Pin 12

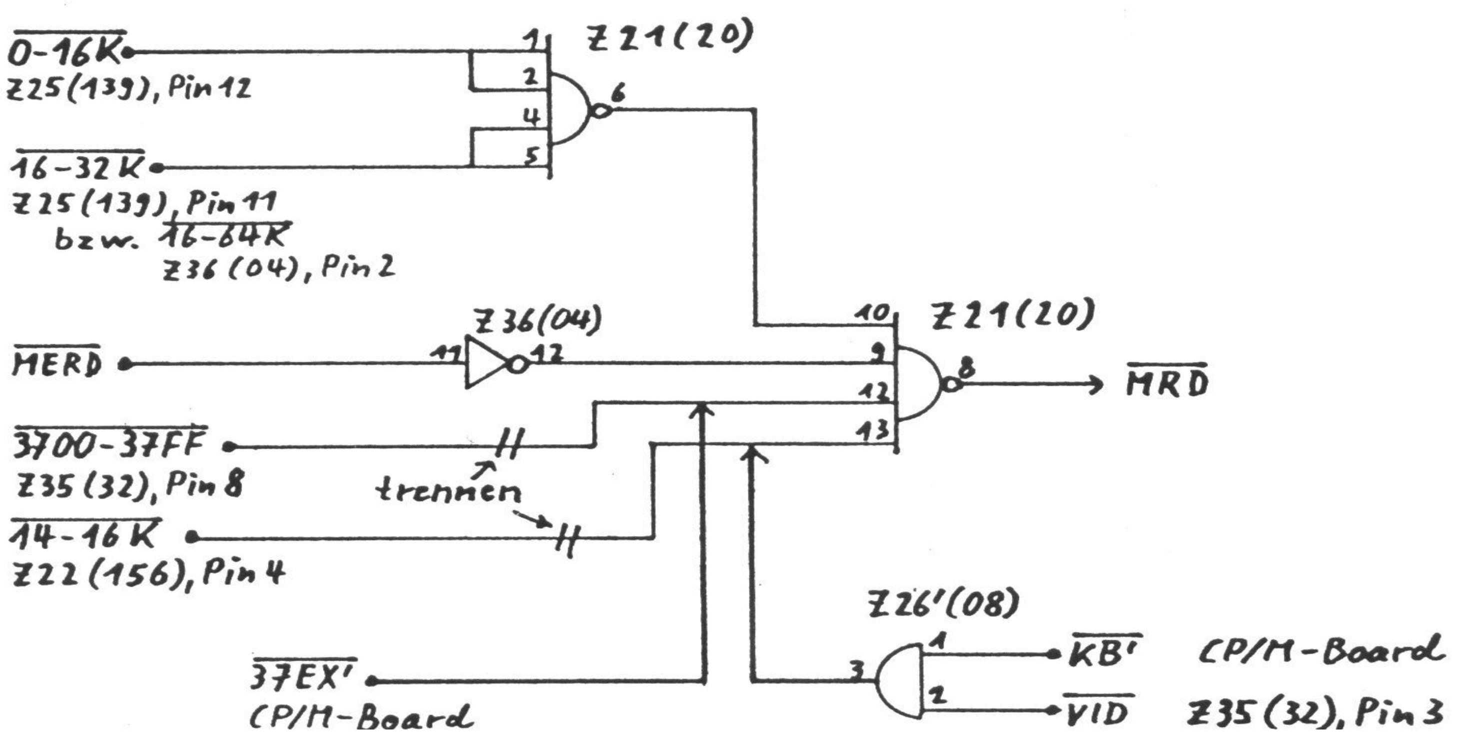
Das dadurch an Pin 11 von IC4 (74LS32) gewonnene Signal 37EX* kann außerdem auch für die Freigabe des 74LS155 Decoders auf dem Expansions Interface benutzt werden, wenn anstelle der beiden Leitungen A14' und A15' nur eine Leitung dorthin geführt werden soll. Der bisherige Anschluß an die Pins 2 und 15 des 74LS155 muß dann natürlich abgetrennt werden.

Die invertierten Adreßleitungen A14' und A15' dürfen dann nicht an das Expansion Interface gelegt werden, wenn dort zuätzliche Hardware Erweiterungen angeschlossen werden, die mit den richtigen Signalen von A14 und A15 versorgt werden müssen.

Im CP/M Modus sind außerdem noch 3/4 des Adreßbereichs von 1K der Tastatur, der im TRS 80 und GENIE durch unvollständige Decodierung verschenkt wird, als RAM verfügbar. Es sind dies die Speicheradressen F900H-FBFFH. Für den TRS 80 Modus wurde die Möglichkeit gelassen, durch die Juper J4 und J5 entweder im Bereich 3900H-3BFFH RAM zur Verfügung zu haben oder die unvollständige Tastatur Decodierung beizubehalten. Es ist nicht auszuschließen, daß Programme die Tastatur auch im Bereich 3900 bis 3BFFH auslesen. Beim Komtek 1 ist die Tastatur bereits voll decodiert.

Mit einem Doppelumschalter kann (bei grundsätzlicher Einstellung von Speicher) im Bedarfsfall auf Tastatur von 3800H-3BFFH zurückgeschaltet werden, wenn man sich diese Rückzugsmöglichkeit offen halten möchte.

Abb. 2: Freigabesignal des Speicherlesetreibers im GENIE, Änderungen beim Einbau des CPM-Boards

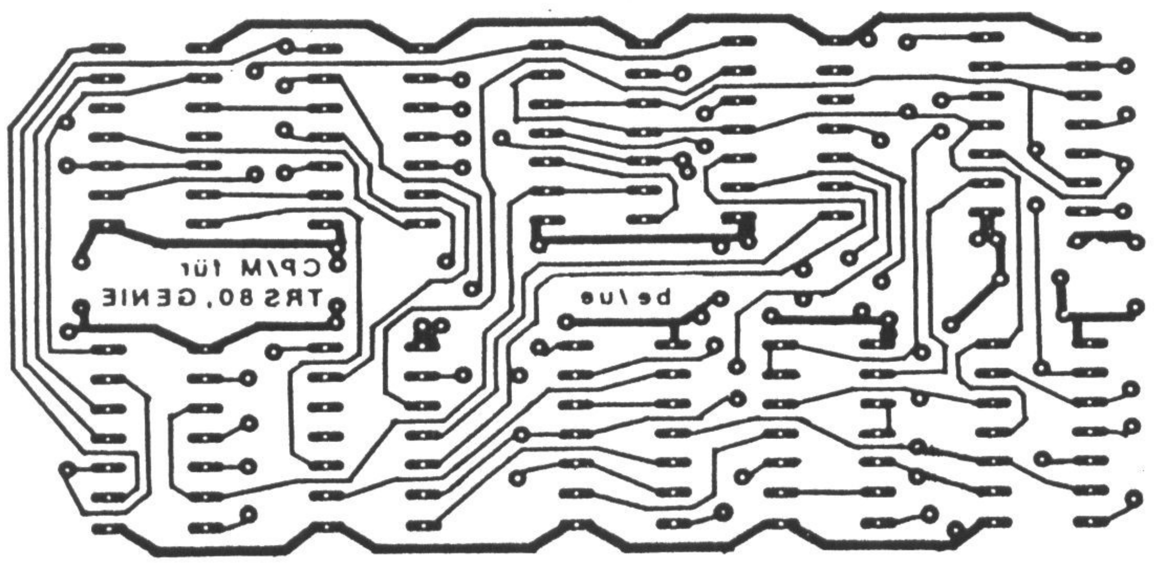


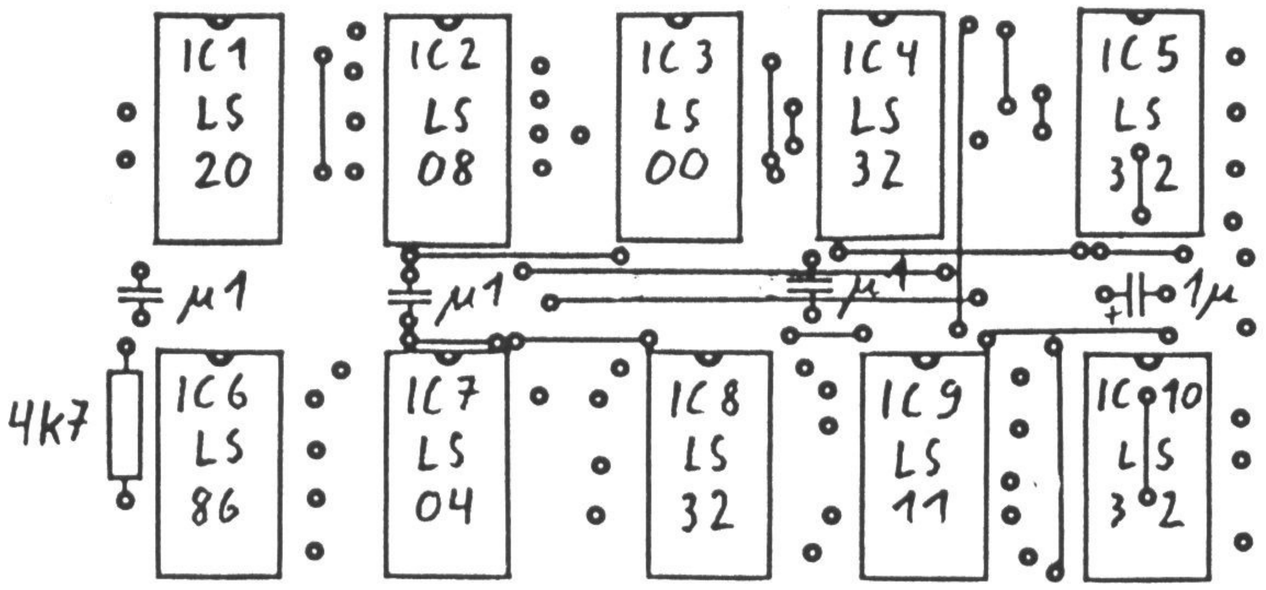
Um beim GENIE diesen Speicherbereich nutzen zu können muß durch AND-Verknüpfung der Signale KB'* des CP/M-Boards und VID*, Pin 3 von Z35 (74LS32) auf dem CPU-Board und Anlegen des dadurch gebildeten Signals an Pin 13 von Z21 (74LS20) auf dem CPU-Board (Abb.2) der Speicherlesetreiber freigegeben werden. Das dafür nötige IC 74LS08 kann auf den freien Platz neben Z25 (74LS139) eingesetzt werden.

Im TRS 80-Modus ist durch die getrennte Umschaltung der Flip Flops auch noch die Möglichkeit gegeben die ROMs abzuschalten und durch RAMs zu überblenden. Das läßt sich dazu nutzen, um den ROM-Inhalt durch ein kleines Maschinenprogramm zunächst in den oberen Speicherbereich zu kopieren, dann durch Ausgabe von 00H an eine der Portadressen 50H-5FH von ROM auf RAM umzuschalten und den ROM-Inhalt dann wieder in seinen ursprünglichen Bereich zurückzuschreiben. Der dann unter gleichen Adressen im RAM vorliegende ROM-Inhalt kann dann beliebig geändert werden. Um Schwachstellen der ROM-Treiberroutinen (z.B. Unterdrücken bestimmter Steuercodes für die Druckerausgabe) zu umgehen, können dann diese Routinen direkt gepatched werden und müssen nicht durch zusätzliche Maschinenprogramme im oberen Speicherbereich ersetzt werden.

Abb. 3: Layout, Ansicht Bestückungsseite

Abb. 4: Bestückung, Drahtbrücken





Das in Abb.3 vorgeschlagene Layout sowie der Bestückungsplan und die externe Beschaltung (Abb.4 und 6) bauen auf eine leicht selbst herzustellende einseitig gedruckte Schaltung mit relativ wenigen Drahtbrücken auf. Der Jumper J1 zwischen IC2 und IC3 ist nur bei Anschluß an den GENIE zu setzen wo anstelle des nicht vorhandenen SYSRES* Signals die Signale RESET* und NMI* herangezogen werden müssen.

Funktion der Jumper

	TRS 80	**GENIE und Komtek 1**
J1	bleibt offen, wenn SYSRES* benutzt wird und nicht RESET* und NMI*	muß gelegt werden, weil SYSRES* nicht verfügbar ist
J2, J3	müssen gelegt werden, IC 10 darf nicht bestückt werden	bleiben offen, IC 10 muß bestückt werden
J4, J5	gilt für alle Geräte	
	J4 gelegt und J5 offen: 3900H-3BFFH ist auch im TRS 80 Modus als RAM verfügbar	
	J4 offen und J5 gelegt: Im TRS 80-Modus belegt die Tastatur den Bereich 3800H-3BFFH (außer bei Komtek 1)	
	Mit einem Doppel Umschalter anstelle einer festen Einstellung mit Jumpern kann zwischen beiden Varianten umgeschaltet werden	

Abb. 5: Lage der Jumper

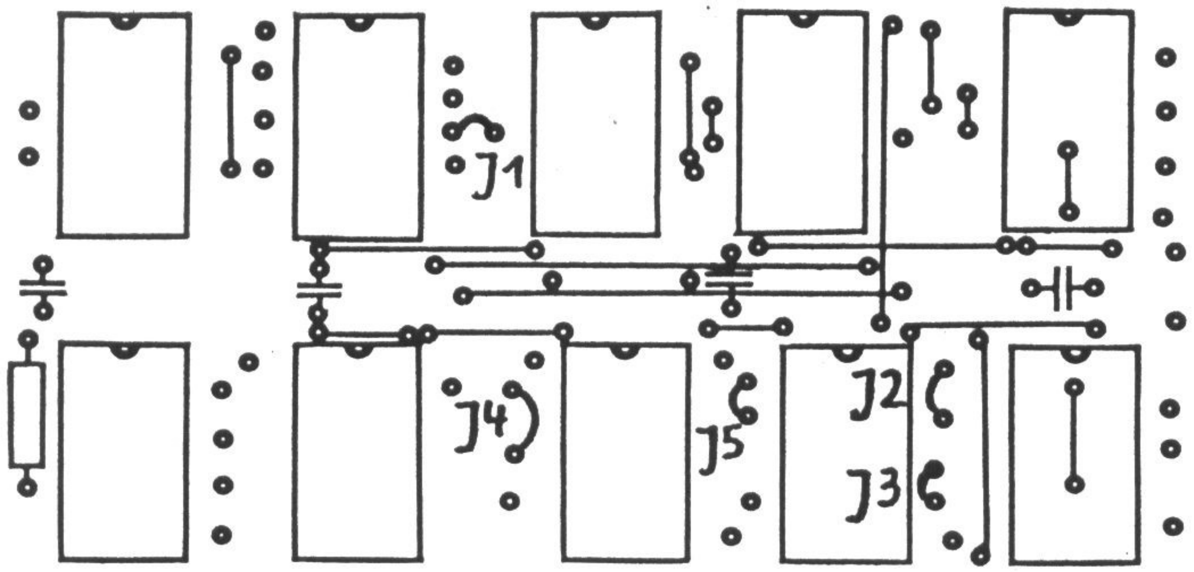
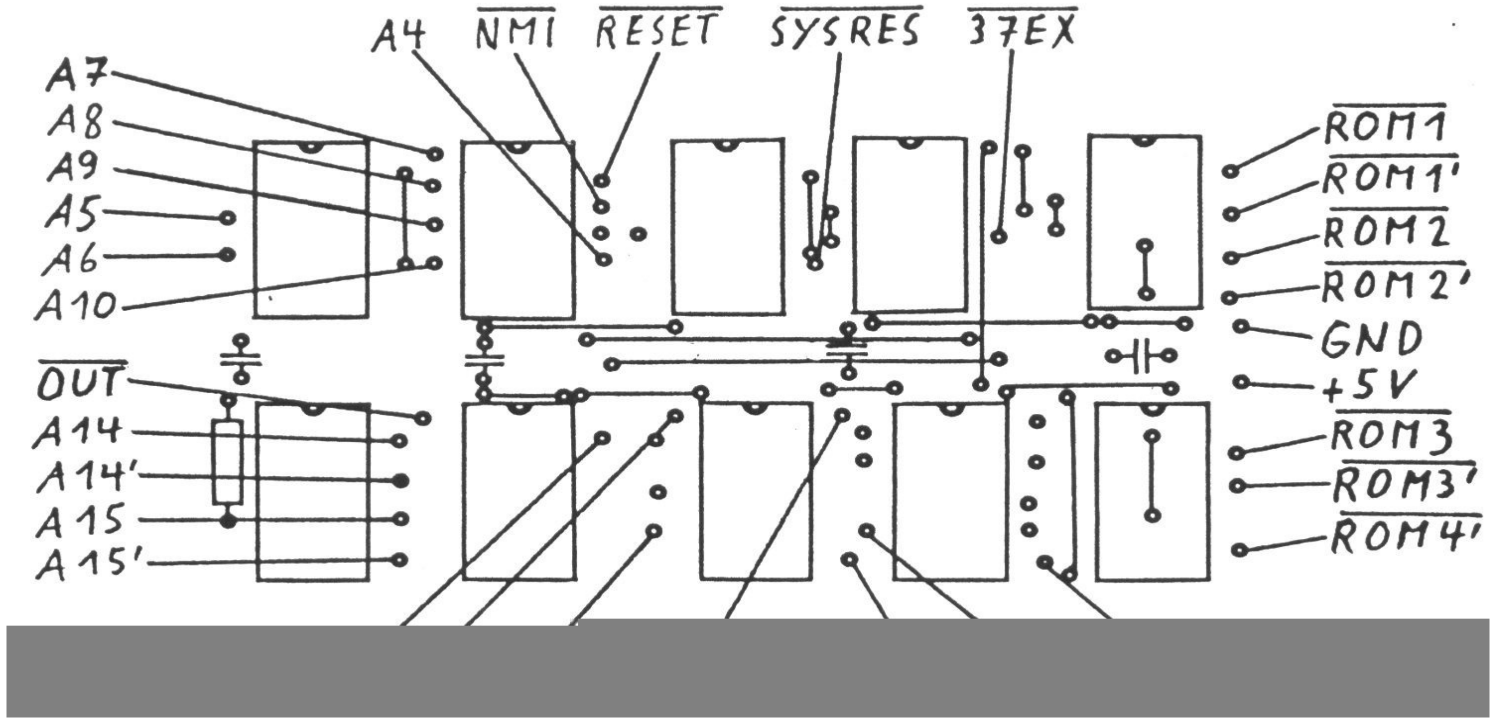


Abb. 6: Anschluß an das CPU-Board



Signal auf dem	**TRS 80**			**GENIE**			**Komtek 1<**		#4
CP/M-Board	IC	Typ	Pin	IC	Typ	Pin	IC	Typ	Pin
A4	Z39	367	7	Z4	367	9	42	2732	4
A5	"	"	9	"	"	7	"	"	3
A6	"	"	5	"	"	11	"	"	2
A7	"	"	11	"	"	5	"	"	1
A8	"	"	3	"	"	13	"	"	23

A9	"	"	13		"	"	3		"	"	22	
A10	Z38	"	3		Z6	"	3		"	"	19	
A11	"	"	13		"	"	13		"	"	21	
A14	"	"	11		"	"	7		33	Z80	4	
A15	"	"	9		"	"	9		"	"	5	
A14'	Z73	32	5	#3	Z25	139	14	#1	53	139	2	#1
A15'	"	"	4	#1	"	"	13	#1	"	"	3	#1
D6	Z75	367	9		Z18	367	13		42	2732	16	
12-14(16)K*	Z21	156	12		Z22	156	5		49	139	1	
OUT*	Z23	32	3		Z15	32	8		38	32	6	
SYSRES*	Z37	02	1		--------------------				--------------------			
RESET*	--------------------				Z2	05	6		33	Z80	26	
NMI*	--------------------				"	"	8		"	"	17	
ROM1*	Z21	156	9		Z22	156	9,10	#1	48	32	11	
ROM2*	"	"	11		"	"	11,12	#1	"	"	3	
ROM3*	--------------------				"	"	6,7	#1	"	"	6	
ROM1'*	Z33	ROMA	20	#1	Z10	3001	20,21	#1	42	2732	18	#1
ROM2'*	Z34	ROMB	20	#1	Z11	3002	20,21	#1	47	"	"	#1
ROM3'*	--------------------				Z12	3003	20,21	#1	52	"	"	#1
ROM4'*	--------------------				Z13	2716	18	#2	57	"	"	#1
RAM*	Z21	156	5	#3	Z36	04	2	#1	(siehe Text)			
RAM'*	Z37	367	15	#1	Z37	367	15	#1	54	32	1	#1
	Z74	00	10	#1					59	245	19	#1
KB*	Z36	32	11	#1	Z35	32	11	#1	49	139	6	
KB'*	Z3	368	1		Z8	368	1		--------------------			

Anschluß des CP/M-Boards an das Expansion Interface

Signal	TRS 80	GENIE	EXP1 von RB Eletronik
	Exp.Intfc. Kabel	Exp.Intfc. Kabel	keine
A14'	Leitung 10 #1	Leitung 23 #1	Änderung
A15'	Leitung 7#1	Leitung 21 #1	nötig
oder stattdessen:			
37EX'	Z39(74LS155)2,14 #1	Z29(74LS155)2,14 #1	Z18(74LS155)2,14 #1

#1 bisherige Leitungen abtrennen
#2 beide Leitungen abtrennen und miteinander verbinden, ohne daß Pin 38 mit daran liegt
#3 die bei der Erweiterung auf 64K mit 4164 ICs gelegte Verbindung muß wieder entfernt werden
#4 die Nummerierung der ICs des CPU-Boards erfolgt gemäß dem Bestückungsplan aus; Info des TRS 80 / GENIE User Club, Hardware Sonderheft; 'Wie aus dem Komtek 1 ein Computer wird

Herleitung eines richtigen Freigabesignals für GENIE und Komtek 1

Das zur RAM-Freigabe beim Umrüsten auf 64K RAM auf dem CPU-Board in den mitgelieferten Anleitungen vorgeschlagene invertierte Signal 0-16K kann bei bestimmten I/0-Befehlen eventuell Schwierigeiten machen. Deshalb sei hier eine Möglicheit beschrieben, wie mit nicht benutzten Gattern auf dem CPM-Board dieser Fehler behoben werden kann.

1) GENIE

1. Das Signal 12-16K* wird nicht an Pin 2 sondern an Pin 12 von IC 8 (LS32) gelegt.
2. Die Verbindung zwischen Pins 3 und 12 von IC 8 (LS32) wird durchtrennt.
3. Das Signal 0-16K* (CPU-Board, Z25, LS139, Pin 12) wird n Pin 3 von IC 7 (LS04) gelegt.
4. Pin 4 von IC 7 (LS04) wird mit Pin 2 von IC 8 (LS32) verbunden.
5. Anstelle von A11 wird MERQ* (CPU-Board, Z25, LS139 Pin 15) an Pin 1 von IC 8 (LS32) angeschlossen.
6. Pin 3 von IC 8 (LS32) wird mit Pin 5 von IC 9 (LS11) verbunden (das bisherige Signal RAM* vom CPU-Board wird abgetrennt).

2: Komtek 1

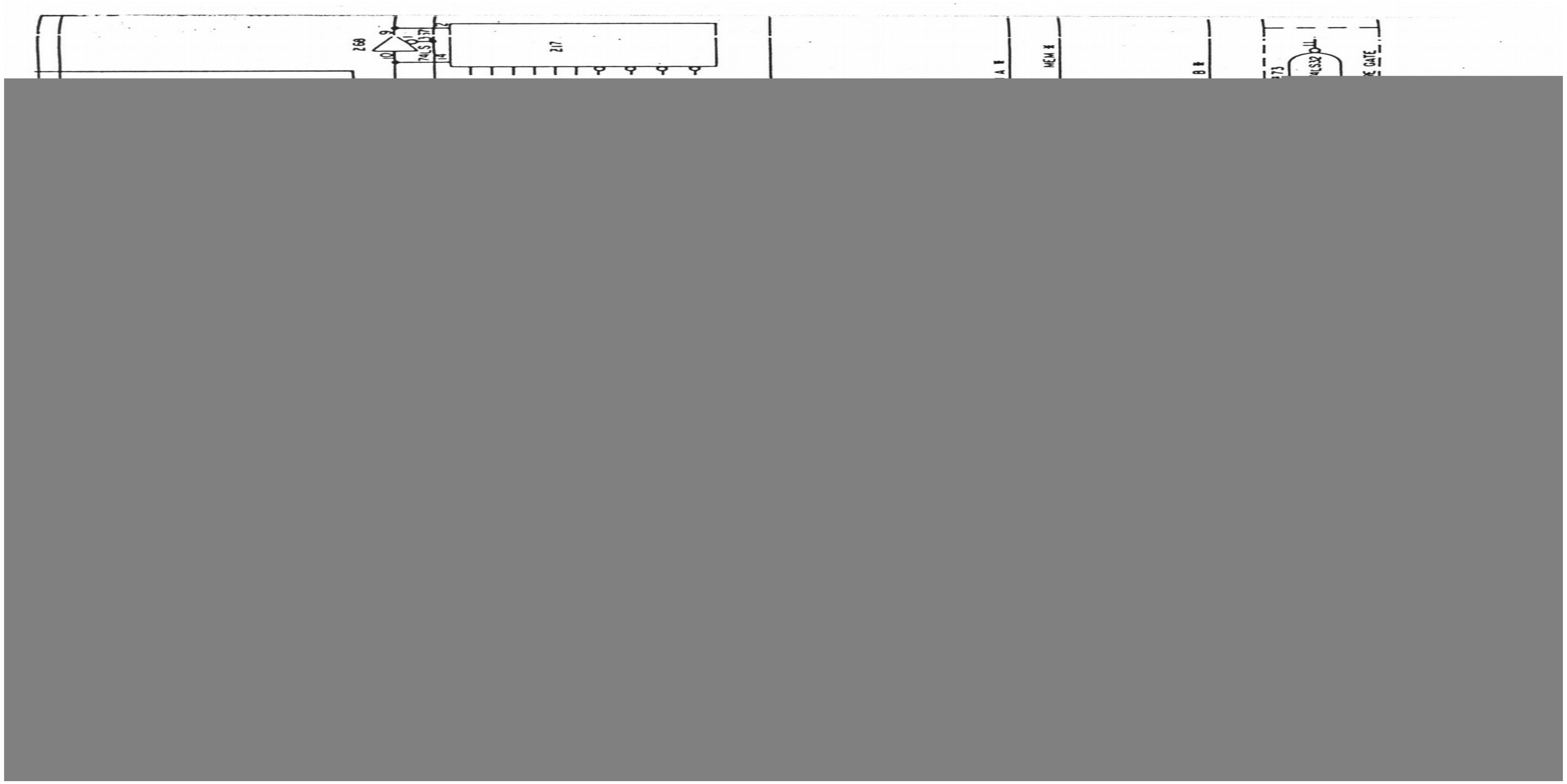
- Die Verbindung zwischen Pins 5 und 9 von IC 8 (LS32) wird durchtrennt.
- Die Leiterbahn zu Pin 9 von IC 7 (LS04) wird durchtrennt.
- Das SignaJ 0-16*K (CPU-Board, Z53, LS139 Pin 4) wird an Pin 9 von IC 7 (LS04) gelegt.
- Das Signal MERQ* (CPU-Board, Z53, LS139 Pin 1) wird an Pin 9 von IC 8 (LS32) angesch1ossen.
- Pin 8 von IC 8 (LS32) wird it Pin 5 van IC 9 (LS11) verbunden (das Signal KB' ist auf dem CPU-Board nicht nötig, da die Tastatur voll decodiert ist).

Umrüsten des TRS 80 auf 64K auf dem CPU-Board

1. Die Pins 1, 8, und 9 aller RAM5 (Z14 - Z20) gemeinsam von den Versorgungsspannungen +12V, -5V und +5V abtrennen
2. +5V an die Pins 8 aller RAMs legen
3. Alle Kondensatoren die an einer der die Pins 1 oder die Pins 9 verbindenden Leitungen liegen, auslóten
4. A14 (Z38, 74LS367, Pin 11) an Pin 14 von Z51, 74LS157 legen
5. A15 (Z38, 74LS367, Pin 9) an Pin 13 von Z51, 74LS157 legen
6. Pin 12 von Z51, 74LS157 mit den Pins 9 der RAMs verbinden
7. Die Leiterbahn von/zu folgenden Pins durchtrennen
   Z21, 74LS155, Pins 2 und 14 sowie 1 und 15
   Z73, 74LS32, Pin 5
8. folgende Pins miteinander verbinden
   Pin 5 von Z73, 74LS32 mit Pin 11 von Z38, 74LS367: A15
   Pin 6 von Z73, 74LS32 mit Pins 1 und 15 von Z21, 74LS156: 0-16K*
   Pin 14 von Z68, 74LS367 mit Pins 2 und 14 von Z21, 74LS156: RAS*

9. Im Jumperfeld X3 müssen folgende Positionen miteinander verbunden sein, 2-15, 3-14, 4-13, 5-12, 6-11, 7-10, 8-9
10. Im Jumperfeld X71 müssen nur die Brücken 1-16, 3-14 und 5-12 gesetzt sein
11. Im TRS 80 Expansion Interface muß die Verbindung zwischen Pin 6 von Z28, 74LS00 und Pin 19 von Z31, 74LS244 durchtrennt werden.

Abb. 7: Änderungen auf dem CPU-Board des TRS 80



256K RAM für Z80-Systeme

Helmut Bernhardt

"64K-Byte RAM sind die obere Grenze für den vom Z80 adressierbaren Speicher". Diese Behauptung ist nur dann richtig, wenn die direkte Adressierung gemeint ist. Durch Banking d.h. Umschalten eines Teilbereichs des 64K-Adreßraumes auf parallele Speicher läßt sich der vom Z80 nutzbare Speicher beliebig ausbauen; 1M-Byte-RAM-Karten werden heute für fast alle Z80-Systeme angeboten.

Gegenüber solchen Karten hat die im folgenden beschriebene Speichererweiterung den Vorteil, daß sie für wenig Geld zu realisieren ist und weder an einen bestimmten Busstandard noch gar einen speziellen Computer gebunden ist. Einzige Voraussetzung ist ein Z80-Prozessor.

Der Nachteil des Projektes besteht darin, daß man sich auf dem CPU-Board ein bißchen auskennen muß und daß es ohne Lötkolben doch nicht geht.

Das Grundprinzip der Speichererweiterung ist ein Ersetzen der vorhandenen 4164-RAMs (oder auch 4116-RAMs) durch die pinkompatiblen 41256-Typen und eine mit wenigen Standard-ICs aufgebaute Banking-Logik, mit der ein gezielter Zugriff auf die vielen K-Byte möglich ist.

Die Banking-Logik

Vom Z80 aus gesehen ist der direkt adressierbare Speicherraum von 64K-Byte in zwei Blocks von 32K-Byte Größe unterteilt. Die obere Hälfte (Adressen 8000H-FFFFH) ist dabei nicht umschaltbar und deshalb vom Z80 immer erreichbar (Common-Bereich). Interrupt-Service-Routinen, der Stack und alle Routinen, auf die ständig zugegriffen werden muß, müssen hier liegen.

Die untere Hälfte ist in mehreren parallelen 32K-Blocks vorhanden. Welcher dieser Blocks im Adreßraum 0000H-7FFFH für den Z80 jeweils verfügbar sein soll, läßt sich durch Ausgabe der Nummer des Blocks (Bank Nr.) an einen Port (Latch) einstellen. Der Block ist dann solange als Speicher verfügbar, bis eine andere Bank Nr. ausgegeben wird. Durch RESET* wird beim Einschalten des Computers automatisch immer die Bank 0 eingestellt, um bei der Systeminitialisierung definierte Verhältnisse vorzugeben.

Von der Aufteilung der 256K (512K, 1024K) RAM her gesehen sind die Verhältnisse etwas anders zu sehen. Die unteren 64K sind nach dem Einschalten des Computers diejenigen, auf die der Z80 zugreifen kann. Alle anderen 32K-Blocks können anstelle des ursprünglichen 32K-Blocks in die untere Hälfte des Adreßraumes des Z80 gelegt werden, wenn eine der Bank Nummern 1-7 (15, 31) ausgegeben wird.

Das wird dadurch erreicht, daß durch die Latchausgänge Q1 und Q2 die Adressen A16 und A17 vorgegeben werden, die einen der vier 64K-Blocks innerhalb eines 256K-Blocks anwählen, Q3 und Q4 codieren einen der bis zu vier 256K-Blocks und Q0 und A15 des Systembus werden dafür herangezogen, die Adresse A15' für den Speicher zu liefern, um dort die untere oder obere Hälfte eines 64K-Blocks zu adressieren.

Die Herleitung von A15' über OR-Verknüpfung von A15 und Q0 sorgt dafür, daß immer dann, wenn A15 high ist, die obere Hälfte des 64K-Blocks angewählt wird (Common-Bereich), während dann, wenn A15 low ist, der Pegel des Latchausgangs Q0 den Pegel von A15' bestimmt und so zur richtigen Anwahl der eingestellten Bank beiträgt.

Da nun aber bei einer Adresse im Common-Bereich (A15=1) nicht nur die obere Hälfte irgendeines durch Q1 bis Q4 codierten 64K-Blocks im RAM sondern ausschließlich die obere Hälfte des unteren 64K-Blocks angewählt werden soll, wird durch den Pegel von A15 am STB*-Eingang des 74LS157-Multiplexers (Abb.1) Low-Pegel ausgegeben.

Der Rest der Schaltung erfüllt die Aufgabe, zu erkennen, ob Bank Nr.1, die den gleichen 32K-Block im RAM wie der Common-Bereich belegen würde, eingestellt ist, und daraufhin die höchste verfügbare Bank zu adressieren. Das geschieht dadurch, daß bei eingestellter Bank Nr.1 das Bitmuster an den Latchausgängen zu einem High-Pegel am Select-Eingang des Multiplexers führt, der dann die Pegel der B-Eingänge, die über 4k7 an +5V gelegt sind, durchschaltet. Diese Sicherung des Common-Bereichs führt dazu, daß unter den Bank-Nummern 1 und 7 (15, 31) der gleiche 32K-Block adressiert wird. Die Bank 7 (bei 256K), 15 (bei 512K) bzw. 31 (bei 1024K) sollte also nicht benutzt werden.

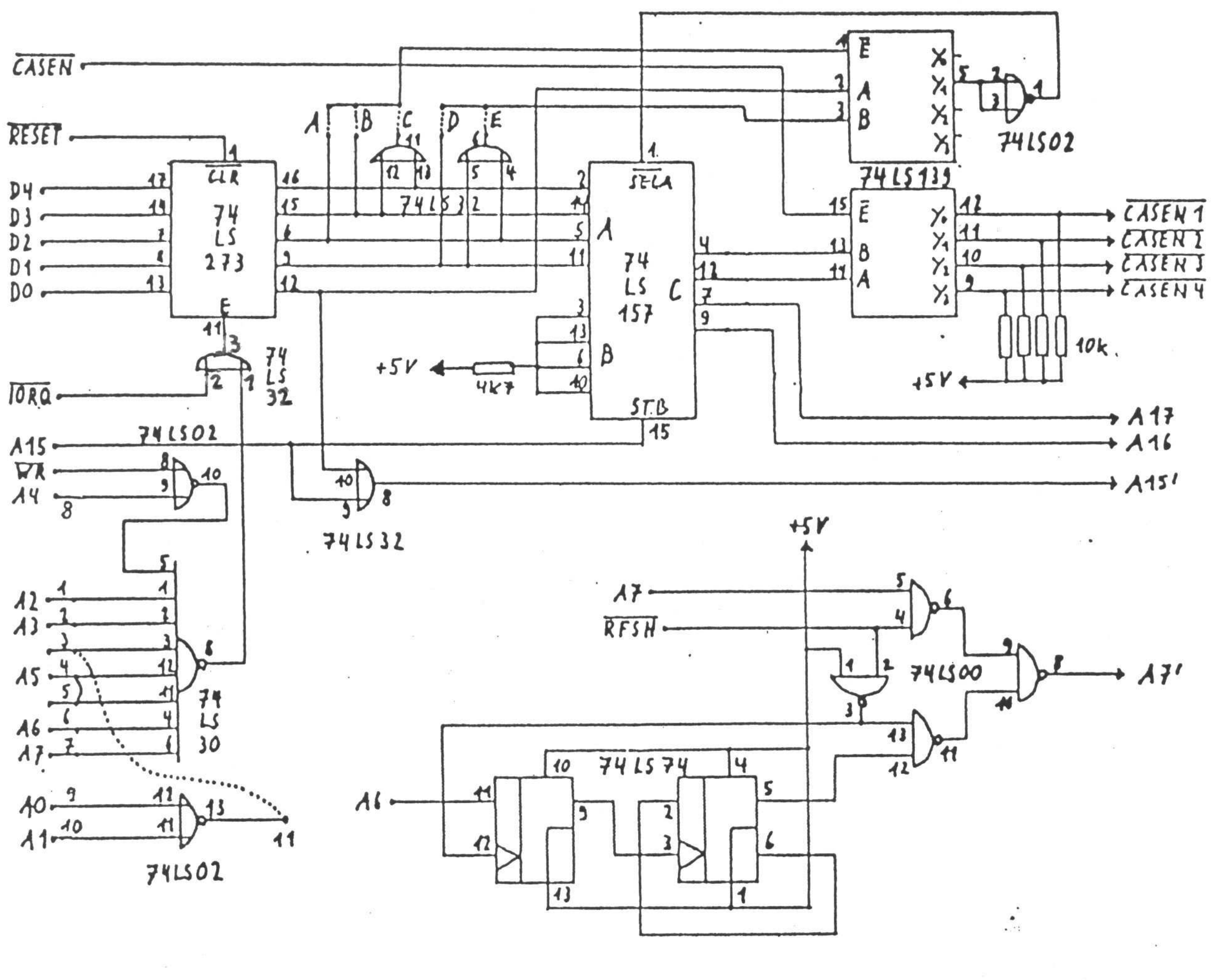


Abb. 1 (oben): Schaltung des Banking-Boards und Generierung einer 8-Bit Refresh-Adresse

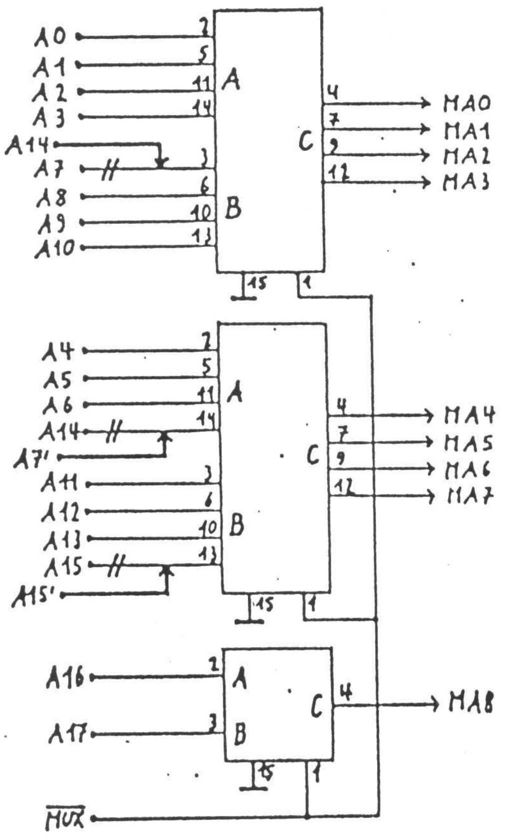


Abb. 2 (links): Änderung und Erweiterung des Adreßmultiplexers für die RAM- Adressen am hypothetischen Beispiel eines Computers, der schon von 4116- RAMs auf 4164-RAMs umgerüstet worden ist. Die Belegung der A-Eingänge mit den niederen Adressen ist nicht zwingend (siehe mc-CP/M-Computer). Von einem zusätzlich nötigen 74LS157 wird nur 1/4 benötigt.

Mehr als 256K RAM lassen sich dann realisieren, wenn im Computer mehrere Reihen mit 8 Sockeln für RAMs vorhanden sind, oder indem die RAM-ICs übereinander gelötet werden, wobei dann Pin 15 (CAS*) des aufgelöteten ICs hochgebogen werden muß. Durch die Drahtbrücken A-E muß die Schaltung dem Speicherausbau angepaßt werden. Je nach Speichergröße sind folgende Drahtbrücken zu legen:

256K	1 Reihe RAMs	Brücken A und D
512K	2 Reihen RAMs	Brücken B und E
1024K	4 Reihen RAMs	Brücken C und E

Diese Drahtbrücken sorgen dafür, daß je nach Speicherausbau der durch Bank Nr.1 angewählte 32K-Block identisch ist mit dem durch die höchste mögliche Bank-Nummer freizugebenden Block.

Bei mehr als 256K RAM müssen die einzelnen 256K-Blocks durch die Signale CASEN1* bis CASEN4* (bzw. deren Verknüpfung mit CAS* über OR-Gatter) freigegeben werden. Bei nur 256K RAM braucht die CAS*-Freigabe, wie sie vorher für 64K vorgelegen hat, nicht geändert zu werden.

Wenn 512K RAM realisiert worden sind, muß auf dem Banking-Board die Leiterbahn zu Pin13 des 74LS139 durchtrennt werden, an diesen Pin ist statt dessen GND zu legen, um zu verhindern, daß bei Ausgabe der Bank-Nr.1 nicht die nicht vorhandene Bank31 anstelle der Bank15 eingestellt wird.

Das Refreshing

Ein weiteres Problem stellt neben der Adressierung innerhalb der 41256-RAMs noch das Refreshing dar. Gegenüber den meisten 4164-Typen mit 128 Refresh-Zyklen (nur einige Ausnahmen) benötigen 41256-RAMs grundsätzlich 256

Refresh-Zyklen. Der Z80 liefert bei seinem Hidden Refresh aber nur eine 7-Bit-Refresh-Adresse.

Dieses Problem ist aber von Michael Hungershausen (c't 9/84, S.22) schon gelöst worden und soll deshalb hier nicht weiter abgehandelt werden. Der Vollständigkeit halber ist in Abb.1 eine der beiden vorgestellten Varianten für die Herleitung einer 8-Bit-Refresh-Afresse wiedergegeben.

Der Adreßmultiplexer

Ebenfalls in Verbindung mit dem Refreshing ist folgendes Problem zu sehen. Durch den Zyklus von 128 Refresh-Adressen ist es nötig, daß beim Adreß-Multiplex die Adressen A0 bis A6 zusammen durchgeschaltet werden. Anstelle von A7 kann aber auch irgendeine höhere Adresse zusammen mit A0-A6 durchgeschaltet werden. Bei Computern, in denen ursprünglich 4116-RAMs verwendet wurden, und die dann spater auf 4164-RAMs umgerüstet wurden, muß sogar davon ausgegangen werden, daß dies der Fall ist, da für die interne Adressierung von 4116- RAMs die Adressen A0-A6 und A7-A13 abwechselnd durchgeschaltet werden.

Es muß bei den Adreßmultiplexern also ermittelt werden, ob A7 mit A0-A6 zusammen durchgeschaltet wird. Wenn dies nicht der Fall ist, muß ermittelt werden, wo A7 anliegt und wo eine höhere Adresse mit A0-A6 zusammen durchgeschaltet wird. Beide Signale sind von den Pins der Multiplexer abzutrennen; an den Pin, wo vorher die höhere Adresse anlag, ist A7' (Abb.1) zu legen und die dort abgetrennte Adreßleitung ist dahin zu führen, wo vorher A7 anlag.

Außerdem muß durch ein huckepack-aufgelötetes 74LS157 (auf einen der schon vorhandenen Multiplexer; bei 74LS157 können die Pins 1,8,15 und 16 direkt angelötet werden) noch das Multiplexen von A16 und A17 möglich gemacht werden. Der Ausgang MA8 (Abb.2) wird (gegebenenfalls über einen 33-Ohm-Widerstand) an die Pins 1 der RAMs geführt.

Anstelle von A15 muß das Signal A15' (Abb.1) an die Adreßmultiplexer geführt werden.

Bei Computern mit 4116-RAMs

müssen alle Speicherchips entfernt werden, und bei Vorhandensein getrennter Daten-Treiber für verschiedene 16K-Blöcke muß auch die Freigabe der Treiber nicht benutzter RAM-Reihen durch Anlegen von +5V über einen 4K7-Widerstand an den E*-Eingang der Treiber verhindert werden.

Da diese Speicherchips mit -5V an Pin 1, +12V an Pin 8 und +5V an Pin 9 versorgt werden, müssen diese Leitungen von den Versorgungsspannungen abgetrennt werden, Die Versorgungsspannung +5V wird an die Pins 8 gelegt. Bei den die Pins 1 und die Pins 9 verbindenden Leitungen werden alle Abblockkondensatoren entfernt. An die Pins 9 wird MA7 und an die Pins 1 wird MA8 der Adreßmultiplexer gelegt.

Das CAS*-Signa1 an Pin 15 der 4116-RAMs ist ein mit einem 16K-Freigabesignal (meistens durch OR, 74LS32) verknüpftes CAS*. Anstelle des 16K-Freigabesignals muß ein low aktives (bei OR-Verknüpfung) Freigabesignal, das den gesamten als RAM vorgesehenen Adreßraum des Z80 umfaßt (nur ROM und memory-mapped I/O ausgeblendet) oder bei Speichererweiterung auf mehr als 256K eines der Signale CASEN1* - CASEN4* verwendet werden.

Banking der oberen 32K

ist nur dann sinnvoll, wenn es sich um einen BASIC-im-ROM-Computer, der nicht CP/M-fähig ist, handelt, und läßt sich durch Invertieren von A15 (durch z.B. 74LS04) für die Schaltung in Abb. 1 erreichen.

Einbau des Banking-Boards

Die Wahl der Portadresse für das Latch zum Einstellen der Bank Nr. wurde flexibel gehalten. Die Portadresse kann sich aus maximal 7, minimal 5 high aktiven und maximal 3, minimal 1 low aktiven Adreßbits zusammensetzen.

Low aktive Adreßleitungen werden an die Punkte 8 bis 10 angeschlossen. Der Punkt 8 muß auf alle Fälle belegt werden. Wenn dann nur noch eine zusätzliche Adreßleitung low sein soll, muß diese an beide Punkte (9 und 10) gelegt werden. Wenn die Punkte 9 und 10 belegt werden, muß Punkt 11 mit einem der Punkte 1-7 verbunden werden. Die high aktiven Adreßleitungen werden an die restlichen Punkte (der Punkte 1-7) gelegt. Nicht belegte Punkte (der Punkte 1-7) werden mit belegten Punkten verbunden. Selbstverständlich müssen A6 und A7 auch noch an die dafür vorgesehenen Punkte (n und m) neben dem 74LS00 angeschlossen werden.

Die Adreßdekodierung sei am Beispiel der Portadresse ECH genauer beschrieben: ECH hat folgendes Bitmuster:

A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	1	1	0	0

Es müssen dann A0, A1 und A4 als low aktive Adreßleitungen an die Punkte 8, 9 und 10 gelegt werden und wegen der Belegung der Punkte 9 und 10 muß Punkt 11 mit z.B. Punkt 3 verbunden werden. Die Adreßleitungen A2, A3, A5, A6 und A7 werden dann z.B. an die Punkte 1, 2, 4, 6 und 7 angeschlossen. Der unbenutzte Punkt 5 muß dann mit z.B. Punkt 4 verbunden werden.

Das Signal CASEN* ist das CAS*-Freigabe-Signal für einen mit 4164-RAMs bestückten Speicher. Wenn vor dem Umrüsten auf 41256-RAMs noch 4116-RAMs vorhanden waren, muß ein solches Signal noch hergeleitet werden. CASEN* muß bei allen Adressen, die weder ROM noch memory-mapped I/O ansprechen, low aktiv sein. Bei TRS 80 und Kompatiblen ist dies ein Signal 16-64K*, das durch Invertieren des Signals 0-16K* und -leider nicht überall durchgeführt- OR-Verknüpfung mit MERQ* hergeleitet wird.

Die Freigabe des Datenbus-Transceivers

für die RAMs und eventuell auch ROMs kann in einigen Computern größere Schwierigkeiten bereiten. Ara einfachsten gestaltet sich die Lösung dieses Problems, wenn nur 8 Stück 4164-RAMs vorhanden sind und der zugehörige Datenbus-Transceiver ausschließlich die RAMs bedient, In diesem Fall kann die Freigabe des Transceivers

so, wie sie war, belassen werden.

Auch dann, wenn vorher mehrere Reihen 4116-RAMs vorhanden waren und später durch Umrüsten auf 4164-RAMs in einer Reihe und Freilassen aller anderen Sockel die Freigabe eines für alle RAMs gemeinsamen Datenbus-Transceivers auf die 4164-RAMs angepaßt wurde, braucht bei Verwendung nun wieder aller Sockel die Freigabe des Transceivers nicht mehr geändert zu werden.

Bei Vorhandensein je eines Transceivers für eine Reihe RAMs (ursprünglich 4116) und nun Bestückung von 2 oder 4 Reihen mit 41256-RAMs müssen zur Freigabe der Transceiver die Signale CASEN1* bis CASEN4* des Banking-Boards verwendet werden.

Wenn für ROM und RAM bisher ein gemeinsamer Datenbus-Transceiver bestanden hat, so kann dessen Freigabe nur bei bisheriger Bestückung mit 4164-RAMs beibehalten werden. In Fällen, wo für ROM und einen 16K-RAM-Block (z.B. ältere GENIE-Modelle) ein gemeinsamer Datenbus-Treiber besteht, kann nur diese eine Reihe RAM-Sockel verwendet werden und die Freigabe muß die vollen 64K abzüglich von Adreßbereichen für memory-mapped I/O umfassen. Wenn in solchen Fällen eine Umrüstung auf 4164-RAMs schon durchgeführt worden ist, ist die Freigabe des Treibers bereits den Anforderungen für den Betrieb von 41256-RAMs mit Banking gerecht.

Anschluß des Banking-Boards

Signal	Anschluß	TRS 80				GENIE I/II				Komtek 1				MC-CP/M-Comp.			
	hier	IC	Typ	Pin		IC	Typ	Pin		IC	Typ	Pin		IC	Typ	Pin	
D0	g	76	367	11		19	367	6		42	2732	9			Z80	14	
D1	f	"	"	13		"	"	10		"	"	10			"	15	
D2	e	75	"	13		"	"	4		"	"	11			"	12	
D3	d	"	"	5		"	"	13		"	"	13			"	8	
D4	c	"	"	7		"	"	11		"	"	14			"	7	
A0	8	55	"	11		17	"	9		"	"	8			"	30	
A1	9	"	"	13		"	"	7		"	"	7			"	31	
A2	1	22	"	11		"	"	5		"	"	6			"	32	
A3	2	"	"	13		"	"	3		"	"	5			"	33	
A4	10	39	"	7		4	"	9		"	"	4			"	34	
A5	4/5	"	"	9		"	"	7		"	"	3			"	35	
A6	6/m	"	"	5		"	"	11		"	"	2			"	36	
A7	7/n	"	"	11		"	"	5		"	"	1			"	37	
A15	h	38	"	9		6	"	9		33	Z80	5			"	5	
SYSRES*	a	37	02	1		5	Z80	26	#1	"	"	26	#1		"	26	
WR* (OUT*)	l	23	32	3		16	367	11		"	"	22			"	22	
IORQ* (OUT*)	k	"	"	"		"	"	9		"	"	20			"	20	
CASEN*	b	74	00	10		36	04	2	#2	43'	04	2	#2		32	1	#7
RFSH*	o	40	Z80	28		16	367	3		33	Z80	28			Z80	28	
A7'	p	51	157	14		24	157	14	#3	30	157	5	#4		157	13	#8
A15'	q	35	"	6		23	"	6	#5	"	"	6	#5		157	14	#8
A16	r	51'	"	2	#6	24'	"	2	#6	34'	"	2	#6		"	3	#6
A17	s	"	"	3	#6	"	"	3	#6	"	"	3	#6		"	2	#6
CASEN1*	w	--------------				--------------				54	32	12			--------------		
CASEN2*	v	--------------				--------------				2	2	1			--------------		
CASEN3*	u	--------------				--------------				-----------------					--------------		
CASEN4*	t	--------------				--------------				-----------------					--------------		
+5V	j	52	157	16		23	157	16		30	157	16			Z80	11	
GND	i	"	"	8		"	"	8		"	2	8			"	29	

#1 Anstelle des reinen RESET*-Signals ist eine AND-Verknüpfung von RESET* und NMI * (Z80, Pin17) zweckmäßiger, um bei jedem Booten Bank 0 einzustellen.

#2 Anstelle dieses invertierten 0-16K*-Signals ist es sicherer, dieses Signal über ein OR-Gatter 74LS32 mit MERQ* (GENIE: IC16, 74LS367, Pin 5/ Komtek 1: IC33, Z80, Pin 19) zu verknüpfen.

#3 Das beim Umrüsten auf 4164-RAMs an diesen Pin gelegte Signal ist an Pin 13 von Z24, 74LS157 zu legen. Die dorthin führende Leiterbahn ist zu durchtrennen.

#4 Das beim Umrüsten auf 4164-RAMs an diesen Pin gelegte Signal ist an Pin 3 von IC 34, 74LS157 zu legen. Die dorthin führende Leiterbahn ist zu durchtrennen.

#5 Die an diesen Pin führende Leiterbahn ist zu durchtrennen.

#6 Dieses IC 74LS157 wird mit den Pins 1,8, 15 und 16 huckepack auf einen anderen 74LS157 Multiplexer gelötet. Die Numerierung bezieht sich auf das IC darunter.

#7 74LS32 auf Höhe des EPROMs

#8 74LS157 beim µ1-Kondensator

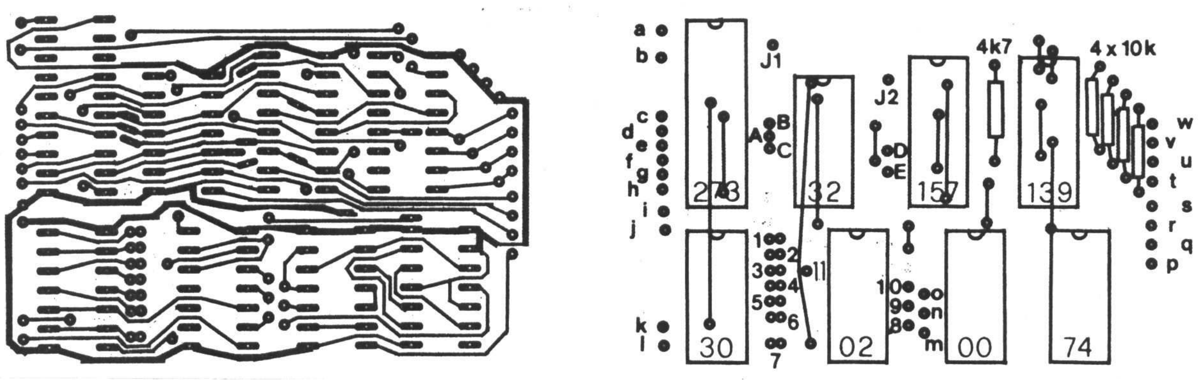


Abb 3: Layout (bestückungsseitig gesehen), Bestückung, Drahtbrücken, Jumper und externe Anschlüsse

---

1987-BHV-2 S.16 / 1987-BHV-3 S.7

Richtigstellung: 256K-Banker

Daß man beim Eintippen mal die Shift-Taste nicht erwischt, ist durchaus nichts besonderes; daß man beim Kerrekturlesen aber nicht merkt, daß da anstelle der beabsichtigen Gänsefüßchen eine '2' steht, ist schon ein mittelprächtiges Verbrechen. Und genau das muß ich hier beichten.

In meiner Einbauanleitung für den 256K-Banker ist mir solches in der Tabelle zur Verdrahtung zwischen Banker und CPU-Board passiert. Beim GENIE I und II liegt IORQ* natürlich nicht an Pin9 von IC2 sondern an Pin9 von IC16, für für das dann auch die Bezeichnung 74LS367 zutrifft.

Für den zweiten Fehler fällt mir leider keine so elegante Ausrede ein. Beim TRS80 sucht man am IC Z55 wergeblich nach einem Pin 17, um sich da das Signal A0 zu holen. Ein 74LS367 hat nur 16 Beinchen und A0 liegt dort an Pin 11.

Erstaunlich, daß noch niemand protestiert hat, weil seine Aufrüstung auf 256K nicht funktioniert. Wahrscheinlich habt Ihr alle diesen Fehler entdeckt und selbständig beim Einbau korrigiert. Nur schade, daß mich niemand darauf hingewiesen hat.

Helmut Bernhardt

Parallel-I/O-Adapter mit dem PPI8255

Helmut Bernhardt

TRS-80, GENIE und Komtek haben weder für IEC-Bus, RS232C noch sonst irgendwelche standartisierten Schnittstellen Anschlüsse vorbereitet. Das einzige (außer Stecker für Drucker und Floppy), was man vorfindet, ist der herausgeführte Systembus. Leider gibt es keinerlei Interfaces, die auf den Systembus aufgesteckt werden und dann Standard-Schnittstellen bereitstellen. Es ist auch nicht zu erwarten, daß irgendein Hersteller für unsere wenig bedeutenden Geräte sowas je anbieten wird. Wir müssen uns da schon selber helfen.

Andererseits ist der Systembus für Bastler viel inreressanter als ein IEC-Bus oder User-Port. Es stehen dabei alle Möglichkeiten offen, irgendwelche Schnittstellen selbst zu bauen und über den Systembusstecker anzuschließen. Man ist nicht auf die vom Hersteller vorgegebenen Möglichkeiten beschränkt.

Eine einfache Parallel-I/O-Schnittstelle mit 24 Leitungen läßt sich mit dem 8255 aufbauen. Neben dem 8255 werden noch 2 weitere ICs zur Portdecodierung benötigt. Die in Abb.1 vorgeschlagene Schaltung benutzt die Portadressen 98H-9BH. Sie ist so einfach, daß sich die Herstellung eines Layouts nicht lohnt. Die Schaltung läßt sich auf einem 5 cm * 8 cm Stück Lochrasterplatine aufbauen.

Der Anschluß an den Systembus des Computers ist in Tabelle 1 wiedergegeben. Für den TRS-80 ergeben sich dabei gewisse Schwierigkeiten. Die Versorgungsspannung von +5V liegt nicht auf dem Systembus-Stecker. Sie muß an geeigneter Stelle auf dem CPU-Board entnommen werden. Außerdem ist das Signal IORQ* nicht vorhanden. Anstelle von IORQ* muss dann auch an Pin 11 des 74LS02 das Signal A2 gelegt werden. Anstelle der nicht verfügbaren Signale RD* und WR* werden die Signale IN* und OUT* an die Pins 5 und 36 des 8255 gelegt.

Der 8255 hat gegenüber dem für den Z80 üblichen PIO-Baustein den Nachteil, daß er keinen IM2-Interrupt liefern kann und daß die Leitungen eines Ports nicht unabhängig voneinander als Input oder Output definiert werden können. Dafür stehen aber in der einfachsten Betriebsart (Mode 0) drei 8-Bit-Ports zur Verfügung. Beim Port C lassen sich zusätzlich noch oberes und unteres Nibble unabhängig voneinander als Input oder Output festlegen.

Ein echtes Handshaking ist mit einzelnen Leitungen des Port C für die Ports A und B ebenfalls möglich, wenn die Betriebsarten Mode 1 oder Mode 2 programmiert werden. Dann muß noch ein zusätzliches IC 74LS05 eingesetzt weden, das die Interrupt-Ausgänge des 8255, PC0 und PC3 invertiert und auf Open-Collector Ausgänge bringt. Außerdem kann darüber auch ein externes, low aktives Strobe-Signal an den INT-Eingangder CPU geführt werden, wenn das Handshaking nicht durch den 8255 erledigt werden soll.

Tabelle 1: Anschuß an den Systembus

	Pin des Systembus-Steckers		
Signal	**GENIE**	**Komtek**	**TRS-80**
D0	15	46	30
D1	14	41	22
D2	12	40	32
D3	16	39	26
D4	20	42	18
D5	11	37	28
D6	18	44	24
D7	17	43	20
A0	10	34	25
A1	7	32	27

Tabelle 2: Pinbelegung des 40poligen I/O-Steckers

Signal	**Pin**	**Pin**	**Signal**
PA4	39	40	PA3
PA5	37	38	PA2
PA6	35	36	PA1
PA7	33	34	PA0
GND	31	32	GND
-	29	30	-
-	27	28	-
-	25	26	-
-	23	24	-
-	21	22	ext.STROBE*
-	19	20	-

A2	9	30	40
A3	8	28	34
A4	6	26	31
A5	5	24	35
A6	4	22	38
A7	3	20	36
IORQ*	38	35	40 #2 (A2)
RD*	41	38	19 #2 (IN*)
WR*	40	36	12 #2 (OUT*)
RESET*	46 #1	33 #1	2 (SYSRES*)
INT*	31	10	21
+5V	19	17,18	39 #3
GND	1,2,	1,2,	8,29,37
	49,50	49,50	

PC7	17	18	PC0
PC6	15	16	PC1
PC5	13	14	PC2
PC4	11	12	PC3
+5V	9	10	+5V
PB7	7	8	-
PB6	5	6	PB1
PB5	3	4	PB2
PB4	1	2	PB3

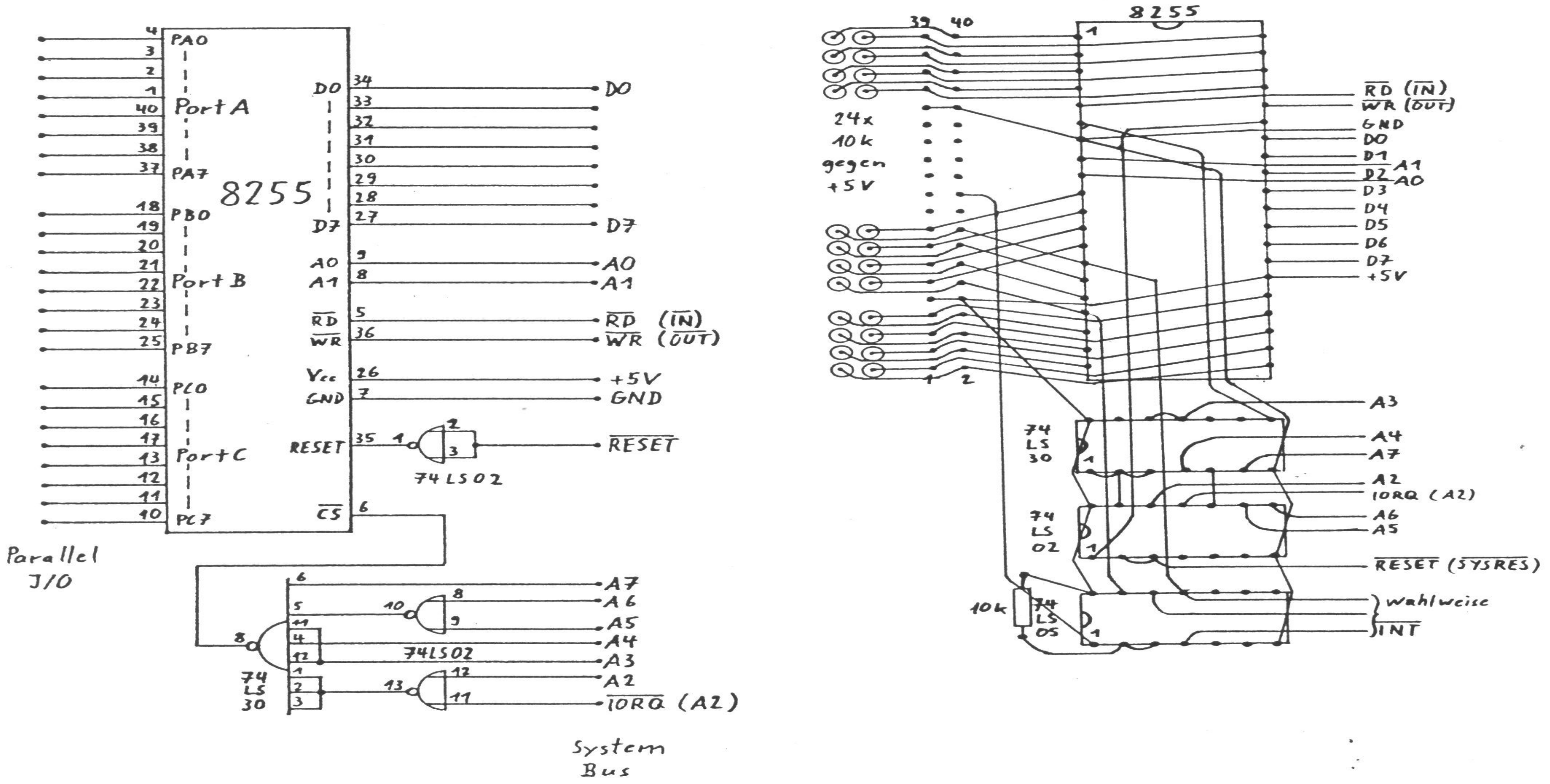
Anmerkungen zu Tabelle 1

#1 anstelle von RESET* sollte ein durch AND-Verknüpfung aus RESET* und NMI* gebildetes SYSRES* benutzt werden

#2 anstelle der nicht vorhandenen Signale IORQ*, RD* und WR* müssen A2, IN* und OUT* benutzt werden

#3 Die Versorgungsspannung ist nur am Expansion-Interface und nicht am Grundgerät zu entnehmen; bei Anschluß an das Grundgerät kann sie an geeigneter Stelle auf dem CPU-Board abgenommen werden oder auch vom angeschlossenen Peripheriegerät geliefert werden, zu dem auch Masse-Verbindung bestehen muß

Abb.1: Schaltung und Verdrahtung des I/O-Boards



1M-Byte RAM im GENIE III

Helmut Berhardt

Einbauanleitung für das 256K-(1M-) Banking-Board
(siehe: 256K RAM für Z80-Systeme)

Wegen nicht vorhandener Hardware-Dokumentation werden hier für den GENIE III alle wichtigen Informationen zum Umrüsten auf 256K, 512K oder 1M RAM zusanmengestellt.

Das Banking-Board wird dabei in geeigneter Weise unter dem CPU-Board angebracht und die Verbindungen zwischen den Boards werden mit dünnen isolierrten Drahtlitzen hergestellt. Bei den meisten Leitungen können die Enden der Litzen direkt auf der Lötseite an die entsprechenden Pins der ICs gelötet werden.

Verbindungen zu Pins von Huckepack-ICs müssen durch Bohrlöcher, die an geeigneter Stelle im CPU-Board angelegt werden, geführt werden. Wenn man das Board gegen das Licht hält, sieht man freie Stellen, wo auf beiden Seiten keine Leiterbahnen verlaufen.

Es sollte darauf geachtet werden, daß die Drahtverbindungen mölichst kurz sind.

Im einzelnen sind folgde Maßnahmen durchzuführen:

1. Sämtliche 4116-RAMs werden durch 41256-RAMs ersetzt
2. Auf der Bestückungsseite wird die breite unter Z8 (74LS00) hindurchführende Leiterbahn (+12V) durchtrennt.
3. Auf der Lötseite wird die außen an den ROM-Sockeln vorbeiführende und mit "-5V" beschriftete Leiterbahn durchtrennt.
4. Auf der Lötseite wird die Verbindung zwischen Pin16 von 212 (74LS161) zu Pin9 des RAMs daneben durchtrennt.
5. Auf der Lötseite wird die breite Leiterbahn (neben der mit "-5V" beschrifteten Leiterbahn) zwischen dem Widerstandsarray und dem Kondensator, neben dem auf der Bestückungsseite "D0" steht, durchtrennt.
6. Zwischen Pin8 des obersten linken (bestückungsseitig gesehen) RAMs und dem äußersten linken Pin des Widerstandsarays wird mit einem nicht zu dünnen isolierten Draht eine Verbindung hergestellt.
7. Im Bereich der RAMs werden alle auf dem Bestückungsplan eingekreisten Kondensatoren ausgelötet.
8. Auf der Bestückungsseite werden die 4 nebeneinander verlaufenden Leiterbahnen von den Pins 4-7 von Z17 (74LS139) zu den Pins 2, 5, 10 und 13 von Z21 (74LS32) durchtrennt.
9. Auf der Lötseite wird die zu Pin3 von Z23 (74LS157) führende Leitung (A7) durchtrennt. Dieser Pin wird mit Pin7 von Z20 (74LS244) verbunden.
   Auf der Lötseite wird die von Pin3 von Z23 unter den ICs 26 und 30 hindurch zu Pin1 des Sockels von EP3 führende Leiterbahn durchtrennt. Der Pin1 von EP3 wird mit A7' des Bankers verbunden.
10. Auf Z23 (74LS157) wird ein zusätzliches IC 74LS157 mit d Rins 1, 8, 15 und 16 huckepack-aufgelötet (gleiche Orientierung); alle anderen Pins werden waagerecht abgebogen. An die Pins 2 und 3 des Huckepack-ICs werden A16 und A17 des Banking-Boards geführt. Pin 4 wird über einen 33-Ohm-Widerstand mit den Pins 1 der RAMs (vorher -5V) verbunden. An Pin5 wird A7' und an Pin6 wird A15' des Banking-Boards geführt. Pin7 wird über einen 33-Ohm-Widerstand mit den Pins 9 der RAMs (vorher +5V) verbunden.
11. Das Signal CASEN* für das Banking-Board wird von Pin1 von Z17 (74LS139) abgenommen.
12. RESET* kann an Pin26 der Z80-CPU abgenommen werden. Wegen der Nutzung des NMI*-Signals zum Booten des DOS ist es aber zweckmäßig, anstelle des reinen RESET* für das Banking-Board lieber ein SYSES-Signal zu erzeugen. Dafür wird auf Z22 (74LSOO) ein 74LS08 mit den Pins 7 und 14 huckepack- aufgelötet. Alle anderen Pins werden hochgebogen. Pin1 dieses ICs wird mit Pin17 des Z80 und Pin2 mit Pin26 des Z80 verbunden. An Pin3 wird das Signal SYSRES* abgenommen, das anstelle von RESET* an das Banking-Board geführt wird.
13. Die Signale D0-D5, AO-A7 und A15 werden von den im Auszug aus dem Bestückungsplan (Abb.1) bezeichneten Pins des ICs Z25 (74LS245), sowie Z20 und Z16 (74LS244) abgenommen.
14. IORQ* wird von Pin3 von Z19 (74LS367), WR* von Pin13 und RFSH* von Pin5 von Z19 (74LS367) abgenommen.
15. +5V und GND können von beliebigen ICs in der Nähe der Stelle, wo das Banking- Board montiert wird, abgenommen werden.

+5V:	Z16, 20, 25 und 30	jeweils Pin 20
	217, 19, 23 und 26	jeweils Pin 16
	Z13, 14, 18, 21, 22 und 24	jeweils Pin 14
GND:	Z16, 20, 25 und 30	jeweils Pin 10
	Z17, 19, 23 und 26	jeweils Pin 8

Z13, 14, 18, 21, 22 und 24 jeweils Pin 7

16. Die Signale CASEN1* bis CASEN4* werden (in dieser Reihenfolge) an die Pins 2, 10, 5 und 13 von Z21 (74LS32) gelegt. Wenn nur 256K realisiert werden sollen, brauchen diese Siale nicht benutzt zu werden. Dann wird nur die linke Reihe RAM-Sockel mit 41256-RAMs bestückt und eine Verbindung zwischen Pin2 von Z21 (74LS32) und Pin1 von 217 (74LS139) hergestellt.
    Wenn 512K gewünscht sind, werden die beiden linken RAM-Reihen bestückt und CASEN1* an Pin2 und CASEN2* an Pin10 von Z21 (74LS32) geführt.
17. Wenn das Banking in den oberen 32K des Z80-Adreßraums erfolgen soll, muß ein invertiertes A15 an das Banking-Board geführt werden. Dafür kann ein freies NAND -Gatter von Z8 (74LSOO) verwendet werden. A15 (von Pin9 von Z20, 74LS244) wird an die Pins 12 und 13 von Z8 gelegt und an Pin11 von Z8 kann das invertierte A15 abgenommen werden.

Um das Banking sowohl unter GDOS als auch unter CP/M (und wenn es das mal geben sollte: CPM/M+) nutzen zu können, ist es zweckmäßig, zwischen A15 selbst und dem invertierten A15 mit einem Umschalter wählen zu können.

Da im GENIE III die Ports EOH bis EFH belegt sind, muß für das Banking-Board eine andere Adresse gewählt werden. Geeignet ist die Portadresse DF; dafür sind dann die Adreßleitungen an folgende Punkte anzuschließen:

A7 an 7 und n	A5 an 8	A2 an 3
A6 an 6 und m	A4 an 5	A1 an 2
	A3 an 4	A0 an 1

Der Punkt 11 wird mit keinem der Punkte 1-7 verbunden. Die Punkte 9 und 10 werden nicht angeschlossen.

Abb.1: Auszug aus dem Bestückungsplan des GIII Motherboards. *Es handelt sich von rechts nach links um die ICs 16, 17, 20, 21, 23, 25, 26, 30 im Bereich zwischen RAMs, EPROMS und CPU.*

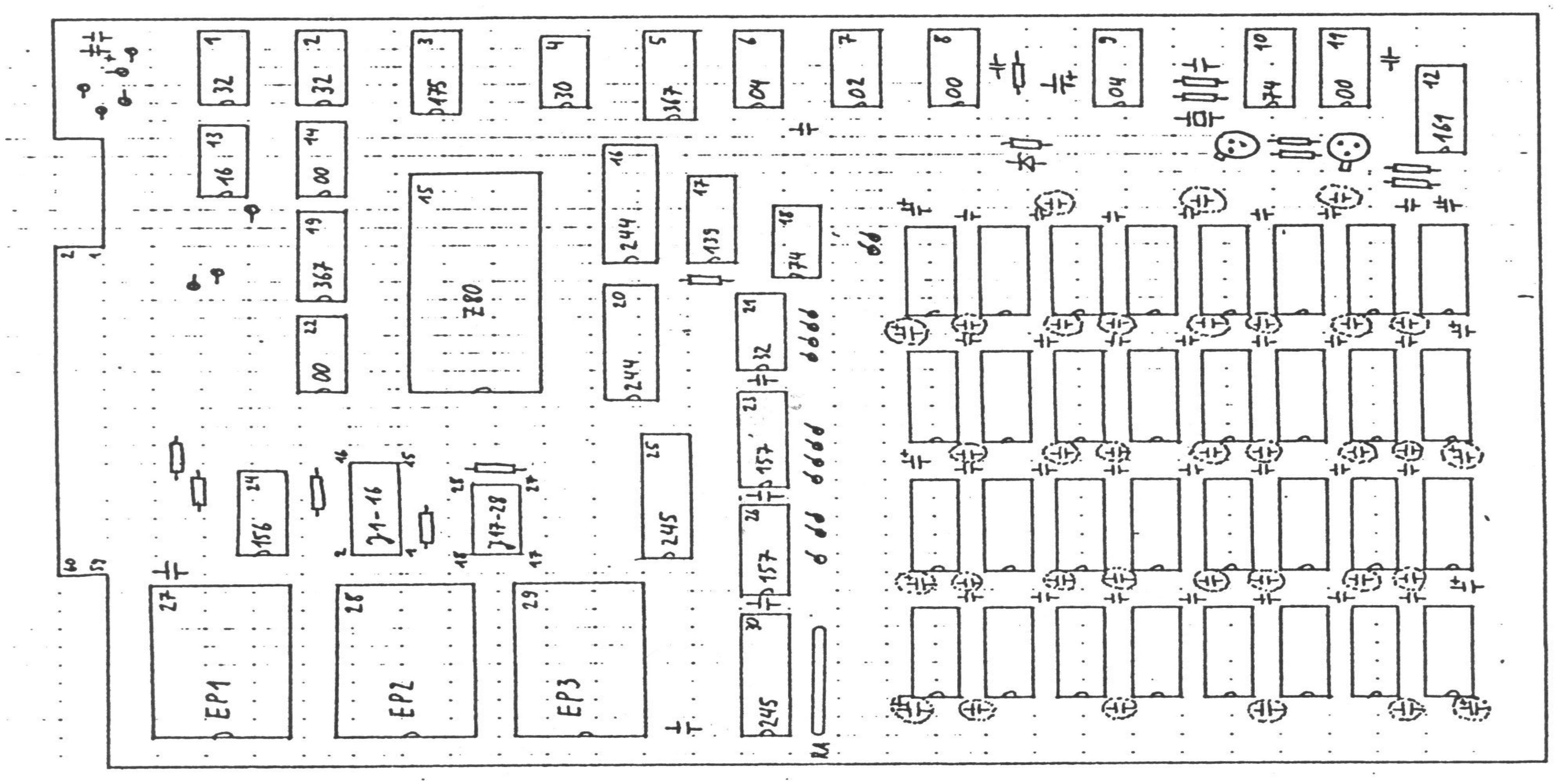


Abb.2: Bestückungsplan des GIII Motherboards

1986-BHV-06 S.24-29

ECB-Bus beim GENIE 1 und 2

Helmut Bernhardt

Soweit allgemein bekannt ist, gibt es für den direkten Anschluß an den Systembus des GENIE lediglich das alte Expansion-Interface EG3014 mit 32K RAM und den Interfaces für Floppy und Drucker oder später dann den Floppy-Controller im Laufwerksgehäuse (der wegen seines langen Kabels kapazitiv unnötig belastet). Außerdem ist noch der Kleinkram wie EG64MBA und eine emtsprechende Centronics- Schnittstelle zu erwähnen. Das war's dann wohl schon.

Wenn man sich nun nicht alle offenstehenden Wünsche mit dem Lötkolben selbst realisieren will, sondern auf das reichhaltige Angebot fertiger Karten für Z80- Computer zurückgreifen möchte, ist es zweckmäßig, sich eine ECB-Bus-Platine zu kaufen und diese über eine Steckverbindung an den Systembusstecker des GENIE anzuschließen. Der ECB-Bus ist der bei Z80-Systemen am meisten verbreitete Busstandard, für den es alle beliebigen Systemerweiterungen zu kaufen gibt.

Dabei ist allerdings abzusehen, daß das Fanout der Treiber der Z80-Signale nicht unendlich ist und die Anzahl der auf dem ECB-Bus angeschlossenen Karten begrenzt ist (besonders dann, wenn die Floppystation über ein unverschämt langes Buskabel angeschlossen ist). Deshalb soll demnächst auch noch eine Steuerkarte für einen gepufferten ECB-Bus vorgestellt werden, die trotz Pufferung auch noch IM2- und DMA-Fähigkeit aufrechterhält.

Bei der Untersuchung der Pinbelegung des ECB-Bus <1> erkennt man die meisten Signale des GENIE-Systembus wieder. Solche Signale können einfach vom GENIE-Bus abgegriffen werden und auf den entsprechenden Pin des ECB-Bus gelegt werden (D0-D7, A0-A15, RD*. WR*, MERQ*, IORQ*, RESET*, NMI*, INT*, RFSH*, HALT*, M1*, WAIT*, BUSREQ*, BUSAK*, CLOCK, +5V, GND).

Weitere Signale spielen für den GENIE keine Rolle. Sie bleiben am ECB-Bus unberücksichtigt (PWRCL*, VCMOS, ext.RFSH*, +15V, und bei einigen Varianten auch noch -15V, A16-A19).

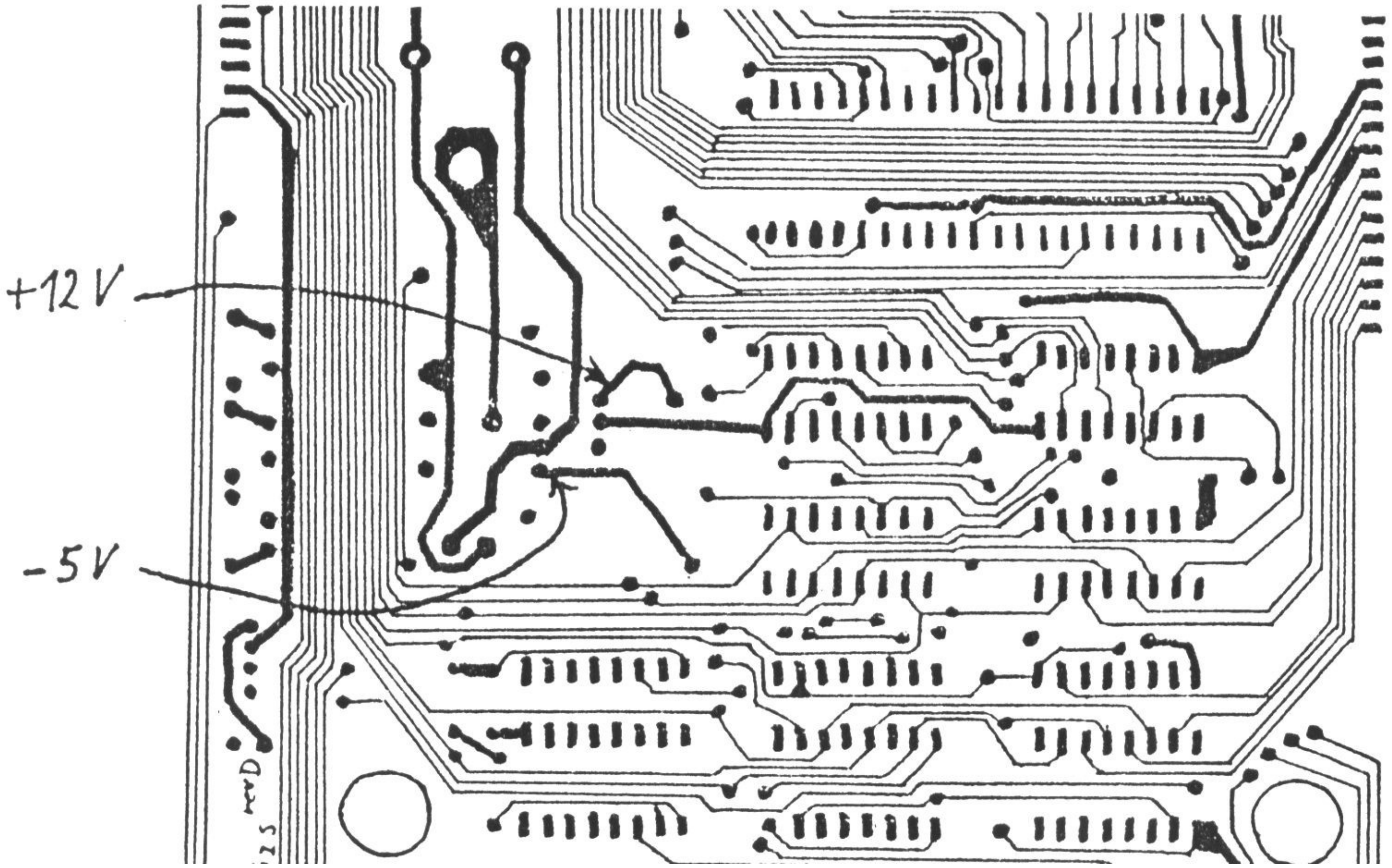


Abb.1: Abgreifen der Versorgungs-Spannungen +12V und -5V auf dem CPU-Board

Die Signale IN* und OUT* liegen nicht auf dem GENIE-Bus, sie müssen an den Pins 32 und 31 des Verbindungssteckers zum Video-Inteface abgegriffen werden. Auch die Versorgungsspannungen +12V und -5V müssen bei Bedarf auf dem CPU- Board an den in Abb.1 bezeichneten Stellen entnommen und gesondert an den ECB-Bus geführt werden.

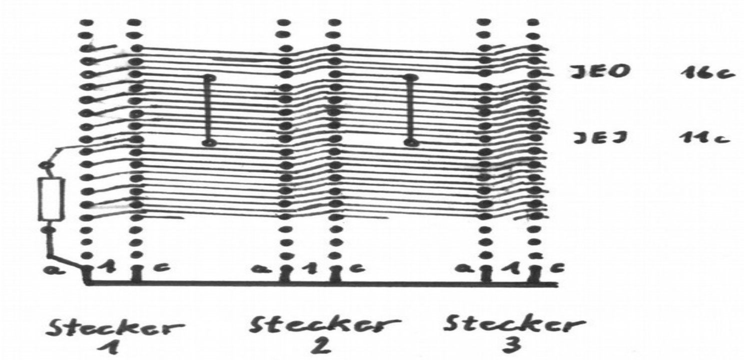


Abb.2: IEI-IEO-Daisy-Chain auf dem ECB-Bus

Es bleiben auf dem ECB-Bus noch die Signale IEI und IEO übrig. Diese bilden die Interrupt-Daisy-Chain zwischen den einzelnen ECB-Bus-Karten. Auf der ECB- Busplatine sind diese Signale nicht von Stecker zu Stecker direkt miteinander verbunden. Es ist immer der Pin IEI (Interrupt Enable In) einer Steckkarte mit dem Pin IEO (Interrupt Enable Out) des Steckers davor verbunden, während der Pin IEO jeden Steckers mit dem Pin IEI des Steckers dahinter verbunden ist. Beim ersten Stecker ist der Pin IEI über einen Widerstand von einigen KOhm an +5V gelegt. Durch diese Busstruktur ist den einzelnen Steckplätzen (vom ersten zum letzten abfallend) ihre Interrupt Priorität zugeordnet.

Eine Karte kann der CPU nur dann einen Interrupt liefern, wenn der Eingang IEI der Karte high ist. Die Z80 Peripherie Bausteine verfügen über entsprechende Pins IEI und IEO. Ein Baustein, der gerade einen Interrupt gemeldet hat, gibt über IEO Low-Pegel aus und unterdrückt dadurch am Eingang IEI des nächsten angeschlossenen Bausteins einen eventuellen Interrupt. Ein Baustein, dessen IEI-Eingang auf Low geht, gibt diesen Pegel dann über seinen IEO-Ausgang an den nächsten Baustein weiter, so daß bei allen nachgeschalteten Bausteinen dadurch das Ausgeben eines Interrupts verhindert wird. Erst wenn der den Interrupt auslösende Baustein auf dem Datenbus die beiden Bytes des Befehls RETI nacheinander erkennt, nimmt er seinen IEO-Ausgang wieder auf High zurück und ermöglicht dadurch niedriger priorisierten Bausteinen, Interrupts an die CPU abzusetzen (vorausgesetzt, sein IEI-Eingang ist nicht durch einen Interrupt eines höher priorisierten Bausteins inzwischen auf Low gesetzt worden).

Während des Abarbeitens einer Interrupt-Service-Routine läßt sich die CPU nicht durch den Interrupt eines höher priorisierten Bausteins stören. Dieser wird erst angenommen, nachdem RETI ausgeführt wurde. Es kommt also nicht zur Verschachtelung von Interrupt-Routinen und dem dadurch verursachten Chaos auf dem Stack.

Auf ECB-Bus-Karten, die selbst keinen Interrupt auslösen können, sind die Pins IEI und IEO des Busteckers miteinander verbunden, so daß die Daisy-Chain nicht unterbrochen wird. Es dürfen auch aus diesem Grund zwischen zwei Karten keine leeren Stecker auf der Busplatine vorhanden sein.

Da die im GENIE üblichen Interrupt-Quellen (Timer, Floppy-Controller und eventuell serielle Schnittstelle) diese Interrupt-Struktur nicht unterstützen, müssen die Signale durch Verwendung eines Z80-CTC auf dem ECB-Bus in diese Struktur eingebunden werden, wenn man den äußerst leistungsfähigen Interrupt- Modus 2 nutzen will. Der Z80-CTC stellt ihnen dann auch einen Vektor für den IM2 zur Verfügung <2;3>.

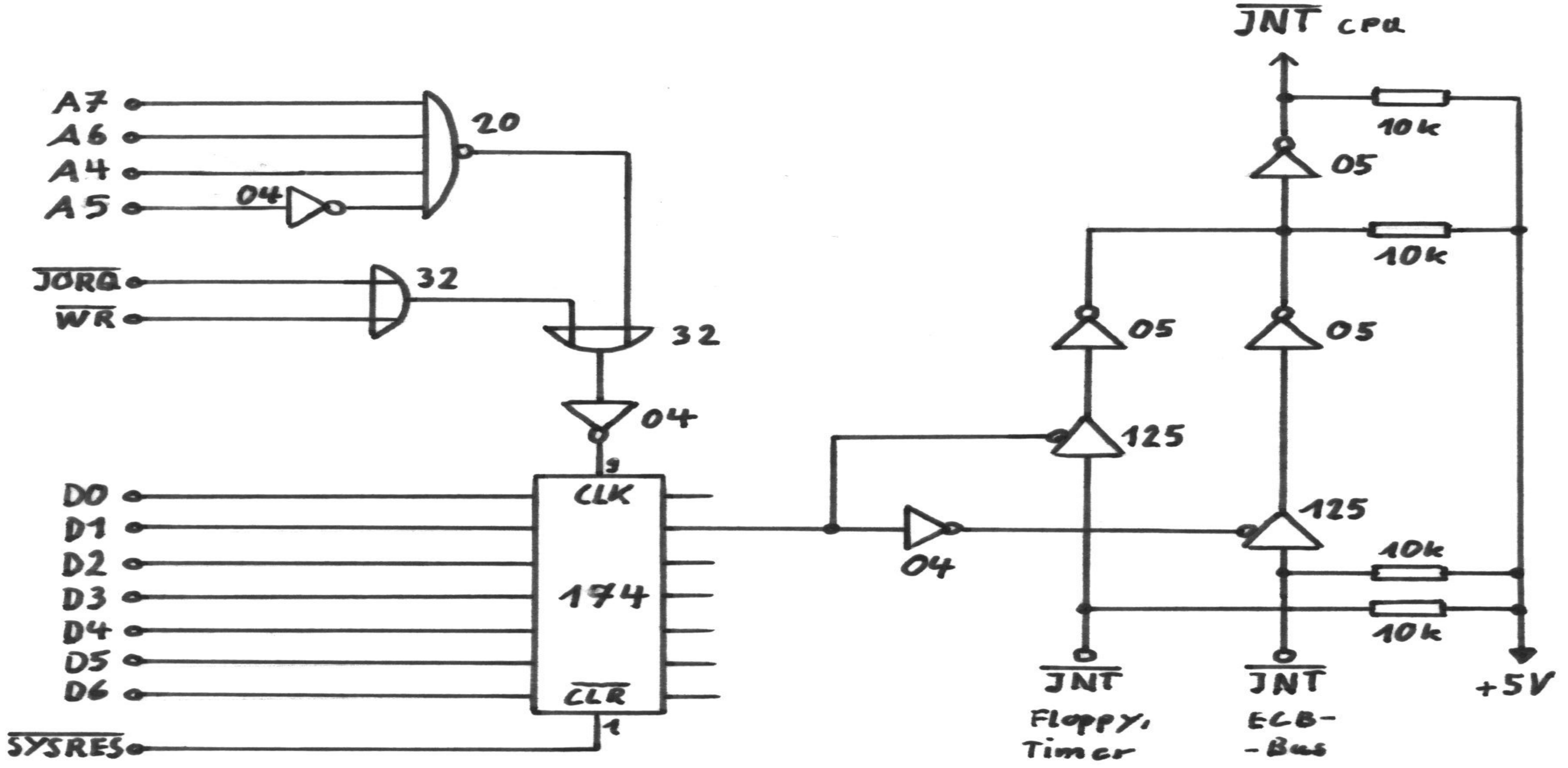


Abb.3: Programmierbare Umleitung des INT-Signals von FDC und Uhr über einen Z80-CTC-Kanal*

Abgesehen vom eventuellen Interrupt einer seriellen Schnittstelle werden die Interrupt-Signale von Timer und FDC über eine gemeinsame Leitung an den INT-Pin der CPU geführt. Durch etwas Hardware (Abb.3) läßt sich über Software dieses Interrupt-Signal entweder direkt an die CPU führen oder über einen Z80-CTC Zählereingang umleiten, wobei dann der IM2 unterstützt wird. Da auch im Betriebssystem beide Interrupts von der gleichen Service-Routine bedient werden, reicht im IM2 auch ein Vektor eines CTC-Kanals aus. In der Interrupt- Tabelle darf aber nicht das Sprungziel 4012H stehen (Interrupt Vektor des IM1 des DOS), da die Interrupt-Service-Routine des DOS nicht mit RETI sondern mit RET endet und der Z80 dann nicht weiß, wann er seinen IEO-Ausgang wieder auf High zurücknehmen soll. Es muß folgender Umweg eingeschlagen werden:

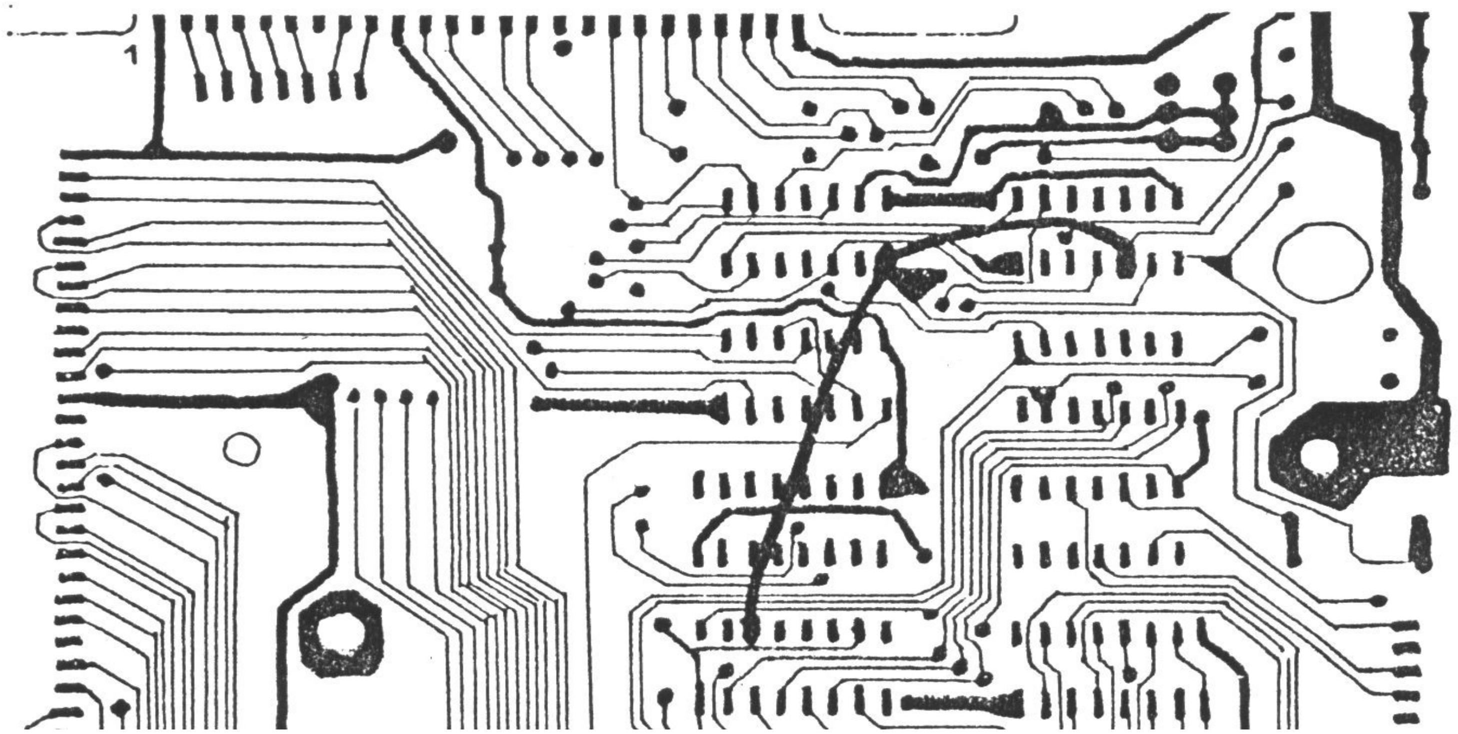
```
INTTAB  DEFW    SERV1   ;ANFANG INTERRUPT TABELLE DÜR IM2
        DEFW    SERV2   ;VEKTOR FÜR DEN NÄCHSTEN INT-LIEFERANTEN
        DEFW    ....    ;U.S.W.
        ....
        DEFW    CTC     ;* SIEHE FUSSNOTE
```

```
                ....
        CTC     CALL    4012H   ;TIMER-, FLOPPY-INT.-ROUTINE
                EI
                RETI
                * Das Lable CTC ist der Anfang der Interrupt-
Service-Routine
                für Timer und Floppy. Seine Adresse ist in der Interrupt-
                Tabelle an der Adresse eingetragen, deren
höherwertiges
                Byte in das I-Register der CPU eingetragen worden ist und
                deren niederwertiges Byte dem entsprechende CTC-Kanal als
                Interrupt-Vektor einprogrammiert worden ist.
```

Schließlich bleiben noch die Signale CCDBS/STADBS* und DODBS/ADDBS* des GENIE-Bus übrig. Diese sind für den Betrieb eines externen DMA-Controllers vorgesehen.

Ein DMA-Controller <4;5> kann, wenn er von der CPU dazu aufgefordert wird, die Kontrolle über den Systembus übernehmen und selbst auf Speicher und Ports zugreifen. Dabei liefert er dann die Bussignale (außer natürlich beim Lesen die Signale D0-D7 selbst. Die Synchronisation mit der CPU erfolgt dadurch, daß der DMAC Low-Pegel auf die Leitung BUSREQ* legt, woraufhin die CPU dann als Quittung Low über den Pin BUSAK* ausgibt und ihre Daten-, Adreß- und Steuerpins in den hochmigen Zustand versetzt.

Abb.4: Anschließen des Signals BUSAK* an die Steuereingänge CCDBS/STADBS* und DODBS/ADDBS* auf dem CPU-Board, Ansicht des CPU-Boards von der Lötseite



Es müssen auch die Treiber dieser CPU-Signale in den hochohmigen Zustand versetzt werden, damit deren Ausgänge nicht gegen die Signale des DMAC arbeiten. Die Freigabeeingänge dieser Treiber lassen sich über die Pins CCDBS/STADBS* und DODBS/ADDBS* steuern. Wenn diese Pins auf low gezogen werden, sperren die Treiber der CPU ihre Ausgänge so, daß dann der DMAC den Bus kontrollieren kann. Es muß also das Signal BUSAK* an diese Pins des GENIE-Bus gelegt werden (Abb.4). Diese Änderung kann auch bestehen bleiben, wenn kein ECB-Bus angeschlossen ist oder wenn kein DMA-Betrieb vorgesehen ist und auch sonst kein Gebrauch dieser Steuereingänge des GENIE-Bus gemacht wird, denn BUSAK* ist normalerweise immer high.

Unberücksichtigt bleibt das Signal PHANTOM*. Wenn dieser Pin des GENIE-Bus auf Low gezogen wird, so werden sämtliche Systemkomponenten im Adreßbereich 0 bis 3FFFH (ROM, Floppy, Drucker, Tastatur und Bildschirm) nicht mehr freigegeben und stattdessen RAM in diesem Bereich zur Verfügung gestellt. Eine gezielte Steuerung des PHANTOM*-Eingangs zum Banking leistet das EG64MBA und ein jüngerer Verwanter davon <6>.

Der Anschluß eines ECB-Bus an den GENIE-Systembusstecker ist am einfachsten, wenn dieser Stecker unbenutzt ist (z.B. Floppy-, Drucker-Interface von RB- Elektronic im Grundgetät). Bei angeschlossenem herkömmlichen Expansion-Interface EG3014 oder neuerer Floppy-Station mit internem Floppy-Controller sowie bei aufgestecktem Drucker-Interface oder EG64MBA treten Schwierigkeiten auf. Am güstigsten ist dabei eine Busweiche, d.h. ein kurzes Stück 50poliges Flachbandkabel mit 3 (oder mehreren) Quetschverbindungen gemäß Abb.5. Daran können dann ECB-Bus und alle anderen Systemkomponenten für den GENIE-Bus gleichzeitig betrieben werden. Um an diese Weiche wiederum die weiblichen Stecker der Peripherie-Geräte anschließen zu können, braucht man männliche Zwischenstücke (Platinenstücke mit beidseitig im 1/10 Zoll-Raster geätzten Streifen.

Es sei noch angemerkt, daß bei dem aus dem EG64MBA herausführenden Systembus das Signal MERQ* nicht das gleiche ist wie bei dem vom CPU-Board kommenden MERQ*. Dieses Signal ist schon mit dem Latch-Ausgang des MBA, der das Banking des Bereichs 3600H-37FFH steuert, verknüpft. Wenn dieser Bereich durch den entsprechenden MBA-Befehl als RAM eingestellt ist, bleibt der MERQ*-Ausgang des MBA ständig high, damit dann nicht der Floppy-Controller im Laufwerksgehäuse angesprochen wird.

Bei dem aufwärtskompatiblen Selbstbau-Banker aus <6> kann man sich das herausgeführte MERQ* entsprechend seiner Systemkonfiguration wählen. Wenn der Floppy-Controller nicht im Laufwerksgehäuse untergebracht ist, sollte dort das richtige MERQ* herausgeführt werden. Auch die Signale A14 und A15 sollten dann die des Systembus sein.

Das Prinzip des Anschlusses des ECB-Bus läßt sich auch auf TRS80 (nur Modell 1) und Komtek 1 übertragen. Beim Komtek müssen bei Einsatz eines CPU-Treiber-Boards <7> die Freigabe-Pins der Treiber (Pins 1 und 19 bei den ICs 74LS244 und nur Pin 19 beim 74LS245) mit dem über ein freies AND-Gatter auf dem 74LS00 dieses Boards invertierten BUSAK* der CPU verbunden werden und dürfen nicht wie bisher an GND liegen. Diese Änderung ist aber nicht nötig, wenn kein DMA-Betrieb vorgesehen ist. Der IM2-Betrieb ist bei dem Treiber-Board schon

berücksichtigt und ohne Änderung möglich.

Beim TRS80 sind beide Möglichkeiten nicht gegeben. Auf DMA-Betrieb kann man verzichten. Die Änderung, die nötig ist, um das CPU-Board IM2-fähig zu machen, ist in <8> beschrieben. Die Signale MRD* und MWR* auf dem TRS80-Bus entsprechen nicht den Signalen RD* uns WR* des ECB-Bus. Dafür sind IN* und OUT* schon auf dem TRS80-Bus vorhanden. Alle nicht vorhandenen Signale müssen direkt auf dem CPU-Board abgegriffen werden.

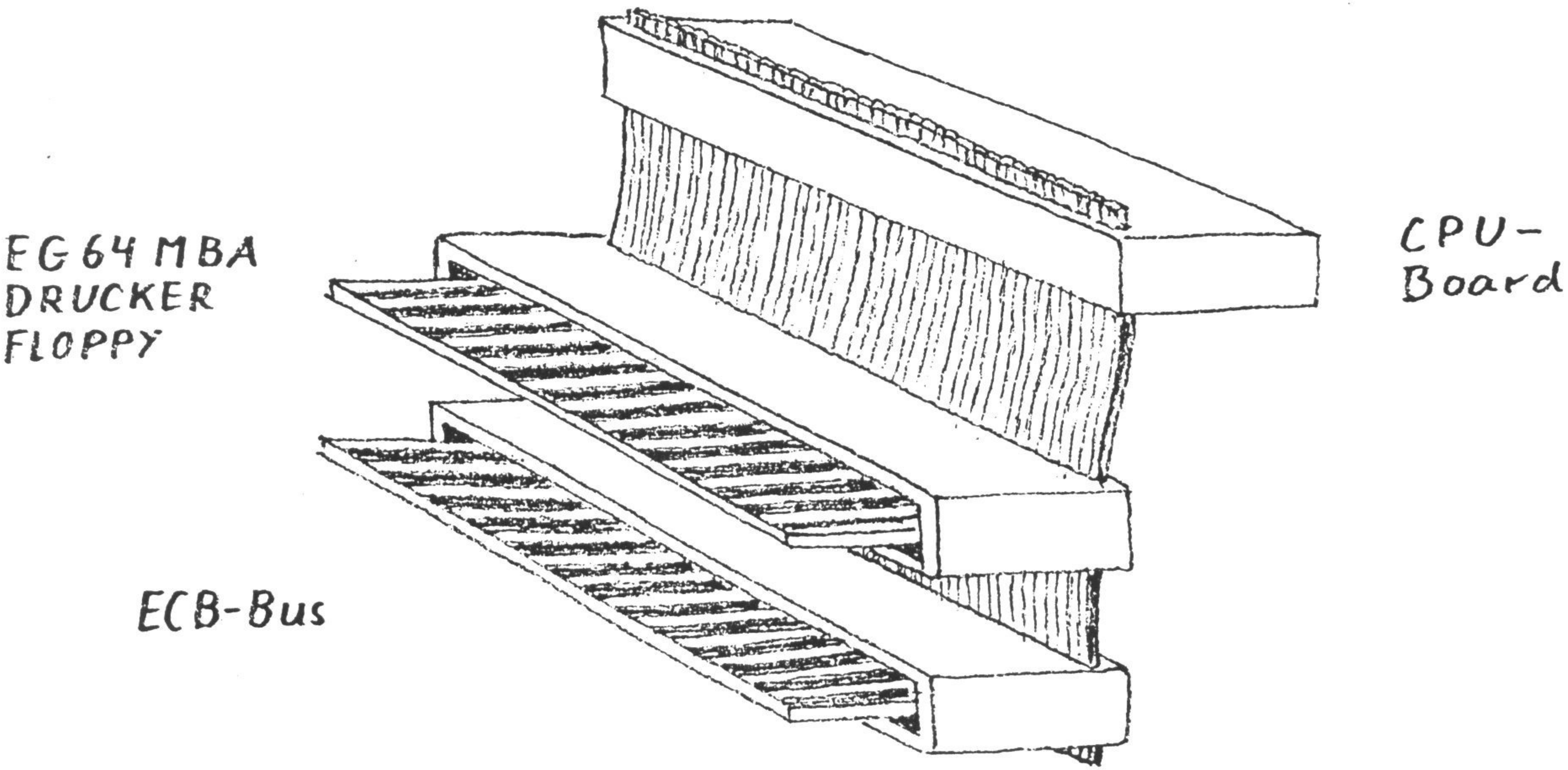


Abb.5: Busweiche zur Vervielfältigung des Systembus-Steckers.

Wie man es auch dreht und wendet, jede Anordnung der Quetschstecker auf dem 50poligen Flachkabel führt dazu, daß die Ober- und Unterseite des GENIE-Bus auf den weitergebenden Steckern gegenüber dem Stecker für das CPU-Board vertauscht sind. Diese Schwierigkeit läßt sich nur dadurch beheben, daß die männlichen Zwischenstücke aus zwei Hälften bestehen, bei denen jeweils gegenüberliegende Streifen der Oberseite der einen Hälfte mit der Unterseite der anderen Hälfte und umgekehrt verbunden sind. Dazu verwendet man am besten kurze isolierte Steifen Schaltdraht. Eine andere Lösung besteht darin, daß der Stecker für das CPU-Board kein Quetschstecker sondern eine Ausführung zum Einlöten in Platinen (Apple-Slot) ist. An die Lötkontakte lassen sich dann die abisolierten Enden des Flachkabels in der richtigen Anordnung löten.

Literatur

<1> Verbindend - Der ECB-Bus / Johannes Assenbaum / c't 12/85, 60-61
<2> Z80 CTC - Einsatz und Programmierung / Rolf Keller / c't 4/85, 92-94
<3> Interrupt-Priorit-Encoder und Vektorgenerator für IM2 mit dem Z80 CTC / H. Bernhardt / Info des GENIE/TRS80-User-Club Bremerhaven 12/85, 10-11
<4> Direkter Speicherzugriff: Schneller als die CPU - So arbeiten DMA-Controller-ICs / Rolf Keller / c't 8/85, 80-83
<5> Z80 DMA - schnell aber kompliziert / Rolf Keller / c't 12/85, 76-79
<6> Hardware-Umschaltung für den Betrieb von CP/M bei den Computern TRS80, GENIE 1 und 2 und Komtek 1 / H. Bernhardt / Info des GENIE/TRS80-User-Club Bremerhaven / Hardware-Sonderheft 11/85, 6-10
<7> Wie aus dem Komtek ein Computer wird / H. Bernhardt / Info des GENIE/TRS80-User-Club Bremerhaven / Hardware-Sonderheft 11/85, 1-5
<8> Interrupt Your 80 - With a hardware-mod, Your 80 is immune to rude interruptions / Dougla C. Fisher / 80 micro January 1983, 258-266

HD64180 als Coprozessor im Z80-System

Helmut Bernhardt

Natürlich kann man mit entsprechenden Änderungen in jedem Z80-Computer den Z80 gegen einen HD64180 austauschen und hat damit die Leistungsfähigkeit dieser CPU zur Verfügung. Man kann aber auch den Z80 beibehalten und zusätzlich einen HD64180 im System betreiben, wobei sich beide CPUs in der Kontrolle des Computers abwechseln können.

Für den Z80 sieht dies so aus, daß er entweder als CPU den Computer steuert oder ganz abgeschaltet ist. Der HD kennt aber zwei verschiedene aktive Zustände. Solange der Z80 Regie führt, muß der HD sich mit 256K RAM und einem EPROM begnügen und hat keinen Zugriff auf den Computer. Wenn der Z80 sich abschaltet und dem HD das System übergibt, kann dieser auf den Computer über die untere Hälfte seines Adreßraumes zugreifen. Sein EPROM ist dann abgeschaltet.

Selbstverständlich muß der HD, wenn er den Computer übernimmt, wissen, was er zu tun hat. Es müssen in der Software beider CPUs Vereinbarungen getroffen werden, die eine Parameterübergabe beim Regierungswechsel ermöglichen. Wenn der HD die Regie an den Z80 übergibt, ist eine solche Parameterübergabe nicht nötig, weil der Z80 seine Arbeit dort wieder aufnimmt, wo er vorher aufgehört hat.

Unter diesen Voraussetzungen ist es möglich, daß der Z80 sich ganz verabschiedet und den HD64180 stattdessen als CPU arbeiten läßt. Andererseits kann der Z80 dem HD ein Programm und einen Satz Daten in den Speicher schreiben, die der HD sich in seine nur ihm zugänglichen oberen 256K RAM kopiert, sich dort hin bankt und die Kontrolle des Computers wieder dem Z80 überläßt. Nun kann er sich in Ruhe mit dem Number-Crunching beschäftigen, bis der Z80 ihn zur Übergabe der Rechenergebnisse auffordert und ihm eventuell neue Aufgaben in den Speicher schreibt.

Eine weitere Beschäftigung des HD kann die Ausgabe von Nachrichten über eine seiner seriellen Schnittstellen sein, ohne daß der währenddessen den Computer steuernde Z80 sich darum kümmern muß, er kann inzwischen Wichtigers verrichten.

Weiterhin kann der HD mit seinen eigenen 256K RAM als intelligente RAM-Floppy mit eigenem Prozessor und eigenen Treiberroutinen mißbraucht werden, wobei dann einer der HD-internen DMA-Controller wirkungsvoll eingesetzt werden kann, um Daten vom/zum Sectorbuffer zu übertragen.

Das Umschalten zwischen den CPUs erfolgt über ein Flip Flop, das nach dem Einschalten des Computers (RESET*) so eingestellt ist, daß der Z80 aktiv ist und die Bustreiber zwischen HD64180 und Computer im hochohmigen Zustand sind. In dieser Kofiguration findet der HD in den unteren 256K seines physikalischen Adreßraumes von 512K sein "Boot-EPROM", das eine Größe von 8 bis 64K-Byte haben kann und innerhalb der 256K gespiegelt ist. In den oberen 256K des HD liegt RAM-Speicher vor, der dem HD immer zur Verfügung steht. Im EPROM muß eine Routine vorhanden sein, die zunächst alles Nötige an Intelligenz ins RAM kopiert und dann den HD in das RAM umbankt. Nachdem der volle logische Adreßraum von 64K in das RAM gebankt wurde, braucht der HD zunächst nur zu warten, bis der Z80 ihm den Computer übergibt. Das kann er im einfachsten Fall durch Untersuchen der Beschreibbarkeit eines Bytes in den unteren 256K oder nach einem geringfügigen Hardware-Patch über einen Interrupt an seinem INT2-Eingang feststellen.

Daraufhin hat er an vereinbarter Stelle im Speicher einen Vektor zu lesen und an die durch den Vektor bezeichnete Stelle zu springen. Das dort stehende Programm entscheidet dann darüber, welche Funktion ihm im folgenden zugedacht ist.

Das Umschalten zwischen den CPUs erfolgt immer durch die gerade aktive CPU mit einem OUT-Befehl an Port FBH, wobei das ausgegebene Datum belanglos ist. Wenn zunächst der Z80 diesen OUT-Befehl ausführt, wird sein BUSRQ*-Eingang low gezogen und die internen Treiber der Z80-Bussignale gehen in den hochohmigen Zustand. Gleichzeitig werden die Bustreiber des HD-Boards eingeschaltet, so daß der HD nun den Systembus des Computers kontolliert. Ein Zugriff auf Speicher und memory mapped I/O des Computers ist aber nur möglich, wenn A18 des HD low ist. Wenn dies nicht der Fall ist, bleibt der Databuffer des Boards hochohmig, so daß der HD dann ohne Störungen seine nur ihm zugänglichen 256K RAM in der oberen Hälfte seines Adreßraumes lesen und beschreiben kann.

Der HD kann nun ebenfalls die Buskontrolle durch einen OUT-Befehl an Port FBH wieder dem Z80 übergeben. Dieser Befehl hat aber nur Wirkung, wenn der HD den Computer kontrolliert. Wenn der Z80 aktiv ist, kann der HD sich durch diesen OUT-Befehl nicht die Kontolle über den Computer erzwingen. Es kann nur der jeweils den Computer kontrollierende Prozessor die Kontrolle an den anderen übergeben.

Der Teil des Coprozessor-Boards, der das Umschalten zwischen den CPUs bewirkt und die Steuerung des BUSRQ*-Eingangs des Z80 und die Freigabe der Bustreiber und des EPROMs des HD-Boards ausführt, ist in Abb.1 gezeigt. Andere wesentliche Schaltungselemente sind in Abb.2 wiedergegeben. Die banalen Dinge wie Beschaltung der RAMs, des EPROMs, der V24-Treiber und der Bustreiber (außer deren Steuereingänge) sind nicht dargestellt.

Die Signale der 50poligen Steckerleiste sind mit den entsprechend bezeichneten Pins des Z80-Sockels bzw. bei einem DMA-fähigen Bus alternativ mit den entsprechenden Pins des Bus zu verbinden. Wenn im Computer durch eine MMU Adressen A16 und A17 erzeugt werden, so kann im einfachsten Fall auf die Verwendung von A16 und A17 des HD verzichtet werden, wobei sich der HD auch der MMU des Computers bedienen muß und seine eigene MMU unwirksam bleibt (außer bei der Steuerung von A18 und der Speicheradressierung in den oberen 256K RAM).

Eine andere Möglichkeit besteht darin, entsprechend der gerade aktiven Betriebsart über einen Multiplexer entweder A16 und A17 der MMU des Computers oder A16 und A17 des HD durchzuschalten. Wenn die MMU des Computers auch ein 8Bit-Refresh für 256K-Bit-RAMs erzeugt, müssen ebenfalls A7 der MMU des Computers und des HD gemultiplext werden. Auch andere durch die MMU des Computers beeinflußte Adressen sind zu multiplexen (A15, A14...?).

Je nach aktiver CPU muß in den Computer entweder der ursprüngliche Takt oder der Takt des HD64180 (Pin 64) eingespeist werden. Es muß also im Computer ein steuerbarer Treiber vorgesehen werden, der den ursprünglichen Systemtakt abschalten kann, wenn der HD das System übernimmt, denn dann wird der HD-Takt über den Bustreiber an den Computer geliefert.

Wenn die RAM-Steuersignale RAS*, MUX* und CAS* des Computers aus dem Systemtakt und aus Z80-Signalen hergeleitet werden, sind hier keine Änderungen nötig. Wenn allerdings dieses Timing aus einem höheren Takt (aus dem eventuell auch der CPU-Takt hergeleitet wird) oder über Gatterlaufzeiten erreicht wird, muß eine andere Schaltung dafür verwendet werden. Eine Möglichkeit ist die auf dem HD-Board verwendete Schaltung (Abb.2), die auch im Computer selbst anstelle der vorhandenen eingesetzt werden kann. Bei Computern mit softumschaltbarer Taktrate sind meistens Schaltungen vorhanden, die so beibehalten werden können. Hier braucht dann nur ein Umschalten zwischen ursprünglichem Systemtakt und Takt des HD realisiert zu werden.

Abb. 1: Bussteuerung der Coprozessor-Karte

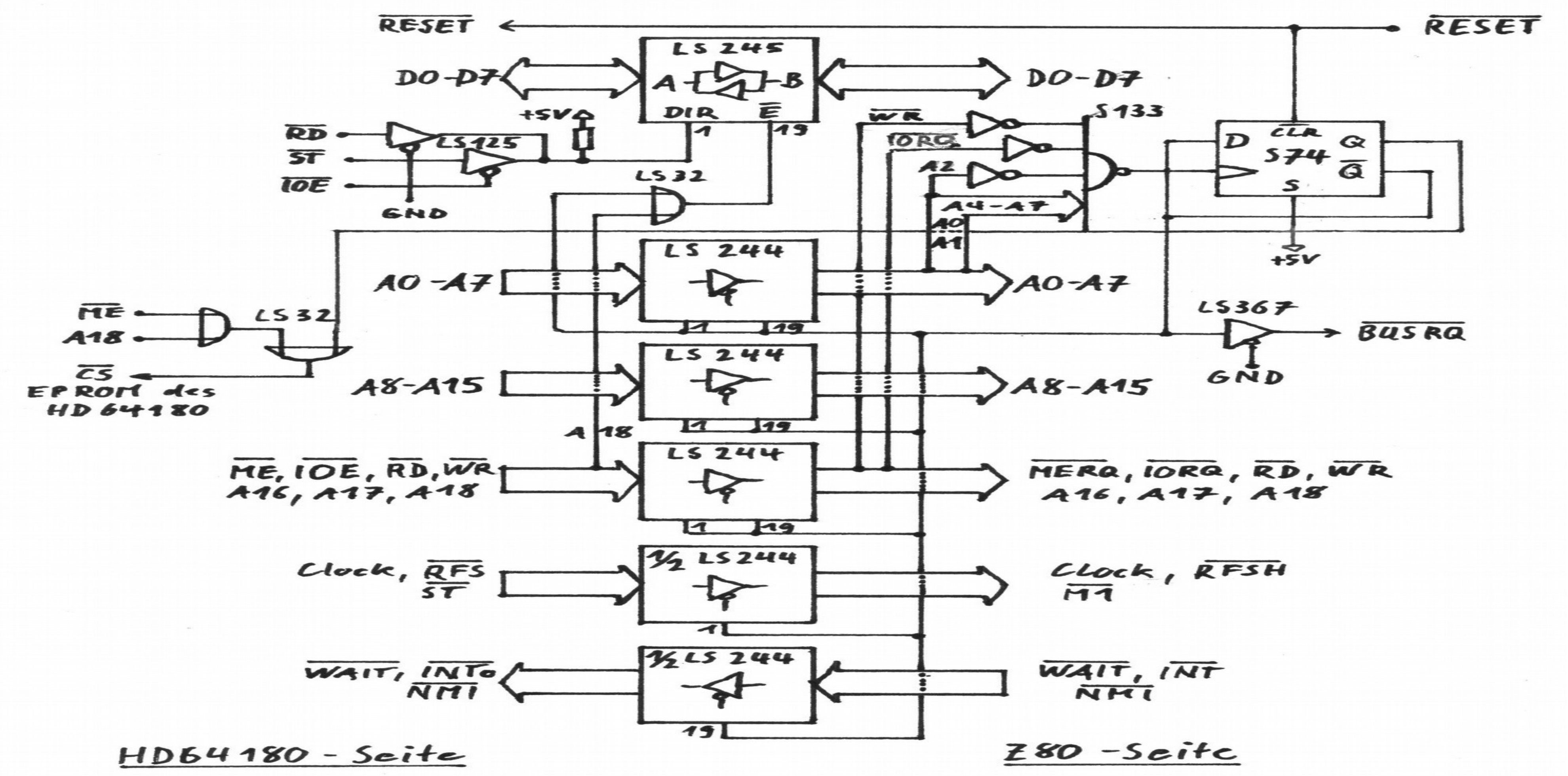


Abb. 2: Steuerung der RAMs der Coprozessor-Karte

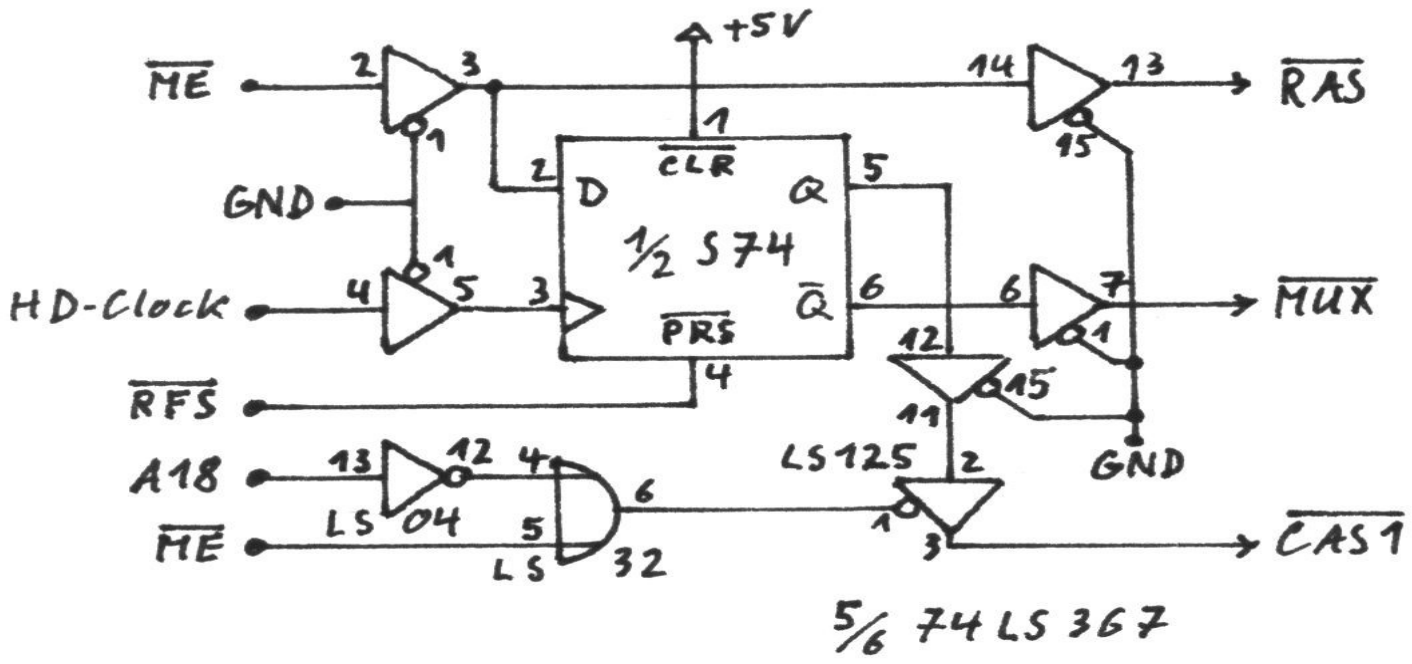


Abb. 3: Layout des HD64180-Boards (Ansicht von der Bestückungsseite)

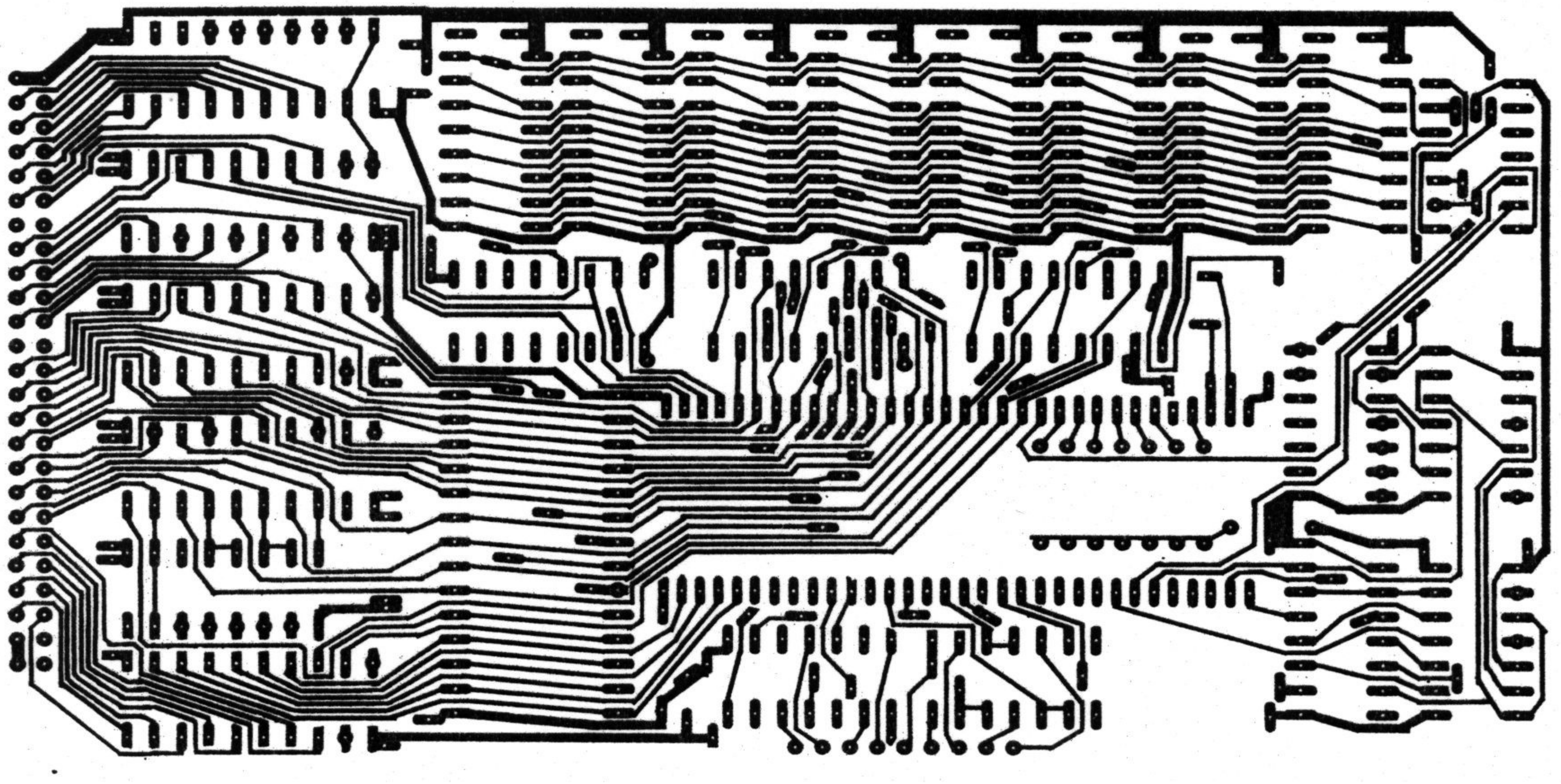


Abb. 4: Drahtbrücken Bestückungsseite

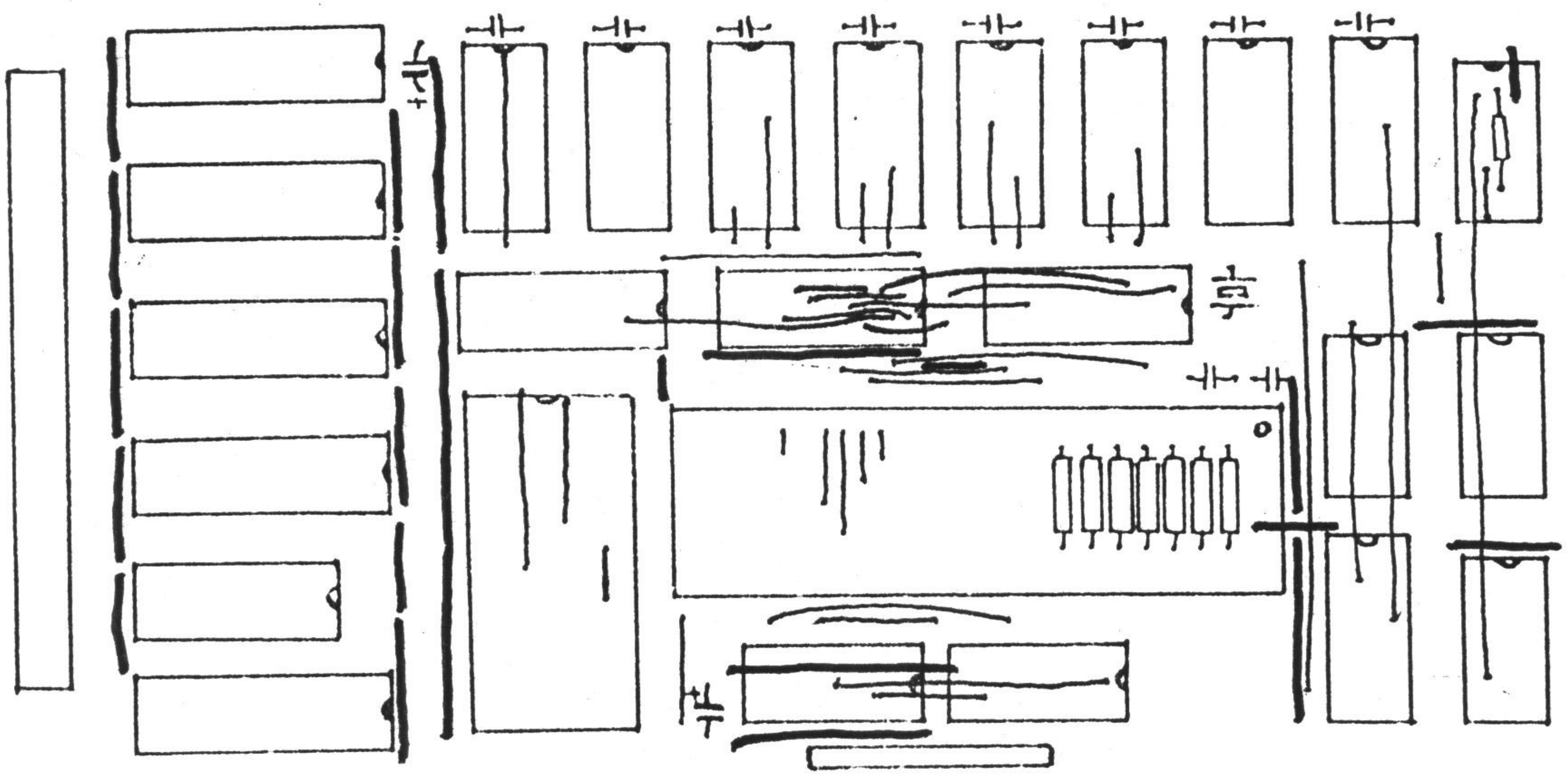


Abb. 5: Bestückungsplan

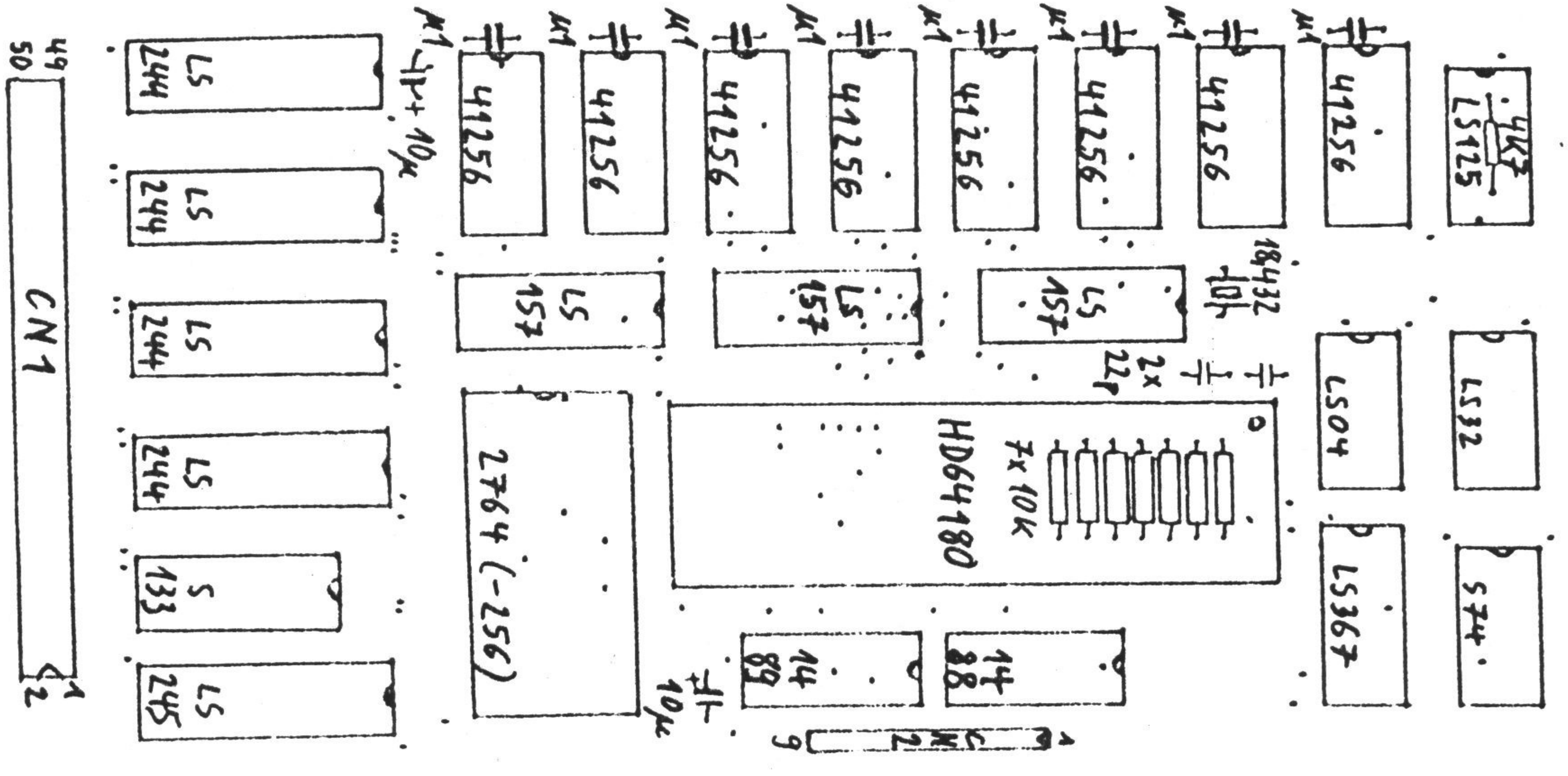
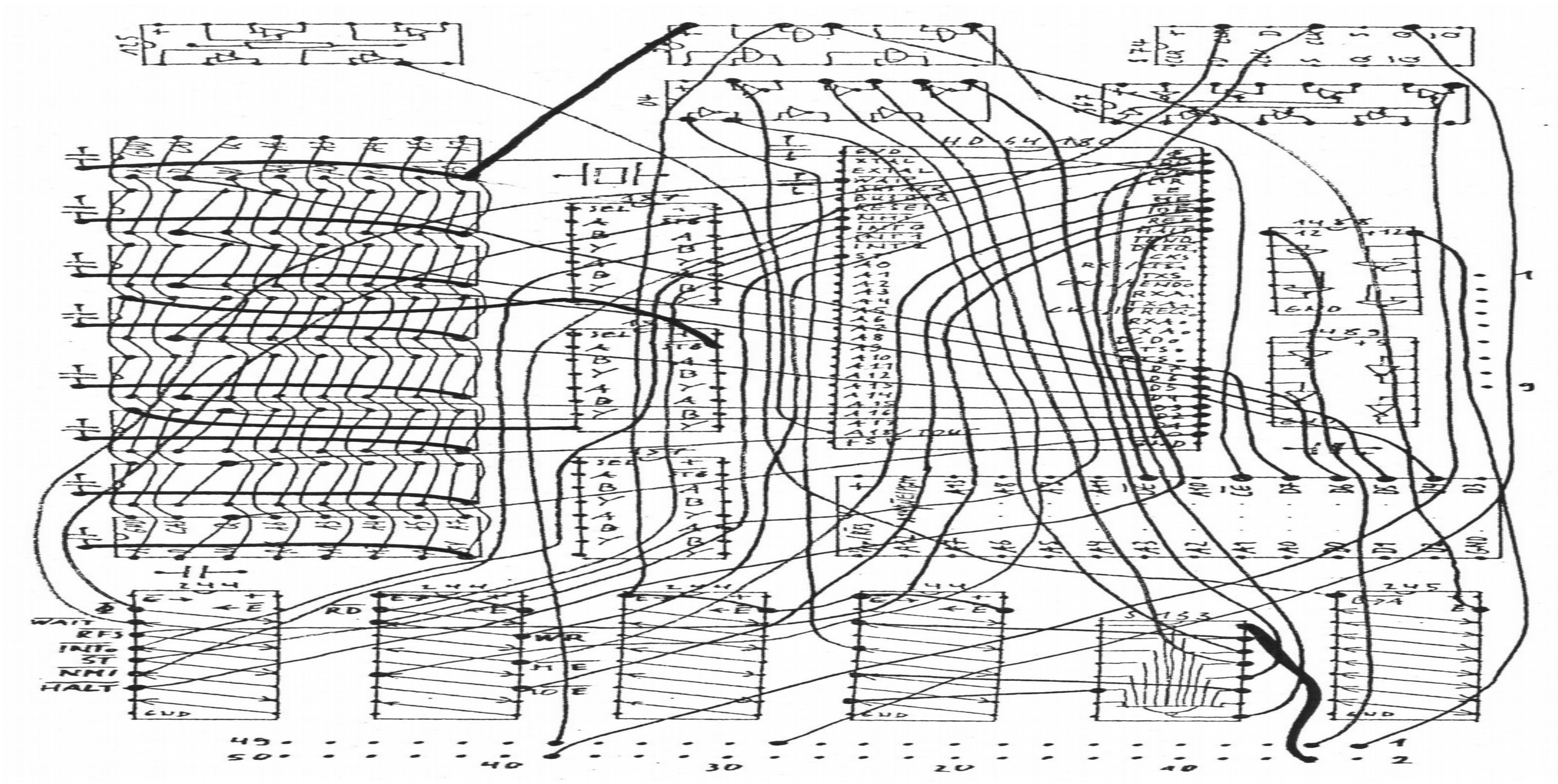


Abb. 6: Drahtbrücken auf der Lötseite (Ansicht von der Bestückungsseite)



Steckerbelegung von CN1
Verbindung zum Z80

Signal	Pin	Pin	Signal
+12V	1	2	+ 5V
-12V	3	4	+ 5V
D3	5	6	D4
D2	7	8	D5
D1	9	10	D6
D0	11	12	D7
A4	13	14	A5
A6	15	16	A7
A0	17	18	Al
A2	19	20	A3
A14	21	22	Al5
Al0	23	24	All
A8	25	26	A9
BUSRQ	27	28	nc
A13	29	30	Al2
A16	31	32	A17
A18	33	34	RD*
IORQ*	35	36	nc out
RESET*	37	38	BUSRQ*
NERO*	39	40	WR*
HALT*	41	42	nc in
NMI*	43	44	Ml*
INT*	45	46	RFSH*
WATT*	47	48	Clock
GND	49	50	GND

Steckerbelegung von CN2
serielle Schnittstellen

Pin	Signal
1	RTS1*
2	TXA0
3	TXA1
4	RTS0*
5	GND
6	CTS0*
7	CTS1*
8	RXA0
9	RXA1

Aufbau der Coprozessor-Karte

Zunächst werden alle Bohrungen mit einem 0,8mm-Bohrer angelegt. Nur die Anschlüsse für die Stecker CN1 und CN2 werden in 1mm Weite gebohrt. Nun können die bestückungsseitigen Drahtbrücken eingesetzt werden. Dafür sollte allgemein dünner Fädeldraht benutzt werden. Nur die im Drahtbrückenplan dicker eingezeichneten Brücken sollen mit Schaltdraht realisiert werden. Anschließend können dann Widerstände, Kondensatoren, der Qartz, die IC-Sockel und die Pfostenstecker eingesetzt werden.

Die abschließende, zeitraubendste Arbeit ist schließlich die freie Verdrahtung mit Fädeldraht auf der Lötseite gemäß Verdrahtungsplan. Auch hier sollen die dick eingezeichneten Verbindungen (für +5V und GND) mit Schaltdraht gelegt werden.

Der Anschluß der Karte kann allgemein mit den entsprechend bezeichneten Pins von CN1 an den Sockel der 280-CPU erfolgen. A16 und A17 können mit entsprechenden Signalen eventuell vorhandener Banking-Schaltungen gemultiplext werden, wobei dann BUSRQ* als Steuersignal verwendet wird.

Die Versorgungsspannungen +12V und -12V für den Schnittstellentreiber des HD64180 sind nur dann anzulegen, wenn die seriellen Schnittstellen des HD aufgebaut werden. Wenn diese nicht benötigt werden, brauchen die ICs 1488 und 1489 sowie der Stecker CN2 nicht eingesetzt zu werden.

Je nach Computer ist dann noch eine mehr oder weniger aufwendige Umschaltung des Systemtaktes und eventuell eine andere Erzeugung der RAM-Steuersignale RAS*, MUX* und CAS* nötig. Wenn außer RAM auch noch andere Baugruppen memory-mapped sind, müssen in die Freigabesignale von RAM und diesen Baugruppen auch noch A16 17 eingehen, um bei Adressierung durch den HD diese Baugruppen innerhalb der möglichen 256K nicht 4mal vorliegen zu haben. Bei Computern mit nur 64K RAM bleiben A16 und A17 unberücksichtigt.

Durch Ersetzen der 4164-RAMs durch 41256-RAMs und Einsetzen eines 8Bit-RefreshGenerators läßt sich der Speicher des Computers selbst auf 256K erweitern, ohne daß eine zusätzliche MMU nötig ist. Dann kann aber nur der HD diesen Speicher voll adressieren; der Z80 hat weiterhin nur auf 64K Speicher Zugriff. Dabei wären dann A16 und A17 des HD mit GND unter Kontolle von BUSRQ* zu multiplexen.

Wenn der HD die Kontrolle über den Computer hat und in seinen oberen 256K adressiert, ist der Datentreiber des HD64180-Boards gesperrt. Die Adressen und Steuersignale werden aber zum Computer durchgeschaltet, um dort ein Refreshing der RAMs aufrecht zu erhalten. Bei Schreibzugriffen auf die oberen 256K des HD erfolgen also auch Schreibzugriffe auf die entsprechende Adresse in den unteren 256K, nur daß dort dann nicht die Daten des HD sondern zufällige Werte des offenen Datenbus ankommen. Um dies zu verhindern, muß A18 so in die Herleitung aller memory-mapped Baugruppen eingreifen, daß ein Zugriff auf diese Baugruppen nur dann verhindert wird, wenn das Coprozessor-Board angeschlossen ist, der HD den Computer kontrolliert und auf seine oberen 256K RAM zugreift. Wenn das Board nicht angeschlossen ist oder wenn der HD bei Kontrolle des Computers in den unteren 256K des Computers adressiert (A18 ist low) müssen diese Baugruppen freigegeben werden, wenn sie adressiert werden. Dies leistet ein Treiber eines 74LS125-ICs und ein Widerstand. Das dabei gewonnene Signal MMEN* ist im Computer an entsprechenden Stellen in die Herleitung solcher Freigabesignale mit einzubeziehen (siehe unten).

Abb. 7: 8-Bit-Refresh Generator

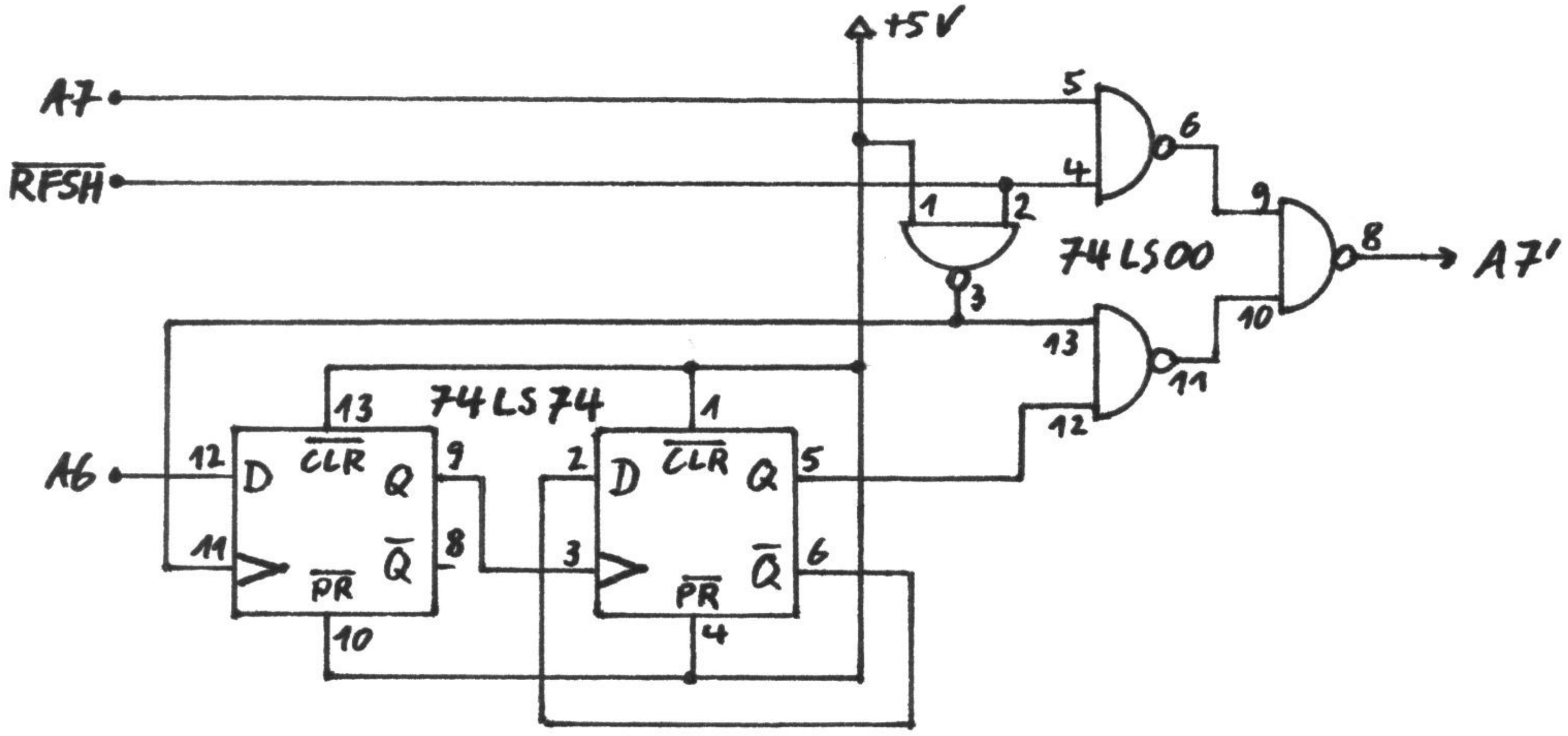


Abb. 8: Erzeugen der Adresse MA8 für 41256-RAMs in Computern ohne Banker

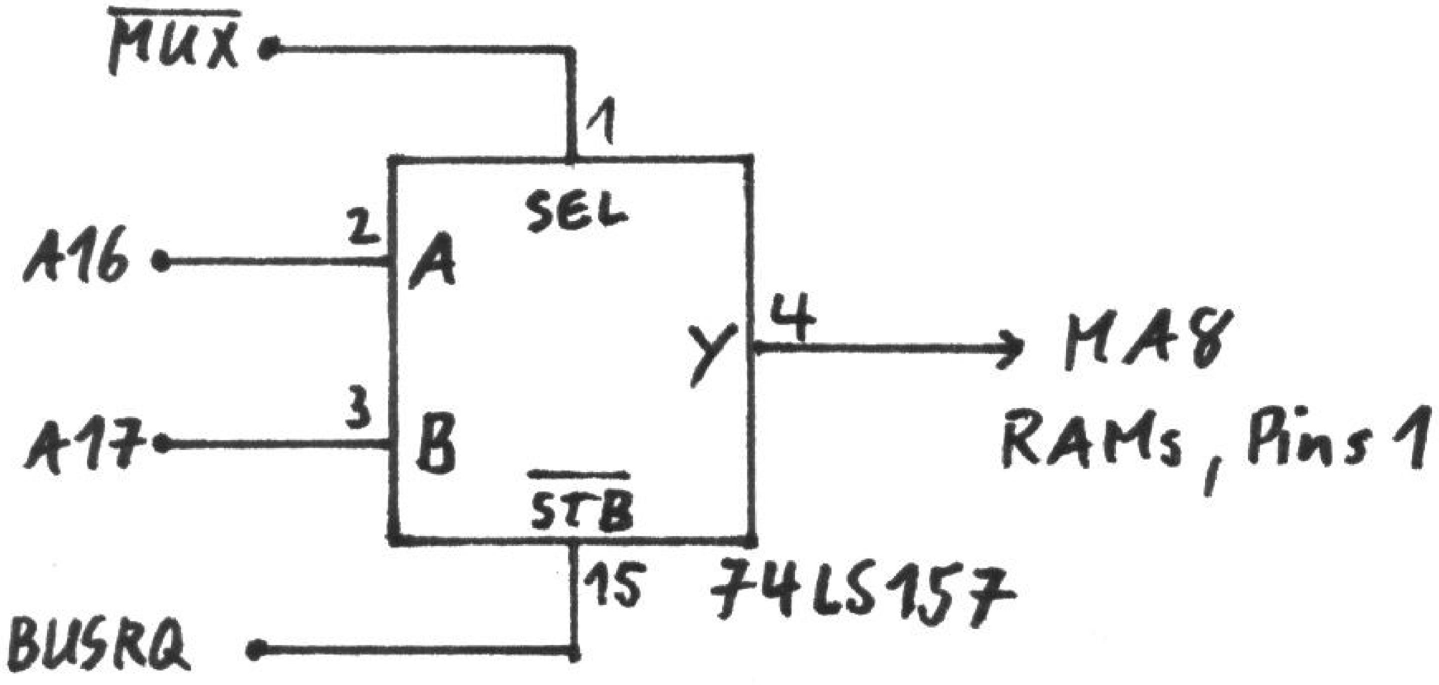


Abb. 9: Erzeugen eines Int2-Interupts bei Out FBh des Z80

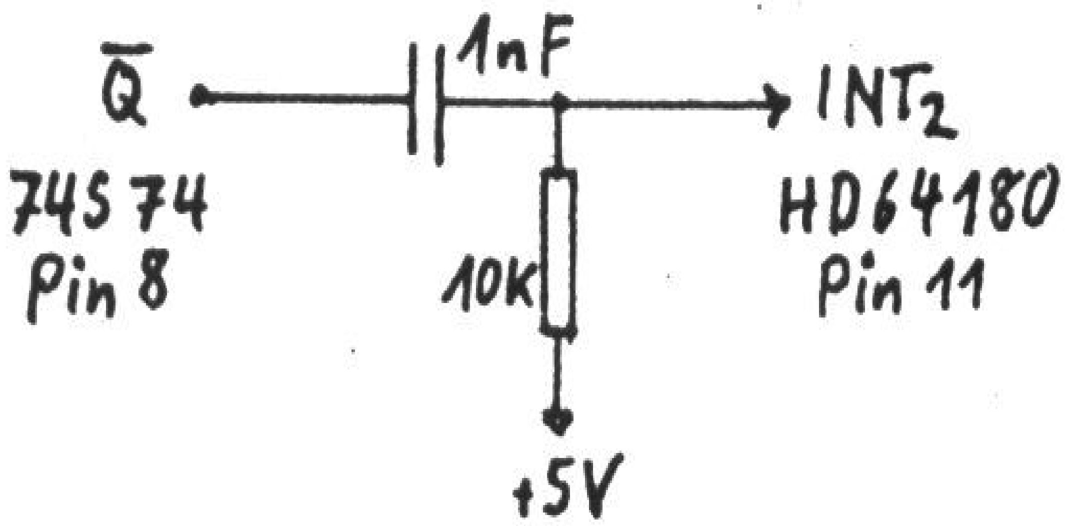
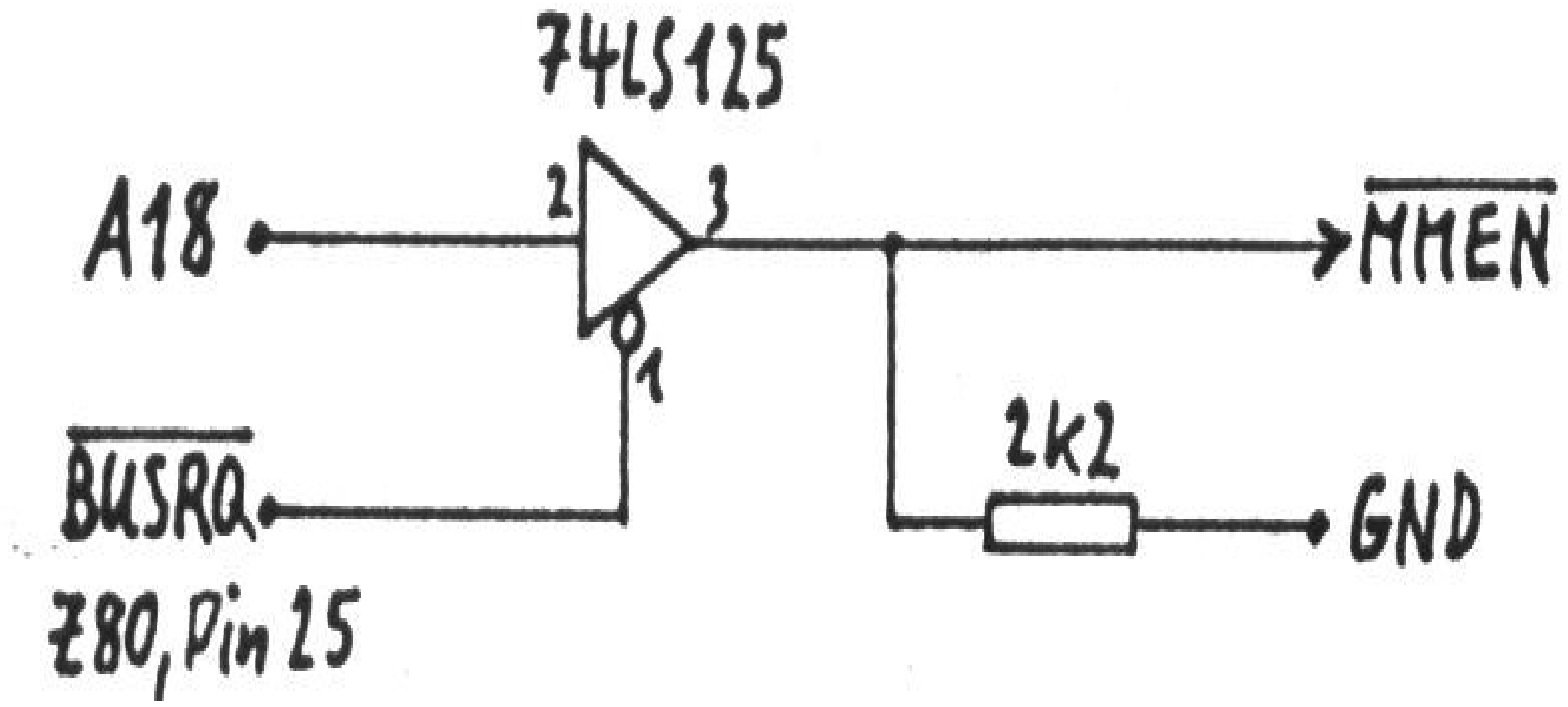


Abb. 10: Herleiten eines Steuersignals für die Freigbe von memory-mapped Baugruppen

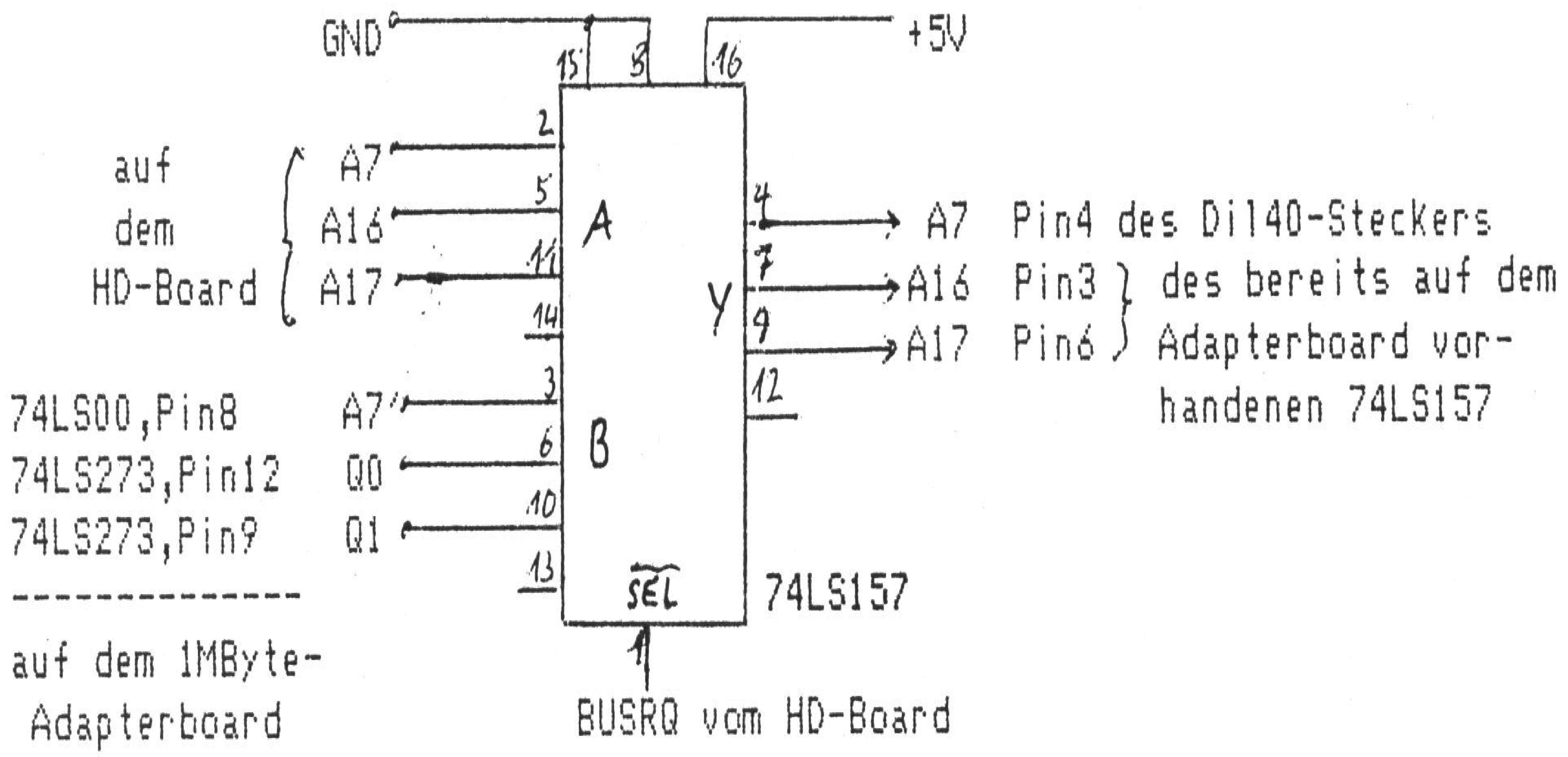


Änderungen im Computer

1) GENIE IIIs mit Erweiterung auf 1M Byte gemäß "Dein GIIIs, die 4 unbekannten Wesen"; Sonderheft des Club 80 Infos, 1986

Auf dem Adapterboard für die 1M Byte-Erweiterung wird das freie Rasterfeld für ein Dil-14-IC auf ein Dil-16-Feld erweitert, wobei dann GND vom Pin 7 auf den neu hinzugekommenen Pin 8 verlegt wird. Hier wird ein 74LS157-IC eingesetzt und gemäß folgender Abbildung beschaltet.

Abb. 11: Multiplex der höheren Adressen von HD64180 und 1M-Erweiterung

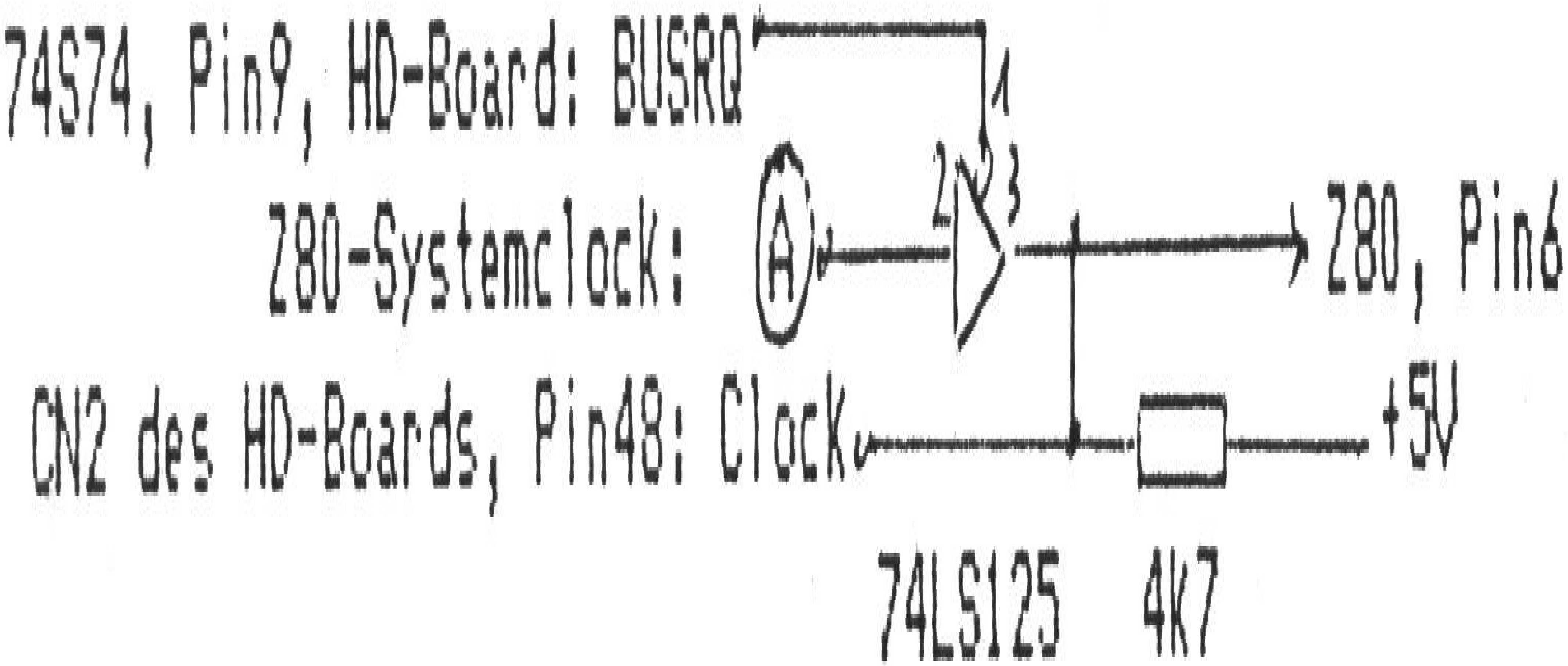


Die vorhandenen Verbindungen der Signale zu den B-Eingängen dieses zusätzlichen LS157 mit den Zielpunkten der '-Ausgänge sind zu durchtrennen.

Durch diese Änderung wird nur das über DO und D1 von Port F3H bewirkte Banking durch die MMU des HD64180 ersetzt, wenn dieser die Regie übernimmt. Das ursprüngliche Banking von nur 256K des GIIIs bleibt auch bei Kontrolle des Computers durch den HD bestehen. Einschränkend für die flexible MMU des HD bleibt hier aber auch die in Hardware durch D6 und D7 von Port F9H einzustellende Common-Bedingung wirksam. Die Programmierung der MMU des HD sollte diesen Voraussetzungen angepaßt werden Common entweder bei 0-16K oder bei 56-64K).

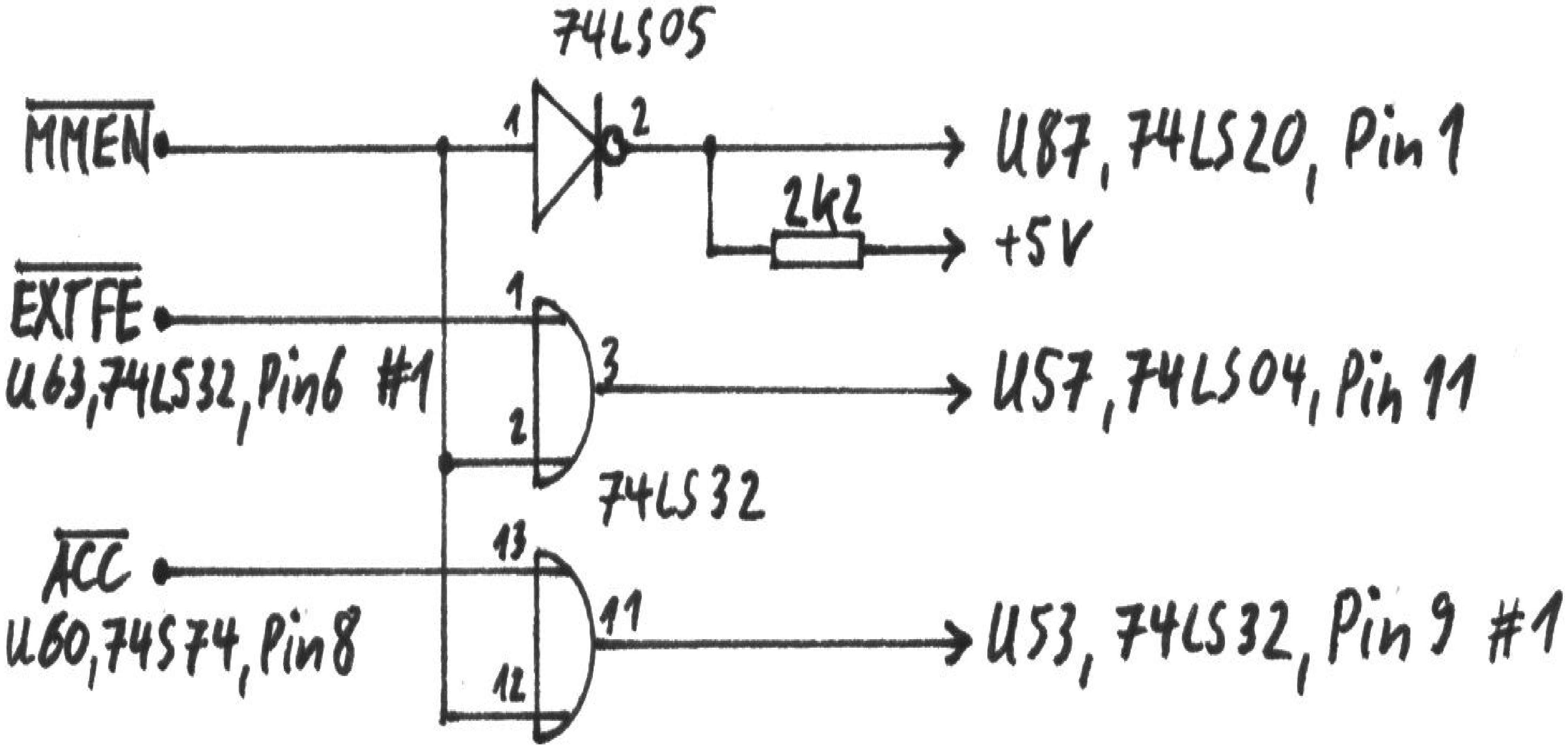
Das Umschalten des Systemtaktes kann durch Zwischenschalten eines Treibers mit Freigabe-Eingang (z.B. 74LS125) zwischen GIIIs-Taktquelle, U44, 74LS161, Pin12 und U39, 74L5125, Pin6 erfolgen. Die Freigabe des Treibers erfolgt durch BUSRQ, Pin27 von CN1 des HD-Boards. An Pin5 von U39 wird außerdem auch das Clock-Signal der Steckerleiste CN1 des HD-Boards gelegt. Durch die Steuerung des Treibers des GIIIs-Taktes und der Bustreiber des HD-Boards ist gewährleistet, daß nur einer der beiden Takte dem Computer zugeführt wird.

Abb. 12: Taktumschaltung zwischen System-Clock und HD-Clock



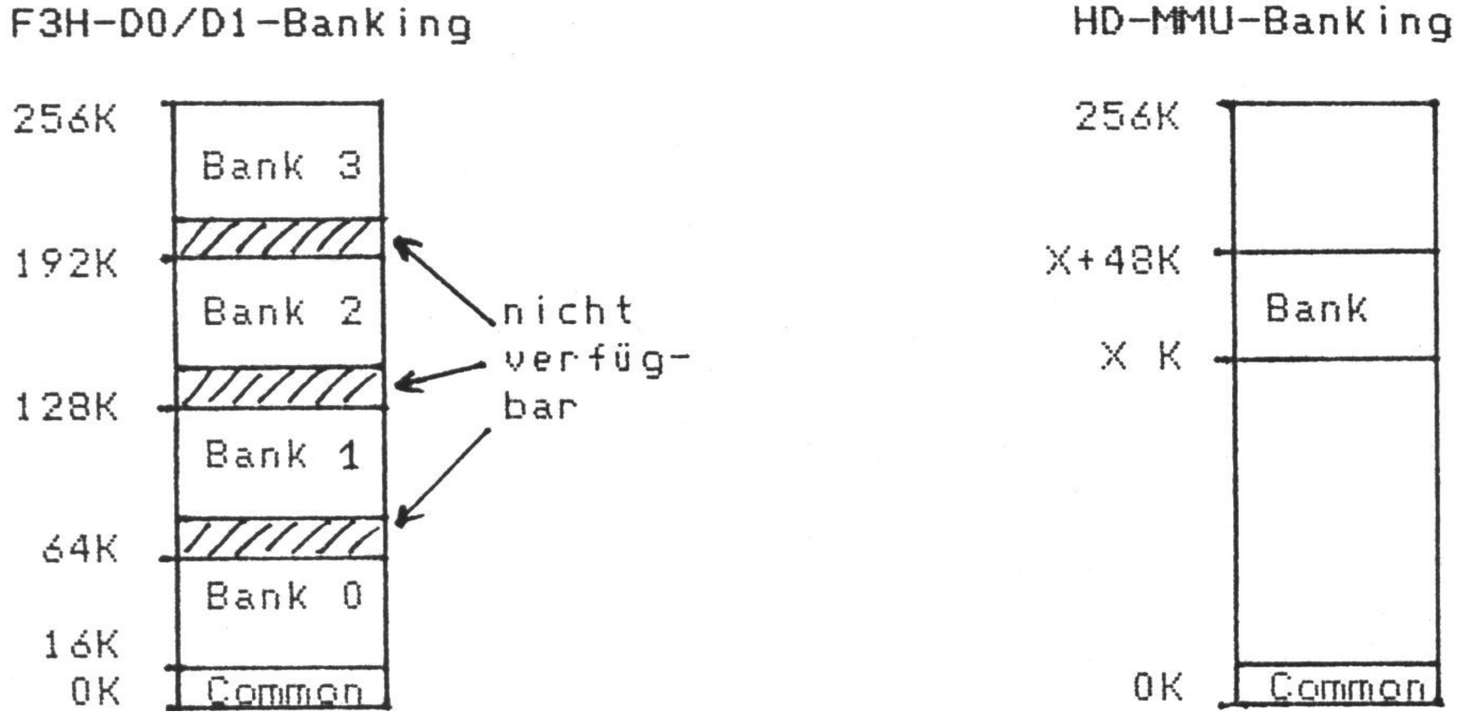
Um den Zugriff auf memory-mapped Baugruppen bei Adressierung der oberen 256K des HD zu verhindern, ist das aus A18 und BUSRQ* hergeleitete Steuersignal MMEN* folgendermaßen zu verwenden. Die dabei einbezogenen Pins (U62, 74LS32, Pin6 und U53, 74LS32, Pin9) sind aus der Fassung zu biegen.

Abb. 13: Eingriff in die Erzeugung der Freigabesignale der memory-mapped Baugruppen des GIIIs



Durch die Hardware des GIIIs ist vorgegeben, daß innerhalb der ursprünglichen 256K je nach Lage von Common (0-16K oder 56-64K) diese Bereiche in den oberen 3 Banks nicht verfügbar waren. In den bei der 1MByte-Erweiterung dazugekommenen 3 x 256K war durch Bit 7 in Port F3H steuerbar, ob bei logischer Adressierung innerhalb des als Common festgelegten Adreßbereichs ein Zugriff auf die absolute 64K-Bank 0 oder auf die relative Bank 0 innerhalb des durch D0 und D1 in Port F3H eingestellten 256K-Blocks zugegriffen wird. Die Unterschiede der beiden Banking-Speicheraufteilungen sind folgende:

Abb. 14: Speichereinteilung beim Banking mit Z80 über D0, D1 von Port F3H und HD64180-MMU



Das HD-Banking ist der in Hardware vorgegebenen Common/Bank-Einteilung der logischen 64K des GIIIs anzupassen.

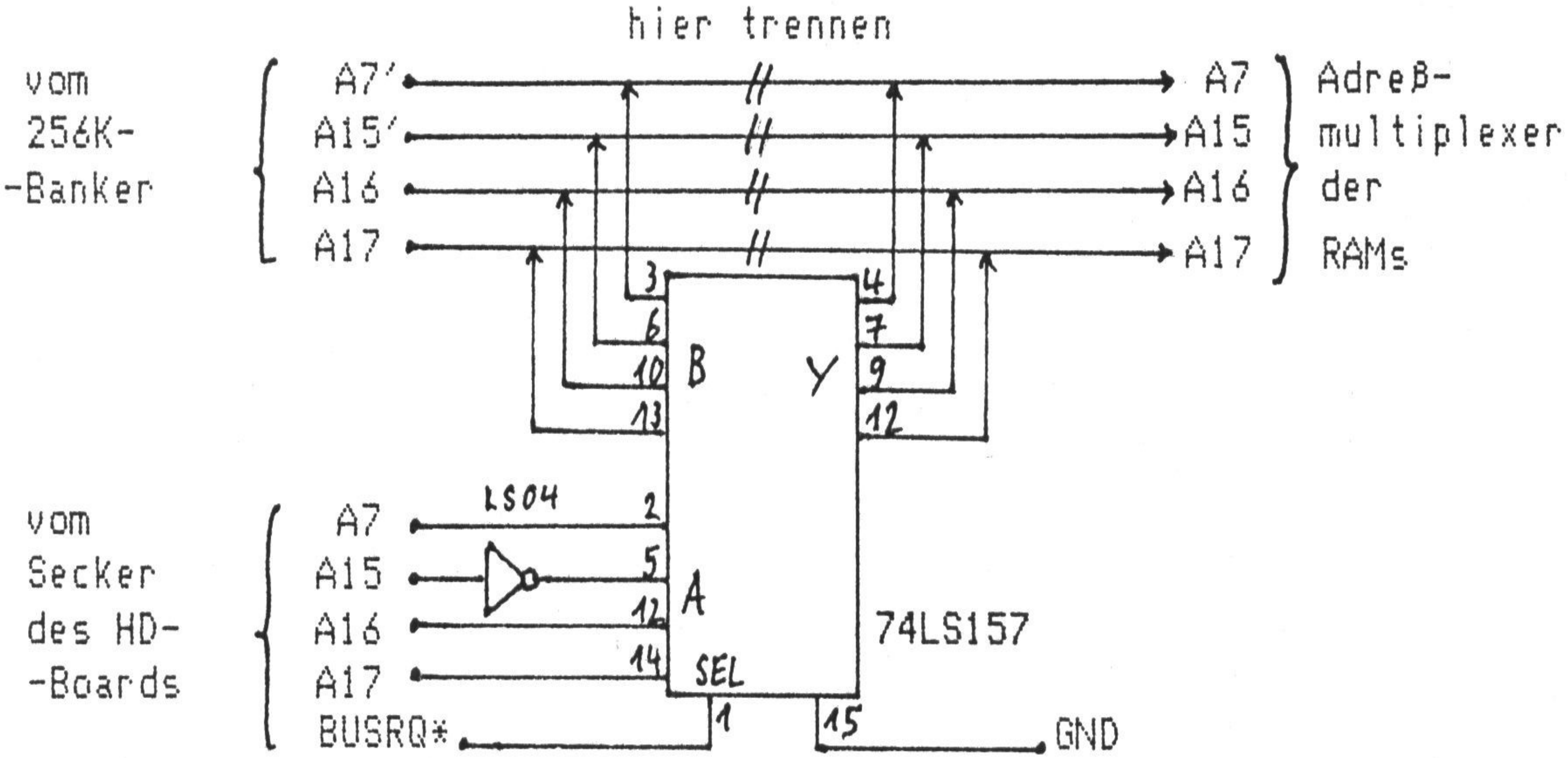
Die MMU des HD ist dafür folgendermaßen zu programmieren:

```
1) Common bei 0-16K:
CBAR : 44H      Common 0 logisch von 0000H bis 3FFFH
                Common 1 logisch von 4000H bis FFFFH
                Bank Area ist nicht definiert
BBR  : 00H      hat keine Bedeutung
CBR  : 00H, 10H, 20H, 30H
                für Banks 0-3 analog zur Speichereineinteilung
                beim Banking mit D0 und D1 in Port F3H des Z80
2) Common bei 56-64K:
CBAR : E0H      Common 0 ist nicht definiert
                Common 1 logisch von E000H bis FFFFH
                Bank Area logisch von 0000H bis DFFFH
CBR : 00H       hat keine Bedeutung
BBR : 00H, 10H, 20H, 30H
                für Banks 0-3 analog zur Speichereinteilung
                beim Banking mit D0 und D1 in Port F3H des Z80
```

2) TRS80, Model I, GENIE I und II und Komtek 1 mit 256K (1M)-Banker gemäß Infos der TRS80/GENIE-User-Clubs, 1986

Zum Umschalten zwischen den Adressen A7, A16 und A17 des Bankers unter Z80-Kontrolle oder der MMU des HD64180, wenn dieser den Computer kontrolliert, ist ein 74LS157-IC nötig, das zwischen Banker und RAM-Adreßmultiplexer geschaltet wird. Dieses IC wird folgendermaßen beschaltet:

Abb. 15: Multiplex der höheren Adressen des HD64180 und des 256K-Bankers



Durch Huckepacklöten auf ein anderes 74LS157-IC mit den Pins 8 und 16 entweder auf dem CPU-Board oder auf dem Banker ist auch die Versorgungsspannung für dieses IC vorhanden. Um unter 280- und HD-Kontrolle beim Banking gleiche physikalische Speicher anzusprechen, sollte A15 des HD-Boards über einen 74LS04-Inverter angeschlossen werden, wenn dies zum Banking der oberen 32K für den Banker des Z80 auch gemacht wurde.

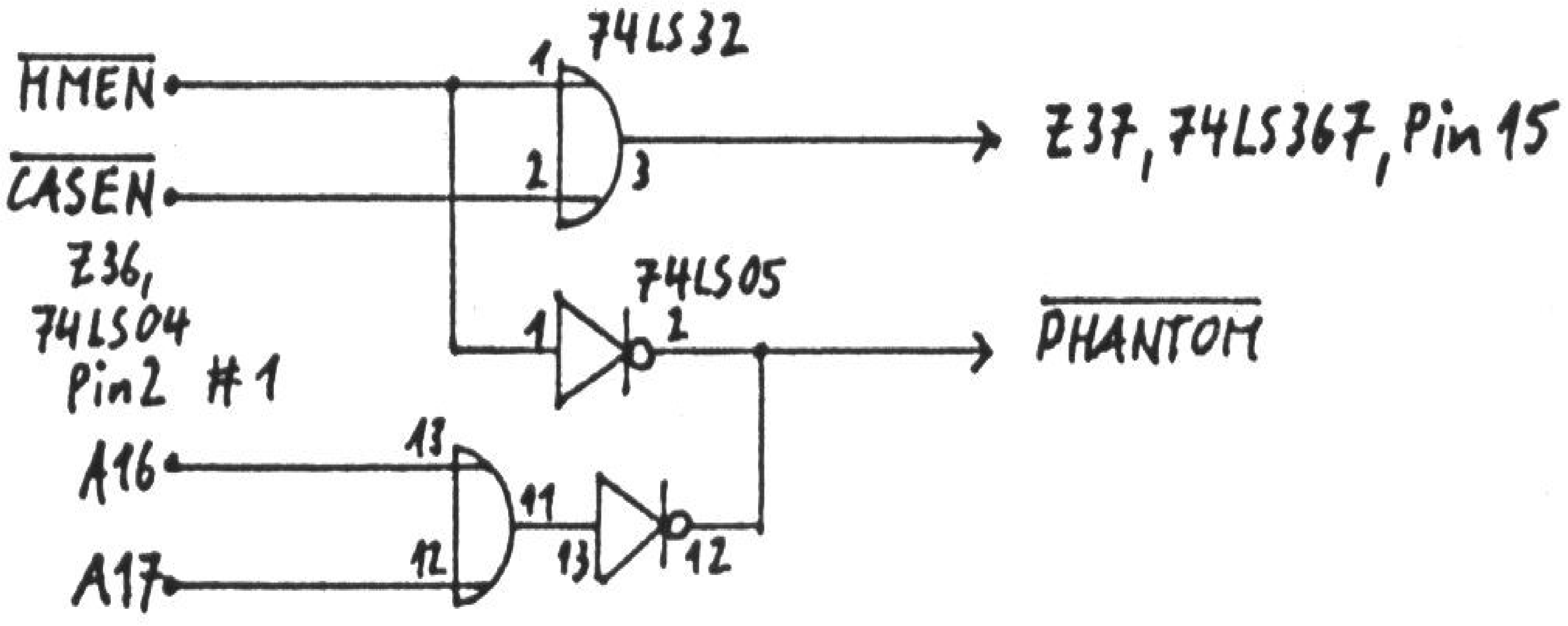
Die Decoderschaltungen für die Freigabe von RAM und ROM sowie memory-mapped I/O im TRS80, GENIE und Komtek 1 berücksichtigen nur die Adressen bis A15. Da ROM und memory-

mapped I/O aber unter HD-Kontrolle nur dann freigegeben werden sollen, wenn auch A16, A17 und A18 low sind, müssen diese Signale in der Decoderschaltung noch berücksichtigt werden.

Dafür werden die Signale A16 und A17 von den Ausgängen des zusätzlichen 74LS157 und das aus A18 und BUSRQ* hergeleitete MMEN* verwendet.

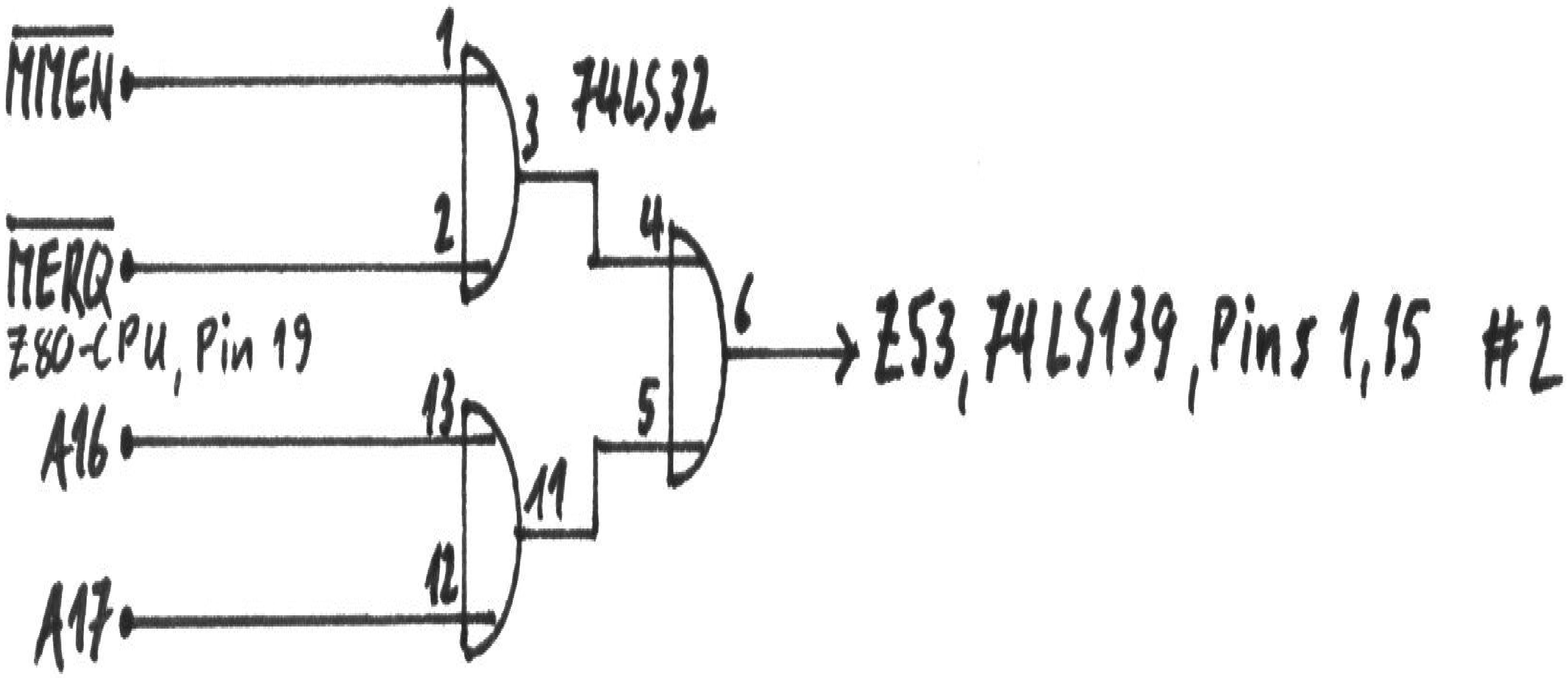
#1 die direkte Verbindung zu Z37, Pin15 wird wieder entfernt Eingriff in die Decoderschaltung bei GENIE I und II

Abb. 16: Eingriff in die Decoderschaltung bei Genie I und II



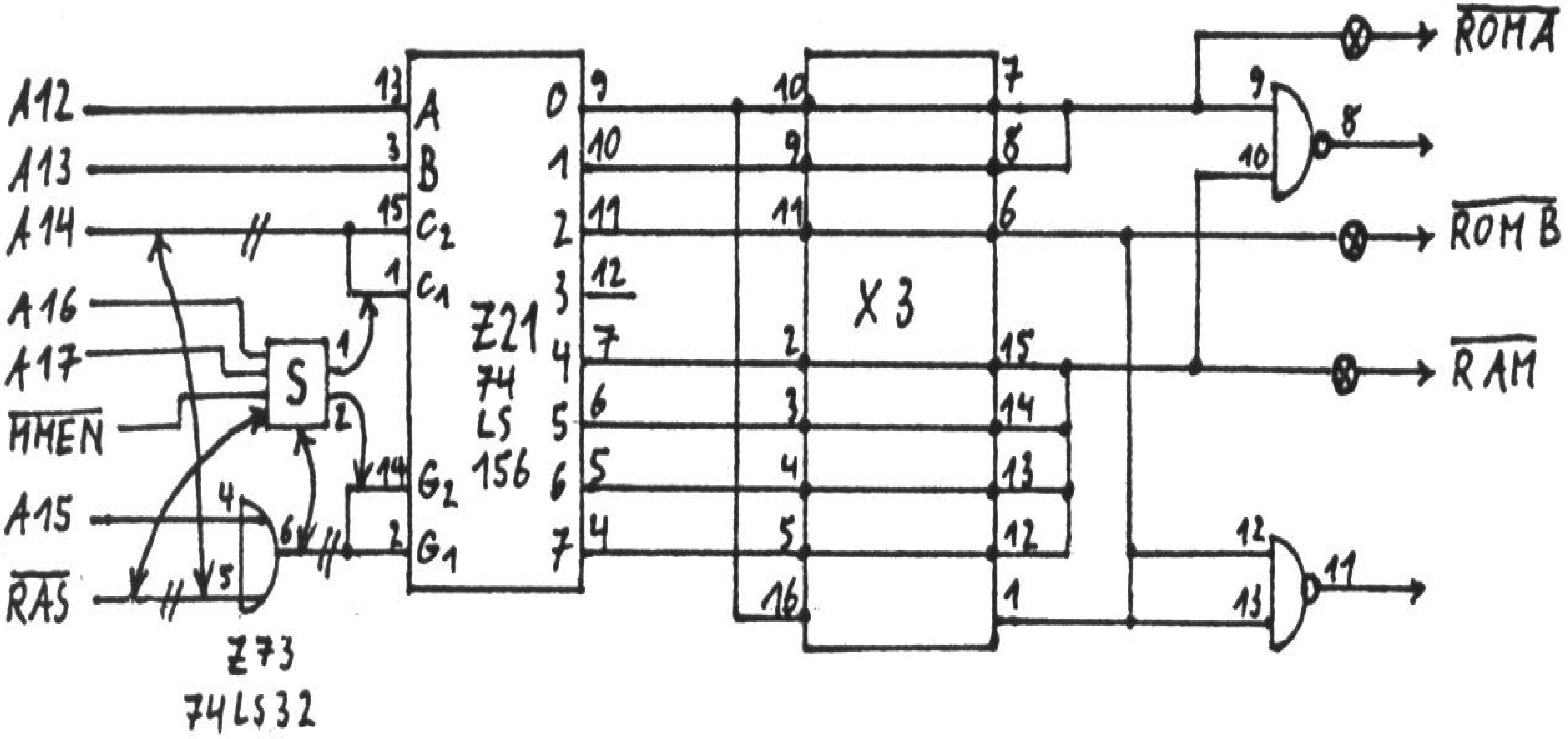
#1 die direkte Verbindung zu Z37,Pin15 wird wieder entfernt

Abb. 17: Eingriff in die Decoderschaltung beim Komtek 1
#2 das bisher an diese Pins gelangende CAS wird abgetrennt*



Im TRS80 ist eine etwas aufwendigere Methode nötig. Zum Teil sind die in der folgenden Abbildung dargestellten Änderungen schon beim Umrüsten auf 64 bzw. 256K RAM auf dem CPU-Board durchgeführt worden.

Abb. 18: Änderungen am Adreßmultiplexer des TRS80
S *links im Bild ist die hierauf folgende Schaltung*

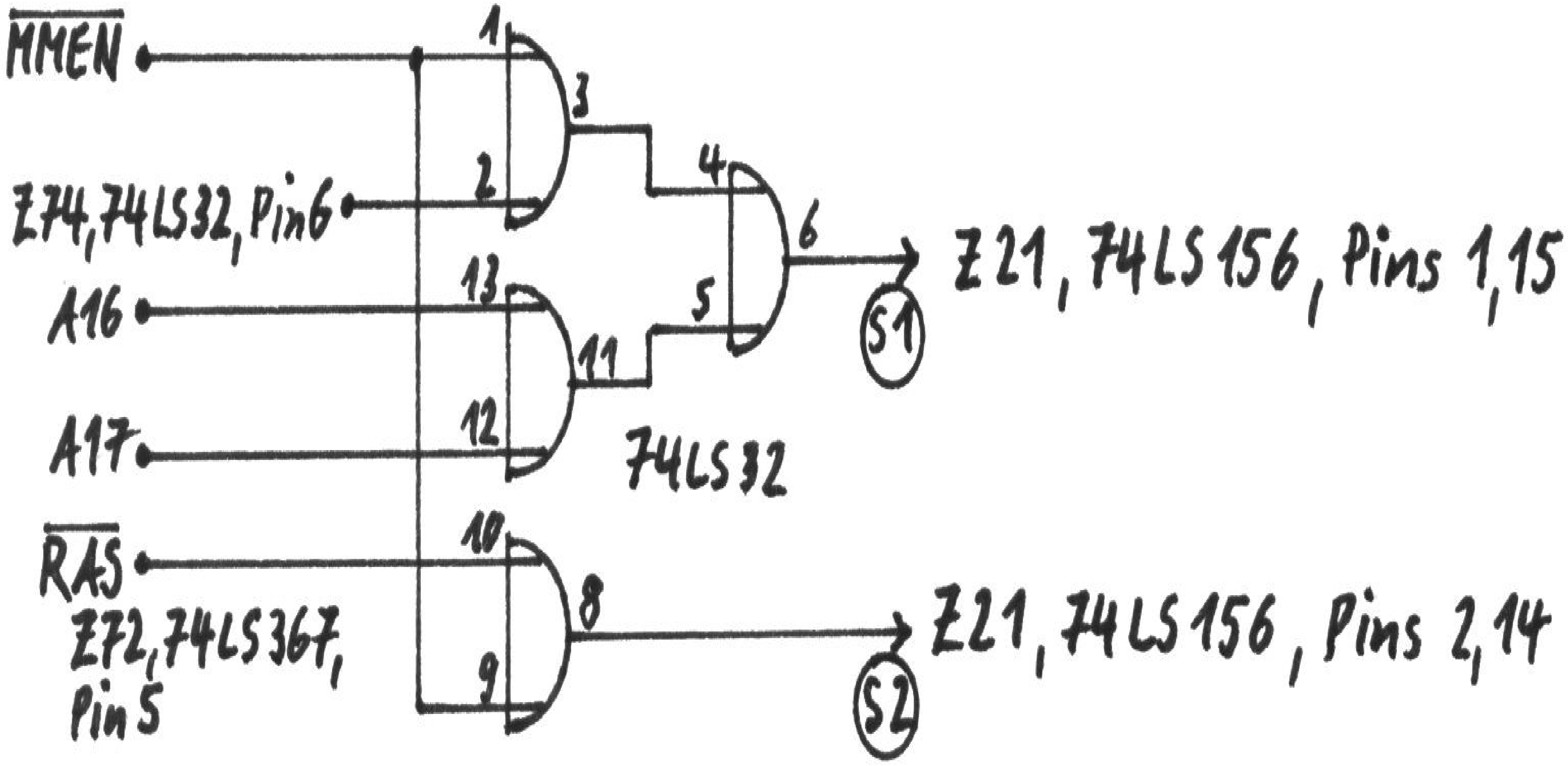


Anstelle von A14 wird der Ausgang S1 an die Pins 1 und 15 von Z21 gelegt.

Anstelle von Pin 6 von Z73 wird der Ausgang S2 an die Pins 2 und 14 von Z21 gelegt.

Anstelle von RAS* wird A14 an Pin 5 von Z73 gelegt.

Im Jumperfeld X3 werden alle Brücken außer 1-16 gelegt. Zusätzlich wird mit einem an den Pins 7 und 14 an geeigneter Stelle huckepack-gelöteten 74LS32 folgende Zusatzschaltung realisiert. ***Abb. 19: Zusatzschaltung S für den Adreßdecoder***

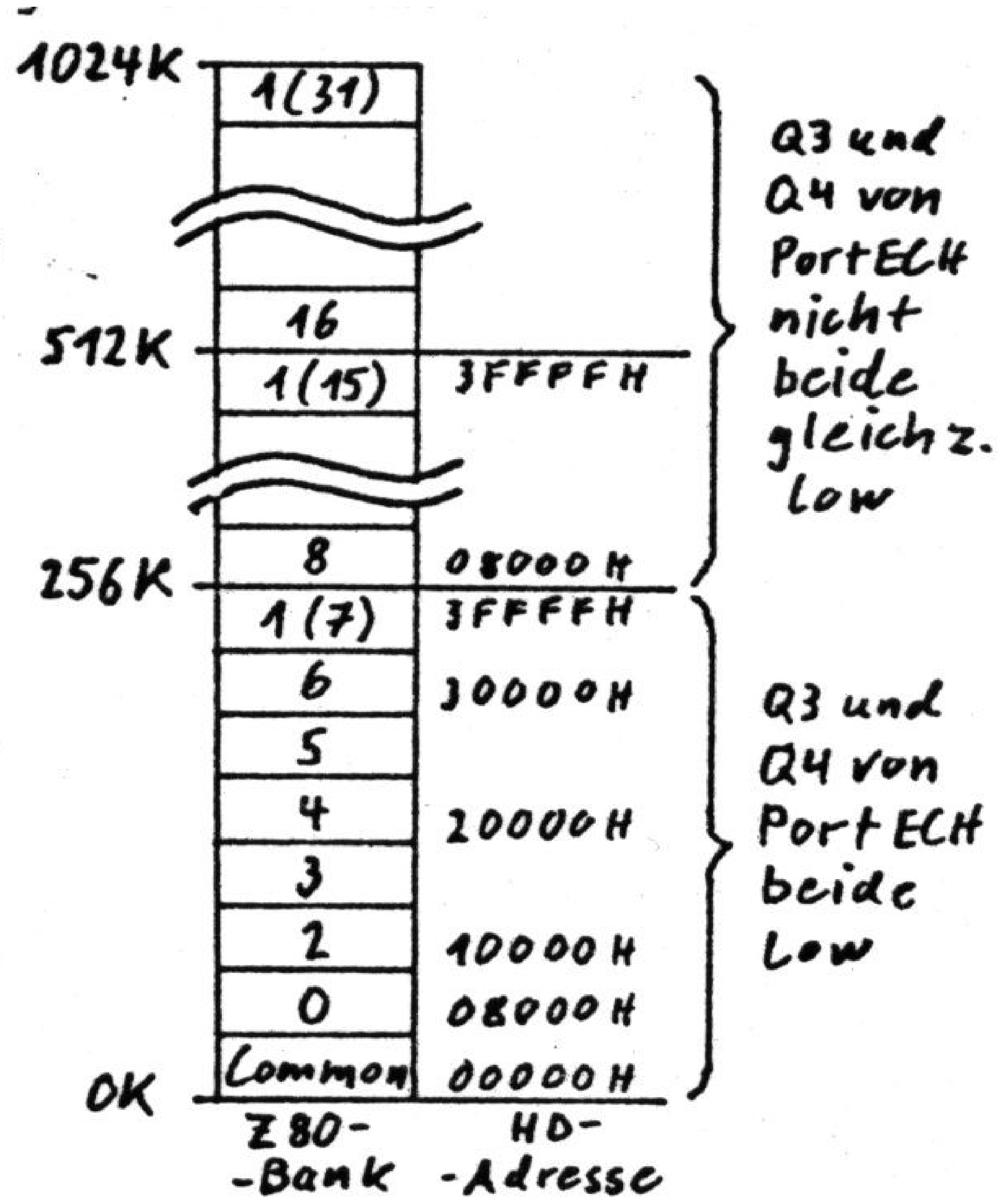


Da der HD im Computer nur 256K adressieren kann, muß ein Zugriff auf darüber hinausgehende Speichergrößen unter Zuhilfenahme des Bankers erfolgen, wobei dan die Bits 0, 1 des Ports ECh, mit denen per Banking innerhalb von 256K adressiert wird, wirkungslos sind, weil die Adressen A15, A16 und A17 des HD dann durchgeschaltet werden. Es muß dann das Banking des HD dem des Bankers angepaßt werden: Common 0 bei 00000H-07FFFH und Common 1 bei 08000H-0FFFFA (für Bank 0). Banking mit der MMU des HD muß dann durch Verschieben von Common 1 um jeweils 32K erfolgen.

D0, D1 und D2 in Port ECA haben für den HD64180 keine Bedeutung; wenn aber später der Z80 wieder den Computer übergeben kriegt, werden diese Bits des Port ECH aktiv. Das kann dann unangenehme Folgen haben, wenn bei mehr als 256K im Computer der HD mit D3 (und D4) in Port ECA auf den Gesamtspeicher zugreift und dabei auch D0, D1 und D2 verändert. Da Port ECH nicht ausgelesen werden kann, muß eine Vereinbarung zwischen den CPUs getroffen werden, daß z.B. bei Übergabe des Computers immer Bank 0 eingestellt sein muß.

Für die Übertragung bisheriger Banking-Software auf den Betrieb der MMU des HD gilt folgende Speichereinteilung und Entsprechung von Banks. Das ist zu berücksichtigen, wenn HD und Z80 abwechselnd auf Daten und Programme in den Banks zugreifen sollen.

Abb. 20: Speicheraufteilung beim Banking mit 256K(1M)-Banker



Die Schaltung des Bankers gibt vor, daß die Bank 1 immer mit der höchsten verfügbaren Bank identisch ist

256K RAM : Bank1 = Bank7
512K RAM : Bank1 = Bank15
1024K RAM : Bank1 = Bank31

Diese jeweils höchste Bank sollte also nicht benutzt werden, um Konflikte mit Bank 1 zu vermeiden.

Speicheraufteilung beim Banking mit 256K(1M)-Banker

Für den HD besteht der Computer nur aus 256K Speicher, innerhalb derer er durch seine MMU oder per DMA adressieren kann. Bei logischen Adressen unterhalb von 8000H wird immer der erste 32K Speicherblock bzw. ROM oder Memory-mapped I/O adressier.Bei logischen Adressen ab 8000H wird die durch die MMU (Common-1-Lage)vorgegebene Bank innerhalb des vom Z80 durch Q3 und Q4 von Port ECh adressierten 256K-Blocks angewählt.

Eine entsprechend Bank 1 ses Bankers mögliche Adreßverschiebung für die MMU des HD findet nicht statt. Bei logischen Adressen oberhalb 7FFFh adreßieren Q3 und Q4 des Port FCh und die physikalische Lage von Common 1 direkt den Speicher.

Der Systemtakt läßt sich ähnlich, wie für den Genie IIIs gezeigt, umschalten, wobei der bisherige Systewmtakt (Punkt A in der bei der Beschreibung für den GIIIs gezeigten Schaltung) an den Eingang des Treibers gelegt wird und der Ausgang anstelle des bisherigen Taktes hier direkt an Pin 6 des Z80 geführt wird. Der Punkt A ist zu finden bei:

TRS80, M1	Z72, 74LS367, Pin 11
Genie I, II	Z37, 74LS367, Pin 3
Komtek 1	Z18, 74LS74, Pin 9

Eventuell muß an Pin 6 des Z80 noch ein Pullup-Widerstand von 4K7 gelegt werden, wenn ein solcher nach dem Abtrennen des direkten Systemtaktes nicht mehr Verbindung mit Pin 6 des Z80 hat.

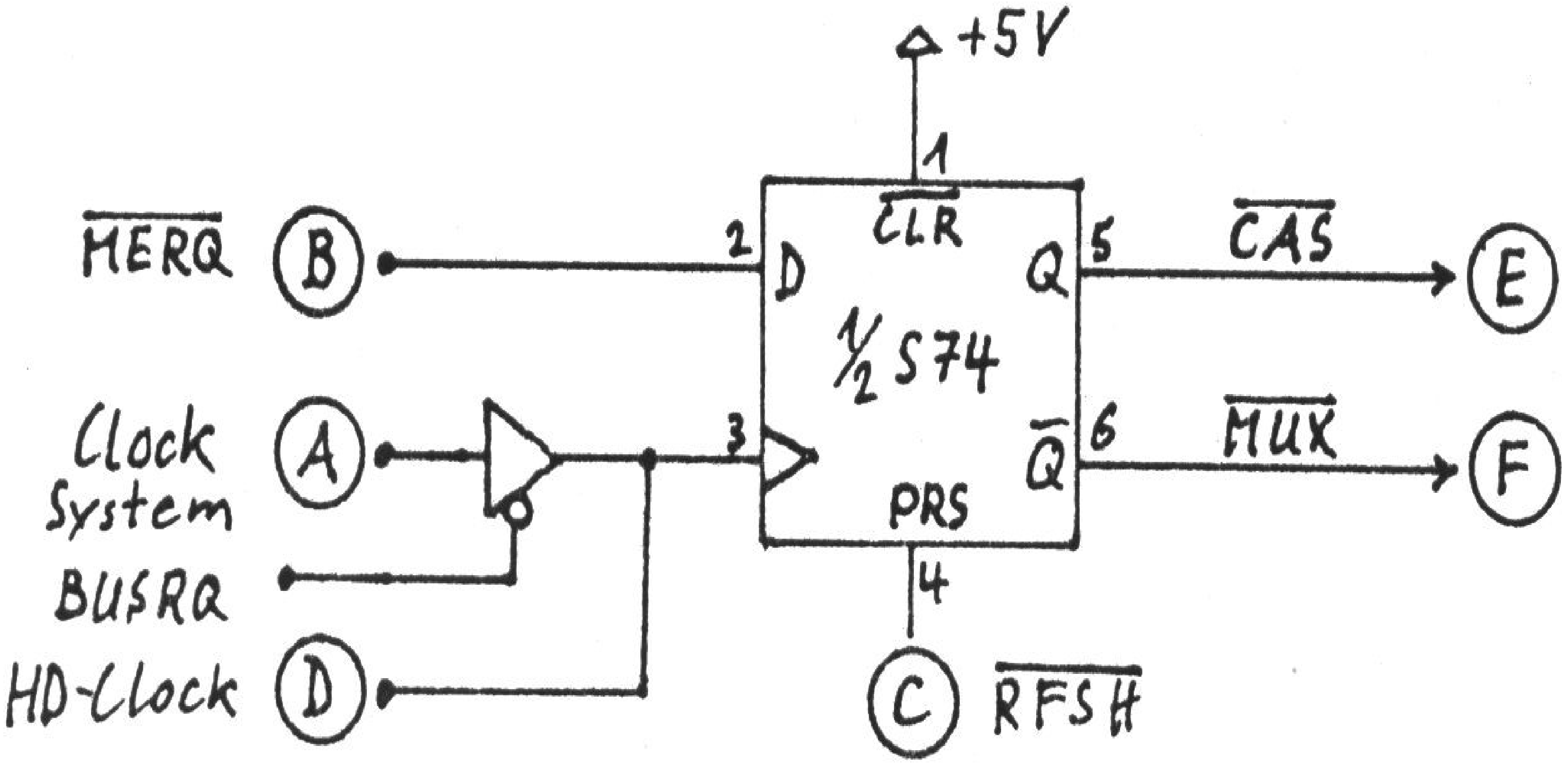
```
Die MMU des HD ist folgendermaßen zu programmieren:
CBAR : 88H      Common 0 logisch bei 0000H-7FFFH
                Common 1 logisch bei 8000H-FFFFH
                Bank Area nicht definiert
BBR : 00H       hat keine Bedeutung
```

```
CBR : 00H, 08H, 10H, 18H, 20H, 28H, 30H
                entsprechend den Banks (für Port ECH)
                0  2  3  4  5  6  1(7,15,31)
```

Zur Anwahl der Banks 8-14 bei 512K bzw. 8-30 bei 1024K sind diese Einträge in das CBR-Register in Kombination mit den Bits (3 und 4) des Ports ECH zu wählen. Zur Anwahl der Bank 1 sollten diese Bits immer high gesetzt werden.

Da bei TRS80 Model 1, GENIE I und II und Komtek 1 die RAM-Steuersignale nicht aus dem Systemtakt selbst sondern aus einem mehrfachen des Systemtaktes hergeleitet werden, muß diese Schaltung durch eine solche ersetzt werden, die den Systemtakt selbst zur Steuerung benutzt. Es muß dabei der an den Z80, Pin 6 gelangende Takt genommen werden, damit die RAM-Steuerung abwechselnd mit dem Z80-Takt oder dem HD64180-Takt synchron läuft. Dafür ist ein 74S74-IC und ein Gatter eines 74LS125-ICs nötig, das die Taktumschaltung (zusammen mit den Bustreibern des HD-Boards) erledigt.

Abb. 20: Erzeugen der RAM-Steuersignale MUX* und CAS* für unterschiedliche Systemtakte



Die in dieser Schaltung bezeichneten Punkte A bis F sind mit folgenden Punkten auf dem CPU-Board zu verbinden:

		TRS80 Model 1			GENIE I u. II			Komtek 1		
Punkt	Signal	IC	Typ	Pin	IC	Typ	Pin	IC	Typ	Pin
A	Z80CLK	72	367	11	37	367	3	18	74	9
B	RAS*	72	367	5	16	367	5	11	05	8
C	RFSH*	40	Z80	28	16	367	3	33	Z80	28
D	HDClock	40	280	6#1	5	Z80	6#1	33	Z80	6#1
E	CAS*	72	367	10#2	37	367	14#2	54	32	2#2
F	MUX*	72	367	2#2	37	367	4#2	30	157	1#2

#1 Die direkte Verbindung der bisherigen Taktquelle (Punkt A) mit Pin 6 des Z80 ist zu durchtrennen, auf den Systembus wird das Signal von Punkt D gelegt.
#2 Die bisher zu diesem Punkt führende Leitung wird durchtrennt.

1987-BHV-02 S.26-31

Die Grundintelligenz der HD 64180-Coprozessorkarte

Arnulf Sopp

Der HD 64180 kann seine erweiterten Fähigkeiten dem Computer nur auf Anforderung des Z80 zur Verfügung stellen. Er muß daher per Software in die Lage versetzt werden, eine solche Anforderung zur Kenntnis zu nehmen und zu interpretieren. Auf Helmut Bernhardts HD-Board ist ein EPROM vorgesehen, das dieses Programm enthält. Sein Inhalt ist im Listing 1 wiedergegeben.

Der HD liest nicht nur nach Reset, sondern auch nach einem TRAP-Interrupt an der Stelle logisch 0000h. Darauf wird hier nicht weiter eingegangen, weil Gerald Schröder darüber bereits schrieb. In diesem Falle verzweigt der HD in eine Fehlerbehandlungsroutine, die den unbekannten Befehl imitiert und dann zum alten Programm zurückkehrt. Diese Routine habe ich fast wörtlich von Geralds Programm übernommen, das in der 17. Ausgabe des Clubinfos abgedruckt ist. Deshalb sei hier nur auf den geringen Unterschied zu Geralds Routine hingewiesen: Die Interrupts müssen nicht mit DI maskiert werden, weil sich die Speicherkonfiguration während der Trap-Behandlung nicht ändert. Ein INT dürfte getrost eintreten. In diesem EPROM entfällt natürlich auch das Banking, das für das Genie 2s im Info beschrieben wurde. Ein TRAP kann im EPROM nicht auftreten, weil keine Illegals verwendet sind. Die Routine liegt daher nur zur Bearbeitung aus dem RAM bereit.

Wurde 0000h nach einem Reset gelesen, finden einige Initialisierungen statt. Für den leistungsfähigen HD-Befehl OTIMR (gib ein Datum aus (HL) auf Port (C) aus, erhöhe C und HL, erniedrige B, wiederhole dies, bis B=0 ist) wird HL als Zeiger auf einige Initialisierungsdaten (Label initab) aufgesetzt. Diese Daten konfigurieren im wesentlichen die MMU (Memory Management Unit) des HD, um aus dem Adreßraum von insgesamt 512 kB folgende "Fenster" der CPU zugänglich zu machen:

Common 0:	nicht definiert
Bank-Area:	0.0000-0.EFFFh
Common 1:	7.F000-7.FFFFh

In dieser Tabelle von Daten sind weitere Informationen für den HD enthalten, die aber für das Verständnis des Folgenden nicht von Bedeutung sind. Soweit sie sich nicht aus den Kommentaren des Listings erklären, möchte ich auf das Datenbuch verweisen.

Die CPU hat nun insgesamt 64 kB Speicher verfügbar, deren physikalische Lage oben dargestellt ist. Nach dem Befehl OTIMR wird HL nun auf das letzte Byte einer zweiten Tabelle gestellt (tabend). Sie enthält Daten für den DMAC (Direct Memory Access Controller). Er kann unabhängig von der Konfiguration der MMU Daten aus jedem beliebigen Teil des Gesamtspeichers in jeden beliebigen anderen transferieren. Die Werte vom Label srclog an besagen in diesem Falle, daß die Daten ab dma0 bis zum Label prt in den physikalischen Speicherbereich ab 7.F000h übertragen werden sollen. Die beiden Bytes davor in der Tabelle bestimmen die Art des DMA-Zugriffs und führen schließlich zum Start des DMA.

Die Routine zur Programmierung des DMA-Controllers (ab Label dma0) ist so gestaltet, daß der Programmierer nicht weiter über die einigermaßen komplizierten Port-Ausgaben bescheidwissen muß. Analog zum Befehl LDIR lädt er lediglich HL mit der Quelladresse, DE mit der Zieladresse (beide innerhalb je eines beliebigen physikalischen 64-kB-Blocks) und BC als Bytezähler. Zusätzlich muß nur noch im Akku angegeben werden, in welchem physikalischen 64-kB-Block des Gesamtspeichers von 512 kB Quelle und Ziel zu suchen sind. Die Bits 0-2 tragen diese Information für die Quelle, Bit 4-6 für das Ziel. Ich habe diese Bits bewußt ausgewählt, weil dann der Akku Quelle und Ziel sozusagen im Klartext, nämlich in je einer Hex-Ziffer enthält. Je nach dem, welche Vorarbeit bei der Programmierung des DMAC schon geleistet ist, kann auch an späterer Stelle eingesprungen werden.

Der EPROM-Inhalt von srclog bis prt-1 ist nun ins obere RAM übertragen. Er enthält auch eine Interrupt-Vektortabelle für den Fall des IM 2. Der Z80 kann nämlich mit einer geringfügigen Hardware-Änderung mit der Ansprache des Ports FBh einen Interrupt am HD-Pin INT2 auslösen, wodurch der HD die Buskontrolle übernimmt. Ebenso ist es denkbar, daß der HD im Interrupt-Betrieb für den Z80 mit irgendwelcher Peripherie kommuniziert. Das EPROM ist auf diese Fälle vorbereitet. Da die Vektoren von Fall zu Fall programmiert werden müssen, sind sie im EPROM alle zu FFFFh gesetzt, so daß sie entweder umgebrannt oder später soft aufgesetzt werden können.

Nur der Vektor für einen Interrupt auf den Pin INT2 des HD ist auf intz80 gesetzt, weil er anstelle des Polling die Busübergabe signalisieren könnte. Die geschähe nun freilich asynchron. Ein Job könnte gerade in Arbeit sein. Daher gibt es das Flag rdyflag, aus dem der HD ablesen kann, ob er die neue Aufgabe bereits übernehmen kann. Wenn nicht, fährt er erst mit dem alten Job fort. Wenn er später wieder bei waitz80 landet und pollt, findet er sofort RAM und tut nun verspätet, was von ihm verlangt wird. Im ausschließlichen Polling-Betrieb merkt der HD nichts von der Z80-Ansprache bevor er nach waitz80 zurückkehrt.

Schließlich wird noch der gesamte Inhalt des EPROMs nach 6.0000-7.FFFF übertragen, damit der HD später auch vom Z80-RAM aus bequemen Zugriff auf Routinen hat, die im EPROM niedergelegt sind. Ohne diese Kopie ginge das nicht, weil das Z80-RAM und das EPROM niemals gleichzeitig erreichbar sind. Solche Routinen können nachträglich ins EPROM gebrannt werden. Ihre Funktion muß nichts mit dem HD zu tun haben.

Da ab physikalisch 7.F000h bereits Common 1 definiert ist, sind die per DMA nach oben geladenen Programmteile sofort verfügbar. Deshalb können bereits die Wertetabelle und das DMA-Unterprogramm im oberen RAM benutzt werden, um das EPROM nach oben zu kopieren.

Damit hat sich das EPROM nun selber überflüssig gemacht. Dennoch bleibt es eingeblendet. Der HD liest nun fortgesetzt an der Speicherstelle 0000h und prüft, ob dort der Opcode 00h für NOP steht. Wenn ja, handelt es sich an dieser Adresse immer noch um sein EPROM. Wenn ein beliebiger anderer Code gelesen wird, liegt dort unten nun der bisherige Z80-Speicher vor. Dies ist das Signal, daß der Z80 soeben die Kontrolle über seinen Systembus an den HD abgab. Kein Betriebssystem wird bei 0000h ausgerechnet mit einem NOP beginnen, so daß Zweifel über

EPROM oder Z80-Memory nicht bestehen können. Und wenn doch, sollte der User ohnehin zuvor für den HD einen JP (Opcode C3h) zu einer Trap-Routine dort gepatcht haben. Auf Computer, die in ihrem unteren Adreßraum ein nicht schreibbares ROM haben, wird an späterer Stelle eingegangen.

Die Adresse 0000h für das Polling resultiert aus der Forderung, daß das Board mit seinem EPROM möglichst in jedem Z80-Rechner ohne Änderungen benutzbar sein sollte. In den TRS-80-kompatiblen Computern mit NEWDOS o. ä. gäbe es viele Möglichkeiten, auf eine andere Adresse auszuweichen. Jedoch schon unter CP/M fangen die Restriktionen an. Ab 0000h beansprucht CP/M den Speicher für sich. Dort steht ein JP-Befehl für den Warmstart mit CTRL-C. Unter CP/M muß deshalb eine Trap-Routine prüfen, ob es sich tatsächlich um einen TRAP handelt und gegf. den überschriebenen CP/M-Sprung nachholen. Dies gilt für viele Computer analog. Nur solche, die NEWDOS/G-DOS/TRSDOS fahren und ein Boot-EPROM besitzen, können sicher sein, daß 0000h nur bei einem TRAP angesprungen wird.

Die beiden Mikroprozessoren müssen einander mitteilen können, was anliegt. Eine Möglichkeit bestünde darin, daß der HD bei einer Ansprache durch den Z80 immer an dieselbe vereinbarte Speicherstelle springt und einfach tut, was da steht. Für diejenigen unter uns, die auch noch im Tiefschlaf vom Hacken träumen, wäre dies zweifellos die flexibelste Lösung. Ein bequemerer Weg, der sogar unter BASIC beschritten werden kann, wurde hier gewählt: An der Stelle 0003h für TRS-80 usw.: kommt später) wird der Stackpointer abgelegt. Alles Weitere, was zur Kommunikation der beiden Mikroprozessoren erforderlich ist, muß nun über den Stack abgewickelt werden. Der Stackpoin-ter des HD wird dazu aus 0003/4h geladen.

Hier gibt es folgende Vereinbarung: Zuunterst im Stack kann AF gePOPt werden. Im Akku steht dann ein Signalbyte, das dem HD verrät, welche von fünf möglichen Anforderungen der Z80 an ihn stellt (was in welcher Reihenfolge dazu auf den Stack gelegt werden muß, ist in den Listings reichlich kommentiert und teilweise tabellarisch zusammengefaßt):

1. Die endgültige Übergabe des Systembus an den HD. Wenn der Akku =00h ist, will sich der Z80 endgültig verabschieden. Der HD übernimmt nun seine Arbeit bis zum nächsten Reset oder Ausschalten. In diesem Falle wird in der nächsthöheren Ebene auf dem Stack eine Adresse erwartet, wo der HD ein paar Bytes zum Umbanken des Common- 1 -Bereiches ablegen kann. Bankt er nämlich noch oben im alten Common 1 dieses einfach weg, findet er an der entsprechenden Adresse im Bereich 0.xxxxh sonst etwas, bloß nicht viel Gescheites. Eine Ebene höher im Stack steht schließlich die Adresse, an der der HD seine Arbeit beginnen kann. Das wäre für NEWDOS/G-DOS z. B. im einfachsten Fall 402Dh, der DOS-Ready-Einsprung. Falls zusätzlich Registerinhalte aus dem Stack übergeben werden sollen, beginnt das Programm für den HD eben mit ein paar POPs.
2. Erledigung eines vorübergehenden Jobs für den Z80. Irgendein Akku-Inhalt zwischen 03h und FEh signalisiert dem HD, daß jetzt eine Einzelaufgabe für den Z80 übernommen werden soll, wonach der HD auf eine erneute Ansprache des Z80 zu warten hat. Nun erwartet der HD in der nächsten StackEbene die Adresse, wo er ins Job-Programm einspringen soll. Um nicht auch noch Informationen über die Anfangs- und Endadresse des zu bearbeitenden Programms übergeben zu müssen, wird kurzerhand der ganze Speicherinhalt von 0000-EFFFh per DMA nach oben geholt. Die Bank-Area, bisher bei 0.0000- 0.EFFFh, wird nach 7.0000-7.EFFFh verlegt. Das Z80-Programm wird nun als nterprogramm behandelt (es muß mit RET enden). Wenn es abgearbeitet ist, kehrt der HD wieder in die Warteschleife waitz80 zurück. Es besteht die Vereinbarung, daß ein solches Programm Rechenergebnisse nur im Bereich ab F800h ablegen darf, und zwar höchstens 2 kB (bis logisch FFFFh). Weiterhin ist klar, daß es ab F000h, wo der HD waltet, nichts mehr verloren hat. Es darf aber nahezu beliebig herumbanken, denn an der Rückkehradresse waitz80 werden die lebenswichtigen Systemkonfigurationen restauriert.
3. Die Abholung von Rechenergebnissen. Wenn ein solcher vorübergehender Job Ergebnisse ab F800h niedergelegt hat, könnte ein weiterer Job so programmiert sein, daß er sie von dort per DMA in den unteren Speicher transferiert. Das wäre für diese simple Aufgabe aber zu aufwendig. Deshalb gibt es die dritte Möglichkeit, daß der Akku genau =01h ist. In diesem Falle "weiß" der HD ohne weitere programmierte Absprache, daß er jetzt 2 kB Daten ab 7.F800 in einen Speicherbereich übertragen soll, der ihm eine Ebene höher im Stack mitgeteilt wird.
   Größere Datenmengen, etwa eine Graphik von 32 kB Umfang oder sehr lange Tabellen, können gleichwohl vom HD errechnet werden. Sie müssen allerdings dann wirklich in einem gesonderten Job-Programm abgeholt werden. Ihr Puffer muß weit genug unterhalb logisch F000h oder jedenfalls bei entsprechender Umsicht unterhalb physikalisch 7.F000h beginnen.
4. Die Kopie des Speichers 0.0000-0.FFFF nach 1.0000-3.FFFF. Das EPROM bietet die Möglichkeit, das untere Viertel der 41256-RAMs in die drei oberen Viertel zu kopieren. Zu diesem Zweck wird der Akku mit FFh geladen und auf den Stack gelegt. Weitere Informationen für den HD sind nicht erforderlich. Nun liegen vier identische Speicherbereiche vor, zwischen denen hin- und hergeschaltet werden kann, weil in der Nachbarbank sofort die passende Fortsetzung gefunden wird. Lediglich dem Stack ist dabei Aufmerksamkeit zu widmen, falls er nicht in einem Common-Bereich liegt und für alle Speicherviertel gültig ist. Das Unterprogramm clone kann von überall her aufgerufen werden, um beispielsweise das Konzept von vier Computern in einem Gehäuse mit der DMA-Fähigkeit des HD komfortabler zu verwirklichen.
5. Daß der Stack vom HD mit POPs und RETs gelesen wird, ist für den Z80 völlig belanglos. Sein Stackpointer geht bei der Busübergabe mit der ganzen CPU in den Ruhestand. Wenn der HD die Buskontrolle zurückgibt, zeigt der SP des Z80 an dieselbe Stelle wie vor dem OUT (FBh),A. Der User muß lediglich daran denken, den vorausgegangenen PUSHes nun die entsprechenden POPs folgen zu lassen, damit z. B. eine RET-Adresse wiedergefunden werden kann. Daß der Stack nicht ausgerechnet im Bereich ab F000h liegen darf, versteht sich. Dort hat der HD sein privates RAM, aus dem er höchstens POPpen könnte, was er selber gePUSHt hätte. Unter BASIC muß daher beispielsweise das Himem begrenzt werden.

   Bei der endgültigen Übergabe der Buskontrolle an den HD ist es so, als wäre im Rechner nie ein Z80 eingebaut gewesen. Man braucht sich um ihn nicht mehr zu kümmern. Allerdings darf der HD dann auch niemals einen Output auf den Port FBh geben, weil er damit die Kontrolle an den Z80 zurückgäbe. Prinzipiell ist das natürlich auch möglich, erfordert aber besondere Umsicht. Da der Z80 genau da weiterarbeiten würde, wo er selber mit einem OUT (FBh),A die Kontrolle abgab, sollte man an der Stelle dahinter z. B. einen Jump nach DOS-Ready oder ähnlich unkritischen Code unterbringen. Es muß auch beachtet werden, daß sich der HD zuvor wieder Common 1 in den oberen Speicher legt, damit er sich dort in sein Schneckenhaus zurückziehen kann.

   In dem unter 2. genannten Fall muß der Z80 selbstredend ein Programm übergeben, das sozusagen mit dem Gehirn des HD denkt. Wenn Common 0, die Bank-Area oder Common 1 dabei manipuliert wird, kann einiges schiefgehen. Der Programmierer kann mit der erforderlichen Vorsicht gleichwohl ohne weiteres über den EPROM-Inhalt und den gesamten Speicher ab 4.0000h verfügen sowie über die Annehmlichkeiten des DMAC, der seriellen Schnittstellen des HD usw. Es ist sogar möglich, nach diesem Muster einen BASIC-Job zu übergeben (wer wollte ernsthaft eine Wurzel in Maschinensprache ziehen?). Da der gesamte Speicher von 0.0005-0.EFFF nach oben verholt wird, darf selbstverständlich auch BASIC/CMD oder auch nur Level 2 dort am Werk sein.

Es dürfen auch Illegals im Programm enthalten sein, denn Geralds Trap-Routine liegt bereit. Dieser Fall ist jedoch nicht vorgesehen. Dazu müßte an der Stelle logisch 0000h ein JP zur Trap-Routine gesetzt werden. Es wäre aber nicht besonders klug, weil die Umschreibung eines Illegals mit zwei, drei Befehlen um Größenordnungen schneller geht als ein Trap.

Im Falle der Übergabe von Rechenergebnissen muß von der Z80-Seite aus nichts weiter bedacht werden. Es versteht sich lediglich, daß von der auf dem Stack abgelegten Adresse an wirklich mindestens 2 kB RAM frei sein müssen. In beiden letztgenannten Fällen knipst der HD im unteren Adreßraum mit OUT (FBh),A sein EPROM wieder an, sobald er das zu bearbeitende Programm nach oben oder die Resultate eines Programms nach unten verfrachtet hat. Der Z80 hat im unteren RAM nun wieder das Sagen.

In allen Fällen muß der Programmierer wissen, daß der HD noch längst nicht auf die Hardware eingestellt ist, in der er die Arbeit übernehmen soll. Bei den Tandy-Kompatiblen ist z. B. der Befehl für den Interrupt-Modus IM 1 ins HD-Programm zu schreiben. Da bei der Busübernahme DI durchgeführt wurde, müssen die Interrupts auch bei Bedarf gesondert wieder enabled werden. Überhaupt kann der Z80 in keinerlei Hinsicht von sich auf andere schließen (Registerinhalte, INT-Behandlung), so daß vor der Busübergabe auf der Z80-Seite sehr sorgfältig an alles Wesentliche gedacht werden muß.

Um das EPROM im Interesse einer allgemeinen Verwendbarkeit auch nicht auf das Genie 3s mit Helmuts Megabyte-Banker maßzuschneidern, darf der Z80 den HD 64180 nur ansprechen, wenn er gerde im unteren Viertel der 256er RAM-Chips arbeitet. Das ist der "Computer Nr. 0", wie wir ihn in dem Sonderheft "Dein G3s, die 4 unbekannten Wesen" genannt haben. Bei Computern mit 64 kB RAM entfällt diese Bedingung.

Dies sind die Möglichkeiten, die der HD sich mit seinem EPROM selber schafft. Wenn er jedoch endgültig oder vorübergehend die Kontrolle über den Systembus übernimmt, muß ihm der Z80 zuvor Quartier gemacht haben. Wegen der möglicherweise auftretenden TRAP-Interrupts sollte an 0000h in eine Error-Trap-Routine verzweigt werden. Wo die Trap-Routine dann steht, ist gleichgültig. Es muß auch nicht die komplette Routine sein, wie sie bereits im EPROM programmiert ist. Da sie nach dem DMA bei den Initialisierungen nach 7.F100 übertragen wurde, wo sie noch immer steht, braucht der Z80-Beitrag zum TRAP nur aus dem entsprechenden Banking und einem Sprung nach dort oben zu bestehen.

Die älteren TRS-80-kompatiblen Rechner lesen von 0000-2FFFh in einem ROM und haben erst ab 4000h freies RAM. Wenn diese Computer einen Banker eingebaut haben, der im unteren Adreßraum RAM einblenden kann, gilt alles bisher Gesagte unverändert. Im RAM muß dann eine Kopie des ROMs vorliegen, die an 0000h aber den Trap-Sprung enthält. Aber ohne Banker entfällt die Möglichkeit, unter der Regie des HD Programme mit Illegals zu fahren, da wegen der nicht änderbaren RST-00h-Routine ein TRAP nicht möglich ist. Solche Programme würden zu einem Boot führen. Am Anfang des Listings muß dann auch spbuff auf eine RAM-Adresse gesetzt werden, z. B. 4200h (DOS-Sektorpuffer).

Im Folgenden soll nun ein Beispiel gegeben werden, wie der Z80 lästige Pflichten loswerden kann, um sich mittlerweile wichtigeren Dingen zu widmen. Das Beispiel besteht hier nur aus dem Löschen des Bildschirms. Eine läppische Sache, die diese digitale Orgie mit einem zweiten Mikroprozessor gar nicht lohnt. Es geht jedoch nur darum, zu zeigen, wie beide CPUs miteinander kommunizieren können. Das Programm ist im Listing 2 wiedergegeben. Der Z80 übergibt im Stack zuerst die Einsprungsadresse des Programms namens job, dann das Signalbyte, aus dem der HD entnehmen soll, daß es nur eine vorübergehende Aufgabe ist. Schließlich wird der Stackpointer an 0003/4h abgelegt.

Jetzt hat der Z80 die Hände frei, um Denkwürdiges zu leisten, während der HD inzwischen 2 kB Pufferraum mit Leerzeichen vollmalt. Wenn der Z80 der Meinung ist, es sei nun an der Zeit, den Bildschirm wirklich zu löschen, holt er sich die 2 kB Blanks vom HD ab. Das geschieht analog zur obigen Prozedur: Auf den Stack kommt zunächst die Pufferadresse, hier der Bildschirm, dann das Signalbyte 01h, damit der HD seinen Puffer hinunterkopiert. Es wäre wesentlich simpler gegangen. Das Job-Programm hätte selbständig per DMA auf den Bildschirm zugreifen können, um ihn um Größenordnungen schneller als gewohnt zu löschen. Wie der User mit den vielfältigen Möglichkeiten jongliert, die ihm das EPROM offenläßt, bleibt ihm schließlich selbst überlassen.

Als weiteres Beispiel wird gern. Listing 3 die Buskontrolle endgültig an den HD 64180 übergeben. Das Signalbyte am unteren Ende des Stack ist 00h. Darüber liegt die Adresse, wo der HD sein kleines Banking-Programm switch ablegen soll. Darüber schließlich findet der HD den Einsprung in seine künftigen Pflichten. Dort wird zunächst aus ihm sozusagen ein Z80 gemacht, damit er mit G-DOS klarkommt. Anwenderprogramme, die er später bearbeiten wird, können mit dem ganzen Spektrum seiner Leistungsfähigkeit arbeiten.

Es gibt eine fünfte Möglichkeit, wie sich der Z80 des HD bedienen kann. Sie unterscheidet sich von den bisher Genannten so grundlegend, daß sie erst jetzt besprochen werden soll. Nachdem die Methode Stack-orientierter Kommunikation vorn Leser verdaut wurde, ist nun hoffentlich nicht mehr mit Konfusion zu rechnen.

6. Der Vektor-Task. Bisher mußte das Byte (SP+0) als LSB eines gePUSHten 16-Bit-Worts nicht beachtet werden. Es beginnt, eine Rolle zu spielen, wenn der Inhalt von (SP+1) (gePUSHtes MSB) 02h beträgt. Dies ist das Signalbyte, um eine beliebige Routine zu bearbeiten, deren Adresse man nicht einmal kennen muß. Das Register C enthält nach POP BC die laufende Nummer der gewünschten Routine, von denen es 128 geben darf. Wird beispielsweise gewünscht, daß das Programm Nr. 9 (ab 0 gezählt) läuft, dann wird ein 16Bit-Register vor der Busübergabe mit 0209h geladen und gePUSHt. 02h ist das Signalbyte für diese Art der Ansprache des HD durch den Z80.

Wo dieses Programm zu suchen ist, steht in einer Tabelle. Die Zahl im Register C wird zunächst verdoppelt, weil jeder Vektor (Zeiger) eine Zwei-Byte-Adresse enthält. Nachdem das MSB auf 00h gesetzt ist, steht nun in BC der Summand zur Basisadresse einer Vektortabelle. BC plus Basis ist dann der Zeiger auf die Adresse, an der das gewünschte Programm steht. Das ist dasselbe Prinzip, das bei Interrupts des Modus IM 2 angewendet wird. Ein solches Programm ist in seiner Flexibilität etwa der endgültigen Busübergabe vergleichbar, kehrt aber immer nach waitz80 zurück. Listing 7 zeigt ein Anwendungsbeispiel.

Mit Listing 4 soll gezeigt werden, wie die Trap-Routine des EPROMs vom Z80-RAM aus aufgerufen werden kann. Das Programm wurde mit dem Genie 3s erstellt, dürfte aber wohl auf allen TRS-80-kompatiblen Computern laufen, die kein ROM an den unteren Adressen haben. Andernfalls muß nach einem anderen Speicherabschnitt für die Routine trap Ausschau gehalten werden. Er darf auf keinen Fall die logische Adresse F000h überschreiten.

Die Routine liest die MMU-Register BBR und CBAR aus und puffert ihren Inhalt, damit er später wieder restauriert werden kann. Dann wird Common 1 wie bei Reset definiert (ab 7.F000h), damit die Trap-Routine unter der logischen Adresse F100h gefunden wird. Da (z. B. unter BASIC) der Stack im Himem liegen könnte, darf bei der Bankerei nichts gePUSHt werden; alles wird über Speicherladebefehle abgewickelt.

Dieses Programm stellt die Voraussetzung, daß nur im User-RAM ein TRAP auftreten wird. Das DOS, das auch in seiner eigenen Bank werkelt, benutzt keine Illegals, so daß von dort nichts zu befürchten ist. Der Anwender muß sich eben damit abfinden, daß er bei der Arbeit in einer anderen Bank nur die dokumentierten Befehle verwenden darf. Die ziemlich lendenlahme TRAP-Prozedur legt ohnehin nahe, Illegals durch andere Befehle zu ersetzen.

Es fällt auch auf, daß das TRAP-Bit nicht gelesen wird. Das Programm setzt einfach voraus, daß ein RST 00h nur bei Trap auftritt, denn bei Reset wird ja im Boot-EPROM (neuerdings auch der Coprozessorkarte) gelesen. Auch bei ROM-Computern mit nachträglich eingebautem Banker ist das so. Ein Reset-Taster existiert dort nicht, und nach dem Einschalten ist sowieso das ROM selektiert. Der angebliche RESET dieser Maschinen ist in Wirklichkeit ein NMI und springt bei 0066h ein. Dort steht nach wie vor der alte Code.

Aber unter CP/M wäre wegen des RST 00h bei Warmstart mit CTRL-C die Prüfung auf TRAP unumgänglich. CP/M stellt ebenfalls die Bedingung, unbedingt die Interrupts zu maskieren, weil es im Himem arbeitet, wo nach dem Umschalten nun eine andere Bank vorliegt. Die oben beschriebene Akrobatik zum Ermitteln des EI/DI-Zustandes braucht dabei aber nicht stattzufinden. Es geht auch so, wie im Listing 5 beschrieben. Der intern belegte Port 34h enthält nämlich nicht nur die Bits UFO und TRAP (Bits 6 und 7), sondern auch ITEO-2 (Bits 0, 1, 2). Wenn diese Bits 0 lauten, werden Interrupts an den korrespondierenden INT-Eingängen der CPU ignoriert. Der Input von 34h kann also an den Ausgang geladen, dann manipuliert und chließlich am Ausgang wieder restauriert werden.

Der HD 64180 bietet die Möglichkeit, eine programmierbare Anzahl von WAIT-Zyklen für Speicher- und I/O-Operationen getrennt zu generieren. Man möchte natürlich seine Maschine so schnell wie möglich arbeiten lassen, also so wenige WAITs einfügen, daß das System noch eben zuverlässig arbeitet. Im Listing 6 wird gezeigt, wie man diese Grenze herausfinden kann. Ein Job-Programm verändert in der EPROM-Tabelle initab das Byte für den Port 32h. Es bestimmt die Anzahl der WAITs, und zwar für den Speicher in den Bits 6-7 und für Portansprachen in den Bits 4-5. Nach der Bearbeitung des Jobs wird die Buskontrolle endgültig an den HD 64180 übergeben. Wenn das System mit diesem Programm abstürzt oder eine I/O-Operation nicht das erwartete Ergebnis bringt, war der Akku zu sparsam beladen.

Durch wiederholten Aufruf dieses Programms mit variiertem Wert für den Akku wird experimentell bestimmt, wie schnell höchstens mit der eigenen Hardware gearbeitet werden kann. Der so gefundene Wert sollte dann immer bei der ersten Ansprache des HD nach einem Reset gern. dem Muster in Listing 6 an den Anfang der Tabelle initab gesetzt werden. Mit korrekt geladenem Akku kann dies auch als Programm zur endgültigen Busübergabe genommen werden. Abgesehen vom Signalbyte für den HD und den Adressen, die im Stack übergeben werden, braucht dem Stack dabei keine weitere Aufmerksamkeit gewidmet zu werden, weil der Stackpointer beim Sprung nach DOS-Ready neu geladen wird.

Alle sieben Listings enthalten bereits die Mnemonics einiger zusätzlicher Befehle, die der HD 64180 kennt. Da ZEUS sie jedoch noch nicht kennt, stehen sie wie Kommentare hinter einem Semikolon. Um ZEUS begreiflich zu machen, was auf die Diskette assembliert werden soll, sind die Hex-Entsprechungen dieser Befehle darunter immer als DBs programmiert.

Kein Programm wird jemals fertig. Der Inhalt dieses EPROMs wird sich noch ändern, besonders in den ersten Monaten der Erprobung in der Praxis. Das Listing 1 ist deshalb vielleicht (oder eher wahrscheinlich) zum Zeitpunkt seiner Veröffentlichung bereits überholt. Wer das EPROM für seine eigene Coprozessorkarte haben möchte, erhält die jeweils neueste Version. Ein aktualisiertes Listing wird nicht mitgeliefert. Sein Inhalt kann mit einem Job-Programm in den unteren Speicher kopiert und dort leicht disassembliert werden.

Von den möglichen 64 kB des EPROMs ist mit bisher gerade einem Kilobyte nur ein Bruchteil für die Grundintelligenz der Coprozessorkarte verbraucht. Viele nützliche Routinen, z. B. ein Spooler mit enormem Pufferplatz oder eine intelligente RAM-Floppy, finden hier noch Platz. Zu diesem Zweck sind bestimmte Adressen mit FFh-Bytes freigehalten worden. FFh kann bei einem bereits gebrannten EPROM nachgebrannt werden. So sind beispielsweise noch sämtliche Restart-Adressen frei (außer RST 00h, versteht sich). Auch Füllbytes, die nachfolgende Routinen oder Tabellen auf "gerade" Adressen schieben sollen, lauten FFh und können noch sinnvoll programmiert werden, ebenso die Tabelle der Vektor-Utilities. Dasselbe gilt für den NMI-Einsprung an 0066h, obwohl die Karte nicht für einen NMI im EPROM (sehr wohl im RAM) vorgesehen ist. Aber irgendein Hard-Freak unter euch könnte sich das nachrüsten wollen.

```
                ; Listing 1: EPROM-Inhalt der Coprozessorkarte

                  *****************************************************
                  Boot-EPROM für die Coprozessorkarte
                  mit dem HD 64180

                  für das Genie 3s und die meisten anderen Z80-Rechner

                  von Helmut Bernhardt

                  (C) 1987 Arnulf Sopp

                  die Trap-Routine entspricht weitestgehend der Vorlage
                  von Gerald Schröder     (C) 1987
                ;******************************************************

                ;Puffer für den Stackpointer des Z80 bei Übergabe der Buskontr. an den HD
                ;die Adresse gilt für Computer mit einem Boot-EPROM und RAM ab 0000
0003            spbuff    EQU     0003h           ;für ROM-Computer: auf freie RAM-Adresse ändern

```

```
0000                      ORG     0000h           ;physikalisch 0.0000
                ;RST 00: nach Reset die MMU des HD 64180 initialisieren usw.
0000 00                   NOP                     ;wegen des Pollings an logisch 0000
0001 ED5E                 IM      2               ;wegen mögl. Ansprache des Z80 über INT2
0003 214501               LD      HL,initab+2     ;Tabelle der internen Port-Initialwerte
0006 1833                 JR      gorst00         ;dort weiter
                ;weitere RST-Routinen können nachgerüstet werden
0008 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;Platzhalter für RST 08
0010 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 10
0018 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 18
0020 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 20
0028 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 28
0030 FFFF                 DW      Offffh,Offffh,Offffh,Offffh     ;dto. RST 30
0038 FF                   DB      Offh,Offh,Offh                  ;dto. RST 38 (JP genügt)
                ;Fortsetzung der RST 00-Routine (Reset oder Trag)
0038 01340C     gorst00   LD      BC,0c34h                ;12 Werte ab Port 34h
                ;         OTIMR                           ;Initialwerte für interne
003E ED                   DB      0edh,93h                ;Ports aufsetzen (s. o.)
                ;Warteprogramm und EPROM-Inhalt per DMA ins Himem transferieren
0040 219800               LD      HL,tabend       ;Ende der Wertetabelle für den DMAC
0043 CD8700               CALL    dma2            ;per DMA Himem-Programm hochtransferieren
0046 68                   LD      L,B             ;HL (- 0000 (Anfang EPROM, H=00, BC=0000)
0047 50                   LD      D,B             ;DE (- 0000 (LSW logische Zieladresse)
0048 58                   LD      E,8
0049 3E60                 LD      A,60h           ;Quellber. 0.xxxx, Zielber. 6.xxxx (MSB)
0048 CD6900               CALL    dma0            ;EPROM-Inhalt nach 6.0000-6.FFFF übertr.
004E C33CF0               JP      inidone+offs    ;Warteroutine anspringen
                ;Platzhalter bis zum NMI-Einsprung an 0066
0051 FFFF                 DW      Offffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh
0061 FF                   DB      Offh,Offh,Offh,Offh,Offh
                ;Einsprung für NMI (nicht vorgesehen, kann aber nachgerüstet werden)
0066 FF         nmi       DB      Offh,Offh,Offh  ;3 Bytes, können mit JP überschr. werden
F000            himem     EQU     0f000h          :Common 1, Arbeitsadresse des Programms
EF97            offs      EQU     himem-$         ;Offset zur Arbeitsadresse des Programms
                ;der folgende Teil des EPROM-Inhalts arbeitet im Speicher ab 7.F000
                ;Unterprogr. für DMA: HL = LSW Quelle, DE = LSW Ziel, BC = Bytezähler,
                ;A Bit 0-2 = MSB der Quelle, A Bit 4-6 = MSB des Ziels
0069 222BF0     dma0      LD      (srclog+offs),HL        ;Quellbereich   LSW     x.HL
006C ED532EF0             LD      (dstlog+offs),DE        ;Zielbereich    LSW     x.DE
0070 ED4331F0             LD      (dmacnt+offs),BC        ;Bytezähler für DMA aufsetzen
0074 F5                   PUSH    AF                      ;phys. Quell- und Ziel-MSB retten
0075 E60F                 AND     0fh                     ;Quellbits maskieren
0077 322DF0               LD      (srcphys+offs),A        ;Quellbereich   phys.   A.HL
007A F1                   POP     AF                      ;MSB des Quell- und Zielblocks
0078 OF                   RRCA                            ;in die unteren Bits rotieren
007C OF                   RRCA
007D OF                   RRCA
007E OF                   RRCA
007F E60F                 AND     Ofh                     ;Zielbits maskieren
0081 3230F0               LD      (dstphys+offs),A        ;Zielbereich    phys.   A.DE
                ;Einsprung in DMA, falls die Tabelle der Adressen vorbereitet ist 0084
0084 2132F0     dma1      LD      HL,tabend+offs          ;Beginn der DMA-Wertetabelle
                ;Bank-unabhängiger Teil des Programms, kann von überallher gecallt werden
0087 012708     dma2      LD      BC,0827h                ;8 Werte ab Port 27 abwärts
                ;         OTDMR                           ;DMAC aufsetzen
008A ED                   DB      0edh,9bh                ;(s. o.)
008C 013102               LD      BC,0231h                ;2 Werte ab Port 31 abwärts
                ;         OTDMR                           ;Rest aufsetzen u. DMA abschießen
008F ED                   DB      0edh,9bh                ;(s. o.)
0091 C9                   RET
                ;Init.-Werte des DMAC zur Kopie des Himem-Programms ins RAM ab 7.F000 0092
```

```
0092 41                      DB      41h     ;30.    DSTAT:  Kanal 0, Zustand 'scharf'
0093 02                      DB      02h     ;31,    DMODE:  Speicher zu Speicher, Burst-Modus
0094 6900          srclog    DW      dma0    :20/1, SAROL/H: Quelladresse    LSW     dma0
0096 00            srcphys   DB      00h     ;22,    SAROB:  MSB     0.xxxx
0097 00F0          dstlog    DW      himem   ;23/4, DAROL/H: Zieladresse     LSW     F000
0099 07            dstphys   DB      07h     ;25,    DAROB:  MSB     7.xxxx
009A 0403          dmacnt    DW prt-dma0     ;26/7, BCROL/H: Bytezähler für die Programmlänge
0098               tabend    EQU     S-1     ;Ende der Tab.. womit der Zeiger HL geladen wird

                   ;nach der Initialis. auf Ansprache des Z80 warten, sie interpretieren
009C 00            rdyflag   DB      00h             ;Flag für noch nicht beendeten Job
009D 21DAF0        waitz80   LD      HL,initab+offs  ;Tabelle der internen Port-Initialwerte
00A0 01320E                  LD      BC,0e32h        ;14 Werte ab Port 32h
                   ;         OTIMR                   ;Initialwerte für interne Ports aufsetzen
00A3 ED                      DB      0edh,93h        ;(s. o.)
00A5 3EF3          inidone   LD      A,inttab+offs/256       ;MSB der Interrupt-Vektortabelle
00A7 ED47                    LD      I,A             ;ins Interrupt-Vektorregister
00A9 D3FB          copdone   OUT     (0fbh),A        ;falls der HD noch die Buskontrolle hat
00AB 3100F1                  LD      SP,trap+offs    ;Stack im freien Bereich
00AE 210000                  LD      HL,0000h        ;Prüfadresse (im Z80-RAM unkritisch)
0081 AF              XOR     A                       ;A (- 00, Flag für beendeten Job
0082 3233F0                  LD      (rdyflag+offs),A        ;Bereitsch. des HD dort vermerken
0085 FB                      EI                      ;für evtl. Z80-Interrupt
0086 7E            polloop   LD      A,(HL)          ;Testbyte laden
0067 B7                      OR      A               ;immer noch 00? (EPROM noch eingeblendet)
0088 28FC                    JR      Z,polloop       ;falls noch EPROM

                   ;der Z80 gab soeben die Kontrolle an den HD ab (auch Entry für Z80-INT)
008A F3            intz80 DI ;wegen unklaren Interrupt-Modus usw.
0088 2A33F0                  LD      HL,(rdyflag+offs)       ;Flag für anhängigen Job
008E 7E                      LD      A,(HL)          ;Flag laden
00BF 67                      OR      A               ;ist z. Zt. noch ein Job in Arbeit?
0000 2802                    JR      Z,nojob         ;falls nicht mehr
0002 ED4D                    RETI                    ;falls ja (nur bei INT vom Z80)
00C4 ED780300      nojob     LD      SP,(spbuff)     ;Stackpointer des Z80 übernehmen

                   ;vereinbarte Vorgaben für den Stack des Z80:
                   ; (SP+0): evtl. Wegweiser in die Utility-Vektortabelle
                   ; (SP+1): Signalbyte für die Art der übergebenen Aufgabe
                   ;
                   ;bei endgültiger Übernahme:
                   ; (SP+2), (SP+3): Platz für den Befehlsstring zum Banking
                   ; (SP+4), (SP+5): Startadresse (Entrypoint) für den HD 64180
                   : evtl. darüber : Registerinhalte, die gePOPt werden können
                   ;
                   ;bei vorübergehendem Job:
                   ; (SP+2), (SP+3): Startadresse (Entrypoint) für den HD 64180
                   ;
                   ;bei Übergabe von Rechenergebnissen:
                   ; (SP+2), (SP+3): Pufferadresse im Z80-RAM
                   ;
                   ;bei Speicherkopie von 0.0000-0.FFFF nach 1.0000-3.FFFF im Z80-RAM
                   ;und bei Abruf einer programmierten Utility aus der Vektortabelle:
                   ; keine weiteren Daten
                   ;
                   ;Vereinbarung für das Signalbyte (SP+0) (Register B nach POP BC)
                   ; B:00:   endgültige Übernahme des Systembusses durch den HD 64180
                   ; B=01:   2 kB Ergebnisse eines früheren Jobs aus 7.F800 übergeben
                   ; B=02:   eine Utility aus der Vektortabelle aufrufen
                   ; B=FF:   0.0000-0.FFFF in die 64-kB-Blocks bis 3.FFFF kopieren
                   ; FF>8>02: einen sonstigen vorübergehenden Job vom Z80 übernehmen

0008 C1                      POP     BC              ;unterste Stack-Ebene lesen
00C9 78                      LD      A,B             ;Task-Signalbyte vom Z80-Stack
00CA 77                      LD      (HL),A          ;Flag für anhängenden Job setzen
00CB 3C                      INC     A               ;A=FF, das RAM 0.0000-0.FFFF kopieren?
00CC 2840                    JR      Z,copy          ;falls ja
00CE 3D                      DEC     A               ;A=00, Kontrolle endgültig übernehmen?
00CF 200F                    JR      NZ,job          ;>00: Job übernehmen o. Ergebn. übergeben

                   ;der Z80 hat im Stack signalisiert, daß er die Kontrolle ganz abgibt
```

```
00D1 2173F0               LD      HL,switch+offs  ;Befehlsfolge zum Umschalten von Common 1
00D4 D1                   POP     DE              ;Zieladr. des Umschaltbefehls a. d. Stack
00D5 D5                   PUSH    DE              ;noch einmal auf den Stack als RET-Adr.
0006 010400               LD      BC,0004h        ;Länge der beiden Befehle
00D9 EDBO                 LDIR                    ;übertragen
0008 C9                   RET                     ;banken und Arbeit des Z80 übernehmen

                switch
                ;         OUT0    (38h),A         ;A=00, Common 1 ab 0.0000 einstellen
00DC ED                   DB      0edh,39h,38h    ;Hex-Entsprechung des Befehls
00DF C9                   RET                     ;an die vom Z80 vorgegebene Adr. springen

                ;der Z80 hat im Stack signalisiert, daß er für den HD 64180 einen Job hat
                ;er muß mit RET aufhören (wird wie ein Unterprogramm behandelt)
00E0 E1         job       POP     HL              ;Einsprungsadresse für den Job
00E1 3100F1               LD      SP,trap+offs    ;Stack wieder im eigenen RAM
00E4 E5                   PUSH    HL              ;Entry des Jobs auf den eigenen Stack
00E5 2134F0               LD      HL,waitz80+offs ;Rücksprungadresse
00E8 E3                   EX      (SP),HL         ;als RET-Adresse a. d. Steck, HL <- Entry
00E9 E5                   PUSH    HL              ;Programm-Entry retten
00EA 3D                   DEC     A               :A=01, Ergebnisse hinunterladen?
00E8 2816                 JR      Z,putres1       ;falls ja
00ED 3D                   DEC     A               ;A=02, eine Utility aus d. Vektortabelle?
00EE 2823                 JR      Z,vectask       ;falls ja
00F0 210000               LD      HL,0000h        ;Quelladresse für DMA x.0000 (LSW)
00F3 54                   LD      D,H             ;Zieladresse dto.
00F4 5D                   LD      E,L
00F5 0100F0               LD      BC,himem        ;Größe des zu übertrag. Speicherblocks
00F8 3E70                 LD      A,70h           ;Quelle phys. 0.xxxx, Ziel 7.xxxx (MSB)
                ;         OUT0    (39h),A         ;Bank-Area ebenfalls ab 7.0000
00FA ED                   DB      0edh,39h,39h    ;(s. 0.)
00FD CD00F0     dmaresl   CALL    dma0+offs       ;Speicher hoch- oder runterkopieren
0100 D3FB       vectret   OUT     (0fbh),A        ;dem Z80 die Buskontrolle zurückgeben
0102 C9                   RET                     ;evtl. Programm anspr., dann nach waitz80

                ;der Z80 hat signalisiert, daß er 2 kB Resultate aus 7.F800 abholen will 0103 D1
                putresl   POP     DE              ;für Ergebnisse: DE = Zielbereich
0104 2100F8               LD      HL,0f800h       ;Puffer für Job-Ergebnisse (Quellbereich)
0107 010008               LD      BC,0800h        ;Länge des Puffers immer 2 kB
010A 3E07                 LD      A,07h           ;Quellbereich 7.xxxx, Zielbereich 0.xxxx
010C 18EF                 JR      dmaresl         ;Ergebnisse übertragen, zurück zu waitz80

                ;der Z80 hat signalisiert, 0.0000-0.FFFF nach 1.0000-3.FFFF zu kopieren
010E CDBAF0     COPY      CALL    clone+offs      ;den Speicher kopieren
0111 1896                 JR      copdone         ;zurück in die Warteschleife

                ;der Z80 ruft einen Task auf, dessen Adresse in der Vektortabelle steht
0113 2197F0     vectask   LD      HL,vectret+offs ;Rückkehradresse für diesen Fall
0116 E3                   EX      (SP),HL         ;auf den Stack, Nonsense-Adresse löschen
0117 47                   LD      B,A             ;8 als MSB des Vektorsummanden =00
0118 CB01                 RLC     C               ;LSB verdoppeln wegen 2 Bytes pro Vektor
011A 2100F2               LD      HL,vectab+offs  ;Beginn der Vektortabelle
011D 09                   ADD     HL,BC           ;Zeiger auf die Task-Adresse stellen
011E 7E                   LD      A,(HL)          ;das LSB der Adresse laden
011E 23                   INC     HL              ;auf das MSB stellen
0120 66                   LD      H,(HL)          ;dieses laden
0121 6F                   LD      L,A             ;HL - Adresse der Utility
0122 E9                   JP      (HL)            ;sie anspringen

                ;UP zur Speicherkopie, das auch unabhängig von A=FF gecallt werden kann
                ;je nach gewünschter Blockgröße, Startbank, Zahl der Banks, Adressen usw.
                ;Einsprung mit unterschiedl. Registerinhalten an verschiedenen Stellen
0123 010000     clone     LD      BC,0000h        ;Bytezähler 0000 : 1.0000 (64 kB)
0126 60                   LD      H,B             ;Quelladresse LSW 0000
0127 69                   LD      L,C
0128 50                   LD      D,8             ;Zieladresse LSW 0000
0129 59                   LD      E,C
012A 3E10                 LD      A,10h           ;Quelle phys. 0.0000, Ziel phys. 1.0000
0120 ED43CDF0             LD      (cloncnt+offs),BC       ;den Zähler aufsetzen
0130 0603                 LD      B,03h           ;3 Banks zu je 64 kB
0132 C5         coploop   PUSH    BC              ;Schleifenzähler retten
```

```
0133 E5         00234           PUSH    HL               ;dto. LSW Quelladresse
0134 F5         00235           PUSH    AF               ;dto. MSB Quell- und Zieladresse
0135 010000     00236           LD      BC,0000h         ;Bytezähler 0000 (:1.0000)
0136            00237 cloncnt   EQU     $-2              ;kann auf andere Größe geändert werden
0138 CD00F0     00238           CALL    dma0+offs        ;eine Bank kopieren
0138 F1         00239           POP     AF               ;MSB Quell- und Zieladresse
013C C610       00240           ADD     A,10h            ;der nächste phys. 64-kB-Block
013E E1         00241           POP     HL               ;LSW Quelladresse
013F C1         00242           POP     BC               ;Schleifenzähler
0140 10F0       00243           DJNZ    coploop          ;bis 3 64-kB-Blocks kopiert sind
0142 C9         00244           RET
                00245
                00246 ;Initialwerte der internen Ports 32-3F (nach RESET erst ab 34 gelesen)
0143 F0         00247 initab    DB      0f0h             ;32, DCNTL: 3 WAITs für Memory und I/O
0144 00         00248           DB      00h              ;33, IL: interne INTs ab 00 in der Tabelle
0145 07         00249           DB      07h              ;34, ITC: alle ext. Interrupts zugelassen
0146 FF         00250           DB      0ffh             ;35: belanglos
0147 83         00251           DB      83h              ;36, RCR: Refreshes möglichst zeitsparend
0148 FF         00252           DB      0ffh             ;37: belanglos
0149 70         00253           DB      70h              ;38, CBR: Common 1 ab 7.x000
014A 00         00254           DB      00h              ;39, BBR: Bank-Area ab  0.x000
014B F0         00255           DB      0f0h             ;3A, CBAR: Common 0 nicht definiert
                00256                                    ;Bits 0-3: Bank-Area: 0.0000 - 0.EFFF
                00257                                    ;Bits 4-7: Common 1 : 7.F000 - 7.FFFF
014C FF         00258           DB      0ffh             ;3B:    belanglos
014D FF         00259           DB      0ffh             ;3C:
014E FF         00260           DB      0ffh             ;3D:
014F FF         00261           DB      0ffh             ;3E:
0150 20         00262           DB      20h              ;3F, ICR: interne Ports ab 00, IOSTOP-Modus
                00263
                00264 ;Füllbytes, um trap0 im Himem auf die "gerade' Adresse F100 zu bekommen
                00265 ;hier liegt auch der HD-eigene Stack (max. 12 Ebenen!)
0151    FFFF    00266           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0161    FFFF    00267           DW      0ffffh,0ffffh,0ffffh,0ffffh
                00268
                00269 ;Fehlerbehandlung bzw. Illegal-Simulation nach einem TRAP-Interrupt
0169 F5         00270 trap      PUSH    AF               ;Akku und Flags retten
                00271 ;         IN0     A,(34h)          ;TRAP-Bit und UF0-Bit einlesen
016A ED         00272           DB      0edh,38h,34h     ;(s. o.)
016D E67F       00273           AND     7fh              ;TRAP-Bit zurücksetzen
                00274 ;         OUT0    (34h),A          ;TRAP löschen
016F ED         00275           DB      0edh,39h,34h     ;(s. o.)
0172 3211F1     00276           LD      (ufo+offs),A     ;UFO retten
0175 F1         00277           POP     AF               ;Akku vom Stack holen
0176 E3         00278           EX      (SP),HL          ;HL mit Trapadresse tauschen
0177 F5         00279           PUSH    AF               ;Akku wieder retten
0178 C5         00280           PUSH    BC               ;retten
0179 3E00       00281           LD      A,00h            ;UFO (Dummy-Operand)
017A            00282 ufo       EQU     $-1              ;je nach dem
0178 CB77       00283           BIT     6,A              ;UF0-Bit gesetzt?
017D 2801       00284           JR      Z,nosub          ;nein, Opcode direkt davor
017F 28         00285           DEC     HL               ;sonst Zeiger -1
0180 2B         00286 nosub     DEC     HL               ;Zeiger auf illegalen 0pcode
0181 7E         00287           LD      A,(HL)           ;Opcode laden
0182 47         00288           LD      B,A              ;und retten
0183 FECB       00289           CP      0cbh             ;CBxx? (SLIA bzw. SLL)
0185 23         00290           INC     HL               ;Zeiger weiterstellen
0186 7E         00291           LD      A,(HL)           ;2. Byte laden
0187 200D       00292           JR      NZ,ddfd          ;falls 0perand mit IX/IY
0189 D620       00293           SUB     20h              ;Akku korrigieren
0186 3228F1     00294           LD      (cbxx+offs),A    ;in den Programmtext laden
018E C1         00295           POP     BC               ;Register restaurieren
018F F1         00296           POP     AF
0190 23         00297           INC     HL               ;PC neu setzen
0191 E3         00298           EX      (SP),HL          ;als RET-Adresse auf den Stack
0192 37         00299           SCF                      ;Cy - 1, Bit 0 bei SLIA/SLL immer 11
0193 CB10       00300           RL      B                ;rotieren (8 ist Dummy-Operand)
0194            00301 cbxx      EOU     S-1              ;0perand für Register B-A
0195 C9         00302           RET
                00303
0196 FECB       00304 ddfd      CP      0cbh             ;ein ähnlicher Befehl für IX/IY?
```

```
0198 2827       00305           JR      Z,ddfdcb        ;falls ja
019A 3252F1     00306           LD      (substit+offs),A        ;nächsten 0pc. in d. Programmtext
019D FE26       00307           CP      26h             ;LD HX/HY,const ?
019F 2807       00308           JR      Z,ldcon         ;falls ja
01A1 FEZE       00309           CP      2eh             ;LD LX/LY,const ?
01A3 2803       00310           JR      Z,ldcon         ;falls ja
01A5 AF         00311           XOR     A               ;keine Konstante: NOP
01A6 1802       00312           JR      gotrap          ;dort weiter
01A8 23         00313 ldcon     INC     HL              :Zeiger auf die Konstante
01A9 7E         00314           LD      A,(HL)          ;diese laden
01AA 3253F1     00315 gotrap    LD      (const+offs),A  ;evtl. Konstante in den Programmtext
01AD 78         00316           LD      A,B             ;DD/FD zurück
01AE 324FF1     00317           LD      (ddfd1+offs),A  ;in den Programmtext setzen
0181 3255F1     00318           LD      (ddfd2+offs),A  ;dort auch
0164 C1         00319           POP     BC              ;Register restaurieren
0185 F1         00320           POP     AF
0186 23         00321           INC     HL              ;PC neu setzen
0167 E3         00322           EX      (SP),HL         ;als RET-Adresse auf den Stack
0188 DDE5       00323 ddfd1     PUSH    IX              ;bzw. IY
018A E3         00324           EX      (SP),HL         ;HL - IX/IY
0168 7F         00325 substit   LD      A,A             ;Operation ausführen (hier Dummy)
016C 00         00326 const     DB      00h             ;evtl. Konstante
01BD E3         00327           EX      (SP),HL         ;neues IX/IY zurück
018E DDE1       00328 ddfd2     POP     IX              ;bzw. IY
01C0 C9         00329           RET
                00330
01C1 78         00331 ddfdcb    LD      A,B             ;DD/FD zurück
01C2 32A2F1     00332           LD      (ddfd3+offs),A  ;in den Programmtext laden
0105 32A7F1     00333           LD      (ddfd4+offs),A  ;dort auch
01C8 23         00334           INC     HL              ;Zeiger auf den Offset
01C9 7E         00335           LD      A,(HL)          ;.d" aus (IX/IY+d) laden
01CA 32A4F1     00336           LD      (dspl1+offs),A  ;in den Programmtext laden
01CD 32A9F1     00337           LD      (dspl2+offs),A  ;dort auch
0100 23         00338           INC     HL              ;Zeiger auf Operation
0101 7E         00339           LD      A,(HL)          ;Befehl laden
0102 E607       00340           AND     07h             ;Bits 0-2 maskieren
01D4 3C         00341           INC     A               ;+1
0105 47         00342           LD      B,A             ;als Zähler
0106 3E3E       00343           LD      A,3eh           ;0pcode 46-7E erzeugen
01D8 C608       00344 opclop1   ADD     A,08h           ;(46/4E/56/5E/66/6E/7E
010A 10FC       00345           DJNZ    opclop1         ;= LD B/C/D/E/H/L/A,(IX+d) )
010C 32A8F1     00346           LD      (load+offs),A   ;in den Programmtext
01DF 7E         00347           LD      A,(HL)          ;Befehl noch einmal laden
01E0 E6F8       00348           AND     0f8h            ;Bits 0-2 löschen
01E2 0F         00349           RRCA                    ;und herausrotieren
01E3 0F         00350           RRCA
01E4 0F         00351           RRCA
01E5 3C         00352           INC     A               ;+1
01E6 47         00353           LD      B,A             ;als Zähler
01E7 3EFE       00354           LD      A,0feh          ;(06/0E/16/1E/26/2E/3E
01E9 C608       00355 opclop2   ADD     A,08h           ;= RLC/RRC/RL/RR/SLA/SRA/SRL/RES/SET ...)
01EB 48         00356           LD      C,B             ;NOP erzeugen (für reta)
01EC FE36       00357           CP      36h             ;SLIA/SLL ?
01EE 200B       00358           JR      NZ,noslia       ;falls nein
01F0 0637       00359           LD      B,37h           ;SCF erzeugen
01F2 7E         00360           LD      A,(HL)          ;Befehl restaurieren
01F3 FE36       00361           CP      36h             ;SLIA/SLL (IX/IY+d) ?
01F5 2002       00362           JR      NZ,nopro        ;falls nein, kein Problem
01F7 0EC9       00363           LD      C,0c9h          ;Opcode RET für reta
01F9 3E16       00364 nopro     LD      A,16h           ;RL erzeugen (statt SLIA/SLL)
01F8 32A5F1     00365 noslia    LD      (opera+offs),A  ;in den Programmtext
01FE 78         00366           LD      A,B             ;SCF oder NOP
01FF 32A1F1     00367           LD      (cflag+offs),A
0202 79         00368           LD      A,C             ;RET oder N0P
0203 32A6F1     00369           LD      (reta+offs),A
0206 C1         00370           POP     BC              ;Register restaurieren
0207 F1         00371           POP     AF
0208 23         00372           INC     HL              ;PC korrigieren
0209 E3         00373           EX      (SP),HL         ;als RET-Adresse auf den Stack
                00374
020A 00         00375 cflag     NOP                     ;oder SCF
```

```
0208 00         00376 ddfd3     NOP                     ;DD oder FD für IX/IY
020C C8         00377           DB      0cbh            ;immer C8
020D 00         00378 dsp11     NOP                     ;Displacement d bei (IX/IY+d)
020E 00         00379 opera     NOP                     ;Operationsanweisung
020F 00         00380 reta      NOP                     ;RET bei SLIA/SLL
0210 00         00381 ddfd4     NOP                     ;DD/FD für IX/IY
0211 7F         00382 load      LD      A,A             ;Ladebefehl (hier Dummy)
0212 00         00383 dsp12     NOP                     ;s. dspll
0213 C9         00384           RET                     ;Simulation des Befehls beendet
                00385
                00386 ;Füllbytes für die "gerade" Adresse F200 der Utility-Vektortabelle
                00387 ;mit freundlicher Empfehlung:
0214 28         00388           DM      '(R) #0000 - (C) 1987 by The HACKT0RY '
0239 48         00389           DM      'Helmut Bernhardt, Arnulf Sopp, Gerald Schröder'
0267 FFFF       00390           DW      0ffffh
                00391
                00392 ;bei F200 die Tabelle der Einsprungsadressen bei vektorisiertem Z80-Task
                00393 ;zunächst alle auf FF (können nachgebrannt oder im RAM programm. werden)
0269 FFFF       00394 vectab    DW      0ffffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh,Offffh
0279 FFFF       00395           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0289 FFFF       00396           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0299 FFFF       00397           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02A9 FFFF       00398           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0289 FFFF       00399           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02C9 FFFF       00400           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0209 FFFF       00401           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02E9 FFFF       00402           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
02F9 FFFF       00403           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0309 FFFF       00404           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0319 FFFF       00405           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0329 FFFF       00406           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0339 FFFF       00407           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0349 FFFF       00408           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
0359 FFFF       00409           DW      0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh,0ffffh
                00410
                00411 ;dahinter die Vektortabelle für Interrupts des Typs 2
                00412 :im EPR0M ist nur der Einsprung für einen INT durch den Z80 initialisiert
0369 FFFF       00413 inttab    DW      0ffffh  ;für INTs auf den INT1-Pin (nicht init.)
0368 51F0       00414           DW      intz80+offs     ;dto. INT2-Pin (INT durch den Z80)
                00415 ;weiter für die internen Interrupts (nicht initialisiert)
                00416 ;(nur bis hierher muß das EPROM gebrannt werden, solange keine weiteren
                00417 ;Interrupt-Vektoren aufgesetzt werden; es folgen vorerst nur FF-Bytes) 036D
FFFF            00418 prt       DW      0ffffh,0ffffh   ;dto. PRT Kanal 0 und 1
0371 FFFF       00419           DW      0ffffh,0ffffh   ;dto. DMA Kanal 0 und 1
0375 FFFF       00420           DW      0ffffh          ;dto. CSI/0
0377 FFFF       00421           DW      0ffffh,0ffffh   ;dto. ASCI Kanal 0 und 1
                00422 ;dasselbe für die übrigen externen Interrupts (nicht weiter gelistet)
0378            00423 end       EQU     S               ;Hilfslabel für Programmende und -lange
                00424
                00425 ;Hilfslabels zur Kontrolle v. Unterprogramm- u. Tabellenadressen im Himem
F000            00426 dmaabs    EQU     dma0+offs       ;Arbeitsadresse des Unterprogramms dma0
F028            00427 dmatabs   EQU     srclog+offs     ;dto. Adressentabelle des DMAC
F0BA            00428 clonabs   EQU     clone+offs      ;dto. Unterprogramm clone0
F00A            00429 initabs   EQU     initab+offs     ;dto. Tabelle der int. Port-Initialwerte
F100            00430 trapabs   EQU     trap+offs       ;dto. Trap-Routine
F200            00431 vectabs   EQU     vectab+offs     ;dto. Utility-Vektortabelle
F300            00432 inttabs   EQU     inttab+offs     ;dto. INT-Vektortabelle
                00433
0000            00434           END
00000 Fehler
cbxx    0194    cflag   020A    clonabs F0BA    cloncnt 0136    clone   0123    const   01BC
copdone 00A9    coploop 0132    copy    010E    ddfd    0196    ddfd1   0188    ddfd2   016E
ddfd3   0206    ddfd4   0210    ddfdcb  01C1    dma0    0069    dmal    0084    dma2    0087
dmaabs  F000    dmacnt  009A    dmaresl 00FD    dmatabs F026    dspl1   020D    dsp12   0212
dstlog  0097    dstphys 0099    end     0378    gorst00 0038    gotrap  01AA    himem   F000
inidone 00A5    initab  0143    initabs F0DA    inttab  0369    inttabs F300    intz80  00BA
job     00E0    ldcon   01A8    load    0211    nmi     0066    nojob   00C4    nopro   01F9
noslia  01FB    nosub   0180    offs    EF97    opclop1 0108    opclop2 01E9    opera   020E
polloop 0066    prt     036D    putresl 0103    rdyflag 009C    reta    020F    spbuff  0003
srclog  0094    srcphys 0096    substit 0168    switch  000C    tabend  0098    trag    0169
trapabs F100    ufo     017A    vectab  0269    vectabs F200    vectask 0113    vectret 0100
```

```
waitz80 009D
                ; Listing 2: Der Z80 gibt dem HD 64180 einen vorübergehenden Job
0003            spbuff    EOU     0003h   ;bei TRS-80 u. ä.: stattdessen RAM-Adresse wählen
5200                      ORG     5200h
5200 211E52     start     LD      HL,job          ;Zeiger auf das Programm für den HD
5203 E5                   PUSH    HL              ;auf dem Steck ablegen
5204 7C                   LD      A,H             ;A - irgendetwas zwischen 02 und FE
5205 F5                   PUSH    AF              ;Signalbyte für 'bitte Job übernehmen!'
5206 ED730300             LD      (spbuff),SP     ;dem HD den Stackpointer mitteilen
520A D3F6                 OUT     (0fbh),A        ;die Buskontrolle an den HD übergeben
                ;jetzt ist der Z80 für einige Mikrosekunden im Schlummerzustand
                ;der HD holt sich soeben den Job nach oben
                ;danach:
520C F1                   POP     AF              ;Stack wieder bereinigen
520D F1                   POP     AF              ;(es waren zwei Ebenen)
                ;         CALL    WICHTIG         ;inzwischen etwas Wesentlicheres tun
520E 210038               LD      HL,3800h        ;Bildschiradresse (sichtbar ab 3C00)
5211 E5                   PUSH    HL              ;als Pufferadresse für den HD
5212 3E01                 LD      A,01h           ;Signalbyte 'bitte Ergebnisse abliefern!'
5214 F5                   PUSH    AF              :für den HD auf den Stack
5215 ED730300             LD      (spbuff),SP     ;wie oben
5219 D3FB                 OUT     (0fbh),A        ;Buskontrolle wieder an den HD übergeben
                ;wieder ist für einen Moment Funkstille, während der der HD die Codefolge
                ;20h, 20h. 20h, ... (Leerzeichen) in den Bildschirmspeicher lädt
                :der Bildschirm ist jetzt gelöscht
                ;danach:
5216 F1                   POP     AF              ;dasselbe wie oben
521C F1                   POP     AF
521D C9                   RET                     ;oder was auch immer
521E 2100F3     job       LD      HL,0f300h       ;Datenpuffer für Rechenergebnisse
5221 1101F3               LD      DE,0f301h       ;die nächste Stelle
5224 01FF07               LD      BC,07ffh        ;Länge 2 k8
5227 3620                 LD      (HL),' '        ;am Pufferanfang ein Leerzeichen ablegen
5229 EDB0                 LDIR                    ;den ganzen Puffer mit Blanks füllen
5226 C9                   RET                     ;in die Warteschleife zurückkehren
5200                      END     start
00000 Fehler
job  521E       spbuff 0003     start 5200
                ; Listing 3: Der Z80 gibt die Buskontrolle endgültig an den HD ab
0003            spbuff    EOU     0003h           ;TRS-80 u. ä.: stattdessen RAM-Adresse wählen!
4200                      ORG     4200h           ;im Sektorpuffer ist Platz genug
4200 211042     start     LD      HL,entry        ;Ansprungsadresse für den HD 64180
4203 E5                   PUSH    HL              ;für den HD auf den Stack
4204 211A42               LD      HL,bank         ;Platz für das Banking-Programm des HD
4207 E5                   PUSH    HL              ;dto.
4208 AF                   XOR     A               ;A - 00, Signal für endgültige Übernahme
4209 F5                   PUSH    AF              ;dto.
420A ED730300             LD      (spbuff),SP     ;dem HD verraten, wo der Stack ist
420E D3FB                 OUT     (0fbh),A        ;und den Bus endgültig übergeben
4210 ED56       entry     IM      1               ;für die Interrupt-Bearbeitung von G-D0S
4212 3E01                 LD      A,01h           ;nur noch Interrupts am Pin INT0 zulassen
                ;         OUT0    (34h),A         ;Ausgabe auf den INT/TRAP-Port
4214 ED                   DB      0edh,39h,34h    ;Hex-Entsprechung des Befehls
4217 C32D40               JP      402dh           ;DOS-Entry, dort steht z. B. auch EI
                bank                              ;Platz für die Banking-Routine des HD
```

```
4200                            END     start
00000 Fehler
bank    421A    entry 4210      spbuff 0003     start 4200
                ; Listing 4: Einsprung in d. Trap-Routine beim Genie 3s unter G-D0S
0000                            ORG     0000h           ;RST-00-Routine
0000 C30039                     JP      trap            ;dort weitermachen
3900                            ORG     3900h           ;hier ist z. B. freies RAM
3900 322739           trap      LD      (accbuf1),A     ;Akku retten (Stack unklar, kein PUSH!)
3903 ED5F                       LD      A,R             ;Interrupt-Enable-Flipflop ins P/V-Flag
3905 F3                         DI                      ;Interrupts maskieren
3906 3EF3                       LD      A,0f3h          ;auf Verdacht: DI
3908 E20D39                     JP      P0,intok        ;falls die Interrupts disabled waren
3908 3EFB                       LD      A,0fbh          ;sonst den Befehl EI laden
390D 324439           intok     LD      (eidi),A        ;und vor den RET-Befehl patchen
                      ;         IN0     A,(38h)         ;den Inhalt des CBR lesen
3910 ED                         DB      0edh,38h,38h    ;(Hex-Entsprechung des Befehls)
3913 323939                     LD      (cbrbuff),A     ;und p,uffern
3916 3E70                       LD      A,70h           ;Common 1 des HD ab 7.x000
                      ;         OUT0    (38h),a         ;auf CBR ausgeben
3918 ED                         DB      0edh,39h,38h    ;(s. o.)
                      ;         IN0     A,(3ah)         ;den Inhalt des CBAR lesen
391B ED                         DB      0edh,38h,3ah    ;(s. o.)
391E 323E39                     LD      (cbarbuf),A     ;und p,uffern
3921 F6F0                       OR      0f0h            ;Common 1 des HD ab 7.F000
                      ;         OUT0    (3ah),A         ;MMU neu konfigurieren
3923 ED                         DB      0edh,39h,3ah    ;(s. o.)
3926 3E00                       LD      A,00            ;Akku restaurieren (00 ist Dummy)
3927                  accbufl   EQU     $-1             ;dort ist der Akku-Inhalt gepuffert
3928 ED733339                   LD      (spbuff),SP     ;Stackpointer retten
392C 3100F1                     LD      SP,0f100h       ;Stack im Common-1-Bereich
392F CDD0F0                     CALL    0f100h          ;an 7F100 steht die Trap-Routine
3932 310000                     LD      SP,0000h        ;den alten Stack restaur. (Dummy-Adresse)
3933                  spbuff    EQU     $-2             ;je nach dem
3935 324339                     LD      (accbuf2),A     ;Akku und Flags retten (ohne Stack, s. o)
3938 3E00                       LD      A,00h           ;alter Inhalt des CBR (00 ist Dummy)
3939                  cbrbuff   EQU     $-1             ;je nach dem
                      ;         OUT0    (38h),A         ;den alten Registerinhalt restaurieren
393A ED                         DB      0edh,39h,38h    ;(s. o.)
393D 3E00                       LD      A,00h           ;dasselbe für das CBAR
393E                  cbarbuf   EQU     1-1
                      ;         OUT0    (3ah),A
393F ED                         DB      0edh,39h,3ah
3942 3E00                       LD      A,00h           ;Akku und Flags restaurieren (s. o.)
3943                  accbuf2   EQU     $-1
3944 FB               eidi      EI      ;oder DI, je nach altem Zustand
3945 C9                         RET     ;das war's
0000                            END
00000 Fehler
accbufl 3927    accbuf2 3943    cbarbuf 393E    cbrbuff 3939    eidi    3944    intok   390D
spbuff 3933     trap    3900
                ; Listing 5: möglicher TRAP-Einsprung unter CP/M u. a.
0000                            ORG     0000h
0000 C30300                     JP      trap
                      ;wo trap im RAM liegen kann, muß je nach CP/M-Version entschieden werden
                      ;         ORG     irgendwo
                      trap
                      ;         IN0     A,(34h)         ;INT/TRAP-Register lesen
0003 ED                         DB      0edh,38h,34h    ;(Hex-Entsprechung des Befehls)
0006 CB7F                       BIT     7,A             ;war das ein TRAP-Interrupt?
                      ;         JP      NZ,warmstart    ;falls nein (Sprungadresse je nach dem)
0008 E67F                       AND     7fh             ;TRAP-Bit löschen
000A 321300                     LD      (portbuf),A     ;den Input von Port 34h puffern
000D AF                         XOR     A               ;A <- 00, auch die ITE-Bits =0 (wie DI)
                      ;         OUT0    (34h),A         ;TRAP resetten und Interrupts disablen
```

```
000E ED                 DB      0edh,39h,34h    ;(s. o.)

                ;hier folgt die Trap-Routine (oder CALL nach 7.F100), und so endet sie:

0011 F5         exitrap PUSH    AF              ;Akku und Flags retten
0012 3E00               LD      A,00h           ;alter Inhalt von Port 34 (Dummy-Operand)
0013            portbuf EQU     $-1             ;Operanden-Byte des Befehls = Port-Inhalt
                ;       OUT0    (34h),A         ;Inhalt des Ports restaurieren
0014 ED                 DB      0edh,39h,34h    ;(s. o.)
0017 F1                 POP     AF              ;Akku und Flags restaurieren
0018 C9                 RET

0000              END
00000 Fehler
exitrap 0011    portbuf 0013    trap    0003
                ; Listing 6: Testprogramm für maximale Arbeitsgeschwindigkeit

0003            spbuff  EQU     0003h           ;TRS-80 u. ä.: auf freie RAM-Adresse ändern!

4200              ORG   4200h                   ;das paßt in den Sektorpuffer

4200 211042     start   LD      HL,entry        :Ansprungadresse für den HD 64180
4203 E5                 PUSH    HL              ;für den HD auf den Stack
4204 211F42             LD      HL,bank         ;Platz für das Banking-Programm des HD
4207 E5                 PUSH    HL              ;dto.
4208 AF                 XOR     A               ;A - 00. Signal für endgültige Übernahme
4209 F5                 PUSH    AF              ;dto.
420A ED730300           LD      (spbuff),SP     ;dem HD verraten, wo der Stack jetzt ist
420E D3FB               OUT     (Ofbh),A        ;und den Bus endgültig übergeben

                ;Einsprung bei der endgültigen Busübernahme durch den HD 64180
4210 ED56       entry   IM      1               ;für die Interrupt-Bearbeitung von G-DOS
4212 3E01               LD      A,01h           ;nur noch Interrupts am Pin INT0 zulassen
                ;       OUT0    (34h),A         ;Ausgabe auf den INT/TRAP-Port
4214 ED                 DB      0edh,39h,34h    ;(Hex-Entsprechung des Befehls)
4217 3E00               LD      A,00h           ;oder anderer Wert zwischen Ox und Fx
                ;       OUT0    (32h),A         ;Zahl der WAIT-Zyklen auf DCNTL ausgeben
4219 ED                 DB      0edh,39h,32h    ;(s. o.)
421C C32040             JP      402dh           ;DOS-Entry, dort steht z. B. auch EI

                bank                            ;Platz für die Banking-Routine des HD

4200                    END     start
00000 Fehler
bank    421F    entry 4210      spbuff 0003     start 4200
                ; Listing 7: Der vektororientierte Ansprung eines Programms

                ;Voraussetzung: Zuvor wurden von einem Job-Programm einige Routinen ir
                ;gendwo ab 4.0000 abgelegt, um den Z80-Speicher davon zu entlasten.
                ;Sie können auch im EPR0M fertig vorliegen, müssen dann aber im RAM mit
                ;OUT (0FBh),A beginnen.
                ;Ihre Entrypoints, an denen z. B. zuerst das evtl. notwendige Banking
                ;stattfinden könnte, sind in der Tabelle vectab (7.F200) der Reihe nach
                ;abgelegt worden.
                ;Hier soll nun das Programm Nr. 4 etwas berechnen. Später holt das Progr.
                ;Nr. 2 die Ergebnisse in den Z80-Speicher, die nicht in den für Resultate
                ;reservierten Raum ab 7.F800 gepaßt hätten.

0003            spbuff  EQU     0003h           ;TRS-80 u. ä.: auf freie RAM-Adresse ändern!

5200                    ORG     5200h
5200 210402     start   LD      HL,0204h        ;Signalbyte 02, Programm Nr. 4 (ab Nr. 0)
5203 E5                 PUSH    HL              ;auf den Stack für den HD 64180
5206 ED730300           LD      (spbuff),SP     ;Stackpointer für den HD ablegen
5208 D3FB               OUT     (0fbh),A        ;die Buskontrolle abgeben

                ;Pause, bis der HD im Prg. Nr. 4 die Kontrolle über den Bus zurückgibt

                ;       CALL    WICHTIG         ;dann z. B. Bedeutendes vollbringen ...

520A D1                 POP     DE              ;altes Signalbyte (Registerpaar egal)
```

```
520B 1E02                   LD      E,02h           ;Programm Nr. 2 (MSB ist noch richtig)
520D E5                     PUSH    HL              ;Anforderung für den HD auf den Stack
520E D3FB                   OUT     (0fbh),A        ;wie oben (korrekter SP noch in 7FFE)

                 ;wieder ein paar Millisekunden Pause ..., dann geht es irgendwie weiter.

5200                        END     start
00000 Fehler
spbuff 0003     start 5200
```

1987-BHV-02 S.31-36 / 1987-Club80 Nr.18 S.??-70

512K-RAM-Adressierung mit flexibler MMU
2 serielle Schnittstellen
2 DMA-Kanäle
2 16-Bit-Timer
Erweiterter Befehlssatz
Höhere Geschwindigkeit
durch

Ersetzen des Z80 durch den HD64180

Helmut Bernhardt

Der Hersteller der HD64180-CPU benötigte zur Darstellung aller Leistungen dieses Z80-aufwärtskompatiblen Prozessors in einem "Datenblatt" ein 206 Seiten starkes Buch. Deshalb soll hier garnicht erst der Versuch gemacht werden, die Vorteile dieses Chips gegenüber seinem Vorgänger Z80 zusammenzufassen. Hier soll nur dem Bastler eine Möglichkeit gezeigt werden, wie dieser Prozessor im alten Z80-Microcomputer die Regie übernehmen soll.

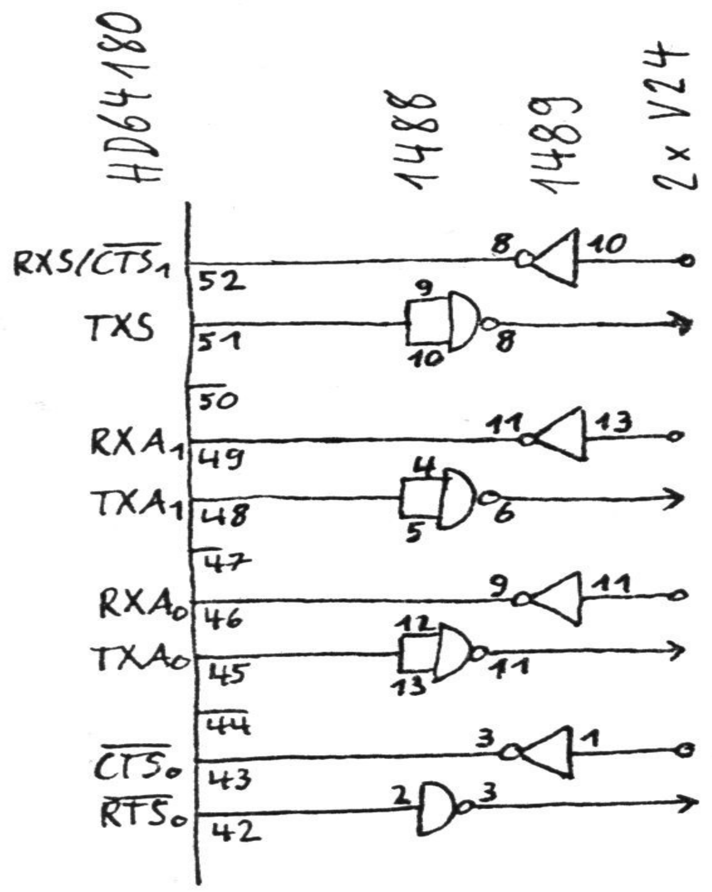
Für die Unterbringung des HD64180 anstelle des Z80 im Computer wird eine kleine Adapterplatine benötigt, an die von der Lötseite ein 40poliger Dilstecker angelötet wird und auf der die neue CPU und 5 weitere ICs untergebracht werden. Diese Platine wird in den Sockel des Z80 gesteckt und liefert über den Dilstecker alle Z80-äquivalenten Signale des HD64180 an den Computer.

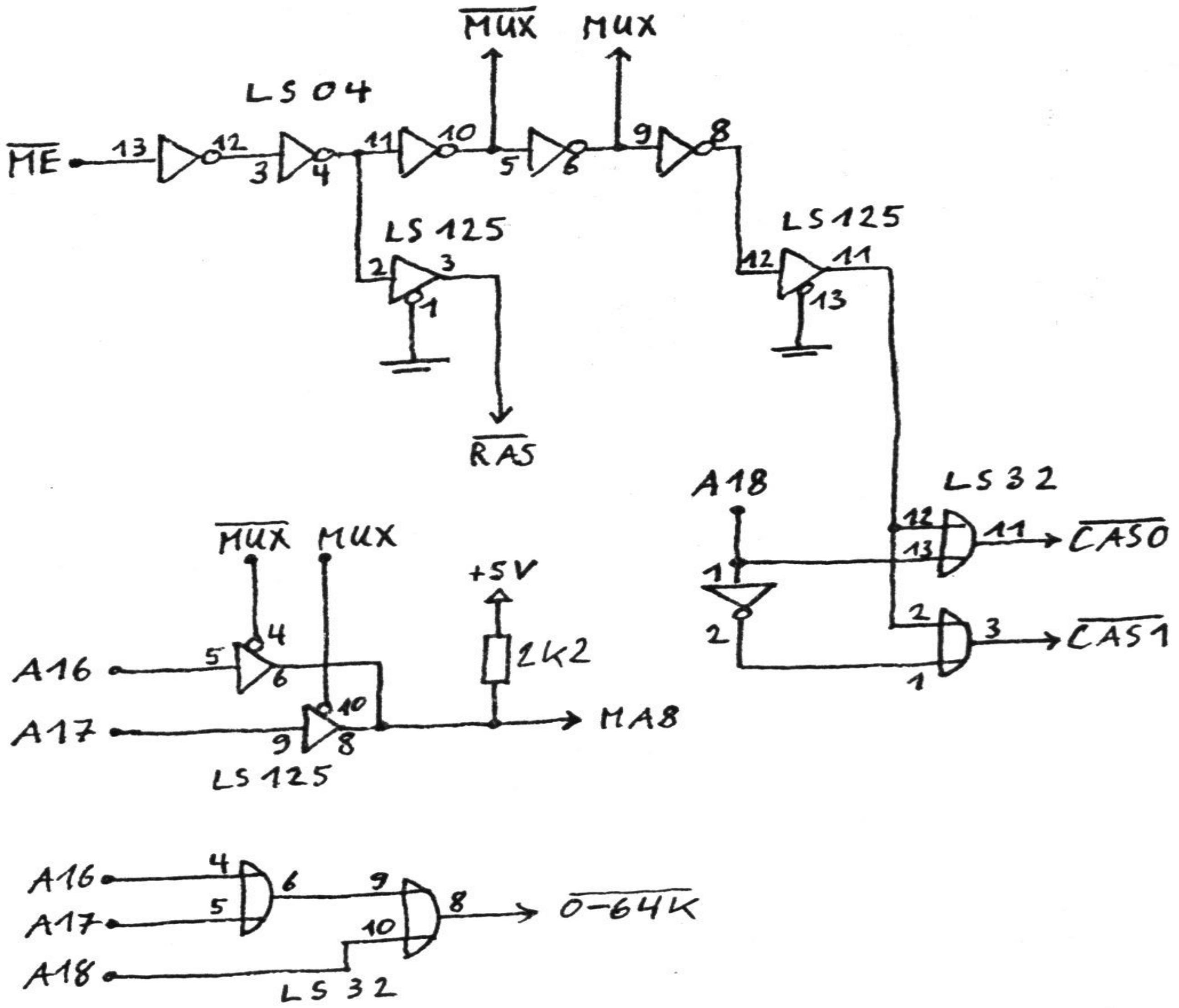
Auf der Platine sind schon zwei Treiber-ICs enthalten, die bei den Signalen der CPU-internen Seriellen Schnittstellen die Pegelwandlungen +12V/-12V => GND/+5V und umgekehrt vornehmen. Der Sendetreiber muß über zwei zusätzliche Leitungen +12 und -12V zugeführt bekommen. Die Signale der seriellen Schnittstellen können vom Stecker direkt abgenommen werden und z.B. an ein Modem geführt werden.

Auf der Adapterplatine wird der Takt für den Prozessor aus einem internen Oszillator und einem externen Qartz hergeleitet. An Pin 64 des HD64180 wird die durch 2 geteilte Qartzfrequenz, die dann dem Systemtakt entspricht bereitgestellt und kann bei Bedarf mit der Drahtbrücke J1 über den Pin 6 des Z80-Sockels an den Computer geliefert werden. Dann muß aber die bisherige Z80-Systemtaktquelle davon abgetrennt werden. Es muß geprüft werden, ob der Takt für den Z80 anderweitig als feste Frequenz verwendet wird. Dann muß dort weiterhin die Taktfrequenz für den Z80 hingeführt werden. Wenn aber der Takt für die Synchronisation mit der CPU benötigt wird, muß dieser vom Pin 64 des HD64180 entnommen werden.

Außerdem werden mit 3 weiteren ICs aus den zusätzlichen Adreßleitungen A16 bis A18 Freigabesignale für die Ansteuerung von 2 x 8 Stück 41256-RAMs und die an die Pins 1 dieser RAMs zu legende gemultiplexte RAM-Adresse MA8 erzeugt. Je nach Funktionsweise des bisherigen Adressmultiplexers für vorher 4164-RAMs wird für die Steuerung des Adressmultiplexers nun MUX oder MUX* vom Stecker CNB verwendet. Für die Wahl zwischen diesen Signalen muß geprüft werden ob das bisherige MUX-Signal low oder high aktiv war, entsprechend ist dann das low aktive MUX* oder das high aktive MUX zu verwenden. Das ist deshalb von Bedeutung, weil sonst bei Refresh die oberen Adressen an die RAMs durchgeschaltet werden, und dann das Refreshing nicht stattfinden kann.

Beschaltung der Pins der seriellen Schnittstellen





Schaltung zur Ansteuerung von 512K RAM auf dem Board

Das Signal RAS* von CNB kann direkt anstelle des bisherigen RAS*-Signals an die RAMs gelegt werden (eventuell über 33 Ohm). Bei Bestückung des Computers mit nur 8 Stück 41256-RAMs wird nur das Signal CAS0* von CNB benötigt. Dieses Signal wird nicht direkt an die RAMs geführt sondern an den Eingang des Gatters gelegt, mit dem die Verknüpfung mit dem RAM-Freigabesignal erfolgt. Diese Stelle in der Schaltung herauszufinden, ist die einzige Anforderung an den Bastler, der sich die Vorteile des HD64180 für seinen Computer zunutze machen will. Hier muß auch geprüft werden, ob das bisherige CAS-Signal high oder low aktiv an diese Stelle gelangte. Wenn hier ein high aktives CAS nötig ist, muß CAS0* noch über einen Inverter geführt werden.

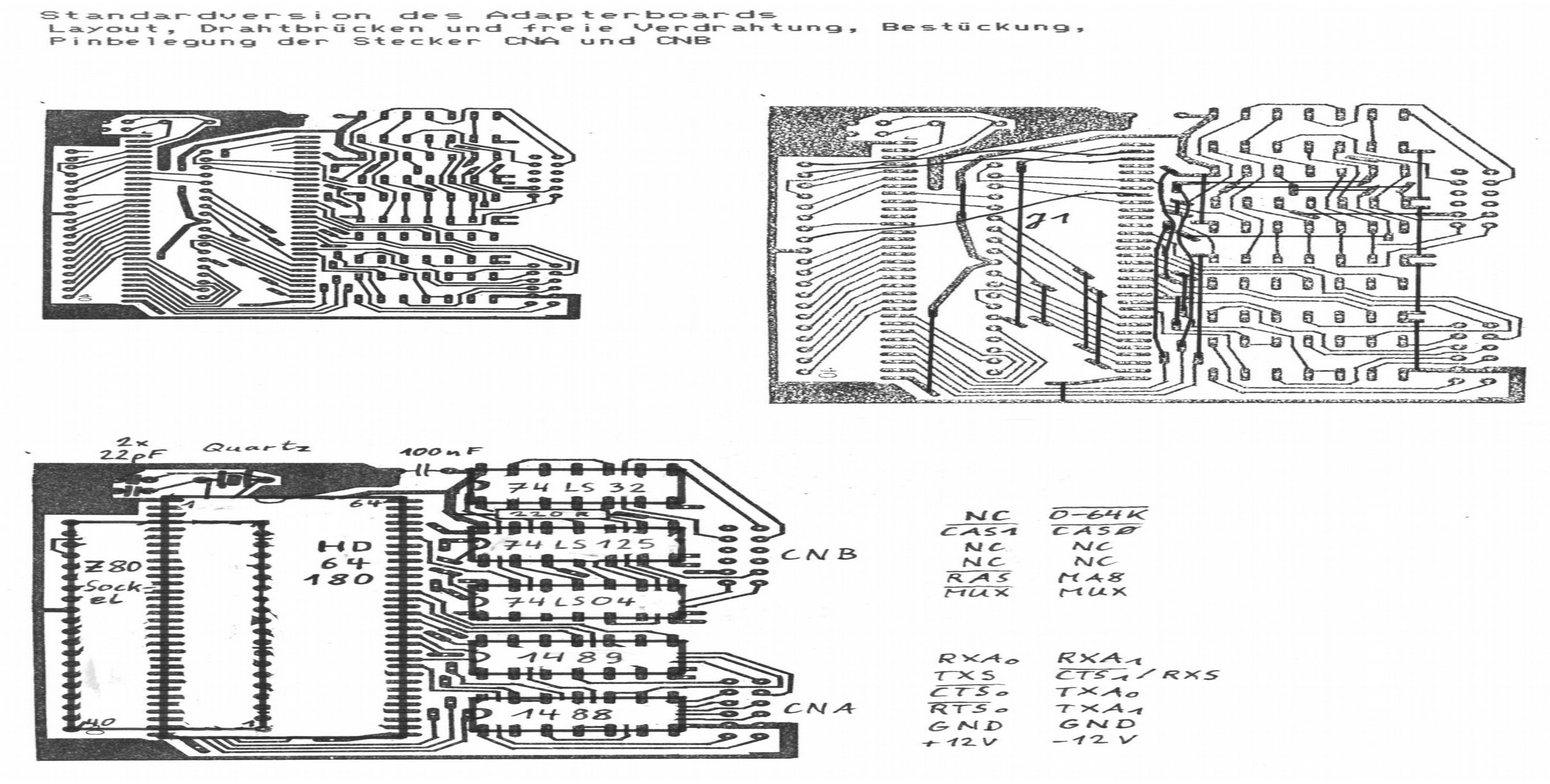
Wenn die bisherige Herleitung dieser Signale aus dem Systemtakt für den Z80 oder über Gatterlaufzeiten aus dem MERQ*-Signal erfolgte, kann diese weiterbenutzt werden. Bei einer Änderung des Systemtaktes beim Umrüsten kann bei größeren Unterschieden zwischen altem und neuem Takt eine Herleitung von RAS*, MUX bzw. MUX* und CAS* aus MERQ* über Gatterlaufzeiten eventuell nicht mehr funktionieren. Hier muß dann probiert werden, wieviele Verzögerungsgatter zwischen den einzelnen Signalen günstig sind.

Wenn das RAS*-, MUX- und CAS*-Timing bisher aus dem Systemtakt für den Z80 (nicht aus einem dazu vielfachen Muttertakt) hergeleitet wurde, kann diese Schaltung weiterbenutzt werden, indem die bisherige Taktquelle unschädlich gemacht wird und stattdessen der Takt von Pin 64 des HD64180 über Pin 6 des Z80-Sockels ins System geführt wird. Eventuelle Treiber des ursprünglichen Systemtaktes zwischen ursprünglicher Taktquelle und Z80 müssen überbrückt werden, um auch Bauteile zu erreichen, die mit dem Takt von diesem Treiber versorgt werden.

Dann werden nicht die Signale RAS*, MUX bzw. MUX* und CAS* des Adapterboards benutzt. Hier wird dann stattdessen das Signal MUX bzw. MUX* (was immer es dort ist) der bisherigen Schaltung an den Freigabe-Eingang des 74LS125-Treibers für A16 auf dem Adapterboard gelegt, der bisher mit diesem Pin verbundene Pin des 74LS04 muß dann aus der Fassung gebogen werden.

Wenn eine weitere Reihe von 8 Sockeln für RAMs vorhanden ist, oder wenn man das Risiko auf sich nehmen will, 8 weitere 41256-RAMs auf die anderen 8 Stück huckepack zu löten, dann wird der Pin 15 dieser zusätzlichen 8 RAMs mit CAS1* von CNB direkt beschaltet. Beim Huckepack-Verfahren dürfen die Pins 15 der beiden übereinander sitzenden ICs also nicht miteinander verbunden werden.

Es sei noch angemerkt, daß die meisten 4164-RAMs mit einer 7-Bit-Refreshadresse auskommen, während 41256-RAMs grundsätzlich eine 8-8it-Refreshadresse benötigen. Es muß also gewährleistet sein, daß A0-A7 beim Multiplexen zusammen durchgeschaltet werden. Bei älteren Geräten ist es wahrscheinlich, daß anstelle von A7 eine höhere Adresse zusammen mit A0-A6 durchgeschaltet wird. In diesem Fall sind A7 und die falsch angeschlossene höhere Adresse an den Multiplexer-Eingängen zu vertauschen.



Ein praktisches Beipiel

Nach diesem allgemeinen philosophischen Prolog, der eigentlich nur dem versierten Bastler prinzipiell zu berücksichtigende Punkte aufzeigt und wenig zum Wie sagt, soll nun an einem praktischen Beispiel gezeigt werden, was zu tun ist.

Das Beispiel ist ein GENIE IIs, dessen Besonderheit darin besteht daß dort eine Umschaltung des Taktes zwischen 1,77 und 5,3 (8) MHz möglilch ist. Wenn die 'LSP'-Taste gedrückt ist, das EPROM eingeschaltet ist und/oder, wenn durch die Software ein Bit in einem Steuerlatch auf 0 gesetzt wird (nach RESET ist dieses Bit immer 0), läuft der Computer nur mit 1,77 MHz. Ein Lesen des EPROMs ist bei höheren Taktraten nicht möglich. Diese Takterzeugung und -umschaltung kann für die HD64180-CPU nicht verwendet werden, mit diesem Prozessor muß man auf eine Taktumschaltung verzichten. Wegen der nach einem RESET selbstauferlegten 4 Waits dieser CPU ist es aber durchaus möglich, ein EPROM auch bei sehr hoher Taktrate noch zu lesen. Nach Abschalten des EPROMs können auch die Waits zurückgenommen werden. Der GENIE IIs läuft so mit ausschließlich 9,216 MHz ohne Probleme. Es muß allerdings darauf geachtet werden, daß vor dem Reaktivieren des EPROMs die Waits wieder eingefügt werden.

Dafür wurde einfach das Adapterboard für die Takterzeugung und -umschaltung entfernt und das 74S74-IC dieses Boards in den Sockel (U18) gesteckt, aus dem das Board entnommen wurde. Der Pin 6 dieses ICs ist aus der Fassung zu biegen. Bei U9, 74LS125 wurden die Pins 8 und 9 aus der Fassung gebogen und diese Pins des Sockels miteinander verbunden. Damit ist der Treiber zwischen alter (abgehängter) Taktquelle und Z80, Pin 6 überbrückt. Auf dem HD64180-Adapterboard werden an den dafür vorgesehenen Stellen ein 18,43 MHz-Quartz und die beiden 22pF-Kondensatoren bestückt. Der Pin 64 des HD64180 und der Pin 6 des Steckers für den Z80-Sockel werden durch einen Draht miteinander verbunden.

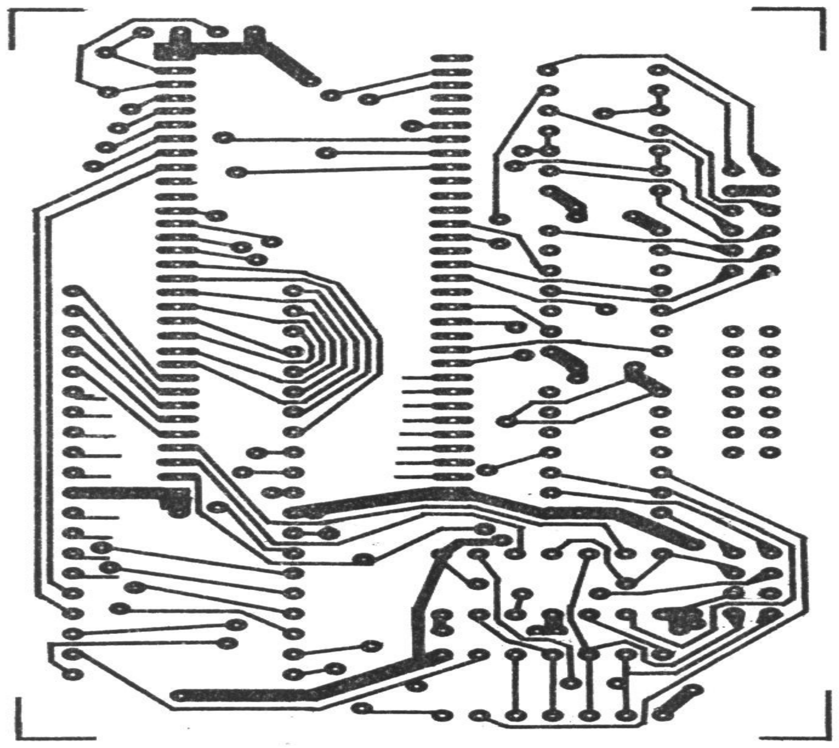
Beim GENIE IIs ist außerdem zu berücksichtigen, daß durch entsprechende Bits im Systemsteuerlatch zwischen RAM und memory-mapped I/O-Baugruppen innerhalb der unteren 16K umgeschaltet werden kann. Die Decodierung der Freigabesignale dieser Baugruppen berücksichtigt natürlich nur die Z80-Adressen bis A15 und nicht die HD64180-Adressen A16 bis A18.

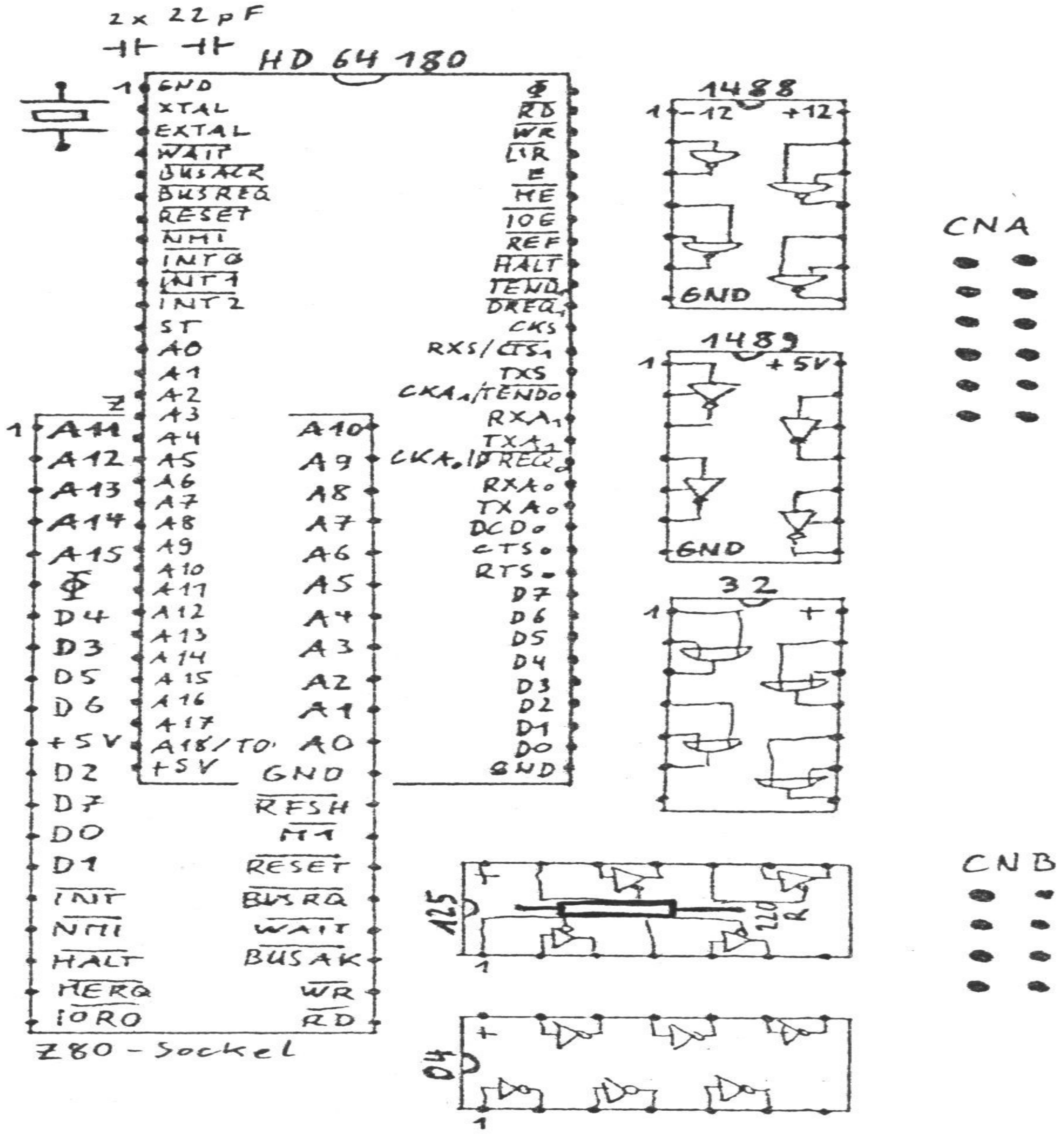
Entsprechend würden diese Baugruppen auch in den entsprechenden Adreßbereichen anderer Banks (physialische Adressen 10000H-7FFFFH) angesprochen werden.

Um dies zu verhindern, muß in der Adressdekoder-Baugruppe auf der I/O-Karte anstelle von A15 ein durch OR-Verknüpfung aus 0-64K* und A15 hergeleitetes 0-32K* Signal verwendet werden, das an den aus der Fassung gebogenen Pin 5 von U13, 74LS138 gelegt wird (Siehe Abbildung).

Aus geometrischen Gründen läßt sich die Standardversion des Layouts des Adapterboards im GENIE IIs nicht verwenden. Ein für diesen Computer geeignetes Layout ist ebenfalls gezeigt, bei dem aber wegen ungünstigerer Anordnung der Bauteile zueinander sehr viel mehr Drahtbrücken und freie Verdrahtungen nötig sind.

Sonderversion für GENIE IIs



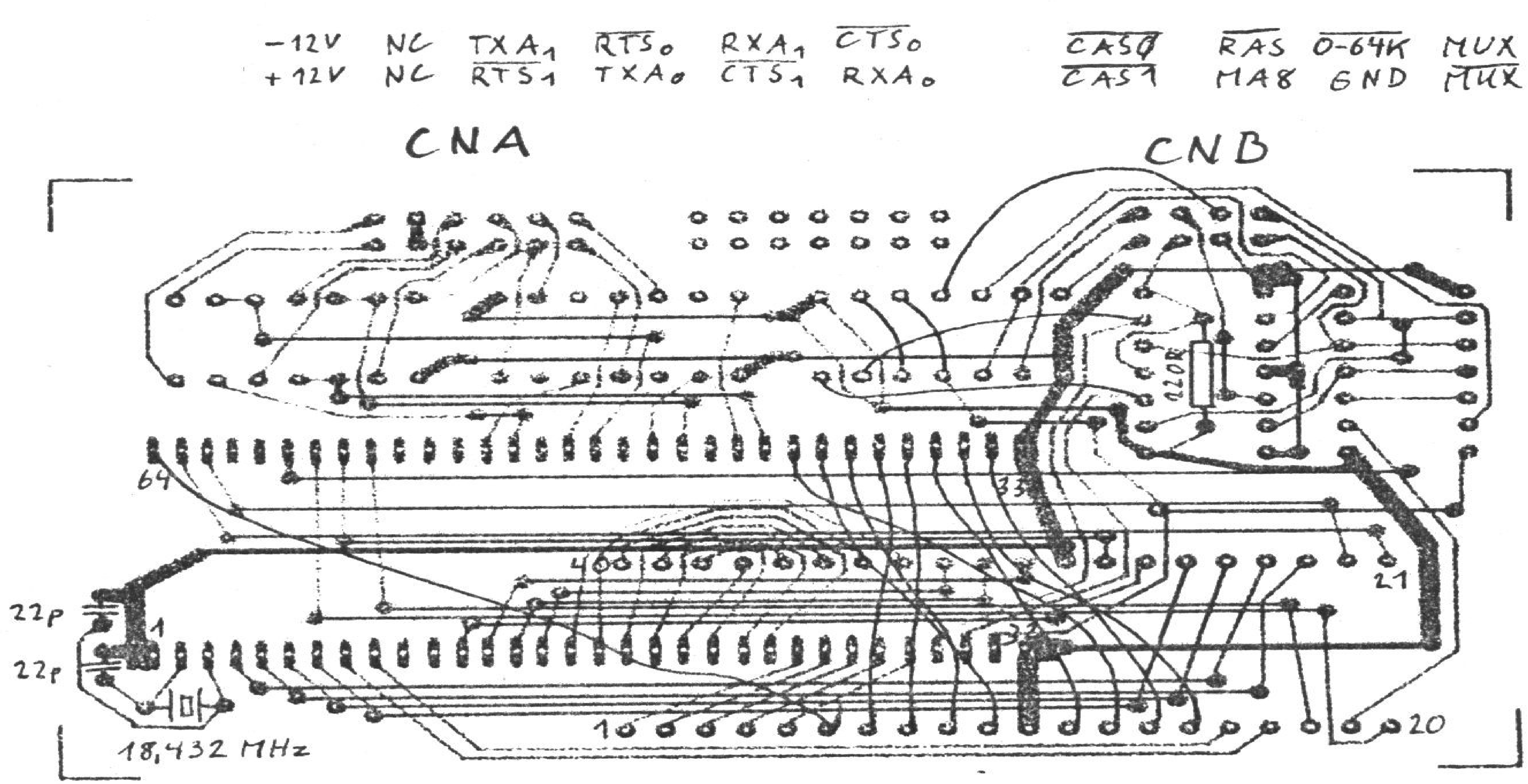


Beim GENIE IIs wird das RAS*-, MUX- und CAS*-Timing aus dem Systemtakt hergeleitet und kann deshalb beibehalten werden. Mux von Pin 1 von U6, 74S157 kann für die Umschaltung von A16 und A17 auf dem HD64180-Adapterboard verwendet werden. Dafür wird durch freie Verdrahtung von Pin 1 von U6 eine Verbindung mit MUX von CNB des Adapterboards hergestellt und der Pin 5 des 74LS04 auf dem Board aus der Fassung gebogen.

Auf dem speziell für den GENIE IIs gebauten Adapterboard muß die Erzeugung des 0-64K*-Signals noch in freier Verdrahtung nachgearbeitet werden. Dazu stehen die OR-Gatter des 74LS32 (Pins 1,2,3 und 13,12,11) noch zur Verfügung. Das Signal kann über den unbenutzten Pin des Steckers CNB dem I/O-Board zugeführt werden.

Die Sonderausführung des Adapterboards für den GENIE IIs entspricht prinzipiell auch der angegebenen Schaltung. Jedoch wurden teilweise andere Gatter innerhalb der ICs benutzt, so daß die Pin-Nummern in der Schaltung nicht überall zutreffen.

Wieviele Waits der einzelne Computer beim HD64180 und höherer Taktrate beibehalten muß, ist im Einzelfall zu untersuchen. Es ist durchaus nicht in jedem Computer ein Takt von 9,216 MHz möglich, auch wenn alle 4 Waits bei Memory-Zugriffen beibehalten werden.



Und wie geht es in meinem Computer?

Auf diese Frage hin muß ich nun doch wieder auf die allgemeingültigen Bemerkungen zurückkommen. Ich habe weder das Geld noch die Zeit noch überhaupt die Lust, diesen Hardware-Patch in allen mir zugänglichen Computern auszuprobieren, um dann für alle Geräte einfach nachzuvollziehende Lötanleitungen vorzulegen.

Ein bißchen Abenteuer und Eigenleistung und dadurch auch mehr Spaß an der Sache soll denen bleiben, die sich an dieses Projekt heranwagen. Als Unterstützung kann ich lediglich eine ungebohrte geätzte Platine (einer der beiden vorgestellten Varianten) für 10,-DM zur Verfügung stellen und eventuell fernmündlich oder -schriftlich helfen, soweit ich vom Stand meiner zwangsläufig begrenzten Hobbyistenkenntnisse und der Versorgung mit Schaltplänen der verschiedenen Geräte dazu in der Lage bin.

Über Einbauanleitungen des HD64180-Adapters in spezielle Computer (und dadurch indirekt über Erfolgsmeldungen) im Info würde ich mich sehr freuen.

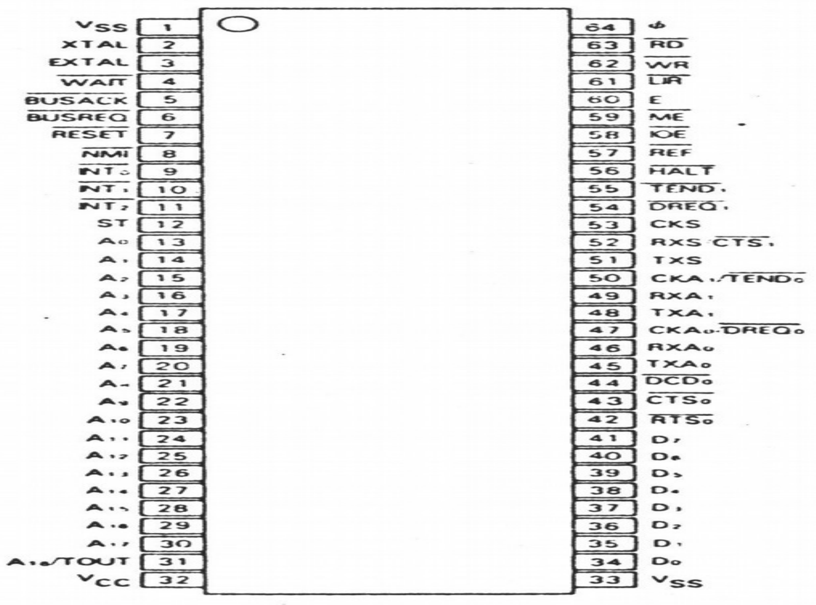
Helmut Bernhardt

Direkt zu verbindende Pins

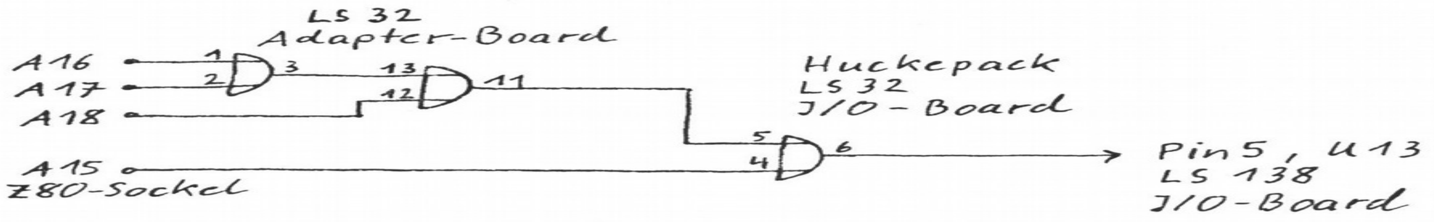
HD64180 Pin	Signal	Z80 Pin
13	A0	30
14	A1	31
15	A2	32
16	A3	33
17	A4	34
18	A5	35
19	A6	36
20	A7	37
21	A8	38
22	A9	39
23	A10	40
24	A11	1
25	A12	2
26	A13	3
27	A14	4
28	A15	5
34	D0	14
35	D1	15
36	D2	12
37	D3	8
38	D4	7
39	D5	9
40	D6	10
41	D7	13
6	BUSREQ*	25
5	BUSACK*	23
7	RESET*	26
8	NMI*	17
9	INT0* = INT*	16
59	ME* = MERQ*	19
58	IOE* = IORQ*	20
63	RD*	21
62	WR*	22
57	REF* = RFSH*	28
12	ST* = M1*	27
56	HALT*	18
4	WAIT*	24
64	PHI	6 #
32	+5V	11
1,33	GND	29

CLOCK ist beim Z80 ein Eingang und beim HD64180 ein Ausgang

Pinout der HD64180-CPU



(Top View)



Herleiten von 0-64K* auf dem Adapterboard der Sonderversion und Verknüpfen mit A15 zur Erzeugung des Signals 0-32K* für den Pin 5 von U13 auf dem I/O-Board

Auslesen des Graphikspeiches

GDP64-Karte des NDR-Klein-Computers im TRS80 und GENIE

Helmut Bernhardt, Jörg Seelmann-Eggebert

Zwischen der in den Computern TRS80 und GENIE wohl am weitesten verbreiteten HRG 1B und der GDP64 des NDR-Klein-Computers bestehen folgende gravierenden Unterschiede:

Feature	**HRG 1B**	**GDP64**
Bildformat	192x384	256x512
Mischbareit des Video-signals mit dem des Text-Bildschirms	ja	nein
Geschwindigkeit der Gra-phikausgabe	langsam	sehr schnell
Auslesbarkeit des Graphik-speichers	ja	nein
Anzahl Bildschirmseiten	1	4
Programmierbarkeit	umständlich	komfortabel
Anschluß an den Computer	beide frei verdrahtet	
BASIC-Treiber vorhanden	ja	ja
Verschleiß an Portadressen	128	32

Die wesentlichsten Nachteile der GDP64 bestehen in der nicht vorhandenen Auslesbareit des Grafik-Speichers, wodurch ein Abspeichern von Bildern auf Diskette ein Ausdrucken der Graphik ein Spiegeln und Invertieren des Bildes und die Implementation eines Graphik-Cursors nicht möglich sind.

Die nicht vorhandene Mischbarkeit des Videsignals mit dem des normalen Textbildschirms kann dadurch umgangen werden, daß entweder ein zweiter Monitor benutzt wird oder eine einfache softgesteuerte (und/oder handgesteuerte) Umschaltung des Videosignals für den Monitor dazugestrickt wird. Ein möglicher Schaltungsvorschlag wird zum Schluß noch vorgestellt.

Zunächst aber soll das Hauptproblem, das Auslesen des Graphikspeichers gelöst werden. Dazu sei vorweg bemerkt, daß wegen des kritischen Timings des EF9396-Graphikprozessors nicht garantiert werden kann, daß jeder Chip dabei funktioniert. Mit einem Pulldown-Widerstand kann der abfallenden Flanke des entsprechenden Signals zwar etwas Dampf gemacht werden, ob das bei dem jeweiligen Exemplar des 9366 dann ausreicht, muß probiert werden. Durch Verringern des Widerstandswertes läßt sich wahrscheinlich auch noch was rausholen.

Für das Auslesen des Graphik-Speichers der GDP64 benötigt man 2 zusätzliche ICs und einen 2k2-Widerstand. Hauptbetandteil ist ein 74LS374-Latch, in das der EF9366 den Inhalt der ihm benannten 8 Bits des Bildwiederholspeichers einträgt und aus dem sich die CPU dieses Datum abholen kann. Außerdem sind noch zwei OR-Gatter zur Ansteuerung des Latch nötig.

Und was muß gemacht werden?

Nicht viel. Bei dem zusätzlichen 74L374 werden außer den Pins 10 und 20 alle Pins um 45 Grad hochgebogen. Dann wird IC 20,74LS245 der GDP64-Karte aus der Fassung gezogen und das 74LS374 mit den Pins 10 und 20 auf den 74LS245 an dessen Pins 10 und 20 angelötet.

Dann wird IC 19, 74LS273 aus der Fassung genommen und auf dem Tisch so neben den eben gebauten Doppeldecker gelegt, wie die ICs 19 und 20 nebeneinander auf der Karte gesteckt haben. Mit möglichst kurzen isolierten Drahtstücken werden dann alle in der folgenden Tabelle aufgeführten Verbindungen zwischen Huckepack-LS374 und 74LS273 hergestellt. Beim 74LS273 sollten die Drahtenden möglichst weit oben am Pin angelötet werden, damit die Pins nachher auch wieder in die Fassung passen. Danach kann das Verbundsystem wieder in die Fassungen gesteckt werden.

Nun werden die Verbindungen des Doppeldecker-74L374 mit den Pins des 74LS166 entsprechend der rechten Hälfte der Tabelle hergestellt, wobei dann der 74LS166 aus der Fassung zu ziehen und locker auf diese zu stellen ist (damit sich die Drahtlängen richtig bemessen lassen und andererseits beim Anlöten keine dauerhafte Verbindung zwischen IC und Fassung hergestellt wird; das entspräche dann auch nicht mehr dem Sinn einer Fassung).

Verbindungen zwischen dem Huckepack-LS374 und IC 19 sowie IC 4

Signal	**IC19, LS273** Pin	**Huckepack LS374** Pin	Pin	**IC4, LS166** Pin
D1	3 ------------- 2		3 -------------- 3	
D3	4 ------------- 5		4 -------------- 5	
D5	7 ------------- 6		7 ------------ 11	
D7	8 ------------- 9		8 ------------ 14	
D6	13 ----------- 12		13 ------------ 12	
D4	14 ----------- 15		14 ------------ 10	
D2	17 ----------- 16		17 -------------- 4	
D0	18 ----------- 19		18 -------------- 2	

Nun wird ein zusätzliches 74LS32 genommen und dessen Pins 1-4, 6 und 8-13 ebenfalls um 45 Grad hochgebogen. Mit den Pins 5, 7 und 1 wird es auf dem aus der Fassung genommenen IC8, 74LS32 an dessen Pins 5, 7 und 14 angelötet. Nun kann auch dieser Doppeldecker wieder in seine Fassung gesteckt werden. Eventuell müssen die abgespreitzten Pins der beiden Huckepack-ICs noch etwas zurechtgebogen werden, damit sie sich nicht gegenseitig berühren und auch keinen Kontakt mit ihren Träger-ICs haben. Dann werden folgende Verbindungen hergestellt:

Huckepack-LS32 Verbinden mit

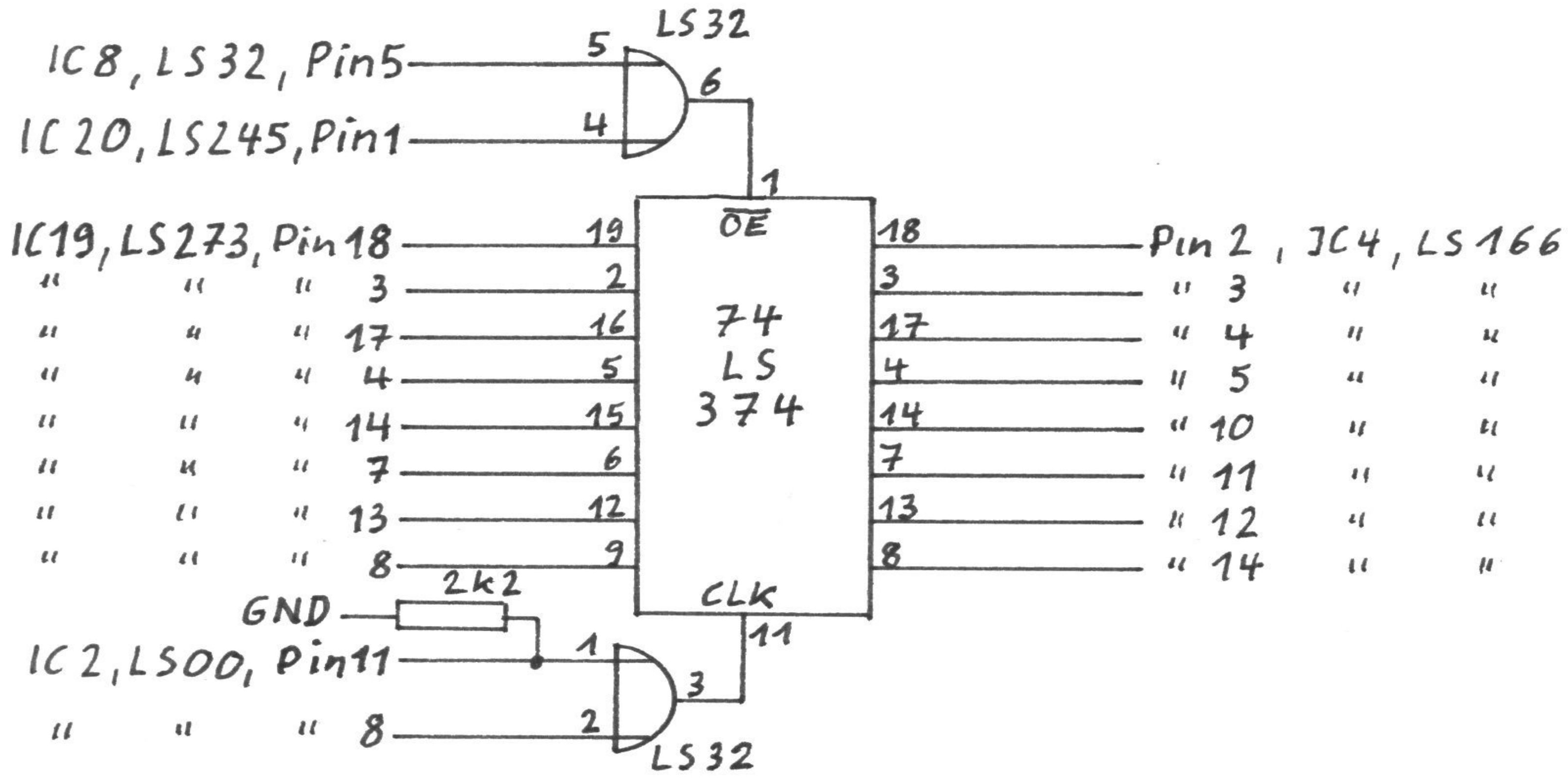
Pin 1	IC2, 74LS00, Pin1
Pin 2	IC2, 74LS00, Pin8
Pin 3	Huckepack-LS374, Pin11
Pin 4	IC20, 74LS245, Pin1
Pin 6	Huckepack-LS374, Pin1

Und schließlich werden noch die Pins 1 und 7 des IC2 74LS00 über einen 2k2-Widerstand verbunden (Pulldown des MFREE-Signals des EF9366).

Damit ist alles getan, um die dem EF9366 vorgegebene Speicherstelle auslesen zu können.

Wenn danach das Auslesen des HRG-Speichers nicht funktioniert, ist das nicht verwunderlich, weil man zunächst etwas falsch gemacht hat; zumindest ich (Helmut) bin das so gewohnt. Dieser Fehler kann aber durch Vergleich des Strippengewirrs mit der Einbauanleitung schnell gefunden werden.

Möglicherweise treten dann beim Auslesen immer noch Fehler auf. Die Ursache liegt dann beim oben beschriebenen Timing des EF9366 mit seinem MFREE-Signal. Man kann dann noch mit einem kleineren Widerstand als 2k2 Versuche machen oder eventuell einen anderen EF9366 ausprobieren. Letztere Variante kann natürlich recht kostspielig werden, bis man ein funktionierndes Exemplar gefunden hat. Alternativ kann man sich damit abfinden, daß es nicht funktioniert und alle Umbauten wieder abreißen (weniger erstrebenswert).



Und was bringt die GDP64 gegenüber der HRG 1B mehr?

An Software-Unterstützung gibt es dank der Bemühungen von Christof Ueberschaar ein 4K-Byte großes BASIC-Treiberprogramm, das ins HIMEM geladen wird und nach Einbinden in das Disk-BASIC sehr komfortable Graphik-Befehle bereitstellt. Gegenüber der Variante des Treibers für die HRG 1B wo die Graphik-Befehle durch ein # eingeleitet werden und daher der BASIC-Interpreter bei jedem Befehl erst prüfen muß, ob ein # vorliegt, ist hier der CMD-Vektor verbogen worden. Es wird dann nur noch bei Auftauchen des CMD-Tokens im Programmtext eine zusätzliche Abfrage gemacht, ob das nach CMD obligatatorische (") folgt und nur dann, wenn das nicht der Fall ist, weiter untersucht, ob das Byte ein gültiger Graphik-Befehl ist, der dann abgearbeitet wird.

Diese Philosophie hat zwar den Nachteil, daß man anstelle des einfachen # jedesmal 3 Zeichen eintippen muß, bevor der eigentliche Graphik-Befehl kommt, andererseits wird dadurch nicht die Abarbeitung der normalen BASIC- Befehle verlangsamt womit ein Teil des Geschwindigeitsvorteils der GDP64 verschenkt wäre.

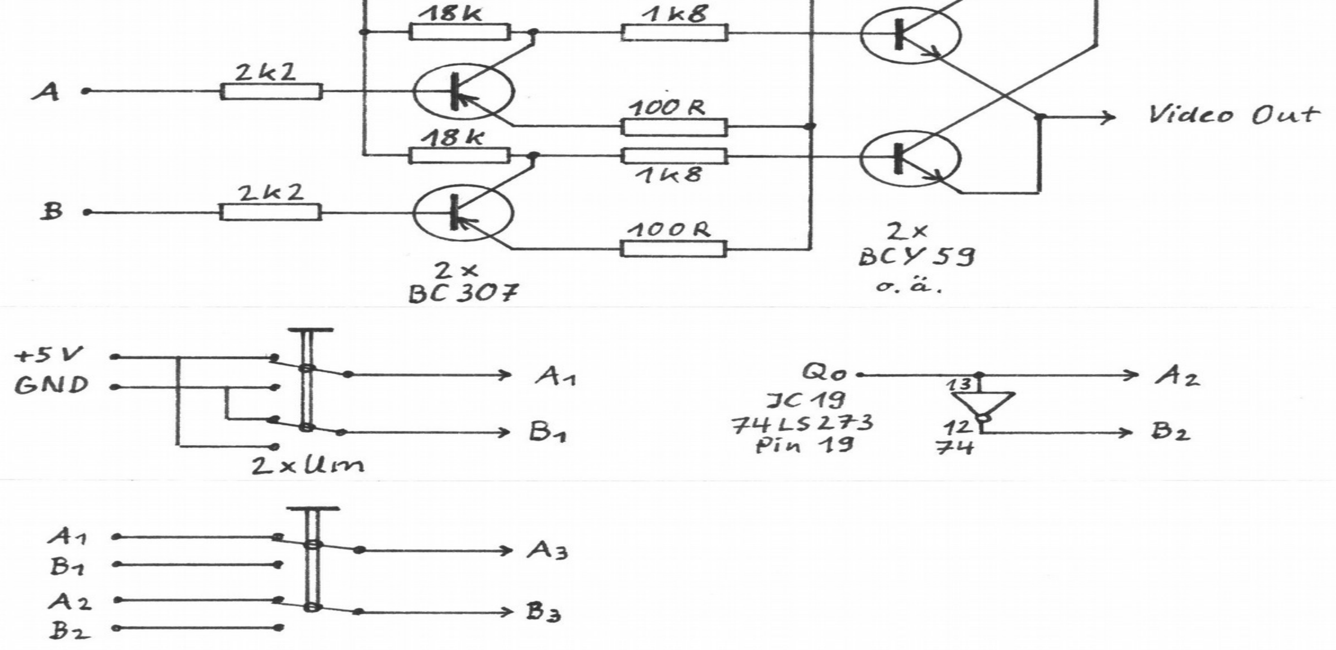
Außerdem ist die Punktadressierung der GDP64 bedeutend einfacher als die der HRG 1B. Es werden dem EF9366 einfach die X- und Y-Koordinaten des Punktes als 16-Bit-Wörter in entsprechende Register geschrieben und dann einer der Befehle "Punkt setzen", "Punkt löschen" oder "Punkt abfragen" ausgeführt. Bei der HRG 1B bedeutet das ein umständliches Berechnen der Speicheradresse und der entsprechenden Bits im Byte.

Noch gravierender ist die Eleganz beim Zeichnen von Linien über Vektorbefehle. Hier werden die Punktkoordinaten (X1/Y1) für den Ausgangspunkt einer Linie in die entsprechenden Register eingetragen und in zwei weitere Register werden die Werte Delta X und Delta Y eigetragen. Es sind dies die Differenzen X2-X1 und Y2-Y1 (Koordinaten von Anfangs- und Endpunkt einer Linie). Diese Differenzen müssen stets positive Zahlen sein. Das Vorzeichen (Vektorrichtung) wird durch entsprechende Bits in den Vektorbefehlen vorgegeben. Es läßt sich auch noch festlegen, ob die Linie durchgezogen, gepunktet oder punktstrichliert ausgegeben werden soll oder ob die Linie überhaupt gezeichnet werden soll oder eine vorher hier gezeichnete Linie wieder gelöscht werden soll.

Wenn die Line dann mit einer Geschwindigkeit von 1,5 Mio Punkten pro Sekunde gezeichnet ist, stehen automatisch in den X- und Y-Koordinatenregistern die Koordinaten X2 und Y2 der eben gezeichneten Linie, so daß von deren Endpunkt durch Vorgabe neuer Werte für Delta X und Delta Y gleich ein weiterer Vektor gezeichnet werden kann, ohne daß die Endpunktskoordinaten der vorherigen Linie noch als Anfangskoordinaten der neuen Linie ausgegeben werden müssen.

Dann hat der EF9366 noch einen eingebauten Zeichengenerator (der aber leider nur die amerikanischen Zeichen kennt:), mit dem ein komfortables Einbinden von Texten in die Graphik möglich ist. Außer Aufrecht- und Schrägschrift sowie in Breite und Höhe voneinander unabhängig in 16 Stufen in der Größe variierbaren Zeichen läßt sich vorgeben, ob normal von links nach rechts, auf dem Kopf stehend von rechts nach links oder um 90 Grad gedreht von unten nach oben oder von oben nach unten geschrieben werden soll. Für die Textausgabe ist vor der Ausgabe der Textzeichen neben dem Festlegen dieser Modi auch noch der Fußpunkt des ersten Zeichens in die Koordinatenregister einzutragen. Nach der Ausgabe eines Zeichens erhöht sich der Pointer auf den Fußpunkt des dahinter auszugebenden Zeichens, wobei die oben beschriebenen Modi natürlich berücksichtigt werden.

Diese Bemerkungen sollen keinesfalls als erschöpfende Beschreibung der Leistung der GDP64 zu werten sein. Für die Programmierung des EF9366 kommt man ohne Datenblatt nicht aus. Es sollte hier nur ein ungefährer Eindruck vermittelt werden, wofür der Aufwand überhaupt getrieben wurde, bei der unsicheren Erfolgsaussicht, den Hardware-Patch überhaupt anzufangen.


A
B
2k2
2k2
18k
18k
1k8
1k8
100 R
100 R
2x
BC 307
2x
BCY 59
o. ä.
Video Out
+5 V
GND
2x Um
A1
B1
Q0
IC 19
74LS273
Pin 19
13
12
74
A2
B2
A1
B1
A2
B2
A3
B3

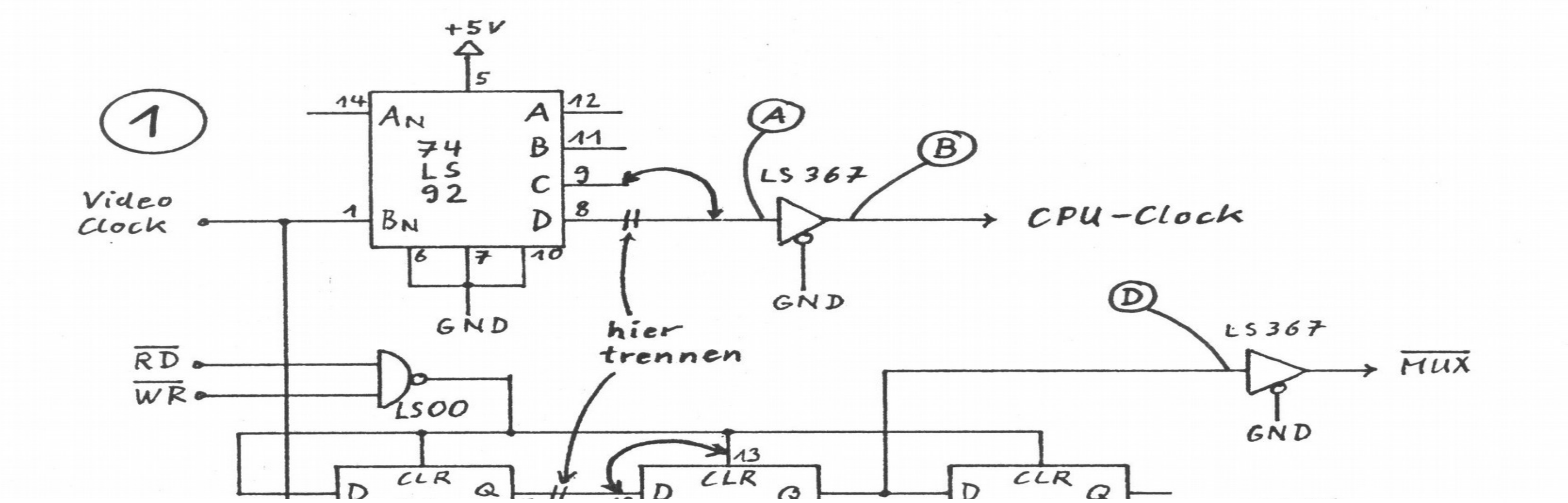
Noch'ne Geschindigkeitserhöhung

Helmut Bernhardt

In den letzten Jahren wurden verschiedene Möglichkeiten für Hardwarepatches zur Geschwindigkeitserhöhung bei TRS80 und GENIE vorgestellt. Allen Vorschlägen gemeinsam war, daß dies bis maximal 3,55 MHz (100% Speedup) möglich war. Voraussetzung war natürlich auch, daß 64K RAM auf dem CPU-Board vorhanden waren und daß ein Z80-A eingesetzt wird. Unterschiedlich war allerdings der Aufwand, der dafür getrieben wurde. Ein ehemaliges Mitglied war sogar der Meinung, daß der Erwerb eines Platinenschnippels mit 5 Allerwelt-ICs für den Gegenwert von 30 großen Bieren ein big Deal war, andere begnügten sich mit einfachem Umverdrahten am Frequenzteiler.

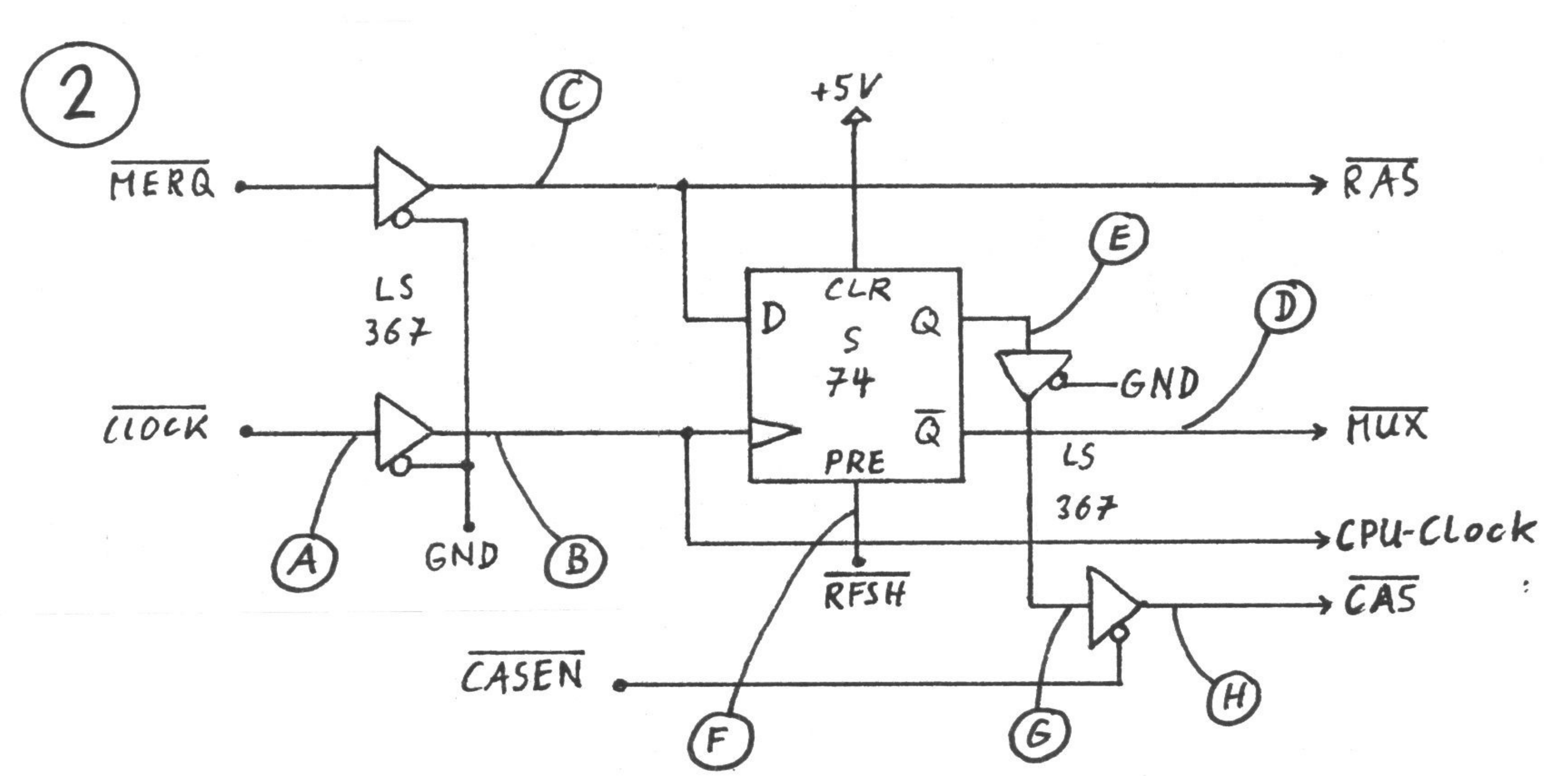
Wie auch immer vorgegangen wurde, bei 3,55 MHz war Schluß. Wenn man diese Grenze überwinden will, muß man vom vorgegebenen Prinzip der Erzeugung der RAM-Steuersignale RAS*, MUX* und CAS* aus dem Systemmuttertakt, der in seinem Wert wegen der Verwendung beim Video-Timing festgelegt ist, abrücken. Dieses Prinzip erzeugt aus dem Muttertakt und den Z80-Signalen RD* und WR* diese Signale über zwei Flip Flops. Wenn man aber am Frequenzteiler für den Z80 einen nicht so weit heruntergeteilten Takt abgreift und nicht gleichzeitig das Timing der RAM-Steuersignale ändert, läuft irgendwann beides nicht mehr synchron. Das Prinzip, wie beim TRS80 und GENIE diese Schaltungen aussehen und wie dort zur Geswindigkeitserhöhung eingegiffen werden kann, ist in Abb. 1 zu sehen.

Abb. 1: Erzeugung der Signale RAS*, MUX* und CAS*



Eine andere Art der Erzeugung der Signale RAS*, MUX* und CAS* hat uns TCS im GENIE IIs vorgeführt. Dort ist eine soft- (und/oder) hardgesteuerte Umschaltung zwischen 1,77 und 5,3 (6,0 oder 8,0) MHz möglich. Dafür konnte bei so stark voneinander abweichenden Geschwindigkeiten natürlich nicht mehr ein konstantes Timing der RAM-Steuersignale verwendet werden. Die RAM-Steuersignale werden dort aus dem jeweils eingestellten CPU-Tatkt und dem Z80-Signal MERQ* hergeleitet. Wegen der dabei möglichen hohen Geschwindigkeiten wird dafür ein 74S74 Flip Flop eingesetzt.

Abb. 2: und so im TRS80 oder GENIE I oder II



Diese Variante läßt sich natürlich auch im TRS80 oder GENIE I oder II verwenden (Abb. 2). Da hierbei sowohl CPU-Takt als auch Steuerung der RAMs aus dem gleichen Takt abgeleitet werden, kann dieser beliebig gewählt werden (solange die Hardware mitspielt). Ein solcher Takt kann entweder am Teiler 74LS92 auf dem CPU-Board abgegriffen werden oder über eine einfache Oszillator-Schaltung zusätzlich erzeugt werden.

Außer den bekannten Problemen mit RAMs im Expansions-Interface und einer flinken Variante der Z80-CPU spielt irgendwann auch die Zugriffszeit der ROMs eine Rolle. In beiden Computern stecken Exemplare, deren Zugriffszeiten so ungünstig wie nur möglich sind. Es gibt aber heute schon 32K-EPROMs (27256) mit 150ns Zugriffszeit und 16K-EPROMs (27128) mit 170ns Zugriffszeit. Wenn man sich den ROM-Inhalt in ein solches IC umfüllt, sollten bedeutend höhere Taktraten möglich sein. Bei Verwendung eines 27256-EPROMs ist durch Umschalten des Pegels am Adreßeingang A14 auch ein Wechsel zwi- sch Original Level 2 und einer aufgebohrten eigenen Schöpfung möglich.

Da diese EPROMS mit ihren 28 Pins aber nicht mehr in einen der bestehenden Sockel passen und beim TRS80 auch nicht pinkompatibel zu den dort verwendeten ROMs sind, müssen ein paar Leitungen durch die Luft dorthin verdrahtet werden. Außerdem müssen die Freigabe-Signale der bisherigen ROMs über AND-Verknüpfungen zu einem gemeinsamen Freigabe-Signal für das

neue EPROM zusammengefaßt werden.

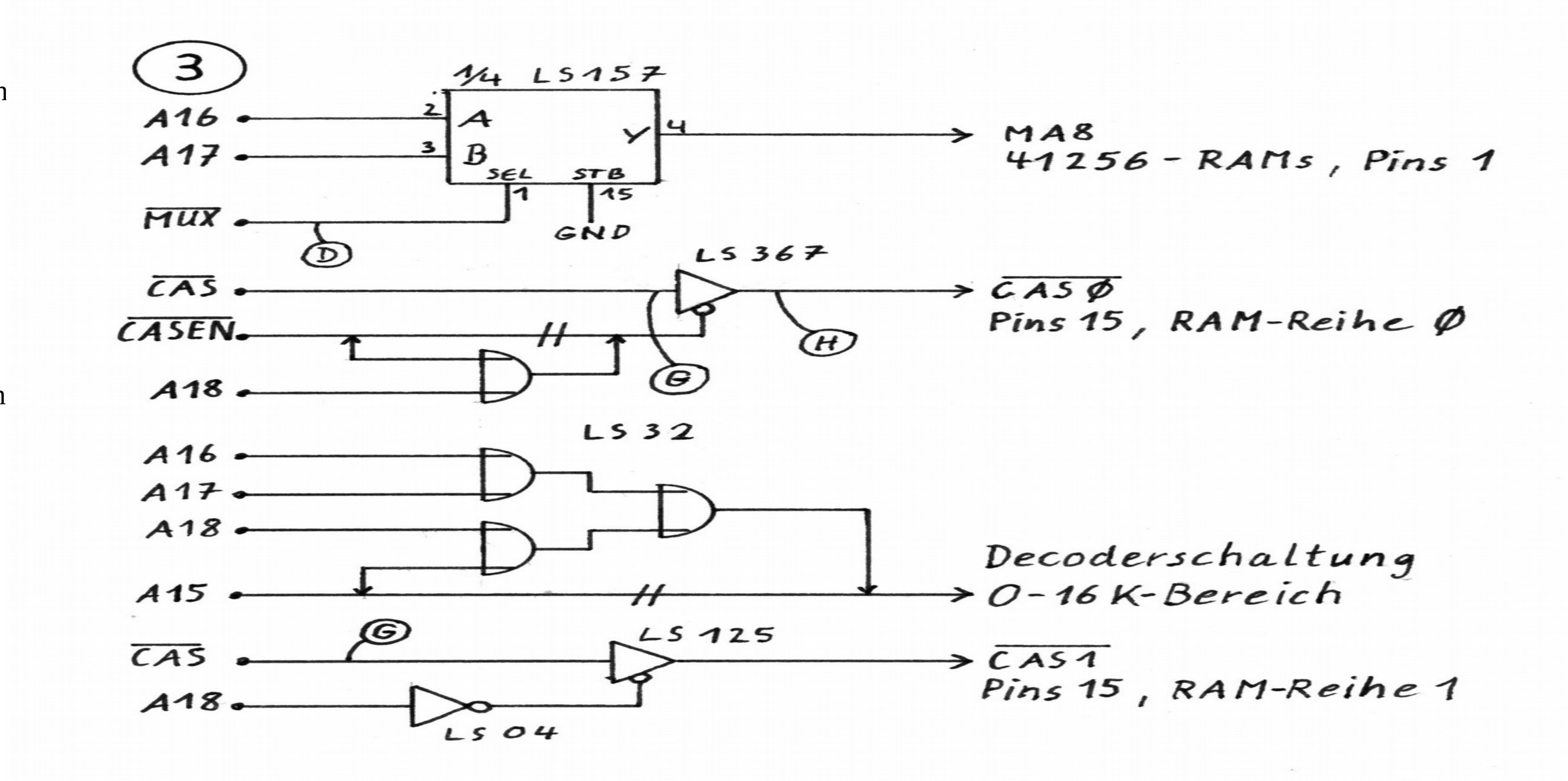
Beim Umändern auf die in Abb.2 dargestellte Variante braucht nur das 74S74 Flip Flop zusätzlich eingesetzt zu werden. Die Treiber aller Signale sind auch im ursprüglichen Zustand schon vorhanden und können weiterhin benutzt werden. Die folgende Tabelle gibt für TRS80 und GENIE I und -II die Punkte an, wo das zusätzliche Flip Flop anzuschließen ist.

Punkt	74S74	Signal	TRS80	GENIE I/II
	Pin		IC Typ Pin	IC Typ Pin
A	-	Clock	37 367 2 #1	72 367 12 #1
B	3	CPU-Clk	37 367 3	72 367 11
C	2	RAS*	16 367 5	72 367 5
D	6	MUX*	37 367 4 #2	72 367 2 #2
E	5	PCAS*	37 367 14 #2	72 367 10 #2
F	4	RFSH*	16 367 3	40 Z80 28

#1 hier kann entweder ein externer Oszillator oder eine aus dem Muttertakt über den 74LS92 heruntergeteilte Frequenz angelegt werden

#2 die bisher zu diesem Pin führende Leitung wird durch

X



Zum Einsatz einer HD64180-CPU ist es nur noch ein kleiner Schritt. Im Unterschied zum Z80 wird der HD64180 an den Pins 2 und 3 direkt mit einem Quartz beschaltet, der die doppelte Frequenz des beabsichtigten Systemtaktes liefert. Der HD64180 halbiert diese Frequenz intern und gibt dann diesen Systemtakt über Pin 64 aus. Systemtakt kann dann über Pin 6 der Z80-Fassung dem Computer zugefürt werden. Dabei muß natürlich die Verbindung zwischen Treiber und Pin 6 der Z80-Fassung durchtrennt werden, damit nicht alter und neuer Takt gegeneinander arbeiten. Der neue Takt braucht nur noch an das 74S74 Flip Flop und - soweit vorgesehen - auf den externen Systembus geführt zu werden.

Abb. 3: Anpassung für den HD64180

Im Zusammenhang mit dem HD64180 läßt sich natürlich auch noch der Speicher auf 256K oder 512K RAM erweitern. Die dabei anfallenden Änderungen gehören aber nicht mehr zum Thema und werden nur anhand von Abb. 3 kurz skizziert.

Obige Überlegungen sind für mich bislang nur Theorie. Für mich besteht im Moment nicht die Dringlichkeit, das in die Tat umzusetzen und andere Dinge dadurch zu verschieben. Ich kann also keinen Erfolg bei diesen Umbauten garantieren und auch nicht sagen, welcher Computer sich wie weit scheuchen läßt.

Vielleit ist ja irgendjemand von euch etwas mutiger und sieht dem Projekt auch einen höheren Stellenwert ab. Es würde mich freuen, wenn dieser Denkanstoß irgendwo eine Hilfe sein kann. Die beschriebenen Maßnahmen sind aber auch Vorausetzung, um in diesen Computern einen HD64180 fahren zu können und deshalb auch für mich wieder interessant (worin die Ursache dieses Beitrags liegt).

Selbstbau Doubler für EXP1

Helmut Bernhardt

Nachdem sich bei mir nun schon der zweite WD1771 Single Density Floppy- Controller verabschiedet hat und der Chip wohl auch bald nicht mehr zu haben ist, entschloß ich mich, mir selbst einen Doubler zu bauen der ohne diesen Käfer auskommt. Es ist auch nicht einzusehen, warum auf einem Doubler zwei Floppycontroller sitzen müssen, wo doch der WD1791 alles das, was der WD1771 kann, (nämlich beim Booten die erste Spur der Diskette in Single Density und gelegentlich auch mal ganze Disketten in SD zu lesen oder zu schreiben (wenn ein unterausgerüsteter Zeitgenosse den digitalen Informationsaustausch nur in diesem Format leisten kann) nebenbei auch erledigen könnte.

Letzteres Argument soll nun aber auch gegenstandslos werden, denn den hier beschriebenen Doubler kann sich leicht jeder selbst bauen und auch finanzieren. Grundlage ist der WD2793 Floppycontroller, der schon aufgrund seiner Bescheidenheit bei der Spannungsversorgung (ausschließlich +5V und GND) mein Wohlwollen verdient hat. Außerdem enthält er auf dem Chip schon einen PLL-Datenseparator und eine Schreibvorcompensation. Dadurch können die meisten ICs auf den Doublern von TCS und dem davon abgekupferten von RB entfallen, die nur diesen Job leisten.

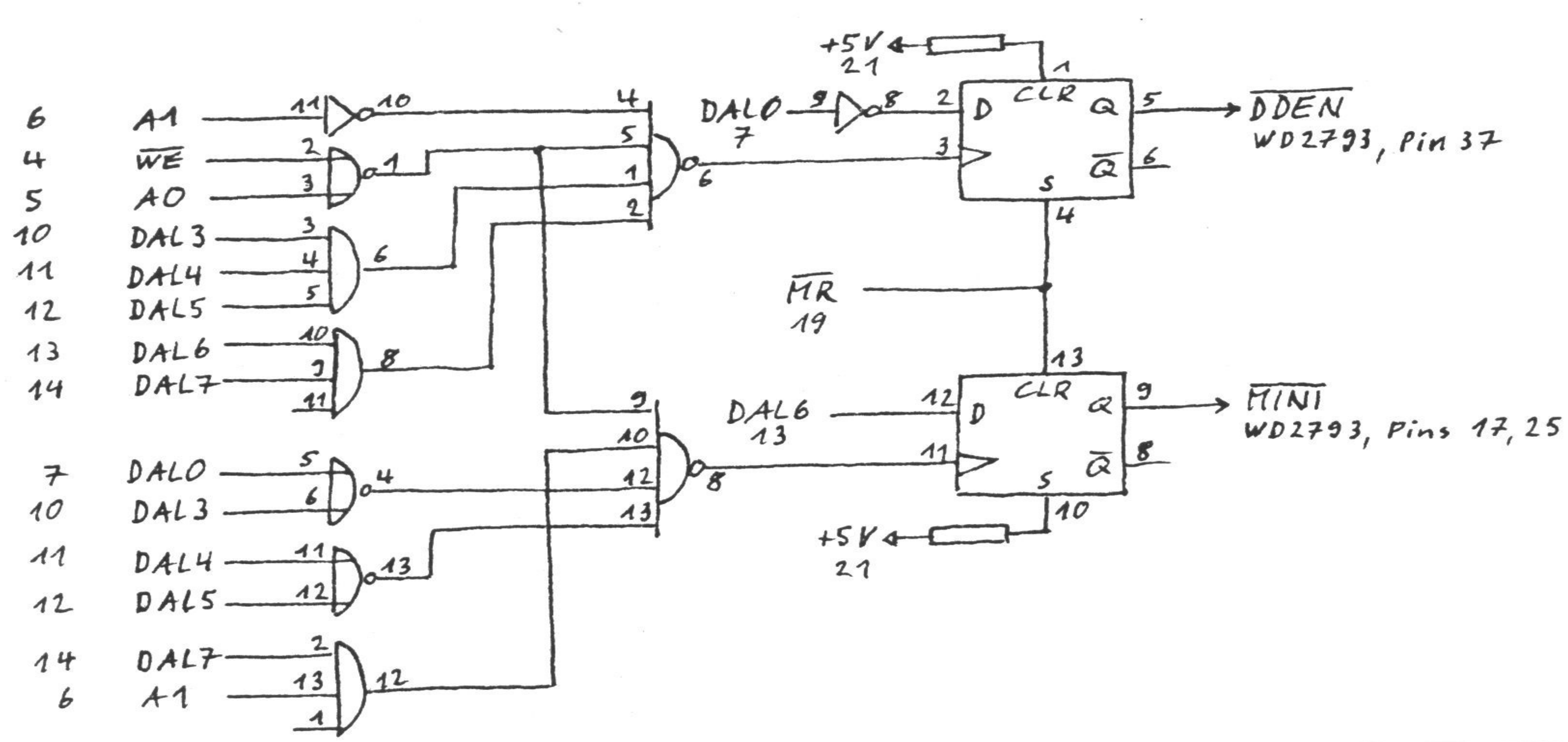
Der WD2793 kann intern den an Pin 24 angelegten Takt von 2MHz (für den Betrieb von 8" Floppies) auf 1MHz (5" Floppies) runterteilen, wenn an den Pins 17 und 23 Low-Pegel angelegt wird. Über Pin 37 läßt sich vorgeben, ob in Single Density (high) oder Double Density (low) gelesen und geschrieben werden soll.

Durch die 5 zusätzlichen ICs links neben dem WD2793 auf dem Doubler Board lassen sich diese Features programmieren. Zwei Flip Flops geben über ihre Q-Ausgänge die Pegel für die Pins 17, 25 und 37 vor. Nach einem RESET ist der Floppycontroller auf Minifloppy und Single Density eingestellt. Durch Ausgabe folgender Werte an das Contollregister mit der memory mapped Adresse 37ECH bzw. das Sektorregister 37EEH lassen sich die Betriebszustände folgendermaßen einstellen:

37ECH:	1111 1xx0 bin (FFH)	Single Density
	1111 1xx1 bin (FEH)	Double Densitv
37EEH:	1000 0xx0 bin (80H)	Minifloppy (5 1/4")
	1100 0xx0 bin (C0H)	Maxifloppy (8")

Diese Daten haben im Control- bzw. Sektorregister keine Wirkung. Aus den Daten und der Schreibadresse werden nur extern die Triggersignale für die Flip Flops decodiert, wobei für die Density D0 und für die Drivegröße D6 den einzustellenden Pegel vorgibt.

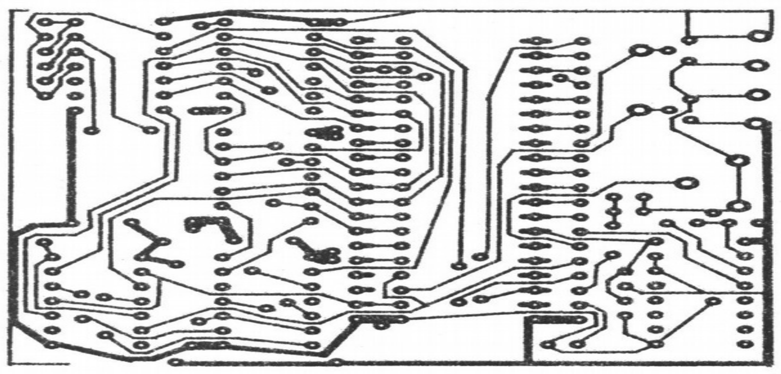
Schaltplan

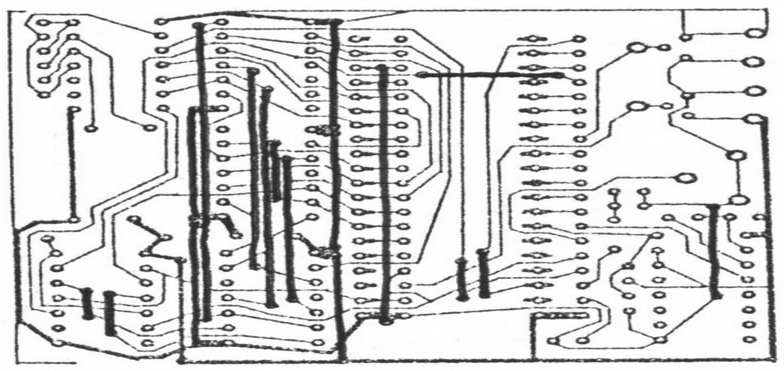


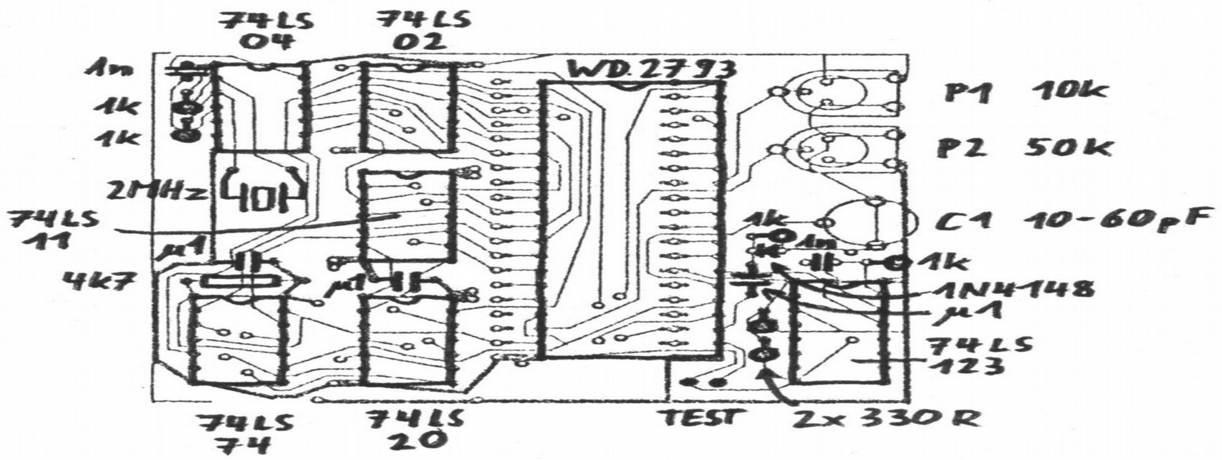
Diese Schaltung ist in Abb.1 gezeigt. Das einseitige Layout und die Anordnung der Drahtbrücken und Bauteile ist aus den folgenden Abbildungen zu ersehen. Die zur Position des WD2793 leicht versetzte Dil40-Anordnung von Lötpunkten wird von der Lötseite mit einem entsprechenden Stecker bestückt, der in den Sockel des WD1771 auf dem EXP1 gesteckt wird. Zur Zugentlastung dieses Steckers sollten an ca. 15-20 der durchragenden Pins auf der Bestückungsseite kurze Drahtstücke angelötet werden. Das verhindert ein eventuelles Abreißen der Leiterbahnen beim Herausziehen des Doulers aus dem WD1771-Sockel.

Beim Einsatz dieses Doublers muß der invertierende Datentreiber 74LS640 neben dem 1771-Sockel durch den nichtinvertierenden, pinkompatiblen Typ 74LS245 ersetzt werden, da der WD2793 keine invertierenden Daten-Ein-/Ausgänge (wie WD1771 und WD1791) hat. Die Entscheidung fiel für den WD2793, der sich nur in seinen nichtinvertierenden Dateneingängen vom WDD2791 unterscheidet, wegen unüberwindlicher Beschaffungsschwierigeiten beim WD2791. Das bedeutet aber leider auch, daß dieser Doubler bei anderen Floppycontroller-Boards nicht verwendet werden kann, da dort nicht einfach ein Treiber ausgewechselt werden kann.

Layout; Drahtbrücken; Bestückung







Auch ein weiterer Nachteil dieses Doublers soll noch erwähnt werden. Für die einwandfreie Funktion von Datenseparator und Schreibvorkompensation sind einige Trimmungen an den Potentiometern P1 und P2 und dem Trimmcondensator C1 vorzunehmen. Dazu ist ein Oszilloskop nötig. Wenn das nicht verfügbar ist, kann im Notfall auch mit Probieren getrimmt werden. Dann werden P1, P2 und C1 auf Mittelstellung gedreht und durch wiederholte Bootversuche und ständiges Verändern der Einstellung von P2 und C1 die günstigste Einstellung für das Lesen einer Diskette gefunden. Mit P1 muß dann noch die Schreibvorkompensation geregelt werden, was durch Schreiben auf Diskette und anschließendes Lesen geprüft werden kann. Die Einstellungen sind nicht sehr kritisch, so daß auch dieses Vorgehen möglich ist.

Die im Datenblatt vorgegebene Einstellung mit Hilfe eines Oszilloskops läuft folgendermaßen ab:

Der Comuter wird mit gedrückter "BREAK" Taste eingeschaltet, um im Level II BASIC zu landen. Es wird der Befehl POKE 14316,255 (auf DD schalten) ausgeführt. Dann werden mit einem Jumper die beiden Stifte "TEST" kurzgeschlossen (Pin 21 des WD2793 an GND). Mit P1 kann dann die Breite der Pulse an Pin 31 (WD) unter Oscarkontrolle auf einen Wert zwischen 100 und 300ns eingestellt werden.

Mit P2 wird dann die Pulsbreite an Pin 29 (TG43) auf 60-65ns eingestellt. Mit C1 wird an Pin 16 (DIRC) eine Frequenz von 250kHz eingestellt.

Um die Hardware-Voraussetzungen, auch 8" Floppies betreiben zu können, zu nutzen, muß die Anordnung der Signale auf dem 34poligen Shugart-Bus für Minifloppies auf die für den 50poligen Bus für 8" Floppies umsortiert werden. Bei älteren 8" Drives muß diesen zusätzlich noch das Signal TG43 (WD2793, Pin 29) über einen Open Collector Inverter zugeführt werden. Neuere 8" Drives benötigen dieses Signal nicht mehr.

Wer sich diesen Doubler aufbauen will, kann für 10 DM (incl. Rückporto) von mir ein geätztes, ungebohrtes Board bekommen und darf sich mit der Beschaffung der Bauteile, dem Bohren, Löten und Trimmen selbst abmühen.

Steckerbelegung Floppy-Kabel

N3: Minifloppy-Stecker

Belegung:

01,02:	GND	/READY,/MINI
03,04:	GND	/HDLD
05,06:	GND	/DS3
07,08:	GND	/INDEX
09,10:	GND	/DS0
11,12:	GND	/DS1
13,14:	GND	/DS2
15,16:	GND	/MOTOR ON
17,18:	GND	/DIR
19,20:	GND	/STEP
21,22:	GND	/WR DATA
23,24:	GND	/WR GATE
25,26:	GND	/TRACK 0
27,28:	GND	/WR PT
29,30:	GND	/RD DATA
31,32:	GND	/SIDE
33,34:	GND	/READY,/DCHG

Funktion:

GND:	Signalmasse
/DS0-3:	Laufwerkauswahl
/HDLD:	Kopf laden
/MOTOR ON:	Motor ein
/DIR:	Spurwechsel-Richtung
/STEP:	Spurwechsel-Impuls
/WR DATA:	Schreibdaten
/WR GATE:	Schreiben ein
/SIDE:	Kopfauswahl
/MINI:	Mini/Maxi-Umschaltung
/READY:	Laufwerk bereit
/INDEX:	Indexloch-Impuls
/TRACK 0:	Kopf auf Spur 0
/WR PT:	Schreibschutz ein
/RD DATA:	Lesedaten
/DCHG:	Diskettenwechsel

N4: Maxifloppy-Stecker

Belegung:

01,02:	GND	/LOW CUR
03,04:	GND	/MOTOR OFF
05,06:	GND	--
07,08:	GND	--
09,10:	GND	/TWO SIDED
11,12:	GND	/DCHG
13,14:	GND	/SIDE
15,16:	GND	--
17,18:	GND	/HDLD
19,20:	GND	/INDEX
21,22:	GND	/READY
23,24:	GND	--
25,26:	GND	/DS0
27,28:	GND	/DS1
29,30:	GND	/DS2
31,32:	GND	/DS3
33,34:	GND	/DIR
35,36:	GND	/STEP
37,38:	GND	/WR DATA
39,40:	GND	/WR GATE
41,42:	GND	/TRACK 0
43,44:	GND	/WR PT
45,46:	GND	/RD DATA
47,48:	GND	--
49,50:	GND	--

Funktion:

GND:	Signalmasse
/DS0-3:	Laufwerkauswahl
/HDLD:	Kopf laden
/MOTOR OFF:	Motor aus
/DIR:	Spurwechsel-Richtung
/STEP:	Spurwechsel-Impuls
/WR DATA:	Schreibdaten
/WR GATE:	Schreiben ein
/SIDE:	Kopfauswahl
/LOW CUR:	Schreibstrom niedrig
/READY:	Laufwerk bereit
/INDEX:	Indexloch-Impuls
/TRACK 0:	Kopf auf Spur 0
/WR PT:	Schreibschutz ein
/RD DATA:	Lesedaten
/DCHG:	Diskettenwechsel
/TWO SIDED:	Zweiseitige Diskette
--	- frei

1987-BHV-03 S.7 / Club80 Nr.19, S.49

Nachtrag: Verbessertes Datenlesen

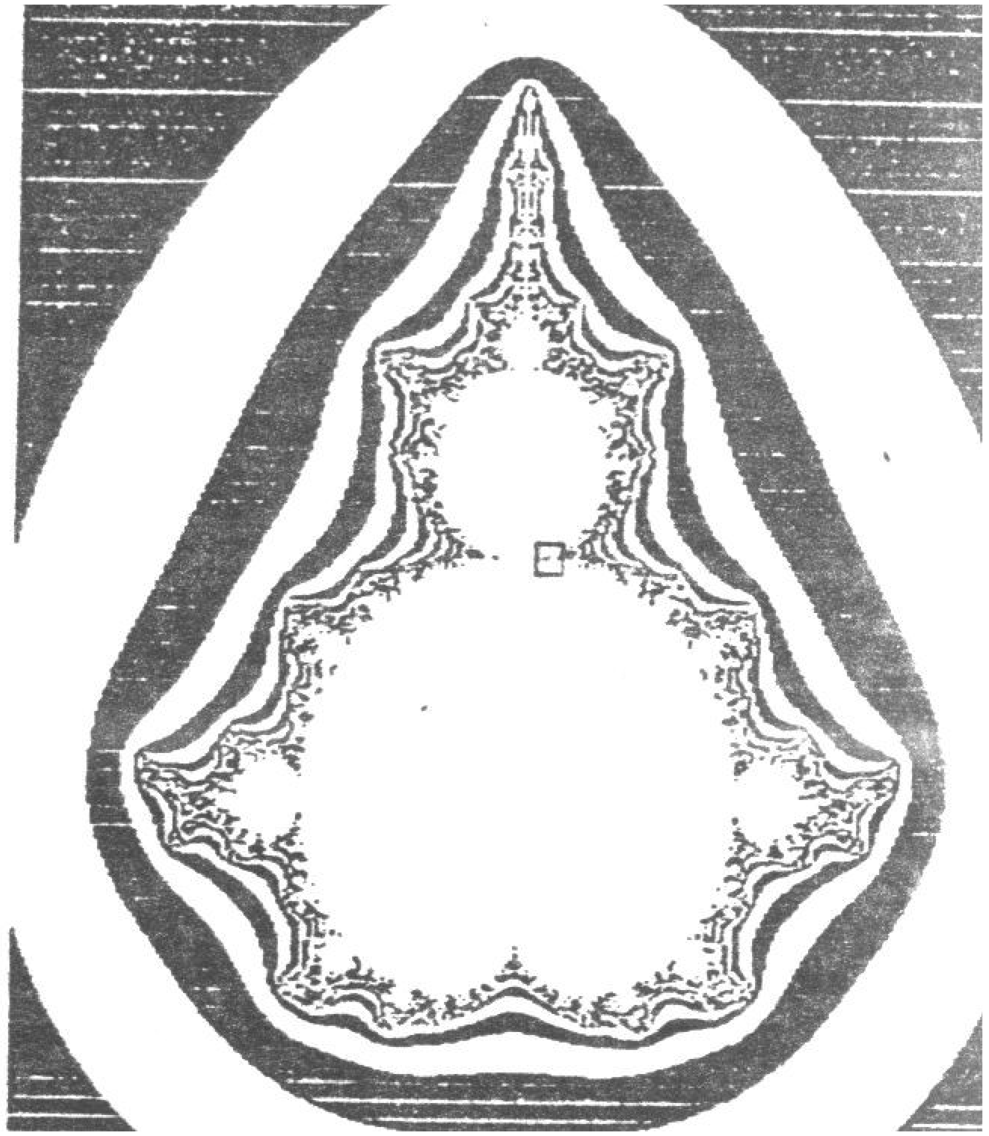
Ein zuverlässigeres Lesen der Daten von Diskette mit dem vor kurzen beschriebenen Selbstbau-Doubler auf Basis des WD2793 läßt sich durch folgende kleine Änderung erreichen.

1) Auf dem Doubler Board wird die Verbindung zwischen Pin 13 des 74LS123 und Pin 27 des WD2793 durchtrennt. Der Pin 27 des WD2793 wird stattdessen mit Pin 4 des 74LS123 verbunden.

2) Auf dem EXP1 Board werden die Pins 13 und 12 des 74LS04 aus der Fassung gebogen bzw. durchtrennt (wenn das IC direkt eingelötet ist). Auf der Lötseite werden die Pins 13 und 12 des 74LS04 miteinander verbunden.

Durch diese Änderung wird die Gatterlaufzeit der überbrückten Inverter eingespart. Die Fehlerquote beim Lesen von der Diskette wird damit drastisch gesenkt.

Das Layout des Doubles ist dem unter Punkt 1) beschriebenen Patch angepaßt. Das Board wird in Zukunft nur noch in dieser Form geätzt. Der Patch gemäß Punkt 2) ist dadurch zwingend.

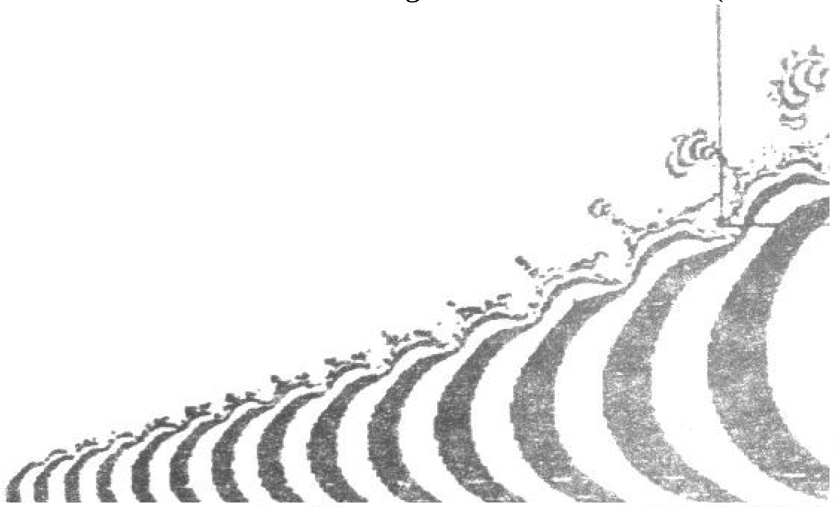


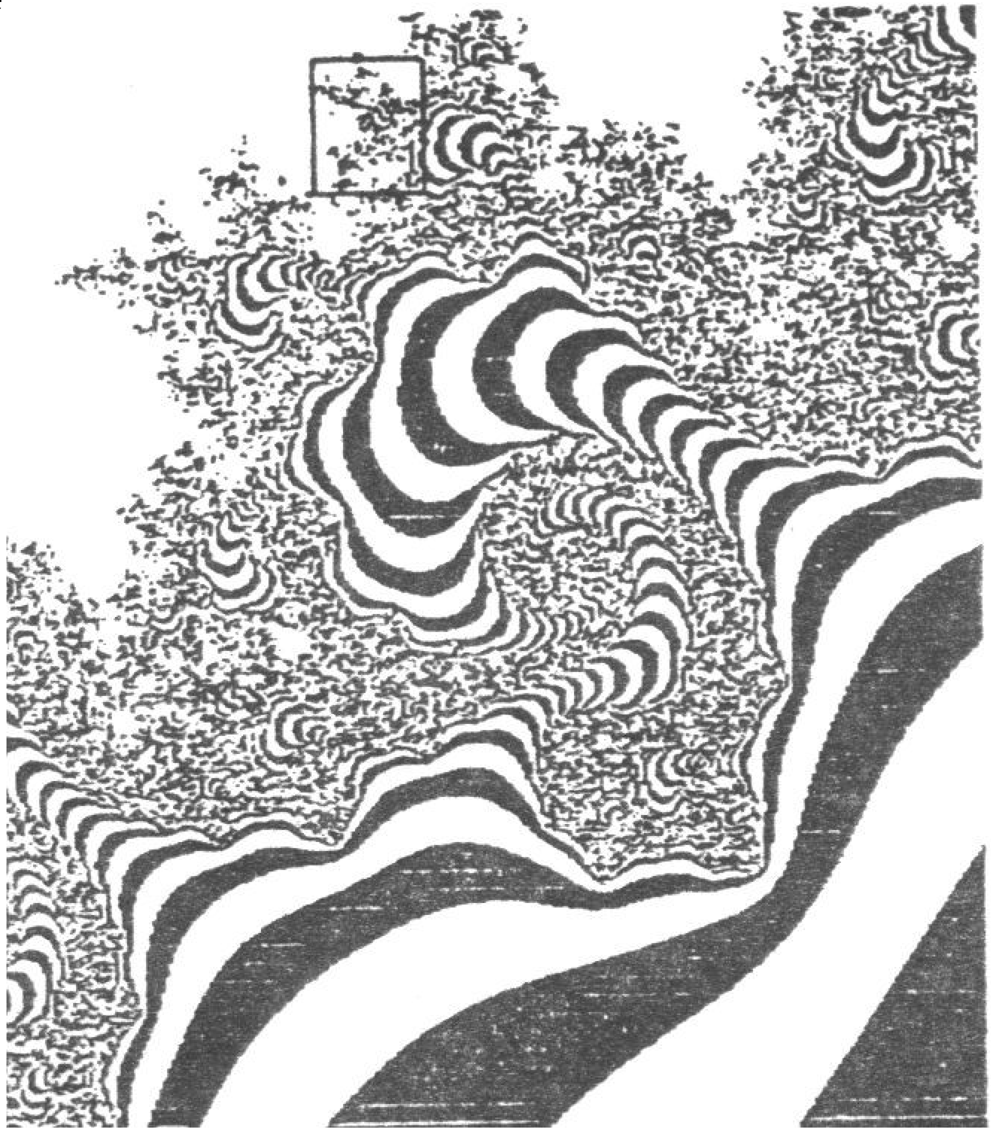
Apfelmännchen auf der GDP64

Helmut Bernhardt

Da der Computer für mich zu 80% ein Spielzeug ist, weil ich keine elektrische Eisenbahn habe, an der ich rumbasteln kann, kommt es recht selten vor, daß ich damit ernsthafte Arbeiten erledige. Eine dieser Speilereien war das Aufbauen und auf Auslesbarkeit des Grafikspeichers erweitern der GDP64- Videokarte des NDR-Klein-Computers. Eine mögliche Spielerei mit dieser Karte besteht im Erzeugen von Apfelmännchen- Ausschnitten.

Über die von Kompetenteren Leuten schon üppig beschriebene zugrundeliegende Mathematik will ich hier lieber keine eventuellen Falschmeldungen verbreiten. Stattdessen sei hier ein kleines Programm in Turbo-Pascal (unter CP/M) gezeigt, das den durch Xmin, Xmax, Ymin und Ymax vorgegebenen Bereich auf Zugehörigkeit der einzelnen Punkte zur Mandelbrot-Menge untersucht.





Ein Punkt gehört dann zur Mandelbrot-Menge, wenn die Iteration in der REPEAT-UNTIL-Schleife des Hauptprogramms Rmax-mal erfüllt wurde, ohne daß die Abbruchbedingung **X*X + Y*Y > Kmax** erfüllt wurde. In diesem Fall wird der durch Xi und Yi vorgegebebe Punkt nicht gesetzt.

Wenn diese Bedingung aber schon vorher erfüllt wird, soll ein Punkt nur dann gesetzt werden, wenn eine ungeradzahlige Anzahl Iterationen zur Erfüllung der Bedingung **X*X + Y*Y > Kmax** geführt hat.

Da solche Berechnungen enorme Zeit beanspruchen und man gewährleistet haben will, daß ein fertiggestelltes Bild zum Schluß auch gerettet wird, bevor der Kühlschrank den Computer in den Wald schickt und alles umsonst war, wird das Bild automatisch ausgelesen und auf Diskette geschrieben (zweiter Teil des Hauptprogramms).

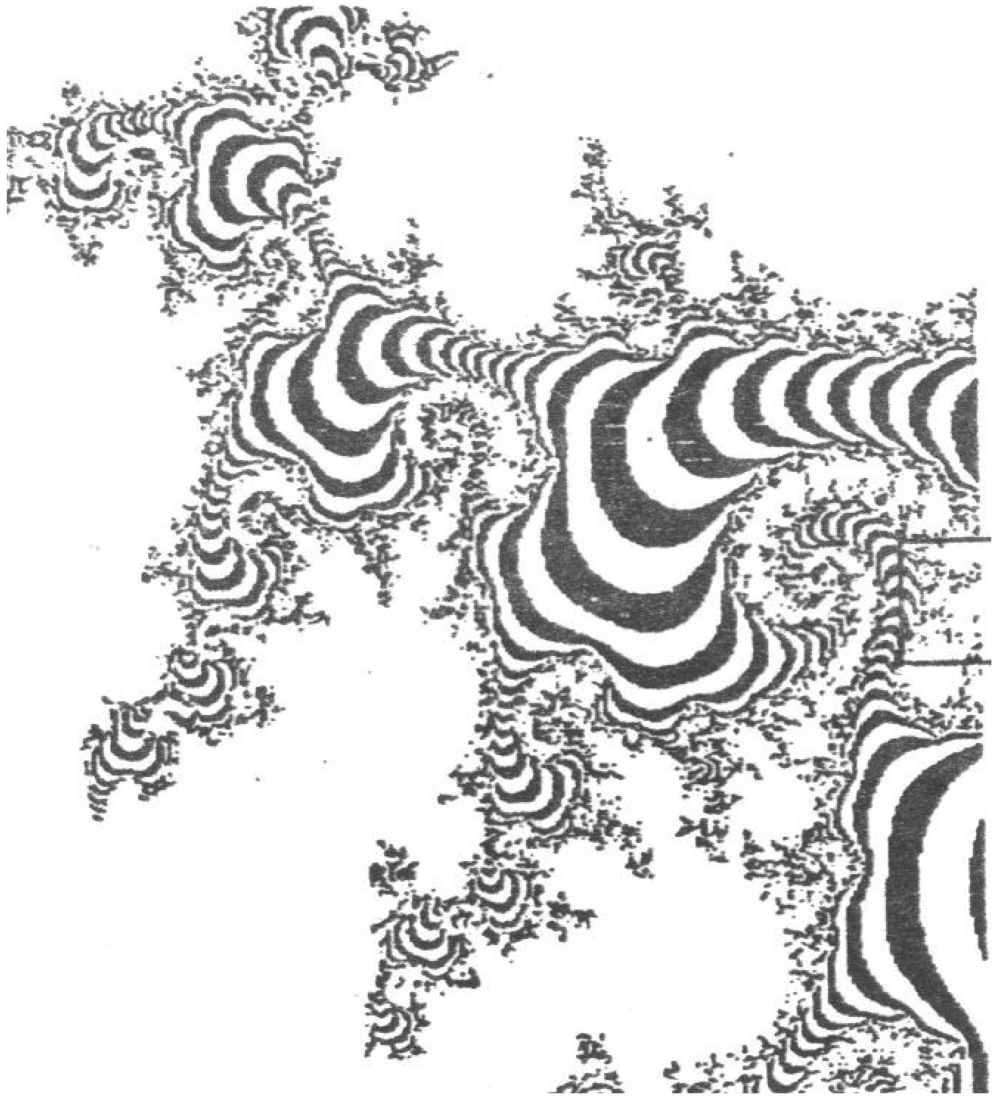
Natürlich läßt sich die relativ banale Rechnerei auch in BASIC unter NEWDOS/80 erledigen. Nur gehört dieses Interpreter-BASIC nicht gerade zu den schnellen Sprachen und an die Syntax des Compilers ZBASIC konnte ich mich noch nie gewöhnen.

Die für den Umgang mit Reals auf

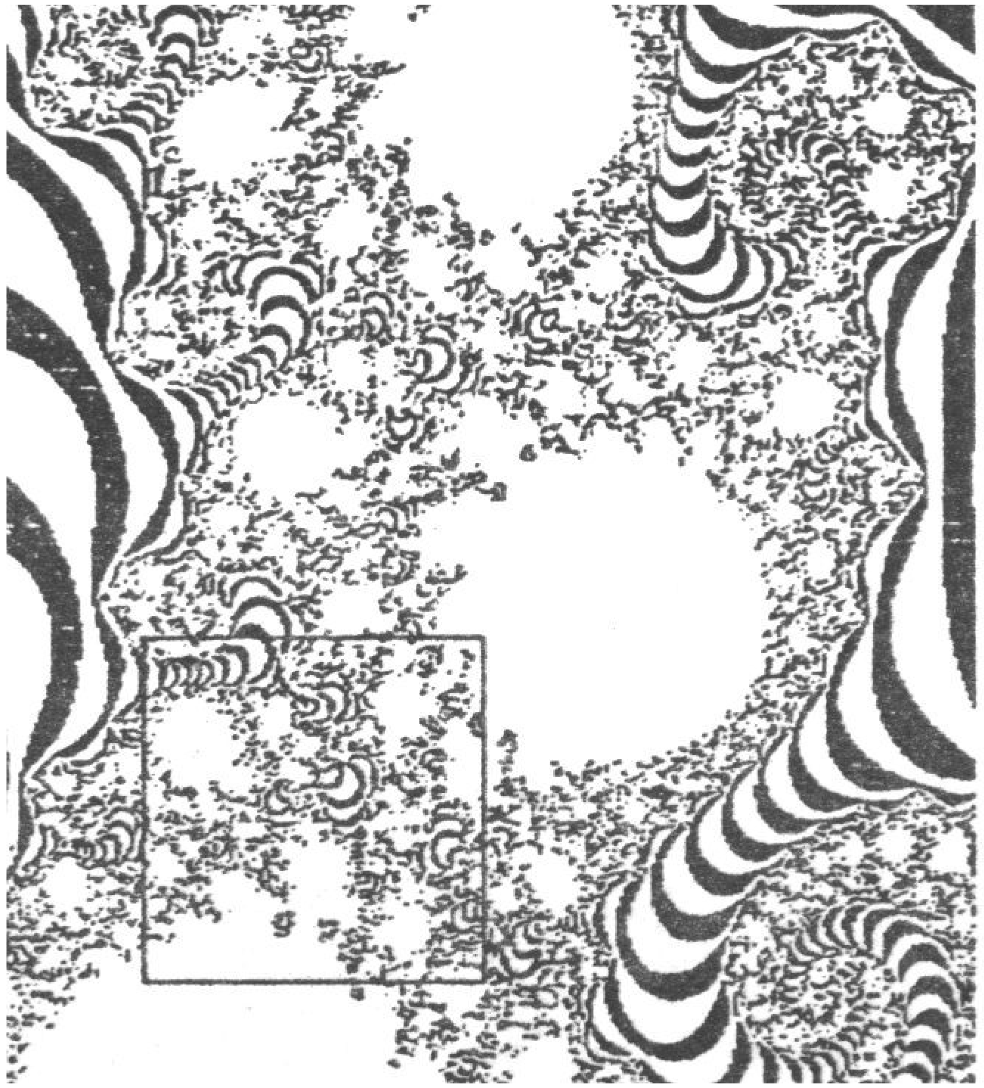
Z80-Rechnern wohl schnellste Sprache Turbo-Pascal benötigte mit einem 9,216MHz laufenden HD64180 dann auch noch volle 4 Tage zur Erzeugung des größten wiedergegebenen Bildes, das mit einem entspechend abgewandelten Programm auf der GRIP-5 mit 768*560 Bildpunkten errechnet wurde. Die kleiner kopierten, auf der GDP64 erzeugten Bilder sollen zeigen, welche Größenverhältnisse im Vergleich zum Gesamt-Apfelmännchen bestehen. Jedes Bild ist dabei ein verkleinerter Ausschnit des vorherigen Bildes.

Die Parameter der einzelnen Bilder sind

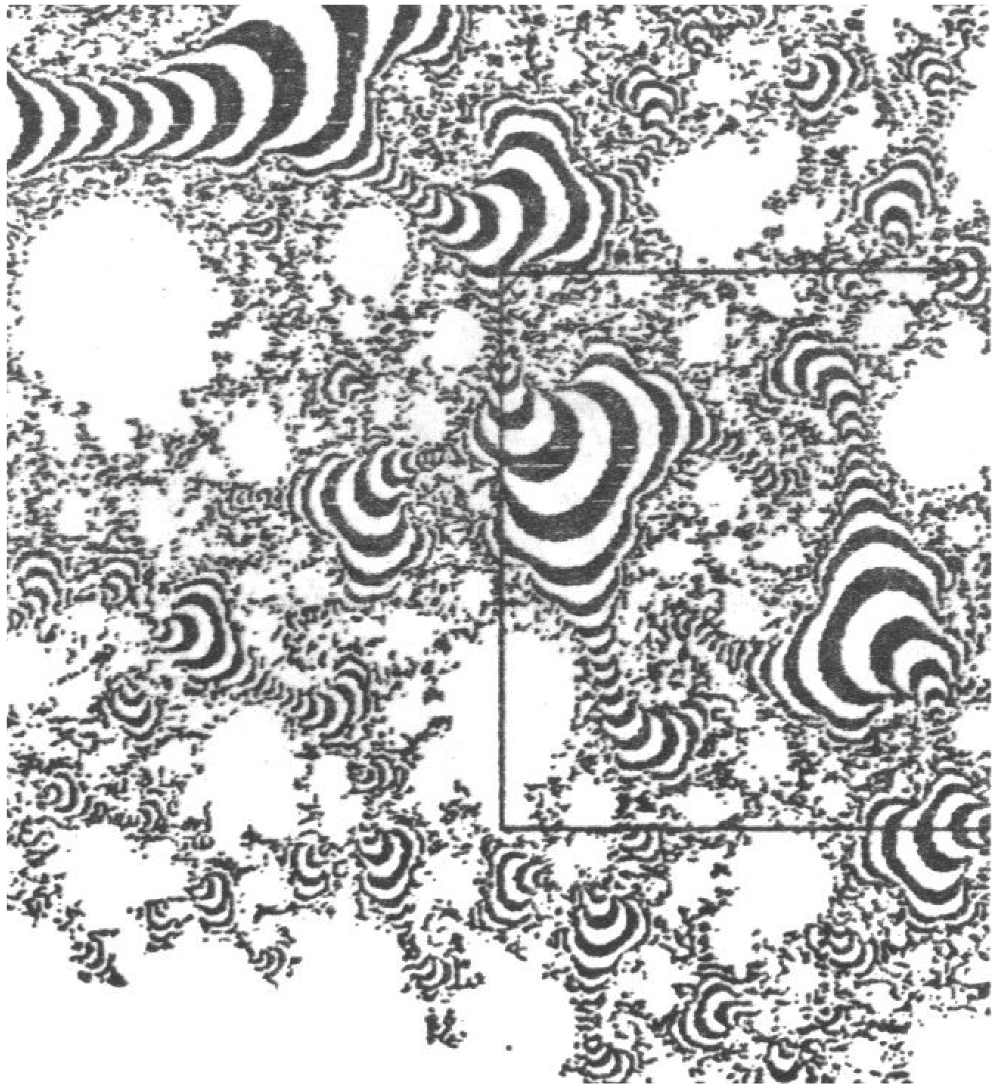
Xmin	Xmax	Ymin	Ymax	Kmax
-0,2	0,5	-1,25	1,25	50
-0,8	-0,7	0,05	0,015	50
-0,795	-0,775	0,135	0,152	200
-0,795	-0,794	0,1434	0,1444	200
-0,79452	-0,79440	0,14432	0,14444	250
-0,79445	-0,79441	0,14434	0,14438	300
-0,79445	-0,79443	0,14435	0,14437	300



```
(* Listing of MANDEL.PAS *)
CONST
  (* fuer die Apfelmänner Berechnung *)
  Xmin  = -0.79445;
  Xmax  = -0.79441;
  Ymin  = 0.14434;
  Ymax  = 0.14438;
  Kmax  = 300;
  fname = "aman18.gra";
  ak = 2;
  Rmax = 50;
  (* fuer die HRG GDP64 *)
  Xhrg  = 512;
  Yhrd  = 256;
  cmd   = $70;
  ctrl1 = $71,
  xmsb  = $78;
  xlsb  = $79;
  ymsb  = $7A;
  ylsb  = $7B;
  hrgrd = $60;
  speicher = 256;
var
  puffer : ARRAY [1..speicher]
OF byte;
  ib, A, B, I, J : byte;
  f : FILE;
  Xn, Yn, Xr, K : integer;
  Xi, Yi, dX, dY, p, q, X,
Xalt, Y : real;
PROCEDURE ready;
var
  status : byte;
BEGIN
  REPEAT
     status := port[cmd];
  UNTIL ( status AND 4) = 4;
END;
PROCEDURE setx (xp : integer);
BEGIN
   ready;
   port[xmsb] := Xp DIV 256;
   port[xlsb] := Xp MOD 256;
END;
PROCEDURE sety (Yp : integer);
BEGIN
   ready;
   port[ymsb] := Yp DIV 256;
   port[ylsb] := Yp MOD 256;
END;
PROCEDURE dot ( V : integer );
BEGIN
   ready;
   IF V = 0 THEN port[cmd] :=
1 ELSE port[cmd] := 0;
   ready;
```

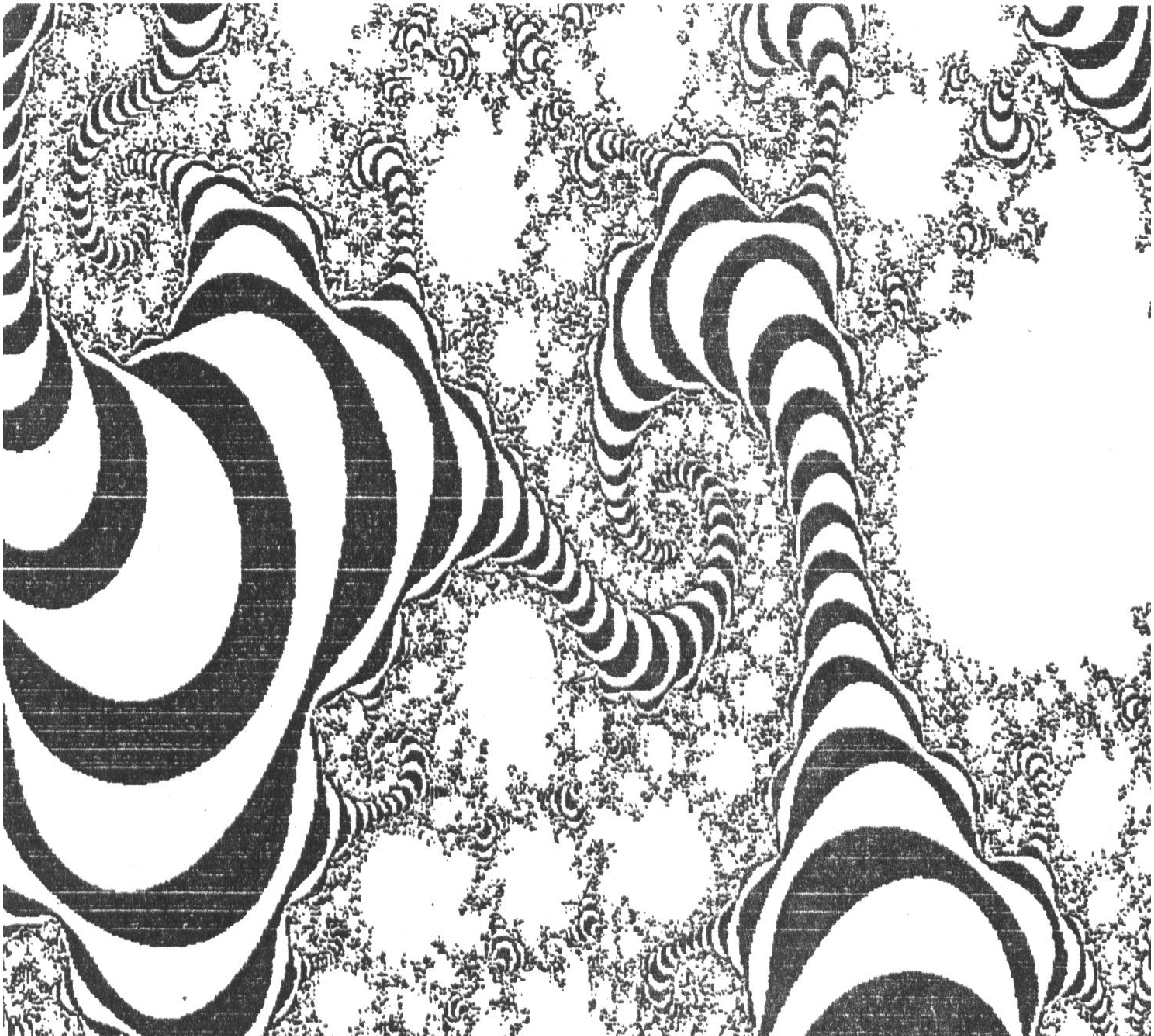


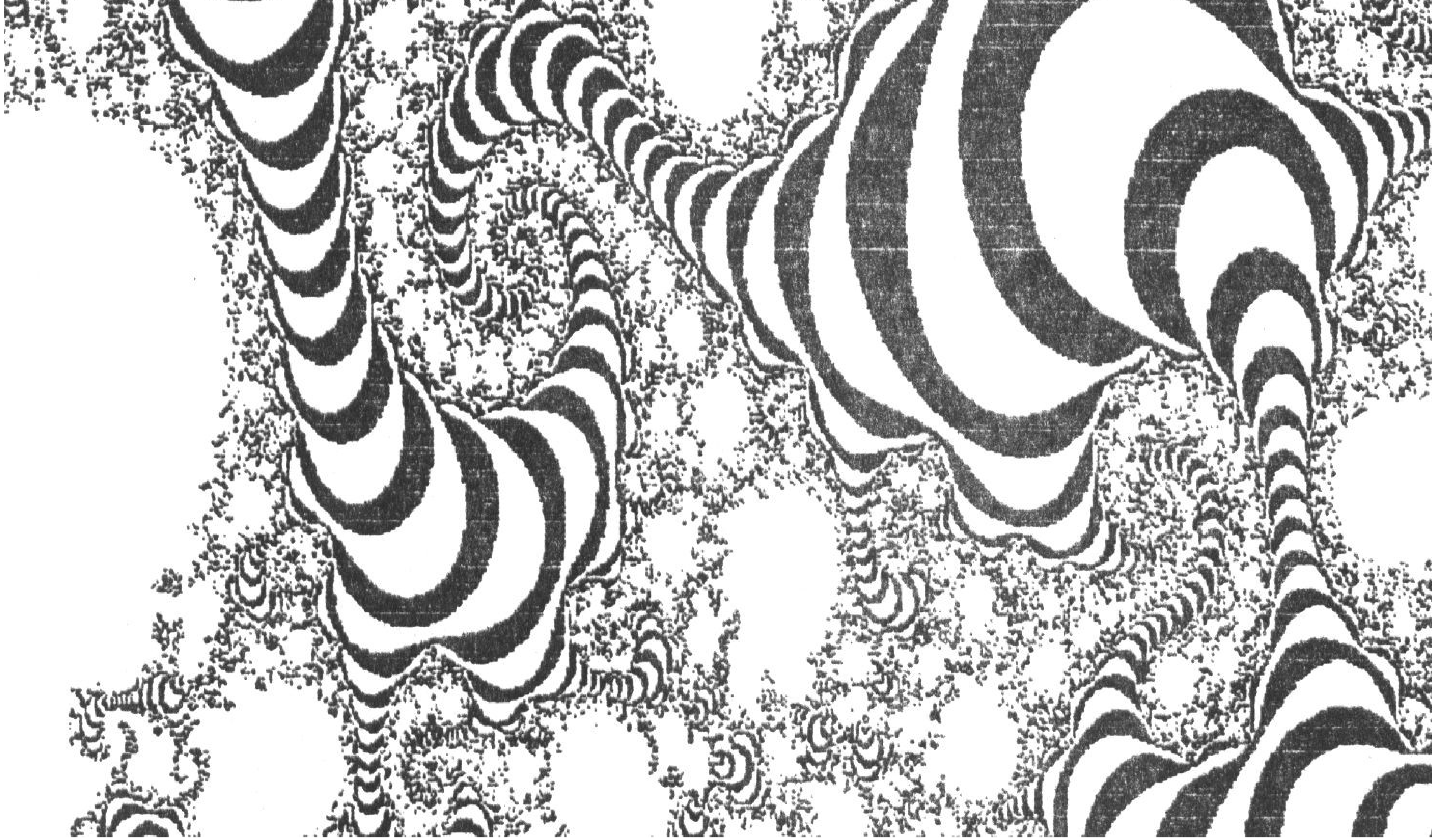
```
   port[cmd] := 128;
END;
PROCEDURE clh;
BEGIN
   ready;
   port[cmd] := 7;
   ready;
   port[cmd] := 33;
END;
BEGIN   (* Hauptprogramm *)
   clh;
   dX := ( Xmax - Xmin ) / Xhrg;
   dY := ( Ymax - Ymin ) / Yhrg;
   FOR Xn := 0 TO Xhrg - 1 DO
   BEGIN  (* Apfelmänner-Berechnung *)
      Xi := Xmin + dX * Xn;
      setx (Xn);
      FOR Yn := 0 To Yhrg -1 DO
      BEGIN
         Y1 := Ymin + dY * Yn;
         K := 0;
         X := 0;
         Y := 0;
         REPEAT
            Xalt:= X;
            X := X * X - Y * Y + Xi;
            Y := 2 * Xalt * Y + Xi;
            K := K + 1;
         UNTIL (X * X + Y * Y > Rmax) OR ( K = Kmax );
         IF K = Kmax THEN K := 0 );
         sety (Yn );
         dot ( K MOD ak );
      End;
   END;
   BEGIN  (* Abspeichern der HRG auf Disk *)
      assign ( f , fname );
      rewrite ( f );
      For a := 0 TO 63 DO
      BEGIN
         Xr := A * B;
         sety ( B );
         FOR B := 0 TO 255 DO
            BEGIN
               sety ( B );
               REPEAT
                  port[cmd] := 15;
                  ready;
                  port[cmd] := 15;
                  J := port[hrgrd];
               UNTIL I = J;
               puffer [ B+1 ] := I;
            END;
            blockwrite (f, puffer, 2);
         END;
         close ( f );
         writeln ( 'Bild fertig und abgespeichert!' );
      END;
   END.
; Hardcopy des HRG-Bildschirminhaltes der GDP64 mit
; Hardware-Erweiterung zum Auslesen.
; Ausdrucken über EPSON MX80 mit Typ 3 ROMs
;
CMD     EQU     70H             ;Ports der GDP64
CTRL1   EQU     CMD+1
CTRL2   EQU     CMD+2
CSIZE   EQU     CMD+3
DELX    EQU     CMD+5
DELY    EQU     CMD+7
XMSB    EQU     CMD+8
XLSB    EQU     CMD+9
YMSB    EQU     CMD+10
```



```
YLSB    EQU     CMD+11
MFREE   EQU     60H             ;Auslesen nur mit
;                                Hardware-Patch
        LD      A,1BH           ;ESC,A,0BH  (8/72" Linespacing)
        CALL    PRINT           ;Zeichen an Drucker
        LD      A,41H
        CALL    PRINT
        LD      A,0BH
        CALL    PRINT
        XOR     A
        OUT     (YMSB),A        ;YMSB immer 0
        LD      B,1             ;Zähler für Bildschirmhälften
B35     LD      A,0FFH          ;Zweite Hälfte fertig?
        CP      B
        JR      NZ,WEITER       ;nein: dann nächste 8 Dots
        LD      A,0CH           ;sonst <cr> an Drucker
        CALL    PRINT           ;und
        JP      0000H           ;WBOOT, wenn fertig
;
WEITER  LD      A,B             ;1 oder 0 in B an XMSB
        OUT     (XMSB),A        ;für zweite oder erste Hälfte
        LD      C,0F8H          ;rechter Rand minus 8 Dots
B40     LD      A,1BH           ;ESC, L, 00H, 03h an Drucker
        CALL    PRINT           ;768 Grafik-Bytes folgen
        LD      a,'L'
        CALL    PRINT
        XOR     A
        CALL    PRINT
        LD      A,'3'
        CALL    PRINT
        LD      A,C             ;X-Koordinate innerhalb einer
        OUT     (XLSB),A        ;Bildschirmhälfte
        LD      E,0FFH          ;Abwärtszähler für Y-Koordinate
B90     LD      A,E             ;nächstes Byte lesen
        OUT     (YLSB),A        ;Y-Wert vorgeben
TEST1   LD      A,0FH           ;Ausgabe anfordern
        OUT     (CMD),A
        CALL    WAIT
        IN      A,(MFREE)       ;HRG-Byte
        LD      D,A             ;zweimal lesen
        LD      A,0FH           ;um eventuelle
        OUT     (CMD),A         ;Lesefehler
        CALL    WAIT            ;zu erkennen
        IN      A,(MFREE)
        CP      D
        JR      NZ,TEST1        ;wenn keine Übereinstimmung
        CPL                     ;Byte invertiert drucken
        CALL    PRINT           ;dreimal drucken, um
        CALL    PRINT           ;Bildschirmformat
        CALL    PRINT           ;ungefähr zu treffen
        DEC     E               ;nächster Y-Wert
        LD      A,0FFH          ;waren das schon alle?
        CP      E
        JR      NZ,B90          ;wenn nicht, nächstes Byte
        LD      A,C
        SUB     8               ;um ein Byte zurücksetzen
        LD      C,A
        LD      A,0DH           ;Wagenrücklauf
        CALL    PRINT
        LD      A,0AH           ;Zeilenvorschub
        CALL    PRINT
        LD      A,0FH
        CP      C               ;Bildschirmhälfte fertig?
        JR      NZ,B40          ;nein: nächste Zeile
        DEC     B               ;sonst nächste Hälfte
        JR      B35
;
PRINT   PUSH    DE
        PUSH    HL
        PUSH    BC
        PUSH    AF
```

```
        LD      E,A             ;und Zeichen
        LD      C,2             ;an Drucker ausgeben
        CALL    0005H
        POP     AF
        POP     BC
        POP     HL
        POP     DE
        RET
;
WAIT    IN      A,(CMD)         ;GDP ready?
        AND     04H
        JR      Z,WAIT
        RET
        END
```





Das 80-Zeichen-Karten-Problem und seine Lösung

Helmut Bernhardt

Vor dem Untergang der offenen und entwicklungsfähigen Computertechnik durch Verdrängung durch den "Industrie-Standard" wurde von einer Aachener Firma für TRS-80 und GENIE eine 80-Zeichen-Karte angeboten. Wenn diese Karte auch in den meisten Fällen nur Probleme bereitete, war sie doch eine Option die durch entsprechende Verbesserung durchaus eine Erweiterung für unsere Computer hätte werden können.

Diese Karte entsprach der TRS-80-Philosophie, daß der-Bildwiederholspeicher in den Adreßraum der CPU eingeb1endet ist und die CPU die Bildschirmverwaltung selbst erledigen muß, Wie das bei unseren Computern für den 64 X 16-Bildschirm ohnehin gemacht wird und Level II ROM und vom DOS auch unterstützt wird. Die 80-Zeichen-Karte wird allerding nicht untertützt. Nur bei neueren TRS-80- und GENIE-Rechnern ist bei der dort schon eingebauten 80-Zeichen-Option ein manchmal funktionierender Treiber im DOS enthalten. Aber all die Programme für den TRS-80, die vom 80 X 24-Bildschirm nichts wissen und die Videoausgabe nicht über das DOS abwickeln, merken nichts davon, so daß bei Umschalten auf 80 Zeichen dann nur Zeichensalat entsteht.

Glücklicherweise gibt es immer Leute, die aus Begeisterung für ein bestimmtes Programm in mühvoller Arbeit dort die Unterstützung solcher Hardware-Erweiterungen einbauen. Dadurch wird ein 80 x 24-Zeichen-Bildchirm aber immer noch nicht TRS-80-kompatibel.

Es sollte noch erwähnt werden, daß im GENIE III und IIIs ausschließlich mit dem CRTC6845 der Bildschirm verwaltet wird. Durch Programmierrung des 6845 kann dieser in 64 x 16- und im 80 x 24-Format arbeiten. Im GIII ist außerdem der Zeichengenerator ein RAM-Speicher in den beliebige Zeichensätze geladen werden können, wobei dann aber immer nur ein Zeichensatz zur Zeit benutzt werden kann. Außerdem verfügt der GIIIs noch über eine flexible HRG, die mit dem Textbildschirm XOR-verknüpft angezeigt werden kann. Trotzdem bleibt es aber Aufgabe der CPU, im Textmodus die Bildschirmadresen zu berechnen und im HRG-Modus beim Zeichnen einer Linie alle Punkte der Linie zu errechnen.

Der eigentliche Sinn eines 80 x 24-Zeichen-Bildschirms ist auch nicht die Anpassung des NEWDOS/80 daran sondern die Nutzung unter CP/M. Für CP/M muß ohnehin das BIOS der Hardware angepaßt werden und da kommt es dann nicht mehr darauf an, ob ein 80 x 24-Bildschirm ähnlich unserem 64 x 16-Bildchirm bedient wird oder ob da völlig andere Verhältnise vorliegen.

Wenn also eine noch so sehr an die TRS-80-Verhältnisse angenäherte 80-Zeichen-Karte dann doch nicht kompatibel ist, dann kann es auch eine bliebig angeschlossene Terminal-Karte sein. Solche Karten bedeuten zudem noch einen gegringeren Programmieraufwand, weil das Anwenderprogramm bei einfacher Zeichenausgabe ohne absolute Cursor-Adressierung sich keine Gedanken um Bildchirmposition, Zeilen- und Seitenende und Scrolling machen muß. Diese Leistungen erbringt se1bst das dümmste Terminal selbst.

Grundsätzlich ist ein Terminal ein Interface zwischen Computer und Anwender und umfaßt daher zur Ausgabe an den Anwender die Bildschirmausgabe (und manchmal zusätzlich auch die Druckerausgabe) und die Tastatureingabe. Die Eingabe erfolgt hauptsächlich über ASCII-Tastaturen, die mit einem Strobe-Signal dem Terminal ein ASCII-Zeichen übergeben. Das wird zum Teil parallel oder auch seriell gelöst.

Das Terminal gibt das Zeichen unbesehen an den Computer weiter und dieser echot es nach Übernahme an das Terminal zurück. Erst dann wird das Zeichen vom Terminal auf den Bildschirm ausgegeben.

In einigen Fällen werden dabei Ausnahmen gemacht. So verstehen Terminals meisten ein bestimmtes Steuerzeichen als Befehl 'Local' und ein anderes (oder das gleiche ein zweites mal) als Befehl 'On line'. Damit kann man umschalten zwischen Ausgabe nur der Zeichen vom Computer und sofortige Ausgabe der Zeichen von der Tastatur. ohne diese an den Computer weiterzureichen.

Manche Terminals können auch mit einem Steuerzeichen in einen Setup-Modus versetzt werden, von wo aus gewisse Parameter des Terminals über die Tastatur geändert werden können. Dazu kann gehören; die Umschaltung des Bildschirmformates die Wahl nationaler Zeichensätze, die Änderung der Baudrate für die Kmmunikation mit dem Computer u.s.w.

Die Verbindung mit dem Computer kann verschiedene Formen annehmen. Die verbreitetste Variante ist ein über eine serielle Schnittstelle angeschlossenes Terminal. Damit ist es möglich, daß das Terminal mit Bildschirm und Tastatur ein eigenständiges Gerät ist und der Computer weiter entfernt stehen kann. Da bei den heutigen Tischgeräten dieser Gesichtspunkt nicht mehr ausschlaggebend ist, werden auch schon viele 'Video-Karten' für diverse Bussysteme als Steckkarten im Computer über den Systytembus angeschlossen. Einige Varianten vereinigen beide Gesichtspunkte und verfügen über z.B. einen ECB-Bus-Stecker, der nur der Zuführung der Versorgungsspannungen dient, und eine RS232C-Verbindung mit dem Computer. Prinipiell ist der Anschluß über den Systembus wegen des parallelen Datentransfers schneller. Bei Baudraten von 9600 oder 19200 ist aber auch der serille Anschluß schnell genug. Auf alle Fälle kann er mit der Ausgabe eines memory mapped Bildchirms, der unter CP/M für die Ausgabe jedes Zeichen ständig ein- und ausgeblendet werden muß, mithalten.

Das hat alles Vor- und Nachteile: Ein Terminal verlangt eine sehr viel leitungsfähigere Tastatur, die schon selbst die Umwandlung von ausgelesenen gedrückten Tasten in ASCII-Codes leisten muß. Bei unseren Rechnern muß das DOS bzw. der Z80 diese Umwandlung erledigen, was entsprechend umfangreiche Eingaberoutinen und auch CPU-Rechenzeit kostet. Da ein Programm aber bei Eingaben ohnehin nur auf die Tastatur wartet, ist das ohne Belang. Anderseits ist es bei einer ASCII-Tastatur auch nicht möglich, Mehrtasten-Befehle einzubauen (123, JKL DFG). Ähnliches kann aber mit programmierbaren Funktiontasten oder einer Interrupt-getriebenen Terminaleingabe auch realisiert werden. Das wird aber in den meisten Fällen aus Angst vor

Interrupt-Programmierung unterlassen.

Programmierbare Funktionstasten von ASCII-Tastaturen werden meistens durch Ausgabe von Codes > 7FH beim Drücken einer 'F'-Taste realisiert. Die Eingaberoutine des Betriebssystems lügt sich dann bei einem solchen eingegebenen Zeichen einen diesem Zeichen zugeordneten String als eingegeben vor.

Als reines Textausgabegerät kann also eine Terminal-Karte in TRS-80-kompatiblen Computern insbesondere unter CP/M als leistungsfähiger Ersatz für eine von der CPU selbst verwaltete, memory mapped 80-Zeichen-Karte gute Dienste leisten. Der Anschluß einer ASCII-Tastatur ist möglich aber nicht zwingend. Gegenüber der minimal-Tastatur des TRS-80 und des GENIE ist eine ASCII-Tastatur aber ein zusätzlicher Komfort, der die übliche Fingerakrobatik bei der Eingabe von Umlauten endlich durch dafür vorgesehene Tasten an der richtigen Stelle ersetzt.

Bei Terminal-Karten existiert ein eigener Prozessor, der den Bildwiederholspeicher verwaltet und die von der CPU übergebnen Zeichen entgegennimmt. Außer den darstellbaren Zeichen (20H-7FH) versteht er die meisten Steuerzeichen (00h-1FH) und setzt diese in entsprechendes Wohlverhalten um. Meistens reichen diese Befehle aber nicht für alle Wünsche der Bildchirmverwaltung aus, so daß außerdem noch ESC-Sequenzen weitere Steuerfunktionen bereitstellen. In der Festlegung solcher ESC-Sequenzen und in deren Menge unterscheiden sich die verchiedenen Terminals voneinander. Die Bedienung eines Terminal ähnelt also mehr der eines Druckers als der des herkömmlichen TRS-80-Bi1dschirms. Und genauso wie bei den Druckern gibt es mehr oder weniger leistungsfähige Terminals.

So gibt es analog den Typenrad-Druckern auch reine Text-Terminals, die nur ASCII-Zeichen (natürlich auch deutsche Umlaute) darstellen können, und andere die wie ein Matrixdrucker auch graphikfähig sind. Eine solche graphikfähige Terminalkarte, die GRIP, hat uns Manfred Held im letzten Info schon vorgestellt und auch gezeigt, daß sowas bei uns durchaus gute Dienste leisten kann. Die neueste Version der GRIP (Vers.5.3) ist in der Lage, HRG mit 768x567 Punkten darzustellen und Text in bis zu 96x35 Zeichen auszugeben. Durch den HD6345-CTRC, der aufwärtskompatibel zum MC6845 ist, wurde dort auch ein augenschonendes Smooth-Scrolling (Hochschieben des Bildschirminhaltes nicht textzeilenweise sondern dotzei1enweise) erreicht. Gegenüber der etwas lahmen Urform GRIP1 ist die GR3P5.53 eine der schnellsten Terminal-Karten für CP/M-Computer überhaupt.

Eine ähnlich leistungsfähige Karte ist die Video 1 von Janich & Claas, die ebenfalls HRG- und Textdarstellung beherrscht.

Weniger empfehlenswert ist die TERM 1 des mc-CP/M-Computers, die zwar wegen des EF9366 Video-Controllers in der Graphik einige interessante Aspekte bietet, die aber wegen der äuerst langsamen Textdarstellung überhaupt nicht zu gebrauchen ist.

Die GDP des ND-Klein-Computers ist eine stark abgemagerte TERM 1 ohne eigenen Terminal-Prozessor. Hier muß die CPU selbst den EF9366 programmieren. Das bringt zwar Geschwindigkeitsvorteile, macht die normale Textausgabe aber recht umtändlich. Diese Karte ist auch im eigentlichen Sinn kein Terminal, ihre Bedeutung ist auch einzig in der schnellen Vetorgraphik des Ef9366 zu sehen.

Beiden Karten mit dem EF9366 ist gemeinsam, daß sie die vereinhahmten HRG-Inhalte nicht wieder hergeben (es sei denn, man rüstet die Eigenschaft in Hardware nach).

In den meisten Fä11en genügt es aber, wenn man für die Implementation von CP/M eine reine 80x24-Text-Terminal-Karte benutzt. CP/M fordert zwar keine 80x24- Zeichen-Darstellung und flexiblere Programme lassen sich auch für 64x16-Bildschirme installieren; die meisten unter CP/M laufendn Programme gehen aber von dieser Bildschirmgröße aus, so daß man beim 64x16-Zeichen-Format einen großen Teil der Informationen nur vorbeihuschen siet und nicht lesen kann. Besonders ärgerlich ist dies bei den in vielen Utilities aus dem Programm heraus aurufbaren HELP-Texten.

Eine sehr elegante Lösung besteht in der Verwendung des c't-Text-Terminals. Neben einer mit maximal 19200 Baud Übertragungsrate sehr schnellen Textausgabe überzeugt diese Karte besonders durch ihr günstiges Preis/Leistungs-Verhältnis. Die Karte läßt sich für ca. 300 DM aufbauen und bietet dabei eine Fülle von Optionen. So lassen sich Zeichen mit bis zu 8 gleichzeitig wirksamen Attributen darstellen und es können gleichzeitig verschiedene Zeichen oder Zeichengruppen auf dem Bildschirm mit verschiedenen Attributen dargestellt werden. Die Attribute umfassen; Normaldarstellung, doppelt breit, doppelt hoch, halb hell, invers, blinkend und unsichtbar. Die Attribute werden durch entsprechende ESC-Sequenen gesetzt und gelten dann für alle folgenden ausgegebenen Zeichen, bis andere Attribut-Befehle augegeben werden.

Literatur

Low-Cost: c't Text-Terminal; Tilman Reh; c't 9/86, 65
Das mc-Terminal; R. Jäger, H.J. Regge, G.Tobergte; mc 9/83, 70
Das mc-Graphik-Terminal: R.-D. Klein; mc 8/83, 68
Graphik-Interface-Prozessor GRIP 1; J.C.Lotter, c't 6/84, 52; c't 7/84, 92

Anmerkungen zum Club-80-ECB-Bus-Projekt

Liebe Club-Freunde, die Ihr den Mut gehabt habt, Euch an die Löterei des ECB- Bus-Projektes heranzuwagen. Euch speziell gilt der folgende Artikel zu dem Thema. Vorher möchte ich aber noch einige Bemerkungen loswerden, die verdeutlichen sollen, warum ich meine etwas abweichenden Vorstellungen eines ECB-Bus-Interfaces denen Eures Projektes entgegenstellen muß.

Die Grundvorstellung des ECB-Bus ist nicht nur das Umsortieren einer wilden Pinanordnung auf dem Systembus-Steckern unserer Computer auf die noch unübersichtlichere des ECB-Bus sondern noch eine Reihe weiterer Strukturen, die speziell auf die besonderen Fähigkeiten der Z80-CPU abgestimmt sind und auch von einem vollständigen ECB-Bus unterstützt werden.

Dazu gehört zunächst der Interrupt Mode 2 (IM2), der voraussetzt, daß die Interrupt-auslösende Karte der CPU beim Interrupt-Acknowledge einen 8Bit-Vektor über den Datenbus mitteilt. Dieser Vektor muß vom ECB-Bus über den Datenbuffer auch zur CPU gelangen können, wenn IORQ und M1 gleichzeitig low sind.

Außerdem müssen die Buffer aller Bussignale bidirektional sein, weil bei DMA- Zugriff auf Komponenten des CPU-Boards der DMA-Controller auf dem ECB-Bus auch die entsprechenden Baugruppen des CPU-Boards adressieren muß. Das bedeutet einen sehr viel größeren Aufwand bei der Steuerung der Treiberrichtung dieser Buffer.

Und schließlich soll der ECB-Bus nicht nur selbstgestrickte Karten aufnehmen sondern hauptsächlich den Einsatz der breiten Palette käulicher Karten für ECB-Systeme ermöglichen. Daher verbietet sich auch die prinzipiell geniale Lösung der Richtungssteuerung des Datenbuffers über Freigabesignale, die die ECB-Bus-Karten liefern. Kommerzielle Karten wissen nicht, daß Euer Bus solche Signale voraussetzt.

Das Konzept Eures ECB-Bus baut auf den Möglicheiten des TRS-80-Systembus auf, die leider wirklich recht bescheiden sind, IM2- und DMA-Fähigkeit sind dort nicht vorhanden, beim GENIE und beim Komtek 1 lassen sich aber alle Eigenschaften des ECB-Bus verwirklichen, ohne intern große Eingriffe machen zu müssen.

Um aber auch beim TRS-80 zumindest 90% der kommerziellen Karten die ohne IM2 und DMA arbeiten, nutzen zu können, ist es unumgänglich von der Richtungssteuerung des Datenbuffers durch die Karten selbst abzukommen. Da ist dann auch die bei mir eingeschlagene Ochsentour nötig die Richtungssteuerung aus den Freigebesignalen der Baugruppen innerhalb des Computers herzuleiten.

Die im folgenden Artikel beschriebene ECB-Bus-Eingangskarte berücksichtigt nicht nur die o.a. Gesichtspunkte sondern bringt noch ein paar zusätzliche Leistungen, die nicht unbedingt an den ECB-Bus gebunden sind, sondern einige andere Hardware-Erweiterungen unterstützen. Diese Features sind aber nicht unbedingt wichtig und können auch ausgespart werden.

Es ist beim GENIE und Komtek ohne weiteres möglich, den ECB-Bus mit dieser Eingangskarte anstelle der alten zu betreiben. Beim TRS-80 sind noch zusätzliche Maßnahmen nötig, die einen voll funktionfähigen ECB-Bus ermöglichen.

Helmut Bernhardt

1988-Club80-02 Nr.23 S.74-84

Gepufferter DMA- und IM2-fähiger ECB-Bus für GENIE I und II

Helmut Bernhardt

Während bei Verzicht auf DMA- und IM2-Fähigkeit die Steuerung des Datenbuffers zwischen Computer und externem ECB-Bus relativ einfach zu realisieren ist (der Buffer darf nur in Richtung CPU treiben, wenn Daten von Baugruppen gelesen werden, die nicht auf der CPU-Seite des Buffers - also auf dem ECB-Bus - sitzen), sind bei der Einbindung dieser Features einige zusätzliche Aspekte zu berücksichtigen.

Der Interrupt-Mode2 der Z80-CPU setzt voraus, daß der Interrupt-Lieferant beim Interrupt-Acknowledge (M1* und IORQ* beide low) ein 8Bit-Wort auf den Datenbus legt, das die CPU als LSB einer Adresse in der Interrupt-Vektor- Tabelle interpretiert. Dieses Byte muß von einer Interrupt-Quelle vom ECB-Bus durch den Datenbuffer auch zur CPU gelangen. Der Buffer muß beim Interrupt-Acknowledge also auch in Richtung CPU treiben.

Bei Übernahme der Daten-, Adreß- und Steuerbusse durch einen DMA-Controller (DMAC) auf dem ECB-Bus muß dieser auch Zugriff auf RAM, EPROM und alle CPU-seitig des Buffers liegenden Baugruppen haben, für Adreß- und Steuerleitungen gestaltet sich diese Richtungssteuerung recht einfach. Hier genügt es, wenn durch das low aktive CPU-Signal BUSAK* (nur dann darf der DMAC arbeiten) die Buffer dieser Signale in Richtung zur CPU treiben, so daß der DMAC dort entsprechend adressieren kann. Für die Steuerung des Datenbuffers sind noch weitere Umstände zu berücksichtigen.

Wenn die CPU den Bus kontrolliert, kann beim Schreiben der Buffer immer in Richtung ECB-Bus treiben. Beim Lesen darf er nur dann nicht in Richtung CPU treiben, wenn interne Baugruppen (CPU-seitig des Buffers) adressiert werden, sonst würde ein von außen gelesenes FFH gegen ein von der intern adressierten Baugruppe ausgegebenes Datenwort arbeiten. Abgesehen davon, daß die gegeneinander arbeitenden Komponenten sich irgendwann gegenseitig zerstören, würde die CPU Fehlinformationen lesen.

Wenn der DMAC <4,5> den Bus steuert, ist Lesen und Schreiben für die Treiberrichtung des Buffers das ganze Gegenteil. Wenn der DMAC Daten einer CPU-seitigen Systemkomponente liest, wobei er dann auch die Leitung RD* low zieht, muß der Buffer in Richtung ECB-Bus treiben. Das darf er aber nur dann, wenn er auf CPU-seitige Komponenten zugreift. Wenn er von einer Baugruppe auf dem ECB-Bus liest, muß der Buffer in Richtung CPU treiben, um dann nicht auf dem ECB-Bus zwei Datenwörter gegeneinander arbeiten zu lassen.

Für die Treiberrichtung des Buffers muß also berücksichtigt werden, ob der DMAC oder die CPU den Bus kontrolliert, ob die adressierte Baugruppe auf dem ECB-Bus oder CPU-seitig des Buffers liegt und ob gelesen oder geschrieben wird.

Abb.1) Steuerung des Databuffers, Portdecodierung, programmierbares Invertieren von A15 sowie A14 UND, A15 Umschalten der Interrupts

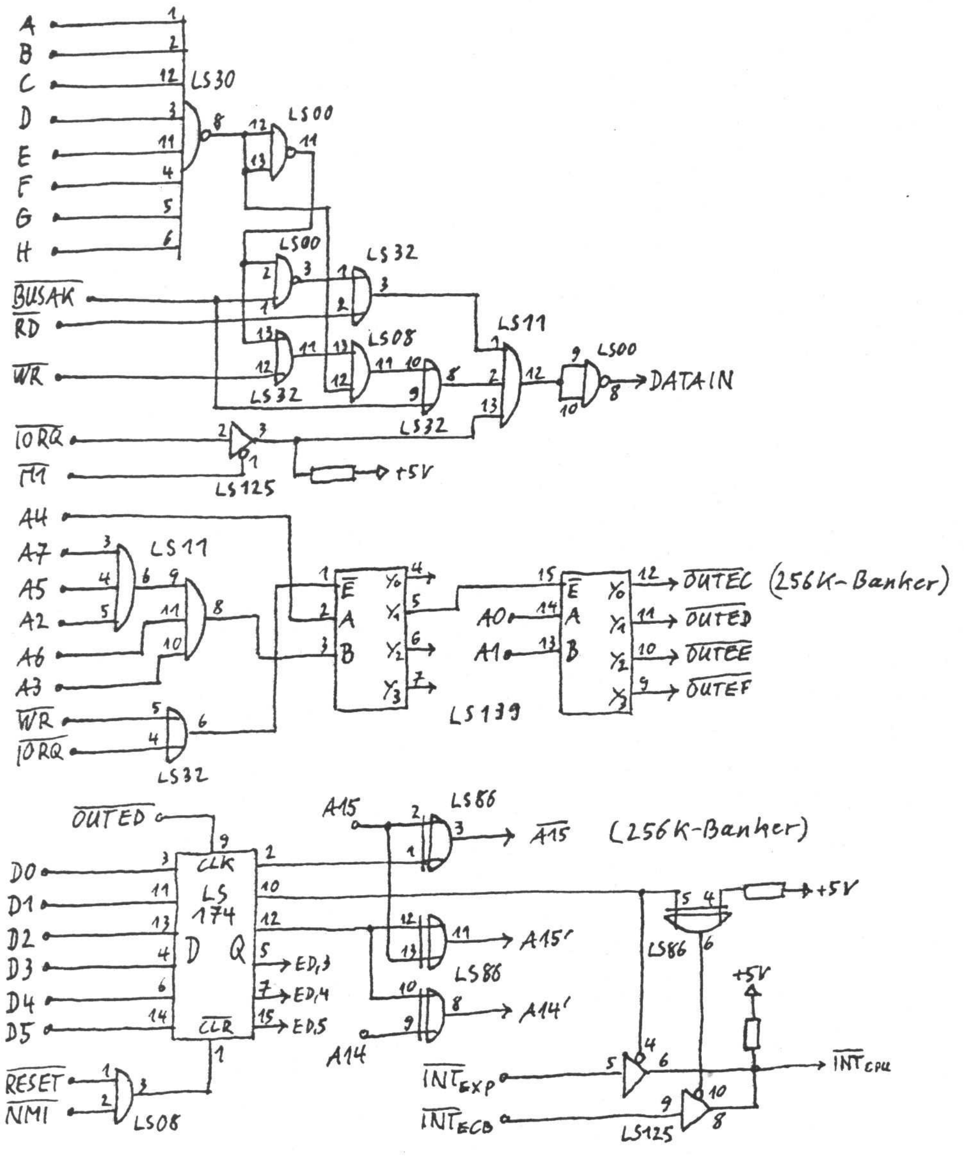
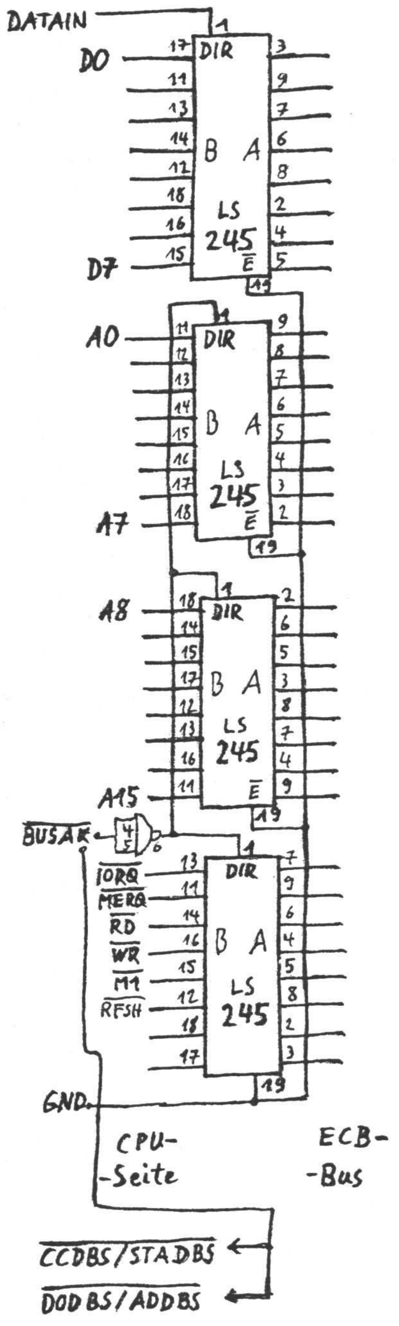


Abb.2) Pufferung des ECB-Bus



In Tabelle 1 sind die für verschiedene Betriebszustände aus obigen Überlegungen abgeleiteten Treiberrichungen (von der CPU aus gesehen) zusammengestellt. Dabei ist zu berücksichtigen, daß zusätzlich auch noch beim Interrupt-Acknowledge die Richtung "rein" sein muß, weil der GENIE keine eigenen vektorisierten INT-Quellen hat. Diese Tabelle ist Grundlage der Schaltung in Abb.1. Die Richtungssteuerung soll ausschlielich aus Signalen, die von CPU-seitigen Baugruppen zu beziehen sind, erfolgen. Nur dann ist auch gewährleistet, daß auf dem ECB-Bus jede handelsübliche Karte funktioniert. Kommerzielle ECB-Bus-Karten legen ein aktives Signal auf den Bus, das in die Buffersteuerung einbezogen werden kann.

Tabelle 1: Treiberrichtung des ECB-Bus Buffers

Bus-Steuerung durch	interne Baugruppe adressiert Read	interne Baugruppe adressiert Write	externe Baugruppe adressiert Read	externe Baugruppe adressiert Write
CPU	raus	raus	rein	raus
DMAC	raus	rein	rein	rein

Der Auwand dafür ist aber nicht unerheblich und der Anwender dieser Bussteuerung muß sein System genau kennen, um die nötigen Steuersignale an die Schaltung legen zu können. Einen Anhaltspunkt dafür, welche Freigabesignale vorkommen können, mag die Tabelle 2 geben. Darin sind die prinzipiell vorhandenen Baugruppen und einige häufig benutzte Hardware-Erweiterungen und die Punkte, wo deren Freigabesignale abzugreifen sind, zusammengestellt. Diese Punkte sind mit jeweils einem der Punkte A bis H auf der Eingangskarte zu verbinden.

Tabelle 2: Interne Feigabesignale

CPU-Board	Adresse	IC	Typ	Pin
RAM	4000-FFFF	37	37	15
	(3900-3BFF) <7>			
ROM1	0000-0FFF	10	3001	20
ROM2	1000-1FFF	11	3002	20
ROM3	2000-2FFF	12	3003	20
ROM4	3000-36FF(37FF) <7>	13	2716	20
Tastatur	3800-3BFF(38FF) <7>	7	368	1
Video-RAM	3C00-3FFF	35	32	3
Video-Board				
Cassette 1	Port FF	20	32	11
Cassette 2	Port FE	bleibt unberücksichtigt #1		
Expansion-Interace	EG3014			
RAM1	8000-BFFF	37	20	10
RAM2	C000-FFFF	37	20	9
Floppy	37E0-37FF	29	139	12
Drucker	37E8 u.Port FD	31	156	9
RS 232 EG3020	Ports F8-F9	37	20	10
EXP1 von RB	37E0-37FF	18	155	2,14
	Port FD	14	32	3
RB V24	Ports 80-88	Z80-CTC		16
		Z80-DART		35
HRG1B von RB	Ports 00-7F	?	155	2,14
Schmidte 80Z.Karte	B000-BFFF	?	245	1
	Ports D0-D1	?	138	15
EG64MBA	Port DF	bleibt unberücksichtigt #1		
EG64MBA+ <6>	Ports DE-DF	"	"	
256K-Banker <10>	Port EC	"	"	
	(u. ED mit dieser Karte)	"	"	
CP/M-Banker <6>	Ports 50-5F	"	"	

#1 diese Baugruppen werden nicht gelesen und brauchen deshalb nicht in die Buffersteuerung mit einbezogen zu werden, das **IN A(0DFH)** des EG64MBA dient nicht der Datenaquisition, es setzt nur das 79LS259-Latch zurück, ein Datenwort wird nicht ausgegeben.

Im Normalfall liegen alle memory-mapped (mm) Baugruppen CPU-seitig des Treibers, so daß dann einzig MERQ* (IC3, 74LS367, Pin5) an die Schaltung geführt werden muß. Wenn aber auch nur eine mm Baugruppe auf dem ECB-Bus steckt, müssen alle anderen mm Freigabesignale einzeln an die Schaltung geführt werden.

Durch Vorschalten weiterer AND-Gatter vor die Eingänge des 74LS30-ICs (z.B. 74LS08, 74LS11, 74LS21) kann die Anzahl der verwendbaren Eingänge für Freigabesignale beliebig erhöht werden (Abb.1).

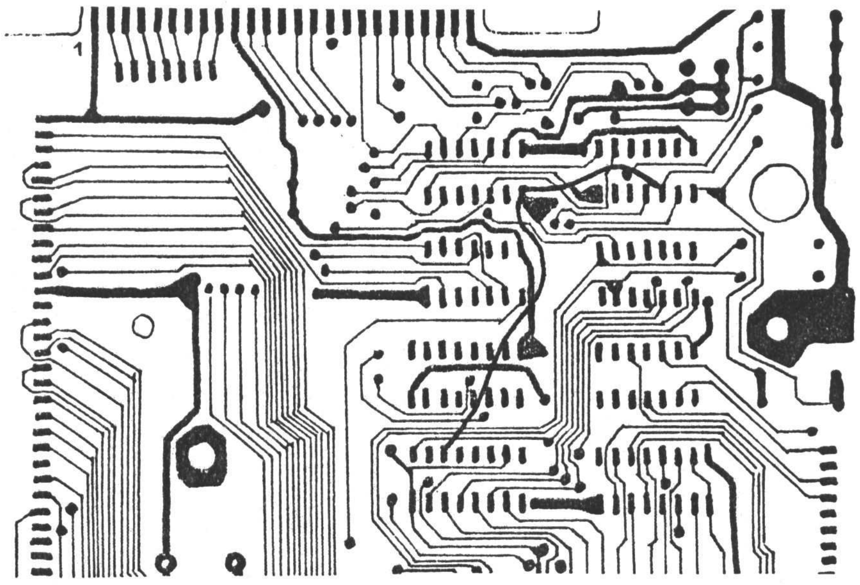


Abb.8) Verbinden von BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS*

Die DMA-Fähigkeit der ECB-Busbuffer alleine reicht nicht aus. Es muß durch Verbinden der Systembus-Leitung BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS* auch dafür gesorgt werden, daß beim DMA-Zugriff die Buffer der CPU-Signale in den hochohmigen Zustand versetzt werden. Das kann entweder auf der Eingangskarte durch Querverdrahtung der entsprechenden Pins von CN2 bzw, CN3 oder durch feste Verdrahtung auf dem CPU-Board gemäß Abb.8 erreicht werden.

Da für die Richtungssteuerung des Datenbuffers ohnehin einige ICs nötig waren, und der ECB-Bus auch IM2-fähig ist, ließ sich mit etwas Mehraufwand auch gleich eine Vektorisierung der Interrupts des 25ms-Timers und des Floppy-Controllers realisieren. Dafür ist Voraussetzung, daß auf dem ECB-Bus auf einer Karte ein freier Z80-CTC-Kanal (2) vorhanden ist, über den durch Software das gemeinsame INT*-Signal des Systems umgeleitet werden kann. Der CTC-Kanal ist dann im Zähler-Modus mit der Zählkonstanten 1 und Triggern durch eine negative Flanke zu programmieren.

Die direkte Verbindung des INT*-Signals (im Schaltplan INTEXP*) von der Floppy-Baugruppe zur CPU ist dafür aufzutrennen. Das Signal INTEXP* ist stattdessen an den entsprechend bezeichneten Punkt der Schaltung und an den freien Triggereingang des CTC-Kanals zu legen. Die INT*-Leitung des ECB-Bus ist an den Punkt INTECB* zu legen und der Ausgang INTCPU* ist mit dem INT- Einang der CPU (Z80, Pin3) zu verbinden.

Damit ist gewährleistet, daß nach dem Einschalten oder Drücken des RESET- Knopfes immer der Interrupt vom FDC-Board direkt zur CPU durchgeschaltet wird und das Betriebssstem im IM1 arbeiten kann. Wenn allerdings andere Baugruppen auf dem ECB-Bus in Echtzeit im IM2 betrieben werden sollen, müssen auch die systemeigenen Interrupts dieses Spiel mitmachen können und das geht nur dann, wenn ihnen der CTC-Kanal beim INT-Acknowledge einen Vektor zur Verfügung stellt.

Die Schaltung leistet es, daß ein **LD A, xxxxxx1xB; OUT (0EDH),A** das INT* Signal des FDC-Boards über den CTC umleitet.

Die Software muß dann aber Rücksicht auf diesen Umstand nehmen. In der INT- Vektor-Tabelle darf nicht einfach 40F2H stehen (INT-Vektor des DOS), da die Serviceroutine des DOS nicht mit RETI sondern mit RET endet und der Z80-CTC dann nicht weiß, wann er seinen IEO-Ausgang wieder auf High zurücknehmen soll. Es muß folgender Umweg eingeschlagen werden.

```
INTAB   DEFW    SERV1   ;Anfang INT-Tabelle
        DEFW     SERV2  ;nächster Vektor
        DEFW     ....   ;u.s.w.
        ....    ....
        DEFW     CTCN   ;Vektor für Serviceroutine des INT vom CTC-Kanal
        ....            ;N, über den die System-Interrupts laufen
        DEFW    LETZT   ;Ende der Vetortabelle

CTCN    CALL    4032H   ;Serviceroutine des D0S
        EI
        RETI
```

Außerdem bleibt es dem Anwender offen, während des IM2-Betriebes anderer Baugruppen den 25ms INT verhungern zu lassen. Wenn beim Lable CTCN einfach ein EI, RETI steht, wird 37E0H nicht mehr gelesen und darauhin erfolgt kein weiterer Interrupt des Timers mehr. Eine Reinitialisierung des Timer-INTs geschieht durch Umprogrammieren des Vektors auf ein anderes Lable, das wieder den CALL 4012H enthält, und einmaliges Lesen von 37E0H.

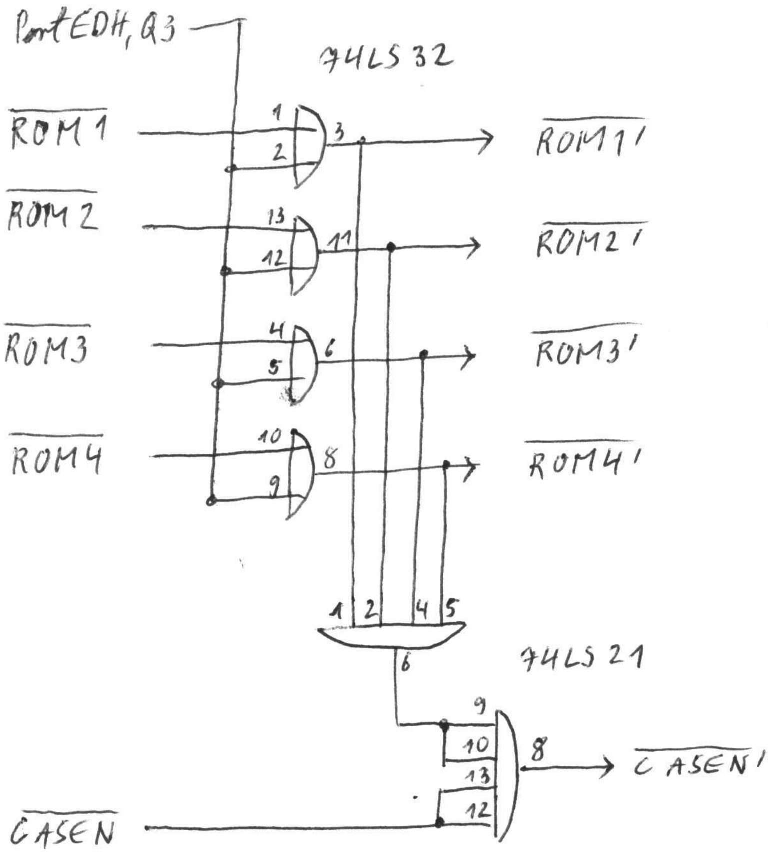
Durch Zurücksetzen von D1 in Port E0H wird wieder der INT des FDC-Boards direkt an die CPU durchgeschaltet und alle Interrupts des ECB-Bus werden unterbunden.

Mit der für die Interrupt-Steuerung benötigten Portdecodierung und mit dem Latch wurden noch einige zusätzliche Funktionen realisiert. Zunächst wurden auch die Portfreigabesignale OUTECH*, OUTEEH* und OUTEFH* hergeleitet. Davon kann OUTECH* zur Freigabe des 74LS273 Latch auf dem 256K-Banker <10> be- nutzt werden. Dieses Signal wird direkt an den Pin11 des 74LS273 geführt, wobei dann aber die bisher dahin führende Leitung zu durchtrennen ist. Damit ist dann das 74LS30 IC auf dem Banker überflüssig und auch die Leitungen A0 bis A5 sowie IORQ* und WR* brauchen nicht mehr dorthin geführt zu werden (A6 und A7 werden dort nicht nur zur Portdecodierung sondern auch noch zur Erzeugung der 8Bit-Refreshadresse benötigt und müssen deshalb weiterhin angeschlossen bleiben).

Mit Bit 0 in Port EDH wurde ein softgesteuertes Invertieren des zum 256K-Banker führenden Signals A15 realisiert. Damit läßt sich mit D0=0 an Port EDH (Einschaltzustand) vorgeben, daß die unteren 32K Adreßbereich des Z80 gebankt werden können, während D0=1 an Port EDH das Banking der oberen 32K festlegt.

Und schließlich kann mit D2=1 an Port EDH ein gemeinsames Invertieren der Adressen A14 und A15 erreicht werden. Die dabei gewonnenen Signale A14' und A15' können anstelle der Signale A14 und A15 an die Decoderschaltung für die 16K (ROMs und mm I/O), IC25, 74LS139 Pins 14 und 13 geführt werden und ge- statten dadurch ein Verlegen dieser Komponenten an das obere Speicherende 48-64K. In diesem Bereich funktionieren dann zwar die ROMs nicht mehr, wenn man aber dann noch ein 74LS32 IC spendiert und mit dem Signal ED,3 (D3 von Port EDH) ein programmierbares Abschalten der ROMs einbaut (Abb.10), hat man auch noch die Funktionen des Omnicron Mappers (compatibler Selbstbau in <9>) für den Betrieb von CP/M realisiert. Dieser leistet (wenn auch mit einem anderen OUT-Befehl) ebenfalls das Einstellen der Systemkonfiguration.

Abb.10) programmierbares Abschalten der ROMs

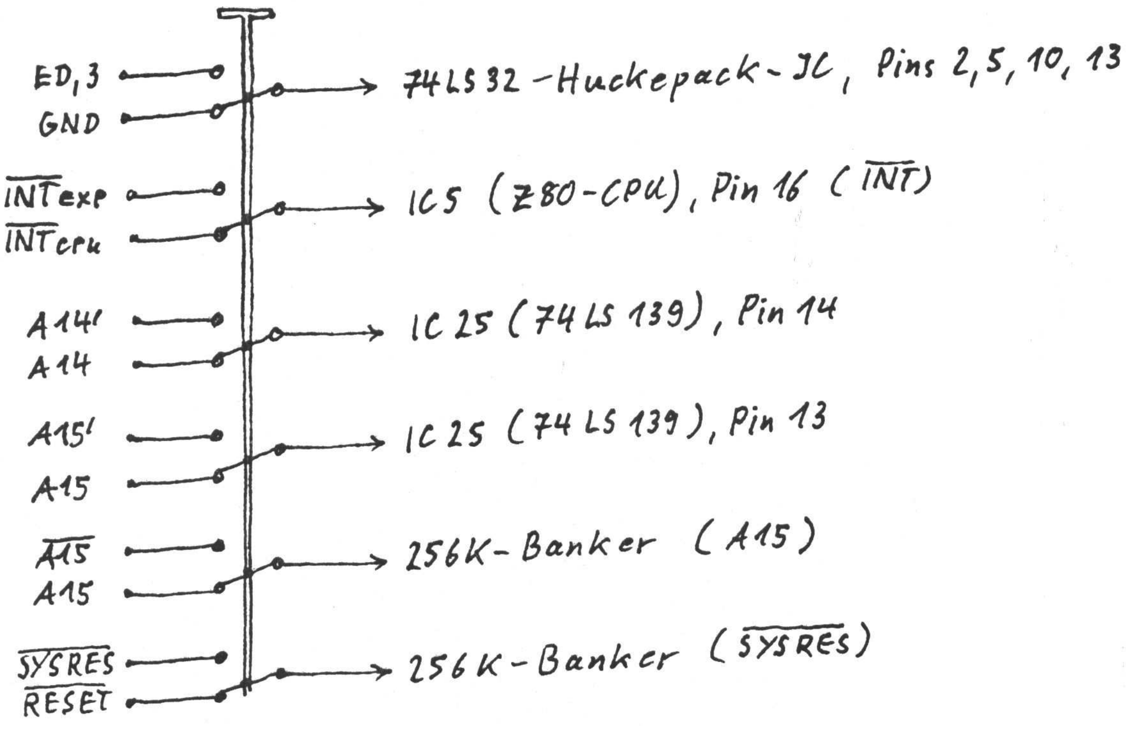



```
0000H-F7DFH   RAM  <7<>   #1
F7E0H-F7FFH   Floppy, Drucker
F800H-FBFFH   Tastatur  (7)
FC00H-FFFFH   Video-RAM
#1 der Omnicron Mapper für den TRS 80
   leistet selbst eine vollständige
   Decodierung des Bereichs des
   Sonder-ROMs des GENIE
```

Ein Patchen der CP/M Systemdisette ist sehr einfach. Auf den Systemspuren läßt sich relativ einfach die Sprungleiste des Loader BIOS finden, die aus einer Reihe von Jumps besteht (C9 nn nn C9 nn....). Daran schließen sich die Disk Parameter Tabellen mit sehr vielen 00H an und unmittelbar danach stehen die Befehle **LD A,40H; OUT (50H),A**. Diese beiden Befehle sind durch die Befehle **LD A,xxxx 11xxB; OUT (0EDH),A** zu ersetzen. Damit läuft das CP/M des Omnicron Mappers dann auch ohne denselben.

Wenn diese Eingriffe auf dem CPU-Board gemacht werden, ist der Computer aber von der ECB-Bus-Eingangskarte abhängig, weil von dieser dann lebenswichtige Signale geliefert werden. Der ECB-Bus braucht selbst nicht angeschlossen zu sein.

Abb.9) Umschalter zwischen ursprünglichen Signalen des CPU-Boards und oder Steuersignale der ECB Eingangskarte"



Von dieser Abhängigkeit kann man sich durch einen Mehrfach- (oder mehrere Wenigfach-) Umschalter lösen. Abb.9 zeigt, wie man damit entweder die ursprünglichen Signale des CPU-Boards oder die steuerbaren Signale dieser Schaltung an die entsprechenden Stellen des CPU-Boards führt, und damit das Gerät entweder wie bislang ohne ECB-Eingangskarte oder mit derselben und allen ihren Mehrleistungen laufen lassen kann.

Um den Umfang der (zusätzlich zu denen des Systembus) an die Eingangskarte geführten Leitungen etwas zu reduzieren, bietet es sich an, IC7, 74LS30 auf der Eingangsarte nicht zu bestücken und dieses IC im Computer an geeigneter Stelle huckepack mit den Pins 7 und 14 auf ein anderes Dil14 IC zu löten und die internen Freigabesignale fest dorthin zu verlegen (Pins 1-6, 11-12). Dann braucht nur noch das Signal INTERN von Pin8 des 74LS30 an die Eingangskarte (CN2 Pin25c) geführt zu werden.

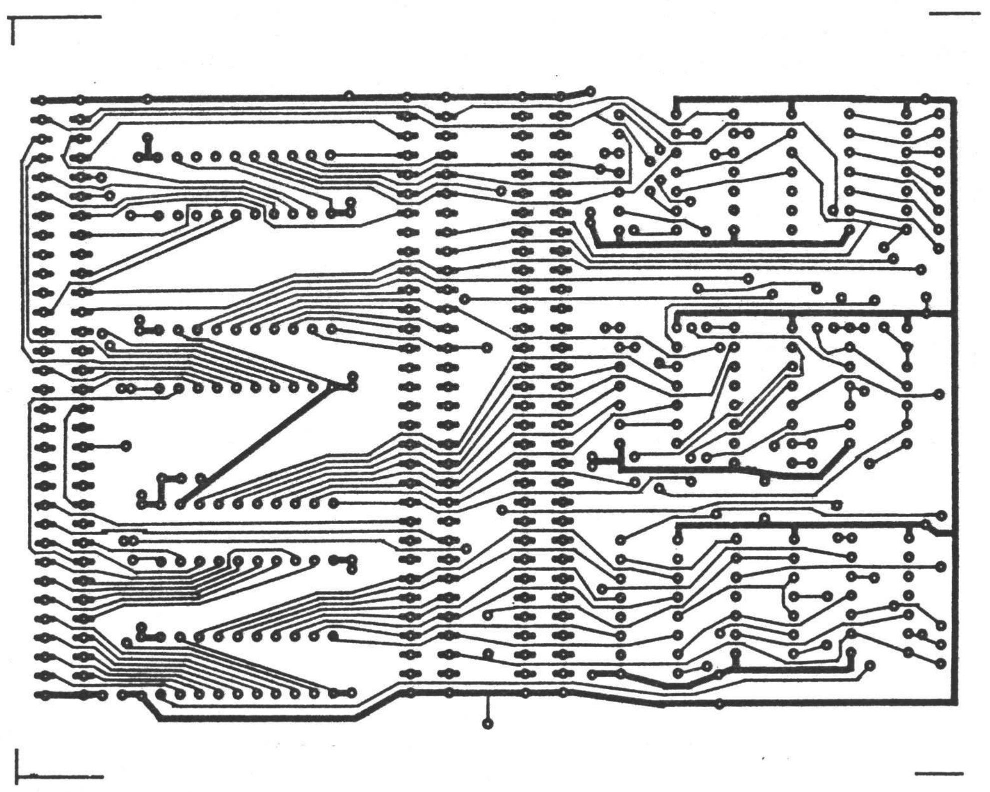


Abb.3) Layout der Eingangskarte

Wenn die zusätzlichen Features der Eingangskarte voll genutzt werden sollen, können alle benötigten Signale (SYSRES* A15* A14' A15' ED,3 OUTECH*) über freie Verdrahtung an unbelegte Pins von CN2 gelötet werden. Im Layout sind nur die Sinale INTCPU* INTEXP* INTECB* und INTERN über CN2 geführt, Dadurch können alle Verbindungen mit dem Computer (nicht nur der Systembus) über CN2 laufen.

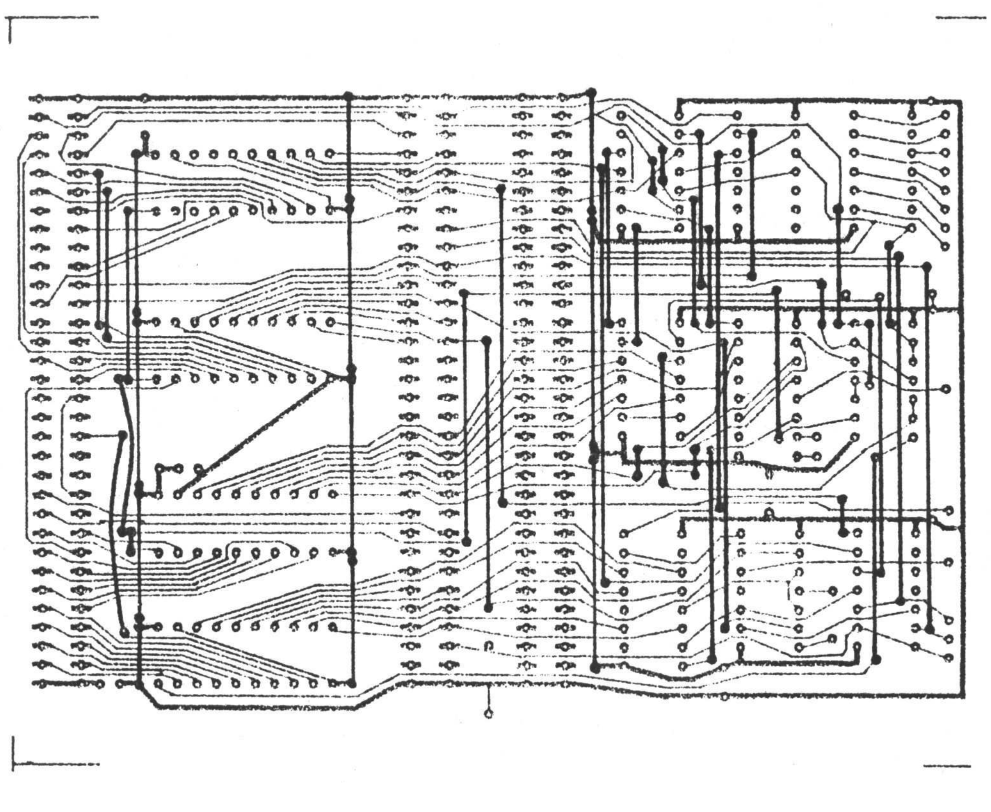


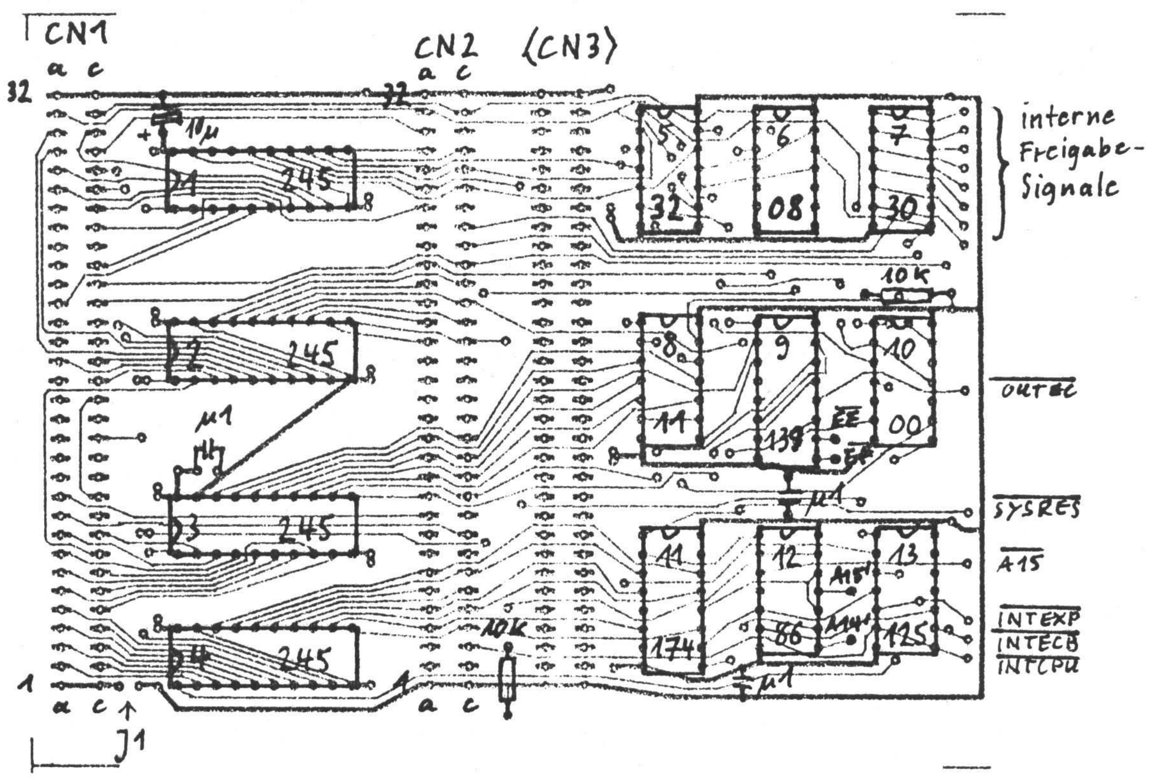
Abb.4) Drahtbrückenplan Bestückungsseite

Wer diese Variante des ECB-Bus bauen möchte, kann bei mir für 10 DM (incl. Rückporto) eine geätzte aber ungebohrte Karte im Europaformat bekommen. Sollte irgendjemand die Karte am Komtek 1 betreiben wollen, kann ich entsprechende Informationen zur Verfügung stellen, die eine Nutzung auch dort ermöglichen. Der Anschluß an den TRS 80 ist nicht ohne Sondermaßnahmen möglich, weil die internen Datentreiber IM2-Betrieb verhindern. Ein Hard- ware-Patch der dagegen hilft, ist in <8> beschrieben. Eine Herleitung der Signale IORQ* MERQ* RD* und WR* aus den daraus gewonnenen Signalen IN* OUT* MWR* und MRD* des TRS 80 Systembus mit Gattern erzeugt Signale mit nicht ganz richtigem Timing, die eventuell Schwiedigkeiten bereiten können; hier ist das Abgreifen der originalen Signale eine bessere Lösung.

Aufbau und Anschließen der Karte

Entgegen der üblichen Technik, solche Karten doppelseitig geätzt zu entwerfen, wurde hier ein einseitiges Layout mit einer unverschämten Menge Drahtbrücken bevorzugt, was das Durchkontaktieren an IC-Sockel-Pins vermeidet. Damit wird beim Aufbau aber viel Fummelei und beim Testen viel Ärger erspart.

Abb.5) Bestückungsplan, J1 ist nur dann zu legen, wenn im GENIE ein stärkrtre Netzteil eingebaut wurde, das den ECB-Bus mitversorgen soll.



Der Stecker CN1 (VG64, ac, Stifte, gewinkelt) wird auf den ECB-Bus gesteckt. Für den Anschluß an den Systembus wurden 2 Varianten vorgesehen. Durch Einbau einer VG64 a,c-Buchsen- leiste bei CN2 von der Lötseite (Abb.7) muß durch einen entsprechenden Drahtverhau zum Gegenstück für den 50poligen Platinenrandstecker des CPU-Boards die Anordnung der Signale umsortiert werden. Wer allerdings ein ästhetisches 50poliges Flachkabel bevorzugt, kann CN2 unbestückt lassen und stattdessen in die nicht angeschlossenen Lötaugen von CN3 von der Lötseite einen geeigneten 50poligen Stecker einlöten. Dann wird aber der zum Umsortieren der Signale nötige Drahtverhau nur auf die Karte zwischen CN2 und CN3 verlagert.

Außer den reinen Bus-Signalen des Computers müssen bei Bedarf auch noch die aktiven Steuersignale der Eingangsarte an den Computer zurückgeührt werden und die im Computer abzugreifenden Freigabesignale an die Eingangskarte geführt werden. Dafür sind am oberen Platinenrand entsprechende Lötpunte vorhanden. Die wichtigeren Signale sind aber auch an Lötpunte von CN2 geführt, womit dann alle Signale über CN2 (und bei entsprechender Querverdrahtung auch CN3) mit dem Computer verbunden werden können. Dafür reicht dann aber der 50polige Platinenrandstecker des CPU-Boards mit seinen 3 freien Pins nicht mehr aus. Da muß dann noch eine zusätliche fliegene Steckverbindung herhalten, die diese zusätzlichen Signale übergibt.

Steckerbelegungen

CN1 (ECB-Bus) Reihe a	NR.	Reihe c	CN2 (vom Systembus) Reihe a	Nr.	Reihe c
GND	32	GND	GND	32	GND
BUSK*	31	RESET*	BUSAk*	31	RESET*
A9	30	MERQ*	CLOCK	30	nc
A13	29	CLOCK	MERQ+	29	RFSH*
RFSH*	28	A15	IORQ*	28	RD*
IORQ*	27	A12	M1*	27	WR*
#1	26	#1	HALT*	26	nc
#1	25	HALT*	nc	25	INTERN
#1	24	RD*	nc	24	INTEXP*
#1	23	#1	NMI*	23	INTECB*
-12V #2	22	WR*	A11	22	A8
01	21	INT*	A10	21	A14
M1*	20	INT	A13	20	A9
#1	19	#1	A12	19	A15
A14	18	A10	nc	18	nc
#1	17	A11	nc	17	nc
#1	16	IEO	nc	16	nc
-5V	15	#1	nc	15	nc
#1	14	D1	A7	14	A6
+12V	13	#1	A5	13	A4
#1	12	A14	A3	12	A2
BUSRQ*	11	IEI	A1	11	A0
WAIT*	10	#1	BUSRQ*	10	nc
A6	9	A7	nc	9	WAIT*
A5	8	A8	nc	8	nc
A4	7	A1	D5	7	D0
A2	6	A3	D6	6	D7
D4	5	A0	D3	5	D2
D3	4	D2	D4	4	D1
D6	3	D7	nc	3	INTCPU*
D5	2	D0	nc	2	nc
+5V	1	+5V	+5V	I	+5V

#1 diese Pins sind zum Teil nicht belegt oder führen Signale die hier nicht benutzt werden

#2 die Lage von -12V wird nicht einheitlich gehandhabt, diese Spannung kann auch woanders liegen (muß bei jeder gekauften Karte geprüft werden)

Stückliste

ICs;
- 4 x 74LS245
- 1 x 74LS174
- 1 x 74LS139
- 1 x 74LS125
- 1 x 74LS88
- 1 x 74LS32
- 1 x 74LS30
- 1 x 74LS11
- 1 x 74LS08
- 1 x 74LS00

Widerstände;
- 2 x 10k

Sonstiges;
- 1 x VG64 a,c Stifte gewinkelt
- 1 x VG64 a,c Buchsen gerade
- 1 x Leiterplatte jede Menge Draht viel Geduld

Literatur

<1> Verbindend - Der ECB-Bus / Johannes Assenbaum / c't 12/85, 60-61

<2> Z80 CTC - Einsatz und Progmmmierung / Rolf Keller / c't 4/85, 92-94

<3> Interrupt-Priority-Encoder und Vektorgenerator für IM2 mit dem Z80 CTC / H. Bernhardt / Info des GENIE/TRS 80- User-Club Bremerhaven 32/85, 10-11

<4> Direkter Speicherzugriff; Schneller als die CPU - So arbeiten DMA-Controller-ICs / Rolf Keller / c't 8/85 80-83

<5> Z80 DMA - Schnell, aber compliziert / Rolf Keller / c't 12/85, 76-79

<6> Hardware-Umschaltung für den Betrieb von CP/M bei den Computern TRS 80 GENIE I und II und Komtek 1 / H. Bernhardt / Info des GENIE/TRS 80-User-Club Bremerhaven, Hardware-Sonderheft 11/85, 1-5

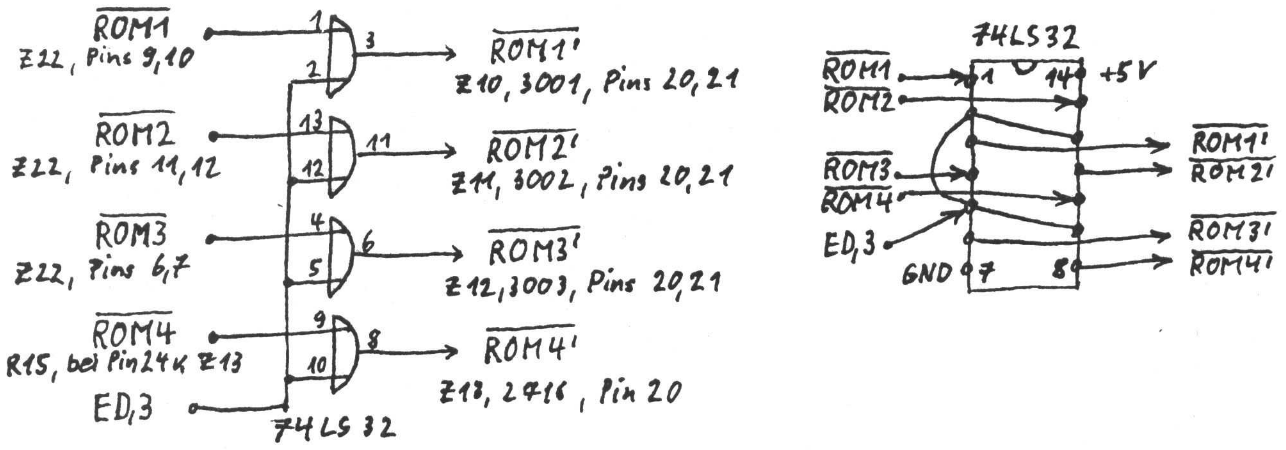
<7> GENIE I und II: Voll decodierter Sonder-ROM; RAM im Bereich 3900H bis 3BFFH / H. Bernhardt / Info des GENIE/TRS 80-User-Club Bremerhaven 11/85,...?

<8> Interrupt Your 80 - Without a Hardware Mod Your 80 is immune to rude Interruptions / Douglas C. Fisher / 80 micro, January 1983, 258-266

<9> CP/M Hardware für TRS 80 GENIE I und II und Komtek 1 / H. Bernhardt, C. Ueberschaar / Info des TRS 80/GENIE-User-Clubs Bremerhaven 02/86, 11-17

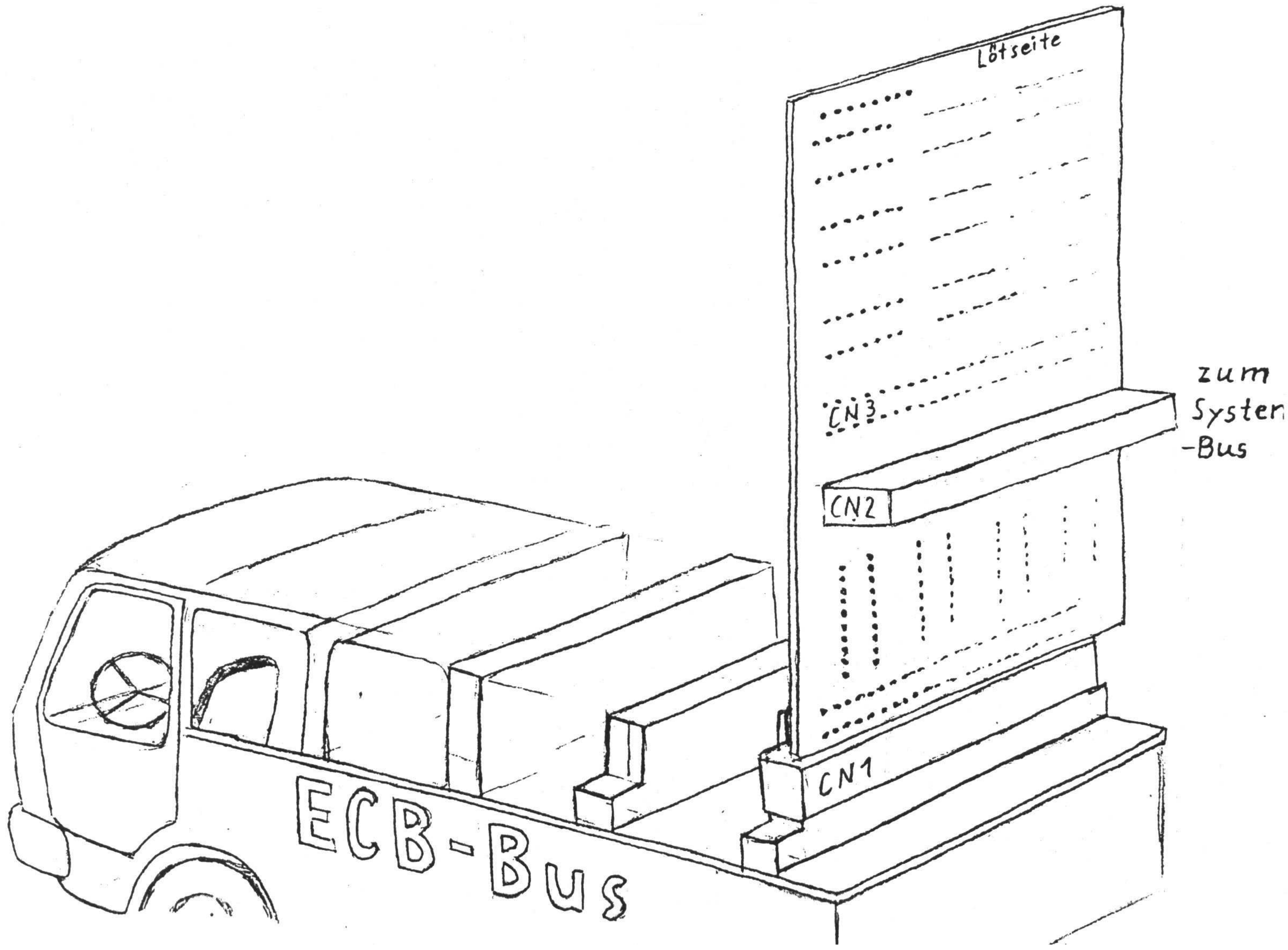
<10> 255K (1M-) RAM für Z80-Systeme / H. Bernhardt / Info des TRS 80/GENIE-User-Clubs Bremerhaven 03/86, 5-10

Abb.6) Beschaltung des Huckepack 74LS32 auf dem CPU-Board zum Ausblenden der ROMs



ROM1* Z22, 74LS156, Pins 9,10 ROM1'* Z10, 3001, Pins 20,21
ROM2* " " " 11,12 ROM2'* Z11, 3002, Pins 20,21
ROM3* " " " 6,7 ROM3'* Z12, 3003, Pins 20,21
ROM4* R15, Seite zu Pin24 von Z13 (R4) ROM4'* Z13 2716, Pin 20
diese Signale sind an die aus der Fassung gebogenen Pins zu löten

Abb.7) Verbinden des ECB-Bus mit dem Systembus des Computers über die Eingangskarte



ECB-BUS Belegung (CONITEC-Norm)

```
Reihe  a      b      c           Bedeutung:

1     +5U     xxx    +5U     :   +/-5V/12V:  Betiebsspannungen
2      D5     xxx    D0      :   UBAT:       Akku C+3,6 Volt
3      D6     xxx    D7      :   GND:        Gemeinsame Masse
4      D3     xxx    D2      :
5      D4     xxx    A0      :   DO-D7:      8-Bit-Datenbus
6      A2     xxx    A3      :   A0-A15:     16-Bit-Adressbus
7      A4     xxx    A1      :   A16-A19:    20-Bit-Adressen
8      A5     xxx    A8      :   A19/MS;     MASTER/SLAVE-Signal
9      A6     xxx    A7      :
10     /WAIT  xxx    xxx     :   /RD:        Lesen
11     /BUSRQ xxx ...IEI     :   /WR:        Schreiben
12     A18  ! xxx :  A19/MS !:   /MERQ:      Speichezugiff
13     +12V   xxx :  xxx     :   /I0RQ:      Ein-/Ausgabe
14     -12V ! xxx :  D1      :
15     -5V    xxx :  -       :   /NMI:       Vorang-Interupt
16     -      xxx :..IEO     :   /INT:       normaler Interupt
17     A17  ! xxx    A11     ;   IET:        INT-Ketteneingang
18     A14    xxx    A10     :   IE0:        INT-Kettenausgang
19     -      xxx    A16  !  :
20     /M1    xxx    /NMI    :   /BUSRQ:     DMA-Busanforderung
21     xxx    xxx    /INT    :   /BUAAK:     DMA-Busfreigabe
22     -      xxx    /WR     :   /BAI:       DMA-Ketteneingang
23  .../BAI ! xxx    -       :   /BAO:       DMA-Kettenausgang
24  :  UBAT   xxx    /RD     :   /RDY:       DMA-Handshake
25  :../BAO ! xxx    /HALT   :
26     /RDY ! xxx    /PCL    :   /RESET:     Rücksetz-Eingang
27     /IORQ  xxx    A12     :   /PCL:       Initialisieren
28     /RFSH  xxx    A15     :   /WAIT:      CPU-Wartezyklus
29     A13    xxx    ClOCK   :   /HAlT;      CPU gestoppt
30     A9     xxx    /MERQ   :   /M1:        1. Maschinenzyklus
31     /BUSAK xxx    /RESET  :   /RFSH:      Auffischzyklus
32     GND    GND    GND     :   CLOCK:      CPU-Takt (4-8 MHz)

xxx  reserviert für Erweiteungen   ...  Kettenleitungen (Daisy Chain)
-    zur freien Verwendung         !    abweichend von der KONTRON-Norm
```

ECB-Busnormen anderer Hesteller auf einen Blick

	CONITEC	E-LAB	ELZET	J&K	KONTRON	MARFLOW	miro	o+r
c10	xxx	A21	--	A16	MBSO	PHI/2	--	A16
a12	A18	A18	A18	/BAI	/BAI	A8	A18	/BAI
c12	A19	A19	A19	A7	MBS1	A19	A19	A17
c13	xxx	-12V	--	A18	MBS2	-12V	A20	A18
a14	-12V	/MMUDIS	-12V	A19	MBS3	xxx	--	A19
a16	--	2PHI	2PHI	2PHI	2PHI	nPHI	2PHI	--
a17	A17	A17	A17	/BAO	/BAO	A17	A17	/BAO
c19	A16	A16	A16	--	MBS5	A16	A16	--
a21	xxx	BDCL	--	--	MBS6	xxx	A22	--
a22	--	/RY	/BOOT	/RDY	MBS7	xxx	A23	--
a23	/BAI	/BAI	/BAI	/PF	DPR	/BAI	/BAI	--
c23	--	A20	/VBLANK	/SEL	MBS4	xxx	A21	--
a25	/BAO	/BAO	/BAO	--	nPHI	/BAO	/BAO	--
a26	/RDY	MEMD	--	DESLCT	WREN	xxx	/RDY	/DESLCT

Nebenstehende Tabelle ohne Gewähr. Achtung! Die unterschiedliche Belegung der -12V-Leitungen kann bei einigen Kombinationen von Karten und Bussen veschiedener Hersteller zu starker Rauchentwicklung bei der Inbetriebnahme fühen! Deshalb -12V vorsichtshalber immer vorher abklemmen.

1988-Club80-24, S.40-42

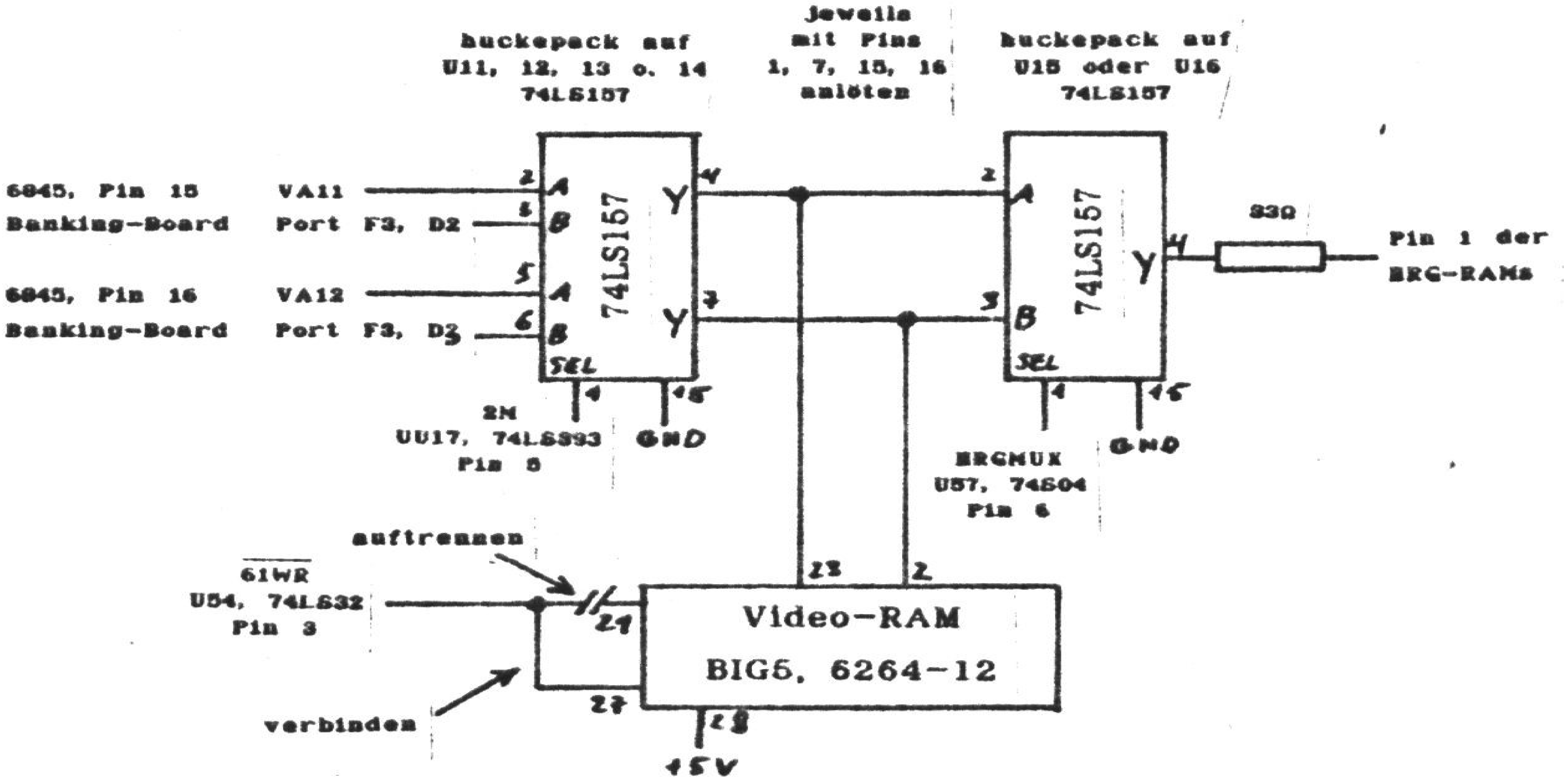
HRG, daß die Schwarte kracht

HRG, daß die Schwarte kracht!

Arnulf Sopp

Im Sonderheft über die Aufrüstung des Genie 3s stellten Helmut und ich unsere Erweiterung vor, mit der alle Speicherbaugruppen dieses Computers eine vierfache Kapazität erhielten. Dazu gehörten auch der Bildschirm mit 8 kB (zuvor 2 kB) und die Graphik mit 256 kB (vorher 64).

Viel später fiel mir auf, daß der Video-Controller (6845 CRTC) 14 Adreßleitungen hat. Damit könnte er sogar 16 kB Bildschirm verwalten. Er war aber noch immer wegen der ursprünglichen 2 kB nur mit 11 Leitungen verkabelt. So wurden die Addressen A11 und A12 des 6845 auch noch an den Bildwiederholspeicher angelötet. Das sieht nun so aus:



Nach dieser Operation ist die Hardware des Videoteils nun folgendermaßen konfiguriert: wie in alten Zeiten kann die CPU nur höchtens 2 kB von 3800-3FFFh gleichzeitig erreichen. Das langt z.B. für 80 X 26 Zeichen. Den Zugriff auf die vollen 8 kB erhält sie in Portionen zu 2 kB über die Bits 2 und 3 des Ports F3h (per OUT-Befehl). So ergeben sich aus der Perspektive des Z80 vier Einzelbildschirme zu 2048 Zeichen. Sie können selbstverständlich nacheinander mit einem zusammenhängenden Text befüllt werden. Das selbe gilt für die Graphik mit achtmal 32 kB oder zweimal 128 kB.

Der CRTC aber ist über 13 Adreßleitungen direkt mit dem Video-RAM verbunden; für ihn ist es ein einziger Speicher, den er stückweise oder auch ganz anzeigen kann (sofern der Monitor mitspielt - meiner tut es nicht: s.u.). Daher besteht nun die Möglichkeit, die CPU in einen Teilbereich des Bildschirms von max. 2 kB Größe werkeln zu lassen, während auf Wunsch wesentlich mehr angezeigt wird.

Nun war es theoretisch möglich, den kompletten Videospeicher von 8192 Zeichen anzuzeigen. Da wäre aber in dem Gedränge nicht nur nichts mehr zu erkennen. Die Hardware des Monitors (nicht des Computers) setzt auch Grenzen in der vertikalen und horizontalen Punktdichte und damit in der Anzahl der angezeigbaren Zeichen senkrecht und waagerecht.

Helmut und ich wollten es wissen. Zum Ausprobieren der maximal möglichen Auflösung ging ich von dem Bildschirmformat 92 X 22 Zeichen bei einer Zeichenhöhe von 12 Scan-Zeilen aus. 92 Zeichen ist bei meinem Monitor die höchstmögliche Zeilenlänge, was ich früher schon einmal ausgetestet hatte. Bei 22 Zeilen ergibt sich eine Bildspeichergröße von knapp 2 kB, also das alte Maximum. 12 Zeilen pro Zeichen erwiesen sich als Obergrenze denn bei 13 Scan-Zeilen begann mein Bildschirm zu schwimmen.

Um die größtmögliche Höhe des Bildschirms rauszukriegen, programmierte ich zunächst den Interlace-Modus. Dabei werden abwechselnd zwei Halbbilder angezeigt, um die Anzahl der möglichen Scan-Zeilen zu verdoppeln. Nach der Schimpansen-Methode (Rumprobieren) fand ich anschließend nach und nach die Maximaldaten für die Register 4-7 des CRTC heraus, um den Bildschirm auch vertikal bis zum Anschlag auszureizen.

Resultat waren 92 X 60 Zeichen, also fast fünfeinhalb kByte Fernsehen. Für ASCII-Texte wird man diese Format kaum je brauchen. Es ist witzlos, halbe Bibeln gleichzeitig auf dem Bildschirm zu haben. Für Graphik- Anwendungen kann jedoch eine etxrem hohe Auflösung sehr interessant sein. Bei einer Zeichenbreite von 8 Punkten und einer Höhe von 12 Dots gibt dieses Format eine Auflösung von 736 X 720 Pixeln her. Und das mal 2 weil der G3s zwei Graphikseiten speichern und anzeigen kann. Theoretisch sind sogar 1024 X 1024 Dots denkbar, was ein

Monitor mit hinreichender Bandbreite dann auch anzeigen sollte. Daher das überlebensgroße 'H' in der Überschrift.

Das im Anschluß gelistete Programm füllt den Bildschirm mit einem Graphik- muster, das erkennen läßt, wo welche 32-kB-Seite des Graphikspeichers beginnt. Zunächst werden die experimentell ermittelten Daten auf den CRTC ausgegeben. Es folgt ein wenig G3s-spezifisches Vorgeplänkel (ASCII-Bild- schirm aus- und Graphik einblenden). Danach werden über die Bits 2 und 3 des Port F3h nacheinander die 4 Bildschirme bzw. Graphik-Memories selektiert und mit dem Punktmuster des Bildschirmzählers beladen. Es entstehen senkrechte Streifen unterschiedlicher Länge und Dicke. Wo sich die Form eines Streifens ändert, beginnt ein neuer der 4 Bildschirme alter Größe.

Diese äußerst üppige Auflösung wird man wohl selten brauchen. Durchaus sinnvoll ist aber die Möglichkeit, in einem gesonderten Abschnitt des Bildschirms ständig ein Menü, eine oder mehrere Statuszeilen oder ein DIR in der Anzeige zu haben, während die CPU darunter oder darüber noch immer über volle 2 kB Videospeicher verfügen kann. Das Menü wird auch nicht durch einen Scroll oder CLS gelöscht, weil nur der CRTC, nicht aber die CPU gleichzeitig den vollen Bildwiederholspeicher von 8 kB beherrscht.

Arnulf Sopp

```
                ; Testroutine, mit der die höchstmögliche Graphikauflösung
                : beim Genie 3s mit Helmut Bernhardts Mega-Banker
                ; durch Probieren ermittelt werden kann
                ;
                ; Sopp

5200

5200  214152    start     LD      HL,data         ;Parametertabelle für den CRTC
5203  D1F710              LD      BC,10f7h        ;Zähler 16 Register, CRTC Datenport F7
5206  AF                  XOR     A               ;A <= 00h CRTC ab Register 0 programmieren
5207  D3F6      setcrtc   OUT     (0f6h),A        ;internes CRTC-Register adressieren
5209  3C                  INC     A               ;auf nächstes von 16 Registern stellen
520A  EDA3                OUTI                    ;Datum auf CRTC ausgeben, Zeiger weiter
520C  20F9                JR      NZ,setcrtc      ;falls noch nicht 16 Daten ausgegeben
520E  DBFA                IN      A,(0fah)        ;Systemport 1 des G3s
5210  F5                  PUSH    AF              ;Inhalt retten
5211  F602                OR      02h             ;Graphik anzeigen
5213  E67F                AND     7fh             ;ASCII-Bildschirm ausschalten
5215  F5                  PUSH    AF              ;diesen Zustand auch retten
5216  F608                OR      08h             ;Graphik für die CPU für RD/WR freigeben
5218  F3                  DI                      ;vorsichtshalber
5219  D3FA                OUT     (0fah),A        ;Systembyte 1 neu schreiben
521B  0604                LD      B,04h           ;Zähler für 4 Bildsch. des Vierfach G3s
521D  AF                  XOR     A               ;A <= 00h, ab Bildsch. 0 des 4fach G3s
521E  210080    fillscr   LD      HL,8000h        ;Anfangsadresse des Grafikspeichers
5221  54                  LD      D,H             ;DE <= 8000h = 32kB Graphik pro Bildsch.
5222  5D                  LD      E,L
5223  D3F3                OUT     (0f3h),A        ;einen v. 4 Gr.-Bildschirmen selektieren
5225  4F                  LD      C,A             ;Inhalt von Port F3h retten
5226  70        setpix    LD      (HL),B          ;Pixelmuster 04, 03, 02, 01 anzeigen
5227  23                  INC     HL              ;nächste Stelle des Graphikspeichers
5228  1B                  DEC     DE              ;Bytezähler herunterzählen
5229  7A                  LD      A,D             ;prüfen ob
522A  B3                  OR      E               ;der Zähler schon abgelaufen ist
522B  20F9                JR      NZ,setpix       ;falls nein: weiter Pixels anzeigen
522D  79                  LD      A,C             ;A <= Inhalt von Port F3h
522E  C604                ADD     A,04h           ;abgelaufen: nächsten Bildschirm selekt.
5230  10EC                DJNZ    fillscr         ;nächst. Bildsch. mir einem Muster füllen
5232  AF                  XOR     A               ;A <= 0, Bildschirm 0 des Vierfach G3s
5233  D3F3                OUT     (0f3h),A        ;zurückstellen
5235  F1                  POP     AF              ;Systenbyte 1 (noch alt angez. Graphik)
5236  D3FA                OUT     (0fah),A        ;ab 8000h wieder RAM einblenden
5238  CD490               CALL    0049h           ;auf irgendeinen Tastendruck warten
523B  F1                  POP     AF              ;ursprüngl. Inhalt von Port FAh (o. Gr.)
523C  D3FA                OUT     (0fah),A        ;restaurieren
523E  C3C901              JP      01c9h           ;CLS (Videoformat restaur.) EI und ENDE

                ; Die Daten für den CRTC können nach Belieben zum Test geändert werden

5241  70        data      DB      70h     ;Reg. 0: Bildschirmbreite112 Zeichen
5242  5V                  DB      5ch     ;     1: angezeigt 92 Zeichen
5243  61                  DB      61h     ;     2: horizontale Sync-Position
```

```
5244  08        00054           DB      08h     ;     3: Breite des Sync-Impulses
5245  20        00055           DB      20h     ;     4: Bildhöhe 64 Z. (Hälfte von Interlace)
5246  10        00056           DB      10h     ;     5: Bildhöhenjustage in Zeilen
5247  1E        00057           DB      1eh     ;     6: angezeigt 60 Zeilen (Hälfte v. Interl.)
5248  1F        00058           DB      1fh     ;     7: vertikale Sync-Position
5249  03        00059           DB      03h     ;     8: Interlace-Modus
524A  08        00060           DB      0bh     ;     9: Zeilen pro Zeichen -1 (=12 lpc)
524B  29        00061           DB      29h     ;    10: Cursor: oberer Rand und Anzeigemodus
524C  09        00062           DB      09h     ;    11: dto. unterer Rand
524D  00        00063           DB      00h     ;    12: Bildschirmanfang (relativ zu 3800h) MSB
524E  00        00064           DB      00h     ;    13: dto. LSB
524F  00        00065           DB      00h     ;    14: Cursoradresse (relativ zu 3800h) MSB
5250  00        00066           DB      99h     ;    15: dto. LSB
                00067
5200            00068           END     start   ; dort Einsprung

00000 Fehler

data    5241    fillscr 521F    setcrtc 5207    setpix  5226    start   5200
```

1988-Club80 Nr.25 S.57-59

Prof 180x und mc68000-ECB

Helmut Bernhardt

Der mc68000-ECB aus mc 1-3/87 ist ein Coprozessor-Board mit einem Motorola 68000 Prozessor auf einer Europakarte für den ECB-Bus. Diese Karte kann in jedem DMA-fähigen ECB-Bus-Computer eingesetzt werden, wobei dann abwechselnd oder auch gleichzeitig Z80- und 68000-Software gefahren werden kann. Im einfachsten Fall kann das 68000-Board aber auch nur dem Z80 eine bis 2MB große RAM-Floppy zur Verfügung stellen.

Wenn der Computer im 68000-Modus läuft, spielt der Z80 (oder im Prof 180x der HD64180) I/O-Knecht für den 68000. Der 68000 bedient sich des BIOS des CP/M 80. Der 8Bit-Prozessor wartet nur auf Aufgaben, die ihm der 68000 stellt.

Die Kommunikation zwischen den beiden CPUs ist für beide unterschiedlich zu handhaben. Der 8Bit-Prozessor kann dem 68000 einen 1Byte-Befehl in ein Latch schreiben, während der 68000 per DMA in den Speicher des Wirts-Computers schreiben und auch daraus lesen kann. In den 16MB-Adreßraum des 68000 ist ein Fenster von 2MB Größe gelegt, in dem die vermeintlichen 64KByte des Z80-Systems erreichbar sind. Zur Adressierung im Z80-System werden dann die Adressen A1-A16 des 68000 auf A0-A15 des ECB-Bus geschaltet und BUSAK* low gezogen. Außerdem werden auch die Z80-kompatiblen Signale MERQ* IORQ*, RD* und WR* erzeugt und der Datentreiber des 68000-Boards zum ECB-Bus wird freigegeben (für D8-D15 des 68000).

Der Prof 180x hat aber nicht nur 64KBte RAM; zur Adressierung innerhalb der 512K des Prof 180x müssen noch die Adressen A16-A18 mit festen Pegeln für den Bereich 40000H-4FFFFH erzeugt werden. In diesem physikalischen Adreßbereich liegt im Prof die TPA, in der ein Kommunikationsprogramm den Datenaustausch mit dem 68000 erledigt und von wo der 68000 sich Daten abholt und wohin er seine Daten an das 8Bit-System überträgt. Das läßt sich noch recht einfach mit einem 74LS367 Tri- state-Treiber, der A16-A18 des ECB-Bus bedient, auf dem 68000-Board erledigen. Dieser Treiber wird mit den Pins 1, 8 und 16 auf IC33, 74LS367 aufgelötet. Die Pins 3, 5 und 7 werden mit A18, A17 und A16 des ECB-Bus verbunden und an Pin 2 wird ein Widerstand von 4K7 gegen +5V angeschlossen. Die Pins 4 und 6 werden an GND gelegt.

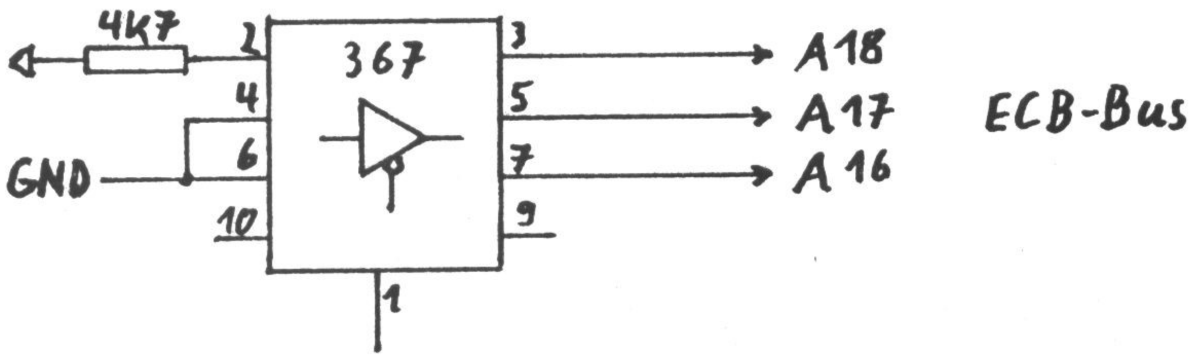


Abb.1: Erzeugen der richtigen Level für A16-A18 für die Adressierung der TPA beim Prof 180x bei physikalisch 40000H-4FFFFH

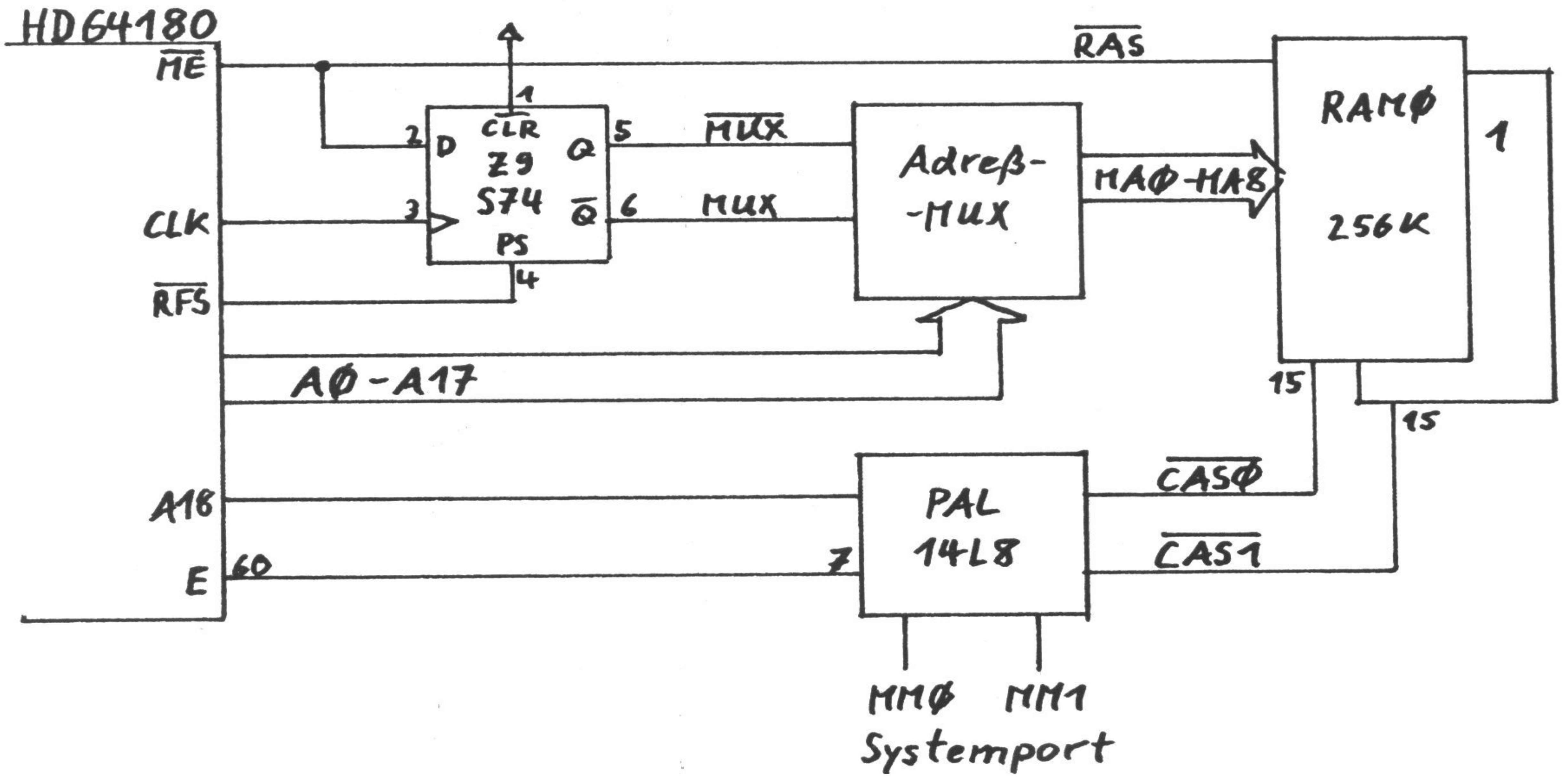
Das eigentliche Problem war weder vorher abzusehen noch durch einen einfachen Patch zu beheben - so schien es zumindest im ersten Augenblick. Die DMA-Fähigkeit des Prof 180x brücksichtigt zunächst die Steuerung der Treiber zum ECB-Bus. Durch ein PAL 14L8 ist die Bussteuerung ohne den sonst für die Berücksichtigung der DMA-Fähigkeit nötigen TTL-Aufwand mit folgender PAL-Gleichung für das Steuersig- nal des Datentreibers gelöst:

```
/DIR = /IOE * /RD * /ADR * A7 * /A5     ;Treiberrichtung bei Lesen interner
                                        ;Ports immer zum BUS hin
     + /ME * /RD * A19                  ;zum Bus bei Lesen der internen 512K
                                        ;A19 ist ein im gleichen Pal erzeugtes
                                        ;Signal zur Unterscheidung von internem
                                        ;und externem RAM
     + RD * LIR * BUSAK                 ;zum Bus auch, wenn weder ein Interrupt-
                                        ;Achnowledge noch DMA-Zugriff noch eine
                                        ;Schreiboperation erfolgt
```

Die Treiber für den Adreßbus und Steuerbus werden einfach durch den Pegel von BUSAK* gesteuert, Wenn BUSAK* low aktiv ist, treiben diese die entsprechenden Signale vom Bus zur CPU-Karte.

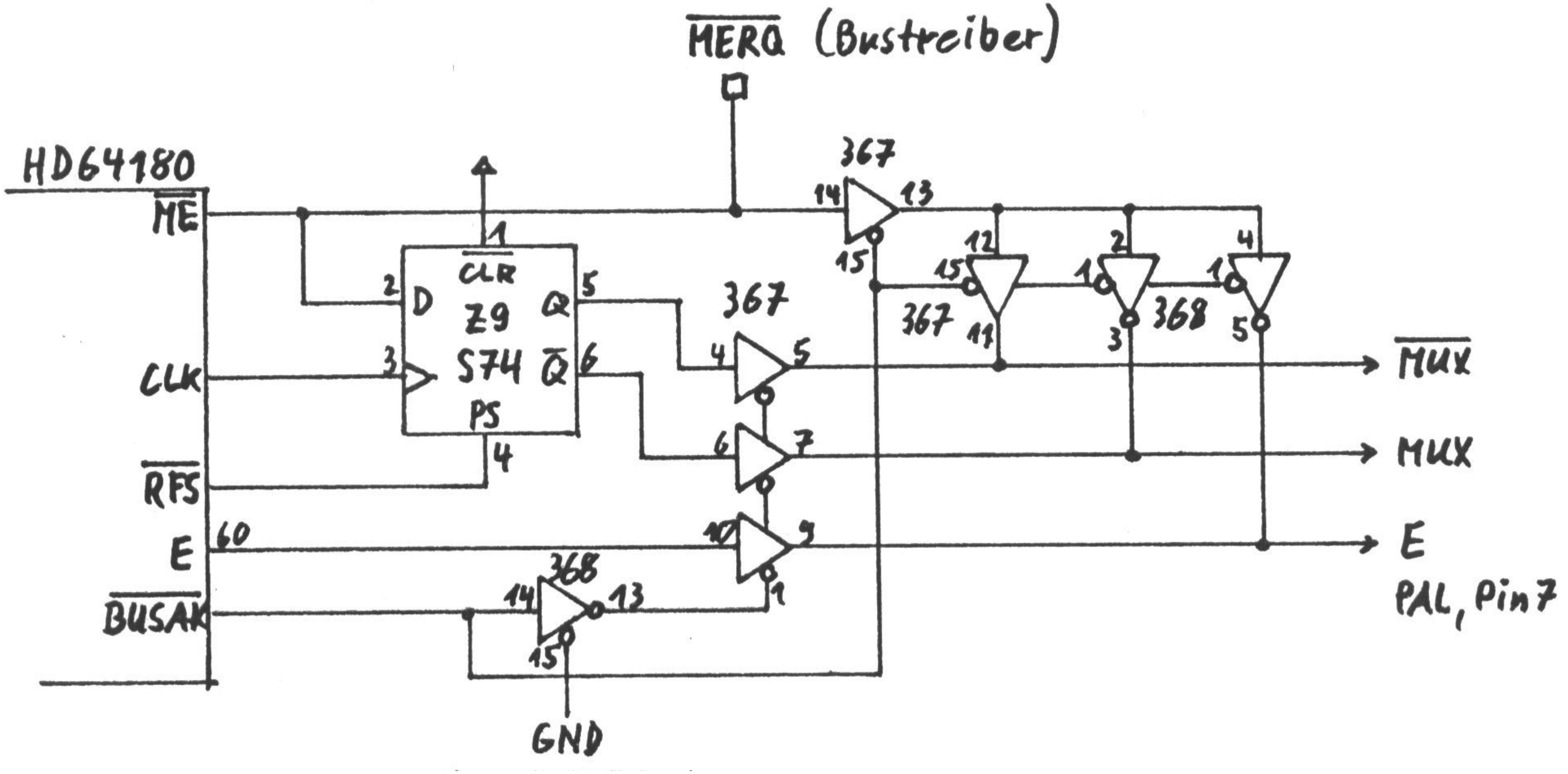
Damit kann ein externer DMA-Controller zwar auf das CPU-Board zugreifen aber noch nicht den Speicher lesen und schreiben, denn das flexibel bei jeder CPU-Taktrate funktionierende Timing der RAM-Steuersignale RAS*, MUX und MUX* und CAS*, das aus HDD4180-spezifischen Signalen abgeleitet wird, kann von einem DMAC nicht unterstützt werden. Die im Prof 180X verwendete Schaltung (Abb.2) zeigt, daß aus dem MERQ* mit der nächsten steigenden Flanke des Systemtaktes die Signale MUX und MUX* erzeugt werden und daß ein vom Systemtakt abhängiges HD64180-speifisches Signal E für das Timing des CAS*-Signals herangezogen wird. Ein asnchron arbeitender DMAC kann diese Signale garnicht bedienen, E ist ein ECB-Bus-Signal und der CPU-Takt ist kein Tristate-Signal, das beim DMA-Zugriff vom DMAC bedient werden kann.

Abb.2: das für jede Taktrate funktionierende Timing der RAM-Steuersignale im Prof 180x



Es galt nun, möglichst unter Beibehaltung des eleganten und flexiblen Timings der RAM-Steuerung durch den HD64180, einem externen DMAC eine weitere Möglichkeit zu schaffen, ausschließlich mit MERQ* (=ME* beim HD64180) und Verzögerungsgliedern die getimeten RAM-Signale zu erzeugen. Wenn dies z.B. mit Gatterlaufzeiten gemacht wird, sind dann allerdings der Bandbreite möglicher Tatraten des DMAC engere Grenzen gesetzt. Die in Abb.3 dargestellte Schaltung erlaubt das durch BUSAK* gesteuerte Umschalten zwischen ursprünglichem Timing des Prof 180x und über Gatterlaufzeiten geregeltes Timing bei Kontrolle durch den externen DMA- Controller. Die Anzahl Gatter für die Verzögerung gilt für einen mit 8MHz getate- ten 68000. Bei Verwendung entsprechend schneller RAMs und EPROMs auf dem 68000- Board macht diese Schaltung aber auch bei einem mit 16MH getakteten 68000 mit.

Abb.3: auf einem Adapterplatinchen im Sockel von Z9, 74S74 wird die Umschaltung zwischen den im Prof 180 erzeugten und den aus MREQ* abgeleiteten RAM-Steuersignale untergebracht, BUSAK* regelt die Umschaltung



Es sei noch erwähnt daß im DMA-Betrieb MUX und CAS* nicht gleichzeitig kommen, wie die Schaltung beim ersten Hinsehen erwarten läßt. Das im PAL aus E erzeugte CAS* ist gegenüber E nochmals um die Gatterlaufzeit des PALs verzögert.

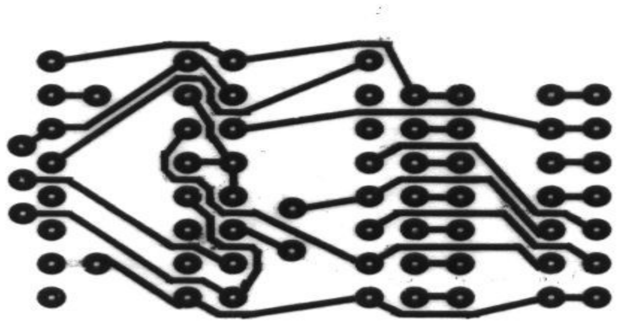


Abb.4: Layout des Adapterplatinchens, Ansicht von der Lötseite

Die Schaltung sieht auf den ersten Blick nach sehr viel Flick- und Sägearbeit am CPU-Board aus. Das ist aber durch Verwenden eines kleinen Adapterplatinchens, das in den Sockel von Z9, 74S74 gesteckt wird und Z9 selbst mit aufnimmt, nicht der Fall. Außer dem 74S74 bringt das Platinchen noch ein 74LS367 (Tristate-Treiber) und ein 74LS368 (invertierender Tristate-Treiber) unter, über den Sockel von Z9 werden fast alle benötigten Signale (MERQ*, CLK, RFS*) dem Platinchen zugeführt und auch die jeweils erzeugten Signale MUX und MUX* abgeleitet. BUSAK* muß durch freie Verdrahtung von J9 und E von Pin60 des HD64180 bezogen werden. Das Ausgangssignal E wird an Pin7 des PALs gelegt. Damit die bisherige Verbindung zwischen Pin60 des HD64180 und Pin7 des PALs unterbrochen wird, muß Pin7 des PALs aus der Fassung gebogen werden.

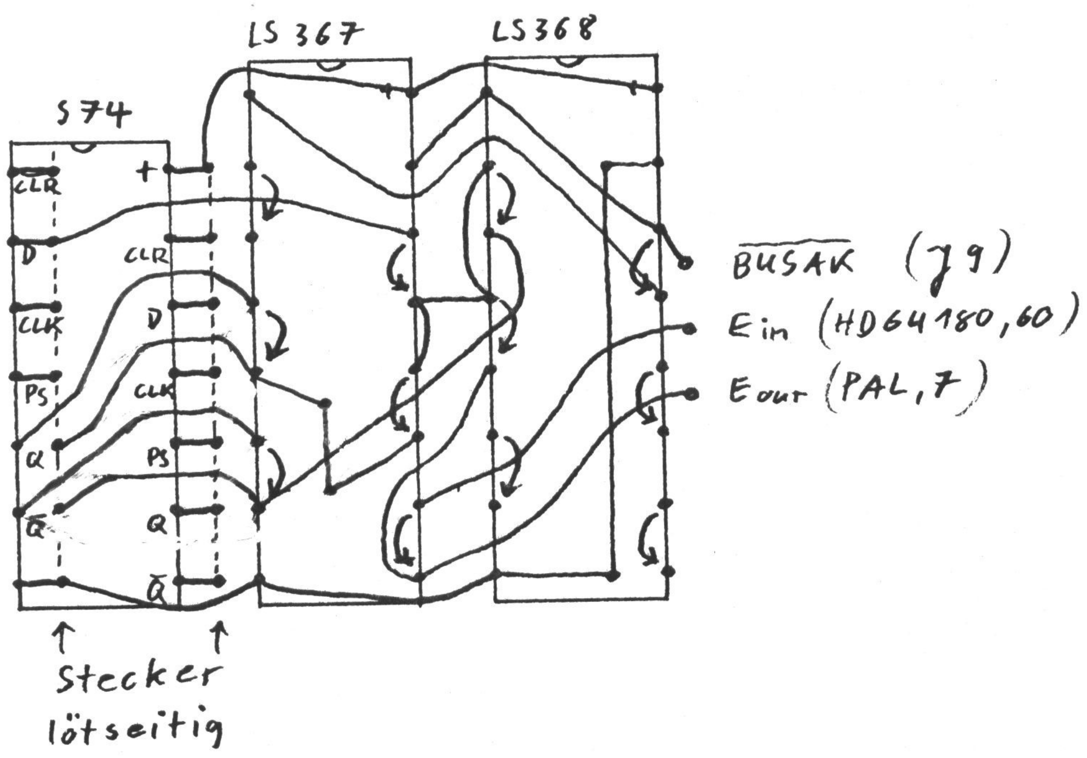


Abb.5: Adapterplatinchen, Planung des Layouts

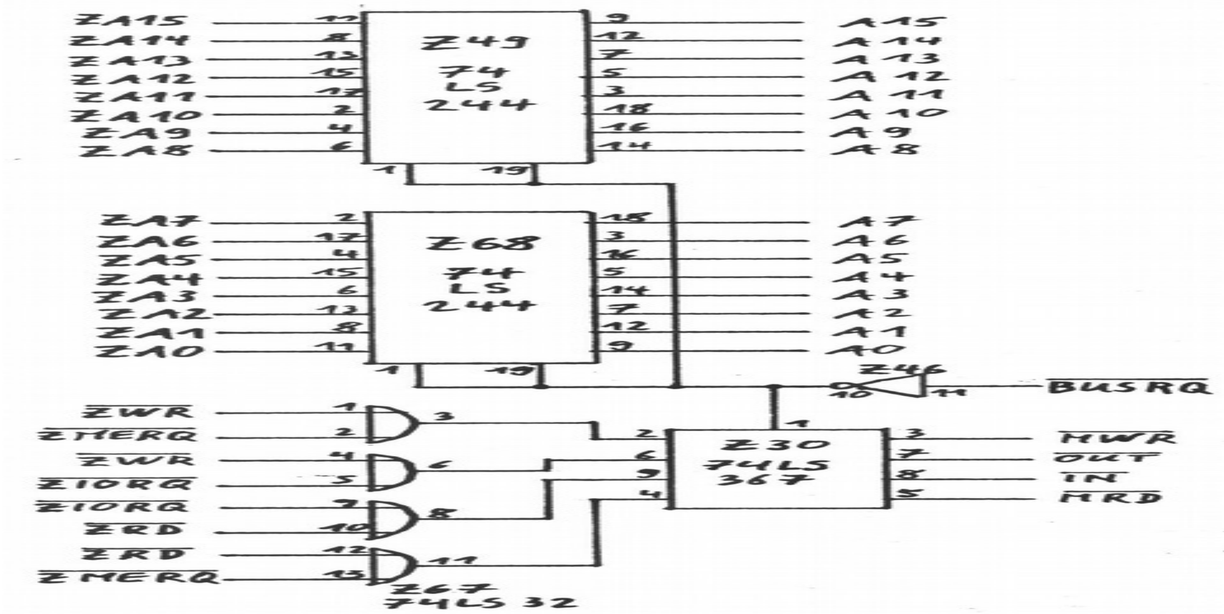
Eine ähnliche Herleitung von MUX und/oder MUX* weisen einige moderne TRS80- kompatible Computer auf, bei denen eine Taktumschaltung zwischen 1,77MHz und einer anderen, deutlich höheren Geschwindigeit möglich ist: Leider ist in all diesen Geräten ein nicht DMA-fähiger Bus konstruiert worden. Ein externer DMAC kann hier nur direkt mit den entsprechenden Signalen der CPU verbunden werden, Unverständlicheiweise werden dann aber die Treiber für Adreß- und Steuerbus doch durch ein invertiertes BUSAK* oder BUSRQ* freigegeben. Offensichtlich haben ganze Generationen von Konstruteuren von einem Urtrottel abgeguckt. Deshalb muß man dann zusätzlich noch die Freigabe dieser Treiber mit GND anstelle mit BUSAK bzw. BUSRQ einrichten.

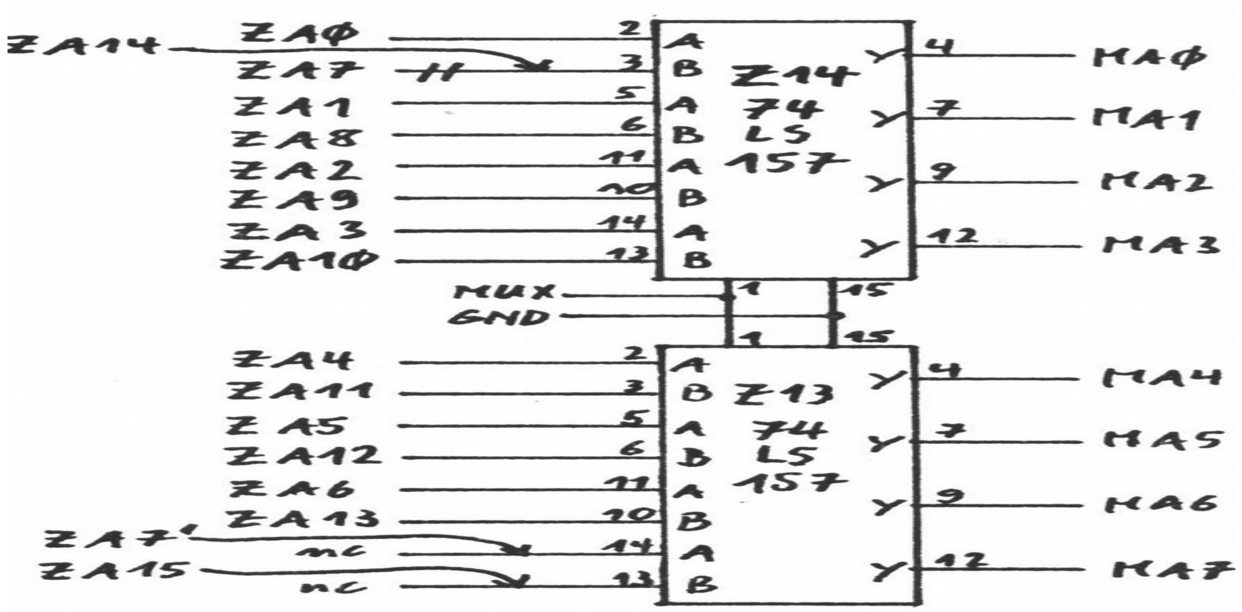
TRS 80 Japan Modell aufbohren

auf
1,77/3,55 MHz Takt-Umschaltung
CP/M-Fähigkeit
256K gebanktes RAM

H. Bernhardt

Aufgrund eines mir zunächst üppig erschienenen, sich aber nachträglich als keinesfalls dem Aufwand adäquat erwiesenen Honorars ließ ich mich breitschlagen, in einem vermeintlichen TRS 80 Modell 1 einen 256K-Banker, einen CP/M-Banker und eine einfache Taktumschaltung einzubauen. Bei Übernahme des Gerätes stellte sich aber heraus daß es sich um ein Japan-Modell (* siehe Ende des Artikels) handelt, bei dem alle meine Unterlagen zum TRS 80 unbrauchbar sind. Es galt also zunächst dessen Schaltung zu analysieren, um dann die nötigen Änderungen für die Erweiterungen vorzunehmen und zu sehen, an welchen Punkten die benötigten Signale abzugreifen sind. Solches hatte ich vor Jahren beim Komtek 1 schon mal durchgeführt, so daß mir der Mut nicht ganz verlorengegangen war.





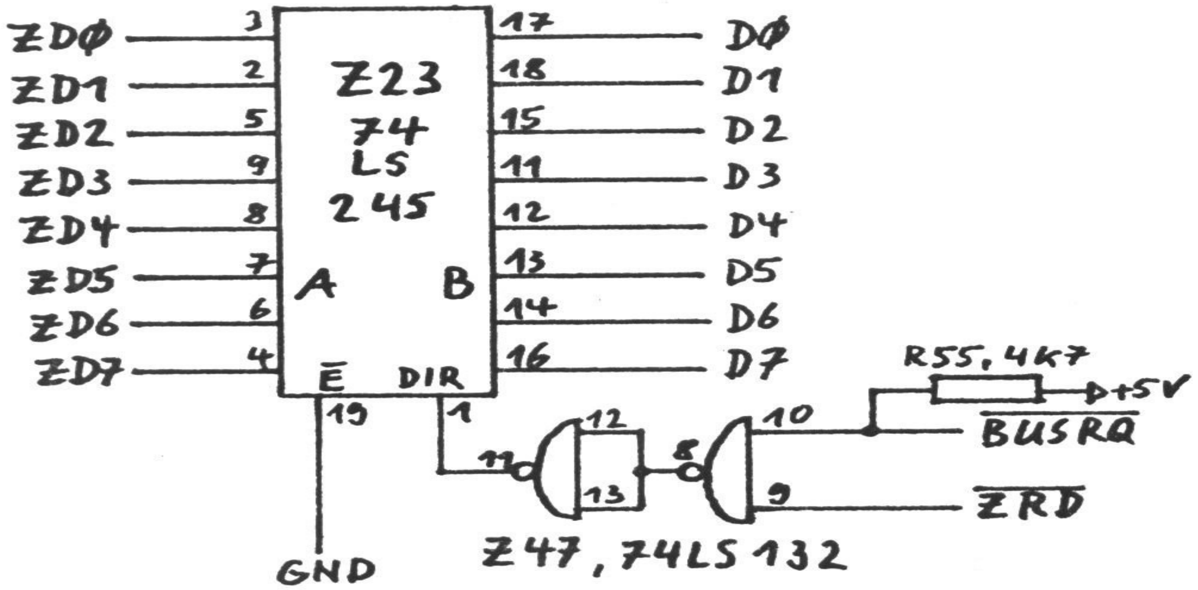


Abb.1:
links: Treiber für Adressen und Daten des Z80 (die mit Z.. bezeichnexen Signale sind die des Z80)
sowie Erzeugen der Signale MWR*, MRD*, 0UT* und IN*

rechts: Adreßmultiplexer für die RAMs (eingezeichnet sind die Änderungen für 4164-RAMs mit 7- oder 8-Bit-Refresh. Für 8-Bit-Refresh-RAMs muß ZA7' von einem Generator für 8-Bit-Refresh-Adressen kommen)

Eventuell besitzt hier im Club noch irgendjemand diese Variante des TRS 80 (vielleicht, ohne es selbst zu wissen, denn äußerlich sehen Original und Fälschung gleich aus), dem die von mir extrahierten Informationen etwas nützen. Deshalb will ich es riskieren, einige Seiten des kostbaren Platzes in diemem Info für diesen Zweck zu verbrauchen. Den 256K-Banker habe ich vor längerer Zeit schon einmal im Club 80 Info beschrieben, so daß hier nur noch dessen Einbau und Anschluß am CPU-Board breitgetreten werden soll.

Den CP/M-Banker (für den es ein angepaßtes CP/M 3,O BIOS gibt, zu dem auch die Source-Codes vorliegen) habe ich vor noch längerer Zeit zusammen mit Christof Ueberschaar entwickelt und im Info des mittlerweile eingegangenen Bremerhavener TRS 8O/GENIE-Clubs veröffentlicht. Um die zahlreichen Mitglieder des Club 80, die dort auch mal Mitlied waren, nicht noch

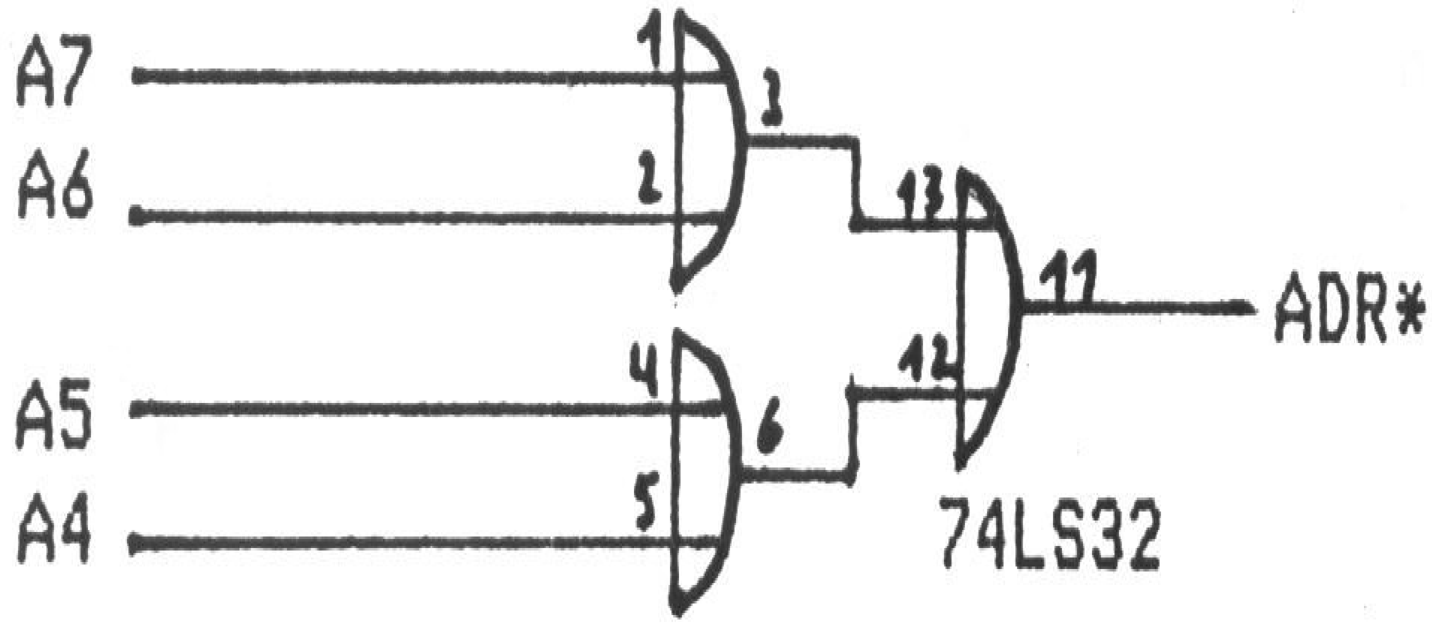
einmal damit zu langweilen, soll nur die Schaltung dieses CP/M-Bankers und eine Skizze mit den Anschlußpunkten auf dem Platinchen vorgestellt werden, wer darüber hinaus Interesse an dem Artikel hat, kann sich mit 1,90 DM Rückporto im Brief bei mir melden.

Der CP/M-Banker setzt voraus, daß im Grundgerät 64K oder 256K RAM vorhanden sind. Durch Ausgabe des Bytes 00H an den Port 50H wird der ROM-Bereich abgeschaltet und im Bereich 0000H-37DFH wird RAM verfügbar gemacht. Diese Einstellung läßt sich durch Software nicht wieder rückgängig machen. Nur ein RESET stellt wieder die urspünglichen ROMs bereit. Durch ein kleines Maschinenprogramm kann der ROM-Inhalt nach oben kopiert werden dann der Umschaltbefehl ausgeführt und schließlich der ROM-Inhalt an seine ursprüngliche Adresse zurückgeschoben werden. Damit ist das Level 2 ROM weich geworden und kann beliebig gepatcht werden.

Wenn an den Port 50H der Wert 40H (D6=1) ausgegeben wird, ist anschließend auch noch der memory mapped I/0-Bereich 37E0H-3FFFH nach F7E0H-FFFFH verlegt worden, so daß ein fast 64K großer RAM-Bereich ab 0000H vorliegt, womit CP/M leben kann.

Es sei noch gesagt, daß die Verwendung des Ports 50H (redundant decodiert auf 50H-5FH) bei gleichzeitig vorhandener HRG1B wegen deren Verschleiß an Portadressen (00H-7FH) eine feinere Decodierung der HRG1B erfordert. Prinzipiell genügt es, anstelle von A7 (an die HRG1B) ein aus A7 und A6 OR-verknüpftes Signal an die Karte zu führen, damit reduziert sich deren Portadreßhunger auf den Bereich 00H-3FH. Da dann aber ohnehin ein zusätzliches IC 74LS32 nötig ist, läßt sich die HRG1B auch noch auf die Portadressen 00H-0FH beschränken.

Abb.0: Dieses Signal reduziert den Portadreßhunger der HRG1B auf 00H-0FH wenn es anstelle von A7 die Port Freigabe steuert



Aber zurück zum Anschluß der Banker an das CPU-Board des Japan-Modells. Zunächst sollen die Änderungen besprochen werden, die den Einsatz von 64K bzw. 256K RAM Chips auf dem CPU-Board ermöglichen.

Auswechseln der 4116-RAMs

Dafür müssen die unsinnigerweise eingelöteten 4116-RAMs entfernt werden. Es werden alle Pins aller RAMs (Z15-Z22) möglichst hoch abgekniffen und die verbleibenden Stümpfe der Pins einzeln ausgelötet und schließlich die Lötlöcher mit einer Entlötpumpe freigemacht.

Vor dem Einlöten der DIL16-Fassungen für die neuen RAMs sind noch ein paar Änderungen nötig, um die +12V-Versorgung von den Pins 8 der RAMs und die -5V-Versorgung von den Pins 1 der RAMs zu entfernen und die +5V-Versorgung von den Pins 9 zu den Pins 8 der RAMs zu verlegen, und um an die Pins 9 ein neu zu erzeugendes gemultiplextes Signal MA7 (und bei 256K-Chips an die Pins 1 ein MA8) zu führen. Dafür sind folgende Arbeiten durchzuführen:

- Alle Kondensatoren (100nF) oberhalb und unterhalb der RAMs, die mit einem Beinchen Kontakt mit den Pins 1 oder 9 der RAMs haben, werden ausgelötet. Die bestückungsseitige, breite, die Pins 1 der RAMs verbindende Leiterbahn wird auf Höhe der 33-Ohm-Widerstände (R101) durchtrennt (-5V an den Pins 1 der RAMs).
- Die lötseitige, breite, um RAM Z15 und die 33-Ohm-Widerstände herum zu Pin 16 von Z14, 74LS157 führende Leiterbahn wird auf Höhe der 33-Ohm Widerstände (R101) durchtrennt (+5V an den Pins 9 der RAMs).
- Die vom Kondensator C15 auf der Lötseite zu Pin 8 von Z19 (RAM) verlaufende breite Leiterbahn wird neben C15 durchtrennt (+12V an den Pins 8 der RAMs).
- Eventuell müssen lötseitig noch einige zusätzliche Kondensatoren von 100nF über die Pins 8 und 16 der RAMs gelegt werden. Optimal ist ein 100nF- Kondensator je RAM. Diese Kondensatoren sollten nach dem Einlöten der DIL16- Sockel für die neuen 4164- bzw. 41256-RAMs lötseitig an die Pins 8 und 16 der Sockel gelötet werden. Es ist dafür zu sorgen, daß keine Kontakte zu anderen Pins entstehen (Isolierhülsen auf die Beinchen der Kondensatoren).
- Für den Einsatz von 4164-RAMs mit 7-Bit-Refresh reicht es, wenn A15 von Z48 Z80-CPU, Pin 5 an Pin 13 von Z13, 74L5157 und A14 von Z48, Z80-CPU, Pin 4 mit den untereinander

verbundenen Pins 9 der RAMs verdrahtet wird.

- Für den Einsatz von 4164-RAMs mit 8-Bit-Refresh oder 41256-RAMs (mit grundsätzlich 8-Bit-Refresh) muß A7 von Pin 3 von Z14, 74LS157 abgetrennt werden. Die lötseitige Leiterbahn zu Pin 3 von Z14 wird direkt neben dem Pin 3 durchtrennt.
- An diesen Pin 3 wird dann A14 von Z48, Z80-CPU, Pin 4 verdrahtet. A15 von Z48, Z80-CPU, Pin 5 wird mit Pin 13 von Z13, 74LS157 und Pin 12 von Z13 wird mit den Pins 9 der RAMs verdrahtet. A7' von einem 8-Bit-Refresh-Generator (z.B. auf dem 256K-Banker) wird an Pin 14 von Z13, 74LS157 gelegt.
- Im Expansion-Interface des Japan-Modells (schwarzes Blechgehäuse) werden alle RAMs 4116 aus den Fassungen gezogen und verworfen.
  Neben den RAMs befinden sich 2 Stück 74LS244-Treiber. Bei beiden ICs wird die bestückungsseitig von den Pins 1 wegführende Leiterbahn direkt neben den Pins 1 durchtrennt.
- Auf der Lötseite wird die die Pins 19 der 74LS244-ICs verbindende Leiterbahn neben der Durchkontaktierung beim 74LS32-IC durchtrennt. Bei beiden 74LS244- ICs wird eine Verbindung zwischen den Pins 1 und 19 hergestellt. Auf der Bestückungsseite wird an einem der 74LS244-ICs ein Widerstand von 2k2 von Pin 1 nach Pin 20 gelegt.

Taktumschaltung

Nach dieser Änderung sollte bei Verwendung von 4164-RAMs mit 7Bit-Refresh der Computer normal mit 48K RAM arbeiten. Da die langsamen RAMs im Expansion- Interface nun nicht mehr benutzt werden, läßt sich eine Speed-Umschaltung zwischen 1,77MHz und 3,55MHz einbauen. Die hier mit einem einfachen Umschalter realisierte Variante schaltet einen der beiden Takte zur CPU durch. Das

Umschalten ist also nicht während des Betriebs möglich (sollte es zumindest nicht, hier hat es in den meisten Fällen ohne Systemabsturz geklappt). Für diese Umschaltung sind folgende Operationen durchzuführen:

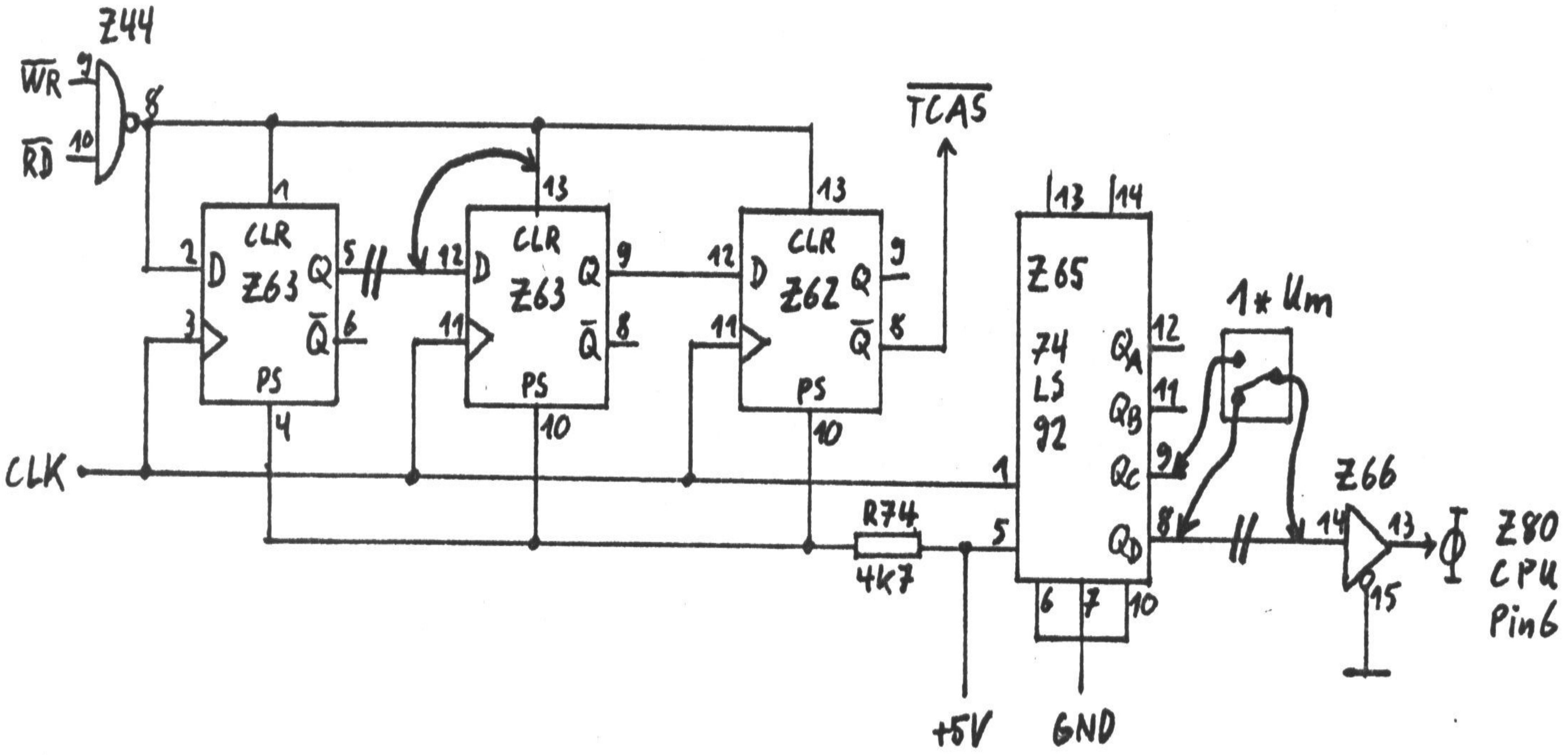


Abb.2: Änderungen für die Umschaltung des Systemtaktes zwischen 1,77 und 3,55 MHz

Auf der Bestückungsseite

- wird die Leiterbahn, die zwischen Pins 13 und 14 von Z65, 74LS92 herauskommt und zwischen Pins 2 und 3 von Z66, 74LS367 verschwindet, durchtrennt.

Die Pins 12 und 13 von Z63, 74LS74 werden miteinander verbunden.
Die Pins 8 und 9 von Z65, 74LS92 werden an die Außenkontakte eines Umschalters gelötet. Der Mittelkontakt des Umschalters wird mit Pin 14 von Z66, 74LS367 verbunden (möglichst kurze Leitungen).

Auf der Lötseite

- wird die Verbindung von Pin 12 von Z63, 74LS74 zur Durchkontaktierung unter dem IC durchtrennt.

CP/M-Banker einbauen

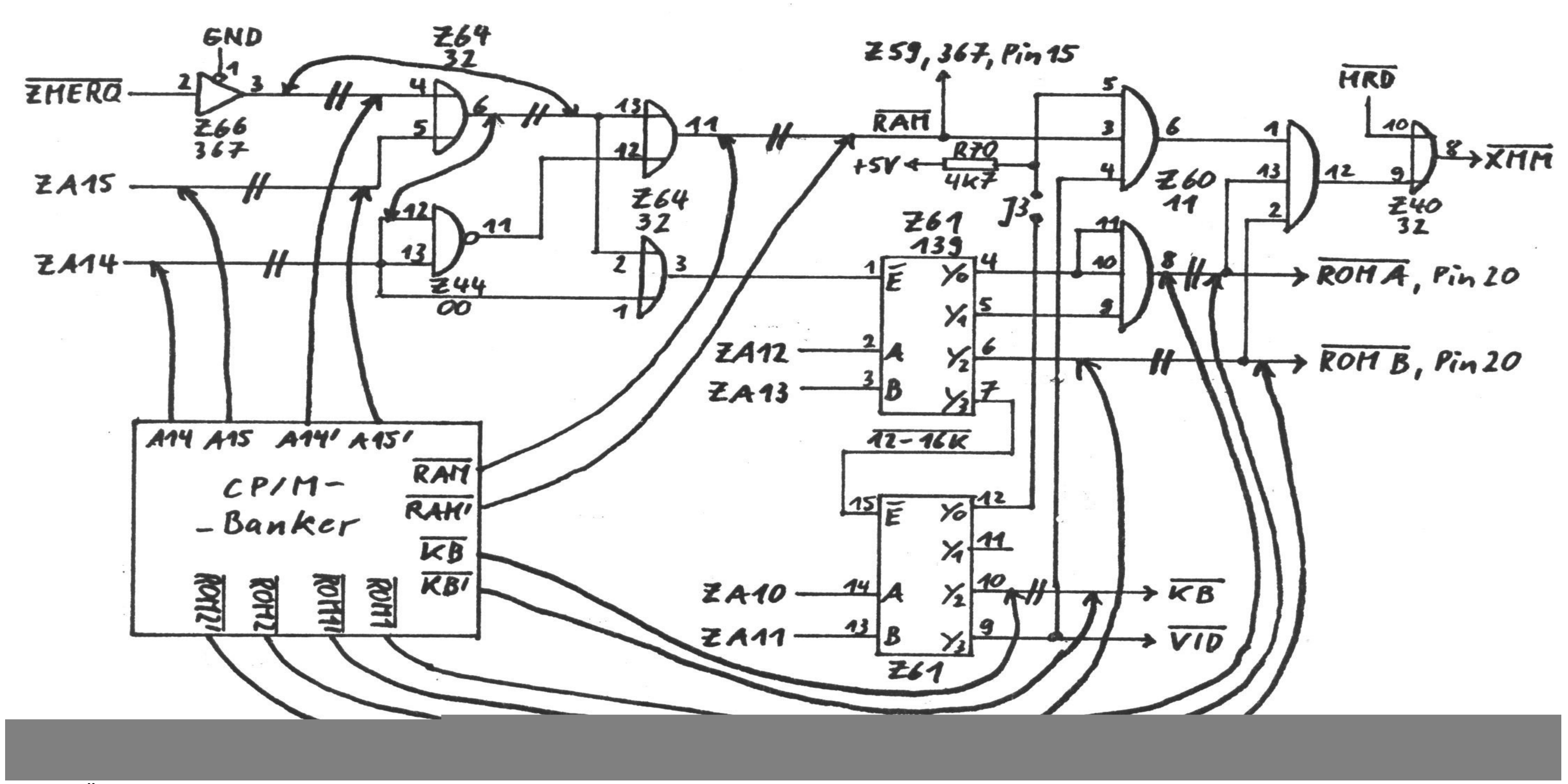


Abb.3: Änderung an der Adreßdecodierung des CPU-Boards und Anschluß des CP/M-Bankers

für den Anschluß des CP/M-Bankers sind die in Abb.3 dargestellten Änderungen in der Adreßdecodierung der unteren 16K auf dem CPU-Board vorzunehmen. Im einzelnen bedeutet dies folgende Arbeiten:

Bestückungsseite:

- Die Leiterbahn, die zwischen den Pins 2 und 3 von Z65, 74LS92 herauskommt und zwischen den Pins 11 und 12 von Z64, 74LS32 verschwindet, wird durchtrennt.
- Der Pin 3 von 266, 74LS367 wird mit Pin 13 von Z64, 74LS32 verbunden.
- Der Pin 6 von Z64, 74LS32 wird mit Pin 12 von Z44, 74LS00 verbunden. Die von Pin 12 von Z49, 74LS244 zur ca. 2 cm entfernten Durchkontaktierung
- führende Leiterbahn wird durchtrennt (A14).
- Die daneben verlaufende (zwischen den Pins 11 und 12 von Z49, 74LS244 herauskommende) Leiterbahn wird durchtrennt (A15).
- Die an Pin 10 von Z61, 74LS139 führende Leiterbahn wird neben dem Pin durchtrennt.

Lötseite:

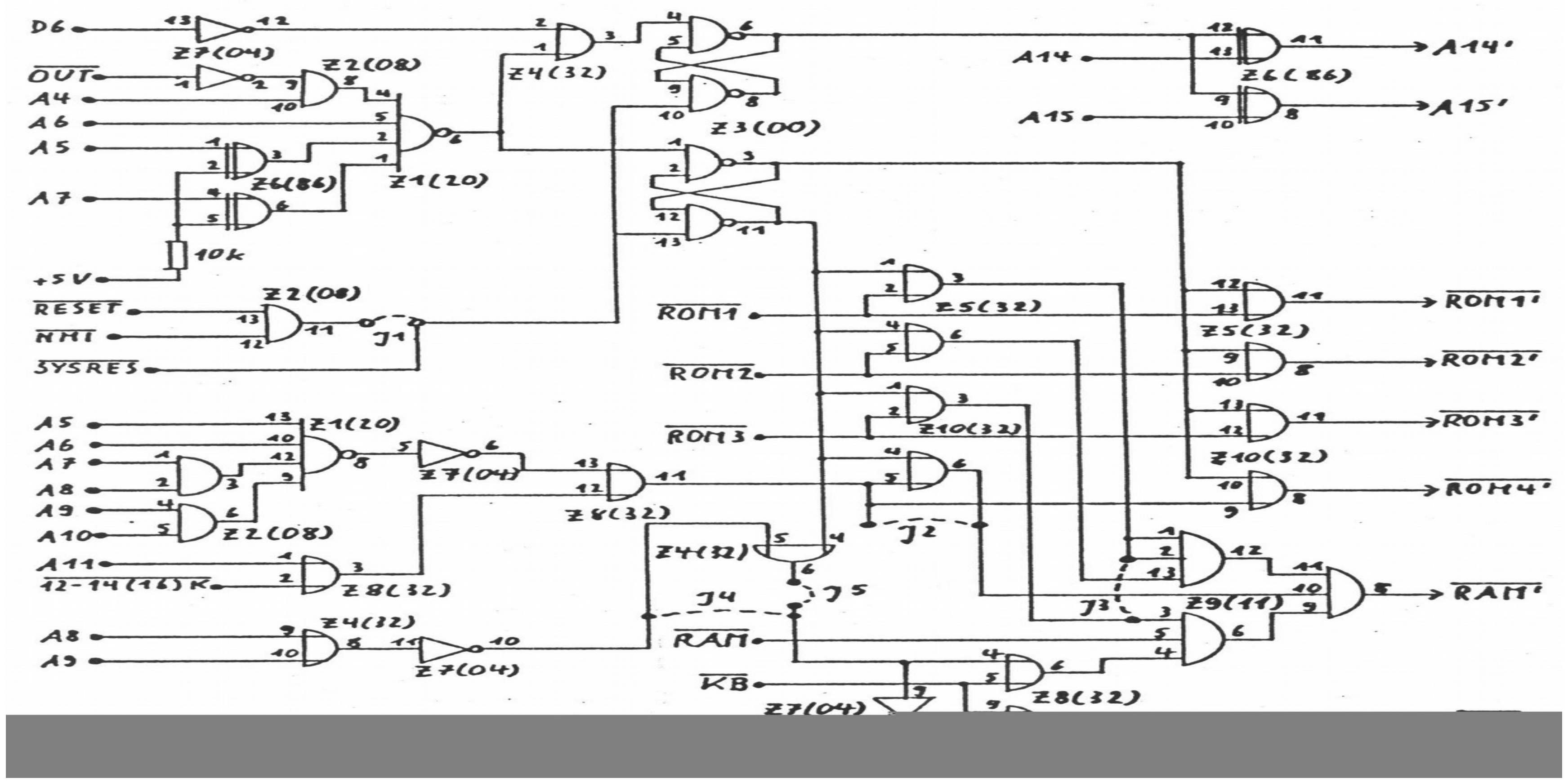
- Die Leiterbahn von Pin 11 von Z64, 74LS32 zur Durchkontaktierung bei Pin 8 von Z64, 74LS32 wird durchtrennt.
- Die Leiterbahn von Pin 13 von Z44, 74LS00 zur Durchkontaktierung wird durchtrennt.
- Die Verbindung zwischen den Pins 2 und 6 von Z64, 74LS32 wird durchtrennt.
- Die Leiterbahn von der Durchkontaktierung auf Höhe der Pins 5 und 10 von Z64, 74LS32 zur Durchkontaktierung auf Höhe der Pins 3,4,11 und 12 von Z44, 74LS00

- wird durchtrennt.

Auf dem CP/M-Banker:

- Das IC Z10, 74LS32 wird nicht bestückt und der Jumper J5 bleibt offen. Die Jumper J1, J2, J3 und J4 werden gelegt.
- Die Anschlüsse SYSRES*, ROM3*, ROM4*, ROM3'*, ROM4'* sowie 37EX* bleiben unberücksichtigt.

Abb.4: Schaltung des CP/M-Boards

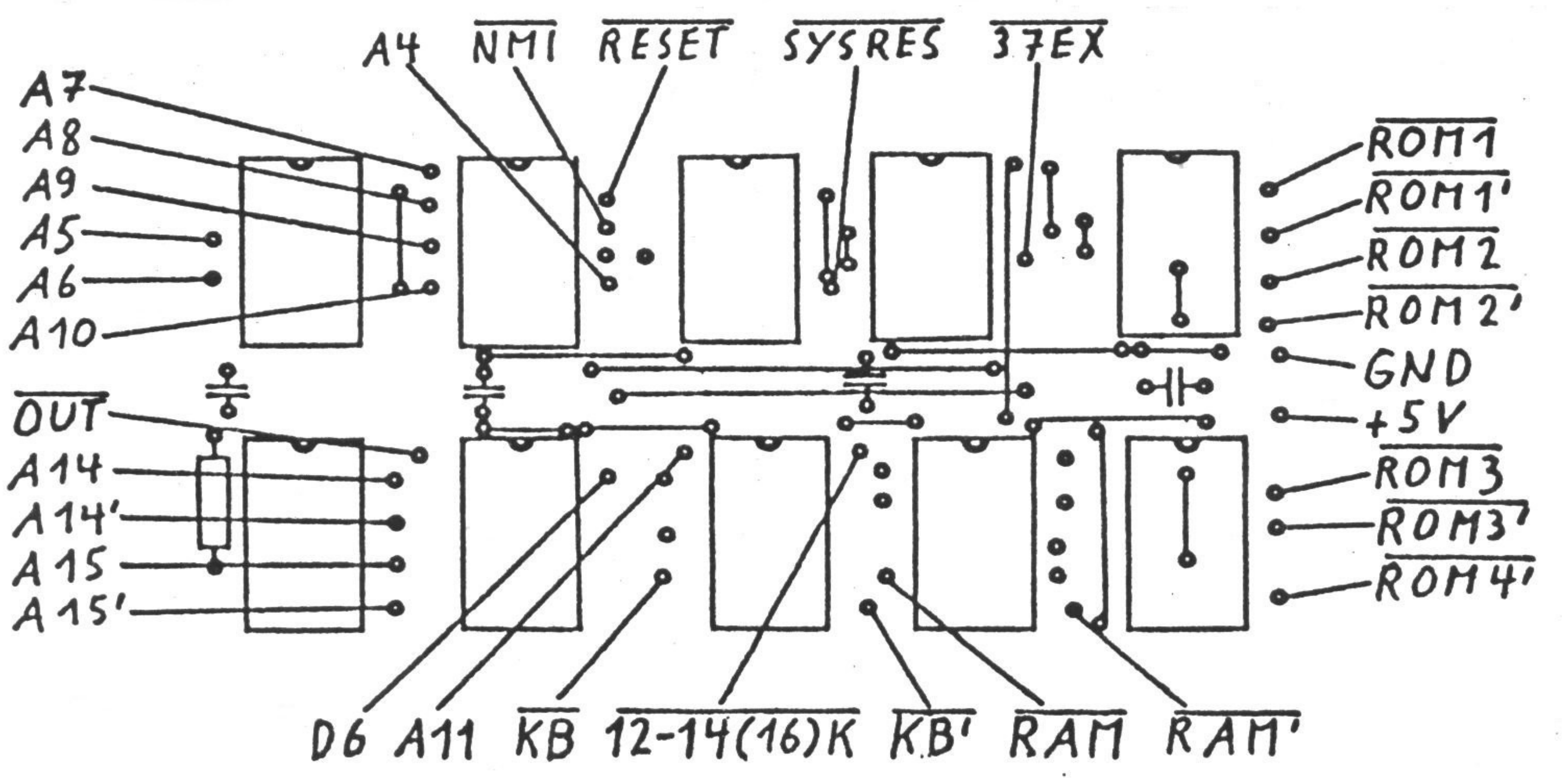


Der CP/M-Banker wird gemäß folgender Tabelle mit dem CPU-Board verbunden:

```
Signale auf dem    Anschluß auf dem
CP/M-Banker        CPU-Board
------------------------------------
   A4         Z68 74LS244 Pin5
   A5          "     "       16
   A6          "     "       3
   A7          "     "       18
   A8         Z49 74LS244 Pin14
   A9          "     "       16
  A10          "     "       18
  A11          "     "       3
  A14          "     "       12
  A15          "     "       9
------------------------------------
12-14(16)K*    Z61 74LS139 Pin15
0UT*           Z30 74LS377 Pin7
RESET*         Z46 74LL04  Pin12
NMI*           Z45 74LS02  Pin10
ROM1*          Z60 74LS11  Pin8
ROM2*          Z61 74L5139 Pin6
RAM*           Z64 74LS32  Pin11
KB*            Z61 74LS139 Pin10
------------------------------------
ROM3'*         Z42 ROM A   Pin20    diese Pins werden aus
R0M2'*         Z43 ROM B   Pin20    der Fassung gebogen
RAM'*          Z49 74LS367 Pin15
KB'*           Keyboard-Stecker Pin10
A14'*   an die Durchkontaktierung, deren Verbindung zu Pin 12
```

von Z49, 74LS244 durchtrennt wurde
A15'*   an die Durchkontaktierung direkt daneben

Abb.5: Anschlüsse an das CPU-Board



256K-Banker einbauen

Um den Anschluß des 256K-Banker vorzubereiten, wird auf Z14, 74LS157 noch ein weiteres IC 74L5157 mit den Pins 1, 8, 15 und 16 huckepack-gelötet; alle anderen Pins dieses ICs werden waagerecht abgebogen. An dieses IC Z14' werden die Signale A7', A15', A16 und A17 des 256K-Bankers angeschlossen

Signal des Bankers	Pin von Z14'
A7'	Pin14
A15'	" 13
A16	" 11
A17	" 10

Die Pins 9 der RAMs werden mit Pin 12 von Z14' und die Pins 1 der RAMs mit Pin 9 von Z14' verbunden. Die eventuell beim Umbau auf 64K RAM (event. mit CP/M-Banker) hergestellte Verbindung zwischen Pin12 von Z13 und den Pins 9 der RAMs wird wieder entfernt. Das anstelle der durchtrennten Leiterbahn (A7 der Z80-CPU) an Pin 3 von Z14 geführte Signal A14 (Pin 4 der Z80-CPU) bleibt angeschlossen.

Die Eingangssignale für den 256K-Banker werden an folgenden Punkten auf dem CPU-Board entnommen und an die entsprechend bezeichneten Punkte des 256K-Bankers geführt (freie Verdrahtung).

Signal	CPU-Board IC	Typ	Pin	256K-Banker Anschlußpunkt
D0	Z23	245	17	g
D1	"	"	18	f
D2	"	"	15	e
A0	Z68	244	9	8
A1	"	"	12	9
A2	"	"	7	1
A3	"	"	14	2
A4	"	"	5	10

```
A5        "    "   16        4 und 5
A6        "    "    3        6 und m
A7        "    "   18        7 und n
A15      Z49  244   9        h
--------------------------------------
SYSRE5*  (siehe unten)       a
0UT*     Z30  377   7        l
RFSH*    Z49 Z80-CPU 28      o
--------------------------------------
```

Bei "nur" 256K RAM werden die Eingangssignale D3 und D4 sowie CASEN* und die Ausgangssignale CASEN1* bis CASEN4* nicht angeschlossen.
Das Signal SYSRES* wird auf dem CP/M-Banker aus RESET* und NMI* erzeugt und kann von dort bezogen werden. Wenn der CP/M-Banker nicht eingebaut wird, muß RESET* (von Z46, 74LS04, Pin 12) oder NMI* (von Z45 74LS02, Pin 10) stattdessen verwendet werden.

Bei den spartanischen Platzverhältnissen im TRS 80 Modell 1 muß beim Einbau von Erweiterungen von vornherein geplant werden, wo die zusätzlichen Platinchen in der Schachtel Platz finden. Das CPU-Board liegt mit der Bestückungsseite nach unten im Gehäuse. Erweiterungen, die darunter liegen sollen, müssen also von der Bestückungsseite her angeschlossen werden. Platz ist unter dem CPU-Board nur in der hinteren Hälfte des Gehäuses vorhanden. Die Anschlußdrähte sollten also gleich so lang gewählt werden, daß diese Einbaulage möglich ist. Allerdings ist in digitalen Schaltungen prinzipiell das Einhalten kurzer Leitungslängen anzustreben. Man sollte also auch nicht zu großzügig mit der zu verwendenden isolierten Litze umgehen.

Obwohl so mancher rechtschaffener Standard-Z80 die 3,55MHz besser verträgt als mancher A-Typ, sollten doch die paar Mark und fünfzig für einen auf 4MHz ausgelegten Z80A investiert werden. Dagegen ist die Investition für die neuen RAM-Chips bei den heutigen Preisen durchaus die Überlegung wert, ob man das dem alten Möbel noch angedeihen lassen will.

Außerdem ist (auch wenn für einen potentiellen Nachbauer der Aufwand für die Analyse des Japan-Modells entfällt) der Arbeitsaufwand und die Wahrscheinlichkeit, einen Fehler zu machen, recht hoch. Der eigene Performance-Ehrgeiz sollte also ruhig überschlagen werden. Auch die CP/M-Fähigkeit ist bei einem nach wie vor auf 64x16 Zeichen beschränten Bildschirm begrenzt tauglich und Tasten für deutsche Umlaute hatte ein TRS 80 Modell 1 noch nie.

Anmerkung Juni 2015:
Damals gab es noch kein Internet und ich konnte mir keine Abbildung des TRS80 Japan Modells ansehen. Anhand des Unterschiedes des CPU-Boards zu mir zugänglich gewesenen TRS80 Geräten meinte ich, ein Japan-Modell vor mir zu haben. Das war aber etwas anderes.

Helmut

Bessere Darstellung von Text und Grafik mit HRG1B

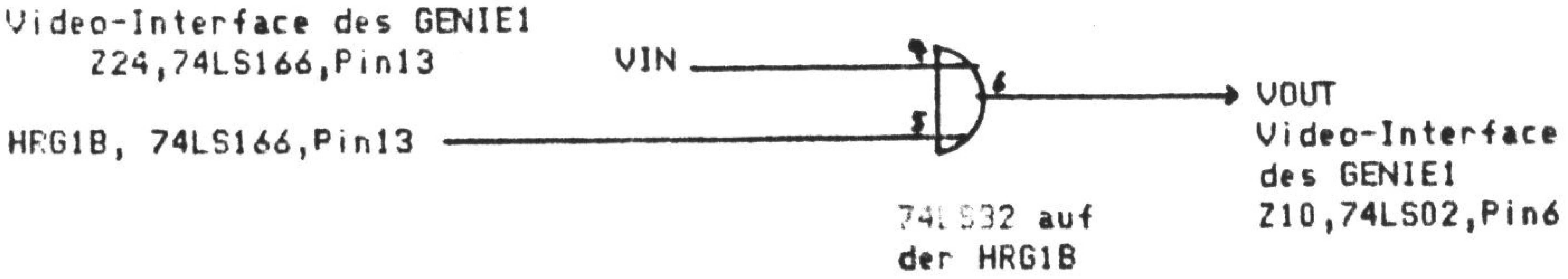
Helmut Bernhardt

Durch das Erzeugen synchroner Videosignale des normalen Text- Video-Interface und der HRG1B konnten diese beiden Signale gemischt werden, so daß beide Bildschirm-Informationen gleichzeitig dargestellt werden. Das ist einerseits ärgerlich, wenn z.B. BASICs READY> ständig nach der Erzeugung eines Bildes das Werk verunziert; es kann andererseits aber auch recht nützlich sein, wenn man zu Diagrammen erläuternde Texte anzeigen möchte. Weniger gelungen ist aber die Art, wie die Video-Signale gemischt werden. Das wird auf der HRG1B durch OR-Verknüpfung der beiden Signale gemacht, was zur Folge hat, daß in hellen Flächen der HRG-Anzeige kein Text zu erkennen ist. Die OR-Verknüpfung bewirkt, daß ein Bildpunkt immer hell gesteuert wird, wenn entweder der entsprechende Punkt in der HRG oder der entsprechende Punkt eines dargestellten Textzeichens oder beide angezeigt werden.

Eine XOR-Verknüpfung der beiden Video-Signale steuert einen Punkt nur dann hell, wenn entweder der HRG-Punkt oder der Punkt eines Textzeichens angezeigt wird. Wenn allerdings beide Videosignale gleichzeitig einen Bildpunkt hell darstellen wollen, wird dieser Punkt dunkel gesteuert. Das hat zur Folge, daß Texte vor hellem HRG-Hintergrund dunkel dargestellt werden. Selbst wenn ein Textzeichen zum Teil vor einem hellen und zum Teil vor einem dunklen Hintergrund dargestellt wird, läßt es sich vollständig erkennen, weil sein Farbwert immer das Gegenteil der Hintergrundfläche ist.

Damit können nicht nur Text und HRG sehr viel besser gleichzeitig dargestellt werden; der Umstand läßt sich bei geschickter Programmierung der HRG auch dazu nutzen, inverse Zeichen darzustellen. Es müssen einfach immer die entsprechenden Felder, in denen im Text-Bildschirm ein Zeichen dargestellt wird, in der HRG hell gesteuert werden. Das bedeutet, daß pro auszugebendem Textzeichen auch 12 Bytes im HRG-Speicher auf FFH zu setzen sind. Die Zuordnung von Text-Video-RAM zu HRG-Adressen läßt sich in früheren Beiträgen von Arnulf Sopp zu diesem Thema entnehmen. Etwas aufwendiger wird es beim Scrollen von Texten, wenn Teilbereiche des Textes invers dargestellt sind. Dann reicht es nicht, nur den ganzen Text-Bildwiederholspeicher um 64 Bytes nach oben zu verschieben, dann müssen auch die hellen Felder der HRG um einen entsprechenden Offset verschoben werden, was nicht mehr mit einem einfachen LDIR zu erledigen ist.

So aufwendig das in Soft auch sein mag, in Hard ist die Änderung auf XOR-Verknüpfung recht banal. Die bisherige OR-Verknüpfung sieht so aus:



Wenn anstelle des OR-Gatters 74LS32 ein XOR-Gatter des pinkompatiblen 74LS86 genommen wird, ist die Sache erledigt. Leider werden von dem 74LS32 noch andere OR-Gatter für andere Zwecke verwendet, so daß man nicht einfach das IC gegen ein 74LS86 auswechseln kann. Man kann aber das 74LS86 mit den Pins 4, 5, 7, 14 huckepack auf das 74LS32 löten, den Pin6 des 74LS32 abkneifen und stattdessen den Pin6 des 74LS86 mit dem Pin15 des Sockels für den Anschluß des 24poligen Flachkabels verbinden.

1989-Club80-10 Nr.28 S.73-77

Eleganter Adreßdecoder für GENIE 1

Helmut Bernhardt

Als ich bei der Fehlersuche in meinem GENIE 1 durch die zahlreichen selbstgeschnitzten Erweiterungen und das damit verbundene Strippengewirr selbst nicht mehr durchblickte, keimte der Gedanke, das alles sehr viel übersichtlicher und völlig neu zu gestalten. Als Ergebnis kam ein kleines Board heraus, das in den Sockel des Z80 zu stecken ist und den Z80 selbst mit aufnimmt. Zusätzlich wird auf dem Board ein 27128-EPROM (das anstelle der dann überflüssigen Original-ROMs das Level-II-Basic enthält), ein 74LS273-Latch, ein PAL16L8 (das ein Freigabesignal für das 74LS273 und einige zusätzliche Freigabesignale erzeugt) und ein Adreßdecoder-PAL20L8 (das die Freigabesignale für alle memory mapped Systemkomponenten erzeugt) untergebracht.

Die Ausgänge des Latch 74LS273 haben folgende System-Steuerfunktionen:

Q0 = 0 : 0000H-2FFFH ist beim Lesen ROM und beim Schreiben RAM
= 1 : 0000H-2FFFH ist beim Lesen und Schreiben RAM

Q1 = 0 : 37E0H-37EFH ist Floppy, Drucker, INT-Status, Drive-Select
= 1 : 37E0H-37EFH ist RAM; Floppy u.s.w. sind über die Ports E0H- EFH erreichbar

Q2 = 0 : 3800H-38FFH ist beim Lesen die Tastatur und beim Schreiben RAM
= 1 : 3800H-38FFH ist beim Lesen und Schreiben RAM

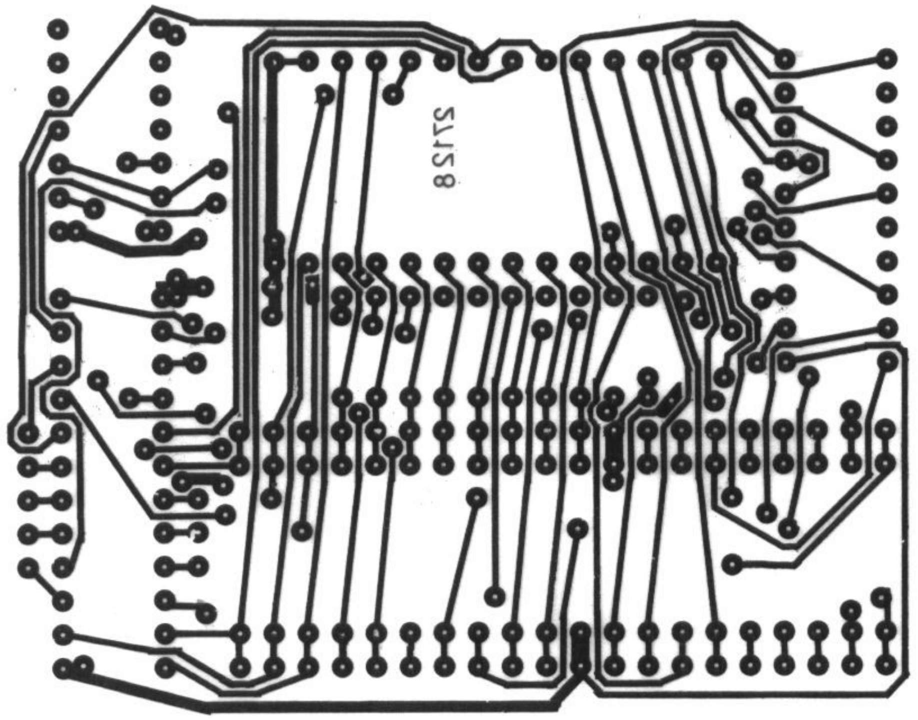
Q3 = 0 : 3C00H-3FFFH ist Video-RAM
= 1 : 3C00H-3FFFH ist RAM
Q4 = 0 : die memory mapped Systemkomponenten liegen im Bereich 0000H-3FFFH
= 1 : die memory mapped Systemkomponenten liegen im Bereich C000H-FFFFH

Unabhängig vom Zustand von Q0-Q4 ist dauerhaft bei 3000H-37DFH, 37F0H- 37FFH und 3900H-3BFFH RAM eingeblendet. Anstelle eines Sonder-ROM war mir RAM bei 3000H wichtiger.

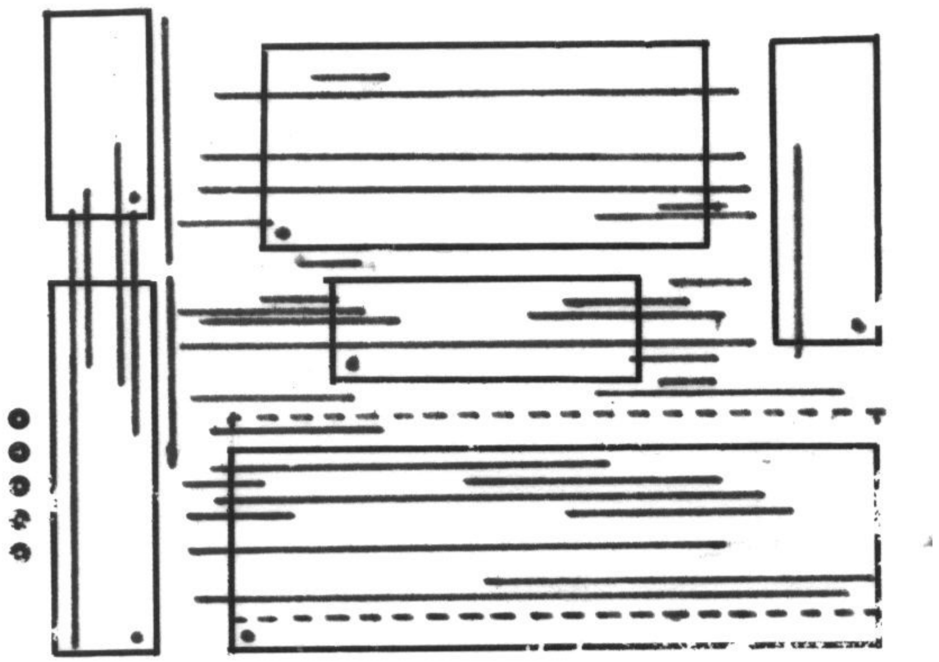
Das Board erzeugt die Freigabe-Signale /ROM, /RAM, /FLO, /KB und /VID und das Steuersignal /MRD für die Lesetreiber des RAMs. Diese Signale sind an die entsprechenden Punkte auf dem CPU-Board zu führen; die ursprünglichen Signale sind zu durchtrennen.

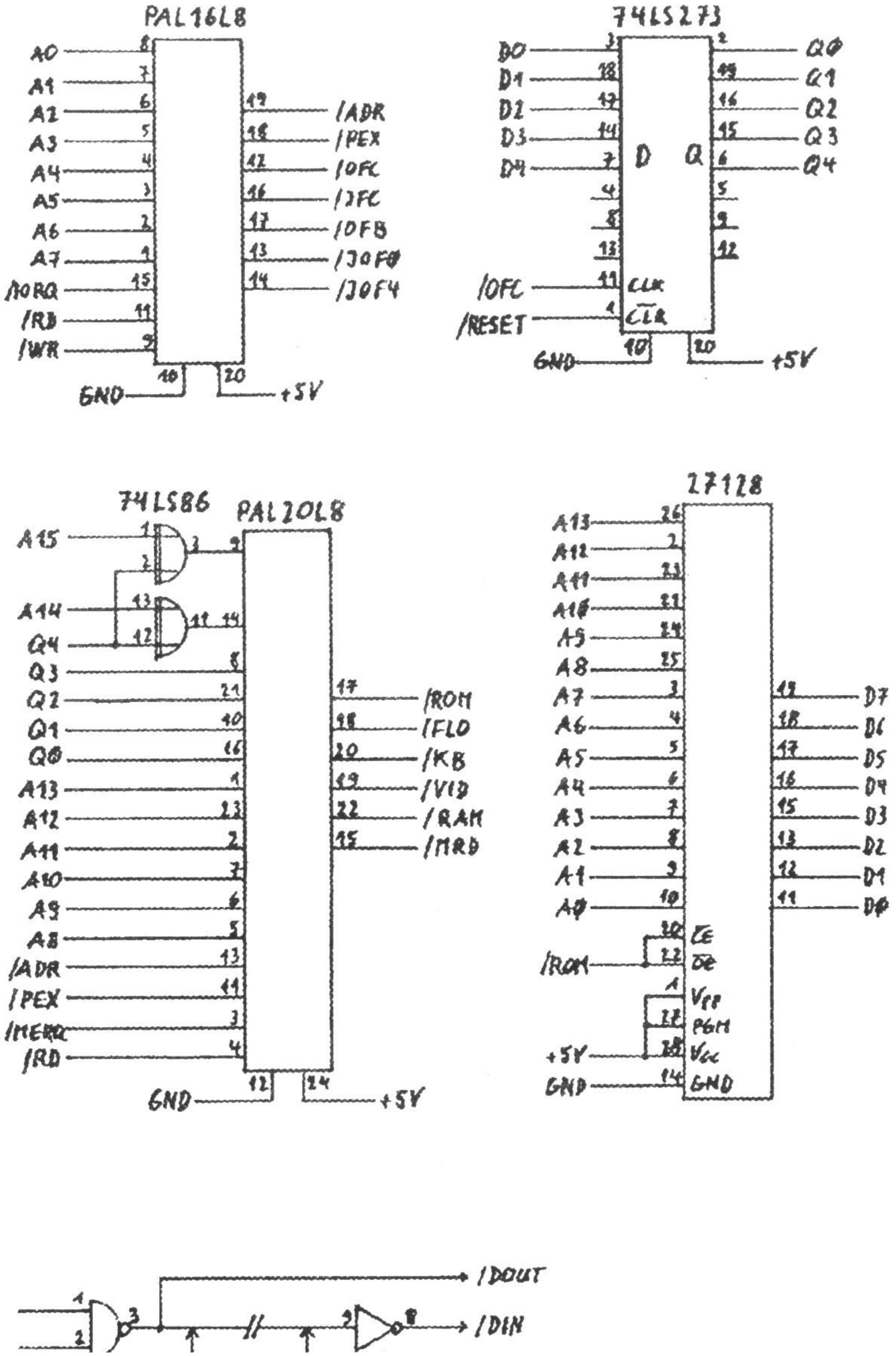
Layout

Schaltplan



Drahtbrücken




PAL16L8
A0
A1
A2
A3
A4
A5
A6
A7
/IORQ
/RD
/WR
/ADR
/PEX
/OFC
/IFC
/OFB
/IOF0
/IOF4
GND
+5V
74LS273
D0
D1
D2
D3
D4
Q0
Q1
Q2
Q3
Q4
D
Q
CLK
CLR
/OFC
/RESET
GND
+5V
74LS86
PAL20L8
A15
A14
Q4
Q3
Q2
Q1
Q0
A13
A12
A11
A10
A9
A8
/ADR
/PEX
/MERQ
/RD
/ROM
/FLO
/KB
/VID
/RAM
/MRD
GND
+5V
27128
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
D7
D6
D5
D4
D3
D2
D1
D0
CE
OE
Vpp
PGM
Vcc
GND
/ROM
+5V
GND
/DOUT
/DIN

Frei zum CPU-Board und zum RB-EXP1 zu verdrahtende Signale

Signal	Pin des 20L8	zu verbinden mit	zu durchtrennende Leitung	
/RAM	22	Z37,74LS367,Pin15	von Z36,74LS04,Pin2	#1
/MRD	17	Z9,74LS367,Pin1	von Z21,74LS20,Pin8	#2
/KB	20	Z7,74LS368,Pin1	von Z35,74LS32,Pin3	
/VID	19	Interboard-Conn. Pin10	von Z37,74LS32,Pin11	
/FLO	18	RB-EXP1, Z18, 74LS15,Pins2,14	RB-EXP1, Z14, 74LS32,Pin11	#3

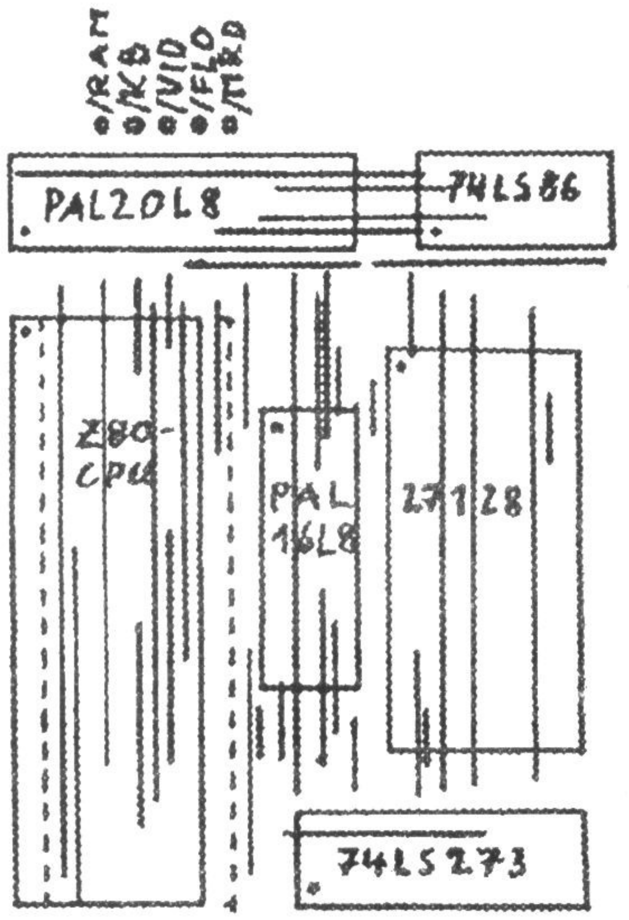
#1 bei Aufrüstung auf 64K (256K) RAM besteht diese Verbindung als freie Verdrahtung; dieser Draht wird entfernt

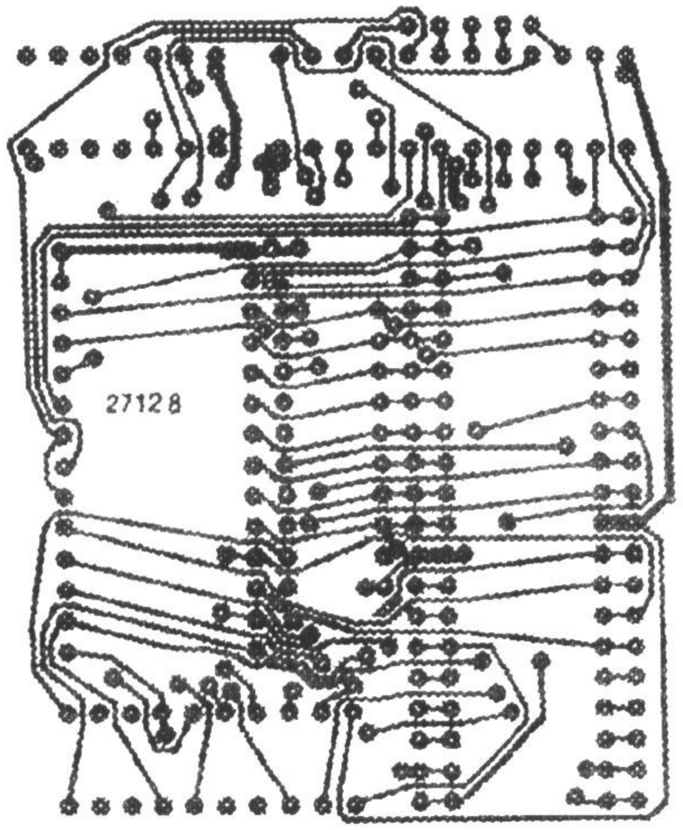
#2 bei Entfernen von Z21,74LS20 entfällt das Durchtrennen

#3 Z13,74LS30 auf dem EXP1 kann entfernt werden anstelle von /MWR von Z15,74LS32,Pin3 auf dem CPU-Board muß /BWR von Z16,74LS367,Pin11 an das EXP1 geführt werden, damit der Betrieb über die Ports E0H-EFH möglich ist

Das Freigabesignal /ROM ist auf dem Huckepack-Board direkt an die Pins 20 und 22 des EPROMs geführt.

Die Erzeugung dieser Freigabesignale ist aus dem Assembler-Listing für das PAL20L8 zu ersehen. Die Adreßeingänge A14 und A15 dieses PALs sind nicht die Adressen A14 und A15 der CPU. Diese Signale werden aus 2 XOR-Gattern (74LS86) aus den Adressen A14 und A15 der CPU erzeugt. Mit Q4 des Latch läßt sich damit steuern, ob A14 und A15 invertiert oder direkt an das PAL gelangen.





PAL-Assembler-Listing für das Adreßdecoder PAL20L8

PAL20L8 GENIE 1 Adreßdecoder mit mm-I/O-Switching
04.09.88 H. Bernhardt

A13 A11 MERQ RD A8 A9 A10 Q3 A15 Q1 PEX GND
ADR A14 MRD Q0 ROM FLO VID KB Q2 RAM A12 VCC

/ROM = /Q0 */MERQ */RD */A15 */A14 */A13
+ /Q0 */MERQ */RD */A15 */A14 * A13 */A12
/FLO = /Q1 */MERQ */A15 */A14 *A13 *A12 */A11 *A10 *A9 *A8
*/ADR + Q1 */PEX
/KB = /Q2 */MERQ */RD */A15 */A14 *A13 *A12 *A11 */A10 */A9 */A8
/VID = /Q3 */MERQ */A15 */A14 *A13 *A12 *A11 *A10
/RAM = /MERQ *ROM *FLO *KB *VID
/MRD = /MERQ */RD *ROM *FLO *KB *VID

Um bei den zur Verfügung stehenden Pins des PAL20L8 auch das Signal /FLO voll ausdecodieren zu können (A4-A15), mußte eine Gruppe von Adressen (A4-A7) schon zusammengefaßt (/ADR) und an einen Pin des PALs geführt werden. Auch für das alternativ von einer Portadresse herzuleitende /FLO war ein vorheriges Ausdecodieren von /IORQ, A4-A7 nötig (/PEX). Diese Signale werden vom PAL16L8 erzeugt, an das für das Erzeugen von

/OUTFC (Freigabe des Steuer-Latch) ohnehin schon die Adressen A0-A7 und /IORQ gelangen.

PAL-Assembler-Listing für das I/O-Decoder-PAL16L8

PAL16L8
GENIE 1 Portdecoder
04.09.88 H.Bernhardt
A7 A6 A5 A4 A3 A2 A1 A0 WR GND
RD OFC IOF0 IOF4 IORQ IFC OFB PEX ADR VCC

/OFC	= /IORQ */WR *A7 *A6 *A5 *A4 *A3 *A2 */A1 */A0	;Pin12
/IFC	= /IORQ */RD *A7 *A6 *A5 *A4 *A3 *A2 */A1 */A0	;Pin16
/OFB	= /IORQ */WR *A7 *A6 *A5 *A4 *A3 */A2 *A1 *A0	;Pin17
/IOF0	= /IORQ *A7 *A6 *A5 *A4 */A3 */A2	;Pin13
/IOF4	= /IORQ *A7 *A6 *A5 *A4 */A3 *A2	;Pin14
/PEX	= /IORQ *A7 *A6 *A5 */A4	;Pin18
/ADR	= A7 *A6 *A5 */A4	;Pin19

Da das PAL16L8 noch einige freie Ausgangs-Pins hatte, wurden noch einige weitere Port-Freigabe-Signale decodiert.

Funktion der Port-Freigabesignale des PAL16L8

/OFC (/OUT FCH)	wird intern zur Freigabe des Sytem-Steuerlatch benutzt
/IFC (/IN FCH)	steht für die Freigabe eines über Port FCH zu lesenden Treibers zur Verfügung
/OFB (/OUT FBH)	Freigabe-Signal, um über Port FBH ein Latch zu be schreiben; kann für die Freigabe des 74LS273 (Pin11) auf dem 256K-Banker benutzt werden (A0-A5 brauchen dann nicht mehr zum Banker geführt zu werden, A6 und A7 sind für das Erzeugen einer 8Bit-Refreshadresse dort aber weiterhin nötig; das 74LS30 auf dem Banker kann entfallen)
/IOF0 und /IOF4; (/In/OUT F0H-F3H; /IN/OUT F4H-F7H)	Freigabe-Signale für z.B. Z80-Peripherie-Bausteine mit 4-Portadressen (PIO, SIO, CTC, DMAC, PPI8255...)

Die Verwendung eines 27128-EPROMs (wovon es auch 150ns-Versionen gibt) erlaubt bedeutend höhere Taktraten als die ursprünglichen ROMs. Wenn dann auch noch das Timing der Signale /RAS, /MUX und /CAS anders gelöst wird, spielen auch die RAMs mit. 5,3MHz sind auch ohne große Eingriffe möglich.

Da das ROM nun innerhalb der Treiber der CPU-Signale liegt, muß zur Steuerung der Daten-Lesetreiber auch das Signal /ROM herangezogen werden. Diese Treiber dürfen nicht freigegeben werden, wenn das ROM gelesen wird.

Um dies zu erreichen, wird Pin9 von IC1,74LS14 durchgekniffen und hochgebogen. Auf IC1 wird ein 74LS08 mit den Pins 7 und 14 huckepack aufgelötet. Alle anderen Pins des 74LS08 werden hochgebogen. Der abgekniffenen und hochgebogene Pin9 des IC1 wird mit dem seitlich gebogenen Pin8 des 74LS08 verlötet. Pin10 des 74LS08 wird mit Pin22 des 27128-EPROMs und Pin9 des 74LS08 mit Pin3 von IC14,74LS00 verbunden (Abbildung).

Zur Freigabe der Speicherlese-Treiber erzeugt das PAL20L8 das Signal /MRD, das an Pin15 von IC9,74LS367 zu legen ist. Durch Entfernen von IC21,74LS20 wird das bisherige Signal entfernt.

Außer IC21 sind auch IC22,74LS156 und IC25,74LS139 überflüssig geworden. Diese ICs können ebenfalls entfernt werden. Und natürlich können auch die alten ROMs rausgeschmissen werden.

Beim Aufbau des Huckepack-Boards sind folgende Punkte zu beachten:

Vor dem Einlöten irgendwelcher Bauteile sind zunächst sämtliche Drahtbrücken zu legen. Als erstes sind dann der Sockel für den Z80 und der Stecker für das Anflanschen des Boards in den Z80-Sockel auf dem CPU-Board in geeigneter Reihenfolge einzulöten. Der Stecker wird auf der Lötseite und der Sockel auf der Bestückungsseite eingebaut. Dabei erschwert das zuerst eingebaute Teil den Einbau des anderen. Was zuerst angelötet wird, sollte eine gründliche Besichtigung der Teile ergeben.

Es sollten alle ICs gesockelt werden. Sowohl beim EPROM als auch bei den PALs kann irgendwann der Wunsch nach geändertem Inhalt aufkommen. Und für Testzwecke bei der Fehlersuche ist es auch bei Standard-ICs zweckmäßig, wenn man diese auswechseln kann oder einfach nur einzelne Pins aus der Fassung biegen kann.

Ungebohrte Boards, mit Level-II befüllte 27128-EPROMs und gebrannte PALs kann ich zum Selbstkostenpreis zur Verfügung stellen.

1989-Club80-10 Nr.28 S.78-99

Es lebe der Standard!

Hercules Graphic Adapter meets Trash 80

Helmut Bernhardt

Genau besehen hat es auch Vorteile, daß der Industriestandard einen so niedrigen Standard hat. Insbesondere die Hercules-Karte, die dort schon nicht unbedingt als schlechteste Video-Karte gilt, ist schon fast zum TRS80 kompatibel. Ja, nicht ganz zu dem, aber doch schon zum GenieIIIs-Video-Interface. Wie im GIIIs gibt es dort eine 2K große Text-Video-Seite (80x25 Zeichen) und ebenso memory mapped. Zusätzlich bietet die Hercules aber auch noch einen 2K großen Attribut-Speicher, womit sich für jedes Zeichen auf dem Bildschirm einzeln Attribute setzen lassen. Und schließlich wird auf der Hercules wie im GenieIIIs mit 2 HRG-Pages mit je 32K Größe (ebenfalls bei beiden memory mapped) Grafik dargestellt. Bei beiden Geräten wird alles von einem CRTC6845 gesteuert.

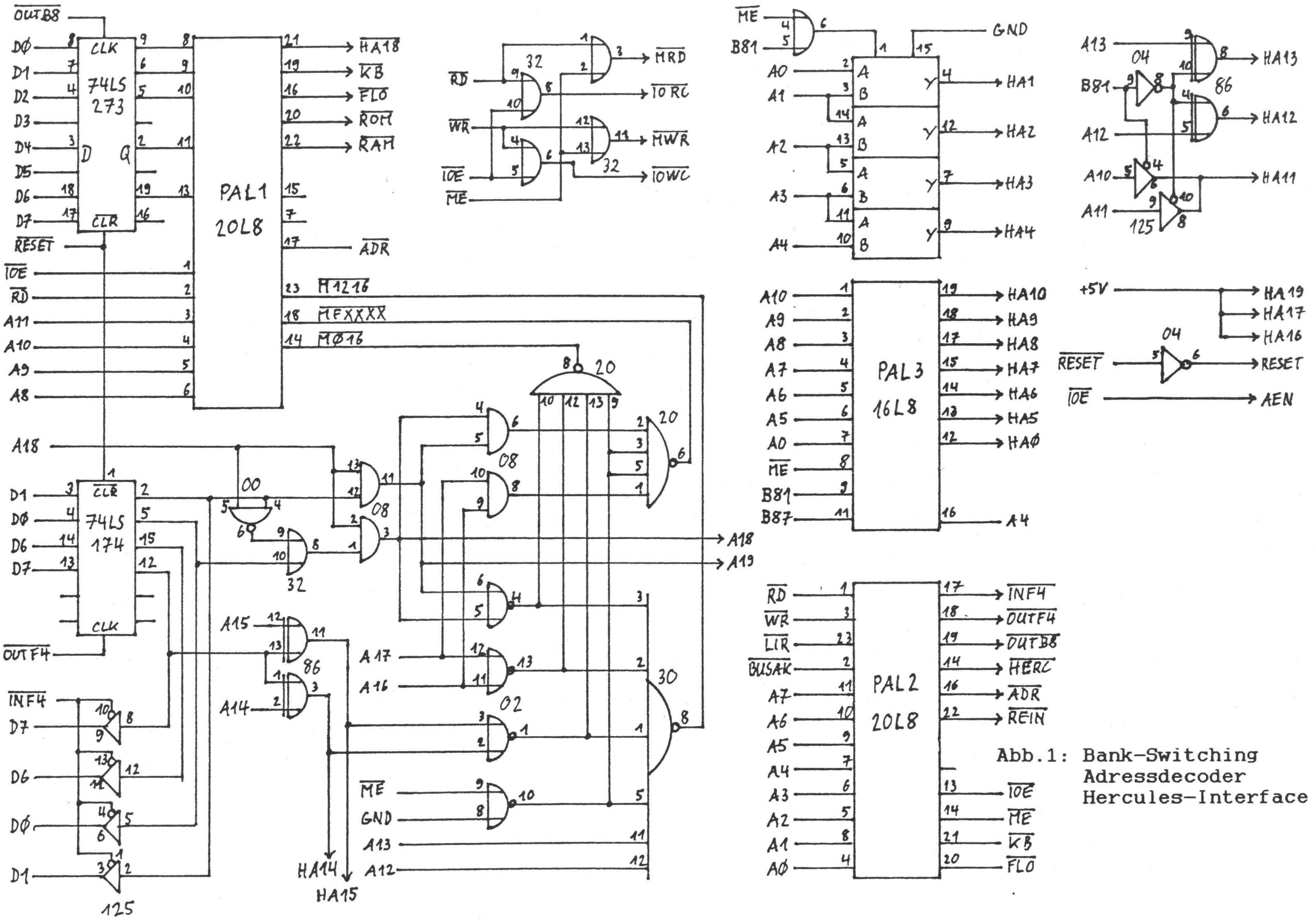
Die maßgeblichste Inkompatibilität ist die Tatsache, daß bei der Hercules Text- und Attribut-Seicher ineinander verschoben sind. Dort werden innerhalb eines lineren 4K-Bereichs alle geradzahligen Adressen für Textzeichen und alle ungeradzahligen Adressen für die dazugehörigen Attribute verwendet. Weniger störend sind die 10Bit-Portadressen des Industrie-Standards, mit denen die internen Register der Karte angesprochen werden. A8 und A9 kann man der Karte bei Z80-Portzugriffen einfach vorgaukeln.

Folgende Tabelle stellt die Ähnlichkeiten und Abweichungen zwischen Hercules und GenieIIIs-Video-Interface zusammen:

Funktion	**Hercules**	**GenieIIIs**
Text-Speicher	B0000-B0FFF (mit A=0)	3800(3C00)-3FFF
Attribut-Speicher	B0000-B0FFF (mit A=1	nicht vorhanden
HRG-Seite 0	B0000-B7FFF	8000-FFFF
HRG-Seite 1	B0000-BBFFF	8000-FFFF
CRTC6845-Ports	3B4H/3B5H	F6H/F7H

Diese Ähnlichkeit ließ mich vor längerer Zeit schon mal darüber brüten, wie man die Hercules anstelle des dort recht lausigen Video-Interface in TRS80 und Genie-Rechnern einsetzen könnte. Zunächst dachte ich an einen Adapter, der mir die nötige Adreßübersetzung macht und die Verbindung zwischen Genie-CPU-Board und Hercules herstellt. Dann dachte ich daran, daß sich dieser Aufwand für das ebenfalls recht lausige Genie-CPU-Board nicht lohnt. Inzwichen ist um diesen reinen Adapter ein ganzes CPU-Baord gewachsen, das noch einiges mehr bietet als das Genie-CPU-Board und dabei auf die Maße einer Europa-Karte geschrumpft ist. Das Board enthält all das, was mir bislang lieb geworden ist:

[1] H64180-CPU
[2] 2 serielle Schnittstellen
[3] 1MB RAM bestehend aus 8 mal 511000-Megachips
[4] auf 1MB erweiterte MMU des HD64180
[5] 27128-EPROM mit Level II BASIC und Initialisierungen
[6] ECB-Bus mit Buspufferung und IM2- und DMA-fähiger Steuerung
[7] Interface zur Hercules
[8] komfortables Switching für memory mapped I/O



Bank-Switching, Adressdecoder, Hercules-Interface)

Damit steht mir ein voll Genie1-kompatibler (nur Spiele lassen sich ohne Bauklötzchengrafik nicht so richtig spielen) Computer zur Verfügung. Die Hercules stellt mir dafür den Standard-64x16-Bildchirm, eine 80-Zeichen-Karte und zwei hochauflösende Grafikseiten mit je 720x348 Punkten zur Verfügung, und das alles für 10,-DM.

Die eigentlichen Kosten fallen bei der Beschaffung der RAMs an. Zugegeben - 256K hätten's beim Genie auch getan, aber das Teil-Layout mit CPU, RAMs, EPROM, RAM-Steuerung und Treibern für die V24-Schnittstellen brauchte ich nur aus der Schublade zu ziehen und beim Rest habe ich mir dann auch keine große Mühe mehr gemacht. So ist

Der Aufbau der Karte

zu einem richtigen kleinen Abenteuer geworden. Das Layout beschränkt sich auf einseitge Leiterbahnführung und versucht, dort so viele Verbindungen unterzubringen, wie irgend möglich. Nur im hinteren Teil der Karte (der aus der Schublade), war es möglich, alles mit Drahtbrücken zu erledigen - bis auf das EPROM, das muß auch von Hand verdrahtet werden, dort sind nur D0-7, +5V und GND angeschlossen. Was von Hand zu verdrahten ist, kann aus der Verdrahtungsliste ersehen werden.

Wer nun immer noch glaubt, daß er dieses Projekt durchstehen kann, und von mir zum Selbstkostenpreis ein geätztes, ungebohrtes Board bekommen hat, muß zunächst mal 1-2 Stunden mit der Bohrmaschine zubringen. Die 3 Stecker für VG64-Leiste, für die Pfostenfeld-Leiste zum Aufstecken der Hercules-Karte und die Pfostenfeld-Leiste für die seriellen Schnittstellen müssen 1mm-Bohrungen erhalten, alle anderen Bohrungen sind in 0,8mm niederzubringen.

Wenn man sich davon wieder erholt hat, legt man sämtliche Drahtbrücken (Drahtbrücken- und Bestückunpsplan) und fängt dann erst an, Kleinteile und IC-Sockel einzulöten. Da die Drahtbrücken teilweise recht eng liegen, sollte grundsätzlich dünner Kupfer-Lackdraht verwendet werden. Bei Drahtbrücken, die +5V und GND führen sollen (solche Brücken enden meistens an etwas breiteren Leiterbahnen), ist etwas dickerer isolierter Schaltdraht vorzuziehen.

Es sei noch gesagt, daß im dichter besiedelten vorderen Bereich (beim Busstecker) in der ersten Reihe ein IC nicht zu bestücken ist. Weil der Platz halt da war, habe ich da vorsichthalber mal die Lötaugen hinterlegt, damit es spätere Bastler bei Erweiterungen einfacher haben.

Auf der Busplatine oder frei improvisiert auf dem CPU-Board müssen die CPU-Signale /BUSRQ, /RESET, /WAIT, /MMI und /INT0 mit 10k-Ohm an +5V gelegt werden. /INT1 und /INT2 müssen direkt am HD64180 über 10k an +5V gelegt werden. Wenn diese Eingänge nicht genutzt werden, genügt ein gemeinsamer Widerstand.

Die 62polige Steckleiste zum Anschluß der Hercules ist für eine spezielle, besonders kleine Karte gedacht, die direkt neben dem für PC-Karten üblichen Platinenrand-Stecker zwei Reihen Durchkontaktierungen aufweist, die freundlicherweise im RM2,54 Rastermaß angelegt sind und eine Weite von 1 mm haben. Wenn hier von der Bestückungseite eine zweireihige Buchsenleiste eingelötet wird, und auf dem CPU-Board die Pfostenfeld-Leiste mit etwa um 1cm verlängerten Beinchen eingelötet wird, können beide Karten direkt aufeinander gesteckt werden.

Der Fernseher darf leider nicht der alte vom Genie1 bleiben. Die Hercules benötigt einen TTL-Monitor (ohne Gehäuse und Netzteil bekommt man sowas im Elektronik-Ramsch-Versandhandel für ca. 40,-DM). Andererseits gibt es auch TTL/BAS-Wandler, denen man aber ein aus "Video" und "Intensity" zusammengeführtes Signal anstelle des reinen Video-Signals zuführen sollte, um nicht auf die Helligkeitssteuerung der Hercules verzichten zu müssen.

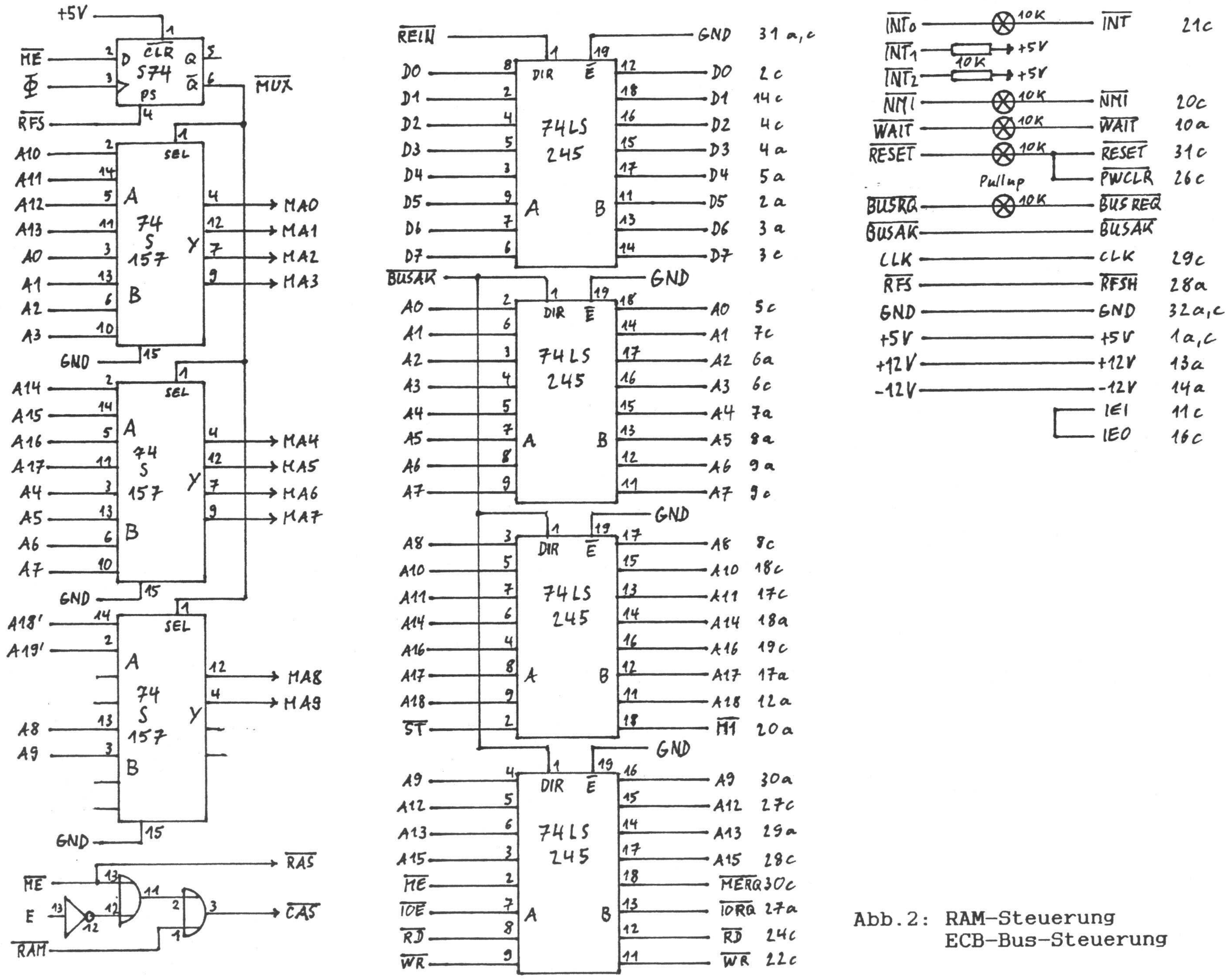


Abb.2: RAM-Steuerung
ECB-Bus-Steuerung

RAM-Steuerung, ECB-Bus-Steuerung

Programmierung der Hercules-Karte

Interne Ports (die Portdressen sind die auf die 8Bit-Decodierung durch das Interface reduzierten Adressen; im PC müssen zusätzlich A8 and A9 high sein)

B4H Adressregister des CRTC6845
B5H Datenregister des CRTC6845
B8H Display-Betriebsart-Steuerkanal
B9H Display-Status-Kanal
BBH Light-Pen Flip Flop zurücksetzen
BCH Drucker-Datenort
BDH Drucker-Statusport
BEH Drucker-Steuerkanal
BFH Konfigurations-Schalter

Für die Programmierung des CRTC6845 ist ein Datenblatt erforderlich. Durch Trial-and-Error Methoden soll man es schaffen können, den Monitor dauerhaft zu disablen. Es sei nur soviel gesagt, daß der CRTC eine ganze Menge Datenregister besitzt, wovon nur die ersten 16 relevant sind. Sie haben alle die Adresse B5H. Auf welches dieser Register jeweils über Port B5H zugegriffen werden kann, bestimmt der Eintrag in das Adressregister B4H. Die Initialisierungsroutine im ROM enthält an der Adresse 3000H 5 Tabellen mit je 16 Daten für die ersten 16 Datenregister des CRTC und eine Routine, die dem CRTC diese Daten einverleibt. Wenn HL auf den Anfang der jeweiligen Tabelle zeigt. Dort mag sich der Softi informieren, der lieber noch exotischere Bildschirm-Formate einstellen möchte. Die grundsätzliche Bedeutung der CRTC-Register ist aus mc 1/88 S.113 zu entnehmen.

Der Display-Betriebsart-Steuerkanal (Port B8H) der Hercules reagiert nur auf die Bits 1, 3, 5 und 7. Die geradzahligen Datenbits haben dort keine Funktion. Da die Einstellungen dieses Registers für die Adreßübersetzung des Interface zur Hercules von Bedeutung sind, wurde dieser Port auf dem CPU-Board nachgebildet und um die geradzahligen Bits bereichert. Setzen und Zurücksetzen einzelner Bits dieses Ports durch Auslesen, Bit verändern und Zurückschreiben ist also nicht möglich. Man muß dafür im RAM über den Zustand des Ports Buch führen.

In der folgenden Tabelle sind die Funktionen der einzelnen Bits des erweiterten Ports B8H zusammengestellt. Die urgsprünglichen Funktionen des Ports auf der Hercules, die nach wie vor gelten, sind mit # markiert.

Display-Mode-Control-Port, B8H

D0 =0: 1K Video-RAM bei 03C00-03FFF (64x16), Keyboard bei 3800-38FF
=1: 2K Video-RAM bei 03800-03FFF (80x25), Keyboard disabled

D1 =0: Text-Video-Mode enabled, HRG disabled #
=1: HRG enabled, Text-Video disabled #

D2 =0: ROM bei 00000-036FF (nur bei /RD, bei /WR RAM)
=1: ROM disabled, bei 00000-036FF liegt RAM

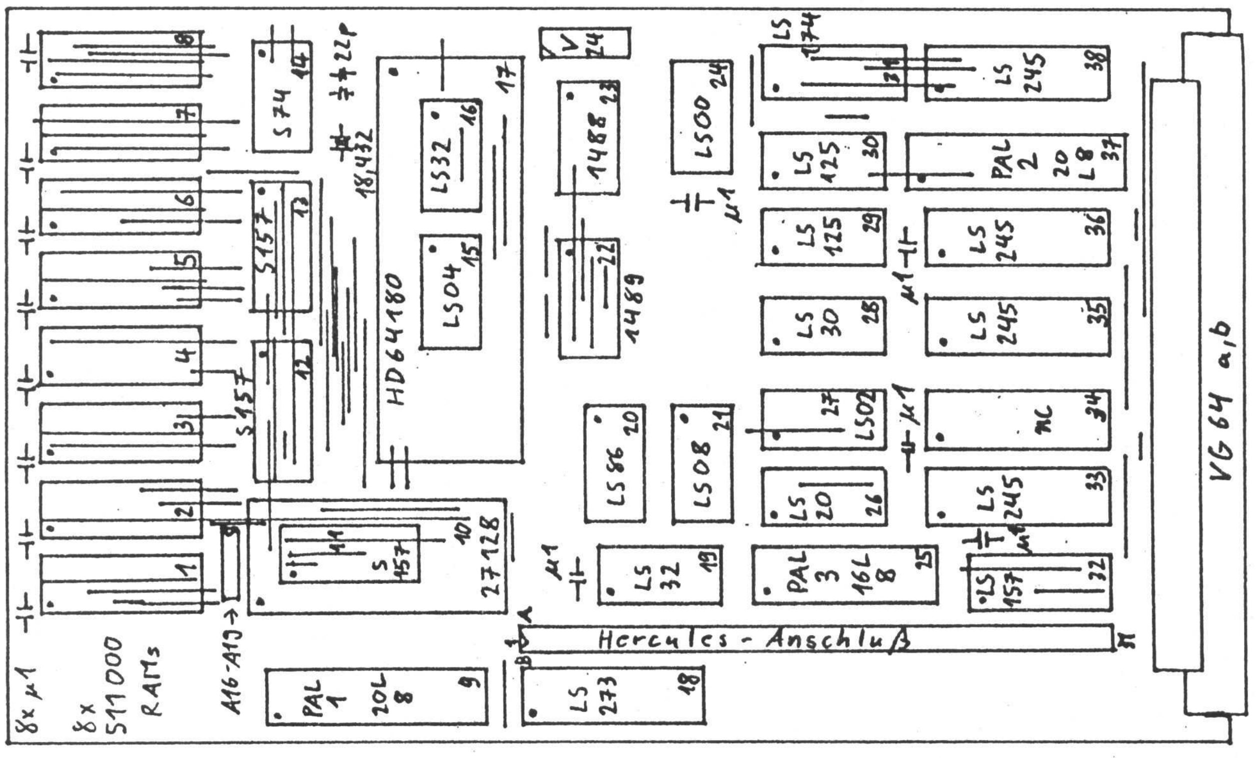
D3 =0: die Hercules gibt kein Video-Signal an den Monitor #
=1: das Video-Signal der Karte gelangt zum Monitor #

D4 =0: Floppy, Drucker u.s.w. bei 037EX memory mapped
=1: 037EX ist RAM, Floppy u.s.w. über Ports EXH

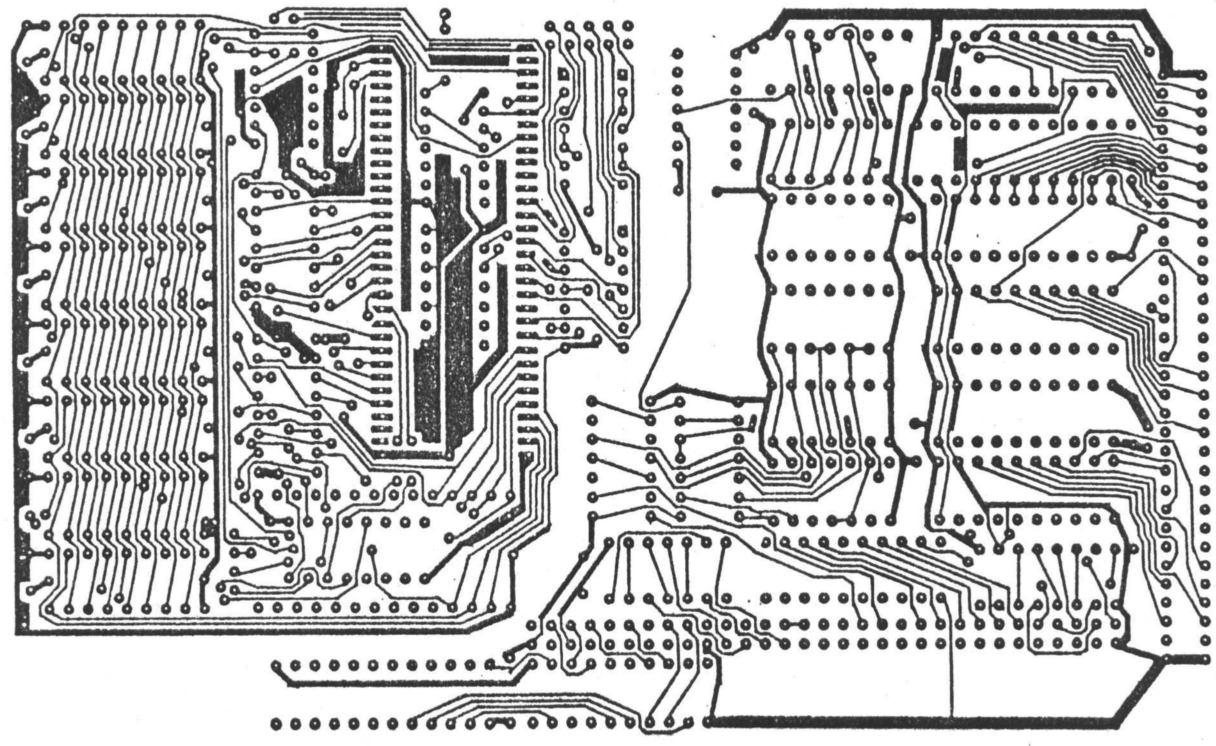
D5 =0: D7=1 des Attribut-Bytes bedeutet "heller Hintergrund" #
=1: D7=1 des Attribut-Bytes bedeutet "Blinken" #

D6 =0: Text- bzw HRG-RAM sind freigegeben
=1: Text- bzw HRG-RAM gesperrt, dort RAM eingblendet

D7 =0: HRG: Page 0 selected #
Text: 2K Textspeicher eingestellt
=1: HRG: Page 1 selected #
Text: 2K Attributspeicher eingstellt



Bestückungs- und Drahtbrücken-Plan



Teil-Layout (von der Bestückungsseite gesehen)

Die Einteilung des 1MB-Speichers

Der HD64180 (in der hier verwendeten DIL64-Form) kann nur 512K Speicher adressieren. Um den vollen 1MB-Speicher zu nutzen, muß durch zwei weitere Bits eines Ports (D0 und D1 von Port F4H) ein Banking von 256-K-Blocks dem Banking der MMU des HD64180 überlagert werden. Die Selektion eines der 4 256K-Blocks erfolgt mit A8 des HD64180 und D0/D1 von Port F4H. Aus diesen 3 Signalen werden die Adressen A18 und A19 für das RAM hergeleitet.

Port F4H		CPU	RAM		256K-
D1	**DO**	**A18**	**A19**	**A18**	**-Block**
X	X	0	0	0	0
0	0	1	0	1	1
0	1	1	0	1	1
1	0	1	1	0	2
1	1	1	1	0	3

Wenn A18 der CPU low ist, wird immer der unterste 256K-Block angewählt. Wenn A18 high ist, wird einer der 3 anderen Blocks selektiert. Welcher djeser 3 Blocks selektiert wird, hängt von den Zuständen von D0 und D1 des Ports F4H ab. Diese Vorgaben gelten sowohl für die MMU als auch für den DMA-Controller.

Nach einem RESET ist die MMU des HD64180 immer so eingetellt, daß der logische 64K-Z80-Adreßraum in den unteren 64K des unteren 256K-Blocks liegt. Um die Standard-Software (Newdos/80, CP/M) zu fahren, braucht die MMU nicht umprogrammiert zu werden. Die Decodierung von RAM und memory mapped I/O schließt auch die Adressen A16-A19 mit ein, so daß diese Baugruppen im Physikalischen Adressraum 00000H-03FFFH liegen. Nur der HRG-Speicher der Hercules-Karte mit 2mal 32K wäre bei 08000H eingeblendet etwas lästig und wurde deshalb an das obere Speicherende bei 0F0000H verbannt. Für die MMU und den DMAC des HD64180 sind dies die Adressen 70000H-7FFFFH, wobei D0 und D1 von Port F4H gesetzt sein müssen, damit der HRG-Speicher adressiert wird.

Neben der Grundeinstellung (D7 an Port F4H low, Zustand nach RESET) gibt es noch eine andere Lage für die memory mapped Baugruppen. Wenn D7=1 an Port F4H ausgegeben wird, liegen ROM, Floppy, Drucker, Tastatur und Text-Video-RAM bei 0C000H-0FFFFH und die untersten 16K sind RAM.

Und schließlich läßt sich innerhalb dieser 16K (wo auch immer die memory mapped Baugruppen eingeblendet sind) sehr komfortabel umkonfigurieren, darüber gibt die Tabelle der Bedeutung der Bits des erweiterten Ports B8H und die Map der Speipcheraufteilung Auskunft. In allen Bereichen, wo die jeweilige memory mapped Baugruppe ausgeblendet ist, liegt dann automatisch RAM vor.

Der Text-Bildwiederholspeicher

Die Hercules-Karte stellt im Text-Modus einen 2K großen Bildwiederholspeicher für Text und einen 2K großen Attribut-Speicher zur Verfügung. Diese Speicher sind ineinander verschoben; innerhalb eines linearen 4K-Bereichs stellen die geraden Adressen den Text-Speicher und die ungeraden Adressen den Attribut-Speicher dar.

Video-Attribute waren aber damals, als der TRS-80 geboren wurde, noch unbezahlbarer Luxus. Deshalb scheint der Attribut-Speicher der Hercules der Kompatibilität zum TRS-80 im Wege zu stehen. Der TRS-80 braucht einen zusammenhängenden Bildwiederhol-Speicher von 1K Größe (und manche jüngeren Brüder können auch mit 2K Textspeicher umgehen).

Dieses unlösbare Problem läßt sich dadurch lösen, daß beim Text-Modus die Adressen A0-A10 der CPU (2K-Bereich) auf die Adressen A1-A11 der Hercules-Karte gelegt werden. Diese Umsetzung darf aber nur bei Zugriff auf den Text- und den Attribut-Speicher erfolgen. Bei HRG-Betrieb und bei Zugriff auf die Ports der Karte müssen alle Adressen direkt durchgeschaltet werden. Die für das Selektieren von Text- oder Attribut-Speicher zuständige Adresse A0 (zur Hercules) wird im Text-Modus von D7 des erweiterten Ports B8H und bei Zugriff auf Hercules-Ports und HRG-Speicher von A0 der CPU bedient. Diese Adressteuerung wird von PAL3 und einem 74LS157-Multiplexer erledigt.

Die damit auf 2K reinen Text-Speicher (und nach Umschalten 2K reinen Attriut-Speicher) beschränkte Bildschirmseite ist aber für den TRS-80, der nur 1K bei 3C00H dafür vorsieht, immer noch zu groß. Deshalb kann mit D0=0 an Port B8H (Einstellung nach der Initialisierung im ROM) die Freigabe nur der oberen Hälfte des Text-Speichers im Bereich C000-3FFF vorgegeben werden. Für das TRS-80-Format 64x16 Zeichen muß dann aber auch der CRTC-6845 entsprechend programmiert werden. Dies wird ebenfalls durch die Initialisierung im ROM erledigt. Das Betriebssystem findet dann beim anschließenden Booten eine voll TRS-80-kompatible Maschine vor.

Wenn D0=1 an Port B8H ausgegeben wird, ist der volle 2K-Text-Speicher im Bereich 3800-3FFF verfügbar. Um eine vernünftige Darstellung auf dem Bildschirm zu haben, muß dann aber auch der CRTC6845 entsprechend initialisiert werden. Jm 80x25-Modus ist die Tastatur dann aber nicht mehr zugänglich. Für das Aulesen der Tastatur muß dann immer erst in den 64x16-Modus zurückgeschaltet werden und nach dem Auslesen der Tastatur für weitere Bildschirm-Ausgaben wieder der 80x25-Modus eingestellt werden. Wenn dabei die Register des CRTC-6845 in Ruhe gelassen werden und nur mit D0 von Port B8H geschaltet wird, bleibt die Darstellung auf dem Bildschirm unbeeinträchtigt. Die Umschaltung betrifft dann nur die Zugriffmöglichkeiten der CPU auf den Bildwiederholspeicher. Das für die Darstellung des Bildpeichers auf dem Monitor verantwortliche Auslesen des 6845 wird dadurch nicht belangt.

Die Initialisierungs-Routine im ROM füllt zunächst den gesamten Attribut-Speicher mit dem Standard-Wert 7. Das bedeutet, daß die Zeichen auf dunklem Hintergrund mit normaler Helligkeit dargestellt werden, wenn der Attribut-Speicher bei der Textausgabe nicht berücksichtigt wird.

Es sind allerdings dies die Zeichen entsprechend Big Blue's Verständnis von ASCII-Code. Die Zeichen mit den Codes 5BH-5FH und 7BH-7FH sind grundsätzlich die amerikanischen Zeichen (geschweifte und eckige Klammern u.s.w.). Den deutschen Umlauten und auch allen anderen nationalen Sonderzeichen sind feste ASCII-Codes mit gesetztem Bit7 zugeordnet. Das hat den Nachteil, daß bei allen von Standard-Programmen unter MEWDOS und CP/M ausgegebenen Texten anstelle der deutschen Umlaute die entsprechenden amerikanischen Zeichen erscheinen. Ein Vorteil besteht aber auch darin, daß amerikanische Sonderzeichen und sämtliche nationalen Umlaute gleichzeitig dargestellt werden können. Für Soft-Freaks besteht hier ein weites Betätigungsfeld in der Anpassung von Tastatur-Treibern und Standard-Programmen.

Die Bildchirm-Attribute

Wenn mit D1=1 und D7=1 an Port B8h die Text-Betriebsart und der Attributspeicher eingeschaltet werden, kann im Adressbereich des Text-Speichers ein Attribut-Byte eingetragen werden, das die Darstellung des ASCII-Zeichens auf gleicher Adresse im Text-Speicher beeinflußt. Die Attribut-Bits wirken nur auf das eine Zeichen und nicht auf den gesamten Bildschirm. Wenn mit Attributen gearbeitet werden soll, muß für jedes Zeichen im Text-Speicher auch ein Byte im Attribut-Speicher eingetragen werden. Beim Scrollen muß auch der Attribut-Speicher genauo wie der Text-Speicher verschoben werden.

D7:	die Wirkung von D7 eines Attribut-Bytes hängt davon ab, welcher Wert für D5 an Port B8H ausgegeben wurde.		
Port B8H, D5=0:			
	D7(Attribut)	=0:	Zeichen erscheint auf dunklem Hintergrund
		=1:	Zeichen erscheint vor halbhellem Hintergrund
Port B8H, D5=1:			
	D7(Attribut)	=0:	kein blinkendes Zeichen
		=1:	Zeichen wird blinkend dargestellt
	D3	=0:	normale Helligkeit der Darstellung
		=1:	Highlighting

D0-D2, D4-D6

D6	D5	D4	D2	D1	D0	Bedeutung
0	0	0	0	0	0	Zeichen nicht dargestellt
0	0	0	0	0	1	unterstrichen dargestellt
0	0	0	1	1	1	normale Darstellung
1	1	1	0	0	0	invere Darstellung

Die HRG der Hercules-Karte

Mit 720x348 Punkten Auflösung und 2 Seiten läßt sich schon sehr schöne Grafik auf die Mattscheibe bringen. Diese Auflösung benötigt natürlich auch Speicherraum für die beiden 32K-Pages. Diese Bereiche analog zum GenieIIIs einfach in den logischen Adreßraum der nach RESET eingestellten (unteren physikalischen) 64K zu legen, erscheint etwas unzweckmäig zu sein. Es müßte dann eine Hälfte des ohnehin nur 64K großen, direkt erreichbaren Arbeitspeichers ausgeschaltet werden. Für Grafik-Anwendungen müßte man sich festlegen, in welcher Hälfte der 64K die Grafikroutinen laufen sollen.

Hier ist der HRG-Speicher ans obere Ende des physikalischen 1MB-Speichers verbannt worden. Das Besudeln des Grafik-Seichers erfolgt im einfachsten Fall durch blockweise Übertragung per DMA. Wer aber trotzdem gerne direkt in das HRG-RAM schreiben möchte, kann sich dieses durch die in die untere oder obere Hälfte seines 64K-Arbeitsspeichers einblenden.

Nach einem RESET ist die Text-Betriebsart eingstellt. Mit D1=1 an Port B8H läßt sich auf die HRG-Betriebsart "Half-Mode" umstellen, wenn in Port BFH D0=1 eingetragen ist. Dabei ist die untere Hälfte (Page 0) erreichbar. Um auch die Page 1 erreichen zu können, muß auch D1=1 an Port BFH ausgegeben werden.

Während der Grafik-Betriebsart ist der Text-Bildspeicher nicht zugänglich. Da dieser aber physikalicher Bestandteil der Page 0 der HRG ist, wird der Inhalt des Textspeichers bei Grafikausgaben an Page 0 überschrieben. Gleiches gilt für den Attribut-Speicher.

Die Tastatur

Dieses Gerät erleidet hier die größten Einbußen in der Präsenz. So ist der Bereich, in dem die Tastatur ausgelesen werden kann, auf den dafür nötigen Platz von 256 Adressen bechränkt. In der 64x16-Text-Betriebsart (D0=0 an Port B8H, die einzige Einstellung, in der die Tastatur erreichbar ist) stehen bei 3900H-3BFFH ein paar Byte frei verfügbaren Speichers bereit. Auch der Speicher parallel zur Tastatur kann bei eingeblendeter Tastatur beschrieben werden; Auslesen läßt er sich nur, wenn auf Grafik-Betriebsart umgestellt wird.

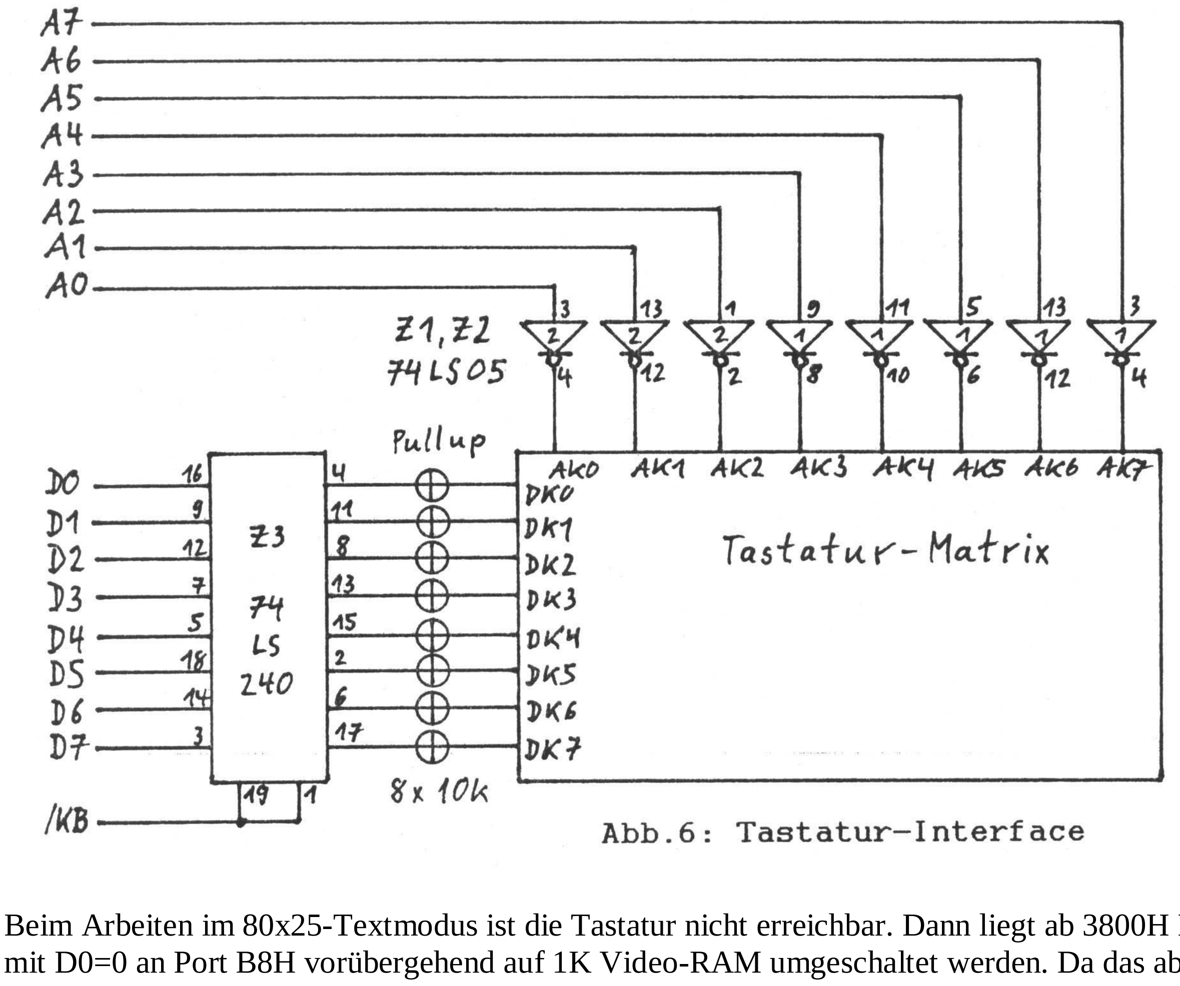


Abb.6: Tastatur-Interface

Beim Arbeiten im 80x25-Textmodus ist die Tastatur nicht erreichbar. Dann liegt ab 3800H Bildwiederhol-Speicher vor. Um die Tastatur zwichendurch auszulesen, muß mit D0=0 an Port B8H vorübergehend auf 1K Video-RAM umgeschaltet werden. Da das aber nicht den Zugriff des CRTC6845 auf den Bildspeicher beeinflußt, zeigt sich das Umschalten nicht in der Darstellung auf dem Bildschirm.

Wenn es auch Wahnsinn ist, eine Tastatur memory-mapped als offene Matrix an einen Computer anzuschließen, weil bei höheren Taktraten und längeren Anschlußleitungen Probleme zu erwarten sind, kommt man bei einem TRS-80-kompatiblen Rechner um diesen Unsinn nicht herum, wenn auch der Betrieb von OLDDOS/80 und der darunter laufenden Programme möglich sein soll.

Ein einfaches Interface (Abb.7) kann auf einem Stück Lochrasterplatine in Fädeltechnik aufgebaut werden und mit einer VG64-Steckerleiste an den ECB-Bus gesteckt werden. Die Tastatur kann daran über ein nicht zu langes, 16poliges Kabel angeschlossen werden.

Die Freigabe des ROMs

Wie es sich für einen Trash-80-kompatiblen Computer gehört, liegen nach dem Einschalt-RESET die 12K Löffel-II-BASIC bei 0000H-2FFFH im ROM rum. Zusätzlich wurde hier auch der Bereich 3000H-36FFH als ROM decodiert. Hier sind die nötigen Initialisierungen der HD64180-CPU und der Hercules-Karte untergebracht. Der Einsprung bei 3050H muß im Level-II-ROM anstelle des ersten Sprungbefehls nach RST 00H gepatcht werden. Die Initialisierung endet mit einem Sprung an das dort vorher vorhandene Sprungziel 0674H.

Wenn D7 an Port F4H high ausgegeben wird, verschiebt sich die Lage des ROM-Inhalt von 0000-36FF nach C000-F6FF. Das ist zwar wenig sinnvoll/ weil der Stoff auf dieser Adresslage nicht läuft, ließ sich aber wegen des für CP/M nötigen Verschiebens des memory mapped I/O nach F7E0-FFFF nicht verhindern. Wenn das ROM aber ausgeschaltet wird (D2 an Port B8H high), stört es dort nicht mehr.

Das Abschalten des ROMs über D2 an Port B8H läßt sich natürlich auch in der ursrünglichen Lage bei 00000H durchführen. Da das ROM ohnehin nur beim Lesen freigegeben wird und beim Schreiben RAM eingeblendet ist (auch wenn das ROM nicht abgeschaltet ist) kann das Kopieren des ROM-Inhaltes in das RAM durch ein einfaches LDIR auf sich selbst (HL = DE = 0) erfolgen. Das wird in der Initialisierung vorsichtshalber auch schon mal gemacht.

Floppy, Drucker und der ganze Rest

Damit die Abwärtskompatibilität zum TRS-80 beibehalten wird, liegen diese Komponenten memory mapped bei 37EXH. Allerdings wurde die Option eingebaut, anstelle dieser Memory-Adressen auch die entprechenden Ports E0H-EFH für das Ansprechen dieser Einheiten benutzen zu können. Dafür wird mit D4=1 an Port B8H auf Betrieb über Ports umgeschaltet.

Wer von seinem alten Genie oder TRS80 her noch einen RB-EXP1 Controller besitzt, hat damit ein weiterhin brauchbares Gerät vorliegen. Diesem Board muß nur noch ein VG64-Stecker spendiert werden. Dieser Stecker wird auf einen 2cm breiten Streifen Lochraster-Platine gelötet, der wiederum mit zwei schmalen Streifen Pertinax oder Sonstigem Kunststoff mit dem RB-EXP1 verschraubt wird (Abb.6). Auf der Lötseite werden dann die ECB-Signale des Steckers mit den entsprechenden Punkten der beiden Anschlußstecker des EXP1 verbunden.

Anschluß des RB-EXP1 Controllers an den ECB-Bus

ECB-Bus Pin	Signal	RB-EXP1	
2c	D0	Pin 9	24poliger
14c	D1	Pin 10	Stecker
4c	D2	Pin 11	
4a	D3	Pin 13	
5a	D4	Pin 14	
2a	D5	Pin 15	
3a	D6	Pin 16	
3c	D7	Pin 17	
5c	A0	Pin 8	
7c	A1	Pin 7	
6a	A2	Pin 6	
6c	A3	Pin 5	
1a	+5V	Pin 24	
32a	GND	Pin 12	
13a	+12V	Pin 11	16poliger
15a	-5V	Pin 1	Stecker
21c	/INT	Pin 4	
31c	/RESET	Pin 5	
z.B. 23a	/FLO	Pins 2,14	74LS155

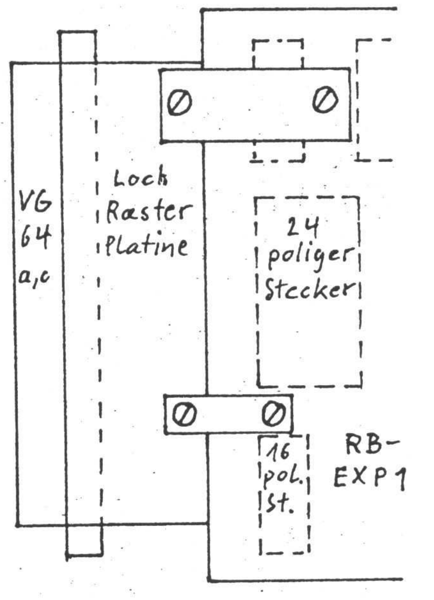


Abb.4: ECB-Bus-Anschluß der RB-EXP1

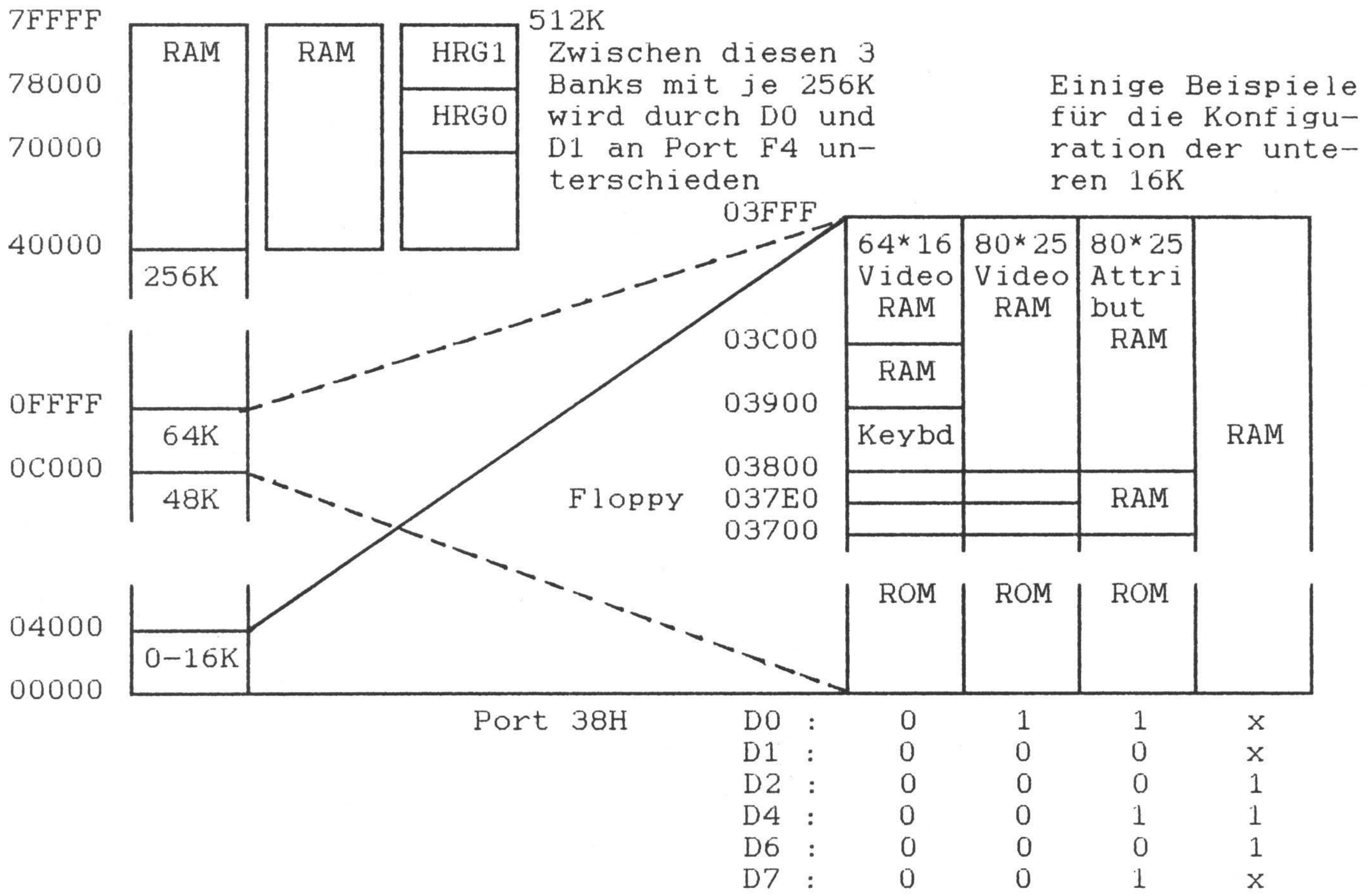
Da das Freigabe-Signal für den Controller bereits auf dem CPU-Board erzeugt wird und auch schon die Umschaltmöglichkeiten beinhaltet, sollte lieber dieses Signal verwendet werden als das auf dem EXP1 hergeleitete Signal /37EX. Dafür wird das Signal /FLO von PAL1 auf dem CPU-Board über eine nicht benutzte ECB-Leitung an den Controller geführt und dort an die Pins 2 und 14 des 74LS155 gelegt. Die beiden 74LS30 beim 16poligen Stecker, die vorher das Signal /37EX und die Drucker-Freigabe über Port FDH geliefert haben, können enternt werden. Die Verwendung des Signals /FLO erübrigt auch die Zuführung der Signale, die in obiger Tabelle nicht angeführt sind und bei Verwendung des EXP1 im Genie oder TR-S80 zusätzlich angeschlossen werden mußten.

RAM-Speicher satt

Grundsätzlich besteht der gesamte physikalische Adreßraum von 1MB aus RAM-Speicher. Nur dort, wo irgendwelche memory mapped I/O-Baugruppen eingeblendet werden, ist der RAM-Speicher nicht erreichbar. Durch Abschalten der mm I/O-Baugruppen wird dort dann automatisch der entsprechende RAM-Bereich eingeblendet.

Im Bereich des ROMs (00000-002FF) und der Tastatur (03800-038FF), wenn diese im 64x16-Textmodus erreichbar ist, kann der RAM-Speicher auch beschrieben werden, ohne daß ROM bzw. Tastatur ausgeschaltet werden müssen. Nur zum Lesen dieser RAM-Bereiche müssen diese Einheiten ausgeblendet werden.

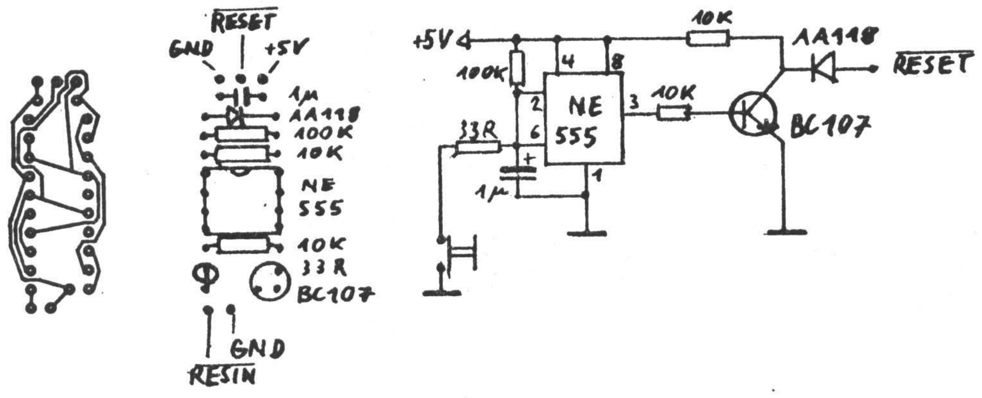
Memory Map



Ein entscheidender Nachteil des Industrie-Standards sei zum Schluß noch erwähnt. Um das vom Genie her bekannte Flickern bei Ausgaben auf den Bildschirm zu verhindern, lassen modernere Hercules-Karten einen Zugriff auf das Video-RAM nur zu, wenn der Elektronenstrahl vom rechten zum linken Bildrand zurückläuft und dabei dunkel gesteuert ist. Die CPU wird während der Zeit des Strahlrücklaufs mit einem Wait (/IO_CH_RDY, Pin A10 des PC-Slots) ruhiggehalten. Offensichtlich ist das Wait-Timing bei HD64180 anders als beim 80286 im PC-AT, denn die Hercules macht im Genie nur bis 5MHz Systemtakt mit; bei höheren Takten erscheint nur noch Müll auf dem Fernseher.

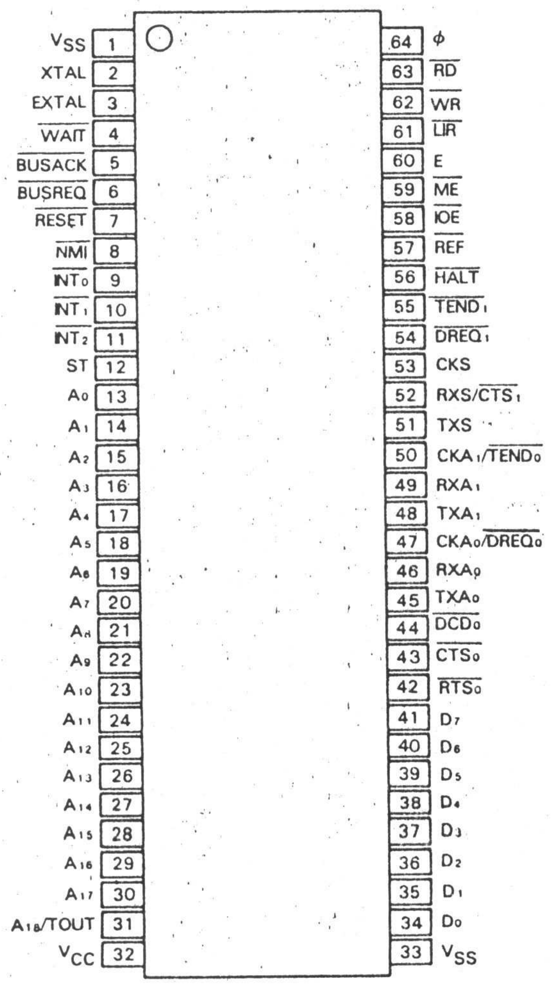
Es gibt aber auch Hercules-Karten, die sich wie ein normales Genie-Video-Interface ohne Rücksicht auf Flickern beschreiben lassen, Eventuell läßt sich damit dann der für das CPU-Board als machbar zu erwartende Takt von 9,216 MHz fahren. Wahrscheinlich muß dann aber ein Wait aus dem Freigabesignal /HA18 für die Speicher der Hercules-Karte erzeugt werden, das den Zugriff der CPU um 1 oder 2 Takte verlängert. Da ich eine solche Karte aber nicht habe, will ich darüber auch nicht weiter nachdenken.

Das System verfügt noch nicht über eine Reset-Schaltung. Diese kann im einfachsten Fall durch Kurzschließen von /RESET (ECB-Bus, 31c) mit GND über einen Taster bereitgestellt werden. Da aber während der Zeit, in der der Taster gedrückt ist, kein Refresh der dynamischen RAMs erfolgt, gehen bei einem Reset Daten verloren. Eine einfache Schaltung mit einem 555-Timer und einem Taster erzeugt sowohl beim Einschalten als auch beim Betätigen des Tasters ein getimtes /RESET-Signal, das ein längeres Ausbleiben des Refresh verhindert (Abb.5).

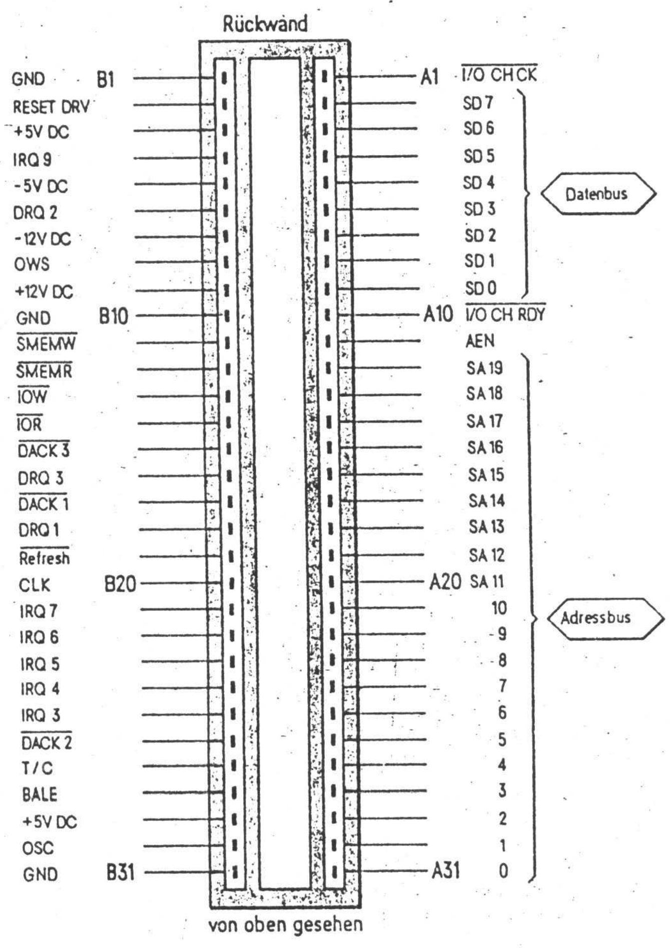


Freie Verdrahtung auf der Lötseite Signal IC,Typ,Pin IC,Typ,Pin ...
D0 18,273,8 10,27128,11
D1 18,273,7 10,27128,12
D2 18,273,4 10,27128,13; PC-Slot,A7
D3 18,273,13 10,27128,15
D4 18,273,8 10,27128,16; PC-Slot,A5
D5 18,273,14 10,27128,17
D6 18,273,17 10,27128,18
D7 18,273,18 10,27128,19
D0 17,64180,34 30,125,5
D1 17,64180,35 30,125,2
D2 17,64180,36 38,245,4
D3 17,64180,37 38,245,5
D4 17,64180,38 38,245,3
D5 17,64180,39 38,245,9
D6 17,64180,40 30,125,11
D7 17,64180,41 31,174,13
/HALT 17,64180,56 VG64-Leiste,25c
/REF 17,64180,57 VG64-Leiste,28a
/IOE 1,PAL1,1 17,64180,58; 19,32,10; PC-Slot,A10; 33,245,7; 37,PAL2,13
/ME 16,32,4 17,64180,59; 19,32,13; 33,245,2; 37,PAL2,14
/LIR 17,64180,61 37,PAL2,23
/WR 17,64180,62 19,32,12; 33,245,9; 37,PAL2,3
/RD 1,PAL1,2 17,64180,63; 19,32,1; 33,245,8; 37,PAL2,1
CLK 17,64180,64 VG64-Leiste,29c
/WAIT 17,64180,4 VG64-Leiste,10,a; PC-Slot,A10

/BUSAK   17,64180,5   37,PAL2,2
/BUSRQ   17,64180,6   VG64-Leiste,11a
/RESET   18,273,1   17,64180,7;   15,04,5;   31,174,1;   VG64-Leiste,31c
/NMI   17,64180,8   VG64-Leiste,20c
/INT0   17,64180,9   VG64-Leiste,21c
/ST   17,64180,12   35,245,2
A0   17,64180,13   10,27128,10;   25,PAL3,7;   32,157,2;   38,245,2
A1   17,64180,14   10,27128,9;   32,157,14/3;   36,245,6
A2   17,64180,15   10,27128,8;   32,157,13/5;   36,245,3
A3   17,64180,16   10,27128,7;   32,157,11/6;   36,245,4
A4   17,64180,17   10,27128,6;   25,PAL3,16;   32,157,10;   35,245,5
A5   17,64180,18   10,27128,5;   25,PAL3,6;   36,245,7
A6   17,64180,19   10,27128,4;   25,PAL3,5;   36,245,8
A7   17,64180,20   10,27128,3;   25,PAL3,4;   36,245,9
A8   1,PAL1,6   10,27,128,24;   17,64180,21;   25,PAL3,3;   35,245,3
A9   1,PAL1,5   10,27,128,25;   17,64180,22;   25,PAL3,2;   35,345,4
A10   1,PAL1,4   10,27,128,21;   17,64180,23;   25,PAL3,1;   35,345,5
A11   1,PAL1,3   10,27,128,23;   17,64180,24;   35,345,7
A12   17,64180,25;   10,27128,2;   20,86,5;   28,30,12;   33,245,5
A13   17,64180,26;   10,27128,27;   20,86,9;   20,30,11;   33,245,6
A14   17,64180,27;   20,86,2;   35,245,6
A15   17,64180,28;   20,86,12;   33,245,3
A16   17,64180,29;   12,157,5;   27,02,11;   35,245,4
A17   17,64180,30;   12,157,11;   27,02,12;   35,245,8
A18   17,64180,31;   24,00,5;   21,08,13/2;   35,245,9
A18'   21,08,4;   11,157,14
A19'   21,08,5;   11,157,2 B81   18,273,6;   25,PAL3,9;   29,125,4;   16,32,5;   15,04,9
/B81   18,273,16;   25,PAL3,11
Q0   30,125,5;   16,32,10
Q1   30,125,2;   21,08,12
X   16,32,8;   21,08,1
Y   24,00,6;   15,32,9
/M016   26,20,8;   1,PAL1,14
/M1216   28,30,8;   1,PAL1,22
MFxxxx   26,20,6;   1,PAL1,18
/OUTB8   37,PAL2,19;   9,273,11
/OUTF4   37,PAL2,18;   31,174,9
/INF4   37,PAL2,17;   30,125,10
/ADR   37,PAL2,15;   1,PAL1,17
/ROM   1,PAL1,20;   10,27128,20;   10,27128,22
/FLO   1,PAL1,16;   37,PAL2,19;   VG64-Leiste,z.B.23c
/KB   1,PAL1,19;   37,PAL2,20;   VG64-Leiste,z.B.23a
/RAM   1,PAL1,22;   16,32,1
/HA18   1,PAL1,21;   PC-Slot,A13
HA15   20,86,11;   PC-Slot,A16
HA14   20,86,3;   PC-Slot,A17
HA11   29,125,6;   PC-Slot,A20
RESET   15,04,6;   PC-Slot,B2
SEL   16,32,6;   32,157,1
+5V   22,1489,14;   20,86,14;   17,64180,32;   10,27128,28;   4,RAM,9
+5V   1,PAL1,24;   2,RAM,9
+5V   25,PAL3,20;   PC-Slot,A12/A14/A15/B29
GND   10,27128,14;   20,86,7;   26,20,7
GND   17,64180,33;10,04,7;   V24-Stecker,9/10
+12V   23,1488,14;   VG64-Leiste,13a
-12V   23,1488,1;   VG64-Leiste,14a



Pinout der HD64180-CPU



Pinout des PC-Slots

;Initialisierung der Hercules-Karte und der HD64180-CPU
;im Sonder-ROM-Bereich ab 3000H

```
;
;Diese Routinen werden aus dem ROM heraus angesprungen
;und übergeben zum Schluß wieder an die TRS-80 Boot-
;Routine. Dafür muß das ROM ab 0000H folgendermaßen
;geändert werden:

;vorher                                  nachher
0000    DI                               0000    DI
0001    XOR     A                        0001    XOR     A
0002    JP      0674h                    0002    JP      3050h
;
;Die Initialisierung schließt mit JP 0674H ab.
;
attrib      EQU     07h         ;D7=1 nicht blinkend bzw
                                ;normaler Hintergrund
                                ;D6-D4=0,  D2-D0=1
                                ;normale Darstellung
mode        EQU     0           ;D7=0   HRG-Page 0
                                ;D5=1   Blinken erlaubt
                                ;D3=1   Video-Signal aktiv
                                ;D0=0   Text-Betriebsart
config      EQU     3           ;D0=1   HRG erlaubt
                                ;D1=1   HRG-Page 2 enabled
;
;
            Org     3000h       ;Anfang Sonder-ROM
;
;Initialisierungdaten für den CRTC6845
;64*16-TRS80-Standard-Einstellung
tab         DEFB    6eh,40h,50h,0ah,14h,06h,10h,12h
            DEFB    02h,0eh,29h,09h,04h,00h,00h,00h
;80*25-Modus
            DEFB    6eh,50h,58h,0ch,1fh,02h,19h,1bh
            DEFB    02h,09h,29h,09h,00h,00h,00h,00h
;64*24-Modus
            DEFB    6eh,40h,50h,0ah,1bh,09h,18h,19h
            DEFB    02h,0ah,29h,09h,02h,00h,00h,00h
;64*32 interlaced
            DEFB    6eh,40h,50h,0ah,13h,02h,10h,11h
            DEFB    03h,0fh,2fh,0fh,00h,00h,00h,00h
;92*22-Modus
            DEFB    70h,5ch,61h,08h,1dh,08h,16h,19h
            DEFB    02h,0bh,29h,09h,00h,00h,00h,00h
```

```
;
                LD      SP,8000
                LD      A,1
                OUT     (0f4h),a        ;obere 256K einstellen
                LD      A,81h           ;Attribut-Speicher mit
                OUT     (0b8h),A        ;Standard-Wert füllen
                LD      HL,3800h        ;Video-RAM ab 3800h
                LD      A,attrib
                LD      (HL),A
                LD      DE,3001h
                LD      BC,07ffh        ;2K Speicher
                PUSH    HL
                PUSH    DE
                PUSH    BC
                LDIR
                POP     BC
                POP     DE
                POP     HL
                LD      A,1             ;Umschalten auf Text
                OUT     (0b8h),A        ;-Speicher
                LD      A,' '           ;mit Blanks füllen
                LD      (HL),A
                LDIR
                LD      HL,0            ;ROM-Inhalt ins RAM
                LD      DE,0            ;kopieren
                LD      BC,3700h
                LDID
;
;HD64180 initialisieren
                LD      A,10h           ;WAITs: Memory 0, I/O 2
                DB      0edh,39h,32h    ;OUT= (DCNTL),A
                LD      A,83h           ;Refresh-Control
                DB      0edh,39h,36h    ;optimieren
                LD      A,44h           ;Common 0-32K, Bank 32-64K
                DB      0edh,39h,3ah    ;einstellen
                XOR     A               ;Common als Bank benutzen
                DB      0edh,39h,3ah    ;Bank bei 08000-0FFFF
;
;CRTC6845 für 64*16 Zeichen initialisieren
                LD      HL,tab          ;HL => CRTC-Parameter
                CALL    setcrtc         ;raus an CRTC
;
;Video-Signal einschalten
                LD      A,08h           ;Video-Signal aktiv
                OUT     (0b8h),A        ;Video bei 3C00-3fff
                                        ;Text-Modus eingesch.
                                        ;Blinken gesperrt
                JP      0674h           ;und weiter im TRS80-
                                        ;Boot-Gebaren
;
;CRTC-6845 initialisieren
setcrtc         XOR     A               ;für Video-Mode-Umschal-
                OUT     (0bfh),A        ;tung nötig
                LD      B,16            ;16 Bytes ausgeben
                LD      C,0b5h          ;CRTC-Daten-Port
                XOR     A               ;CRTC-Register-Nr ab 0
crtclp          OUT     (0b4h),A        ;1. Regsiter
                INC     A               ;A => nächstes Register
                OUTI                    ;raus das Zeugs
                JR      NZ,crtclp
                LD      A,config        ;HRG voll freigeben
                OUT     (0bfh),A
                RET
                END
```

Assembler-Listing für PAL1
PAL20L8
Helmut Bernhardt 22.01.89

Erzeugen der mm -Freigabesignale
ioe rd a10 a11 a9 a8 nc b80 b81 b82 b84 gnd
b86 m016 nc flo adr mfxxxx kb rom ha18 ram m1216 vcc

/ha18 = /b86 */b81 */b80 */m1216 *a11 *a10
+ /b86 */b81 * b80 */m1216 *a11
+ /b86 * b81 */mfxxxx

/kb = /b86 */b80 */rd */m1216 *a11 */a10 */a9 */a8

/flo = /b84 */m1216 */a11 *a10 *a9 *a8 */adr
+ b84 */ioe */adr

/rom = /b82 */m016 *m1216 */rd
+ /b82 */m1216 */a11 */a10 */rd
+ /b82 */m1216 */a11 * a10 *a9 */rd
+ /b82 */m1216 */a11 * a10 *a9 */a8 */rd

/ram = ioe *ha18 *kb *flo *rom

Assembler-Listing für PAL2
PAL20L8
Helmut Bernhardt 24.01.89
I/O-Decoder und Datentreiber-Steuerung
rd busak wr a0 a2 a3 a4 a1 a5 a6 a7 gnd
ioe me adr herc inf4 outf4 outb8 flo kb rein lir vcc

/inf4 = /ioe /rd *a7 *a6 *a5 *a4 */a3 *a2 */a1 */a0

/outf4 = /ioe */wr *a7 *a6 *a5 *a4 */a3 *a2 */a1 */a0

/outb8 = /ioe */wr *a7 */a6 *a5 *a4 *a3 */a2 */a1 */a0

/adr = a7 *a6 *a5 */a4

/herc = /ioe *a7 */a6 *a5 *a4

/rein = busak */ioe */lir
+ /ioe */rd *inf4 *herc
+ /busak *rd
+ /flo */rd
+ /kb */rd

Assembler-Listing für PAL3
PAL16L8
Helmut Bernhardt 23.01.89
Adreßübersetzung für die Hercules-Karte
a10 a9 a8 a7 a6 a5 a0 me b81 gnd
b87 ha0 ha5 ha6ha7 a4 ha8 ha9 ha10 vcc

/ha0 = /me * /b81 * /87
+ /me * b81 * /a0
* /me * /a0

/ha5 = /me * /b81 * /a4
+ /me * b81 * /a5
+ me * /a5

/ha6 = /me * /b81 * /a5
+ /me * b81 * /a6
+ me * a6

/ha7 = /me * /b81 * /a6
+ /me * b81 * /a7
+ me * /a7

/ha8 = /me * /b81 * /a7
+ /me * b81 * /a8

/ha10 = /me * /b81 * /a9
+ /me * b81 * /a10

Pinbelegung des V24-Steckers CN2

Signal	Pin	Pin	Signal
/CTS1	1	2	TXS1
/RTS0	3	4	RXA0
/CTS0	5	6	RXA1
TXA0	7	8	TXA1
GND	9	10	GND

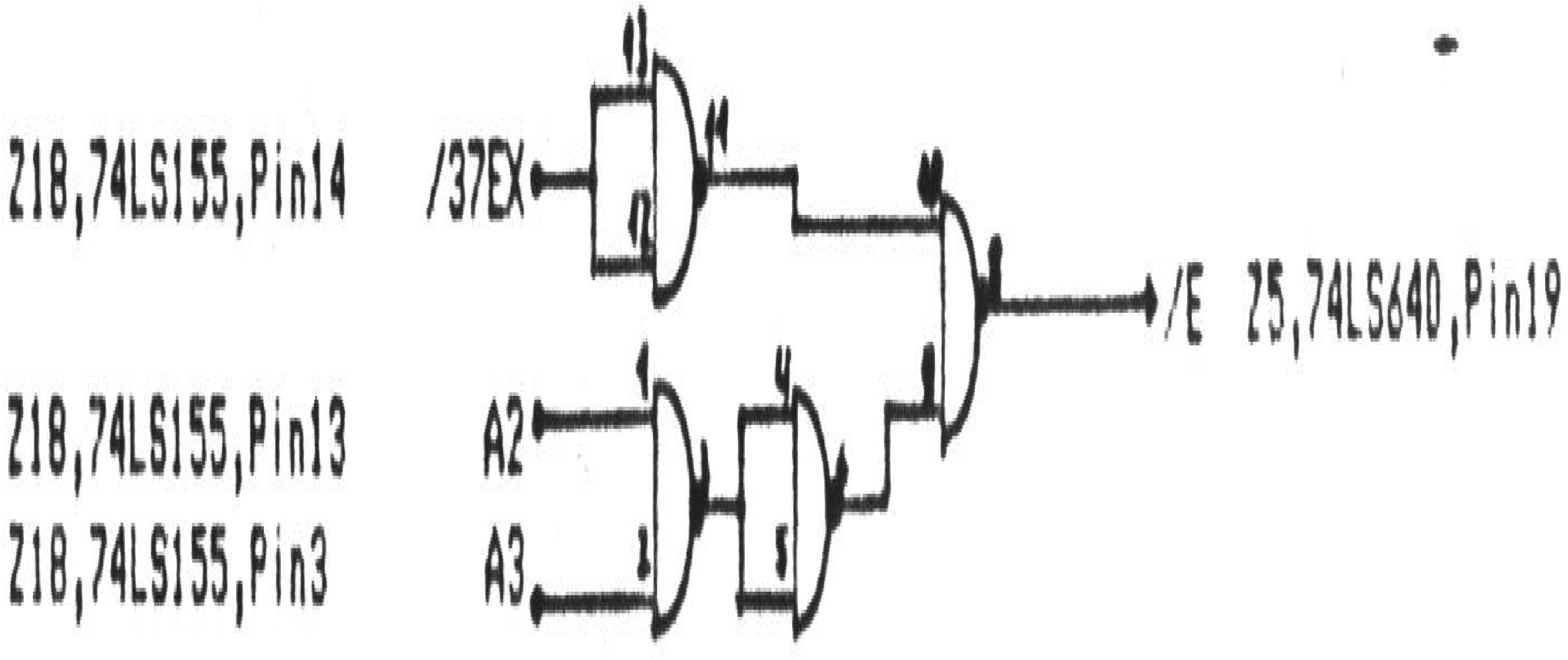
Endlich Schluß mit den Problemen beim EXP1-Floppy- Controller von RB-Elektronik

Helmut Bernhardt

Mehrmals hatte ich schon Probleme mit EXP1-Floppy- Controllern. Nach versuchsweisem Auswechseln einiger ICs war der Spuk dann meist vorbei; - leider aber immer nur für begrenzte Zeit. Inzwischen habe ich einen grundsätzlichen Fehler in der Schaltung des EXP1 erkannt, der dafür verantwortlich ist:

Die Freigabe des invertierenden Datentreibers 74LS640 wird durch AND-Verknüpfung der Signale /37ECRD (Z18, 74LS155, Pin4) und /37ECWR (Z16, 74LS155, Pin12) erzeugt. Das hat zur Folge, daß bei Schreibzugriffen auf den Floppy-Controller WD1791 der Treiber erst im gleichen Moment freigegeben wird, wie das /WR-Signal am FDC mit seiner negativen Flanke die Daten in den Controller pustet. Durch die Gatterlaufzeit bei der Freigabe des Treibers können hinter dem Treiber noch keine gültigen Daten anliegen. Es ist erstaunlich, daß das Board überhaupt funktioniert hat.

Abhilfe schafft ein Freigabesignal für den Treiber, das unabhängig von /RD und /WR ist. Dieses läßt sich mit einem 74LS00 folgendermaßen erzeugen:

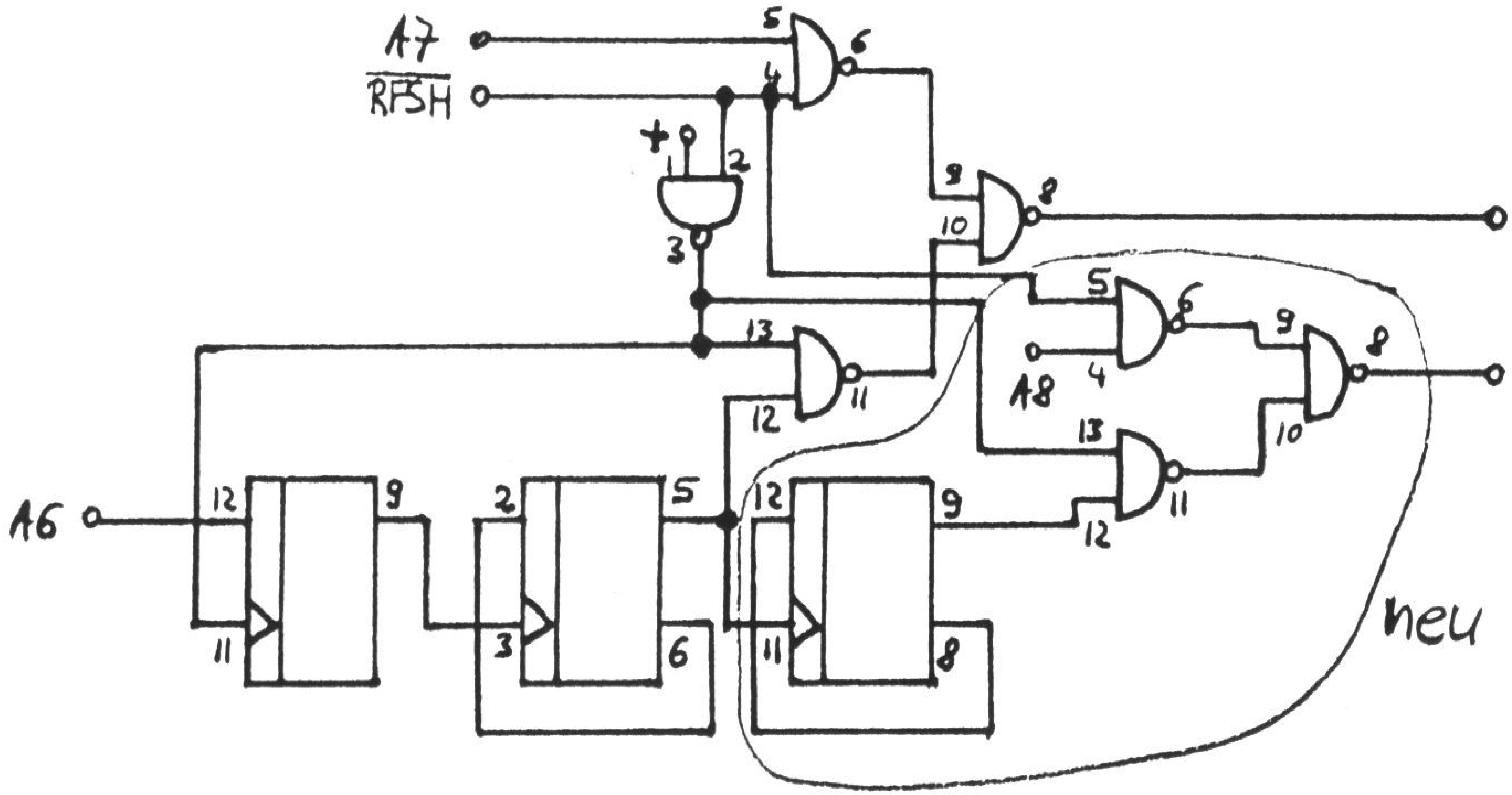


Das dafür nötige 74LS00 wird mit den Pins 7 und 14 auf irgendein DIL14-IC (**nur nicht 74LS92**) huckepack-gelötet und mit den entsprechenden Pins von Z18 und Z5 verdrahtet. Der Pin 8 von Z15,74LS08 wird aus der Fassung gebogen.

1MB-Banker mit 1MB

Da Helmuts Banker ein ganzes Megabyte verwalten kann, liegt es natürlich nahe, den Speicher auch soweit auszubauen. Die Möglichkeit mit vier übereinander gestapelten 256er RAMs funktioniert zwar und sieht phänomenal interessant aus, ist aber sicher nicht der Weisheit letzter Schluß. Einmal braucht man viel Platz nach oben und zweitens werden die Dinger ganz schön warm, entsprechend ist der Strom, der da verheizt wird. Inzwischen sind aber die Mega-Chips auch schon in bezahlbaren Regionen angelangt und damit löst sich fast alles in Wohlgefallen auf. Probleme bereiten nur noch die Größen der Chips und der Refresh. Da die neuen Käfer zwei Beine mehr haben als die alten und die Belegung von der der 256er etwas abweicht, habe ich einen Platinenstreifen mit den 18-poligen Fassungen bestückt und die Signale einfach zu den alten Fassungen rübergefädelt. Geht einwandfrei und läuf auch mit 10MHz problemlos. Aber jetzt kommts. Für die 256er mußte der Refresh-Zähler bereits auf 8 Bit aufebohrt werden und die Mega-Chips brauchen sogar einen 9-Bit-Refreshzähler. Mit zwei weiteren ICs läßt sich aber auch dieses Problem ohne allzugroßen Aufwand lösen. Wenn man beim Umbauen mit dem Refresh anfängt, muß sich der Rechner hinterher noch ganz genauso verhalten wie vorher, da es die 256er RAMs überhaupt nicht stört, wenn sie mit einer 9-Bit-Adresse betan werden.

Alexander Schmid



1990-Club80-07 Nr.31 S.17-22

Harddisk-Einbindung ins CP/M3-BIOS

hier am Beispiel des OMTI 5527 RLL Controllers und einer Platte mit 615 Zylindern und 4 Köpfen (32 MB im RLL Format)

Helmut Bernhardt, Peter Petersen

Die Hardware

Der Aufwand in Hardware, um einen PC-Harddisk-Controller wie die verschiedenen OMTI-Typen an den ECB-Bus oder auch an andere Z80-Bus-Systeme anzuschließen, ist minimal. Beim ECB-Bus bestehen die Probleme hauptsächlich in der ungeeigneten Geometrie des Controllers, der die Breite der Europakarten von 10 cm überschreitet. Die Montage auf eine ECB-Bus Trägerkarte, die auch die beiden zur Busanpassung nötigen ICs aufnimmt, muß für jedes Gerät individuell gelöst werden.

Die Busanpassung beschränkt sich auf die Umverlegung der ECB-Adressen an die Slotadressen des Controllers, damit bei einem Zugriff der CPU unter einer für den Controller vorgesehenen Portadresse am Controller die Portadresse 320H(-323H), unter der der Controller im PC decodiert ist, anliegt. Außerdem müssen aus den ECB-Signalen /RD, /WR, und /IORQ die PC-Bus-Signale /IOR und /IOW mit zwei OR-Gattern eines 74LS32 erzeugt werden. Auch das low aktive /RESET des ECB-Bus muß über einen 74LS04-Inverter in das high aktive RESET des PC-Bus umgesetzt werden.

Da 4 aufeinanderfolgende Portadressen benötigt werden, müssen A0 und A1 direkt an den Controller gelangen. Für eine Adreßlage bei D0H-D3H am ECB-Bus ist am Ende dieses Artikels eine Beschaltung gezeigt, die folgende Pegel-Übersetzung bewirkt:

```
Adresse        :  A9  A8  A7  A6  A5  A4  A3  A2  A1  A0
---------------------------------------------------------
ECB   Port D0H :   =   =   1   1   0   1   0   0   x   x
PC   Port 320H :   1   1   0   0   1   0   0   0   x   x
```

Außerdem ist an AEN und A10-A19 des Controllers GND und an /DACK3 und /MEMR +5V zu legen. Außer dem OMTI 5510 benötigen alle OMTI-Controller am Slot-Pin B9 +12V Versorgungsspannung. Selbstverständlich müssen die Daten D0-D7 an die gleichen Eingänge des Controllers gelegt werden.

Die Software

Wesentlich anspruchsvoller ist die Software-Einbindung der Festplatte in das BIOS des CP/M plus.

Mit den unter CP/M bestehenden Möglichkeiten zur Verwaltung der Files auf einer Platte (Unterteilung in User 0 bis User 15) lassen sich nur begrenzte Mengen an Files übersichtlich handhaben. Deshalb ist es bei größeren Platten sinnvoll, diese in mehrere logische Laufwerke zu partitionieren. Noch handhabbar sind dabei Größen von einigen Megabyte Kapazität. Danach könnte eine 32MB-Platte in 4 Partitionen aufgeteilt werden. Die sich anbietende Aufteilung, daß alle Spuren unter einem Kopf als ein Laufwerk behandelt werden, bedeutet, daß nach je 26 Sektoren der Kopf eine Spur weiter bewegt werden muß. Der Spurwechsel ist bei Festplattenzugriffen aber die zeitraubendste Aktion und sollte so wenig wie möglich vorkommen.

Deshalb wurde die Partitionierung hier auch anders durchgeführt: Eine logische Spur umfaßt alle 4 Spuren (Köpfe) eines Zylinders. Es werden nacheinander erst die 4 Spuren unter allen 4 Köpfen gelesen/geschrieben, bevor der Kopfschlitten um eine Position weiterbewegt wird. Dadurch können nacheinander 104 Sektoren verholt werden, ohne zwischendurch steppen zu müssen. Einer Partition wird dabei ein zusammenhängender Bereich von Zylindern zugeordnet: die Nummer des letzten absoluten Zylinders einer Partition ist die Anzahl Spuren und die Nummer des letzten absoluten Zylinders der vorherigen Partition ist die Anzahl reservierter Systemspuren.

```
Spur-Nr.    0 1         154          307          460          615
Partit. 0    =############--------------------------------------
Partit. 1    =============#############-------------------------
Partit. 2    =========================#############-------------
Partit. 3    ======================================#############
       ===== reservierte Systemspuren
       ##### Datenspuren der Partition
```

Das BDOS kennt für Disk-I/O nur die Adressen einer Tabelle mit dem Namen @dtbl (drive table). In dieser Tabelle stehen die Adressen von 16 Disk-Parameter-Headern (DPHs) für 16 Laufwerke in der Reihenfolge ihrer logischen Namen (A, B, ...P). Nicht existierende Laufwerke erhalten hier den Eintrag 0. Wenn in einem Sourcecode-Modul nur die Drive Table und die DPHs und DPBs für RAM-Floppy und Harddisk enthalten sind, könnte das so aussehen:

```
        public  @dtbl                    ; Drive Table
        public  fdsd0,fdsd1,fdsd2,fdsd3  ; dph's fuer 4 Floppies
        public  fdram                    ; dph RAM-Floppy
        public  hdsk0,hdsk1,hdsk2,hdsk3  ; dph's fuer Harddisk 0-3
        extrn   hdlogi,hdinit,hd1init    ; Harddisk Routinen
        extrn   hdwrit,hdread
        maclib  xcpm3           ; mit hdpb-Macro für große Drives
        cseg                    ; Drivetable und dpb's im Common
; Drive Table (zeigt auf die DPB's der verfuegbaren Laufwerke)
@dtbl   dw      fdsd0,fdsd1,fdsd2,fdsd3  ; Floppy Drives
        dw      fdram                    ; RAM-Floppy
        dw      hdsk0,hdsk1,hdsk2,hdsk3  ; Harddisks
        dw      0,0,0,0,0,0,0            ; nicht benutzt
;Die Einträge in @dtbl weisen auf Adressen der DPHs, wobei die Lables
;nicht am Anfang der xDPHs stehen. Der Aufbau der xDPHs sieht so aus:
        desg            ; die DPHs koennen im gebankten RAM stehen
; extended disk parameter header (xDPH) der Harddisk-Partitionen
drva:   dw      hdwrit          ; Routinen zum Schreiben
        dw      hdread          ;   und Lesen eines Sektors
        dw      hdlogi          ; Einloggen des Drives
        dw      hdinit          ; Initialisieren des HDC
        dw      0               ; relatives drive 0
        dw      0               ; media flag :  not used
hdsk0   dph     0,hddpb0        ; Skew-Tabelle und xDPB
drvb:   dw      hdwrit,hdread,hdlogi
        dw      hdinit          ; kann auf RET zeigen
        db      1               ; relative drive 1
        db      0               ; media flag :  not used
hdsk2   dpb     0,hddpb1
;------------------------------
drvd:   dw      hdwrit,hdread,hdlogi
        dw      hd1init         ; kann auf RET zeigen
        db      3               ; relative drive 3
```

```
        db      0               ; media flag :  not used
hdsk3   dph     0,hddpb3
```

'dph' ist ein Macro-Aufruf, der in CPM3.LIB aufgelöst wird. Am Anfang des Moduls, in dem das xDPH steht, ist

'   maclib   cpm3'

anzugeben, damit RMAC den Macro bearbeiten kann (aus später benannten Gründen sollte lieber XCPM3.LIB benutzt werden).

Da der Skew der Festplatten bereits beim Formatieren durch physika- lischen Versatz der Sektoren auf der Spur vorgegeben wird, ist ein Soft-Skew nicht nötig: anstelle der Adresse einer Skew-Tabelle kann als erster Parameter für dph '0' angegeben werden.

Die Einträge hddpb0 bis hddpb3 geben die Adressen der xDPB's der vier Harddisk-Partitionen an. Diese können wieder mit einem Macro erzeugt werden:

```
        cseg            ; die DPHs müssen im Common liegen
; hard disk parameter block macros
hddpb0  hdpb    512,104,154,4096,2048,1,8000h   ; partition 0
hddpb1  hdpb    512,104,307,4096,2048,154,8000h ; partition 1
hddpb2  hdpb    512,104,460,4096,2048,307,8000h ; partition 2
hddpb3  hdpb    512,104,615,4096,2048,460,8000h ; partition 3
;                |   |   |   |    |    |   |
;                |   |   |   |    |    |   +----flag für nonre-
;                |   |   |   |    |    |        movable media
;                |   |   |   |    |    +----reserv. Systemspuren
;                |   |   |   |    +----max. Anzahl DIR-Einträge
;                |   |   |   +----Blockgröße (die durch einen
;                |   |   |        DIR-Eintrag verwaltet wird)
;                |   |   +----Anzahl Zylinder des Laufwerks
;                |   +----Sektoren pro Zylinder (4 tracks mit 26 secs)
                 +----Bytes pro Sektor
```

Aufgrund des optionalen Eintrags 8000h wird das Laufwerk vom BDOS als 'nonremovable media' angesehen, woraufhin nicht bei jedem Zugriff ein neues Einloggen durchgeführt wird.

'hdpb' ist (ähnlich dem dpb in CPM3.LIB) ein in XCPM3.LIB zusätzlich vorhandener Macro, der auch für Laufwerke mit einer Kapazität von mehr als 8 MB einen richtigen Disk Parameter Block erzeugt.

Das Source-Modul (angenommen, es heißt MOVE.ASM) das diese Datenstrukturen enthält, wird mit **'rmac move $pz sz'** assembliert (in einem SUBMIT-File sind 2 $-Zeichen einzutragen). M80 kann die Macrobibliotheken CPM3 und XCPM3 nicht benutzen. Da für die Datenstrukturen aber keinerlei Maschinenbefehle nötig sind, braucht man sich auch nicht unbedingt mit den von RMAC einzig verstandenen 8080-Mnemonics herumschlagen. Die eigentlichen Disk-I/O-Routinen können in einem anderen Modul formuliert werden, wobei man sich die Fähigkeit von M80 zunutzemachen kann, Z80 Mnemonics zu verstehen.

Im Kopf des Moduls mit diesen Datenstrukturen sind sämtliche Lables wie @dtbl auf die aus anderen Modulen zugegriffen wird mit 'public' bekanntzugeben, und alle Einträge in diesen Tabellen, die nicht in diesem Modul selbst als Lable auftauchen, mit 'extrn' als außerhalb stehend zu benennen.

Bei ausreichend bereigestelltem RAM-Speicher legt GENCPM Direktory Buffer in Bank 0 (Systembank) und Data-Buffer in anderen dafür angegebenen Banks an (siehe unten). Die Routinen 'hdwrit' und 'hdread' müssen also den Datentransfer zwischen Controller und verschiedenen Banks unterstüzen. Beim Aufruf bekommen sie in den in BIOSKRNL zugewiesenen Variablen entsprechende Parameter übergeben.

```
@dbnk   ( 8Bit) Source-/Ziel-Bank für Holen/Ablegen eines Sektors
                beim Schreiben/Lesen
@dma    (16Bit) RAM-Adressen in der Bank, von wo ab mit aufsteigender
                Adresse die Daten des Sektors zu holen/abzulegen sind
@adrv   ( 8Bit) absolute Laufwerks-Nummer (entsprechend Position
                in qdtbl)
@rdrv   ( 8Bit) relative Laufwerks-Nummer (Zählung der Laufwerke
                für einen Controller)
@trk    (16Bit) Spur des Laufwerks
@sect   (16Bit) Sektor der Spur
```

Diese Daten stehen alle in der Systembank 0 und müssen ausgewertet werden, bevor die in @dbnk vorgegebene Bank eingestellt wird und diese Daten dann nicht mehr zur Verfügung stehen.

Die Übertragung von Daten erfogt direkt vom/zum Controller zur/von der Buffer-Bank. Da diese Bank dann eingeschaltet sein muß, kann die Übertragung nur von dieser Routine im Common durchgeführt werden. Das hat den Vorteil, daß der Common-Bereich keinen 512Byte-Zwischenspeicher enthalten muß, was auf Kosten der TPA geht, und daß nur ein Datentransfer nötig ist, was Geschwindigkeit bringt.

Befehle an den OMTI-Controller bestehen aus einer Folge von 6 Bytes, die nacheinander an die relative Portadresse 0 des Controllers zu übertragen sind. Das erste Byte ist das eigentliche Befehlsbyte und die folgenden 5 Bytes sind Parameter des Befehls. Diese Bytefolge wird zunächst im RAM erzeugt und dann an den Controller ausgegeben. Das Parameterfeld im RAM sieht so aus:

```
cfield:                 ; Befehlsparameter-Feld für HDC
head:   db      0       ; D7 = Bit 10 der Zylinder-Nummer
                        ; D6 = 0. D5 = Nummer des zu bedienenden Lauf-
                        ; werks (0 oder 1; an dem OMTI lassen sich
                        ; 2 Festplatten anschließen) nicht mit der
                        ; Partition eines Laufwerks verwechsel!
                        ; D4-D0 : Head-Nummer
sect:   db      0       ; D6,D7 : Bit 9 und 8 der Zylinder-Nummer
                        ; D5-D0 : Sector-Nummer
cylind: db      0       ; D7-D0 : Bit 7 - 0 der Zylinder-Nummer
bcount: db      1       ; Interleave-Count beim Formatieren
                        ; Blockcount bei MultIO
termin: db      2       ; D7-D5 : Command-Controll-Byte
                        ; D4-D0 : nicht benutzt
```

Für die Beschreibung der 'termin'-Bytes sei auf das Handbuch der OMTI-Controller verwiesen (das wollen wir uns hier sparen).

Vor der Übertragung dieser Parameter ist der eigentliche Befehl direkt an den Controller auszugeben. Der Inhalt dieses Feldes wird in der Routine hdrw_common

berechnet und in die Tabelle eingetragen 'hdscmd' übergibt die Bytes an den Controller.

Die zwingend im Common anzusiedelnden Routinen 'hdrd' und 'hdwr' sowie der von beiden genutzte Ausgang 'termcmd1', der wieder die Bank 0 einschaltet, machen zusammen knapp 60 Bytes des kostbaren Common-Speichers aus, dem 512 Bytes eines Zwischenspeichers bei der anderen Variante gegenüberstehen.

Bei einem Prof mit 9,2 MHz Systemtakt ist damit ein Interleave 1:1 möglich. Wegen insgesamt bereitgestellter Data-Buffer von 64K RAM erscheint ein erster Zugriff auf die Festplatte recht langsam, weil dann immer gleich riesige Datenmengen bewegt werden. Nachfolgende Zugriffe können dann aber aus dem Buffer leben und erreichen dann die Geschwindigkeit einer RAM-Floppy.

Weiteres Tuneup am BIOS

Da die Speicheraufteilung des Prof 180 recht ungünstig ist und dadurch bedingt auch die RAM-Floppy-Routinen umständlich mit '?xmove' über einen 128 Byte Zwischenspeicher im common arbeiten, wurde auch die ganze Speicherkonfiguration nach Conitec's Vorstellung aufgegeben und durch eine sinnvollere ersetzt.

```
Speicherbereich         Adreßlage nach Conitec        neue Adreßlage
----------------------------------------------------------------------
Systembank                  00000h-0efffh             00000h-0ffffh
Common                      0f000h-0ffffh             0f000h-0ffffh
TPA                         40000h-4efffh             10000h-1efffh
CCP-Buffer                  4f000h-4ffffh             1f000h-1ffffh
weitere Banks                -----------              20000h-2ffffh
RAM-Floppy                  10000h-3ffffh             20000h-7ffffh
                          + 50000h-7ffffh
----------------------------------------------------------------------
```

Die ursprüngliche Einteilung von Conitec ermöglicht den Einsatz von 4164-RAMs, wobei die Selektion der beiden 64K-Blöcke durch den Pegel von A18 erfolgt (was bei 41256-RAMs auch sinnvoll ist). Entsprechend liegen diese beiden Blöcke auf den physikalischen Adressen 0xxxxh und 4xxxxh. Da bei den heutigen RAM-Preisen der Einsatz von 4164-RAMs unwahrscheinlich ist und hier 41256-RAMs stecken, wurde auf die Möglichkeit des Abrüstens auf 128K verzichtet, so daß der Bereich, der für Banks und/oder RAM-Floppy zur Verfügung steht, zusammenhängt. Das vereinfacht die RAM-Floppy-Routinen gewaltig.

Der Zugriff auf die RAM-Floppy erfolgt nur im DMA-Betrieb und zur Vereinfachung und Beschleunigung wurde die Sektorgröße von 128 auf 1024 Bytes erhöht und die Größe einer Spur auf 64K festgelegt. Die Track-Nr kann dann direkt (durch Addition des Offsets des RAM-Floppy-Anfanfs) in das Adreß-Zusatzbyte für den DMAC umgesetzt werden und aus der Sektor-Nummer kann durch Shiften um 2 Bits der Adreßanteil A15-A10 erhalten werden, A9-A0 sind immer 0 für die Anfangsadresse eines Sektors.

Die RAM-Floppy-Routinen sind in RFLO.MAC versammelt. DPH und DPB der RAM-Floppy sowie die Routinen und Tabellen für Bankumschaltung und DMA sind neben den DPHs und DPBs für die Festplatte in MOVE.ASM enthalten.

Durch die neue Speichereinteilung muß GENCPM bei der Erzeugung der Memory Segment Table auch mit anderen Parametern versorgt werden, als den in GENCPM.DAT bisher enthaltenen. Bei 64K Data-Buffer sind insgesamt 3 Segmente anzugeben (Common und TPA sowie CCP-Buffer werden hier nicht mitgezählt. Für jedes Segment sind dann je 3 Bytes anzugeben, die folgende Bedeutung haben.

```
  1) high Byte der logischen Anfangsadresse der Bank
  2) high Byte der Länge der Bank
  3) Bank-Nummer
```

Eine Angabe '00.80.02' bedeutet, daß die Bank 2 von 0000h bis 7fffh reicht (logische Adressen). Wo diese Bank physikalisch im Speicher liegt, spielt keine Rolle.

Für die Bank 0 sind diese Angaben (speziel beim Prof 180) etwas komplizierter. Der Prof 180 benutzt 4 KByte des Boot-ROMs (bzw. des auf gleiche Adressen ins RAM kopierten ROM-Inhalts) vom BIOS des CP/M aus. Deshalb muß für die Systembank als high Byte der Anfangsadresse 10 an- gegeben werden, damit diese ROM-Routinen nicht überschrieben werden. Als Länge ist die Differenz des von GENCPM vorher ausgegebenen Anfangs des gebanten BDOS abzüglich dieser Adresse anzugeben. Die Bank-Nr ist natürlich 00.

```
    UM unseren CP/M-Diskothekar nicht mit Format-Konvertier-Problemen
    zu überlasten, bitte ich darum, bei Interesse an den BIOS-Sources
    für die Festplatte und die neue Variante der RAM-Floppy mir eine
    formatierte Diskette mit den gerade verfügbaren Angaben zum Format
    und möglichst einigen Text-Files darauf (damit ich prüfen kann, ob
    ich das Format richtig eingestellt habe) zu schicken. Beiliegen-
    des Rückporto wird auch gerne gesehen.              Helmut
```

Zum Thema Harddisk unter CP/M geben folgende Artikel was her:

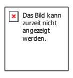
Andreas Zippel 'Hart, schnell und sicher'. c't 8/86, S.52ff; c't 9/86, S.105; c't 10(6 S.144ff

Martin Rost, Detlef Grell 'Harddisk-Controller einmal anders'. c't 4/87, S.138ff; c't 9/87, S.138ff

Christian Persson 'OMTI-Alternative'. c't 5/89, S. 244

Anschluß des OMTI-Controllers 5527 an den ECB-Bus

```
   ECB-Bus          PC-Slot              ECB-Bus          PC-Slot
Pin-Nr Signal    Signal Pin-Nr        Pin-Nr Signal    Signal Pin-Nr
------------------------------        ------------------------------
   c9  A7-----------A9  A22              c2  D0-----------D0  A9
   a9  A6-----------A8  A23             c14  D1-----------D1  A8
   a8  A5--------A7,A6  A24,A25          c4  D2-----------D2  A7
   a7  A4-----------A5  A26              a4  D3-----------D3  A6
   c6  A3-----------A4  A27              a5  D4-----------D4  A5
   a6  A2--------A3,A2  A28,A29          a2  D5-----------D5  A4
   c7  A1-----------A1  A30              a3  D6-----------D6  A3
   c5  A0-----------A0  A31              c3  D7-----------D7  A2
```



Club 80 Projekt:

Bau einer Terminal-Karte auf Basis der PC-Hercules-Karte

Der GENIE 1 mit Hercules-Karte als Video-Interface ist ja ganz witzig aber für die meisten Club-Mitglieder uninteressant, weil die Strippenzieherei einfach zu aufwendig ist und die Festlegung auf die GENIE-Kompatibilität zu beengend ist.

Um das günstige Preis/Leistungs-Verhältnis der Hercules allgemeiner zugänglich zu machen, wurde angeregt, als Projekt im Club 80 eine Terminal-Karte zu bauen, die zusammen mit der Hercules in jedem Computer als Konsole eingesetzt werden kann. Das Terminal soll mit Z80-Power und entsprechender Betriebssoftware fast ein eigenständiger Text-/Grafik-Computer sein, der in Arbeitsteilung mit der CPU Video-Ausgaben erledigt.

Da eine EGA- oder VGA-Karte ähnlich wie die Hercules-Karte funktioniert (erweiterter I/O-Adreßbereich, anderer Memory-Adreßbereich), ist das Prinzip auch auf diese Karten anwendbar. Es bestehen dort aber bei den über die IBM-Vorgaben hinausgehenden Features unterschiedliche Lösungen, die nicht ausreichend dokumentiert sind. Außerdem benötigen diese Karten Monitore, deren Preise weit über den Aktionsradius des Bastlers hinausgehen, so daß wir uns zunächst auf die Hercules-Karte beschränken sollten. Wenn wir das im Griff haben, können wir immer noch überlegen, ob wir nach den Sternen greifen wollen.

Von Andreas Magnus habe ich als letzte Meldung vor längerer Zeit erfahren, daß er auch schon einen entsprechenden Artikel ins Info gesetzt hat; ich weiß allerdings nicht, was er dort schon angekündigt hat. Deshalb möge man mir nachsehen, wenn einige Aussagen hier zum zweiten mal erscheinen. Da inzwischen schon etwas festere Vorstellungen bestehen, ist dieser Beitrag aber unumgänglich.

Die Kommunikation der Terminal-Karte mit der CPU soll über einen bidirektionalen Parallelport (Datenaustausch) und einen Statusport (Handshaking) erfolgen. Zwischen CPU und Terminal werden Befehle der CPU an das Terminal vereinbart, die das Terminal zur selbständigen Bearbeitung der Hercules anregen. Die bisherige Vorstellung sieht dabei im Textmodus die Emulation eines Televideo 950 Terminals und im Grafikmodus die eines Tektronix 4014 Grafikterminals vor.

Während für die Hardware des Terminals schon feste Vorstellungen bestehen, ist für die Betriebssoftware noch alles offen. Zur Erstellung eines Betriebsprogramms für das Terminal sind alle Club-Mitglieder herzlich eingeladen. Der Umfang der zu erstellenden Software ist so groß, daß jedes Gramm Gehirnschmalz im Club gefragt ist.

Durch das Herumwerkeln mehrerer Programmierer ist die Funktion eines Software-Koordinators nötig (Freiwillige bitte vortreten !). Dieser Job bedeutet, einen Überblick über die Gesamtarbeit zu gewinnen, diese in Module zu unterteilen und die Zusammenarbeit der einzelnen Module durch Festlegen von gemeinsam genutzten Datenfeldern und zu übernehmenden und zu übergebenden Parametern der einzelnen Module (in den CPU-Registern) zu gewährleisten. Diese Vorgaben sind von den Programmierern der einzelnen Module einzuhalten. Wer dazu Vorstellungen hat, möge bitte nicht durch bescheidene Zurückhaltung versuchen, einen guten Eindruck zu machen.

Auch wenn jemand zu Teilaspekten der Betriebssoftware Ideen hat und die Programmierung dieses Parts übernehmen möchte (oder auch nur seine Idee einbringen möchte), kann er einen wertvollen Beitrag zu dem Projekt leisten.

Als Bestechung in Naturalien bekommt jeder, der sich an dem Projekt beteiligt, von mir eine geätzte Terminalkarte geschenkt, die er allerdings selbst und mit eigener Finanzierung der Bauteile aufbauen muß. Hercules-Karten kann ich auch günstig beibringen (zur Zeit für 40,-DM). Wer den Job des Software-Koordinators übernehmen will, bekommt auch die Hercules geschenkt (wenn der Club mit seinem gewaltigen Konto das nicht übernehmen will, stifte ich die Karte).

Um das ganze Projekt etwas konkreter darzustellen, will ich im folgenden die bestehenden Vorstellungen zur Hardware und Software, wie Andreas Magnus und ich uns das vorstellen, breittreten. Wie gesagt, davon ist noch nichts endgültig. Insbesondere bei der Software-Vorstellung kann noch alles ganz anders gestaltet werden. Das Folgende ist also nur als Diskussionsgrundlage zu werten.

So könnte die Hardware aussehen:

Das Terminal ist ein Z80-Computer mit 32K Arbeitsspeicher und 32K Hercules-Video-RAM. Nach dem Einschalten besteht der Arbeitsspeicher beim Lesen aus EPROM und beim Schreiben aus RAM. Durch ein LDIR wird der EPROM-Inhalt bei der Initialisierung ins RAM kopiert. Danach wird das EPROM abgeschaltet und beim Lesen und Schreiben in den unteren 32K erreicht der Z80 das RAM.

Die oberen 32K des Z80 sind dem Video-RAM der Hercules vorbehalten. Hier können die beiden 32K HRG-Pages der Hercules bzw. das Text-/ Attribut-Video-RAM eingeblendet werden. Im Textmodus bietet die Hercules-Karte 8 linear hintereinander liegende Seiten mit je 4K Text-/Attribut-Speicher.

Im I/O-Adreßraum des Terminal-Z80 sind die Hercules-internen Ports, das Interface zum Host-Computer, ein Latch zur Einstellung der Konfiguration des Terminals und ein Tastatur-Interface unterzubringen. Zur Vereinfachung der Hardware scheinen folgende Portadressen die sinnvollsten zu sein:

```
  B0H-BFH   interne Ports der Hercules-Karte
               (siehe Artikel zur Hercules im GENIE)
  00H       Status-Register des Interface zum Host-Computer
  01H       Daten-Register des Interface zum Host-Computer
  02H       Konfigurations-Latch
  03H       Tastatur
Das Konfigurations-Latch unterstützt bislang folgende Befehle:
  OUT  (02H),0  Einschalten des Text-/Attribut-Speichers im Textmodus
                bzw. der HRG-Seite 0 im Grafik-Modus
  OUT  (02H),1  Einschalten der HRG-Seite 1; im Text-Modus ist der
                Text-Speicher nicht verfügbar
  OUT  (02H),2  im Bereich 0-32K ist bei Leseoperationen das EPROM
                eingeblendet
  OUT  (02H),3  sowohl beim Lesen als auch beim Schreiben liegt im Be-
                reich 0-32K RAM vor
  OUT  (02H),5  und danach
  OUT  (02H),4  Zurücksetzen des Interface der PC-XT-Tastatur
```

Die Ausgabe der Werte 6-15 kann für weitere Konfigurations-Einstellungen benutzt werden. D0 gibt dabei jeweils den Pegel einer entsprechenden Steuerleitung wieder (Q3-Q7 des 74LS259-Latch) und D1-D3 geben vor, welche Steuerleitung geschaltet werden soll.

Die Portadressen, unter denen das Terminal für den Host-Computer erreichbar sein soll, lassen sich über Jumper in Schritten von 4 beliebig einstellen. Die Basisadresse X+0 ist das Status-Register. D0 des Status-Registers zeigt an, ob das Terminal ein weiteres Byte annehmen kann:

```
  D0=0 :  das Terminal kann ein weiteres Byte annehmen
    =1 :  es darf noch kein weiteres Byte an das Terminal ausgegeben
          werden
D7 gibt an, ob das Terminal bei der Ausgabe von Daten an den Host ein
neues Byte bereitgestellt hat:
  D7=0 :  es steht kein neues Byte des Terminals bereit
    =1 :  das Terminal hat ein neues Byte ausgegeben
Je nach Zustand der Bits 0 und 7 des Status-Ports kann die CPU an den
Daten-Port X+1 ein weiters Byte ausgeben bzw. von dort ein weiteres
Byte abholen.
Für den Z80 des Terminals liegen die Adressen der Ports des Interface
fest (00H = Status-Register; 01H = Daten-Register). Durch Lesen des
Satus-Registers kann der Terminal-Z80 feststellen, ob er Daten entge-
gennehmen kann/muß bzw. ausgeben kann/muß:
  D0=0 :  es kann/muß ein weiteres Byte an den Host ausgegeben werden
          (der Host hat das letzte Byte abgeholt)
    =1 :  es darf noch kein weiteres Byte an den Host ausgegeben wer-
          den
  D7=0 :  es liegt kein neues Byte vom Host vor
    =1 :  der Host hat ein neues Byte ausgegeben, das abzuholen ist

Zur Vereinfachung der Kommunikation zwischen Host und Terminal wird
am Terminal-Z80 ein Interrupt erzeugt, wenn der Host ein neues Datum
ausgegeben hat.
```

Als Tastatur kann wahlweise eine 8Bit-Parallel-ASCII-Tastatur oder eine serielle PC-XT-Tastatur verwendet werden. Beide Interfaces sind bei der bislang bestehenden Prototypen-Karte vorhanden und können durch Jumper selektiert werden. Wenn von der Tastatur ein Byte gekommen ist, wird am Terminal-Z80 ein /NMI ausgelöst.

Und das sollte die Software unterstützen:

```
In allen vorgesehenen Betriebsmodi des Terminals sind Umschaltbefehle
möglich, die in einen jeweils anderen Modus umschalten oder sonstige
Spezialaufgaben erledigen:
  1)  Einschalten des TVI 950 Textmodus (Grundeinstellung)
  2)  Einschalten des Tektronix 4014 Grafikmodus
  3)  Umschalten zwischen Local Mode und OnLine-Betrieb
  4)  Übernehmen eines Maschinenprogramms vom Host;
      Der Befehl enthält als Parameter Ladeadresse, Länge und Call-Adr.
  5)  Umschalten zwischen Video- und Drucker-Ausgabe
  6)  Übernahme eines Initialisierungsstrings und der Bit-Grafik-Ein-
      leitung für den jeweils an der Hercules angeschlossenen Drucker
  7)  Hardcopy des Bildschirmes (sowohl Text als auch Grafik, was
      gerade eingestellt ist)
```

Die ersten 3 Punkte machen schon ein recht ansehnliches Terminal aus. Die Übernahme und optionale Ausführung von Maschinenprogrammen hat hauptsächlich in der Entwicklungsphase der Software ihre Bedeutung. Außerdem erlaubt sie auch später noch den Mißbrauch der Z80-Power des Terminals für Anwenderzwecke.

Der Punkt 5 erschlägt mit dem Terminal auch gleich eine Drucker-Schnittstelle. Die Punkte 6 und 7 drängen sich bei der gemeinsamen Lage von Video- und Drucker-Interface geradezu auf.

Als Grundstock an Treiberroutinen müssen die Initialisierung des Textbildschirms und des Grafikbildschirms sowie ein Tastatur-Treiber vorhanden sein. Für die beiden Video-Modi muß je ein Interpreter für die im jeweiligen Modus gültigen Control-Codes und Escape-Sequenzen existieren, der auch immer die oben angeführten globalen Umschaltbefehle mit berücksichtigt.

Die Befehle des Terminals Grip 5.x

Um irgendwo anzufangen, habe ich mir die Leistungen der GRIP 5.X von Conitec als Vorlage für das Hercules-Terminal herangezogen. Dort werden ebenfalls ein TVI 950 Textterminal und ein Tektronix 4014 Grafikterminal unterstützt, eine Tastatur und ein Drucker angeschlossen und ähnliche Zusatzleistungen bereitgestellt.

Die mit dem Hercules-Terminal möglichen Befehle der GRIP seien im Folgenden mit knapper Beschreibung zusammengestellt. Alle Features der GRIP, die sich bei derselben als unschön erwiesen haben, sind hier garnicht erst aufgeführt.

```
ESC ESC  'A'    1BH 1BH 41H     TVI 950 Textmodus einschalten
                                (Einstellung nach RESET)
TVI 950 Steuercodes:
^H      08H     Cursor nach links
^I      09H     Cursor nach rechts (mit Scroll)
^K      0BH     Cursor eine Zeile hoch
^V      16H     Cursor eine Zeile runter (mit Scroll)
^J      0AH     Line Feed (wie ^V)
^M      0DH     Carriage Return, Cursor an den Zeilenanfang
^_      0FH     Neue Zeile (= Carriage Return + Line Feed)
^^      1EH     Cursor Home (Zeile 0, Spalte 0)  ohne ClrScr
^Z      1AH     Bildschirm löschen
^L      0CH     Form Feed (:= Clear Screen + Cursor Home)
^G      07H     Rabatz auslösen (muß außerhalb des Terminals
                erzeugt werden)
TVI 950 Escape-Sequenzen;
ESC '+'         Form Feed (wie ^L)
ESC ';'         Form Feed (wie ^L)
ESC '.' a       Cursor Attribut setzen
        a = '0' unsichtbarer Cursor
            '1' Blinkender Block  (nach Reset)
            '2' stehender Block
            '3' blinkender Unterstrich
            '4' stehender Unterstrich
            '5' '6' '7' diverse User Cursor
ESC '=' y x     Cursorposition setzen
ESC '?'         Corsorposition abfragen
                Antwort:  y  x
        y       Zeilennummer + 20H
        x       Spaltennummer + 20H
                bei Antwort auf die Abfrage ist in (y, x, cr)
                das Bit 7 gesetzt
ESC '1'         Bildschirm um 1 Zeile hochscrollen
ESC '2'         Bildschirm um 1 Spalte nach rechts scrollen
ESC '3'         Bildschirm um 1 Zeile runterscrollen
ESC '4'         Bildschirm um 1 Spalte nach links scrollen
        der Cursor bewegt sich mit dem Text bis zum Rand
ESC 'E'         Leerzeile an der Cursorposition einfügen
ESC 'R'         Zeile an Cursorposition löschen
        die nachfolgenden Zeilen werden entsprechend bewegt
ESC 'T'         Zeile ab Cursor bis Ende löschen
ESC 'Y'         Zeile ab Cursor bis Ende löschen
ESC 'G' x       Zeichenattribute setzen
        x: D0=1: Zeichen unsichtbar             möglich
           D1=1: Zeichen durchgestrichen        nicht möglich
           D2=1: Zeichen invers dargestellt     möglich
           D3=1: Zeichen unterstichen           nicht möglich
           D4,D5 = 0,0: Subscript               nicht möglich
                   0,1: Indexschrift            nicht möglich
                   1,0: Superscript             nicht möglich
                   1,1: normale Schrift default möglich
```

```
        D6=1:   Breitschrift                    nicht möglich
        D7      ist immer low auszugeben
ESC ')'         Invertieren einschalten
ESC '('         Invertieren ausschalten
ESC 'n'         Bildschirm eingeschaltet
ESC 'o'         Bildschirm dunkel
ESC '$'         Strichgraphik ein       Multiplan-kompatibel
ESC '%'         Strichgraphik aus       Zeichen anstelle der
                                        Großbuchstaben
ESC ESC 'T'     Tektronix-Grundmodus einschalten
```

Tektronix-Control-Codes

```
^M      0DH     Alphamodus ein, Cursor an den Anfang der Zeile,
                Zeile der letzten Grafikposition setzen
^_      1FH     Alphamodus ein, Inkrementalmodus aus Curdor auf
                die letzten Grafik-Koordinaten setzen
^Ö      1CH     Punktmodus aus
^U      1DH     Vektormodus ein; der erste Vektor wird
                nicht gezeichnet, die Koordinaten werden
                als neuer Anfangspunkt übernommen
^G      07H     Sonderfunktion (nach ^U): der Vektor wird
                doch gezeichnet
^^      1EH     Inkrementalmodus ein
^Q      11H     lösche mit Schwarz (Hintergrundfarbe)
^R      12H     zeichne invers (Vordergrund schwarz<=>weiß)
^S      13H     zeichne mit weiß(=> Vordergrundfarbe)
                die letzten drei Steuercodes sind nicht Bestandteil
                des Tektronix-Befehlssatzes sondern Erweiterungen
                der GRIP
```

Tektronix-ESC-Sequenzen

```
ESC '^L'        Clear Screen, Alphamodus ein, Cursor
ESC '1'         Bildschirm um 1 Zeile hochscrollen
E5c '2'         Bildschirm um 1 Spalte nach rechts scrollen
ESC '3'         Bildschirm um 1 Zeile runterscrollen
ESC '4'         Bildschirm um 1 Spalte nach links scrollen
        die letzte Grafik-Koordinate scrollt mit, die Posi-
        tion relativ zum Nullpunkt ändert sich also
ESC 'G' X       Zeichenattribute für Alpamodus setzen
                Attribute sind die gleichen wie bei TVI 950
ESC '~'         Vektortyp := durchgehende Linie (default)
ESC 'a'         Vektortyp := gepunktete Linie
ESC 'b'         Vektortyp := Strichpunk Linie
ESC 'c'         Vektortyp := kurz gestichelte Linie
ESC 'd'         Vektortyp := lang gestichelte Linie
ESC 'F'         Fülle Fläche, in der der Grafik-Cursor steht
                mit Muster des Zeichens 7FH
```

Der Vektormodus (mit ^U einschalten)

```
Bei der Hercules beträgt die Grafikauflösung 720 (x) * 348 (y) Punkte.
Der Ursprung (0,0) liegt unten links. Eine Koordinate wird durch 2
10Bit-Werte angegeben, die als 4 Bytes mit 5 signifikanten Bits
(D4-D0) an das Terminal übertragen werden. Welche Funktion das jeweil-
ige Byte hat, wird durch D5-D7 codiert:
        0,0,1   obere 3 Bit der Y-Koordinate
        0,1,1   untere 3 Bit der Y-Koordinate
        0,0,1   obere 3 Bit der X-Koordinate
        0,1,0   untere 3 Bit der X-Koordinate
Die Übertragung erfolgt in der Reihenfolge: Yhigh, Ylow, Xhigh, Xlow
Folgende Formate zur Datenübergabe sin erlaubt:
        Yhigh,Xlow              für die nicht übergebene Anteile
        Yhigh,Xhigh,Xlow        der Koordinaten werden die zuletzt
        Ylow,Xlow               übergebenen Werte wieder verwandt
        Xlow
Mit dem Empfang des Xlow-Bytes ist die Übertragung eines Koordinaten-
Wertepaares abgeschlossen, das Terminal zeichnet dann einen Vektor
vom letzten Koordinatenpunkt zum neu angegebenen Punkt.
```

Der Inkrementalmodus

```
zeichnet von der letzten Grafik-Koordinate in 1-Punkt-Schritten
weiter, die Werte (X,Y) des Grafik-Cursors laufen mit. Die Richtung
wird durch die folgenden Zeichen vorgegeben:
        F    D    E             'F' nach oben links
                                'D' nach oben
        B    *    A             'E' nach oben rechts
```

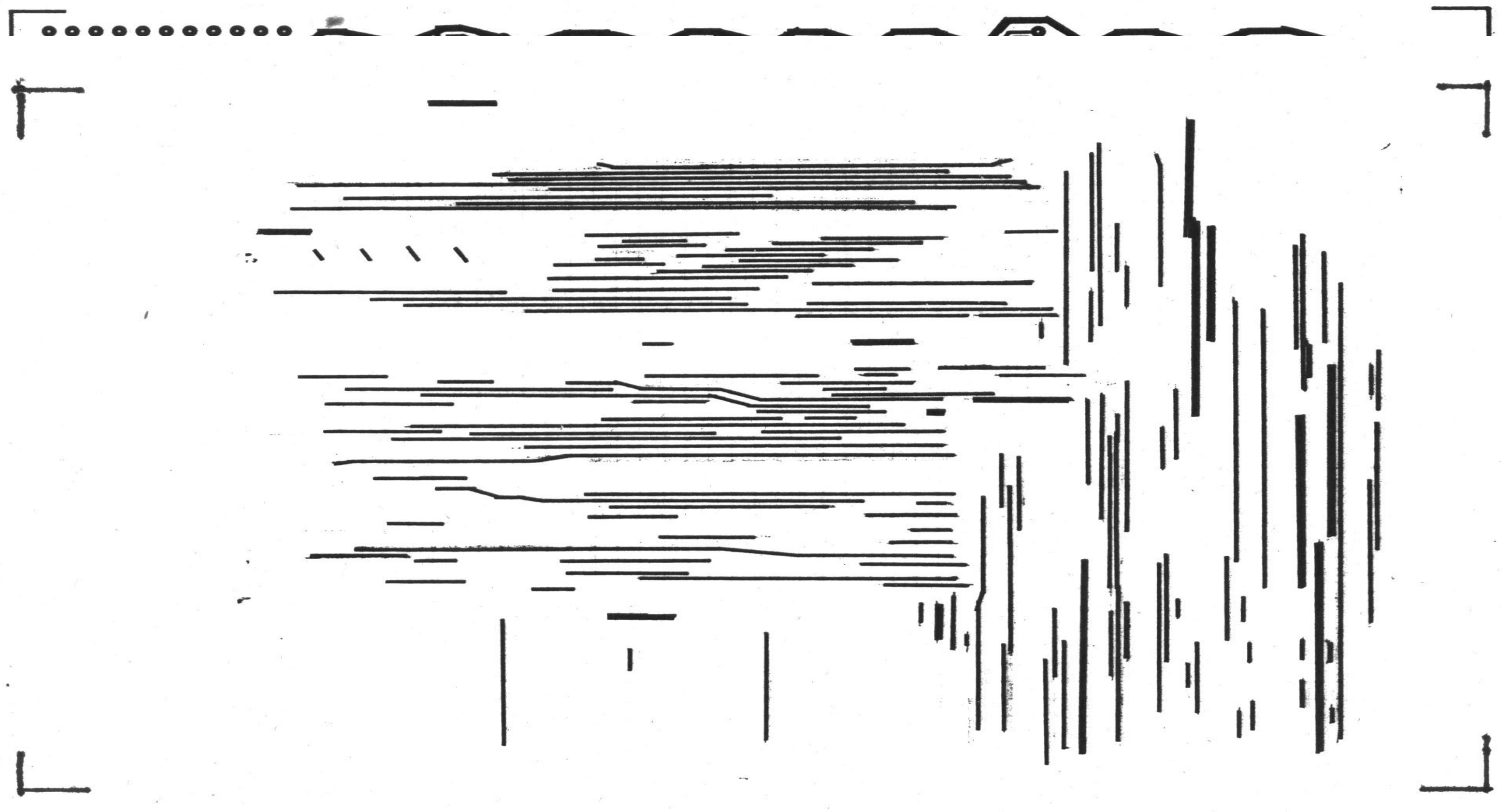
```
                              'B' nach links
        J    H    I           'A' nach rechts   u.s.w.
Durch 'P' 'Q' 'R' 'T' werden die Zeichenmodi eingestellt. Ein
Modus ist so lange wirksam, bis ein neuer eingeschaltet wird. Nach
dem Einschalten des Inkrementalmodus muß einer dieser Befehle folgen:
' '     Punkte nicht zeichnen, nur Grafik-Cursor bewegen
'P'     Punkte zeichnen (weiß, wie ^S)
'Q'     Punkte löschen (schwarz, wie ^Q)
'R'     Punkte invertieren (wie ^R)
```

1990-Club80-34, S.25-34

Club-80 Terminal auf Basis der Hercules-Karte des PC

Helmut Bernhardt, Stefan Nitschke 22.05.91

Nun ist es schon wieder 1 Jahr her, daß auf dem Club-Treffen 1990 der Bau einer Terminalkarte, die sich des günstigen Leistungs/PreisVerhältnisses der Herculeskarte für den PC bedient, beschlossen wurde. Nachdem ein erfolgsversprechendes Konzept stand, der Hardware-Koordinator noch schriftlich um Stellungnahme gebeten wurde (die bislang immer noch nicht vorliegt), auf diese Stellungnahme artig ein paar Wochen gewartet wurde, haben wir das damalige Konzept an Jens Neueder abgeschickt. Wahrscheinlich erscheint es zusammen mit diesen Beiträgen irgendwann in den nächsten Jahren mal in einem Info.



oben: Layout (von der Bestückungsseite gesehen)
unten: Drahtbrücken (von der Lötseite gesehen)

Da wir nicht warten wollten, bis eventuell mal ein Info erscheint und dann eventuell doch noch mehr Mitstreiter bei dem Projekt helfen, haben wir schon mal angefangen und eine lauffähige erste Version fertiggestellt. An dem ursprünglichen Hardwarekonzept hat sich inzwischen nichts geändert und auch das, was als Software bislang läuft, entspricht dem TVI950-Standard (mit den paar zusätzlichen und sehr nützlichen Features wie z.B. Program-Download).

Da bleibt noch die Einbindung der Tektronix 4014 Grafikbefehle, des Druckertreibers, eines Hardcopy-Befehls und einiger zusätzlicher Befehle (u.a. Snapshot: Abspeichern des Bildschirminhalts auf Disk). Diese recht umfangreichen Aufgaben stehen immer noch an.

Zusätzlich haben wir uns überlegt, daß man die 8 Text-Pages der Herculeskarte dazu benutzen kann, um damit Window-Technik zu realisieren. Durch einen Befehl werden dem Terminal die Kordinaten des oberen linken Punktes sowie die Anzahl Zeilen und Spalten im Fenster mitgeteilt. Das Terminal kopiert daraufhin die derzeitige Textseite in die nächste, löscht darin den angegebenen Fensterbereich und positioniert den Cursor in der oberen linken Ecke des neuen Fensters. Der Befehl 'Fensterlöschen' schaltet einfach wieder zurück auf die vorherige Textseite. Nötige Statusinformationen werden in Window Control Blocks für jede Fensterebene geführt, so daß nach dem Entfernen eines Fensters der vorherige Bildschirm mit allen Einstellungen (Attribut, Cursor) wieder restauriert wird. Das ist aber zur Zeit noch im Entstehen.

Das Terminal ist ein Z80-Computer mit 32k Arbeitsspeicher und 32K Hercules-Video-RAM. Nach dem Einschalten besteht der Arbeitsspeicher beim Lesen aus EPROM und beim Schreiben aus RAM. Durch ein LDIR wird der EPROM-Inhalt bei der Initialisierung ins RAM kopiert. Danach wird das EPROM abgeschaltet und beim Lesen und Schreiben in den unteren 32K erreicht der Z80 das RAM.

Die oberen 32K des Z80 sind dem Video-RAM der Hercules vorbehalten. Hier können die beiden 32K HRG-Pages der Hercules bzw. das Text-/Attribut-Video-RAM eingeblendet werden. Im Textmodus bietet die Hercules-Karte 8 linear hintereinander liegende Seiten mit je 4K Text-/Attribut-Speicher. Innerhalb einer Textseite stehen die gradzahligen Adressen als Textspeicher und

die ungradzahligen Adressen als Speicher für die zugehörigen Attribute zur Verfügung. Jeweils zwei Speicheradressen stehen für eine Druckposition auf dem Bildschirm.

Die Bits eines Attributbytes haben folgende Funktion:
(siehe D5 des Display Mode Control Ports B8h)

D7 = 0	nicht blinkend bzw. normaler Hintergrund
D7 = 1	blinkend bzw. halbheller Hintergrund
D6-D4 = 0, D2-D0 = 1	normale Darstellung
D6-D4 = 1, D2-D0 = 0	inverse Darstellung

Das gilt aus der Sicht des Z80, der den Speicher der Hercules mit Zeichen und Attributen zu befüllen hat. Der CRTC6845 auf der Hercules hat während der Darstellung eines Zeichens (bzw. der Dotzeile eines Zeichens) gleichzeitig das Zeichen und Attribut auszulesen und während des Shiftens der Bits zum Videosignal die Informationen des Attributs auf das Videosignal wirken zu lassen. Für den 6845 ist eine Textseite in 2K Worten organisiert - das ist bei der Berechnung des Inhalts des Scan-Offset-Registers des 6845 zu berücksichtigen.

Im I/O-Adreßraum des Terminal-Z80 sind die Hercules-internen Ports, das Interface zum Host-Computer, ein Latch zur Einstellung der Konfiguration des Terminals und ein Tastatur-Interface unterzubringen; als da wären:

00H	Status-Register des Interface zum Host-Computer	
01H	Daten-Register des Interface zum Host-Computer	
02H	Konfigurations-Latch	
03H	Tastatur-Datenregister	
B0H-BFH	interne Ports der Hercules-Karte	
B4H	Adreßregister des CRTC-6845	
B5H	Datenregister des CRTC-6845	
B8H	Display-Mode-Control-Port der Hercules	
	D1 = 0	Text-Video-Mode enabled, HRG disabled
	D1 = 1	HRG enabled, Text-Video disabled
	D3 = 0	die Hercules gibt kein Video-Signal an den Monitor
	D3 = 1	das Video-Signal der Karte gelangt zum Monitor
	D5 = 0	D7=1 des Attribut-Bytes bedeutet "heller Hintergrund"
	D5 = 1	D7=1 des Attribut-Bytes bedeutet "Blinken"
	D7 = 0	HRG: Page 0 selectet
	D7 = 1	HRG: Page 1 selected

B9H	Display-Status-Kanal (nach mc)	
BAH	Display-Statusport (nach Computer 2000)	
	D0 = 1	horizontaler Zeilenrücklauf
	D3 = 1	Video-Echo, Signal zum Monitor
	D7 = 0	vertikaler Strahlrücklauf
BBH	Light-Pen Flip Flop zurücksetzen (wird kaum noch in Hardware ermöglicht)	
BCH	Drucker-Datenport (Schreiben)	
BDH	Drucker Statusport (Lesen)	
	D3 = 0	Error
	D4 = 1	Selected & Ready
	D6 = 0	(für 5 Microsecs) /ACK low aktiv
	D7 = 1	Ready high aktiv
BEH	Drucker-Steuerkanal	
	D0 = 0	/STROBE, mindestens 5 Microsecs
	D1 = 1	Linefeed
	D2 = 0	Reset Printer, mindestens 50 Microsecs
	D3 = 1	Select, Drucker freigeben
	D4 = 1	IRQ7 (high aktiver Interrupt) freigeben, wenn der Drucker ein Byte übernommen hat
BFH	Konfigurations-Schalter	
	D0 = 1	HRG erlaubt, gilt nur für Page 0
	D1 = 1	auch HRG-Page 1 erlaubt

Das Konfigurations-Latch (Port 02h) unterstützt bislang die Befehle:

OUT (02H),0	Einschalten des Text-/Attribut-Speiches im Texmodus bzw. der HRG-Seite 0 im Grafik-Modus
OUT (02H),1	Einschalten der HRG-Seite 1; im Text-Modus ist der Text-Speicher nicht verfügbar
OUT (02H),2	im Bereich 0-32K ist bei Leseoperationen das EPROM eingeblendet
OUT (02H),3	Sowohl beim Lesen als auch beim Schreiben liegt im Bereich 0-32K RAM vor

OUT (02H),5 und danach...

OUT (02H),4 Zurücksetzen des Interface der PC-Tastatur

Die Ausgabe der Werte 6-15 kann für weitere Konfigurations-Einstellungen benutzt werden. D0 gibt dabei jeweils den Pegel einer entsprechenden Steuerleitung wieder (Q3-Q7 des 74LS259-Latch) und D1-D3 geben vor, welche Steuerleitung geschaltet werden soll. Auf der Karte ist noch ein Lochrasterfeld frei, in dem entsprechende Hardware-Erweiterungen untergebracht werden können. U. a. erscheint es zweckmäßig zu sein, Rabatz-I/O-Hardware unterzubringen, die über Q3 (mit den Steuercodes 06H/07H) zu löblichem Tun angeregt wird.

Zur Vereinfachung der Kommunikation zwischen Host und Terminal wird am Terminal-Z80 ein Interrupt erzeugt, wenn der Host ein neues Datum ausgegeben hat. Diese Technik erfordert in der Software-Entwicklung zwar etwas mehr Sorgfalt, sie ermöglicht aber auch, daß der Z80 des Terminals bei Konjunkturflauten mit anderen sinvollen Aufgaben beschäftigt werden kann und zwischendurch mal eben ein Byte vom Host übernehmen und ins Fernsehen schicken kann.

Als Tastatur kann wahlweise eine 8Bit-ASCII-Paralleltastatur oder eine serielle PC-Tastatur verwendet werden. Für den Anschluß einer 8Bit-ASCII-Paralleltastatur besteht ein 74LS244 Lesetreiber. Der /STROBE-Eingang der Tastatur ist direkt an den /NMI-Pin des Z80 angeschlossen. Die Übernahme von Tastaturdaten erfolgt in Echtzeit im Hintergrund. Alle Tastaturdaten werden in einem 32Byte-Ringpuffer zwischengespeichert. Der Ringpuffer wird in der Hauptprogrammschleife ständig überwacht und bei Bedarf werden die Tastatureingaben an den Host entsorgt. Danach ist gewährleistet, daß keine Tastatureingaben verlorengehen, und daß bei der Übertragung sonstiger Daten (z.B. Bildschirmdump) an den Host nicht Tastaturdaten dazwischengeraten. Während solcher Jobs befindet sich der Z80 nicht in der Hauptprogrammschleife.

Alternativ kann auch eine PC-Tastatur angeschlossen werden. Diese Teile geben keine ASCII-Daten sondern Scancodes aus. Solch ein Scancode wird gesendet, wenn eine Taste gedrückt und auch wenn eine Taste losgelassen wird. Make- und Break-Codes einer Taste unterscheiden sich dabei nur durch D7 des übertragenen Bytes.

Die Codes der PC-Tastatur werden seriell mit TTL-Pegeln übertragen. Zur Synchronisation liefert die Tastatur auch ein CLK-Signal. Auf der Terminal-Karte besteht ein Seriell/Parallel-Wandler, der selbständig die 8Bit der Tastatur sammelt und nach dem achten Bit einen /NMI erzeugt. Der Z80 kann dann über den Port 03H den Scancode der Tastatur als 8Bit-Wort auslesen und durch aufeinanderfolgende Ausgabe der Werte 4 und 5 an Port 02H den Seriell/Parallel-Wandler für die nächste Tastatureingabe zurücksetzen.

Die /NMI-Serviceroutine führt Buch über den Zustand der Tasten mit Umschaltfunktionen und erzeugt mit diesen Informationen aus den Scancodes der normalen Tasten ASCII-Codes, die im Ringpuffer abgelegt werden.

Wenn eine PC-Tastatur angeschlossen wird, braucht der 74LS244 nicht bestückt zu werden, und wenn eine Paralleltastatur zum Einsatz kommt, ist kein 74LS322 nötig. Durch den Jumper JP2 wird vorgegeben, welcher Baustein aktiv sein soll.

1-2 gebrückt: Interface für eine PC-Tastatur2-3 gebrückt: Interface für eine Paralleltastatur
JP2

Bei Anschluß der PC-Tastatur muß außerdem JP3 gesteckt werden, damit am Z80 ein /NMI ankommen kann.

Das Terminal ist mit dem Host über eine bidirektionale parallele Schnittstelle verbunden. Für den Z80 des Terminals liegen die Adressen des Ports des Interface fest (00H = Statusregister; 01H = Datenregister) Durch Lesen des Statusregisters kann der Terminal-Z80 feststellen, ob er Daten entgegennehmen kann/muß bzw. ausgeben kann bzw. muß:

D0 = 0	es kann/muß ein weiteres Byte an den Host ausgegeben werden (der Host hat das letzte Byte abgeholt)
D0 = 1	es darf noch kein weiteres Byte an den Host ausgegeben werden.
D7 = 0	es liegt kein neues Byte vom Host vor
D7 = 1	der Host hat ein neues Byte ausgegeben, das abgeholt werden muß

Die Portadressen, unter denen das Terminal für den Host-Computer erreichbar sein soll, lassen sich über Jumper in Schritten von 4 beliebig einstellen. Die Basisadresse X+0 ist das Statusregister.

D6 des Statusregisters zeigt an, ob das Terminal ein weiteres Byte annehmen kann:

D6 = 1 das Terminal kann ein weiteres Byte annehmen

D6 = 0 es darf noch kein weiteres Byte an das Terminal ausgegeben werden

D7 gibt an, ob das Terminal bei der Ausgabe von Daten an den Host ein neues Byte bereitgestellt hat:

D7 = 1 es steht kein neues Byte des Terminals bereit

D7 = 0 das Terminal hat ein neues Byte ausgegeben

Je nach Zustand der Bits 6 und 7 des Statusports kann der Host an den Datenport X+1 ein weiteres Byte ausgeben bzw. von dort ein weiteres Byte abholen.

Über das Jumperfeld JP1 wird die Portadresse des Terminals eingestellt. Die Jumper haben folgende Funktion:

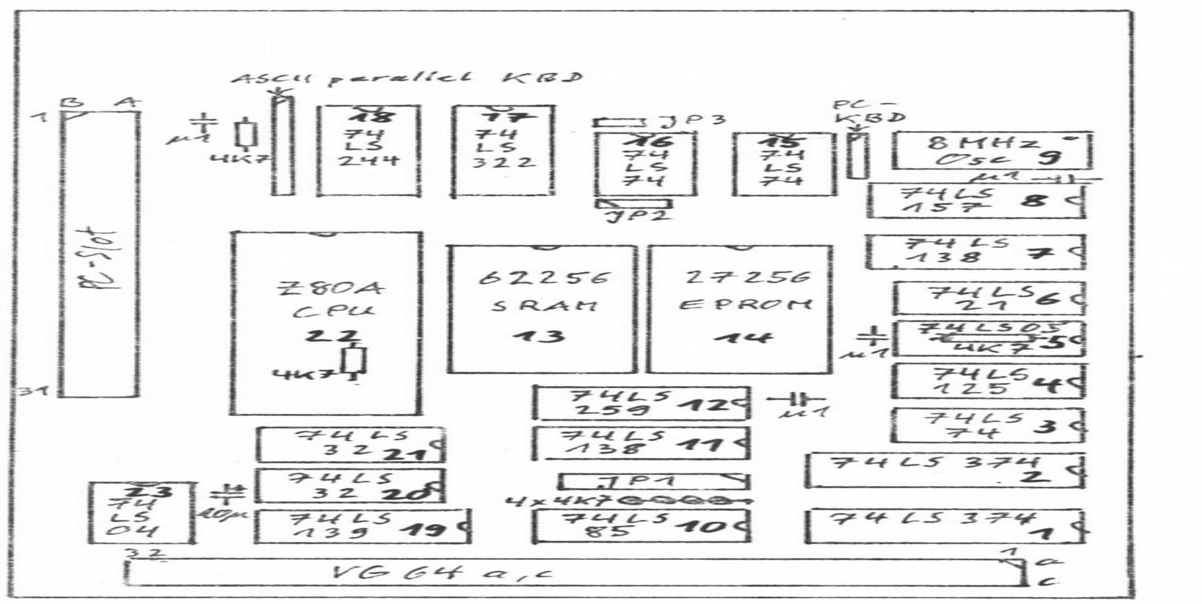
1 - 2	offen:	A7 der Portadresse ist high
	gesteckt:	A7 der Portadresse ist low
3 - 4	offen:	A6 der Portadresse ist high
	gesteckt:	A6 der Portadresse ist low
5 - 6	offen:	A5 der Portadresse ist high
	gesteckt:	A5 der Portadresse ist low
7 - 8	offen:	A4 der Portadresse ist high
	gesteckt:	A4 der Portadresse ist low
11 - 9	gesteckt, 11 -13 offen	A3 der Portadresse ist high
11 -13	gesteckt, 11 - 9 offen:	A3 der Portadresse ist low
12 -10	gesteckt, 12 -14 offen:	A2 der Portadresse ist high
12 -14	gesteckt, 12 -10 offen:	A2 der Portadresse ist low

A1 wird nicht decodiert; die Ports X+0 und X+1 sind in den Adressen x+2 und X+3 gespiegelt.

Um kompatibel zur GRIP von Conitec zu sein, müssen die Portadressen C0h und C1h decodiert werden, dafür sind folgende Jumper zu setzen: 5-6, 7-8, 11-13, 12-14.

Der Aufbau der Karte

Bestückungs-Plan



ist wie bei allen Bastlerlösungen etwas komplizierter als bei industriell gefertigten, durchkontaktierten Karten. Wer schon mal eine doppelseitig geätzte Karte selbst durchkontaktieren mußte, weiß ein Lied davon zu singen. Da kann es durchaus einfacher sein, auf einem einseitigen Board die nötigen Querverbindungen in Drahtbrücken anzulegen. Bei dieser Karte ist beides möglich (auch gemischt). Entweder verwendet man die Leiterbahnen auf der Bestückungsseite direkt und legt die nötigen Durchkontaktierungen an, oder man benutzt sie nur als Hinweise dafür, wo Drahtbrücken gezogen werden müssen. Das Layout der Bestückungsseite weist keine sich überkreuzenden Drahtbrücken auf (was im Layout auch schlecht ohne Kurzschlüsse möglich ist); wo die Leiterbahnen nicht zu eng beieinander liegen, kann auch unisoloierter Draht verwendet werden. Bei den etwas dicker geätzten Leiterbahnen sollte als Brücke Klingeldraht und bei den dünnen Leiterbahnen dünner Kupferlackdraht verwendet werden (wenn man lieber Drahtbrücken legen möchte).

Erweiterung des ECB-Interface des Hercules-Terminals auf 4 Portadressen für die gesonderte Druckerausgabe

Für die gleichzeitige Ausgabe von Daten für die Console und den Drucker über das gleiche Interface ohne besondere Unterscheidungsmöglichkeiten des Terminals anhand irgendwelcher Statusbits lassen sich grundsätzlich Probleme erwarten.

Es wäre möglich (wie bei der GRIP) durch ein gesetztes D7 im Datum dem Terminal zu signalisieren, daß dieses Byte für den Drucker gedacht ist. Damit ist aber die Möglichkeit verbaut, einen Drucker mit IBM-Zeichen- satz 2 voll zu nutzen.

Hier soll aus der Not eine Tugend gemacht werden und die unvollständige Decodierung des ECB-Interface auf 4 Portadressen für unterschiedliche Ausgabeports für Console und Drucker genutzt werden. Die Ausgabe an die Console erfolgt nach wie vor über die Adressen

Basis + 0: Consolen-Status (Lesen)
Basis + 1: Console Daten (Lesen und Schreiben)

und der Drucker ist über die Adressen

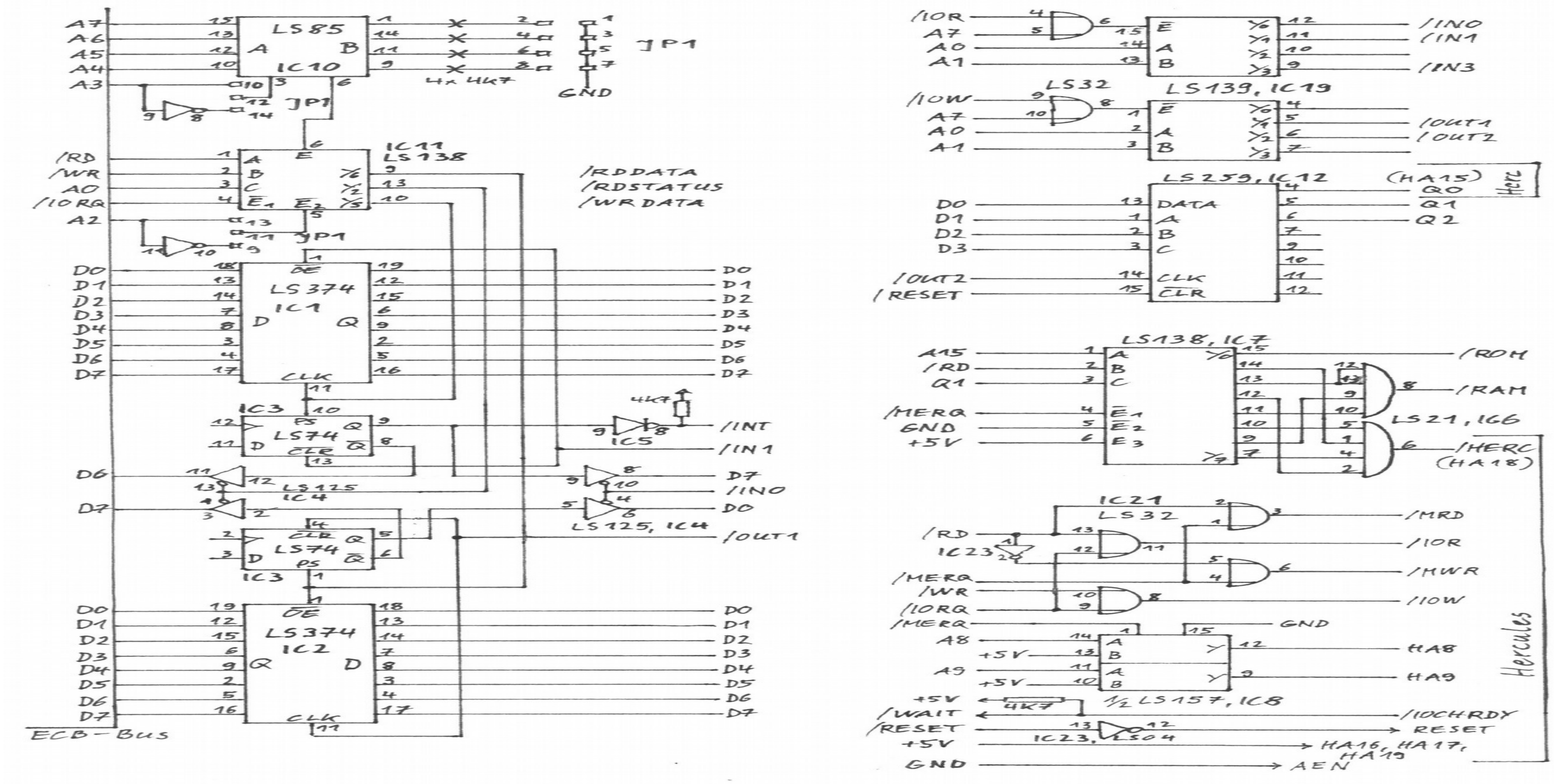
Basis + 2: Drucker-Status D7=0: Ready (Lesen)
Basis + 3: Drucker Daten (Schreiben)

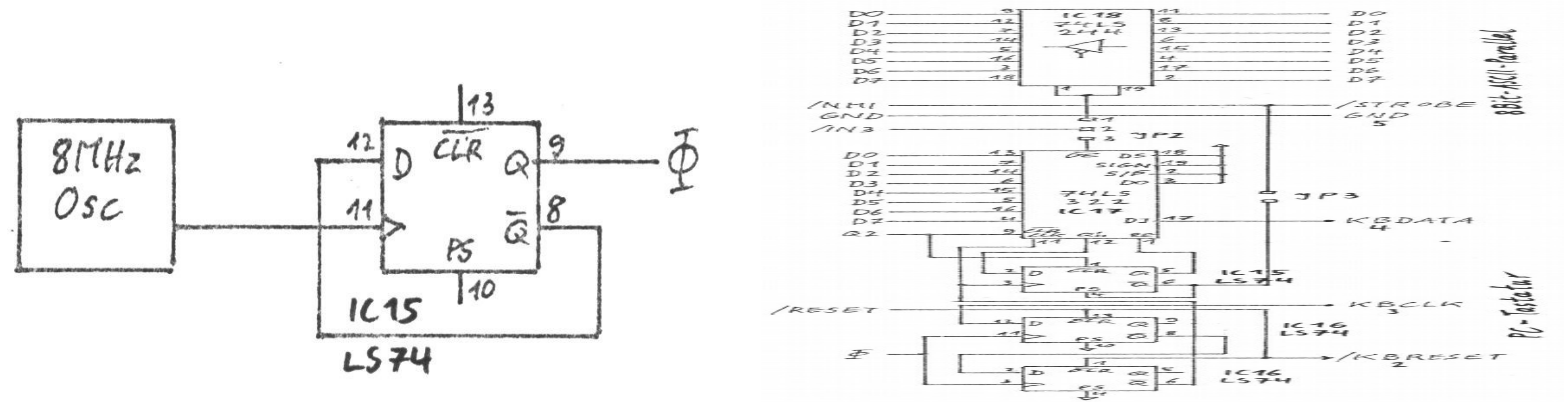
erreichbar. Physikalisch bleiben die Latches für Consolen- und Drucker-Datenport identisch. Das Terminal kann nur anhand von D6 im Statusport des Interface erkennen, ob das Datum an die Console oder an den Drucker ausgegeben wurde.

D6 in Port 00h ist 0: Datum geht an die Console
D6 in Port 00h ist 1: Datum geht an den Drucker

Dafür muß der Pegel von A1 des ECB-Bus bei der Ausgabe des Host an das Terminal in einem Flip Flop gespeichert werden und der Z80 des Terminals muß diesen Wert über einen Treiber auslesen können.

Interface zum ECB-Bus und zur Hercules-Karte





oben: Takterzeugung; rechts: Tastatur-Interface

ist auf IC3, 74LS74 ein weiteres 74LS74 mit den Pins 7 und 14 huckepacks aufzulöten und auf IC4 ist ein weiteres 74LS125 mit den Pins 7(GND), 10(/IN0) und 14(+5V) huckepacks aufzulöten.

Die Pins 9 dieser beiden aufgelöteten ICs sind miteinander zu verbinden.

An Pin 12 des 74LS74 wird A1 des ECB-Bus (Pin 7c) gelegt.
Pin 11 des 74LS74 wird mit Pin 10 von IC11, 74LS138 (/WRDATA) verbunden.
Pin 8 des 74LS125 wird mit Pin 4 von IC2, 74LS374 verbunden (D6).

In Software

ist in das Betriebsprogramm des Terminals in der Interrupt-Service-Routine bei Ausgaben vom Host nach der Ermittlung des Status des ECB-Interface anhand des Pegels von D6 in Port 00h festzustellen, ob das Byte in Port 01h an die Console (D6=0) oder an den Drucker (D6=1) ausgegeben werden soll.

Die Befehle des Terminals Grip 5.x

Um irgendwo anzufangen, habe ich mir die Leistungen der GRIP 5.X von Conitec als Vorlage für das Hercules-Terminal herangezogen. Dort wer- den ebenfalls ein TVI 950 Textterminal und ein Tektronix 4014 Grafik- terminal unterstützt, eine Tastatur und ein Drucker angeschlossen und ähnliche Zusatzleistungen bereitgestellt.

Die mit dem Hercules-Terminal möglichen Befehle der GRIP seien im Folgenden mit knapper Beschreibung zusammengestellt. Alle Features der GRIP, die sich bei derselben als unschön erwiesen haben, sind hier garnicht erst aufgeführt.

```
ESC ESC  'A'    1BH 1BH 41H     TVI 950 Textmodus einschalten
                                (Einstellung nach RESET)
TVI 950 Steuercodes:
^H      08H     Cursor nach links
^I      09H     Cursor nach rechts (mit Scroll)
^K      0BH     Cursor eine Zeile hoch
^V      16H     Cursor eine Zeile runter (mit Scroll)
^J      0AH     Line Feed (wie ^V)
^M      0DH     Carriage Return, Cursor an den Zeilenanfang
^_      0FH     Neue Zeile (= Carriage Return + Line Feed)
^^      1EH     Cursor Home (Zeile 0, Spalte 0)  ohne ClrScr
^Z      1AH     Bildschirm löschen
^L      0CH     Form Feed (:= Clear Screen + Cursor Home)
^G      07H     Rabatz auslösen (muß außerhalb des Terminals
                erzeugt wedden)
TVI 950 Escape-Sequenzen:
ESC '+'         Form Feed (wie ^L)
ESC ';'         Form Feed (wie ^L)
ESC '.' a       Cursor Attribut setzen
        a = '0' unsichtbarer Cursor
            '1' Blinkender Block  (nach Reset)
            '2' stehender Block
            '3' blinkender Unterstrich
            '4' stehender Unterstrich
            '5' '6' '7' diverse User Cursor
ESC '=' y x     Cursorposition setzen
ESC '?'         Corsorposition abfragen
                Antwort:  y  x
        y       Zeilennummer + 20H
        x       Spaltennummer + 20H
                bei Antwort auf die Abfrage ist in (y, x, cr)
                das Bit 7 gesetzt
ESC '1'         Bildschirm um 1 Zeile hochscrollen
ESC '2'         Bildschirm um 1 Spalte nach rechts scrollen
ESC '3'         Bildschirm um 1 Zeile runterscrollen
ESC '4'         Bildschirm um 1 Spalte nach links scrollen
        der Cursor bewegt sich mit dem Text bis zum Rand
ESC 'E'         Leerzeile an der Cursorposition einfügen
ESC 'R'         Zeile an Cursorposition löschen
        die nachfolgenden Zeilen werden entsprechend bewegt
ESC 'T'         Zeile ab Cursor bis Ende löschen
ESC 'Y'         Zeile ab Cursor bis Ende löschen
ESC 'G' x       Zeichenattribute setzen
        x: D0=1: Zeichen unsichtbar             möglich
           D1=1: Zeichen durchgestrichen        nicht möglich
           D2=1: Zeichen invers dargestellt     möglich
           D3=1: Zeichen unterstichen           nicht möglich
           D4,D5 = 0,0: Subscript               nicht möglich
                   0,1: Indexschrift            nicht möglich
                   1,0: Superscript             nicht möglich
```

```
                  1,1: normale Schrift default möglich
        D6=1:   Breitschrift                    nicht möglich
        D7      ist immer low auszugeben
ESC ')'         Invertieren einschalten
ESC '('         Invertieren ausschalten
ESC 'n'         Bildschirm eingeschaltet
ESC 'o'         Bildschirm dunkel
ESC '$'         Strichgraphik ein       Multiplan-kompatibel
ESC '%'         Strichgraphik aus       Zeichen anstelle der
                                        Großbuchstaben
```

ESC ESC 'T'   Tektronix-Grundmodus einschalten

Tektronix-Control-Codes

```
^M      0DH     Alphamodus ein, Cursor an den Anfang der Zeile,
                Zeile der letzten Grafikposition setzen
^_      1FH     Alphamodus ein, Inkrementalmodus aus Curdor auf
                die letzten Grafik-Koordinaten setzen
^Ö      1CH     Punktmodus aus
^U      1DH     Vektormodus ein; der erste Vektor wird
                nicht gezeichnet, die Koordinaten werden
                als neuer Anfangspunkt übernommen
^G      07H     Sonderfunktion (nach ^U): der Vektor wird
                doch gezeichnet
^^      1EH     Inkrementalmodus ein
^Q      11H     lösche mit Schwarz (Hintergrundfarbe)
^R      12H     zeichne invers (Vordergrund schwarz<=>weiß)
^S      13H     zeichne mit weiß(=> Vordergrundfarbe)
                die letzten drei Steuercodes sind nicht Bestandteil
                des Tektronix-Befehlssatzes sondern Erweiterungen
                der GRIP
```

Tektronix-ESC-Sequenzen

```
ESC '^L'        Clear Screen, Alphamodus ein, Cursor
ESC '1'         Bildschirm um 1 Zeile hochscrollen
E5c '2'         Bildschirm um 1 Spalte nach rechts scrollen
ESC '3'         Bildschirm um 1 Zeile runterscrollen
ESC '4'         Bildschirm um 1 Spalte nach links scrollen
        die letzte Grafik-Koordinate scrollt mit, die Posi-
        tion relativ zum Nullpunkt ändert sich also
ESC 'G' X       Zeichenattribute für Alpamodus setzen
                Attribute sind die gleichen wie bei TVI 950
ESC '~'         Vektortyp := durchgehende Linie (default)
ESC 'a'         Vektortyp := gepunktete Linie
ESC 'b'         Vektortyp := Strichpunk Linie
ESC 'c'         Vektortyp := kurz gestichelte Linie
ESC 'd'         Vektortyp := lang gestichelte Linie
ESC 'F'         Fülle Fläche, in der der Grafik-Cursor steht
                mit Muster des Zeichens 7FH
```

Der Vektormodus (mit ^U einschalten)

```
Bei der Hercules beträgt die Grafikauflösung 720 (x) * 348 (y) Punkte.
Der Ursprung (0,0) liegt unten links. Eine Koordinate wird durch 2
10Bit-Werte angegeben, die als 4 Bytes mit 5 signifikanten Bits
(D4-D0) an das Terminal übertragen werden. Welche Funktion das jeweil-
ige Byte hat, wird durch D5-D7 codiert:
        0,0,1   obere 3 Bit der Y-Koordinate
        0,1,1   untere 3 Bit der Y-Koordinate
        0,0,1   obere 3 Bit der X-Koordinate
        0,1,0   untere 3 Bit der X-Koordinate
Die Übertragung erfolgt in der Reihenfolge: Yhigh, Ylow, Xhigh, Xlow
Folgende Formate zur Datenübergabe sind erlaubt:
        Yhigh,Xlow              für die nicht übergebene Anteile
        Yhigh,Xhigh,Xlow        der Koordinaten werden die zuletzt
        Ylow,Xlow               übergebenen Werte wieder verwandt
        Xlow
Mit dem Empfang des Xlow-Bytes ist die Übertragung eines Koordinaten-
Wertepaares abgeschlossen, das Terminal zeichnet dann einen Vektor
vom letzten Koordinatenpunkt zum neu angegebenen Punkt.
```

Der Inkrementalmodus

```
zeichnet von der letzten Grafik-Koordinate in 1-Punkt-Schritten
weiter, die Werte (X,Y) des Grafik-Cursors laufen mit. Die Richtung
wird durch die folgenden Zeichen vorgegeben:
        F    D    E             'F' nach oben links
                                'D' nach oben
```

```
        B    *    A             'E' nach oben rechts
                                'B' nach links
        J    H    I             'A' nach rechts   u.s.w.
Durch 'P' 'Q' 'R' 'T' werden die Zeichenmodi eingestellt. Ein
Modus ist so lange wirksam, bis ein neuer eingeschaltet wird. Nach
dem Einschalten des Inkrementalmodus muß einer dieser Befehle folgen:
' '     Punkte nicht zeichnen, nur Grafik-Cursor bewegen
'P'     Punkte zeichnen (weiß, wie ^S)
'Q'     Punkte löschen (schwarz, wie ^Q)
'R'     Punkte invertieren (wie ^R)
```

1991-Club80-10 Nr.35 S.40-41

Anmerkungen zum Club-Terminal und erste Erfahrungen

Alexander Schmid

Nachdem Helmut noch jede Menge anderer Projekte am laufen und sein Terminal noch nicht ganz fertiggestellt hat, sind Stefan Nitschke und ich bis jetzt leider die einzigen, die in den vollen Genuß dieses Teils kommen (bei dieser Gelegenheit möchte ich Andreas Magnus nochmal herzlich danken, der das Ding auf dem "ausgefallenen" Clubtreffen in stundenlanger Fleißarbeit debuggt hat). Wie schon in einem der letzten Infos passend bemerkt, ist der Aufbau nicht ganz einfach, besser gesagt muß man aufpassen wie ein Schießhund, aber wenn es mal läuft, ist es allererste Sahne und braucht den Vergleich mit kommerziellen Terminals in keinster Weise zu scheuen - Im Gegenteil, ich würde es mit keinem anderen Terminal tauschen, das ich bis jetzt gesehen habe.

Einige der Punkte, die es gegenüber anderen Terminals hervorheben:

Punkt 1: Geschwindigkeit

Helmut hat als Taktfrequenz zwar nur 4 MHz vorgesehen, aber bei mir läuft es ohne Probleme mit 10 MHz und damit bringt es eine maximale Übertragungsrate von knapp 12000 Zeichen (nicht Bit!) pro Sekunde. Mit einem Takt von 12 MHz bin ich zwar schon auf 15000 Zeichen gekommen, aber das wollte ich dem Z80 auf die Dauer nicht zumuten, da ihm dabei etwas sehr warm ums Herz geworden ist.

Zum Vergleich: Bei einer seriellen Übertragung bräuchte man dafür rund 120000 Baud (1 Startbit + 8 Datenbits + 1 Stopbit = 10 Bit Pro Zeichen) und das dürfte man kaum irgendwo finden, 38400 Baud sind meistens das höchste der Gefühle, wenn nicht gar nur 19200 - oder 9600. Wie man im Info nachlesen kann, wird das Hercules-Terminal ja über Ports und somit mit 8 Bit auf einmal betan, daher der Geschwindigkeitsunterschied. Wen es vielleicht wundert, daß die Datenrate nicht noch wesentlich höher ist, dem sei gesagt, daß da die Hercules-Karte ganz gewaltig bremst, weil sie den Z80 zwecks störungsfreier Ausgabe nur während des Zeilenrücklaufs, also, wenn der Elektronenstrahl nicht schreibt, auf den Bildschirm-Speicher zugreifen läßt (Besitzer alter TRASH-80 und Video Genies wissen, was ich meine). Daneben braucht das BIOS natürlich auch seine Zeit, die Zeichen zu verarbeiten und auszugeben.

Punkt 2: Bildschirm und Tastatur

Da das Teil mit einer handelsüblichen Feld-, Wald- und Wiesen Hercules-Karte arbeitet, kann man jeden beliebigen TTL-Monitor anschließen, egal ob grün, orange oder weiß und genauso geht es mit der Tastatur. Die Komponenten sind einzeln austauschbar und man kann das System so zusammenstellen, wie man es gerne hätte. Im Idealfall schmeißt man den PC weg und führt Keyboard und Monitor dem Terminal zu. Austauschtastaturen für Terminals kosten wegen der kleineren Stückzahlen mindestens das Doppelte einer guten PC-Tastatur (300 Mark aufwärts) und der Fernseher liegt bestimmt auch weit jenseits eines Hercules-Monitors, wenn es ihn überhaupt einzeln bzw. als Reparaturteil gibt. Wenn die Hercules selber mal den Geist aufgeben sollte, kommt man mit rund 30 Mark direkt billig weg. Die Freude am Club-Terminal dürfte damit also für lange Zeit gesichert sein.

Die Bildschirmausgabe selber ist über jeden Zweifel erhaben, gestochen scharf und flimmerfrei und die Zeichen haben etwas mehr Körper und sind dadurch angenehmer zu lesen als die übliche Computer-Schrift, die wie bei einigen alten 9-Nadel-Druckern aussieht. Insgesamt stehen 10 Attribute (normal, hell, normal unterstrichen, hell unterstrichen, invers und das alles nochmal blinkend) zur Verfügung, was man auch nur bei den Spitzenmodellen unter den normalen Terminals findet. Zur Textdarstellung kann man aus dem vollen 8-Bit-IBM Zeichensatz mit Semigrafik und allen Sonderzeichen schöpfen, wobei nationale Umcodierungen natürlich untersützt werden.

Punkt 3: Software

Eine faszinierende Funktion des Hercules-Terminals, die ich so oder ähnlich noch nirgends gesehen habe, ist, daß man ihm vom Host-Rechner ein neues Betriebsprogramm schicken kann. Das Terminal ist ja praktisch ein eigenständiger Einplatinencomputer, den man mehr oder weniger beliebig programmieren kann. Die Software steht vollständig als Sourcecode zur Verfügung und wenn einem etwas nicht paßt, ändert man es und schickt es anschließend zum Terminal. So braucht man nicht den ganzen Rechner zu demontieren und keine EPROMs zu brennen, um kleine Updates ausprobieren oder z.B. die Belegung der Funktionstasten zu ändern.

Natürlich gibt es auch gute Terminalkarten, wie etwa die GRIP von conitec, zu kaufen, aber die sind sauteuer und die Software muß man erstmal disassemblieren, bevor man was daran ändern kann (hoffentlich ist es kein Microcontroller...). Bei den richtigen Terminals ist man sowieso auf das angewiesen, was das Ding von Haus aus kann. Das Club-Terminal kann man für rund 100 Mark aufbauen und damit dürfte es im Preis-/Leistungsverhältnis sehr einsam und verlassen weit an der Spitze stehen. Kostenlos bekommt man auch noch eine hochauflösende Grafik mit 720 x 348 Punkten und je nach Hercules-Karte eine parallele Druckerschnittstelle dazu. Wer für etwas weniger Geld und dafür etwas mehr Arbeit eine professionelle Mensch-Maschine-Schnittstelle sucht, für den ist das Club-Terminal schlichtweg die ideale Lösung.

Was das Terminal am Ende alles kann, ist eigentlich nur durch die Phantasie begrenzt, und davon habt ihr sicherlich eine Menge (und traut Euch nur nicht, die Ideen im Info zu veröffentlichen...). Wenn sich jemand also richtig an der Software mit Assembler, Linker und allem Zubehör, austoben will, die hochauflösende Grafik und der Druckertreiber stehen im Moment noch zur freien Verfügung.

Alexander Schmid

2MB-RAM-Erweiterung für Prof 180x

Helmut Bernhardt

Zur Erweiterung des Speichers über 512K auf dem CPU-Board des Prof 180x hinaus hat Conitec die Turbo-RAM vorgesehen. Das 'Turbo' in dieser Bezeichnung ist aber mit Vorsicht zu genießen. Da diese Speichererweiterung auf dem ECB-Bus steckt, liegen zwischen CPU und diesem Speicher 2 Datentreiber. Bei 9,2 MHz ist die Turbo-RAM nur mit einem Wait zu betreiben, was dann auch beim Zugriff auf die ursprünglichen 512K enorm bremst. Hier sei eine Alternative vorgeführt, die ohne Waits eine Erweiterung um 2MB in Form eines Huckepack-Boards auf dem CPU-Board bringt.

Um zu erklären, wie das machbar ist, soll zunächst auf die Verhältnisse eingegangen werden, wie im Prof der Zugriff auf verschiedene Speicher geregelt ist. Da dies alles hauptsächlich im PAL 14L8 gesteuert wird, sei das disassemblierte PAL-Listing hier wiedergegeben:

```
PAL14L8
ausgelesen mit c't PAL-Brenner, 1988 Be
Grundversion nach Conitec

me ioe rd lir busak a18 e mm0 mm1 adr a5 gnd
a1 a7 a19 rom cas0 cas1 fdc gate ioswl dir a19 vcc

/dir    = /ioe * /rd * /adr * az * /a5          ;Richtungssteuerung
        + /me /rd /a19                          ;ECB-Bus-Daten
        + rd * lir * busak                      ;treiber
/iosel  = /ioe * lir * /a18 * /adr * a7 * /a5 * /a2
/fdc    = /ioe * lir * /a18 * /adr * a7 * /a5 * a2
gate    = mm0 * /mm1                            ;Freigabe ECB-Bus-
        + mm1                                   ;-treiber
/a19    = /mm1                                  ;Freigabe für
        + a18 * /mm0                            ;externen Speicher
        + /a18 * mm0
/rom    = /me * /a18 * /rd * /mm0 * /mm1
/cas0   = /me * /a18 * e * rd * /mm0 * /mm1
        + /me * /a18 * e * mm0
/cas1   = /me *  a18 * e * /mm0
        + /me *  a18 * e *  mm0 * /mm1
```

'**adr**' ist ein Eingangssignal für das PAL, in dem A3, A4, und A6 entsprechend den zu erzeugenden I/O-Freigabesignalen im Bereich D8H-DFH decodiert sind. '**e**' ist ein spezielles Taktsignal des HD64180, das hier zum Timing der /cas-Signale verwendet wird. **mm0** und **mm1** sind Ausgänge eines 74LS259-Latches mit einzeln setzbaren Bits. Die übrigen Eingangssignale des Pals sind die normalen Signale der CPU.

Die I/O-mapped Freigabesignale **/iosel** und **/fdc** sind hier uninteressant. Für die Steuerung des ECB-Businterface und der memory-mapped Resourcen werden im PAL die Signale **a19, /gate, /rom, /cas0** und **/cas1** erzeugt. /rom, /cas0 und /cas1 sind low aktive Freigabesignale für das ROM und für je einen 256K-Block RAM auf dem CPU-Board. /gate, das an die Enable-Pins der 74LS245-Bustreiber gelegt ist, steuert die grundsätzliche Möglichkeit der CPU, auf den ECB-Bus zugreifen zu können. Dieses Signal muß high sein, wenn die Prof 180x-Karte im Slave-Modus für sich allein werkeln soll und eine andere CPU-Karte den Bus kontrollieren soll.

/dir steuert die Richtung, in der der 74ALS245-Datentreiber die Daten vom/zum ECB-Bus treibt; /dir = 0 bedeutet, daß die Daten vom Prof 180 zum ECB-Bus getrieben werden. In die Richtungsteuerung geht nicht nur ein, ob gelesen oder geschrieben wird (/rd) sondern auch, ob die adressierte Baugruppe auf dem CPU-Board oder 'draußen' auf dem Bus liegt. Wenn eine Baugruppe auf dem CPU-Board gelesen wird, muß der Bustreiber zum Bus hin treiben, um auf dem internen Datenbus keinen Datenkonflikt zu erzeugen. Außerdem wird auch das Signal **/busak** ausgewertet, da bei Zugriff eines externen DMA-Controllers auf Baugruppen des CPU-Bords die Treiberrichtung in Bezug auf /rd umgekehrt sein muß.

a19 ist ein high aktives Freigabesignal, das auf den ECB-Bus gelegt ist und bei Conitec u.a. für die Steuerung der Turbo-RAM benutzt wird.

Die Anlyse des disassemblierten PAL-Listings ergibt für die Steuerung der Freigaben von ROM und internem und externem RAM die Beziehungen:

Steuersignale mm1	mm0	A18	freigegebene Baugruppe	Pegel v. /gate	Pegel v. A19
0	0	0	ROM bei /rd; RAM bei /wr	1	0
0	0	1	RAM Bank 1	1	0
0	1	0	RAM Bank 0	0	0
0	1	1	RAM Bank 1	0	0
1	0	0	externes RAM	0	1

```
  1    0    1     RAM Bank 1                  0         0
  1    1    0     RAM Bank 0                  0         0
  1    1    1     externes RAM                0         1
```

Wenn mm0 und mm1 beide low sind, ist /gate high und es kann nicht auf Speicher auf dem ECB-Bus zugegriffen werden. Durch Umschalten auf mm0=1 wird das beim Lesen in die unteren 256K eingeblenete ROM abgeschaltet. Mit mm1=1 wird der grundsätzliche Zugriff auf externe Speicher ermöglicht. Dann wird mit mm0 gesteuert, ob der externe Speicher in die unteren oder die oberen 256K des HD64180-Adreßraums eingeblendet wird.

Für das Bereitstellen des bei Conitec 'externen' Erweiterungsspeichers nun innerhalb des CPU-Boards muß die Bussteuerung nun etwas anders gestaltet werden. Außerdem müssen Freigbesignale für zwei weitere RAM-Banks mit je 1 MB geschaffen werden. Dafür muß das PAL zunächst entrümpelt werden. Auf das Signal /gate kann man verzichten, wenn man die CPU-Karte nur als Master fahren will. Die Pins 19 der 74ALS245- -Bustreiber werden einfach an Masse gelegt.

Da kein externer Speicher zu steuern ist, wird auch das Signl a19 nicht mehr benötigt. Es sind dann zwei Ausgänge und ein Eingang am PAL freigeworden. Die beiden Ausgänge sollen die Freigabesignale für die beiden zustäzlichen 1MB-RAM-Blocks ergeben und der Eingang wird für die Selektion eines dieser beiden Blocks mit einem Signal Q2 beaufschlagt. Dann bekäme ein neues PAL den Inhalt:

```
PAL20L8
08.08.90                H. Bernhardt
Erweiterung des Prof 180x auf 2,5 MB RAM

me ioe rd lir busak a18 e mm0 mm1 adr a5 gnd
a2 a7 cas2 rom cas0 cas1 fdc cas3 iosel dir q2 vcc

/dir    = /ioe * /rd * /adr * a7 * /a5
        + /me * /rd
        + rd * lir * busak
/iosel  = /ioe * lir */a18 * /adr *  a7 * /a5 * /a2
/fdc    = /ioe * lir */a18 * /adr *  a7 * /a5 * a2
/rom    = /me * /a18     * /mm0 * /mm1 * /rd
/cas0   = /me * /a18 * e * /mm0 * /mm1 *  rd
        * /me * /a18 * e *  mm0
/cas1   = /me *  a18 * e * /mm0
        + /me *  a18 * e *  mm0 * /mm1
/cas2   = /me * /a18 * e * /mm0 *  mm1 * /q2
        + /me *  a18 * e *  mm0 *  mm1 * /q2
/cas3   = /me * /a18 * e * /mm0 *  mm1 *  q2
        + /me *  a18 * e *  mm0 *  mm1 *  q2
```

Das Steuersignal q2 und auch noch zwei Pseudoadressen zur Selektion eines von 4 Blocks mit 256K innerhalb eines 1MB-Blocks im Erweite- rungsspeicher werden mit einem zusätzlich bereitgestellten Latch 74LS174 erzeugt. Dieses Latch wird durch einen Out-Befehl an Port D7H beschrieben. D0 und D1 geben darin die Pegel der Pseudoadressen a18 und a19 und D2 den Pegel von q2 vor.

Die gesamte Erweiterungsschaltung incl. RAMs kommt auf einem Huckepackboard unter, das in die Sockel der Adreßmultiplexer Z17 und Z18 (74HC158) sowie Z11 (74HC51) und des Datenlatch Z15 (74LS374) der Druckerschnittstelle gesteckt wird und diese ICs mit aufnimmt. Damit stehen die gemultiplexten Adressen, die Daten, die Steuersignale /RAS, MUX und /MUX auf dem Huckepackboard zur Verfügung. Die Paeudo- Adressen a18' und a19' werden mit einem zusätzlichen 74HC51 Gatter gemultiplext.

Es müssen dann nur noch folgende Signale durch freie Verdrahtung zwischen Huckepackbord und CPU-Bord verbunden werden:

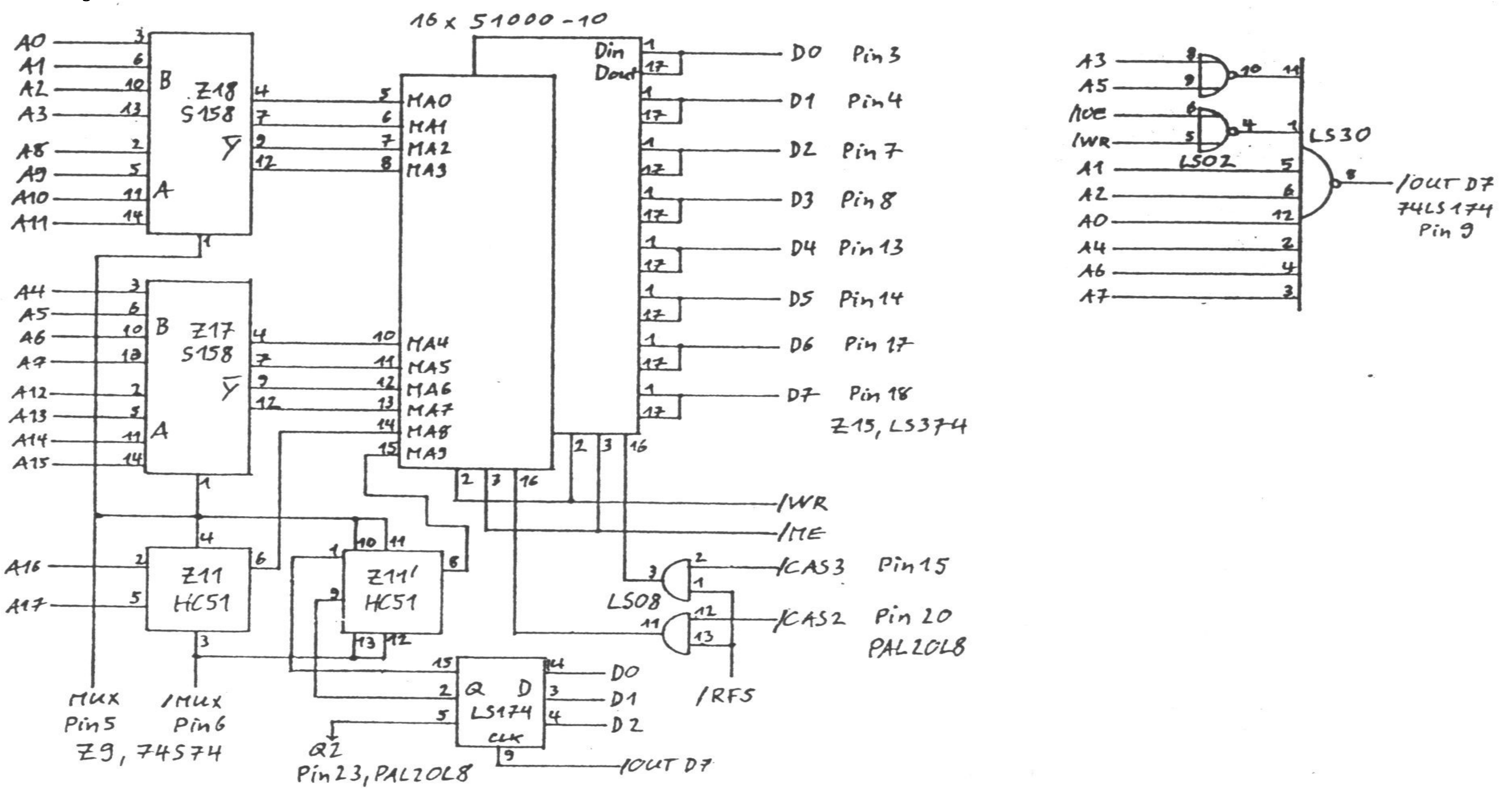
```
/ioe  (74LS02, Pin6)            von Pin5 des Z7, 74ALS245
/wr   (RAMs, Pin2)              von Pin6 des Z7, 74ALS245
/me   (RAMs, Pin3)              von Pin2 des Z7, 74ALS245
/rfs  (74LS08, Pins 1,13)       von HD64180, Pin57
q2    (74LS174, Pin5)           an Pin23 des PAL
/cas3 (74LS08, Pin12)           von Pin8 des PAL
/cas2 (74LS08, Pin1)            von Pin13 des PAL
```

Und wofür ist so viel Speicher gut? Natürlich wird der erstmal als RAM-Floppy genutzt. Anstatt diesen Speicher zusammen mit den restlichen 320K, die im Grundspeicher schon als RAM-Floppy genutzt werden, als eine RAM-Floppy zu betreiben, soll hier eine zweite RAM-Floppy von 2MB Kapazität eingerichtet werden. Das vereinfacht die Software etwas, wenn nicht ständig geprüft werden muss, ob ein zu übertragender Sektor nun auf der Erweiterung oder im Grundspeicher zu holen ist. Der benötigte Treiber läßt sich aus dem für die 'alte' RAM-Floppy bestehenden Treiber ohne große Änderungen umformen. Es kommt beim Login nur hinzu, daß mm1 auf high gesetzt werden muß.

Bei den Lese- und Schreib-Routinen ist noch nachzurüsten, daß der in (@trk) übergebene Wert aufzuteilen ist in D0 und D1, die zusammen mit einem gesetzten D2 das Adreßzusatzbyte des

DMAC ergeben, und daß D2-D4 nach zweimaligem Rechtsshiften als D0-D2 an den Port D7H ausgegeben werden. In der Login-Routine der 'alten' RAM-Floppy sollte vorsichtshalber grundsätzlich mm1 low gesetzt werden.

Abb.1: Schaltung



Folgende Datenstrukturen sind für das BDOS vorzugeben:

```
        cseg            ;Drive Table
@dtbl   dw      fdsd0,fdsd1,fdsd2,fdsd3 ; 4 Floppy Drives
        dw      fdram                   ; 'alte' RAM-Floppy
        dw      f2dram                  ; 2MB-RAM-Floppy
        dw      hdsk0,hdsk1,hdsk2,hdsk3 ; 4 Harddisk-Partitionen

        dseg            ;dph's koennen in Bank 0 liegen
                ; extended disk parameter header 2MB-RAM-Floppy
        dw      f2write
        dw      f2read
        dw      f2login
        dw      f2init
        dw      0,0
fd2parm dph     0,dpbr2

        cseg            ;disk parameter block fuer 2MB-RAM-Floppy
dpbr2   dpb     1024,64,2048,4096,512,0,8000
        ;1024 Bytes pro Sektor,   64 Sektoren pro Track
        ;2048 Tracks, Blockgroesse 4k, 512 DIR-Entries
```

```
        ;keine Systemspuren, unremocable media
```

Da diese 2MB (wie das sonst meistens bei RAM-Floppies für 8Bit-Computer aussieht) nicht I/O-mapped sind, kann man darin auch Programme laufen lassen. Es liegt dann allerdings am Können des Programmierers, mit dem BDOS ein Abkommen zu treffen, daß dieses nach dem Abarbeiten einer BDOS-Funktion wieder ordnungsgemäß die Bank und auch noch die entsprechende 256K-Portion einschaltet, in der das Programm den BDOS-Call abgesetzt hat.

Dieses Problem sei an die Soft-Cracks weitergegeben.

Abb.2: Bestückungs- und Anschlußplan: J1 bleibt bei 1MB-Cips offen.

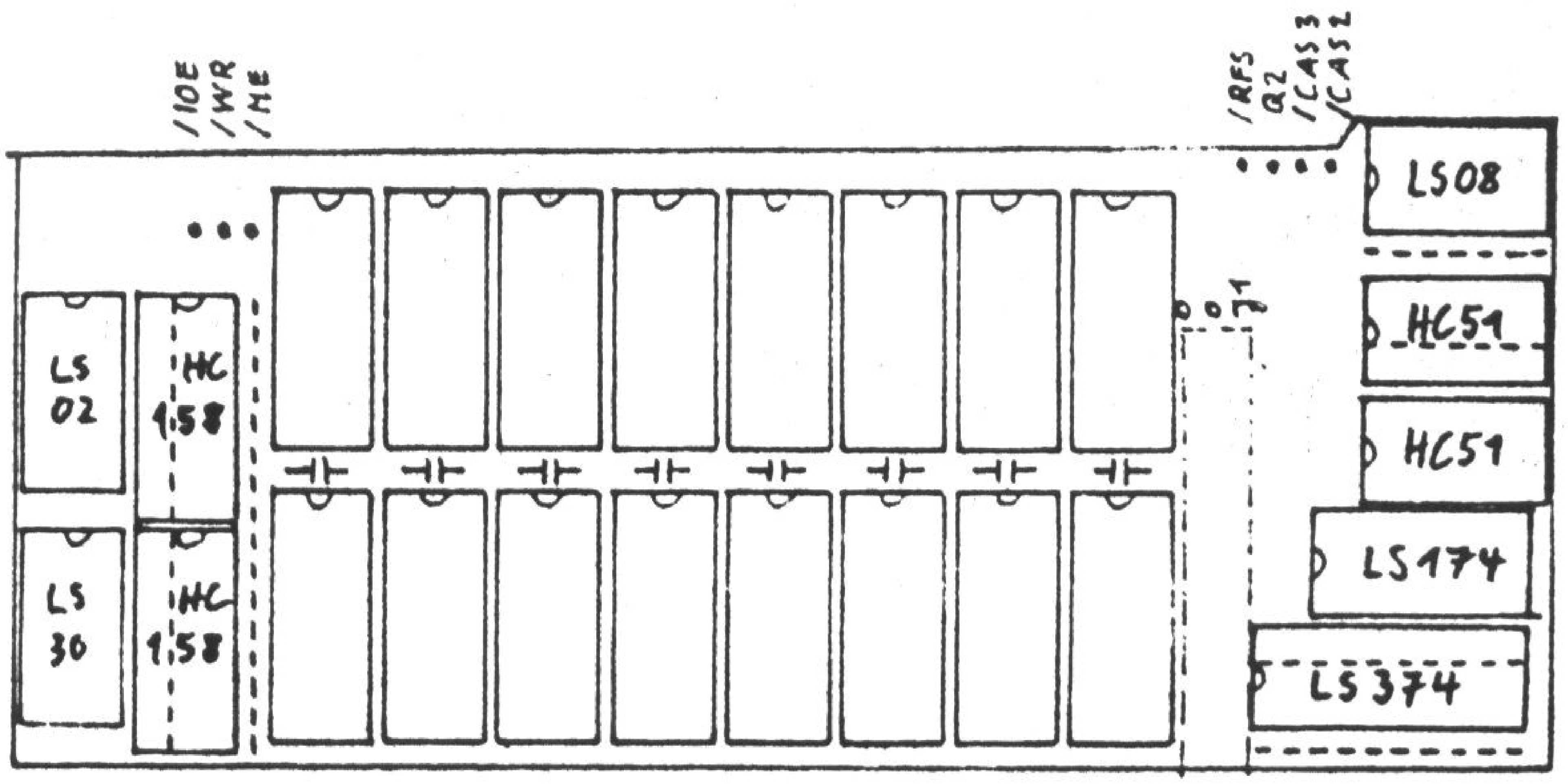


Abb.3: Layout der Bestückungsseite der 2MB-Erweiterung

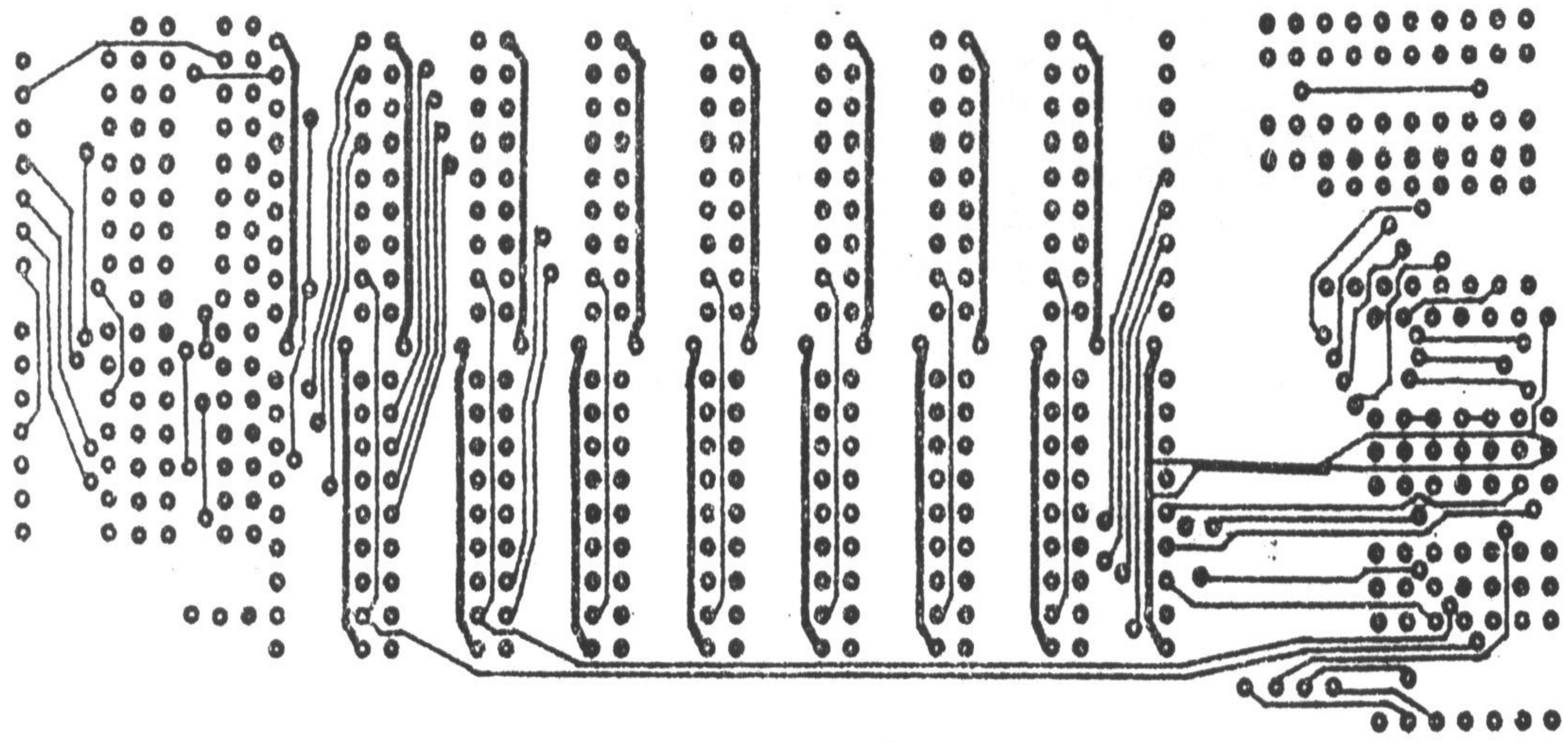
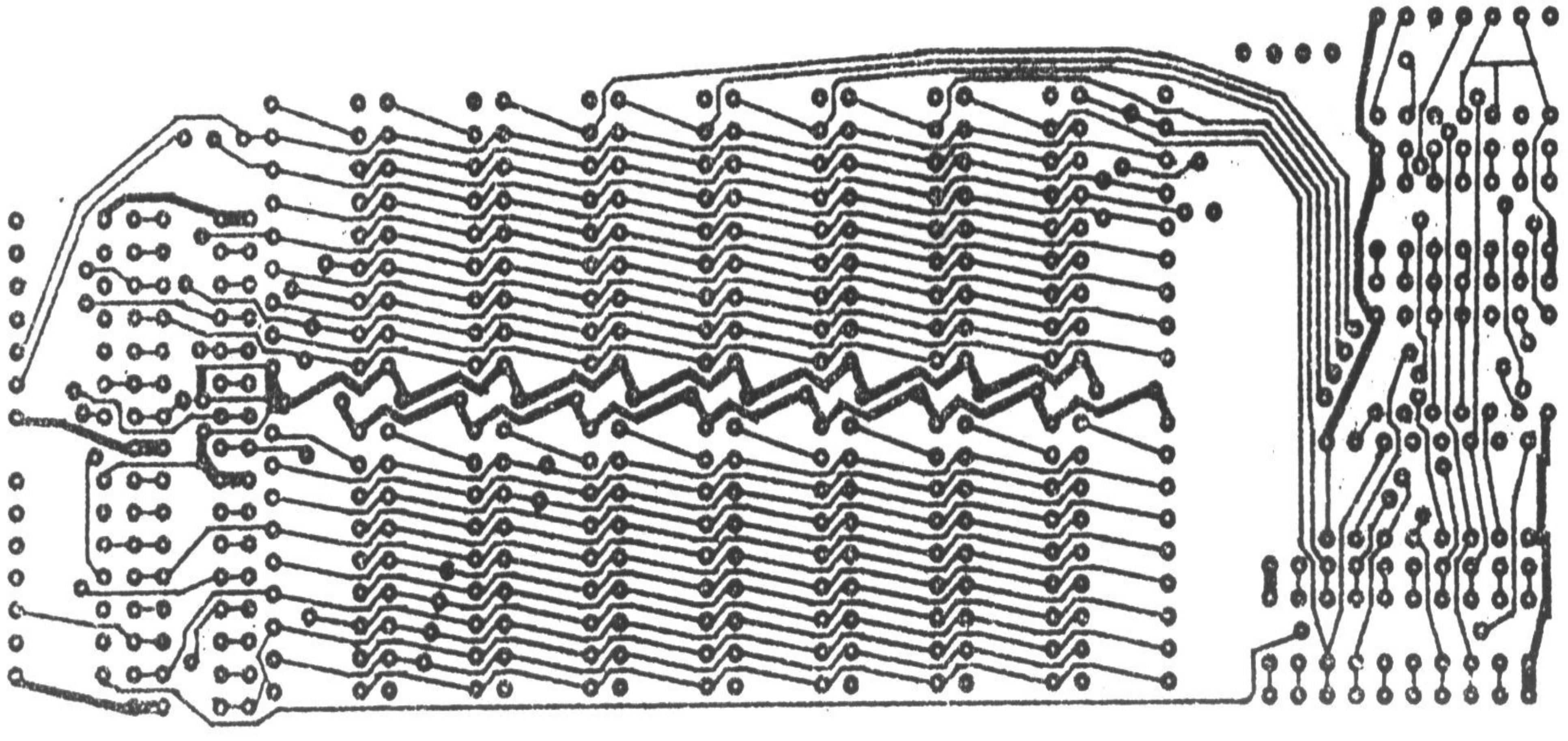


Abb.4: Layout der Lötseite der 2MB-Erweiterung



Textkonvertierung IBM-PC nach CP/M

Helmut Bernhardt

Wer abwechselnd unter CP/M und unter DRDOS (oder womöglich gar MSDOS) Texte schreibt, hat stets Probleme, diese Texte zwischen beiden Systemen auszutauschen. Das geringere Übel ist dabei, daß die auf dem PC eingegebenen deutschen Umlaute und Grafikzeichen unter CP/M nicht mehr zu erkennen sind; viel schlimmer ist, daß gelegentlich ab einer bestimmtem Stelle im Text unter CP/M vermeintlich nichts mehr da ist.

Woran liegt das? ---Daran:

Unter CP/M gibt es den 7Bit-ASCII-Code, mit dem 128 verschiedene Zeichen codiert werden können. Die ersten 32 Codes werden noch für Steuerzeichen für das Terminal, den Drucker oder wen auch immer belegt, so daß mit 96 Codes nicht allzuviele darstellbaren Zeichen unterschieden werden können. So hat man schon für die verschiedenen nationalen Sonderzeichen die Konvention eingeführt, daß deren Codes immer bei 5Bh-5Eh und 7Bh-7Eh liegen. Welches nationale Sonderzeichen mit einem dieser Codes erzeugt werden soll, kann z.B. beim Drucker über ein Mäuseklavier eingestellt werden oder allgemeiner über eine ESC-Sequenz vorgegeben werden.

Bei der Druckerausgabe ist es noch möglich, durch Umschalten mit entsprechenden ESC-Sequenzen abwechselnd die Sonderzeichen aller nationalen Zeichensätze innerhalb eines Textes zu benutzen. Dann müssen in dem Text-File diese Steuersequenzen enthalten sein und es handelt sich nicht mehr um einen reinen ASCII-Text, der universell übertragbar ist.

Ein Terminal kann aber immer nur eines der mit einem der Codes 5Bh-5Eh und 7Bh-7Eh gemeinten Zeichen darstellen, das dann in allen Fällen auf dem Bildschirm erscheint. Gleichzeitige Darstellung von z.B. [und Ä ist nicht möglich.

Dieses Übel wurde beim PC durch den 8Bit-ASCII-Code beseitigt. Das 8te Bit wird dafür verwendet, weitere 128 darstellbare Zeichen zu codieren. Die Codes 5Bh-5Eh und 7Bh-7Eh sind eindeutig den Zeichen [\]^{|} zugeordnet und alle anderen nationalen Sonderzeichen haben eindeutige Codes mit gesetztem Bit 7. Daneben gibt es noch etliche weitere Sonderzeichen und auch Grafikzeichen zum Erzeugen von Linien und Rahmen in einfacher, doppelter und gemischter Linienführung.

Da unter CP/M das Bit 7 eines ASCII-Codes nicht definiert ist, benutzen einige Textprogramme und auch Terminals es zur Unterbringung eines Attribut-Bits oder ignorieren es einfach. In diesem Fall werden die PC-Zeichen mit Codes 8Ch-9Fh als Steuerzeichen 00h-1Fh erkannt (z.B. Wordstar), was im Fall von "Ü" mit dem Code 9Ah als 1Ah = ^Z erkannt und als 'End of File' interpretiert wird. Der Rest des Files hinter dem Ü ist dann nicht zugänglich.

Da mit dem unter CP/M 3.0 laufenden Programm MSDOS.COM, dessen Turbo Pascal Source in dem Artikel "CP/M 3.0 goes MSDOS' in der c't abgedruckt war, 5¼"-360K-MSDOS-Disketten gelesen und geschrieben werden können, ist ein Transfer von Texten zwischen den Systemen physikalisch möglich. Es muß nur noch eine Anpassung der ASCII-Codes vorgenommen werden, falls das Text-File von DRDOS nach CP/M übertragen werden soll. In anderer Richtung ergibt eine Code-Umwandlung wenig Sinn, weil nicht feststeht, welches nationale Sonderzeichen anstelle der eckigen und geschweiften Klammern gemeint ist, oder ob nicht hier tatsächlich mal eine solche Klammer gemeint ist.

Eine Anpassung der Zeichen des PC-Zeichensatzes No2 an den 7Bit-ASCII- Zeichensatz leistet das beiliegende kleine Pascal-Programm, das einfache 8Bit-ASCII-Files (von Quick Star, dem Editor von Turbo Pascal, EDITOR.EXE von DRDOS...) umcodiert. Texte von MS-Word, Word Perfect, Star Writer PC, Word Star... lassen sich nicht (solange sie nicht mit einem entsprechenden Utility in reine ASCII-Texte konvertiert wurden) übertragen.

Das Programm läßt sich mit Turbo Pascal unter DRDOS und unter CP/M kompilieren und läuft unter beiden Betriebssystemen. Es werden nicht nur die deutschen Umlaute übersetzt sondern auch die Selbstlaute mit Akzenten in entsprechende Selbstlaute ohne Akzent übersetzt. Die Grafikzeichen werden je nach Form in die Zeichen ! - + übersetzt, wobei man unter CP/M die Linien leidlich erkennen kann, die im PC-Zeichensatz dargestellt sind. Für nicht übertragbare Zeichen wird ein Blank erzeugt.

Tabelle: Zeichen des PC-Zeichensatzes No2 und die daraus für den 7Bit-ASCII-Zeichensatz erzeugten Entsprechungen.

	8xh		9xh		Axh		BxH		Cxh		Dxh		Exh		Fxh	
0	Ç	C	É	E	á	a	░		└	+	╨	−	α	a	≡	=
1	ü	}	æ	ë	í	i	▒		┴	−	╤	−	β	~	±	+
2	é	e	Æ	[	ó	o	▓		┬	−	╥	−	Γ	T	≥	>
3	â	a	ô	o	ú	u	│	!	├	!	╙	+	π		≤	<
4	ä	{	ö	:	ñ	n	┤	!	─	−	╘	+	Σ		⌠	
5	à	a	ò	o	Ñ	N	╡	!	┼	+	╒	+	σ		⌡	
6	å	a	û	u	ª	a	╢	!	╞	!	╓	+	µ		÷	
7	ç	c	ù	u	º	o	╖	+	╟	!	╫	+	τ		≈	
8	ê	e	ÿ	y	¿		╕	+	╚	+	╪	+	Φ		°	
9	ë	e	Ö	\	⌐		╣	!	╔	+	┘	+	Θ		∙	
A	è	e	Ü	]	¬		║	!	╩	−	┌	+	Ω		·	
B	ï	i	¢	C	½		╗	+	╦	−	█		δ		√	
C	î	i	£		¼		╝	+	╠	!	▄		∞		ⁿ	
D	ì	i	¥	Y	¡		╜	+	═	−	▌		ø		²	2
E	Ä	[	₧		«	<	╛	+	╬	+	▐		∈		■	
F	Å	A	ƒ		»	>	┐	+	╧	−	▀		∩			

Da diese Umcodierung über eine Tabelle erledigt wird, kann man sich dort noch weitere Stellvertreter-Zeichen für den 7Bit-Zeichnsatz einbauen, ohne an der Programmlogik herumschnitzen zu müssen.

```
program PC2CPM
{uses crt;      nur unter Turbo Pascal 4.0 nötig, kann sonst kein UPCASE}

const logo : string = 'PC => CP/M Textkonvertierung,   (C) 06.09.91 Be';

Umcode : array [0..127] of Byte =
($43,$7D,$65,$61, $7B,$61,$61,$63, $65,$65,$65,$69, $69,$69,$5B,$65,
 $45,$7b,$5b,$7c, $7c,$6f;$75,$75, $79,$5c,$5d,$63, $20,$59,$20,$20,
 $61,$69,$6f,$75, $6e,$4e,$61,$6f, $20,$20,$20,$20, $20,$20,$3c,$3c,
 $20,$20,$20,$21, $21,$21,$21,$2b, $2b,$21,$21,$2b, $2b,$2b,$2b,$2b,
 $2b,$2d,$2d,$21, $2d,$2b,$21,$21, $2b,$2b,$2d,$2d, $21,$2d,$2b,$2b,
 $2d,$2d,$2d,$2d, $2b,$2b,$2b,$2b, $2b,$2b,$2b,$20, $20,$20,$20,$20,
 $61,$7e,$54,$20, $20,$20,$20,$20, $20,$20,$20,$20, $20,$20,$20,$20,
 $3d,$2b,$3e,$3c, $20,$20,$20,$20, $20,$20,$20,$20, $20,$32,$20,$20;
{       Position in der Tabelle ist PC-ASCII-Code abzüglich 128.
        Wert in der Tabelle ist der entsprechende CP/M-ASCII-Code
        Für nicht übertragbare Zeichen wird ein Blank erzeugt}

var  PC, CPM               : file of byte;
     Quelle, Ziel          : string;
     Ptr, scan, i, Zeichen : byte;

procedure GetSource;
begin
  repeat
    write('PC-Text-File : ');   readln(Quelle);
    for scan := 1 to length (Quelle do Quelle[scan] :=
        upcase(Quelle[scan];
```

```
    assign(PC,Quelle);   {$I-}   reset(PC);   {$I+}
  until oiresult=0;
end;

begin
  if ParamCount <> 0 then begin
    Quelle := ParamStr(1); for scan := 1 to length(Quelle) do
      Quelle[scan] := upcase(Quelle[scan]
    assign(PC,Quelle);   {$I-}   reset(PC);   {$I+}
    if ioresult <> 0 then begin
      writeln('kann ',Quelle, ' nicht oeffnen!');   GetSource;
    end;
  end;
  else GetSource;

  Ziel:='';  scan:=1;
  repeat
    Zile := Ziel + Quelle[scan];   scn := scan + 1;
    if Quelle[scan] = '.' than scan := length(Quelle);
  until scan = length(Quelle);

  Ziel := Ziel + '.CPM';
  assign(CPM,Ziel);   {$I-}   rewrite(CPM);   {$I+}
  if ioresult <> 0 then begin
    writeln('kann ',Ziel,' nicht anlegen!');   Halt;
  end;

  writeln(logo);
  repeat
    read(PC,Zeichen);
    if (Zeichen and $80) <> 0 then begin
      Ptr := (Zeichen and $7F);   Zeichen := umcode[Ptr]:
    end;
    write(CPM,Zeichen);
  until eof(PC);
  close(PC);   close(CPM);
  writeln(Quelle,' in,' Ziel. konvertiert !')
end.
```

VGA-Karte anstelle der Hercules-Karte am Club-80-Terminal

Helmut Bernhardt

So sehr viel was anderes als die Hercules-Karte ist die VGA-Karte auch nicht. Bei beiden Karten ist der Bildwiederholspeicher sowohl im Text- als auch im Grafik-Modus memory mapped und die CPU hat sich selbst darum zu kümmern, an welche Adressen die Zeichen bzw. die Grafik-Bytes zu verholen sind. Beide Karten sind völlig passiv und weisen keinerlei Videoprozessor-Funktionen auf.

VGA-Karten gibt es in einer Vielzahl von Erscheinungsformen, sie können einen 8- oder 16-Bit-Bus haben, sie können im Fall eines 16-Bit-Bus in 8-Bit-Slots des PC funktionieren oder dies auch unterlassen, sie können ihr BIOS in 8- oder 16-Bit-Breite zugänglich machen, sie können 256K oder auch bis 2MB Video-RAM enthalten, sie können maximal 256, 32K, 64K oder 16 mega Farben gleichzeitig auf dem Monitor darstellen, sie können eine einfache VGA-Karte oder eine dazu abwärtskompatible Video-Karte mit zusätzlichem Grafik-Prozessor (Windows-Accellerator) sein, sie können die Super-VGAModi VESA-kompatibel oder auch mit Wildwuchs-Einbindungen oder auch garnicht unterstützen, und es ist nicht abzusehen, welche Überraschungen noch zu erwarten sind bzw. bereits existieren.

Für den PC-Benutzer ist diese Vielfalt - abgesehen von Performance- Unterschieden - relativ schnuppe, dort funktioniert die Karte fast immer und mit entsprechenden Treibern auch unter Windows mit ihren eventuell zusätzlichen Eigenschaften.

Wem eine VGA-Karte im Club-80-Terminal laufen soll, dürfen einige Umstände nicht gegeben sein. Von den Abmessungen her darf es nicht eine der riesigen Lappen sein, die an ECB-Bus nicht mehr als Huckepack-Karte auf das Terminal paßt. Es darf auch keine 16-Bit-Karte, die im 8-Bit-Slot nicht läuft, sein, und sie muß registerkompatibel zur ursprünglichen VGA von IBM sein (was meistens der Fall ist).

Hier sollen die Grundlagen der Einbindung der VGA-Karte in das Betriebsprogramm des Club-80-Terminals herausgestellt werden.

Leider geht es nicht ganz ohne Hardware-Eingriffe, deshalb soll erst dieser Aspekt behandelt werden.

Hardware-Anpassung

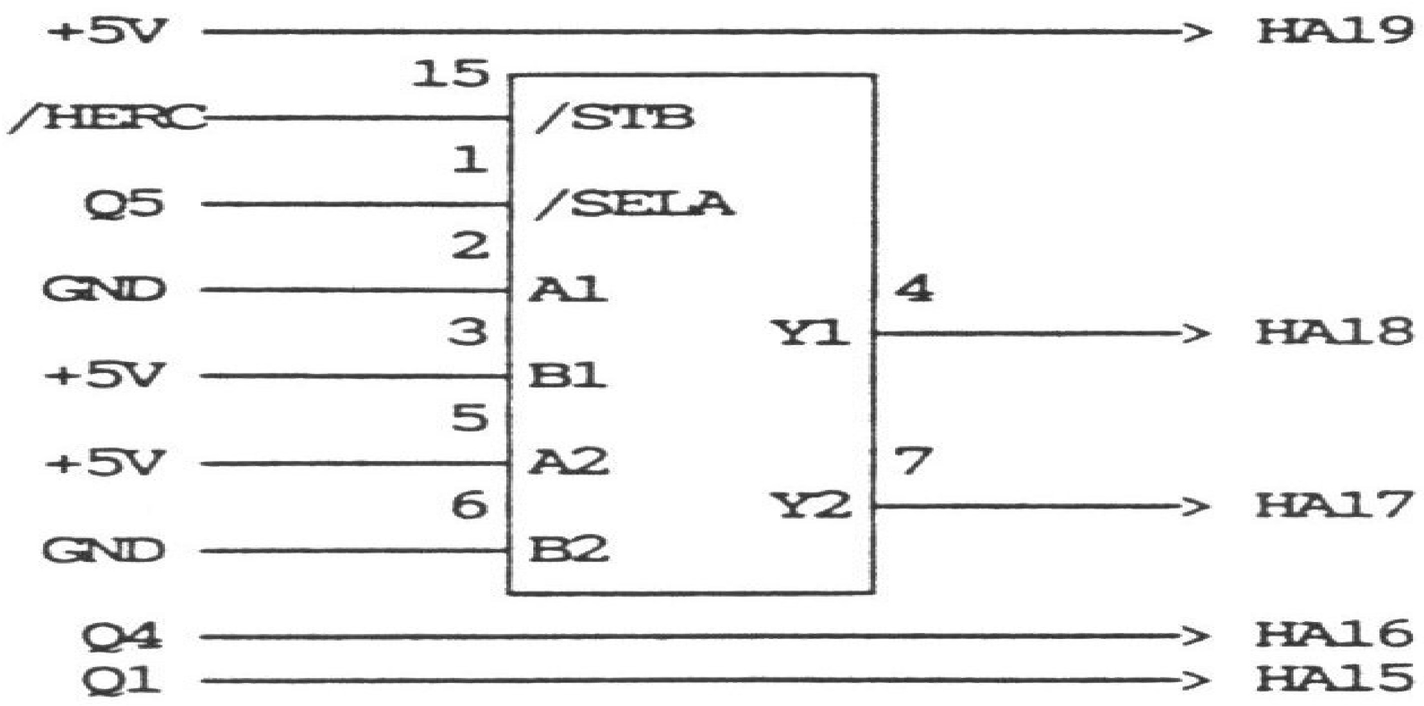
Die Hercules-Karte belegt im PC zwei 32K-Video-Seiten an den Adressen B0000..B7FFF und B8000..BFFFF. Zur Erzeugung des Freigabesignals für den Speicherzugriff genügte es, eine low aktive Adreßleitung A18 auf die Hercules-Karte einwirken zu lassen, wenn die CPU ein low aktives /MERQ und einen High Pegel auf A15 ausgab. Die an die Hercules-Karte gelangenden Adressen A16, A17 und A19 waren dauerhaft high und der Pegel von A15 (zur Hercules) wurde durch ein Bit eines Latch gesteuert.

Die VGA-Karte benutzt diese Speicherbereiche auch als Video-RAM in den Text-Modi. Im Grafik-Modus wird aber der Bereich A0000. .AFFFF benutzt und das BIOS in einem ROM auf der VGA-Karte belegt den Bereich C0000. .C7FFF. Natürlich läuft die darin enthaltene Software nicht auf dem Z80 des Terminals -

es gibt aber eine Menge wichtiger Tabellen in diesem ROM und eine Reihe von Zeichensätzen (die für die Benutzung in den Textmodi erst in die Speicherebene 2 geladen werden müssen). Die Pegel der Adressen A15..A19, die für die Freigabe der einzelnen Bereiche maßgeblich sind, seien hier zusammengestellt -daraus ergibt sich dann auch die einfache Schaltung, die diese Bereiche ansteuern kann.

```
Adreß-          A19 A18 A17 A16 A15     Speicherart
bereich         (zur VGA-Karte)
--------------------------------------------------------------
A0000..A7FFF     1   0   1   0   0      Grafik, erste Hälfte
```

A8000..AFFFF	1	0	1	0	1	Grafik, zweite Hälfte
B0000..B7FFF	1	0	1	1	0	Text, monochrom
B8000..BFFFF	1	0	1	1	1	Text, color
C0000..C7FFF	1	1	0	0	0	BIOS-ROM



Es ist zu erkennen, daß A19 für alle Bereiche high ist. Diese Leitung kann an +5V angeschlossen bleiben. A18 und A17 haben entweder das Muster 01 oder 10, wenn hier 00 oder 11 anliegt, wird die VGA nicht angesprochen. Damit läßt sich eine Freigabe der Karte herleiten, in die die bisherige Freigabe (/HERC = /MERQ * A15) einbezogen wird. Dafür ist ein zusätzliches IC 74HCT157 nötig, das in das Rasterfeld am oberen Ende der Terminal-Karte eingelötet und folgendermaßen beschaltet wird:

/HERC ist das bisherige Freigabe-Signal für die Hercules-Karte, dessen Verbindung zu HA18 (A18 des PC-Slot) getrennt wird und Q5, Q4 und Q1 sind Ausgänge des 74HCT-259-Latch (wobei Q1 bereits an HA15 und +5V an HA19 des PC-Slot liegen.

Wenn /HERC nicht low aktiv ist, sind alle Ausgänge des 74HCT157-Multiplexers high (oder low - so genau weiß ich das nicht aus dem Kopf), auf alle Fälle liegt dann nicht eines der Freigabemuster 01 oder 10 an HA18/HA17.

Nun aber zur Programmierung:

Im Laufe der Jahre hat sich die Video-Ausgabe beim PC von ursprünglich einer maximalen Auflösung von 640*200 Punkten im Monochrom-Modus auf heute 1024*768 Punkten mit gleichzeitig 256 aus 256k Farben oder auch 640*480 Punkten mit gleichzeitig 16 mega Farben entwickelt. Für eine Reihe von Auflösungen haben sich dafür Standards entwickelt, andere werden nie benutzt bzw. von jedem Hersteller mit anderen Modus-Nummern bedacht. Hier werden deshalb nur die von IBM abgesegneten Modi 00h bis 13h behandelt; die folgenden Überlegungen gelten dann auch für alle mit IBM registerkompatibel VGA-Karten.

Standard Video Modi der VGA

Modus		Auflösung	Farben	Zeichen- Box	Seiten	Adresse	Länge	Berechnung der Offsetadresse
0h	Text	40x25	16	8x8	8	B800	800	2*(40*y+x)
1h	Text	40x25	16	8x8	8	B800	800	2*(40*y+x)
2h	Text	80x25	16	8x16	8	B800	1000	2*(80*y+x)
3h	Text	80x25	16	8x16	8	B800	1000	2*(80*y+x)
4h	Grafik	320x200	4	8x8	1	B800	4000	8152*(y and 2)+y*40 + x div 4
5h	Grafik	320x200	4	8x8	1	B800	4000	8152*(y and 2)+y*40 + x div 4
6h	Grafik	640x200	2	8x8	1	B800	4000	8152*(y and 2)+y*40 + x div 4
7h	Text	80x25	mono	9x14	8	B000	1000	2*(80*y+x)
8h	reserviert							
9h	reserviert							
Ah	reserviert							
Bh	reserviert							
Ch	reserviert							
Dh	Grafik	320x200	16	8x8	8	A000	2000	y*40 + x div 8
Eh	Grafik	640x200	16	8x8	4	A000	4000	y*80 + x div 8
Fh	Grafik	640x350	mono	8x14	2	A000	8000	y*80 + x div 8

```
10h  Grafik 640x350   16   8x14  2      A000  8000      y*80 + x div 8
11h  Grafik 640x480  mono  8*16  1      A000  A000      y*80 + x div 8
12h  Grafik 640x480   16   8*16  1      A000  A000      y*80 + x div 8
13h  Grafik 320x200  256   8*8   1      A000  FFFF      320*y +x
```

in den Modi 0h bis 3h (und 7h bei der VGA) hat die EGA-Karte 14 und die VGA-Karte 16 Rasterzeilen pro Zeichen. **Erweiterte Super-VGA-Modi für Tseng ET4000 Chip**

(und HiColor-DAC) Diese Auflösungen können bei anderen Super-VGA-Karten mit anderen Modus-Nummern zu bedienen sein; erst der VESA-Standard hat hier nachträglich Einheitlichkeit gebracht, ältere Karten wußten davon noch nichts. Es ist auch nicht bekannt, wo für diese Modi in der Parameter-Table die Daten stehen.

```
   29h  Grafik 800x600  16   8*16   1   A000  FFFF
   2Eh  Grafik 640x480  256  8*16   1   A000  FFFF
        Grafik 640x480  32k  8*16   1   A000  FFFF mit Sierra-DAC
   30h  Grafik 800x600  256  8*16   1   A000  FFFF
        Grafik 800x600  32k  8*16   1   A000  FFFF mit Sierra-DAC
   37h  Grafik1024x768  16   8*16   1   A000  FFFF
   38h  Grafik1024x768  256  8*16   1   A000  FFFF
        Grafik1024x768  32k  8*16   1   A000  FFFF mit Sierra-DAC
```

Grundsätzlich gibt es zwei Strategien, wie dem Z80 des Terminals all die Informationen zur Initialisierung eines Video-Modus zur Verfügung gestellt werden:

1. Auf dem PC wird durch entsprechende Programme festgestellt, wo im BIOS-ROM diese Informationen vorliegen und der Z80 bedient sich aus dem BIOS-ROM selbst.
2. Alle Informationen zur Initialisierung der relevanten Modi werden auf dem PC ausgelesen und in ein EPROM gebrannt, das anstelle des BIOS-EPROM in die VGA-Karte gesteckt wird.

Für beide Methoden soll hier das Vorgehen im Einzelnen beschrieben werden. Die zweite Methode kann eventuell bedeuten, daß der Inhalt des EPROM noch byteweise auf zwei EPROMs aufgespalten werden muß, wenn eine 16-Bit-VGA vorliegt.

Mit welchen Werten der Registerdschungel einer VGA-Karte in den verschiedenen Modi zu initialisieren ist, weiß das BIOS der VGA-Karte aus einem Feld von Tabellen im BIOS-ROM. Beim Hochfahren des PC wird eine Initialisierungs-Routine im VGA-BIOS aufgerufen, die außer der Initialisierung des Modus 03h u.a. auch einen Save-Table-Pointer an der Adresse 0000h:04A8h ablegt, der (wie nicht schwer zu erraten) auf eine Save-Table zeigt. Diese Save-Table ist ein Feld von weiteren Pointern deren erster auf den Anfang der Tabellen mit den Registerinhalten für die verschiedenen Video-Modi zeigt. Behufs dieser Information könnt man an die Initialisierungs-Werte für die Register der VGA heran. Der Offset der Werte für einen Video-Modi 00h bis 13h (bzw. 11h bei der EGA-Karte) geht aus diesen beiden Arrays hervor.

```
const VGAOfs: array[0. .$13] of Word =
        ($5C0,$5C0,$600,$600, $100,$140,$180,$640,
         $200,$240,$280,$2C0, $300,$340,$380,$440,
         $480, $680, $6C0, $700) ;
const EGAOfs: array[0..16] of Word =
        ($4C0,$500,$540,$580, $100,$140,$180,$1CO,
         $200,$240,$280,$2C0, $300,$340,$380,$440, $480);
```

Aufbau der Parameter-Tabelle eines Video Modus, je Modus 64 Bytes

```
                                                    IndexAdr  DataAdr  Index
----------------------------------------------------------------------------
   0    Byte    Anzahl Bildschirm-Textspalten           BIOS-Variablen
   1    Byte    Anzahl Bildschirm-Textzeilen
   2    Byte    Zeichenhöhe in Rasterzeilen
   3    Word    Speichergröße pro Bildseite
                Sequenzer Register                 3C4h       3C5h
   5    Byte    Clocking Mode Reg.                                       1
   6    Byte    Map Mask Reg.                                            2
   7    Byte    Character Generator Select Reg.                          3
   8    Byte    Memory Mode Reg.       4
                Allgemeine Register
   9    Byte    Miscellaneous Output Reg.               3C2h (W), 30Ch (R)
                CRTC Register                           3B4h/3D4h 3B5h/3D5h
  10    Byte    Horizontal Total Reg.                                    0
  11    Byte    Horizontal Display End Reg.                              1
  12    Byte    Start Horizontal Blanking Reg.                           2
  13    Byte    End Horizontal Blanking Reg.                             3
  14    Byte    Start Horizontal Retrace Reg.                            4
  15    Byte    End Horizontal Retrace Reg.                              5
  16    Byte    Vertical Total Reg.                                      6
  17    Byte    Overflow Reg.                                            7
  18    Byte    Preset Row Scan Reg.                                     8
  19    Byte    Maximun Scan Line Reg.                                   9
```

```
 20    Byte    Cursor Start Reg.                                       10
 21    Byte    Cursor End Reg.                                         11
 22    Byte    Start Address High Reg.                                 12
 23    Byte    Start Address Low Reg.                                  13
 24    Byte    Cursor Location High Reg.                               14
 25    Byte    Cursor location Low Reg.                                15
 26    Byte    Vertical Retrace Start Reg.                             16
 27    Byte    Vertical Retrace End Reg.                               17
 28    Byte    Vertical Display End Reg.                               18
 29    Byte    Offset Reg.                                             19
 30    Byte    Underline Location Reg.                                 20
 31    Byte    Start Vertical Blanking Reg.                            21
 32    Byte    End Vertical Blanking Reg.                              22
 33    Byte    Mode Control Reg.                                       23
 34    Byte    Line Compare Reg.                                       24
               Attribute Controller Register         3C0h 3C0h(W),3C1h(R)
 35    Byte    Palette Reg. 0                                           0
 36    Byte    Palette Reg. 1                                           1
 37    Byte    Palette Reg. 2                                           2
 38    Byte    Palette Reg. 3                                           3
 39    Byte    Palette Reg. 4                                           4
 40    Byte    Palette Reg. 5                                           5
 41    Byte    Palette Reg. 6                                           6
 42    Byte    Palette Reg. 7                                           7
 43    Byte    Palette Reg. 8                                           8
 44    Byte    Palette Reg. 9                                           9
 45    Byte    Palette Reg. 10                                         10
 46    Byte    Palette Reg. 11                                         11
 47    Byte    Palette Reg. 12                                         12
 48    Byte    Palette Reg. 13                                         13
 49    Byte    Palette Reg. 15                                         14
 50    Byte    Palette Reg. 16                                         15
 51    Byte    Mode Control Reg.                                       16
 52    Byte    Overscan Reg.                                           17
 53    Byte    Color Plane Enable Reg.                                 18
 54    Byte    Horizontal Pel Panning Reg.                             19
               Graphics Controller Register            3CEh    3CFh
 55    Byte    Set/Reset Reg.                                           0
 56    Byte    Enable Set/Reset Reg.                                    1
 57    Byte    Color Compare Reg.                                       2
 58    Byte    Data Rotate Reg.                                         3
 59    Byte    Read Map Select Reg.                                     4
 60    Byte    Mode Reg.                                                5
 61    Byte    Miscellaneous Reg.                                       6
 62    Byte    Color Don't Care Reg.                                    7
 63    Byte    Bit Mask Reg.                                            8
----------------------------------------------------------------------------
```

Mit SAVEROMP wird ein Z80-Source-Listing RCMTABLE.ASM erzeugt, das per DEFB die Tabellen der Initialisierungdaten für die relevanten Video-Modi enthält.

Eine andere Strategie besteht darin, die verschiedenen Video-Modi der VGA durch das BIOS einstellen zu lassen und im jeweiligen Modus die Inhalte der Register auszulesen und diese in der Struktur der ROM-Tabelle in eine Datei zu schreiben. Das Programm SAVEREGS erzeugt auf diese Art as Sourcelisting RECTABLE.ASM.

Soft-Zeichensätze laden

In den Text-Modi gilt es außerdem noch, daß im Zeichengenerator-RAM (Speicherebene 2) ein passender Zeichensatz geladen ist. Zeichensätze in den Auflösungen 8*8, 8*14, 8*16, 9*14 und 9*16 liegen im VGA-RCM gebrauchsbereit vor. Bei der Darstellung von 9*14 und 9*16 Rasterpunkten pro Zeichen wird durch die Hardware nach je 8 Rasterpunkten entsprechend der Information aus dem Zeichensatz ein neunter Punkt in der Farbe des Hintergrunds erzeugt.

Den Anfang eines Zeichensatzes im ROM kann man sich über eine Funktion des VGA-BIOS-Interrupt 10h ebenfalls vom BIOS holen. Damit sind auch die Zeichensätze für andere Benutzer als einen 80X86 zugänglich. Man kann entweder mit GETVPARM sich ein Listing ZEIGER.ASM erzeugen, das alle Pointer auf Tabellen und Zeichensätze im VGA-BIOS (übersetzt in die Sicht das Z80 im Terminal) enthält, oder sich mit dem Programm SAVEFONT ein Listing VGAFONTS.ASM erzeugen, das die Zeichensätze selbst enthält.

Laden der Color-Lookup-Table

Damit ist auf dem Bildschirm aber noch nicht viel zu sehen, denn die Werte in den 16 Palettenregistern ergeben noch nicht direkt die auf dem Monitor dargestellte Farbe, diese 8 Bit Werte

stellen Zeiger in die Color Lookup Table (CLUT) dar. die CLUT ist ein Register-Array vcn 256 3-Byte-Registern worin jeweils nur die Bits 0..5 von Bedeutung sind, es sind dies die Werte der Farbintensitäten (0..63) für die Farbkomponenten R, G und B.

Der lange Weg zun Zustandekommen einer Farbe auf dem Monitor ist dieser: Im Attribut-Byte eines Zeichens im Video-RAM kodieren die Bits 0..3 die Farb-Nr. des Textzeichens und die Bits 4..7 die Farb-Nr. des Hintergrunds. Für beide Komponenten wird aus dem Palettenregister mit der Farb-Nr. der Inhalt des Registers als Zeiger in die CLUT verwendet. Die dort vorliegenden Werte für R, G und B werden schließlich über den Video-DAC analog gewandelt und zum Monitor geschickt.

Die Standard-Inhalte der CLUT kann man sich auch wieder durch ein kleines Programm (SAVECLUT) vom BIOS einstellen lassen, auslesen und als Assembler-Source-Listing (FARBEN.ASM) auf die Diskette schreiben lassen.

Für dieses Problem habe ich noch keine Alternativ-Lösung gefunden, um die an versteckter Stelle im VGA-BIOS irgendwo vorliegenden Werte aus dem BIOS-ROM heraus direkt zu benutzen. Wenn das BIOS-ROM der VGA seihst im Terminal benutzt werden soll, muß FARBEN.ASM in das Be-triebsprogramm des Terminals gelinkt werden.

Damit hätte man dann sehen mal die Informationen, mit welchen Inhalten die Register der VGA im jeweiligen Video-Modus zu versorgen sind. Dabei ist zu bedenken, daß die Register 0..6 des CRTC erst beschrieben werden können, wenn im Register 11h D7=0 gesetzt wird. Der CRTC liegt in den Monochromodi auf der Portadresse 3B4h/3B5h und in den Farbmodi bei 3D4h/3D5h. Monochrom und Farbe wird durch D0 des Miscellaneous Output Registers vorgegeben. Man kann sich dieses Bit in der Tabelle (Byte 9) ansehen.

Während der Programmierung der Register, die für das Video-Timing zuständig sind, sollte im Reset Register des Sequenzers (Index 0) immer 03h stehen. Anschließend muß dort 00h eingetragen werden, um den Resetzustand der VGA zu beenden.

Um zu zeigen, was bei der Initialisierung der Register nacheinander zu erfolgen hat, sei das Programm FINITEST und darin insbesondere die Procedure write_regs(mode:byte) ein Anhaltspunkt. Sie ist in Z80-As-sembler umzusetzen und in das Terminal-Betriebsprogramm anstelle der derzeitigen Initialisierung der Hercules-Karte einzumassieren. Als grundsätzlicher Initialisierungs-Video-Modus sollte Modus 03h verwendet werden.

Die 3-Digit-Portadressen des PC reduzieren sich beim Club-80-Terminal auf die niederwertigsten beiden Digits (A8 und A9 werden bei Portzugriffen auf die Video-Karte immer high-gesteuert, so daß automatisch das höchstwertige Digit '3' ausgegeben wird). Das hier auch erfolgende Update der BIOS-Variablen NrOfColumns bis Page-Size braucht nicht in Z80-Assembler übertragen zu werden.

Es gibt zusätzlich zu den oben behandelten Registern noch einige weitere Register, die hier kurz einzeln vorgestellt werden sollen:

```
Feature Control Register: Port 3BAh/3DAh (wr), 3CAh (rd, nur VGA)
        Dieses Register wird im PC sowohl bei EGA- als auch bei VGA-Karten
        in allen Videomodi mit 00h geladen.

Input Status #0 Register: Port 3C2h (nur Lesen)

        D0..D3: unbenutzt
        D4:     0= Dip-Schalter geschlossen
                1= Dip-Schalter offen
                Lesen eines von 4 Dip-Schaltern, D3 und D2 im
                Miscellaneous Output Reg. geben den Schalter vor
        D5:*    logischer Pegel von Pin 19 des Feature Connectors
        D6:*    Logischer Pegel von Pin 17 des Feature Connectors
        D7:     0= Kein Vertical Retrace Interrupt IRQ2 auf getreten
                1= Vertical Retrace Interrupt IRQ2 aufgetreten
           * nur bei EGA

Input Status #1 Register: Port 3BAh/3DAh (nur Lesen)
        D0:     1= Hor/Vert Retrace aktiv
        D1:     1= Lichtgriffelimpuls aufgetreten
        D2:     0= Lichtgriffelschalter gedrückt
        D3:     1= Vertical Retrace (nicht Horizontal)
        D4,D5:  gibt log. Pegel zweier Video-Signale wieder
                welche beiden Signale, gibt das Color Plane Enable
                Reg. des Attribute Controller vor

Reset Register des Sequenzers: Port 3C4h Index(:=0), Part 3C5h Data
        D0,D1:  0,0: VGA-Hardware ist aktiv
                1,1: Reset-Zustand, keine Signale zum Monitor

Color Select Register des Attribute Controllers; Port 3C0h Index
   (:=20), 3C1h Data
        D0..D3: Farbregister Adreßbit 4..7
```

ist in allen Videomodi mit 00h zu befüllen

CGA- und Hercules-compatible Register

Diese Register lassen sich nur erreichen, wenn die Karte ein Umschalten auf Register-Kompatibilität zu diesen Karten erlaubt. Das Umschalten erfolgt durch Ausgabe von D0=0 (für CGA) und D0,D1=0 (für Hercules) an das Mode Control Register des CRTC und Bedröhnen der für die jeweilige Karte im folgenden aufgeführten Register mit den für diese Karte gängigen Werten.

```
   6845-Register:               Port 3B4h/3D4h (Index), 3B5h/3D5h (Daten)
   Mode Control Register:       Port 3B8h/3D8h
   Status Register:             Port 3BAh/3DAh
   Configuration Register       Port 3BFh       (nur Hercules)
   Color Select Register:       Port 3D9h       (nur CGA)
```

Diese Register haben die gleichen Funktionen wie bei der Hercules-Karte bzw. der CGA-Karte. Da sie für die Hercules-Karte im Beitrag zum Club-80-Terminal bereits beschrieben wurden, wird hier kein weiterer Platz verschwendet.

Spezielle Register, die für die tägliche Arbeit wichtig sind

die Standardwerte für diese Register stehen in der Parameter-Table und werden durch InitVGA eingetragen

Map Mask Register des Sequenzers: Port 3C4h Index(: =2), 3C5h Data

D0..03: 1= Schreiben in die RAM-Ebene 0..3 freigegeben

Character Map Select Register des Sequenzers:
Port 3C4h Index(:=3), 3C5h Data

D5(nur VGA), D1, D0: Nr des Blocks des primären Zeichensatz
D6(nur VGA), D3, D2: Nr des Blocks des sekundären Zeichens.

Die Zeichensätze werden in den Textmodi in der Steicherebene 2 in aufeinanderfolgenden 8K.Bereichen gespeichert. Ob das darzustellende Zeichen aus dem primären oder sekundären Zeichensatz zu nehmen ist, entscheidet das Attributbit 3. Damit lassen sich 512 verschieden Zeichen gleichzeitig darstellen. Die Anzahl der Farben, in denen ein Zeichen dargestellt werden kann, reduziert sich in diesem Modus auf 8.

Mode Control Register des CRTC: Port 3B4h/3D4h Index(:=23),
3B5h/3D5h Data

```
        DO:     0= CPU-A13 := Rasterzeilenadreßbit 0
                1= CPU-A13 wirkt direkt auf das Video-RAM
        D1:     0= CPU-A14 := Rasterzeilenadreßbit 1
                1= CPU-A14 wirkt direkt auf das Video-RAM
                        wenn D0=0, verhält sich die VGA wie eine CGA
                        wenn D0 und D1 = 0, verhält sich die VGA wie
                           eine Hercules
        D2..D6: sollte man in Ruhe lassen
        D7:     0= CRTC-Reset, alle Register werden mit 0 geladen
```

Das Miscellaneous Register des Grahics Controllers:
Port 3CEh Index(:=6), 3CFh Data

```
        D0:     0= Textmodus, 1= Grafikmodus
        D1:     0= Adreßbit 0 unverändert       siehe [1] Seite 392
                1= Adreßbit 0 substituiert
        D3,D2:  00= A0000..BFFFF Video-RAM Adresse
                01= A0000..AFFFF 10= B0000..B7FFF
                11= B8000..BFFFF
                (dadurch erübrigt sich leider nicht der Hardpatch)
```

Das Index Register des Attribute Controllers:
Port 3C0h (wr), 3C1h (rd)

Wie bei CRTC u.s.w. wird hier in Bit 0..4 vorgegeben, auf welches Datenregister (3C0h wr, 3C1h rd) zugegriffen werden soll. Zusätzlich steuert D5, ob die CPU (D5=0) oder die Video-Hardware (D5=1) auf die Register zugreifen kann. D5 ist nach Verändern der Register immer auf 1 zu setzen.

Durch Lesen des Input Status #1 Registers wird auf das Indexregister für Schreibzugriffe geschaltet, beim nächsten Schreibzugriff wird dann das Datenregister angesprochen.

```
Die Register des Video-DAC:
        Zugriffe auf den Video-DAC sollten nur während des Vertical
        Retrace stattfinden (D0,D3 des Input Status #1 Reg. high).
  PEL Adress Read Mode Register: Port 3C7h (wr) 3C8h (rd, Wert+1)
  PEL Adress Write Mode Register: Port 3C8h (rd/wr)
        diese beiden Register sind Index Register mit unterschied-
        lichen Adressen für Lesen oder Schreiben des eigentlichen
        PEL Data Registers. Beim Lesen des PEL Address Read Mode
        Registers erhält man einen um 1 erhöhten Wert gegenüber dem
        vorher eingeschriebenen Wert.
  PEL Data Register:    Port 3C9h (rd/wr)
        Durch dreimaliges Lesen/Schreiben dieses Registers erhält/
        schreibt man nacheinander die 6Bit-Farbwerte für Rot, Grün
        und Blau. Danach wird automatisch die Adresse im Index Re-
        gister erhöht, sodaß sofort die nächsten 3 Farbwerte gele-
        sen/geschrieben werden können.
DAC State Register:     Port 3C7h (rd)
        D1,D0:  00= Lesemodus aktiv, 11= Schreibmodus aktiv
PEL Mask Register:      Port 3C6h (rd/wr)
        8Bit AND-Maske für Farbregister-Adresse; damit lassen sich
        einzelne Bits, die von den Palettenregistern kommen, desak-
        tivieren.
```

Das Laden der Zeichensätze

Die Adresse der Zeichensätze im BIOS-ROM läßt sich das Programm GETVPARM durch das BIOS selbst geben und trägt sie (ungerechnet auf die Adressen im Terminal) in VGA.ASM ein.

In den Textmodi beinhaltet Video-Speicher eine lineare Folge von Bytepaaren (Zeichen und Attribut), wobei eine lineare Zuordnung der Bildschirmposition zur Adresse im Video-RAM besteht. Aus den ersten beiden Einträgen in der Parameter-Tabelle eines Modus können Zeilenlänge und Anzahl Bildschirmzeilen entnommen werden. Die Größe des für eine Seite benötigten Video-RAMs ergibt sich aus

2 * Zeichen pro Zeile * Zeilen pro Seite

Literatur:
[1] Matthias Uphoff: Die Programmierung der ESA/VGA Grafikkarte; Addison-Wesley; ISBN 3-89319-274-3
[2] Michael Tischer: PC Intern 3.0; Data Becker Verlag 1992; ISBN 3-89011-591-8
[3] Arno Fritz: Regenbogen im Eigenbau - Programmierung der 256--Color-Modi des ET3000; mc 8/91, S.120
[4] Michael Schulte: Heißer Farb-Ofen - Wie man aus dem ET4000--Chipsatz das Beste herausholt; mc 12/91, S.140
[5] Dirk Meiners: Farbe bekennen - Auswahl der Farbapaletten bei VGA-Karten; c't 4/91, S.330
[6] Peer Meier: Schöne bunte Welt, Hardwarenahe Programmierung von VGA-Karten; c't 4/91, S.338 und c't 5/91, S.292
[7] Detlef Grell: Kleine Extras - Video-Modi und Optionen bei Super-VGAs einstellen; c't 1/92, S.156

Das Thema mit seinem ursprünglich gedachten Zweck wird nur wenige Leute interessieren; die Pascal-Sources beinhalten aber alles, was zur direkten VGA-Programmierung nötig ist, und sind deshalb eventuell auch für PC-Programmierer interessant.

Damit in diesem Info noch Platz für weitere Beiträge bleibt, habe ich darauf verzichtet, die. 110 KB Source-Listings der PC-Programme und die damit erzeugten Assembler-Sources für den Terminal-Z80 hier abzudrucken. Wer daran Interesse hat, schickt mir eine leere Diskette und 2,—DM in Briefmarken und erhält dann im PC-Format die Programme.

Sidecar mit 4 Rädern

Amiga's Sidecar wird ein selbständiger PC

Helmut Bernhardt, Christof Ueberschaar

Eine der aufwendigsten Varianten, einen PC-XT zu betreiben, besteht darin, anstelle einer Tastatur und einer Video-Karte mit angeschlossenem Monitor einen kompletten 68000-Computer mit einem aufwendigen Interface für solche Kleinigkeiten zu bemühen. Dieser vornehme Stil wurde beim Gespann Amiga1000 und dem dazugehörigen Sidecar gewahrt. Kurze Zeit nach der Einführung wurde diese aufwendige Kombination durch die Nachfolgemodelle Amiga500 und Amiga2000 abgelöst. Die Motherboards des Amiga Sidecars werden heute (1988) noch im Elektronik-Ramschhandel für ca. 15 DM verkauft.

Bei näherer Untersuchung stellt sich heraus, daß dieses Board fast ein vollständiges XT-Motherboard mit 256K RAM (aufrüstbar auf 512K) und zusätzlichem Floppy-Controller ist. Um aus dem Sidecar einen selbständigen PC zu machen, muß nur noch ein Tastatur-Interface nachgerüstet werden und eine Video-Karte in eines der leider nur 3 vorhandenen Slots gesteckt werden. Dabei mag es tröstlich sein, daß ein Floppy-Controller bereits auf dem Motherboard vorhanden ist.

Aus den Schaltplänen eines anderen PC-Kompatiblen geht hervor, wie der Anschluß einer Tastatur zu erfolgen hat. Danach sollen die seriell/parallel-gewandelten Scan-Codes der Tastatur über den als Eingang konfigurierten Port A eines 8255-Parallelport-Bausteins eingelesen werden. Die Steuerung der Tastatur wird über D6 und D7 des als Ausgang eingestellten Port B des 8255 durchgeführt. Die anderen Bits des Port B erledigen weitere Aufgaben wie Sound-Ausgabe, Rücksetzen von Flags bei Parity-Fehlern im Speicher oder Fehlern bei Erweiterungskarten und Anwahl der unteren oder oberen Vierergruppe des Dil8-Schalters (zur Vorgabe der Systemkonfiguration) zum Auslesen über die untere Hälfte des Port C des 8255. Die obere Hälfte des Port C dient zum Auslesen der Fehlerflags und des jeweiligen Pegels am Lautsprechereingang.

Einen 8255 sucht man im Sidecar aber vergebens. Andererseits müssen bis auf Parity-Check (die Sidecar-Hardware sieht dies nicht vor) und Tastatur-Anschluß (was durch das Amiga-Interface bereitgestellt wird) alle anderen Funktionen des 8255 auch im Sidecar auf andere Art realisiert sein, denn die entsprechenden Steuerungen funktionieren dort einwandfrei.

Auf der Suche nach Fragmenten des 8255 stößt man zunächst auf U25, 74HCT174. Mit diesem Latch werden 5 Ausgabe-Bits des Port B des 8255 bereitgestellt. Die Pins 2, 4 und 6 bleiben dort aber unberücksichtigt.

Der Port C des 8255 wird durch U20, 74HCT257 (Auslesen des Dil8-Schalters SW1 über D0-D3) und U26, 74HCT367 (Auslesen von Fehlerflags über D4-D7) dargestellt. Port A, über den die Scan-Codes der Tastatur einzulesen sind, fehlt gänzlich und eine Möglichkeit, die über die verfügbaren Bits von Port B eingestellte Systemkonfiguration zurückzulesen, ist auf dem Motherboard auch nicht zu finden. Diese Funktionseinheiten werden offensichtlich vom Amiga-Interface bereitgestellt und müssen bei dessen Wegfall nachgerüstet werden.

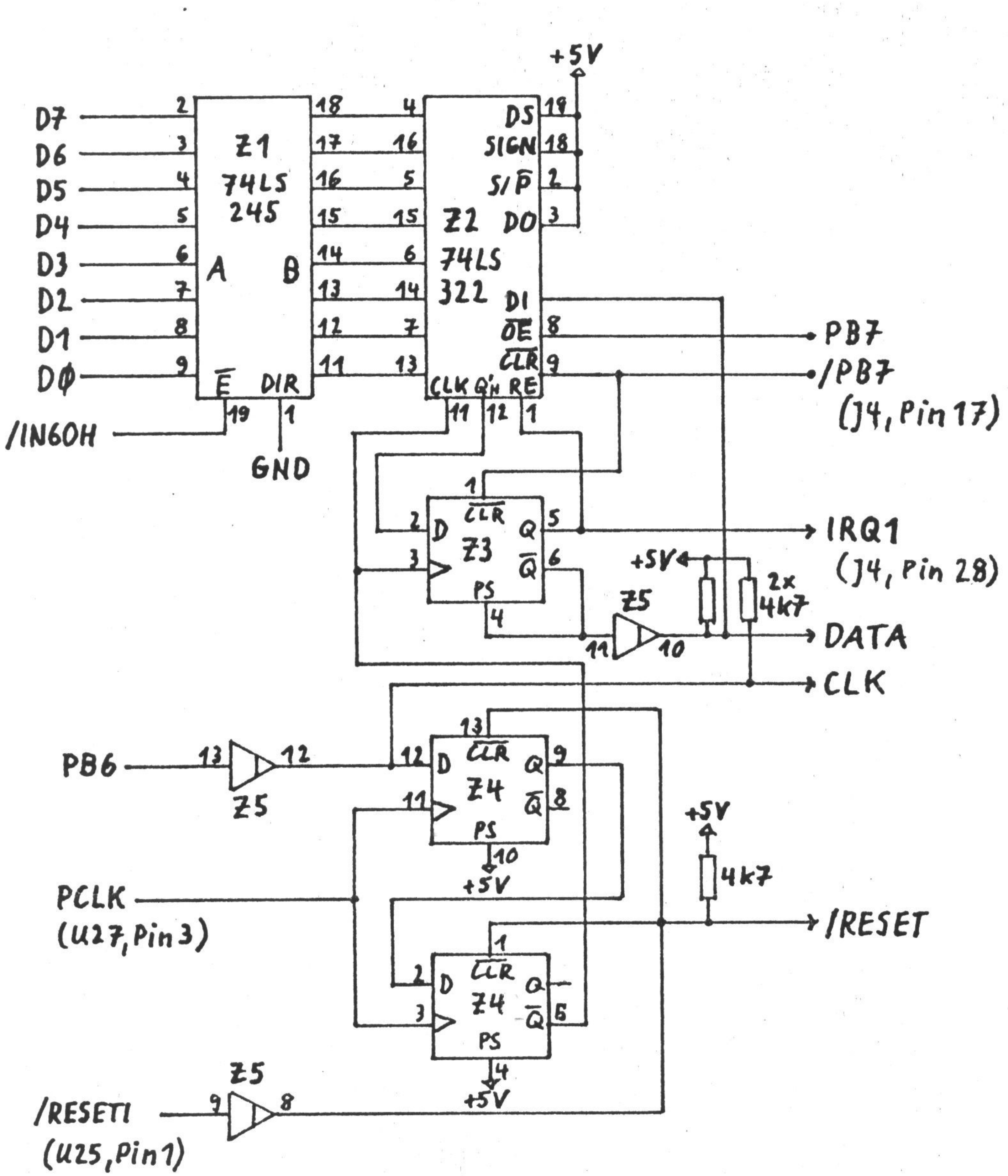


Abb. 1.1 Nachgebildeter Port A eines 8255 mit Seriell/Parallel-Wandler und Handshaking zwischen Computer und Tastatur

Die in Abbildung 1.1 gezeigte Schaltung stellt den mit einem 74LS245

nachempfundenen Port A des 8255 dar, über den ein 74LS322-Seriell/ Parallel-Wandler angeschlossen ist. Das über D6 und D7 von Port B abgewickelte Handshaking zwischen Tastatur und Computer wird mit einigen Flip Flops und Open Collector Treibern realisiert. Das Freigabesignal zum Auslesen von Port A und auch die Freigabesignale zum Lesen und Schreiben von Port B werden mit der in Abbildung 1.2 wiedergegebenen Schaltung erzeugt.

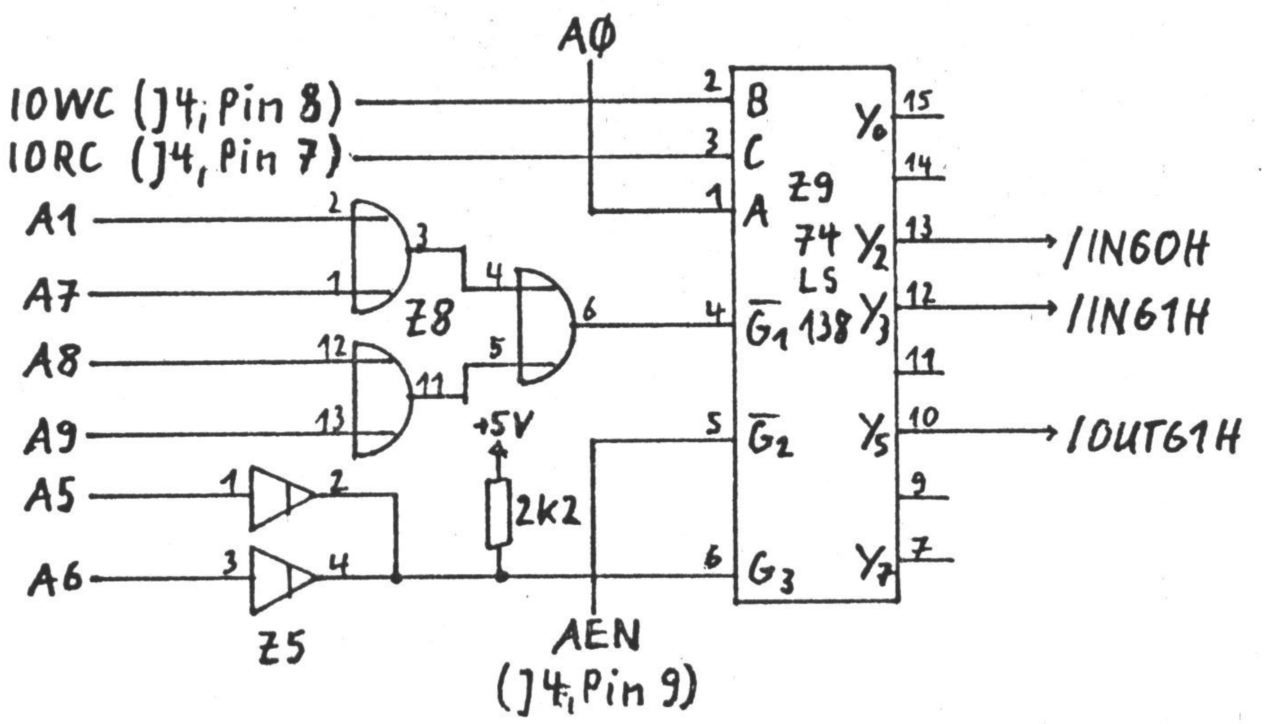


Abb. 1.2 Erzeugen der Freigabe-Signale des Tastatur-Ports

Der Port B läßt sich gemäß Abbildung 1.3 auch rücklesbar gestalten. Um den Verdrahtungsaufwand zwischen Zusatzschaltung und Motherboard in Grenzen zu halten, wird der Port B zusätzlich zu der in U25 auf dem Motherboard schon vorhandenen Form auf dem Tastatur-Interface nochmals durch einen 74LS576 und einen 74LS245 unter Berücksichtigung aller 8 Bits zur Verfügung gestellt (Abb. 1.3).

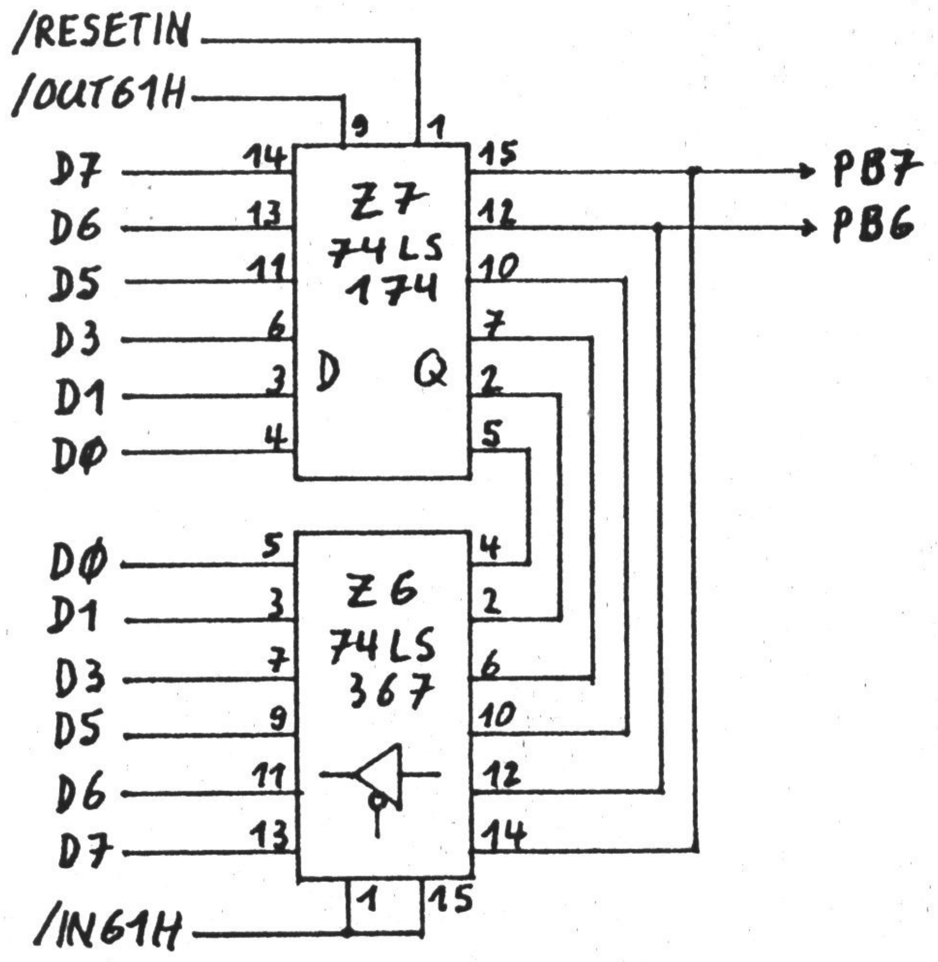


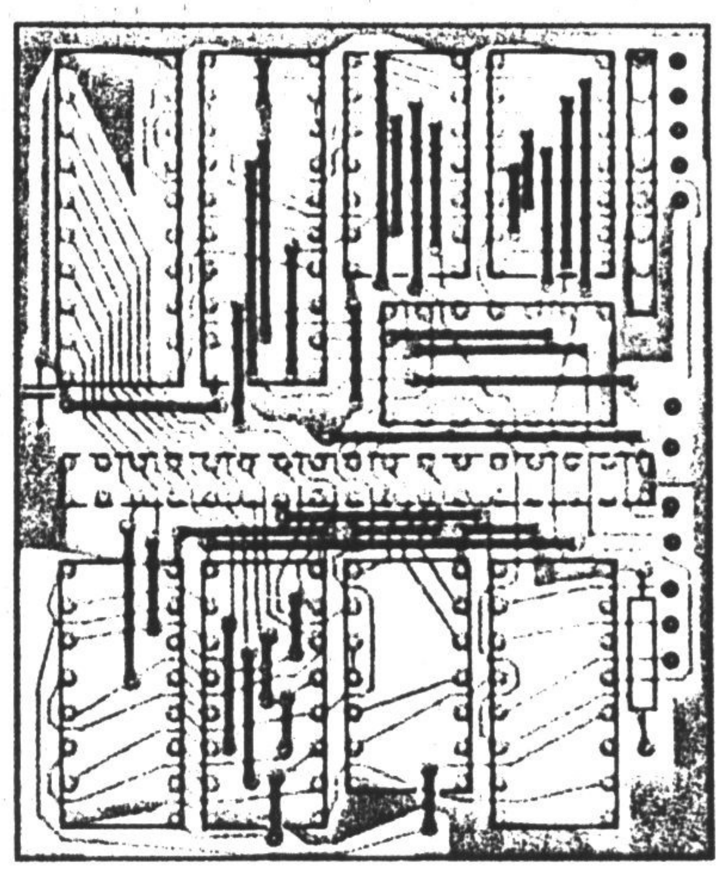
Abb. 1.3 Der auslesbare Port B des 8255

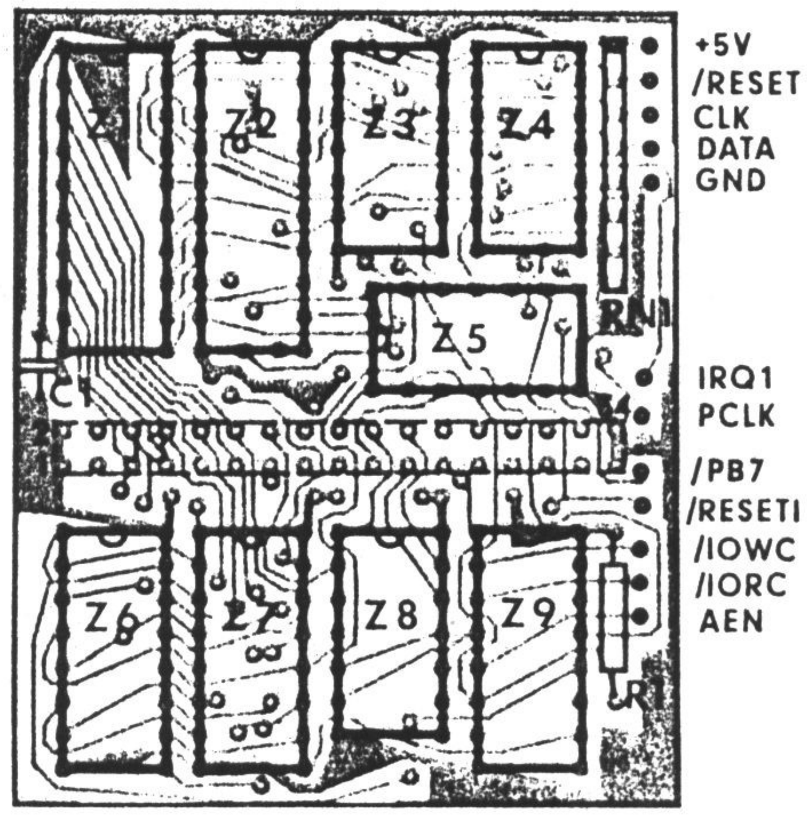
Diese Schaltungselemente lassen sich auf einer kleinen Platine unterbringen, die anstelle des Amiga-Interface auf die Steckerleiste J3 aufzusetzen ist. Dafür muß von der Lötseite des Platinchens eine 34polige Pfostenbuchse angelötet werden. Bei Beschaffungsschierigkeiten leistet auch eine VG64-a,b-Buchse nach Ablängen auf 34 Kontakte diesen Dienst.

Außer den auf J3 zugänglichen Signalen (+5V, GND, Daten und Adressen) müssen noch einige Signale von J4 und von 2 Punkten auf dem Motherboard zugeführt werden. Die Anschlußpunkte auf dem Platinchen sind aus Abbildung 2.3 zu ersehen, sie müssen mit folgenden Punkten auf dem Motherboard verbunden werden.

2.1 beim einseitigen Layout sind etliche Drahtbrücken nötig

2.2 Bestückung und Anschluß der freien Verdrahtungen vom Motherboard





Signal von/zu Tastatur-Interface	**Anschlußpunkt auf dem Motherboard**
IRQ1	J4, Pin 28
PCLK	U27, 74LS74, Pin 3
/PB7	J4, Pin 17
/RESET IN	U25, 74HCT174, Pin1
/IOWC	J4, Pin 8
/IORC	J4, Pin 7
AEN	J4, Pin 9

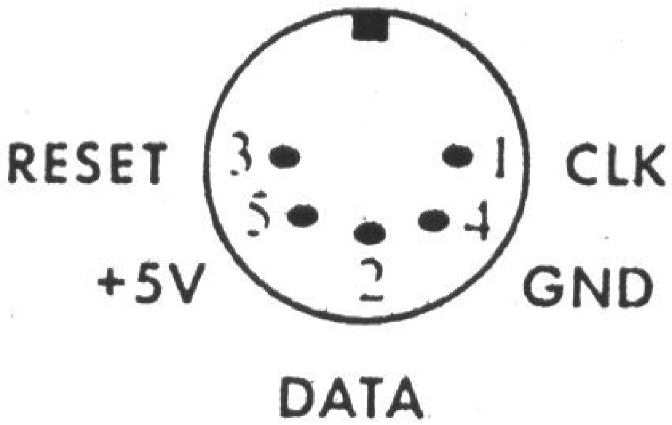
Stückliste

Z1	74LS245	R1 2k2
Z2	74LS322	RN1 Widerstandsnetzwerk
Z3	74LS74	8-Pin, 7*4K7 oder 5-Pin, 4*4K7
Z4	74LS74	C1 100nF Keramik
Z5	7407	J3 34polige Buchsenleiste
Z6	74LS245	(lötseitig bestücken)
Z7	74LS576	Drahtbrücken (isolierter Draht)
Z8	74LS32	Leiterkärtchen
Z9	74LS138	

Der einfachen Schaltung entsprechend ist das Layout einseitig mit einer entsprechenden Anzahl Drahtbrücken (Abb. 2.2) ausgeführt. Die Drahtbrücken müssen vor dem Einlöten der ICs bzw. der IC-Sockel (nur anreihbare Typen verwenden) in dünnem Kupferlack-Draht oder Wrapdraht gezogen werden, um bei der teilweise dichten Lage Kurzschlüsse zu vermeiden.

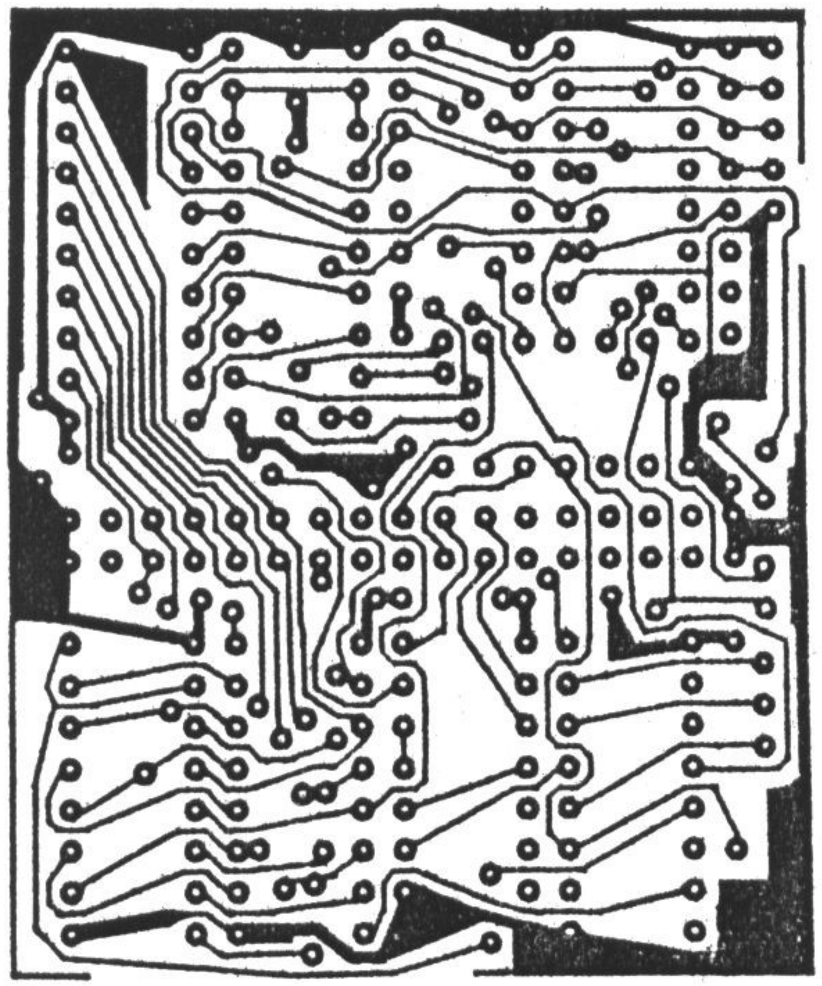
Neben dem Widerstandsnetzwerk sind die Anschlüsse für die Tastatur zu finden. Die Belegung der üblichen 5poligen Diodenstecker von PC-Tastaturen geht aus Abbildung 3 hervor.

Abb. 3 Signalbelegung der 5poligen Diodenbuchse am PC für den Anschluß der Tastatur



Ganz ohne Eingriffe auf dem Sidecar-Motherboard geht es leider doch nicht. Pin 9 von U27, 74LS74, der ebenfalls den Tastatur-Hardware-Interupt IRQ1 bedient, muß durchtrennt werden. Für eine einwandfreie Funktion ohne Amiga-Interface sind die Schalter 1 und 2 des Dil4-Schalters SW2 auf ON zu stellen, damit nicht die Ausgänge eines CD4020-Zählers die Bits 0 und 3 des Datenbus bei Zugriff auf den Videokarten-Statusport 3BAh (Schalter 1 OFF) bzw. 3DAh (Schalter 1 ON) belegen.

2.3 Das Layout des Tastatur-Interface, bestückungsseitig gesehen



Ein PC-XT mit 4,77MHz Takt und 512K RAM ist nun nichts Umwerfendes, er reicht aber für die meisten Anwendungen aus und kann auf diese Weise sehr kostengünstig realisiert werden. Wer etwas mehr herausholen will, kann einen V20-Processor und ein V20-BIOS einsetzen und gewinnt dadurch nochmal gute 10% mehr Geschwindigkeit.

Leider paßt er wegen der Anordnung der Slots in kein übliches PC-Gehäuse und stellt nach Einstecken von Video-Karte und Harddisk-Controller auch nur noch ein freies Slot zur Verfügung. Auch beim Anschluß eines Netzteils muß man die einzelnen Kabel dem anders belegten Anschlußstecker auf dem Motherboard anpassen.

Belegung der Pfostenleisten J3 und J4 im Sidecar

Stecker J3

+5V	1 **o**	**o** 2	+5V	+5V	1 **o**	**o** 2	+5V
		o 4	nc	/MEMR	3 **o**	**o** 4	/MEMW
D0	5 **o**	**o** 6	D1			**o** 6	nc
D2	7 **o**	**o** 8	D3	/IORC	7 **o**	**o** 8	/IOWC
D4	9 **o**	**o** 10	D5	AEN	9 **o**	**o** 10	/DACK0
D6	11 **o**	**o** 12	D7	GND	11 **o**	**o** 12	CLK
A0	13 **o**	**o** 14	A1	GND	13 **o**	**o** 14	IRQ3
A2	15 o	o 16	A3	IRQ7	15 **o**	**o** 16	IRQ4

A4	17	**o**	**o**	18	A5	/PB7	17	**o**	**o**	18	U29,Pin11
A6	19	**o**	**o**	20	A7	I/OCHRDY	19	**o**	**o**	20	/RESET IN
A8	21	**o**	**o**	22	A9	/WR	21	**o**	**o**	22	U40,Pin13
A10	23	**o**	**o**	24	A11	U40,Pin11	23	**o**	**o**	24	nc
A12	25	**o**	**o**	26	A13	nc	25	**o**	**o**	26	nc
A14	27	**o**	**o**	28	A15	U40,Pin12	27	**o**	**o**	28	IRQ1
A16	29	**o**	**o**	30	A17	SW2,4	29	**o**	**o**	30	SW2,1
A18	31	**o**	**o**	32	A19	SW2,3	31	**o**	**o**	32	SW2,2
GND	33	**o**	**o**	34	GND	GND	33	**o**	**o**	34	GND

Dein GIIIs, die 4 unbekannten Wesen

A: Hardware-Erweiterung für 1 MB RAM Helmut Bernhardt, Arnulf Sopp

Die 128 Refreshzyklen des Z80 lassen sich auf 256 Zyklen erweitern, was die Verwendung von 41256-RAMs ermöglicht (Abb. 3). Dadurch kann jeder Z80-Computer auf mindestens das vierfache Speichervolumen aufgebohrt werden, indem man die vorhandenen 4164-RAMs durch 41256-Typen ersetzt. Die beiden zusätzlich nötigen Adressen kann ein I/O-adressiertes Latch liefern (Abb. 1). Damit kann man zwischen vier Blocks mit je 64 kB um- schalten.

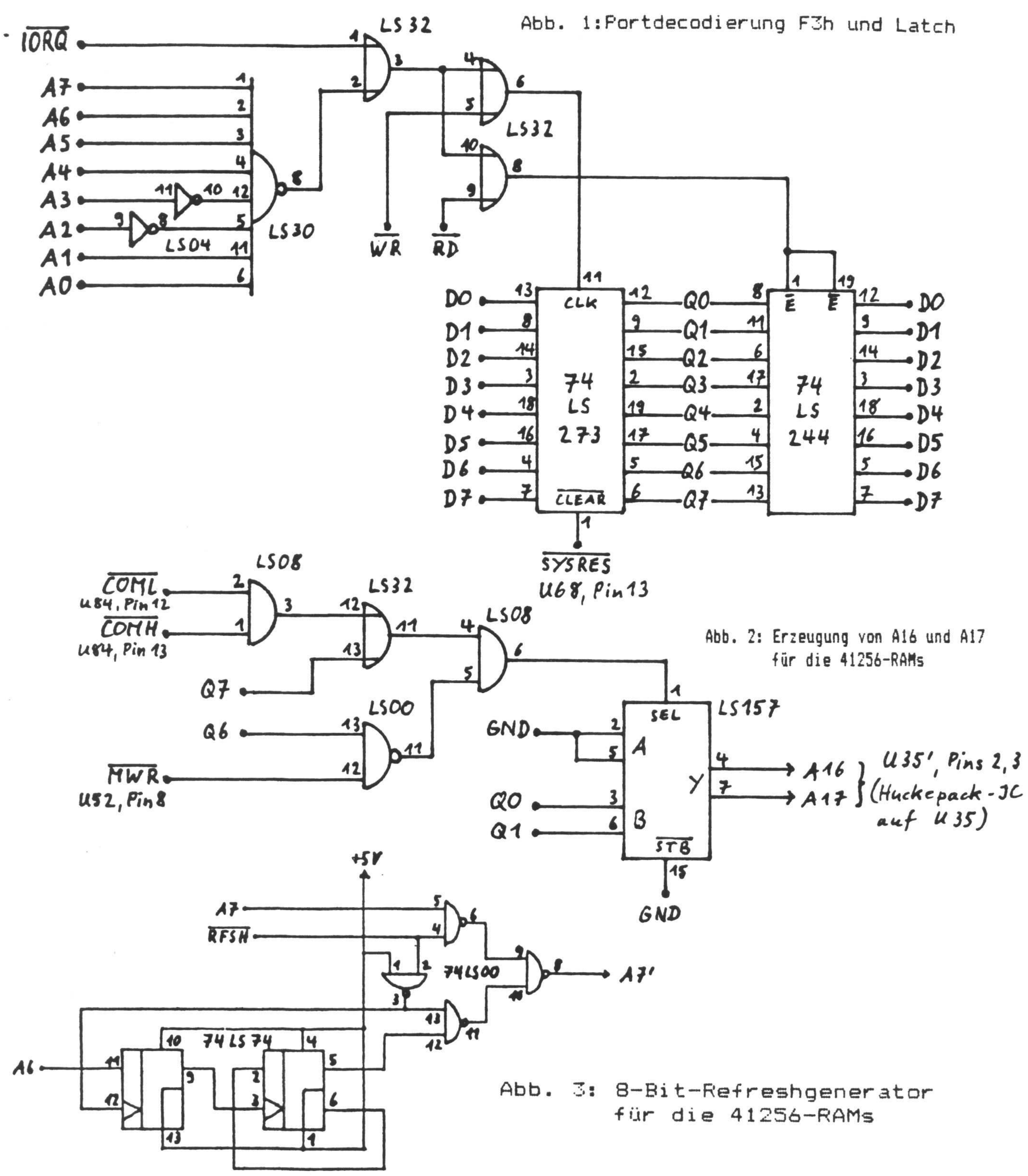


Abb. 1, 2 und 3

Dabei würden dann aber der Stack, Interrupt-Serviceroutinen und alles an Software abgekniffen werden, was für eine einwandfreie Funktion nötig ist. Der Z80 würde sich in der neuen Bank, wo noch kein Programm enthalten ist, verlaufen. Deshalb muß ein Teilbereich des Speichers auch beim Umschalten der Bank erhalten bleiben, wo der Z80 alles Lebenswichtige vorfindet.

Im Genie III s ist das mit vier Banks in Form von vier Reihen 4164-RAMs so gelöst, daß mit dem Zustand von Bit 0 des Ports F9h festgelegt wird, ob der Adreßbereich 0000h-3FFFh oder der Bereich E000h-FFFFh nicht mit umgeschaltet wird (Common-Bereich):

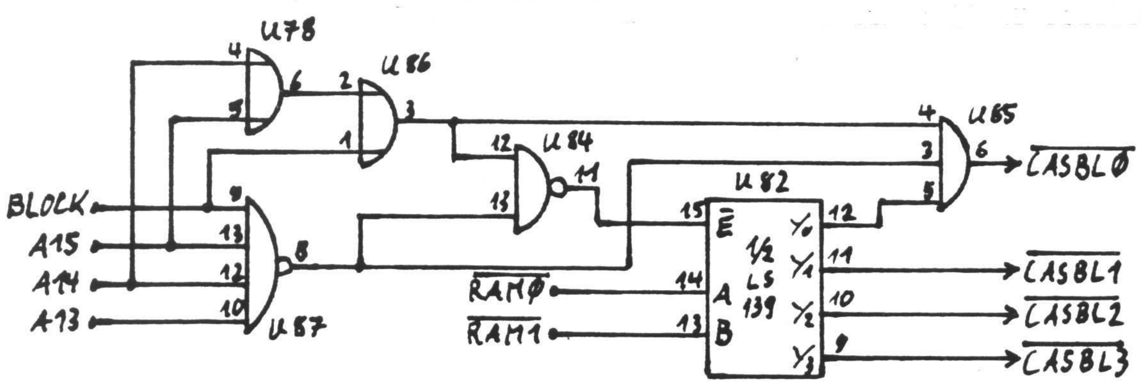
Port F9h, Bit 0:
- 0: Common = 0000h-3FFFh
- 1: Common = E000h-FFFFh

Die Auswahl unter den vier Banks geschieht mit den Bits 6 und 7 desselben Ports:

Port F9h, Bits 6, 7:
- 00: Bank 0
- 01: Bank 1
- 10: Bank 2
- 11: Bank 3

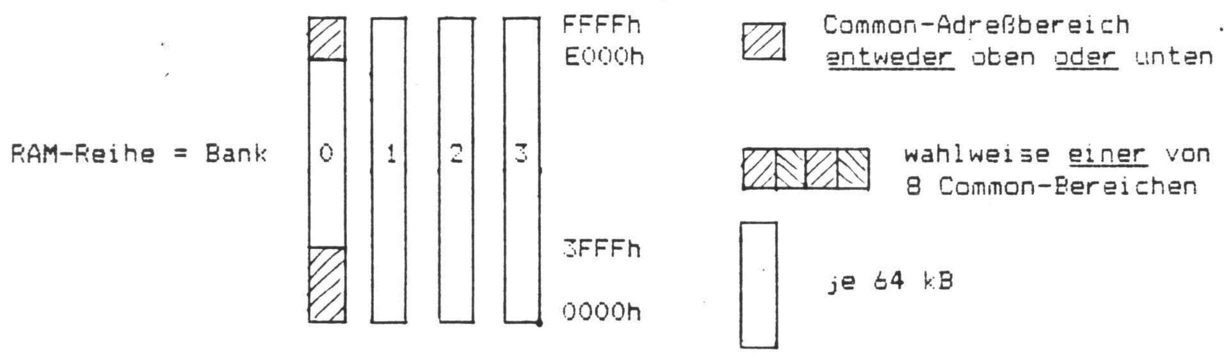
Unabhängig von der eingeschalteten Bank wird bei Zugriff auf den als Common definierten Bereich immer auf den RAM-Block 0 zurückgeschaltet. Das wird durch entsprechende Steuerung der CAS*-Freigabesignale für die vier RAM-Reihen erreicht (Abb. 4).

Abb, 4: CAS*-Freigabesignale der 4 64-kb-Blocks

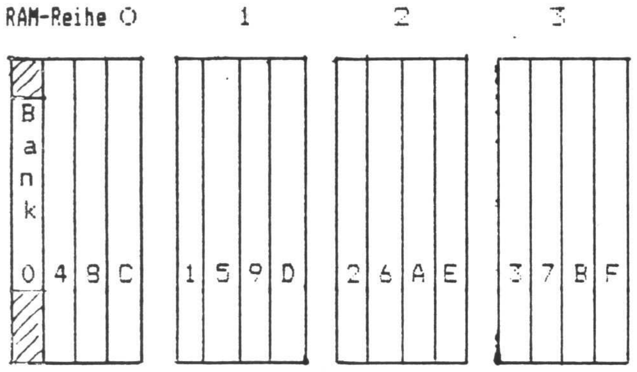


Zur Vorgabe der Adressen A16 und A17 für 41256-RAMs anstelle der ursprünglichen 4164-RAMs wurde ein unter der Portadresse F3h decodiertes Latch 74LS273 (Abb, 1) herangezogen, dessen Ausgänge Q0 und Q1 diese Adressen liefern. Damit läßt sich innerhalb eines 256-kB-Blocks (8 Stück 41256-RAMs) vorgeben, welches Viertel der RAM-Bausteine ansprechbar sein soll. Die RAM-Reihe selbst wird weiterhin über Port F9h, Bits 6 und 7 selektiert. Das Zusammenspiel beider Ports bei der Adressierung jeweils eines Blocks zu 64 kB veranschaulicht die Abb. 6.

Abb. 5: Die Speicherkonfiguration des Genie III s
a) im Originalzustand



b) 1 Computer mit 1 MB RAM



C) 4 Computer mit je 256 kB RAM

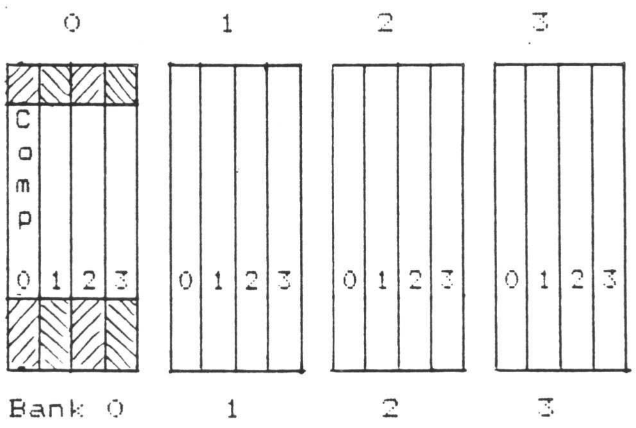
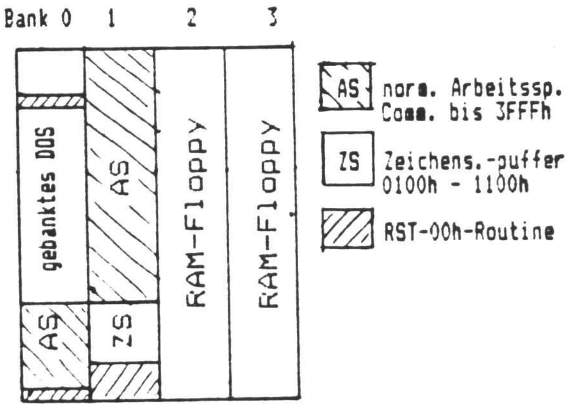


Abb. 6: Die Adressierung der 1664-kB-Blocks

Abb.6 : Die Adressierung der 16 64-kB-Blocks

Port F9h	....	00xxxxxx	01xxxxxx	10xxxxxx	11xxxxxx
Port F3h	xxxxxx00	0	1	2	3
	xxxxxx01	4	5	6	7
	xxxxxx10	8	9	A	B
	xxxxxx11	C	D	E	F

Abb. 7: Die Speicherbelegung pro Computer (auch im Originalzustand)



Das Umschalten auf die RAM-Reihe 0 beim Zugriff auf den als Common definierten Adreßbereich (Port F9h, Bit 0) leistet eine entsprechende Schaltung im GIIIs. Die Common-Bedingung wird auf Wunsch (s. u.) auch bei der Umschaltung innerhalb der Viertel der 41256-Chips beachtet: Die Selektion des unteren Viertels einer Reihe 41256-RAMs wird Uber die Steuerung des Pins 1 (Select) des 74LS157-Multiplexers in der abgebildeten Schaltung erreicht. Das wäre prinzipiell durch die AND-Verknüpfung der beiden low aktiven Signale COML* und COMH* (U84, 74LS00, Pins 12 und 13) möglich. Wenn eines dieser Signale low ist, wird Uber low am SEL-Eingang des Multiplexers GND auf die Ausgänge durchgeschaltet und nicht die Bits Q0 und Q1 des Latch. COML* und COMH* werden low, also aktiv, wenn der unten bzw. oben im 64-kB-Adreßraum als Common definierte Speicherbereich angesprochen wird.

Hier wurde aber zusätzlich die Möglichkeit geschaffen, über Q7 des Latch (Port F3h, Bit 7) die Steuerung des SEL-Eingangs des Multiplexers zu unterbinden. Wenn D7 des Ports F3h high ausgegeben ist (Bit im Output gesetzt), wird die automatische Umschaltung auf das untere Viertel einer 256K-RAM-Reihe unterbunden. Dann ist nur noch die ursprüngliche Steuerung der CAS*-Freigabesignale vom Zugriff auf den festgelegten Common-Bereich belangt (Selektion einer RAM-Reihe, aber nicht des unteren Viertels).

Das Verhalten des Computers läßt sich mit Q7=1 aus dem Normalverhalten (nur 1 aktiver Common-Bereich, 16 Banks) in einen zweiten Zustand umschalten: Es gibt dann 6 Blocks zu je 256 kB (Port F3h, Bits 0 und 1) mit je einem eigenen Common-Bereich an der durch Port F9h, Bit 0 definierten Adresse. Jeder dieser Blocks kann unabhängig hiervon je nach Port F9, Bits 6 und 7 eine der vier RAM-Reihen benutzen. Das entspricht der parallelen Existenz von vier Computern mit je vier Banks zu 64 kB.

In anderen Worten: Wahlweise stellt die hier vorgestellte Schaltung einem Computer 1 MB RAM zur Verfügung, oder vier voneinander unabhängige Computer verwalten je 256 kB RAM. Die zweite Variante entspricht dem Originalzustand des GIIIs - mal vier. Das Nebeneinander der nunmehr 16 Blocks zu je 64 kB ist in Abb. 5 b und c dargestellt.

Da das verwendete Latch 74LS273 aber 8 Bits verfügbar hat, sollte damit weiteres Tune-Up verwirklicht werden.

Der Bildwiederholspeicher und der Zeichengenerator sind 6116LP15-RAMs. Diese Einheiten können unter Verwendung von 6264LP15-Speichern ebenfalls vervierfacht

werden, wenn deren zusätzliche Adreßleitungen A11 und A12 mit den Ausgängen 02 und 03 des Latch beschaltet werden. Über D2 und D3 am Port F3h läßt sich einer von vier Bildwiederholspeichern mit eigenem Zeichengenerator-RAM selektieren.

Und schließlich sollte auch noch der HRG-Speicher von 64 auf 256 kB durch Austausch der 4164-RAMs gegen 41256-Typen auf acht Pages aufgebohrt werden. Die Selektion eines Viertels des neuen Speichers sollte über D4 und D5 am Port F3h erfolgen. Innerhalb des adressierten Viertels könnte dann weiterhin wie bisher eine von zwei Pages zu 32 kB gewählt werden. Die alte Software würde dadurch in ihrer Funktionsfähigkeit nicht beeinträchtigt werden.

Aus nicht verständlichen Gründen funktionierte das Umschalten bei einem Versuch nicht, Q4 und Q5 als Adressen A16 und A17 über einen Multiplexer 74LS157 (auch 74HC157) auf die neuen 41256-RAMs des HRG-Speichers zu legen. Sein Pin 1 wurde durch das gleiche Signal gesteuert wie die beiden schon existierenden Multiplexer. Evtl, kann die Verwendung eines S- anstatt des LS- bzw. HC-Typs das Problem lösen. Vielleicht kann uns jemand mit einem Tip weiterhelfen?

Auch eine andere Funktion, die nach Schaltplan einwandfrei arbeiten sollte, versagte den Dienst. Durch NAND-Verknüpfung von Q6 des Latch und MWR* sollte erreicht werden, daß bei high gesetztem Bit 6 im Port F3h beim Lesen des Speichers auf Computer 0 geschaltet wird, und nur beim Schreiben der RAM-Speicher des mit DO und D1 festgelegten Computers angesprochen wird. Damit wäre es besonders einfach geworden, den Speicherinhalt von Computer Nr. 0 auf seine drei Nachbarn zu übertragen, um sie mit Initial-Information zu laden. Das hätte der Fähigkeit des EG 64 MBA entsprochen, mit einem einfachen LDIR den ROM-Inhalt auf das parallele RAM zu kopieren. Auch dieses Feature funktioniert wider Erwarten nicht.

Im Layout wurden trotzdem die Schaltungen dieser nicht funktionie- renden Elemente beibehalten. Evtl, gelingt es noch, diese Features später zum Laufen zu kriegen. Ansonsten steht einer anderweitigen Verwendung der Latch-Ausgänge Q4, Q5 und Q6 nichts im Wege. Die im folgenden beschriebene Software kommt ohne diese Funktionen aus. Die nicht funktionsfähigen Teile der Schaltung haben insb. auch keinen negativen Effekt auf sie, so daß sie im Hinblick auf späteres Gelingen durchaus bereits bestückt werden können.

Die drei Gruppen zu je zwei Bits am Port F3h sind voneinander unabhängig ansprechbar. So kann z.B. über D2 und D3 zwischen mehreren Bildschirmen und Zeichensätzen gewählt werden, D4 und D5 können im Graphikspeicher einen Spooler mit luxuriöser Kapazität zur Verfügung stellen, während D0 und D1 ständig low bleiben, also gewissermaßen im Computer Nr. 0 gearbeitet wird. Und Computer Nr. 3 kann eine Graphik ändern, die Computer Nr. 1 erstellte usw..

B: Die Software zur Umschaltung zwischen vier Rechnern

Während dieser Text entsteht, lauert im selben Gehäuse, aber dennoch in einem zweiten Computer ZEUS mit dem Quelltext des Programms, das nun vorgestellt werden soll. So kann während der Arbeit mal eben hinübergeschaltet werden, um vergessene Einzelheiten ins Gedächtnis zurückzurufen. Und damit ist auch schon gesagt, was das Ganze soll. Wir sozial Schwachen, die sich nur einen Rechner leisten können, brauchen auf den Komfort mehrerer Maschinen nun nicht mehr zu verzichten.

Da gibt es Blocks, RAM-Reihen, Banks, Adreßräume usw.. Um die Verwirrung in Grenzen zu halten, schlagen wir folgende Sprachregelung vor: Ein Block ist eine bestimmte Datenmenge, die gleichzeitig bearbeitet wird. Eine RAM-Reihe sind acht Speicherchips 41256, die über Port F9h, Bits 6 und 7 gleichzeitig selektiert sind. Funktional gesehen ist es sinnvoll, sie als Banks zu bezeichnen. Die Speicherviertel innerhalb einer Reihe von 41256-RAMs, um die es in diesem Beitrag geht, sollen hier Computer genannt werden, denn das Programm verwirklicht das Konzept von vier Rechnern, die einander auf Knopfdruck abwechseln.

Dieser Knopfdruck sollte nicht aus Versehen erfolgen können. Deshalb wird mit der Kombination von Shift-CTRL und einer Taste von 0-3 im Ziffernblock umgeschaltet. Dazwischen liegen rd. 35 cm, so daß Zufälle ausgeschlossen sind. Gleichwohl wäre ein solches Versehen nicht tragisch, denn eine Umschaltung ist sowohl mit, wie auch ohne Absicht erlaubt und ungefährlich. Alle Speicherinhalte bleiben bestehen, sogar die restliche Blinkzeit eines Cursors wird auf die Nanosekunde genau beim Zurückschalten nachgeholt. Der Tastendruck wird während der Interrupt-Bearbeitung erkannt. Hierzu wird ein CALL nach 45BEh (Erkennen der Dreitastenbefehle) an der Stelle 4601h umgeleitet (Z. 490 im Listing).

Das alte Lied: Wohin mit Erweiterungen des Betriebssystems? Diesmal mußte ein Speicherbereich dran glauben, der von TCS bei der Modifikation des Microsoft-Interpreters offenbar vergessen wurde. Ab 0000h wird weder nach dem Einschalten noch bei einem Reset gelesen, denn dann ist das Boot-EPROM aktiviert. In diesem Speicher-Niemandsland wird vom Programm erkannt, ob Shift-CTRL-Ziffer gedrückt wurde. Dazu stehen aber nur 8 Bytes bis zum Beginn der RST-08h-Routine zur Verfügung. Deshalb verabschiedet sich das neue RST 00h an der Stelle 0007h kurzerhand aus der Bank 0 und macht in Bank 1 weiter, wo das Betriebssystem erst wieder bei 0100h (Zeichensatz-Puffer) Platz beansprucht.

Diese Umschaltung geschieht mit dem Bit 0 von Port F9h. Da die normale Arbeitsbank Bank 1 ist, wobei Common unten in Bank 0 liegt, ist der Trick simpel: Das Common-Bit wird auf 1 gesetzt, so daß die CPU, ohne es zu merken, nun auch unten in Bank 1 liest, denn: Common, d. h. der Bereich, wo immer in Bank 0 gearbeitet wird, liegt jetzt ab E000h. Im Listing hat dieses Programmsegment das Label rst00 (Z. 664). Es liegt sowohl in Bank 0 als auch in Bank 1 bereit, so daß die Umschaltung aus beiden Banks auf die gleiche Weise geschieht. Es wird einfach D0 des Ports F9h invertiert.

Beim Umschalten aus Bank 0 (Common unten) wird die scheinbar nahtlose Fortsetzung in Bank 1 gefunden (Label bank1). Beim Zurückschalten aber muß rechtzeitig vor dem Erreichen der anschließenden RST-08h-Routine zur Fortsetzung der INT-Serviceroutine gesprungen werden. Deswegen wurde zuvor in Bank 1 HL mit 45BEh geladen, der Adresse, wo es weitergehen soll (s. o.). In Z. 495 wurde zu diesem Zweck der Befehl JP (HL) (Code E9h) in Bank 0 ans Ende dieses Segments gepatcht.

Nachdem HL die Adresse der Fortsetzung enthält, wird die Tastatur in 3880h und 38C0h nach den Tasten Shift und CTRL sowie nach den Ziffern 0-3 im Dezimalblock befragt. Wenn diese Tastenkombination nicht gedrückt wurde, will der User offenbar nicht den Computer wechseln. Es erfolgt ein Rücksprung nach 0000h, wo wieder die alte Bank-Konfiguration eingestellt wird. Es ist, als wäre nichts geschehen.

Wurde aber Shift-CTRL-Ziffer eingegeben, wird zunächst der Zifferncode in einen Zähler für die gewünschte Computer-Nr. umgerechnet und nach B geladen. Da sich die vier Computer außer der CPU natürlich auch alle Ports teilen müssen, werden die Zustände der beiden einzig wirklich lebenswichtigen, FAh und F9h, zwischengepuffert, jener sofort, dieser etwas später. Bei dieser Gelegenheit wird auf FAh das Bit 5 gesetzt, um einen Schreibschutz für den Adreßbereich des ehemaligen TRS-80-ROMs zu erreichen. Es könnte ja sein, daß ein Programm ihn als Sektorpuffer o. ä. benutzt.

In der Folge bekommt die Umschaltroutine auch ihren eigenen Stack. Das muß sein, weil er z. B. bei BASIC im Himem liegt, wo ja jetzt die Bank 0 selektiert ist; der Stack wäre unerreichbar! Der alte Stapelzeiger wird gepuffert. Je nach dem kurz zuvor aufgesetzten Computer-Zähler wird so oft zum Ausgangswert -15h (=+EBh) 15h addiert, bis die Nr. des zuständigen Computers im Akku steht.

Die Zahl 15h ergibt sich folgendermaßen: Beispielsweise bei der Selektion von RAM 1, Bildschirm und Zeichensatz Nr. 1 und Graphikspeicher Nr. 1 liegt am Port F3h der Wert xx01.0101b an (xx sind die beiden Steuerbits, die keinem eigenen Computer zugeordnet sind). Abgesehen von den Bits 6 und 7 ist das die Zahl 15h. Um einen Computer höher zu gehen, wird deshalb 15h addiert. Der so gewonnene Output für den Port F3h wird an die Adresse mit dem Label newcomp gepatcht.

Jetzt müssen alle Register gerettet werden, weil die benachbarten Computer nicht jeder eine eigene CPU haben. Die INT-Routine hat die Hälfte dieser Arbeit schon erledigt, so daß nur noch die Indexregister und der alternative Registersatz gepushed zu werden brauchen.

Nun kann umgeschaltet werden. Das Bit 7 wird zuvor gesetzt, damit bei Common-Einstellung nicht in die Bank 0 des Computers Nr. 0 zurückgeschaltet wird. Modifikationen, die der User oder das System dort vorgenommen hat, sollen ja nicht für alle vier Computer gelten.

Nach dem Umschalten auf einen Nachbarcomputer wird alles wieder re- stauriert. Es beginnt mit dem Output auf den Port F9h, der aber zunächst noch in E zwischengelagert wird. Zur Restauration des Bildschirmformats (auch den Video-Controller gibt es nur einmal) müssen nämlich noch im oberen Common-Bereich, der ja

z. Zt. noch selektiert ist, die CRTC-Parameter aus 37F0h-37FFh ausgelesen und auf den Controller ausgegeben werden.

Diese Video-Codes liegen nun wiederum im unteren Common. Also hilft nur eine Umschaltung auf Bank 0, die beide Common-Bereiche gleichzeitig zugänglich macht. Daher der Umweg Uber das Unterprogramm vidpar, das die komplette Bank 0 verwalten kann. Zur Erinnerung: Der Arbeitsspeicher ist die Bank 1, daher diese Umstände, vidpar macht sich die Arbeit leicht, indem es ein fertiges Unterprogramm des Betriebssystems ab 3551h aufruft, das die Codes an den Video-Controller ausgibt. Zuletzt schaltet es wieder auf Bank 1, Common oben und kehrt zurück.

Jetzt werden auch der Port FAh, die CPU-Arbeitsregister und der Stackpointer restauriert. Wohlgemerkt sind es die Registerinhalte, die der Nachbarcomputer irgendwann zuvor für sich selber auf den eigenen Stack der Umschaltroutine gelegt hat. Was soeben gerettet wurde, liegt im alten Computer sicher. Mit dem abschließenden Sprung nach 0000h (Label rst00) wird Common wieder nach unten verlegt und die INT-Routine - diesmal diejenige des neuen Computers - zuende bearbeitet.

Außer der Umschaltung mit dem komplizierten Tastendruck geht es auch durch Aufruf des Programms mit seinem Namen. Um dabei nicht jedesmal die Floppy rotieren zu lassen, ist es hier eine Systemdatei, so daß sie gleich beim Booten in den Speicher geladen wird. Sie erledigt verschiedene Aufgaben:

Bevor die Umschaltroutine arbeiten kann, sind etliche Initialisierungen notwendig. Hauptsächlich muß natürlich der Computer Nr. 0 auf die drei Nachbarn kopiert werden, damit die CPU nach dem Umschalten sofort sinnvollen Code lesen kann. Bei Testläufen hat es sich erwiesen, daß nur vom Computer 0 aus die Kopie stattfinden darf. Es könnte ja sein, daß der User ein wenig damit herumexperimentieren möchte oder sein digital bereits ebenso verseuchtes Kind nicht weiß, daß Mama oder Papa schon den Rechner wechselte. Deshalb wird bald nach dem Einsprung (Label test0) überprüft, ob die Bits 0 und 1 unseres neuen Systembytes 2 (zuständig für den Hauptspeicher) beide auf 0 stehen. Ist das nicht der Fall, wird einfach nur der derzeit selektierte Computer angezeigt. Diese Anzeige kann auch mit einem dem Aufruf folgenden Fragezeichen erhalten werden.

Ein weiteres Argument kann dem Befehl folgen: Wenn eine Ziffer von 0-4 eingegeben wurde, wird der so angewählte Computer selektiert. Allerdings nur dann, wenn zuvor die Initialisierung stattgefunden hat. Das merkt das Programm daran, ob an der Stelle 0000h in der Bank 1 bereits der Befehl DI steht. Ist das nicht der Fall, wird mit einer Fehlermeldung das Unternehmen abgebrochen. Wenn aber alles klar ist, ergibt die eingegebene Ziffer bereits die gewünschte Computer-Nr.. Es kann daher in Bank 1 an der Stelle switch mitten in die RST-00h-Routine gesprungen werden, wobei die Ermittlung des Computers und - sehr wichtig! - die Tastaturabfrage übersprungen werden.

Zusätzlich gibt es folgende Argumente: Wie in G-DOS üblich, kann das Ein- und Ausschalten des Features der vier Computer mit J (für "ja") und N (für "nein") erfolgen. J wird auch angenommen, wenn kein Argument folgt. N kann in jedem der vier Computer eingegeben werden. Dann bleibt immer dasselbe Viertel der 41256-RAMs selektiert, um in diesem Computer beispielsweise eine unverfälschte INT-Routine zu Testzwecken zu haben. Dennoch kann mit SYS,BK,n (n = Nr. des Computers) umgeschaltet werden; nur die Tastenkombination Shift-CTRL-n ist deaktiviert. Und dies gilt auch nur für den gerade selektierten Computer. Mit R (für "recover") werden die vier Computer reaktiviert, wenn ein zwischenzeitliches Reset oder ein SYS,BK,N es abschaltete. Auch hier muß zuvor bereits die Ini tialisierung erfolgt sein, sonst wird ein Fehler gemeldet.

Falls die Initialisierung der Nachbarcomputer gewünscht wird und erlaubt ist, wird's jetzt hektisch: Im Verlauf der nächsten Mikrosekunden werden der Stackpointer, das Systembyte 0 (Port F9h) und das Systembyte 1 (Port FAh) als Initialwerte in die RST-00h-Routine gepatcht. Deren kürzerer Teil, der in Bank 0 liegen soll, wird nach 0000h verschoben. Das bereits erwähnte Byte E9h für JP (HL) kommt dahinter. Durch verschiedene Manipulationen der beiden Systembytes 0 und 1 werden Bildschirme aus- und wieder angeknipst, expandiert und wieder komprimiert, Tastaturen abgewürgt, Zeichensätze freigegeben, Common-Bereiche umdefiniert und dergleichen mehr. Es würde zu weit führen, das hier im einzelnen zu erläutern. Die Kommentare im Listing protokollieren es jeweils. Eine Hilfe zum Verständnis kann auch die Tabelle am Ende dieses Abschnitts geben.

Die Kopie des Hauptspeichers kann nicht auf einen Schlag geschehen. Das liegt daran, daß jeweils ein Common-Bereich erforderlich ist, der immer in der Bank 0 arbeitet. Das Feature, das mit Q6 des Latches ermöglicht werden sollte, funktionierte ja leider nicht. Soll also beispielsweise im Common-Adreßraum gerade die Bank 2 kopiert werden, ist sie einfach nicht da. Daher wird das Unterprogramm COPY, das später erklärt werden soll, sowohl in das obere als auch in das untere Common kopiert. Die beiden Routinen bearbeiten dann jeweils nur eine Speicherhälfte von viermal 32 kB. Mit dem oben residierenden Kopierprogramm wird gleich VIDPAR mitverschoben.

Zuerst wird der Bildschirm vervierfacht. Dazu muß er zunächst in einen Puffer im unteren Adreßraum der Bank 1 kopiert werden. Er findet Platz hinter dem Zeichensatzpuffer ab 1100h. Beim Schreiben des gewählten Zeichensatzes in die drei benachbarten Zeichengeneratoren steht das Adreß-LSB des gerade bearbeiteten Zeichens im Bildschirm an der Stelle 3C00h. Um den dadurch veränderten Bildschirm nicht umständlich restaurieren zu müssen, erfolgt zuerst die Kopie des Zeichensatzes, erst anschließend wird der Bildschirminhalt aus dem Puffer in den Bildwiederholspeicher geladen. Das Zeichen von 3C00h wurde im Puffer ja nicht verändert. Die Auswahl des gerade aktuellen Bildschirms bzw. Zeichensatzes geschieht Über die Bits 2 und 3 des Ports F3h.

Um die beiden Kopierroutinen aufrufen zu können, ist ab Z. 576 wieder einiges Banking erforderlich. Auch hier mag der Verweis auf die Kommentare und die Tabelle ausreichen. Die Routinen selbst arbeiten folgendermaßen:

Am Label copy wird ein Zähler mit 32 aufgesetzt. Die Datenmenge, für die der erforderliche Puffer ausreicht, beträgt 1 kB. 32 kB pro Bank übernimmt eins der beiden Kopierprogramme. Gleich anschließend wird dieser Zähler gerettet und derjenige für die Banks geladen. Jetzt wird aus jeder Bank des Computers Nr. 0 ein Block zu 1 kB in den Puffer geholt. Schließlich wird in der inneren Schleife, welche die drei Nachbarcomputer abzählt, der Pufferinhalt auf die Nachbarn kopiert. Um die Adresse des zu bearbeitenden 1-kB-Blocks zu aktualisieren, genügt es, ihr MSB um 4 zu erhöhen (ab Z. 654), denn 0400h = 1 kB.

Um die Adressen beim Blocktransferbefehl LDIR nicht zu zerstören, müssen sie fortlaufend gerettet und wieder restauriert werden. Neben den LDIR-Befehlen selbst sind es diese PUSHes und POPs, die beim Kopieren die meiste Zeit verbrauchen. Insgesamt läuft das Programm deshalb etwa dreieinhalb Sekunden, bei immerhin einem Megabyte eine erträgliche Zeit. Um möglichst schnelle 8-Bit-Befehle anwenden zu können, werden nur die MSBs geladen. Die LSBs betragen immer 00h, so daß sie nicht berücksichtigt werden müssen. Diese - im Computer-Maßstab - extrem langwierige Prozedur läuft jedoch nur nach dem Aufruf des Programms ab. Nach der Initialisierung geschieht das Umschalten zwischen den Computern hundertmal schneller als ein Blinzeln.

Nach den Kopiervorgängen bleibt nur noch übrig, die Systembytes 0 und 1 sowie den Stackpointer zu restaurieren. Mit RET wird ins DOS zurückgekehrt.

Wie gesagt, müssen sich die vier Computer die gesamte Hardware teilen, lediglich die Speicherbausteine haben sie für sich. Der User muß deshalb wissen, daß gewisse Einschränkungen in Kauf zu nehmen sind. So ist es beispielsweise nicht möglich, mit der seriellen Schnittstelle auf Computer Nr. 0 eine Mailbox anzurufen und zwischendurch damit einen seriellen Drucker zu bedienen. Die Schnittstelle wird beim Umschalten nicht berücksichtigt. Dasselbe gilt für die Floppies, die PIO und evtl, angeschlossene Peripherie auf dem Systembus. Dennoch können alle Bauteile von allen Computern benutzt werden. Es darf eben nur nicht umgeschaltet werden, wenn sich z. B. gerade die Floppy dreht.

In der hier vorgestellten Form ist das Programm Teil der Systemdatei SYS25/SYS. Dies erschien sinnvoll, weil es ganz entschieden ins Betriebssystem gehört und notfalls auch unter Mini-DOS mitten in einem Programmablauf abrufbar sein sollte. Da die Library des Genie III s schon voll ist, wird es mit dem SYS-Befehl (hier SYS,BK für "BANK") aktiviert, der an dieser Stelle bereits vorgestellt wurde. So erklärt sich das Vorgeplänkel am Beginn des Listings, wo zum alten Inhalt von SYS25/SYS (Bedienung der F-Tasten) der Test auf den neuen Requestcode DBh für SYS,BK eingefügt wurde. Was zwischen den Zeilen 27 und 405 steht, ist mit LIST OFF ausgespart, denn die F-Tasten sind hier nicht unser Thema. Dies gilt auch für die Label-Definitionen bis Z. 22.

Es gibt keine Einschränkungen des Speicherplatzes. Nach dem Laden steht SYS25/SYS zunächst im DOS-Overlay-Bereich ab 4D00h, der für diese Zwecke vorgesehen und reserviert ist. Die Bereiche, die die RST-00h-Routine und das Unterprogramm vidpar belegen, sind völlig frei und werden weder vom Betriebssystem noch von den

Autoren bekannten Anwenderprogrammen beansprucht. Für das RAM-Floppy-Programm MEMDISK/CMD gilt sogar, daß es vor dem Aufruf bereits aktiviert sein darf. Es wird nicht Uberschrieben, auch nicht die Dateien, die in der RAM-Floppy stehen. Alles wird einfach in die Nachbarcomputer hinüberkopiert und kann auch von dort aus benutzt werden.

Wie sich herausstellte, kann aber leider nicht unter BASIC CMD'SYS,BK<,J>' eingegeben werden, um BASIC/CMD solo oder gar mit bereits geladenem Programm in die drei Nachbarcomputer hinüberzukopieren. Vielleicht gelingt es in der Folge, das ebenfalls noch zu ermöglichen. Einstweilen muß es unter DOS geschehen, wonach dann in jedem der vier Computer BASIC/CMD gesondert gestartet werden muß. Auf jeden Fall aber kann nach erfolgter Initialisierung mitten im Lauf eines BASIC-Programms auf einen Nachbarcomputer umgeschaltet werden. Mit anderen Programmen sind bisher diese Schwierigkeiten nicht aufgetreten. Im Zweifelsfall hilft immer die Methode, die eben für BASIC beschrieben wurde.

Eine weitere Besonderheit muß beim Betrieb mit BASIC beachtet werden. In der RST-00h-Routine kehrt der Akku mit einem Wert >0 zurück (Output für den Port F9h). Das Z-Flag ist rückgesetzt, d. h. die NZ-Bedingung ist gegeben. BASIC versteht das als DOS-Fehler, gibt eine entsprechende Meldung aus und sagt ‘READY’. Deshalb kann man zwar ohne weiteres den Befehl CMD"SYS,BK,n' (n ist die Computer-Nr.) über die Tastatur eingeben, wenn es aber innerhalb eines Programms geschieht, wird dieses wegen des angeblich auf getretenen Fehlers abgebrochen. Das wäre mit geringen Modifikationen zu beheben gewesen, die aber Speicherplatz im kostbareren oberen Adreßraum gekostet hätten. So wurde darauf verzichtet, denn eine Umschaltung, die ein BASIC-Programm selbsttätig steuert, ist praxisfern.

Das Verändern von SYS-Files und das Implementieren eines neuen Library-Befehls (auch in der Zweit-Lib mit SYS) wurde hier schon mehrfach gezeigt. Es soll nicht erneut erklärt werden. Wer sich scheut, auf diese Weise ins Betriebssystem einzugreifen, kann das Programm in der Zeile 421 mit ORG 5200h beginnen und ein CMD-File daraus machen. Dann muß das END-Statement die Startadresse sidekck haben. Ansonsten ändert sich dabei nichts.

Zuletzt sollen noch diejenigen Datenbits der hier erwähnten Ports erläutert werden, die für das Programm von Belang sind:

```
      Bit 7    6     5    4     3     2     1     0
-----------------------------------------------------------------------
F3h   Common           =HRG-Nummer=     =Bildsch. +=     =Haupt-Sp. Nr=
      Defin.                       Zeichens NR=
-----------------------------------------------------------------------
F9h   =Orig. Banks=                            Zeichen- Common
                                          Satz   Defin
-----------------------------------------------------------------------
FAh                Schreib-  Video ab                mm-I/O
                   schutz    3800h                   an/aus
-----------------------------------------------------------------------
```

Sollte es mit der Software zur Verwirklichung von 4 Computern in einem Schwierigkeiten geben, kann Arnulf Sopp weiterhelfen.

```
4D00        00023       ORG   4d00h
4D00  FEFB  00024 sys25 CP    0fbh        ;Lib-Befehl F# ?
4D02  2822  00025             JR    Z,fkey           ;falls ja
4004  FEDB  00026       CP    0dbh        ;Lib-Befehls SYS,BK ?
4D06  CA2A50      00027             JP    Z,sidekck  ;falls ja
            00405
            00406
            00407 ;***************************************************
            00408 ;
            00009 ;            Sidekick für das Genie IIIs
            00010 ;            4 Computer in einem Gehäuse
            00411 ;
            00412 ;        Hardware: 1 MB RAM, 8 kB Bildschirm,
            00413 ;        16 kB Zeichensatz, 256 kB Graphik
            00414 ;
            00415 ;        (C) 1986 by Arnulf Sopp
            00416 ;        starring:   Helmut Bernhardt
            00417 ;
            00418 ;***************************************************
            00419
            00420 ;Test, ob vier Computer bereits initialisiert sind
5015  DBF9  00421 tstinit     I     A,(0f9h)   ;Systembyte 0
5017  57    00422       LD    D,A         ;retten
5018  F601  00423       OR    01h         ;unten Bank 1 selektieren
501A  F3    00424       DI                ;vorsichtshalber
501B  D3F9  00425       OUT   (0f9h),A
501D  3A0000      00426       LD    A,(0000h)  ;Anf. der RST-00-Routine
5020  FEF3  00427       CP    0f3h        ;steht dort bereits DI ?
5022  7A    00428       LD    A,D         ;Systembyte 0
5023  D3F9  00429       OUT   (0f9h),A    ;restaurieren
5025  C8    00430       RET   Z           ;o. k., falls schon DI
5026  E1    00431       POP   HL          ;RET-Adresse vom Stack
5027  C3224D      00432 jperr JP    error       ;raus mit Fehlerbeding.
            00433
            00434 ;Tests, ob Befehl mit Argument eingegeben wurde
502A  DBF3  00435 sidekck     IN    A,(0f3h)   ;Systembyte 2
502C  E603  00436       AND   03h         ;nur Hauptspeicher-Bits
502E  4F    00437       LD    C,A         ;retten
502F  CDD54C      00438       CALL  4cd5h       ;folgt etwas dem Befehl?
```

```
5032  2841         JR    Z,test0          ;falls nein
5034  7E           LD    A,(HL)           ;was folgt?
5035  FE52         CP    'R'        ;Recover 4 Computer?
5037  200A         JR    NZ,norecov ;falls nein
5039  CD1550            CALL  tstinit          ;schon initialisiert?
503C  3EC9         LD    A,0c9h           ;RET-Befehl
503E  329150            LD    (retrecv),A ;dort patchen
5041  1838         JR    init       ;dort weiter
5043  FE4A   norecov CP 'J'        ;'ja' für initialisieren?
5045  282E         JR    Z,test0          ;falls ja
5047  FE4E         CP    'N'        ;'nein' für abschalten?
5049  2007         JR    NZ,notoff  ;falls nicht
504B  21BE45            LD    HL,45beh   ;alte Adr. in   INT-Routine
504E  220146            LD    (4601h),HL ;dort restaurieren
5051  C9           RET              ;dann Ende
5052  FE3F   notoff      CP    '?'        ;Status-Abfrage?
5054  2819         JR    Z,dspcmp1  ;falls ja
5056  FE30         CP    'O'        ;Computer-Nr.eingegeben?
5058  38CD         JR    C,jperr          ;falls falsche  Eingabe
505A  FE34         CP    '4'        ;nur Comp. Nr.  0-3
505C  30C9         JR    NC,jperr   ;falls falsche  Eingabe

       ;neuen Computer selektieren, falls schon initialisiert
505E  E603         AND   03h        ;ASCII -> binär 0-3

5060  3C           INC   A          ;für Zähler B:  1-4
5061  47           LD    B,A        ;Computerzähler aufsetzen
5062  CD1550            CALL  tstinit          ;schon initialisiert?
5065  E1           POP   HL         ;RET-Adresse
5066  DBF9         IN    A,(0f9h)   ;Systembyte 0
5068  F601         OR    01h        ;Common oben, Bk. 1 unten
506A  D3F9         OUT   (0f9h),A   ;Systembyte neu ausgeben
506C  C31E00            JP    switch-rst00     ;dort Computer umschalten

       ;aktuelle Computer-Nr. anzeigen
506F  79     dspcmp1     LD    A,C        ;Computer-Nr.
5070  F630     dspcmp2  OR    30h        ;binär -> ASCII
5072  C33300            JP    0033h      ;anzeigen und raus

       ;falls Comp. Nr. 0, dann initialisieren, sonst anzeigen
5075  79     test0 LD    A,C        ;Computer-Nr.
5076  B7           OR    A          ;Nr. 0 selektiert?
5077  20F7               JR    NZ,dspcmp2 ;nur anzeigen, falls nein
5079  D3F3         OUT   (0f3h),A   ;auch Video Nr. 0 usw.

       ;INT-Patch in SYS0, RST-00-Routine für Bank 0 verschieben
507B  ED73D751     init  LD    (spbuf1),SP ;Initialstack
507F  216B51            LD    HL,rst00   ;RST-00h-Routine
5082  010700            LD    BC,bank1-rst00   ;Länge der Rout, in Bk. 0
5085  50           LD    D,B        ;DE <- Zieladresse 0000
5086  58           LD    E,B
5087  F3           DI               ;DISABLE INTERUPT
5088  ED530146          LD    (4601h),DE ;CALL 0000 statt 45BE
508C  EDB0         LDIR             ;RST-00-Routine -> Bank 0

       ;Befehl zum Weitersprung schließt RST-00-R. in Bank 0 ab
508E  3EE9         LD    A,0e9h           ;Befehl JP (HL)
5090  12           LD    (DE),A           ;dahinter patchen (0007h)

       ;Rout. zur Kopie der ob. Speicherhälfte nach 3800 laden
5091  DBFA   retrecv     IN    A,(0fah)   ;Systembyte 1
5093  F5           PUSH  AF         ;retten
5094  32C851            LD    (fabuf),A  ;Patch in RST 00, Bank 1
5097  F611         OR    11h        ;Großbildsch. , mm-I/O aus
5099  D3FA         OUT   (0fah),A   ;Systembyte 1 neu ausgeb.
509B  08           EX    AF,AF'           ;nochmals retten
509C  213751            LD    HL,copy          ;Kopierprg. f. ob. Hälfte
509F  110038            LD    DE,combot  ;im unt. Common-Bereich
50A2  0E34         LD    C,rst00-copy     ;Länge der Routine
50A4  EDB0         LDIR             ;nach 3800 übertragen

       ;Routine zur Restauration der Bildschirmparameter und
       ;zur Kopie der unteren Speicherhälfte nach F600 laden
50A6  DBF9         IN    A,(0f9h)   ;Systembyte 0
50A8  F5           PUSH  AF         ;retten
```

```
50A9  F601  00513       OR    01h         ;Common oben, Bank 1 unt.
50AB  F5    00514       PUSH  AF          ;nochmals retten
50AC  32BD51      00515       LD    (f9buf),A  ;Patch in RST 00, Bank 1
50AF  D3F9  00516       OUT   (0f9h),A    ;geänd. Systemb. ausgeben
50B1  212E51      00517       LD    HL,vidpar  ;CRTC-Restaur., Kopierr.
50B4  1100F6      00518       LD    DE,comtop  ;im oberen Common-Bereich
50B7  0E3D  00519       LD    C,rst00-vidpar   ;Länge der Routine
50B9  EDB0  00520       LDIR              ;nach F600 kopieren
            00521
            00522 ;RST-00-Routine für Bank 1 in diese Bank nach 0000 laden
50BB  50    00523       LD    D,B         ; DE <- 0000h, wie oben
50BC  58    00524       LD    E,B   ;(Zieladr. RST-00-Rout.)
50BD  0E70  00525       LD    C,endrst00-rst00    ;Länge RST-00-R.
50BF  EDB0  00526       LDIR                      ;RST-00-Routine -> Bank 1
            00527
            00528 ;memory-mapped I/O einschalten, Zeichen-RAM freigeben
50C1  F602  00529       OR    02h         ;Zeichensatz freigeben
50C3  D3F9  00530       OUT   (0f9h),A
50C5  08    00531       EX    AF,AF'            ;geändertes Systembyte 1
50C6  F5    00532       PUSH  AF          ;wird noch gebraucht
50C7  E6FE  00533       AND   0feh        ;mm-I/O einblenden
50C9  D3FA  00534       OUT   (0fah),A    ;Systembyte 1 neu ausgeb.
            00535
            00536 ;Großbildschirm ab 3800 in Bank 1 ab 1100 zwischenspeich.
50CB  210038      00537       LD    HL,3800h    ;Anfang Bildschirm
50CE  110011      00538       LD    DE,1100h    ;Anfang Puffer
50D1  0608  00539       LD    B,08h       ;BC <- Länge 2 kB
50D3  EDB0  00540       LDIR              ;Bildschirm puffern
            00541
            00542 ;Zeichensatz in 3 neue Zeichengeneratoren kopieren
            00543 ;(3 obere Viertel der beiden 6264)
50D5  0603  00544        LD   B,03h       ;3 Bildsch. u. Zeichens.
50D7  C5    00545 vidloop    PUSH  BC          ;Bildschirmzähler retten
50D8  78    00546       LD    A,B         ;Bildschirm-Nr.
50D9  07    00547       RLCA              ;Bit 0-1 -> 2-3
50DA  07    00548       RLCA
50DB  D3F3  00549       OUT   (0f3h),A    ;Bildschirmseite selekt.
50DD  2601  00550       LD    H,01h       ;HL <- Beginn Zeichensatz
50DF  AF    00551       XOR   A           ;ab ASCII 00
50E0  32003C      00552 chrlop1    LD    (3c00h),A  ;LSB Zeichenadresse
50E3  08    00553       EX    AF,AF'            ;retten
50E4  1684  00554       LD    D,84h       ;MSB in D
50E6  0610  00555       LD    B,10h       ;16 Bytes pro Zeichen
50E8  7E    00556 chrlop2    LD    A,(HL)           ;1 Byte aus dem Puffer
50E9  12    00557       LD    (DE),A            ;ins Zeichensatz-RAM
50EA  23    00558       INC   HL          ;auf nächstes Byte
50EB  7A    00559       LD    A,D         ;Adreß-MSB
50EC  C608  00560       ADD   A,08        ;um 2 kB erhöhen
50EE  57    00561       LD    D,A         ;neues MSB
50EF  10F7  00562       DJNZ  chrlop2           ;bis 1 Zeichen eingelesen
50F1  08    00563       EX    AF,AF'            ;LSB Zeichenadresse
50F2  3C    00564       INC   A           ;nächster ASCII-Wert
50F3  20EB  00565       JR    NZ,chrlop1 ;bis 256 Zeichen eingel.
            00566
            00567 ;Bildschirm aus dem Puffer in 3 neue Video-Speicher kop.
            00568 ;(3 obere Viertel des einzelnen 6264)
50F5  1638  00569             LD    D,38h       ;Bildschirmadresse
50F7  0608  00570       LD    B,08h       ;BC <- Bildschirmlänge
50F9  EDB0  00571       LDIR              ;Bi1dschirmseite füllen
50FB  C1    00572       POP   BC          ;Bildschirmzähler
50FC  10D9  00573       DJNZ  vidloop           ;bis 4 Bildschirme fertig
            00574
            00575 ;Vorber. d. Hauptspeicherkopien, Aufruf d. Kopierroutinen
50FE  F1    00576       POP   AF          ;Systembyte 1
50FF  D3FA  00577       OUT   (0fah),A    ;mm-I/0 ausblenden
5101  F1    00578       POP   AF          ;Systembyte 0
5102  D3F9  00579       OUT   (0f9h),A    ;Common oben, Bank 1 unt.
5104  60    00580             LD    H,B         ;untere Speicherhälfte
5105  16F7  00581       LD    D,comtop+256/256  ;DE <- Pufferadr.
5107  ED73255100582           LD    (spbuf2),SP ;Stackpointer retten
510B  3100F7      00583       LD    SP,comtop+256    ;eig. Stack im ob. Common
510E  CD09F6      00584       CALL  copy+offset1     ;unteren Speicher kopier.
5111  212D40      00585       LD    HL,402dh    ;DOS-Ready als RET-Adr.
5114  E5    00586       PUSH  HL          ;auf den Initial-Stack
5115  E6FE  00587       AND   0feh        ;Common unten
```

```
5117  D3F9  00588       OUT   (0f9h),A    ;Systembyte 0 neu ausgeb.
5119  210080      00589       LD    HL,8000h    ;obere Speicherhälfte
511C  1630  00590       LD    D,30h       ;DE <- Pufferadresse
511E  310039      00591       LD    SP,combot+256    ;Stack im unteren Common
5121  CD0038      00592       CALL  copy-offset2     ;oberen Speicher kopieren
5124  310000      00593       LD    SP,0000h    ;Stackptr. restaurieren
5125        00594 spbuf2      EQU   $-2         ;(variabel)
5127  F1    00595       POP   AF          ;altes Systembyte 0
5128  D3F9  00596       OUT   (0f9h),A    ;restaurieren
512A  F1    00597       POP   AF          ;dto. Systembyte 1
512B  D3FA  00598       OUT   (0fah),A
512D  C9    00599       RET               ;zurück i. Betriebssystem
            00600
            00601 ;Bildschirm-Parameter restaurieren (Arbeitsadresse F600)
F600        00602 comtop      EQU   0f600h           ;im oberen  Common-Bereich
A4D2        00603   offset1   EQU   comtop-$    ;für Kopie  d. unt. Hälfte
512E  D3F9  00604 vidpar      OUT   (0f9h),A    ;Bank 0 selektieren
5130  CD5135      00605       CALL  3551h       ;Videoparameter setzen
5133  7B    00606       LD    A,E         ;Systembyte 0
5134  D3F9  00607       OUT   (0f9h),A    ;restaurieren
5136  C9    00608             RET
            00609
            00610 ;Speicher von Computer Nr. 0 auf Nachbarcomputer kopieren
            00611 ;(Arb.-adr. 3800 f. ob. Hälfte o. hinter vidpar f. unten)
3800        00612 combot      EQU   3800h       ;im unteren Common-Ber.
1937        00613 offset2     EQU   $-combot    ;für Kopie  der ob. Hälfte
            00614
            00615 ;32 Blocks zu je 1 kB in 3 Computer mit je 4 Banks kop.
5137  0620  00616 copy  LD    B,32        ;32 Blocks zu 1 kB kop.
5139  C5    00617 coplop1     PUSH  BC          ;Blockzähler retten
            00618
            00619 ;1 Block zu je 1 kB vor Kopie im Puffer Zwischenspeichern
513A  0604  00620       LD    B,4         ;4 Banks
513C  C5    00621 coplop2     PUSH  BC          ;Bankzähler retten
513D  E5    00622       PUSH  HL          ;dto. Zeiger
513E  D5    00623       PUSH  DE
513F  0604  00624       LD    B,04h       ;0400 = 1 kB
5141  EDB0  00625       LDIR              ;Block puffern
5143  E1    00626       POP   HL          ;Quelle und Ziel tauschen
5144  D1    00627       POP   DE
            00628
            00629 ;1 Block zu je 1 kB in 1 Bank von 3 Computern kopieren
5145  0603  00630       LD    B,3         ;Zähler 3 Nachbarcomputer
5147  C5    00631 coplop3     PUSH  BC          ;Computerzähler retten
5148  78    00632       LD    A,B         ;lfd. Computer-Nr.
5149  D3F3  00633       OUT   (0f3h),A    ;Computer selektieren
514B  0604  00634       LD    B,04h       ;1 kB
514D  E5    00635       PUSH  HL          ;Zeiger retten
514E  D5    00636       PUSH  DE
514F  EDB0  00637       LDIR              ;1 Block in Nachbarcomp.
5151  D1    00638       POP   DE          ;Zeiger restaurieren
5152  E1    00639       POP   HL
5153  C1    00640       POP   BC          ;dto. Computerzähler
5154  10F1  00641       DJNZ  coplop3          ;bis 1 Bl. in 4 C., 1 Bk.
            00642
            00643 ;dasselbe in der nächsthöheren Bank
5156  AF    00644       XOR   A           ;Computer Nr. 0 selekt.
5157  D3F3  00645       OUT   (0f3h),A
5159  DBF9  00646       IN    A,(0f9h)    ;Systembyte 0
515B  C640  00647       ADD   A,40h       ;1 Bank höher
515D  D3F9  00648       OUT   (0f9h),A    ;selektieren
515F  EB    00649       EX    DE,HL       ;Zeiger rücktauschen
5160  C1    00650       POP   BC          ;Bankzähler
5161  10D9  00651       DJNZ  coplop2          ;b. 1 Bl. in 3 C., 4 Bks.
            00652
            00653 ;dasselbe für den nächsten Block der Speicherhälfte
5163  24    00654       INC   H           ;auf nächsten Block
5164  24    00655       INC   H           ;(= 0400 addieren)
5165  24    00656       INC   H
5166  24    00657       INC   H
5167  C1    00658       POP   BC          ;Blockzähler
5168  10CF  00659       DJNZ  coplop1          ;bis 1 Memory-Hälfte kop.
516A  C9    00660       RET               ;zum Aufrufer
            00661
            00662 ;RST-00h-Rout., Segment in Bank 0 u. 1 (Arbeitsadr. 0000)
```

```
                ;von Bank 0 auf Bank 1 und umgekehrt schalten
516B  F3        rst00 DI            ;keine Störungen
516C  DBF9            IN    A,(0f9h          ;Systembyte 0
516E  EE01            XOR   01h         ;Bank 0 <-> 1
5170  D3F9            OUT   (0f9h),A
                ;in Bank 0 an der Adresse 0007 folgt hier JP (HL)

                ;Fortsetzung in Bank 1 ab 0007
                ;INT-Forts. sichern, prüfen, ob Sh.-CTRL-Ziffer gedrückt
5172  21BE45    bank1 LD    HL,45beh    ;Forts, d. INT-Routine
5175  3A8038          LD    A,(3880h)   ;Tastatur Steuerzeichen
5178  FE03            CP    03h         ;Shift-CTRL gedrückt?
517A  20EF            JR    NZ,rst00    ;falls nein
517C  3AC038          LD    A,(38c0h)   ;Tastatur Ziffernblock
517F  E60F            AND   0fh         ;0 - 3 gedrückt?
5181  28E8            JR    Z,rst00          ;falls nein

                ;Tastendruck erfolgt, Zähler für Computer-Nr. aufsetzen
5183  0600            LD    B,00h       ;Anfangsw. Computer-Nr.
5185  04        getcomp     INC   B          ;B hochzählen
5186  0F              RRCA              ;Tastenbit ins Cy
5187  30FC            JR    NC,getcomp ;falls noch nicht gefund.

                ;Systembytes, Stack und Adresse der Programmforts, retten
                ;(ab hier auch Bearbeitung mit Befehl SYS,BK,n)
5189  DBFA      switch      IN    A,(0fah)    ;Systembyte 1
518B  E6DF            AND   0dfh        ;Schreibschutz aufheben
518D  D3FA            OUT   (0fah),A    ;Systembyte neu ausgeben
518F  F620            OR    20h         ;Schreibschutz 0000-2FFF
5191  325D00          LD    (fabuf-rst00)    ;Systembyte 1 retten
5194  ED736C00        LD    (spbuf1-rst00),SP   ;SP retten
5198  3100F7          LD    SP,comtop+256    ;eigener Stack im Common
519B  E5              PUSH  HL          ;Fortsetzungsadresse
519C  DBF9            IN    A,(0f9h)    ;Systembyte 0
519E  325200          LD    (f9buf-rst00),A  ;retten

                ;Nachbarcomp. errechnen (RAM, Video, Zeichens., HRG-RAM)
51A1  3EEB            LD    A,0ebh           ; -15,  Anfw. Comp. & Video
51A3  C615      getall      ADD   A,15h      ;Cmp., Vid. u. Grph. ab 0
51A5  10FC            DJNZ  getall           ;bis neuer Comp, errechn.
51A7  324A00          LD    (newcomp-rst00),A   ;dort patchen

                ;alle Alternativregister des bisherigen Computers retten
51AA  08              EX    AF,AF'
51AB  D9              EXX
51AC  DDE5            PUSH  IX
51AE  FDE5            PUSH  IY
51B0  F5              PUSH  AF
51B1  E5              PUSH  HL
51B2  D5              PUSH  DE
51B3  C5              PUSH  BC

                ;neuen Computer selektieren, dessen Videoparameter setzen
51B4  3E00            LD    A,00h       ;Output für Port F3
51B5            newcomp     EQU   $-1        ;je nach Tastendruck
5186  E63F            AND   3fh         ;Steuerbits ausmaskieren
51B8  F680            OR      80h       ;Common nur comp.-intern
51BA  D3F3            OUT   (0f3h),A    ;Nachbarcomputer    selekt.
51BC  1E00            LD    E,00h       ;Systembyte 0 (kommt In
51BD            f9buf EQU   $-1         ;vidpar in den Akku)
51BE  21F037          LD    HL,37f0h    ;dort CRTC-Parameter
51C1  0610            LD    B,10h       ;16 Codes
51C3  AF              XOR   A           ;ab CRTC-Register 0
51C4  CD00F6          CALL  vidpar+offset1   ;Video-Attribute    restaur.

                ;Systembytes, Register usw. restaur. und Programm forts.
51C7  3E00            LD    A,00h       ;Systembyte 1
51C8            fabuf EQU   $-1         ;je nach dem
51C9  D3FA            OUT   (0fah),A    ;restaurieren
51CB  C1              POP   BC          ;alle Register restaur.
51CC  D1              POP   DE
51CD  E1              POP   HL
51CE  F1              POP   AF
51CF  FDE1            POP   IY
51D1  DDE1            POP   IX
```

```
51D3  D9    00738       EXX                 ;in den Alternativsatz
51D4  08    00739       EX    AF,AF'
51D5  E1    00740       POP   HL            ;Fortsetzungsadresse
51D6  310000      00741       LD    SP,0000h    ;SP restaurieren
51D7        00742 spbuf1      EQU   $-2         ;(variabel)
51D9  1890  00743       JR    rst00         ;umschalten und zurück
            00744 endrst00
            00745
4D00        00746 END   sys25               ;dort Einsprung

00000 Fehler
```

```
 10 REM: Testprogramm, beschränkt sich auf das Umschalten des Bildschirms.
 20 REM: Die vier Bildschirme wechseln einander ab. Zuvor muß das Initiali-
 30 REM: sierungsprogramm mit SYS,BK<,J> vom DOS aus aufgerufen worden sein.
 40 REM:
 50 A=INP(&HF3):REM          Zustand des neuen Systembytes 2
 60 A=AAND&HF3:REM           Video-Bits  ausmaskieren
 70 B=B+&H04:REM       Video-Bits  hochschalten
 80 B=BAND&H0C:REM           höhere Bits nicht belangen
 90 OUTSHF3,A+B:REM          auf nächsten Bildschirm wechseln
100 F0RI=0T0100:NEXT:REM     ein bißchen warten
110 G0TO60:REM               dasselbe Spielchen von vorne
```

Dasselbe in Assembler:

```
5200        00001 ORG   5200h
5200  DBF3  00002 Start IN    A,(0f3h)    ;Systembyte 2
5202  C604  00003 loop  ADD   A,04h       ;nächste Bildschirmseite
5204  E60F  00004       AND   0fh         ;nur bis Bit 3
5206  D3F3  00005       OUT   (0f3h),A    ;neu ausgeben
5208  08    00006             EX    AF, AF'          ;retten
5209  010080      00007       LD    BC,8000h    ;Zähler für Verzögerung
520C  0B    00008 delay DEC   BC          ;herunterzählen
520D  78    00009       LD    A,B         ;prüfen, ob Zähler
520E  B1    00010       OR    C           ;schon abgelaufen
520F  20FB  00011       JR    NZ,delay    ;falls noch nicht
5211  08    00012       EX    AF,AF'            ;Systembyte 2
5212  18EE  00013       JR    loop        ;und von vorne
5200        00014 END   Start             ;dort Einsprung

00000 Fehler
```

C: Einbau des Erweiterungsboards und Anschluß der damit erzeugten / dafür benötigten Signale

Da die meisten der für diese Schaltung benötigten Signale direkt vom Z80 geliefert werden, war es zweckmäßig, ein Board zu entwerfen, das anstelle des Z80 in dessen Sockel gesteckt wird und den Prozessor selbst aufnimmt. Damit sind dann nur noch zehn Signale über freie Verdrahtung mit dem CPU-Board zu verbinden. Diese Leitungen sind an einen zweireihigen Pfostenstecker geführt. Die Signale COML* und COMH* werden von den Pins 12 und 13 von U84, 74LS00 abgenommen und in beliebiger Anordnung an die Pins 3 und 4 des Steckers geführt. MWR* an Pin 5 des Steckers ist von Pin 8 von U52 (74LS32) zu beziehen, SYSRES* an Pin 6 des Steckers kommt von Pin 13 von U68, 74LS74.

Die Adressen A16 und A17 für die dynamischen RAMs (für den Arbeitsspeicher an den Pins 1 und 2 und für den HRG-Speicher an den Pins 9 und 10 des Steckers S2) können nicht direkt an die RAMs gelegt werden. Sie müssen gemultiplext an ihre jeweiligen Pins 1 geführt werden. Dafür ist zunächst die Computerplatine auszubauen und die Pins 1 aller 32 (Arbeitsspeicher) sowie 8 RAMs (Graphik) untereinander zu verbinden.

Dann werden auf U35 und U15 (beide 74S157) je ein 74S157-IC mit den Pins 1, 8, 15 und 16 huckepack aufgelötet. Dafür muß das untere IC aus der Fassung genommen werden, um es nicht versehentlich in derselben festzulöten. Selbstverständlich müssen beim Auf einander löten beide ICs die gleiche Orientierung haben (Kerbe auf der gleichen Seite). Die nicht verbundenen Pins des oberen ICs werden um ca. 45* hochgebogen, um keinen Kontakt mit dem unteren zu haben.

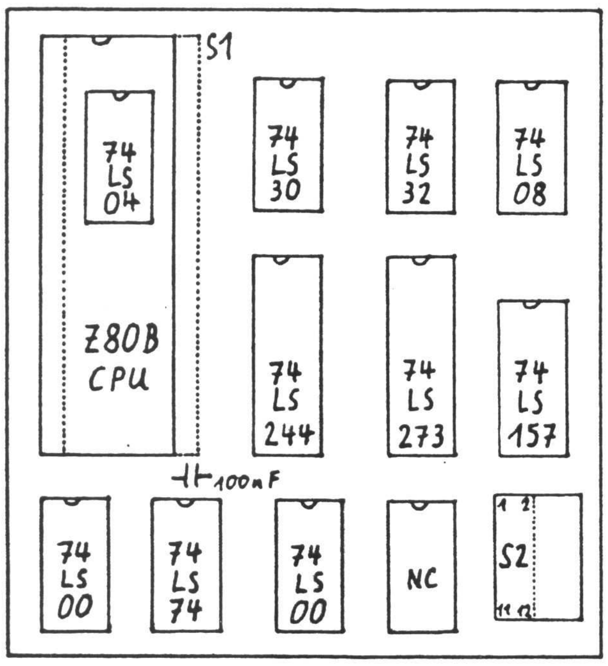
An die Pins 13 und 14 des auf U35 angebrachten 74S157 (hier genügt auch ein 74LS157) werden zwei isolierte Litzen gelötet, die an die Pins 1 und 2 des Steckers geführt werden. Zwei Litzen, die an die Pins 13 und 14 des auf U15 gesetzten ICs gelötet sind, werden an die Pins 9 und 10 des Steckers geführt. In beiden Fällen spielt die Anordnung der beiden Leitungen keine Rolle. Die Pins 12 der beiden Huckepack-ICs werden je über einen 33-Ω-Widerstand mit dem Pin 1 eines der benachbarten 41256-RAMs verbunden. Dafür muß das jeweilige RAM-IC aus der Fassung genommen und der Draht vorsichtig mit wenig Lötzinn an das obere dicke Ende des Pins 1 gelötet werden. Es ist ebenso möglich, bei herausgenommener Mutterplatine den Widerstand um deren Rand herum auf die Lötseite zu führen. Es kann zweckmäßig sein, den in die Leitung eingefügten 33-Ω-Widerstand mit einem Stück Schrumpf schlauch davor zu bewahren, Kontakt mit irgendwelchen IC-Pins zu bekommen.

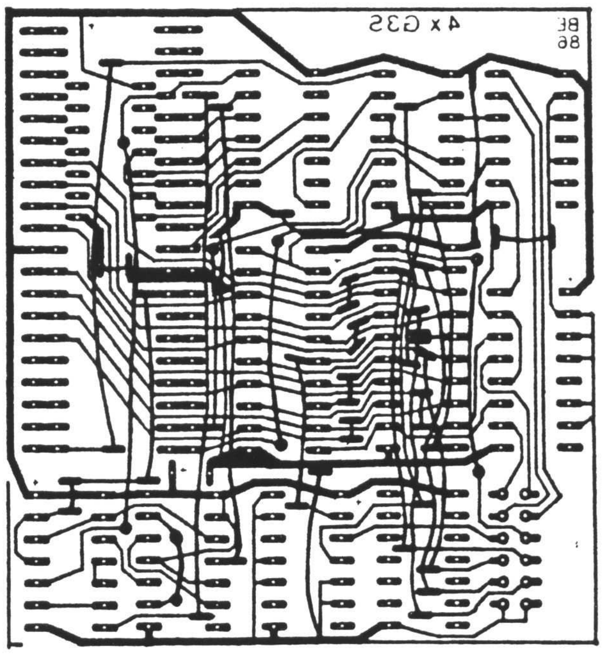
Die zur Vervierfachung von Video-RAM und Zeichengenerator (BIG4, BIG5 und BIG6) benötigten 6264LP15-ICs passen nicht mehr in die Fassungen der 6116LP15-RAMs. Hier hilft folgende Vorgehensweise: Die 6116-RAMs werden aus den Fassungen genommen. Die drei Kondensatoren an den Stirnseiten der Sockel werden ausgelötet und auf dem Rücken der 6264-RAMs mit kurzen Drahtstücken zwischen die Pins 14 und 28 gelötet. Die Pins 26 und 28 jedes 6264-Bausteins werden miteinander verbunden. Das IC, das in den Sockel von BIG4 gesteckt werden soll, muß an der Stirnseite (mit Kerbe) so weit beschliffen werden, daß es neben U87 mit den überstehenden Pins 1, 2, 27 und 28 in den Sockel paßt. Die beiden anderen 6264-ICs passen ohne zusätzliche Maßnahmen mit den links überstehenden vier Pins in die Sockel. Vor den Einstecken werden aber die Pins 23 aller drei RAMs um 45* abgebogen.

Die Pins 2 aller drei Speicher werden miteinander und mit dem Pin 8 des Steckers S2 des 4XG3s-Boards verbunden. Auch die hochgebogenen Pins 23 werden untereinander und mit dem Pin 7 des Steckers S2 verbunden. An die Pins 27 der RAMs werden folgende Signale gelegt (die vorher an die Pins 21 der 6116-Bausteine gelangt waren):

Pin 27 von RAM-IC	Signal	von	IC	Typ	Pin
BIG4	61WR *		U54	32	3
BIG5, BIG6	CHRWR*		U5	244	1

Sollten beim Einbau der Aus-1-mach-4-Erweiterung Fragen oder Probleme auftreten, kann sich der Bastler an Helmut Bernhardt wenden. Ungebohrte Leerplatinen bietet er zum Selbstkostenpreis an.





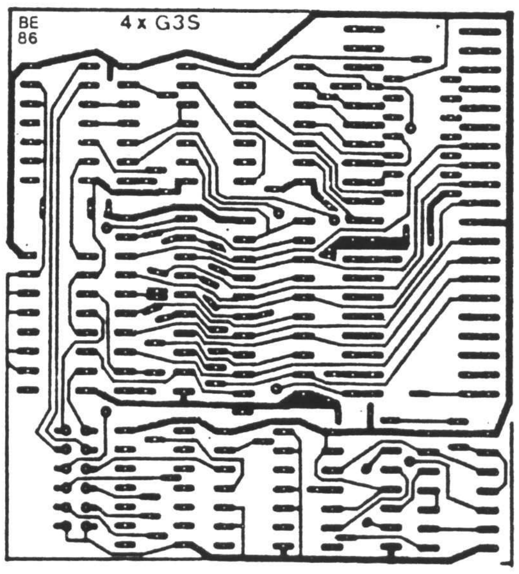
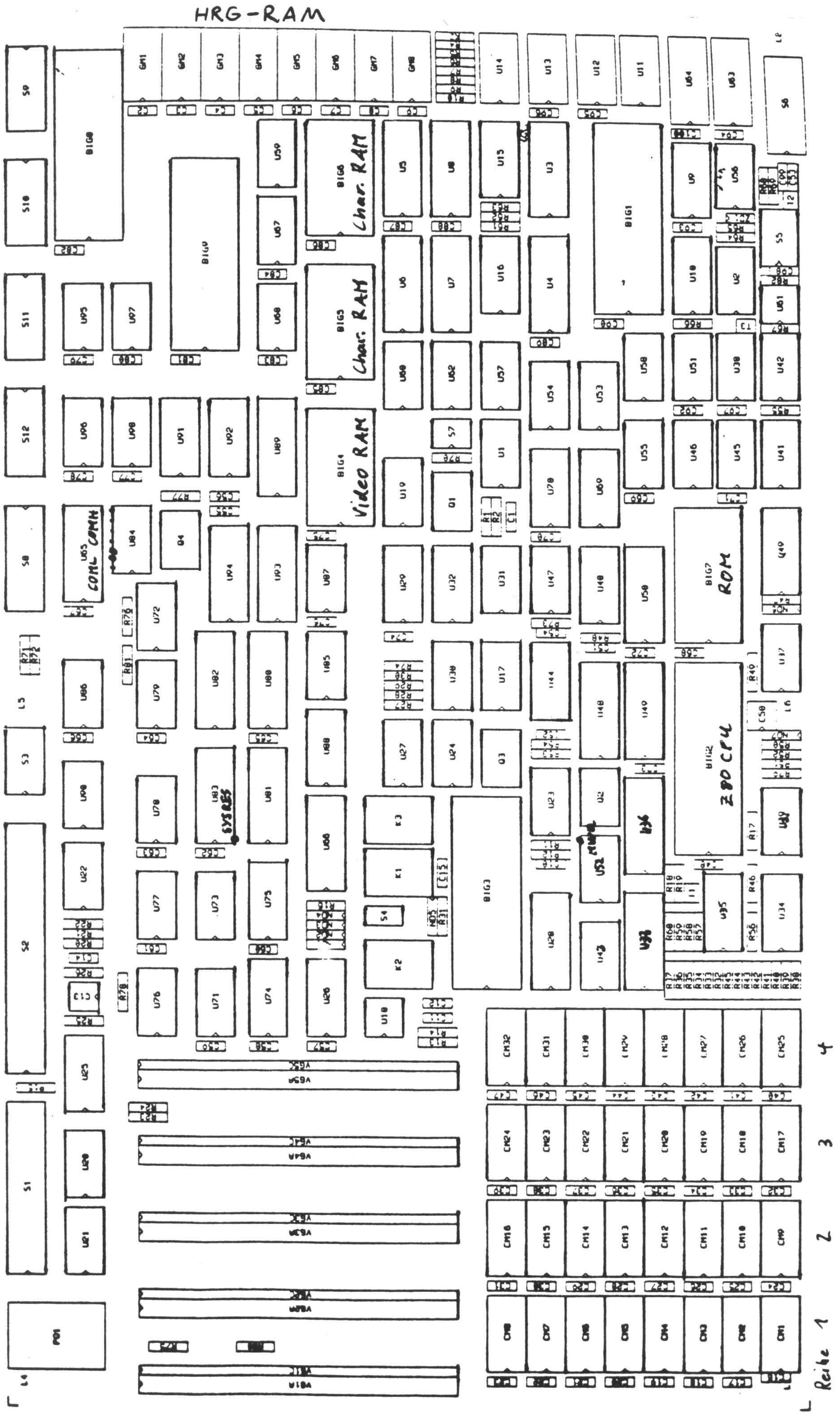


Abb. 8: (oben links) Bestückungsplan der Platine (leicht vergrößert)
S1: 2 einreihige Steckerleisten für den 280-Sockel auf der Mutterplatine, je 20 Pins
S2: 1 zweireihiger geknickter Pfostenstecker, 12 Pins

Abb. 9: (oben rechts) Platine des Bankers, Bestückungsseite mit Drahtbrücken (leicht vergrößert)

Abb. 10: (rechts) Platine des Bankers, Lötseite

Abb. 11: (unten) Die Mutterplatine des GIIIs



D: Eine nützliche Anwendung für das Bit Q6

In Abschnitt A dieses Artikels müssen wir leider zugeben, daß die vereinfachte und beschleunigte Kopie des Computers Nr. 0 auf seine drei Nachbarn mit Hilfe eines gesetzten Bit 6 im Port F3h leider nicht funktionierte. Auf dieses Feature kann auch verzichtet werden. Es würde die Kopierzeit von ohnehin nur dreieinhalb Sekunden um vielleicht eine halbe Sekunde verringern können. Da man nur einmal, höchstens zweimal am Tag diese Initialisierung vornimmt, ist die Wartezeit nicht so wichtig. Um Q6 aber nicht zu verschenken, haben wir es bis zum Aufspüren des Fehlers einer anderen interessanten Verwendung zugeführt:

Das Boot-EPROM des Genie III s ist ohne Treiber direkt mit der CPU verbunden. Daher ist es grundsätzlich möglich, Daten nicht nur aus dem EPROM zu lesen, sondern es auch damit zu beschreiben. Es gibt dazwischen keinen unidirektionalen Treiber, der diese Datenrichtung verbieten würde. Nun kann freilich in einem ROM nicht schreibenderweise etwas geändert werden, denn es ist eben ein Read-Only-Memory. Es ist jedoch möglich, ein statisches RAM parallel dazu zu stecken. Dazu muß lediglich das WR*-Signal der CPU auf den WE*-Pin des RAMs gelegt werden. Eine einfache Logik trifft die Auswahl zwischen dem EPROM und dem statischen RAM. Die Abb. 10 zeigt die Beschaltung der nunmehr zwei Speicherbausteine anstelle des bisherigen EPROMs, nämlich dieses selbst und ein statisches RAM des Typs 6264.

Ein möglichst kurzes Flachbandkabel sitzt mit einem 28-poligen Ouetschstecker im EPROM-Sockel. An seinem anderen Ende ist eine Lochrasterplatine entweder direkt aufgelötet oder mit einer Pfostensteckerverbindung befestigt. Fast alle Signale aus dem Sockel werden auf die korrespondierenden Pins der beiden Bausteine einfach durchgeschleift. Nur das Signal ROMEN* (ROM-enable) muß auf die in der Abbildung beschriebene Weise mit Bit 6 des Ports F3h verknüpft werden, um eine Auswahl zwischen dem EPROM und dem neuen statischen RAM treffen zu können. Außerdem wird MWR* von S2 des Bankers, Pin 5 dem Pin 27 (WE*) des RAMs zugeführt.

Q6 wird an Pin 5 des 74LS274 auf der Lötseite der Banker-Platine abgegriffen und auf einen freien Pin von S2 geführt, so daß sich die Verbindung mit der RAM/EPROM-Bank vereinfacht. Es ist nicht erforderlich, die weiterführende Leiterbahn zum 74LS00 zu durchtrennen und gegen GND zu legen, denn das Kopierfeature funktioniert vorerst sowieso nicht. Es kann also nichts passieren, und nach der Korrektur des Fehlers ist weniger Restaurationsarbeit erforderlich (falls man überhaupt noch restaurieren will, nachdem man die Vorzüge dieser ROM/RAM-Bank erkannt hat).

Der Platz zwischen dem EPROM-Sockel und der Zusatzplatine für den Banker ist so eng, daß zwar das EPROM noch eben dazwischen paßte, der neue Ouetschstecker jedoch nicht mehr ohne weiteres. Um die Zusatzplatine dennoch fest und kontaktsicher in den Sockel der CPU stecken zu können, empfiehlt es sich, das Deckelteil des Steckers nach dem Anquetschen des Kabels wieder zu entfernen und die hochstehenden Spitzen der Kontakt- klemmen abzuschleifen. Diese werden anschließend mit

Isolierband beklebt, damit sie nicht die Lötseite der Zusatzplatine berühren können.

Wer den Banker nicht, wohl aber das parallele RAM neben dem EPROM einbauen möchte, kann sich diese Mühe sparen. Dann steht allerdings selbstredend nicht das Bit 6 des Ports F3h zur Verfügung. Ein Umschalter, der anstelle von Q6 wahlweise GND oder über einen Widerstand von 10 kΩ +5V an die zugehörigen Eingänge des NAND-Gatters legt, erfüllt denselben Zweck.

Um beim Einschalten oder nach einem Reset auf jeden Fall das Boot-EPROM zu erreichen, bekommt es den Zustand 0 des Bits 6 zugeordnet. Wenn Q6 aber 1 lautet, ist das RAM selektiert.

Der zugängliche Adreßraum dieser neu geschaffenen EPROM/RAM-Bank ist 12 kB groß (0000-2FFFh). Das Original-EPROM des Genie III s (2732) belegt davon nur 4 kB (0000-0FFFh) . Es kann einfach durch einen 2764 ersetzt werden, in dessen doppelt großem Speicherplatz interessante Features untergebracht werden können. Der 28-polige Sockel führt auch an den vier bisher nicht benutzten zusätzlichen Pins die dafür erforderlichen Signale. Das statische RAM 6264 belegt von vornherein den doppelten Adreßraum von 8 kB (0000-1FFFh).

Bei 1 MB Hauptspeicher grenzt die vergleichsweise bescheidene Zugabe von 8 kB an Memory-Raffgier, aber bei dem Konzept von vier Computern existiert ohne relativ aufwendiges Banking kein gemeinsamer Speicher, der zur Übergabe von Daten geeignet wäre. Dies ist in der EPROM/RAM-Bank der Fall. Wird nämlich Bit 2 des Ports FAh low gesetzt, dann ist ursprünglich auf jeden Fall im Adreßraum 0-12 kB das EPROM ansprechbar. Mit dieser Modifikation ist es nun, je nach Q6, wahlweise das EPROM oder das RAM. So können die vier Computer dort bei Selektion des RAMs auf bequeme Weise miteinander kommunizieren.

Eine andere Anwendung ist für Tüftler interessant, die das Boot-EPROM gerne nach ihrem eigenen Geschmack ändern möchten. Wenn in seinem Adreßraum auch ein RAM liegt, kann eine Versuchsversion des neuen EPROM-Inhalts dort probegefahren werden, ohne gleich ein EPROM dafür zu opfern. Hier empfiehlt es sich, das Flachkabel nach außen zu verlegen, um beim Einstecken eines neuen EPROMs nicht jedesmal den Computer zerlegen zu müssen. Nun ist es zwar nicht möglich, einfach durch Aufruf mit dem Programmnamen den EPROM-Code ins parallele RAM zu laden und dort zu starten. Das anschließend gelistete Programm besorgt das aber.

Die Datei muß zuvor zwischengepuffert werden, um sie anschließend nach dem Umschalten auf die EPROM/RAM-Bank und der Selektion des RAMs dort ablegen zu können. Die Programm-Laderoutine des G-DOS (Einsprung an 4419h) verfrachtet aber Programme immer gleich an ihre Bestimmungsadresse, was in diesem Falle mit der Fehlermeldung 'Ladeversuch auf ROM-Speicherplatz' quittiert würde. Deshalb muß diese Routine (in Grossers DOS-Buch LOADP genannt) zuvor durch einen Patch geändert werden:

An der Stelle 4C4Dh in SYS0 wird das MSB der Ladeadresse eines Records gelesen. In dem hier vorgestellten Programm wird dort eine Umleitung (Label devia) gezapft, die feststellt, wie dieses MSB lautet. Wenn die Adresse unterhalb 3000h liegt, muß es sich um den Code für das neue EPROM handeln, den wir zunächst puffern wollen. Deshalb wird dann in Z. 41 das Bit 7 des MSB gesetzt, so daß z. B. aus der Adresse 0500h die Adresse 8500h wird. So steht nach dem Laden das Programm ab 8000h im Speicher.

Im ersten Segment der Laderoutine muß nun noch der Patch in SYS0 rückgängig gemacht werden. Schließlich sorgt noch eine Sicherung dafür, daß ein fehlerhaft geladenes Programm nicht die Testbedingungen verfälscht: Bei einem auf getretenen Fehler ist die NZ-Bedingung erfüllt. In diesem Fall erfolgt einfach unverrichteter Dinge eine Rückkehr ins DOS (Z. 18).

Wenn aber alles geklappt hat, wird zunächst in der EPROM/RAM-Bank das RAM selektiert, indem das Bit 6 des neuen Systembytes 2 (Port F3h) high gesetzt wird. Danach wird Uber das Systembyte 1 die Bank zugeschaltet. Jetzt ist von 0000-1FFFh das statische RAM zugänglich. Die Zeiger für die Quelle (Puffer ab 8000h), das Ziel (RAM ab 0000h) und der Zähler für die Programmlänge (2000h) werden aufgesetzt und der Code an seinen Bestimmungsort geladen.

Die DOS-Laderoutine legt die Einsprungsadresse eines Programms in 4403h ab. Von dort wird sie nach HL geladen, so daß mit JP (HL) das Programm angesprungen werden kann. Für ein Boot-EPROM wird diese Adresse immer 0000h sein, aber auf diese Weise kann auch ein beliebiges anderes Programm in der neuen Bank abgearbeitet werden.

Beim Einsprung in das Boot-Programm wird geprüft, ob die F1-Taste gedrückt ist, um anstelle eines Boot im Monitor zu bleiben. Das Laden des Programms dauert je nach Größe lange genug, um nach dem Aufruf der Laderoutine rechtzeitig diese Taste zu erreichen. So können sowohl die Bootfähigkeit des neuen EPROM-Inhalts als auch der vielleicht modifizierte Monitor getestet werden.

Der Loader benutzt hier für den neuen EPROM-Inhalt den Filenamen BOOTEPRH/CMD. Das ist natürlich willkürlich. Der Name kann beliebig dem eigenen Geschmack angepaßt werden. Es muß auch nicht unbedingt ein Boot-Programm sein. Diese Routine lädt, was sie erwischt, und springt sie dann an.

Falls Schwierigkeiten beim Einbau oder Betrieb des Statt-ROM-RAMs auftreten (auch solo ohne den Banker), können beide Autoren weiterhelfen.

Abb. 12: Schaltung für die EPROM/RAM-Bank

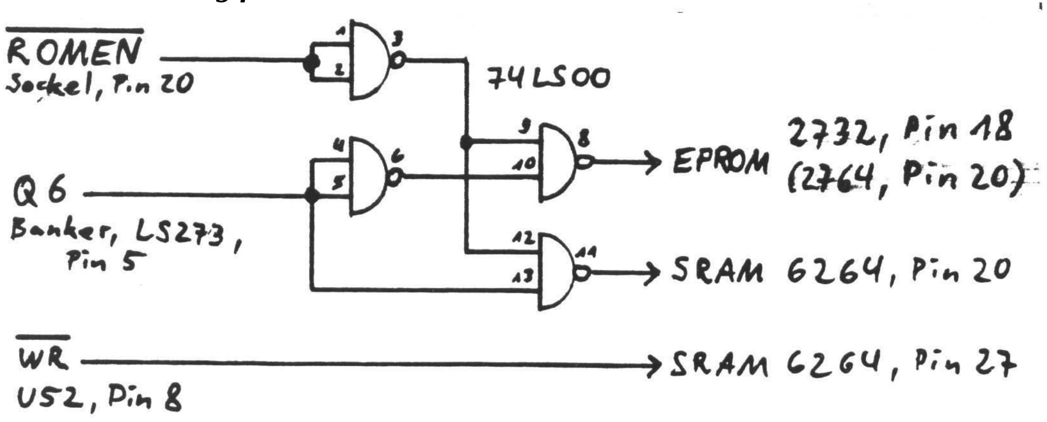
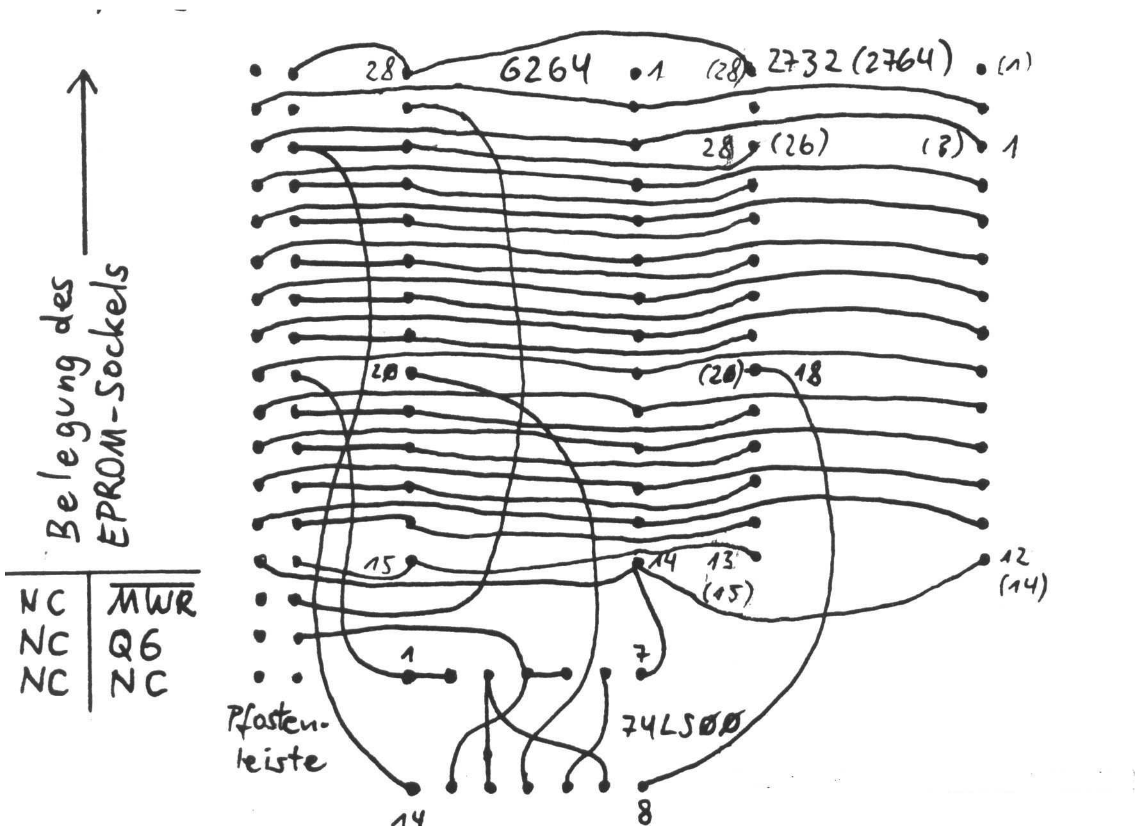


Abb. 13: Verdrahtungsplan (Lötseite)




```
                ;***********************************************************
                ; Routine zum Laden von Programmen, die im statischen RAM
                ; neben dem Boot-EPROM (0000-1FFF) laufen sollen
                ;
                ; (C) 1986 by Arnulf Sopp
                ;***********************************************************

3000            ORG   3000h

                ;Umleitung im UP LOADP (Programm laden) in SYS0 setzen,
                ;Programm BOOTEPRH/CMD nach 8000 verschoben laden
3000  212C30    Start LD    HL,devia    ;Umleitung zur Modifikat.
3003  224E4C          LD    (4c4eh),HL  ;in LOADP patchen
3006  213530          LD    HL,doscall  ;Zeiger auf DOS-Befehl
3009  CD1944          CALL  4419h       ;diesen ausführen
300C  21654C          LD    HL,4c65h    ;alten Code in LOADP
300F  224E4C          LD    (4c4eh),HL  ;restaurieren
3012  C0              RET   NZ          ;Ende, falls Fehler

                ;EPROM/RAM-Bank und dort wiederum das RAM selektieren
3013  F3              DI                ;es wird brenzlig
3014  DBF3            IN    A,(0f3h)    ;Systembyte 2
3016  F640            OR    40h         ;RAM statt EPROM
3018  D3F3            OUT   (0f3h),A    ;selektieren
301A  3EC0            LD    A,0c0h      ;auf RAM/EPROM-Bank
301C  D3FA            OUT   (0fah),A    ;Bank freigeben

                ;Prg. von 8000 nach 0000 im Boot-RAM laden und anspringen
301E  210080          LD    HL,8000h    ;Puffer des Boot-Progr.
3021  55              LD    D,L         ;DE <- 0000
3022  5D              LD    E,L         ;(= Zieladresse)
3023  010020          LD    BC,2000h    ;Programmlänge
3026  EDB0            LDIR              ;Programm ins Boot-RAM
3028  2A0344          LD    HL,(4403h)  ;Puffer der Startadresse
302B  E9              JP    (HL)        ;Programm anspringen

                ;Patch in LOADP: Code bis 3000 soll nach 8000 ff. kommen
302C  CD654C    devia CALL  4c65h       ;MSB Ladeadresse holen
302F  FE30            CP    30h         ;zum Boot-Progr. gehörig?
3031  D0              RET   NC          ;norm. weiter, falls nein
3032  F680            OR    80h         ;ab 8000 laden
3034  C9              RET               ;normal weiter

                ;DOS-Befehl zum Laden des neuen Boot-Programms
3035  4C        doscall DM  'LOAD,BOOTEPRH/CMD',0dh ;DOS-Bef., ENTER

3000            END   start ;dort Einsprung

00000 Fehler

devia 302C      doscall     3035  start 3000
4D00            ORG   4d00h
4D00  FEFB      sys25 CP    0fbh        ;Lib-Befehl F# ?
4D02  2822            JR    Z,fkey      ;falls  ja
```

```
4D04  FEDB  00026       CP    0dbh       ;Lib-Befehls SYS,BK ?
4D06  CA2A50     00027       JP    Z,sidekck  ;falls ja
```

1986 Club80-Info Nr. 17, S.54

Nachtrag zu

Dein GIIIs, die 4 unbekannten Wesen

Helmut Bernhardt, Arnulf Sopp

Wir mußten in diesem Artikel ratlos zugeben, daß die Vervierfachung des Grafik-Speichers nicht geklappt hatte! Sie klappt jetzt. Schuld war ein unbemerkter Lötzinntropfen, der den Pin 1 der RAMs (gemultiplexte Adressen A16 und A17) per Kurzschluß auch noch mit Refreshes beglückte.

Die Schaltung ist also vollkommen in Ordnung und kann nachgebaut werden. Man sollte aber etwas vorsichtiger löten als wir. Es lohnt sich. Vielleicht sind 8 Grafikseiten für alberne Bilder wirklich überflüssig. Aber da es sich um gewöhnlichen RAM-Speicher handelt, ist der Zugewinn von 256KB durchaus attraktiv. Welcher "Kompatible" kann schon mit 1 1/4 MB RAM, gleichzeitig mit 4 bzw. 8 Bildschirmen und 4 Zeichensätzen aufwarten"

Dein Genie 3s, die 5 unbekannten Wesen

Nun soll auch ein Z180-Prozessor Chef im System sein können

Helmut Bernhardt, Arnulf Sopp

Vorbemerkung: Im Folgenden ist viel von "wir" die Rede. Dies aus Gründen der Sprachökonomie. Jeder von uns machte aber in Wirklichkeit das, was er am besten konnte: Helmut die Hardware, Arnulf die Software. Das Pflichtenheft für beides entstand unter Mitwirkung etlicher Gebinde Hopfenbrause.

Die Hardware

Vor längerer Zeit hatten wir unter ähnlichem Titel mal eine Erweiterung für den Genie IIIs beschrieben (1), mit der sämtliche Speicher (User-RAM, Video-RAM, HRG-RAM und Zeichengenerator-RAM) auf die vierfache Größe erweitert werden konnten. Außerdem fiel dabei noch ein SRAM ab, das parallel zum Boot-EPROM arbeitete. Dafür haben wir in den Sockel der Z80-CPU ein Adapter-Board gesteckt, das den Z80 aufnahm und einen schreib- und lesbaren Port bereitstellte, mit dessen Bits die Umschaltung dar Speicherbanks gesteuert wurde.

Über dieses primitive Banking kann z.B. ein Genie 2s-Anwender, der auf eine HD64180-CPU umgerüstet hat (2), nur milde lächeln, wenn er seinen Computer mit 9,2 MHz (gegenüher 7,2 MHz des G3s) fährt. Es galt, dieses Manko beim G3s auszubügeln. Dafür haben wir ein neues Adapter-Board gebaut, auf dem nun kein Z80 (auch kein HD64180), sondern ein Z180 steckt. Der Z180 hat (wie auch der neuere HD64180-PLCC) 20 Adreßleitungen und kann damit über seine MMU und die beiden DMA-Känle 1 Megabyte RAM direkt adressieren.

Um diese CPU so, wie sie gedacht ist, direkt zu nutzen, hätte das riesige Betriebssystem G-DOS 2.3 (oder Calva-DOS 2.4, die auf 1 Megabyte angepaßte Version) vollkommen auf HD64180-MMU-Banking umgeschrieben werden müssen. In über 63 kB SYS- und anderen systemeigenen Files hätten alle OUT-Befehle an die System-Steuerports F9h und FAh gesucht und auf entsprechende Z180-MMU-Befehle geändert werden müssen. Außer viel Arbeit hätte das auch die Kompatibilität mit anderen Computern dieser und ähnlicher Bauart gekostet. Da entschieden wir uns dafür, dem Calva-DOS sein Banking über die Systemteuerports zunächst mal zu erhalten. Die Ports F9h, FAh und FBh (früher F3h) auf dem Adapter-Board haben ihre volle Funktionsfähigkeit behalten.

Nun ist aber der Einsatz eines HD64180 bzw. Z180 witzlos, wenn dessen MMU und DMAC nicht benutzt werden können. Wir haben beide Möglichkeiten eingerichtet: Nach dem Einschalt-Reset sind alle Systemport-Bits auf 0 gesetzt und ein voll G3s-kompatibler Computer incl. Erweiterung auf vierfache Speichergröße liegt vor. Durch Setzen des früher anders belegten Bits D6 in Port FBh kann aber umgeschaltet werden in einen lupenreinen Z180-Computer: alle Banking- und Common-Definitions-Bits der Systemports haben dann keine Funktion mehr. Das RAM wird ausschließlich durch die Adreßleitungen der CPU adressiert.

Nur das Einblenden von memory mapped I/O-Baugruppen in die dafür vorgesehenen physikalischen Adreßbereiche erfolgt, wie im G3s-Modus, durch die entsprechenden Systemport-Bits. Gleichwohl sind deren logische Adressen variabel. Der G3s dekodiert bei ihrer Selektion nur die Adreßbits 0-1. Wie der Z180 derzeit sein physikalisches Megabyte in bis zu drei logische Adreßbereiche unterteilt, indem er mit A12-A19 jongliert, da redet ihm der G3s nicht drein.

Um die Ausmaße eines Adapter-Boards mit diesen Umschaltmöglichkeiten (s. Foto) in Grenzen zu halten, mußten wir zwei PALs einsetzen. PAL 1 erzeugt lediglich die Port-Freigabesignale /OUTFB und /INFB (und noch einige weitere Freigabesignale - s. PAL-Listing), mit denen das Latch 74LS273 und der Lesetreiber 74LS244 angesteuert werden, die den zusätzlichen Port BFh realisieren.

Die eigentliche Steuerung der RAM-Adressierung leistet PAL 2. Durch die Ausgänge /CAS0-/CAS3 wird eine der vier Reihen von 41256er RAM-Chips selektiert. A16 und A17 wählen darin einen 64 kB-Block an. Für die Adressierung im Z180-Modus gelangen die Adressen A16-A19 an das PAL. Im G3s-Modus erfolgt das Banking durch die Bits D6 und D7 von Port F9h sowie D0 und D1 von Port FBh. D7 von Port FBh entscheidet, ob bei logischer Adressierung im eingestellten Common-Bereich (0000h-3FFFh oder E000h-FFFFh) nur innerhalb eines 256 kB-Bereichs oder innerhalb des gesamten 1 MB-Speichers der unterste 64 kB-Block adressiert werden soll.

Der Zugriff auf den eingestellten Common-Bereich wird durch das high aktive Signal COMLH (U84, 74LS00 Pin 11 auf dem CPU-Board) angezeigt. Dieses Signal wird ebenfalls an das PAL geführt (wie auch D1 von Port FBh. das die Auswirkung von COMLH steuert), Durch D6 von Port FBh erfolgt schließlich die Steuerung ob der G3s-Modus oder der Z180-Modus für die Adressierung des RAMs zuständig ist. Das PAL-Assemblerlisting veranschaulicht alle diese Steuerfunktionen.

Damit die Signale/CAS0 - /CAS3 anstelle, der bisherigen Signale /CAS0 - /CAS3 an die RAMs gelangen, ist U43, 74LS125 zu entfernen. Die Signale /CAS0 - /CAS3 des Adapter-Boards werden den RAMs über die Pins 3, 6, 11 und 8 der Fassung von U43 zugeführt. Die an diesen Pins liegenden Pull up Widerstände R57-R60 sollten ausgelötet werden. Die Versorgung mit +5V und GND kann von den Pins 14 und 7 derselben Fassung erfolgen. Und schließlich liefert ihr Pin 9 auch noch das Signal /CAS für den Adapter.

Alle Signale, die dem Adapter-Board vom CPU-Board zugeführt werden müssen, werden über den Stecker CN2 geführt; ebenso die Signale die das Adapter-Board dem CPU-Board liefert. Nur die ursrünglichen Z80-Pin-Signale laufen über den Sockel des Z80.

Der Einsatz eines Z180 bringt aber auch neue Probleme mit sich. Beim Z80 wird der Sysemtakt extern erzeugt und der CPU über Pin 6 eingespeist. Beim HD64180 und beim Z180 wird der Systemtakt, von der CPU geliefert. Dieser Takt wird intern aus einem von außen zugeführten, doppelt so hohen Takt hergeleitet.

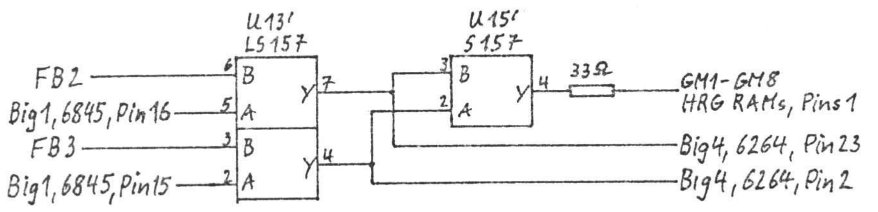
Der G3s hat die Möglichkeit (leider auch die Notwendigkeit), zwischen einer hohen Frequenz und den normalen 1,77 MHz des TRS-80 umzuschalten. Diese Umschaltung muß bereits mit dem doppelten Muttertakt erfolgen. Es muß also zwischen 18.332 und 3,55 MHz umgeschaltet werden. Dadurch kann die CPU nicht mehr mit einem normalen Quarz, an den Pins XT und EXT beschaltet werden. Es müssen externe Oszillatorschaltungen benutzt werden. Der jeweils selektierte Takt wird über einen Treiber (74LS125) an den Pin EXT der CPU geschaltet. Die Auswahl des jeweiligen Taktes wird durch die Signale LO (U41, 74LS74, Pin 9) und HI (U41, 74LS74 Pin 8) auf dem CPU-Board gesteuert.

Die bisherige Taktquelle im System wird durch Entfernen von U44, 74LS161 abgeschaltet. Damit der nun von der CPU gelieferte Systemtakt auch auf dem CPU-Board verfügbar wird, müssen die Pins 5 und 6 von U39, 74LS125 aus der Fassung gebogen werden und die Pins 5 und 6 der Fassung miteinander verbunden werden. Die Widerstände R51, R52 und R79 sollten entfernt werden.

Die für das ursprüngliche 256 kB-Banking des G3s verantwortlichen Bits 6 und 7 des Ports F9h (hier als Signale F96 und F97 bezeichnet) sind von den Pins 14 und 13 von U88, 73LS139 zu beziehen.

Die vom Port FBh auf dem Adapter-Board zur Vervierfachung von Video- RAM, Zeichensatz-RAM und HRG-RAM gelieferten Bits 2-5 (FB2-FB5) werden in dieser Reihenfolge an folgende IC-Pins der Hauptplatine geführt:

U13', 73LS157, Pin 6
U13', 74LS157, Pin 3
BIG5 und BIG6, 6264, Pins 23 (diese Pins aus der Fassung biegen)
BIG5 und BIG6, 6264, Pins 2 (diese Pins sind bereits außerhalb der Fasung)



U13' ist ein mit den Pins 1, 8, 15 und 16 huckepack auf U13, 74LS157 gelötetes IC 74LS157. Die weitere Beschaltung dieses ICs gibt folgende Abbildung wieder:

U15' ist ebenfalls ein auf U15, 74LS157 mit den Pins 1, 8, 15 und 16 huckepack gelötetes weiteres IC 74S157.

Die Pins 15 und 16 von BIG1, 6845 sind aus der Fassung zu biegen und mit U13' zu verdrahten.

Schließlich sind noch die Adressen A16 und A17 von CN2 den RAMs gemultiplext zuzuführen. Das Multiplexen bsorgt ein auf U34, 74S157 wiederum mit den Pins 1, 8, 15 und 16 huckepack gelöteter 74S157 (U34'), der an den Pins 13 und 14 mit A16 und A17 zu versorgen ist. Der Multiplex-Ausgang Pin 12 wird über 33 Ω mit den Pins 1 aller RAMs (32x 41256) verbunden.

Zur Vereinfachung des bis hierhin Gesagten sei das alles noch einmal in einer Tabelle wiedergegeben:

Verbindungen von CN2 zum CPU-Board

Stecker CN2 Pin	Signal	CPU-Board IC	Typ	Pin	
1	+5V	U43	LS125	14 *	# huckepack gelötete
2	GND	U43	LS125	7 *	ICs 74LS(S)157
3	/HI	U41	LS74	8	(s. Text und Abil-
4	/LO	U41	LS74	9	dungen
5	FB3	U13'	LS157	3 #	
6	FB2	U13'	LS157	6 #	
7	FB5	BIG5/BIG6	6264	7	
8	FB4	BIG5/BIG6	6264	23	(aus der Fassung biegen)
9	F96	U88	LS139	14	
10	F97	U88	LS139	13	* U43 entfernen und
11	/CASO	U43	LS125	3 *	Sockel auslöten,
12	COMLH	U88	LS139	15	Signale von CN2 an
13	/CAS2	U43	LS125	11 *	die Lötpunkte der
14	/CAS3	U43	LS125	8 *	entsprechenden Pins
15	/CAS	U43	LS125	9 '	des Sockels von U43
16	/CAS1	U43	LS125	6 '	anlöten
17	A16	U33'	S157	13 #	
18	A17	U34'	S157	13 #	

Außerdem sind auf dem CPU-Board noch folgende (z. T. beim Einbau des 4xG3s-Z80-Adapters schon durchgeführten) Änderungen nötig:

Die aus den Fassungen gbogenen Pins 1 von U16, U15 und U16' (alle, 74S157) sind mit Pin 5 von U57, 74S04 zu verbinden. Die Pins 27 von BIG5 und BIG6 (6264), die beide außerhalb der Fassung hängen, werden mit Pin 1 von U54, 74LS32 verbunden. Pin 27 von BIG4, 6264 wird mit Pin 3 von U54, 74LS32 verlötet.

Abb. 2:Pinout des Z180 und seines Sockels

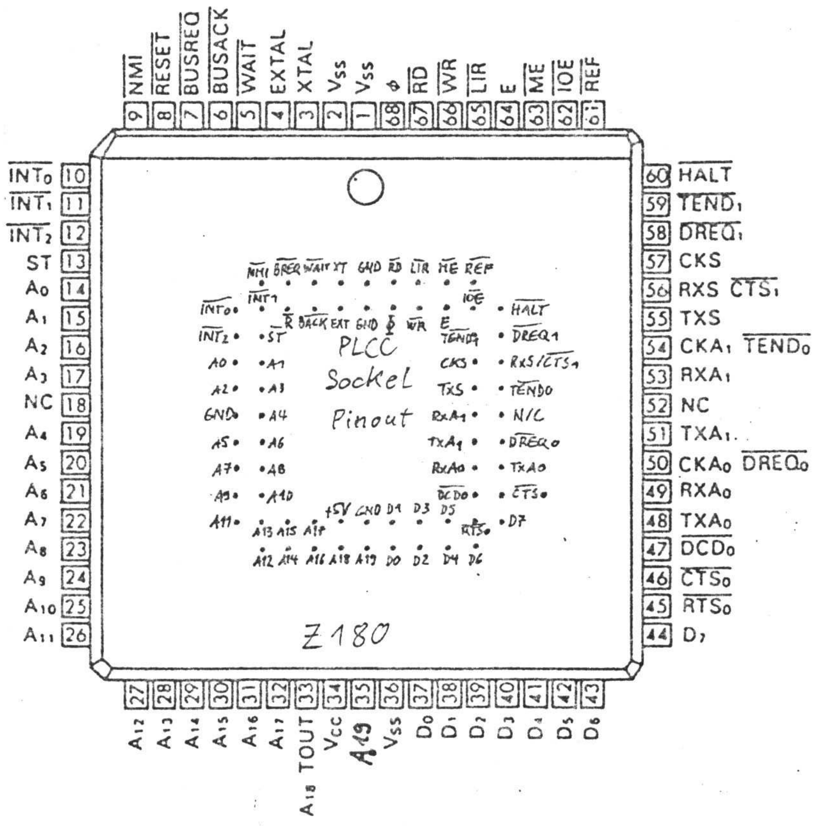
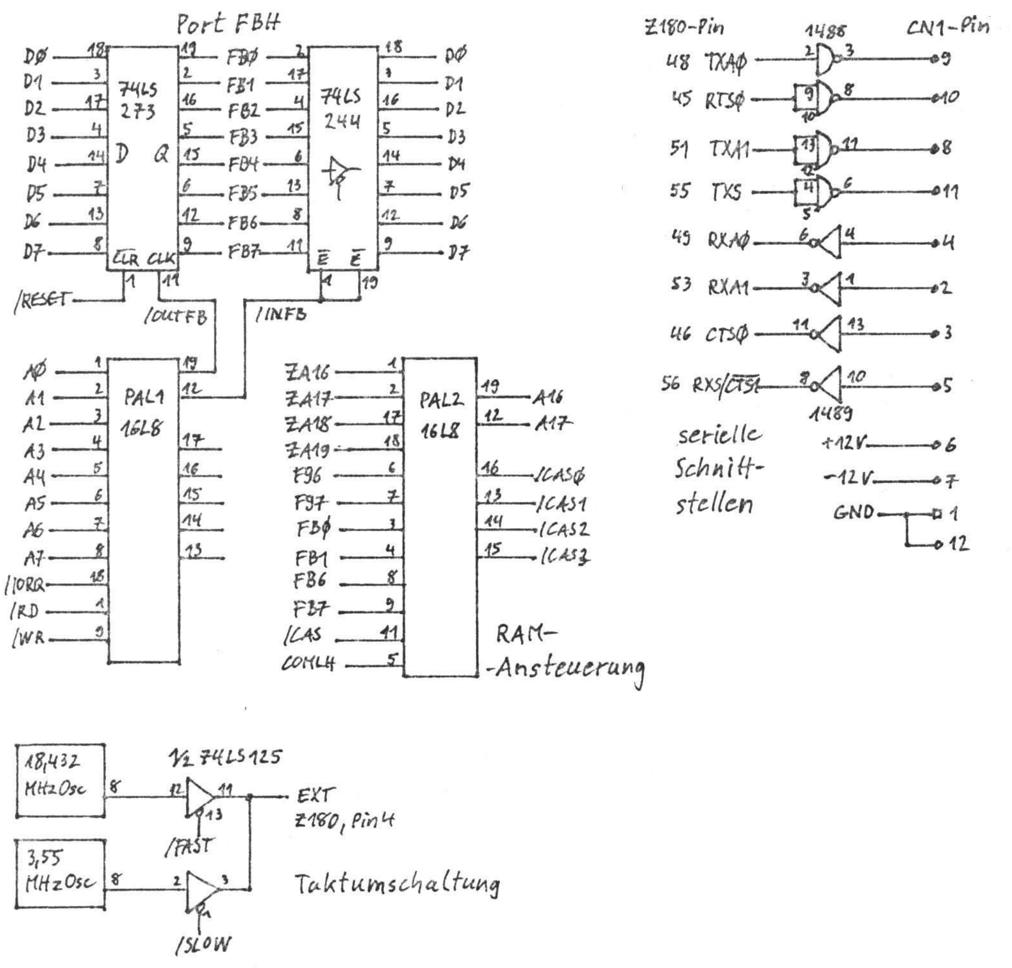


Abb. 3: Auszüge aus der Schaltung

Z180-Pin	Signal	Z80-Adapter-Pin
14	A0	30
15	A1	31
16	A2	32
17	A3	33
18	A4	34
19	A5	35
20	A6	36
21	A7	37
22	A8	38
23	A9	39
24	A10	40
29	A14	4
30	A15	5
40	D3	8
42	D5	9
43	D6	10
10	/INT0	16
9	/NMI	17
66	/WR	21
62	/IORQ	20
7	/BUSRQ	25



6	/BUSAK	23
8	/RESET	26
65	/M1	27
5	/WAIT	24
61	/RFSH	28
68	CLK	6

von Hand zu verdrahten zwischen Z180-CPU und Z80-Adapter-Stecker

CN2-Belegung

Signal	Pin		Signal
+5V	1	2	GND
/FAST	3	4	/SLOW
FB3	5	6	FB2
FB5	7	8	FB4
F96	9	10	F97
/CAS0	11	12	COMLH
/CAS2	13	14	/CAS3
/CAS	15	16	/CAS1
A16	17	18	A17

Die Listings der beiden PALs

```
PAL16L8
10.03.90  H. Bernhardt, PAL1
Portdecodierung im GIIIs mit Z180 für zusätzl. Systemports

A0 A1 A2 A3 A4 A5 A6 A7 WR GND
RD OUTFB OUTFC OUTF3 IODC INF3 INFC IORQ INFB VCC

/INFB  = /IORQ * /RD * A7 * A6 *A5 * A4 *  A3 * /A2 +  A1 *  A0
/OUTFB = /IORQ * /WR * A7 * A6 *A5 * A4 *  A3 * /A2 +  A1 *  A0
/INFC  = /IORQ * /RD * A7 * A6 *A5 * A4 *  A3 *  A2 + /A1 * /A0
/OUTFC = /IORQ * /WR * A7 * A6 *A5 * A4 *  A3 *  A2 + /A1 * /A0
/INF3  = /IORQ * /RD * A7 * A6 *A5 * A4 * /A3 * /A2 +  A1 *  A0
/OUTF3 = /IORQ * /WR * A7 * A6 *A5 * A4 * /A3 * /A2 +  A1 *  A0
/IODC  = /IORQ *     * A7 * A6 *A5 * A4 *  A3 * /A2
```

```
PAL16L8
10.03.90  H. Bernhardt, PAL2
RAM-Steuerung im GIIIs mit Z180

ZA16 ZA17 FB0 FB1 COMLH F96 F97 FB6 FB7 GND
CAS A17 CAS1 CAS2 CAS3 CAS0 ZA18 ZA19 A16 VCC

/ZA16  = /FB6 * /F96
       + /FB6 * COMLH
       +  FB& * /ZA16
/ZA17  = /FB6 * /F97
       + /FB6 * COMLH
       +  FB6 * /ZA17
/CAS0  = /FB6 * /CAS  * /FB1  * /FB0
       + /FB6 * /CAS  * COMLH * /FB7
       +  FB6 * /CAS  * /ZA19 * /ZA18
/CAS1  = /FB6 * /CAS  * /FB1  *  FB0  * /COMLH
       + /FB6 * /CAS  * /FB1  *  FB0  * FB7
       +  FB6 * /CAS  * /ZA19 *  ZA18
/CAS2  = /FB6 * /CAS  *  FB1  * /FB0  * /COMLH
       + /FB6 * /CAS  *  FB1  * /FB0  * FB7
       +  FB6 * /CAS  *  ZA19 * /ZA18
/CAS3  = /FB6 * /CAS  *  FB1  *  FB0  * /COMLH
       + /FB6 * /CAS  *  FB1  *  FB0  * FB7
       +  FB& * /CAS  *  ZA19 *  ZA18
```

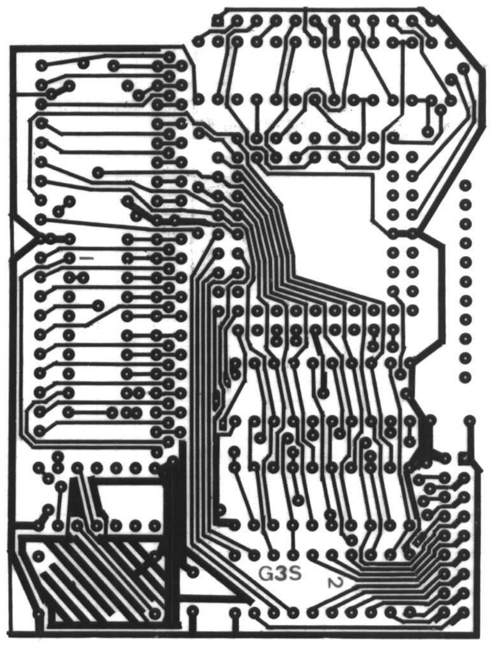


Abb. 4: Layout (lötseitig gesehen)

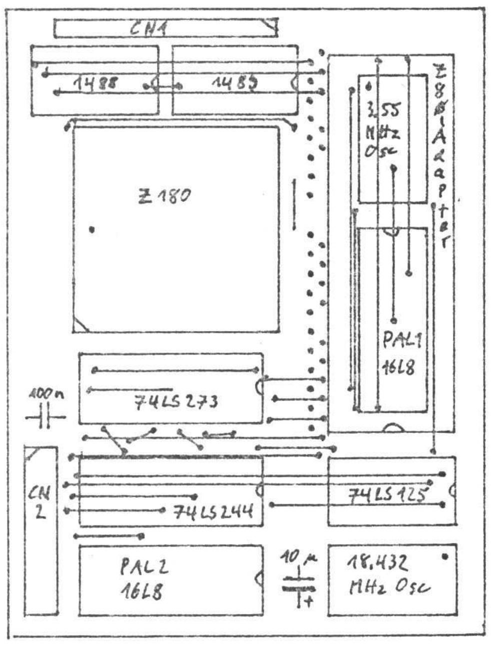


Abb. 5: Bestückungsplan

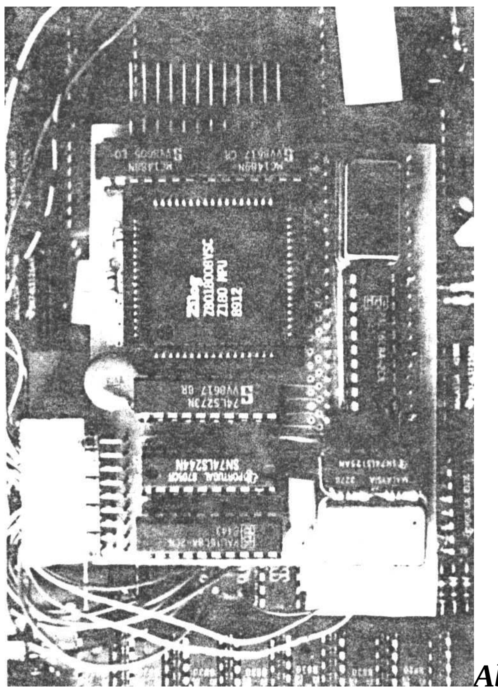


Abb. 6: aufgebautes, angeschlossenes Board

Die Software

Der Z180 verleitet mit seiner interessanten MMU (Memory Management Unit) und seinem DMAC (Direct Memary Access Controller) geradezu zu Spielereien. Der Leser sei versichert daß wir ihnen auch ausgiebig frönten. So ist es z. B. kein Problem, den Bildschirm oder irgendeinen anderen physikalichen Speicherabschnitt logisch gleich dreimal an verschiedenen Adressen im 64 kB-Fenster der CPU vorliegen zu haben. Hier soll aber ganz biererernst nur vorgestellt werden, was an Änderungen im DOS und Unterstützungen für das DOS prgrammiert wurde, um die Vorteile der neuen CPU nutzen zu können.

Zum Verständnis dessen, was sich nun eigentlich zum Besseren ändert, möchten wir außer den im Hardware-Teil schon genannten Veröffentlichungen noch auf ein Sonderheft des Club-80 (3) und einen Info-Artikel (4) hinweisen, die zum Thema HD64180 erschienen sind. Diese CPU ist softwaremäßig identisch mit dem Z180. Eine erneute Beschreibung der im wesentlichen gleichen Eigenschaften erübrigt sich an dieser Stelle.

Es sind vier Programme, die für den Z180 geändert bzw. neu erstellt wurden:

Patches in zwei verschiedenen Teilen von SYS0/SYS konfigurieren den Z180, um seine Arbeitsgeschwindigkeit und seine MMU auf optimale Werte einzustellen und um seinen DMAC dem System verfügbar zu machen.

SYS26/SYS ist zuständig, um die SYS-Files des DOS in Bank 0 des Speichers zu puffern. Es wird nun nur noch angesprungen, wenn nicht der Rechtspeil gedrückt wurde (was bedeuten würde, daß das DOS nicht gepuffert werden soll). In SYS26 steht nun der DMA-Treiber aus dem neuen SYS0 bereits zur Verfügung, was alle Blockladevorgänge beschleunigt. Die Pufferung der DOS-Moduln benötigt jetzt nicht mehr das aufwendige Banking, das bisher erforderlich war, denn der DMAC kann von überall her den vollen Adreßbereich des Z180 überstreichen.

Eine Änderung in der SYS-Laderoutine, für den DOS-Kern ebenfalls in SYS26 untergebracht, lädt nun die SYS-Programme bei einem DOS-Request mit RST 28h per DMA entsprechend schneller und weniger speicheraufwendig. Daß im Zuge dessen viel Platz in SYS26 frei wurde, der edleren Zwecken zur Verfügung gestellt werden kann, sei nebenbei erwähnt. Zudem ist der besondere Sektorpuffer 3A00h-3AFFh für die SYS-Moduln jetzt nicht mehr notwendig - freier Speicher für andere Anwendungen.

Dann entstand ein Anwenderfile (bei uns TRAP/CMD geheißen), das hauptsächlich Gerald Schröders Illegal-Trap-Programm in Bank 0 und Bank 1 an die logische Adresse 0000h patcht. Hier springt der Z180 nämlich hin, wenn er einen Befehl antrifft, den er nicht kennt (für nähere Erleuterungen möchten wir auf Geralds Artikel im Info Nr. 17 verweisen). Außerdem wird hier die Delay-Routine an 0060h verlangsamt, denn der Z180 arbeitet zu schnell, um mit dem alten Verzögerungsprogramm noch die korrekten Wartezeiten zu erzielen.

Und schließlich zimmerten wir ein Utility-Programm namens SYSTEST/CMD für den DOS-Tüftler (Listing 5). Wenn ein SYS-File, umgeschrieben wurde und nach Murphy's Gesetz noch Fehler enthalten muß, kann es zunächst in der Bank 0 gepuffert und (notfalls bei offenem Floppy-Türchen) von dort aus getestet werden. Damit erübrigt sich gleichzeitig die Notwendikeit, vor dem Test erneut zu booten, um die Datei in die Bank 0 zu kriegen, von wo sie beim gebankten DOS aufgerufen wird.

Alle Änderungen in SYS0 und SYS26 sowie die neuen Files TRAP/CMD und SYSTEST/CMD sind in den Listings wiedergegeben. Weitere Teile, die sich in den beiden SYS-Programmen nicht geändert haben, sind mit LIST OFF ausgespart worden, um den Rahmen dieses Sonderhefts nicht zu sprengen. Dasselbe gilt für TRAP/CMD, soweit es Geralds bereits bekanntes Trap-Programm betrifft. Damit hat der Leser natürlich die Schwierigkeit, daß unklar bleibt, wo die Patches in den Files liegen sollen, und was sich in der Folge sonst noch ändert. Sorry - da können wir nur anbieten, eine Systemdiskette bei Arnulf anzufordern. Die Listings sollen lediglich den Umgang mit den neuen Möglichkeiten des Z180 demonstrieren und die sich ergebenden Voreile besingen.

Es werden auch kleinere Teile der Listings unverständlich bleiben, weil sie sich auf nicht gelistete Programmabschnitte beziehen. Es handelt sich nämlich nicht um besondere Schöpfungen für dieses Sonderinfo, sondern um die Sources für das geänderte DOS. Wenn etwa Labels auftauchen, die sonst nirgends wiederzufinden sind, einfach darüber hinweglesen!

Die genannten Programme sind auch noch längst nicht der Weisheit letzter Schluß. So ist es beispielsweise wünschenswert, einige Konfigurationen des Z180 (WAIT-Zyklen, Refreshes, MMU, vielleicht noch mehr) bereits im Boot-EPROM erledigen zu lassen. So weit sind wir jedoch selbst noch nicht. Gegf. wird man in einem kommenden Clubinfo Weiteres dazu vorfinden.

Soweit Befehle enthalten sind, die zwar der Z180, nicht aber der Z80 kennt, sind sie mit ihren Hex-Entsprechungen als DB-(DEFB-)Statememts eingefügt. ZEUS kennt sie nämlich bedauerlicherweise auch nicht. In einem Kommentar erscheint dann jeweils der Befehl in Z180-Assembler.

Die Programme sind essentiell für das Betriebssystem. Ohne Betriebssystem kein Computer. Daher wird unbedingt empfohlen, alles zunächst mit einer Kopie der Systemdiskette auszuprobieren (für SYS29 kann aber auch SYSTEST gefahren werden!). Erst, wenn sich die Kopie als funktionstüchtig erwiesen hat, hat die alte Systemdisk ausgedient!

Nun zu den Programmen im einzelnen:

SYS0/SYS ist die Seele des G3s. Es ist in seinem ersten Teil eine Fortsetzung von DOS/SYS. Hier ist die erste Modifikation für den Z180 untergebracht: Die Anzahl der WAIT-Zyklen für Speicher- und Portzugriffe wird eingestellt, und zwar auf den schnellstmöglichen Wert. Sollte er sich im Einzelfall als Überforderung des Systems herausstellen, kann der Wert 00h für den Akku mit DDE auf einen anderen Betrag gepatcht werden (XOR A hätte den Akku ebenfalls gelöscht, jedoch ohne die Möglichkeit einer einfachen Änderung). Danach wird die MMU des Z180 auf den Normalzustand des G3s eingerichtet. So kann, etwa für einen DMA (s. u.), die Z180 Speicherkonfiguration in aller Regel geschaltet werden, ohne daß sich etwas am Banking-Zustand ändert.

Die zusätzlichen Befehle (s. Listing 1) werden einfach in die (zuvor per Disassembly gewonnene) Source eingefügt, das Programm neu assembliert. Es besteht aus zwei Sektoren, die in die ersten beiden Sektoren von SYS0/SYS gepatcht werden (das kann z. B. SUPERZAP).

Das Listing 2 zeigt ebenfalls einen Patch in SYS0. Anfangs wird getestet, ob der Rechtspfeil während des Bootens gedrückt wurde. Ist das nicht der Fall, möchte der User das DOS in Bank 0 des Computers puffern, um es später ohne die Laufwerke verfügbar zu haben. Dieser Test war bisher in SYS26 untergebracht. Es erschien uns überflüssig, SYS26 anzuspringen, wenn es bei gedrücktem Rechtspfeil ohnehin gleich wieder verlassen wird.

Dann enthält das Listing einen Treiber für den DMAC des Z180. Er erwartet beim Einsprung in HL die Anfangsadresse des zu transferierenden Blocks, in DE die Zieladresse, in BC den Bytezähler. Bis hierher entspricht der Treiber dem Z80-Befehl LDIR. Neu ist der Inhalt des Akkus, der in seinen beiden Nibbles die Adressen A16-19 des Quell- und des Zielbereichs enthält. So kann ein DMA auf das volle Megabyte des Z180-Adressraums zugreifen.

Die internen DMAC-Register SAROB (Kanal 0, Quelle, Bits 16-19) und DAROB (Kanal 0, Ziel, Bits 16-19) werden nur in ihren unteren 4 Bits beachtet. So war es zur Laufzeit- und Platzersparnis möglich, auf das Löschen der oberen 4 Bits zu verzichten: Der Befehl RRD (Zeile 94) rotiert die hier relevanten Bits 3-7 ins untere Nibble von DAROB, der folgende Befehl LD (scrphys),A packt einfach den kompletten Akku mit seinem unwichig gewordenen oberen Nibble nach SAROB.

In Zeile, 102 wird die G3s-MMU deaktiviert, indem in Port FBh das Bit 6 gesetzt wird (s. Hardware-Teil). Nur so kann der DMA-Treiber über das ganze Megabte verfügen. Ansonsten könnte nur das 64 kB-Fenster, das gerade vom G3s bearbeitet wurde, vom DMAC überstrichen werden. Also ist beim Einsprung in die DMA-Routine unbedingt darauf zu achten, daß die MMU-Konfiguration des Z180 mit der des G3s zumindest in wichtigen Speicherabschnitten übereinstimmt: Der Treiber muß sich nach dem Umschalten selber wiederfinden können, der Stack muß ebenfalls an derselben logischen Adresse vorliegen. Erst nach dem Rückschaltbefehl (Zeile 111) ist Entwarnung gegeben.

Bei der Rückkehr aus dem Programm enthalten alle Register ihren alten Wert, der gerettet wurde. Im Gegensatz dazu gibt der Befehl LDIR die Quell- und Zielregister um den Betrag des Bytezählers erhöht zurück. BC selbst ist dann 0000h. Dieser Service hätte mit in den Treiber eingebaut werden können, der verfügbare Platz reichte jedoch nicht. Zudem haben beide Möglichkeiten ihre Vor- und Nachteile.

An dieser Portion von SYS0 ist nur der filerelative Sektor 10h interessant. Das ist der diskreltive Sektor 15h. Die Source gem. Listing 2 wird assembliert. Von der so entstandenen CMD-Datei muß nun (mit SUPERZAP oder wie auch immer) der erste Sektor nach SYS0 kopiert werden, und zwar in den eben bezeichneten Sektor. Wichtig ist, daß dieser Sektor des CMD-Programms genau am relativen Byte FFh aufhört, also ganz unten rechts. Danach, im nächsten Sektor, folgt der Rest, der mit einem neuen Record-Header beginnt. Bis dorthin können NOPs bzw. DW 0000h- oder DB 00h-Statements eingefügt werden. Um das Puzzle zu erleichtern, folgt hier ein Sektor-Dump der den relativen Sektor 10h von SYS0 wiedergibt:

```
file:  SYS0/SYS     drv: 0    frs: 0010h     drs: O015h
00:  018C 0032 3E40 D3F9 FBC1 D1E1 C9E5 D5C5    2 >@
10:  3E01 F3D3 F9C3 0AF0 3A07 43F6 3032 8232  >       : C 02 2
20:  E60F FE04 2o11 3AF8 42CB 5F28 0A3A 4038        : B  ( :@8
30:  E640 3EFC CC02 4421 7F32 1180 4401 0B00   @>   D! 2  D
40:  EDB0 1180 4421 0042 CD24 4428 0BFE 1820      D! B $D(
50:  2B3E 0132 0743 180E EBCD 284C 201E 1162  +> 2 C   (L   b
60:  32D5 1198 31E9 2100 3011 AD4E 0100 023E  2   1 ! 0  N   >
70:  10CD BD35 E1F1 32F9 42C3 6744 F5CD C901     5  2 8 gD
80:  3E46 EF4F 564c 782F 5359 533A 300D 0170  >F OVLx/SYS:0  p
90:  BB35 22F0 35ED 53F3 35ED 43F6 35E5 21F5   5" 5 s 5 C 5 #
A0:  3577 ED67 32F2 3521 F735 C501 2708 F5D8  5w g2 5! 5
B0:  FBF5 F640 F3D3 FBED 9B01 3102 ED9B F1D3     @      1
C0:  FBF1 C1E1 C940 0200 0000 0000 0000 0000      @
D0:  0000 0000 0000 0000 0000 0000 0000 0000
E0:  0000 0000 0000 0000 0000 0000 0000 0000
F0:  0000 0000 0000 0000 0000 0000 0000 0000
```

Es ist ausreichend, das im Hex- und im ASCII-Teil *gelb hinterlegte* Zählbyte des Record-Headers (rel. Sektorbyte 8Fh) genau so zu zappen um den Rest auszunullen. Dann dürfte nichts mehr schiefgehen. Voraussetzung ist, daß das Zählbyte auch an dieser Stelle steht. Ansonsten muß es eben auf andere Weise angepaßt werden, damit das letze Datenbyte des Sektors an der sektorrelativen Stelle FFh zu liegen kommt.

Listing 3 behandelt ein paar Änderungen in SYS26. Dort wird, wie schon erwähnt, das Puffern des DOS im Speicher veranlaßt. Gleichzeitig wird der DOS-Kern gepatcht: Soll ein SYS-File mit RST 28h geladen werden, wird nicht mehr die Floppy angeworfen. Statt dessen wird das Programm aus der Bank 0 geholt.

Dafür dient ein besonderes Minimal-DIR: Pro SYS-Modul enthält es zwei Bytes. Das erste ist das MSB der Pufferadresse in Bank 0. Da das LSB immer 00h ist, da also jedes SYS-File am Anfang eines 256Byte-Blocks beginnt, genügt das MSB. Das zweite Byte gibt die Länge des Files in Sektoren an. Gepuffert werden nur die Programme SYS1-19. GDOS/SYS, IHHALT/SYS und SYS0/SYS haben hier keine Bedeutung und bleiben deshalb auf der Floppy. Ihr Platz im Mini-DIR ist mit Nonsene (GDOS) bzw. mit Nullen (INHALT und SYS0) gefüllt.

Eine spezielle Laderoutine für SYS-Programme liest dieses DIR und transferiert dann die Sektoren ursprünglich nach 3A00h, einen besonderen Sektorpuffer. Von dort wird das File sektorweise in den Overlay-Speicher des DOS geladen und schließlich angesprungen.

Hier die Änderungen die die Vorteile des Z180 ausnutzen: Der DMAC benötigt kein Banking, um Daten zu transportieren. Die Unterprogramme, die das DOS in den Speicher laden und es später beim Betrieb von dort wieder abholen, konnten deshalb wesentlich verkürzt (und damit beschleunigt) werden.

Das UP putsys puffert die SYS-Files. Dazu wird zunächst der Teil von INHALT/SYS ab 5200h in den Speicher geladen, der SYS-Programme enthält. Der Platzbedarf von 0800h Bytes zerstört dabei nach Reset den Inhalt des Anwenderspeichers. Daher wird (ab Label boot) zunächst dieser Bereich in die Bank 1 gerettet um später restauriert zu werden. Dann wird das besondere SYS-DIR im Block 3900h eingerichtet. Die ersten Bytes des Blocks müssen ausgenullt werden, damit bei einem DOS-Request nicht versucht wird, INHALT oder SYS0 aus der Bank 0 zu holen; die Nullen sind Signalbytes für das Verbot. Anschließend wird ab 4100h in Bank 0 das restliche DOS abgelegt.

Mit der Restauration des Anwenderspeichers und dem Transfer des SYS-DIR an seine endgültige Adresse 4000h in Bank 0 endet dieser Teil von SYS26.

Der Patch im DOS-Kern, der SYS-Dateien aus der Bank 0 holt, ist anschließend gelistet. Der verfügbare Speicherplatz im residenten Teil von SYS0 (ab 4BE1h) reicht dazu nicht aus. Ein notwendiges Unterprogramm macht deshalb an der Stelle 3738h weiter (was auch bisher schon so war). Dieses UP kann nun per DMA auf aufwendiges Banking verzichten, das zuvor nötig war. Immerhin 13 Bytes wurden frei die Arbeitsgeschwindigkeit stieg enorm.

Das nächste Programm, das mit Listing 4 hier vorgestellt werden soll, ist in dieser Form ein reines Anwender-File. Wegen seiner Wichtigkeit erscheint es sinnvoll, es später einer SYS-Datei einzuverleiben, vielleicht gleich SYS26. Es nimmt ein paar Patches im BASIC-Interpreter vor, die dem Z180 gerecht werden.

Seine Arbeitsgeschwindigkeit ist deutlich höher als die des Z80, weil viele Befehle weniger Taktzyklen beanspruchen. Dadurch beschleunigt sich die Delay-Routine an 0060h bedenklich. Wartezeiten, die dort z.B. für den Floppy-Controller abgetrödelt werden, sind nun zu kurz. Eine geringfügige Änderung schafft Abhilfe, so daß die Verzögerungen fast genau mit der alten Zeiten identisch werden:

In der alten Version wird mit einem JR NZ-Befehl in die Schleife zurückverzeigt, wenn der Zähler BC noch nicht abgelaufen ist. Die neue Variante durchläuft zuerst den Befehl RET Z, und zwar jedesmal. Hat der Zähler noch etwas Stoff, geht es mit bedinungslosem JR zurück in die Schleife. Der zusätzliche Return-Befehl, der bei jedem Schleifendurchlauf gelesen werden muß, kostet die erwünschten Taktzyklen zur Verlangsamung.

Dies alles trifft freilich nur bei der langsameren Takt-Fequenz von 1,77 MHz zu, die über Port FAh, Bit 6 gewählt werden kann. Alle kritischen Verzögerungsschleifen des DOS werden jedoch, soweit uns bekannt, erst nach dieser Taktumschaltung durchlaufen. Wo nicht, ist die verkürzte Laufzeit durch den High-Speed-Takt des Z180-Boards immerhin noch erträglich zu schnell.

Direkt an die Delay-Schleife schließt sich der NMI-Service an. Der TRS-80 benutzt ihn als Reset-Ersatz. Beim G3s ist der NMI unbenutzt (hard- warermäßig übrigens auch nicht zugänlich - die freien Tasten links oben und unten in der Haupttastatur können mit dem NMI-Pin der CPU verdrahtet werden). Unsere Serice-Routine läßt einfach nur die Interrupts wieder zu. Damit können z. B. die Dreitastenbefehle aktiviert werden, wenn man etwa jederzeit mit dem Debugger den Speicher und die Reister im Auge haben möchte.

Hauptbestandteil dieses Programms ist jedoch Gerald Schröders Error-Trap-Routine. Dorthin verzweigt der Z180, wenn ein ihm unbekannter Opcode angetroffen wird. Auch hier möchten wir wieder auf schon erschienene Erklärungen verweisen, um Wiederholungen zu vermeiden (Info Nr. 17).

Beim Einsprung am Label **start** wird der Beginn der Trap-Routine, nämlich ein Sprung nach 1650h, nach 0000h ff. gezapt. Vor dem eigentlichen JP findet sich ein NOP. Hier kann der User bei einem Trap-Interrupt auch etwas anderes als diesen Sprung veranlassen. Mit F7h (RST 30h) an dieser Stelle wird z. B. in den Debuger gesprungen. Besonders in der Experimentierphase, wenn der Z180 noch neu ist, wird man das gelegentlich wollen.

Bei 1650h sind die BASIC-Befehle im Klartext abgelegt. Das Trap-Programm überschreibt sie. Wer gerne in BASIC arbeitet, kann so natürlich nicht verfahren. Eine andere Adresse muß her. In einem kommenden Info wird die Lösung des Problems zu lesen sein.

In Bank 1 soll ebenfalls Geralds Trap-Programm verfügbar sein. Dort ist ab 0000h Platz dafür, ein JP erübrigt sich. Diese neue Ladeadresse erfordert die Relokation aller festen Adressen im Trap-Pogramm. Das beginnt mit Zeile 188 im Listing. Der Vorgang ist etwas verzwickt. Wir möchten aber darauf verzichten, ihn über die Kommentare im Listing hinaus zu erleutern. Dies kann kein Lehrbuch über Z80-Assembler sein). Alle kritischen Adressen wurden mit einem Label versehen, addr1-addr15. addr1 ist im Listing sichtbar, die übrigen nicht. Diese Labels finden sich in der Tabelle addrtab wieder.

Nur ein Detail soll beleuchtet werden. Weil der Leser u. U. nicht gleich RAMDISK (s. u.) über JP 4405h abfahren möchte: Beim Sprung nach 4405h wird über den Requestcode 63h nach SYS1 verzweigt. Mit, diesem code im Akku (ACHTUNG: mit etlichen anderen nicht!) wird als erstes der Stack neu eingerichtet, weil dabei eine Rückkehr zum Aufrufer nicht stattfinden soll. Für den CALL dma und die PUSHes und POPs im DMA-Treiber genügt ein vorübergehender Stack an beliebiger Stelle. Er wird hier (zeile 199,) kurz und schnell unter die Trap-Routine gelegt.

Bei einem anderen Aussprung aus TRAP/CMD, ganz besonders mit RET, muß zuvor der Original-Stack unbedingt gerettet und nachher restauriert werden. Das geht mit dem Strickmuster nach dem auch die Zeilen 192 und 189/190 (in dieser Reihenfolge!) funkionieren. Das hätten wir unter Dreingabe von ein paar Bytes und Mikrosekunden natürlich auch hier so machen können, aber dieser schlitzohrigere Weg war halt gar zu verlockend.

Nachdem nun die Trap-Routine auch nach Bank 1 tranferiert ist, wird zuletzt die RAM-Disk des G3s aktiviert. Dieser Programmbestandteil wäre verzichbar. Da wir aber immer mit der RAM-Disk arbeiten, kann das getrost gleich mit erledigt weren. Wenn nein: Unbedingt die Aussprungs-Bedingungen beachten (Stack, s. o.). Übrigens ist der ursprüngliche Name des Proramms MEMDISK/CMD. Hier handelt es sich um eine geänderte Version.

SYSTEST transferiert ein CMD-Programm, das als SYS-File des DOS dienen soll, in die Bank 0, von wo die SYS-Dateien mit RST 28h aufgerufen werden. Prinzipiell geht das mit jeder Datei, in der Regel wird es sich jedoch um neue Versionen eines echten DOS-Moduls handeln.

Es beginnt gem. Listing 5 mit der Prüfung, ob das DOS überhaupt schon in Bank 0 gepuffert ist. Das ist die Voraussetzung, weil sonst die neue Datei nicht mit RST 28h erreichbar wäre. Sollte das noch nicht geschehen sein, wird SYS26 (s. d.) mit FCh im Akku zuerst aufgerufen, um das DOS zu banken. Zum Test dient ein Lückenfüller-NOP, das bei der geänderten SYS-Laderoutine an der Stelle 4C0Bh stehen muß.

Das eigentliche Arbeitsprogrogramm klärt nun zunächst, was für ein CMD-File in we1ches SYS-Programm kopiert werden soll. Diese Angaben können jedoch schon beim Aufruf gemacht werden, denn die Befehlssyntax ist

SYSTEST<,FILSPEC<,SYSNR>>.

Wenn der zweite oder beide Befehlsparameter weggelassen werden, wird nach ihnen gefragt. Dabei wird FILESPEC in der üblichen Weise, z.B. auch mit Laufwerksnummer, eingegeben. SYSNR ist nur eine Dezimzahl für SYSx/SYS.

In der Bank 0 befinden sich nur die Files SYS1-29. Das Programm prüft daher auch, ob mit SYSNR eine zulässige Datei benannt wurde. Außerdem darf die Testdatei höchstens so lang sein wie die Zieldatei in Bank 0. Sonst würde das dahinter liegende SYS-File übesschrieben werden. In diesem Falle, wie überhaupt bei jedem denkbaren Fehler, bricht das Pragramm mit einer DOS-Fehlermeldung ab. Dies geschieht mit einem Sprung nach DOSERR (4409h), wo der Stack neu eingerichtet wird. Daher durfte getrost ignoriert werden, ob gerade ein PUSH oder CALL den Stack verbogen hat.

Auch hier sollen allzu weitschweifige Erklärungen des Programms vermieden werden. Die Kommentare mögen genügen. Für die ausgiebigen Aufrufe etlicher DOS-Routinen möchten wir auf Grossers "DOS-Buch" (5) verweisen, ohne das wir es übrigens wohl auch nicht geschafft hätten. Es sei nur darauf hingewiesen, daß wieder der DMA benutzt werden konnte, und daß ein komfortabler Befehl des Z180. MLT ww (3), das Umrechnen einer Dezimal- in eine Binärzahl erleichterte (zeile 31/32).

In der kommenden Zeit, liegt eine interessante Aufgabe in der weiteren Umgestaltung des Systems auf den Z180 hin. Wir würden uns freuen, wenn auch die Nachbauer des Z180-Boards gelegentlich mit Software im Club-Info vertreten wären. Vorschlag: Auch und gerade die RAM-Disk kann von der neuen CPU profitieren.

Literatur:

(1) Bernhardt, Sopp, Sonder-Info "Dein GIIIs, die 4 unbekannten Wesen", Michelbach,/Bilz 1986

(2) Bernhardt, "512k-RAM-Addressierung mit flexibler MMU", Club-Info 18, Michelbach/Bilz 1987

(3) Bernhardt, Schröder Sonder-Info "HD64180". Gschlachtenbretzingen 1989

(4) Schröder, "Z80, HD64180 und Illegals", Club-Info 17, Michelbach/Bilz 1987

(5) Grosser, "Das DOS-Buch", Aachen 1985

```
          00001 ;Listing 1:      Patch in SYS0/SYS, rel. Sektor 00h:
          00002 ;SYS0/SYS  konfiguriert den Z180 für G-DOS
          00003 ;FRS 006   (hier nicht interessierende Teile sind mit LIST OFF
          00004 ;DRS 05h   ausgespart - erkennbar an den springenden Zeilennummern)
          ; mit LIST OFF ausgespart
          00035
4010 3E00   00036 m4d10 LD    A,00h          ;A <- 00: Memory: 0 WAITs, I/O: 1 WAIT
4012 ED          00037      DB    0edh,39h,32h          ;OUT0(32h),A
4015 3E83   00038       LD    A,83h          ;RFSH alle 80 Zyklen 2 Zyklen lang
4017 ED          00039      DB    0edh,39h,36h          ;OUT0(36h),A
4D1A 21624E 00040       LD    HL,mmutab      ;Tabelle der MMU-Werte
4010 013803 00041       LD    BC,endtab-mmutab'256+38h ;3 Werte ab Port 38
4020 ED     00042       DB    0edh,93h       ;001110 (Werte ausgeben)
          ; mit LIST OFF ausgespart
          00208
          00209 ;Tabelle zum Einrichten der MMU des 2180
4E62 10          00210 mmutab     DB    10h              ;Common 1 ab     1.1000
4E63 10          00211      DB    10h            ;Bank-Area ab    1.4000
4E64 E4          00212      DB    0e4h           ;hierfür die logischen Adressen
          00213 endtab
00000 Fehler

          00001 ;Listing 2: Patch in SYS0/SYS im relativen Sektor 10h:
          00002 ;SYS0/SYS  stellt einen DMA-Treiber zur Verfügung
          00003 ;FRS 10h   (unteressierende Teile sind mit LIST OFF
          00004 ;DRS 15h   ausgespart - erkennbar an springenden Zeilennummern)
          00005
          ; mit LIST OFF ausgespart
          00037 ;Test auf Rechtspfeil: DOS banken oder nicht? (früher in SYS26/SYS)
3229 3A4038 00038       LD    A,(m3840)      ;Tastatur Steuerzeichen
322C E640   00039       AND   40h            ;Rechtspfeil gedrückt? (nein: DOS banken)
322E 3EFC   00040       LD    A,0fch                ;Requestcode für SYS26/SYS
3230 CCO244 00041       CALL  Z,m4402               ;mit OST 28h SYS26 aufrufen (DOS banken)
3233 217F32      00042 m322c LD    HL,m3275       ;Filename 'OVL4/SYS'
          ; mit LIST OFF ausgespart
          00081
          00082 ;Unterprogr. für DMA: HL = LSW Quelle, DE = LSW Ziel, BC = Bytezähler,
          00083 ;A Bit 0-3 = MSB der Quelle, A Bit 4-7 . MSB des Ziels
          00084 ;(MSB . most significant byte, A16-19, obere 4 Bits der Adresse,
          00085 ; LSW . less      "      word, A0-15, untere 16 "    "     "   )
          00086
35BB        00087       ORG   35bbh
35BB 22F035 00088 dma   LD    (srclog),HL    ;Quellbereich log. LSW x.HL
35BE ED53F335    00089      LD    (dstlog),DE    ;Zielbereich log. LSW x.DE
35C2 ED43F635    00090      LD    (bytcnt),BC    ;Bytezähler 0.BC
35C6 E5          00091      PUSH  HL
35C7 21F535 00092       LD    HL,dstphys     ;Zielbereich  Phys. MSB     Ah.DE
35CA 77          00093      LD    (HL),A                ;MSB setzen (016-19: Bit 4-7)
35CB ED67   00094       RRD                  ;Nibbles rotieren (A16-19: Bit 0-3)
35CD 32F235 00095       LD    (srcphys),A    ;Quellbereich phys. MSB     Al.HL
35D0 21F735 00096       LD    HL,tabend      ;Beginn der DMA-Wertetabelle
35D3 C5          00097      PUSH  BC
35D4 012708 00098       LD    BC,0827h       ;8 Werte ab Port 27 abwärts
35D7 F5          00099      PUSH  AF
35D8 DBFB   00100       IN    A,(0fbh)       ;Systembyte 2
35DA F5          00101      PUSH  AF                ;retten
35DB F640   00102       OR    40h            ;MMU des Z180 freigeben
35DD F3          00103      DI
35DE D3FB   00104       OUT   (Ofbh),A
          00105 ;           OTDMR ;DMAC aufsetzen
35E0 ED          00106      DB    0edh,9bh              ;(Hex-Entsprechung des Befehls)
35E2 013102 00107       LD    BC,0231h       ;2 Werte ab Port 31 abwärts
          00108 ;           OTDMR                 ;Rest aufsetzen u. DMA abschießen
35E5 ED          00109      DB    0edh,9bh
35E7 F1          00110      POP   AF                ;altes Systembyte 2
35E8 D3FB   00111       OUT   (0fbh),A

35EA F1          00112      POP   AF
35EB C1          00113      POP   BC
35EC E1          00114      POP   HL
35ED C9          00115      RET
          00116
          00117 ;Wertetabelle für den DMA-Controller (fallende Reihenfolge für OTDMR)
35EE 40          00118      DB    40h   ;30, DSTAT: Kanal 0, Zustand 'scharf'
```

```
35EF 02          00119       DB    02h   ;31, DMODE: Speicher zu Speicher, Burst-Modus
35F0 0000  00120 srclog      DW    0000h ;20/1, SAROL/H: Duelladresse LSW x.####
35F2 00          00121 srcphys    DB    00h   ;22, SAR0B: MS8  #.xxxx
35F3 0000  00122 dstlog      DW    0000h ;23/4, DAR0L/H: Zieladresse LSW   x.#1114
35F5 00          00123 dstphys    DB    00h   ;25, DAROB: MSB  #.xxxx
35F6 0000  00124 bytcnt      DW    0000h ;26/7, BCR0L/H: Bytezähler für die Programmlänge
35F7       00125 tabend      EQU   $-1   ;Ende der Tab., womit der Zeiger HL geladen wird
00000 Fehler

           00001 ;Listing 3: Patch in SYS26/SYS:
           00002 ;SYS26/SYS  neue Routine zum Puffern und Laden von SYS-Files
           00003 ;in Bank 0 des User-RAMs
           00004 ;(hier nicht interessierende Teile sind mit LIST OFF
           00005 ;ausgespart - erkennbar an den springenden Zeilennummern)
           00006
           ; mit LIST OFF ausgespart
35BB       00012 dma  EQU   35bbh ;Adresse des DMA-Treibers
           ; mit LIST OFF ausgespart
           00331
           00332 ;UP zum Übertragen von SYS-Files in die Bank 0
           00333 ;          E:    HL = Adresse RAM-DIR-Eintrag
           00334 ;                D  = Länge des Files in Sektoren
           00335 ;                E  = Adreß-MSB des Files in der Bank 0
           00336
4F1C E5          00337 putsys     PUSH  HL                ;39xx, DIR-Eintrag des akt. SYS-Files
4F1D D5          00338       PUSH  DE                ;E . MSB des SYS-Files, D . Länge in Skt.
4F1E C5          00339       PUSH  BC
4F1F F5          00340       PUSH  AF
4F20 73          00341       LD    (HL),E                  ;MSB der Zieladresse ins SYS-DIR
4F21 23          00342       INC   HL                ;nächste Stelle
4F22 72          00343       LD    (HL),D                  ;Zählbyte eintragen
4F23 210042 00344       LD    HL,m4200          ;Sektorpuffer
4F26 010001 00345       LD    BC,0100h          ;Länge 1 Skt.
4F29 53          00346       LD    D,E               ;DE <- Zieladresse (MSB) des Sektors
4F2A 59          00347       LD    E,C               ;(LSB immer =00)
4F2A 3E01   00348       LD    A,01h             ;von Bank 1 nach Bank 0
4F2D CDB835 00349       CALL  dma               ;Block übertragen
4F30 F1          00350       POP   AF
4F31 C1          00351       POP   BC
4F32 D1          00352       POP   DE
4F33 E1          00353             POP   HL
4F34 FB          00354       EI
4F35 C9          00355       RET
           00356
           00357 ;Einsprung bei Boot: RAM-Inhalt retten, SYS-Files in der Bank 0 puffern
4F36 210052 00358 boot LD    HL,m5200          ;Beginn des zu überschreibenden Bereichs
4F39 110008 00359       LD    DE,0800h          ;Zwischenpuffer dafür in Bank 1
4F3C 42          00360       LD    B,D               ;BC <- 0800, zu überschreibender Bereich
4F3D 48          00361       LD    C,E
4F3E E5          00362       PUSH  HL
4F3F D5          00363       PUSH  DE
4F40 3E11   00364       LD    A,11h             ;von Bank 1 nach Bank 1
4F42 CDB835 00365       CALL  dma               ;RAM-Inhalt retten
4F45 ED73914E    00366       LD    (m4f10),SP        ;alten Stach retten
4F49 CDCB4E 00367       CALL  m4f55             ;DIR-Sektoren mit SYS-Entries laden
4F4C 210039 00368       LD    HL,m3900          ;vorläufiger SYS-DIR-Puffer
4F4F 110139 00369       LD    DE,m3900+1        ;zum Ausnullen vorbereiten

4F52 78          00370       LD    A,E               ;A <- 01, Zähler für 1 Durchgang
4F53 32074F 00371       LD    (filelen),A       ;dort ablegen
4F56 0E06   00372       LD    C,06h             ;Zähler 6 Bytes (f3 = 00 seit m4f55)
4F58 75          00373       LD    (HL),L                  ;1. Byte des SYS-DIR-Puffers ausnullen
4F59 EDB0   00374       LDIR                    ;Rest ausnullen (hier schneller als DMA)
4F5B 1E41   00375       LD    E,41h             ;ab Page 4100 die SYS-Files puffern
           ; mit LIST OFF ausgespart
4FA0 E1          00407       POP   HL                ;Adresse des gepufferten RAM-Inhalts
4FA1 D1          00408       POP   DE                ;seine frühere Ladeadresse
4FA2 44          00409       LD    B,H               ;BC <- 0800h, Blocklänge (C=00 seit LDIR)
4FA3 3E11   00410       LD    4,11h             ;von Bank 1 nach Bank 1
4FA5 CDBB35 00411       CALL  dma               ;RAM-Inhalt ab 5200h restaurieren
4FA8 210039 00412       LD    HL,m3900          ;Puffer für SYS-DIR
4FAB 110040 00413       LD    DE,m4000          ;endgültige Adresse dafür in Bank 0
4FAE 0601   00414       LD    B,01h             ;Länge 256 Bytes (C = 00)
4F80 AF          00415       XOR   A                 ;A - 00: Bank 0 -> Bank 0, 'kein Fehler'
4F81 C3BB35 00416       JP    dma               ;Block übertragen und zurück ins DOS
```

```
            00417
            00418 ;Patch im DOS-Kern: Einsprung bei DOS-Request mit RST 28h
03D03       00419 offs4 EAU   $-4be1h                ;Abstand zur Ladeadresse
4FB4 3E40   00420 m4be1 LD    A,m4000/256      ;DIR des gebankten DOS (Adreß-MSB)
4F66 323D37 00421       LD    (addrmsb-offs3),A      ;in Laderoutine patchen
4FB9 CD3837 00422       CALL  m3738-offs3      ;in den Sektorpuffer übertragen
4FBC 2642   00423       LD    H,m4200/256      ;Sektorpuffer (MS8)
4FBE 341743 00424       LD    4,(4317h)        ;SYS-Kenner
4FC1 07           00425       RLCA                   ;2 (wegen 2 Bytes pro Eintrag)
4FC2 6F           00426       LD    L,A              ;als LSB des Zeigers laden
4FC3 7E           00427       LD    A,(HL)                 ;Adreß-MSB des SYS-Files in Bank 0
4FC4 323D37 00428       LD    (addrmsb-offs3),A      ;in die Laderoutine patchen
4FC7 23           00429       INC   HL               ;auf Längenbyte stellen
4FC8 7E           00430       LD    A,(HL)                 ;laden
4FC9 B7           00431       OR    A                ;= 00? (dann INHALT/SYS oder SYSO/SYS)
4FCA 2815   00432       JR    Z,m4c0e                ;Fehler, falls ja (nicht vorhanden)
4FCC 213837 00433       LD    HL,m3738-offs3         ;modifizierte Sektor-Laderoutine
4FCF 226D4C 00434       LD    (4c6dh),HL       ;in SYSO patchen
4FD2 CD2B4C 00435       CALL  4c2bh            ;SYS-File laden
4FD5 221E4C 00436       LD    (4c1eh),HL       ;Startadresse des SYS-Files dort patchen
4FD8 213644 00437       LD    HL,4436h         ;alte Sektor-Laderoutine
4FD8 226D4C 00438       LD    (4c6dh),HL       ;restaurieren
4FDE 00           00439       NOP                    ;Füller bis zur alten Fortsetzung
4FE1        00440 m4c0e EQU   §+2
            00441
            00442 ;neue Laderoutine für SYS-Files
18A7        00443 offs3 EQU   $-3738h
4FDF E5           00444 m3738 PUSH  HL
4FE0 D5           00445       PUSH  DE
4FE1 C5           00446       PUSH  BC
4FE2 210000 00447       LD    HL,0000h         ;Duellblock in Bank 0
4FE4        00448 addrmsb     EQU   $-1              ;MSB wird jeweils gepatcht
4FE5 7C           00449       LD    A,H              ;Adreß-MSB
4FE6 3C           00450       INC   A                ;100h Bytes weiter
4FE7 323D37 00451       LD    (addrmsb-offs3),A      ;MSB für nächsten Block patchen
4FEA 3E10   00452       LD    A,10h            ;aus Bank 0 nach Bank 1
4FEC 110042 00453       LD    DE,m4200         ;Sektorpuffer
4FEF 010001 00454       LD    BC,0100h         ;Sektorlänge
4FF2 CDBB35 00455       CALL  dma              ;Block übertragen
4FF5 C1           00456       POP   BC
4FF6 D1           00457       POP   DE
4FF7 E1           00458       POP   HL
4FF8 AF           00459       XOR   A                ;Flag 'kein Fehler'
4FF9 C9           00460       RET
00000 Fehler

            00001 ;Listing 4:      Einige Änderungen für den Z180 im BASIC-Interpreter
            00002 ;TRAP/CMD        Hauptsächlich Delay und Illegal-TRAP-Routine
            00003 ;               (Gerals Schröders TRAP-Routine idt mit LIST-OFF aus-
            00004 ;               gespart, nachzulesen Club-Info Nr. 17, nur ihr
            00005 ;               GIIIs-spezifischer Anfang ist gelistet
            00006
0000        00007 target1     EQU   0000h            ;Zieladresse der Trap-Routine in Bank 1
1650        00008 target0     EQU   1650h            ;dto. in Bank 0, vorläufig im BASIC Bereich
35BB        00009 dma   EQU   35BB             ;Aresse des DMA-Treibers
            00010
5200        00011       ORG   5200h
            00012
            00013 ;Trap-Einsprung für Adresse 9
5200 00           00014 lowmem      NOP                    ;Platz für eventl. F7 (RST 30, Debugger)
5201 C35016 00015       JR    target0                ;Einsprung bei Trap-Interrupt
            00016
            00017 ;Einsprung der Warteschleife und NMI-Routineab Adresse 0060
5204 0B           00018 delay DEC   BC               ;(alter Teil der Delay-Routine
5205 78           00019       LD    A,B              ;bleibt unverändert)
5206 B1           00020       OR    C
5207 C8           00025       Ret   Ẑ                ;neu: wird immer bearbeitet (Verrzögerung)
5208 18FA   00022       JR    delay            ;jetzt Laufzeit etwa wie Z80
            00023
520A FB           00024 nmi   EI                     ;NMI: Interrupt zulassen
520B C9           00025       RET
            00026
            00027 ;Fehlerbehandlung bzw. Illegal-Simulation nach einem Trap-Interrupt
            00028 ;vorläufige Version: kein anderer RST 00h außer TRAP berücksichtigt!
388C        00029 offs  EQU   $-target0        ;Abstand zur Arbeitsadresse
```

```
520C F3           00030 trap  DI                        ;Interrupt aus
520D F5           00031       PUSH  AF                  ;Akku und Flags retten
520E DBFA   00032       IN    A,(0fah)          ;System-Byte 1
5210 32FA16 00033       LD    (fsbuff-offs),A         ;puffern
5211        00034 addr1 EQU   $-2
5213 E6DF   00035       AND   0dfh              ;evtl. Schreibschutz aufheben
5215 03FA   00036       OUT   (0fah),A          ;System-Byte neu ausgeben
            00171
52C7 DBFA   00172 start IN    A,(0fah)          ;System-Byte 1
52C9 E6DF   00173       AND   0dfh              ;evtl. Schreibschutz aufheben
52CB F3           00174       DI
52CC D3FA   00175       OUT   (0fah),A          ;neu ausgeben
52CE 240052 00176       LD    HL,(lowmem)       ;Code für Adresse 0000h
52D1        00177       LD    (0000h),HL        ;am Zielort ablegen
52D4 1A0202 00178       LD    HL,(lowmem+2)           ;Fortsetzung, insgesamt 4 Byte
52D7 220200 00179       LD    (0003h),HL
52DA 52DD   00180       LD    HL,delay          ;neue Verzögerungs-Routine
52DD 116000 00181       LD    DE,0060h          ;Zieladresse
52E0 010B00 00182       LD    BC,trap-delay           ;Länge
52E3 EDB0   00183       LDIR                    ;(wenige Bytes) LDIR schneller als DMA
52E5 115016 00184       LD    DE,target0        ;Zieladresse (HL bereits auf Quelle trap)
52E8 018800 00185       LD    BC,atart-trap           ;Länge
52EB 3E01   00186       LD    A,01h             ;von Bank1 nach Bank0
52ED CDBB35 00187       CAll  dma               ;Block übertragen (schneller als LDIR)
52F0 060F   00188       LD    B,ebdtab-addrtab/2      ;Anzahl zu relozierender Abressen
52F2 311E53 00189 adrlop      LD    SP,addrtab        ;Stack auf Adressentabelle
52F3        00190 spbuff      EQU   $-2
52F5 E1           00191       POP   HL                ;eine Adresse abholen
52F6 ED73F352     00192       LD    (spbuf),SP        ;Tabelle als Stack retten
52FA F9           00193       LD    SP,HL             ;dortige Adresse = Stack
52FB E1           00194       POP   HL                ;dortiges Word abholen
52FC ED52   00195       SBC   HL,DE             ;auf neue Basisadresse umrechnen
52FE E5           00196       PUSF  HL                ;neues Woed ablegen
52FF 10F1   00197       DJNZ  addrlop                 ;bis alle Adressen bearbeitet
5301 210C52 00198       LD    HL,trap                 ;Adresse der Trap-Routine

5304 F9           00199       LD    SP,HL             ;unterhalb neuer Stack
5305 110000 00200       LD    DE,target1        ;Zieladresse in Bank 1
5308 01BB00 00201       LD    BC,start-trap           ;Länge der TRAP-Routine
530B 3E11   00202       LD    4,11h             ;von Bank 1 nach Bank 1
530D CDBB35 00203       CALL  dma               ;TRAP-Routine nach Bank 1 verladen
5310 211653 00204       LD    HL,ramdisk        ;Aufruf der RAM-Disk (kann entfallen)
5313 C30544 00205       JR    4405h             ;als DOS-Befehl ausführen und ENDE
            00206
5316 52           00207 ramdisk    DM    'RAMDISK',Odh          ;Page des RAM-Disk-Programms, CR
            00208
            00209 ;Tabelle der zu relozierenden Adressen in der TRAF-Routine
531E 1152   00210 addrtab    DW    addr1,addr2,addr3,addr4,addr5,addr6,addr7,addr8
532E 7352   00211      DW    addr9,addr10,addr11,addr12,addr13,addr14,addr15
            00212 endtab
00000 Fehler

            00001 ;Listing 5:      Ein geändertes SYS-File soll vor dem endgültigen Gebrauch
            00002 ;SYSTEST/CMD     in Bank 0 gepuffert und von dort aus getestet werden.
            00003
6000        00004       ORG   6000h
            00005
35BB        00006 dma   EQU   35bbh             ;DMA-Treiber
            00007
6000 340B4C 00008 start LD    4,(4c0bh)         ;bei depuffertem DOS steht dort 00
6003 87           00009       OR    A                 ;ist das DOS in Bank 0 schon gepuffert?
6004 2807   00010       JR    Z,go_baby         ;alles klar, falls ja
6006 E5           00011       PUSH  HL                ;Befehlszeiger retten
6007 3EFC   00012       LD    A,0fch                  ;Requestcode für SYS26: DOS erst bunkern
6009 CD0244 00013       CALL  4402h             ;dort steht RST 28h für den DOS-Request
600C E1           00014       POP   HL                ;Befehlszeiger
600D CDD54C 00015 go_baby     CALL  4cd5h             ;HL auf nächsten Befehlsparameter stellen
6010 2006   00016       JR    NZ,makefcb        ;falls etwas folgt
6012 11AF60 00017       LD    DE,filespc        ;Text 'Name ..."
6015 CD9560 00018       CALL  getinpt                 ;ausgeben, Dateinamen einlesen
6018 11E060 00019 makefcb     LD    DE,fcb                  ;FCB-Adresse
601B CD1C44 00020       CALL  441ch             ;Dateinamen in den FCB übertragen
601E CD2044 00021       CALL  4420h             ;FCB eröffnen
6021 206D   00022       JR    NZ,errexit        ;falls Fehler aufgetreten
6023 CDD54C 00023       CALL  4cd5h             ;HL auf nächsten Befehlsparameter stellen
```

```
6026 2006          JR    NZ,getsys         ;falls etwas folgt
6028 11C960        LD    DE,sysmbr         ;Text 'SYS-File ..."
602B CD9560        CALL  getinpt                 ;ausgeben, SYS-Nr. einlesen
602E CD9F60  getsys      CALL  getciph                 ;1. Ziffer binär umwandeln
6031 78            LD    A,8               ;Binärziffer
6032 280B          JR    Z,putsys          ;falls Eingabe beendet (SYS1-9)
6034 0E0A          LD    C,0ah             ;Faktor 10d für zweistellige Zahlen
             ;     MLT   BC                ;BC = B *  C: B <- 00, C <- 10*B
6036 ED            DB    0edh,4ch          ;Hex-Entsprechung des Befehls
6038 CD9F60        CALL  getciph           ;2. Ziffer einlesen (muß letzte sein)
603B 2053          JR    NZ,errexit        ;falls Eingabe noch nicht beendet
603D 78            LD    A,8               ;neue Ziffer (binär)
603E 81            ADD   A,C               ;+ 10mal 1. Ziffer
603F 3C      putsys      INC   A                 ;+2, weil GDOS und INHALT keine Nr. haben
6040 3C            INC   A
6041 FE03          CP    03h               ;SYS1? ('kleinste' erlaubte SYS-Datei)
6043 3849          JR    C,illnmbr         ;Fehler: Eingabe von SYSO
6045 FE20          CP    1fh+1             ;SY529? ('höchste' erlaubte SYS-Datei)
6047 3045          JR    NC,illnmbr        ;falls Quatsch wie etwa SYS4711
6049 07            RLCA                    ;'2, weil 2 Bytes pro Eintrag im SYS-DIR
604A 08            EX    AF,AF'                  ;retten
604B 210040        LD    HL,4000h          ;dort in Bank 0 das SYS-DIR
604E 110080        LD    DE,8000h          ;Puffer dafür in Bank 1

6051 05            PUSH  DE                ;für später
6052 45            LD    B,L               ;BC <- 0040h = 64d, Länge des SYS-DIR
6053 4C            LD    C,H
6054 3E10          LD    A,10h             ;von Bank 0 nach Bank 1
6056 CDBB35        CALL  dma               ;SYS-DIR in den Puffer übertragen
6059 08            EX    AF,AF'                  ;aufbereitete SYS-Nr.
605A 5F            LD    E,A               ;DE <- Adresse des Eintrags im SYS-DIR
605B 1A            LD    A,(DE)                  ;MSB der Adresse des SYS-Files in Bank 0
605C 328560        LD    (sysaddr),A       ;dort ablegen
605F 13            INC   DE                ;auf Längenbyte (Blockzähler) stellen
6060 14            LD    A,(DE)                  ;laden
6061 5F            LD    E,A               ;DE <- Länge des SYS-Files (LSB)
6062 55            LD    D,L               ;MSB . 00 (max. 35 Sektoren bei SYS6)
6063 2AEC60        LD    HL,(fcb+0ch)      ;2. u. 1. EOF-Byte, gibt Länge in Skt. an
6066 7D            LD    A,L               ;Zähler für zu ladende Skt. (LS8 genügt)
6067 3C            INC   A                 ;0.1, weil ab 0 gezählt, Cy lösch. wg. SBC
6068 47            LD    B,A               ;B als Zähler aufsetzen
6069 328860        LD    (bytecnt),A       ;als DMA-Bytezähler (MSB) patchen
606C E052          SBC   HL,DE             ;beide Filelängen vergleichen
606E 3E3B          LD    A,3bh             ;Fehlercode 'unzureich. Speicherplatz'
6070 301E          JR    NC,errexit        ;falls Testfile zu lang
6072 E1            POP   HL                ;Puffer ab 8000h in Bank 1
6073 11E060        LD    DE,fcb            ;Zeiger auf den FCB
6076 22E360  ldloop      LD    (fcb+03h),HL            ;Pufferadresse in den FCB schreiben
6079 24            INC   H                 ;Pufferzeiger 256 Bytes weiterstellen
607A CD3644        CALL  4436h             ;einen Sektor einlesen
607D 2011          JR    NZ,errexit        ;falls ein Fehler auftrat
607F 10F5          DJNZ  ldloop            ;bis alle Sektoren eingelesen sind
6081 2680          LD    H,8000h/256       ;HL <- Puffer 8000h in Bank 1 (LSB . 00)
6083 110000        LD    DE,0000h          ;Zieladresse in Bank 0 (LSB immer 00)
6085         sysaddr     EQU   $-1               ;MSB der SYS-Adresse in Bank 0
6086 010000        LD    BC,0000h          ;Bytezähler (LSB immer 00)
6088         bytecnt     EQU   $-1               ;MSB wird gepatcht
6089 3E01          LD    A,01h             ;von Bank 1 nach Bank 0
6088 C3BB35        JP    dma               ;neues SYS-File speichern und ENDE

608E 3E29    illnmbr     LD    A,29h             ;Fehlercode 'unzul. log. Dateinummer'
6090 E67F    errexit     AND   7fh               ;NZ-Bedingung (Fehler), keine Rückkehr
6092 C30944        JP    4409h             ;Fehlermeldung ausgeben und ENDE

6095 EB      getinpt     EX    DE,HL             ;HL - Zeiger auf Text
6096 CD6744        CALL  4467h             ;Text anzeigen
6099 0620          LD    B,20h             ;max. 32d Zeichen (sehr reichlich)
609B EB            EX    DE,HL             ;HL - DOS-Eingabepuffer für Eingabe
609C C34000        JP    0040h             ;Eingabe einlesen und zurück

609F 7E      getciph     LD    A,(HL)                  ;Zeichen laden
60A0 FE30          CP    '0'               ;kleinste erlaubte Ziffer?
60A2 38EA          JR    C,illnmbr         ;Fehler, falls kleiner
60A4 FE3A          CP    '9'+1             ;größte erlaubte Ziffer?
60A6 30E6          JR    NC,illnmbr        ;Fehler, falls größer
```

```
60A8 E60F  00098       AND   0fh              ;ASCII -) binär
60AA 47         00099       LD    B,A              ;Eingabe binär puffern
60AB 23         00100       INC   HL               ;Befehlszeiger nachstellen
60AC C3D54C 00101       JP    4cd5h            ;nächsten Befehlsparam. holen und zurück
           00102
           00103 ;verschiedene Texte und Puffer
60AF 4E         00104 filespc DM 'Name des Testprogramms? ',0eh,03h      ;Text, Churs., EOS
60C9 53         00105 sysnsbr DM 'SYS-File-Nr.',09h,'(1-29)? ',0eh,03h   ;dto. mit TAB
           00106 fcb                               ;Puffer für FCB
           00107
6000       00108       END   start
00000 Fehler
```

Anhang

Probleme mit den Illegals

Geralds Trap-Programm ist eine wertvolle Hilfe, dennoch kein Allheilmittel. Von den vielen denkbaren Illegals, die im Code-Muster des Z80 stecken, sind längst nicht alle bereits entschlüsselt und dokumentiert. Die Trap-Routine müßte daher bei jeder neuen Entdeckung aktualisiert werden.

Und der Z180 benutzt selber für seine neuen Befehle solche ehemals weißen Stellen in der Code-Matrix des Z80. Glücklicherweise beschieden sich seine Designer mit solchen Codes, die mit dem Switch-Byte EDh beginnen, so daß ein notwendiges Debugging verdächtiger Programme in Grenzen bleibt. Nämlich auch Illegals mit EDh werden schon irgendetwas beim Z80 bewirken, was auch immer. Die Illegal-Trap-Routine würde in diesem Falle überhaupt nicht angesprungen, denn es wäre für den Z180 ja ein sinnvoller Befehl.

Einer dieser beiden Gründe wird es wohl sein, weshalb nach der Umrüstung des G3s auf den Z180 ausgerechnet mit ZEUS/CMD zahlreiche Abstürze auftraten. Aber vielleicht wurde auch Geralds Programm falsch abgetippt. Egal - es mußte etwas geschehen, denn mit ZEUS steht und fällt die Soft-Seite des ganzen Unternehmens:

DSMBLR ist ein Disassembler, der gottlob keine Illegals akzeptiert, wie es z. B. ZEUS durchaus tut. Wenn man daher ein Programm disassembliert, das Illegals enthält, zuckt DSMBLR ratlos die Achseln und setzt DM-(DEFM-) Statements, wo er in seinem Vorrat keinen Z80-Befehl findet (DSMBLR und ZEUS behandeln DB und DM gleich, soweit es sich um einzelne Bytes handelt).

So haben wir ZEUS ohne jegliches Screening disassembliert und nach DMs gesucht. An vier Stellen wurden wir fündig (Listing 6). In einem gesonderten File, mit dem wir seit jeher ZEUS patchen (ZAPZEUS/CMD, Listing 7), wurden Simulationen der Illegals programmiert, in ZEUS selbst CALLs nach dort gepatcht.

ZAPZEUS (für diesen Zweck genügen die gelisteten Teile) kann so abgetippt und auf Papier assembliert werden. Die Bestandteile bis Zeile 46 werden etwa mit Hilfe von FED/CMD, mitten in ZEUS gezapt. Was dahinter liegt, muß im letzten Sektor von ZEUS angehängt werden. Der Ende-String (02-02-xy-xx) wird als neuer Record-Header (01-nn-yy-xx) geändert. Das Anhängsel-Programm bringt seinen eigenen Ende-String mit. Dafür ist es wichtig, daß seine Einsprungsadresse (END addr) die von ZEUS sein muß (s. Listing).

Ob damit alles erledigt ist, muß die zukünftige Praxis zeigen. Ohne ein Screening-File für DSMBLR werden nämlich Texte und Tabellen wie Programmcode disassembliert. So ergeben sich Verschiebungen im "Programm"-Ablauf, die etwa ein Illegal-Switch-Byte wie einen harmlosen Bestandteil eines Sprungbefehls (Adresse, Distanz) erscheinen lassen können. Da das aber nur auftreten kann, wenn ein Illegal direkt an einen reinen Datenblock grenzt, darf man mit Recht auf Erfolg hoffen.

Auch das Gegenteil wäre eine mögliche Fehlerquelle: Eine Tabelle, die ohne Screening ja nicht als solche behandelt wird, enthält ein verdächtiges Byte. Der Zufall will es, daß der Inhalt der Tabelle wie sinnvoller Programmcode aussieht. Ein dort gepatchter CALL würde die Tabelle verfälschen. Wir können nur raten, in solchen Fällen die betreffenden Stellen ganz besonders sorgfältig zu untersuchen ...

... und unbedingt das Ergebnis zur Veröffentlichung an Jens-80 Neueder zu schicken!

```
           00001 ;Listing 6:      Illegals in ZEUS/CMD
           00002 ;ZEUS/CMD  (hier nicht interessierende Teile sind mit LIST OFF
           00003 ;    ausgespart - erkennbar an den springenden Zeilennummern)
           00004
           ; mit LIST OFF ausgespart
52E7 DD         00305       DM    Oddh        ;DD 7D:'
52E8 7D         00306       LD    A,L         ;LD A,LX
52E9 FE28  00307       CP    '+'
           00308
5809 CB         00694       DM    0cbh        ;CB 31 06 01:
580A 310601 00695       LD    SP,m0106   ;SLIA C - LD 8,01
           00696
5045 C8         00731       DM    0cbh        ;C3 31 06 01:
5846 310601 00732       LD    SP,m0106   ;SLIA C - LD 8,01
           00733
64E9 CB         00700       DM    0cbh        ;CB 37:
6AEA 37         00701       SCF               ;SLIA A
           00702
00000 Fehler
           ; mit LIST OFF ausgespart
           00001 ;Listing 7:      Illegal-Trap direkt in ZEUS/CMD
           00002 ;ZAPZEUS/CMD     (hier nicht interessierende Teile sind mit LIST 0FF
           00003 ;ausgespart - erkennbar an den springenden Zeilennummern)
           ; mit LIST OFF ausgespart
           00034
52E7       00035       ORG   52e7h       ;dort Illegal LD A,LX
52E7 CD2830 00036       CALL  ld_a_1x          ;simulieren
           00037
```

```
5809        00038       ORG   5b09h       ;dort Illegal SLIA C
5809 CD3330 00039       CALL  slia_c            ;simulieren
            00040
5845        00041       ORG   5b45h       ;dort Illegal SLIA C
5845 CD3330 00042       CALL  slia_c            ;simulieren
            00043
6AE9        00044       ORG   6ae9h       ;dort Illegal SLIA A
6AE9 37           00045       SCF               ;Cy für Bit 0 setzen
6AEA 17           00046       RLA               ;rotieren
            ; mit LIST OFF ausgespart
            00076
3000        00077       ORG   3000h       ;wo es nicht stört
            ; mit LIST OFF ausgespart
            00106
3028 DDE5   00107 ld_a_1x    PUSH  IX          ;auf den Stack
302A E3           00108       EX    (SP),HL           ;HL retten, HL - IX
3028 7D           00109       LD    A,L         ;A - LX
302C E3           00110       EX    (SP),HL           ;HL und IX restaurieren
302D DDE1   00111       POP   IX
302F F2BH   00112       CP    2bh         ;überschriebenen Befehl nachholen
3031 1B05   00113       JR    return            ;zurück zur Fortsetzung
            00114
3033 CB21   00115 slia_c     SLA   C           ;C shiften
3035 0C           00116       INC   C           ;Bit 0 setzen
3036 0601   00117       LD    B,01h       ;überschriebenen Befehl nachholen
            00118
3038 E3           00119 return     EX    (SP),HL           ;HL - RET-Adresse
3039 23           00120       INC   HL          ;ein Byte später
303A E3           00121       EX    (SP),HL          ;zurücktauschen
303B C9           00122       RET               ;an die neue Adresse zurückkehren
            00123
5312        00124       END   5312h       ;hier Einsprung in ZEUS
00000 Fehler
```

Erratum

Helmut Bernhardt

Eine ganzen Menge Artikel über die Verwendung von 41256er RAMs in Z80-Systemen baut auf die Erweiterung des Refresh-Zählers auf 8 Bit auf. Der Z80 Prozessor liefert einen 7-Bit Refresh-Zähler, der bei einem 64 kB Adressraum ausreicht und die meisten 4164 RAMs auch frisch hält. Es gab in den frühen 80er Jahren aber auch 4164 RAMs, die einen 8-Bit Refreshzähler benötigten. Um solche RAMs trotzdem mit dem Z80 benutzen zu können, entwickelte Michael Hungershausen zwei Schaltungen, die den Refresh-Zähler um eine Adresse erhöhen können. Diese Schaltungen fügen mit A7 die achte Refresh Adresse zu A0-A6 hinzu (aus A6 und /RFSH).

8-Bit-Refresh für Z80, Michael Hungershausen, c't 1984, Heft 9, S. 22

Analog kann man mit diesen Schaltungen auch 41256er RAMs, die grundsätzlich 8-Bit Refreshzähler benötigten, benutzen. Die eine Variante benutzt NAND- und die andere NOR-Gatter. In den Artikeln, die diese Schaltung nutzten, wurde die Variante mit den NAND-Gattern benutzt. Alexander Schmid hat in dieser Schaltung aus der Zeitschrift c't einen Druckfehler entdeckt und dies im Club80 Info Nr. 30 veröffentlicht.

Die Schaltung aus der c't wurde in diesen Projekten verwendet:
256K für Z80-Systeme
1M-Byte im GENIE III
TRS80 Japan Modell aufbohren
Dein GIIIs die 4 unbekannten Wesen

8-Bit-Refresh für Z80

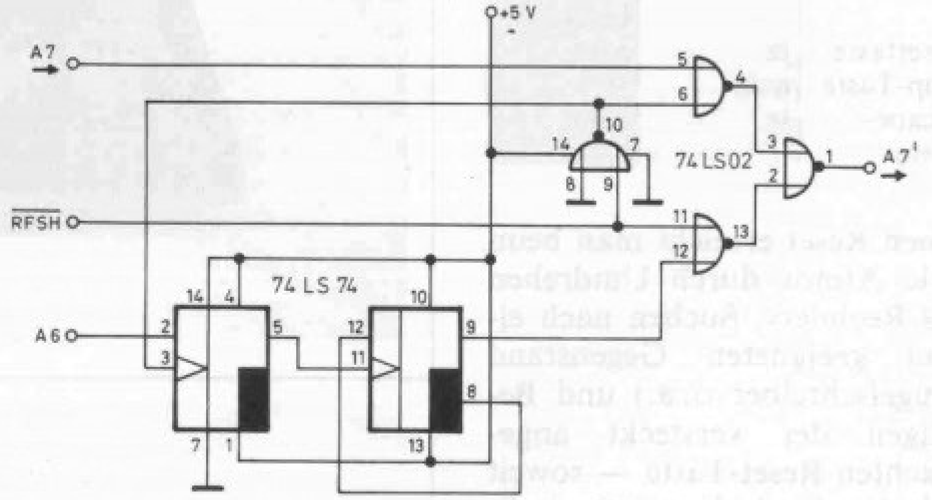
Michael Hungershausen

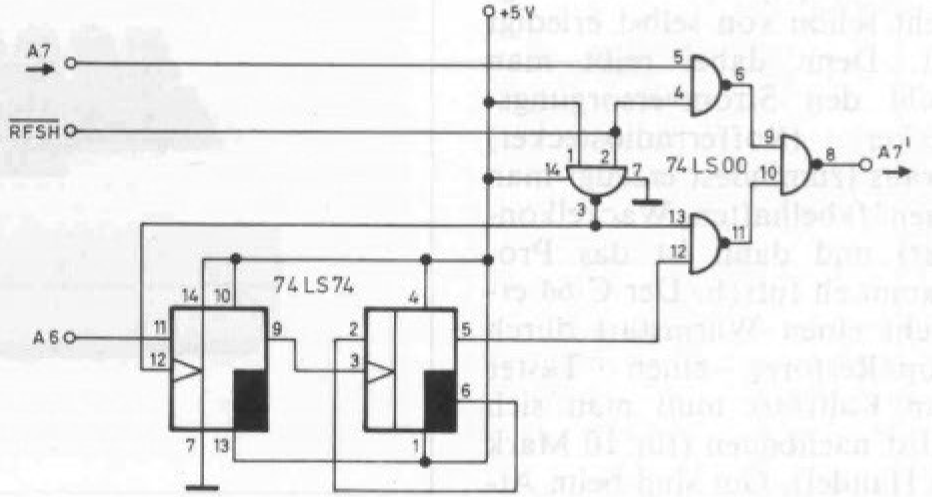
In einer Reihe von — nicht nur — Z-80-Computern wurde bis vor wenigen Jahren der RAM-Bereich von zumeist 16 KBytes durch die Benutzung der dynamischen RAMs 4116 realisiert. Mit der Entwicklung von 64-KBit-Bausteinen zu erschwinglichen Preisen ist hier eine Erweiterung auf den vierfachen Speicherbereich mit geringstem Aufwand möglich geworden.

Das werden — wie ich — viele Computerbesitzer ausprobiert haben. Manche aber werden — ebenfalls wie ich — die herbe Enttäuschung erlebt haben, daß exakt die Hälfte des 64-KByte-Speichers die gespeicherten Daten innerhalb von etwa 0,25 bis 60 Sekunden wortwörtlich vergaß. (Erstaunlich langsam, wenn man bedenkt, daß der Refresh alle zwei Millisekunden erfolgen soll.) Meine Speicherbausteine trugen den Stempel der Firma Siemens; ICs von TI machen den gleichen Zirkus: Sie erwarten im Unterschied zu den alten 4116ern und den meisten neuen 4164-ICs eine Auffrischung von 256 Zellen je Zyklus, also eine 8 Bit breite Refresh-Adresse. Die CPU Z 80 dagegen hat nur einen 7-Bit-Refresh-Zähler. Um nicht noch einmal rund 150 Mark für neue Speicher auszugeben, entwickelte ich eine simple Schaltung zur Erweiterung des Z-80-Refreshs um ein Bit.

Nur zwei ICs

Mit dem Flipflop FF1 wird das höchste Bit der Refresh-Adresse (A6) mit der fallenden Flanke von RFSH zwischengespeichert. Nachgeschaltet ist ein Teiler durch 2 zur Erzeugung von A7. Wenn das RFSH-Signal aktiv ist, schaltet der aus den vier Gattern bestehende Multiplexer anstatt der 'echten' Adresse den Teilerausgang auf die Leitung A7. Um eventuellen Bastelkistenbeständen Rechnung zu tragen (die Bauteileversorgung ist ja im Moment wirklich grausig), seien hier zwei Versionen mit wahlweise einem NAND- oder einem NOR-Baustein gezeigt. Beide Schaltungen sind getestet und laufen seit etwa einem Jahr problemlos in VIDEO GENIES. Ein Wort noch zum Einbau: Ich habe die TTL-Bausteine huckepack auf andere TTL-ICs gelötet und eine Freihandverdrahtung vorgenommen. Es geht schnell und sieht phänomenal interessant aus . . . □





Club 80 Info Nr.30 S.30

Nachtrag zum Banker

Alexander Schmid

Sollte es wahr sein, daß bis jetzt außer Arnulf Sopp und mir noch niemand Probleme mit dem Banker von Helmut Bernhardt hatte? Keine unmotivierten Abstürze oder Aufhänger? Na gut. Vielleicht hängt das Auftreten auch etwas mit der Taktgeschindigkeit zusammen und die ist bei uns halt etwas höher. Trotzdem hat die Schaltung einen echten Fehler, den ich hier mit freundlicher Genehmigung von Helmut breittreten darf.

Nach über einem Jahr Suchen und Ärgern habe ich ihn endlich gefunden. Der Käfer besteht darin, daß am 74LS74 auf dem Banker die Pins 11 und 12 vertauscht sind, was auf einen Druckfehler in einer alten c't zurückgeht, aus der Helmut einen Teil der Schaltung übernommen hat. Da vergleicht man immer wieder das Layout mit dem Schaltplan, wundert sich, daß alles

stimmt und trotzdem nichts geht und dann sowas. Diese Verwechselung hat die unangenehme Folge, daß der Refresh für die dynamischen RAMs nicht richtig funktioniert und die Bits unter Umständen nach Lust und Laune umkippen. Beim TRS80/ Genie unter NEWDOS und GDOS macht sich das im Normalfall nicht bemerkbar, weil der Prozessor dort durch die Interrupts kreuz und quer durch's RAM springt und so schon alleine durch die Speicherzugriffe während der Interruptroutinen für eine ausreichende Erfrischung sorgt. Wenn die Interrupts aber abgeschaltet sind, was z.B. bei mir unter CP/M der Fall war, kann es Ärger geben, wenn z.B. eine Schleife zu kurz wird.

Klarheit kann man sich mit folgendem Programm schaffen:

```
        ORG     6000h           ;oder sonstwo
start   DI                      ;keinen Mogelrefresh machen
        LD      A,8             ;Zähler
        LD      B,A
        LD      C,A
Loop    INC     C
        JR      NZ,loop         ;innere Schleife
        INC     B
        JR      NZ,loop
        INC     A
        JR      NZ,loop         ;äußere Schleife
        JP      4400h           ;ins DOS
        END start
```

Durch die kurzen Schleifen bleibt der Speicherzugriff auf wenige Adressen beschränkt und der ganze Rest kann nur durch die Logik auf dem Banker behandelt werden. Wenn sich der Banker nach 1-2 Wochen noch nicht wieder gemeldet hat, könnt Ihr davon ausgehen, daß im restlichen RAM nur noch Schrott steht, womit sich auch das DOS verabschiedet hat. Nach dem Neubooten dürfte dann überall nur noch wirres Zeug stehen, was Ihr mit Debug o.ä. überprüfen könnt. Wenn alles stimmt, sollte nach ungefähr zwei Minuten wieder das DOS-Ready erscheinen und der Rest des Speicherinhalts unverändert sein. Wem das zu lange dauert, der kann A mit einem größeren Wert laden, wobei es aber auch wieder nicht zu schnell gehen darf, weil die RAMs, wenn sie einem guten Tag haben, durchaus ein paar Sekunden stabil sein können. Die paar Minuten sollte das doch wert sein. Zur Überprüfung könnt Ihr vorher noch irgendein Programm in die RAMdisk kopieren, die ja denselben Speicher benutzt. Wenn das Programm nachher noch läuft, ist mit ziemlicher Sicherheit alles in Ordnung. Wenn nicht, solltet Ihr schnellstens den Lötkolben heizen.

Die internen Ports des HD64180

Helmut Bernhardt

Sämtliche gegenüber dem Z80 zusätzlich im HD64180 zur Verfügung gestellten Leistungen sind über Portadressen zu programmieren. Für diese internen Ports der CPU wird ein I/O-Adreßbereich von 3FH Ports verbraucht.

Wie prinzipiell auch schon beim Z80 kann mit dem HD64180 ein I/O-Adreßraum von 64K-Ports differenziert werden. Beim Z80 werden dafür ausschließlich die Befehle IN A,(C) und OUT (C),A sinnvoll sein. Bei diesen Befehlen werden die Inhalte der Register C (AO-A7) und B (A8-A15) als Adressen auf den Bus gelegt. Bei den normalen In- und OUT - Befehlen mit Ausgabe einer 8Bit-Portaddesse im Befehlsoperanden wird der Inhalt des Accu auf A8-A15 des Adressbus gelegt. Diese Befehle sind in Sytemen, deren I/O-Adreßdecoding auf 16Bit-Adressen aufbaut nicht verwendbar.

Die internen Ports des HD64180 liegen in der Zero-Page (A8-A15 sind low) des 64K-Adreßdreßraums. Um diese internen Ports zu lesen und zu schreiben, hat der HD64180 spezielle Befehle, die automatisch A8-A15 low ausgeben, wenn auf die internen Ports zugegriffen wird.

Traditionsbedingt decodieren aber sämtliche I/O-mapped Periherie-Geräte für Z80-Systeme nur A0-A7 und beschränken den I/O-Adreßraum damit auf 256 Ports. Diese Baugruppen fühlen sich bei Zugriffen auf die internen Ports ebenfalls angesrochen, wenn ihre 8Bit-Adresse mit der zero-Page-Adresse des angesprochenen internen Ports übereinstimmt. Beim Lesen eines internen Ports werden deshalb die Datentreiber des HD64180-Chips zur Umwelt geschlossen, so daß keine Kollisionen von internen und externen Daten möglich sind. Beim Schreiben gelangen aber die Daten, die für die internen Ports bestimmt sind, auch nach außen und werden von einem auf gleicher Adresse liegenden externen Port auch vereinnahmt. Verrückt spielende Peripherie-Geräte wären die Folge, wenn nicht noch eine zusätzliche Sicherung im HD64180 eingebaut wäre.

Der zusammenhängende Adreßbereich von 3FH Zero-Page-Ports für die Register des HD64180 beginnt default bei 00H. Durch Ändern der Bits 7 und 6 des IRC-Registers (3FH) lassen sich die 3FH Ports innerhalb der Zero-Page verschieben.

D7	D6	Adreßbereich der internen Ports
0	0	00 - 3F (nach RESET)
0	1	40 - 7F
1	0	80 - BF
1	1	C0 - FF

Die im folgenden benutzte Angabe von Portadressen geht davon aus, daß D6 und D7 von ICR beide low sind und die internen Ports bei 00-3F in der Zero-Page liegen. Die relativen Adressen und Funktionen sind in der Tabelle aus dem Prof-180-Handbuch zu ersehen. Mit dieser Tabelle alleine ist man aber noch lange nicht in der Lage, die internen Ports sinnvoll zu programmieren. Vielleicht helfen die folgenden Ausführungen da weiter:

Zunächst seien die Angaben zum IRC-Register, 'I/O Control Register', Port 3H) vervollständigt. D5 (IOSTP) = 0 ermöglicht den normalen Betrieb der H64180-internen Peripherie-Ports (Einstellung nachRESET). Wenn ISTP auf 1 gesetzt wird, sind die Funktionen von ASC1, CSi/0 und PRT (serielje Schnittstellen und Reload-Timer unterbunden.

Damit ein bei 9,26 MHz Taktrate tickender HD64180 auch mit langsamer Perpherie laufen kann, gibt es nicht nur den WAiT-Pin, über den die Peripherie die CPU um Geduld bitten kann; die CPU (und auch die internen DMA-Kanäle) können auch selbständig WAIT-Zyklen einlegen, wobei zwischen memory-mapped I/O und I/O-mapped Zugriffen unterschieden wird. Mit dem oberen Nibble von Port 32H (DCNTL) lassen sich unterschiedliche Anzahlen WAITs einstellen;

D7	D6	Anzahl WAITs bei memory mapped Adressierung
0	0	0
0	1	1
1	0	2
1	1	3 (nach RESET)

		Anzahl WAITs bei	
D5	D4	I/O Zugriffen	/INT0-Acknowledge

0	0	1	2
0	1	2	4
1	0	3	5
1	i	4	6

Be Zugriff auf die internen Ports werden unabhängig von D5 und D4 keine WAITs eingefügt. Nur für die Datenregister von ASCI, CSI/O und PRT werden je nach Zustand bis zu 4 WAITs für die Synchronisatin mit der CPU eingefügt.

Die Bits 0-3 von DCNTL dienen der Steuerung des DMA-Betriebs und werden dort behandelt.

RCR, Refresh Control Register, Port 36H

Gegenüber dem Hidden Refresh des Z80 während jeden M1-Zyklus' gibt der HD64180 jeweils nach einer bestimmten Anzahl von Taktzyklen eine 8Bit--Refreshadresse aus. Durch die 8Bit-Adresse können 256K-Bit- und 1M-Bit-Speicher-Chips ohne zusätzliche Refresh-Logik verwendet werden.

Ein Refresh erfolgt asynchron zur CPU-Arbeit nach einer über RCR D0 und D1 eingestellten Anzahl von Taktzyklen:

D1	D0	Anzahl Takte zwischen zwei Refresh-Zyklen
0	0	10
0	1	20
1	0	40
1	1	80

Mit D7 = 0 an RCR wird der Refresh-Controller abgeschaltet (für statische RAMs kann das Refreshing entfallen. D7= 1 (nach RESET) aktiviert den Refresh-Controller.

#####################

Mit D0 läßt sich das Refresh-Timing den Möglichkeiten des Speichers anpassen:

D0 = 0 : Refresh ist 2 Taktzyklen lang

D0 = 1 : Refresh ist 3 Taktzyklen lang (nach RESET voreingestellt)

D2-D5 im RCR-Register haben keine Funktion.

ICT, Interrupt/Trap-Register, Port 34H

D7 (Trap) wird immer dann von der CPU auf 1 gesetzt, wenn diese auf einen illegalen Befehl gestoßen ist (z.B. undefinierte Z80-Befehle). Durch einen OUT0-Befehl kann D7 wieder auf 0 gesetzt werden (nicht aber auf 1). Nach RESET ist D7=0.

Ein illegaler Op-Code löst den für den HD64180 höchst priorisierten Trap-Interrupt aus. Die Service Routine kann dann aus dem Zustand von D6 (UFO) erkennen, ob der Fehler im 2ten oder 3ten Byte des Op-Codes auftrat.

D6 erlaubt die Korrektur des beim Trap auf den Stack geretteten Programm-Counters. Bei D6=0 zeigt der gerettete PC auf das zweite Byte des als falsch erkannten Befehls. Bei D6=1 zeigt der PC auf das 3. Byte des falschen Befehls. D6 kann nur gelesen werden.

D0, D1 ubd D2 können zur Maskierung des externen Interrups /INT0, /INT1 und /INT2 verwendet werden. Wenn das entsprechende Bit auf 0 gestzt ist, ist der Interrupt gesperrt: auf 1 gesetzt wird der entsprechende Interrupt freigegeben. Nach RESET ist nur /INT0 freigegeben (D0=1) und /INT1 und /INT2 sind gesperrt (D1, D2 = 0).

IL, Interrupt Vector Low Register, Port 33H

Die externen Interrupts /INT0 im Mode 2 sowie /INT1 und /INT2 grundsätzlich und alle internen Interrupts werden von Service-Routinen bedient, deren Startadressen in e3iner Tabelle abgelegt sind. Die Technik ist die des IN2 beim Z80. Im I-Register der CPU werden die oberen 8 Bit (A8-A15) der Adresse der Vector-Tabelle abgelegt. Die unteren 8 Bit der Adresse (Pointer in die Vector-Tabelle) liefert das Interrupt-auslösende Peripherie-Gerät beim Interrupt-Ackowledge über den Datenbus. Die CPU ignoriert davon D0, um eine gerade Adresse zu erzeugen, und bildet aus den beiden Hälften einen Pointer in die Vectortabelle. Die dort in der Tabelle abgelegte Adresse ist die Startadresse der Service-Routine.

Beim HD64180 benötigt nur der /INT0, der dem /INT des Z80 im Mode 2 entspricht, einen extern gelieferten Adreßanteil (A0-A7 uber D0-D7) des Datenbus, um in der Vectortabelle die Adresse der Service-Routine zu finden. Damit können abwechselnd mehrere INT-Quellen über nur einen Interrupt-Eingang der CPU angeschlossen werden.

Die Adressen der übrigen Interrupt-Quellen des HD64180 sind (mit gewissen Freiheiten) festgelegt.

So stellen die ersten beiden Bytes der Tabelle (Adresse mit A0-A7=0) die Adresse mit der Service-Routine füt /INT1 dar. Die nächsten beiden Bytes liefern den Vektor für /INT2. Die Reihenfolge der Vektoren ist durch die Interrupt-Quellen festgelegt; es läßt sich aber die relative Lage der Gruppe von Vektoren (32 Bytes) innerhalb der 256-Byte-Tabelle in 32-Bytes-Schritten verschieben. Wo diese Adressengruppe relativ in der Tabelle liegt, wird durch D7-D5 des IL-Registers vorgegeben (nach RESET sind diese low und die Gruppe liegt am Anfang der Vektor--Tabelle).

wenn D7-D5 alle low sind, liegen die Vektoren für /INT1, /INT2 und die internen Interrupts ab Adresse XX00H in der Tabelle; wenn D7-D5 das Bitmuster 001 haben, liegen diese Vektoren ab XX20H in der Tabelle; u.s.w

Die Bits 0-4 werden von der internen Interrupt-Quelle selbst erzeugt und geben die relative Adresse innerhalb der 32-Byte-Gruppe in der Tabelle vor. Die Tabelle 2.7.2 des 64180-Handbuchs (die hier ohne freundliche Genehmigung von Hitachi wiedergegeben ist) gibt die Reihenfolge der Einträge für die Interruptquellen und die damit verknüfte Prioritäts-Hirarchie vor.

Table 2.7.2 Interrupt Source and Lower Vector

Interrupt--Source	Priority	IL b7	IL b6	IL b5	Fixed Code b4	b3	b2	b1	b0
/INT1	highest	.	.	.	0	0	0	0	0
/INT2		.	.	.	0	0	0	1	0
PRT channel 0		.	.	.	0	0	1	0	0
PRT channel 1		.	.	.	0	0	1	1	0
DMA channel 0		.	.	.	0	1	0	0	0
DMA channel 1		.	.	.	0	1	0	1	0
CSI/0		.	.	.	0	1	1	0	0
ASCI channel 0		.	.	.	0	1	1	1	0
ASCI channel 1	lowest	.	.	.	1	0	0	0	0

Außer den maskierten Interrupts gibt es noch den oben erwähnten Trap, der eine Verzweigung zur logischen Adresse 0000H zur Folge hat. Wenn diese logische Adresse mit der pysikalischen Adresse 0000H identisch ist (Banking), entspricht dies dem /RESET-Vektor. Durch Prüfen von D7 in Port 32H (ICT) kann erkannt werden, ob ein Trap oder ein RESET vorliegt.

Ein /NMI bewirkt, wie beim Z80 einen CALL 00066H, wobei auch hier die logische Adresse innerhalb der eingestellten Bank und nicht unbedingt die absolute Adresse 00066H relevant ist.

Der /INT0 kennt wie der /INT des Z80 noch die Modi 0 und 1

Mode 0 : Beim Acknowledge wird ein 1Byte-Befehl vom Datenbus gelesen Und ausgeführt (RST XX) Dieses Byte muss die Interrupt-Quelle liefern.

Mode 1 : Bei einem Interrupt am /INT0-Pin erfolgt ein RST 38H. Die Adresse 0038H liegt in der per MMU eingestellte Bank (logische Adresse).

Weitere Steuerungen der internen Interrupts werden bei der Beschreibung der jeweiligen Funktionsgruppen gezeigt. Dazu aber erst im nächsten Info.

Table 2.5.1 Internal I/O Register Address Map (1)

	Register	Mnemonic	Address
ASCI	ASCI Control Regitser A Ch 0	CNTLA0	00H
	AStI Control Register A Ch 1	CNTLA1	01H
	ASCI Control Register B Ch 0	CNTLBO	02H
	ASCI Control Register B Ch 1	CNTLB1	03H
	ASCI Status Register Ch 0	STATO	04H
	ASCI Status Register Ch 1	STAT1	05H
	ASCI Transmit Data Register Ch 0	TOR0	06H
	ASCl Transmit Data Register Ch 1	TDR1	07H
	ASCI Receive Data Register Ch 0	RDR0	08H
	ASCI Receive DBta Register Ch 1	RDR1	09H
CSI/O	CSI/O Control Register	CNTR	0AH
	CSI/O Transmit/Receive Data Reister	TRDR	0BH
Timer	Timer Data Register Ch OL	TMDROL	0CH
	Timer Data Register Ch OH	TMDROH	0DH
	Reload Register Ch OL	RLDROL	0EH
	Reload Register Ch OH	RLDRoD	0FH
	Timer Control Register	TCR	10H

	(Reserved)		11H
			12H-13H
	Timer Data Register Ch 1L	TMDROL	14H
	Timer Data Register Ch 1H	TMDROH	15H
	Reload Register Ch 1L	RLDROL	16H
	Reload Register Ch 1H	RLDROH	17H
Outhers	Free Running Counter	FRC	18H
	(Reserved)		19H-1FH
DMA	DMA Source Address Register Ch OL	SAROL	20H
	DMA Source Address Register Ch OH	SAROH	21H
	DMA Source Address Register th OB	SAROB	22H
	DMA Destination Address Register Ch OL	DAROL	13H
	DMA Destination Address Register Ch OH	DAROH	24H
	DMA Destination Address Register Ch OB	DAROB	25H
	DMA Byte Count Register Ch OL	BCROL	26H
	DNA Byte Count Register Ch OH	BCROH	27H
	DMA Memory Address Register Ch 1L	MARIL	28H
	DMA Memory Address Register Ch 1H	MARLH	29H
	DMA Memory Address Register Ch 1B	MARIB	2AH
	DMA I/O Address Register Ch 1L	IAR1L	2BH
	DMA I/O Address Register Ch 1H	IAR1H	2CH
	(Reserved)		2DH
	DMA Byte Count Register Ch 1L	BCR1L	2EH
	DMA Byte Count Register Ch 1H	BCR1H	2FH
	DMA Status Register	DSTAT	30H
	DMA Mode Register	DMODE	31H
	DM/WAITE Control Register	DCNTL	32H
INT	IL Register Interrupt Vector Low Reg.		33H
	INT/TRAP Register		34H
	(Reserved)		35H
Refresh	Refresh Control Register	RCR	36H
	(Reserved)		37H
MMU	MMU Common Base Register	CBR	38H
	MMU Bank Base Register	BBR	39H
	MMU Common/Bank Area Register	CBAR	3AH
I/O	(Reserved)		38H-3EH
	I/O Control Register	ICR	3FH

Die MMU (Memory Management Unit) des HD64180

Helmut Bernhardt

Der auch vom HD4180 benutzte Befehlssatz des Z80 kennt nur 64K-umfassende Adressoperanden. Die Software kann nur 64K-Byte Speicher adressieren. Durch die MMU ist es möglich, Anteile dieses logichen 64K-Adreßraumes in verschiedene Bereiche des physikalischen 512K-Adreßraumes zu legen.

Der durch Software ansprechbare logische 64K-Adreßraum kann in bis zu 3 Bereiche unterteilt werden: Common0, Bank und Common1. Common0 beginnt immer bei 0000H und kann in 4K-Schritten beliebig groß angelegt werden. Common1 liegt immer unter der oberen Grenze FFFFH des logischen Adreßraumes und kann ebenfalls in 4K-Schritten beliebig weit nach unten reichend vorgegeben werden. Bank umfaßt den Bereich zwscchen Common0 und Common1. Für die Aufteilung des logischen Adreßraumes sind folgende Konfigurationen möglich:

```
+---------+     +---------+     +---------+     +---------+
| Common1 |     | Common1 |     | Common1 |     | Common1 |
+---------+     |         |     |         |     |         |
| Bank    |     +---------+     +---------+     |         |
+---------+     |         |     |         |     |         |
| Conmon0 |     | Bank    |     | Common0 |     |         |
+---------+     +---------+     +---------+     +---------+
```

Während Common 1 mit einer Größe von minimal 4K unterhalb von FFFFH immer vorhanden ist, lassen sich die Bereiche Commn0 und Bank auch ganz unterdrücken. Die Adressen der Grenzen zwischen den einzelnen Bereichen lassen sich in 4K-Schritten verschieben. Über je 4 Bits des CBAR-Registers Port 3AH werden die Adressen A15-A12 der Untergrenzen

der Bereiche Common1 und Bank festgelegt.
CBAR, (Common/Bank Base Register), Port 3AH

Bit	7	6	5	4	3	2	1	0
Name	CA3	CA2	CA1	CA0	BA3	BA2	BA1	BA0
Zugriff	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Dabei represetieren D7-D4 die Adressen A15-A12 die Untergrenze des Common1-Bereichs und D3-D0 die Adressen A15-A12 der Untergrenze des Bank-Bereichs. Nach RESET wird dieses Register immer mit F0 initialisiert. Damit ist die für ein gebanktes CP/M 3.0 günstigste Einteilung in Common1 bei F000H-FFFFH und Bank bei 0000H-EFFFH vorgegeben.
Wenn z.B. C4H an CBAR ausgegeben wird, liegt die Untergrenze von Common 1 bei C000H und die Untergrene von Bank bei 4000. Die resultierende Speicheraufteilung sieht dann so aus:

```
    Common1  C000H-FFFFH
    Bank     4000H-BFFFH
    Common0  0000H-3FFFH
```

Die Nibble des Bytes an CBAR sind die oberen Adreß-Digits der Untergrenzen von Common1 (oberes Nibble) und Bank (unteres Nible).
Eine Ausagbe von F0H an CBAR bedeutet, daß Common1 bei F000H-FFFFH liegt und der restliche Speicher 0000G-EFFFH Bank-Bereich ist. Common0 existiert dann nicht (durch RESET voreingetellt).
Eine dem GDOS 2.4 des GENIE IIIs ähnliche Configuration (Common bei 0000h-3FFFH und Bank bei 4000H-FFFFH läßt sich mit 44H an CBAR einstellen. Hier fallen die Untergrenzen von Common1 und Bank beide auf die Adresse 4000H. Der Bank-Bereich existiert dann nicht, Common0 wird als Common benutzt und Banking erfolgt mit Verschieben des Common1-Bereichs im physikalischen Speicher.
Um eine dem 256K-(1MB-)Banker für Z80-Systeme ähnliche Konfiguration vorzueben müßte (8H in CBAR eingetragen werden. Dann liegt die Grenze zwischen den beiden existierenden Bereichen CommonO und Common1 be8000H. Das Banking erfolgt auch hier durch Verschieben von Common1 im physikalischen Adreßraum.
Adreßübersetzung
Common0, soweit es durch die Untergrene von Bank zugelassen ist, beginnt hier an der physikalischen Adresse 00000H des 512K-Speichers.
Die Bereiche Bank und Common1 lassen sich in verschiedene Stellen in den physikalischen Speicher legen. Die physikalische Adressen, bei der diese Bereiche anfangen, lassen sich auch wieder in 4K-Schritten an beliebiger Stelle festlegen und ändern. Dazu dienen die Ports BBR und CBR. Die Bits 6-0 in diesen Registern geben den physikalischen Adreß--Offset vor, der bei Zugriff auf logische Adressen im Common1- bzw. im Bank-Bereich zu den logischen Adressen addiert werden. D6-DO dieser Ports legen die Adressen A18-A12 dieses Offsets fest.

CBR Common Base Register), Port 38H

Bit	7	6	5	4	3	2	1	O
Name	-	CB6	CB5	CB4	CB3	C82	CB1	CB0
Zugriff		R/W	R/W	R/W	R/W	R/W	R/W	R/W

BBR (Bank Base Register), Port 39H

Bit	7	6	5	4	3	2	1	O
Name	-	BB6	BB5	BB4	BB3	BB2	BB1	BB0
Zugiff		R/W	R/W	R/W	R/W	R/W	R/W	R/W

Nach einem RESET sind alle Bits in CBR und BBR low. In dieser Konfguration erfolgt bei Zugriff auf den Common1-Bereich und den Bank-Bereich keine Adreßtranslation. Die logische Adresse ist gleichzeitig auch die physikasische Adresse. A18-A16 werden stänig low ausgegeben.
Bei Zugriff auf eine logische Adresse im Bank-Bereich wird zu dieser Adresse die in BBR für A18-A12 festgelegte Offset-Adresse addiert und der Speicherzugriff erfolgt an der physikalischen Adresse, die die Summe aus logischer Adresse und Offset ist. Das läßt sich an einem Beispiel einfacher erklären.
Die MMU-Register seien folgendermaßen belegt:
CBAR: C4H   BBR: 40H   CBR: 00H
Damit (siehe obiges Beispiel für CBAR) liegt der Bank-Bereich logisch bei 4000H-BFFFH. Der Adreß-Offset für logische Zugriffe auf den Bank-Bereich beträgt 40000H. Wenn nun an der Adresse 9C84H, die im logischen Bank-Bereich liegt, ein Byte gelesen oder abgelegt werden soll, oder an diese Adresse gesprungen werden soll (im Programm), dann wird tatsächlich die Adresse 49C84H ausgegeben.

```
logische Adresse    9C84H          1001 1100 1000 0100 B
Adreß-Offset       40000H      100 0000 0000 0000 0000 B
                +  ------    + -------------------------
```

```
physik. Adresse    49C84H      100 1001 1100 1000 0100 B
```

mit 00H in CBR erfolgt bei Zuriff auf logische Adressen im Common1-Bereich keine Addreßübersetzung (der Offset ist 00000H).Banking erfolgt durch Ausgabe verschiedener Addreß-Offsets an das BBR-Register. Durch die Auflösung in 4K-Schritte für die Vorgabe des Offets lassen sich auch Banks einstellen, die sich mit anderen (vorher oder- nachher eingestellten) Banks oder mit Common 1 überschneiden.

```
Speicher-Konfiguration bei       +---------+ 7FFFFH  Speichergrenze
    CBAR : C4H                   |         |
    BBR  : 40H                   |         |
    CBR  : 00H                   +---------+ 4COOOH
                                 |  Bank   |
                                 |         |
                                 +---------+ 4C400H
                                 |         |
                                 +---------+ 40000H Offset durch BBR
                                 |         |
                                 |         |
FFFFH +---------+                +---------+ OFFFFH
      | Common1 |                | Common1 |        kein Offset in CBR
C000H +---------+                +---------+ 0C000H
      |  Bank   |                |         |
      |         |                |         |
0000H +---------+                +---------+ 04000H
      |         |                | Common0 |
      +---------+----------------+---------+ 00000H Common0 beginnt
                                                    immer bei 00000h
logischer Adreßraum        physkalischer Adreßraum
```

Um bei dieser (zugegeben etwas unsinnigen) Speicherkonfiguration den 512K-Adreßraum durch Banking des Bereichs 4000H-BFFFH sinnvoll auszunutzen, kann fo1gende Speicheraufteilung in Banks angewandt werden:

Bank Nr.	phys. Anfangs- u. End-adresse		Adreß-Offset	Byte an BBR
0	04000H	0BFFFH	00000H	00H
1	10000H	17FFFH	0C000H	0CH
2	18000H	1FFFFH	14000H	14H
3	20000H	27FFFH	1C000H	1CH
	....	....	....	...
13	70000H	77FFFH	6C000H	6CH
14	78000H	7FFFFH	74000H	74H

Unter Aussparung des Cmonon0- und des Common1-Bereichs bei 00000H und bei 0C000H mit je 16K Größe füllen die Banks lückenlos den physikalischen Speicher aus.

Es sind aber auch Einstellungen möglich, bei denen sich die Bereiche von Banks oder einer Bank und einem Common-Bereich überschneiden. Wenn für das Einschalten von Bank1 in obiger Tabelle anstelle von 0CH der Wert 08H an BBR ausgegeben wird, liegt der physikalische Adreßbereich der Bank bei 0C000H-13FFFH. Ein Zugriff auf den logischen Bereich 4000H-7FFFH bedeutet dann einen Zugriff auf den physikalischen Bereich 0C000H-0FFFFH, dies ist der Bereich von Common1. Die unterschiedlichen logischen Adreßbereiche 3000H-7FFFH und C000H-FFFFH sin physikalisch identisch. Die Programmierung der MMU des H64180 sollte also nicht durch Ausprobieren sondern durch gründliches Überlegen der Wenns und Abers erfolgen.

Den DMA-Controller interessieren die Einstellungen der MMU-Register nicht, er arbeitet mit 19Bit-Speicheradressen, die direkt als physikalische Adressen ausgegeben werden.

Der neuere PLCC-Typ des HD4180 hat bei 68 Pins auch noch den Adreßpin A19 herausgeführ. Dadurch erhalten dort die Bits D7 in BBR und CBR (und auch die Bits D3 in den DMAC-Adreßregister-Zusatzbytes) die Funktion, einen 1MB-Seicherraum adressieren zu können. Dort ist für eine 1MB-Speicheradressierung keine zusätzliche Logik mehr nötig.

Der DMA-Controller des HD64180

Helmut Bernhardt

Der On Chip DMAC (Direct Memory Access Controller) des HD64180 bietet gegenber einem eigenständigen DMAC (z.B. Z80-DMAC) gewisse Vorteile

1) beim RESET liegen definierte Einstellungen vor; ein Hängen des Systems kommt nicht vor

2) entsprechend den Adressierungmöglichkeiten der CPU kann auch der DMAC 512K adressieren

3) da CPU und DMAC auf einem Chip und dadurch beide auf dem CPU-Bord sitzen, kann eine aufwendige DMA-fähige Steuerung der Bustreiber entfallen, ohne auf DNA verzichten zu müssen

Der DMAC des HD64180 verfügt über 2 Kanäle mit unterschiedlichen Eigenschaften:

Kanal 0 kann
   Speicher <-> Spejcher -
   Speicher <-> I/O - und
   Speicher <-> memory mapped I/O - Transfers durchführen
   Dabei kann bei jedem Speicherzugriff die Adresse inkrementiert, dekrementiert oder unverändert belassen werden. Dje Speicher <-> Speicher - Übertragung kann im Burst- oder im Cycle Steal Mode erfolgen. Die beiden Chip-internen seriellen Schnittstellen (ASCI) können über den DMA-Kanal 0 mit Handshake bedient werden

Kanal 1 kann nur Memory <-> I/O - Transfers durchführen. Dabei Xenn die Speicheadresse nach jedem Zugriff inkrementiert oder dekrementiert werden.

Die Ports des DMA-Controllers

Kanal 0

20H	SAROL	Source Address Register Channel 0 low	A0 - A7
21H	SAROH	Source Address Register Channel 0 high	A8 - A15
22H	SAROB	Source Address Register Channel 0 bank	A16 - A18
23H	DAROL	Destination Address Register Channel 0 low	A0 - A7
24H	DAROH	Destination Address Register Channel 0 high	A8 - A15
25H	DAROB	Destination Address Register Cnannel 0 bank	A16 - A18
26H	BCROL	Byte Count Register Channel 0 low	D0 - D7
27H	BCROH	Byte Count Register Channel 0 high	D8 - D15

Kanal 1

28H	MAR1L	Memory Address Register Channel 1 low	A0 - k7
29H	MAR1H	Memory Address Register Channel 1 high	A8 - A15
2AH	MAR1B	Memory Address Register Channel 1 bank	A16 - A18
2BH	IAR1L	I/O Address Register Channel 1 low	A0 - A7
2CH	IAR1H	I/O Address Register Channel 1 high	A8 - A15
2DH		reserved	
2EH	BCR1L	Byte Count Register Channel 1 low	D0 - D7
2FH	BCR1H	Byte Count Register Channel 1 high	D8 - D15

für beide Kanäle

30H	DSTAT	Status Register
31H	DMODE	Mode Register
32H	DCNTL	DMA/WAIT Control Register

Für Kanal 0 ist vorgegeben, wo die Adressen für 'Quelle' und 'Ziel' einzutrage sind. Dabei ist es in beiden Fällen möglich, Speicheradessen oder Ports anzugeben. Bei Ports ist dann der Inhalt des Zusatzadreßregisters (A16-A18) ohne Bedeutung. Memory mapped I/O unterscheidet sich von normalen Speicheradressen nur dadurch, daß dort auch ein Hardware-Handshaking möglich ist.

Kanal 1 legt fest, daß unabhängig von der Funktion 'Qelle' oder 'Ziel' die Speicheradresse in die Ports 28H-2AH und die I/O-Adresse in die Ports 2BH, 2CH eingetragen wird. Die Übertragungsrichtung sowie der Modus (Inkrementieren, Dekrementieren oder feste Adresse für Ziel und Quelle) werden in Port 32H (DCNTL), Bits O-3 festgelegt.

Für beide Kanäle kann durch die Vorgaben in den jeweiligen Byte Count Registern (16 Bit) eine Übertragung von bis zu 64K am Stück vorgegeben werden.

Für allgemeine Steuervorgaben (write) und Statusinforationen (read) sind die Ports 30H-32H zuständig:

DMA Status Register, DSTAT, Port 30H

Bit	7	6	5	4	3	2	1	O
Name	DE1	DE0	/DW1	/DWE0	DIE1	DIE0	-	DME
Zugriff	R/W	R/W	W	W	R/W	R/W		R

DE1 : DMA Enable Channel 1
      Ein DMA-Transfer über Kanal 1 wird durch Ausgabe von DE1=1 gestartet. Um DE1 zu setzen, muß gleichzeitig /DWE1 ausgegeben

werden; nur dann wird der Wert für DE1 übernommen. Wenn DE1=1 gesetzt wird, wird automatisch gleichzeitig auch DME gesetzt.

DE0 : DMA Enable Channel 0

Um einen DMA-Transfer über Kanal 0 zu starten, müssen gleichzeitig DE0=1 und /DEW0=0 an DSTAT ausgegeben werden. DME wird dabei automatisch gesetzt.

Der DMA-Transfer läßt sich (im Cycle Steal Mode) bei beiden Kanälen unterbrechen, wenn DEx=0 und /DWEx=0 an DSTAT ausgegeben werden. Mit der Ausgabe von DEx=1 und /DWEx=0 kann der Transfer dann weitergeführt werden. Nach beendetem Transfer wird DEx automatisch =1 gesetzt.

/DWE1, /DWE0: DMA Bit Write Enable Channel 1, -Channel 0

Um DE1 oder DE0 zu setzen oder zu löschen, muß gleichzeitig das entsprechende Bit /DWE1 bzw. /BWE0 low ausgegeben werden.

DIE1, DIE0: DMA Interrupt Enable Channel 1, -Channel 0

Wenn DIEx für Kanal x als 1 gesetzt wird, erfolt nach Beenden des DMA-Tansfers ein CPU-Interrupt. DIEx = 0 (nach RESET voreingestellt) verhindert den Interrupt.

DMA Mode Register, DMODE, Port 31H

Bit	7	6	5	4	3	2	1	0
Name	-	-	DM1	DM0	SM1	SM0	MOD	-
Zugriff			R/W	R/W	R/W	R/W	R/W	

Über DODE wrrd derÜbertraungs-Modus für ausschließlich Kanal 0 vorgegeben. DM1 und DM0 geben für das Ziel und SM1 und SM0 für die Quelle die Übertragungsbedingungen vor:

DM1	DM0	Ziel und Bedingungen
0	0	Memory mit Adreßinkrement
0	1	Memory mit Adreßdekrement
1	0	Memory mit feststehender Adresse
1	1	I/O mit feststehender Adresse

SM1	SM0	Quelle und Bedingungen
0	0	Memory mit Adreßinkrement
0	1	Memory mit Adreßdekrement
1	0	Memory mit feststehender Adresse
1	1	I/O mit feststehender Adresse

DM1, DM0, SM1 und SM0 werden bei RESET auf 0 gesetzt.

KMODE hat keine Funktion, wenn Quelle oder Ziel als I/O festgelegt werden. Dann bestimmt der Triggerimpuls am Pin /DREQ0 das Timing. Bei Memory <-> Memory-Transfers bestimmt KMODE die Transferart.

MMOD=0 : Cycle Steal Mode; Nach jedem DMA-Transfer darf die CPU einen Zyclus ausführen.

MMOD=1 : Burst Mode; die CPU erhält die Kontrolle über den Bus erst dann wieder, wenn die gesamte DMA-Übertragung beendet ist.

DMA/WAIT Control Register, DCNTL, Port 32H

Bit	7	6	5	4	3	2	1	0
Name	MWI1	MWI0	IWI1	IWI0	DMS1	DMS0	DIM1	DIM0
Zugriff	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MWI1, MWI0, IWI1 und IWI0 steuern die Anzahl WAITs für Memory- und Port-Zugriffe durch die CPU und den DMAC.Diese Bits wurden bereits im im letzten Teil besprochen.

DMS1, DMS0, DIM1 und DIM0 sind Steuerbits für den Kanal 1 der ausschließlich Memory <-> I/O- und I/O <-> Memory-Übertragungen steuert. Diese Übertragungen werden durch DERQ1 getriggert. Da auch beim Kanal 0 getriggerte Übertragungen möglich sind, kann mit DMS0 auch hier die Triggerart festgelegt werden.

DMS1, DMS0 : DMA Request Sense

legen die Art des Triggerns für DERQ1 (Kanal 1) und DERQ0 (Kanal 0) fest. Wenn das Bit auf 0 gesetzt ist,wird durch einen Low-Pegel am DERQ-Pin getriggert. Der DERQ-Pin wird zu Beginn des zweiten Taktes eines CPU-Maschinen-Zyklus übernommen. Wenn zu diesem Zeitpunkt am DERQ-Pin Low-Pegel anliegt, wird der Maschinen-Zyklus noch beendet und dann ein DMA-Transfer durchgeführt.

Wenn DMSx auf 1 gesetzt ist, wird die DMA-Übertragungdurch eine negative Flanke am DERQx-Pin getriggert.

DIM1, DIM0 : Channel 1 I/O and Memorx Mode

Festlegen von Quelle und Ziel und Inkrementieren/Dekrementieren von Speicheradressen.

DIM1	DIM0	Transfer-Richtung	Inc./Dec.
0	0	Memory -> I/O	Inkrementieren

0	1	Memory -> I/O	Dekrementieren
1	0	I/O -> Memory	Inkrementieren
1	1	I/O -> Memory	Dekrementieren

Die I/O-Adresse wird in allen Fällen nicht verändert.

Programmieren der DMA-Kanäle

Kanal 0, Memory <-> Memory

Die reine Memory<->Memory- Übertragung wird nicht durch den DERQ0-Pin getriggert. Das Timing erfolgt je nach Modus (Burst- oder Cicle Steal) inzwei verschiedenen Arten. In beiden Fällen hält die DMA-Übertragung an, bis der Byte Count Zähler auf 0 heruntergezählt. Im Burst Mode werden alle Bytes nacheinander übertragen und die CPU erhält erst dann wieder die Kontrolle über den Bus. Im Cycle Steal Mode wird immer abwechselnd ein Maschinen-Zyklus der CPU und eine Übertragung des DMAC durchgeführt.

Es werden nacheinander folgende Register programmiert:

1) Laden der Memory Adress Source- und Destination-Register
2) in DMODE: Memory<->Memory und Adreß-Inkrement/Dekrement über SM1, SM0, DM1, DM0 festlegen
3) Anzahl zu übertragender Bytes in die Byte Count Regjster schreiben
4) Burst- oder Cycle Steal Mode in MMOD von DCNTL vorgeben
5) Übertragung mit DE0=1 und geichzeitig /DWE0=0 in DSTAT starten

Kanal 0, Memory <-> I/O oder memory mapped I/O

Das Timing des DMA-Transfers wird durch die Pins /DREQ0 und /TEND0 des HD64180 gesteuert.

/DREQ0 : über diesen Eingang wird ein DMA-Transfer getriggert.
TEMD0 : über diesen Ausgang wrrd das Ende eines DMA-Blocktransfers (BRC0=OOH) angezeigt.

Diese Pins sind mit den Clock Inputs/Outputs CKA0 und CKA1 der ASCI- -Schnittstelle gemultiplext. Wenn der DMA-Kanal 0 für Memory <-> I/O oder memory mapped I/O programmiert wird, fungieren diese Pins als DMA Handshake-signale. CKA1D in CNTLA1 wird automatisch high gesetzt.

Wenn vor der ansteigenden Flanke von T3 eines CPU-Maschinenzyklus eine fallende Flanke von /DREQ0 auftrat (wenn taktflanken-getriggert programmiert) bzw. bei der steigenden Flanke von T3 Low-Pegel an DREQ0 vorliegt (wenn pegel-getriggert programmiert), wird anstelle des nächsten CPU-Maschinenzyklus ein DMA-Transfer ennes Bytes (Lesen und Schreiben) durchgeführt.

Initialisierung:

1) Memry- und (memory-mapped) I/O- Quell- und Ziel-Adressen in SAR0 und DAR0 eintragen. I/O-Adressen sind 16 Bit breit, A16 und A17 sind als 0 einzutragen.
2) durch Setzen von SM0, SM1, DM0 und DM1 in DMODE vorgeben, ob Memory <-> I/O oder Memory <-> memory mapped I/O und ob Adressen inrementiert oder dekrementiert werden sollen.
3) Anzahl zu übertragender Bytes in CCR0 einragen
4) durch DMS0 in DCNTL flanken- oder pegelgetriggert an /DREQ0 vorgeben
5) durch DIE0 in DSTAT DMA-Ende-Interrupt freigeben oder sperren
6) DE0=1 und gleichzeitig /DWE0=0 in DSTAT eintragen

/DREQ0 steuert dann die DMA-Übertragung.

Kanal 0, Memory <-> ASCI

Bei Übertragungen über die 2-Kanal-ASCI-Schnittstelle wird das Timing; nicht über /DREQ0 gesteuert. Die ASCI-Statusbits erzeugen intern ein /DREQ0. Es sind dies die Bits TDRE (Transmmt Data Register Empty) und RDRF (Receive Data Register Full).

Initiaisierung:

1) Quell- und Ziel-Adressen in SAR0 und DAR0 vorgeben. Die I/O-Adresse- ist ein ASCI-Register (06H-09H), das mit A0 - k7 vorgegeben wird. A8 - A15 sind low zu sezten. A18 ist ohne Bedeutung. 16 und A17 steuern das Timing über die ASCI-Statusbits:

   bei Empfangen über ASCI: '

A17	A16	D-Transfer-Request durch
0	0	/DREQ0 bei Übertragung nicht über ASCI
0	1	RDRF (ASCI, Kanal 0)
1	0	RDRF (ASCI, Kanal 1)
1	1	reserviert

   bei Senden über ASCI

A17	A16	DMA-Transfer-Request durch
0	0	/DREQ0 bei Übertragung nicht über ASCI

```
    0   1    TDRE (ASCI Kanal 0)
    1   0    TDRE (ASCI Kanal 1)
    1   1       reserviert
```

2) durch SM0, SM1, DM0 und DM1 in DMODE Memory <-> I/O - Transfer und Inkrementieren oder Dekrementielen der Memory-Adressen vorgeben
3) Anahl zu übertragender Bytes in BCR0 vorgeben
4) DMS0 in DCNTL muß für "flankengetriggert" programmiert werden
5) DIE0 in DSTAT kann DMA-Ende-Interrupt zulassen oder sperren
6) Mit DE0=1 und gleichzeitig /DWE0=0 an DSTAT wird der DMA-Transfer Jnitialisiert und dann durch die ASCI-Statusbits gesteuert

Die ASCI-Empfänger/ -Sender müssen initialisiert werden, um den ersten DMA-Transfer zu starten. Der ASCI-Empfänger muß leer sein (RDRF=0) bzw. der ASCI-Sender muß mit einem ersten Byte geladen sein (TDRE=0).
Das erste Byteß muß von der CPU an den ASCI-Sender gegeben werden; alle weiteren Bytes werden per DMA übertragen.

Kanal 1, Memory <-> I/O

bis auf einige unterschiedliche Register und Status-/Control-Bits funktioniert ier DMA-Transfer über Kanal 1 wie der Memory <-> I/O Transfer über Kanal 0

Initialisierung:
1) Memory-Adresse (19 Bit) in MAR1 ff eintragen
2) I/O-Adresse (16 Bit) in IAR1 ff eintragen
3) mit DIM0 und DIM1 von DCNTL Quelle und Ziel sowie Dekrementieren oder Inkrementieren der Memory-Adresse festlegen
4) in DMS1 von DCNTL vorgeben, ob /DREQ1 flanken- oder pegel-getrig-gert arbeiten soll
5) über DIE1=1 in STAT einen DMA-Ende-Interrupt zulassen oder sperren
6) DE1=1 und gleichzetig /DWE1=0 in DSTAT eintragen

Das Timing wird dann über /DREQ1 gesteuert; /TEMD1 gibt das Ende eines DMA-Blocktransfers dem I/O-Gerät bekannt.
Diese kurze Übersicht der Programmierng berücksichtig die Reihen-folge der eizelnen Schritte, wie sie von Hitachi im HD64180-Datenbuch vorgechlagen ist. Solange man das Register DSTAT immer zuletzt be-schreib, kann in der Reihenfolge des Befüllens der anderen Ports eine sinnvollere Ordnung eingebracht werden. So lassen sich z.B. beim Memo-ry <-> Memory DMA-Transfer über Kanal 0 die hintereinander liegenden Ports für Qell- und Zieladressen und Übertrgungslänge mit dem Befehl OTIMR aus einer entsprechenden Tabelle im Speicher befüllen und nur noch DMODE und DSTAT sind mit einelnen OUT0-Befehlen zu versorgen.
Die Nutzung der DMA-Kanäle für Memory <-> I/O oder memory mapped I/O Übertragungen setzt voraus, daß die Hardware die Signale /DREQ0 bzw. /DREQ1 unu /TEND0 bzw. /TEND berücksichtigt. Der Prof180 wickelt z.B. den Datenaustausch mit dem µPD765-Floppy-Controller über den DMA-Kanal 1 ab und hat entsrrechend die Signale /DREQ1 und /TEND1 über Inverter zwischen CPU und FDC verbunden.

Die seriellen Schnittstellen (ASCI und CSI/O) des HD64180

Gerald Schröder

Der HD64180 stellt drei serielle Schnittstellen zur Verfügung. Bei zweien handelt es sich um "Asynchronous Serial Connection Interfaces" (wie RS-232/V24) und die dritte heißt "C1ocked Serial Input/Output".

Aschronous Serial Communication Interfces

Aufgabe dieser beiden Scbnittstellen ist die serielle Verbindung zu einem Modem, Termial, (RS-232-)Drucker o.Ä. Zusätzlich gibt es einen "Multipro-zessor-Modus", mit dem sich leicht ein System mit mehreren HD64180s auf-bauen ließe, wobei die ASCIs dann als (Verbindung zum) System-Bus dienen würden.
Jeder Kanal belegt 5 Ports: einen zum Lesen und einen zum Scbreiben der Daten, einen für den Status (Zustand) des Kanals, zwei für die Einstellung des Kanals (Baudrate usw.) und einige Zusatz-Informationen. Die Kanäle unterscheiden sich in einigen Features, was ich aber jeweils anmerken werde.
Die einzelnen Ports (Kanal 0 und Kanal 1):

Name	Port	Bezeichnung
CNTLA0	0	Control Register A Channel 0

CNTLB0	2	Control Register B Channel 0
STAT0	4	Status Register Channel 0
TDR0	6	Transmit Data Register Channel 0
RDR0	8	Receive Data Register Channel 0
CRTLA1	1	Control Register A Channel 1
CRTLB1	3	Control Register B Channel 1
STAT1	5	Status Register Channel 1
TDR1	7	Transmit Data Register Channel 1
RDR1	9	Receive Data Register Channel 1

Die einzelnen Bits der Ports:

CNTLA (Control A)

7	6	5	4	3	2	1	0
MPE	RE	TE	Kanal 0: RTS=* Kanal 1: CKA1D	Read: MPBR Write: EFR	MOD2	MOD1	MOD0

MPE   = Multiprocessor Mode Enable
RE   = Receiver Enable
TE   = Transmitter Enable
RTSO* = Request To Send Channel 0 (aktiv Low, nur Kanal 0!)
CKA1D = Asynchronous Clock Channel 1 Disable (nur Kanal 1!)
MPBR = Multiprocessor Bit Receive (nur beim Lesen!)
EFR   = Error Flag Reset (rur beim Schreiben!)
MODx = ASCII Data Format Mode

Anmerkung: Bit 4 (D4) hat bei Kanal 0 eine andere Bedeutung als bei Kanal 1. Bit 3 (D3) bedeutet beim Schreiben etwas anderes als beim Lesen.

CNTLB (Control B)

7	6	5	4	3	2	1	0
MPBT	MP	Read: CTS* Write: PS	PEO	DR	SS2	SS1	SS0

MPBT = Multiprocessor Bit Transmit
MP   = Multiprocessor Mode
CTS   = Clear To Send (nur beim Lesen!)
PS   = Prescale (nur beim Scbreiben!)
PEO   = Parity Evea/Odd
DR   = Divide Ratio
SSx   = Source/Speed Select

STAT (Status)

7	6	5	4	3	2	1	
RDRF	OVRN	PE	FE	RIE	Kanal 0: DCD0* Kanal 1: CTS1E	TDRE	TIE

RDRF = Receive Data Register Full (nur lesbar)
OVRN = Overrun Error (nur lesbar)
PE   = Parity Error (nur lesbar)
FE   = Framing Error (nur lesbar)
RIE   = Receive Interrupt Enable
DCD0* = Data Carrier Detect (aktiv Low, nur lesbar, nur Kanal 0!)
CTS1E = Clear To Send Channel 1 Enable (nur Kanal 1!)
TDRE = Transmit Data Register Empty (nur lesbar)
TIE   = Transmit Interrupt Enable

Einstellung der Kanäle

Über CNTLB wird die Baudrate, der Multiprozessor-Mode und die Parität festgelegt. D0 bis D3 und D5 sind für die Baudrate zuständig. Sie legen fest, durch welche Werte die Taktfrequenz geteilt wird.

D5	1.Teiler	D3	2.Teiler	D2	D1	D0	3.Teiler
0	10	0	16	0	0	0	1
1	20	1	64	0	0	1	2
				0	1	0	4
				0	1	1	8
				1	0	0	16
				1	0	1	32
				1	1	0	64
				1	1	1	externer Takt

Der externe Takt wird von den Pins CKA0 (gemultiplext mit DREQ0*) bzw. CKA1 (gemultiplext mit TENDt) abgenommen, wobei sich die Baudrate aus dem externen Takt geteilt durcb 16 oder 64 (abhängig von D3) eribt. Dies ist der Reset-Zustand. Falls die Baudrate vom System-Takt abgeleitet wird, wird der resultierende Takt über CKA0/1 ausgegeben!

Zwei Rechenbeispiele (bei einem Systemtakt von 9,216 MHz):

19200 Baud: 19200 = 9.316.000 DIV 30 DIV 16 DIV 1 => D5-D0 = 1x0000
  300 Baud: 300 = 9.216.000 DIV 30 DIV 64 DIV 16 => D5-D0 = 1x1100

D4 ist für die Art der Parität zuständig, falls Paritäts-Prüfung gewünscht wird: D4=0: gerade Parität (EVEN); D4=1: ungerade Parität (ODD).

Zu guter Letzt D6 und D7, die für den Multiprozessor-Modus zuständig sind:
Wenn D6=1, dann wird statt des Paritäts-Bits ein Multiprozessor-Bit gesen-
det/empfangen. Wenn D7=1, wird beim Seden das Multiprozessor-Bit auf 1
gesetzt. Das könnte z.B. bedeuten; "Achtung, an alle anderen Prozessoren,
es kommt was für alle!"
Weitere Einstellungen erfolen über CNTLA. Mit D2 bis D0 wird das Daten-
format festgelegt:

Bit	bedeutet
D2=0	7-Bit-Daten
=1	8-Bit-Daten
D1=0	keine Parität
=1	Paritt
D0=0	1 Stopit
=1	2 Stop-Bits

Wenn D3 mit 0 beschrieben wird, werden alle Fehlermeldungen für diesen
Kanal (im Register STAT0/1) gelöscht. Mit D4 kann bei Kanal 0 der Pin RTS0*
gesetzt/gelöscht werden (Request to Send = Gegenseite darf senden). Bei
Kanal 1 hingegen erfolgt hier die Auswahl, ob der Pin CKA1/TEND0* als Takt-
eingang (CKA1, D4=0) oder als Kennung für das Ende eines Blocktransfers
(TEND0, D4=1) benutzt wird.
Mit D5 wird der Sender ein/ausgeschaltet, mit D6 der Empfänger. D7 hat nur
i Multiprozessor-Modus eine Funktion. Wenn n D7=1, werden nur Daten mit
gesetztem MPB-Bit (s. D7 in CNTLB) angenommen, z.B. im Modus: "Empfange
nur Daten für alle Prozessoren!". Bei D7=0 werden alle Daten angenomme. So
ist folgende Kommunikation denkbar: Der Master-Prozessor sendet an alle
Prozessare durch gesetztes MPB-Bit gekennzeichnet, welche beiden Przes-
soren sich unterhalten dürfen. Diese schalten dann D7 auf 0 und wickeln
ihre Kommunikation ohne MPB-Bit ab. Dann schalten sie D7 auf 1 und der
Master-Prozessor wird benachrichtigt, die Sache wieder zu übernehen.
Zu guter Letzt noch die Einstellungen, die über das Status-Register STAT
vorgenommen werden. D0 legt fest, ob ein Interrupt erfolgt, wenn der Kanal
sedebereit ist. Wenn D3=1, wird auch ein Interrupt ausgelöst, wenn ein
Zeichen emfangen wurde oder ein Fehler auftrat. Bei Kanal 1 kann über D2
ausgewählt werden, ob der Pin CTS1*/RXS als CTS* (D2=1, Clear TO Send =
HD64180 darf senden oder als Daten-Epfangs-Pin für CSI/0 (D2=0) dient.
Senden/Empfangen
Der Stats des Kanals kann über STAT befragt werden. D1=1 bedetet, daß
Daten gesedet werden dürfen. Wenn dann ein Byte in das Sende-Register
TDR geschrieben wird, geht D1 auf 0, bis das Zeichen gesendet wurde.
D7=1 bedeutet, daß ein Zeichen empfangen wurde und jetzt aus dem Empfangs-
Register RDR gelesen werden kann oder daß bei Empfangen ein Fehler auf-
getreten ist. Nach dem Lesen des Bytes aus RDR wird D7 gelöscht. Folgende
Fehler werden erkannt:

```
D6  Overrun  (letztes Byte wurde nicht schnell genug aus RDR geholt)
D5  Parity   (das Paritäts-Bit des Zeichens warfalsch)
D4  Framing  (das/die Stop-Bits stimmten nicht)
```

Falls ein Febler auftrat, muß dieser gelöscht werden, indem D3 von CNTLLA
mit 0 beschrieben wird. Achtung: dabei werden dann auch die anderen Ein-
stellungen von CNTLA geändert!
Über D2 aus STAT kann bei Kanal 0 noch der Zustand des Signals "Data
Carrier Detect" abgefragt werden. D2 wird auf 1 gesetzt, wenn das Signal am
Pin DCD0* irgendwann mal HIGH war oder noch ist. D2 wird nach dem Lesen auf
0 gesetzt, wenn inzwischen DCD0*=0 ist. Solange D2=1, wird nichts empfan-
gen. Diese Verbindung ist nur bei bestimmten Modems sinnvoll.
Über D3 aus CTNLA kann das Multiprozessor-Bit des letzten empfangenen
Bytes abgefragt werden. D5 aus CNTLA gibt Ausunft über den Zustand des
Pins CTS*. Für Kanal 0 gilt, daß der Kanal erst sendebereit wird STA0,
D1=1), wenn der Pin CTS0*=0. CTS1 kann nur abgefragt werden und hat nur
einen Einfluß auf die Sedebereitschaft, wenn D2 aus STAT1 auf 1 gesetzt wurde (CTS1E = CTS-Enable Kanal 1).
Clocked Serial Input Output
Diese Schnittstelle dient zur Verbindung mehrerer Prozessoren, wobei die
anderen Prozessoren nicht unbedigt HD6480s sein mssen. Aber sie müssen
über eine gemeinsame (sychronisierte) Taktauelle verfügen und die Sende-/
Empfangs-Operation muß in beiden Prozessoren gleichzeitig beginnen!
Der HD64180 kann nicht gleichzeitig empfangen und seden. Es gibt nur ein
Sede/Empfangregister: TRDR (Transmit/Receive Data Register), Port 0Bh.
Es sind auch nur einige Einstellungen möglich, so daß nur ein Kontroll-
Regiser nötig ist: CNTR (Control), Port 0Ah. Der Aufbau dieses Registers:

```
CNTR  (Control)
7    6    5    4    3    2    1    1
--------------------------------------
```

```
EF   EIE  RE   TE   -    SS2  SS1  SS0
EF  = End Flag  (nur lesbar)
EIE = End Interrupt Enable
RE  = Receive Eable
TE  = Transit Enable
SSx = Speed Select
```

Einstellungen

Es kann nur die Baudrate eingestellt werden. Wenn die Baudrate neu gesetzt wird, sollte nicht gleicheitig ein Sende-/Empfaggs-Zyklus (s.u. gestartet werden. Der System-Takt wird durch die folenden Werte geteilt (bier für 9.216 MHz):

```
SS2 SS1 SS0 Teiler  ergibt Baudrate
-----------------------------------
0   0   0     20      460800
0   0   1     40      230400
0   1   0     80      115200
0   1   1    160       57600
1   0   0    320       28800
1   0   1    640       14400
1   1   0   1280        7200
1   1   1   externer Takt  (kleiner als 460800)
```

Bei 111 wird der extere Takt über den Pin CKS eingelesen; asonsten wid der resultierende Takt hier ausgegeben.

Empfangen/Senden

Zuerst muß abgefragt werden, ob der Sender gerade aktiv ist. Erst wenn D7 aus CNTR gesetzt ist, kann die Operation beginnen. Zum Senden wird das Byte in TRDR geschrieben und dann D4 in CNTR gesetzt. Gleichzeitig wird die Bauarate (wieder) gesetzt. Sie sollte aber nicht verändert werden, da es sonst Sende-Probleme geben kann. Die Operation ist erst beendet, wenn D7 wieder auf 1 geht. Falls D6 gesetzt wurde, wird dann ein Interrupt ausgelöst.

Beim Empfangen wird D5 gesetzt (zusammen mit der Baudrate). Nun beginnt der Empfangs-Vorgang, der immer gleich lang dauert. Wenn D7 wieder auf 1 geht, kann das Byte aus TRDR gelesen werden. Wenn vor dem Empfangen D6 gesetzt wurde, wird ein Interrupt ausgelöst, sobald D7 auf 1 springt. Es ist zu beachten, daß der Empfangs-Pin (RXS) mit CTS* für Kanal 1 gemultiplext ist. Vor dem Empfang muß also über D2 von STAT1 die Funktion "Pin RXS/CTS1* als RXS* ausgewählt werden.

Die Timer des HD64180

Gerald Schröder

Der HD64180 verfügt über zwei programmierbare 16-Bit-Zähler und einen nicht-programmierbaren 8-Bit-Zähler. Der 8-Bit-Zähler wird alle 10 Takte dekrementiert und kann nur gelesen werden, und zwar über Port 18h. Das Beschreiben dieses Ports kann Fehler im Timing des HD64180 verursachen.
Die 16-Bit-Zähler werden alle 20 Takte dekrementiert. Einer der 16-Bit-Zähler kann über den Pin TOUT an einen anderen Chip melden, daß er den Zählerstand 0000 erreicht hat. Beide Zähler können einen Interrupt auslösen, wenn sie 0000 erreichen.

Die Ports

```
Port   Abk.      Name
----------------------------------------------------
0Ch    TMDR0L    Timer Data Register Channel 0 Low
0Dh    TMDR0H    Timer Data Register Channel 0 High
0Eh    RLDR0I    Reload Data Register Channel 0 Low
0Fh    RLDR0H    Reload Data Register Channel 0 High
10h    TCR       Timer Control Register
14h    TMDR1L    Timer Data Reister Cbanel 1 Low
15h    TMDR1H    Timer Data Regigter Channel 1 High
16h    RLDR1L    Reload Data Register Channel 1 Low
17h    RLDR1H    Reload Data Register Channel 1 high
```

TCR (Control)

```
  0    1    2    3    4    5    6    7
TDE0 TDE1 TOC0 TOC1 TIE0 TIE1 TIF0  TIF1
DDE0/1  Timer Decrement Enble Channel 0/1
```

TOC0/1 Timer Output Control (beide für Kanal 1!)
TIE0/1 Timer Interrupt Enable Channel 0/1
TIF0/1 Timer Interrupt Flag Chanel 0/1
Programmierung
Zuerst sollte der Timer gestoppt werden, indem TDE in TCR auf 0 gesetzt wird. Dann das gewünschte Zählintervall in RLDR und (!) TMDR eintragen; die Eintragung in TMDR ist nötig, weil der Zähler sonst bei dem alten Wert staret. Es handelt sich hier um 16-Bit-Werte, so daß bei einem Takt von 9,216 MHz Warteschleifen von ca. 2,2 Microsekunden bis ca. 0,142 Sekunden realisiert werden können; bzw. der Zähler erreicht 460800 (Iatervall 0001) bis 7,03125 (Intervall FFFFh) Mal in der Sekunde den Stand 0000.
Der Zähler fängt erst an zu zählen, wenn TDE in TCR auf 1 gesetzt wird. Gleichzeitig wird über TIE ausgewählt, ob ein Interrupt ausgelöst werden soll, wenn TMDR den Zählestand 0000 erreicht. Für Kanal 1 kann über TOC noch augewählt werden, welche Reaktion an Pin TOUT (Timer Output) beim Erreichen von 0000 erfolgt (Achtung: TOUT ist mit Adreßbit A18 gemulti-plext):

TOC0	TOC1	Reaktion TOUT
0	0	keine (Pin als A18)
0	1	TOUT wird getoggelt (0 -> 1) (1 -> 0)
1	0	TOUT auf 0
1	1	TOUT auf 1

Der Zähler TMDR darf zwischendurch ausgelesen werden, aber nur in der Reihenfolge Low-Byte und dann High-Byte. Sobald TMDR den Zählerstand 0000 erreicht, wird der in RLDR stehende Wert in TMDR geladen und es wird wieder mit dem Zählen angefangen. Außerdem wird TIF in TCR gesetzt. Falls vorher TIF gesetzt wurde, wird auch ein Interrupt ausgelöst.
TIF wird erst gelöscht, wenn zuerst (!) TCR und dann TMDR (in der Reihen-folge Low-/High-Byte) gelesen werden. Ansonsten bleibt es gesetzt, was bei der Rückkehr von einer Interrupt-Routine sofort einen neuen Interrupt ver-ursacht!

Die zusätzlichen Befehle das HD64180

Helmut Bernhardt

Wer in seinem Cmoputer den Z80 durch einen HD64180 ersetzt hat, wird sicher schon ein plötzliches Booten oder einen Absturz bei Aufruf von bestimmten Programmen bemerkt haben, die vorher mit dem Z80 einwand-frei zusamnmengearbeitet haben. Zu solchen Programmen zählen REZILOG unter CP/M oder ZEUS unter NEWDOS/80. Der Fehler liegt nicht am Pro-zessor sondern am Programm. Der Programmierer hat dort Befehle benutzt, die von Zilog nicht dokumentiert sind aber trotzdem auf dem Z80 zur Verfügung stehen. Es sind die die Illegals mit Doppelbyte-Op--codes.
Hitachi hat nun andere Vorstellungen entwickelt, wie diese freien Op-codes verwendet werden können, und eine Reihe dieser Codes den in An-hang zusammengestellten Befehlen zugeordnet. Gegenüber den Illegals des Z80 sind das sinnvolle Ergänzungen, die der Architektur des HD64180 entgegenkommen und u.a. eine elegante Programmierung der in-ternen Ports der CPU ermöglichen. Die ebenfalls hinzugekommene Multi-plikation ist leider nur auf 8Bit-Faktoren beschränt geblieben.
Der wichtigste Befehl ist OTIMR, der eine Anzahl von Bytes, die in B vorgegeben wird, ab einer Adresse im Speicher, auf die HL zeigt, an aufeinander folgende interne Zeropake-Ports ausgibt, wobei die Adres-se des ersten Ports in C vortugeben ist. Damit kann z.B. der DMA-Con-troller schnell mit Parmetern für eine Übertragung versorgt werden.
In mc 1/87, S.74 wurde gezeigt, wie die Illegals durch nrmale Befehle ersetzt werden können. Danach kann man von Hand entprechende Prgram-me auf normale Befehle umstellen. Was der Normalverbraucher zu Fuß macht, läßt der Experte die CPU zur Laufzeit des Illegal-verseuchten Programms selbst machen. Eventuell schreibt Gerald dazu nochmal was.
Bei Erreichen eines unbekannten Opcodes wird der höchstpriorisierte TRAP-Interrupt ausgeführt, der einen CALL 0000H gleichkommt. Er ist mit einem RESET vergleichbar, nur dass kein richtiges /RESET-Signal er-folgt, das die gesamte Hardware zurücksetzt. Dadurch wird auch kein Floppy-Controller in den Einschaltzustand zurückgesetzt und auch kein ausgeblendetes EPROM wieder eingeschaltet.
Bei 0000H liegt dann ROM vor. Dort kann bei der Initialisierung ein

Sprung auf die TRAP-Service-Routine gepatcht werden. Diese Routine kann mit dem Stack und den Flags im TRAP-Register herausfinden, ob ein TRAP die Ursache des Aufrufs war, oder ob die Adresse 0000H durch Software aufgerufen wurde; bei einem Trap kann sie ermitteln, welches Byte im Programm den TRAP auslöste und dem entprechend alternative normale Befehle abarbeiten, und anschließend wieder die Kontrolle an das unterbrochene Programm zurückgeben.
Wenn eine solche TRAP-Service-Routine nicht vorliegt, bedeutet der CALL 0000H einen Warmstart unter CP/M oder einen Absturz unter NEWDOS/80, weil die BOOT-Routine im ROM nicht den Floppy-Controller auf Sinle Density zurücksetzt (deshalb funktioniert beim GENIE auch der Befehl BOOT oder ein JP 0000H nicht, wenn mit einem Doubler gearbeitet wird; der TRS 80 decodiert die Adresse 0000H und erzeugt darus ein /RESET-Signal).
Natürlich funktioniert der TRAP-Interrupt nur bei den Z80-Illegls, für die der HD64180 keine neuen Befehle eingeführt hat. Wenn ein illegaler Opcode auftaucht, der einem HD64180-Befehl entpricht, so wird dieser Befehl nach Hitachi's Vorstellung ordnungsgemäß ausgeführt. Der Erfolg ist dann aber nicht der, den sich der Illegal--Programmierer einst gedacht hat.
Zusätzliche Befehle des HD64180
Multiplikation
```
ED 4C      MULT  BC    ;BC := B * C          |  17 Taktzyklen
ED 5C      MULT  DE    ;DE ;= D * E          |
ED 6C      MULT  HL    ;HL := H * L          |
ED 7C      MULT  SP    ;SP := SPhigh * SPlow |
```
Logisches AND zum Setzen der Flags, ohne die Register zu verändern
```
ED 04      TEST  B     ; A and B             |  7 TaktzyXlen
ED 0C      TEST  C     ; A and C             |
ED 14      TEST  D     ; A and D             |
ED 1C      TEST  E     ; A and E             |
ED 24      TEST  H     ; A and H             |
ED 2C      TEST  L     ; A and L             |
ED 3C      TEST  n     ; A and A             |
ED 34      TE5T  (HL)  ; A and (HL)          |  10 Takt2yklen
ED 64 nn   TEST  nn    ; A and nn            |   9 Taktzyklen
ED 74 pp   TESTIO pp   ; (C) and pp          |  12 Taktzyklen
```
Zeropage I/O-Befehle
```
ED 00 pp    IN0  B, {pp)    ; B := (pp)      |  12 Taktzyklen
ED 08 pp    IN0  C, (pp)    ; C := (pp)      |
ED 10 pp    IN0  D, (pp)    ; D := (pp)      |
ED 18 pp    IN0  E, (pp)    ; E := (pp)      |
ED 20 pp    IN0  H, (pp)    ; H := (pp)      |
ED 28 pp    IN0  L, (pp)    ; L := (pp)      |
ED 30 pp    IN0  F, (pp)    ; S,Z,P <- (pp)  |
ED 38 pp    IN0  A, (pp)    ; A := (pp)      |
ED 01 pp    OUT0 (pp),B     ; (pp) := B      |  13 TaXt2yklen
ED 09 pp    OUT0 (pp),C     ; (pp) := C      |
ED 11 pp    OUT0 (pp),D     ; (pp) := D      |
ED 19 pp    OUT0 (pp),E     ; (pp) := E      |
ED 21 pp    OUT0 (pp),H     ; (pp) := H      |
ED 29 pp    OUT0 (pp),L     ; (pp) := L      |
ED 39 pp    OUT0 (pp),A     ; (pp} := A      |
ED 83       OTIM     ; (C) := (HL), HL:= HL+1, C := C+1, B := B-1
                                                 14 Taktzyklen
ED 93       OTIMR    ; (C)..(C+B) := (HL)..(HL+B), B:=0
                     ; oder: (C):=(HL), C:=C+1, HL:=HL+1, B:BB-1
                     ; wiederhole bis B=0       B*16 Taktzyklen
ED 8B       OTDM     ; (C):=(HL), HL:=HL-1, C;=C-1, B:=B-1
                                                 14 Taktzyklen
ED 9B       OTDMR    ; (C)..(C-B) := (H).. (HL-B),  B:=0
                     ; oder: (C):=(HL), C:=C-1, HL:=HL-1, B:=B-1
                                                 B*16 Taktzyklen
ED 76       SLP      ; Sleep: Einstellen aller Aktivitäten incl.
                     ; Refresh, bis /RESET oder /IMT eintritt
```